Physics-based Modeling Techniques for Analysis and Design of Advanced Suspension Systems with Experimental Validation

Alireza Farjoud

Dissertation submitted to the faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
In
Mechanical Engineering

Mehdi Ahmadian, Chair
Muhammad R. Hajj
Michael W. Hyer
Daniel J. Inman
Steve C. Southward
Saied Taheri

January 17, 2011
Blacksburg, Virginia

Keywords: Vehicle Systems and Suspension Modeling, Smart Materials and Intelligent Systems, Magneto Rheological (MR) fluids, Rheology of Non-Newtonian Fluids, Experimental Testing and Validation

Physics-based Modeling Techniques for Analysis and Design of Advanced Suspension Systems with Experimental Validation

Alireza Farjoud

ABSTRACT

This research undertakes the problem of vibration control of vehicular and structural systems using intelligent materials and controllable devices. Advanced modeling tools validated with experimental test data are developed to help with understanding the fundamentals as well as advanced and novel applications of smart and conventional suspension systems.

The project can be divided into two major parts. The first part is focused on development of novel smart suspensions using Magneto-Rheological (MR) fluids in unique configurations in order to improve efficiency, controllability, and safety of today's vehicles. In this part of the research, attention is paid to fundamentals as well as advanced applications of MR technology. Extensive rheological studies, both theoretical and experimental, are performed to understand the basic behaviors of MR fluids as complex non-Newtonian fluids in novel applications. Using the knowledge obtained from fundamental studies of MR fluids, unique application concepts are investigated that lead to design, development, and experimental testing of two new classes of smart devices: *MR Hybrid Dampers* and *MR Squeeze Mounts*. Multiple generations of these devices are built and tested as proof of concept prototypes. Advanced physics-based mathematical models are developed for these devices. Experimental test data are used to validate the models and great agreement is obtained. The models are used as design tools at preliminary as well as detailed design stages of device development. The significant finding in this part of the research is that MR fluids can deliver a much larger window of controllable force in squeeze mode compared to shear and valve modes which can be used in various applications.

The second part of the research is devoted to the development of innovative design tools for suspension design and tuning. Various components of suspension systems are studied and modeled using a new physics-based modeling approach. The component of main interest is the shim stack assembly in hydraulic dampers which is modeled using energy and variational methods. A major finding is that the shims should be modeled individually in order to represent the sliding effects properly when the shim stack is deflected. Next, the individual component models are integrated into a full suspension model. This model is then used as a tool for suspension design, synthesis, and tuning. Using this design tool, suspension engineers in manufacturing companies and other industrial sections can easily perform parametric studies without the need to carry out time consuming and expensive field and laboratory tests.

In memory of my grandfather,

MOHAMMAD KARIM ZIAEI

And, to my beloved parents,

FATEMEH ZIAEI

MOHAMMAD REZA FARJOOD

ACKNOWLEDGEMENTS

Professor Mehdi Ahmadian, CVeSS director and my research supervisor, has been certainly a great advisor and an effective leader during the course of my PhD research. He has always been a source of fruitful and valuable guidance and knowledge in my research. However, my learning from Dr. Ahmadian goes well beyond common research and technical topics and spans other aspects of life, in general. I have always enjoyed my conversations with Dr. Ahmadian about various aspects of life besides academia. I would like to express my deepest gratitude to him for his endless support, care, friendship, and guidance. Also, for his support and encouragement in helping me start the path of what I have always liked to be, a teacher, I am deeply indebted.

I would like to thank Dr. Muhammad Hajj, Dr. Michael Hyer, Dr. Daniel Inman, Dr. Steve Southward, and Dr. Saied Taheri for their advice and for serving on my graduate committee. I would like to especially thank Dr. Michael Hyer for his advice and guidance on modeling of the shim stack assemblies.

I am sincerely thankful to my dear friend and colleague, Mr. Michael Craft, CVeSS senior research scientist, for all his help during my research work. He is a great source of practical and novel ideas and I have always enjoyed our discussions. I am also very thankful to Dr. Nima Mahmoodi, for his friendship and significant contributions in the perturbation solution of the MR fluid flow. Thanks are also due to Mr. Ryan Cavey, my colleague and friend, for building the squeeze mode rheometer and conducting the first set of experiments. I would like to thank Mr. Clement Nagode, my colleague and good friend, who was always willing to offer help with a smile on his face. We shared a lot of interests both in research and hobbies. I am thankful to Mr. William Burke, my friend, for his help with the hybrid damper design and testing. Thanks are also due to Mrs. Sue Teel, CVeSS office manager, for providing me with a friendly and productive work environment. I also would like to thank my friend, Mr. Xinjie Zhang, from Jilin University, China, who helped me with some of the MR squeeze mount experiments. I also thank Dr. Florin Marcu, my good friend, for his help with the data acquisition system setup.

Dr. Mehdi Nikkhah and Mr. Amin Zareain-Jahromi, my great friends and roommates, deserve a big thank you for all their support and friendship during the past four years. Their friendship during all these years made this journey truly memorable for me.

Funding is a major part of every research activity and I would like to sincerely thank the Mechanical Engineering Department at Virginia Tech, General Motors, and NASA for supporting me during my research. I am also thankful to Dr. Fernando Goncalves of Lord Corp. for donating the MR fluids needed for testing and experiments.

Most importantly, I would like to sincerely thank my wonderful family for their never-ending love and support throughout my studies. They have always been a source of motivation and encouragement for me. Therefore, I would like to dedicate this dissertation to my parents, M. R. Farjood and F. Ziaei, and my beloved brothers Ehsan and Amir Sina.

Alireza Farjoud

November 2010

Blacksburg, Virginia

Contents

List of Figures

List of Tables

Chapter 1
Introduction

The research work presented in this document undertakes the problem of vibration control of vehicle and structural systems. Advanced modeling techniques validated with experimental test data are developed in this research to help with understanding the fundamentals as well as applications of smart suspensions.

1.1. Motivation

Magneto-Rheological (MR) fluids are being used in many commercially available smart devices for vibration control of various systems. They are attractive to these applications because of their unique rheological behavior. They can change from a free flowing fluid to a solid like material in milli-seconds when a magnetic field is applied. This property can be used in many applications. MR dampers and MR rotary brakes have taken advantage of this property to be among the most widely used MR controllable devices for vibration control of various systems. Although these devices have proven to be very effective in vibration mitigation of many small-scale systems, there has not been much progress in expanding their application to larger systems. The reasons are the high density and high cost of MR fluids. Using MR fluids in large scale applications requires a large amount of MR fluid which makes the device extremely heavy and expensive. In this research, novel and efficient ways of application of MR fluids in intelligent devices are investigated in order to extend the range of applications to larger scale systems. For this purpose, it is proposed that MR fluids be used in another flow mode not currently being used in any intelligent MR device. This flow mode is the squeeze mode which is the least understood flow mode of MR fluids.

The primary application in mind for smart devices developed in this research project is vehicle systems. Devices developed as proof of concepts will eventually be used as a component in a smart suspension system. Therefore, it is essential to have a good design and synthesis tool for study of suspensions. The rest of this research work was devoted to development of advanced and novel design tools for vehicle suspension systems. Extensive theoretical and experimental studies were done to develop a reliable and robust design and synthesis tool for vehicle suspensions. This tool will benefit the automotive industry in terms of cost, time, and understanding the behavior of advanced suspensions used in today's vehicles. Using this tool will eliminate a lot of laboratory and field testing which are time consuming and costly.

1.2. Objectives

The main objectives of this research can be summarized as,

- Investigation of novel applications of smart materials for vibration control of vehicular and structural systems
- Understanding the fundamentals of the behavior of smart materials used for this study
- Development of mathematical models that can be used as tools to design and develop novel intelligent devices
- Development of mathematical models for design and tuning of smart suspensions as one of the applications of smart materials

1.3. Approach

In order to use MR fluids in squeeze mode in intelligent MR devices, basic and fundamental behaviors of MR fluids in squeeze mode have to be fully understood. Therefore, attention was paid to rheological study of MR fluids in squeeze mode. Both experimental and theoretical studies have been performed. It was found that MR fluids in squeeze mode are capable of providing a much larger window of controllable

force compared to other flow modes currently in use. Hence, MR fluids in squeeze mode are the perfect candidates for large scale and high force capacity intelligent devices.

For the rheological study of MR fluids in squeeze mode, a novel "*Squeeze Mode Rheometer"* was designed, built, and used for testing various MR fluids in the squeeze mode (Figure 1-1). The rheometer was capable of testing MR fluids under different conditions. Experimental data obtained from the squeeze mode rheometer testing showed that MR fluids are stronger in the squeeze mode compared to other flow modes.

Applications of MR fluids in the squeeze mode for intelligent devices were also studied extensively during this research project. Two novel devices were designed and developed to study the feasibility of application of MR fluids in the squeeze mode for vibration control applications. These two devices are, *"MR Hybrid Damper"* and *"MR Squeeze Mount"*. Both theoretical and experimental studies were performed on these two devices (Figure 1-1). Experimental data showed that these two devices can be used as high force capacity devices. Although, the development and testing of the MR hybrid damper was challenging and future applications of MR hybrid dampers as high force capacity smart dampers require design improvements, the preliminary test results served as a proof of concept for intelligent devices used in high force capacity applications. Some of these challenges will be mentioned in later chapters.

The MR squeeze mount was extensively tested and different generations of MR squeeze mounts were developed. Test results were very promising and it was shown that MR squeeze mounts can provide a wide controllable force range along a very short stroke suitable for many applications such as engine mounts and impact dampers.

Extensive theoretical study and mathematical modeling work was done to develop a reliable design tool for the MR squeeze mounts. Various models were developed and all of them were validated using test data. The difference between the models is in their capability in predicting only one or more flow parameters. A simple model was developed and validated that could predict the total squeezing force

generated by the MR squeeze mount. This model is sufficient for initial stages of MR mount design. A more detail model was also developed for design improvements and optimization of MR mount performance. In addition to total force, this model was capable of predicting flow field, shear rate distribution, pressure distribution, and yield surface and plug flow region locations.

The next step of this research was the investigation of suspension systems beyond MR technology in a more generic fashion. This step is important because it provides essential information and insight about the fundamentals of suspension systems performance which will become crucial when considering smart and controllable suspensions. This stage of the project resulted in a design and synthesis tool for suspension systems which was initiated by modeling individual components of a suspension. Then, all various modeled components were assembled into a full mathematical model using a quarter-car system. Finally, a computer program was developed consisting of the parameters of all components. This program was used as a design tool, as shown in Figure 1-2.

The hydraulic damper was the focus of this modeling work. Extensive research was performed to develop an accurate mathematical model of a hydraulic damper. Various parts of a hydraulic model were modeled and assembled. However, the shim stack assembly was of the utmost interest. This is because no accurate mathematical model of shim stack deflection exists in open literature to date. All the modeling works on hydraulic dampers have considered the shim stack as a single blow-off valve or have modeled only one deflecting disk instead of a shim stack. Energy and variational methods were used in this research to model the shim stack deflections. It was found that significant error occurs as a result of considering the shim stack as a single disk and consequently ignoring the shear deformations.

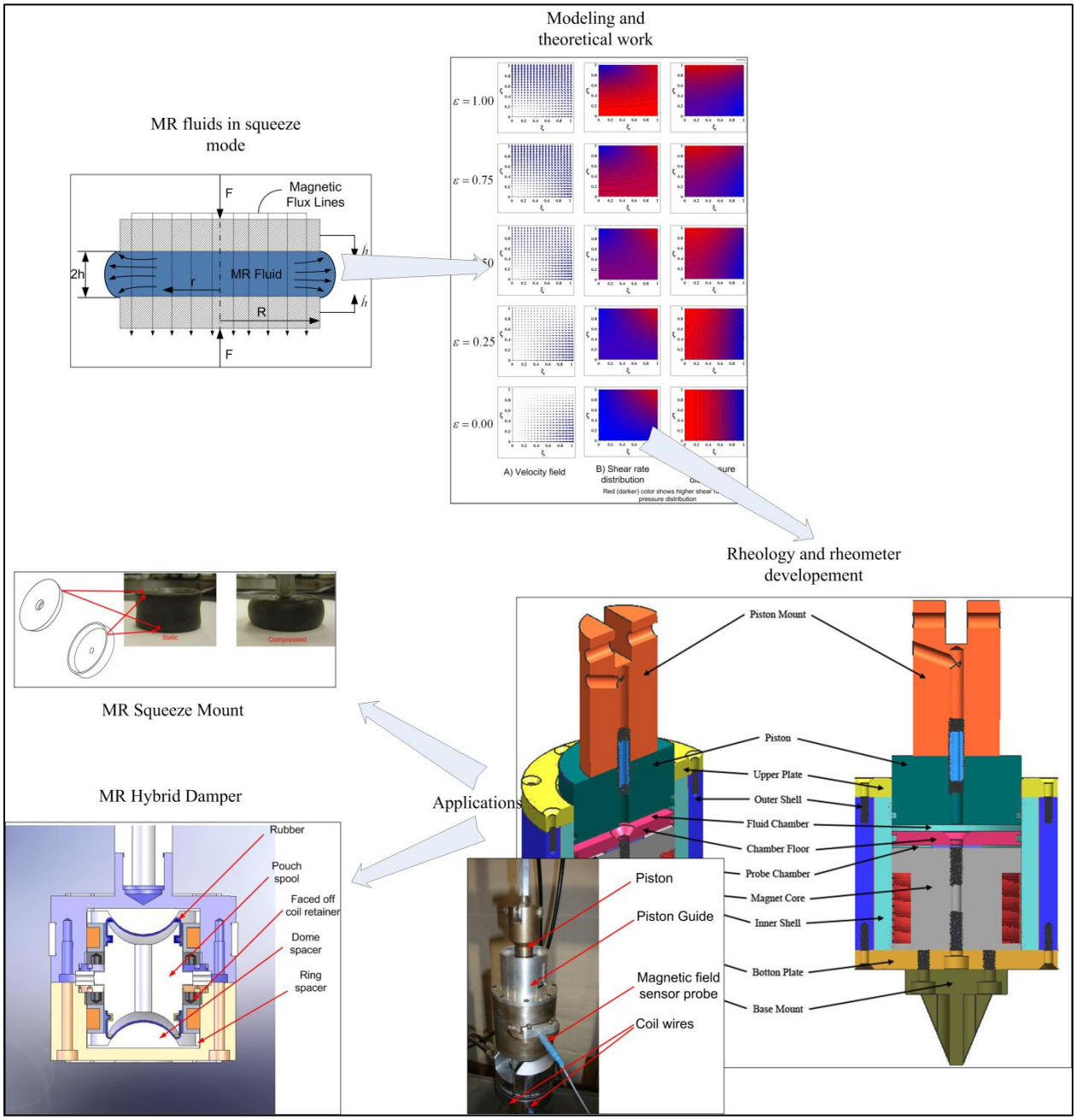

Figure 1-1 Theoretical and Experimental works performed to better understand the fundamentals and applications of MR fluids in squeeze mode

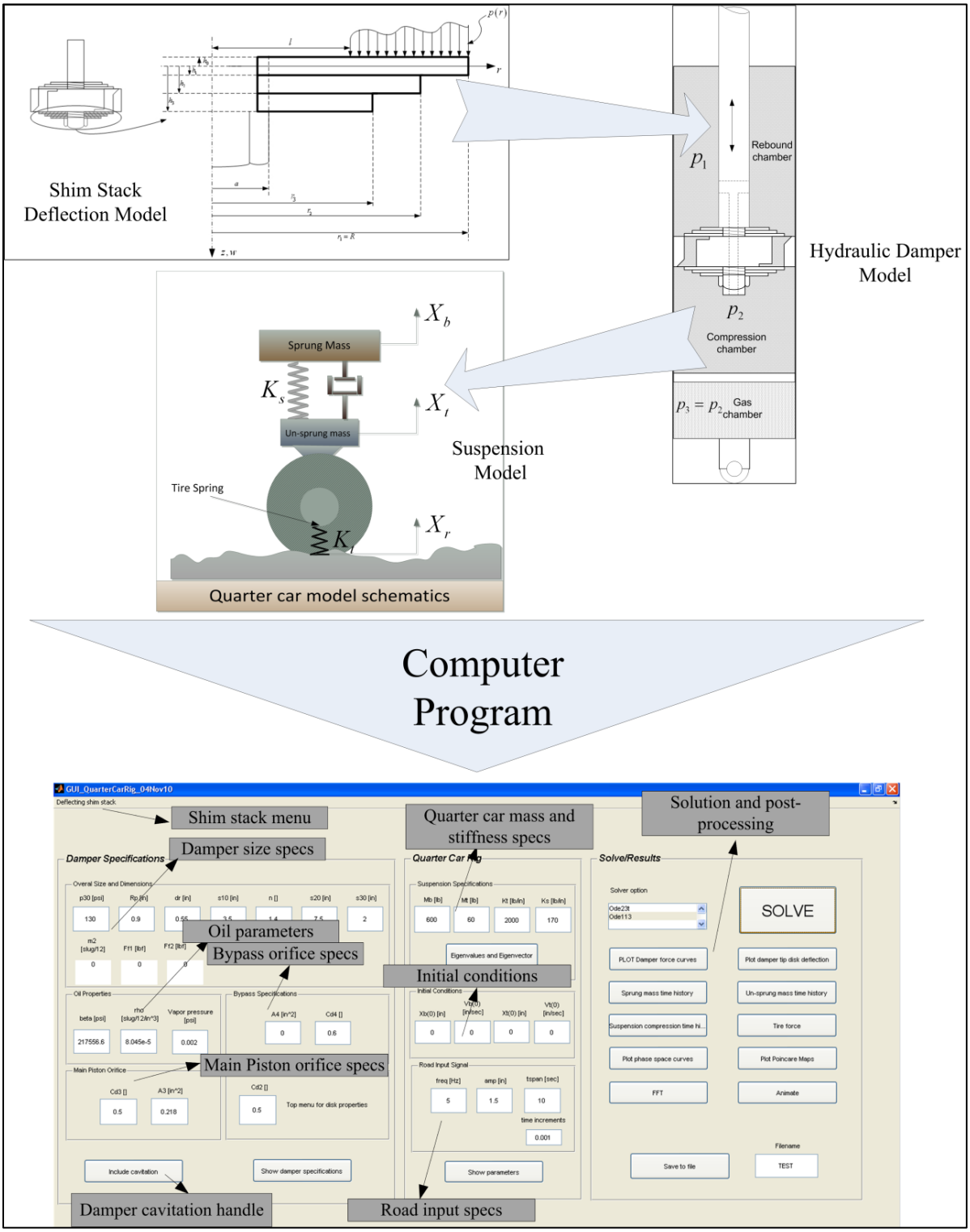

Figure 1-2 Various suspension components were modeled and integrated into a full suspension model

The shim stack model was integrated into the hydraulic model of a mono-tube damper and this model was validated using experimental test data. Excellent agreement between test data and model predictions were obtained.

The last step in developing a design tool for suspension analysis was to integrate all components into a suspension model. A quarter-car model was used. The hydraulic model with the incorporated shim stack model was integrated into the quarter-car model along with other major components. This mathematical model was used to study the effects of various suspension parameters on the overall dynamic behavior of a vehicle. The full suspension model was then used as a powerful design and synthesis tool for suspension design and tuning. Using this design tool, suspension engineers in manufacturing companies and other industrial sections can easily perform parametric studies without the need to carry out time consuming and expensive field and laboratory tests.

1.4. Outline

Chapter 2 is the literature review. MR fluids and MR technology are introduced in this chapter and major contributions in this field are mentioned. Literature reviews for other topics included in the document are presented in their corresponding chapter as they become more relevant.

Mathematical modeling of MR fluids in squeeze mode is presented in Chapter 3. This chapter introduces the two mathematical models used to predict flow parameters in squeezing flows of MR fluids.

Design and experimental testing of the squeeze mode rheometer are presented in Chapter 4. Design concepts of the squeeze mode rheometer as well as its major components are introduced and primary test results of various MR fluids in the squeeze mode using the squeeze mode rheometer are presented and discussed.

Chapter 5 is devoted to introducing the two novel devices that use MR fluids in the squeeze mode. This chapter covers the design principles and concepts of the MR hybrid damper and the MR squeeze mount.

Various components of these devices are introduced and experimental test data are presented and discussed.

Shim stack assemblies and their deflection analysis are presented in Chapter 6 and the mathematical model of shim stack deflection is verified.

Chapter 7 explains modeling of hydraulic mono-tube dampers. Various components of a hydraulic damper are modeled separately and then assembled into a full functional mathematical model. The shim stack deflection model presented in Chapter 6 is also integrated in the hydraulic model. The mathematical model developed in Chapter 7 is validated using experimental test data and the results are presented.

In Chapter 8, the suspension model is discussed and various components are integrated into a single model. This model is used to investigate a sample case and effects of some of the suspension parameters are studied.

The work performed in this research project is summarized and concluded in Chapter 9.

1.5. Contributions

The major contributions of this research can be summarized as follows.

- *Performing basic theoretical and experimental studies on MR fluids in the squeeze mode.* There is very little literature about the fundamental characteristics of MR fluids in the squeeze mode. This step was extremely important during this research because it provided significant findings to help with design and development of novel MR devices that use MR fluid in the squeeze mode.
- *Development of advanced mathematical models for MR fluids in the squeeze flows.* There is no universally accepted mathematical model for the flow of MR fluids in the squeeze mode. In this research, advanced mathematical models were developed and experimentally validated. These models can be used as design tools for development of novel squeeze MR devices.

- *Design and development of two classes of MR devices that use MR fluids in squeeze mode.* MR hybrid dampers and MR squeeze mounts were designed, built, and tested during this research to investigate the applications of MR fluids in the squeeze mode. These devices are among the first intelligent MR devices that use MR fluid in the squeeze mode.
- *Mathematical modeling of shim stack assemblies in hydraulic dampers.* There is no available mathematical model for the deflection of shim stack assemblies to date. In all the previous models, shim stacks were modeled using simplifying assumptions that ignore the sliding effects of shims as they are deflected. In this work, this sliding behavior was included in the mathematical model and it was shown that the simplifying assumption will introduce a significant amount of error in the analysis.
- *Mathematical modeling of hydraulic dampers.* A detail mathematical model of mono-tube dampers was developed that included the main components of a commonly used hydraulic damper. This model included the shim stack assembly deflection model and was used to study the effects of various shim stack parameters on damper performance.
- *Development of a suspension design and synthesis tool.* The mathematical model of the hydraulic damper was integrated into a suspension model. This suspension model can be used as a design and synthesis tool for design and tuning of various types of suspension systems.

Chapter 2
Literature Review

This chapter covers the fundamentals necessary for this research work. This research is focused around MR technology and advanced suspension systems modeling and testing. Therefore, in this chapter, MR fluids, MR technology, and associated research works are discussed. Since the squeeze mode flow of MR fluids is the focus of this research, a separate section is devoted to squeeze flow of MR fluids and squeeze flow rheometry at the end of this chapter.

This research also includes different research studies on various suspension systems, components, and other dynamic systems. Literature reviews of these topics are presented in their relevant chapters.

2.1. Field Responsive Fluids

Field responsive fluids are a type of fluids that respond rheologically to a magnetic or electric field. These fluids are Magneto-Rheological (MR) fluids, Electro-Rheological (ER) fluids, and Ferro fluids.

2.1.1 Magneto-Rheological (MR) Fluids

Magnetorheology is a branch of rheology that deals with flow and deformation of materials under an applied magnetic field. The discovery of MR fluids is credited to Jacob Rabinow [1]. Magnetorheological (MR) fluids are suspensions of non-colloidal (~0.05-10 μm), multi-domain, and magnetically soft particles in organic or aqueous liquids. Many different ceramic metals and alloys have been described and can be used to prepare MR fluids as long as the particles are magnetically multi-domain and exhibit low levels of magnetic coercivity. Particle size, shape, density, particle size distribution, saturation magnetization, and coercive field are important characteristics of the magnetically active dispersed phase.

Other than magnetic particles, base fluids, surfactants, anticorrosion additives are important factors that affect rheological properties, stability and redispersibility of the MR fluid.

In the "off" state, in terms of their consistency, MR fluids appear similar to liquid paints and exhibit comparable levels of apparent viscosity (0.1 to 1 Pa.s at low shear rates) [2]. Their apparent viscosity changes significantly within a few milliseconds when the magnetic field is applied. The change in viscosity is completely reversible when the magnetic field is removed*. Once a magnetic field is applied, it induces a dipole in each of the magnetic particles. The inter-particle forces originating from the magnetic interactions lead to a material with higher apparent viscosity. This dipolar interaction is responsible for the chain like formation of the particles in the direction of the field (Figure 2-1).

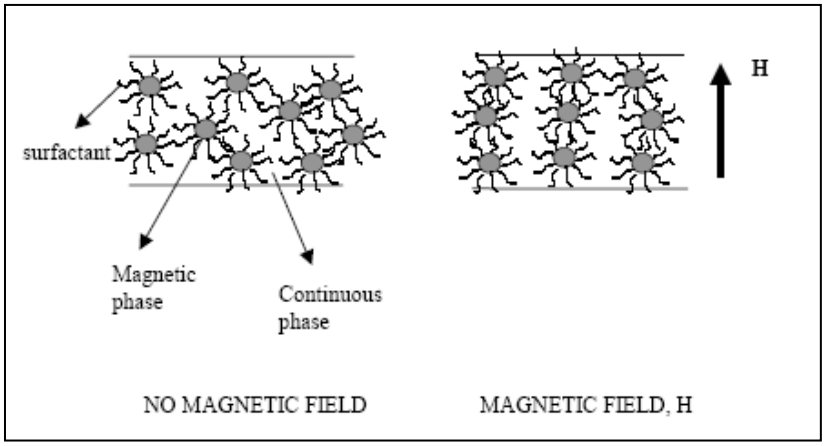

Figure 2-1 Schematic of the formation of chain-like formation of magnetic particles in MR fluids in the direction of an applied magnetic field, (Genc, S., *Synthesis and Properties of Magnetorheological Fluids*, in *School of Engineering*. 2002, University of Pittsburgh. [3], used under fair use)

It is also believed that in addition to magnetic interactions between two particles, the formation of the chains contribute to a certain level of increase in the apparent viscosity. Particles held together by magnetic field and the particle chains resist a certain amount of shear stress without breaking, making

* As will be discussed in Chapter 5, if particle aggregation (clumping behavior) is seen in the MR fluid, this reversibility is reduced.

them behave like a solid. When this shear stress exceeds a critical value, the structure breaks and the material starts to flow. MR fluid effect is often characterized by Bingham Plastic model which will be discussed in the succeeding sections. The critical value of the shear stress necessary to break the structure is the "yield stress" of the material.

Response time of MR fluids to an applied magnetic field is in the order of milliseconds. Note that this is the response of the material itself to a magnetic field and is not a device response time where the effects of the magnetic circuits are included [4, 5].

For MR fluids, materials with the lowest coercivity and highest saturation magnetization are preferred, because as soon as the field is taken off, MR fluid should come to its demagnetized state in milliseconds. Due to its low coercivity and high saturation magnetization, high purity carbonyl iron powder appears to be the main magnetic phase of most practical MR fluid compositions.

2.1.2 Electro-Rheological (ER) Fluids

ER fluids are suspensions of electrically polarizable particles dispersed in electrically insulating oil [6]. ER fluid is typically composed of 0.5 to 100 μm particles of cornstarch, silica, barium titanate or semiconductors [6]. For particles such as silica, polyelectrolytes need to be added to cause the absorption of water onto the particulate material to enhance the ER effect, thus increasing the electrostatic force of attraction between the particles. Water also creates a conductive layer on the surface of the particles in which the ions in the water can drift in response to an electric field. These materials are called extrinsically polarizable materials in which the ER effect results from interfacial polarization. ER effect decreases as the amount of absorbed water decreases. Therefore, at temperatures of approximately 50 degrees Celsius (50^{o} C), ER activity decreases significantly and thus the temperature instability limits the potential use of the ER fluid. Materials such as ferroelectrics, inorganics, semiconductor polymers, metals, coated conductors, and liquid crystals have also been reported as producing water-free ER suspensions and these materials are called intrinsically polarizable materials and they function by bulk

polarization or interfacial polarization. They have lower thermal coefficient of yield stress which may help expand the temperature range of ER activity.

Similar to MR fluids, upon the application of electric field, particles become polarized and the local electric field is distorted. The polarizability of the particles is increased by the migration of the mobile charges to areas with greatest field concentration. This gives rise to larger dipole moments that attract one another and cause the particles to form chains in the direction of the field. ER fluid is characterized by the Bingham plastic model in which the change in viscosity from particle chain interactions under shear corresponds to the yield stress. Many of the models, developed for ER fluids can be adopted for MR fluids in low magnetic fields. At high magnetic fields, however, linear models used to treat ER fluids are no longer valid for MR fluids, due to the non-linearity and magnetic saturation of particles. ER fluids were mostly developed for valves, mounts, clutches, brakes, and dampers. However, not much progress has been made in their commercialization.

2.1.3 Ferrofluids

Ferrofluids also known as magnetic liquids, are colloidal suspensions of ultra-fine (typically 5-10 nm), single domain magnetic particles such as iron oxides (Fe2O3, Fe3O4), Mn-Zn ferrites [7], Fe and Co in either aqueous or non-aqueous liquids. Since the particle size of the magnetic phase is very small, under ordinary field strengths, thermal agitation gives rise to Brownian forces that can overcome the alignment of the dipoles. Thus, MR fluids are based on ferromagnetic particles and ferrofluids are based on super paramagnetic materials. Instead, ferrofluids experience a body force on the entire material that is proportional to the magnetic field gradient. Ferrofluids exhibit field dependent viscosity but they exhibit no yield stress ($\tau_y = 0$) under magnetic fields. Ferrofluids are used in rotary seals, magnetic bearings, and motor dampers [8, 9].

2.2. Comparison of Field Responsive Fluids

More recently, MR fluids have gained considerably more attention than their electric analogue, electrorheological (ER) fluids.

One of the advantages of MR fluids is the higher yield stress value than ER fluids. The reason for having higher yield stress for MR fluids is the higher magnetostatic energy density, $\mu_0 H^2$ of MR fluids, compared with electrostatic energy density, $\varepsilon_0 E^2$ of ER fluids, where, μ_0 is the permeability of free space, H is the magnetic field, and E is the electric field. Low voltage power supplies for MR fluids and relative temperature stability between –40 and $+150^{o}$ C make them more attractive materials than ER fluids. It is important to note that ER fluids can suffer from material instabilities such as dielectric break down under high fields, but MR fluids have high material stabilities. High material stability means such factors as extreme temperatures, impurities or excessive magnetic fields have a negligible effect on the material rheology. This high material stability is also referred to as robustness. Because of this robustness, industrial applications of these materials have been successful without the need for extensive basic research.

The ultimate magnetic property is "ductile". This ductile behavior is associated with Bingham behavior. This means that as the magnetic field is increased, the magnetization of the particles saturates and any increase in magnetic field will cause a diminishing increase in the magnetorheological effect. The ultimate electric property in ER fluids, however, is contrary to MR fluids in that ER fluids fail electrically in a "brittle" manner. When the applied electric field exceeds a maximum value, the ER material breaks down via an electro-static discharge and the electrorheological effect is momentarily interrupted.

A long time scale is often necessary to develop or dissipate a strong magnetic field. Controllable devices that use MR fluids require electro-magnetic circuits to generate variable magnetic fields. These electro-magnetic circuits require approximately L/R seconds to develop their full magnetic fields, where, L is the

inductance of the electro-magnetic circuit and *R* is the resistance. For large systems, this development time can be in the order of 0.25 seconds. This time scale restricts the control implementation to low frequency or static applications. This time scale is also in contrast to those of ER fluid devices. These devices use electrodes whose electric field development time is equal to *RC* and have a time scale response in the order of micro seconds. One of the reasons for extensive research in ER fluids is to take advantage of their very short time scale.

Physically heavy apparatus are necessary for MR fluids. Iron used in the magnetic particles of the fluid and iron of the electro-magnetic circuit can become heavy and cumbersome for large systems. ER devices on the other hand, are relatively light because electric fields can be generated by thin sheets of metal acting as electrodes.

Ferrofluids do not exhibit a yield stress, but show an increase in their viscosity. The viscosity under an applied magnetic field increases almost twice as much as the viscosity when there is no magnetic field applied. Since ferrofluids are synthesized by colloidal magnetic particles, these fluids are more stable than MR fluids which are based on non-colloidal magnetic particles. A comparison of MR, ER, and ferrofluids is summarized in Table 2-1.

Table 2-1 Comparison of field responsive fluids, (Genc, S., *Synthesis and Properties of Magnetorheological Fluids*, in *School of Engineering*. 2002, University of Pittsburgh. [3], used under fair use)

	MR Fluids	**ER Fluids**	**Ferrofluids**
Particulate material	Iron, ferrites, etc.	Zeolites, Polymers, SiO_2, $BaTiO_3$	Ceramics, Ferrites, iron, cobalt, etc.
Particle size	0.1-10μm	0.1-10μm	2-10nm
Suspending fluid	Nonpolar oils, polar liquids, water and other	Oils	Oils, water
Density (g/cc)	3-5	1-2	1-2

Off viscosity (mPa-s)	100-1000	50-1000	2-500
Required field	~3 kOe	3kV/mm	~1 kOe
Field Induced changes	$\tau_y(B) \sim 100kPa$	$\tau_y(B) \sim 10kPa$	$\frac{\Delta\eta(B)}{\eta(0)} \sim 2$
Device excitation	Electromagnetics or permanent magnets	High voltage	Permanent magnet

2.3. History of Magnetorheological Effect

The magnetorheological (MR) effect was first observed in the 1940's by Jacob Rabinow, an engineer who was working at the US National Bureau of Standards. He observed that the presence of a magnetic field would cause some fluids to thicken and become solid. This discovery led to the subsequent patent of Rabinow's "Magnetic Fluid Force and Torque Transmitting Device" [1].

This discovery occurred almost concurrently with Willis Winslow's discovery of electrorheological (ER) fluids [10, 11]. These fluids were found to solidify when subjected to high electric fields. Although these two fluids where discovered at approximately the same time, most of the research and development of controllable fluids had been focused on the ER effect.

The primary feature of MR fluids is their controllability and reaction time. MR fluids react in milli-seconds to an applied magnetic field. To fully utilize these characteristics, one must have an electro-magnetic system that reacts at least as fast as the MR fluid itself. Because the technology did not offer practical control systems with sufficient reaction times to make use of the fast MR material reaction time, use of MR materials as a variable control device was not explored. This contributed to dormancy of MR material research, which lasted until the 1970's.

Research in MR fluids slowed down in 1980's, but did not go dormant. This slowdown may be attributed to the rise in interest in electrorheological materials, as they are the counterpart of MR fluids. The

research on electrorheological materials showed a high degree of sensitivity of the material to contaminants and in the 1990's the focus switched back to the more robust MR fluids.

The robustness of MR fluids has been investigated and confirmed through research and device development. Some of the investigations on the robust behavior of MR fluids focused on such issues as temperature [12], magnitude of applied magnetic field vs. power supply [13], history dependency (repeatability of cycling the applied magnetic field on and off) [2], stability of the carrier medium, and MR material reaction time. These investigations revealed that all of these issues favor the use of MR fluids in practical applications. A temperature range between -40° *C* and 150° *C* has minimal effect on the yield stress [14, 15]. A maximum value of yield stress is obtained when magnetic saturation of the fluid is reached. Any further increase in magnetic field has a diminishing effect on the rheological properties of the MR fluid. The yield limit is associated with the magnetic saturation limit and is caused by limits on the magnetic flux density ($\vec{B}$) in the MR fluid.

The response of MR fluid is reproducible when the magnetic field is cycled. There is negligible magnetic memory [14, 15] that would cause a magnetic history dependent effect. Rosensweig [16] refers to particle size as the reason for little or no magnetic memory. Ferrofluids (particle sizes on the order of nano-meters) do have magnetic memory. The magnetic memory can be viewed as the ability of the particles to retain their dipole alignment after the magnetic field has been removed. MR fluid particles are large enough so that gravity will cause the particles to move when no magnetic field is applied. This movement promotes rotation and displacement of the dipoles so they are no longer aligned. Thus, MR fluids have a negligible magnetic memory. Ferrofluid particles are of such a size that the Brownian forces are greater than gravity forces. These Brownian forces keep the particles in place after the magnetic field has been removed. Thus alignment of particles remains and magnetic memory exists in ferrofluids.

MR fluids are currently being utilized by industry and some commercial devices are made using the magnetorheological effect, like MR linear dampers and rotary clutches and brakes. Research on MR

material is being performed simultaneously by industry and academia. Industrial research focuses on optimization of the MR effect by developing new MR fluids and MR devices. Academic research is focusing on developing analytical descriptions of the fluid behavior and investigating novel applications of MR technology.

2.4. Composition of MR Fluids

The compositions of MR fluid typically consist of three components: the ferromagnetic particles, a carrying fluid, and a stabilizer.

2.4.1 Ferromagnetic Particles

Iron and iron-base alloys are the principal magnetosoft materials in use today [2, 17, 18]. The ferromagnetic particles are typically made from carbonyl iron and the magnetosoft particles are usually spherical in shape. A particle size range of 1-10 μm is preferred although broader range of 0.1-100 μm is also acceptable. In addition, the magnetosoft particles should have no residue magnetization and can randomly distribute in fluids. The traditional powders are iron particles, such as carbonyl particles and reduced iron particles. Iron alloy particles including iron-cobalt alloys and iron-nickel alloys are found to produce better performance than the traditional ones [17]. The drawback, however, is the high cost of such alloys in comparison to carbonyl iron particles.

2.4.2 Carrier Fluid

The carrier fluid serves as a continuous isolating medium. Preferably, the carrier liquid should have a viscosity value in the range of 0.01-1.0 Pa-s at 40^{o} *C* [2]. In general, an ideal carrying liquid should have a low initial viscosity, good anti-corrosion properties, low toxicity, and a wide range of temperature stability. Some commonly used carrier fluids are silicon oil, kerosene and synthetic oil [2]. Water and glycol also may be used as the carrier fluid [19].

2.4.3 Stabilizer

The stabilizer serves to provide added stability to the fluid composition. Stability of MR fluids is one of the most important properties of the fluid. It determines to a large extent the lifetime and durability of the fluid. There are three distinct types of stability associated with MR fluids: agglomerative stability, sedimental stability, and thermal stability [2]. Agglomerative stability refers to the resistance of the particles to the formation of aggregates. Sedimental or settling stability ensures that the particles do not settle down with time. Thermal stability refers to the ability of the fluid to function at reduced and elevated temperatures, and its resistance to degradation when held at elevated temperatures over long periods of time. Most of the work on stability of MR fluids in general, has been on the first two kinds of stability [2]. The stabilizer particles can be spherical in shape and preferably be non-porous. The stabilizer particles are approximately 0.005-0.015 μm in diameter. The relatively small diameter of the stabilizer particles results in a relatively large collective surface area. Stabilizer particles having a surface area of about 350 to 400 m^2/g are adopted. Typical stabilizers include inorganic silicon compounds. Table 2-2 shows the general properties of MR fluids.

Table 2-2 General properties of MR fluids, (Kordonsky, V. I., Z. P. Shulman, S. R. Gorodkin, S. A. Demchuk, I. V. Prokhorov, E. A. Zaltsgendler, and B. M. Khusid. *Physical Properties of Magnetizable Structure-Reversible Media*. in *5th International Conference on Magnetic Fluids, 18-22 Sept. 1989.* 1990. Riga, USSR. [20], used under fair use)

Specification	Remark
High Dynamic Yield Strength	Instantaneous generation of from 0 to 90 Pa yield stress for precise real-time control
Wide Operating Temperature Range	-40^{o} to 150^{o} with less than 10% variation in maximum dynamic stress
Millisecond Response Time	Ideal for use in applications requiring continuously variable, high-precision control

Non-Abrasiveness	Additives in MR fluids reduce abrasiveness, enabling devices to achieve required life expectancies
Low Sedimentation Level	The formulation of MR fluids solves sedimentation and stratification problems found in other MR fluids
Current Density	Can energize with permanent magnets
Specific Gravity	2-4
Ancillary Materials	Iron/Steel
Color	Black/Grey

The use of dispersant is not always sufficient for the provision of stability. The problem of stability over time had been dealt with in a number of ways.

It has been proposed [18] that the addition of a magneto-solid powder as a dispersant helps to form a spatial structure within the carrier fluid due to its intrinsic magnetic moment in the absence of an externally applied magnetic field. This spatial structure not only provides the necessary temporal stability, but in the presence of an external magnetic field these magnetic particles play an active role and help to enhance the MR effect. The formation of this structure which occurs over a period of time and is quickly destroyed under shear flow conditions is referred to as a thixotropic network.

Thixotropy is defined [21] as a decrease of the apparent viscosity under shear stress, followed by gradual recovery with time when the stress is removed. In order to prevent settling of the dispersed phase, it is desirable that a thixotropic network be formed [20, 22]. A thixotropic network is defined as a suspension of particles that at low or zero shear rates form a loose network or structure over a period of time. This structure is often referred to as “clusters” or “flocculates”. This loosely knit three-dimensional structure

reduces particle settling. When a shearing force is applied to the fluid, this structure is quickly dispersed and the degree of rigidity obtained from the thixotropic formation vanishes.

Thixotropic agents which have been found to be effective are low molecular weight hydrogen bonding agents or polymer modified metal oxides. Various colloids can be added in order to enhance this effect. Silica gel is also a preferred thixotropic agent and its porous time–induced structure not only provides stability against particle settling but its hygroscopic quality allows for the removal of any undesirable moisture.

A thixotropic agent which has been found to not only provide stability against settling but also to greatly enhance the MR effect is the addition of a porous carbon fiber powder [23]. The intertwined fibers have been found to mechanically suspend the particulate phase. When added to a carrier fluid, these fibers tend to gel the fluid and prevent settling. However when exposed to shear, this structure breaks down and displays excellent flow properties.

2.5. Magnetic Properties of MR Fluids

Besides the magnetic properties of isolated particles, the static magnetic properties of MR fluids characterized by *B-H* and *M-H* hysteresis are also important. Theoretical models for fluids and devices need magnetization as an input. These magnetic properties will also be helpful in predicting the dependence of the MR response on the applied current in the device.

At very low magnetic fields, the magnetic dipole moment induced in particles within MR fluid is given by [24],

$$m = 4\pi\mu_0 R^3 \beta H \tag{2-1}$$

$$\beta = \frac{\mu_p - \mu_f}{\mu_p + 2\mu_f} \tag{2-2}$$

where, R is the particle radius and μ_p and μ_f are the relative permeabilities of the particle and the fluid, respectively.

At higher fields where the magnetization of the particles reach saturation, the magnitude of the moment becomes independent of the field, as indicated by [25],

$$m = \frac{4}{3}\pi\mu_0 R^3 M_S \tag{2-3}$$

where, $\mu_0 M_S$ is the saturation magnetization.

The saturation magnetization of the fluid, $\mu_0 M_f$, is related to the saturation magnetization of the bulk magnetic solid, $\mu_0 M_S$, through the volume fraction ϕ of the solid present

$$\mu_0 M_f = \phi\mu_0 M_S \tag{2-4}$$

2.6. Commercial MR Fluids

The Lord Corporation is the leader in the field of commercial MR fluids and MR devices. Various types of MR fluids and devices are manufactured by Lord Corporation, such as dampers, rotary brakes, and various types of MR fluids.

Table 2-3 presents the main properties of different types of Lord MR fluids. Furthermore, the characteristics of MRFs such as plastic viscosity, magnetic induction, and yield stress of Lord MRF-132DG oil-based MR fluid are shown in Figures 2-2 – 2-4.

Table 2-3 Magnetorheological fluids (Lord Corp.), used under fair use

	MRF-122-2ED	MRF-132DG	MRF-140CG	MRF-241ES	MRF-336AG
BASE FLUID	Hydrocarbon	Hydrocarbon	Hydrocarbon	Water	Silicone
COMPATIBILITY					
Buna N (Nitrile)	Good	Good	Good	Good	Good
Butyl	Poor	Poor	Poor	Good	Good
EPDM/EPR	Poor	Poor	Poor	Good	Good
Fluoroelastomer	Good	Good	Good	Good	Good
Natural Rubber*	Poor	Poor	Poor	Poor	Good
Neoprene	Good	Good	Good	Good	Good
Silicone	Fair	Fair	Fair	Fair	Poor
Iron	Good	Good	Good	Good	Good
Stainless Steel	Good	Good	Good	Good	Good
Aluminum	Good	Good	Good	Fair	Good
Polyurethane	Good	Good	Good	Poor	Fair
OPEN/CLOSED SYSTEM	Open or Closed	Open or Closed	Open or Closed	Closed	Closed
OPERATING TEMPERATURE, °C(°F)	-40 to +130 (-40 to +266)	-40 to +130 (-40 to +266)	-40 to +130 (-40 to +266)	-10 to +70 (14 to 158)	-40 to +150 (-40 to +302)
VISCOSITY					
Pa @40°C (104°F) calculated as slope 500-800 sec^{-1}	0.07 ± 0.02	0.092 ± 0.015	0.225 ± 0.325	—	—
Pa at 10 sec^{-1}	—	—	—	10.8 ± 1.5	8.5
Pa at 50 sec^{-1}	—	—	—	2.2 ± 0.4	—

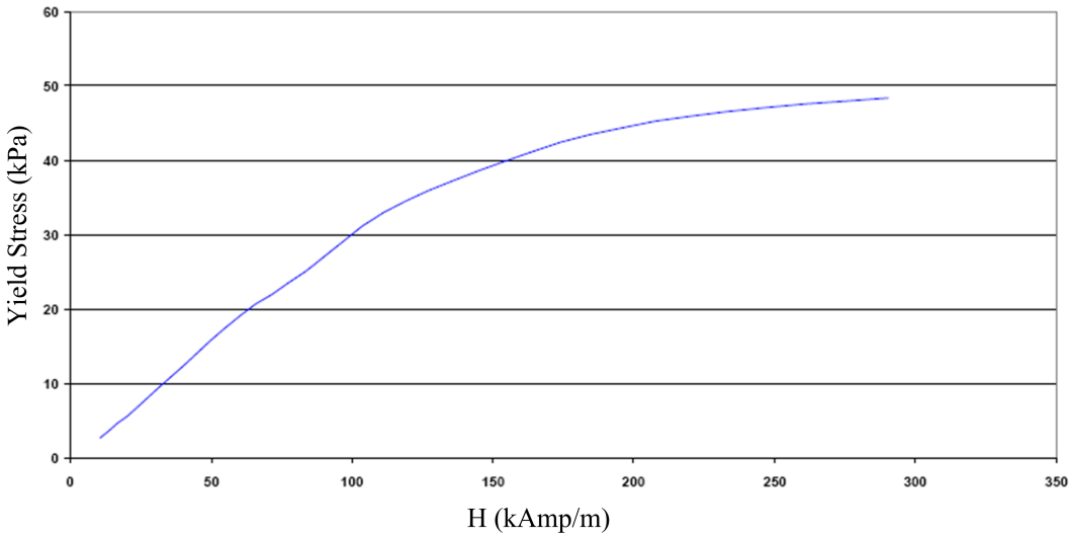

Figure 2-2 Yield stress vs. magnetic field strength (Lord Corp.), used under fair use

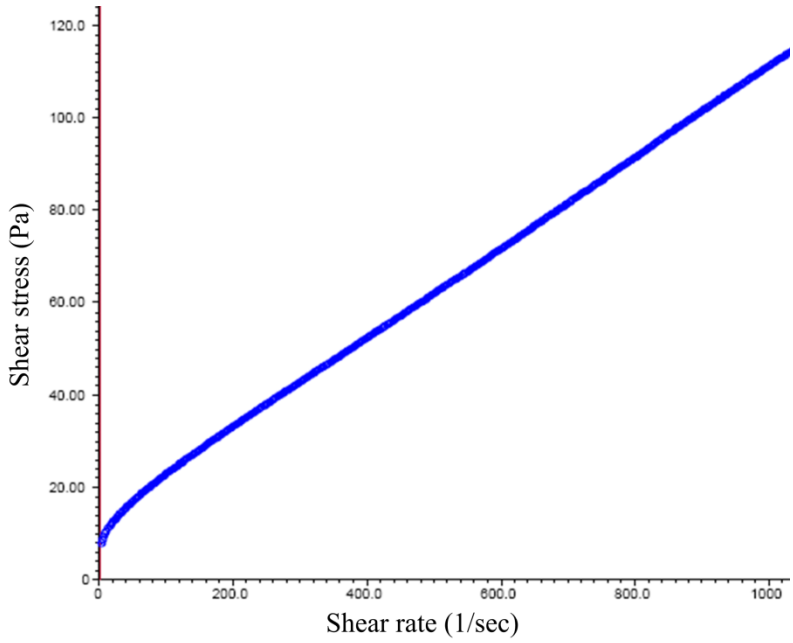

Figure 2-3 Shear stress as a function of shear rate with no magnetic field applied at 40°C (Lord Corp.), used under fair use

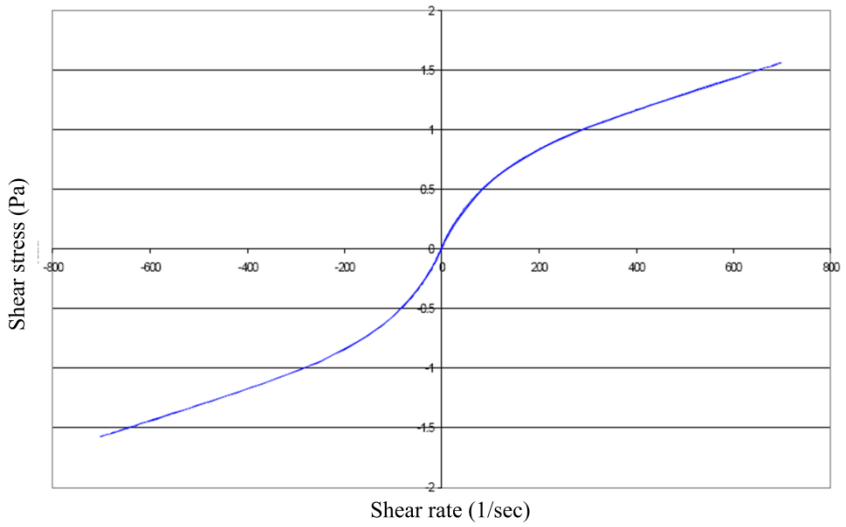

Figure 2-4 Typical magnetic properties of MRF-132DG (Lord Corp.), used under fair use

The biggest challenge in making a good MR fluid is to have high yield stress, temperature stability, redispersibility, and durability. The second biggest challenge of MR fluids is the material science studies such as surface chemistry and polymer physics. To our knowledge, there are not many systematic published studies on this aspect of MR fluid technology. Making durable MR fluids is also another challenge. There have been temperature studies in determining the yield stress of MR fluids at different

temperatures and these studies revealed the temperature stability of MR fluids. However, there are not many studies conducted by exposing the MR fluid to high shear rates and high shearing stresses. Although the "off-state" viscosity of MR fluids is a crucial parameter for having a high MR effect, it has not been investigated in detail. The rheology of MR fluids in their off state is also worth studying.

2.7. Rheology of MR Fluids

Generally, there are two distinct rheological domains over which MR fluids operate: the pre- and post-yield regions. The pre-yield region is characterized by a strong viscoelastic nature while the post-yield region shows a dominant viscous behavior. These two rheological domains are separated by a yield point, which varies as a function of the applied magnetic field [15, 26]. In describing the flow behavior of MR fluid, shear rate, temperature, volume fraction, and field strength are often considered as important variables.

In the pre-yield regime, the fluid acts like a solid and shows viscoelastic behavior. Mathematically, the pre-yield regime can be described by Figure 2-5.

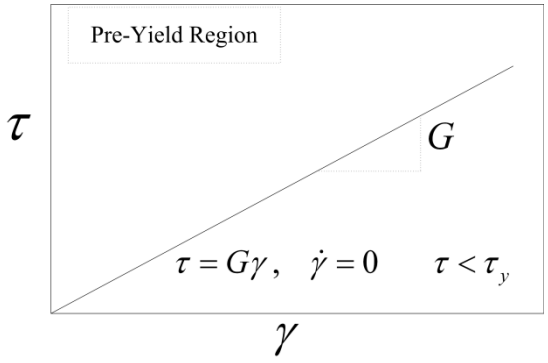

Figure 2-5 MRF behavior in pre-yield regime

For $\tau \geq \tau_y$, i.e. the post-yield regime, the material starts to flow. A typical shear stress versus shear rate behavior observed for a MR fluid in the presence of a magnetic field is illustrated in Figure 2-6 [27]. In order to describe the rheology of MR fluids, the classical Newtonian definition of the apparent viscosity

as the linear relationship between the shear stress and the shear rate can be used. But a Newtonian type of approximation is only valid for a MR fluid in the absence of a magnetic field.

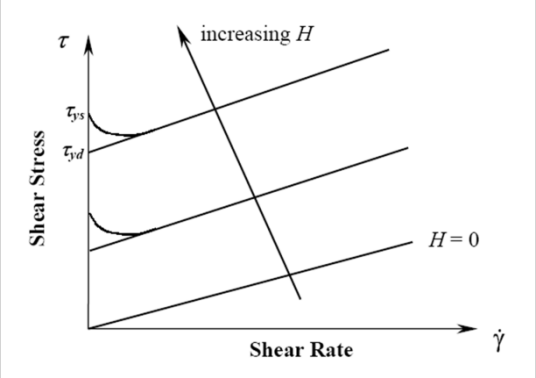

Figure 2-6 Post-yield shear behavior of MRF

In Figure 2-6, τ_{yd} and τ_{ys} are respectively the dynamic and static yield stresss of the MR fluid. The dynamic yield stress, τ_{yd}, reflects the flow characteristics of MR fluids whereas the static yield stress, τ_{ys}, is defined as the stress necessary to initiate flow within the MR fluid. The dynamic yield stress is governed by the behavior of the MR fluid in the post-yield flow region and is obtained by extrapolation of the shear stress versus shear rate measurements as shown in Figure 2-6. It is seen that the dynamic yield stress is smaller than the static yield stress which is true for both ER and MR fluids [28-31]. The static yield stress is due to the transient fracture of the particle chains which is highly dependent upon particle size, particle shape, and flow history of the material [31]. The post-yield behavior is used to model the properties of MR fluids for practical and industrial purposes.

The importance of the "off-state" viscosity of MR fluids comes from the figure of merit for MR fluids which is given by the "turn up" ratio defined as the ratio of "on-state" yield stress to the "off-state" viscosity. "On-state" refers to the state of the MR fluid under an applied magnetic field and the on-state yield stress behavior depends on the magnetic properties and the volume fraction of the magnetic phase [15]. The off-state viscosity, which is a function of carrier liquid, additives, surfactants [32], particle loading, and Particle Size Distribution (PSD) [33], is the value when no magnetic field is applied. Due to

the addition of additives and surfactants and changes in magnetic particle microstructure during shear, most MR fluids exhibit thixotropic behavior and shear thinning [34]. The breakup of weak agglomerates or bonds in the shear field is a major cause of a shear thinning behavior of MR fluids.

The structure of MR fluids is anisotropic (Figure 2-7) due to formation of particles along field lines of the applied magnetic field. Thus, yield stress will depend on the orientation of the magnetic field. The direction of three fundamental orientations between magnetic field and shear direction where only shear in the *x*-direction is considered, is given in Figure 2-7. Most MR fluid applications such as clutches, brakes, and dampers use the situation in which the magnetic field is applied along the gradient direction (Case B). For optical polishing application, Case C is preferred [35]. Shorey and co-workers compared the anisotropy of MR fluids by measuring the shear stress for case B and C where both cases have a field orthogonal to the direction of the flow. They found that in general, yield stresses were similar in magnitude but slightly higher for Case C [35]. They concluded that the slight difference in the yield stress may be due to the internal structure formation which would need to be investigated further.

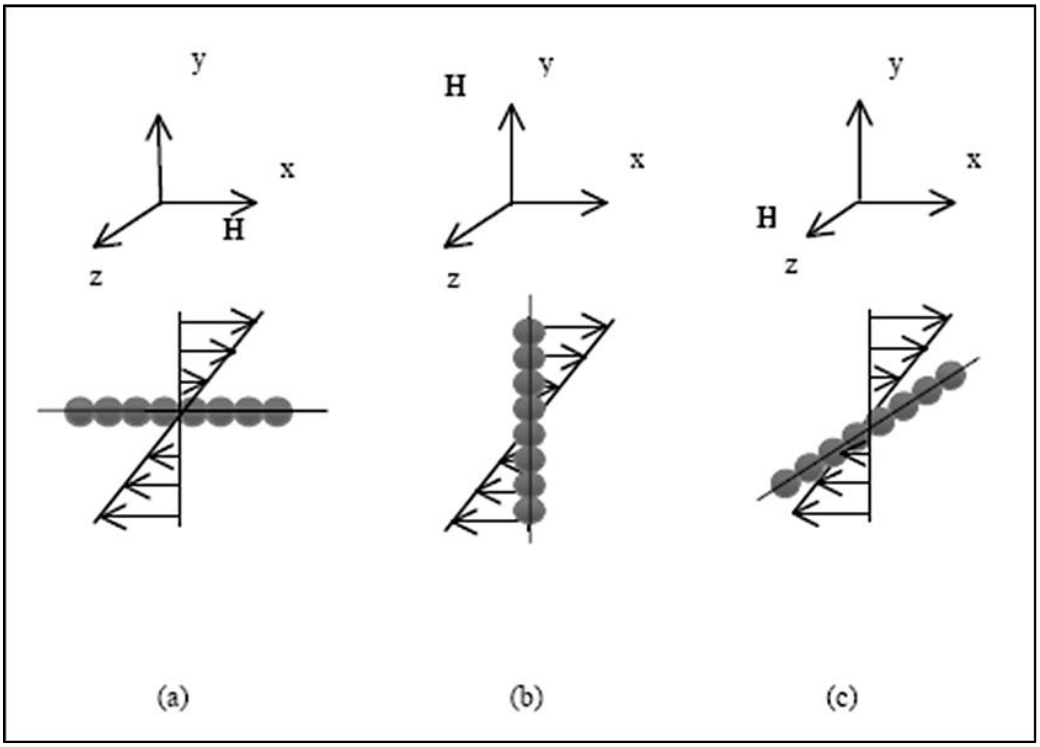

Figure 2-7 Anisotropy of MR fluids. The value of the yield stress depends on the direction of the applied magnetic field and the shear direction, (Genc, S., *Synthesis and Properties of Magnetorheological Fluids*, in *School of Engineering*. 2002, University of Pittsburgh. [3], used under fair use)

2.8. Rheometry

Design and realization of an actuator with MR fluid requires exact description of the rheological and magnetic properties of the MR fluid. Basic classes of rheometry are considered as stress driven and strain rate driven. Rotational rheometers such as concentric cylinder rheometer, cone and plate rheometer, parallel plate rheometer, and double Couette rheometer are basic types of rheometers used in rheological measurements [36-38] as shown in Figure 2-8. Shear stress, shear rate, strain and viscosity relations for different geometries are given in Table 2-4.

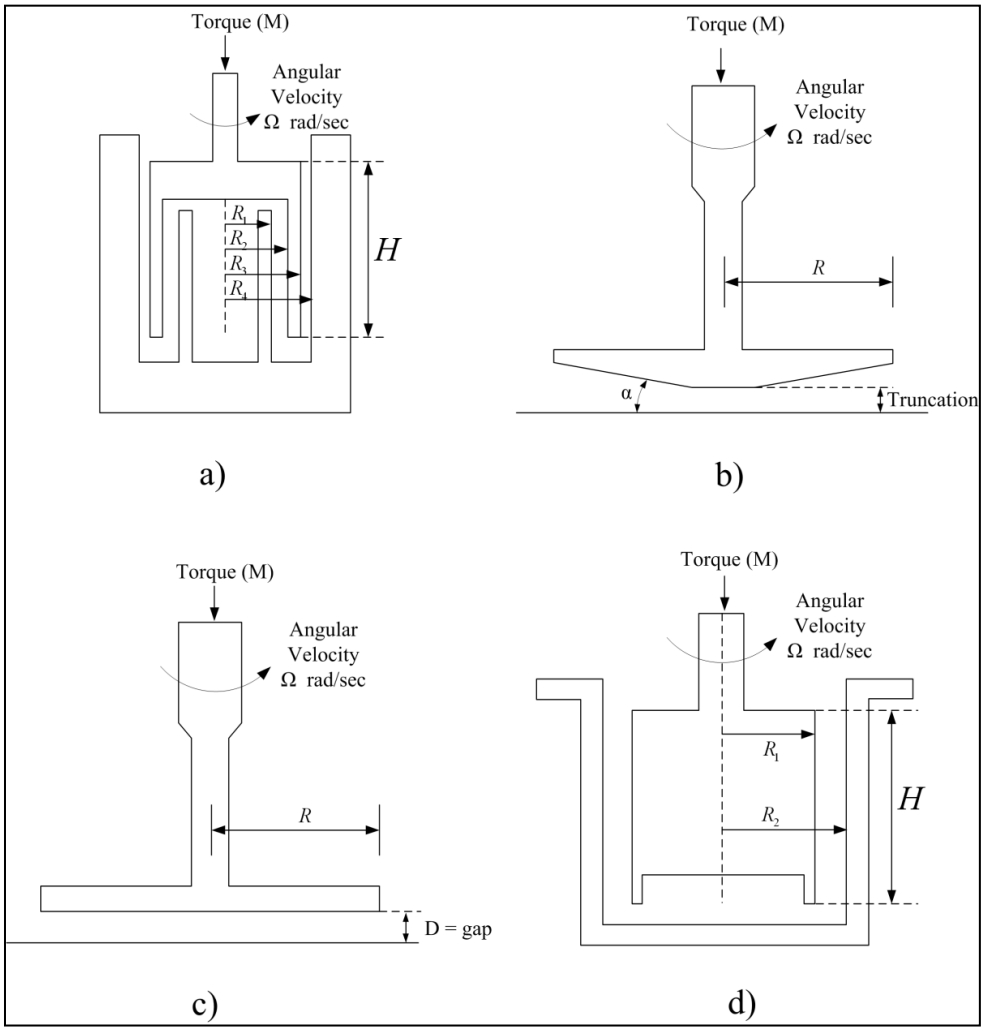

Figure 2-8 Types of rheometer geometries: a) double concentric cylinder, b) cone and plate, c) Parallel plate, and d) concentric cylinder, (Genc, S., *Synthesis and Properties of Magnetorheological Fluids*, in *School of Engineering*. 2002, University of Pittsburgh. [3], used under fair use)

Table 2-4 Equations of rheological properties for different geometries. In these equations, M is the torque, h is the height, R is the radius (Figure 2 8), Ω is the angular velocity, θ is the angular displacement, and α is the cone angle, (Laun, H. M., C. Kormann, and N. Willenbacher, *Rheometry on Magnetorheological (Mr) Fluids. I. Steady Shear Flow in Stationary Magnetic Fields.* Rheologica Acta, 1996. 35(5): p. 417. [37], Lemaire, E. and G. Bossis, *Yield Stress and Wall Effects in Magnetic Colloidal Suspensions.* Journal of Physics D: Applied Physics, 1991. 24(8): p. 1473. [39], de Gans, B. J., C. Blom, J. Mellema, and A. P. Philipse. *The Development of a Magnetorheometer and Its Use in Investigating Ferrofluids*. 1998. Yonezawa, Japan: World Scientific. [40], used under fair use)

	Shear stress	Shear rate	Strain	Viscosity
Concentric Cylinder	$\frac{M}{\left(2\pi R^2{}_{ave} h\right)}$	$\frac{2\Omega R_1{}^2 R_2{}^2}{r^2\left(R_2{}^2 - R_1{}^2\right)}$	$\frac{\theta R_{ave}}{R_2 - R_1}$	$\frac{M\left(R_2{}^2 - R_1{}^2\right)}{4\pi h\Omega R_1{}^2 R_2{}^2}$

Parallel Plate	$\frac{M}{2\pi R^3}$	$\frac{\Omega R}{h}$	$\frac{R\theta}{h}$	$\frac{\pi\Omega M R^4}{2h}$
Cone and Plate	$\frac{3M}{2\pi R^3}$	$\frac{\Omega}{\alpha}$	$\frac{\theta}{\alpha}$	$\frac{3\alpha M}{2\pi R^3\Omega}$
Double Concentric	$\frac{M}{2\pi h\left(R_1^2+R_4^2\right)}$	$\frac{\Omega R_4^2}{\left(R_4^2-R_3^2\right)}+\frac{\Omega R_1^2}{\left(R_2^2-R_1^2\right)}$	$\frac{\theta R_4^2}{\left(R_4^2-R_3^2\right)}+\frac{\theta R_1^2}{\left(R_2^2-R_1^2\right)}$	$\frac{M\left(R_4^2-R_3^2\right)\left(R_2^2-R_1^2\right)}{2\pi h}$

The concentric cylinder rheometers are best for lower viscosity systems and high shear rates. The gravitational settling of suspensions has less effect than cone and plate and parallel plate. Shear rate across the gap is not uniform whereas in cone and plate rheometers, the shear rate is constant. Cone and plate rheometers are good for high and low viscosity systems. This geometry, however, is not good at high shear rates. Rotational parallel plate rheometer is very useful for obtaining viscosity and normal stresses at high shear rates. It allows changing gap heights very easily, which can be very useful in terms of determining the wall slip at two different gaps [41] .

A rotational parallel-plate geometry inserted into a coil was used by Lemaire and Bossis [42] whereas Laun et al.[37] presented measurements performed with a concentric rheometer. A cone and plate type rheometer was utilized by Gans and co-workers in order to measure rheological behavior of the inverse ferrofluids. By choosing a small cone angle, they minimized the sample variations in the radial direction [43]. Shorey and co-workers developed a magnetometry for understanding properties of MR fluids for optical polishing [35].

2.9. Micro-Structure

As stated, the effect of a magnetic field on MR fluids causes the particles to form chain-like fibrils and creates a micro-structure within the MR fluid. These chain-like fibrils align themselves parallel to the

applied magnetic field. At small length scales, the fibril phenomenon is due to magnetic dipole interaction. The fibril phenomenon has been modeled considering the magnetic dipole interaction of adjacent particles along a single fibril at a constant magnetic saturation level [15]. The adjacent particle interaction effects were averaged over the entire fluid in order to model the effect at longer length scales. The fibril chain is assumed to consist of perfect spherical magnetic particles aligned in single columns (Figure 2-9) along with the following assumptions and simplifications.

- Magnetic field, $\vec{H}$, is constant in magnitude and orientation throughout the fluid

- Magnetic saturation of the fluid particles is uniform

- Linear magneto-static principles are appropriate

- Particles are perfect spheres of radius *r*

- Particles are aligned with perfect symmetry

- Only adjacent particles interaction (top & bottom) are calculated

- No multi-dipole interaction or adjacent fibril interaction (side to side) is considered

- Particle fibril volume remains constant as the fibril is sheared

- For a single particle, the fibril volume is $2h\pi w^2$, where *h* is the half height of a particle fibril volume and *w* is the radius of the particles fibril volume

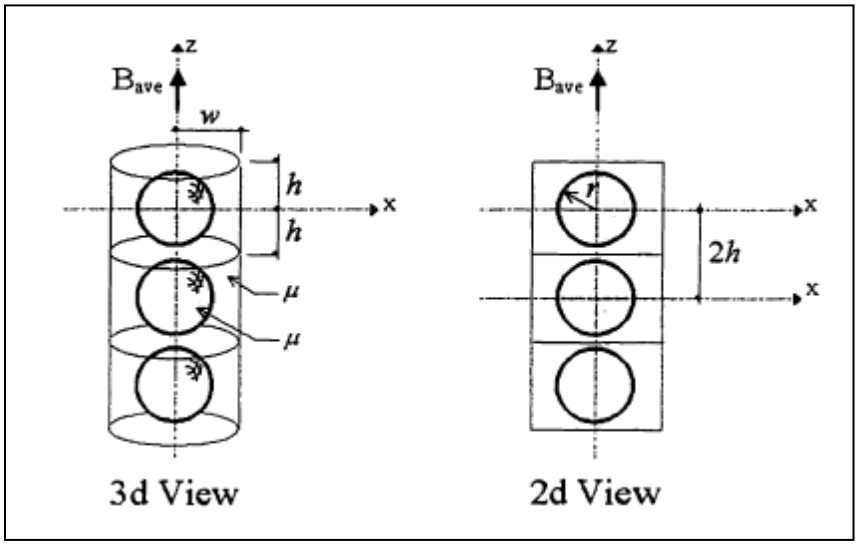

Figure 2-9 Magnetic dipole fibril model, (Baltimore, C. V., *Field-Flow Orientation Effects in Magnetorheological Fluids*, in *Civil and Environmental Engineering Dept.* 1998, Duke University. [44], used under fair use)

Tang et al. (1995) determined the force along z-axis between two adjacent particles to be

$$F = 24\pi\mu_0\mu_f r^2(\beta H)^2\left(\frac{r}{2h}\right)^4 \tag{2-5}$$

The role of induced magnetic flux ($\vec{B}$),applied magnetic field ($\vec{H}$), and magnetization ($\vec{M}$) are some of the most important variables in the response of MR fluids due to their non-linear relationship.

2.10.MR Flow Modes

All devices that use MR fluids can be classified as operating in a: (a) valve mode (fixed poles), (b) direct shear mode (relatively movable poles), (c) squeeze mode, or (d) combination of these modes [45]. Diagrams of these basic modes of operation are shown in Figure 2-10. Examples of valve mode devices include servo-valves, dampers, shock absorbers, and actuators. Shear mode devices include clutches, brakes, chucking and locking devices, dampers, and structural composites.

Pressure drop developed in a device based on pressure driven flow mode can be divided into a field independent viscous component, Δp_η , and an applied field dependent induced yield stress component, Δp_τ . These pressure drops may be approximated by [46]:

$$\Delta p_\eta = \frac{12\eta QL}{g^3 w} \tag{2-6}$$

$$\Delta p_\tau(H) = \frac{c\tau_y L}{g} \tag{2-7}$$

where, Q is volumetric flow rate, η is viscosity with no applied field, and τ_y is the yield stress developed in response to an applied magnetic field. Parameter c has a value that ranges from a minimum value of 2 (for $\Delta p_\tau / \Delta p_\eta$ less than 1) to a maximum value of 3 (for $\Delta p_\tau / \Delta p_\eta$ greater than 100). The total pressure drop in a pressure driven flow mode device is approximately equal to the sum of Δp_τ and Δp_η .

In a like manner, the force developed by a direct-shear device can be divided into viscous, F_η , and field induced yield stress, F_τ , components,

$$F_\eta = \frac{\eta SA}{g} \tag{2-8}$$

$$F_\tau(H) = \tau_y(H)A \tag{2-9}$$

where, the total force developed by a direct-shear device is the sum of F_η and F_τ .

While less well-understood than the other modes, the squeeze mode has been used in some small-amplitude vibration dampers. In this research, MR fluids in squeeze mode are extensively studied. Both rheological behavior and applications of MR fluids in squeeze mode are investigated.

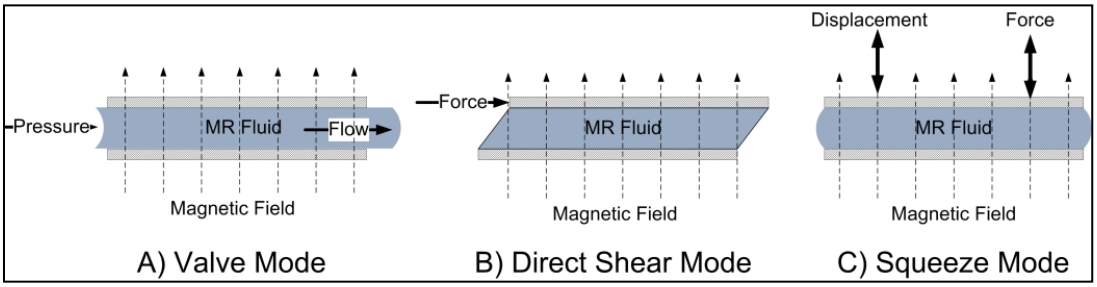

Figure 2-10 Basic operating modes for controllable fluid devices

To date, several MR fluid devices have been developed for commercial use by Lord Corporation [5, 47]. Linear MR fluid dampers have been designed for use as secondary suspension elements in vehicles. MR fluid rotary brakes are smooth acting, proportional brakes which are more compact and require substantially less power than competing systems. MR fluid vibration dampers for real-time, active-control of damping have been used in numerous industrial applications.

In civil engineering applications, expected damping forces and displacements are rather large in magnitude. Therefore, MR dampers primarily operate under direct shear mode might be impractical. Usually valve mode or its combination with direct shear mode is employed. These dampers are capable of meeting real-world requirements and are presently either in commercial production or in pre-production trials.

2.11. MR Fluid Constitutive Models

2.11.1 Yield Stress

Yield stress of MR fluids mainly depends on saturation magnetization and volume fraction of magnetic particles. In the analytical models developed by Ginder and co-workers, yield stress increases linearly with increasing volume fraction [15, 28]. At high volume fractions, however, exponential increase of yield stress with increasing volume fraction was reported by Volkova and Chin [48]. This can be attributed to the higher packing of particles where the affine deformation can be restricted leading to higher stresses [48].

2.11.2 Viscosity

Viscosity of a fluid can be defined as the constant of proportionality between shear stress of a fluid and its velocity gradient or shear rate:

$$\tau = \eta\dot{\gamma} \tag{2-10}$$

For Newtonian fluids, this relationship is linear and η is constant. For some fluids, including MR and ER fluids, this relationship is non-linear and other means of relating the shear stress to the shear rate is necessary.

Figure 2-11 shows several types of rheological behavior. Curve *E* represents a Newtonian fluid with constant viscosity. Curve *D* represents a shear-thinning (pseudoplastic) fluid while curve *F* represents a shear-thickening (dilatant) fluid. Curves *A*, *B*, and *C* represent fluids possessing a yield stress.

Often, flow behavior of a fluid is described by its "apparent viscosity". This term represents the slope of the line connecting the origin to a specific point on the shear stress vs. shear rate plot. The "differential viscosity" represents the derivative of shear stress with respect to shear rate. For fluids which possess a yield stress, apparent viscosity is sometimes used to describe their rheological behavior; however, this one parameter model does not always provide sufficient insight into a fluid's true behavior.

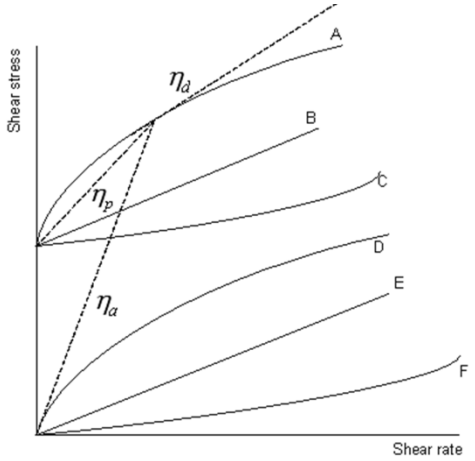

Figure 2-11 Viscous types of rheological behavior

2.11.3 Bingham Model

A simple Bingham visco-plasticity model [49], as shown in Figure 2-12, is effective at describing the essential field-dependent fluid characteristics. In this model, total shear stress τ is given by

$$\tau = \tau_y(H)\,\mathrm{sgn}(\dot{\gamma}) + \eta\dot{\gamma} \tag{2-11}$$

where, τ_y is yield stress caused by the applied field, H is the magnitude of applied magnetic field, $\dot{\gamma}$ is shear strain rate, and η is field-independent plastic viscosity, defined as the slope of measured post-yield shear stress versus shear rate.

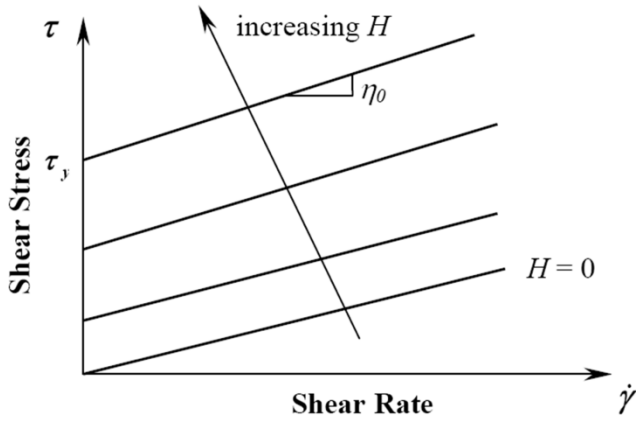

Figure 2-12 Idealized Bingham plastic behavior of MR fluids

2.11.4 Herschel-Bulkley Model

Note that the fluid post-yield viscosity is assumed to be a constant in Bingham model. But in many situations, it is observed that the viscosity decreases as the strain (shear) rate increases. This phenomenon is called "shear-thinning" which is shown in Figure 2-13 and Figure 2-14. Because MR fluids exhibit shear-thinning effect, Herschel-Bulkley visco-plasticity model [50] can be employed to accommodate this effect. In this model, the constant post-yield plastic viscosity in Bingham model is replaced with a power law model dependent on shear strain rate. Therefore,

$$\tau = \left[\tau_y(H) + K\left|\dot{\gamma}\right|^n\right]\mathrm{sgn}(\dot{\gamma}) \tag{2-12}$$

where, n and K are positive fluid parameters. K is called consistency index and n is the power law index of the Herschel-Bulkley model. It can be seen that the equivalent plastic viscosity of the Herschel-Bulkley model is

$$\eta_e = K\left|\dot{\gamma}\right|^{(n-1)} \tag{2-13}$$

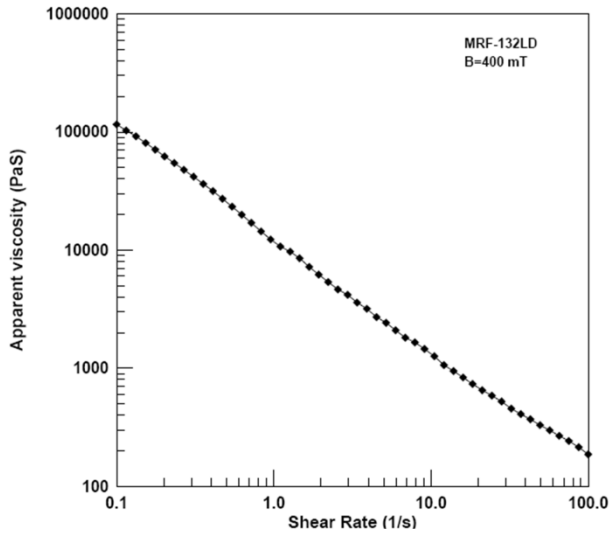

Figure 2-13 Decrease of apparent viscosity with shear rate, (Weihua, L., *Rheology of Mr Fluids and Mr Damper Dynamic Response: Experimental and Modeling Approaches*. 2001, Nanyang Technological University: Singapore. [51], used under fair use)

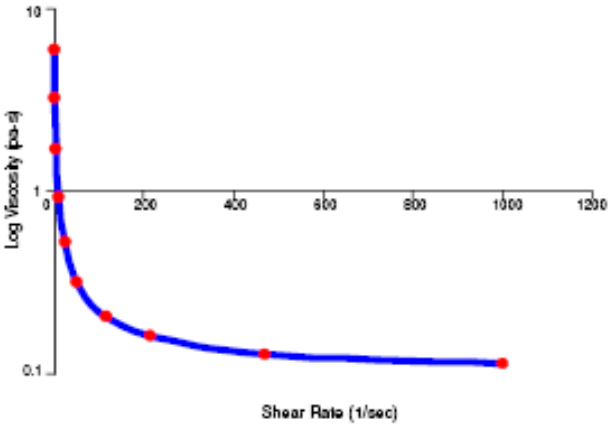

Figure 2-14 Shear thinning in MRF, (Guangqiang, Y., *Large-Scale Magnetorheological Fluid Damper for Vibration Mitigation: Modeling, Testing and Control*. 2001, University of Notre Dame. [19], used under fair use)

Equation (2-13) indicates that equivalent plastic viscosity, η_e , decreases as shear strain rate, $\dot{\gamma}$, increases when *n<1* (shear-thinning). Furthermore, this model can also be used to describe fluid shear-thickening effect when *n>1*. The Herschel-Bulkley model reduces to the Bingham model when *n=1*, therefore $\eta = K$.

Figure 2-15 shows a typical comparison between Bingham model and Herschel-Bulkley model. As shown, Herschel-Bulkley model is capable of predicting the shear-thinning behavior of MR fluids while Bingham model is not. This advantage raises the interest of using Herschel-Bulkley model in analyzing flow of MR fluids at high shear rates where the effects of shear-thinning is of great concern.

In Figure 2-15, data are based on experiments done by Weihua, L. [51] with parameters of Table 2-5.

Table 2-5 Parameters for the Bingham and the Herschel-Bulkey model, (Weihua, L., *Rheology of Mr Fluids and Mr Damper Dynamic Response: Experimental and Modeling Approaches*. 2001, Nanyang Technological University: Singapore. [51], used under fair use)

Model	Model Equation	Identified Parameters
Bingham model	$\tau = \tau_{yd} + \eta\dot{\gamma}$	$\tau_{yd} = 12910\ Pa$ $\eta = 44\ Pa - s$
Herschel-Bulkley model	$\tau = \tau_{yd} + K\dot{\gamma}^n$	$\tau_{yd} = 10200\ Pa$ $K = 1058\ Ps - s^n$ $n = 0.41$

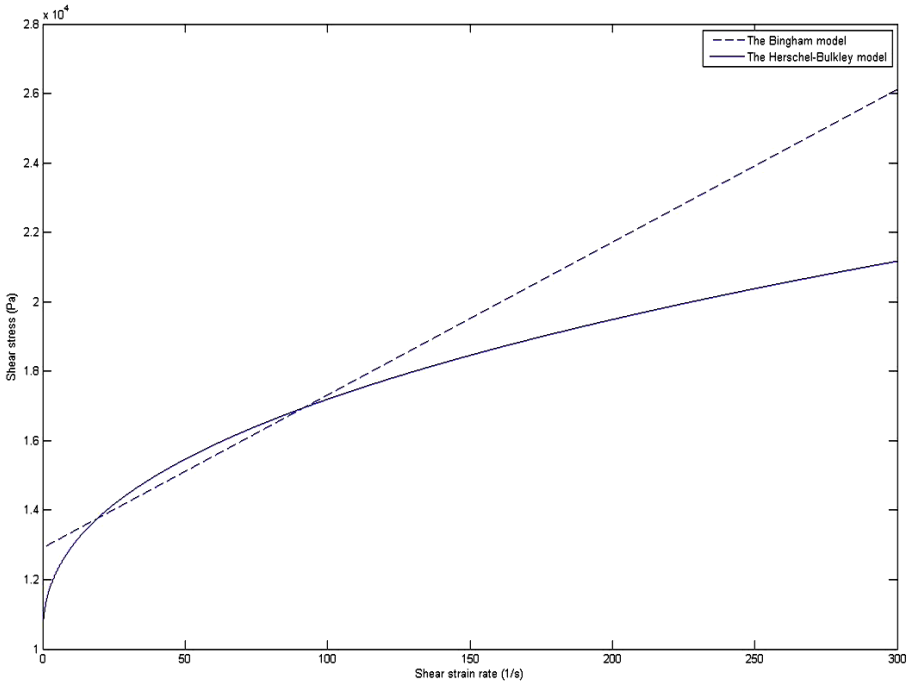

Figure 2-15 Comparison between the Bingham model and the Herschel-Bulkley model

2.12. Applications of MR Fluids

As stated earlier, the main advantages of MR effect are reversibility of properties, fast response, low power requirements, and wide temperature stability [5, 52]. Hence, MR fluids have tremendous potential in providing simple and effective interfaces between electronic controls and mechanical systems. MR fluids are expected to play a significant role in modern industrial areas including power transmission systems, aerospace and aeronautic servo systems, active control of structure vibration, MEMS, adaptive structure, optical grinding, and robot systems [45, 53-57]. Most of MR applications to date have been on the development of dampers, clutches, valves, engine mounts, and brakes [5, 34, 47, 52, 53, 55, 58, 59].

Optical polishing, which was first initiated by Kordonski and co-workers, is another promising application of MR fluids [57, 60]. MR fluid contains a nonmagnetic polishing abrasive. Under high shear, the flow of non-magnetic abrasive particles causes material removal. The most commonly used polishing abrasive/carrier liquid combination for optical polishing for all optical glasses and crystals is cerium oxide/water combination. Abrasives like alumina and diamond are used for materials other than glasses.

Lord Corporation has manufactured MR fluid devices for commercial applications including heavy-duty vehicle seat suspensions, rotary brakes that provide tunable resistance for exercise equipment and vibration dampers for various industrial applications such as dampers for washing machines (www.lord.com). An MR damper for commercial applications such as seat suspension is shown in Figure 2-16.

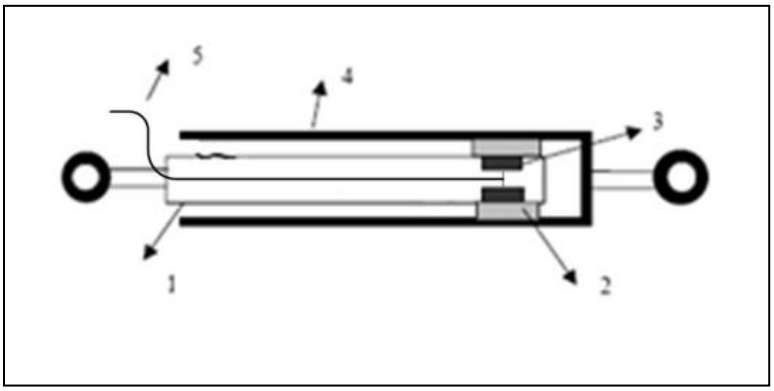

Figure 2-16 MR fluid damper, 1) Plastic shaft, 2) Sponge saturated with MR fluid, 3) Coil, 4) Steel tube, 5) Wire supplying current (Lord Corp.), used under fair use

Figure 2-17 shows a schematic of a damper used for seismic applications. A rotary brake is also shown in Figure 2-18. Figure 2-19 shows a three-dimensional view of an MR clutch.

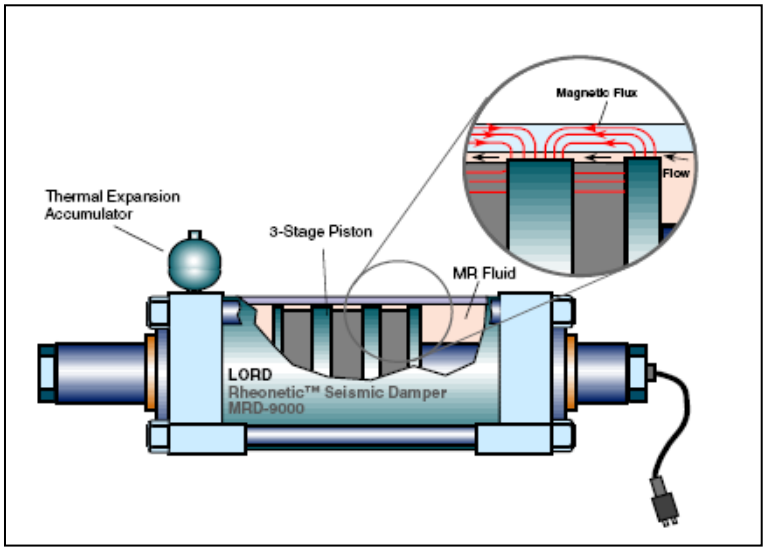

Figure 2-17 Schematic of a prototype damper for seismic applications [47], used under fair use

Figure 2-18 MR rotary brake (Lord Corp.), used under fair use

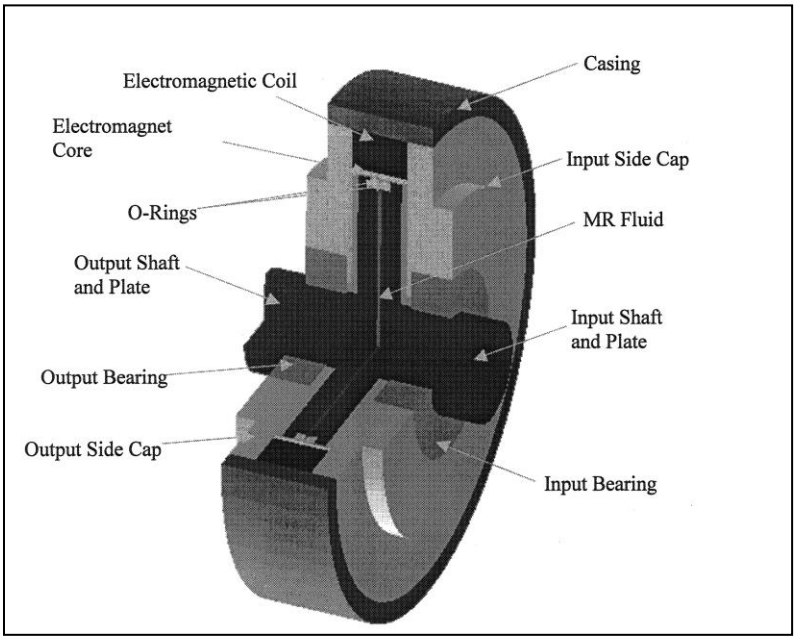

Figure 2-19 A three dimensional view of a prototype MR brake, (Kavlicoglu, B. M., F. Gordaninejad, C. A. Evrensel, N. Cobanoglu, Y. Liu, A. Fuchs, and G. Korol. *A High-Torque Magneto-Rheological Fluid Clutch*. in *Smart Structures and Materials 2002: Damping and Isolation, 18-20 March 2002, Proceedings of the SPIE - The International Society for Optical Engineering*. 2002. San Diego, CA, USA: SPIE-Int. Soc. Opt. Eng. [61], used under fair use)

It has been found that application of a magnetic field produces a significant change in heat transfer properties [20]. When direction of both heat and magnetic flux are aligned, heat transfer properties are significantly enhanced while a mutually perpendicular orientation of magnetic field and direction of heat flux significantly decreases heat transfer.

2.13. Stability of MR Fluids

Stability and redispersibility of MR fluids have been one of the most important issues of these materials. Stable MR fluids are considered to exhibit no or very little amount of particle settling. For dilute systems, dependence of sedimentation velocity of a spherical particle can be obtained from Stoke's law as follows [62]:

$$v = \frac{2}{9}\frac{R_s^{\,2}(\Delta\rho)g}{\eta} \tag{2-14}$$

R_s is particle radius, $\Delta\rho$ is difference in density of magnetic phase and carrier liquid, η is viscosity of carrier liquid and g is the gravitational acceleration ($9.8 m/s^2$). Since, less viscous liquids will aggravate the settling of the particles in an MR fluid, Rankin and co-workers formulated a suspension with viscoplastic continuous phase (e.g., grease) to prevent sedimentation [63]. When the yield stress of the viscoplastic medium is bigger than the critical yield stress that was defined for each particulate material and particle radius, the particles are suspended. Although, for most of the applications the figure of merit for MR fluids is to keep the off-state viscosity as small as possible, for applications such as control of seismic vibrations, paste-like MR fluids can be more appropriate since gravitational settling over an extended period of time can be prevented.

If settled magnetic particles in MR fluid can be easily dispersed by applying very little mechanical energy, i.e., stirring or shaking, then the fluid is said to be *redispersible*. Some fluids exhibit poor redispersibility and therefore "cake-like" formation is observed at the bottom of the fluid which would degrade MR response.

Another potential problem associated with MR fluids is In-Use-Thickening (IUT). In an early version of Lord Corporation's MotionMaster RD-1005 truck seat damper, the off-state force (which would ideally remain constant at the initial low value over the life of the product) increased by a factor of 2.5 after

600000 on-state cycles, rendering it ineffective [64]. This increase in the off-state force is due to an increase in the off-state viscosity of the fluid, a phenomenon known as "In-Use Thickening" (IUT) and believed to be caused by spalling of the surface layer of the particles in the MR compound. After some years of fluid development, this problem was overcome and presently MR fluids can endure long on-state periods without deterioration.

Exact MR fluid life depends on a variety of factors, including shear rate, temperature, and duration. An estimate of the usable life of an MR fluid can be obtained from the following relation [64],

$$LDE = \frac{1}{V}\int_0^{life} P dt \tag{2-15}$$

where, V is the fluid volume (since LDE is generally given in J/cm^3, this should be expressed in cm^3), and P is dissipated power. Hence, LDE is the energy dissipated per unit volume of the fluid throughout its operation and can be compared to the permissible value obtained from experimental data.

2.14. Wall Slip Effects

Slip occurs in flow of two or multi-phase suspensions because of the displacement of dispersed phases away from the walls of geometries used in rheometer leaving a layer of particles next to the wall more dilute than in bulk dispersion. For suspensions, gravity can enhance slip effect especially in parallel plate or cone and plate geometries. During flow, shear induced particle migration can occur due to shear rate gradient [65] and this results in a low viscosity layer adjacent to the wall that leads to an “apparent wall slip”. Conditions that lead to large and significant slip effects can be summarized as: large particles as dispersed phase, smooth wall, small flow dimensions, low speeds and flow rates, walls and particles carrying electrostatic charges which would cause the walls of the platens to repel adjacent particles [66]. Slipping at walls can be prevented by roughening the surface of walls as to increase the friction between

the walls and the suspension. In rough surfaces, particles become trapped whereas in smooth surfaces, they can roll over small bumps more easily.

It is proposed by some researchers [67] that yielding of MR fluids is not generally due to rupture of the aggregates but rather to their slip on the plates of rheometers. This slipping occurs when the applied stress overcomes the solid friction of the particles on the plates. This yield stress is commonly known as frictional yield stress and is obtained by extrapolating the stress toward zero shear rates in a log-log scale of the plot of shear stress vs. shear rate.

2.15. Dynamic Investigation of MR Fluids

Many MR devices operate in dynamic modes where fluid is subjected to oscillatory shear [47]. Though development of MR fluids and devices requires knowledge of their response under such condition, MR rheology under such conditions has not been studied extensively and is not well understood [68].

Linear viscoelastic properties of ER and MR suspensions for sufficiently small strain amplitudes have been investigated extensively by many researchers [68-72]. However, the non-linear viscoelastic properties of ER and MR fluids under large amplitude oscillatory shear still needs further investigation. Although there is a unifying theory that describes linear viscoelasticity, there is no unifying constitutive theory for the non-linear case.

MR fluids, as well as other viscoelastic materials, such as polymers, emulsions, or ER fluids, exhibit linear viscoelastic properties at sufficiently small strain range ($\gamma \le \gamma_{lin}$). Here, γ_{lin} is a critical strain, which is used to separate linear viscoelastic region from non-linear viscoelastic region.

Assuming MR fluid is subjected to an oscillatory strain with sufficiently small amplitude defined by:

$$\gamma = \gamma_0 e^{i\omega t} \tag{2-16}$$

where, $\gamma_0 \ll 1$ is strain amplitude and ω is angular frequency in *rad/sec*. Shear rate is the time derivation of γ, which is in the form

$$\dot{\gamma} = \gamma_0 \omega e^{i(\omega t + \pi/2)} \tag{2-17}$$

In this case, shear stress response (τ) to oscillatory shear would be

$$\tau = \tau_0 e^{i(\omega t + \delta)} \tag{2-18}$$

where, δ is called mechanical loss angle, whose range is $0^\circ < \delta < 90^\circ$, as shown in Figure 2-21. For fluids, the constitutive equation is $\tau = \eta \frac{\partial v}{\partial y} = \eta \dot{\gamma}$, where, η is the viscosity and $\partial v / \partial y$ is velocity gradient. If the fluid is subjected to an oscillatory strain, $\gamma = \gamma_0 e^{i\omega t}$, then the response to this oscillatory strain is a shear stress with 90^o difference in phase $\tau = \eta \gamma_0 e^{i(\omega t + \frac{\pi}{2})}$. If this oscillatory excitation is applied to an elastic solid with the constitutive equation of $\tau = G\gamma$, the response would be $\tau = G\gamma_0 e^{i\omega t}$ which is in phase with the strain. So, the mechanical loss angle of $\delta = 0^\circ$ corresponds to perfect elastic materials and $\delta = 90^\circ$ for purely viscous materials. The relationship between stress and strain under sinusoidal deformation for elastic, viscous and viscoelastic material is shown in Figure 2-20.

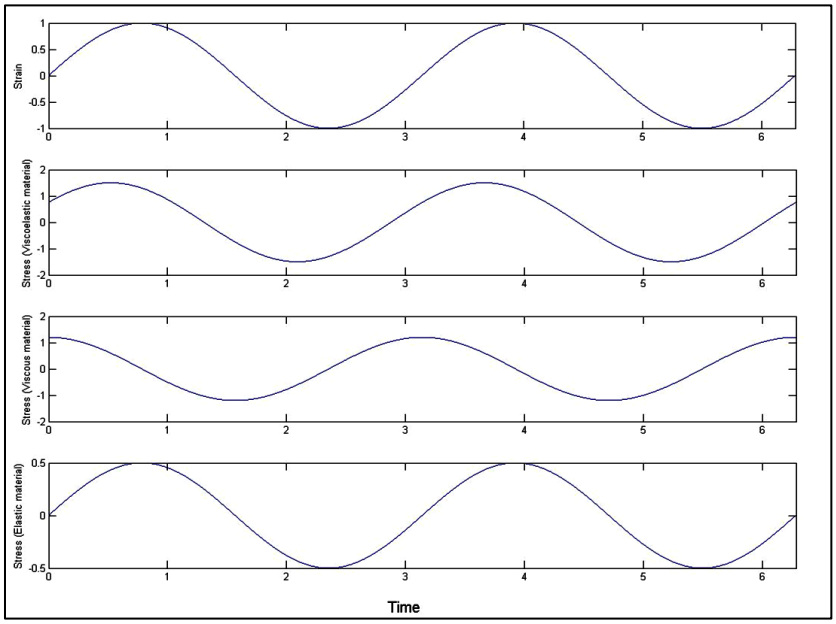

Figure 2-20 Stress and strain relationship in oscillatory measurement

For a linear viscoelastic material, the response stress, τ , can be divided into two components, an elastic stress (τ') in phase with strain, and a viscous stress (τ'') in phase with strain rate (Figure 2-21).

$$\tau^* = \tau' + i\tau'' \tag{2-19}$$

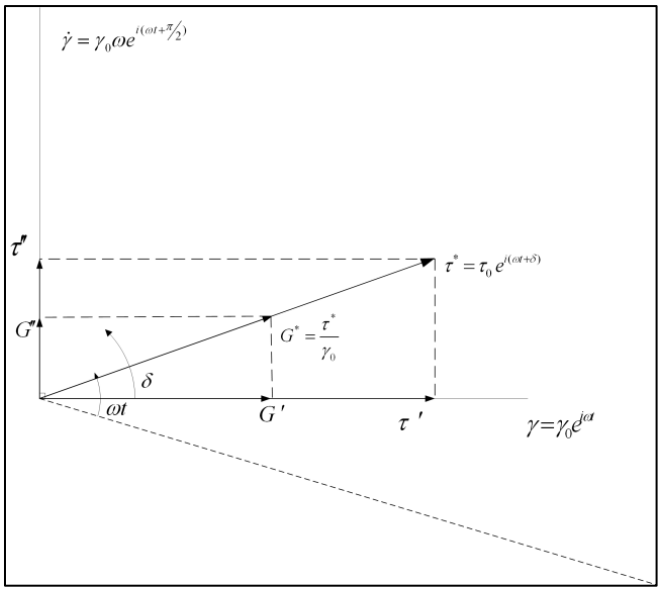

Figure 2-21 Strain and stress in dynamic measurement and their corresponding components

Dynamic shear modulus of a viscoelastic material is defined by dividing the complex shear stress by the applied strain amplitude

$$G^* = \frac{\tau^*}{\gamma_0} = \frac{\tau'}{\gamma_0} + i\frac{\tau''}{\gamma_0} = G' + iG'' \tag{2-20}$$

where, G' is the storage modulus and G'' is the loss modulus presented by

$$G' = \left|G^*\right|\cos(\delta) \tag{2-21}$$

$$G'' = \left|G^*\right|\sin(\delta) \tag{2-22}$$

The storage modulus, G', represents the ability of the viscoelastic material to store the energy of deformation, which contributes to the material or device stiffness. The loss modulus, G'', represents the ability of the material to dissipate the energy of deformation as heat. The loss angle, δ, is an indication of whether the material is more solid-, or liquid-like.

Another equation widely used for viscoelastic materials is

$$\tan\delta = \frac{G''}{G'} \tag{2-23}$$

$\tan\delta$ is called loss factor, which is the ratio of the two complex components. The two quantities, G'' and $\tan\delta$, are of interest for vibration damping. If they are small for a given vibration frequency and magnetic field, then damping will be small. For large values of G'' and $\tan\delta$, damping will be augmented.

2.16.Measures of Controllability

Controllability of MR fluids is most often assessed using either shear ratio or excess shear concepts.

2.16.1 Shear Ratio

Shear ratio of an MR fluid can be described as the ratio of the shear stress under an applied magnetic field divided by the shear stress of the fluid in the absence of a magnetic field.

$$\text{shear ratio} = \frac{\tau_H}{\tau_0} \tag{2-24}$$

This parameter allows one to judge the controllability of the fluid at a certain shear rate. The shear ratio varies with shear rate and is typically found to decrease with increasing shear rate which is due to the shear-thinning of the MR fluid by increasing the shear rate.

2.16.2 Excess Shear

Excess shear in an MR fluid can be described as the difference between the observed shear stress under an applied magnetic field and the Newtonian shear stress under no applied magnetic field.

$$\tau_e = \tau_H - \tau_0 \tag{2-25}$$

Some other measures of controllability are defined in literature and are useful for specific applications. For example Huang [73] uses the control torque ratio (ratio of the field induced torque to the viscous torque) as a parameter to measure controllability of a rotary brake for design purposes.

2.17. Physical Phenomena behind the MR Effect

When subjected to a magnetic field, magnetic particles within an MR fluid orient themselves along the direction of magnetic force and form a stable structure. It is this field induced structure which is responsible for the rheological behavior of magnetic suspensions. Bossis [74] states that "*the chaining of particles and the phase separation which are due to attractive forces between the dipoles induced on each solid particle by the external field, is at the heart of the ER or MR effect."* It has also been stated that the

equilibrium between the breaking and reforming of chains results in the MR fluid's field induced yield stress [29].

Several researchers have found that the application of a magnetic field will result in formation of separate, densely packed, elongated, practically non-interacting aggregates which possess a very small (close to pint) contact area with the wall [75]. Bossis [74] has also stated that the experimentally observed structure does not consist of equally spaced chains of particles but of thick columns.

When considering the resistance to flow of the fluid, MR effect is dependent on relative orientations of magnetic field and fluid flow direction. This resistance to flow has been found to be most pronounced at a mutually perpendicular orientation of magnetic field and direction of fluid flow [20]. There is not much research done on other modes of operation of MR fluids such as parallel field/flow orientation and investigation into other modes of operation, and thus possible applications, for magnetorheological materials is hampered by the relative lack of data and models [44]. Figure 2-22 and Figure 2-23 show the difference between the chain formation in the perpendicular field/flow orientation and parallel field/flow orientation.

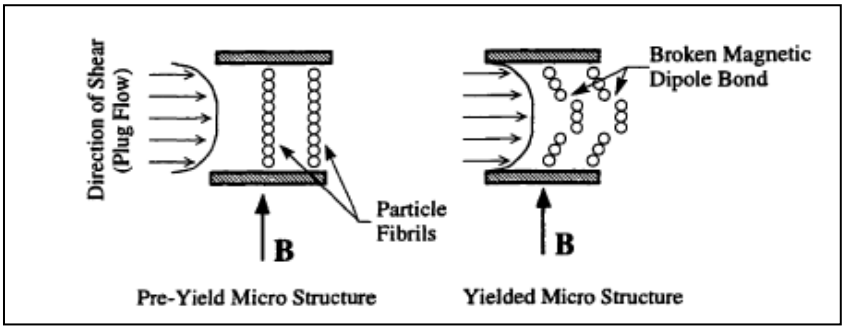

Figure 2-22 The MR effect as the rupture of particle chains (fibrils) in perpendicular field/flow orientation, (Baltimore, C. V., *Field-Flow Orientation Effects in Magnetorheological Fluids*, in *Civil and Environmental Engineering Dept.* 1998, Duke University. [44], used under fair use)

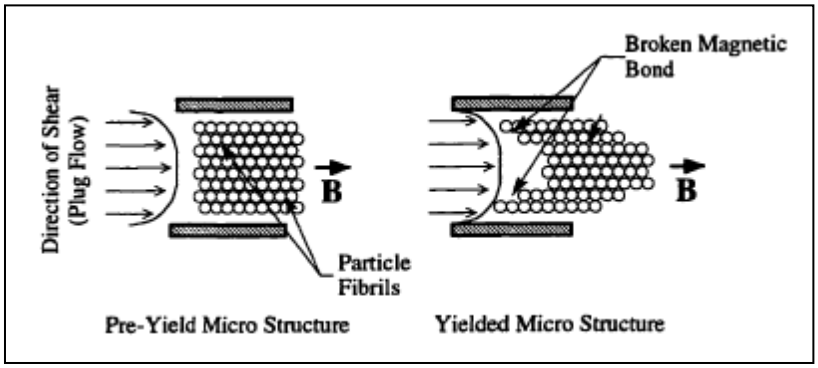

Figure 2-23 The MR effect viewed as rupture between particle chains in parallel field/flow orientation, (Baltimore, C. V., *Field-Flow Orientation Effects in Magnetorheological Fluids*, in *Civil and Environmental Engineering Dept.* 1998, Duke University. [44], used under fair use)

2.18. Super Strong MR Fluids

Since the discovery of MR fluids, there has been some expectation that MR fluids may revolutionize certain industrial sectors. This possibility will become reality only if we can enhance the strength of MR fluids. The current strength of MR fluids, for example, is not sufficient for automobile clutches and manufacturing flexible fixtures. Stronger MR fluids are needed for these applications. In addition, it is especially desirable if these strong MR fluids only require a moderate magnetic field. MR applications need electromagnets inside the MR devices. If the required magnetic field is strong, the electromagnets will be heavy and bulky, making the MR device bulky. The electromagnets are also well known for their delay time since the magnetic coils are an *RL* circuit. To produce a strong magnetic field, the magnetic coils must have a low dc resistance, *R*, and a high inductance, *L*. The stronger the magnetic field, the longer the delay time. For a large electromagnet, delay time *L/R* can be as long as several seconds. While MR fluids themselves respond to an external field rapidly, the delay time of the electromagnets lengthens the response time of MR devices. Only if the required magnetic field is moderate will the size and delay time of electromagnets not become an issue and MR devices will remain agile. This, however, is not an easy task. Efforts in searching for new MR materials in the past decades achieved limited results. The results of current research on the material properties of MR fluid imply that improving the physical properties of magnetic particles alone is unlikely to bring the strength of MR fluids to a required level

[76, 77]. To improve the microstructure of the MR fluid is the key. Current research in this area is concentrated on employing a "rapid compression-assisted aggregation process" to force MR fluids to form a microstructure that is much stronger than the single chain structure [78].

Theoretical simulations and experimental results confirm that formation of a conventional MR microstructure has two developmental stages [77, 79]. The first stage is the chain formation process, which only takes several milliseconds to complete after the magnetic field is applied. The second stage is a relatively slow process, involving the aggregation of chains into thick columns. In simulation, this process takes several seconds. When MR system is large, this process may take a couple of minutes.

For the formation of MR chains, two forces play an important role. One is the dipolar interaction and the other is the Brownian forces. In comparison with the dipolar interaction, the Brownian force is very weak. For most MR fluids, the ratio of the interaction energy of two magnetic dipoles to the thermal energy is quite large. Therefore, at the first stage of chain formation, the Brownian force is negligible in comparison with the interaction between two dipoles. However, once the chains are formed, the dipolar interaction is screened. The interaction between two straight chains is weak and short-ranged. Then the Brownian force plays an important role and easily makes the chains bent or randomly fluctuating. This phenomenon is associated with the instability of a one-dimensional solid. The attracting force between curved chains is stronger than that of straight chains [80]. Therefore, the chains' fluctuations under Brownian motion help their aggregation together. Once several chains get together to form a column, the 3D column is not easy to bend or fluctuate. The Brownian force has little effect on the columns. Thus the column-to-column aggregation is difficult in the unassisted aggregation process and the developed columns are of limited thickness.

The rheological properties and strength of MR fluids are related to the field-induced microstructure. For example, a thick column structure is stronger than a single-chain structure [78]. However, most agile MR

applications require quick response and cannot wait long enough for the columns to form. Moreover, the unassisted aggregation process only produces columns consisting of 10–15 chains [81, 82].

2.18.1 Week Points of MR Microstructure

To develop strong MR fluids, we need to know the rheological properties of induced MR structures. In particular, we want to identify the weak points of these structures and find a way to strengthen them. To begin with, we examine the single-chain structure, which is the basis of all other MR structures.

Consider applying an external shear force on a chain. Under force, the chain deforms, becomes slanted, and then breaks into two broken chains if the shear force exceeds a threshold. The breaking point should be the chain's weak point under shear. The conventional wisdom assumes that the breaking point is in the middle of a chain [78, 80]. The system is symmetric around the middle between the two poles; based on this symmetry, it is natural to guess that the chain could break at the middle. However, this is not the case. After a magnetic field is applied, the space is no longer isotropic. Assume that the shear process is quasi-static such that the chain relaxes into the lowest energy state at each shearing step. Under such a deformation, the configuration with the minimum energy is no longer symmetric. The symmetric configuration becomes unstable. In particular, during the initial deformation, a gap emerges at one end. The gap widens as the strain increases. When the strain exceeds a critical value, the chain breaks. The breaking point is at either end, but not in the middle. In another words, the weak points of an MR chain under shear are at its two ends (Figure 2-24 and Figure 2-25).

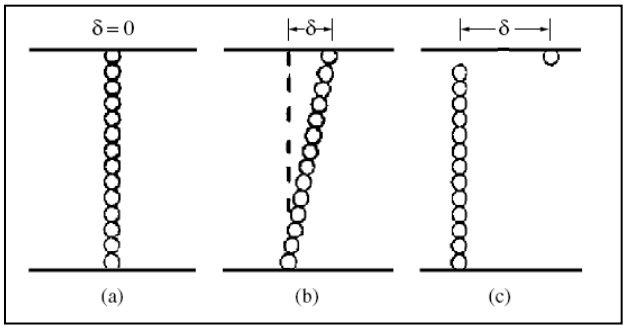

Figure 2-24 (a) A single chain without shear. (b) A slanted chain under a small shear strain. The gap is between the first and second particles. (c) A broken chain when the shear strain exceeds the critical value, (Tao, R., *Super-Strong Magnetorheological Fluids.* Liquids and Soft Matter, Journal of Physics Condensed Matter, 2001. 13(50): p. 979-999. [77], used under fair use)

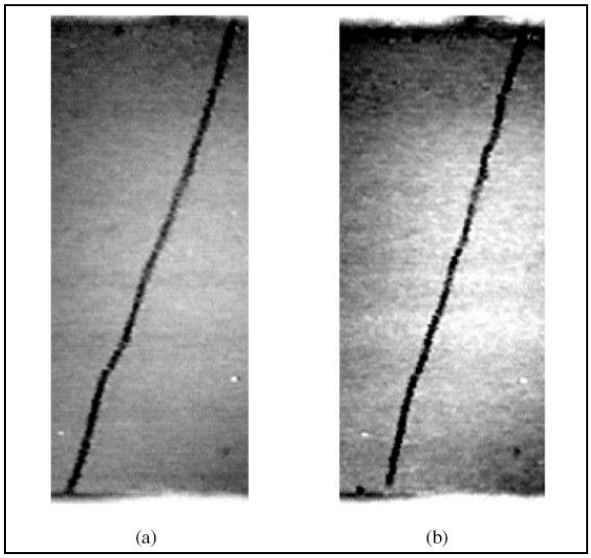

Figure 2-25 (a) Under a shear force, the ER chain becomes slanted. (b) The chain breaks between the first and second particles when the shear strain exceeds the critical value, (Tao, R., *Super-Strong Magnetorheological Fluids.* Liquids and Soft Matter, Journal of Physics Condensed Matter, 2001. 13(50): p. 979-999. [77], used under fair use)

2.18.2 Super Strong MR Fluids

After understanding the microstructure of MR fluids and identifying its weak points, we are in a position to seek a new agile approach to producing super-strong MR fluids. Let us emphasize that the strength of MR fluids comes solely from the induced microstructure. Different microstructures deliver different strengths. Unfortunately, the unassisted-aggregation process, as discussed before, is slow, produces columns of a limited thickness, and is thus not very useful. A technology that can rapidly produce thick columns with strong and robust ends is therefore needed, i.e., thick columns with no weak points at their

ends. This technology is a compression-assisted-aggregation process [83]. Immediately after a magnetic field is applied, we compress the MR fluid along the field direction before a shear force is applied. The magnetic field produces chains in milliseconds. The compression pushes these chains to form thick columns with strong and robust ends. In addition, each column consists of at least a couple of hundred chains. Once the weak points of MR microstructure are strengthened and the columns are very thick, the MR fluids become super-strong.

This process produces strong MR fluids [77, 84, 85]. Shear stress versus shear strain for a typical MR fluid with and without compression-assisted aggregation is shown in Figure 2-26.

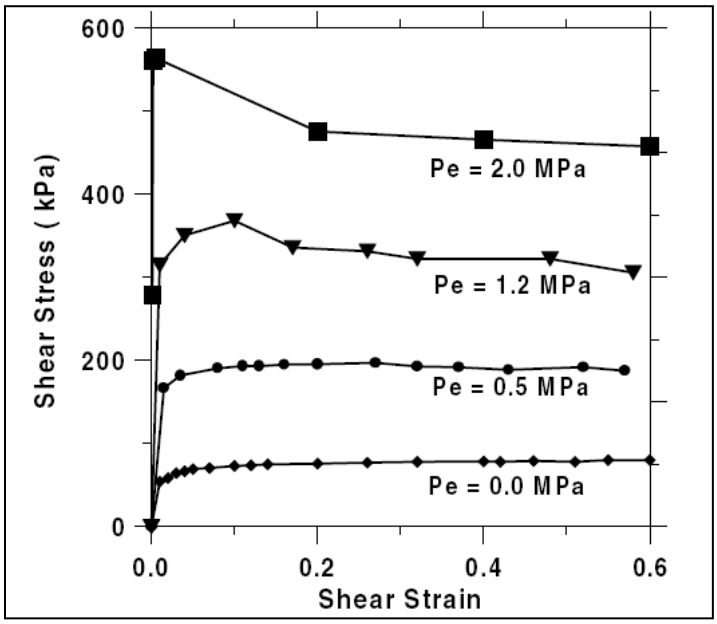

Figure 2-26 a Shear stress versus shear strain curves with and without compression. The magnetic field for all curves is 372 kA m−1, (Tao, R., *Super-Strong Magnetorheological Fluids.* Liquids and Soft Matter, Journal of Physics Condensed Matter, 2001. 13(50): p. 979-999. [77], used under fair use)

This compression-assisted-aggregation process can be illustrated as Figure 2-27. When a magnetic field is applied, magnetic particles quickly form chains. As MR fluid is compressed, chains get shorter and bent. When the chains are bent, the attraction between the chains becomes stronger and pulls these chains quickly together [86]. Meanwhile, as many particles are pushed by the plates, the ends of the columns are

much thicker than their middle. These are the desirable robust structures. The columns produced by this compression process are much thicker than the product of natural aggregation.

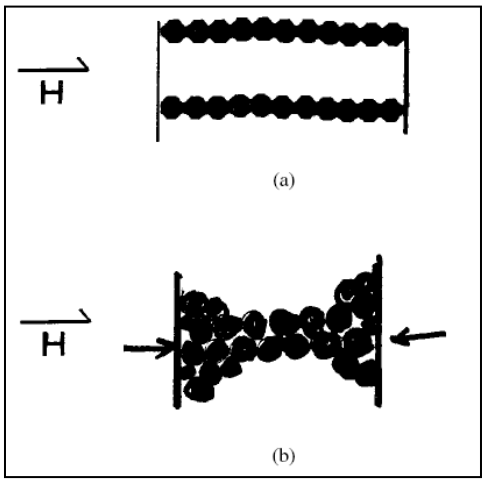

Figure 2-27 Formation of robust MR microstructure during compression-assisted aggregation. (a) Chains before the compression. (b) The compression forces the chains to aggregate into thick columns with robust ends, (Tao, R. and Q. Jiang, *Simulation of Structure Formation in an Electrorheological Fluid.* Physical Review Letters, 1994. 73(1): p. 205. [86], used under fair use)

Super-strong MR fluids are expected to have many industrial applications. For example, the strength of super-strong MR fluids exceeds the requirement for manufacturing flexible fixtures and automobile clutches. The current approach is also applicable to ER fluids. If the structure-enhanced yield stress of ER fluids can also be 10 times higher than the yield stress without compression-assisted-aggregation, this method will enable ER fluids to have a yield stress strong enough for many industrial applications.

2.19.Limitations of MR Technology

While MR fluids have attracted considerable interests in both academic and industrial fields, very few MR devices have been commercialized. This is mainly due to the limitations of MR fluids, such as particle settling, field saturation, wall effects, and response time [13, 15, 26].

Because of density mismatch between solid particles and carrying fluid, particle settling is inevitable even though a certain stabilizer is added to the suspension to delay or reduce the settling [2].

As discussed earlier, maximum yield stress of MR fluids is dependent on their saturation magnetization, so it cannot be increased infinitely by just increasing the external magnetic field.

Wall effect is used to test stress transmission efficiency, which depends on electrode surface [13]. Experimental results show that a rough surface has higher transmission efficiency than a smooth surface [13]. Therefore, it is recommended to use rough surface with high permeability as electrodes [29].

In applying MR fluids, their response time also should be taken into account. It is known that MR fluids have two time constants τ_1 and τ_2 [2]. τ_1 is called chain formation time, which is within one milli-second. τ_2 is system response, including field set up which is around 10 milli-seconds.

2.20.MR Fluids in Squeeze Mode

In this research, squeeze flow of MR fluids is extensively studied analytically and experimentally. MR fluids of various types are tested and their rheological behavior is studied. Mathematical models of different complexity levels are developed to predict MR fluid behavior in squeeze mode. Applications of MR squeeze mode are investigated and two applications are analytically and experimentally investigated. These two applications are namely, MR hybrid dampers and MR squeeze mounts. This section will provide a background about squeeze flow and rheometry and introduces the main concepts and contributions in this area.

As mentioned earlier, MR fluids are used in three distinct flow modes, shown in Figure 2-10. In valve mode, MR fluid is flowing due to a pressure difference perpendicular to magnetic field lines and magnetic poles are stationary. In direct shear mode, one of the poles is moving with respect to other pole and MR fluid is being sheared. The least understood mode is squeeze mode. In squeeze mode, MR fluid is placed between two approaching plates and compressed while magnetic field lines pass through the fluid. Recent research has shown that, in squeeze mode, MR fluids can provide a much larger range of controllable force in small operational envelopes compared to shear and valve modes [87, 88]. While few publications

exist on the behavior of MR fluids in squeeze mode, devices using squeeze mode may take advantage of the very large range of adjustment that squeeze mode offers. Hence, MR devices utilizing squeeze mode can be stronger, smaller, and cheaper than currently available MR devices. Large structural vibration absorption systems, impact dampers, and engine mounts are examples of devices that can benefit from MR squeeze mode control.

It is thought that this mode effectively places the ferrous particle chains in a situation similar to columnar buckling [89]. As with other modes of operation, the strength of the ferrous particle chains is dependent on the intensity of the magnetic field. However, unlike other modes, the useable force generated from squeeze mode in not due to an increase in apparent viscosity. Typically, in squeeze mode devices there is little of no flow of MR fluid. The force supported in this mode is mechanical property of the chains rather than the apparent viscosity change of the fluid.

Mazlan et al. [90] and Kulkarni et al. [91] performed experimental investigations on MR fluids in squeeze mode and obtained stress-strain relationships for squeeze flow of MR fluids. They could show experimentally that tested MR fluids were capable of attaining high compression stresses. Tian and coworkers [92] performed similar experimental investigations on Electro-rheological (ER) fluids. Although very informative experimentally, none of the mentioned works provide a mathematical model about the behavior of MR fluid in squeeze mode. As will be discussed in subsequent chapters, in this research, mathematical models are developed and validated using experimental data [88]. The models are capable of accurately predicting the squeeze force for various conditions and could be used as design tools.

Much of the work performed on applications of MR fluids in squeeze mode to date has focused on vibration controllers for rotor systems [87]. Forte et al. [93], Ahn et al. [94], and Carmignani et al. [95] have developed an MR squeeze film damper system for rotor applications. Wang et al. [96] investigated the dynamic performance of this system and subsequently went on to analyze the mechanical properties

of the film and the unbalanced response characteristics of the MR squeeze film in the damper-rigid rotor system [97]. A thorough study of the stress-strain characteristics in compression of MR fluids in squeeze mode is still not complete.

Another application that has adopted the use of squeeze mode operation is fluid-filled elastomer vibration mounts. In this device a hollow elastomer puck is filled with MR fluid to provide a means of adjusting the stiffness of the device [98]. In some designs, the elastomer puck is capped by ferrous pole plates on the top and bottom to promote the magnetic flux path through the MR fluid gap. Typically, a vibration isolator such as this would be found in a mechanical system with a known operating frequency, such as an engine turning at a constant RPM. In a case such as this, it is possible to design a simple non-adjustable mount with a stiffness that is matched to the system's mass and input frequency in a way that will cancel much of the transmitted vibration. The need for a mount with adjustable stiffness arises when the operating parameters of the system vary such that it may pass through a resonance zone with the mount, which has the potential to create forces far beyond normal conditions and potentially damage equipment. This can commonly occur during the startup or shut down of large-scale steam turbines used in power generation. A mount with adjustable stiffness can be continuously matched to changing operating conditions to provide the optimum vibration cancellation characteristics. These devices are still in their infancy and have yet to be implemented on a commercial scale, but the work by Southern et al. [98] shows promising potential for this application.

Although fairly new in MR technologies, squeeze flow characterization is an old problem in fluid mechanics. "Squeezing flows" are flows in which a material (not necessarily a fluid) is deformed between two parallel surfaces approaching each other (Figure 2-28). Due to the change in geometry, squeeze flows are inherently transient flows.

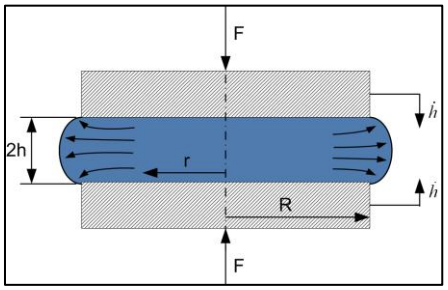

Figure 2-28 Squeeze flow schematics

Squeezing flows are encountered in numerous applications including engineering, biology, food science, and rheometry. Squeeze flow is an attractive technique for measuring rheological properties of highly viscous liquids and materials that create difficulties in conventional rheometers, for example, fluids with an apparent yield stress (e.g., MR fluids in this study), fluids that tend to slip at instrument walls, or materials with large particles. Some advantages of squeeze flow rheometry include ease of set up with existing force testing frames and wide range of achievable strain rates [99].

Stefan [100] was the first to investigate the squeeze flow of Newtonian fluids. Scott [101] extended the work to power law fluids. Since then, various aspects of squeeze flows have been studied by various investigators focusing on rheometry, parameter identification, constitutive equations, numerical simulation, etc. A good review of the squeeze flow rheometry and its applications was done by Engmann and coworkers [99].

Mcclelland and Finlayson [102] developed a theoretical relationship between applied force and plate separation for squeezing flows of viscoelastic liquids. Roussel et al. [103] developed an analytical solution for the squeeze flow problem of a Bingham fluid using variational analysis and compared the results with other published analyses. Adams et al. [104] numerically studied the behavior of an elasto-viscoplastic material using finite element analysis and compared the theoretical simulations with test data. Florides et al. [105] studied the flow and shape evolution during squeezing process of a Bingham fluid using finite element numerical techniques.

Wilson [106] studied a Bingham material being squeezed between two parallel plates and investigated the "squeeze flow paradox". Lawrence and Corfield [107] studied the kinematic inconsistency of the squeezing flow due to the lubrication assumption and showed that the inconsistency can be resolved by re-scaling the variables. Smyrnaios and Tsamopoulos [108] examined the Bingham and Papanastasiou [109] models and showed qualitatively that un-yielded material must exist only around the two stagnation points of flow around the center of the disks.

Roussel and Lanos [110] used experimental results from squeeze tests for parameter identification of several plastic materials. Adams et al. [111] considered the squeeze flow of plastic fluids and compared the conventional lubrication theory with a plasticity analysis and concluded that a major difference between the two approaches was the assignment of the value of the local wall normal stress at the edge of a specimen. Meeten [112] tested various structured fluids and obtained relationships for experimental determination of yield stress and compared the results with other models. Sherwood [113] proposed a method for extracting rheological data from squeeze-flow tests based on the lubrication approximation for a generalized Newtonian fluid.

Meeten [114] experimentally studied the effects of plate roughness in squeeze-flow rheometry of Newtonian and non-Newtonian fluids. Matsoukas and Mitsoulis [115] considered geometry effects on plug flow region size in squeeze flow of Bingham fluids. They showed that axisymmetric geometries result in smaller plug flow regions compared to planar surfaces, while larger aspect ratios give larger plug flow regions for the same Bingham numbers.

Chapter 3
Mathematical Modeling of MR Fluids in Squeeze Mode

This chapter is devoted to mathematical modeling of MR fluids in the squeeze mode. Because there is no universally accepted mathematical model for MR fluids in the squeeze mode, substantial amount of time and effort has been spent on mathematical modeling of squeeze flow of MR fluids. Two approaches have been taken to solve the squeeze problem of MR fluids. The first approach resulted in a simple mathematical model that can accurately predict the total squeezing force. This mathematical model is validated using experimental test data obtained from a novel squeeze mode rheometer (as will be discussed in the next chapter). Although this model is capable of predicting the squeeze force accurately, in later stages of design of MR squeeze mode devices, some additional information is needed. Having a good understanding of flow field, shear rate distribution, and pressure distribution, in addition to squeeze force, will greatly help in design of MR devices. For example, plug flow shape and size are important design considerations that can be studied from shear rate distribution plots. In MR dampers, the damping coefficient is a function of plug thickness. In MR valves, plug thickness is used to control the flow rate through and the pressure drop across, the MR valve. An MR rotary clutch can transfer maximum torque if the entire gap is plug given no wall slip effects are present. In order to have a mathematical model capable of studying flow field, shear rate distribution, and pressure distribution, a second mathematical model was developed. This mathematical model is more mathematically involved than the simple model and may not be needed for preliminary design stages of MR devices but will be helpful for optimization of the existing designs or design of devices in which design decisions highly depend on shear rate and pressure

distributions. Perturbation techniques are used in this model and as a result, the final solutions are obtained as closed-form equations which reduce the need for more complicated and computationally expensive numerical methods. Similar to the first model, the second model is also validated using experimental data. Validations of the mathematical models are presented after the experimental test setup and rheometer design are presented. In this chapter, both models are introduced, discussed, and the final model predictions are presented.

3.1. Simplified Model

As discussed in the literature review, in the squeeze mode, MR fluid is placed between two approaching surfaces and compressed while a magnetic field is passed through the MR fluid. This section discusses a simple model that is developed to predict the total squeezing force as the two squeezing surfaces approach each other. Two separate geometries have been considered. In the first geometry, the upper surface moves axially in a cylinder and it has a center hole through which the excess MR fluid will escape through, to prevent fluid lock up as MR fluid is compressed. This is shown in Figure 3-1a. This geometry is similar to the operating condition of the squeeze mode rheometer which will be discussed in the next chapter. The second geometry is consisting of two circular disks that approach each other and compress the MR fluid in between. The excess MR fluid will escape from the circumference of the plates to prevent fluid lock up. This is shown in Figure 3-1b. This geometry is used to design and build an "MR pouch" that will be discussed in later chapters.

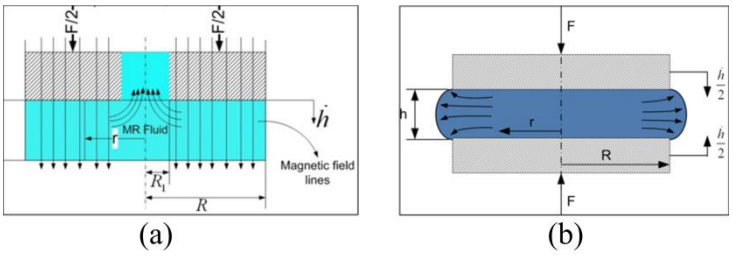

Figure 3-1 Geometries used for squeeze flow of MRF

3.1.1 Configuration I: A Cylinder with a Center Hole

Figure 3-2 schematically shows the squeeze flow configuration for MR fluid. As the plates move towards each other, the excess MR fluid moves through the middle hole to prevent fluid lock up. Other geometries that may be used for MR fluids in squeeze mode will be discussed later in this chapter. The geometry shown in Figure 3-2 is similar to the geometry of the squeeze-mode rheometer that will be discussed in Chapter 4.

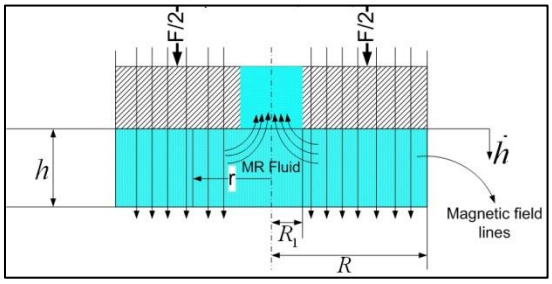

Figure 3-2 Schematics of squeeze flow problem. In this case, the excess MR fluid escapes from the middle hole as the plates move towards each other.

As the MR fluid is squeezed, it is assumed that the pressure in the MR fluid is built up due to three different factors [116]:

- Pressure due to the viscosity of the MR fluid. This pressure is present at all times regardless of the presence of a magnetic field.
- The pressure due to the MR effect. This pressure is caused by the resistance of the chains in the MR structure, and is present only when the magnetic field is applied.
- The pressure due to the inertia of MR fluid. Since the amount of MR fluid between the disks is small compared to other inertias, this pressure can be neglected.

3.1.1.1 Viscous Pressure

To analyze the viscosity effects of this geometry, it is assumed that the fluid radial flow can be approximated by [117],

$$q(r) = -\frac{b(r)h^3}{12\eta}\left(\frac{\partial p_\eta}{\partial r}\right) \tag{3-1}$$

where, p_η is the pressure due to viscosity, η is the viscosity of the MR fluid, h is the gap size, and

$$\begin{aligned} b(r) &= 2\pi r \\ q(r) &= -\pi\left(r^2 - R_1^{\,2}\right)\dot{h} \end{aligned} \tag{3-2}$$

$b(r)$ is schematically shown in Figure 3-3. $q(r)$ is the rate of change of volume when the upper disk is moving at a speed of $\dot{h}$ (Figure 3-2).

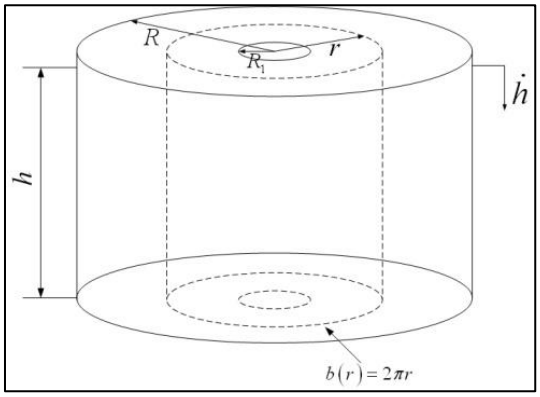

Figure 3-3 b(r) is the base of the cylinder at r and h is the gap size

Replacing equations (3-2) into equation (3-1) results

$$\left(\frac{\partial p_\eta}{\partial r}\right) = \frac{6\eta\left(r^2 - R_1^{\,2}\right)}{rh^3}\dot{h} \tag{3-3}$$

Integrating equation (3-3) with respect to r with the boundary condition $p_\eta(R_1) = 0$, the viscous pressure as a function of radius is calculated as,

$$p_\eta(r) = \frac{3\eta\dot{h}}{h^3}\left(r^2 - 2R_1^{\,2}\ln r\right) + \frac{3\eta\dot{h}R_1^{\,2}}{h^3}\left(2\ln R_1 - 1\right) \tag{3-4}$$

3.1.1.2 Pressure Due to MR Effect

The pressure due to MR effect depends on the yield stress of the MR fluid. The yield stress (τ_y) of the MR fluid is a function of material properties as well as magnetic field density. It is assumed that the MR fluid behaves like a Bingham fluid in the presence of a magnetic field. The pressure differential can be expressed by [49],

$$\frac{\partial p_{MR}}{\partial r} = \frac{2\tau_y}{h}\operatorname{sgn}\left(\dot{h}\right) \tag{3-5}$$

where, τ_y is the shear mode yield stress in the Bingham model. Assuming $\dot{h} > 0$ downward, when the MR fluid is squeezed, equation (3-5) becomes

$$\frac{\partial p_{MR}}{\partial r} = \frac{2\tau_y}{h} \tag{3-6}$$

As the MR fluid is squeezed, the yield stress is increased because the chains bend and form thicker chains which are more difficult to break. An empirical relation is proposed by R. Tao [77] which relates the shear yield stress to the compressive pressure exerted on the fluid.

$$\tau_y = \tau_{y0} + K_H P_{MR} \tag{3-7}$$

where, τ_{y0} is the shear yield stress at no compressive pressure, K_H is a multiplication factor, and P_{MR} is the pressure created in the MR fluid due to the MR effect.

Replacing equation (3-7) in equation (3-6) results in

$$\frac{\partial P_{MR}}{\partial r} = \frac{2}{h}\left(\tau_{y0} + K_H P_{MR}\right) \tag{3-8}$$

Integrating equation (3-8), assuming the yield stress is zero at $r = R_1$ (the pressure, P_{MR} is also zero), the pressure due to the MR effect is

$$P_{MR} = \frac{\tau_{y0}}{K_H}\left[-1+e^{\left(\frac{2K_H(r-R_1)}{h}\right)}\right] \tag{3-9}$$

3.1.1.3 Total Pressure and Force

As stated earlier, the pressure due to inertia of the MR fluid is neglected. Therefore, the total pressure is the sum of the viscous pressure and the pressure due to MR effect.

$$p(r) = p_\eta(r) + p_{MR}(r) \tag{3-10}$$

$$p(r) = \frac{3\eta\dot{h}}{h^3}\left[r^2 - R_1^2 - 2R_1^2\ln\left(\frac{r}{R_1}\right)\right] + \frac{\tau_{y0}}{K_H}\left[-1+e^{\left(\frac{2K_H(r-R_1)}{h}\right)}\right] \tag{3-11}$$

Assuming the upper disk is moving downward and integrating the total pressure on the surface of the upper disk the total force would be

$$\begin{aligned} F &= 2\pi\int_{R_1}^{R} rp(r)dr \\ F &= \frac{-\pi\eta\dot{h}}{2h^3}\left[3R_1^4 + 12R_1^2R^2\ln\left(\frac{R}{R_1}\right) - 3R^4\right] \\ &\quad -\frac{\pi\tau_{y0}}{2K_H^3}\left[2K_HR_1h - 2K_H^2R_1^2 - h^2 + \left(h^2 - 2RK_Hh\right)e^{\left(\frac{2K_H(R-R_1)}{h}\right)} + 2K_H^2R^2\right] \end{aligned} \tag{3-12}$$

The first term on the right hand side of equation (3-12) is the force generated by the viscosity of the MR fluid and second term is the force caused by the magnetic effect of the MR fluid. As can be seen in equation (3-12), for relatively large gap sizes the viscous force is very small and can be neglected. But as the gap gets smaller, the viscous force increases rapidly, inversely proportional to the 3rd power of the gap size.

3.1.1.4 Non-dimensional Model

Transforming equation (3-12) to a non-dimensional form gives the ability to better understand the behavior of the MR fluid and make some means to compare its performance in squeeze mode with other modes. The force equation could be non-dimensionalized by introducing the following non-dimensional parameters

$$\begin{aligned} \zeta &= \frac{h}{R} \\ Bn &= \frac{R\tau_{y0}}{\eta \dot{h}} \\ \psi &= \frac{F}{\pi R^2 \tau_{y0}} \\ R_1^* &= \frac{R_1}{R} \end{aligned} \tag{3-13}$$

ζ is the non-dimensional gap size and represents the aspect ratio of the gap. A small value of ζ represents a small gap size, while a large value shows a large gap size. The Bingham number, Bn, can be considered as the ratio of the MR effect to the viscous effect. Therefore, a large Bn shows that the viscous effects are relatively small compared to the MR effect.

The term, $\psi = F / \pi R^2 \tau_{y0}$, is the ratio of the squeeze force to the yield stress multiplied by the area of the pole. It can be used to compare squeeze mode and direct shear mode forces. The denominator of ψ can be interpreted as the force acting on the same area and the same MR fluid to yield the MR fluid in direct shear mode, as is schematically shown in Figure 3-4. Therefore, ψ shows how much larger or smaller the squeeze mode force is, when compared to the shear force.

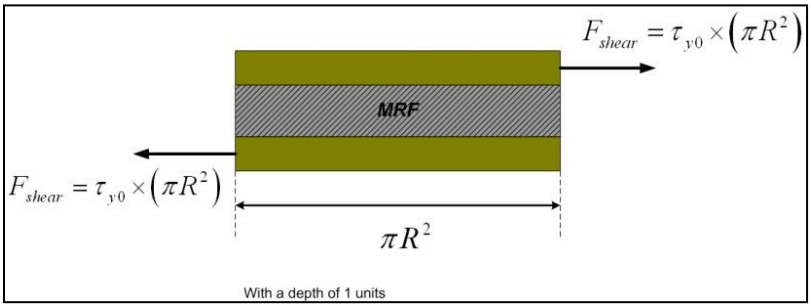

Figure 3-4 Schematics of the force needed to yield the MR fluid in direct shear mode

Using the non-dimensional parameters introduced in equation (3-13), the non-dimensional force equation can be obtained as

$$\psi = \frac{1}{\zeta^3 Bn}\left(\frac{3}{2} - \frac{3}{2}R_1^{*4} + 6R_1^{*2}\ln R_1^*\right) + \frac{1}{K_H}\left(R_1^{*2} - 1\right) + \frac{\zeta}{K_H{}^2}\left(e^{\left(\frac{2K_H\left(1-R_1^*\right)}{\zeta}\right)} - R_1^*\right) + \frac{\zeta^2}{2K_H{}^3}\left(1 - e^{\left(\frac{2K_H\left(1-R_1^*\right)}{\zeta}\right)}\right) \tag{3-14}$$

For large Bingham numbers, the first term, which is the viscous term, is negligible compared to the MR effect. For this case, the non-dimensional force would become approximately,

$$\psi = \frac{1}{K_H}\left(R_1^{*2} - 1\right) + \frac{\zeta}{K_H{}^2}\left(e^{\left(\frac{2K_H\left(1-R_1^*\right)}{\zeta}\right)} - R_1^*\right) + \frac{\zeta^2}{2K_H{}^3}\left(1 - e^{\left(\frac{2K_H\left(1-R_1^*\right)}{\zeta}\right)}\right) \tag{3-15}$$

In equation (3-15), the non-dimensional force is only a function of geometry and gap size. Because the yield stress of the MR fluid is included in the non-dimensional force ψ, equation (3-15) can also be used when the yield stress varies as the gap size and magnetic field density vary. At each gap size, the numeric value of ψ can be multiplied by $\pi R^2 \tau_{y0}(h)$ to obtain the squeeze force *F*. Note that when using equation (3-15), the yield stress, τ_{y0}, can be a function of the gap size, *h*. Therefore, this model can include the variations in the yield stress due to the change in gap size.

3.1.2 Configuration II: Two Circular Disks

The second design configuration that will be considered is shown in Figure 3-5. As shown, in this configuration, two circular plates approach each other and compress the MR fluid in between.

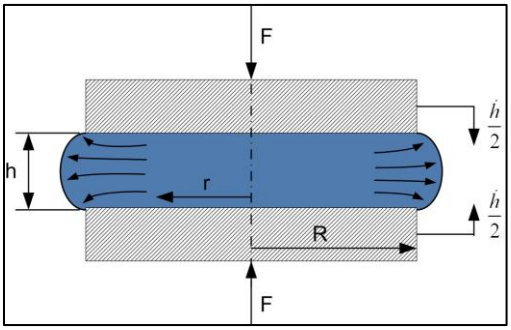

Figure 3-5 Two circular plates compressing MR fluid. The excess MR fluid will escape from the circumference of the plates

Similar to the previous case considered in section 3.1.1, we assume the fluid radial flow can be approximated by the equation for laminar flow q in a wide channel given by,

$$q = \frac{2b(h/2)^3}{3\eta}\left(-\frac{\partial p_\eta}{\partial r}\right) \tag{3-16}$$

Note that if $\partial p_\eta / \partial r > 0,\ q < 0$: flow is in the negative r direction. If $\partial p_\eta / \partial r < 0,\ q > 0$: flow is the positive r direction. Given the geometry of the MR pouch as defined in Figure 3-5, the dimensions for the flow in a wide channel are

$$\begin{aligned} b &= 2\pi r \\ q &= \pi r^2 \dot{h} \end{aligned} \tag{3-17}$$

Substituting (3-17) into (3-16) gives us the relationship of the viscous pressure in the pouch as a function of the geometry considering that h/b <<1 [118]

$$\frac{\partial p_\eta}{\partial r} = \frac{-6\eta r \dot{h}}{h^3} \tag{3-18}$$

Integrating with respect to *r* using the boundary condition $p_\eta(R)=0$, the viscous pressure would become

$$p_\eta(r)=\frac{-3\eta\dot{h}}{h^3}\left(r^2-R^2\right) \tag{3-19}$$

Considering that the MR fluid behaves as a Bingham fluid, the flow of MR fluid can be approximated as flow through a wide channel (shown in Figure 3-6) [49]:

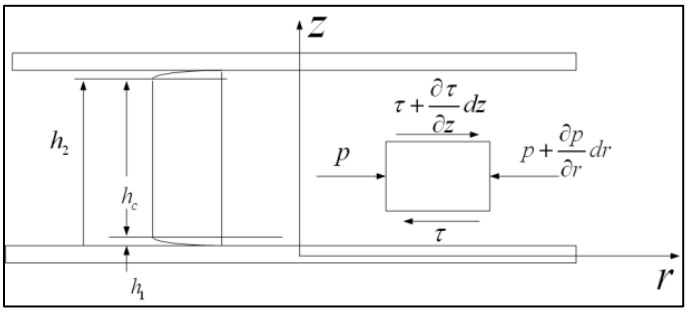

Figure 3-6: An element of the MR fluid in squeeze mode.

Neglecting the inertial effects, the equilibrium of a fluid element, $\sum F_r=0$ gives

$$\frac{\partial p_{MR}}{\partial r}=\frac{\partial\tau}{\partial z} \tag{3-20}$$

Integrating with respect to *z* results in,

$$\tau=\frac{\partial p_{MR}}{\partial r}z+C_1 \tag{3-21}$$

Because of symmetry of the flow with respect to *z*, at $z=h/2$,

$$\tau(z)\Big|_{z=\frac{h}{2}}=0 \tag{3-22}$$

Solving for C_1 yields,

$$\tau=\frac{\partial p_{MR}}{\partial r}\left(z-\frac{h}{2}\right) \tag{3-23}$$

It is also noteworthy that the shear stress is linear with respect to *z*, which is to be expected. At $z = h_1$, $\tau = \tau_y$ (considered plug flow)

$$\tau_y = \frac{\partial p_{MR}}{\partial r}\left(h_1 - \frac{h}{2}\right) \tag{3-24}$$

$$h_1 = \frac{\tau_y}{\frac{\partial p_{MR}}{\partial r}} + \frac{h}{2} \tag{3-25}$$

Assuming $h_1 << h_2$, $h_c = h$ (seen in Figure 3-6) and,

$$\frac{\partial p_{MR}}{\partial r} = \frac{-2\tau_y}{h} \tag{3-26}$$

Similar to previous case, it can be assumed that the yield stress of the MR fluid is a function of the magnetic flux density and the amount of compression pressure [77].

$$\tau_y = \tau_{y0} + K_H p_{MR} \tag{3-27}$$

The total force on the MR element is then found by integrating the total pressure ($p_\eta + p_{MR}$) on the surface of the poles,

$$F = \pi R^2 \tau_{y0}\left(\frac{-1}{K_H} + \frac{1}{2}\frac{h^2 e^{\frac{2K_H R}{h}}}{{K_H}^3 R^2} + \frac{3}{2}\frac{R^2 \eta \dot{h}}{h^3 \tau_{y0}} - \frac{1}{2}\frac{h^2}{{K_H}^3 R^2} - \frac{h}{{K_H}^2 R}\right) \tag{3-28}$$

Using equation (3-28), the behavior of the MR squeeze mode force element can be plotted and compared against test data. This will be done when discussing model validation and experimental testing of "MR pouch" in later chapters.

3.2. Perturbation Model

This section covers the description and details of a more complicated model for squeeze flow of MR fluids. Perturbation techniques are used to solve the system of Partial Differential Equations (PDEs). This model is developed for the MR pouch and therefore, it assumes a squeezing configuration similar to Figure 3-5 redrawn in Figure 3-7.

3.2.1 Problem Statement and Assumptions

Consider an MR fluid contained between two parallel plates that are initially separated by a gap of height $2h_0$ (as shown in Figure 3-7). A normal force *F* is applied to the plates to cause the fluid in the gap to be squeezed out in the radial direction. Note that in this section, the gap between the plates is defined as *2h* and the plates' velocities are $\dot{h}$ for each plate due to the symmetrical nature of the problem. Therefore, this notation is different from that one of previous sections.

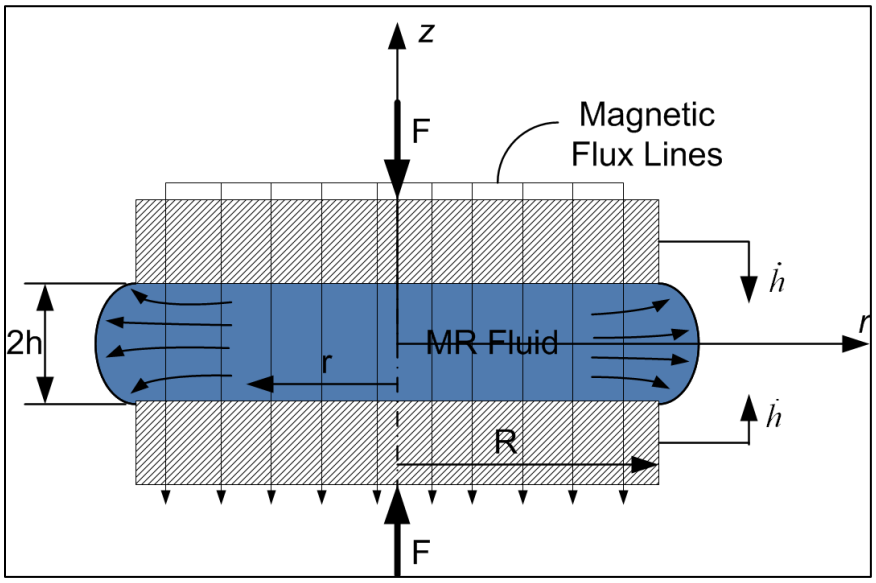

Figure 3-7 MR fluid contained between two approaching parallel circular disks

It is assumed that the quasi-static assumption can be used and the flow is axisymmetric. As a result, velocity and pressure are only functions of *r*- and *z*-coordinates, the θ-component of velocity is zero, and

the plates' velocity, $\dot{h}$, is small. Additionally, it is assumed that inertial and gravitational effects are small and can be neglected. The velocity field (v_r, v_z), shear rate ($\dot{\Gamma}$) distribution, pressure (p) distribution, and squeeze force, F, are desired as functions of gap size.

The system of equations will be solved in non-dimensional forms in subsequent sections. The following parameters are defined in a non-dimensionalized form as:

$$\begin{aligned} V &= \frac{-\dot{h}R}{h} \\ \underline{\underline{\dot{\Gamma}}} &= \frac{\underline{\underline{\dot{\gamma}}}h}{V} \\ \underline{\underline{T}} &= \frac{\underline{\underline{\tau}}}{\eta_0 V/h} \\ \varepsilon &= \frac{h}{R} \end{aligned} \tag{3-29}$$

V is the characteristic velocity, $\underline{\underline{\tau}}$ is the stress tensor, $\underline{\underline{\dot{\gamma}}}$ is the rate of strain tensor, and $\underline{\underline{\dot{\Gamma}}}$ and $\underline{\underline{T}}$ are non-dimensional rate of strain and stress tensors, respectively. The aspect ratio of the gap is denoted by ε.

3.2.2 Constitutive Equation

Experimental rheological tests have shown that MR fluids behave as quasi-Newtonian fluids at very low shear rates (< 10^{-3} 1/s). Resiga [119] performed extensive rheological tests on MRF-132DG and found that a rheological model can be obtained for MR fluids by blending a quasi-Newtonian behavior at very low shear rates with a Herschel-Bulkley model for large shear rates. Herschel-Bulkley model was used because a shear-thinning behavior was observed at very high shear rates. If shear rates encountered in the problem are not too high to be in the shear-thinning region, such as the problem dealt with in this paper, the shear-thinning behavior can be ignored to simplify mathematics. Instead, a Bingham behavior is used.

$$\tau(\dot{\gamma}) = \tau_N(\dot{\gamma})W_1(\dot{\gamma}) + \tau_{Bn}(\dot{\gamma})W_2(\dot{\gamma}) \tag{3-30}$$

where, $\dot{\gamma}$ is shear rate and

$$\begin{aligned} \tau_N(\dot{\gamma}) &= \eta_0 \dot{\gamma} \\ \tau_{Bn}(\dot{\gamma}) &= \tau_y + \eta_0 \dot{\gamma} \end{aligned} \tag{3-31}$$

Off-state viscosity is shown as η_0 and τ_y is yield stress. The value of the off-state viscosity is provided by the MR fluid supplier and is considered to be constant and independent of magnetic field density. The blending functions, W_1 and W_2 should be chosen such that at low shear rates, $W_1(\dot{\gamma}) >> W_2(\dot{\gamma})$ and at high shear rates, $W_2(\dot{\gamma}) >> W_1(\dot{\gamma})$, as discussed in [119]. In the present work, the following functions are used as blending functions,

$$\begin{aligned} W_1(\dot{\gamma}) &= e^{-n\dot{\gamma}} \\ W_2(\dot{\gamma}) &= 1 - e^{-n\dot{\gamma}} \end{aligned} \tag{3-32}$$

The value of n shows how fast or slow the low shear rate quasi-Newtonian behavior is switched to a Bingham behavior as shear rate is increased, as shown in Figure 3-8. Replacing equations (3-31) and (3-32) in (3-30) and simplification results in viscosity to be of the form

$$\eta(\dot{\gamma}) = \eta_0 + \frac{\tau_y\left(1 - e^{-n\dot{\gamma}}\right)}{\dot{\gamma}} \tag{3-33}$$

Using equation (3-33) as viscosity for MR fluid, the constitutive equation of MR fluid can be written as,

$$\underline{\underline{\tau}} = -\left[\eta_0 + \frac{\tau_y\left(1 - e^{-n\dot{\gamma}}\right)}{\dot{\gamma}}\right]\underline{\underline{\dot{\gamma}}} \tag{3-34}$$

where, the sign convention of [120] is used. Equation (3-34) is known as the Papanastasiou constitutive equation [109]. Although the off-state viscosity is constant and independent of the magnetic field density, when shear rate approaches zero, the startup (pre-yield) viscosity is $\eta_0 + n\tau_y$ which depends on the

magnetic field through the values of the yield stress and n. $1/n$ can be considered as the characteristic shear rate as discussed in [119].

Using the parameters introduced in (3-29), the Papanastasiou model can be written in non-dimensional form as

$$\underline{\underline{T}} = -\left[1 + \frac{Bn\left(1-e^{-N\dot{\Gamma}}\right)}{\dot{\Gamma}}\right]\underline{\underline{\dot{\Gamma}}} \tag{3-35}$$

where, $Bn = \tau_y h/(V\eta_0)$ is the Bingham number and $N = nV/h$ is the non-dimensional regularization parameter for Papanastasiou model. $\dot{\Gamma}$ is the non-dimensional shear rate value and will be discussed in next section.

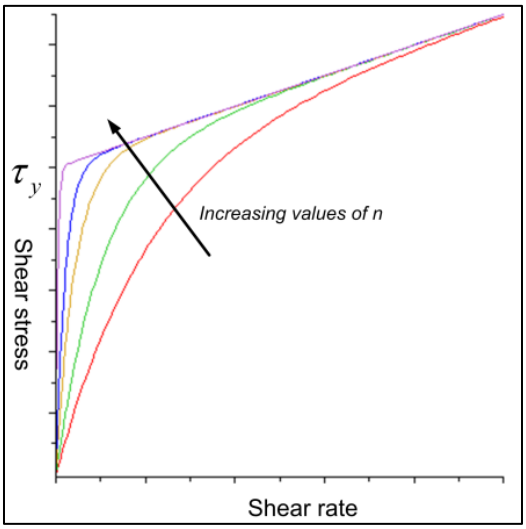

Figure 3-8. Non-dimensional Papanastasiou model as *n* varies

As far as MR technology is concerned, there have been numerous studies on mathematical modeling of MR fluids at high shear rates [121-123]. But, at low shear rates, there is no universally accepted mathematical model for behavior of MR fluids.

3.2.3 System of Equations

Since the problem is axisymmetric, only *r*- and *z*-components of the equation of motion need to be solved. A perturbation solution is desired. Because the plates' speed is $\dot{h}$, the *z*-component of velocity is in the order of $O\left(-\dot{h}\right)$. Using the continuity equation, it can be easily seen that the *r*-component of the velocity is in the order of $O\left(-R\dot{h}/h\right)$. Hence, velocity components are expanded in terms of the gap aspect ratio (ε) as

$$v_r = V\left[\tilde{v}_{r0} + \varepsilon\tilde{v}_{r1} + \varepsilon^2\tilde{v}_{r2} + ...\right] \tag{3-36}$$

$$v_z = V\left[\varepsilon\tilde{v}_{z1} + \varepsilon^2\tilde{v}_{z2} + ...\right] \tag{3-37}$$

where, V is the characteristic velocity defined in (3-29) and the tilde denotes dimensionless quantities. Introducing the non-dimensional coordinates $\zeta = z/h$ and $\xi = r/R$, and using definitions in (3-29), the equations of motion and continuity can be written in non-dimensional form as

$$r-component:\ 0 = -\varepsilon\frac{\partial P}{\partial \xi} - \left(\varepsilon\frac{1}{\xi}\frac{\partial}{\partial \xi}\left(\xi T_{rr}\right) - \varepsilon\frac{T_{rr}}{\xi} + \frac{T_{rz}}{\partial \zeta}\right) \tag{3-38}$$

$$z-component:\ 0 = -\frac{\partial P}{\partial \zeta} - \left(\varepsilon\frac{1}{\xi}\frac{\partial}{\partial \xi}\left(\xi T_{rz}\right) + \frac{\partial T_{zz}}{\partial \zeta}\right) \tag{3-39}$$

$$continuity:\ 0 = \frac{1}{\xi}\frac{\partial}{\partial \xi}\left(\xi\frac{v_r}{V}\right) + \frac{\partial}{\partial \zeta}\left(\frac{v_z}{V}\right) \tag{3-40}$$

$P = ph/\left(\eta_0 V\right)$ is the non-dimensional pressure. The relationship between stress components in equations (3-38) and (3-39), and velocity components is obtained from the constitutive equation (3-35). The shear rate in equation (3-35) is defined as

$$\dot{\Gamma} = \frac{\dot{\gamma} h}{V} = \frac{h}{V}\sqrt{\left(\frac{\partial v_r}{\partial z} + \frac{\partial v_z}{\partial r}\right)^2 + 2\left(\frac{v_r}{r}\right)^2 + 2\left(\frac{\partial v_r}{\partial r}\right)^2 + 2\left(\frac{\partial v_z}{\partial z}\right)^2} \tag{3-41}$$

After substituting for velocity components in equation (3-41) and using the corresponding expansions in (3-36) and (3-37), shear rate can be written in terms of non-dimensional quantities.

$$\dot{\Gamma} = \left\{\left(\frac{\partial \tilde{v}_{r0}}{\partial \zeta}\right)^2 + \varepsilon\left(2\frac{\partial \tilde{v}_{r0}}{\partial \zeta}\frac{\partial \tilde{v}_{r1}}{\partial \zeta}\right) + \varepsilon^2\left[\left(\frac{\partial \tilde{v}_{r1}}{\partial \zeta}\right)^2 + 2\left(\frac{\tilde{v}_{r0}}{\xi}\right) + 2\left(\frac{\partial \tilde{v}_{r0}}{\partial \xi}\right)^2\right] + O\left(\varepsilon^3\right)\right\}^{1/2} \tag{3-42}$$

Assuming the gap aspect ratio is small, equation (3-42) can be expanded as

$$\dot{\Gamma} = \frac{-\partial \tilde{v}_{r0}}{\partial \zeta}\left(1 + \frac{\partial \tilde{v}_{r1}/\partial \zeta}{\partial \tilde{v}_{r0}/\partial \zeta}\varepsilon + \frac{\left(\tilde{v}_{r0}/\xi\right)^2 + \left(\partial \tilde{v}_{r0}/\partial \xi\right)^2}{\left(\partial \tilde{v}_{r0}/\partial \zeta\right)^2}\varepsilon^2 + O\left(\varepsilon^3\right)\right) \tag{3-43}$$

Due to the quasi-static assumption, the shear rates encountered in the problem are small. This allows for writing the viscosity in equation (3-35) as

$$\tilde{\eta}\left(\dot{\Gamma}\right) = 1 + \frac{Bn\left(1 - e^{-N\dot{\Gamma}}\right)}{\dot{\Gamma}} = 1 + BnN - \frac{1}{2}BnN^2\dot{\Gamma} + O\left(\dot{\Gamma}^2\right) \tag{3-44}$$

Equation (3-44) implies that the viscosity of the fluid can be considered a linear function of shear rate at shear rates approaching zero. However, this approximation is only valid given the exponent $N\dot{\Gamma}$ is sufficiently small. As shear rate is increased, the linear approximation will deviate from the Papanastasiou empiricism, as shown in Figure 3-9.

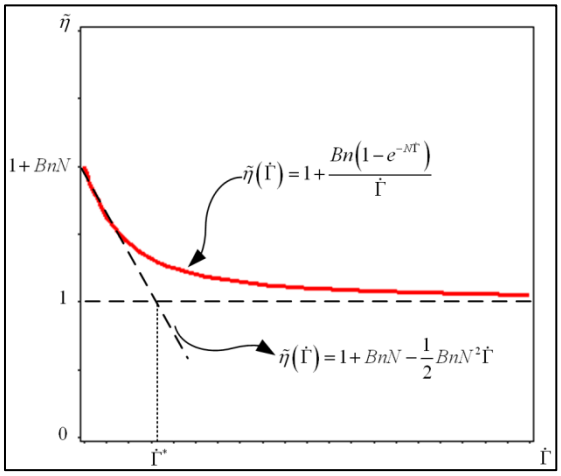

Figure 3-9. The Papanastasiou empiricism and its Taylor expansion

In order to have a good approximation, the value of shear rate should be small. At higher shear rates, viscosity can be considered to be constant.

$$\eta(\dot{\Gamma}) = \begin{cases} 1 + BnN - \frac{1}{2}BnN^2\dot{\Gamma} & \dot{\Gamma} \le \dot{\Gamma}^* \\ 1 & \dot{\Gamma} > \dot{\Gamma}^* \end{cases} \tag{3-45}$$

Combining equations (3-43) and (3-44), the fluid viscosity can be written in terms of ε.

$$\tilde{\eta}(\dot{\Gamma}) = A + B\varepsilon + C\varepsilon^2 \tag{3-46}$$

where,

$$\begin{aligned} A &= 1 + BnN + \frac{1}{2}BnN^2\frac{\partial \tilde{v}_{r0}}{\partial \zeta} \\ B &= \frac{1}{2}BnN^2\frac{\partial \tilde{v}_{r1}}{\partial \zeta} \\ C &= \frac{1}{2}BnN^2\frac{\left(\tilde{v}_{r0}/\xi\right)^2 + \left(\partial \tilde{v}_{r0}/\partial \xi\right)}{\partial \tilde{v}_{r0}/\partial \zeta} \end{aligned} \tag{3-47}$$

The last variable that needs to be expanded is pressure. Performing a scaling analysis on the *r*-component of the equation of motion reveals that the pressure should be expanded as

$$P = P_{-1}\varepsilon^{-1} + P_0 + P_1\varepsilon^1 + O\left(\varepsilon^2\right) \tag{3-48}$$

Inserting the expanded forms of velocities and pressure in equations (3-38) through (3-40) and equating the equal powers of ε, the following system of differential equations is obtained.

$$O\left(\varepsilon^{-1}\right): -\frac{\partial P_{-1}}{\partial \zeta} = 0 \tag{3-49}$$

$$O\left(\varepsilon^0\right): \begin{cases} -\dfrac{\partial P_0}{\partial \zeta} = 0 & (a) \\ -\dfrac{\partial P_{-1}}{\partial \xi} + \dfrac{\partial}{\partial \zeta}\left(A\dfrac{\partial \tilde{v}_{r0}}{\partial \zeta}\right) = 0 & (b) \\ \dfrac{1}{\xi}\dfrac{\partial}{\partial \xi}\left(\xi \tilde{v}_{r0}\right) + \dfrac{\partial}{\partial \zeta}\tilde{v}_{z1} = 0 & (c) \end{cases} \tag{3-50}$$

$$O\left(\varepsilon^1\right): \begin{cases} -\dfrac{\partial P_1}{\partial \zeta} + \dfrac{1}{\xi}\dfrac{\partial}{\partial \xi}\left(\xi A\dfrac{\partial \tilde{v}_{r0}}{\partial \zeta}\right) + 2\dfrac{\partial}{\partial \zeta}\left(A\dfrac{\partial \tilde{v}_{z1}}{\partial \zeta}\right) = 0 & (a) \\ -\dfrac{\partial P_0}{\partial \xi} + \dfrac{\partial}{\partial \zeta}\left(A\dfrac{\partial \tilde{v}_{r1}}{\partial \zeta} + B\dfrac{\partial \tilde{v}_{r0}}{\partial \zeta}\right) = 0 & (b) \\ \dfrac{1}{\xi}\dfrac{\partial}{\partial \xi}\left(\xi \tilde{v}_{r1}\right) + \dfrac{\partial \tilde{v}_{z2}}{\partial \zeta} = 0 & (c) \end{cases} \tag{3-51}$$

3.2.4 Boundary Conditions

Before the system of equations (3-49) through (3-51) can be solved, the boundary conditions must be specified. Considering the assumptions mentioned earlier and assuming no slip at walls, the boundary conditions are

$$\begin{aligned} &\text{At } z = h: \quad v_z = \dot{h}, \quad v_r = 0 \\ &\text{At } z = 0: \quad v_z = 0, \quad \frac{\partial v_r}{\partial z} = 0 \\ &\text{At } r = R \text{ and } z = h: \quad \pi_{rr} = p_a \end{aligned} \tag{3-52}$$

where, p_a is the atmospheric pressure and π_{rr} is the total normal stress in *r*-direction. It should be noted that the last boundary condition is an approximation. At *t=0*, the free fluid surface is at *r=R* and the third

boundary condition in (3-52) is reasonably realistic. However, as the gap size is decreased during the squeezing process, the fluid free surface moves beyond $r=R$ and it is arguable to assume $\pi_{rr} = p_a$ at a point inside the fluid. This problem is discussed in more detail in [102, 120].

When the dimensionless expansions for velocity, stress, and pressure are combined with the equations from (3-52), the non-dimensional boundary conditions, in terms of the expanded variables, are obtained.

$$\begin{aligned}
&\text{At } \zeta = 1\text{: } \tilde{v}_{z1} = -1,\ \tilde{v}_{zn} = 0\ \ n > 1,\ \tilde{v}_{rn} = 0\ \ n \geq 0 \\
&\text{At } \zeta = 0\text{: } \tilde{v}_{zn} = 0,\ \frac{\partial \tilde{v}_{rn}}{\partial \zeta} = 0\ \ n \geq 0 \\
&\text{At } \zeta = 1 \,\&\, \xi = 1\text{: } P_{-1} = 0,\ P_0 = P_a = {p_a h}\big/{\eta_0 V},\ \ P_n + T_{rr,n} = 0\ \ n \geq 1
\end{aligned} \tag{3-53}$$

$T_{rr,n}$ is the n^{th} order term in the expansion of T_{rr}. The perturbation equations in (3-49) through (3-51) associated with boundary conditions in (3-53) now form a well posed system of differential equations and can be solved in a sequential manner.

3.2.5 Solution to the Perturbation Problem

From equation (3-49) and (a) in (3-50), it can immediately be seen that $P_{-1} = P_{-1}(\xi)$ and $P_0 = P_0(\xi)$, respectively. Assuming small nonlinearities and using the appropriate boundary conditions, $\tilde{v}_{r0}$ is obtained from equation (b) in (3-50).

$$\tilde{v}_{r0} = \frac{1}{2(1+BnN)} \frac{\partial P_{-1}}{\partial \xi} \left(\zeta^2 - 1\right) \tag{3-54}$$

In order to find pressure, the overall principle of conservation of mass is applied to the fluid gap. In non-dimensional form, and in terms of expanded variables, this principle can be written as,

$$\begin{aligned}
&\int_0^1 \tilde{v}_{r0}\, d\zeta = \frac{1}{2}\xi \\
&\int_0^1 \tilde{v}_{rn}\, d\zeta = 0\ \ n \geq 1
\end{aligned} \tag{3-55}$$

Using equations (3-55) and (3-54), the lowest order pressure and *r*-component of velocity are found.

$$\begin{aligned} P_{-1} &= \frac{-3(1+BnN)}{4}\left(\xi^2-1\right) \\ \tilde{v}_{r0} &= \frac{-3}{4}\xi\left(\zeta^2-1\right) \end{aligned} \tag{3-56}$$

Substituting for $\tilde{v}_{r0}$ from equation (3-56) in equation (c) from (3-50) and using the boundary conditions outlined in (3-53), $\tilde{v}_{z1}$ is found.

$$\tilde{v}_{z1} = \frac{1}{2}\left(\zeta^3-3\zeta\right) \tag{3-57}$$

A similar procedure is undertaken to solve the differential equations of the $O\left(\varepsilon^1\right)$ order, shown in (3-51). After solving the equations in (3-51), other expanded velocity and pressure components are found to be:

$$\begin{aligned} &\tilde{v}_{r1} = 0 \\ &\tilde{v}_{z2} = 0 \\ &P_{-1} = \frac{-3}{4}(1+BnN)\left(\xi^2-1\right) \\ &P_0 = P_a \\ &P_1 = \frac{3}{2}(1+BnN)\left(\zeta^2-1\right)+\frac{9}{4}BnN^2\left(-\frac{1}{2}\xi\zeta^3+\xi\zeta-\frac{1}{2}\right) \end{aligned} \tag{3-58}$$

Now that all unknowns are found, they can be combined via equations (3-36), (3-37), and (3-48) to obtain the velocity field, shear rate distribution, pressure distribution, and squeeze force. It should be noted that, one could extend the solution to higher orders of ε and solve for higher order expanded velocity and pressure variables. In this study, the analyses were carried up to the order of ε^2 initially, and it was found that adding the higher orders does not have a significant effect on the overall solution. Hence, the solution is presented for the ε^1 order.

3.2.6 Velocity Field

The *r*- and *z*-component of velocity are obtained from inserting the solutions in equations (3-36) and (3-37).

$$\begin{aligned}\hat{v}_r &= \frac{v_r}{V} = \tilde{v}_{r0} + \varepsilon\tilde{v}_{r1} = \frac{-3}{4}\xi\left(\zeta^2 - 1\right)\\ \hat{v}_z &= \frac{v_z}{V} = \varepsilon\tilde{v}_{z1} + \varepsilon^2\tilde{v}_{z2} = \frac{1}{2}\left(\zeta^3 - 3\zeta\right)\varepsilon\end{aligned} \tag{3-59}$$

As can be seen, the *r*-component is of $O\left(\varepsilon^0\right)$ order and the *z*-component is of $O\left(\varepsilon^1\right)$. When the gap's aspect ratio is large, these two components are of the same order and hence, the velocity field consists of both *z*- and *r*- direction velocities. However, when $\varepsilon \ll 1$, the *z*-direction velocity is very small compared to *r*-direction velocity and the total velocity consists of only *r*-direction velocity. This is shown in Figure 3-10.

3.2.7 Shear Rate Distribution

Following equation (3-42), the shear rate is found as a function of coordinates and gap size.

$$\dot{\Gamma} = \frac{3}{2}\sqrt{\xi^2\zeta^2 + \left(\zeta^2 - 1\right)^2\varepsilon^2} \tag{3-60}$$

As mentioned earlier, the plug flow region is important in the design and analysis of MR devices. The plug flow region in a material with yield is defined as the region in the flow in which the fluid stress is less than its yield stress. In the plug flow region, shear rate is zero and the material behaves like a bulk solid. Plug flow regions must be either stationary or move as a rigid body. In the present analysis, the material model does not show an ideal pre-yield/post-yield behavior (shown in Figure 3-8). As a result, the plug flow region and yield surface of the material can only be detected approximately. However, this approximation is adequate for design and synthesis of MR devices that are the focus of this study.

Figure 3-10 shows contour plots of shear rate in the solution domain (upper right quadrant of the total flow domain from Figure 3-7). As is seen, the regions with low shear rates (possible plug flow regions) are around stagnation points at the centers of the pole plates (around $\xi = 0$, $\zeta = 1$). Lawrence and Corfield [107] stated that regardless of the flow geometry and other conditions it is likely that the region in the vicinity of a stagnation point at a no-slip boundary will be plug flow region.

3.2.8 Yield Surface Location

The results shown in Figure 5 qualitatively show the approximate location of plug region. In order to quantitatively determine the plug flow region, the Von Mises yield criterion can be used,

$$\sqrt{\frac{1}{2} II_{\underline{\underline{\tau}}}} = \tau_y \tag{3-61}$$

where, $II_{\underline{\underline{\tau}}}$ is the second invariant of the stress tensor. Equation (3-61) can be written in terms of stress tensor components as,

$$\sqrt{\frac{1}{2}\tau_{rr}^{\ 2} + \frac{1}{2}\tau_{\theta\theta}^{\ 2} + \frac{1}{2}\tau_{zz}^{\ 2} + \tau_{rz}^{\ 2}} = \tau_y \tag{3-62}$$

Transforming into non-dimensional variables, equation (3-62) becomes,

$$\sqrt{\frac{1}{2}T_{rr}^{\ 2} + \frac{1}{2}T_{\theta\theta}^{\ 2} + \frac{1}{2}T_{zz}^{\ 2} + T_{rz}^{\ 2}} = Bn \tag{3-63}$$

where, the viscosity, $\tilde{\eta}\left(\dot{\Gamma}\right)$, is written as equation (3-46). After some simplification, the yield surface equation is obtained,

$$Bn = \frac{3}{8}\left|-4\left(1 + BnN\right) + 3BnN^2\xi\zeta\right|\sqrt{\left(3\zeta^4 - 6\zeta^2 + 3\right)\varepsilon^2 + \xi^2\zeta^2} \tag{3-64}$$

The loci of points where equation (3-64) is satisfied construct the yield surface. Equation (3-64) is solved numerically using MATLAB. Figure 3-11 shows yield surface location as the gap size is decreased. As is

seen, at large gap sizes, the plug region is around the center of the upper plate but when gap size becomes smaller, the plug region changes shape and consists both ξ and ζ axes. This is in agreement with the lubrication assumption for the plug flow region. However, this plug flow shape seems to be kinematically inconsistent because it is radially stretching outwards and therefore cannot be un-yielded. This may be due to assuming a strict yield criterion in material models and has been shown in [107]. The strict yield criterion is an approximation of the actual material behavior and it can be argued that there is no material with a true yield stress. Additionally, the lubrication assumption is also an approximation that ignores the axial stress components ($\tau_{rz} = \tau_{zz} = 0$) and solves the *r*-component of the equation of motion. These two approximations are not fully consistent. In this paper, the constitutive equation used does not allow MR fluid to show an actual pre-yield/post-yield behavior.

3.2.9 Pressure Distribution

Using the solutions for expanded pressure components in equations (3-56) and (3-58), the total pressure becomes

$$P = \frac{-3}{4}(1+BnN)\left(\xi^2-1\right)\varepsilon^{-1} + P_a + \left[\frac{3}{2}(1+BnN)\left(\zeta^2-1\right) + \frac{9}{4}BnN^2\left(-\frac{1}{2}\xi\zeta^3 + \xi\zeta - \frac{1}{2}\right)\right]\varepsilon \qquad (3\text{-}65)$$

At relatively large gap sizes, the $O\left(\varepsilon^1\right)$ term in equation (3-65) is significant and contributes to the total pressure in the fluid. Therefore, the total pressure is a function of both radial and axial coordinates. As the gap size is decreased, the $O\left(\varepsilon^{-1}\right)$ rapidly increases in magnitude and becomes the dominant factor in total pressure. As seen in equation (3-65), this term only is a function of the radial coordinate. As a result, the total pressure becomes a function of ξ only. Hence, the solution reduces to the lubrication approximation and assumes the pressure is only a function of the radial coordinate for small gap sizes. Contour plots of pressure are shown in Figure 3-10 for various values of ε.

3.2.10 Total Squeezing Force

The total force is obtained by integrating total stress on the surface of the plates.

$$\hat{F} = \frac{F}{\pi R^2 \eta_0 V / h} = 2\int_0^1 \left(P - P_a + T_{zz} \right)_{\zeta=1} \xi d\xi \tag{3-66}$$

After substituting the solutions obtained from equations (3-65) and (3-35) into equation (3-66) and some simplification, the total force can be written in non-dimensional form as,

$$\hat{F} = \frac{3}{8}\left(1 + BnN\right)\varepsilon^{-1} - \frac{3}{8} BnN^2 \varepsilon \tag{3-67}$$

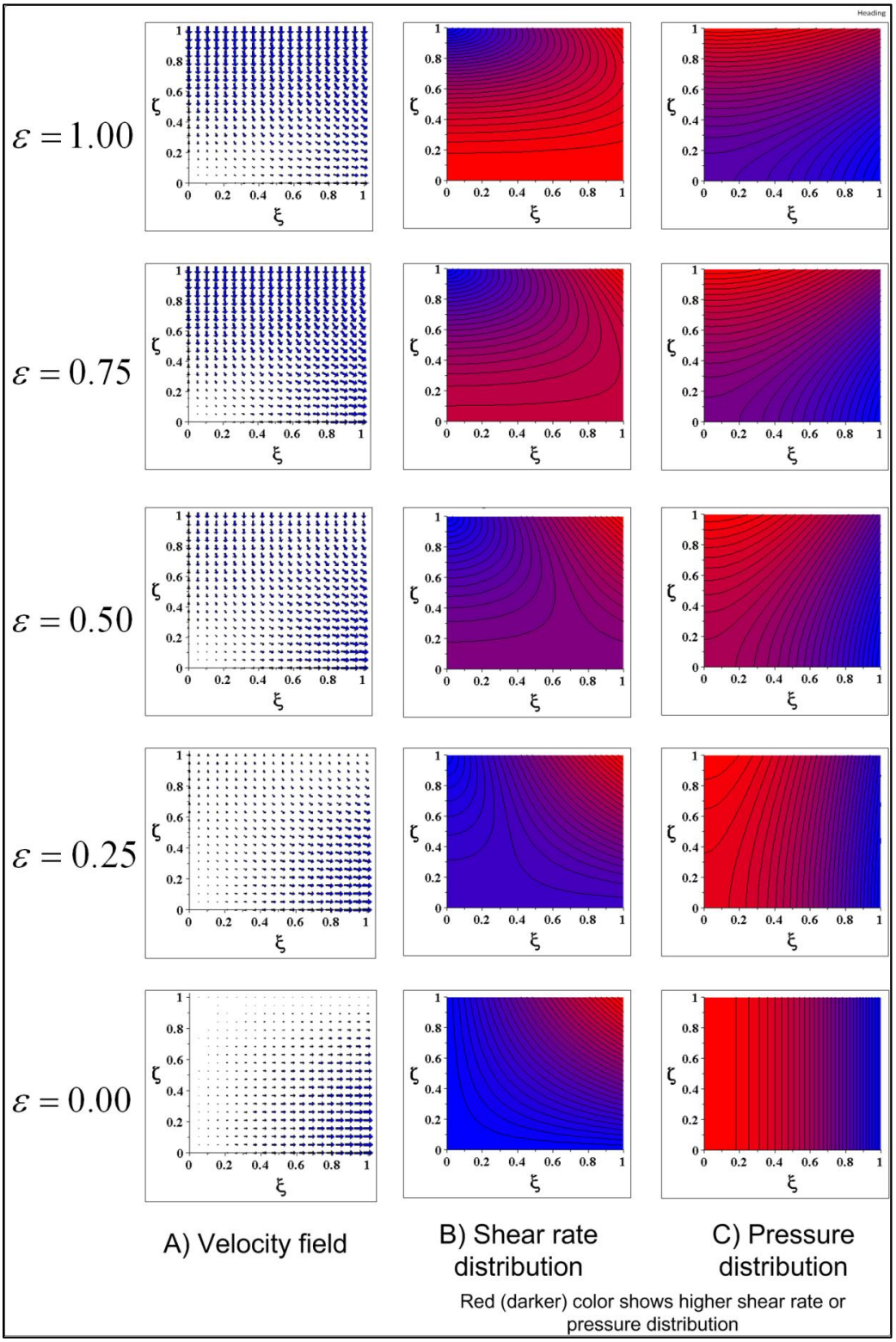

Figure 3-10. Model results for velocity field, shear rate distribution, and pressure distribution

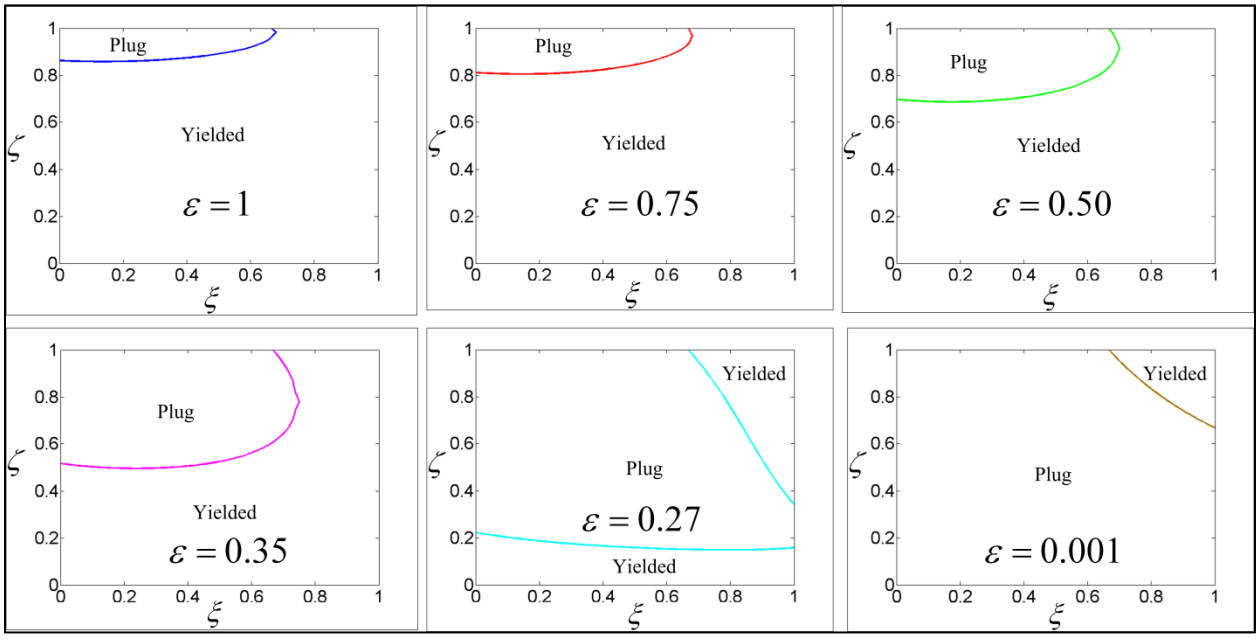

Figure 3-11 Yield surface location as gap size is decreased

3.2.11 Mathematical Model Verification

In this section, it is shown that when the Papanastasiou's exponent, *n*, approaches infinity, the developed model reduces to the Bingham solution. It is also shown that at higher shear rates when the viscosity is considered to be constant, the model results in the lubrication approximation for a Newtonian fluid.

As discussed in section 3.2.3, the linear approximation of equation (3-44) is valid if $\dot{\Gamma} \le \dot{\Gamma}^*$. Using equation (3-45), the range in which the linear approximation is valid can be obtained.

$$N\dot{\Gamma}^* = 2 \tag{3-68}$$

From equation (3-60), the maximum value of shear rate occurs as $(\xi,\zeta)=(1,1)$ and is $\dot{\Gamma}_{\max} = 3/2$. Using this result and equation (3-68), the maximum value of N is $4/3$.If $N > 4/3$, the viscosity can be considered constant and the system of perturbation equations can be solved using this assumption. In the case of small shear rates ($\dot{\Gamma} < \dot{\Gamma}^*$), it can be seen that

$$N = \frac{-n\dot{h}R}{h^2} < \frac{4}{3} \Rightarrow n < \frac{-4h^2}{3R\dot{h}} \tag{3-69}$$

Equation (3-69) shows the limiting value of the Papanastasiou's exponent, *n*, for the linear approximation of equation (3-44). In order for this approximation to be valid as *n* approaches infinity, the value of the plate velocity, $\dot{h}$, has to decrease. Therefore, if *n* approaches infinity (reducing to the ideal Bingham model), the plate velocity should approach zero; consistent with the implications of the quasi-static assumption made earlier in the paper.

The force equation of (3-67) can be rewritten as

$$F = \frac{\pi\tau_y R^3}{2h}\left(\frac{3}{4}N - \frac{3}{4}N^2\varepsilon^2\right) - \frac{3}{8}\frac{\pi R^4\eta_0\dot{h}}{h^3} \tag{3-70}$$

At the limiting case when $N = 4/3$, equation (3-70) reduces to

$$F = \frac{\pi\tau_y R^3}{2h}\left(1 - \frac{4}{3}\varepsilon^2\right) - \frac{3}{8}\frac{\pi R^4\eta_0\dot{h}}{h^3} \tag{3-71}$$

This equation reduces to the Bingham lubrication solution for small gap sizes [99, 124]. The $O\left(\varepsilon^2\right)$ term becomes negligible as the gap aspect ratio is reduced. In case of $\dot{\Gamma} > \dot{\Gamma}^*$, if shear thinning effects can be neglected, the fluid is considered as a Newtonian fluid with a constant viscosity as shown in Figure 3-9. In this case, *A*, *B*, and *C* in equation (3-47) will become

$$A = 1,\ B = C = 0 \tag{3-72}$$

Using equation (3-72) along with perturbation equations (3-49)-(3-51) and the appropriate boundary conditions in (3-53), the squeeze flow problem can be solved and gives

$$\begin{aligned} &P_{-1} = \frac{-3}{4}\left(\xi^2 - 1\right),\quad P_0 = P_a,\quad P_1 = \frac{3}{2}\left(\zeta^2 - 1\right) \\ &\tilde{v}_{r0} = \frac{-3}{4}\xi\left(\zeta^2 - 1\right),\quad \tilde{v}_{r1} = 0,\quad \tilde{v}_{r2} = 0 \\ &\tilde{v}_{z1} = \frac{1}{2}\left(\zeta^3 - 3\zeta\right),\quad v_{z2} = 0,\quad v_{z3} = 0 \end{aligned} \tag{3-73}$$

Using the force equation (3-66) and substituting for the appropriate parameters from equation (3-73), the total squeezing force for this case is obtained as

$$\hat{F} = \frac{F}{\pi R^2 \eta_0 V / h} = \frac{3}{8}\varepsilon^{-1} \tag{3-74}$$

And can be rewritten to result

$$F = \frac{-3}{8}\frac{\pi R^2 \eta_0 \dot{h}}{h^3} \tag{3-75}$$

which is in agreement with solutions obtained for a Newtonian fluid using lubrication theory [99, 124].

Model validation for this model will be presented in later chapters after experimental setup and test procedures are introduced.

Chapter 4
Rheology of MR Fluids in Squeezing Flows

Chapter 4 is devoted to the experimental studies performed on rheological behavior of MR fluids in the squeeze mode. In this chapter, the squeeze mode rheometer designed and built at CVeSS is introduced. The rheometer is used for testing MR fluids in the squeeze mode under various conditions. The same rheometer was also used for MR pouch testing, which will be discussed in later chapters. Test setup and test procedures are discussed in this chapter and at the end, test results and observations are presented. Finally, mathematical model validation is presented using experimental data obtained from the rheometer.

4.1. Design Requirements

In designing a squeeze mode rheometer, different important issues should be considered, including rheometer actuation, MR fluid flow path, magnetic field path, and data acquisition. For the squeeze mode, the relative movement of the poles is parallel to the magnetic field. This mode requires linear actuators to move the poles toward each other. Pneumatic, hydraulic, or electric actuators can be used for actuation.

As the poles move closer to each other, the volume between the poles is decreased. As a result, the excess MR fluid has to flow outside. This needs to be considered when designing the squeeze mode rheometer and appropriate flow paths should be incorporated in the rheometer.

The magnetic circuit design is one of the most important parts of the rheometer design. The magnetic circuit has to pass through the MR fluid and at the same time, it has to be maximum in the gap. The

dimensions and material properties of each part of the rheometer have to be carefully selected to guide the magnetic circuit as desired.

Data acquisition consists of collecting all the needed data from the rheometer test. The required measurements for the squeeze mode rheometer include force, displacement (gap size), electric current supplied to energize MR fluid and, magnetic field strength.

Based on the above design criteria, a squeeze mode rheometer was designed and built. The schematics of the rheometer and different parts are shown in Figure 4-1 and Figure 4-2.

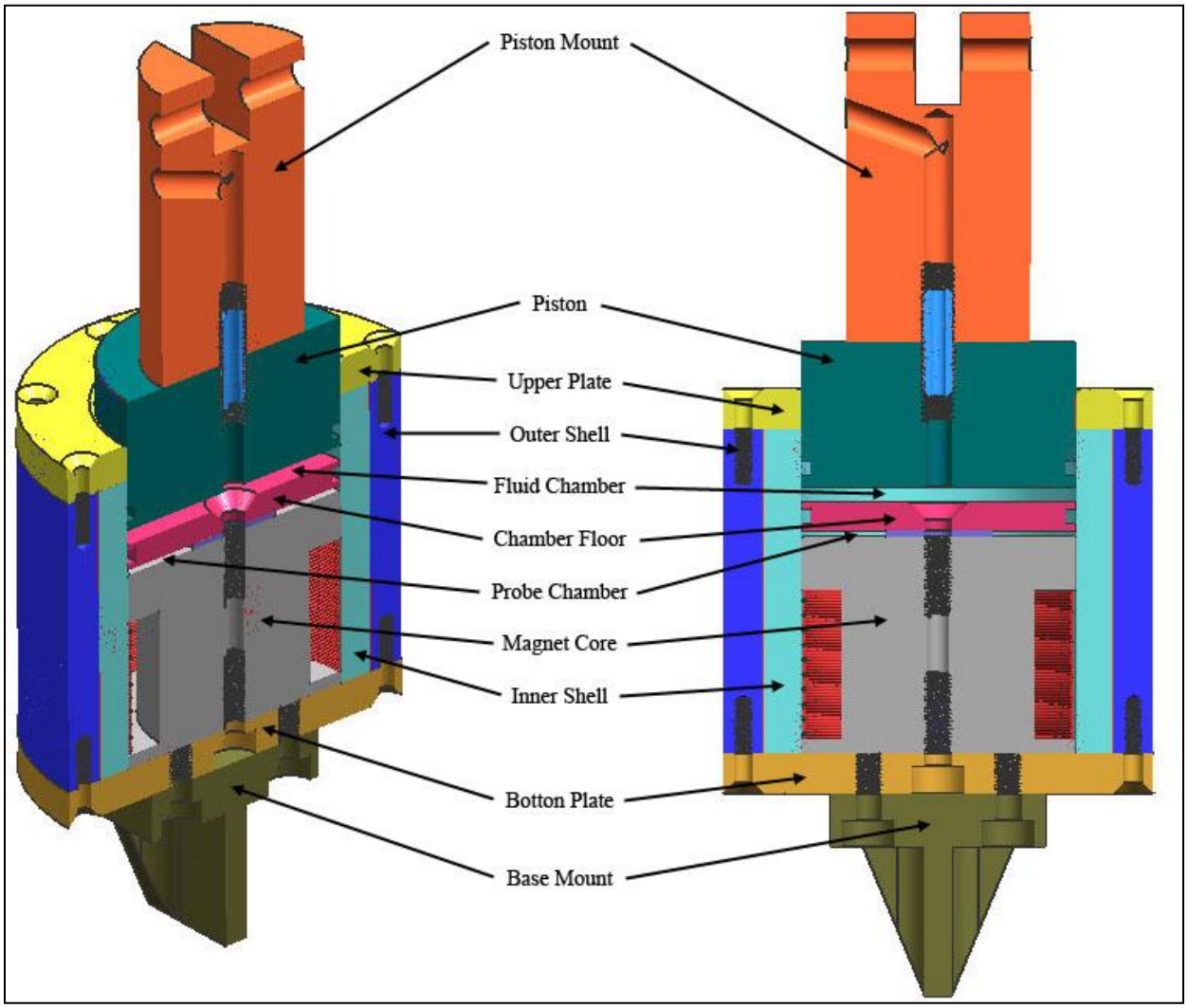

Figure 4-1 A cross-sectional view of the rheometer with major components labeled

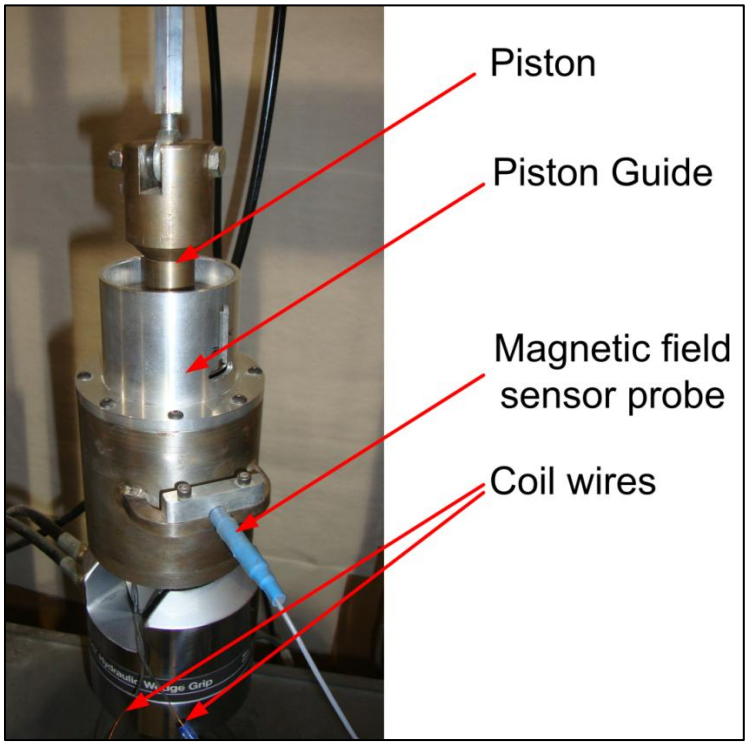

Figure 4-2 The squeeze mode rheometer mounted in the MTS load frame (the aluminum piston guide is not shown in the cross-section view of Figure 4-1)

In the assembly shown, only the piston and piston mount are able to move relative to the rest of the rheometer. The MR fluid test chamber is located directly below the piston, which means that the volume of this chamber, and thus the MR fluid within, is dependent on the position of the piston. A steel disk forms the floor of the test chamber. Both the chamber floor disk and piston are sealed to the aluminum inner shell by way of Buna-N O-rings. An air gap of approximately 0.050 in., which is created by a stainless steel washer, is located between the chamber floor and the top of the magnet core. This air gap provides a location for a Gauss-meter probe that is used to verify the magnetic flux generated by the device. The electromagnet, which uses a 12L14 steel core, is located below the Gauss probe gap and is bolted to the bottom plate. The bottom plate joins the base mount and the core to the outer shell. The steel outer shell provides the return path for the magnetic flux. The aluminum inner shell directs the magnetic flux path through the MR fluid test chamber and also provides a smooth sealing surface for the piston and chamber floor.

4.2. Rheometer Actuation and Installation

The rheometer actuation is done by a hydraulic MTS load frame available in CVeSS lab facilities. The MTS load frame is an 810 series Material Testing System mated with a 5.5 kip load cell and a hydraulic actuator capable of producing a 10.5kN force with a dynamic stroke of ±2.5 in. (Figure 4-3). The actuator is powered by an MTS Model 505.20 SilentFlo Hydraulic Power Unit. This hydraulic pump, shown in Figure 4-4, has a flow rate of 20 gal/min and an operating pressure of 3000 psi.

Figure 4-3 MTS 810 series Material Testing System

Figure 4-4 MTS SilentFlo hydraulic power unit

4.3. MR Fluid Flow Path

The rheometer developed in this research is intended for use predominantly in quasi-static-type testing. This makes the fluid management aspect of the design less of a concern when compared to a rheometer that operates over a wide range of frequencies or velocities. With the intended use in mind, a simple approach is taken in regards to managing excess MR fluid. MR fluid is allowed to escape from the test chamber via a single hole drilled through the center of the circular piston that squeezes the fluid. As shown in Figure 4-5, the passage continues into the center of piston mount where it is plumbed, at an angle, out of the rheometer. A small reservoir is attached to the exit of this passage via ¼" clear vinyl tubing. Under operation, the reservoir and tubing are filled with a sufficient volume of fluid as to prevent air from being introduced into the test chamber.

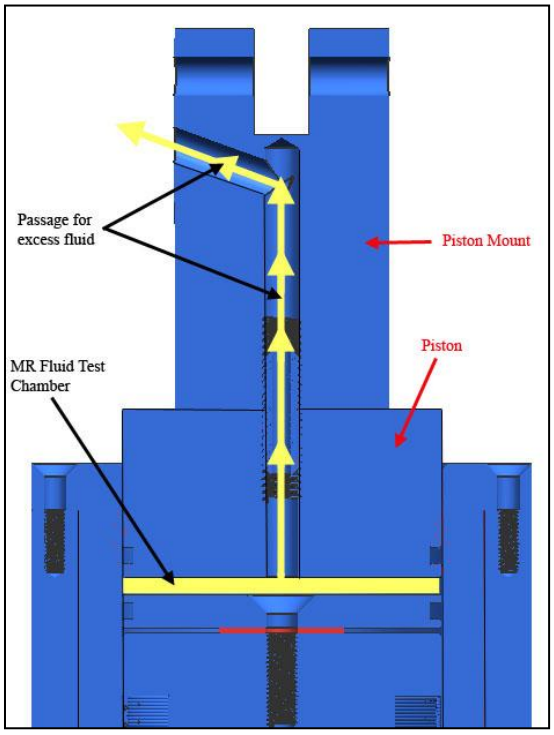

Figure 4-5 MR fluid is allowed in and out of the test chamber through a passage in the piston and piston mount

4.4. Magnetic Field

As discussed, one of the areas critical to the design of the rheometer is the management of the magnetic flux path. Much of the design for this rheometer focuses on the magnetic circuit. The goal for this design is to be able to generate a sufficient level of magnetic flux to saturate the test fluid with a current supply of 4 amps. In order to maximize the magnetic flux with a fixed current supply, the number of necessary coil turns must be calculated, and the magnetic reluctance of the magnetic circuit must be minimized. The rheometer has a coil made with 22AWG wire with approximately 640 turns. As shown in Figure 4-6, the magnetic circuit is formed placing low reluctance materials in strategic locations to form a complete loop for the magnetic flux to follow. The low reluctance materials in these areas are a mix of 1018 mild steel and 12L14 steel, both of which exhibit good magnetic permeability properties and machining characteristics. A high reluctance aluminum shell surrounding the coil magnet is positioned along the length of the rheometer to discourage the magnetic flux path from deviating away from the MR fluid chamber. In this configuration, nearly all of the magnetic flux flows through the MR fluid.

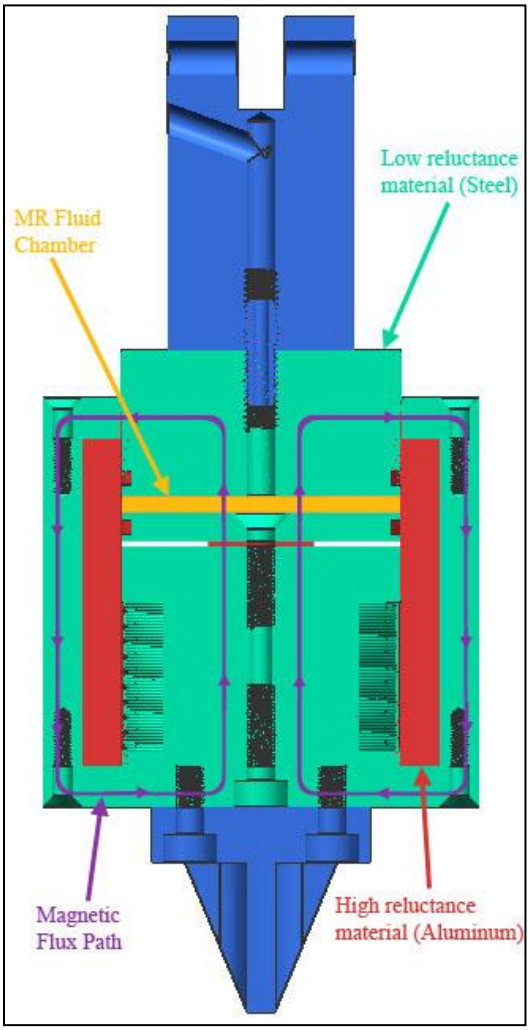

Figure 4-6 The path of magnetic flux is controlled using magnetic conductors and insulators

Air has a fairly large reluctance value relative to steel; hence, it is generally not desirable in the magnetic flux loop. There are two points in the design where an air gap has the opportunity of being introduced. The first is in the interface between the sliding piston and the upper plate. The piston must have enough clearance to slide freely, but a large gap reduces the magnetic flux generated through the circuit. The piston was sized such that its diameter is only 0.001 in. smaller than the inner bore of the upper plate and aluminum shell. With this design, the reluctance contribution from the air gap is assumed negligible. The second area where an air gap is introduced into the magnetic flux path is in the space allocated for the Gauss-meter probe. This air gap is necessary for the probe, but the size of the gap was reduced as much as

possible while still allowing enough space for the probe to be easily inserted and positioned. The air gap for the probe is set at 0.050 in. by way of a stainless steel washer that sits between the chamber floor and the coil magnet.

More details about the design of the squeeze mode rheometer can be found in [125].

4.4.1 Magnetic Field Simulation

A software package called Finite Element Method Magnetics (FEMM) was used throughout the design process of the rheometer. With this software, the magnetic circuit of the rheometer can be quickly analyzed allowing for multiple design iterations prior to component manufacturing. The program is a simple finite element code capable of solving 2-D and axisymmetrical magnetic problems. The magnetic circuit portion of the rheometer design is axisymmetric in nature, thus a more complex 3-D finite element solver is not required. As shown in Figure 4-7, the rheometer design is entered as a series of lines and areas. Each area is assigned a magnetic material property. The leftmost edge of each drawing in Figure 4-7 represents the axis of symmetry. The software assumes that each area drawn to the right of this line is uniformly revolved around the axis of symmetry, thus creating a cylindrical volume. The right-hand picture of Figure 4-7 is color coded by material definitions used in FEMM. The shades of green represent various ferrous materials that readily pass magnetic flux, while the shades of red represent non-ferrous materials that significantly inhibit the magnetic flux path. The area of blue represents the coil magnet.

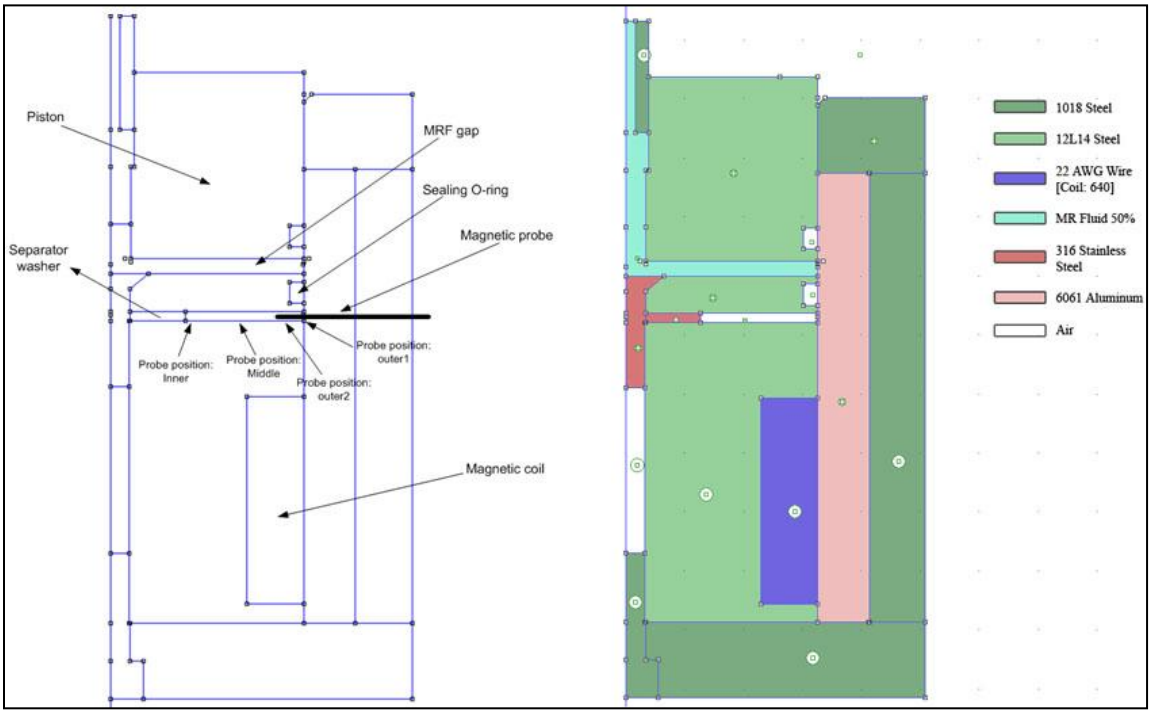

Figure 4-7 The rheometer design as entered in the FEMM program

The leads from the coil magnet are connected to a DC power supply. The coil magnet is powered by a Protek DC Power Supply model 30058, which has a choice of either voltage or current control. For the rheometer testing the current control mode was used.

One feature of the software useful to this research is the analysis of the magnetic flux path and flux density. A contour plot of the magnetic flux path and flux density is shown in Figure 4-8. The flux path is marked by the contour lines, which form a loop through the structure. This plot provides a clear visual of how the ferrous material directs the path of the magnetic flux. The colors in this plot represent the flux density distribution within the rheometer.

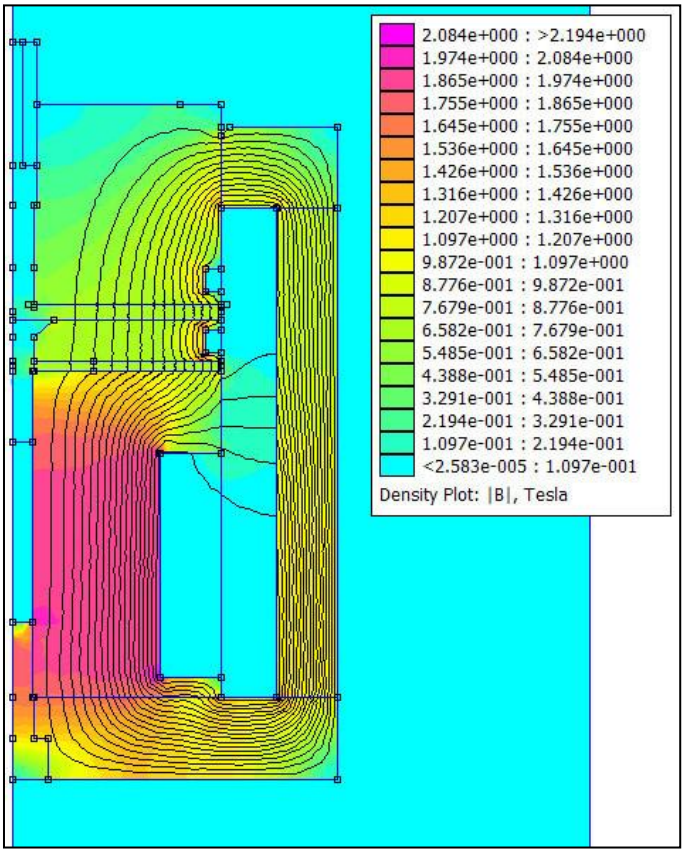

Figure 4-8 FEMM results with 3A of current supplied to the coil magnet

In order to assure the reliability of the FEMM simulations, mesh convergence studies are performed. Results show that by increasing the number of elements, the Finite Element solutions converge. A typical mesh convergence study is shown in Figure 4-9 for the squeeze more rheometer at a gap size of 0.1 in. and electric current of 1 Amp with an MRF of 20% volumetric iron particles in the rheometer.

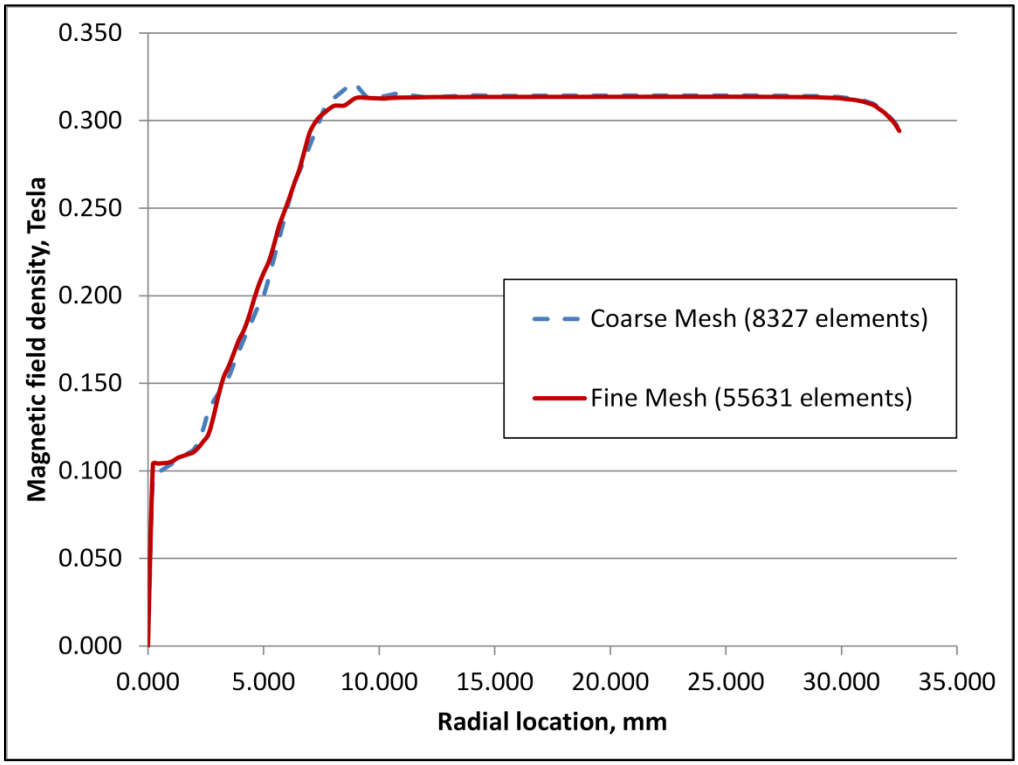

Figure 4-9 A mesh convergence study for the rheometer magnetic analysis. Gap size is 0.1 in. with 1 Amp electric current. An MR fluid with a 20% volumetric particles is in the rheometer gap.

The areas of greatest interest are the Gauss-meter probe gap located above the magnet core and the MR fluid test chamber located below the piston. It is desirable to have a uniform flux density across the both of these areas. A comparison plot of the flux density distribution in these two areas is presented in Figure 4-10. This plot shows the flux density as function of the radius in the test chamber of the rheometer. It is clear from this plot that the flux density is constant and uniform through the majority of the radius. Also, the flux density in the gauss probe gap closely follows the density found in the MR fluid test chamber. This model indicates that flux density measurements taken in the air gap below the test chamber should closely follow the conditions experienced through the MR fluid.

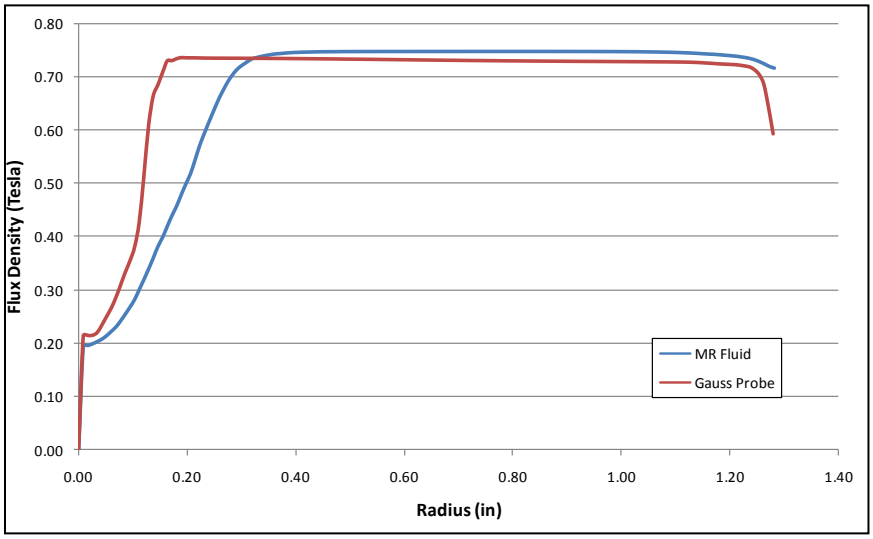

Figure 4-10 The flux density across both the MR fluid chamber and the Gauss-meter probe area for a current supply of 3 amps

The magnetic flux density measurements are done using a magnetic probe. The probe is an "ultra-thin" (0.2 mm x 1.5 mm) transverse probe from F.W. Bell. This probe is connected to an F.W. Bell 5080 Gauss-meter as shown in Figure 4-11. Figure 4-12 shows the rheometer and the magnetic probe positioned in the probe gap at a pre-determined radial location.

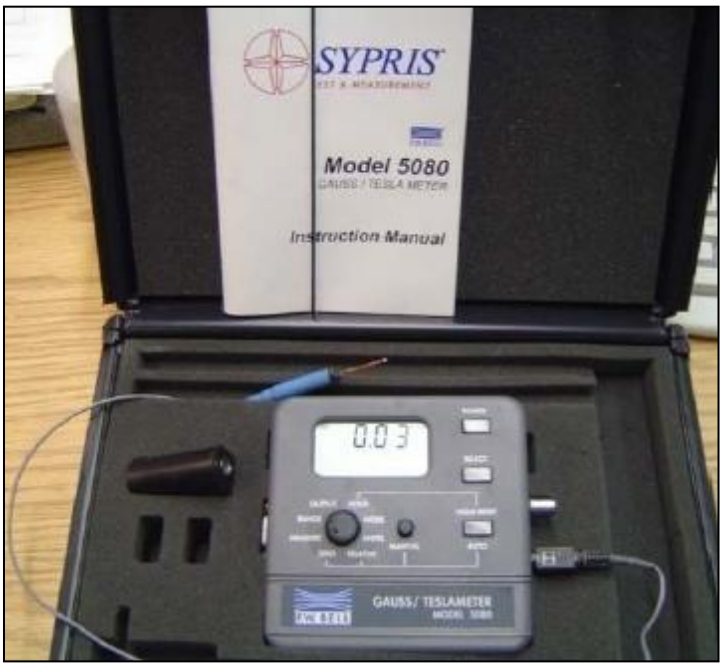

Figure 4-11 Gauss-meter and "ultra-thin" transverse probe

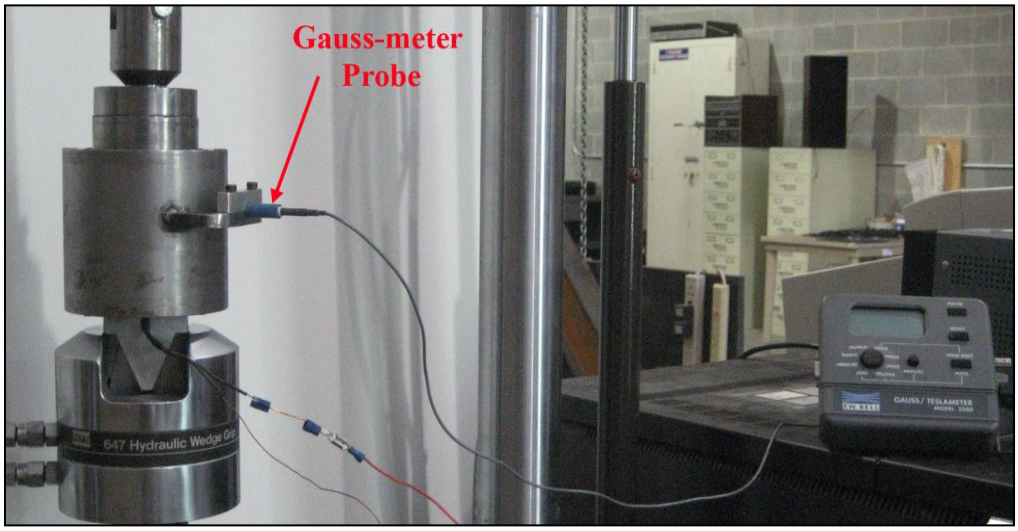

Figure 4-12 The Gauss-meter probe installed in the rheometer

4.4.2 Variations of the Magnetic Field with Gap Size

As the gap size changes, the reluctance of the magnetic field varies. This results in a change in the magnetic flux, B, as the rheometer piston moves. The amount of this variation should be considered in calculations. FEMM analysis shows that the magnetic field changes 41% as the gap size decreases from 0.2 in. to 0.05 in. at 1 Amp. Figure 4-13 shows the magnetic field distribution for a 0.2 in. gap at 1 Amp. A MR fluid with a 20% iron particles was used for the FEMM analysis. As can be seen in Figure 4-13, the magnetic field strength is 0.2559 Tesla in the MR cavity of the rheometer (note that the magnetic field is uniform along the radius as is shown in Figure 4-13 right).

As the piston moves down, the reluctance is decreased and as a result, the magnetic flied strength is increased. Figure 4-14 shows the magnetic field when the gap is 0.1 in. at 1 Amp. The magnetic field is increased to 0.3134 Tesla. At a gap of 0.05 in., the magnetic field is 0.3627 Tesla as shown in Figure 4-15.

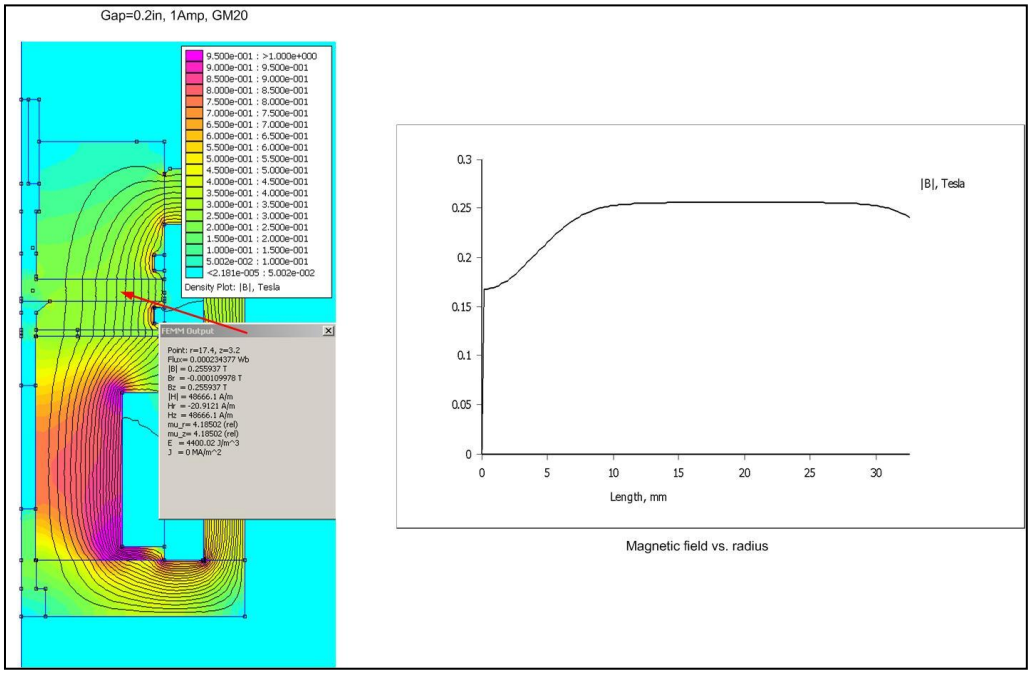

Figure 4-13 Magnetic flux B for a gap size of 0.2 in. at 1 Amp (with a 20% MR fluid in the rheometer)

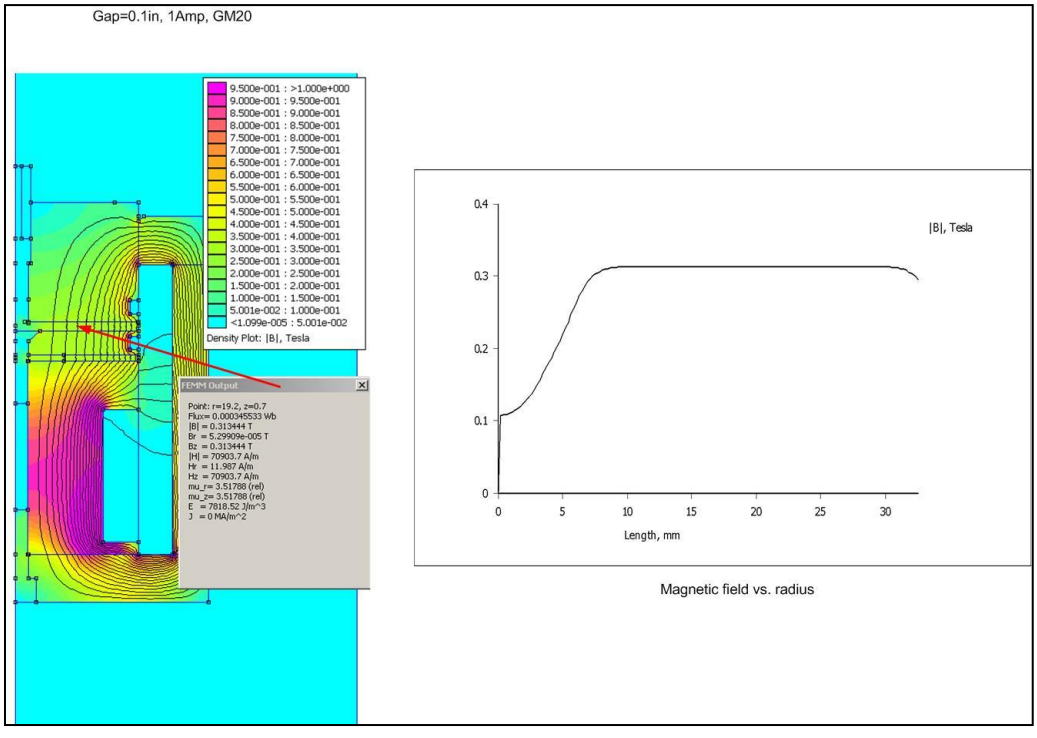

Figure 4-14 Magnetic flux B for a gap size of 0.1 in. at 1 Amp

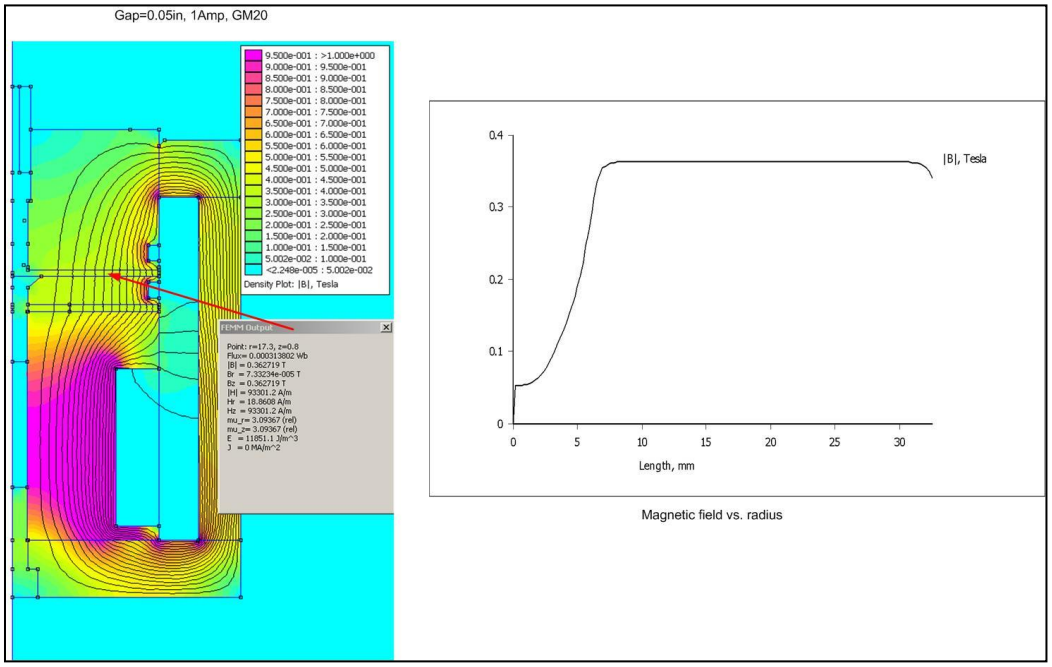

Figure 4-15 Magnetic flux B for a gap size of 0.05 in. at 1 Amp

The variations of the magnetic field as a function of the gap size can be approximated by a linear relationship. Figure 4-16 shows a comparison between the FEMM simulations and the test results at 1 Amp. The variation of the magnetic field versus the gap size is linear, and will be explained further in Section 4.6.

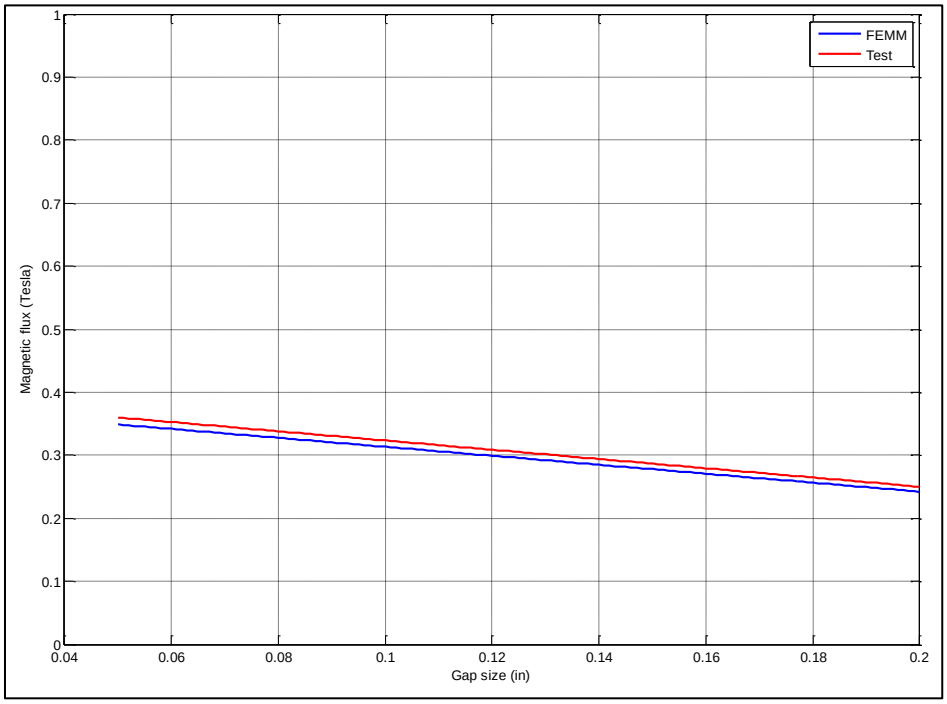

Figure 4-16 Comparison between the FEMM and the test results at 1 Amp

4.5. Data Acquisition

The actuator is controlled via the MTS model 458.20 analog controller pictured in Figure 4-17. This controller is capable of producing a variety of signals including square, ramp, and sinusoidal waves of variable amplitudes and frequencies. Displacement and force measurements are passed from the controller to the PC-based dSPACE data acquisition system. The dSPACE system is used in conjunction with Matlab/Simulink and Control Desk software to create user definable layouts to collect data from various channels. Along with the displacement and force channel, an additional channel is allocated for the Gaussmeter probe measurement.

Figure 4-17 MTS 458 analog hydraulic controller

With the rheometer clamped into the load frame, the MTS 458 controlling the actuator, the power supply connected to the coil leads, and the dSPACE data acquisition system recording force, displacement, and magnetic flux density, the rheometer is ready for operation, as seen in Figure 4-18. Testing procedure and test results are discussed in the next chapter.

Figure 4-18 The rheometer installed in the load frame ready for testing

4.6. Squeeze Mode Rheometer Test Setup

4.6.1 Testing Parameters

There are many different parameters engaged in testing MR fluid in squeeze mode. These parameters can be categorized into setup parameters, output parameters, and signal processing parameters. In this section, these parameters are explained briefly. Figure 4-19 shows a chart in which the testing parameters are schematically illustrated.

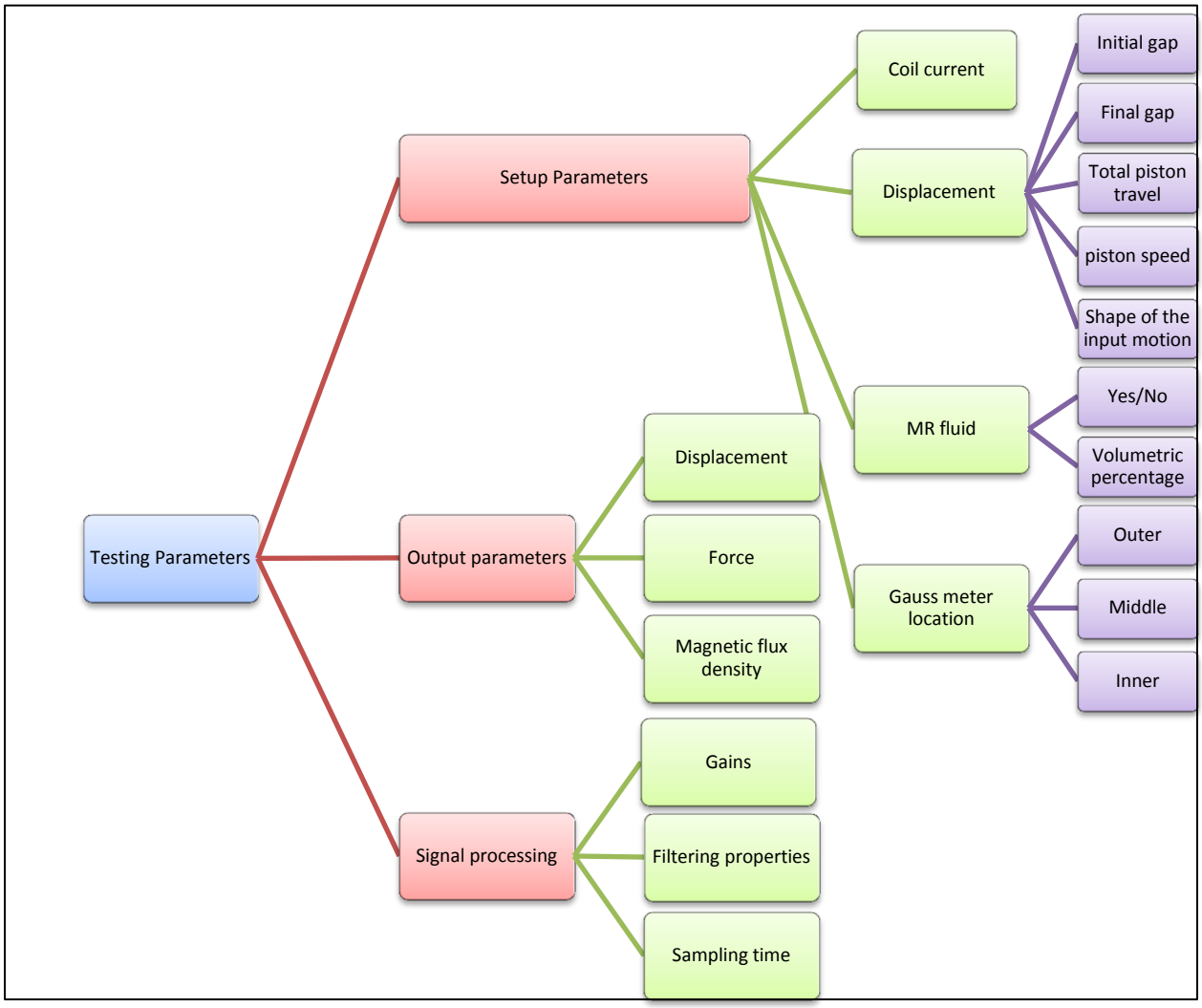

Figure 4-19 Testing parameters for Squeeze mode rheometer

4.6.2 Setup Parameters

These parameters are Displacement parameters, MR fluid parameters, and the Gauss meter parameters. These parameters are determined before the test, based on the desired test specification.

The power supply to the rheometer is a DC power supply which can be used both in voltage control and current control modes. Current control mode is used for the rheometer tests assuring a constant current and hence, magnetic field density during each test. The value of the current is one of the setup parameters.

The displacement parameters consist of initial gap, final gap, total piston travel, piston speed (frequency), and the shape of the input displacement signal (sine, ramp, etc.). Tests have been carried out for different initial gaps and final gaps, as well as piston speeds. Most of the tests have been carried out using a ramp displacement input. The ramp input has a constant velocity and is suitable for quasi-static testing.

The presence of the MR fluid and the type of MR fluid are also two parameters that need to be determined before testing. Some of the tests are done without any MR fluid in the rheometer chamber, namely, the friction tests, the magnetic validation tests, and the magnetic pull force tests. Different types of MR fluids are tested. These fluids are different in volumetric percentage of magnetic particles and the type of additives.

As discussed in previous sections, the gauss meter probe is inserted into a small air gap below the MR chamber separated from the MR fluid chamber. The gauss meter probe can move along the radius of the rheometer to measure the change of the magnetic field strength along the radius. It can be close to the center (inner position), close to the edge (outer position), or in between (middle position), as is shown in Figure 4-20 and Figure 4-21.

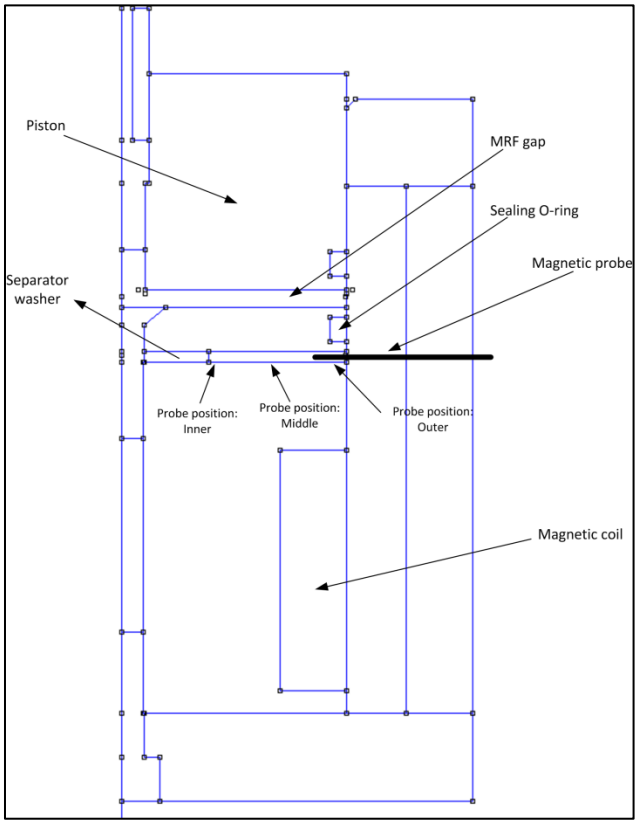

Figure 4-20 A cross section of the rheometer body showing three different positions of the gauss meter probe

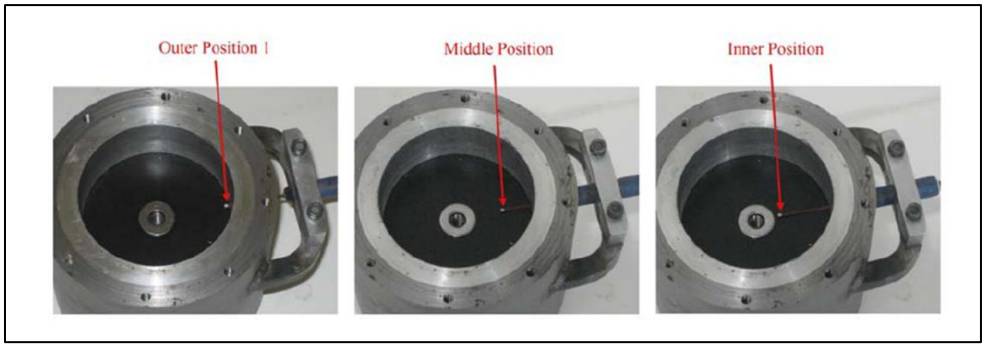

Figure 4-21 Magnetic probe locations in the rheometer probe chamber

4.6.3 Test Procedure

After the rheometer is mounted in the MTS load frame, as explained in the previous chapter, the rheometer is filled with a sufficient quantity of well shaken MR fluid. The MR fluid needs to be shaken before use. Although MR fluids usually have some additives and surfactants to prevent or delay settling, the iron particles settle after a certain time depending on the type of additives and surfactants.

After the rheometer is filled with MR fluid, the piston is moved and inserted into the MR cavity and the desired initial gap size is ensured in the rheometer. This is done by using calipers as shown in Figure 4-22. Then the span (total piston movement) and the frequency of the motion are adjusted on the MTS controller. After that, the control unit is activated and the piston starts to move down and squeeze the MR fluid. After the minimum gap size is reached, the piston moves back to the initial position with the same speed.

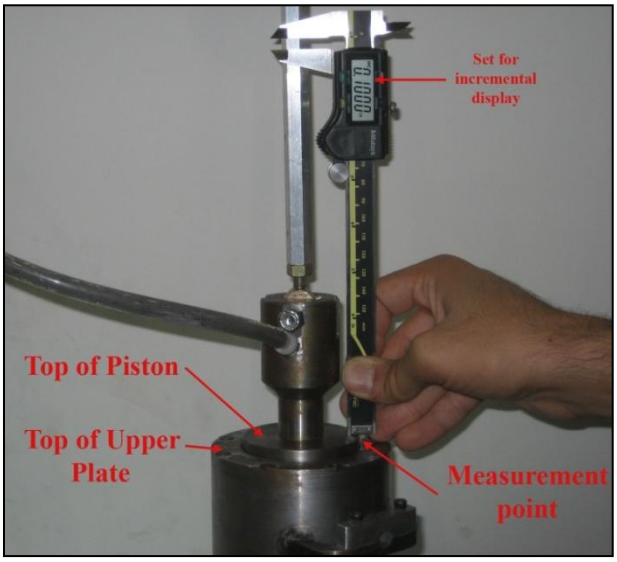

Figure 4-22 Setting the desired initial gap

It should be noted that, because of the configuration of the MTS load frame, the lower jaw moves and the upper jaw is stationary. So, in this test setup, the rheometer body moves upward.

4.6.4 Output Parameters

Output parameters are displacement, force, and magnetic field strength. Displacement and force are recorded by the LVDT transducer and load sensor mounted in the MTS load frame. The magnetic field strength is recorded by the Gauss-meter. As mentioned before, a dSPACE data acquisition system is used to record data. A screen shot of the data acquisition setup is shown in Figure 4-23.

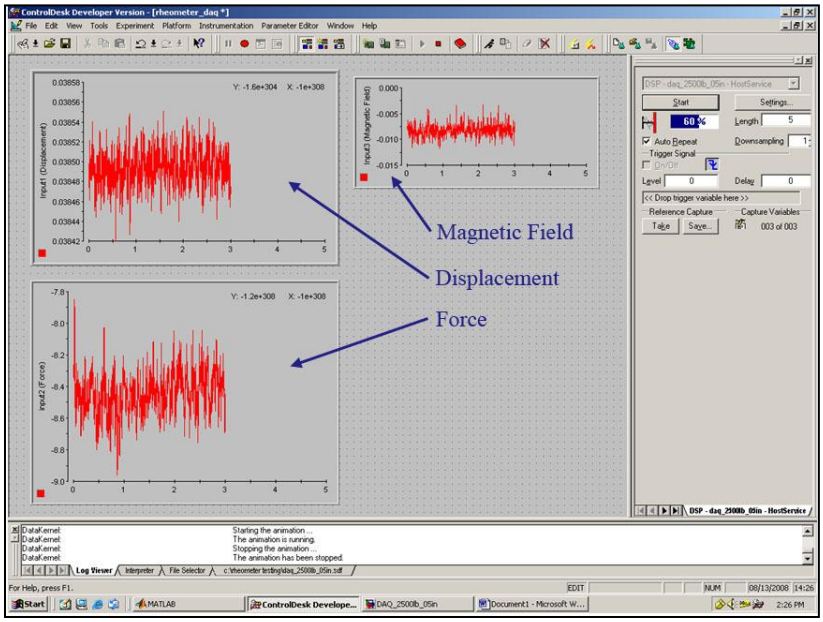

Figure 4-23 Screen shot of DSPACE layout

4.6.5 Signal Processing Parameters

The parameters involved in signal processing are gains, filtering, and sampling time (frequency). The gains are chosen based on the range of displacements and forces involved in tests. Filtering is done by using a simple first-order unity-gain low-pass discrete time filter with the following specifications:

$$H(z) = \frac{0.095z}{z - 0.905} \tag{4-1}$$

The sampling time was 0.001 seconds (a sampling rate of 1000Hz).

4.6.6 Magnetic Analysis and Validation of the Rheometer

Before starting the tests, the rheometer was checked to make sure that the magnetic circuit was working as expected. The magnetic field strength was measured using the Gauss-meter probe in three designated locations (Figure 4-20) and the results were compared to the FEMM magnetic analysis simulations.

At each location of the gauss meter probe the current was increased from zero to its maximum (4Amps) and then reduced to zero to obtain the hysteresis associated with the magnetic field strength. These tests

were done with a rheometer gap size of 0.1 in. without any MR fluid in the rheometer. The magnitude of the electric current was changed by hand.

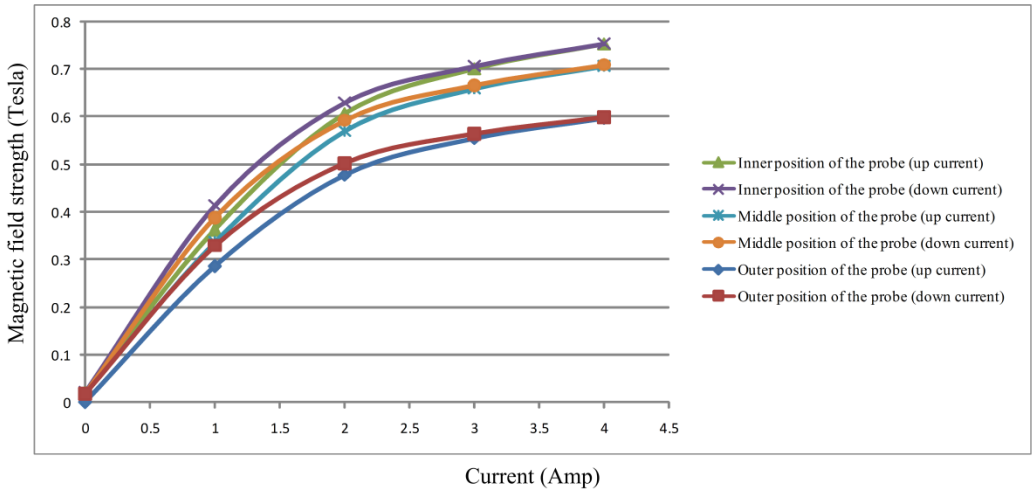

Figure 4-24 Magnetic field strength vs. current at different positions of the gauss meter probe (at a rheometer gap size of 0.1 in. and no MR fluid)

Figure 4-24 shows that there is a small hysteretic effect. It also shows that the maximum value of magnetic field strength occurs at the inner probe position. The outer position has the minimum magnetic field strength. This is possibly because of presence of an O-ring groove above the probe cavity at the outer location. See Figure 4-20.

Comparison between the FEMM analysis and the test is shown in Figure 4-25. As can be seen, the FEMM results agree with the measurements from the magnetic probe. This validates the accuracy of the finite element model and verifies use of the probe readings. Although the readings agree with the FEMM results, there is a slight gradient seen in the probe readings which is not captured in the FEMM results. It may be sue to small alignment and position errors in placing the magnetic probe at the inner location which can lead to the probe touching the separator washer, as shown in Figure 4-21.

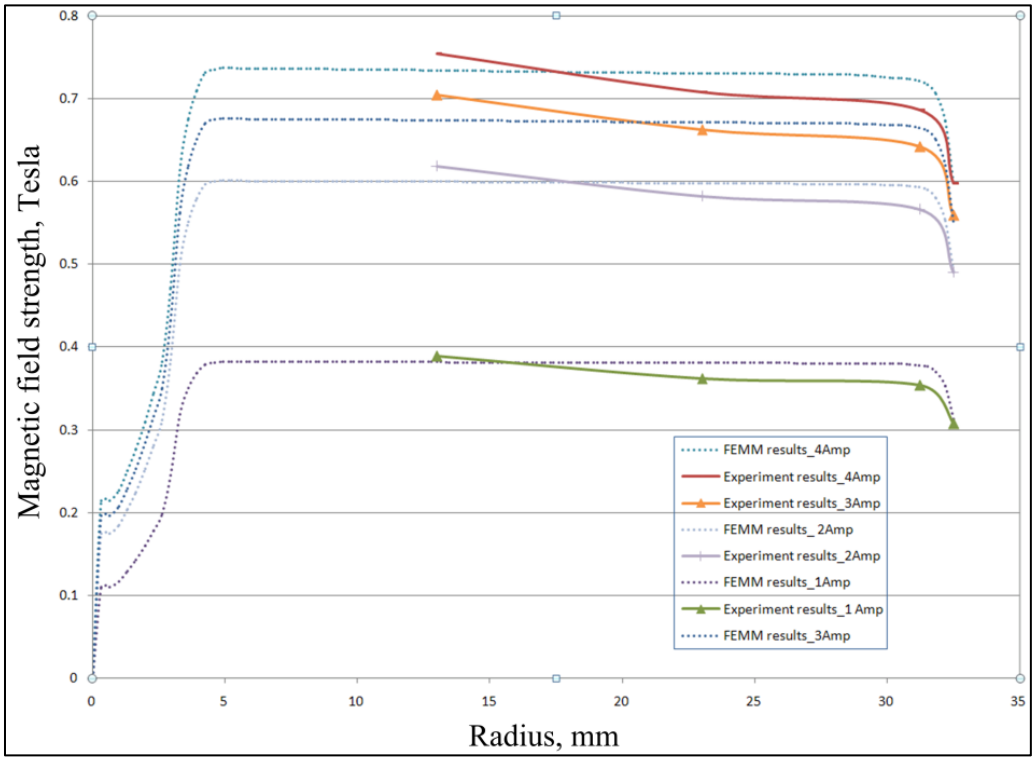

Figure 4-25 Magnetic field strength vs. radius for different currents (0.1in. gap size)

4.7. MR Fluid Squeeze Test Results

In this section, the main test results are presented and discussed. A variety of MR fluids are tested under different conditions. There are mainly two types of MR fluids tested. One type is the MR fluids received from an industrial sponsor. This fluid is referred to as "Test Fluid 1" for the purpose of this discussion. The MR fluid contains 50% iron particles by volume, is more viscous in off-state, and has little to no additives. The other three MR fluids tested are commercially available MR fluids from Lord Corporation. These fluids have lower viscosity in off-state and their viscosity increases with the magnetic field strength.

Before testing the MR fluids, the rheometer was tested without any MR fluid to investigate the friction force between the piston and the rheometer body. Figure 4-26 shows friction force as a function of time for a ramp input of 0.2 Hz frequency and a stroke of 0.15 in. (gap size oscillates between 0.2 in. and 0.05 in.).

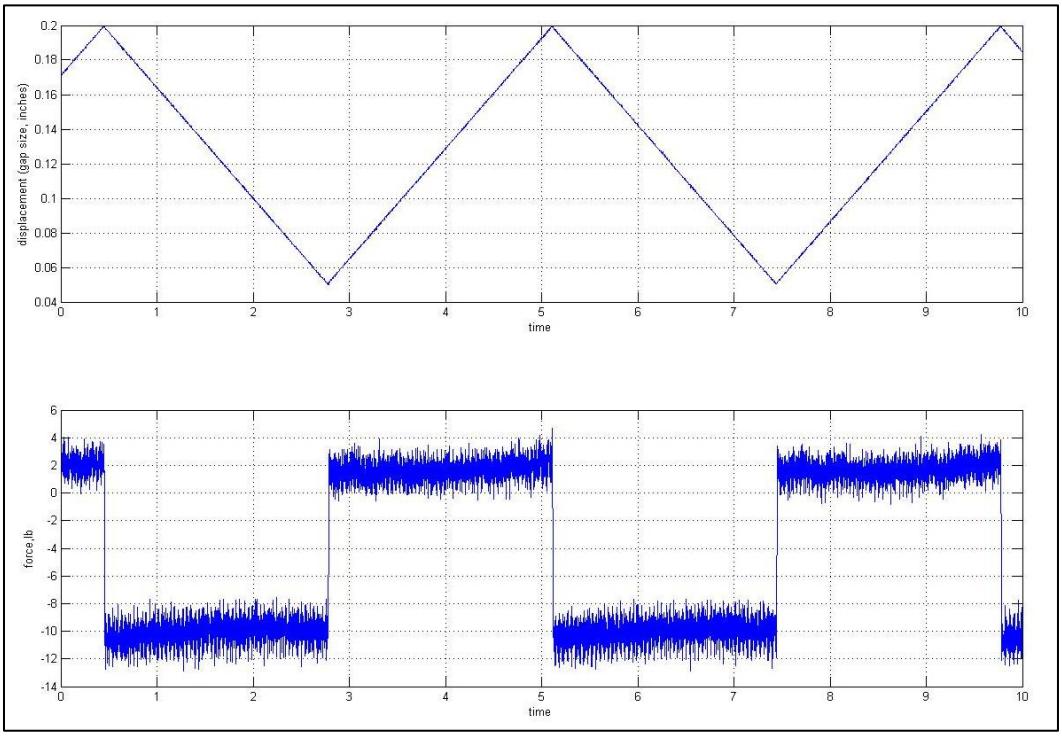

Figure 4-26 Friction force between the piston and the rheometer body

For Test Fluid 1, the peak force is as high as 2500 lb, due to the high volumetric percentage of the iron particles, while the minimum force is zero at the initial gap size. This large range of controllable force is very promising for many applications. As is shown in Figure 4-27, the squeeze force can easily be controlled using a magnetic field. At 0 Amp, the force is very low (MR fluid acting like a Newtonian fluid although with a high viscosity). At higher currents the force increases as the MR fluid becomes stronger. The controllable force range is as low as 100 lb at 0 Amp and as high as 2500 lb at 4 Amp. The relatively high squeeze force at 0 Amp is due to the high viscosity of the fluid (because of its high particle content).

This high force caused some damage to the base adapter of the rheometer. So, it was decided to test this MR fluid at a higher gap size and for smaller strokes. This MR fluid was squeezed from an initial gap size of 0.2 in. to a final gap size of 0.15 in. (0.05 in. stroke). Although the amount of compression was only 0.05 in. the force was as high as 1800 lb. The force displacement plot of squeeze tests for Test Fluid 1 is

shown in Figure 4-27. The numbers in parenthesis on the legend are test data file ID numbers used to track the test data and the specific conditions under which each test was conducted. The piston speed is very slow assuring quasi-static condition (0.03 Hz corresponding to 0.003 in./sec).

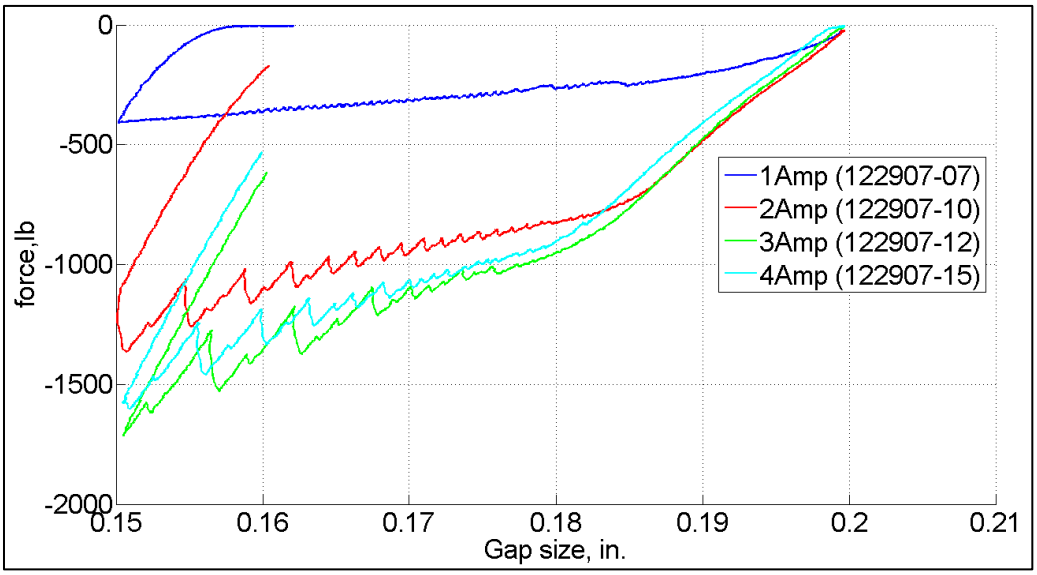

Figure 4-27 Force displacement curve for Test Fluid 1 (50% iron particles by volume)- gap size 0.2 in. to 0.15 in.

One of the behaviors observed during testing was the saw tooth behavior. As can be seen in Figure 4-27, the squeeze force increases and suddenly drops to a lower value, after which it increases again. This is repeated and makes the resulting test curves have a saw tooth shape. This behavior is mainly caused by the air bubbles trapped in the MR fluid. As the force is increased, some of the bubbles explode and escape from the center hole. As a result, the force suddenly drops. But as the piston moves down, the MR fluid is squeezed more and the force increases again.

One of the extensively tested MR fluids was MRF-120RD from Lord Corporation. This is an oil-based MR fluid with a volumetric percentage of 20% of iron particles. The test results presented here are from squeeze tests from an initial gap size of 0.1 in. and a final gap size of 0.05 in. (total travel of 0.05 in.). Tests are carried out for 1, 2, 3, and 4 Amp. The amount of force does not show a significant change at 4 Amp compared to 3 Amp due to the magnetic saturation of the MR fluid. Figure 4-28 shows the force displacement curve for MRF-120RD fluid at different currents.

One of the important properties of MR fluids in squeeze mode, shown in Figure 4-28, is the inability to withstand tensile forces. After the minimum gap is reached, the piston contracts at the same rate of approach but the force rapidly decreases from its maximum value to zero.

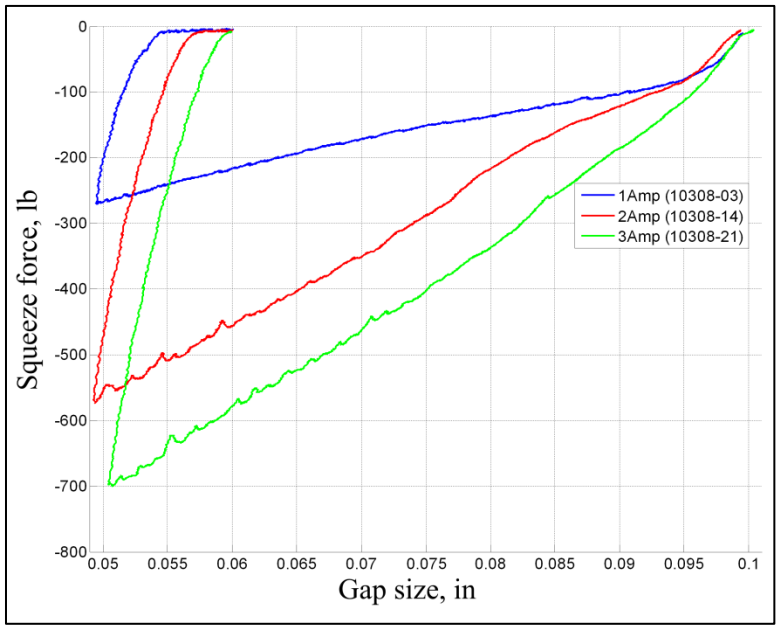

Figure 4-28 Force displacement curve, MRF-120RD, gap size 0.1 in. to 0.05 in.

Because the amount of force did not change considerably at 4Amp compared to 3Amp, the test results for 4Amp are not shown. This is because at 3 Amp, the MR fluid is at its saturation magnetization point. In addition, increasing the electric current to 4 Amp does not make a significant change in the magnetic field in the rheometer because the rheometer body also reaches its maximum saturation limit (Figure 4-30). At 3 Amp, the magnetic field in the rheometer core is approximately 2 Tesla (Figure 4-8), which according to the B-H curve of 12L14 steel (Figure 4-29), is well within the magnetic saturation of the material. Any increase in the coil current beyond 3Amp will not cause an increase in the magnetic field.

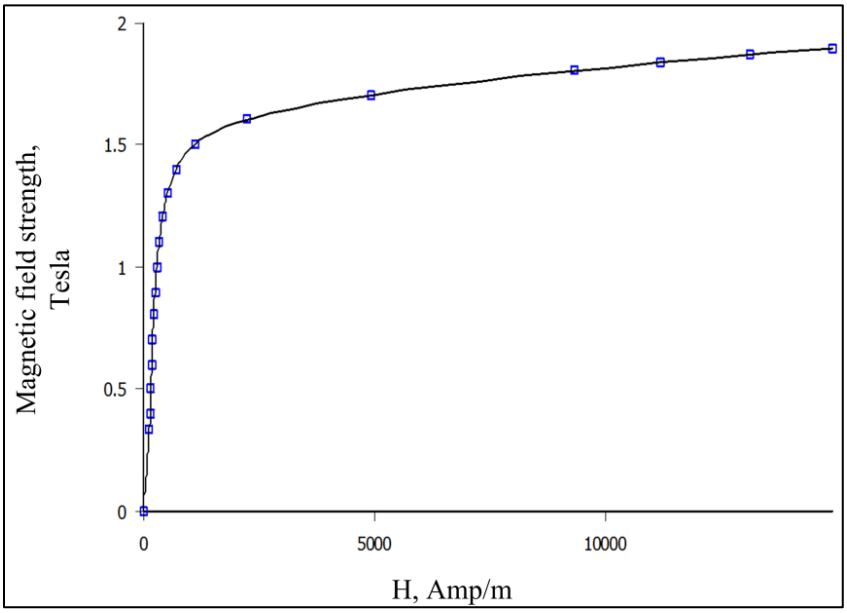

Figure 4-29 B-H curve for 12L14 steel

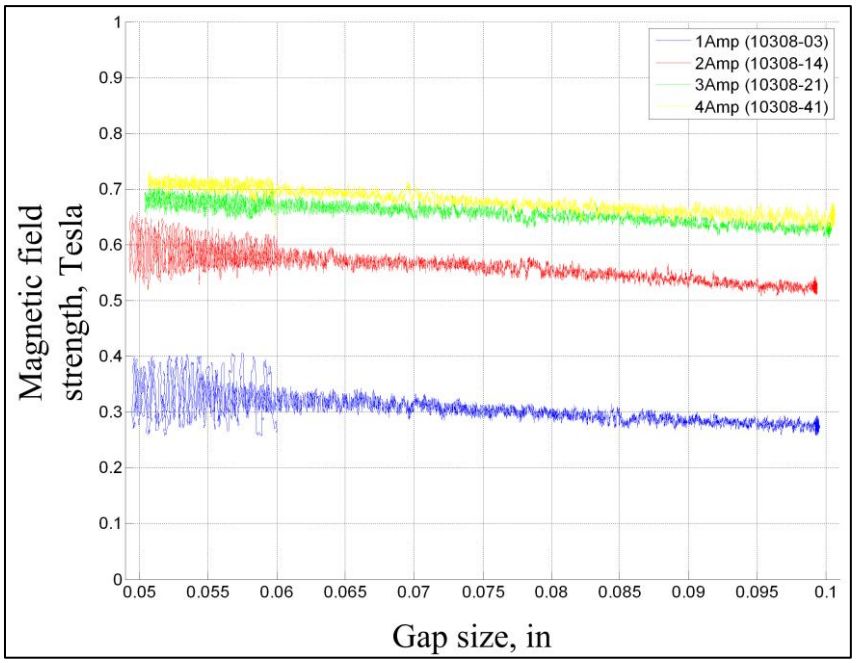

Figure 4-30 Magnetic field vs. gap size, MRF-120RD, gap size 0.1 in. to 0.05 in.

Figure 4-30 is obtained for tests carried out with MRF120-RD. As can be seen, the magnetic field does not increase significantly at 4Amp. Also note that, at each current, the magnetic field increases as the gap size is decreased due to decreasing reluctance in the magnetic circuit. Although the volumetric percentage

of MR fluid has an effect on the magnetic field, this effect is small compared to the effect of gap size. For other MR fluids tested at the same gap size range, the magnetic field density can be read from Figure 4-30.

An interesting behavior observed in tests was the "clumping effect". At high currents (depending on type and composition of the MR fluid), the iron particles are trapped in the magnetic field forming chains, but the carrier fluid is free to flow. Therefore, as the piston moves, the iron particles stay in the gap and the carrier fluid leaves the gap leaving iron particles behind. After a few repetitions of tests, the amount of iron particles is considerably higher in the gap than initially tested as a large percentage of the carrier fluid has left the gap. Essentially, this makes the volumetric percentage of the MR fluid increase and consequently, increases the force in both the On- and Off-states. Clumping behavior is schematically shown in Figure 4-31.

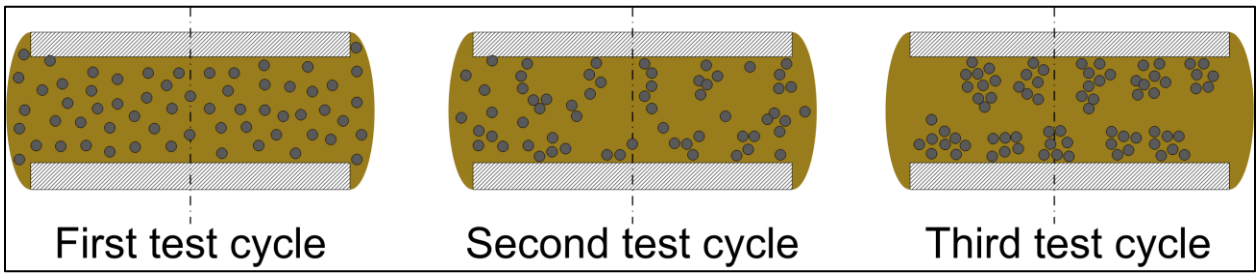

Figure 4-31 Clumping effect schematics after three test cycles without mixing MR fluid between test runs

To investigate this effect, the MR fluid was tested 10 times at each value of current. It was observed that at low currents (1 and 2 Amp for MRF-120RD), test results are consistent (Figure 4-32and Figure 4-33). At 2 Amp, there is a very slight clumping effect occurring.

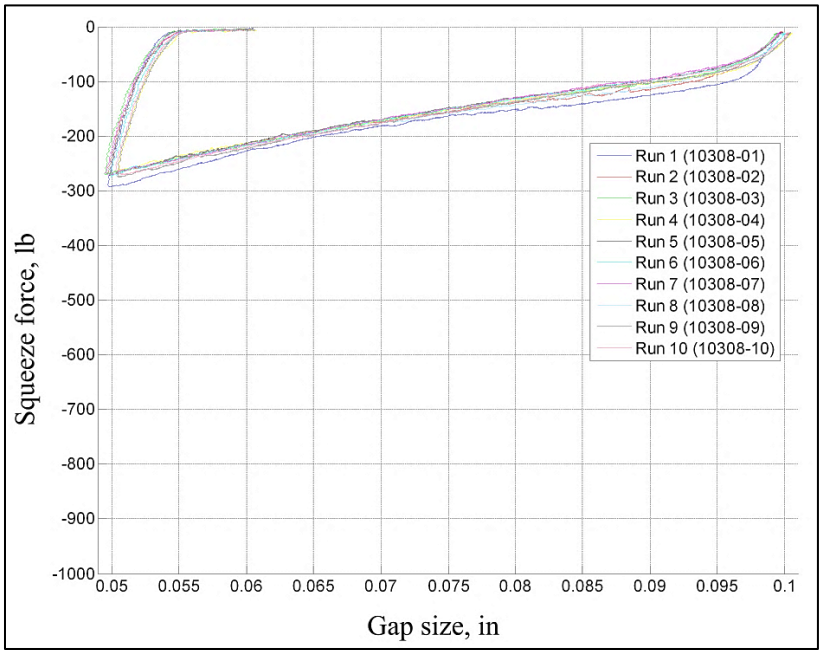

Figure 4-32 Consistent results for MRF-120RD, 1 Amp, gap size 0.1 in. to 0.05 in.

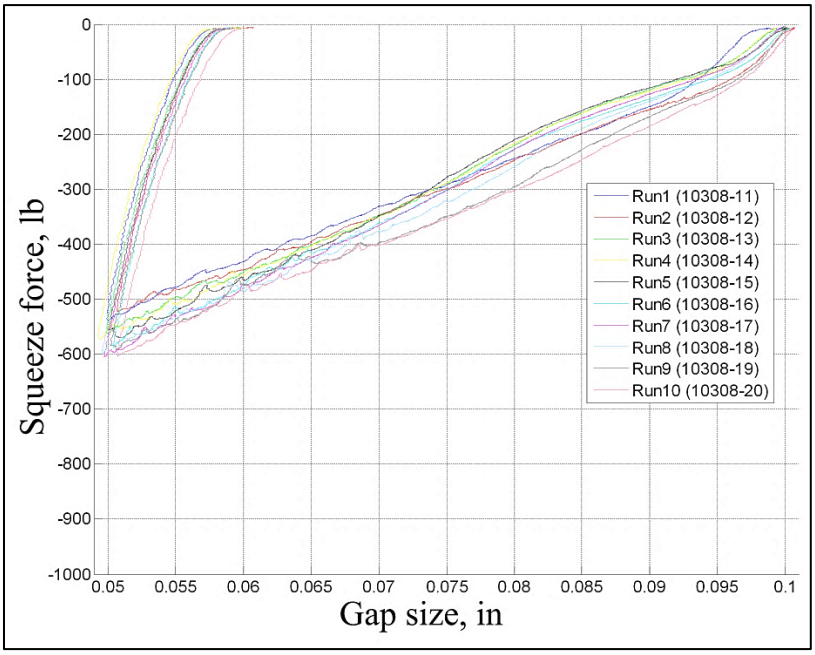

Figure 4-33 Slight clumping for MRF-120RD at 2 Amp, gap size 0.1 in. to 0.05 in.

As the value of current increases, the clumping behavior becomes more severe. Figure 4-34 shows the clumping effect for MRF-120RD at 3Amp. The force increases by almost 300lb (43% increase) after 10 repetitions of tests. The overlaid plot of the tests is shown in Figure 4-35.

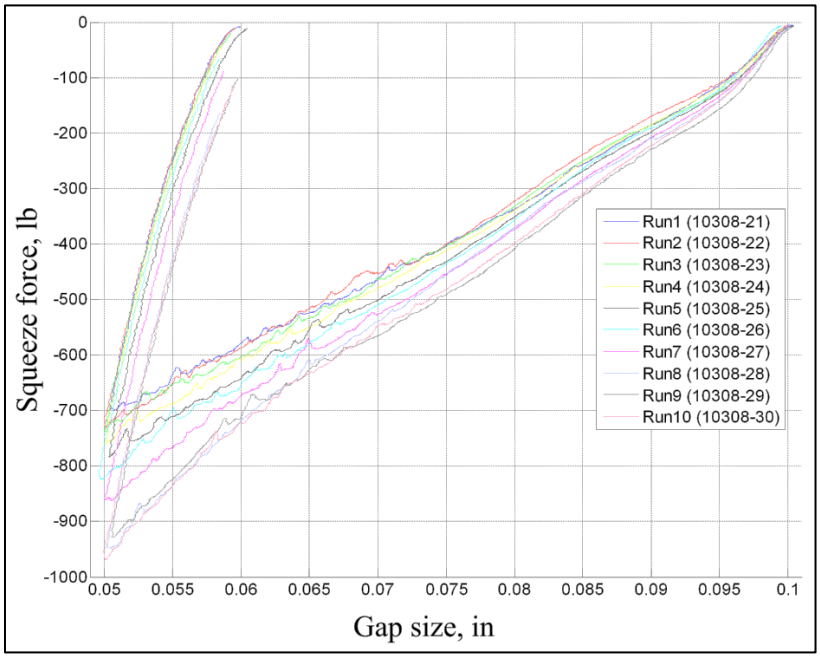

Figure 4-34 Severe clumping for MRF-120RD, 3 Amp, gap size 0.1 in. to 0.05 in.

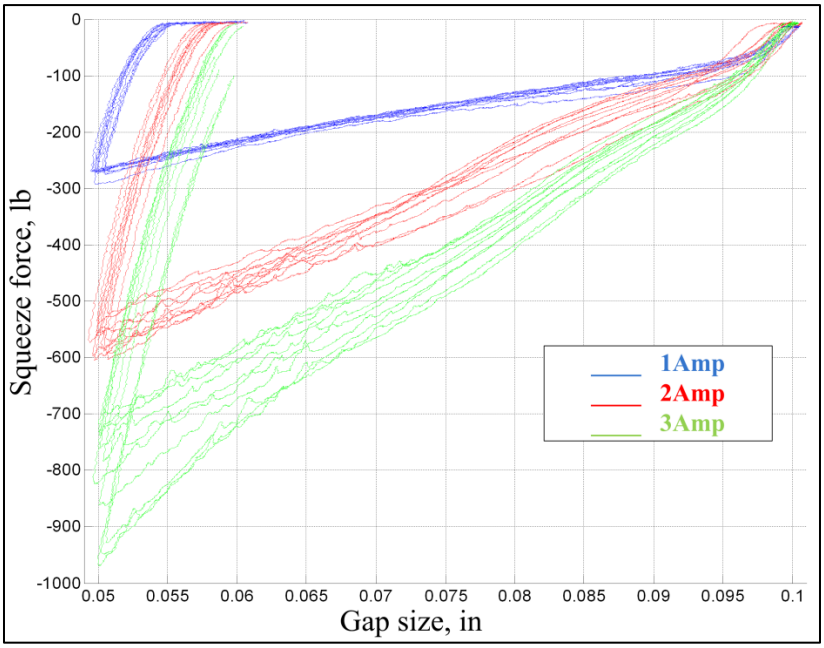

Figure 4-35 Overlaid plot of test results for MRF-120RD, gap size 0.1 in. to 0.05 in.

As clumping happens, some clumps of MR fluid are formed in the MR fluid gap. These clumps have high percentage of iron particles. Deforming and making these clumps to flow needs a very large amount of

force. Figure 4-36 shows the aggregates of MR fluid around the edges of the rheometer. Figure 4-37 shows the MR fluid clumps on the surface of the piston.

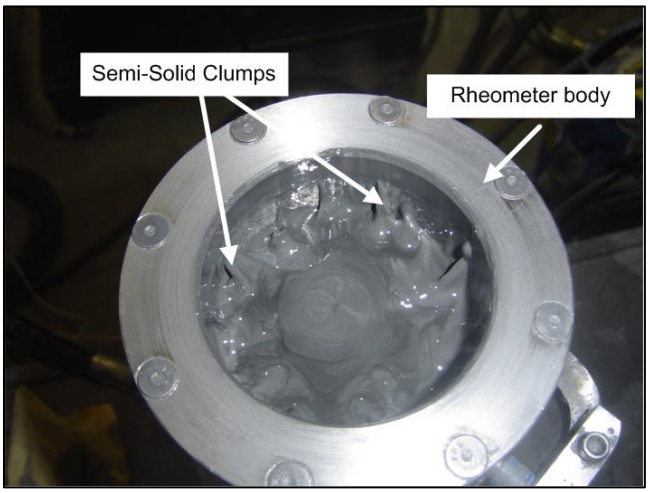

Figure 4-36 Clumping effect. Clumps are formed around the edges of the rheometer cylinder

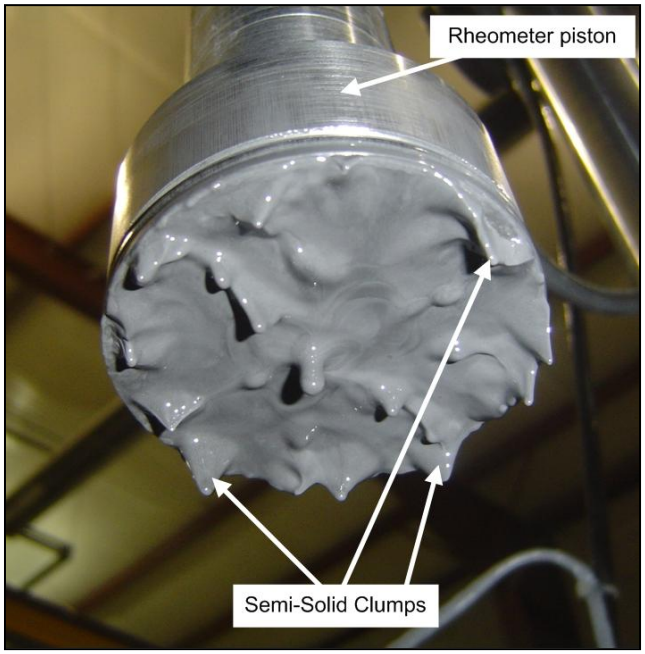

Figure 4-37 Clumping effect. Clumps on the surface of the rheometer piston

Two other MR fluids from Lord Corporation were also tested: MRF-122EG and MRF-132DG. Both of these fluids are oil-based MR fluids with 22% and 32% volumetric percentage of MR particles, respectively. Figure 4-38 and Figure 4-39 show the properties of MRF-122EG. Properties of MRF-

132DG are shown in Figure 4-40 and Figure 4-41. Data are adopted from Lord material technical data sheets.

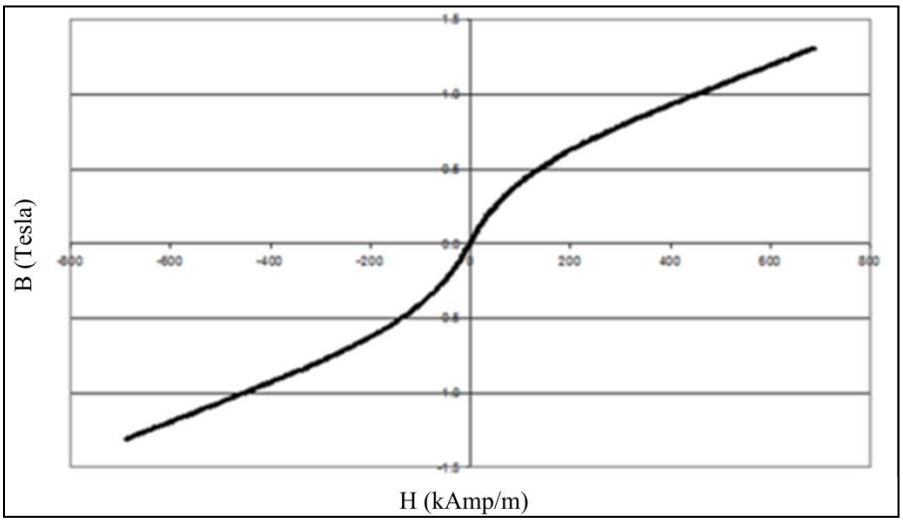

Figure 4-38 B-H curve for MRF-122EG. Lord Corp., used under fair use

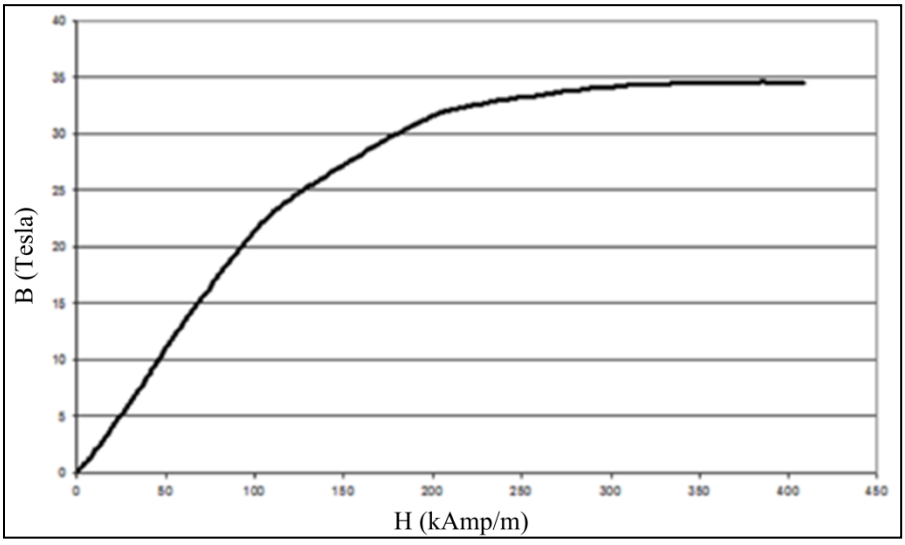

Figure 4-39 Yield stress vs. magnetic field intensity (H) for MRF-122EG. Lord Corp., used under fair use

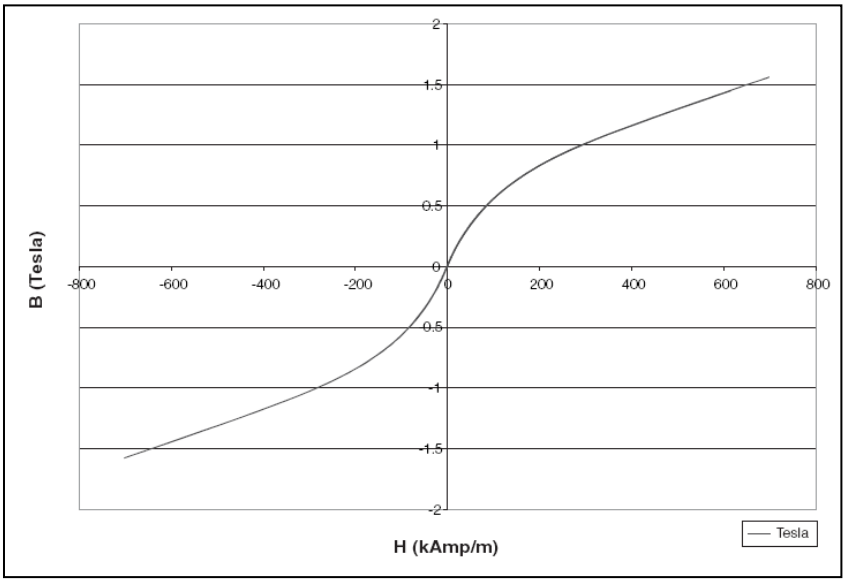

Figure 4-40 B-H curve for MRF-132DG. Lord Corp., used under fair use

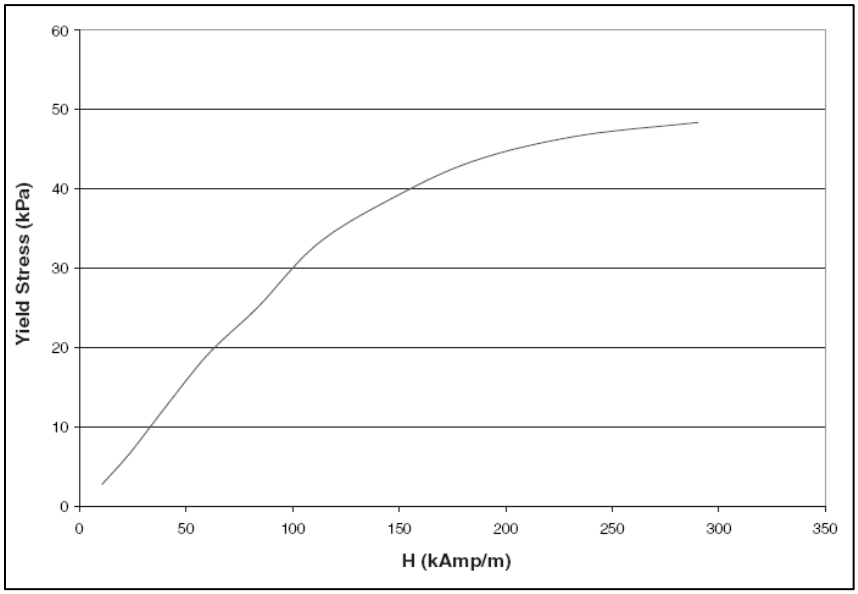

Figure 4-41 Yield stress vs. magnetic field intensity (H) for MRF-132DG. Lord Corp., use under fair use

Similar to tests with MRF-120RD, shown previously, MRF-122EG and MRF-132DG are squeezed from 0.1 in. to 0.05 in. The rheometer is tested as before at a piston speed of 0.003 in./sec. The clumping effect is seen for both fluids, but is more severe for MRF-122EG. This may be due to the chemical composition of the MR fluid. However, this information is not available for comparison. Figure 4-42 shows the force-displacement curve for MRF-122EG for a gap size range of 0.1 in. to 0.05 in. As can be seen, the

clumping happens at 1 Amp and is more severe than MRF-120RD. The amount of the squeeze force increases by 1500 lb during 5 test runs.

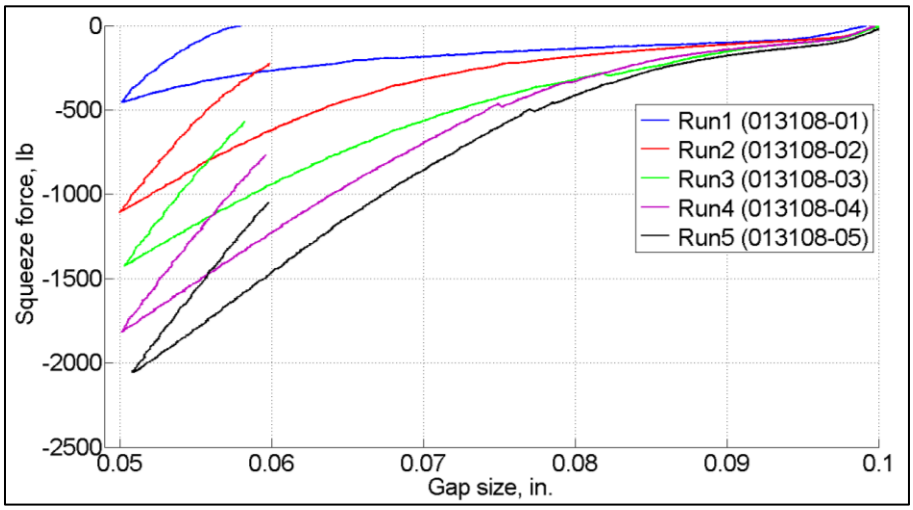

Figure 4-42 Force displacement curve, MRF-122EG, 1 Amp, gap size 0.1 in. to 0.05 in.

The clumping effect is even more profound at higher currents. Figure 4-43 shows that the squeeze force is increased more than 1500 lb during 4 test runs. To prevent possible damage to the rheometer, the MR fluid clumping effect was not tested at 3 Amp.

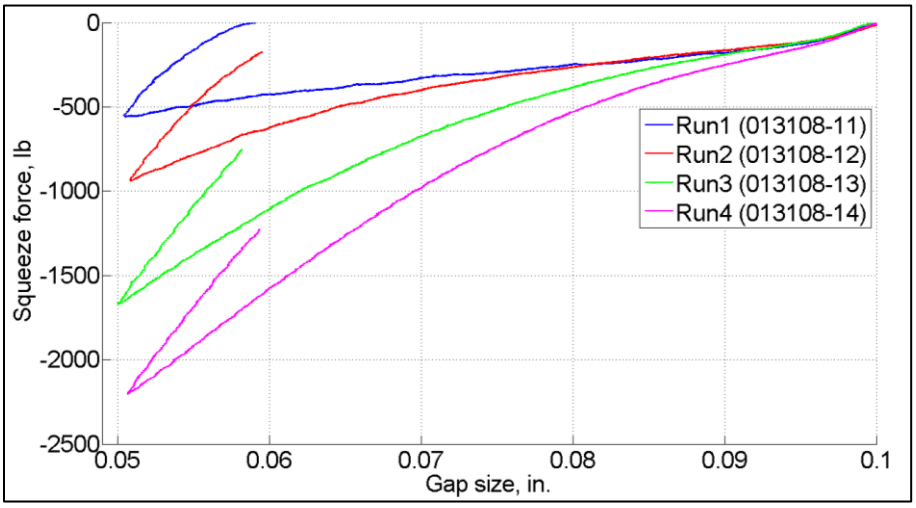

Figure 4-43 Force displacement curve for MRF-122EG, 2 Amp, gap size 0.1 in. to 0.05 in.

MRF-122EG was also tested at an initial gap size of 0.2 in. and a final gap size of 0.15 in. (0.05 in. squeeze test with 0.03 Hz piston frequency). Figure 4-44 shows the force displacement curve for MRF-

122EG at 1 Amp at a gap size range of 0.2 in. to 0.15 in. The amount of force is considerably lower than in Figure 4-42. This shows that the force is mainly dependent on the gap size. Although not as severe as that seen in Figure 4-42 or Figure 4-43, Figure 4-44 shows a slight clumping behavior shown by the MR fluid.

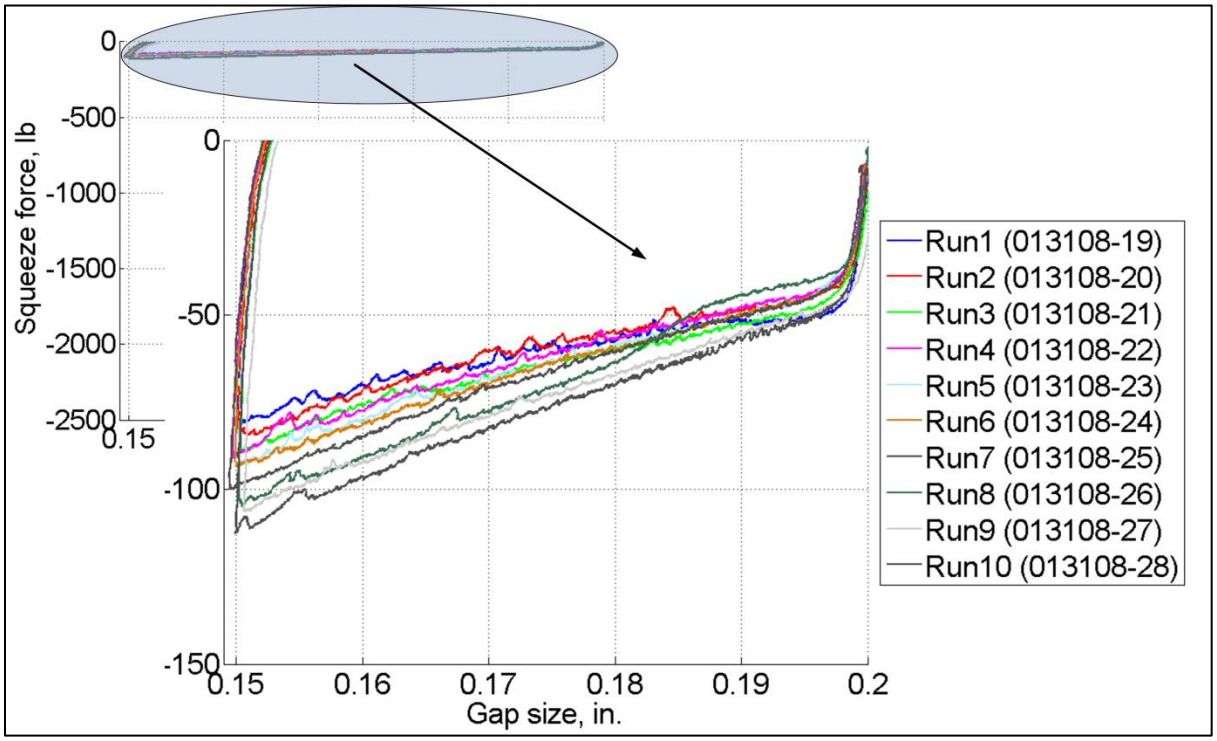

Figure 4-44 Force displacement curve for MRF-122EG, 1 Amp, gap size 0.2 in. to 0.15 in. Slight clumping observed

At 2 Amp, the squeeze force increases because of higher magnetic field strength. The clumping behavior is also considerably higher than 1 Amp tests. See Figure 4-45.

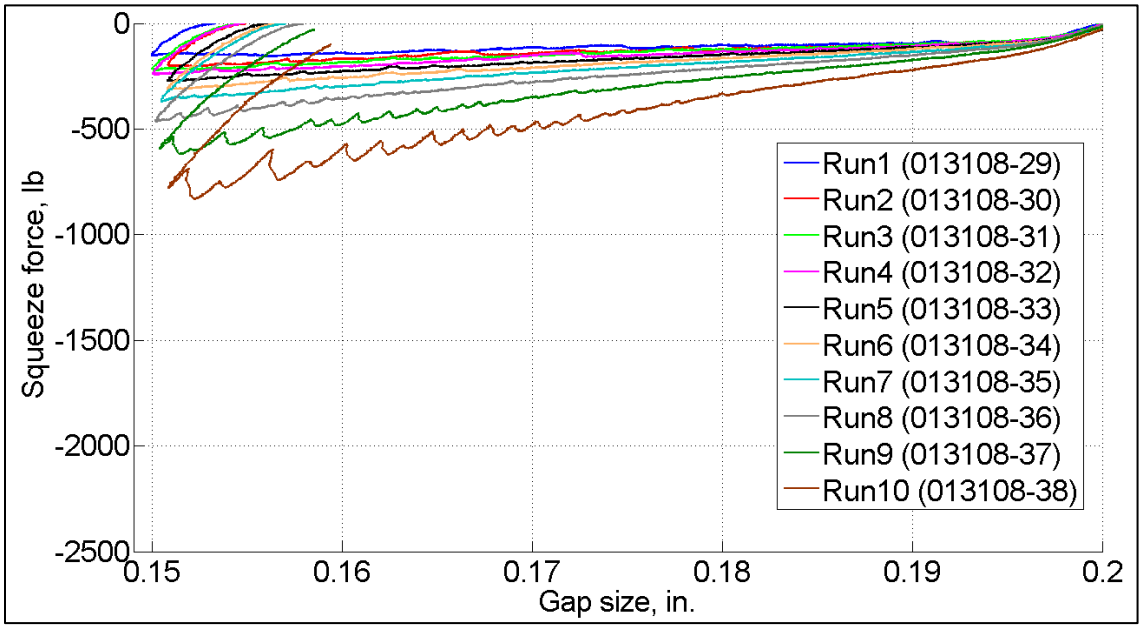

Figure 4-45 Displacement curve for MRF-122EG, 2 Amp, gap size 0.2 in. to 0.15 in. Clumping observed

As discussed earlier, at higher gap sizes, the magnetic field is weaker. Figure 4-46 shows the change of magnetic field strength at different currents as the piston starts from the initial gap size of 0.2 in. and approached the final gap size of 0.15 in. As the gap size decreases, the magnetic field strength increases because of the lower reluctance in the magnetic circuit path.

It should be noted that the change in the magnetic field strength with respect to the gap size is linear as shown in Figure 4-47. This linear behavior simplifies the calculations that relate the variations of magnetic field strength to the gap size.

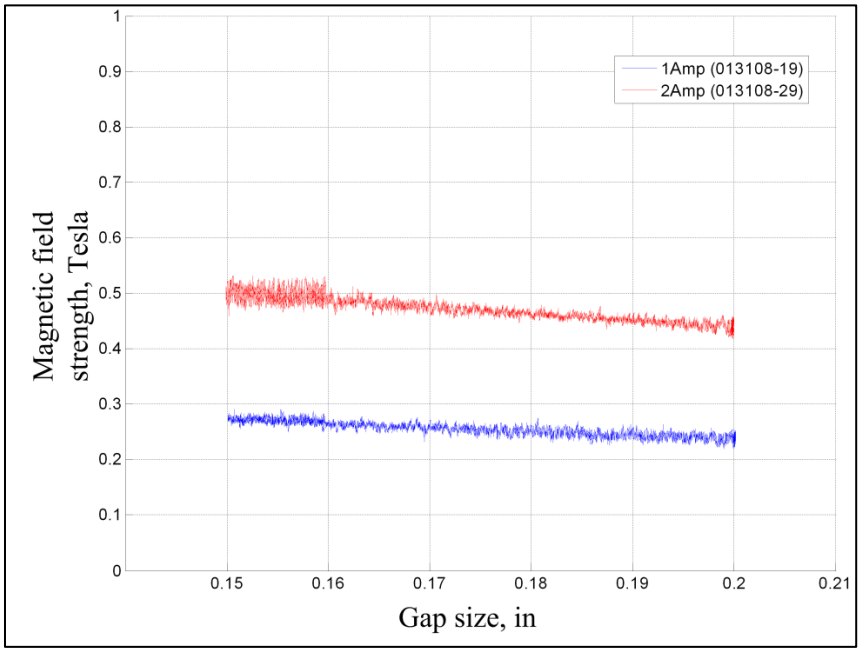

Figure 4-46 Magnetic field vs. gap size, MRF-122EG, gap size 0.1 in. to 0.05 in.

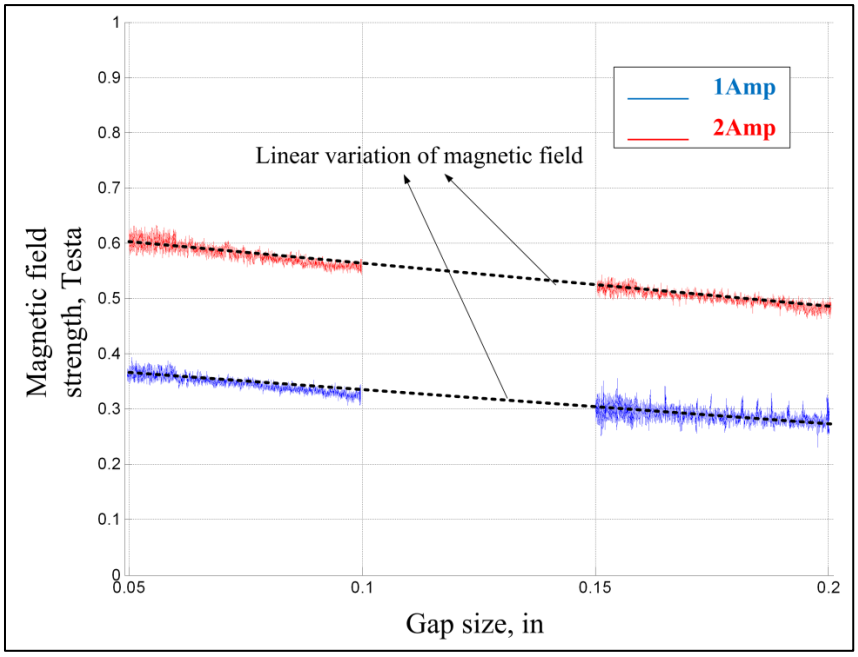

Figure 4-47 Overlaid plot of magnetic field strength vs. gap size. The change in the magnetic field strength is linear with respect to the gap size changes

The overlaid plots of test results for MRF-122EG for different gap size ranges are shown in Figure 4-48 and Figure 4-49 for 1 Amp and 2 Amp, respectively.

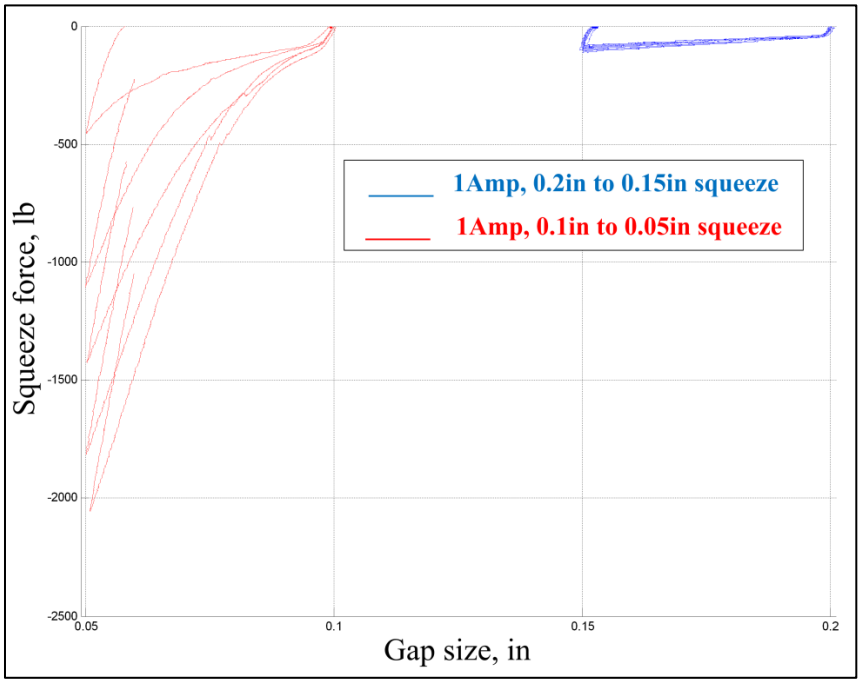

Figure 4-48 Overlaid plot of squeeze tests for MRF-122EG, 1 Amp

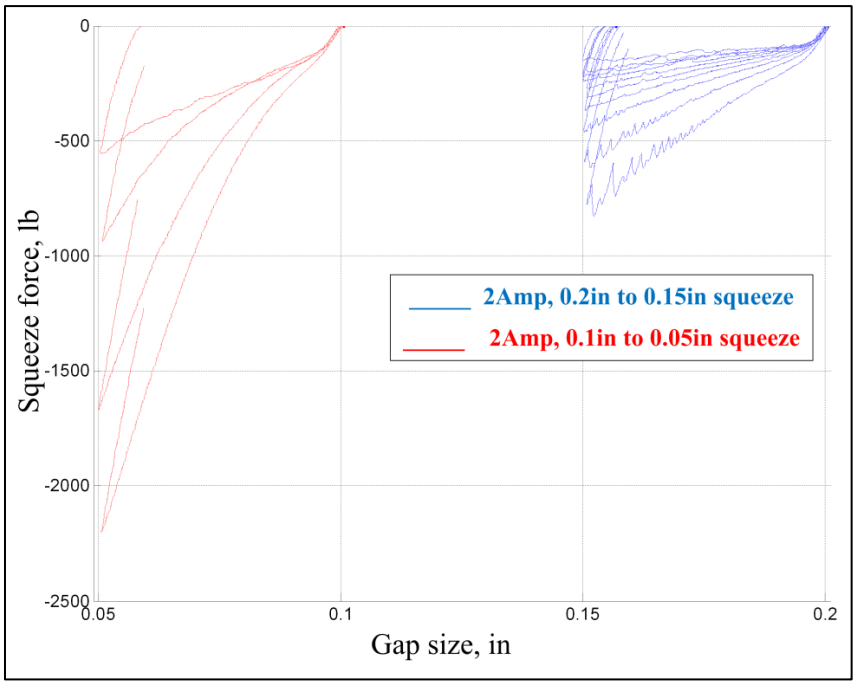

Figure 4-49 Overlaid plot of squeeze tests for MRF-122EG, 2 Amp

The last fluid tested, was Lord's MRF-132DG fluid. The clumping effect is observed to be less for this fluid when compared with MRF-122EG. Although MRF-132DG has higher volumetric percentage of iron

particles (32%) than MRF-122EG (22%), it shows less of the clumping behavior. The reason is not well known. Perhaps, this is due to the specific chemical composition of MRF-132DG, which is not available. At 1 Amp, there is no clumping observed and the 10 runs of testing shows no increase in the squeeze force for a gap size range of 0.2 in. to 0.15 in. At gap size range of 0.1 in. to 0.05 in., MRF-132EG shows some clumping behavior. See Figure 4-50.

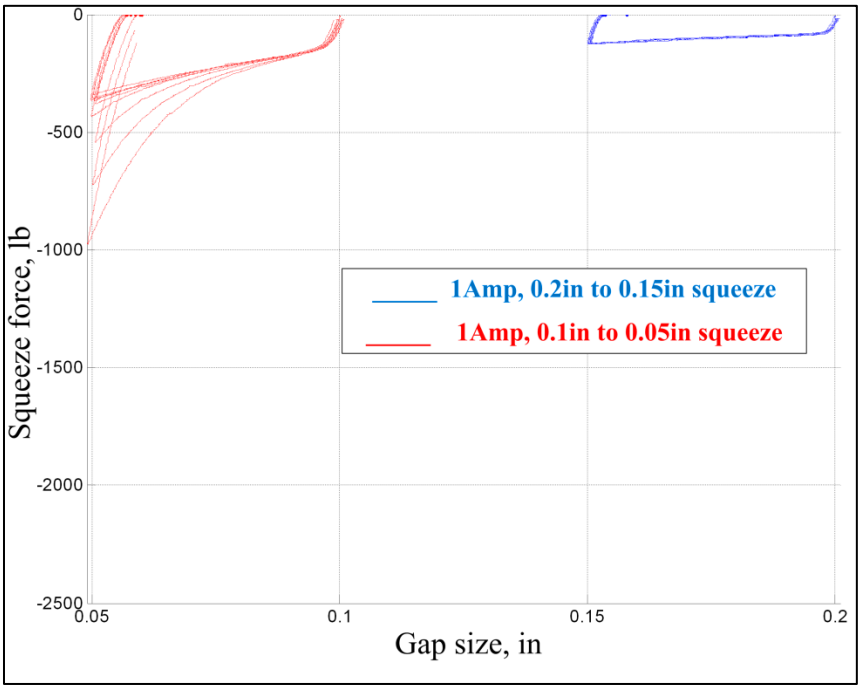

Figure 4-50 Overlaid plot of squeeze tests for MRF-132DG, 1 Amp

At 2 Amp, the clumping behavior is observed, as shown in Figure 4-51. However, the severity of the clumping is not as high as same tests for MRF-122EG.

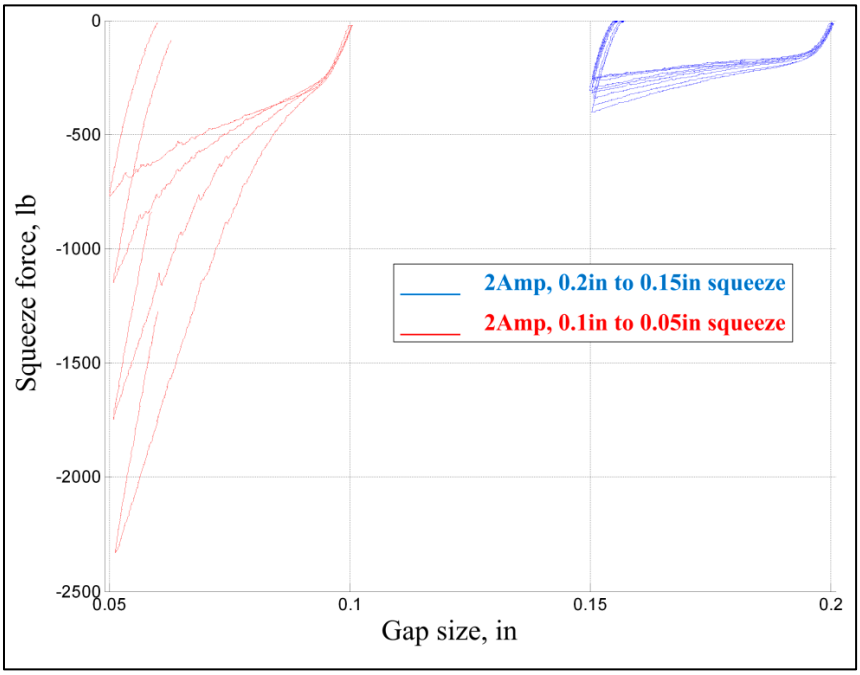

Figure 4-51 Overlaid plot of squeeze tests for MRF-132DG, 2 Amp

The clumping effect will be studied in more detail using the MR pouch test results. In later chapters, this effect is quantified and extensively investigated. This effect can be greatly reduced by introducing an oscillating magnetic field (magnetic dither) during piston return of each test cycle, as will be discussed.

4.8. Force Controlled Tests

Test results discussed so far are based on displacement control of the rheometer. The input signal is a ramp displacement and the squeeze force, measured with the load cell on the MTS frame is the output.

If the force is used as the input to the rheometer (force-controlled mode), the MR fluid will flow based on the force that is exerted. In this case the force is the input and the displacement would be the output. With the force-controlled mode, the yield behavior of MR fluid is more obvious in the beginning of the tests. As force increases, the MR fluid resists the movement of the piston without yield. This can be identified as a high-slope region in the force-displacement curves. Once the MR fluid yields, the slope decreases and the MR fluid flows. Figure 4-52 shows the results of the force-controlled tests on MRF-120RD. For

the force-controlled tests, the initial gap is 0.2 in. and the final gap depends on the current and the amount of force exerted. The force is increased from 0 lb to 220 lb with a constant rate (2 lb per seconds). At higher currents the MR fluid is more resistive and the piston moves a shorter distance.

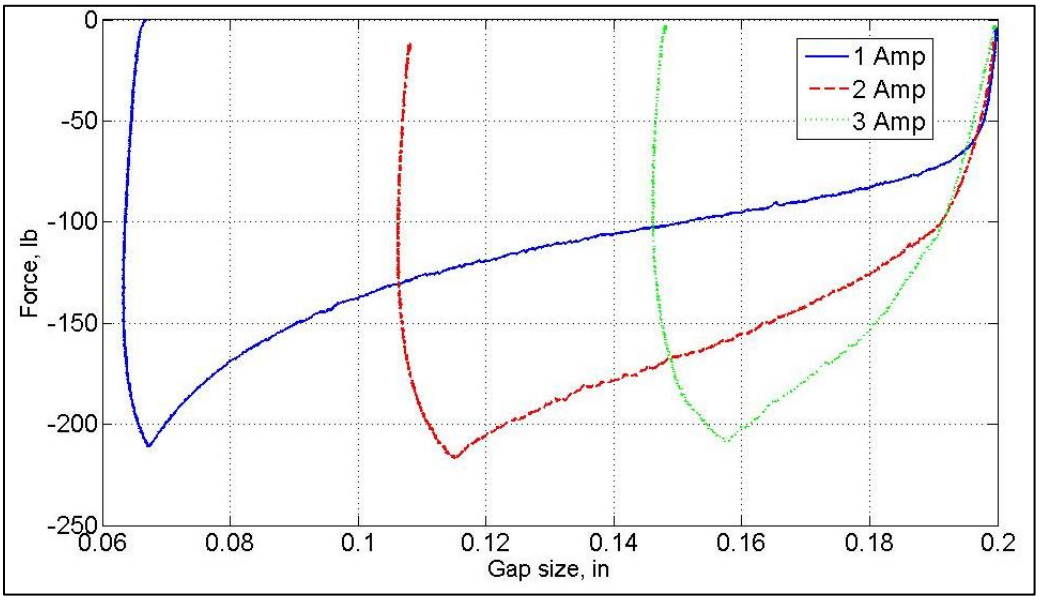

Figure 4-52 Force displacement curve for MRF-120RD in force controlled mode.

4.9. Mathematical Model Validation

The mathematical model introduced in the previous chapter agrees with the test results, especially the force-controlled tests because they better illustrate the pre-yield phase of the MR fluid in squeeze mode. At 1Amp, the average magnetic field density is 0.3 Tesla. At this magnetic field strength, the properties of the MR fluid can be approximated as:

$$\eta = 6.09 \times 10^{-6} \text{ Reyn}$$
$$\tau_y = 1.9 \text{ Psi}$$

The average velocity of the piston is 0.0135 in./sec and $K_H = 0.01$. The non-dimensional force would become:

$$\psi = \frac{7.7133\times10^{-8}}{\zeta^3} - 89.2187 + 10^4\zeta\left(e^{\frac{0.0178}{\zeta}} - 0.1078\right) + 5\times10^5\zeta^2\left(1 - e^{\frac{0.0178}{\zeta}}\right) \quad (4\text{-}2)$$

Equation (4-2) shows that the viscous term is very small compared to the term due to the MR effect. A comparison between the test data and the non-dimensional mathematical model is obtained as shown in Figure 4-53.

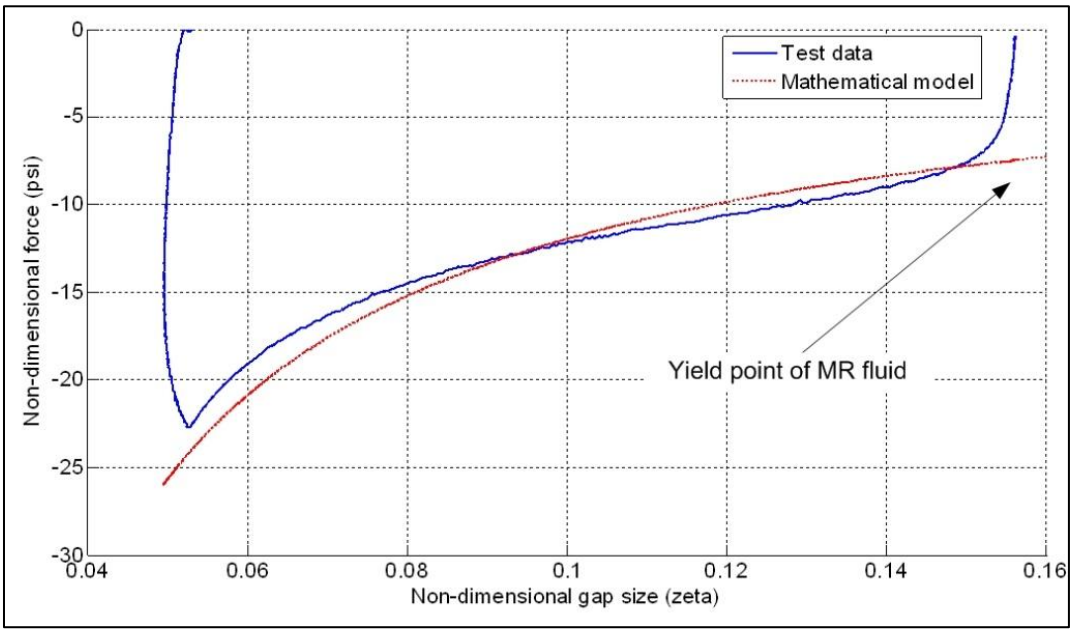

Figure 4-53 Comparison between test data and the mathematical model for MRF-120RD at 1 Amp

Although there are many complications in the flow of MR fluids in squeeze mode [90], this simple non-dimensional model is capable of accurately predicting the basic behavior of MR fluid. This non-dimensional model can be used for the design of MR devices based on squeeze mode. The maximum force for 1 Amp at *h*=0.07 in. is approximately 200 lb. The pressure due to this force on the piston surface is 38.86 Psi. Using a value of $K_H = 0.01$, results in

$$\tau_y = \tau_{y0} + K_H P = 1.9 + 0.01\times38.86 = 2.29 Psi \quad (4\text{-}3)$$

This shows that the equivalent shear yield stress, τ_y, at h=0.07 in. and 1Amp (0.3 Tesla) is 2.29 Psi for MRF-120RD. This corresponds to an increase of 20.5% in the shear yield stress.

As shown in Figure 4-53, the magnitude of the non-dimensional force at the yield point of MR fluid in squeeze mode is approximately 7. This implies that the amount of force required to yield the fluid in squeeze mode is almost 7 times larger than the force needed to yield the same MR fluid in direct shear mode, when $\zeta = 0.16$. As the non-dimensional gap size decreases, the magnitude of the non-dimensional force increases.

As discussed in chapter 3, a more detailed mathematical was also developed for squeeze flow of MR fluids which is capable of predicting flow field, shear rate distribution, and pressure distribution as well as total squeezing force. This model will be validated using test data on the MR pouch and its validation will be presented after introducing the MR pouch and associated test results.

4.10. Conclusion

Through extensive experimental test data, it was shown that MR fluids are capable of carrying a wide range of forces in the squeeze mode, achieved along very short strokes. Several tests were carried out with the squeeze mode rheometer. All test results show similar trends. The amount of the force achieved depends on the type of the MR fluid, magnetic field density, and gap size. This large range of controllable force along a short stroke can be of interest for many applications requiring a large controllable force in envelopes that can only accommodate small strokes. Two possible applications will be discussed in later chapters experimentally and analytically.

A clumping effect was also seen for the tested MR fluids and its severity depended on the magnetic field density, type of the MR fluid, and volumetric percentage of the MR fluid. The design of MR devices that work based on squeeze mode requires a solution to reduce the clumping effect. Clumping causes the

carrying liquid to separate from iron particles reducing test repeatability. The clumping behavior and possible techniques to reduce clumping will be discussed later.

The mathematical model presented in Chapter 3 was able to illustrate the basic behavior of MR fluids in squeeze mode and can be used for preliminary design purposes. The mathematical model is in agreement with the test results. However, the force-controlled test results have a better match with the mathematical model. At force-controlled tests, MR fluid can resist the movement of the piston before yielding. While in displacement-controlled tests, the piston starts to move right after the test is started and MR fluid has to flow.

Chapter 5
Applications of MRF Squeeze Flow in Suspension Systems

As discussed in the previous chapters, MR fluids are capable of providing a much larger range of controllable force in the squeeze mode compared to other conventionally used flow modes (valve and shear modes). This property of MR fluids in the squeeze mode can be used in a variety of applications. The purpose of this chapter is to present the work that has been done to study the applications of MRF squeeze flow in advanced suspension systems. This chapter discusses the feasibility of application of squeeze mode as a novel way of working MR fluids in intelligent systems. Two different devices are presented in this chapter, *"MR Hybrid Damper"* and *"MR Squeeze Mount"*. Design, development, and testing of these two novel devices are presented in detail in this chapter. Mathematical models are also developed for these devices. The mathematical model for MR squeeze mounts was presented in previous chapters and in this chapter, it will be validated using the experimental test data obtained from testing the MR mount. The mathematical model for the hybrid damper needs more elaborate discussion and will be presented in a separate chapter.

5.1. MR Hybrid Damper Design

The basic principle of the Hybrid damper utilizes the properties of MR fluid in squeeze-mode flow. Conventional MR dampers have fairly limited force capacity range. To develop high load range capacity, conventional MR dampers require a large amount of MR fluid making them relatively expensive and heavy (due to the high density of MR fluid). The concept of MR hybrid dampers is to develop highly

adjustable force capacity dampers using only a small amount of MR fluid. MR fluid acts as the "pilot" fluid in the MR hybrid damper. These dampers will have the hydraulic oil as the main operating fluid, just as typical hydraulic dampers do.

A Hybrid Damper piston accommodates an MR controllable unit. This MR unit is capable of delivering variable force. As discussed in previous chapters, MR fluids are capable of carrying a large range of forces in squeeze mode. This property of MR fluids, in squeeze mode, represents the basis of control for the performance of MR hybrid dampers. As the resistance of the MR fluid is increased (by increasing the magnetic field strength passing through MR fluid), the orifice area for the hydraulic oil is decreased and the damper becomes stiffer. If the magnetic field strength is decreased, the MR fluid becomes less resistive and the orifice area offers less resistance to the hydraulic fluid flow, resulting in less damping force.

The MR hybrid damper is mainly different from the conventional hydraulic damper in its main piston. Accordingly, the focus of the design effort was devoted specifically to the design of the damper piston. An ordinary hydraulic damper assembly is used for all other damper components.

5.1.1 Conceptual Design and Working Principles

Figure 5-1 shows schematics of the main piston in a MR Hybrid Damper. As the main piston moves in the damper body, the oil exerts force on the MR spool and moves it. As the MR spool moves, the orifices open and the oil can pass from one side of the main piston to the other side.

As the piston moves up, the oil pressure in the upper chamber is increased. The upper oil chamber is connected to the MR spool through some passages (1). As the pressure is increased, the force on the MR spool lip is increased (2). This force is opposed by the MR force on the lower MR fluid cavity (3). The MR spool movement is directly related to the magnitude of this MR force. As the MR spool moves downward, the orifices around the MR spool lip are cleared partially (4). When the orifices are cleared, the oil will escape through the orifices (5) after which it deflects a disk valve and flows to the lower oil

chamber (6). Based on the resistance of the MR fluid and the amount of cleared orifice area, the damper stiffness can be controlled.

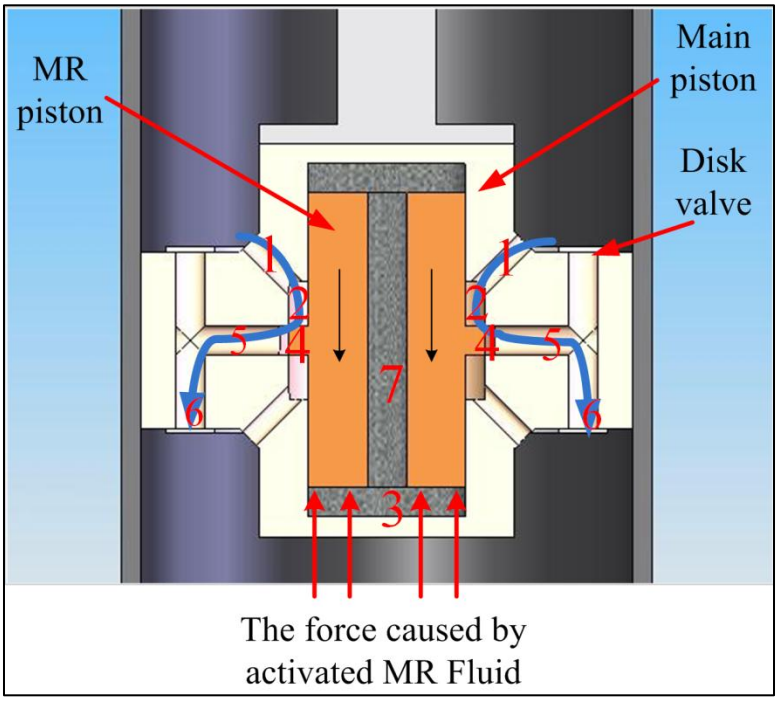

Figure 5-1 A conceptual design for the MR hybrid damper main piston

As the MR spool moves downward, the MR fluid is compressed and the excess of MR fluid will pass through a center hole to the upper MR cavity (7). It should be noted that the deflection disks provide a minimum amount of damping. Their primary purpose is for preventing the oil to pass directly to the other side of piston without moving the spool.

A complementary sequence of events occurs as the piston moves downward.

5.1.2 Generation I Damper

The first generation of the MR hybrid damper was designed and built based on the same concept explained in the previous section. Based on the geometric and performance constraints, an OHLINS CCJ 23/8 hydraulic damper was chosen as the base damper. The damper has a length of 23 in. (597 mm) and a stroke of 8 in. (203 mm). The main dimensions of the piston and other consequent parts were decided

based on the dimensions of the OHLINS CCJ 32/8 damper. Figure 5-2 shows a picture of the OHLINS damper.

Figure 5-2 OHLINS CCJ 23/8 damper

The damper piston drawing is shown in Figure 5-3. The piston consists of two caps that contain all other piston parts. The damper rod screws into the upper cap and the lower cap is attached to the upper cap by screws. One coil is placed in each cap and is fixed by a coil retainer. The coil retainer holds the dynamic seal that seals the MR spool. The center ring is placed between the two caps in the middle of the piston. The MR spool lip is blocking the center ring orifices in its neutral position.

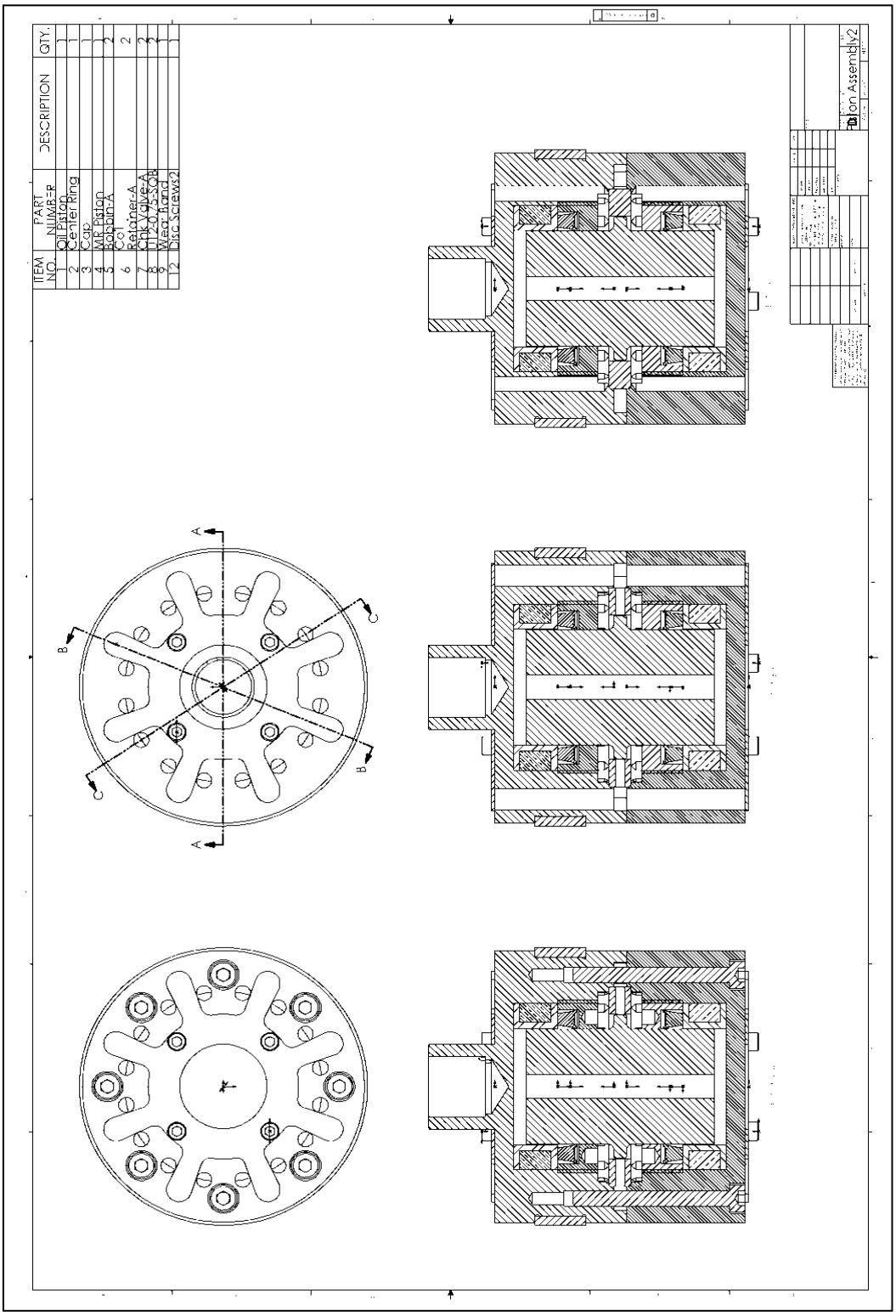

Figure 5-3 Assembly drawing of the damper piston

One deflecting-disk valve is attached to the piston at each side of the piston by four screws. A wear band is installed in a groove on the outer surface of the upper cap. The wear band keeps the piston in the center of the damper body and prevents scratching the damper body by the piston. It also minimizes oil blow-by from one side of the piston to the other during operation. The details of all the piston component designs are provided in the following sections.

5.1.2.1 Upper Cap

The upper cap is shown in Figure 5-4. The upper cap is one of the most complex parts in the design because of its many different oil passages. There are two main types of passages. First, the inlet passages guide the oil into the piston where it pushes the MR spool. Second, the outlet passages guide the oil from inside the piston to the other side of the piston. These outlet passages are covered by the deflecting-disk valve on the outside of the piston. There are a total of 8 inlet and 8 outlet passages. It should be noted that one of the outlet passages is used to pass the wire leads to the damper rod as shown in Figure 5-4.

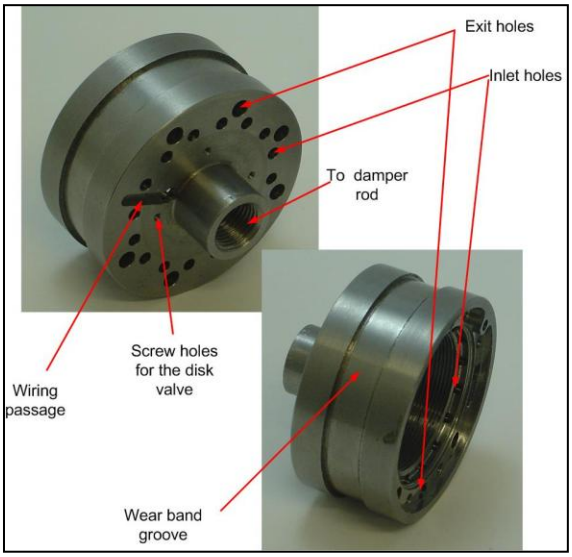

Figure 5-4 Piston upper cap

5.1.2.2 Lower Cap

The lower cap is shown in Figure 5-5. The inlet and outlet passages in each cap coincide with one another once they are assembled. Similar to the top cap, one of the outlets is used as a wire passage to guide the wires to the damper rod.

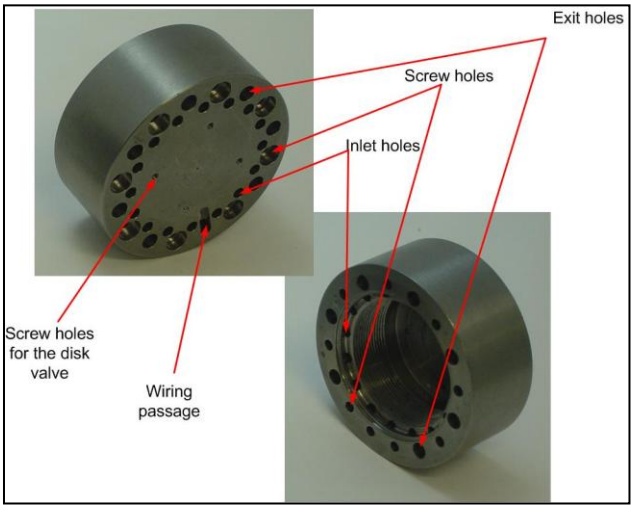

Figure 5-5 Piston lower cap

5.1.2.3 MR Spool

The MR spool is the only moving part in the piston assembly and moves relative to the piston. The MR spool can move 2mm up and 2mm down from its neutral position. The MR spool is shown in Figure 5-6. The through hole in the MR spool provides a passage for the MR fluid to flow to the other side on the MR spool as the spool is moved. This hole was needed to account for the volume change in the MR section when the spool moves.

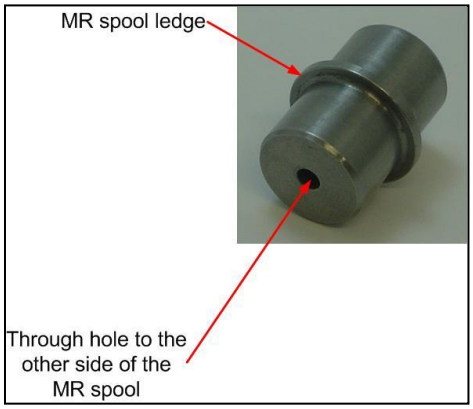

Figure 5-6 MR spool

5.1.2.4 Coils

Two coils are used in the piston to magnetize the MR fluid. The coil spools were initially made of Aluminum, although the recent versions are made of Delrin to eliminate scratching of the spool during moment. 24 AWG wire was used to wind the coils, resulting in approximately 50 turns (Figure 5-7). The wire leads are passed through outlet holes in the upper and lower caps and then routed to the damper rod.

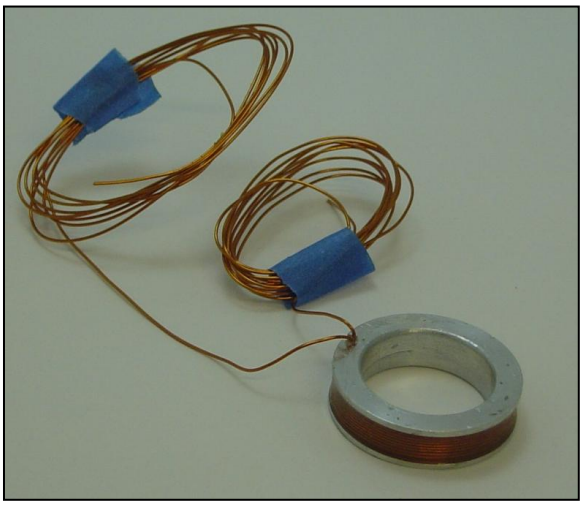

Figure 5-7 magnet wire coil (24AWG wire, 50 turns)

5.1.2.5 Coil Retainer

Coil retainers are screwed into the threads in the caps and secure the coils in place. They also hold the dynamic seals. The coil retainers are screwed into the caps using a spanner wrench. The coil retainer is shown in Figure 5-8.

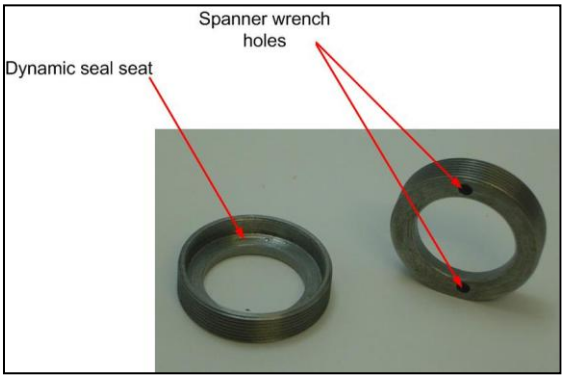

Figure 5-8 Coil retainer

5.1.2.6 Dynamic Seal

Dynamic seals are installed on the coil retainers and provide sealing between the MR spool and the coil retainer. The seals prevent the leakage of the MR fluid into the hydraulic oil in the damper.

Figure 5-9 Dynamic seal

5.1.2.7 Center Ring

The center ring is placed between the two caps and is clamped and sealed by the clamping force of the cap screws. Some transverse holes are drilled in the center ring which act as guides for the oil to pass from one cap to the other cap.

Figure 5-10 Center ring

5.1.2.8 Deflection Disk (Disk Valve)

The disk valve used in the damper is shown in Figure 5-11. The outlet passages in each cap are blocked by a disk valve. The disk valves prevent reverse flow as well as provide a minimum amount of damping.

Figure 5-11 Disk valve

5.1.2.9 Piston Assembly

The overall piston assembly of the Gen. I damper is shown in Figure 5-12.

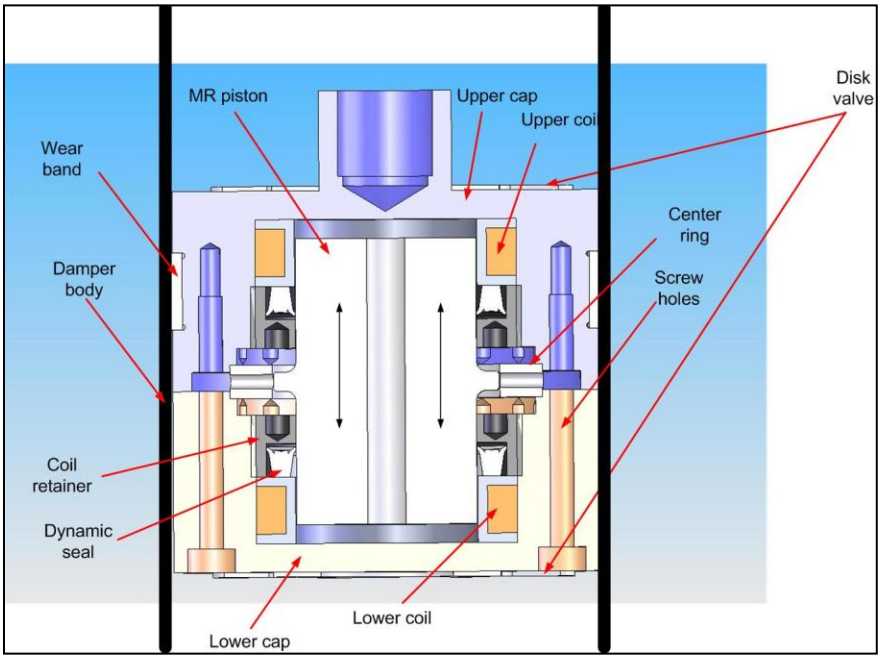

Figure 5-12 Piston assembly - Gen. I

Figure 5-13 shows the assembled damper piston.

Figure 5-13 Assembled damper piston

5.1.3 Damper Magnetic Analysis

The FEMM software is used for magnetic analysis of the MR hybrid damper. The coils were made from 24 AWG wire with 50 turns. The coils were connected in series and the maximum current through the coils was 5 Amp. Figure 5-14 shows the FEMM results. The magnetic field density is approximately 0.7 Tesla at 5 Amp. This amount of the magnetic field density can energize the MR fluid up to its saturation limit.

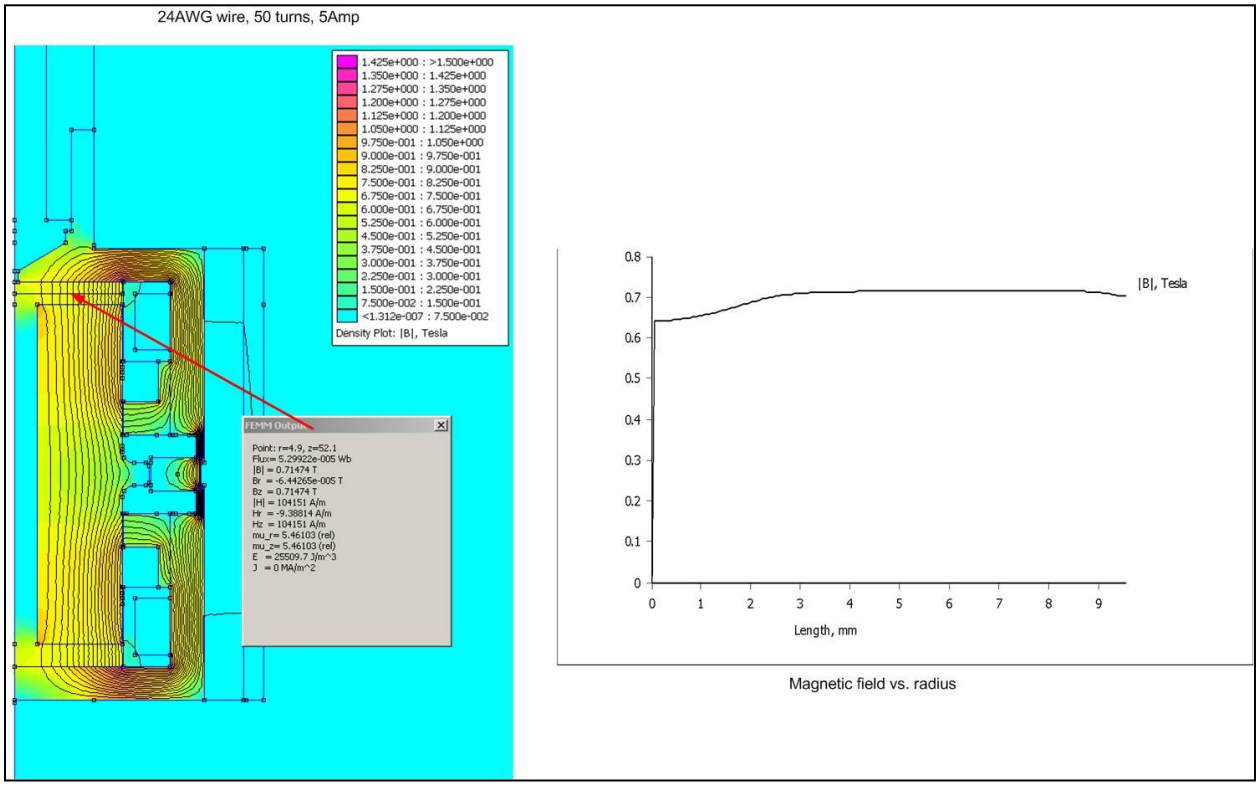

Figure 5-14 Magnetic analysis of the MR hybrid damper at 5 Amp

If the wires were connected so that the magnetic field from one of the coils is opposing the other magnetic field, the magnetic fields were weaker. Figure 5-15 shows this case. As can be seen, the magnetic field density is decreased from 0.715 Tesla in additive magnetic field configuration (Figure 5-14) to 0.67 Tesla in subtractive magnetic field configuration (Figure 5-15).

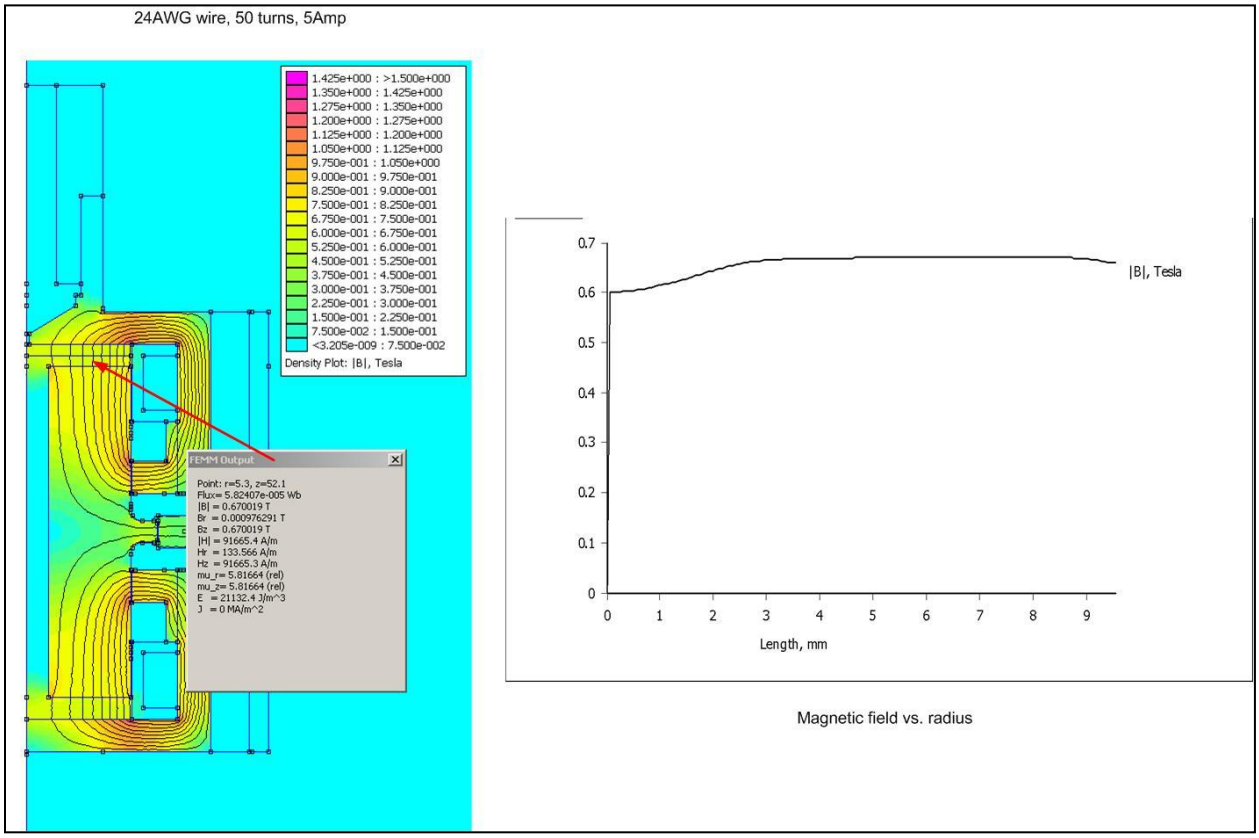

Figure 5-15 FEMM results for the MR hybrid damper at 5 Amp. Subtractive magnetic fields

The variations in the magnetic field as the MR spool moves can be considered in a similar way to the squeeze mode rheometer.

5.1.4 Improvements in Damper Hardware and Design

As will be discussed in next sections, the first generation of MR hybrid damper showed encouraging results. However, there were some problems discovered during the assembly process and others encountered during testing. One of the main issues was oil bypassing the MR spool valve in the piston. This behavior resulted in poor performance of the damper. This problem was caused because the piston was not sealed properly and some of the oil could pass through the piston without passing through the designed passages. Also, leakage of MR fluid into the damper hydraulic oil occurred. This problem occurred because of the inappropriate sealing of MR fluid cavities.

To solve the first leakage problem, some modifications were done on the damper piston. A gasket paper was added between the two piston caps and the MR spool lip was rebuilt to have a thicker lip providing more overlap with the center ring. These modifications are discussed in next section on testing of the MR hybrid damper. The modified MR spool and the assembled piston, with the gasket paper, are shown in Figure 5-16 and Figure 5-17. Additionally, four through holes were added to the MR spool to reduce the possible clumping of MR fluid observed in squeeze mode rheometer testing.

Figure 5-16 Modified MR spool

Figure 5-17 Assembled MR piston with gasket paper

The second leakage problem (leakage of MR fluid out and mixing with damper oil) was one of the major problems encountered. The surface finish of the MR spool was modified to improve sealing. Also, tighter dynamic seals were used. Although these modifications decreased the leakage of the damper, the problem was not completely eliminated. This led to the second generation of MR hybrid dampers.

The second generation hybrid damper design uses a pouch concept to enclose the MR fluid in an isolated container. The MR fluid is filled in a pouch attached to the MR spool. With this design, there was no need to have dynamic seals. This improves the performance as will be explained.

A new spool was fabricated to accommodate the new pouch design. This spool was designed with a small groove in the circumference for pouch attachment. The ends of the MR spool were machined to have a dome shaped form. This would allow the pouch to bulge outside of the MR activation region. Figure 5-18 and Figure 5-19 show the modified MR spool (also called pouch spool) and its assembled photo, respectively.

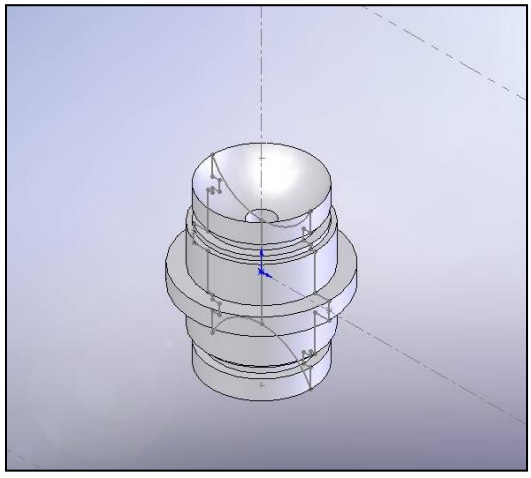

Figure 5-18 Modified MR spool

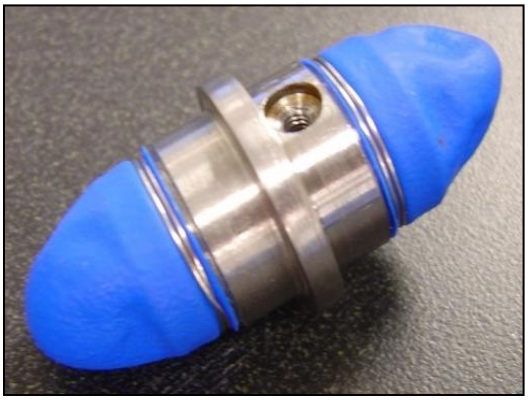

Figure 5-19 Assembled MR pouch spool

Because the MR fluid is in a contained space, there is no need for the dynamic seals. The space freed up from removing the seals was used as a room for the pouch to bulge in. A cross-section of the Gen. II damper is shown in Figure 5-20.

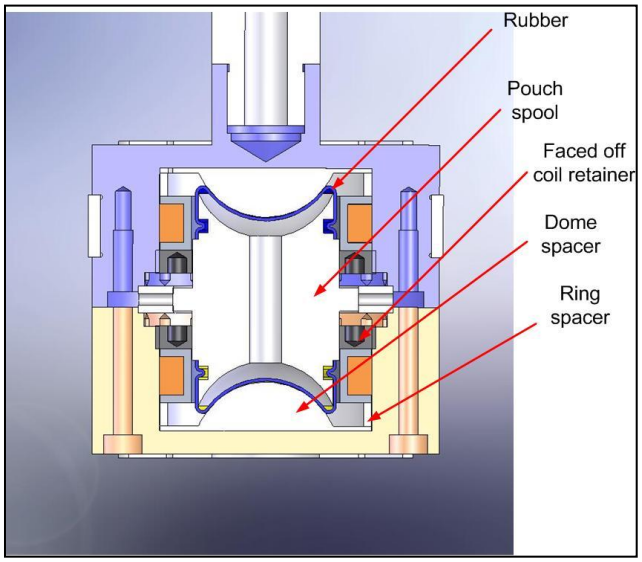

Figure 5-20 Cross-section of the Gen. II damper

As can be seen in Figure 5-20, in this design the coils are moved towards the center ring, creating a cavity under the coil for the rubber to bulge out. The interior parts of the Gen. II MR hybrid damper are shown in Figure 5-21.

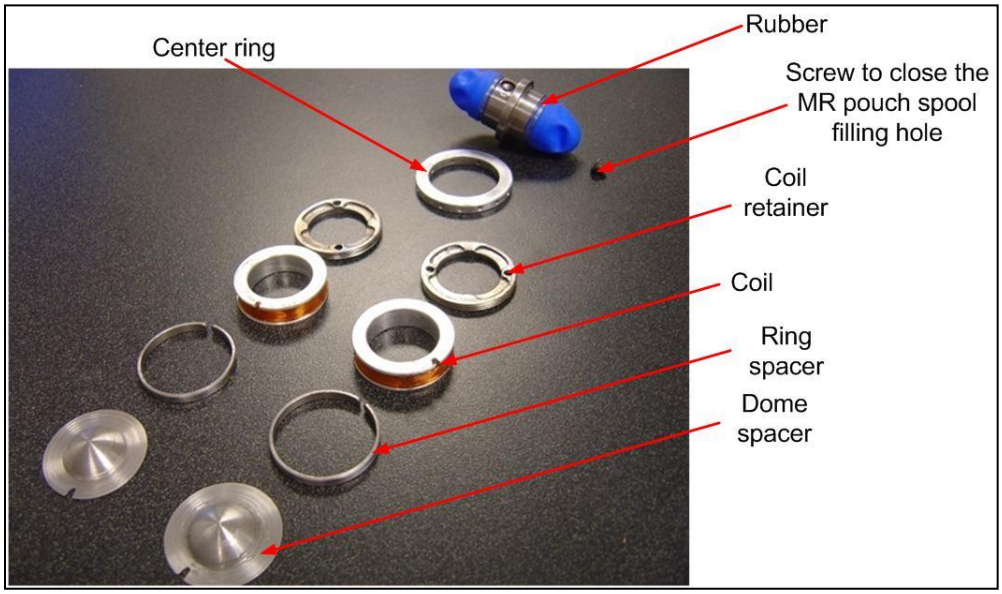

Figure 5-21 Gen. II MR hybrid damper piston interior parts

The cross-section drawings of the Gen. II MR hybrid damper is shown in Figure 5-22.

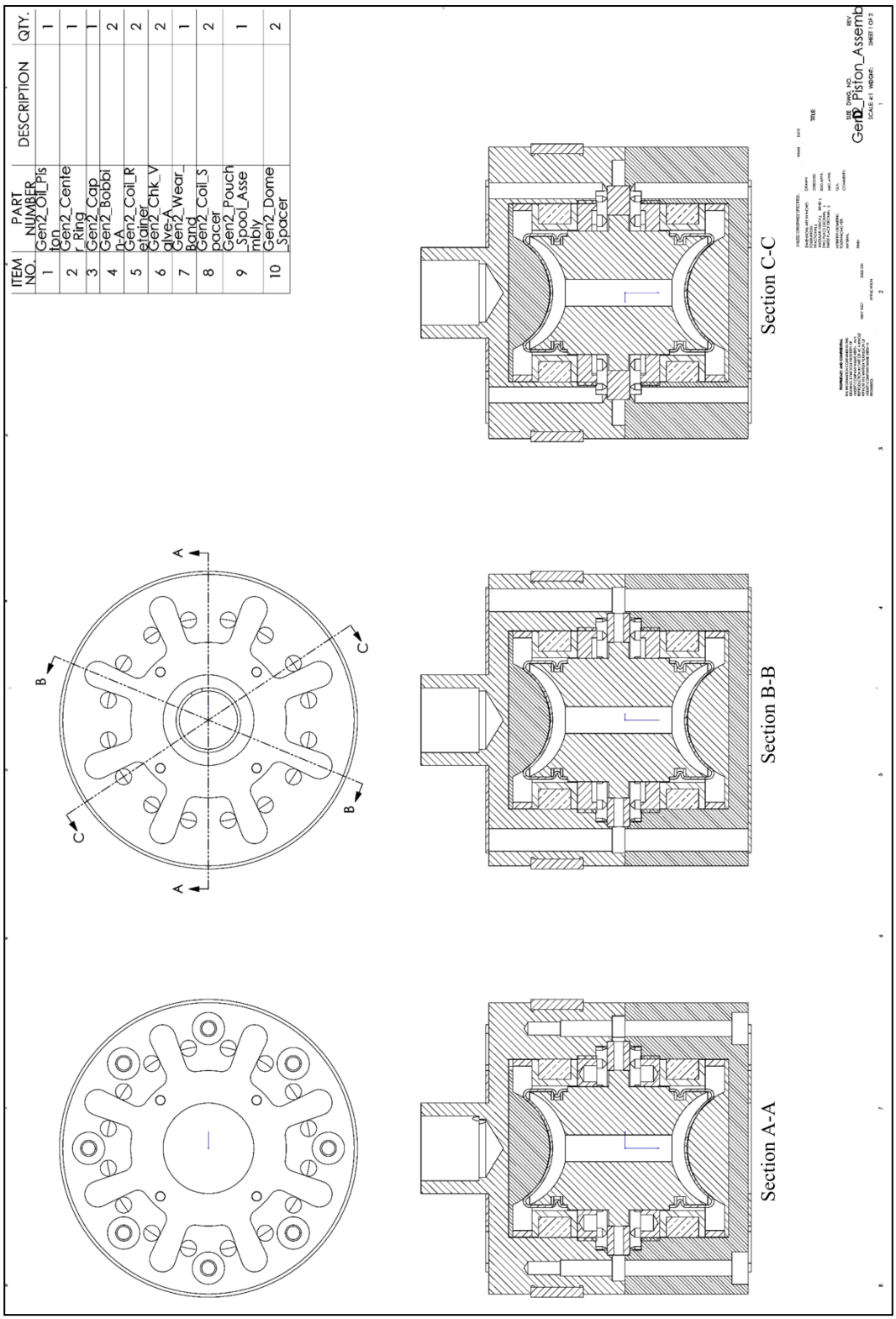

Figure 5-22 Gen. II MR hybrid damper drawing

The Gen. II MR hybrid damper was tested and encouraging results were obtained. This damper can be used as a proof of concept showing that these dampers are capable of showing controllable force range with a small amount of MR fluid in squeeze mode. The test results are presented and discussed in the next section.

5.2. MR Hybrid Damper Testing

In this section, test results of the MR hybrid damper are discussed. The hybrid damper was tested under a variety of conditions to characterize its behavior. The testing revealed many damping behaviors and other issues which are explained and discussed.

The same MTS machine used for rheometer testing was used for damper testing. The damper was tested mainly with a Haversine input displacement with different frequencies and amplitudes. Figure 5-23 shows the MR hybrid damper mounted on the MTS load frame. The load cell is located on the upper jaw. The LVDT displacement sensor is located on the lower jaw. The stroke of the damper is 7 in. For testing, the damper was set in its mid-stroke (approximately 3.5 in.). Once this setup is configured, the damper is ready for testing.

The coil leads of the damper are connected to a DC power supply which supplies a 0-5 Amp of DC current for the MR circuit in the piston.

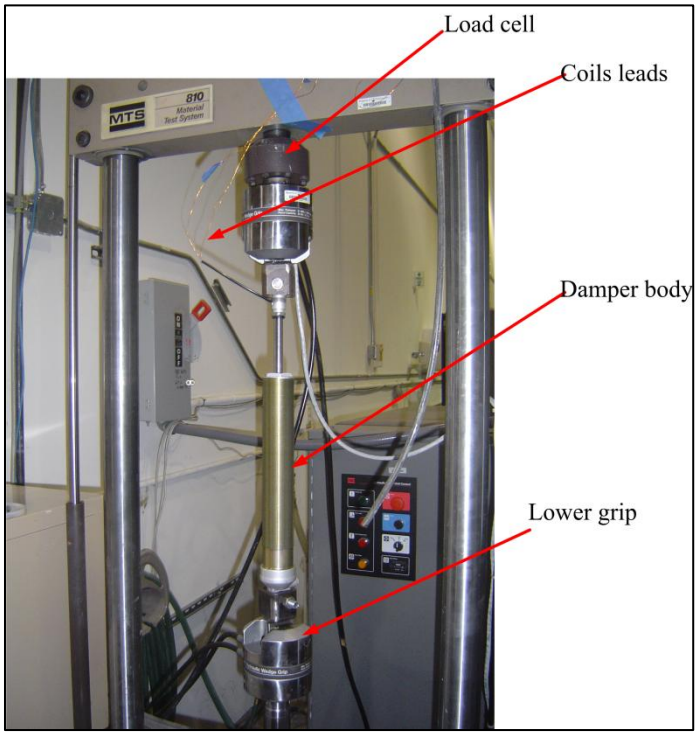

Figure 5-23 MR hybrid damper mounted on MTS load frame

The first set of tests with the hybrid damper was not successful because of the leakage problems in the damper piston. The leakage caused low damping forces from the damper. The source of leakage was thoroughly investigated and two sources were found in the damper piston.

- Leakage between the MR spool and the center ring. This leakage problem was addressed by modifying the MR spool. The MR spool was made with tighter tolerances and thicker lip to prevent leakage. See Figure 5-24.

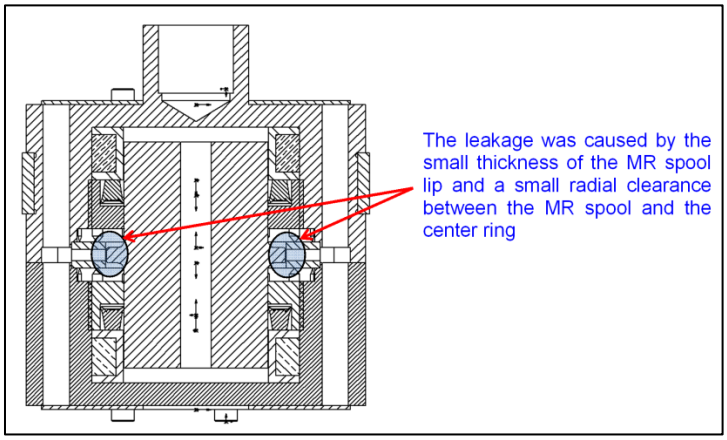

Figure 5-24 Leakage between the MR spool and the center ring

- Leakage between the center ring and the piston caps. The problem was caused because it was tried to seal two different surfaces by clamping the two piston halves using screws. These surfaces are shown in Figure 5-25. This issue was addressed by adding a gasket paper between the caps and using a slightly larger center ring to seal internally with the caps, shown in Figure 5-26. The outer sealing was provided by the added gasket paper.

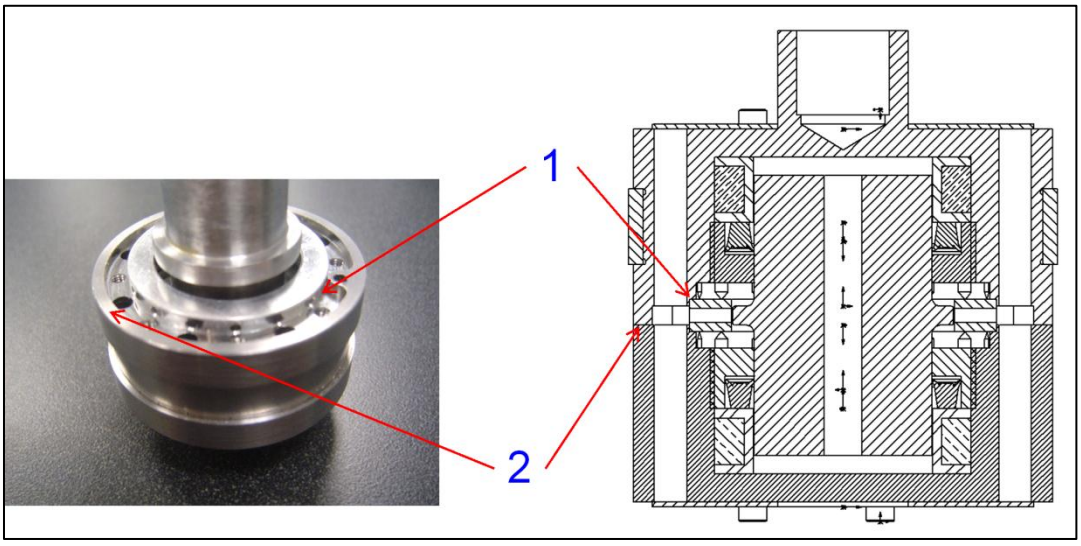

Figure 5-25 Leakage between the center ring and the piston halves

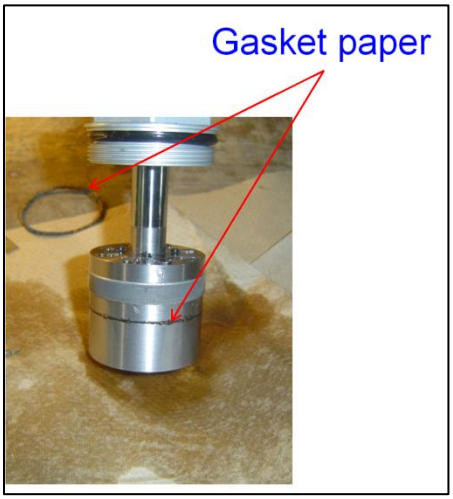

Figure 5-26 Gasket paper added between the piston halves to prevent leakage

After these modifications, the damper was reassembled and tested. MRF-132DG was used to fill the MR sections of the damper. The coils in the piston were wound with 24 AWG wire allowing a maximum current of 5 Amp. The coils were connected in series. As the power supply to the coils was turned on, both coils were activated and MR fluid in both cavities was energized.

Test results were encouraging. Figure 5-27 and Figure 5-28 show test results for 0.5 Hz and a peak to peak stroke of 2 in. When the current was increased, the amount of damper force also increased. It should be noted that undesirable peaks in damping force were also observed. These peaks will be discussed shortly.

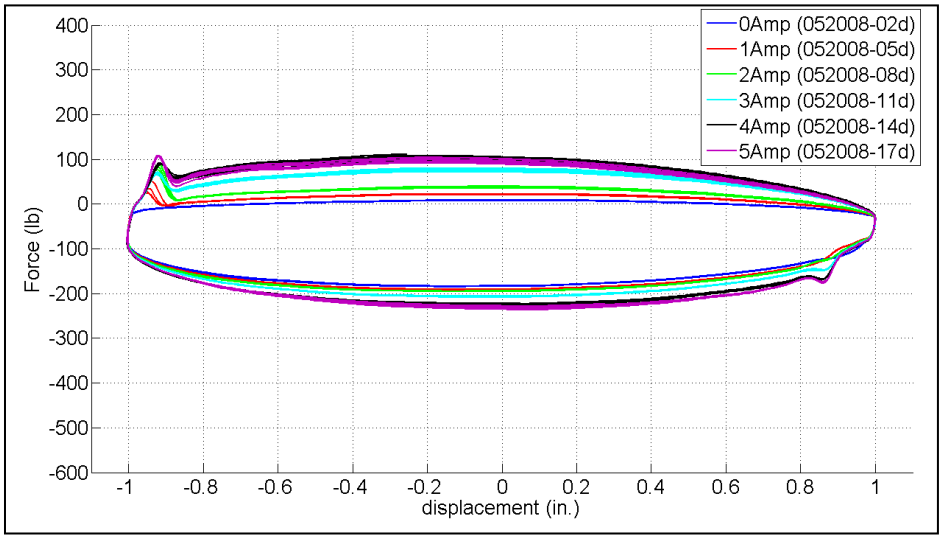

Figure 5-27 Force displacement curve, 0.5 Hz, 2 in. stroke

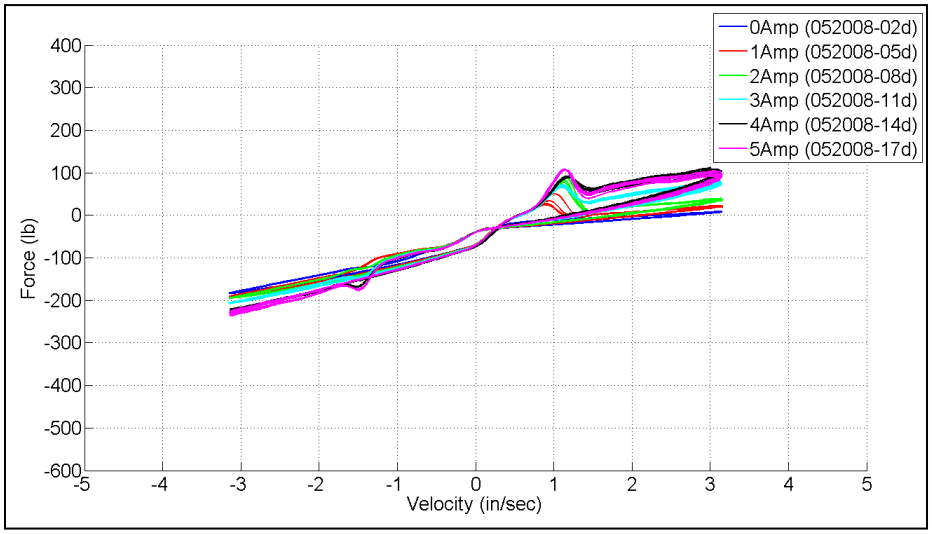

Figure 5-28 Force velocity curve, 0.5 Hz, 2 in. stroke

Figure 5-29 and Figure 5-30 show the test results for 1Hz. The force at 1Hz is higher than at 0.5 Hz because of higher damper speed. Also, as the current is increased the damper force is increased. This increase in force with increase in current is the desirable behavior of the MR hybrid damper. Figure 5-29 shows a 200 lb increase in the maximum force when the current is increased from 0 Amp to 5 Amp.

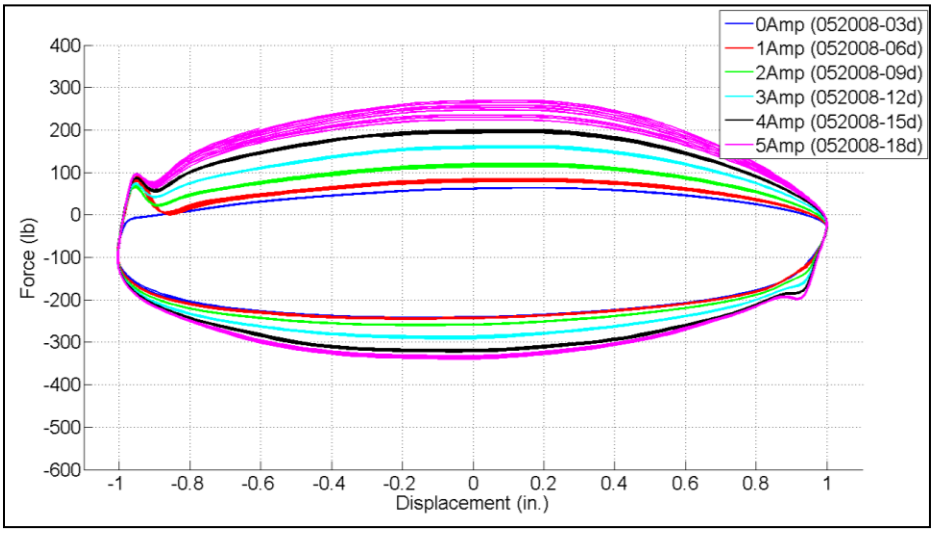

Figure 5-29 Force displacement curve, 1 Hz, 2 in. stroke

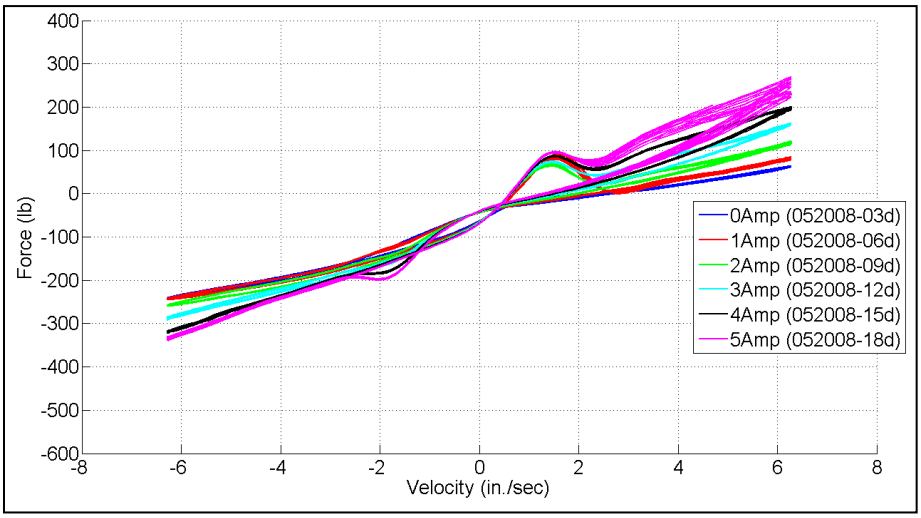

Figure 5-30 Force velocity curve, 1 Hz, 2 in. stroke

At higher frequencies, the separation in damping curves due to variable current was not seen. This is expected to be due to the high inertia of the MR spool that is dominating the spool performance at high frequencies. Figure 5-31 and Figure 5-32 show the damping curves for 1in peak-to-peak stroke at 3 Hz. The force is increased to a threshold and suddenly drops. The maximum peak value depends on the current. As Figure 5-31 shows, at 0 Amp there is no peak.

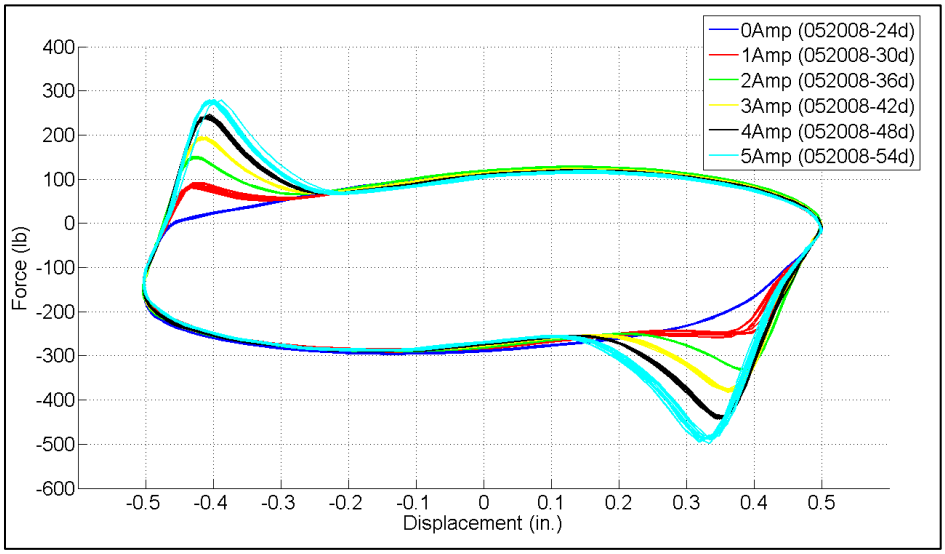

Figure 5-31 Force displacement curve, 1 in. stroke, 3 Hz

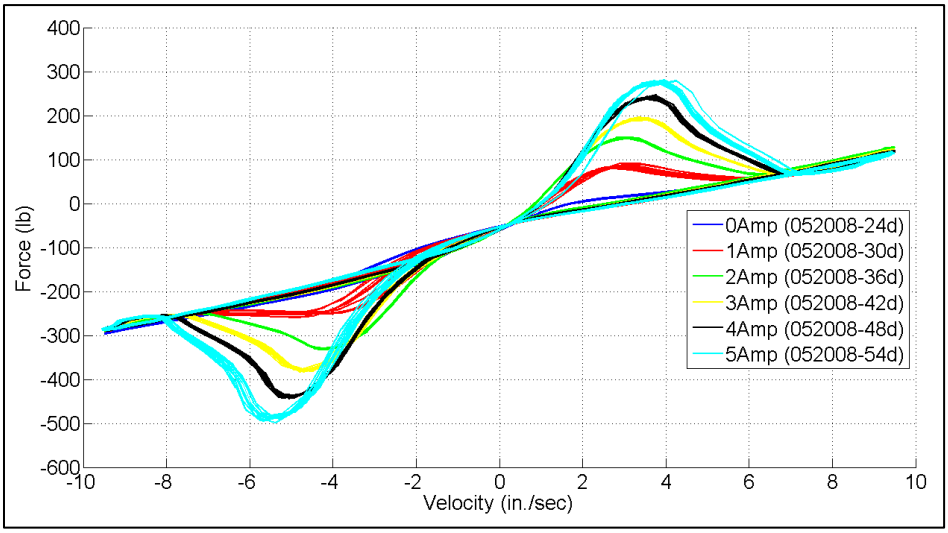

Figure 5-32 Force velocity curve, 1 in. stroke, 3 Hz

The peaks were found to be due to the magnetic attraction between the MR spool and piston cap when the current is on. As the current is increased, this magnetic attraction force is increased and a higher force is needed to separate the spool from the piston cap. This results in higher oil pressure and thus, higher damper forces. Once the MR spool is separated from the cap, the magnetic attraction suddenly drops and the damper force decreases. See Figure 5-33.

Another possible cause of the peaks is that at the end points, the MR spool lip is very close to the coil retainers and there is not enough space for the oil to get into and push the MR spool to the other direction as the damper reverses direction (Figure 5-33)

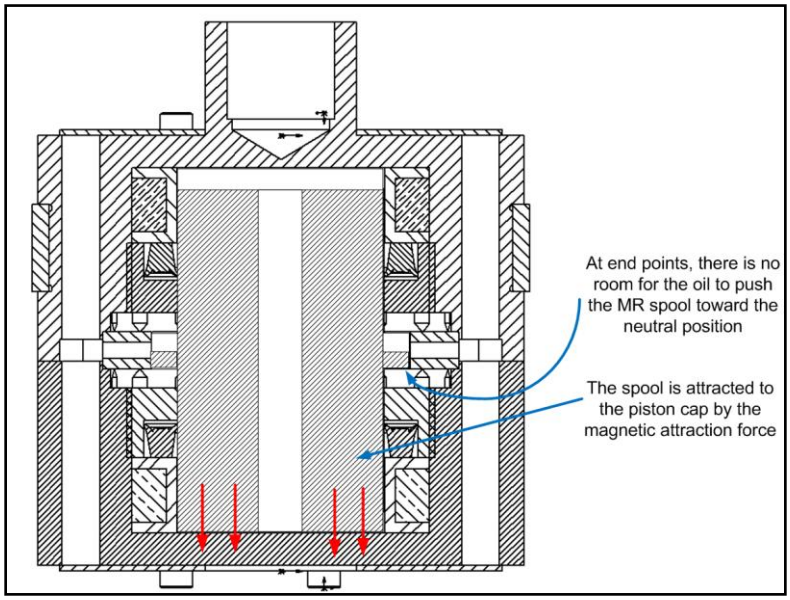

Figure 5-33 Sources of the peaks in damping curves

In order to solve the magnetic attraction problem, a thin gasket paper was added under each coil to raise the coil and coil retainer assembly. With that done, the MR spool hits the coil retainer first and stops before touching the cap surface. The magnetic attraction force is negligible as long as the MR spool has not touched the piston cap.

To address the second problem causing the peaks, the coil retainer was modified and a few grooves were added to it to ensure that the space under the MR spool lip is large enough for the oil to easily push the MR spool lip.

Another major issue in the tests was the problem of MR fluid leakage from the MR cavity and mixing with oil. After few cycles of testing, the MR fluid was mixed with the oil and the damper did not show any change in behavior with varying current. The surface finish of the MR spool was improved and tighter seals were also used. This made the damper capable of being used for longer periods of time, but it

still did not prevent the mixing of MR fluid and oil. In addition, adding tighter seals made the peaks in damping curves more pronounced. The friction between the MR spool and the dynamic seals was too high and moving the spool needed a lot of pressure and hence, damper force. Figure 5-34 shows the force displacement curve for 2 in. stroke and 1Hz. Peaks are seen in the damping curve because of high friction between the MR spool and the dynamic seals. As the oil pressure overcomes the friction force, the MR spool moves and the orifices open. Hence, pressure drops and the damping force is decreased rapidly, causing the peaks.

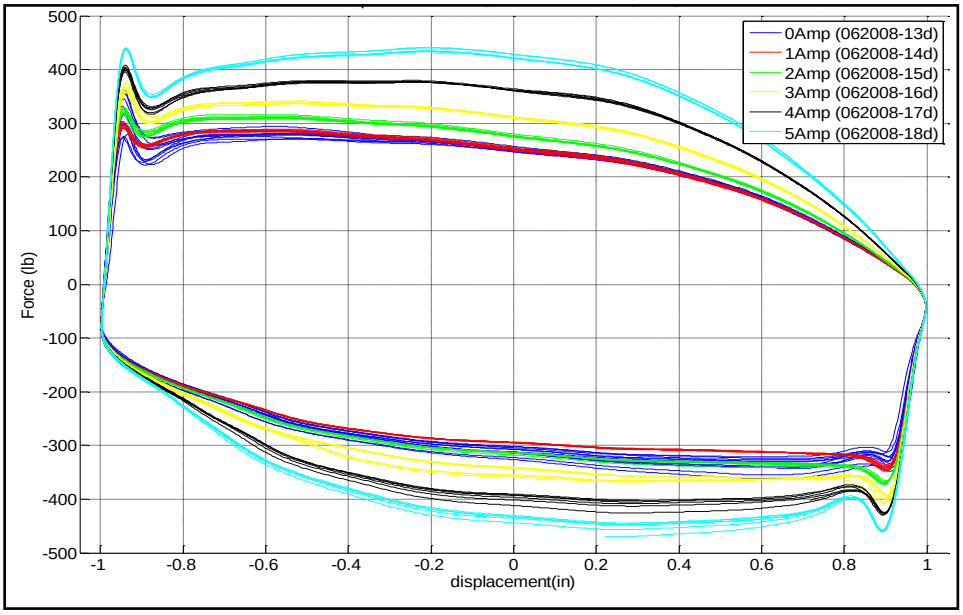

Figure 5-34 Force displacement curve, 2 in. stroke, 1 Hz, tight dynamic seals

In order to remove the sealing issue, a pouch design was introduced in the damper. With the new pouch design, the MR fluid was contained in a rubber pouch and there was no need for any kind of dynamic seal in the damper piston. The MR spool was modified according to the new pouch design. The new spool utilized small, nitrile MR pouches. Figure 5-35 shows a schematic and a photo of the new pouch design and modified spool.

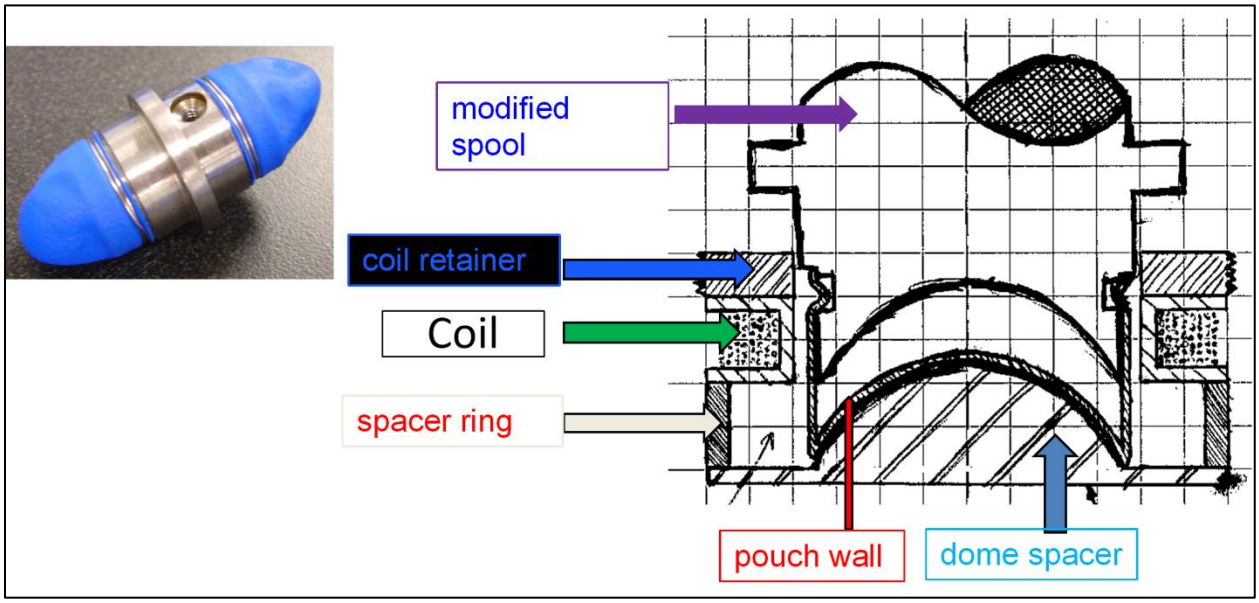

Figure 5-35 The new MR spool with the pouch design

In this design, the coil is moved towards the center ring creating a cavity under the coil for the rubber pouch to bulge out. The dome spacer geometry was used to ensure that the rubber will bulge outside the MRF squeezing space and into the cavity. Figure 5-21 shows the main components of the interior of the damper piston.

It should be noted that, in the pouch design, a rubber layer is added in the magnetic circuit. This increases the reluctance in the magnetic flux path and decreases the magnetic field strength.

Figure 5-36 shows a magnetic circuit analysis of the damper piston with the pouch design at maximum current of 5 Amp. The maximum magnetic field density is approximately 0.4 Tesla in this design. However, the previous design (without the pouch) had a maximum magnetic field density of 0.7 Tesla. In addition, the previous designs had a homogeneous magnetic field along the MR spool radius. But, the pouch design does not have a constant magnetic field density in the MR cavity because of the curvature of the MR spool.

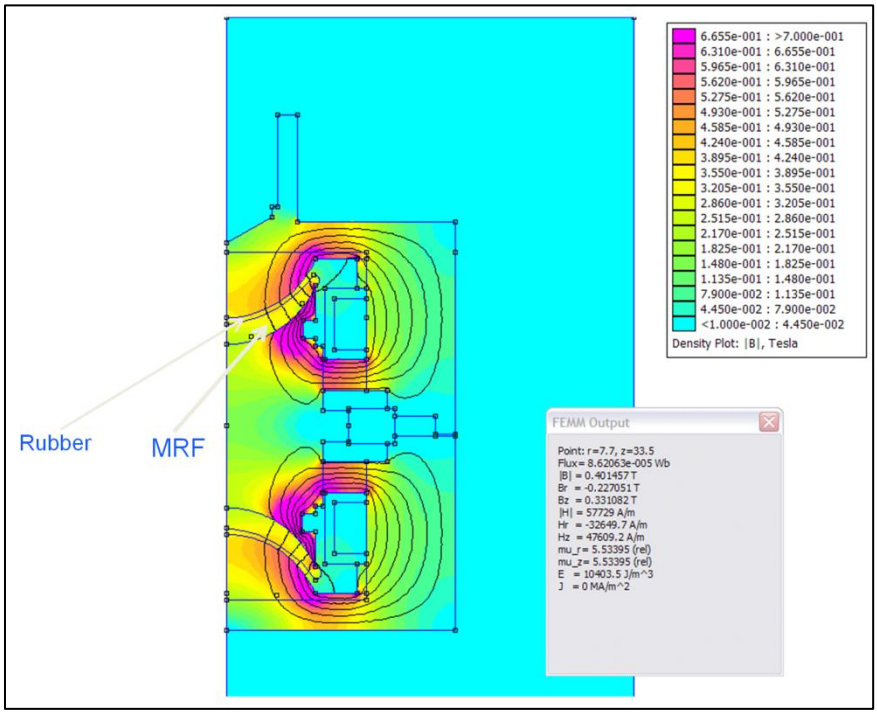

Figure 5-36 FEMM analysis of the damper piston at 5Amp

Tests were carried out with the new pouch design piston at different frequencies and strokes. Figure 5-37 and Figure 5-38 show the damping curves for 2 in. stroke, 1 Hz tests. As can be seen, because there are no dynamic seals in the piston, the curves are very smooth and no peaks are seen in the test results. As the current is increased, the maximum force is increased. This shows the controllability of the damper.

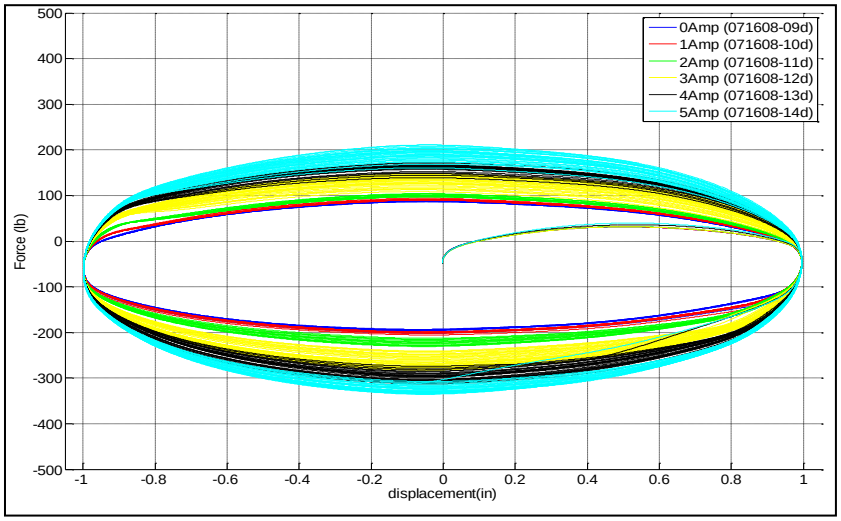

Figure 5-37 Force displacement curve, 2 in. stroke, 0.5 Hz

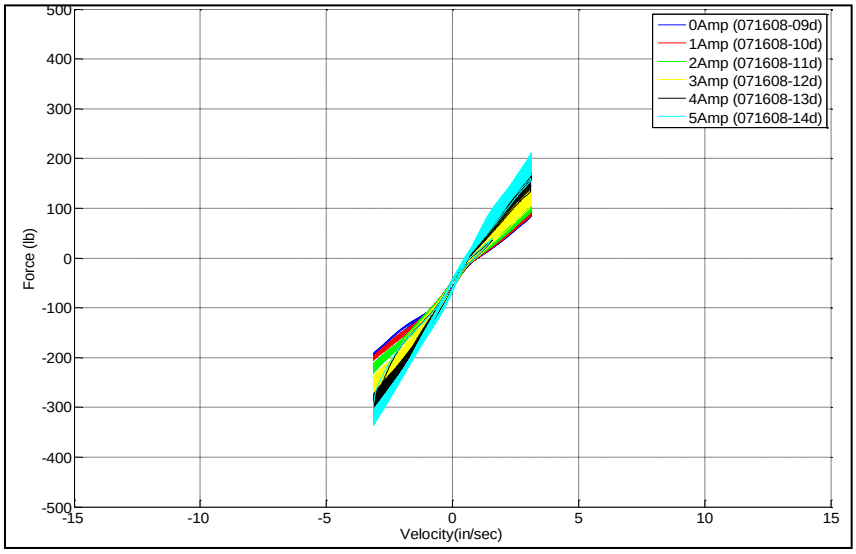

Figure 5-38 Force velocity curve, 2 in. stroke, 0.5 Hz

Figure 5-39 and Figure 5-40 show the test results at 1Hz. During the test it was observed that the force increased at each cycle at a constant current (Figure 5-39). The damper force should be constant at each value of current. This increase in force at constant currents is suspected to be due to the clumping behavior of the MR fluid, as explained in the rheometer tests chapter, but this is not fully understood.

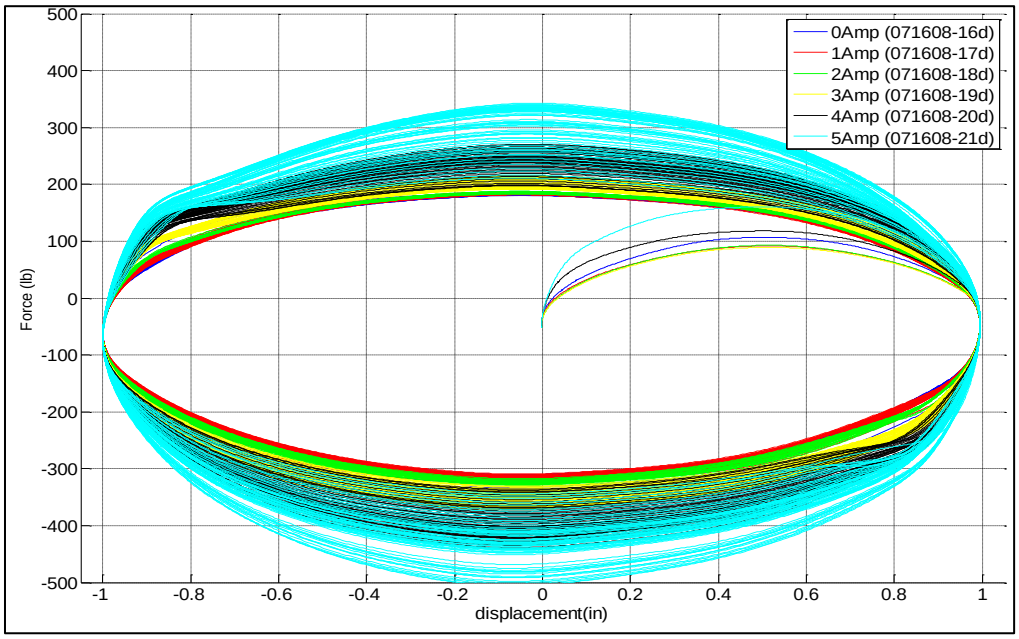

Figure 5-39 Force displacement curve, 2 in. stroke, 1 Hz

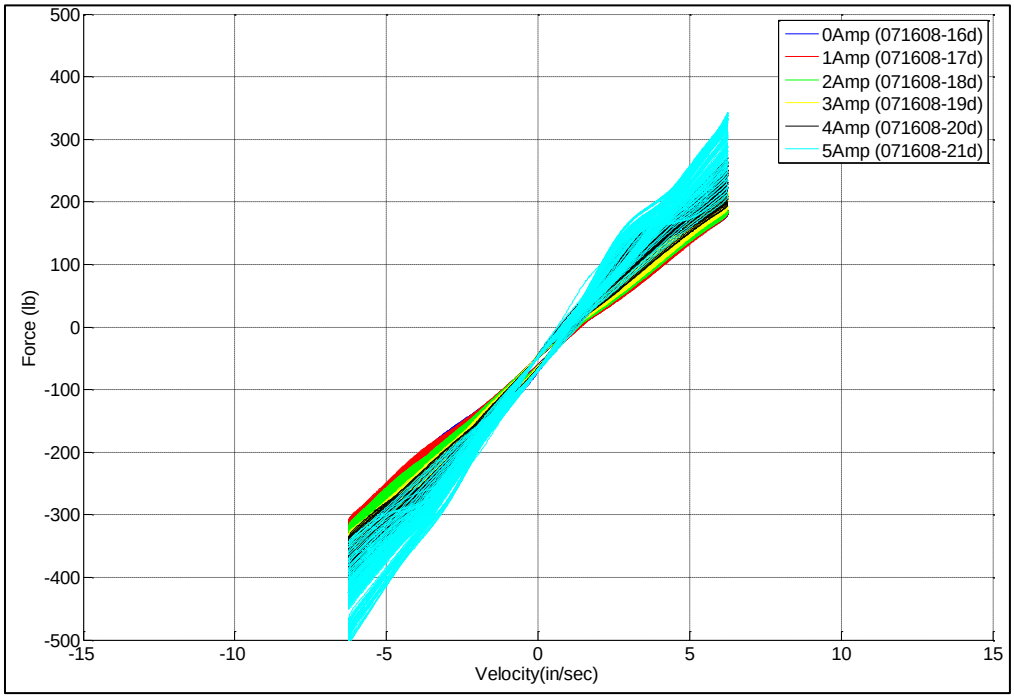

Figure 5-40 Force velocity curve, 2 in. stroke, 1 Hz

Test results in Figure 5-39and Figure 5-40 show that the hybrid damper could deliver a controllable force range based on the amount of electric current passing to the MR circuit. Although there were some practical issues encountered during testing of this MR hybrid damper, this primary prototype could be considered as a proof of concept for a new class of smart dampers.

5.3. MR Squeeze Mount

This section will present design and test results of the MR squeeze mounts as a second application of MR fluid squeeze flow. These mounts can be used in a wide variety of applications that need a large range of controllable force in small operating envelopes. Engine mounts in vehicles can be a good example. In this section, design details and testing procedures are presented. This work considers *quasi-static* tests only. Full dynamic testing is required to evaluate the performance of MR squeeze mounts before they can be confidently used as engine mounts. Dynamic testing and evaluations are the topic of another research project currently in progress at CVeSS.

For the initial squeeze mode MR fluid testing, as discussed in previous chapters, the rheometer fluid cavity was filled with a set amount of fluid (according to the desired initial gap) and then the piston was inserted and the tests were performed. In these tests, the volume of the MR fluid in the activation region was changing based on the position of the piston in the rheometer. MR fluid could escape to a hole drilled in the middle of the piston, accommodating this volume change. When this configuration was implemented in the MR hybrid damper, some problems with the sealing made assembly and testing very challenging. These difficulties led to the idea of an MR pouch (mount)[†] as a separate, modular unit that can be implemented more easily and, at the same time, solve the MR fluid leakage problem.

Different designs were tried and finally the two-disk pouch design concept was developed. Different design iterations are presented as follows.

5.3.1 Gen. I MR Pouch

The two disk MR pouch consists of two steel disks and a rubber embodiment that is sealed permanently to the disks. MR fluid was filled in the pouch and the filling hole was closed and sealed. Figure 5-41 shows the first generation of the MR pouch.

[†] In this document, the terms “MR mount” and “MR pouch” are used interchangeably.

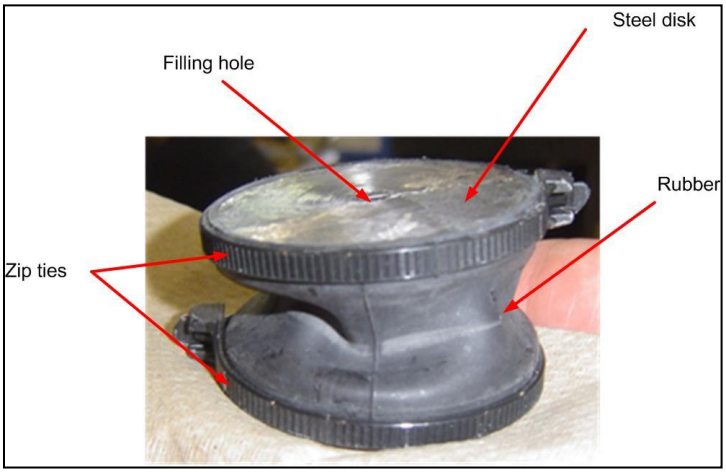

Figure 5-41 Gen. I MR pouch

The first generation of MR pouch was tested in the squeeze mode rheometer using a similar test procedure used for MR fluid bath tests. Tests results were very encouraging. The pouch, as a self-sealed and modular unit, was able to deliver a very large range of force. However, the Gen. I MR pouch suffered from two issues. The first issue was the non-uniform shape of the rubber. When the pouch was at its free length, the rubber had an inward, concave shape. As the pouch was squeezed during testing, the rubber was pressurized and at a certain point, bulged out suddenly. This sudden change resulted in a rapid change in the force. The second problem was the attachment of the rubber to the disks. Zip-ties and epoxy were used to attach and seal the rubber to the steel disks. As the pouch was squeezed, the rubber under the ratcheting mechanism of the zip-tie was not properly supported, and would result in a leak. Additionally, the sharp ends of the zip-ties pinched the rubber especially at small gap sizes. This issue severely reduced the life of the pouch.

5.3.2 Gen. II MR Pouch

To improve the life and performance of the pouch, each disk was machined to have two small grooves on the circumference. Instead of zip-ties and epoxy, safety wire was used to attach the rubber to disks. This design proved to be very well sealed and surprisingly durable. Figure 5-42 shows a picture of the second

generation of the MR pouch. The ends of the wires were covered with epoxy to prevent puncturing of the rubber during the tests.

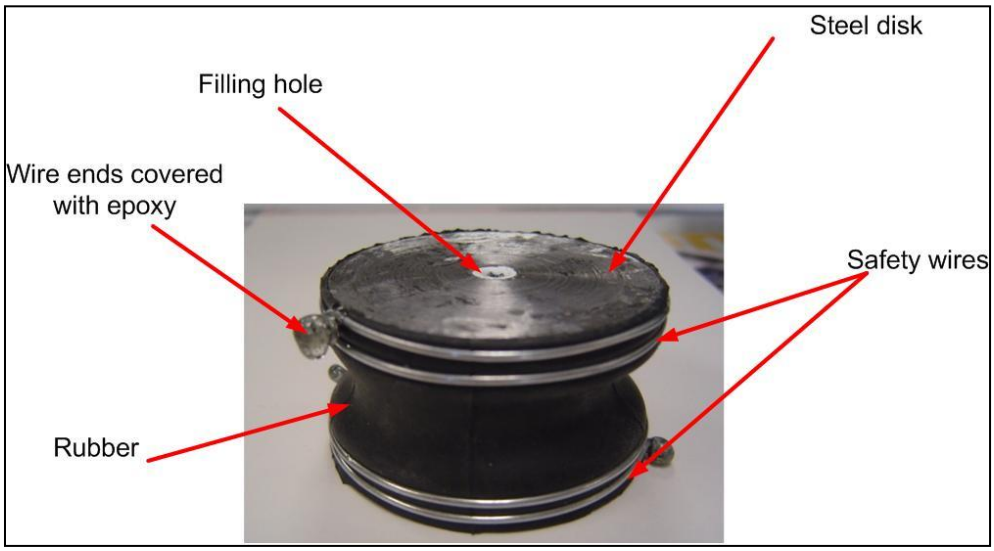

Figure 5-42 Gen. II MR pouch

Although the MR pouch had very good durability in terms of leakage, the rubber still bulged inwards and the sudden drops in force (due to change in shape of the rubber from bulge-in to bulge-out) were observed in test results. Figure 5-43 shows the force-displacement curves of the Gen. II MR pouch test results. The pouch was squeezed from an initial gap of 0.3 in. to 0.1 in. As can be seen, the pouch is capable of delivering over 1000 lb of force at 2.5 Amp. The pouch was not tested at higher currents to prevent possible damage to the rheometer. This force range is very promising for a pouch with only 1.75 in. diameter. The pouch can be used in a large variety of applications because of small size, high force to stroke ratio, and its controllability.

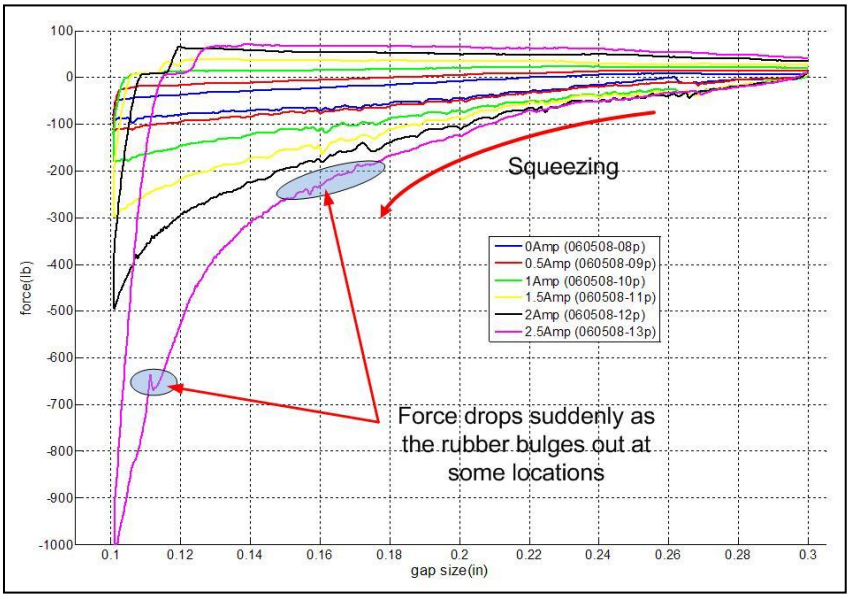

Figure 5-43 Force-displacement curve, Gen. II MR pouch, gap size 0.3 in. to 0.1 in.

5.3.3 Gen. III MR Pouch

As discussed, the first and second generations of MR pouch suffered from the irregular shape of rubber. To solve this problem, a new MR pouch design was proposed. Due to the special design of the pouch, the rubber is always bulging out and has a uniform and regular shape. Figure 5-44 shows the Gen. III MR pouch in its static and compressed configurations. As can be seen, the rubber completely bulges out.

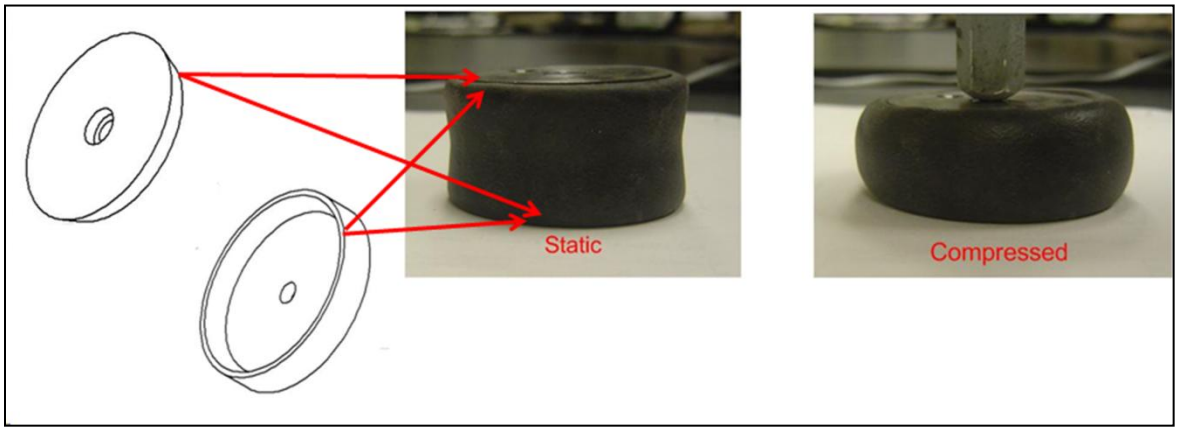

Figure 5-44 Gen. III MR pouch

In this design, each side consists of two parts that are connected to each other by a screw. The rubber is clamped in between the two parts and the screw is tightened, sealing the pouch. The drawings of the MR pouch parts are shown in Figure 5-45 and Figure 5-46.

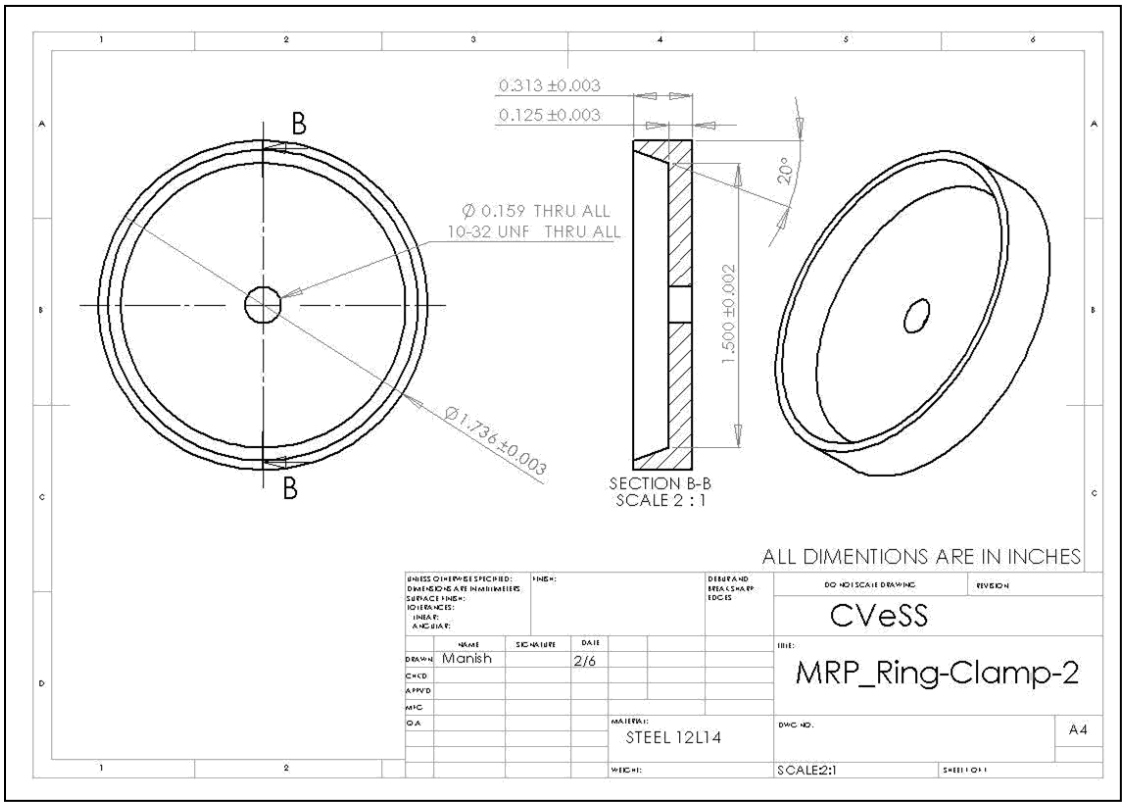

Figure 5-45 Gen. III MR pouch cap, inner part

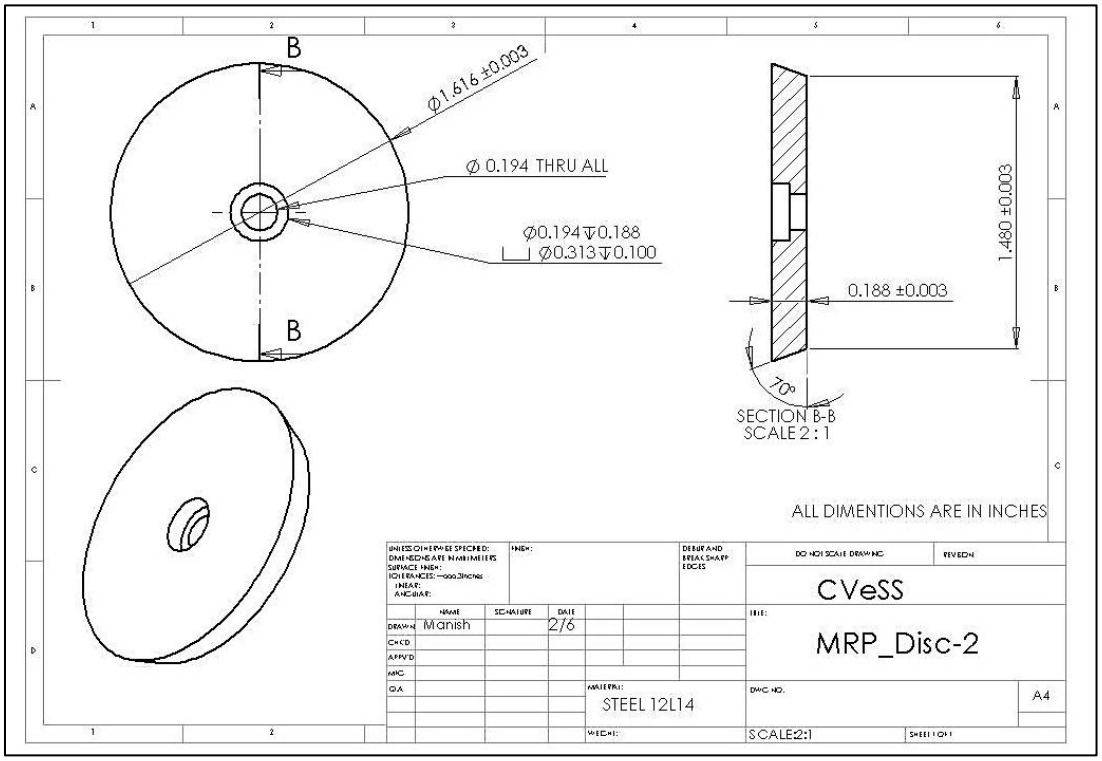

Figure 5-46 Gen. III MR pouch, outer part

The test results of the Gen. III MR pouch showed very smooth curves due to the elimination of the bulge-in to bulge-out behavior, as shown in Figure 5-47. The rubber always bulges out and does not interfere with the squeezing operation.

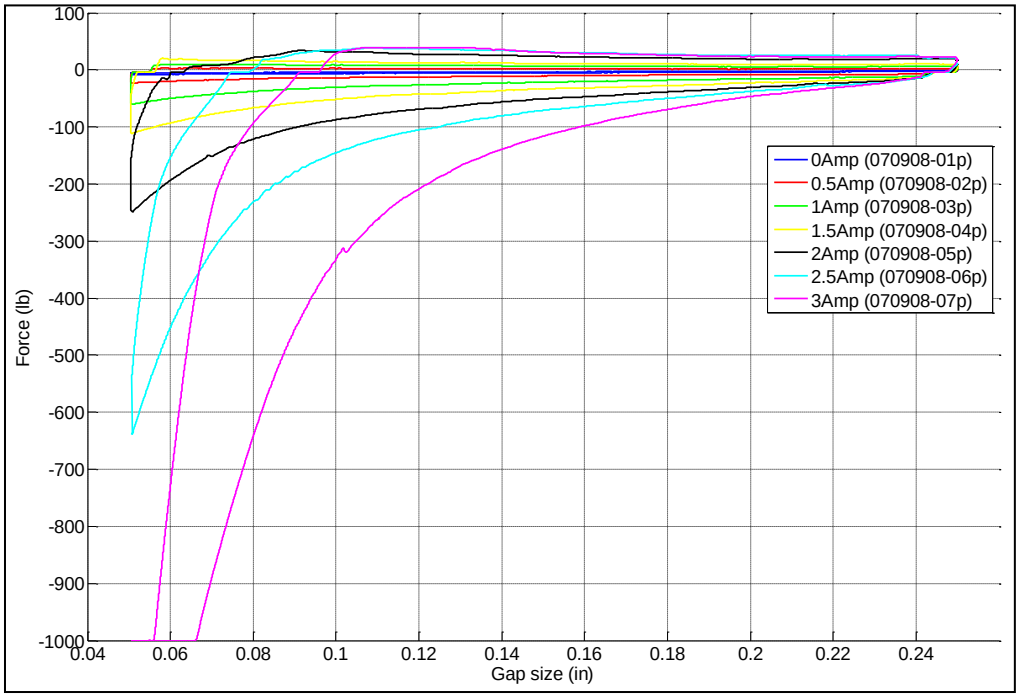

Figure 5-47 Force-displacement curve, Gen. III MR pouch, gap size 0.25 in. to 0.05 in.

The Gen. III MR pouch has been the best generation of MR pouches that has been built at CVeSS so far. It is the easiest to assemble, is very durable, has very high load capacity, and smooth behavior due to the absence of the rubber buckling.

MR fluids used for pouch testing are oil-based MR fluids. The rubber is also made from oil-based material. It was observed that the MR fluid and rubber were reacting with one another over time. This reaction changed the physical properties of the MR pouch during testing. Figure 5-48 shows the rubber before and after reacting with MR fluid. As can be seen, the shape of the rubber has completely changed. It was observed that the rubber became softer after reacting with MR fluid.

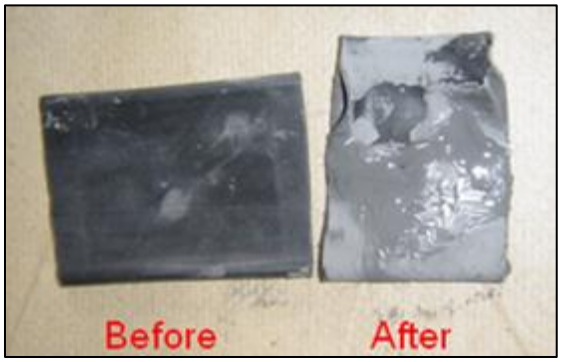

Figure 5-48 Pouch rubber before and after reaction with MR fluid

This material inconsistency problem was studied and a new rubber was made from polyurethane (Polytek 74-30 RTV). This material does not react with the MR fluid and can be used for long term operations of the MR pouch without any physical or chemical changes. A pouch membrane made of this material is shown in Figure 5-49.

Figure 5-49 Polyurethane rubber made for use in MR pouches

A rotary mold was made to fabricate the thin-walled membrane. To make a membrane, a finite amount of the mixed, two-part polyurethane was added to a cylinder with sealing end caps. This mold was then spun at high speed until the rubber was set. After some time, the rotary mold was stopped and the membrane was left to cure. The membrane thickness can easily be adjusted with the rotary mold, although the thickness range of interest was 0.010-0.030 in.

5.3.4 Magnetic Simulations of the MR Mount

The initial experimental test results presented in Chapter 4 were based on the *"MR fluid pool"* in the rheometer. In these tests, MR fluid was directly poured into the rheometer cavity. After the MR pouch idea was developed, the same rheometer was used for testing. Due to the large size of the pouch (thick pole plates plus MR gap in the pouch), the rheometer piston was set at a higher position to start testing. As a result of this increase in rheometer cavity, the magnetic analysis had to be performed again to assure that the MR mount is being activated the same way as the initial MR fluid pool tests. The main concern was the homogeneity of the magnetic field density in the pouch gap. The magnetic field density should be homogenous in the MR fluid gap to ensure symmetry and also simplify data analysis and mathematical modeling. Figure 5-50 shows the magnetic field density distribution in the MR mount at an electric current value of 3 Amps and a pouch gap size of 0.25 in.

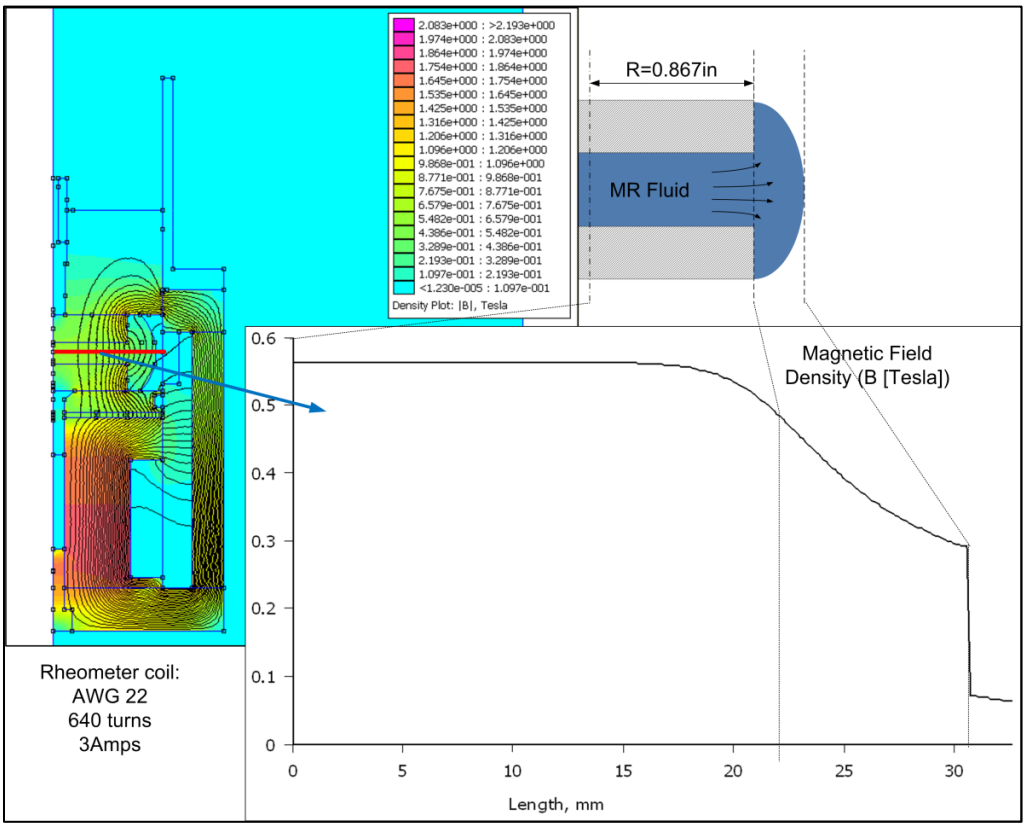

Figure 5-50 Magnetic analysis of the MR mount in the squeeze mode rheometer as 3 Amps and pouch gas size of 0.25 in.

As shown in Figure 5-50, the magnetic field distribution in the pouch is homogeneous and has a constant value along the radius. This magnetic analysis was carried out for other values of current and gap sizes and a similar trend was observed. Therefore, it is assured that the magnetic field density in the pouch is homogenous. However, it should be noted that due to the larger rheometer cavity compared to MR fluid pool experiments, the average magnetic field density is smaller in MR pouch experiments.

5.3.5 MR Mount Experimental Testing

In order to provide a magnetic field in the fluid gap of the MR mount, it is placed in the squeeze mode rheometer introduced in Chapter 4.

5.3.5.1 Test Procedure

Before the start of the experiments, the MR fluid is mixed thoroughly and de-gassed by being placed in a vacuum chamber for 15 to 20 minutes. After the MR fluid is de-gassed and the amount of air in the fluid is minimal, the pouch is filled with MR fluid through a filling hole in the upper plate assembly. The process of filling the pouch should be performed very carefully and slowly so that the amount of air introduced in the pouch is minimized. After the pouch is filled with enough MR fluid depending on the desired free height of the pouch, the hole is blocked using a screw. At this point the pouch is ready to be tested. The test procedure is as follows. First, the pouch is placed in the rheometer cavity and the initial desired gap is adjusted. Then the electric power supply is turned on and the desired electric current is adjusted. Finally, the piston starts to move down and squeezes the pouch.

Setting the initial gap is a very important step of the testing process. In order to make sure that the gap size readings are most accurate, the piston is moved downward into the rheometer cavity until the pouch upper and lower plated contact each other (no magnetic field is applied at this time). Using the displacement control feature of the MTS load frame, the piston is lowered until the force reading is approximately 50 lb due to contact of the pouch plates. At this point, the MR pouch gap size reading is set to zero. Then, using the MTS control module, the piston is moved until the desired pouch gap size is

obtained. At this point, the hydraulic controller and the data acquisition system can be activated to start the test.

There are two ways of actuating the MTS load frame. One way is to control the displacement. This method is called the "displacement-controlled" method. Alternatively, the force can be controlled. The results that will be presented in this paper are based on the later approach, "force-controlled" mode. After the initial gap is adjusted, the force is increased from 0lb to a certain desired value at a steady rate. The rate of increase in force should be slow to ensure the quasi-static condition. Here, the force was increased from 0 lb to 300 lb with a rate of 2 lb/sec.

5.3.5.2 Magnetic Pull Force

When electric current is passed through the rheometer coil, the magnetic field exerts a force on the rheometer piston and tries to pull the piston into the rheometer cavity. This *"Magnetic Pull Force"* is in the opposite direction of the squeeze force. As a result, the force sensor on the MTS test frame reads a smaller force that the actual squeeze force developed by the MR mount. This magnetic pull force should be considered when studying MR mount force curves. The magnetic pull force depends on the magnetic field density (electric current passing through the rheometer coil). At 0 Amp, there is not magnetic pull force and as the electric current is increased, the magnetic pull force is increased. To measure this magnetic pull force, the rheometer is tested without any MR mount in the rheometer as various values of current. The force reading from the force sensor is the magnetic force combined with friction. They are fitted with an exponential function so that they can be used along with test data obtained from the MR mount experiments. The form of this exponential function is

$$MPF = ae^{bh} + ce^{dh} \tag{5-1}$$

where, *a*, *b*, *c*, and *d* are constants and depend on the magnetic field density (or electric current). *h* is the rheometer gap (when no mount is in the rheometer) corresponding to the mount gap for the squeeze tests.

Table 5-1 Magnetic pull force and Coulomb friction parameters for MR mount tests in squeeze mode rheometer

Current (Amp)	*a (lb)*	*b (1/in.)*	*c (lb)*	*d (1/in.)*	*Coulomb friction (lb)*
1	234.1	-54.16	23.92	-9.23	10
2	202.3	-25.56	42.28	-6.47	20
3	187.6	-21.75	75.37	-6.19	25
4	200.4	-21.91	87.02	-5.64	25

Another effect of the magnetic field on the motion of the rheometer piston is its influence on the friction between the rheometer piston and rheometer body. As the magnetic field density is increased, this friction is increased. At 0 Amp, this friction is very low (approx. 3 lb) and at the highest magnetic field density can be as large as 25 lb. This magnetic field density dependent friction force should also be considered in the study of the squeeze force in addition to the magnetic pull force.

5.3.5.3 Test Results

Figure 5-51 shows the test results for a forced-controlled experiment of the MR mount. MRF-132DG was used to fill the MR mount. The initial gap was set to 0.25 in. and the force was increased from 0 lb to 300 lb at a constant rate and decreased back to 0 lb at the same rate. As mentioned in previous section, although the force measured by the sensor was increased from 0 lb to 300 lb, the magnetic field exerted a tension force on the rheometer piston depending on the magnetic field density and gap size. This "magnetic pull force" was included in the plot shown in Figure 5-51. For 0 Amp, there was no magnetic pull force and the actual squeeze force increased from 0 lb to 300 lb.

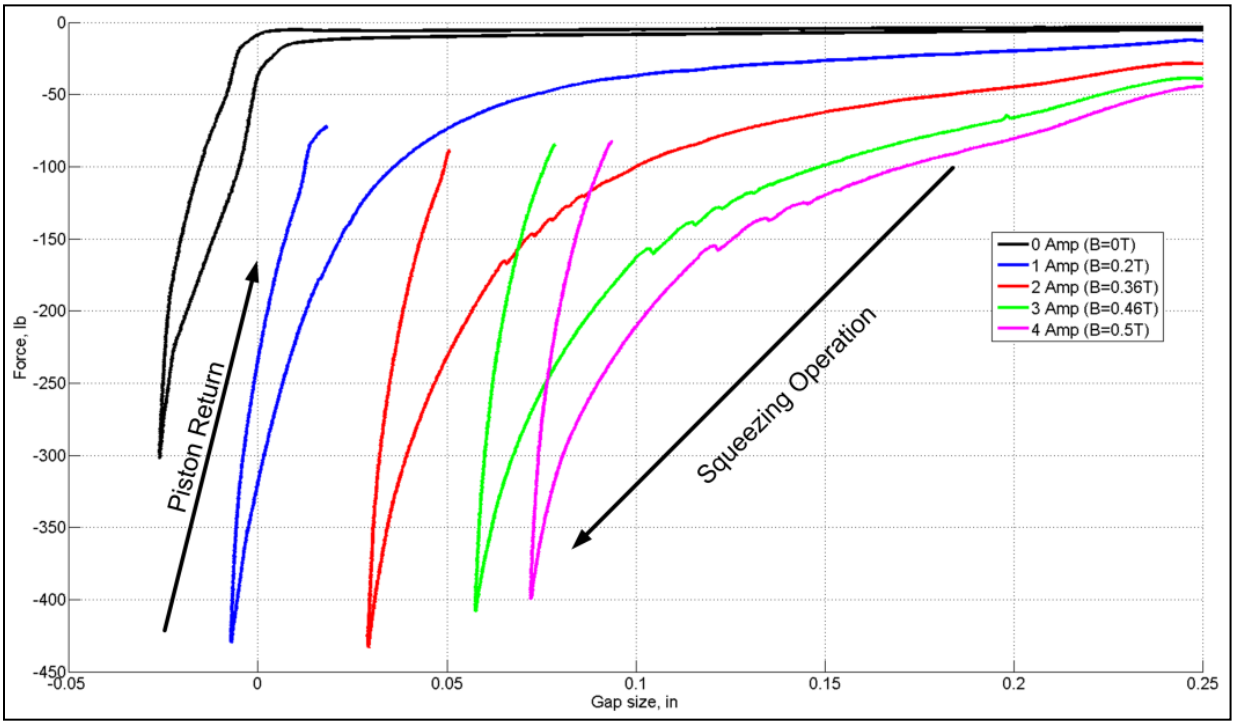

Figure 5-51. Experimental data for squeeze tests of MR pouch under force-controlled mode

The experimental results show that the pole displacement varies depending on the amount of electric current for a desired force range. As the magnetic field increases, the MR fluid resistance increases and the pouch pole plate moves a smaller stroke. At no magnetic field (0 Amp), the pouch gap size approaches zero and the rubber clamped between the pole plates is squeezed allowing for some more movement of the rheometer piston which is shown as negative displacement in Figure 5-51. This shows that the off-state viscous effects are small compared to magnetic properties of the MR fluids and the resistance due to change in the rheology of the fluid is the dominant factor. After the maximum force (minimum gap) is reached, the force is decreased back to zero and the rheometer piston returns upward. It is important to note that when the force is decreasing back to zero, the rheometer piston moves a much shorter distance compared to the compression stroke as shown in Figure 5-51. This shows the inability of MR fluid to push back the piston upward and return it to the initial starting position. Therefore, it is concluded that MR fluid needs to be in compression in order to deliver its largest controllable force capacity.

Another behavior shown in Figure 5-51 is the small difference between the 3 Amp and 4 Amp curves. This is because at 3 Amp, the rheometer magnetic core approaches its magnetic saturation limit. As a result of magnetic saturation, rheological properties of MR fluid will not change significantly as the electric current is increased.

5.3.6 Mathematical Model Validation

Experimental data presented in Section 5.3.5.3 is used to validate the mathematical models developed in Chapter 3. In Chapter 3, two different mathematical models were presented for the MR squeeze mount. The first mathematical model was a simple model and could only predict the total squeezing force. The second model was developed using perturbation techniques and was capable of predicting flow field, shear rate distribution, and pressure distribution in addition to the total squeezing force. This section is devoted to validation of these two models using test data obtained from MR squeeze mount experiments discussed in previous section.

5.3.6.1 Validation of the First Model

The first mathematical model developed in Chapter 3 for the MR mount can be validated using displacement-controlled test results presented in Figure 5-47. While this model accounts for the viscous and MR force as related to the height of the gap, a relationship between h and spring force of the pouch assembly was also necessary. This spring force is due to the pressure from the stretching of the elastic membrane as the pouch is compressed. This was found empirically when $\tau_y = 0$ and determined to be $F_{membrane} = -26h + 9$ for the set of data shown in Figure 5-47. F is in pounds and h is in inches. The membrane force is dependent on the membrane material type, thickness, and free length. It should also be noted that h in this simple mode is the gap size. But in the perturbation model, h represents half of the gap size.

Using the total squeeze force equation for this model, the behavior of the MR squeeze mode force element was plotted and compared against test data. The value for $\tau_{y0} = 0$ was found based on the flux density, *B*, in the experiments and correlating this with data from the manufacturer of the MR fluid to find the shear stress. η was found from the manufacturer data and K_H was found from correlating test data and mathematical model. The comparison of some pouch testing can be seen in Figure 5-52.

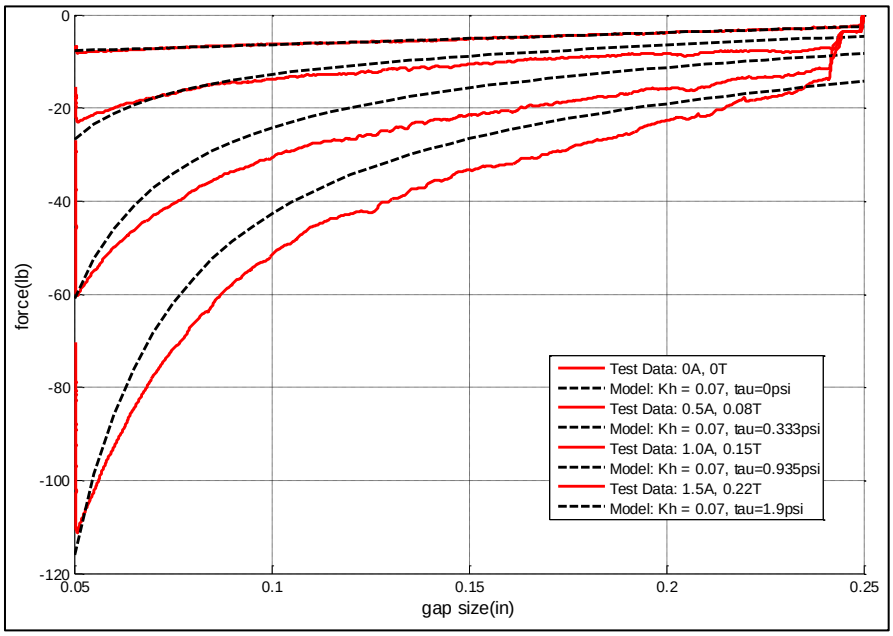

Figure 5-52 Test data and model correlation for the simple mathematical model developed in Chapter 3

While the comparisons are within reason, the model begins to deviate from the testing as the coil current is increased. This may be due to a nonlinear relationship of K_H and the flux density, fluid clumping during testing, or a nonlinear behavior of the pouch spring rate in relation to the coil current. This deviation is seen considerably less by using the perturbation model and force-controlled test data as will be discussed in next section.

5.3.6.2 Validation of the Perturbation Model

The perturbation model for predicting the behavior of the MR squeeze mount was introduced in detail in Chapter 3. This model was developed in non-dimensional form. In this section, this mathematical

model is used in its dimensional form using parameter definitions introduced in Chapter 3. Figure 5-53 shows a comparison between test data and model prediction at 0 Amp. As can be seen, when there is no magnetic current, the pouch gets squeezed easily and there is little resistance from MR fluid until the pole plates contact each other. After the pole plates have contacted, the membrane between the clamps is compressed elastically and displacements measurements show negative values.

Figure 5-53 is used to find the effect of the pouch membrane stiffness on the total squeezing force before the pole plated contact each other. Similar to the previous section, it was found that the force due to membrane stiffness obeys

$$F_{membrane} = -52h + 9 \quad lb \tag{5-2}$$

where, h is half of the gap. As shown in Figure 5-53, the membrane stiffness effects are very small compared to the large squeezing forces obtained at non-zero magnetic fields (Figure 5-51).

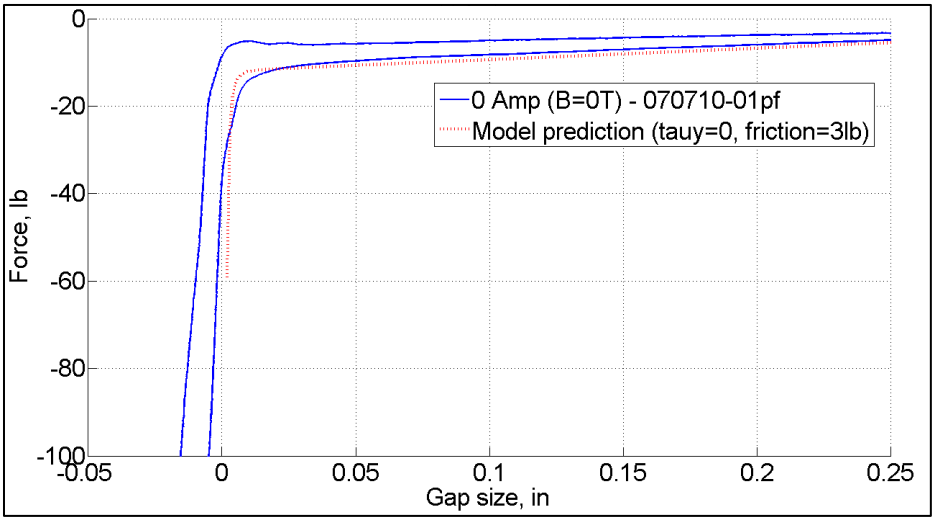

Figure 5-53. Test results and model prediction at no electric current

Figure 5-54 shows a comparison of squeeze force between the model and test data for different rheometer coil currents. The compression portion of tests is being considered only.

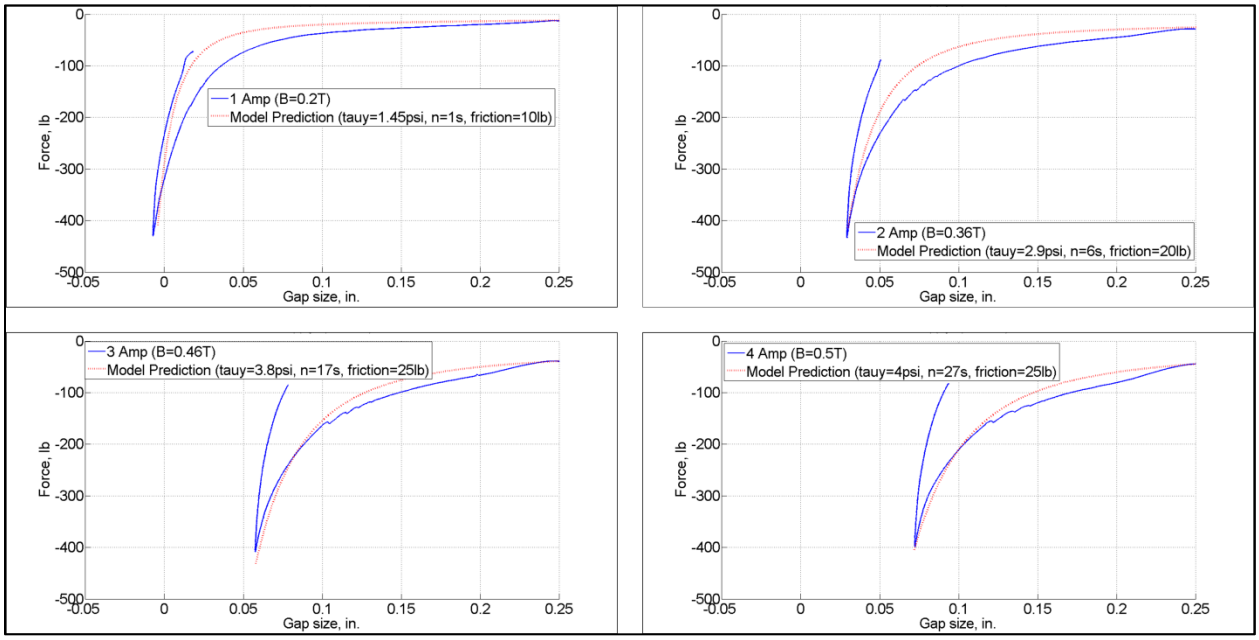

Figure 5-54. Comparison between test data and mathematical model (solid lines correspond to test data and dashed lines to model predictions)

As can be seen, the model and test data agree very well. The value for yield stress used in the model was found based on the flux density, *B*, measured by the magnetic probe in the experiments and correlating this with data from the manufacturer of the MR fluid to find the shear stress. The off-state viscosity, η_0, was found from the manufacturer data and the value of the material parameter, *n*, in Papanastasiou model was obtained from correlating the model with test data. The friction between the piston and rheometer housing was also included in model predictions in Figure 5-54. As the electric current increases, the magnetic force between the piston and rheometer housing increases causing higher Coulomb friction forces. Table 5-2 lists numerical values obtained from model validation.

Table 5-2. Parameters of Papanastasiou model. Yield stress and off-state viscosity values were obtained from data provided by Lord. *n* was obtined from model validation

Current value	*B* (T)	η_0 (Re)	τ_y (psi)	*n* (s)
0 Amp	0	6.09e-6	0	0
1 Amp	0.2	6.09e-6	1.45	1
2 Amp	0.36	6.09e-6	2.9	6
3 Amp	0.46	6.09e-6	3.8	17

Current value	B (T)	η_0 (Re)	τ_y (psi)	n (s)
4 Amp	0.5	6.09e-6	4	27

As shown in Table 5-2, as the magnetic field density increases, values of the yield stress and n increase. At a magnetic field density of 0T, MR fluid behaves as a Newtonian fluid with no yield stress. As the magnetic field is increased, MR fluid switches behavior from a Newtonian fluid to a Bingham fluid with a higher pre-yield viscosity ($\eta_0 + n\tau_y$). As the value of n increases, this change in behavior occurs at lower shear rates ($1/n$). Using the numerical values in Table 5-2, the stress-shear rate relationship for the MR fluid can be obtained using Papanastasiou model (Chapter 3) as shown in Figure 5-55.

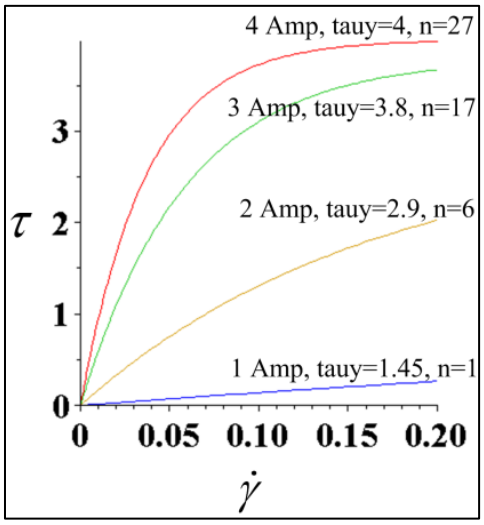

Figure 5-55. Papanastasiou model for various magnetic field densities using numerical values of Table 5-2.

5.4. Clumping Effect Investigations

As previously mentioned, it was observed in squeeze tests that after each test run, the force is increased. This so-called, *"Clumping Behavior"* was introduced in Chapter 4. The clumping effect manifests itself in particle aggregates that can be seen after few test cycles. Because there is very little fluid flow in squeeze mode, MR fluid cannot be mixed during experiment and during each test cycle, iron particles are trapped in the magnetic field while the carried oil flows outside the magnetic field easily.

Figure 5-56 shows a time history of a 5 cycle clumping test on the MR pouch. The pouch filled with well-mixed MR fluid (MRF-132DG) was placed in the rheometer cavity and the piston was moved in a quasi-static manner for 5 cycles from an initial gap size of 0.25 in. to a minimum gap size of 0.1 in. As can be seen in Figure 5-56, the force is considerably increased after each cycle and reaches to 100 lb after the fifth cycle. Figure 5-57 shows the force-gap size curve for the same clumping experiment.

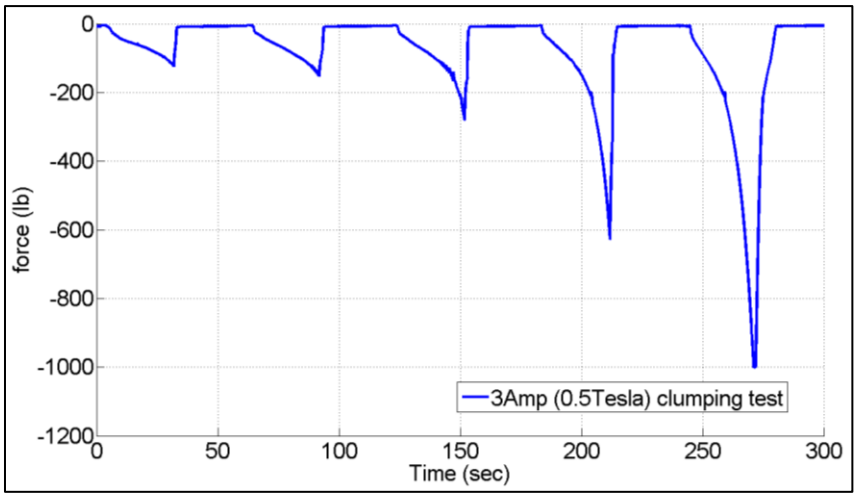

Figure 5-56 Clumping test of 5 cycles on the MR mount at 3 Amps

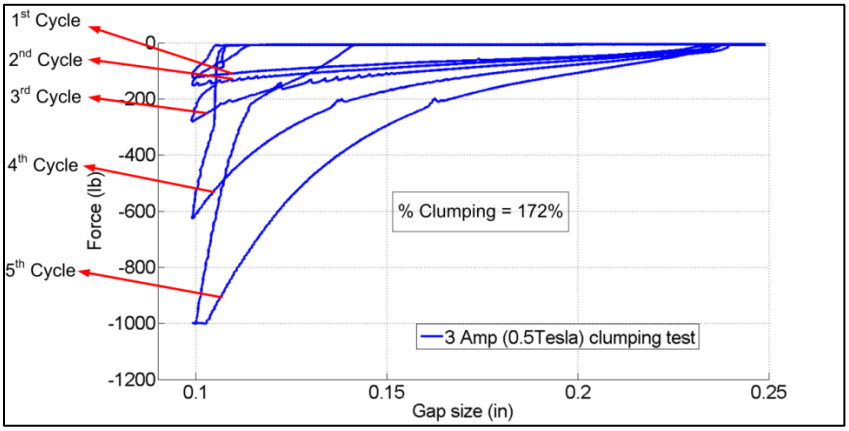

Figure 5-57 Force vs. gap size curves for clumping experiments on MR pouch

Clumping behavior significantly reduces repeatability of experiments. This behavior needs to be minimized in order for the squeeze mode to be used as an effective flow mode in MR devices. The major challenge is that in squeeze mode, there is little to no flow of MR fluid and MR fluid cannot be re-mixed

during operation similar to conventional shear or valve modes. MR fluid in squeeze mode has to be mixed *"Non-Mechanically"* via some other means. In this section, some ways of improving the clumping behavior is studied and possible future improvement techniques are presented.

In order to have a quantitative measure of the clumping behavior in squeeze flow, *"Percentage Clumping"* is defined as

$$\%\ \text{Clumping} = \frac{\text{Avg. squeeze force per inch squeeze}\big|_{\text{Last Cycle}} - \text{Avg. squeeze force per inch squeeze}\big|_{\text{First Cycle}}}{\text{Avg. squeeze force per inch squeeze}\big|_{\text{First Cycle}} \times \text{ Number of cycles}} \times 100 \qquad (5\text{-}3)$$

For test results shown in Figure 5-57, percentage clumping is found to be 172% showing that after each test cycle maximum force is increased by an average value of 172% of the maximum force in previous test cycle. The definition of (5-3) gives average values over a number of test cycles. But, as can be seen in Figure 5-57, the severity of clumping becomes worse for cycles 2-3 compared to cycle 1-2. The average value obtained from equation (5-3) is enough to quantitatively study the clumping effect.

5.4.1 Magnetic Field Density Dither

Because mechanical methods cannot be used to mix the MR fluid in squeeze mode due to small flow, some other means should be considered. One parameter whose effect on clumping can be studied is the magnetic field density. All tests on MR fluid presented so far were based on constant magnetic field density. In this section, effects of variable magnetic field density on the clumping behavior are studied.

In order to be able to control the magnetic field density, a QUANSER Linear Current Amplifier was added to the rheometer electric circuit. This current amplifier was callable of delivering a 7 Amps continuous output and a 9 Amps peak output. It used a 36 V power supply and had a wide bandwidth.

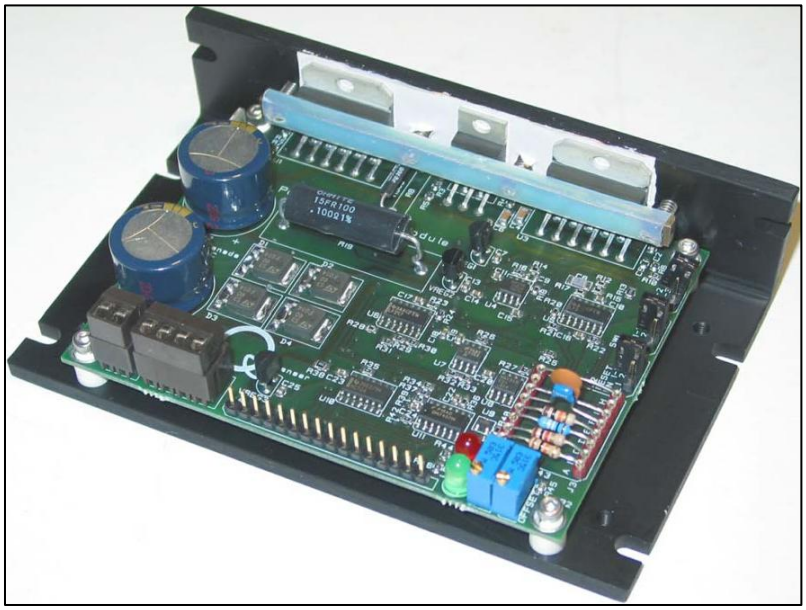

Figure 5-58 The linear current amplifier used to control the electric current passing through the rheometer coil

In order to investigate the effect of variable magnetic field density on the clumping behavior, during "piston return" portions of the tests, a magnetic dither was applied to the MR fluid. The squeezing portion of the experiments was the same as previous experiments using a constant magnetic field.

Various magnetic dithers could be applied to the MR fluid during the piston return. The dithers can have different amplitudes, frequencies, durations, and DC-offsets, although high frequencies could not be achieved due to the large time constant of the rheometer electric circuit. Various combinations were studied and it was seen that under certain conditions, the magnetic dither could improve the clumping behavior significantly. A sine wave form was chosen to dither the MR fluid. Figure 5-59 shows a comparison between a 5 cycle clumping test without magnetic dither and the same test with a magnetic dither during piston return. Dither amplitude was +/-1 Amp and frequency was 1 Hz. As shown in Figure 5-59, the current dither greatly reduces the clumping. Clumping was reduced from 172% to 29.5% by using a sine current dither during piston return. This reduction is because during piston return, the dither can cause the iron particles to escape aggregates and form new chains. The dither, however, could not eliminate clumping completely.

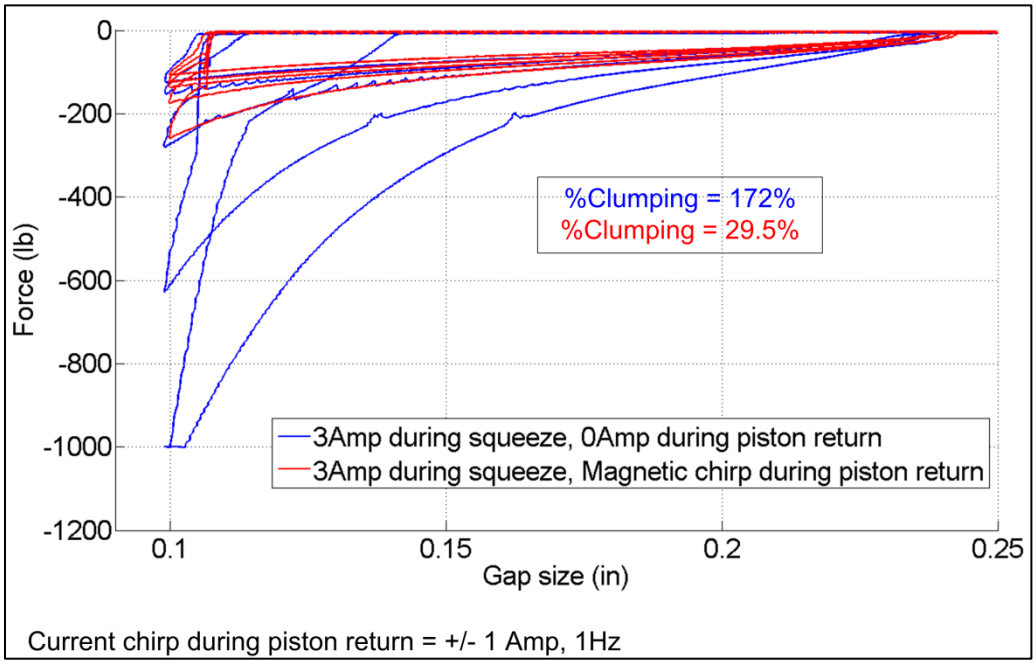

Figure 5-59 Effect of magnetic dither during piston return on clumping behavior

It should be mentioned that various combinations of current dither amplitude and frequency were studied but no pattern was found showing a relation between dither signal properties (amplitude and frequency) and reduction in clumping.

Results shown in Figure 5-59 are very promising. Clumping was reduced significantly by using a dither signal during piston return. However, further improvements can be made.

5.4.2 Further Improvements in Clumping Behavior

Although current dither greatly reduced clumping, this behavior should be minimized in MR squeeze devices. Further improvements in clumping can be achieved by not only using variable magnetic field density as shown earlier, but also by changing the magnetic field lines. As the magnetic field lines change direction, iron particles tend to direct themselves along the new magnetic field lines. This, along with a change in the magnetic field density, can help the particle aggregates to re-disperse and further improve clumping.

In order to be able to change the field direction during testing, the rheometer design has to be modified. This can be achieved by redesigning the rheometer piston and implementing a secondary coil, called the *"piston coil"*. During the normal squeezing portion of the experiments, the piston coil is turned off and the magnetic field lines are the same as before, as shown in Figure 5-60.

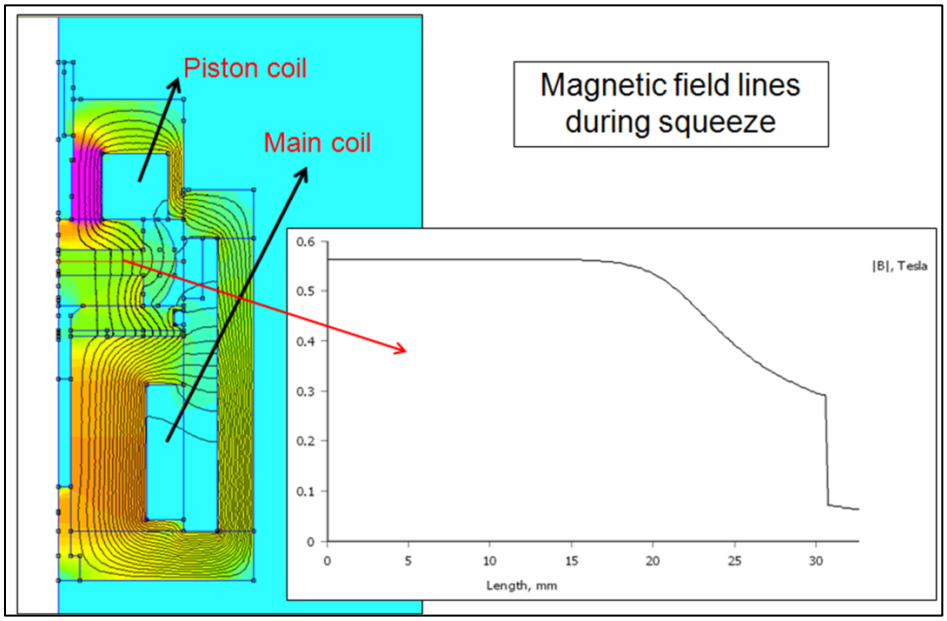

Figure 5-60 During the squeeze portion of the tests the piston coil is turned off

During piston return, the piston coil can be turned on and the magnetic field lines will change direction due to addition of another magnetic field source to the magnetic field generated by the rheometer main coil, shown in Figure 5-61.

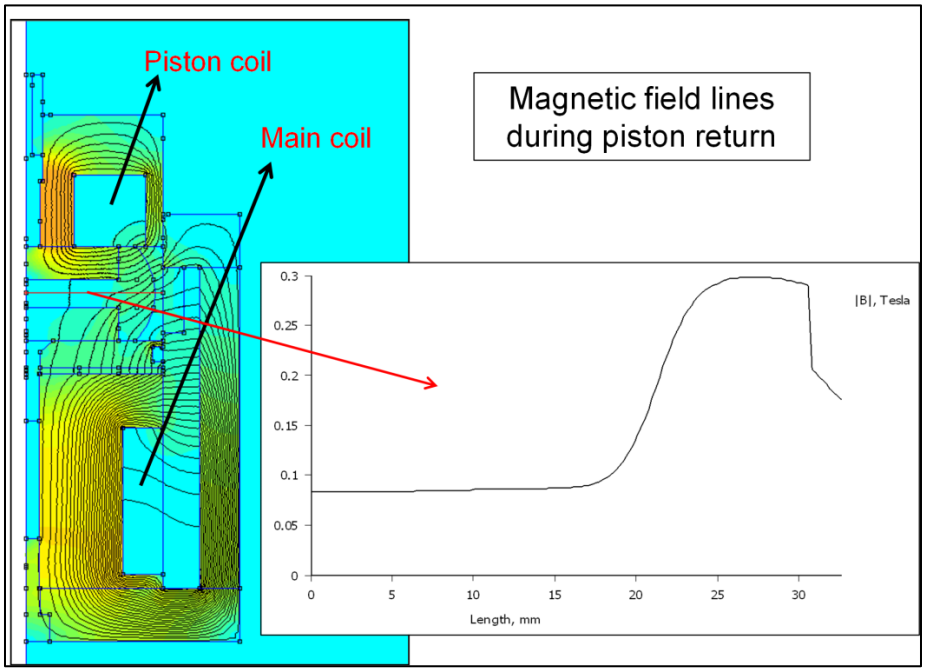

Figure 5-61 During piston return, piston coil is turned on resulting in overall magnetic field direction change

As can be seen in Figure 5-60 and Figure 5-61, both magnetic field direction and density change as the piston coil is turned on. During the squeezing operation, the magnetic field lines are parallel to the direction of the piston motion and the magnetic field density is homogeneous in the MR pouch gap, as expected and desired. When the piston coil is turned on during piston return, magnetic field lines direction and magnetic field density change significantly. In this case, the maximum value of the magnetic field density is outside of the pole plates, causing the iron particles to escape from the gap and move outwards. If an alternating electric current is passed through the piston coil, magnetic field lines and magnetic field density switch between Figure 5-60 and Figure 5-61. This cyclic change in magnetic field is expected to help the iron particle escape the aggregates and improve clumping.

This idea has not been experimentally investigated and will be proposed as a future work to extend the work performed in this project. It is believed, however, that this technique can improve clumping beyond what was achievable by only varying magnetic field density, discussed in the previous section.

Chapter 6
Study of Shim Stack Assemblies in Hydraulic Dampers for Suspension System Design and Tuning

In the previous chapters, novel technologies were investigated to improve the performance of suspension systems. The squeeze flow was extensively studied both theoretically and experimentally, and was found to be attractive for certain applications such as hybrid dampers and various mount configurations for mitigating vibrations. The next step is to integrate this novel technology into current suspensions. In doing so, it is essential that conventional suspension components be completely understood. Therefore, in this part of the presented research, conventional suspension systems are studied using advanced mathematical tools. The component of interest is the hydraulic damper and its related sub-components.

This Chapter is focused on studying the behavior of shim stack assemblies in shock absorbers[‡]. In hydraulic dampers, a stack of circular disks (shims) is mounted on each side of the main piston to create a pressure drop as the hydraulic oil is passed through the piston from one chamber to the other (Figure 6-1

[‡] In this document, the terms "Shock Absorber" and "Hydraulic Damper" are interchangeably used. However, "Shock Absorber" seems to be less appropriate to use because dampers *dissipate* the energy of the suspension motions rather than *absorbing* it.

and Figure 6-2). A stiff shim stack creates a high pressure drop across the piston, resulting in high damping. A softer shim stack creates less pressure drop which results in less damping.

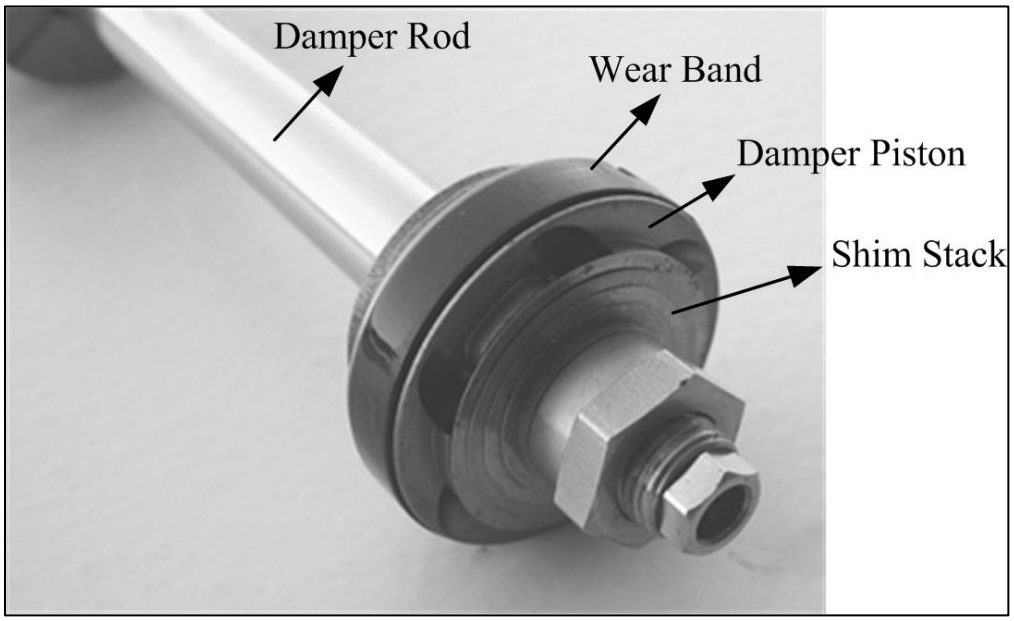

Figure 6-1 Main piston assembly of a typical hydraulic damper and its major components

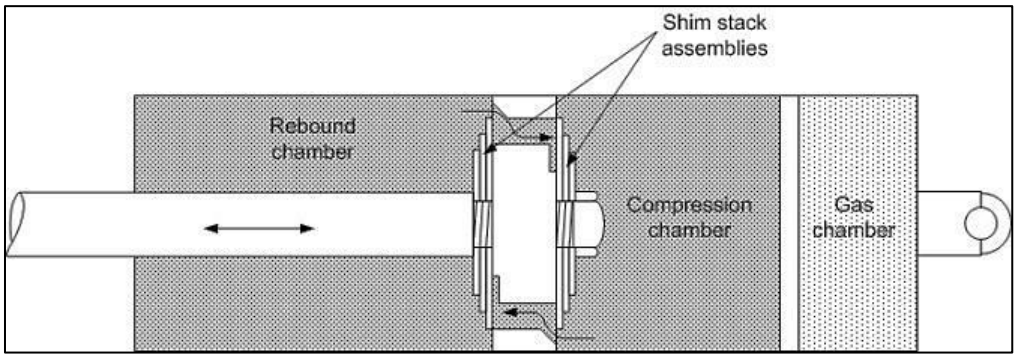

Figure 6-2 Hydraulic damper piston and shim stack assembly

In practice, shims can be added to or removed from the shim stack assembly to obtain the desired level of damping. This requires taking the damper apart, making changes to the shim stack assembly, and putting the damper back together, which takes a considerable amount of time and effort. The goal of the study discussed in this chapter is to provide an accurate model of the shim stack assembly deflection in order to predict the level of damping for different configurations of the shim stack. The shims assembled one upon the other will deflect under the load created by the hydraulic oil, and at the same time, will slide with

respect to each other. This important characteristic of the shim stack needs to be accounted for in the mathematical model and makes the analysis more challenging. Usually, the deflection of the shim stack is approximated by the deflection of a single disk and formulas for a single disk are used [126]. As will be shown in the next sections, this will introduce a large amount of error in the analysis of hydraulic dampers due to lower stiffness of the shim stack assembly compared to a single disk with similar geometry. This is due to the fact that the shims can slide with respect to each other while deflecting.

Literature review reveals that to date, there are no published studies on deflection analysis of shim stacks in hydraulic dampers. There are, however, numerous studies on various aspects of deflection and stress analysis of single disks and sandwich plates under different loading and boundary conditions. Wahl et al. [127] studied stress and deflection of flat circular disks with central holes and obtained simplified formulas for maximum stress and maximum deflection. They also compared their results with experiments for some cases. Essenburg and Gulati [128] studied contact of two axisymmetric plates and showed that at least in the absence of additional assumptions, classical plate theory cannot treat the problem of the contact of two plates because the conditions on the transmitted force and the derivative of the deflection cannot be satisfied. Smith [129] considered the effects of shear deformations on large deflections of circular sandwich plates and presented an exact solution for axisymmetric bending of sandwich plates using power series. Turvey [130] worked on the effect of shear deformations on the yielding of circular plates and presented correction factors to relate the first yield stress and deflection due to classical plate theory (CPT) to shear deformation plate theory (SDPT). Lee et al. [131] considered the axisymmetric bending of clamped thick circular plates using a higher-order shear deformation theory and compared the maximum deflection and stress for different plate theories. Wang [132] studied the deflection of sandwich plates and represented exact relationships between the deflection of isotropic sandwich plates and their corresponding Kirchhoff plates. Hyer and Jilani [133] contributed to predicting the axisymmetric deformations of disk-style actuators considering geometric nonlinear effects. Chung and Wang [134] worked on the optimal design of stepped circular plates, including shear deformations, and

presented a canonical exact deflection expression for stepped circular plates. There are also numerous books on deflection and stress analysis of various plates and shells [135-137].

In the presented study here, energy and variational methods [138] are used to analyze the deflection of the shim stack assembly. Two different cases are considered: 1) a single stepped disk with piecewise constant thickness sections and, 2) a shim stack with the same geometry. Total potential energy of each of the cases is calculated and minimized using the Rayleigh-Ritz approximation and Lagrange multiplier methods.

The Rayleigh-Ritz method is used as an approximation to solve the deflection problems of a stepped disk and the shim stack assembly using FSDT. In this method, functions with some unknown coefficients are used as deflection functions. The more realistic the types of these functions are, the more accurate the Rayleigh-Ritz method would be. A good approximation for the stepped disk and shim stack problems is the exact solution for a single disk with constant thickness under similar loading and boundary conditions. Hence, in section 6.3, the exact solution for a single disk of constant thickness is discussed using FSDT. In sections 6.4 and 6.5, the stepped disk and shim stack assembly are discussed, respectively. In section 6.6, a numeric example is presented and discussed. Section 6.7 focuses on verification of the mathematical model and the chapter is summarized and concluded in section 6.8.

6.1. Problem Statement

Consider the shim stack shown in Figure 6-3. The shim stack consists of three disks stacked one upon the other and each clamped at the inner radius ($r = a$) and free at the outer radius ($r = R$). A loading, $p(r)$, per unit area, is acting on the shim stack causing it to deflect in the positive *z*-direction (downward). It is assumed that the loading and boundary conditions are axisymmetric. In real conditions, non-symmetric loading can occur due to the existence of non-annular piston orifices. As the shim stack is deflected, the

disks bend due to the force transmitted to them through the upper disks. As the friction between the disks is minimal, the disks can easily slide on each other as they bend.

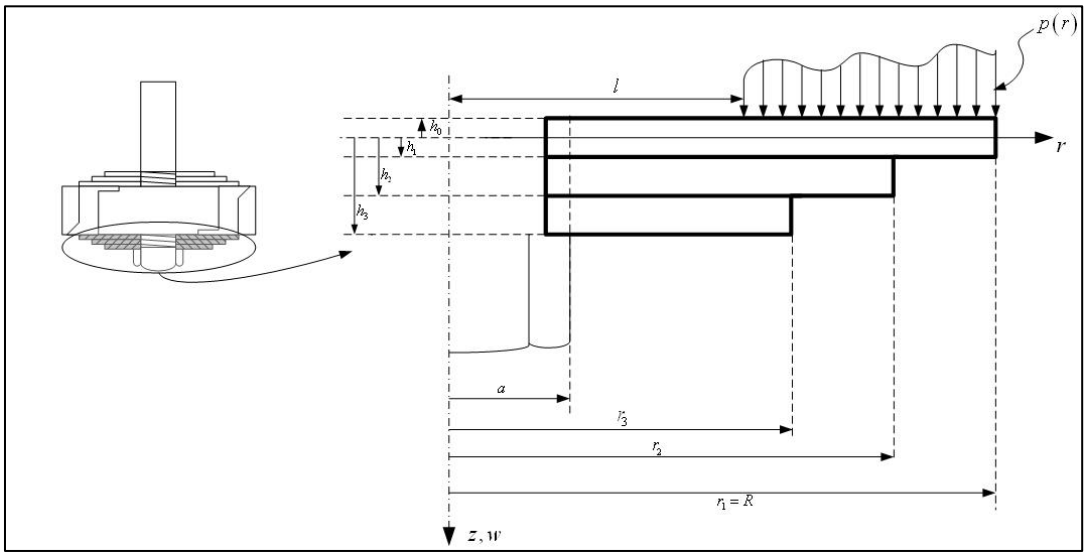

Figure 6-3 The shim stack assembly consisting three disks

As the shim stack bends, a cross-section in the upper disk such as *A*-*B* in Figure 6-4 will rotate to *A'*-*B'* and a cross-section in the middle disk such as *B*-*C* will rotate to *B''*-*C'*, etc. Because the disks can slide on each other freely, point *B* on the upper disk will move to the left *(B')* and point *B* on the lower disk will move to the right *(B'')*.

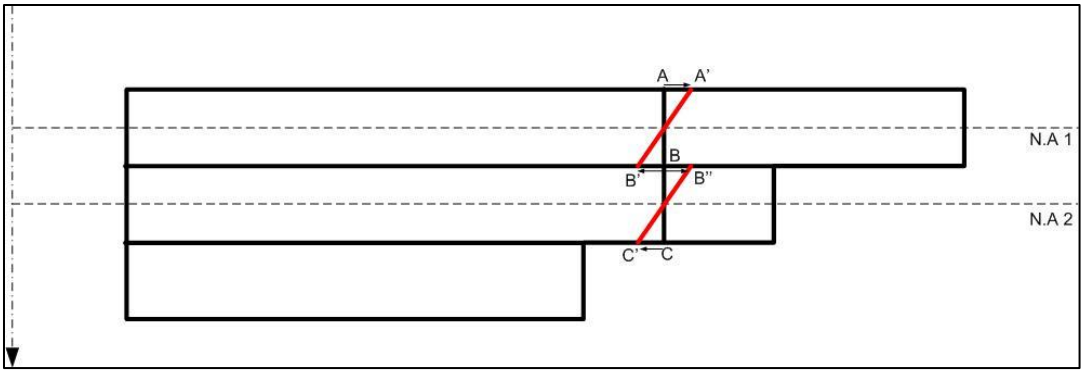

Figure 6-4 In a shim stack, the cross-section rotation in each disk is independent of other disks

Now, consider that the disks are constrained, for example, by using epoxy of zero thickness between the pairs of disks (which now resembles a stepped disk), as shown in Figure 6-5. In this case, point *B* on the

upper disk and point *B* on the lower disk are not two separate points anymore, but one point. Hence, this point can only move in one direction because there is resistance due to the constraining epoxy. Due to this constraint, it is more difficult to deflect the stepped disk compared to the shim stack.

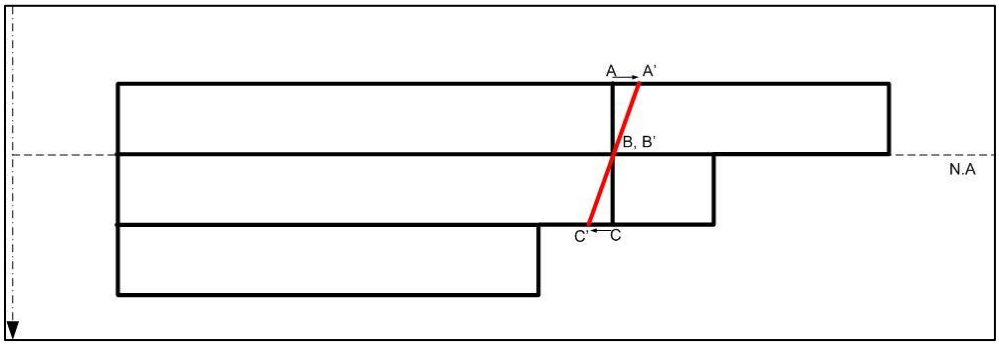

Figure 6-5 If the disks are constrained using superglue, the shim stack resembles a stepped disk

As mentioned, in order to be able to accurately predict the deflection of a shim stack under loading, plate theories that include shear deformations should be used. In annular plates, the deflection due to shear may constitute a considerable portion of the total deflection [126]. Hence, the FSDT is used in the presented work.

As a circular disk is deflected, every point on the disk moves in both normal and radial directions. In a circular disk with constant thickness, the radial and normal deflections are decoupled. But in a stepped disk, the radial deflection will result in tangential stresses which affect the deflection in the normal direction due to change in location of neutral plane from one section to another. This effect is studied in section 6.2. Classical Plate Theory (CPT) is used to investigate the effect of these radial deflections and the stepped disk deflection problem is solved exactly. It is found that this effect is not of considerable importance for the geometries considered and can be neglected.

6.2. Exact Solution for a Stepped Disk

In this section, the problem of a stepped disk, as shown in Figure 6-6, is solved exactly using CPT. Each section of the disk with constant thickness can be considered as a single disk with constant thickness

attached to another disk with a different thickness. From CPT, the solution for radial and axial deflections for a disk with constant thickness, subjected to a uniform pressure, p_0, is [133]

$$\begin{aligned} w(r) &= \frac{p_0 r^4}{64\tilde{D}} + a_1 r^2 + a_2 \ln(r) + a_3 r^2 \ln(r) + a_4 \\ u(r) &= a_5 r + \frac{a_6}{r} \end{aligned} \tag{6-1}$$

where, w is the deflection in the z-direction, u is the radial deflection in the r-direction, p_0 is the constant pressure acting on the disk, a_i are constants, and $\tilde{D}$ is the bending rigidity defined as,

$$\tilde{D} = \frac{Et^3}{12(1-\nu^2)} \tag{6-2}$$

E is the Young's modulus of the disk, t is the thickness of the disk, and ν is its Poisson's ratio.

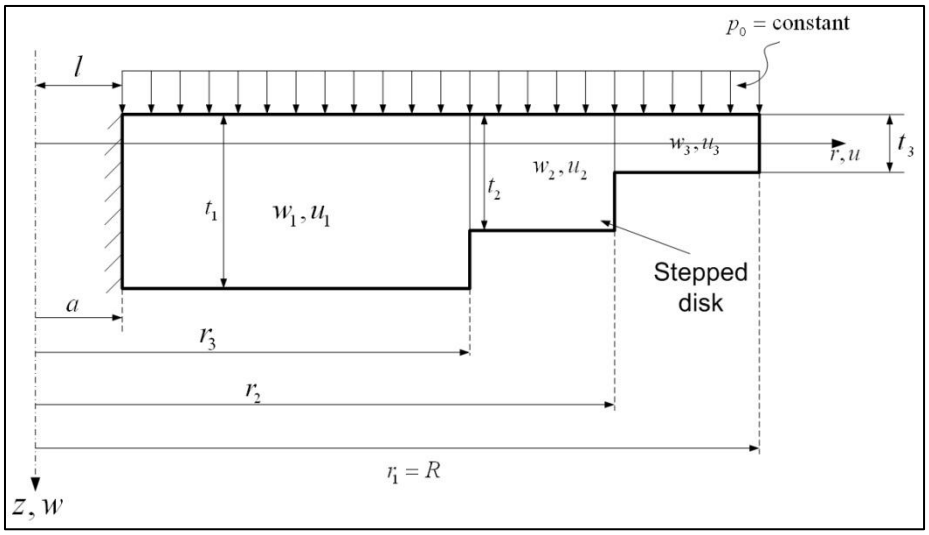

Figure 6-6 Schematics of a stepped disk with varying thickness under a constant pressure

The radial and axial deflections for each section of the stepped disk in Figure 6-6 with constant thickness are in the form of equations (6-1) with different constants. The stepped disk of Figure 6-6 consists of three sections. Hence, three different functions need to be used for w and three different functions are

needed for *u*, resulting in 18 unknown coefficients. In order to find these constants, the following boundary, continuity, and equilibrium conditions are imposed.

6.2.1 Boundary Conditions

At the inner radius, the deflection components and the slope are zero. At the outer radius, the normal force (N_r), shear force (Q_r), and bending moment (M_r) resultants are zero.

$$
\begin{aligned}
at\ r = a &\begin{cases} u_1 = 0 \\ w_1 = 0 \\ w_1' = 0 \end{cases} \\
at\ r = R &\begin{cases} rN_{r3} = \tilde{A}_3 r\left(u' + \nu \dfrac{u}{r} \right) = 0 \\ -rQ_{r3} = \tilde{D}_3 r\left(w''' + \dfrac{1}{r} w'' - \dfrac{1}{r^2} w' \right) = 0 \\ -rM_{r3} = \tilde{D}_3 r\left(w'' + \dfrac{\nu}{r} w' \right) = 0 \end{cases}
\end{aligned}
\tag{6-3}
$$

where, t_3 is the thickness of the outer disk, $\tilde{D}_3 = \dfrac{E t_3^{\,3}}{12\left(1-\nu^2\right)}$ is its bending rigidity, and $\tilde{A}_3 = \dfrac{E t_3}{1-\nu^2}$ is its extensional stiffness.

6.2.2 Continuity Conditions

The axial deflection, *w*, and the slope, w', are continuous at $r = r_2$ and $r = r_3$. The continuity conditions on radial deflection, *u*, can be obtained from the diagram shown in Figure 6-7.

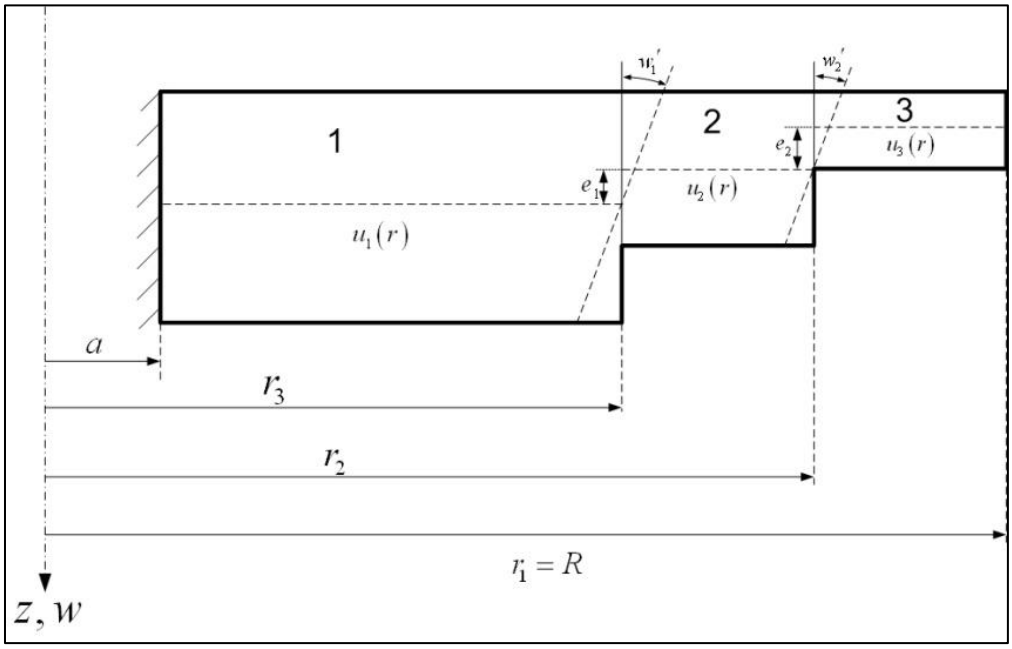

Figure 6-7 Continuity condition on the radial deflections in a stepped disk

At $r = r_3$, the radial deflection of the neutral plane of section 2 is equal to the radial deflection of the neutral plane of section 1 superimposed with the radial deflection due to the rotation of the cross-section at $r=r_3$. At $r = r_2$, a similar condition applies. Referring to Figure 6-7, it is seen that the vertical distance between the neutral planes of sections 1 and 2, and 2 and 3 are e_1 and e_2 , respectively.

The resulting kinematic continuity conditions can be written as

$$\begin{aligned} at\ r = r_3 &\begin{cases} u_2 = u_1 + e_1 w_1' \\ w_1 = w_2 \\ w_1' = w_2' \end{cases} \\ at\ r = r_2 &\begin{cases} u_3 = u_2 + e_2 w_2' \\ w_2 = w_3 \\ w_2' = w_3' \end{cases} \end{aligned} \tag{6-4}$$

6.2.3 Equilibrium Conditions

At locations where the cross section thickness changes, the equilibrium conditions on normal force, shear force, and bending moment can be obtained from a free body diagram as shown in Figure 6-8.

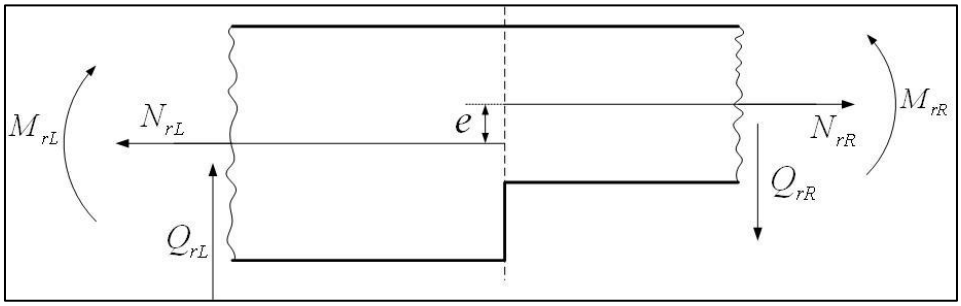

Figure 6-8 FBD at the location where thickness changes

Due to the eccentricity in the location of the neutral planes, the normal forces create a couple that make the bending moment discontinuous at $r = r_3$ and $r = r_2$. So, the resulting equilibrium conditions are

$$\begin{aligned} at\ r = r_3 &\begin{cases} N_{r1} = N_{r2} \\ Q_{r1} = Q_{r2} \\ M_{r2} = M_{r1} + e_1 N_{r1} \end{cases} \\ at\ r = r_2 &\begin{cases} N_{r2} = N_{r3} \\ Q_{r2} = Q_{r3} \\ M_{r3} = M_{r2} + e_2 N_{r2} \end{cases} \end{aligned} \tag{6-5}$$

Equations (6-3), (6-4), and (6-5) form a system of 18 equations with 18 unknowns. Solving this system of equations will result in the exact answer for the deflection of the stepped disk of Figure 6-6.

To investigate the effect of radial deformations on the normal deformations, the stepped disk problem is solved ignoring the radial deformations, *u*. The procedure is similar to what is described so far. Figure 6-9 shows a comparison between the normal deflections, *w*, for the given geometric and material properties. As shown, including the radial deformation only changes the normal deflection slightly. The tip deflection is 6.4% more if the radial deflections are included, i.e., the disk is slightly more flexible. Hence, in the present analysis, the effect of the radial deflections will be ignored. Although the effects of radial deflections was assessed using CPT, it is expected that for the FSDT the same scenario will occur.

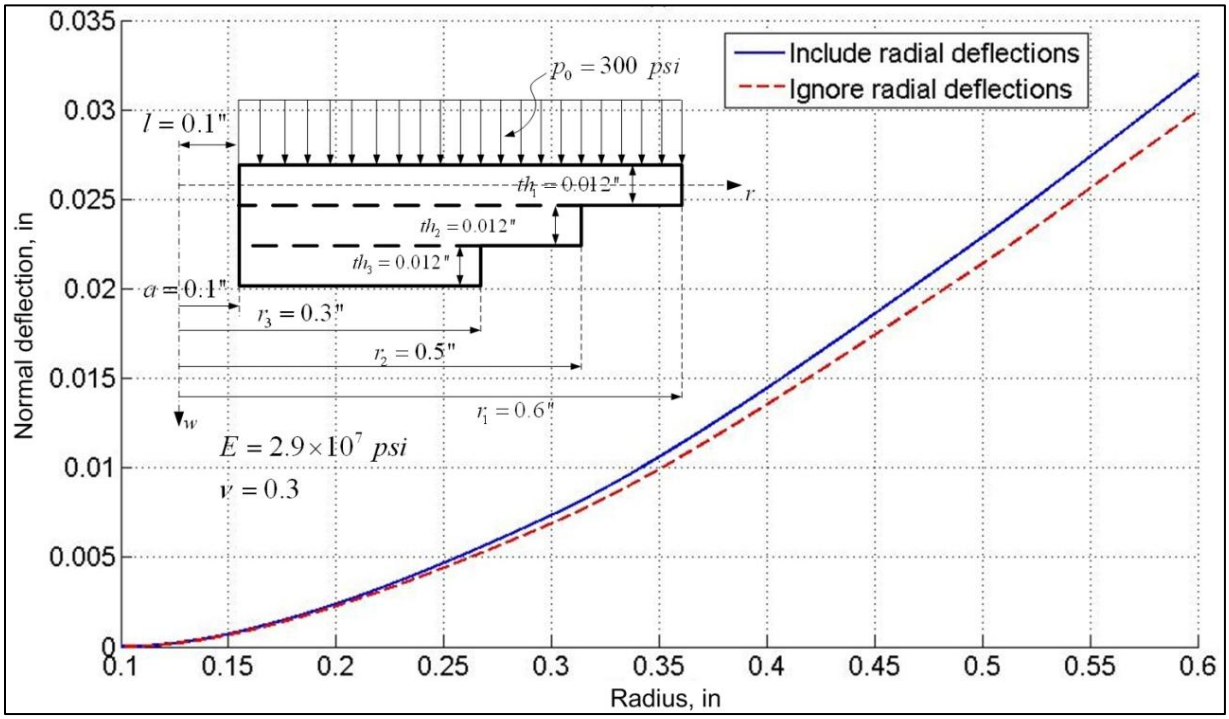

Figure 6-9 Including the radial deflections will slightly affect the normal deflections

6.3. Deflection of a Single Annular Disk with Constant Thickness (FSDT)

In order to analyze the deflection of the shim stack, the Rayleigh-Ritz method is used. This method requires the use of approximation functions for deflection and cross-section rotation (if FSDT is used). Good approximations for deflection and cross-section rotation of a shim stack are the exact analytical solutions of a single disk with constant thickness under similar loading and boundary conditions. In this section, the closed form solution for a single disk with constant thickness clamped at the inner edge and free at the outer edge is obtained using FSDT which includes shear deformations. Energy and variational methods are used to obtain the governing differential equations and consistent boundary conditions.

Consider a single annular disk with constant thickness as shown in Figure 6-10. The disk is clamped at the inner edge ($r = R_1$) and is free at the outer edge ($r = R_2$). A load, $p(r)$, per area of the disk is acting on the disk. The disk has a thickness of *h*.

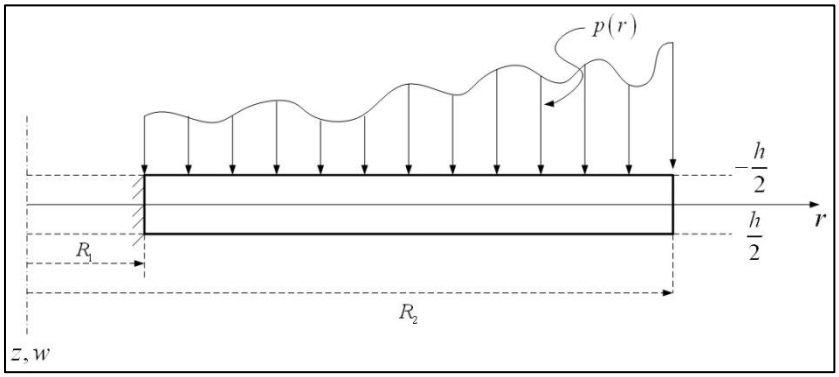

Figure 6-10 A single annular disk clamped at the inner edge and free at the outer edge

First order shear deformation theory is used to obtain the exact solution. The FSDT takes into account the shear deformations which are important in distinguishing between the stepped disk and the shim stack. The theory is called "*Hencky theory"* and the key assumptions are

1- Planar sections remain plane (similar to the classical Euler-Bernoulli theory) but are not normal to the reference line after deformation.

2- $\tau_{zz} = \tau_{r\theta} = \tau_{\theta z} = 0$, i.e., these stress components have little contributions to the strain energy.

3- Deformations are infinitesimal.

4- Linear elastic behavior is assumed.

The assumed displacements for the first order shear deformation theory are

$$\begin{aligned} u(r,\theta,z) &= -z\psi^0(r) \\ v(r,\theta,z) &= 0 \\ w(r,\theta,z) &= w^0(r) \end{aligned} \tag{6-6}$$

where, *u*, *v*, and *w* are the deformations in *r*, θ, and *z* directions, respectively. ψ^0 is the cross-section rotation and the $(\ )^0$ sign indicates that the deflections are those of cross-section reference surface, the mid surface of the disk. It is also assumed that the deflection of the disk is in the elastic range and will not cause any plastic deformations.

The total strain energy of the disk is

$$U = \frac{1}{2}\iiint_{\forall}\left(\tau_{rr}\varepsilon_{rr} + \tau_{\theta\theta}\varepsilon_{\theta\theta} + \tau_{rz}\gamma_{rz}\right)d\forall \tag{6-7}$$

where, U is the total strain energy, τ_{ij} are the stress components that contribute to the strain energy, ε_{ij} are the normal strain components, γ_{rz} is the shear strain, and $\forall$ is the volume of the disk. The strain-displacement relations in polar coordinates for the axisymmetric case are

$$\begin{aligned}\varepsilon_{rr} &= \frac{\partial u}{\partial r}\\ \varepsilon_{\theta\theta} &= \frac{u}{r}\\ \gamma_{zr} &= \frac{\partial u}{\partial z} + \frac{\partial w}{\partial r}\end{aligned} \tag{6-8}$$

Replacing equations (6-6) in equations (6-8) and dropping the notation $(\ )^0$ results in

$$\begin{aligned}\varepsilon_{rr} &= -z\psi'\\ \varepsilon_{\theta\theta} &= -\frac{z}{r}\psi\\ \gamma_{rz} &= w' - \psi\end{aligned} \tag{6-9}$$

For a linear elastic material, Hooke's law states that

$$\begin{aligned}\tau_{rr} &= \frac{E}{1-\nu^2}\left(\varepsilon_{rr} + \nu\varepsilon_{\theta\theta}\right)\\ \tau_{\theta\theta} &= \frac{E}{1-\nu^2}\left(\varepsilon_{\theta\theta} + \nu\varepsilon_{rr}\right)\\ \tau_{rz} &= G\gamma_{rz}\end{aligned} \tag{6-10}$$

where, G is the shear modulus of the disk material. Substituting equations (6-9) and (6-10) into equation (6-7), the total strain energy of the disk becomes

$$U = \frac{1}{2}\iiint\left\{\frac{E}{1-\nu^2}\left(z^2\psi'^2 + \frac{z^2}{r^2}\psi^2 + 2\nu\frac{z^2}{r}\psi\psi'\right) + kG\left(w' - \psi\right)^2\right\}d\forall \tag{6-11}$$

where, *k* is the shear correction factor. In first order plate theories, the shear stress through the thickness is assumed to be constant, which is not compatible with the surface shear forces being zero. Hence, the factor *k* is used to correct this problem. In the literature, many shear correction factors are found theoretically and experimentally for different situations [139-141]. The shear correction factors depend not only on the material properties and geometric parameters, but also on the loading and boundary conditions [126]. For the present work, a value of *k=0.833* was used for the shear correction factor. This shear correction factor is used for a single disk with constant thickness in the literature.

Considering a disk with constant thickness, as shown in Figure 6-10, the volume integral of equation (6-11) reduces to

$$U = \pi \int_{R_1}^{R_2} \left\{ \tilde{D} \left(r\psi'^2 + \frac{\psi^2}{r} + 2\nu\psi\psi' \right) + kGhr\left(w'^2 - 2w'\psi + \psi^2 \right) \right\} dr \tag{6-12}$$

The work done by the pressure loading is

$$V = \int_{R_1}^{R_2} q(r) w(r) dr \tag{6-13}$$

where, $q(r) = 2\pi r p(r)$. Using equations (6-12) and (6-13), the total potential energy of the disk becomes

$$\Pi = U - V = \pi \int_{R_1}^{R_2} \left\{ \tilde{D} \left(r\psi'^2 + \frac{\psi^2}{r} + 2\nu\psi\psi' \right) + kGhr\left(w'^2 - 2w'\psi + \psi^2 \right) \right\} dr - \int_{R_1}^{R_2} 2\pi r p w dr \tag{6-14}$$

According to the principle of minimum total potential energy, the disk deflects such that the total potential energy is minimum. This implies that the first variation of the total potential energy should be zero. That is,

$$\begin{aligned}\delta\Pi = \pi\{&\int_{R_1}^{R_2}\left[2\tilde{D}\left(r\psi'+\nu\psi\right)\right]\delta\left(\psi'\right)dr+\int_{R_1}^{R_2}\left[2\tilde{D}\left(\frac{\psi}{r}+\nu\psi'\right)-2kGhr\left(w'-\psi\right)\right]\delta\psi\,dr\\&+\int_{R_1}^{R_2}\left[2kGhr\left(w'-\psi\right)\right]\delta\left(w'\right)dr\}-\int_{R_1}^{R_2}2\pi rp\delta w dr=0\end{aligned} \tag{6-15}$$

Integrating the first and third integrals in (6-15) by parts, followed by some simplification yields

$$\begin{aligned}\delta\Pi = \pi\{&\left[2\tilde{D}\left(r\psi'+\nu\psi\right)\right]\delta\psi\Big|_{R_1}^{R_2}-\int_{R_1}^{R_2}\frac{d}{dr}\left[2\tilde{D}\left(r\psi'+\nu\psi\right)\right]\delta\psi\,dr\\&+\int_{R_1}^{R_2}\left[2\tilde{D}\left(\frac{\psi}{r}+\nu\psi'\right)-2kGhr\left(w'-\psi\right)\right]\delta\psi\,dr+\left[2kGhr\left(w'-\psi\right)\right]\delta w\Big|_{R_1}^{R_2}\\&-\int_{R_1}^{R_2}\frac{d}{dr}\left[2kGhr\left(w'-\psi\right)\right]\delta w dr\}-\int_{R_1}^{R_2}2\pi rp\delta w dr=0\end{aligned} \tag{6-16}$$

For the first variation of the total potential energy in equation (6-16) to be zero, all the terms should be identically zero. This results in the Euler-Lagrange equations and the permissible boundary conditions. The Euler-Lagrange equations are

$$\begin{cases}\dfrac{d}{dr}\left(rM_r\right)-M_\theta-krQ_r=0 & (a)\\ -k\dfrac{d}{dr}\left(rQ_r\right)-rp\left(r\right)=0 & (b)\end{cases} \tag{6-17}$$

where,

$$\begin{aligned}M_r &= -\tilde{D}\left(\psi'+\nu\frac{\psi}{r}\right)\\ M_\theta &= -\tilde{D}\left(\frac{\psi}{r}+\nu\psi'\right)\\ Q_r &= Gh\left(w'-\psi\right)\end{aligned} \tag{6-18}$$

are the bending moment resultant in r direction, bending moment resultant in θ-direction, and the shear force resultant, respectively. The variationally consistent boundary conditions at $r=R_1$ and $r=R_2$ can be stated as

$$\begin{cases}\text{Either} \\ 1)\, rM_r = 0 \\ \text{Or} \\ 2)\ \delta(\psi) = 0 \end{cases}$$
$$\text{AND} \qquad (6\text{-}19)$$
$$\begin{cases}\text{Either} \\ 3)\, rQ_r = 0 \\ \text{Or} \\ 4)\ \delta(w) = 0 \end{cases}$$

For an annular disk clamped at the inner edge and free at the outer edge, as shown in Figure 6-10, the deflection and the cross-section rotation are zero at $r = R_1$; and at $r = R_2$, the moment and the shear resultants are zero, i.e.,

$$at\ \ r = R_1 :\quad \psi = 0\ \ and\ \ w = 0 \qquad (6\text{-}20)$$

$$at\ \ r = R_2 :\quad M_r = 0\ \ and\ \ Q_r = 0 \qquad (6\text{-}21)$$

The exact answer for deflection and cross-section rotation of the disk shown in Figure 6-10 under a general loading, *q*, is obtained by solving equations (6-17) in conjunction with the boundary conditions (6-20) and (6-21). For a uniform pressure ($p(r) = p_0$) on the disk, equation (6-17b) can be solved to yield

$$Q_r(r) = \frac{p_0}{2}\left(\frac{R_2^{\,2}}{r} - r\right) \qquad (6\text{-}22)$$

Replacing the solution for Q_r in the second equation in (6-17) and solving for cross-section rotation, ψ, results

$$\psi(r) = \frac{c_1}{r} + c_2 r + \frac{k p_0}{4\tilde{D}}\left(\frac{1}{4}r^3 - R_2^{\,2} r \ln r\right) \qquad (6\text{-}23)$$

To obtain the solution for the deflection, *w*, equations (6-22) and (6-23) are replaced into the third equation in (6-18) and the resulting first order differential equation is solved, yielding

$$w(r)=\frac{p_0}{2Gh}\left(R_2^{\,2}\ln r-\frac{r^2}{2}\right)+c_2\frac{r^2}{2}+c_1\ln r+\frac{kp_0}{64\tilde{D}}r^4-\frac{kp_0R_2^{\,2}}{8\tilde{D}}\left(r^2\ln r-\frac{r^2}{2}\right)+c_3 \tag{6-24}$$

The constants c_1, c_2, and c_3 can be found using the following boundary conditions

$$\begin{aligned} w(R_1)&=0\\ \psi(R_1)&=0\\ M_r(R_2)&=0 \end{aligned} \tag{6-25}$$

Letting *k=1* and $G\to\infty$, equation (6-24) reduces to the classical solution for the deflection of a single disk with constant thickness, as discussed in equation (6-1).

The general solutions obtained in equations (6-23) and (6-24) will be used in the Rayleigh-Ritz method for the shim stack deflection analysis.

6.4. Case A: A Stepped Disk

In this section, a stepped disk, referred to as Case A, is considered and its deflection is predicted using the Rayleigh-Ritz method. The total potential energy of the disk is calculated using approximate displacement functions of the form of equations (6-23) and (6-24). The obtained total potential energy is then minimized using the Lagrange multipliers method and the deflection of the stepped disk is found.

Consider the stepped disk of Figure 6-11. The stepped disk is clamped at $r=a$ and free at $r=R$. A loading, $p(r)$, per unit area of the disk is applied to the disk.

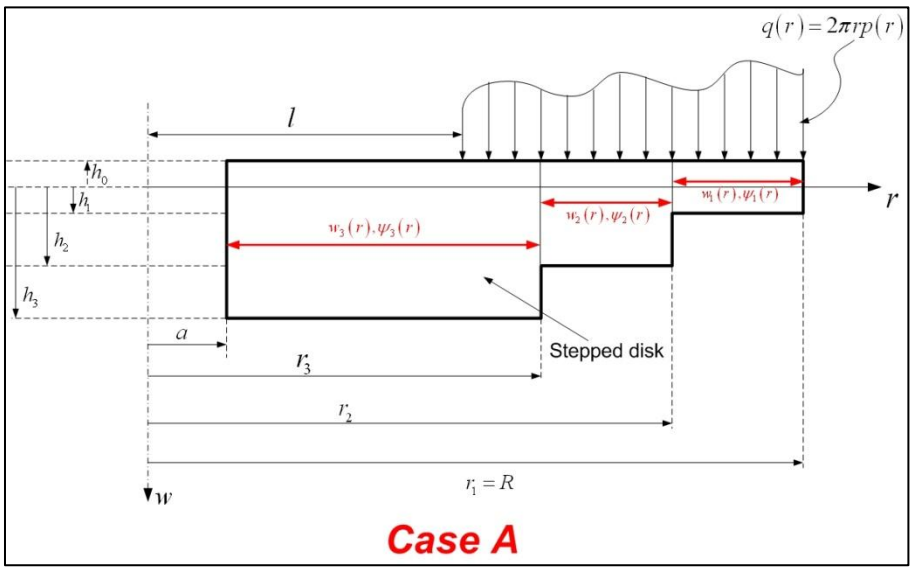

Figure 6-11 Case A: a 3-stepped disk clamped at the inner edge and free at the outer edge

Because the basic assumptions used for Case A are the same as assumptions used for a single disk with constant thickness, equation (6-11) is valid for Case A. Using equation (6-11) and (6-13) the total potential energy of case A can be written as

$$\Pi_A=\pi\int_r\left\{\int_z z^2dz\right\}F_1(\psi,\psi')dr+\pi\int_r\left\{\int_z dz\right\}F_2(w',\psi)dr-\int_l^R q(r)w(r)dr \tag{6-26}$$

where, $F_1(\psi,\psi')=\dfrac{E}{1-\nu^2}\left(r(\psi')^2+\dfrac{\psi^2}{r}+2\nu\psi\psi'\right)$ and $F_2(w',\psi)=rGk(w'-\psi)^2$. The z-integrals in equation (6-26) are computed with the z-axis being the mid-plane of each section with constant thickness.

Case A, again, can be considered as a combination of rings of constant thickness attached to each other. Each of these rings is an annular disk with constant thickness for which equations (6-23) and (6-24) are valid. Hence, the integrals in equation (6-26) can be broken down in a manner such that for each section with constant thickness one deflection function and one cross-section rotation is used in the Rayleigh-Ritz method. This is shown in Figure 6-11. In the Rayleigh-Ritz method, the deflection and cross-section rotation of each section with constant thickness is assumed to be of the form of equations (6-23) and (6-24) with unknown coefficients. These forms of solutions are substituted in the total potential energy

expression (6-26) and the integrations are carried out. This results in an expression in terms of the unknown coefficients which then will be minimized to solve for the coefficients.

Breaking the integrals in equation (6-26) in a proper manner consistent with the geometry of the stepped disk results in

$$\begin{aligned}\Pi_A = &\frac{\pi}{12}\left(h_3-h_0\right)^3\int_a^{r_3} F_1\left(\psi_3,\psi_3'\right)dr + \frac{\pi}{12}\left(h_2-h_0\right)^3\int_{r_3}^{r_2} F_1\left(\psi_2,\psi_2'\right)dr \\ &+\frac{\pi}{12}\left(h_1-h_0\right)^3\int_{r_2}^{r_1} F_1\left(\psi_1,\psi_1'\right)dr + \pi\left(h_3-h_0\right)\int_a^{r_3} F_2\left(w_3',\psi_3\right)dr \\ &+\pi\left(h_2-h_0\right)\int_{r_3}^{r_2} F_2\left(w_2',\psi_2\right)dr + \pi\left(h_1-h_0\right)\int_{r_2}^{r_1} F_2\left(w_1',\psi_1\right)dr \\ &-\int_l^{r_3} q(r)w_3(r)dr - \int_{r_3}^{r_2} q(r)w_2(r)dr - \int_{r_2}^{r_1} q(r)w_1(r)dr\end{aligned} \tag{6-27}$$

From the analytical solution for a single disk in section 6.3, the form of approximation functions for deflection and cross-section rotation are

$$\begin{cases} w_i(r) = A_i r^4 + B_i r^2 + C_i \ln(r) + D_i r^2 \ln(r) + L_i r + H_i \\ \psi_i(r) = M_i r^3 + N_i r + \dfrac{S_i}{r} + T_i r \ln(r) + V_i \end{cases}, \quad i = 1,2,3 \tag{6-28}$$

where, A_i, B_i, C_i, D_i, L_i, H_i, M_i, N_i, S_i, T_i, and V_i are the unknown coefficients that need to be solved for by minimizing the total potential energy. It is best not to skip the lower order terms in the Rayleigh-Ritz method. Therefore, a linear term is added to the approximation functions for deflection suggested by equation (6-24) and a constant term is added to the approximation functions for cross-section rotation suggested by equation (6-23).

The deflection and the cross-section rotation should be continuous over the length of the disk. Also, at $r = a$, the deflection and cross-section rotation are zero. This results in the following constraints that need to be satisfied by the problem solution, in addition to minimizing the total potential energy.

$$
\begin{aligned}
&at\ r=a \begin{cases} eqn_{A1}=w_3(a)=0 \\ eqn_{A2}=\psi_3(a)=0 \end{cases} \\
&at\ r=r_3 \begin{cases} eqn_{A3}=w_2(r_3)-w_3(r_3)=0 \\ eqn_{A4}=\psi_2(r_3)-\psi_3(r_3)=0 \end{cases} \\
&at\ r=r_2 \begin{cases} eqn_{A5}=w_1(r_2)-w_2(r_2)=0 \\ eqn_{A6}=\psi_1(r_2)-\psi_2(r_2)=0 \end{cases}
\end{aligned} \tag{6-29}
$$

A solution that minimizes the total potential energy and at the same time, satisfies the constraints (6-29) is desired. In order to find such a solution, the Lagrange expression is formed as

$$
LE_A=\Pi_A+\sum_{j=1}^{6}\lambda_j eqn_{Aj} \tag{6-30}
$$

where, Π_A is the total potential energy of Case A, eqn_{Aj} are the constraints in (6-29), and λ_j are the Lagrange multipliers that are unknowns.

Minimizing the Lagrange expression in (6-30) assures that the total potential energy is minimized and the constraints in (6-29) are satisfied, through solving the following non-linear equations

$$
\begin{cases}
\dfrac{\partial LE_A}{\partial A_i}=0\ ,\ \dfrac{\partial LE_A}{\partial B_i}=0\ ,\ \dfrac{\partial LE_A}{\partial C_i}=0\ ,\ \dfrac{\partial LE_A}{\partial D_i}=0 \\
\dfrac{\partial LE_A}{\partial L_i}=0\ ,\ \dfrac{\partial LE_A}{\partial H_i}=0\ ,\ \dfrac{\partial LE_A}{\partial M_i}=0\ ,\ \dfrac{\partial LE_A}{\partial N_i}=0 \\
\dfrac{\partial LE_A}{\partial S_i}=0\ ,\ \dfrac{\partial LE_A}{\partial T_i}=0\ ,\ \dfrac{\partial LE_A}{\partial V_i}=0\ ,\ \dfrac{\partial LE_A}{\partial \lambda_j}=0
\end{cases}
\quad ,\quad i=1,2,3\ ,\ j=1,2,...,6 \tag{6-31}
$$

Equations (6-31) are solved using the Levenberg-Marquardt algorithm in MATLAB. After the unknown coefficients are found, they will be replaced in equations (6-28) to obtain the expressions for the deflection and the cross-section rotation of the stepped disk for Case A.

6.5. Case B: The Shim Stack Assembly

In this section, the shim stack, shown in Figure 6-3, is analyzed using FSDT. The main difference between a shim stack, referred to as Case B, and a stepped disk with the same geometry (Case A) is that in a shim stack the disks can slide on each other when the shim stack is deflected. As mentioned, the shear stresses in the stepped disk are higher than in a shim stack, as in a shim stack, the shear stresses at the interfaces between the disks are zero if friction is neglected.

The total potential energy of the shim stack (Case B) is obtained in the same way as for Case A because the same plate theory (FSDT) is used for both cases.

$$\Pi_B = \pi \int_r \left\{ \int_z z^2 dz \right\} F_1(\psi, \psi') dr + \pi \int_r \left\{ \int_z dz \right\} F_2(w', \psi) dr - \int_l^R q(r) w(r) dr \tag{6-32}$$

where, $F_1(\psi, \psi') = \frac{E}{1-\nu^2}\left(r(\psi')^2 + \frac{\psi^2}{r} + 2\nu\psi\psi' \right)$ and $F_2(w', \psi) = rGk(w' - \psi)^2$. The z-integrals in equation (6-32) are computed with the z-axis being the centroid of each section with constant thickness. Similar to Case A, a value of $k=0.833$ was used for the shear correction factor.

Because the disks can slide on each other independently, breaking the integrals of equation (6-32) is different from Case A (equation (6-27)). In Case B, for each section of the beam that deflects consistently, one deflection approximation function is assumed. Also, for each disk one cross-section deflection is assumed. This is shown in Figure 6-12.

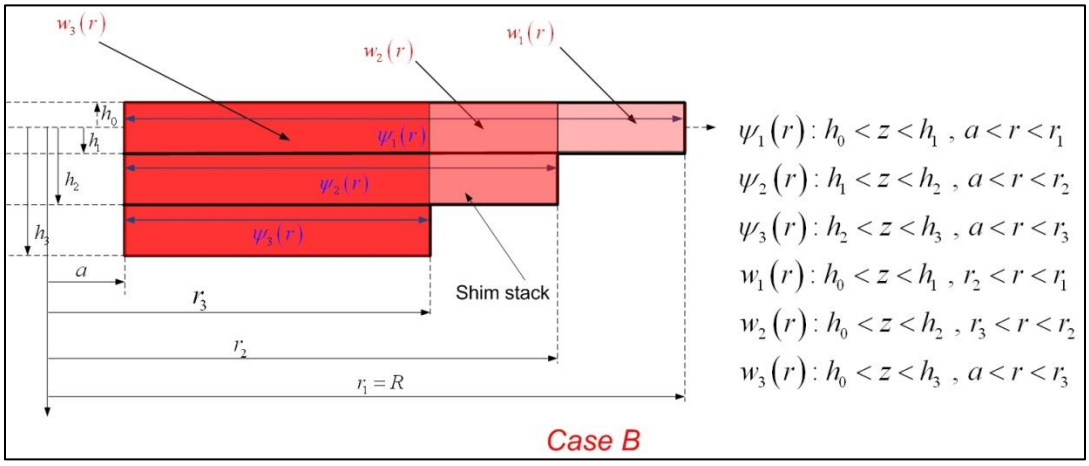

Figure 6-12 Range of various approximation functions used for case B

The form of the approximation functions are the same as those used for Case A, mentioned in equations (6-28).

Breaking the integrals in the total potential energy relation (6-32), in a proper manner, as shown in Figure 6-12, results in

$$\begin{aligned}\Pi_B = {} & \pi\int_a^{r_3}\left\{\int_{h_2}^{h_3} z^2 dz\right\}F_1(\psi_3,\psi_3')dr + \pi\int_a^{r_2}\left\{\int_{h_1}^{h_2} z^2 dz\right\}F_1(\psi_2,\psi_2')dr \\ & + \pi\int_a^{r_1}\left\{\int_{h_0}^{h_1} z^2 dz\right\}F_1(\psi_1,\psi_1')dr + \pi\int_a^{r_3}\left\{\int_{h_2}^{h_3} dz\right\}F_2(w_3',\psi_3)dr \\ & + \pi\int_a^{r_3}\left\{\int_{h_1}^{h_2} dz\right\}F_2(w_3',\psi_2)dr + \pi\int_{r_3}^{r_2}\left\{\int_{h_1}^{h_2} dz\right\}F_2(w_2',\psi_2)dr \\ & + \pi\int_a^{r_3}\left\{\int_{h_0}^{h_1} dz\right\}F_2(w_3',\psi_1)dr + \pi\int_{r_3}^{r_2}\left\{\int_{h_0}^{h_1} dz\right\}F_2(w_2',\psi_1)dr \\ & + \pi\int_{r_2}^{r_1}\left\{\int_{h_0}^{h_1} dz\right\}F_2(w_1',\psi_1)dr - \int_l^{r_3} q(r)w_3(r)dr - \int_{r_3}^{r_2} q(r)w_2(r)dr \\ & - \int_{r_2}^{r_1} q(r)w_1(r)dr\end{aligned} \tag{6-33}$$

The deflection and the cross-section rotation should be continuous over the length of each disk. Also, at $r = a$, the deflections and cross-section rotations are zero. This results in the following constraints that need to be satisfied

$$\begin{aligned} &at\ r = a \begin{cases} eqn_{B1} = w_3(a) = 0 \\ eqn_{B2} = \psi_1(a) = 0 \\ eqn_{B3} = \psi_2(a) = 0 \\ eqn_{B4} = \psi_3(a) = 0 \end{cases} \\ &at\ r = r_3 \left\{ eqn_{B5} = w_2(r_3) - w_3(r_3) = 0 \right. \\ &at\ r = r_2 \left\{ eqn_{B6} = w_1(r_2) - w_2(r_2) = 0 \right. \end{aligned} \tag{6-34}$$

In order to make the total potential energy in (6-33) minimum, and at the same time, satisfy the constraints (6-34), the Lagrange expression is formed as

$$LE_B = \Pi_B + \sum_{j=1}^{6} \lambda_j eqn_{Bj} \tag{6-35}$$

where, Π_B is the total potential energy of Case B, eqn_{Bj} are the constraints in (6-34), and λ_j are the Lagrange multipliers that are unknowns. Similar to Case A, the unknown coefficients and the Lagrange multipliers will be found by solving the system of equations

$$\begin{cases} \frac{\partial LE_B}{\partial A_i} = 0\,,\ \frac{\partial LE_B}{\partial B_i} = 0\,,\ \frac{\partial LE_B}{\partial C_i} = 0\,,\ \frac{\partial LE_B}{\partial D_i} = 0 \\ \frac{\partial LE_B}{\partial L_i} = 0\,,\ \frac{\partial LE_B}{\partial H_i} = 0\,,\ \frac{\partial LE_B}{\partial M_i} = 0\,,\ \frac{\partial LE_B}{\partial N_i} = 0 \\ \frac{\partial LE_B}{\partial S_i} = 0\,,\ \frac{\partial LE_B}{\partial T_i} = 0\,,\ \frac{\partial LE_B}{\partial V_i} = 0\,,\ \frac{\partial LE_B}{\partial \lambda_j} = 0 \end{cases} \quad ,\ i = 1,2,3\,,\ j = 1,2,...,6 \tag{6-36}$$

After solving the system of equations (6-36), the obtained values for the coefficients are replaced in equations (6-28) to obtain the expressions for deflection and cross-section rotation of the shim stack assembly as functions of *r*.

6.6. Numerical Results and Discussion

In this section, some sample cases are considered using the techniques discussed. Numerical results for Cases A and B are compared with each other. The material properties used for the sample problems are

$$\begin{aligned} E &= 2.9\times10^{7}\ psi \\ \nu &= 0.3 \\ G &= \frac{E}{2(1+\nu)}\ psi \\ k &= 0.833 \end{aligned} \tag{6-37}$$

and the pressure on the disk is assumed to be constant $p_0 = 300\ psi$ which implies a linearly varying loading q. This value of p_0 is encountered in real conditions for the pressure difference across the shim stack in a hydraulic damper. The geometrical values are shown in Figure 6-13. The stepped disk of Case A and the shim stack of Case B have the same boundary conditions, i.e. clamped at $r=a$ and free at $r=R$.

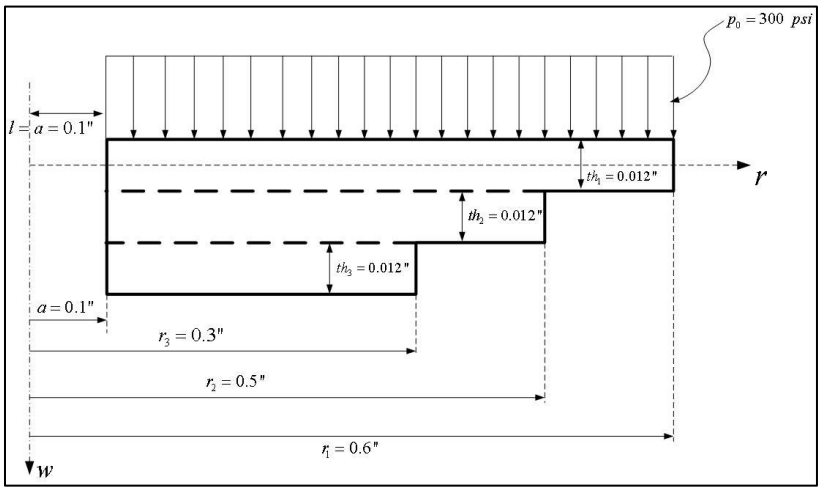

Figure 6-13 Numerical values for geometrical parameters in case A and B

The systems of equations in (6-31) and (6-36) are solved separately for the values shown in equations (6-37) and Figure 6-13, using MATLAB and are compared with each other. The Levenberg-Marquardt algorithm, a built-in solver of MATLAB, was used as the solver.

Figure 6-14 shows the predicted results for geometric values of Figure 6-13. The shim stack deflects 0.197 in. at the tip and the stepped disk deflects 0.030 in. at its tip. This shows that the shim stack deflects 6.5 times the stepped disk of similar geometry. This is a considerable difference and the approximations assuming that the shim stack behaves similar to a stepped disk are far from reality. Hence, there is a need to treat the shims individually as in Case B of this study. As mentioned, because the shims in the shim stack can slide on each other while deflecting, the shim stack can deflect more easily compared to a stepped disk of similar geometry.

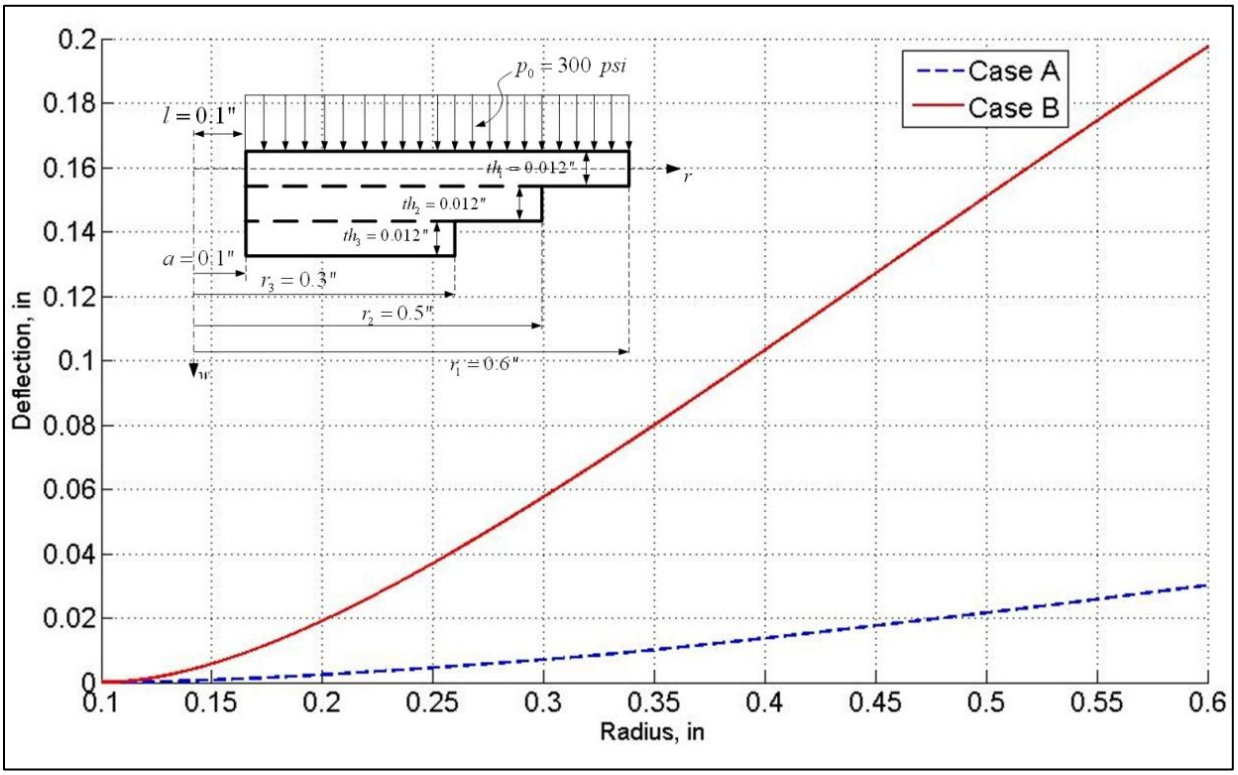

Figure 6-14 Deflection vs. radius, comparison between case A and B

6.7. Mathematical Model Verification

The mathematical model and the computer code are verified by running some sample examples. Let $r_3 = r_2 = r_1 = 0.6$ in. The stepped disk will become a single disk with a constant thickness of $th = 3 \times 0.012 = 0.036$ in. and the shim stack will become a stack of three similar disks. Other parameters

are similar to Figure 6-13. It is expected that the result from Case A be consistent with the analytical solution of a single disk with constant thickness (*th=0.036* in.*)*, as discussed in section 6.3. A comparison between the analytical deflection solution (6-24) and Case A is shown in Figure 6-15.

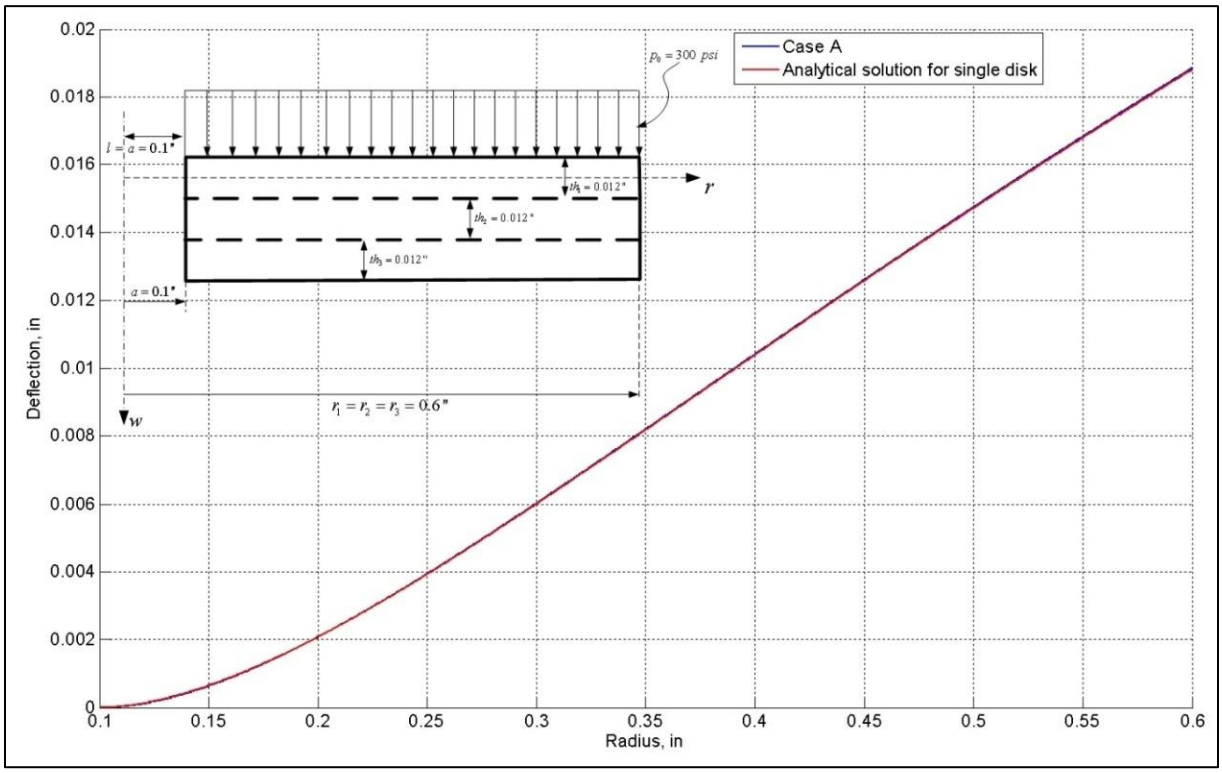

Figure 6-15 Comparison between the analytical solution and case A

When $r_1 = r_2 = r_3$, Case B resembles a stack of similar disks. It is expected that this stack deflects three times less than a single disk in the stack. This is confirmed by Figure 6-16 that contains the deflection plot of a single disk 0.012 in. thick but dividing the calculated deflection by a third.

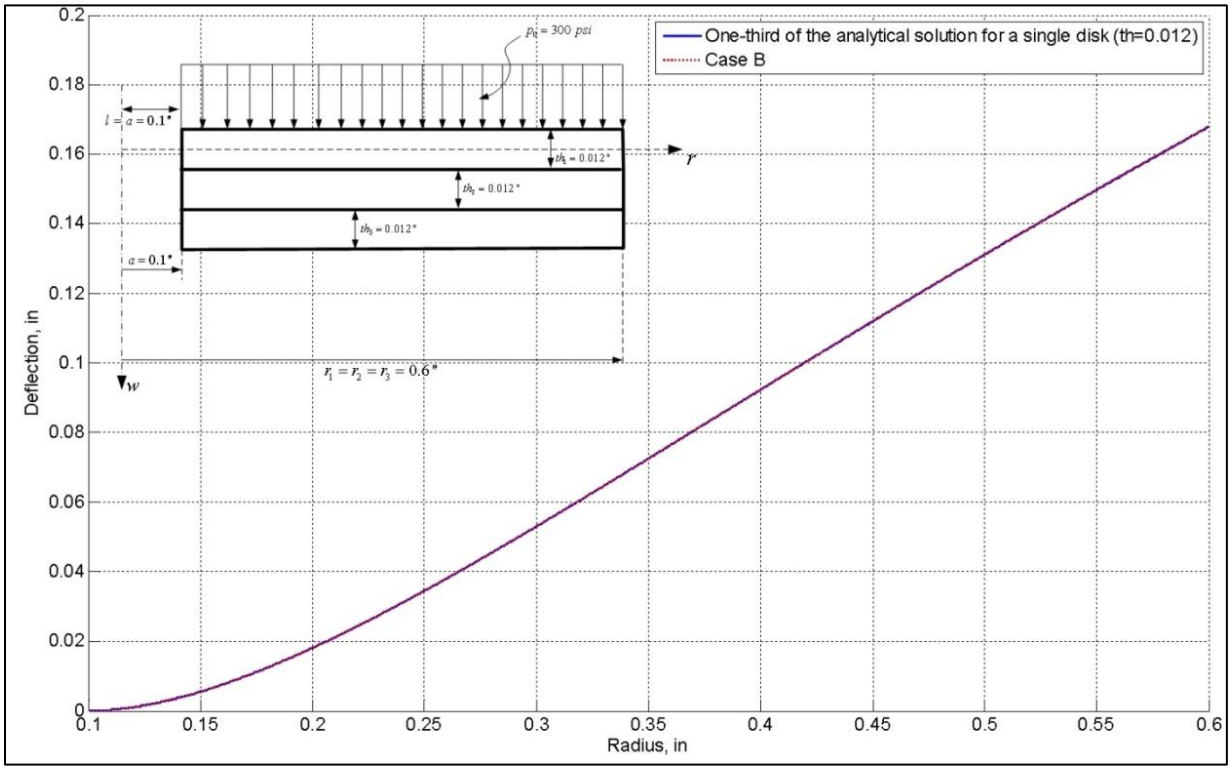

Figure 6-16 Comparison between the analytical solution and case B

If $r_2 = r_3 = a$, the stepped disk and the shim stack assembly both will reduce to a single disk with constant thickness of $th = 0.012$ in., inner radius of $a = 0.1$ in., and outer radius of $r_3 = 0.6$ in.. Hence, both Cases A and B should show the same deflection curve consistent with the analytical solution in section 6.3. This is shown in Figure 6-17.

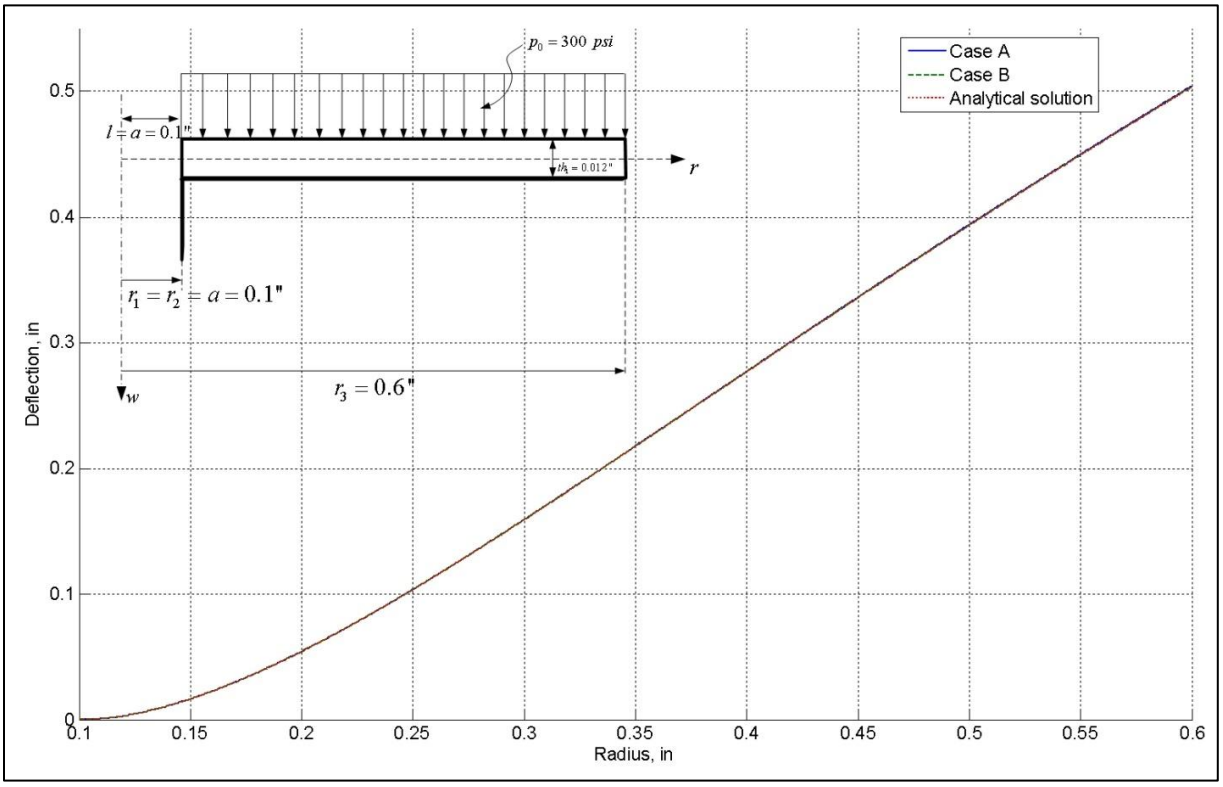

Figure 6-17 Cases A and B reduce to the analytical solution when only one disk is considered

6.8. Summary and Conclusion

A detailed mathematical model of a stack of annular disks was presented. The model was presented for a stack of three disks but it can be extended to include as many disks as desired. The First-Order Shear Deformation Theory (FSDT) was used. The model of the shim stack assembly was compared to the model of a single stepped disk with the same geometry. As shown in the numerical examples, the shim stack will deflect considerably more than a stepped disk with the same geometry under similar loading and boundary conditions. Hence, approximating a shim stack with a single stepped disk is not a good approximation and introduces a large amount of error in the analysis. The mathematical model was verified by examining some simple cases whose solutions are analytically known. The effects of the radial deflections were also studied using the CPT and it was observed that the radial deflections have not a

considerable effect on the axial deflections. The analysis developed here is expected to save designers considerable time and expense when considering hydraulic systems such as discussed.

Mathematical model validations will be presented along with the full hydraulic damper model in the next chapter.

Chapter 7
Modeling and Experimental Evaluation of Hydraulic Dampers

This chapter discusses detailed mathematical modeling and experimentation of hydraulic mono-tube dampers with emphasis on major damper components and their effects on damper performance. These components include, the shim stack assembly, by-pass and main piston orifices, gas chamber, floating piston, hydraulic oil, frictions between various components, and oil cavitation.

7.1. Introduction

There has been no published detailed analysis of the effects of shim stack design in a hydraulic damper to date. As mentioned in the previous chapter, other damper models have used simplifying assumptions for the shim stack deflection and effects of the shim stack have not been completely studied. In this chapter, various parameters affecting the nonlinear characteristics of mono-tube dampers such as the hysteresis region are studied. The model presented in this chapter can be used for design purposes and helps in developing controllable valvings based on shim stacks. It can also be used in designing controllable bypasses and tuning suspension systems. The mathematical model is validated by comparison against experimental test results carried out on an OHLINS CCJ 23/8 mono-tube damper, in CVeSS test facilities.

Suspension system design plays a vital role in handling and comfort characteristics of vehicles and requires detailed consideration. Dynamics of dampers and their characteristics strongly influence the

overall response of the suspension system to inputs. In order to have an acceptable level of comfort and handling, the dampers should be designed and tuned according to various criteria such as force, maximum displacement, speed, etc. Hence, having a good understanding of damper force characteristics and their dependencies on damper component geometric and material properties is extremely important.

A damper is commonly known as a shock absorber. This is, however, a misleading term to be used for dampers. The shocks are "absorbed" by tires and springs in a car. The main objective of a damper is to dissipate any energy of the vertical motion of the body or wheels due to control inputs, disturbance by rough roads, and driving conditions.

Dampers can be divided into two categories, friction (solid elements) and hydraulic (fluid elements). The friction dampers are distinct due to the associated problems such as sensitivity to water or oil contamination. Other problems with the first type are due to the Coulomb friction characteristics of dampers [142]. Hydraulic dampers are currently used in today's vehicles and are divided into mono- and twin-tube dampers, as shown in Figure 7-1. Each type has its own advantages and disadvantages. The mono-tube damper is more sensitive to stone impact because the active cylinder is exposed to the environment. The twin-tube damper usually must be mounted with the reservoir end on the suspension which makes the damper vulnerable to severe agitation [142]. Aeration is also a big problem for twin-tube dampers in severe conditions. Also, twin tube dampers dissipate heat less effectively.

In vibrations theory, dampers are modeled as linear systems that create force proportional to velocity. This over-simplification is mainly due to the mathematical difficulties that arise from considering dampers as non-linear systems. Assuming that a damper is linear results in equations for which the solutions are well understood and documented. Today's modern hydraulic dampers, however, exhibit non-linear characteristics. This makes the modeling aspect of damper analysis more important. Mathematical models should be able to accurately capture the non-linear effects seen in hydraulic dampers.

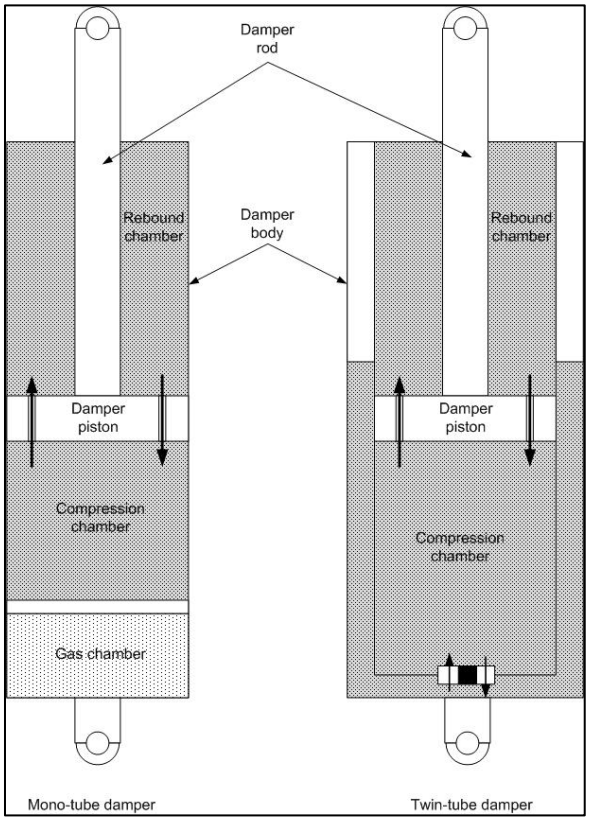

Figure 7-1 Schematics of mono-tube and twin-tube hydraulic dampers

This chapter will discuss a mathematical model in the context of a mono-tube hydraulic damper.

There have been numerous studies on different aspects of hydraulic dampers characteristics and performance [143-147]. The first detailed mathematical model of hydraulic dampers was developed by Lang [148]. He developed a model of a twin-tube hydraulic damper based on dynamic pressure flow characteristics of the shock absorber fluid and dynamic action of the valves. The physical phenomena considered by Lang were the compressibility of the hydraulic oil, elastic deflection of the damper cylinder, vaporization of the oil, leakage around the damper piston, and the flow through the valves. He used an analog computer for simulations and could obtain good levels of agreement between the mathematical model and test results. Reybrouck [149] worked on a non-linear parametric model of shock absorbers. The damper parameters were identified by fitting the measurements with the simulated results. Several identification iterations were needed to achieve good results. Lee [150] studied the numerical

modeling of mono-tube hydraulic dampers. The only parameters that needed to be determined from test data were four discharge coefficients. Lee used test data to validate his mathematical model. He used Finite Element Analysis to calculate the deflection of the disk valves on the damper piston. Lee's model, however, only considered one deCarbon disk valve on the damper piston and not a shim stack. Herr et. al. [151] introduced a Computational Fluid Dynamics (CFD) method, combined with a dynamic modeling technique to study the flow and performance of automotive shock absorbers. Duym et al. [152] developed a physical model of the hysteretic behavior in twin-tube dampers and found that besides the oil compressibility, the compressibility of a variable gas phase in the shape of bubbles are the root of the hysteretic effects. Duym et al. [153] and Surace et al. [154] worked on an alternative way of studying the characteristics of shock absorbers using the *"restoring force method"*. They were able to explain the hysteretic behavior of dampers in term of the dependency of the force on displacement in addition to damper velocity. Moon and Lee [155] developed a mathematical model for a displacement-sensitive shock absorber (DSSA) using fluid flow principles.

As the piston moves in a hydraulic damper, oil pressure is increased in one chamber and decreased in the other chamber across the piston. This pressure differential will force the oil to flow through the narrow passages in the piston, deflect the shim stack assembly mounted on the piston, and flow to the other pressure chamber, as shown in Figure 7-2. If the pressure drop across the shim stack is high, the damper force is large and the damper is stiff. If there is little pressure drop across the shim stack due to its small stiffness, the damper force is small. Therefore, the shim stack geometric and material properties play an important role in force characteristics of the hydraulic damper; an accurate model of the behavior of the shim stack relating the overall damper force characteristics to the shim stack properties can greatly improve the ability of mathematical models to predict the damper behavior.

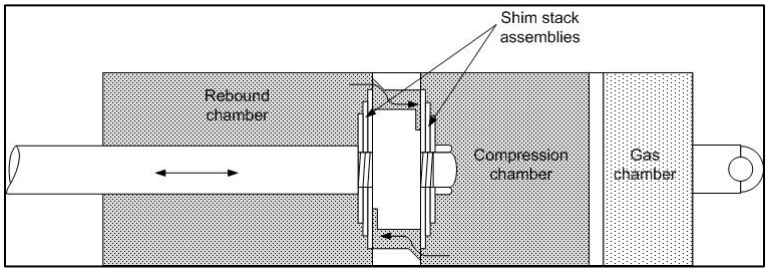

Figure 7-2 Schematics of shim stack assembly mounted on the piston in a mono-tube hydraulic damper

Currently, the effects of shim stack on damper behavior are determined experimentally. Different shim stack assemblies are mounted on damper piston and the damper is tested until the desired performance criteria are met. This procedure requires a lot of effort and takes a long time. The purpose of this part of the presented research is to develop a mathematical model that can be used as a design tool for studying different design parameters such as shim stack geometric and material properties, bypass valve properties, and overall damper and hydraulic oil properties.

Studies on damper modeling have used simplifying assumptions when dealing with shim stack deflections. They have either included only one disk in the model [150] or have approximated the shim stack deflection by a blow-off valve with an equivalent stiffness [147, 149, 152]. As a result, the effects of geometrical and material properties of shim stack on damper behavior could not be investigated.

The physical phenomena considered in this study, are the fluid flow dynamics of the hydraulic oil, compressibility of the hydraulic oil, deflection of the shim stack, dynamics of the floating piston, pressure chamber gas dynamics, and coulomb frictions between the moving components (main piston, rod sealing, and the floating piston) and the damper cylinder.

Continuity equations for compression and rebound chambers are written along with the state equation of the gas in the floating chamber, equation of motion of the floating piston, and deflection of the shim stack. Individual component equations are combined to form a system of non-linear ordinary differential equations (ODEs) and the resulted system is solved numerically.

The simulation results are validated against experimental test data on an OHLINS CCJ 23/8 mono-tube damper. The damper is fully rebuildable and has a fully extended length of 23in. and compressed length of 8in. A similar damper was used as the base damper for development of MR hybrid damper discussed in previous chapters. In experimental results presenter in this chapter, none of the damper components were altered. The only modification was the different shim stack assemblies mounted on a symmetric piston.

7.2. Damper Components Modeling

The individual components modeled are rebound and compression chambers, gas chamber, floating piston, bypass, and the shim stack, as shown in Figure 7-3. As the piston moves down into the damper cylinder (compression), the pressure in the compression chamber is increased and the pressure in the rebound chamber is decreased. This creates a pressure difference across the piston ($\Delta p = p_2 - p_1$). As a result of this pressure difference, the high pressure oil in the compression chamber flows through the piston. The oil pressure is dropped by some amount due to the restrictions from the orifices in the piston (P_5). In order to flow to the rebound chamber, the oil needs to deflect the shim stack. If the shim stack is not pre-deflected during damper assembling, the shim stack will deflect and the oil will pass to the rebound chamber. If the shim stack is pre-deflected, the oil pressure, p_5, should be high enough to overcome the pre-deflection. In this case, the shim stack is deflected and the oil flows to the rebound chamber. In addition to the flow of the oil through the piston orifices, depending on the opening of the bypass, some flow occurs through the bypass valve (Q_{22}). Due to the insertion of the piston rod into the damper cylinder, the floating piston is moved to compensate for the change in volume. When the piston is moving upwards (rebound), a similar sequence of events occurs.

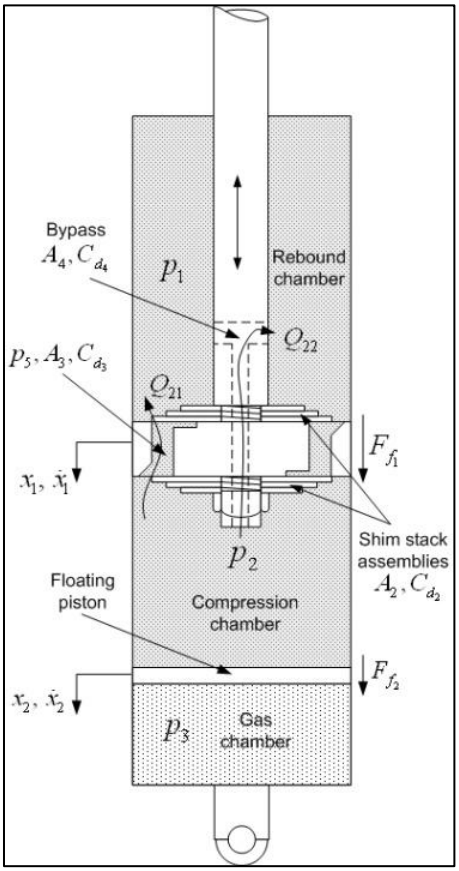

Figure 7-3 Modeled components

To develop the mathematical model of the hydraulic damper, continuity equations for chambers 1 and 2 are used along with the constitutive equation of the gas (usually Nitrogen) in the gas chamber and the equation of motion for the floating piston.

7.1. Continuity Equation for Chambers 1 and 2

The integral form of the continuity equation (conservation of mass) is

$$\frac{\partial \rho}{\partial t}\iiint d\forall + \iint \rho\left(\vec{V}.\vec{n}\right)dA = 0 \tag{7-1}$$

where, ρ is the oil density, t is time, $d\forall$ is an infinitesimal volume element, $\vec{V}$ is the velocity vector, and $\vec{n}$ is the normal vector to the infinitesimal area element dA. Using the appropriate control volumes for chambers 1 and 2, equation (7-1) results in

Chamber 1

$$Q_2 = \left(A_p - A_r\right)\frac{dx_1}{dt} + \frac{1}{\beta}\left(\left(A_p - A_r\right)x_1 + V_{10}\right)\frac{dp_1}{dt} \tag{7-2}$$

Chamber 2

$$-Q_2 = A_p\left(\frac{dx_2}{dt} - \frac{dx_1}{dt}\right) + \frac{1}{\beta}\left(V_{20} - A_p x_1 + V_{30} - V_3\right)\frac{dp_2}{dt} \tag{7-3}$$

where, $Q_2 = Q_{21} + Q_{22}$ is the volumetric flow of oil from the compression chamber to the rebound chamber, A_p is the area of the piston, A_r is the rod area, $\beta = \rho\frac{\partial p}{\partial \rho}$ is the bulk modulus of the oil, and V_{10}, V_{20}, V_{30} are the initial volumes of the rebound, compression, and gas chambers, respectively. The volumetric flow rate, Q_2, consists of two parts, the first part is the flow passed through the shim stack $\left(Q_{21}\right)$, and the second part is the flow passed through the bypass valve $\left(Q_{22}\right)$. The flow rate passed through the shim stack depends on the deflection of the shim stack and piston orifice area while the flow passed through the bypass depends on the area of bypass and its discharge coefficient. Each flow rate is related to the pressure difference via

$$Q_{21} = C_{d_2} A_2 \operatorname{sgn}\left(p_5 - p_1\right)\sqrt{\frac{2}{\rho}\left|p_5 - p_1\right|} \tag{7-4}$$

$$Q_{22} = C_{d_4} A_4 \operatorname{sgn}\left(p_2 - p_1\right)\sqrt{\frac{2}{\rho}\left|p_2 - p_1\right|} \tag{7-5}$$

where, C_{d_2} is the discharge coefficient of the area created by the deflection of the shim stack, A_2 is the area created by the shim stack deflection, C_{d_4} is the discharge coefficient of the bypass and A_4 is the bypass area. The piston orifice $\left(A_3,\ C_{d_3}\right)$ is in series with the shim stack, hence, the same flow $\left(Q_{21}\right)$ will pass through the piston orifice and the shim stack.

$$Q_{21} = C_{d_3} A_3 \operatorname{sgn}(p_2 - p_5)\sqrt{\frac{2}{\rho}|p_2 - p_5|} \tag{7-6}$$

Using equations (7-4) and (7-6), the intermediate pressure, p_5, can be found as

$$p_5 = \frac{C_{d_2}{}^2 A_2^2 p_1 + p_2 C_{d_3}{}^2 A_3^2}{C_{d_3}{}^2 A_3^2 + C_{d_2}{}^2 A_2^2} \tag{7-7}$$

Equation (7-7) is replaced in equations (7-4) and (7-5) to obtain the equations for flow rates in terms of the pressures in chambers 1 and 2, orifice areas, and discharge coefficients.

7.2. Constitutive Equation for the Gas Chamber

Assuming adiabatic process for the gas in chamber 3, the equation of state for the gas is

$$p_{30} V_{30}{}^n = p_3 V_3^n \tag{7-8}$$

where, p_{30} and V_{30} are the initial pressure and volume of chamber 3, respectively. If Nitrogen is used in chamber 3, *n=1.4*.

7.3. Equation of Motion of the Floating Piston

The equation of motion for the floating piston can be written as

$$(p_2 - p_3) A_p - \operatorname{sgn}\left(\frac{dx_2}{dt}\right) F_{f_2} = m_2 \frac{d^2 x_2}{dt^2} \tag{7-9}$$

where, m_2 is the floating piston mass, and F_{f_2} if the friction between the floating piston and the damper cylinder.

The acceleration term on the right hand side of equation (7-9) can be related to pressure by using the relation between the volume and location of the floating piston and the equation of state of the gas as mentioned in equation (7-8).

$$\frac{d^2x_2}{dt^2} = -\frac{1}{A_p}\left\{-\frac{1}{n}V_{30}p_{30}^{\frac{1}{n}}\left[-\left(\frac{1}{n}+1\right)p_3^{-\left(\frac{1}{n}+2\right)}\left(\frac{dp_3}{dt}\right)^2 + p_3^{-\left(\frac{1}{n}+1\right)}\frac{d^2p_3}{dt^2}\right]\right\} \tag{7-10}$$

Substituting equation (7-10) in equation (7-9) gives

$$\begin{aligned}&\left(p_2 - p_3\right)A_p - \operatorname{sgn}\left(\frac{dx_2}{dt}\right)F_{f_2} = \\ &\quad m_2\left\{-\frac{1}{A_p}\left\{-\frac{1}{n}V_{30}p_{30}^{\frac{1}{n}}\left[-\left(\frac{1}{n}+1\right)p_3^{-\left(\frac{1}{n}+2\right)}\left(\frac{dp_3}{dt}\right)^2 + p_3^{-\left(\frac{1}{n}+1\right)}\frac{d^2p_3}{dt^2}\right]\right\}\right\}\end{aligned} \tag{7-11}$$

7.4. State-Space Form of Equations

Denoting $p_4 = \frac{dp_3}{dt}$ and rearranging equations (7-2), (7-3), and (7-11), the system of non-linear ordinary differential equations can be written in a state-space form.

$$\begin{aligned}&\frac{dp_1}{dt} = \\ &\frac{-\beta}{x_1A_p - x_1A_r + V_{10}}\times \\ &\left(\begin{array}{l}-C_{d_2}A_2\operatorname{sgn}\left(p_2 - p_1\right)\sqrt{\frac{2}{\rho}\left|\frac{C_{d_3}{}^2A_3{}^2\left(p_2 - p_1\right)}{C_{d_3}{}^2A_3{}^2 + C_{d_2}{}^2A_2{}^2}\right|} + \ldots \\ \ldots C_{d_4}A_4\operatorname{sgn}\left(-p_2 + p_1\right)\sqrt{\frac{2}{\rho}\left|-p_2 + p_1\right|} + \frac{dx_1}{dt}\left(A_p - A_r\right)\end{array}\right)\end{aligned} \tag{7-12}$$

$$\begin{aligned}&\frac{dp_2}{dt} = \\ &\frac{-\beta p_3^{\frac{1}{n}}}{n\left(-V_{20}p_3^{\frac{1}{n}} + A_px_1p_3^{\frac{1}{n}} - V_{30}p_3^{\frac{1}{n}} + p_{30}^{\frac{1}{n}}V_{30}\right)}\times \\ &\left(\begin{array}{l}nC_{d_2}A_2\operatorname{sgn}\left(-p_2 + p_1\right)\sqrt{\frac{2}{\rho}\left|\frac{C_{d_3}{}^2A_3{}^2\left(-p_2 + p_1\right)}{C_{d_3}{}^2A_3{}^2 + C_{d_2}{}^2A_2{}^2}\right|} + \ldots \\ \ldots nC_{d_4}A_4\operatorname{sgn}\left(-p_2 + p_1\right)\sqrt{\frac{2}{\rho}\left|-p_2 + p_1\right|} - V_{30}p_{30}^{\frac{1}{n}}p_3^{-\frac{1+n}{n}}p_4 + nA_p\frac{dx_1}{dt}\end{array}\right)\end{aligned} \tag{7-13}$$

$$\frac{dp_3}{dt} = p_4 \tag{7-14}$$

$$\frac{dp_4}{dt} = \frac{1}{m_2 V_{30} p_{30}^{\left(\frac{1}{n}\right)} p_3^{\left(-\frac{1+n}{n}\right)} n} \left[\begin{array}{l} A_p{}^2 n^2 p_2 - A_p{}^2 n^2 p_3 - \mathrm{sgn}(p_4) F_{f_2} n^2 A_p + \ldots \\ \ldots m_2 V_{30} p_{30}^{\left(\frac{1}{n}\right)} p_3^{\left(-\frac{1+2n}{n}\right)} p_4{}^2 + m_2 V_{30} p_{30}^{\left(\frac{1}{n}\right)} p_3^{\left(-\frac{1+2n}{n}\right)} p_4{}^2 n \end{array} \right] \tag{7-15}$$

The above system of nonlinear ODEs should be solved numerically for $p_1, p_2, p_3,$ and p_4. The discharge coefficients are determined experimentally for various orifice configurations and are available in the literature. More accurate values for the discharge coefficients are obtained from matching the experimental data with the simulations results.

7.5. Shim Stack Deflection

The system of ODEs in equations (7-12)-(7-15) cannot be solved until the relation between A_2 and pressure difference across the piston is determined. This relationship is obtained by considering the deflection of the shim stack. The area created by the deflection of the shim stack depends on the pressure difference across the shim stack $(p_5 - p_1)$, the geometric and material properties of the shim stack, as well as number of shims in the shim stack assembly. For this study, a shim stack consisting of three disks is considered. The analysis can be expanded to include more shims in the shim stack assembly. Also, the shim stack configuration and piston valvings are assumed to be the same in compression and rebound.

Consider the shim stack assembly shown in Figure 7-4. The shim stack consists of three disks stacked one upon the other and each clamped at $r=a$ (not necessarily the inner radius of the shims) and free at $r=R$. A loading, $q(r)$, per unit length in r-direction, is acting on the shim stack causing it to deflect in positive z-direction (downward).As the shim stack is deflected, the disks bend due to the force transmitted to them

through the upper disks. Neglecting the friction between the disks, the disks are considered to easily slide on each other as they deflect.

Chapter 6 presented a detail analysis of the deflection of shim stack assemblies. The analysis presented here for hydraulic model of dampers uses the shim stack model presented in Chapter 6 to calculate the deflection of the shim stack due to the pressure difference across the stack cause by the motion of the damper piston.

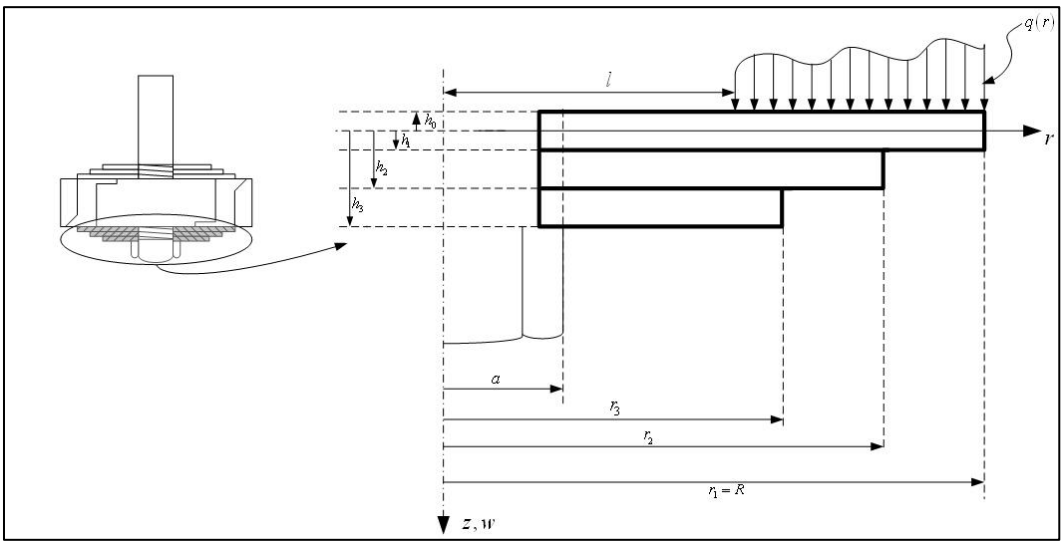

Figure 7-4 Shim stack assembly geometrical parameters

After using the shim stack model presented in Chapter 6, the area created by the deflection of the shim stack is calculated as

$$A_2 = 2\pi \times R \times w_1(R) \tag{7-16}$$

where, $w_1(R)$ is the tip deflection of the shim stack assembly, as shown in Figure 7-4.

7.6. Total Damper Force

Figure 7-5 shows the free body diagram of the damper cylinder. The forces acting on the damper cylinder are due to the pressure in chambers 1 and 2, and the friction forces. F_{f_1} is the friction between the moving

parts (damper piston and damper rod) and the damper cylinder. The Coulomb friction force between the floating piston and the damper cylinder, F_{f_2}, was neglected for this study because its effect compared to the total damping force is small.

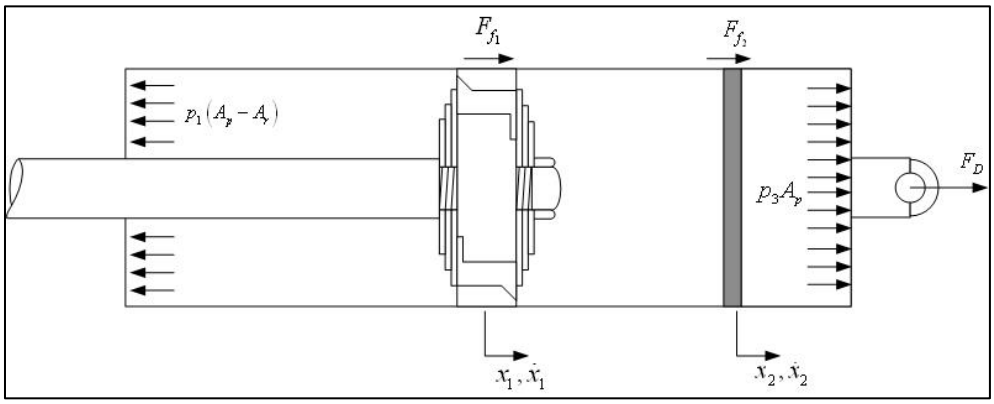

Figure 7-5 Free body diagram for the damper cylinder

From the free body diagram shown in Figure 7-5, the total damper force can be determined as

$$F_D = p_1\left(A_p - A_r\right) - p_3 A_p - F_{f_2}\,\mathrm{sgn}\left(\dot{x}_2\right) - F_{f_1}\,\mathrm{sgn}\left(\dot{x}_1\right) \tag{7-17}$$

After the system of equations (7-12)-(7-15) is solved using appropriate initial conditions, the values of pressure in chambers 1 and 3 will be inserted in equation (7-17) as functions of time and the total damper force can be determined.

7.7. Computer Programming

The system of equations (7-12)-(7-15) is solved using MATLAB. A GUI code is written in MATLAB that takes in the values of the geometric and material properties of different components of the damper and solves for the pressures $p_1, p_2,$ and p_3 as functions of time, as shown in Figure 7-6.

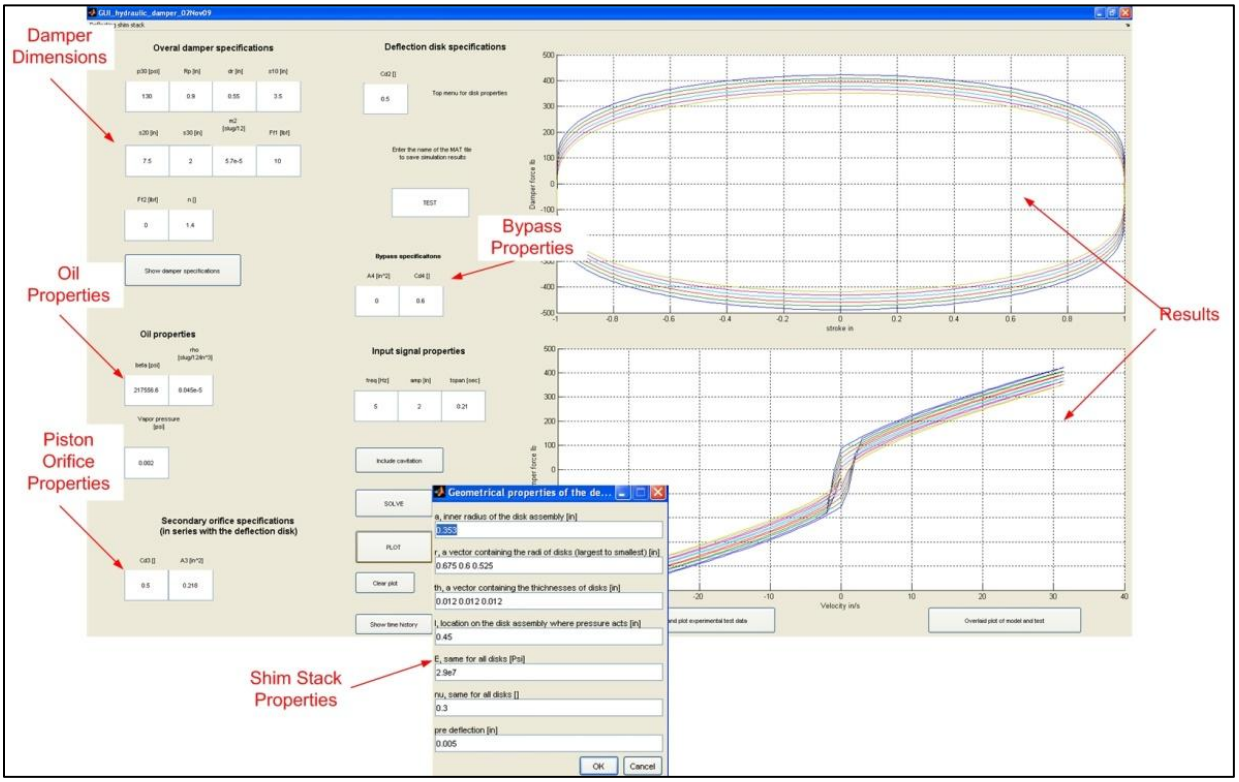

Figure 7-6 Mathematical model computer code layout

At each time step, the pressure of chambers and, accordingly, the pressure differential across the piston are known. Using the shim stack analysis, the deflection of the shim stack can be calculated which results in the area created by the deflection of the shim stack. Using this area and the given bypass opening, the flow of oil from chamber 1 to chamber 2 or vice versa can be determined. Hence, the pressures in the chambers can be determined at the next time step.

At each time step, the system of equations for shim stack deflection calculation should be solved (Equations (7-18)) . This takes a long time and reduces the overall speed of the damper model code.

$$\begin{cases} \dfrac{\partial LE}{\partial A_i}=0\,,\ \dfrac{\partial LE}{\partial B_i}=0\,,\ \dfrac{\partial LE}{\partial C_i}=0\,,\ \dfrac{\partial LE}{\partial D_i}=0 \\ \dfrac{\partial LE}{\partial L_i}=0\,,\ \dfrac{\partial LE}{\partial H_i}=0\,,\ \dfrac{\partial LE}{\partial M_i}=0\,,\ \dfrac{\partial LE}{\partial N_i}=0 \\ \dfrac{\partial LE}{\partial S_i}=0\,,\ \dfrac{\partial LE}{\partial T_i}=0\,,\ \dfrac{\partial LE}{\partial V_i}=0\,,\ \dfrac{\partial LE}{\partial \lambda_j}=0 \end{cases} \quad,\quad i=1,2,3\,,\ j=1,2,...,6 \tag{7-18}$$

As discussed in Chapter 6, the solution for the system of equations (7-18) gives the deflection of the shim stack at each time step as function of radial coordinate, *r*. But, in the damper model, only the tip deflection is needed to calculate the area, A_2. This means that a large amount of computation is performed that is not necessary. However, there is no way to find the shim stack tip deflection without solving for all the unknown coefficients in (7-18). As will be discussed in the next section, the tip deflection of the shim stack assembly varies linearly with the pressure difference across the shim stack. Therefore, this linear relationship can be found for the shim stack prior to solving the damper model. After this relationship is found, it can be used as a replacement for system of equations (7-18). This removes the necessity for solving the shim stack deflection at each time step and greatly increases the speed of the computer code.

7.8. Experimental Results and Discussion

The mathematical model discussed in section 7.2, is implemented into a computer code in MATLAB as outlined in Section 7.7 and is validated against experimental test data. An OHLINS CCJ 23/8 mono-tube damper is used as the base damper for tests. A symmetric piston was used and a shim stack consisting of three disks was mounted on each side of the piston. For consistency, all experiments were carried out at 85^{o} F damper body temperature.

A Roehrig 2K-EMA damper dynamometer, available at testing facilities in the Center for Vehicle Systems and Safety (CVeSS), was used to test the damper. Figure 7-7 shows the damper and the dynamometer used in this study.

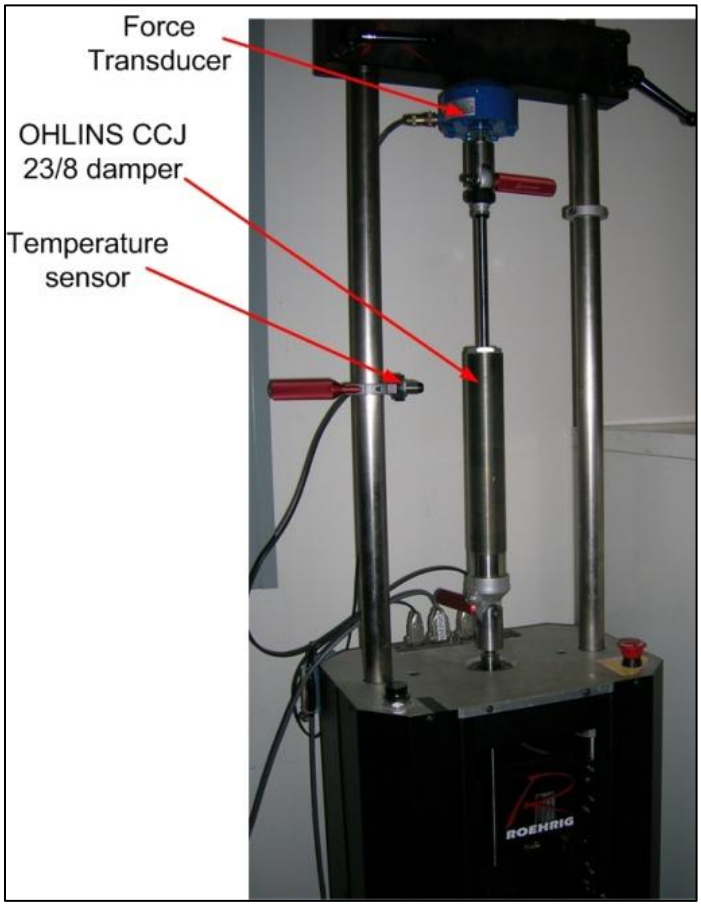

Figure 7-7 OHLINS mono-tube damper mounted on the Roehrig dynamometer

The geometric and material parameters used in the model are based on the OHLINS damper. These parameter values are shown in Figure 7-8.

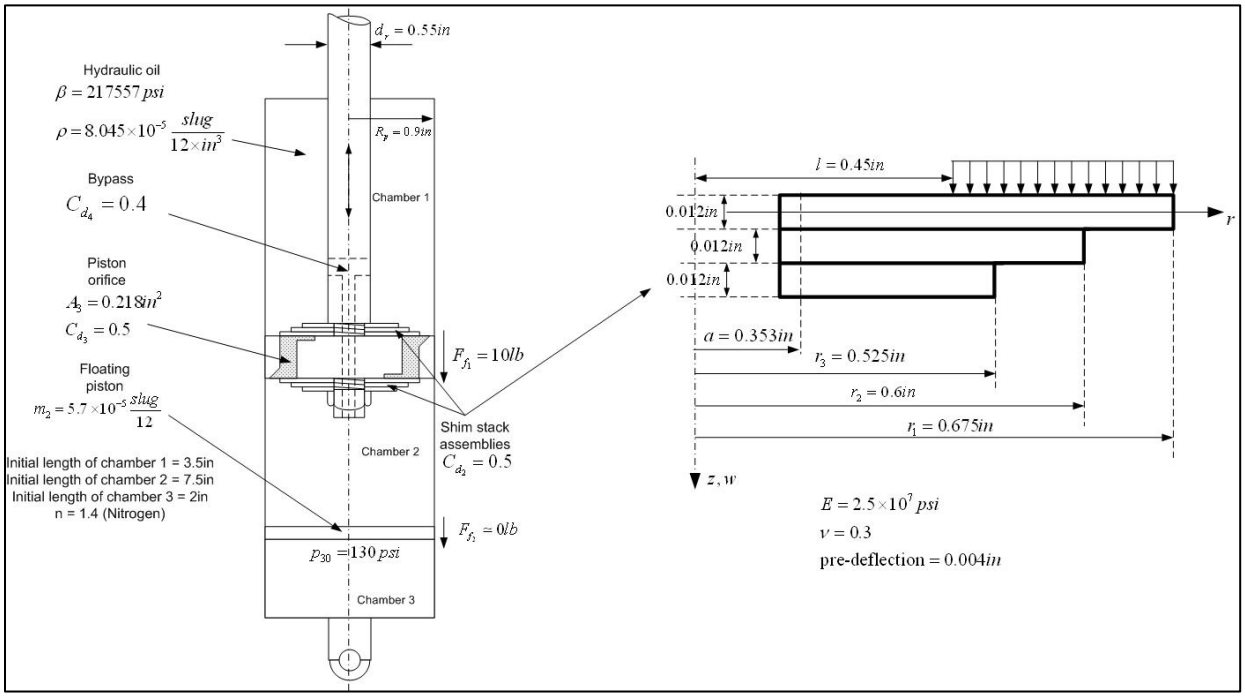

Figure 7-8 Numerical values for different parameters used in the mathematical model

7.9. Shim Stack Tip Deflection

As mentioned, solving the shim stack deflection problem at each time step is not efficient and reduces the speed of the code. Figure 7-9 shows the tip deflection, $w_1(R)$, as a function of the pressure difference obtained from solving the system of equations (7-18) for few pressure difference values.

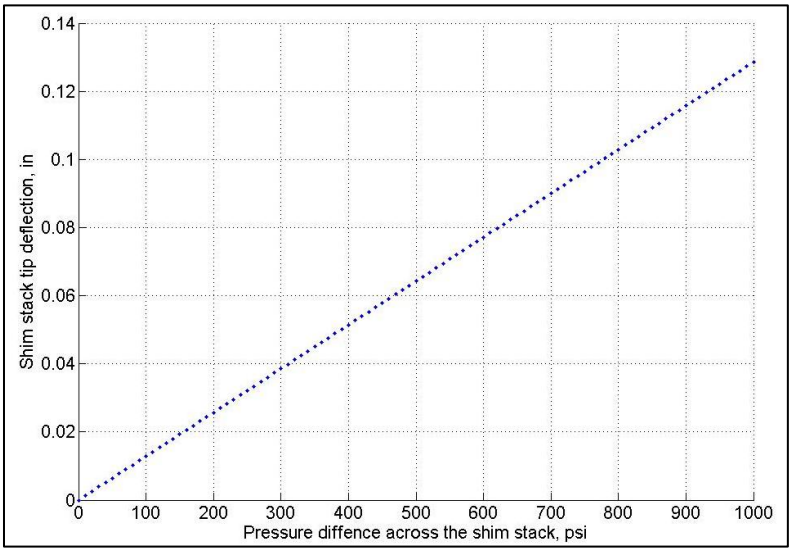

Figure 7-9 Shim stack tip deflection vs. pressure difference across the shim stack

As can be seen in Figure 7-9, the relationship between the shim stack tip deflection and pressure difference is a linear relationship. The computer code finds this linear relationship before solving the damper state-space ODEs. Then, the obtained linear equation is used to find the tip deflection and, consequently, the area created by the shim stack deflection at each time step.

7.10. Numerical Results and Experimental Data Comparison

The mathematical model and the experimental data obtained from tests on the OHLINS damper are compared for different values of stroke, frequency, and bypass openings. Good level of agreement is obtained. A typical comparison is shown in Figure 7-10 for two bypass area values, $A_4 = 0\ in.^2$ and $A_4 = 0.0312\ in.^2$ at a frequency of 5 Hz and 2 in. stroke. As shown, the mathematical model can accurately predict the behavior of the damper. Compression displacement, velocity, and force are negative. A sine wave signal was used as the input displacement to the damper.

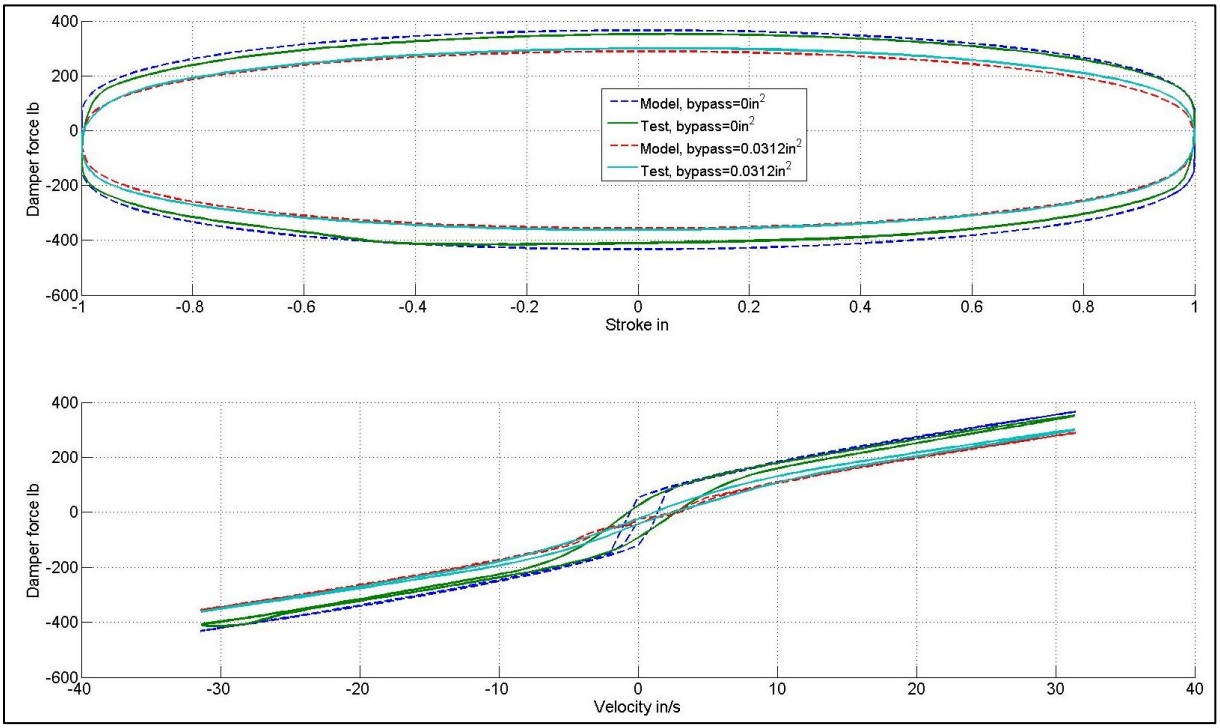

Figure 7-10 Comparison between model and experimental data at two bypass opening values, 2 in. stroke, 5 Hz

Figure 7-11 shows the simulation results for the shim stack tip deflection magnitudes versus time at 5 Hz for one cycle. As can be seen, if the bypass is closed, the shim stack opens earlier than when the bypass is open. In both cases, some time is elapsed before the shim stack deflects due to the pre-deflection of the shim stack. When the bypass is closed, the shim stack needs to deflect more to open enough area for the oil to pass according to the piston velocity.

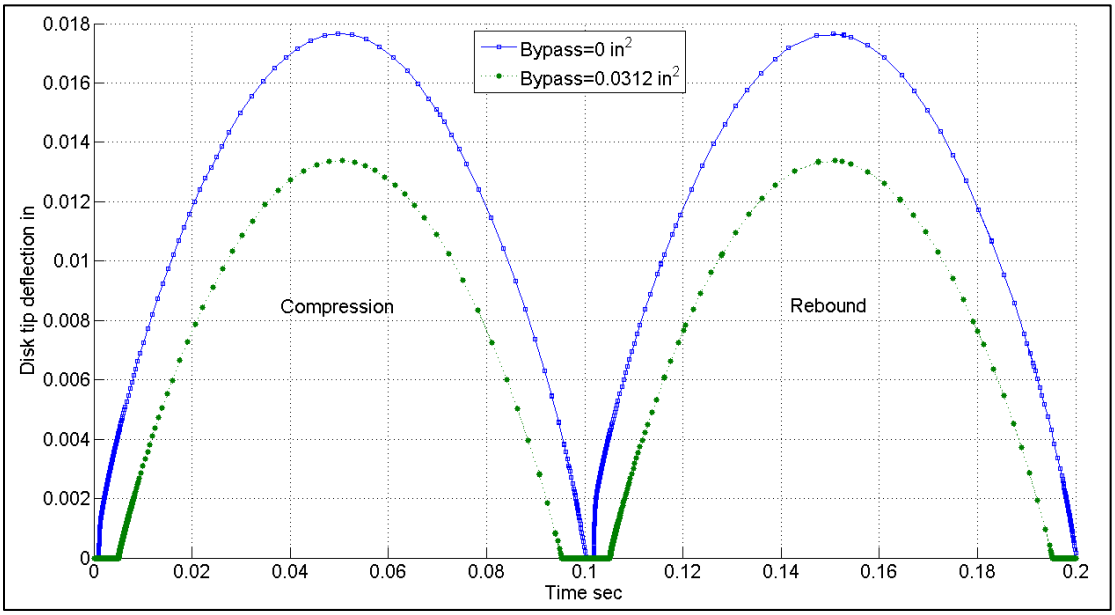

Figure 7-11 Shim stack tip deflection vs. time for two values of bypass openings

7.11. Shim Stack Parametric Study

As one of the main objectives of this study is to investigate the effects of the shim stack on the characteristics of the damper, this section is devoted to discuss these effects. The effects of geometric parameters such as thickness and pre-deflection of the shim stack are studied as well as material properties, such as the shims' stiffness. The stroke and frequency for the results that will be presented are 2 in. and 5 Hz, respectively. Other damper parameters are the same as those shown in Figure 7-8. Figure 7-12 shows the effect of shim stack disk thickness on the damper force characteristics. The thicknesses of three disks in the shim stack assembly were chosen to be equal for the results shown in Figure 7-12. The pre-deflection is 0.004 in. and the bypass valve is closed.

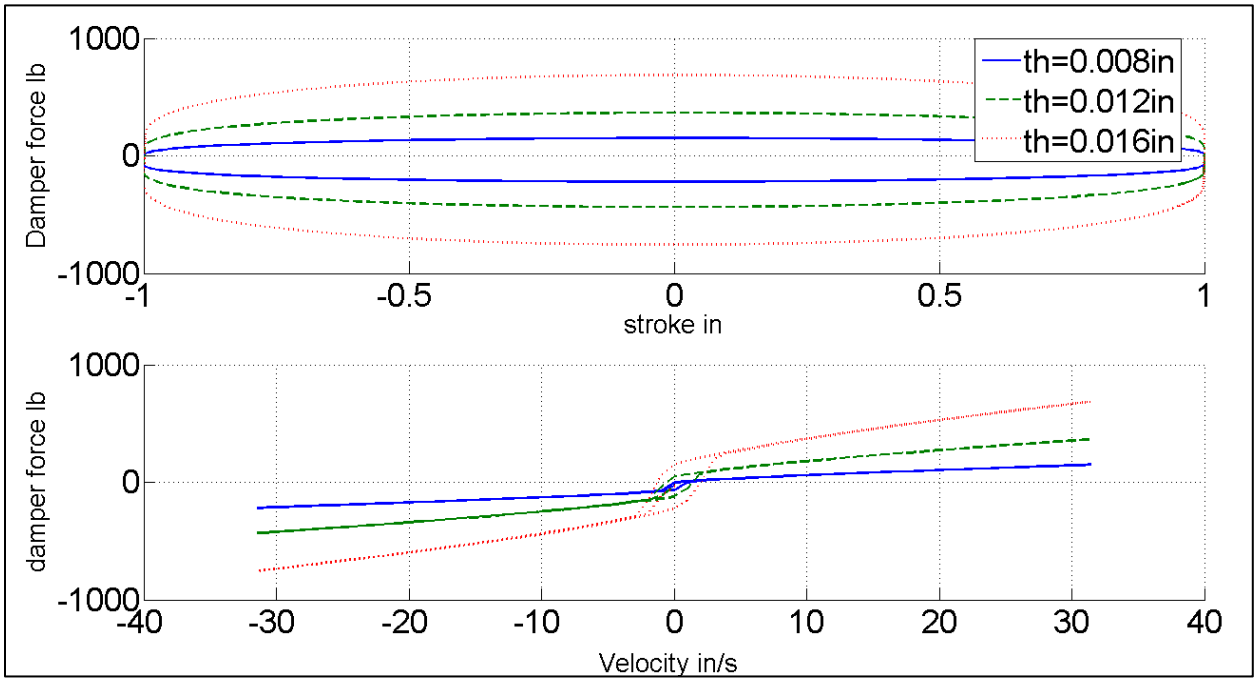

Figure 7-12 Effect of disk thickness on damper force curves for 0.004 in. pre-deflection and closed bypass

As shown in Figure 7-12, increasing the thickness of the shims makes the damper stiffer. The slope of the force-velocity curve increases as the shim thickness is increased. Also, due to existence of pre-deflection and the bypass being closed, the hysteretic region around zero velocity crossing increases as the shim thickness is increased. If the bypass is opened or the pre-deflection is set to zero, the hysteretic region will collapse. Figure 7-13 shows the same results for the shim stack thickness study when the shim stack is not pre-deflected. As can be seen, the hysteretic region observed in Figure 7-12 is collapsed to a constant area independent of shim thickness. Hence, the hysteretic region around zero velocity crossing is a function of shim thickness if the shim is pre-deflected or the bypass is closed.

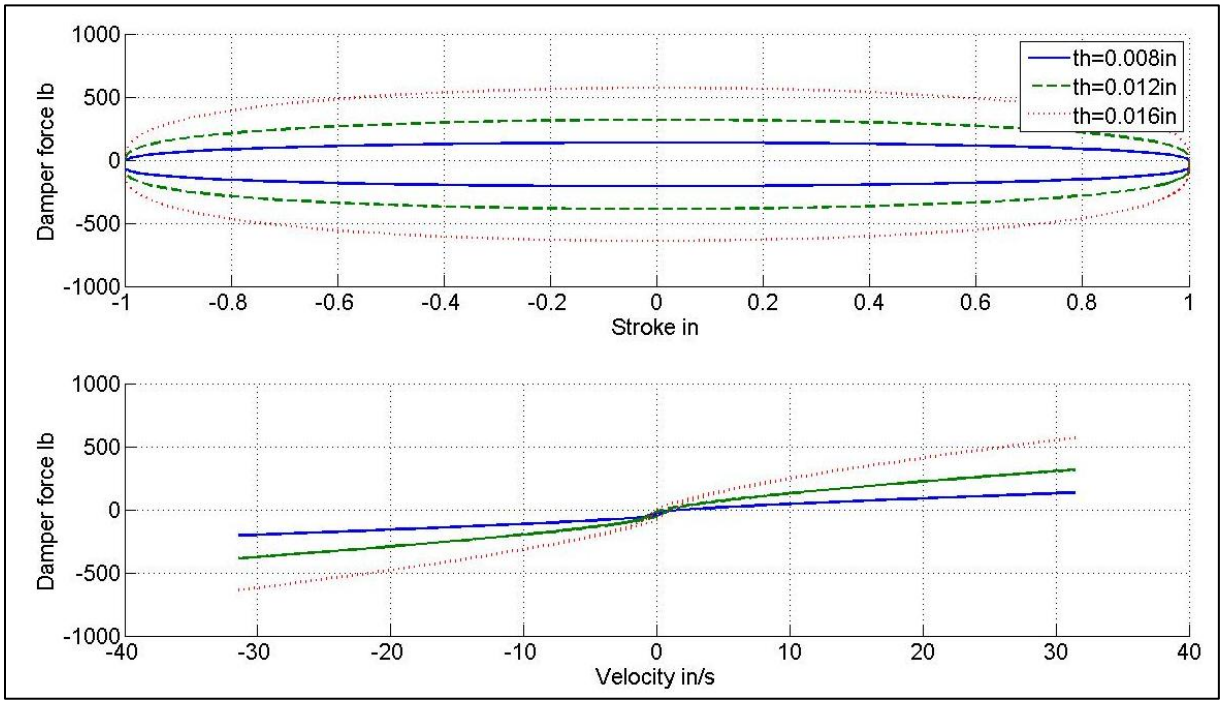

Figure 7-13 Effect of disk thickness on damper force curves for zero pre-deflection and closed bypass

Figure 7-14 shows the effect of the shim stack pre-deflection on the damper behavior. As the pre-deflection is increased, a larger pressure difference is needed across the shim stack to deflect it. As a result, the damper becomes stiffer. However, unlike the shim thickness, pre-deflection does not change the slope of the force-velocity curves, as shown in Figure 7-14. It should also be mentioned that adding pre-deflection increases the size of the hysteretic region around zero velocity crossing, similar to increasing the shim thickness. Again, this hysteretic region is collapsed if the bypass is opened.

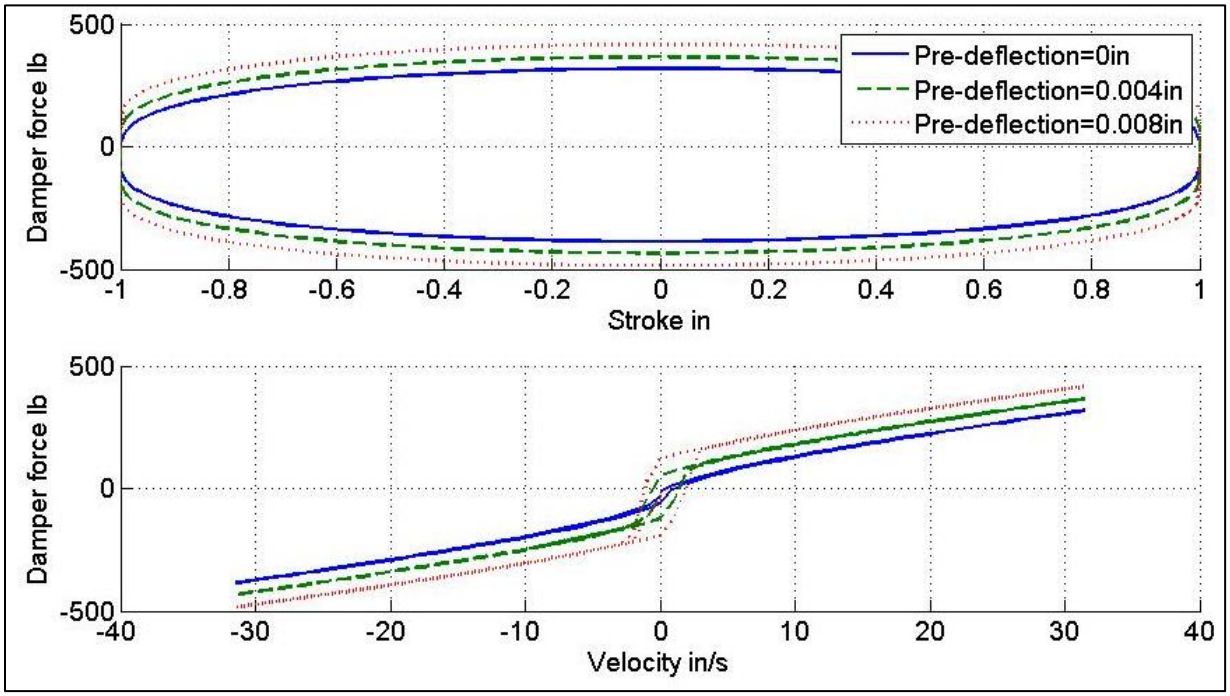

Figure 7-14 Effect of shim stack pre-deflection on damper force curves for closed bypass

If the bypass is open, the resistance to oil flow is minor at low velocities because small amounts of oil can pass through the bypass area. Hence, the damping force is small at zero velocity crossing. Additionally, because the bypass area is constant during damper strokes, unlike the area created by shim stack deflection, the damper force curves are concaved upwards which is similar to the force characteristics of fluid flow through orifices with constant area. As soon as the piston velocity is high enough to create pressure differences large enough to deflect the shim stack, the force curves concavity will change. Also, at low velocities, the damper force characteristics are independent of the shim stack pre-deflection if the bypass is open. This can be seen in Figure 7-15.

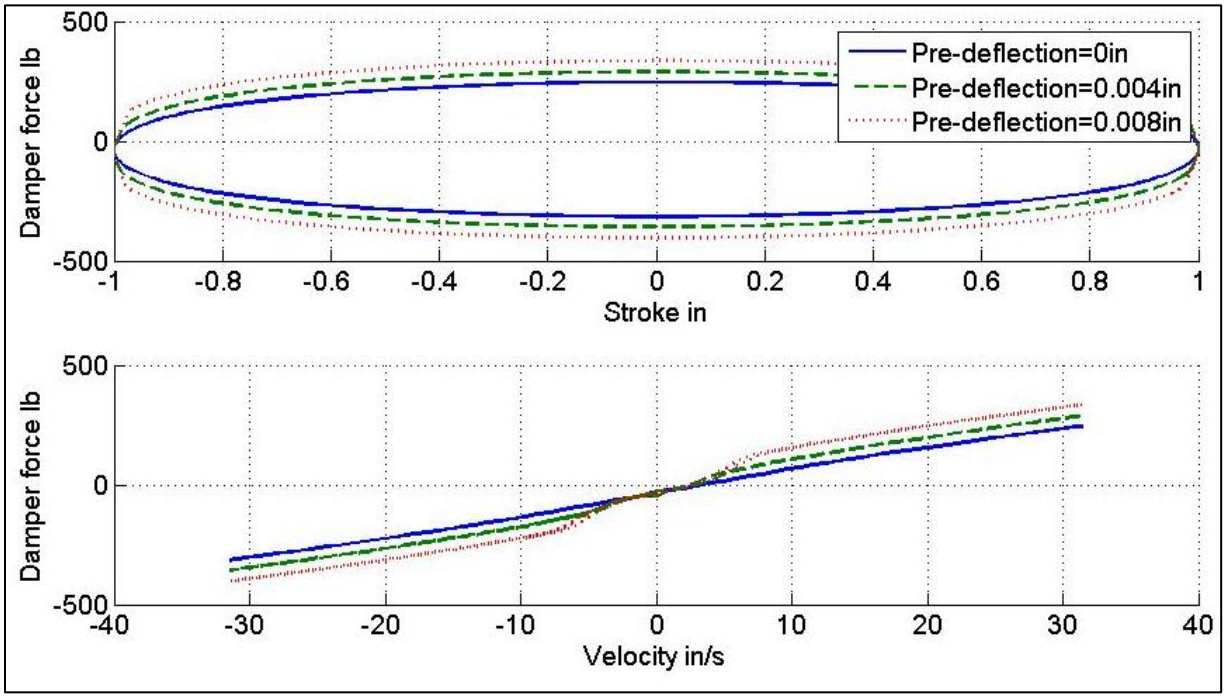

Figure 7-15 Effect of shim stack pre-deflection of damper force curves for open bypass

The material properties of the disks in the shim stack also play an important role in the overall damper force characteristics. Figure 7-16 shows the effect of the shim stack material properties on the damper performance for zero shim stack pre-deflection and closed bypass. All three shims in the shim stack are assumed to be made of the same material. As shown in Figure 7-16, as the shims become stiffer, the damper force is also increased and the damper becomes stiffer. As the Young's modulus of the shim stack is increased, the slope of the force-velocity curve increases. Also note that the shim stack's Young's modulus has no effect on the damper behavior in the low velocity regions. However, if the shim stack is pre-deflected, the stiffness of shims becomes important in low velocity regions. This is shown in Figure 7-17. When the shim stack is pre-deflected, the hydraulic oil needs to build up enough pressure in order to overcome the pre-deflection of the shim stack at zero velocity crossing which depends on both the pre-deflection and shims' modulus (E).

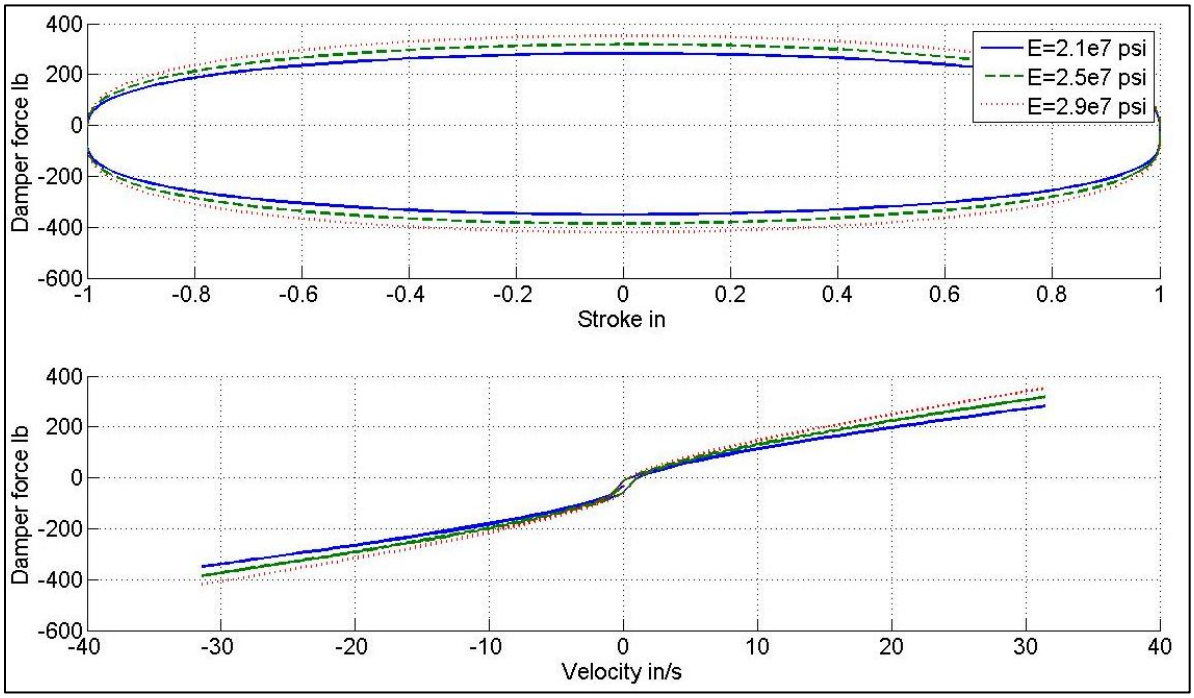

Figure 7-16 Effect of shim stack material properties on damper force curves for zero pre-deflection and closed bypass

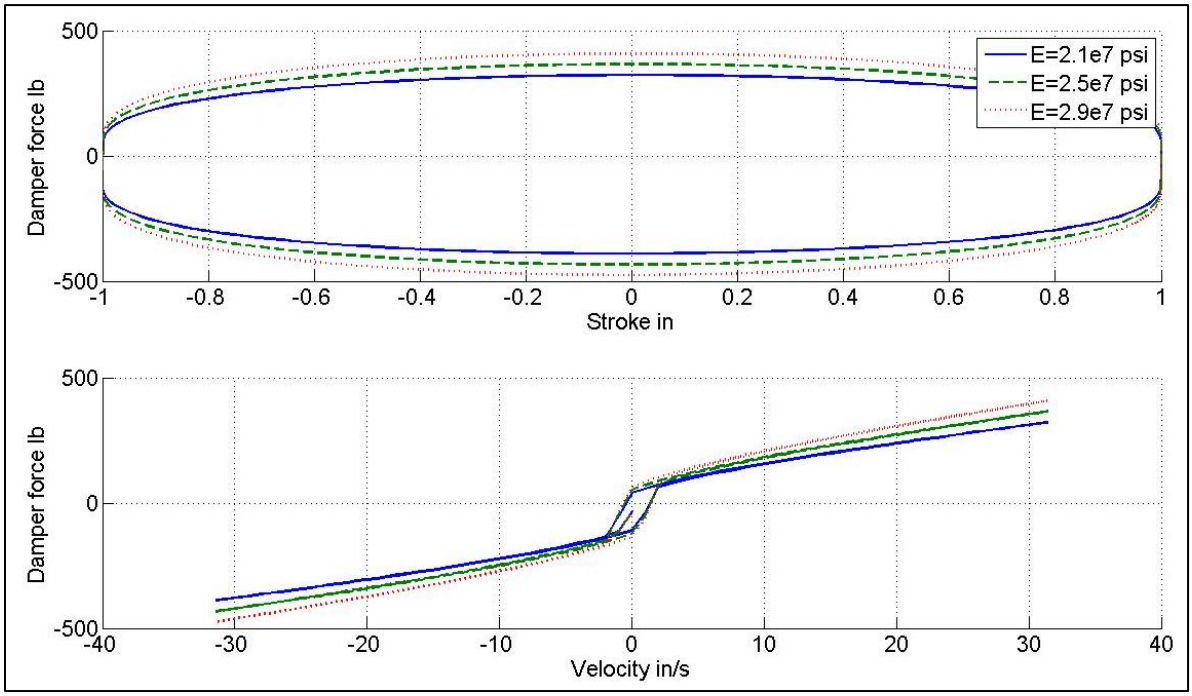

Figure 7-17 Effect of shim stack material properties on damper force curves for 0.004in pre-deflection and closed bypass

7.12. Effects of Other Damper Parameters

Using the mathematical model developed, effects of other damper parameters can be studied on the performance of the damper. For example, the bulk modulus and density of oil can be studied for their effects on damper behavior. Figure 7-18 shows the effect of the oil's bulk modulus on damper behavior. As the modulus is increased, the oil becomes less compressible and hence, the hysteretic region is reduced.

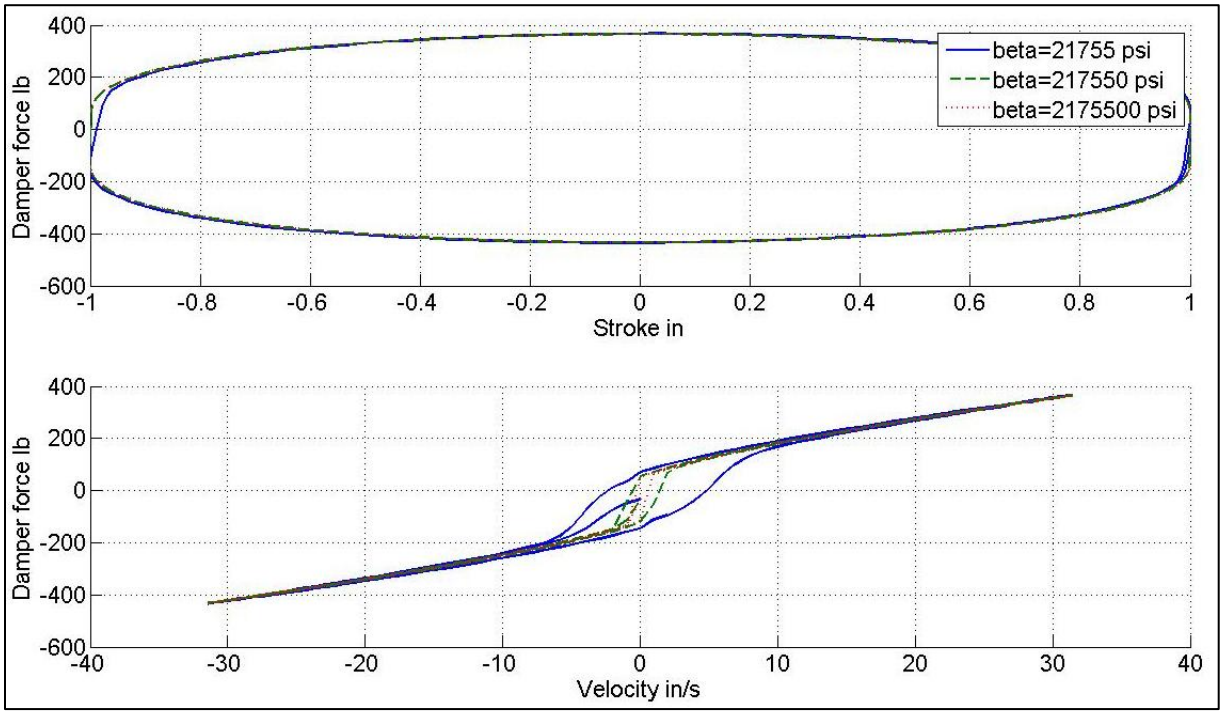

Figure 7-18 Effect of oil bulk modulus on damper force curves for 0.004 in. pre-deflection and closed bypass

Effects of oil density on damper performance are shown in Figure 7-19. As shown, oil density changes the slope of the force-velocity curve similar to shim stack stiffness and shims' thickness shown in Figure 7-16 and Figure 7-12, respectively. At low velocities, damper characteristics are not affected by the oil density. This is because the inertial forces due to the oil density are not significant at low velocities.

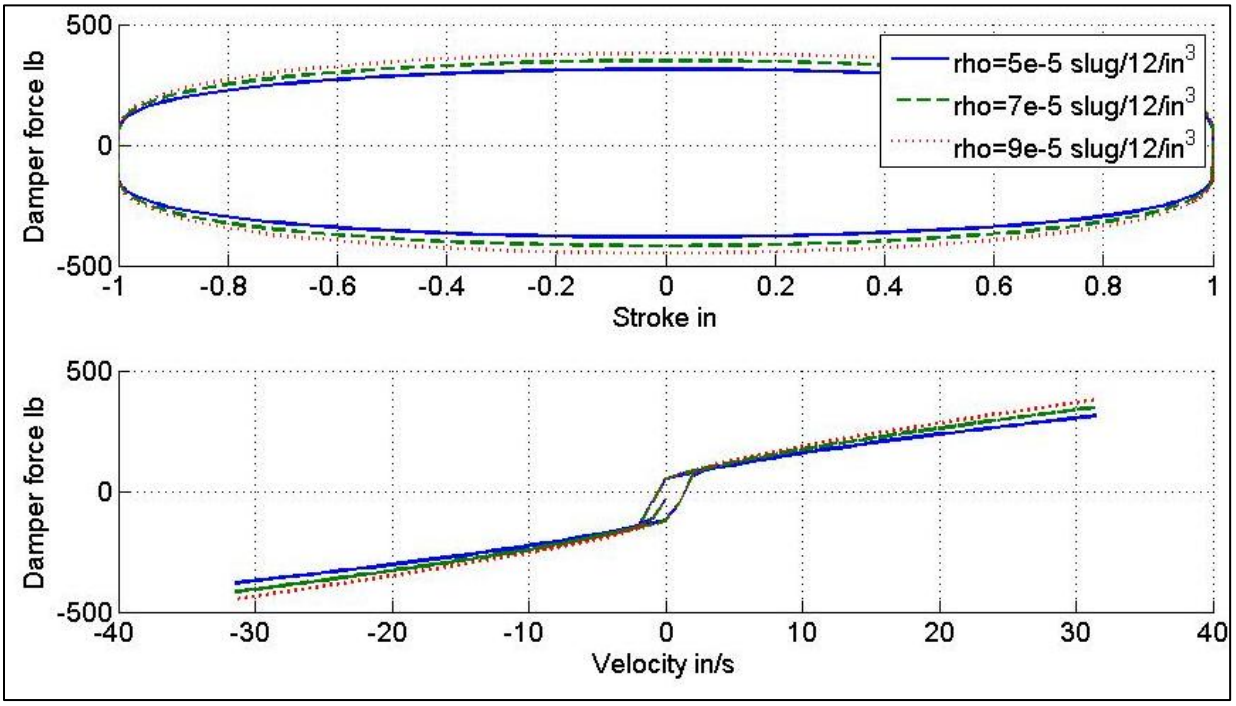

Figure 7-19 Effect of oil density on damper force curves for 0.004 in. pre-deflection and closed bypass

7.13. Summary and Conclusion

In Chapter 7, a detailed mathematical model for mono-tube hydraulic dampers was developed and validated with experimental data. Experimental data and mathematical model results were compared for various different situations and good agreement was observed, but only one typical comparison was shown in Figure 7-10. The experimental data and model results satisfactorily match for other situations.

Effects of shim thickness, shim stack pre-deflection, and shims' stiffness were studied. Some other parameters such as bypass valve opening, oil's bulk modulus, and oil density were also investigated. Similarly, other damper parameters can be investigated and their effects can be studied. A GUI code was built using MATLAB enhancing the ease of use of the code.

Mathematical model results show that increasing the thickness of the shims in the shim stack will increase the stiffness of the damper and increase the slope of the force-velocity curve. The Young's modulus of shims and oil density will also affect the slope of force-velocity curve in a similar fashion. Pre-deflection

of the shim stack will increase the damper force. However, it has no effect on the slope of the force-velocity curve. Pre-deflection of the shim stack will also increase the size of the hysteretic region at low velocities if the bypass is closed. It is found that the hysteretic region is greatly affected by the bypass valve opening, pre-deflection of the shim stack, and the oil bulk modulus but, it is not affected by oil density.

Chapter 8
Effects of Suspension Components Parameters on Vehicle Dynamic Behavior

Full understanding of the effects of damper components, discussed in the previous chapters, on suspension system behavior is better understood if the damper model is integrated into a suspension model. In this chapter, the damper model is inserted into a quarter car model and simulation results are presented. The purpose of this chapter is to analyze the effects of damper parameters (e.g. shim stack, hydraulic oil, by-pass orifices, etc.) on the overall dynamics of the vehicle.

This chapter will present the details of a theoretical study of a quarter-car suspension that includes the mathematical models developed in the previous chapters. The overall model that will be developed in this chapter can be used as a tool for suspension design and tuning. For example, effects of various shim stack assemblies with different configurations can be studied using this model and the behavior of the overall suspension system can be analyzed before the actual tuning process takes place. This will save considerable time and reduces the cost in suspension design and tuning.

The model developed in this chapter has not been validated using experimental test data on a quarter car rig. However, because individual components of the system, such as the hydraulic damper model, are validated (Chapter 7), it is expected that the simulation results of the quarter car model are close to reality. Experimental validation of this quarter car model can be the focus of a future project. The ultimate goal of

such study could be the investigation and tuning of suspension systems such as the one shown in Figure 8-1.

Figure 8-1 The 8-post shaker rig and the VT cupcar used to study suspension system performance at Virginia Tech. (CVeSS/VIPER test facilities)

8.1. Model Assumptions and Problem Statement

Figure 8-2 shows the schematics of the quarter car to be modeled. The tire stiffness is assumed to be constant K_t , and the suspension stiffness is shown by K_s. The hydraulic damper model discussed in Chapter 7 is inserted into the quarter car model with all its features except for the dynamics of the floating piston. The dynamics of the damper floating piston will be removed from the quarter car model because it is assumed that the floating piston will have small effects on the large scale dynamic behavior of the vehicle. The floating piston dynamics can be added to the quarter car model easily but simulation time will increase due to addition of two differential equations to the system.

X_r is the road input displacement, X_t is the tire displacement, and X_b is the sprung mass displacement.

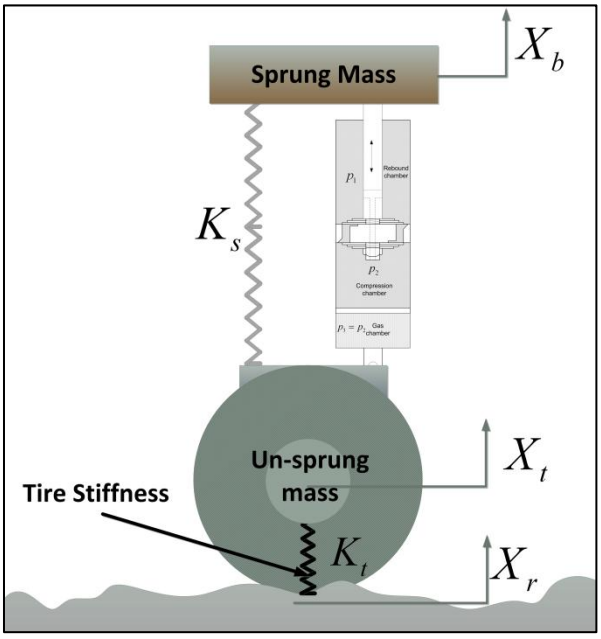

Figure 8-2 The hydraulic damper model of Chapter 7 is integrated in a quarter car model

8.2. System of Equations

When the dynamics of the damper floating piston are neglected, the gas pressure in the gas chamber is equal to the hydraulic oil pressure in chamber 2.

$$p_2 = p_3 \tag{8-1}$$

The intermediate pressure (p_5) and the flow between the chambers (Q_2) from Chapter 7 are rewritten as,

$$p_5 = \frac{C_{d_2}{}^2 A_2{}^2 p_1 + p_2 C_{d_3}{}^2 A_3{}^2}{C_{d_3}{}^2 A_3{}^2 + C_{d_2} A_2{}^2} \tag{8-2}$$

$$Q_2 = -C_{d_2} A_2 \operatorname{sgn}(-p_2 + p_1)\sqrt{\frac{2}{\rho}\left|\frac{C_{d_3}{}^2 A_3{}^2 (-p_2 + p_1)}{C_{d_3}{}^2 A_3{}^2 + C_{d_2}{}^2 A_2{}^2}\right|} - C_{d_4} A_4 \operatorname{sgn}(-p_2 + p_1)\sqrt{\frac{2}{\rho}|-p_2 + p_1|} \tag{8-3}$$

In contrast to the analysis in Chapter 7, in this chapter, both damper ends can move. The lower damper end is attached to the un-sprung mass (X_t) and the upper end is attached to the sprung mass (X_b), as shown in Figure 8-3.

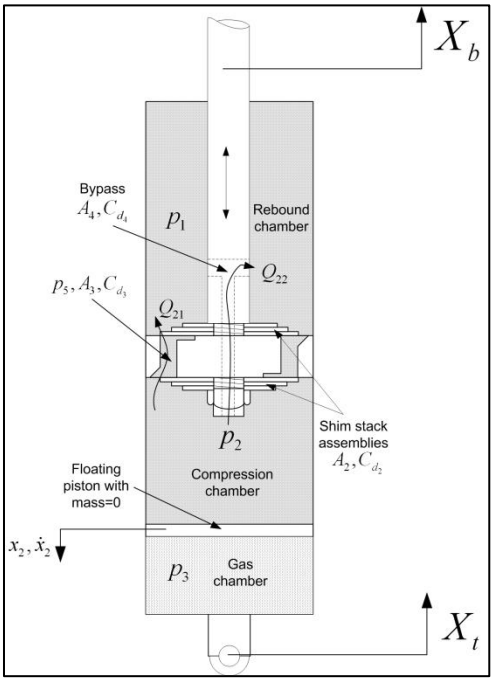

Figure 8-3 The hydraulics damper used in the quarter car model. The inertia of the floating piston is neglected.

The continuity equations for chambers 1, 2, and 3 can be written as

$$Q_2 = \left(A_p - A_r\right)\frac{d\left(X_t - X_b\right)}{dt} + \frac{1}{\beta}\left[\left(A_p - A_r\right)\left(X_t - X_b\right) + V_{10}\right]\frac{dp_1}{dt} \tag{8-4}$$

$$-Q_2 = A_p\left[\frac{dX_b}{dt} + \frac{dx_2}{dt}\right] + \frac{1}{\beta}\left(V_{20} + A_p X_b + V_{30} - V_3\right)\frac{dp_2}{dt} \tag{8-5}$$

$$\frac{dx_2}{dt} + \frac{dX_t}{dt} = \frac{-1}{A_p}\frac{dV_3}{dt} \tag{8-6}$$

Replacing equation (8-3) in (8-4) and solving for the rate of change of pressure in chamber 1 results,

$$\frac{dp_1}{dt} = -\frac{\beta}{A_p X_t - A_p X_b - A_r X_t + A_r X_b + V_{10}} \times \left(C_{d_2} A_2 \operatorname{sgn}\left(-p_2 + p_1\right)\sqrt{\frac{2}{\rho}\left|\frac{{C_{d_3}}^2 {A_3}^2\left(p_2 - p_1\right)}{{C_{d_3}}^2 {A_3}^2 + {C_{d_2}}^2 {A_2}^2}\right|} + \ldots \right.$$
$$\left. \ldots C_{d_4} A_4 \operatorname{sgn}\left(-p_2 + p_1\right)\sqrt{\frac{2}{\rho}\left|-p_2 + p_1\right|} + A_p\frac{dX_t}{dt} - A_p\frac{dX_b}{dt} - A_r\frac{dX_t}{dt} + A_r\frac{dX_b}{dt} \right) \tag{8-7}$$

The gas in chamber 3 is assumed to undergo an adiabatic process, similar to the assumption made in Chapter 7.

$$p_{30}{V_{30}}^{n} = p_3 V_3^{n} \tag{8-8}$$

Combining equations (8-1), (8-6), and (8-8), floating piston velocity can be obtained as,

$$\frac{dx_2}{dt} = \frac{1}{nA_p} V_{30}\, p_{30}^{\frac{1}{n}}\, p_2^{-\left(\frac{1}{n}+1\right)} \frac{dp_2}{dt} - \frac{dX_t}{dt} \tag{8-9}$$

Using equations (8-3), (8-5), (8-8), and (8-9), the rate of change of pressure in chamber 2 will be obtained.

$$\frac{dp_2}{dt} = \frac{-n\beta\left(-C_{d_2}A_2\,\mathrm{sgn}\left(-p_2+p_1\right)\sqrt{\frac{2}{\rho}\left|\frac{{C_{d_3}}^2{A_3}^2\left(-p_2+p_1\right)}{{C_{d_3}}^2{A_3}^2+{C_{d_2}}^2{A_2}^2}\right|} - C_{d_4}A_4\,\mathrm{sgn}\left(-p_2+p_1\right)\sqrt{\frac{2}{\rho}\left|-p_2+p_1\right|} + A_p\frac{dX_b}{dt}\right)}{\left(\beta V_{30}\, p_{30}^{\frac{1}{n}}\, p_2^{-\frac{1+n}{n}} + nV_{20} + nA_pX_b + nV_{30} - n\left(\frac{p_{30}{V_{30}}^{n}}{p_2}\right)^{\frac{1}{n}}\right)} \tag{8-10}$$

Next, the equations of motion for the two degree of freedom system of the quarter car rig should be written. Using the parameters of Figure 8-4, equations of motion are obtained.

$$-f_d - k_s\left(X_b - X_t\right) - M_b g = M_b \ddot{X}_b \tag{8-11}$$

$$f_d + k_s\left(X_b - X_t\right) - k_t\left(X_t - X_r\right) - M_t g = M_t \ddot{X}_t \tag{8-12}$$

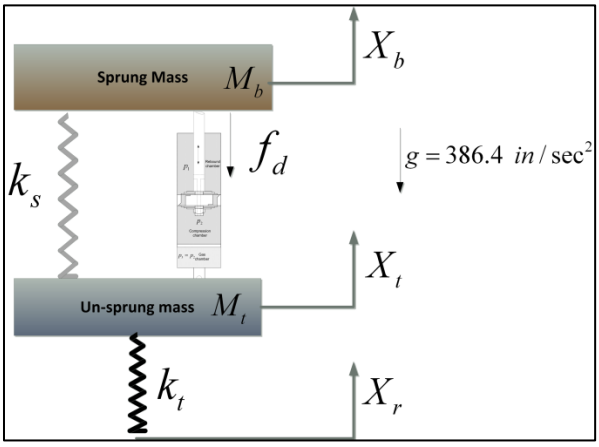

Figure 8-4 Physical model of the quarter car system

The damper force in equations (8-11) and (8-12) is considered to be positive when the damper is in extension and negative when it is being compressed. It should be mentioned that because the units of length in this work is inches, the units that should be used for masses M_t and M_b are slug/12. Also, the displacements of Figure 8-4 are measured from the uncompressed configuration of the system and not the static equilibrium point. Therefore, weights of the sprung and un-sprung masses are included in equations of motion.

To transform the equations into state-space form, the following state-space variables are defined,

$$\begin{aligned} P_1 &= p_1 \\ P_2 &= p_2 \\ P_3 &= X_b \\ P_4 &= V_b = \frac{dX_b}{dt} \\ P_5 &= X_t \\ P_6 &= V_t = \frac{dX_t}{dt} \end{aligned} \tag{8-13}$$

Using the state variables defined in (8-13), equations (8-7), (8-10), (8-11), and (8-12) can be written in state-space form.

$$\frac{dP_1}{dt} = -\frac{\beta}{A_p P_5 - A_p P_3 - A_r P_5 + A_r P_3 + V_{10}} \times \left(\begin{array}{l} C_{d_2} A_2 \operatorname{sgn}\left(-P_2 + P_1\right) \sqrt{\frac{2}{\rho}\left|\frac{{C_{d_3}}^2 {A_3}^2 \left(P_2 - P_1\right)}{{C_{d_3}}^2 {A_3}^2 + {C_{d_2}}^2 {A_2}^2}\right|} + \ldots \\ \ldots C_{d_4} A_4 \operatorname{sgn}\left(-P_2 + P_1\right) \sqrt{\frac{2}{\rho}\left|-P_2 + P_1\right|} + A_p P_6 - A_p P_4 - A_r P_6 + A_r P_4 \end{array} \right) \quad (8\text{-}14)$$

$$\frac{dP_2}{dt} = \frac{-n\beta\left(-C_{d_2} A_2 \operatorname{sgn}\left(-P_2 + P_1\right) \sqrt{\frac{2}{\rho}\left|\frac{{C_{d_3}}^2 {A_3}^2 \left(-P_2 + P_1\right)}{{C_{d_3}}^2 {A_3}^2 + {C_{d_2}}^2 {A_2}^2}\right|} - C_{d_4} A_4 \operatorname{sgn}\left(-P_2 + P_1\right) \sqrt{\frac{2}{\rho}\left|-P_2 + P_1\right|} + A_p P_4 \right)}{\left(\beta V_{30} {p_{30}}^{\frac{1}{n}} {P_2}^{-\frac{1+n}{n}} + nV_{20} + nA_p X_b + nV_{30} - n\left(\frac{{p_{30}}{V_{30}}^n}{P_2}\right)^{\frac{1}{n}} \right)} \quad (8\text{-}15)$$

$$\frac{dP_3}{dt} = P_4 \quad (8\text{-}16)$$

$$\frac{dP_4}{dt} = \frac{1}{M_b}\left(-f_d - k_s P_3 + k_s P_5 - M_b g\right) \quad (8\text{-}17)$$

$$\frac{dP_5}{dt} = P_6 \quad (8\text{-}18)$$

$$\frac{dP_6}{dt} = \frac{1}{M_t}\left(f_d + k_s P_3 - \left(k_s + k_t\right) P_5 + k_t X_r - M_t g\right) \quad (8\text{-}19)$$

In the state-spate equations (8-14)-(8-19), the damper force is

$$f_d = P_1\left(A_p - A_r\right) - P_2 A_p \quad (8\text{-}20)$$

where, the internal damper friction forces are ignored because of their small contribution to dynamic forces encountered in the quarter car model.

The state-space equations of (8-14)-(8-19) can now be solved with a given initial condition for the six states. There are, however, some kinematic constraints that should be satisfied. These kinematic constraints are tire-road contact loss, maximum damper compression, and maximum damper extension.

8.3. Kinematic Constraints

Three major kinematic constraints are implemented into the quarter car system model to represent the real scenarios that could happen during real operation of the vehicle. These scenarios are:

1. The tire cannot undergo tensile forces. Instead, it loses contact with the road.
2. When the suspension system reaches its maximum compression in extreme conditions, rubber mounts prevent more suspension deflection. These rubber mounts are called *"bump stops"*.
3. When the suspension system reaches its maximum extension in extreme conditions, the *"Extension bump stop"* prevents more suspension deflection.

8.3.1 Tire-Road Contact Loss

The tire loses contact with the road when the distance between the tire and the road becomes more than their initial distance. Therefore, the tire force becomes zero if tire loses contact with the road.

$$\text{if} \quad X_t - X_r > 0 \Rightarrow F_t = 0 \tag{8-21}$$

This constraint can be applied in computer codes by introducing a contact coefficient multiplied with the tire stiffness in the equations and setting it to zero when there is no contact. Its value is one if the tire is contacting the road.

After the tire loses contact with the road, only the gravitational force acts on the system until the tire contacts the road again.

8.3.2 Maximum Suspension Compression

The maximum damper compression is

$$\text{Max. Comp.} = s_{20} + s_{30} - th_{float} \tag{8-22}$$

where, s_{20} and s_{30} are the initial lengths of chambers 2 and 3, respectively. th_{float} is the thickness of the floating piston.

When the length of chamber 3 (gas chamber) reaches zero, the damper locks up because there is no volume of the gas chamber to compensate for the extra volume due to the insertion of the piston rod. Hence, when damper is compressed close to this limit, bump stops hit each other preventing the suspension to undergo more compression. This behavior can be considered as a semi-elastic impact (with a coefficient of restitution C_R) between the sprung and un-sprung masses.

Mathematically, the maximum suspension compression constraint can be written as

$$\begin{aligned} &\text{if } \left(s_3 \le s_{3\min}\right) \text{ and } \left(V_t > V_b\right), \text{ then:} \\ &V_{t_f} = \frac{M_t V_{t_i} + M_b V_{b_i} + M_b C_R \left(V_{b_i} - V_{t_i}\right)}{M_t + M_b} \\ &V_{b_f} = \frac{M_t V_{t_i} + M_b V_{b_i} + M_t C_R \left(V_{t_i} - V_{b_i}\right)}{M_t + M_b} \end{aligned} \tag{8-23}$$

where, subscript *i* denotes the quantities before impact, subscript *f* denoted the quantities after impact, and $s_{3\min}$ is the minimum allowable length of chamber 3. C_R is the coefficient of restitution. A value of one for the coefficient of restitution shows a 100% elastic impact and a value of zero indicates a 100% plastic impact.

8.3.3 Maximum Suspension Extension

When damper runs out of extension stroke, an impact occurs between sprung and un-sprung masses, similar to the previous case with the maximum compression.

This condition can be mathematically represented as

$$\text{if } \left(X_b - X_t > s_{10} - s_{1\min}\right) \text{ and } \left(V_b > V_t\right), \text{ then:}$$

$$V_{t_f} = \frac{M_t V_{t_i} + M_b V_{b_i} + M_b C_R \left(V_{b_i} - V_{t_i}\right)}{M_t + M_b} \tag{8-24}$$

$$V_{b_f} = \frac{M_t V_{t_i} + M_b V_{b_i} + M_t C_R \left(V_{t_i} - V_{b_i}\right)}{M_t + M_b}$$

where, $s_{1\min}$ is the minimum allowable length of chamber 1.

8.3.4 Maximum Shim Stack Tip Deflection

In hydraulic dampers, shim stack assembly cannot deflect more than an allowable value. This is due to the existence of a rigid stop that prevent the shim stack from large deflections and hence, not allowing for plastic deformation of disks. This kinematic constraint is implemented in the quarter car model.

8.4. The Computer Program

In order to solve the system of equations and show the desired simulation results, a MATLAB code is written. This code includes all the features mentioned in previous chapters with respect to the shim stack deflection analysis and hydraulic damper model. A GUI layout, shown in Figure 8-5, makes the use of the code very easy. The code is also able to generate animation inputs of the simulation results that can be animated using various software. Here, Simulink will be used for generating animations.

The computer code can generate the following outputs for study of suspension performance:

- Damper force curves (force-velocity and force-displacement)
- Shim stack tip deflection
- Sprung mass time history
- Un-sprung mass time history
- Suspension compression time history
- Tire force time history

- Phase-space diagrams
- Poincare maps
- FFT of state variables
- Generating data for creating animations (using Simulink)

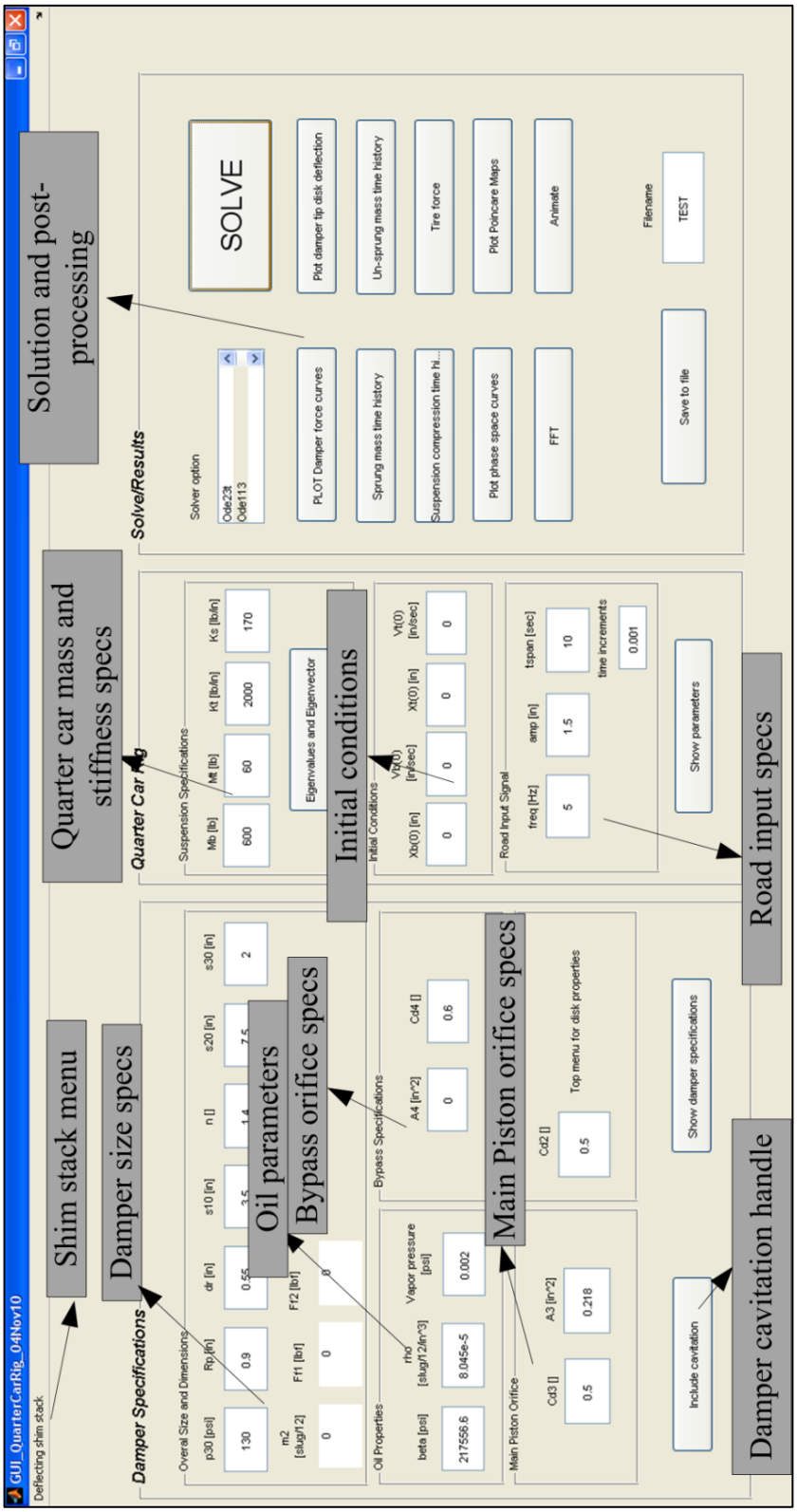

Figure 8-5 MATLAB GUI code for the quarter car model

8.5. Simulation Results

In this section, simulation results of a study performed on a hydraulic damper similar to what discussed in previous chapter is presented. The dimensions of the damper are similar to the previous chapter and correspond to dimensions of an OHLINS CCJ 23/8 damper. Other dimensions and specifications are based on configurations encountered in real situations.

8.5.1 The Base Example

Numeric values used as the base example are shown in Tables 1-4 and Figures 8-6 through 8-8. The code is run for this base example and effects of various suspension components on the response of the system are studied. The shim stack properties are the main focus in this study.

Table 8-1 Numeric values used in the base example – Shim stack assembly

Parameter symbol	Parameter name	Parameter value	Parameter symbol	Parameter name	Parameter value
a	Inner radius of the disk assembly	0.353 in.	th_1	Thickness of the largest disk	0.012 in.
r_1	Radius of the largest disk	0.675 in.	th_2	Thickness of the second disk	0.012 in.
r_2	Radius of the second disk	0.600 in.	th_3	Thickness of the smallest disk	0.012 in.
r_3	Radius of the smallest disk	0.525 in.	pre_def	Pre-deflection	0.005 in.
l	Location on the disk assembly where pressure acts	0.45 in.	C_{d2}	Discharge coefficient for the shim stack	0.5
E	Young's modulus	2.9e7 Psi	k	Shear correction factor	0.833
ν	Poison's ratio	0.3			

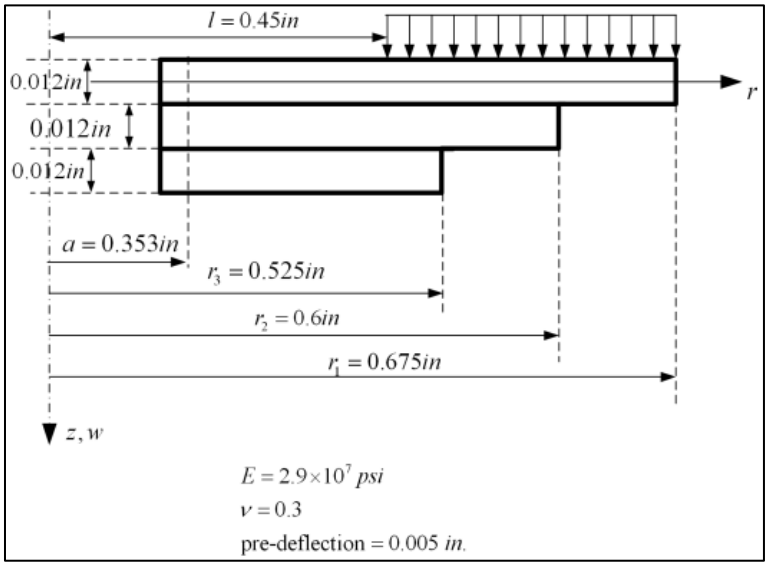

Figure 8-6 Numeric vlaues for the shim stack parameters

Table 8-2 Numeric values used in the base example – Shim stack assembly

Parameter symbol	Parameter name	Parameter value	Parameter symbol	Parameter name	Parameter value
p_{30}	Initial pressure in gas chamber	130 Psi	β	Oil bulk modulus	217557 Psi
R_p	Piston radius	0.9 in.	ρ	Oil density	8.045e-5 slug/12/in^3
d_r	Rod diameter	0.55 in.	-	Oil vapor pressure	0.002 Psi
s_{10}	Initial length of rebound chamber	3.5 in.	C_{d3}	Main piston discharge coefficient	0.5
s_{20}	Initial length of compression chamber	7.5 in.	A_3	Main piston orifice area	0.218 in.^{2}
s_{30}	Initial length of gas chamber	2 in.	C_{d4}	By pass orifice discharge coefficient	0.6
n	Gas constant	1.4	A_4	Bypass orifice area	0 in.^{2}

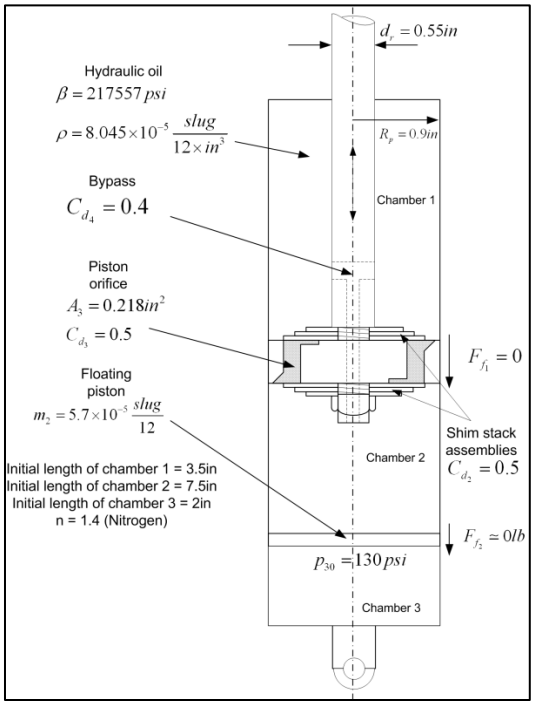

Figure 8-7 Numeric values for damper parameters

Table 8-3 Numeric values used in the base example – Quarter car model

Parameter symbol	Parameter name	Parameter value	Parameter symbol	Parameter name	Parameter value
M_b	Sprung mass	600 lb	V_{b0}	Initial velocity of sprung mass	0 in./sec
M_t	Un-sprung mass	60 lb	X_{t0}	Initial displacement of un-sprung mass	0 in.
K_t	Tire stiffness	2000 lb/in.	V_{t0}	Initial velocity of un-sprung mass	0 in./sec
K_s	Suspension stiffness	170 lb/in.	*freq*	Road input frequency (sine waveform)	5 Hz
X_{b0}	Initial displacement of sprung mass	0 in.	*amp*	Road input amplitude (sine waveform)	0.5 in.

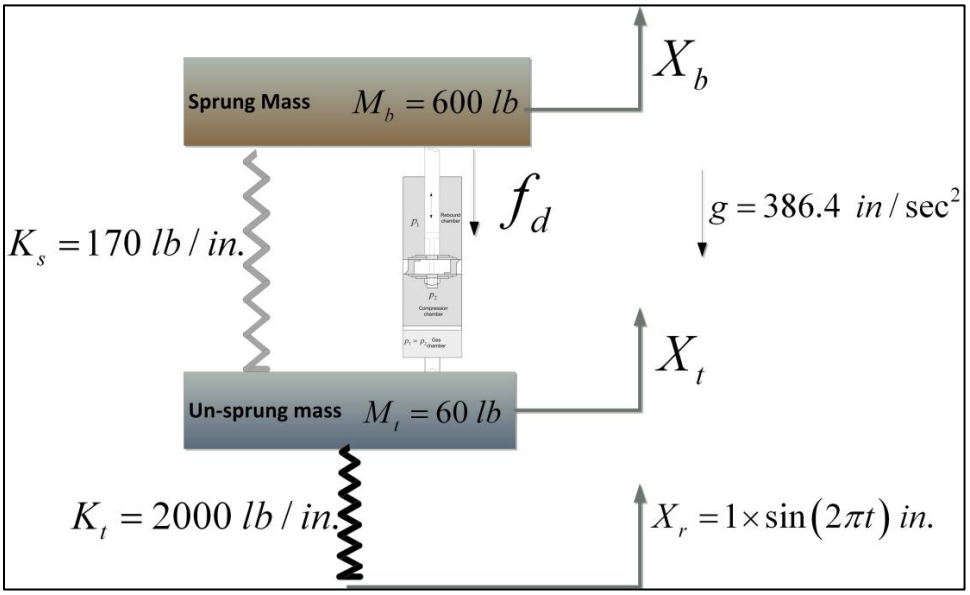

Figure 8-8 Numeric values for the quarter car system

Table 8-4 Numeric values used in the base example – Solver options and control parameters

Parameter symbol	Parameter name	Parameter value	Parameter symbol	Parameter name	Parameter value
-	Solver choice	ode23t/ode113	-	Maximum shim stack tip deflection	0.2 in.
InitialStep	Initial time step for ode solver	5e-9 sec	-	Minimum allowed length of rebound chamber	0.5 in.
AbsTol	Absolute tolerance of the ode solver	1e-6	-	Minimum allowed length of gas chamber	0.24 in.
RelTol	Relative tolerance of the ode solver	1e-3	C_R	Coefficient of restitution	0.5
MaxStep	Maximum time step for ode solver	1e-2 sec			

Various MATLAB solvers can be used to solve the state-space equations. However, it was found that "ode23t" and "ode113" are the best choices in terms of speed and accuracy. The default solver is ode23t. This solver is the fastest solver and is able to handle the normal situations encountered in the problem.

ode113 is used when damper kinematic constraints occur. In this case, ode113 is capable of more efficiently solve the equations compared to ode23t. The code allows the user to conveniently choose between these two solvers.

8.5.2 Base Example Simulation Results

Using the numeric values mentioned in the previous section, the base example problem is solved and results are presented and discussed in this section. The base example is solved using ode23t and total run time to solve for 5 seconds of simulation time is about 30 seconds.

Figure 8-9 shows the shim stack tip deflection time history. Total simulation time is 5 sec but only 2 seconds are shown for more clarity of the curves. The system's response becomes steady after 1 sec. As can be seen in Figure 8-9, shim stack deflection depends on the pressure difference across the damper piston. Because there was a 0.005 in. of pre-deflection, at some points, the shim stack does not deflect until the pressure difference across the damper piston is large enough to overcome the pre-deflection. This is shown as small zero deflection regions and corresponding pressure spikes in Figure 8-9. Sprung and un-sprung mass responses are shown in Figure 8-10 and Figure 8-10 for the base example, respectively.

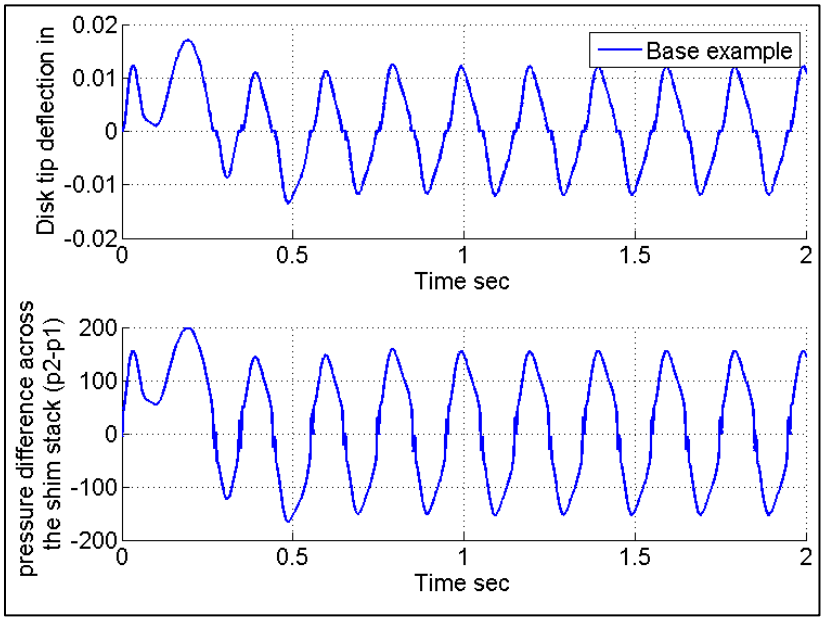

Figure 8-9 Shim stack tip deflection - Base example

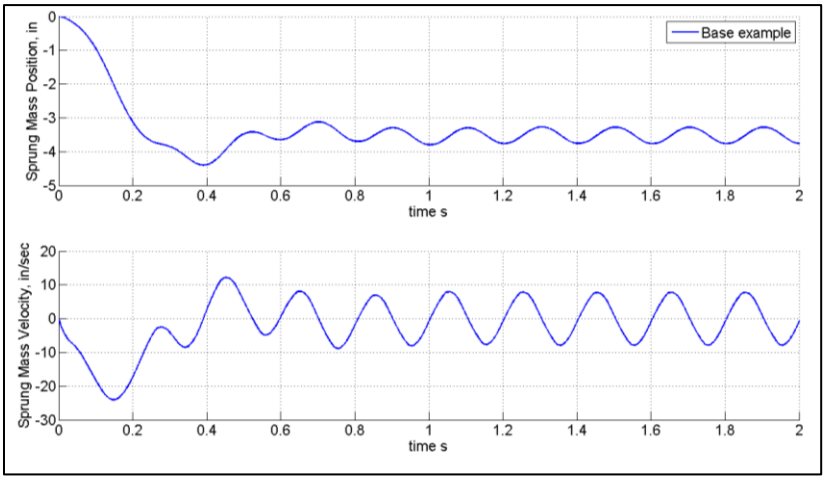

Figure 8-10 Sprung mass response - Base example

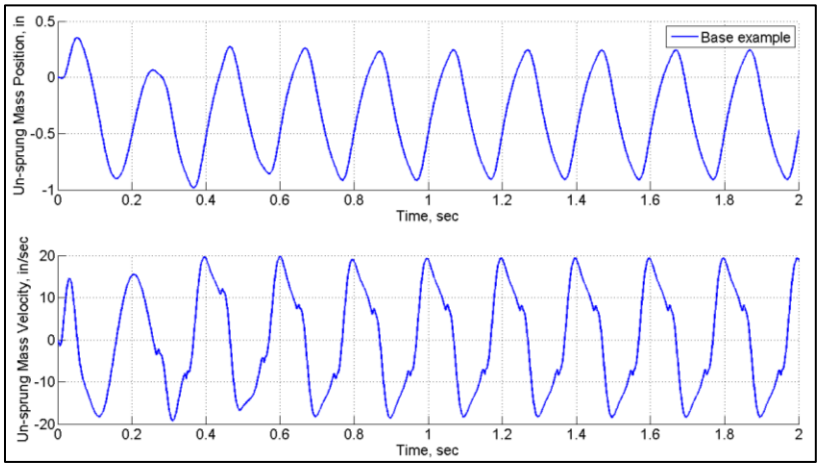

Figure 8-11 Un-sprung mass response - Base example

Figure 8-12 shows the suspension compression versus time. As can be seen, after about 1 sec, the response enters the steady-state phase. The suspension will oscillate around 3.5in. of compression. The tire force is shown in Figure 8-13.

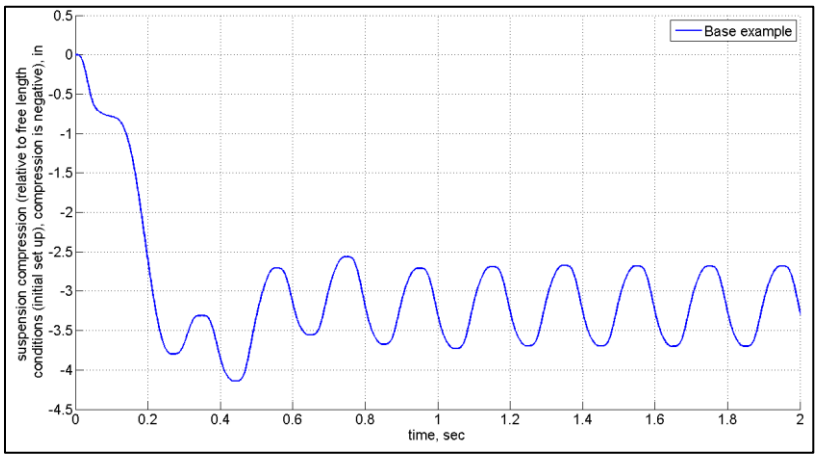

Figure 8-12 Suspension compression - Base example

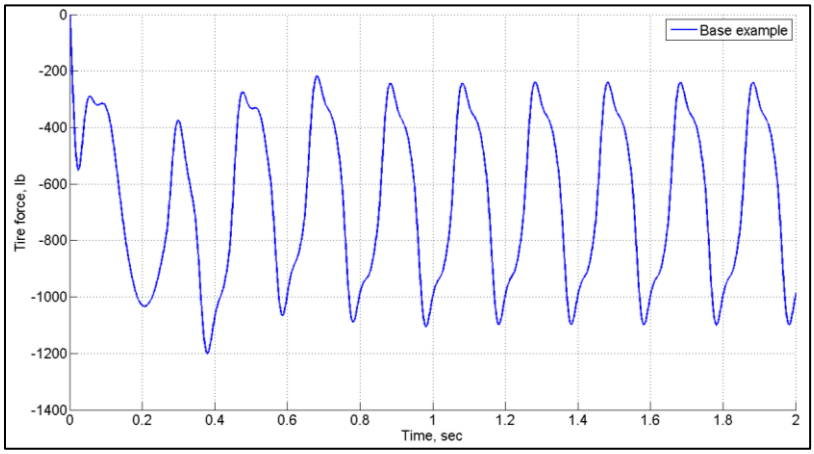

Figure 8-13 Tire force - Base example

Two phase space plots are shown in Figure 8-14. In order to improve the sprung mass response, peak-to-peak velocity and displacement of the sprung mass should decrease. Therefore, one should try to adjust the suspension components parameters such that the area of the phase space plots decrease with respect to the base example.

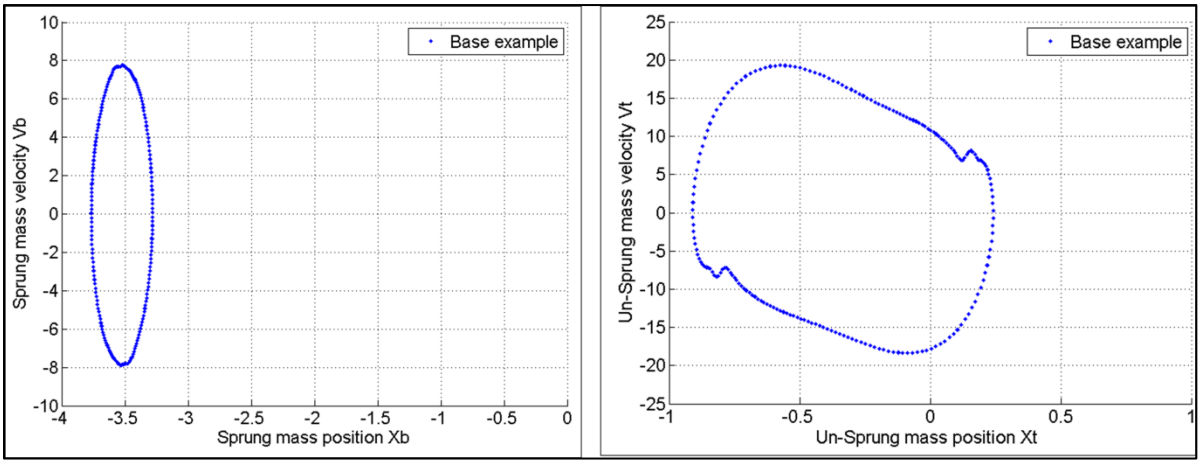

Figure 8-14 Phase space curves - Base example

8.5.3 Effects of Suspension Component Parameters on Dynamic Response of the System

As mentioned earlier, various parameters can be changed in the model and their effects can be studied on the overall system response. This is very useful for design and tuning of suspensions. In this section and the next sections, various parameters are changes and their effects are studied. These parameters are,

- Shim thickness
- Initial gas chamber pressure
- Shim stack pre-deflection
- By-pass orifice opening

In each case, only one parameter is changed with respect to the base example and effects are compared against the base example presented in the previous section.

To study the effect of shim thickness, let's assume the thickness of shims is changed to 0.01 in. each (Case 1). It is expected that the damper will be softer and produces less force compared to the case where 0.012 in. shims were used. Figure 8-15 shows the shim stack tip deflection for a shim stack consisting of 0.01 in. thick shims. As shown, because the shim stack is softer compared to the base example, it deflects more which results in less pressure drop across the damper main piston.

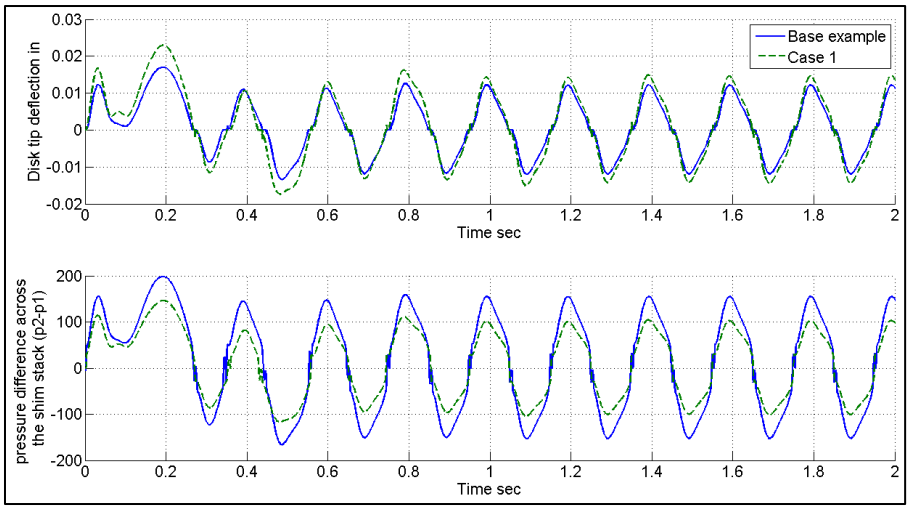

Figure 8-15 Shim stack tip deflection - Case 1

The sprung mass response is shown in Figure 8-16. It can be concluded that changing the shim thickness to 0.01 in. reduces the sprung mass peak-to-peak displacement and velocity. This is more obvious in the phase-space plots shown in Figure 8-17. As is seen, the phase-space curves for Case 1 lay inside the

curves for the base example. This shows that, Case 1 has smaller sprung and un-sprung mass displacements and velocities after steady state conditions are reached.

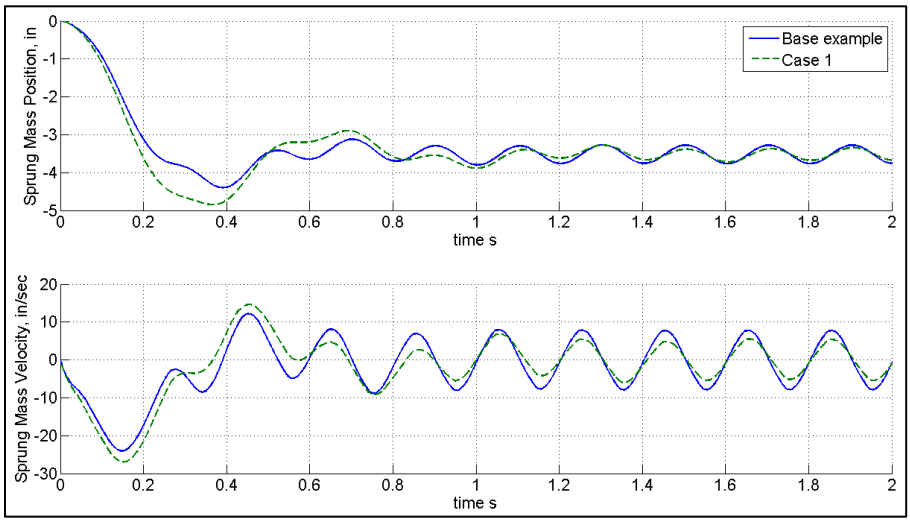

Figure 8-16 Sprung mass response - Case 1

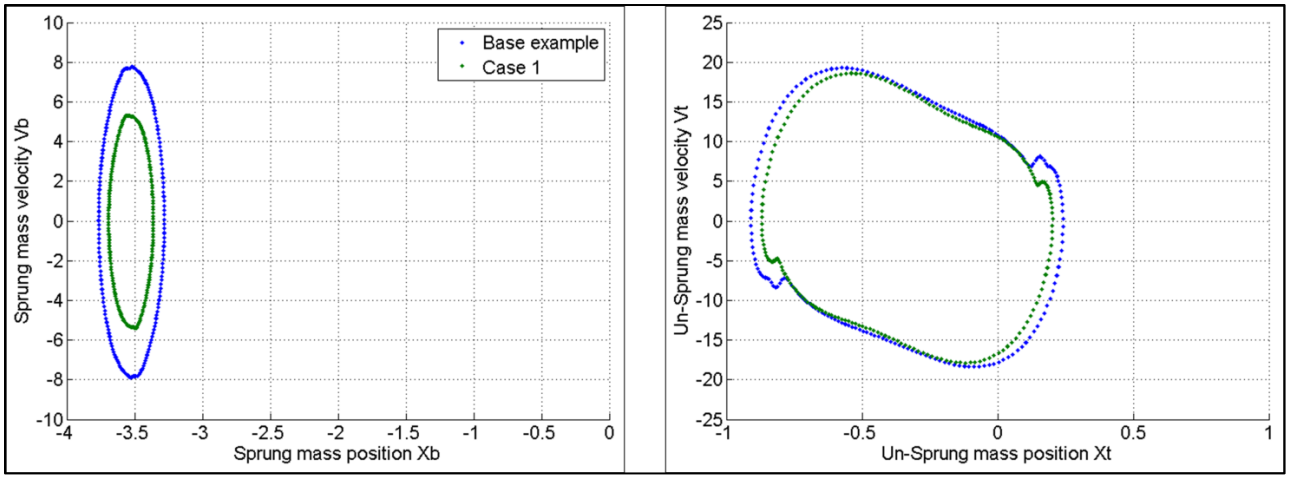

Figure 8-17 Phase space curves - Case 1

A comparison between tire forces is shown in Figure 8-18 for base example and Case 1. It can be concluded that decreasing shim thickness to 0.01 in. can result in less tire forces for a sine wave form input with a frequency of 5 Hz and amplitude of 0.5 in.

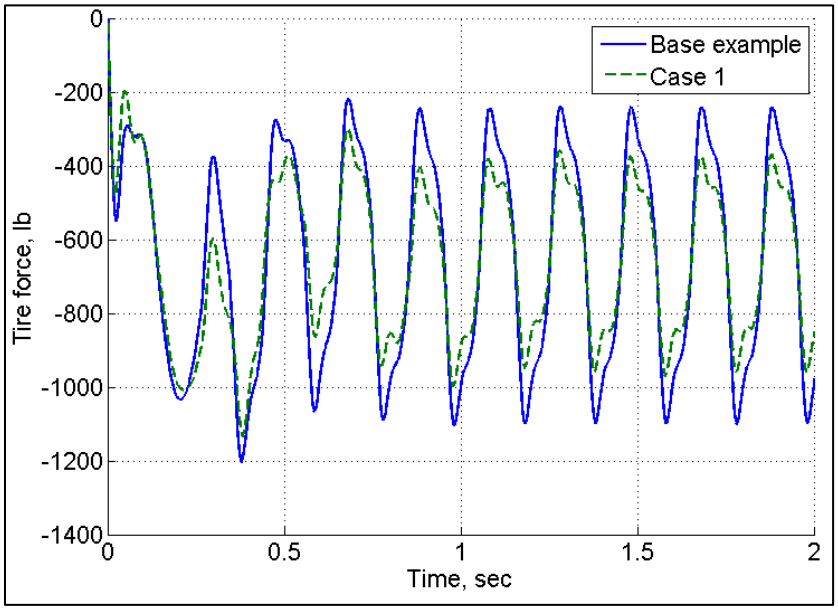

Figure 8-18 Tire force - Case 1

Similar to the analysis performed for shim thickness effects, other parameters can be studied and their effects can be investigated. Table 8-5 shows various cases considered here. It should be mentioned that this is only an example in which few parameters are considered. Many other parameters can be studied using this model.

Table 8-5 Parameters studied in this section and their numeric values

	Parameter studied	Original parameter value in the base example	Current value
Case 1	Shim thickness	0.012 in.	0.010 in.
Case 2	Pre-deflection	0.005 in.	0 in.
Case 3	Initial gas pressure	130 Psi	250 Psi
Case 4	By-pass orifice opening	0 $in.^2$	0.02 $in.^2$

Figure 8-19 shows a comparison between shim stack tip deflections for various cases considered in Table 8-5. It can be seen that opening the by-pass orifice will cause the shims to deflect less because a part of flow between damper oil chambers occurs through the by-pass orifice.

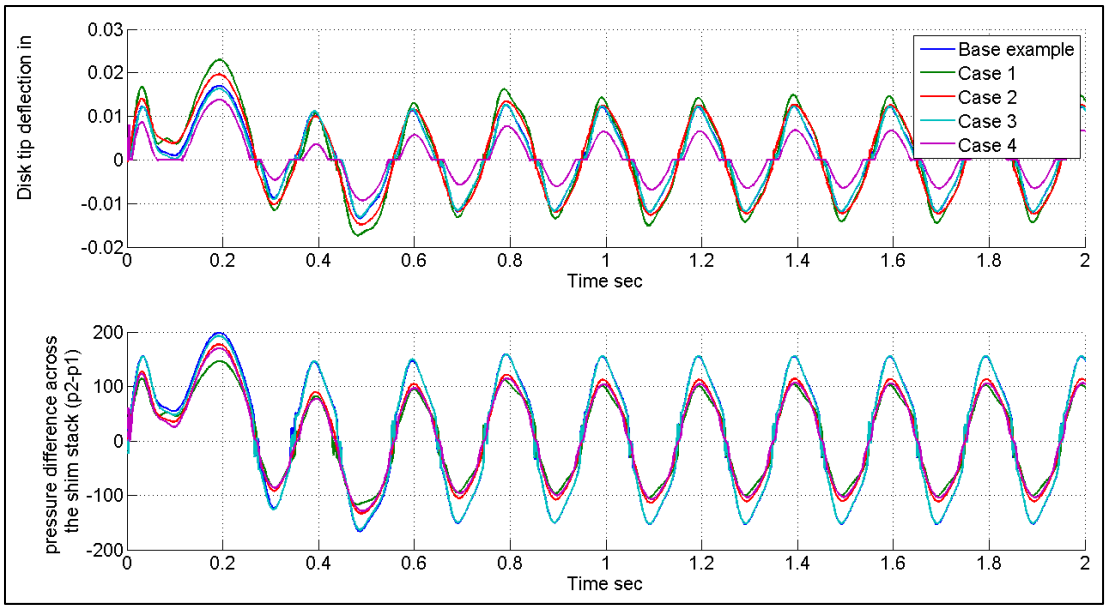

Figure 8-19 Shim stack tip deflection – Comparison

Time histories of the sprung and un-sprung masses are shown in Figure 8-20 and Figure 8-21. When the initial gas pressure is increased (Case 3), the sprung mass will oscillate around a new equilibrium point, as shown in Figure 8-20. This can also be seen in phase-space plots shown in Figure 8-22.

The un-sprung mass response is almost the same for all cases, as shown in Figure 8-21.

As mentioned, if it is desired to decrease the peak-to-peak displacement and velocity of the sprung mass, Case 1 and Case 4 are the best choices because they result in the smallest phase-space curve, as shown in upper left of Figure 8-22.

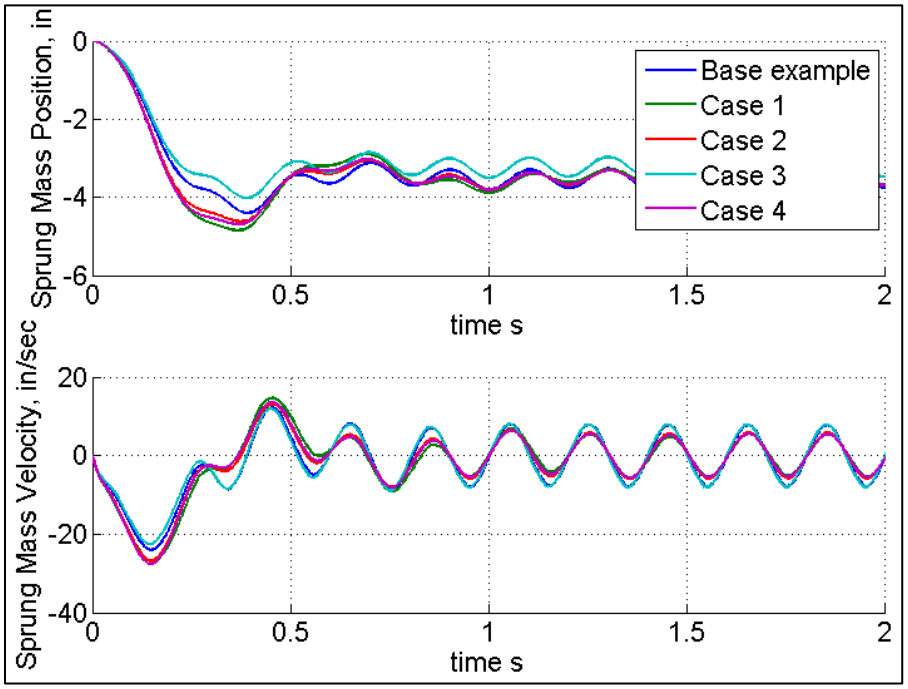

Figure 8-20 Sprung mass response – Comparison

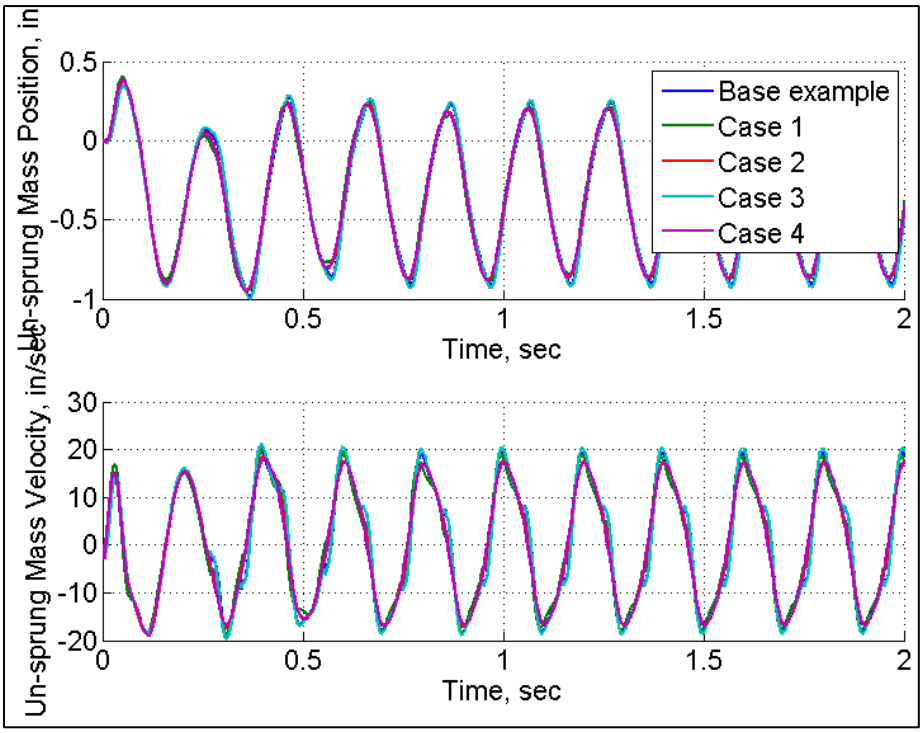

Figure 8-21 Un-sprung mass response – Comparison

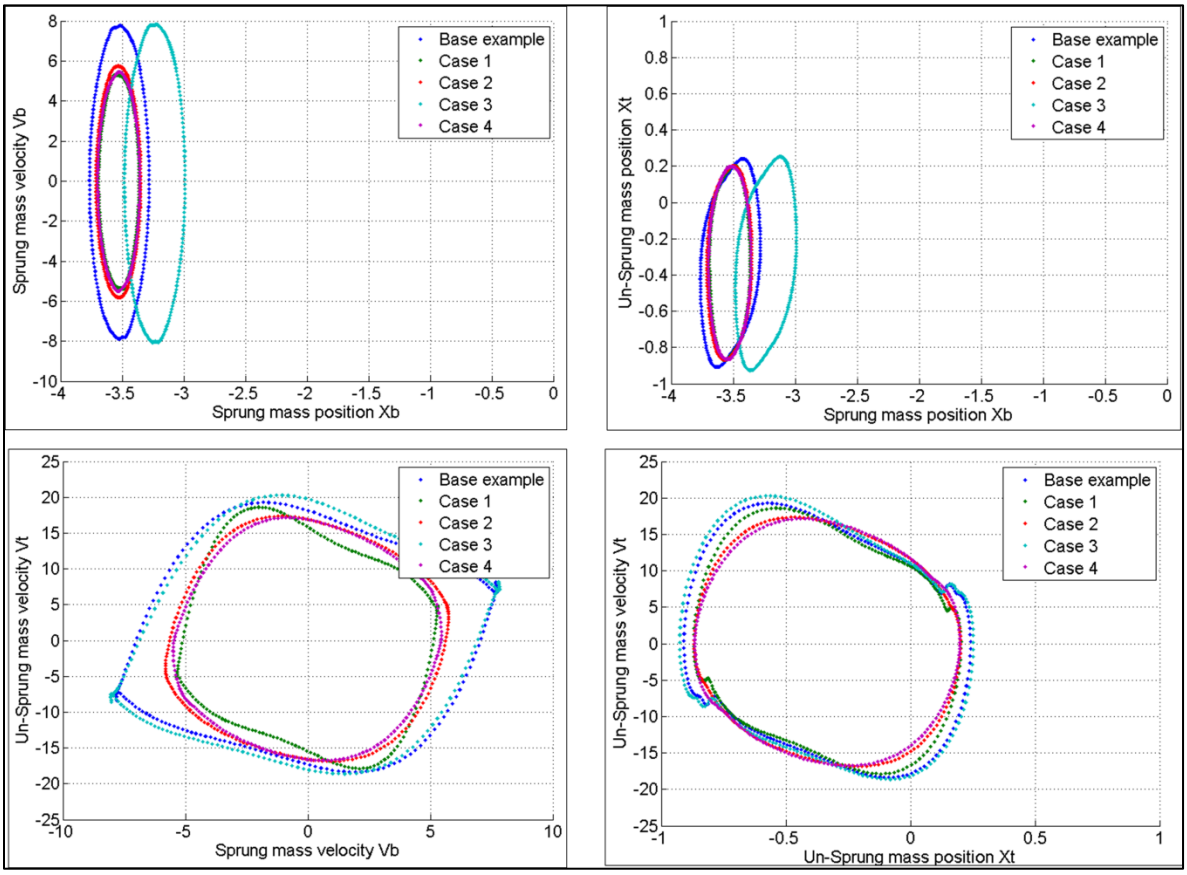

Figure 8-22 Phase-space curves – Comparison

Tire forces for different cases are compared in Figure 8-23. The base example and Case 3 result in highest tire forces due to thick shims and higher gas charge pressure, respectively. Tire force is reduced the most by decreasing shim thickness among the four cases considered.

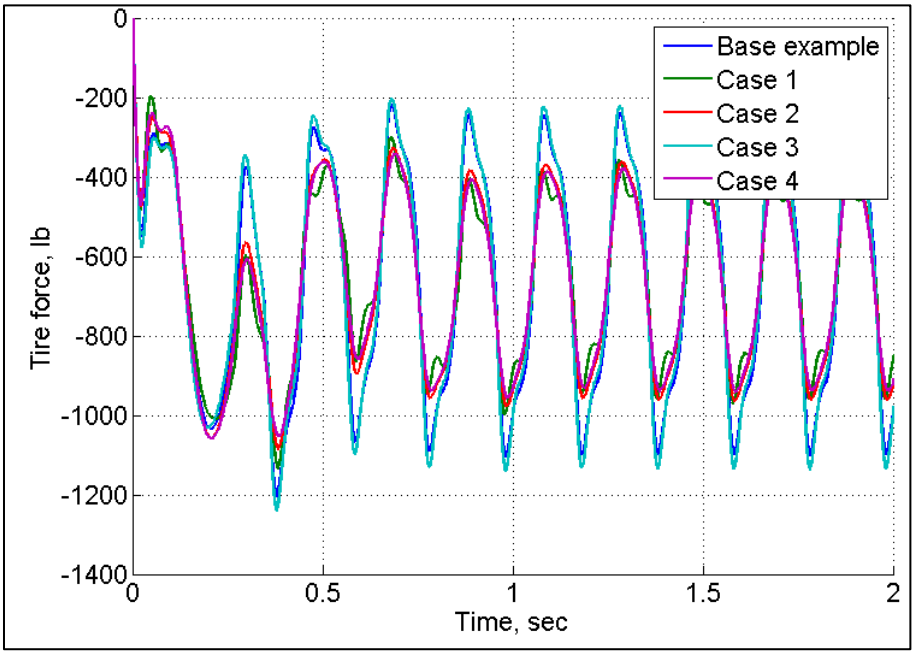

Figure 8-23 Tire force - Comparison

The example cases considered above demonstrate how various parameters can be studied and their effects on overall dynamic behavior of the system be investigated.

The road input is a sine waveform with adjustable frequency and amplitude. Other types of road input can also be used. This is proposed as a future extension of this work to expand the capabilities of the presented model.

Chapter 9
Summary and Conclusion

The research work presented considered the problem of vibration control of vehicular and structural systems. Advanced modeling techniques validated with experimental test data were developed to help with understanding the fundamentals as well as novel applications of smart suspensions.

Magneto-Rheological (MR) fluids are being used in many commercially available smart devices for vibration control of various systems. They are attractive to these applications because of their unique rheological behavior. They can change from a free flowing fluid to a solid like material in milli-seconds when a magnetic field is applied. This property can be used in many applications. MR dampers and MR rotary brakes have taken advantage of this property to be among the most widely used MR controllable devices for vibration control of various systems. Although these devices have proven to be very effective in vibration mitigation of many small-scale systems, there has not been much progress in expanding their application to larger systems. The reason is the high density and cost of MR fluids. Using MR fluids in large scale applications requires a large amount of MR fluid which makes the device heavy and expensive. In this research, novel and efficient ways of application of MR fluids in intelligent devices were investigated in order to extend the range of applications to larger scale systems. For this purpose, it was proposed that MR fluids be used in another flow mode not currently being used in any intelligent MR device. This flow mode is the squeeze mode which is the least understood flow mode of MR fluids.

In order to use MR fluids in the squeeze mode in intelligent MR devices, basic and fundamental behaviors of MR fluids in the squeeze mode needed to be fully understood. Therefore, attention was paid to rheological study of MR fluids in the squeeze mode. Both experimental and theoretical studies were

performed. A squeeze mode rheometer was designed and built to experimentally test MR fluids in the squeeze mode. The rheometer was used for testing MR fluids in the squeeze mode under various conditions. By extensive experimental testing, it was shown that MR fluids were capable of carrying a wide range of controllable force in the squeeze mode. This force range was achievable along a short stroke. All test results showed similar trends. The amount of the force achieved depended on the type of the MR fluid, magnetic field density, and gap size. This large range of controllable force along a short stroke can be of interest for many applications requiring a large controllable forces in envelopes that can only accommodate small strokes. Hence, MR fluids in the squeeze mode are good candidates for large scale and high force capacity intelligent devices.

Two possible applications for MR fluid squeeze flow were discussed experimentally and analytically. It was found that the squeeze mode can be successfully used in smart MR devices. *MR Hybrid Dampers* and *MR Squeeze Mounts* were two samples of this new class of intelligent devices. MR hybrid dampers were tested and test results showed that they can deliver controllable range of forces. Although MR hybrid dampers served as proof of concept prototypes for use of MR fluids in a new mode, sealing and frequency band width need to be improved before MR hybrid dampers can be confidently used in commercial large force capacity applications.

The MR squeeze mount test results showed that this device was capable of achieving very high controllable forces in small operational envelopes. Quasi-static tests and modeling were performed and promising results were obtained. MR squeeze mounts can be used as engine mounts and other vibration isolation devices in high load applications.

A clumping effect was also seen for the tested MR fluids and its severity depended on the magnetic field density, type of the MR fluid, and volumetric percentage of the MR fluid. The design of MR devices that work based on the squeeze mode requires a solution to reduce the clumping effect. Clumping causes the carrying liquid to separate from iron particles reducing test repeatability. The clumping behavior and

possible techniques to reduce clumping were studied and it was found that a magnetic dither can help with re-dispersing the iron particles.

The mathematical models presented in chapter 3 were able to illustrate the basic behavior of MR fluids in the squeeze mode and can be used for design and optimization purposes. Two approaches were taken to solve the squeeze problem of MR fluids. The first approach resulted in a simple mathematical model that can accurately predict the total squeezing force. This mathematical model was validated using experimental test data obtained from the squeeze mode rheometer. Although this model was capable of predicting the squeeze force accurately, in later stages of design of MR squeeze mode devices, some additional information is needed. In order to have a mathematical model capable of studying flow field, shear rate distribution, and pressure distribution, a second mathematical model was developed. This model was more mathematically involved than the simple model and may not be needed for preliminary design stages of MR devices. But it can be extremely helpful for optimization of existing designs or design of devices in which design decisions highly depend on shear rate and pressure distributions. Perturbation techniques were used in this model and as a result, the final solutions were obtained as closed-form equations which reduced the need for more complicated and computationally expensive numerical methods. Similar to the first model, the second model was also validated using experimental data.

The primary application in mind for smart devices developed in this research project was vehicle systems. Devices developed as proof of concepts will eventually be used as components in smart suspension systems. Therefore, it was essential to have a good design and synthesis tool for study of suspensions. The second part of this research work was devoted to development of advanced and novel design tools for vehicle suspension systems. Extensive theoretical and experimental studies were done to develop a reliable and robust design and synthesis tool for vehicle suspensions. This design and synthesis tool was

based on advanced physics-based mathematical models and is superior to any published mathematical model currently available in the literature.

Of particular interest was the deflection analysis of the shim stack assemblies in hydraulic dampers. A detailed mathematical model of a stack of annular disks was presented. The model was presented for a stack of three disks but it can be extended to include as many disks as desired. The First-Order Shear Deformation Theory (FSDT) was used. The model of the shim stack assembly was compared to the model of a single stepped disk with the same geometry. It was shown that the shim stack deflected considerably more than a stepped disk with the same geometry under similar loading and boundary conditions. The mathematical model was verified by examining some limit cases whose solutions are known analytically. Effects of the radial deflections were also studied using the Classical Plate Theory (CPT) and it was observed that the radial deflections did not have a considerable effect on the axial deflections. The analysis developed is expected to save designers considerable time and expense when considering hydraulic systems such as discussed.

The shim stack model was incorporated into a hydraulic damper model and a detailed mathematical model for mono-tube hydraulic dampers was developed and validated using experimental test data. Experimental data and mathematical model results were compared for different situations and good agreement was observed. Effects of shim thickness, shim stack pre-deflection, and shims' stiffness were studied. Some other parameters such as bypass valve opening, oil's bulk modulus, and oil density were also investigated. Similarly, other damper parameters can be investigated and their effects can be studied. A GUI code was built using MATLAB enhancing the ease of use of the code. Mathematical model results show that increasing the thickness of the shims in the shim stack will increase the stiffness of the damper and increase the slope of the force-velocity curve. The Young's modulus of shims and oil density will also affect the slope of force-velocity curve in similar fashion. Pre-deflection of the shim stack will increase the damper force. However, it has no effect on the slope of the force-velocity curve. Pre-

deflection of the shim stack will also increase the size of the hysteretic region at low velocities if the bypass is closed. It is found that the hysteretic region is greatly affected by the bypass valve opening, pre-deflection of the shim stack, and the oil bulk modulus, but it is not affected by oil density.

Full understanding of the effects of damper components, discussed in Chapters 6 and 7, on suspension system behavior is better understood if the damper model is integrated into a suspension model. In this work, the damper model was inserted into a quarter car suspension model and simulation results were presented. The purpose was to analyze the effects of damper parameters (e.g. shim stack, hydraulic oil, by-pass orifices, etc.) on the overall dynamics of the vehicle. Chapter 8 presented the details of a theoretical study of a quarter-car suspension that included the mathematical models developed in the previous chapters. The full suspension model was then used as a powerful design and synthesis tool for suspension design and tuning. Using this design tool, suspension engineers in manufacturing companies and other industrial sections can easily perform parametric studies without the need to carry out time consuming and expensive field and laboratory tests. For example, effects of various shim stack assemblies with different geometries can be studied using this model and the behavior of the overall suspension system can be analyzed before the actual tuning process takes place.

Appendices

Appendix A. The GUI Code for Analysis of MR Squeeze Mount Test Results

```
   % The GUI form for plotting the MR pouch test results   09-July-08
clear all
clc
prompt={'enter the name of the Data file'};
    dlg_title = 'Input dialog box';
    num_lines 6;
    def     = {'070908_01p'};
test_file= inputdlg(prompt,dlg_title,num_lines,def);
test_file=char(test_file);
nom_of_files=size(test_file,1);
col=struct('colors',{'blue','red','green','yellow','black','c','m'});
colstr={col.colors}
 %FILTER
    Ny=500; %rad/sec- Nyquist freq(=Fs/2)
    wn=10;  %cutoff freq of the filter
    ncf=wn/Ny ; %Normalized Cutoff Freq
    n=2 ;%order of the filter
    [num,den] = butter(n,ncf);
    G=tf(num,den);
s=struct('plots',{'Gap size vs time','Force vs time','Magnetic field vs time','Force vs gap
size','Magnetic field vs gap size','Overlaid plot of gap size,Magnetic field,and Force vs
time'});
    str={s.plots};
    [selection,ok] = listdlg('PromptString','Select a
plot','SelectionMode','multiple','ListString',str);
Dash=char(nom_of_files,1);
for i=1:nom_of_files
    Dash(i)='-';   %creating the - sign for legend in the graphs. the underline sign ( - ) does
not show properly in the legend
End
for NOF=1:nom_of_files
    load (num2str([test_file(NOF,1:length(test_file)),'.mat']))
    time=pf.X.Data;
    force=pf.Y(1).Data;
    Magnetic_field=pf.Y(2).Data;
    MicroProfiler=pf.Y(3).Data;
    displ=pf.Y(4).Data;
    displ_filtered=filtfilt(num,den,displ);
    force_filtered=filtfilt(num,den,force);
    Magnetic_field_filtered=filtfilt(num,den,Magnetic_field);
    if ok==1
        for i=1:length(selection)
            if selection(i)==1
                figure(11)
                hold on
                plot(time,displ_filtered,'Color',char(colstr(NOF))),grid on,xlabel('Time
(sec)'),ylabel('gap size (in)')
                legend([test_file(1:NOF,1:length(test_file)-4) Dash(1:NOF)
test_file(1:NOF,length(test_file)-3:length(test_file))])
            end
            if selection(i)==2
                figure(2)
                hold on
```

```
                plot(time,force_filtered,'Color',char(colstr(NOF))),grid on,xlabel('Time
(sec)'),ylabel('force (lb)')
                legend([test_file(1:NOF,1:length(test_file)-4) Dash(1:NOF)
test_file(1:NOF,length(test_file)-3:length(test_file))])
            end
            if selection(i)==3
                figure(3)
                hold on
                plot(time,Magnetic_field_filtered,'Color',char(colstr(NOF))),grid on,xlabel('Time
(sec)'),ylabel('Magnetic field strength in the probe gap (Tesla)')
                legend([test_file(1:NOF,1:length(test_file)-4) Dash(1:NOF)
test_file(1:NOF,length(test_file)-3:length(test_file))])
            end
            if selection(i)==4
                figure(4)
                hold on
                plot(displ_filtered,force_filtered,'Color',char(colstr(NOF))),grid on,xlabel('Gap
size (in)'),ylabel('Force (lb)')
                legend([test_file(1:NOF,1:length(test_file)-4) Dash(1:NOF)
test_file(1:NOF,length(test_file)-3:length(test_file))])
            end
            if selection(i)==5
                figure(5)
                hold on
                plot(displ_filtered,Magnetic_field_filtered,'Color',char(colstr(NOF))),grid
on,xlabel('Gap size (in)'),ylabel('Magnetic field (Tesla)')
                legend([test_file(1:NOF,1:length(test_file)-4) Dash(1:NOF)
test_file(1:NOF,length(test_file)-3:length(test_file))])
            end
            if selection(i)==6
                figure(6)
                hold on
                subplot(3,1,1)
                 plot(time,displ_filtered,'color',char(colstr(NOF))),grid on,xlabel('time
(sec)'),ylabel('Gap size (in)')
                 legend([test_file(1:NOF,1:length(test_file)-4) Dash(1:NOF)
test_file(1:NOF,length(test_file)-3:length(test_file))])
                 hold on
                 subplot(3,1,2)
                 plot(time,Magnetic_field_filtered,'color',char(colstr(NOF))),grid
on,xlabel('time (sec)'),ylabel('Magnetic field in the probe cavity (Tesla)')
                 hold on
                 subplot(3,1,3)
                 plot(time,force_filtered,'color',char(colstr(NOF))),grid on,xlabel('time
(sec)'),ylabel('Force (lb)')
            end
        end
    end
end
```

Appendix B. The Code for the Shim Stack Deflection Analysis

```
%Shim stack deflection calculations for a three disks on eachother
%inclusing the shear effects based on 080609-1 page 9
clear all
clc
global R r1 r2 r3 a l h0 h1 h2 h3 E nu p0 k G h r n J
r1=0.7085;
r2=0.5905;
r3=0.512;
a=0.3525;
l=0.615;
th1=0.008; % thickness of the disks
th2=0.008;
th3=0.008;
h0=-th1/2;
h1=th1/2;
h2=th2+h1;
h3=h2+th3;
r_inc=0.00001; %increments in r vectors
fig_num=2;
E=2.9e7;
nu=0.3;
G=E/(2*(1+nu));
p0=1000;
k=.833; %shear correction factor
%****************CASE A***********************
r3_caseA=[a:r_inc:r3];
r2_caseA=[r3:r_inc:r2];
r1_caseA=[r2:r_inc:r1];
r_caseA=[r3_caseA,r2_caseA,r1_caseA];
ic_caseA=zeros(39,1);
options = optimset('NonlEqnAlgorithm','lm','MaxFunEvals',50000,'TolFun',1e-15,'TolX',1e-
15,'DiffMaxChange',1e-1,'MaxIter',10000);
display('A')
tic
[ABC_A] = fsolve(@caseA_30sep09_final,ic_caseA,options);
toc
A1_A=ABC_A(1);
B1_A=ABC_A(2);
C1_A=ABC_A(3);
d1_A=ABC_A(4);
L1_A=ABC_A(5)
H1_A=ABC_A(6);
A2_A=ABC_A(7);
B2_A=ABC_A(8);
C2_A=ABC_A(9);
d2_A=ABC_A(10);
L2_A=ABC_A(11)
H2_A=ABC_A(12);
A3_A=ABC_A(13);
B3_A=ABC_A(14);
C3_A=ABC_A(15);
d3_A=ABC_A(16);
L3_A=ABC_A(17)
H3_A=ABC_A(18);
M1_A=ABC_A(19);
N1_A=ABC_A(20);
S1_A=ABC_A(21);
T1_A=ABC_A(22);
V1_A=ABC_A(23)
M2_A=ABC_A(24);
N2_A=ABC_A(25);
S2_A=ABC_A(26);
T2_A=ABC_A(27);
V2_A=ABC_A(28)
```

```
M3_A=ABC_A(29);
N3_A=ABC_A(30);
S3_A=ABC_A(31);
T3_A=ABC_A(32);
V3_A=ABC_A(33)
lambda1_A=ABC_A(34);
lambda2_A=ABC_A(35);
lambda3_A=ABC_A(36);
lambda4_A=ABC_A(37);
lambda5_A=ABC_A(38);
lambda6_A=ABC_A(39);
w1_caseA=A1_A*r1_caseA.^4+B1_A*r1_caseA.^2+C1_A*log(r1_caseA)+d1_A*r1_caseA.^2.*log(r1_caseA)+L1_
A*r1_caseA+H1_A;
w2_caseA=A2_A*r2_caseA.^4+B2_A*r2_caseA.^2+C2_A*log(r2_caseA)+d2_A*r2_caseA.^2.*log(r2_caseA)+L2_
A*r2_caseA+H2_A;
w3_caseA=A3_A*r3_caseA.^4+B3_A*r3_caseA.^2+C3_A*log(r3_caseA)+d3_A*r3_caseA.^2.*log(r3_caseA)+L3_
A*r3_caseA+H3_A;
w_caseA=[w3_caseA,w2_caseA,w1_caseA];
wprime1_caseA=4*A1_A*r1_caseA.^3+2*B1_A*r1_caseA+C1_A./r1_caseA+2*d1_A*r1_caseA.*log(r1_caseA)+d1
_A*r1_caseA+L1_A;
wprime2_caseA=4*A2_A*r2_caseA.^3+2*B2_A*r2_caseA+C2_A./r2_caseA+2*d2_A*r2_caseA.*log(r2_caseA)+d2
_A*r2_caseA+L2_A;
wprime3_caseA=4*A3_A*r3_caseA.^3+2*B3_A*r3_caseA+C3_A./r3_caseA+2*d3_A*r3_caseA.*log(r3_caseA)+d3
_A*r3_caseA+L3_A;
wprime_caseA=[wprime3_caseA,wprime2_caseA,wprime1_caseA];
psi1_caseA=M1_A*r1_caseA.^3+N1_A*r1_caseA+S1_A./r1_caseA+T1_A.*r1_caseA.*log(r1_caseA)+V1_A;
psi2_caseA=M2_A*r2_caseA.^3+N2_A*r2_caseA+S2_A./r2_caseA+T2_A.*r2_caseA.*log(r2_caseA)+V2_A;
psi3_caseA=M3_A*r3_caseA.^3+N3_A*r3_caseA+S3_A./r3_caseA+T3_A.*r3_caseA.*log(r3_caseA)+V3_A;
psi_caseA=[psi3_caseA,psi2_caseA,psi1_caseA];
figure(fig_num)
hold all
plot(r_caseA,w_caseA),grid on
%shear stress in the stepped disk
shear_caseA=G*(wprime_caseA-psi_caseA);
figure(fig_num+1)
hold all
plot(r_caseA,shear_caseA),grid on
%*********************************************************************
%******************CASE B**********************************
r3_caseB=[a:r_inc:r3];
r2_caseB=[r3:r_inc:r2];
r1_caseB=[r2:r_inc:r1];
r_caseB=[r3_caseB,r2_caseB,r1_caseB];
ic_caseB=zeros(39,1);
display('B')
tic
[ABC_B] = fsolve(@caseB_30sep09_final,ic_caseB,options);
toc
A1_B=ABC_B(1);
B1_B=ABC_B(2);
C1_B=ABC_B(3);
d1_B=ABC_B(4);
L1_B=ABC_B(5);
H1_B=ABC_B(6);
A2_B=ABC_B(7);
B2_B=ABC_B(8);
C2_B=ABC_B(9);
d2_B=ABC_B(10);
L2_B=ABC_B(11);
H2_B=ABC_B(12);
A3_B=ABC_B(13);
B3_B=ABC_B(14);
C3_B=ABC_B(15);
d3_B=ABC_B(16);
L3_B=ABC_B(17);
H3_B=ABC_B(18);
M1_B=ABC_B(19);
N1_B=ABC_B(20);
S1_B=ABC_B(21);
```

```
T1_B=ABC_B(22);
V1_B=ABC_B(23);
M2_B=ABC_B(24);
N2_B=ABC_B(25);
S2_B=ABC_B(26);
T2_B=ABC_B(27);
V2_B=ABC_B(28);
M3_B=ABC_B(29);
N3_B=ABC_B(30);
S3_B=ABC_B(31);
T3_B=ABC_B(32);
V3_B=ABC_B(33);
lambda1_B=ABC_B(34);
lambda2_B=ABC_B(35);
lambda3_B=ABC_B(36);
lambda4_B=ABC_B(37);
lambda5_B=ABC_B(38);
lambda6_B=ABC_B(39);
w1_caseB=A1_B*r1_caseB.^4+B1_B*r1_caseB.^2+C1_B*log(r1_caseB)+d1_B*r1_caseB.^2.*log(r1_caseB)+L1_
B*r1_caseB+H1_B;
w2_caseB=A2_B*r2_caseB.^4+B2_B*r2_caseB.^2+C2_B*log(r2_caseB)+d2_B*r2_caseB.^2.*log(r2_caseB)+L2_
B*r2_caseB+H2_B;
w3_caseB=A3_B*r3_caseB.^4+B3_B*r3_caseB.^2+C3_B*log(r3_caseB)+d3_B*r3_caseB.^2.*log(r3_caseB)+L3_
B*r3_caseB+H3_B;
w_caseB=[w3_caseB,w2_caseB,w1_caseB];
figure(fig_num)
hold all
plot(r_caseB,w_caseB),grid on
 %**********************************************************************
 %**********************CASE I*********************************************
th=[th1 th2 th2]; % thickness of the disks
r=[r1 r2 r3];
r=[r a];  %r(n+1)=a,
n=length(th);
h=zeros(1,n);
h(1)=th(1)/2;  %h0=h1
for i=2:n
    h(i)=h(i-1)+th(i);
end
R=r(1);
IC_I=[0 0 0 0 0 0 0];
[ABC_I] = fsolve(@caseI_09Aug09_n_disks_nonzeroinnerradius_r_axis_OK,IC_I);
r_case1=[a:r_inc:r(1)];
w_case1=ABC_I(1)*(r_case1/R).^4+ABC_I(2)*(r_case1/R).^2.*log(r_case1/R)+ABC_I(3)*(r_case1/R).^2+A
BC_I(4)*(r_case1/R)+ABC_I(5);
w_tip_case1=ABC_I(1)*(r(1)/R).^4+ABC_I(2)*(r(1)/R).^2.*log(r(1)/R)+ABC_I(3)*(r(1)/R).^2+ABC_I(4)*
(r(1)/R)+ABC_I(5);
figure(fig_num)
hold all
plot(r_case1,w_case1),grid on
%*********************************end of case 1*********************************
 %*******************************CASE 2*********************************
%find the location of l
for i=1:length(r)
    if l>r(i)
        J=i-1;
        break
    end
end
IC_II=zeros(1,7*n);
[ABC_II] = fsolve(@caseII_09Aug09_n_disks_nonzeroinnerradius_r_axis_OK,IC_II);
for i=1:n
    A(i)=ABC_II(5*i-4);
    B(i)=ABC_II(5*i-3);
    C(i)=ABC_II(5*i-2);
    d(i)=ABC_II(5*i-1);
    F(i)=ABC_II(5*i);
end
figure(fig_num)
```

```
hold all
rr_case2=[];
w_case2=[];
for i=1:n
    r_case2=[r(i+1):r_inc:r(i)];

w_case2=[(A(i)*(r_case2/R).^4+B(i)*(r_case2/R).^2.*log(r_case2/R)+C(i)*(r_case2/R).^2+d(i)*(r_cas
e2/R)+F(i)),w_case2];
    rr_case2=[r_case2,rr_case2];
end
plot(rr_case2,w_case2),grid on
w_tip_case2=ABC_II(1)*(r(1)/R).^4+ABC_II(2)*(r(1)/R).^2.*log(r(1)/R)+ABC_II(3)*(r(1)/R).^2+ABC_II
(4)*(r(1)/R)+ABC_II(5)
%**********************************************************************
% pct_tip_A_B=(w_caseB(length(w_caseB))-w_caseA(length(w_caseA)))/w_caseA(length(w_caseA))*100
% pct_tip_A_II=(w_caseA(length(w_caseA))-w_case2(length(w_case2)))/w_case2(length(w_case2))*100
```

Appendix C. The GUI Code for the Mathematical Model of Mono-tube Dampers

```
function varargout = GUI_hydraulic_damper_09Oct09(varargin)
% GUI_HYDRAULIC_DAMPER_09OCT09 M-file for GUI_hydraulic_damper_09Oct09.fig
%      GUI_HYDRAULIC_DAMPER_09OCT09, by itself, creates a new GUI_HYDRAULIC_DAMPER_09OCT09 or
raises the existing
%      singleton*.
%
%      H = GUI_HYDRAULIC_DAMPER_09OCT09 returns the handle to a new GUI_HYDRAULIC_DAMPER_09OCT09
or the handle to
%      the existing singleton*.
%
%      GUI_HYDRAULIC_DAMPER_09OCT09('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in GUI_HYDRAULIC_DAMPER_09OCT09.M with the given input arguments.
%
%      GUI_HYDRAULIC_DAMPER_09OCT09('Property','Value',...) creates a new
GUI_HYDRAULIC_DAMPER_09OCT09 or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before GUI_hydraulic_damper_04June09_OpeningFunction gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to GUI_hydraulic_damper_09Oct09_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help GUI_hydraulic_damper_09Oct09

% Last Modified by GUIDE v2.5 28-Oct-2009 12:18:14

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @GUI_hydraulic_damper_09Oct09_OpeningFcn, ...
                   'gui_OutputFcn',  @GUI_hydraulic_damper_09Oct09_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT
% --- Executes just before GUI_hydraulic_damper_09Oct09 is made visible.
function GUI_hydraulic_damper_09Oct09_OpeningFcn(hObject, ~, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to GUI_hydraulic_damper_09Oct09 (see VARARGIN)
% Choose default command line output for GUI_hydraulic_damper_09Oct09
handles.output = hObject;
% Update handles structure
guidata(hObject, handles);
% UIWAIT makes GUI_hydraulic_damper_09Oct09 wait for user response (see UIRESUME)
% uiwait(handles.figure1);
% --- Outputs from this function are returned to the command line.
function varargout = GUI_hydraulic_damper_09Oct09_OutputFcn(~, ~, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    structure with handles and user data (see GUIDATA)
% Get default command line output from handles structure
varargout{1} = handles.output;
function p30_Callback(hObject, ~, ~)
% hObject    handle to p30 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of p30 as text
%        str2double(get(hObject,'String')) returns contents of p30 as a double
global p30
p30 = str2double(get(hObject,'String'));
% --- Executes during object creation, after setting all properties.
function p30_CreateFcn(hObject, ~, ~)
% hObject    handle to p30 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
global p30
p30 = str2double(get(hObject,'String'));
function Rp_Callback(hObject, ~, ~)
% hObject    handle to Rp (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of Rp as text
%        str2double(get(hObject,'String')) returns contents of Rp as a double
global Rp
Rp = str2double(get(hObject,'String'));
% --- Executes during object creation, after setting all properties.
function Rp_CreateFcn(hObject, ~, ~)
% hObject    handle to Rp (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
global Rp
Rp = str2double(get(hObject,'String'));
function dr_Callback(hObject, ~, ~)
% hObject    handle to dr (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of dr as text
%        str2double(get(hObject,'String')) returns contents of dr as a double
global dr
dr = str2double(get(hObject,'String'));
% --- Executes during object creation, after setting all properties.
function dr_CreateFcn(hObject, ~, ~)
% hObject    handle to dr (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
global dr
dr = str2double(get(hObject,'String'));
function s10_Callback(hObject, ~, ~)
% hObject    handle to s10 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of s10 as text
%        str2double(get(hObject,'String')) returns contents of s10 as a double
```

```
global s10
s10 = str2double(get(hObject,'String'));
% --- Executes during object creation, after setting all properties.
function s10_CreateFcn(hObject, ~, ~)
% hObject    handle to s10 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
global s10
s10 = str2double(get(hObject,'String'));
function s20_Callback(hObject, ~, ~)
% hObject    handle to s20 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of s20 as text
%        str2double(get(hObject,'String')) returns contents of s20 as a double
global s20
s20 = str2double(get(hObject,'String'));
% --- Executes during object creation, after setting all properties.
function s20_CreateFcn(hObject, ~, ~)
% hObject    handle to s20 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
global s20
s20 = str2double(get(hObject,'String'));
function s30_Callback(hObject, ~, ~)
% hObject    handle to s30 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of s30 as text
%        str2double(get(hObject,'String')) returns contents of s30 as a double
global s30
s30 = str2double(get(hObject,'String'));
% --- Executes during object creation, after setting all properties.
function s30_CreateFcn(hObject, ~, ~)
% hObject    handle to s30 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
global s30
s30 = str2double(get(hObject,'String'));
function m2_Callback(hObject, ~, ~)
% hObject    handle to m2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of m2 as text
%        str2double(get(hObject,'String')) returns contents of m2 as a double
global m2
m2 = str2double(get(hObject,'String'));
% --- Executes during object creation, after setting all properties.
function m2_CreateFcn(hObject, ~, ~)
% hObject    handle to m2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
```

```
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
global m2
m2 = str2double(get(hObject,'String'));
function Ff1_Callback(hObject, ~, ~)
% hObject    handle to Ff1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of Ff1 as text
%        str2double(get(hObject,'String')) returns contents of Ff1 as a double
global Ff1
Ff1 = str2double(get(hObject,'String'));
% --- Executes during object creation, after setting all properties.
function Ff1_CreateFcn(hObject, ~, ~)
% hObject    handle to Ff1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
global Ff1
Ff1 = str2double(get(hObject,'String'));
function Ff2_Callback(hObject, ~, ~)
% hObject    handle to Ff2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of Ff2 as text
%        str2double(get(hObject,'String')) returns contents of Ff2 as a double
global Ff2
Ff2 = str2double(get(hObject,'String'));
% --- Executes during object creation, after setting all properties.
function Ff2_CreateFcn(hObject, ~, ~)
% hObject    handle to Ff2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
global Ff2
Ff2 = str2double(get(hObject,'String'));
function Cd3_Callback(hObject, ~, ~)
% hObject    handle to Cd3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of Cd3 as text
%        str2double(get(hObject,'String')) returns contents of Cd3 as a double
global Cd3
Cd3 = str2double(get(hObject,'String'));
% --- Executes during object creation, after setting all properties.
function Cd3_CreateFcn(hObject, ~, ~)
% hObject    handle to Cd3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
global Cd3
Cd3 = str2double(get(hObject,'String'));
function n_Callback(hObject, ~, ~)
% hObject    handle to n (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of n as text
%        str2double(get(hObject,'String')) returns contents of n as a double
global n
n = str2double(get(hObject,'String'));
% --- Executes during object creation, after setting all properties.
function n_CreateFcn(hObject, ~, ~)
% hObject    handle to n (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
global n
n = str2double(get(hObject,'String'));
function beta_Callback(hObject, ~, ~)
% hObject    handle to beta (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of beta as text
%        str2double(get(hObject,'String')) returns contents of beta as a double
global beta
beta = str2double(get(hObject,'String'));
% --- Executes during object creation, after setting all properties.
function beta_CreateFcn(hObject, ~, ~)
% hObject    handle to beta (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
global beta
beta = str2double(get(hObject,'String'));
function rho_Callback(hObject, ~, ~)
% hObject    handle to rho (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of rho as text
%        str2double(get(hObject,'String')) returns contents of rho as a double
global rho
rho = str2double(get(hObject,'String'));
% --- Executes during object creation, after setting all properties.
function rho_CreateFcn(hObject, ~, ~)
% hObject    handle to rho (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
global rho
rho = str2double(get(hObject,'String'));
function E_Callback(hObject, ~, ~)
% hObject    handle to E (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of E as text
%        str2double(get(hObject,'String')) returns contents of E as a double
global E
E = str2double(get(hObject,'String'));
% --- Executes during object creation, after setting all properties.
function E_CreateFcn(hObject, ~, ~)
% hObject    handle to E (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
```

```
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
global E
E = str2double(get(hObject,'String'));
function nu_Callback(hObject, ~, ~)
% hObject    handle to nu (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of nu as text
%        str2double(get(hObject,'String')) returns contents of nu as a double
global nu
nu = str2double(get(hObject,'String'));
% --- Executes during object creation, after setting all properties.
function nu_CreateFcn(hObject, ~, ~)
% hObject    handle to nu (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
global nu
nu = str2double(get(hObject,'String'));
function a_Callback(hObject, ~, ~)
% hObject    handle to a (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of a as text
%        str2double(get(hObject,'String')) returns contents of a as a double
global a
a = str2double(get(hObject,'String'));
% --- Executes during object creation, after setting all properties.
function a_CreateFcn(hObject, ~, ~)
% hObject    handle to a (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
global a
a = str2double(get(hObject,'String'));
% function b_Callback(hObject, ~, ~)
% hObject    handle to b (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
%
% Hints: get(hObject,'String') returns contents of b as text
%        str2double(get(hObject,'String')) returns contents of b as a double
% global b
% b = str2double(get(hObject,'String'));
%
% --- Executes during object creation, after setting all properties.
% function b_CreateFcn(hObject, ~, ~)
% hObject    handle to b (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
% if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
%     set(hObject,'BackgroundColor','white');
% end
% global b
% b = str2double(get(hObject,'String'));
```

```
function th_Callback(hObject, ~, ~)
% hObject    handle to th (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of th as text
%        str2double(get(hObject,'String')) returns contents of th as a double
global th
th = str2double(get(hObject,'String'));
% --- Executes during object creation, after setting all properties.
function th_CreateFcn(hObject, ~, ~)
% hObject    handle to th (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
global th
th = str2double(get(hObject,'String'));
function Cd2_Callback(hObject, ~, ~)
% hObject    handle to Cd2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of Cd2 as text
%        str2double(get(hObject,'String')) returns contents of Cd2 as a double
global Cd2
Cd2 = str2double(get(hObject,'String'));
% --- Executes during object creation, after setting all properties.
function Cd2_CreateFcn(hObject, ~, ~)
% hObject    handle to Cd2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
global Cd2
Cd2 = str2double(get(hObject,'String'));
function A4_Callback(hObject, ~, ~)
% hObject    handle to A4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of A4 as text
%        str2double(get(hObject,'String')) returns contents of A4 as a double
global A4
A4 = str2double(get(hObject,'String'));
% --- Executes during object creation, after setting all properties.
function A4_CreateFcn(hObject, ~, ~)
% hObject    handle to A4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
global A4
A4 = str2double(get(hObject,'String'));
function Cd4_Callback(hObject, ~, ~)
% hObject    handle to Cd4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Cd4 as text
%        str2double(get(hObject,'String')) returns contents of Cd4 as a double
global Cd4
Cd4 = str2double(get(hObject,'String'));
```

```
% --- Executes during object creation, after setting all properties.
function Cd4_CreateFcn(hObject, ~, ~)
% hObject    handle to Cd4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
global Cd4
Cd4 = str2double(get(hObject,'String'));
function freq_Callback(hObject, ~, ~)
% hObject    handle to freq (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of freq as text
%        str2double(get(hObject,'String')) returns contents of freq as a double
global freq
freq = str2double(get(hObject,'String'));
% --- Executes during object creation, after setting all properties.
function freq_CreateFcn(hObject, ~, ~)
% hObject    handle to freq (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
global freq
freq = str2double(get(hObject,'String'));
function amp_Callback(hObject, ~, ~)
% hObject    handle to amp (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of amp as text
%        str2double(get(hObject,'String')) returns contents of amp as a double
global amp
amp = str2double(get(hObject,'String'));
% --- Executes during object creation, after setting all properties.
function amp_CreateFcn(hObject, ~, ~)
% hObject    handle to amp (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
global amp
amp = str2double(get(hObject,'String'));
function tspan_Callback(hObject, ~, ~)
% hObject    handle to tspan (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of tspan as text
%        str2double(get(hObject,'String')) returns contents of tspan as a double
global tspan
tspan = str2double(get(hObject,'String'));
tspan=0:0.001:tspan; %ode solver returns solutions only at the times in the vector tspan
% --- Executes during object creation, after setting all properties.
function tspan_CreateFcn(hObject, ~, ~)
% hObject    handle to tspan (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
```

```
    set(hObject,'BackgroundColor','white');
end
global tspan
tspan = str2double(get(hObject,'String'));
tspan=0:0.001:tspan; %ode solver returns solutions only at the times in the vector tspan
% --- Executes on button press in solve.
function solve_Callback(~, ~, ~) %#ok<*DEFNU>
% hObject    handle to solve (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global dr Ar Rp Ap p30 s10 s20 s30 V10 V20 V30 X1 X1dot tspan P T Pvap cavhandle def_counter
disk_def tTime options wait filename
tic
Ar=pi/4*dr^2;
Ap=pi*Rp^2;
V10=(Ap-Ar)*s10;
V20=Ap*s20;
V30=Ap*s30;
options = optimset('Algorithm','levenberg-marquardt','MaxFunEvals',50000,'TolFun',1e-
15,'TolX',1e-15,'DiffMaxChange',1e-1,'MaxIter',10000,'Display','off'); %options for fsolve to
solve the deflection of the shim (or disk)
def_counter=1;  % a counter to increase the length of the length of the deflection matrix
tTime=zeros(1,10000);
disk_def=zeros(1,10000);
X1=inline('amp*(sin(pi*freq*T)).^2');
X1dot=inline('pi*freq*amp*sin(2*pi*freq*T)');
p0=[p30 p30 p30 0];
wait = waitbar(0,'Running code, Please wait...');
options_ode=odeset('InitialStep',5e-8,'AbsTol',1e-7,'RelTol',1e-4,'MaxStep',1e-2); %initial time
step for ODE solver and other solver parameters
[T,P]=ode23t(@pressure_Englishunit_09Oct09,tspan,p0);%,options_ode);
close(wait)
if cavhandle==0
    for j=1:length(P(:,1))
        if P(j,1)<Pvap
            P(j,1)=Pvap;
        end
        if P(j,2)<Pvap
            P(j,2)=Pvap;
        end
    end
end
% save the variables in a file for later use
save(filename,'-append')
warndlg(['elapsed time was ' num2str(toc)],'time spent')
% --- Executes on button press in plot.
function plot_Callback(~, ~, handles)
% hObject    handle to plot (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global Ar Ap Ff1 Ff2 amp freq X1 X1dot P T Fd filename
xx1=-X1(T,amp,freq);
xx1dot=-X1dot(T,amp,freq);
clear Fd
Fd(:,1)=P(:,1)*(Ap-Ar)-P(:,3)*Ap-sign(X1dot(T,amp,freq))*Ff1-sign(P(:,4))*Ff2;
axes(handles.axes1)
hold all
plot(xx1(:)+amp/2,Fd(:)),grid on,xlabel('stroke in'),ylabel('Damper force lb')
axes(handles.axes2)
hold all
plot(xx1dot(:),Fd(:)),grid on,xlabel('Velocity in/s'),ylabel('damper force lb')
save(filename,'-append')
% --- Executes on button press in damp_spec.
function damp_spec_Callback(~, ~, ~)
% hObject    handle to damp_spec (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
close(figure(1))
figure('Name','Damper specs.','NumberTitle','on','Position',[500 100 666 640])
```

```
set(gca,'DataAspectRatio',[1 1 1])
pic=imread('C:\damperfigforGUI','jpg');
image(pic)
function A3_Callback(hObject, ~, ~)
% hObject    handle to A3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of A3 as text
%        str2double(get(hObject,'String')) returns contents of A3 as a double
global A3
A3 = str2double(get(hObject,'String'));
% --- Executes during object creation, after setting all properties.
function A3_CreateFcn(hObject, ~, ~)
% hObject    handle to A3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
global A3
A3 = str2double(get(hObject,'String'));
function Pvap_Callback(hObject, ~, ~)
% hObject    handle to Pvap (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of Pvap as text
%        str2double(get(hObject,'String')) returns contents of Pvap as a double
global Pvap
Pvap = str2double(get(hObject,'String'));
% --- Executes during object creation, after setting all properties.
function Pvap_CreateFcn(hObject, ~, ~)
% hObject    handle to Pvap (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
global Pvap
Pvap = str2double(get(hObject,'String'));
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
% --- Executes on button press in cavitation.
function cavitation_Callback(hObject, ~, ~)
% hObject    handle to cavitation (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hint: get(hObject,'Value') returns toggle state of cavitation
button_state = get(hObject,'Value');
global cavhandle
if button_state == get(hObject,'Max')
    % Toggle button is pressed-take approperiate action
   cavhandle=0;
elseif button_state == get(hObject,'Min')
    % Toggle button is not pressed-take appropriate action
   cavhandle=1;
end
% --- Executes on button press in clear_plot.
function clear_plot_Callback(~, ~, handles)
% hObject    handle to clear_plot (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
cla(handles.axes1)
cla(handles.axes2)
% --- Executes on button press in time_hist.
function time_hist_Callback(~, ~, ~)
% hObject    handle to time_hist (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    structure with handles and user data (see GUIDATA)
global Ar Ap Ff1 Ff2 amp freq X1dot P T Fd
% xx1=-X1(T,amp,freq);
% xx1dot=-X1dot(T,amp,freq);
clear Fd
Fd(:,1)=P(:,1)*(Ap-Ar)-P(:,3)*Ap-sign(X1dot(T,amp,freq))*Ff1-sign(P(:,4))*Ff2;
figure
subplot(2,2,1)
plot(T,P(:,1)),grid on, xlabel('time s'),ylabel('pressure in chamber 1 Psi')
subplot(2,2,2)
plot(T,P(:,2)),grid on, xlabel('time s'),ylabel('pressure in chamber 2 Psi')
subplot(2,2,3)
plot(T,P(:,3)),grid on, xlabel('time s'),ylabel('pressure in chamber 3 Psi')
subplot(2,2,4)
plot(T,Fd),grid on,xlabel('time s'),ylabel('damper force lb')
% --- Executes on button press in pushbutton8.
function pushbutton8_Callback(~, ~, ~)
% hObject    handle to pushbutton8 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
close(figure(2))
figure('Name','Disk specs.','NumberTitle','on','Position',[500 250 666 400])
set(gca,'DataAspectRatio',[1 1 1])
pic2=imread('C:\diskfigforGUI','jpg');
image(pic2)
% --- Executes on button press in plot_new.
function plot_new_Callback(~, ~, ~)
% hObject    handle to plot_new (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global Ar Ap Ff1 Ff2 amp freq X1 X1dot P T Fd
xx1=-X1(T,amp,freq);
xx1dot=-X1dot(T,amp,freq);
clear Fd
Fd(:,1)=P(:,1)*(Ap-Ar)-P(:,3)*Ap-sign(X1dot(T,amp,freq))*Ff1-sign(P(:,4))*Ff2;
close(figure(3))
figure(3)
subplot(2,1,1)
hold all
plot(xx1(:)+amp/2,Fd(:)),grid on,xlabel('stroke in'),ylabel('Damper force lb')
subplot(2,1,2)
hold all
plot(xx1dot(:),Fd(:)),grid on,xlabel('Velocity in/s'),ylabel('damper force lb')
% --- Executes on button press in import_ex_data.
function import_ex_data_Callback(~, ~, ~)
% hObject    handle to import_ex_data (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global exp_x exp_Force exp_Temp exp_v filename
path = 'D:\ORGANIZED DOCUMENTS\HYDRAULID DAMPER';
%path for file window
nfiles = 1;           %number of files to load
first_row = 52;       %first row of data in CSV file
fsamp = 2000;          %sample rate, Hz
figs_together = 1;    %plots figs in one subplot, or in seperate plots
series_together = 1;  %plots data series in same plot
cd(path);
fig=5;
colors = 'bgrcmkbgrcmkbgrcmkbgrcmk';
for i=1:nfiles
    [file, path] = uigetfile('*.asc',['Select CSV Shock Data File #',num2str(i)]);
    cd(path);


    dd = csvread(file,first_row-1,0);

    exp_x    = dd(:,1);
    exp_Force = dd(:,2);
    exp_Temp = dd(:,3);
```

```
    exp_v     = dd(:,7);

    N = length(exp_x);
    exp_t = (0:(N-1)) * 1/fsamp;
    %shockdata = struct('x',exp_x,'v',exp_v,'F',exp_Force,'T',exp_Temp,'time',exp_t);
    %eval(['save ', filename(1:length(filename)-4), ' shockdata'])


    if figs_together==1

        if series_together == 1
            figure(fig+1),
            subplot(231),plot(exp_t,exp_x,colors(i)), xlabel('Time, s'), ylabel('Position,
in'),hold on,grid on
            subplot(232),plot(exp_t,exp_v,colors(i)), xlabel('Time, s'), ylabel('Velocity,
in/s'),hold on,grid on
            subplot(233),plot(exp_t,exp_Force,colors(i)), xlabel('Time, s'), ylabel('Force,
lbs'),hold on,grid on
            subplot(234),plot(exp_t,exp_Temp,colors(i)), xlabel('Time, s'), ylabel('Temperature,
^oF'),hold on,grid on
            subplot(235),plot(exp_v,exp_Force,colors(i)), xlabel('Velocity, in/s'),
ylabel('Force, lbs'),hold on,grid on
            subplot(236),plot(exp_x,exp_Force,colors(i)), xlabel('Position, in'), ylabel('Force,
lbs'),hold on,grid on
        else
            figure(fig+1),
            subplot(231),plot(exp_t,exp_x), xlabel('Time, s'), ylabel('Position, in'),grid on
            subplot(232),plot(exp_t,exp_v), xlabel('Time, s'), ylabel('Velocity, in/s'),grid on
            subplot(233),plot(exp_t,exp_Force), xlabel('Time, s'), ylabel('Force, lbs'),grid on
            subplot(234),plot(exp_t,exp_Temp), xlabel('Time, s'), ylabel('Temperature, ^oF'),grid
on
            subplot(235),plot(exp_v,exp_Force), xlabel('Velocity, in/s'), ylabel('Force,
lbs'),grid on
            subplot(236),plot(exp_x,exp_Force), xlabel('Position, in'), ylabel('Force, lbs'),grid
on
            fig=fig+1;
        end

    else

        if series_together == 1
            figure(fig+1),plot(exp_t,exp_x,colors(i)), xlabel('Time, s'), ylabel('Position, in'),
hold on,grid on
            figure(fig+2),plot(exp_t,exp_v,colors(i)), xlabel('Time, s'), ylabel('Velocity,
in/s'), hold on,grid on
            figure(fig+3),plot(exp_t,exp_Force,colors(i)), xlabel('Time, s'), ylabel('Force,
lbs'), hold on,grid on
            figure(fig+4),plot(exp_t,exp_Temp,colors(i)), xlabel('Time, s'), ylabel('Temperature,
^oF'), hold on,grid on
            figure(fig+5),plot(exp_v,exp_Force,colors(i)), xlabel('Velocity, in/s'),
ylabel('Force, lbs'), hold on,grid on
            figure(fig+6),plot(exp_x,exp_Force,colors(i)), xlabel('Position, in'), ylabel('Force,
lbs'), hold on,grid on
        else
            figure(fig+1),plot(exp_t,exp_x), xlabel('Time, s'), ylabel('Position, in'),grid on
            figure(fig+2),plot(exp_t,exp_v), xlabel('Time, s'), ylabel('Velocity, in/s'),grid on
            figure(fig+3),plot(exp_t,exp_Force), xlabel('Time, s'), ylabel('Force, lbs'),grid on
            figure(fig+4),plot(exp_t,exp_Temp), xlabel('Time, s'), ylabel('Temperature,
^oF'),grid on
            figure(fig+5),plot(exp_v,exp_Force), xlabel('Velocity, in/s'), ylabel('Force,
lbs'),grid on
            figure(fig+6),plot(exp_x,exp_Force), xlabel('Position, in'), ylabel('Force,
lbs'),grid on
            fig=fig+6;
        end

    end
end
save(filename,'-append')
```

```
% --- Executes on button press in Overlaid_sim_ex.
function Overlaid_sim_ex_Callback(~, ~, ~)
% hObject    handle to Overlaid_sim_ex (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global Ar Ap Ff1 Ff2 amp freq X1 X1dot P T Fd exp_x exp_Force  exp_v
xx1=-X1(T,amp,freq);
xx1dot=-X1dot(T,amp,freq);
clear Fd
Fd(:,1)=P(:,1)*(Ap-Ar)-P(:,3)*Ap-sign(X1dot(T,amp,freq))*Ff1-sign(P(:,4))*Ff2;
%close(figure(3))
figure(3)
subplot(2,1,1)
hold all
plot(xx1(:)+amp/2,Fd(:)),grid on,xlabel('stroke in'),ylabel('Damper force lb')
plot(exp_x,exp_Force)       %test results plot
subplot(2,1,2)
hold all
plot(xx1dot(:),Fd(:)),grid on,xlabel('Velocity in/s'),ylabel('damper force lb')
plot(exp_v,exp_Force)       %test results plot
% --- Executes on button press in tip_disk_def.
function tip_disk_def_Callback(~, ~, ~)
% hObject    handle to tip_disk_def (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global disk_def tTime
figure (4)
hold all
plot(tTime,disk_def,'.'),grid on,xlabel('Time sec'),ylabel('disk tip deflection in')
% ---------------------------------------------------------------------
function shim_menu_Callback(~, ~, ~)
% hObject    handle to shim_menu (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% ---------------------------------------------------------------------
function single_disk_Callback(~, ~, ~)
% hObject    handle to single_disk (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global R1 R2 th deflection_case h E nu filename
deflection_case=1;
prompt = {'R1 [in]','R2 [in]','th [in]','E [Psi]','nu []'};
dlg_title = 'Properties of the single disk';
num_lines = 1;
def = {'0.353','0.709','0.008','2.9e7','0.3'};
answer = inputdlg(prompt,dlg_title,num_lines,def);
R1=str2double(cell2mat(answer(1)));
R2=str2double(cell2mat(answer(2)));
th=str2double(cell2mat(answer(3)));
E=str2double(cell2mat(answer(4)));
nu=str2double(cell2mat(answer(5)));
h=th;
save(filename,'-append')
% ---------------------------------------------------------------------
function Case_A_Callback(~, ~, ~)
% hObject    handle to Case_A (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global deflection_case r1 r2 r3 a l h0 h1 h2 h3 E nu k G IC_A filename
deflection_case=2;
prompt = {'a, inner radius of the disk assembly [in]','r, a vector containing the radi of disks
(largest to smallest) [in]','th, a vector containing the thichnesses of disks [in]','l, location
on the disk assembly where pressure acts [in]','E, same for all disks [Psi]','nu, same for all
disks []'};
dlg_title = 'Geometrical properties of the deflecting shim stack';
num_lines = 1;
def = {'0.1','0.6 0.5 0.3','0.012 0.012 0.012','0.101','2.9e7','0.3'};
answer = inputdlg(prompt,dlg_title,num_lines,def);
a=str2double(cell2mat(answer(1)));
```

```
r=str2num(cell2mat(answer(2)));
th=str2num(cell2mat(answer(3)));
l=str2double(cell2mat(answer(4)));
E=str2double(cell2mat(answer(5)));
nu=str2double(cell2mat(answer(6)));
r1=r(1);
r2=r(2);
r3=r(3);
k=5/6; %shear coefficient factor for case A
G=E/(2*(1+nu));
IC_A=zeros(39,1);
h0=-th(1)/2;  %h0=-h1
h1=th(1)/2;
h2=h1+th(2);
h3=h2+th(3);
save(filename,'-append')
% --------------------------------------------------------------------
function Case_B_Callback(~, ~, ~)
% hObject    handle to Case_B (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global deflection_case r1 r2 r3 a l h0 h1 h2 h3 E nu k G IC_B filename
deflection_case=3;
prompt = {'a, inner radius of the disk assembly [in]','r, a vector containing the radi of disks
(largest to smallest) [in]','th, a vector containing the thichnesses of disks [in]','l, location
on the disk assembly where pressure acts [in]','E, same for all disks [Psi]','nu, same for all
disks []'};
dlg_title = 'Geometrical properties of the deflecting shim stack';
num_lines = 1;
def = {'0.1','0.6 0.5 0.3','0.012 0.012 0.012','0.101','2.9e7','0.3'};
answer = inputdlg(prompt,dlg_title,num_lines,def);
a=str2double(cell2mat(answer(1)));
r=str2num(cell2mat(answer(2)));
th=str2num(cell2mat(answer(3)));
l=str2double(cell2mat(answer(4)));
E=str2double(cell2mat(answer(5)));
nu=str2double(cell2mat(answer(6)));
r1=r(1);
r2=r(2);
r3=r(3);
k=5/6; %shear coefficient factor for case B
G=E/(2*(1+nu));
IC_B=zeros(1,39);
h0=-th(1)/2;  %h0=-h1
h1=th(1)/2;
h2=h1+th(2);
h3=h2+th(3);
save(filename,'-append')
% --- Executes during object creation, after setting all properties.
function figure1_CreateFcn(~, ~, ~)
% hObject    handle to figure1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
clc
% --- Executes during object creation, after setting all properties.
function axes1_CreateFcn(~, ~, ~)
% hObject    handle to axes1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% --- Executes during object creation, after setting all properties.
function solve_CreateFcn(~, ~, ~)
% hObject    handle to solve (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% --------------------------------------------------------------------
function stepped_disk_caseII_cpt_Callback(~, ~, ~)
% hObject    handle to stepped_disk_caseII_cpt (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
global deflection_case a l h0 h E nu IC_II J r nd filename R
deflection_case=4;
prompt = {'a, inner radius of the disk assembly [in]','r, a vector containing the radi of disks
(largest to smallest) [in]','th, a vector containing the thichnesses of disks [in]','l, location
on the disk assembly where pressure acts [in]','E, same for all disks [Psi]','nu, same for all
disks []'};
dlg_title = 'Geometrical properties of the deflecting shim stack';
num_lines = 1;
def = {'0.1','0.6 0.5 0.3','0.012 0.012 0.012','0.101','2.9e7','0.3'};
answer = inputdlg(prompt,dlg_title,num_lines,def);
a=str2double(cell2mat(answer(1)));
r=str2num(cell2mat(answer(2)));
th=str2num(cell2mat(answer(3)));
l=str2double(cell2mat(answer(4)));
E=str2double(cell2mat(answer(5)));
nu=str2double(cell2mat(answer(6)));
r=[r a];  %r(n+1)=a,
for i=1:length(r)
    if l>r(i)
        J=i-1;
        break
    end
end
IC_II=zeros(1,21);
nd=length(th); %nd is used because n was used for the gas constant in hyd model
h0=-th(1)/2;  %h0=-h1
h(1)=th(1)/2;
h(2)=h(1)+th(2);
h(3)=h(2)+th(3);
R=r(1);
save(filename,'-append')
function datasave_Callback(hObject, ~, ~)
% hObject    handle to datasave (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of datasave as text
%        str2double(get(hObject,'String')) returns contents of datasave as a double
global filename
filename =(get(hObject,'String'));
save(filename)
% --- Executes during object creation, after setting all properties.
function datasave_CreateFcn(hObject, ~, ~)
% hObject    handle to datasave (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
global filename
filename =(get(hObject,'String'));
save(filename)
```

Appendix D. The MATLAB Function for the Governing Equations of the Model of Mono-tube Dampers

```
function dp=pressure_Englishunit_14June10(t,p)
%the function that contains the ODEs for the hydraulic damper and quarter car rig EOM. The
%tip deflection of the shim stack (case B) and stepped disk (case A) is
%approximated by a line as a function of the pressure difference (p0)
%across the shim stack.
global R1 R2 beta Cd2 Cd3 Ar Ap E A2 A3 rho p30 n  V10 V20 V30 x1dot m2 amp freq x1 Xr  Cd4 A4 nu
h AAA tspan def_counter disk_def tTime deflection_case r1 p0 wait slopeA slopeB pre_def Mb Mt Ks
Kt cavhandle Pvap
p0=abs(p(1)-p(2));
%getting the instaneous displ and velocity from the main code
x1=p(7)-p(5);  %x1=Xt-Xb (x1 is the amount that the damper is compressed or extended)
x1dot=p(8)-p(6);  %x1dot=Vt-Vb;
xr=Xr(t,amp,freq);
%DISK DEFLECTION CALCULATIONS
%% A simple deflection disk
%See 201108-1 page 12 sol1
if deflection_case==1
    AAA=h^3*pi/(12*(1-nu^2));
    w_tip_single_disk =
1/64*pi*R2^4*p0/(AAA*E)+1/32*R2^2*pi*p0*(4*R2^4*log(R2)+4*R2^4*nu*log(R2)-
4*nu*R2^2*R1^2*log(R1)+4*R2^2*R1^2*log(R1)-nu*R2^4+R2^4+nu*R1^4-R1^4)/(AAA*E*(-
nu*R1^2+R2^2+R2^2*nu+R1^2))-1/16*pi*p0*R1^2*R2^2*(4*R2^2*log(R2)+4*R2^2*nu*log(R2)-R2^2*nu+R2^2-
4*log(R1)*R2^2-4*log(R1)*nu*R2^2+R1^2+nu*R1^2).*log(R2)/(AAA*E*(-nu*R1^2+R2^2+R2^2*nu+R1^2))-
1/8*R2^2*pi*p0*R2^2.*log(R2)/(AAA*E)+1/16*R2^2*pi*p0*R2^2/(AAA*E)+1/64*pi*p0*R1^2*(R1^4+16*R2^4*l
og(R1)*nu*log(R2)-16*R2^4*log(R1)^2-5*R2^2*R1^2-6*R2^4-nu*R1^4+16*R2^4*log(R1)*log(R2)-
16*R2^4*log(R1)^2*nu+4*nu*R2^2*R1^2*log(R1)-
8*R2^4*nu*log(R2)+4*log(R1)*nu*R2^4+4*R2^2*R1^2*log(R1)-
8*R2^4*log(R2)+12*log(R1)*R2^4+3*R1^2*R2^2*nu-2*nu*R2^4)/(AAA*E*(-nu*R1^2+R2^2+R2^2*nu+R1^2));
    if w_tip_single_disk<pre_def
        A2=0;
    else
        A2=2*pi*R2*(w_tip_single_disk-pre_def);
    end
    disk_def(def_counter)=w_tip_single_disk;

end
%% STEPPED DISK DEFLECTION (Case A) %Based on 300909-1 page 1 and works done before that
if deflection_case==2
    w_tip_caseA=p0*slopeA;
    if w_tip_caseA<pre_def
        A2=0;
    else
        A2=2*pi*R2*(w_tip_caseA-pre_def);
    end
    %     [ABC_A] = fsolve(@caseA_30sep09_final_for_hyd_mod,IC_A,options);
    %     w_tip_caseA=ABC_A(1)*r1^4+ABC_A(2)*r1^2+ABC_A(3)*log(r1)+ABC_A(4)
    %     *r1^2*log(r1)+ABC_A(5)*r1+ABC_A(6);
    disk_def(def_counter)=w_tip_caseA;
%     figure(18)
%     hold all
%     plot(p0,w_tip_caseA,'.','color','blue');
%     figure(19)
%     hold all
%     plot(t,w_tip_caseA,'.','color','blue');
%     IC_A=ABC_A;
end
%% Shim stack assembly (Case B)
%Based on 300909-1 page 1 and works done before that
if deflection_case==3
%     [ABC_B] = fsolve(@caseB_30sep09_final_for_hyd_mod,IC_B,options);
%
w_tip_caseB=ABC_B(1)*r1^4+ABC_B(2)*r1^2+ABC_B(3)*log(r1)+ABC_B(4)*r1^2*log(r1)+ABC_B(5)*r1+ABC_B(
6);
    w_tip_caseB=p0*slopeB;
```

```
    if w_tip_caseB<pre_def
        A2=0;
    else
        A2=2*pi*R2*(w_tip_caseB-pre_def);
    end
    disk_def(def_counter)=w_tip_caseB;
%     figure(16)
%     hold all
%     plot(p0,w_tip_caseB,'.','color','green');
%     figure(17)
%     hold all
%     plot(t,w_tip_caseB,'.','color','green');
%     IC_B=ABC_B; %values of the solution will be used as IC for the next time step
end
%% Cavitation
if cavhandle==0
    if p(1)<Pvap
            p(1)=Pvap;
    end
    if p(2)<Pvap
            p(2)=Pvap;
    end
end
%% Finding the piston and damper force
fd=p(1)*(Ap-Ar)-p(3)*Ap+p30*Ar; %p30*Ar is added to Fd to account for static spring force exerted
by the damper in static conditions (See 090610-1 page 21)
tTime(def_counter)=t;
def_counter=def_counter+1;
%% ODE system
dp=zeros(8,1);
    dp(1)=-beta * (Cd2 * A2 * sign(p(1)-p(2)) * sqrt(0.2e1) * sqrt(0.1e1 / rho * abs(Cd3 ^ 2 * A3
^ 2 * (p(2) - p(1)) / (Cd3 ^ 2 * A3 ^ 2 + Cd2 ^ 2 * A2 ^ 2))) + Cd4 * A4 * sign(-p(2) + p(1)) *
sqrt(0.2e1) * sqrt(0.1e1 / rho * abs(-p(2) + p(1))) + Ap * p(8) - Ap * p(6) - Ar * p(8) + Ar *
p(6)) / (Ap * p(7) - Ap * p(5) - Ar * p(7) + Ar * p(5) + V10);
    %dp(1)=-beta*(-sign(p(2)-p(1))*(Cd2*A2+Cd4*A4)*2^(1/2)*(abs(Cd3^2*A3^2*(-
p(2)+p(1))/(rho*(Cd3^2*A3^2+Cd2^2*A2^2))))^(1/2)+x1dot*Ap-x1dot*Ar)/(x1*Ap-x1*Ar+V10);
    dp(2)=-beta * (-Cd2 * A2 * sign(p(1)-p(2)) * sqrt(0.2e1) * sqrt(0.1e1 / rho * abs(Cd3 ^ 2 *
A3 ^ 2 * (-p(2) + p(1)) / (Cd3 ^ 2 * A3 ^ 2 + Cd2 ^ 2 * A2 ^ 2))) - Cd4 * A4 * sign(-p(2) + p(1))
* sqrt(0.2e1) * sqrt(0.1e1 / rho * abs(-p(2) + p(1))) + Ap * p(6) + Ap *
(1/Ap/n*V30*p30^(1/n)*p(3)^(-1/n-1)*p(4)-p(8))) / (V20 + Ap * p(5) + V30 -
(p30*V30^n/p(3))^(1/n));
    %dp(2)=-beta*p(3)^(1/n)*(-sign(p(2)-p(1))*(Cd2*A2+Cd4*A4)*2^(1/2)*(abs(Cd3^2*A3^2*(-
p(2)+p(1))/(rho*(Cd3^2*A3^2+Cd2^2*A2^2))))^(1/2)*n-V30*p30^(1/n)*p(3)^(-
(1+n)/n)*p(4)+x1dot*n*Ap)/(n*(-V20*p(3)^(1/n)+Ap*x1*p(3)^(1/n)-V30*p(3)^(1/n)+p30^(1/n)*V30));
    dp(3)=p(4);
% if p(4)>0
%     dp(4)=1/(m2*V30*p30^(1/n)*p(3)^(-(1+n)/n)*n)*(Ap^2*n^2*p(2)-Ap^2*n^2*p(3)-
Ff2*n^2*Ap+m2*V30*p30^(1/n)*p(3)^(-(1+2*n)/n)*p(4)^2+m2*V30*p30^(1/n)*p(3)^(-
(1+2*n)/n)*p(4)^2*n+m2*n^2*Ap*(1/Mt*(fd+Ks*p(5)-(Ks+Kt)*p(7)+Kt*xr)));
% else
%     dp(4)=1/(m2*V30*p30^(1/n)*p(3)^(-(1+n)/n)*n)*(Ap^2*n^2*p(2)-
Ap^2*n^2*p(3)+Ff2*n^2*Ap+m2*V30*p30^(1/n)*p(3)^(-(1+2*n)/n)*p(4)^2+m2*V30*p30^(1/n)*p(3)^(-
(1+2*n)/n)*p(4)^2*n+m2*n^2*Ap*(1/Mt*(fd+Ks*p(5)-(Ks+Kt)*p(7)+Kt*xr)));
%end
%   dp(4)=(Ap ^ 2 * n ^ 2 * p(2) - Ap ^ 2 * n ^ 2 * p(3) + m2 * V30 * p30 ^ 1 / n * p(3) ^ (-(1 +
2 * n) / n) * p(4) ^ 2 + m2 * V30 * p30 ^ 1 / n * p(3) ^ (-(1 + 2 * n) / n) * p(4) ^ 2 * n + m2 *
(1/Mt*(fd+Ks*p(5)-(Ks+Kt)*p(7)+Kt*xr)) * n ^ 2 * Ap) / m2 / V30 / p30 ^ 1 / n / p(3) ^ (-(1 + n)
/ n) / n;
    dp(4)=(Ap^2*n^2*p(2)-Ap^2*n^2*p(3)+m2*V30*p30^(1/n)*p(3)^(-
(1+2*n)/n)*p(4)^2+m2*V30*p30^(1/n)*p(3)^(-(1+2*n)/n)*p(4)^2*n+m2*(1/Mt*(fd+Ks*p(5)-
(Ks+Kt)*p(7)+Kt*xr))*n^2*Ap)/(m2*V30*p30^(1/n)*p(3)^(-(1+n)/n)*n);
    dp(5)=p(6);
    dp(6)=1/Mb*(-fd-Ks*p(5)+Ks*p(7));
    dp(7)=p(8);
    dp(8)=1/Mt*(fd+Ks*p(5)-(Ks+Kt)*p(7)+Kt*xr);
waitbar(t/max(tspan),wait,['Running code, Please wait... t= ' num2str(t)]);
```

Appendix E. The MATLAB Function for a Stepped Disk

```
function FF=caseA_30sep09_final_for_hyd_mod(x)
% x is a vector such that x(1)=A1, x(2)=B1, x(3)=C1, ... in the raylegh ritz method
global r1 r2 r3 a l h0 h1 h2 h3 E nu p0 k G
A1=x(1);
B1=x(2);
C1=x(3);
d1=x(4);
L1=x(5);
H1=x(6);
A2=x(7);
B2=x(8);
C2=x(9);
d2=x(10);
L2=x(11);
H2=x(12);
A3=x(13);
B3=x(14);
C3=x(15);
d3=x(16);
L3=x(17);
H3=x(18);
M1=x(19);
N1=x(20);
S1=x(21);
T1=x(22);
V1=x(23);
M2=x(24);
N2=x(25);
S2=x(26);
T2=x(27);
V2=x(28);
M3=x(29);
N3=x(30);
S3=x(31);
T3=x(32);
V3=x(33);
lambda1=x(34);
lambda2=x(35);
lambda3=x(36);
lambda4=x(37);
lambda5=x(38);
lambda6=x(39);
if l>=r2
FF=[(1/3)*p0*pi*l^6-(1/3)*p0*pi*r1^6+lambda1*r2^4+pi*(h1-h0)*G*k*(-(4/3)*T1*r1^6*log(r1)-
4*A1*r2^8+(8/3)*d1*r1^6*log(r1)+(4/3)*T1*r2^6*log(r2)+(8/5)*r2^5*V1+4*A1*r1^8-
(8/3)*d1*r2^6*log(r2)+r2^8*M1+(2/9)*T1*r1^6-(8/5)*r2^5*L1-(8/9)*r2^6*d1-2*r2^4*C1-(8/3)*r2^6*B1-
(2/9)*T1*r2^6-r1^8*M1+(8/5)*r1^5*L1+(8/9)*r1^6*d1+2*r1^4*C1+(8/3)*r1^6*B1-(8/5)*r1^5*V1-
2*r1^4*S1-(4/3)*r1^6*N1+2*r2^4*S1+(4/3)*r2^6*N1);
    (1/2)*p0*pi*l^4-(1/2)*p0*pi*r1^4+lambda1*r2^2+pi*(h1-h0)*G*k*(2*d1*r1^4*log(r1)-
2*B1*r2^4+T1*r2^4*log(r2)-2*d1*r2^4*log(r2)-2*r2^2*C1-
T1*r1^4*log(r1)+(4/3)*r2^3*V1+2*r2^2*S1+2*B1*r1^4+r2^4*N1-(8/3)*A1*r2^6-
(1/4)*T1*r2^4+(8/3)*A1*r1^6+2*r1^2*C1-(4/3)*r1^3*V1-2*r1^2*S1-r1^4*N1-
(2/3)*r1^6*M1+(4/3)*r1^3*L1+(1/2)*r1^4*d1+(1/4)*T1*r1^4+(2/3)*r2^6*M1-(4/3)*r2^3*L1-
(1/2)*r2^4*d1);
    -(1/2)*p0*pi*l^2+(1/2)*p0*pi*r1^2+p0*pi*l^2*log(l)-p0*pi*r1^2*log(r1)+lambda1*log(r2)+pi*(h1-
h0)*G*k*(-2*d1*r2^2*log(r2)+(1/2)*M1*r2^4-2*C1*log(r2)-2*L1*r2+2*C1*log(r1)+T1*r2^2*log(r2)-
2*B1*r2^2+2*d1*r1^2*log(r1)+2*V1*r2+2*S1*log(r2)-2*A1*r2^4-(1/2)*T1*r2^2+2*A1*r1^4-2*S1*log(r1)-
2*V1*r1-N1*r1^2-(1/2)*M1*r1^4+2*L1*r1+2*B1*r1^2+(1/2)*T1*r1^2-T1*r1^2*log(r1)+N1*r2^2);
    -(1/8)*p0*pi*l^4+(1/8)*p0*pi*r1^4+(1/2)*p0*pi*l^4*log(l)-
(1/2)*p0*pi*r1^4*log(r1)+lambda1*r2^2*log(r2)+pi*(h1-h0)*G*k*(-
(1/4)*r2^4*d1+(2/9)*r1^3*L1+(1/4)*r2^4*N1-(2/9)*r1^3*V1-
(1/4)*r1^4*N1+d1*r1^4*log(r1)+(2/9)*r2^6*M1+(8/9)*A1*r1^6-(2/9)*r2^3*L1-
(2/9)*r1^6*M1+(1/4)*r1^4*d1-(8/9)*A1*r2^6-(1/2)*B1*r2^4+(1/2)*B1*r1^4-
d1*r2^4*log(r2)+(2/9)*r2^3*V1+(4/3)*V1*r2^3*log(r2)+2*B1*r1^4*log(r1)+(2/3)*M1*r2^6*log(r2)-
```

```
2*C1*r2^2*log(r2)+2*S1*r2^2*log(r2)-2*B1*r2^4*log(r2)+N1*r2^4*log(r2)-
N1*r1^4*log(r1)+T1*r2^4*log(r2)^2+(8/3)*A1*r1^6*log(r1)-(2/3)*M1*r1^6*log(r1)+2*C1*r1^2*log(r1)-
(4/3)*L1*r2^3*log(r2)-(8/3)*A1*r2^6*log(r2)-2*S1*r1^2*log(r1)-(4/3)*V1*r1^3*log(r1)-
T1*r1^4*log(r1)^2+(4/3)*L1*r1^3*log(r1)-2*d1*r2^4*log(r2)^2+2*d1*r1^4*log(r1)^2);
    (2/3)*p0*pi*l^3-(2/3)*p0*pi*r1^3+lambda1*r2+pi*(h1-h0)*G*k*(2*r2*S1-2*C1*r2+(2/3)*r2^3*N1-
L1*r2^2+(2/3)*T1*r2^3*log(r2)-(4/3)*d1*r2^3*log(r2)+L1*r1^2+(2/5)*r2^5*M1-(2/3)*T1*r1^3*log(r1)-
(2/9)*d1*r2^3+r2^2*V1-(8/5)*A1*r2^5+(4/3)*d1*r1^3*log(r1)-(2/9)*T1*r2^3+(8/5)*A1*r1^5-
(2/3)*r1^3*N1-(2/5)*r1^5*M1+2*C1*r1+(4/3)*B1*r1^3+(2/9)*d1*r1^3-r1^2*V1-2*r1*S1+(2/9)*T1*r1^3-
(4/3)*B1*r2^3);
    p0*pi*l^2-p0*pi*r1^2+lambda1;
    -lambda1*r2^4+lambda3*r3^4+pi*(h2-h0)*G*k*(-(8/3)*d2*r3^6*log(r3)-
4*A2*r3^8+(8/3)*r2^6*B2+2*r2^4*C2+(8/5)*r2^5*L2-(8/5)*r2^5*V2+(8/3)*d2*r2^6*log(r2)+4*A2*r2^8-
r2^8*M2-(4/3)*r2^6*N2-2*r2^4*S2-(4/3)*T2*r2^6*log(r2)-(8/9)*r3^6*d2-
(2/9)*T2*r3^6+(8/5)*r3^5*V2+r3^8*M2+(4/3)*r3^6*N2+2*r3^4*S2-(8/3)*r3^6*B2-2*r3^4*C2-
(8/5)*r3^5*L2+(8/9)*r2^6*d2+(4/3)*T2*r3^6*log(r3)+(2/9)*T2*r2^6);
    -lambda1*r2^2+lambda3*r3^2+pi*(h2-h0)*G*k*(-2*B2*r3^4+(8/3)*A2*r2^6-
2*r2^2*S2+2*r2^2*C2+(1/2)*r2^4*d2+(4/3)*r2^3*L2-(2/3)*r2^6*M2-r2^4*N2+2*B2*r2^4-(4/3)*r2^3*V2-
(1/4)*T2*r3^4+T2*r3^4*log(r3)-2*r3^2*C2-(1/2)*r3^4*d2-
(4/3)*r3^3*L2+(2/3)*r3^6*M2+r3^4*N2+(4/3)*r3^3*V2-(8/3)*A2*r3^6+2*r3^2*S2-
2*d2*r3^4*log(r3)+2*d2*r2^4*log(r2)-T2*r2^4*log(r2)+(1/4)*T2*r2^4);
    -lambda1*log(r2)+lambda3*log(r3)+pi*(h2-h0)*G*k*(-2*C2*log(r3)-
T2*r2^2*log(r2)+2*A2*r2^4+2*B2*r2^2+2*d2*r2^2*log(r2)+2*C2*log(r2)-2*S2*log(r2)+2*L2*r2-
(1/2)*M2*r2^4-N2*r2^2-2*V2*r2-(1/2)*T2*r3^2+T2*r3^2*log(r3)-2*B2*r3^2+2*S2*log(r3)-
2*L2*r3+(1/2)*M2*r3^4-2*A2*r3^4-2*d2*r3^2*log(r3)+N2*r3^2+2*V2*r3+(1/2)*T2*r2^2);
    -lambda1*r2^2*log(r2)+lambda3*r3^2*log(r3)+pi*(h2-h0)*G*k*((8/9)*A2*r2^6-(1/4)*r2^4*N2-
2*S2*r2^2*log(r2)-N2*r2^4*log(r2)-(2/3)*M2*r2^6*log(r2)-(4/3)*V2*r2^3*log(r2)-
(8/3)*A2*r3^6*log(r3)-(4/3)*L2*r3^3*log(r3)+2*B2*r2^4*log(r2)-
2*B2*r3^4*log(r3)+(4/3)*L2*r2^3*log(r2)+(4/3)*V2*r3^3*log(r3)+2*C2*r2^2*log(r2)+(8/3)*A2*r2^6*log
(r2)-T2*r2^4*log(r2)^2+d2*r2^4*log(r2)-d2*r3^4*log(r3)+(2/3)*M2*r3^6*log(r3)+(1/4)*r2^4*d2-
(2/9)*r3^3*L2+(2/9)*r2^3*L2+2*S2*r3^2*log(r3)+2*d2*r2^4*log(r2)^2-(8/9)*A2*r3^6-
2*C2*r3^2*log(r3)+(2/9)*r3^3*V2+(1/4)*r3^4*N2-(1/4)*r3^4*d2-(2/9)*r2^3*V2+T2*r3^4*log(r3)^2-
2*d2*r3^4*log(r3)^2-(1/2)*B2*r3^4+(1/2)*B2*r2^4-(2/9)*r2^6*M2+(2/9)*r3^6*M2+N2*r3^4*log(r3));
    -lambda1*r2+lambda3*r3+pi*(h2-h0)*G*k*(-L2*r3^2+(8/5)*A2*r2^5+(4/3)*B2*r2^3-
(2/3)*T2*r2^3*log(r2)+L2*r2^2+2*C2*r2-(2/5)*r2^5*M2-(2/9)*d2*r3^3+(2/5)*r3^5*M2-(2/9)*T2*r3^3-
(4/3)*B2*r3^3-2*C2*r3+(4/3)*d2*r2^3*log(r2)+(2/3)*r3^3*N2+2*r3*S2+r3^2*V2-(8/5)*A2*r3^5-
(4/3)*d2*r3^3*log(r3)+(2/9)*d2*r2^3+(2/3)*T2*r3^3*log(r3)+(2/9)*T2*r2^3-(2/3)*r2^3*N2-2*r2*S2-
r2^2*V2);
    -lambda1+lambda3;
    pi*(h3-h0)*G*k*(-(4/3)*T3*r3^6*log(r3)-(8/3)*d3*a^6*log(a)-4*A3*a^8-(8/9)*a^6*d3-
(8/5)*a^5*L3+a^8*M3+(8/5)*a^5*V3+2*a^4*S3+(4/3)*a^6*N3-(8/3)*a^6*B3-
(2/9)*T3*a^6+4*A3*r3^8+(4/3)*T3*a^6*log(a)-2*a^4*C3+2*r3^4*C3+(8/9)*r3^6*d3+(8/5)*r3^5*L3-
r3^8*M3-(8/5)*r3^5*V3-2*r3^4*S3-(4/3)*r3^6*N3+(8/3)*r3^6*B3+(2/9)*T3*r3^6+(8/3)*d3*r3^6*log(r3))-
lambda3*r3^4+lambda5*a^4;
    pi*(h3-h0)*G*k*(-2*d3*a^4*log(a)-2*B3*a^4+T3*a^4*log(a)-(8/3)*A3*a^6-
(1/4)*T3*a^4+2*B3*r3^4+(2/3)*a^6*M3-(1/2)*a^4*d3-2*a^2*C3+a^4*N3-(4/3)*a^3*L3+2*a^2*S3-
(2/3)*r3^6*M3+(1/2)*r3^4*d3+2*r3^2*C3-r3^4*N3+(4/3)*r3^3*L3-2*r3^2*S3-
(4/3)*r3^3*V3+(8/3)*A3*r3^6+(1/4)*T3*r3^4+2*d3*r3^4*log(r3)-T3*r3^4*log(r3)+(4/3)*a^3*V3)-
lambda3*r3^2+lambda5*a^2;
    pi*(h3-h0)*G*k*(-(1/2)*M3*r3^4+T3*a^2*log(a)-2*C3*log(a)+2*S3*log(a)+2*V3*a+N3*a^2-
(1/2)*T3*a^2+2*C3*log(r3)-2*d3*a^2*log(a)+2*L3*r3-2*A3*a^4-2*L3*a+(1/2)*M3*a^4-
2*B3*a^2+2*B3*r3^2+2*A3*r3^4-2*S3*log(r3)-2*V3*r3-N3*r3^2+(1/2)*T3*r3^2-
T3*r3^2*log(r3)+2*d3*r3^2*log(r3))-lambda3*log(r3)+lambda5*log(a);
    pi*(h3-h0)*G*k*((2/3)*M3*a^6*log(a)-2*C3*a^2*log(a)-
2*B3*a^4*log(a)+N3*a^4*log(a)+T3*a^4*log(a)^2+2*S3*a^2*log(a)-
(8/3)*A3*a^6*log(a)+(4/3)*V3*a^3*log(a)+(4/3)*L3*r3^3*log(r3)-T3*r3^4*log(r3)^2-
2*S3*r3^2*log(r3)-(4/3)*L3*a^3*log(a)+2*B3*r3^4*log(r3)+2*C3*r3^2*log(r3)+(8/3)*A3*r3^6*log(r3)-
(2/3)*M3*r3^6*log(r3)-(4/3)*V3*r3^3*log(r3)-N3*r3^4*log(r3)-(8/9)*A3*a^6+d3*r3^4*log(r3)-
d3*a^4*log(a)+(1/4)*a^4*N3+(2/9)*r3^3*L3-(2/9)*r3^3*V3-
(1/2)*B3*a^4+(1/2)*B3*r3^4+2*d3*r3^4*log(r3)^2-2*d3*a^4*log(a)^2-
(2/9)*a^3*L3+(8/9)*A3*r3^6+(2/9)*a^3*V3-(2/9)*r3^6*M3-(1/4)*r3^4*N3-
(1/4)*a^4*d3+(1/4)*r3^4*d3+(2/9)*a^6*M3)-lambda3*r3^2*log(r3)+lambda5*a^2*log(a);
    pi*(h3-h0)*G*k*((4/3)*d3*r3^3*log(r3)+(2/3)*T3*a^3*log(a)-L3*a^2-(8/5)*A3*a^5-
(2/9)*d3*a^3+a^2*V3+(2/5)*a^5*M3+(2/3)*a^3*N3+2*a*S3-(2/9)*T3*a^3+L3*r3^2+2*C3*r3-2*C3*a-
(4/3)*B3*a^3+(4/3)*B3*r3^3+(8/5)*A3*r3^5+(2/9)*d3*r3^3-r3^2*V3-(2/5)*r3^5*M3-(2/3)*r3^3*N3-
2*r3*S3+(2/9)*T3*r3^3-(4/3)*d3*a^3*log(a)-(2/3)*T3*r3^3*log(r3))-lambda3*r3+lambda5*a;
    -lambda3+lambda5;
    (1/72)*pi*(h1-h0)^3*E*(-12*nu*S1*r2^4*r1^2+12*nu*S1*r1^4*r2^2-12*nu*N1*r2^6*r1^2-
12*nu*M1*r2^8*r1^2+4*V1*r1^5*r2^2+12*N1*r1^6*r2^2+6*T1*r1^6*r2^2-
12*S1*r1^4*r2^2+12*nu*M1*r1^8*r2^2-4*V1*r2^5*r1^2-12*N1*r2^6*r1^2-
```

```
20*M1*r2^8*r1^2+20*M1*r1^8*r2^2+12*nu*V1*r1^5*r2^2-12*nu*V1*r2^5*r1^2+12*nu*N1*r1^6*r2^2-
6*T1*r2^6*r1^2+12*S1*r2^4*r1^2+12*nu*T1*r1^6*log(r1)*r2^2-
12*nu*T1*r2^6*log(r2)*r1^2+12*T1*r1^6*log(r1)*r2^2-12*T1*r2^6*log(r2)*r1^2)/((1-
nu^2)*r2^2*r1^2)+lambda2*r2^3+pi*(h1-
h0)*G*k*((1/2)*C1*r2^4+(2/3)*d1*r2^6*log(r2)+(1/3)*T1*r1^6*log(r1)-(1/4)*M1*r2^8-
(2/3)*d1*r1^6*log(r1)+(1/4)*M1*r1^8+(2/5)*L1*r2^5+(1/18)*T1*r2^6+A1*r2^8+(2/9)*d1*r2^6-
(2/5)*r2^5*V1-(1/2)*r2^4*S1-(1/3)*r2^6*N1-A1*r1^8-(1/18)*T1*r1^6-(2/5)*L1*r1^5-(1/2)*C1*r1^4-
(2/3)*B1*r1^6+(2/5)*r1^5*V1+(1/2)*r1^4*S1+(1/3)*r1^6*N1-(2/9)*d1*r1^6-
(1/3)*T1*r2^6*log(r2)+(2/3)*B1*r2^6);
    (1/72)*pi*(h1-h0)^3*E*(-12*nu*M1*r2^6*r1^2-
12*nu*V1*r2^3*r1^2+12*nu*N1*r1^4*r2^2+12*V1*r1^3*r2^2+12*M1*r1^6*r2^2-12*nu*N1*r2^4*r1^2-
12*V1*r2^3*r1^2-12*M1*r2^6*r1^2-
12*N1*r2^4*r1^2+12*N1*r1^4*r2^2+12*nu*V1*r1^3*r2^2+12*nu*M1*r1^6*r2^2+12*T1*r1^4*log(r1)*r2^2-
12*T1*r2^4*log(r2)*r1^2+12*nu*T1*r1^4*log(r1)*r2^2-12*nu*T1*r2^4*log(r2)*r1^2)/((1-
nu^2)*r2^2*r1^2)+lambda2*r2+pi*(h1-h0)*G*k*(-(1/2)*T1*r2^4*log(r2)+(2/3)*L1*r2^3-(1/2)*N1*r2^4-
d1*r1^4*log(r1)+d1*r2^4*log(r2)+(1/2)*T1*r1^4*log(r1)+(1/4)*d1*r2^4+B1*r2^4+(1/8)*T1*r2^4-
(1/3)*M1*r2^6-(2/3)*r2^3*V1-r2^2*S1-(1/8)*T1*r1^4+(1/2)*N1*r1^4-(2/3)*L1*r1^3-C1*r1^2-B1*r1^4-
(4/3)*A1*r1^6+(2/3)*r1^3*V1+r1^2*S1+(1/3)*M1*r1^6-(1/4)*d1*r1^4+C1*r2^2+(4/3)*A1*r2^6);
    (1/72)*pi*(h1-h0)^3*E*(-12*nu*M1*r2^4*r1^2+12*nu*M1*r1^4*r2^2-12*M1*r1^4*r2^2+12*r1^2*S1-
12*r2^2*S1+12*nu*S1*r2^2-12*V1*r2^2*r1-12*nu*S1*r1^2-
12*nu*V1*r2*r1^2+12*V1*r2*r1^2+12*nu*V1*r2^2*r1+12*nu*T1*log(r1)*r2^2*r1^2+12*T1*log(r2)*r2^2*r1^
2-12*T1*log(r1)*r2^2*r1^2+12*M1*r2^4*r1^2-12*nu*T1*log(r2)*r2^2*r1^2)/((1-
nu^2)*r2^2*r1^2)+lambda2/r2+pi*(h1-h0)*G*k*(2*S1*log(r1)-T1*r2^2*log(r2)+2*L1*r2-
2*S1*log(r2)+2*d1*r2^2*log(r2)+T1*r1^2*log(r1)+2*B1*r2^2-2*d1*r1^2*log(r1)+2*C1*log(r2)-
(1/2)*M1*r2^4+(1/2)*T1*r2^2-2*V1*r2-N1*r2^2-2*C1*log(r1)-(1/2)*T1*r1^2-2*B1*r1^2-
2*A1*r1^4+2*V1*r1+N1*r1^2+(1/2)*M1*r1^4-2*L1*r1+2*A1*r2^4);
    (1/72)*pi*(h1-h0)^3*E*(-
6*M1*r2^6*r1^2+12*V1*r2^3*r1^2+12*nu*T1*r1^4*log(r1)^2*r2^2+12*nu*S1*log(r1)*r2^2*r1^2+12*N1*r1^4
*log(r1)*r2^2+12*V1*r1^3*log(r1)*r2^2-12*N1*r2^4*log(r2)*r1^2+12*S1*log(r2)*r2^2*r1^2-
12*S1*log(r1)*r2^2*r1^2-
12*nu*V1*r2^3*log(r2)*r1^2+12*nu*V1*r1^3*log(r1)*r2^2+12*nu*M1*r1^6*log(r1)*r2^2+12*nu*N1*r1^4*lo
g(r1)*r2^2-12*nu*M1*r2^6*log(r2)*r1^2-12*nu*N1*r2^4*log(r2)*r1^2+12*M1*r1^6*log(r1)*r2^2-
12*V1*r2^3*log(r2)*r1^2-12*M1*r2^6*log(r2)*r1^2-12*nu*S1*log(r2)*r2^2*r1^2-
12*V1*r1^3*r2^2+6*M1*r1^6*r2^2-12*nu*T1*r2^4*log(r2)^2*r1^2-
12*T1*r2^4*log(r2)^2*r1^2+12*T1*r1^4*log(r1)^2*r2^2-6*T1*r2^4*r1^2+6*T1*r1^4*r2^2)/((1-
nu^2)*r2^2*r1^2)+lambda2*r2*log(r2)+pi*(h1-h0)*G*k*(-(4/3)*A1*r1^6*log(r1)-
(2/9)*A1*r2^6+(1/18)*M1*r2^6-(1/2)*C1*r2^2-S1*r2^2*log(r2)-(2/9)*r1^3*V1+(1/3)*M1*r1^6*log(r1)-
(1/2)*N1*r2^4*log(r2)+B1*r2^4*log(r2)+d1*r2^4*log(r2)^2+C1*r2^2*log(r2)+S1*r1^2*log(r1)+(1/2)*N1*
r1^4*log(r1)-B1*r1^4*log(r1)+(1/4)*T1*r2^4*log(r2)-(1/4)*T1*r1^4*log(r1)+(2/3)*L1*r2^3*log(r2)-
(1/3)*M1*r2^6*log(r2)-(2/3)*L1*r1^3*log(r1)+(2/3)*V1*r1^3*log(r1)+(4/3)*A1*r2^6*log(r2)-
d1*r1^4*log(r1)^2-(2/3)*V1*r2^3*log(r2)-C1*r1^2*log(r1)+(2/9)*r2^3*V1-
(1/8)*N1*r1^4+(1/8)*N1*r2^4-
(1/18)*M1*r1^6+(1/4)*B1*r1^4+(1/2)*C1*r1^2+(2/9)*A1*r1^6+(2/9)*L1*r1^3+(1/16)*T1*r1^4+(1/2)*r2^2*
S1-(2/9)*L1*r2^3-(1/4)*B1*r2^4-(1/16)*T1*r2^4-(1/2)*T1*r2^4*log(r2)^2-
(1/2)*r1^2*S1+(1/2)*T1*r1^4*log(r1)^2);
    (1/72)*pi*(h1-h0)^3*E*(-12*nu*N1*r2^3*r1^2+4*M1*r1^5*r2^2+12*N1*r1^3*r2^2-12*T1*r1^3*r2^2-
4*M1*r2^5*r1^2-12*N1*r2^3*r1^2+12*nu*M1*r1^5*r2^2-12*nu*M1*r2^5*r1^2+12*nu*N1*r1^3*r2^2-
12*S1*r2^2*r1-
12*nu*S1*r2*r1^2+12*S1*r2*r1^2+12*nu*S1*r2^2*r1+12*T1*r1^3*log(r1)*r2^2+12*T1*r2^3*r1^2-
12*nu*T1*r2^3*log(r2)*r1^2+12*nu*T1*r1^3*log(r1)*r2^2-
12*T1*r2^3*log(r2)*r1^2+12*V1*log(r1)*r2^2*r1^2-12*V1*log(r2)*r2^2*r1^2)/((1-
nu^2)*r2^2*r1^2)+lambda2+pi*(h1-h0)*G*k*(-
V1*r2^2+V1*r1^2+(4/3)*d1*r2^3*log(r2)+(4/3)*B1*r2^3+(8/5)*A1*r2^5+(2/3)*T1*r1^3*log(r1)+2*C1*r2+L
1*r2^2-(2/5)*M1*r2^5+(2/9)*d1*r2^3+(2/9)*T1*r2^3-2*S1*r2-(2/3)*N1*r2^3-2*C1*r1-(4/3)*B1*r1^3-
(8/5)*A1*r1^5+2*S1*r1+(2/3)*N1*r1^3+(2/5)*M1*r1^5-(2/9)*d1*r1^3-L1*r1^2-(2/9)*T1*r1^3-
(4/3)*d1*r1^3*log(r1)-(2/3)*T1*r2^3*log(r2));
    pi*(h2-h0)*G*k*((1/4)*M2*r2^8-A2*r2^8-(2/9)*d2*r2^6-
(2/5)*L2*r2^5+(1/3)*r2^6*N2+(1/2)*r2^4*S2+(2/5)*r2^5*V2-(1/2)*C2*r2^4-
(2/3)*B2*r2^6+(2/9)*d2*r3^6+(2/5)*L2*r3^5-(1/3)*r3^6*N2-(1/2)*r3^4*S2-
(2/5)*r3^5*V2+(1/2)*C2*r3^4+(2/3)*B2*r3^6-(1/18)*T2*r2^6-
(2/3)*d2*r2^6*log(r2)+(1/3)*T2*r2^6*log(r2)+(2/3)*d2*r3^6*log(r3)-
(1/3)*T2*r3^6*log(r3)+A2*r3^8+(1/18)*T2*r3^6-(1/4)*M2*r3^8)+(1/72)*pi*(h2-h0)^3*E*(-
12*nu*S2*r3^4*r2^2-12*nu*N2*r3^6*r2^2-12*nu*V2*r3^5*r2^2-
12*T2*r3^6*log(r3)*r2^2+12*T2*r2^6*log(r2)*r3^2-
12*nu*T2*r3^6*log(r3)*r2^2+12*nu*T2*r2^6*log(r2)*r3^2-12*S2*r2^4*r3^2+12*N2*r2^6*r3^2-
6*T2*r3^6*r2^2-12*N2*r3^6*r2^2-
12*nu*M2*r3^8*r2^2+12*nu*M2*r2^8*r3^2+4*V2*r2^5*r3^2+6*T2*r2^6*r3^2+12*S2*r3^4*r2^2-
4*V2*r3^5*r2^2-
```

```
20*M2*r3^8*r2^2+20*M2*r2^8*r3^2+12*nu*S2*r2^4*r3^2+12*nu*N2*r2^6*r3^2+12*nu*V2*r2^5*r3^2)/((1-
nu^2)*r2^2*r3^2)+lambda4*r3^3-lambda2*r2^3;
    pi*(h2-h0)*G*k*((1/2)*N2*r2^4+(2/3)*r2^3*V2+r2^2*S2-(4/3)*A2*r2^6-(1/4)*d2*r2^4-
(2/3)*L2*r2^3+(1/3)*M2*r2^6-C2*r2^2-B2*r2^4+(1/4)*d2*r3^4+(2/3)*L2*r3^3-(1/2)*N2*r3^4-
(1/3)*M2*r3^6+C2*r3^2+B2*r3^4-(1/8)*T2*r2^4-(1/2)*T2*r3^4*log(r3)-
d2*r2^4*log(r2)+(1/2)*T2*r2^4*log(r2)+d2*r3^4*log(r3)+(1/8)*T2*r3^4-(2/3)*r3^3*V2-
r3^2*S2+(4/3)*A2*r3^6)+(1/72)*pi*(h2-h0)^3*E*(-12*nu*M2*r3^6*r2^2-12*nu*V2*r3^3*r2^2-
12*T2*r3^4*log(r3)*r2^2-
12*nu*T2*r3^4*log(r3)*r2^2+12*T2*r2^4*log(r2)*r3^2+12*nu*T2*r2^4*log(r2)*r3^2+12*nu*N2*r2^4*r3^2+
12*M2*r2^6*r3^2-12*nu*N2*r3^4*r2^2-12*V2*r3^3*r2^2-12*M2*r3^6*r2^2+12*V2*r2^3*r3^2-
12*N2*r3^4*r2^2+12*N2*r2^4*r3^2+12*nu*M2*r2^6*r3^2+12*nu*V2*r2^3*r3^2)/((1-
nu^2)*r2^2*r3^2)+lambda4*r3-lambda2*r2;
    pi*(h2-h0)*G*k*(-T2*r3^2*log(r3)+2*S2*log(r2)+2*V2*r2+N2*r2^2-2*C2*log(r2)-2*A2*r2^4-
2*L2*r2+(1/2)*M2*r2^4-2*B2*r2^2+2*C2*log(r3)+2*A2*r3^4+2*L2*r3-(1/2)*M2*r3^4+2*B2*r3^2-
(1/2)*T2*r2^2-2*d2*r2^2*log(r2)+T2*r2^2*log(r2)+2*d2*r3^2*log(r3)+(1/2)*T2*r3^2-2*V2*r3-N2*r3^2-
2*S2*log(r3))+(1/72)*pi*(h2-h0)^3*E*(-12*nu*M2*r3^4*r2^2+12*nu*V2*r2*r3^2-
12*V2*r2*r3^2+12*nu*S2*r3^2-
12*nu*S2*r2^2+12*V2*r2^2*r3+12*T2*log(r3)*r2^2*r3^2+12*nu*T2*log(r2)*r2^2*r3^2-
12*T2*log(r2)*r2^2*r3^2-12*nu*T2*log(r3)*r2^2*r3^2-12*nu*V2*r2^2*r3-
12*M2*r2^4*r3^2+12*M2*r3^4*r2^2+12*r2^2*S2-12*r3^2*S2+12*nu*M2*r2^4*r3^2)/((1-
nu^2)*r2^2*r3^2)+lambda4/r3-lambda2/r2;
    pi*(h2-h0)*G*k*(-(2/9)*r2^3*V2+(1/16)*T2*r2^4-(1/4)*B2*r3^4-(1/2)*C2*r3^2+(1/18)*M2*r3^6-
(2/9)*L2*r3^3+(1/4)*B2*r2^4+(1/2)*C2*r2^2-(1/18)*M2*r2^6+(2/9)*L2*r2^3+(2/9)*A2*r2^6-
(1/2)*r2^2*S2+(2/9)*r3^3*V2-(1/16)*T2*r3^4-(2/9)*A2*r3^6+(1/2)*r3^2*S2+(1/4)*T2*r3^4*log(r3)-
(1/4)*T2*r2^4*log(r2)+(1/8)*N2*r3^4-(1/8)*N2*r2^4-d2*r2^4*log(r2)^2-(1/3)*M2*r3^6*log(r3)-
S2*r3^2*log(r3)+C2*r3^2*log(r3)+(4/3)*A2*r3^6*log(r3)+B2*r3^4*log(r3)+d2*r3^4*log(r3)^2-
(1/2)*N2*r3^4*log(r3)-(2/3)*L2*r2^3*log(r2)+(1/3)*M2*r2^6*log(r2)-(4/3)*A2*r2^6*log(r2)-
B2*r2^4*log(r2)-C2*r2^2*log(r2)+S2*r2^2*log(r2)+(2/3)*V2*r2^3*log(r2)+(1/2)*N2*r2^4*log(r2)-
(2/3)*V2*r3^3*log(r3)-
(1/2)*T2*r3^4*log(r3)^2+(1/2)*T2*r2^4*log(r2)^2+(2/3)*L2*r3^3*log(r3))+(1/72)*pi*(h2-
h0)^3*E*(6*M2*r2^6*r3^2-12*V2*r2^3*r3^2-
6*M2*r3^6*r2^2+12*V2*r3^3*r2^2+12*S2*log(r3)*r2^2*r3^2+12*V2*r2^3*log(r2)*r3^2-
12*nu*T2*r3^4*log(r3)^2*r2^2-12*M2*r3^6*log(r3)*r2^2-
12*N2*r3^4*log(r3)*r2^2+12*M2*r2^6*log(r2)*r3^2+12*nu*T2*r2^4*log(r2)^2*r3^2-
12*T2*r3^4*log(r3)^2*r2^2+6*T2*r2^4*r3^2-6*T2*r3^4*r2^2-12*nu*M2*r3^6*log(r3)*r2^2-
12*nu*N2*r3^4*log(r3)*r2^2+12*nu*M2*r2^6*log(r2)*r3^2+12*nu*S2*log(r2)*r2^2*r3^2+12*nu*V2*r2^3*lo
g(r2)*r3^2-12*V2*r3^3*log(r3)*r2^2+12*N2*r2^4*log(r2)*r3^2-12*S2*log(r2)*r2^2*r3^2-
12*nu*S2*log(r3)*r2^2*r3^2-
12*nu*V2*r3^3*log(r3)*r2^2+12*nu*N2*r2^4*log(r2)*r3^2+12*T2*r2^4*log(r2)^2*r3^2)/((1-
nu^2)*r2^2*r3^2)+lambda4*r3*log(r3)-lambda2*r2*log(r2);
    pi*(h2-h0)*G*k*((4/3)*d2*r3^3*log(r3)+V2*r2^2+2*S2*r2+(2/3)*N2*r2^3-(8/5)*A2*r2^5-
(2/9)*d2*r2^3-L2*r2^2+(2/5)*M2*r2^5-(4/3)*B2*r2^3-2*C2*r2+(2/9)*d2*r3^3+L2*r3^2-
(2/5)*M2*r3^5+(4/3)*B2*r3^3+2*C2*r3-(2/9)*T2*r2^3-(4/3)*d2*r2^3*log(r2)+(2/3)*T2*r2^3*log(r2)-
(2/3)*T2*r3^3*log(r3)+(2/9)*T2*r3^3-2*S2*r3-(2/3)*N2*r3^3+(8/5)*A2*r3^5-V2*r3^2)+(1/72)*pi*(h2-
h0)^3*E*(12*nu*S2*r2*r3^2-12*nu*N2*r3^3*r2^2-12*nu*M2*r3^5*r2^2-
12*S2*r2*r3^2+12*S2*r2^2*r3+12*T2*r2^3*log(r2)*r3^2+12*nu*T2*r2^3*log(r2)*r3^2-
12*T2*r3^3*log(r3)*r2^2-12*nu*T2*r3^3*log(r3)*r2^2-12*nu*S2*r2^2*r3-12*T2*r2^3*r3^2-
12*N2*r3^3*r2^2+12*T2*r3^3*r2^2+4*M2*r2^5*r3^2+12*N2*r2^3*r3^2-4*M2*r3^5*r2^2-
12*V2*log(r3)*r2^2*r3^2+12*V2*log(r2)*r2^2*r3^2+12*nu*N2*r2^3*r3^2+12*nu*M2*r2^5*r3^2)/((1-
nu^2)*r2^2*r3^2)+lambda4-lambda2;
    pi*(h3-h0)*G*k*(-(2/3)*B3*r3^6-(1/3)*T3*a^6*log(a)+(1/4)*M3*r3^8+(2/3)*B3*a^6-(2/5)*a^5*V3-
(1/2)*a^4*S3-
(1/3)*a^6*N3+(2/5)*L3*a^5+(1/18)*T3*a^6+(2/9)*d3*a^6+(1/2)*C3*a^4+A3*a^8+(2/3)*d3*a^6*log(a)+(1/3
)*T3*r3^6*log(r3)-(2/3)*d3*r3^6*log(r3)+(2/5)*r3^5*V3+(1/2)*r3^4*S3+(1/3)*r3^6*N3-(2/5)*L3*r3^5-
(1/18)*T3*r3^6-(2/9)*d3*r3^6-(1/2)*C3*r3^4-A3*r3^8-(1/4)*M3*a^8)+(1/72)*pi*(h3-
h0)^3*E*(12*nu*T3*r3^6*log(r3)*a^2-12*nu*T3*a^6*log(a)*r3^2-12*nu*M3*a^8*r3^2-
6*T3*a^6*r3^2+6*T3*r3^6*a^2+12*nu*M3*r3^8*a^2+12*N3*r3^6*a^2-
12*S3*r3^4*a^2+4*V3*r3^5*a^2+12*S3*a^4*r3^2-4*V3*a^5*r3^2+12*T3*r3^6*log(r3)*a^2-
12*T3*a^6*log(a)*r3^2+12*nu*S3*r3^4*a^2+12*nu*V3*r3^5*a^2-12*nu*V3*a^5*r3^2+12*nu*N3*r3^6*a^2-
12*nu*N3*a^6*r3^2-12*nu*S3*a^4*r3^2-12*N3*a^6*r3^2+20*M3*r3^8*a^2-20*M3*a^8*r3^2)/((1-
nu^2)*r3^2*a^2)+lambda6*a^3-lambda4*r3^3;
    pi*(h3-h0)*G*k*(-B3*r3^4-(1/4)*d3*r3^4+(1/2)*N3*r3^4+B3*a^4+(1/4)*d3*a^4-(2/3)*a^3*V3-a^2*S3-
(1/3)*M3*a^6+(2/3)*L3*a^3+(1/8)*T3*a^4+C3*a^2+(4/3)*A3*a^6-
(1/2)*T3*a^4*log(a)+d3*a^4*log(a)+(1/2)*T3*r3^4*log(r3)-
d3*r3^4*log(r3)+(2/3)*r3^3*V3+r3^2*S3+(1/3)*M3*r3^6-(2/3)*L3*r3^3-(1/8)*T3*r3^4-C3*r3^2-
(4/3)*A3*r3^6-(1/2)*N3*a^4)+(1/72)*pi*(h3-h0)^3*E*(12*T3*r3^4*log(r3)*a^2-
12*nu*T3*a^4*log(a)*r3^2+12*nu*T3*r3^4*log(r3)*a^2-12*V3*a^3*r3^2-
12*nu*N3*a^4*r3^2+12*V3*r3^3*a^2+12*nu*N3*r3^4*a^2+12*M3*r3^6*a^2-
```

```
12*T3*a^4*log(a)*r3^2+12*nu*V3*r3^3*a^2+12*nu*M3*r3^6*a^2-12*nu*M3*a^6*r3^2-12*nu*V3*a^3*r3^2-
12*M3*a^6*r3^2+12*N3*r3^4*a^2-12*N3*a^4*r3^2)/((1-nu^2)*r3^2*a^2)+lambda6*a-lambda4*r3;
    pi*(h3-h0)*G*k*(-2*B3*r3^2+2*S3*log(r3)-T3*a^2*log(a)+(1/2)*T3*a^2+2*B3*a^2-N3*a^2-
(1/2)*M3*a^4+2*L3*a-2*V3*a+2*C3*log(a)+2*A3*a^4-(1/2)*T3*r3^2+T3*r3^2*log(r3)-
2*d3*r3^2*log(r3)+2*d3*a^2*log(a)+N3*r3^2+(1/2)*M3*r3^4-2*L3*r3+2*V3*r3-2*C3*log(r3)-2*A3*r3^4-
2*S3*log(a))+(1/72)*pi*(h3-h0)^3*E*(-
12*T3*log(r3)*r3^2*a^2+12*nu*T3*log(r3)*r3^2*a^2+12*T3*log(a)*r3^2*a^2-12*V3*r3*a^2-
12*M3*r3^4*a^2+12*M3*a^4*r3^2-12*nu*T3*log(a)*r3^2*a^2+12*nu*S3*a^2-12*nu*V3*r3^2*a+12*V3*r3^2*a-
12*nu*S3*r3^2+12*nu*V3*r3*a^2+12*nu*M3*r3^4*a^2-12*nu*M3*a^4*r3^2-12*a^2*S3+12*r3^2*S3)/((1-
nu^2)*r3^2*a^2)+lambda6/a-lambda4/r3;
    pi*(h3-h0)*G*k*(-(1/2)*T3*a^4*log(a)^2-(2/9)*L3*a^3-(2/9)*A3*a^6-
(1/4)*B3*a^4+(1/2)*a^2*S3+(2/9)*a^3*V3-(1/8)*N3*r3^4+(1/8)*N3*a^4-
(1/16)*T3*a^4+(2/9)*L3*r3^3+(1/4)*B3*r3^4-
(1/3)*M3*a^6*log(a)+C3*a^2*log(a)+(1/16)*T3*r3^4+(2/9)*A3*r3^6+(1/18)*M3*a^6+(1/2)*C3*r3^2+(1/4)*
T3*a^4*log(a)-(1/4)*T3*r3^4*log(r3)-S3*a^2*log(a)-(2/3)*V3*a^3*log(a)+(2/3)*L3*a^3*log(a)-
(1/2)*N3*a^4*log(a)+d3*a^4*log(a)^2+(2/3)*V3*r3^3*log(r3)-C3*r3^2*log(r3)-
B3*r3^4*log(r3)+(1/3)*M3*r3^6*log(r3)-(2/3)*L3*r3^3*log(r3)-(2/9)*r3^3*V3+S3*r3^2*log(r3)-
(4/3)*A3*r3^6*log(r3)+(4/3)*A3*a^6*log(a)+B3*a^4*log(a)+(1/2)*N3*r3^4*log(r3)-
d3*r3^4*log(r3)^2+(1/2)*T3*r3^4*log(r3)^2-(1/2)*r3^2*S3-(1/2)*C3*a^2-
(1/18)*M3*r3^6)+(1/72)*pi*(h3-h0)^3*E*(12*nu*T3*r3^4*log(r3)^2*a^2+12*M3*r3^6*log(r3)*a^2-
12*N3*a^4*log(a)*r3^2-12*V3*a^3*log(a)*r3^2-12*M3*a^6*log(a)*r3^2-12*nu*S3*log(a)*r3^2*a^2-
12*V3*r3^3*a^2+6*M3*r3^6*a^2-6*M3*a^6*r3^2-6*T3*a^4*r3^2+6*T3*r3^4*a^2+12*V3*a^3*r3^2-
12*T3*a^4*log(a)^2*r3^2+12*N3*r3^4*log(r3)*a^2+12*T3*r3^4*log(r3)^2*a^2-
12*nu*T3*a^4*log(a)^2*r3^2-12*S3*log(r3)*r3^2*a^2-12*nu*N3*a^4*log(a)*r3^2-
12*nu*V3*a^3*log(a)*r3^2+12*nu*M3*r3^6*log(r3)*a^2+12*nu*N3*r3^4*log(r3)*a^2+12*nu*V3*r3^3*log(r3
)*a^2+12*nu*S3*log(r3)*r3^2*a^2+12*V3*r3^3*log(r3)*a^2+12*S3*log(a)*r3^2*a^2-
12*nu*M3*a^6*log(a)*r3^2)/((1-nu^2)*r3^2*a^2)+lambda6*a*log(a)-lambda4*r3*log(r3);
    pi*(h3-h0)*G*k*((4/3)*d3*a^3*log(a)-(4/3)*B3*r3^3-
(2/9)*d3*r3^3+V3*r3^2+(4/3)*B3*a^3+(2/9)*d3*a^3-(2/3)*N3*a^3-(2/5)*M3*a^5+L3*a^2-
2*S3*a+(2/9)*T3*a^3+2*C3*a+(8/5)*A3*a^5-(2/3)*T3*a^3*log(a)+(2/3)*T3*r3^3*log(r3)-
(4/3)*d3*r3^3*log(r3)+(2/3)*N3*r3^3+(2/5)*M3*r3^5-L3*r3^2+2*S3*r3-(2/9)*T3*r3^3-2*C3*r3-
(8/5)*A3*r3^5-V3*a^2)+(1/72)*pi*(h3-h0)^3*E*(12*V3*log(r3)*r3^2*a^2-12*V3*log(a)*r3^2*a^2-
12*nu*T3*a^3*log(a)*r3^2+12*nu*T3*r3^3*log(r3)*a^2+12*T3*r3^3*log(r3)*a^2-12*S3*r3*a^2-
12*N3*a^3*r3^2+12*N3*r3^3*a^2+4*M3*r3^5*a^2-12*T3*r3^3*a^2-4*M3*a^5*r3^2+12*T3*a^3*r3^2-
12*T3*a^3*log(a)*r3^2-
12*nu*S3*r3^2*a+12*S3*r3^2*a+12*nu*S3*r3*a^2+12*nu*M3*r3^5*a^2+12*nu*N3*r3^3*a^2-
12*nu*M3*a^5*r3^2-12*nu*N3*a^3*r3^2)/((1-nu^2)*r3^2*a^2)+lambda6-lambda4;
    A1*r2^4+B1*r2^2+C1*log(r2)+d1*r2^2*log(r2)+L1*r2+H1-A2*r2^4-B2*r2^2-C2*log(r2)-
d2*r2^2*log(r2)-L2*r2-H2;
    M1*r2^3+N1*r2+S1/r2+T1*r2*log(r2)+V1-M2*r2^3-N2*r2-S2/r2-T2*r2*log(r2)-V2;
    A2*r3^4+B2*r3^2+C2*log(r3)+d2*r3^2*log(r3)+L2*r3+H2-A3*r3^4-B3*r3^2-C3*log(r3)-
d3*r3^2*log(r3)-L3*r3-H3;
    M2*r3^3+N2*r3+S2/r3+T2*r3*log(r3)+V2-M3*r3^3-N3*r3-S3/r3-T3*r3*log(r3)-V3;
    A3*a^4+B3*a^2+C3*log(a)+d3*a^2*log(a)+L3*a+H3;
    M3*a^3+N3*a+S3/a+T3*a*log(a)+V3];
end
if l>=r3 && l<r2
    FF=[lambda1*r2^4-(1/3)*p0*pi*r1^6+(1/3)*p0*pi*r2^6+pi*(h1-
h0)*G*k*((2/9)*T1*r1^6+2*r1^4*C1+(8/3)*r1^6*B1+(8/9)*r1^6*d1+(8/5)*r1^5*L1-(8/5)*r1^5*V1-
(4/3)*r1^6*N1-2*r1^4*S1+(8/3)*d1*r1^6*log(r1)-
(8/3)*d1*r2^6*log(r2)+(4/3)*T1*r2^6*log(r2)+4*A1*r1^8+r2^8*M1-2*r2^4*C1-(8/3)*r2^6*B1-
(8/9)*r2^6*d1-(8/5)*r2^5*L1+(8/5)*r2^5*V1+(4/3)*r2^6*N1+2*r2^4*S1-(2/9)*T1*r2^6-
(4/3)*T1*r1^6*log(r1)-4*A1*r2^8-r1^8*M1);
        lambda1*r2^2-(1/2)*p0*pi*r1^4+(1/2)*p0*pi*r2^4+pi*(h1-h0)*G*k*(-
2*d1*r2^4*log(r2)+(1/4)*T1*r1^4+(8/3)*A1*r1^6-
(2/3)*r1^6*M1+(4/3)*r1^3*L1+2*r1^2*C1+(1/2)*r1^4*d1+T1*r2^4*log(r2)+2*B1*r1^4+2*r2^2*S1+(4/3)*r2^
3*V1+r2^4*N1-(8/3)*A1*r2^6+(2/3)*r2^6*M1-(4/3)*r2^3*L1-2*r2^2*C1-(1/2)*r2^4*d1-
(1/4)*T1*r2^4+2*d1*r1^4*log(r1)-T1*r1^4*log(r1)-2*B1*r2^4-2*r1^2*S1-(4/3)*r1^3*V1-r1^4*N1);
        lambda1*log(r2)+p0*pi*r2^2*log(r2)+(1/2)*p0*pi*r1^2-(1/2)*p0*pi*r2^2-
p0*pi*r1^2*log(r1)+pi*(h1-h0)*G*k*(T1*r2^2*log(r2)-2*d1*r2^2*log(r2)-
2*S1*log(r1)+2*A1*r1^4+2*B1*r1^2+(1/2)*T1*r1^2+2*d1*r1^2*log(r1)-
T1*r1^2*log(r1)+2*C1*log(r1)+2*V1*r2+(1/2)*M1*r2^4+N1*r2^2-2*L1*r2-2*A1*r2^4-2*B1*r2^2-
(1/2)*T1*r2^2+2*S1*log(r2)-2*C1*log(r2)-2*V1*r1-(1/2)*M1*r1^4-N1*r1^2+2*L1*r1);
        lambda1*r2^2*log(r2)+(1/2)*p0*pi*r2^4*log(r2)-(1/8)*p0*pi*r2^4-
(1/2)*p0*pi*r1^4*log(r1)+(1/8)*p0*pi*r1^4+pi*(h1-h0)*G*k*(d1*r1^4*log(r1)-
(2/9)*r2^3*L1+(1/4)*r2^4*N1+(2/9)*r2^3*V1+(2/9)*r2^6*M1-(2/9)*r1^3*V1-2*d1*r2^4*log(r2)^2-
(1/4)*r2^4*d1-
(2/9)*r1^6*M1+(8/3)*A1*r1^6*log(r1)+(4/3)*L1*r1^3*log(r1)+2*C1*r1^2*log(r1)+(2/3)*M1*r2^6*log(r2)
+N1*r2^4*log(r2)-2*B1*r2^4*log(r2)-2*C1*r2^2*log(r2)+T1*r2^4*log(r2)^2+2*S1*r2^2*log(r2)-
```

```
(8/3)*A1*r2^6*log(r2)-(2/3)*M1*r1^6*log(r1)-N1*r1^4*log(r1)-
(4/3)*L1*r2^3*log(r2)+(4/3)*V1*r2^3*log(r2)-T1*r1^4*log(r1)^2+2*B1*r1^4*log(r1)-
(4/3)*V1*r1^3*log(r1)-2*S1*r1^2*log(r1)-d1*r2^4*log(r2)+(2/9)*r1^3*L1-
(1/2)*B1*r2^4+(1/4)*r1^4*d1-(1/4)*r1^4*N1+(8/9)*A1*r1^6-
(8/9)*A1*r2^6+2*d1*r1^4*log(r1)^2+(1/2)*B1*r1^4);
        lambda1*r2+(2/3)*p0*pi*r2^3-(2/3)*p0*pi*r1^3+pi*(h1-
h0)*G*k*((2/5)*r2^5*M1+(2/3)*r2^3*N1+(2/9)*T1*r1^3-2*r1*S1-
r1^2*V1+(8/5)*A1*r1^5+(4/3)*B1*r1^3+(4/3)*d1*r1^3*log(r1)+(2/3)*T1*r2^3*log(r2)+L1*r1^2-
(2/9)*d1*r2^3-2*C1*r2-(8/5)*A1*r2^5-(4/3)*B1*r2^3-(2/9)*T1*r2^3+2*r2*S1+r2^2*V1-
(2/3)*T1*r1^3*log(r1)-L1*r2^2-(4/3)*d1*r2^3*log(r2)-(2/5)*r1^5*M1-
(2/3)*r1^3*N1+(2/9)*d1*r1^3+2*C1*r1);
        lambda1+p0*pi*r2^2-p0*pi*r1^2;
        -lambda1*r2^4+lambda3*r3^4+(1/3)*p0*pi*l^6-(1/3)*p0*pi*r2^6+pi*(h2-
h0)*G*k*((4/3)*T2*r3^6*log(r3)-(8/3)*d2*r3^6*log(r3)-
(4/3)*r2^6*N2+(4/3)*r3^6*N2+r3^8*M2+2*r3^4*S2-2*r2^4*S2-
(8/5)*r2^5*V2+(8/3)*r2^6*B2+2*r2^4*C2+(8/5)*r2^5*L2+(2/9)*T2*r2^6+(8/9)*r2^6*d2+(8/5)*r3^5*V2-
(8/3)*r3^6*B2-2*r3^4*C2-(8/5)*r3^5*L2-4*A2*r3^8+4*A2*r2^8-(2/9)*T2*r3^6-(8/9)*r3^6*d2-
r2^8*M2+(8/3)*d2*r2^6*log(r2)-(4/3)*T2*r2^6*log(r2));
        -lambda1*r2^2+lambda3*r3^2+(1/2)*p0*pi*l^4-(1/2)*p0*pi*r2^4+pi*(h2-h0)*G*k*(-
(2/3)*r2^6*M2+2*r2^2*C2+(1/2)*r2^4*d2-r2^4*N2+T2*r3^4*log(r3)+(2/3)*r3^6*M2-
2*r3^2*C2+(8/3)*A2*r2^6-(4/3)*r2^3*V2-2*r2^2*S2+(1/4)*T2*r2^4+(4/3)*r2^3*L2+2*B2*r2^4-
2*d2*r3^4*log(r3)-(1/2)*r3^4*d2+r3^4*N2-(8/3)*A2*r3^6-2*B2*r3^4+(4/3)*r3^3*V2+2*r3^2*S2-
(1/4)*T2*r3^4-(4/3)*r3^3*L2-T2*r2^4*log(r2)+2*d2*r2^4*log(r2));
        -lambda1*log(r2)+lambda3*log(r3)-(1/2)*p0*pi*l^2+p0*pi*l^2*log(l)+(1/2)*p0*pi*r2^2-
p0*pi*r2^2*log(r2)+pi*(h2-h0)*G*k*(2*B2*r2^2+2*C2*log(r2)-2*B2*r3^2+2*A2*r2^4-2*V2*r2-
2*S2*log(r2)-N2*r2^2-(1/2)*M2*r2^4+2*L2*r2+(1/2)*T2*r2^2-2*A2*r3^4+2*V2*r3+2*S2*log(r3)-
2*C2*log(r3)+N2*r3^2+(1/2)*M2*r3^4-2*L2*r3-(1/2)*T2*r3^2+T2*r3^2*log(r3)-2*d2*r3^2*log(r3)-
T2*r2^2*log(r2)+2*d2*r2^2*log(r2));
        -lambda1*r2^2*log(r2)+lambda3*r3^2*log(r3)-
(1/8)*p0*pi*l^4+(1/2)*p0*pi*l^4*log(l)+(1/8)*p0*pi*r2^4-(1/2)*p0*pi*r2^4*log(r2)+pi*(h2-
h0)*G*k*(2*C2*r2^2*log(r2)+(1/4)*r2^4*d2+(2/3)*M2*r3^6*log(r3)-
(8/3)*A2*r3^6*log(r3)+(2/9)*r3^3*V2-(2/9)*r2^3*V2-
(4/3)*L2*r3^3*log(r3)+2*S2*r3^2*log(r3)+N2*r3^4*log(r3)+(8/3)*A2*r2^6*log(r2)-
(2/3)*M2*r2^6*log(r2)+(4/3)*L2*r2^3*log(r2)+(1/4)*r3^4*N2-2*S2*r2^2*log(r2)-N2*r2^4*log(r2)-
2*C2*r3^2*log(r3)+(4/3)*V2*r3^3*log(r3)+T2*r3^4*log(r3)^2-2*B2*r3^4*log(r3)+2*B2*r2^4*log(r2)-
(1/2)*B2*r3^4-(4/3)*V2*r2^3*log(r2)-T2*r2^4*log(r2)^2+d2*r2^4*log(r2)-
d2*r3^4*log(r3)+(1/2)*B2*r2^4-2*d2*r3^4*log(r3)^2+2*d2*r2^4*log(r2)^2+(8/9)*A2*r2^6-
(1/4)*r3^4*d2-(2/9)*r2^6*M2-(8/9)*A2*r3^6-(2/9)*r3^3*L2+(2/9)*r3^6*M2-
(1/4)*r2^4*N2+(2/9)*r2^3*L2);
        -lambda1*r2+lambda3*r3+(2/3)*p0*pi*l^3-(2/3)*p0*pi*r2^3+pi*(h2-h0)*G*k*(-
(4/3)*d2*r3^3*log(r3)-r2^2*V2+L2*r2^2+(8/5)*A2*r2^5+(2/9)*d2*r2^3-2*r2*S2-(2/3)*r2^3*N2-
(2/5)*r2^5*M2+2*C2*r2+(2/9)*T2*r2^3+(4/3)*B2*r2^3+r3^2*V2-(8/5)*A2*r3^5-L2*r3^2-
(2/9)*d2*r3^3+2*r3*S2+(2/3)*r3^3*N2+(2/5)*r3^5*M2-2*C2*r3-(2/9)*T2*r3^3-
(4/3)*B2*r3^3+(2/3)*T2*r3^3*log(r3)+(4/3)*d2*r2^3*log(r2)-(2/3)*T2*r2^3*log(r2));
        -lambda1+lambda3+p0*pi*l^2-p0*pi*r2^2;
        -lambda3*r3^4+lambda5*a^4+pi*(h3-h0)*G*k*((4/3)*T3*a^6*log(a)+(4/3)*a^6*N3+(8/5)*a^5*V3-
(2/9)*T3*a^6-(8/3)*a^6*B3-2*a^4*C3-(8/9)*a^6*d3-
(8/5)*a^5*L3+2*a^4*S3+a^8*M3+(8/3)*r3^6*B3+2*r3^4*C3+(8/9)*r3^6*d3+(8/5)*r3^5*L3-2*r3^4*S3-
r3^8*M3-(4/3)*r3^6*N3-(8/5)*r3^5*V3+(2/9)*T3*r3^6-(8/3)*d3*a^6*log(a)-(4/3)*T3*r3^6*log(r3)-
4*A3*a^8+(8/3)*d3*r3^6*log(r3)+4*A3*r3^8);
        -lambda3*r3^2+lambda5*a^2+pi*(h3-h0)*G*k*(2*a^2*S3+(2/3)*a^6*M3-2*a^2*C3+(4/3)*a^3*V3-
(1/4)*T3*a^4-(4/3)*a^3*L3-(8/3)*A3*a^6-(1/2)*a^4*d3+(8/3)*A3*r3^6+(1/2)*r3^4*d3-2*r3^2*S3-
(2/3)*r3^6*M3+2*r3^2*C3-(4/3)*r3^3*V3+a^4*N3+(1/4)*T3*r3^4+(4/3)*r3^3*L3-r3^4*N3-2*d3*a^4*log(a)-
T3*r3^4*log(r3)-2*B3*a^4+2*d3*r3^4*log(r3)+T3*a^4*log(a)+2*B3*r3^4);
        -lambda3*log(r3)+lambda5*log(a)+pi*(h3-h0)*G*k*(-2*B3*a^2+2*V3*a+T3*a^2*log(a)-
(1/2)*T3*a^2+2*S3*log(a)-2*L3*a+(1/2)*M3*a^4+N3*a^2-2*A3*a^4-(1/2)*M3*r3^4-
N3*r3^2+2*A3*r3^4+2*B3*r3^2-2*V3*r3+(1/2)*T3*r3^2-2*S3*log(r3)-
2*d3*a^2*log(a)+2*L3*r3+2*d3*r3^2*log(r3)-T3*r3^2*log(r3)-2*C3*log(a)+2*C3*log(r3));
        -lambda3*r3^2*log(r3)+lambda5*a^2*log(a)+pi*(h3-
h0)*G*k*((8/3)*A3*r3^6*log(r3)+(2/9)*a^3*V3-d3*a^4*log(a)+2*S3*a^2*log(a)+(4/3)*V3*a^3*log(a)-
(4/3)*L3*a^3*log(a)+T3*a^4*log(a)^2-2*C3*a^2*log(a)-(8/9)*A3*a^6-(2/9)*r3^3*V3-(1/4)*a^4*d3-
(2/9)*a^3*L3+d3*r3^4*log(r3)+N3*a^4*log(a)+(2/3)*M3*a^6*log(a)-(8/3)*A3*a^6*log(a)+(2/9)*a^6*M3-
2*B3*a^4*log(a)+(4/3)*L3*r3^3*log(r3)-T3*r3^4*log(r3)^2-N3*r3^4*log(r3)-
(2/3)*M3*r3^6*log(r3)+2*B3*r3^4*log(r3)+2*C3*r3^2*log(r3)-(4/3)*V3*r3^3*log(r3)-
2*S3*r3^2*log(r3)-(1/4)*r3^4*N3-2*d3*a^4*log(a)^2+2*d3*r3^4*log(r3)^2-
(1/2)*B3*a^4+(8/9)*A3*r3^6+(1/4)*r3^4*d3+(2/9)*r3^3*L3+(1/4)*a^4*N3-(2/9)*r3^6*M3+(1/2)*B3*r3^4);
        -lambda3*r3+lambda5*a+pi*(h3-h0)*G*k*(-(2/9)*T3*a^3-
(4/3)*B3*a^3+(2/5)*a^5*M3+(2/3)*a^3*N3+2*a*S3+a^2*V3-(2/9)*d3*a^3-2*C3*a-
(8/5)*A3*a^5+(8/5)*A3*r3^5+(4/3)*B3*r3^3-(2/5)*r3^5*M3-(2/3)*r3^3*N3-2*r3*S3-
```

```
r3^2*V3+(2/9)*d3*r3^3+2*C3*r3+(2/9)*T3*r3^3+(2/3)*T3*a^3*log(a)-(4/3)*d3*a^3*log(a)-L3*a^2-
(2/3)*T3*r3^3*log(r3)+(4/3)*d3*r3^3*log(r3)+L3*r3^2);
        -lambda3+lambda5;
        lambda2*r2^3+(1/72)*pi*(h1-h0)^3*E*(-
20*M1*r2^8*r1^2+20*M1*r1^8*r2^2+12*nu*V1*r1^5*r2^2+12*nu*S1*r1^4*r2^2-12*nu*V1*r2^5*r1^2-
12*nu*S1*r2^4*r1^2-12*nu*N1*r2^6*r1^2+12*nu*N1*r1^6*r2^2-
12*nu*T1*r2^6*log(r2)*r1^2+12*T1*r1^6*log(r1)*r2^2-
12*T1*r2^6*log(r2)*r1^2+12*nu*T1*r1^6*log(r1)*r2^2-6*T1*r2^6*r1^2-4*V1*r2^5*r1^2-
12*nu*M1*r2^8*r1^2+12*N1*r1^6*r2^2+4*V1*r1^5*r2^2-12*N1*r2^6*r1^2-
12*S1*r1^4*r2^2+6*T1*r1^6*r2^2+12*S1*r2^4*r1^2+12*nu*M1*r1^8*r2^2)/((1-nu^2)*r1^2*r2^2)+pi*(h1-
h0)*G*k*((2/5)*L1*r2^5-(1/18)*T1*r1^6+(1/3)*r1^6*N1+(1/2)*r1^4*S1+(2/5)*r1^5*V1-
(2/3)*B1*r1^6+(2/3)*d1*r2^6*log(r2)-
(2/3)*d1*r1^6*log(r1)+(1/4)*M1*r1^8+(2/9)*d1*r2^6+(1/2)*C1*r2^4+A1*r2^8+(2/3)*B1*r2^6+(1/18)*T1*r
2^6-(1/3)*r2^6*N1-(1/2)*r2^4*S1-(2/5)*r2^5*V1+(1/3)*T1*r1^6*log(r1)-(1/4)*M1*r2^8-
(1/3)*T1*r2^6*log(r2)-(2/5)*L1*r1^5-(2/9)*d1*r1^6-(1/2)*C1*r1^4-A1*r1^8);
        lambda2*r2+(1/72)*pi*(h1-h0)^3*E*(-12*N1*r2^4*r1^2+12*N1*r1^4*r2^2-12*nu*V1*r2^3*r1^2-
12*nu*M1*r2^6*r1^2+12*nu*M1*r1^6*r2^2+12*nu*V1*r1^3*r2^2+12*T1*r1^4*log(r1)*r2^2-
12*nu*T1*r2^4*log(r2)*r1^2-12*T1*r2^4*log(r2)*r1^2+12*nu*T1*r1^4*log(r1)*r2^2-12*nu*N1*r2^4*r1^2-
12*V1*r2^3*r1^2+12*M1*r1^6*r2^2+12*V1*r1^3*r2^2-12*M1*r2^6*r1^2+12*nu*N1*r1^4*r2^2)/((1-
nu^2)*r1^2*r2^2)+pi*(h1-h0)*G*k*((1/4)*d1*r2^4+(2/3)*L1*r2^3-(1/4)*d1*r1^4+d1*r2^4*log(r2)-
(1/8)*T1*r1^4+(1/3)*M1*r1^6+r1^2*S1+(2/3)*r1^3*V1-
(4/3)*A1*r1^6+(1/2)*T1*r1^4*log(r1)+(1/2)*N1*r1^4+C1*r2^2+B1*r2^4+(4/3)*A1*r2^6+(1/8)*T1*r2^4-
(1/3)*M1*r2^6-r2^2*S1-(2/3)*r2^3*V1-(1/2)*T1*r2^4*log(r2)-(1/2)*N1*r2^4-(2/3)*L1*r1^3-C1*r1^2-
B1*r1^4-d1*r1^4*log(r1));
        lambda2/r2+(1/72)*pi*(h1-h0)^3*E*(12*nu*M1*r1^4*r2^2-
12*nu*M1*r2^4*r1^2+12*T1*log(r2)*r1^2*r2^2-12*T1*log(r1)*r1^2*r2^2-
12*nu*V1*r1^2*r2+12*nu*V1*r1*r2^2+12*V1*r1^2*r2-12*nu*S1*r1^2+12*nu*S1*r2^2-
12*V1*r1*r2^2+12*nu*T1*log(r1)*r1^2*r2^2-12*nu*T1*log(r2)*r1^2*r2^2+12*r1^2*S1-12*r2^2*S1-
12*M1*r1^4*r2^2+12*M1*r2^4*r1^2)/((1-nu^2)*r1^2*r2^2)+pi*(h1-h0)*G*k*(-(1/2)*T1*r1^2-2*L1*r1-
2*C1*log(r1)+(1/2)*M1*r1^4+N1*r1^2+2*V1*r1-
2*A1*r1^4+2*S1*log(r1)+2*d1*r2^2*log(r2)+2*B1*r2^2+2*A1*r2^4+(1/2)*T1*r2^2+2*L1*r2+2*C1*log(r2)-
(1/2)*M1*r2^4-N1*r2^2-2*V1*r2-2*d1*r1^2*log(r1)+T1*r1^2*log(r1)-2*S1*log(r2)-T1*r2^2*log(r2)-
2*B1*r1^2);
        lambda2*r2*log(r2)+(1/72)*pi*(h1-h0)^3*E*(12*V1*r2^3*r1^2+6*M1*r1^6*r2^2-12*V1*r1^3*r2^2-
6*T1*r2^4*r1^2-6*M1*r2^6*r1^2+6*T1*r1^4*r2^2+12*T1*r1^4*log(r1)^2*r2^2-
12*T1*r2^4*log(r2)^2*r1^2+12*S1*log(r2)*r1^2*r2^2-12*S1*log(r1)*r1^2*r2^2-
12*nu*M1*r2^6*log(r2)*r1^2-
12*V1*r2^3*log(r2)*r1^2+12*N1*r1^4*log(r1)*r2^2+12*V1*r1^3*log(r1)*r2^2-
12*nu*N1*r2^4*log(r2)*r1^2+12*M1*r1^6*log(r1)*r2^2+12*nu*S1*log(r1)*r1^2*r2^2-
12*nu*V1*r2^3*log(r2)*r1^2-12*N1*r2^4*log(r2)*r1^2-
12*nu*S1*log(r2)*r1^2*r2^2+12*nu*T1*r1^4*log(r1)^2*r2^2+12*nu*N1*r1^4*log(r1)*r2^2-
12*nu*T1*r2^4*log(r2)^2*r1^2-
12*M1*r2^6*log(r2)*r1^2+12*nu*M1*r1^6*log(r1)*r2^2+12*nu*V1*r1^3*log(r1)*r2^2)/((1-
nu^2)*r1^2*r2^2)+pi*(h1-h0)*G*k*(-(2/3)*V1*r2^3*log(r2)+d1*r2^4*log(r2)^2-(1/2)*N1*r2^4*log(r2)-
(2/3)*L1*r1^3*log(r1)+(1/3)*M1*r1^6*log(r1)+S1*r1^2*log(r1)+(2/3)*V1*r1^3*log(r1)-
(4/3)*A1*r1^6*log(r1)-B1*r1^4*log(r1)-d1*r1^4*log(r1)^2-S1*r2^2*log(r2)+(2/9)*L1*r1^3-
(1/3)*M1*r2^6*log(r2)-(1/8)*N1*r1^4+(1/8)*N1*r2^4-(2/9)*L1*r2^3-
(1/2)*T1*r2^4*log(r2)^2+(2/9)*r2^3*V1+(1/18)*M1*r2^6-
(1/2)*C1*r2^2+(1/2)*T1*r1^4*log(r1)^2+(1/2)*C1*r1^2-
(1/2)*r1^2*S1+(1/2)*r2^2*S1+(4/3)*A1*r2^6*log(r2)+(1/16)*T1*r1^4-(2/9)*A1*r2^6+(1/4)*B1*r1^4-
(2/9)*r1^3*V1-(1/4)*T1*r1^4*log(r1)+(1/4)*T1*r2^4*log(r2)+C1*r2^2*log(r2)+B1*r2^4*log(r2)-
C1*r1^2*log(r1)+(2/9)*A1*r1^6-(1/16)*T1*r2^4-
(1/4)*B1*r2^4+(2/3)*L1*r2^3*log(r2)+(1/2)*N1*r1^4*log(r1)-(1/18)*M1*r1^6);
        lambda2+(1/72)*pi*(h1-h0)^3*E*(12*nu*M1*r1^5*r2^2-12*nu*N1*r2^3*r1^2-
12*nu*M1*r2^5*r1^2+12*nu*N1*r1^3*r2^2-12*nu*S1*r1^2*r2+12*nu*S1*r1*r2^2+12*S1*r1^2*r2-
12*S1*r1*r2^2-12*T1*r2^3*log(r2)*r1^2+12*T1*r1^3*log(r1)*r2^2-
12*nu*T1*r2^3*log(r2)*r1^2+12*nu*T1*r1^3*log(r1)*r2^2-
12*V1*log(r2)*r1^2*r2^2+12*V1*log(r1)*r1^2*r2^2-4*M1*r2^5*r1^2+12*T1*r2^3*r1^2-
12*N1*r2^3*r1^2+4*M1*r1^5*r2^2+12*N1*r1^3*r2^2-12*T1*r1^3*r2^2)/((1-nu^2)*r1^2*r2^2)+pi*(h1-
h0)*G*k*(-L1*r1^2+(2/5)*M1*r1^5+(2/3)*N1*r1^3+2*S1*r1-(8/5)*A1*r1^5-(2/9)*T1*r1^3-(2/9)*d1*r1^3-
(2/3)*T1*r2^3*log(r2)+V1*r1^2+2*C1*r2+(4/3)*B1*r2^3+(8/5)*A1*r2^5+(2/9)*T1*r2^3+L1*r2^2-
(2/5)*M1*r2^5-(2/3)*N1*r2^3-2*S1*r2+(2/9)*d1*r2^3-(4/3)*d1*r1^3*log(r1)+(2/3)*T1*r1^3*log(r1)-
V1*r2^2+(4/3)*d1*r2^3*log(r2)-2*C1*r1-(4/3)*B1*r1^3);
        -lambda2*r2^3+lambda4*r3^3+(1/72)*pi*(h2-h0)^3*E*(6*T2*r2^6*r3^2-
20*M2*r3^8*r2^2+20*M2*r2^8*r3^2-12*nu*M2*r3^8*r2^2-4*V2*r3^5*r2^2-
12*N2*r3^6*r2^2+12*S2*r3^4*r2^2-6*T2*r3^6*r2^2-
12*S2*r2^4*r3^2+12*nu*M2*r2^8*r3^2+4*V2*r2^5*r3^2+12*N2*r2^6*r3^2-12*nu*V2*r3^5*r2^2-
12*nu*S2*r3^4*r2^2-12*nu*N2*r3^6*r2^2+12*nu*V2*r2^5*r3^2+12*nu*S2*r2^4*r3^2+12*nu*N2*r2^6*r3^2-
12*T2*r3^6*log(r3)*r2^2+12*nu*T2*r2^6*log(r2)*r3^2-
```

```
12*nu*T2*r3^6*log(r3)*r2^2+12*T2*r2^6*log(r2)*r3^2)/((1-nu^2)*r2^2*r3^2)+pi*(h2-
h0)*G*k*((2/3)*d2*r3^6*log(r3)+A2*r3^8-(2/3)*B2*r2^6+(1/4)*M2*r2^8+(2/3)*B2*r3^6-(2/9)*d2*r2^6-
(2/5)*L2*r2^5+(1/2)*r2^4*S2-(1/2)*C2*r2^4+(1/3)*r2^6*N2-(1/18)*T2*r2^6+(2/5)*r2^5*V2-
(1/3)*T2*r3^6*log(r3)-(1/4)*M2*r3^8+(2/9)*d2*r3^6+(2/5)*L2*r3^5-(1/2)*r3^4*S2+(1/2)*C2*r3^4-
(1/3)*r3^6*N2+(1/18)*T2*r3^6-(2/5)*r3^5*V2-A2*r2^8-(2/3)*d2*r2^6*log(r2)+(1/3)*T2*r2^6*log(r2));
        -lambda2*r2+lambda4*r3+(1/72)*pi*(h2-h0)^3*E*(-12*nu*N2*r3^4*r2^2+12*N2*r2^4*r3^2-
12*N2*r3^4*r2^2-12*V2*r3^3*r2^2-
12*M2*r3^6*r2^2+12*nu*N2*r2^4*r3^2+12*V2*r2^3*r3^2+12*M2*r2^6*r3^2-12*nu*V2*r3^3*r2^2-
12*nu*M2*r3^6*r2^2+12*nu*M2*r2^6*r3^2+12*nu*V2*r2^3*r3^2+12*nu*T2*r2^4*log(r2)*r3^2-
12*nu*T2*r3^4*log(r3)*r2^2-12*T2*r3^4*log(r3)*r2^2+12*T2*r2^4*log(r2)*r3^2)/((1-
nu^2)*r2^2*r3^2)+pi*(h2-h0)*G*k*(d2*r3^4*log(r3)-(4/3)*A2*r2^6+(4/3)*A2*r3^6-
B2*r2^4+(1/2)*N2*r2^4+(2/3)*r2^3*V2-C2*r2^2-(1/4)*d2*r2^4-(2/3)*L2*r2^3+(1/3)*M2*r2^6-
(1/8)*T2*r2^4+r2^2*S2+B2*r3^4-(2/3)*r3^3*V2-(1/2)*N2*r3^4+C2*r3^2+(1/4)*d2*r3^4+(2/3)*L2*r3^3-
(1/3)*M2*r3^6+(1/8)*T2*r3^4-r3^2*S2-(1/2)*T2*r3^4*log(r3)-d2*r2^4*log(r2)+(1/2)*T2*r2^4*log(r2));
        -lambda2/r2+lambda4/r3+(1/72)*pi*(h2-h0)^3*E*(12*M2*r3^4*r2^2-12*M2*r2^4*r3^2+12*r2^2*S2-
12*r3^2*S2-12*nu*M2*r3^4*r2^2+12*nu*M2*r2^4*r3^2-12*V2*r2*r3^2+12*nu*S2*r3^2+12*V2*r2^2*r3-
12*nu*V2*r2^2*r3+12*nu*V2*r2*r3^2-12*nu*S2*r2^2+12*T2*log(r3)*r2^2*r3^2-12*T2*log(r2)*r2^2*r3^2-
12*nu*T2*log(r3)*r2^2*r3^2+12*nu*T2*log(r2)*r2^2*r3^2)/((1-nu^2)*r2^2*r3^2)+pi*(h2-
h0)*G*k*(2*d2*r3^2*log(r3)+2*A2*r3^4-2*A2*r2^4-T2*r3^2*log(r3)+2*S2*log(r2)-2*C2*log(r2)-2*L2*r2-
2*B2*r2^2-(1/2)*T2*r2^2+(1/2)*M2*r2^4+2*V2*r2+N2*r2^2+2*C2*log(r3)-
2*S2*log(r3)+2*L2*r3+2*B2*r3^2+(1/2)*T2*r3^2-(1/2)*M2*r3^4-2*V2*r3-N2*r3^2-
2*d2*r2^2*log(r2)+T2*r2^2*log(r2));
        -lambda2*r2*log(r2)+lambda4*r3*log(r3)+(1/72)*pi*(h2-h0)^3*E*(12*nu*V2*r2^3*log(r2)*r3^2-
12*nu*V2*r3^3*log(r3)*r2^2-
12*nu*N2*r3^4*log(r3)*r2^2+12*nu*N2*r2^4*log(r2)*r3^2+12*nu*M2*r2^6*log(r2)*r3^2-
12*nu*M2*r3^6*log(r3)*r2^2-12*M2*r3^6*log(r3)*r2^2-12*T2*r3^4*log(r3)^2*r2^2-
12*S2*log(r2)*r2^2*r3^2-12*N2*r3^4*log(r3)*r2^2+12*V2*r2^3*log(r2)*r3^2-12*V2*r3^3*log(r3)*r2^2-
12*nu*T2*r3^4*log(r3)^2*r2^2+12*T2*r2^4*log(r2)^2*r3^2+12*S2*log(r3)*r2^2*r3^2+12*nu*T2*r2^4*log(
r2)^2*r3^2+12*M2*r2^6*log(r2)*r3^2+12*N2*r2^4*log(r2)*r3^2-
12*nu*S2*log(r3)*r2^2*r3^2+12*nu*S2*log(r2)*r2^2*r3^2-
6*M2*r3^6*r2^2+12*V2*r3^3*r2^2+6*M2*r2^6*r3^2-12*V2*r2^3*r3^2+6*T2*r2^4*r3^2-6*T2*r3^4*r2^2)/((1-
nu^2)*r2^2*r3^2)+pi*(h2-h0)*G*k*(-(1/8)*N2*r2^4+(1/8)*N2*r3^4-
(2/9)*r2^3*V2+(2/9)*A2*r2^6+(1/2)*C2*r2^2+(1/18)*M2*r3^6+(4/3)*A2*r3^6*log(r3)+d2*r3^4*log(r3)^2-
(1/2)*N2*r3^4*log(r3)-(1/3)*M2*r3^6*log(r3)-S2*r3^2*log(r3)+B2*r3^4*log(r3)-
(2/3)*V2*r3^3*log(r3)+(2/3)*V2*r2^3*log(r2)+(2/3)*L2*r3^3*log(r3)+C2*r3^2*log(r3)-
C2*r2^2*log(r2)-d2*r2^4*log(r2)^2+(1/2)*N2*r2^4*log(r2)-B2*r2^4*log(r2)-
(4/3)*A2*r2^6*log(r2)+(1/3)*M2*r2^6*log(r2)+S2*r2^2*log(r2)-(2/3)*L2*r2^3*log(r2)-
(1/4)*T2*r2^4*log(r2)+(1/4)*T2*r3^4*log(r3)+(1/2)*r3^2*S2+(1/2)*T2*r2^4*log(r2)^2-
(1/16)*T2*r3^4+(2/9)*r3^3*V2+(2/9)*L2*r2^3-(1/2)*T2*r3^4*log(r3)^2-(1/2)*C2*r3^2-(1/18)*M2*r2^6-
(2/9)*L2*r3^3-(1/2)*r2^2*S2+(1/16)*T2*r2^4-(2/9)*A2*r3^6-(1/4)*B2*r3^4+(1/4)*B2*r2^4);
        -lambda2+lambda4+(1/72)*pi*(h2-h0)^3*E*(12*T2*r3^3*r2^2-12*T2*r2^3*r3^2-12*N2*r3^3*r2^2-
4*M2*r3^5*r2^2+12*N2*r2^3*r3^2+4*M2*r2^5*r3^2-12*nu*N2*r3^3*r2^2-
12*nu*M2*r3^5*r2^2+12*nu*M2*r2^5*r3^2+12*nu*N2*r2^3*r3^2-12*S2*r2*r3^2+12*S2*r2^2*r3-
12*nu*S2*r2^2*r3+12*nu*S2*r2*r3^2+12*V2*log(r2)*r2^2*r3^2-
12*V2*log(r3)*r2^2*r3^2+12*nu*T2*r2^3*log(r2)*r3^2-12*nu*T2*r3^3*log(r3)*r2^2-
12*T2*r3^3*log(r3)*r2^2+12*T2*r2^3*log(r2)*r3^2)/((1-nu^2)*r2^2*r3^2)+pi*(h2-h0)*G*k*(-
(2/3)*T2*r3^3*log(r3)-L2*r2^2-(8/5)*A2*r2^5-2*C2*r2+(2/3)*N2*r2^3-(2/9)*T2*r2^3-(4/3)*B2*r2^3-
(2/9)*d2*r2^3+V2*r2^2+2*S2*r2+(2/5)*M2*r2^5+L2*r3^2+(8/5)*A2*r3^5+2*C2*r3-
(2/3)*N2*r3^3+(2/9)*T2*r3^3-V2*r3^2+(4/3)*B2*r3^3+(2/9)*d2*r3^3-2*S2*r3-
(2/5)*M2*r3^5+(4/3)*d2*r3^3*log(r3)+(2/3)*T2*r2^3*log(r2)-(4/3)*d2*r2^3*log(r2));
        -lambda4*r3^3+lambda6*a^3+(1/72)*pi*(h3-h0)^3*E*(-12*N3*a^6*r3^2-4*V3*a^5*r3^2-
6*T3*a^6*r3^2+12*S3*a^4*r3^2-12*nu*M3*a^8*r3^2+12*N3*r3^6*a^2+4*V3*r3^5*a^2+6*T3*r3^6*a^2-
12*S3*r3^4*a^2+12*nu*M3*r3^8*a^2-20*M3*a^8*r3^2+20*M3*r3^8*a^2-
12*T3*a^6*log(a)*r3^2+12*T3*r3^6*log(r3)*a^2-
12*nu*T3*a^6*log(a)*r3^2+12*nu*T3*r3^6*log(r3)*a^2+12*nu*N3*r3^6*a^2+12*nu*S3*r3^4*a^2-
12*nu*V3*a^5*r3^2-12*nu*N3*a^6*r3^2-12*nu*S3*a^4*r3^2+12*nu*V3*r3^5*a^2)/((1-
nu^2)*r3^2*a^2)+pi*(h3-h0)*G*k*((2/3)*B3*a^6+(1/18)*T3*a^6-(1/3)*a^6*N3-(1/2)*a^4*S3-
(2/5)*a^5*V3+(2/5)*L3*a^5+(2/9)*d3*a^6+(1/2)*C3*a^4+A3*a^8-(1/2)*C3*r3^4-A3*r3^8-(2/3)*B3*r3^6-
(1/18)*T3*r3^6+(1/3)*r3^6*N3+(1/2)*r3^4*S3+(2/5)*r3^5*V3-(2/5)*L3*r3^5-
(2/9)*d3*r3^6+(2/3)*d3*a^6*log(a)-(1/3)*T3*a^6*log(a)+(1/4)*M3*r3^8-(1/4)*M3*a^8-
(2/3)*d3*r3^6*log(r3)+(1/3)*T3*r3^6*log(r3));
        -lambda4*r3+lambda6*a+(1/72)*pi*(h3-h0)^3*E*(-12*M3*a^6*r3^2-12*nu*N3*a^4*r3^2-
12*V3*a^3*r3^2+12*M3*r3^6*a^2+12*nu*N3*r3^4*a^2+12*V3*r3^3*a^2-12*N3*a^4*r3^2+12*N3*r3^4*a^2-
12*nu*T3*a^4*log(a)*r3^2-
12*T3*a^4*log(a)*r3^2+12*nu*T3*r3^4*log(r3)*a^2+12*T3*r3^4*log(r3)*a^2+12*nu*M3*r3^6*a^2-
12*nu*V3*a^3*r3^2-12*nu*M3*a^6*r3^2+12*nu*V3*r3^3*a^2)/((1-nu^2)*r3^2*a^2)+pi*(h3-
h0)*G*k*((4/3)*A3*a^6-(2/3)*a^3*V3-(1/3)*M3*a^6+(2/3)*L3*a^3+(1/4)*d3*a^4+C3*a^2-a^2*S3-
C3*r3^2+r3^2*S3-(4/3)*A3*r3^6+(2/3)*r3^3*V3+B3*a^4+(1/8)*T3*a^4+(1/3)*M3*r3^6-(2/3)*L3*r3^3-
```

```
(1/4)*d3*r3^4-B3*r3^4-(1/8)*T3*r3^4-(1/2)*T3*a^4*log(a)+d3*a^4*log(a)+(1/2)*T3*r3^4*log(r3)-
(1/2)*N3*a^4-d3*r3^4*log(r3)+(1/2)*N3*r3^4);
        -lambda4/r3+lambda6/a+(1/72)*pi*(h3-
h0)^3*E*(12*nu*V3*r3*a^2+12*nu*T3*log(r3)*r3^2*a^2+12*M3*a^4*r3^2-12*M3*r3^4*a^2+12*r3^2*S3-
12*a^2*S3-12*nu*T3*log(a)*r3^2*a^2+12*nu*S3*a^2+12*T3*log(a)*r3^2*a^2-12*T3*log(r3)*r3^2*a^2-
12*nu*S3*r3^2+12*nu*M3*r3^4*a^2-12*nu*M3*a^4*r3^2-12*nu*V3*r3^2*a+12*V3*r3^2*a-12*V3*r3*a^2)/((1-
nu^2)*r3^2*a^2)+pi*(h3-h0)*G*k*(2*B3*a^2+2*C3*log(a)-2*V3*a-(1/2)*M3*a^4+2*L3*a-
N3*a^2+2*A3*a^4+(1/2)*T3*a^2+N3*r3^2-2*A3*r3^4-(1/2)*T3*r3^2-2*B3*r3^2-
2*C3*log(r3)+2*V3*r3+(1/2)*M3*r3^4-2*L3*r3-T3*a^2*log(a)+2*d3*a^2*log(a)-
2*S3*log(a)+T3*r3^2*log(r3)-2*d3*r3^2*log(r3)+2*S3*log(r3));
        -lambda4*r3*log(r3)+lambda6*a*log(a)+(1/72)*pi*(h3-h0)^3*E*(-6*T3*a^4*r3^2-
12*T3*a^4*log(a)^2*r3^2-12*nu*N3*a^4*log(a)*r3^2-12*nu*V3*a^3*log(a)*r3^2-12*M3*a^6*log(a)*r3^2-
12*nu*S3*log(a)*r3^2*a^2-
12*N3*a^4*log(a)*r3^2+12*T3*r3^4*log(r3)^2*a^2+12*nu*V3*r3^3*log(r3)*a^2-12*V3*a^3*log(a)*r3^2-
12*nu*T3*a^4*log(a)^2*r3^2+12*M3*r3^6*log(r3)*a^2+12*S3*log(a)*r3^2*a^2+12*nu*T3*r3^4*log(r3)^2*a
^2+12*V3*r3^3*log(r3)*a^2-12*S3*log(r3)*r3^2*a^2+12*nu*N3*r3^4*log(r3)*a^2-
12*nu*M3*a^6*log(a)*r3^2+12*N3*r3^4*log(r3)*a^2+12*nu*M3*r3^6*log(r3)*a^2-
6*M3*a^6*r3^2+12*V3*a^3*r3^2+6*M3*r3^6*a^2-
12*V3*r3^3*a^2+12*nu*S3*log(r3)*r3^2*a^2+6*T3*r3^4*a^2)/((1-nu^2)*r3^2*a^2)+pi*(h3-
h0)*G*k*((1/2)*C3*r3^2+(1/2)*a^2*S3-(2/9)*A3*a^6-(1/2)*C3*a^2-(1/2)*T3*a^4*log(a)^2-
(1/4)*B3*a^4+(1/16)*T3*r3^4+(2/9)*A3*r3^6-(2/9)*L3*a^3-
(1/16)*T3*a^4+(1/4)*B3*r3^4+(1/2)*T3*r3^4*log(r3)^2+(4/3)*A3*a^6*log(a)+C3*a^2*log(a)-
(2/3)*V3*a^3*log(a)-S3*a^2*log(a)+(2/3)*L3*a^3*log(a)-(1/2)*N3*a^4*log(a)+d3*a^4*log(a)^2-
B3*r3^4*log(r3)-(1/3)*M3*a^6*log(a)-(4/3)*A3*r3^6*log(r3)-C3*r3^2*log(r3)-
(2/3)*L3*r3^3*log(r3)+(1/2)*N3*r3^4*log(r3)-
d3*r3^4*log(r3)^2+S3*r3^2*log(r3)+(2/3)*V3*r3^3*log(r3)+(1/3)*M3*r3^6*log(r3)+B3*a^4*log(a)+(1/4)
*T3*a^4*log(a)-(1/4)*T3*r3^4*log(r3)+(2/9)*L3*r3^3-(1/18)*M3*r3^6+(2/9)*a^3*V3-
(1/2)*r3^2*S3+(1/8)*N3*a^4-(1/8)*N3*r3^4+(1/18)*M3*a^6-(2/9)*r3^3*V3);
        -lambda4+lambda6+(1/72)*pi*(h3-h0)^3*E*(12*nu*S3*r3*a^2-4*M3*a^5*r3^2+12*T3*a^3*r3^2-
12*N3*a^3*r3^2+4*M3*r3^5*a^2-12*T3*r3^3*a^2+12*N3*r3^3*a^2+12*V3*log(r3)*r3^2*a^2-
12*V3*log(a)*r3^2*a^2+12*nu*T3*r3^3*log(r3)*a^2-12*nu*T3*a^3*log(a)*r3^2-
12*T3*a^3*log(a)*r3^2+12*T3*r3^3*log(r3)*a^2-12*nu*M3*a^5*r3^2-
12*nu*N3*a^3*r3^2+12*nu*N3*r3^3*a^2+12*nu*M3*r3^5*a^2-12*nu*S3*r3^2*a+12*S3*r3^2*a-
12*S3*r3*a^2)/((1-nu^2)*r3^2*a^2)+pi*(h3-h0)*G*k*((8/5)*A3*a^5+(4/3)*B3*a^3-
(2/3)*N3*a^3+2*C3*a+(2/9)*T3*a^3-2*S3*a-(2/5)*M3*a^5+L3*a^2+(2/9)*d3*a^3-(8/5)*A3*r3^5-
(4/3)*B3*r3^3+(2/3)*N3*r3^3-2*C3*r3-(2/9)*T3*r3^3+2*S3*r3+(2/5)*M3*r3^5-L3*r3^2-(2/9)*d3*r3^3-
(2/3)*T3*a^3*log(a)+(4/3)*d3*a^3*log(a)-V3*a^2+(2/3)*T3*r3^3*log(r3)-
(4/3)*d3*r3^3*log(r3)+V3*r3^2);
        A1*r2^4+B1*r2^2+C1*log(r2)+d1*r2^2*log(r2)+L1*r2+H1-A2*r2^4-B2*r2^2-C2*log(r2)-
d2*r2^2*log(r2)-L2*r2-H2;
        M1*r2^3+N1*r2+S1/r2+T1*r2*log(r2)+V1-M2*r2^3-N2*r2-S2/r2-T2*r2*log(r2)-V2;
        A2*r3^4+B2*r3^2+C2*log(r3)+d2*r3^2*log(r3)+L2*r3+H2-A3*r3^4-B3*r3^2-C3*log(r3)-
d3*r3^2*log(r3)-L3*r3-H3;
        M2*r3^3+N2*r3+S2/r3+T2*r3*log(r3)+V2-M3*r3^3-N3*r3-S3/r3-T3*r3*log(r3)-V3;
        A3*a^4+B3*a^2+C3*log(a)+d3*a^2*log(a)+L3*a+H3;
        M3*a^3+N3*a+S3/a+T3*a*log(a)+V3];
end
if l>=a && l<r3
    FF=[(1/3)*p0*pi*r2^6-(1/3)*p0*pi*r1^6+lambda1*r2^4+pi*(h1-h0)*G*k*((8/5)*r1^5*L1-
r1^8*M1+2*r1^4*C1+(8/3)*r1^6*B1-(8/5)*r1^5*V1-2*r1^4*S1-(2/9)*T1*r2^6-4*A1*r2^8-
(4/3)*T1*r1^6*log(r1)+(8/3)*d1*r1^6*log(r1)+(4/3)*r2^6*N1-
(8/3)*d1*r2^6*log(r2)+(4/3)*T1*r2^6*log(r2)+4*A1*r1^8-(8/9)*r2^6*d1+r2^8*M1-
(8/5)*r2^5*L1+2*r2^4*S1+(8/5)*r2^5*V1-(8/3)*r2^6*B1-2*r2^4*C1-
(4/3)*r1^6*N1+(8/9)*r1^6*d1+(2/9)*T1*r1^6);
        (1/2)*p0*pi*r2^4-(1/2)*p0*pi*r1^4+lambda1*r2^2+pi*(h1-h0)*G*k*((8/3)*A1*r1^6-2*r1^2*S1-
r1^4*N1-(4/3)*r1^3*V1+(1/4)*T1*r1^4+2*r1^2*C1+(1/2)*r1^4*d1-(2/3)*r1^6*M1-(4/3)*r2^3*L1-
2*B1*r2^4+2*d1*r1^4*log(r1)-T1*r1^4*log(r1)-2*d1*r2^4*log(r2)+2*B1*r1^4+T1*r2^4*log(r2)-
(1/4)*T1*r2^4+(2/3)*r2^6*M1-(1/2)*r2^4*d1-2*r2^2*C1+r2^4*N1+2*r2^2*S1-
(8/3)*A1*r2^6+(4/3)*r2^3*V1+(4/3)*r1^3*L1);
        -(1/2)*p0*pi*r2^2+(1/2)*p0*pi*r1^2+p0*pi*r2^2*log(r2)-
p0*pi*r1^2*log(r1)+lambda1*log(r2)+pi*(h1-h0)*G*k*(-2*V1*r1-N1*r1^2-
(1/2)*M1*r1^4+2*A1*r1^4+2*L1*r1-2*S1*log(r1)+2*B1*r1^2+2*C1*log(r1)-
2*C1*log(r2)+2*d1*r1^2*log(r1)-T1*r1^2*log(r1)-2*d1*r2^2*log(r2)+T1*r2^2*log(r2)+(1/2)*T1*r1^2-
(1/2)*T1*r2^2-2*B1*r2^2+2*S1*log(r2)-2*L1*r2-2*A1*r2^4+2*V1*r2+(1/2)*M1*r2^4+N1*r2^2);
        -(1/8)*p0*pi*r2^4+(1/8)*p0*pi*r1^4+(1/2)*p0*pi*r2^4*log(r2)-
(1/2)*p0*pi*r1^4*log(r1)+lambda1*r2^2*log(r2)+pi*(h1-
h0)*G*k*((2/9)*r2^3*V1+(1/4)*r2^4*N1+(1/4)*r1^4*d1+2*d1*r1^4*log(r1)^2+(8/9)*A1*r1^6-
(2/9)*r2^3*L1+T1*r2^4*log(r2)^2-(4/3)*L1*r2^3*log(r2)+N1*r2^4*log(r2)-
(2/9)*r1^6*M1+d1*r1^4*log(r1)-(4/3)*V1*r1^3*log(r1)+(8/3)*A1*r1^6*log(r1)-d1*r2^4*log(r2)-
```

```
2*C1*r2^2*log(r2)+2*B1*r1^4*log(r1)+(4/3)*V1*r2^3*log(r2)-(8/9)*A1*r2^6-
2*d1*r2^4*log(r2)^2+(2/3)*M1*r2^6*log(r2)-(8/3)*A1*r2^6*log(r2)-2*B1*r2^4*log(r2)-
T1*r1^4*log(r1)^2+2*C1*r1^2*log(r1)-(1/4)*r1^4*N1-2*S1*r1^2*log(r1)-N1*r1^4*log(r1)-
(2/3)*M1*r1^6*log(r1)+2*S1*r2^2*log(r2)+(2/9)*r1^3*L1-
(2/9)*r1^3*V1+(4/3)*L1*r1^3*log(r1)+(2/9)*r2^6*M1-(1/4)*r2^4*d1+(1/2)*B1*r1^4-(1/2)*B1*r2^4);
        (2/3)*p0*pi*r2^3-(2/3)*p0*pi*r1^3+lambda1*r2+pi*(h1-h0)*G*k*(L1*r1^2+(2/9)*d1*r1^3-
r1^2*V1-2*r1*S1-(2/3)*r1^3*N1+(8/5)*A1*r1^5+2*C1*r1-(2/5)*r1^5*M1-(4/3)*B1*r2^3-L1*r2^2-
(2/3)*T1*r1^3*log(r1)+(4/3)*d1*r1^3*log(r1)+(2/3)*T1*r2^3*log(r2)-(4/3)*d1*r2^3*log(r2)-
(2/9)*T1*r2^3-(8/5)*A1*r2^5-2*C1*r2-
(2/9)*d1*r2^3+r2^2*V1+(2/9)*T1*r1^3+(2/5)*r2^5*M1+(2/3)*r2^3*N1+2*r2*S1+(4/3)*B1*r1^3);
        p0*pi*r2^2-p0*pi*r1^2+lambda1;
        (1/3)*p0*pi*r3^6-(1/3)*p0*pi*r2^6+pi*(h2-h0)*G*k*((4/3)*T2*r3^6*log(r3)-
4*A2*r3^8+2*r2^4*C2-(8/5)*r2^5*V2+(8/5)*r3^5*V2+(8/5)*r2^5*L2-r2^8*M2-
2*r2^4*S2+(8/3)*r2^6*B2+(8/9)*r2^6*d2+(2/9)*T2*r2^6-2*r3^4*C2+4*A2*r2^8-
(8/5)*r3^5*L2+r3^8*M2+2*r3^4*S2-(8/3)*r3^6*B2-(8/9)*r3^6*d2-(2/9)*T2*r3^6-(4/3)*r2^6*N2-
(8/3)*d2*r3^6*log(r3)-(4/3)*T2*r2^6*log(r2)+(8/3)*d2*r2^6*log(r2)+(4/3)*r3^6*N2)-
lambda1*r2^4+lambda3*r3^4;
        (1/2)*p0*pi*r3^4-(1/2)*p0*pi*r2^4+pi*(h2-h0)*G*k*(T2*r3^4*log(r3)-2*r3^2*C2-
2*B2*r3^4+2*r2^2*C2+(1/2)*r2^4*d2+(4/3)*r2^3*L2+(8/3)*A2*r2^6+(1/4)*T2*r2^4-(4/3)*r2^3*V2-
2*r2^2*S2-r2^4*N2-(2/3)*r2^6*M2-(1/2)*r3^4*d2-(4/3)*r3^3*L2+2*B2*r2^4-(8/3)*A2*r3^6-
(1/4)*T2*r3^4+(4/3)*r3^3*V2+2*r3^2*S2+r3^4*N2+(2/3)*r3^6*M2-2*d2*r3^4*log(r3)-
T2*r2^4*log(r2)+2*d2*r2^4*log(r2))-lambda1*r2^2+lambda3*r3^2;
        -(1/2)*p0*pi*r3^2+(1/2)*p0*pi*r2^2+p0*pi*r3^2*log(r3)-p0*pi*r2^2*log(r2)+pi*(h2-
h0)*G*k*(-2*B2*r3^2-2*C2*log(r3)+2*B2*r2^2+2*A2*r2^4-2*S2*log(r2)-(1/2)*M2*r2^4+2*L2*r2-2*V2*r2-
N2*r2^2+(1/2)*T2*r2^2-2*A2*r3^4-2*d2*r3^2*log(r3)+2*C2*log(r2)-
T2*r2^2*log(r2)+2*S2*log(r3)+(1/2)*M2*r3^4-2*L2*r3+2*V2*r3+N2*r3^2-
(1/2)*T2*r3^2+T2*r3^2*log(r3)+2*d2*r2^2*log(r2))-lambda1*log(r2)+lambda3*log(r3);
        -(1/8)*p0*pi*r3^4+(1/8)*p0*pi*r2^4+(1/2)*p0*pi*r3^4*log(r3)-
(1/2)*p0*pi*r2^4*log(r2)+pi*(h2-h0)*G*k*((2/9)*r3^6*M2-(2/9)*r2^3*V2+(2/9)*r3^3*V2-
(1/4)*r2^4*N2+2*S2*r3^2*log(r3)+N2*r3^4*log(r3)-(4/3)*L2*r3^3*log(r3)-(8/3)*A2*r3^6*log(r3)-
(2/9)*r3^3*L2-(4/3)*V2*r2^3*log(r2)-(2/3)*M2*r2^6*log(r2)-2*C2*r3^2*log(r3)-
T2*r2^4*log(r2)^2+(8/3)*A2*r2^6*log(r2)-
N2*r2^4*log(r2)+(4/3)*V2*r3^3*log(r3)+(2/3)*M2*r3^6*log(r3)-2*B2*r3^4*log(r3)+(1/4)*r2^4*d2-
(2/9)*r2^6*M2+2*B2*r2^4*log(r2)+2*C2*r2^2*log(r2)-(1/4)*r3^4*d2-
2*S2*r2^2*log(r2)+(4/3)*L2*r2^3*log(r2)+T2*r3^4*log(r3)^2-
(8/9)*A2*r3^6+(2/9)*r2^3*L2+2*d2*r2^4*log(r2)^2+d2*r2^4*log(r2)+(8/9)*A2*r2^6+(1/4)*r3^4*N2+(1/2)
*B2*r2^4-(1/2)*B2*r3^4-d2*r3^4*log(r3)-2*d2*r3^4*log(r3)^2)-
lambda1*r2^2*log(r2)+lambda3*r3^2*log(r3);
        (2/3)*p0*pi*r3^3-(2/3)*p0*pi*r2^3+pi*(h2-h0)*G*k*(-L2*r3^2+(4/3)*B2*r2^3+(8/5)*A2*r2^5-
(2/5)*r2^5*M2-(2/3)*r2^3*N2+(2/9)*T2*r2^3-2*r2*S2-r2^2*V2+2*C2*r2+(2/9)*d2*r2^3-(4/3)*B2*r3^3-
(4/3)*d2*r3^3*log(r3)+L2*r2^2-(8/5)*A2*r3^5+(2/5)*r3^5*M2+(2/3)*r3^3*N2-
(2/9)*T2*r3^3+2*r3*S2+r3^2*V2-(2/3)*T2*r2^3*log(r2)-2*C2*r3-
(2/9)*d2*r3^3+(2/3)*T2*r3^3*log(r3)+(4/3)*d2*r2^3*log(r2))-lambda1*r2+lambda3*r3;
        p0*pi*r3^2-p0*pi*r2^2-lambda1+lambda3;
        -(1/3)*p0*pi*r3^6+(1/3)*p0*pi*l^6+pi*(h3-h0)*G*k*(-2*r3^4*S3-4*A3*a^8-(8/5)*r3^5*V3-
(2/9)*T3*a^6-(4/3)*T3*r3^6*log(r3)+(8/3)*d3*r3^6*log(r3)-(4/3)*r3^6*N3+(4/3)*T3*a^6*log(a)-
(8/3)*d3*a^6*log(a)-(8/3)*a^6*B3-2*a^4*C3-(8/9)*a^6*d3-
(8/5)*a^5*L3+a^8*M3+(4/3)*a^6*N3+2*a^4*S3+(8/5)*a^5*V3-
r3^8*M3+(2/9)*T3*r3^6+(8/3)*r3^6*B3+2*r3^4*C3+(8/9)*r3^6*d3+(8/5)*r3^5*L3+4*A3*r3^8)-
lambda3*r3^4+lambda5*a^4;
        (1/2)*p0*pi*l^4-(1/2)*p0*pi*r3^4+pi*(h3-h0)*G*k*(-
2*B3*a^4+2*r3^2*C3+(1/2)*r3^4*d3+(4/3)*r3^3*L3-(2/3)*r3^6*M3-T3*r3^4*log(r3)-r3^4*N3-2*r3^2*S3-
(4/3)*r3^3*V3+(1/4)*T3*r3^4+2*d3*r3^4*log(r3)+T3*a^4*log(a)-2*d3*a^4*log(a)-(8/3)*A3*a^6-
2*a^2*C3-(1/2)*a^4*d3-(4/3)*a^3*L3+(2/3)*a^6*M3+a^4*N3+2*a^2*S3+(4/3)*a^3*V3-
(1/4)*T3*a^4+(8/3)*A3*r3^6+2*B3*r3^4)-lambda3*r3^2+lambda5*a^2;
        -p0*pi*r3^2*log(r3)-(1/2)*p0*pi*l^2+p0*pi*l^2*log(l)+(1/2)*p0*pi*r3^2+pi*(h3-h0)*G*k*(-
2*C3*log(a)+2*B3*r3^2+2*L3*r3-(1/2)*M3*r3^4-N3*r3^2-2*V3*r3-2*S3*log(r3)-
(1/2)*T3*a^2+2*d3*r3^2*log(r3)-T3*r3^2*log(r3)+T3*a^2*log(a)-2*d3*a^2*log(a)-2*A3*a^4-2*B3*a^2-
2*L3*a+(1/2)*M3*a^4+N3*a^2+2*V3*a+2*S3*log(a)+(1/2)*T3*r3^2+2*A3*r3^4+2*C3*log(r3))-
lambda3*log(r3)+lambda5*log(a);
        -(1/2)*p0*pi*r3^4*log(r3)-(1/8)*p0*pi*l^4+(1/2)*p0*pi*l^4*log(l)+(1/8)*p0*pi*r3^4+pi*(h3-
h0)*G*k*((8/9)*A3*r3^6+(1/4)*r3^4*d3+2*C3*r3^2*log(r3)-(4/3)*V3*r3^3*log(r3)-T3*r3^4*log(r3)^2-
2*S3*r3^2*log(r3)-N3*r3^4*log(r3)+(8/3)*A3*r3^6*log(r3)+(2/3)*M3*a^6*log(a)-(8/3)*A3*a^6*log(a)-
2*B3*a^4*log(a)-2*C3*a^2*log(a)+2*S3*a^2*log(a)+(4/3)*V3*a^3*log(a)+2*B3*r3^4*log(r3)-
(2/3)*M3*r3^6*log(r3)+N3*a^4*log(a)-(4/3)*L3*a^3*log(a)-
(1/4)*r3^4*N3+(4/3)*L3*r3^3*log(r3)+T3*a^4*log(a)^2-(2/9)*r3^3*V3-d3*a^4*log(a)-(2/9)*r3^6*M3-
(2/9)*a^3*L3+2*d3*r3^4*log(r3)^2+(2/9)*a^3*V3-(1/2)*B3*a^4+(1/2)*B3*r3^4+d3*r3^4*log(r3)-
(1/4)*a^4*d3-2*d3*a^4*log(a)^2-(8/9)*A3*a^6+(2/9)*r3^3*L3+(2/9)*a^6*M3+(1/4)*a^4*N3)-
lambda3*r3^2*log(r3)+lambda5*a^2*log(a);
```

```
        (2/3)*p0*pi*l^3-(2/3)*p0*pi*r3^3+pi*(h3-h0)*G*k*(-
L3*a^2+2*C3*r3+(2/9)*d3*r3^3+(4/3)*B3*r3^3-(2/5)*r3^5*M3-(2/3)*r3^3*N3+(2/9)*T3*r3^3-
(2/3)*T3*r3^3*log(r3)+(2/3)*T3*a^3*log(a)-(4/3)*d3*a^3*log(a)+2*a*S3+a^2*V3-(8/5)*A3*a^5-2*C3*a-
(2/9)*d3*a^3-(4/3)*B3*a^3+(2/5)*a^5*M3+(2/3)*a^3*N3-(2/9)*T3*a^3-2*r3*S3-
r3^2*V3+(8/5)*A3*r3^5+L3*r3^2+(4/3)*d3*r3^3*log(r3))-lambda3*r3+lambda5*a;
        p0*pi*l^2-p0*pi*r3^2-lambda3+lambda5;
        (1/72)*pi*(h1-h0)^3*E*(-
12*nu*N1*r2^6*r1^2+12*nu*N1*r1^6*r2^2+12*nu*S1*r1^4*r2^2+12*nu*V1*r1^5*r2^2-
12*T1*r2^6*log(r2)*r1^2+12*T1*r1^6*log(r1)*r2^2+12*N1*r1^6*r2^2-12*S1*r1^4*r2^2-
12*nu*M1*r2^8*r1^2+12*S1*r2^4*r1^2-6*T1*r2^6*r1^2-4*V1*r2^5*r1^2-
12*N1*r2^6*r1^2+12*nu*M1*r1^8*r2^2+4*V1*r1^5*r2^2+6*T1*r1^6*r2^2-12*nu*V1*r2^5*r1^2-
12*nu*S1*r2^4*r1^2+20*M1*r1^8*r2^2-20*M1*r2^8*r1^2+12*nu*T1*r1^6*log(r1)*r2^2-
12*nu*T1*r2^6*log(r2)*r1^2)/((1-nu^2)*r2^2*r1^2)+lambda2*r2^3+pi*(h1-h0)*G*k*((1/18)*T1*r2^6-
(1/2)*C1*r1^4-(2/9)*d1*r1^6-A1*r1^8-(2/3)*B1*r1^6+(1/2)*r1^4*S1+(2/5)*r1^5*V1-
(2/5)*L1*r1^5+(1/3)*r1^6*N1-(1/4)*M1*r2^8+(1/4)*M1*r1^8+(1/3)*T1*r1^6*log(r1)-
(2/3)*d1*r1^6*log(r1)-(1/3)*T1*r2^6*log(r2)+(2/3)*d1*r2^6*log(r2)+A1*r2^8-(2/5)*r2^5*V1-
(1/2)*r2^4*S1+(2/3)*B1*r2^6+(1/2)*C1*r2^4+(2/9)*d1*r2^6-(1/18)*T1*r1^6-
(1/3)*r2^6*N1+(2/5)*L1*r2^5);
        (1/72)*pi*(h1-h0)^3*E*(-12*nu*M1*r2^6*r1^2+12*nu*M1*r1^6*r2^2+12*nu*V1*r1^3*r2^2-
12*nu*V1*r2^3*r1^2-12*T1*r2^4*log(r2)*r1^2+12*T1*r1^4*log(r1)*r2^2+12*M1*r1^6*r2^2-
12*nu*N1*r2^4*r1^2-12*V1*r2^3*r1^2-
12*M1*r2^6*r1^2+12*V1*r1^3*r2^2+12*nu*N1*r1^4*r2^2+12*N1*r1^4*r2^2-
12*N1*r2^4*r1^2+12*nu*T1*r1^4*log(r1)*r2^2-12*nu*T1*r2^4*log(r2)*r1^2)/((1-
nu^2)*r2^2*r1^2)+lambda2*r2+pi*(h1-h0)*G*k*(-C1*r1^2-(1/4)*d1*r1^4-(2/3)*L1*r1^3+(2/3)*r1^3*V1-
B1*r1^4+r1^2*S1+(1/3)*M1*r1^6-(1/2)*N1*r2^4+(4/3)*A1*r2^6-
d1*r1^4*log(r1)+(1/2)*T1*r1^4*log(r1)+d1*r2^4*log(r2)+(1/8)*T1*r2^4-
(1/2)*T1*r2^4*log(r2)+(1/2)*N1*r1^4-(2/3)*r2^3*V1-r2^2*S1+B1*r2^4+C1*r2^2+(1/4)*d1*r2^4-
(4/3)*A1*r1^6-(1/8)*T1*r1^4-(1/3)*M1*r2^6+(2/3)*L1*r2^3);
        (1/72)*pi*(h1-h0)^3*E*(12*V1*r2*r1^2+12*nu*M1*r1^4*r2^2-12*nu*V1*r2*r1^2+12*nu*S1*r2^2-
12*V1*r2^2*r1-12*nu*S1*r1^2+12*nu*V1*r2^2*r1+12*nu*T1*log(r1)*r2^2*r1^2-
12*nu*T1*log(r2)*r2^2*r1^2+12*T1*log(r2)*r2^2*r1^2-12*M1*r1^4*r2^2+12*M1*r2^4*r1^2+12*r1^2*S1-
12*r2^2*S1-12*nu*M1*r2^4*r1^2-12*T1*log(r1)*r2^2*r1^2)/((1-nu^2)*r2^2*r1^2)+lambda2/r2+pi*(h1-
h0)*G*k*(-2*L1*r1+2*V1*r1-2*B1*r1^2-2*C1*log(r1)-2*A1*r1^4+(1/2)*M1*r1^4+N1*r1^2-(1/2)*T1*r1^2-
2*S1*log(r2)-2*d1*r1^2*log(r1)+T1*r1^2*log(r1)+2*d1*r2^2*log(r2)-T1*r2^2*log(r2)+2*S1*log(r1)-
N1*r2^2-(1/2)*M1*r2^4+2*A1*r2^4+2*C1*log(r2)+2*B1*r2^2-2*V1*r2+(1/2)*T1*r2^2+2*L1*r2);
        (1/72)*pi*(h1-h0)^3*E*(-12*V1*r2^3*log(r2)*r1^2+12*V1*r1^3*log(r1)*r2^2-
12*N1*r2^4*log(r2)*r1^2+6*T1*r1^4*r2^2+6*M1*r1^6*r2^2+12*V1*r2^3*r1^2-6*M1*r2^6*r1^2-
12*V1*r1^3*r2^2-12*M1*r2^6*log(r2)*r1^2-
12*nu*T1*r2^4*log(r2)^2*r1^2+12*nu*T1*r1^4*log(r1)^2*r2^2-
12*nu*S1*log(r2)*r2^2*r1^2+12*M1*r1^6*log(r1)*r2^2+12*N1*r1^4*log(r1)*r2^2+12*S1*log(r2)*r2^2*r1^
2-
12*S1*log(r1)*r2^2*r1^2+12*nu*M1*r1^6*log(r1)*r2^2+12*nu*V1*r1^3*log(r1)*r2^2+12*nu*N1*r1^4*log(r
1)*r2^2-12*nu*M1*r2^6*log(r2)*r1^2-12*nu*V1*r2^3*log(r2)*r1^2-12*nu*N1*r2^4*log(r2)*r1^2-
6*T1*r2^4*r1^2+12*nu*S1*log(r1)*r2^2*r1^2-
12*T1*r2^4*log(r2)^2*r1^2+12*T1*r1^4*log(r1)^2*r2^2)/((1-
nu^2)*r2^2*r1^2)+lambda2*r2*log(r2)+pi*(h1-h0)*G*k*((1/16)*T1*r1^4+(1/2)*T1*r1^4*log(r1)^2-
(1/4)*T1*r1^4*log(r1)+(1/4)*T1*r2^4*log(r2)+(1/8)*N1*r2^4-(1/8)*N1*r1^4-
(1/18)*M1*r1^6+(1/2)*C1*r1^2-(2/9)*L1*r2^3+(1/4)*B1*r1^4-(1/2)*T1*r2^4*log(r2)^2-
(1/2)*r1^2*S1+(1/3)*M1*r1^6*log(r1)-(2/9)*r1^3*V1-(1/4)*B1*r2^4-
C1*r1^2*log(r1)+(1/2)*N1*r1^4*log(r1)+S1*r1^2*log(r1)-B1*r1^4*log(r1)+(2/3)*V1*r1^3*log(r1)-
(2/3)*L1*r1^3*log(r1)-(4/3)*A1*r1^6*log(r1)-d1*r1^4*log(r1)^2+(4/3)*A1*r2^6*log(r2)-
(1/3)*M1*r2^6*log(r2)-(2/9)*A1*r2^6+(1/2)*r2^2*S1+(2/9)*r2^3*V1-
(2/3)*V1*r2^3*log(r2)+(2/3)*L1*r2^3*log(r2)-S1*r2^2*log(r2)-
(1/2)*N1*r2^4*log(r2)+d1*r2^4*log(r2)^2+B1*r2^4*log(r2)+C1*r2^2*log(r2)-(1/16)*T1*r2^4-
(1/2)*C1*r2^2+(1/18)*M1*r2^6+(2/9)*L1*r1^3+(2/9)*A1*r1^6);
        (1/72)*pi*(h1-h0)^3*E*(12*S1*r2*r1^2+12*nu*N1*r1^3*r2^2-
12*nu*N1*r2^3*r1^2+12*nu*M1*r1^5*r2^2-12*nu*S1*r2*r1^2-
12*S1*r2^2*r1+12*nu*S1*r2^2*r1+12*T1*r1^3*log(r1)*r2^2-12*T1*r2^3*log(r2)*r1^2+12*T1*r2^3*r1^2-
12*N1*r2^3*r1^2-4*M1*r2^5*r1^2+12*N1*r1^3*r2^2-
12*T1*r1^3*r2^2+4*M1*r1^5*r2^2+12*V1*log(r1)*r2^2*r1^2-12*V1*log(r2)*r2^2*r1^2-
12*nu*M1*r2^5*r1^2+12*nu*T1*r1^3*log(r1)*r2^2-12*nu*T1*r2^3*log(r2)*r1^2)/((1-
nu^2)*r2^2*r1^2)+lambda2+pi*(h1-h0)*G*k*(-2*C1*r1-(2/9)*d1*r1^3-L1*r1^2+(2/3)*N1*r1^3+2*S1*r1-
(4/3)*B1*r1^3-(8/5)*A1*r1^5+(2/5)*M1*r1^5+(2/9)*T1*r2^3-V1*r2^2-
(4/3)*d1*r1^3*log(r1)+(2/3)*T1*r1^3*log(r1)+(4/3)*d1*r2^3*log(r2)-(2/3)*T1*r2^3*log(r2)+V1*r1^2-
(2/3)*N1*r2^3-(2/5)*M1*r2^5+(8/5)*A1*r2^5-2*S1*r2+2*C1*r2+(4/3)*B1*r2^3+L1*r2^2+(2/9)*d1*r2^3-
(2/9)*T1*r1^3);
        (1/72)*pi*(h2-h0)^3*E*(20*M2*r2^8*r3^2-20*M2*r3^8*r2^2-12*N2*r3^6*r2^2-
12*nu*M2*r3^8*r2^2+12*S2*r3^4*r2^2-4*V2*r3^5*r2^2-6*T2*r3^6*r2^2-
12*S2*r2^4*r3^2+6*T2*r2^6*r3^2+4*V2*r2^5*r3^2+12*N2*r2^6*r3^2+12*nu*M2*r2^8*r3^2-
```

```
12*nu*V2*r3^5*r2^2-12*nu*N2*r3^6*r2^2+12*nu*N2*r2^6*r3^2+12*nu*V2*r2^5*r3^2-
12*nu*S2*r3^4*r2^2+12*nu*S2*r2^4*r3^2-12*T2*r3^6*log(r3)*r2^2+12*T2*r2^6*log(r2)*r3^2-
12*nu*T2*r3^6*log(r3)*r2^2+12*nu*T2*r2^6*log(r2)*r3^2)/((1-nu^2)*r2^2*r3^2)+pi*(h2-
h0)*G*k*((1/4)*M2*r2^8-(1/3)*T2*r3^6*log(r3)-(1/4)*M2*r3^8-A2*r2^8-(2/5)*L2*r2^5-
(1/18)*T2*r2^6+(1/3)*r2^6*N2+(1/2)*r2^4*S2+(2/5)*r2^5*V2-(1/2)*C2*r2^4-(2/3)*B2*r2^6-
(2/9)*d2*r2^6+A2*r3^8+(2/5)*L2*r3^5+(1/18)*T2*r3^6-(1/3)*r3^6*N2-(2/3)*d2*r2^6*log(r2)-
(1/2)*r3^4*S2-
(2/5)*r3^5*V2+(1/2)*C2*r3^4+(2/3)*B2*r3^6+(2/9)*d2*r3^6+(2/3)*d2*r3^6*log(r3)+(1/3)*T2*r2^6*log(r
2))-lambda2*r2^3+lambda4*r3^3;
        (1/72)*pi*(h2-h0)^3*E*(12*N2*r2^4*r3^2-12*N2*r3^4*r2^2-12*V2*r3^3*r2^2-12*M2*r3^6*r2^2-
12*nu*N2*r3^4*r2^2+12*nu*N2*r2^4*r3^2+12*V2*r2^3*r3^2+12*M2*r2^6*r3^2-12*nu*M2*r3^6*r2^2-
12*nu*V2*r3^3*r2^2+12*nu*M2*r2^6*r3^2+12*nu*V2*r2^3*r3^2-12*T2*r3^4*log(r3)*r2^2-
12*nu*T2*r3^4*log(r3)*r2^2+12*nu*T2*r2^4*log(r2)*r3^2+12*T2*r2^4*log(r2)*r3^2)/((1-
nu^2)*r2^2*r3^2)+pi*(h2-h0)*G*k*(-(1/2)*N2*r3^4+d2*r3^4*log(r3)-(1/8)*T2*r2^4-
(2/3)*L2*r2^3+(1/3)*M2*r2^6-(1/2)*T2*r3^4*log(r3)+r2^2*S2+(2/3)*r2^3*V2-B2*r2^4-C2*r2^2-
(1/4)*d2*r2^4+(1/2)*N2*r2^4+(1/8)*T2*r3^4+(2/3)*L2*r3^3-(1/3)*M2*r3^6-r3^2*S2-
(2/3)*r3^3*V2+B2*r3^4+C2*r3^2+(1/4)*d2*r3^4-(4/3)*A2*r2^6-
d2*r2^4*log(r2)+(1/2)*T2*r2^4*log(r2)+(4/3)*A2*r3^6)-lambda2*r2+lambda4*r3;
        (1/72)*pi*(h2-h0)^3*E*(12*M2*r3^4*r2^2-12*M2*r2^4*r3^2-12*r3^2*S2+12*r2^2*S2-
12*nu*M2*r3^4*r2^2+12*nu*M2*r2^4*r3^2-12*V2*r2*r3^2+12*nu*S2*r3^2+12*nu*V2*r2*r3^2-
12*nu*V2*r2^2*r3-12*nu*S2*r2^2+12*V2*r2^2*r3+12*nu*T2*log(r2)*r2^2*r3^2+12*T2*log(r3)*r2^2*r3^2-
12*T2*log(r2)*r2^2*r3^2-12*nu*T2*log(r3)*r2^2*r3^2)/((1-nu^2)*r2^2*r3^2)+pi*(h2-h0)*G*k*(-
2*S2*log(r3)-2*A2*r2^4+2*V2*r2-(1/2)*T2*r2^2-2*L2*r2+(1/2)*M2*r2^4+N2*r2^2-2*C2*log(r2)-
2*B2*r2^2+2*d2*r3^2*log(r3)+2*S2*log(r2)+2*A2*r3^4-2*V2*r3+(1/2)*T2*r3^2+2*L2*r3+T2*r2^2*log(r2)-
(1/2)*M2*r3^4-N2*r3^2+2*C2*log(r3)+2*B2*r3^2-T2*r3^2*log(r3)-2*d2*r2^2*log(r2))-
lambda2/r2+lambda4/r3;
        (1/72)*pi*(h2-h0)^3*E*(-6*T2*r3^4*r2^2-12*nu*V2*r3^3*log(r3)*r2^2-
12*nu*N2*r3^4*log(r3)*r2^2-12*nu*S2*log(r3)*r2^2*r3^2-
12*nu*M2*r3^6*log(r3)*r2^2+12*nu*N2*r2^4*log(r2)*r3^2+12*nu*V2*r2^3*log(r2)*r3^2+12*nu*M2*r2^6*lo
g(r2)*r3^2+6*T2*r2^4*r3^2-12*T2*r3^4*log(r3)^2*r2^2+12*T2*r2^4*log(r2)^2*r3^2-
12*V2*r2^3*r3^2+12*nu*S2*log(r2)*r2^2*r3^2-12*V2*r3^3*log(r3)*r2^2-12*nu*T2*r3^4*log(r3)^2*r2^2-
12*N2*r3^4*log(r3)*r2^2-
12*M2*r3^6*log(r3)*r2^2+12*nu*T2*r2^4*log(r2)^2*r3^2+6*M2*r2^6*r3^2+12*N2*r2^4*log(r2)*r3^2+12*M2
*r2^6*log(r2)*r3^2+12*V2*r2^3*log(r2)*r3^2-12*S2*log(r2)*r2^2*r3^2-
6*M2*r3^6*r2^2+12*V2*r3^3*r2^2+12*S2*log(r3)*r2^2*r3^2)/((1-nu^2)*r2^2*r3^2)+pi*(h2-
h0)*G*k*((1/2)*C2*r2^2-(1/2)*r2^2*S2-(1/2)*T2*r3^4*log(r3)^2-(2/9)*r2^3*V2-(2/9)*A2*r3^6-
(1/8)*N2*r2^4+(1/8)*N2*r3^4-(1/2)*N2*r3^4*log(r3)-
(1/3)*M2*r3^6*log(r3)+(4/3)*A2*r3^6*log(r3)+B2*r3^4*log(r3)-S2*r3^2*log(r3)-
(2/3)*L2*r2^3*log(r2)+S2*r2^2*log(r2)-B2*r2^4*log(r2)+(1/2)*N2*r2^4*log(r2)-
d2*r2^4*log(r2)^2+(1/3)*M2*r2^6*log(r2)-(4/3)*A2*r2^6*log(r2)-
(2/3)*V2*r3^3*log(r3)+C2*r3^2*log(r3)+(2/3)*L2*r3^3*log(r3)+d2*r3^4*log(r3)^2+(1/2)*T2*r2^4*log(r
2)^2-(1/4)*T2*r2^4*log(r2)+(1/4)*T2*r3^4*log(r3)-(1/4)*B2*r3^4+(1/4)*B2*r2^4-(1/16)*T2*r3^4-
(2/9)*L2*r3^3+(1/2)*r3^2*S2-(1/18)*M2*r2^6+(2/9)*L2*r2^3+(2/9)*r3^3*V2-
(1/2)*C2*r3^2+(1/18)*M2*r3^6+(2/9)*A2*r2^6+(1/16)*T2*r2^4+(2/3)*V2*r2^3*log(r2)-C2*r2^2*log(r2))-
lambda2*r2*log(r2)+lambda4*r3*log(r3);
        (1/72)*pi*(h2-h0)^3*E*(-12*T2*r2^3*r3^2-12*N2*r3^3*r2^2+12*T2*r3^3*r2^2-
4*M2*r3^5*r2^2+4*M2*r2^5*r3^2+12*N2*r2^3*r3^2-12*nu*M2*r3^5*r2^2-
12*nu*N2*r3^3*r2^2+12*nu*M2*r2^5*r3^2+12*nu*N2*r2^3*r3^2-
12*V2*log(r3)*r2^2*r3^2+12*V2*log(r2)*r2^2*r3^2-12*S2*r2*r3^2+12*nu*S2*r2*r3^2-
12*nu*S2*r2^2*r3+12*S2*r2^2*r3-12*T2*r3^3*log(r3)*r2^2+12*T2*r2^3*log(r2)*r3^2-
12*nu*T2*r3^3*log(r3)*r2^2+12*nu*T2*r2^3*log(r2)*r3^2)/((1-nu^2)*r2^2*r3^2)+pi*(h2-h0)*G*k*(-
V2*r3^2-(8/5)*A2*r2^5+(8/5)*A2*r3^5+2*S2*r2-(2/9)*T2*r2^3-L2*r2^2+(2/5)*M2*r2^5+(2/3)*N2*r2^3-
(4/3)*B2*r2^3-2*C2*r2-(2/9)*d2*r2^3+V2*r2^2-2*S2*r3+(2/9)*T2*r3^3+L2*r3^2-
(4/3)*d2*r2^3*log(r2)+(2/3)*T2*r2^3*log(r2)-(2/5)*M2*r3^5-
(2/3)*N2*r3^3+(4/3)*B2*r3^3+2*C2*r3+(2/9)*d2*r3^3+(4/3)*d2*r3^3*log(r3)-(2/3)*T2*r3^3*log(r3))-
lambda2+lambda4;
        (1/72)*pi*(h3-h0)^3*E*(20*M3*r3^8*a^2-20*M3*a^8*r3^2-4*V3*a^5*r3^2+12*S3*a^4*r3^2-
12*nu*M3*a^8*r3^2+12*nu*M3*r3^8*a^2-12*N3*a^6*r3^2-6*T3*a^6*r3^2-12*nu*N3*a^6*r3^2-
12*nu*V3*a^5*r3^2-
12*S3*r3^4*a^2+4*V3*r3^5*a^2+12*N3*r3^6*a^2+6*T3*r3^6*a^2+12*nu*V3*r3^5*a^2+12*nu*S3*r3^4*a^2+12*
nu*N3*r3^6*a^2-12*nu*S3*a^4*r3^2+12*nu*T3*r3^6*log(r3)*a^2-
12*nu*T3*a^6*log(a)*r3^2+12*T3*r3^6*log(r3)*a^2-12*T3*a^6*log(a)*r3^2)/((1-
nu^2)*a^2*r3^2)+pi*(h3-h0)*G*k*(-(1/4)*M3*a^8-(1/2)*C3*r3^4-(2/9)*d3*r3^6-(2/3)*B3*r3^6-
(2/3)*d3*r3^6*log(r3)-(1/18)*T3*r3^6-(2/5)*L3*r3^5-(2/5)*a^5*V3-
(1/3)*T3*a^6*log(a)+(2/3)*d3*a^6*log(a)-(1/2)*a^4*S3-
(1/3)*a^6*N3+A3*a^8+(1/2)*C3*a^4+(2/9)*d3*a^6+(2/3)*B3*a^6+(1/18)*T3*a^6+(2/5)*L3*a^5-
A3*r3^8+(2/5)*r3^5*V3+(1/2)*r3^4*S3+(1/3)*r3^6*N3+(1/3)*T3*r3^6*log(r3)+(1/4)*M3*r3^8)-
lambda4*r3^3+lambda6*a^3;
```

```
        (1/72)*pi*(h3-h0)^3*E*(12*N3*r3^4*a^2-12*N3*a^4*r3^2-12*V3*a^3*r3^2-
12*nu*N3*a^4*r3^2+12*nu*N3*r3^4*a^2-12*M3*a^6*r3^2-
12*nu*M3*a^6*r3^2+12*V3*r3^3*a^2+12*M3*r3^6*a^2+12*nu*V3*r3^3*a^2+12*nu*M3*r3^6*a^2-
12*nu*V3*a^3*r3^2-12*T3*a^4*log(a)*r3^2+12*nu*T3*r3^4*log(r3)*a^2-
12*nu*T3*a^4*log(a)*r3^2+12*T3*r3^4*log(r3)*a^2)/((1-nu^2)*a^2*r3^2)+pi*(h3-h0)*G*k*(-
(1/2)*N3*a^4-C3*r3^2-(1/4)*d3*r3^4-d3*r3^4*log(r3)-B3*r3^4-(2/3)*L3*r3^3+r3^2*S3+(2/3)*r3^3*V3-
(1/8)*T3*r3^4-(4/3)*A3*r3^6+(1/2)*T3*r3^4*log(r3)-(1/2)*T3*a^4*log(a)+d3*a^4*log(a)-
(1/3)*M3*a^6+(4/3)*A3*a^6+C3*a^2+(1/4)*d3*a^4+B3*a^4+(2/3)*L3*a^3-a^2*S3-
(2/3)*a^3*V3+(1/8)*T3*a^4+(1/3)*M3*r3^6+(1/2)*N3*r3^4)-lambda4*r3+lambda6*a;
        (1/72)*pi*(h3-h0)^3*E*(-12*nu*S3*r3^2-12*nu*V3*a*r3^2+12*V3*a*r3^2+12*nu*V3*a^2*r3-
12*nu*T3*log(a)*a^2*r3^2-12*V3*a^2*r3+12*nu*S3*a^2+12*M3*a^4*r3^2-12*M3*r3^4*a^2+12*r3^2*S3-
12*a^2*S3+12*nu*M3*r3^4*a^2+12*T3*log(a)*a^2*r3^2-12*nu*M3*a^4*r3^2-
12*T3*log(r3)*a^2*r3^2+12*nu*T3*log(r3)*a^2*r3^2)/((1-nu^2)*a^2*r3^2)+pi*(h3-h0)*G*k*(-2*A3*r3^4-
2*S3*log(a)-2*C3*log(r3)-2*d3*r3^2*log(r3)+T3*r3^2*log(r3)-2*B3*r3^2+N3*r3^2+2*V3*r3-
T3*a^2*log(a)+2*d3*a^2*log(a)+(1/2)*T3*a^2-(1/2)*M3*a^4+2*L3*a+2*A3*a^4+2*C3*log(a)+2*B3*a^2-
N3*a^2-2*V3*a-(1/2)*T3*r3^2+(1/2)*M3*r3^4-2*L3*r3+2*S3*log(r3))-lambda4/r3+lambda6/a;
        (1/72)*pi*(h3-h0)^3*E*(-6*T3*a^4*r3^2+6*T3*r3^4*a^2+12*V3*a^3*r3^2-6*M3*a^6*r3^2-
12*V3*r3^3*a^2+6*M3*r3^6*a^2+12*S3*log(a)*a^2*r3^2+12*nu*T3*r3^4*log(r3)^2*a^2-
12*nu*S3*log(a)*a^2*r3^2-
12*N3*a^4*log(a)*r3^2+12*nu*M3*r3^6*log(r3)*a^2+12*nu*N3*r3^4*log(r3)*a^2-
12*nu*V3*a^3*log(a)*r3^2-12*nu*M3*a^6*log(a)*r3^2-
12*nu*N3*a^4*log(a)*r3^2+12*N3*r3^4*log(r3)*a^2+12*V3*r3^3*log(r3)*a^2+12*nu*V3*r3^3*log(r3)*a^2-
12*S3*log(r3)*a^2*r3^2-12*V3*a^3*log(a)*r3^2+12*M3*r3^6*log(r3)*a^2-12*nu*T3*a^4*log(a)^2*r3^2-
12*M3*a^6*log(a)*r3^2+12*nu*S3*log(r3)*a^2*r3^2-
12*T3*a^4*log(a)^2*r3^2+12*T3*r3^4*log(r3)^2*a^2)/((1-nu^2)*a^2*r3^2)+pi*(h3-h0)*G*k*(-
(1/16)*T3*a^4+(1/4)*B3*r3^4+S3*r3^2*log(r3)-d3*r3^4*log(r3)^2-
(1/4)*T3*r3^4*log(r3)+(1/4)*T3*a^4*log(a)-(1/2)*T3*a^4*log(a)^2+(1/8)*N3*a^4-(1/8)*N3*r3^4-
(2/9)*A3*a^6+(2/9)*A3*r3^6+(2/9)*a^3*V3-(1/4)*B3*a^4-B3*r3^4*log(r3)-
C3*r3^2*log(r3)+(1/2)*N3*r3^4*log(r3)+(2/3)*V3*r3^3*log(r3)-
(2/3)*L3*r3^3*log(r3)+(2/3)*L3*a^3*log(a)-(1/3)*M3*a^6*log(a)-(1/2)*N3*a^4*log(a)-
S3*a^2*log(a)+B3*a^4*log(a)+C3*a^2*log(a)+(4/3)*A3*a^6*log(a)-
(2/3)*V3*a^3*log(a)+d3*a^4*log(a)^2+(1/3)*M3*r3^6*log(r3)+(1/2)*T3*r3^4*log(r3)^2+(1/2)*C3*r3^2+(
1/2)*a^2*S3+(2/9)*L3*r3^3-(1/2)*C3*a^2-(1/18)*M3*r3^6-(2/9)*L3*a^3+(1/16)*T3*r3^4-(1/2)*r3^2*S3-
(2/9)*r3^3*V3-(4/3)*A3*r3^6*log(r3)+(1/18)*M3*a^6)-lambda4*r3*log(r3)+lambda6*a*log(a);
        (1/72)*pi*(h3-h0)^3*E*(-12*nu*S3*a*r3^2+12*S3*a*r3^2+12*nu*S3*a^2*r3-
12*S3*a^2*r3+12*T3*a^3*r3^2-12*N3*a^3*r3^2-4*M3*a^5*r3^2-
12*nu*M3*a^5*r3^2+4*M3*r3^5*a^2+12*N3*r3^3*a^2-12*T3*r3^3*a^2+12*V3*log(r3)*a^2*r3^2-
12*V3*log(a)*a^2*r3^2+12*nu*M3*r3^5*a^2+12*nu*N3*r3^3*a^2-12*nu*N3*a^3*r3^2-
12*nu*T3*a^3*log(a)*r3^2+12*nu*T3*r3^3*log(r3)*a^2-
12*T3*a^3*log(a)*r3^2+12*T3*r3^3*log(r3)*a^2)/((1-nu^2)*a^2*r3^2)+pi*(h3-h0)*G*k*(-V3*a^2-
(8/5)*A3*r3^5-2*C3*r3-(4/3)*d3*r3^3*log(r3)+(2/3)*T3*r3^3*log(r3)-(2/9)*T3*r3^3-(4/3)*B3*r3^3-
(2/9)*d3*r3^3+(2/3)*N3*r3^3+2*S3*r3-(2/5)*M3*a^5+(4/3)*d3*a^3*log(a)-
(2/3)*T3*a^3*log(a)+L3*a^2+(8/5)*A3*a^5+2*C3*a+(2/9)*T3*a^3+(4/3)*B3*a^3+(2/9)*d3*a^3-
(2/3)*N3*a^3-2*S3*a+(2/5)*M3*r3^5-L3*r3^2+V3*r3^2)-lambda4+lambda6;
        A1*r2^4+B1*r2^2+C1*log(r2)+d1*r2^2*log(r2)+L1*r2+H1-A2*r2^4-B2*r2^2-C2*log(r2)-
d2*r2^2*log(r2)-L2*r2-H2;
        M1*r2^3+N1*r2+S1/r2+T1*r2*log(r2)+V1-M2*r2^3-N2*r2-S2/r2-T2*r2*log(r2)-V2;
        A2*r3^4+B2*r3^2+C2*log(r3)+d2*r3^2*log(r3)+L2*r3+H2-A3*r3^4-B3*r3^2-C3*log(r3)-
d3*r3^2*log(r3)-L3*r3-H3;
        M2*r3^3+N2*r3+S2/r3+T2*r3*log(r3)+V2-M3*r3^3-N3*r3-S3/r3-T3*r3*log(r3)-V3;
        A3*a^4+B3*a^2+C3*log(a)+d3*a^2*log(a)+L3*a+H3;
        M3*a^3+N3*a+S3/a+T3*a*log(a)+V3];
end
```

Appendix F. The MATLAB Function for the Shim Stack Assembly Deflection

```
function FF=caseB_30sep09_final_for_hyd_mod(x)
% x is a vector such that x(1)=A1, x(2)=B1, x(3)=C1, ... in the raylegh ritz method
global  r1 r2 r3 a l h0 h1 h2 h3 E nu p0 k G
A1=x(1);
B1=x(2);
C1=x(3);
d1=x(4);
L1=x(5);
H1=x(6);
A2=x(7);
B2=x(8);
C2=x(9);
d2=x(10);
L2=x(11);
H2=x(12);
A3=x(13);
B3=x(14);
C3=x(15);
d3=x(16);
L3=x(17);
H3=x(18);
M1=x(19);
N1=x(20);
S1=x(21);
T1=x(22);
V1=x(23);
M2=x(24);
N2=x(25);
S2=x(26);
T2=x(27);
V2=x(28);
M3=x(29);
N3=x(30);
S3=x(31);
T3=x(32);
V3=x(33);
lambda1=x(34);
lambda2=x(35);
lambda3=x(36);
lambda4=x(37);
lambda5=x(38);
lambda6=x(39);
if l>=r2
    FF=[(1/3)*p0*pi*l^6-(1/3)*p0*pi*r1^6+pi*(h1-h0)*G*k*((4/3)*T1*r2^6*log(r2)-
(8/3)*d1*r2^6*log(r2)-(4/3)*T1*r1^6*log(r1)+(8/3)*d1*r1^6*log(r1)-4*A1*r2^8-
(2/9)*T1*r2^6+r2^8*M1-(8/5)*r2^5*L1-(8/9)*r2^6*d1-(8/3)*r2^6*B1-
2*r2^4*C1+(8/5)*r2^5*V1+2*r2^4*S1+(4/3)*r2^6*N1+(8/9)*r1^6*d1+(8/3)*r1^6*B1+2*r1^4*C1+(2/9)*T1*r1
^6-(8/5)*r1^5*V1-2*r1^4*S1-(4/3)*r1^6*N1-r1^8*M1+(8/5)*r1^5*L1+4*A1*r1^8)+lambda1*r2^4;
        (1/2)*p0*pi*l^4-(1/2)*p0*pi*r1^4+pi*(h1-h0)*G*k*(T1*r2^4*log(r2)-2*d1*r2^4*log(r2)-
T1*r1^4*log(r1)+2*d1*r1^4*log(r1)-2*B1*r2^4-(8/3)*A1*r2^6-(1/4)*T1*r2^4+r2^4*N1+(2/3)*r2^6*M1-
(4/3)*r2^3*L1-(1/2)*r2^4*d1-2*r2^2*C1+(4/3)*r2^3*V1+2*r2^2*S1+(8/3)*A1*r1^6+(1/4)*T1*r1^4-
(2/3)*r1^6*M1+(4/3)*r1^3*L1+(1/2)*r1^4*d1+2*r1^2*C1-(4/3)*r1^3*V1-2*r1^2*S1-
r1^4*N1+2*B1*r1^4)+lambda1*r2^2;
        -(1/2)*p0*pi*l^2+(1/2)*p0*pi*r1^2+p0*pi*l^2*log(l)-p0*pi*r1^2*log(r1)+pi*(h1-
h0)*G*k*(T1*r2^2*log(r2)-2*d1*r2^2*log(r2)-T1*r1^2*log(r1)+2*d1*r1^2*log(r1)-2*C1*log(r2)-
(1/2)*T1*r2^2-2*A1*r2^4-2*B1*r2^2+2*V1*r2+N1*r2^2+(1/2)*M1*r2^4-
2*L1*r2+2*S1*log(r2)+(1/2)*T1*r1^2+2*A1*r1^4+2*B1*r1^2-2*S1*log(r1)-2*V1*r1-N1*r1^2-
(1/2)*M1*r1^4+2*L1*r1+2*C1*log(r1))+lambda1*log(r2);
        -(1/8)*p0*pi*l^4+(1/8)*p0*pi*r1^4+(1/2)*p0*pi*l^4*log(l)-(1/2)*p0*pi*r1^4*log(r1)+pi*(h1-
h0)*G*k*(d1*r1^4*log(r1)-
d1*r2^4*log(r2)+2*S1*r2^2*log(r2)+N1*r2^4*log(r2)+T1*r2^4*log(r2)^2+(2/9)*r2^6*M1+(2/3)*M1*r2^6*l
og(r2)-(4/3)*L1*r2^3*log(r2)+(4/3)*V1*r2^3*log(r2)-N1*r1^4*log(r1)-
(2/3)*M1*r1^6*log(r1)+2*C1*r1^2*log(r1)+(4/3)*L1*r1^3*log(r1)-
(4/3)*V1*r1^3*log(r1)+2*B1*r1^4*log(r1)+(8/3)*A1*r1^6*log(r1)-2*S1*r1^2*log(r1)-
```

```
2*C1*r2^2*log(r2)-(1/2)*B1*r2^4+(1/2)*B1*r1^4+(1/4)*r2^4*N1-T1*r1^4*log(r1)^2-(8/9)*A1*r2^6-
2*d1*r2^4*log(r2)^2+2*d1*r1^4*log(r1)^2-(8/3)*A1*r2^6*log(r2)-(2/9)*r2^3*L1-
(1/4)*r2^4*d1+(2/9)*r2^3*V1+(8/9)*A1*r1^6+(2/9)*r1^3*L1-(2/9)*r1^6*M1+(1/4)*r1^4*d1-
(1/4)*r1^4*N1-(2/9)*r1^3*V1-2*B1*r2^4*log(r2))+lambda1*r2^2*log(r2);
        (2/3)*p0*pi*l^3-(2/3)*p0*pi*r1^3+pi*(h1-h0)*G*k*(-
(4/3)*d1*r2^3*log(r2)+(4/3)*d1*r1^3*log(r1)+(2/3)*T1*r2^3*log(r2)-(2/3)*T1*r1^3*log(r1)-
L1*r2^2+(2/5)*r2^5*M1-(2/9)*d1*r2^3+(2/3)*r2^3*N1-(8/5)*A1*r2^5-(4/3)*B1*r2^3+2*r2*S1-2*C1*r2-
(2/9)*T1*r2^3+r2^2*V1-(2/5)*r1^5*M1+(2/9)*d1*r1^3-
(2/3)*r1^3*N1+(4/3)*B1*r1^3+(8/5)*A1*r1^5+(2/9)*T1*r1^3-r1^2*V1-
2*r1*S1+2*C1*r1+L1*r1^2)+lambda1*r2;
        p0*pi*l^2-p0*pi*r1^2+lambda1;
        pi*(h2-h1)*G*k*((4/3)*T2*r3^6*log(r3)-(4/3)*T2*r2^6*log(r2)+4*A2*r2^8-(2/9)*T2*r3^6-
(8/3)*r3^6*B2-2*r3^4*C2-(8/9)*r3^6*d2+(4/3)*r3^6*N2-(8/5)*r3^5*L2+r3^8*M2-
(4/3)*r2^6*N2+(8/5)*r2^5*L2-r2^8*M2+(2/9)*T2*r2^6+(8/3)*r2^6*B2+2*r2^4*C2+(8/9)*r2^6*d2-
(8/5)*r2^5*V2+(8/3)*d2*r2^6*log(r2)-2*r2^4*S2+2*r3^4*S2-4*A2*r3^8+(8/5)*r3^5*V2-
(8/3)*d2*r3^6*log(r3))+pi*(h1-h0)*G*k*(4*A2*r2^8-2*r2^4*S1-(8/5)*r2^5*V1-(4/3)*r2^6*N1-r2^8*M1-
(8/3)*r3^6*B2-2*r3^4*C2-(8/9)*r3^6*d2-
(8/5)*r3^5*L2+(8/5)*r2^5*L2+(8/3)*r2^6*B2+2*r2^4*C2+(8/9)*r2^6*d2-
(4/3)*T1*r2^6*log(r2)+(8/3)*d2*r2^6*log(r2)-
4*A2*r3^8+2*r3^4*S1+(8/5)*r3^5*V1+(4/3)*r3^6*N1+r3^8*M1-
(2/9)*T1*r3^6+(4/3)*T1*r3^6*log(r3)+(2/9)*T1*r2^6-(8/3)*d2*r3^6*log(r3))-
lambda1*r2^4+lambda2*r3^4;
        pi*(h2-h1)*G*k*(-T2*r2^4*log(r2)+2*B2*r2^4-2*d2*r3^4*log(r3)+(4/3)*r3^3*V2-
(1/4)*T2*r3^4+2*r3^2*S2-(8/3)*A2*r3^6+(2/3)*r3^6*M2-(4/3)*r3^3*L2+r3^4*N2-2*r3^2*C2-
(1/2)*r3^4*d2+(1/4)*T2*r2^4-(4/3)*r2^3*V2-2*r2^2*S2+(8/3)*A2*r2^6+2*r2^2*C2+(1/2)*r2^4*d2-
(2/3)*r2^6*M2+(4/3)*r2^3*L2-r2^4*N2-2*B2*r3^4+2*d2*r2^4*log(r2)+T2*r3^4*log(r3))+pi*(h1-
h0)*G*k*(2*B2*r2^4-2*d2*r3^4*log(r3)-(4/3)*r2^3*V1-r2^4*N1-2*r2^2*S1-(2/3)*r2^6*M1-(8/3)*A2*r3^6-
(4/3)*r3^3*L2-2*r3^2*C2+(1/4)*T1*r2^4-(1/2)*r3^4*d2+(8/3)*A2*r2^6-
T1*r2^4*log(r2)+2*r2^2*C2+(1/2)*r2^4*d2+(4/3)*r2^3*L2-
2*B2*r3^4+2*d2*r2^4*log(r2)+(4/3)*r3^3*V1+r3^4*N1+2*r3^2*S1+(2/3)*r3^6*M1-
(1/4)*T1*r3^4+T1*r3^4*log(r3))-lambda1*r2^2+lambda2*r3^2;
        pi*(h2-h1)*G*k*(2*C2*log(r2)+2*d2*r2^2*log(r2)-T2*r2^2*log(r2)-
2*d2*r3^2*log(r3)+2*V2*r3+2*S2*log(r3)-(1/2)*T2*r3^2+(1/2)*M2*r3^4+N2*r3^2-2*A2*r3^4-2*B2*r3^2-
2*L2*r3+(1/2)*T2*r2^2-N2*r2^2+2*A2*r2^4+T2*r3^2*log(r3)-2*V2*r2+2*L2*r2-(1/2)*M2*r2^4+2*B2*r2^2-
2*S2*log(r2)-2*C2*log(r3))+pi*(h1-h0)*G*k*(2*C2*log(r2)+2*d2*r2^2*log(r2)-2*d2*r3^2*log(r3)-
2*V1*r2-(1/2)*M1*r2^4-2*A2*r3^4-2*B2*r3^2-2*S1*log(r2)-N1*r2^2+(1/2)*T1*r2^2-2*L2*r3+2*A2*r2^4-
T1*r2^2*log(r2)+2*L2*r2+2*B2*r2^2-2*C2*log(r3)+2*V1*r3+(1/2)*M1*r3^4+2*S1*log(r3)+N1*r3^2-
(1/2)*T1*r3^2+T1*r3^2*log(r3))-lambda1*log(r2)+lambda2*log(r3);
        pi*(h2-h1)*G*k*(-(2/9)*r2^3*V2+(2/9)*r3^6*M2-T2*r2^4*log(r2)^2-(4/3)*V2*r2^3*log(r2)-
N2*r2^4*log(r2)+(4/3)*L2*r2^3*log(r2)+2*C2*r2^2*log(r2)+(2/3)*M2*r3^6*log(r3)-2*B2*r3^4*log(r3)-
2*S2*r2^2*log(r2)+(1/4)*r2^4*d2+2*S2*r3^2*log(r3)-2*C2*r3^2*log(r3)-
(4/3)*L2*r3^3*log(r3)+(4/3)*V2*r3^3*log(r3)+T2*r3^4*log(r3)^2-
(2/3)*M2*r2^6*log(r2)+N2*r3^4*log(r3)-
(8/3)*A2*r3^6*log(r3)+2*B2*r2^4*log(r2)+(8/3)*A2*r2^6*log(r2)+2*d2*r2^4*log(r2)^2-
(2/9)*r3^3*L2+(1/2)*B2*r2^4-(1/2)*B2*r3^4+(8/9)*A2*r2^6-(8/9)*A2*r3^6-(1/4)*r3^4*d2-
(1/4)*r2^4*N2-2*d2*r3^4*log(r3)^2+(2/9)*r3^3*V2-d2*r3^4*log(r3)+d2*r2^4*log(r2)+(1/4)*r3^4*N2-
(2/9)*r2^6*M2+(2/9)*r2^3*L2)+pi*(h1-h0)*G*k*(-
(2/9)*r2^6*M1+(4/3)*L2*r2^3*log(r2)+2*C2*r2^2*log(r2)-2*B2*r3^4*log(r3)+(1/4)*r2^4*d2-
2*C2*r3^2*log(r3)-(4/3)*L2*r3^3*log(r3)-(2/9)*r2^3*V1-(2/3)*M1*r2^6*log(r2)+N1*r3^4*log(r3)-
(8/3)*A2*r3^6*log(r3)+2*B2*r2^4*log(r2)+(8/3)*A2*r2^6*log(r2)+(2/3)*M1*r3^6*log(r3)+2*S1*r3^2*log
(r3)+(4/3)*V1*r3^3*log(r3)-(4/3)*V1*r2^3*log(r2)-T1*r2^4*log(r2)^2-2*S1*r2^2*log(r2)-
N1*r2^4*log(r2)+T1*r3^4*log(r3)^2+2*d2*r2^4*log(r2)^2-(2/9)*r3^3*L2+(1/2)*B2*r2^4-
(1/2)*B2*r3^4+(8/9)*A2*r2^6+(2/9)*r3^3*V1-(8/9)*A2*r3^6+(2/9)*r3^6*M1-(1/4)*r3^4*d2-
2*d2*r3^4*log(r3)^2-(1/4)*r2^4*N1+(1/4)*r3^4*N1-d2*r3^4*log(r3)+d2*r2^4*log(r2)+(2/9)*r2^3*L2)-
lambda1*r2^2*log(r2)+lambda2*r3^2*log(r3);
        pi*(h2-h1)*G*k*((4/3)*d2*r2^3*log(r2)-
(2/3)*T2*r2^3*log(r2)+(2/3)*r3^3*N2+2*r3*S2+r3^2*V2+L2*r2^2+(2/3)*T2*r3^3*log(r3)-
(4/3)*d2*r3^3*log(r3)-(8/5)*A2*r3^5-(2/9)*T2*r3^3-(4/3)*B2*r3^3-
2*C2*r3+(8/5)*A2*r2^5+(2/9)*d2*r2^3+2*C2*r2-(2/3)*r2^3*N2-2*r2*S2-r2^2*V2-
(2/5)*r2^5*M2+(2/9)*T2*r2^3+(4/3)*B2*r2^3+(2/5)*r3^5*M2-L2*r3^2-(2/9)*d2*r3^3)+pi*(h1-
h0)*G*k*((4/3)*d2*r2^3*log(r2)+(2/3)*T1*r3^3*log(r3)-(2/9)*T1*r3^3+L2*r2^2-(4/3)*d2*r3^3*log(r3)-
r2^2*V1-(2/3)*r2^3*N1-2*r2*S1-(2/5)*r2^5*M1-(8/5)*A2*r3^5-(4/3)*B2*r3^3-2*C2*r3+(8/5)*A2*r2^5-
(2/3)*T1*r2^3*log(r2)+(2/9)*d2*r2^3+2*C2*r2+(4/3)*B2*r2^3-
L2*r3^2+r3^2*V1+(2/3)*r3^3*N1+2*r3*S1+(2/5)*r3^5*M1+(2/9)*T1*r2^3-(2/9)*d2*r3^3)-
lambda1*r2+lambda2*r3;
        -lambda1+lambda2;
        pi*(h3-h2)*G*k*((8/3)*r3^6*B3+2*r3^4*C3+(8/9)*r3^6*d3+(8/5)*r3^5*L3-r3^8*M3-
(4/3)*r3^6*N3-2*r3^4*S3-
(8/3)*d3*a^6*log(a)+(4/3)*T3*a^6*log(a)+(8/3)*d3*r3^6*log(r3)+(2/9)*T3*r3^6-4*A3*a^8-
```

```
(8/5)*r3^5*V3-(8/3)*a^6*B3-2*a^4*C3-(8/9)*a^6*d3-
(8/5)*a^5*L3+a^8*M3+(4/3)*a^6*N3+2*a^4*S3+(8/5)*a^5*V3-(2/9)*T3*a^6-
(4/3)*T3*r3^6*log(r3)+4*A3*r3^8)+pi*(h2-
h1)*G*k*((4/3)*T2*a^6*log(a)+(8/3)*r3^6*B3+2*r3^4*C3+(8/9)*r3^6*d3+(8/5)*r3^5*L3-(2/9)*T2*a^6-
(8/3)*d3*a^6*log(a)+(8/3)*d3*r3^6*log(r3)+2*a^4*S2+(4/3)*a^6*N2+a^8*M2+(8/5)*a^5*V2-r3^8*M2-
(8/5)*r3^5*V2-(4/3)*T2*r3^6*log(r3)-4*A3*a^8-(8/3)*a^6*B3-2*a^4*C3-(8/9)*a^6*d3-(8/5)*a^5*L3-
2*r3^4*S2-(4/3)*r3^6*N2+(2/9)*T2*r3^6+4*A3*r3^8)+pi*(h1-h0)*G*k*((8/5)*a^5*V1-
(2/9)*T1*a^6+a^8*M1+(4/3)*a^6*N1+2*a^4*S1+(4/3)*T1*a^6*log(a)-r3^8*M1-(4/3)*r3^6*N1-2*r3^4*S1-
(8/5)*r3^5*V1+(2/9)*T1*r3^6-
(4/3)*T1*r3^6*log(r3)+(8/3)*r3^6*B3+2*r3^4*C3+(8/9)*r3^6*d3+(8/5)*r3^5*L3-
(8/3)*d3*a^6*log(a)+(8/3)*d3*r3^6*log(r3)-4*A3*a^8-(8/3)*a^6*B3-2*a^4*C3-(8/9)*a^6*d3-
(8/5)*a^5*L3+4*A3*r3^8)-lambda2*r3^4+lambda3*a^4;
        pi*(h3-h2)*G*k*((1/4)*T3*r3^4+(8/3)*A3*r3^6-
2*d3*a^4*log(a)+T3*a^4*log(a)+2*d3*r3^4*log(r3)-
T3*r3^4*log(r3)+2*r3^2*C3+(1/2)*r3^4*d3+(4/3)*r3^3*L3-(2/3)*r3^6*M3-r3^4*N3-2*r3^2*S3-
(4/3)*r3^3*V3-2*B3*a^4-(8/3)*A3*a^6-(1/4)*T3*a^4-2*a^2*C3-(1/2)*a^4*d3-
(4/3)*a^3*L3+(2/3)*a^6*M3+a^4*N3+2*a^2*S3+(4/3)*a^3*V3+2*B3*r3^4)+pi*(h2-
h1)*G*k*(T2*a^4*log(a)+(8/3)*A3*r3^6-
2*d3*a^4*log(a)+2*d3*r3^4*log(r3)+2*r3^2*C3+(1/2)*r3^4*d3+(4/3)*r3^3*L3-(1/4)*T2*a^4-
(4/3)*r3^3*V2-2*r3^2*S2-r3^4*N2+(2/3)*a^6*M2+(4/3)*a^3*V2+2*a^2*S2+a^4*N2-(2/3)*r3^6*M2-
T2*r3^4*log(r3)-2*B3*a^4-(8/3)*A3*a^6-2*a^2*C3-(1/2)*a^4*d3-
(4/3)*a^3*L3+(1/4)*T2*r3^4+2*B3*r3^4)+pi*(h1-h0)*G*k*(-
(1/4)*T1*a^4+(2/3)*a^6*M1+a^4*N1+2*a^2*S1+T1*a^4*log(a)+(4/3)*a^3*V1+(1/4)*T1*r3^4-(4/3)*r3^3*V1-
(2/3)*r3^6*M1-r3^4*N1-2*r3^2*S1-T1*r3^4*log(r3)+(8/3)*A3*r3^6-
2*d3*a^4*log(a)+2*d3*r3^4*log(r3)+2*r3^2*C3+(1/2)*r3^4*d3+(4/3)*r3^3*L3-2*B3*a^4-(8/3)*A3*a^6-
2*a^2*C3-(1/2)*a^4*d3-(4/3)*a^3*L3+2*B3*r3^4)-lambda2*r3^2+lambda3*a^2;
        pi*(h3-h2)*G*k*(2*A3*r3^4-T3*r3^2*log(r3)+T3*a^2*log(a)-
2*d3*a^2*log(a)+2*d3*r3^2*log(r3)+2*B3*r3^2+2*L3*r3-(1/2)*M3*r3^4-N3*r3^2-2*V3*r3-
2*S3*log(r3)+(1/2)*T3*r3^2-(1/2)*T3*a^2-2*C3*log(a)-2*A3*a^4-2*B3*a^2-
2*L3*a+(1/2)*M3*a^4+N3*a^2+2*V3*a+2*S3*log(a)+2*C3*log(r3))+pi*(h2-h1)*G*k*(-
(1/2)*T2*a^2+T2*a^2*log(a)+2*A3*r3^4-2*d3*a^2*log(a)+2*d3*r3^2*log(r3)+2*B3*r3^2+2*L3*r3-2*V2*r3-
N2*r3^2-(1/2)*M2*r3^4+2*S2*log(a)+2*V2*a+N2*a^2+(1/2)*M2*a^4-2*S2*log(r3)-T2*r3^2*log(r3)-
2*C3*log(a)-2*A3*a^4-2*B3*a^2-2*L3*a+(1/2)*T2*r3^2+2*C3*log(r3))+pi*(h1-
h0)*G*k*(2*S1*log(a)+2*V1*a+N1*a^2+T1*a^2*log(a)+(1/2)*M1*a^4+(1/2)*T1*r3^2-(1/2)*M1*r3^4-
N1*r3^2-2*V1*r3-2*S1*log(r3)-(1/2)*T1*a^2-T1*r3^2*log(r3)+2*A3*r3^4-
2*d3*a^2*log(a)+2*d3*r3^2*log(r3)+2*B3*r3^2+2*L3*r3-2*C3*log(a)-2*A3*a^4-2*B3*a^2-
2*L3*a+2*C3*log(r3))-lambda2*log(r3)+lambda3*log(a);
        pi*(h3-h2)*G*k*(2*S3*a^2*log(a)-(4/3)*L3*a^3*log(a)+(4/3)*V3*a^3*log(a)-2*B3*a^4*log(a)-
(8/3)*A3*a^6*log(a)+(2/3)*M3*a^6*log(a)+N3*a^4*log(a)-T3*r3^4*log(r3)^2-
2*S3*r3^2*log(r3)+2*C3*r3^2*log(r3)+(4/3)*L3*r3^3*log(r3)-2*C3*a^2*log(a)-2*d3*a^4*log(a)^2-
(4/3)*V3*r3^3*log(r3)+2*B3*r3^4*log(r3)+d3*r3^4*log(r3)-d3*a^4*log(a)-
N3*r3^4*log(r3)+T3*a^4*log(a)^2+2*d3*r3^4*log(r3)^2-(1/4)*a^4*d3+(2/9)*r3^3*L3+(1/4)*a^4*N3-
(8/9)*A3*a^6-(1/4)*r3^4*N3+(8/9)*A3*r3^6-
(1/2)*B3*a^4+(1/2)*B3*r3^4+(8/3)*A3*r3^6*log(r3)+(2/9)*a^6*M3-
(2/3)*M3*r3^6*log(r3)+(2/9)*a^3*V3+(1/4)*r3^4*d3-(2/9)*r3^3*V3-(2/9)*r3^6*M3-
(2/9)*a^3*L3)+pi*(h2-h1)*G*k*(-(4/3)*L3*a^3*log(a)-2*B3*a^4*log(a)-
(8/3)*A3*a^6*log(a)+N2*a^4*log(a)+2*C3*r3^2*log(r3)+(4/3)*L3*r3^3*log(r3)-
2*C3*a^2*log(a)+(4/3)*V2*a^3*log(a)-(2/3)*M2*r3^6*log(r3)+T2*a^4*log(a)^2-(4/3)*V2*r3^3*log(r3)-
N2*r3^4*log(r3)+(2/3)*M2*a^6*log(a)-T2*r3^4*log(r3)^2-2*S2*r3^2*log(r3)+2*S2*a^2*log(a)-
2*d3*a^4*log(a)^2+2*B3*r3^4*log(r3)+d3*r3^4*log(r3)-
d3*a^4*log(a)+(2/9)*a^6*M2+2*d3*r3^4*log(r3)^2-(1/4)*a^4*d3+(2/9)*r3^3*L3-
(8/9)*A3*a^6+(8/9)*A3*r3^6-(1/2)*B3*a^4+(1/2)*B3*r3^4+(8/3)*A3*r3^6*log(r3)-(2/9)*r3^3*V2-
(2/9)*r3^6*M2+(1/4)*r3^4*d3+(1/4)*a^4*N2-(1/4)*r3^4*N2-(2/9)*a^3*L3+(2/9)*a^3*V2)+pi*(h1-
h0)*G*k*(-(4/3)*L3*a^3*log(a)-2*B3*a^4*log(a)-
(8/3)*A3*a^6*log(a)+2*C3*r3^2*log(r3)+(4/3)*L3*r3^3*log(r3)-
2*C3*a^2*log(a)+(4/3)*V1*a^3*log(a)+N1*a^4*log(a)+(2/3)*M1*a^6*log(a)+T1*a^4*log(a)^2-
2*d3*a^4*log(a)^2-N1*r3^4*log(r3)+2*S1*a^2*log(a)-2*S1*r3^2*log(r3)-(2/3)*M1*r3^6*log(r3)-
(4/3)*V1*r3^3*log(r3)-T1*r3^4*log(r3)^2+2*B3*r3^4*log(r3)+d3*r3^4*log(r3)-
d3*a^4*log(a)+2*d3*r3^4*log(r3)^2-(1/4)*a^4*d3+(2/9)*r3^3*L3-(8/9)*A3*a^6+(8/9)*A3*r3^6-
(1/2)*B3*a^4+(1/2)*B3*r3^4-
(2/9)*r3^6*M1+(8/3)*A3*r3^6*log(r3)+(1/4)*r3^4*d3+(2/9)*a^6*M1+(2/9)*a^3*V1+(1/4)*a^4*N1-
(2/9)*r3^3*V1-(1/4)*r3^4*N1-(2/9)*a^3*L3)-lambda2*r3^2*log(r3)+lambda3*a^2*log(a);
        pi*(h3-h2)*G*k*((2/9)*d3*r3^3+(8/5)*A3*r3^5+(2/3)*T3*a^3*log(a)-(4/3)*d3*a^3*log(a)-
(2/3)*T3*r3^3*log(r3)+(4/3)*d3*r3^3*log(r3)+(4/3)*B3*r3^3+2*C3*r3-(2/5)*r3^5*M3-(2/3)*r3^3*N3-
2*r3*S3-r3^2*V3+(2/9)*T3*r3^3-L3*a^2-(2/9)*d3*a^3-(8/5)*A3*a^5-(4/3)*B3*a^3-
2*C3*a+(2/5)*a^5*M3+(2/3)*a^3*N3+2*a*S3+a^2*V3-(2/9)*T3*a^3+L3*r3^2)+pi*(h2-
h1)*G*k*((2/3)*T2*a^3*log(a)+(2/9)*d3*r3^3+(8/5)*A3*r3^5-
(4/3)*d3*a^3*log(a)+(4/3)*d3*r3^3*log(r3)+(4/3)*B3*r3^3+2*C3*r3-(2/9)*T2*a^3-r3^2*V2-2*r3*S2-
(2/3)*r3^3*N2-(2/5)*r3^5*M2+a^2*V2+2*a*S2+(2/3)*a^3*N2+(2/5)*a^5*M2-(2/3)*T2*r3^3*log(r3)-L3*a^2-
```

```
(2/9)*d3*a^3-(8/5)*A3*a^5-(4/3)*B3*a^3-2*C3*a+(2/9)*T2*r3^3+L3*r3^2)+pi*(h1-h0)*G*k*(-
(2/9)*T1*a^3+a^2*V1+(2/5)*a^5*M1+(2/3)*a^3*N1+2*a*S1+(2/9)*T1*r3^3-r3^2*V1-(2/5)*r3^5*M1-
(2/3)*r3^3*N1-2*r3*S1+(2/3)*T1*a^3*log(a)-(2/3)*T1*r3^3*log(r3)+(2/9)*d3*r3^3+(8/5)*A3*r3^5-
(4/3)*d3*a^3*log(a)+(4/3)*d3*r3^3*log(r3)+(4/3)*B3*r3^3+2*C3*r3-L3*a^2-(2/9)*d3*a^3-(8/5)*A3*a^5-
(4/3)*B3*a^3-2*C3*a+L3*r3^2)-lambda2*r3+lambda3*a;
        -lambda2+lambda3;
        (1/72)*pi*(h1-h0)^3*E*(12*nu*V1*r1^5*a^2+12*nu*S1*r1^4*a^2+12*nu*N1*r1^6*a^2-
12*nu*V1*a^5*r1^2-12*nu*N1*a^6*r1^2-12*nu*S1*a^4*r1^2+12*T1*r1^6*log(r1)*a^2-
12*N1*a^6*r1^2+12*S1*a^4*r1^2-6*T1*a^6*r1^2-4*V1*a^5*r1^2-12*nu*M1*a^8*r1^2+12*N1*r1^6*a^2-
12*S1*r1^4*a^2+6*T1*r1^6*a^2+4*V1*r1^5*a^2+12*nu*M1*r1^8*a^2-12*T1*a^6*log(a)*r1^2-
12*nu*T1*a^6*log(a)*r1^2+12*nu*T1*r1^6*log(r1)*a^2-20*M1*a^8*r1^2+20*M1*r1^8*a^2)/((1-
nu^2)*r1^2*a^2)+pi*(h1-h0)*G*k*((2/3)*B3*a^6+A3*a^8-(2/5)*a^5*V1-
(1/3)*a^6*N1+(2/5)*L3*a^5+(2/9)*d3*a^6+(1/18)*T1*a^6+(1/2)*C3*a^4+(2/3)*d3*a^6*log(a)+(1/2)*r3^4*
S1+(2/5)*r3^5*V1+(1/3)*r3^6*N1-(2/5)*L3*r3^5-(2/9)*d3*r3^6-(1/18)*T1*r3^6-(1/2)*C3*r3^4-
(2/3)*B3*r3^6-A3*r3^8-(1/2)*a^4*S1-(1/3)*T1*a^6*log(a)+(1/3)*T1*r3^6*log(r3)-
(2/3)*d3*r3^6*log(r3)-(1/4)*M1*a^8+(1/4)*M1*r3^8)+pi*(h1-h0)*G*k*(-(1/2)*r3^4*S1-(2/5)*r3^5*V1-
(1/3)*r3^6*N1+(1/18)*T1*r3^6-(1/3)*T1*r3^6*log(r3)-(1/4)*M1*r3^8-(2/5)*L2*r2^5-(2/9)*d2*r2^6-
(1/2)*C2*r2^4-(2/3)*B2*r2^6-A2*r2^8+(1/2)*r2^4*S1-
(1/18)*T1*r2^6+(1/3)*r2^6*N1+(2/5)*r2^5*V1+(1/3)*T1*r2^6*log(r2)-
(2/3)*d2*r2^6*log(r2)+(1/4)*M1*r2^8+(2/9)*d2*r3^6+(1/2)*C2*r3^4+(2/3)*B2*r3^6+A2*r3^8+(2/5)*L2*r3
^5+(2/3)*d2*r3^6*log(r3))+pi*(h1-h0)*G*k*((2/3)*d1*r2^6*log(r2)+(1/3)*T1*r1^6*log(r1)-
(2/3)*d1*r1^6*log(r1)+(2/5)*L1*r2^5-(1/2)*r2^4*S1+(2/9)*d1*r2^6+A1*r2^8+(1/18)*T1*r2^6-
(1/3)*r2^6*N1-(2/5)*r2^5*V1+(2/3)*B1*r2^6+(1/2)*C1*r2^4-(1/3)*T1*r2^6*log(r2)-
(2/5)*L1*r1^5+(1/2)*r1^4*S1-(2/9)*d1*r1^6-(1/18)*T1*r1^6-(2/3)*B1*r1^6-A1*r1^8-
(1/2)*C1*r1^4+(2/5)*r1^5*V1+(1/3)*r1^6*N1-(1/4)*M1*r2^8+(1/4)*M1*r1^8)+lambda4*a^3;
        (1/72)*pi*(h1-h0)^3*E*(12*nu*V1*r1^3*a^2+12*nu*M1*r1^6*a^2-12*nu*M1*a^6*r1^2-
12*nu*V1*a^3*r1^2+12*nu*T1*r1^4*log(r1)*a^2-12*nu*N1*a^4*r1^2-12*M1*a^6*r1^2-
12*V1*a^3*r1^2+12*nu*N1*r1^4*a^2+12*M1*r1^6*a^2+12*V1*r1^3*a^2+12*T1*r1^4*log(r1)*a^2-
12*T1*a^4*log(a)*r1^2-12*nu*T1*a^4*log(a)*r1^2-12*N1*a^4*r1^2+12*N1*r1^4*a^2)/((1-
nu^2)*r1^2*a^2)+pi*(h1-h0)*G*k*((1/8)*T1*a^4+B3*a^4+(4/3)*A3*a^6+C3*a^2-
(1/3)*M1*a^6+(2/3)*L3*a^3+(1/4)*d3*a^4-(1/8)*T1*r3^4+r3^2*S1+(2/3)*r3^3*V1+(1/3)*M1*r3^6-
(2/3)*L3*r3^3-(1/4)*d3*r3^4+d3*a^4*log(a)-C3*r3^2-B3*r3^4-(4/3)*A3*r3^6-a^2*S1-(2/3)*a^3*V1-
(1/2)*T1*a^4*log(a)-d3*r3^4*log(r3)+(1/2)*T1*r3^4*log(r3)-(1/2)*N1*a^4+(1/2)*N1*r3^4)+pi*(h1-
h0)*G*k*((1/8)*T1*r3^4-r3^2*S1-(2/3)*r3^3*V1-(1/3)*M1*r3^6-(1/2)*T1*r3^4*log(r3)+d2*r3^4*log(r3)-
(1/2)*N1*r3^4-(2/3)*L2*r2^3-(1/4)*d2*r2^4-B2*r2^4-(4/3)*A2*r2^6+r2^2*S1+(2/3)*r2^3*V1-
(1/8)*T1*r2^4-
C2*r2^2+(1/3)*M1*r2^6+(1/2)*N1*r2^4+(1/4)*d2*r3^4+B2*r3^4+(4/3)*A2*r3^6+(2/3)*L2*r3^3+C2*r3^2-
d2*r2^4*log(r2)+(1/2)*T1*r2^4*log(r2))+pi*(h1-h0)*G*k*(d1*r2^4*log(r2)+(1/2)*T1*r1^4*log(r1)-
d1*r1^4*log(r1)-r2^2*S1-(2/3)*r2^3*V1+(1/8)*T1*r2^4+(1/4)*d1*r2^4+(2/3)*L1*r2^3-
(1/3)*M1*r2^6+B1*r2^4+(4/3)*A1*r2^6+C1*r2^2+(2/3)*r1^3*V1+r1^2*S1-(1/4)*d1*r1^4-(2/3)*L1*r1^3-
(1/8)*T1*r1^4-(4/3)*A1*r1^6-C1*r1^2-B1*r1^4+(1/3)*M1*r1^6-(1/2)*N1*r2^4+(1/2)*N1*r1^4-
(1/2)*T1*r2^4*log(r2))+lambda4*a;
        (1/72)*pi*(h1-h0)^3*E*(12*nu*V1*r1*a^2+12*nu*M1*r1^4*a^2-12*nu*M1*a^4*r1^2-
12*V1*r1*a^2+12*nu*S1*a^2+12*V1*r1^2*a-12*nu*S1*r1^2-12*nu*V1*r1^2*a-
12*T1*log(r1)*r1^2*a^2+12*T1*log(a)*r1^2*a^2+12*nu*T1*log(r1)*r1^2*a^2+12*M1*a^4*r1^2-
12*M1*r1^4*a^2+12*r1^2*S1-12*a^2*S1-12*nu*T1*log(a)*r1^2*a^2)/((1-nu^2)*r1^2*a^2)+pi*(h1-
h0)*G*k*(2*C3*log(a)+(1/2)*T1*a^2+2*B3*a^2+2*A3*a^4+2*L3*a+(1/2)*M1*r3^4+N1*r3^2+2*V1*r3-2*L3*r3-
2*B3*r3^2-2*A3*r3^4-2*C3*log(r3)-(1/2)*T1*r3^2-(1/2)*M1*a^4-N1*a^2-2*V1*a-
T1*a^2*log(a)+2*d3*a^2*log(a)+T1*r3^2*log(r3)-2*d3*r3^2*log(r3)-2*S1*log(a)+2*S1*log(r3))+pi*(h1-
h0)*G*k*(-(1/2)*M1*r3^4-N1*r3^2-2*V1*r3+(1/2)*T1*r3^2-T1*r3^2*log(r3)-2*S1*log(r3)-2*L2*r2-
2*B2*r2^2-2*A2*r2^4+(1/2)*M1*r2^4+N1*r2^2+2*V1*r2-(1/2)*T1*r2^2-
2*C2*log(r2)+T1*r2^2*log(r2)+2*S1*log(r2)+2*B2*r3^2+2*A2*r3^4+2*L2*r3+2*C2*log(r3)+2*d2*r3^2*log(
r3)-2*d2*r2^2*log(r2))+pi*(h1-h0)*G*k*(2*d1*r2^2*log(r2)+T1*r1^2*log(r1)-2*d1*r1^2*log(r1)-
(1/2)*M1*r2^4-N1*r2^2-2*V1*r2+(1/2)*T1*r2^2+2*A1*r2^4+2*L1*r2+2*B1*r2^2-
T1*r2^2*log(r2)+2*C1*log(r2)+2*V1*r1+N1*r1^2+(1/2)*M1*r1^4-2*A1*r1^4-2*C1*log(r1)-2*L1*r1-
2*B1*r1^2-(1/2)*T1*r1^2-2*S1*log(r2)+2*S1*log(r1))+lambda4/a;
        (1/72)*pi*(h1-h0)^3*E*(6*T1*r1^4*a^2-12*V1*r1^3*a^2+6*M1*r1^6*a^2+12*V1*a^3*r1^2-
6*M1*a^6*r1^2+12*nu*T1*r1^4*log(r1)^2*a^2+12*S1*log(a)*r1^2*a^2+12*V1*r1^3*log(r1)*a^2+12*M1*r1^6
*log(r1)*a^2-12*nu*T1*a^4*log(a)^2*r1^2-12*M1*a^6*log(a)*r1^2-12*S1*log(r1)*r1^2*a^2-
6*T1*a^4*r1^2+12*N1*r1^4*log(r1)*a^2-
12*nu*S1*log(a)*r1^2*a^2+12*nu*S1*log(r1)*r1^2*a^2+12*nu*N1*r1^4*log(r1)*a^2-
12*N1*a^4*log(a)*r1^2+12*nu*V1*r1^3*log(r1)*a^2+12*nu*M1*r1^6*log(r1)*a^2-
12*nu*M1*a^6*log(a)*r1^2-12*nu*N1*a^4*log(a)*r1^2-12*V1*a^3*log(a)*r1^2-
12*nu*V1*a^3*log(a)*r1^2+12*T1*r1^4*log(r1)^2*a^2-12*T1*a^4*log(a)^2*r1^2)/((1-
nu^2)*r1^2*a^2)+pi*(h1-h0)*G*k*(-(1/8)*N1*r3^4+(1/8)*N1*a^4-
(1/4)*T1*r3^4*log(r3)+(1/4)*T1*a^4*log(a)-
(1/2)*T1*a^4*log(a)^2+(1/2)*T1*r3^4*log(r3)^2+d3*a^4*log(a)^2-(1/16)*T1*a^4-
(2/9)*A3*a^6+C3*a^2*log(a)-(1/4)*B3*a^4-(1/2)*r3^2*S1+(1/16)*T1*r3^4-(2/9)*L3*a^3-
(1/3)*M1*a^6*log(a)+(4/3)*A3*a^6*log(a)+(1/18)*M1*a^6-(1/2)*C3*a^2+B3*a^4*log(a)-C3*r3^2*log(r3)-
```

```
(2/9)*r3^3*V1-d3*r3^4*log(r3)^2+(2/3)*L3*a^3*log(a)-(1/2)*N1*a^4*log(a)-(2/3)*V1*a^3*log(a)-
S1*a^2*log(a)-(2/3)*L3*r3^3*log(r3)+(2/9)*L3*r3^3+(1/2)*N1*r3^4*log(r3)-
(4/3)*A3*r3^6*log(r3)+(2/3)*V1*r3^3*log(r3)+S1*r3^2*log(r3)+(1/3)*M1*r3^6*log(r3)-
B3*r3^4*log(r3)-
(1/18)*M1*r3^6+(1/2)*C3*r3^2+(2/9)*a^3*V1+(1/2)*a^2*S1+(2/9)*A3*r3^6+(1/4)*B3*r3^4)+pi*(h1-
h0)*G*k*(-(1/8)*N1*r2^4+(1/8)*N1*r3^4+(1/4)*T1*r3^4*log(r3)-(1/4)*T1*r2^4*log(r2)-
(1/2)*T1*r3^4*log(r3)^2+(1/2)*T1*r2^4*log(r2)^2+(1/2)*r3^2*S1-(1/16)*T1*r3^4+(2/9)*r3^3*V1-
(1/2)*N1*r3^4*log(r3)-(2/3)*V1*r3^3*log(r3)-S1*r3^2*log(r3)-
(1/3)*M1*r3^6*log(r3)+(1/18)*M1*r3^6+(2/3)*L2*r3^3*log(r3)-(4/3)*A2*r2^6*log(r2)-
B2*r2^4*log(r2)+(2/3)*V1*r2^3*log(r2)+S1*r2^2*log(r2)+(1/3)*M1*r2^6*log(r2)-
(2/3)*L2*r2^3*log(r2)-(2/9)*r2^3*V1-
(1/2)*r2^2*S1+(4/3)*A2*r3^6*log(r3)+(2/9)*A2*r2^6+(1/4)*B2*r2^4+B2*r3^4*log(r3)-
C2*r2^2*log(r2)+(2/9)*L2*r2^3-
(1/18)*M1*r2^6+(1/2)*C2*r2^2+(1/16)*T1*r2^4+d2*r3^4*log(r3)^2+(1/2)*N1*r2^4*log(r2)-
d2*r2^4*log(r2)^2+C2*r3^2*log(r3)-(1/4)*B2*r3^4-(1/2)*C2*r3^2-(2/9)*L2*r3^3-
(2/9)*A2*r3^6)+pi*(h1-h0)*G*k*(-(2/9)*r1^3*V1-(2/9)*A1*r2^6-(1/2)*C1*r2^2-
(1/8)*N1*r1^4+(1/8)*N1*r2^4-(1/4)*T1*r1^4*log(r1)+(1/4)*T1*r2^4*log(r2)-(1/18)*M1*r1^6-
(1/2)*T1*r2^4*log(r2)^2+(2/9)*L1*r1^3+(2/9)*A1*r1^6+(1/16)*T1*r1^4+(1/2)*C1*r1^2+(1/4)*B1*r1^4-
(2/3)*V1*r2^3*log(r2)-S1*r2^2*log(r2)-
(1/3)*M1*r2^6*log(r2)+(2/9)*r2^3*V1+(1/2)*r2^2*S1+C1*r2^2*log(r2)+(4/3)*A1*r2^6*log(r2)+d1*r2^4*l
og(r2)^2+B1*r2^4*log(r2)+(1/18)*M1*r2^6-(1/16)*T1*r2^4-
(1/2)*N1*r2^4*log(r2)+(2/3)*V1*r1^3*log(r1)-
(4/3)*A1*r1^6*log(r1)+(2/3)*L1*r2^3*log(r2)+(1/2)*N1*r1^4*log(r1)-d1*r1^4*log(r1)^2-
(2/3)*L1*r1^3*log(r1)-C1*r1^2*log(r1)-B1*r1^4*log(r1)+(1/3)*M1*r1^6*log(r1)+S1*r1^2*log(r1)-
(2/9)*L1*r2^3-(1/4)*B1*r2^4-(1/2)*r1^2*S1+(1/2)*T1*r1^4*log(r1)^2)+lambda4*a*log(a);
        (1/72)*pi*(h1-h0)^3*E*(12*nu*S1*r1*a^2+12*nu*N1*r1^3*a^2+12*nu*M1*r1^5*a^2-
12*nu*M1*a^5*r1^2-12*nu*N1*a^3*r1^2-12*S1*r1*a^2+12*S1*r1^2*a-
12*nu*S1*r1^2*a+12*T1*r1^3*log(r1)*a^2-12*N1*a^3*r1^2-4*M1*a^5*r1^2+12*T1*a^3*r1^2-
12*T1*r1^3*a^2+12*N1*r1^3*a^2+4*M1*r1^5*a^2+12*V1*log(r1)*r1^2*a^2-12*V1*log(a)*r1^2*a^2-
12*T1*a^3*log(a)*r1^2-12*nu*T1*a^3*log(a)*r1^2+12*nu*T1*r1^3*log(r1)*a^2)/((1-
nu^2)*r1^2*a^2)+pi*(h1-h0)*G*k*((2/9)*T1*a^3+(8/5)*A3*a^5+2*C3*a-
(2/5)*M1*a^5+L3*a^2+(2/9)*d3*a^3+(4/3)*B3*a^3-(2/9)*T1*r3^3+2*S1*r3+(2/3)*N1*r3^3+(2/5)*M1*r3^5-
L3*r3^2-(2/9)*d3*r3^3-(4/3)*B3*r3^3-2*C3*r3-(8/5)*A3*r3^5-2*S1*a-
(2/3)*N1*a^3+(4/3)*d3*a^3*log(a)-(2/3)*T1*a^3*log(a)-(4/3)*d3*r3^3*log(r3)+(2/3)*T1*r3^3*log(r3)-
V1*a^2+V1*r3^2)+pi*(h1-h0)*G*k*((2/9)*T1*r3^3-2*S1*r3-(2/3)*N1*r3^3-(2/5)*M1*r3^5-
(2/3)*T1*r3^3*log(r3)-V1*r3^2-L2*r2^2-(2/9)*d2*r2^3-2*C2*r2-(4/3)*B2*r2^3-
(8/5)*A2*r2^5+(2/3)*N1*r2^3+2*S1*r2-
(2/9)*T1*r2^3+(2/5)*M1*r2^5+(2/3)*T1*r2^3*log(r2)+V1*r2^2+(4/3)*d2*r3^3*log(r3)+2*C2*r3+(4/3)*B2*
r3^3+(8/5)*A2*r3^5+L2*r3^2+(2/9)*d2*r3^3-(4/3)*d2*r2^3*log(r2))+pi*(h1-
h0)*G*k*((4/3)*d1*r2^3*log(r2)-(4/3)*d1*r1^3*log(r1)+(2/3)*T1*r1^3*log(r1)-(2/3)*N1*r2^3-
2*S1*r2+(2/9)*d1*r2^3+(2/9)*T1*r2^3-(2/5)*M1*r2^5+(8/5)*A1*r2^5+2*C1*r2+(4/3)*B1*r2^3-
(2/3)*T1*r2^3*log(r2)+L1*r2^2+2*S1*r1+(2/3)*N1*r1^3-(2/9)*d1*r1^3-(8/5)*A1*r1^5-L1*r1^2-2*C1*r1-
(4/3)*B1*r1^3-(2/9)*T1*r1^3+(2/5)*M1*r1^5-V1*r2^2+V1*r1^2)+lambda4;
        -(1/72)*pi*(h2-h1)^3*E*(12*nu*S2*a^4*r2^2+12*nu*N2*a^6*r2^2-12*nu*V2*r2^5*a^2-
12*nu*S2*r2^4*a^2+12*nu*V2*a^5*r2^2-
12*nu*N2*r2^6*a^2+12*nu*T2*a^6*log(a)*r2^2+12*T2*a^6*log(a)*r2^2-12*T2*r2^6*log(r2)*a^2-
12*nu*T2*r2^6*log(r2)*a^2-20*M2*r2^8*a^2+20*M2*a^8*r2^2+12*N2*a^6*r2^2-12*S2*a^4*r2^2-
6*T2*r2^6*a^2-4*V2*r2^5*a^2+6*T2*a^6*r2^2+4*V2*a^5*r2^2-12*nu*M2*r2^8*a^2-
12*N2*r2^6*a^2+12*S2*r2^4*a^2+12*nu*M2*a^8*r2^2)/((1-nu^2)*a^2*r2^2)+pi*(h2-h1)*G*k*(-
(1/3)*T2*a^6*log(a)-(2/3)*d3*r3^6*log(r3)-(1/4)*M2*a^8+(2/5)*r3^5*V2+(1/2)*r3^4*S2-(2/9)*d3*r3^6-
(1/2)*C3*r3^4-(1/18)*T2*r3^6-(2/5)*L3*r3^5-(2/5)*a^5*V2-(1/2)*a^4*S2-
(1/3)*a^6*N2+A3*a^8+(2/3)*B3*a^6+(2/9)*d3*a^6+(1/2)*C3*a^4+(1/18)*T2*a^6+(2/5)*L3*a^5+(1/3)*r3^6*
N2-A3*r3^8-(2/3)*B3*r3^6+(2/3)*d3*a^6*log(a)+(1/3)*T2*r3^6*log(r3)+(1/4)*M2*r3^8)+pi*(h2-
h1)*G*k*((1/4)*M2*r2^8+(2/3)*d2*r3^6*log(r3)-(2/5)*r3^5*V2-(1/2)*r3^4*S2-
(2/3)*d2*r2^6*log(r2)+(1/2)*C2*r3^4+A2*r3^8+(2/3)*B2*r3^6-A2*r2^8-(2/9)*d2*r2^6-
(1/2)*C2*r2^4+(1/2)*r2^4*S2+(1/3)*r2^6*N2-(2/5)*L2*r2^5+(2/5)*r2^5*V2-(2/3)*B2*r2^6-
(1/18)*T2*r2^6+(1/18)*T2*r3^6+(1/3)*T2*r2^6*log(r2)+(2/5)*L2*r3^5-(1/3)*r3^6*N2-
(1/3)*T2*r3^6*log(r3)-(1/4)*M2*r3^8+(2/9)*d2*r3^6)+lambda5*a^3;
        -(1/72)*pi*(h2-h1)^3*E*(12*nu*M2*a^6*r2^2-12*nu*V2*r2^3*a^2+12*nu*V2*a^3*r2^2-
12*nu*M2*r2^6*a^2+12*nu*T2*a^4*log(a)*r2^2+12*T2*a^4*log(a)*r2^2-12*T2*r2^4*log(r2)*a^2-
12*nu*T2*r2^4*log(r2)*a^2-12*N2*r2^4*a^2+12*N2*a^4*r2^2+12*V2*a^3*r2^2+12*M2*a^6*r2^2-
12*V2*r2^3*a^2+12*nu*N2*a^4*r2^2-12*nu*N2*r2^4*a^2-12*M2*r2^6*a^2)/((1-nu^2)*a^2*r2^2)+pi*(h2-
h1)*G*k*(d3*a^4*log(a)-(1/2)*N2*a^4+r3^2*S2-(1/2)*T2*a^4*log(a)-(2/3)*a^3*V2-(1/4)*d3*r3^4-
C3*r3^2-B3*r3^4-(2/3)*L3*r3^3+(4/3)*A3*a^6+(1/8)*T2*a^4-a^2*S2-
(1/3)*M2*a^6+(1/4)*d3*a^4+C3*a^2+B3*a^4+(2/3)*L3*a^3+(1/3)*M2*r3^6-
d3*r3^4*log(r3)+(1/2)*T2*r3^4*log(r3)+(1/2)*N2*r3^4+(2/3)*r3^3*V2-(4/3)*A3*r3^6-
(1/8)*T2*r3^4)+pi*(h2-h1)*G*k*((2/3)*L2*r3^3-
d2*r2^4*log(r2)+(1/2)*T2*r2^4*log(r2)+(1/2)*N2*r2^4+d2*r3^4*log(r3)-
r3^2*S2+C2*r3^2+(4/3)*A2*r3^6+B2*r3^4-(4/3)*A2*r2^6-(1/8)*T2*r2^4-C2*r2^2+(1/3)*M2*r2^6-
```

```
(2/3)*L2*r2^3-(1/4)*d2*r2^4+(2/3)*r2^3*V2+r2^2*S2-B2*r2^4-(1/3)*M2*r3^6-
(1/2)*T2*r3^4*log(r3)+(1/4)*d2*r3^4-(1/2)*N2*r3^4-(2/3)*r3^3*V2+(1/8)*T2*r3^4)+lambda5*a;
        -(1/72)*pi*(h2-h1)^3*E*(12*nu*M2*a^4*r2^2-12*nu*M2*r2^4*a^2-12*V2*a*r2^2+12*nu*S2*r2^2-
12*nu*S2*a^2+12*V2*a^2*r2+12*nu*V2*a*r2^2-
12*nu*V2*a^2*r2+12*T2*log(r2)*a^2*r2^2+12*nu*T2*log(a)*a^2*r2^2-12*nu*T2*log(r2)*a^2*r2^2-
12*T2*log(a)*a^2*r2^2-12*M2*a^4*r2^2+12*M2*r2^4*a^2-12*r2^2*S2+12*a^2*S2)/((1-
nu^2)*a^2*r2^2)+pi*(h2-h1)*G*k*(-T2*a^2*log(a)-2*S2*log(a)+(1/2)*T2*a^2+N2*r3^2+(1/2)*M2*r3^4-
2*B3*r3^2+2*V2*r3-2*L3*r3+2*A3*a^4-N2*a^2-(1/2)*M2*a^4+2*C3*log(a)+2*B3*a^2-2*V2*a+2*L3*a-
2*C3*log(r3)+T2*r3^2*log(r3)-2*d3*r3^2*log(r3)+2*S2*log(r3)-2*A3*r3^4-
(1/2)*T2*r3^2+2*d3*a^2*log(a))+pi*(h2-h1)*G*k*(2*L2*r3+T2*r2^2*log(r2)+2*S2*log(r2)-
2*d2*r2^2*log(r2)+2*d2*r3^2*log(r3)-N2*r3^2-(1/2)*M2*r3^4+2*C2*log(r3)+2*B2*r3^2-(1/2)*T2*r2^2-
2*B2*r2^2+(1/2)*M2*r2^4-2*L2*r2-2*C2*log(r2)-2*A2*r2^4+N2*r2^2+2*V2*r2-2*V2*r3+2*A2*r3^4-
T2*r3^2*log(r3)-2*S2*log(r3)+(1/2)*T2*r3^2)+lambda5/a;
        -(1/72)*pi*(h2-h1)^3*E*(6*M2*a^6*r2^2-12*V2*a^3*r2^2-
6*M2*r2^6*a^2+12*V2*r2^3*a^2+12*T2*a^4*log(a)^2*r2^2-12*T2*r2^4*log(r2)^2*a^2-
6*T2*r2^4*a^2+6*T2*a^4*r2^2+12*S2*log(r2)*a^2*r2^2+12*V2*a^3*log(a)*r2^2-12*V2*r2^3*log(r2)*a^2-
12*nu*T2*r2^4*log(r2)^2*a^2+12*nu*M2*a^6*log(a)*r2^2+12*M2*a^6*log(a)*r2^2+12*N2*a^4*log(a)*r2^2+
12*nu*T2*a^4*log(a)^2*r2^2+12*nu*S2*log(a)*a^2*r2^2-12*nu*S2*log(r2)*a^2*r2^2-
12*S2*log(a)*a^2*r2^2-12*N2*r2^4*log(r2)*a^2-12*M2*r2^6*log(r2)*a^2+12*nu*V2*a^3*log(a)*r2^2-
12*nu*N2*r2^4*log(r2)*a^2-12*nu*M2*r2^6*log(r2)*a^2-
12*nu*V2*r2^3*log(r2)*a^2+12*nu*N2*a^4*log(a)*r2^2)/((1-nu^2)*a^2*r2^2)+pi*(h2-h1)*G*k*(-
(1/2)*r3^2*S2-(1/8)*N2*r3^4+(1/8)*N2*a^4+(2/9)*a^3*V2+(1/4)*T2*a^4*log(a)-(1/4)*T2*r3^4*log(r3)-
S2*a^2*log(a)-
(1/3)*M2*a^6*log(a)+(4/3)*A3*a^6*log(a)+B3*a^4*log(a)+(2/3)*L3*a^3*log(a)+C3*a^2*log(a)+d3*a^4*lo
g(a)^2+(1/2)*a^2*S2-(1/16)*T2*a^4-(2/9)*r3^3*V2-
(1/18)*M2*r3^6+(1/18)*M2*a^6+(1/16)*T2*r3^4+(1/2)*T2*r3^4*log(r3)^2-(1/2)*T2*a^4*log(a)^2-
(2/3)*V2*a^3*log(a)+(2/3)*V2*r3^3*log(r3)-B3*r3^4*log(r3)-(2/3)*L3*r3^3*log(r3)-C3*r3^2*log(r3)-
(4/3)*A3*r3^6*log(r3)+(1/2)*N2*r3^4*log(r3)+S2*r3^2*log(r3)+(1/3)*M2*r3^6*log(r3)-
d3*r3^4*log(r3)^2-(2/9)*A3*a^6-(1/4)*B3*a^4-(2/9)*L3*a^3-
(1/2)*C3*a^2+(2/9)*L3*r3^3+(1/2)*C3*r3^2+(2/9)*A3*r3^6+(1/4)*B3*r3^4-(1/2)*N2*a^4*log(a))+pi*(h2-
h1)*G*k*((1/2)*r3^2*S2-(1/8)*N2*r2^4+(1/8)*N2*r3^4+(1/4)*T2*r3^4*log(r3)-
(1/4)*T2*r2^4*log(r2)+(2/9)*r3^3*V2+(1/18)*M2*r3^6-(1/18)*M2*r2^6+(1/16)*T2*r2^4-(1/16)*T2*r3^4-
(1/2)*r2^2*S2-(2/9)*r2^3*V2-(1/2)*T2*r3^4*log(r3)^2-
B2*r2^4*log(r2)+d2*r3^4*log(r3)^2+(2/3)*L2*r3^3*log(r3)+(2/3)*V2*r2^3*log(r2)+C2*r3^2*log(r3)+(1/
3)*M2*r2^6*log(r2)+B2*r3^4*log(r3)-(2/3)*V2*r3^3*log(r3)-(1/2)*N2*r3^4*log(r3)-S2*r3^2*log(r3)-
(1/3)*M2*r3^6*log(r3)+(4/3)*A2*r3^6*log(r3)-(2/3)*L2*r2^3*log(r2)-
(4/3)*A2*r2^6*log(r2)+(1/2)*N2*r2^4*log(r2)-d2*r2^4*log(r2)^2+S2*r2^2*log(r2)-
C2*r2^2*log(r2)+(2/9)*A2*r2^6+(1/4)*B2*r2^4+(2/9)*L2*r2^3+(1/2)*C2*r2^2-(1/4)*B2*r3^4-
(1/2)*C2*r3^2-(2/9)*L2*r3^3-(2/9)*A2*r3^6+(1/2)*T2*r2^4*log(r2)^2)+lambda5*a*log(a);
        -(1/72)*pi*(h2-h1)^3*E*(-12*nu*N2*r2^3*a^2-
12*nu*M2*r2^5*a^2+12*nu*N2*a^3*r2^2+12*nu*M2*a^5*r2^2-12*S2*a*r2^2+12*S2*a^2*r2+12*nu*S2*a*r2^2-
12*nu*S2*a^2*r2+12*T2*a^3*log(a)*r2^2-12*T2*r2^3*log(r2)*a^2+12*nu*T2*a^3*log(a)*r2^2-
12*nu*T2*r2^3*log(r2)*a^2+12*N2*a^3*r2^2-12*T2*a^3*r2^2-4*M2*r2^5*a^2-
12*N2*r2^3*a^2+4*M2*a^5*r2^2+12*T2*r2^3*a^2-12*V2*log(r2)*a^2*r2^2+12*V2*log(a)*a^2*r2^2)/((1-
nu^2)*a^2*r2^2)+pi*(h2-h1)*G*k*(-V2*a^2+(4/3)*d3*a^3*log(a)+(2/5)*M2*r3^5-(2/3)*N2*a^3-2*C3*r3-
(4/3)*B3*r3^3+2*S2*r3-L3*r3^2-(2/9)*d3*r3^3+(2/9)*T2*a^3-
(2/5)*M2*a^5+(8/5)*A3*a^5+2*C3*a+(4/3)*B3*a^3-2*S2*a+L3*a^2+(2/9)*d3*a^3-(8/5)*A3*r3^5-
(2/3)*T2*a^3*log(a)+(2/3)*T2*r3^3*log(r3)-(4/3)*d3*r3^3*log(r3)+V2*r3^2+(2/3)*N2*r3^3-
(2/9)*T2*r3^3)+pi*(h2-h1)*G*k*(-(4/3)*d2*r2^3*log(r2)+L2*r3^2+V2*r2^2+(4/3)*d2*r3^3*log(r3)-
(2/5)*M2*r3^5+(4/3)*B2*r3^3+2*C2*r3+(2/3)*T2*r2^3*log(r2)-(4/3)*B2*r2^3-(2/9)*T2*r2^3-
(2/9)*d2*r2^3-(8/5)*A2*r2^5-2*C2*r2-L2*r2^2+(2/3)*N2*r2^3+(2/5)*M2*r2^5+2*S2*r2-2*S2*r3-
(2/3)*T2*r3^3*log(r3)+(2/9)*d2*r3^3+(8/5)*A2*r3^5-V2*r3^2-(2/3)*N2*r3^3+(2/9)*T2*r3^3)+lambda5;
        (1/72)*pi*(h3-h2)^3*E*(12*T3*r3^6*log(r3)*a^2+12*nu*T3*r3^6*log(r3)*a^2-
12*N3*a^6*r3^2+12*S3*a^4*r3^2-6*T3*a^6*r3^2-4*V3*a^5*r3^2+12*N3*r3^6*a^2+12*nu*M3*r3^8*a^2-
12*nu*V3*a^5*r3^2-20*M3*a^8*r3^2-12*nu*N3*a^6*r3^2-
12*nu*S3*a^4*r3^2+12*nu*V3*r3^5*a^2+12*nu*N3*r3^6*a^2+12*nu*S3*r3^4*a^2+20*M3*r3^8*a^2-
12*S3*r3^4*a^2-12*nu*M3*a^8*r3^2+6*T3*r3^6*a^2+4*V3*r3^5*a^2-12*nu*T3*a^6*log(a)*r3^2-
12*T3*a^6*log(a)*r3^2)/((1-nu^2)*r3^2*a^2)+pi*(h3-h2)*G*k*((2/9)*d3*a^6-
A3*r3^8+(1/18)*T3*a^6+(1/3)*T3*r3^6*log(r3)+(2/3)*d3*a^6*log(a)-(1/3)*T3*a^6*log(a)-
(2/3)*d3*r3^6*log(r3)-(2/3)*B3*r3^6-(1/2)*C3*r3^4-
(2/5)*L3*r3^5+(1/3)*r3^6*N3+(1/2)*r3^4*S3+(2/5)*r3^5*V3-(1/18)*T3*r3^6-(2/9)*d3*r3^6-
(1/4)*M3*a^8+A3*a^8+(2/3)*B3*a^6+(1/2)*C3*a^4+(2/5)*L3*a^5-(1/3)*a^6*N3-(1/2)*a^4*S3-
(2/5)*a^5*V3+(1/4)*M3*r3^8)+lambda6*a^3;
        (1/72)*pi*(h3-h2)^3*E*(12*T3*r3^4*log(r3)*a^2+12*nu*T3*r3^4*log(r3)*a^2-
12*nu*T3*a^4*log(a)*r3^2-12*M3*a^6*r3^2-12*V3*a^3*r3^2+12*M3*r3^6*a^2+12*nu*N3*r3^4*a^2-
12*nu*N3*a^4*r3^2-12*N3*a^4*r3^2-12*nu*V3*a^3*r3^2-
12*nu*M3*a^6*r3^2+12*nu*V3*r3^3*a^2+12*nu*M3*r3^6*a^2+12*N3*r3^4*a^2+12*V3*r3^3*a^2-
12*T3*a^4*log(a)*r3^2)/((1-nu^2)*r3^2*a^2)+pi*(h3-h2)*G*k*(-(4/3)*A3*r3^6+d3*a^4*log(a)-
(1/2)*T3*a^4*log(a)-d3*r3^4*log(r3)+(1/2)*T3*r3^4*log(r3)-B3*r3^4-C3*r3^2-
```

```
(2/3)*L3*r3^3+(1/3)*M3*r3^6+r3^2*S3+(2/3)*r3^3*V3-(1/8)*T3*r3^4-(1/4)*d3*r3^4-
(1/2)*N3*a^4+(4/3)*A3*a^6+B3*a^4+C3*a^2+(2/3)*L3*a^3-(1/3)*M3*a^6-a^2*S3-
(2/3)*a^3*V3+(1/8)*T3*a^4+(1/4)*d3*a^4+(1/2)*N3*r3^4)+lambda6*a;
        (1/72)*pi*(h3-h2)^3*E*(12*V3*r3^2*a-12*nu*T3*log(a)*r3^2*a^2+12*nu*T3*log(r3)*r3^2*a^2-
12*nu*S3*r3^2+12*M3*a^4*r3^2-12*a^2*S3+12*r3^2*S3-12*nu*M3*a^4*r3^2+12*nu*M3*r3^4*a^2-
12*V3*r3*a^2+12*nu*S3*a^2-12*nu*V3*r3^2*a+12*nu*V3*r3*a^2+12*T3*log(a)*r3^2*a^2-12*M3*r3^4*a^2-
12*T3*log(r3)*r3^2*a^2)/((1-nu^2)*r3^2*a^2)+pi*(h3-h2)*G*k*(-2*d3*r3^2*log(r3)-2*A3*r3^4-
T3*a^2*log(a)+2*d3*a^2*log(a)+T3*r3^2*log(r3)-2*B3*r3^2-(1/2)*T3*r3^2-
2*L3*r3+(1/2)*M3*r3^4+N3*r3^2+2*V3*r3-2*C3*log(r3)-
2*S3*log(a)+2*A3*a^4+2*B3*a^2+(1/2)*T3*a^2+2*L3*a-(1/2)*M3*a^4-N3*a^2-
2*V3*a+2*C3*log(a)+2*S3*log(r3))+lambda6/a;
        (1/72)*pi*(h3-h2)^3*E*(6*M3*r3^6*a^2+12*V3*a^3*r3^2-12*V3*r3^3*a^2-
12*T3*a^4*log(a)^2*r3^2+12*T3*r3^4*log(r3)^2*a^2-
12*nu*S3*log(a)*r3^2*a^2+12*nu*S3*log(r3)*r3^2*a^2+12*nu*V3*r3^3*log(r3)*a^2+12*nu*M3*r3^6*log(r3
)*a^2-
12*nu*V3*a^3*log(a)*r3^2+12*nu*N3*r3^4*log(r3)*a^2+12*S3*log(a)*r3^2*a^2+12*M3*r3^6*log(r3)*a^2+1
2*V3*r3^3*log(r3)*a^2-6*M3*a^6*r3^2+12*N3*r3^4*log(r3)*a^2-
12*nu*N3*a^4*log(a)*r3^2+12*nu*T3*r3^4*log(r3)^2*a^2-12*S3*log(r3)*r3^2*a^2-
12*nu*M3*a^6*log(a)*r3^2-12*V3*a^3*log(a)*r3^2-12*N3*a^4*log(a)*r3^2-12*nu*T3*a^4*log(a)^2*r3^2-
12*M3*a^6*log(a)*r3^2-6*T3*a^4*r3^2+6*T3*r3^4*a^2)/((1-nu^2)*r3^2*a^2)+pi*(h3-h2)*G*k*(-
(1/2)*r3^2*S3-(1/2)*C3*a^2+(1/2)*a^2*S3-(1/18)*M3*r3^6+(1/18)*M3*a^6+(2/9)*a^3*V3+(2/9)*A3*r3^6-
(1/4)*B3*a^4+(1/4)*B3*r3^4+(1/4)*T3*a^4*log(a)-(1/4)*T3*r3^4*log(r3)-
(1/8)*N3*r3^4+(1/8)*N3*a^4+(2/3)*V3*r3^3*log(r3)-d3*r3^4*log(r3)^2+(1/3)*M3*r3^6*log(r3)-
C3*r3^2*log(r3)+(2/3)*L3*a^3*log(a)-(1/2)*N3*a^4*log(a)-S3*a^2*log(a)+C3*a^2*log(a)-
(1/3)*M3*a^6*log(a)+(4/3)*A3*a^6*log(a)+B3*a^4*log(a)+d3*a^4*log(a)^2-(2/3)*V3*a^3*log(a)-
(2/9)*L3*a^3-(1/16)*T3*a^4-(2/9)*A3*a^6+S3*r3^2*log(r3)+(1/16)*T3*r3^4+(1/2)*T3*r3^4*log(r3)^2-
(1/2)*T3*a^4*log(a)^2-(2/3)*L3*r3^3*log(r3)+(1/2)*N3*r3^4*log(r3)-B3*r3^4*log(r3)-
(4/3)*A3*r3^6*log(r3)+(1/2)*C3*r3^2-(2/9)*r3^3*V3+(2/9)*L3*r3^3)+lambda6*a*log(a);
        (1/72)*pi*(h3-h2)^3*E*(12*S3*r3^2*a+12*T3*r3^3*log(r3)*a^2+12*nu*T3*r3^3*log(r3)*a^2-
12*nu*T3*a^3*log(a)*r3^2-4*M3*a^5*r3^2-12*N3*a^3*r3^2+12*T3*a^3*r3^2-12*nu*M3*a^5*r3^2-
12*nu*N3*a^3*r3^2+12*nu*M3*r3^5*a^2+12*nu*N3*r3^3*a^2-12*S3*r3*a^2-
12*nu*S3*r3^2*a+12*nu*S3*r3*a^2+12*V3*log(r3)*r3^2*a^2-
12*V3*log(a)*r3^2*a^2+4*M3*r3^5*a^2+12*N3*r3^3*a^2-12*T3*r3^3*a^2-12*T3*a^3*log(a)*r3^2)/((1-
nu^2)*r3^2*a^2)+pi*(h3-h2)*G*k*((2/3)*T3*r3^3*log(r3)+(2/9)*d3*a^3+(4/3)*d3*a^3*log(a)-
(2/3)*T3*a^3*log(a)-(4/3)*d3*r3^3*log(r3)-(4/3)*B3*r3^3-2*C3*r3-
L3*r3^2+(2/5)*M3*r3^5+(2/3)*N3*r3^3+2*S3*r3-(2/9)*T3*r3^3-(2/9)*d3*r3^3-V3*a^2-
(8/5)*A3*r3^5+(2/9)*T3*a^3+(8/5)*A3*a^5+(4/3)*B3*a^3+2*C3*a+L3*a^2-(2/5)*M3*a^5-(2/3)*N3*a^3-
2*S3*a+V3*r3^2)+lambda6;
        A1*r2^4+B1*r2^2+C1*log(r2)+d1*r2^2*log(r2)+L1*r2+H1-A2*r2^4-B2*r2^2-C2*log(r2)-
d2*r2^2*log(r2)-L2*r2-H2;
        A2*r3^4+B2*r3^2+C2*log(r3)+d2*r3^2*log(r3)+L2*r3+H2-A3*r3^4-B3*r3^2-C3*log(r3)-
d3*r3^2*log(r3)-L3*r3-H3;
        A3*a^4+B3*a^2+C3*log(a)+d3*a^2*log(a)+L3*a+H3;
        M1*a^3+N1*a+S1/a+T1*a*log(a)+V1;
        M2*a^3+N2*a+S2/a+T2*a*log(a)+V2;
        M3*a^3+N3*a+S3/a+T3*a*log(a)+V3];
end
if l>=r3 && l<r2
    FF=[(1/3)*p0*pi*r2^6-(1/3)*p0*pi*r1^6+pi*(h1-h0)*G*k*(-(8/3)*d1*r2^6*log(r2)+4*A1*r1^8-
(2/9)*T1*r2^6+(8/5)*r2^5*V1+2*r2^4*S1-(8/3)*r2^6*B1-2*r2^4*C1-(8/9)*r2^6*d1-
(8/5)*r2^5*L1+r2^8*M1+(4/3)*r2^6*N1+(4/3)*T1*r2^6*log(r2)+(2/9)*T1*r1^6-(8/5)*r1^5*V1-
2*r1^4*S1+(8/3)*d1*r1^6*log(r1)-
(4/3)*T1*r1^6*log(r1)+(8/3)*r1^6*B1+2*r1^4*C1+(8/9)*r1^6*d1+(8/5)*r1^5*L1-r1^8*M1-(4/3)*r1^6*N1-
4*A1*r2^8)+lambda1*r2^4;
        (1/2)*p0*pi*r2^4-(1/2)*p0*pi*r1^4+pi*(h1-h0)*G*k*(T1*r2^4*log(r2)-
2*d1*r2^4*log(r2)+2*B1*r1^4-(1/4)*T1*r2^4+(4/3)*r2^3*V1+r2^4*N1+2*r2^2*S1+(2/3)*r2^6*M1-
(4/3)*r2^3*L1-(1/2)*r2^4*d1-2*r2^2*C1-(8/3)*A1*r2^6+2*d1*r1^4*log(r1)+(1/4)*T1*r1^4-
(4/3)*r1^3*V1-r1^4*N1-2*r1^2*S1-(2/3)*r1^6*M1+(4/3)*r1^3*L1+(1/2)*r1^4*d1+2*r1^2*C1-
T1*r1^4*log(r1)+(8/3)*A1*r1^6-2*B1*r2^4)+lambda1*r2^2;
        p0*pi*r2^2*log(r2)-(1/2)*p0*pi*r2^2+(1/2)*p0*pi*r1^2+pi*(h1-h0)*G*k*(2*C1*log(r1)-
(1/2)*T1*r2^2+2*V1*r2+N1*r2^2-2*L1*r2+(1/2)*M1*r2^4-2*B1*r2^2+2*S1*log(r2)-2*A1*r2^4-
2*C1*log(r2)+T1*r2^2*log(r2)+(1/2)*T1*r1^2-2*V1*r1-N1*r1^2+2*L1*r1-(1/2)*M1*r1^4+2*B1*r1^2-
2*S1*log(r1)+2*d1*r1^2*log(r1)-T1*r1^2*log(r1)+2*A1*r1^4-2*d1*r2^2*log(r2))-
p0*pi*r1^2*log(r1)+lambda1*log(r2);
        -(1/8)*p0*pi*r2^4+(1/8)*p0*pi*r1^4+(1/2)*p0*pi*r2^4*log(r2)+pi*(h1-h0)*G*k*(-
(2/9)*r2^3*L1-(4/3)*V1*r1^3*log(r1)+2*d1*r1^4*log(r1)^2-T1*r1^4*log(r1)^2-
(2/9)*r1^6*M1+(1/4)*r2^4*N1-(8/9)*A1*r2^6+(2/9)*r2^3*V1-(1/4)*r2^4*d1+(2/9)*r2^6*M1-
2*B1*r2^4*log(r2)+(4/3)*V1*r2^3*log(r2)-
(8/3)*A1*r2^6*log(r2)+2*S1*r2^2*log(r2)+N1*r2^4*log(r2)+T1*r2^4*log(r2)^2-N1*r1^4*log(r1)-
```

```
(4/3)*L1*r2^3*log(r2)+(2/3)*M1*r2^6*log(r2)-2*d1*r2^4*log(r2)^2-d1*r2^4*log(r2)+d1*r1^4*log(r1)-
2*C1*r2^2*log(r2)+2*B1*r1^4*log(r1)+(1/2)*B1*r1^4-(1/2)*B1*r2^4-
2*S1*r1^2*log(r1)+(4/3)*L1*r1^3*log(r1)-
(2/3)*M1*r1^6*log(r1)+(8/3)*A1*r1^6*log(r1)+2*C1*r1^2*log(r1)+(8/9)*A1*r1^6-
(2/9)*r1^3*V1+(1/4)*r1^4*d1+(2/9)*r1^3*L1-(1/4)*r1^4*N1)-
(1/2)*p0*pi*r1^4*log(r1)+lambda1*r2^2*log(r2);
        (2/3)*p0*pi*r2^3-(2/3)*p0*pi*r1^3+pi*(h1-h0)*G*k*((2/3)*T1*r2^3*log(r2)+L1*r1^2-2*C1*r2-
(4/3)*B1*r2^3-(2/9)*T1*r2^3-(2/9)*d1*r2^3+(2/5)*r2^5*M1+(2/3)*r2^3*N1+2*r2*S1+r2^2*V1-
(8/5)*A1*r2^5-(4/3)*d1*r2^3*log(r2)-
(2/3)*T1*r1^3*log(r1)+2*C1*r1+(4/3)*B1*r1^3+(2/9)*T1*r1^3+(2/9)*d1*r1^3-
(2/5)*r1^5*M1+(4/3)*d1*r1^3*log(r1)-(2/3)*r1^3*N1-2*r1*S1-r1^2*V1+(8/5)*A1*r1^5-
L1*r2^2)+lambda1*r2;
        p0*pi*r2^2-p0*pi*r1^2+lambda1;
        pi*(h2-h1)*G*k*(-2*r2^4*S2-(8/5)*r2^5*V2-4*A2*r3^8+2*r2^4*C2-
r2^8*M2+(2/9)*T2*r2^6+(8/3)*r2^6*B2-(8/3)*r3^6*B2+(8/5)*r2^5*L2+(8/9)*r2^6*d2-(4/3)*r2^6*N2-
(8/3)*d2*r3^6*log(r3)+(4/3)*T2*r3^6*log(r3)-(4/3)*T2*r2^6*log(r2)+(8/3)*d2*r2^6*log(r2)-
(8/5)*r3^5*L2-(8/9)*r3^6*d2+(4/3)*r3^6*N2+4*A2*r2^8+2*r3^4*S2+(8/5)*r3^5*V2-2*r3^4*C2+r3^8*M2-
(2/9)*T2*r3^6)+pi*(h1-h0)*G*k*(-(2/9)*T1*r3^6+r3^8*M1+2*r3^4*S1+(4/3)*r3^6*N1+(8/5)*r3^5*V1-
(4/3)*r2^6*N1-(8/5)*r2^5*V1+(2/9)*T1*r2^6-r2^8*M1-2*r2^4*S1+(4/3)*T1*r3^6*log(r3)-
(4/3)*T1*r2^6*log(r2)-4*A2*r3^8+2*r2^4*C2+(8/3)*r2^6*B2-
(8/3)*r3^6*B2+(8/5)*r2^5*L2+(8/9)*r2^6*d2-(8/3)*d2*r3^6*log(r3)+(8/3)*d2*r2^6*log(r2)-
(8/5)*r3^5*L2-(8/9)*r3^6*d2+4*A2*r2^8-2*r3^4*C2)+(1/3)*p0*pi*l^6-(1/3)*p0*pi*r2^6-
lambda1*r2^4+lambda2*r3^4;
        pi*(h2-h1)*G*k*(2*r2^2*C2-2*B2*r3^4+(1/2)*r2^4*d2+(4/3)*r2^3*L2-(2/3)*r2^6*M2-r2^4*N2-
2*r2^2*S2-(4/3)*r2^3*V2+(1/4)*T2*r2^4+(8/3)*A2*r2^6-(1/4)*T2*r3^4-(8/3)*A2*r3^6-
2*d2*r3^4*log(r3)-T2*r2^4*log(r2)+2*d2*r2^4*log(r2)+T2*r3^4*log(r3)+2*B2*r2^4-2*r3^2*C2-
(1/2)*r3^4*d2-(4/3)*r3^3*L2+(2/3)*r3^6*M2+r3^4*N2+2*r3^2*S2+(4/3)*r3^3*V2)+pi*(h1-
h0)*G*k*((2/3)*r3^6*M1+(4/3)*r3^3*V1+2*r3^2*S1-(1/4)*T1*r3^4+r3^4*N1-r2^4*N1-(2/3)*r2^6*M1-
2*r2^2*S1+(1/4)*T1*r2^4-(4/3)*r2^3*V1+T1*r3^4*log(r3)-T1*r2^4*log(r2)+2*r2^2*C2-
2*B2*r3^4+(1/2)*r2^4*d2+(4/3)*r2^3*L2+(8/3)*A2*r2^6-(8/3)*A2*r3^6-
2*d2*r3^4*log(r3)+2*d2*r2^4*log(r2)+2*B2*r2^4-2*r3^2*C2-(1/2)*r3^4*d2-
(4/3)*r3^3*L2)+(1/2)*p0*pi*l^4-(1/2)*p0*pi*r2^4-lambda1*r2^2+lambda2*r3^2;
        pi*(h2-h1)*G*k*(2*B2*r2^2-2*C2*log(r3)+2*A2*r2^4+2*L2*r2-(1/2)*M2*r2^4-N2*r2^2-
2*S2*log(r2)+(1/2)*T2*r2^2-2*V2*r2+2*d2*r2^2*log(r2)-T2*r2^2*log(r2)-
2*d2*r3^2*log(r3)+T2*r3^2*log(r3)+2*V2*r3+2*C2*log(r2)-2*B2*r3^2-2*A2*r3^4-
2*L2*r3+(1/2)*M2*r3^4+N2*r3^2+2*S2*log(r3)-(1/2)*T2*r3^2)+pi*(h1-h0)*G*k*((1/2)*T1*r2^2-
2*V1*r2+(1/2)*M1*r3^4+2*V1*r3-(1/2)*T1*r3^2+2*S1*log(r3)+N1*r3^2-N1*r2^2-(1/2)*M1*r2^4-
2*S1*log(r2)+T1*r3^2*log(r3)-T1*r2^2*log(r2)+2*B2*r2^2-
2*C2*log(r3)+2*A2*r2^4+2*L2*r2+2*d2*r2^2*log(r2)-2*d2*r3^2*log(r3)+2*C2*log(r2)-2*B2*r3^2-
2*A2*r3^4-2*L2*r3)-(1/2)*p0*pi*l^2+(1/2)*p0*pi*r2^2+p0*pi*l^2*log(l)-p0*pi*r2^2*log(r2)-
lambda1*log(r2)+lambda2*log(r3);
        pi*(h2-h1)*G*k*((2/9)*r2^3*L2+(1/4)*r3^4*N2-(1/4)*r3^4*d2-
(1/4)*r2^4*N2+(1/4)*r2^4*d2+(2/9)*r3^6*M2-(2/9)*r3^3*L2+(8/9)*A2*r2^6-
(2/9)*r2^6*M2+(1/2)*B2*r2^4-(8/9)*A2*r3^6+(2/9)*r3^3*V2-(1/2)*B2*r3^4+d2*r2^4*log(r2)-
d2*r3^4*log(r3)+(4/3)*L2*r2^3*log(r2)-(4/3)*V2*r2^3*log(r2)-2*B2*r3^4*log(r3)-
(8/3)*A2*r3^6*log(r3)+2*B2*r2^4*log(r2)-(2/9)*r2^3*V2+(8/3)*A2*r2^6*log(r2)+2*C2*r2^2*log(r2)-
T2*r2^4*log(r2)^2-2*S2*r2^2*log(r2)-N2*r2^4*log(r2)-
(2/3)*M2*r2^6*log(r2)+T2*r3^4*log(r3)^2+(2/3)*M2*r3^6*log(r3)+2*S2*r3^2*log(r3)-
2*C2*r3^2*log(r3)-
(4/3)*L2*r3^3*log(r3)+N2*r3^4*log(r3)+(4/3)*V2*r3^3*log(r3)+2*d2*r2^4*log(r2)^2-
2*d2*r3^4*log(r3)^2)+pi*(h1-h0)*G*k*(-(2/9)*r2^6*M1+(2/9)*r2^3*L2-(1/4)*r3^4*d2+(1/4)*r2^4*d2-
(2/9)*r3^3*L2+(1/4)*r3^4*N1+(8/9)*A2*r2^6+(1/2)*B2*r2^4-(8/9)*A2*r3^6-
(1/2)*B2*r3^4+(2/9)*r3^6*M1+d2*r2^4*log(r2)-d2*r3^4*log(r3)+(4/3)*L2*r2^3*log(r2)-
2*B2*r3^4*log(r3)-
(8/3)*A2*r3^6*log(r3)+2*B2*r2^4*log(r2)+(8/3)*A2*r2^6*log(r2)+2*C2*r2^2*log(r2)+(2/9)*r3^3*V1-
2*C2*r3^2*log(r3)-(4/3)*L2*r3^3*log(r3)+T1*r3^4*log(r3)^2+N1*r3^4*log(r3)-(2/3)*M1*r2^6*log(r2)-
N1*r2^4*log(r2)-T1*r2^4*log(r2)^2-2*S1*r2^2*log(r2)-
(4/3)*V1*r2^3*log(r2)+(2/3)*M1*r3^6*log(r3)+2*S1*r3^2*log(r3)+(4/3)*V1*r3^3*log(r3)-
(2/9)*r2^3*V1+2*d2*r2^4*log(r2)^2-(1/4)*r2^4*N1-2*d2*r3^4*log(r3)^2)-
(1/8)*p0*pi*l^4+(1/8)*p0*pi*r2^4+(1/2)*p0*pi*l^4*log(l)-(1/2)*p0*pi*r2^4*log(r2)-
lambda1*r2^2*log(r2)+lambda2*r3^2*log(r3);
        pi*(h2-h1)*G*k*(-r2^2*V2-(2/3)*r2^3*N2-2*r2*S2-
L2*r3^2+2*C2*r2+(4/3)*B2*r2^3+(2/9)*T2*r2^3-(2/9)*T2*r3^3-
(2/5)*r2^5*M2+(8/5)*A2*r2^5+(2/9)*d2*r2^3-(2/3)*T2*r2^3*log(r2)+(4/3)*d2*r2^3*log(r2)-
(8/5)*A2*r3^5-(4/3)*d2*r3^3*log(r3)+(2/3)*T2*r3^3*log(r3)-
(2/9)*d2*r3^3+(2/5)*r3^5*M2+L2*r2^2+r3^2*V2+(2/3)*r3^3*N2+2*r3*S2-2*C2*r3-(4/3)*B2*r3^3)+pi*(h1-
h0)*G*k*(-(2/3)*r2^3*N1-2*r2*S1+(2/3)*T1*r3^3*log(r3)-
(2/9)*T1*r3^3+r3^2*V1+(2/5)*r3^5*M1+(2/3)*r3^3*N1+2*r3*S1-r2^2*V1-(2/5)*r2^5*M1+(2/9)*T1*r2^3-
(2/3)*T1*r2^3*log(r2)-
```

```
L2*r3^2+2*C2*r2+(4/3)*B2*r2^3+(8/5)*A2*r2^5+(2/9)*d2*r2^3+(4/3)*d2*r2^3*log(r2)-(8/5)*A2*r3^5-
(4/3)*d2*r3^3*log(r3)-(2/9)*d2*r3^3+L2*r2^2-2*C2*r3-(4/3)*B2*r3^3)+(2/3)*p0*pi*l^3-
(2/3)*p0*pi*r2^3-lambda1*r2+lambda2*r3;
        p0*pi*l^2-p0*pi*r2^2-lambda1+lambda2;
        pi*(h3-h2)*G*k*(4*A3*r3^8+(8/3)*d3*r3^6*log(r3)+2*a^4*S3+a^8*M3-(8/5)*a^5*L3-
(8/9)*a^6*d3-2*a^4*C3-(8/3)*a^6*B3+(8/5)*a^5*V3+(4/3)*a^6*N3-
(2/9)*T3*a^6+(8/9)*r3^6*d3+2*r3^4*C3+(8/3)*r3^6*B3-(8/5)*r3^5*V3-2*r3^4*S3-(4/3)*r3^6*N3-
r3^8*M3+(8/5)*r3^5*L3+(2/9)*T3*r3^6-4*A3*a^8-(4/3)*T3*r3^6*log(r3)-
(8/3)*d3*a^6*log(a)+(4/3)*T3*a^6*log(a))+pi*(h2-
h1)*G*k*((8/5)*a^5*V2+2*a^4*S2+a^8*M2+(4/3)*a^6*N2-(8/5)*r3^5*V2-2*r3^4*S2-r3^8*M2-
(4/3)*r3^6*N2+(4/3)*T2*a^6*log(a)-(2/9)*T2*a^6+(2/9)*T2*r3^6-
(4/3)*T2*r3^6*log(r3)+4*A3*r3^8+(8/3)*d3*r3^6*log(r3)-(8/5)*a^5*L3-(8/9)*a^6*d3-2*a^4*C3-
(8/3)*a^6*B3+(8/9)*r3^6*d3+2*r3^4*C3+(8/3)*r3^6*B3+(8/5)*r3^5*L3-4*A3*a^8-
(8/3)*d3*a^6*log(a))+pi*(h1-h0)*G*k*((4/3)*T1*a^6*log(a)+4*A3*r3^8-
(4/3)*T1*r3^6*log(r3)+(8/3)*d3*r3^6*log(r3)-(8/5)*a^5*L3-(8/9)*a^6*d3-2*a^4*C3-
(8/3)*a^6*B3+(8/5)*a^5*V1+2*a^4*S1+(8/9)*r3^6*d3+2*r3^4*C3+(8/3)*r3^6*B3+(8/5)*r3^5*L3-4*A3*a^8-
(8/5)*r3^5*V1-2*r3^4*S1+(2/9)*T1*r3^6-r3^8*M1-(4/3)*r3^6*N1-(8/3)*d3*a^6*log(a)+(4/3)*a^6*N1-
(2/9)*T1*a^6+a^8*M1)+lambda3*a^4-lambda2*r3^4;
        pi*(h3-h2)*G*k*(-(1/2)*a^4*d3+a^4*N3+2*B3*r3^4+2*a^2*S3-T3*r3^4*log(r3)+(4/3)*a^3*V3-
(8/3)*A3*a^6-(1/4)*T3*a^4+(8/3)*A3*r3^6+2*d3*r3^4*log(r3)+(1/2)*r3^4*d3+2*r3^2*C3+(1/4)*T3*r3^4-
(2/3)*r3^6*M3-r3^4*N3-2*r3^2*S3-(4/3)*r3^3*V3+(4/3)*r3^3*L3+(2/3)*a^6*M3-2*B3*a^4+T3*a^4*log(a)-
2*d3*a^4*log(a)-(4/3)*a^3*L3-2*a^2*C3)+pi*(h2-h1)*G*k*(a^4*N2-r3^4*N2-
(1/4)*T2*a^4+T2*a^4*log(a)+2*a^2*S2+(4/3)*a^3*V2+(2/3)*a^6*M2-(1/2)*a^4*d3+(1/4)*T2*r3^4-
2*r3^2*S2-(4/3)*r3^3*V2-(2/3)*r3^6*M2-T2*r3^4*log(r3)+2*B3*r3^4-
(8/3)*A3*a^6+(8/3)*A3*r3^6+2*d3*r3^4*log(r3)+(1/2)*r3^4*d3+2*r3^2*C3+(4/3)*r3^3*L3-2*B3*a^4-
2*d3*a^4*log(a)-(4/3)*a^3*L3-2*a^2*C3)+pi*(h1-h0)*G*k*(-(1/2)*a^4*d3+2*B3*r3^4+2*a^2*S1-
(8/3)*A3*a^6+(8/3)*A3*r3^6+2*d3*r3^4*log(r3)+(1/2)*r3^4*d3+2*r3^2*C3+(4/3)*r3^3*L3-2*B3*a^4-
r3^4*N1-(2/3)*r3^6*M1+a^4*N1+(1/4)*T1*r3^4-(4/3)*r3^3*V1+T1*a^4*log(a)+(2/3)*a^6*M1-
T1*r3^4*log(r3)-2*d3*a^4*log(a)-(4/3)*a^3*L3-2*a^2*C3-(1/4)*T1*a^4+(4/3)*a^3*V1-
2*r3^2*S1)+lambda3*a^2-lambda2*r3^2;
        pi*(h3-h2)*G*k*(-2*C3*log(a)+2*C3*log(r3)-T3*r3^2*log(r3)-
2*A3*a^4+2*V3*a+N3*a^2+2*d3*r3^2*log(r3)+(1/2)*M3*a^4-(1/2)*T3*a^2+2*S3*log(a)-
2*L3*a+2*A3*r3^4+2*B3*r3^2-N3*r3^2-(1/2)*M3*r3^4-2*V3*r3+(1/2)*T3*r3^2-
2*S3*log(r3)+2*L3*r3+T3*a^2*log(a)-2*d3*a^2*log(a)-2*B3*a^2)+pi*(h2-
h1)*G*k*(2*S2*log(a)+2*V2*a+(1/2)*M2*a^4-(1/2)*T2*a^2-2*S2*log(r3)-2*V2*r3-
(1/2)*M2*r3^4+(1/2)*T2*r3^2+T2*a^2*log(a)+N2*a^2-N2*r3^2-T2*r3^2*log(r3)-
2*C3*log(a)+2*C3*log(r3)-2*A3*a^4+2*d3*r3^2*log(r3)-2*L3*a+2*A3*r3^4+2*B3*r3^2+2*L3*r3-
2*d3*a^2*log(a)-2*B3*a^2)+pi*(h1-h0)*G*k*(-2*C3*log(a)+2*C3*log(r3)+2*V1*a-T1*r3^2*log(r3)-
2*A3*a^4+2*d3*r3^2*log(r3)-2*L3*a+2*S1*log(a)+T1*a^2*log(a)+2*A3*r3^4+2*B3*r3^2+2*L3*r3-
2*S1*log(r3)+(1/2)*T1*r3^2-N1*r3^2-(1/2)*M1*r3^4-2*d3*a^2*log(a)-2*B3*a^2-
(1/2)*T1*a^2+N1*a^2+(1/2)*M1*a^4-2*V1*r3)+lambda3*log(a)-lambda2*log(r3);
        pi*(h3-h2)*G*k*(2*B3*r3^4*log(r3)+2*C3*r3^2*log(r3)-T3*r3^4*log(r3)^2-
(8/3)*A3*a^6*log(a)-(2/3)*M3*r3^6*log(r3)-N3*r3^4*log(r3)-2*S3*r3^2*log(r3)-
(4/3)*V3*r3^3*log(r3)+(4/3)*V3*a^3*log(a)-2*C3*a^2*log(a)-
(4/3)*L3*a^3*log(a)+T3*a^4*log(a)^2+2*S3*a^2*log(a)+(2/3)*M3*a^6*log(a)+N3*a^4*log(a)-
2*B3*a^4*log(a)+(8/3)*A3*r3^6*log(r3)+(4/3)*L3*r3^3*log(r3)-(2/9)*a^3*L3-
d3*a^4*log(a)+d3*r3^4*log(r3)+(2/9)*a^6*M3-2*d3*a^4*log(a)^2+2*d3*r3^4*log(r3)^2-
(1/4)*r3^4*N3+(1/4)*a^4*N3-(8/9)*A3*a^6+(1/2)*B3*r3^4-(1/2)*B3*a^4+(1/4)*r3^4*d3+(8/9)*A3*r3^6-
(1/4)*a^4*d3-(2/9)*r3^3*V3+(2/9)*r3^3*L3-(2/9)*r3^6*M3+(2/9)*a^3*V3)+pi*(h2-
h1)*G*k*(2*B3*r3^4*log(r3)+2*C3*r3^2*log(r3)-(8/3)*A3*a^6*log(a)-2*C3*a^2*log(a)-
(4/3)*L3*a^3*log(a)+N2*a^4*log(a)+T2*a^4*log(a)^2-
2*B3*a^4*log(a)+(4/3)*V2*a^3*log(a)+(2/3)*M2*a^6*log(a)-(4/3)*V2*r3^3*log(r3)-N2*r3^4*log(r3)-
2*S2*r3^2*log(r3)-(2/3)*M2*r3^6*log(r3)-T2*r3^4*log(r3)^2+2*S2*a^2*log(a)-
(2/9)*r3^3*V2+(2/9)*a^6*M2+(8/3)*A3*r3^6*log(r3)+(4/3)*L3*r3^3*log(r3)-(2/9)*a^3*L3-
d3*a^4*log(a)+d3*r3^4*log(r3)-2*d3*a^4*log(a)^2+2*d3*r3^4*log(r3)^2-(8/9)*A3*a^6+(1/2)*B3*r3^4-
(1/2)*B3*a^4-(2/9)*r3^6*M2+(1/4)*r3^4*d3+(8/9)*A3*r3^6-(1/4)*a^4*d3+(1/4)*a^4*N2-
(1/4)*r3^4*N2+(2/9)*r3^3*L3+(2/9)*a^3*V2)+pi*(h1-h0)*G*k*(2*B3*r3^4*log(r3)+2*C3*r3^2*log(r3)-
(1/4)*r3^4*N1-(8/3)*A3*a^6*log(a)-2*C3*a^2*log(a)-(4/3)*L3*a^3*log(a)-
2*B3*a^4*log(a)+(8/3)*A3*r3^6*log(r3)+(4/3)*L3*r3^3*log(r3)-
(4/3)*V1*r3^3*log(r3)+T1*a^4*log(a)^2-
2*S1*r3^2*log(r3)+(4/3)*V1*a^3*log(a)+2*S1*a^2*log(a)+N1*a^4*log(a)+(2/3)*M1*a^6*log(a)-
(2/3)*M1*r3^6*log(r3)-T1*r3^4*log(r3)^2-N1*r3^4*log(r3)-(2/9)*r3^6*M1+(2/9)*a^3*V1-(2/9)*a^3*L3-
d3*a^4*log(a)+d3*r3^4*log(r3)-2*d3*a^4*log(a)^2+2*d3*r3^4*log(r3)^2-(8/9)*A3*a^6+(1/2)*B3*r3^4-
(1/2)*B3*a^4-(2/9)*r3^3*V1+(1/4)*r3^4*d3+(8/9)*A3*r3^6-
(1/4)*a^4*d3+(2/9)*r3^3*L3+(2/9)*a^6*M1+(1/4)*a^4*N1)+lambda3*a^2*log(a)-lambda2*r3^2*log(r3);
        pi*(h3-h2)*G*k*(L3*r3^2+(4/3)*d3*r3^3*log(r3)+a^2*V3+2*a*S3+(2/3)*a^3*N3-(8/5)*A3*a^5-
(2/9)*T3*a^3-(2/9)*d3*a^3+(2/5)*a^5*M3-2*C3*a-r3^2*V3-2*r3*S3-(2/3)*r3^3*N3+(8/5)*A3*r3^5-
(2/3)*T3*r3^3*log(r3)+(2/9)*T3*r3^3+(2/9)*d3*r3^3+(4/3)*B3*r3^3-(2/5)*r3^5*M3+2*C3*r3-L3*a^2-
(4/3)*d3*a^3*log(a)-(4/3)*B3*a^3+(2/3)*T3*a^3*log(a))+pi*(h2-h1)*G*k*(-
```

```
(2/9)*T2*a^3+(2/3)*T2*a^3*log(a)+(2/9)*T2*r3^3-
(2/3)*T2*r3^3*log(r3)+(2/3)*a^3*N2+2*a*S2+a^2*V2+(2/5)*a^5*M2-(2/3)*r3^3*N2-2*r3*S2-r3^2*V2-
(2/5)*r3^5*M2+L3*r3^2+(4/3)*d3*r3^3*log(r3)-(8/5)*A3*a^5-(2/9)*d3*a^3-
2*C3*a+(8/5)*A3*r3^5+(2/9)*d3*r3^3+(4/3)*B3*r3^3+2*C3*r3-L3*a^2-(4/3)*d3*a^3*log(a)-
(4/3)*B3*a^3)+pi*(h1-h0)*G*k*(L3*r3^2+(4/3)*d3*r3^3*log(r3)-(8/5)*A3*a^5-(2/9)*d3*a^3-
2*C3*a+(2/5)*a^5*M1+(8/5)*A3*r3^5+(2/9)*d3*r3^3+(4/3)*B3*r3^3+2*C3*r3-L3*a^2-(2/5)*r3^5*M1-
(2/3)*r3^3*N1-2*r3*S1-r3^2*V1-
(2/3)*T1*r3^3*log(r3)+(2/3)*a^3*N1+2*a*S1+a^2*V1+(2/3)*T1*a^3*log(a)+(2/9)*T1*r3^3-
(4/3)*d3*a^3*log(a)-(4/3)*B3*a^3-(2/9)*T1*a^3)+lambda3*a-lambda2*r3;
        lambda3-lambda2;
        (1/72)*pi*(h1-h0)^3*E*(20*M1*r1^8*a^2-20*M1*a^8*r1^2-12*nu*N1*a^6*r1^2-12*nu*V1*a^5*r1^2-
12*nu*S1*a^4*r1^2-
12*nu*T1*a^6*log(a)*r1^2+12*nu*V1*r1^5*a^2+12*nu*N1*r1^6*a^2+12*nu*S1*r1^4*a^2+12*T1*r1^6*log(r1)
*a^2+12*nu*T1*r1^6*log(r1)*a^2-12*T1*a^6*log(a)*r1^2-12*nu*M1*a^8*r1^2-6*T1*a^6*r1^2-
4*V1*a^5*r1^2-12*N1*a^6*r1^2+12*S1*a^4*r1^2-
12*S1*r1^4*a^2+6*T1*r1^6*a^2+12*nu*M1*r1^8*a^2+12*N1*r1^6*a^2+4*V1*r1^5*a^2)/((1-
nu^2)*r1^2*a^2)+pi*(h1-h0)*G*k*((2/9)*d3*a^6+(2/3)*d3*a^6*log(a)-
(1/4)*M1*a^8+(2/5)*L3*a^5+(1/3)*T1*r3^6*log(r3)+(1/4)*M1*r3^8-
(2/5)*L3*r3^5+(1/3)*r3^6*N1+(1/2)*r3^4*S1+(2/5)*r3^5*V1-(2/3)*B3*r3^6-(1/3)*a^6*N1-(1/2)*a^4*S1-
(2/5)*a^5*V1-A3*r3^8-(1/2)*C3*r3^4-(2/3)*d3*r3^6*log(r3)-(1/3)*T1*a^6*log(a)-
(1/18)*T1*r3^6+(2/3)*B3*a^6+(1/18)*T1*a^6+A3*a^8+(1/2)*C3*a^4-(2/9)*d3*r3^6)+pi*(h1-h0)*G*k*(-
(2/9)*d2*r2^6+(1/2)*C2*r3^4+(2/3)*B2*r3^6+A2*r3^8+(2/5)*L2*r3^5+(2/9)*d2*r3^6-
(2/5)*L2*r2^5+(2/5)*r2^5*V1+(1/2)*r2^4*S1+(1/3)*r2^6*N1-(1/2)*C2*r2^4-(2/3)*B2*r2^6-
(1/18)*T1*r2^6-A2*r2^8+(2/3)*d2*r3^6*log(r3)+(1/4)*M1*r2^8+(1/3)*T1*r2^6*log(r2)-
(2/3)*d2*r2^6*log(r2)-(1/3)*T1*r3^6*log(r3)-(1/4)*M1*r3^8-(1/3)*r3^6*N1-(1/2)*r3^4*S1-
(2/5)*r3^5*V1+(1/18)*T1*r3^6)+pi*(h1-h0)*G*k*(-(2/5)*r2^5*V1-(1/2)*r2^4*S1-
(1/3)*r2^6*N1+(1/18)*T1*r2^6-(1/4)*M1*r2^8-
(1/3)*T1*r2^6*log(r2)+(1/4)*M1*r1^8+(2/5)*r1^5*V1+(1/2)*C1*r2^4+(2/3)*B1*r2^6+(2/9)*d1*r2^6+(2/5)
*L1*r2^5+A1*r2^8+(2/3)*d1*r2^6*log(r2)+(1/3)*r1^6*N1-(1/2)*C1*r1^4-(2/3)*B1*r1^6-(1/18)*T1*r1^6-
(2/9)*d1*r1^6-(2/5)*L1*r1^5+(1/2)*r1^4*S1+(1/3)*T1*r1^6*log(r1)-(2/3)*d1*r1^6*log(r1)-
A1*r1^8)+lambda4*a^3;
        (1/72)*pi*(h1-h0)^3*E*(12*N1*r1^4*a^2-12*N1*a^4*r1^2-12*nu*M1*a^6*r1^2-
12*nu*V1*a^3*r1^2+12*nu*M1*r1^6*a^2+12*nu*V1*r1^3*a^2+12*T1*r1^4*log(r1)*a^2-
12*nu*T1*a^4*log(a)*r1^2+12*nu*T1*r1^4*log(r1)*a^2-12*T1*a^4*log(a)*r1^2-12*nu*N1*a^4*r1^2-
12*M1*a^6*r1^2-12*V1*a^3*r1^2+12*M1*r1^6*a^2+12*nu*N1*r1^4*a^2+12*V1*r1^3*a^2)/((1-
nu^2)*r1^2*a^2)+pi*(h1-h0)*G*k*((1/2)*T1*r3^4*log(r3)+d3*a^4*log(a)-(1/2)*N1*a^4-
(1/2)*T1*a^4*log(a)+(1/4)*d3*a^4+(1/2)*N1*r3^4-(2/3)*L3*r3^3+(1/3)*M1*r3^6+r3^2*S1+(2/3)*r3^3*V1-
B3*r3^4+(2/3)*L3*a^3-(1/3)*M1*a^6-a^2*S1-(2/3)*a^3*V1+B3*a^4-(1/8)*T1*r3^4-C3*r3^2-
(4/3)*A3*r3^6+(4/3)*A3*a^6+(1/8)*T1*a^4-d3*r3^4*log(r3)+C3*a^2-(1/4)*d3*r3^4)+pi*(h1-h0)*G*k*(-
(2/3)*L2*r2^3-
(1/4)*d2*r2^4+d2*r3^4*log(r3)+(4/3)*A2*r3^6+B2*r3^4+C2*r3^2+(2/3)*L2*r3^3+(1/4)*d2*r3^4-
(4/3)*A2*r2^6-B2*r2^4-C2*r2^2+(2/3)*r2^3*V1+r2^2*S1+(1/3)*M1*r2^6-
(1/8)*T1*r2^4+(1/2)*T1*r2^4*log(r2)+(1/2)*N1*r2^4-d2*r2^4*log(r2)-(1/2)*T1*r3^4*log(r3)-
(1/2)*N1*r3^4-(1/3)*M1*r3^6-r3^2*S1-(2/3)*r3^3*V1+(1/8)*T1*r3^4)+pi*(h1-h0)*G*k*(-(2/3)*r2^3*V1-
r2^2*S1-(1/3)*M1*r2^6+(1/8)*T1*r2^4-(1/2)*T1*r2^4*log(r2)-
(1/2)*N1*r2^4+(1/2)*N1*r1^4+C1*r2^2+B1*r2^4+(1/4)*d1*r2^4+(2/3)*L1*r2^3+(4/3)*A1*r2^6+r1^2*S1+(2/
3)*r1^3*V1+d1*r2^4*log(r2)-d1*r1^4*log(r1)+(1/3)*M1*r1^6-C1*r1^2-B1*r1^4-(1/8)*T1*r1^4-
(1/4)*d1*r1^4+(1/2)*T1*r1^4*log(r1)-(2/3)*L1*r1^3-(4/3)*A1*r1^6)+lambda4*a;
        (1/72)*pi*(h1-h0)^3*E*(-12*nu*M1*a^4*r1^2+12*nu*V1*r1*a^2-
12*nu*V1*r1^2*a+12*nu*M1*r1^4*a^2-12*nu*S1*r1^2+12*V1*r1^2*a+12*nu*S1*a^2-12*V1*r1*a^2-
12*T1*log(r1)*r1^2*a^2+12*T1*log(a)*r1^2*a^2+12*nu*T1*log(r1)*r1^2*a^2-
12*nu*T1*log(a)*r1^2*a^2+12*M1*a^4*r1^2-12*M1*r1^4*a^2+12*r1^2*S1-12*a^2*S1)/((1-
nu^2)*r1^2*a^2)+pi*(h1-h0)*G*k*(-2*d3*r3^2*log(r3)+2*B3*a^2+2*d3*a^2*log(a)-2*S1*log(a)-
(1/2)*T1*r3^2+(1/2)*T1*a^2+2*A3*a^4+2*C3*log(a)+2*S1*log(r3)-2*A3*r3^4-2*C3*log(r3)-
2*L3*r3+(1/2)*M1*r3^4+N1*r3^2+2*V1*r3+2*L3*a-(1/2)*M1*a^4-N1*a^2-2*V1*a+T1*r3^2*log(r3)-
T1*a^2*log(a)-2*B3*r3^2)+pi*(h1-h0)*G*k*(-
2*L2*r2+2*B2*r3^2+2*C2*log(r3)+2*A2*r3^4+2*L2*r3+2*V1*r2+N1*r2^2+(1/2)*M1*r2^4-2*B2*r2^2-
(1/2)*T1*r2^2-2*C2*log(r2)-2*A2*r2^4+2*d2*r3^2*log(r3)-
2*d2*r2^2*log(r2)+2*S1*log(r2)+T1*r2^2*log(r2)+(1/2)*T1*r3^2-2*S1*log(r3)-(1/2)*M1*r3^4-N1*r3^2-
2*V1*r3-T1*r3^2*log(r3))+pi*(h1-h0)*G*k*(-2*V1*r2-N1*r2^2-(1/2)*M1*r2^4+(1/2)*T1*r2^2-
2*S1*log(r2)-
T1*r2^2*log(r2)+2*d1*r2^2*log(r2)+2*S1*log(r1)+2*B1*r2^2+2*A1*r2^4+2*C1*log(r2)+2*L1*r2+N1*r1^2+2
*V1*r1-(1/2)*T1*r1^2-2*B1*r1^2-2*A1*r1^4-2*C1*log(r1)+(1/2)*M1*r1^4-
2*d1*r1^2*log(r1)+T1*r1^2*log(r1)-2*L1*r1)+lambda4/a;
        (1/72)*pi*(h1-h0)^3*E*(-12*nu*M1*a^6*log(a)*r1^2-12*S1*log(r1)*r1^2*a^2-
12*nu*N1*a^4*log(a)*r1^2-
12*nu*T1*a^4*log(a)^2*r1^2+12*N1*r1^4*log(r1)*a^2+12*nu*V1*r1^3*log(r1)*a^2+6*T1*r1^4*a^2-
6*T1*a^4*r1^2+12*M1*r1^6*log(r1)*a^2+12*nu*T1*r1^4*log(r1)^2*a^2+12*V1*r1^3*log(r1)*a^2+12*nu*M1*
r1^6*log(r1)*a^2+12*nu*N1*r1^4*log(r1)*a^2-12*V1*r1^3*a^2+6*M1*r1^6*a^2+12*V1*a^3*r1^2-
```

```
6*M1*a^6*r1^2-12*V1*a^3*log(a)*r1^2-12*nu*V1*a^3*log(a)*r1^2-12*N1*a^4*log(a)*r1^2-
12*M1*a^6*log(a)*r1^2+12*S1*log(a)*r1^2*a^2+12*nu*S1*log(r1)*r1^2*a^2-
12*nu*S1*log(a)*r1^2*a^2+12*T1*r1^4*log(r1)^2*a^2-12*T1*a^4*log(a)^2*r1^2)/((1-
nu^2)*r1^2*a^2)+pi*(h1-
h0)*G*k*((1/2)*T1*r3^4*log(r3)^2+(1/2)*C3*r3^2+(2/9)*A3*r3^6+(2/9)*a^3*V1+(2/9)*L3*r3^3+(1/2)*a^2
*S1-(2/9)*A3*a^6+(1/4)*B3*r3^4-(1/18)*M1*r3^6+(1/16)*T1*r3^4-(1/4)*B3*a^4-
(1/16)*T1*a^4+(1/18)*M1*a^6-(2/9)*r3^3*V1-(1/2)*C3*a^2-
C3*r3^2*log(r3)+d3*a^4*log(a)^2+(1/2)*N1*r3^4*log(r3)-(1/2)*N1*a^4*log(a)+C3*a^2*log(a)-
(2/9)*L3*a^3-(1/2)*T1*a^4*log(a)^2-(4/3)*A3*r3^6*log(r3)-
(1/4)*T1*r3^4*log(r3)+(1/4)*T1*a^4*log(a)-(1/2)*r3^2*S1-
(1/8)*N1*r3^4+(1/8)*N1*a^4+(1/3)*M1*r3^6*log(r3)+(2/3)*V1*r3^3*log(r3)-(2/3)*V1*a^3*log(a)-
(2/3)*L3*r3^3*log(r3)+B3*a^4*log(a)+(2/3)*L3*a^3*log(a)-(1/3)*M1*a^6*log(a)+S1*r3^2*log(r3)-
B3*r3^4*log(r3)-S1*a^2*log(a)-d3*r3^4*log(r3)^2+(4/3)*A3*a^6*log(a))+pi*(h1-h0)*G*k*(-
(1/2)*T1*r3^4*log(r3)^2+(1/2)*T1*r2^4*log(r2)^2+(1/4)*B2*r2^4+(1/16)*T1*r2^4-(2/9)*L2*r3^3-
(2/9)*A2*r3^6+(2/9)*A2*r2^6-(1/4)*B2*r3^4+(1/18)*M1*r3^6-(1/16)*T1*r3^4-
(2/9)*r2^3*V1+(2/9)*L2*r2^3-(1/18)*M1*r2^6+(2/9)*r3^3*V1-(1/2)*C2*r3^2-(1/2)*r2^2*S1-
(1/2)*N1*r3^4*log(r3)+(1/4)*T1*r3^4*log(r3)-(1/4)*T1*r2^4*log(r2)+(1/2)*r3^2*S1+(1/8)*N1*r3^4-
(1/8)*N1*r2^4-(1/3)*M1*r3^6*log(r3)-(2/3)*V1*r3^3*log(r3)-
S1*r3^2*log(r3)+(2/3)*L2*r3^3*log(r3)+d2*r3^4*log(r3)^2+B2*r3^4*log(r3)+C2*r3^2*log(r3)+(4/3)*A2*
r3^6*log(r3)-B2*r2^4*log(r2)+(2/3)*V1*r2^3*log(r2)+(1/2)*N1*r2^4*log(r2)-
d2*r2^4*log(r2)^2+(1/3)*M1*r2^6*log(r2)-(2/3)*L2*r2^3*log(r2)-C2*r2^2*log(r2)-
(4/3)*A2*r2^6*log(r2)+S1*r2^2*log(r2)+(1/2)*C2*r2^2)+pi*(h1-
h0)*G*k*((1/2)*T1*r1^4*log(r1)^2+(1/16)*T1*r1^4-(1/2)*T1*r2^4*log(r2)^2+(2/9)*A1*r1^6-
(2/9)*A1*r2^6+(1/4)*B1*r1^4-(1/18)*M1*r1^6-(2/9)*L1*r2^3-(1/16)*T1*r2^4+(1/2)*C1*r1^2-
(1/4)*B1*r2^4+(2/9)*L1*r1^3+(2/9)*r2^3*V1+(1/18)*M1*r2^6+(1/2)*r2^2*S1-
(2/9)*r1^3*V1+(1/4)*T1*r2^4*log(r2)-(1/8)*N1*r1^4-(1/4)*T1*r1^4*log(r1)+(1/8)*N1*r2^4-
(1/2)*r1^2*S1-(2/3)*V1*r2^3*log(r2)-(1/2)*N1*r2^4*log(r2)-(1/3)*M1*r2^6*log(r2)-
S1*r2^2*log(r2)+B1*r2^4*log(r2)+(2/3)*L1*r2^3*log(r2)+(4/3)*A1*r2^6*log(r2)+C1*r2^2*log(r2)+(2/3)
*V1*r1^3*log(r1)-(2/3)*L1*r1^3*log(r1)-d1*r1^4*log(r1)^2+d1*r2^4*log(r2)^2-
B1*r1^4*log(r1)+S1*r1^2*log(r1)-
(4/3)*A1*r1^6*log(r1)+(1/3)*M1*r1^6*log(r1)+(1/2)*N1*r1^4*log(r1)-C1*r1^2*log(r1)-
(1/2)*C1*r2^2)+lambda4*a*log(a);
        (1/72)*pi*(h1-h0)^3*E*(-12*nu*M1*a^5*r1^2-
12*nu*N1*a^3*r1^2+12*nu*M1*r1^5*a^2+12*nu*N1*r1^3*a^2+12*nu*S1*r1*a^2-
12*nu*S1*r1^2*a+12*S1*r1^2*a-12*S1*r1*a^2-12*T1*a^3*log(a)*r1^2+12*T1*r1^3*log(r1)*a^2-
12*nu*T1*a^3*log(a)*r1^2+12*nu*T1*r1^3*log(r1)*a^2-12*V1*log(a)*r1^2*a^2+12*V1*log(r1)*r1^2*a^2-
4*M1*a^5*r1^2+12*T1*a^3*r1^2-12*N1*a^3*r1^2-12*T1*r1^3*a^2+12*N1*r1^3*a^2+4*M1*r1^5*a^2)/((1-
nu^2)*r1^2*a^2)+pi*(h1-h0)*G*k*(-(4/3)*d3*r3^3*log(r3)+(4/3)*d3*a^3*log(a)+2*C3*a-
V1*a^2+(2/9)*T1*a^3+(8/5)*A3*a^5+(2/9)*d3*a^3+(2/3)*T1*r3^3*log(r3)+V1*r3^2-(2/9)*T1*r3^3-
(8/5)*A3*r3^5-(2/9)*d3*r3^3-L3*r3^2-
(2/3)*T1*a^3*log(a)+(2/5)*M1*r3^5+(2/3)*N1*r3^3+2*S1*r3+L3*a^2-(2/5)*M1*a^5-(2/3)*N1*a^3-2*S1*a-
(4/3)*B3*r3^3+(4/3)*B3*a^3-2*C3*r3)+pi*(h1-h0)*G*k*(-(2/9)*d2*r2^3-
2*C2*r2+2*C2*r3+(4/3)*B2*r3^3+(8/5)*A2*r3^5+L2*r3^2+(2/9)*d2*r3^3-(8/5)*A2*r2^5-
L2*r2^2+2*S1*r2+(2/3)*N1*r2^3+(2/5)*M1*r2^5-(2/9)*T1*r2^3-
(4/3)*B2*r2^3+(4/3)*d2*r3^3*log(r3)+(2/3)*T1*r2^3*log(r2)-(4/3)*d2*r2^3*log(r2)+V1*r2^2-
(2/3)*T1*r3^3*log(r3)-V1*r3^2+(2/9)*T1*r3^3-(2/5)*M1*r3^5-(2/3)*N1*r3^3-2*S1*r3)+pi*(h1-
h0)*G*k*(-2*S1*r2-(2/3)*N1*r2^3-(2/5)*M1*r2^5+(2/9)*T1*r2^3-(2/3)*T1*r2^3*log(r2)-
V1*r2^2+(4/3)*d1*r2^3*log(r2)+V1*r1^2+(2/5)*M1*r1^5+2*C1*r2+(4/3)*B1*r2^3+(8/5)*A1*r2^5+(2/9)*d1*
r2^3+L1*r2^2+2*S1*r1+(2/3)*N1*r1^3+(2/3)*T1*r1^3*log(r1)-(4/3)*d1*r1^3*log(r1)-2*C1*r1-
(4/3)*B1*r1^3-(8/5)*A1*r1^5-(2/9)*T1*r1^3-(2/9)*d1*r1^3-L1*r1^2)+lambda4;
        -(1/72)*pi*(h2-h1)^3*E*(12*T2*a^6*log(a)*r2^2-
20*M2*r2^8*a^2+20*M2*a^8*r2^2+12*nu*V2*a^5*r2^2-12*S2*a^4*r2^2+6*T2*a^6*r2^2-4*V2*r2^5*a^2-
12*nu*M2*r2^8*a^2-6*T2*r2^6*a^2+12*S2*r2^4*a^2-
12*N2*r2^6*a^2+12*N2*a^6*r2^2+4*V2*a^5*r2^2+12*nu*M2*a^8*r2^2+12*nu*N2*a^6*r2^2-
12*nu*S2*r2^4*a^2-12*nu*V2*r2^5*a^2+12*nu*S2*a^4*r2^2-12*nu*N2*r2^6*a^2-
12*T2*r2^6*log(r2)*a^2+12*nu*T2*a^6*log(a)*r2^2-12*nu*T2*r2^6*log(r2)*a^2)/((1-
nu^2)*r2^2*a^2)+pi*(h2-h1)*G*k*((2/9)*d3*a^6+(1/2)*C3*a^4+(1/18)*T2*a^6+A3*a^8-
(1/3)*T2*a^6*log(a)-(1/2)*C3*r3^4-(1/18)*T2*r3^6-A3*r3^8+(2/3)*B3*a^6+(2/3)*d3*a^6*log(a)-
(1/3)*a^6*N2-(1/2)*a^4*S2-(2/5)*a^5*V2+(2/5)*L3*a^5-
(2/3)*B3*r3^6+(1/3)*r3^6*N2+(1/2)*r3^4*S2+(2/5)*r3^5*V2-(2/5)*L3*r3^5+(1/3)*T2*r3^6*log(r3)-
(2/9)*d3*r3^6-(2/3)*d3*r3^6*log(r3)-(1/4)*M2*a^8+(1/4)*M2*r3^8)+pi*(h2-h1)*G*k*((1/18)*T2*r3^6-
(1/3)*r3^6*N2-(1/2)*r3^4*S2-(2/5)*r3^5*V2-(1/3)*T2*r3^6*log(r3)-A2*r2^8-(1/2)*C2*r2^4-
(2/3)*B2*r2^6+(1/2)*r2^4*S2+(2/5)*r2^5*V2-(1/18)*T2*r2^6+(1/3)*r2^6*N2-(2/5)*L2*r2^5-
(2/9)*d2*r2^6-(2/3)*d2*r2^6*log(r2)+(1/3)*T2*r2^6*log(r2)+(2/3)*d2*r3^6*log(r3)-
(1/4)*M2*r3^8+(2/9)*d2*r3^6+(2/5)*L2*r3^5+(1/4)*M2*r2^8+A2*r3^8+(1/2)*C2*r3^4+(2/3)*B2*r3^6)+lamb
da5*a^3;
        -(1/72)*pi*(h2-h1)^3*E*(12*nu*T2*a^4*log(a)*r2^2+12*T2*a^4*log(a)*r2^2+12*N2*a^4*r2^2-
12*N2*r2^4*a^2-12*nu*N2*r2^4*a^2-12*V2*r2^3*a^2-
12*M2*r2^6*a^2+12*M2*a^6*r2^2+12*nu*N2*a^4*r2^2+12*V2*a^3*r2^2+12*nu*M2*a^6*r2^2-
```

```
12*nu*V2*r2^3*a^2+12*nu*V2*a^3*r2^2-12*nu*M2*r2^6*a^2-12*T2*r2^4*log(r2)*a^2-
12*nu*T2*r2^4*log(r2)*a^2)/((1-nu^2)*r2^2*a^2)+pi*(h2-
h1)*G*k*((1/4)*d3*a^4+B3*a^4+(1/8)*T2*a^4+(4/3)*A3*a^6+d3*a^4*log(a)-(1/4)*d3*r3^4-B3*r3^4-
(1/8)*T2*r3^4-(4/3)*A3*r3^6-d3*r3^4*log(r3)-a^2*S2-(1/2)*T2*a^4*log(a)-(2/3)*a^3*V2+C3*a^2-
(1/3)*M2*a^6+(2/3)*L3*a^3+r3^2*S2+(2/3)*r3^3*V2-C3*r3^2+(1/3)*M2*r3^6-
(2/3)*L3*r3^3+(1/2)*T2*r3^4*log(r3)-(1/2)*N2*a^4+(1/2)*N2*r3^4)+pi*(h2-h1)*G*k*((1/8)*T2*r3^4-
r3^2*S2-(2/3)*r3^3*V2-(1/3)*M2*r3^6-(1/2)*T2*r3^4*log(r3)-(2/3)*L2*r2^3-C2*r2^2-
B2*r2^4+(2/3)*r2^3*V2+(1/3)*M2*r2^6+r2^2*S2-(1/8)*T2*r2^4-(4/3)*A2*r2^6-
d2*r2^4*log(r2)+(1/2)*T2*r2^4*log(r2)-(1/4)*d2*r2^4-
(1/2)*N2*r3^4+(4/3)*A2*r3^6+(1/2)*N2*r2^4+(1/4)*d2*r3^4+(2/3)*L2*r3^3+C2*r3^2+B2*r3^4+d2*r3^4*log
(r3))+lambda5*a;
        -(1/72)*pi*(h2-h1)^3*E*(12*nu*T2*log(a)*r2^2*a^2-
12*nu*T2*log(r2)*r2^2*a^2+12*T2*log(r2)*r2^2*a^2-12*T2*log(a)*r2^2*a^2+12*a^2*S2-12*r2^2*S2-
12*M2*a^4*r2^2+12*M2*r2^4*a^2-12*nu*M2*r2^4*a^2+12*nu*M2*a^4*r2^2-12*nu*S2*a^2+12*V2*r2*a^2-
12*nu*V2*r2*a^2+12*nu*V2*r2^2*a+12*nu*S2*r2^2-12*V2*r2^2*a)/((1-nu^2)*r2^2*a^2)+pi*(h2-h1)*G*k*(-
T2*a^2*log(a)+2*C3*log(a)+(1/2)*T2*a^2+2*A3*a^4-2*C3*log(r3)-(1/2)*T2*r3^2-2*A3*r3^4+2*B3*a^2-
2*V2*a-N2*a^2+2*d3*a^2*log(a)-(1/2)*M2*a^4+2*L3*a-2*B3*r3^2+2*V2*r3+N2*r3^2+(1/2)*M2*r3^4-
2*L3*r3+T2*r3^2*log(r3)-2*d3*r3^2*log(r3)-2*S2*log(a)+2*S2*log(r3))+pi*(h2-
h1)*G*k*((1/2)*T2*r3^2-2*V2*r3-N2*r3^2-(1/2)*M2*r3^4-T2*r3^2*log(r3)-2*L2*r2-2*A2*r2^4-2*B2*r2^2-
2*C2*log(r2)+(1/2)*M2*r2^4+N2*r2^2+2*V2*r2-(1/2)*T2*r2^2-
2*d2*r2^2*log(r2)+T2*r2^2*log(r2)+2*d2*r3^2*log(r3)-
2*S2*log(r3)+2*S2*log(r2)+2*L2*r3+2*A2*r3^4+2*B2*r3^2+2*C2*log(r3))+lambda5/a;
        -(1/72)*pi*(h2-h1)^3*E*(-12*nu*S2*log(r2)*r2^2*a^2-12*N2*r2^4*log(r2)*a^2-
6*T2*r2^4*a^2+12*V2*r2^3*a^2+12*nu*S2*log(a)*r2^2*a^2-
12*S2*log(a)*r2^2*a^2+12*nu*T2*a^4*log(a)^2*r2^2+12*V2*a^3*log(a)*r2^2-
6*M2*r2^6*a^2+6*M2*a^6*r2^2+12*N2*a^4*log(a)*r2^2+12*S2*log(r2)*r2^2*a^2+12*M2*a^6*log(a)*r2^2-
12*nu*V2*r2^3*log(r2)*a^2+12*nu*N2*a^4*log(a)*r2^2-12*nu*T2*r2^4*log(r2)^2*a^2-
12*V2*r2^3*log(r2)*a^2-12*M2*r2^6*log(r2)*a^2-
12*T2*r2^4*log(r2)^2*a^2+12*T2*a^4*log(a)^2*r2^2+12*nu*V2*a^3*log(a)*r2^2-
12*nu*N2*r2^4*log(r2)*a^2+12*nu*M2*a^6*log(a)*r2^2-12*nu*M2*r2^6*log(r2)*a^2+6*T2*a^4*r2^2-
12*V2*a^3*r2^2)/((1-nu^2)*r2^2*a^2)+pi*(h2-h1)*G*k*((4/3)*A3*a^6*log(a)+B3*a^4*log(a)-
(1/2)*T2*a^4*log(a)^2+(1/2)*a^2*S2-(1/18)*M2*r3^6-
(1/8)*N2*r3^4+(1/2)*C3*r3^2+(2/9)*A3*r3^6+(1/4)*T2*a^4*log(a)-(1/4)*T2*r3^4*log(r3)-
(1/2)*r3^2*S2+(1/16)*T2*r3^4+(2/9)*L3*r3^3-(2/9)*A3*a^6+(1/4)*B3*r3^4-S2*a^2*log(a)-
(2/3)*V2*a^3*log(a)+d3*a^4*log(a)^2-(1/3)*M2*a^6*log(a)+(2/3)*L3*a^3*log(a)-
(2/3)*L3*r3^3*log(r3)-(1/4)*B3*a^4-(1/2)*C3*a^2+(1/18)*M2*a^6-
(1/2)*N2*a^4*log(a)+C3*a^2*log(a)+(1/3)*M2*r3^6*log(r3)+S2*r3^2*log(r3)+(2/3)*V2*r3^3*log(r3)-
d3*r3^4*log(r3)^2-(4/3)*A3*r3^6*log(r3)-B3*r3^4*log(r3)+(1/2)*N2*r3^4*log(r3)-C3*r3^2*log(r3)-
(2/9)*r3^3*V2-(1/16)*T2*a^4+(2/9)*a^3*V2-
(2/9)*L3*a^3+(1/2)*T2*r3^4*log(r3)^2+(1/8)*N2*a^4)+pi*(h2-h1)*G*k*(-
(2/9)*r2^3*V2+(1/2)*T2*r2^4*log(r2)^2-(1/2)*r2^2*S2+(1/18)*M2*r3^6+(1/8)*N2*r3^4-
(1/8)*N2*r2^4+(1/4)*B2*r2^4+(1/4)*T2*r3^4*log(r3)-
(1/4)*T2*r2^4*log(r2)+(1/2)*r3^2*S2+(1/16)*T2*r2^4-(1/16)*T2*r3^4-(2/9)*L2*r3^3-
(2/9)*A2*r3^6+(2/9)*A2*r2^6-(1/4)*B2*r3^4+(2/9)*L2*r2^3-(1/2)*C2*r3^2-(1/18)*M2*r2^6-
(1/3)*M2*r3^6*log(r3)-S2*r3^2*log(r3)-(2/3)*V2*r3^3*log(r3)-
(1/2)*N2*r3^4*log(r3)+(4/3)*A2*r3^6*log(r3)-(4/3)*A2*r2^6*log(r2)-B2*r2^4*log(r2)-
(2/3)*L2*r2^3*log(r2)-d2*r2^4*log(r2)^2-
C2*r2^2*log(r2)+(1/2)*N2*r2^4*log(r2)+(1/3)*M2*r2^6*log(r2)+(2/9)*r3^3*V2-
(1/2)*T2*r3^4*log(r3)^2+S2*r2^2*log(r2)+(2/3)*V2*r2^3*log(r2)+d2*r3^4*log(r3)^2+C2*r3^2*log(r3)+(
2/3)*L2*r3^3*log(r3)+B2*r3^4*log(r3)+(1/2)*C2*r2^2)+lambda5*a*log(a);
        -(1/72)*pi*(h2-h1)^3*E*(-
12*nu*T2*r2^3*log(r2)*a^2+12*T2*a^3*log(a)*r2^2+12*nu*M2*a^5*r2^2-12*T2*a^3*r2^2-4*M2*r2^5*a^2-
12*N2*r2^3*a^2+12*T2*r2^3*a^2+4*M2*a^5*r2^2+12*N2*a^3*r2^2-12*nu*N2*r2^3*a^2-
12*nu*M2*r2^5*a^2+12*nu*N2*a^3*r2^2+12*S2*r2*a^2-12*nu*S2*r2*a^2+12*nu*S2*r2^2*a-12*S2*r2^2*a-
12*T2*r2^3*log(r2)*a^2-
12*V2*log(r2)*r2^2*a^2+12*V2*log(a)*r2^2*a^2+12*nu*T2*a^3*log(a)*r2^2)/((1-
nu^2)*r2^2*a^2)+pi*(h2-h1)*G*k*((2/9)*d3*a^3+(8/5)*A3*a^5+2*C3*a-
(2/3)*T2*a^3*log(a)+(4/3)*d3*a^3*log(a)-(2/9)*d3*r3^3-(8/5)*A3*r3^5-2*C3*r3-
(4/3)*d3*r3^3*log(r3)+(2/9)*T2*a^3+(4/3)*B3*a^3-2*S2*a-(2/3)*N2*a^3-(2/5)*M2*a^5+L3*a^2-
(2/9)*T2*r3^3-(4/3)*B3*r3^3+2*S2*r3+(2/3)*N2*r3^3+(2/5)*M2*r3^5-L3*r3^2+(2/3)*T2*r3^3*log(r3)-
V2*a^2+V2*r3^2)+pi*(h2-h1)*G*k*((2/9)*T2*r3^3-2*S2*r3-(2/3)*N2*r3^3-(2/5)*M2*r3^5-
(2/3)*T2*r3^3*log(r3)-L2*r2^2-(8/5)*A2*r2^5-(4/3)*B2*r2^3+(2/5)*M2*r2^5+(2/3)*N2*r2^3-
(2/9)*T2*r2^3+2*S2*r2-2*C2*r2-(4/3)*d2*r2^3*log(r2)+(2/3)*T2*r2^3*log(r2)+(4/3)*d2*r3^3*log(r3)-
V2*r3^2-(2/9)*d2*r2^3+2*C2*r3+V2*r2^2+(2/9)*d2*r3^3+L2*r3^2+(8/5)*A2*r3^5+(4/3)*B2*r3^3)+lambda5;
        -(1/72)*pi*(h3-h2)^3*E*(20*M3*a^8*r3^2-
20*M3*r3^8*a^2+12*nu*M3*a^8*r3^2+6*T3*a^6*r3^2+12*N3*a^6*r3^2-12*S3*a^4*r3^2+4*V3*a^5*r3^2-
12*nu*M3*r3^8*a^2-6*T3*r3^6*a^2-12*N3*r3^6*a^2+12*S3*r3^4*a^2-
4*V3*r3^5*a^2+12*nu*N3*a^6*r3^2+12*nu*V3*a^5*r3^2+12*nu*S3*a^4*r3^2-12*nu*N3*r3^6*a^2-
12*nu*S3*r3^4*a^2-12*nu*V3*r3^5*a^2-12*nu*T3*r3^6*log(r3)*a^2+12*T3*a^6*log(a)*r3^2-
```

```
12*T3*r3^6*log(r3)*a^2+12*nu*T3*a^6*log(a)*r3^2)/((1-nu^2)*a^2*r3^2)+pi*(h3-h2)*G*k*(-
(2/9)*d3*r3^6-(1/2)*C3*r3^4+(2/5)*r3^5*V3-(2/3)*B3*r3^6-
A3*r3^8+(1/4)*M3*r3^8+(1/3)*r3^6*N3+(1/2)*r3^4*S3-(1/18)*T3*r3^6+(2/3)*d3*a^6*log(a)-
(1/3)*T3*a^6*log(a)+(1/3)*T3*r3^6*log(r3)+A3*a^8+(2/9)*d3*a^6+(1/2)*C3*a^4-
(2/5)*a^5*V3+(2/3)*B3*a^6-(1/3)*a^6*N3-(1/2)*a^4*S3+(1/18)*T3*a^6-(2/5)*L3*r3^5+(2/5)*L3*a^5-
(2/3)*d3*r3^6*log(r3)-(1/4)*M3*a^8)+lambda6*a^3;
        -(1/72)*pi*(h3-h2)^3*E*(12*T3*a^4*log(a)*r3^2+12*N3*a^4*r3^2-
12*N3*r3^4*a^2+12*nu*N3*a^4*r3^2+12*M3*a^6*r3^2+12*V3*a^3*r3^2-12*nu*N3*r3^4*a^2-12*M3*r3^6*a^2-
12*V3*r3^3*a^2+12*nu*V3*a^3*r3^2+12*nu*M3*a^6*r3^2-12*nu*M3*r3^6*a^2-12*nu*V3*r3^3*a^2-
12*nu*T3*r3^4*log(r3)*a^2+12*nu*T3*a^4*log(a)*r3^2-12*T3*r3^4*log(r3)*a^2)/((1-
nu^2)*a^2*r3^2)+pi*(h3-h2)*G*k*(-(1/4)*d3*r3^4-C3*r3^2-(4/3)*A3*r3^6+r3^2*S3+(2/3)*r3^3*V3-
B3*r3^4+(1/2)*N3*r3^4+d3*a^4*log(a)+(1/3)*M3*r3^6-(1/8)*T3*r3^4-
(1/2)*T3*a^4*log(a)+(1/2)*T3*r3^4*log(r3)+(2/3)*L3*a^3-a^2*S3-
(2/3)*a^3*V3+(1/4)*d3*a^4+C3*a^2+(4/3)*A3*a^6+B3*a^4-(1/3)*M3*a^6+(1/8)*T3*a^4-d3*r3^4*log(r3)-
(2/3)*L3*r3^3-(1/2)*N3*a^4)+lambda6*a;
        -(1/72)*pi*(h3-h2)^3*E*(-12*nu*V3*a^2*r3-12*V3*a*r3^2-12*nu*S3*a^2+12*nu*V3*a*r3^2-
12*M3*a^4*r3^2+12*M3*r3^4*a^2-12*r3^2*S3+12*a^2*S3+12*nu*M3*a^4*r3^2-
12*nu*M3*r3^4*a^2+12*nu*S3*r3^2+12*V3*a^2*r3+12*nu*T3*log(a)*a^2*r3^2-12*nu*T3*log(r3)*a^2*r3^2-
12*T3*log(a)*a^2*r3^2+12*T3*log(r3)*a^2*r3^2)/((1-nu^2)*a^2*r3^2)+pi*(h3-
h2)*G*k*(T3*r3^2*log(r3)-2*C3*log(r3)-2*A3*r3^4-2*d3*r3^2*log(r3)-2*B3*r3^2-
(1/2)*T3*r3^2+N3*r3^2+2*V3*r3+2*S3*log(r3)-2*L3*r3+(1/2)*M3*r3^4-
T3*a^2*log(a)+2*d3*a^2*log(a)+2*B3*a^2+(1/2)*T3*a^2-N3*a^2-2*V3*a+2*C3*log(a)+2*A3*a^4+2*L3*a-
(1/2)*M3*a^4-2*S3*log(a))+lambda6/a;
        -(1/72)*pi*(h3-h2)^3*E*(-
12*T3*r3^4*log(r3)^2*a^2+12*T3*a^4*log(a)^2*r3^2+12*nu*T3*a^4*log(a)^2*r3^2+12*N3*a^4*log(a)*r3^2
-12*nu*T3*r3^4*log(r3)^2*a^2+12*V3*a^3*log(a)*r3^2-
12*nu*N3*r3^4*log(r3)*a^2+12*nu*N3*a^4*log(a)*r3^2-12*nu*V3*r3^3*log(r3)*a^2-
12*nu*M3*r3^6*log(r3)*a^2+12*nu*V3*a^3*log(a)*r3^2+12*nu*S3*log(a)*a^2*r3^2+12*M3*a^6*log(a)*r3^2
-12*V3*r3^3*log(r3)*a^2-12*N3*r3^4*log(r3)*a^2-12*M3*r3^6*log(r3)*a^2-12*nu*S3*log(r3)*a^2*r3^2-
12*S3*log(a)*a^2*r3^2+12*S3*log(r3)*a^2*r3^2+12*nu*M3*a^6*log(a)*r3^2+6*M3*a^6*r3^2-
12*V3*a^3*r3^2-6*M3*r3^6*a^2+12*V3*r3^3*a^2+6*T3*a^4*r3^2-6*T3*r3^4*a^2)/((1-
nu^2)*a^2*r3^2)+pi*(h3-h2)*G*k*(-(1/16)*T3*a^4-(2/9)*L3*a^3+(1/16)*T3*r3^4-
(1/2)*T3*a^4*log(a)^2+(1/2)*T3*r3^4*log(r3)^2+(1/4)*T3*a^4*log(a)-
(1/4)*T3*r3^4*log(r3)+(2/3)*V3*r3^3*log(r3)+S3*r3^2*log(r3)-B3*r3^4*log(r3)-d3*r3^4*log(r3)^2-
(4/3)*A3*r3^6*log(r3)-(1/2)*N3*a^4*log(a)+(2/3)*L3*a^3*log(a)-(1/3)*M3*a^6*log(a)-(1/2)*C3*a^2-
(1/8)*N3*r3^4+(1/8)*N3*a^4+(1/2)*N3*r3^4*log(r3)+(1/3)*M3*r3^6*log(r3)-(2/3)*V3*a^3*log(a)-
S3*a^2*log(a)+B3*a^4*log(a)+d3*a^4*log(a)^2+(4/3)*A3*a^6*log(a)+(2/9)*a^3*V3-
(2/9)*A3*a^6+(1/18)*M3*a^6+(1/2)*a^2*S3+(2/9)*A3*r3^6-(2/9)*r3^3*V3-(1/2)*r3^2*S3+(1/4)*B3*r3^4-
(1/18)*M3*r3^6+(1/2)*C3*r3^2-(1/4)*B3*a^4-C3*r3^2*log(r3)+C3*a^2*log(a)-
(2/3)*L3*r3^3*log(r3)+(2/9)*L3*r3^3)+lambda6*a*log(a);
        -(1/72)*pi*(h3-h2)^3*E*(-12*nu*S3*a^2*r3-
12*S3*a*r3^2+12*nu*S3*a*r3^2+12*T3*a^3*log(a)*r3^2-
12*T3*a^3*r3^2+12*N3*a^3*r3^2+4*M3*a^5*r3^2+12*T3*r3^3*a^2-12*N3*r3^3*a^2-4*M3*r3^5*a^2-
12*V3*log(r3)*a^2*r3^2+12*V3*log(a)*a^2*r3^2+12*nu*N3*a^3*r3^2+12*nu*M3*a^5*r3^2-
12*nu*N3*r3^3*a^2-12*nu*M3*r3^5*a^2+12*S3*a^2*r3-
12*nu*T3*r3^3*log(r3)*a^2+12*nu*T3*a^3*log(a)*r3^2-12*T3*r3^3*log(r3)*a^2)/((1-
nu^2)*a^2*r3^2)+pi*(h3-h2)*G*k*((2/3)*T3*r3^3*log(r3)-2*C3*r3-(4/3)*B3*r3^3+(2/5)*M3*r3^5-
(4/3)*d3*r3^3*log(r3)-(8/5)*A3*r3^5+(2/3)*N3*r3^3+2*S3*r3+V3*r3^2-L3*r3^2-(2/9)*T3*r3^3-
(2/3)*T3*a^3*log(a)+(4/3)*d3*a^3*log(a)+(8/5)*A3*a^5-(2/3)*N3*a^3-
2*S3*a+(2/9)*d3*a^3+2*C3*a+(4/3)*B3*a^3-(2/5)*M3*a^5+L3*a^2+(2/9)*T3*a^3-V3*a^2-
(2/9)*d3*r3^3)+lambda6;
        A1*r2^4+B1*r2^2+C1*log(r2)+d1*r2^2*log(r2)+L1*r2+H1-A2*r2^4-B2*r2^2-C2*log(r2)-
d2*r2^2*log(r2)-L2*r2-H2;
        A2*r3^4+B2*r3^2+C2*log(r3)+d2*r3^2*log(r3)+L2*r3+H2-A3*r3^4-B3*r3^2-C3*log(r3)-
d3*r3^2*log(r3)-L3*r3-H3;
        A3*a^4+B3*a^2+C3*log(a)+d3*a^2*log(a)+L3*a+H3;
        M1*a^3+N1*a+S1/a+T1*a*log(a)+V1;
        M2*a^3+N2*a+S2/a+T2*a*log(a)+V2;
        M3*a^3+N3*a+S3/a+T3*a*log(a)+V3];
end
if l>=a && l<r3
    FF=[(1/3)*p0*pi*r2^6+lambda1*r2^4+pi*(h1-h0)*G*k*(-(8/5)*r2^5*L1-(8/9)*r2^6*d1-r1^8*M1-
(8/5)*r1^5*V1+2*r2^4*S1-(8/3)*r2^6*B1-2*r2^4*C1-
(2/9)*T1*r2^6+r2^8*M1+(4/3)*r2^6*N1+(8/5)*r2^5*V1-
4*A1*r2^8+(8/3)*r1^6*B1+2*r1^4*C1+4*A1*r1^8+(2/9)*T1*r1^6+(8/5)*r1^5*L1+(8/9)*r1^6*d1-2*r1^4*S1-
(8/3)*d1*r2^6*log(r2)+(4/3)*T1*r2^6*log(r2)-(4/3)*T1*r1^6*log(r1)+(8/3)*d1*r1^6*log(r1)-
(4/3)*r1^6*N1)-(1/3)*p0*pi*r1^6;
        (1/2)*p0*pi*r2^4+lambda1*r2^2+pi*(h1-h0)*G*k*(-(4/3)*r1^3*V1+2*r1^2*C1-(8/3)*A1*r2^6-
(1/4)*T1*r2^4+T1*r2^4*log(r2)+(4/3)*r2^3*V1-2*r2^2*C1-(4/3)*r2^3*L1+(2/3)*r2^6*M1+r2^4*N1-
```

```
(1/2)*r2^4*d1-2*d1*r2^4*log(r2)+2*r2^2*S1-2*B1*r2^4-(2/3)*r1^6*M1-r1^4*N1+(1/2)*r1^4*d1-
2*r1^2*S1+(8/3)*A1*r1^6+2*B1*r1^4+(1/4)*T1*r1^4+(4/3)*r1^3*L1+2*d1*r1^4*log(r1)-T1*r1^4*log(r1))-
(1/2)*p0*pi*r1^4;
        -(1/2)*p0*pi*r2^2+lambda1*log(r2)+pi*(h1-h0)*G*k*(-2*S1*log(r1)+2*B1*r1^2-2*A1*r2^4-
(1/2)*T1*r2^2+2*S1*log(r2)-2*B1*r2^2+N1*r2^2+2*V1*r2+(1/2)*M1*r2^4-2*L1*r2-
2*C1*log(r2)+2*A1*r1^4+2*C1*log(r1)+(1/2)*T1*r1^2-N1*r1^2-2*V1*r1-(1/2)*M1*r1^4+2*L1*r1-
2*d1*r2^2*log(r2)+T1*r2^2*log(r2)+2*d1*r1^2*log(r1)-
T1*r1^2*log(r1))+p0*pi*r2^2*log(r2)+(1/2)*p0*pi*r1^2-p0*pi*r1^2*log(r1);
        lambda1*r2^2*log(r2)+pi*(h1-h0)*G*k*(-
d1*r2^4*log(r2)+2*S1*r2^2*log(r2)+(4/3)*V1*r2^3*log(r2)+2*C1*r1^2*log(r1)+(4/3)*L1*r1^3*log(r1)-
2*S1*r1^2*log(r1)+(2/9)*r1^3*L1-2*C1*r2^2*log(r2)-(8/3)*A1*r2^6*log(r2)-
(1/4)*r2^4*d1+N1*r2^4*log(r2)-T1*r1^4*log(r1)^2-(4/3)*V1*r1^3*log(r1)+(8/3)*A1*r1^6*log(r1)-
(1/4)*r1^4*N1+(2/9)*r2^6*M1-2*B1*r2^4*log(r2)-
(2/3)*M1*r1^6*log(r1)+(2/9)*r2^3*V1+(2/3)*M1*r2^6*log(r2)+d1*r1^4*log(r1)-
(1/2)*B1*r2^4+(1/2)*B1*r1^4-2*d1*r2^4*log(r2)^2+2*d1*r1^4*log(r1)^2-(2/9)*r2^3*L1+(1/4)*r1^4*d1-
(4/3)*L1*r2^3*log(r2)+2*B1*r1^4*log(r1)+(8/9)*A1*r1^6-(8/9)*A1*r2^6-(2/9)*r1^6*M1+(1/4)*r2^4*N1-
(2/9)*r1^3*V1-N1*r1^4*log(r1)+T1*r2^4*log(r2)^2)+(1/2)*p0*pi*r2^4*log(r2)-
(1/8)*p0*pi*r2^4+(1/8)*p0*pi*r1^4-(1/2)*p0*pi*r1^4*log(r1);
        lambda1*r2+pi*(h1-h0)*G*k*(-(8/5)*A1*r2^5-2*r1*S1-r1^2*V1+(2/9)*d1*r1^3-
(2/9)*T1*r2^3+(2/5)*r2^5*M1-2*C1*r2-
(4/3)*B1*r2^3+(2/3)*T1*r2^3*log(r2)+(2/3)*r2^3*N1+2*r2*S1+r2^2*V1-(2/9)*d1*r2^3-L1*r2^2-
(2/3)*r1^3*N1+L1*r1^2+(2/9)*T1*r1^3+(8/5)*A1*r1^5-(2/5)*r1^5*M1+2*C1*r1+(4/3)*B1*r1^3-
(4/3)*d1*r2^3*log(r2)-(2/3)*T1*r1^3*log(r1)+(4/3)*d1*r1^3*log(r1))+(2/3)*p0*pi*r2^3-
(2/3)*p0*pi*r1^3;
        lambda1+p0*pi*r2^2-p0*pi*r1^2;
        lambda2*r3^4+(1/3)*p0*pi*r3^6-(1/3)*p0*pi*r2^6-lambda1*r2^4+pi*(h2-h1)*G*k*(-
(8/3)*d2*r3^6*log(r3)+(4/3)*T2*r3^6*log(r3)+r3^8*M2+(4/3)*r3^6*N2+2*r3^4*S2-
(8/3)*r3^6*B2+(8/5)*r3^5*V2-2*r3^4*C2-(8/9)*r3^6*d2-(8/5)*r3^5*L2-
(2/9)*T2*r3^6+(8/3)*d2*r2^6*log(r2)+(8/3)*r2^6*B2-(4/3)*T2*r2^6*log(r2)-
(8/5)*r2^5*V2+2*r2^4*C2+(8/9)*r2^6*d2+(8/5)*r2^5*L2-r2^8*M2-(4/3)*r2^6*N2-
2*r2^4*S2+4*A2*r2^8+(2/9)*T2*r2^6-4*A2*r3^8)+pi*(h1-h0)*G*k*(-(8/3)*d2*r3^6*log(r3)-
(8/3)*r3^6*B2-2*r3^4*C2-(8/9)*r3^6*d2-
(8/5)*r3^5*L2+(8/3)*d2*r2^6*log(r2)+(8/3)*r2^6*B2+2*r2^4*C2+(8/9)*r2^6*d2+(8/5)*r2^5*L2+4*A2*r2^8
-(2/9)*T1*r3^6-(4/3)*r2^6*N1-r2^8*M1-2*r2^4*S1-(8/5)*r2^5*V1+(2/9)*T1*r2^6-
4*A2*r3^8+(4/3)*T1*r3^6*log(r3)-
(4/3)*T1*r2^6*log(r2)+(4/3)*r3^6*N1+r3^8*M1+2*r3^4*S1+(8/5)*r3^5*V1);
        lambda2*r3^2+(1/2)*p0*pi*r3^4-(1/2)*p0*pi*r2^4-lambda1*r2^2+pi*(h2-h1)*G*k*(-
2*d2*r3^4*log(r3)+T2*r3^4*log(r3)+(4/3)*r3^3*V2-(8/3)*A2*r3^6-(1/2)*r3^4*d2-
(4/3)*r3^3*L2+(2/3)*r3^6*M2+r3^4*N2+2*r3^2*S2-2*r3^2*C2-r2^4*N2-2*r2^2*S2-(4/3)*r2^3*V2-
(1/4)*T2*r3^4+(8/3)*A2*r2^6+2*d2*r2^4*log(r2)-T2*r2^4*log(r2)+(1/2)*r2^4*d2+(4/3)*r2^3*L2-
(2/3)*r2^6*M2+2*r2^2*C2+(1/4)*T2*r2^4+2*B2*r2^4-2*B2*r3^4)+pi*(h1-h0)*G*k*(-T1*r2^4*log(r2)-
2*d2*r3^4*log(r3)-(8/3)*A2*r3^6-(1/2)*r3^4*d2-(4/3)*r3^3*L2-
2*r3^2*C2+(8/3)*A2*r2^6+2*d2*r2^4*log(r2)+(1/2)*r2^4*d2+(4/3)*r2^3*L2+2*r2^2*C2+(2/3)*r3^6*M1+r3^
4*N1+2*r3^2*S1+(4/3)*r3^3*V1+2*B2*r2^4-(1/4)*T1*r3^4-(2/3)*r2^6*M1-r2^4*N1-2*r2^2*S1-
(4/3)*r2^3*V1+(1/4)*T1*r2^4-2*B2*r3^4+T1*r3^4*log(r3));
        lambda2*log(r3)-(1/2)*p0*pi*r3^2+(1/2)*p0*pi*r2^2+p0*pi*r3^2*log(r3)-p0*pi*r2^2*log(r2)-
lambda1*log(r2)+pi*(h2-h1)*G*k*(2*V2*r3-2*L2*r3+(1/2)*M2*r3^4+N2*r3^2-
(1/2)*T2*r3^2+2*S2*log(r3)+T2*r3^2*log(r3)-2*d2*r3^2*log(r3)-2*B2*r3^2-2*A2*r3^4-2*V2*r2+2*L2*r2-
(1/2)*M2*r2^4-N2*r2^2+2*B2*r2^2+2*A2*r2^4-T2*r2^2*log(r2)+2*d2*r2^2*log(r2)+(1/2)*T2*r2^2-
2*S2*log(r2)+2*C2*log(r2)-2*C2*log(r3))+pi*(h1-h0)*G*k*(-2*L2*r3-2*d2*r3^2*log(r3)-2*B2*r3^2-
2*A2*r3^4+2*L2*r2+2*B2*r2^2+2*A2*r2^4+2*d2*r2^2*log(r2)+2*S1*log(r3)+(1/2)*M1*r3^4+N1*r3^2+2*V1*r
3-(1/2)*T1*r3^2-(1/2)*M1*r2^4-N1*r2^2-2*V1*r2-2*S1*log(r2)+2*C2*log(r2)+(1/2)*T1*r2^2-
2*C2*log(r3)+T1*r3^2*log(r3)-T1*r2^2*log(r2));
        lambda2*r3^2*log(r3)-(1/8)*p0*pi*r3^4+(1/8)*p0*pi*r2^4+(1/2)*p0*pi*r3^4*log(r3)-
(1/2)*p0*pi*r2^4*log(r2)-lambda1*r2^2*log(r2)+pi*(h2-h1)*G*k*((8/9)*A2*r2^6-(1/4)*r2^4*N2-
(2/9)*r2^6*M2-(8/9)*A2*r3^6-(8/3)*A2*r3^6*log(r3)+(4/3)*V2*r3^3*log(r3)+T2*r3^4*log(r3)^2-
2*B2*r3^4*log(r3)-2*C2*r3^2*log(r3)+N2*r3^4*log(r3)-N2*r2^4*log(r2)+2*B2*r2^4*log(r2)-
2*S2*r2^2*log(r2)-T2*r2^4*log(r2)^2-
(4/3)*V2*r2^3*log(r2)+(8/3)*A2*r2^6*log(r2)+2*S2*r3^2*log(r3)-
(4/3)*L2*r3^3*log(r3)+(2/3)*M2*r3^6*log(r3)+(4/3)*L2*r2^3*log(r2)-
(2/3)*M2*r2^6*log(r2)+2*C2*r2^2*log(r2)+(1/2)*B2*r2^4+2*d2*r2^4*log(r2)^2-
(1/2)*B2*r3^4+(1/4)*r3^4*N2+(2/9)*r3^3*V2+(2/9)*r2^3*L2-(2/9)*r3^3*L2-
(1/4)*r3^4*d2+(1/4)*r2^4*d2-d2*r3^4*log(r3)+d2*r2^4*log(r2)-(2/9)*r2^3*V2-
2*d2*r3^4*log(r3)^2+(2/9)*r3^6*M2)+pi*(h1-h0)*G*k*((8/9)*A2*r2^6-(1/4)*r2^4*N1+(2/9)*r3^3*V1-
(8/9)*A2*r3^6-(8/3)*A2*r3^6*log(r3)-2*B2*r3^4*log(r3)-
2*C2*r3^2*log(r3)+2*B2*r2^4*log(r2)+(8/3)*A2*r2^6*log(r2)-
(4/3)*L2*r3^3*log(r3)+(4/3)*L2*r2^3*log(r2)+2*C2*r2^2*log(r2)-(2/3)*M1*r2^6*log(r2)-
T1*r2^4*log(r2)^2+T1*r3^4*log(r3)^2+N1*r3^4*log(r3)+2*S1*r3^2*log(r3)+(4/3)*V1*r3^3*log(r3)-
2*S1*r2^2*log(r2)-N1*r2^4*log(r2)+(2/3)*M1*r3^6*log(r3)-
```

```
(4/3)*V1*r2^3*log(r2)+(1/2)*B2*r2^4+2*d2*r2^4*log(r2)^2-(1/2)*B2*r3^4+(2/9)*r2^3*L2-
(2/9)*r2^6*M1-(2/9)*r3^3*L2-(1/4)*r3^4*d2+(2/9)*r3^6*M1+(1/4)*r2^4*d2-(2/9)*r2^3*V1-
d2*r3^4*log(r3)+d2*r2^4*log(r2)+(1/4)*r3^4*N1-2*d2*r3^4*log(r3)^2);
        lambda2*r3+(2/3)*p0*pi*r3^3-(2/3)*p0*pi*r2^3-lambda1*r2+pi*(h2-h1)*G*k*(-2*C2*r3-
(4/3)*B2*r3^3+(2/3)*r3^3*N2+2*r3*S2+r3^2*V2-(4/3)*d2*r3^3*log(r3)+(2/5)*r3^5*M2-
(2/9)*d2*r3^3+(2/3)*T2*r3^3*log(r3)-(8/5)*A2*r3^5-(2/9)*T2*r3^3-
(2/3)*T2*r2^3*log(r2)+2*C2*r2+(4/3)*B2*r2^3+(2/9)*d2*r2^3+(8/5)*A2*r2^5+(4/3)*d2*r2^3*log(r2)-
(2/3)*r2^3*N2-2*r2*S2-r2^2*V2-(2/5)*r2^5*M2+(2/9)*T2*r2^3+L2*r2^2-L2*r3^2)+pi*(h1-h0)*G*k*(-
2*C2*r3-(4/3)*B2*r3^3-(4/3)*d2*r3^3*log(r3)-(2/9)*d2*r3^3-
(8/5)*A2*r3^5+2*C2*r2+(4/3)*B2*r2^3+(2/9)*d2*r2^3+(8/5)*A2*r2^5+(4/3)*d2*r2^3*log(r2)+2*r3*S1+r3^
2*V1-(2/9)*T1*r3^3-2*r2*S1-r2^2*V1+(2/5)*r3^5*M1+(2/3)*r3^3*N1+L2*r2^2+(2/9)*T1*r2^3-
(2/5)*r2^5*M1-(2/3)*r2^3*N1-L2*r3^2+(2/3)*T1*r3^3*log(r3)-(2/3)*T1*r2^3*log(r2));
        lambda2+p0*pi*r3^2-p0*pi*r2^2-lambda1;
        -lambda2*r3^4+(1/3)*p0*pi*l^6-(1/3)*p0*pi*r3^6+pi*(h3-
h2)*G*k*(4*A3*r3^8+(8/3)*d3*r3^6*log(r3)-(8/5)*a^5*L3-(8/9)*a^6*d3-(8/3)*a^6*B3-
2*a^4*C3+a^8*M3+2*a^4*S3+(4/3)*a^6*N3+(8/5)*a^5*V3-
(2/9)*T3*a^6+2*r3^4*C3+(8/5)*r3^5*L3+(8/9)*r3^6*d3+(8/3)*r3^6*B3-r3^8*M3-2*r3^4*S3-4*A3*a^8-
(4/3)*r3^6*N3-(8/5)*r3^5*V3+(2/9)*T3*r3^6-(4/3)*T3*r3^6*log(r3)-
(8/3)*d3*a^6*log(a)+(4/3)*T3*a^6*log(a))+pi*(h2-
h1)*G*k*(4*A3*r3^8+(2/9)*T2*r3^6+(4/3)*T2*a^6*log(a)-r3^8*M2-(4/3)*r3^6*N2-
2*r3^4*S2+(8/5)*a^5*V2+(8/3)*d3*r3^6*log(r3)-(8/5)*a^5*L3-(8/9)*a^6*d3-(8/3)*a^6*B3-2*a^4*C3-
(4/3)*T2*r3^6*log(r3)+2*r3^4*C3+(8/5)*r3^5*L3+(8/9)*r3^6*d3+(8/3)*r3^6*B3-
(2/9)*T2*a^6+a^8*M2+(4/3)*a^6*N2+2*a^4*S2-(8/5)*r3^5*V2-4*A3*a^8-(8/3)*d3*a^6*log(a))+pi*(h1-
h0)*G*k*(2*a^4*S1+(4/3)*a^6*N1+a^8*M1+(8/5)*a^5*V1-(2/9)*T1*a^6-2*r3^4*S1-(4/3)*r3^6*N1-r3^8*M1-
(8/5)*r3^5*V1+(2/9)*T1*r3^6+4*A3*r3^8+(8/3)*d3*r3^6*log(r3)+(4/3)*T1*a^6*log(a)-(8/5)*a^5*L3-
(8/9)*a^6*d3-(8/3)*a^6*B3-2*a^4*C3+2*r3^4*C3+(8/5)*r3^5*L3+(8/9)*r3^6*d3+(8/3)*r3^6*B3-
(4/3)*T1*r3^6*log(r3)-4*A3*a^8-(8/3)*d3*a^6*log(a))+lambda3*a^4;
        -lambda2*r3^2+(1/2)*p0*pi*l^4-(1/2)*p0*pi*r3^4+pi*(h3-h2)*G*k*(2*B3*r3^4-
(8/3)*A3*a^6+2*a^2*S3+a^4*N3+(2/3)*a^6*M3-2*d3*a^4*log(a)-(4/3)*a^3*L3+(4/3)*a^3*V3-(1/2)*a^4*d3-
2*a^2*C3-(1/4)*T3*a^4+(8/3)*A3*r3^6-2*r3^2*S3-r3^4*N3-(2/3)*r3^6*M3+T3*a^4*log(a)-
2*B3*a^4+(4/3)*r3^3*L3-(4/3)*r3^3*V3+(1/2)*r3^4*d3+2*r3^2*C3+(1/4)*T3*r3^4-
T3*r3^4*log(r3)+2*d3*r3^4*log(r3))+pi*(h2-h1)*G*k*(T2*a^4*log(a)+2*B3*r3^4-
(1/4)*T2*a^4+(2/3)*a^6*M2+a^4*N2+2*a^2*S2+(4/3)*a^3*V2-(8/3)*A3*a^6-T2*r3^4*log(r3)-
2*d3*a^4*log(a)-(4/3)*a^3*L3-(1/2)*a^4*d3-2*a^2*C3+(8/3)*A3*r3^6+(1/4)*T2*r3^4-(2/3)*r3^6*M2-
r3^4*N2-2*r3^2*S2-(4/3)*r3^3*V2-
2*B3*a^4+(4/3)*r3^3*L3+(1/2)*r3^4*d3+2*r3^2*C3+2*d3*r3^4*log(r3))+pi*(h1-h0)*G*k*(-(4/3)*r3^3*V1-
2*r3^2*S1-r3^4*N1-(2/3)*r3^6*M1+(4/3)*a^3*V1+2*a^2*S1+a^4*N1+(2/3)*a^6*M1-
(1/4)*T1*a^4+(1/4)*T1*r3^4+2*B3*r3^4+T1*a^4*log(a)-(8/3)*A3*a^6-2*d3*a^4*log(a)-(4/3)*a^3*L3-
(1/2)*a^4*d3-2*a^2*C3+(8/3)*A3*r3^6-T1*r3^4*log(r3)-
2*B3*a^4+(4/3)*r3^3*L3+(1/2)*r3^4*d3+2*r3^2*C3+2*d3*r3^4*log(r3))+lambda3*a^2;
        -lambda2*log(r3)-(1/2)*p0*pi*l^2+(1/2)*p0*pi*r3^2+p0*pi*l^2*log(l)-
p0*pi*r3^2*log(r3)+pi*(h3-h2)*G*k*(2*C3*log(r3)-2*A3*a^4+2*S3*log(a)-2*L3*a+2*V3*a-2*B3*a^2-
(1/2)*T3*a^2+(1/2)*M3*a^4+N3*a^2+2*A3*r3^4-2*S3*log(r3)-2*C3*log(a)+2*L3*r3-
2*V3*r3+2*B3*r3^2+(1/2)*T3*r3^2-(1/2)*M3*r3^4-N3*r3^2-
T3*r3^2*log(r3)+2*d3*r3^2*log(r3)+T3*a^2*log(a)-2*d3*a^2*log(a))+pi*(h2-h1)*G*k*(-
T2*r3^2*log(r3)+2*C3*log(r3)+(1/2)*T2*r3^2-2*S2*log(r3)-(1/2)*M2*r3^4-N2*r3^2+T2*a^2*log(a)-
2*V2*r3-2*A3*a^4-2*L3*a-2*B3*a^2+2*A3*r3^4-(1/2)*T2*a^2+2*S2*log(a)+(1/2)*M2*a^4+N2*a^2+2*V2*a-
2*C3*log(a)+2*L3*r3+2*B3*r3^2+2*d3*r3^2*log(r3)-2*d3*a^2*log(a))+pi*(h1-h0)*G*k*(2*S1*log(a)-
(1/2)*M1*r3^4+2*V1*a+N1*a^2+(1/2)*M1*a^4+(1/2)*T1*r3^2+2*C3*log(r3)-2*A3*a^4-(1/2)*T1*a^2-
2*V1*r3-2*S1*log(r3)+T1*a^2*log(a)-2*L3*a-2*B3*a^2+2*A3*r3^4-N1*r3^2-T1*r3^2*log(r3)-
2*C3*log(a)+2*L3*r3+2*B3*r3^2+2*d3*r3^2*log(r3)-2*d3*a^2*log(a))+lambda3*log(a);
        -lambda2*r3^2*log(r3)-(1/8)*p0*pi*l^4+(1/8)*p0*pi*r3^4+(1/2)*p0*pi*l^4*log(l)-
(1/2)*p0*pi*r3^4*log(r3)+pi*(h3-h2)*G*k*(-(4/3)*V3*r3^3*log(r3)+2*C3*r3^2*log(r3)-
T3*r3^4*log(r3)^2+(2/3)*M3*a^6*log(a)-2*B3*a^4*log(a)-2*S3*r3^2*log(r3)-(2/3)*M3*r3^6*log(r3)-
N3*r3^4*log(r3)+2*B3*r3^4*log(r3)-2*C3*a^2*log(a)-(8/3)*A3*a^6*log(a)+T3*a^4*log(a)^2-
(4/3)*L3*a^3*log(a)+2*S3*a^2*log(a)+(4/3)*L3*r3^3*log(r3)+(4/3)*V3*a^3*log(a)+(1/2)*B3*r3^4-
(1/2)*B3*a^4-(2/9)*a^3*L3-(2/9)*r3^6*M3+(2/9)*a^6*M3-(2/9)*r3^3*V3-
(1/4)*a^4*d3+(2/9)*a^3*V3+(1/4)*a^4*N3-(1/4)*r3^4*N3+(2/9)*r3^3*L3+(8/3)*A3*r3^6*log(r3)-
d3*a^4*log(a)+d3*r3^4*log(r3)+N3*a^4*log(a)+(8/9)*A3*r3^6-
(8/9)*A3*a^6+(1/4)*r3^4*d3+2*d3*r3^4*log(r3)^2-2*d3*a^4*log(a)^2)+pi*(h2-
h1)*G*k*(2*C3*r3^2*log(r3)-2*B3*a^4*log(a)+2*B3*r3^4*log(r3)-2*C3*a^2*log(a)-(8/3)*A3*a^6*log(a)-
(4/3)*L3*a^3*log(a)+(4/3)*L3*r3^3*log(r3)+(4/3)*V2*a^3*log(a)-(4/3)*V2*r3^3*log(r3)-
T2*r3^4*log(r3)^2-2*S2*r3^2*log(r3)+(2/3)*M2*a^6*log(a)-(2/3)*M2*r3^6*log(r3)-
N2*r3^4*log(r3)+2*S2*a^2*log(a)+T2*a^4*log(a)^2+N2*a^4*log(a)+(1/2)*B3*r3^4-
(1/2)*B3*a^4+(2/9)*a^6*M2-(2/9)*a^3*L3-(2/9)*r3^3*V2-
(1/4)*a^4*d3+(1/4)*a^4*N2+(2/9)*r3^3*L3+(8/3)*A3*r3^6*log(r3)-
d3*a^4*log(a)+d3*r3^4*log(r3)+(8/9)*A3*r3^6-(8/9)*A3*a^6+(2/9)*a^3*V2-
(1/4)*r3^4*N2+(1/4)*r3^4*d3-(2/9)*r3^6*M2+2*d3*r3^4*log(r3)^2-2*d3*a^4*log(a)^2)+pi*(h1-
h0)*G*k*((1/4)*a^4*N1+2*C3*r3^2*log(r3)-2*B3*a^4*log(a)+2*B3*r3^4*log(r3)-2*C3*a^2*log(a)-
```

```
(8/3)*A3*a^6*log(a)-
(4/3)*L3*a^3*log(a)+(4/3)*L3*r3^3*log(r3)+T1*a^4*log(a)^2+N1*a^4*log(a)+(2/3)*M1*a^6*log(a)+2*S1*
a^2*log(a)-T1*r3^4*log(r3)^2-N1*r3^4*log(r3)+(4/3)*V1*a^3*log(a)-(4/3)*V1*r3^3*log(r3)-
(2/3)*M1*r3^6*log(r3)-2*S1*r3^2*log(r3)+(1/2)*B3*r3^4-(1/2)*B3*a^4-(2/9)*a^3*L3+(2/9)*a^6*M1-
(2/9)*r3^3*V1-(1/4)*r3^4*N1-(2/9)*r3^6*M1-(1/4)*a^4*d3+(2/9)*r3^3*L3+(8/3)*A3*r3^6*log(r3)-
d3*a^4*log(a)+d3*r3^4*log(r3)+(8/9)*A3*r3^6-
(8/9)*A3*a^6+(2/9)*a^3*V1+(1/4)*r3^4*d3+2*d3*r3^4*log(r3)^2-
2*d3*a^4*log(a)^2)+lambda3*a^2*log(a);
        -lambda2*r3+(2/3)*p0*pi*l^3-(2/3)*p0*pi*r3^3+pi*(h3-h2)*G*k*(L3*r3^2-(2/3)*r3^3*N3-
(8/5)*A3*a^5+(2/5)*a^5*M3+(2/3)*a^3*N3+a^2*V3+2*a*S3-(4/3)*B3*a^3-2*C3*a-(2/9)*d3*a^3-
(2/9)*T3*a^3+(8/5)*A3*r3^5-(2/5)*r3^5*M3-r3^2*V3-2*r3*S3-
L3*a^2+(4/3)*B3*r3^3+2*C3*r3+(2/9)*d3*r3^3+(2/9)*T3*r3^3+(2/3)*T3*a^3*log(a)-
(2/3)*T3*r3^3*log(r3)+(4/3)*d3*r3^3*log(r3)-(4/3)*d3*a^3*log(a))+pi*(h2-h1)*G*k*(L3*r3^2-r3^2*V2-
(2/3)*r3^3*N2-2*r3*S2+(2/3)*T2*a^3*log(a)-(2/5)*r3^5*M2+(2/9)*T2*r3^3-(8/5)*A3*a^5-
(2/3)*T2*r3^3*log(r3)-(4/3)*B3*a^3-2*C3*a-(2/9)*d3*a^3+(8/5)*A3*r3^5-
(2/9)*T2*a^3+a^2*V2+(2/3)*a^3*N2+2*a*S2+(2/5)*a^5*M2-
L3*a^2+(4/3)*B3*r3^3+2*C3*r3+(2/9)*d3*r3^3+(4/3)*d3*r3^3*log(r3)-(4/3)*d3*a^3*log(a))+pi*(h1-
h0)*G*k*(a^2*V1+2*a*S1-(2/3)*r3^3*N1-(2/5)*r3^5*M1+(2/3)*a^3*N1+(2/5)*a^5*M1+L3*r3^2-
(8/5)*A3*a^5+(2/9)*T1*r3^3-r3^2*V1-2*r3*S1+(2/3)*T1*a^3*log(a)-(2/9)*T1*a^3-(4/3)*B3*a^3-2*C3*a-
(2/9)*d3*a^3+(8/5)*A3*r3^5-(2/3)*T1*r3^3*log(r3)-
L3*a^2+(4/3)*B3*r3^3+2*C3*r3+(2/9)*d3*r3^3+(4/3)*d3*r3^3*log(r3)-(4/3)*d3*a^3*log(a))+lambda3*a;
        -lambda2+p0*pi*l^2-p0*pi*r3^2+lambda3;
        (1/72)*pi*(h1-h0)^3*E*(12*T1*r1^6*log(r1)*a^2-
12*T1*a^6*log(a)*r1^2+12*nu*V1*r1^5*a^2+12*nu*S1*r1^4*a^2-12*nu*V1*a^5*r1^2-12*nu*S1*a^4*r1^2-
12*nu*N1*a^6*r1^2+12*nu*N1*r1^6*a^2-20*M1*a^8*r1^2+20*M1*r1^8*a^2+12*S1*a^4*r1^2-4*V1*a^5*r1^2-
12*N1*a^6*r1^2-12*nu*M1*a^8*r1^2-6*T1*a^6*r1^2+12*nu*M1*r1^8*a^2+6*T1*r1^6*a^2-
12*S1*r1^4*a^2+12*N1*r1^6*a^2+4*V1*r1^5*a^2-
12*nu*T1*a^6*log(a)*r1^2+12*nu*T1*r1^6*log(r1)*a^2)/((1-nu^2)*a^2*r1^2)+pi*(h1-h0)*G*k*(-
(1/2)*a^4*S1-(1/3)*a^6*N1-(1/2)*C3*r3^4-(2/3)*B3*r3^6-
(2/5)*L3*r3^5+(2/9)*d3*a^6+(1/2)*C3*a^4+(2/3)*B3*a^6+(2/5)*L3*a^5+A3*a^8-(2/5)*a^5*V1-
A3*r3^8+(2/5)*r3^5*V1-
(1/18)*T1*r3^6+(2/3)*d3*a^6*log(a)+(1/18)*T1*a^6+(1/2)*r3^4*S1+(1/3)*r3^6*N1-(2/9)*d3*r3^6-
(1/4)*M1*a^8-(2/3)*d3*r3^6*log(r3)+(1/3)*T1*r3^6*log(r3)+(1/4)*M1*r3^8-
(1/3)*T1*a^6*log(a))+pi*(h1-h0)*G*k*(-
(2/3)*d2*r2^6*log(r2)+(2/3)*B2*r3^6+(1/2)*C2*r3^4+(2/9)*d2*r3^6-(1/18)*T1*r2^6-A2*r2^8-
(2/3)*B2*r2^6-(1/2)*C2*r2^4-
(2/9)*d2*r2^6+(1/2)*r2^4*S1+(1/4)*M1*r2^8+(2/5)*L2*r3^5+(1/3)*r2^6*N1+(2/5)*r2^5*V1-
(2/5)*L2*r2^5-(2/5)*r3^5*V1+(1/18)*T1*r3^6-(1/2)*r3^4*S1-
(1/3)*r3^6*N1+(2/3)*d2*r3^6*log(r3)+(1/3)*T1*r2^6*log(r2)-(1/3)*T1*r3^6*log(r3)-
(1/4)*M1*r3^8+A2*r3^8)+pi*(h1-h0)*G*k*(-A1*r1^8-(2/9)*d1*r1^6-
(1/18)*T1*r1^6+A1*r2^8+(2/5)*L1*r2^5+(1/2)*C1*r2^4+(2/3)*B1*r2^6+(2/9)*d1*r2^6-
(2/3)*B1*r1^6+(1/4)*M1*r1^8+(1/2)*r1^4*S1+(1/3)*r1^6*N1+(2/5)*r1^5*V1-(2/5)*L1*r1^5-
(1/2)*C1*r1^4+(2/3)*d1*r2^6*log(r2)+(1/3)*T1*r1^6*log(r1)-(2/3)*d1*r1^6*log(r1)+(1/18)*T1*r2^6-
(1/2)*r2^4*S1-(1/4)*M1*r2^8-(1/3)*r2^6*N1-(2/5)*r2^5*V1-(1/3)*T1*r2^6*log(r2))+lambda4*a^3;
        (1/72)*pi*(h1-
h0)^3*E*(12*nu*T1*r1^4*log(r1)*a^2+12*T1*r1^4*log(r1)*a^2+12*nu*V1*r1^3*a^2-12*nu*V1*a^3*r1^2-
12*nu*M1*a^6*r1^2+12*nu*M1*r1^6*a^2-12*N1*a^4*r1^2+12*N1*r1^4*a^2-12*V1*a^3*r1^2-
12*M1*a^6*r1^2+12*nu*N1*r1^4*a^2+12*V1*r1^3*a^2-12*nu*N1*a^4*r1^2+12*M1*r1^6*a^2-
12*T1*a^4*log(a)*r1^2-12*nu*T1*a^4*log(a)*r1^2)/((1-nu^2)*a^2*r1^2)+pi*(h1-h0)*G*k*(-
(1/3)*M1*a^6-(2/3)*a^3*V1-a^2*S1-B3*r3^4-(2/3)*L3*r3^3-
(1/4)*d3*r3^4+C3*a^2+B3*a^4+(2/3)*L3*a^3+(1/4)*d3*a^4+(4/3)*A3*a^6+(1/8)*T1*a^4-(4/3)*A3*r3^6-
(1/8)*T1*r3^4-(1/2)*T1*a^4*log(a)+d3*a^4*log(a)+(1/3)*M1*r3^6+(2/3)*r3^3*V1+r3^2*S1-(1/2)*N1*a^4-
C3*r3^2-d3*r3^4*log(r3)+(1/2)*T1*r3^4*log(r3)+(1/2)*N1*r3^4)+pi*(h1-
h0)*G*k*(B2*r3^4+C2*r3^2+(1/4)*d2*r3^4-(1/8)*T1*r2^4-(4/3)*A2*r2^6-B2*r2^4-C2*r2^2-
(1/4)*d2*r2^4+(1/2)*N1*r2^4+(2/3)*L2*r3^3+(1/3)*M1*r2^6+(2/3)*r2^3*V1+r2^2*S1+(1/2)*T1*r2^4*log(r
2)-(2/3)*L2*r2^3+(1/8)*T1*r3^4+d2*r3^4*log(r3)-(1/3)*M1*r3^6-(2/3)*r3^3*V1-r3^2*S1-
d2*r2^4*log(r2)-(1/2)*T1*r3^4*log(r3)-(1/2)*N1*r3^4+(4/3)*A2*r3^6)+pi*(h1-h0)*G*k*(-
(1/4)*d1*r1^4+(4/3)*A1*r2^6+C1*r2^2+B1*r2^4+(2/3)*L1*r2^3+(1/4)*d1*r2^4+(1/2)*N1*r1^4-B1*r1^4-
(2/3)*L1*r1^3+r1^2*S1+(2/3)*r1^3*V1-(1/8)*T1*r1^4+(1/3)*M1*r1^6-
C1*r1^2+d1*r2^4*log(r2)+(1/2)*T1*r1^4*log(r1)-d1*r1^4*log(r1)+(1/8)*T1*r2^4-(1/2)*N1*r2^4-
(1/3)*M1*r2^6-(2/3)*r2^3*V1-r2^2*S1-(1/2)*T1*r2^4*log(r2)-(4/3)*A1*r1^6)+lambda4*a;
        (1/72)*pi*(h1-h0)^3*E*(12*nu*V1*a^2*r1-12*nu*S1*r1^2-12*nu*V1*a*r1^2-
12*T1*log(r1)*a^2*r1^2+12*T1*log(a)*a^2*r1^2+12*nu*M1*r1^4*a^2-
12*nu*M1*a^4*r1^2+12*V1*a*r1^2+12*nu*S1*a^2-12*V1*a^2*r1+12*r1^2*S1-12*a^2*S1+12*M1*a^4*r1^2-
12*M1*r1^4*a^2-12*nu*T1*log(a)*a^2*r1^2+12*nu*T1*log(r1)*a^2*r1^2)/((1-nu^2)*a^2*r1^2)+pi*(h1-
h0)*G*k*(-T1*a^2*log(a)-(1/2)*M1*a^4+2*L3*a+(1/2)*T1*a^2+2*C3*log(a)-2*V1*a-N1*a^2-
2*B3*r3^2+2*B3*a^2+2*A3*a^4-2*A3*r3^4+2*d3*a^2*log(a)+(1/2)*M1*r3^4-2*L3*r3-(1/2)*T1*r3^2-
2*C3*log(r3)+2*V1*r3+N1*r3^2-2*S1*log(a)-2*d3*r3^2*log(r3)+T1*r3^2*log(r3)+2*S1*log(r3))+pi*(h1-
h0)*G*k*(2*C2*log(r3)+2*L2*r3+2*B2*r3^2-(1/2)*T1*r2^2-2*A2*r2^4-2*B2*r2^2-2*C2*log(r2)-
```

```
2*L2*r2+(1/2)*M1*r2^4+2*S1*log(r2)+2*V1*r2+N1*r2^2+T1*r2^2*log(r2)+2*d2*r3^2*log(r3)-
(1/2)*M1*r3^4+(1/2)*T1*r3^2-2*V1*r3-N1*r3^2-2*d2*r2^2*log(r2)-T1*r3^2*log(r3)-
2*S1*log(r3)+2*A2*r3^4)+pi*(h1-h0)*G*k*(-2*L1*r1-
2*C1*log(r1)+2*A1*r2^4+2*C1*log(r2)+2*B1*r2^2+2*L1*r2+2*S1*log(r1)-
2*B1*r1^2+N1*r1^2+2*d1*r2^2*log(r2)+2*V1*r1+(1/2)*M1*r1^4-2*A1*r1^4-
(1/2)*T1*r1^2+T1*r1^2*log(r1)-2*d1*r1^2*log(r1)+(1/2)*T1*r2^2-(1/2)*M1*r2^4-2*S1*log(r2)-2*V1*r2-
N1*r2^2-T1*r2^2*log(r2))+lambda4/a;
        (1/72)*pi*(h1-h0)^3*E*(12*N1*r1^4*log(r1)*a^2+12*nu*N1*r1^4*log(r1)*a^2-
12*M1*a^6*log(a)*r1^2-
12*V1*a^3*log(a)*r1^2+12*nu*T1*r1^4*log(r1)^2*a^2+12*S1*log(a)*a^2*r1^2+12*M1*r1^6*log(r1)*a^2+6*
T1*r1^4*a^2-12*nu*T1*a^4*log(a)^2*r1^2-12*S1*log(r1)*a^2*r1^2-12*N1*a^4*log(a)*r1^2-
12*nu*S1*log(a)*a^2*r1^2-6*T1*a^4*r1^2+12*nu*S1*log(r1)*a^2*r1^2+12*V1*r1^3*log(r1)*a^2-
12*nu*N1*a^4*log(a)*r1^2-12*nu*V1*a^3*log(a)*r1^2-
12*nu*M1*a^6*log(a)*r1^2+12*nu*V1*r1^3*log(r1)*a^2+12*nu*M1*r1^6*log(r1)*a^2+12*V1*a^3*r1^2-
12*T1*a^4*log(a)^2*r1^2+12*T1*r1^4*log(r1)^2*a^2-6*M1*a^6*r1^2+6*M1*r1^6*a^2-12*V1*r1^3*a^2)/((1-
nu^2)*a^2*r1^2)+pi*(h1-h0)*G*k*((1/8)*N1*a^4-
(1/8)*N1*r3^4+(1/2)*a^2*S1+(1/18)*M1*a^6+(1/2)*C3*r3^2+(1/4)*B3*r3^4-
(1/4)*B3*a^4+(2/9)*a^3*V1+(2/9)*L3*r3^3-(1/2)*r3^2*S1-(1/2)*C3*a^2+(2/9)*A3*r3^6-(2/9)*r3^3*V1-
(2/9)*A3*a^6-(1/18)*M1*r3^6-S1*a^2*log(a)-
(1/2)*N1*a^4*log(a)+(4/3)*A3*a^6*log(a)+B3*a^4*log(a)+d3*a^4*log(a)^2+C3*a^2*log(a)+(2/3)*L3*a^3*
log(a)-d3*r3^4*log(r3)^2-B3*r3^4*log(r3)-
(4/3)*A3*r3^6*log(r3)+(2/3)*V1*r3^3*log(r3)+S1*r3^2*log(r3)+(1/2)*N1*r3^4*log(r3)-
(1/16)*T1*a^4+(1/3)*M1*r3^6*log(r3)-(2/3)*L3*r3^3*log(r3)-C3*r3^2*log(r3)-(1/3)*M1*a^6*log(a)-
(2/3)*V1*a^3*log(a)+(1/16)*T1*r3^4-(1/2)*T1*a^4*log(a)^2-(2/9)*L3*a^3+(1/4)*T1*a^4*log(a)-
(1/4)*T1*r3^4*log(r3)+(1/2)*T1*r3^4*log(r3)^2)+pi*(h1-h0)*G*k*(-(1/2)*C2*r3^2+(1/8)*N1*r3^4-
(1/8)*N1*r2^4-(1/18)*M1*r2^6+(2/9)*A2*r2^6-(1/4)*B2*r3^4-
(2/9)*A2*r3^6+(1/16)*T1*r2^4+(1/2)*C2*r2^2-
(2/9)*L2*r3^3+(2/9)*L2*r2^3+(1/2)*r3^2*S1+(1/4)*B2*r2^4-
(1/2)*r2^2*S1+(2/9)*r3^3*V1+(1/18)*M1*r3^6-(2/3)*V1*r3^3*log(r3)-S1*r3^2*log(r3)-
(1/2)*N1*r3^4*log(r3)-(1/3)*M1*r3^6*log(r3)-
B2*r2^4*log(r2)+(1/2)*N1*r2^4*log(r2)+S1*r2^2*log(r2)+(2/3)*L2*r3^3*log(r3)+C2*r3^2*log(r3)+d2*r3
^4*log(r3)^2+(4/3)*A2*r3^6*log(r3)-d2*r2^4*log(r2)^2-(1/16)*T1*r3^4-
(4/3)*A2*r2^6*log(r2)+B2*r3^4*log(r3)+(1/3)*M1*r2^6*log(r2)-(2/3)*L2*r2^3*log(r2)-
C2*r2^2*log(r2)+(2/3)*V1*r2^3*log(r2)-(2/9)*r2^3*V1-
(1/4)*T1*r2^4*log(r2)+(1/4)*T1*r3^4*log(r3)+(1/2)*T1*r2^4*log(r2)^2-
(1/2)*T1*r3^4*log(r3)^2)+pi*(h1-h0)*G*k*((1/8)*N1*r2^4-(1/8)*N1*r1^4+(1/18)*M1*r2^6-
(1/4)*B1*r2^4-(1/18)*M1*r1^6-(1/16)*T1*r2^4+(1/2)*C1*r1^2+(1/16)*T1*r1^4-
(1/2)*r1^2*S1+(1/4)*B1*r1^4-(1/2)*C1*r2^2+(2/9)*L1*r1^3-
(2/9)*r1^3*V1+(2/9)*A1*r1^6+(1/2)*r2^2*S1-(2/9)*L1*r2^3-(1/2)*N1*r2^4*log(r2)-S1*r2^2*log(r2)-
(2/9)*A1*r2^6-(1/3)*M1*r2^6*log(r2)-(2/3)*V1*r2^3*log(r2)+(2/9)*r2^3*V1+(1/4)*T1*r2^4*log(r2)-
(1/2)*T1*r2^4*log(r2)^2-
(1/4)*T1*r1^4*log(r1)+(1/2)*T1*r1^4*log(r1)^2+B1*r2^4*log(r2)+(2/3)*L1*r2^3*log(r2)+d1*r2^4*log(r
2)^2+C1*r2^2*log(r2)+(4/3)*A1*r2^6*log(r2)+S1*r1^2*log(r1)-
d1*r1^4*log(r1)^2+(2/3)*V1*r1^3*log(r1)+(1/3)*M1*r1^6*log(r1)-(2/3)*L1*r1^3*log(r1)-
B1*r1^4*log(r1)+(1/2)*N1*r1^4*log(r1)-(4/3)*A1*r1^6*log(r1)-C1*r1^2*log(r1))+lambda4*a*log(a);
        (1/72)*pi*(h1-h0)^3*E*(12*nu*S1*a^2*r1-12*nu*S1*a*r1^2-
12*V1*log(a)*a^2*r1^2+12*V1*log(r1)*a^2*r1^2+12*nu*M1*r1^5*a^2+12*nu*N1*r1^3*a^2-
12*nu*M1*a^5*r1^2-12*nu*N1*a^3*r1^2+12*S1*a*r1^2-12*S1*a^2*r1-12*N1*a^3*r1^2-
4*M1*a^5*r1^2+12*T1*a^3*r1^2+12*N1*r1^3*a^2-12*T1*r1^3*a^2+4*M1*r1^5*a^2-
12*T1*a^3*log(a)*r1^2+12*T1*r1^3*log(r1)*a^2-
12*nu*T1*a^3*log(a)*r1^2+12*nu*T1*r1^3*log(r1)*a^2)/((1-nu^2)*a^2*r1^2)+pi*(h1-h0)*G*k*(L3*a^2-
2*S1*a-(2/3)*N1*a^3-(4/3)*B3*r3^3-(2/9)*d3*r3^3+2*C3*a+(4/3)*B3*a^3+(2/9)*d3*a^3+(8/5)*A3*a^5-
(2/5)*M1*a^5-(8/5)*A3*r3^5+(2/5)*M1*r3^5-(2/9)*T1*r3^3+(2/9)*T1*a^3-2*C3*r3-
L3*r3^2+2*S1*r3+(2/3)*N1*r3^3-V1*a^2+(4/3)*d3*a^3*log(a)-
(4/3)*d3*r3^3*log(r3)+(2/3)*T1*r3^3*log(r3)+V1*r3^2-(2/3)*T1*a^3*log(a))+pi*(h1-
h0)*G*k*(L2*r3^2+(4/3)*B2*r3^3+2*C2*r3-(8/5)*A2*r2^5-(4/3)*B2*r2^3-2*C2*r2-
L2*r2^2+V1*r2^2+(2/9)*d2*r3^3-(2/9)*T1*r2^3+2*S1*r2+(2/3)*N1*r2^3+(2/5)*M1*r2^5-(2/9)*d2*r2^3-
(2/5)*M1*r3^5+(2/9)*T1*r3^3+(4/3)*d2*r3^3*log(r3)-2*S1*r3-(2/3)*N1*r3^3+(2/3)*T1*r2^3*log(r2)-
(4/3)*d2*r2^3*log(r2)-(2/3)*T1*r3^3*log(r3)-V1*r3^2+(8/5)*A2*r3^5)+pi*(h1-h0)*G*k*(-
(4/3)*B1*r1^3-(2/9)*d1*r1^3-(8/5)*A1*r1^5-
L1*r1^2+(4/3)*B1*r2^3+(8/5)*A1*r2^5+2*C1*r2+L1*r2^2+(2/9)*d1*r2^3+V1*r1^2+(2/3)*N1*r1^3+2*S1*r1-
(2/9)*T1*r1^3+(2/5)*M1*r1^5-2*C1*r1+(4/3)*d1*r2^3*log(r2)-
(4/3)*d1*r1^3*log(r1)+(2/3)*T1*r1^3*log(r1)-V1*r2^2+(2/9)*T1*r2^3-2*S1*r2-(2/3)*N1*r2^3-
(2/5)*M1*r2^5-(2/3)*T1*r2^3*log(r2))+lambda4;
        (1/72)*pi*(h2-h1)^3*E*(12*nu*S2*r2^4*a^2+12*nu*V2*r2^5*a^2-
12*nu*M2*a^8*r2^2+20*M2*r2^8*a^2-20*M2*a^8*r2^2-12*nu*N2*a^6*r2^2-12*nu*S2*a^4*r2^2-
12*nu*V2*a^5*r2^2+12*nu*N2*r2^6*a^2-6*T2*a^6*r2^2+12*S2*a^4*r2^2-12*N2*a^6*r2^2-
4*V2*a^5*r2^2+4*V2*r2^5*a^2+12*nu*M2*r2^8*a^2+12*N2*r2^6*a^2+12*nu*T2*r2^6*log(r2)*a^2+12*T2*r2^6
*log(r2)*a^2-12*nu*T2*a^6*log(a)*r2^2-12*T2*a^6*log(a)*r2^2-12*S2*r2^4*a^2+6*T2*r2^6*a^2)/((1-
```

```
nu^2)*a^2*r2^2)+pi*(h2-
h1)*G*k*((2/3)*d3*a^6*log(a)+(1/3)*T2*r3^6*log(r3)+(1/4)*M2*r3^8+(1/3)*r3^6*N2+(1/2)*r3^4*S2+(2/5
)*r3^5*V2-(2/5)*L3*r3^5-(2/9)*d3*r3^6-(1/2)*C3*r3^4-A3*r3^8+(2/3)*B3*a^6-
(1/4)*M2*a^8+(1/18)*T2*a^6-(1/3)*a^6*N2-(1/2)*a^4*S2-
(2/5)*a^5*V2+(2/5)*L3*a^5+(2/9)*d3*a^6+(1/2)*C3*a^4+A3*a^8-(1/18)*T2*r3^6-(2/3)*B3*r3^6-
(2/3)*d3*r3^6*log(r3)-(1/3)*T2*a^6*log(a))+pi*(h2-
h1)*G*k*(A2*r3^8+(1/2)*C2*r3^4+(2/3)*B2*r3^6+(2/3)*d2*r3^6*log(r3)+(2/5)*L2*r3^5+(2/9)*d2*r3^6+(1
/3)*T2*r2^6*log(r2)-(1/2)*C2*r2^4-(2/3)*B2*r2^6-(2/9)*d2*r2^6-A2*r2^8-(2/3)*d2*r2^6*log(r2)-
(2/5)*L2*r2^5-(1/18)*T2*r2^6+(2/5)*r2^5*V2+(1/2)*r2^4*S2+(1/3)*r2^6*N2+(1/4)*M2*r2^8-
(1/3)*T2*r3^6*log(r3)-(1/4)*M2*r3^8-(1/3)*r3^6*N2-(1/2)*r3^4*S2-
(2/5)*r3^5*V2+(1/18)*T2*r3^6)+lambda5*a^3;
        (1/72)*pi*(h2-h1)^3*E*(12*N2*r2^4*a^2-12*N2*a^4*r2^2-12*nu*M2*a^6*r2^2-12*nu*V2*a^3*r2^2-
12*nu*N2*a^4*r2^2+12*nu*V2*r2^3*a^2+12*nu*M2*r2^6*a^2-12*M2*a^6*r2^2-
12*V2*a^3*r2^2+12*V2*r2^3*a^2+12*T2*r2^4*log(r2)*a^2+12*nu*N2*r2^4*a^2+12*M2*r2^6*a^2-
12*T2*a^4*log(a)*r2^2+12*nu*T2*r2^4*log(r2)*a^2-12*nu*T2*a^4*log(a)*r2^2)/((1-
nu^2)*a^2*r2^2)+pi*(h2-h1)*G*k*(-
(1/2)*T2*a^4*log(a)+(1/2)*T2*r3^4*log(r3)+d3*a^4*log(a)+(1/2)*N2*r3^4+r3^2*S2+(2/3)*r3^3*V2+(1/3)
*M2*r3^6-(2/3)*L3*r3^3-C3*r3^2-(1/4)*d3*r3^4-(4/3)*A3*r3^6+(1/8)*T2*a^4+B3*a^4-(1/2)*N2*a^4-
a^2*S2-(2/3)*a^3*V2-(1/3)*M2*a^6+(2/3)*L3*a^3+C3*a^2+(1/4)*d3*a^4-d3*r3^4*log(r3)+(4/3)*A3*a^6-
(1/8)*T2*r3^4-B3*r3^4)+pi*(h2-
h1)*G*k*(d2*r3^4*log(r3)+(4/3)*A2*r3^6+C2*r3^2+B2*r3^4+(2/3)*L2*r3^3+(1/4)*d2*r3^4-
B2*r2^4+(1/2)*T2*r2^4*log(r2)-C2*r2^2-d2*r2^4*log(r2)-(1/4)*d2*r2^4-
(4/3)*A2*r2^6+(2/3)*r2^3*V2+r2^2*S2-(2/3)*L2*r2^3+(1/3)*M2*r2^6-(1/8)*T2*r2^4+(1/2)*N2*r2^4-
(1/2)*T2*r3^4*log(r3)-(1/2)*N2*r3^4-r3^2*S2-(2/3)*r3^3*V2-(1/3)*M2*r3^6+(1/8)*T2*r3^4)+lambda5*a;
        (1/72)*pi*(h2-h1)^3*E*(12*nu*M2*r2^4*a^2-12*nu*M2*a^4*r2^2+12*nu*S2*a^2-12*V2*a^2*r2-
12*nu*S2*r2^2+12*V2*a*r2^2+12*nu*V2*a^2*r2-12*nu*V2*a*r2^2+12*M2*a^4*r2^2-
12*T2*log(r2)*a^2*r2^2+12*r2^2*S2+12*T2*log(a)*a^2*r2^2-12*nu*T2*log(a)*a^2*r2^2-
12*a^2*S2+12*nu*T2*log(r2)*a^2*r2^2-12*M2*r2^4*a^2)/((1-nu^2)*a^2*r2^2)+pi*(h2-h1)*G*k*(-
2*d3*r3^2*log(r3)+T2*r3^2*log(r3)-T2*a^2*log(a)+2*S2*log(r3)+N2*r3^2+2*V2*r3-
2*C3*log(r3)+(1/2)*M2*r3^4-2*L3*r3+2*d3*a^2*log(a)-(1/2)*T2*r3^2-2*A3*r3^4+2*B3*a^2-2*S2*log(a)-
N2*a^2-2*V2*a+2*C3*log(a)-(1/2)*M2*a^4+2*L3*a+(1/2)*T2*a^2+2*A3*a^4-2*B3*r3^2)+pi*(h2-
h1)*G*k*(2*A2*r3^4+2*B2*r3^2+2*C2*log(r3)+2*L2*r3+2*d2*r3^2*log(r3)-2*B2*r2^2-
2*d2*r2^2*log(r2)+T2*r2^2*log(r2)-2*A2*r2^4+2*V2*r2+N2*r2^2-(1/2)*T2*r2^2-2*C2*log(r2)-
2*L2*r2+(1/2)*M2*r2^4+2*S2*log(r2)-T2*r3^2*log(r3)-2*S2*log(r3)-N2*r3^2-2*V2*r3-
(1/2)*M2*r3^4+(1/2)*T2*r3^2)+lambda5/a;
        (1/72)*pi*(h2-h1)^3*E*(-6*T2*a^4*r2^2-12*nu*S2*log(a)*a^2*r2^2-
12*T2*a^4*log(a)^2*r2^2+12*T2*r2^4*log(r2)^2*a^2+12*N2*r2^4*log(r2)*a^2+6*T2*r2^4*a^2+12*nu*M2*r2
^6*log(r2)*a^2+12*nu*V2*r2^3*log(r2)*a^2-12*V2*a^3*log(a)*r2^2+12*nu*S2*log(r2)*a^2*r2^2-
12*N2*a^4*log(a)*r2^2-12*nu*T2*a^4*log(a)^2*r2^2+12*nu*N2*r2^4*log(r2)*a^2-12*M2*a^6*log(a)*r2^2-
12*S2*log(r2)*a^2*r2^2+12*nu*T2*r2^4*log(r2)^2*a^2+12*S2*log(a)*a^2*r2^2+12*M2*r2^6*log(r2)*a^2-
12*nu*M2*a^6*log(a)*r2^2-12*nu*N2*a^4*log(a)*r2^2+12*V2*r2^3*log(r2)*a^2-
12*nu*V2*a^3*log(a)*r2^2-6*M2*a^6*r2^2+12*V2*a^3*r2^2-12*V2*r2^3*a^2+6*M2*r2^6*a^2)/((1-
nu^2)*a^2*r2^2)+pi*(h2-h1)*G*k*((1/2)*C3*r3^2+(1/4)*B3*r3^4-
(1/4)*B3*a^4+(1/2)*T2*r3^4*log(r3)^2+(2/9)*L3*r3^3-(1/2)*C3*a^2+(2/9)*A3*r3^6-(2/9)*A3*a^6-
C3*r3^2*log(r3)-d3*r3^4*log(r3)^2-(1/2)*N2*a^4*log(a)-(4/3)*A3*r3^6*log(r3)-B3*r3^4*log(r3)-
(2/3)*L3*r3^3*log(r3)+(2/3)*V2*r3^3*log(r3)-(1/3)*M2*a^6*log(a)+(1/2)*N2*r3^4*log(r3)-
S2*a^2*log(a)+(1/3)*M2*r3^6*log(r3)+d3*a^4*log(a)^2+B3*a^4*log(a)-
(2/3)*V2*a^3*log(a)+(2/3)*L3*a^3*log(a)+C3*a^2*log(a)+(4/3)*A3*a^6*log(a)-
(1/8)*N2*r3^4+(1/8)*N2*a^4+(2/9)*a^3*V2-(1/2)*r3^2*S2-(1/16)*T2*a^4-
(1/2)*T2*a^4*log(a)^2+S2*r3^2*log(r3)-(2/9)*L3*a^3-
(2/9)*r3^3*V2+(1/16)*T2*r3^4+(1/2)*a^2*S2+(1/18)*M2*a^6+(1/4)*T2*a^4*log(a)-(1/18)*M2*r3^6-
(1/4)*T2*r3^4*log(r3))+pi*(h2-h1)*G*k*((1/2)*T2*r2^4*log(r2)^2-(1/2)*C2*r3^2+(2/9)*A2*r2^6-
(1/4)*B2*r3^4-(2/9)*A2*r3^6+(1/2)*C2*r2^2-(1/2)*T2*r3^4*log(r3)^2-
(2/9)*L2*r3^3+(2/9)*L2*r2^3+(1/4)*B2*r2^4-
(2/3)*V2*r3^3*log(r3)+d2*r3^4*log(r3)^2+B2*r3^4*log(r3)+(4/3)*A2*r3^6*log(r3)+C2*r3^2*log(r3)-
(1/2)*N2*r3^4*log(r3)-(1/3)*M2*r3^6*log(r3)+(1/8)*N2*r3^4-(1/8)*N2*r2^4+(2/3)*L2*r3^3*log(r3)-
d2*r2^4*log(r2)^2-
(2/3)*L2*r2^3*log(r2)+(1/2)*N2*r2^4*log(r2)+S2*r2^2*log(r2)+(1/3)*M2*r2^6*log(r2)-
B2*r2^4*log(r2)-(4/3)*A2*r2^6*log(r2)-C2*r2^2*log(r2)+(2/3)*V2*r2^3*log(r2)+(1/2)*r3^2*S2-
S2*r3^2*log(r3)+(1/16)*T2*r2^4-(1/18)*M2*r2^6+(2/9)*r3^3*V2-(1/2)*r2^2*S2-(1/16)*T2*r3^4-
(2/9)*r2^3*V2+(1/18)*M2*r3^6+(1/4)*T2*r3^4*log(r3)-(1/4)*T2*r2^4*log(r2))+lambda5*a*log(a);
        (1/72)*pi*(h2-h1)^3*E*(12*T2*a^3*r2^2+12*nu*M2*r2^5*a^2-12*nu*M2*a^5*r2^2-
12*nu*N2*a^3*r2^2+12*nu*N2*r2^3*a^2-12*S2*a^2*r2+12*S2*a*r2^2+12*nu*S2*a^2*r2-12*nu*S2*a*r2^2-
12*N2*a^3*r2^2-4*M2*a^5*r2^2+12*N2*r2^3*a^2+4*M2*r2^5*a^2-
12*T2*r2^3*a^2+12*nu*T2*r2^3*log(r2)*a^2-12*V2*log(a)*a^2*r2^2+12*V2*log(r2)*a^2*r2^2-
12*T2*a^3*log(a)*r2^2+12*T2*r2^3*log(r2)*a^2-12*nu*T2*a^3*log(a)*r2^2)/((1-
nu^2)*a^2*r2^2)+pi*(h2-h1)*G*k*(-(4/3)*d3*r3^3*log(r3)+(4/3)*d3*a^3*log(a)-
(2/3)*T2*a^3*log(a)+V2*r3^2-(2/9)*T2*r3^3+(2/3)*N2*r3^3+2*S2*r3-L3*r3^2+(2/5)*M2*r3^5-
(2/9)*d3*r3^3-2*C3*r3+(8/5)*A3*a^5+(4/3)*B3*a^3-V2*a^2+(2/9)*T2*a^3-(2/3)*N2*a^3-2*S2*a+L3*a^2-
```

```
(2/5)*M2*a^5+(2/9)*d3*a^3+2*C3*a+(2/3)*T2*r3^3*log(r3)-(8/5)*A3*r3^5-(4/3)*B3*r3^3)+pi*(h2-
h1)*G*k*((4/3)*d2*r3^3*log(r3)+(4/3)*B2*r3^3+(2/9)*d2*r3^3+2*C2*r3+L2*r3^2+(8/5)*A2*r3^5-
(4/3)*B2*r2^3-(4/3)*d2*r2^3*log(r2)-(2/9)*d2*r2^3-2*C2*r2-
(8/5)*A2*r2^5+(2/3)*T2*r2^3*log(r2)+2*S2*r2+(2/3)*N2*r2^3-L2*r2^2+(2/5)*M2*r2^5-
(2/9)*T2*r2^3+V2*r2^2-V2*r3^2+(2/9)*T2*r3^3-(2/3)*N2*r3^3-2*S2*r3-(2/5)*M2*r3^5-
(2/3)*T2*r3^3*log(r3))+lambda5;
        (1/72)*pi*(h3-
h2)^3*E*(12*nu*V3*r3^5*a^2+12*nu*S3*r3^4*a^2+12*nu*T3*r3^6*log(r3)*a^2+12*nu*N3*r3^6*a^2-
12*nu*V3*a^5*r3^2-12*nu*S3*a^4*r3^2-12*nu*N3*a^6*r3^2+12*N3*r3^6*a^2-
12*S3*r3^4*a^2+6*T3*r3^6*a^2-12*nu*M3*a^8*r3^2+4*V3*r3^5*a^2+12*nu*M3*r3^8*a^2-
4*V3*a^5*r3^2+12*S3*a^4*r3^2-12*N3*a^6*r3^2-6*T3*a^6*r3^2+12*T3*r3^6*log(r3)*a^2-
12*nu*T3*a^6*log(a)*r3^2-12*T3*a^6*log(a)*r3^2+20*M3*r3^8*a^2-20*M3*a^8*r3^2)/((1-
nu^2)*r3^2*a^2)+pi*(h3-h2)*G*k*((1/4)*M3*r3^8+(2/5)*L3*a^5-
(1/3)*a^6*N3+(2/3)*B3*a^6+A3*a^8+(2/9)*d3*a^6+(1/18)*T3*a^6+(1/2)*C3*a^4-
(2/5)*L3*r3^5+(2/5)*r3^5*V3+(1/2)*r3^4*S3+(1/3)*r3^6*N3-(2/3)*B3*r3^6-A3*r3^8-
(1/3)*T3*a^6*log(a)+(2/3)*d3*a^6*log(a)-(1/2)*a^4*S3-(1/4)*M3*a^8-(2/5)*a^5*V3-(2/9)*d3*r3^6-
(1/18)*T3*r3^6-(1/2)*C3*r3^4-(2/3)*d3*r3^6*log(r3)+(1/3)*T3*r3^6*log(r3))+lambda6*a^3;
        (1/72)*pi*(h3-h2)^3*E*(12*nu*V3*r3^3*a^2+12*nu*T3*r3^4*log(r3)*a^2-
12*nu*T3*a^4*log(a)*r3^2+12*nu*M3*r3^6*a^2-12*nu*V3*a^3*r3^2-
12*nu*M3*a^6*r3^2+12*M3*r3^6*a^2+12*nu*N3*r3^4*a^2+12*V3*r3^3*a^2-12*V3*a^3*r3^2-12*M3*a^6*r3^2-
12*nu*N3*a^4*r3^2+12*T3*r3^4*log(r3)*a^2-12*T3*a^4*log(a)*r3^2+12*N3*r3^4*a^2-
12*N3*a^4*r3^2)/((1-nu^2)*r3^2*a^2)+pi*(h3-h2)*G*k*((1/2)*N3*r3^4-(2/3)*L3*r3^3+d3*a^4*log(a)-
(1/2)*T3*a^4*log(a)+(2/3)*r3^3*V3-(2/3)*a^3*V3+(2/3)*L3*a^3-
(1/3)*M3*a^6+B3*a^4+(4/3)*A3*a^6+(1/4)*d3*a^4+C3*a^2+(1/8)*T3*a^4+r3^2*S3+(1/3)*M3*r3^6-B3*r3^4-
a^2*S3-(1/2)*N3*a^4-(4/3)*A3*r3^6-(1/4)*d3*r3^4-C3*r3^2-(1/8)*T3*r3^4+(1/2)*T3*r3^4*log(r3)-
d3*r3^4*log(r3))+lambda6*a;
        (1/72)*pi*(h3-h2)^3*E*(12*V3*r3^2*a+12*nu*V3*r3*a^2-12*nu*V3*r3^2*a-12*V3*r3*a^2-
12*nu*S3*r3^2+12*nu*S3*a^2+12*nu*M3*r3^4*a^2-12*nu*T3*log(a)*r3^2*a^2-12*nu*M3*a^4*r3^2-
12*a^2*S3+12*r3^2*S3-12*M3*r3^4*a^2+12*M3*a^4*r3^2-
12*T3*log(r3)*r3^2*a^2+12*T3*log(a)*r3^2*a^2+12*nu*T3*log(r3)*r3^2*a^2)/((1-
nu^2)*r3^2*a^2)+pi*(h3-h2)*G*k*(2*S3*log(r3)+2*L3*a+2*B3*a^2-
2*V3*a+2*C3*log(a)+2*A3*a^4+(1/2)*T3*a^2+N3*r3^2+(1/2)*M3*r3^4-2*L3*r3-2*B3*r3^2+2*V3*r3-
2*C3*log(r3)-2*A3*r3^4-(1/2)*M3*a^4-N3*a^2-2*S3*log(a)-(1/2)*T3*r3^2+T3*r3^2*log(r3)-
2*d3*r3^2*log(r3)+2*d3*a^2*log(a)-T3*a^2*log(a))+lambda6/a;
        (1/72)*pi*(h3-h2)^3*E*(6*M3*r3^6*a^2+6*T3*r3^4*a^2+12*V3*r3^3*log(r3)*a^2-
12*nu*S3*log(a)*r3^2*a^2+12*nu*V3*r3^3*log(r3)*a^2+12*nu*M3*r3^6*log(r3)*a^2+12*nu*N3*r3^4*log(r3
)*a^2-12*nu*V3*a^3*log(a)*r3^2-12*nu*N3*a^4*log(a)*r3^2-
12*S3*log(r3)*r3^2*a^2+12*S3*log(a)*r3^2*a^2+12*M3*r3^6*log(r3)*a^2+12*N3*r3^4*log(r3)*a^2-
12*nu*M3*a^6*log(a)*r3^2+12*nu*T3*r3^4*log(r3)^2*a^2-12*V3*a^3*log(a)*r3^2-
12*M3*a^6*log(a)*r3^2+12*nu*S3*log(r3)*r3^2*a^2-12*N3*a^4*log(a)*r3^2-12*nu*T3*a^4*log(a)^2*r3^2-
6*T3*a^4*r3^2-12*V3*r3^3*a^2+12*V3*a^3*r3^2-6*M3*a^6*r3^2+12*T3*r3^4*log(r3)^2*a^2-
12*T3*a^4*log(a)^2*r3^2)/((1-nu^2)*r3^2*a^2)+pi*(h3-
h2)*G*k*((1/2)*a^2*S3+(1/4)*B3*r3^4+(1/2)*C3*r3^2-(1/2)*r3^2*S3-(1/2)*T3*a^4*log(a)^2-
(1/8)*N3*r3^4+(1/8)*N3*a^4+(1/2)*T3*r3^4*log(r3)^2-(2/9)*A3*a^6-S3*a^2*log(a)-(1/4)*B3*a^4-
(1/16)*T3*a^4-(2/9)*r3^3*V3-(2/9)*L3*a^3+(1/4)*T3*a^4*log(a)-
(1/4)*T3*r3^4*log(r3)+(1/16)*T3*r3^4+(2/9)*A3*r3^6-(1/2)*C3*a^2-
(1/2)*N3*a^4*log(a)+(2/9)*L3*r3^3+(2/3)*L3*a^3*log(a)-d3*r3^4*log(r3)^2-
(1/3)*M3*a^6*log(a)+B3*a^4*log(a)-(4/3)*A3*r3^6*log(r3)+(1/3)*M3*r3^6*log(r3)-
B3*r3^4*log(r3)+(2/3)*V3*r3^3*log(r3)+S3*r3^2*log(r3)-C3*r3^2*log(r3)+d3*a^4*log(a)^2-
(2/3)*V3*a^3*log(a)-
(2/3)*L3*r3^3*log(r3)+(1/2)*N3*r3^4*log(r3)+C3*a^2*log(a)+(4/3)*A3*a^6*log(a)+(1/18)*M3*a^6-
(1/18)*M3*r3^6+(2/9)*a^3*V3)+lambda6*a*log(a);
        (1/72)*pi*(h3-h2)^3*E*(12*S3*r3^2*a+12*nu*S3*r3*a^2-12*nu*S3*r3^2*a-
12*S3*r3*a^2+12*nu*M3*r3^5*a^2+12*nu*N3*r3^3*a^2+12*T3*r3^3*log(r3)*a^2+12*nu*T3*r3^3*log(r3)*a^2
-12*nu*T3*a^3*log(a)*r3^2-12*nu*N3*a^3*r3^2-12*nu*M3*a^5*r3^2-
12*V3*log(a)*r3^2*a^2+12*V3*log(r3)*r3^2*a^2+4*M3*r3^5*a^2+12*N3*r3^3*a^2-12*T3*r3^3*a^2-
12*N3*a^3*r3^2+12*T3*a^3*r3^2-4*M3*a^5*r3^2-12*T3*a^3*log(a)*r3^2)/((1-nu^2)*r3^2*a^2)+pi*(h3-
h2)*G*k*(V3*r3^2+(2/3)*N3*r3^3-(2/3)*N3*a^3+L3*a^2-
2*S3*a+(2/9)*d3*a^3+(4/3)*B3*a^3+2*C3*a+(8/5)*A3*a^5+(2/9)*T3*a^3+(2/5)*M3*r3^5-L3*r3^2+2*S3*r3-
V3*a^2-(2/5)*M3*a^5-(2/9)*d3*r3^3-(4/3)*B3*r3^3-2*C3*r3-(8/5)*A3*r3^5-(2/9)*T3*r3^3-
(4/3)*d3*r3^3*log(r3)+(2/3)*T3*r3^3*log(r3)-(2/3)*T3*a^3*log(a)+(4/3)*d3*a^3*log(a))+lambda6;
        A1*r2^4+B1*r2^2+C1*log(r2)+d1*r2^2*log(r2)+L1*r2+H1-A2*r2^4-B2*r2^2-C2*log(r2)-
d2*r2^2*log(r2)-L2*r2-H2;
        A2*r3^4+B2*r3^2+C2*log(r3)+d2*r3^2*log(r3)+L2*r3+H2-A3*r3^4-B3*r3^2-C3*log(r3)-
d3*r3^2*log(r3)-L3*r3-H3;
        A3*a^4+B3*a^2+C3*log(a)+d3*a^2*log(a)+L3*a+H3;
        M1*a^3+N1*a+S1/a+T1*a*log(a)+V1;
        M2*a^3+N2*a+S2/a+T2*a*log(a)+V2;
        M3*a^3+N3*a+S3/a+T3*a*log(a)+V3];
```

```
end
```

Appendix G. The GUI Code for the Quarter-car Suspension Model

```
function varargout = GUI_QuarterCarRig_04Nov10(varargin)
% GUI_QUARTERCARRIG_04NOV10 M-file for
% GUI_QuarterCarRig_04Nov10.fig
%      GUI_QUARTERCARRIG_04NOV10, by itself, creates a new GUI_QUARTERCARRIG_04NOV10 or raises
the existing
%      singleton*.
%
%      H = GUI_QUARTERCARRIG_04NOV10 returns the handle to a new GUI_QUARTERCARRIG_04NOV10 or the
handle to
%      the existing singleton*.
%
%      GUI_QUARTERCARRIG_04NOV10('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in GUI_QUARTERCARRIG_04NOV10.M with the given input arguments.
%
%      GUI_QUARTERCARRIG_04NOV10('Property','Value',...) creates a new GUI_QUARTERCARRIG_04NOV10
or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before GUI_hydraulic_damper_04June09_OpeningFunction gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to GUI_QuarterCarRig_04Nov10_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help GUI_QuarterCarRig_04Nov10
% Last Modified by GUIDE v2.5 07-Nov-2010 12:02:57
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @GUI_QuarterCarRig_04Nov10_OpeningFcn, ...
                   'gui_OutputFcn',  @GUI_QuarterCarRig_04Nov10_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT
% --- Executes just before GUI_QuarterCarRig_04Nov10 is made visible.
function GUI_QuarterCarRig_04Nov10_OpeningFcn(hObject, ~, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to GUI_QuarterCarRig_04Nov10 (see VARARGIN)
% Choose default command line output for GUI_QuarterCarRig_04Nov10
handles.output = hObject;
% Update handles structure
guidata(hObject, handles);
% UIWAIT makes GUI_QuarterCarRig_04Nov10 wait for user response (see UIRESUME)
% uiwait(handles.figure1);
% --- Outputs from this function are returned to the command line.
function varargout = GUI_QuarterCarRig_04Nov10_OutputFcn(~, ~, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
% Get default command line output from handles structure
varargout{1} = handles.output;
function Sprungmass_Callback(hObject, ~, ~)
% hObject    handle to Sprungmass (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of Sprungmass as text
%        str2double(get(hObject,'String')) returns contents of Sprungmass as a double
global Mb
Mb = str2double(get(hObject,'String'))*0.0310809502/12;  %converting to slugs
% --- Executes during object creation, after setting all properties.
function Sprungmass_CreateFcn(hObject, ~, ~)
% hObject    handle to Sprungmass (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
global Mb
Mb = str2double(get(hObject,'String'))*0.0310809502/12;
function Unsprungmass_Callback(hObject, ~, ~)
% hObject    handle to Unsprungmass (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of Unsprungmass as text
%        str2double(get(hObject,'String')) returns contents of Unsprungmass as a double
global Mt
Mt = str2double(get(hObject,'String'))*0.0310809502/12;
% --- Executes during object creation, after setting all properties.
function Unsprungmass_CreateFcn(hObject, ~, ~)
% hObject    handle to Unsprungmass (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
global Mt
Mt = str2double(get(hObject,'String'))*0.0310809502/12;
function Tirestiffness_Callback(hObject, ~, ~)
% hObject    handle to Tirestiffness (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of Tirestiffness as text
%        str2double(get(hObject,'String')) returns contents of Tirestiffness as a double
global Kt
Kt = str2double(get(hObject,'String'));
% --- Executes during object creation, after setting all properties.
function Tirestiffness_CreateFcn(hObject, ~, ~)
% hObject    handle to Tirestiffness (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
global Kt
Kt = str2double(get(hObject,'String'));
function SpringStiffness_Callback(hObject, ~, ~)
% hObject    handle to SpringStiffness (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of SpringStiffness as text
%        str2double(get(hObject,'String')) returns contents of SpringStiffness as a double
global Ks
```

```
Ks = str2double(get(hObject,'String'));
% --- Executes during object creation, after setting all properties.
function SpringStiffness_CreateFcn(hObject, ~, ~)
% hObject    handle to SpringStiffness (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
global Ks
Ks = str2double(get(hObject,'String'));
function p30_Callback(hObject, ~, ~)
% hObject    handle to p30 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of p30 as text
%        str2double(get(hObject,'String')) returns contents of p30 as a double
global p30
p30 = str2double(get(hObject,'String'));
%save(filename,'p30','-append');
% --- Executes during object creation, after setting all properties.
function p30_CreateFcn(hObject, ~, ~)
% hObject    handle to p30 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
global p30
p30 = str2double(get(hObject,'String'));
%save(filename,'p30','-append');
function Rp_Callback(hObject, ~, ~)
% hObject    handle to Rp (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of Rp as text
%        str2double(get(hObject,'String')) returns contents of Rp as a double
global Rp
Rp = str2double(get(hObject,'String'));
%save(filename,'Rp','-append');
% --- Executes during object creation, after setting all properties.
function Rp_CreateFcn(hObject, ~, ~)
% hObject    handle to Rp (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
global Rp
Rp = str2double(get(hObject,'String'));
%save(filename,'Rp','-append');
function dr_Callback(hObject, ~, ~)
% hObject    handle to dr (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of dr as text
%        str2double(get(hObject,'String')) returns contents of dr as a double
global dr
dr = str2double(get(hObject,'String'));
%save(filename,'dr','-append');
% --- Executes during object creation, after setting all properties.
function dr_CreateFcn(hObject, ~, ~)
% hObject    handle to dr (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
global dr
dr = str2double(get(hObject,'String'));
%save(filename,'dr','-append');
function s10_Callback(hObject, ~, ~)
% hObject    handle to s10 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of s10 as text
%        str2double(get(hObject,'String')) returns contents of s10 as a double
global s10
s10 = str2double(get(hObject,'String'));
%save(filename,'s10','-append');
% --- Executes during object creation, after setting all properties.
function s10_CreateFcn(hObject, ~, ~)
% hObject    handle to s10 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
global s10
s10 = str2double(get(hObject,'String'));
%save(filename,'s10','-append');
function s20_Callback(hObject, ~, ~)
% hObject    handle to s20 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of s20 as text
%        str2double(get(hObject,'String')) returns contents of s20 as a double
global s20
s20 = str2double(get(hObject,'String'));
%save(filename,'s20','-append');
% --- Executes during object creation, after setting all properties.
function s20_CreateFcn(hObject, ~, ~)
% hObject    handle to s20 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
global s20
s20 = str2double(get(hObject,'String'));
%save(filename,'s20','-append');
function s30_Callback(hObject, ~, ~)
% hObject    handle to s30 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of s30 as text
%        str2double(get(hObject,'String')) returns contents of s30 as a double
global s30
s30 = str2double(get(hObject,'String'));
%save(filename,'s30','-append');
% --- Executes during object creation, after setting all properties.
function s30_CreateFcn(hObject, ~, ~)
% hObject    handle to s30 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
```

```
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
global s30
s30 = str2double(get(hObject,'String'));
%save(filename,'s30','-append');
function m2_Callback(hObject, ~, ~)
% hObject    handle to m2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of m2 as text
%        str2double(get(hObject,'String')) returns contents of m2 as a double
global m2
m2 = str2double(get(hObject,'String'));
%save(filename,'m2','-append');
% --- Executes during object creation, after setting all properties.
function m2_CreateFcn(hObject, ~, ~)
% hObject    handle to m2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
global m2
m2 = str2double(get(hObject,'String'));
%save(filename,'m2','-append');
function Ff1_Callback(hObject, ~, ~)
% hObject    handle to Ff1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of Ff1 as text
%        str2double(get(hObject,'String')) returns contents of Ff1 as a double
global Ff1
Ff1 = str2double(get(hObject,'String'));
%save(filename,'Ff1','-append');
% --- Executes during object creation, after setting all properties.
function Ff1_CreateFcn(hObject, ~, ~)
% hObject    handle to Ff1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
global Ff1
Ff1 = str2double(get(hObject,'String'));
%save(filename,'Ff1','-append');
function Ff2_Callback(hObject, ~, ~)
% hObject    handle to Ff2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of Ff2 as text
%        str2double(get(hObject,'String')) returns contents of Ff2 as a double
global Ff2
Ff2 = str2double(get(hObject,'String'));
%save(filename,'Ff2','-append');
% --- Executes during object creation, after setting all properties.
function Ff2_CreateFcn(hObject, ~, ~)
% hObject    handle to Ff2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
```

```
end
global Ff2
Ff2 = str2double(get(hObject,'String'));
%save(filename,'Ff2','-append');
function Cd3_Callback(hObject, ~, ~)
% hObject    handle to Cd3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of Cd3 as text
%        str2double(get(hObject,'String')) returns contents of Cd3 as a double
global Cd3
Cd3 = str2double(get(hObject,'String'));
%save(filename,'Cd3','-append');
% --- Executes during object creation, after setting all properties.
function Cd3_CreateFcn(hObject, ~, ~)
% hObject    handle to Cd3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
global Cd3
Cd3 = str2double(get(hObject,'String'));
%save(filename,'Cd3','-append');
function n_Callback(hObject, ~, ~)
% hObject    handle to n (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of n as text
%        str2double(get(hObject,'String')) returns contents of n as a double
global n
n = str2double(get(hObject,'String'));
%save(filename,'n','-append');
% --- Executes during object creation, after setting all properties.
function n_CreateFcn(hObject, ~, ~)
% hObject    handle to n (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
global n
n = str2double(get(hObject,'String'));
%save(filename,'n','-append');
function beta_Callback(hObject, ~, ~)
% hObject    handle to beta (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of beta as text
%        str2double(get(hObject,'String')) returns contents of beta as a double
global beta
beta = str2double(get(hObject,'String'));
%save(filename,'beta','-append');
% --- Executes during object creation, after setting all properties.
function beta_CreateFcn(hObject, ~, ~)
% hObject    handle to beta (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
global beta
beta = str2double(get(hObject,'String'));
```

```
%save(filename,'beta','-append');
function rho_Callback(hObject, ~, ~)
% hObject    handle to rho (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of rho as text
%        str2double(get(hObject,'String')) returns contents of rho as a double
global rho
rho = str2double(get(hObject,'String'));
%save(filename,'rho','-append');
% --- Executes during object creation, after setting all properties.
function rho_CreateFcn(hObject, ~, ~)
% hObject    handle to rho (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
global rho
rho = str2double(get(hObject,'String'));
%save(filename,'rho','-append');
function E_Callback(hObject, ~, ~)
% hObject    handle to E (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of E as text
%        str2double(get(hObject,'String')) returns contents of E as a double
global E
E = str2double(get(hObject,'String'));
%save(filename,'E','-append');
% --- Executes during object creation, after setting all properties.
function E_CreateFcn(hObject, ~, ~)
% hObject    handle to E (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
global E
E = str2double(get(hObject,'String'));
%save(filename,'E','-append');
function nu_Callback(hObject, ~, ~)
% hObject    handle to nu (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of nu as text
%        str2double(get(hObject,'String')) returns contents of nu as a double
global nu
nu = str2double(get(hObject,'String'));
%save(filename,'nu','-append');
% --- Executes during object creation, after setting all properties.
function nu_CreateFcn(hObject, ~, ~)
% hObject    handle to nu (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
global nu
nu = str2double(get(hObject,'String'));
%save(filename,'nu','-append');
function a_Callback(hObject, ~, ~)
% hObject    handle to a (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of a as text
%        str2double(get(hObject,'String')) returns contents of a as a double
global a
a = str2double(get(hObject,'String'));
%save(filename,'a','-append');
% --- Executes during object creation, after setting all properties.
function a_CreateFcn(hObject, ~, ~)
% hObject    handle to a (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
global a
a = str2double(get(hObject,'String'));
%save(filename,'a','-append');
% function b_Callback(hObject, ~, ~)
% hObject    handle to b (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
%
% Hints: get(hObject,'String') returns contents of b as text
%        str2double(get(hObject,'String')) returns contents of b as a double
% global b
% b = str2double(get(hObject,'String'));
%
% --- Executes during object creation, after setting all properties.
% function b_CreateFcn(hObject, ~, ~)
% hObject    handle to b (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
% if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
%     set(hObject,'BackgroundColor','white');
% end
% global b
% b = str2double(get(hObject,'String'));
function th_Callback(hObject, ~, ~)
% hObject    handle to th (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of th as text
%        str2double(get(hObject,'String')) returns contents of th as a double
global th
th = str2double(get(hObject,'String'));
%save(filename,'th','-append');
% --- Executes during object creation, after setting all properties.
function th_CreateFcn(hObject, ~, ~)
% hObject    handle to th (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
global th
th = str2double(get(hObject,'String'));
%save(filename,'th','-append');
function Cd2_Callback(hObject, ~, ~)
% hObject    handle to Cd2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of Cd2 as text
```

```
%        str2double(get(hObject,'String')) returns contents of Cd2 as a double
global Cd2
Cd2 = str2double(get(hObject,'String'));
%save(filename,'Cd2','-append');
% --- Executes during object creation, after setting all properties.
function Cd2_CreateFcn(hObject, ~, ~)
% hObject    handle to Cd2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
global Cd2
Cd2 = str2double(get(hObject,'String'));
%save(filename,'Cd2','-append');
function A4_Callback(hObject, ~, ~)
% hObject    handle to A4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of A4 as text
%        str2double(get(hObject,'String')) returns contents of A4 as a double
global A4
A4 = str2double(get(hObject,'String'));
%save(filename,'A4','-append');
% --- Executes during object creation, after setting all properties.
function A4_CreateFcn(hObject, ~, ~)
% hObject    handle to A4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
global A4
A4 = str2double(get(hObject,'String'));
%save(filename,'A4','-append');
function Cd4_Callback(hObject, ~, ~)
% hObject    handle to Cd4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of Cd4 as text
%        str2double(get(hObject,'String')) returns contents of Cd4 as a double
global Cd4
Cd4 = str2double(get(hObject,'String'));
%save(filename,'Cd4','-append');
% --- Executes during object creation, after setting all properties.
function Cd4_CreateFcn(hObject, ~, ~)
% hObject    handle to Cd4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
global Cd4
Cd4 = str2double(get(hObject,'String'));
%save(filename,'Cd4','-append');
function freq_Callback(hObject, ~, ~)
% hObject    handle to freq (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of freq as text
%        str2double(get(hObject,'String')) returns contents of freq as a double
global freq
freq = str2double(get(hObject,'String'));
```

```
%save(filename,'freq','-append');
% --- Executes during object creation, after setting all properties.
function freq_CreateFcn(hObject, ~, ~)
% hObject    handle to freq (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
global freq
freq = str2double(get(hObject,'String'));
%save(filename,'freq','-append');
function amp_Callback(hObject, ~, ~)
% hObject    handle to amp (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of amp as text
%        str2double(get(hObject,'String')) returns contents of amp as a double
global amp
amp = str2double(get(hObject,'String'));
%save(filename,'amp','-append');
% --- Executes during object creation, after setting all properties.
function amp_CreateFcn(hObject, ~, ~)
% hObject    handle to amp (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
global amp
amp = str2double(get(hObject,'String'));
%save(filename,'amp','-append');
function tspan_Callback(hObject, ~, ~)
% hObject    handle to tspan (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of tspan as text
%        str2double(get(hObject,'String')) returns contents of tspan as a double
global tspan tincr
tspan = str2double(get(hObject,'String'));
tspan=0:tincr:tspan; %ode solver returns solutions only at the times in the vector tspan
%save(filename,'tspan','-append');
% --- Executes during object creation, after setting all properties.
function tspan_CreateFcn(hObject, ~, ~)
% hObject    handle to tspan (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
global tspan tincr
tspan = str2double(get(hObject,'String'));
tspan=0:tincr:tspan; %ode solver returns solutions only at the times in the vector tspan
%save(filename,'tspan','-append');
% --- Executes on button press in solve.
function solve_Callback(~, ~, ~) %#ok<*DEFNU>
% hObject    handle to solve (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global dr Ar Rp Ap p30 s10 s20 s30 V10 V20 V30 Xr Xrdot tspan P T Pvap cavhandle def_counter
disk_def tTime options wait filename deflection_case h h1 h3 R1 R2 a r1 slopeA slopeB p0 Xb0 Vb0
Xt0 Vt0 Mb Mt Kt Ks freq TP contact tincr Ttemp Ptemp odechoice
tic
```

```
Ar=pi/4*dr^2;
Ap=pi*Rp^2;
V10=(Ap-Ar)*s10;
V20=Ap*s20;
V30=Ap*s30;
options = optimset('Algorithm','levenberg-marquardt','MaxFunEvals',50000,'TolFun',1e-
15,'TolX',1e-15,'DiffMaxChange',1e-1,'MaxIter',10000,'Display','off'); %options for fsolve to
solve the deflection of the shim (or disk)
def_counter=1;  % a counter to increase the length of the length of the deflection matrix
maxtimestep=1e-2;
tTime=zeros(1,floor(max(tspan)/maxtimestep));
disk_def=zeros(1,floor(max(tspan)/maxtimestep));
% Xr=inline('amp*(sin(pi*freq*T)).^2');     %HaverSine input
% Xrdot=inline('pi*freq*amp*sin(2*pi*freq*T)');
Xr=inline('amp*sin(2*pi*freq*T)');   %Sine input
Xrdot=inline('2*amp*pi*freq*cos(2*pi*freq*T)');
contact=1;
TP=(Mb+Mt)*386.4/Kt;
I_p0=[p30 p30 Xb0 Vb0 Xt0 Vt0]; %intial conditions for the damper+quarter car rig ODE
wait = waitbar(0,'Running code, Please wait...');
options_ode=odeset('InitialStep',5e-9,'AbsTol',1e-6,'RelTol',1e-3,'MaxStep',maxtimestep);
%initial time step for ODE solver and other solver parameters
if deflection_case==2 %case A
    %To save time, we don't case1 (single disk) to find the maximum
    %pressure difference across the shim
%     deflection_case=1; %to solve the single disk problem first
    h=h3+h1;
    R1=a;
    R2=r1;
%     [TT,PP]=ode23t(@pressure_Englishunit_14June10,tspan/2,I_p0,options_ode);
%     p0_max=max(abs(PP(:,1)-PP(:,2)));
     clear R1 R2 h PP TT
%    p0=p0_max;
%    deflection_case=2; %role back to case A
    p0=1000;
    p0_max=p0;
    [ABC_A]=fsolve(@caseA_30sep09_final_for_hyd_mod,zeros(1,39),options);
    A1_A=ABC_A(1);
    B1_A=ABC_A(2);
    C1_A=ABC_A(3);
    d1_A=ABC_A(4);
    L1_A=ABC_A(5);
    H1_A=ABC_A(6);
    w_tip=A1_A*r1^4+B1_A*r1^2+C1_A*log(r1)+d1_A*r1^2*log(r1)+L1_A*r1+H1_A;
    slopeA=w_tip/p0_max;
    clear w_tip p0_max p0
    def_counter=1;
%     tTime=zeros(1,1000);
%     disk_def=zeros(1,1000);
end
if deflection_case==3 %case B
%     deflection_case=1; %to solve the single disk problem first
    h=h3+h1;
    R1=a;
    R2=r1;
%     [TT,PP]=ode23t(@pressure_Englishunit_14June10,tspan/2,I_p0,options_ode);
%     p0_max=max(abs(PP(:,1)-PP(:,2)));
     clear R1 R2 h PP TT
%     p0=p0_max;
%     deflection_case=3;
    p0=1000;
    p0_max=p0;
    [ABC_B]=fsolve(@caseB_30sep09_final_for_hyd_mod,zeros(1,39),options);
    A1_B=ABC_B(1);
    B1_B=ABC_B(2);
    C1_B=ABC_B(3);
    d1_B=ABC_B(4);
    L1_B=ABC_B(5);
    H1_B=ABC_B(6);
```

```
    w_tip=A1_B*r1^4+B1_B*r1^2+C1_B*log(r1)+d1_B*r1^2*log(r1)+L1_B*r1+H1_B;
    slopeB=w_tip/p0_max;
    clear w_tip p0_max p0
    def_counter=1;
%     tTime=zeros(1,1000);
%     disk_def=zeros(1,1000);
end
P=zeros(max(tspan)/tincr,6);
T=zeros(max(tspan)/tincr,1);
%NOW solve the main ode
[T,P]=eval([odechoice '(@pressure_Englishunit_04Nov10,tspan,I_p0,options_ode)']);
%[T,P]=ode23t(@pressure_Englishunit_04Nov10,tspan,I_p0,options_ode);
close(wait)
% save the variables in a file for later use
%save(filename,'-append')
warndlg(['elapsed time was ' num2str(toc)],'time spent')
% --- Executes on button press in plot.
function plot_Callback(~, ~, ~)
% hObject    handle to plot (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global Ar Ap Ff1 Ff2 amp freq X1 X1dot P T Fd filename Fp
%plot(T,P(:,7))
%xx1=-X1(T,amp,freq);
%xx1dot=-X1dot(T,amp,freq);
clear Fd Fp
Fd(:,1)=P(:,1)*(Ap-Ar)-P(:,2)*Ap; %force exerted on damper body
%axes(handles.axes1)
figure(19)
subplot(2,1,1)
hold all
plot(-(P(:,5)-P(:,3)),Fd(:,1)),grid on,xlabel('stroke in'),ylabel('Damper force lb')
%axes(handles.axes2)
subplot(2,1,2)
hold all
plot(-(P(:,6)-P(:,4)),Fd(:,1)),grid on,xlabel('Velocity in/s'),ylabel('damper force lb')
%save(filename,'-append')
% --- Executes on button press in damp_spec.
function damp_spec_Callback(~, ~, ~)
% hObject    handle to damp_spec (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
close(figure(1))
figure('Name','Damper specs.','NumberTitle','on','Position',[500 100 666 640])
set(gca,'DataAspectRatio',[1 1 1])
pic=imread('C:\damperfigforGUI','jpg');
image(pic)
function A3_Callback(hObject, ~, ~)
% hObject    handle to A3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of A3 as text
%        str2double(get(hObject,'String')) returns contents of A3 as a double
global A3
A3 = str2double(get(hObject,'String'));
%save(filename,'A3','-append')
% --- Executes during object creation, after setting all properties.
function A3_CreateFcn(hObject, ~, ~)
% hObject    handle to A3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
global A3
A3 = str2double(get(hObject,'String'));
```

```
%save(filename,'A3','-append')
function Pvap_Callback(hObject, ~, ~)
% hObject    handle to Pvap (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of Pvap as text
%        str2double(get(hObject,'String')) returns contents of Pvap as a double
global Pvap
Pvap = str2double(get(hObject,'String'));
%save(filename,'Pvap','-append')
% --- Executes during object creation, after setting all properties.
function Pvap_CreateFcn(hObject, ~, ~)
% hObject    handle to Pvap (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
global Pvap
Pvap = str2double(get(hObject,'String'));
%save(filename,'Pvap','-append')
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
% --- Executes on button press in cavitation.
function cavitation_Callback(hObject, ~, ~)
% hObject    handle to cavitation (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hint: get(hObject,'Value') returns toggle state of cavitation
button_state = get(hObject,'Value');
global cavhandle
if button_state == get(hObject,'Max')
    % Toggle button is pressed-take approperiate action
   cavhandle=0;
elseif button_state == get(hObject,'Min')
    % Toggle button is not pressed-take appropriate action
   cavhandle=1;
end
% --- Executes on button press in clear_plot.
function clear_plot_Callback(~, ~, handles)
% hObject    handle to clear_plot (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
cla(handles.axes1)
cla(handles.axes2)
% --- Executes on button press in Sprung_mass_time_hist.
function Sprung_mass_time_hist_Callback(~, ~, ~)
% hObject    handle to Sprung_mass_time_hist (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global Ar Ap Ff1 Ff2 amp freq X1dot P T Fd
% xx1=-X1(T,amp,freq);
% xx1dot=-X1dot(T,amp,freq);
clear Fd
Fd(:,1)=P(:,1)*(Ap-Ar)-P(:,2)*Ap; %force exerted on damper body
figure(132)
subplot(2,1,1)
hold all
plot(T,P(:,3)),grid on, xlabel('time s'),ylabel('Sprung Mass Position, in')
subplot(2,1,2)
hold all
plot(T,P(:,4)),grid on, xlabel('time s'),ylabel('Sprung Mass Velocity, in/sec')
% --- Executes on button press in pushbutton8.
function pushbutton8_Callback(~, ~, ~)
% hObject    handle to pushbutton8 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
close(figure(2))
figure('Name','Disk specs.','NumberTitle','on','Position',[500 250 666 400])
```

```
set(gca,'DataAspectRatio',[1 1 1])
pic2=imread('C:\diskfigforGUI','jpg');
image(pic2)
% --- Executes on button press in Unsprung_mass_time_hist.
function Unsprung_mass_time_hist_Callback(~, ~, ~)
% hObject    handle to Unsprung_mass_time_hist (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global Ar Ap Ff1 Ff2 amp freq X1 X1dot P T Fd
% xx1=-X1(T,amp,freq);
% xx1dot=-X1dot(T,amp,freq);
clear Fd
Fd(:,1)=P(:,1)*(Ap-Ar)-P(:,2)*Ap; %force exerted on damper body
%close(figure(3))
figure (90)
subplot(2,1,1)
hold all
plot(T,P(:,5)),grid on,xlabel('Time, sec'),ylabel('Un-sprung Mass Position, in')
subplot(2,1,2)
hold all
plot(T,P(:,6)),grid on,xlabel('Time, sec'),ylabel('Un-sprung Mass Velocity, in/sec')
% --- Executes on button press in import_ex_data.
function import_ex_data_Callback(~, ~, ~)
% hObject    handle to import_ex_data (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global exp_x exp_Force exp_Temp exp_v
path = 'Z:\ORGANIZED DOCUMENTS\HYDRAULID DAMPER';
%path for file window
nfiles = 1;           %number of files to load
first_row = 52;       %first row of data in CSV file
fsamp = 2000;          %sample rate, Hz
figs_together = 1;    %plots figs in one subplot, or in seperate plots
series_together = 1;  %plots data series in same plot
cd(path);
fig=50;
colors = 'bgrcmkbgrcmkbgrcmkbgrcmk';
for i=1:nfiles
    [file, path] = uigetfile('*.asc',['Select CSV Shock Data File #',num2str(i)]);
    cd(path);
    dd = csvread(file,first_row-1,0);
    exp_x    = dd(:,1);
    exp_Force = dd(:,2);
    exp_Temp = dd(:,3);
    exp_v     = dd(:,7);
    N = length(exp_x);
    exp_t = (0:(N-1)) * 1/fsamp;
    %shockdata = struct('x',exp_x,'v',exp_v,'F',exp_Force,'T',exp_Temp,'time',exp_t);
    %eval(['save ', filename(1:length(filename)-4), ' shockdata'])
    if figs_together==1
        if series_together == 1
            figure(fig+1),
            subplot(231),plot(exp_t,exp_x,colors(i)), xlabel('Time, s'), ylabel('Position,
in'),hold on,grid on
            subplot(232),plot(exp_t,exp_v,colors(i)), xlabel('Time, s'), ylabel('Velocity,
in/s'),hold on,grid on
            subplot(233),plot(exp_t,exp_Force,colors(i)), xlabel('Time, s'), ylabel('Force,
lbs'),hold on,grid on
            subplot(234),plot(exp_t,exp_Temp,colors(i)), xlabel('Time, s'), ylabel('Temperature,
^oF'),hold on,grid on
            subplot(235),plot(exp_v,exp_Force,colors(i)), xlabel('Velocity, in/s'),
ylabel('Force, lbs'),hold on,grid on
            subplot(236),plot(exp_x,exp_Force,colors(i)), xlabel('Position, in'), ylabel('Force,
lbs'),hold on,grid on
        else
            figure(fig+1),
            subplot(231),plot(exp_t,exp_x), xlabel('Time, s'), ylabel('Position, in'),grid on
            subplot(232),plot(exp_t,exp_v), xlabel('Time, s'), ylabel('Velocity, in/s'),grid on
            subplot(233),plot(exp_t,exp_Force), xlabel('Time, s'), ylabel('Force, lbs'),grid on
```

```
            subplot(234),plot(exp_t,exp_Temp), xlabel('Time, s'), ylabel('Temperature, ^oF'),grid
on
            subplot(235),plot(exp_v,exp_Force), xlabel('Velocity, in/s'), ylabel('Force,
lbs'),grid on
            subplot(236),plot(exp_x,exp_Force), xlabel('Position, in'), ylabel('Force, lbs'),grid
on
            fig=fig+1;
        end
    else

        if series_together == 1
            figure(fig+1),plot(exp_t,exp_x,colors(i)), xlabel('Time, s'), ylabel('Position, in'),
hold on,grid on
            figure(fig+2),plot(exp_t,exp_v,colors(i)), xlabel('Time, s'), ylabel('Velocity,
in/s'), hold on,grid on
            figure(fig+3),plot(exp_t,exp_Force,colors(i)), xlabel('Time, s'), ylabel('Force,
lbs'), hold on,grid on
            figure(fig+4),plot(exp_t,exp_Temp,colors(i)), xlabel('Time, s'), ylabel('Temperature,
^oF'), hold on,grid on
            figure(fig+5),plot(exp_v,exp_Force,colors(i)), xlabel('Velocity, in/s'),
ylabel('Force, lbs'), hold on,grid on
            figure(fig+6),plot(exp_x,exp_Force,colors(i)), xlabel('Position, in'), ylabel('Force,
lbs'), hold on,grid on
        else
            figure(fig+1),plot(exp_t,exp_x), xlabel('Time, s'), ylabel('Position, in'),grid on
            figure(fig+2),plot(exp_t,exp_v), xlabel('Time, s'), ylabel('Velocity, in/s'),grid on
            figure(fig+3),plot(exp_t,exp_Force), xlabel('Time, s'), ylabel('Force, lbs'),grid on
            figure(fig+4),plot(exp_t,exp_Temp), xlabel('Time, s'), ylabel('Temperature,
^oF'),grid on
            figure(fig+5),plot(exp_v,exp_Force), xlabel('Velocity, in/s'), ylabel('Force,
lbs'),grid on
            figure(fig+6),plot(exp_x,exp_Force), xlabel('Position, in'), ylabel('Force,
lbs'),grid on
            fig=fig+6;
        end

    end
end
%save(filename,'-append')
% --- Executes on button press in Overlaid_sim_ex.
function Overlaid_sim_ex_Callback(~, ~, ~)
% hObject    handle to Overlaid_sim_ex (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global Ar Ap Ff1 Ff2 amp freq X1 X1dot P T Fd exp_x exp_Force  exp_v
xx1=-X1(T,amp,freq);
xx1dot=-X1dot(T,amp,freq);
clear Fd
Fd(:,1)=P(:,1)*(Ap-Ar)-P(:,3)*Ap-sign(X1dot(T,amp,freq))*Ff1-sign(P(:,4))*Ff2;
%close(figure(3))
figure (204)
subplot(2,1,1)
hold all
plot(xx1(:)+amp/2,Fd(:)),grid on,xlabel('Stroke in'),ylabel('Damper force lb')
plot(exp_x(0.1*length(exp_x):0.8*length(exp_x)),exp_Force(0.1*length(exp_x):0.8*length(exp_x)))
%test results plot (cutting the transients at the begining and end of test)
subplot(2,1,2)
hold all
plot(xx1dot(:),Fd(:)),grid on,xlabel('Velocity in/s'),ylabel('Damper force lb')
plot(exp_v(0.1*length(exp_x):0.8*length(exp_x)),exp_Force(0.1*length(exp_x):0.8*length(exp_x)))
%test results plot
% --- Executes on button press in tip_disk_def.
function tip_disk_def_Callback(~, ~, ~)
% hObject    handle to tip_disk_def (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global disk_def tTime  pre_def X1 amp freq X1dot P T P1_interp P2_interp
%plots the disk deflection that constitutes the area for the oil flow.
%(tip_def - pre_def)
```

```
for i=1:length(tTime)
    if disk_def(i)>=pre_def
        disk_def(i)=disk_def(i)-pre_def;
    else
        disk_def(i)=0;
    end
end
P1_interp=interp1(T,P(:,1),tTime);  %Interpolating P1
P2_interp=interp1(T,P(:,2),tTime);
for i=1:length(tTime)
    if P1_interp(i)>P2_interp(i)
        disk_def(i)=-disk_def(i); %in rebound (P1>P2) define the disk deflection to be negative
    end
end
figure(110)
subplot(2,1,1)
hold all
plot(tTime,disk_def),grid on,xlabel('Time sec'),ylabel('Disk tip deflection in')
subplot(2,1,2)
hold all
plot(T,P(:,2)-P(:,1)),grid on,xlabel('Time sec'),ylabel('pressure difference across the shim
stack (p2-p1)')
% figure(111)
% hold all
% plot(-X1(tTime,amp,freq)+amp/2,disk_def,'.'),grid on, xlabel('Displacement,in'),ylabel('Disk
tip deflection, in')
% figure(112)
% hold all
% plot(-X1dot(tTime,amp,freq)+amp/2,disk_def,'.'),grid on, xlabel('Velocity,in/sec'),ylabel('Disk
tip deflection, in')
% --------------------------------------------------------------------
function shim_menu_Callback(~, ~, ~)
% hObject    handle to shim_menu (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% --------------------------------------------------------------------
function single_disk_Callback(~, ~, ~)
% hObject    handle to single_disk (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global R1 R2 th deflection_case h E nu  pre_def
deflection_case=1;
prompt = {'R1 [in]','R2 [in]','th [in]','E [Psi]','nu []','pre deflection [in]'};
dlg_title = 'Properties of the single disk';
num_lines = 1;
def = {'0.353','0.675','0.012','2.9e7','0.3','0.005'};
answer = inputdlg(prompt,dlg_title,num_lines,def);
R1=str2double(cell2mat(answer(1)));
R2=str2double(cell2mat(answer(2)));
th=str2double(cell2mat(answer(3)));
E=str2double(cell2mat(answer(4)));
nu=str2double(cell2mat(answer(5)));
pre_def=str2double(cell2mat(answer(6)));
h=th;
%save(filename,'-append')
% --------------------------------------------------------------------
function Case_A_Callback(~, ~, ~)
% hObject    handle to Case_A (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global deflection_case r1 r2 r3 a l h0 h1 h2 h3 E nu k G IC_A  pre_def
deflection_case=2;
prompt = {'a, inner radius of the disk assembly [in]','r, a vector containing the radi of disks
(largest to smallest) [in]','th, a vector containing the thichnesses of disks [in]','l, location
on the disk assembly where pressure acts [in]','E, same for all disks [Psi]','nu, same for all
disks []','pre deflection [in]'};
dlg_title = 'Geometrical properties of the deflecting shim stack';
num_lines = 1;
def = {'0.353','0.675 0.6 0.525','0.012 0.012 0.012','0.45','2.9e7','0.3','0.005'};
```

```
answer = inputdlg(prompt,dlg_title,num_lines,def);
a=str2double(cell2mat(answer(1)));
r=str2num(cell2mat(answer(2)));
th=str2num(cell2mat(answer(3)));
l=str2double(cell2mat(answer(4)));
E=str2double(cell2mat(answer(5)));
nu=str2double(cell2mat(answer(6)));
pre_def=str2double(cell2mat(answer(7)));
r1=r(1);
r2=r(2);
r3=r(3);
k=5/6; %shear coefficient factor for case A
G=E/(2*(1+nu));
IC_A=zeros(39,1);
h0=-th(1)/2;  %h0=-h1
h1=th(1)/2;
h2=h1+th(2);
h3=h2+th(3);
%save(filename,'-append')
% --------------------------------------------------------------------
function Case_B_Callback(~, ~, ~)
% hObject    handle to Case_B (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global deflection_case r1 r2 r3 a l h0 h1 h2 h3 E nu k G IC_B  pre_def r th
deflection_case=3;
prompt = {'a, inner radius of the disk assembly [in]','r, a vector containing the radi of disks
(largest to smallest) [in]','th, a vector containing the thichnesses of disks [in]','l, location
on the disk assembly where pressure acts [in]','E, same for all disks [Psi]','nu, same for all
disks []','pre deflection [in]'};
dlg_title = 'Geometrical properties of the deflecting shim stack';
num_lines = 1;
def = {'0.353','0.675 0.6 0.525','0.012 0.012 0.012','0.45','2.9e7','0.3','0.005'};
answer = inputdlg(prompt,dlg_title,num_lines,def);
a=str2double(cell2mat(answer(1)));
r=str2num(cell2mat(answer(2)));
th=str2num(cell2mat(answer(3)));
l=str2double(cell2mat(answer(4)));
E=str2double(cell2mat(answer(5)));
nu=str2double(cell2mat(answer(6)));
pre_def=str2double(cell2mat(answer(7)));
r1=r(1);
r2=r(2);
r3=r(3);
k=5/6; %shear coefficient factor for case B
G=E/(2*(1+nu));
IC_B=zeros(1,39);
h0=-th(1)/2;  %h0=-h1
h1=th(1)/2;
h2=h1+th(2);
h3=h2+th(3);
%save(filename,'-append')
% --- Executes during object creation, after setting all properties.
function figure1_CreateFcn(~, ~, ~)
% hObject    handle to figure1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
clc
% --- Executes during object creation, after setting all properties.
function axes1_CreateFcn(~, ~, ~)
% hObject    handle to axes1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% --------------------------------------------------------------------
% function stepped_disk_caseII_cpt_Callback(~, ~, ~)
% % hObject    handle to stepped_disk_caseII_cpt (see GCBO)
% % eventdata  reserved - to be defined in a future version of MATLAB
% % handles    structure with handles and user data (see GUIDATA)
% global deflection_case a l h0 h E nu IC_II J r nd filename R
```

```
% deflection_case=4;
% prompt = {'a, inner radius of the disk assembly [in]','r, a vector containing the radi of disks
(largest to smallest) [in]','th, a vector containing the thichnesses of disks [in]','l, location
on the disk assembly where pressure acts [in]','E, same for all disks [Psi]','nu, same for all
disks []'};
% dlg_title = 'Geometrical properties of the deflecting shim stack';
% num_lines = 1;
% def = {'0.1','0.6 0.5 0.3','0.012 0.012 0.012','0.101','2.9e7','0.3'};
% answer = inputdlg(prompt,dlg_title,num_lines,def);
% a=str2double(cell2mat(answer(1)));
% r=str2num(cell2mat(answer(2)));
% th=str2num(cell2mat(answer(3)));
% l=str2double(cell2mat(answer(4)));
% E=str2double(cell2mat(answer(5)));
% nu=str2double(cell2mat(answer(6)));
%
% r=[r a];  %r(n+1)=a,
% for i=1:length(r)
%     if l>r(i)
%         J=i-1;
%         break
%     end
% end
% IC_II=zeros(1,21);
% nd=length(th); %nd is used because n was used for the gas constant in hyd model
%
% h0=-th(1)/2;  %h0=-h1
% h(1)=th(1)/2;
% h(2)=h(1)+th(2);
% h(3)=h(2)+th(3);
%
% R=r(1);
%
% save(filename,'-append')
% --- Executes during object creation, after setting all properties.
function solve_CreateFcn(~, ~, ~)
% hObject    handle to solve (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% --- Executes on button press in Phase_space_plot.
function Phase_space_plot_Callback(~, ~, ~)
% hObject    handle to Phase_space_plot (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global P T
figure(78)
hold all
plot(P(floor(0.8*length(T)):length(T),3),P(floor(0.8*length(T)):length(T),4),'.'),grid on,
xlabel('Sprung mass position Xb'),ylabel('Sprung mass velocity Vb')
figure(79)
hold all
plot(P(floor(0.8*length(T)):length(T),5),P(floor(0.8*length(T)):length(T),6),'.'),grid on,
xlabel('Un-Sprung mass position Xt'),ylabel('Un-Sprung mass velocity Vt')
figure(80)
hold all
plot(P(floor(0.8*length(T)):length(T),3),P(floor(0.8*length(T)):length(T),5),'.'),grid on,
xlabel('Sprung mass position Xb'),ylabel('Un-Sprung mass position Xt')
figure(81)
hold all
plot(P(floor(0.8*length(T)):length(T),4),P(floor(0.8*length(T)):length(T),6),'.'),grid on,
xlabel('Sprung mass velocity Vb'),ylabel('Un-Sprung mass velocity Vt')
% --- Executes on button press in Poincare.
function Poincare_Callback(~, ~, ~)
% hObject    handle to Poincare (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global P T freq
Period=1/freq;
figure(25)
```

```
set(gcf,'Position',[150 650 500 450])
hold on
axis([min(P(:,3)) max(P(:,3)) min(P(:,4)) max(P(:,4))])
figure(26)
hold on
set(gcf,'Position',[150 100 500 450])
axis([min(P(:,5)) max(P(:,5)) min(P(:,6)) max(P(:,6))])
figure(27)
hold on
set(gcf,'Position',[800 650 500 450])
axis([min(P(:,3)) max(P(:,3)) min(P(:,5)) max(P(:,5))])
figure(28)
hold on
set(gcf,'Position',[800 100 500 450])
axis([min(P(:,4)) max(P(:,4)) min(P(:,6)) max(P(:,6))])
TT=0:Period:T(length(T));
PP=interp1(T,P,TT);
for i=floor(length(TT)*0.8):length(TT)  %discarding the transients
        figure(25)
        hold on
        plot(PP(i,3),PP(i,4),'.'),grid on,xlabel('Xb'),ylabel('Vb'),title('Poincare section for
P3 and P4')
        figure(26)
        hold on
        plot(PP(i,5),PP(i,6),'.'),grid on,xlabel('Xt'),ylabel('Vt'),title('Poincare section for
P5 and P6')
        figure(27)
        hold on
        plot(PP(i,3),PP(i,5),'.'),grid on,xlabel('Xb'),ylabel('Xt'),title('Poincare section for
P3 and P5')
        figure(28)
        hold on
        plot(PP(i,4),PP(i,6),'.'),grid on,xlabel('Vb'),ylabel('Vt'),title('Poincare section for
P4 and P6')
end
function ICXb_Callback(hObject, ~, ~)
% hObject    handle to ICXb (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of ICXb as text
%        str2double(get(hObject,'String')) returns contents of ICXb as a double
global Xb0
Xb0 = str2double(get(hObject,'String'));
% --- Executes during object creation, after setting all properties.
function ICXb_CreateFcn(hObject, ~, ~)
% hObject    handle to ICXb (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
global Xb0
Xb0 = str2double(get(hObject,'String'));
function ICVb_Callback(hObject, ~, ~)
% hObject    handle to ICVb (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of ICVb as text
%        str2double(get(hObject,'String')) returns contents of ICVb as a double
global Vb0
Vb0 = str2double(get(hObject,'String'));
% --- Executes during object creation, after setting all properties.
function ICVb_CreateFcn(hObject, ~, ~)
% hObject    handle to ICVb (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
```

```
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
global Vb0
Vb0 = str2double(get(hObject,'String'));
function ICXt_Callback(hObject, ~, ~)
% hObject    handle to ICXt (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of ICXt as text
%        str2double(get(hObject,'String')) returns contents of ICXt as a double
global Xt0
Xt0 = str2double(get(hObject,'String'));
% --- Executes during object creation, after setting all properties.
function ICXt_CreateFcn(hObject, ~, ~)
% hObject    handle to ICXt (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
global Xt0
Xt0 = str2double(get(hObject,'String'));
function ICVt_Callback(hObject, ~, ~)
% hObject    handle to ICVt (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of ICVt as text
%        str2double(get(hObject,'String')) returns contents of ICVt as a double
global Vt0
Vt0 = str2double(get(hObject,'String'));
% --- Executes during object creation, after setting all properties.
function ICVt_CreateFcn(hObject, ~, ~)
% hObject    handle to ICVt (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
global Vt0
Vt0 = str2double(get(hObject,'String'));
function datasave_Callback(hObject, ~, ~)
% hObject    handle to datasave (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of datasave as text
%        str2double(get(hObject,'String')) returns contents of datasave as a double
global filename
filename =(get(hObject,'String'));
% --- Executes during object creation, after setting all properties.
function datasave_CreateFcn(hObject, ~, ~)
% hObject    handle to datasave (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
global filename
filename =(get(hObject,'String'));
% --- Executes on button press in Eigvalues.
function Eigvalues_Callback(~, ~, ~)
% hObject    handle to Eigvalues (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global Mb Mt Ks Kt
M=[Mb 0;0 Mt];
K=[Ks -Ks;-Ks Ks+Kt];
[V,D]=eig(K,M);
nat_freq_1=sqrt(D(1,1))/(2*pi);
nat_freq_2=sqrt(D(2,2))/(2*pi);
warndlg(['natural frequencies of the system are ' num2str(nat_freq_1) 'Hz and '
num2str(nat_freq_2) 'Hz '],'Eigenvalues and Eigenvectors')
disp(['eigenvectors are: '])
disp(V)
% --- Executes on button press in pushbutton22.
function pushbutton22_Callback(~, ~, ~)
% hObject    handle to pushbutton22 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global T P
figure(44)
hold all
plot(T,P(:,3)-P(:,5)),grid on,xlabel('time, sec'),ylabel('suspension compression (relative to
free length conditions (initial set up), compression is negative), in')
% --- Executes on button press in Tire_compression.
function Tire_compression_Callback(~, ~, ~)
% hObject    handle to Tire_compression (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global Xr P T freq amp Kt
figure(33)
hold all
tireforce=zeros(1,length(T));
for i=1:length(T)
    if P(i,5)-Xr(T(i),amp,freq)<0
        tireforce(i)=Kt*(P(i,5)-Xr(T(i),amp,freq));
    end
end
plot(T,tireforce),grid on, xlabel('Time, sec'),ylabel('Tire force, lb')
% --- Executes on button press in animate.
function animate_Callback(~, ~, ~)
% hObject    handle to animate (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global T P Xr freq amp animation_input
tire_freeh=10;
suspension_freeh=20;
CGroad=[0 0];
CGtire=[0 tire_freeh];
CGbody=[0 tire_freeh+suspension_freeh];
widthroad=2;
heightroad=0.2;
widthtire=2;
heighttire=0.5;
widthbody=2;
heightbody=1;
animation_input=[T P Xr(T,amp,freq)];
% --- Executes on button press in savetofile.
function savetofile_Callback(~, ~, ~)
% hObject    handle to savetofile (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global p30 Rp dr s10 s20 s30 m2 Ff1 Ff2 n beta rho Pvap cavhandle Cd3 A3 Cd2 A4 Cd4 freq amp
tspan tincr Mb Mt Kt Ks Xb0 Vb0 Xt0 Vt0 a r th l deflection_case E nu pre_def k G IC_B IC_A h0 h1
h2 h3 R1 R2 P T filename animation_input Ap Ar disk_def tTime Xr
save(filename)
msgbox({['p30 = ' num2str(p30)],['Rp = ', num2str(Rp)],['dr = ', num2str(dr)],['s10 = ',
num2str(s10)],['s20 = ', num2str(s20)],['s30 = ', num2str(s30)],['beta = ', num2str(beta)],['rho
= ', num2str(rho)],['Pvap = ', num2str(Pvap)],['Rp = ', num2str(Rp)],['Cd3 = ',
num2str(Cd3)],['A3 = ', num2str(A3)],['Cd2 = ', num2str(Cd2)],['A4 = ', num2str(A4)],['Cd4 = ',
num2str(Cd4)],['freq = ', num2str(freq)],['amp = ', num2str(amp)],['tspan = ',
```

```
num2str(max(tspan))],['Mb (slug/12) = ', num2str(Mb)],['Mt (slug/12) = ', num2str(Mt)],['Kt = ',
num2str(Kt)],['Ks = ', num2str(Ks)],['Xb0 = ', num2str(Xb0)],['Vb0 = ', num2str(Vb0)],['Xt0 = ',
num2str(Xt0)],['Vt0 = ', num2str(Vt0)],['deflection_case = ', num2str(deflection_case)],['r = ',
num2str(r)],['th = ', num2str(th)],['pre_def = ', num2str(pre_def)]},'Current parameter values')
% --- Executes on button press in showparameters.
function showparameters_Callback(~, ~, ~)
% hObject    handle to showparameters (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global p30 Rp dr s10 s20 s30 m2 Ff1 Ff2 n beta rho Pvap cavhandle Cd3 A3 Cd2 A4 Cd4 freq amp
tspan tincr Mb Mt Kt Ks Xb0 Vb0 Xt0 Vt0 a r th l deflection_case E nu pre_def k G IC_B IC_A h0 h1
h2 h3 R1 R2 P T filename animation_input
msgbox({['p30 = ' num2str(p30)],['Rp = ', num2str(Rp)],['dr = ', num2str(dr)],['s10 = ',
num2str(s10)],['s20 = ', num2str(s20)],['s30 = ', num2str(s30)],['beta = ', num2str(beta)],['rho
= ', num2str(rho)],['Pvap = ', num2str(Pvap)],['Rp = ', num2str(Rp)],['Cd3 = ',
num2str(Cd3)],['A3 = ', num2str(A3)],['Cd2 = ', num2str(Cd2)],['A4 = ', num2str(A4)],['Cd4 = ',
num2str(Cd4)],['freq = ', num2str(freq)],['amp = ', num2str(amp)],['tspan = ',
num2str(max(tspan))],['Mb (slug/12) = ', num2str(Mb)],['Mt (slug/12) = ', num2str(Mt)],['Kt = ',
num2str(Kt)],['Ks = ', num2str(Ks)],['Xb0 = ', num2str(Xb0)],['Vb0 = ', num2str(Vb0)],['Xt0 = ',
num2str(Xt0)],['Vt0 = ', num2str(Vt0)],['deflection_case = ', num2str(deflection_case)],['r = ',
num2str(r)],['th = ', num2str(th)],['pre_def = ', num2str(pre_def)]},'Current parameter values')
function time_incre_Callback(hObject, ~, ~)
% hObject    handle to time_incre (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of time_incre as text
%        str2double(get(hObject,'String')) returns contents of time_incre as a double
global  tincr
tincr = str2double(get(hObject,'String'));
% --- Executes during object creation, after setting all properties.
function time_incre_CreateFcn(hObject, ~, ~)
% hObject    handle to time_incre (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
global  tincr
tincr = str2double(get(hObject,'String'));
% --- Executes on button press in FFT.
function FFT_Callback(~, ~, ~)
% hObject    handle to FFT (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global T P tincr
TT=T(floor(0.8*length(T)):length(T));
PP=P(floor(0.8*length(T)):length(T),:);
Fs=1/tincr;
NFFT = 2^nextpow2(length(TT)); % Next power of 2 from length of y
Y = fft(PP,NFFT)/length(TT);
f = Fs/2*linspace(0,1,NFFT/2+1);
% Plot single-sided amplitude spectrum.
figure(158)
hold all
plot(f,2*abs(Y(1:NFFT/2+1,3))),grid on, axis([0 30 0 10])
title('Single-Sided Amplitude Spectrum of p3')
xlabel('Frequency (Hz)')
ylabel('|Xb|')
figure(159)
hold all
plot(f,2*abs(Y(1:NFFT/2+1,4))),grid on, axis([0 30 0 10])
title('Single-Sided Amplitude Spectrum of p3')
xlabel('Frequency (Hz)')
ylabel('|Vb|')
figure(160)
hold all
plot(f,2*abs(Y(1:NFFT/2+1,5))),grid on, axis([0 30 0 10])
```

```
title('Single-Sided Amplitude Spectrum of p3')
xlabel('Frequency (Hz)')
ylabel('|Xt|')
figure(161)
hold all
plot(f,2*abs(Y(1:NFFT/2+1,6))),grid on, axis([0 50 0 50])
title('Single-Sided Amplitude Spectrum of p3')
xlabel('Frequency (Hz)')
ylabel('|Vt|')
% --- Executes on selection change in listbox1.
function listbox1_Callback(hObject, eventdata, handles)
% hObject    handle to listbox1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: contents = cellstr(get(hObject,'String')) returns listbox1 contents as cell array
%        contents{get(hObject,'Value')} returns selected item from listbox1
global odelist odenum odechoice
odelist=cellstr(get(hObject,'String'));
odenum=get(hObject,'Value');
odechoice=cell2mat(odelist(odenum))
% --- Executes during object creation, after setting all properties.
function listbox1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to listbox1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: listbox controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
global odechoice
odechoice='ode23t';
```

Appendix H. The MATLAB Function for Governing Equations of the Suspension Model

```
function dp=pressure_Englishunit_04Nov10(t,p)
%the function that contains the ODEs for the hydraulic damper and quarter car rig EOM. The
%tip deflection of the shim stack (case B) and stepped disk (case A) is
%approximated by a line as a function of the pressure difference (p0)
%across the shim stack.
global R1 R2 beta Cd2 Cd3 Ar Ap E A2 A3 rho p30 n  V10 V20 V30 x1dot m2 amp freq x1 Xr  Cd4 A4 nu
h AAA tspan def_counter disk_def tTime deflection_case r1 p0 wait slopeA slopeB pre_def Mb Mt Ks
Kt cavhandle Pvap TP contact s20 s30 s10
p0=abs(p(1)-p(2));
%getting the instaneous displ and velocity from the main code
x1=p(5)-p(3);  %x1=Xt-Xb (x1 is the amount that the damper is compressed or extended)
x1dot=p(6)-p(4);  %x1dot=Vt-Vb;
xr=Xr(t,amp,freq);
%DISK DEFLECTION CALCULATIONS
%% A simple deflection disk
%See 201108-1 page 12 sol1
if deflection_case==1
    AAA=h^3*pi/(12*(1-nu^2));
    w_tip_single_disk =
1/64*pi*R2^4*p0/(AAA*E)+1/32*R2^2*pi*p0*(4*R2^4*log(R2)+4*R2^4*nu*log(R2)-
4*nu*R2^2*R1^2*log(R1)+4*R2^2*R1^2*log(R1)-nu*R2^4+R2^4+nu*R1^4-R1^4)/(AAA*E*(-
nu*R1^2+R2^2+R2^2*nu+R1^2))-1/16*pi*p0*R1^2*R2^2*(4*R2^2*log(R2)+4*R2^2*nu*log(R2)-R2^2*nu+R2^2-
4*log(R1)*R2^2-4*log(R1)*nu*R2^2+R1^2+nu*R1^2).*log(R2)/(AAA*E*(-nu*R1^2+R2^2+R2^2*nu+R1^2))-
1/8*R2^2*pi*p0*R2^2.*log(R2)/(AAA*E)+1/16*R2^2*pi*p0*R2^2/(AAA*E)+1/64*pi*p0*R1^2*(R1^4+16*R2^4*l
og(R1)*nu*log(R2)-16*R2^4*log(R1)^2-5*R2^2*R1^2-6*R2^4-nu*R1^4+16*R2^4*log(R1)*log(R2)-
16*R2^4*log(R1)^2*nu+4*nu*R2^2*R1^2*log(R1)-
8*R2^4*nu*log(R2)+4*log(R1)*nu*R2^4+4*R2^2*R1^2*log(R1)-
8*R2^4*log(R2)+12*log(R1)*R2^4+3*R1^2*R2^2*nu-2*nu*R2^4)/(AAA*E*(-nu*R1^2+R2^2+R2^2*nu+R1^2));
    if w_tip_single_disk<pre_def
        A2=0;
    else
        A2=2*pi*R2*(w_tip_single_disk-pre_def);
    end
    disk_def(def_counter)=w_tip_single_disk;

end
%% STEPPED DISK DEFLECTION (Case A)
%Based on 300909-1 page 1 and works done before that
if deflection_case==2
    w_tip_caseA=p0*slopeA;
    if w_tip_caseA<pre_def
        A2=0;
    else
        A2=2*pi*R2*(w_tip_caseA-pre_def);
    end
    %     [ABC_A] = fsolve(@caseA_30sep09_final_for_hyd_mod,IC_A,options);
    %
    %     w_tip_caseA=ABC_A(1)*r1^4+ABC_A(2)*r1^2+ABC_A(3)*log(r1)+ABC_A(4)
    %     *r1^2*log(r1)+ABC_A(5)*r1+ABC_A(6);
    disk_def(def_counter)=w_tip_caseA;
%     figure(18)
%     hold all
%     plot(p0,w_tip_caseA,'.','color','blue');
%     figure(19)
%     hold all
%     plot(t,w_tip_caseA,'.','color','blue');
%     IC_A=ABC_A;
end
%% Shim stack assembly (Case B)
%Based on 300909-1 page 1 and works done before that
if deflection_case==3
%     [ABC_B] = fsolve(@caseB_30sep09_final_for_hyd_mod,IC_B,options);
```

```
%
w_tip_caseB=ABC_B(1)*r1^4+ABC_B(2)*r1^2+ABC_B(3)*log(r1)+ABC_B(4)*r1^2*log(r1)+ABC_B(5)*r1+ABC_B(
6);
    w_tip_caseB=p0*slopeB;
    if w_tip_caseB<pre_def
        A2=0;
    else
        A2=2*pi*R2*(w_tip_caseB-pre_def);
    end
    disk_def(def_counter)=w_tip_caseB;
%     figure(16)
%     hold all
%     plot(p0,w_tip_caseB,'.','color','green');
%     figure(17)
%     hold all
%     plot(t,w_tip_caseB,'.','color','green');
%     IC_B=ABC_B; %values of the solution will be used as IC for the next time step
end
%% Checking the shim stack maximum deflection
%disk deflection cannot be more than a maximum allower value
if A2/(2*pi*R2)>=0.2  %maximum allowed disk deflection is 0.2 in for now
    A2=2*pi*R2*0.2;
    disk_def(def_counter)=0.2;
end
%% Cavitation
if cavhandle==0
    if p(1)<Pvap
            p(1)=Pvap;
            disp(['damper cavitating at t= ' num2str(t)]);
    end
    if p(2)<Pvap
            p(2)=Pvap;
            disp(['Damper Cavitating at t= ' num2str(t)]);
    end
end
%% Finding the piston and damper force
%fp=p(1)*(Ap-Ar)-p(2)*Ap-Ff1*sign(x1dot);
fd=p(1)*(Ap-Ar)-p(2)*Ap;%           +p30*Ar; %p30*Ar is added to Fd to account for static spring
force exerted by the damper in static conditions (See 090610-1 page 21)
tTime(def_counter)=t;
def_counter=def_counter+1;
%% Kinematic conditions check
s3=((p30*V30^n/p(2))^(1/n))/Ap;  %length of chamber 3
if  (p(3)-p(5))>(s10-0.5) && p(4)>p(6) % maximum damper extension ,if the damper pistin is
hitting the body (V1=0)
    vti=p(6);  %velocities before impact
    vbi=p(4);
    p(6)=(Mt*vti+Mb*vbi+Mb*0.5*(vbi-vti))/(Mt+Mb);  %coefficient of restitution (0.5)
    p(4)=(Mt*vti+Mb*vbi+Mt*0.5*(vti-vbi))/(Mt+Mb);
%       p(1)=p30;
%       p(2)=p30;
 %    p(3)=p(5)+s10-0.5;
    disp(['damper is running out of extension stroke at t= ' num2str(t)])
end
if (p(5)-xr)>0  %Checking the tire road separation
    contact=0;
    disp(['tire loosing contact with the road at t= ' num2str(t)]);
else
    contact=1;
end
if (s3<=0.25) && (p(6)>p(4))  %max damper compression happens when chamber 3 becomes small
%(p(5)-p(3))>(s20+s30-0.5) && (p(6)>p(4))    %maximum damper (suspension compression) is s20+s30
    vti=p(6);  %velocities before impact
    vbi=p(4);
    p(6)=(Mt*vti+Mb*vbi+Mb*0.5*(vbi-vti))/(Mt+Mb);  %coefficient of restitution (0.5)
    p(4)=(Mt*vti+Mb*vbi+Mt*0.5*(vti-vbi))/(Mt+Mb);
%     p(5)=p(3)+s20+s30-0.01;
%     p(1)=p30;   %pressure drops when the piston hits the housing
%     p(2)=p30;
```

```
    disp(['maximum suspension compression occured at t= ' num2str(t)]);
end
%% ODE system
dp=zeros(6,1);
   dp(1)=-beta*(Cd2*A2*sign(-p(2)+p(1))*sqrt(2)*sqrt(abs(Cd3^2*A3^2*(p(2)-
p(1))/(Cd3^2*A3^2+Cd2^2*A2^2)))*sqrt(1/rho)+Cd4*A4*sign(-p(2)+p(1))*sqrt(2)*sqrt(abs(-
p(2)+p(1))/rho)+Ap*p(6)-Ap*p(4)-Ar*p(6)+Ar*p(4))/(Ap*p(5)-Ap*p(3)-Ar*p(5)+Ar*p(3)+V10);
   if s3<=0.25 %minimum length of chamber 3 (max damper compression) (see page 7 of 041110-1)
      disp(['chamber 3 minimum length reached at t= ' num2str(t)]);
      dp(2)=n*Ap/(p30^(1/n)*V30*p(2)^(-(1+1/n)))*p(6);
   else
       dp(2)=n*beta*(Cd2*A2*sign(-p(2)+p(1))*sqrt(2)*sqrt(abs(Cd3^2*A3^2*(p(2)-
p(1))/(Cd3^2*A3^2+Cd2^2*A2^2)))*sqrt(1/rho)+Cd4*A4*sign(-p(2)+p(1))*sqrt(2)*sqrt(abs(-
p(2)+p(1))/rho)-Ap*p(4))/(beta*V30*p30^(1/n)*p(2)^(-(1+n)/n)+n*V20+n*Ap*p(3)+n*V30-
n*(p30*V30^n/p(2))^(1/n));
   end
   dp(3)=p(4);
   dp(4)=1/Mb*(-fd-Ks*p(3)+Ks*p(5)-Mb*386.4);
   dp(5)=p(6);
   dp(6)=1/Mt*(fd+Ks*p(3)-(Ks+contact*Kt)*p(5)+contact*Kt*xr-Mt*386.4);
waitbar(t/max(tspan),wait,['Running code, Please wait... t= ' num2str(t)]);
```

References

1. Rabinow, J., *Magnetic Fluid Torque and Force Transmitting Device*. 1952: United States (US), 2575360
2. Ashour, O., C. A. Rogers, and W. Kordonsky, *Magnetorheological Fluids: Materials, Characterization, and Devices.* Journal of Intelligent Material Systems and Structures, 1996. 7(2): p. 123-130.
3. Genc, S., *Synthesis and Properties of Magnetorheological Fluids*, in *School of Engineering*. 2002, University of Pittsburgh.
4. Bossis, G., S. Lacis, A. Meunier, and O. Volkova, *Magnetorheological Fluids.* Journal of Magnetism and, 2002(8): p. 5492-5498.
5. Carlson, J. D., D. M. Catanzarite, and K. A. St. Clair. *Commercial Magneto-Rheological Fluid Devices*. in *5th International Conference on Electro-Rheological Fluids, Magneto-Rheological Suspensions and Associated Technology, July 1995, International Journal of Modern Physics B*. 1996. Sheffield, UK: World Scientific.
6. Havelka, K. O. and P. J. W., *Electrorheological Technology: The Future Is Now*. Chemtec, 1996. 26(6): p. 36-45.
7. Auzans, E., D. Zins, E. Blums, and R. Massart, *Synthesis and Properties of Mn-Zn Ferrite Ferrofluids.* Journal of Materials Science, 1999. 34(6): p. 1253-60.
8. Berkovsky, B. M., V. F. Medvedev, and M. S. Krakov, *Magnetic Fluids: Engineering Applications*. 1993, New York: Oxford University Press.
9. Raj, K. and R. Moskowitz. *Commercial Applications of Ferrofluids*. in *5th International Conference on Magnetic Fluids, 18-22 Sept. 1989, Journal of Magnetism and Magnetic Materials*. 1990. Riga, USSR.
10. Winslow, W. M., *Method and Means for Translating Electrical Impulses into Mechanical Force*. 1947: U.S.
11. Winslow, W. M., *Fluid Controlled Hydraulic Device*. 1953: U.S.
12. Weiss, K. D. and T. G. Duclos. *Controllable Fluids: The Temperature Dependence of Post-Yield Properties*. in *Fourth International Conference on Electrorheological (ER) Fluids, 20-23 July 1993, International Journal of Modern Physics B*. 1993. Feldkirch, Austria.
13. Tang, X. and H. Conrad, *Quasistatic Measurements on a Magnetorheological Fluid*. Journal of Rheology, 1996. 40(6): p. 1167-78.
14. Ginder, J. M., L. D. Elie, and L. C. Davis, *Magnetic Fluid-Based Magnetorheological Fluids*. 1996: U.S.
15. Ginder, J. M., L. C. Davis, and L. D. Elie. *Rheology of Magnetorheological Fluids: Models and Measurements*. in *Proceedings of the 5th International Conference on Electro-Rheological*

Fluids, Magneto-Rheological Suspensions and Associated Technology, 10-14 July 1995. 1996. Sheffield, UK: World Scientific.

16. Rosensweig, R. E. *Directions in Ferrohydrodynamics*. in *Proceedings of the Thirtieth Annual Conference on Magnetism and Magnetic Materials, 27-30 Nov. 1984, Journal of Applied Physics*. 1985. San Diego, CA, USA.
17. Carlson, J. D. and K. D. Weiss, *Magnetorheological Materials Based on Alloy Particles*. 1995: U.S.
18. Kordonsky, W. I. and S. A. Demchuk. *Additional Magnetic Dispersed Phase Improves the Mr-Fluid Properties*. in *Proceedings of the 5th International Conference on Electro-Rheological Fluids, Magneto-Rheological Suspensions and Associated Technology, 10-14 July 1995*. 1996. Sheffield, UK: World Scientific.
19. Guangqiang, Y., *Large-Scale Magnetorheological Fluid Damper for Vibration Mitigation: Modeling, Testing and Control*. 2001, University of Notre Dame.
20. Kordonsky, V. I., Z. P. Shulman, S. R. Gorodkin, S. A. Demchuk, I. V. Prokhorov, E. A. Zaltsgendler, and B. M. Khusid. *Physical Properties of Magnetizable Structure-Reversible Media*. in *5th International Conference on Magnetic Fluids, 18-22 Sept. 1989.* 1990. Riga, USSR.
21. Harris, J., *Rheology of Non-Newtonian Flow*. 1977: Longman gruop limited.
22. Margida, A. J., K. D. Weiss, and J. D. Carlson. *Magnetorheological Materials Based on Iron Alloy Particles*. in *5th International Conference on Electro-Rheological Fluids, Magneto-Rheological Suspensions and Associated Technology, July 1995, International Journal of Modern Physics B*. 1996. Sheffield, UK: World Scientific.
23. Shtarkman, E. M., *Fluid Responsive to a Magnetic Field*. 1991: U.S.
24. Gans, B. J. D., *Magnetorheology of an Inverse Ferrofluid*. 2000, University on Twente: Twente. p. 135.
25. Lemaire, E., G. Bossis, and Y. Grasselli. *Yield Stress and Structuration of Magnetorheological Suspensions*. in *Sixth International Conference on Magnetic Fluids, 20-24 July 1992, Journal of Magnetism and Magnetic Materials*. 1993. Paris, France.
26. Shulman, Z. P., V. I. Kordonsky, E. A. Zaltsgendler, I. V. Prokhorov, B. M. Khusid, and S. A. Demchuk, *Structure, Physical Properties and Dynamics of Magnetorheological Suspensions.* International Journal of Multiphase Flow, 1986. 12(6): p. 935-955.
27. Yalcintas, M., *An Analitical and Experimental Investigation of Electrorheological Material Based Adaptive Structures*. 1995.
28. Ginder, J. M. and L. C. Davis, *Shear Stresses in Magnetorheological Fluids: Role of Magnetic Saturation.* Applied Physics Letters, 1994. 65(26): p. 3410.
29. Lemaire, E., G. Bossis, and O. Volkova. *Deformation and Rupture Mechanisms of Er and Mr Fluids*. in *Proceedings of the 5th International Conference on Electro-Rheological Fluids, Magneto-Rheological Suspensions and Associated Technology, 10-14 July 1995*. 1996. Sheffield, UK: World Scientific.
30. Bossis, G., E. Lemaire, O. Volkova, and H. Clercx, *Yield Stress in Magnetorheological and Electrorheological Fluids: A Comparison between Microscopic and Macroscopic Structural Models.* Journal of Rheology, 1997. 41(3): p. 687-704.

31. Volkova, O., G. Bossis, M. Guyot, V. Bashtovoi, and A. Reks, *Magnetorheology of Magnetic Holes Compared to Magnetic Particles.* Journal of Rheology, 2000. 44(1): p. 91-104.

32. Potanin, A. A., S. M. Shrauti, D. W. Arnold, and A. M. Lane, *Rheological Probing of Structure and Pigment-Resin Interactions in Magnetic Paints.* Rheologica Acta, 1998. 37(1): p. 89-96.

33. Foister, R. T., *Magnetorheological Fluids*. 1997: U.S.

34. Ginder, J. M., *Behavior of Magnetorheological Fluids.* MRS Bulletin, 1998. 23(8): p. 26-29.

35. Shorey, A. B., W. I. Kordonski, S. R. Gorodkin, S. D. Jacobs, R. F. Gans, K. M. Kwong, and C. H. Farny, *Design and Testing of a New Magnetorheometer.* Review of Scientific Instruments, 1999. 70(11): p. 4200-4206.

36. Macosko, C. W., *Book Review: Rheology: Principles, Measurements, and Applications.* Journal of Colloid and Interface Science, 1996. 178(1): p. 382.

37. Laun, H. M., C. Kormann, and N. Willenbacher, *Rheometry on Magnetorheological (Mr) Fluids. I. Steady Shear Flow in Stationary Magnetic Fields.* Rheologica Acta, 1996. 35(5): p. 417.

38. Janocha, H. and B. Rech, *Messungen an Mr-Fluessigkeiten Mit Rotationsviskosimetern-Measurements of Mr-Fluids Using Rotational Viscometers.* Rheology, 1994. 4(4): p. 198-203.

39. Lemaire, E. and G. Bossis, *Yield Stress and Wall Effects in Magnetic Colloidal Suspensions.* Journal of Physics D: Applied Physics, 1991. 24(8): p. 1473.

40. de Gans, B. J., C. Blom, J. Mellema, and A. P. Philipse. *The Development of a Magnetorheometer and Its Use in Investigating Ferrofluids*. 1998. Yonezawa, Japan: World Scientific.

41. Yoshimura, A. and P. K. Prud'homme, *Wall Slip Corrections for Couette and Parallel Disk Viscometers.* Journal of Rheology, 1988. 32(1): p. 53-67.

42. Lemaire, E. and G. Bossis, *Yield Stress and Wall Effects in Magnetic Colloidal Suspensions.* Journal of Physics D (Applied Physics), 1991. 24(8): p. 1473-7.

43. de Gans, B. J., C. Blom, J. Mellema, and A. P. Philipse. *The Development of a Magnetorheometer and Its Use in Investigating Ferrofluids*. in *Proceedings of the 6th International Conference on Electro-Rheological Fluids, Magneto-Rheological Suspensions and their Applications, 22-25 July 1997*. 1998. Yonezawa, Japan: World Scientific.

44. Baltimore, C. V., *Field-Flow Orientation Effects in Magnetorheological Fluids*, in *Civil and Environmental Engineering Dept.* 1998, Duke University.

45. Carlson, J. D. and B. F. Spencer, Jr. *Magnetorheological Fluid Dampers: Scalability and Design Issue for Application to Dynamic Hazard Mitigation*. in *workshop on structural control*. 1996. Hong Kong, China.

46. Baba, K., N. Yabushita, H. Kasumoto, H. Murakami, and H. Inoue. *Model Refrence Adaptive Control on Soli-Structure Interaction Systems under Earthquake Disturbance*. in *structural control*. 1998.

47. Jolly, M. R., J. W. Bender, and J. D. Carlson, *Properties and Applications of Commercial Magnetorheological Fluids.* Journal of Intelligent Material Systems and Structures, 1999. 10(1): p. 5-13.

48. Chin, B. D., J. H. Park, M. H. Kwon, and O. O. Park, *Rheological Properties and Dispersion Stability of Magnetorheological (Mr) Suspensions.* Rheologica Acta, 2001. 40(3): p. 211-19.

49. Phillips, R. W., *Engineering Applications of Fluids with a Vairable Yield Stress*. 1969, University of California, Berkeley: California.

50. Herschel, W. H. and R. Bulkley. *Model for Time Dependant Behavior of Fluids*. in *American Society of Testing Materials*. 1926.

51. Weihua, L., *Rheology of Mr Fluids and Mr Damper Dynamic Response: Experimental and Modeling Approaches*. 2001, Nanyang Technological University: Singapore.

52. Carlson, J. D., *A Growing Attraction to Magnetic Fluids.* Machine Design, 1994. 8: p. 61-66.

53. Carlson, J. D. and M. J. Chrzan, *Magnetorheological Fluid Dampers*. 1994: U.S.

54. Carlson, J. D., M. J. Chrzan, and M. O. James, *Magnetorheological Fluid Dampers*. 1994: U.S.

55. Bansbach, E. A., *Torque Transfer Apparatus Using Magnetorheological Fluids*. 1998: U.S.

56. Carlson, J. D., *Low-Cost Mr Fluid Sponge Devices.* Journal of Intelligent Material Systems and Structures, 1999. 10(8): p. 589-94.

57. Kordonski, W. I. and S. D. Jacobs, *Model of Magnetorheological Finishing.* Journal of Intelligent Material Systems and Structures, 1996. 7: p. 131-137.

58. Carlson, J. D. and B. F. Spencer, Jr. *Magneto-Rheological Fluid Dampers for Semi-Active Seismic Control*. in *Int. Conf. on motion and vibraiton control*. 1996. Chiba.

59. Phule, P. P., *Synthesis of Novel Magnetorheological Fluids.* MRS Bulletin, 1998. 23(8): p. 23-5.

60. Jacobs, S. D., S. R. Arrasmith, I. A. Kozhinova, L. L. Gregg, A. B. Shorey, H. J. Romanofsky, D. Golini, W. I. Kordonski, P. Dumas, and S. Hogan, *Mrf: Computer-Controlled Optics Manufacturing.* American Ceramic Society Bulletin, 1999. 78(12): p. 42-48.

61. Kavlicoglu, B. M., F. Gordaninejad, C. A. Evrensel, N. Cobanoglu, Y. Liu, A. Fuchs, and G. Korol. *A High-Torque Magneto-Rheological Fluid Clutch*. in *Smart Structures and Materials 2002: Damping and Isolation, 18-20 March 2002, Proceedings of the SPIE - The International Society for Optical Engineering*. 2002. San Diego, CA, USA: SPIE-Int. Soc. Opt. Eng.

62. Hiemenz, P. J., *Principle of Colloid and Surface Chemistry*. Vol. 4. 1986, New York: Marcel Dekker, Inc.

63. Rankin, P. J., A. T. Horvath, and D. J. Klingenberg, *Magnetorheology in Viscoplastic Media.* Rheologica Acta, 1999. 38(5): p. 471-7.

64. Toy, M. L., L. E. Scriven, C. W. Macosko, N. K. Nelson, Jr., and R. D. Olmsted, *Nonhomogeneities in Couette Flow of Ferrite Suspensions.* Journal of Rheology, 1991. 35(5): p. 887-920.

65. Acrivos, A., *Shear-Induced Particle Diffusion in Concentrated Suspensions of Noncolloidal Particles.* Journal of Rheology, 1995. 39(5): p. 813-26.

66. Barnes, H. A., *A Review of the Slip (Wall Depletion) of Polymer Solutions, Emulsions and Particle Suspensions in Viscometers: Its Cause, Character, and Cure.* Journal of Non-Newtonian Fluid Mechanics, 1995. 56(3): p. 221-51.

67. de Vicente, J., M. T. Lopez-Lopez, J. D. G. Duran, and F. Gonzalez-Caballero, *Shear Flow Behavior of Confined Magnetorheological Fluids at Low Magnetic Field Strengths.* Rheologica Acta, 2004. 44(1): p. 94-103.

68. Weiss, K. D., J. D. Carlson, and D. A. Nixon, *Viscoelastic Properties of Magneto- and Electro-Rheological Fluids.* Journal of Intelligent Material Systems and Structures, 1994. 5(6): p. 772-5.

69. McLeish, T. C. B., T. Jordan, and M. T. Shaw, *Viscoelastic Response of Electrorheological Fluids. I. Frequency Dependence.* Journal of Rheology, 1991. 35(3): p. 427-48.

70. Korobko, E. V. and Z. P. Shul'man. *Viscoelastic Behavior of Electrorheological Fluids*. in *Int. Conf. on ERFs*. 1990. Techomic, Lancaster, PA.

71. Brooks, D., J. Goodwin, C. Hjelm, and C. Zukoski. *Viscoelastic Studies on an Electrorheological Fluid*. in *IEE Colloquium on Electrically Active Fluids (Digest No. 14), 13 Feb. 1985*. 1985. London, UK: IEE.

72. Shulman, Z. P., E. V. Korobko, and Y. G. Yanovskii, *The Mechanism of the Viscoelastic Behaviour of Electrorheological Suspensions.* Journal of Non-Newtonian Fluid Mechanics, 1989. 33(2): p. 181-96.

73. Huang, J., J. Q. Zhang, Y. Yang, and Y. Q. Wei. *Analysis and Design of a Cylindrical Magneto-Rheological Fluid Brake*. in *10th International Manufacturing Conference in China (IMCC 2002), Oct 11 2002*. 2002. Fujian, China: Elsevier Science Ltd.

74. Bossis, G., H. Clercx, Y. Grasselli, and E. Lemaire. *Theoretical Analysis of Field-Induced Structures in E.R. And M.R. Fluids*. in *Fourth International Conference on Electrorheological (ER) Fluids, 20-23 July 1993, International Journal of Modern Physics B*. 1993. Feldkirch, Austria.

75. Kordonsky, W. I. and S. R. Gorodkin. *Magnetorheological Fluid-Based Seal*. in *Proceedings of the 5th International Conference on Electro-Rheological Fluids, Magneto-Rheological Suspensions and Associated Technology, 10-14 July 1995*. 1996. Sheffield, UK: World Scientific.

76. Ginder, J. M. and L. C. Davis, *Shear Stresses in Magnetorheological Fluids: Role of Magnetic Saturation.* Applied Physics Letters, 1994. 65(26): p. 3410-12.

77. Tao, R., *Super-Strong Magnetorheological Fluids.* Liquids and Soft Matter, Journal of Physics Condensed Matter, 2001. 13(50): p. 979-999.

78. Gulley, G. L. and R. Tao, *Static Shear Stress of Electrorheological Fluids.* Physical Review E (Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics), 1993. 48(4): p. 2744-51.

79. Tao, R. and J. M. Sun, *Three-Dimensional Structure of Induced Electrorheological Solid.* Physical Review Letters, 1991. 67(3): p. 398-401.

80. Halsey, T. C. and J. E. Martin, *Electrorheological Fluids.* Scientific American, 1993. 269(4): p. 58-64.

81. Chen, T.-j., X. Zhang, R. N. Zitter, and R. Tao, *Deformation of an Electrorheological Chain under Flow.* Journal of Applied Physics, 1993. 74(2): p. 942.

82. Chen, T.-J., R. N. Zitter, and R. Tao, *Laser Diffraction Determination of the Crystalline Structure of an Electrorheological Fluid.* Physical Review Letters, 1992. 68(16): p. 2555-2558.

83. Tang, X., X. Zhang, R. Tao, and Y. Rong, *Structure-Enhanced Yield Stress of Magnetorheological Fluids.* Journal of Applied Physics, 2000. 87(5): p. 2634-2638.

84. Tang, X., X. Zhang, R. Tao, and Y. Rong, *Structure-Enhanced Yield Stress of Magnetorheological Fluids.* Journal of Applied Physics, 2000. 87(5): p. 2634-8.

85. Tang, X., X. Zhang, and R. Tao. *Enhance the Yield Shear Stress of Magnetorheological Fluids*. in *7th International Conference on Electro-Rheological Fluids and Magneto-Rheological Suspension, 9-23 July 1999, International Journal of Modern Physics B*. 2001. Honolulu, HI, USA: World Scientific.

86. Tao, R. and Q. Jiang, *Simulation of Structure Formation in an Electrorheological Fluid.* Physical Review Letters, 1994. 73(1): p. 205.

87. Mazlan, S. A., N. B. Ekreem, and A. G. Olabi, *The Performance of Magnetorheological Fluid in Squeeze Mode.* Smart Materials and Structures, 2007. 16(5): p. 1678-1682.

88. Farjoud, A., R. Cavey, A. Mehdi, and M. Craft, *Magneto-Rheological Fluid Behavior in Squeeze Mode.* Journal of Smart Materials and Structures, 2009. 18(9): p. 8.

89. Carlson, J. D. and M. R. Jolly, *Mr Fluid, Foam and Elastomer Devices.* Mechatronics, 2000. 10(4-5): p. 555-69.

90. Mazlan, S. A., N. B. Ekreem, and A. G. Olabi, *Apparent Stress-Strain Relationships in Experimental Equipment Where Magnetorheological Fluids Operate under Compression Mode.* Journal of Physics D: Applied Physics, 2008. 41(9): p. 095002 (6 pp.).

91. Kulkarni, P., C. Ciocanel, S. L. Vieira, and N. Naganathan, *Study of the Behavior of Mr Fluids in Squeeze, Torsional and Valve Modes.* Journal of Intelligent Material Systems and Structures, 2003. 14(2): p. 99-104.

92. Tian, Y., S. Wen, and Y. Meng, *Compressions of Electrorheological Fluids under Different Initial Gap Distances.* Physical Review E, 2003. 67(5): p. 1-6.

93. Forte, P., M. Parterno, and E. Rustighi, *A Magnetorheological Fluid Damper for Rotor Applications.* International journal of rotating machinery, 2004. 10: p. 175-182.

94. Ahn, K. K., H. J. Y., and B. S. Yang, *A New Type Controllable Squeeze Film Damper Using an Electromagnet.* Journal of Vibration and Acoustics, 2004. 126: p. 380-383.

95. Carmignani, C., P. Forte, and E. Rustighi, *Design of a Novel Magneto-Rheological Squeeze-Film Damper.* Smart Materials and Structures, 2006. 15(1): p. 164-70.

96. Wang, J., G. Meng, N. Feng, and E. J. Hahn, *Dynamic Performance and Control of Squeeze Mode Mr Fluid Damper-Rotor System.* Smart Materials and Structures, 2005. 14(4): p. 529-39.

97. Wang, J., N. Feng, G. Meng, and E. J. Hahn, *Vibration Control of Rotor by Squeeze Film Damper with Magnetorheological Fluid.* International journal of intelligent material systems and structures, 2006. 17: p. 353-357.

98. Southern, B. M., *Design and Characterization of Tunable Magneto-Rheological Fluid-Elastic Mounts*, in *Mechanical Engineering*. 2008, Virginia Polytechnic Institute and State University: Blacksburg, VA.

99. Engmann, J., C. Servais, and A. S. Burbidge, *Squeeze Flow Theory and Applications to Rheometry: A Review.* Journal of Non-Newtonian Fluid Mechanics, 2005. 132(1-3): p. 1-27.

100. Stefan, J. and K. Sitzgber, *Versuche Uber Die Scheinbare Adhasion.* Sitz. Kais. Akad. Wiss. Math. Natur. Wien., 1874. 69(2): p. 713-735.

101. Scott, J. R., *Theory and Application of the Parallel-Plate Plastometer.* Trans. Inst. Rubber. Ind., 1931. 7: p. 169-186.

102. McClelland, M. A. and B. A. Finlayson, *Squeezing Flow of Elastic Liquids.* Journal of Non-Newtonian Fluid Mechanics, 1983. 13: p. 181-201.

103. Roussel, N., C. Lanos, and Z. Toutou, *Identification of Bingham Fluid Flow Parameters Using a Simple Squeeze Test.* Journal of Non-Newtonian Fluid Mechanics, 2006. 135(1): p. 1-7.

104. Adams, M. J., I. Aydin, B. J. Briscoe, and S. K. Sinha, *A Finite Element Analysis of the Squeeze Flow of an Elasto-Viscoplastic Paste Material.* Journal of Non-Newtonian Fluid Mechanics, 1997. 71: p. 41-57.

105. Florides, G., a. Alexandrou, and G. Georgiou, *Flow Development in Compression of a Finite Amount of a Bingham Plastic*. Journal of Non-Newtonian Fluid Mechanics, 2007. 143(1): p. 38-47.

106. Wilson, S. D. R., *Squeezing Flow of a Bingham Material*. Journal of Non-Newtonian Fluid Mechanics, 1993. 47: p. 211-19.

107. Lawrence, C. J. and G. M. Corfield. *Non-Viscometric Flow of Viscoplastic Materials: Squeeze Flow*: Imperial college press.

108. Smyrnaios, D. N. and J. A. Tsamopoulos, *Squeeze Flow of Bingham Plastics.* Journal of Non-Newtonian Fluid Mechanics, 2001. 100(1-3): p. 165-90.

109. Papanastasiou, T. C., *Flows of Materials with Yield.* Journal of Rheology, 1987. 31(5): p. 385-404.

110. Roussel, N. and C. Lanos, *Plastic Fluid Flow Parameters Identification Using a Simple Squeezing Test.* Applied Rheology, 2003. 13: p. 132-141.

111. Adams, M. J., B. Edmondson, D. G. Caughey, and R. Yahya, *An Experimental and Theoretical Study of the Squeeze-Film Deformation and Flow of Elastoplactic Fluids.* Journal of Non-Newtonian Fluid Mechanics, 1994. 51: p. 61-78.

112. Meeten, G. H., *Yield Stress of Structured Fluids Measured by Squeeze Flow.* Rheologica Acta, 2000. 39(4): p. 399-408.

113. Sherwood, J., *Model-Free Inversion of Squeeze-Flow Rheometer Data.* Journal of Non-Newtonian Fluid Mechanics, 2005. 129(2): p. 61-65.

114. Meeten, G., *Effects of Plate Roughness in Squeeze-Flow Rheometry*. Journal of Non-Newtonian Fluid Mechanics, 2004. 124(1-3): p. 51-60.

115. Matsoukas, A. and E. Mitsoulis, *Geometry Effects in Squeeze Flow of Bingham Plastics.* Journal of Non-Newtonian Fluid Mechanics, 2003. 109: p. 231-240.

116. Carlson, J. and B. F. Spencer, *Magnetorheological Fluid Dampers:Vscalability and Design Issue for Application to Dynamic Hazard Mitigation*, in *Workshop on structural control*. 1996: Hong Kong, China.

117. White, F. M., *Fluid Mechanics*. 3rd ed. 1994, New York: McGraw-Hill. 736.

118. Jolly, M. R. and J. D. Carlson. *Controllable Squeeze Film Damping Using Magnetorheological Fluid*. 1996. Bremen, Germany: AXON Technologie Consult GmbH.

119. Susan-resiga, D., *A Rheological Model for Magneto-Rheological Fluids.* Journal of Intelligent Material Systems and Structures, 2009. 20: p. 1001-1010.

120. Bird, R. B., R. C. Armstrong, and O. Hassager, *Dynamics of Polymeric Liquids*. 1987: John Wiley & Sons.

121. Jolly, M. R., J. D. Carlson, and B. C. Munoz, *A Model of the Behavior of Magnetorheological Materials.* Smart Materials and Structures, 1996. 5(5): p. 607-14.

122. Ciocanel, C., G. Lipscomb, and N. G. Naganathan. *A Constitutive Equation for Magnetorheological Fluid Characterization*. in *Smart Structures and Materials 2005 - Active Materials: Behavior and Mechanics, Mar 7-10 2005*. 2005. San Diego, CA, United States: International Society for Optical Engineering, Bellingham, WA 98227-0010, United States.

123. Dorfmann, A., R. W. Ogden, and A. S. Wineman, *A Three-Dimensional Non-Linear Constitutive Law for Magnetorheological Fluids, with Applications.* International Journal of Non-Linear Mechanics, 2007. 19(5): p. 565-572.

124. Chan, T. W. and D. G. Baird, *An Evaluation of a Squeeze Flow Rheometer for the Rheological Characterization of a Filled Polymer with a Yield Stress.* Rheologica Acta, 2002. 41(3): p. 245-56.

125. Cavey, R., *Design and Development of a Squeeze-More Rheometer for Evaluating Magneto-Rheological Fluids*, in *Mechanical Engineering Dept.* 2008, Virginia Polytechnic Institute and State University: Blacksburg. p. 152.

126. Roark, R. J. and W. C. Young, *Roark's Formulas for Stress and Strain*. sixth ed. 1989, New York: McGraw-Hill.

127. Wahl, A. M. and G. Lobo. *Stresses and Deflections in Flat Circular Plates with Central Holes*. in *American Society of Mechanical Engineers -- Meeting, Oct 7-10 1929*. 1929. New York, NY, United States: American Society of Mechanical Engineers (ASME).

128. Essenburg, F. and S. T. Gulati. *On Contact of Two Axisymmetric Plates*. in *ASME Meeting APMW-26, Aug 30-Sep 1 1965*. 1965. New York, NY, United States: American Society of Mechanical Engineers (ASME).

129. Smith, J. C. V. *Effect of Shear Deformation on Large Deflections of Circular Sandwich Plates*. in *8th AIAA-ASME -- Structures, Structural Dynamics and Matls Conference, Mar 29-31 1967*. 1967.

130. Turvey, G. J., *Effect of Shear Deformation on the Yielding of Circular Plates.* Computers and Structures, 1984. 18(2): p. 307-310.

131. Lee, K. H., N. R. Senthilnathan, and S. P. Lim, *Axisymmetric Bending of Thick Circular Plates.* Mechanics Research Communications, 1990. 17(2): p. 111-116.

132. Wang, C. M., *Deflection of Sandwich Plates in Terms of Corresponding Kirchhoff Plate Solutions.* Archive of Applied Mechanics, 1995. 65(Copyright 1996, FIZ Karlsruhe): p. 408-14.

133. Hyer, M. W. and A. B. Jilani, *Predicting the Axisymmetric Manufacturing Deformations of Disk-Style Benders.* Journal of Intelligent Material Systems and Structures, 2000. 11(5): p. 370-381.

134. Chung, K. C. and C. M. Wang, *Optimal Design of Stepped Circular Plates with Allowance for the Effect of Transverse Shear Deformation.* International Journal of Mechanical Sciences, 2002. 44(Copyright 2002, IEE): p. 1163-77.

135. Gould, P. L., *Analysis of Shells and Plates*. 1999, Upper Saddle River, N.J.: Prentice Hall.

136. Ventsel, E. and T. Krauthammer, *Thin Plates and Shells: Theory, Analysis, and Applications*. 2001, New York: Marcel Dekker.

137. Hyer, M. W. and S. R. White, *Stress Analysis of Fiber-Reinforced Composite Materials*. 2009, Lancaster, Pa: DEStech Publications, Inc.

138. Shames, I. H. and C. L. Dym, *Energy and Finite Element Methods in Structural Mechanics*. 1985, Washington: Hemisphere Pub. Corp.

139. Hutchinson, J. R., *Shear Coefficients for Timoshenko Beam Theory.* Transactions of the ASME. Journal of Applied Mechanics, 2001. 68(Copyright 2001, IEE): p. 87-92.

140. Kaneko, T., *On Timoshenko's Correction for Shear in Vibrating Beams.* Journal of Physics D: Applied Physics, 1975. 8(16): p. 1927-1936.

141. Puchegger, S., S. Bauer, D. Loidl, K. Kromp, and H. Peterlik, *Experimental Validation of the Shear Correction Factor.* Journal of Sound and Vibration, 2003. 261(1): p. 177-184.

142. Dixon, J. C., *Tires, Suspension and Handling* 1996, Warrendale, PA: Society of Automotive Engineers. 621.

143. Talbott, M. S. and J. Starkey, *An Experimentally Validated Physical Model of a High-Performance Mono-Tube Damper*, in *SAE Motorsports Engineering Conference and Exhibition*. 2002, Society of Automotive Engineers: Indianapolis, Indiana.

144. van Kasteel, R., C.-G. Wang, L. Qian, J.-Z. Liu, and G.-H. Ye, *A New Shock Absorber Model for Use in Vehicle Dynamics Studies.* Vehicle System Dynamics, 2005. 43(9): p. 613-31.

145. Eyres, R. D., A. R. Champneys, and N. A. J. Lieven, *Modelling and Dynamic Response of a Damper with Relief Valve.* Nonlinear Dynamics, 2005. 40: p. 119-47.

146. Simms, A. and D. Crolla, *The Influence of Damper Properties on Vehicle Dynamic Behaviour*, in *SAE 2002 World Congress*. 2002, SAE: Detroit, Michigan.

147. Duym, S. W. R., *Simulation Tools , Modelling and Identification , for an Automotive Shock Absorber in the Context of Vehicle Dynamics.* Vehicle System Dynamics, 2000. 33: p. 261-285.

148. Lang, H. H., *A Study of the Characteristics of Automotive Hydraulic Dampers at High Stroking Frequencies*, in *Mechanical Engineering dept.* 1977, University of Michigan.: Ann Arbor. p. 231.

149. Reybrouck, K., *A Nonlinear Parametric Model of an Automotive Shock Absorber*, in *SAE International Congress and Exposition*. 1994, SAE: Detroit, Michigan.

150. Lee, K., *Numerical Modelling for the Hydraulic Performance Prediction of Automotive Monotube Dampers.* Vehicle System Dynamics, 1997. 28(1): p. 25-39.

151. Herr, F., T. Mallin, J. Lane, and S. Roth. *Flow Analysis and Modeling of Shock Absorbers*. in *Proceedings of the 1998 ASME International Mechanical Engineering Congress and Exposition, November 15, 1998 - November 20, 1998*. 1998. Anaheim, CA, USA: ASME.

152. Duym, S., R. Stiens, G. V. Baron, and K. Reybrouck, *Phyisical Modeling of the Hysteretic Behaviour of Automotive Shock Absorbers*, in *SAE International Congress and Exposition*. 1997, Society of Automotive Engineers: Detroit, Michigan. p. 125-137.

153. Duym, S., *An Alternative Force State Map for Shock Absorbers.* Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering, 1997. 211(3): p. 175–179.

154. Surace, C., K. Worden, and G. R. Tomlinson, *An Improved Nonlinear Model for an Automotive Shock Absorber.* Journal of Nonlinear Dynamics, 1991. 3: p. 413-429.

155. Moon, B. Y. and C. T. Lee, *Simulation and Experimental Validation of Vehicle Dynamic Characteristics for Displacement-Sinsitive Shock Absorber Using Fluid-Flow Modeling.* Mechanical Systems and Signal Processing, 2006. 20: p. 373-88.

Index