Structural Shape Optimization of

Three Dimensional Finite Element Models

by

Stephen A. Hambric

Thesis submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

Master of Science

in

Mechanical Engineering

APPROVED:

Dr. Charles E. Knight, Chairman

Dr. Charles F. Reinholtz

Dr. Robert H. Fries

August 26, 1987

Blacksburg, Virginia

Structural Shape Optimization of

Three Dimensional Finite Element Models

by

Stephen A. Hambric

Dr. Charles E. Knight, Chairman

Mechanical Engineering

(ABSTRACT)

The thesis presents a three dimensional shape optimization program which analyzes models made up of linear isoparametric elements. The goal of the program is to achieve a near uniform model stress state and thereby to minimize material volume.

The algorithm is iterative, and performs two analyses per iteration. The first analysis is a static stress analysis of the model for one or more load cases. Based on results from the static analysis, an expansion analysis is performed. Model elements are expanded or contracted based on whether they are stressed higher or lower than a reference stress. The shape changing is done by creating an expansion load vector using the differences between the calculated element stresses and the reference stress. Expansion displacements are solved for, and instead of using them to calculate stresses, the displacements are added to the nodal coordinates to reshape the structure. This process continues until a user defined convergence tolerance is met.

Four programs were used for the analysis process. Models were created using a finite element modeling program called I-DEAS or CAEDS. The I-DEAS output files were converted to input files for the optimizer by a conversion program. The model was optimized using the shape optimization process described above. Post- processing was done using a program written with a graphical programming language called graPHIGS.

Models used to test the program were: a cylindrical pressure vessel with nonuniform thickness, a spherical pressure vessel with nonuniform thickness, a torque arm, and a draft sill casting of a railroad hopper car. Results were compared to similar studies from selected references.

Both pressure vessels converged to near uniform thicknesses, which compared well with the reference work. In a two dimensional analysis, the torque arm volume decreased 24 percent, which compared well with published results. A three dimensional analysis showed a volume reduction of 13 percent, but there were convergence problems. Finally, the draft sill casting was reduced in volume by 9 percent from a manually optimized design.

Acknowledgements

My greatest thanks go to the members of my advisory committee: Dr. Charles E. Knight, Dr. Charles F. Reinholtz, and Dr. Robert H. Fries. Special thanks to Dr. Knight, who provided excellent guidance and advice, and allowed me freedom to move in my own directions on research; and to Dr. Reinholtz, who was extremely helpful with the preparation for my defense.

I also extend thanks to: , for willingly answering my dumbest questions about shape optimization; the Mechanical Engineering Graduate Students, most notably ; who helped me forget my research frustrations by frequently joining me in consuming many gallons of cheap beer; , for answering numerous questions concerning graPHIGS and other CAD problems without yelling at me; and the professors and administrative staff of the Mechanical Engineering Department, who treated me with respect and as a fellow engineer all year.

Finally, for providing much needed love and moral support, I thank my longtime girlfriend ; my parents ; and my other relatives around the world.

Table of Contents

List of Illustrations

Chapter 1

INTRODUCTION

Shape optimization is the changing of the geometry of a machine part so that its stress distributions are smoothed and model volume is reduced. A part performs better if there are no wild fluctuations in stress distribution; and less volume means less material, which is an advantage when parts are mass produced.

Optimization is commonly done by hand using trial and error techniques. The engineer removes or adds material as needed, basing these changes on testing or in-service failure. Engineers now have finite element analysis to check redesigns more easily, and it also has the potential for shape optimization. Much research has been done (see literature review) and many methods exist for shape optimization. However most methods are two dimensional and have no standard approaches. In many previous algorithms, different models require specialized preparation before they are optimized. Sometimes large regions of a model will be held fixed to force the structure into some predetermined shape. Sometimes programming will be altered to accommodate different models. No truly general method exists.

The goal here is to create a three dimensional shape optimization method that may be applied to any part without altering the algorithm used, or requiring excessive preparation. This study investigates the use of an element volume expansion and contraction algorithm to alter a structure's shape. If the algorithm is successful, it could become a simple method of shape optimization.

The program would first perform a normal finite element analysis on a part. Strain energy densities for each element are calculated and compared to a user defined reference stress. The elements would then be changed based on the strain energy density differences. An overly stressed element would expand, and an understressed element would shrink. The process would then repeat until the model stresses converged about the reference stress. This method could be applied to any three dimensional model made up of hexahedron isoparametric elements.

This thesis starts with a literature review to describe the history of shape optimization. In the review, some common shape optimization variables are defined, and different methods are described. Following the literature review, a detailed overview of the main body is presented. The main body of the report contains: the model optimization procedure explaining how software is used in the shape optimization process; the theories used in the construction of this software; case studies of models optimized by the software, with comparisons to similar studies by reference authors; and a conclusion with suggestions for future work.

Chapter 2

LITERATURE REVIEW

As mentioned earlier, shape optimization is the altering of a machine part's geometry to achieve a smooth stress state and volume reduction. A survey of structural shape optimization studies is found in the paper by Haftka [1]. The report contains 139 references that deal mainly with shape optimization of the boundaries of two and three dimensional bodies. The difficulties of working with constantly changing finite element models are discussed. These include problems such as the choice of design variables, and the effects of automatic mesh generation.

Shape optimization generally involves using one of two types of design variables. The first type is a sizing design variable, which is a simple geometric dimension of a structure. Examples are a plate thickness or a bar cross sectional area. These variables are altered so a part will meet a design stress requirement. The second type of variable is a shape design variable, which is used for the more solid and complex geometries analyzed by finite element analysis.

An example of shape design variables are boundary nodal coordinates. An algorithm calculates stresses at elements on a defined boundary and alters the shape of the boundary based on those stresses. Oda and Yamazaki [2] use such a technique in their study using models made up of axisymmetric elements.

They obtained optimum shapes by iteratively changing models until they had become fully stressed shapes. A fully stressed structure has a near constant stress field and a minimum amount of material. Since all of their models were thick walled vessels under internal pressure, the stresses at the outer boundaries were analyzed, and the boundary nodal coordinates were used as design variables. When the outer walls of each model became fully stressed, the shape was optimized. The program developed here uses the fully stressed shape approach, but uses the nodal coordinates of the entire model as design variables. Strain energy densities for all elements are calculated, and nodal coordinates are modified based on them. Oda and Yamazaki [2] use pressure vessel models to support their solution. In this study a spherical and a cylindrical model are compared to their results.

Other programs use the nodes of elements in a certain model region as design variables. Stresses are calculated for the entire model, but only the nodes of the defined regions are allowed to move while all other areas remain fixed. An example of this method is found in Braibant and Fleury's study [3].

They use boundary nodes as design variables. To change their models, only certain nodes, called masters, have displacements calculated for them in the shape change algorithm. The displacements for each iteration are found using a sensitivity analysis. Differential equations based on virtual work, with master node displacements as the variables, are solved. Only a few master nodes may be used, or the solution will require excessive computer time.

A problem that arises often in shape optimization is that sometimes a model will change so much that element distortion causes the analysis to become inaccurate. To solve this problem, some shape optimization algorithms use automatic mesh refinement. The mesh refiner will redefine the entire mesh at certain points in the analysis, and also give high stress regions a greater element concentration. Bennett and Botkin [4] use mesh regeneration in their two dimensional optimizer.

They define only the boundaries of their models and the mesh generator creates an initial mesh. A finite element analysis is performed on the structure, and based on the stress results, the mesh refiner regenerates the mesh giving areas of stress concentration more elements. The finite element analysis is performed again, and the defined boundary elements are changed based on the new stress results to modify the structure. This process is long and costly, and restricts models to two dimensions. A torque arm is used to test their algorithm. In this thesis, a similar model is constructed, but a three dimensional analysis is done in addition to the two dimensional one.

A common element of each of these studies is that in the shape changing portion of their algorithms, only the boundaries of structures are altered. This is due to the difficulty and expense of solving sensitivity equations. Design elements must be specified with great care to achieve useful optimizations. In this thesis, a linear approximation to the nonlinear problem of shape changing is attempted. The algorithm will use linear shape modifications to iteratively solve the nonlinear problem and achieve an optimum shape solution. All the nodes of a model may be moved to reshape the structure without requiring excessive calculations and computer time.

After a normal finite element analysis of a model, an expansion load vector is formed based on element stresses. A new set of displacements is calculated using another normal finite element analysis, and is added to the current set of nodal coordinates. The degree of expansion or contraction in an iteration is specified by the user. Note that

no sensitivity analysis is required, and that all nodes in the model may be used as design variables.

Chapter 3

OVERVIEW

This section provides a detailed overview of the main body of the thesis. Summaries of each chapter are given. The thesis is divided into three main sections: software use, software theory and development, and case studies.

3.1 Software Use

Chapter 4 explains how a model is created, optimized, and post processed using a mix of commercial and local software. The discussions concentrate on how to use these programs.

In this study, a model is made up of two sets of information: the static case model, which is solved using a standard finite element analysis; and the expansion phase information, which determines how the model will change shape.

The static case information is entered using the commercial software I-DEAS or CAEDS, a product of the Structural Dynamics Research Corporation in Cincinnati, Ohio. The finite element module is used in this case to define the required information. The shape of a model is defined by geometric primitives. Meshes are then generated over the shape of the model. Note these meshes are not defined automatically, but the user has full control over the final mesh. Static constraints and loads are also defined. A data file is written when the model definition is complete.

The expansion phase information is entered using the program CONVERT [6] , which also acts as a data conversion program for the data file written by I-DEAS. Expansion data is entered using a menu format. When the program is finished, the data files used by STOPFEP are written.

STOPFEP [6] performs the shape optimization of the model defined by I-DEAS and CONVERT. The structure is reshaped iteratively until a user defined convergence tolerance is met. Two output files are written, one containing element stresses and other execution information; and the other file containing nodal coordinates for each iteration of the optimization process.

The file with the nodal coordinates is fed into the program ANIMATE, which allows its user to quickly animate the entire optimization process on a graphics terminal screen. The model may be moved and zoomed in on easily to observe certain regions.

3.2 Software Theory and Development

Chapter 5 describes theories used and how each program is structured and written. Special functions of CONVERT include constraint plane definition and element

connectivity checking. Since I-DEAS has no way of defining constraint planes, they are defined in CONVERT. Planes for both the static and expansion phase may be defined. Four points define a region to be specified as a constraint plane. The first three points define the plane, and all four points define the boundaries of the region to be constrained. I-DEAS sometimes incorrectly defines elements, so CONVERT checks connectivity as it reads element information in. Any incorrectly defined element is redefined correctly.

STOPFEP uses a standard finite element approach using multiple load cases. Special features of STOPFEP are the expansion phase and the implementation of constraint planes. If a surface not on a primary plane (xy, xz, yz) should be fixed either to model a static constraint or to fix a surface in the expansion phase, it may be held using a constraint plane. All nodes defined on the plane are restricted to movement on that plane.

The expansion phase, which alters the shape of the structure, uses a new load vector created using an initial strain vector. The initial strain vector is created using the expansion coefficient alpha, and the differences between calculated element strain energy densities and the reference stress converted to element strain energy densities. Displacements are calculated for this expansion load vector, and are added to the nodal coordinates to alter the structure.

ANIMATE is written using the FORTRAN programming language with calls to the graphical programming language graPHIGS. The nodal coordinates for each iteration of the optimization are used to create pictures of the model. The pictures are stored in a graphics terminal's memory, and may be drawn immediately on the terminal screen. All elements of the model may be drawn, or only the outer boundaries. If a boundary picture is specified, the program scans all elements to find singly defined lines and draws the outline of the model. Input devices are programmed to position the

model in any orientation, zoom in on certain regions, and animate the optimization process.

3.3 Case Studies

Chapter 6 contains the supporting models used to test the program. Two pressure vessel quarter sections with nonuniform thicknesses are analyzed, one from a cylindrical vessel and the other from a spherical vessel. The goal of the analysis was to achieve uniform model thicknesses for each case. Since the final shape was already known, the models were good tests for the program. Results were compared to theory using Lame's solutions and the results in [2]. Both models achieved near uniform vessel thicknesses, with the cylindrical vessel performing better than the spherical. Stress results were found to be accurate. Model volumes increased in both cases due to the reference stress selected, qut the analyses were only meant to test the program.

A Torque arm with a center hole is analyzed two ways: in two dimensions and in three dimensions. Results are compared to those in reference [4] and stress results checked using a beam approximation. The two dimensional model converged fairly well, although some elements overlapped in the final design. Model volume was reduced by 24 percent. The three dimensional analysis results were not as good as the two dimensional results, although an interesting final shape developed. Model volume was only reduced by 13 percent. Stress results agreed with the beam calculations.

Finally, a model of a rail car draft sill casting was optimized and compared to the work done by Roach [5]. The casting model was originally created by Roach, who optimized it by trial and error techniques. Model volume was reduced by nine percent,

but a distorted final shape resulted. The distortion was caused by the sparseness of the finite element model. To achieve a usable final design, a more detailed mesh would have to be created.

Chapter 4

SOFTWARE USE

This section explains how a model is created, its data files modified, its shape optimized, and how it is post processed. A user's guide containing full documentation for all programs listed here is in reference [6].

4.1 Model Information

A model to be optimized requires two different types of input information to define it. The first type is the static case data for the normal finite element analysis of the model; the second type is expansion case information to determine how the model is allowed to change.

The static case information includes the finite element model itself, made up of nodes and elements; a set of nodal constraints; and up to ten load cases, made up of nodal forces and element face pressures, to simulate the in service state of the model.

The expansion case information includes a different set of constraints, which are used to preserve certain model geometry features such as holes, slots, and walls, during the optimization of the model.

Static or expansion constraints may be simple kinematic nodal degrees of freedom (d.o.f.) constraints or constraint planes. The kinematic constraints are nodes that are fixed in the x, y, or z direction, while constraint planes simulate skewed constraints. Nodes on a constraint plane are restricted to movement on that plane.

Note that constraint planes may only be defined for planes not parallel to the primary planes (xy, xz, yz). Any plane parallel to one of these primary planes may be held fixed by applying kinematic constraints to its nodes.

Also required in the model definition are material properties and optimization control variables. The material properties needed are Young's Modulus and Poisson's ratio. The optimization control variables are the expansion coefficient alpha, which determines how much the element volumes change in an iteration; the reference stress, to which the model stresses will try to converge; and the convergence tolerance, which determines when the optimization is complete.

4.2 Pre-Processing with SDRC - I-DEAS or CAEDS

To define the basic, static case model, the SDRC I-DEAS or CAEDS Finite Element Modeling (FEM) module is used. The FEM module is a finite element model

creation and analysis program, and is used to define nodes, elements, static case kinematic constraints, and load sets made up of nodal forces and element face pressures.

A detailed description of how to create a finite element model like the one described above is in reference [6]. A brief description follows. First, the model's geometry is defined using points, lines, and arcs. Edges are then defined using the geometry primitives, and surfaces are defined by edges. Five or six surfaces may then be used to make up a volume. Several volumes may exist in a single model. Three dimensional meshes are then defined over the volumes, and nodes and elements are generated using the meshes.

With the model complete; nodal constraints, nodal forces, and element face pressures may be added and put into sets. When all inputs are completed, data is written out to a file.

4.3 Data File Conversion with Program CONVERT

Once the model has been defined, the expansion phase information must be added, constraint planes defined for either phase, and the STOPFEP input files required must be written using the information in the I-DEAS data file. The I-DEAS data file may be modified by hand to include the expansion phase information, and the STOPFEP input files created as a result. Hand modification is tedious however, and the model would have to be carefully scanned to find all nodes on desired constraint planes.

The program CONVERT accomplishes all of these tasks by reading the data in the original file, and using an interactive menu format for input. The menu allows the user to define any static phase constraint planes, expansion phase constraint planes, expan-

sion phase nodal constraints, material property information, and optimization control variables. Once all data is entered to the user's satisfaction, the input files for the optimizer are written.

Constraint planes in STOPFEP are defined by the first three nodes of a list of all nodes constrained to motion on that plane. CONVERT simplifies the user input by finding all nodes on a specified plane and putting them in a list.

To define constraint planes for either the static or expansion phase, the user selects a region of the model to be defined as a constraint plane by entering four outer points of the region. The four points define a quadrilateral in space, inside of which are all the nodes on the desired constraint plane. These points may be the coordinates of four nodes on the outside edges of the plane. The program uses the first three points entered to form plane equations and scans all nodes in the region defined by the four points to find those on the plane. When the final data files are output, each plane will be written as a list of nodes, with the first three nodes defining the plane.

Kinematic nodal constraints for the expansion phase are entered in CONVERT. A node's x, y, or z d.o.f. may be held fixed, left free, or released if it was held fixed for the static analysis.

All other variables are entered directly when their corresponding menu choices are selected.

4.4 Shape Optimization with STOPFEP

After that all the information has been entered and compiled in the correct format in the input files, the model is analyzed and optimized by the program STOPFEP

(STructural OPtimization Finite Element Program). The program performs static analyses on the model and modifies it until the user defined convergence tolerance is reached, or terminates by exceeding the maximum allowable number of iterations.

Before a complete optimization run is made, a data check is done to ensure that all required model information exists and is correct. If the model information is correct, a run using only one iteration is performed with all write flags on (static and expansion displacements, current nodal coordinates). The designer then examines the initial stress states and may choose a reference stress for the model to converge about. The expansion displacements can be raised or lowered by altering the expansion coefficient alpha. With these preliminaries complete, the program can then achieve a successful optimization.

The optimization control variables: the expansion coefficient alpha, the reference stress, and the convergence tolerance; are all selected by the user using the initial state of the model and trial and error.

The expansion coefficient alpha determines how much a structure's nodal coordinates are modified in an optimization iteration. The higher alpha is, the higher the nodal displacements are.

Reference stress may be specified based on the stress state of the model. If the designer wishes to remove material from a certain region of a model, the reference stress is set higher than the stresses in that region. The reference stress could also be set equal to a design stress, but obtaining a completely uniform model stress state is rare, and stresses in some regions could be higher than the design stress.

Convergence tolerance will vary depending on model complexity and the value of the expansion coefficient alpha. In general, the tolerance should be raised with complex models, and lowered if alpha is low. Running the program for a few iterations should give the user an estimate for an acceptable convergence tolerance.

No user interaction is required once the program begins execution. Screen writes by the program at certain points in the analysis keep the user informed.

Output includes all the model information and element stresses for each load case and each iteration. If the user desires, static displacements, used to find element stresses; and expansion displacements, used to modify the model, may be written. Included in the output is another file of data, containing element connectivity and nodal coordinates for each iteration. This information is needed for the post processing program ANIMATE.

4.5 Post Processing with Program ANIMATE

To quickly and easily examine the results of the optimization, some sort of graphical post processor is required. The program ANIMATE was written for this purpose. The program allows the user to display the geometric shape after each iteration of the optimization process on a graphics terminal screen, achieving a quick animation of the entire analysis. The user may also rotate the model about the x, y, and z axes; move the model drawing anywhere on the screen; and zoom in on any portion of the model.

All model information is stored in the terminal's memory, and no redrawing is required. All the processes described above are executed quickly and smoothly.

When the program begins execution, it requests the filename of the model to be viewed and the following control information: number of nodes, number of elements, and number of iterations in the analysis. Two kinds of display styles may be used: wireframe and boundary. The wireframe display style shows each of the elements of the model in their entirety, while the boundary style shows just the outline of the model.

Wireframe display is used to see how the elements are changing, and the boundary style shows a less cluttered outline of the actual part.

When ANIMATE has finished processing all the input data, the original model is drawn on the screen and input devices are set up for use. Using these devices, the user may rotate the model about the x, y, and z axes; move forward or backward through the optimization process; zoom in on desired sections; and move the model to different locations.

Chapter 5

SOFTWARE THEORY AND DEVELOPMENT

This section explains the structures and theories used in the development of the programs CONVERT, STOPFEP, and ANIMATE. All program listings, along with detailed descriptions of program executions and subroutine lists, are found in [6].

Recall from the previous chapter that CONVERT is used to add expansion phase information and write out the input files necessary for the program STOPFEP. STOPFEP performs the finite element analyses needed to alter and structurally optimize the given model. ANIMATE is then used to graphically display the results of the optimization.

5.1 CONVERT

Two of the main features of CONVERT are the input of nodal kinematic constraints for the expansion phase and constraint planes for the static and expansion cases. Expansion phase kinematic constraints are added by entering the node number, and the x, y, and z constraints. The expansion phase kinematic constraints are completely different from those of the static phase. Nodal d.o.f. may be fixed, left free, or released. A nodal d.o.f. may be released if it was held fixed in the static analysis. A '1' constrains a d.o.f., a '0' leaves it alone, and a '-1' releases a d.o.f. constrained in the static phase.

Constraint planes for either phase are created by entering four outer points of a desired plane. Only three points are necessary to define a plane, however such a plane would extend infinitely. At times a model will have a region on a plane that does not need to be completely constrained. The four points the user inputs define a region as well as a plane. Only nodes within the defined region that are on the defined plane will be added to the constraint plane list. After the first three points define the given plane, the fourth point is tested to ensure it is on the plane. If the point passes the test, the algorithm proceeds. If it does not, the user is instructed to reenter all four points.

The program will use these boundary points to define a plane equation. All the nodes in the model are then scanned and their coordinates input into the defined plane equation. If a node causes the result of the equation to be zero, the node is on the plane, and is inserted into that plane's list of nodes. This procedure is repeated for each defined constraint plane. The end result is a list of nodes for each constraint plane.

The maximum and minimum x, y, and z coordinates of the four points entered define a box in space containing the nodes on the desired constraint plane, and other nodes in the nearby region. Only the nodes in the box are tested for the plane.

The plane equations are formed below, where the subscripted coordinates are those of the first three points used to define the plane. The unsubscripted coordinates are the coordinates of each node being checked.

$$\begin{aligned} \delta &= c1\,(x - x_1) + c2\,(y - y_1) + c3\,(z - z_1) \\ &\text{where} \\ c1 &= (y_2 - y_1)(z_3 - z_1) - (y_3 - y_1)(z_2 - z_1) \\ c2 &= (x_3 - x_1)(z_2 - z_1) - (x_2 - x_1)(z_3 - z_1) \\ c3 &= (x_2 - x_1)(y_3 - y_1) - (x_3 - x_1)(y_2 - y_1) \end{aligned} \qquad [1]$$

The value of the variable delta is used to determine if a node lies on the plane. Ideally, if a node is on the defined plane, delta will be zero. Due to computer inaccuracy, this will not identify all the nodes. Therefore the program variable epsilon is used to measure the smallness of delta. Any node with a delta less than epsilon is on the defined plane. By trial and error, an epsilon of two times the lowest c value (c1, c2, and c3 defined above), was found to give accurate results.

Another function of CONVERT is to test element connectivity, since I-DEAS sometimes incorrectly defines elements. Incorrectly defined elements cause negative element volumes, which will cause a finite element solver to fail. CONVERT tests each element's definition as it reads it in. The conventional definition for an eight noded, three dimensional element begins by numbering the first four nodes of the bottom face in a counter clockwise direction. By the right hand rule, the bottom face should point up toward the top face, which is also numbered in a counter clockwise direction using the last four nodes.

The program defines three vectors: vector A from node one to node two, vector B from node one to node four, and vector D from node one to node five. Vector C is formed by crossing A into B, and should point in approximately the same direction as vector D. A picture of these vectors is in figure Figure 1.

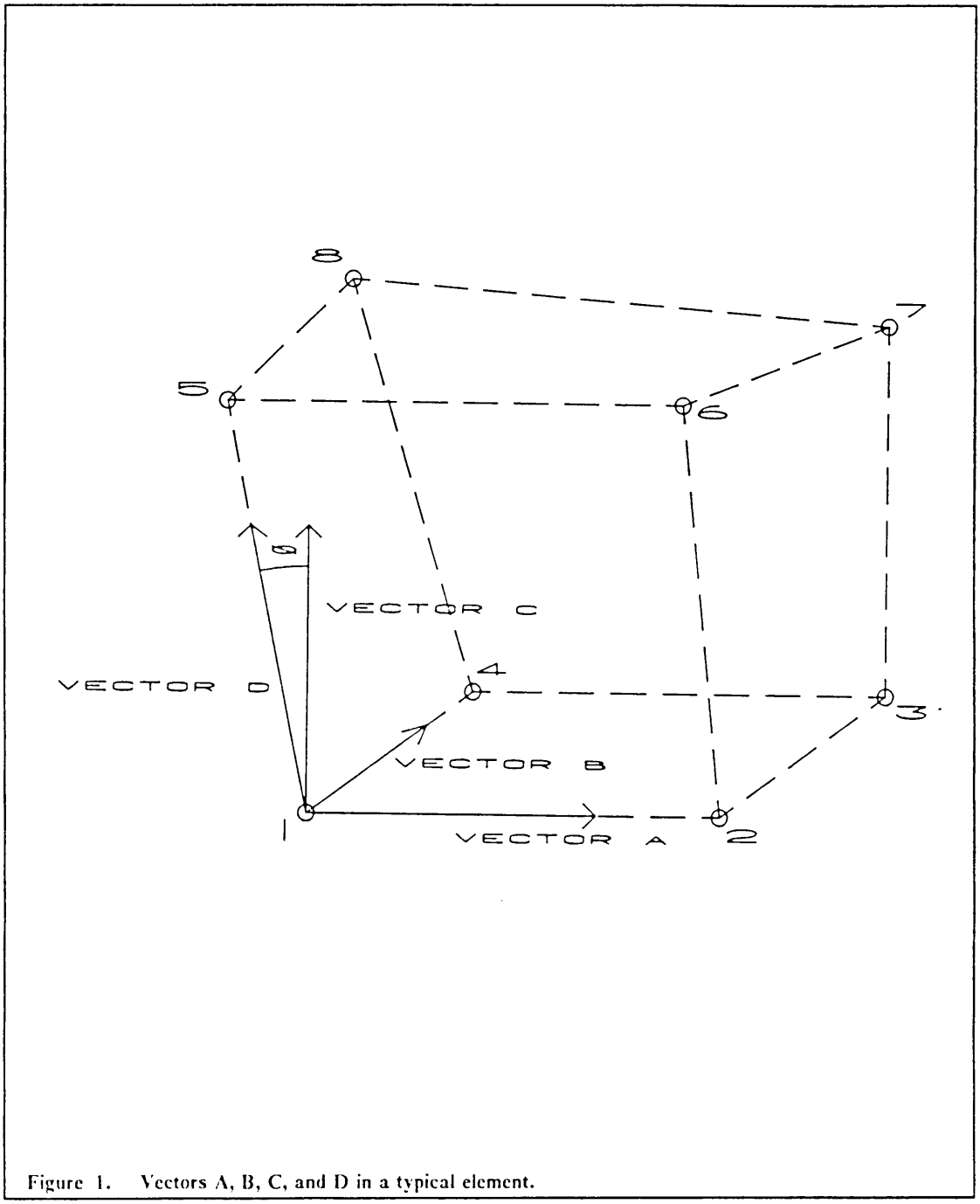


Figure 1. Vectors A, B, C, and D in a typical element.

The angle, θ, between the vectors C and D is:

$$\theta = \cos^{-1}\left[\frac{\overline{C} \cdot \overline{D}}{|C|\,|D|}\right] \qquad [2]$$

If this angle is small, the faces are correctly defined. If the angle is approaching 90 degrees, the element is badly distorted; and if it is greater than 90 degrees, the vectors are pointing in opposite directions and the element faces are incorrectly defined. To correct this, the faces are switched by interchanging the first four nodes of the element with the last four.

5.2 STOPFEP

STOPFEP is made up of a standard finite element analysis program coupled with shape modification routines. Most of the shape modification information is calculated using the routines of the finite element solver. The solver was modified to handle more than one load case.

The program was originally written by C.E. Knight and I.R. Grosse, and it was debugged and enhanced by S.A. Hambric. The key modifications made here include: implementing multiple load case capability, which allows the model to be optimized based on different types of in service loading; debugging the computer coding for the constraint plane application to load vectors and displacement vectors; optimizing the code; and making the program user friendly.

As mentioned before, the program uses a two part analysis. The first part is a static analysis, which performs a finite element analysis of the model with applied loads and constraints. Stresses are calculated for each load vector, and the maximum element strain energy densities are selected from each load case to form the expansion load vector.

The second part of the analysis is the expansion phase. The maximum strain energy densities for each element, the reference stress, and the expansion coefficient, are used to calculate an initial strain vector for each element. Using these initial strains, the expansion load vector is formed. This load vector and the expansion phase stiffness matrix are used to calculate a new set of displacements. These are the displacements used to modify the model and are added to the current nodal coordinates.

The program checks for convergence and continues iterating if the convergence tolerance is not met.

The following algorithm is made up of a conventional finite element formulation with added constraint plane capability. Much of the algorithm was implemented by I. R. Grosse. A basic knowledge of finite elements is assumed. The text by Cook [7] is an excellent reference source.

The stiffness matrices are formed first, one for the static phase using the static case constraint set; and one for the expansion phase using the expansion case constraints. The matrices are constructed on an element basis and assembled to get the global stiffness matrices.

Nodal constraints on global coordinate d.o.f. are implemented by removing corresponding equations, but constraint planes involve matrix manipulations to couple the appropriate d.o.f.. A constraint matrix must be formed for each element and used to alter the element's stiffness matrix before it is added to the global matrix. The constraint matrices are also used to modify all load vectors and calculated displacement vectors.

To form an element constraint matrix, the following operations are performed. The first three nodes are taken from each constraint plane node list and are used to define the plane. The planes are implemented by eliminating the nodal z d.o.f. in favor of the x and y d.o.f.. Plane equations are:

$$(z - z_1) = \frac{-c1}{c3}(x - x_1) - \frac{c2}{c3}(y - y_1)$$

where

$$\begin{aligned} c1 &= (y_2 - y_1)(z_3 - z_1) - (y_3 - y_1)(z_2 - z_1) \\ c2 &= (x_3 - x_1)(z_2 - z_1) - (x_2 - x_1)(z_3 - z_1) \\ c3 &= (x_2 - x_1)(y_3 - y_1) - (x_3 - x_1)(y_2 - y_1) \end{aligned} \qquad [3]$$

The variables alpha and beta will become entries in the element constraint matrix, and are calculated as:

$$\alpha = \frac{-c1}{c3}$$
$$\beta = \frac{-c2}{c3} \qquad [4]$$

Where c1, c2, and c3 are calculated in equation 3.

During the element stiffness matrix formulation, each node in the element is looped over to determine if it is a constraint plane node. If it is, the alpha and beta calculated in equation 4 will be put in the element constraint matrix. The constraint matrix is an n x n identity matrix with n equal to the total d.o.f. of the element with the following changes: the column corresponding to the node's z d.o.f. is zeroed out; in the row corresponding to the node's z d.o.f., the value corresponding to the node's x d.o.f. is set equal to alpha, and the value corresponding to the node's y d.o.f. is set equal to beta. This is done for each node of the element that is on a constraint plane.

Stiffness matrices are altered on an element basis by:

$$[\overline{K}^e] = [C^e]^T [K^e] [C^e]$$
where
$[K^e]$ = the element stiffness matrix
$[C^e]$ = the element constraint matrix
$[\overline{K}^e]$ = the transformed element stiffness matrix [5]

This is done for both the static and expansion phase matrices using the constraint planes defined for each phase. These calculations amount to a coordinate transformation from global x, y, z coordinates to in-plane x', y', and z' coordinates. Due to the zeroes in the constraint matrices, the modified stiffness matrix will not be positive definite. This problem is solved by putting dummy positive numbers in the diagonals that ended up being zero. These substitutions will cause incorrect displacement calculations

for the corresponding z′ d.o.f., but since they were constrained out in favor of x′ and y′ d.o.f., they are set to zero anyway after the displacements are calculated.

After both stiffness matrices are formed and modified, the load vectors for each load case are formed. Nodal loads are added to element face pressure contributions to form element load vectors. Constraint planes transformations are added by:

$$\{\bar{r}^e\} = [C^e]^T \{r^e\}$$

where

$\{r^e\}$ = the element load vector

$\{\bar{r}^e\}$ = the transformed element load vector [6]

The element load vectors are then added into the global load vector as follows:

$$\{\bar{r}\} = \sum_{e=1}^{\text{nume}} \{\bar{r}^e\}$$

where

$\{\bar{r}\}$ = the transformed global load vector [7]

The program can now solve for the static displacements for each load case by the conventional finite element equation:

$$[\bar{K}]\{\bar{d}\} = \{\bar{r}\}$$

where

$[\bar{K}]$ = the transformed global stiffness matrix

$\{\bar{d}\}$ = the transformed displacement vector [8]

Constraint plane transformations are applied to the displacement vector to yield global coordinate components:

$$\{d\} = \{\bar{d}\}[C]$$

where

$\{d\}$ = the final global displacement vector [9]

The element stress state for each set of displacements is:

$$\{\sigma^e\} = [E^e][B^e]\{d^e\}$$

where

$\{\sigma^e\}$ = the element stress vector

$[E^e]$ = the element material stiffness matrix

$[B^e]$ = the element strain displacement matrix

[10]

Element stresses are used to calculate a strain energy density for each element. When stress calculations are complete, there exists a set of element strain energy densities for each load case. Optimization should be based on the maximum strain energy densities for each element. Therefore each set of densities is scanned, and a single set containing only the maximums is formed.

The user defined reference stress is then converted to the reference strain energy density. This is subtracted from the set of maximum calculated strain energy densities. Multiplication by the expansion coefficient alpha determines the magnitude of the initial strains in the element:

$$\{\varepsilon_o^e\} = \alpha(\varphi^e - \varphi_r)[1 \;\; 1 \;\; 1 \;\; 0 \;\; 0 \;\; 0]^T$$

where

$\{\varepsilon_o^e\}$ = the element initial strain vector

α = the expansion coefficient

φ^e = the element strain energy density

φ_r = the reference strain energy density

[11]

Next, using these strains, an expansion load vector is calculated. This is done on an element basis, and summed to form a global expansion load vector. Constraint plane transformations are applied as they were to the load vectors in the static phase.

$$\{r^e\} = \int\int\int_{v^e} [B^e]^T [E^e] \{\varepsilon_o^e\}\, dv$$
$$\{r\} = \sum_{e=1}^{nume} \{r^e\} \qquad [12]$$
$$\{\bar{r}\} = [C]\{r\}$$

This expansion load vector is used to calculate a new set of displacements, which then have expansion phase constraint plane transformations applied to them. These displacements are transformed to global coordinates and then added to the nodal coordinates to reshape the structure, and the iteration is complete.

The leveling of strain energy density distributions is tested after the stress computations. The convergence tolerance is calculated by using a scaled euclidian norm for iteration i. The equation uses strain energy densities for each element from the current and previous iterations:

$$s^i = \frac{\sqrt{\sum_{e=1}^{nume} (\varphi_a^{e^i} - \varphi_a^{e^{i-1}})^2}}{\sqrt{\sum_{e=1}^{nume} (\varphi_a^{e^i})^2}} \qquad [13]$$

where

s^i = the convergence tolerance
φ = the element strain energy density
nume = the number of elements

If this value is less than the user defined tolerance, the model has converged. If it is greater than for previous iterations, the solution is diverging and the program stops. In the first iteration, an arbitrarily high value is assigned to ensure the program will go through at least two iterations. All subsequent iterations are treated normally.

5.3 *ANIMATE*

The program is used to display the optimization process on screen, and to allow the user to observe different sections of the model by rotation, location, and zoom functions. It is written using the FORTRAN programming language and uses calls to graPHIGS subroutines to perform all graphical functions. Documentation listings for graPHIGS are found in reference [6].

ANIMATE draws a model as it appears in each iteration in separate pictures. These pictures are stored in memory so the user may recall any of them to a graphics terminal screen. Various input devices are programmed to accept user commands at any time to manipulate the model onscreen into any desired position and orientation.

graPHIGS differs from other graphics languages such as GKS in that it stores entire pictures for use at any time. No matter how complex the drawing in a picture is, it may be stored in the intelligent workstation memory and flashed on the screen instantly. The user does not wait for lines and text to be drawn.

For this reason graPHIGS is ideal for the animation of the optimization process. Each iteration of the process is put in a different picture. The pictures may then be sequentially displayed to simulate animation.

Pictures are filled up with model drawings for each iteration by reading in element connectivity once and storing the information, then reading in all nodal coordinates for each iteration and drawing the lines that define the models. If a normal wireframe drawing is specified, each element of the model is looped over and lines are drawn from node to node using the coordinates of the nodes that define an element. The result is a series of drawn boxes in a picture that make up a drawing of a model. This process is repeated for each iteration of the optimization process.

If a boundary drawing is requested, all elements are looped over; but instead of simply drawing each line of an element, all lines of an element are compared to all the lines of all the remaining elements of the model. If a match is found, the tested line is not on the outer boundary, and is not drawn. If no match exists, the line is drawn in the picture. This testing algorithm is performed only for the first iteration of the optimization, and the nodes defining the boundary lines are stored. The stored nodes are used in all subsequent iterations to draw modified model boundaries in the other pictures.

Different model locations and orientations are produced on screen by using the concept of views. Different views may be defined for any picture to examine it in a different way. Views may be moved in any direction, and rotated in any way using transformations. ANIMATE works by defining a view based on user input, and then puts the current picture containing the current iteration of the optimization in that view. Views may be constantly redefined. If for example the user slowly turns a valuator dial to rotate the model about an axis, the appearance onscreen is that of a constantly rotating model.

Once the program completes drawing pictures of each iteration of an optimization process and storing the pictures in the graphics terminal memory, the algorithm sets up input devices and awaits user input. The program is then in a wait and respond mode.

Input is given through input devices attached to the IBM 5080 terminals. The attached input devices used here are the Choice board, which uses choice keys to perform the translate and zoom operations; the valuator dials, which are used for rotation about the x, y, and z axes; the locator and tablet, which are used to define zoom boxes and new locations on the screen; and finally string input on the keyboard.

These inputs may be programmed in one of three modes: Request mode, where data may only be entered when the program asks for it; Sample mode, where the program

samples data from an input device without user interaction; and Event mode, when the program waits for user input and responds to it.

Event mode is the type used most in ANIMATE. This mode allows the user to perform any given action at any time. When the user requests certain actions, such as zoom or locate, the keyboard and locator are put in request mode to complete the operation.

When input is made, it is to redefine the current view of the model, or to bring up the previous or following picture of the optimization. Views are redefined by moving the observation point, expanding or shrinking the observation window, or multiplying a rotation matrix by the window and observation point defining the view. Once a new view is calculated, the current picture is placed in it and the operation is complete. To show the next picture of an optimization, the current picture is removed from the current view, and the next picture placed in that view. Done repeatedly, this will produce an animation of the entire optimization process with a model in any specified view.

Chapter 6

CASE STUDIES

Three types of structures were studied to evaluate this optimization approach. Where appropriate, results are compared with reference studies. Results are also compared to theoretical or approximate solutions. The following models were analyzed:

- **Pressure Vessel**: a spherical model and a cylindrical model were analyzed. Both were given nonuniform thicknesses and were expected to converge to uniform thickness models. Results are compared to Lame's solutions and to a similar analysis by Oda and Yamazaki [2].

- **Torque Arm**: a two dimensional and a three dimensional optimization were performed. The two dimensional results were compared to those of Bennet and Botkin [4]. Stress calculations were verified using a beam approximation.

- **Draft Sill Casting**: a multiple load case model of a Norfolk Southern design of a draft sill casting is optimized. The model was originally created and analyzed by Roach [5] in 1986. Results are compared to previous finite element analyses.

The pressure vessel models were used as an initial program test. The models were simple, and the solution was known. The torque arm model was selected to test the program's ability to handle more complicated models, first in a two dimensional analysis,

then in a three dimensional one. After getting satisfactory results from these analyses, a complex model of a draftsill casting would test the program's multiple load case capability.

6.1 Method of Presentation

The following method of presentation is used for each case study. First a description of each model is given, along with accompanying pictures. All constraints and load cases are described, along with material properties and optimization control variables. In a results section, the initial and final stress states are described. Pictures of the final model are shown, along with a plot of how model volume changed from iteration to iteration. Computer CPU times for a static analysis, a single iteration, and the entire optimization are compiled. At the end of each section is a discussion of results, describing how the model performed and comparing results to theory and referenced work.

6.2 *Pressure Vessels*

Two models were created, both quarter section slices of pressure vessels. Only quarter section slices are needed due to vessel symmetry. The first model was from a cylindrical vessel, and the second from a spherical vessel. Both models were created with varying thicknesses and are shown in Figure 2 and Figure 3. An engineering drawing of both models is in Appendix C. The goal of the analysis was to verify that a uniform thickness for each model is obtained. Although the solution to the problem is obvious, the models are a good means of testing the program since the solution is known.

Steel was used as the material, with a Young's Modulus of 206.8 GPa, and an assumed yield strength of about 180 Mpa. Results will be compared to those reported by Oda and Yamazaki [2].

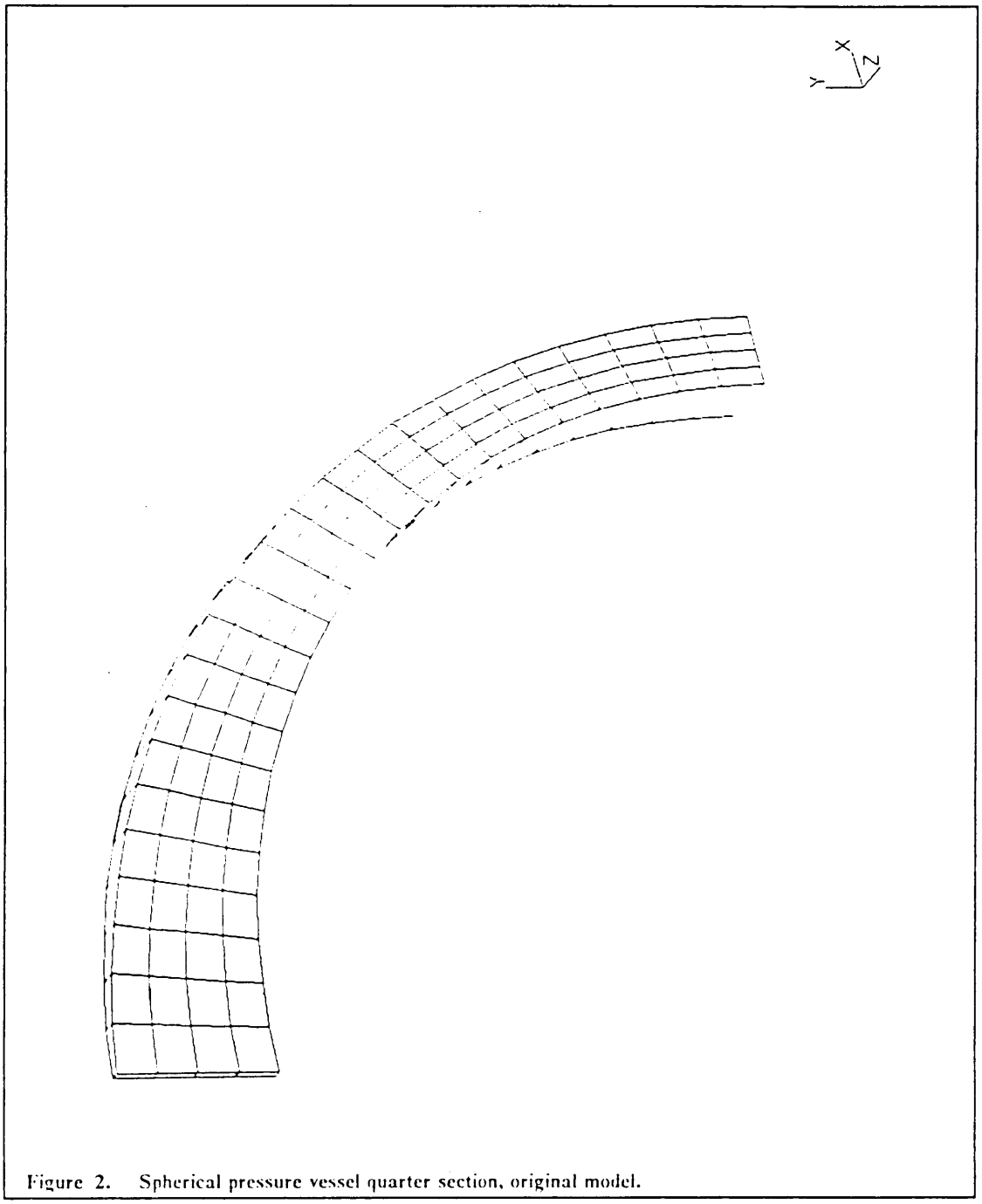


Figure 2. Spherical pressure vessel quarter section, original model.

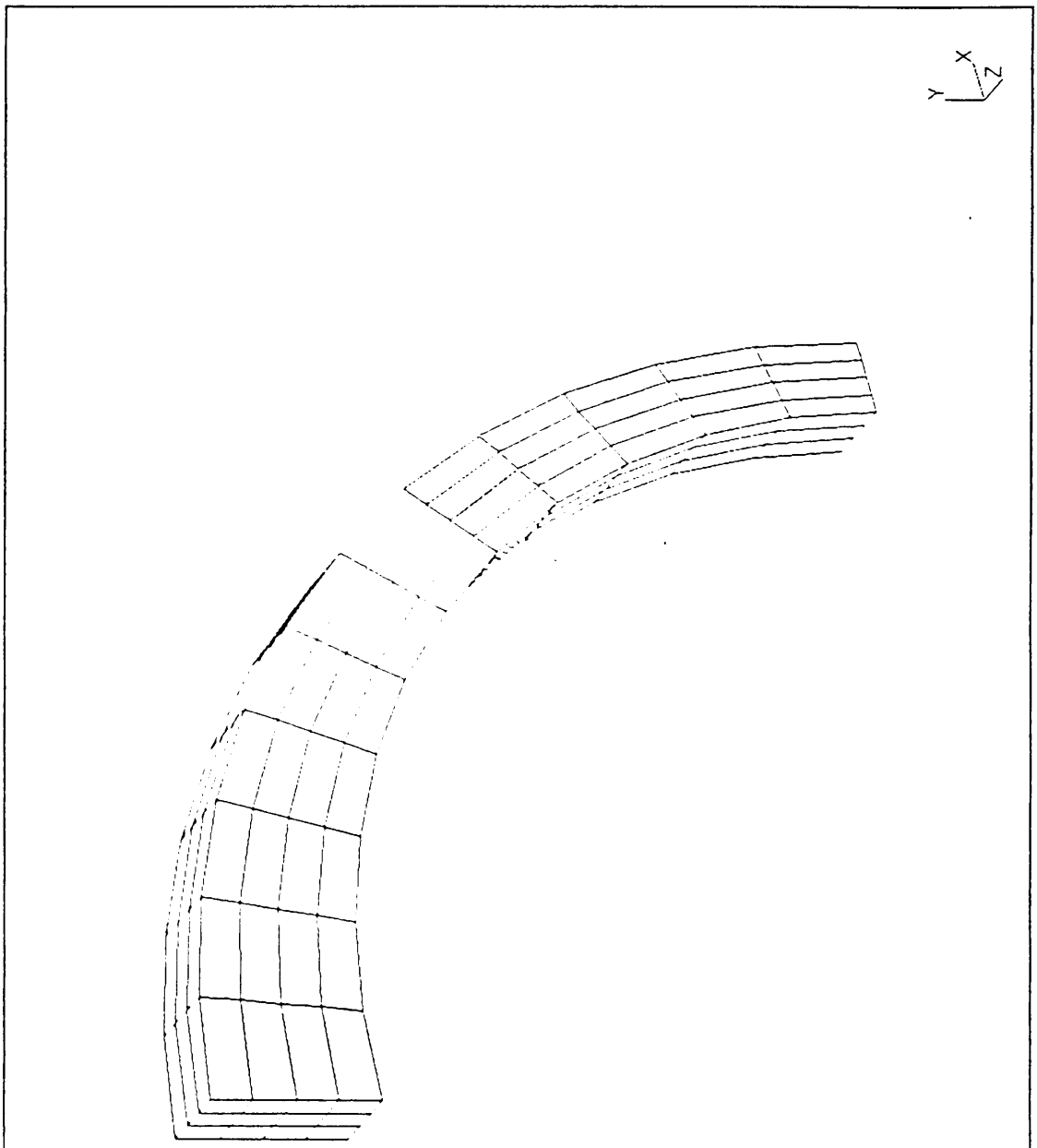


Figure 3. Cylindrical pressure vessel quarter section, original model.

6.2.1 Spherical Vessel

6.2.1.1 Model Description

The model was one element deep, with four elements in the radial direction and 24 elements in the angular direction. The vessel had a uniform inner radius of 100 mm and an outer radius varying from 110 mm to 125 mm. The following symmetry constraints were applied to simulate the rest of the pressure vessel. The top and bottom faces were held fixed in the tangential direction by nodal d.o.f. constraints. The back face is flush with the xy plane and was constrained by eliminating all z d.o.f. on that face. The front face is not on any primary global coordinate plane, and was constrained using a constraint plane. One load case was applied to simulate a constant internal pressure, and used element face pressures of 1.0 Mpa applied to all elements on the inner face. The model was given an expansion coefficient of 5.0×10^{-6}, a reference stress of 2.5 Mpa, well below the assumed yield strength of 180 Mpa; and a convergence tolerance of 0.05. A model approximation exists at the sphere apex, where the slice should converge to a line rather than a face. Extremely poor stress results occured when hexahedron elements were degenerated to tetrahedrons. Therefore, the model actually has a small polar hole.

6.2.1.2 Stress States and Other Results

Stresses were largest on the inside of the vessel and decreased to a minimum at the outside, which is predicted by Lame' (see Appendix A). This distribution was greatest at the bottom, thinner end of the quarter section; and smallest at the top, thicker end; which was also expected. The thinner the vessel wall is, the higher the stresses will be.

The maximum Von Mises stress was 6.36 Mpa, located in the inner element of the bottom face. The minimum Von Mises stress was 1.55 Mpa, located in the outer element of the top face. These values give a maximum/minimum ratio of 4.10. Also note the average stress is higher than the reference stress of 2.5 Mpa, so there will be more elements expanding than contracting in the optimization, causing an increase in material volume.

In the converged model, the stress distribution still showed larger stresses on the inside and lower stresses on the outside. This time however, the distribution did not vary much throughout the quarter section since the thicknesses were almost uniform throughout.

Comparing maximum and minimum stresses showed that the maximum Von Mises stress was again in the inner element of the bottom face, but was lowered to 3.36 Mpa. The minimum Von Mises stress was also in the same location as before and equal to 1.96 Mpa, which gave a final maximum to minimum ratio of 1.71.

The model converged in eight iterations. Figure 4 shows a graph of model volume vs. number of iterations. The graph shows an increasing volume, which may seem contrary to the basic goal of the program. The goal in this analysis however was to achieve a near uniform thickness model, a goal that was achieved. The graph shows a convergence of model volume to a value of about 25 cubic centimeters.

The CPU time for a single static analysis was 1.20 seconds, 2.37 seconds for a single iteration, and 19.0 seconds for the complete optimization. These values indicate the entire optimization process required about 16 times the computer time a single static analysis required. A picture of the final model is shown in Figure 5.

The figure shows near uniform wall thicknesses, with a top thickness of 0.0163 meters and a bottom thickness of 0.0106 meters. The initial top to bottom thickness ratio was 2.5, with the final ratio equal to 1.5.

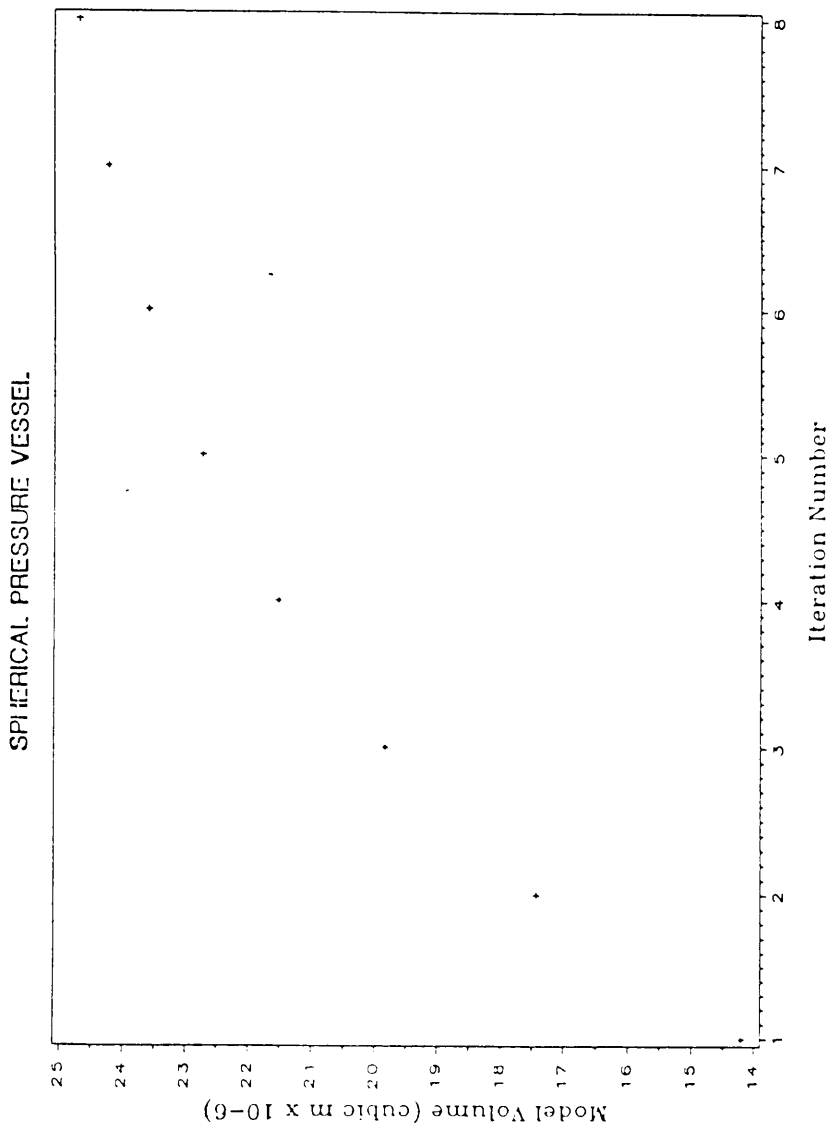


Figure 4. Model volume vs. number of iterations for spherical pressure vessel model.

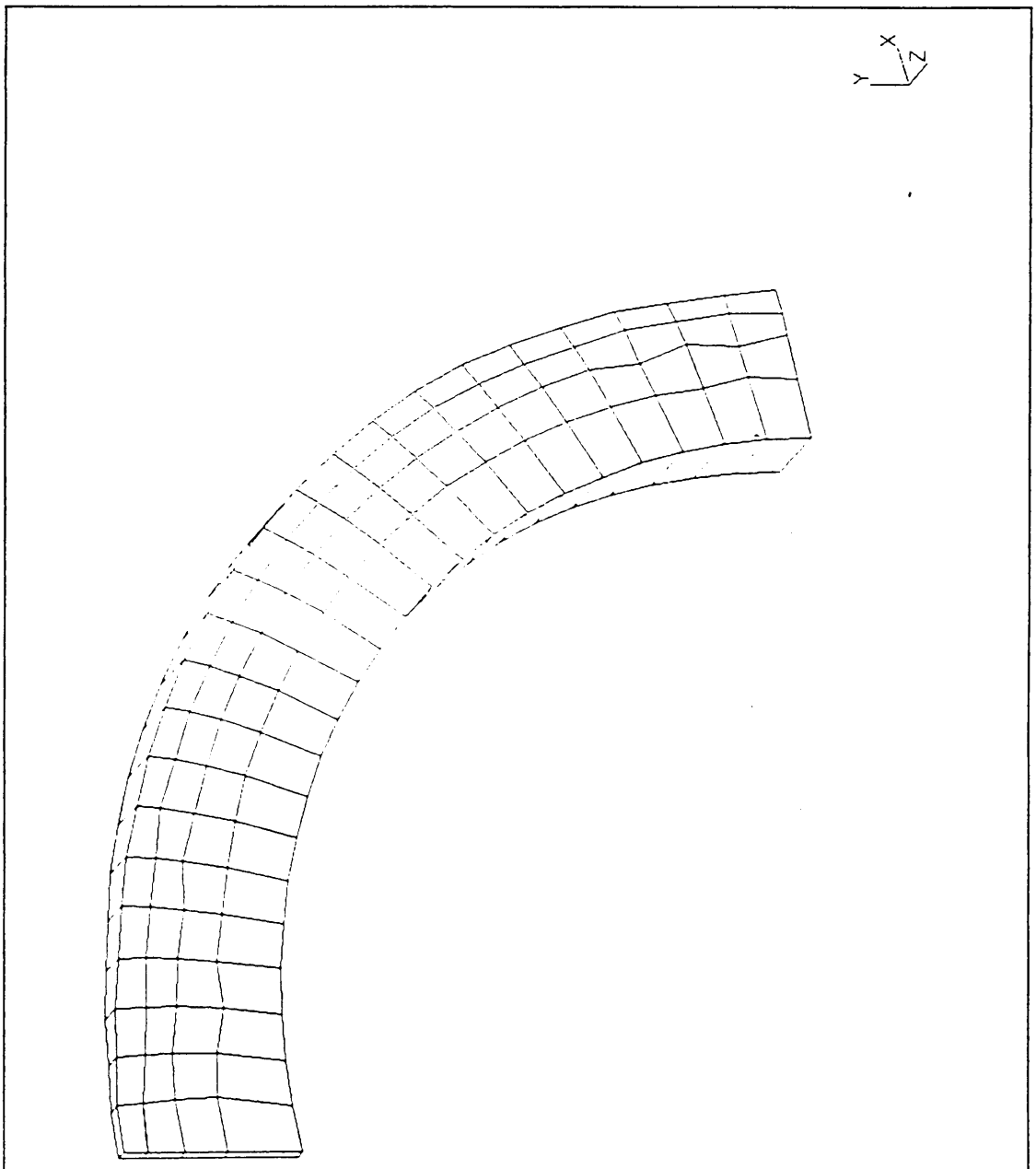


Figure 5. Spherical pressure vessel quarter section, final model.

6.2.2 Cylindrical Pressure Vessel

6.2.2.1 Model Description

The model was three elements deep with four elements in the radial direction and 12 in the angular direction. Radial dimensions are the same as in the spherical model. Constraints were applied to simulate the surrounding pressure vessel. The top and bottom faces were fixed in the tangential direction by nodal d.o.f. constraints as were the front and back face. Again one load case was applied to simulate a constant internal pressure, with element face pressures of 1.0 MPa applied to the entire inner face. The model was given an expansion coefficient of 5.0 x10 ^{6}, a reference stress of 2.5 Mpa, and a convergence tolerance of 0.05; the same control variables given to the spherical model.

6.2.2.2 Stress States and Other Results

Stresses varied from a maximum on the inside to a minimum on the outside, with stresses decreasing as the vessel thickness increased; which was expected. There was no stress variation in the z direction through the thickness. The maximum Von Mises stress was 12.4 Mpa located in the inner elements of the bottom face, with the minimum Von Mises stress equal to 0.35 Mpa located in the outer elements of the top face. This gives a max/min ratio of 35.0.

In the final optimized model the maximum Von Mises stresses were again in the inner elements of the bottom face, and were lowered to 3.68 Mpa. The minimum Von Mises stresses were also in the same locations as before and were raised to 1.77 Mpa. The max/min ratio was lowered to 2.08.

The model converged in 12 iterations. Figure 6 shows a graph of model volume vs. number of iterations. The model volume increases, as it did in the spherical vessel analysis, converging to about 60 cubic centimeters. The volume increase is acceptable since the goal of the analysis was to obtain a near uniform vessel thickness. The CPU time for a single static analysis was 1.67 seconds, 3.33 seconds for a single iteration, and 40.0 seconds for the complete optimization. The complete optimization required about 24 times the CPU time as a single static analysis. A picture of the final model is shown in Figure 7.

Measuring vessel thicknesses shows a top wall thickness of 0.036 meters and a bottom thickness of 0.033 meters. The final thickness ratio was 1.09, a large improvement over the initial value of 2.5.

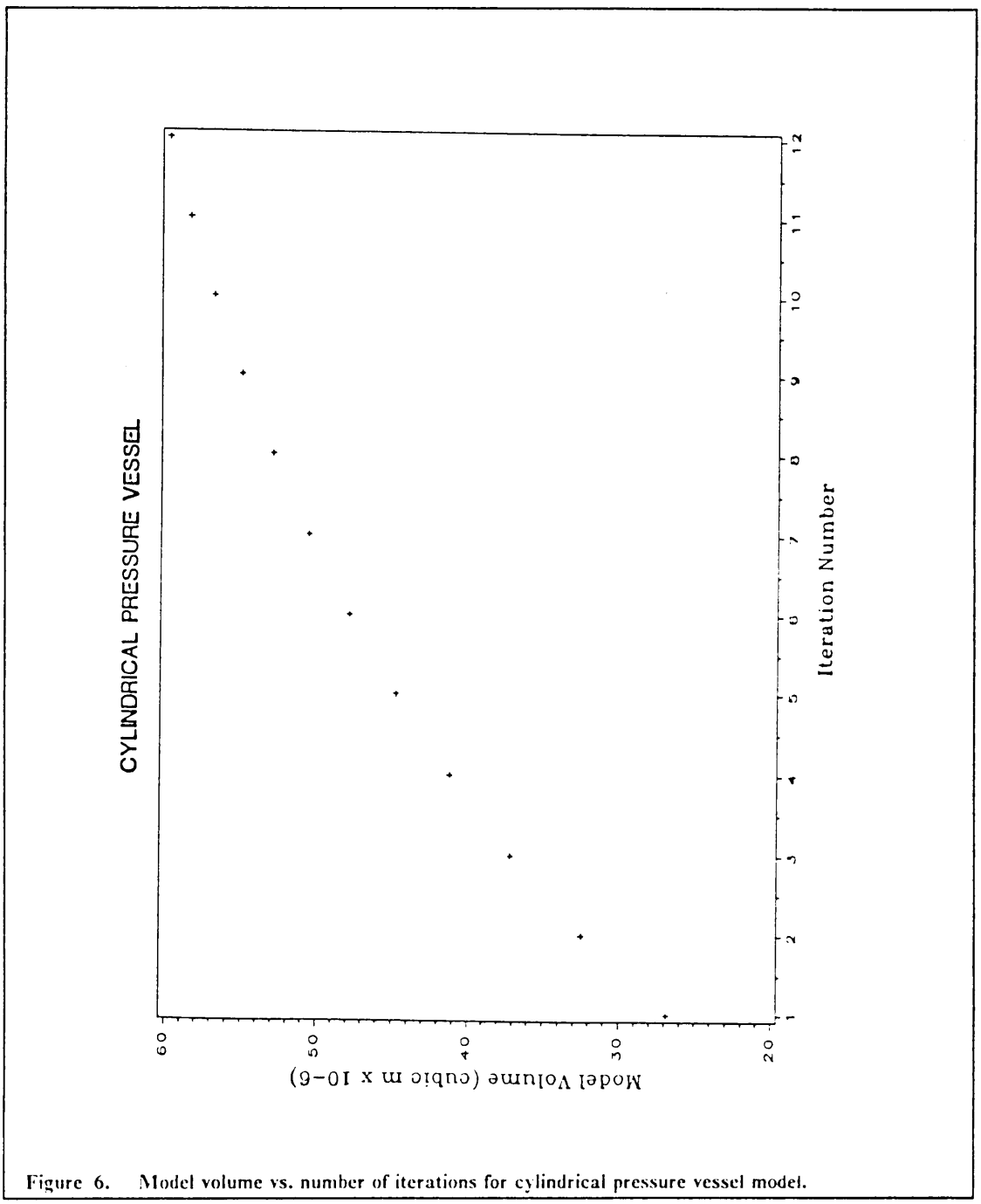


Figure 6. Model volume vs. number of iterations for cylindrical pressure vessel model.

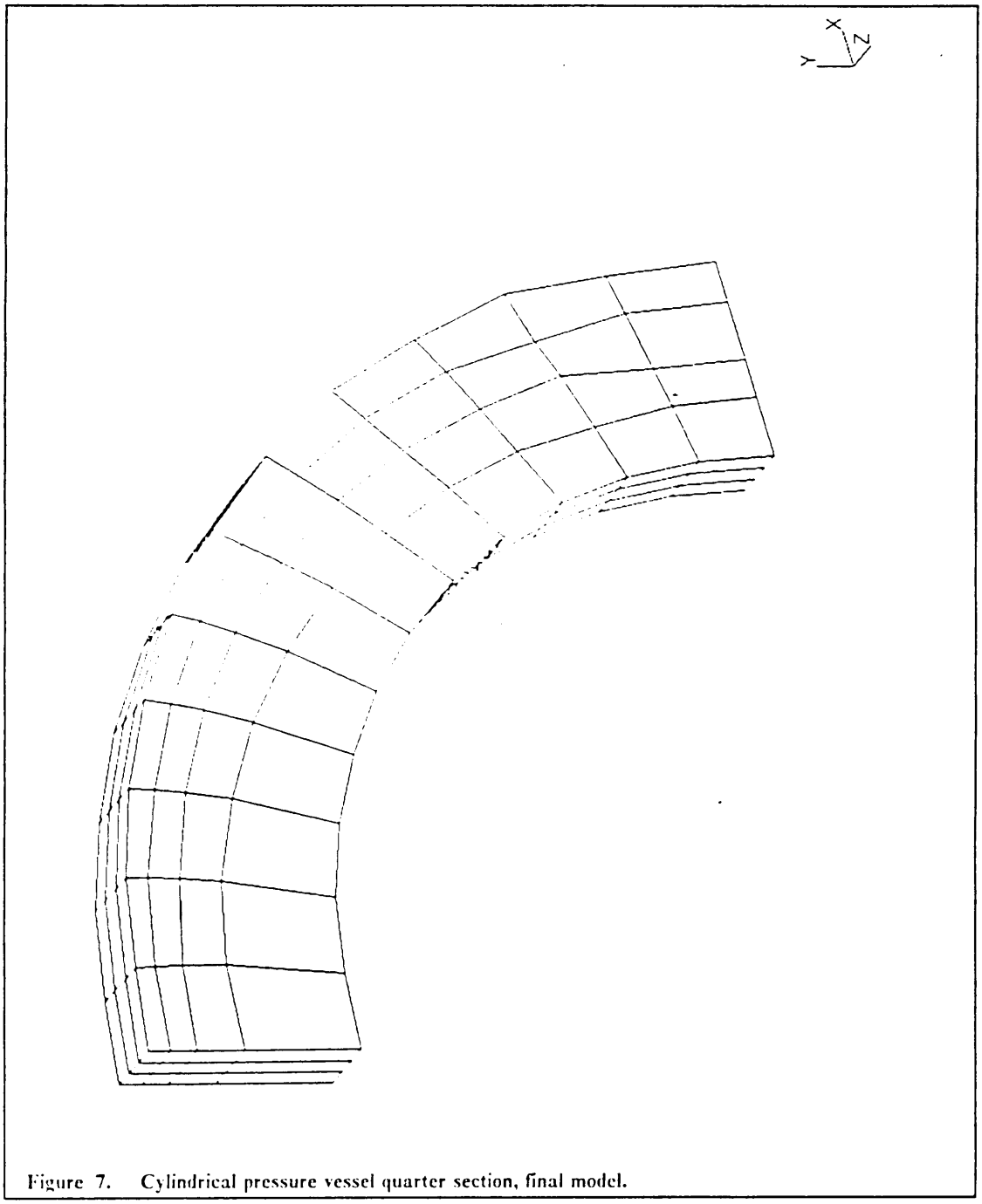


Figure 7. Cylindrical pressure vessel quarter section, final model.

6.2.3 Discussion

Both models moved out to near uniform thickness quickly (first four iterations for the spherical case and first nine iterations for the cylindrical case), and spent the last few iterations trying to converge about the reference stress. This was a futile task considering the stress distribution in a pressure vessel, with stresses ranging from a maximum at the inner face, to a minimum at the outer face. The program attempted to lower the max/min stress ratios by expanding the inner elements and shrinking the outer ones, which decreases the accuracy of the analysis. This caused the centroids of the elements to be closer together near the center of the wall. Since stresses are calculated at Gauss points and averaged at the center of the element, this caused the Von Mises stresses to appear to converge.

The cylindrical vessel's final wall thickness ratio (1.09) converged better than the spherical vessel's ratio (1.50). This discrepancy is due to the elements at the top of the spherical vessel model. These elements have very poor aspect ratios, with the thicknesses in the z direction almost ten times less than any other dimension. Stresses calculated in these elements are a little high, causing them to expand too much in the expansion analysis. Recall also the small hole in the top of the model, which will cause a stress concentration at the top elements. The error was not drastic though, as Figure 5shows.

Reference [2]'s axisymmetric model converged in 15 iterations and also ended up with near uniform thickness. However they held model volume constant and based convergence on stress distributions on the inner and outer surfaces of the model. Stress distribution progresses from largest at the bottom to lowest at the top. When the distribution was near constant, their model had converged.

Their model converged in 15 iterations, with model volume remaining almost constant. The final model had an outer radius ratio of 1.007, as compared to a value of 1.02 for the cylindrical vessel model and 1.05 for the spherical vessel model in this study. Therefore the method presented here produced similar results using fewer iterations. Also, in this method the model volume was allowed to vary, and convergence based on the overall model stress state; this is a more general method of optimization.

Stress results for both models were checked using Lame's equations. Calculations are in Appendix A.

6.3 Torque Arm

Using a single model, two analyses were made. The first was a two dimensional analysis which is compared to the results of Bennett and Botkin [4]; the other was a three dimensional analysis, which was not attempted in referenced work, performed to evaluate the effectiveness of the program.

6.3.1 Model Description

A picture of the initial model is in Figure 8. The hole on the left was constrained at all nodes to simulate a weld. The hole on the right was left free during the static analysis, but constrained in all directions during the expansion analysis to preserve hole size. Steel was again the assumed material, with a Young's Modulus of 206.8 GPa, and a yield strength of about 180 Mpa. An engineering drawing of the torque arm with all pertinent dimensions is in Appendix C.

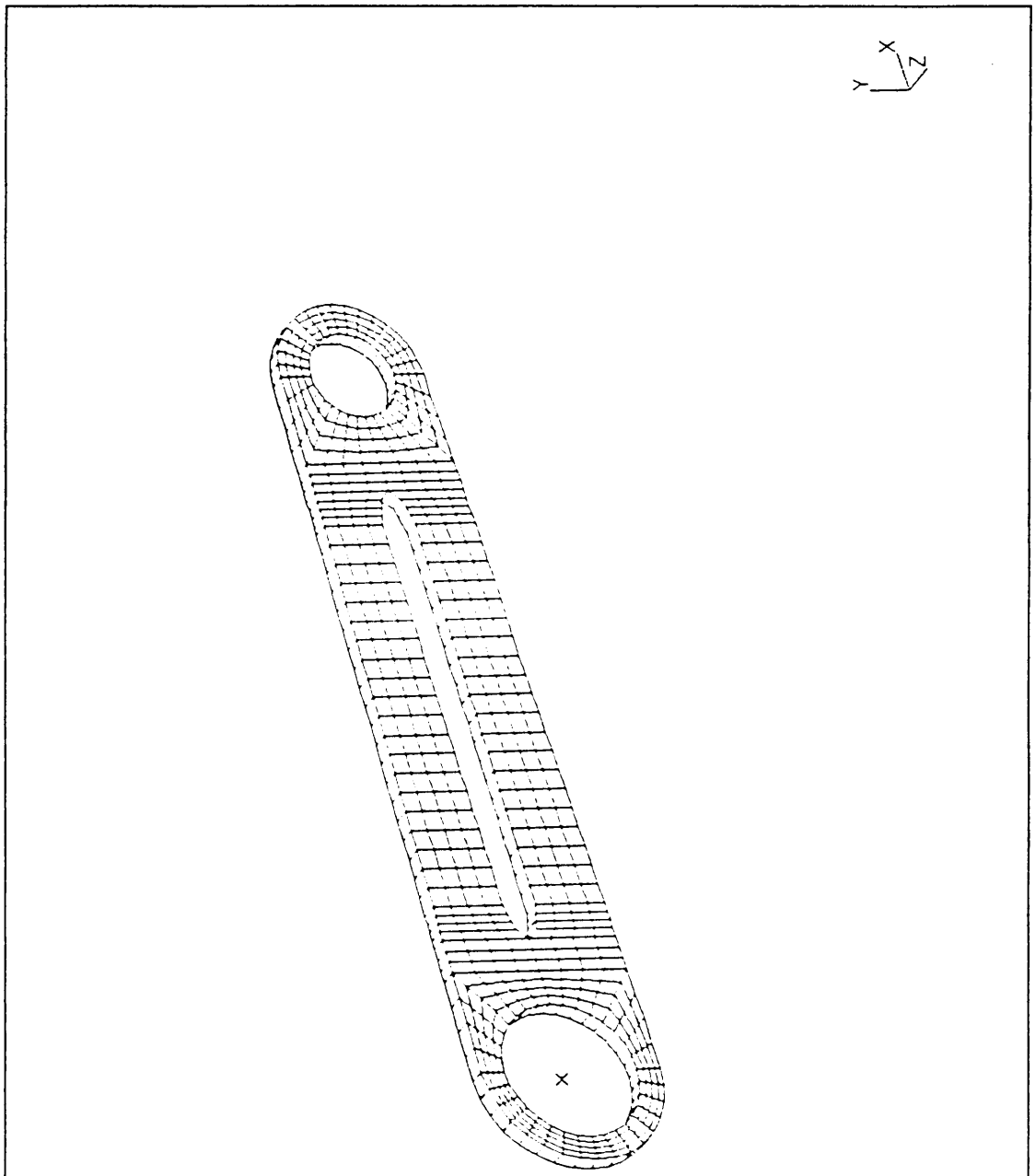


Figure 8. Torque arm, initial model.

The model was one element deep, with 528 elements on the xy plane. A bending force was applied by distributing y direction nodal forces at nodes on the upper half of the right side hole. The equivalent force was 4063 N. An axial load of half the magnitude of the bending load was applied by distributing x direction nodal forces at nodes on the right half of the right side hole.

For the two dimensional simulation, all the nodes of the model were constrained in the z direction for the expansion analysis, which effectively produced two dimensional optimization. For the three dimensional case all nodes were free to move except those at the left and right holes. Control variables for the two dimensional case were: an alpha of 1.0 $x10^{-7}$, a reference stress of 50.0 Mpa, and a convergence tolerance of 0.025. For the three dimensional case, control variables were: an alpha of 0.5 $x10^{-7}$, a reference stress of 50.0 Mpa, and a convergence tolerance of 0.025. These values were chosen based on the model stress state, and trial and error by attempting some short optimizations. The lower alpha value for the three dimensional case was due to the sensitivity of the model (see results).

6.3.2 Stress States and Other Results

The model subsection of interest is the region between the holes on the right and left sides. The material on the outside of the far holes has stresses that are about ten times less than the center material. Therefore maximum and minimum stresses will be taken from this center region.

For both cases the initial stress state was the same. The maximum Von Mises stress was 89.3 Mpa, located at the bottom of the model directly below the left end of the center hole. The minimum Von Mises stress was 3.62 Mpa, located at the lower left side of the far right hole, which is directly opposite to the direction of the resultant load. The maximum minimum stress ratio was 24.7.

The axial stress contribution to the model stresses was fairly constant throughout the middle of the model, with stresses at about 2.5 Mpa. The bending stresses increased going from right to left, which was expected. Bending stresses were also greater at the top and bottom and less near the center hole; ranging from 6.0 Mpa to 83.0 Mpa. Bending stresses were dominant, and the overall distribution was similar to the bending stress distribution. These stress calculations are checked using a beam approximation in Appendix B.

For the two dimensional case, the final maximum and minimum stresses were 66.7 Mpa and 9.46 Mpa respectively; with the same locations as in the initial analysis. This gave a maximum/minimum stress ratio of 7.05, a reduction of 17.65 from the initial ratio. For the three dimensional case, the final maximum stress was 88.0 Mpa and the final minimum stress was 8.28 Mpa. The final ratio was 10.6, a reduction of 14.1 from the initial state.

The two dimensional model converged in 20 iterations and the three dimensional model could only reach seven iterations. For the two dimensional case, the CPU times were: 6.14 seconds for a single static analysis, 12.28 seconds for a single iterations, and 4:05.6 minutes for the entire process. Therefore the entire process used about 40 times the CPU time as a single static analysis did. For the three dimensional analysis, the CPU times were: 6.14 seconds static analysis, 12.58 seconds single iteration, and 1:28.06 minutes total. The entire process required about 14 times the computer time the static analysis did.

Figure 9 shows model volume vs. iterations for the two dimensional model; and Figure 10 shows model volume vs. iterations for the three dimensional case. In the two dimensional case, model volume converges to about 270 cubic centimeters. In the three dimensional case, model volume is not converging enough to make a final volume approximation. The two dimensional model decreased in volume by 24 percent, and the three dimensional model volume decreased by 13 percent.

Pictures of the final two dimensional model are in Figure 11 and Figure 12. Pictures of the final three dimensional analysis model are in Figure 13, and Figure 14.

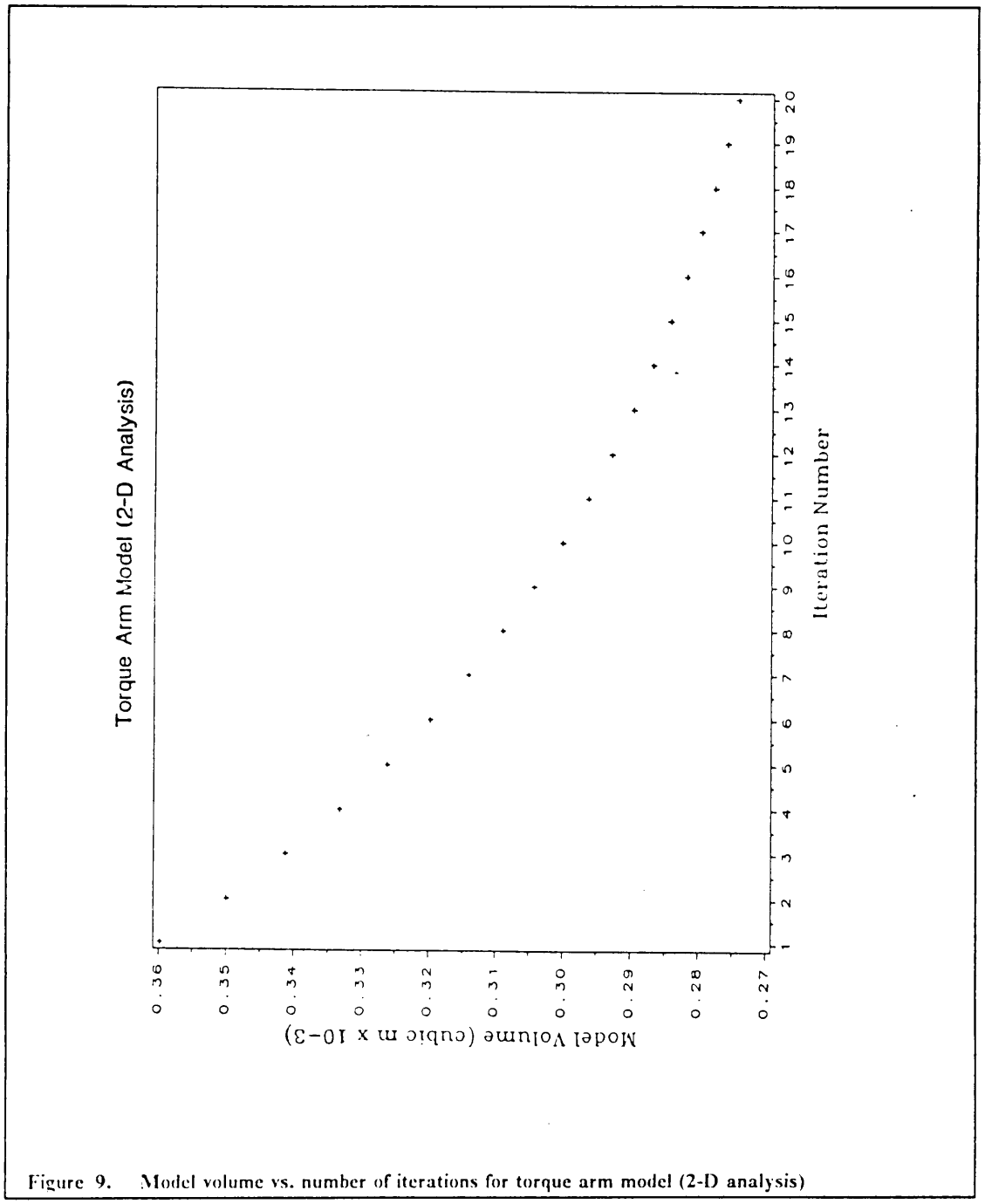


Figure 9. Model volume vs. number of iterations for torque arm model (2-D analysis)

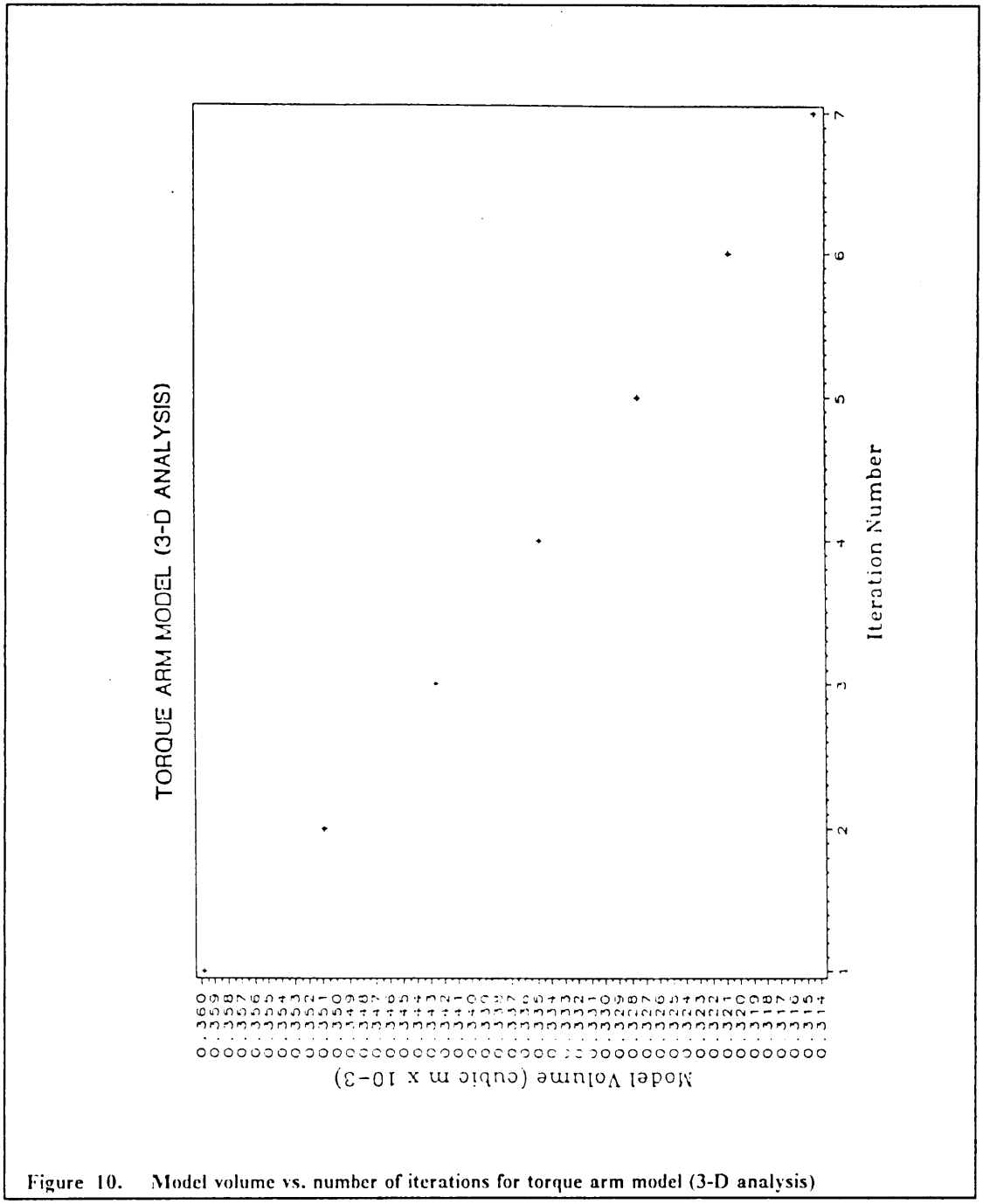


Figure 10. Model volume vs. number of iterations for torque arm model (3-D analysis)

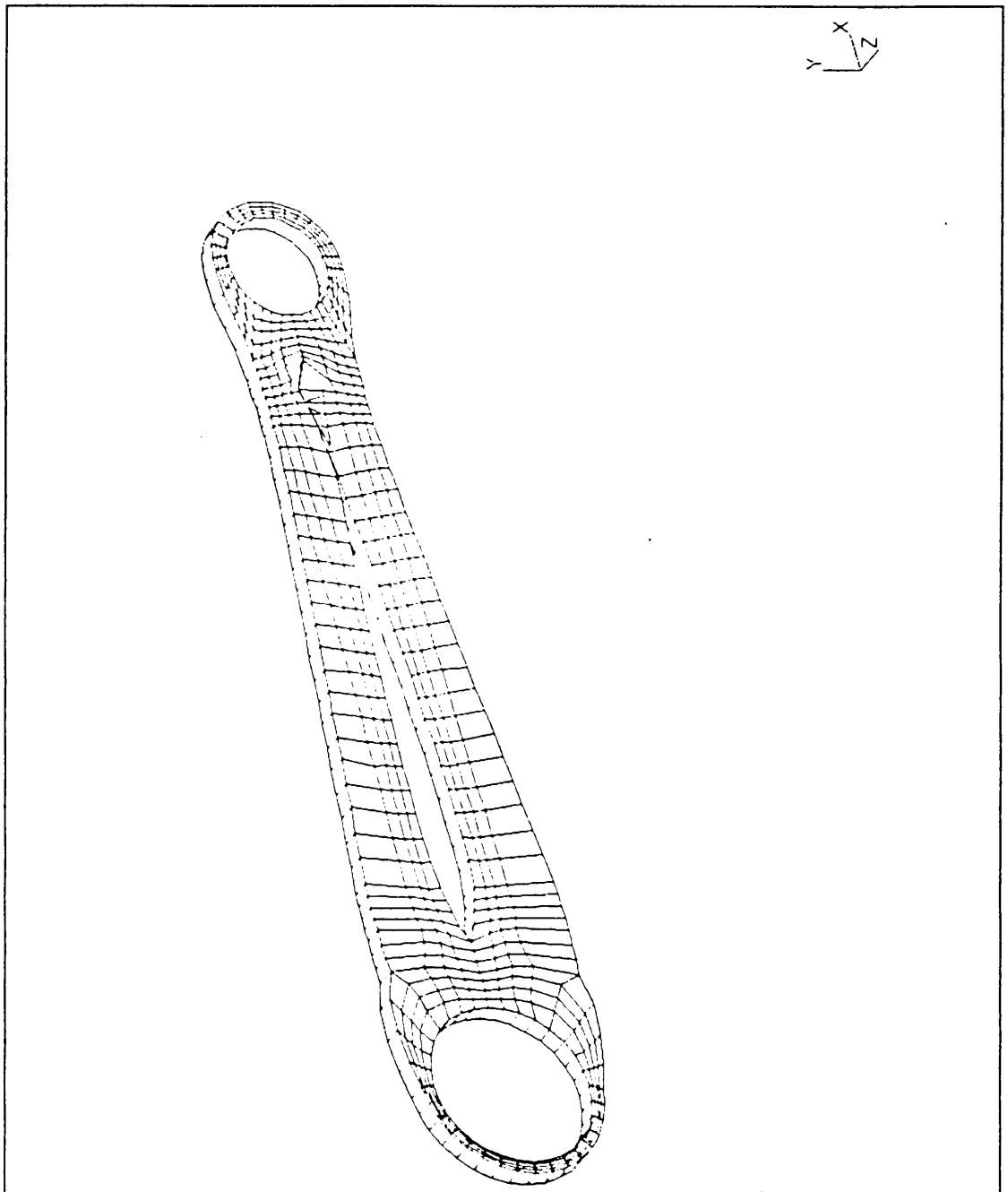


Figure 11. Torque arm, final model (2-D analysis).

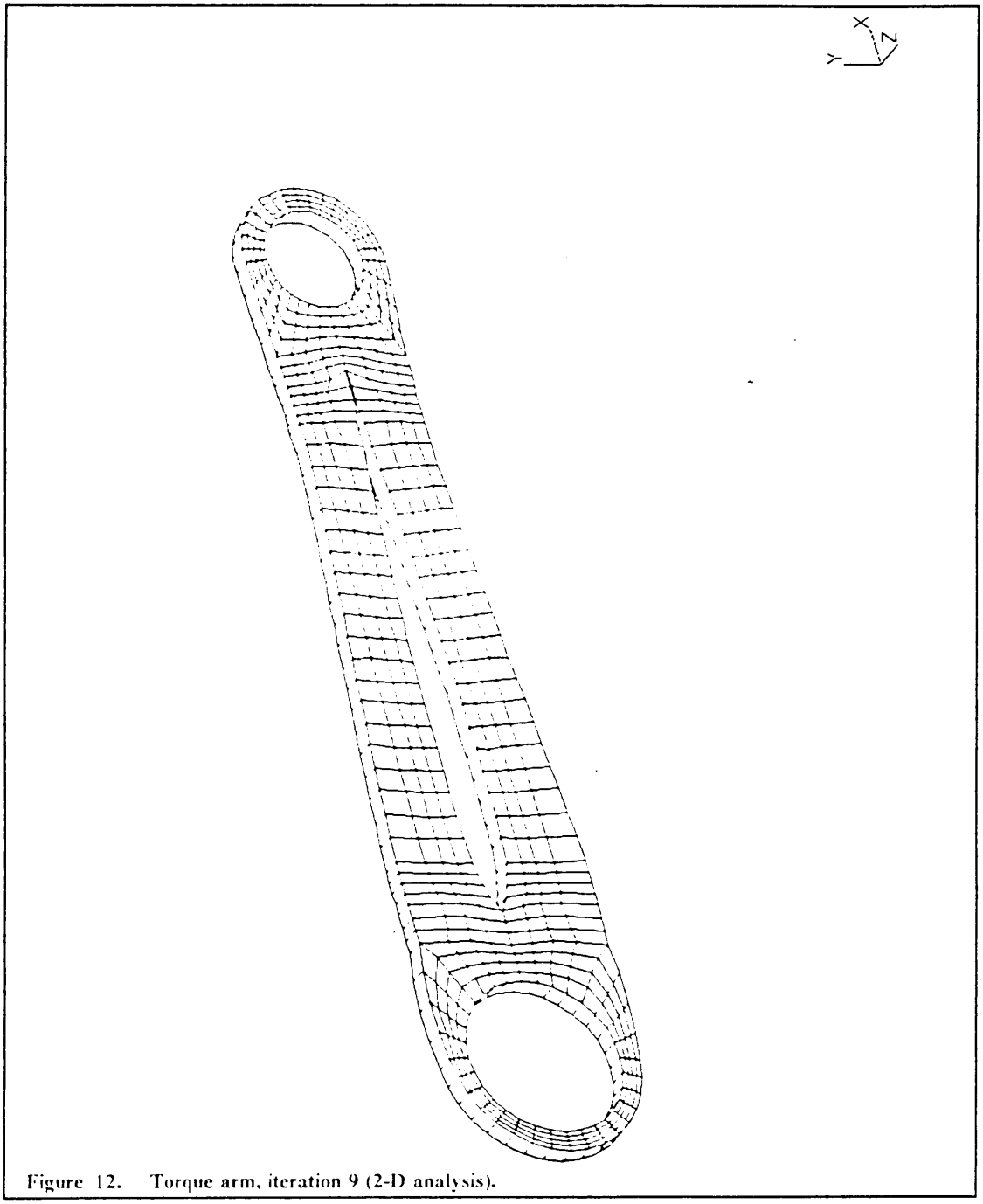


Figure 12. Torque arm, iteration 9 (2-D analysis).

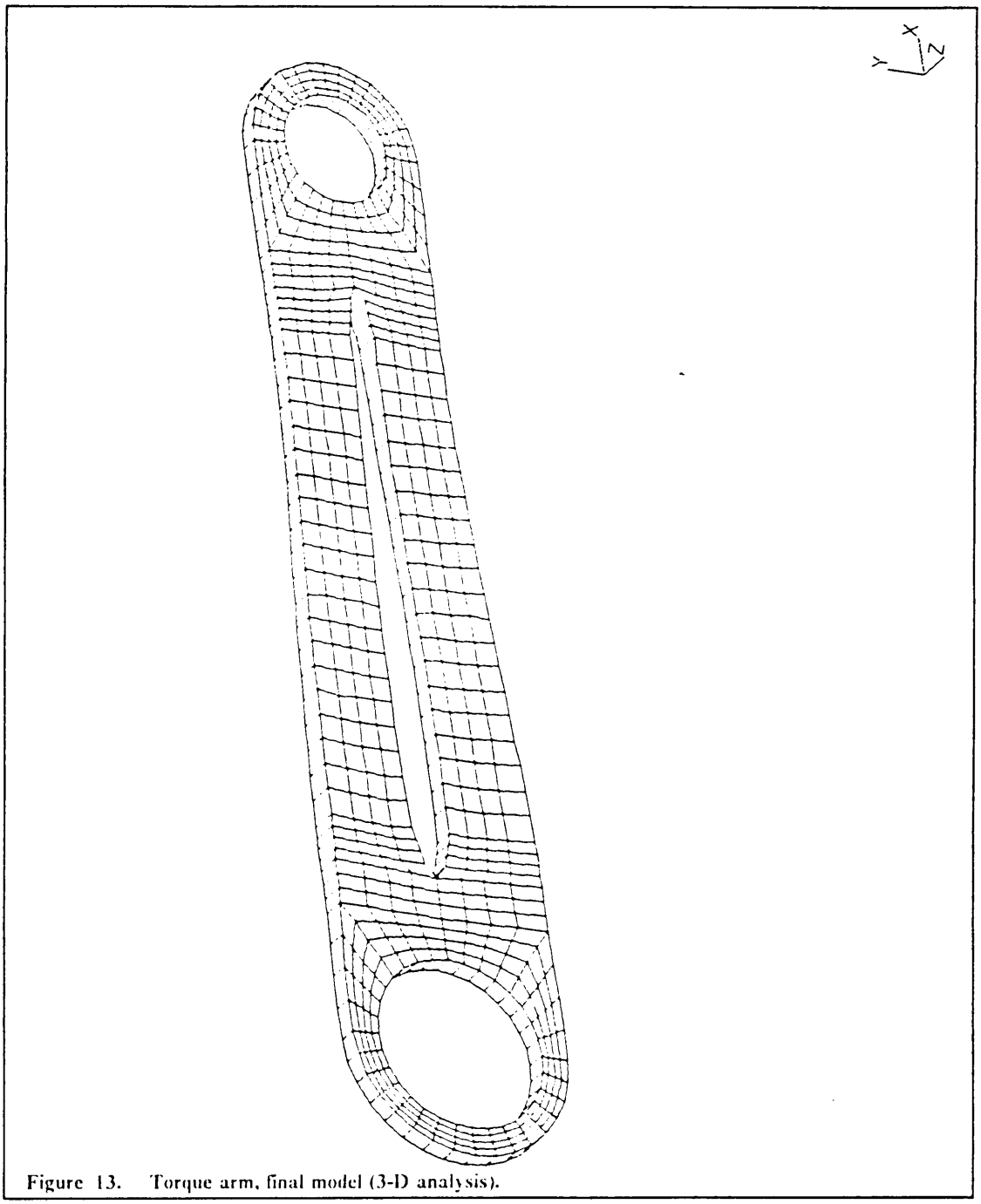


Figure 13. Torque arm, final model (3-D analysis).

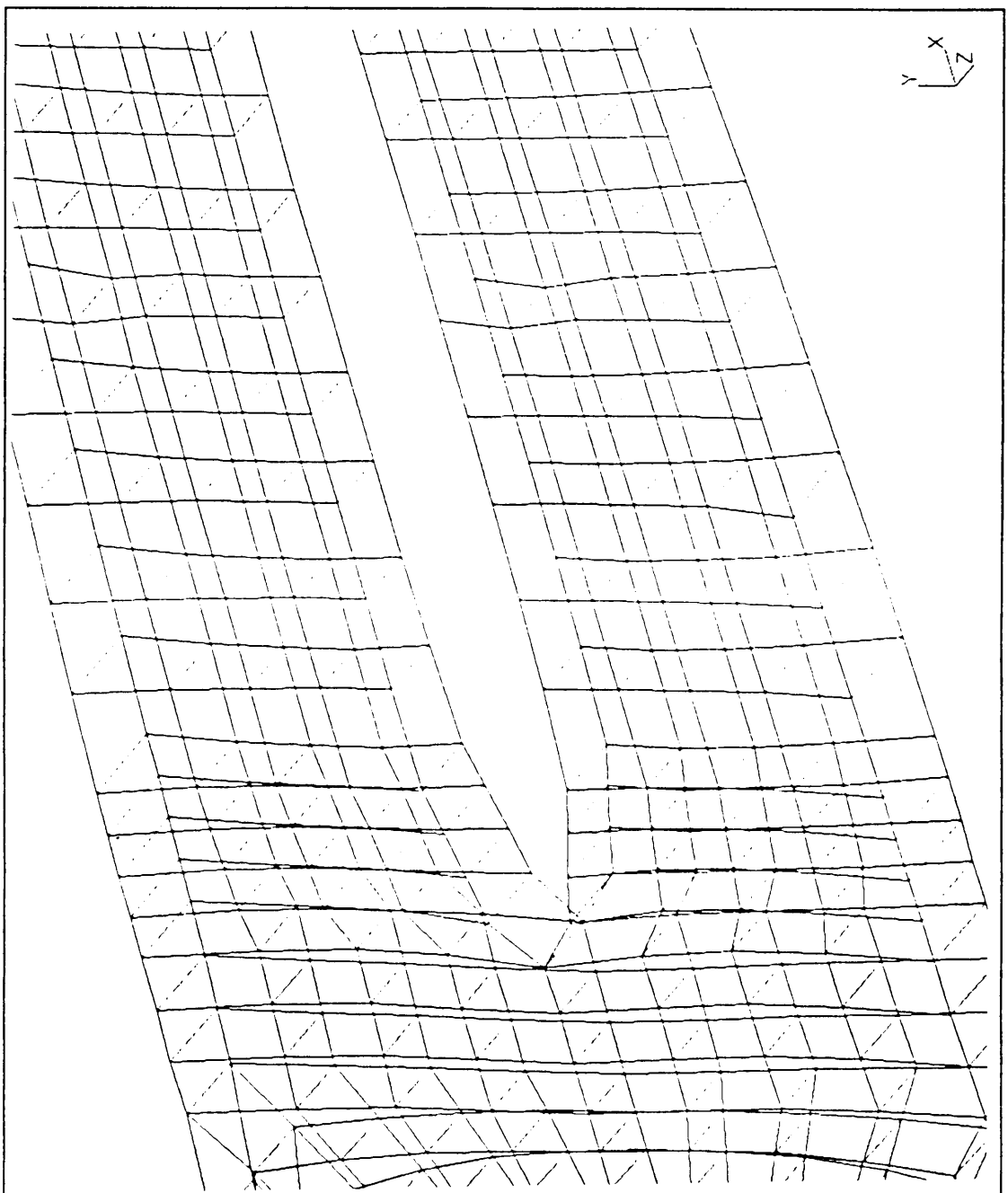


Figure 14. Zoom view of final torque arm model (3-D analysis).

6.3.3 Discussion

The two dimensional final model had elements overlapping at the right side of the center hole after iteration 9. This is a common problem in shape optimization, and no feasible solution exists to keep overlapping from occurring. However, the designer could close the hole where the elements overlap.

Figure 12 shows the model in the iteration before the overlapping began. Material around the far holes shrunk to about two thirds of the original thickness. The center hole expanded at the left side, and closed at the right. Material on the outer boundary contracted.

Figure 11 shows the model in its final state. The center hole closed significantly at the right side. Material around the far holes shrunk to about half of the original thicknesses, and the outer boundaries contracted more. The notches formed at the far left side were caused by element distortion, and should be smoothed out for final design.

Results are compared to reference [4]. Figure 15 shows an iteration history of the author's model. The author's center hole is smaller than the one of this model, but not that much. Most torque arm designs use center holes longer than the one in the figure, and the author's final design shows a substantial increase in hole length. The center hole has assumed a similar shape to the analysis here, with the author using constraint arcs to achieve the circular appearance of the hole, and to prevent the overlapping that occurred in the analysis here. The outer boundaries also assume similar shapes, with the exception of the loss of material around the outer holes and the errant notches in this model. The author used triangular elements, a mesh regenerator, and boundary design elements for his optimization.

The author's torque arm required 45 iterations to achieve convergence, with convergence based on the stress variations of the model. Although this analysis required an enormous amount of computer time, the optimization achieved a 55 percent reduction in model volume. The final shape is also smooth, efficient, and simple to manufacture. The method used in this report achieved a final shape approaching that of the authors', but used fewer iterations. Unfortunately in the optimization reported here, element distortion had an effect on the final results, preventing the dramatic volume loss found in the reference from occurring.

In the three dimensional analysis, the model did not change much in the xy plane. Changes similar to those of the 2-D analysis occurred, but to a less degree. Since the elements expanded or contracted uniformly in this analysis, changes occurred in the z direction that are not seen in an xy plane plot. Figure 13 and Figure 14 show the effects of z direction movement. Elements around the far holes shrank inward. The elements around the center hole expanded on the outer edges, which are the regions of highest stress, and shrank on the inner edges, giving the surface a concave appearance. This phenomenon is more easily seen in Figure 14. If the three dimensional analysis is used as a final design, the part would have to be cast due to its geometry.

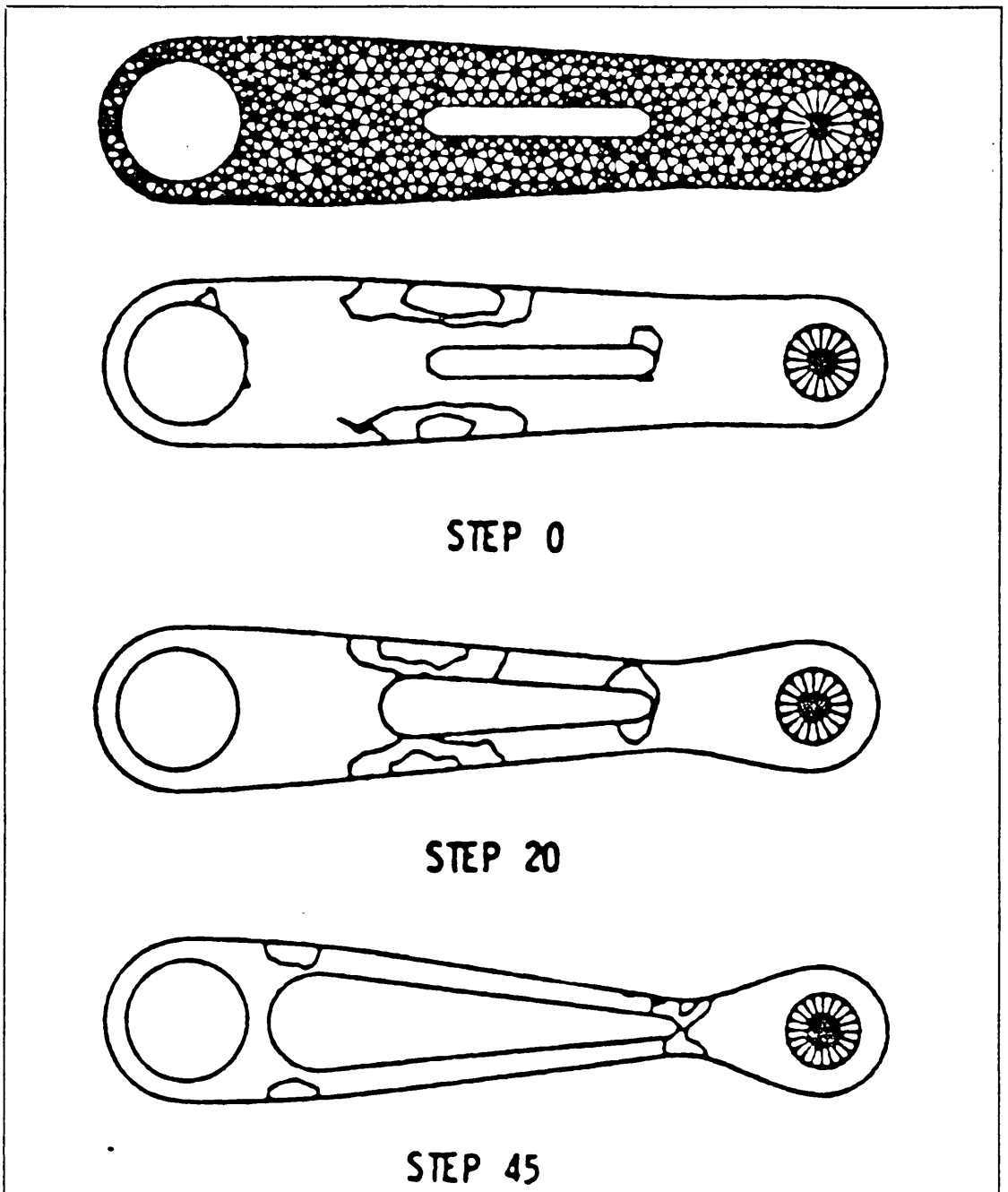


Figure 15. Bennett and Botkin's torque arm iteration history.

Problems occurred in both analyses. In the two dimensional analysis, the effects of distorted elements is seen in the strange notches at the left side. To solve this problem; a mesh regenerator could be used, the mesh refined at those regions, or the designer can smooth out the boundary following the analysis. The closing up of the center hole could be valid, but constraint arcs could be used as in [4] to preserve a circular ended hole.

In the three dimensional case, the model solution started to diverge after the seventh iteration. Altering the variable alpha would not prevent the divergence. This is a perplexing problem, since stress results were accurate, and the model changes were acceptable until the divergence. The inability of the program to achieve a convergence casts some doubt on the validity of the method in three dimensions.

A possible solution to the problem would be increasing the number of elements throughout the thickness to three or four. Perhaps using only a one element thickness leads to numerical difficulties in a three dimensional shape modification. However this would create stiffness matrices so large that the analysis would become infeasible.

6.4 *Draft Sill Casting*

The model of the draft sill casting was created by Roach [5]. Roach created the model for Norfolk Southern and did a manual optimization of it using I-DEAS and the SUPERB finite element solver. In his conclusion, Roach recommended using a three dimensional shape optimizer to shrink the model walls; which he said would significantly reduce model weight.

Roach gives the following description of a draft sill casting in his report: "The draftsill is a cast steel part weighing approximately 1,100 lbs (4890 N) and measuring approximately 6 ft (1.8 m) long and 13 in (33 cm) square. The sill is attached to each end of a hopper car by a long box shaped member called a centersill which runs under the length of the car. The purpose of the draftsill is to transmit to the hopper car all loads applied by the draftgear assembly. The draftgear consists of the coupler and follower blocks and fits into the draftsill in an area called the draft pocket."

The draftsill and draftgear are shown in drawings in Appendix D.

6.4.1 Model description

Pictures of the model appear in Figure 16, Figure 17, Figure 18, Figure 19, and Figure 20. The first figure shows the bottom side of the model, the next three show subsections of the overall side, and the final figure shows the model's top side. The model was created by manual entering of nodes and elements, and uses several hexahedron elements degraded to tetrahedron elements. Although degraded tetrahedron elements give poor stress results, the purpose of the optimization attempt was to verify

that the program could work with a complex model loaded with more than one load case.

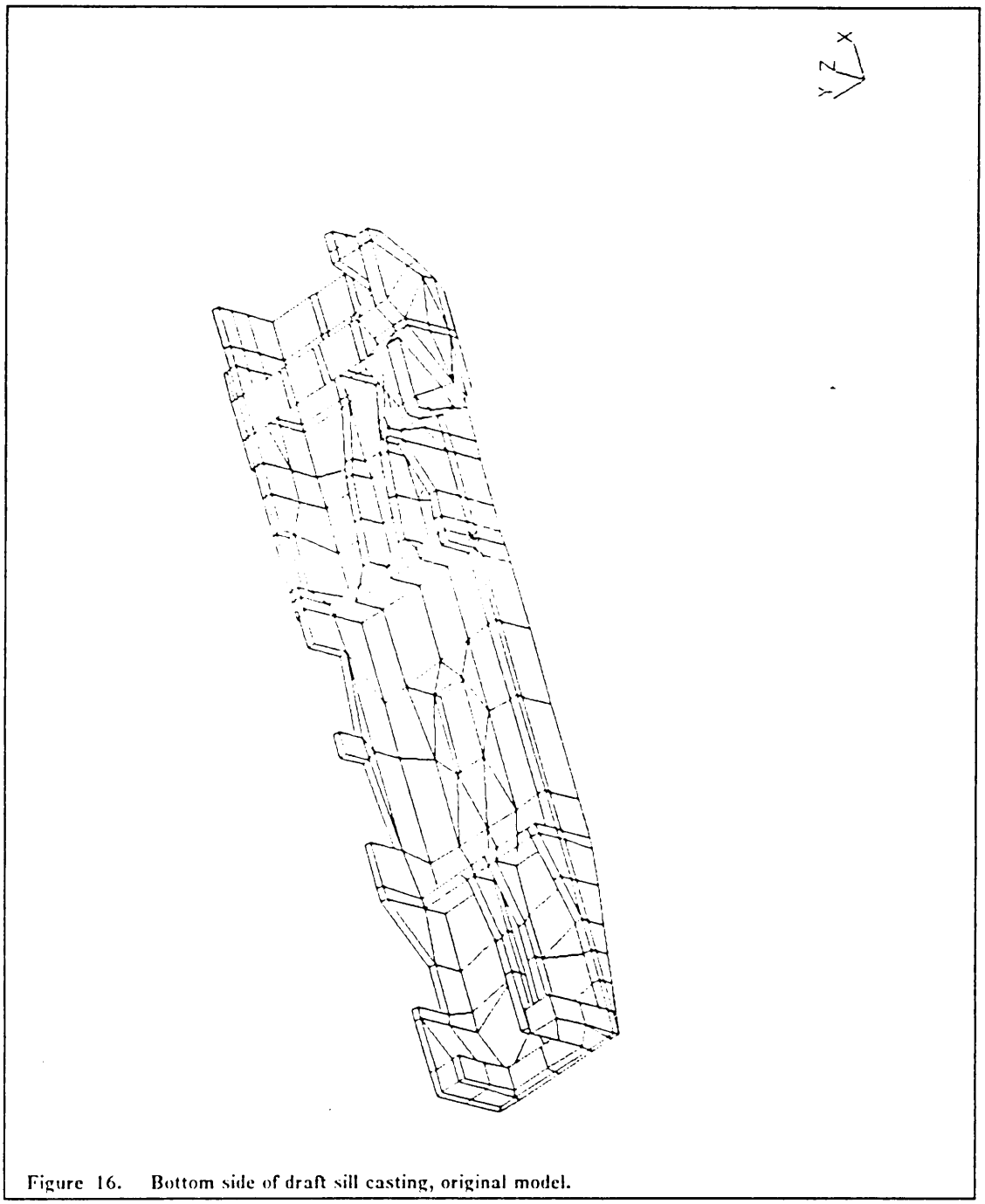


Figure 16. Bottom side of draft sill casting, original model.

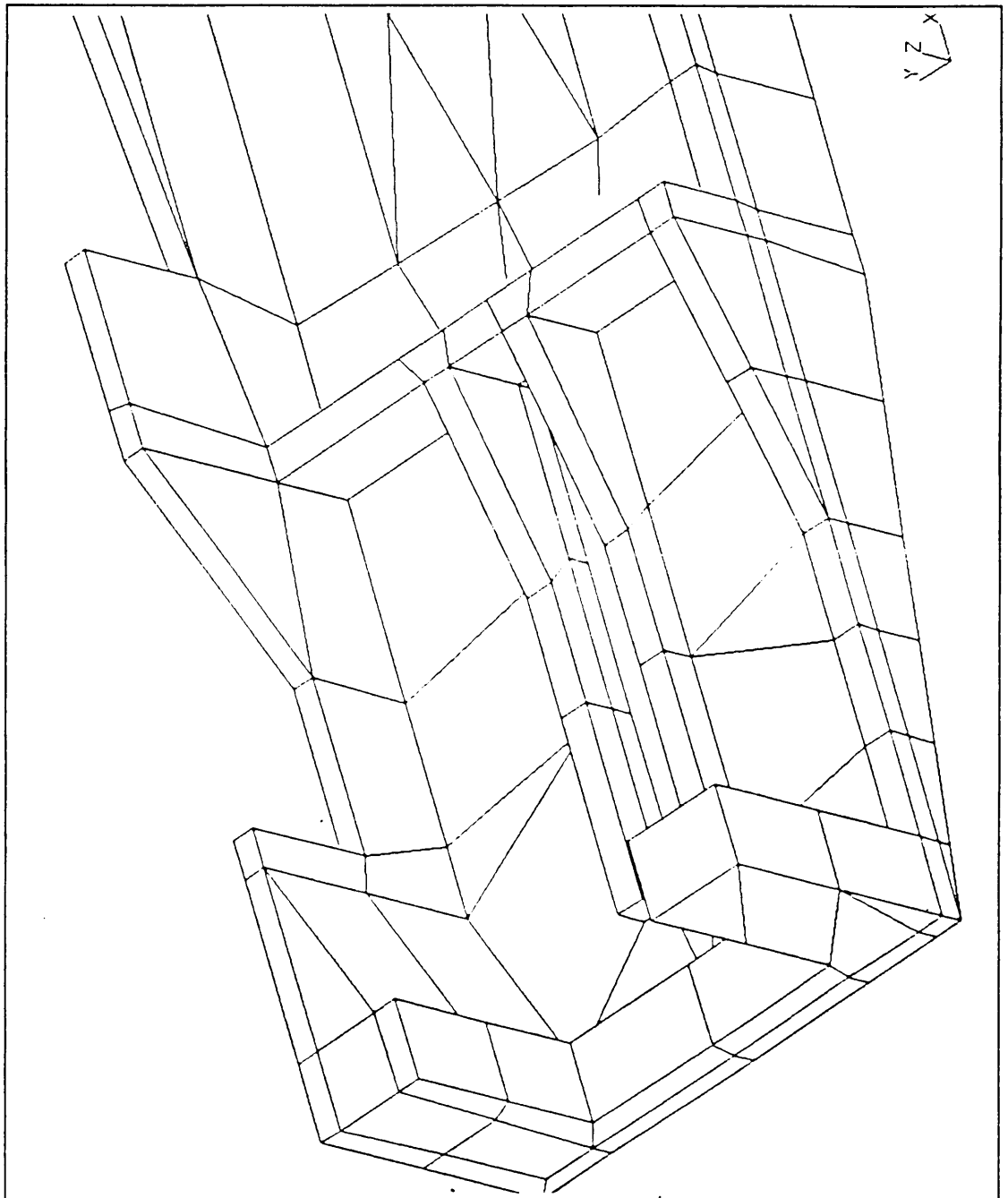


Figure 17. Far left subsection of draft sill casting, original model.

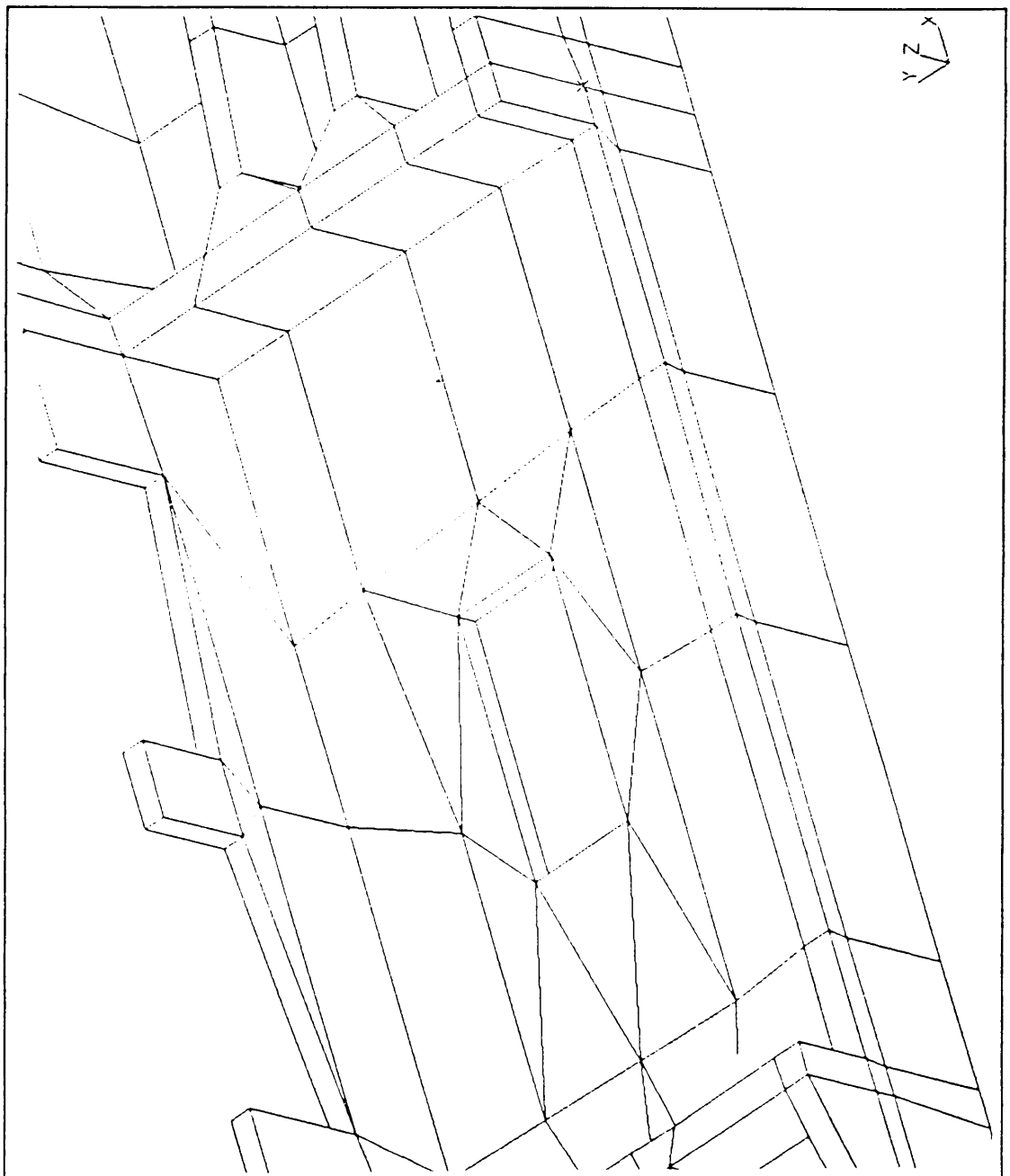


Figure 18. Middle subsection of draft sill casting, original model.

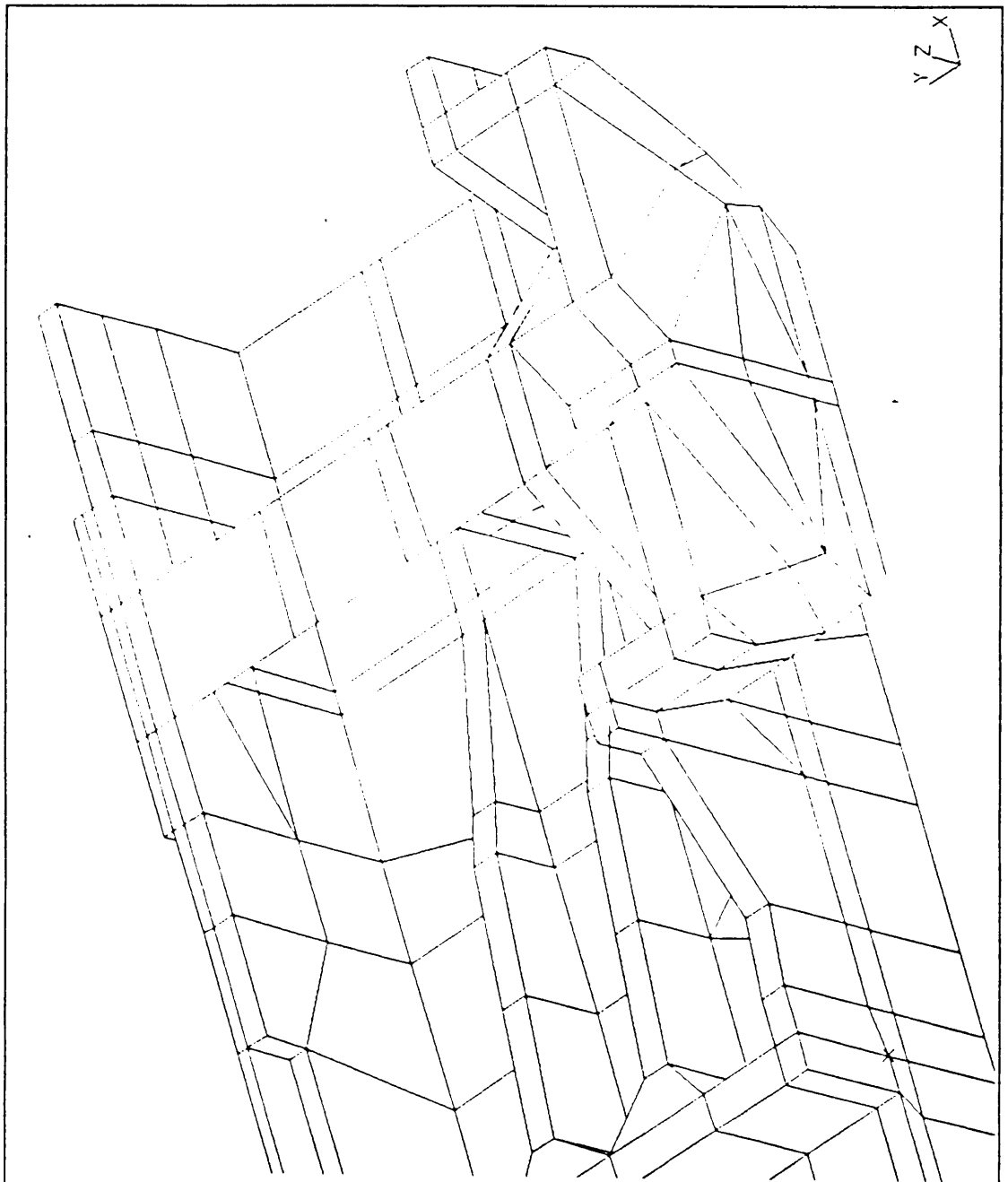


Figure 19. Far right subsection of draft sill casting, original model.

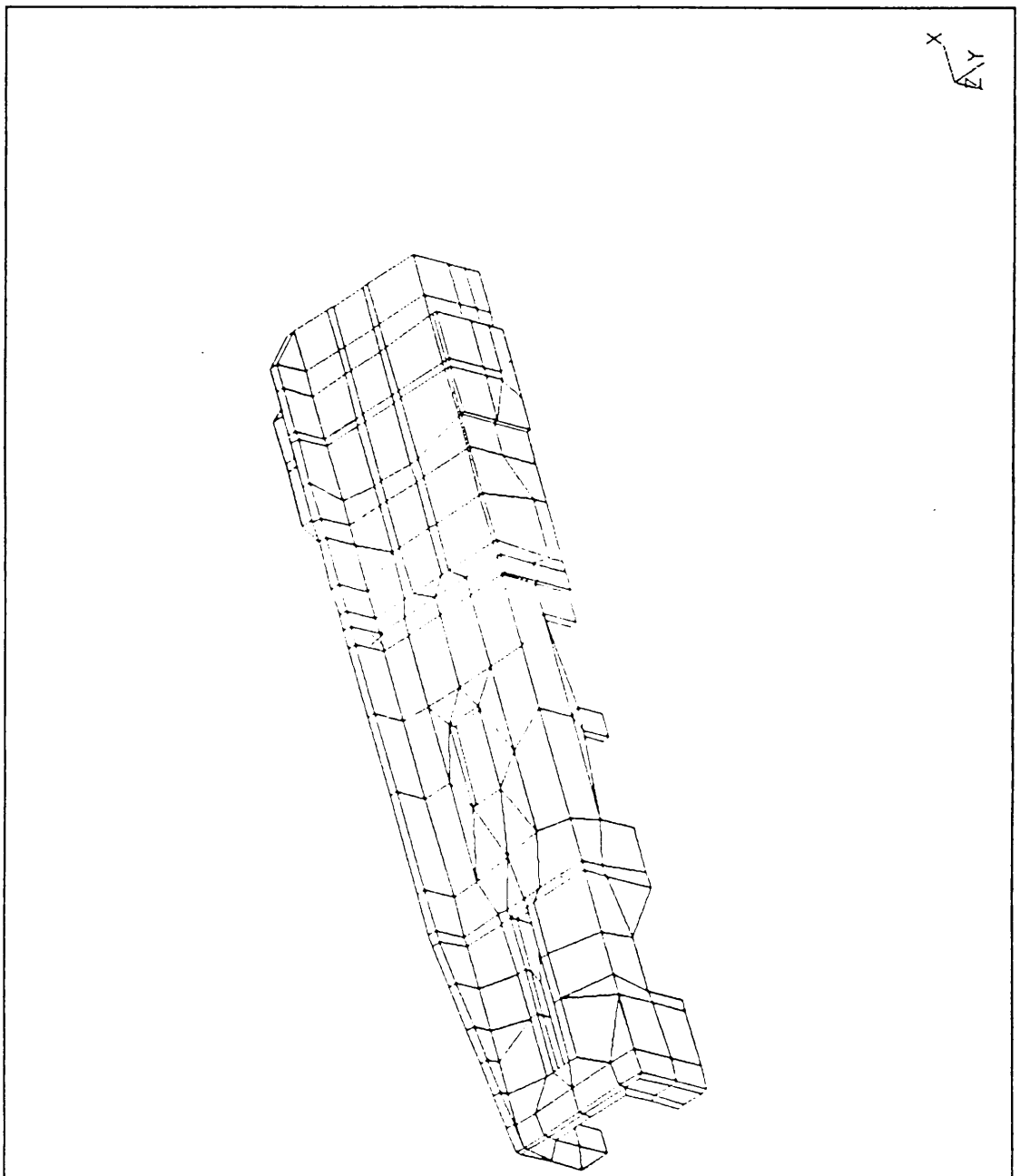


Figure 20. Top side of draft sill casting, original model.

Only half of the draft sill casting is modeled due to symmetry. The model is made up of 678 nodes and 330 elements. Four load cases were used in the analysis:

- Load case 1 : Draft load, simulates generated load when hopper car is pulled by another car. Element face pressures are applied in the - x direction to the first wall from the left (on the yz plane), on its right side.

- Load case 2 : Compressive end load, simulates generated load when hopper car is pushed by another car. Face pressures are applied in the x direction to the second wall from the left (on the yz plane), on its left side.

- Load case 3 : Carbody lift, simulates the car being lifted at its coupler. Face pressures are applied in the - y direction at the first wall from the bottom (on the xz plane) of the keyslot. Pressures are applied on the top side.

- Load case 4 : Downward vertical load, simulates generated load when draftgear is not correctly aligned and is forced downward. Face applied the same as in load case 3, but on the opposite wall and face.

Included in all load cases is a vertical load, which represents the structural weight of the hopper car as it is applied to the draftsill through the bolster beam. Face pressures are applied in the - y direction located above the centerplate (right, top side of drawing) and distributed over the body bolster beam contact area.

The same constraints Roach used for his static analyses were used here. In the expansion phase, the following regions were constrained to preserve certain required model geometries.

- Far left face in the x direction;

- Far left slot in all directions;

- Left side ribs facing bottom hole in all directions;

- Entire back face on inner side in y direction;

- Both walls on sides facing each other in x direction;

- Middle floor on top side in z direction;

- Outer face of the centerplate at the far right side in all directions;
- Far right face in the x direction.

The material used is specified as cast steel, with Young's Modulus of 30 Mpsi (206.9 Gpa) and a maximum yield strength of 70 ksi (482.7 Mpa). An alpha of 0.5, a stress reference of 70 ksi (482.7 Mpa) and a convergence tolerance of 0.04 were used. The stress reference is the yield strength of the steel, which Norfolk Southern designers found acceptable as a design stress. The expansion coefficient and convergence tolerance were chosen by trial and error.

6.4.2 Stress States and Other Results

Stresses in the first iteration ranged from very low (around 2 ksi (13.8 Mpa)) at the far right end, to very high (around 75 ksi (517 Mpa)) at the far left end. The max stresses exceed the yield strength, an indication that the model is underdesigned. However Norfolk Southern engineers calculated similar stress results and concluded that some yielding would not cause the part to fail.

Comparing stress results with Roach show load cases 1, 3, and 4 yield similar stresses. Load case 2 results here were considerably lower however. Since the other three load case results were accurate and only the maximum stress results are used for the expansion analysis, this does not greatly affect the final model.

Stresses did not vary significantly from the first iteration to the final iteration, which is due to the complexity of the model. There was a significant material loss of 238 cubic inches (3,900 cubic cm) though (see Figure 21), a loss of 8.6 %. Pictures of the final model are shown in Figure 22, Figure 23, Figure 24, Figure 25, and Figure 26. These figures use the same views as the figures of the original model, so changes may be ob-

served easily. The CPU times for the analysis were: 0:05.72 minutes for a static analysis, 0:09.64 minutes for a single analysis, and 1:07.5 minutes for the entire process.

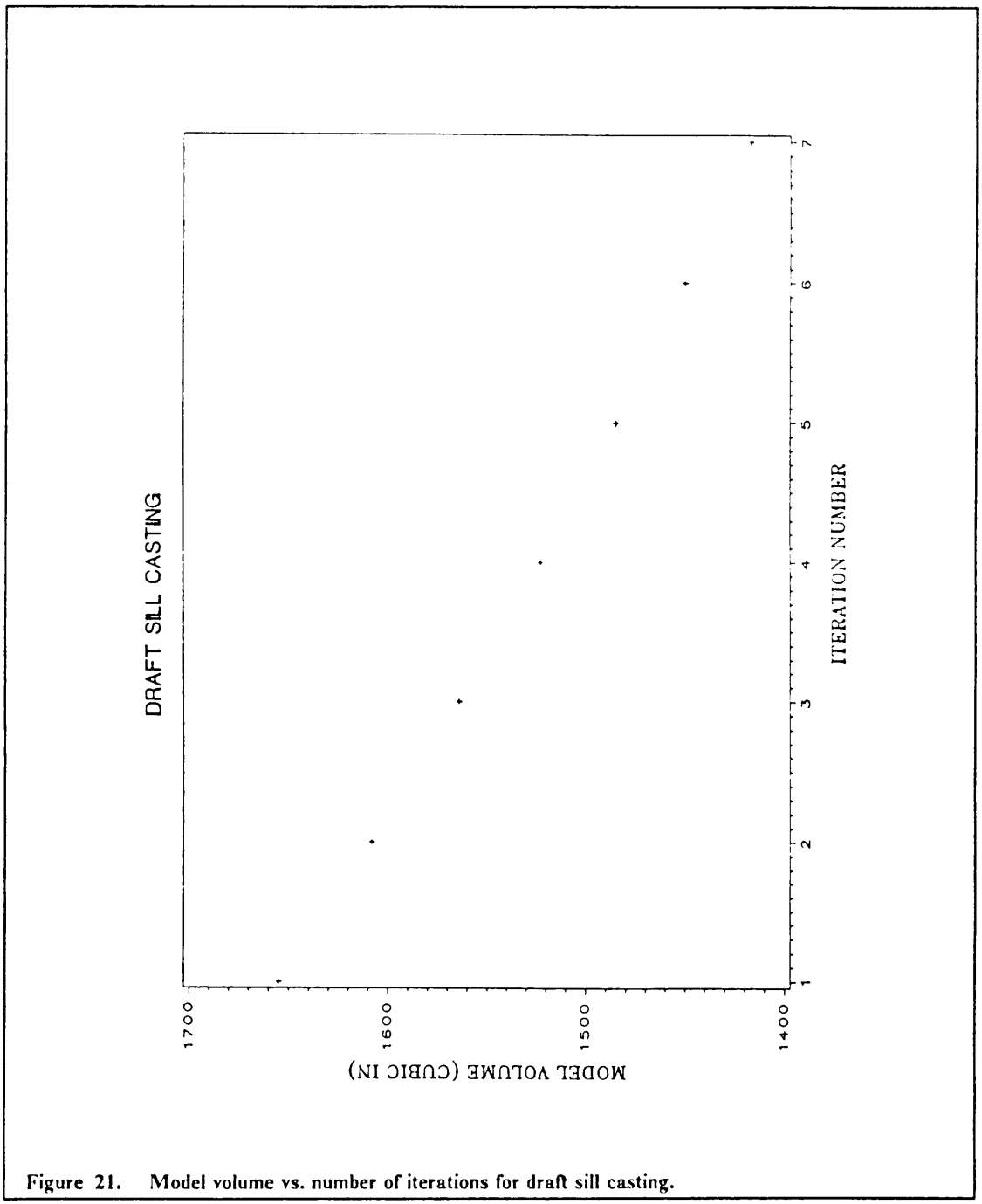


Figure 21. Model volume vs. number of iterations for draft sill casting.

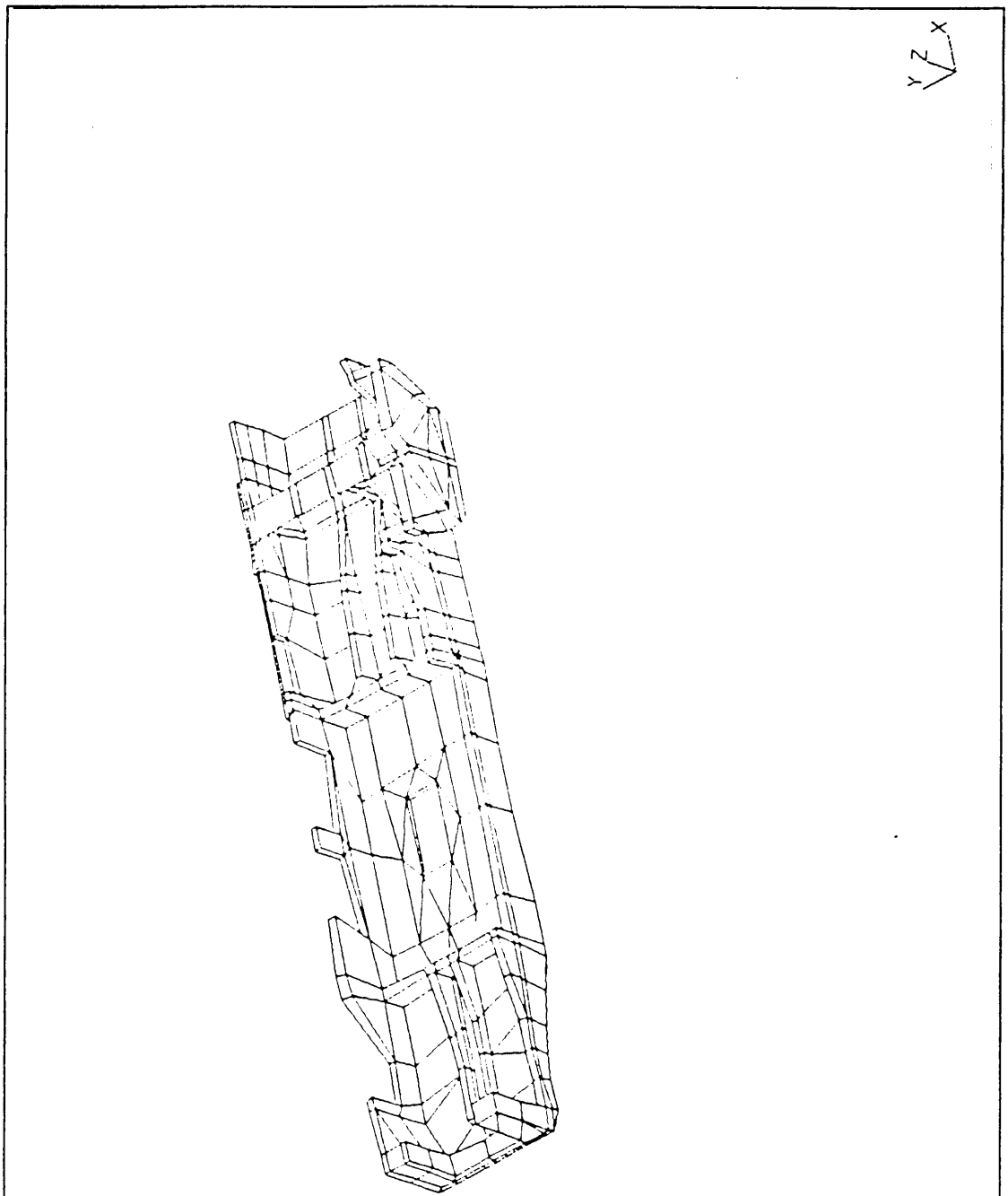


Figure 22. Bottom side of draft sill casting, final model.

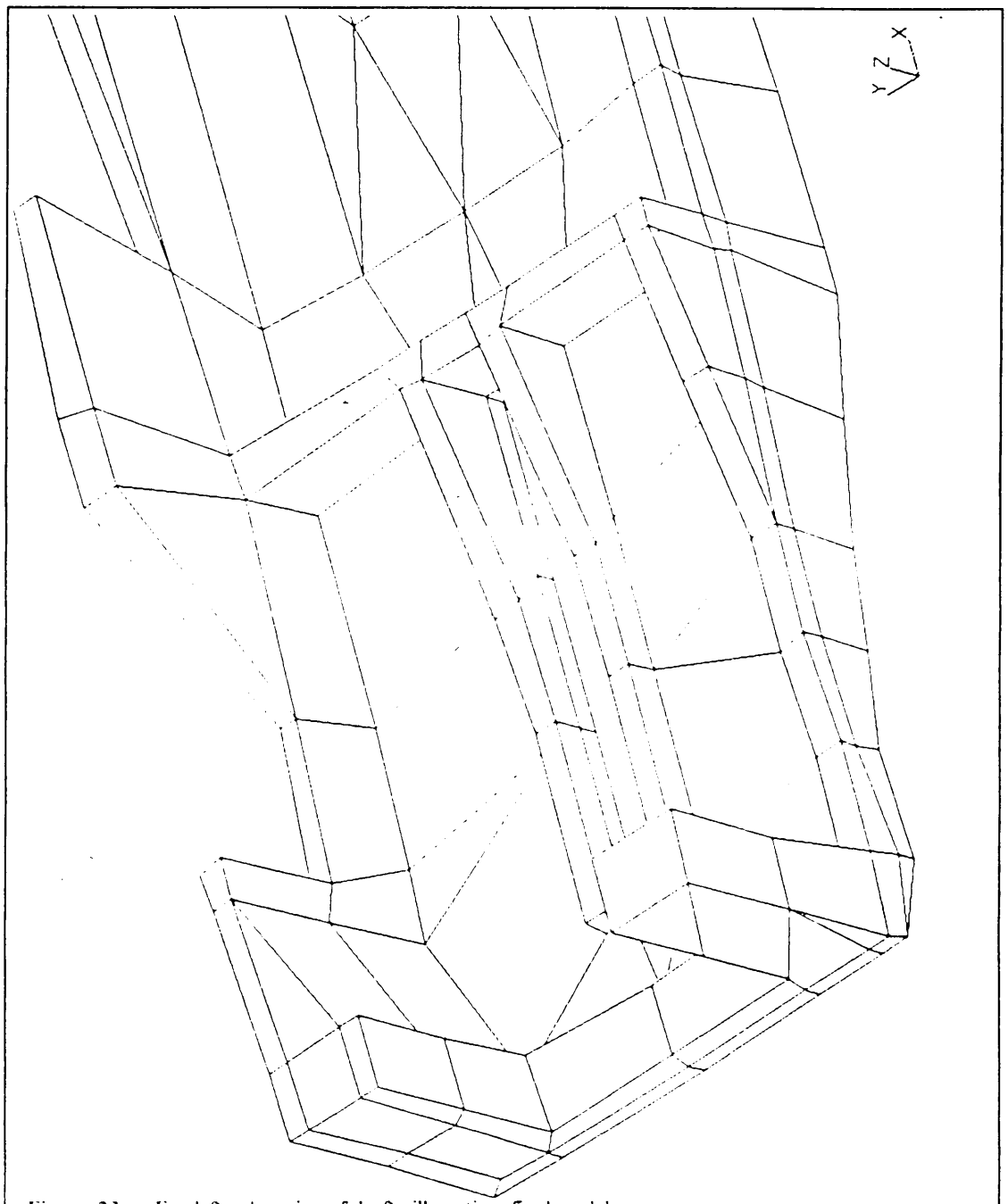


Figure 23. Far left subsection of draft sill casting, final model.

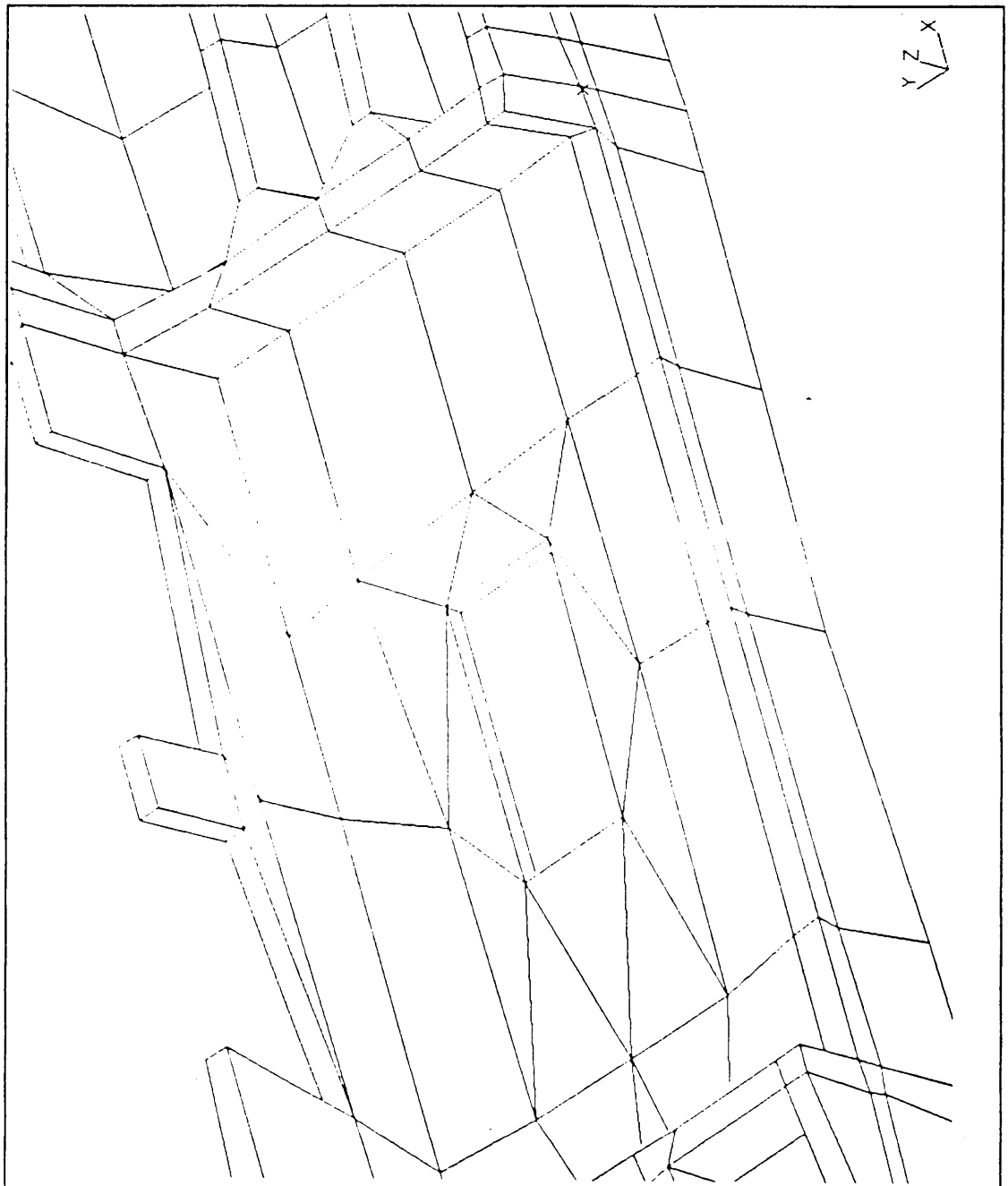


Figure 24. Middle subsection of draft sill casting, final model.

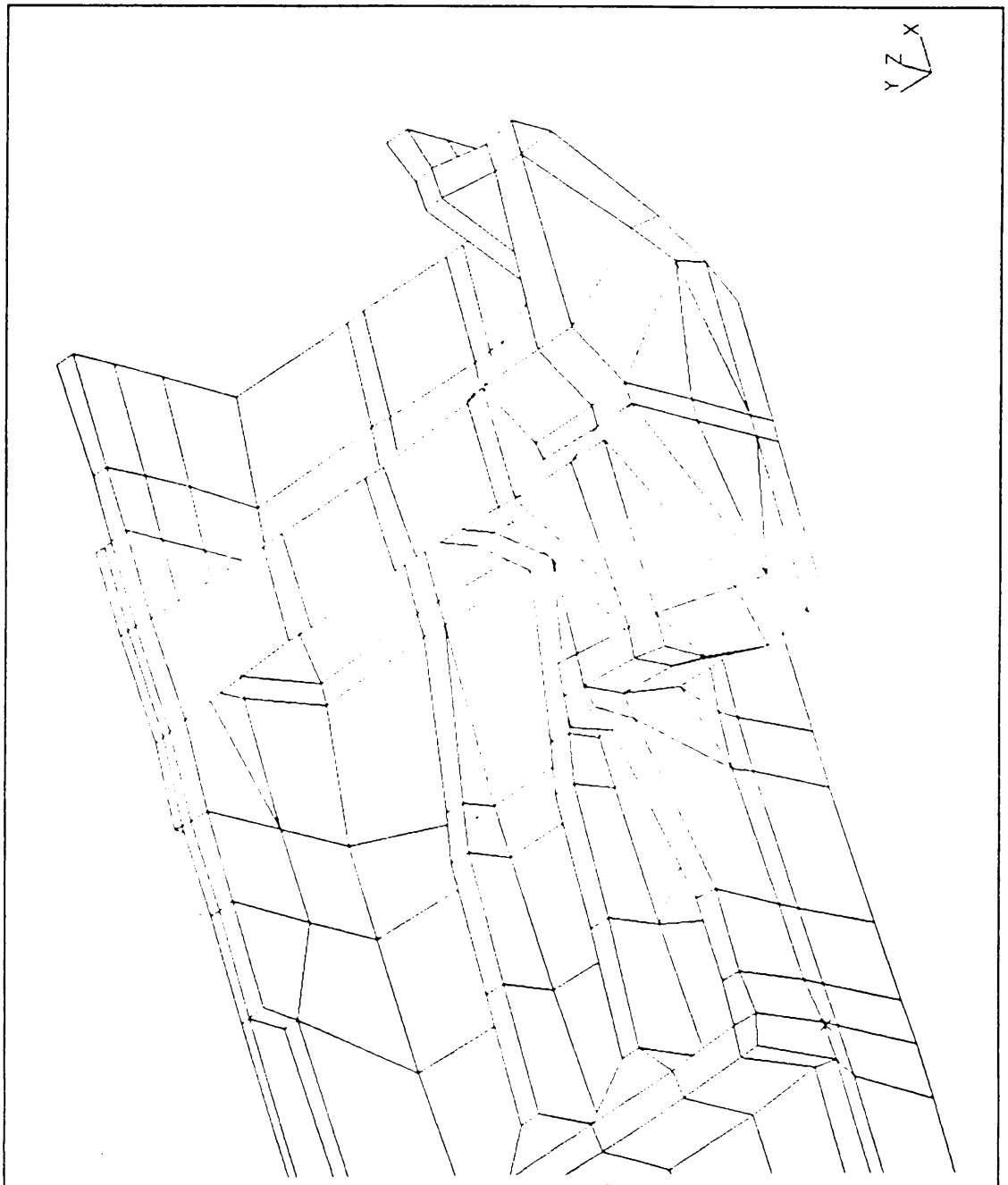


Figure 25. Far right subsection of draft sill casting, final model.

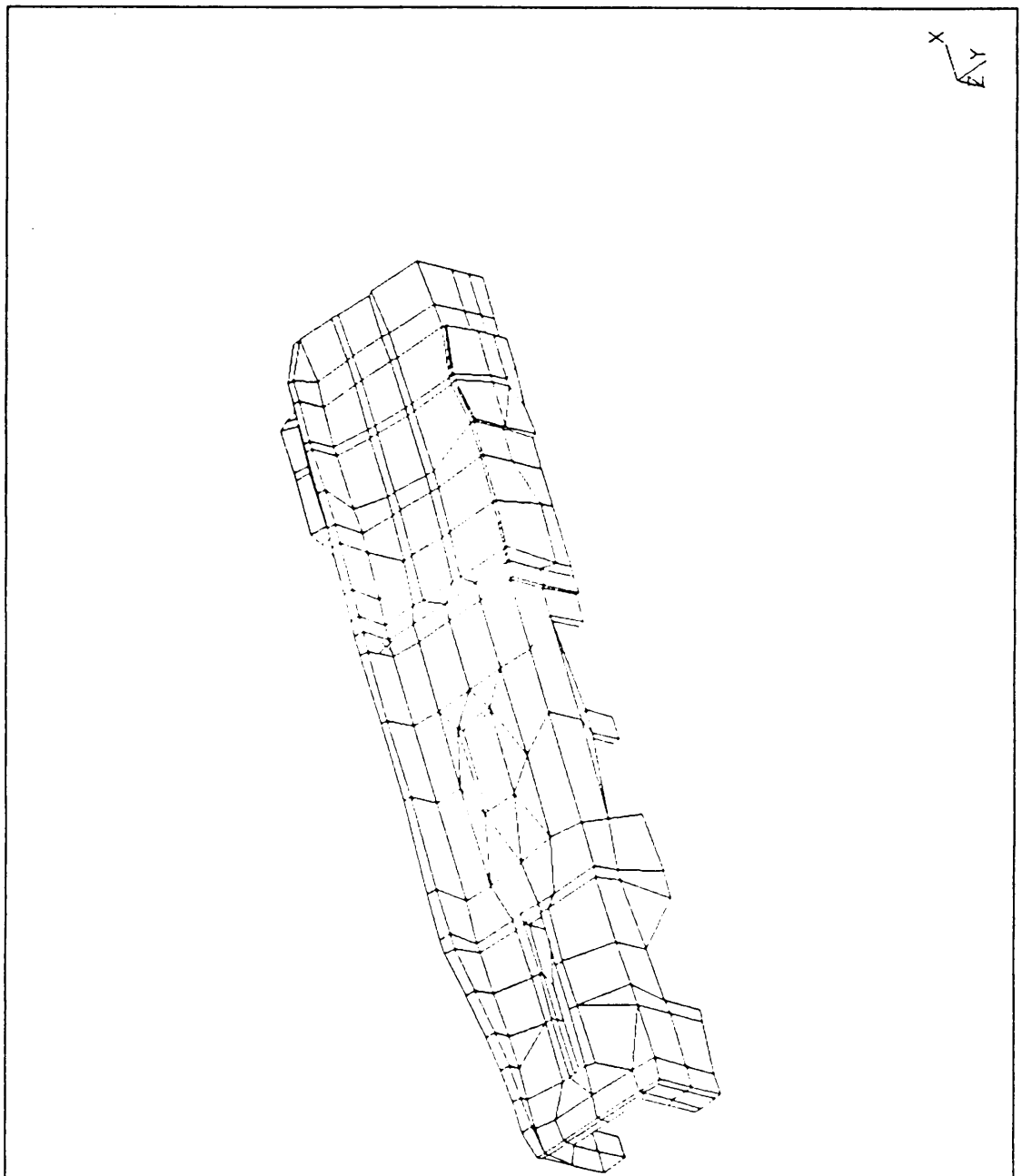


Figure 26. Top side of draft sill casting, final model.

6.4.3 Discussion

The final model is slightly distorted as the pictures show. Variable wall thicknesses as well as curved boundaries appear. These may be smoothed by the designer however, and since the part is cast, no major manufacturing problems will occur.

The stress discrepancies of load case 2 are disturbing, as are the model's varying element sizes. Recall the degraded tetrahedron elements used in the model cause poor stress results, and due to the sparsity of the mesh, some areas with stress concentrations may have been overlooked. For more detailed and accurate results, the model should be regenerated using a mapped mesh generator to produce a more detailed, uniform mesh using only hexahedron elements. A better mesh would give better stress results. With more accurate stresses and a finer mesh, the final model would have a smoother geometry.

As the model stands now, entire walls and floors are modeled by less than ten elements. Not only do stresses vary greatly, but so do the final geometries.

In spite of the problems encountered in the analysis, the program demonstrated its ability to work with a complex model, and to base shape changing on stress results from multiple load cases. The difficulties described above are all attributed to the model, rather than the method used to optimize it.

Chapter 7

CONCLUSION

As the program stands now, it is a valid, simple, method for shape optimization. No excessive model preparation is required, and analyses do not require excessive CPU time to run.

The pressure vessel models were analyzed accurately, and converged to uniform thickness models without difficulty. The torque arm model was analyzed correctly, as the supporting calculations show, and significant volume loss occurred in the optimization process. The draft sill casting analysis also gave reasonable stress results, and was reduced in volume by almost 10 percent.

Problems did present themselves however. In the two dimensional torque arm analysis, elements overlapped at one end; and distorted elements caused the appearance of notches about an end hole. In the three dimensional analysis, solution divergence occurred after about a nine percent volume loss. The draft sill casting lost volume, but attained a slightly warped appearance.

Problem causes are with the presented method and with the models. The three dimensional torque arm analysis could have diverged due to a model thickness of only one element. The draft sill casting model was sparse and included several hexahedron elements that were degraded to tetrahedron elements, which caused poor stress results and the warped final model. The distortion in the final torque arm model however, is only attributed to the program's inability to refine meshes and prevent element distortion from having an effect in the analysis.

Some possible program refinements which could solve the above problems are:

- Compare models made of parabolic elements to the linear ones analyzed here. Parabolic elements would give better stress results.
- Consider mesh regeneration after a certain point to eliminate the effects of element distortion. Also using mesh refinement in areas of stress concentration will give better stress results and shape modifications.
- Implement constraint arcs to supplement nodal d.o.f. constraints and constraint planes to help eliminate the problem of overlapping elements.

The results given here, while not perfect, are promising. The expansion analysis method appears to be a valid and simple means of shape optimization. Computer time is not exorbitant, and no sensitivity analysis is required to reshape the structure. All nodal coordinates may be used as design variables, rather than only a select few, which eliminates costly trial and error optimizations.

Simple models, such as the pressure vessels, may be optimized quickly and easily without using excessive computer time. Complex models, such as the torque arm and draft sill casting, must be created with care, and with detailed and accurate meshes to prevent inaccurate results and solution divergence. With the modifications suggested above, larger models may be modified and optimized better, with more useful final shapes resulting.

REFERENCES

1. Haftka, R.T. and Grandhi, R.V., "Structural Shape Optimization - A Survey.", *Computer Methods in Applied Mechanics and Engineering*, Vol. 57, 1986, pp. 91-100.

2. Oda, J. and Yamazaki, K., "On a Technique to Obtain an Optimum Strength Shape of an Axisymmetric Body by the Finite Element Method", ASME paper number 150-2 1977.

3. Braibant, V. and Fleury, C., "Shape Optimal Design - A CAD Oriented Formulation", *Engineering with Computers*, 1986, pp. 193-204.

4. Bennett, J.A. and Botkin, M.E., "Shape Optimization of Two-Dimensional Structures with Geometric Problem Description and Adaptive Mesh Refinement", Research publication number GMR-4297, presented at the AIAA/ASME/ASCE/AHS Structures, Structural Dynamics, and Materials Conference, Lake Tahoe, Nevada 1983.

5. Roach, D., "A Finite Element Analysis and Redesign of the Draftsill Casting on a Railroad Hopper Car", Department of Mechanical Engineering, Virginia Polytechnic Institute and State University, 1986.

6. Hambric, S.A., Grosse, I.R., and Knight, C.E., "Structural Shape Optimization Program User's Guide", Department of Mechanical Engineering, Virginia Polytechnic Institute and State University, 1987.

7. Cook, R.D., *Concepts and Applications of Finite Element Analysis*, 2nd Ed., John Wiley and Sons, 1981.

8. Cook, R.D.; Young, W.C., *Advanced Mechanics of Materials*, MacMillan, 1985.

Appendix A

PRESSURE VESSEL CALCULATIONS USING LAME' EQUATIONS

A.1 Spherical pressure vessel equations and calculations

Lame's equation for the radial stress, for σ_r, is used to approximate σ_x in the elements on the bottom face and σ_y in the elements on the top face. The equation for the tangential stress, σ_t , is used to approximate σ_y in elements on the bottom face and σ_x in elements on the top face.

Lame's equation for σ_r, was found in reference [8] :

$$\sigma_r = \frac{(p_i - p_o)a^3b^3}{r^3(a^3 - b^3)} + \frac{p_ob^3 - p_ia^3}{a^3 - b^3}$$

$$\sigma_t = \frac{p_ob^3(2r^3 + a^3)}{2r^3(a^3 - b^3)} - \frac{p_ia^3(2r^3 + b^3)}{2r^3(a^3 - b^3)} \qquad [1]$$

Two expressions will be found for each equation. One using the bottom outer radius, and one using the top outer radius. The following constants are defined:

$$\begin{aligned} p_o &= 0.0Pa \\ p_i &= 1.0MPa \\ a &= 0.10m \\ b_{bot} &= 0.11m \\ b_{top} &= 0.125m \end{aligned}$$

Making these substitutions gives the following expressions for the bottom face:

$$\begin{aligned} \sigma_r &= 3.0212Mpa - \frac{4021.0}{r^3} \\ \sigma_t &= 3.0212Mpa + \frac{2011.0}{r^3} \end{aligned} \qquad [2]$$

and the following expressions for the top face:

$$\begin{aligned} \sigma_r &= 1.0492Mpa - \frac{2049.2}{r^3} \\ \sigma_t &= 1.0492Mpa + \frac{1024.6}{r^3} \end{aligned} \qquad [3]$$

Tables of σ_x and σ_y values and the corresponding theoretical values for σ_r and σ_t are shown below. There appears to be an error in tangential stresses calculated in the top of the vessel. This could be due to the poor aspect ratio in the elements in that part of the model. Element thicknesses in the z direction were lowered to about one tenth of the other element dimensions. This discrepancy did not seriously damage the optimization process, although the vessel thickness at the top did not quite match the thickness at the bottom.

The radial stress at the top agreed well with the calculated results, as did radial stress at the bottom. Tangential stress at the bottom was a little lower than the calculated

values, but not significantly so. Recall these calculations are only approximate in that this model is not a uniform thickness model. The nonuniformity will cause departures from theory.

Table 1. Predicted and actual stresses at top of spherical pressure vessel model

$r(m)$	$\sigma_r(Mpa)$ (Lame')	$\sigma_r(Mpa)$ (F.E.)	$\sigma_t(Mpa)$ (Lame')	$\sigma_t(Mpa)$ (F.E.)
0.105	0.772	0.804	1.96	3.80
0.110	0.553	0.546	1.84	3.11
0.116	0.298	0.315	1.72	2.42
0.121	0.107	0.093	1.62	1.64

Table 2. Predicted and actual stresses at bottom of spherical pressure vessel model

$r(m)$	$\sigma_r(Mpa)$ (Lame')	$\sigma_r(Mpa)$ (F.E.)	$\sigma_t(Mpa)$ (Lame')	$\sigma_t(Mpa)$ (F.E.)
0.101	0.882	0.828	4.97	5.72
0.104	0.554	0.569	4.81	5.45
0.106	0.355	0.336	4.71	5.21
0.109	0.127	0.113	4.62	4.89

A.2 Cylindrical pressure vessel equations and calculations

Lame's equation for radial stress, σ_r , is used to approximate σ_x in the elements on the bottom face and σ_y in the elements on the top face. The equation for tangential stress, σ_t , is used to approximate σ_y in elements on the bottom face and σ_x in elements on the top face.

Lame's equation for σ_r with no external pressure was found in reference [8]:

$$\sigma_r = \frac{p_i a^2}{b^2 - a^2}\left[1.0 - \frac{b^2}{r^2}\right] \qquad [4]$$

The equation for σ_t is:

$$\sigma_t = \frac{p_i a^2}{b^2 - a^2}\left[1.0 + \frac{b^2}{r^2}\right] \qquad [5]$$

Using the following constants:

$$\begin{aligned} p_i &= 1.0MPa \\ a &= 0.10m \\ b_{bot} &= 0.11m \\ b_{top} &= 0.125m \end{aligned}$$

yields the following equations:

$$\begin{aligned} \sigma_{rbot} &= 4.762Mpa - \frac{57{,}260}{r^2} \\ \sigma_{tbot} &= 4.762Mpa + \frac{57{,}260}{r^2} \end{aligned} \qquad [6]$$

$$\sigma_{rtop} = 1.778Mpa - \frac{27,780}{r^2}$$
$$\sigma_{ttop} = 1.778Mpa + \frac{27,780}{r^2} \qquad [7]$$

Tables of σ_x and σ_y values and the corresponding theoretical values for σ_r and σ_t are shown below. The predicted radial stresses agree well with calculations at the top and bottom of the model. Predicted tangential stresses do not vary as much as the calculated ones, but the finite element results average about the predicted ones. As before, note the Lame equations are only approximations in a nonuniform thickness model.

Table 3. Predicted and actual stresses at top of cylindrical pressure vessel model

$r(m)$	$\sigma_r(Mpa)$ (Lame')	$\sigma_r(Mpa)$ (F.E.)	$\sigma_t(Mpa)$ (Lame')	$\sigma_t(Mpa)$ (F.E.)
0.105	0.742	0.672	4.35	8.13
0.110	0.517	0.369	4.16	5.16
0.116	0.287	0.175	3.84	2.49
0.121	0.119	0.052	3.68	0.35

Table 4. Predicted and actual stresses at bottom of cylindrical pressure vessel model

$r(m)$	$\sigma_r(Mpa)$ (Lame')	$\sigma_r(Mpa)$ (F.E.)	$\sigma_t(Mpa)$ (Lame')	$\sigma_t(Mpa)$ (F.E.)
0.101	0.886	0.949	10.5	13.6
0.104	0.699	0.924	10.1	11.4
0.106	0.366	0.360	9.89	8.99
0.109	0.178	0.273	9.61	6.79

Appendix B

SUPPORTING CALCULATIONS FOR TORQUE ARM ANALYSIS

Two types of loads are applied to the torque arm; a bending load and an axial load. To check the results of chapter 5, the arm is modeled as a rectangular bar with approximately the same dimensions. The bar has a thickness of 0.01 m, a width of 0.1 m, and a length (from hole to hole) of 0.4 m.

An axial load of 2031 N is applied at one end of the model. An approximate model area is:

$$\begin{aligned} A &= t\,w \\ &= (0.01m)\,(0.10m) \\ &= 0.001m^2 \end{aligned}$$

Axial stress is approximated by:

$$\frac{F}{A} = \frac{2{,}031.0}{0.001m}$$
$$= 2.03Mpa$$

Calculated stresses ranged from 2.4 to 2.6 Mpa, so the approximation is valid and the results are confirmed.

A bending load of 4063 N is applied at the far right end of the model. The moment of inertia of the model is about:

$$I = \frac{b\,h^3}{12}$$
$$= \frac{(0.01)\,(0.1)^3}{12}$$
$$= 0.83333\,(10^{-6})\,m^4$$

Bending stress is approximated by:

$$\sigma = \frac{M\,c}{I}$$

with the moment equal to the applied force times the distance along the arm. The maximum distance along the arm is about 0.4 m and the minimum about 0.1 m. The minimum distance is from the location of the applied load at the center of the right side hole to the left edge of that hole. The maximum distance from the centroid to the outer boundary is about 0.05 m, with the minimum distance about 0.01 m.

Using these values, bending stress varies from about 5 Mpa to 95 Mpa. The calculated stresses run from 6.0 Mpa to 83.0 Mpa. The approximation is therefore valid and the computations are confirmed.

Appendix C

DRAWINGS OF CASE STUDY MODELS

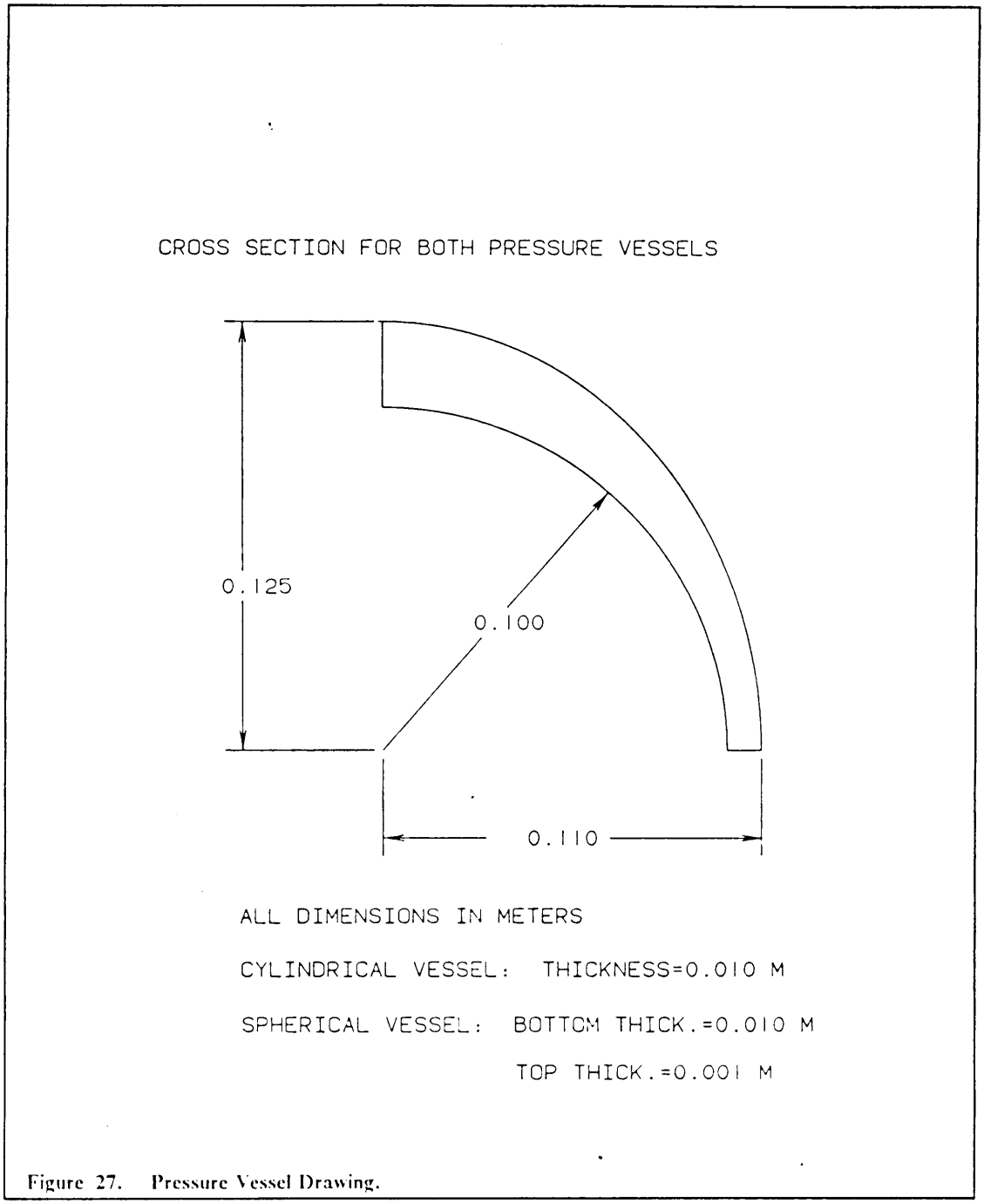


Figure 27. Pressure Vessel Drawing.

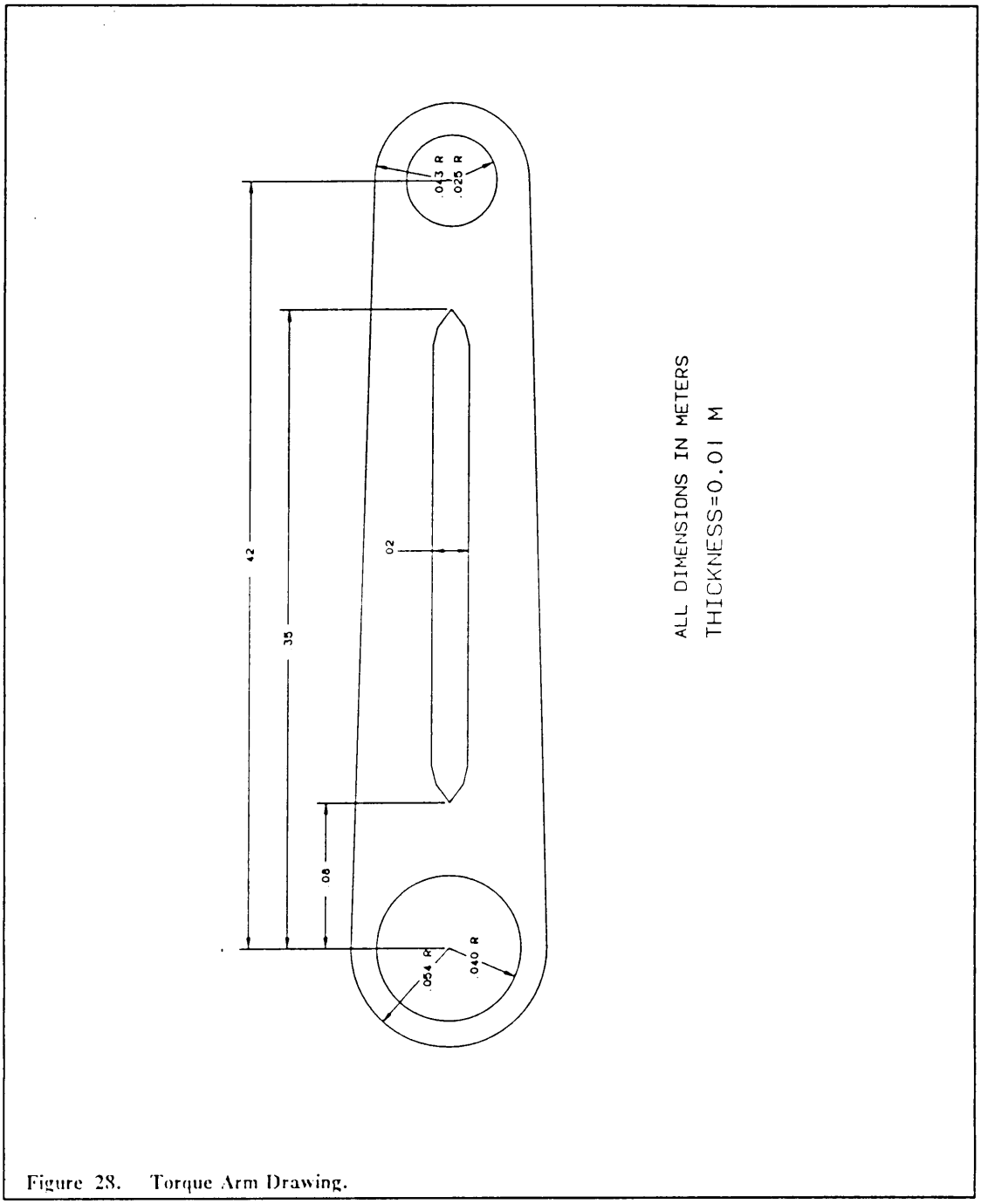
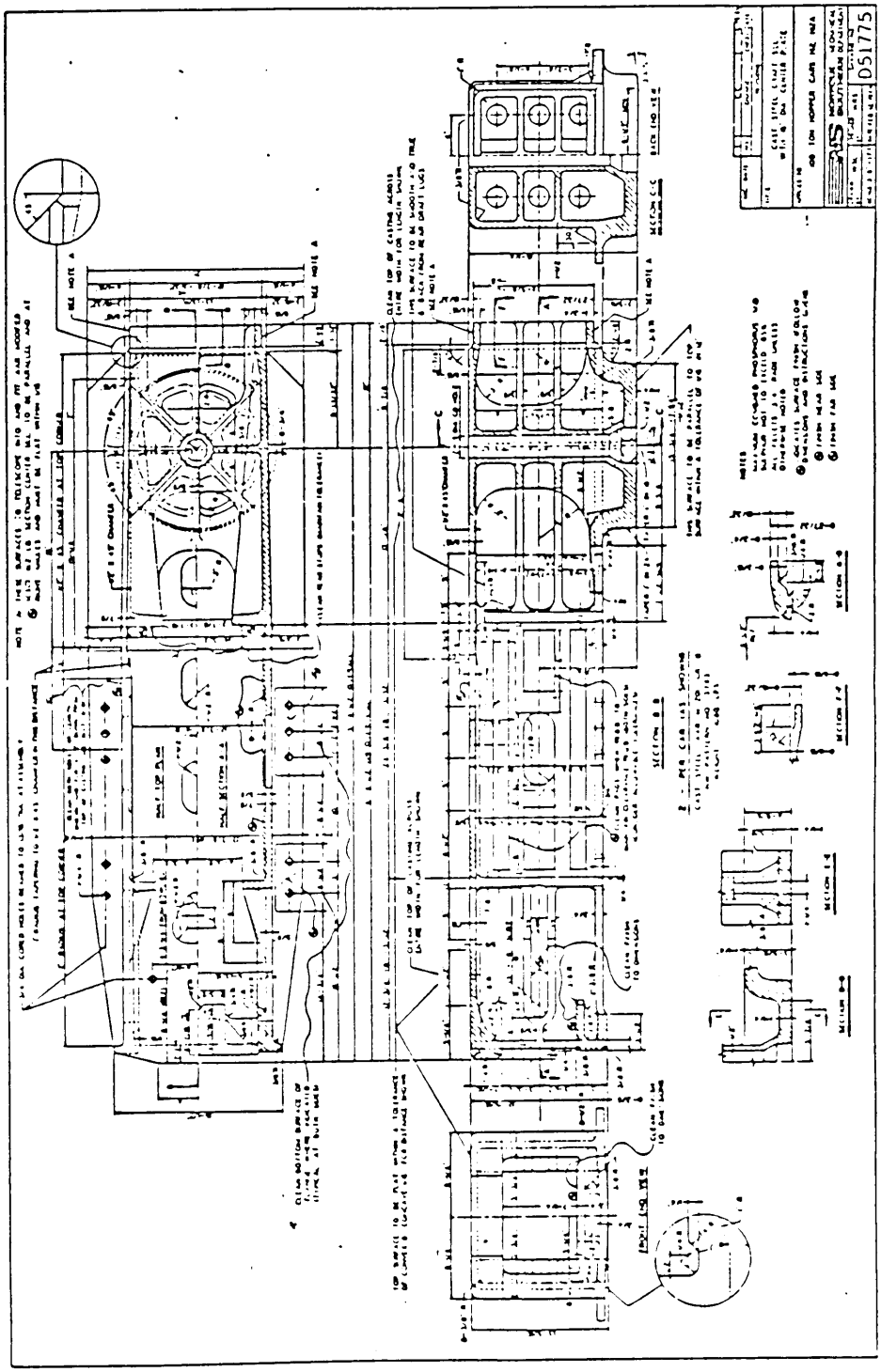


Figure 28. Torque Arm Drawing.

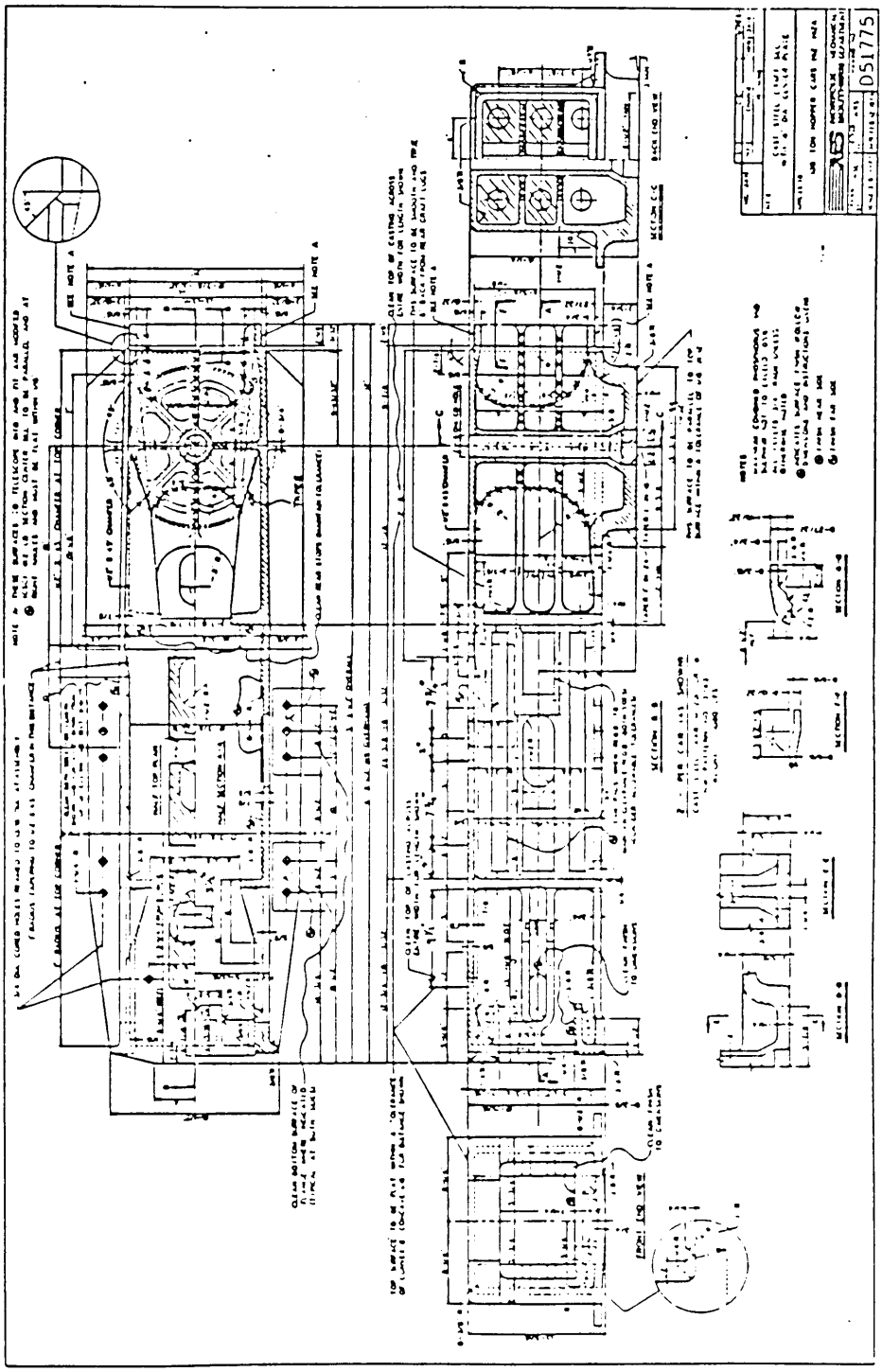
Appendix D

DRAFT SILL CASTING DRAWINGS

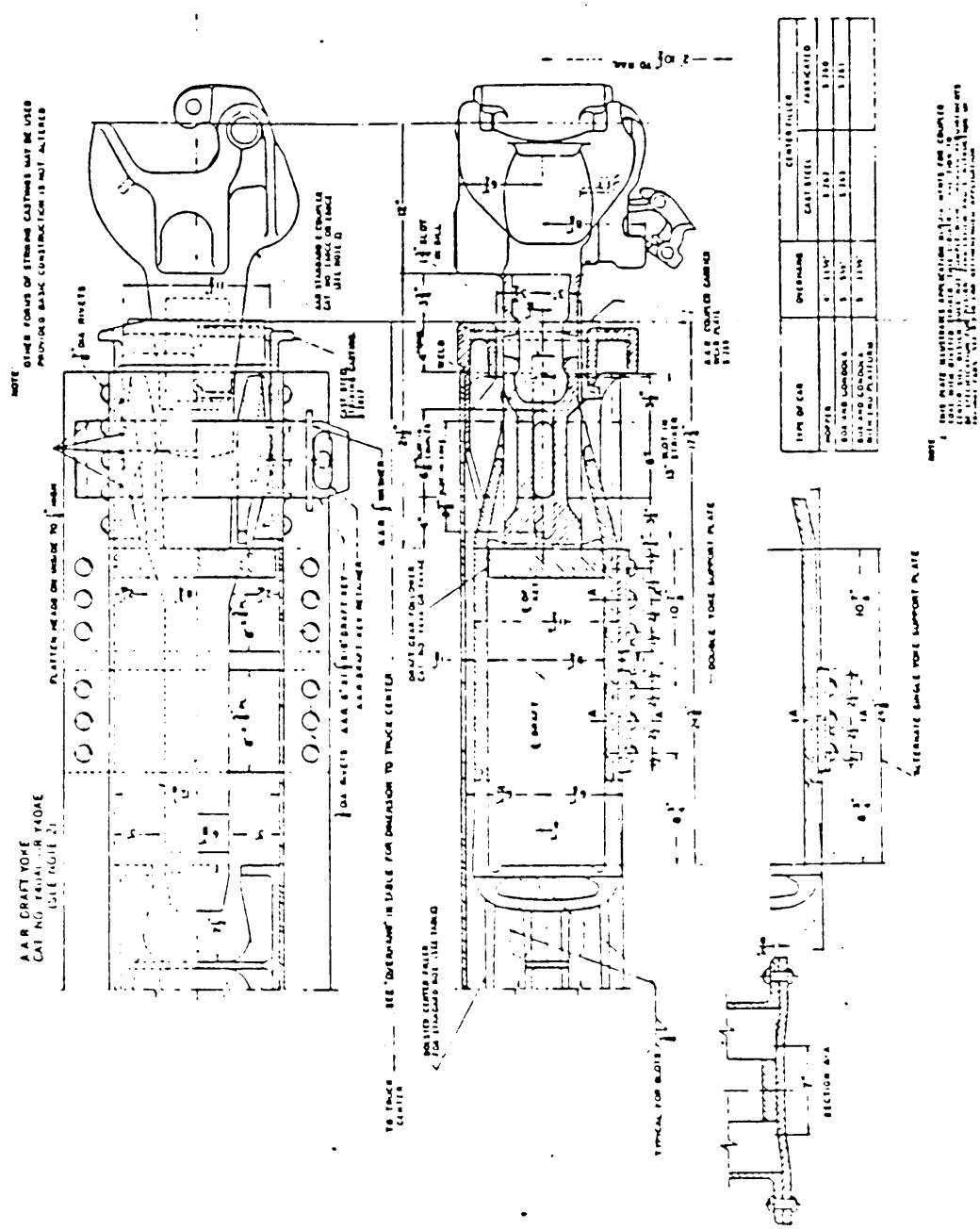
The following figures were taken directly from reference [5].



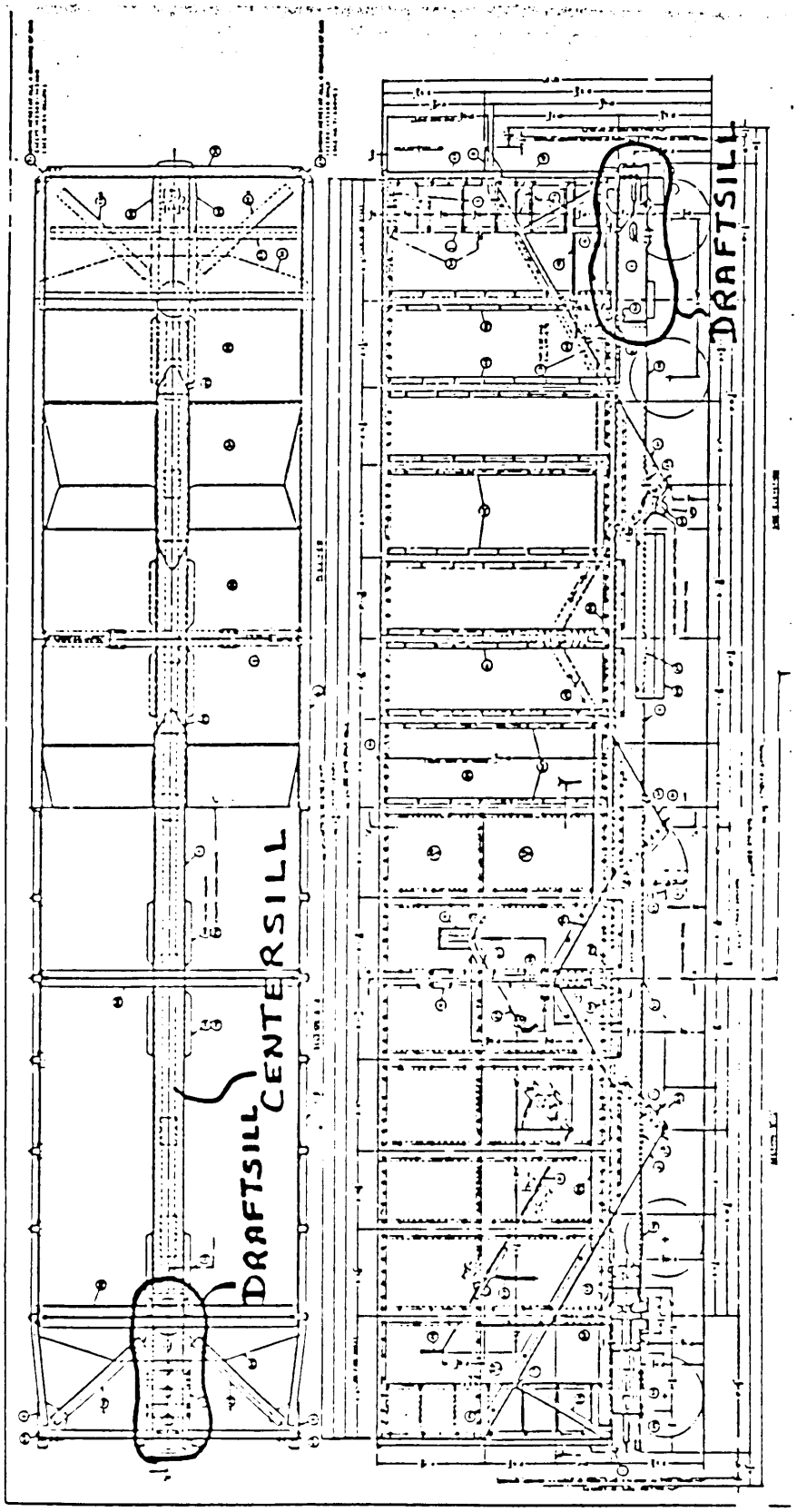
1. Blueprint of the original draftsill casting.



2. Blueprint of the original draftsill casting showing the final design changes made.



3. Blueprint showing a cross-section of the draftgear assembly as it fits into the draftsill.



4. Overall drawing showing how the draftsill fits into the hopper car.

The vita has been removed from the scanned document