An Investigation of Heat Transfer Coefficient and Film Cooling Effectiveness in a Transonic Turbine Cascade

By
Dwight E. Smith

Thesis Submitted to
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science

in

Mechanical Engineering

Thomas E. Diller, Chairman
Wing Ng
Joseph A. Schetz

March 28, 1999
Blacksburg, Virginia

Keywords: heat transfer, film cooling, and transonic turbine

An Investigation of Heat Transfer Coefficient and Film Cooling Effectiveness in a Transonic Turbine Cascade

(Abstract)

This study is an investigation of the film cooling effectiveness and heat transfer coefficient of a two-dimensional turbine rotor blade in a linear transonic cascade. Experiments were performed in Virginia Tech's Transonic Cascade Wind Tunnel with an exit Mach number 0f 1.2 and an exit Reynolds numbers of $5x10^6$ to simulate real engine flow conditions. The freestream and coolant flows were maintained at a total temperature ratio of 2 ±0.4 and a total pressure ratio of 1.04. The freestream turbulence was approximately 1%. There are six rows of staggered, discrete cooling holes on and near the leading edge of the blade in a showerhead configuration. Cooled air was used as the coolant. Experiments were performed with and without film cooling on the surface of the blade. The heat transfer coefficient was found to increase with the addition of film cooling an average of 14% overall and to a maximum of 26% at the first gauge location. The average film cooling effectiveness along the chord-wise direction of the blade is 25%. Trends were found in both the uncooled and the film-cooled experiments that suggest either a transition from a laminar to a turbulent film regime or the existence of three-dimensionality in the flow-field over the gauges.

Acknowledgements

This work was supported by the Air Force Office of Scientific Research (AFOSR) under grant F08671-9601062, monitored by Dr. Jim M. McMichael and Dr. Mark Glauser. We would like to thank Messrs. Scott Hunter, Monty Shelton and Mark Pearson of General Electric Aircraft Engines for their collaboration on this project.

This is an achievement that I could not have made without the help and support of many people. First and foremost I would like to thank God; he gave me strength, guidance, and the will to endure. Secondly, I have to thank my family for always being there and for believing that I could do it. There were times when it was just your faith in me that kept me going. Certainly I need to thank my office mates – Oliver, Jim and Hank – for all of their help and input. I have really enjoyed working with you guys. I would also like to thank Drs. Diller, Ng, and Schetz for all of their advice and support. In addition I would like to thank the Air Force Office of Scientific Research for their support of this project.

I would also like to give thanks to all the people who have made my stay here in Blacksburg, VA an enjoyable experience:

Juan Carlos Alvarez, my partner in crime…thanks for all the laughs.

María Cristina Sánchez …thanks for your friendship and for all the dance lessons.

Tyler Langley, my roommate…it's been a great tour of duty. No wait, yours is just beginning!

Félix Javier Nevárez …thanks for your friendship, I'll see you in PR.

The Latin Community…you've been great!

The Swedish Contingent…Skål !!!

Table of Contents

CHAPTER 1: INTRODUCTION.. 1

CHAPTER 2: LITERATURE REVIEW.. 4

CHAPTER 3: EXPERIMENTAL SETUP AND INSTRUMENTATION.. 8

3.1 CASCADE WIND TUNNEL COMPRESSOR SYSTEM.. 9
3.1.1 Heated Flow.. 9
3.1.2 Coolant Supply System.. 11
3.2 TEST SECTION.. 12
3.2.1 Instrumented Blade.. 13
3.2.1.1 Pressure Side.. 15
3.2.1.2 Suction Side.. 17
3.3 INSTRUMENTATION.. 19
3.3.1 Data Acquisition.. 19
3.3.2 Heat Flux Microsensor.. 19
3.3.3 Kulite Pressure Transducer.. 23
3.3.4 Amplifiers.. 24
3.3.4.1 Amp-6.. 24
3.3.4.2 Measurements Group 2310.. 25
3.4 EXPERIMENTAL SETUP.. 25
3.4.1 Uncooled Blade.. 25
3.4.1.1 Overview.. 25
3.4.1.2 Setup.. 26
3.4.1.3 Procedure.. 26
3.4.2.1 Overview.. 27
3.4.2.2 Setup.. 27
3.4.2.3 Procedure.. 28
3.4.3 Film-cooled Blade (Ambient Coolant).. 28
3.4.3.1 Overview.. 28
3.4.3.2 Setup.. 29
3.4.3.3 Procedure.. 29

CHAPTER 4: EVALUATION OF TEST SECTION.. 31

4.1 OVERVIEW.. 31
4.2 OIL FLOW VISUALIZATION.. 31
4.3 SCHLIEREN AND SHADOWGRAPH PHOTOGRAPHY.. 32
4.4 STATIC PRESSURE MEASUREMENTS.. 34

CHAPTER 5: GENERAL ANALYSIS.. 36

5.1 UNCOOLED ANALYSIS.. 36
5.2 COLD COOLANT ANALYSIS.. 41

CHAPTER 6: STEADY HEAT TRANSFER EXPERIMENTS.. 46

6.1 UNCOOLED HEAT TRANSFER EXPERIMENTS.. 46
6.1.1 Objectives.. 46
6.1.2 Discussion.. 46
6.2 COLD COOLANT HEAT TRANSFER EXPERIMENTS.. 50
6.2.1 Objectives.. 50
6.2.2 Discussion.. 50

CHAPTER 7: CONCLUSIONS AND FUTURE WORK.. 57

REFERENCES.. 59

APPENDIX A: AUTHOR'S PUBLISHED RESEARCH 60

APPENDIX B: COMPUTER PROGRAMS 77

APPENDIX B-1: HFS CALIBRATION 78
APPENDIX B-2: DATA PROCESSING 92

APPENDIX C: HFM COEFFICIENTS AND SENSITIVITIES 133

APPENDIX D: AMBIENT COOLANT EXPERIMENTS 134

List of Figures

Figure 1: Typical Turbofan Engine 1
Figure 2: Schematic of Transonic Cascade Wind Tunnel 8
Figure 3: Circulation Loop for Heated Flows 10
Figure 4: Schematic of Coolant Supply System 11
Figure 5: Picture and Schematic of Test Section 13
Figure 6: Picture of Entire Instrumented Blade 14
Figure 7: Schematic of Blade Manufacturing 15
Figure 8: Pressure-side Piece of Instrumented Blade 16
Figure 9: Schematic of Coolant Hole Alignment 17
Figure 10: Suction-side Piece of Instrumented Blade 18
Figure 11: Schematic of HFS Thermopile 22
Figure 12: Picture of Surface Oil-flow Visualization 32
Figure 13: Schlieren Photograph of Flow in Blade Passages 33
Figure 14: Shadowgraph of Cooling Film on Blade 34
Figure 15: Mach Number Distribution Along Blade 35
Figure 16: Sample Time History of an Uncooled Run 38
Figure 17: Data Analysis of an Uncooled Run 39
Figure 18: Time History of Heat Transfer Coefficient (uncooled) 40
Figure 19: Consistency Check of Uncooled Analysis 41
Figure 20: Sample Time History of a Film-cooled Run 43
Figure 21: Film-cooled Data Analysis (cold coolant) 44
Figure 22: Heat Transfer Coefficient vs. Time (cold coolant) 45
Figure 23: Sample Time History of a Film-cooled Run (ambient coolant) 135
Figure 24: Data Analysis of an Ambient Coolant Run 136
Figure 25: Time History of Heat Transfer Coefficient (ambient coolant) 137
Figure 26: Uncooled Heat Transfer Coefficients Along Blade 47
Figure 27: Comparison of Experimental and Predicted $T_{o,\infty}$ - T_r 49
Figure 28: Cold Coolant Heat Transfer Coefficients Along Blade 52
Figure 29: Effect of Experimental vs Predicted $T_{o,\infty}$ - T_r on eta 54
Figure 30: Comparison of Trends in h (uncooled, cold coolant) 55
Figure 31: Ambient Coolant Heat Transfer Coefficient Along Blade 141
Figure 32: Comparison of Heat Transfer Coefficient (cold, ambient) 142

List of Tables

Table 1: Tunnel Parameters 3
Table 2: Comparison of Uncooled vs. Film-cooled Heat Transfer Coefficient 56
Table 3: Test Matrix for Ambient Coolant Experiments 139
Table 4: Comparison of Ambient and Cold-Film Cases 143

Nomenclature

C	Specific heat (J/kg·K)
G	Gain (V/V)
L_{ss}	Total length of the suction side of the blade (m)
M	Mach number (-)
P	Pressure (psig)
R	Resistance associated with the temperature sensor of the HFM (Ohms)
Re	Reynolds number based on chord length of blade (-)
S	Sensitivity associated with the HFM (μV/W/cm^2)
T	Temperature (°C)
V	Voltage (volts)
h	Heat transfer coefficient (W/m^2·°C)
k	Thermal conductivity (W/m^2·°C)
q	Heat flux (W/cm^2)
r_c	Recovery factor for high-speed flows (-)
s	Distance measured from the stagnation point to some point along the suction side of the blade (m)
u_∞	Velocity at the edge of the film layer (m/s)

Greek

α	Thermal diffusivity (m^2/s)
δ	Material thickness through which heat is conducted (m)
η	Adiabatic film cooling effectiveness (%)
ρ	Density (kg/m^3)

Subscripts

HFS	Associated with the heat flux sensor
KUL	Associated with Kulite pressure transducers

R	Recovery
RTS	Associated with the resistance temperature sensor
aw	Adiabatic wall
c	Coolant
o	Total
w	Turbine blade wall

Chapter 1: Introduction

There is a strong demand in today's gas turbine industry for higher turbine inlet temperatures. The reason for this push is that higher turbine inlet temperatures can be translated into improved turbine efficiency.

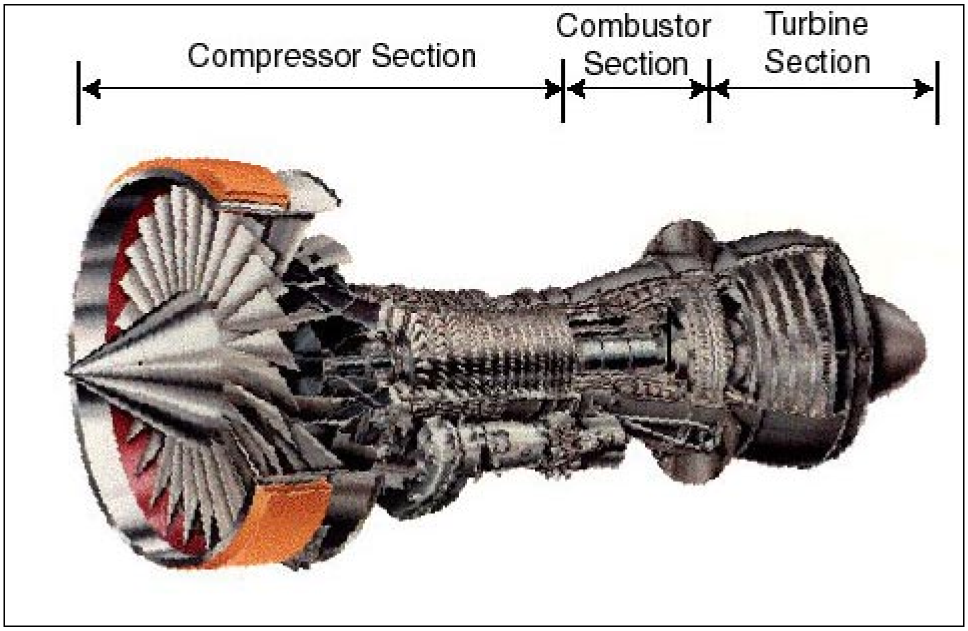


Figure 1: Typical Turbofan Engine

However, turbine inlet temperatures are limited by the thermal loads that the turbine blades are capable of withstanding. At present, the inlet temperatures to gas turbines are pushing the limits of the materials used in their construction. In order to maintain safety and to prolong the life of turbine blades, it has become necessary to protect them from the hot engine environments. One of the ways in which turbine blades are protected is through the use of film cooling. In film cooling, cool air is forced through holes on the surface of the blade. This air then forms a thin, protective layer between the hot combustion gases and the surface of the blade.   Because the air needed to cool the blades is drawn directly from the compressor section of a turbine, it is seen as a loss of work. Therefore, it is necessary to develop an understanding of how film cooling affects heat transfer into the blade in order to minimize the amount of air withdrawn from the compressor.

At present, analytical methods are limited in providing this information because of the complexities of the flow fields and interactions taking place. For this reason, experimental measurements must be used to determine heat transfer coefficients and film cooling effectiveness. In turn, this information can be used as a baseline for future computational studies.

Some of the things that are known to affect cooling films and heat transfer on turbine blades are shock and wake passing caused by the stators just upstream of the rotors, compressibility, curvature of the blade, and the size, chord-wise orientation and inclination of the cooling holes. There are many studies performed on flat plates which look at the effects these parameters have on film cooling, but the results from these studies are not always applicable to real engine conditions. Also, none of these flat plate studies investigated the unsteady effects of shocks and wakes on film cooling. The Virginia Tech Transonic Wind Tunnel is capable of studying both the steady and the unsteady effects of freestream turbulence and shock wave passing on the film cooling layer and the heat transfer into turbine blades.

The intent of this thesis is to investigate the steady heat transfer coefficient and the effectiveness of a film cooled turbine rotor blade with realistic geometry in flow conditions similar to those found in a real engine. Table 1 shows the parameters which were used for the experiments. These values were chosen in order to simulate real engine conditions. This work will form the basis for future experiments which will look at the unsteady effects which shock wave passing and free stream turbulence have on the cooling films and heat transfer into the blade.

Table 1: Tunnel Parameters

Exit Mach No.	1.2
Exit Reynolds No.	5×10^6
Temperature Ratio ($T_{o,c}/T_{o,\infty}$)	2 ± 0.4
Pressure Ratio ($P_{o,c}/P_{o,\infty}$)	1.04
Freestream Turbulence Level	1 %

Chapter 2: Literature Review

In an attempt to provide information about the heat transfer into turbine blades which employ film cooling, many flat plate experiments have been performed. These experiments were easy to perform and allowed researchers to amass a great deal of information about the parameters that affect heat transfer into a film-cooled surface. A good compilation of the experimental work done on flat plates was put together by Goldstein (1971). These works were important beginnings, but the results can not describe actual engine flows because they lack curvature, cross- stream pressure gradients, and the highly accelerated flows seen on the pressure and suction sides of a turbine blade. The results of the flat plate work can provide insights into the heat transfer on the suction side, however, because the suction side flows are somewhat similar to flow over a flat plate.

Schwarz et al. (1990) investigated the effects of curvature and blowing ratio on the film cooling effectiveness of a convex surface, i.e. a surface similar to the suction side of a blade, and made comparisons to a concave surface. They found that curved surfaces had better film cooling effectiveness than flat plates. The reason for this is that the cross-stream pressure gradient acts counter to the lifting momentum forces of the jets and forces them to stay on or close to the surface of the blade. Also, Schwarz et al. found that at low blowing rates, film cooling is more effective on the suction side of the blade than on the pressure side. High injection rates on the suction side of the blade promote lifting of the coolant film from the surface and lessen the protection that film provides. This work was an intermediate step between flat plates and investigations of heat transfer into realistic blade geometries.

Most of the work done with realistic blade geometries were short duration experiments which captured the transient surface temperature history. For short periods, the assumption of one-dimensional, semi-infinite conduction into the blade is valid for heat transfer into the blade. The solution for this case is well known and can be used to determine the heat flux into the blade since the temperature history is known. Several experiments were performed using this method of obtaining the surface heat flux.

Horton et al. (1985) investigated the heat transfer into a realistic turbine blade profile mounted in a linear cascade. Thin-film resistance sensors were used in a short duration blow down tunnel to capture the transient surface temperature history. The exit Mach number was 0.94 and the freestream to coolant temperature ratio was 1.3, so some attempts at matching real engine conditions were made as well. Air was used as the coolant on the blades. Horton et al. were able to obtain measurements of the heat transfer coefficient at several locations along both the pressure and suction sides of the blade.

In 1995, researchers at G. E. Aircraft Engines performed heat transfer measurements on a film-cooled inlet nozzle guide vane (NGV) in a linear cascade with realistic flow conditions [Abuaf et al., 1995]. Thermocouples embedded in a thin-walled turbine blade were used to measure the transient surface temperature. The exit Reynolds number was on the order of $3x10^6$, the freestream to coolant density ratio was 2, and the blowing ratio was varied between 1.5 and 2.7. Air was used as the coolant in their cooling scheme. Heat transfer coefficient and effectiveness profiles were obtained for both sides of the blade, but both profiles displayed noise in the results immediately downstream of the cooling holes. Also in 1995, Ekkad et al. investigated the effects of freestream turbulence on the heat transfer into a film-cooled turbine blade. Thermotropic liquid crystals (TLC's) were used to obtain the transient surface temperature history. TLC's have an advantage over gauges in that they provide information over the entire surface to which they are applied, and not just to discrete locations. They concluded that high freestream turbulence leads to an increase in heat transfer coefficient.

Researchers at Oxford University used thin-film gauges to study the heat transfer into a heavily film-cooled NGV [Guo et al., 1996 and 1997]. There measurements were performed in an annular cascade with attempts made to simulate realistic engine conditions (13% freestream turbulence, $Re \cong 2x10^6$, $M \cong 0.96$). A mixture of SF_6 and Argon was used as a coolant. The heat transfer coefficient and film cooling effectiveness profiles were determined for the pressure side of the NGV.

In 1998, Drost et al. used TLC's to record the transient surface temperature history of an NGV airfoil exposed to a step input in heat transfer while in a linear cascade. They used a foreign gas as a coolant to obtain a realistic density ratio. They concluded that the film cooling effectiveness was higher near the cooling holes and that the mainstream turbulence level had a weak effect on suction side effectiveness, but that it increased suction side heat transfer coefficients.

All of these experiments were performed in such a way that the heat flux was measured indirectly using the transient surface temperature history. Researchers at Virginia Tech are using a thin-film gauge which is capable of direct measurements of both the local heat flux and surface temperature simultaneously [reference Drew, Holmberg, Doughty]. This gauge, the HFM-7 is the newest version, will be used in the present work to investigate heat transfer into a film-cooled transonic turbine inlet NGV.

Chapter 3: Experimental Setup and Instrumentation

The experiments that were investigated in this study were all performed in the Virginia Tech Transonic Wind Tunnel. This tunnel is an intermittent blow-down facility with an open discharge, and it can be used for both transonic and supersonic testing. The compressor system feeds a large volume of pressurized air into a storage tank, and on-off valves determine whether flow is directed into the supersonic or the transonic facilities. A fast-acting control valve regulates the release of the pressurized air into the test section of the transonic portion of the tunnel.

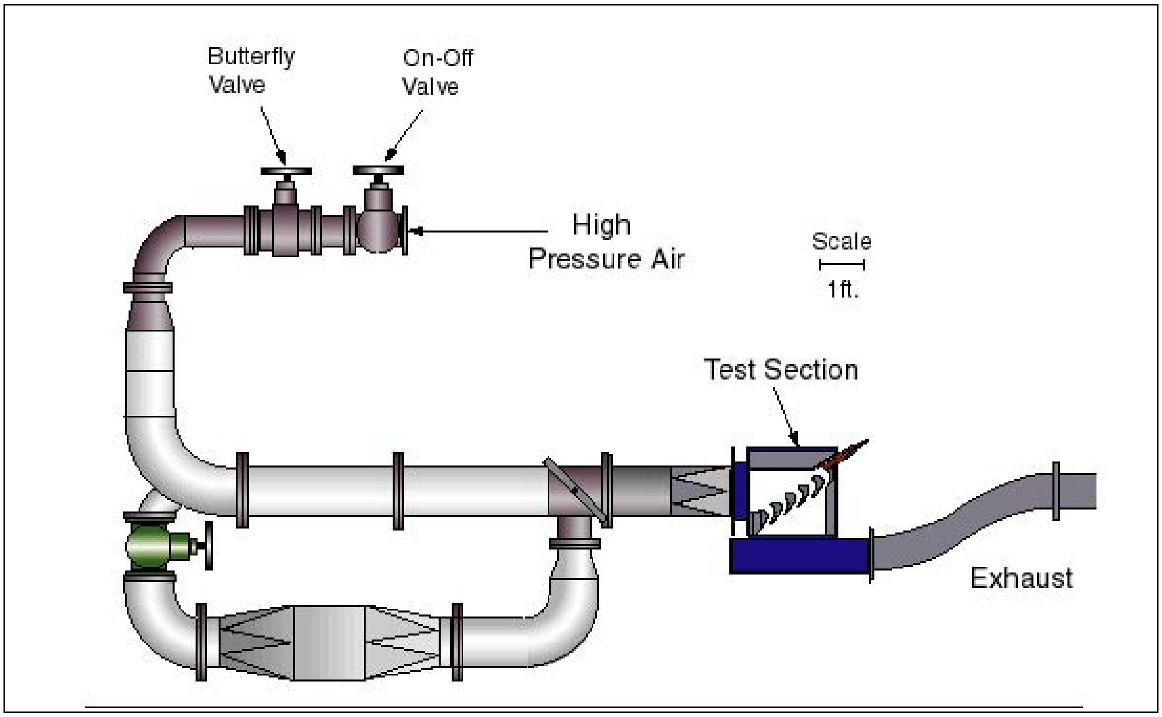


Figure 2: Schematic of Transonic Cascade Wind Tunnel

3.1 Cascade Wind Tunnel Compressor System

The compressor, which feeds the cascade wind tunnel, is an Ingersoll-Rand model HHE 334 four-stage, reciprocating compressor that is capable of pressurizing air up to 750psig. The pressurized air is circulated through a dryer to remove moisture and fed into a storage tank. Measurements have shown that the relative humidity of the air leaving the dryer is less than 7%. The tank is connected to the cascade wind tunnel by a system of 14” diameter, Schedule 30, carbon steel pipes. This system of pipes also includes an on-off valve and a fast-acting control valve. The control valve is a butterfly valve that is regulated by a feed-forward program that monitors the tunnel’s total pressure and temperature as well as the tank pressure, and then calculates the optimum valve position to maintain the desired flow conditions. The control program attempts to maintain the total pressure at the inlet of the test section, and thus the test section's inlet Mach number, to within 10% of the set point. This program, named WINDN3.C, was written in Turbo-C and is stored on the tunnel control computer. The mass flow rate of air through the tunnel is approximately 10 kg/s and the typical run time is 30 seconds**.**

3.1.1 Heated Flow

Some of the experiments performed in this study require that the mainstream flow be much hotter than ambient conditions in order to simulate the temperature ratios that might be seen in actual engine conditions. The cascade wind tunnel has a closed loop heating system, which allows for such heated mainstream flows. A schematic of the system is shown in Figure 3.

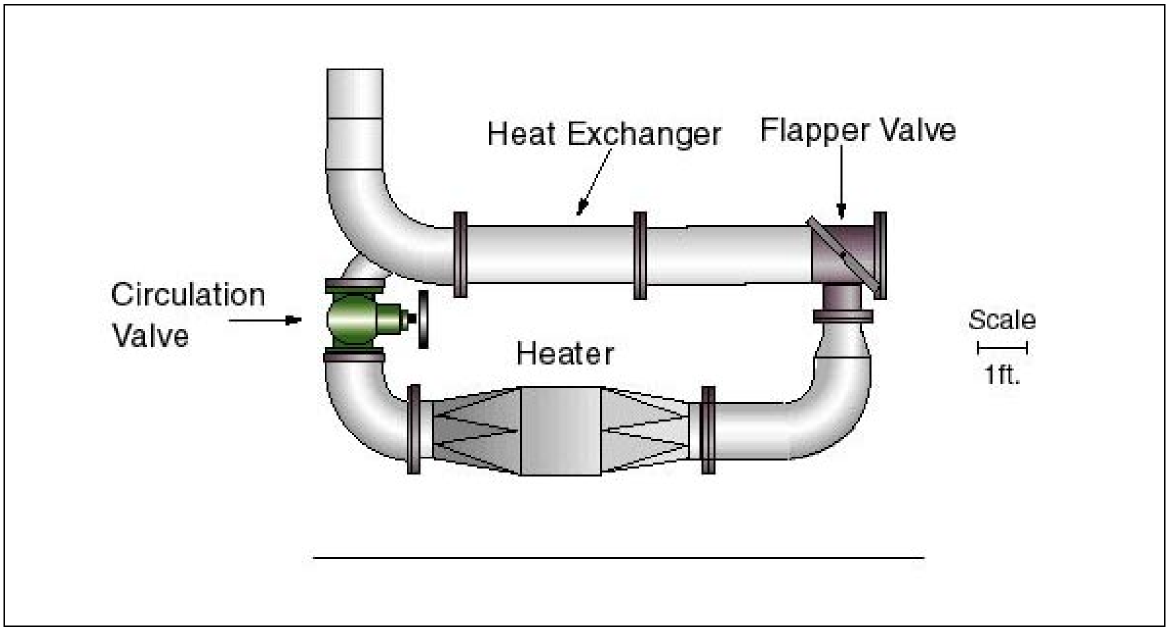


Figure 3: Circulation Loop for Heated Flows

The heating system is an insulated loop of pipe closed off from the tunnel supply and the test section by three valves. The circulation valve is opened in order to begin preheating. Once the flapper valve and the upstream on-off valve are shut, two 36kW heaters are turned on and supply energy to the air. A fan forces the heated air to circulate through the loop and over a bank of copper tubes. There are 350, one-meter-long, copper tubes in the bank. A thermocouple mounted in the flow monitors the temperature of the air. The air in the loop is heated up to 20°C above the desired starting temperature and allowed to reach equilibrium before the two valves are opened and testing is done. When air flows over the bank of copper tubes, it acts as a heat exchanger and transfers energy to the mainstream flow. For the experiments in this investigation, the desired starting total temperature for heated flows was 100°C and there was typically a drop in temperature of 70°C over the course of a run.

3.1.2 Coolant Supply System

The coolant supply system which supplies air to the film cooling holes is a separate, smaller version of the system that supplies air to the mainstream flow. A schematic of the coolant supply system is shown here in Figure 4.

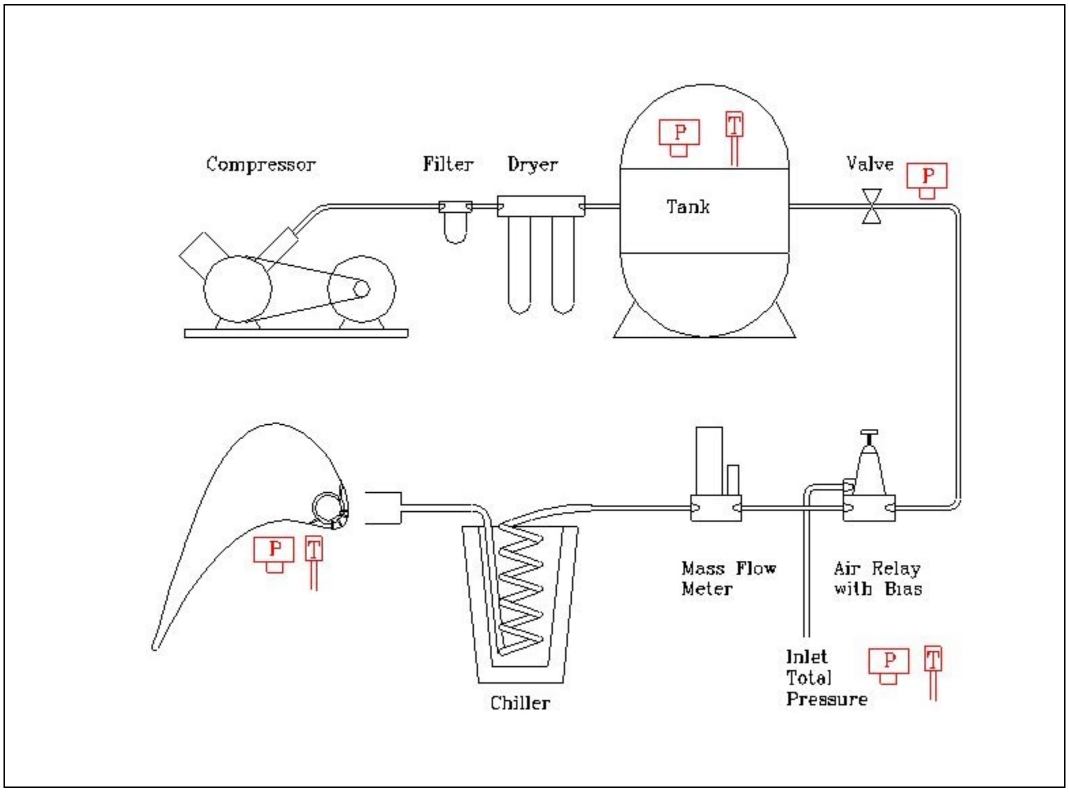


Figure 4: Schematic of Coolant Supply System

An Ingersoll-Rand 5Hp, two-stage, reciprocating compressor is used to supply pressurized air up to 150 psig at a flow rate of up to 25 SCFM. This air then flows through a dryer that removes the moisture from the air before it is stored in a large tank. Hereafter, this tank will be referred to as the coolant supply tank to distinguish it from the tank used to provide the mainstream flow. The relative humidity of the air leaving the dryer is less than 4%. A Marsh-Bellofram type 72 air relay is used to regulate the pressure of the air released from the storage tank. The air relay uses the mainstream flow’s total pressure measured just upstream of the test section as a reference and offsets it appropriately so that the desired plenum to tunnel total pressure ratio may be maintained. A plenum to mainstream pressure ratio of 1.04 was used in this work. The mass flow of the air supplied to the plenum is measured using an orifice plate with a calibration provided by the manufacturer. Usually, the mass flow was approximately 15 g/s. A heat exchanger, which uses liquid nitrogen as the coolant, is employed to lower the temperature of the air entering the coolant holes of the blade.

3.2 Test Section

The test section that contains the turbine blades investigated in this study was made from aluminum. Clear, acrylic windows were used for the endwalls so that flow visualizations could be made through optical means such as shadowgraphs and Schlieren photographs. Shadowgraphs can show extreme gradients in density such as those caused by shocks; Schlieren photography can capture weaker density gradients, and so can give a better picture of flow phenomena within the boundary layer. The actual test section and a schematic are shown side by side in Figure 5.

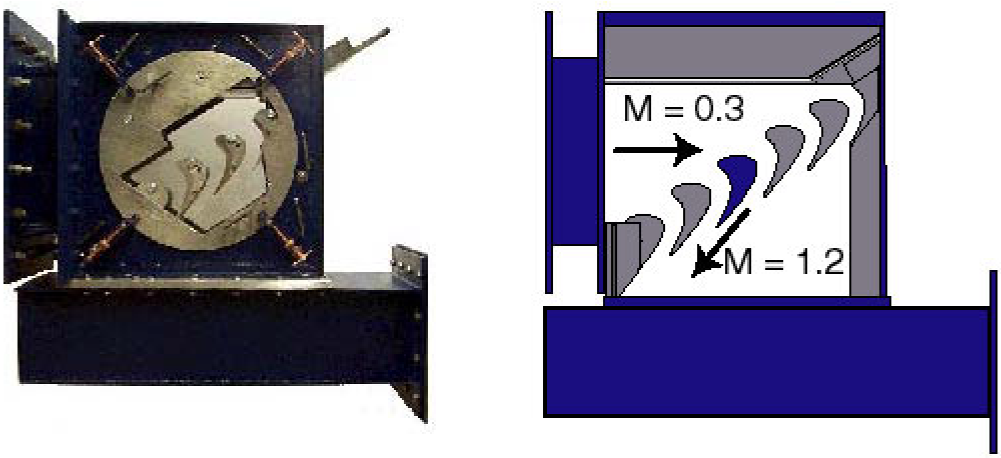


Figure 5: Picture and Schematic of Test Section

The test section is located just downstream of a nozzle that accelerates the flow up to Mach 0.3 prior to entering the test section. After entering the test section, the flow then passes over a set of turbine inlet rotor blades. There are four full blades and two half-blades contained in the test section; a total of five flow passages. The blade that was instrumented and used in this study is the second full blade from the bottom left in Figure 5, and it is shown darkened. The flow accelerates up to as much as Mach 1.2 as it passes over the blades and exits the blade passages. The blades are fixed into the endwalls at a fixed incidence angle of 58 degrees.

3.2.1 Instrumented Blade

The turbine blades used in this work are manufactured according to a generic, high-turning rotor blade design donated by General Electric. A showerhead cooling scheme is employed near the leading edge of the instrumented blade for film cooling. The entire blade can be seen in Figure 6.

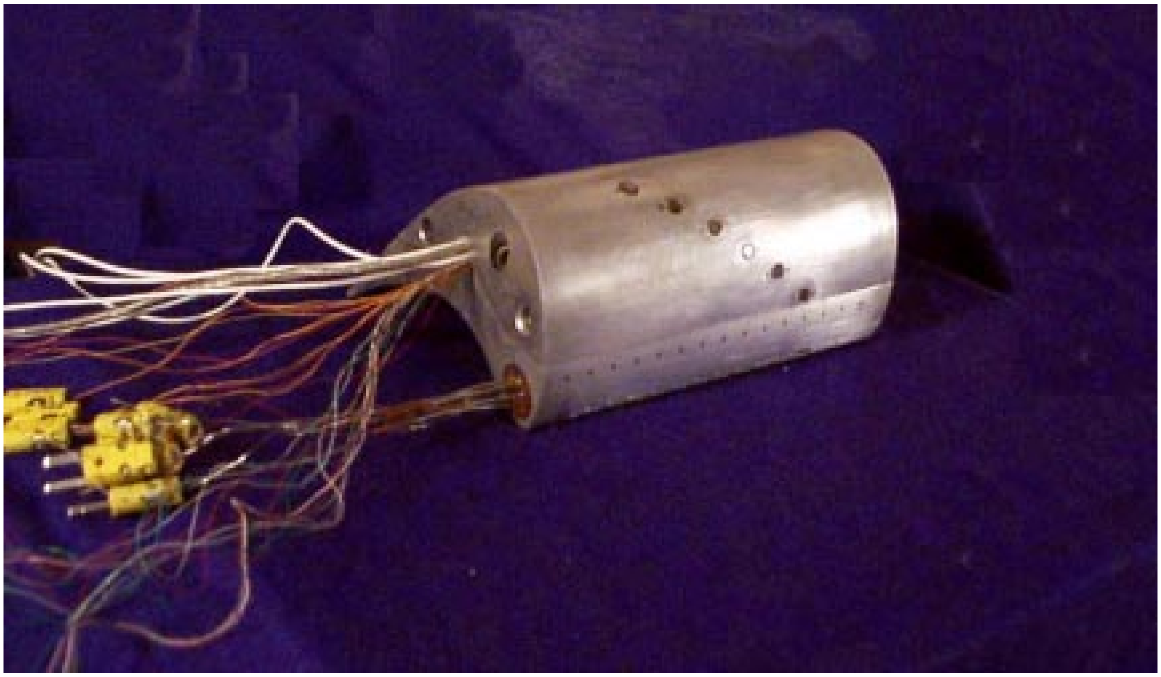


Figure 6: Picture of Entire Instrumented Blade

Six rows of cooling holes are located near the leading edge of the blade. Beginning approximately 25 mm after the last row of cooling holes on the suction side, six sets of gauges were installed on the blade. Each set of gauges is composed of a heat flux microsensor (HFM), a surface thermocouple, and a Kulite pressure transducer. In order to install all of this instrumentation without unnecessary disruption of the blade surface the blade was made in two pieces, a portion of the suction side and a pressure side. This method of manufacturing allowed the external blade profile to be maintained while allowing room for the wiring of instrumentation within the blade.

It is important to maintain the external profile of the blade in order to avoid affecting the air as it flows over the surface of the blade. The blade has a six-inch span and a chord length of 4.5 inches. With such a large blade, it was possible to design the two pieces of the blade such that they joined in areas where there was not a lot of curvature. The two pieces were manufactured separately using a wire EDM. The two junctions were polished to form a smooth surface. A schematic showing the arrangement and shape of the two pieces can be seen in Figure 7.

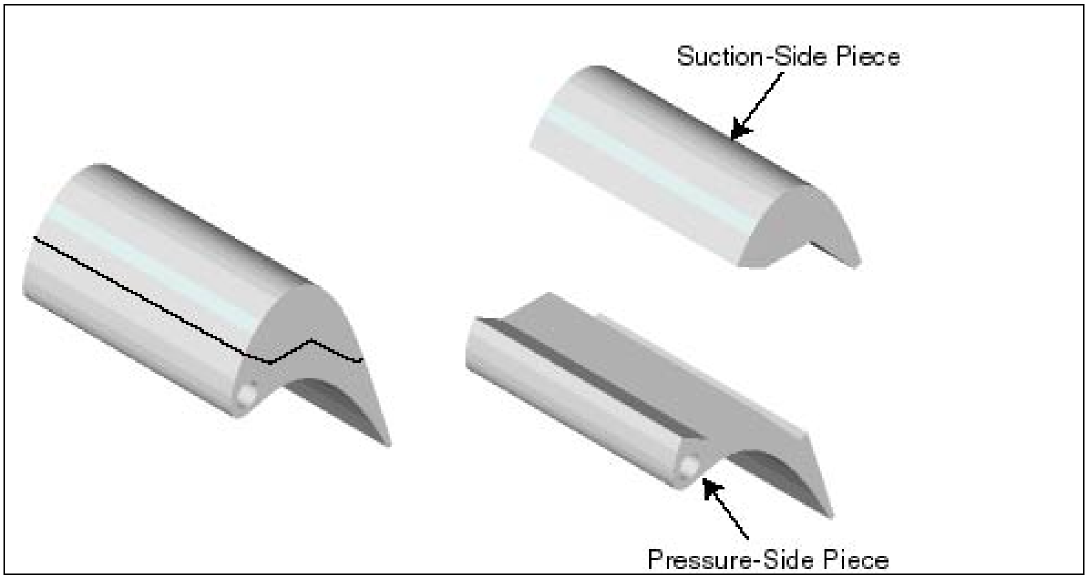


Figure 7: Schematic of Blade Manufacturing

All of the pressure side and a portion of the suction side are contained in the lower piece of the blade, which will hereafter be referred to as the pressure-side piece. The other piece of the blade, containing the rest of the suction side, will be referred to as the suction-side piece.

3.2.1.1 Pressure Side

The pressure-side piece of the blade contains the plenum and all of the coolant holes that produce the film. A picture of the pressure side can be seen in Figure 8.

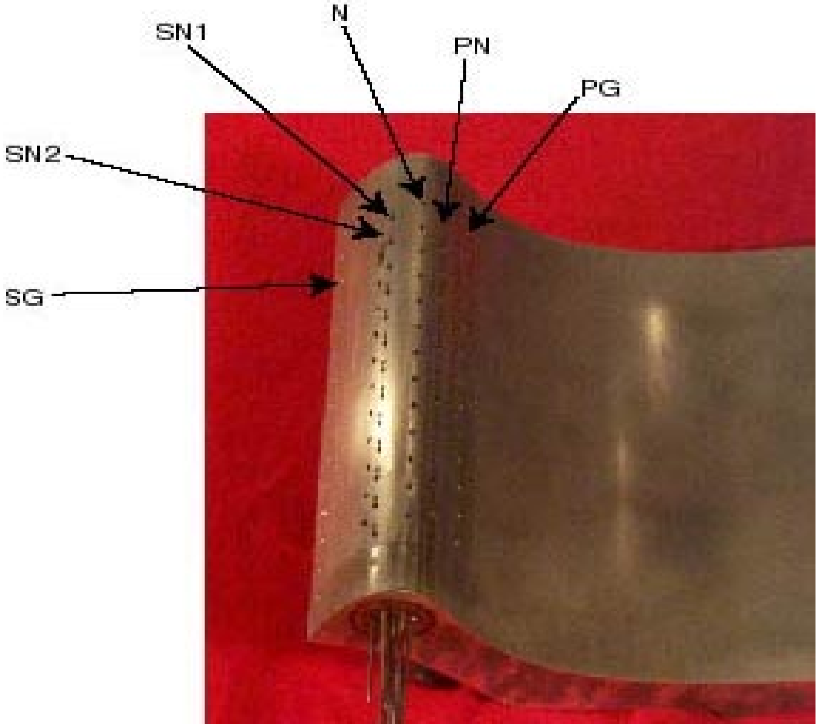


Figure 8: Pressure-side Piece of Instrumented Blade

A plenum designed for minimum pressure change across its length is used to supply air to the cooling holes. The plenum is insulated with polyethylene sulfone (0.12 W/m K) so that the cold air used for cooling does not affect the blade temperature which, in turn, could affect the air temperature. The cooling holes are staggered and discrete, which is similar to the blade cooling schemes found in actual engines. The four rows of holes nearest the leading edge all have 90 and 30-degree streamwise orientation and spanwise inclination angles, respectively. The last two rows of holes, one on either side of the pressure side piece, are in the streamwise direction but have 45-degree inclination angles. A schematic of the alignment of the cooling holes can be seen in Figure 9.

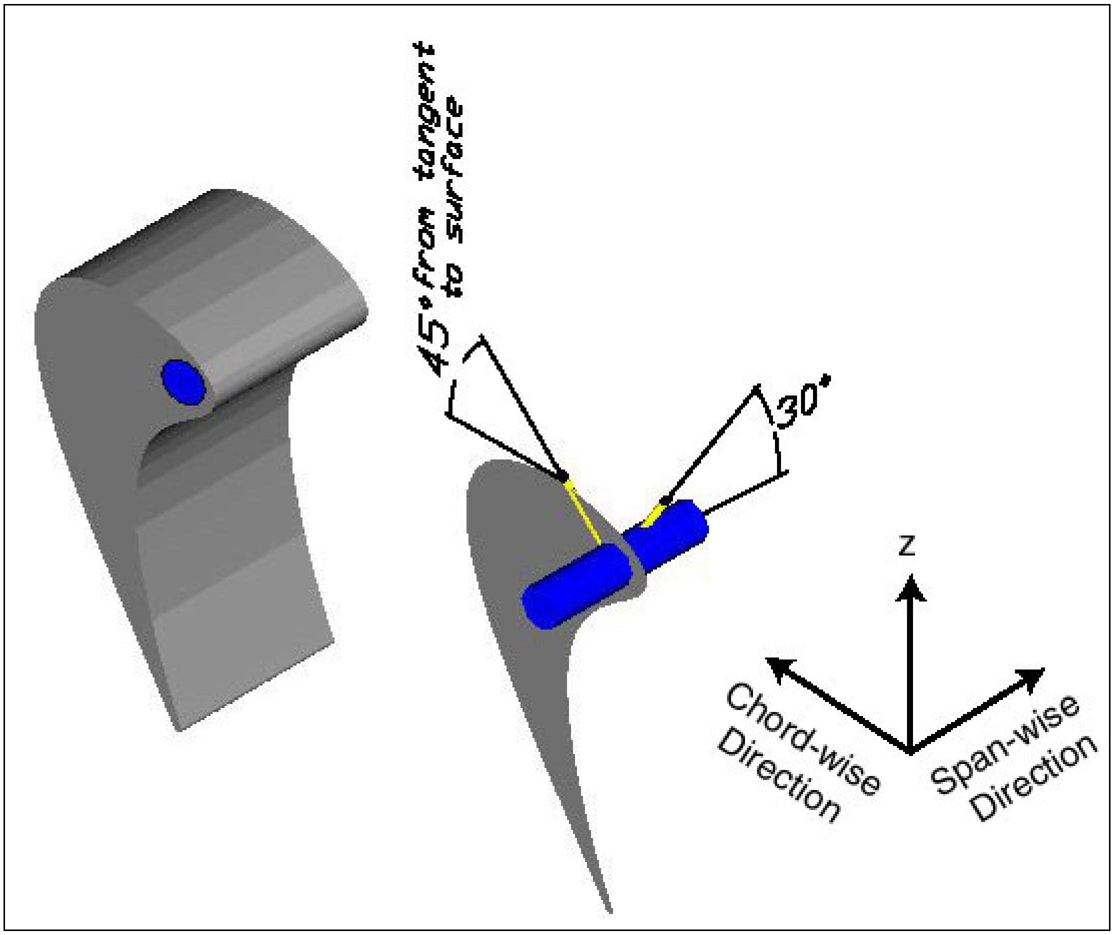


Figure 9: Schematic of Coolant Hole Alignment

Thermocouples are inserted into the second hole from the far end wall on each row of cooling holes in the blade in order to measure the temperature of the coolant. The thermocouples are of the type K, and were ordered from Omega. A static pressure tap is located in the hole nearest the far end wall of each coolant row so that the local freestream Mach number can be determined.

3.2.1.2 Suction Side

The suction-side piece contains the majority of the instrumentation used on the blade. Six triplets of gauges are staggered span-wise and chord-wise along its surface. This arrangement can be easily seen in Figure 10.

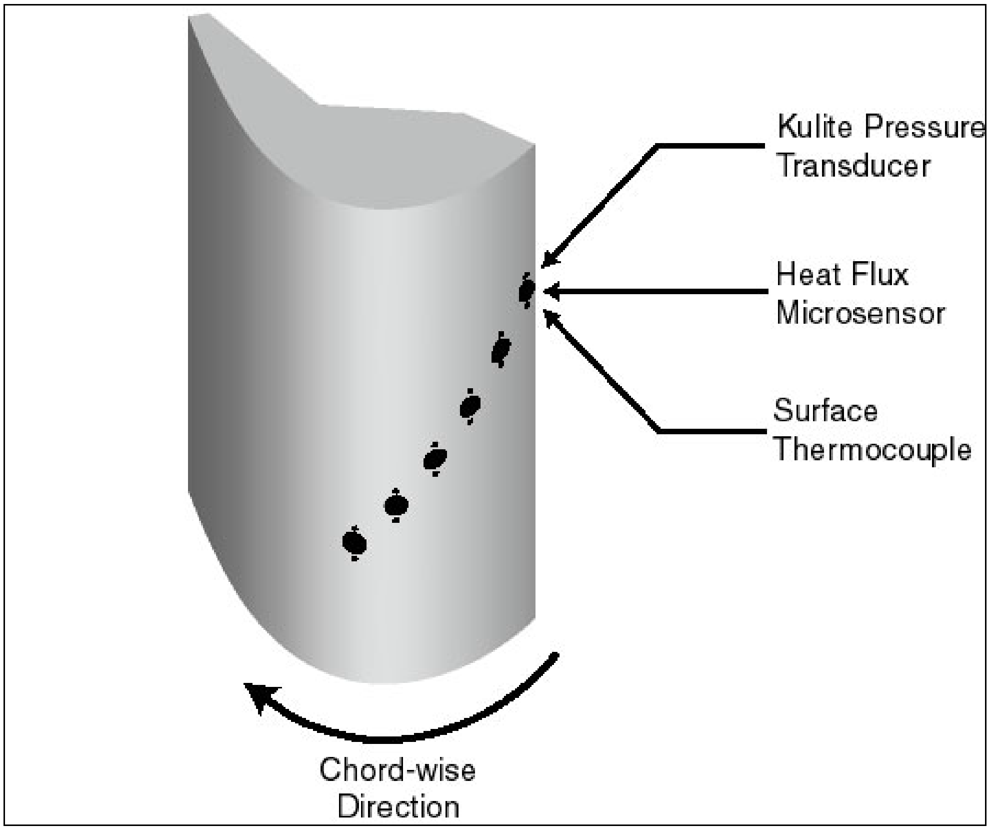


Figure 10: Suction-side Piece of Instrumented Blade

Each triplet of gauges is composed of a HFM, a surface thermocouple and a Kulite pressure transducer. All of the instrumentation was pressed into the blade so that there would be a good thermal contact between the blade material and the gauges. Using the six-inch blade, the curvature on the suction side is not extreme and installing the gauges made only very small disruptions to the surface. However, to avoid the downstream effects of the previous gauges, subsequent gauges are staggered in the span-wise direction. The sets of gauges were also staggered chord-wise so that the variation of heat transfer and effectiveness could be seen along the blade.

3.3 Instrumentation

3.3.1 Data Acquisition

Data was taken as input on a National Instruments' AT-MIO-16XE-50 acquisition board. This board can sample at rates of up to 20,000 samples per second, but has only eight differential channels. In order to record all of the data that were required for this study, two National Instrument's AMUX-64T analog multiplexers were used to expand the number of channels that could be sampled. A Labview program was written in order to set data acquisition parameters and to record the data to a file.

3.3.2 Heat Flux Microsensor

All heat flux measurements in this work were taken using a commercial heat flux microsensor (HFM) produced by Vatell Inc. The HFM-7, the gage used in these experiments, is composed of two separate sensors: a thin-film resistance temperature element and a heat flux microsensor. The HFM-7 was chosen because future work on this project will involve investigating the effect of shock wave interaction with the film-cooling boundary layer. The entire shock event is only about 50 μs, so the gage chosen had to have a very fast time response. The HFM-7 was used because it is a thin-film sensor with a time response on the order of 10μs, so it is capable of capturing the effect of shock wave passing on heat transfer and blade temperature.

Blade temperature is measured by the HFM-7 using a resistance temperature sensor (RTS) which is sputtered onto the surface of the gage. The RTS is a thin-film element that is only 2 microns thick and is deposited in a circular pattern around the heat flux sensor. The RTS operates on the principle that the resistance of metals changes in a predictable way with temperature. A current of 0.1 mA is used to excite the RTS. Over a range of temperatures from 0°C – 250°C, the change in resistance is nearly linear. Outside of this range, the change in resistance can still be estimated using a cubic polynomial of the form,

$$T(R) = a \cdot R^3 + b \cdot R^2 + c \cdot R + d \qquad (1)$$

Where R is the measured resistance in ohms
a, b, c, and d are the coefficients of the polynomial

The polynomial coefficients are determined through a calibration performed by the manufacturer. The coefficients for the gauges used in this work can be seen in Appendix C. The experiments performed in this work were all within the range of temperatures for which the change in resistance is still linear, so only the linear coefficients c and d were used for the RTS calculations. Changes in temperature are dependent upon changes in resistance, so it is important that both the initial temperature, T_o, and the initial resistance, R_o, be known. T_o is measured using a thermocouple on the blade, and R_o is calculated using the equation,

$$R_o = e \cdot T_o + f \qquad (2)$$

where e and f are coefficients determined from the calibration of the RTS. The resistance of the RTS at other temperatures is then calculated by,

$$R = \frac{V_{RTS}}{I_{RTS} \cdot G_{RTS}} + R_o \qquad (3)$$

Where V_{RTS} is the output of the RTS in µV
I_{RTS} is the excitation of the RTS in amps
G_{RTS} is the gain of the amplified RTS signal

Temperature is then calculated using equation (1). This temperature is used when calculating the heat flux.

The heat flux sensing element (HFS) is used to measure the flow of heat into and out of the blade. The HFS is a passive differential thermopile sensor made up of 280 thermocouple pairs. Each of the thermocouple pairs consists of a Nichrome/Constantan thermocouple junction. Although the voltage output of this particular thermocouple junction is small, it was chosen because it can withstand temperatures up to 1000°C. The thermocouple junctions were deposited in a serpentine pattern on the surface of the gage. The HFS operates on the principle that the heat flux through a material, assuming one-dimensional conduction and steady state conditions, is proportional to the temperature difference across the material. This principle is illustrated in Fourier's equation,

$$q = \frac{k}{\delta}\left(T_1 - T_3\right) \qquad (4)$$

Where q is heat flux (W/cm^2)
k is the thermal conductivity (W/cm^2 C)
δ is the thickness (cm)
T_1 is the substrate temperature (°C)
T_3 is the insulated surface temperature (°C)

A schematic of one of the thermocouple junction pairs in the HFS can be seen in Figure 11.

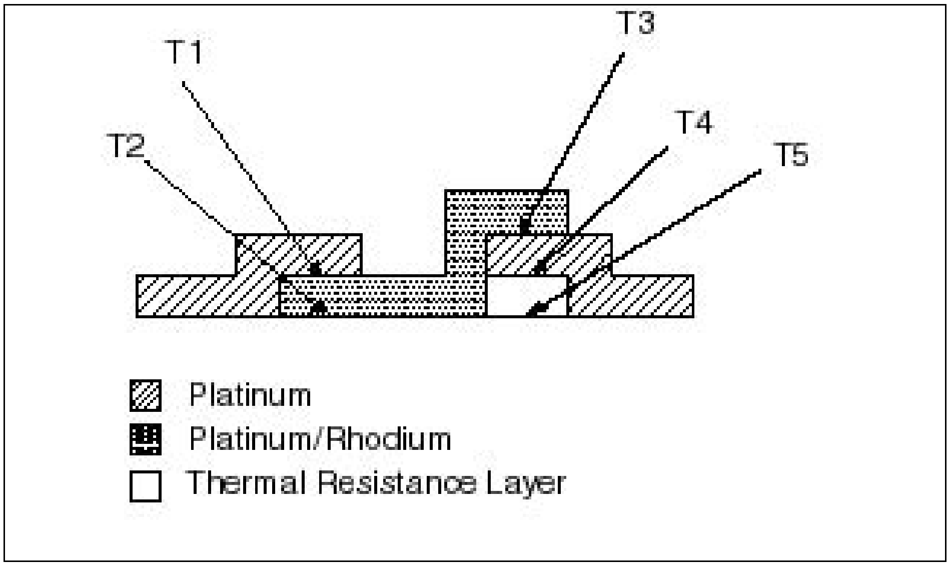


Figure 11: Schematic of HFS Thermopile

One junction of each thermocouple blade is sputtered onto the aluminum nitride substrate and a thin layer of material with a low thermal conductivity insulates the other junction. When heat flows into the gage, this thermal resistance causes the insulated junction to reach a higher temperature than that of the junction deposited directly onto the high conductivity substrate. It is assumed that the high conductivity substrate has a uniform temperature (T2 = T5). The temperature difference across the thermal resistance layer is proportional to the voltage output across the two thermocouple junctions. Heat flux into the blade will result in a positive voltage, and flux out of the blade will cause a negative voltage. The sensitivity of the HFS is itself dependent upon temperature, so the temperature measured by the RTS is used to correct the output of the HFS.

The heat flux that passes into the HFS sensor is calculated using the following equation,

$$q = \frac{V_{HFS} / G_{AMP}}{S_T} \tag{5}$$

Where q is the heat flux in W/cm^2
V_{HFS} is the output voltage in µV
G_{AMP} is the gain of the amplified signal
S_T is the temperature dependent sensitivity

S_T, the temperature dependent sensitivity, is determined during the calibration of the HFS. The sensitivity of the HFS varies linearly with temperature and is given by the equation,

$$S_T = g \cdot T + h \tag{6}$$

Where g is the change in sensitivity with temperature (µV/W/cm^2/°C)
h is the HFS sensitivity at room temperature (µV/W/cm^2)

The coefficients g and h in the equation above are usually provided by the manufacturer. For the gauges used in this work, however, the sensitivity coefficient at room temperature was found using a transient calibration technique which is described in Appendix A. The sensitivity coefficients for each gauge in this work are listed in Appendix C. The change in sensitivity of the HFS with temperature is small, usually on the order of µV/W/cm^2/°C, so the coefficient g was considered to have a negligible effect. The temperature used in the above calculation is determined from the RTS. Since both sensors are on the same gage and receive the same incident flux, the temperature recorded by the RTS is the result of the heat flux measured by the HFS.

3.3.3 Kulite Pressure Transducer

Kulite XCQ-062-50 high-frequency transducers were used to take some of the pressure measurements used in this study. This gage uses a diaphragm connected to a piezo-electric crystal to generate a voltage in response to pressure differences across the diaphragm. A small metal screen was used to prevent flying particles in the mainstream flow from damaging the transducer. The gages are 1.7mm in diameter and have temperature compensation.

Pressure is directly translated into voltage by the crystal, so fast time responses of up to 100 kHz are possible if the Kulite is unscreened and up to 25 kHz with the screen in place. The voltage output of the Kulite is linear in proportion to the pressure it experiences. The typical value is about 2 mV/psi. Pressure is calculated from the Kulite output using the following equation,

$$P = \frac{V_{KUL}}{S_{KUL} \cdot G} \tag{7}$$

Where P is the pressure in psig
S_{KUL} is the Kulite sensitivity in mV/psi
G is the gain of the amplified signal

All of the Kulites used in this work were amplified with Measurement Group 2310 amplifiers set on a gain of 100V/V.

3.3.4 Amplifiers

3.3.4.1 Amp-6

The voltage output signals from both the RTS and HFS sensors were amplified with the AMP-6 amplifier, which is produced by Vatell, Inc. for use with the HFM-7 gage. A single LEMO connection on the AMP-6 takes input from both of the sensors on the gage. The AMP-6 also supplies a constant excitation current of 0.1 mA to the RTS and has channel specific gains. The amplifier is allowed to warm up at room temperature before the outputs of the two sensors are “zeroed.”

In the case of the RTS, this means that a potentiometer is set to read the same resistance as that of the RTS at room temperature. Any changes in resistance caused by fluctuating temperatures are referenced to this “zeroed” resistance. The AMP-6 has gains of 1, 100, 200, and 500 for the amplified RTS signal.

For the HFS, another potentiometer is used as an offset to reduce the voltage output to zero. This is done when the gage and its surroundings are at the same temperature and no heat flux is incident on the blade. The AMP-6 has gains of 1, 100, 500, 1000, and 5000 for the HFS.

3.3.4.2 Measurements Group 2310

All measurements taken with the Kulite pressure transducers were amplified with a Measurements Group 2310 amplifier. The 2310 is a multi-channel amplifier designed for use with strain gages inputs—foil or piezo-resistive based. Excitation voltages can be provided from 0.5 Vdc up to 15 Vdc, and the gain of the output signal can be set from 1 to 11,000. The 2310 amplifier was allowed to warm up before the bridge was balanced.

3.4 Experimental Setup

3.4.1 Uncooled Blade

3.4.1.1 Overview

In order to understand how film cooling affects the turbine blade used in this work it was necessary to first determine the steady heat transfer coefficient profile in the streamwise direction without film cooling. Once obtained, this profile was used as a basis of comparison for the cases which are performed with film cooling present. This case (steady h w/o film cooling) was performed in the Virginia Tech Transonic Wind Tunnel, an intermittent blowdown facility which discharges to atmosphere. For these tests, a heated mainstream flow was used.

3.4.1.2 Setup

The instrumented blade described in section 3.2.1 was inserted into the test section, and the entire assembly was placed into the transonic wind tunnel. Although no film cooling was used during the actual run, the coolant supply system was connected to the plenum chamber of the instrumented blade. The coolant system was used to cool the blade down before performing experiments. Data was recorded on a computer.

3.4.1.3 Procedure

The instrumented blade was placed into the cascade test section in order to perform steady heat transfer measurements. The tunnel's mainstream air flow was heated up to 100ºC as described earlier in the section on heated flow. The coolant supply compressor system was started and used to pump air into the coolant supply storage tank until the tank pressure reached 170 psig. Liquid nitrogen was released into the heat exchanger until it was approximately half-full. The instrumented blade was cooled down to 0°C using cold air from the coolant supply tank which was allowed to flow through heat exchanger and into the plenum and cooling holes. Once the blade reached zero, the coolant supply system was disconnected and a plug was placed into the plenum of the instrumented blade. The plug prevented the mainstream flow from entering the plenum through the coolant holes nearest the leading edge and blowing out the holes that are farther away. The tunnel control program was started, and separate, but simultaneous, time resolved heat transfer and temperature measurements were taken. The data produced by the six sets of gages staggered in the chord-wise direction along the instrumented blade was recorded to be analyzed later.

3.4.2 Film-cooled Blade (Cold Coolant)

3.4.2.1 Overview

The steady heat transfer coefficient profile along a turbine rotor blade was determined in these tests, which were performed in the Virginia Tech Transonic Cascade Wind Tunnel facility. The heat transfer coefficient is used to help determine the chord-wise variation of the adiabatic film cooling effectiveness. These two pieces of information, h and η, will also be used in future experiments that investigate the unsteady effects of shock wave passing on the film cooling of turbine blades. A heated mainstream flow was passed over an instrumented turbine rotor blade that had a showerhead film-cooling scheme.

3.4.2.2 Setup

The instrumented turbine rotor blade was inserted into the aluminum test section and the test section was placed into the transonic blowdown facility. Film cooling was used in these tests, so the coolant supply system was connected to the plenum chamber of the instrumented blade. The coolant system was used to cool down the blade before performing these experiments.

3.4.2.3 Procedure

The instrumented blade was inserted into the test section of the linear cascade in order to perform steady heat transfer experiments with film cooling. The tunnel was preheated to 100°C as described in the section on heated flow. The coolant supply compressor system was started and used to pump air into the coolant supply storage tank until the tank pressure reached 170 psig. Liquid nitrogen was released into the heat exchanger until it was approximately half-full. The instrumented blade was then cooled down to 0°C using air from the coolant supply tank which was allowed to flow through the heat exchanger, become cold, and then flow into the plenum and cooling holes. The difference of the coolant total pressure in the plenum to the ambient total pressure was set to a nominal value of 2.4, using the air relay described in section 3.1.2. This was done before the tunnel's mainstream flow was started. This pressure difference between the coolant and the ambient air was determined through trial and error to give the desired coolant to freestream total pressure ratio of 1.04 once the tunnel's mainstream flow was started. Once these conditions had been reached, the tunnel control program was started and separate, simultaneous measurements of heat flux and temperature were taken. Data were collected from a set of six runs with film cooling.

3.4.3 Film-cooled Blade (Ambient Coolant)

3.4.3.1 Overview

Tests were performed in the Virginia Tech Transonic Wind Tunnel to determine the heat transfer coefficient profile generated on a turbine rotor blade with film cooling when the temperatures of the mainstream flow and the coolant are approximately the same. The heat transfer coefficient used in calculating convection heat flux is a proportionality constant that is used to separate the effects of temperature and flow phenomena on heat transfer. The results from these tests will give an idea of how good the assumption is that the effects of flow phenomena on convection heat transfer are contained within the heat transfer coefficient.

3.4.3.2 Setup

The instrumented blade was inserted into the test section and the entire assembly was placed into the transonic wind tunnel. The coolant supply system was connected to the plenum of the instrumented blade and used to cool down the blade before performing the tests. The storage tank was loaded up to 170 psig. The mainstream flow was not heated for these experiments.

3.4.3.3 Procedure

The instrumented blade was placed into the cascade test section in order to perform steady heat transfer measurements with ambient temperature air as the coolant. The tunnel's mainstream air flow was not heat and allowed to remain at ambient temperature in order to accomplish this. The coolant supply compressor system was started and used to pump air into the coolant supply storage tank until the tank pressure reached 170 psig. Liquid nitrogen was released into the heat exchanger until it was approximately half-full. Air was allowed to flow from the coolant supply storage tank, through the heat exchanger, and then into the plenum and out the cooling holes. Once the instrumented blade reached zero, the coolant supply system was temporarily shut down while the copper tubing which transported the cooling air from the heat exchanger to the plenum was warmed using a heat gun. After the condensation build-up had been removed and the tubing had been warmed, the heat exchanger in the coolant supply system was bypassed and ambient temperature air was sent to the plenum chamber of the blade. The tunnel control program was started, and separate, but simultaneous, time-resolved heat transfer and temperature measurements were taken. The data used in this study were collected from various gauges over four runs made with film cooling where the injected fluid was at the same temperature as the mainstream flow.

Chapter 4: Evaluation of Test Section

4.1 Overview

The blade geometry used in this study was a generic design which was donated by GE along with their analytical predictions of what the flow field surrounding the blade should be. Before any serious testing could be done, it was necessary to determine what the actual flow field looked like and whether it matched the prediction. Also, the extent to which the endwalls influenced the flow over the blades and knowing what the cooling film looks like are both important. To get this information, several flow visualization techniques were used.

4.2 Oil Flow Visualization

An oil flow visualization was used to capture the disturbed flow near the endwalls. In Figure 12 the boundary layer induced by the end-wall can be seen. The endwall vortices grow quickly along the blade, but the six-inch span of the blade used is large enough that the disturbed flow does not pass over the area where the blade has been instrumented. The blade was instrumented in a 3.5 inch interval centered on the middle of the blade. This lies between the streamlines in the oil flow visualization which indicate the locations of the end-wall vortices. Figure 12 shows that the heat transfer measurements taken in this study should not be affected by the disruptions near the endwalls

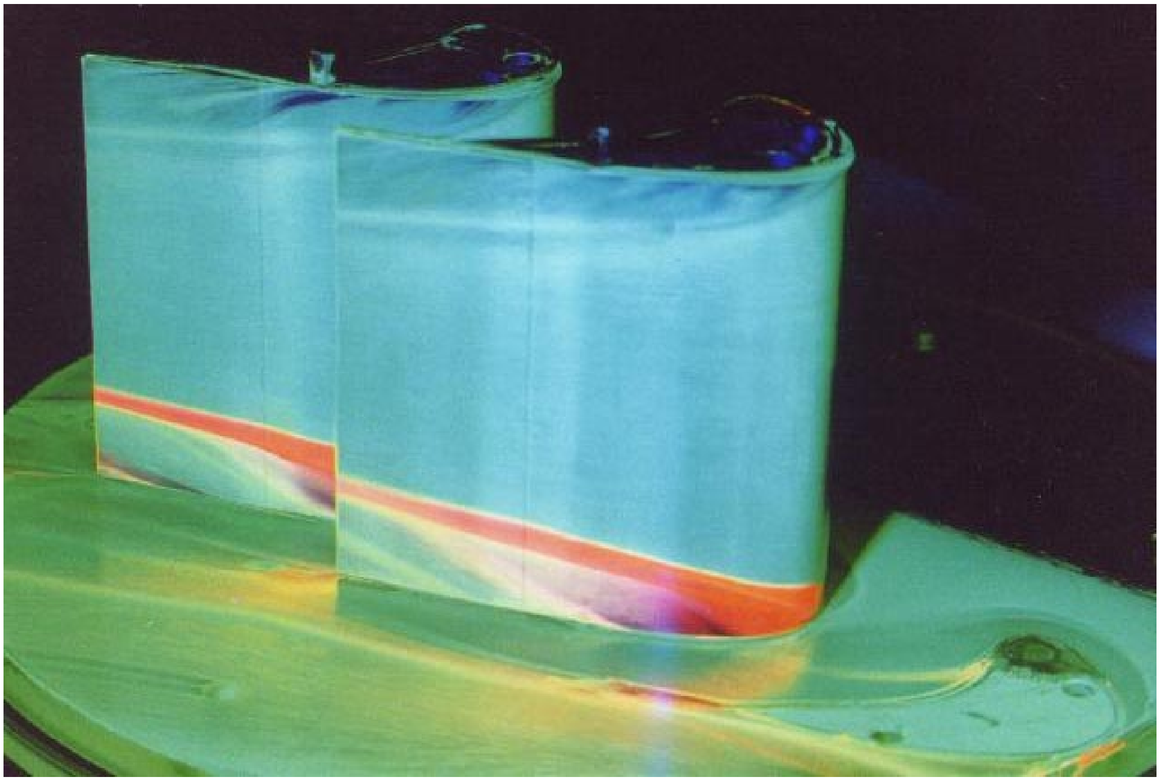


Figure 12: Picture of Surface Oil-flow Visualization

4.3 Schlieren and Shadowgraph Photography

Ensuring that the flow was periodic is also very important. To resolve this question, Schlieren photographs were taken of the blade passages. The test section contains four full blades and two half blades, or five flow passages. Figure 13 shows a Schlieren photograph taken of the passage above the suction side of the instrumented blade. This picture shows that there is periodicity in the blade passages and also displays the expected trailing edge shock. Several pictures were taken and the shock angle remained constant and consistent with a maximum speed of Mach 1.4 before the shock. This indicates that the tunnel facility is capable of producing very repeatable flow conditions.

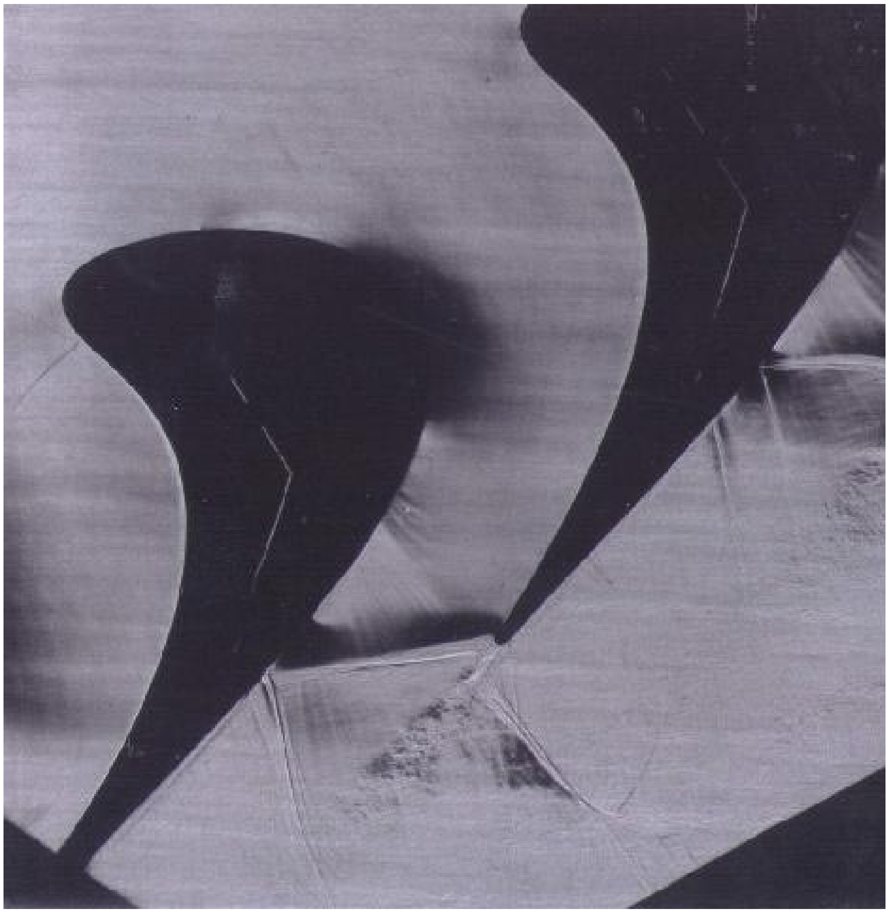


Figure 13: Schlieren Photograph of Flow in Blade Passages

Shadowgraph photography was used to examine the flow generated when the cooling system was employed. Figure 14 shows a shadowgraph of the instrumented blade’s leading edge and suction side with coolant injection into the boundary layer. From this picture it can be seen that the stagnation point on the leading edge forces the jets from coolant holes along the nose to fold over onto the pressure side of the blade. Therefore, the coolant film present on the suction side of the blade is the result of only three rows of coolant holes. This shadowgraph shows that there is definitely a film present on the suction side and that it is attached. Turbulent cell structures can be seen in the film cooling layer, but it should be noted that this is an average across the blade.

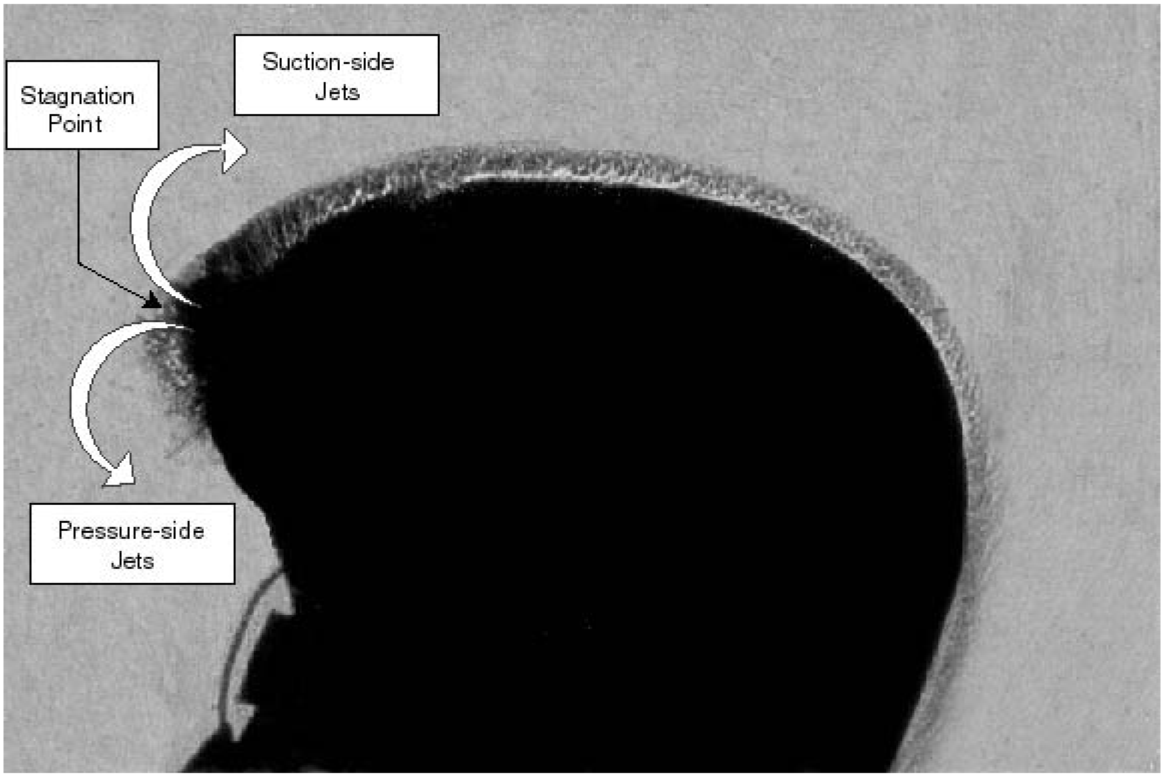


Figure 14: Shadowgraph of Cooling Film on Blade

4.4 Static Pressure Measurements

In order to check the actual flow field against the analytical prediction, static pressure measurements were taken using the Kulite pressure transducers which were spaced along the blade as was shown in Figure 10. Using these measurements it was possible to determine the chord-wise Mach number profile and to compare it with the predictions from GE. Figure 15 shows the actual Mach number profile along with the predicted values. The actual profiles are within 10% of the predicted profile at all locations.

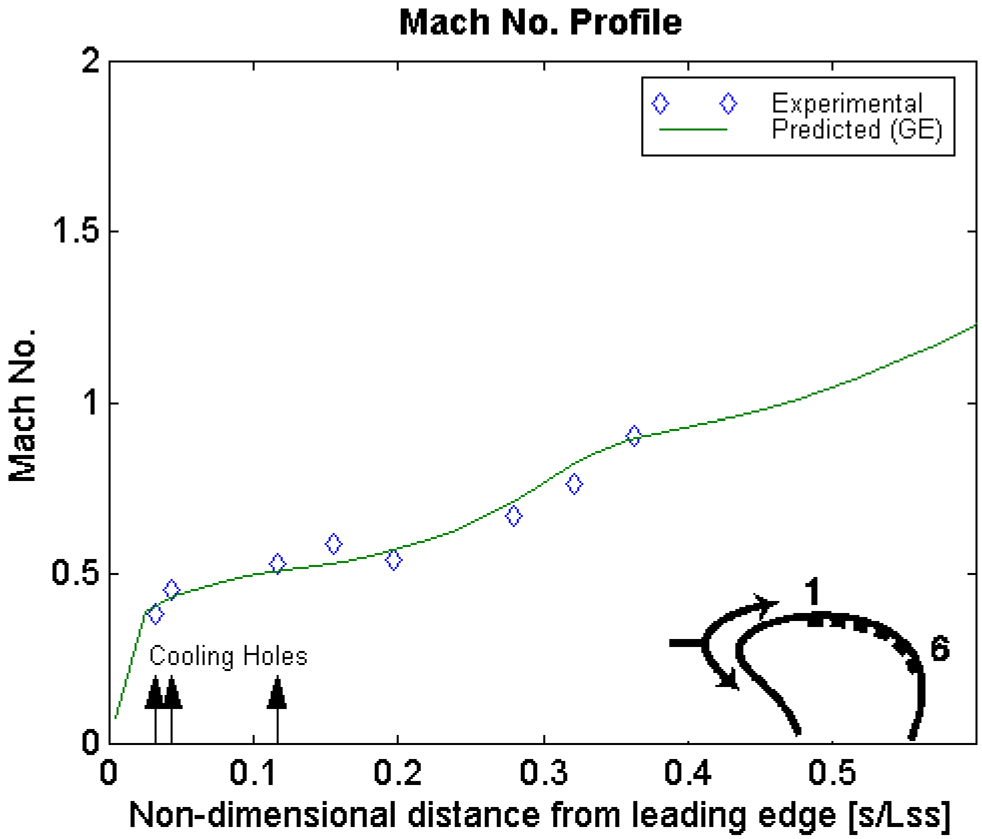


Figure 15: Mach Number Distribution Along Blade

Chapter 5: General Analysis

This study was performed in order to better understand the heat transfer taking place at the surface of turbine blades. An overview of the analysis used on the experimental data will be covered in this section. The dominant mode of heat transfer into the blade is convection, and the most important pieces of information are the steady heat transfer coefficient and the film cooling effectiveness. In low speed flows, convection heat transfer is characterized by the equation,

$$q = h \cdot (T_\infty - T_w) \tag{8}$$

where T_∞ = the freestream total temperature in ºC
T_w = the wall temperature in ºC
h = heat transfer coefficient in W/(m·ºC)
q = local heat flux in W/m

5.1 Uncooled Analysis

For high speed flows, such as those in turbines, the driving force of heat transfer is the recovery temperature and Equation (8) is rewritten as,

$$q = h \cdot (T_r - T_w) \tag{9}$$

where T_r is the recovery temperature. The recovery temperature can be calculated from the freestream total temperature using the equation,

$$T_r = T_\infty - (1 - r_c) \cdot \frac{u_\infty^2}{2 \cdot C_p} \tag{10}$$

where u_∞ = the freestream velocity in m/s
C_p = the specific heat of the air in J/(kg·K)
r_c = the recovery factor

The recovery factor for laminar and turbulent flows is usually considered to be $Pr^{1/2}$ and $Pr^{1/3}$, respectively [Kays and Crawford, 1993]. The difference between the total freestream temperature and the recovery temperature is,

$$T_{o,\infty} - T_r = (1 - r_c) \cdot \frac{u_\infty^2}{2 \cdot C_p} \quad . \tag{11}$$

In all of the experiments in this work, the time histories of both the local heat flux and the wall temperature are known for the duration of the run, as can be seen in Figure 16. The dashed lines in the figure enclose the data which were used in the uncooled analysis. Approximately twenty seconds worth of data sampled at 100 Hz, about 2000 data points, were used in the analysis. The only unknowns in Equation (9) are the heat transfer coefficient, h, and the recovery temperature, T_r. The trace labeled T_r in the figure is the time history of the recovery temperature calculated from the measured freestream total temperature. The time history of the recovery temperature at each gauge location was generated using $T_{o,\infty}$ - T_r and Equation (10). Equation (9) can be rewritten using equations (10) and (11) to give,

$$\mathrm{q} = \mathrm{h} \cdot \left[\left(\mathrm{T}_{\mathrm{o},\infty} - (\mathrm{T}_{\mathrm{o},\infty} - \mathrm{T}_{\mathrm{r}})\right) - \mathrm{T}_{\mathrm{w}}\right] \tag{12}$$

or,

$$\mathrm{q} = \mathrm{h} \cdot \left(\mathrm{T}_{\mathrm{o},\infty} - \mathrm{T}_{\mathrm{w}}\right) - \mathrm{h} \cdot (\mathrm{T}_{\mathrm{o},\infty} - \mathrm{T}_{\mathrm{r}}) \tag{13}$$

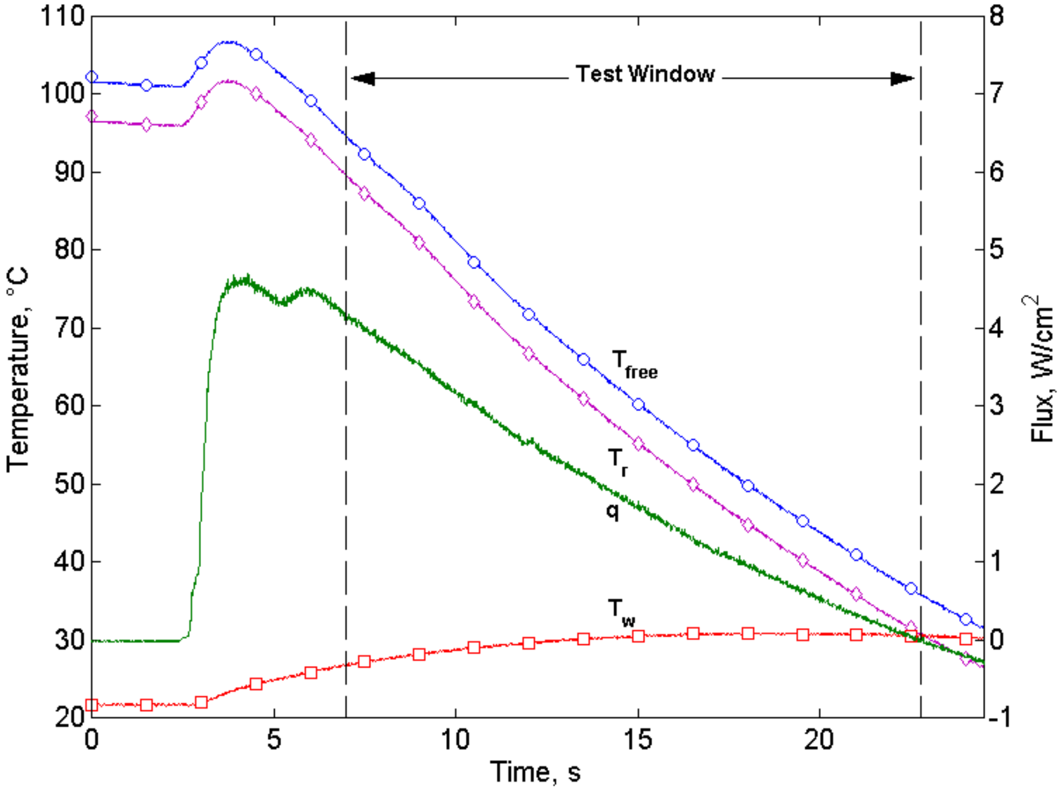


Figure 16: Sample Time History of an Uncooled Run

When the data from the time history of an uncooled run such as that shown in Figure 16 is plotted according to Equation (13), the result is a very linear curve in which the local heat flux is linearly dependent upon the difference between the freestream total temperature and the local wall temperature. A linear fit can be made through the data such that the slope of the line is the steady heat transfer coefficient and the intercept with the x-axis is the temperature difference, $T_{o,\infty}$ - T_r. An example of the data plotted in this manner can be seen in Figure 17.

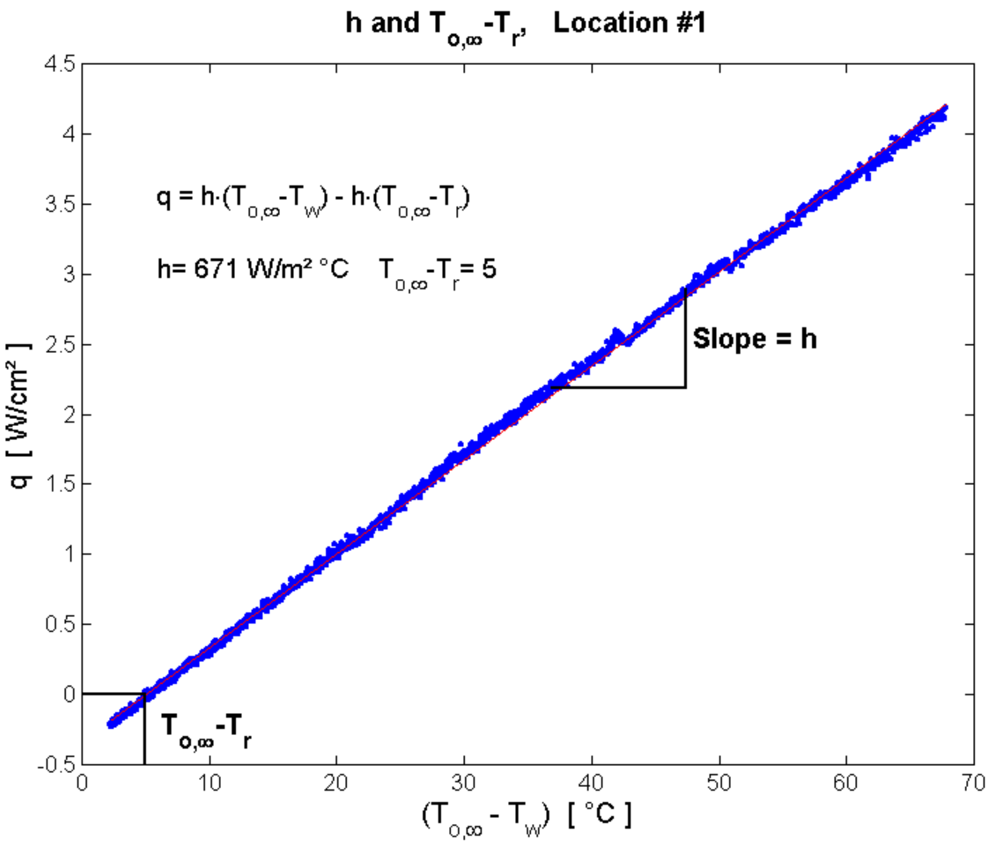


Figure 17: Data Analysis of an Uncooled Run

If the recovery temperature is known, the heat transfer coefficient can then be plotted as a function of time by using Equation (9). The time history of the recovery temperature is calculated by first assuming that the $T_{0,\infty}$ - T_r values determined in the analysis are constant over the run, and then by subtracting these values from the freestream total temperature. The assumption that the $T_{0,\infty}$ - T_r at every location is a constant over the run is supported by static pressure measurements which show that the Mach number distribution along the blade, and thus the flow field, remain constant throughout the run. Figure 18 is a typical plot of the time history of the heat transfer coefficient for uncooled runs. It can be seen that the heat transfer coefficient is constant over the course of the run and that the average value coincides with the value calculated from the linear fit. The magnitude of the oscillations about the mean value increases as the run time progresses. This increase in oscillation is due to increased uncertainty as the flux approaches zero and is caused by the decrease in difference between the recovery and blade temperatures; effectively, the analysis is dividing a large number by a relatively small number.

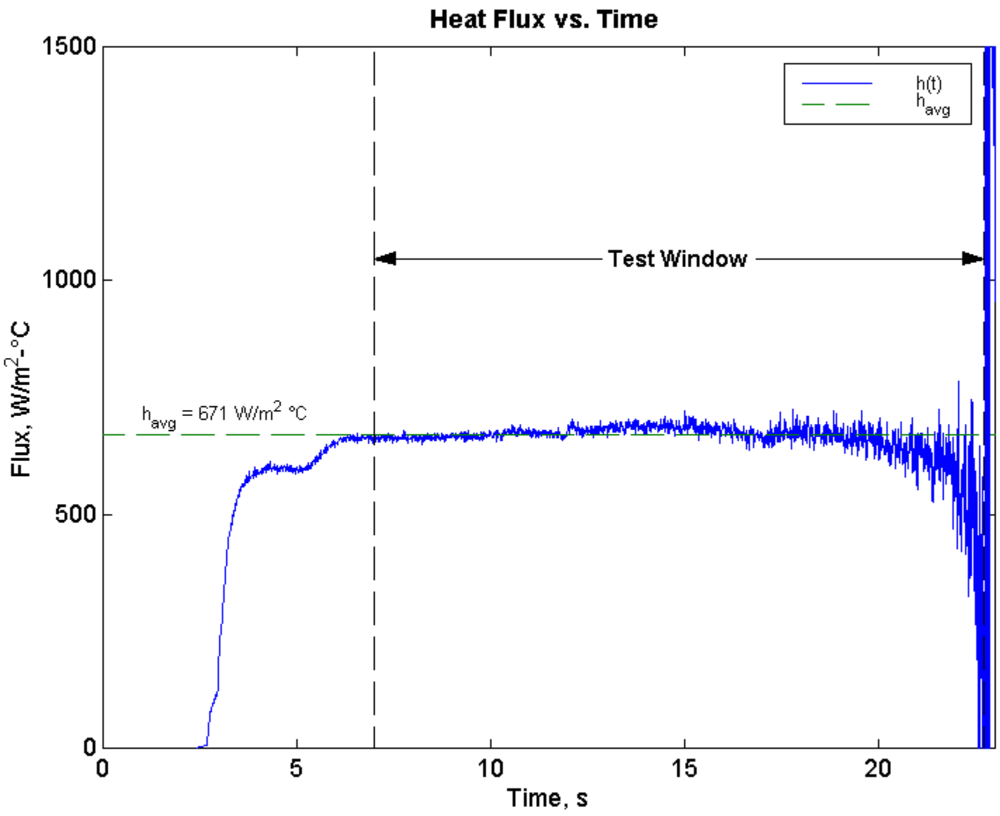


Figure 18: Time History of Heat Transfer Coefficient (uncooled)

As a consistency check of the analysis used on the uncooled data, the recovery and wall temperatures are plotted along with the local heat flux over the run. It should be noted that the local heat flux at this position approaches zero when the recovery and wall temperatures converge. Equation (9) shows that this is to be expected, so Figure 19 provides support for the values of h and $T_{o,\infty}$ - T_r found using this analysis.

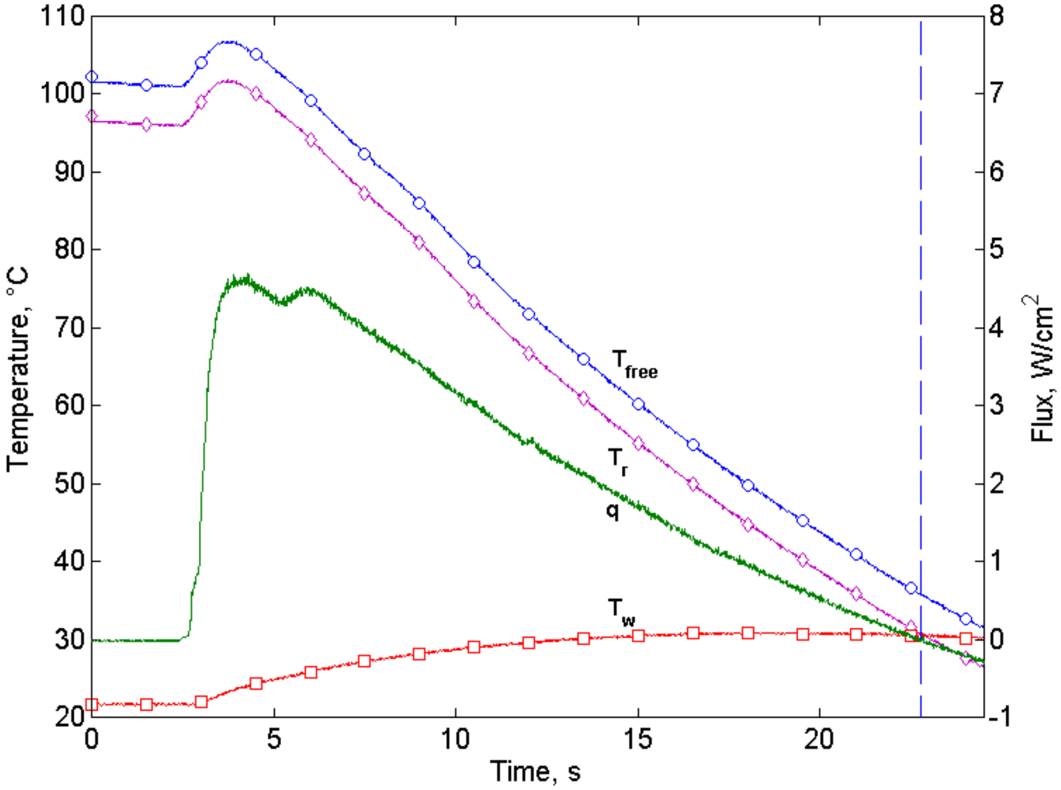


Figure 19: Consistency Check of Uncooled Analysis

5.2 Cold Coolant Analysis

Equation (9) is useful for cases where film cooling is not employed, but if fluid is injected into the boundary layer then the driving force of the heat transfer is dependent upon both the freestream and the injected fluid. For high speed flows with film injection equation (9) can be written as,

$$q = h \cdot (T_{aw} - T_w) \tag{14}$$

where T_{aw} is the driving force of the heat transfer. For high speed flows with no fluid injection, T_{aw} is equivalent to T_r. With coolant injection, however, the adiabatic wall temperature will lie in between the freestream and coolant temperatures(T_c); the magnitude of T_{aw} depends upon the relative magnitudes of T_∞ and T_c as well as flow characteristics such as the blowing and density ratios.

T_{aw} is usually non-dimensionalized using a coolant characteristic called the film effectiveness, η, which is given in the equation (15),

$$\eta = \frac{T_{aw} - T_r}{T_c - T_r} \tag{15}$$

Solving for T_{aw} in equation (15) and substituting it into equation (14) yields,

$$q = h \cdot \left[\left(T_r - T_w\right) - \eta\left(T_r - T_c\right) \right] \tag{16}$$

which can be rearranged to give,

$$\frac{q}{\left(T_r - T_c\right)} = h \cdot \left[\frac{\left(T_r - T_w\right)}{\left(T_r - T_c\right)} \right] - h \cdot \eta \tag{17}$$

The temperatures of the wall, coolant, and freestream as well as the local heat flux into the turbine blade were all recorded for approximately 20 seconds during each run. The data recorded during a typical run with film cooling can be seen in Figure 20. The vertical dashed lines in Figure 20 indicate the portion of the data which was used in the analysis. It should also be noted that there is only one coolant temperature shown although there are three rows of coolant holes on the suction side of the blade. The coolant temperature time history shown is a mass average of the three rows of coolant jets. With these variables and the freestream velocity known, the recovery temperature can be calculated. The only unknowns in Equation (16), then, are the steady heat transfer coefficient and the film cooling effectiveness.

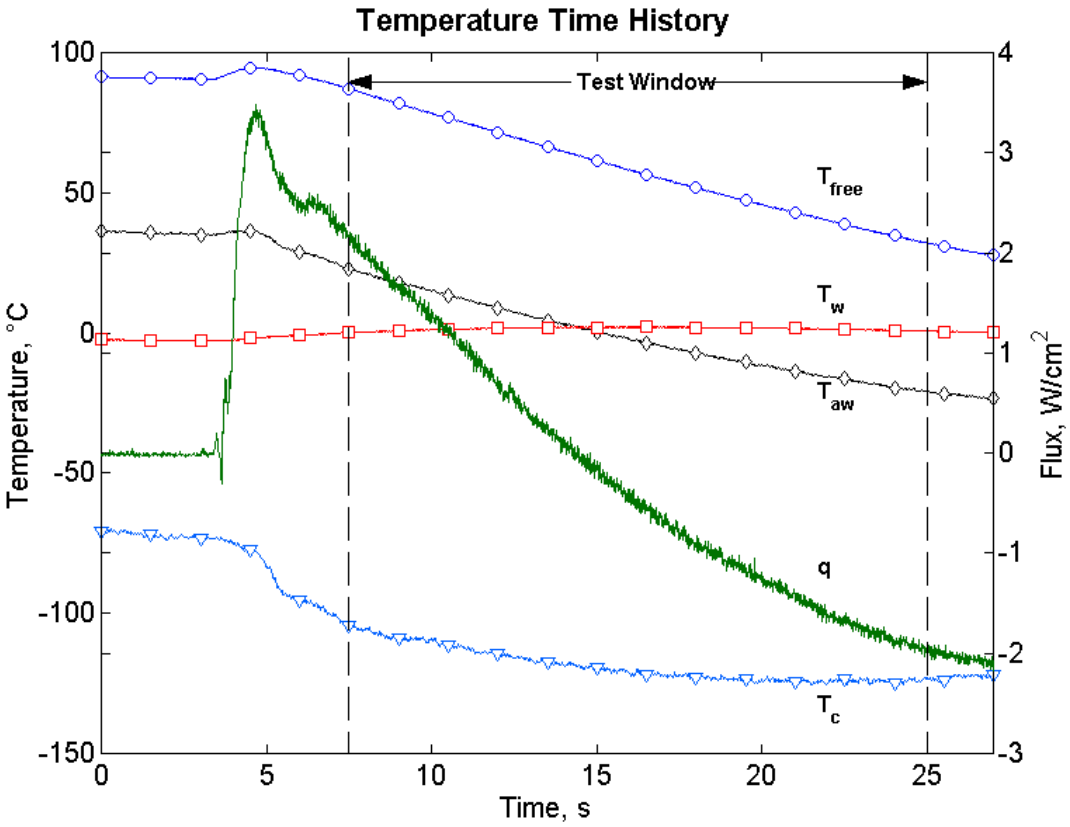


Figure 20: Sample Time History of a Film-cooled Run

If it is assumed that the heat transfer coefficient has only a weak dependence on temperature, then Equation (16) can be plotted such that the local heat flux divided by the recovery to coolant temperature difference (T_r - T_c) should be a linear function of a temperature difference ratio([T_r - T_w]/[T_r - T_c]). The slope of this linear curve is the steady heat transfer coefficient with film cooling and the x-axis intercept is the film cooling effectiveness. The mass averaged coolant temperature as well as the local wall temperature and heat flux are measured directly; the $T_{o,\infty}$ - T_r values calculated in the uncooled experiments were used in this analysis.

If the time history data from Figure 20 are plotted according to Equation (16), the result should be a straight line with the heat transfer coefficient equal to the slope and the film cooling effectiveness equal to the x-axis intercept. An example of the data plotted in this manner is shown in Figure 21. It can be seen that the data is indeed linear which supports the assumption that the heat transfer coefficient has very little dependence on temperature over the range of temperatures used in this study. If the heat transfer coefficient had a strong dependence upon temperature then some trend other than linear should have been found.

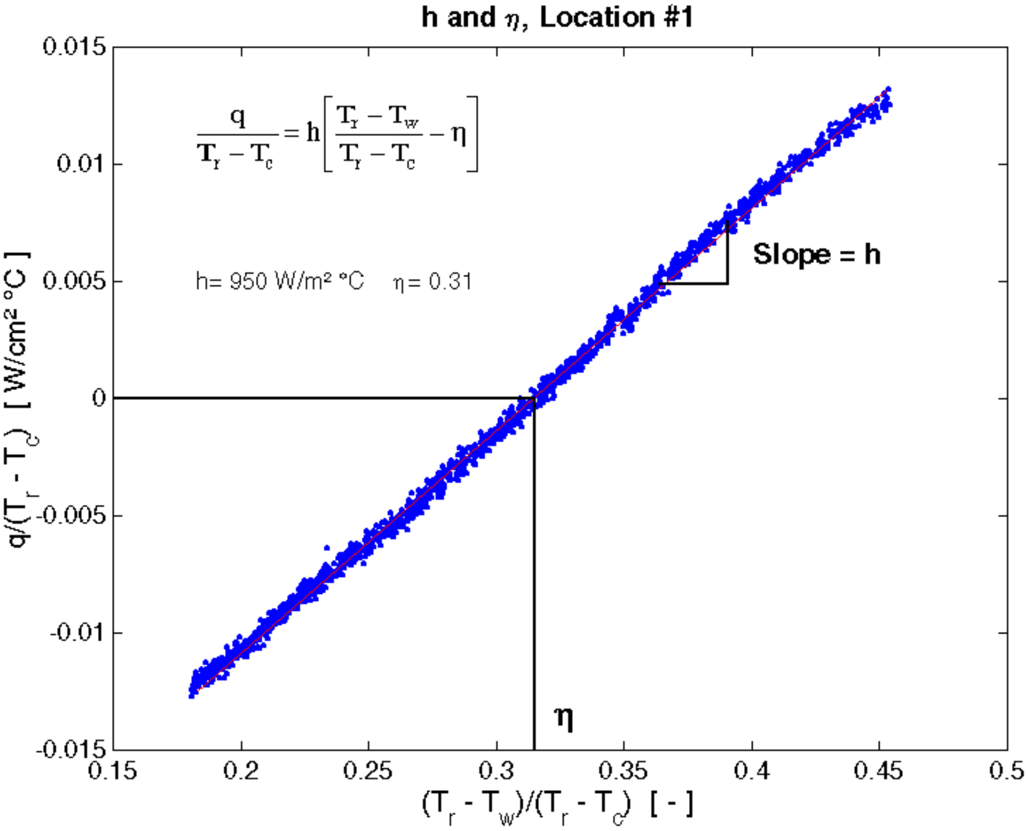


Figure 21: Film-cooled Data Analysis (cold coolant)

After determining values for the heat transfer coefficient and the film effectiveness, the adiabatic wall temperature for the film-cooled case can be determined and used to generate a time history of the heat transfer coefficient. This generated time history can be seen in Figure 22. The magnitude of the oscillations in heat transfer coefficient about the mean increase as the adiabatic wall temperature and the local wall temperature converge. A comparison of the time when this convergence occurs on Figure 20 with the increase in oscillations on Figure 22 will show this relationship.

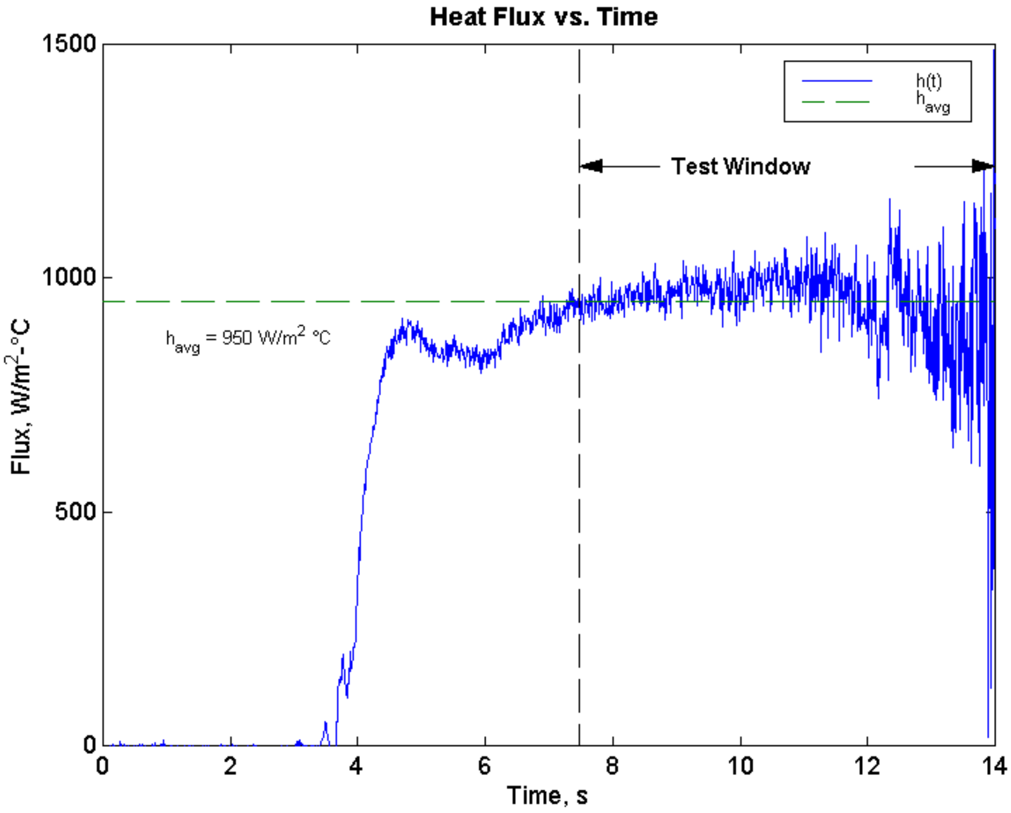


Figure 22: Heat Transfer Coefficient vs. Time (cold coolant)

Just as in the uncooled case, the heat transfer coefficient rises quickly from zero as flow enters the test section and then levels off at the value of the average heat transfer coefficient calculated in the cooled analysis earlier. The steady heat transfer coefficient with film cooling is a constant, like the uncooled case. This means that over the temperature ranges used in this study, the steady heat transfer coefficient with and without film cooling have a weak dependence upon temperature.

Chapter 6: Steady Heat Transfer Experiments

6.1 Uncooled Heat Transfer Experiments

6.1.1 Objectives

Steady heat transfer information about the 2-D turbine inlet rotor blade used in this study is important for understanding how blades like this may perform under actual engine conditions. The unsteady phenomena in the engine environment(shocks, wakes, etc.) cause variations in the mean heat transfer into the blade, but it may be that the mean heat transfer rate is what is of most importance to thermal designers of turbine blades [reference Oxford, 1995 & 1997]. The emphasis of this study is on the effects of film cooling on the steady heat transfer into the turbine blade, but as a baseline, the steady heat transfer coefficient and the recovery temperature on a blade without film cooling need to be determined.

6.1.2 Discussion

The data from the six uncooled runs were analyzed in the manner described in the uncooled analysis section of Chapter 5, and the results along the blade for the steady heat transfer coefficients without film cooling are displayed in Figure 26. The picture in the bottom right-hand corner of the figure shows the relative locations of the six HFM gauges on the suction side of the blade. The gauge locations have been non-dimensionalized using the suction side length. The average value of all locations, around 700 W/m^2·°C, is consistent with results reported in research done by other institutions given the flow conditions of the wind tunnel used in this work[reference appendix and sources].

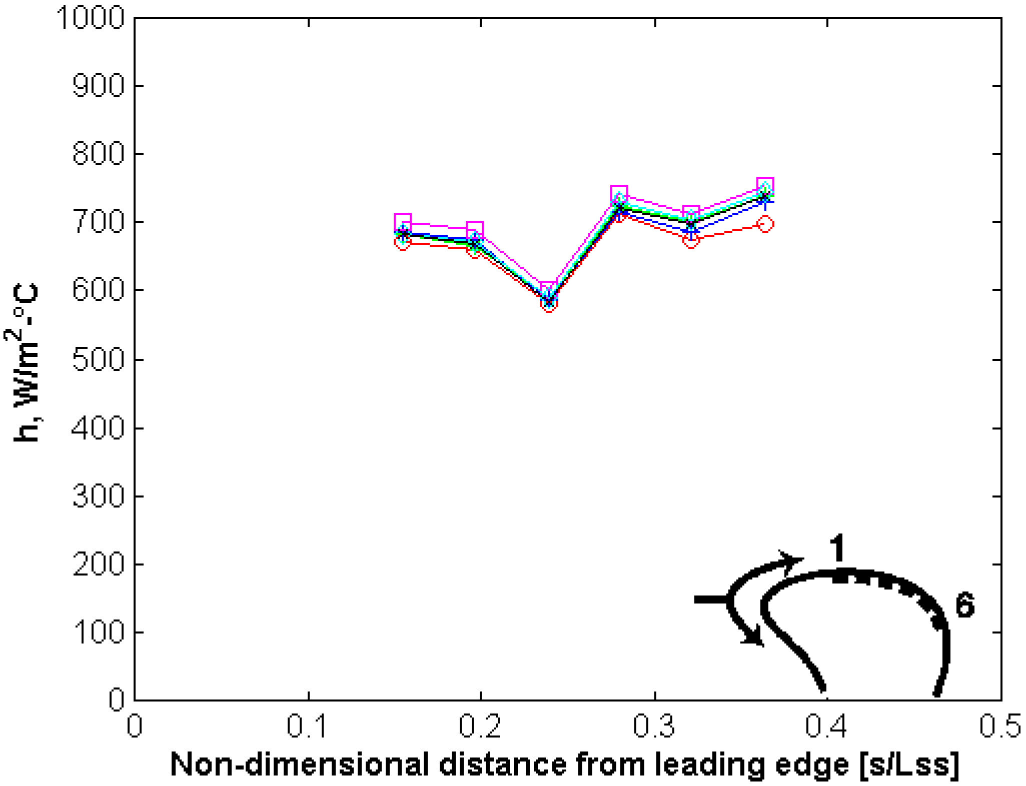


Figure 23: Uncooled Heat Transfer Coefficients Along Blade

The trend in the data from gauge location #1 back to gauge location #6 is unexpected, however. The suction side of the blade is similar to a flat plate with a pressure gradient. The heat transfer coefficient for a flat plate with no pressure gradient decreases with the distance from the leading edge. Therefore, it is expected that the suction side heat transfer coefficient should decrease as the non-dimensional length increases, but this is not the case in Figure 26. Initially, there is a decrease in heat transfer coefficient across the first three locations, but then there is a rise in heat transfer coefficient over the last set of gauges. There are many possible causes of this trend: non-two dimensional flow, disturbance caused by the coolant holes, the strong pressure gradient, or transition. Researchers at the Swiss Federal Institute noticed similar trends in the heat transfer coefficient on a turbine rotor blade without film cooling[Drost et al., 1998]. By tripping the boundary layer, these researchers were able to get the expected decreasing trend in heat transfer coefficient. Time constraints prevented the boundary layer tripping technique from being performed before the conclusion of this study. However, the emphasis in this study was in finding the average level of the heat transfer coefficient under steady conditions to provide a basis for unsteady investigations, and this is not affected by the observed trends.

The value of $T_{o,\infty}$ - T_r calculated from experimental data can be plotted versus a predicted value based upon a correlation for the recovery factor in Equation (11) and knowledge of the freestream velocity. If it is assumed that the flow field is turbulent, the recovery factor can be approximated with $Pr^{1/3}$ [Kays and Crawford, 1990]. Knowing the Mach number distribution along the blade and using the approximated recovery factor, a prediction of the $T_{o,\infty}$ - T_r profile can be made. Figure 27 shows the experimental $T_{o,\infty}$ - T_r profile plotted along with the predicted value.

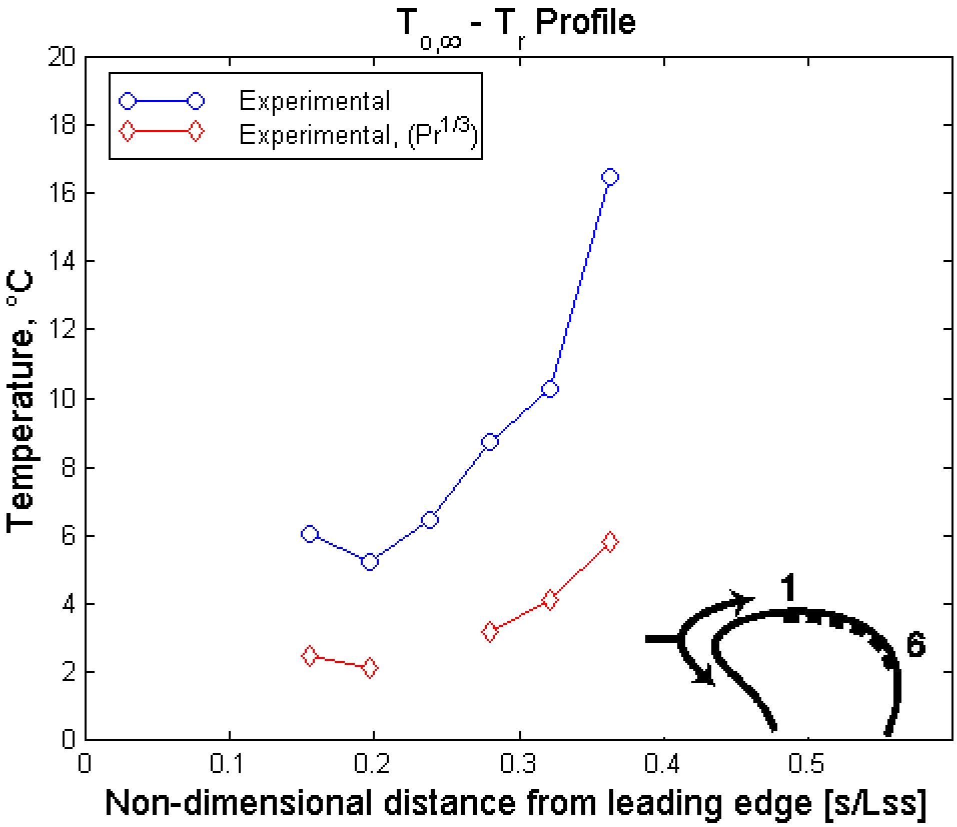


Figure 24: Comparison of Experimental and Predicted $T_{o,\infty}$ - T_r

This is a difficult measurement to make due to the uncertainties in temperature measurements, so some degree of imprecision is expected. The $T_{o,\infty}$ - T_r values calculated using experimental data in the uncooled analysis are smaller than the predicted values at all locations. However, the general trend is the same and the experimental and predicted values are of the same order of magnitude. This is an important point, because the two methods of determining $T_{o,\infty}$ - T_r are not linked. The experimental value is solely a function of the temperature and heat flux measurements made, while the predicted value depends on pressure measurements and the approximated recovery factor, $r_c = Pr^{1/3}$. This analysis is important because it provides a method of determining the recovery factor on turbine blades. The experimental values for $T_{o,\infty}$ - T_r at each gauge location were used to calculate the difference between the freestream and recovery temperatures for all data calculations in the cold coolant runs. The effect of using the experimental values, as opposed to the predictions, on the steady heat transfer coefficient and film cooling effectiveness will be examined later in section 6.3.

6.2 Cold Coolant Heat Transfer Experiments

6.2.1 Objectives

Information about steady heat transfer with film cooling on the 2-D turbine inlet rotor blade used in this work is important in the design of future blades. Knowing the steady heat transfer coefficient and the recovery temperature on a blade without film cooling, a comparison can now be made to a blade with film cooling. The cold coolant heat transfer experiments were performed in order to investigate the heat transfer coefficient and the adiabatic film cooling effectiveness in order to determine the effect film cooling has on the turbine rotor blade used in this work.

6.2.2 Discussion

Data from all six gauges were gathered over six runs with film cooling. The dashed lines in Figure 20 represent the set of data that was actually used in the data analysis. Approximately twenty seconds of data were analyzed for each run with film cooling. Since the data was recorded at a sampling frequency of 100Hz, there are approximately 2000 data points gathered for each run. The coolant temperature shown is a mass averaged quantity determined from the temperatures and mass flows of the three rows of coolant holes whose injectants fold over onto the suction side. Figure 20 illustrates the transient nature of the experiments in the blow down wind tunnel.

The data from the six runs with a cold coolant injected as a film cooling layer were all analyzed in this manner. The results for the steady heat transfer coefficient along the blade with film cooling can be seen in Figure 28. The gauge locations are again plotted using the length of the suction side of the blade to non-dimensionalize the distance measured from the stagnation point to each gauge location. The three arrows in the bottom left of the figure represent the locations of the coolant hole on the suction side of the blade. Earlier, in Figure 14 a shadowgraph picture of the coolant flow was shown. This photograph indicates that only these three rows of coolant holes cover the suction side of the blade.

The average value of the film-cooled heat transfer coefficient, around 800 $W/m^2\cdot°C$, is of the same order of magnitude as other researchers have found for cascade turbine blades with film cooling given the flow conditions of our wind tunnel and the geometry of our test section. A rise in heat transfer coefficient can be seen when the uncooled heat transfer distribution is compared to the film-cooled distribution. This rise in heat transfer coefficient is expected due to the increase in the level of turbulence caused by the injection of a cooling fluid into the boundary layer. However, the injected coolant also lowers the adiabatic wall temperature by an average of about 50°C. This decrease in the adiabatic wall temperature more than compensates for the rise in heat transfer coefficient, so the overall local heat flux into the blade is decreased by the injection of a cold fluid into the boundary layer.

The heat transfer coefficient distribution with film cooling displays trends similar to those found without film cooling. With film cooling, however, the initial decrease is more emphasized. It is suspicious that the injection of coolant into the boundary layer has no effect on the location of the possible transition from laminar to turbulent flow. There is a lot of fluid being injected, and this disturbance should have caused any transition to occur earlier than in the uncooled case. The observed trend could still be the result of disturbances in the flow caused by the coolant holes or non-uniform spreading of the coolant film over the gauges. As stated earlier, however, the trend does not prevent the goal of this work from being achieved.

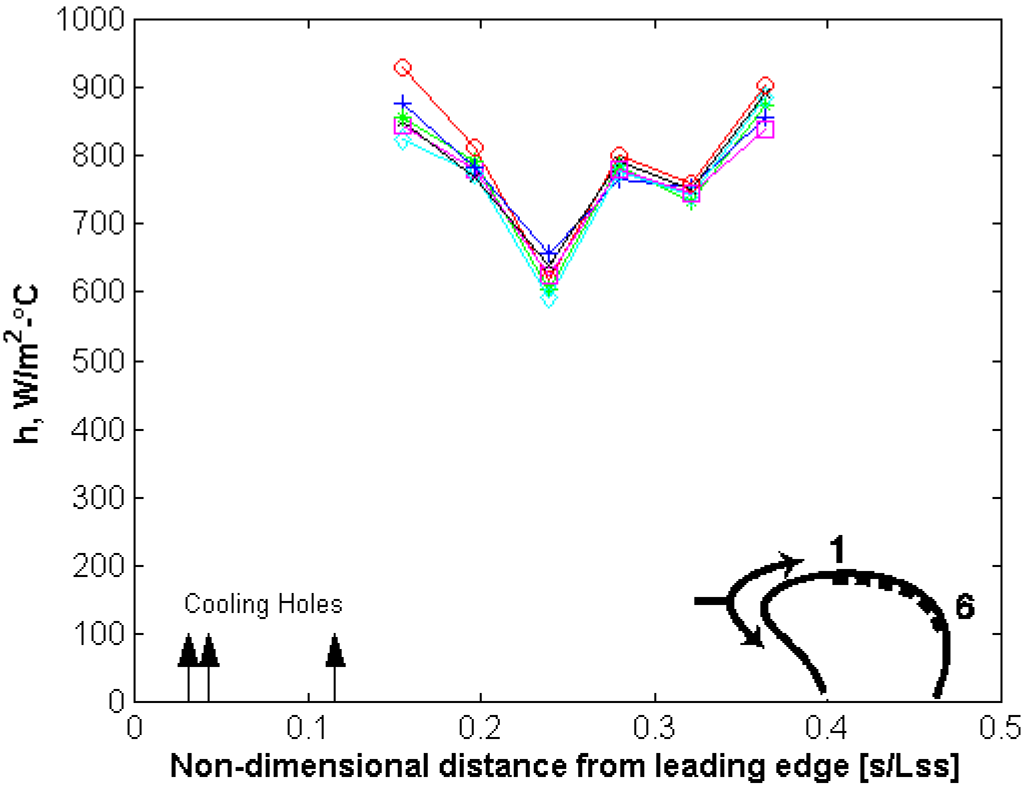


Figure 25: Cold Coolant Heat Transfer Coefficients Along Blade

Without further attempts to explain the observed trends in the heat transfer coefficient distribution along the blade, the rest of the film-cooled results will be presented. The results of the film cooling effectiveness distribution are displayed in Figure 29. The effectiveness distribution is plotted versus the same non-dimensionalized distance used earlier in Figures 23 - 25. The effectiveness of the film varies erratically up and down from gauge to gauge as you move farther along the suction side of the blade. It is suspected that this variation has the same cause as the trends observed in heat transfer coefficient. The average value of the film cooling effectiveness is around 25%.

In the earlier section on the uncooled analysis it was mentioned that the recovery temperatures were calculated from the linear fit and not by using the correlation relating the Prandtl number to the recovery factor. The difference between the two methods was illustrated earlier in Figure 24. The effect the two methods of calculating recovery factor have on the film cooling effectiveness can also be seen in Figure 26. It is evident that using the experimental recovery factors does not have a major effect on the film cooling effectiveness.

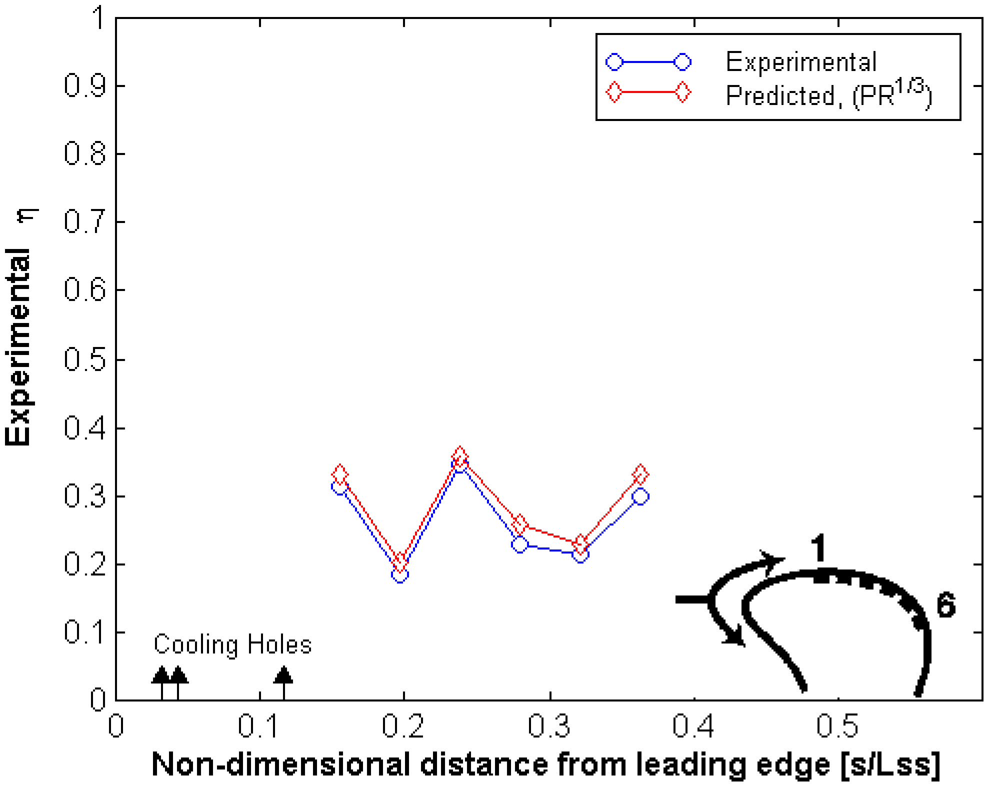


Figure 26: Effect of Experimental vs Predicted $T_{o,\infty}$ - T_r on eta

A comparison can be made of the steady heat transfer coefficient with and without film cooling present on the surface of the blade. Figure 27 shows the mean values from the six runs for both the uncooled and film-cooled cases. It is evident from this figure that presence of a film cooling layer increased the steady heat transfer coefficient along the blade.

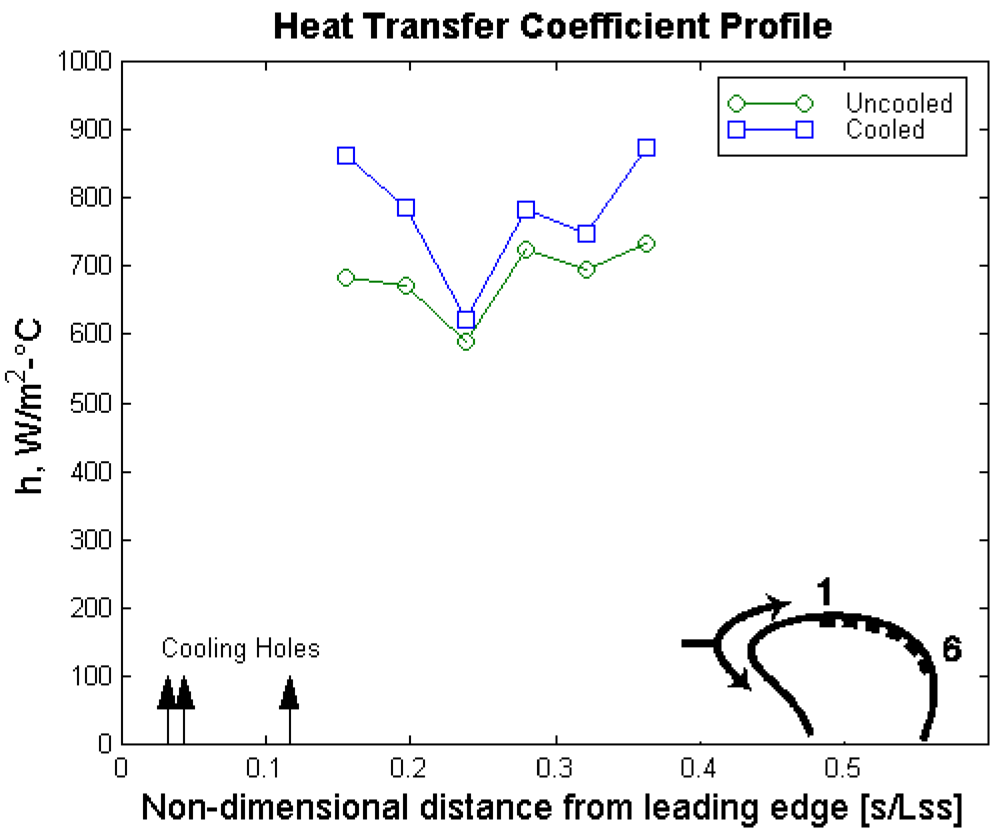


Figure 27: Comparison of Trends in h (uncooled, cold coolant)

The injection of the fluid disturbs the boundary layer and encourages mixing, both of which lead to the increase in heat transfer coefficient. Both the uncooled and the film-cooled results display the same trends; they are slightly more exaggerated in the film-cooled case, however. This figure shows that there is no change in the location of the transition region with the addition of film cooling. As stated earlier, this lack of response to the addition of film cooling suggests that the trends may not be the result of transition.

The actual values of the mean heat transfer coefficient of the six runs for both cases can be seen in Table 2. Also included in the table is the percent increase in heat transfer coefficient at each gauge location. The maximum increase of 26% occurs at the first gauge location which is approximately 9.4 hole diameters back from the last row of coolant holes. From the data in Table 2, it is evident that the percent increase in heat transfer coefficient from the uncooled case to the film-cooled case follows the same trend as seen in both cases separately.

Table 2: Comparison of Uncooled vs. Film-cooled Heat Transfer Coefficient

Gauge Location	**Uncooled Heat Transfer Coefficient ($W/m^2\cdot°C$)**	**Film-cooled Heat Transfer Coefficient ($W/m^2\cdot°C$)**	**Percent Increase %**
1	684	864	26
2	672	785	17
3	588	622	6
4	724	783	8
5	696	747	7
6	734	875	19

Chapter 7: Conclusions and Future Work

The instrumented turbine rotor blade was placed into the Virginia Tech Transonic Wind Tunnel where testing was performed. The Mach and Reynolds numbers at the exit of the test section were set to 1.2 and $5x10^6$,respectively, in order to simulate real engine flow conditions. The freestream and coolant flows were maintained at a total temperature ratio of 2 ±0.4 and a total pressure ratio of 1.04. The freestream turbulence level was approximately 1%. Two different sets of experiments were run: 1) heated mainstream flow with no film cooling, 2) heated mainstream flow with cold coolant.

Comparing the results of these two experiments showed that there was a decrease of heat transfer into the blades from the experiments without film cooling to those with film cooling, even though there was an increase in the steady heat transfer coefficient. The reason for the decrease in heat flux levels was the decrease in the adiabatic wall temperature, the driving force of the heat transfer, brought on by the injection of a cold coolant into the boundary layer. The average value of the heat transfer coefficient increased from 700 $W/(m^2 \cdot °C)$ without cooling to 850 $W/(m^2 \cdot °C)$ with film cooling. The average film cooling effectiveness along the blade was found to be 25%.

With the addition of film cooling, then, it can be concluded that the heat transfer coefficient increases approximately 21% over the section of the suction side of the rotor blade investigated in this work. From the uncooled runs, a difference was seen between the recovery temperature calculated using the uncooled analysis and the recovery temperature calculated using the correlation, $r_c = Pr^{1/3}$. This difference, approximately

6ºC, is not significant in the calculation of the film cooling effectiveness because it is based upon temperature differences on the order of 80ºC.

The results from both experiments displayed an unexpected trend in the heat transfer coefficient measured along the blade. In all cases, there is an initial decrease in heat transfer coefficient followed by a sudden increase almost back up to the starting level. This falling-rising trend in the heat transfer coefficient could be the result of either three-dimensionality in the flow-field (end wall effects, spanwise variations, etc.) or onset of the transition from a laminar to a turbulent boundary layer. The specific cause of the trends was not investigated in this study.

This work only investigated the steady heat transfer into a two-dimensional turbine inlet rotor blade. In the future, the cause of the trends in the steady heat transfer coefficient should be investigated. Oil drop visualizations could be performed along the span of the blade to check for spanwise variations and the boundary layer could be tripped near the leading edge to prevent transition from occurring later on the surface of the blade. In addition, the effect of different turbulence levels and blowing parameters on the steady heat transfer should be determined. Once the issues around the steady heat transfer into the blade have been resolved, the effects of unsteady events such as passing shocks and wakes should be investigated.

References

Abuaf, N., Bunker, R., Lee, C.P., G.E. Aircraft Engines, 1995, "Heat Transfer and Film Cooling Effectiveness in a LInear Cascade, " 40th IGTI Conference, ASME 95-GT-3.

Drost, U., Bölcs, A., Swiss Federal Institute of Technology, 1998, "Investigation of Detailed Film Cooling Effectiveness and Heat Transfer Distributions on a Gas Turbine Airfoil," ASME 98-GT-20.

Ekkad, S.V., Hui Du, and Je-chin Han, Texas A& M University, 1995, "Local Heat Transfer Coefficient and Film Effectiveness Distributions on a Cylindrical Leading Edge Model Using a Transient Liquid Crystal Image Method," ASME 95-WA/HT-3.

Goldstein, R.J., “Measurements in Heat Transfer,” Hemisphere Pub. Corp., 1976.

Guo, S.M., Lai, C.C., Jeong, J.H., Jones, T.V., and Oldfield, M.L.G., University of Oxford, 1997, "Use of Liquid Crystal Techniques to Measure Film Cooling Heat Transfer and Effectiveness," 90th Symposium of the Propulsion and Energetics Panel, AGARD 20-24.

Guo, S.M., Jones, T.V., University of Oxford, and Lock, G.D., University of Bath, 1996, "Gas Turbine Heat Transfer Measurements with Engine Simulated Film Cooling," 2nd European Conference on Turbomachinery Fluid Dynamics and Thermodynamics, Technological Institute, Section on Aeronautics.

Horton, F.G., Shultz, D.L., Forest, A.E., University of Oxford, 1985, "Heat Transfer Measurements with Film Cooling on a Turbine Balade Profile in Cascade," ASME 95-GT-117.

Kays, W.M., Crawford, M.E., “Convective Heat and Mass Transfer,” McGraw-Hill Inc., pp 380-388, 1993.

Schwarz, S.G., Goldstein, R.J., Eckert, E.R.G., Tulane University/University of Minnesota, 1990, "The Influence of Curvature on Film Cooling Performance," 35th IGTI Conference, ASME 90-GT-10.

Appendix A: Author's Published Research

A-1 Popp et al., 1999

Steady and Unsteady Heat Transfer in a Transonic
Film-Cooled Turbine Cascade

ASME Paper 99-GT-259

A-2 Smith et al., 1999

A Comparison of Radiation versus Convection Calibration
of Thin-Film Heat Flux Gauges

ASME Paper 99-??-???

STEADY AND UNSTEADY HEAT TRANSFER IN A TRANSONIC FILM COOLED TURBINE CASCADE

O. Popp, D. E. Smith, J. V. Bubb, H. C. Grabowski III, T.E. Diller, J. A. Schetz, Wing-Fai Ng
Virginia Polytechnic Institute and State University
Blacksburg, VA 24061

ABSTRACT

This paper reports on an investigation of the heat transfer on the suction side of a transonic film cooled turbine rotor blade in a linear cascade. Heat transfer coefficient and film effectiveness are first determined for steady conditions. The unsteady effects of a passing shock on the heat transfer are then investigated. The film cooling pattern used is a showerhead design with three rows on the suction side, one row at the stagnation point and two rows on the pressure side. The experiments were performed at engine representative temperature and pressure ratios using air as coolant. Heat transfer measurements are obtained using a Heat Flux Microsensor, and surface temperature is monitored with a surface thermocouple. Static pressure is monitored with a Kulite pressure transducer. The shock emerging from the trailing edge of the NGV and impinging on the rotor blades is modeled by passing a shock wave along the leading edges of the cascade blades. The steady-state heat transfer coefficient is 8% higher with film cooling than without film cooling. Shock heating of the freestream flow is determined to be the major contribution to the unsteady variation of heat flux, leading to an increase of about 30°C to 35°C in recovery temperature and adiabatic wall temperature.

NOMENCLATURE

Symbols

B	blowing ratio $(\rho u)_c/(\rho u)_f$
d	cooling hole diameter (1 mm)
c_p	specific heat of air, 1005 J/(kg K) in Eq.(3)
h	heat transfer coefficient w/o film cooling
h_c	heat transfer coefficient w/ film cooling
I	momentum ratio $(\rho u^2)_c/(\rho u^2)_f$
M	density ratio ρ_c/ρ_f
p	static pressure
Pr	Prandtl Number (0.71 in Eq.(9))
q	heat flux per unit area
q'_{max}	peak value of the unsteady component of heat flux
q_ε	bias in heat flux measurement (Eq.(8))
r	recovery factor in Eq.(3) and Eq.(6)
T_{aw}	local adiabatic wall temperature
T_c	coolant exit temperature
T_d	T_t-T_r Eq.(3)
T_d^*	real value of measured T_d
T_p	coolant temperature in the cooling plenum
T_r	local recovery temperature
T_t	freestream total temperature
T_w	local wall or blade temperature
T_ε	bias in temperature measurement (T_r-T_w) (Eq.(5))
u	local freestream velocity in Eq.(3)
η	film effectiveness defined in Eq.(7)

Superscripts

´	unsteady variation

Subscripts

$_c$	coolant or w/ film cooling
$_f$	freestream

INTRODUCTION

The efficiency of a gas turbine engine increases with turbine inlet temperature. In the ongoing effort to raise the turbine inlet temperature the gas stream temperature is made to greatly exceed the operating temperatures of blade materials, requiring elaborate blade cooling techniques to be developed. One of these methods is to spread a thin layer of cold air between the hot gas and the surface to be protected, referred to as film cooling. The quest for higher thrust to weight ratios in the development of aero-engines has led to the design of nozzle guide vanes (NGV) with supersonic exit velocities. The rotor blades consequently are not only subject to wake but also shock impingement as they pass behind the NGV's at very high speed. The effect of this unsteady process on the heat transfer to the rotor blade has been the topic of a variety of research programs.

The vast majority of results have been presented by the research team led by Schultz and Jones at the University of Oxford. Johnson et al. (1990) investigated the unsteady heat flux on rotor blades in a linear cascade simulating the wake and shock with a rotating bar mechanism upstream of the cascade. They found a turbulent spot forming on the leading edge produced after the collapse of the shock induced separation. Travelling along the suction side, this turbulent spot increases the heat transfer. Boundary layer transition due to wake impingement was observed to further enhance unsteady heat flux. Moss et al. (1997) performed tests in a rotating annular cascade indicating that the unsteady disturbances caused by the NGV's have little influence on the heat transfer coefficient and the time averaged heat flux. They indicate that the unsteady heat flux is caused mostly by the time variation in relative total temperature. The mean heat transfer level therefore is not strongly affected by the presence of the NGV's. None of these experiments involved rotor blade film cooling. Film cooling experiments have been done on the same blade geometry (Horton et al. 1985) but did not include unsteady effects.

Hilditch et al. (1995) performed time resolved heat transfer measurements on an axial turbine rotor and compared his results with data from the University of Oxford and MIT. The rotor blades were not cooled and no analysis was done to discriminate shock and wake effects.

Similarly, Abhari and Epstein (1992) measured unsteady heat flux on a film cooled rotor in a rotating transonic turbine stage. They observed large fluctuations of heat transfer over a blade passing period but did not distinguish between the effects of shocks and wakes.

Heidmann et al. (1997) experimentally and numerically investigated the effect of wake passing on the time-averaged heat flux in a film cooled annular cascade, modeling the wake using a rotating bar mechanism.

Hale et al. (1997) modeled the effect of wake passing in a quasi-steady way using a stationary strut. Increases in heat transfer coefficient were measured for a number of locations on the blade, particularly on the pressure side.

Nix et al. (1997) analyzed in detail the progression of a shock through the same cascade and its effect on the unsteady heat transfer. When averaged over a 200µs blade passing event, a maximum increase of heat flux of 60% was measured due to shock passing.

The intent in the present study is to measure and interpret the unsteady heat transfer due to an isolated shock, as opposed to a combination of wake and shock. The focus of the research has been extended to film cooled blades.

EXPERIMENTAL APPARATUS

Wind Tunnel Facility, Cascade and Shock Apparatus

The experiments necessary for this investigation were performed in the transonic blowdown wind tunnel at Virginia Tech. A passive heating device is available to achieve high (120°C) inlet temperatures to the cascade. It consists of many copper tubes that are preheated prior to running the tunnel. Fig. 1 shows the wind tunnel with the heating loop. With the present cascade, the facility allows run times of up to 35 seconds with the inlet pressure controlled. The test-section and cascade built for this investigation are shown in Fig. 2. The cascade consists of four full and two half blades forming five passages (see Fig. 2). The blade design is a generic, high-turning, first stage rotor geometry. It is scaled up three times to accommodate the cooling scheme and instrumentation. The span is 15.3 cm (6") and the aerodynamic chord is 13.6 cm (5.4"). Pitch and axial chord are 11.4 cm (4.5"). The Reynolds Number based on aerodynamic chord and exit conditions is about $6 \cdot 10^6$. The Mach Number distribution was shown to correspond to design conditions. To simulate the shock emerging from the trailing edge of a NGV, a shock tube creates a shock wave which is sent along the leading edges of the cascade (see Fig. 2). The shock strength can be varied to obtain realistic pressure ratios. For the present investigation a shock strength of 1.08 (ratio of local static pressure behind shock and local static pressure before shock impact) was chosen. Accordingly, the shock Mach number is about 1.03 relative to the freestream flow.

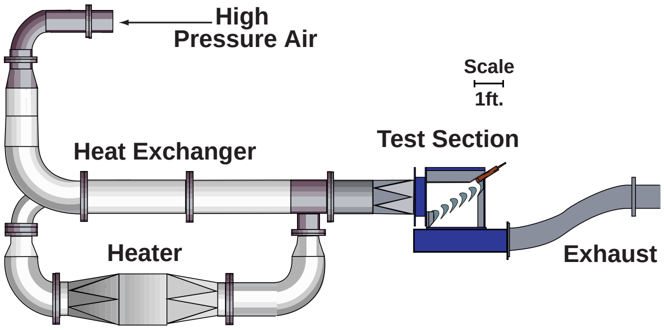


Figure 1: Wind Tunnel Facility

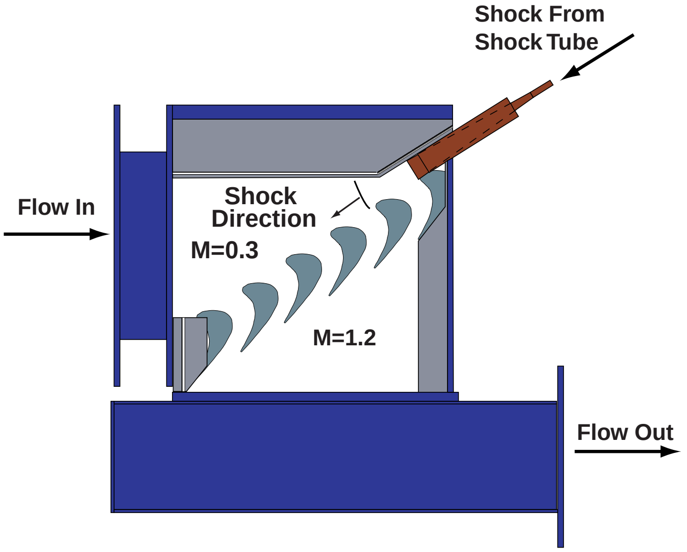


Figure 2: Cascade and Shock Apparatus

Cooling Pattern

A schematic of the showerhead film cooling design is shown in Fig. 3. All coolant holes are cylindrical and straight. The pressure and suction side gill holes form angles with the local chordwise tangent of 45° and 30°, respectively, and have no inclination in the radial direction. All other rows of holes are normal to the local chordwise tangent but angled 60° in the radial direction. Each row consists of 14 holes with a diameter of 1.04 mm (0.041") and a spacing of 9.14 mm (0.360"). The rows are staggered half the spacing with respect to the neighboring rows, yielding an overall pitch/diameter ratio of 4.39. Length/diameter ratios vary from 11.5 for the suction side gills to 4.4 for the suction side nose #2 row. Only the suction side gills and the suction side rows #1 and #2 actually affect the suction side heat transfer. The coolant ejected through the stagnation point row actually flows along the pressure side as observed from shadowgraph pictures. That means that in spite of pressure measurements locating the stagnation point right at the stagnation point row exit, it must be shifted towards the suction side.

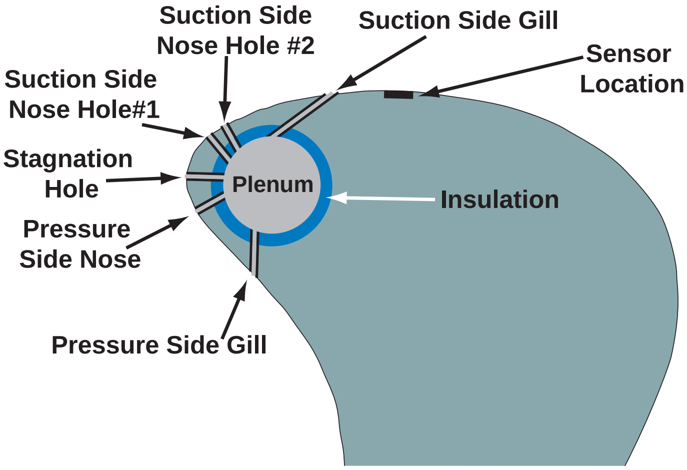


Figure 3: Blade Cooling Scheme

The nominal ratio of coolant to freestream total pressure for these experiments is 1.04. Therefore, the Momentum Ratio for each row of holes is kept constant while Density Ratio and Blowing Ratio vary with coolant temperature as shown later in section 'Steady-State Data Analysis and Results, With Film Cooling'. The temperature ratio T_t/T_c decreases from about 1.9 early in the experiment to about 1.5 late in the run. Homogeneous blowing through the cooling holes was checked by traversing a total and static pressure probe along the centerline of the cooling plenum. The resulting linear velocity distribution indicated uniform blowing. Coolant exit temperatures are measured with very small exposed junction thermocouples protruding into the exit of the last set of cooling holes. Conduction errors in those measurements were investigated experimentally. It was determined that these errors were negligible for the application in the blade. For the experiments without film cooling the plenum is fully plugged with a tightly fitting Nylon rod. The coolant supply is shown in Fig. 4. The two stage reciprocating compressor provides pressurized air at 12 bar (160 psig) to the storage tank. The dryer lowers the humidity to below three percent relative humidity.

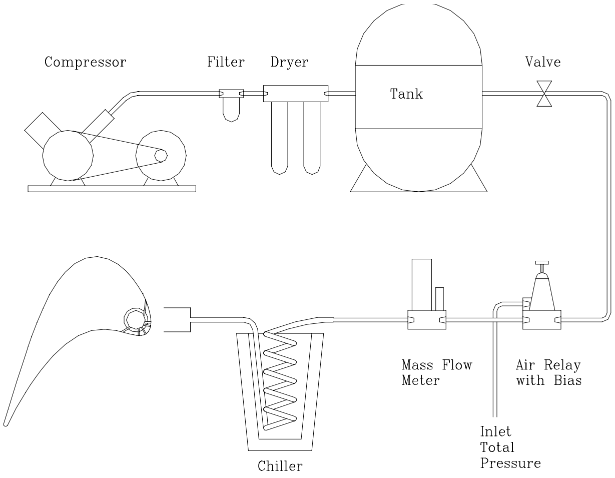


Figure 4: Coolant Supply

In order to control the difference between coolant and freestream total pressure, an air relay is used with the freestream total pressure as the signal and an adjustable bias. The chiller is a copper tube heat exchanger immersed in liquid nitrogen. It provides coolant temperatures down to –100°C in the plenum.

Sensors

The measurement location in this investigation is indicated in Fig. 5. Three different sensors are placed staggered in the spanwise direction. A surface thermocouple monitors the local blade temperature. It was designed to have thermal properties similar to the surrounding blade material (aluminum), and it is press fit to provide good thermal contact. The surface static pressure is measured with a Kulite pressure transducer, so that pressure variations due to the shock passing can be captured. Heat flux is measured using a heat flux microsensor. This sensor is described in detail in Diller (1993).

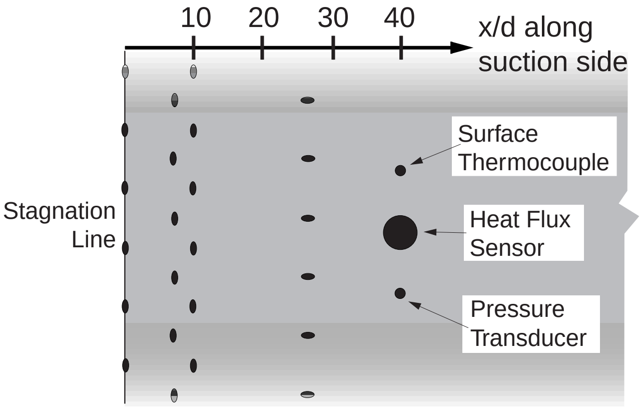


Figure 5: Plan View of a Section of the Suction Side Showing Sensor Locations and Coolant Exits

It behaves similar to a first-order system with a time constant of about 6μs. Therefore, it is capable of tracing rapid changes like a shock passing with sufficient accuracy. The substrate material of the sensor is aluminum nitride which has thermal properties similar to aluminum but is electrically insulating. Consequently, the temperature history of the gauge should closely follow the local blade temperature. The active diameter of the gage is 5.3 mm (0.21”). Accordingly the gauge extends from 10 to 15 cooling hole diameters downstream of the suction side gills. The relative size and location of the sensor are expected to provide a spatially averaged value of heat flux. For the steady-state investigation, all signals are sampled at 100 Hz and filtered at 50 Hz. Both pressure and heat flux signal are sampled at 1 MHz and filtered at 40 kHz for the unsteady investigation.

Optical Access

To monitor the state of the cooling film and to visualize the shock passing process, shadowgraph pictures were taken either using Polaroid film (steady-state) or a high speed digital camera (shock passing). The digital camera is capable of taking four successive pictures with a frequency of up to 1 MHz. The high speed capability is necessary to investigate in detail the effect of the passing shock.

STEADY-STATE DATA ANALYSIS AND RESULTS

Without Film Cooling

The general definition of the heat transfer coefficient used here is

$$q = h \cdot (T_{aw} - T_w) \qquad (1)$$

With no cooling film present, the adiabatic wall temperature is the recovery temperature. Therefore

$$q = h \cdot (T_r - T_w) \qquad (2)$$

The difference between the freestream total temperature and the recovery temperature is a function of the freestream velocity and the recovery factor

$$T_d = T_t - T_r = (1 - r) \cdot \frac{u^2}{2 \cdot c_p} \qquad (3)$$

This difference T_d is a constant throughout the run. Therefore Eq. (2) can be written as

$$q = h \cdot (T_t - T_w) - h \cdot T_d \qquad (4)$$

Eq. (4) is a linear equation with the independent variable (T_t-T_w) and the dependent variable q. The slope is the heat transfer coefficient h, and T_d is the intercept at q=0 as illustrated in Fig. 6. The temperatures T_t, T_w, and the heat flux q vary during the experiment, since the passive heating device is cooling down as the freestream air is drawing heat from the copper tubes. The blade temperature is increasing during the tunnel run, so the overall temperature difference (T_t-T_w) and the heat flux q decrease (see Fig. 7). Assuming that h is not a function of the temperatures involved, one can obtain the heat transfer coefficient and T_d by fitting Eq. (4) to the data. A typical graph illustrating this technique is shown in Fig. 6. The data shows linear behavior as expected. The difference between the freestream total and wall temperatures never actually reaches zero. It typically spans the range from 70°C down to 10°C. The intercept at q=0 is, therefore, an extrapolation which seems justified. The corresponding time history of heat transfer coefficient (h) and recovery temperature obtained using the calculated T_d is shown in Fig. 7 along with the total temperature (T_t), the blade temperature (T_w), and the heat flux (q) during a run.

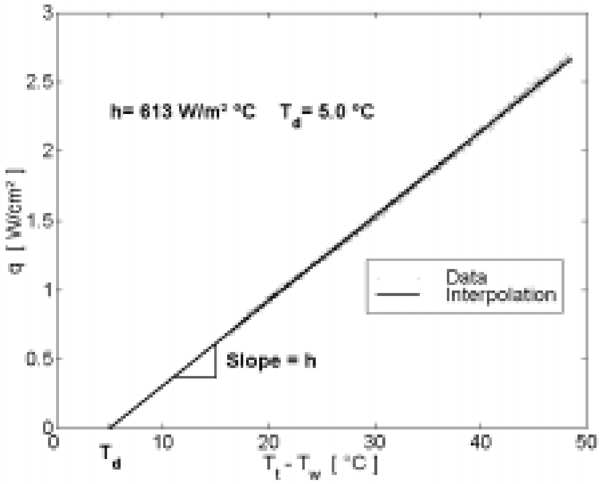


Figure 6: Interpolation for h and T_d (Uncooled Run #6)

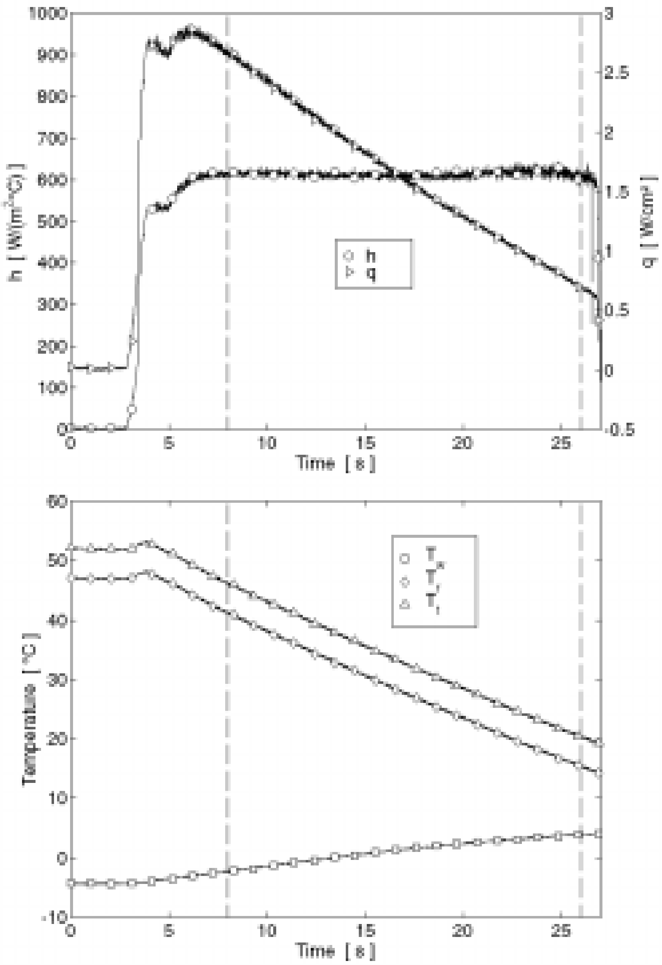


Figure 7: Time History of T_t, T_r, T_w, h, q (Uncooled Run #6)
(Dashed Lines Indicate Range of Useful Data)

An error analysis shows that bias errors in both the heat flux and the temperature measurement do not affect the resulting heat transfer coefficient. This is because an offset of the data in either the x or y-direction ((T_r-T_w) and q respectively) does not change the slope of the curve. Accordingly, the experimental scatter for the heat transfer coefficient is small. T_d on the other hand is more severely affected by measurement uncertainty. A bias error in both temperature and heat flux measurement transfers directly into an error in recovery temperature

$$T_d = T_d^* - T_\varepsilon + \frac{q_\varepsilon}{h} \tag{5}$$

Therefore, the scatter is larger. The experimental results for several repeating runs are shown in Table 1. The difference between total and recovery temperature based on

$$r = \sqrt{Pr} \tag{6}$$

and the local Mach Number of 0.6 would yield a value of T_d of 4.2°C, assuming an average T_t. Shadowgraph and Schlieren pictures had shown that the boundary layer at the blade location of interest is laminar. It needs to be stated that all the experimental results of T_d are higher than the ones based on isothermal flat plate calculations. The reason for this is either in the measurement accuracy (Eq.(5)) or in the steep pressure gradient.

Run #	1	2	3	4	5	6	7	8	9
h [W/m^2°C]	635	620	639	654	664	613	625	623	637
T_d [°C]	6.9	8.6	7.0	7.7	8.8	5.0	6.9	7.3	4.7

Table 1: Run-to-Run Variation of h and T_d

With Film Cooling

To indicate the state of the cooling film, shadowgraphs and Schlieren pictures are taken both with Polaroid film and the high-speed digital camera. The shadowgraph in Fig. 8 shows that the film is attached to the surface showing turbulent structures. The pressure side is hidden by instrumentation outside of the cascade.

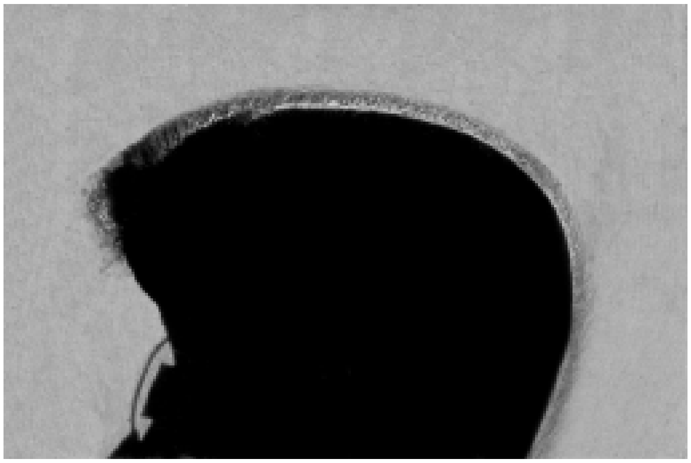


Figure 8: Shadowgraph Showing Attached Film

With film cooling, the adiabatic wall temperature is usually expressed in terms of the film effectiveness.

$$\eta = \frac{T_{aw} - T_r}{T_c - T_r} \quad \text{or} \quad T_{aw} = T_r - \eta \cdot (T_r - T_c) \tag{7}$$

Substituting Eq. (5) into Eq. (1) yields

$$q = h_c \cdot (T_r - T_w - \eta \cdot (T_r - T_c)) \tag{8}$$

Dividing by (T_r-T_c) yields

$$\frac{q}{T_r - T_c} = h_c \cdot \frac{T_r - T_w}{T_r - T_c} - h_c \cdot \eta \tag{9}$$

Since the temperatures and heat flux levels change considerably during the experiment, a wide range of values is obtained, as illustrated in Fig. 9. Assuming that h_c and η are not functions of temperatures, Eq. (9) is a linear relation between the fraction on the left hand side and the temperature ratio on the right hand side. The slope is the heat transfer coefficient, and the intercept at q=0 is the film effectiveness. The recovery temperature is calculated by subtracting the average T_d of 6.9°C from the freestream total temperature. The coolant temperature is determined using the mass flow averaged exit coolant temperatures from the three rows of cooling holes affecting the suction side. A representative example is shown in Fig. 9. Since the data follows the linear interpolation closely, it can be stated that the assumptions leading up to this interpretation of the data are correct. Specifically, the heat transfer coefficient and the film effectiveness do not vary significantly throughout the run.

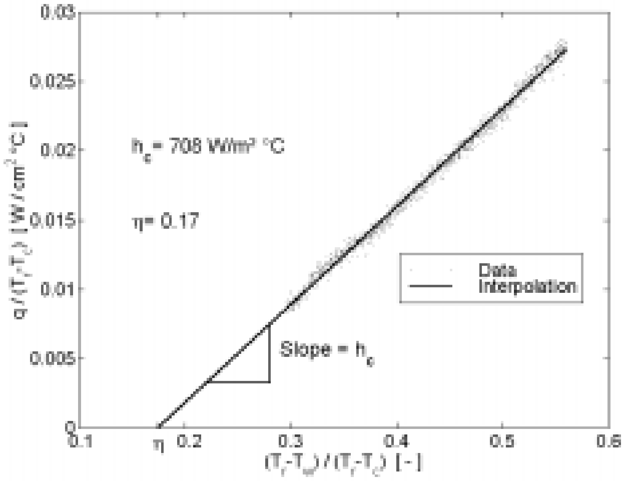
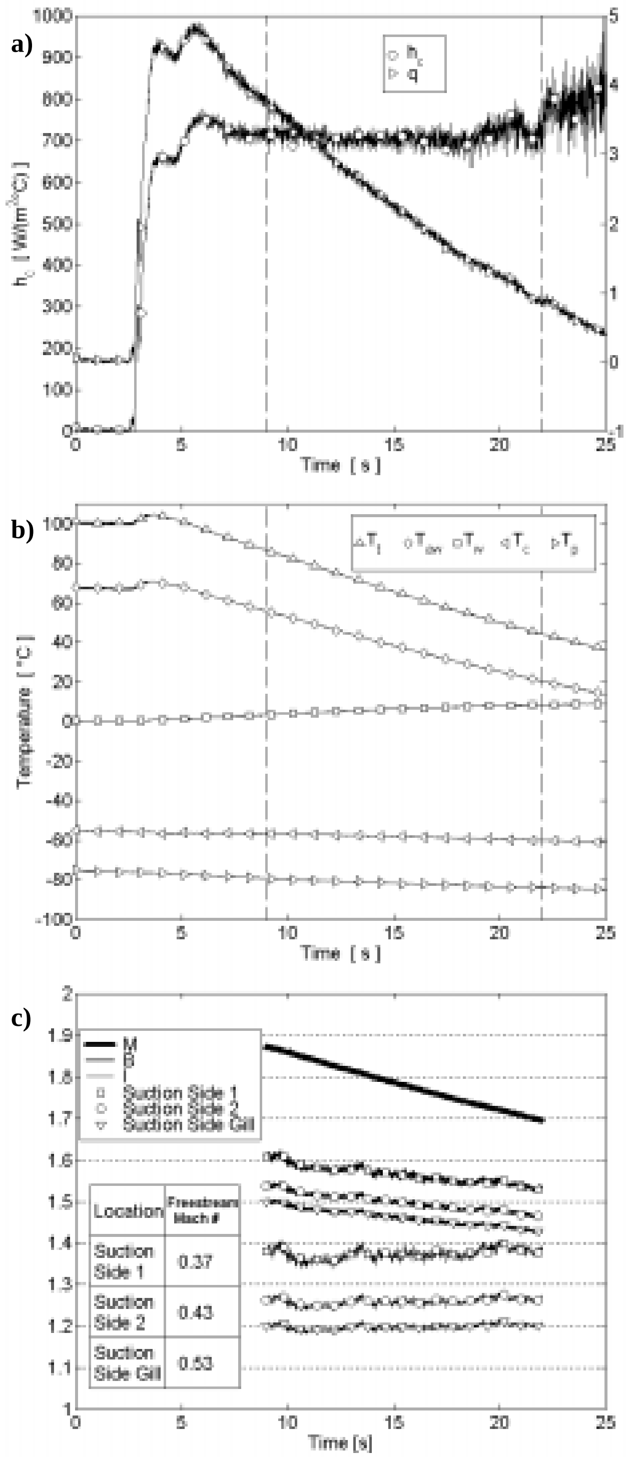


Figure 9: Interpolation for h_c and η (Run #5)

Fig. 10 a) shows the time history of the heat transfer coefficient calculated using the film effectiveness in Fig. 9. The heat transfer coefficient determined this way is very uniform throughout the time window used for the data analysis. In Fig. 10 b) all temperatures involved during this particular experiment are shown. T_{aw} is based on the film effectiveness determined in Fig. 9. It is evident that the coolant exit temperature is significantly higher than the coolant total temperature in the plenum indicating high heat transfer rates in the cooling holes. Fig. 10 c) gives the Density Ratio, Blowing Ratios and Momentum Ratios for all three cooling hole locations of interest. Since the Density Ratio is not a function of the local freestream Mach Number it is the same for all coolant exit locations. Since the total pressure ratio is kept relatively constant throughout the run, the

Momentum Ratios are close to uniform. Even though the Momentum Ratios are repeatable, the coolant total temperature varies somewhat from run to run, since there was no physical control for this parameter. As shown in Fig. 10 b) the freestream total temperature changes with time. That causes the decrease of Density Ratio and Blowing Ratio shown in Fig. 10 c). All Ratios are based on isentropic flow through the holes. Realistically, it would be hard to analyze the flow through the cooling holes since not only frictional effects but also high heat transfer rates (see Fig. 10 b)) would have to be taken into account. Based on coolant mass flow measurements an average discharge coefficient of 0.66 for all cooling holes was determined.

The average heat transfer coefficient with film cooling (686 W/(m^2°C)) is 8% higher than h without film cooling (634 W/(m^2°C)). Using a thermal conductivity of 0.030 W/mK and the axial chord (0.114 m) to obtain the Nusselt number, yields values of 2410 and 2610 for the experiments without and with film cooling, respectively. The average film effectiveness of 15.3% appears to be rather low.



Location	Freestream Mach #
Suction Side 1	0.37
Suction Side 2	0.43
Suction Side Gill	0.53

Figure 10: Time Histories from Cooled Run #5
a) Heat Transfer and Heat Transfer Coefficient
b) All Relevant Temperatures
c) Density Ratio, Blowing Ratio, Momentum Ratio
(Dashed Lines Indicate Range of Useful Data)

In low speed cascade tests with one closely spaced row of holes on the suction side of a large scale blade model, Ito et al. (1978) found comparably low values for film effectiveness for similar Momentum Ratios and gauge locations. Values between about 4% and 17% are found for gauge locations between x/D=10 and x/D=15 and Momentum Ratios between 1.0 and 2.3.

An error analysis for the method presented shows that the heat transfer coefficient is sensitive to bias errors in temperature and heat transfer measurement. Therefore, a larger scatter in the experimental results can be expected. The film effectiveness is also subject to higher scatter, as it is calculated from the intercept and the heat transfer coefficient. Results for heat transfer coefficient and film effectiveness are shown in Table 2. If the theoretical value for T_d (4.2°C) was used instead of the experimental value (6.9°C) the results for h_c and η would be a few percent higher.

Run #	11	12	13	14	15	16	17	18
h_c [W/m^2°C]	709	623	672	715	708	704	685	675
η [%]	16.6	12.0	17.0	16.6	16.8	14.6	15.1	13.5

Table 2: Run-to-Run Variation of h_c and η

SHOCK PASSING DATA ANALYSIS AND RESULTS

Shock Passing without Film cooling

For the analysis of the passing shock event Eq. (2) will be rewritten in such a way that all properties that are a function of time will be broken down into a mean value before shock impact (no superscript) and a time varying component (superscript ´). The unsteady heat flux during the event of a passing shock can then be expressed in terms of mean and fluctuating components:

$$q + q' = (h + h') \cdot (T_r + T_r' - T_w) \qquad (10)$$

The wall temperature does not change during the short duration of the shock passing. Expanding the right hand side of Eq. (10) yields:

$$q + q' = h \cdot (T_r - T_w) + h' \cdot (T_r - T_w) + h \cdot T_r' + h' \cdot T_r' \qquad (11)$$

Subtracting the mean heat flux on both sides yields the fluctuating component of heat transfer:

$$q' = h' \cdot (T_r - T_w) + h \cdot T_r' + h' \cdot T_r' \qquad (12)$$

The goal of this investigation is to determine the time variation of heat transfer coefficient (h´) and recovery temperature (T_r´) during the shock event which constitute the three components of unsteady heat transfer on the right hand side of Eq. (12). In Fig. 11, the traces of heat flux during a shock passing event are shown for different runs. The numbering corresponds to the run numbers in the steady state experiments (Table 1). The different levels of heat flux before the run are due to the fact that the shock is purposely initiated at different times during the run, i.e. at different temperature levels. Run #10 is not listed in Table 1 since the temperature differences were intentionally kept very small for this particular experiment. Therefore h and T_d could not be determined from this run. The time history of heat flux for times later than about 400 µs is of no interest since it is dominated by the interaction of shock reflections and later the contact surface emerging from the shock tube. These phenomena are not observed in the engine. Therefore, this investigation focuses on the impact of the first shock front primarily.

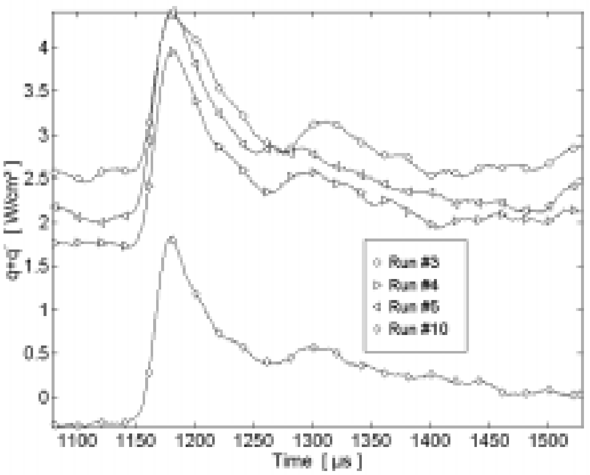


Figure 11: Heat Flux Traces from all Uncooled Experiments

In Fig. 12, the mean components of heat flux have been removed, leaving the traces of q´ indicated on the left y-axis. Apparently, all the fluctuating components of heat flux are similar.

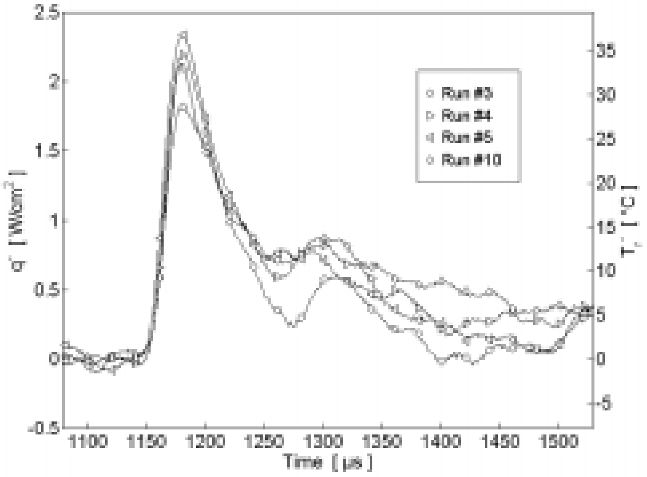


Figure 12: Unsteady Heat Flux and Recovery Temperature from all Uncooled Experiments

The first term on the right hand side of Eq. (12), h´·(T_r-T_w), indicates that the unsteady heat flux (q´) is a function of the overall temperature difference (T_r-T_w) if h´ is significant. If q´ is a function of (T_r-T_w), then the maximum or peak heat flux (q'_{max}) would also have to be a function of this temperature difference. In Fig. 13 the peak heat flux q'_{max} is plotted versus (T_r-T_w).

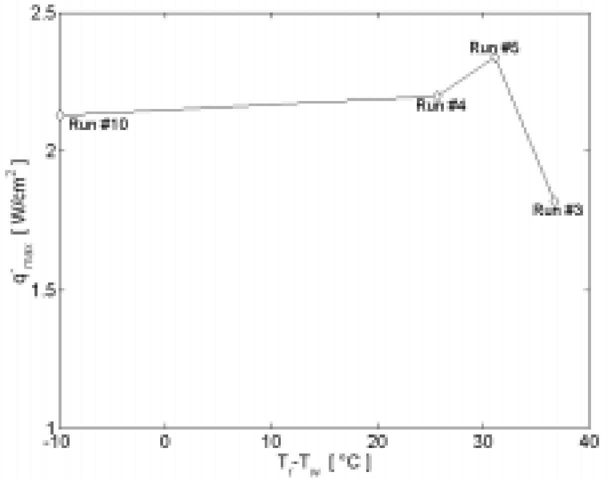


Figure 13: Peak Heat Flux vs. (T_r-T_w) from all Uncooled Experiments

There is no clear correlation between the two variables. Hence, q'_{max} does not strongly dependent on (T_r-T_w). This can only be the case if h´ is much smaller than h. Assuming h´ to be negligible and dropping all the terms containing h´ on the right hand side of Eq. (12), leaves an equation for the unsteady change of recovery temperature

$$T_r' = \frac{q'}{h} \tag{13}$$

Fig. 12 shows the time histories of this temperature variation indicated on the right y-axis. The heat transfer coefficient used here is the mean of the results of all the steady experiments. Since the scatter is relatively small, an average time variation of recovery temperature is used in the analysis of the experiments with film cooling.

Shock Passing with Film cooling

When film cooling is present, Eq. (12) still applies with the recovery temperature now replaced by the adiabatic wall temperature

$$q' = h_c' \cdot (T_{aw} - T_w) + h_c \cdot T_{aw}' + h_c' \cdot T_{aw}' \tag{14}$$

Analogous to the analysis for the uncooled case, it is the aim of this investigation to quantify the contribution of T'_{aw} and h'_c to the overall variation of heat flux. T'_{aw} can be expressed in terms of fluctuating components of recovery temperature (T'_r) and film effectiveness (η´)

$$T_{aw}' = T_r' \cdot (1-\eta) - \eta' \cdot (T_r - T_c) - \eta' \cdot T_r' \tag{15}$$

A further question of interest is how much the change in recovery temperature and film effectiveness affect the variation of T_{aw}. The time variation of recovery temperature is one of the results of the uncooled unsteady investigation. It is the goal of this investigation to determine the contributions of T'_r, h'_c, and η´ to the unsteady heat transfer. The coolant temperature is considered to be a constant.

Fig. 14 shows the pressure traces recorded by the blade mounted Kulite pressure transducer for representative experiments with and without film cooling. All the time histories of static pressure are very repeatable, asserting that the comparison of different runs is possible.

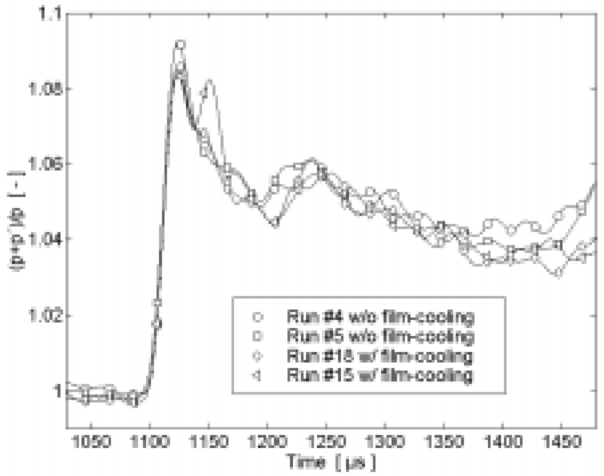


Figure 14: Pressure Traces from Representative Experiments with and without Film cooling

In Fig. 15, all the heat flux traces at shock impact are shown. Again, the differences in heat flux level are due to different temperature levels at the time of the shock release. Run #19 is not listed in Table 2. Intentionally, the temperature differences were kept very small and neither heat transfer coefficient nor film effectiveness could be determined from this experiment.

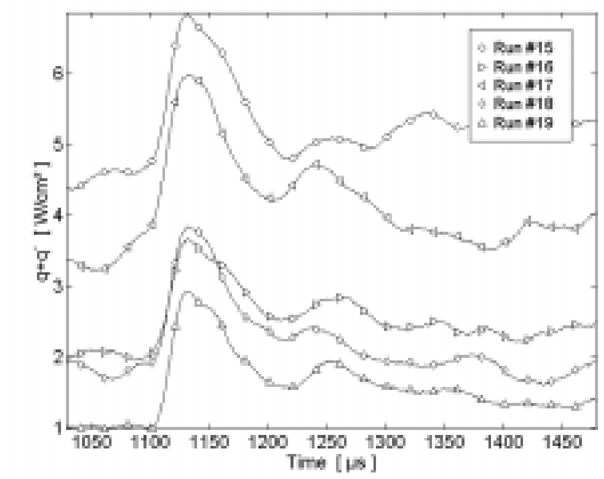


Figure 15: Heat Flux Traces from all Cooled Experiments

In Fig. 16, the same traces are shown after their mean values before shock impact have been removed. The traces of unsteady heat flux with and without film cooling (Figs. 16 and 12, respectively) are very similar in terms of magnitude and shape. Hence, the modes of heat transfer for both cases must be similar. The shock does not seem to influence the heat transfer coefficient or the mixing in the boundary layer (η) in any significant manner, otherwise these time histories would have to appear different for the cases with and without film cooling. Furthermore, pictures taken with the high-speed camera indicate that the cooling-film is not severely affected by the passing shock.

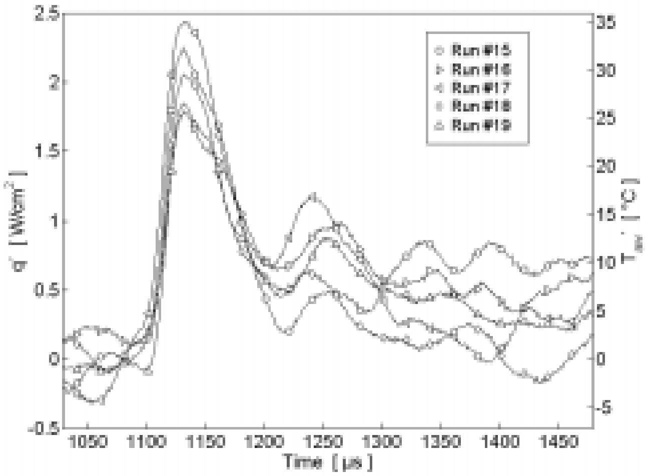


Figure 16: Variation of Heat Flux and Adiabatic Wall Temperature from all Cooled Experiments

The unsteady heat flux as expressed in Eq. (14) contains the term $h'_c \cdot (T_{aw}-T_w)$. It represents the first-order term of the contribution of h'_c to the unsteady heat flux. If h'_c is significant, the unsteady heat flux and the peak unsteady heat flux q'_{max} should correlate with $(T_{aw}-T_w)$. In Fig. 17 the peak heat flux is plotted against $(T_{aw}-T_w)$.

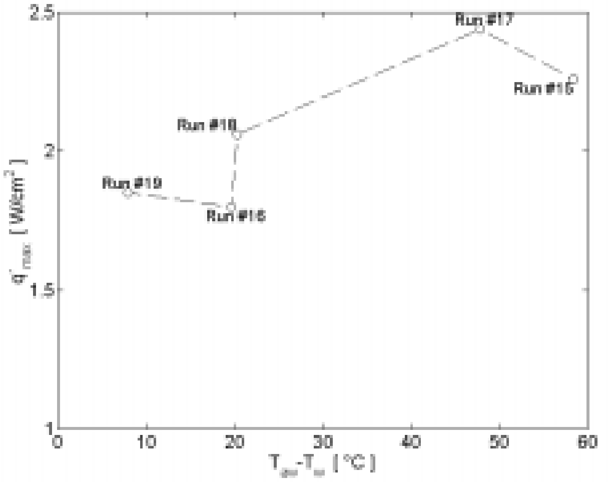


Figure 17: Peak Heat Flux vs. $(T_{aw}-T_w)$ from all Cooled Experiments

There is no strong dependency between q'_{max} and $(T_{aw}-T_w)$ evident. This can only be the case if h'_c is of minor significance. Neglecting all the terms containing h'_c in Eq. (14) yields a relation between the fluctuating components of T_{aw} and q

$$T'_{aw} = \frac{q'}{h_c} \qquad (16)$$

The traces of adiabatic wall temperature calculated from Eq. (16) for all the runs are shown in Fig. 16 scaled on the right y-axis. The similarity of the fluctuations seems to allow for an ensemble averaging of the different runs, shown in Fig. 18. Eq. (15) contains an expression for the contribution of T'_r to the variation of the adiabatic wall temperature

$$T'_{aw} = T'_r \cdot (1-\eta) - \eta' \cdot (T_r - T_c) - \eta' \cdot T'_r \qquad (15)$$

The averaged time variation of T_{aw} and the first term on the right side of Eq. (14) $(T'_r \cdot (1-\eta))$ are shown in Fig. 18. For the time variation of recovery temperature, the ensemble average from the uncooled tests are used. The film effectiveness used is the averaged result from the steady film cooled experiments.

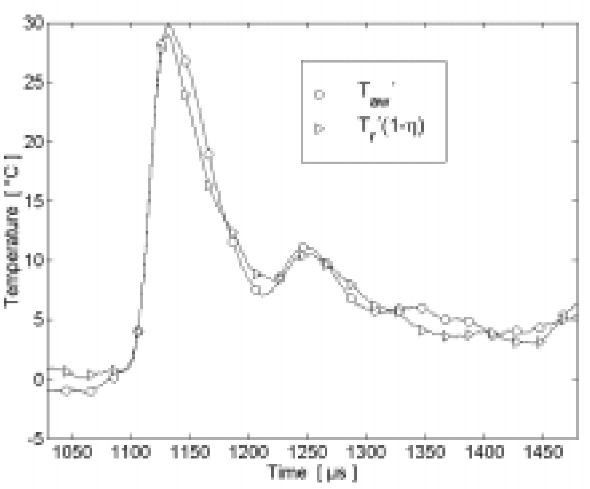


Figure 18: Variation of T_{aw} and the Contribution of T_r

The two traces are of very similar magnitude and shape. This suggests that the remaining terms on the right side of Eq. (14) $(\eta' \cdot (T_r - T_c)$ and $\eta' \cdot T_r')$ are small and consequently η' is small. It has to be concluded that the major contribution to the unsteady heat flux is the change of recovery temperature. The variations of h, h_c, and η have to be considered secondary effects. For the uncooled case this has been stated before by Moss et al. (1995). They performed on-rotor measurements of pressure and heat flux. Calculating the change of relative total temperature from the pressure measurement and assuming a constant heat transfer coefficient to calculate heat transfer, they found very good agreement between this calculation and the actual measurement. The present study suggests that this observation is also true for film cooled blades. Extrapolating to engine application, this implies that the time-averaged increase of heat transfer caused by passing shocks is small, since the relative total temperature is by definition varying around its mean value.

CONCLUSIONS

An experimental setup for the investigation of steady and unsteady heat transfer on film cooled transonic turbine blades was designed and built. For uncooled blades, one experiment in the transient facility was shown to be sufficient for the determination of heat transfer coefficient and recovery temperature. For film cooled blades, a method was presented to obtain heat transfer coefficient and film effectiveness from one experiment.

An analysis of the time resolved shock passing event with and without film cooling showed that the major contribution to the unsteady heat transfer is due to the fluctuation of recovery temperature caused by the shock. Heat transfer coefficient and film effectiveness were shown not to vary significantly during the interaction of the shock with the blade surface.

ACKNOWLEDGMENT

This work was supported by the Air Force Office of Scientific Research (AFOSR) under grant F08671-9601062, monitored by Dr. Jim M. McMichael and Dr. Mark Glauser. We would like to thank Messrs. Scott Hunter, Monty Shelton and Mark Pearson of General Electric Aircraft Engines for their collaboration on this project.

REFERENCES

Johnson, A.B., Oldfield, M.L.G., Rigby, M.J., Giles, M.B., 1990, "Nozzle Guide Vane Shock Propagation and Bifurcation in a Transonic Turbine Rotor", ASME 90-GT-310

Moss, R.W., Sheldrake, C.D., Ainsworth, R.W., Smith, A.D., Dancer, S.N., 1995, "Unsteady Pressure and Heat Transfer Measurements on a Rotating Blade Surface in a Transient Flow Facility", AGARD CP-571 pp. 22.1-22.9

Moss, R.W., Ainsworth, R.W., Garside, T., 1997, "Effects of Rotation on Blade Surface Heat Transfer: An Experimental Investigation", ASME 97-GT-188

Horton, F.G., Schultz, D.L., Forest A.E., 1985, "Heat Transfer Measurements With Film Cooling on a Turbine Blade Profile Cascade", ASME 85-GT-117

Hilditch, M.A., Smith, G.C., Anderson, S.J., Chana, K.S., Jones, T.V., Ainsworth, R.W., Oldfield, M.L.G., 1995, "Unsteady Measurements in an Axial Flow Turbine" AGARD CP-571 pp. 24.1-24.8

Abhari, R.S., Epstein, A.H., 1992, "An Experimental Study of Film Cooling in a Rotating Transonic Turbine", ASME 92-GT-201

Heidmann, J.D., Lucci, B.L., Reshotko, E., 1997, "An Experimental Study of the Effect of Wake Passing on Turbine Blade Film Cooling", ASME 97-GT-255

Hale, J.H., Diller, T.E., Ng, W.F., 1997, "Effects of Wake on Turbine Blade Heat Transfer in a Transonic Cascade", ASME 97-GT-130

Nix, A.C., Reid, T., Peabody, H., Ng, W.F., Diller, T.E., Schetz, J.A., 1997, "Effects of Shock Wave Passing on Turbine Blade Heat Transfer in a Transonic Cascade", AIAA-97-0160

Diller, T.E., 1993, "Advances in Heat Flux Measurement", Advances in Heat Transfer, Vol. 23, Ads, J.P., Hartnett et al., Academic Press, Boston, 1993, pp. 279-368

Ito, S., Goldstein, R.J., Eckert, E.R.G.,"Film Cooling on a Gas Turbine Blade", Journal of Engineering for Power, 1978, Vol. 100, pp. 476-481

A COMPARISON OF RADIATION VERSUS CONVECTION CALIBRATION OF THIN-FILM HEAT FLUX GAUGES

D. E. SMITH, J. V. BUBB, O. POPP, T.E. DILLER
Virginia Polytechnic Institute and State University
Blacksburg, VA 24061

STEPHEN J. HEVEY
Vatell Corporation
Christiansburg, VA 24073

ABSTRACT

A transient, in-situ method was examined for calibrating thin-film heat flux gauges using experimental data generated from convection and radiation tests. Also, a comparison is made between this transient method and the standard substitution calibration technique. Six Vatell Corporation HFM-7 type heat flux gauges were mounted on the surface of a 2-D, first-stage turbine rotor blade. These gauges were subjected to radiation from a heat lamp and in a separate experiment to a convective heat flux generated by flow in a transonic cascade wind tunnel. A second set of convective tests was performed using jets of cooled air impinging on the surface of the gauges. Direct measurements of both the time-resolved heat flux and surface temperature on the blade were simultaneously taken. The heat flux input was used to predict a surface temperature response using a one-dimensional, semi-infinite conduction model into a substrate with known thermal properties. The sensitivities of the gauges were determined by correlating the semi-infinite predicted temperature response to the measured temperature response. A finite difference code was used to model the penetration of the heat flux into the substrate in order to estimate the time for which the semi-infinite assumption was valid. The results from these tests showed that the gauges accurately record both the convection and radiation modes of heat transfer. The radiation and convection tests yielded gauge sensitivities that agreed to within ±17%.

NOMENCLATURE

Symbols

C	specific heat of substrate (J/kg·°C)
S_{conv}	convection tests gauge sensitivity (µV/W/cm^2)
S_{sub}	standard substitution gauge sensitivity (µV/W/cm^2)
S_{rad}	radiation tests gauge sensitivity (µV/W/cm^2)
S_{trans}	transient tests gauge sensitivity (µV/W/cm^2)
S_{jet}	impinging jet tests gauge sensitivity (µV/W/cm^2)
$S_{in\text{-}situ}$	in-situ tests gauge sensitivity (µV/W/cm^2)
T_{calc}	calculated temperature (°C)
T_{exp}	experimental temperature (°C)
k	thermal conductivity of substrate (W/m·°C)
q_{calc}	calculated heat flux (W/cm^2)
q_{exp}	experimental heat flux (W/cm^2)
t	time (s)
ε	emissivity of coating applied to gauges
ρ	density of the substrate (kg/m^3)
τ	time allowance for semi-infinite assumption (s)

INTRODUCTION

The accurate measurement of heat flux into a surface is very important to researchers concerned with thermal systems. For example, information about the steady and unsteady heat transfer into turbine blades is in high demand by the turbine industry and its thermal designers. However, the difficulties involved in measuring high-speed thermal phenomena limit the available information. Another area where heat flux measurements are necessary is in the determination of material properties. If the incident heat flux on a material and the corresponding temperature rise are known, the thermal properties of that material may be determined. These are only two of the many areas in which heat flux measurements are critical. The accurate measurement of heat flux is usually a challenge, however.

One difficulty is that the installation of a gauge on a surface will always result in a disturbance to the thermal system being investigated. This disturbance can be minimized if the thermal resistance of the gauge is similar to that of the material in which it is embedded, and if the area to thickness ratio of the gauge is large enough to promote one-dimensional conduction through the gauge. Another difficulty in measuring heat flux is the calibration of the gauges. Moffatt (1995) said that errors in the estimate of gauge sensitivities on the order of ±10% are reasonable due to the difficulties involved with calibration. Moreover, it is often difficult or impossible to calibrate a gauge once it has been mounted in an experimental setup.

One of the present methods for calibrating heat flux gauges consists of a substitution technique where the gauge to be calibrated is subjected to a known radiative heat flux and a direct calibration of the gauge is made. In addition to the substitution calibration, a transient in-situ technique can be used with gauges which have a fast response time provided the temperature is known as well [Hager et al., 1994]. It has been shown by Baker and Diller (1993) that the surface temperature time history can be calculated from the time history of the measured heat flux. This calculated temperature can then be used in a transient method which assumes a one-dimensional, semi-infinite model to calibrate the heat flux gauges.

For this study, six heat flux microsensor (HFM) thin-film gauges, produced by Vatell, Inc., were mounted on the surface of a first-stage turbine rotor blade made of aluminum. The gauges were used to measure, simultaneously, the local time-resolved heat flux and surface temperature. The HFM is equally responsive to both radiation and convection heat fluxes, so both modes of heat transfer can be used to provide incident fluxes to the gauges for calibration purposes. The transient calibration technique can be used with the data generated by applying high levels of heat flux to the gauges mounted in the aluminum blade. However, it is difficult to use the substitution technique on these gauges while they are in the blade so a different set of gauges were exposed to a radiative flux and the data was used to perform calibrations using both the substitution and the transient techniques. A comparison was then made between the two calibration methods for both convection and radiation.

INSTRUMENTATION

Heat Flux Microsensors

The HFM is composed of two separate sensors, a resistance temperature sensor (RTS) and a heat flux sensor (HFS). The HFS uses a passive differential thermopile made up of 280 thermocouple pairs to generate a voltage proportional to the incident heat flux. The active area of the HFM is 4mm in diameter, two microns thick, and is deposited on an aluminum-nitride substrate, an electrically insulating material with thermal properties close to those of the aluminum blade. The heat flux gauge has been shown to have a time response on the order of 10μs [Holmberg, 1995]. The small thickness-to-area ratio and the thermal properties of the substrate ensure that the thermal disruption caused by the gauge will be minimal.

Two different types of HFM's were used in this work: HFM-7's and HFM-6's. The difference between the two types is in the material which is used to form the thermocouple pairs. The HFM-6 gauge uses a Nichrome/Platinum thermocouple pair and has a lower sensitivity ($S \approx 30\mu V/W/cm^2$) than the HFM-7 which uses a Nichrome/Constantan pair ($S \approx 150\mu V/W/cm^2$). All of the gauges use the same material for the substrate, however. The six gauges mounted in the aluminum blade are all HFM-7's and will hereafter be referred to as gauges B1 through B6. Of the five gauges used in the substitution experiments, two are HFM-7's and three are HFM-6's. These gauges will hereafter be referred to as gauges G7/1, G7/2, and G6/1 through G6/3, respectively.

Blade Instrumentation

The six HFM gauges were all mounted on the suction side surface of a generic two-dimensional, high-turning, first stage turbine rotor blade. The gauges were staggered along the chord of the blade as well as along its span. Each of the gauges was press-fitted flush with the surface of the blade. The curvature of the blade made using the substitution technique extremely difficult and the press-fit meant that the gauges could not be removed from the blade without a very high possibility of incurring damage to the gauges. Therefore, the transient technique was used in-situ to calibrate the six HFM's embedded in the blade.

Substitution Experiments

Radiation Bench Tests of Unmounted Gauges

The substitution technique is a steady state calibration which uses a reference heat flux gauge to measure the incident flux from a heat lamp and then substitutes an un-calibrated gauge for the known assuming the conditions are the same. An accurate calibration using the substitution method can not be made with the six gauges which were already embedded in the blade. The curvature of the blade makes it difficult to position the heat lamp the same with respect to both the reference gauge and the embedded HFM. Small changes in the relative positionings of the two gauges to the lamp can result in large calibration errors, since the incident heat flux differs with the exact orientation of the lamp. Therefore, five HFM gauges which had not been mounted in the blade were tested instead (G7/1-2 and G6/1-3). The magnitude of the sensitivities of three of these gauges (G6/1-3) was smaller due to the different materials used to make the thermocouple pairs for the HFS and RTS, but the accuracy of the sensitivity estimate was unaffected.

As the reference gauge, a circular-foil heat flux gauge painted with a high emissivity coating (0.94), was exposed to a heat lamp to determine the amount of heat flux generated by the lamp. Both the lamp and the circular-foil gauge were mounted in a fixture which maintained their relative orientation and positioning. The five HFM gauges were also coated, mounted in the fixture, and then exposed to the heat lamp as is shown in Fig. 5. Six tests were performed on each gauge. The data generated by each HFM was recorded at 2000Hz. Sensitivities determined using data from this test will be referred to as S_{sub}.

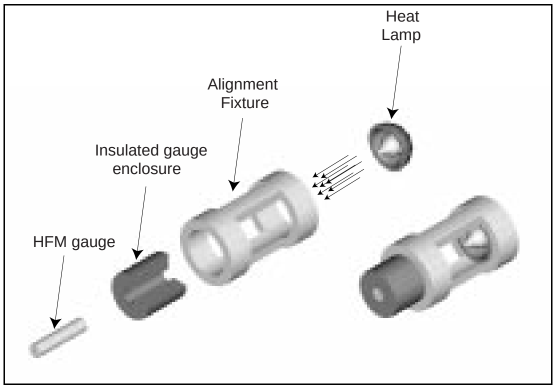


Figure 1: Radiation Tests for Substitution Method

Substitution Calibration

The substitution calibration is widely used in research and industry for the calibration of heat flux gauges. Its use in the calibration of HFM’s was documented by Hager et al. (1994). The known heat flux is applied to the gauge and the voltage output can be directly calibrated using equation (1).

$$S_{HFS}=\frac{V_{HFS}}{Q_{REF}} \qquad (1)$$

The reference gauge, a NIST calibrated circular-foil gauge, is accurate to within ± 3%.

Transient Experiments

In-situ Transonic Convection Tests

The mounted HFM's were used to investigate the convective heat transfer into a turbine blade, so some of the convection experiments were performed in the transonic blowdown wind tunnel at Virginia Tech. A set of resistance heaters are used to preheat copper tubes located in the path of the main flow upstream of the test section. The tubes act as a passive heat exchanger and allow the flow to be heated to high (120°C) temperatures before entering the cascade. Figure 2. shows the wind tunnel with the heating loop and the cascade test section. The cascade test section contains four full blades and two half blades, or five flow passages. Figure 3. shows the location of the instrumented blade in the cascade test section. The heated flow passing over the instrumented blade provides a large convective heat flux to the gauges. Six tunnel runs were made and the data generated by each gauge was recorded simultaneously. The HFM's were sampled at 100Hz and filtered at 50Hz. The wind tunnel is capable of running for up to 35s with the inlet total pressure controlled, but only a very small fraction of the data recorded by the HFM's is needed for the transient calibration. An important point to note is that the transient calibration technique can be used to perform an in-situ calibration each time a tunnel experiment is performed. The calibration performed at the beginning of an experiment can then be compared to the manufacturer's calibration to determine if the gauge's calibration has drifted. Sensitivities determined using data from this test will be referred to as $S_{in\text{-}situ}$.

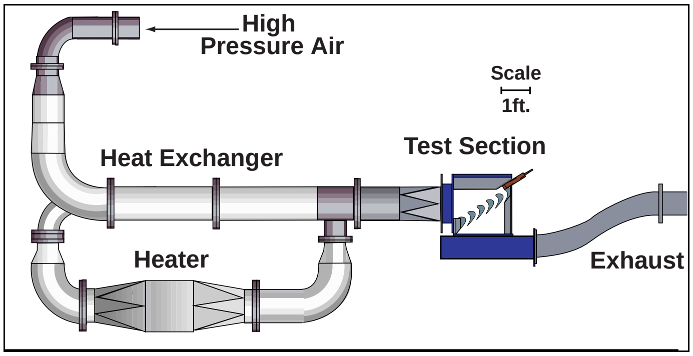


Figure 2: Wind Tunnel Facility

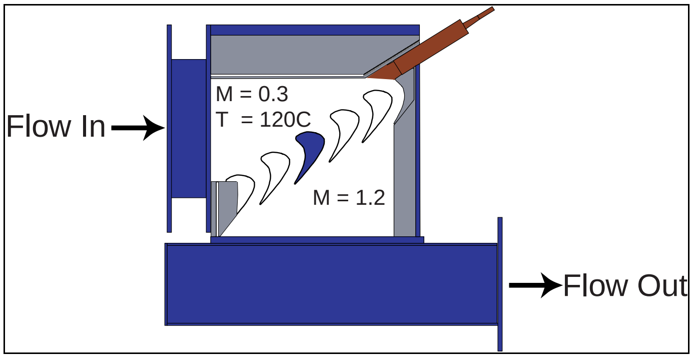


Figure 3: Cascade Test Section

Impinging Jet Bench Tests

In addition to the transonic convection experiments, which were performed in the blowdown wind tunnel, an impinging jet with cold (-100°C) air was used to provide a large negative incident heat flux to the gauges. These experiments required that the blade be taken out of the cascade test section. The jet was 0.5 inches in diameter and was mounted eight diameters from the blade as is shown in Fig 4. A large tank was loaded to 150 psi and used to supply pressurized air for the impinging jet. For the transient technique, the incident heat flux need not be known, so it was not necessary to measure the velocity of the jets. For each test, a shutter placed between the gauge and the jet was manually removed to provide a quick increase in the level of heat flux. The response of the HFM to the jet was recorded at 1000Hz and filtered at 50Hz. Separate tests were performed for each gauge. Sensitivities determined using data from this test will be referred to as S_{jet}.

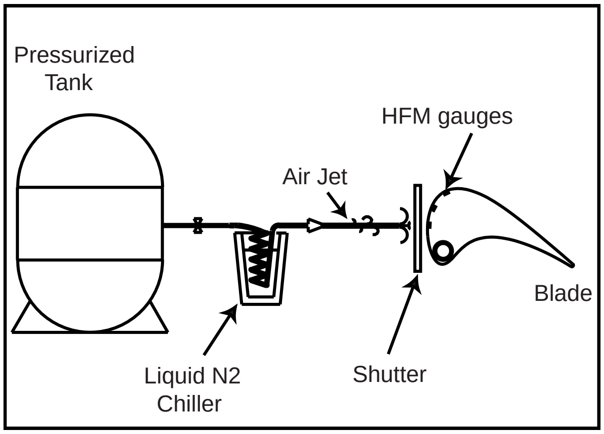


Figure 4: Jet Impingement on Blade

Radiation Bench Tests of Gauges in Blade

Another set of radiation bench tests was performed on the gauges in the instrumented blade after it had been removed from the cascade test section. All of the gauges were painted with a high emissivity coating (ε = 0.94), and a heat lamp was used to subject each gauge to an incident heat flux as shown in Fig. 5. The heat lamp was positioned directly over each gauge and the radiative heat flux was made to simulate a step input by using a shuttering mechanism; once the lamp had reached its full illumination, the shutter was quickly removed and the gauge was exposed to the incident flux. Again, it should be noted that when using the transient calibration, the incident heat flux does not need to be a known value. It is therefore not necessary to position the lamp in the exact same orientation over each gauge as was required with the substitution method. This test was performed once on each gauge and the data generated by each HFM was recorded at 2000Hz, unfiltered. Sensitivities determined using data from this test will be referred to as S_{rad}.

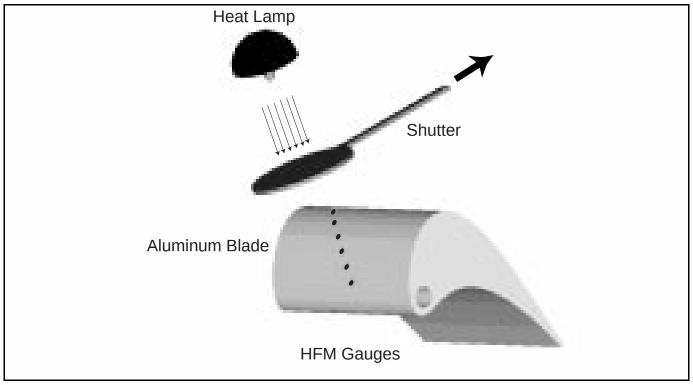


Figure 5: Transient Radiation Tests

Radiation Bench Tests of Unmounted Gauges

The data generated by applying radiative heat flux to the unmounted gauges as described earlier in the substitution experiments section can also be analyzed using the transient method. To do this, data was sampled at 2000Hz, unfiltered, in order to record the initial response of the HFM to the incident heat flux. Sensitivities determined using data from this test will be referred to as S_{trans}.

Transient Calibration

The transient method of calibrating heat flux gauges was used to analyze the data generated by all of the tests described earlier, including the substitution bench tests. The transient calibration makes use of the fact that for very short time periods, the surface heat flux can be calculated from the surface temperature if a one-dimensional, semi-infinite model is assumed [Diller, 1993]. The time period is usually a fraction of a second, and so time-resolved temperature measurements must be taken in order to use this method of calibration. The time response of the HFM gauge is on the order of 10µs, so it is capable of providing these time-resolved measurements for both the surface temperature and the heat flux simultaneously. Baker and Diller (1993) developed a method of determining surface temperature from heat flux using a Green's function approach. By assuming the substrate to be initially at a uniform temperature, T_o , and treating the series of heat flux data points, q_j , as impulses, the surface temperature time history can be reconstructed using equation (2),

$$T_{calc}(t_n) - T_o = \frac{2}{\sqrt{\pi}\sqrt{k\rho C}} \sum_{j=0}^{n-1} q_j \left[\sqrt{t_n - t_j} - \sqrt{t_n - t_{j+1}}\right] \qquad (2)$$

Equation (2) shows that the transient technique requires that the thermal properties of the substrate be known. An algorithm was written to determine the sensitivity of the HFS by minimizing the temperature difference between the measured and calculated temperature histories, T_{exp} - T_{calc}.

Validity of 1-D, Semi-infinite Assumption

The use of the transient method is dependent upon the validity of the one-dimensional, semi-infinite conduction model. Therefore, it is necessary to determine how long this assumption may be considered valid. The active area of the HFM gauge is a circle 4mm in diameter; the HFS occupies the center of that area and the RTS is laid down in a ring surrounding it. The HFS and the RTS are located close enough to each other that it might be assumed that they measure the same thermal phenomena. The two-dimensional heat transfer effects, however, are felt first at the edge of the sensor. Therefore, the difference of the temperature at this location from the one-dimensional model is the limiting case for the transient calibration technique. A two-dimensional finite difference conduction code was written to model the penetration of a step input of heat flux into the substrate. The code was used to determine the maximum time, τ , for which the one-dimensional, semi-infinite assumption was valid.

The parameters used in the code are shown in Fig. 6. The heat flux step input of 10^5 W/m^2 used is typical of the levels measured during heated runs in the transonic blowdown wind tunnel and of the heat lamp used in the radiation bench tests. The step input was applied to a circular area with a diameter approximately four times that of the HFM gauge. This area is the same size as that of the heat lamp, which is the limiting case in this study since the wind tunnel provides a flux over a larger area and two-dimensional heat transfer effects are felt much later in time. The code used cylindrical coordinates to accurately model the gauge, and the thermal properties shown in the figure were used for both the gauge and the surrounding material. The code was discretized in time as well as space; a mesh size of 0.4mm and a time step of 0.005s were chosen.

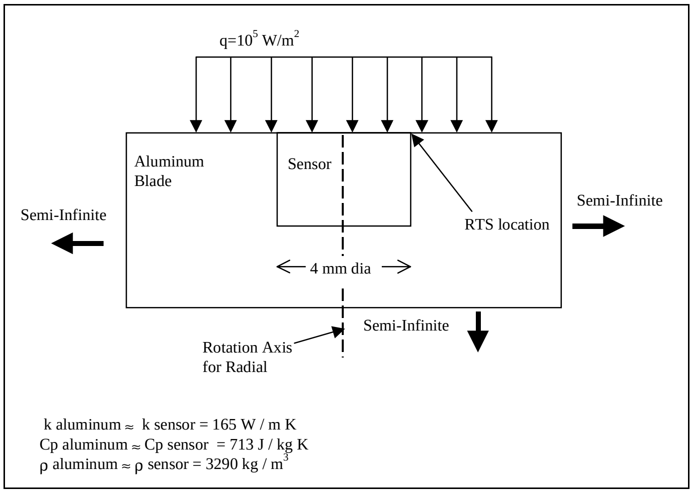


Figure 6: Finite Difference Model

The time history of the RTS surface temperature along with the solution of the one-dimensional, semi-infinite conduction model can be seen in Fig. 7. As time progresses, two-dimensional heat transfer effects are seen to become more important as the RTS temperature diverges from the one-dimensional model. In the two-dimensional code, thermal energy is being transported away radially as well as axially, so the temperature predicted at the location of the RTS is less than that of the model which has no radial transport of thermal energy. A time, τ, was chosen such that the error in calculated heat flux from the one-dimensional, semi-infinite model was less than 5%. For the parameters shown in Fig. 6., τ was found to be 125ms.

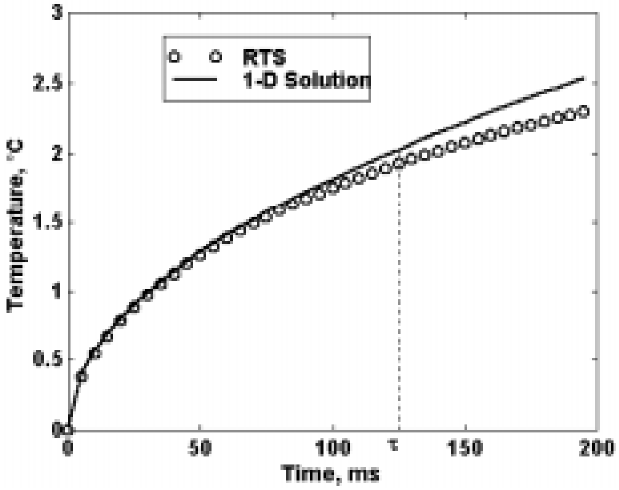


Figure 7: Determination of Time Allowance, τ

RESULTS

A comparison of the substitution calibration method with the transient method was done first to determine if the transient method is viable as a calibration technique. The data generated in the substitution experiments were analyzed using both the substitution technique and the transient technique. The product of the material properties ($[k \cdot \rho \cdot C]^{1/2}$ = 19,670 in SI units) used in the transient technique are 10% higher than those shown in Fig. 6 due to a change in the aluminum nitride supplier. Six tests were performed for each gauge. The calibration results for both methods can be seen in Table 1. As mentioned earlier, gauges G6-1 through G6-3 are of a different HFM type and so the magnitude of their sensitivities are smaller than those of the other gauges. The difference in the magnitude of sensitivity between HFM-7's and HFM-6's is solely a function of the thermocouple materials used. All of the gauge sensitivities determined with the transient method were found to be in excellent agreement with the substitution results. Experimental uncertainty for each gauge was calculated by establishing a 90% confidence interval around the mean of the six tests performed using a student-t distribution.

Table 3: Results of Substitution Calibration

Gauge Number	S_{sub}	S_{trans}
G6-1	18.01	17.91 ±1.3%
G6-2	15.33	16.58 ±7.3%
G6-3	25.50	28.72 ±5.1%
G7-1	108.6	111.33 ±9.7%
G7-2	189.05	191 ±9.2%

These results show that the transient calibration technique is in excellent agreement with the substitution technique used in most calibrations of heat flux gauges.

Since the transient method is a viable calibration method, the suitability of convection tests, as opposed to the radiation tests commonly used, was investigated. To do this, two different sets of convection tests were performed and compared to a radiation bench test. All three sets of tests were performed on the HFM's embedded in the blade. One set of tests was performed in the Virginia Tech transonic cascade wind tunnel. The blade was then removed from the test section and the two bench tests, convection using a cold impinging jet and radiation using a heat lamp, were performed. Two convection bench tests and one radiation bench test were made. The data from all three sets of tests were analyzed according to the transient calibration outlined earlier. Figure 8 is a plot of the measured heat flux and surface temperature along with the calculated surface temperature for a sample radiation bench test. The calculated temperature history is a smooth curve underlying the noisier history of the measured surface temperature. The calculated temperature has less noise due to the integration process used to construct it (Baker and Diller, 1993). Figure 8 shows the strength of the transient calibration method; the time histories of the measured surface temperature and the calculated temperature are in excellent agreement, which they should be if an accurate calibration is done.

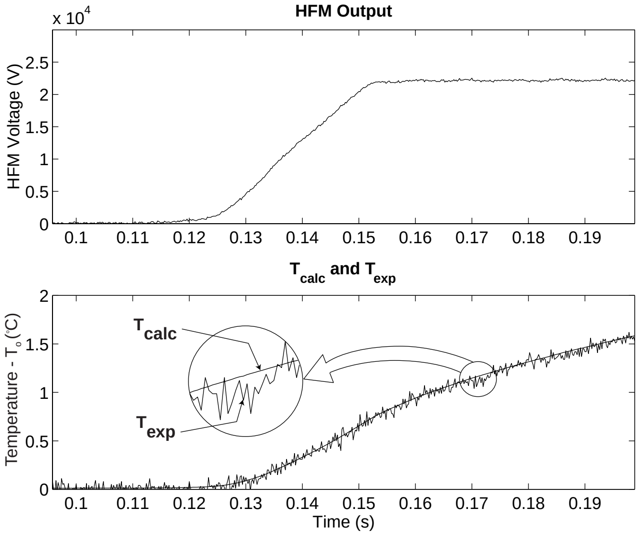


Figure 8: Results of Transient Calibration

The results from all three sets of tests can be seen in Table 2. The results for S_{rad}, $S_{in\text{-}situ}$, and S_{jet} are the mean values over all tests for each gauge. The values listed as S_{conv} are the combined results of both the in-situ and impinging jet convection tests. The uncertainty for each gauge is once again calculated by determining the 90% confidence interval around the mean. With the exception of gauge B2, the S_{rad} values all lie within the confidence intervals of the combined convection results.

Table 4: Results of Transient Calibration

Gauge	S_{rad}	$S_{in\text{-}situ}$	S_{jet}	S_{conv}
B1	264	233	274	243 ±17%
B2	114	105	102	104 ±3.7%
B3	148	147	146	147 ±5.2%
B4	150	129	152	135 ±15.2%
B5	177	176	177	176 ±8.8%
B6	155	153	143	150 ±10.7%

CONCLUSIONS

Accurate calibrations are necessary in order to investigate the thermal phenomena reported by heat flux gauges. These calibrations are difficult to make and often impossible to repeat once the gauge has been installed in an experimental setup. A transient method of calibrating thin-film heat flux gauges with fast time responses has been shown to be a viable calibration technique. The strengths of the transient calibration technique are that it can be performed in-situ, the incident heat flux need not be known, and that gauge calibrations can be performed at the start of each run if an appropriate level of heat flux is present.

ACKNOWLEDGMENT

This work was supported by the Air Force Office of Scientific Research (AFOSR) under grant F08671-9601062, monitored by Dr. Jim M. McMichael and Dr. Mark Glauser. We would like to thank Messrs. Scott Hunter, Monty Shelton and Mark Pearson of General Electric Aircraft Engines for their collaboration on this project.

REFERENCES

Johnson, L.P., Diller, T.E., 1995, “Measurements With a Heat Flux Microsensor Deposited on a Transonic Turbine Blade,” IEEE 95CH3482-7.

Diller, T.E., 1996, “Methods of Determining Heat Flux From Temperature Measurements”, ISA 0227-7576.

Hager, J.M., Terrell, J.P., Silverston, E., Diller, T.E., 1994, “In-Situ Calibration of a Heat Flux Microsensor Using Surface Temperature Measurements”, ISA 94-1034.

Holmberg, D.G., Diller, T.E., 1996, “High-Frequency Heat Flux Sensor Calibration and Modeling”, ASME *Journal of Fluids Engineering*, Vol. 117, pp. 659-664.

Baker, K.I., and Diller, T.E., 1993, "Unsteady Surface Heat Flux and Temperature Measurements." ASME 93-HT-33.

Moffat, R.J., and Danek, C., 1995, "Calibrating Heat Flux Gauges for Convection Applications", NIST/NSF Workshop on Heat Flux Transducer Calibration.

Schultz, D.L. and Jones, T.V., 1973, "Heat-Transfer Measurements in Short-Duration Hypersonic Facilities", AGARD AG-165.

Appendix B: Computer Programs

B-1 HFS Calibration

- **frontend.m**
- **frontend2.m**
- **qtodave.m**

B-2 Data Processing

- **datasum2.m**
- **data1123b.m**
- **data922b_ambient.m**

Appendix B-1: HFS Calibration

Frontend.m

```
close all
%Inititalize Section
S=NaN;
PREPT=NaN;

a = figure('Units','points', ...
      'Color',[0.8 0.8 0.8], ...
   'Position',[168.75 88.5 537 418.5], ...
      'Tag','Fig1');
% Temperature axis
axs(1) = axes('Parent',a, ...
      'Units','points', ...
      'CameraUpVector',[0 1 0], ...
      'CameraUpVectorMode','manual', ...
      'Color',[1 1 1], ...
      'Position',[54 305.25 170.25 96.75], ...
      'Tag','Axes2', ...
      'XColor',[0 0 0], ...
      'YColor',[0 0 0], ...
      'ZColor',[0 0 0]);
c = text('Parent',axs(1), ...
      'Color',[0 0 0], ...
      'HandleVisibility','callback', ...
      'HorizontalAlignment','center', ...
      'Position',[0.495575 1.05469 0], ...
      'String','Temperature', ...
      'Tag','Axes2Text1', ...
      'VerticalAlignment','bottom');
set(get(c,'Parent'),'Title',c);
% Flux Axes
axs(2) = axes('Parent',a, ...
      'Units','points', ...
      'CameraUpVector',[0 1 0], ...
      'CameraUpVectorMode','manual', ...
      'Color',[1 1 1], ...
      'Position',[54.75 168.75 170.25 96.75], ...
      'Tag','Axes2', ...
      'XColor',[0 0 0], ...
      'YColor',[0 0 0], ...
      'ZColor',[0 0 0]);
c = text('Parent',axs(2), ...
      'ButtonDownFcn','ctlpanel SelectMoveResize', ...
      'Color',[0 0 0], ...
      'HandleVisibility','callback', ...
      'HorizontalAlignment','center', ...
      'Interruptible','off', ...
      'Position',[0.495575 1.05469 0], ...
      'String','Flux', ...
      'Tag','Axes2Text1', ...
      'VerticalAlignment','bottom');
```

```
% Fit1 axis
axs(3) = axes('Parent',a, ...
      'Units','points', ...
      'CameraUpVector',[0 1 0], ...
      'CameraUpVectorMode','manual', ...
      'Color',[1 1 1], ...
      'Position',[300 20 170.25 96.75], ...
      'Tag','Axes2', ...
      'XColor',[0 0 0], ...
      'YColor',[0 0 0], ...
      'ZColor',[0 0 0]);
c = text('Parent',axs(3), ...
      'Color',[0 0 0], ...
      'HandleVisibility','callback', ...
      'HorizontalAlignment','center', ...
      'Position',[0.495575 1.05469 0], ...
      'String','Fit 1', ...
      'Tag','Axes2Text1', ...
      'VerticalAlignment','bottom');
set(get(c,'Parent'),'Title',c);
% Fit2 axis
axs(4) = axes('Parent',a, ...
      'Units','points', ...
      'CameraUpVector',[0 1 0], ...
      'CameraUpVectorMode','manual', ...
      'Color',[1 1 1], ...
      'Position',[54 20 170.25 96.75], ...
      'Tag','Axes2', ...
      'XColor',[0 0 0], ...
      'YColor',[0 0 0], ...
      'ZColor',[0 0 0]);
c = text('Parent',axs(4), ...
      'Color',[0 0 0], ...
      'HandleVisibility','callback', ...
      'HorizontalAlignment','center', ...
      'Position',[0.495575 1.05469 0], ...
      'String','Fit 2', ...
      'Tag','Axes2Text1', ...
      'VerticalAlignment','bottom');
set(get(c,'Parent'),'Title',c);

%Text Labels
b = uicontrol('Parent',a, ...
      'Units','points', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'HorizontalAlignment','left', ...
      'Position',[319.5 207 64.5 9.75], ...
      'String','Sample Freq (Hz)', ...
      'Style','text', ...
      'Tag','StaticText1');
b = uicontrol('Parent',a, ...
      'Units','points', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'HorizontalAlignment','left', ...
      'Position',[319.5 247 64.5 9.75], ...
      'String','Gauge #', ...
      'Style','text', ...
      'Tag','StaticText1');
b = uicontrol('Parent',a, ...
      'Units','points', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'HorizontalAlignment','left', ...
      'Position',[280 247 30 9.75], ...
      'String','Zero', ...
      'Style','text', ...
      'Tag','StaticText1');
b = uicontrol('Parent',a, ...
      'Units','points', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'HorizontalAlignment','left', ...
      'Position',[388.5 247 40.5 9.75], ...
      'String','HFS Col', ...
      'Style','text', ...
      'Tag','StaticText1');
b = uicontrol('Parent',a, ...
      'Units','points', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'HorizontalAlignment','left', ...
      'Position',[436.5 247 40.5 9.75], ...
      'String','RTS Col', ...
      'Style','text', ...
      'Tag','StaticText1');
b = uicontrol('Parent',a, ...
      'Units','points', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'HorizontalAlignment','left', ...
      'Position',[388.5 207 40.5 9.75], ...
      'String','HFS Gain', ...
      'Style','text', ...
      'Tag','StaticText1');
b = uicontrol('Parent',a, ...
      'Units','points', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'HorizontalAlignment','left', ...
      'Position',[436.5 207 40.5 9.75], ...
      'String','RTS Gain', ...
      'Style','text', ...
      'Tag','StaticText1');
b = uicontrol('Parent',a, ...
      'Units','points', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'HorizontalAlignment','left', ...
      'Position',[436.5 177 50 9.75], ...
      'String','Run time (ms)', ...
      'Style','text', ...
      'Tag','StaticText1');
b = uicontrol('Parent',a, ...
      'Units','points', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'HorizontalAlignment','left', ...
      'Position',[448 260 60.75 9.75], ...
      'String','RTS in V?', ...
      'Style','text', ...
      'Tag','StaticText1');
```

```
hfile = uicontrol('Parent',a, ...
      'Units','points', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'HorizontalAlignment','left', ...
      'Position',[319.5 260 60.75 9.75], ...
      'String',['File: '], ...
      'Style','text', ...
   'Tag','StaticText1');
b = uicontrol('Parent',a, ...
      'Units','points', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'HorizontalAlignment','left', ...
      'Position',[380 170 40 9.75], ...
      'String','Sensitivity', ...
      'Style','text', ...
      'Tag','StaticText1');
hS= uicontrol('Parent',a, ...
      'Units','points', ...
   'BackgroundColor',[1 .8 .8], ...
   'FontWeight','bold',...
      'HorizontalAlignment','left', ...
      'Position',[380 160 40 9.75], ...
      'String',num2str(S*1000,3), ...
      'Style','text', ...
      'Tag','StaticText1');
b = uicontrol('Parent',a, ...
      'Units','points', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'HorizontalAlignment','left', ...
      'Position',[230 375 60.75 9.75], ...
      'String','20 Hz Filter?', ...
      'Style','text', ...
      'Tag','StaticText1');
b = uicontrol('Parent',a, ...
      'Units','points', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'HorizontalAlignment','left', ...
      'Position',[230 350 60.75 9.75], ...
      'String','40 Hz Filter?', ...
      'Style','text', ...
      'Tag','StaticText1');
b = uicontrol('Parent',a, ...
      'Units','points', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'HorizontalAlignment','left', ...
      'Position',[230 200 60.75 9.75], ...
      'String','20 Hz Filter?', ...
      'Style','text', ...
      'Tag','StaticText1');
b = uicontrol('Parent',a, ...
      'Units','points', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'HorizontalAlignment','left', ...
      'Position',[230 175 60.75 9.75], ...
      'String','40 Hz Filter?', ...
      'Style','text', ...
      'Tag','StaticText1');
```

```
% Edit boxes
hRtsc = uicontrol('Parent',a, ...
      'Units','points', ...
      'BackgroundColor',[1 1 1], ...
      'HorizontalAlignment','left', ...
      'Position',[435 230 35.25 15], ...
      'String','1', ...
      'Style','edit', ...
      'Tag','EditText1');
hHfsg = uicontrol('Parent',a, ...
      'Units','points', ...
      'BackgroundColor',[1 1 1], ...
      'HorizontalAlignment','left', ...
      'Position',[387.75 190 35.25 15], ...
      'String','1', ...
      'Style','edit', ...
      'Tag','EditText1');
hHfsc = uicontrol('Parent',a, ...
      'Units','points', ...
      'BackgroundColor',[1 1 1], ...
      'HorizontalAlignment','left', ...
      'Position',[387.75 230 35.25 15], ...
      'String','2', ...
      'Style','edit', ...
      'Tag','EditText1');
hRtsg = uicontrol('Parent',a, ...
      'Units','points', ...
      'BackgroundColor',[1 1 1], ...
      'HorizontalAlignment','left', ...
      'Position',[435 190 35.25 15], ...
      'String','1', ...
      'Style','edit', ...
      'Tag','EditText1');
hRunt = uicontrol('Parent',a, ...
      'Units','points', ...
      'BackgroundColor',[1 1 1], ...
      'HorizontalAlignment','left', ...
      'Position',[435 160 35.25 15], ...
      'String','50', ...
      'Style','edit', ...
      'Tag','EditText1');
hSampf = uicontrol('Parent',a, ...
      'Units','points', ...
      'BackgroundColor',[1 1 1], ...
      'HorizontalAlignment','left', ...
      'Position',[319.5 190 35.25 15], ...
      'String','5000', ...
      'Style','edit', ...
      'Tag','EditText1');
hgn = uicontrol('Parent',a, ...
      'Units','points', ...
      'BackgroundColor',[1 1 1], ...
      'HorizontalAlignment','left', ...
      'Position',[319.5 230 35.25 15], ...
      'String','187', ...
      'Style','edit', ...
      'Tag','EditText1');
hP = uicontrol('Parent',a, ...
      'Units','points', ...
      'BackgroundColor',[1 1 1], ...
      'HorizontalAlignment','left', ...
      'Position',[280 230 35.25 15], ...
      'String',num2str(PREPT), ...
      'Style','edit', ...
      'Tag','EditText1');

%Buttons
zoom off
hZoom = uicontrol('Parent',a, ...
      'Units','points', ...
      'Callback','action=''Zoom'';,frontend2', ...
      'Position',[320 350 55 29.25], ...
      'String','Zoom off', ...
      'Tag','Pushbutton1');
hRunb = uicontrol('Parent',a, ...
      'Units','points', ...
      'Callback','action=''Run'';,frontend2', ...
      'Position',[320 153.75 55 30.75], ...
      'String','Run!', ...
      'Tag','Pushbutton2');
b = uicontrol('Parent',a, ...
      'Units','points', ...
      'Callback','action=''Load'';,frontend2,action=''Update'';,fronten
d2', ...
      'Position',[320 320 55 29.25], ...
      'String','Load Data', ...
      'Tag','Pushbutton1');
b = uicontrol('Parent',a, ...
      'Units','points', ...
      'Callback','action=''Update'';,frontend2', ...
      'Position',[380 320 55 29.25], ...
      'String','Update Input', ...
      'Tag','Pushbutton1');
b = uicontrol('Parent',a, ...
      'Units','points', ...
      'Callback','action=''PickZero'';,frontend2', ...
      'Position',[380 350 55 29.25], ...
      'String','Pick Zero', ...
      'Tag','Pushbutton1');

%Check boxes
htype = uicontrol('Parent',a, ...
      'Units','points', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'Position',[435 260 12 11.25], ...
      'Style','checkbox', ...
      'Tag','Checkbox1', ...
      'Value',0);

f21 = uicontrol('Parent',a, ...
      'Units','points', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'Position',[280 375 12 11.25], ...
      'Style','checkbox', ...
      'Tag','Checkbox1', ...
      'Value',0);
f41 = uicontrol('Parent',a, ...
      'Units','points', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'Position',[280 350 12 11.25], ...
      'Style','checkbox', ...
      'Tag','Checkbox1', ...
      'Value',0);
f22 = uicontrol('Parent',a, ...
      'Units','points', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'Position',[280 200 12 11.25], ...
      'Style','checkbox', ...
      'Tag','Checkbox1', ...
      'Value',0);
f42 = uicontrol('Parent',a, ...
      'Units','points', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'Position',[280 175 12 11.25], ...
      'Style','checkbox', ...
      'Tag','Checkbox1', ...
      'Value',0);
```

Frontend2.m

```
switch action

case 'Run'
   set(hRunb,'String','Running...')

   LONGDATA=str2num(get(hRunt,'String'))/1000*SAMPFREQ+PREPT;
   PREPT=str2num(get(hP,'String'));
   [S, TOUT,QN]=qtodave(T,Q,SAMPFREQ,PREPT,LONGDATA);          %Run the
Analysis!

   %S = Sensitivity from temperatures

   low   = PREPT-15;
   high  = LONGDATA;

   axes(axs(3));
   plot(time(low:high),T(low:high),time(low:high),TOUT(low:high));
   title('TEMP - T exp & T from q (using best S)')

   QE=Q*1000000/S; % [ W/m^2 ]
   QE=QE-mean(QE(1:PREPT));

   axes(axs(4));
   plot(time(low:high),QN(low:high),time(low:high),QE(low:high));
   title('FLUX - q exp & q from t exp  ')

   set(hS,'String',num2str(S*10000,3))
   disp(['Sensivity = ' num2str(S)])
   set(hRunb,'String','Run!')

case 'Zoom'
   if get(hZoom,'String')=='Zoom off'
      set(hZoom,'String','Zoom on ','FontWeight','bold')
      zoom on
   else
      set(hZoom,'String','Zoom off','FontWeight','normal')
      zoom off
   end

case 'Load'
   j=path;
   j=j(1);
   eval(['cd ' j ':\Afosr\Data\ ']);
   [jim, bubb]=uigetfile('*.asc', 'Load Data File');
   [datanames,data]=loadset([bubb jim]);
   set(hfile,'String',jim)

case 'PickZero'
   temp=ginput(1);
   PREPT=round(temp(1)*SAMPFREQ);
   axes(axs(1))
   top=get(gca,'Ylim');
   hold on
   plot([PREPT/SAMPFREQ PREPT/SAMPFREQ],[top(1) top(2)],'r')
   hold off
   axes(axs(2))
   top=get(gca,'Ylim');
   hold on
   plot([PREPT/SAMPFREQ PREPT/SAMPFREQ],[top(1) top(2)],'r')
   hold off
   set(hP,'String',num2str(PREPT))

case 'Update'

   Tcolumn=str2num(get(hRtsc,'String'));
   Qcolumn=str2num(get(hHfsc,'String'));
   Ttype=get(htype,'Value');             % 1 = Temp [RTS V],  0 = Temp
[° C]
   gn = str2num(get(hgn,'String'));
   PREPT=str2num(get(hP,'String'));
   SAMPFREQ=str2num(get(hSampf,'String'));
   RTSGain = str2num(get(hRtsg,'String'));
   HFSGain = str2num(get(hHfsg,'String'));
   To = 18;

   if Ttype==1
      T=data(:,Tcolumn);
      if gn==1
         a       = 0;
         b       = 0;
         c       = 3.93326;    % °C/Ohm
         d       = -495.849;   % °C
         e       = 0.261826;   % Ohm/°C
         f       = 125.7314;   % Ohm
         Ro      = f+To*e;
      end
      if gn==2
         a       = 0;
         b       = 0;
         c       = 2.57488;
         d       = -518.276;
         e       = 0.39751;
         f       = 200.8981;
         Ro      = f+To*e;
      end
      if gn==3
         a       = 0;
         b       = 0;
         c       = 3.9724;
         d       = -480.423;
         e       = 0.255602;
         f       = 120.7617;
         Ro      = f+To*e;
      end
      if gn==4
         a       = 0;
         b       = 0;
         c       = 3.89763;
         d       = -488.813;
         e       = 0.261826;
         f       = 125.1414;
         Ro      = f+To*e;
      end
      if gn==5
         a       = 0;
         b       = 0;
         c       = 2.4478;
         d       = -477.73;
         e       = 0.415353;
         f       = 194.8265;
         Ro      = f+To*e;
      end
      if gn==6
         a       = 0;
         b       = 0;
         c       = 4.13174;
         d       = -476.724;
         e       = 0.243154;
         f       = 115.3224;
         Ro      = f+To*e;
      end
      if gn==7             %This is the gauge in the one-gauge blade
         a       = 0;
         b       = 0;
         c  =  3.7912;
         d  = -578.43;
         e  =  0.263769;
         f  =  152.573;
         Ro = f+To*e;
      end
```

```
      if gn==187                       %This gauge's data is for Vatell
HFM-6
         a  =  0;
         b  =  0;
         c  =  3.13016;
         d  = -450.736;
         e  =  0.333554;
         f  =  143.3088;
         Ro = f+To*e;
      end
      if gn==364                       %This gauge's data is for Vatell
HFM-6
         a  = -1.303e-5;
         b  =  9.439e-3
         c  =  0.79242;
         d  = -340.014;
         e  =  0.338154;
         f  =  169.0287;
         Ro = f+To*e;
      end
      if gn==365                       %This gauge's data is for Vatell
HFM-6
         a  = -1.218e-5;
         b  =  8.465e-3;
         c  =  1.16537;
         d  = -371.229;
         e  =  0.328151;
         f  =  166.2963;
         Ro = f+To*e;
      end
      if gn==488                       %This gauge's data is for Vatell
HFM-6
         a  =  0;
         b  =  0;
         c  =  3.67096;
         d  = -461.079;
         e  =  0.281530;
         f  =  125.1238;
         Ro = f+To*e;
      end
      if gn==489                       %This gauge's data is for Vatell
HFM-6
         a  =  0;
         b  =  0;
         c  =  2.92087;
         d  = -466.645;
         e  =  0.350839;
         f  =  159.3302;
         Ro = f+To*e;
      end
      if gn==588                       %This gauge was given to Oliver
for time response
         a  =  0;                      % It is an HFM-7
         b  =  0;
         c  =  3.90041;
         d  =  -454.83;
         e  =  0.256383;
         f  =  116.6107;
         Ro =  f+To*e;
      end


      R = Ro+T./(RTSGain*100E-6);
      %T = c.*(Ro+T./(RTSGain*100E-6))+d;
      T = a.*R.^3+b.*R.^2+c.*R+d;               %Just for the
Vatell_Data4 stuff!!!!
      else
      T = data(:,Tcolumn);
   end
   Q = data(:,Qcolumn)/HFSGain;   % HFS data in Volts
   time=0: 1/SAMPFREQ : (length(T)-1)/SAMPFREQ;

   %Apply the filter if boxes are checked
   [B1, A1]=butter(4,[18/50   22/50],'stop');  % 20 Hz filter
   [B2, A2]=butter(4,[38.5/50 42/50],'stop');  % 40 Hz filter

   % Filter Temperature
   if get(f21,'Value')==1
      T=filter(B1,A1,T);
   end
   if get(f41,'Value')==1
      T=filter(B2,A2,T);
   end

   % Filter HF
   if get(f22,'Value')==1
      Q=filter(B1,A1,Q);
   end
   if get(f42,'Value')==1
      Q=filter(B2,A2,Q);
   end




   axes(axs(1))
   plot(time,T,'.');
   top=get(gca,'Ylim');
   if exist('PREPT')
      hold on
      plot([PREPT/SAMPFREQ PREPT/SAMPFREQ],[top(1) top(2)],'r')
      hold off
   end
   title('Temperature')

   axes(axs(2))
   plot(time,Q,'.')
   top=get(gca,'Ylim');
   if exist('PREPT')
      hold on
      plot([PREPT/SAMPFREQ PREPT/SAMPFREQ],[top(1) top(2)],'r')
      hold off
   end
   title('Flux')
```

```
end


%  This addition is to plot Q, Texp, and Tcalc for the paper...
%figure(2)
%tmean = mean(T(low:(low+50)));
%subplot(2,1,1), ...
%   plot(time(low:high),Q(low:high));
%   title('HFM Output','fontweight','bold');
%   xlabel('Time (s)');
%   ylabel('HFM Voltage (V)');
%   set(gca,'YLim',[0 3e-4]);
%   set(gca,'XLim',[time(low) time(high)]);
%subplot(2,1,2), ...
%   plot(time(low:high),T(low:high)-
tmean,time(low:high),TOUT(low:high)-tmean);
%   title('T_{calc} and T_{exp}','fontweight','bold');
%   xlabel('Time (s)');
%   ylabel('Temperature - T_o (^{o}C)');
%   set(gca,'YLim',[0 2]);
%   set(gca,'XLim',[time(low) time(high)]);
%set(gcf,'color',[1 1 1]);
```

qtodavej.m

```
function[FS, TOUT,QN]=qtodave(T,Q,SAMPFREQ,PREPT,LONGDATA)
% FS   Sensitivity
% TOUT Temperature calculated from heat flux with determined
sensitivity
% QN   Calculated Heat Flux from Temperature
% T    Measured Surface Temperature
% Q    Raw voltage from HFS on Volts
% SAMPFREQ  Sampling frequency in Hz
% PREPT     Initial number of points used for zeroing HFS and
temperature data
% LONGDATA  Number of points in T and Q. The calculations are done
using the points
%           from PREPT + 1 until LONGDATA

format compact;
k  =  165;           %W/mK    Substrate (Aluminum Nitride as of
12/10/98)
rho= 3290;           %kg/m^3  Substrate
cp =  713;           %J/kgK   Substrate
alpha=k/(rho*cp);
beta=(k*rho*cp)^0.5;
Nmax = LONGDATA;
Nmin = PREPT;
qexp = Q*1000000;    % change from Volts to μV (gain has been taken out
in main code gco)
qexp = qexp-mean(qexp(1:PREPT));   %zeroed HFS data in μV
tavg = mean(T(1:PREPT));
texp = T-tavg;
dt   = 1/SAMPFREQ;
```

```
tmax = dt*(length(texp)-1);
t    = linspace(0,tmax,length(texp));

% The idea behind the code is taken from David Holmbergs (Loren ??) old
Fortran
% code (qt96vt). The way it's being executed is somewhat more
straightforward and deterministic:
% The raw heat flux voltage (in µV) is used to calculate a temperature
history. This temperature
% history is compared to the real measured temperature history by
linearly fitting the two
% sets of data. The resulting slope is immediately the sensitivity of
the gauge.
% In a second step (which is not necessary but insightful) the surface
heat flux is calculated
% from the surface temperature. The raw HFS Voltage is linearily fitted
to the resulting heat flux.
% The inverse of the slope of this fit is another indicator for the
gauge sensitivity.
% If the two do not agree to your full satisfaction, something is
wrong.
%
% Now back to the first step, the calculation of the surface
temperature from heat flux voltage.
% All the equations in the comments are written in sort of a Latex way.
'_' somewhat low. '^' means
% somewhat high.
% e.g. int_{t_{i-1}}^{t_i} means 'integral from t sub i-1 to t sub i'.
You'll figure it out.
% The basic equation is T(0,t)=sqrt(alpha/pi)*1/k*int_0^t
(q(0,tau)/sqrt(t-tau))dtau. It is taken
% from the Greens function solution to the 1-D problem
(q(0,t)=f(t),T(x,0)=To,T(inf,t)=To.
% The heat flux on the surface (q(0,t) is interpolated linearily
between the recorded
% data points:
% For t_{i-1}<tau<t_{i}
% q(0,tau)=q(0,t_{i})-(q(0,t_{i})-q(0,t_{i-1})/dt*(t_{i}-tau)
% The integration with this interpolation then goes like this:
% T(0,t_{n})=sum_{i=2}^n[int_{t_{i-1}}^{t_i}[[q(0,t_i}-(q(0,t_i)-
q(0,t_{i-1}))/dt*(t_i-tau)]/sqrt(t_n-tau)]dtau]
% This integration and summation is performed and leads to the equation
in the heart of the following
% for loop.

Tcalc(1:Nmin+1)= 0*[1:1:Nmin+1];  %Set the values of Tcalc to zero in
the zeroed region
for i=Nmin+2:Nmax
   Tcalc(i)=0;
   for j=Nmin+2:i
      Tcalc(i) = Tcalc(i)...
         +2*qexp(j)*((t(i)-t(j-1))^0.5-(t(i)-t(j))^0.5)...
         +2*(qexp(j)-qexp(j-1))*(t(j)-t(i))*((t(i)-t(j))^0.5-(t(i)-t(j-
1))^0.5)/dt...
         +2/3*(qexp(j)-qexp(j-1))*((t(i)-t(j))^1.5-(t(i)-t(j-
1))^1.5)/dt;
   end
   Tcalc(i)=Tcalc(i)*(alpha/pi)^0.5/k;
end
Tcalc = Tcalc';
[P2,S]= polyfit(texp(Nmin:Nmax),Tcalc(Nmin:Nmax),1);
disp('Sensitivity from temperature fit in µV/(W/m^2):')
FS    = P2(1)
Tcalc  = Tcalc/P2(1)+tavg;
TOUT   = Tcalc;

% In the second step the surface heat flux is calculated from the
measured temperatures.
% The raw voltages from the HFS are compared to this resulting heat
flux and linearily fitted.
% Actually the calculated result is fitted to the voltage data because
there is a lot
% more noise in the calculated result than in the measured data. The
inverse
% of the slope is a second possible value for the gauge sensitivity.
% The equation to calculate heat flux is taken from  W.J. Cook,
'Determination of
% Heat Transfer Rates from Transient Surface Temperature Measurements',
AIAA Journal Vol 8
% pp1366-1368. The paper is referring to W.J. Cook and E.J. Felderman,
'Reduction of Data from
% Thin-Film Heat-Transfer Gages: A Concise Numerical Technique', AIAA
Journal Vol 4 pp561-562 1966.
% This paper is referring to Vidal, R.J., Wittcliff, C.E., Bartlett,
C.E., Logan, J.G., 'Investigation
% of Stagnation Point Heat Transfer in the C.A.L. hypersonic sgock
tunnel', Cornell Aeronautical Lab
% Rept. AA-966-A-1 (November 1955) and Jones, J.J., 'Shock tube heat
transfer measurements on a inner
% surface of a cylinder (simulating a flat plate) for stagnation
temperature range 41000° to 8300°R',
% NASA TN D-54 (1959). But do not worry, the last two papers can
nowhere be found, so you better not
% waste your time.

qn(1:Nmin+1)= 0*[1:1:Nmin+1];
for i=Nmin+2:Nmax
   qn(i)=0;
   for j=Nmin+2:i
      qn(i)=qn(i)...
         +((texp(j)-texp(j-1))/((t(i)-t(j))^0.5+(t(i)-t(j-1))^0.5));
   end
   qn(i)=2*beta*qn(i)/pi^0.5;
end
qn   = qn';
[P1,S]= polyfit(qexp(Nmin:Nmax),qn(Nmin:Nmax),1);
disp('Sensitivity from heat flux fit ( in µV/(W/m^2));')
FSH  = 1/P1(1)
qexp = qexp/FSH;
QN   = qn;
```

Appendix B-2: Data Processing

datasum2.m

```
clear all;

%Blade Geometry
suction  = 9.760;                          % Suction side length
(inches)
pressure = 6.001;                          % Pressure side length
(inches)
archord   = 5.314;                         % Arial chord (inches)
(stagnation pt to stagnation pt)
axchord    = 4.643;                        % Axial chord (inches)
saxchord  = axchord / 3;                   % Small blade chord
(inches)
ssuction  = suction / 3;                   % Small blade suction side
length (inches)
gl = [ 1.515 1.916 2.328 2.730 3.137 3.548]; % Gauge Locations (inches)
(front stag pt)
gld = (gl-1.13)/0.03937;                   % Gauge location in
multiples of hole diameters (1mm)from sg
cl = [ -0.765 -0.231 0 0.305 0.422 1.130];   % Coolant Exit locations
(inches) (from stag pt)
ngl = gl./suction;                         % Non-dimensional gauge
location (by SS length)
ncl = cl./suction;                         % Non-dimension coolant
exit (by SS length)
Re = 5E6;                                  % Re based on exit


%h -> Some uncooled runs taken 11/23 100 Hz
g1_11_23_nc=[670.88 683.58 685.47 701.43 683.40 681.91];
g2_11_23_nc=[659.08 666.60 675.48 689.71 672.97 669.42];
g3_11_23_nc=[579.49 588.30 585.87 601.88 589.84 584.56];
g4_11_23_nc=[712.42 725.12 714.25 742.52 729.04 722.03];
g5_11_23_nc=[675.11 701.03 687.49 713.72 702.60 696.88];
g6_11_23_nc=[698.15 739.19 731.43 753.34 744.95 739.53];

%h -> Some ambient coolant, ambient flow tests taken 9/22 100Hz (for
plot 3)
g1_9_22_ac=[748.09 735.81 901.11 NaN];
g2_9_22_ac=[611.95 576.58 778.07 NaN];
g3_9_22_ac=[453.25 447.33 NaN    642.39];
g4_9_22_ac=[677.57 641.81 NaN    770.85];
g5_9_22_ac=[NaN    NaN    717.61 706.45];
g6_9_22_ac=[NaN    NaN    708.05 695.23];

%h -> Some ambient coolant, ambient flow tests taken 9/22 100Hz (for
plot 1); repeat of above data...
g1_ac=[748.09 735.81 901.11];
g2_ac=[611.95 576.58 778.07];
g3_ac=[453.25 447.33 642.39];
```

```
g4_ac=[677.57 641.81 770.85];
g5_ac=[717.61 706.45];
g6_ac=[708.05 695.23];


%h -> Some cooled runs taken 11/23 100 Hz
g1_11_23_cc=[929.07 855.27 876.59 845.55 824.95 849.85];
g2_11_23_cc=[812.49 792.55 783.21 779.50 773.31 769.07];
g3_11_23_cc=[618.71 602.78 655.89 625.57 591.44 638.93];
g4_11_23_cc=[800.12 786.60 764.07 780.86 777.95 791.01];
g5_11_23_cc=[759.09 732.53 752.73 744.08 740.82 750.03];
g6_11_23_cc=[904.11 874.74 856.80 839.00 884.32 890.46];

%Tdrop -> Some uncooled runs taken 11/23 100 Hz using STC's
g1_11_23_tdrop=[5.036 5.746 6.248 7.222 5.312 6.554];
g2_11_23_tdrop=[4.353 5.021 5.384 6.238 4.663 5.742];
g3_11_23_tdrop=[6.080 6.292 6.668 7.025 6.010 6.621];
g4_11_23_tdrop=[8.276 8.254 9.058 9.132 8.753 8.993];
g5_11_23_tdrop=[9.902 9.672 10.60 10.67 10.24 10.45];
g6_11_23_tdrop=[16.40 16.24 16.69 16.96 16.01 16.64];

%Tdrop -> Some uncooled runs taken 11/23 100 Hz using RTS's
g1_11_23_tdrop_rts=[6.295 7.056 7.474 8.486 6.507 7.850];
g2_11_23_tdrop_rts=[7.039 7.664 8.075 8.844 7.356 8.354];
g3_11_23_tdrop_rts=[8.226 8.481 8.830 9.211 8.146 8.791];
g4_11_23_tdrop_rts=[8.276 10.75 11.54 11.55 11.19 11.41];
g5_11_23_tdrop_rts=[10.79 12.32 13.19 13.03 12.60 12.93];
g6_11_23_tdrop_rts=[16.58 16.14 17.24 17.12 16.23 16.66];

%eta -> Some cooled runs taken 11/23 100 Hz using STC's
g1_11_23_eta=[0.3144 0.2993 0.3316 0.3134 0.2941 0.3329];
g2_11_23_eta=[0.1849 0.1868 0.1868 0.1928 0.1673 0.1849];
g3_11_23_eta=[0.3532 0.3317 0.3527 0.3452 0.3291 0.3498];
g4_11_23_eta=[0.2293 0.2260 0.2223 0.2247 0.2136 0.2290];
g5_11_23_eta=[0.1900 0.1760 0.2063 0.1960 0.1818 0.2008];
g6_11_23_eta=[0.2733 0.2821 0.2721 0.2765 0.2966 0.2798];

%eta -> Some cooled runs taken 11/23 100 Hz using RTS's
g1_11_23_eta_rts=[0.3128 0.2979 0.3391 0.3226 0.2864 0.3320];
g2_11_23_eta_rts=[0.1828 0.1824 0.1958 0.2039 0.1569 0.1826];
g3_11_23_eta_rts=[0.3520 0.3301 0.3622 0.3564 0.3225 0.3503];
g4_11_23_eta_rts=[0.2320 0.2293 0.2316 0.2369 0.2157 0.2322];
g5_11_23_eta_rts=[0.2005 0.1932 0.2378 0.2260 0.2001 0.2227];
g6_11_23_eta_rts=[0.3028 0.2938 0.2972 0.2994 0.3014 0.2956];

%eta -> Some cooled runs taken 11/23 100 Hz using Pr^(1/3)
g1_11_23_eta_pr13=[0.3280 0.3161 0.3439 0.3306 0.3180 0.3459];
g2_11_23_eta_pr13=[0.2003 0.2059 0.2020 0.2135 0.1962 0.2012];
g3_11_23_eta_pr13=[0.3644 0.3454 0.3631 0.3589 0.3481 0.3608];
g4_11_23_eta_pr13=[0.2562 0.2589 0.2489 0.2602 0.2613 0.2567];
g5_11_23_eta_pr13=[0.2200 0.2140 0.2347 0.2355 0.2352 0.2314];
g6_11_23_eta_pr13=[0.3176 0.3333 0.3141 0.3312 0.3635 0.3232];

%Average Tfree for uncooled runs
tavg_nc = [ 57.555 53.324 58.679 51.427 52.899 55.150 ];

%Mach No. -> Some uncooled runs taken 11/23 100 Hz
```

```
g1_11_23_mach=[0.5452 0.5593 0.6200 0.6091 0.5690 0.6279];
g2_11_23_mach=[0.5427 0.5425 0.5437 0.5415 0.5394 0.5453];
%g3_11_23_mach=[0.3532 0.3317 0.3527 0.3452 0.3291 0.3498];
g4_11_23_mach=[0.6692 0.6690 0.6667 0.6656 0.6646 0.6689];
g5_11_23_mach=[0.7605 0.7605 0.7616 0.7584 0.7591 0.7605];
g6_11_23_mach=[0.8980 0.9046 0.9038 0.9010 0.9029 0.9027];

close all;
format short g;
names   ='e:\afosr\dwight\machno.txt';
%names    ='c:\Users\Dwight\Thesis\machno.txt';
[tempnames,data]=loadset(names);
Xcm       = 3*data(:,1)/2.54/9.76;
MachNo    = data(:,2);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Graphical Output                                        %
%                                                        %
%Figure 1: StdDev plot of all six gages                  %
%Figure 2: Summary of all runs, all locations            %
%Figure 3: All uncooled runs, all locations              %
%Figure 4: All ambient runs, all locations               %
%Figure 5: All cooled runs, all locations                %
%Figure 6: Cooled vs. Uncooled runs, all locations       %
%Figure 7: Cooled vs. Ambient runs, all locations        %
%Figure 8: Film Effectiveness, all locations             %
%Figure 9: Mach No. Profile, all locations               %
%Figure 10: Tdrop Profile, all locations                 %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

x     = ngl';

%Figure 1:  Heat Transfer Coefficient Summary of All Runs
figure(1)
gubbo = [ngl(1) ngl(1) ngl(1);...
         ngl(2) ngl(2) ngl(2);...
         ngl(3) ngl(3) ngl(3);...
         ngl(4) ngl(4) ngl(4);...
         ngl(5) ngl(5) ngl(5);...
         ngl(6) ngl(6) ngl(6)];
dubbo =...
   [ mean(g1_11_23_nc) mean(g1_11_23_cc) mean(g1_ac);...
     mean(g2_11_23_nc) mean(g2_11_23_cc) mean(g2_ac);...
     mean(g3_11_23_nc) mean(g3_11_23_cc) mean(g3_ac);...
     mean(g4_11_23_nc) mean(g4_11_23_cc) mean(g4_ac);...
     mean(g5_11_23_nc) mean(g5_11_23_cc) mean(g5_ac);...
     mean(g6_11_23_nc) mean(g6_11_23_cc) mean(g6_ac)];
ebbo = ...
   [ std(g1_11_23_nc) std(g1_11_23_cc) std(g1_ac);...
     std(g2_11_23_nc) std(g2_11_23_cc) std(g2_ac);...
     std(g3_11_23_nc) std(g3_11_23_cc) std(g3_ac);...
     std(g4_11_23_nc) std(g4_11_23_cc) std(g4_ac);...
     std(g5_11_23_nc) std(g5_11_23_cc) std(g5_ac);...
     std(g6_11_23_nc) std(g6_11_23_cc) std(g6_ac)];

errorbar( gubbo, dubbo, ebbo,'o');
legend('Uncooled','Cooled', 'Ambient');
```

```
title(['Heat Transfer Coefficient Profile'],'FontWeight','bold');
xlabel('Non-dimensional distance from leading edge [s/Lss]');
ylabel('Mean h, W/m^2-°C');
axis([0 0.6 0 1000]);
ytot = get(gca,'YLim');
ynom    = ytot(2)/10;
axis(axis);
arrow([ncl(4) 0 0], [ncl(4) ynom 0]);
arrow([ncl(5) 0 0], [ncl(5) ynom 0]);
arrow([ncl(6) 0 0], [ncl(6) ynom 0]);
set(gcf,'color',[1 1 1]);
hold on;
gauge([]);
hold off;

figure(2)
Xone    = {ngl ngl};
Yone    = {dubbo(:,1) dubbo(:,2)};
s1      = {1 1};
m1      = {'o-' 'd-'};
Xtwo    = {ngl};
Ytwo    = {dubbo(:,3)};
s2      = {1};
m2      = {'s-'};
[AX,H1,H2,H3,H4]=fancyplot2(Xone,Yone,s1,m1,Xtwo,Ytwo,s2,m2);
title(['Heat Transfer Coefficient Profile'],'FontWeight','bold');
xlabel('Non-dimensional distance from leading edge [s/Lss]');
axes(AX(1));
ylabel('h, W/m^2-°C');
axis([0 0.6 0 1000]);
ytot = get(gca,'YLim');
ynom    = ytot(2)/10;
axis(axis);
arrow([ncl(4) 0 0], [ncl(4) ynom 0]);
arrow([ncl(5) 0 0], [ncl(5) ynom 0]);
arrow([ncl(6) 0 0], [ncl(6) ynom 0]);
axes(AX(2));
axis([0 0.6 0 1000]);
box on;
set(H1(1), 'color', [0 0.5 0]);
set(H3(1), 'color', [0 0.5 0]);
set(H1(2), 'color', 'b');
set(H3(2), 'color', 'b');
set(H2(1), 'color', 'r');
set(H4(1), 'color', 'r');
legend([H3 H4],'Uncooled','Cooled','Ambient');
set(H3,'linestyle','none');
set(H4,'linestyle','none');
set(gcf,'color',[1 1 1]);
set(gca,'YTickLabel',' ');
hold on;
G2 = gauge([]);
hold off;


%Figure 3:  Heat Transfer Coefficient Data, Uncooled
figure(3)
```

```
grubber = [x x x x x x];
flubber = ...
   [ g1_11_23_nc; g2_11_23_nc; g3_11_23_nc;...
      g4_11_23_nc; g5_11_23_nc; g6_11_23_nc];
dubber=['''ro-''';'''g*-''';'''b+-''';'''ms-''';'''cd-''';'''kx-'''];
for k = 1:6,
   plot(grubber(:,k),flubber(:,k),eval(dubber(k,:)));
   hold on;
end
hold off;
title(['Heat Transfer Coefficient Profile,
Uncooled'],'FontWeight','bold');
xlabel('Non-dimensional distance from leading edge [s/Lss]');
ylabel('h, W/m^2-°C');
axis([0 0.5 0 1000]);
ytot = get(gca,'YLim');
ynom    = ytot(2)/10;
axis(axis);
arrow([ncl(4) 0 0], [ncl(4) ynom 0]);
arrow([ncl(5) 0 0], [ncl(5) ynom 0]);
arrow([ncl(6) 0 0], [ncl(6) ynom 0]);
set(gcf,'color',[1 1 1]);
hold on;
G3 = gauge([]);
hold off;


%Figure 4:  Heat Transfer Coefficient Data, Ambient
figure(4)

abc  = [x x x x x x];
ezas = ...
   [ g1_9_22_ac; g2_9_22_ac; g3_9_22_ac;...
      g4_9_22_ac; g5_9_22_ac; g6_9_22_ac];
onetwothree  =['''ro-''';'''g*-''';'''b+-''';'''ms-''';'''cd-''';'''kx-
'''];
for k = 1:4,
   plot(abc(:,k),ezas(:,k),eval(onetwothree(k,:)));
   hold on;
end
hold off;
title(['Heat Transfer Coefficient Profile, Ambient
'],'FontWeight','bold');
xlabel('Non-dimensional distance from leading edge [s/Lss]');
ylabel('h, W/m^2-°C');
axis([0 0.5 0 1000]);
ytot = get(gca,'YLim');
ynom    = ytot(2)/10;
axis(axis);
arrow([ncl(4) 0 0], [ncl(4) ynom 0]);
arrow([ncl(5) 0 0], [ncl(5) ynom 0]);
arrow([ncl(6) 0 0], [ncl(6) ynom 0]);
set(gcf,'color',[1 1 1]);
hold on;
G4 = gauge([]);
hold off;
```

```
%Figure 5:  Heat Transfer Coefficient Data, Cooled
figure(5)

mecha = [x x x x x x];
lecha = ...
   [ g1_11_23_cc; g2_11_23_cc; g3_11_23_cc;...
      g4_11_23_cc; g5_11_23_cc; g6_11_23_cc];
high  =['''ro-''';'''g*-''';'''b+-''';'''ms-''';'''cd-''';'''kx-'''];
for k = 1:6,
   plot(mecha(:,k),lecha(:,k),eval(high(k,:)));
   hold on;
end
hold off;
title(['Heat Transfer Coefficient Profile,
Cooled'],'FontWeight','bold');
xlabel('Non-dimensional distance from leading edge [s/Lss]');
ylabel('h, W/m^2-°C');
axis([0 0.5 0 1000]);
ytot = get(gca,'YLim');
ynom    = ytot(2)/10;
axis(axis);
arrow([ncl(4) 0 0], [ncl(4) ynom 0]);
arrow([ncl(5) 0 0], [ncl(5) ynom 0]);
arrow([ncl(6) 0 0], [ncl(6) ynom 0]);
set(gcf,'color',[1 1 1]);
hold on;
G5 = gauge([]);
hold off;



%Figure 6:  Average Cooled vs. Average Uncooled
figure(6)

digger  = dubbo(:,1);
trigger = dubbo(:,2);
bigger  = 100*(trigger - digger)./digger;
Xone    = {ngl};
Yone    = {digger};
s1      = {1};
m1      = {'o-'};
Xtwo    = {ngl};
Ytwo    = {trigger};
s2      = {1};
m2      = {'s-'};
[AX,H1,H2,H3,H4]=fancyplot2(Xone,Yone,s1,m1,Xtwo,Ytwo,s2,m2);
%title(['Heat Transfer Coefficient Profile, Cooled vs.
Uncooled'],'FontWeight','bold');
xlabel('Non-dimensional distance from leading edge [s/Lss]');
axes(AX(1));
ylabel('h, W/m^2-°C');
axis([0 0.6 0 1000]);
ytot = get(gca,'YLim');
ynom    = ytot(2)/10;
axis(axis);
arrow([ncl(4) 0 0], [ncl(4) ynom 0]);
arrow([ncl(5) 0 0], [ncl(5) ynom 0]);
arrow([ncl(6) 0 0], [ncl(6) ynom 0]);
axes(AX(2));
%ylabel('h, W/m^2-°C');
axis([0 0.6 0 1000]);
set(H1(1), 'color', [0 0.5 0]);
set(H3(1), 'color', [0 0.5 0]);
set(H2(1), 'color', 'b');
set(H4(1), 'color', 'b');
set(gca,'YTickLabel',' ');
%set(H2(1), 'color', 'r');
%set(H4(1), 'color', 'r');
legend([H3, H4], 'Uncooled', 'Cooled');
set(gcf,'color',[1 1 1]);
box on;
hold on;
G6 = gauge([]);
hold off;


%Figure 7:  Average Cooled vs. Average Ambient
figure(7)

dig  = dubbo(:,3);
trig = dubbo(:,2);
%big  = 100*(trig - dig)./dig;
Xone    = {ngl};
Yone    = {dig};
s1      = {1};
m1      = {'o-'};
Xtwo    = {ngl};
Ytwo    = {trig};
s2      = {1};
m2      = {'s-'};
[AX,H1,H2,H3,H4]=fancyplot2(Xone,Yone,s1,m1,Xtwo,Ytwo,s2,m2);
title(['Heat Transfer Coefficient Profile, Cooled vs.
Ambient'],'FontWeight','bold');
xlabel('Non-dimensional distance from leading edge [s/Lss]');
axes(AX(1));
ylabel('h, W/m^2-°C');
axis([0 0.6 0 1000]);
ytot = get(gca,'YLim');
ynom    = ytot(2)/10;
axis(axis);
arrow([ncl(4) 0 0], [ncl(4) ynom 0]);
arrow([ncl(5) 0 0], [ncl(5) ynom 0]);
arrow([ncl(6) 0 0], [ncl(6) ynom 0]);
axes(AX(2));
%ylabel('Percent Difference, %');
axis([0 0.6 0 1000]);
set(H1(1), 'color', [0 0.5 0]);
set(H3(1), 'color', [0 0.5 0]);
set(H2(1), 'color', 'b');
set(H4(1), 'color', 'b');
set(gca,'YTickLabel',' ');
%set(H2(1), 'color', 'r');
%set(H4(1), 'color', 'r');
```

```
legend([H3, H4], 'Ambient', 'Cooled');
set(gcf,'color',[1 1 1]);
box on;
hold on;
G7 = gauge([]);
hold off;


%Figure 8:  Film Cooling Effectiveness Summary of All Runs
figure(8)
gubb = [ngl(1); ngl(2); ngl(3); ngl(4); ngl(5); ngl(6)];
dubb = [mean(g1_11_23_eta); mean(g2_11_23_eta); mean(g3_11_23_eta);...
      mean(g4_11_23_eta); mean(g5_11_23_eta); mean(g6_11_23_eta)];
rubb = [mean(g1_11_23_eta_rts); mean(g2_11_23_eta_rts);
mean(g3_11_23_eta_rts);...
        mean(g4_11_23_eta_rts); mean(g5_11_23_eta_rts);
mean(g6_11_23_eta_rts)];
ebb  = [mean(g1_11_23_eta_pr13); mean(g2_11_23_eta_pr13);...
        mean(g3_11_23_eta_pr13); mean(g4_11_23_eta_pr13);...
        mean(g5_11_23_eta_pr13); mean(g6_11_23_eta_pr13)];
Xone    = {ngl};
Yone    = {rubb};
s1      = {1};
m1      = {'o-'};
Xtwo    = {ngl};
Ytwo    = {ebb};
s2      = {1};
m2      = {'d-'};
[AX,H1,H2,H3,H4]=fancyplot2(Xone,Yone,s1,m1,Xtwo,Ytwo,s2,m2);
title(['Film Effectiveness Profile'],'FontWeight','bold');
xlabel('Non-dimensional distance from leading edge [s/Lss]');
axes(AX(1));
ylabel('Experimental Eta');
axis([0 0.6 0 0.5]);
axes(AX(2));
%ylabel('Predicted Eta');
axis([0 0.6 0 0.5]);
ytot = get(gca,'YLim');
ynom    = ytot(2)/10;
axis(axis);
arrow([ncl(4) 0 0], [ncl(4) ynom  0]);
arrow([ncl(5) 0 0], [ncl(5) ynom  0]);
arrow([ncl(6) 0 0], [ncl(6) ynom  0]);
set(gcf,'color',[1 1 1]);
set(H1(1),'color','b')
set(H3(1),'color','b')
%set(H1(2),'color',[0 0.5 0])
%set(H3(2),'color',[0 0.5 0])
set(H2(1),'color','r')
set(H4(1),'color','r')
set(gca,'YTickLabel',' ');
legend([H3,H4],'Experimental','Predicted, (PR^{1/3})');
hold on;
G8 = gauge([]);
hold off;

%Figure 9:  Mach No. Summary of Uncooled Runs
```

```
figure(9)
tavg     = mean(tavg_nc+273.15);
gamma    = 1.4;
R        = 287;
Cp       = 1006.5;
happy    = [ ngl(1); ngl(2); ngl(3); ngl(4); ngl(5); ngl(6) ];
sleepy   = [ mean(g1_11_23_mach); mean(g2_11_23_mach); NaN;
mean(g4_11_23_mach);...
             mean(g5_11_23_mach); mean(g6_11_23_mach) ];
Ufree    = sqrt(gamma*R*tavg).*sleepy;
for j = 1:length(Ufree),
   Ufreesqd(j) = Ufree(j)^2;
end
subbo = [ mean(g1_11_23_tdrop); mean(g2_11_23_tdrop);
mean(g3_11_23_tdrop);...
          mean(g4_11_23_tdrop); mean(g5_11_23_tdrop);
mean(g6_11_23_tdrop) ];
rc       = -1*(1 - ((2*Cp.*subbo)./Ufreesqd'));
Xone    = {[ncl(4) ncl(5) ncl(6) ngl]};
Yone    = {[0.38 0.45 0.53 sleepy']};
s1      = {1};
m1      = {'d'};
Xtwo    = {Xcm};
Ytwo    = {MachNo};
s2      = {1};
m2      = {'-'};
[AX,H1,H2,H3,H4]=fancyplot2(Xone,Yone,s1,m1,Xtwo,Ytwo,s2,m2);
title(['Mach No. Profile'],'FontWeight','bold');
xlabel('Non-dimensional distance from leading edge [s/Lss]');
axes(AX(1));
ylabel('Mach No.');
axis([0 0.6 0 2]);
axes(AX(2));
axis([0 0.6 0 2]);
ytot = get(gca,'YLim');
ynom     = ytot(2)/10;
axis(axis);
arrow([ncl(4) 0 0], [ncl(4) ynom  0]);
arrow([ncl(5) 0 0], [ncl(5) ynom  0]);
arrow([ncl(6) 0 0], [ncl(6) ynom  0]);
set(gcf,'color',[1 1 1]);
set(H1(1), 'color', 'b','linestyle','none');
set(H3(1), 'color', 'b','linestyle','none');
set(H2(1), 'color', [0 0.5 0]);
set(H4(1), 'color', [0 0.5 0]);
set(gca,'YTickLabel',' ');
legend([H3,H4],'Experimental','Predicted (GE)');
box on;
hold on;
G9 = gauge([]);
hold off;


%Figure 10:  Tdrop Summary of All Runs
figure(10)
Pr13      = 0.7075^(1/3);
Pr12      = 0.7075^(1/2);
```

```
Tdropcalc13 = ((1-Pr13)/(2*Cp)).*Ufreesqd;
Tdropcalc12 = ((1-Pr12)/(2*Cp)).*Ufreesqd;
for j = 1:6,
   Td1(j) = ((1-rc(j))/(2*Cp)).*Ufreesqd(j);
end
%rubbo = [ ngl(1); ngl(2); ngl(3); ngl(4); ngl(5); ngl(6) ];
cubbo = [ mean(g1_11_23_tdrop); mean(g2_11_23_tdrop);
mean(g3_11_23_tdrop);...
      mean(g4_11_23_tdrop); mean(g5_11_23_tdrop); mean(g6_11_23_tdrop)
];
rubbo = [ mean(g1_11_23_tdrop_rts); mean(g2_11_23_tdrop_rts);
mean(g3_11_23_tdrop_rts);...
          mean(g4_11_23_tdrop_rts); mean(g5_11_23_tdrop_rts);
mean(g6_11_23_tdrop_rts) ];
%abbo  = [ std(g1_11_23_tdrop); std(g2_11_23_tdrop);
std(g3_11_23_tdrop);...
%          std(g4_11_23_tdrop); std(g5_11_23_tdrop);
std(g6_11_23_tdrop) ];
%errorbar( rubbo, cubbo, abbo,'o');
Xone    = {ngl};
Yone    = {cubbo};
s1      = {1};
m1      = {'o-'};
Xtwo    = {ngl};
Ytwo    = {Tdropcalc13};
s2      = {1};
m2      = {'d-'};
[AX,H1,H2,H3,H4]=fancyplot2(Xone,Yone,s1,m1,Xtwo,Ytwo,s2,m2);
title(['Tdrop Profile, Actual vs. Calculated'],'FontWeight','bold');
xlabel('Non-dimensional distance from leading edge [s/Lss]');
axes(AX(1));
ylabel('Temperature, °C');
%legend('Tdrop');
%title(['Streamwise Tdrop Profile'],'FontWeight','bold');
%xlabel('Non-dimensional distance from leading edge [s/Lss]');
%ylabel('Tdrop, °C');
axis([0 0.6 0 20]);
ytot = get(gca,'YLim');
ynom    = ytot(2)/10;
axis(axis);
arrow([ncl(4) 0 0], [ncl(4) ynom 0]);
arrow([ncl(5) 0 0], [ncl(5) ynom 0]);
arrow([ncl(6) 0 0], [ncl(6) ynom 0]);
axes(AX(2));
axis([0 0.6 0 20]);
set(H1(1),'color','b')
set(H3(1),'color','b')
%set(H1(2),'color',[0 0.5 0])
%set(H3(2),'color',[0 0.5 0])
set(H2(1),'color','r')
set(H4(1),'color','r')
%set(H2(2),'color',[0.75 0 0.75])
%set(H4(2),'color',[0.75 0 0.75])
legend([H3, H4], 'Experimental','Calculated, (Pr^{1/3})',2);
set(gcf,'color',[1 1 1]);
set(gca,'YTickLabel',' ');
box on;
```

```
hold on;
G10 = gauge([]);
hold off;
```

data1123b.m

```
% data112398.m
% Data Ananylsis for Turbine Heat Transfer Group
% 6 triplet blade analysis
% O popp
%
%

close all;
format short g;
clear all;

% ---- Input Data Files ------------
% This is the section you change...
%
% typeflag = 0   No Coolant Flow
%                h = q / (Tfree - Tblade)
%                   Tdrop = (1-r)*u^2/2Cp
%
% typeflag = 1   Cold Coolant Flow (using measured Tdrop)
%                h = ( q / (Tr - Tcool) ) / ( (Tr - Tblade)/(Tr -
Tcool) )
%                n = (Taw - Tr) / (Tcool - Tr)
%                   Tr = Tfree - Tdrop (Tdrop=6.9°C)
%
% typeflag = 2   Cold Coolant Flow (using Pr approx for Tr)
%                h = ( q / (Tr - Tcool) ) / ( (Tr - Tblade)/(Tr -
Tcool) )
%                n = (Taw - Tr) / (Tcool - Tr)
%                   Tr = Tfree - Pr^.33 (Tdrop=0.89°C)
% front,back     Analysis window for each data set
%
% data           data(time,column#, run#)
%

names    ='e:\Afosr\Data\11_23\Cooling\run11_23_10.asc';
typeflag = 1;
front    = 600;
back     = 2500;
qoffset  = 0;
To       = 20;
Po       = 13.7;
times    = 2040;   %Time of shock impact *100
checketa = 2;

% ---- OK now stop changing things from here on out  -------
% ---- The rest is golden.... ----------------------------

disp('Loading data files...')
[datanames,data]=loadset(names);
disp('Performing Analysis...')
```

```
NF=size(data,3);                                   %Number of Data
Files


   Torr         = data(:,6);
   Tfree        = data(:,7);
   Tplen        = data(:,8);
   Tsg       = data(:,14);
   Tsn2      = data(:,15);
   Tsn1      = data(:,16);
   Tn           = data(:,21);
   Tpn          = data(:,22);
   Tpg          = data(:,23);
   Tblade6   = data(:,24);
   Tblade5   = data(:,29);
   Tblade4   = data(:,30);         %using Tblade5
   Tblade3   = data(:,30);
   Tblade2   = data(:,31);
   Tblade1   = data(:,32);
   Psg       = data(:,34);       %offset fudged
   Psn2      = data(:,35);          %offset fudged
   Psn1      = data(:,36);        %offset fudged
   Pn        = data(:,37);
   Ppn       = data(:,38);
   Ppg       = data(:,39);
   Pdp       = data(:,40);       %offset fudged
   Pup       = data(:,41);
   Pfree     = data(:,43);
   RTS5         = data(:,10);
   RTS6         = data(:,11);
   RTS4      = data(:,45);
   RTS3      = data(:,46);
   RTS2      = data(:,47);
   RTS1      = data(:,48);
   Pplen     = data(:,52);   % P or dP ???
   Kul6      = data(:,53);
   Kul5      = data(:,54);
   Kul4      = data(:,55);
   Kul3      = data(:,56);
   Kul2      = data(:,57);
   Kul1      = data(:,58);
   HFS6         = data(:,12);
   %  HFS6      = HFS6 - mean(HFS6(1:100));   % Zeroed the data
   %  [b,a]=ellip(4,.5,20,.5);                % Create Elliptical
Filter
   %  HFS6=filter(b,a,HFS6);                  % Apply filter

   HFS5      = data(:,59);
   HFS4      = data(:,60);
   HFS3      = data(:,61);
   HFS2      = data(:,62);
   HFS1      = data(:,63);



   len       = length(Tn);
   t         = linspace(0,len/100,len)';
```

```
%Tfree-Trec for all locations from uncooled runs
if checketa == 0
   Tdrop1    = 5.95;
   Tdrop2    = 5.15;
   Tdrop3    = 5.78;
   Tdrop4    = 9.03;
   Tdrop5    = 10.18;
   Tdrop6    = 16.33;
elseif checketa == 1
   Tdrop1    = 2.4693;
   Tdrop2    = 2.0991;
   Tdrop3    = 2.638;    %interpolated between positions #2 and #4
   Tdrop4    = 3.176;
   Tdrop5    = 4.1204;
   Tdrop6    = 5.8046;
else checketa == 2
   Tdrop1    = 7.278;
   Tdrop2    = 7.8887;
   Tdrop3    = 8.6142;
   Tdrop4    = 10.786;
   Tdrop5    = 12.477;
   Tdrop6    = 16.662;
end

%HFS #1 CALIBRATION
HFSGain1 = 100;
   g1       =  0;         % µV/((W/cm^2)°C)
hcal1    = 234;        % µV/(W/cm^2)
RTSGain1 = 500;
c1       = 3.93326;    % °C/Ohm
d1       = -495.849;   % °C
e1       = 0.261826;   % Ohm/°C
f1       = 125.7314;   % Ohm
Ro1      = f1+To*e1;

%HFS #2 CALIBRATION
HFSGain2 = 100;
   g2       =  0;
hcal2    = 105;
RTSGain2 = 500;
c2       = 2.57488;
d2       = -518.276;
e2       = 0.39751;
f2       = 200.8981;
Ro2      = f2+To*e2;


%HFS #3 CALIBRATION
HFSGain3 = 100;
   g3       =  0;
hcal3    = 145;
RTSGain3 = 500;
c3       = 3.9724;
d3       = -480.423;
e3       = 0.255602;
f3       = 120.7617;
```

```
Ro3      = f3+To*e3;


%HFS #4 CALIBRATION
HFSGain4 = 100;
   g4       =  0;
hcal4    = 130;
RTSGain4 = 500;
c4       = 3.89763;
d4       = -488.813;
e4       = 0.261826;
f4       = 125.1414;
Ro4      = f4+To*e4;


%HFS #5 CALIBRATION
HFSGain5 = 100;
g5       =  0;
hcal5    = 171;
RTSGain5 = 1;
c5       = 2.4478;
d5       = -477.73;
e5       = 0.415353;
f5       = 194.8265;
Ro5      = f5+To*e5;


%HFS #6 CALIBRATION
HFSGain6 = 1;
   g6       =  0;
hcal6    = 153;
RTSGain6 = 1;
c6       = 4.13174;
d6       = -476.724;
e6       = 0.243154;
f6       = 115.3224;
Ro6      = f6+To*e6;

%Kulite Sensitivities
KS1      = 1.688;    % mV/psi
KS2      = 1.696;
KS3      = 1.727;
KS4      = 1.743;
KS5      = 1.724;
KS6      = 2.000;

%Kulite Gains
KG1      = 10;
KG2      = 10;
KG3      = 10;
KG4      = 10;
KG5      = 10;
KG6      = 10;



%Flux at all locations
```

```
q1   = HFS1.*1000000./(HFSGain1.*(g1.*Tblade1+hcal1));
q2   = HFS2.*1000000./(HFSGain2.*(g2.*Tblade2+hcal2));
q3   = HFS3.*1000000./(HFSGain3.*(g3.*Tblade3+hcal3));
q4   = HFS4.*1000000./(HFSGain4.*(g4.*Tblade4+hcal4));
q5   = HFS5.*1000000./(HFSGain5.*(g5.*Tblade5+hcal5));
q6   = HFS6.*1000000./(HFSGain6.*(g6.*Tblade6+hcal6));

%RTS Temperature at all locations
RTS1 = c1.*(Ro1+RTS1./(RTSGain1*100E-6))+d1;
RTS2 = c2.*(Ro2+RTS2./(RTSGain2*100E-6))+d2;
RTS3 = c3.*(Ro3+RTS3./(RTSGain3*100E-6))+d3;
RTS4 = c4.*(Ro4+RTS4./(RTSGain4*100E-6))+d4;
RTS5 = c5.*(Ro5+RTS5./(RTSGain5*100E-6))+d5;
RTS6 = c6.*(Ro6+RTS6./(RTSGain6*100E-6))+d6;

Tblade6   = RTS6 + (mean(RTS1(1:100)) - mean(RTS6(1:100)));
Tblade5   = RTS5 + (mean(RTS1(1:100)) - mean(RTS5(1:100)));
Tblade4   = RTS4;
Tblade3   = RTS3;
Tblade2   = RTS2;
Tblade1   = RTS1;

%RTS5=RTS4;       %RTS5 and RTS6 too noisy to be used
%RTS6=RTS4;


Pfree=Pfree+Po;
Pplen=Pplen+Pfree;
Pratio=Pplen./Pfree;
Tratio=(Tfree+273.15)./(Tplen+273.15);

%Pressures and Mach Numbers at all locations
Kul1 = Kul1*1000/(KG1*KS1)+Po;
Kul2 = Kul2*1000/(KG2*KS2)+Po;
Kul3 = Kul3*1000/(KG3*KS3)+Po;
Kul4 = Kul4*1000/(KG4*KS4)+Po;
Kul5 = Kul5*1000/(KG5*KS5)+Po;
Kul6 = Kul6*1000/(KG6*KS6)+Po;
M1   = (5*((Pfree./Kul1).^(2/7)-1)).^0.5;
M2   = (5*((Pfree./Kul2).^(2/7)-1)).^0.5;
M3   = (5*((Pfree./Kul3).^(2/7)-1)).^0.5;
M4   = (5*((Pfree./Kul4).^(2/7)-1)).^0.5;
M5   = (5*((Pfree./Kul5).^(2/7)-1)).^0.5;
M6   = (5*((Pfree./Kul6).^(2/7)-1)).^0.5;
M1out= mean(M1(front:back));
M2out= mean(M2(front:back));
M3out= mean(M3(front:back));
M4out= mean(M4(front:back));
M5out= mean(M5(front:back));
M6out= mean(M6(front:back));



%Free-Stream Mach Number at all coolant exit locations
Mpg  =(5*((Pfree./(Pfree-Ppg)).^(2/7)-1)).^0.5;
Mpn  =(5*((Pfree./(Pfree-Ppn)).^(2/7)-1)).^0.5;
Mn   =(5*((Pfree./(Pfree-Pn)).^(2/7)-1)).^0.5;
```

```
   Msn1 =(5*((Pfree./(Pfree-Psn1)).^(2/7)-1)).^0.5;
   Msn2 =(5*((Pfree./(Pfree-Psn2)).^(2/7)-1)).^0.5;
   Msg  =(5*((Pfree./(Pfree-Psg)).^(2/7)-1)).^0.5;

   %Density Ratio at all coolant exit locations (the same for all 
locations)
   DR   =Pratio.^(2/7).*Tratio;    %coolant/freestream

   %Momentum Ratio at all coolant exit locations
   MRpg  =5*(Pratio.^(2/7).*(1+0.2*Mpg.^2)-1)./Mpg.^2;    
%coolant/freestream
   MRpn  =5*(Pratio.^(2/7).*(1+0.2*Mpn.^2)-1)./Mpn.^2;
   MRn   =5*(Pratio.^(2/7).*(1+0.2*Mn.^2)-1)./Mn.^2;
   MRsn1 =5*(Pratio.^(2/7).*(1+0.2*Msn1.^2)-1)./Msn1.^2;
   MRsn2 =5*(Pratio.^(2/7).*(1+0.2*Msn2.^2)-1)./Msn2.^2;
   MRsg  =5*(Pratio.^(2/7).*(1+0.2*Msg.^2)-1)./Msg.^2;

   %Blowing Ratio at all coolant exit locations
   BRpg  =(DR.*MRpg).^0.5;    %coolant/freestream
   BRpn  =(DR.*MRpn).^0.5;
   BRn   =(DR.*MRn).^0.5;
   BRsn1 =(DR.*MRsn1).^0.5;
   BRsn2 =(DR.*MRsn2).^0.5;
   BRsg  =(DR.*MRsg).^0.5;

   %Massflow ratios
   mr    =real(Msn1./(1+0.2*Msn1.^2).^3);
   mrsn1 =mean(mr(front:back));
   mr    =real(Msn2./(1+0.2*Msn2.^2).^3);
   mrsn2 =mean(mr(front:back));
   mr    =real(Msg./(1+0.2*Msg.^2).^3);
   mrsg  =mean(mr(front:back));
   ms    =mrsn1+mrsn2+mrsg;
   mrsn1 =mrsn1./ms;
   mrsn2 =mrsn2./ms;
   mrsg  =mrsg./ms;


   % ---- Average Values, Cold Coolant  ----

%   coolant=mrsg*Tsg+mrsn2*Tsn2+mrsn1*Tsn1;
   coolant=Tsg;



   if typeflag == 1                     %Cooled Analysis

      Tr1    = Tfree - Tdrop1;
      Tr2    = Tfree - Tdrop2;
      Tr3    = Tfree - Tdrop3;
      Tr4    = Tfree - Tdrop4;
      Tr5    = Tfree - Tdrop5;
      Tr6    = Tfree - Tdrop6;


      deltaT1     = (Tr1 - Tblade1)./(Tr1 - coolant);
      deltaT2     = (Tr2 - Tblade2)./(Tr2 - coolant);
      deltaT3     = (Tr3 - Tblade3)./(Tr3 - coolant);
      deltaT4     = (Tr4 - Tblade4)./(Tr4 - coolant);
      deltaT5     = (Tr5 - Tblade5)./(Tr5 - coolant);
      deltaT6     = (Tr6 - Tblade6)./(Tr6 - coolant);

      qprime1     = q1./(Tr1 - coolant);
      qprime2     = q2./(Tr2 - coolant);
      qprime3     = q3./(Tr3 - coolant);
      qprime4     = q4./(Tr4 - coolant);
      qprime5     = q5./(Tr5 - coolant);
      qprime6     = q6./(Tr6 - coolant);

      [p1,S1] = POLYFIT( deltaT1( front:back ), qprime1( front:back ) ,
1);
      [p2,S2] = POLYFIT( deltaT2( front:back ), qprime2( front:back ) ,
1);
      [p3,S3] = POLYFIT( deltaT3( front:back ), qprime3( front:back ) ,
1);
      [p4,S4] = POLYFIT( deltaT4( front:back ), qprime4( front:back ) ,
1);
      [p5,S5] = POLYFIT( deltaT5( front:back ), qprime5( front:back ) ,
1);
      [p6,S6] = POLYFIT( deltaT6( front:back ), qprime6( front:back ) ,
1);

      h1=10000*p1(1);
      h2=10000*p2(1);
      h3=10000*p3(1);
      h4=10000*p4(1);
      h5=10000*p5(1);
      h6=10000*p6(1);

      n1=-p1(2)./p1(1);
      n2=-p2(2)./p2(1);
      n3=-p3(2)./p3(1);
      n4=-p4(2)./p4(1);
      n5=-p5(2)./p5(1);
      n6=-p6(2)./p6(1);

      mass    =
3.9941*sqrt(75.0437.*Pdp.*(Pup+Po)./(460+(Torr*9/5)+32));


      %  ----  Display Summary of h and eta  ----

      disp(' ');
      disp(' ');
      disp('Location          h          eta');
      disp(['   1           ', num2str(h1,5),'      ', num2str(n1,4)]);
      disp(['   2           ', num2str(h2,5),'      ', num2str(n2,4)]);
      disp(['   3           ', num2str(h3,5),'      ', num2str(n3,4)]);
      disp(['   4           ', num2str(h4,5),'      ', num2str(n4,4)]);
      disp(['   5           ', num2str(h5,5),'      ', num2str(n5,4)]);
      disp(['   6           ', num2str(h6,5),'      ', num2str(n6,4)]);
%      disp(['Mass flow : ', num2str(mean(mass(front:back)),5)]);
%      disp(' ');
%      disp(['Tblade 1 2 3 4 5 6 :  ',num2str(Tblade1(times),3),'
',num2str(Tblade2(times),3),' ',...
%            num2str(Tblade3(times),3),' ',num2str(Tblade4(times),3),'
',num2str(Tblade5(times),3),' ',num2str(Tblade6(times),3)]);
%      disp(['Ttc Tsn1 Tsn2 Tsg Tavg :',num2str(Tplen(times),5),'
',num2str(Tsn1(times),5),' ',...
%            num2str(Tsn2(times),5),' ',num2str(Tsg(times),5),'
',num2str(coolant(times),5)]);
%      disp(['       Sn1 Sn2 Sg']);
%      disp(['DR:  ',num2str(DR(times),3),' ',num2str(DR(times),3),'
',num2str(DR(times),3)]);
%      disp(['BR:  ',num2str(BRsn1(times),3),'
',num2str(BRsn2(times),3),' ',num2str(BRsg(times),3)]);
%      disp(['MR:  ',num2str(MRsn1(times),3),'
',num2str(MRsn2(times),3),' ',num2str(MRsg(times),3)]);

   end


   if typeflag == 0                %Uncooled Analysis
      deltaT1 = Tfree - Tblade1;
      deltaT2 = Tfree - Tblade2;
      deltaT3 = Tfree - Tblade3;
      deltaT4 = Tfree - Tblade4;
      deltaT5 = Tfree - Tblade5;
      deltaT6 = Tfree - Tblade6;

      qprime1=q1;
      qprime2=q2;
      qprime3=q3;
      qprime4=q4;
      qprime5=q5;
      qprime6=q6;

      [p1,S1] = POLYFIT( deltaT1( front:back ), qprime1( front:back ) ,
1);
      [p2,S2] = POLYFIT( deltaT2( front:back ), qprime2( front:back ) ,
1);
      [p3,S3] = POLYFIT( deltaT3( front:back ), qprime3( front:back ) ,
1);
      [p4,S4] = POLYFIT( deltaT4( front:back ), qprime4( front:back ) ,
1);
      [p5,S5] = POLYFIT( deltaT5( front:back ), qprime5( front:back ) ,
1);
      [p6,S6] = POLYFIT( deltaT6( front:back ), qprime6( front:back ) ,
1);

      h1=10000*p1(1);
      h2=10000*p2(1);
      h3=10000*p3(1);
      h4=10000*p4(1);
      h5=10000*p5(1);
      h6=10000*p6(1);

      Tdrop1=-p1(2)/p1(1);
      Tdrop2=-p2(2)/p2(1);
      Tdrop3=-p3(2)/p3(1);
      Tdrop4=-p4(2)/p4(1);
      Tdrop5=-p5(2)/p5(1);
      Tdrop6=-p6(2)/p6(1);

      disp(' ');
      disp(' ');
      disp('Location          h        Tdrop       Mach#');
      disp(['   1           ', num2str(h1,5),'      ',
num2str(Tdrop1,4),'      ', num2str(M1out,4)]);
      disp(['   2           ', num2str(h2,5),'      ',
num2str(Tdrop2,4),'      ', num2str(M2out,4)]);
      disp(['   3           ', num2str(h3,5),'      ',
num2str(Tdrop3,4),'       ','NaN']);
      disp(['   4           ', num2str(h4,5),'      ',
num2str(Tdrop4,4),'      ', num2str(M4out,4)]);
      disp(['   5           ', num2str(h5,5),'      ',
num2str(Tdrop5,4),'      ', num2str(M5out,4)]);
      disp(['   6           ', num2str(h6,5),'      ',
num2str(Tdrop6,4),'       ', num2str(M6out,4)]);
      disp(' ');
      disp(' ');

   end

   %  ----  Output Graphs  -----

   if (typeflag==1)
      disp(['Outputing Graphs...']),disp(' ')
      disp(['File #' num2str(1) ' ' datanames])
      disp('1=Blade Temps          2=Time History'   );
      disp('3=Slope and Intercept  4=Heat Flux');
      disp('5=Span Mach Number');
      disp('6=Coolant Temperatures ');
      disp('7=Pressure Ratio ');
      disp('8=Density Ratio ');
      disp('9=Blowing Ratio ');
      disp('10=Momentum Ratio ');
      disp('11=h vs time ');
      disp('13=STC vs. RTS');
      graphs=input('See what graphs? ','s');
   elseif (typeflag==0)
      disp(['Outputing Graphs...']),disp(' ')
      disp(['File #' num2str(1) ' ' datanames])
      disp('1=Blade Temps          2=Time History'   );
      disp('3=Slope and Intercept  4=Heat Flux');
      disp('5=Span Mach Number     12=h vs time');
      graphs=input('See what graphs? ','s');
   end
      graphs=str2num(graphs);


   if isempty(graphs)
      graphs='0';
   end

   if (sum(graphs==1))      % ----   Blade Temperatures
      figure
      set(gcf,'Units','Normalized','Position',[.2,.05,1-.2,.8])

plot(t,Tblade1,t,Tblade2,t,Tblade3,t,Tblade4,t,Tblade5,t,Tblade6);

legend('Tblade1','Tblade2','Tblade3','Tblade4','Tblade5','Tblade6')
      title(['Blade Temperatures  '],'FontWeight','bold')
      xlabel('Time, s')
      ylabel('Temperature, °C')
      set(gcf,'color',[1 1 1]);
   end

   if (sum(graphs==2) )  % ----   Time History
      figure
      Tr1   = Tfree - Tdrop1;
      Taw1  = Tr1 + n1*(coolant - Tr1);
      Tad1  = Taw1;
      set(gcf,'Units','Normalized','Position',[.2,.05,1-.2,.8])
      Xone = {t t};
      Yone = {Tfree Tblade3};
      Xtwo = {t};
      Ytwo = {q3};
      s1   = {150 150};
      m1   = {'d' 'o'};
      s2   = {10};
      m2   = {'-'};
      [AX,H1,H2,H3,H4]=fancyplot2(Xone,Yone,s1,m1,Xtwo,Ytwo,s2,m2)
      axes(AX(1));
      ylabel('Temperature, °C')
      hold on;
      bot=get(gca,'Ylim');
      plot([back/100 back/100],[bot(1) bot(2)],'--','color','k')
      plot([front/100 front/100],[bot(1) bot(2)],'--','color','k')
      hold off;
      line([0 40],[bot(2) bot(2)],'color','k')
      axes(AX(2));
      ylabel('q, W/cm²')
      set(H3,'linestyle','-')
      set(H4,'linestyle','-','color',[0 0.65 0])
      %set(H1(4),'color','k')
      %set(H3(4),'color','k')
      set(H1(2),'color','m')
      set(H3(2),'color','m')
      %set(H1(2),'color','r')
      %set(H3(2),'color','r')
      set(H2,'color',[0 0.65 0])
      [L1,L2]=legend([H3 H4],'T_{freestream}', 'T_{blade}', 'q')
      set(L2(1:3,2),'FontSize',.183,'FontWeight','bold')
      set(H3,'linestyle','none')
      set(H4,'linestyle','none')
      Title(['Time History  '],'FontWeight','bold')
      xlabel('Time, s')
      set(gcf,'color',[1 1 1])
      set(gcf,'color',[1 1 1]);
      box on;
   end

   if (sum(graphs==3) )   % ----   h and eta graph
```

```
      figure
      set(gcf,'Units','Normalized','Position',[.2,.05,1-.2,.8])
      hold on;
      plot(deltaT1(front:back),qprime1(front:back),'.')
      endpts=[ deltaT1(front) deltaT1(back) ];
      fit=polyval(p1,endpts);
      plot(endpts,fit,'r');
      hold off;
      set(gcf,'color',[1 1 1]);
      if (typeflag==1)
         Title(['h and \eta,  ' ' Location #1'],'FontWeight','bold')
         Xlabel('(T_r - T_{blade})/(T_r - T_{coolant})  [ - ]')
         Ylabel('q/(T_r - T_{coolant})  [ W/cm² °C ]')
         xtext=endpts(2);
         ytext=( qprime1(front) + qprime1(back) )/2;
         text(xtext,ytext,['h= ' num2str(h1) ' W/m² °C    \eta= '
num2str(n1) ])
      elseif(typeflag==0)
         Title(['h and T_{drop},  ' datanames ' Location
#1'],'FontWeight','bold')
         Xlabel('(T_r - T_{blade})  [ °C ]')
         Ylabel('q  [ W/cm² ]')
         xtext=endpts(2);
         ytext=( qprime1(front) + qprime1(back) )/2;
         text(xtext,ytext,['h= ' num2str(h1) ' W/m² °C    T_{drop}= '
num2str(Tdrop1) ])
      end

      figure
      set(gcf,'Units','Normalized','Position',[.2,.05,1-.2,.8])
      hold on;
      plot(deltaT2(front:back),qprime2(front:back),'.')
      endpts=[ deltaT2(front) deltaT2(back) ];
      fit=polyval(p2,endpts);
      plot(endpts,fit,'r');
      hold off;
      set(gcf,'color',[1 1 1]);
      if (typeflag==1)
         Title(['h and \eta,  ' datanames ' Location
#2'],'FontWeight','bold')
         Xlabel('(T_r - T_{blade})/(T_r - T_{coolant})  [ - ]')
         Ylabel('q/(T_r - T_{coolant})  [ W/cm² °C ]')
         xtext=endpts(2);
         ytext=( qprime2(front) + qprime2(back) )/2;
         text(xtext,ytext,['h= ' num2str(h2) ' W/m² °C    \eta= '
num2str(n2) ])
      elseif(typeflag==0)
         Title(['h and T_{drop},  ' datanames ' Location
#2'],'FontWeight','bold')
         Xlabel('(T_r - T_{blade})  [ °C ]')
         Ylabel('q  [ W/cm² ]')
         xtext=endpts(2);
         ytext=( qprime2(front) + qprime2(back) )/2;
         text(xtext,ytext,['h= ' num2str(h2) ' W/m² °C    T_{drop}= '
num2str(Tdrop2) ])
      end
      figure
      set(gcf,'Units','Normalized','Position',[.2,.05,1-.2,.8])
      hold on;
      plot(deltaT3(front:back),qprime3(front:back),'.')
      endpts=[ deltaT3(front) deltaT3(back) ];
      fit=polyval(p3,endpts);
      plot(endpts,fit,'r');
      hold off;
      set(gcf,'color',[1 1 1]);
      if (typeflag==1)
         Title(['h and \eta,  ' ' Location #3'],'FontWeight','bold')
         Xlabel('(T_r - T_{blade})/(T_r - T_{coolant})  [ - ]')
         Ylabel('q/(T_r - T_{coolant})  [ W/cm² °C ]')
         xtext=endpts(2);
         ytext=( qprime3(front) + qprime3(back) )/2;
         text(xtext,ytext,['h= ' num2str(h3) ' W/m² °C    \eta= '
num2str(n3) ])
      elseif(typeflag==0)
         Title(['h and T_{drop},  ' datanames ' Location
#3'],'FontWeight','bold')
         Xlabel('(T_r - T_{blade})  [ °C ]')
         Ylabel('q  [ W/cm² ]')
         xtext=endpts(2);
         ytext=( qprime3(front) + qprime3(back) )/2;
         text(xtext,ytext,['h= ' num2str(h3) ' W/m² °C    T_{drop}= '
num2str(Tdrop3) ])
      end

      figure
      set(gcf,'Units','Normalized','Position',[.2,.05,1-.2,.8])
      hold on;
      plot(deltaT4(front:back),qprime4(front:back),'.')
      endpts=[ deltaT4(front) deltaT4(back) ];
      fit=polyval(p4,endpts);
      plot(endpts,fit,'r');
      hold off;
      set(gcf,'color',[1 1 1]);
      if (typeflag==1)
         Title(['h and \eta,  ' datanames ' Location
#4'],'FontWeight','bold')
         Xlabel('(T_r - T_{blade})/(T_r - T_{coolant})  [ - ]')
         Ylabel('q/(T_r - T_{coolant})  [ W/cm² °C ]')
         xtext=endpts(2);
         ytext=( qprime4(front) + qprime4(back) )/2;
         text(xtext,ytext,['h= ' num2str(h4) ' W/m² °C    \eta= '
num2str(n4) ])
      elseif(typeflag==0)
         Title(['h and T_{drop},  ' datanames ' Location
#4'],'FontWeight','bold')
         Xlabel('(T_r - T_{blade})  [ °C ]')
         Ylabel('q  [ W/cm² ]')
         xtext=endpts(2);
         ytext=( qprime4(front) + qprime4(back) )/2;
         text(xtext,ytext,['h= ' num2str(h4) ' W/m² °C    T_{drop}= '
num2str(Tdrop4) ])
```

```
      end

      figure
      set(gcf,'Units','Normalized','Position',[.2,.05,1-.2,.8])
      hold on;
      plot(deltaT5(front:back),qprime5(front:back),'.')
      endpts=[ deltaT5(front) deltaT5(back) ];
      fit=polyval(p5,endpts);
      plot(endpts,fit,'r');
      hold off;
      set(gcf,'color',[1 1 1]);
      if (typeflag==1)
         Title(['h and \eta,  ' datanames ' Location
#5'],'FontWeight','bold')
         Xlabel('(T_r - T_{blade})/(T_r - T_{coolant})  [ - ]')
         Ylabel('q/(T_r - T_{coolant})  [ W/cm² °C ]')
         xtext=endpts(2);
         ytext=( qprime5(front) + qprime5(back) )/2;
         text(xtext,ytext,['h= ' num2str(h5) ' W/m² °C    \eta= '
num2str(n5) ])
      elseif(typeflag==0)
         Title(['h and T_{drop},  ' datanames ' Location
#5'],'FontWeight','bold')
         Xlabel('(T_r - T_{blade})  [ °C ]')
         Ylabel('q  [ W/cm² ]')
         xtext=endpts(2);
         ytext=( qprime5(front) + qprime5(back) )/2;
         text(xtext,ytext,['h= ' num2str(h5) ' W/m² °C    T_{drop}= '
num2str(Tdrop5) ])
      end


      figure
      set(gcf,'Units','Normalized','Position',[.2,.05,1-.2,.8])
      hold on;
      plot(deltaT6(front:back),qprime6(front:back),'.')
      endpts=[ deltaT6(front) deltaT6(back) ];
      fit=polyval(p6,endpts);
      plot(endpts,fit,'r');
      hold off;
      set(gcf,'color',[1 1 1]);
      if (typeflag==1)
         Title(['h and \eta,  ' datanames ' Location
#6'],'FontWeight','bold')
         Xlabel('(T_r - T_{blade})/(T_r - T_{coolant})  [ - ]')
         Ylabel('q/(T_r - T_{coolant})  [ W/cm² °C ]')
         xtext=endpts(2);
         ytext=( qprime6(front) + qprime6(back) )/2;
         text(xtext,ytext,['h= ' num2str(h6) ' W/m² °C    \eta= '
num2str(n6) ])
      elseif(typeflag==0)
         Title(['h and T_{drop},  ' datanames ' Location
#6'],'FontWeight','bold')
         Xlabel('(T_r - T_{blade})  [ °C ]')
         Ylabel('q  [ W/cm² ]')
         xtext=endpts(2);
         ytext=( qprime6(front) + qprime6(back) )/2;
         text(xtext,ytext,['h= ' num2str(h6) ' W/m² °C    T_{drop}= '
num2str(Tdrop6) ])
      end

   end


   if (sum(graphs==4))      % ----   Heat Flux
      figure
      set(gcf,'Units','Normalized','Position',[.2,.05,1-.2,.8])
      plot(t,q6,t,q5,t,q4,t,q3,t,q2,t,q1)
      legend('q6','q5','q4','q3','q2','q1')
      title(['Heat Flux  '],'FontWeight','bold')
      xlabel('Time, s')
      ylabel('Flux, W/m^2-°C')
      set(gcf,'color',[1 1 1]);
   end

   if (sum(graphs==5))      % ----   Streamwise Mach-Number Profile
      figure
      set(gcf,'Units','Normalized','Position',[.2,.05,1-.2,.8])
      plot(t,M1,t,M2,t,M4,t,M5,t,M6);
      legend('1','2','4','5','6')
      title(['Streamwise Mach-Number Profile,  ',
datanames],'FontWeight','bold')
      xlabel('Time, s')
      ylabel('Pressure, psia')
      set(gcf,'color',[1 1 1]);
   end

   if (sum(graphs==6))
      figure
      set(gcf,'Units','Normalized','Position',[.2,.05,1-.2,.8])
      plot(t,Tplen,t,Tsn1,t,Tsn2,t,Tsg,t,coolant)
      legend('T_{plenum}','T_{sn1}','T_{sn2}','T_{sg}','T_{average}')
      title('Coolant Temperatures')
      xlabel('Time  [ s ]')
      ylabel('Temperature  [ °C ]')
      set(gcf,'color',[1 1 1]);
   end

   if (sum(graphs==7) )
      figure
      set(gcf,'Units','Normalized','Position',[.2,.05,1-.2,.8])
      H1=plot(t(front:back),Pratio(front:back));
      title('Time History of Pressure Ratio','FontWeight','bold')
      xlabel('Time  [ s ]')
      ylabel('Pressure Ratio  [ - ]')
      set(gcf,'color',[1 1 1]);
   end


   if (sum(graphs==8) )
      figure
      set(gcf,'Units','Normalized','Position',[.2,.05,1-.2,.8])
      H1=plot(t(front:back),DR(front:back));
      title('Time History of Density Ratio','FontWeight','bold')
      xlabel('Time  [ s ]')
      ylabel('Density Ratio  [ - ]')
      set(gcf,'color',[1 1 1]);
   end


   if (sum(graphs==9) )
      figure
      set(gcf,'Units','Normalized','Position',[.2,.05,1-.2,.8])

H1=plot(t(front:back),BRsn1(front:back),t(front:back),BRsn2(front:back)
,t(front:back),BRsg(front:back));
      legend(H1,'SN1','SN2','SG')
      title('Time History of Blowing Ratio','FontWeight','bold')
      xlabel('Time  [ s ]')
      ylabel('Blowing Ratio  [ - ]')
      max1=max([max(BRsn1(front:back)) max(BRsn2(front:back))
max(BRsg(front:back))]);
      min1=min([min(BRsn1(front:back)) min(BRsn2(front:back))
min(BRsg(front:back))]);
      ytext=(max1-min1)/2+min1;
      xtext= 0.1*(max(t(front:back))-
min(t(front:back)))+min(t(front:back));
      text(xtext,ytext,['Density Ratio= ' num2str(mean(DR(front:back)))
])
      set(gcf,'color',[1 1 1]);
   end

   if (sum(graphs==10) )
      figure
      set(gcf,'Units','Normalized','Position',[.2,.05,1-.2,.8])

H1=plot(t(front:back),MRsn1(front:back),t(front:back),MRsn2(front:back)
,t(front:back),MRsg(front:back));
      legend(H1,'SN1','SN2','SG')
      title('Time History of Momentum Ratio','FontWeight','bold')
      xlabel('Time  [ s ]')
      ylabel('Momentum Ratio  [ - ]')
      max1=max([max(MRsn1(front:back)) max(MRsn2(front:back))
max(MRsg(front:back))]);
      min1=min([min(MRsn1(front:back)) min(MRsn2(front:back))
min(MRsg(front:back))]);
      ytext=(max1-min1)/2+min1;
      xtext= 0.1*(max(t(front:back))-
min(t(front:back)))+min(t(front:back));
      text(xtext,ytext,['Density Ratio= ' num2str(mean(DR(front:back)))
])
      set(gcf,'color',[1 1 1]);
   end

   if (sum(graphs==11))      % ----   Heat Flux vs Time Graphs
      figure
      set(gcf,'Units','Normalized','Position',[.2,.05,1-.2,.8])
      Tr1   = Tfree - Tdrop1;
      Taw1  = Tr1 + n1*(coolant - Tr1);
      Tad1  = Taw1;
      htime = 10000*q1./(Taw1 - Tblade1);
      l1    = plot(t,htime);
      l2    = line([t(1) t(length(t))], [h1 h1],'linestyle','--
','color',[0 0.5 0]);
      l3    = legend([l1 l2],'h(t)','h_{avg}');
      set(l3,'FontWeight','bold');
      set(gca,'YLim',[0 1500]);
      set(gca,'XLim',[0 22]);
      title(['Heat Flux vs. Time '],'FontWeight','bold')
      xlabel('Time, s')
      ylabel('Flux, W/m^2-°C')
      text(1,870,['h_{avg} = ' num2str(round(h1)) ' W/m^2 °C']);
      set(gcf,'color',[1 1 1]);

      figure
      set(gcf,'Units','Normalized','Position',[.2,.05,1-.2,.8])
      Xone  = {t t t t t};
      Yone  = {Tfree Tr1 Taw1 Tblade1 coolant};
      s1    = {150 150 150 150 150};
      m1    = {'o' 'd' 's' 'v' '^'};
      Xtwo  = {t};
      Ytwo  = {q1};
      s2    = {150};
      m2    = {'-'};
      [AX,H1,H2,H3,H4] = fancyplot2(Xone,Yone,s1,m1,Xtwo,Ytwo,s2,m2);
      set(H3(1),'color','b','linestyle','-');
      set(H3(2),'color',[0.75 0 0.75],'linestyle','-');
      set(H3(3),'color','r','linestyle','-');
      set(H3(4),'color','m','linestyle','-');
      set(H3(5),'color','k','linestyle','-');
      set(H4(1),'color', [0 0.5 0],'linestyle','-');
      set(H1(1),'color','b','linestyle','-');
      set(H1(2),'color',[0.75 0 0.75],'linestyle','-');
      set(H1(3),'color','r','linestyle','-');
      set(H1(4),'color','m','linestyle','-');
      set(H1(5),'color','k','linestyle','-');
      set(H2(1),'color', [0 0.5 0],'linestyle','-');
      title(['Temperature Time History '],'FontWeight','bold')
      xlabel('Time, s')
      axes(AX(1));
      ylabel('Temperature, °C');
      axes(AX(2));
      ylabel('Flux, W/m^2-°C');
      legend([H3 H4],'Tfree','Trec', 'Taw', 'Twall', 'Tc', 'q');
      set(H3,'linestyle','none');
      set(H4,'linestyle','none');
      r1 = get(AX(2),'YLim');
      r2 = get(AX(2),'XLim');
      line([22 22], [r1(1) r1(2)],'linestyle','--');
      line([r2(1) r2(2)], [0 0],'linestyle','--');
      set(gcf,'color',[1 1 1]);
      box on;
   end

   if (sum(graphs==12))      % ----   Heat Flux vs Time Graphs,.
Uncooled
      figure
      set(gcf,'Units','Normalized','Position',[.2,.05,1-.2,.8])
      Tr1   = Tfree - Tdrop1;
      Taw1  = Tr1;
      htime = 10000*q1./(Taw1 - Tblade1);
      l1    = plot(t,htime);
      l2    = line([t(1) t(length(t))], [h1 h1],'linestyle','--
','color',[0 0.5 0]);
      l3    = legend([l1 l2],'h(t)','h_{avg}');
      set(l3,'FontWeight','bold');
      set(gca,'YLim',[0 1500]);
      set(gca,'XLim',[0 23]);
      title(['Heat Flux vs. Time '],'FontWeight','bold')
      xlabel('Time, s')
      ylabel('Flux, W/m^2-°C')
      text(1,(round(h1)+40),['h_{avg} = ' num2str(round(h1)) ' W/m^2
°C']);
      set(gcf,'color',[1 1 1]);

      figure
      set(gcf,'Units','Normalized','Position',[.2,.05,1-.2,.8])
      Xone  = {t t t};
      Yone  = {Tfree Taw1 Tblade1};
      s1    = {150 150 150};
      m1    = {'o' 'd' 's'};
      Xtwo  = {t};
      Ytwo  = {q1};
      s2    = {150};
      m2    = {'-'};
      [AX,H1,H2,H3,H4] = fancyplot2(Xone,Yone,s1,m1,Xtwo,Ytwo,s2,m2);
      set(H3(1),'color','b','linestyle','-');
      set(H3(2),'color',[0.75 0 0.75],'linestyle','-');
      set(H3(3),'color','r','linestyle','-');
      set(H4(1),'color', [0 0.5 0],'linestyle','-');
      set(H1(1),'color','b','linestyle','-');
      set(H1(2),'color',[0.75 0 0.75],'linestyle','-');
      set(H1(3),'color','r','linestyle','-');
      set(H2(1),'color', [0 0.5 0],'linestyle','-');
      title(['Temperature Time History '],'FontWeight','bold')
      xlabel('Time, s')
      axes(AX(1));
      ylabel('Temperature, °C');
      axes(AX(2));
      set(gca,'YLim',[-1 8]);
      ylabel('Flux, W/m^2-°C');
      legend([H3 H4],'Tfree', 'Taw', 'Twall', 'q');
      set(H3,'linestyle','none');
      set(H4,'linestyle','none');
      r1 = get(AX(2),'YLim');
      r2 = get(AX(2),'XLim');
      line([23 23], [r1(1) r1(2)],'linestyle','--');
      line([r2(1) r2(2)], [0 0],'linestyle','--');
      set(gcf,'color',[1 1 1]);
      box on;
   end

   if (sum(graphs==13))      % ----   STC vs. RTS
      figure
      dent = mean(Tblade1(1:100)) - mean(RTS1(1:100));
```

```
Xone  = {t};
Yone  = {Tblade1};
s1    = {150};
m1    = {'o-'};
Xtwo  = {t};
Ytwo  = {RTS1+dent};
s2    = {150};
m2    = {'d-'};
[AX,H1,H2,H3,H4] = fancyplot2(Xone,Yone,s1,m1,Xtwo,Ytwo,s2,m2);
set(gcf,'Units','Normalized','Position',[.2,.05,1-.2,.8])
legend([H3 H4],'STC #1','RTS #1')
set(H3,'linestyle','none')
set(H4,'linestyle','none')
title(['STC versus RTS'],'FontWeight','bold')
xlabel('Time, s')
ylabel('Temperature, °C')
set(gcf,'color',[1 1 1]);

figure
dent = mean(Tblade2(1:100)) - mean(RTS2(1:100));
Xone  = {t};
Yone  = {Tblade2};
s1    = {150};
m1    = {'o-'};
Xtwo  = {t};
Ytwo  = {RTS2+dent};
s2    = {150};
m2    = {'d-'};
[AX,H1,H2,H3,H4] = fancyplot2(Xone,Yone,s1,m1,Xtwo,Ytwo,s2,m2);
set(gcf,'Units','Normalized','Position',[.2,.05,1-.2,.8])
legend([H3 H4],'STC #2','RTS #2')
set(H3,'linestyle','none')
set(H4,'linestyle','none')
title(['STC versus RTS'],'FontWeight','bold')
xlabel('Time, s')
ylabel('Temperature, °C')
set(gcf,'color',[1 1 1]);

figure
dent = mean(Tblade3(1:100)) - mean(RTS3(1:100));
Xone  = {t};
Yone  = {Tblade1};
s1    = {150};
m1    = {'o-'};
Xtwo  = {t};
Ytwo  = {RTS3+dent};
s2    = {150};
m2    = {'d-'};
[AX,H1,H2,H3,H4] = fancyplot2(Xone,Yone,s1,m1,Xtwo,Ytwo,s2,m2);
set(gcf,'Units','Normalized','Position',[.2,.05,1-.2,.8])
legend([H3 H4],'STC #3','RTS #3')
set(H3,'linestyle','none')
set(H4,'linestyle','none')
title(['STC versus RTS'],'FontWeight','bold')
xlabel('Time, s')
ylabel('Temperature, °C')
set(gcf,'color',[1 1 1]);
```

```
figure
dent = mean(Tblade4(1:100)) - mean(RTS4(1:100));
Xone  = {t};
Yone  = {Tblade4};
s1    = {150};
m1    = {'o-'};
Xtwo  = {t};
Ytwo  = {RTS4+dent};
s2    = {150};
m2    = {'d-'};
[AX,H1,H2,H3,H4] = fancyplot2(Xone,Yone,s1,m1,Xtwo,Ytwo,s2,m2);
set(gcf,'Units','Normalized','Position',[.2,.05,1-.2,.8])
legend([H3 H4],'STC #4','RTS #4')
set(H3,'linestyle','none')
set(H4,'linestyle','none')
title(['STC versus RTS'],'FontWeight','bold')
xlabel('Time, s')
ylabel('Temperature, °C')
set(gcf,'color',[1 1 1]);

figure
dent = mean(Tblade5(1:100)) - mean(RTS5(1:100));
Xone  = {t};
Yone  = {Tblade5};
s1    = {150};
m1    = {'o-'};
Xtwo  = {t};
Ytwo  = {RTS5+dent};
s2    = {150};
m2    = {'d-'};
[AX,H1,H2,H3,H4] = fancyplot2(Xone,Yone,s1,m1,Xtwo,Ytwo,s2,m2);
set(gcf,'Units','Normalized','Position',[.2,.05,1-.2,.8])
legend([H3 H4],'STC #5','RTS #5')
set(H3,'linestyle','none')
set(H4,'linestyle','none')
title(['STC versus RTS'],'FontWeight','bold')
xlabel('Time, s')
ylabel('Temperature, °C')
set(gcf,'color',[1 1 1]);

figure
dent = mean(Tblade6(1:100)) - mean(RTS1(6:100));
Xone  = {t};
Yone  = {Tblade6};
s1    = {150};
m1    = {'o-'};
Xtwo  = {t};
Ytwo  = {RTS6+dent};
s2    = {150};
m2    = {'d-'};
[AX,H1,H2,H3,H4] = fancyplot2(Xone,Yone,s1,m1,Xtwo,Ytwo,s2,m2);
set(gcf,'Units','Normalized','Position',[.2,.05,1-.2,.8])
legend([H3 H4],'STC #6','RTS #6')
set(H3,'linestyle','none')
set(H4,'linestyle','none')
title(['STC versus RTS'],'FontWeight','bold')
```

```
      xlabel('Time, s')
      ylabel('Temperature, °C')
      set(gcf,'color',[1 1 1]);
   end



% ---- Summary sheet for all runs (always outputs)
%clear textcell;
%header = str2mat(...
%   'Summary Sheet',...
%   '--------------------');
%if typeflag==0
%   strtype='Uncooled Run';
%   strsec=['Tdrop: ' num2str(Tdrop3)];
%elseif typeflag==1
%   strtype='Cooled Run Tdrop = 6.9°C';
%   strsec=['eta: ' num2str(n1)];
%else
%   strtype='Cooled Run Tdrop = Pr^.33';
%   strsec=['eta: ' num2str(n1)];
%end
%textcell = {...
%      ['Type: ' strtype],...
%      ['Length: ' num2str(len) ' points'],...
%      ['Window: ' num2str(front) ' - ' num2str(back)],...
%      ['q offset: ' num2str(qoffset) ' W/m²'],...
%      ['h: ' num2str(h1)) ' W/m² °C'],...
%      [strsec],...
%      ' '};
```

data922b_ambient.m

```
% data91898.m
% Data Ananylsis for Turbine Heat Transfer Group
% 6 triplet blade analysis
% O popp
%
%

close all;
format short g;
clear all;

% ---- Input Data Files ------------
% This is the section you change...
%
% typeflag = 0   No Coolant Flow
%                h = q / (Tfree - Tblade)
%                   Tdrop = (1-r)*u^2/2Cp
%
% typeflag = 1   Cold Coolant Flow (using measured Tdrop)
%                h = ( q / (Tr - Tcool) ) / ( (Tr - Tblade)/(Tr -
Tcool) )
%                n = (Taw - Tr) / (Tcool - Tr)
%                   Tr = Tfree - Tdrop (Tdrop=6.9°C)
```

```
%
% typeflag = 2   Cold Coolant Flow (using Pr approx for Tr)
%                h = ( q / (Tr - Tcool) ) / ( (Tr - Tblade)/(Tr -
Tcool) )
%                n = (Taw - Tr) / (Tcool - Tr)
%                   Tr = Tfree - Pr^.33 (Tdrop=0.89°C)
%
% typeflag = 4   Ambient Coolant Flow
%                h = q / (Tfree - Tblade)
%                   Tdrop = (1-r)*u^2/2Cp
%                Taw = Tfree - Tdrop
%
% front,back     Analysis window for each data set
%
% data           data(time,column#, run#)
%

names    ='e:\Afosr\Data\09_22\acaf3_922.asc';
typeflag = 4;
range    = 2;        %front & back = ± "range" percent using 100*(Tf-
Tc)/Tf
front    = 700;
back     = 2400;
qoffset  = 0;
To       = 26;
Po       = 13.7;
times    = 2040;   %Time of shock impact *100


% ---- OK now stop changing things from here on out  -------
% ---- The rest is golden.... -----------------------------

disp('Loading data files...')
[datanames,data]=loadset(names);
disp('Performing Analysis...')
NF=size(data,3);                                   %Number of Data
Files


   Torr         = data(:,16);
   Tfree        = data(:,9);
   Tplen        = data(:,8);
   Tsg       = data(:,15);
   Tsn2      = data(:,14);
   Tsn1      = data(:,13);
   Tn           = data(:,12);
   Tpn          = data(:,11);
   Tpg          = data(:,10);
   Tblade6   = data(:,2);
   Tblade5   = data(:,3);
   Tblade4   = data(:,3);         %using Tblade5
   Tblade3   = data(:,6);
   Tblade2   = data(:,6);
   Tblade1   = data(:,7);
   Psg       = data(:,1)*0;       %offset fudged
   Psn2      = data(:,1)*0;          %offset fudged
   Psn1      = data(:,1)*0;        %offset fudged
```

```
Pn        = data(:,1)*0;
Ppn       = data(:,1)*0;
Ppg       = data(:,1)*0;
Pdp       = data(:,17);       %offset fudged
Pup       = data(:,18);
Pfree     = data(:,19);
RTS5         = data(:,22)*0;
RTS6         = data(:,21)*0;
RTS4      = data(:,23)*0;
RTS3      = data(:,24)*0;
RTS2      = data(:,23)*0;
RTS1      = data(:,24)*0;
Pplen     = data(:,20);   % P or dP ???
Kul6      = data(:,21);
Kul5      = data(:,22);
Kul4      = data(:,23);
Kul3      = data(:,24);
Kul2      = data(:,25);
Kul1      = data(:,26);
HFS6         = data(:,28)*0;
HFS5      = data(:,29)*0;
HFS4      = data(:,28);
HFS3      = data(:,29);
HFS2      = data(:,30);
HFS1      = data(:,31);


len       = length(Tn);
t         = linspace(0,len/100,len)';


%HFS #1 CALIBRATION
HFSGain1 = 1000;
   g1       =  0;         % µV/((W/cm^2)°C)
hcal1    = 234;        % µV/(W/cm^2)
RTSGain1 = 500;
c1       = 3.93326;    % °C/Ohm
d1       = -495.849;   % °C
e1       = 0.261826;   % Ohm/°C
f1       = 125.7314;   % Ohm
Ro1      = f1+To*e1;

%HFS #2 CALIBRATION
HFSGain2 = 1000;
   g2       =  0;
hcal2    = 105;
RTSGain2 = 500;
c2       = 2.57488;
d2       = -518.276;
e2       = 0.39751;
f2       = 200.8981;
Ro2      = f2+To*e2;


%HFS #3 CALIBRATION
HFSGain3 = 1000;
   g3       =  0;
```

```
hcal3    = 145;
RTSGain3 = 500;
c3       = 3.9724;
d3       = -480.423;
e3       = 0.255602;
f3       = 120.7617;
Ro3      = f3+To*e3;


%HFS #4 CALIBRATION
HFSGain4 = 1000;
   g4       =  0;
hcal4    = 130;
RTSGain4 = 500;
c4       = 3.89763;
d4       = -488.813;
e4       = 0.261826;
f4       = 125.1414;
Ro4      = f4+To*e4;


%HFS #5 CALIBRATION
HFSGain5 = 1000;
g5       =  0;
hcal5    = 171;
RTSGain5 = 500;
c5       = 2.4478;
d5       = -477.73;
e5       = 0.415353;
f5       = 194.8265;
Ro5      = f5+To*e5;


%HFS #6 CALIBRATION
HFSGain6 = 1000;
   g6       =  0;
hcal6    = 153;
RTSGain6 = 500;
c6       = 4.13174;
d6       = -476.724;
e6       = 0.243154;
f6       = 115.3224;
Ro6      = f6+To*e6;

%Kulite Sensitivities
KS1      = 1.688;    % mV/psi
KS2      = 1.696;
KS3      = 1.727;
KS4      = 1.743;
KS5      = 1.724;
KS6      = 2.000;

%Kulite Gains
KG1      = 10;
KG2      = 10;
KG3      = 10;
KG4      = 10;
```

```
KG5      = 10;
KG6      = 10;



%Flux at all locations
q1   = HFS1.*1000000./(HFSGain1.*(g1.*Tblade1+hcal1));
q2   = HFS2.*1000000./(HFSGain2.*(g2.*Tblade2+hcal2));
q3   = HFS3.*1000000./(HFSGain3.*(g3.*Tblade3+hcal3));
q4   = HFS4.*1000000./(HFSGain4.*(g4.*Tblade4+hcal4));
q5   = HFS5.*1000000./(HFSGain5.*(g5.*Tblade5+hcal5));
q6   = HFS6.*1000000./(HFSGain6.*(g6.*Tblade6+hcal6));

%RTS Temperature at all locations
RTS1 = c1.*(Ro1+RTS1./(RTSGain1*100E-6))+d1;
RTS2 = c2.*(Ro2+RTS2./(RTSGain2*100E-6))+d2;
RTS3 = c3.*(Ro3+RTS3./(RTSGain3*100E-6))+d3;
RTS4 = c4.*(Ro4+RTS4./(RTSGain4*100E-6))+d4;
RTS5 = c5.*(Ro5+RTS5./(RTSGain5*100E-6))+d5;
RTS6 = c6.*(Ro6+RTS6./(RTSGain6*100E-6))+d6;
%RTS5=RTS4;       %RTS5 and RTS6 too noisy to be used
%RTS6=RTS4;


Pfree=Pfree+Po;
Pplen=Pplen+Pfree;
Pratio=Pplen./Pfree;
Tratio=(Tfree+273.15)./(Tplen+273.15);

%Pressures and Mach Numbers at all locations
Kul1 = Kul1*1000/(KG1*KS1)+Po;
Kul2 = Kul2*1000/(KG2*KS2)+Po;
Kul3 = Kul3*1000/(KG3*KS3)+Po;
Kul4 = Kul4*1000/(KG4*KS4)+Po;
Kul5 = Kul5*1000/(KG5*KS5)+Po;
Kul6 = Kul6*1000/(KG6*KS6)+Po;
M1   = (5*((Pfree./Kul1).^(2/7)-1)).^0.5;
M2   = (5*((Pfree./Kul2).^(2/7)-1)).^0.5;
M3   = (5*((Pfree./Kul3).^(2/7)-1)).^0.5;
M4   = (5*((Pfree./Kul4).^(2/7)-1)).^0.5;
M5   = (5*((Pfree./Kul5).^(2/7)-1)).^0.5;
M6   = (5*((Pfree./Kul6).^(2/7)-1)).^0.5;


%Find "front" and "back" @ coolant w/in 15% of Tfree
pdiff = Tfree-Tsg;
for j = 1:length(pdiff),
   Tdiff(j) = round(pdiff(j));
end
pig = find(Tdiff ==  range);
dig = find(Tdiff == -range);
front = 700;
%front = pig(1);
back  = dig(length(dig));
nofpts = back - front;
```

```
% ---- Average Values, Cold Coolant  ----

%   coolant=mrsg*Tsg+mrsn2*Tsn2+mrsn1*Tsn1;
   coolant=Tsg;


   if typeflag == 4                %Ambient Analysis
      deltaT1 = Tfree - Tblade1;
      deltaT2 = Tfree - Tblade2;
      deltaT3 = Tfree - Tblade3;
      deltaT4 = Tfree - Tblade4;
      deltaT5 = Tfree - Tblade5;
      deltaT6 = Tfree - Tblade6;

      qprime1=q1;
      qprime2=q2;
      qprime3=q3;
      qprime4=q4;
      qprime5=q5;
      qprime6=q6;

      [p1,S1] = POLYFIT( deltaT1( front:back ), qprime1( front:back) ,
1);
      [p2,S2] = POLYFIT( deltaT2( front:back ), qprime2( front:back) ,
1);
      [p3,S3] = POLYFIT( deltaT3( front:back ), qprime3( front:back) ,
1);
      [p4,S4] = POLYFIT( deltaT4( front:back ), qprime4( front:back) ,
1);
      [p5,S5] = POLYFIT( deltaT5( front:back ), qprime5( front:back) ,
1);
      [p6,S6] = POLYFIT( deltaT6( front:back ), qprime6( front:back) ,
1);

      h1 = 10000*p1(1);
      h2 = 10000*p2(1);
      h3 = 10000*p3(1);
      h4 = 10000*p4(1);
      h5 = 10000*p5(1);
      h6 = 10000*p6(1);

      Tdrop1=-p1(2)/p1(1);
      Tdrop2=-p2(2)/p2(1);
      Tdrop3=-p3(2)/p3(1);
      Tdrop4=-p4(2)/p4(1);
%      Tdrop5=-p5(2)/p5(1);
%      Tdrop6=-p6(2)/p6(1);

      Taw1 = Tfree -Tdrop1;
      Taw2 = Tfree -Tdrop2;
      Taw3 = Tfree -Tdrop3;
      Taw4 = Tfree -Tdrop4;
%      Taw5 = Tfree -Tdrop5;
%      Taw6 = Tfree -Tdrop6;

      disp(' ');
      disp(' ');
      disp('Location          h');
      disp(['   1           ', num2str(h1,5)]);
      disp(['   2           ', num2str(h2,5)]);
      disp(['   3           ', num2str(h3,5)]);
      disp(['   4           ', num2str(h4,5)]);
      disp(['   5           ', num2str(h5,5)]);
      disp(['   6           ', num2str(h6,5)]);
      disp(' ');
      disp(' ');

   end

   %  ----  Output Graphs  -----

   if (typeflag==4)
      disp(['Outputing Graphs...']),disp(' ')
      disp(['File #' num2str(1) ' ' datanames])
      disp('1=Blade Temps          2=Time History'   );
      disp('3=Slope and Intercept  4=Heat Flux');
      disp('5=h vs time');
      graphs=input('See what graphs? ','s');
   end
      graphs=str2num(graphs);


   if isempty(graphs)
      graphs='0';
   end

   if (sum(graphs==1))      % ----   Blade Temperatures
      figure
      set(gcf,'Units','Normalized','Position',[.2,.05,1-.2,.8])

plot(t,Tblade1,t,Tblade2,t,Tblade3,t,Tblade4,t,Tblade5,t,Tblade6);

legend('Tblade1','Tblade2','Tblade3','Tblade4','Tblade5','Tblade6')
      title(['Blade Temperatures,  ', datanames],'FontWeight','bold')
      xlabel('Time, s')
      ylabel('Temperature, °C')
   end

   if (sum(graphs==2) )  % ----   Time History
      figure
      set(gcf,'Units','Normalized','Position',[.2,.05,1-.2,.8])
      Xone = {t t t};
      Yone = {Tfree Tblade1 coolant};
      Xtwo = {t};
      Ytwo = {q1};
      s1   = {150 150 150};
      m1   = {'d' 'o' 's'};
      s2   = {10};
      m2   = {'-'};
      [AX,H1,H2,H3,H4]=fancyplot2(Xone,Yone,s1,m1,Xtwo,Ytwo,s2,m2)
      axes(AX(1));
      ylabel('Temperature, °C')
      set(gca,'YLim',[Tblade3(front)-5 Tplen(front)+10])
      hold on;
      bot=get(gca,'Ylim');
      plot([back/100 back/100],[bot(1) bot(2)],'--','color','k')
      plot([front/100 front/100],[bot(1) bot(2)],'--','color','k')
      hold off;
      line([0 40],[bot(2) bot(2)],'color','k')
      axes(AX(2));
      ylabel('q, W/cm²')
      set(H3,'linestyle','-')
      set(H4,'linestyle','-','color',[0 0.65 0])
      set(H1(1),'color','b')
      set(H3(1),'color','b')
      set(H1(3),'color','m')
      set(H3(3),'color','m')
      set(H1(2),'color','r')
      set(H3(2),'color','r')
      set(H2,'color',[0 0.65 0])
      [L1,L2]=legend([H3 H4],'T_{freestream}', 'T_{blade}',
'T_{coolant}', 'q')
      set(L2(1:4,2),'FontSize',.11,'FontWeight','bold')
      set(H3,'linestyle','none')
      set(H4,'linestyle','none')
      Title(['Time History  '],'FontWeight','bold')
      xlabel('Time, s')
      set(gcf,'color',[1 1 1])
   end

   if (sum(graphs==3) )   % ----   h and eta graph

      figure
      set(gcf,'Units','Normalized','Position',[.2,.05,1-.2,.8])
      hold on;
      plot(deltaT1(front:back),qprime1(front:back),'.')
      endpts=[ deltaT1(front) deltaT1(back) ];
      fit=polyval(p1,endpts);
      plot(endpts,fit,'r');
      hold off;
      if (typeflag==1)
         Title(['h and \eta,  ' datanames ' Location
#1'],'FontWeight','bold')
         Xlabel('(T_r - T_{blade})/(T_r - T_{coolant})  [ - ]')
         Ylabel('q/(T_r - T_{coolant})  [ W/cm² °C ]')
         xtext=endpts(2);
         ytext=( qprime1(front) + qprime1(back) )/2;
         text(xtext,ytext,['h= ' num2str(h1) ' W/m² °C    \eta= '
num2str(n1) ])
      elseif(typeflag==4)
         Title(['h and T_{drop},  ' datanames ' Location
#1'],'FontWeight','bold')
         Xlabel('(T_r - T_{blade})  [ °C ]')
         Ylabel('q  [ W/cm² ]')
         xtext=endpts(2);
         ytext=( qprime1(front) + qprime1(back) )/2;
         text(xtext,ytext,['h= ' num2str(h1) ' W/m² °C    T_{drop}= '
num2str(Tdrop1) ])
      end
      figure
      set(gcf,'Units','Normalized','Position',[.2,.05,1-.2,.8])
      hold on;
      plot(deltaT2(front:back),qprime2(front:back),'.')
      endpts=[ deltaT2(front) deltaT2(back) ];
      fit=polyval(p2,endpts);
      plot(endpts,fit,'r');
      hold off;
      if (typeflag==1)
         Title(['h and \eta,  ' datanames ' Location
#2'],'FontWeight','bold')
         Xlabel('(T_r - T_{blade})/(T_r - T_{coolant})  [ - ]')
         Ylabel('q/(T_r - T_{coolant})  [ W/cm² °C ]')
         xtext=endpts(2);
         ytext=( qprime2(front) + qprime2(back) )/2;
         text(xtext,ytext,['h= ' num2str(h2) ' W/m² °C    \eta= '
num2str(n2) ])
      elseif(typeflag==4)
         Title(['h and T_{drop},  ' datanames ' Location
#2'],'FontWeight','bold')
         Xlabel('(T_r - T_{blade})  [ °C ]')
         Ylabel('q  [ W/cm² ]')
         xtext=endpts(2);
         ytext=( qprime2(front) + qprime2(back) )/2;
         text(xtext,ytext,['h= ' num2str(h2) ' W/m² °C    T_{drop}= '
num2str(Tdrop2) ])
      end


      figure
      set(gcf,'Units','Normalized','Position',[.2,.05,1-.2,.8])
      hold on;
      plot(deltaT3(front:back),qprime3(front:back),'.')
      endpts=[ deltaT3(front) deltaT3(back) ];
      fit=polyval(p3,endpts);
      plot(endpts,fit,'r');
      hold off;
      if (typeflag==1)
         Title(['h and \eta,  ' datanames ' Location
#3'],'FontWeight','bold')
         Xlabel('(T_r - T_{blade})/(T_r - T_{coolant})  [ - ]')
         Ylabel('q/(T_r - T_{coolant})  [ W/cm² °C ]')
         xtext=endpts(2);
         ytext=( qprime3(front) + qprime3(back) )/2;
         text(xtext,ytext,['h= ' num2str(h3) ' W/m² °C    \eta= '
num2str(n3) ])
      elseif(typeflag==4)
         Title(['h and T_{drop},  ' datanames ' Location
#3'],'FontWeight','bold')
         Xlabel('(T_r - T_{blade})  [ °C ]')
         Ylabel('q  [ W/cm² ]')
         xtext=endpts(2);
         ytext=( qprime3(front) + qprime3(back) )/2;
         text(xtext,ytext,['h= ' num2str(h3) ' W/m² °C    T_{drop}= '
num2str(Tdrop3) ])
      end
      figure
      set(gcf,'Units','Normalized','Position',[.2,.05,1-.2,.8])
      hold on;
      plot(deltaT4(front:back),qprime4(front:back),'.')
      endpts=[ deltaT4(front) deltaT4(back) ];
      fit=polyval(p4,endpts);
      plot(endpts,fit,'r');
      hold off;
      if (typeflag==1)
         Title(['h and \eta,  ' datanames ' Location
#4'],'FontWeight','bold')
         Xlabel('(T_r - T_{blade})/(T_r - T_{coolant})  [ - ]')
         Ylabel('q/(T_r - T_{coolant})  [ W/cm² °C ]')
         xtext=endpts(2);
         ytext=( qprime4(front) + qprime4(back) )/2;
         text(xtext,ytext,['h= ' num2str(h4) ' W/m² °C    \eta= '
num2str(n4) ])
      elseif(typeflag==4)
         Title(['h and T_{drop},  ' datanames ' Location
#4'],'FontWeight','bold')
         Xlabel('(T_r - T_{blade})  [ °C ]')
         Ylabel('q  [ W/cm² ]')
         xtext=endpts(2);
         ytext=( qprime4(front) + qprime4(back) )/2;
         text(xtext,ytext,['h= ' num2str(h4) ' W/m² °C    T_{drop}= '
num2str(Tdrop4) ])
      end

     % figure
     % set(gcf,'Units','Normalized','Position',[.2,.05,1-.2,.8])
     % hold on;
     % plot(deltaT5(front:back),qprime5(front:back),'.')
     % endpts=[ deltaT5(front) deltaT5(back) ];
     % fit=polyval(p5,endpts);
     % plot(endpts,fit,'r');
     % hold off;
     % if (typeflag==1)
     %    Title(['h and \eta,  ' datanames ' Location
#5'],'FontWeight','bold')
     %    Xlabel('(T_r - T_{blade})/(T_r - T_{coolant})  [ - ]')
     %    Ylabel('q/(T_r - T_{coolant})  [ W/cm² °C ]')
     %    xtext=endpts(2);
     %    ytext=( qprime5(front) + qprime5(back) )/2;
     %    text(xtext,ytext,['h= ' num2str(h5) ' W/m² °C    \eta= '
num2str(n5) ])
     % elseif(typeflag==4)
     %    Title(['h and T_{drop},  ' datanames ' Location
#5'],'FontWeight','bold')
     %    Xlabel('(T_r - T_{blade})  [ °C ]')
     %    Ylabel('q  [ W/cm² ]')
     %    xtext=endpts(2);
     %    ytext=( qprime5(front) + qprime5(back) )/2;
     %   text(xtext,ytext,['h= ' num2str(h5) ' W/m² °C    T_{drop}= '
num2str(Tdrop5) ])
     % end
```

```
      %figure
      %set(gcf,'Units','Normalized','Position',[.2,.05,1-.2,.8])
      %hold on;
      %plot(deltaT6(front:back),qprime6(front:back),'.')
      %endpts=[ deltaT6(front) deltaT6(back) ];
      %fit=polyval(p6,endpts);
      %plot(endpts,fit,'r');
      %hold off;
      %if (typeflag==1)
      %   Title(['h and \eta,  ' datanames ' Location
#6'],'FontWeight','bold')
      %   Xlabel('(T_r - T_{blade})/(T_r - T_{coolant})  [ - ]')
      %   Ylabel('q/(T_r - T_{coolant})  [ W/cm² °C ]')
      %   xtext=endpts(2);
      %   ytext=( qprime6(front) + qprime6(back) )/2;
      %   text(xtext,ytext,['h= ' num2str(h6) ' W/m² °C    \eta= '
num2str(n6) ])
      %elseif(typeflag==4)
      %   Title(['h and T_{drop},  ' datanames ' Location
#6'],'FontWeight','bold')
      %   Xlabel('(T_r - T_{blade})  [ °C ]')
      %   Ylabel('q  [ W/cm² ]')
      %   xtext=endpts(2);
      %   ytext=( qprime6(front) + qprime6(back) )/2;
      %   text(xtext,ytext,['h= ' num2str(h6) ' W/m² °C    T_{drop}= '
num2str(Tdrop6) ])
      %end

   end


   if (sum(graphs==4))      % ----   Heat Flux
      figure
      set(gcf,'Units','Normalized','Position',[.2,.05,1-.2,.8])
      plot(t,q6,t,q5,t,q4,t,q3,t,q2,t,q1)
      legend('q6','q5','q4','q3','q2','q1')
      title(['Heat Flux,  ', datanames],'FontWeight','bold')
      xlabel('Time, s')
      ylabel('Flux, W/m^2-°C')
   end



if (sum(graphs==5))      % ----   Heat Flux vs Time Graphs, Ambient
      figure
      set(gcf,'Units','Normalized','Position',[.2,.05,1-.2,.8])
      htime = 10000*q1./(Taw1 - Tblade1);
      l1    = plot(t,htime);
      l2    = line([t(1) t(length(t))], [h1 h1],'linestyle','--
','color',[0 0.5 0]);
      l3    = legend([l1 l2],'h(t)','h_{avg}');
      set(l3,'FontWeight','bold');
      set(gca,'YLim',[0 1000]);
      set(gca,'XLim',[0 23]);
      title(['Heat Flux vs. Time '],'FontWeight','bold')
      xlabel('Time, s')
      ylabel('Flux, W/m^2-°C')
      text(1,(round(h1)+40),['h_{avg} = ' num2str(round(h1)) ' W/m^2
°C']);
      set(gcf,'color',[1 1 1]);

      figure
      set(gcf,'Units','Normalized','Position',[.2,.05,1-.2,.8])
      Xone  = {t t t};
      Yone  = {Tfree Taw1 Tblade1};
      s1    = {150 150 150};
      m1    = {'o' 'd' 's'};
      Xtwo  = {t};
      Ytwo  = {q1};
      s2    = {150};
      m2    = {'-'};
      [AX,H1,H2,H3,H4] = fancyplot2(Xone,Yone,s1,m1,Xtwo,Ytwo,s2,m2);
      set(H3(1),'color','b','linestyle','-');
      set(H3(2),'color',[0.75 0 0.75],'linestyle','-');
      set(H3(3),'color','r','linestyle','-');
      set(H4(1),'color', [0 0.5 0],'linestyle','-');
      set(H1(1),'color','b','linestyle','-');
      set(H1(2),'color',[0.75 0 0.75],'linestyle','-');
      set(H1(3),'color','r','linestyle','-');
      set(H2(1),'color', [0 0.5 0],'linestyle','-');
      title(['Temperature Time History '],'FontWeight','bold')
      xlabel('Time, s')
      axes(AX(1));
      set(gca,'YLim',[-10 30]);
      ylabel('Temperature, °C');
      axes(AX(2));
      set(gca,'YLim',[-1 3]);
      ylabel('Flux, W/m^2-°C');
      legend([H3 H4],'Tfree', 'Taw', 'Twall', 'q');
      set(H3,'linestyle','none');
      set(H4,'linestyle','none');
      r1 = get(AX(2),'YLim');
      r2 = get(AX(2),'XLim');
      %line([23 23], [r1(1) r1(2)],'linestyle','--');
      %line([r2(1) r2(2)], [0 0],'linestyle','--');
      set(gcf,'color',[1 1 1]);
      box on;
   end

nofpts
```

Appendix C: HFM Coefficients and Sensitivities

Gauge #	Serial #	a	b	c	d	e	f	S_{HFS}
1	387	0	0	3.93326	-495.849	0.261826	125.7314	234
2	388	0	0	2.57488	-518.276	0.39751	200.8981	105
3	389	0	0	3.9724	-480.423	0.255602	120.7617	145
4	391	0	0	3.89763	-488.813	0.261826	125.1414	130
5	397	0	0	2.4478	-477.73	0.415353	194.8265	171
6	390	0	0	4.13174	-476.724	0.243154	115.3224	153

* These gauges are all Vatell type HFM-7's.

Appendix D: Ambient Coolant Experiments

Ambient Coolant Analysis

This study was performed in order to better understand the heat transfer occurring at the surface of turbine blades. In particular, the dependence of the heat transfer coefficient upon temperature differences in the flowfield was investigated. The heat transfer coefficient is generally held to be a function of the flowfield only, but steep temperature gradients affect the fluid properties of the flowfield and have some effect on the heat transfer coefficient. By comparing the cold film-cooled experiments, which have steep temperature gradients in the film layer, to ambient temperature film-cooled without steep temperature gradients, some conclusions may be made about the behavior of the heat transfer coefficient. Equation (14) is again used to describe heat transfer from a surface in the presence of high-speed flows with fluid injection into the boundary layer,

$$\mathrm{q} = \mathrm{h} \cdot \left(\mathrm{T}_{\mathrm{aw}} - \mathrm{T}_{\mathrm{w}}\right) \qquad \text{(D-1)}$$

where T_{aw} is the driving force of the heat transfer. For the experiments with ambient temperature film injection, T_{aw} is assumed to be equivalent to the recovery temperature. This assumption is made because the although the boundary layer is disturbed, the same driving temperatures are present as in the uncooled case. The heat transfer equation used in the analysis of the ambient coolant data is Equation (D-2),

$$q = h \cdot \left(T_{\infty} - T_{w}\right) - h \cdot \left(\mathrm{T}_{0,\infty} - \mathrm{T}_{r}\right) \qquad \text{(D-2)}$$

The values of $T_{o,\infty}$ - T_r along the blade and the local heat flux and wall temperature as well as the freestream total temperature are either known or measured during the run; this can be seen in Figure D-1. The dashed lines in the figure represent data which were used in the ambient coolant analysis. The interval was chosen to include all data between the points where the freestream temperature and the mass averaged coolant temperature were within two degrees of each other. Typically this amounted to about ten seconds of data taken at 100Hz or about 1000 data points. Some runs were taken with as much as 1500 points or as few as 700.

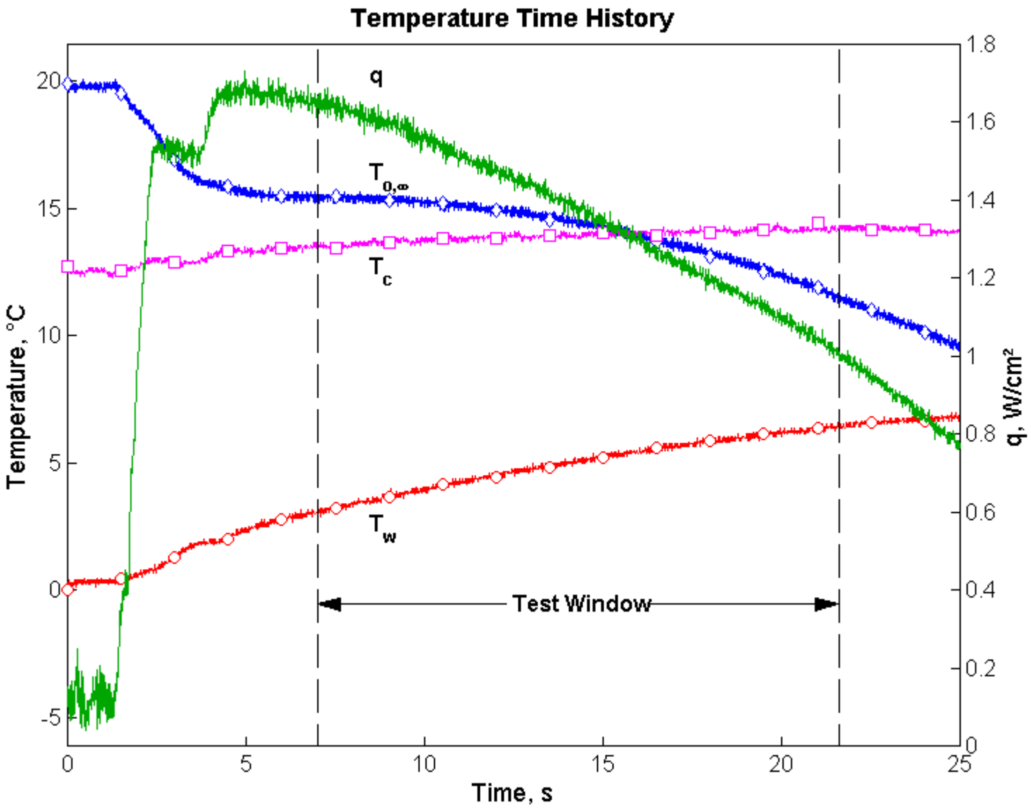


Figure D-1: Sample Time History of a Film-cooled Run (ambient coolant)

When the data from the time history of an ambient coolant run such as that shown in Figure D-1 is plotted according to Equation (D-2), the result is a very linear curve in which the local heat flux is linearly dependent upon the difference between the freestream total temperature and the local wall temperature. Theoretically, the film cooling effectiveness for this case should be zero. In actuality what is encountered is scatter around zero due to calculation errors, therefore the film effectiveness was not pursued for these experiments. An example of the data plotted in this manner can be seen in Figure D-2.

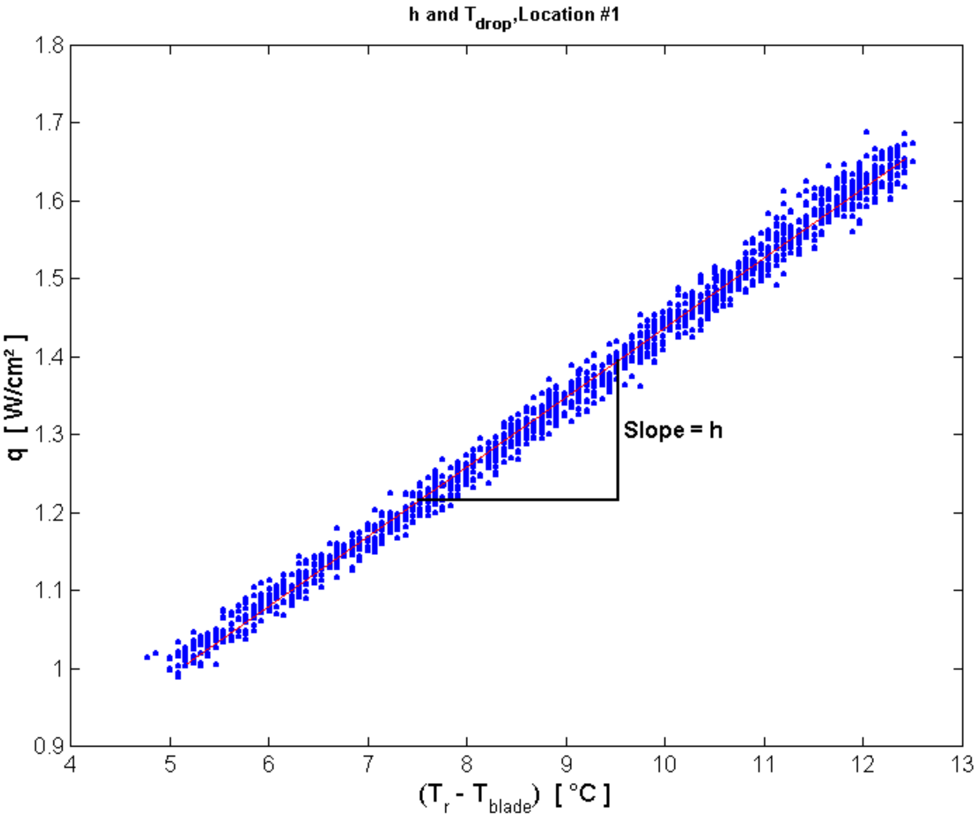


Figure D-2: Data Analysis of an Ambient Coolant Run

After determining the average heat transfer coefficient over the run, the time history can be plotted as seen in Figure D-3. As in the two earlier cases, the heat transfer coefficient is relatively constant over the run after the initial rise caused by the starting of the wind tunnel. It can also be seen in the figure that the average value of the heat transfer function over the run lies on top of the time history curve after the initial rise. This is a good consistency check for the ambient analysis.

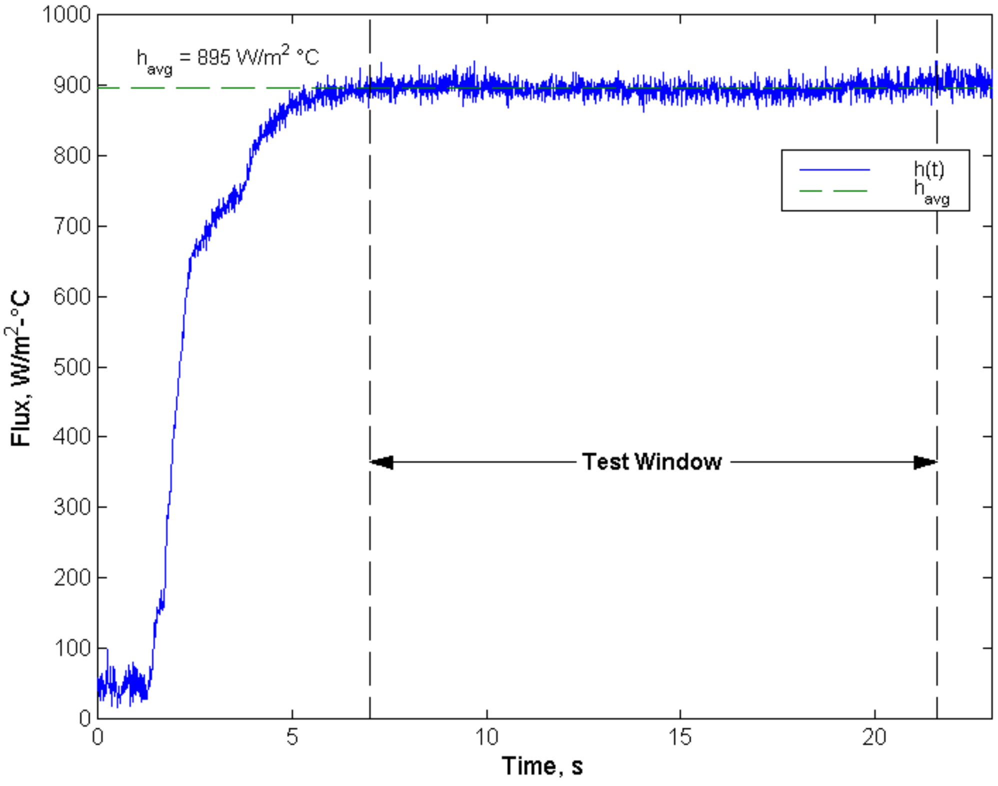


Figure D-3: Time History of Heat Transfer Coefficient (ambient coolant)

Ambient Coolant Heat Transfer Experiments

Objectives

The objective of these experiments is to determine the extent to which the heat transfer coefficient depends upon temperature gradients. It has been assumed in this work that the heat transfer coefficient is mainly a function of the flow field, and that it has only a weak dependence upon temperature. For the two cases already discussed, flow over a turbine blade with and without film cooling, the flow is heated to 100°C and the blade is cooled to 0°C, so there are temperature gradients in the boundary layer. In the film-cooled case, the gradients are increased due to the injection of coolant into the boundary layer. The transient nature of these experiments causes the temperature gradients to change over the course of a run. The gradients are largest at the beginning of a run and smaller near the end, due to the convergence of the blade wall and freestream temperatures over the course of the run. For the gradient in the uncooled case, the temperature ratio of interest is the freestream to blade wall ratio which ranges from 1.4 to 1.0 over the run. The temperature gradient in the film-cooled case is caused mainly by the freestream to coolant temperature ratio, which changes from 2.4 to 1.6 over the course of the run.

It has been shown in Figures 18 and 22 that the heat transfer coefficient remained almost constant over the run despite the initial temperature gradients and the changes in them as the run progressed. This can also be illustrated in Figures 17 and 21, which show the graphical representation of the data analyses for the two cases. In both of these figures the slope of the data represents the heat transfer coefficient, and in both cases the slope is very constant. What these two sets of figures suggest is that the heat transfer coefficient is not affected by the ranges of temperature gradients which exist over the course of an uncooled run or a run in which a cold coolant is injected into the boundary layer, but rather that it is affected by the change in the flow field due to the disturbance induced by the coolant injection.

The next step is to determine if the magnitude of the heat transfer coefficient, in the case of fluid injection into the boundary layer, is affected by the absence of a temperature difference between the coolant and the freestream. The previous two experiments act as limiting cases: 1) the uncooled case showed the independence of heat transfer coefficient over a range of temperature gradients with no boundary layer disturbance, and 2) the film-cooled case with a cold injectant shows the independence of the heat transfer coefficient to ranges of temperature gradients with a film disturbance. The next experiment, a film-cooled turbine blade with ambient coolant and freestream temperatures, investigates what happens to the heat transfer coefficient when there is a disturbance to the boundary layer and the freestream to coolant temperature ratio is approximately one.

Discussion

Figure D-1 shows a sample of the transient time histories for the local heat flux at one location as well as the temperatures of the mainstream flow, the blade wall, and the coolant. The dashed lines in the figure represent the data which was used in the ambient coolant analysis described earlier. A total of four runs were made with film cooling where the injectant was close to the ambient temperature, however, due to data acquisition limitations at the time these experiments were performed not every gauge was recorded for each run. Table D-1 shows the test matrix that was used to take data for this case. This matrix was chosen so that at least two data points were taken for each gauge and three on the first four.

Table D-1: Test Matrix for Ambient Coolant Experiments

Gauge #	Run 1	Run 2	Run 3	Run 4

1	o	*	+	
2	o	*	+	
3	o	*		•
4	o	*		•
5			+	•
6			+	•

The data gathered using this test matrix were all analyzed according to the ambient coolant analysis and the results can be seen in Figure D-4. The repeatability of the results for this case is very inconsistent. The reasons behind this inconsistency are: 1) there were not many tests performed, 2) the small temperature differences used in the analysis generate calculation errors, and 3) the uncertainties in measuring temperature. At least twice as many tests were performed on each gauge in the previous two cases as were performed in this one due to time and data acquisition constraints. The small temperature differences become significant when a division operation needs to be done, and the uncertainty in our temperature measurements can make this worse.

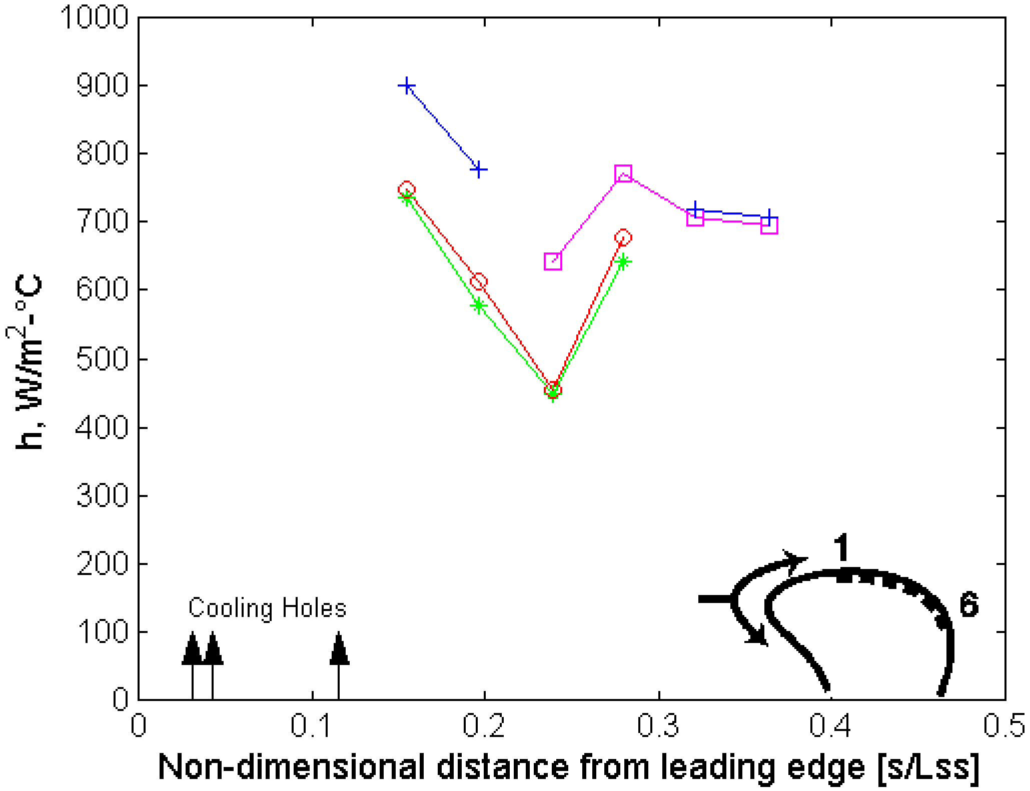


Figure D-4: Ambient Coolant Heat Transfer Coefficient Along Blade

The general trend however is similar to the results of both the uncooled and the cold coolant experiments. Again, the first three gauges show a decreasing trend which reverses itself beginning with the fourth gauge. For the ambient coolant case, however, the fifth and six gauges show a slight decrease in heat transfer coefficient. It should be noted that there are only two data points for each of these gauge locations, and this decrease could be just scatter in the data. The reason for these trends are the same as those hypothesized earlier.

In Figure D-5, a comparison of the mean values of the heat transfer coefficient for the cold and the ambient coolant cases can be seen. The three arrows represent the location of coolant hole exits and the drawing in the bottom right of the figure shows the relative gauge locations on the blade. With the exception of the last two gauges, as noted earlier, the trends in both cases are the same. It is encouraging that the two cases generally have the same trend, however, there is a difference in magnitude between the two. The magnitude of the ambient coolant case is in all cases lower than that of the cold coolant case. The actual values of the mean heat transfer coefficient as well as the percent decrease in heat transfer coefficient from the cold coolant case to the ambient coolant is shown in Table D-2.

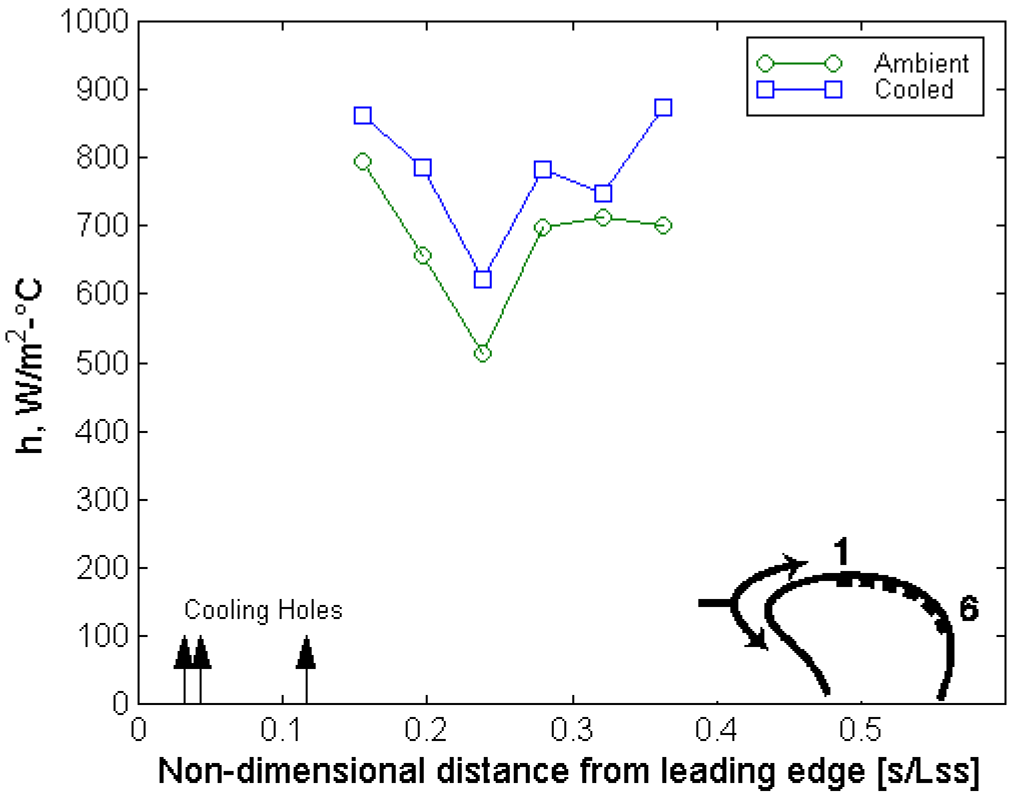


Figure D-5: Comparison of Heat Transfer Coefficient (cold, ambient)

From Table D-2 it can be seen that the percent differences in heat transfer coefficient are of the same order of magnitude as those found in Table 2 comparing the uncooled case to the cold coolant case. Therefore, the differences in heat transfer coefficient observed between the ambient and cold coolant cases must be considered significant. This result is somewhat unexpected.

Table D-2: Comparison of Ambient and Cold-Film Cases

Gauge Location	**Cold Coolant Heat Transfer Coefficient ($W/m^2 \cdot °C$)**	**Ambient Coolant Heat Transfer Coefficient ($W/m^2 \cdot °C$)**	**Percent Decrease %**
1	864	795	8
2	785	655	16
3	622	514	17
4	783	697	11
5	747	712	5
6	875	702	20

The assumption was made in each of the earlier analyses that the heat transfer coefficient had a weak dependence upon temperature. This assumption was in agreement with a part of the results from all cases, where it can be shown that the magnitude of the heat transfer coefficient does not vary over the course of a run. The differences in the two cases do not prove conclusively that there is a dependence upon temperature in the heat transfer coefficient, however. The variations in heat transfer coefficient for the cold and ambient coolant cases could have been caused by scatter in the data and/or the influence of the change in the blowing parameters between the two cases. Previous research has shown the effects of the blowing, momentum, and density ratios on the heat transfer coefficient [reference research here???]. The strongest factor was found to be the momentum ratio, and it was this ratio that was constrained for the ambient coolant experiments. It was stated that wide variations in density ratio had little effect, but none of the research investigated a ratio near unity, which is the value for the density ratio that was attempted in the ambient case. It could be that the density ratio plays a bigger role when it is nearer to unity.

The third set of experiments, the ambient mainstream with ambient coolant tests, were performed to check the assumption that the heat transfer coefficient is affected most by the flow field and has only a weak dependence upon temperature. Attempts were made to generate the same flow field as in the cold coolant experiments, but without the coolant to mainstream temperature difference. A comparison of the results of this experiment with those of the cold coolant experiments show that there is a decrease in heat transfer coefficient associated with the ambient coolant tests that is of the same order of magnitude as that between the experiments with and without cooling. It should be noted however, that the blowing parameters in the ambient coolant tests were not the same as those of the cold coolant tests. Only the momentum ratio was capable of being controlled in these experiments; the blowing and density ratios were different for the ambient coolant case, and this may have been the cause of the magnitude of the difference.

Vita

Dwight E. Smith was born on March 10, 1974, to two highschool teachers in Kingstree, South Carolina. His family includes a brother, two sisters, one niece, and numerous cousins, aunts, and uncles. Dwight spent two years at the SC Governor’s School for Science and Mathematics before graduating and attending Clemson University where he obtained his BS in Mechanical Engineering.