1991

Simulation of an unsteady thin oil film

Gary L. Stevens
San Jose State University

Follow this and additional works at: https://scholarworks.sjsu.edu/etd_theses

Recommended Citation

Stevens, Gary L., "Simulation of an unsteady thin oil film" (1991). *Master's Theses*. 167.
DOI: <https://doi.org/10.31979/etd.3nrc-7sj9>
https://scholarworks.sjsu.edu/etd_theses/167

This Thesis is brought to you for free and open access by the Master's Theses and Graduate Research at SJSU ScholarWorks. It has been accepted for inclusion in Master's Theses by an authorized administrator of SJSU ScholarWorks. For more information, please contact scholarworks@sjsu.edu.

INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

University Microfilms International
A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
313/761-4700 800/521-0600

Order Number 1344313

Simulation of an unsteady thin oil film

Stevens, Gary Lance, M.S.

San Jose State University, 1991

U·M·I
300 N. Zeeb Rd.
Ann Arbor, MI 48106

SIMULATION OF AN UNSTEADY THIN OIL FILM

A Thesis

Presented to

**The Faculty of the Department of Mechanical Engineering
San Jose State University**

**In Partial Fulfillment
of the Requirements for the Degree
Master Of Science**

By

Gary L. Stevens

May, 1991

**APPROVED FOR THE DEPARTMENT OF
MECHANICAL ENGINEERING**

Dr. Russ Westphal, Associate Professor

Dr. Mike Green, Faculty Advisor

Dr. Robert Clothier, Graduate Coordinator

Dr. Tai-Ran Hsu, Department Chair

APPROVED FOR THE UNIVERSITY

ABSTRACT

SIMULATION OF AN UNSTEADY THIN OIL FILM

by Gary L. Stevens

Oil film skin friction interferometry has been employed by researchers since the mid-1970s to measure oil thinning rate and determine the **time-averaged** skin friction in a wide variety of fluid flows. However, to-date nobody has attempted **unsteady** skin friction measurements.

This thesis provides a simulation of unsteady thin oil film behavior using a numerical approach. The intent is to create an analytical tool for use in investigating unsteady skin friction measurements.

To accomplish the simulation, an explicit, finite difference computer code was written to solve for the temporal and spatial variation in oil film thickness subject to time and spatially varying skin friction. Two test cases which had analytical solutions were run to validate the program. Then, a sample calculation was run which used unsteady variation of the boundary conditions. The results were reviewed for reasonableness, and conclusions were drawn regarding the ability of the code to properly simulate unsteady thin oil film behavior.

TABLE OF CONTENTS

I.	INTRODUCTION	1
	Background	1
	Thin Oil Film Behavior	5
	Objective of This Study	8
II.	MODEL DEVELOPMENT	9
	Equation Classification	9
	Equation Simplification	10
	Numerical Solution Approach	12
	Description of Computer Code	15
III.	RESULTS AND DISCUSSION	19
	Test Case #1 [$\tau^* = \text{constant}$]	22
	Test Case #2 [$\partial \tau^* / \partial x^* = \text{constant}$]	27
	Sample Calculation [$\tau^*(x,t)$]	33
IV.	SUMMARY AND CONCLUSIONS	43
V.	REFERENCES	46

APPENDICES

A.	EQUATION DERIVATION	50
B.	PROGRAM "THINFILM" LISTING	56
C.	PROGRAM "FINDTIME" LISTING	65
D.	PROGRAM "ANIMATE" LISTING	68
E.	ANALYTICAL SOLUTIONS TO TEST CASES #1 AND #2	71
F.	PROGRAM "TAUGEN" LISTING	76
G.	PROGRAM "SLOPE" LISTING	78

LIST OF ILLUSTRATIONS

1.	Skin Friction Measurement Techniques	2
2.	Thin Oil Film Skin Friction Measurement Technique	4
3.	Fundamental Equations	6
4.	Error for Test Case #1 (Time Step = $1.0\Delta x^*/h^*$, N = 201)	25
5.	Error for Test Case #1 (Time Step = $0.5\Delta x^*/h^*$, N = 201)	25
6.	Error for Test Case #1 (Time Step = $0.25\Delta x^*/h^*$, N = 201)	26
7.	Error for Test Case #1 (Time Step = $1.0\Delta x^*/h^*$, N = 101)	26
8.	Initial h^* Profile for Test Case #2	29
9.	Error for Test Case #2 (Time Step = $1.0\Delta x^*/h^*$, N = 201)	31
10.	Error for Test Case #2 (Time Step = $0.5\Delta x^*/h^*$, N = 201)	31
11.	Error for Test Case #2 (Time Step = $0.25\Delta x^*/h^*$, N = 201)	32
12.	Error for Test Case #2 (Time Step = $0.5\Delta x^*/h^*$, N = 101)	32
13.	Shear Stress Data From Digital Simulation . .	33
14.	Output Oil Thickness Profiles for Sample Calculation ($t^* = 0, 100, 200, 300, 400$) . . .	38

LIST OF ILLUSTRATIONS (continued)

15.	Output Oil Thickness Profiles for Sample	
	Calculation ($t^* = 500, 600, 700, 800, 900$) . .	38
16.	Output Oil Thickness Profiles for Sample	
	Calculation ($t^* = 1000, 1100, 1200, 1300, 1333$)	39
17.	Comparison of Slope Change Near Leading	
	Edge of Oil Film	41
18.	Comparison of Slope Change Near Leading'	
	Edge of Oil Film ($1 < t^* < 2$)	41

LIST OF TABLES

1.	Test Cases Evaluated With Program "THINFILM"	19
2.	Nominal Input Parameters	21
3.	Dimensionless Inputs for Test Case #1	
	$[\tau^* = \text{constant}]$	23
4.	Stability Results for Test Case #1	24
5.	Dimensionless Inputs for Test Case #2	
	$[\partial \tau^* / \partial x^* = \text{constant}]$	28
6.	Stability Results for Test Case #2	30
7.	Dimensionless Inputs for Sample Calculation	
	$[\tau^*(x, t)]$	37
8.	Comparison of Results	
	Sample Calculation vs. Test Case #1	
	(Predicted h^* at $x^* = 1.0$)	40

NOMENCLATURE

A	MacCormack method stability parameter, $\partial F/\partial h^*$
B, D	Generalized partial differential equation coefficients (functions of h , x and t)
F	MacCormack method parameter, $\tau^* h^{*2}/2$
g_x	stream-wise gravitational constant
g_y	vertical gravitational constant
h	oil film thickness
h^*	dimensionless oil film thickness, h/L
L	length of oil film
N	number of spatial grid points
p	pressure
p_o	time-averaged pressure
p'	pressure fluctuation: $p = p_o + p'$
p^*	dimensionless pressure, p/τ_o
t	time
t^*	dimensionless time, $t\tau_o/\mu$
Δt	time step
Δt^*	dimensionless time step, $\Delta t\tau_o/\mu$
u	stream-wise oil velocity component
u_F	stream-wise oil velocity at the test surface interface
v	vertical oil velocity component
v_F	vertical oil velocity at the test surface interface

NOMENCLATURE (continued)

V	shear stress convective speed
x	stream-wise coordinate distance along test surface
x^*	dimensionless distance along test surface, x/L
Δx	spatial grid spacing
Δx^*	dimensionless spatial grid spacing, $\Delta x/L$
y	vertical coordinate distance above test surface
μ	oil dynamic viscosity
τ	shear stress
τ_o	mean shear stress
τ'	shear stress fluctuation: $\tau = \tau_o + \tau'$
τ^*	dimensionless shear stress, τ/τ_o
τ_L^*	dimensionless shear stress at last spatial grid point ($x^* = 1$)
τ_o^*	dimensionless shear stress at first spatial grid point ($x^* = 0$)
ν	oil kinematic viscosity

I. INTRODUCTION

Background

Over the years, researchers from various disciplines have been concerned with the measurement of skin friction. The reasons for conducting such measurements differ depending upon the application. For example, an aerodynamicist may be interested in the drag created by skin friction when air flows over an airfoil. Regardless of the differences in the purpose of the measurements, the underlying concept is the same for most applications. The pursuit of improved and more efficient designs has created an overwhelming desire and need for accurate measurements of skin friction. Because of this demand, several techniques have been developed over the years in an attempt to achieve this goal.

The techniques developed for measuring skin friction can be generally grouped into two categories: indirect and direct. Many of the techniques used in each of these categories are shown in Figure 1. Extensive reviews of most of these methods were documented by Winter [1] and Bertelrud [2]. A review of recent advances in compressible skin friction measurement techniques was provided by Settles [3].

Indirect methods measure parameters indirectly related to or caused by skin friction, and then infer the skin friction from the results. The wall wire technique is an

Figure 1. Skin Friction Measurement Techniques

example of an indirect method; the heat flux is measured from a wire embedded in a surface exposed to a fluid flow. Through the Reynolds analogy, the heat flux can be related to the skin friction of the fluid moving over the surface [1-3]. For this case, the quantity being measured is heat transfer, not skin friction, so the latter must be obtained indirectly from the measured quantity. A chief difficulty with this method is the breakdown of the heat transfer analogy for complex flows [3]. Similar problems are common to many of the indirect methods of skin friction measurement.

Direct methods utilize measured quantities which are unrelated to the constitutive properties of the flowing

fluid. Primary measured quantities for these methods typically consist of force, displacement, viscosity, etc. The floating element measurement technique is an example of a direct method; the restoring force on a floating element installed on the test surface is measured. This method has been noted for its difficulty, especially with rising Mach number [3].

A direct method which has received attention because of its relative ease of implementation and accuracy is the oil film method. An oil film on a flat, smooth surface will spread and thin when a fluid flows over it. The thinning is caused primarily by the shear stress (skin friction) which the fluid exerts on the oil. Thus, if the change in thickness of the oil film can be accurately measured, the skin friction causing the film thinning can be determined. The measurement apparatus can often be designed such that there are no other significant competing parameters which cause the oil film to thin. This minimizes uncertainties in the skin friction assessment; thus, the accuracy of the skin friction determination with the oil film technique is limited only by the ability to measure the change in thickness of the oil film accurately.

The technique most often used to measure the thinning of oil films is called laser interferometry, and is schematically shown in Figure 2. A laser and photodetector arrangement is used to accurately measure the average

Figure 2. Thin Oil Film Skin Friction Measurement Technique

rate-of-change in thickness of an oil film subjected to an imposed shear force caused by air flowing over the surface in question. The laser beam is directed at the test surface; the portion of the incident beam which is reflected from the surface of the oil film interferes with the portion which reflects from the test surface. The interference signal is captured by a photodetector; as the oil film thins, a corresponding change in relative interference is measured. With the incident light wavelength known, the oil thickness change is known, and as a result, so is the skin friction.

The interferometry technique of skin friction measurement was originally proposed by Tanner and Blows [4].

A review of this method applied to compressible flows was provided by Settles [3]. Since the mid-1970s, laser interferometry has been employed by numerous researchers to measure the oil thinning rate and thereby determine the **time-averaged** skin friction in a wide variety of fluid flows. Several of these previous studies are documented in the literature [5-10].

Numerical analysis of the interferometry technique has been studied by several researchers in an effort to establish some limits of applicability for the theoretical approach used by this method. Lubrication theory is used to relate the rate of thinning to the applied shear stress at the oil-air interface. One of the most recent numerical treatments was provided by Murphy and Westphal [9], where the authors suggested that the oil film method would in principle allow for **unsteady** skin friction measurements as well as time-averaged measurements. That study provides the underlying basis from which the current study advances.

Thin Oil Film Behavior

As described above, an oil film on a flat, smooth surface will spread and thin when a fluid flows over it. The motion of such an oil film is determined by three forces: pressure gradient produced by the fluid flow, gravitational force and skin friction. The current study limits itself to flat test surfaces; as a result, the

gravity term is eliminated from further consideration.

As with all fluid flow problems, the behavior of the thinning oil film can be fully described by the fundamental equations governing fluid flow behavior. These equations are shown in Figure 3. For the current study, constant fluid properties are assumed. Additionally, it was elected to treat the imposed boundary conditions as a function of time and the stream-wise coordinate direction only. As a result, the problem becomes two-dimensional.

CONTINUITY:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0$$

MOMENTUM:

$$\begin{aligned} \frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} &= g_x - \frac{1}{\rho} \frac{\partial p}{\partial x} + \nu \left[\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right] \\ \frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} &= g_y - \frac{1}{\rho} \frac{\partial p}{\partial y} + \nu \left[\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right] \end{aligned}$$

inertia terms

pressure &
gravity terms

viscous stress
terms

Figure 3. Fundamental Equations

For thin (e.g., $h \ll L$) oil films on a flat surface, use of creeping flow approximations (lubrication theory) allows

the inertial terms in the governing equations to be neglected when compared to the viscous stress terms. Further simplification can be realized after order-of-magnitude analysis of the viscous stress terms.

With the incorporation of the simplifications described above into the governing equations, Murphy and Westphal [9] developed a new equation of motion for a thin oil film:

$$\frac{\partial h}{\partial t} = \frac{\tau h}{\mu} \left[\left[\frac{h}{\tau} \frac{dp}{dx} - 1 \right] \frac{\partial h}{\partial x} + \frac{h^2}{3\tau} \frac{d^2 p}{dx^2} - \frac{h}{2\tau} \frac{\partial \tau}{\partial x} \right] \quad (1)$$

The solution to equation (1) subject to prescribed pressure and shear stress boundary conditions provides the thickness of an oil film as a function of time and distance along a flat test surface, $h(x,t)$. Simplified forms of equation (1) have received treatment in more recent literature. Sherman [11] provides investigation of equation (1) for the case of zero pressure gradient¹.

A particular solution to equation (1) results if only time-averaged (e.g., τ is a function of x only) skin friction measurements are considered. For this case, the measured skin friction is inversely proportional to the

¹A slight discrepancy between equation (1) and the work given in [11] exists. The author of the textbook was consulted and it was verified that these discrepancies are the result of typographical errors contained in the [11] study.

product of oil film thickness and time [4]. Thus, if the pressure gradient is known and the thickness variation can be accurately measured, the average skin friction exerted on the oil can be determined. Because the oil film is extremely thin and possesses a low Reynolds number, the oil film skin friction is essentially equal to the test surface skin friction, which is the quantity desired.

Despite the potential of equation (1) to solve for unsteady (e.g., τ is a function of both x and t) skin friction [9], all of the past work with oil film interferometry has been confined to measuring time-averaged shear stress, $\tau(x)$. To-date, no one has attempted unsteady skin friction measurements.

Objective of This Study

The objective of the present study is to apply further analysis and numerical techniques to solve equation (1) using unsteady variations in the shear stress boundary condition. The resulting solution will provide a means of simulating the characteristics of the laser interferometry system used for measuring the time-varying oil film thickness. With such a simulation, applications of interferometry to oil flow beneath a turbulent boundary layer can be explored.

II. MODEL DEVELOPMENT

Equation Classification

A complete derivation of equation (1) is shown in Appendix A. Equation (1) describes the variation of oil film thickness as a function of time and distance along the test surface, subject to an imposed pressure gradient and shear stress (skin friction). Classification of this equation is important in order to apply the most appropriate numerical technique to approximate the solution to the partial differential equation.

Equation (1) can be generalized into the following form:

$$\frac{\partial h}{\partial t} + B \frac{\partial h}{\partial x} = D \quad (2)$$

where B and D are functions of h, x and t. Equation (2) has the form of the inviscid Burger's equation, a well known, thoroughly researched model equation of applied mathematics that serves as a non-linear analog of fluid mechanics equations. This partial differential equation has frequently been used as a prototype for convective phenomena by researchers [12-16], and has the form of the wave equation. This equation is classified as a first-order, hyperbolic, quasi-linear (because B is a function of h, x and t), non-homogeneous (for the case where D is not equal to zero) partial differential equation.

Equation Simplification

A significant observation was made by casting equation (1) into dimensionless form using the following change of variables:

$$\begin{aligned}x^* &= x/L & h^* &= h/L & \tau^* &= \tau/\tau_0 \\p^* &= p/\tau_0 & t^* &= t\tau_0/\mu\end{aligned}$$

Substituting these variables into equation (1)' results in the following simplified equation:

$$\frac{\partial h^*}{\partial t^*} = - \frac{\partial}{\partial x^*} \left[\frac{\tau^* h^{*2}}{2} - \frac{h^{*3}}{3} \frac{\partial p^*}{\partial x^*} \right] \quad (3)$$

Equation (3) takes the form of the inviscid Burger's equation, with both a time-dependent ($\partial/\partial t^*$) term and a convective ($\partial/\partial x^*$) term. As a result, unsteady variations in the boundary conditions are propagated spatially as well as temporally throughout the solution. This behavior makes the equation inherently non-linear in nature.

For unsteadily varying boundary conditions, the pressure and shear stress are composed of a mean component and a fluctuating component. By definition, the variation of the mean components with time is zero. However, the variation of the fluctuating components with time is not zero, and therefore becomes important for unsteady measurement considerations. These variations can be accounted for by

defining the pressure and shear stress terms as follows:

$$p^*(x, t) = \frac{p_o + p'}{\tau_o} = p_o^*(x^*) + p'^*(x^*, t^*) \quad (4)$$

$$\tau^*(x, t) = \frac{\tau_o + \tau'}{\tau_o} = 1 + \tau'^*(x^*, t^*) \quad (5)$$

For the case of a constant, mean pressure gradient, (i.e., $p_o^* = \text{constant}$), $\partial p_o^* / \partial x^* = 0$. Thus, differentiation of equation (4) with respect to x^* yields:

$$\frac{\partial p^*}{\partial x^*} = \frac{\partial p_o^*}{\partial x^*} + \frac{\partial p'^*}{\partial x^*} = \frac{\partial p'^*}{\partial x^*} \quad (6)$$

Substituting equations (5) and (6) into equation (3) yields:

$$\frac{\partial h^*}{\partial t^*} = - \frac{\partial}{\partial x^*} \left[\frac{h^{*2}}{2} (1 + \tau'^*) - \frac{h^{*3}}{3} \frac{\partial p'^*}{\partial x^*} \right] \quad (7)$$

The definition of fluctuating velocity and the Reynolds analogy give rise to the fact that the shear stress ($1 + \tau'^*$) and pressure ($\partial p'^* / \partial x^*$) terms within the square brackets of equation (7) are of the same order of magnitude [17]. Thus, for small h^* , the pressure gradient (second) term inside the square brackets can be neglected compared to the convective shear (first) term. Thus, the oil film equation without mean pressure gradient takes the following simplified, final form:

$$\frac{\partial h^*}{\partial t^*} = - \frac{\partial}{\partial x^*} \left[\frac{\tau^* h^{*2}}{2} \right] \quad (8)$$

Equation (8) is the simplest equation which still has a convective nature. This simplification significantly clarified the direction of research for an appropriate numerical method technique which could be used to approximate the solution to the differential equation, and further demonstrated that the unsteady pressure' term could be neglected.

Numerical Solution Approach

With the applicable equation classified and simplified into the form of equation (8), an appropriate numerical technique for solution was selected. Finite difference methods were chosen for their simplicity and ease of programming.

After investigating several different finite difference techniques [18], the MacCormack method was chosen for application to equation (8). This method is a modified version of the Lax-Wendroff scheme [18], which was one of the first second-order, finite difference methods developed for approximating hyperbolic partial differential equations. The MacCormack method was chosen based on several factors:

- (1) The method is explicit, and therefore would not

require excessive computer memory or execution time.

- (2) The method is easy to program.
- (3) The method has been noted for its particular usefulness in solving non-linear equations [18].
- (4) The method is a predictor-corrector type, and as a general rule is very stable.
- (5) The method has been successfully applied to Burger's equation, which is the form of the equation at hand.

When applied to the inviscid Burger's equation (8), the MacCormack method takes the following form [18]:

Predictor:

$$h_j^{* \overline{n+1}} = h_j^{* n} - \frac{\Delta t^*}{\Delta x^*} (F_{j+1}^n - F_j^n) \quad (9)$$

Corrector:

$$h_j^{* n+1} = \frac{1}{2} \left[h_j^{* n} + h_j^{* \overline{n+1}} - \frac{\Delta t^*}{\Delta x^*} (F_j^{\overline{n+1}} - F_{j-1}^{\overline{n+1}}) \right] \quad (10)$$

where Δt is the time step, Δx is the grid spacing, the superscripts refer to the time step under consideration (e.g., $n = 0, 1, 2, 3, \dots$), and the subscripts refer to the spatial grid point under consideration ($j = 1, 2, 3, \dots, N$). The bars over the superscripts refer to intermediate values associated with the predictor step. Thus, the predictor step uses only those values solved for during the previous

time step. To determine the h^* value for the current time step, the corrector step uses intermediate (predictor) values and increments the previously computed value of h^* . The parameter, F , represents the convective argument contained in Burger's equation, and is defined from equation (8) as follows [18]:

$$F = \frac{\tau^* h^{*2}}{2}, \quad (11)$$

As with most explicit, finite difference schemes, the ability of the MacCormack method to produce a stable solution is determined by appropriately specifying a time step, Δt [18-22]. A stable solution refers to the case where truncation errors do not accumulate in a manner such that the approximation behaves erratically, often causing the computation to "blow up." Thus, a small enough time step must be used which suppresses continued, unbounded growth of the truncation error. Specifying a conservatively small time step may also introduce errors [18], so care must be taken not to "over specify" the time step.

The time step required to achieve a stable solution for the MacCormack method as applied to the inviscid Burger's equation is given by [18]:

$$\Delta t^* \leq \frac{\Delta x^*}{|A|} \quad (12)$$

where the parameter, A , is defined as $\partial F / \partial h^*$. Note that there are no requirements for uniform grid spacing with the MacCormack method; as a result, the time step can vary according to equation (12) during the approximation to satisfy unequal grid spacing.

From the definitions of the parameters A and F provided above, it is evident that different time steps will result from different shear stress boundary conditions. As an example, consider two cases: (1) constant shear stress, and (2) unsteady shear stress. For the case of constant shear stress, it is seen from equation (11) that the parameter, A , produces only one term involving $\partial h^* / \partial x^*$ (e.g., $\partial \tau^* / \partial x^*$ is zero). On the other hand, for the case of unsteady shear stress, application of the chain rule to equation (11) using the definition of A will yield an additional term involving $\partial \tau^* / \partial x^*$. Activating this additional term by specifying a fluctuating component of shear stress has a drastic effect on the time step needed to achieve a stable solution.

Equations (9) through (12) are the formulas necessary for applying the MacCormack method to the approximation of the thin oil film partial differential equation described by equation (8).

Description of Computer Code

A FORTRAN computer program was written which applied the MacCormack scheme given by equations (9) through (12) to

equation (8). The programming was done on a 386-SX, sixteen megahertz personal computer (PC) using Version 5.0 of the Microsoft FORTRAN Compiler [23]. A listing of the program, entitled "THINFILM," is provided in Appendix B.

Program inputs are obtained from two sources: interactively during the initial part of program execution, and from two user-specified input files. The interactive inputs include the input file names, the output file name and the maximum t^* for solution. The first input file, called the oil thickness (h) file, contains the number of grid points, N , the test surface length, L , the oil viscosity, μ and an initial h^* value for each grid point (initial condition). The second input file, called the shear stress (TAU) file, contains the number of shear stress boundary condition points, the mean shear stress, τ_o , the convective speed of the shear stress data, V , and the shear stress data, τ^* (boundary condition). Free-format, ASCII file conventions were used for both input files. Allocatable arrays were used in the program, so the random access memory (RAM) requirement will depend on the number of grid points and shear stress boundary condition points required by the program inputs.

Program output is directed to a binary-format output file, the name of which is specified by the user as part of the interactive input. The output consists of the number of grid points, N , followed by one record for each stored time

step of the solution. Each record contains the t^* value, followed by the predicted h^* value for all grid points. Binary format was used to minimize file space, since output files can be quite large. In addition, output is only stored at a frequency corresponding to one thousand points per second in real time. This convention was chosen to conserve disk space while still providing output at a frequency consistent with that minimally required by experimental studies. In most cases, this results in only a portion of the calculated predictions being stored. For those cases where the predictions are less than one thousand per second, all output is stored.

In order to process the computed results, two output extraction programs were written. The first program, entitled "FINDTIME," extracts single records from the binary output file generated by program "THINFILM," and writes the data in ASCII format to a user-specified file. These results are then easily processed into graphs using most commercially available plotting software packages. During the initial part of program execution, the user must interactively specify the "THINFILM" output file name, the desired t^* to extract, and the name of the ASCII output file to be created. The resulting output file contains the t^* for the record extracted, the number of grid points, N , and the predicted h^* value for each grid point. Multiple records can be extracted into individual output files during

a single program run. A listing of program "FINDTIME" is provided in Appendix C.

The second output extraction program, entitled "ANIMATE," graphically plots the output generated by program "THINFILM" in animated form on a PC screen using VGA graphics. The user must interactively input the "THINFILM" output file name and an animation time delay factor. Each predicted $h^*(x)$ profile is then plotted to the screen, along with a display of the corresponding t^* value. The user-specified time delay is used between subsequent profile displays, thus allowing the user to slow down the animation of the oil film thinning. This makes it easy for the user to observe the predicted oil film behavior and identify those records to be studied in more detail. Once identified, a detailed study of any given record can be initiated by using program "FINDTIME." A listing of program "ANIMATE" is provided in Appendix D.

Solution time for a typical problem is on the order of several hours on the PC used for this study, depending upon the length of solution desired and the number of grid points. For test case #3 described below, twelve seconds of real time and one hundred grid points were solved for. The elapsed computer time for this case was approximately twelve and a half hours.

III. RESULTS AND DISCUSSION

Two test cases were executed to assess the adequacy and validity of program "THINFILM." These cases are described in Table 1. The two test cases each have analytical solutions, as derived in Appendix E. The analytic solutions were coded into modified versions of program "THINFILM" so that continuous comparisons of the predicted solution and the actual solution could be made at each instant in time. Equation (8) and the corresponding simplifications were considered for all cases.

Table 1

Test Cases Evaluated With Program "THINFILM"

<u>Test Case No.</u>	<u>Description</u>
1	Constant shear stress (i.e., τ^* is constant, so $\partial\tau^*/\partial x^*$ is zero).
2	Linear shear (i.e., $\tau^* = c_1 x^* + c_2$ where c_1 and c_2 are constants, so $\partial\tau^*/\partial x^*$ is constant).

After the two test cases were successfully completed, a sample calculation was performed which used an unsteady variation of the shear stress boundary condition. The

purpose of this calculation was to test the program's ability to simulate oil flow beneath a turbulent boundary layer.

Nominal parameters were established for the test cases, and these were consistently used in the sample calculation as well. Experiments have shown that the boundary layer thickness for real, turbulent, compressible flows is on the order of a few centimeters or less [10]. It is important to keep the stream-wise extent of measurement to less than a boundary layer thickness so that the mean shear stress over the tested region will not change appreciably. Additionally, lengths on the order of one to ten millimeters are necessary for optical measurement. Therefore, to meet both of these requirements, test lengths on the order of a few tens of millimeters or less were used for all calculations in the current study. By definition, the dimensionless stream-wise distance, x^* , ranges from zero to one for all cases.

An oil kinematic viscosity of ten centistokes and a specific gravity of 0.9 was used for all calculations. Larger viscosities are commercially available; however, the chosen value represents approximately the smallest viscosity of a readily available silicon oil which has not been solvent thinned [10].

Oil film thickness measurements have shown that the oil film thickness, h , is on the order of one micron per

centimeter [10]; therefore, this value was incorporated into all calculations and made dimensionless for the appropriate test length under consideration. A value for the mean shear stress, τ_o , equal to one Newton per square meter was chosen for all calculations to be consistent with this profile assumption [10].

A summary of all nominal inputs used for the current study is provided in Table 2.

Table 2
Nominal Input Parameters

	<----- I N P U T V A R I A B L E S ----->				
Calculation	N	L	μ	h	τ^1
<u>Description</u>	<u> </u>	<u>(meter)</u>	<u>(kg/m-sec)</u>	<u>(initial)</u>	<u>variation</u>
Test Case #1	201	0.020	0.009	10 $\mu\text{m}/\text{cm}$ (linear)	constant
Test Case #2	201	0.020	0.009	Figure 8	linear
Sample Calc.	101	0.010	0.009	10 $\mu\text{m}/\text{cm}$ (linear)	unsteady (Fig. 13)

¹Note: $\tau_o = 1.0 \text{ N/m}^2$ for all cases.

Test Case #1 [$\tau^* = \text{constant}$]

For the case of constant shear stress, τ^* is constant, so $\partial\tau^*/\partial x^*$ is zero. A particular, exact solution to equation (8) under these conditions is derived in Appendix E using the initial condition $h^*(x^*) = 0.001x^*$ at $t^* = 0$ (Table 3). The solution takes the following form [4]:

$$h_a^* = \frac{x^*}{(\tau^* t^* + 1000)} \quad (13)$$

This equation was incorporated into the MacCormack subroutine of a modified version of program "THINFILM." In this manner, a continuous comparison of the predicted results could be made with the analytical solution for all grid points at each time step.

The time step required to satisfy the stability criteria for this case is determined from equation (12), noting that $\tau^* = 1 + \tau'^*$:

$$A = \frac{\partial F^*}{\partial h^*} = \frac{\partial}{\partial h^*} \left[\frac{h^{*2}}{2} (1 + \tau'^*) \right] = h^* (1 + \tau'^*) + \frac{\partial \tau'^*}{\partial h^*} \frac{h^{*2}}{2} \quad (14)$$

For constant shear, τ'^* and $\partial\tau'^*/\partial h^*$ are zero. Therefore, $A = h^*$, and the required time step is:

$$\Delta t \leq \frac{\Delta x^*}{h^*} \quad (15)$$

The dimensionless inputs shown in Table 3, which are based on the nominal parameters given in Table 2, were used for this case. These values are typical values used in actual experimental studies.

Table 3
Dimensionless Inputs for Test Case #1
 $[\tau^* = \text{constant}]$

Variable	Value
τ^*	1.00 at $x^* = 0$ to 1.00 at $x^* = 1.0$ (e.g., constant)
Initial h^* Profile	0 at $x^* = 0$ to 0.001 at $x^* = 1.0$ (e.g., linear)

For the time step defined by equation (15), the maximum value of the initial h^* profile (e.g., h^* at $x^* = L$) was used for conservatism. At each time step, the error between the predicted and theoretical solution was determined by:

$$\text{Error} = \left[\frac{\sum_{j=1}^N \left[h_{j,p}^* - h_{j,a}^* \right]^2}{N} \right]^{0.5} \quad (16)$$

where the p and a subscripts indicate the predicted and analytical values for h^* , respectively, at each spatial grid

point, i. The errors for runs with time steps of $1.0\Delta x^*/h^*$, $0.5\Delta x^*/h^*$ and $0.25\Delta x^*/h^*$ are shown in Figures 4, 5 and 6, respectively. The error for a case with half the number of grid points ($N = 101$) and a time step of $1.0\Delta x^*/h^*$ is shown in Figure 7. All other inputs for this latter case were as shown in Tables 2 and 3, and the time step size was modified according to equation (15) for the decreased grid spacing.

The stability results of Figures 4 through 7 are summarized in Table 4 in terms of the t^* value beyond which there was no appreciable increase in the computed error. As can be inferred from these results, the chosen grid spacing is adequate. The approximated solution is considered very good, with the error several orders of magnitude below the predicted h^* values.

Table 4
Stability Results for Test Case #1

<u>Time Step</u>	<u>Number of Grid Points, N</u>	<u>t^* (Error = constant)¹</u>	<u>Ref. Figure</u>
$1.0\Delta x^*/h^*$	201	300	4
$0.5\Delta x^*/h^*$	201	30	5
$0.25\Delta x^*/h^*$	201	30	6
$1.0\Delta x^*/h^*$	101	200	7

¹This is the value of t^* beyond which the error defined by equation (16) showed no significant increase.

Figure 4. Error for Test Case #1
(Time Step = $1.0\Delta x^*/h^*$, $N = 201$)

Figure 5. Error for Test Case #1
(Time Step = $0.5\Delta x^*/h^*$, $N = 201$)

Figure 6. Error for Test Case #1
(Time Step = $0.25\Delta x^*/h^*$, $N = 201$)

Figure 7. Error for Test Case #1
(Time Step = $1.0\Delta x^*/h^*$, $N = 101$)

Test Case #2 [$\partial \tau^* / \partial x^* = \text{constant}$]

For the case of a linear variation in shear stress (e.g., $\tau^* = mx^* + b$, where m and b are arbitrary constants), $\partial \tau^* / \partial x^*$ is constant (equal to m). A particular solution to equation (8) under these conditions is derived in Appendix E [24]. For $b = \tau_0^* = 1.0$:

$$h^* = \frac{2}{(\tau_L^* - 1) t^*} \left[1 - \tau^{*-0.5} \right], \quad (17)$$

where τ_L^* is the τ^* value at $x^* = L$.

This equation was incorporated into the MacCormack subroutine of a second modified version of program "THINFILM." Again, a continuous comparison of the predicted results was made with the analytical solution for all grid points at each time step using equation (16).

The stability criteria for this case is again determined from equation (12). Referring to equation (14), the $(1 + \tau'^*)$ term is of order two. Using the chain rule on the second term:

$$\frac{\partial \tau'^*}{\partial h^*} = \frac{\partial \tau'^*}{\partial x^*} \frac{\partial x^*}{\partial h^*} + \frac{\partial \tau'^*}{\partial t^*} \frac{\partial t^*}{\partial h^*} \quad (18)$$

For this case, $\partial \tau'^* / \partial t^* = 0$, and for the inputs shown in Table 5, $\partial \tau'^* / \partial x^* \approx 0.3$ and $\partial x^* / \partial h^* \approx 1,000$. Therefore, $\partial \tau'^* / \partial h^* \approx 300$. According to equation (14):

$$A = h^* \left[2 + \frac{h^*}{2} \frac{\partial \tau'^*}{\partial h^*} \right] \approx h^* \left[2 + \frac{0.001}{2} 300 \right] = 2.15h^* \quad (19)$$

Thus, the required time step is:

$$\Delta t^* \leq \frac{\Delta x^*}{2.15 h^*} = 0.47 \frac{\Delta x^*}{h^*} \quad (20)$$

The dimensionless inputs shown in Table 5, which are based on the nominal parameters given in Table 2, were used for this case. These values are typical values used in actual experimental studies.

Table 5
Dimensionless Inputs for Test Case #2
 $[\partial \tau^* / \partial x^* = \text{constant}]$

Variable	Value
τ^*	1.00 at $x^* = 0$ to 1.30 at $x^* = 1.0$ (e.g., linear)
Initial h^* Profile	Obeys equation (17) relationship (see Figure 8)

Figure 8. Initial h^* Profile for Test Case #2

For the time step defined by equation (20), the maximum value of the initial h^* profile (e.g., h^* at $x^* = L$) was used for conservatism. At each time step, the error between the predicted and theoretical solution was determined by equation (16). The errors for runs with time steps of $1.0\Delta x^*/h^*$, $0.5\Delta x^*/h^*$ and $0.25\Delta x^*/h^*$ are shown in Figures 9, 10 and 11, respectively. The error for a case with half the number of grid points ($N = 101$) and a time step of $0.25\Delta x^*/h^*$ is shown in Figure 12. All other inputs for this latter case were as shown in Tables 2 and 5, and the time step size was modified according to equation (20) for the

increased grid spacing.

The results of Figures 9 through 12 are summarized in Table 6 in terms of the t^* value beyond which there was no appreciable increase in the computed error. As can be inferred from these results, the chosen grid spacing is adequate for a time step size of $0.5\Delta x^*/h^*$ or less. Instabilities exist for a time step of $1.0\Delta x^*/h^*$, which is consistent with the requirement imposed by equation (20). For the other runs, the approximated solution is considered very good, with the error several orders of magnitude below the predicted h^* values.

Table 6
Stability Results for Test Case #2

<u>Time Step</u>	<u>Number of Grid Points, N</u>	<u>t^* (Error = constant)¹</u>	<u>Ref. Figure</u>
$1.0\Delta x^*/h^*$	201	70	9
$0.5\Delta x^*/h^*$	201	≈ 0	10
$0.25\Delta x^*/h^*$	201	≈ 0	11
$1.0\Delta x^*/h^*$	101	≈ 0	12

¹This is the value of t^* beyond which the error defined by equation (16) showed no significant increase.

Figure 9. Error for Test Case #2
(Time Step = $1.0\Delta x^*/h^*$, $N = 201$)

Figure 10. Error for Test Case #2
(Time Step = $0.5\Delta x^*/h^*$, $N = 201$)

Figure 11. Error for Test Case #2
(Time Step = $0.25\Delta x^*/h^*$, $N = 201$)

Figure 12. Error for Test Case #2
(Time Step = $0.25\Delta x^*/h^*$, $N = 101$)

Sample Calculation [$\tau^*(x,t)$]

A case was selected for execution on "THINFILM" to test the program's ability to accommodate an arbitrary, unsteady variation in the boundary conditions, thus simulating oil flow beneath a turbulent boundary layer. For this case, the shear stress boundary condition, τ^* , was specified as a function of x^* and t^* . The shear stress data was obtained from a three-dimensional, time-dependent, digital simulation of the full Navier-Stokes equations [25-26]. A representative, two-dimensional slice of shear stress data from the digital simulation is shown in Figure 13. This data has been made dimensionless by dividing all points by the average shear stress value (so that $\tau_0 = 1.0 \text{ N/m}^2$).

Figure 13. Shear Stress Data From Digital Simulation

In order for the shear stress boundary conditions to be varied with time simulating a turbulent boundary layer, logic was incorporated into the MacCormack subroutine of program "THINFILM" so that the shear stress data could be "convected" across the spatial grid. This was accomplished by shifting the shear stress data downstream one spatial grid point for each time step. The desired convective speed was established by linearly interpolating the shear stress data. In this manner, the convective speed could be decreased by integer multiples for a constant grid spacing until the desired speed was achieved. For example, doubling the number of shear stress points would halve the convective speed, tripling the number of points would reduce the speed by a factor of three, etc. The program written to perform this interpolation, entitled "TAUGEN," is provided in Appendix F. The only input requirement for program "THINFILM" regarding this interpolation is that the number of shear stress data points must exceed the number of spatial grid points used.

For the initial h^* profile (initial condition), the linear profile utilized in Test Case #1 was chosen. This was an arbitrary choice based on the fact that the input would have to be "adjusted" by the computation to satisfy the equations within the first few time steps of program execution if a successful, converged solution was to be obtained.

The data shown in Figure 13 was originally defined by 128 points with a grid spacing of approximately 0.5 mm (e.g., $L = 64$ mm). Since the grid spacing for the case at hand is 0.1 mm (Table 2), an interpolation factor of five (e.g., $5 \times 128 = 640$ τ^* points) had to be used in program "TAUGEN" to align the shear stress data with the chosen spatial grid.

The desired convection speed, V , is twenty meters per second based on previously known experimental results [10]. Convecting this one spatial grid point per time step requires the following time step:

$$\Delta t = \frac{\Delta x}{V} = \frac{0.1 \times 10^{-3} \text{ m}}{20 \text{ m/sec}} = 5 \times 10^{-6} \text{ second}$$

Or, in non-dimensional form using the parameters shown in Table 2:

$$\Delta t^* = \frac{\Delta t \tau_0}{\mu} = \frac{(5 \times 10^{-6} \text{ sec}) (1.0 \text{ N/m}^2)}{0.009 \text{ kg/m-sec}} = 5.56 \times 10^{-4} \quad (21)$$

The stability criteria for this case is again determined from equation (12). Referring to equation (14), the $(1+\tau'^*)$ term is of order two. The chain rule expansion given by equation (18) is again used to estimate the magnitude of $\partial \tau'^* / \partial h^*$:

$$\frac{\partial r'^*}{\partial x^*} \approx 1.44$$

(Obtained from analysis of shear stress data)

$$\frac{\partial x^*}{\partial h^*} \approx \frac{1}{0.001} = 1,000 \quad (\text{see Table 7})$$

$$\frac{\partial r'^*}{\partial t^*} \approx 25.92$$

(Obtained from analysis of shear stress data)

$$\frac{\partial t^*}{\partial h^*} \approx \frac{1}{1 \times 10^{-6}} = 1 \times 10^6 \quad (\text{Estimated from Test Cases \#1 and \#2 solution output})$$

$$\text{So,} \quad \frac{\partial r'^*}{\partial h^*} \approx (1.44)(1,000) + (25.92)(1 \times 10^6) = 2.59 \times 10^7 \quad (22)$$

Now, according to equation (14):

$$A \approx h^* \left[2 + \frac{h^*}{2} \frac{\partial r'^*}{\partial h^*} \right] = 0.001 \left[2 + \frac{0.001}{2} 2.59 \times 10^7 \right]$$

or $A \approx 12.96 \quad (23)$

Thus, the time step required for stability is:

$$\Delta t_s^* \leq \frac{\Delta x^*}{A} \approx \frac{0.01}{12.96} = 7.71 \times 10^{-4} > \Delta t^* \quad (24)$$

Thus, the time step is below that required for stability for this case. A careful assessment of the results is in order to ensure that this time step is not too small for a good, stable solution (e.g., the time step has not been "over-specified").

The dimensionless inputs shown in Table 7, which are

based on the nominal parameters given in Table 2, were used for this case. These values are typical values used in actual experimental studies.

Table 7
Dimensionless Inputs for Sample Calculation
 $[\tau^*(x,t)]$

Variable	Value
τ^*	unsteady variation (Figure 13)
Initial h^* Profile	0 at $x^* = 0$ to 0.001 at $x^* = 1.0$ (e.g., linear)

The computed solution was obtained for this case for a maximum t^* of 1333 ($t = 12.0$ seconds). The resulting $h^*(x^*)$ profiles for several time steps throughout the solution are given in Figures 14, 15 and 16. The elapsed time to obtain the solution for this case was approximately twelve hours and thirty minutes. The (binary) output file size was approximately five megabytes.

Figure 14. Output Oil Thickness Profiles for Sample Calculation ($t^* = 0, 100, 200, 300, 400$)

Figure 15. Output Oil Thickness Profiles for Sample Calculation ($t^* = 500, 600, 700, 800, 900$)

Figure 16. Output Oil Thickness Profiles for Sample Calculation ($t^* = 1000, 1100, 1200, 1300, 1333$)

As a check on the results, the approximated h^* for this case at $x^* = 1.0$ (e.g., $x = L$) was compared to that predicted for Test Case #1. The overall oil thinning is dominated by the mean thinning rate, which is governed by the mean shear stress. Since, by definition, the mean shear stress is constant, the results are expected to be similar to those of Test Case #1. However, the h^* predictions are not expected to exactly correspond with those of Test Case #1 because of the unsteady variation in the shear stress boundary condition.

The comparison between this sample calculation and Test Case #1 is made in Table 8 and Figures 17 and 18. Table 8 gives a comparison between predicted h^* values for both cases. Figure 17 shows a plot of the slope of the leading edge of the oil film (e.g., $\partial h^* / \partial x^*$ in the vicinity of $x^* = 0$) as a function of t^* for both cases. Figure 18 is a magnified section of Figure 17, where the undulations caused by the unsteady variation of the shear stress boundary condition are evident for the sample calculation. The data for both of these figures was generated from the output of program "THINFILM" using a least squares fit of the h^* prediction for the first five spatial grid points, forced through $h^* = 0$ at the first grid point (e.g., $x^* = 0$). The computer program written to perform this procedure, entitled "SLOPE," is provided in Appendix G.

Table 8. Comparison of Results
Sample Calculation vs. Test Case #1
(Predicted h^* at $x^* = 1.0$)

t^*	t	h^*	h^*	Difference
	(sec.)	Sample Calculation	Test Case #1	(%)
111.1	1.0	9.0347×10^{-4}	9.0000×10^{-4}	+ 0.39
555.6	5.0	6.5176×10^{-4}	6.4286×10^{-4}	+ 1.38
777.8	7.0	5.7207×10^{-4}	5.6250×10^{-4}	+ 1.70
1111.1	10.0	4.8341×10^{-4}	4.7368×10^{-4}	+ 2.05
1333.3	12.0	4.3814×10^{-4}	4.2857×10^{-4}	+ 2.23

Figure 17. Comparison of Slope Change Near
Leading Edge of Oil Film

Figure 18. Comparison of Slope Change Near
Leading Edge of Oil Film ($1 < t^* < 2$)

As expected, the results appear very favorable, with the results of the sample calculation closely following those of Test Case #1. It is interesting to note the difference between the two solutions as shown in Figure 17. For the same mean shear stress, it was expected that the two solutions would be identical, with the only visible difference being the unsteady variation for the sample calculation as shown in Figure 18. The fact that the two solutions are not identical suggests a small 'truncation error may have accumulated with the small time step used in the sample calculation, as shown by equation (24). This observation may be consistent with "over-specifying" the time step, as previously discussed. Despite this small discrepancy, the results of the sample calculation are considered to be very good, with manageable execution time and output file size.

IV. SUMMARY AND CONCLUSIONS

In an attempt to address unsteady skin friction measurements, computer modeling of thin oil film behavior was undertaken in the current study. Such an analytical tool could prove valuable in the future investigation of **unsteady** skin friction measurements.

As a start, the continuity and Navier-Stokes equations were simplified. For thin oil films on a flat surface, use of creeping flow approximations (lubrication theory) allowed the inertial terms in these equations to be neglected when compared to the viscous terms. Further simplification was realized after order-of-magnitude analysis of the viscous terms, and converting the remaining variables into dimensionless form. The resulting partial differential equation (8) was classified as a first-order, hyperbolic, quasi-linear, non-homogeneous, partial differential equation which took the form of the inviscid Burger's equation. This equation describes the variation of the oil film thickness as a function of time, distance along the test surface, imposed pressure gradient and imposed shear stress.

Further analysis and numerical techniques were applied to solve equation (8) using unsteady variations in the shear stress boundary condition. This solution provides a means of simulating the characteristics of the interferometry system used for measuring the time-varying oil film

thickness. With such a simulation, applications of interferometry can be further explored in detail, especially with regard to simulated oil flow beneath a turbulent boundary layer.

An explicit, finite difference computer code was written which used the MacCormack method to approximate the solution to the governing differential equation. This method was chosen for its simplicity, stability and past success at solving Burger's equation. Two test cases which had analytical solutions were run to verify proper program execution. Comparisons were made for these two cases between the predicted and the theoretical solutions with excellent agreement.

Finally, a sample calculation was performed which utilized an unsteady variation of the shear stress boundary condition. The shear stress data used for this calculation was obtained from a three-dimensional, digital simulation of the full Navier-Stokes equations. Because of the dominant influence of the **mean** component of shear stress on the oil film thinning rate, the results of this case were compared to those of Test Case #1 ($\tau^* = \text{constant}$). Some minor differences in the results were found: (1) the sample calculation results "undulated" as expected due to the imposed variation in the boundary condition, and (2) a slight difference in the **mean** solutions existed, most likely caused by the significantly reduced time step used in the

sample calculation.

The sample calculation reveals that application of unsteady boundary conditions gives rise to a drastic reduction in the required time step necessary to achieve stability. There appears to be a small truncation error associated with the smaller time step, which accumulates in the mean solution. However, it appears evident from the results that careful selection of the input parameters will still allow the stability requirements for the MacCormack method to be met without a significant degradation in the solution. Appropriate results can be realized with manageable computer execution time and output. Despite this, further studies with higher order, implicit, finite difference schemes may prove useful in achieving a more accurate and efficient approximation.

Collectively, the results of this study indicate that the computer code is functioning properly and that digital simulation of thin oil film behavior with variable shear stress boundary conditions is achievable on a computer. It is therefore concluded that the computer code is thoroughly adequate for the intended purpose of simulating the flow of oil beneath a turbulent boundary layer. Further, the results of the sample calculation imply that measurements of **unsteady** variations in skin friction are possible, as long as the measurement device has the ability to detect successive "peaks" similar to those shown in Figure 18.

V. REFERENCES

1. Winter, K.G., "An Outline of the Techniques Available for the Measurement of Skin Friction in Turbulent Boundary Layers," Progress in Aerospace Sciences, Volume 18, 1977, pp. 1-57.
2. Bertelrud, A., "Skin Friction Measurement Techniques in Incompressible Turbulent Flow - A Literature Survey," The Aeronautical Research Institute of Sweden, Technical Note FFA-AU-726, 1972.
3. Settles, G.S., "Recent Skin Friction Techniques for Compressible Flows," AIAA/ASME 4th Fluid Mechanics, Plasma Dynamics and Lasers Conference, AIAA-86-1099, May 12-14, 1986.
4. Tanner, L.H., and Blows, L.G., "A Study of the Motion of Oil Films on Surfaces in Air Flow, with Application to the Measurement of Skin Friction," J. Phys. E.: Sci. Instrum., Volume 9, 1976, pp. 194-202.
5. Tanner, L.H., and Kulkarni, V.G., "The Viscosity Balance Method of Skin Friction Measurement: Further Developments Including Applications to Three-Dimensional Flow," J. Phys. E.: Sci. Instrum., Volume 9, 1976, pp. 1114-1121.

6. Tanner, L.H., "The Application of Fizeau Interferometry of Oil Films to the Study of Surface Flow Phenomena," Optics and Lasers in Engineering, Volume 2, 1981, pp. 105-118.
7. Monson, D.J., "A Nonintrusive Laser Interferometer Method for Measurement of Skin Friction," Experiments in Fluids, Volume 1, 1983, pp. 15-22.
8. Monson, D.J., "A Laser Interferometer for Measuring Skin Friction in Three-Dimensional Flows," AIAA Technical Notes, April 1984, pp. 557-559.
9. Murphy, J.D., and Westphal, R.V., "The Laser Interferometer Skin-Friction Meter: A Numerical and Experimental Study," J. Phys. E.: Sci. Instrum., Volume 19, 1986, pp. 744-751.
10. Westphal, R.V., Bachalo, W.D., and Houser, M.H., "Improved Skin Friction Meter," NASA Technical Memorandum 88216, March 1986.
11. Sherman, F.S., Viscous Flow, McGraw-Hill Book Company, 1990, pp.233-250.
12. Shampine, L.F., and Thompson, R.J., "Difference Methods for Nonlinear First-Order Hyperbolic Systems of Equations," Math. Comp., Vol. 24, No. 109, January 1970, pp. 45-56.
13. Gourlay, A.R., and Morris, J.L., "Finite-Difference Methods for Nonlinear Hyperbolic Systems," Math. Comp., Volume 22, No. 101, January 1968, pp. 28-39.

14. Abarbanel, S., and Gottlieb, D., "Higher Order Accuracy Finite Difference Algorithms for Quasi-Linear, Conservation Law Hyperbolic Systems," Math. Comp., Volume 27, No. 123, July 1973, pp. 505-523.
15. Keller, Herbert B., and Thomee, Vidar, "Unconditionally Stable Difference Methods for Mixed Problems for Quasi-Linear Hyperbolic Systems in Two Dimensions," Comm. Pure Applied Math., Volume XV, 1962, pp. 63-73.
16. Kowalski, Z., "A Difference Method for Certain Hyperbolic Systems of Non-Linear Partial Differential Equations of the First Order," Ann. Polon. Math., Volume XIX, 1967, pp. 313-322.
17. Hinze, J.O., Turbulence, 2nd edition, McGraw-Hill Book Company, 1975, pp. 15-30.
18. Anderson, Dale A., Tannehill, John C., and Pletcher, Richard H., Computational Fluid Mechanics and Heat Transfer, Hemisphere Publishing Corporation, McGraw-Hill Book Company, 1984.
19. Goldberg, M., "A Note on the Stability of an Iterative Finite-Difference Method for Hyperbolic Systems," Math. Comp., Vol. 27, No. 121, January 1973, pp. 41-44.
20. McGuire, G.R., and Morris, J.L., "Explicit-Implicit Schemes for the Numerical Solution of Nonlinear Hyperbolic Systems," Math. Comp., Volume 29, No. 130, April 1975, pp. 407-424.

21. Goldberg, M., and Tadmor, E., "Convenient Stability Criteria for Difference Approximations of Hyperbolic Initial-Boundary Value Problems," Math. Comp., Vol. 44, No. 170, April 1985, pp. 361-377.
22. Goldberg, M., and Tadmor, E., "Convenient Stability Criteria for Difference Approximations of Hyperbolic Initial-Boundary Value Problems, II," Math. Comp., Volume 48, No. 178, April 1987, pp. 503-520.
23. Microsoft FORTRAN Compiler, Version 5.0, Microsoft Corporation, Redmond, WA, 1989.
24. Monson, D.J., Driver, D.M., and Szodruch, J., "Application of a Laser Interferometer Skin Friction Meter in Complex Flows," ICIASF '81 Record, IEEE Pub. 81CH1712-9, September 30, 1981.
25. Westphal, R.V., "Some Effects of Finite Spatial Resolution on Skin Friction Measurements in Turbulent Boundary Layers," AIAA-3753, July 1988.
26. Kim, J., Moin, P., and Moser, R., "Turbulent Statistics in Fully Developed Channel Flow at Low Reynolds Number," J. Fluid Mech., Volume 177, April 1987, pp. 133-166.

APPENDIX A

EQUATION DERIVATION

The Navier-Stokes Equation for two-dimensional flow is given as:

$$\rho \frac{D\bar{V}}{Dt} = \rho \bar{g} - \nabla \bar{p} + \mu \nabla^2 \bar{V} \quad (A-1)$$

where the bars indicate vectors. In the direction of flow along a horizontal plate (level surface), equation (A-1) becomes (for two-dimensional flow):

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} = g_x - \frac{1}{\rho} \frac{\partial p}{\partial x} + \nu \left[\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right] \quad (A-2)$$

Creeping flow approximation (lubrication theory) and order-of-magnitude analysis reduces equation (A-2) to the following:

$$-\frac{1}{\rho} \frac{dp}{dx} + \nu \frac{d^2 u}{dy^2} = 0 \quad (A-3)$$

where the partial derivatives have been replaced by exact derivatives (this is shown by analyzing the y-direction momentum equation). Thus,

$$\nu \frac{d^2 u}{dy^2} = \frac{1}{\rho} \frac{dp}{dx} \quad \text{or} \quad u'' = \frac{1}{\mu} \frac{dp}{dx} \quad (A-4)$$

Integrating equation (A-4):

$$u' = \frac{1}{\mu} \frac{dp}{dx} y + C_1 \quad (A-5)$$

$$\text{at } y = h, \quad \tau = \mu \frac{du}{dy} = \mu u' \quad \text{or} \quad u' = \frac{\tau}{\mu}$$

$$\text{so,} \quad C_1 = \frac{1}{\mu} \left[\tau - \frac{dp}{dx} h \right] \quad (A-6)$$

Substituting (A-6) into (A-5):

$$u' = \frac{1}{\mu} \frac{dp}{dx} y + \frac{1}{\mu} \left[\tau - \frac{dp}{dx} h \right] \quad (A-7)$$

Integrating equation (A-7):

$$u = \frac{1}{\mu} \frac{dp}{dx} \frac{y^2}{2} + \frac{1}{\mu} \left[\tau - \frac{dp}{dx} h \right] y + C_2 \quad (A-8)$$

at $y = 0$, $u = 0$ and $C_2 = 0$, so:

$$u = \frac{y}{\mu} \left[\frac{dp}{dx} \frac{y}{2} + \tau - \frac{dp}{dx} h \right]$$

$$\text{or:} \quad u = \frac{y}{\mu} \left[\tau - \frac{dp}{dx} \left(h - \frac{1}{2} y \right) \right] \quad (A-9)$$

where τ and $\frac{dp}{dx}$ are specified functions of x and t .

$$\text{at } y = h, \quad u_F = \frac{h}{\mu} \left[\tau - \frac{dp}{dx} \left(h - \frac{1}{2} h \right) \right] \quad (A-10)$$

$$u_F = \frac{h}{\mu} \left[\tau - \frac{dp}{dx} \frac{h}{2} \right] \quad (A-11)$$

$$u_F = \frac{r h}{\mu} \left[1 - \frac{dp}{dx} \frac{h}{2r} \right] \quad (A-12)$$

The continuity equation for two-dimensional flow is given as:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \quad \text{or} \quad \frac{\partial u}{\partial x} = -\frac{\partial v}{\partial y} \quad , \quad (A-13)$$

Writing equation (A-9) in expanded form, the following is obtained:

$$u = \frac{y}{\mu} \tau - \frac{h y}{\mu} \frac{dp}{dx} + \frac{y^2}{2\mu} \frac{dp}{dx} \quad (A-14)$$

Differentiating equation (A-14):

$$\frac{\partial u}{\partial x} = \frac{y}{\mu} \frac{\partial \tau}{\partial x} - \frac{y}{\mu} \frac{dp}{dx} \frac{\partial h}{\partial x} - \frac{h y}{\mu} \frac{d^2 p}{dx^2} + \frac{y^2}{2\mu} \frac{d^2 p}{dx^2} \quad (A-15)$$

$$\frac{\partial v}{\partial y} = -\frac{\partial u}{\partial x} = -\frac{y}{\mu} \frac{\partial \tau}{\partial x} + \frac{y}{\mu} \frac{dp}{dx} \frac{\partial h}{\partial x} + \frac{h y}{\mu} \frac{d^2 p}{dx^2} - \frac{y^2}{2\mu} \frac{d^2 p}{dx^2} \quad (A-16)$$

Integrating equation (A-16):

$$v = -\frac{y^2}{2\mu} \frac{\partial \tau}{\partial x} + \frac{y^2}{2\mu} \frac{dp}{dx} \frac{\partial h}{\partial x} + \frac{h y^2}{2\mu} \frac{d^2 p}{dx^2} - \frac{y^3}{6\mu} \frac{d^2 p}{dx^2} + C_3 \quad (A-17)$$

at $y = 0$, $v = 0$, so $C_3 = 0$

Evaluating equation (A-17) at $y = h$:

$$v = -\frac{h^2}{2\mu} \frac{\partial \tau}{\partial x} + \frac{h^2}{2\mu} \frac{dp}{dx} \frac{\partial h}{\partial x} + \frac{h^3}{2\mu} \frac{d^2 p}{dx^2} - \frac{h^3}{6\mu} \frac{d^2 p}{dx^2} \quad (A-18)$$

$$v_F = -\frac{h^2}{2\mu} \frac{\partial \tau}{\partial x} + \frac{h^2}{2\mu} \frac{dp}{dx} \frac{\partial h}{\partial x} + \frac{h^3}{3\mu} \frac{d^2 p}{dx^2} \quad (A-19)$$

Integrating equation (A-13) with respect to y across the thickness of the oil film:

$$\int_0^h \frac{\partial u}{\partial x} dy + \int_0^h \frac{\partial v}{\partial y} dy = 0 \quad (A-20)$$

Using Leibnitz's Rule for differentiating integrals, the first term of equation (A-20) may be written as:

$$\int_0^h \frac{\partial u}{\partial x} dy = \frac{\partial}{\partial x} \int_0^h u dy - \frac{\partial h}{\partial x} u_{(y=h)} + \frac{\partial(0)}{\partial x} u_{(y=0)} \quad (A-21)$$

$$\text{or} \quad \int_0^h \frac{\partial u}{\partial x} dy = \frac{\partial}{\partial x} \int_0^h u dy - \frac{\partial h}{\partial x} u_F \quad (A-22)$$

The second term of equation (A-20) is:

$$\int_0^h \frac{\partial v}{\partial y} dy = \int_0^h dv = v_F \quad (A-23)$$

Thus, equation (A-20) becomes:

$$\frac{\partial}{\partial x} \int_0^h u dy - \frac{\partial h}{\partial x} u_F + v_F = 0 \quad (A-24)$$

The integral form of the conservation of mass for unit density is:

$$\dot{m}_{out} - \dot{m}_{in} + \frac{\partial}{\partial t}(\dot{m}_{cv}) = 0 \quad (A-25)$$

where:

$$\dot{m}_{cv} = h \, dx, \quad \dot{m}_{in} = \int_0^h u \, dy, \quad \dot{m}_{out} = \dot{m}_{in} + \frac{\partial}{\partial x}(\dot{m}_{in}) \, dx$$

Thus, equation (A-25) becomes:

$$\frac{\partial}{\partial x} \left[\int_0^h u \, dy \right] dx = - \frac{\partial}{\partial t} (h \, dx) \quad (A-26)$$

$$\text{so,} \quad \frac{\partial}{\partial x} \int_0^h u \, dy = - \frac{\partial h}{\partial t} \quad (A-27)$$

Substituting equation (A-27) into (A-24):

$$\frac{\partial h}{\partial t} = v_F - \frac{\partial h}{\partial x} u_F \quad (A-28)$$

Substituting equations (A-12) and (A-19) into (A-28), the following is obtained:

$$\frac{\partial h}{\partial t} = - \frac{h^2}{2\mu} \frac{\partial \tau}{\partial x} + \frac{h^2}{2\mu} \frac{dp}{dx} \frac{\partial h}{\partial x} + \frac{h^3}{3\mu} \frac{d^2 p}{dx^2} - \frac{\tau h}{\mu} \left[1 - \frac{h}{2\tau} \frac{dp}{dx} \right] \frac{\partial h}{\partial x} \quad (A-29)$$

$$\frac{\partial h}{\partial t} = \frac{\tau h}{\mu} \left[-\frac{h}{2\tau} \frac{\partial \tau}{\partial x} + \frac{h}{2\tau} \frac{dp}{dx} \frac{\partial h}{\partial x} + \frac{h^2}{3\tau} \frac{d^2 p}{dx^2} - \frac{\partial h}{\partial x} + \frac{h}{2\tau} \frac{dp}{dx} \frac{\partial h}{\partial x} \right]$$

(A-30)

And finally:

$$\frac{\partial h}{\partial t} = \frac{\tau h}{\mu} \left[-\frac{h}{2\tau} \frac{\partial \tau}{\partial x} + \left[\frac{h}{\tau} \frac{dp}{dx} - 1 \right] \frac{\partial h}{\partial x} + \frac{h^2}{3\tau} \frac{d^2 p}{dx^2} \right]$$

(A-31)

APPENDIX B

PROGRAM "THINFILM" LISTING

```

C*****
C**          PROGRAM:  THINFILM.FOR          **
C**          -----                          **
C**          **                                     **
C** Programmed by: Gary L. Stevens          Date: October 28, 1990 **
C**                      Last Revised: 02/24/91 [gls]          **
C**          **                                     **
C**          MASTER'S THESIS PROJECT          **
C**          Mechanical Engineering Department, San Jose State University **
C**          Fall Quarter 1990 (ME-298) & Spring Quarter 1991 (ME-299) **
C**          Advisor: R.V. Westphal          **
C**-----**
C** Description:                               **
C** -----                               **
C** This program uses an explicit MacCormack finite difference method **
C** to approximate the solution to the thin oil film equation. The **
C** equation takes the following form:          **
C**                      2                  **
C**          dh' d (TAU' h' )                **
C**          -- = --                        **
C**          dt' dx' 2                      **
C**          **                                     **
C** where the primes denote non-dimensionalized forms of h, x, TAU **
C** and t. This equation is classified as a 1st-order, hyperbolic, **
C** non-linear, non-homogeneous partial differential equation. It is **
C** also known in the literature as Burger's Equation.          **
C*****
C
C          CHARACTER*15 REV
C
C          Current revision information.
C
C          REV = '02/24/91 [gls]'
C
C          Obtain the required file names from the user.
C
C          10 CALL GETFILE (REV)
C
C          Read the input files.
C          (MacCormack subroutine is called from READFILE subroutine)
C
C          CALL READINPT
C
C          Finished! Run another case?
C

```

```

        ANSWER = ' '
        WRITE (*,108)
108  FORMAT (//,5X,'FINISHED!  Another case? (Y or N [DEFAULT]): ', $)
        READ (*,202) ANSWER
202  FORMAT (A1)
        IF (ANSWER .EQ. 'Y' .OR. ANSWER .EQ. 'y') GO TO 10
        STOP '      Program stopped!'
        END
C
C*****
      SUBROUTINE GETFILE (REV)
C+++++
C++   This subroutine prints the program banner, obtains the required ++
C++   input and output file names from the user, and opens the files.  ++
C+++++
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
      COMMON /INPUT/ FILE1,FILE2,OUTFILE,IFLFLG
C
      CHARACTER*30 OUTFILE,FILE1,FILE2,ANSWER*1,AMPM*2
      CHARACTER*1 A,B,C,D,E,F,REV*15
C
C   Assign the banner border variables (standard ASCII characters).
C
      A = CHAR (201)
      B = CHAR (205)
      C = CHAR (187)
      D = CHAR (186)
      E = CHAR (200)
      F = CHAR (188)
C
C   Get time and date and print banner.
C
      CALL GETTIM (IHR,IMIN,ISEC,I100TH)
      CALL GETDAT (IYR,IMON,IDAY)
      IF (IHR .GT. 12) THEN
        IHR = IHR-12
        AMPM = 'pm'
      ELSE
        AMPM = 'am'
      END IF
      WRITE (*,10) IMON,IDAY,IYR,IHR,IMIN,AMPM,A,
&B,B,B,B,B,B,B,B,B,B,B,B,B,B,B,B,B,B,B,B,B,B,B,B,B,B,B,C,D,D,E,
&B,B,B,B,B,B,B,B,B,B,B,B,B,B,B,B,B,B,B,B,B,B,B,B,B,B,B,F,REV,
&B,B,B,B,B,B,B,B,B,B,B,B,B,B,B,B,B,B,B,B,B,B,B,B,B,B,B,B,B,B,
&B,B,B,B,B,B,B,B,B,B,B,B,B,B,B,B,B,B,B,B,B,B,B,B,B,B,B,B,B,B
10  FORMAT (//,20X,'Date: ',I2,'/',I2.2,'/',I4,'      Time: ',
&I2,':',I2.2,1X,A2,
&      /,25X,28A1,
&      /,25X,A1,' PROGRAM: THINFILM.FOR ',A1,

```

```

&          /,25X,28A1,
&          /,15X,'      Thin oil film thickness approximation.',
&          /,15X,'      Programmed by Gary L. Stevens   10/28/90',
&          /,15X,'      Last Revised:  ',A15,
&          /,18X,42A1)
C
C Obtain the input file names from the user.
C
  5 WRITE (*,20)
20  FORMAT (/,5X,'Enter the oil thickness (h) INPUT file name : ',,$)
  READ (*,30) FILE1
30  FORMAT (A30)
35  WRITE (*,40)
40  FORMAT (/,5X,'Enter the shear stress (TAU) INPUT file name: ',,$)
  READ (*,30) FILE2
C
C Open the user-specified input files.
C
  OPEN (01,FILE=FILE1,STATUS='OLD',ERR=996)
  OPEN (02,FILE=FILE2,STATUS='OLD',ERR=997)
C
C Obtain output file name and, if provided, open file (binary).
C
45  IFLFLG = 0
  ANSWER = 'N'
  WRITE (*,50)
50  FORMAT (/,5X,'Do you want FILE output? (Y or N [DEFAULT]): ',,$)
  READ (*,60) ANSWER
60  FORMAT (A1)
  IF (ANSWER .EQ. 'Y' .OR. ANSWER .EQ. 'y') THEN
    IFLFLG = 1
    WRITE (*,70)
70  FORMAT (/,5X,'Enter the OUTPUT file name : ',,$)
    READ (*,30) OUTFILE
    OPEN (03,FILE=OUTFILE,FORM='BINARY',STATUS='UNKNOWN',ERR=998)
  END IF
C
C Return to the main routine.
C
  RETURN
C
C Error opening files!
C
996 ANSWER = 'Y'
  WRITE (*,80) FILE1
80  FORMAT (/,5X,'ERROR! Invalid input file! (',A30,')',
&          /,5X,'      Try again? (Y [DEFAULT] or N) : ',,$)
  READ (*,60) ANSWER
  IF (ANSWER .EQ. 'N' .OR. ANSWER .EQ. 'n') STOP ' PROGRAM STOPPED!'
  GO TO 5
997 ANSWER = 'Y'

```



```

        WRITE (*,85) FILE2
85  FORMAT (/ ,5X,'ERROR!  Invalid input file!  (' ,A30,')',
      &    / ,5X,'          Try again? (Y [DEFAULT] or N)          : ', $)
      READ (*,60) ANSWER
      IF (ANSWER .EQ. 'N' .OR. ANSWER .EQ. 'n') STOP ' PROGRAM STOPPED!'
      GO TO 35
998 ANSWER = 'Y'
      WRITE (*,90) OUTFILE
90  FORMAT (/ ,5X,'ERROR!  Invalid output file!  (' ,A30,')',
      &    / ,5X,'          Try again? (Y [DEFAULT] or N)          : ', $)
      READ (*,60) ANSWER
      IF (ANSWER .EQ. 'N' .OR. ANSWER .EQ. 'n') STOP ' PROGRAM STOPPED!'
      GO TO 45
C
      END
C
C*****
      SUBROUTINE READINPT
C+++++
C++  This subroutine reads the program input from the user-specified ++
C++  input files, gets the necessary input from the user, and allocates ++
C++  the required memory for the data arrays.                                ++
C+++++
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
      COMMON /DATA/  IXPTS,NTAUPTS,DT,DX,TMAX,IXPRT,IOUT
      COMMON /DATA/  LENGTH,TAUO,SPEED,MU
      COMMON /INPUT/ FILE1,FILE2,OUTFILE,IFLFLG
C
      CHARACTER*30 OUTFILE,FILE1,FILE2
      DIMENSION H [ALLOCATABLE] (:,:)
      DIMENSION TAU [ALLOCATABLE] (:)
      DIMENSION TAUST [ALLOCATABLE] (:)
      REAL LENGTH,MU
C
C  Read header line from the oil thickness input file and allocate H array.
C
      READ (01,*) IXPTS,LENGTH,MU
      ALLOCATE (H(0:IXPTS-1,2),STAT=ISTAT)
      IF (ISTAT .NE. 0) THEN
        WRITE (*,10) IXPTS
10      FORMAT (//,10X,'ERROR!  Unable to allocate memory for H array!'
      &        / ,10X,'          (One 2-D array of ',I4,' points)')
        STOP '          PROGRAM STOPPED.'
      END IF
C
C  Read data from the oil thickness input file and determine maximum h.
C
      HMAX = 0.0
      WRITE (*,15)

```

```

15 FORMAT (/,10X,'Reading oil thickness input. . . .')
   DO 20 I=1,IXPTS
   READ (01,*) H(I-1,1)
   HMAX = MAX (H(I-1,1),HMAX)
20 CONTINUE
C
C Calculate delta-x.
C
   DX = 1.0/(IXPTS-1)
C
C Read header line from the shear stress (TAU) input file and allocate TAU
C and TAUST arrays.
C
   READ (02,*) NTAUPTS,TAU0,SPEED
   IF (NTAUPTS .LT. IXPTS)
   & STOP '**** ERROR! # TAU points < # H points! PROGRAM STOPPED!'
   ALLOCATE (TAU(0:NTAUPTS-1),TAUST(0:NTAUPTS-1),STAT=K$STAT)
   IF (K$STAT .NE. 0) THEN
   WRITE (*,30) NTAUPTS
30  FORMAT (//,10X,'ERROR! Unable to allocate memory for TAU ',
   &          'arrays!'
   &          /,10X,'          (Two 1-D arrays of ',I4,' points each)')
   STOP '          PROGRAM STOPPED.'
   END IF
C
C Read data from the shear stress (TAU) input file.
C
   WRITE (*,35)
35  FORMAT (/,10X,'Reading shear stress input. . . .')
   DO 40 I=1,NTAUPTS
   READ (02,*) TAU(I-1)
40 CONTINUE
C
C Get desired screen output point from user.
C
   50 WRITE (*,60) IXPTS
   60 FORMAT (/,5X,'There are ',I4,' X grid points.',
   &          /,5X,'Which one do you want printed to the screen?: ',I4)
   READ (*,*) IXPRT
   IF (IXPRT .LT. 1 .OR. IXPRT .GT. IXPTS) THEN
   WRITE (*,70) IXPTS
   70  FORMAT (/,5X,'ERROR! Input must be between 1 and ',I4,'!'
   &          /,5X,'          Try again!')
   GO TO 50
   END IF
C
C Calculate delta-t.
C
   DELTAT = DX*LENGTH/SPEED
   DT = DELTAT*TAU0/MU
C

```

```

C Calculate output frequency (save 1000 points/second)
C
      IOUT = INT((0.001/DELTAT)+0.5)
      IF (IOUT .LT. 1) IOUT = 1
C
C Get maximum t* for solution from user.
C
      WRITE (*,80)
      80 FORMAT (/,5X,'Enter the maximum t* for solution          : ', $)
      READ (*,*) TMAX
C
C Close the input files.
C
      CLOSE (01)
      CLOSE (02)
C
C Call the MacCormack subroutine and solve the PDE.
C
      CALL MCRMCK (H,TAU,TAUST)
C
C Deallocate arrays and return to the main routine.
C
      DEALLOCATE (H,TAU,TAUST,STAT=LSTAT)
      RETURN
      END
C
C*****
      SUBROUTINE MCRMCK (H,TAU,TAUST)
C+++++
C++ This subroutine uses the explicit MacCormack finite difference ++
C++ method to approximate the solution to the thin oil film partial ++
C++ differential equation (1st-order, non-linear, hyperbolic). For ++
C++ details of the method, refer to: ++
C++ ++
C++ Dale A. Anderson, John C. Tannehill and Richard H. Pletcher, ++
C++ "Computational Fluid Mechanics and Heat Transfer," Hemisphere ++
C++ Publishing Corporation, McGraw-Hill Book Company, 1984. ++
C++ (see Section 4-4.3, pages 146-147) ++
C++ ++
C++ The results are written to the screen for the point requested by ++
C++ the user, and all points are written to the output file. ++
C+++++
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
      COMMON /DATA/ IXPTS,NTAUPTS,DT,DX,TMAX,IXPRT,IOUT
      COMMON /DATA/ LENGTH,TAU0,SPEED,MU
      COMMON /INPUT/ FILE1,FILE2,OUTFILE,IFLFLG
C
      CHARACTER*30 OUTFILE,FILE1,FILE2
      DIMENSION H(0:IXPTS-1,2),TAU(0:NTAUPTS-1),TAUST(0:NTAUPTS-1)

```

```

      REAL LENGTH,MU
C
C   Write the number of X points to the output file (if file output is
C   desired).
C
      IF (IFLFLG .EQ. 1)
        & WRITE (03) IXPTS,SNGL(LENGTH),SNGL(SPEED),SNGL(TAU0),SNGL(MU)
C
C   Write the output header to the screen (for debugging).
C
      WRITE (*,10) IXPTS,(IXPTS-1)*DX
C   10 FORMAT (//,15X,'Output for point # ',I4,', X* = ',F7.5,
C   &          //,15X,'      t*      ',5X,' Approx. h* ',
C   &          //,15X,'-----',5X,'-----')
C
C   Start of t loop.
C
      N = 0
      L = 0
C
C   Write all points to the output file every IOUT iterations, and only the
C   one point specified by the user to the screen every IOUT iterations.
C
      20 IF (IFLFLG .EQ. 1 .AND. MOD(N,IOUT) .EQ. 0)
        & WRITE (03) SNGL((N*DT)),(SNGL(H(K,1)),K=0,IXPTS-1)
C      IF (MOD(N,IOUT) .EQ. 0) WRITE (*,30) (N*DT),H(IXPTS-1,1)
C      30 FORMAT (15X,1PE12.5,5X,1PE12.5)
        IF (N*DT .GE. TMAX) GO TO 999
        N = N+1
C
C   Initialize variables used for convecting TAU.
C
      L = L+1
      IF (L .GT. NTAUPTS) L=1
      K = 0
C
C   Convect TAU by one grid point.
C
      IF (L .EQ. 1) THEN
        DO 40 M=1,IXPTS
          TAUST(K) = TAU(M-1)
          K = K+1
40      CONTINUE
      ELSE IF (L .GE. 2 .AND. L .LE. IXPTS) THEN
        DO 50 M=NTAUPTS-L+2,NTAUPTS
          TAUST(K) = TAU(M-1)
          K = K+1
50      CONTINUE
        DO 60 M=1,IXPTS-L+1
          TAUST(K) = TAU(M-1)
          K = K+1

```

```

60  CONTINUE
    ELSE
        DO 70 M=NTAUPTS-L+2,NTAUPTS-L+IXPTS+1
            TAUST(K) = TAU(M-1)
            K = K+1
        70  CONTINUE
    END IF

C
C  Start of X loop.
C  For each time step, compute h for all X points.
C
C***** FOR FIRST POINT (J=0) ONLY
C
        HPROLD = 0.0
        H(0,2) = 0.0
C
        DO 80 J = 1,IXPTS-1
C
C***** FOR LAST POINT (J=IXPTS-1) ONLY
C
            IF (J .EQ. IXPTS-1) THEN
C
C  Predictor step.
C
                F1 = 0.5*TAUST(J)*H(J,1)**2
                F0 = 0.5*TAUST(J-1)*H(J-1,1)**2
                HPRNEW = H(J,1) - (DT/DX)*(F1-F0)
C
C***** FOR ALL POINTS EXCEPT FIRST (J=0) AND LAST (J=IXPTS-1)
C
                ELSE
C
C  Predictor step.
C
                    F1 = 0.5*TAUST(J+1)*H(J+1,1)**2
                    F0 = 0.5*TAUST(J)*H(J,1)**2
                    HPRNEW = H(J,1) - (DT/DX)*(F1-F0)
                END IF
C
C  Corrector step.
C
                F3 = 0.5*TAUST(J)*HPRNEW**2
                F2 = 0.5*TAUST(J-1)*HPROLD**2
                H(J,2) = 0.5*(H(J,1) + HPRNEW - (DT/DX)*(F3-F2))
C
C  End of X loop.
C
                HPROLD = HPRNEW
            80 CONTINUE
C
C  Update h array.

```

```

C      DO 90 K=0,IXPTS-1
        H(K,1) = H(K,2)
    90 CONTINUE
C
C      End of t loop.
C
        GO TO 20
C
C      Close the output file (if necessary) and return to the calling subroutine.
C
    999 IF (IFLFLG .EQ. 1) CLOSE (03)
        RETURN
        END

```

APPENDIX C

PROGRAM "FINDTIME" LISTING

```

C*****
C**          PROGRAM:  FINDTIME.FOR          **
C**          -----                          **
C**          **                                          **
C** Programmed by: Gary L. Stevens          Date: February 10, 1990 **
C**                      Last Revised: 02/24/91 [gls]          **
C**          **                                          **
C**          MASTER'S THESIS PROJECT          **
C**          Mechanical Engineering Department, San Jose State University **
C**          Fall Quarter 1990 (ME-298) & Spring Quarter 1991 (ME-299) **
C**          Advisor: R.V. Westphal          **
C**-----**
C** Description:          **
C** -----          **
C** This program searches program "THINFILM" output files for the t* **
C** specified by the user. All grid points for this t* are then **
C** written to a user-specified output file so that the h* profile for **
C** this t* can easily be plotted.          **
C*****
C
      CHARACTER*30 DATAFILE,OUTFILE,ANSWER*1
      CHARACTER*1 A,B,C,D,E,F,REV*15,AMPM*2
      DIMENSION H [ALLOCATABLE] (:)
C
C Current revision information.
C
      REV = '02/24/91 [gls]
C
C Assign the banner border variables (standard ASCII characters).
C
      A = CHAR (201)
      B = CHAR (205)
      C = CHAR (187)
      D = CHAR (186)
      E = CHAR (200)
      F = CHAR (188)
C
C Get the current time and date and print the banner.
C
      CALL GETTIM (IHR,IMIN,ISEC,I100TH)
      CALL GETDAT (IYR,IMON,IDAY)
      IF (IHR .GT. 12) THEN
        IHR = IHR-12
        AMPM = 'pm'
      ELSE

```



```

      IF (TIME .LT. TSTAR) GO TO 45
C
C   Write data to output file.
C
      WRITE (02,50) NUMPTS,TIME,(H(I),I=1,NUMPTS)
      50 FORMAT (1X,I5,1PE12.5,/, (1X,1PE12.5))
C
C   Close output file and ask for another t* from user.
C
      CLOSE (02)
C
C   End of file found before finding desired t*.
C
      DEALLOCATE (H)
      REWIND (01)
      GO TO 25
998 WRITE (*,55) TIME
      55 FORMAT (/,5X,'END OF FILE!  POINT NOT FOUND (TMAX = ',1PE12.5,')')
C
C   Finished with data file.  Another case?
C
999 CLOSE (01)
      ANSWER = 'N'
      WRITE (*,60)
      60 FORMAT (/,5X,'Another case? (Y or N [DEFAULT]): ', $)
      READ (*,70) ANSWER
      70 FORMAT (A1)
      IF (ANSWER .EQ. 'y' .OR. ANSWER .EQ. 'Y') GO TO 5
C
C   Finished!  Program stopped.
C
      STOP '      PROGRAM STOPPED!'
      END

```

APPENDIX D

PROGRAM "ANIMATE" LISTING

```

C*****
C**                                PROGRAM:  ANIMATE.FOR                                **
C**                                -----                                **
C**                                **
C** Programmed by:  Gary L. Stevens          Date:  February 10, 1990          **
C**                                Last Revised:  02/24/91  [glS]                **
C**                                **
C**                                MASTER'S THESIS PROJECT                        **
C**      Mechanical Engineering Department, San Jose State University          **
C**      Fall Quarter 1990 (ME-298) & Spring Quarter 1991 (ME-299)            **
C**                                Advisor:  R.V. Westphal                      **
C**-----**
C** Description:                                **
C** -----                                **
C** This program readss program "THINFILM" output files and displays          **
C** the h* profiles graphically on the screen.                                **
C*****
C
C      INCLUDE 'FGRAPH.FI'
C      INCLUDE 'FGRAPH.FD'
C
C      CHARACTER*30 DATAFILE,ANSWER*1,TSTAR*12
C      DIMENSION H [ALLOCATABLE] (:)
C
C      INTEGER*2 IVIDEO,IDUMMY
C      RECORD /RCCOORD/ CURPOS
C      RECORD /XYCOORD/ XY
C
C      Initialize variables.
C
C      L = 1
C      ANSWER = 'N'
C
C      Obtain the THINFILM data file name from the user.
C
C      5 WRITE (*,10)
C      10 FORMAT (/ ,5X,'Enter the THINFILM data file name: ', $)
C      READ (*,20) DATAFILE
C      20 FORMAT (A30)
C      WRITE (*,25)
C      25 FORMAT (/ ,5X,'Enter the time delay scale factor: ', $)
C      READ (*,*) DELAY
C      IF (DELAY .LT. 0.01 .OR. DELAY .GT. 10.0) DELAY = 1.0
C
C      Open the THINFILM data file, read # of points and allocate array.

```

```

C      OPEN (01,FILE=DATAFILE,FORM='BINARY',STATUS='OLD')
      READ (01) NPTS,DUM1,DUM2,DUM3,DUM4
      ALLOCATE (H(NPTS),STAT=ISTAT)
      IF (ISTAT .NE. 0) STOP '      NOT ENOUGH MEMORY!  PROGRAM STOPPED.'

C
C      Set video mode (VGA) and clear screen.
C
      IVIDEO = SETVIDEOMODE ($VRES16COLOR)
      IF (IVIDEO .EQ. 0) STOP 'CANNOT SET VIDEO MODE!  PROGRAM STOPPED.'
      CALL CLEARSCREEN ($GCLEARSCREEN)

C
C      Position cursor and print screen title.
C
      IDUMMY = SETTEXTCOLOR (15)                ! 15 = BRIGHT WHITE
      IDUMMY = SETCOLOR (15)                    ! 15 = BRIGHT WHITE
      IDUMMY = RECTANGLE ($GBORDER,160,5,455,90)
      IDUMMY = RECTANGLE ($GBORDER,158,3,457,92)
      CALL SETTEXTPOSITION (2,22,CURPOS)
      CALL OUTTEXT ('      PROGRAM:  ANIMATE.FOR      ')
      CALL SETTEXTPOSITION (3,22,CURPOS)
      CALL OUTTEXT ('      -----      ')
      CALL SETTEXTPOSITION (4,22,CURPOS)
      CALL OUTTEXT ('Program "THINFILM" output animation')
      CALL SETTEXTPOSITION (5,22,CURPOS)
      CALL OUTTEXT (' By: Gary L. Stevens   02/24/91  ')

C
C      Draw a rectangle to represent the flat surface.
C
      IDUMMY = SETCOLOR (9)                    ! 9 = LIGHT BLUE
      IDUMMY = RECTANGLE ($GFILLINTERIOR,5,400,635,410)

C
C      Define graphics viewport for plotting h* profile.
C
      CALL SETVIEWPORT (20,199,621,399)
      IDUMMY = SETCOLOR (14)                  ! 14 = YELLOW
      IDUMMY = SETTEXTCOLOR (14)              ! 14 = YELLOW

C
C      Read the data file one record at a time.
C
      40 READ (01,END=999) TIME,(H(I),I=1,NPTS)

C
C      Put t* into character format and print to screen.
C
      WRITE (TSTAR,'(1PE12.5)') TIME
      CALL SETTEXTPOSITION (10,33,CURPOS)
      CALL OUTTEXT ('t* = ' // TSTAR)

C
C      Determine scaling factor based on h*(NPTS) for first record only.
C      Clear screen only for subsequent profiles.
C

```

```

        IF (L .EQ. 1) THEN
            XSCALE = 600/FLOAT(NPTS-1)
            YSCALE = H(NPTS)/200
            L = 2
        END IF
C
C Clear the viewport and move to the profile starting position.
C
        CALL CLEARSCREEN ($GVIEWPORT)
        CALL MOVETO (0,200-(H(1)/YSCALE),XY)
C
C Draw profile.
C
        DO 50 I=2,NPTS
            IDUMMY = LINETO (I*XSCALE,200-(H(I)/YSCALE))
        50 CONTINUE
C
C Wait delay before plotting next time step.
C
        DO 60 I=1,DELAY*10000
        60 CONTINUE
        GO TO 40
C
C End of file. Pause before continuing.
C
    999 CALL SETTEXTPOSITION (12,13,CURPOS)
        CALL OUTTEXT ('FINISHED! END OF FILE.')
        PAUSE ' Hit <enter> to continue. . .'
C
C Finished with data file. Another case?
C
        CLOSE (01)
        DEALLOCATE (H)
        ANSWER = 'N'
        IVIDEO = SETVIDEOMODE ($DEFAULTMODE)
        CALL CLEARSCREEN ($GCLEARSCREEN)
        WRITE (*,70)
    70 FORMAT (//,5X,'Another case? (Y or N [DEFAULT]): ', $)
        READ (*,80) ANSWER
    80 FORMAT (A1)
        IF (ANSWER .EQ. 'y' .OR. ANSWER .EQ. 'Y') GO TO 5
C
C Finished! Stop program.
C
        STOP ' PROGRAM STOPPED!'
        END

```

APPENDIX E

ANALYTICAL SOLUTIONS TO TEST CASES #1 AND #2

Test Case #1 [$r^* = \text{constant}$]

If r^* is constant, then $\partial r^* / \partial x^*$ is zero. Equation (8) reduces to:

$$\frac{\partial h^*}{\partial t} = - r^* h^* \frac{\partial h^*}{\partial x^*} \quad (B-1)$$

Using separation of variables, the following solution for $h^*(x^*, t^*)$ is assumed:

$$h^*(x^*, t^*) = X(x^*) T(t^*) \quad (B-2)$$

where $X(x^*)$ is a function of x^* only and $T(t^*)$ is a function of t^* only. Substituting equation (B-2) into (B-1), the following is obtained:

$$X T' = - r^* X T X' T$$

or

$$(T'/T^2) = - r^* X'$$

$$\text{Thus,} \quad (T'/T^2) = \text{constant} = C_1 = - r^* X' \quad (B-3)$$

Solving the left side of equation (B-3):

$$T'/T^2 = C_1$$

$$- 1/T = C_1 t^* + C_2$$

$$\text{or} \quad T = -1/(C_1 t^* + C_2) \quad (B-4)$$

Solving the right side of equation (B-3):

$$- r^* X' = C_1$$

$$\text{or} \quad X = -(C_1/r^*)x^* + C_3 \quad (B-5)$$

$$\text{Thus, } h^*(x^*, t^*) = [-1/(C_1 t^* + C_2)] [-(C_1/\tau^*)x^* + C_3] \quad (\text{B-6})$$

The boundary conditions are:

at $x^* = 0$, $h^* = 0$ for all t^* , so $C_3 = 0$.

at $t^* = 0$, $h^* = 0.001x^*$ (Table 3), so:

$$\begin{aligned} h^* &= (-1/C_2)(-C_1/\tau^*)x^* = [C_1/(C_2\tau^*)]x^* \\ &= (C_4/\tau^*)x^* \end{aligned}$$

$$\text{so } (C_4/\tau^*) = 0.001$$

$$\text{or } C_4 = 0.001 \tau^*$$

$$\text{Thus, } h^*(x^*, t^*) = C_1 x^* / (C_1 \tau^* t^* + C_2 \tau^*) \quad (\text{B-7})$$

Dividing top and bottom by $C_2 \tau^*$ and making use of the definition of $C_1/C_2 = C_4$, equation (B-7) becomes:

$$h^* = \frac{x^*}{(\tau^* t^* + 1000)} \quad (\text{B-8})$$

Test Case #2 [$\partial \tau^* / \partial x^* = \text{constant}$]

For $\tau^* = mx^* + b$ (m and b are arbitrary constants), $\partial \tau^* / \partial x^*$ is constant (equal to m). For this case, equation (8) takes the following form:

$$\frac{\partial h^*}{\partial t^*} = -\tau^* h^* \frac{\partial h^*}{\partial x^*} - \frac{h^{*2}}{2} \frac{\partial \tau^*}{\partial x^*} \quad (\text{B-9})$$

As with Test Case #1, $h^*(x^*, t^*)$ is assumed to be composed of separate functions of x^* and t^* . Substituting equation (B-2) into (B-9), the following is obtained:

$$X T' = -\tau^* X T X' T - \frac{m}{2} X^2 T^2 \quad (\text{B-10})$$

where m has been substituted for $\partial \tau^* / \partial x^*$. Dividing by XT^2 , the following is obtained:

$$\frac{T'}{T^2} = -\tau^* X' - \frac{m}{2} X \quad (\text{B-11})$$

The left side of equation (B-11) is identical to the left side of equation (B-3), so the $T(t^*)$ solution is given by equation (B-4). The right side of equation (B-11) is:

$$-\tau^* X' - \frac{m}{2} X = C_1$$

or

$$X' = \frac{dX}{dx^*} = \left[-C_1 - \frac{m}{2} X \right] \frac{1}{\tau^*} \quad (\text{B-12})$$

Separating variables and noting that $dr^* = m dx^*$, equation (B-12) becomes:

$$\frac{dX}{c_1 + \frac{m}{2} X} = - \frac{1}{m} \frac{dr^*}{r^*} \quad (B-13)$$

Integrating equation (B-13), the following is obtained:

$$X = \frac{2}{m} \left[c_5 r^{*-0.5} - c_1 \right] \quad (B-14)$$

where c_5 is an arbitrary constant resulting from the integration process. Combining equations (B-4) and (B-14):

$$h^*(x^*, t^*) = \frac{2}{m(c_1 t^* + c_2)} \left[c_1 - c_5 r^{*-0.5} \right] \quad (B-15)$$

Now, at $x^* = 0$, $h^* = 0$ and $r^* = b$, so $c_1 = c_5 b^{-0.5}$. Substituting this into equation (B-15) and simplifying:

$$h^*(x^*, t^*) = \frac{2}{mt^* + c_6} \left[1 - \left[\frac{r^*}{b} \right]^{-0.5} \right] \quad (B-16)$$

where $c_6 = \frac{c_2}{c_5 b^{-0.5}}$

But c_6 is an arbitrary initial time, t_0 . A reference time frame can always be chosen such that t_0 is zero. Thus,

equation (B-16) reduces finally to:

$$h^*(x^*, t^*) = \frac{2}{mt^*} \left[1 - \left[\frac{\tau^*}{b} \right]^{-0.5} \right] \quad (B-17)$$

But $m = \tau_L^* - \tau_0^*$, where τ_L^* and τ_0^* are the values of τ^* at $x^* = L$ and $x^* = 0$, respectively. Thus, equation (B-17) takes the following final form:

$$h^*(x^*, t^*) = \frac{2}{(\tau_L^* - \tau_0^*) t^*} \left[1 - \left[\frac{\tau^*}{\tau_0^*} \right]^{-0.5} \right] \quad (B-17)$$

APPENDIX F

PROGRAM "TAUGEN" LISTING

```

C*****
C**                                PROGRAM:  TAUGEN.FOR                                **
C**                                -----                                **
C**                                **
C**  Programmed by:  Gary L. Stevens          Date:  January 29, 1991          **
C**                   Last Revised:  01/29/91  [gls]                          **
C**                                **
C**                                MASTER'S THESIS PROJECT                        **
C**      Mechanical Engineering Department, San Jose State University          **
C**      Fall Quarter 1990 (ME-298) & Spring Quarter 1991 (ME-299)            **
C**      Advisor:  R.V. Westphal                                              **
C**-----**
C**  Description:                                                              **
C**                                **
C**  This program provides interpolated values of TAU based on the            **
C**  simulation data provided by R.V. Westphal.  The input simulation          **
C**  data is read from a file, linearly interpolated to the desired          **
C**  number of points (<= 5000), and the results are written to a user      **
C**  specified output file.                                                  **
C*****
C
      CHARACTER*40 TITLE,INFILE*20,OUTFILE*20
      DIMENSION TAU(129),TAUST(5000)
C
C  Obtain user input.
C
      WRITE (*,10)
10  FORMAT (//,15X,'*****',
&        /,15X,'** PROGRAM:  TAUGEN.FOR **',
&        /,15X,'*****',
&        /,15X,'Shear simulation data interpolator.',
&        /,15X,'Programmed by Gary L. Stevens  01/29/91',
&        /,15X,'Last Revised: 01/28/91  [gls]',
&        /,15X,'=====')
      WRITE (*,20)
20  FORMAT (//,10X,'Enter the interpolation factor, m: ',)$)
      READ (*,*) M
      WRITE (*,30)
30  FORMAT (10X,'Enter the output file name      : ',)$)
      READ (*,40) OUTFILE
40  FORMAT (A20)
C
C  Initialize variables.
C
      NPTS = 128                      ! Number of points of simulation data.

```

```

      TAUM = 183.9                ! Mean value of TAU from simulation data.
      INFILE = 'GETTAUX.OUT'      ! Input file name.
C
C  Open input and output files.
C
      OPEN (01,FILE=INFILE,STATUS='OLD')
      OPEN (02,FILE=OUTFILE,STATUS='UNKNOWN')
C
C  Read header line from input file and print to screen.
C
      READ (01,50) TITLE
50  FORMAT (A40)
      WRITE (*,60) TITLE
60  FORMAT (//,10X,'Simulation data title: ',A40)
C
C  Read shear data from input file.
C
      READ (01,*) (TAU(I), I=1,NPTS)
      TAU(NPTS+1) = TAU(1)
C
C  Interpolate data.
C
      DO 70 J=1,NPTS
      TAUST((J-1)*M+1) = TAU(J)/TAUM
      DO 80 L=1,(M-1)
      TAUST((J-1)*M+1+L) = (TAU(J) + L*((TAU(J+1)-TAU(J))/M))/TAUM
80  CONTINUE
70  CONTINUE
C
C  Write results to output file.
C
      WRITE (02,90) M*NPTS
90  FORMAT (2X,I4)
      WRITE (02,100) (TAUST(I), I=1,M*NPTS)
100 FORMAT ( (2X,1PE12.5) )
C
C  Finished! Stop program.
C
      WRITE (*,110) NPTS,M*NPTS,M
110 FORMAT (//,10X,'Finished! ',I4,' points interpolated to ', I4,
&' points (m = ',I2,').',/)
      STOP '          PROGRAM STOPPED!'
      END

```

APPENDIX G

PROGRAM "SLOPE" LISTING

```

C*****
C**          PROGRAM:  SLOPE.FOR          **
C**          -----                    **
C**          **
C** Programmed by: Gary L. Stevens      Date: March 21, 1991  **
C**                      Last Revised: 03/21/91 [glS]          **
C**          **
C**          MASTER'S THESIS PROJECT      **
C**          Mechanical Engineering Department, San Jose State University **
C**          Fall Quarter 1990 (ME-298) & Spring Quarter 1991, (ME-299) **
C**          Advisor: R.V. Westphal      **
C**-----**
C** Description:                          **
C** -----                          **
C** This program calculates the slope of a least-squares, best-fit **
C** straight line through the first twenty grid points contained in **
C** program "THINFILM" output. The line is forced through (0,0).  **
C** This data is used for comparing the sample calculation results to **
C** the results of Test Case #1.        **
C*****
C
C          CHARACTER*30 DATAFILE,OUTFILE
C          DIMENSION H(500)
C
C Obtain the THINFILM data file name from the user.
C
C          WRITE (*,10)
C 10 FORMAT (/ ,5X, 'Enter the THINFILM output data file name      : ', $)
C          READ (*,20) DATAFILE
C 20 FORMAT (A30)
C
C Open the THINFILM data file.
C
C          OPEN (01, FILE=DATAFILE, FORM='BINARY', STATUS='OLD')
C
C Obtain the output file name from the user.
C
C          WRITE (*,30)
C 30 FORMAT (/ ,5X, 'Enter the output file name                      : ', $)
C          READ (*,20) OUTFILE
C
C Open the output data file and read # of points.
C
C          OPEN (02, FILE=OUTFILE, STATUS='UNKNOWN')
C          READ (01) NUMPTS, DUM1, DUM2, DUM3, DUM4

```

```

C
C Read data (8,000 points maximum).
C
    DO 40 I=1,8000
    READ (01,END=999) TIME,(H(J),J=1,NUMPTS)
C
C Initialize variables.
C
    PROD = 0.
    SQUARE = 0.
C
C For each t* step, compute slope for first twenty grid points.
C SLOPE1 = Slope for sample calculation.
C SLOPE2 = Slope for Test Case #1 (from analytical solution).
C
    DO 45 K=0,20
    X      = K*(1./(NUMPTS-1))
    PROD   = PROD + X*H(K+1)
    SQUARE = SQUARE + X*X
45 CONTINUE
    SLOPE1 = PROD/SQUARE
    SLOPE2 = 1./(TIME+1000.)
C
C Write data to output file.
C
    WRITE (02,50) TIME,SLOPE1,SLOPE2
50 FORMAT (1X,1PE12.5,1X,1PE12.5,1X,1PE12.5)
40 CONTINUE
C
C Deallocate array and close output files.
C
999 CLOSE (01)
    CLOSE (02)
C
C Finished! Program stopped.
C
    STOP ' PROGRAM STOPPED!'
    END

```