PROJECT ADMINISTRATION DATA SHEET

Project No. E-16-606
Project Director: Dr. S. G. Lekoudis
School/Lab Aerospace Engineering
Sponsor: Lockheed-Georgia Co., Marietta, GA

Type Agreement: Purchase Order # CX86876
Award Period: From 6/1/81 To 2/28/82 (Performance) (Reports)
Sponsor Amount: $30,000
Cost Sharing: N/A
Contracted through: GTRI/EDITT
Title: Studies in Three-Dimensional Turbulent Boundary Layer Separation from Smooth Surfaces

ADMINSITRATIVE DATA

OCA CONTACT Leamon R. Scott
1) Sponsor Technical Contact: Dr. H. Plumblee, Dept. 72-11, Zone 403
   Lockheed-Georgia Co., Marietta, GA 30063

2) Sponsor Admin./Contractual Contact: Bill Britton, Zone 383; Lockheed-Georgia Co., Marietta, GA 30063; phone 424-5250

Reports: See Deliverable Schedule
Security Classification: N/A
Defense Priority Rating: N/A

RESTRICTIONS

See Attached Supplemental Information Sheet for Additional Requirements

Travel: Foreign travel must have prior approval - Contact OCA in each case. Domestic travel requires sponsor approval where total will exceed greater of $500 or 125% of approved proposal budget category.

Equipment: Title vests with none proposed

COMMENTS:

COPIES TO:

Administrative Coordinator
Research Property Management
Accounting Office
Procurement/EES Supply Services

Research Security Services
Reports Coordinator (OCA)
Legal Services (OCA)
Library, Technical Reports

EES Research Public Relation
Project File (OCA)
Other: _______________________________
SPONSORED PROJECT TERMINATION SHEET

Date: September 23, 1983

Project Title: "Studies in Three Dimensional Turbulent Boundary Layer Separation From Smooth Surfaces"
Project No: E-16-606
Project Director: Dr. S. G. Lekoudis
Sponsor: Lockheed-Georgia Company

Effective Termination Date: 2/28/82
Clearance of Accounting Charges: 2/28/82
Grant/Contract Closeout Actions Remaining:
None

☐ Final Invoice and Closing Documents
☐ Final Fiscal Report
☐ Final Report of Inventions
☐ Govt. Property Inventory & Related Certificate
☐ Classified Material Certificate
☐ Other

Assigned to: Aerospace Engineering

COPIES TO:
Administrative Coordinator
Research Property Management
Accounting
Procurement/EES Supply Services

Research Security Services
Reports Coordinator (OCA)
Legal Services (OCA)
Library

EES Public Relations (2)
Computer Input
Project File
Other
Contract No. E-16-606

"STUDIES IN THREE-DIMENSIONAL TURBULENT BOUNDARY LAYER SEPARATION FROM SMOOTH SURFACES"

Progress Report

for the period June 1, 1981 - October 31, 1981

to

Lockheed-Georgia Company
Dept. 72-11, Zone 403
Marietta, Georgia 30063

by

S.G. Lekoudis
School of Aerospace Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332
# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>1</td>
</tr>
<tr>
<td>Task &quot;0&quot;</td>
<td>2</td>
</tr>
<tr>
<td>Task 1</td>
<td>22</td>
</tr>
<tr>
<td>Task 2</td>
<td>28</td>
</tr>
<tr>
<td>Task 3</td>
<td>29</td>
</tr>
<tr>
<td>Appendix A</td>
<td>30</td>
</tr>
<tr>
<td>References</td>
<td>43</td>
</tr>
</tbody>
</table>
1. INTRODUCTION

This progress report summarizes the work done under the contract E-16-606 from the Lockheed-Georgia Company to the Georgia Institute of Technology, during the time period 6/1/81 - 10/31/81. This report is submitted according to Reference 1.

All tasks are examined. However there is progress reported on task "0" which is not included in Reference 1. The reason is that the work reported under task "0" was initially planned to be done under a consulting agreement between S.G. Lekoudis and the Lockheed-Georgia Company. Because of difficulties in distinguishing the efforts, and the relation of the work in task "0" to separation, all efforts were performed under this contract and are reported in this document.
This task involves the coupling of two programs. The first is a code that uses the most complete viscous/inviscid iteration technique available, to compute viscous transonic flows over single airfoils. The method, developed by the research department of Grumman Aircraft, is described in Reference 2 and in paper No. 10 of Reference 3. The second code solves the linearized two-dimensional Navier-Stokes equations for shock/boundary layer interactions. The method, developed by G. Inger and his co-workers, is described in paper No. 18 of Reference 3 and in Reference 4.

Before explaining the coupling of the two codes, some justification for the effort is appropriate. It is known (Reference 3) that viscous effects dominate the performance of supercritical airfoils. Navier-Stokes solutions for flows around single airfoils, at interesting Reynolds numbers, are very expensive. Moreover grid refinement studies that verify convergence, as far as truncation errors, are not always available. Viscous/inviscid coupling schemes also have their shortcomings. Most of them neglect pressure gradient effects in the cross stream direction. There are two areas of the flowfield around single airfoils where these pressure gradients are known to be important. One is around the trailing edge, as shown by Melnik and his co-workers (Reference 2). The other is the region of shock/boundary layer interaction (Papers No. 4 and 15 of Reference 3).

Normal pressure gradient effects close to the trailing edge were accounted for in a code developed over a number of years at Grumman
Aircraft (Reference 2). Similar processes for shock/boundary layer interactions were developed by Stanewsky et al (Paper 4 of Reference 3). In this task, a computer program was developed, using both procedures, that resulted in the most sophisticated viscous/inviscid coupling procedure for computing transonic flows over single airfoils that exists. The method maintains the attractive features of viscous/inviscid coupling which are the good numerical resolution of separately computed regions of the flow, and the economy of the calculations.

Solutions of the linearized Navier-Stokes equations, for normal shocks interacting with turbulent boundary layers in transonic flow, have been obtained by Inger and his co-workers (Paper No. 18 of Reference 3). The obliqueness of the shock, for flow around airfoils (see Figure 1a) is empirically accounted for by evaluating the angle \( \beta \) from

\[
\beta = 90.0^\circ - 37.8 \sqrt{M'} - 1.0
\]

where \( M' \) is the Mach number computed from inviscid theory, at the surface of the airfoil, before the shock. The incoming Mach number that enters the calculations for the interaction is then \( M = M' \sin \beta \). For the cases investigated \( \beta \) is around 74 degrees. The subscripts \( b \), \( s \) and \( a \) denote before the interaction, at the root of the shock, and after the interaction. Assuming that the "incompressible" shape factor is \( H = (H_i)_b \), the incoming Reynolds number is \( R = (R_\delta)_b \), the pre-shock Mach number is \( M \), and \( R = \log_{10} (R_\delta)_b \), Inger's analysis gives:

\[
(c_f)_a = (0.252 R + 3.4273 - 5.5 M + 3.15 H - H^2)(c_f)_b
\]
\[ l_{\text{up}} = (9.4 H - 108.0 M + 40 M^2 + 61.124)(\delta^*_b) \] \hspace{1cm} (3)

\[ l_{\text{dn}} = (0.25 R - 6,414.8 + 8,758.4 H - 2,756.9 H^2 + (10,639 - 14,659 H + 4,686 H^2)M \]
\[ + (-4,439 + 6,157 H - 1,992 H^2)M(\delta^*_b) \] \hspace{1cm} (4)

\[ (c_f)_a = [ (4,568 - 6,079 H + 2,061 H^2)R + (2,085.47 - 2,695.05 H + 874.1 H^2)M + (-931.01 + 1,201.8 H - 389.7 H^2)M^2 - 1,188.548 + 1,539.911 H - 500.049 H^2 ]c(c_f)_b \] \hspace{1cm} (5)

\[ \delta^*_a = \delta^*_b \left[ 1 + (5.17 + 8.65*(H - 1.3))*(M - 1.0)* \right. \]
\[ \left. (1.11 * \tanh (R - 2.35)) \right] \] \hspace{1cm} (6)

Schematic variations of \( \delta^* \) and \( c_f \) in the interaction region are shown in Figure 1b. Also \( l_{\text{up}} = x_a - x_b \) and \( l_{\text{dn}} = x_d - x_s \). Equations (2) to (6) were obtained by curve fitting numerical solutions of the linearized Navier-Stokes equations for normal shocks interacting with unseparated turbulent boundary layers.

The problem that arises in the coupling of this procedure is that \( l_{\text{up}} \) and \( l_{\text{dn}} \) are of the order of \( 10^{-3} \) to \( 10^{-2} \) of the airfoil chord, and thus, are sometimes smaller than the spacing of the denser grid used in the
inviscid flow computations. Noting that further grid refinement would not change the width of the shock computed from the inviscid code, and to avoid this problem, we used a method which is justified by the asymptotic approach to equilibrium (at \( x_b \) and \( x_a \)) of the flow variables, according to the interaction theory. The incoming boundary layer profile at \( b \) is determined by checking the slope of \( M'(x) \) from the inviscid calculations. This location is denoted by the grid point \( N_b \) in Figure 1c. The location \( N_a \) is determined by checking the slope \( M' = M'(x) \) after the last supersonic point. Then the shock location \( N_s \) is equal to \( \frac{1}{2}(N_b + N_a) \). This procedure, locates the interaction "box" shown in Figure 1b at the center of the numerically smeared shock area given by the potential flow calculations. If \( \ell_{up} \) is smaller than \( x(N_s) - x(N_b) \) the boundary layer properties are kept constant till \( x(N_s) - \ell_{up} \), and equal to the ones at \( N_b \). The boundary layer calculations are initiated after \( x(N_s) + \ell_{dn} \). To the author's knowledge, no Navier-Stokes solution exists with dense enough grid to capture the details of the interaction, as provided by the analysis used here.

Results for the RAE2829 airfoil (Reference 5), using the above procedure, are compared with viscous/inviscid coupling where boundary layer theory is used to march under the shock, as developed by Melnik and his co-workers (Reference 2). Figures 2 to 5 show computed upper surface displacement thicknesses. Figures 6 to 9 show the corresponding skin friction and Figures 10 to 13 the \( \frac{C_p}{\alpha} \) distribution.

The results are summarized in the Figures 14 and 15 where \( C_L - \alpha \) curves are plotted.

From these results, it seems that the interaction is responsible for a loss in lift. Also the pressure distribution changes only close to the
shock, as compared with the original code (Reference 2). Thus the capability of the original code to accurately predict measured pressure distributions is maintained in the new program (Reference 6). Moreover the computed shock is "crisper" and moves slightly ahead, as compared with the one computed using simple boundary layer theory underneath it. At the time of the writing of this report another airfoil, the LG4-512 is being used to evaluate the developed method and comparisons with experiments are being done.

It is recommended that the method be used to study the initiation of shock/induced separation. Although the theory is not valid at separation, it should give a good indication when it is about to occur, because of its ability to accurately compute pressure distributions. The computing times are not affected by the interaction and they are almost identical to the original code (Reference 6).
Figure 1 Schematic of the Flowfield in the Shock/Boundary Layer Interaction Region.
DISPLACEMENT THICKNESS

- No interaction
- Interaction

RAE2822

\[ M_{\infty} = 0.73 \]
\[ \alpha = 1.8^\circ \]
\[ R_e = 2 \times 10^6 \]
DISPLACEMENT THICKNESS

- No interaction
- Interaction

RAK2822
M∞ = 0.73
α = 2.0°
Rc = 2 x 10^6
DISPLACEMENT THICKNESS

- No interaction
- Interaction

RAE 2872

\( M_\infty = 0.73 \)

\( \alpha = 2.2^\circ \)

\( R_c = 2 \times 10^6 \)
DISPLACEMENT THICKNESS

- No interaction
- Interaction

\[ R_{e2} = 2 \times 10^6 \]
\[ M_{\infty} = 0.73 \]
\[ \alpha = 2.4^\circ \]
RAE2822
$Re_c = 0.73$
$\alpha = 1.8^\circ$
$Re_c = 2 \times 10^6$

- No interaction
- Interaction
- No interaction

- Interaction

RAE2822

\( M_\infty = 0.73 \)
\( \alpha = 2.0^\circ \)
\( R_e = 2 \times 10^6 \)
SKIN FRICTION

- No interaction
- Interaction

RAF.2822

\( M = 0.73 \)
\( \alpha = 2.2^\circ \)
\( \kappa = 2 \times 10^6 \)
SKIN FRICTION

- No interaction
- Interaction

RAE2822

\( M_o = 0.73 \)
\( \alpha = 2.4^\circ \)
\( R_e = 2 \times 10^6 \)
PRESSES DISTRIBUTION

- No interaction
  □ Interaction

[RCAE2822]

\[ M_{\infty} = 0.73 \]
\[ \alpha = 1.8^\circ \]
\[ R_c = 2 \times 10^6 \]
PRESSES DISTRIBUTION

- No interaction
- Interaction

RAE 2822

M_\infty = 0.73
\alpha = 2.0^\circ
R_c = 2 \times 10^6
PRES residue Distribution

- No interaction
- Interaction

RAE2322

\[ M_\infty = 0.73 \]
\[ \alpha = 2.2^\circ \]
\[ R_c = 2 \times 10^6 \]
- No interaction
• Interaction

RAE2822

\( \frac{M}{\infty} = 0.73 \)
\( \alpha = 2.4^\circ \)

\( R_c = 2 \times 10^6 \)
RAE2822
$M_\infty = 0.73$

$R_c = 2 \times 10^6$

- No interaction
- Interaction

Figure 14
RAE.2822

\[ M_{\infty} = 0.73 \]

\[ k_c = 6 \times 10^6 \]

- No interaction
- Interaction

**Figure 15**
The objective of this task is to develop a procedure that can be used to compute three-dimensional boundary layer flows close to separation. Use of the thin shear layer equations at separation is not possible in the direct mode, i.e., when the external pressure distribution is prescribed. The reason is the singularity of the thin shear layer equations at separation, that makes the numerical integration of the equations impossible past the location of the separation. In this discussion, by separation we mean catastrophic separation and not recirculating bubbles.

Experiments show that normal stresses are important close to separation (Reference 7). Thus, the assumption that pressure gradients normal to the wall are negligible, used in the thin shear layer equations, might not be a good approximation close to separation. For flows at interesting Reynolds numbers, the subject seems controversial because, for some cases, reasonable agreement was obtained with viscous/inviscid coupling schemes that use simple boundary layer theory (Paper No. 26 of Reference 3 and Reference 8). For some other uses, inclusion of normal pressure gradients seemed necessary (Paper No. 30 of Reference 3). Some comments on the subject are made at the end of the discussion about this task.

Under this task a procedure was developed, that combines the capability of computing boundary layers past the separation point with the ability to account for pressure gradient effects normal to the wall. A description of this procedure for two-dimensional flows follows.
It is known (Reference 9, 10 and 11) that the boundary layer singularity is removable when the equations are being solved in the so-called inverse mode. In this mode, the displacement thickness $\delta^*$ is prescribed and the pressure gradient is being computed. In this way, calculations can proceed past the location of separation. Also, procedures have developed that account for nonnegligible pressure gradients normal to the wall (Reference 12). It seems reasonable to combine the two methods into one, and have a procedure that allows calculations with normal pressure gradient effects through separation.

Assume a two-dimensional, boundary layer flow growing on a wall ($x$-coordinate), with a prescribed displacement thickness $\delta^*(x)$. Let $y$ be the coordinate normal to the wall, $u$ and $v$ the velocity components in the $x$ and $y$ direction respectively, and $p$ is the pressure. Also, assume that at each streamwise station $x$, $p = c(x) f_x(y)$ and at the initial station, a velocity profile is available. The following steps would do the job.

1) Calculate the boundary layer at the next streamwise step using the inverse mode, with $\delta^*(x)$ as given, but with $\partial p/\partial x$ partially "known" function of $y$. In this process, $C(x)$ is obtained at the next $x$ station, together with the external freestream velocity.

2) Calculate $\partial p/\partial y$ from the $y$-momentum equation at the next station, using the velocities computed. Thus, obtain a new $f_x(y)$ at the next station.

3) Repeat steps 1-2 for all streamwise stations. If reverse flow is encountered, its ok.

4) Repeat steps 1-3 using the new "eigenfunctions" $f_x(y)$ for the pressure and use central differences for $\partial p/\partial x$, until convergence is obtained.
Notice that the old values of $p$ are used through one sweep in the $x$-direction. This is because it was found (Reference 12 where the direct mode was used) that this way the process converged. This procedure of updating the pressure corresponds to a Jakobi iteration, instead of a Gauss-Seidel iteration.

The procedure described can use any of the existing turbulence models. If coupling with an inviscid code is required, it can be done by iteratively equating the boundary layer edge velocities computed by the procedure, with the ones from the inviscid code that "sees" an equivalent body, displaced by $\delta^*$. In order to check this procedure, two boundary layer programs were combined. The first solves the two-dimensional incompressible laminar and turbulent boundary layer equations for arbitrary pressure gradients in the direct mode. The second is a boundary layer program that solves the same equations in the inverse mode (Reference 18). During checkout of the second program, mistakes have been found in the code and have been corrected. A list of the combined program is provided in Appendix A, together with some explanation of what the subroutines do. The input parameters are:

1) Number of streamwise stations (NXT)
2) Station where transition from laminar to turbulent flow occurs (NTR)
3) Station where the program switches from direct to inverse mode (INV)
4) Step size of the grid normal to the wall at the first step ($\Delta_{n1}$)
5) Factor for the geometric growth of the grid normal to the wall (VGP)
6) Freestream velocity (UREF)
7) Reynolds number based on the coordinate of the last streamwise station
8) The coordinates of the streamwise stations. Notice that $x(NXT) = 1$.
9) The pressure coefficient $C_p$ at the first INV-1 stations.
10) The displacement thickness $\delta^*$ at the last NXT-INV stations.

The program uses an eddy-viscosity model for the turbulence calculations. The program runs in both the direct and the inverse mode, and separated laminar profiles have been obtained. However there are difficulties in converging with separated turbulent profiles and work is being done to overcome the problem. The next step will be to code the described method using the code described in Appendix A as the base.

The extension of the procedure to three-dimensions is, in principle, straightforward. In three-dimensions, one of the two separation patterns may exist. The first is the closed pattern (Figure 16a) where streamlines coming from the stagnation region never reach the region with backflow. The second is the open pattern (Figure 16b). Both have been discussed in the literature (Reference 13). Remembering that the ultimate objective of this effort is to compute the loads on a realistic configuration, using viscous/inviscid interaction, at high Reynolds numbers, we examine these patterns separately.

Computing through the separation line of the closed type will require the solution of the three-dimensional boundary layer equations in the inverse mode. Such solutions have been generated recently in France (Reference 14) using integral techniques. Sophisticated turbulence models will require finite-difference solutions of the boundary layer equations using the inverse mode. Such solutions have not appeared yet. The same solutions are required, if the scheme described for the two-dimensional problem shows that pressure gradients normal to the wall have a significant effect in the location of separation. However if the pressure gradient in
the cross-stream direction turns out to be of minor significance, a viscous/inviscid coupling can proceed with a closed separation line predicted by simple boundary layer theory. Such a calculation is possible and is the simplest attempt to compute the flowfield around a body with massive separation. It might be that boundary layer separation from wing surfaces at high Reynolds numbers is such a type of separation. However, for the case of afterbodies, of equal or maybe of more importance, is the case of the open separation.

Computing a separation line of the open type could be accomplished with a use of three-dimensional boundary layer theory in the direct mode, plus the technique described previously for the two-dimensional case. In this type of separation a vortex sheet would spring from the separation line. Experiments (References 15, 16) indicate that counterrotating streamwise vortices might be responsible for the vortex sheet that emanates from the smooth surface. Thus, while the flow has a large streamwise component of the velocity, without any indication of backflow, crossflow of opposite signs at the two sides of the vortex generates the open separation. To apply the procedure described before one would use the equivalent in three-dimensions of the work reported in Reference 12.

Assume an external pressure distribution \( p(x,z) \) given, where \( x \) and \( z \) define the surface of the developing boundary layer. In a viscous/inviscid coupling procedure, this would correspond to the state of the iterative procedure where the inviscid flow has just been recomputed. The following steps would do the job, with an assumption of \( p(x,y,z) \) that matches the given pressure distribution at the boundary layer edge.

1) Calculate the boundary layer at the next streamwise plane, but with \( \partial p/\partial x \) and \( \partial p/\partial z \) "known" functions of \( y \).
2) Using the computed velocities, compute the pressure from the y-momentum equation starting with the known pressure at the boundary layer edge.

3) Repeat steps 1 and 2 for all the streamwise planes.

4) Repeat steps 1-3 using the newly computed pressures, until convergence is achieved.

From the above discussion, it is obvious that the capabilities of simple boundary layer theory in predicting the location of separation for three-dimensional, high Reynolds number, turbulent flows has not been really investigated in any depth. In this task a technique was developed that simply combines two previously used procedures, calculations in the inverse mode and incorporation of the y-momentum equation in the calculations, into a way of computing two or three-dimensional boundary layer flows past the separation.
4. TASK 2

Although work on this task has not started, some comments are appropriate. S. Ragab of Lockheed-Georgia has developed a three-dimensional boundary layer code for laminar flows around an ellipsoid of revolution. Because of the care and thorough testing of the numerics of this program, it is proposed that the new code will be used for this task. Thus the code developed by Nash and Scraggs and mentioned in Reference 1 will not be used. Dr. Ragab is continuing his work on the code with the incorporation of an eddy viscosity model.
5. TASK 3

Again, although work on this task has not started, some comments are appropriate. Work on the potential flow with free vortices is continuing at NSRDC (Reference 17). In order to obtain the computer code (Reference 1), Lockheed might have to follow a procedure as a defense contractor, because the code is not releaseable otherwise. This problem is being investigated.

Acknowledgements

The many hours of discussions about separation with Dr. S. Ragab of Lockheed-Georgia, are appreciated very much.
6. APPENDIX A

Direct-Inverse Two-Dimensional Incompressible Boundary Layer Program for Laminar and Turbulent Flows

<table>
<thead>
<tr>
<th>INVERSE (Main Routine)</th>
<th>Performs the downstream marching and the iteration process</th>
</tr>
</thead>
<tbody>
<tr>
<td>INPUT</td>
<td>Reads input data</td>
</tr>
<tr>
<td>IVPL</td>
<td>Initiates a profile at the first station</td>
</tr>
<tr>
<td>GRID</td>
<td>Defines the grid normal to the wall</td>
</tr>
<tr>
<td>EDDY I</td>
<td>Calculates the Reynolds stresses when the program is in the inverse mode</td>
</tr>
<tr>
<td>EDDY</td>
<td>Calculates the Reynolds stresses when the program is in the direct mode</td>
</tr>
<tr>
<td>CMOM</td>
<td>Computes the coefficients of the momentum equation when the program is in the direct mode</td>
</tr>
<tr>
<td>ICONZ1</td>
<td>Computes the coefficients of the momentum equation when the program is in the inverse mode</td>
</tr>
<tr>
<td>SOLV4</td>
<td>Inverts the block-tridiagonal matrix of the resulting finite-difference formulation of the boundary layer equations</td>
</tr>
<tr>
<td>OUTPUT</td>
<td>Prints the output quantities</td>
</tr>
</tbody>
</table>
PROGRAM INVERSE

PROGRAM INVERSE (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
COMMON /BLCSS/NP,NX,XP,XP,J,INV,ETA(1),DET(1).A(1),ETA0(1),USO(1),CIAA,ETA(61),A(61),
1 ETA(61),USO(61),GAMMA1,GAMMA2,URRE,REYN
COMMON /BLCSS/X(60),Z(60),P1(60),P2(60),CL(60),RX(60),CF,P1
1 UC,TC,ETA(61)
COMMON /BLCSS/XP(61),U(61),V(61),W(61)

ITMAX=50
MA=0
NX=1
CALL INPUT
CALL GRID
CALL INV
P2(1)=0.0
P1(1)=0.5
STEP=UE(INV-1)/URRE
SC=SOR(TX(INV-1)/(UE(INV-1)/URRE))
DO 10 J=1,NP
WRITE(6,9100)NX,XIX
NST=0
IF(NX.GT.1)CEL(NX)=0.5*(X(NX)+X(NX-1))/(X(NX)-X(NX-1))
IF(NX.GE.INV)UE(NX)=UE(NX-1)
IF(NX.GE.INV)GO TO 60
P1P=P1(NX)+CEL(NX)
P2P=P2(NX)+CEL(NX)
NST=NST+1
IF(NX.LT.INV)GO TO 30
H1=X(NX-1)-X(NX-2)
H2=X(NX)-X(NX-1)
DUDS=H1*X(NX-1)*UE(NX)*H2*2*UE(NX-2)-(H1+H2)**2
P2(NX)=X(NX)*DUDS/UE(HY)
P1(NX)=0.5*P2(NX)+1.0
P1P=P1(NX)+CEL(NX)
P2P=P2(NX)+CEL(NX)
IF(NX.NE.INV)GO TO 30
IF(IT.NE.1)OR.HA.NE.1)I=0) TO 37
DO 40 J=1,NP
F(J,1)=F(J,1)*STEP
F(J,2)=F(J,2)*STEP
U(J,1)=U(J,1)*STEP
U(J,2)=U(J,2)*STEP
V(J,1)=V(J,1)*STEP
V(J,2)=V(J,2)*STEP
M(J,1)=W(J,1)
M(J,2)=W(J,1)
CONTINUE
DET(1)=ETA(1)*STEP
DO 45 J=2,NX
DET(J)=ETA(J-1)*VS
A(J)=0.5*DET(J-1)
ETA(J)=ETA(J-1)+D*<A(J-1)
RX(NX)=UE(NX)*X(NX)/G
30 CONTINUE
IF(IT.LE. ITMAX) GO TO 70

PROGRAM INVERSE 74/74 OPT=1 PMAMP

WRITE(6,2500)
GO TO 90
60 IF(NX.GE.INV) GO TO 70
IF(NX.GE.NTR) CALL CMOM
GO TO 2
65 IF(NX.GE.NTR) CALL EDLY
CALL ICONZ1
2 CALL SOLV4
51 IF(NX.GE.NTR) GO TO 62
IF(ABS(DLEV(1)).GT.1.1E-3) GO TO 75
GO TO 75
70 IF(ABS(DLEV(1)/V(1,2)+0.5*DLEV(1))).GT.0.02) GO TO 60
75 IF(NX.EQ.1) GO TO 93
IF(NP.EQ.61) GO TO 92
IF(ABS(V(NP,2)).LE.1.0E-3) GO TO 92
CALL GROWTH
75 IF(1) IF(NX.GE.INV) FA=1
GO TO 60
90 CALL OUTPUT
GO TO 25
80 2500 FORMAT(*1*,16X,25ITGRATIONS EXCEEDED ITMAX)
910C FORMAT(*1*,4H1X*,13,5X,3HX =,F15.3)
END
SUBROUTINE INPUT
COMMON /BLCO/ NP, NX, NXT, MTR, INV, ELTA, VGP, GNV, DETA(61), A(61),
1 ETA(1), ESD(1), RAYA1, XAHA1, UREF, REYN
COMMON /BLCC/ X(60), UE(60), P1(60), P2(60), CEI(60), RX(60), CF, P1!
1 DENT PRT, TET(160)
DIMENSION CP(60)
UNIT 6
READ (5, 8200) NXT, MTR, INV, D: T A1, VGP, UREF, REYN
NXM=INV-1
DO 2 I=1, NXT
READ (5, 8200) X(I)
2 READ (5, 8200) CP(I)
DO 4 I=INV, NXM
3 READ (5, 8200) CP(I)
3 READ (5, 8200) CP(I)
4 READ (5, 8200) CP(I)
4 READ (5, 8200) CP(I)
WRITE(6, 9000) NXT, MTR, INV, ETA, DETA, VGP, REYN
5 UE(J)=UREF*SORT(1.0-CP(J))
5 UE(J)=UREF*SORT(1.0-CP(J))
DO 80 I=2, NXM
7 IF(I.EQ.NXM) GO TO 80
A1= (X(I)-X(I-1))* (X(I+1)-X(I-1))
A2= (X(I)-X(I-1))* (X(I+1)-X(I))
A3= (X(I+1)-X(I))* (X(I)+X(I-1))
DUDS=-(X(I+1)-X(I))/A1*UE(I-1) + (X(I+1)-2.0*X(I)+X(I-1))/ A2*UE(I)
A2=UE(I) + (X(I)-X(I-1))/A3*UE(I+1)
A3= (X(I)-X(I-1))* (X(I)-X(I-2))
DO 70 I=1, NXM
50 A1= (X(I+2)-X(I-2))* (X(I)-X(I-1))
A2= (X(I)-X(I-1))* (X(I)-X(I-2))
DO 70 I=1, NXM
70 CONTINUE
P2(I)=C1/UE(I)*DUDS
DO 90 J=1, NXM
80 WRITE(6, 91) J, X(J), UE(J)
80 WRITE(6, 91) J, X(J), UE(J)
90 RETURN
900 FORMAT(3(5, 1F10.0))
920 FORMAT(F24.10)
900 FORMAT(1H3, 6NXT =, 13, 16NXT =, 13, 15NVT =, 6NXT =, 13,
1 1H, 6NXT =, 13, 14*5, 5, 5*6, 5*)
1 1H, 6NXT =, 13, 14*5, 5, 5*6, 5*)
9 FORMAT(10X, *J=*, 13, , X=*, X=*, X=*)
9 FORMAT(10X, *J=*, 13, , X=*, X=*, X=*)
10 FORMAT(10X, *J=*, 13, , X=*, X=*, X=*, X=*)
10 FORMAT(10X, *J=*, 13, , X=*, X=*, X=*, X=*)
END
SUBROUTINE GROWTH
COMMON /BLOP/ NP, NX, NXT, NTK, INV, ETA, VGP, CNV, DETAE(61), A(61)
1
COMMON /BLCK/ ETA(61), DSU, GAMMA1, GAMMA2, UREF
COMMON /BLOP/ F(61, 2), U(61, 2), V(61, 2), W(61, 2), B(61, 2)
1
NP0 = NP
NP1 = NP + 1
NP = NP + 1
IF (NX.EQ.0) NP = NP + 3
IF (NP .GT. 61) NP = 61
DO 35 J = NP0, NP
F(J, 1) = U(NP0, 1) * (ETA(J) - ETA(NP0)) + F(NP0, 1)
U(J, 1) = U(NP0, 1)
V(J, 1) = 0
W(J, 1) = 0
35 CONTINUE
NNP = NP - (NP1 - 1)
WRITE (6, 6000) NNP
RETURN
6000 FORMAT (1HU, 5X, 13H, ETA GROWTH + .13, 14H - POINTS ADDED)
END
SUBROUTINE GRID

COMMON /BLC/ NP, NX, NXT, INTR, INV, ETA(N), VGP, CNU, ETA(61), A(61),
1 ETA(61), O5, 01, GAMA, GAMA, UREF

1 IF((VGP - 1.0) * LE. 0.001) GO TO 5
NP = ALOG10((ETA/N /ETA(1)) * (VGP - 1.0) + 1.0) / ALOG10(VGP) + 1.0001
GO TO 15
5 IF(NP.LE.61) GO TO 15
WRITE(6, 9000)
STOP
10 ETA(1) = 0.0
DO 20 J = 2, 61
ETA(J) = VGP * ETA(J - 1)
A(J) = 0.5 * ETA(J - 1)
20 ETA(J) = ETA(J - 1) + ETA(J - 1)
RETURN
9000 FORMAT(1H0, 36HNP EXCEEDED 61 -- PROGRAM TERMINATED )
END
SUBROUTINE EDDY:
COMMON/BLIC/NX,NXAT,NTR,INV,ETAL,VGP,CNU,DETA(61),A(61),
1 ETA(61),03U(61),GAMMA1,GAMMA2,UREF,REYN
COMMON/BLIC/X(60),U(60),P(60),P2(60),CEL(60),RX(60),CF,
1 Pi,P2,P2,KTHETA(60)
COMMON/BLCP/F(61,2),U(61,2),V(61,2),W(61,2),B(61,2)
1 DELV(61),U=FL(61),DELU(61),DELU(61)
DIMENSION EDV(61)
F1=0.0
DO 30 J=2,NP
F2=(U(J,2)/U(NP,2))*(1.-U(J,2)/U(NP,2))
THE=THE+0.5*(F1+F2)*DETA(J-1)
30 F1=F2
THE=SORT(REYN)
RTHE=UE(NX)*THE/CNU
IF(RTHE.GE.5000.) GO TO 40
Z1=RTHE/425.-1.-0
PT=0.55*(1.0-EXP(SQRT(Z1)-.-29.*Z1))
A1=0.0168*1.55*(1.+P)
GO TO 45
40 A1=0.0168
CONTINUE
20 IFLG=0
RZ2=SORT(REYN)
RZ4=SORT(RZ2)
EDVO=A1*RZ2*U(NP,2)*ETA(NP)-F(NP,2)
J=1
30 IF(IFLG.EQ.1) GO TO 30
UEOUT=U(UE(NX)/UREF)*RZ4/SORT(VABS)
PPLUS=PZ(NX)*UEOUT**3/REYN
PA=1.01.8*PPLUS
IF(PA.LE.0.0) PA=0.5
35 YOA=RZ*ETA(J)*SORT(VABS*PA)/26.0
EL=1.
IF(YOA.LT.0.) EL=(1.-EXP(-YOA))
EDVI=0.16*RZ2*ABS(V(J,2))*EL*ETA(J)**2
IF(EDVI.LT.0.1) GO TO 100
IFLG=1
40 EDV(J)=EDVO
GO TO 110
100 EDV(J)=EDVI
IF(J.LE.J) GO TO 110
IF(EDV(J).GT.EDV(J-1)) GO TO 110
EDV(J)=EDV(J-1)+EDV(J-1)-EDV(J-2)*VGP
EDV(J)=EDVO
IFLG=1
45 EDV(J)=EDV(J)
IF(J.EQ.NP) GO TO 90
110 B(J,2)=1.0*EDV(J)
J=J+1
IF(J.LE.NP) GO TO 90
RETURN
END
SUBROUTINE ECUY
COMMON /BLCO/ NP, NX, NX0, INV, ETA, VGP, CNU, DELTA(61), A(61),
1 ETA(61), CSG(60), GAMMA1, GAMMA2, UREF
COMMON /BLCC/ X(60), UE(60), P1(60), P2(60), CEL(60), RX(60), CF, P1
1, PTHETA(60)
COMMON /BLCP/ F(61, 2), U1(61, 2), V(61, 2), W(61, 2), B(61, 2)
1 DELV(61), DELF(61), DELU(61), DELW(61)

F1=0.
THE=0.
DO 30 J=2, NP
F2=U(J, 2) * (1.0-U(J, 2))
THE=THE*0.5 * (F1+F2) * DELTA(J-1)
F1=F2
THE=THE*X(NX)/SQRT(RX(RX))
THE=THE/UE(INX) * THE/CNU
IF(THETAGE.5000.0) GO TO 3
ZI=THE/425.0
PI=0.55 * (1.0-EXP(SQRT(2.0)-0.298 * ZI))
AI=0.0168 * 1.55/(1.+PI)
GO TO 45

40 A1=0.0168
CONTINUE
GAMTR=1.0
UEINTG=0.0
U1=1.0/UE(NTR-1)
DO 10 I=NTR, NX
U2=1.0/UE(I)
UEINTG=UEINTG+ (U1+U2) * (X(I)-X(I-1)) * 0.5
U1=U2
10
GG=0.35*0.04*UE(NX)**3/(RX(NTR-1)**1.34*CNU**2)
EXPTM=GG* (X(NX)-X(NTR-1)) * UEINTG
IF(EXPTM.LE.10.0) GO TO 15
WRITE(6, 9160) GG, UEINTG, EXPTM
GO TO 20

15 GAMTR=1.0-EXP(-EXPTM)
CONTINUE
IF(LGO=0
RX2=SQRT(RX(NX))
RX4=SQRT(RX2)
PPLUS=P2(NX)/RX4*V(1, 2)**1.5
RX216=RX2**0.16
CN=SQRT(11.0-11.8*PPLUS)
CRSQV=CN*RX4*SQRT(V(1, 2))/26.3
J=1
EDVO=A1*RY2*(ETA(NP)-F(NP, 2)+F(1, 2)) * GAMTR
IF(IFLAG.0, 40) GO TO 10
YN=CRSQV*ETA(J)
EDVI=RX216*ETA(J)**2*V(J, 2)*(1.0-EXP(-YOA))**2*GAMTR
IF(EDVI.LT.EDVG) GO TO 20

50 IFLAG=1
100 EDV=EDVO
GO TO 300
200 EDV=EDVI
300 B1=J+1
J=J+1
IF(J.NE.NP) GO TO 50
RETURN

SUBROUTINE EDDY
C OPT = 1 PMDFP
FTN 4.8+526

9100 FORMAT(1H0, 2X, 3HGG=, 11H0, 3X, 7HLEINTG=, 11H0, 7HEXPTM=, 1E14.14)
END
SUBROUTINE GMOM

COMMON /BLGO/ NP, NX, NTR, INVT, ETA, VGP, CNU, DTG(61), A(61), 
               ETA(61), U(61), P1(61), P2(61), CEL(61), RX(61), CF, P; 

COMMON /BLGC/ X(61), U(61), P1(61), P2(61), CEL(61), RX(61), CF, P; 

COMMON /BLCP/ F(61,2), U(61,2), v(61,2)04(61,2)0(61,2) 
               DELG 
               DELW 

COMMON /BLCA/ S1(61), S2(61), S3(61), S4(61), S5(61), S6(61), S7(61) 
               S8(61), R1(61), R2(61), R3(61), R4(61) 

DATA GAMMA1,0,., GAMMA2,1,0/ 

DO 100 J=2,NP 
      USB=0.5*(U(J,2)**2+U(J-1,2)**2) 
      FV0=0.5*(F(J,2)**2+F(J-1,2)**2) 
      UB0=0.5*(U(J,2)+U(J-1,2)) 
      VB0=0.5*(V(J,2)+V(J-1,2)) 
      DERV=8*(B(J,2)*V(J,2)-B(J-1,2)*V(J-1,2))/DETA(J-1) 
      IF(NX.GT.1) GO TO 10 
      CFB=0.0 
      CVB=0.0 
      CFB=0.0 
      GO TO 10 

      CBF=0.5*(F(J,1)+F(J-1,1)) 
      CBF=0.5*(F(J,1)+F(J-1,1)) 
      CBV=0.5*(F(J,1)+F(J-1,1)) 
      CBV=0.5*(F(J,1)+F(J-1,1)) 
      CUB=0.5*(U(J,1)+U(J-1,1)) 
      CUB=0.5*(U(J,1)+U(J-1,1)) 
      CVD=0.5*(V(J,1)+V(J-1,1)) 
      CVD=0.5*(V(J,1)+V(J-1,1)) 
      DERB=8*(B(J,1)*V(J,1)-B(J-1,1)*V(J-1,1))/DETA(J-1) 
      IF(NX.EQ.1) GO TO 30 
      CLB=0.5*(IP*P1(NX-1)*CFV3+P2(NX-1)*(1+C-CUSB)) 
      CRB=-P2(NX)+CEL(NX)*(CFV3-CUSB)-CLB 
      GO TO 35 

      CRB=-P2(NX) 
      R2(J)=CRB*(DERB+P1*FV8+P2*USB-CEL(NX)**(CF3*V8-CVB*FB)) 
      R1(J)=F(J-1,2)-F(J,2)+DETA(J-1)*U3 
      R3(J)=U(J,2)-U(J,2)+DETA(J-1)*V3 
      R4(J)=0.0 
      CONTINUE 

      K1(NP)=0.0 
      R2(NP)=0.0 
      R3(NP)=0.0 
      R4(NP)=0.0 
      RETURN 

END
SUBROUTINE IC0H171
COMMON/3LCO/NP, NX, NT, NT, IV, ETA, VGF, CNU, DETA(61)
1 \( A, b(N), T(N), OSO (61), \) GAMMA1, GAMMA2, UREF, REYN
COMMON/3LCC/X(60), UE(60), P1(60), P2(60), CEL(60), RX(60), CF, P1
1 \( \) RTHETA(60)

1 COMMON/3LCA/S{[6], S{[6]}, S{[6]}, S{[6]}, S{[6]}, S{[6]}, S{[6]}, S{[6]}
1 COMMON/3LCA/F(1), 2, U(b1,2), V(b1,2), W(61,2),
1 \( B(61,2), D(61,2), Q(61,2), V(61,2), CELU(61), DELW(61)
1 \( \) BEL=1.0, \( \) X(NX)-X(NX-1))
1 \( \) GAMMA1=1.0
1 \( \) GAMMA2=ETA(NP)-DSO(NX)*SORT(REYN)
1 \( \) DO 30 J=2,NP
1 \( \) FLARE=1.0
1 \( \) IF(U(J,2)=L.T.0) FLARE=0.
1 \( \) FB=0.5*(F(J,2)+F(J-1,2))
1 \( \) UB=0.5*(U(J,2)+U(J-1,2))
1 \( \) VB=0.5*(V(J,2)+V(J-1,2))
1 \( \) FYB=0.5*(F(J,2)+V(J-1,2)+F(J-1,2)*V(J-1,2))
1 \( \) US3=0.5*(U(J,2)+U(J-1,2))
1 \( \) WS3=0.5*(W(J,2)+W(J-1,2))
1 \( \) DERBV=(B(J,2)*V(J-1,2)-8(J-1,2)*V(J-1,2))
1 \( \) DETA(J-1)
1 \( \) CF3=0.5*(F(J,1)+F(J-1,1))
1 \( \) CB3=0.5*(U(J,1)+U(J-1,1))
1 \( \) CV3=0.5*(V(J,1)+V(J-1,1))
1 \( \) CF3=0.5*(F(J,1)+V(J,1)+F(J-1,1)*V(J-1,1))
1 \( \) CUB3=0.5*(U(J,1)+U(J-1,1))
1 \( \) CBS3=0.5*(U(J,1)+U(J-1,1))
1 \( \) COERBV=(B(J,1)*V(J,1)-8(J-1,1)*V(J-1,1))
1 \( \) DETA(J-1)
1 \( \) CRB=BEL*(CF3B+CGWD-CUSH*FLARE)-CDEBV
1 \( \) S1(J)=B(J,2)/DETA(J-1)+0.5*BEL*(F(J,2)-CF3)
1 \( \) S2(J)=0.5*(F(J,2)+DETA(J-1)+0.5*BEL*(F(J,2)-CF3)
1 \( \) S3(J)=BEL*(V(J,2)-CV3)
1 \( \) S4(J)=0.5*(BEL*(V(J,2)-CV3))
1 \( \) S5(J)=BEL*(U(J,2)+F(J,2))
1 \( \) S6(J)=BEL*(U(J,2)+F(J,2))
1 \( \) S7(J)=BEL*(U(J,2)+F(J,2))
1 \( \) S8(J)=BEL*(U(J,2)+F(J,2))
1 \( \) R1(J)=F(J,1)+U(J,1)+DETA(J-1)*UB
1 \( \) R3(J)=U(J,1)+U(J,1)+DETA(J-1)*VB
1 \( \) R2(J)=CRB-(0EPRV+J'a.*FVO.+9,:.Ls(W::J-USERIARO-BEL*(CF9
1 \( \) V(2)-1
1 \( \) BEL*(V(J,2)-CV3))
1 \( \) VS(J)=BEL*(U(J,2)+F(J,2))
1 \( \) R4(J,1)=0.0
1 \( \) R1(J)=0.0
1 \( \) R2(J)=0.0
1 \( \) R3(NP)=W(NP,2)*GAMMA2-F(NP,2)
1 \( \) R4(NP)=0.0
1 \( \) RETURN
1 \( \) END
DO = W3(NP) * (A13(NP) * A24(NP) - A14(NP) * A23(NP) - A12(NP) * A21(NP) + A11(NP) * A23(NP))

A13(NP) * A22(NP) - GAMMA1 * (W2(NP) - A14(NP) * A23(NP) - A12(NP) * A24(NP))

A13(NP) * A23(NP) - A11(NP) * A22(NP) - A12(NP) * A21(NP) + A14(NP) * A24(NP) + A11(NP) * A23(NP) - A12(NP) * A24(NP)

A11(NP) * (A23(NP) - A13(NP) * A24(NP) - A12(NP) * A21(NP)) + GAMMA1 * (W2(NP) - A14(NP) * A23(NP) - A12(NP) * A24(NP))

A23(NP) - A13(NP) * A24(NP) - A12(NP) * A21(NP) + A14(NP) * A24(NP) + A11(NP) * A23(NP) - A12(NP) * A24(NP)

A11(NP) * (A23(NP) - A13(NP) * A24(NP) - A12(NP) * A21(NP)) + GAMMA1 * (W2(NP) - A14(NP) * A23(NP) - A12(NP) * A24(NP))

A23(NP) - A13(NP) * A24(NP) - A12(NP) * A21(NP) + A14(NP) * A24(NP) + A11(NP) * A23(NP) - A12(NP) * A24(NP)

J = J + 1

CC1 = DELU(J + 1) - W3(J) - A(J + 1) * DELV(J + 1)

CC2 = DELU(J + 1) - W4(J)

CC3 = A13(J) * A(J + 1) - A12(J)

CC4 = W11(J) - A12(J) * CC1 - A13(J) * CC2

CC5 = A23(J) - A(J + 1) * A22(J)

CC6 = W2(J) - A22(J) * CC1 - A24(J) * CC2

DEN0 = A11(J) * CC5 - A21(J) * CC3

DELV(J) = (CC4 * CC6 * CC3 * CC5 * CC0) / DEN0

DELU(J) = CC2

DELU(J) = CC1 - A(J + 1) * DELV(J)

IF(J, G.E., 2) GO TO 25

WRITE(6, 90) J, DELU(J)

DO 30 J = 1, NP

F(J, 2) = F(J, 2) + DELF(J)

30 CONTINUE

IF(NX, G.E., INV) U(NX) = W(NP, 2) * DELF

RETURN

9000 FORMAT (1I4, 3X, 6E14.6)
SUBROUTINE OUTPUT

COMMON /BLCC/ NP, NX, NXT, NTR, INV, ETA, VGP, CHU, DELS(61), A(61),
       ETA(61), USU(61), GAMMA1, GAMMA2, UREF, REYN

COMMON /BLCC/ X(J), U(J), P1(60), P2(60), CEL(60), RX(60), CF, P1

COMMON /BLCC/ F(J2,2), V(J2,2), W(J2,2), B(J2,2), DELS(61), DELF(61), DELU(61), DELH(61)

DIMENSION YODS(J), UP(J)

IF(NX, GE, INV) GO TO 600

F1 = 0.0
THETA1 = J, 0
DO 150 J = 2, NP
F2 = U(J, 2) * (U(J, 2) - U(J2, 2))
THETA1 = THETA1 + (F1 + F2) * 3.5 * DELTA(J - 1)
F1 = F2
THETA1 = THETA1 / SORT(REYN)
DELS = (ETA(NP) - F(NP, 2)) / (U(J, NP) - U(J2, NP))

150  THETA1 = THETA1 / SORT(REYN)
THETA1 = THETA1 / SORT(REYN)
H = DELS / THETA1
CF = 2.0 * V(1, 2) / SORT(REYN)
RTHETA(N) = U(N) * DELS / CHU
RDELS = UREF * DELS / CHU

DO 500 J = 1, NP
YODS(J) = ETA(J) / SORT(REYN) / DELS
UP(J) = SORT(UREF / CHU) + V(J, 2) * DELS
WRITE(6, 450) (J, ETA(J), F(J, 2), U(J, 2), V(J, 2), B(J, 2)
WRITE(6, 450) (J, YODS(J), J = 1, NP)
WRITE(6, 450) (J, UP(J), J = 1, NP)
WRITE(6, 450) (J, V(J, 2), J = 1, NP)
WRITE(6, 450) (J, B(J, 2), J = 1, NP)

210  NX = NX + 1
IF(NX, GT, NXT) GO TO 600

DO 250 J = 1, NP
F(J, 1) = F(J, 2)
U(J, 1) = U(J, 2)
V(J, 1) = V(J, 2)
W(J, 1) = W(J, 2)
RETURN

STOP

FORMAT(1H9, 2X, 1HJ, 4X, 3HETA, 9X, 1HF, 15X, 1HU, 13X, 1HV, 13X, 1HB,
       13X, 7HOUODYODS, 3X, 4HYODS)

1 13X, 7HOUODYODS, 3X, 4HYODS

FORMAT(1H9, 2X, 1HJ, 4X, 3HETA, 9X, 1HF, 15X, 1HU, 13X, 1HV, 13X, 1HB,
       13X, 7HOUODYODS, 3X, 4HYODS)

1 13X, 7HOUODYODS, 3X, 4HYODS

END.
7. REFERENCES


