GEORGIA INSTITUTE OF TECHNOLOGY | OFFICE OF CONTRACT ADMINISTRATION

PROJECT ADMINISTRATION DATA SHEET

[x] ORIGINAL
[ ] REVISION NO. ____

Project No. E-16-606 | DATE: 6/2/81

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 | Contracted through:

Cost Sharing: N/A | GTRI/~~GIT~~

Title: Studies in Three-Dimensional Turbulent Boundary Layer Separation from Smooth Surfaces

ADMINISTRATIVE 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 Requiremen

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 Security Services	EES Research Public Relation
Research Property Management	✓ Reports Coordinator (OCA)	Project File (OCA)
Accounting Office	Legal Services (OCA)	Other: ____
Procurement/EES Supply Services	Library, Technical Reports	

GEORGIA INSTITUTE OF TECHNOLOGY OFFICE OF CONTRACT ADMINISTRATION

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 (School/~~Laboratory~~)

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 ____________

FORM OCA 10:781

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

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.

2. TASK "0"

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 β from

$$\beta = 90.0^{\circ} - 37.8 * \sqrt{M' - 1.0} \qquad (1)$$

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 β 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 $(R_{\delta^*})_b$, the pre-shock Mach number is M, and $R = \log_{10} (R_{\delta^*})_b$, Inger's analysis gives:

$$(c_f)_s = (0.252 * R + 3.4273 - 5.5 M + 3.15 H - H^2)(c_f)_b \qquad (2)$$

$$\ell_{up} = (9.4\ H - 108.0\ M + 40\ M^2 + 61.124)(\delta_b^*) \tag{3}$$

$$\ell_{dn} = (0.25\ R - 6,414.8 + 8,758.4\ H - 2,756.9\ H^2 +$$

$$(10,639 - 14,659\ H + 4,686\ H^2)M$$

$$+ (-4,439 + 6,157\ H - 1,992\ H^2)M^2)(\delta^*)_b \tag{4}$$

$$(c_f)_a = \Big[(4,568 - 6,079\ H + 2,061\ H^2)R + (2,085.47$$

$$- 2,695.05\ H + 874.1\ H^2)M + (-931.01 + 1,201.8\ H$$

$$- 389.7\ H^2)\ M^2 - 1,188.548 + 1,539.911\ H - 500.049\ H^2\Big]c(c_f)_b \tag{5}$$

$$\delta_a^* = \delta_b^* \Big[1 + (5.17 + 8.65 * (H - 1.3)) * (M - 1.0) *$$

$$(1.11 * \tanh\ (R - 2.35))\Big] \tag{6}$$

Schematic variations of δ^* and c_f in the interaction region are shown in Figure 1b. Also $\ell_{up} = x_s - x_b$ and $\ell_{dn} = x_a - x_s$. 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_{up} and ℓ_{dn} are of the order of $10^{-3} \div 10^{-2}$ 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 ℓ_{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 C_p 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 LG4-612 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).

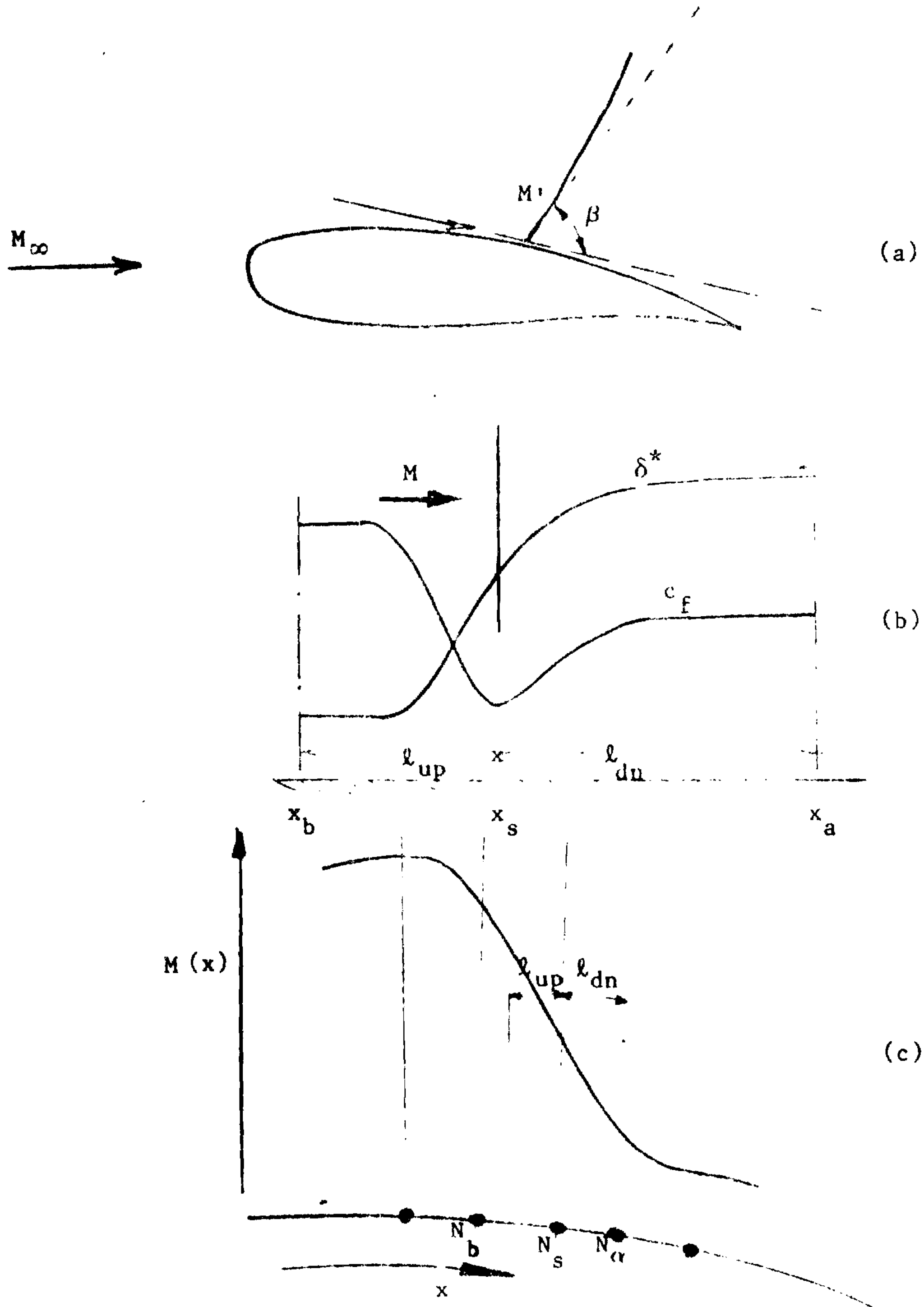
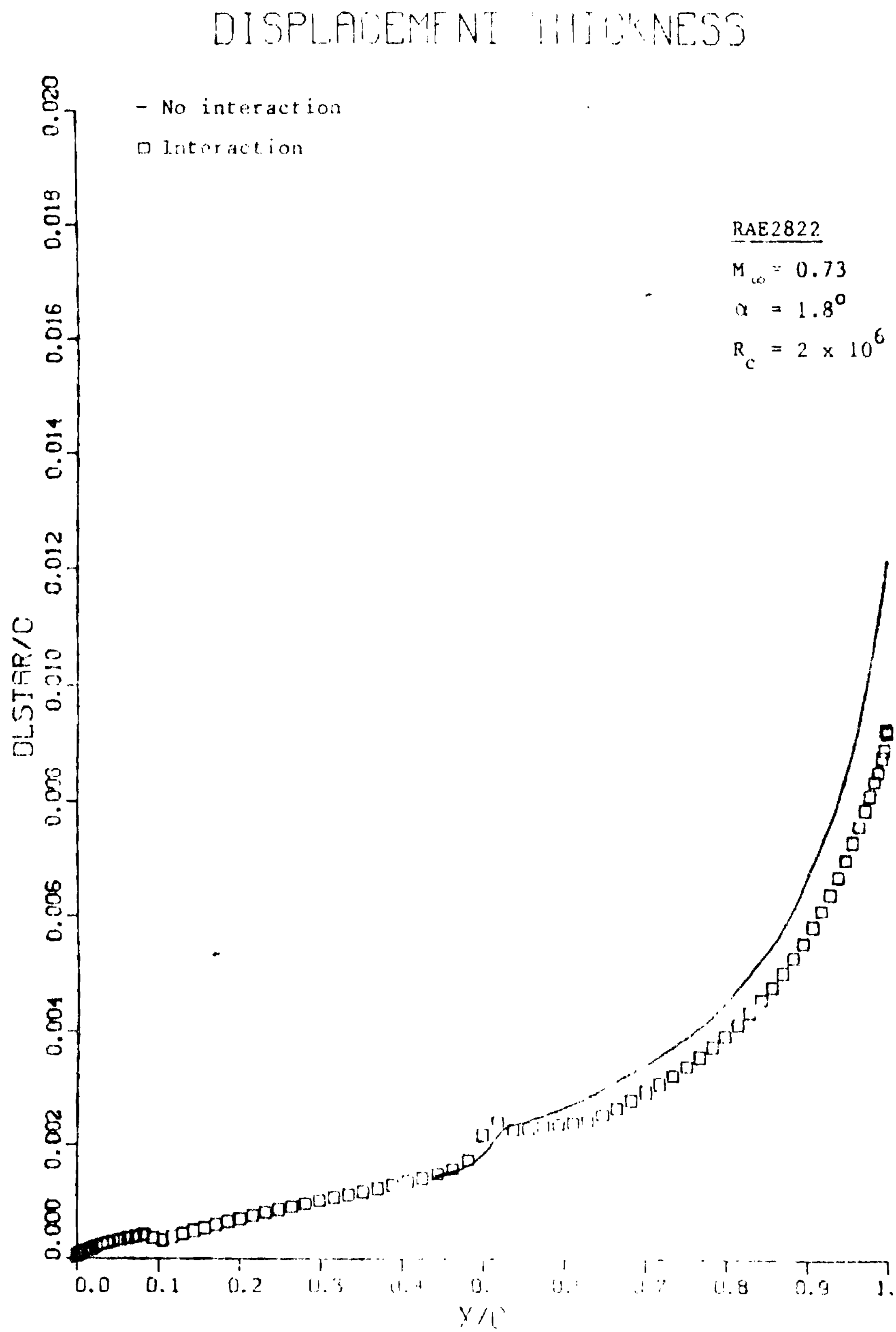
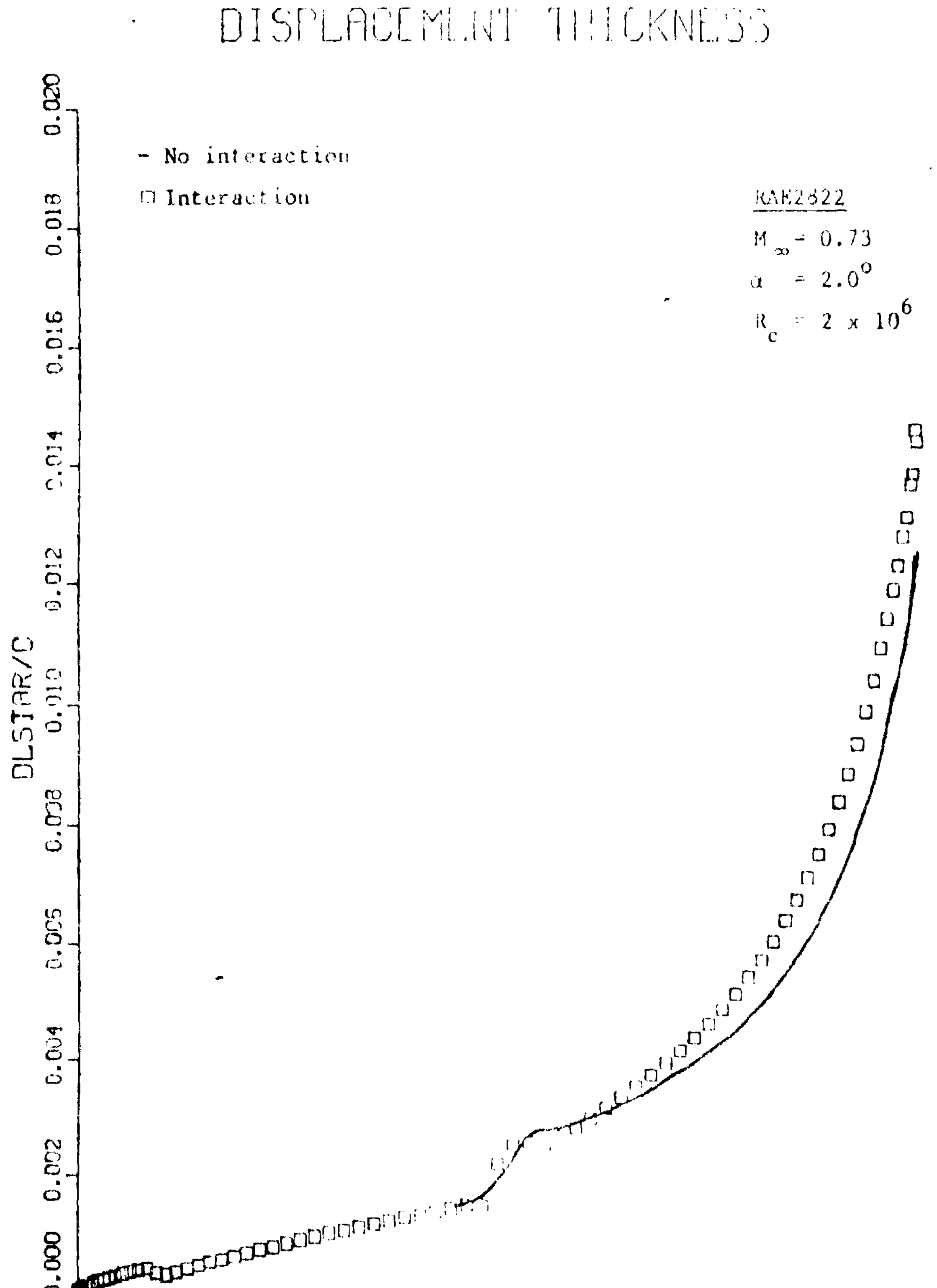


Figure 1 Schematic of the Flowfield in the Shock/Boundary Layer Interaction Region.


DISPLACEMENT THICKNESS
- No interaction
□ Interaction
RAE2822
$M_\infty = 0.73$
$\alpha = 1.8^o$
$R_c = 2 \times 10^6$
DLSTAR/C
0.000
0.002
0.004
0.006
0.008
0.010
0.012
0.014
0.016
0.018
0.020
0.0
0.1
0.2
0.8
0.9
1.0
Y/C


DISPLACEMENT THICKNESS
- No interaction
□ Interaction
RAE2822
M ∞ = 0.73
α = 2.0°
Rc = 2 x 10^6
DLSTAR/C
0.000
0.002
0.004
0.006
0.008
0.010
0.012
0.014
0.016
0.018
0.020
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

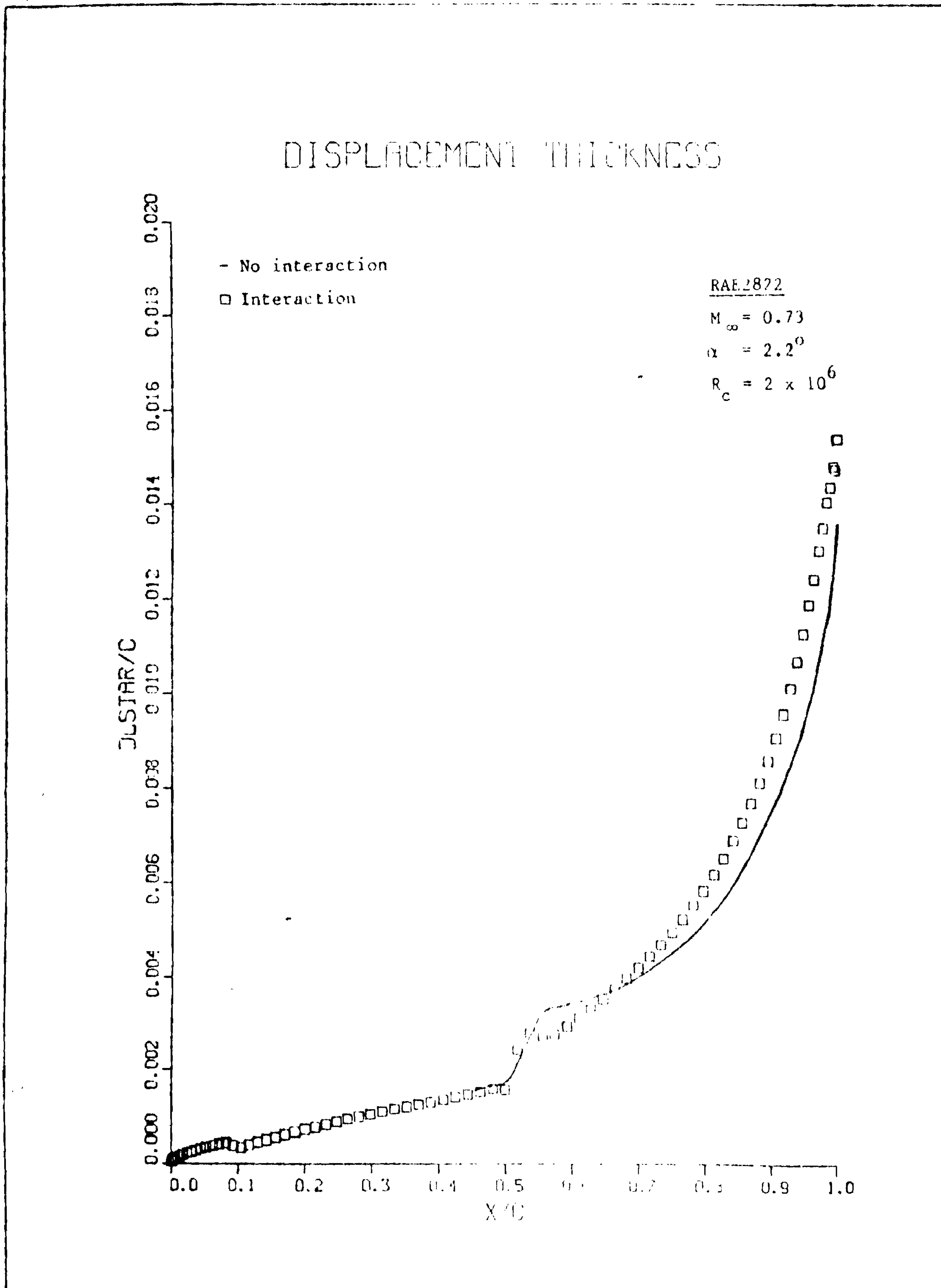
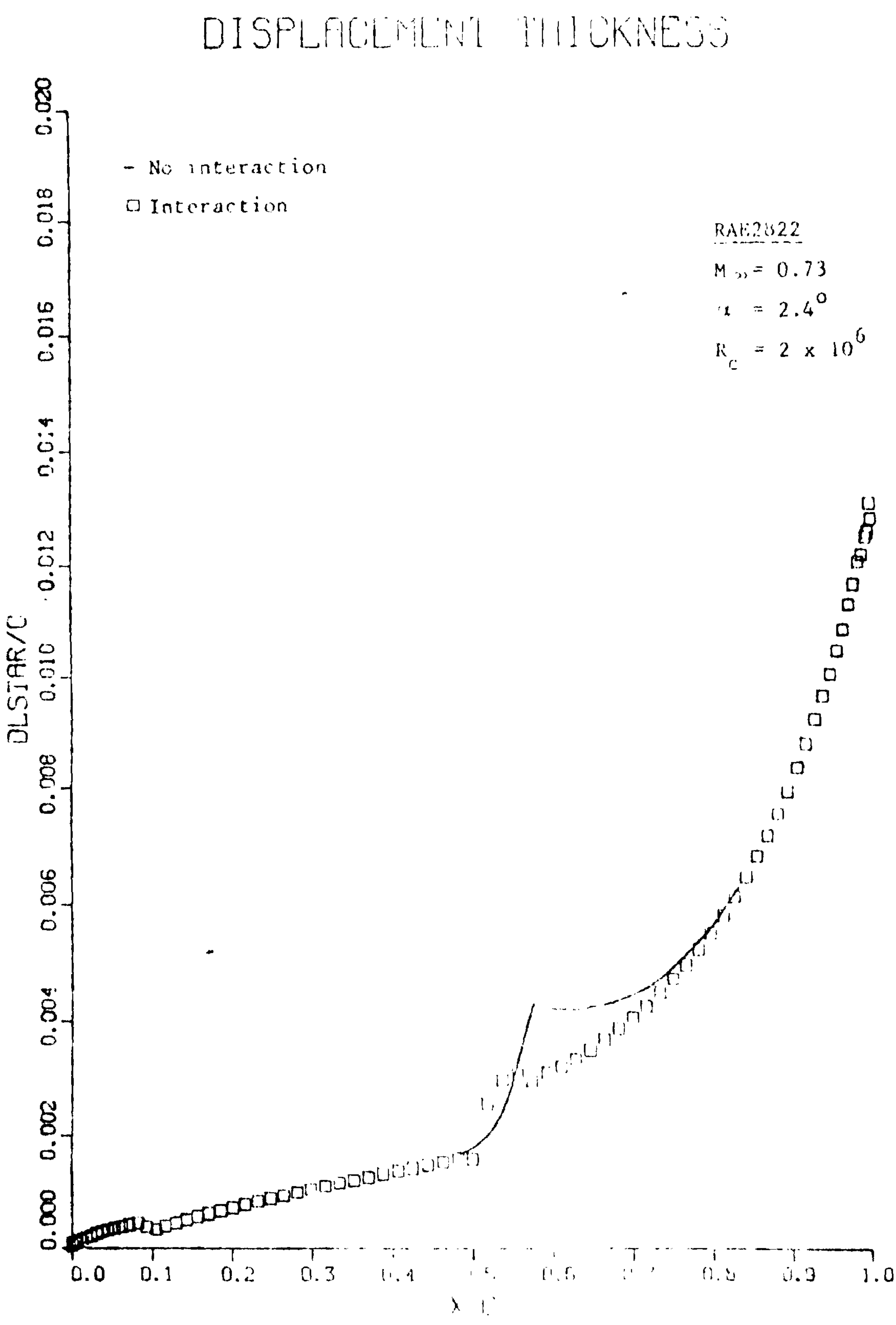
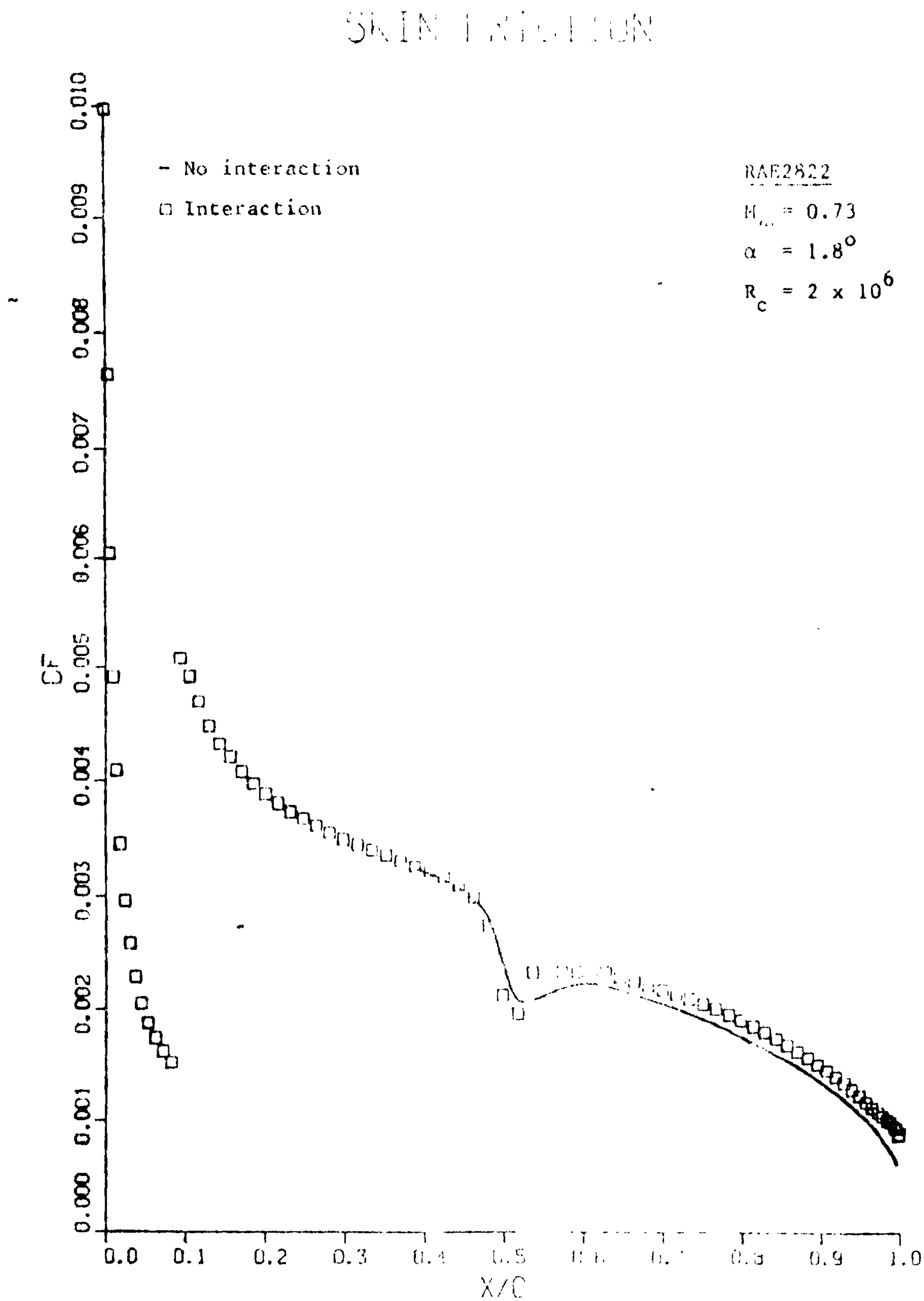

DISPLACEMENT THICKNESS
- No interaction
□ Interaction
RAE 2822
$M_\infty = 0.73$
$\alpha = 2.2^o$
$R_c = 2 \times 10^6$
DLSTAR/C
0.000
0.002
0.004
0.006
0.008
0.010
0.012
0.014
0.016
0.018
0.020
0.0
0.1
0.2
0.3
0.4
0.5
0.9
1.0
X/C

Figure 4



DISPLACEMENT THICKNESS
- No interaction
□ Interaction
RAE2822
$M_\infty = 0.73$
$\alpha = 2.4^o$
$R_c = 2 \times 10^6$
DLSTAR/C
0.000
0.002
0.004
0.006
0.008
0.010
0.012
0.014
0.016
0.018
0.020
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
X/C

Figure 3


SKIN FRICTION
- No interaction
□ Interaction
RAE2822
M_∞ = 0.73
α = 1.8°
R_c = 2 x 10^6
CF
0.010
0.009
0.008
0.007
0.006
0.005
0.004
0.003
0.002
0.001
0.000
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
X/C

Figure

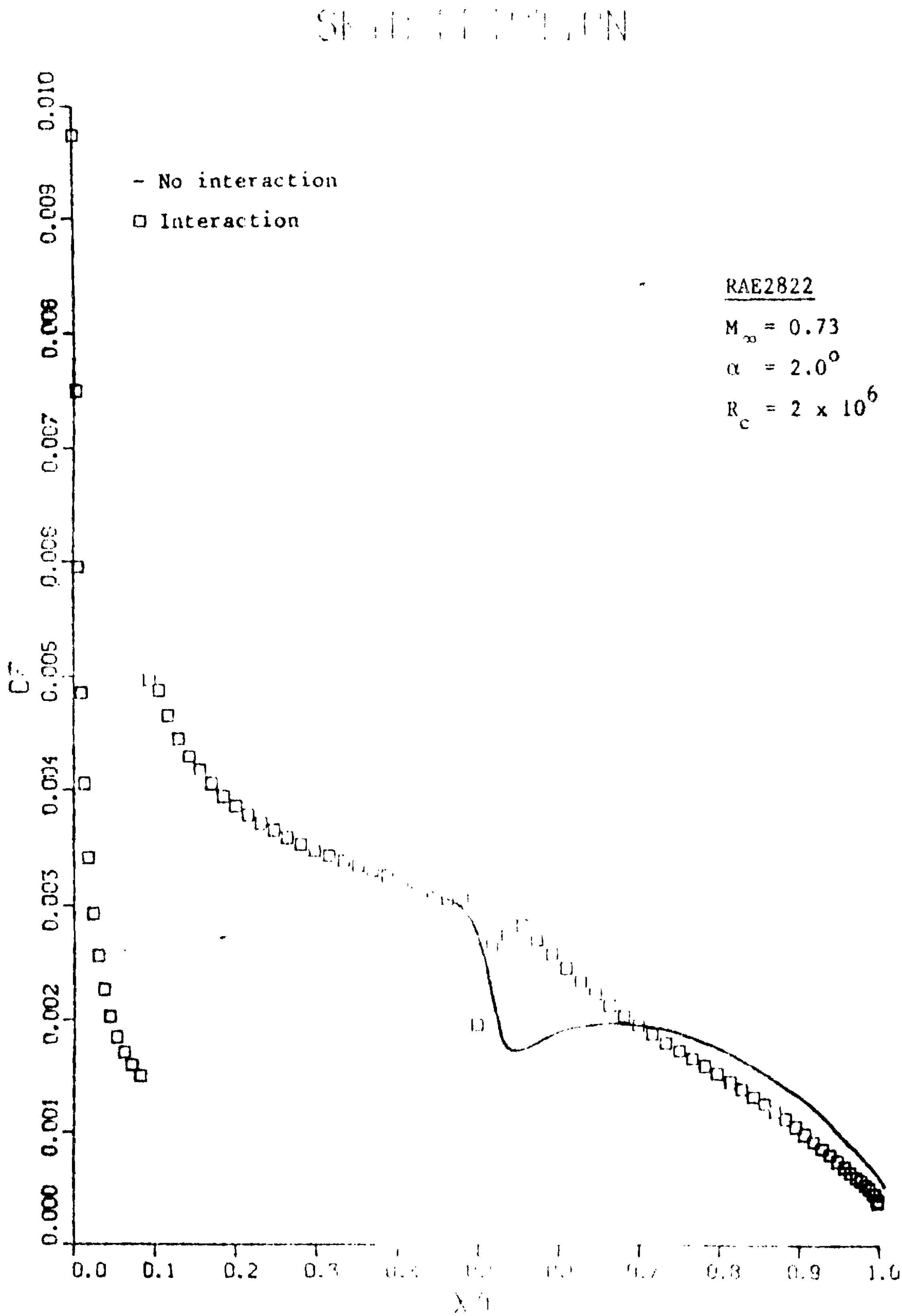
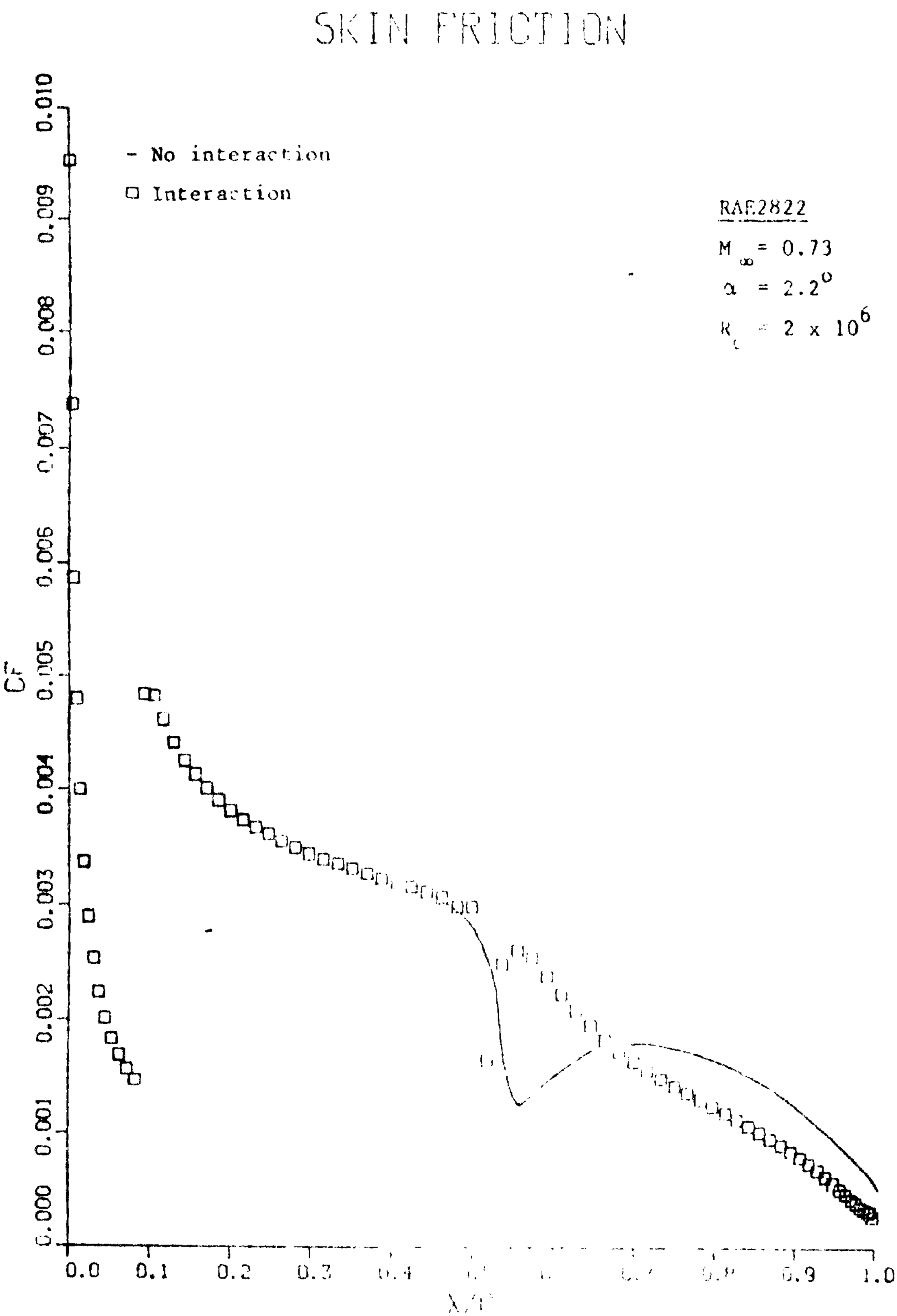
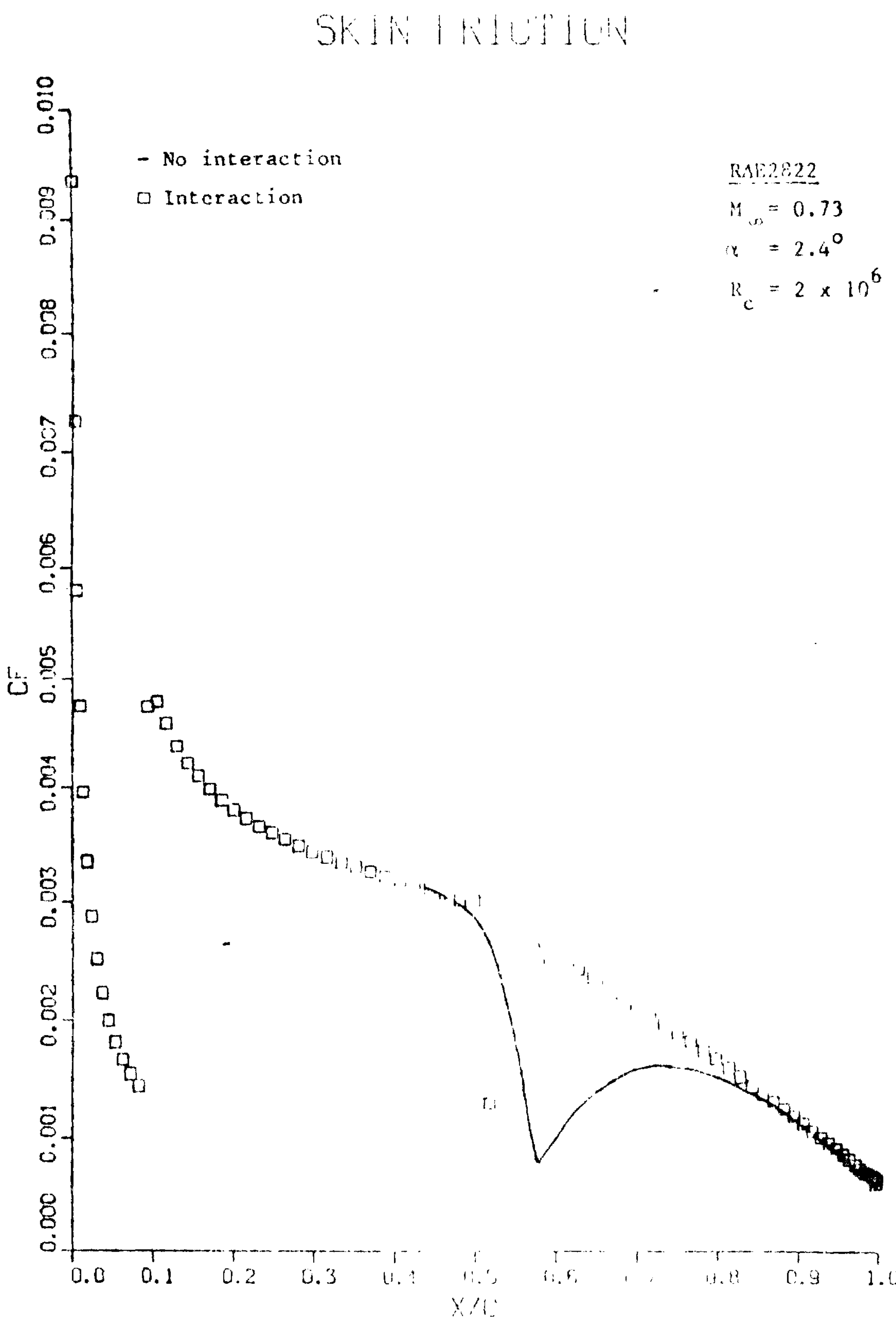
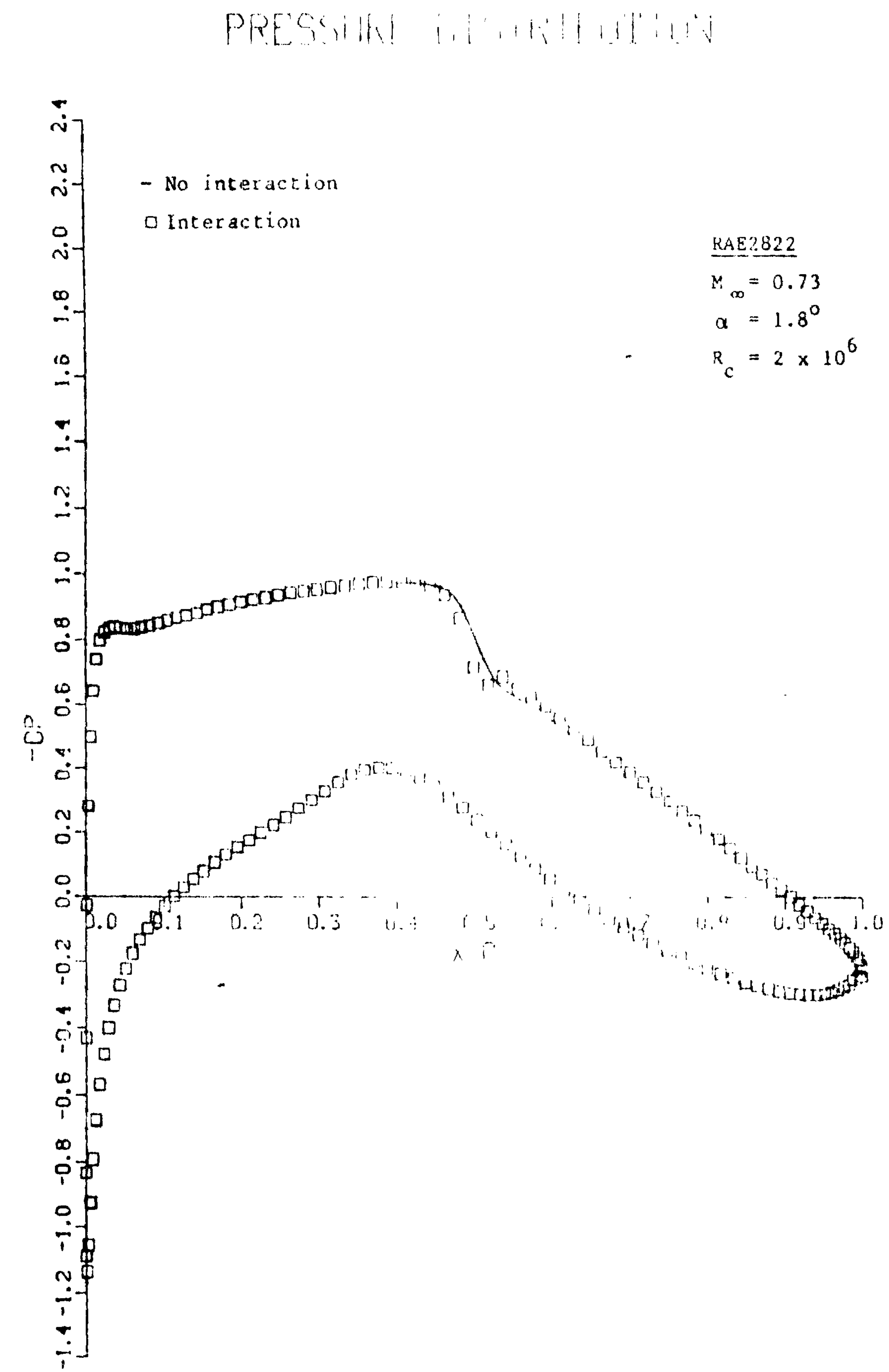

0.010
0.009
0.008
0.007
0.006
0.005
0.004
0.003
0.002
0.001
0.000
Cf
- No interaction
□ Interaction
RAE2822
M_∞ = 0.73
α = 2.0°
R_c = 2 x 10^6
0.0
0.1
0.2
0.3
0.9
1.0

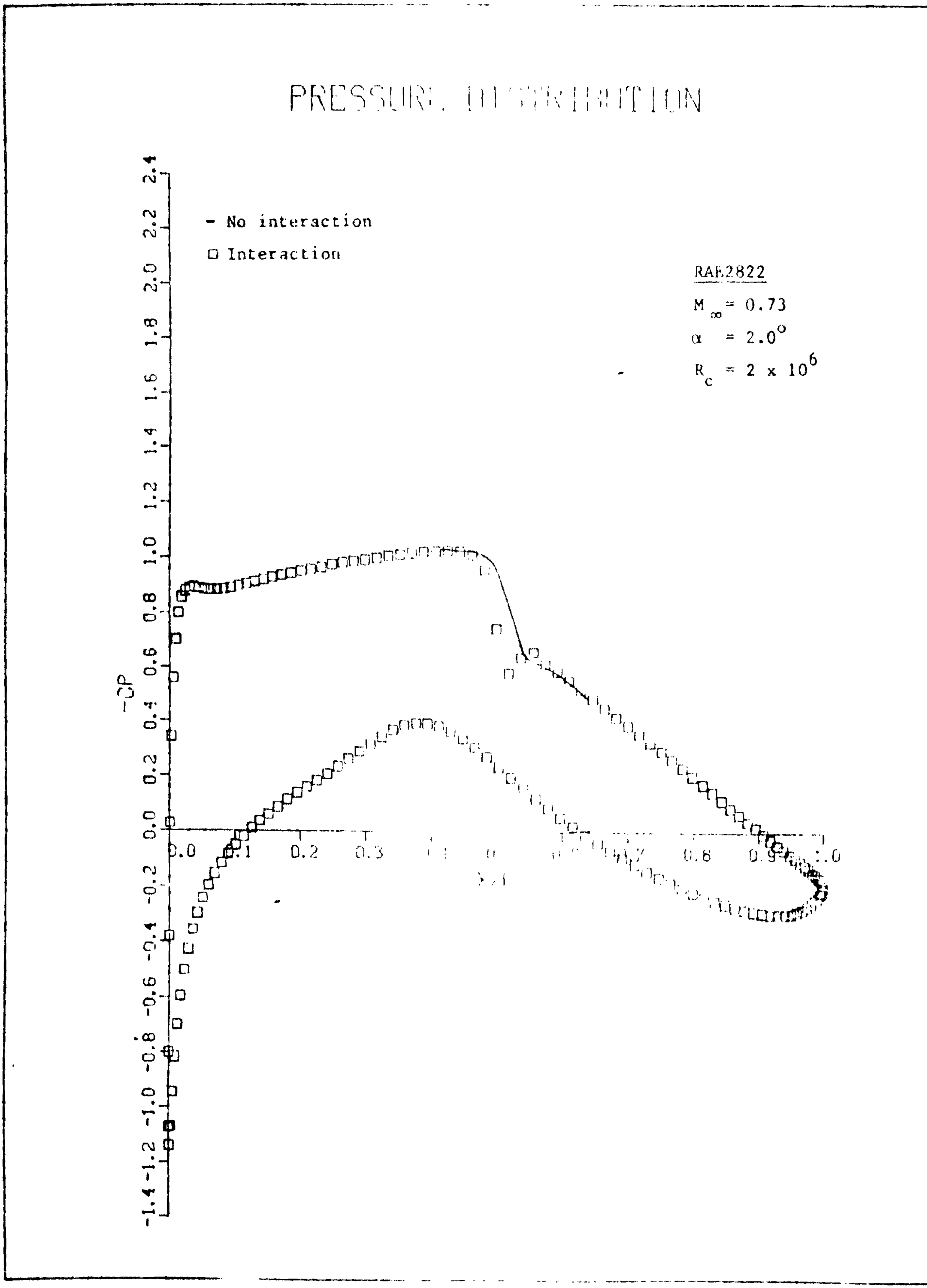
Figure 7



SKIN FRICTION
- No interaction
□ Interaction
RAE2822
M∞ = 0.73
α = 2.2°
Rc = 2 x 10^6
CF
0.000
0.001
0.002
0.003
0.004
0.005
0.006
0.007
0.008
0.009
0.010
0.0
0.1
0.2
0.3
0.9
1.0


SKIN FRICTION
- No interaction
□ Interaction
RAE2822
$M_\infty = 0.73$
$\alpha = 2.4°$
$R_c = 2 \times 10^6$
CF
0.000
0.001
0.002
0.003
0.004
0.005
0.006
0.007
0.008
0.009
0.010
0.0
0.1
0.2
0.3
0.9
1.0
X/C

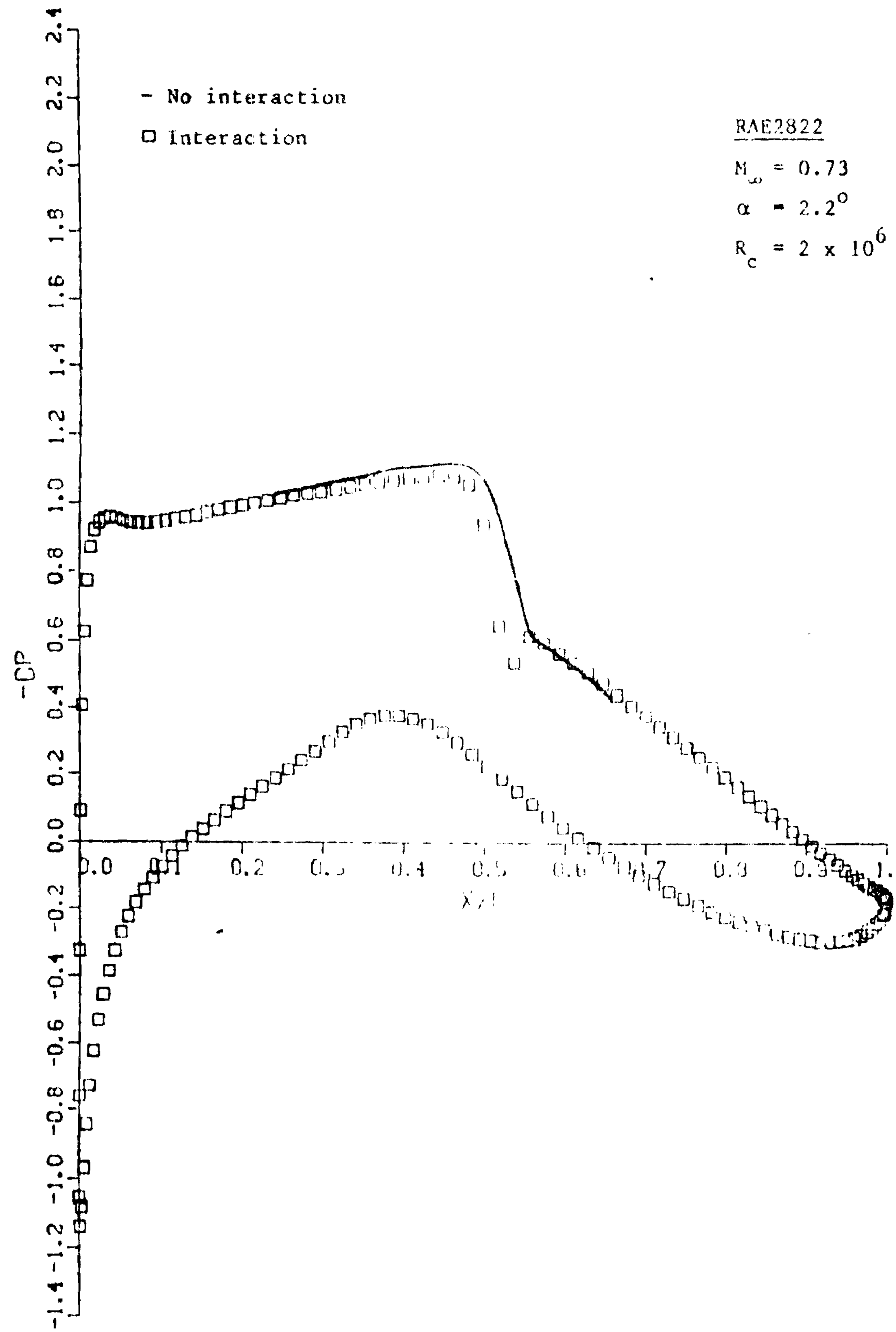

PRESSURE DISTRIBUTION
- No interaction
□ Interaction
RAE2822
$M_\infty = 0.73$
$\alpha = 1.8^o$
$R_c = 2 \times 10^6$
-CP
2.4
2.2
2.0
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
-0.2
-0.4
-0.6
-0.8
-1.0
-1.2
-1.4
0.0
0.1
0.2
0.3
1.0

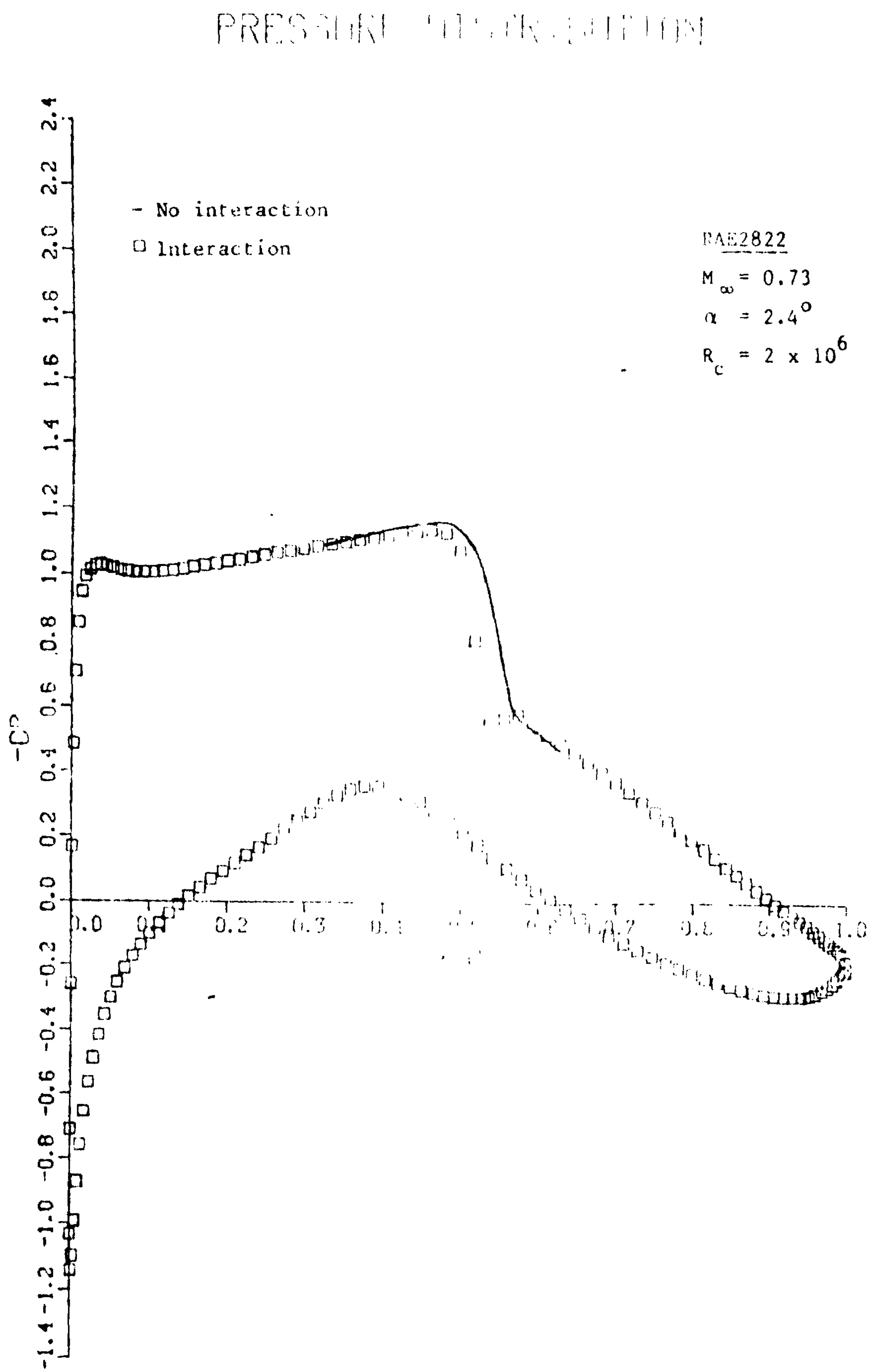


- No interaction
□ Interaction
RAE2822
$M_\infty = 0.73$
$\alpha = 2.0^o$
$R_c = 2 \times 10^6$
-CP
2.4
2.2
2.0
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
-0.2
-0.4
-0.6
-0.8
-1.0
-1.2
-1.4
0.0
0.2
0.3
0.8

Figure 1?

PRESSURE DISTRIBUTION




- No interaction
□ Interaction
RAE2822
$M_\infty = 0.73$
$\alpha = 2.4^\circ$
$R_c = 2 \times 10^6$
$-C_P$
2.4
2.2
2.0
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
-0.2
-0.4
-0.6
-0.8
-1.0
-1.2
-1.4
0.0
0.2
0.3
0.8
1.0

Figure

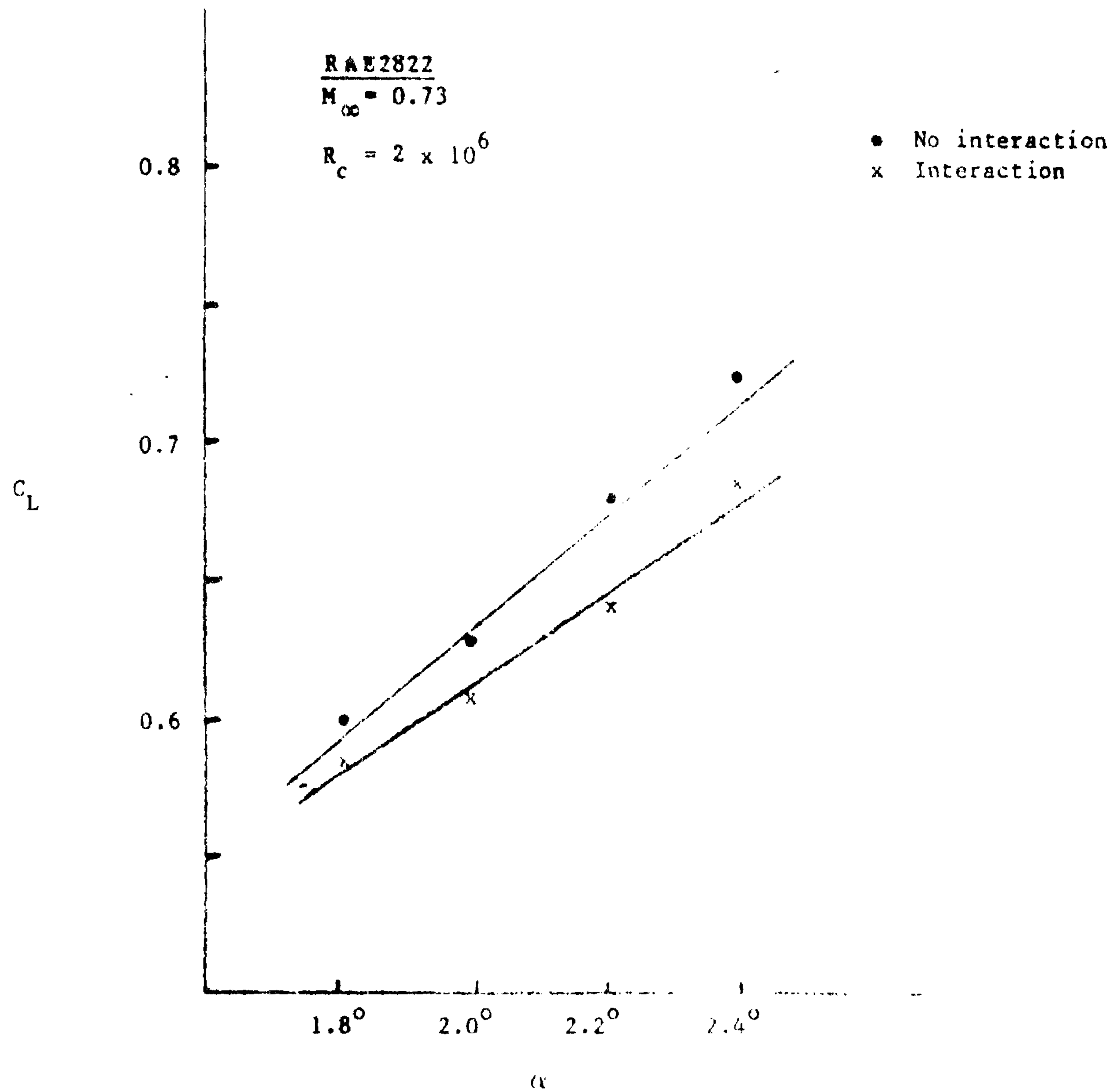

RAE2822
$M_\infty = 0.73$
$R_c = 2 \times 10^6$
• No interaction
x Interaction
0.8
0.7
0.6
C_L
1.8°
2.0°
2.2°
2.4°
α

Figure 14

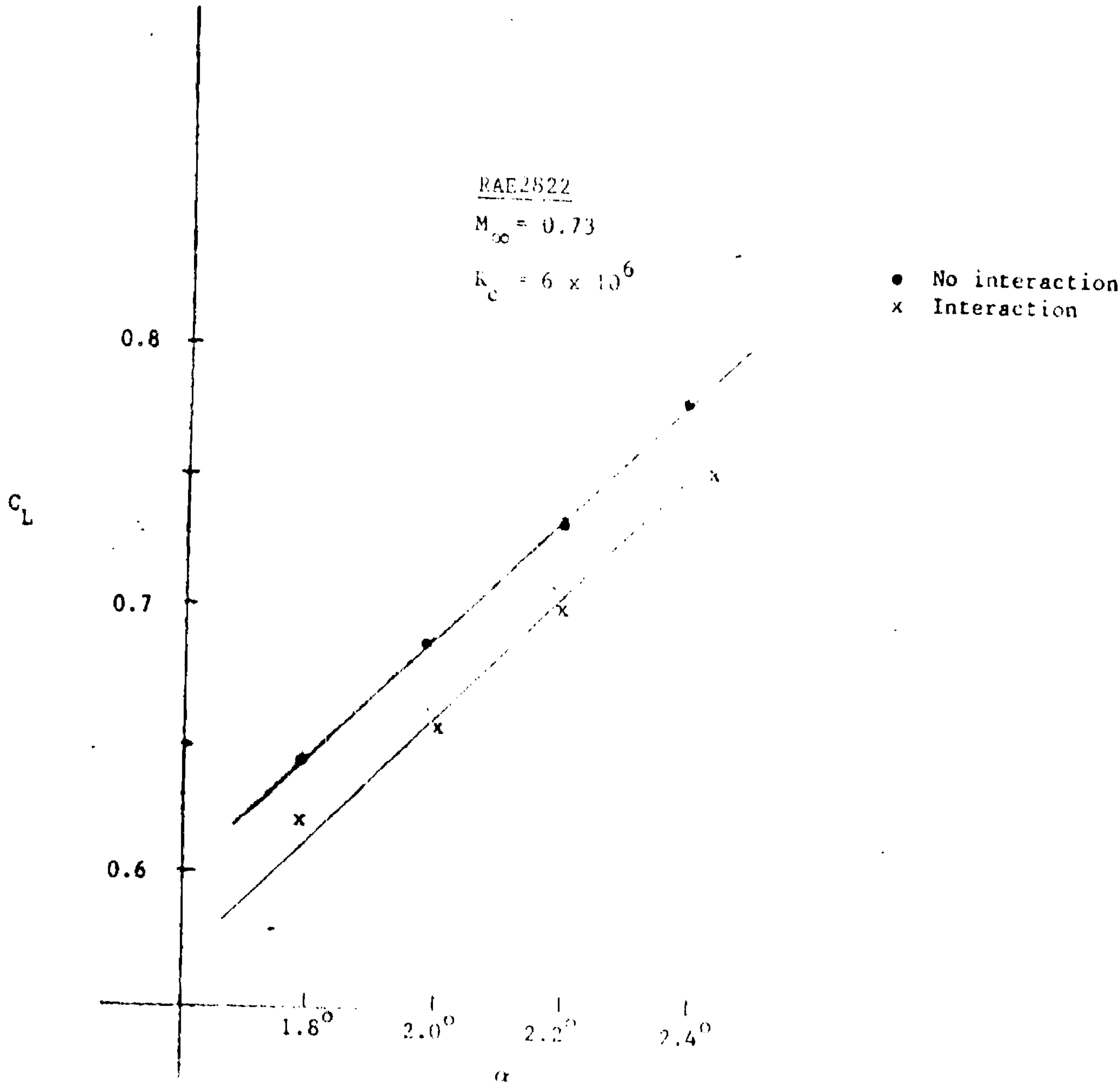

RAE2822
$M_\infty = 0.73$
$R_c = 6 \times 10^6$
● No interaction
x Interaction
0.8
0.7
0.6
C_L
1.8°
2.0°
2.2°
2.4°
α

Figure 15

3. TASK 1

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 δ^* 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 δ^*.

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 (Δ_{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 δ^* 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

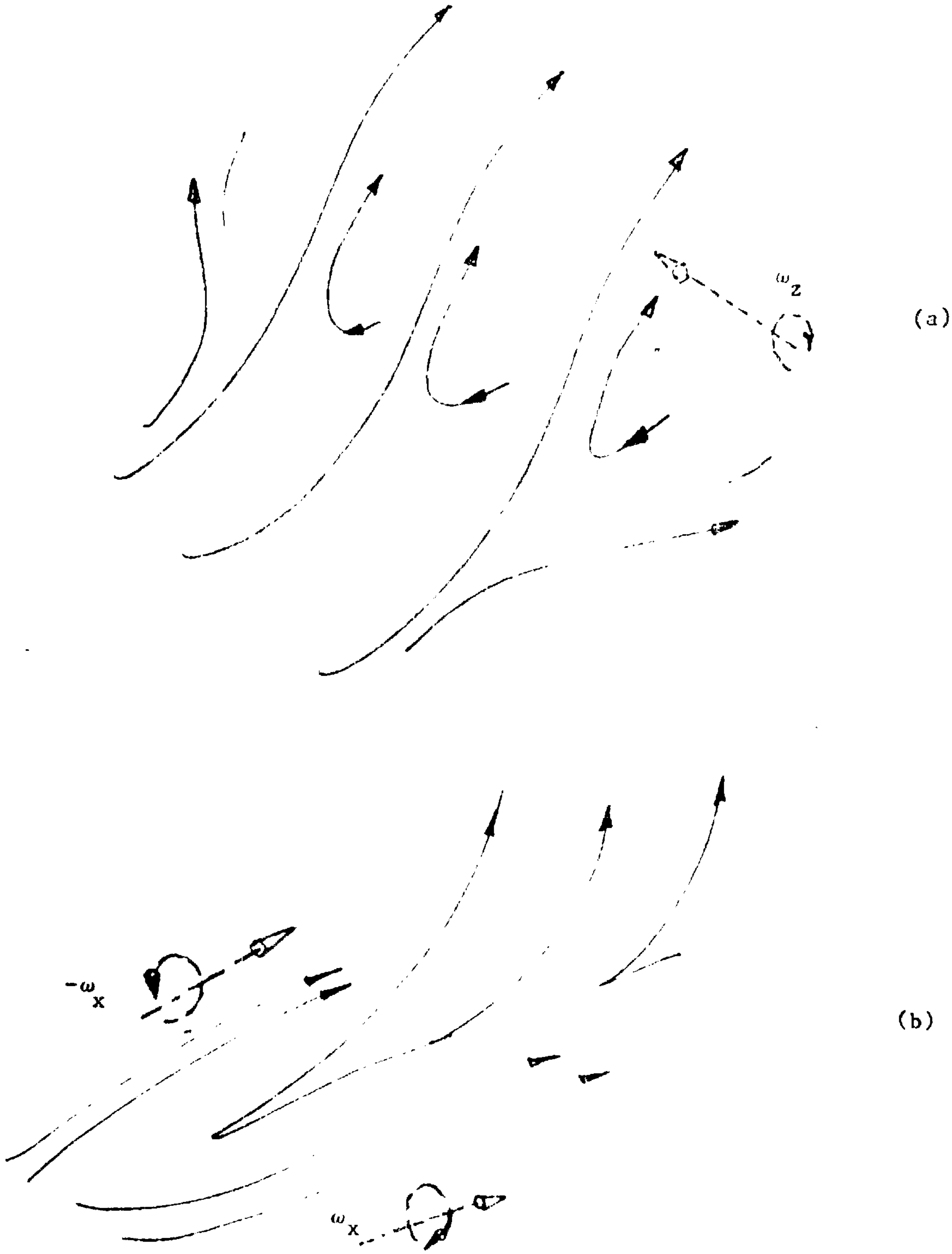


Figure 16. Separation Patterns

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

INVERSE (Main Routine)	Performs the downstream marching and the iteration process
INPUT	Reads input data
IVPL	Initiates a profile at the first station
GRID	Defines the grid normal to the wall
EDDY 1	Calculates the Reynolds stresses when the program is in the inverse mode
EDDY	Calculates the Reynolds stresses when the program is in the direct mode
CMOM	Computes the coefficients of the momentum equation when the program is in the direct mode
ICONZ1	Computes the coefficients of the momentum equation when the program is in the inverse mode
SOLV4	Inverts the block-tridiagonal matrix of the resulting finite-difference formulation of the boundary layer equations.
OUTPUT	Prints the output quantities

```
      PROGRAM INVERSE(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
      COMMON /BLCO/ NP,NX,NXT,NTR,INV,ETAE,VGP,CNU,DETA(61),A(61),
     1             ETA(61),USD(61),GAMMA1,GAMMA2,UREF,REYN
      COMMON /BLCC/ X(60),UE(60),P1(60),P2(60),CEL(60),RX(60),CF,P1(
     1             ,RTHETA(60)
      COMMON /BLCP/ F(61,2),U(61,2),V(61,2),W(61,2)
     1       ,B(61,2),DELV(61),DELF(61),DELU(61),DELW(61)
      ITMAX=50
      MA=0
      CEL(1)=0.0
      NX=1
      CALL INPUT
      CALL GRID
      CALL IVPL
      P2(1)=0.0
      P1(1)=0.5
      STEP=UE(INV-1)/UREF
      STEPH=SQRT(X(INV-1)/(UE(INV-1)/UREF))
      DO 10 J=1,NP
      W(J,1)=STEP
   10 W(J,2)=STEP
   25 WRITE(6,9100) NX,X(NX)
      IT=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)
   60 IT=IT+1
      IF(NX.LT.INV) GO TO 30
      H1=X(NX-1)-X(NX-2)
      H2=X(NX)-   X(NX-1)
      DUDS=(H1*(H1+2*H2)*UE(NX)+H2**2*UE(NX-2)-(H1+H2)**2
     1       *UE(NX-1))/(H1*H2*(H1+H2))
      P2(NX)=X(NX)*DUDS/UE(NX)
      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.MA.NE.0)    GO TO 30
      DO 40 J=1,NP
      F(J,1)=F(J,1)*STEP*STEPH
      F(J,2)=F(J,1)
      U(J,1)=U(J,1)*STEP
      U(J,2)=U(J,1)
      V(J,1)=V(J,1)*STEP/STEPH
      V(J,2)=V(J,1)
      W(J,1)=STEP
      W(J,2)=W(J,1)
   40 CONTINUE
      DETA(1)=DETA(1)*STEPH
      DO 45 J=2,61
      DETA(J)=DETA(J-1)*VGP
      A(J)=0.5*DETA(J-1)
   45 ETA(J)=ETA(J-1)+DETA(J-1)
   30 RX(NX)=UE(NX)*X(NX)/CNU
```

```
      IF(IT.LE.ITMAX) GO TO 70
```



```
      WRITE(5,2500)
      GO TO 90
70     IF(NX.GE.INV) GO TO 1
       IF(NX.GE.NTR) CALL EDDY
       CALL CMOM
       GO TO 2
1      IF(NX.GE.NTR) CALL EDDY1
       CALL ICONZ1
2      CALL SOLV4
51    IF(NX.GE.NTR) GO TO 62
      IF(ABS(DELV(1)).GT.1.E-5) GO TO 60
      GO TO 75
52    IF(ABS(DELV(1)/(V(1,2)+0.5*DELV(1))) .GT. 0.02) GO TO 60
75    IF(NX.EQ.1) GO TO 90
      IF(NP.EQ.61) GO TO 90
      IF(ABS(V(NP,2)).LE.1.0E-03) GO TO 90
      CALL GROWTH
      IT=0
      IF(NX.GE.INV) FA=1
      GO TO 60
90    CALL OUTPUT
      GO TO 25
2500  FORMAT(*1*,16X,25HITERATIONS EXCEEDED ITMAX)
9100  FORMAT(*1*,4HNX =,I3,5X,3HX =,F10.3)
      END
```

```
      SUBROUTINE INPUT
      COMMON /BLCO/ NP,NX,NXT,NTR,INV,ETAE,VGP,CNU,DETA(61),A(61),
     1              ETA(61),DSD(61),GAMMA1,GAMMA2,UREF,REYN
      COMMON /BLCC/ X(60),UE(60),P1(60),P2(60),CEL(60),RX(60),CF,P1
     1              ,RTHETA(60)
      DIMENSION CP(60)
      ETAE=6.0
      READ(5,8000) NXT,NTR,INV,DETA(1),VGP,UREF,REYN
      NXM=INV-1
      DO 2 I=1,NXT
    2 READ(5,8200) X(I)
      DO 3 I=1,NXM
    3 READ(5,8200) CP(I)
      DO 4 I=INV,NXT
    4 READ(5,8200) DSD(I)
      CNU=UREF*X(NXT)/REYN
      WRITE(6,9000) NXT,NTR,INV,ETAE,DETA(1),VGP,REYN
      DO 5 J=1,NXM
    5 UE(J)=UREF*SQRT(1.0-CP(J))
      DO 80 I=2,NXM
      IF(I.EQ.NXM) GO TO 60
      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+1)-X(I-1))
      DUDS=-(X(I+1)-X(I))/A1*UE(I-1) + (X(I+1)-2.0*X(I)+X(I-1))/
     1       A2*UE(I) + (X(I)-X(I-1))/A3*UE(I+1)
      GO TO 70
60    A1=(X(I-1)-X(I-2))*( X(I)-X(I-2))
      A2=(X(I-1)-X(I-2))*(X(I)-X(I-1))


      A3=(X(I)-X(I-1))*(X(I)-X(I-2))
      DUDS=(X(I)-X(I-1))/A1*UE(I-2)-(X(I)-X(I-2))/A2*UE(I-1)+
     1       (2.0*X(I)-X(I-2)-X(I-1))/A3*UE(I)
70    P2(I)=X(I)/UE(I)*DUDS
80    CONTINUE
      P2(1)=1.0
      DO 90 I=1,NXM
90    P1(I)=0.5*(P2(I)+1.0)
      DO 8 J=1,NXM
      UEOUREF=UE(J)/UREF
    8 WRITE(6,9) J,X(J),CP(J),P2(J),UEOUREF
      DO 10 J=INV,NXT
   10 WRITE(6,11) J,X(J),DSD(J)
      RETURN
8000   FORMAT(3I5,4F10.0)
8200  FORMAT(F20.10)
9000  FORMAT(1H0,6HNXT   =,I3,14X,6HNTR   =,I3,14X,6HINV   =,I3/
     1         1H ,6HETAE  =,E14.6,3X,6HDETA1=,E14.6,3X,6HVGP   =,E14.6,
     1         6HREYN =,E14.6/)
    9 FORMAT(10X,*J=*,I3,4X,*X=*,E10.4,4X,*CP=*,E10.4,4X,
     1         *S/UXDUDS=*,E10.4,4X,*UEOUREF=*,E10.4)
   11 FORMAT(10X,*J=*,I3,4X,*X=*,E10.4,4X,*DELTASTARIN=*,E10.4)
      END
```

```
      SUBROUTINE GROWTH
      COMMON /BLCO/ NP,NX,NXT,NTR,INV,ETAE,VGP,CNU,DETA(61),A(61)
     1             ETA(61),DSU(61),GAMMA1,GAMMA2,UREF
      COMMON /BLCP/ F(61,2),U(61,2),V(61,2),W(61,2),B(61,2)
     1       ,DELV(61),DELF(61),DELU(61),DELW(61)
      NP0=NP
      NP1=NP+1
      NP=NP+1
      IF(NX.EQ.NTR) NP=NP+3
      IF(NP.GT.61) NP=61
      DO 35 J=NP1,NP
      F(J,1)=U(NP0,1)*(ETA(J)-ETA(NP0))+F(NP0,1)
      U(J,1)=U(NP0,1)
      V(J,1)=0.0
      B(J,1)=B(NP0,1)
      W(J,1)=W(NP0,1)
      F(J,2)=U(NP0,2)*(ETA(J)-ETA(NP0))+F(NP0,2)
      U(J,2)=U(NP0,2)
      V(J,2)=V(J,1)
      B(J,2)=B(NP0,2)
      W(J,2)=W(NP0,2)
   35 CONTINUE
      NNP=NP-(NP1-1)
      WRITE(6,6000) NNP
      RETURN
 6000 FORMAT(1H0,5X,13HETAE GROWTH -,I3,14H -POINTS ADDED)
      END
```

```
      SUBROUTINE GRID

      COMMON /BLCO/ NP,NX,NXT,NTR,INV,ETAE,VGP,CNU,DETA(61),A(61),
     1              ETA(61),DSD(61),GAMMA1,GAMMA2,UREF
        IF((VGP-1.0) .LE. 0.001) GO TO 5
      NP =ALOG((ETAE/DETA(1))*(VGP-1.0)+1.0)/ALOG(VGP)+1.0001
      GO TO 10
5     NP=ETAE/DETA(1) + 1.0001
10    IF(NP.LE.61) GO TO 15
      WRITE(6,9000)
      STOP
15    ETA(1)=0.0
      DO 20 J=2,61
      DETA(J)=VGP*DETA(J-1)
      A(J)=0.5*DETA(J-1)
20    ETA(J)=ETA(J-1)+DETA(J-1)
      RETURN
9000  FORMAT(1H0,36HNP EXCEEDED 61 --PROGRAM TERMINATED    )
      END
```

```
      SUBROUTINE EDDY1
      COMMON/BLCO/NP,NX,NXT,NTR,INV,ETAE,VGP,CNU,DETA(61),A(61),
     1            ETA(61),OSO(61),GAMMA1,GAMMA2,UREF,REYN
      COMMON/BLCC/X(60),UE(60),P1(60),P2(60),CEL(60),RX(60),CF,
     1            P1P,P2P,RTHETA(60)
      COMMON/BLCP/F(61,2),U(61,2),V(61,2),W(61,2),B(61,2)
     1            ,DELV(61),DELF(61),DELU(61),DELW(61)
      DIMENSION EDV(61)
      F1=0.0
      THE=0.0
      DO 30 J=2,NP
      F2=(U(J,2)/U(NP,2))*(1.0-U(J,2)/U(NP,2))
      THE=THE+0.5*(F1+F2)*DETA(J-1)
   30 F1=F2
      THE=THE/SQRT(REYN)
      RTHE=UE(NX)*THE/CNU
      IF(RTHE.GE.5000.) GO TO 40
      Z1=RTHE/425.-1.0
      PI=0.55*(1.0-EXP(SQRT(Z1)-0.298*Z1))
      A1=0.0168*1.55/(1.+PI)
      GO TO 45
   40 A1=0.0168
   45 CONTINUE
   20 IFLG=0
      RZ2=SQRT(REYN)
      RZ4=SQRT(RZ2)
      EDVO=A1*RZ2*(U(NP,2)*ETA(NP)-F(NP,2))
      J=1
   80 IF(IFLG.EQ.1) GO TO 90
      VABS=ABS(V(1,2))
      UEOUT=(UE(NX)/UREF)*RZ4/SQRT(VABS)
      PPLUS=P2(NX)*UEOUT**3/REYN
      PA=1.-11.8*PPLUS
      IF(PA.LE.0.0) PA=0.5


      YOA=RZ4*ETA(J)*SQRT(VABS*PA)/26.0
      EL=1.
      IF(YOA.LT.4.) EL=(1.-EXP(-YOA))
      EDVI=0.16*RZ2*ABS(V(J,2))*(EL*ETA(J))**2
      IF(EDVI.LT.EDVO) GO TO 100
      IFLG=1
   90 EDV(J)=EDVO
      GO TO 110
  100 EDV(J)=EDVI
      IF(J.LE.2) 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
      IF(EDV(J).LT.EDVO) GO TO 110
      EDV(J)=EDVO
      IFLG=1
  110 B(J,2)=1.0+EDV(J)
      J=J+1
      IF(J.LE.NP) GO TO 80
      RETURN
      END
```

```
      SUBROUTINE EDDY
      COMMON /BLCO/ NP,NX,NXT,NTR,INV,ETAE,VGP,CNU,DETA(61),A(61),
     1              ETA(61),LSD(61),GAMMA1,GAMMA2,UREF
      COMMON /BLCC/ X(60),UE(60),P1(60),P2(60),CEL(60),RX(60),CF,P1I
     1       ,PTHETA(60)
      COMMON /BLCP/ F(61,2),U(61,2),V(61,2),W(61,2),B(61,2),
     1    DELV(61),DELF(61),DELU(61),DELW(61)
      F1=0.0
      THE=0.0
      DO 30 J=2,NP
      F2=U(J,2)*(1.-U(J,2))
      THE=THE+0.5*(F1+F2)*DETA(J-1)
30    F1=F2
      THE=THE*X(NX)/SQRT(RX(NX))
      RTHE=UE(NX)*THE/CNU
      IF(RTHE.GE.5000.0) GO TO 40
      Z1=RTHE/425.-1.0
      PI=0.55*(1.-EXP(SQRT(Z1)-0.298*Z1))
      A1=0.0168*1.55/(1.+PI)
      GO TO 45


40    A1=0.0168
45    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
10    U1=U2
      GG=8.35E-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(6,9100) GG,UEINTG,EXPTM
      GO TO 20
15    GAMTR=1.0-EXP(-EXPTM)
20    CONTINUE
      IFLGD=0
      RX2=SQRT(RX(NX))
      RX4=SQRT(RX2)
      PPLUS=P2(NX)/(RX4*V(1,2)**1.5)
      RX216=RX2*0.16
      CN=SQRT(1.0-11.8*PPLUS)
      CRSQV=CN*RX4*SQRT(V(1,2))/26.0
      J=1
      EDVO=A1*RX2*(ETA(NP)-F(NP,2)+F(1,2))*GAMTR
50    IF(IFLGD.EQ.1) GO TO 100
      YOA=CRSQV*ETA(J)
      EDVI=RX216*ETA(J)**2*V(J,2)*(1.0-EXP(-YOA))**2*GAMTR
      IF(EDVI.LT.EDVO) GO TO 200
      IFLGD=1
100   EDV=EDVO
      GO TO 300
200   EDV=EDVI
300   B(J,2)=1.0+EDV
      J=J+1
      IF(J.LE.NP) GO TO 50
      RETURN
```

SUBROUTINE EDDY 74/74 OPT=1 PMDMP FTN 4.8+528

```
9100  FORMAT(1H0,2X,3HGG=,E14.6,3X,7HUEINTG=,E14.6,2X,6HEXPTM=,E14.6)
      END
```

```
      SUBROUTINE CMOM
      COMMON /BLCO/ NP,NX,NXT,NTR,INV,ETAE,VGP,CNU,DETA(61),A(61),
     1              ETA(61),USU(61),GAMMA1,GAMMA2,UREF
      COMMON /BLCC/ X(60),UE(60),P1(60),P2(60),CEL(60),RX(60),CF,P2
     1              ,RTHETA(60)
      COMMON /BLCP/ F(61,2),U(61,2),V(61,2),W(61,2),B(61,2)
     1      ,DELV(61),DELF(61),DELU(61),DELW(61)
      COMMON /BLCA/S1(61),S2(61),S3(61),S4(61),S5(61),S6(61),S7(61)
     1              S8(61),R1(61),R2(61),R3(61),R4(61)
      DATA GAMMA1/0.0/, GAMMA2/1.0/
      DO 100 J=2,NP
      USB=0.5*(U(J,2)**2+U(J-1,2)**2)
      FVB=0.5*(F(J,2)*V(J,2)+F(J-1,2)*V(J-1,2))
      FB=0.5*(F(J,2)+F(J-1,2))
      UB=0.5*(U(J,2)+U(J-1,2))
      VB=0.5*(V(J,2)+V(J-1,2))
      DERBV=(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
      CFVB=0.0
      CUSB=0.0
      GO TO 20
10    CFB=0.5*(F(J,1)+F(J-1,1))
      CVB=0.5*(V(J,1)+V(J-1,1))
      CFVB=0.5*(F(J,1)*V(J,1)+F(J-1,1)*V(J-1,1))
      CUSB=0.5*(U(J,1)**2+U(J-1,1)**2)
      CDERBV=(B(J,1)*V(J,1)-B(J-1,1)*V(J-1,1))/DETA(J-1)
20    S1(J)=B(J,2)/DETA(J-1)+(P1P*F(J,2)-CEL(NX)*CFB)*0.5
      S2(J)=-B(J-1,2)/DETA(J-1)+(P1P*F(J-1,2)-CEL(NX)*CFB)*0.5
      S3(J)=0.5*(P1P*V(J,2)+CEL(NX)*CVB)
      S4(J)=0.5*(P1P*V(J-1,2)+CEL(NX)*CVB)
      S5(J)=-P2P*U(J,2)
      S6(J)=-P2P*U(J-1,2)
      S7(J)=0.0
      S8(J)=0.0
      IF(NX.EQ.1) GO TO 30
      CLB=CDERBV+P1(NX-1)*CFVB+P2(NX-1)*(1.0-CUSB)
      CRB=-P2(NX)+CEL(NX)*(CFVB-CUSB)-CLB
      GO TO 35
   30 CRB=-P2(NX)
35    R2(J)=CRB-(DERBV+P1P*FVB-P2P*USB-CEL(NX)*(CFB*VB-CVB*FB))
      R1(J)=F(J-1,2)-F(J,2)+DETA(J-1)*UB
      R3(J-1)= U(J-1,2)-U(J,2)+DETA(J-1)*VB
      R4(J-1)=0.0
100   CONTINUE
      R1(1)=0.0
      R2(1)=0.0
      R3(NP)=0.0
      R4(NP)=0.0
      RETURN
      END
```

```
      SUBROUTINE ICONZ1
      COMMON/BLCO/NP,NX,NXT,NTP,INV,ETAE,VGP,CNU,DETA(61)
     1          ,A(61),ETA(61),DSO(61),GAMMA1,GAMMA2,UREF,REYN
      COMMON/BLCC/X(60),UE(60),P1(60),P2(60),CEL(60),RX(60),CF,P1
     1          ,RTHETA(60)


      COMMON/BLCA/S1(61),S2(61),S3(61),S4(61),S5(61),S6(61),
     1            S7(61),S8(61),R1(61),R2(51),R3(61),R4(61)
      COMMON/BLCP/F(61,2),U(61,2),V(61,2),W(61,2),
     1          B(61,2),DELV(61),DELF(61),DELU(61),DELW(61)
      BEL=1.0/(X(NX)-X(NX-1))
      GAMMA1=1.0
      GAMMA2=ETA(NP)-DSO(NX)*SQRT(REYN)
      DO 30 J=2,NP
      FLARE=1.C
      IF(U(J,2).LT.0) FLARE=0.
      FB=0.5*(F(J,2)+F(J-1,2))
      UB=0.5*(U(J,2)+U(J-1,2))
      VB=0.5*(V(J,2)+V(J-1,2))
      FVB=0.5*(F(J,2)*V(J,2)+F(J-1,2)*V(J-1,2))
      USB=0.5*(U(J,2)**2+U(J-1,2)**2)
      WSB=0.5*(W(J,2)**2+W(J-1,2)**2)
      DERBV=(B(J,2)*V(J,2)-B(J-1,2)*V(J-1,2))/
     1      DETA(J-1)
      CFB=0.5*(F(J,1)+F(J-1,1))
      CUB=0.5*(U(J,1)+U(J-1,1))
      CVB=0.5*(V(J,1)+V(J-1,1))
      CFVB=0.5*(F(J,1)*V(J,1)+F(J-1,1)*V(J-1,1
     1     ))
      CUSB=0.5*(U(J,1)**2+U(J-1,1)**2)
      CWSB=0.5*(W(J,1)**2+W(J-1,1)**2)
      CDERBV=(B(J,1)*V(J,1)-B(J-1,1)*V(J-1,1))/
     1      DETA(J-1)
      CRB=BEL*(CFVB+(CWSB-CUSB*FLARE))-CDERBV
      S1(J)=B(J,2)/DETA(J-1)+0.5*BEL*(F(J,2)-CFB)
      S2(J)=-B(J-1,2)/DETA(J-1)+0.5*BEL*(F(J-1,2)-CFB)
      S3(J)=C.*BEL*(V(J,2)+CVB)
      S4(J)=0.5*BEL*(V(J-1,2)+CVB)
      S5(J)=-BEL*U(J,2)*FLARE
      S6(J)=-BEL*U(J-1,2)*FLARE
      S7(J)=BEL*W(J,2)
      S8(J)=BEL*W(J-1,2)
      R1(J)=F(J-1,2)-F(J,2)+DETA(J-1)*UB
      R3(J-1)=U(J-1,2)-U(J,2)+DETA(J-1)*VB
      R2(J)=CRB-(DERBV+BEL*FVB+BEL*(WSB-USB*FLARE)-BEL*(CFB*VB
     1      -CVB*FB))
      R4(J-1)=0.0
   30 CONTINUE
      R1(1)=0.0
      R2(1)=0.0
      R3(NP)=W(NP,2)*GAMMA2-F(NP,2)
      R4(NP)=0.0
      RETURN
      END
```

```
      SUBROUTINE  SOLV4
      COMMON/BLCO/NP,NX,NXT,NTR,INV,ETAE,VGP,CNU,DETA(61),
     1            A(61),ETA(61),DSD(61),GAMMA1,GAMMA2,UREF
      COMMON/BLCC/X(60),UE(60),P1(60),P2(60),CEL(60),RX(60),CF,P1P,
     1            P2P,RTHETA(60)
      COMMON/BLCA/S1(61),S2(61),S3(61),S4(61),S5(61),S6(61),
     1            S7(61),S8(61),R1(61),R2(61),R3(61),R4(61)
      COMMON/BLCP/F(61,2),U(61,2),V(61,2),W(61,2),
     1            B(61,2),DELV(61),DELF(61),DELU(61),DELW(61)
      DIMENSION  A11(61),A12(61),A13(61),A14(61),A21(61),A22(61),
     1           A23(61),A24(61),G11(61),G12(61),G13(61),G14(61),
     2           G21(61),G22(61),G23(61),G24(61),W1(61),W2(61),
     3           W3(61),W4(61)
      GAMMA2=-GAMMA2
      A11(1)=1.0
      A12(1)=0.0
      A13(1)=0.0
      A14(1)=0.0
      A21(1)=0.0
      A22(1)=1.0
      A23(1)=0.0
      A24(1)=0.0
      W1(1)=R1(1)
      W2(1)=R2(1)
      W3(1)=R3(1)
      W4(1)=R4(1)
       DO 10 J=2,NP
      AA1=A13(J-1)-A(J)*A12(J-1)
      AA2=A23(J-1)-A(J)*A22(J-1)
      AA3=S2(J)-A(J)*S6(J)
      DET=AA2*A11(J-1)-AA1*A21(J-1)
      AJS=A(J)**2
      G11(J)=-(AA2+A21(J-1)*AJS)/DET
      G12(J)=(A11(J-1)*AJS+AA1)/DET



      G13(J)=A12(J-1)*G11(J)+A22(J-1)*G12(J)+A(J)
      G14(J)=A14(J-1)*G11(J)+A24(J-1)*G12(J)
      G21(J)=(S4(J)*AA2-A21(J-1)*AA3)/DET
      G22(J)=(A11(J-1)*AA3-S4(J)*AA1)/DET
      G23(J)=A12(J-1)*G21(J)+A22(J-1)*G22(J)-S6(J)
      G24(J)=A14(J-1)*G21(J)+A24(J-1)*G22(J)-S8(J)
      A11(J)=1.0
      A12(J)=-A(J)-G13(J)
      A13(J)=A(J)*G13(J)
      A14(J)=-G14(J)
      A21(J)=S3(J)
      A22(J)=S5(J)-G23(J)
      A23(J)=S1(J)+A(J)*G23(J)
      A24(J)=S7(J)-G24(J)
      W1(J)=R1(J)-G11(J)*W1(J-1)-G12(J)*W2(J-1)-W3(J-1)*G13(J)
     1        -G14(J)*W4(J-1)
      W2(J)=R2(J)-G21(J)*W1(J-1)-G22(J)*W2(J-1)-W3(J-1)*G23(J)
     1        -G24(J)*W4(J-1)
      W3(J)=R3(J)
      W4(J)=R4(J)
   10 CONTINUE
      D=GAMMA1*(A13(NP)*A[illegible](NP)*A23(NP)-A12(NP)*A23(NP)+
     1    A13(NP)*A22(NP))+GAMMA2*(A11(NP)*A23(NP)-A13(NP)*A21(NP))
```

```
      DF=W3(NP)*(A13(NP)*A24(NP)-A14(NP)*A23(NP)-A12(NP)*A23(NP)+
     1    A13(NP)*A22(NP))-GAMMA2*(W4(NP)*(A12(NP)*A23(NP)-A13(NP)
     2    *A22(NP))-W1(NP)*A23(NP)+W2(NP)*A13(NP))
      DU=W4(NP)*(GAMMA1*(A24(NP)*A13(NP)-A23(NP)*A14(NP))+GAMMA2*
     1    (A11(NP)*A23(NP)-A21(NP)*A13(NP)))+GAMMA1*(W2(NP)*A13(NP
     2    )-W1(NP)*A23(NP))-W3(NP)*(A13(NP)*A21(NP)-A11(NP)*A23(NP
      DV=GAMMA1*(W1(NP)*A24(NP)-W2(NP)*A14(NP))+W3(NP)*(A21(NP)*
     1    A14(NP)-A11(NP)*A24(NP))-GAMMA2*(W1(NP)*A21(NP)-W2(NP)*
     2    A11(NP))+W4(NP)*(GAMMA1*(A22(NP)*A14(NP)-A24(NP)*A12(NP))
     3    +GAMMA2*(A21(NP)*A12(NP)-A11(NP)*A22(NP)))+GAMMA1*(W1(NP)
     4    *A22(NP)-W2(NP)*A12(NP))+W3(NP)*(A21(NP)*A12(NP)-A11(NP)*
     5    A22(NP))
       DW=GAMMA1*(W2(NP)*A13(NP)-W1(NP)*A23(NP))+W4(NP)*(A12(NP)*
     1    A23(NP)-A13(NP)*A22(NP)))+W3(NP)*(A11(NP)*A23(NP)-
     2    A13(NP)*A21(NP))
      DET=D
      DELF(NP)=DF/DET
      DELU(NP)=DU/DET
      DELV(NP)=DV/DET
      DELW(NP)=DW/DET
      J=NP
   20 J=J-1
      CC1=DELU(J+1)-W3(J)-A(J+1)*DELV(J+1)
      CC2=DELW(J+1)-W4(J)
      CC3=A13(J)-A(J+1)*A12(J)
      CC4=W1(J)-A12(J)*CC1-A14(J)*CC2
      CC5=A23(J)-A(J+1)*A22(J)
      CC6=W2(J)-A22(J)*CC1-A24(J)*CC2
      DENO=A11(J)*CC5-A21(J)*CC3
      DELF(J)=(CC4*CC5-CC3*CC6)/DENO
      DELV(J)=(A11(J)*CC5-A21(J)*CC4)/DENO
      DELW(J)=CC2
      DELU(J)=CC1-A(J+1)*DELV(J)
      IF(J.GE.2) GO TO 20
      WRITE(6,9000) V(1,2),DELV(1)
      DO 30 J=1,NP
      F(J,2)=F(J,2)+DELF(J)


      U(J,2)=U(J,2)+DELU(J)
      V(J,2)=V(J,2)+DELV(J)
      W(J,2)=W(J,2)+DELW(J)
   30 CONTINUE
      IF(NX.GE.INV) UF(NX)=W(NP,2)*UREF
      RETURN
 9000 FORMAT(1H ,3X,6HV(1)= ,E14.6,2X,6HDELV1=,E14.6)
      END
```

```
      SUBROUTINE OUTPUT
      COMMON /BLCO/ NP,NX,NXT,NTR,INV,ETAE,VGP,CNU,DETA(61),A(61),
     1              ETA(61),USU(61),GAMMA1,GAMMA2,UREF,REYN
      COMMON /BLCC/ X(60),UE(60),P1(60),P2(60),CEL(60),RX(60),CF,P1
     1              ,RTHETA(60)
      COMMON /BLCP/ F(61,2),U(61,2),V(61,2),W(61,2),B(61,2)
     1              ,DELV(61),DELF(61),DELU(61),DELW(61)
      DIMENSION YODS(61),UP(61)
      IF(NX.GE.INV) GO TO 400
      F1=0.0
      THETA1=0.0
      DO 150 J=2,NP
      F2=U(J,2)*(1.-U(J,2))
      THETA1=THETA1+(F1+F2)*0.5*DETA(J-1)
150   F1=F2
      THETA=THETA1*X(NX)/SQRT(RX(NX))
      DELS=(ETA(NP)-F(NP,2))*X(NX)/SQRT(RX(NX))
      H =DELS/THETA
      CF=2.0*V(1,2)/SQRT(RX(NX))
      RTHETA(NX)=UE(NX)*THETA/CNU
      RDELS=UE(NX)*DELS/CNU
      DO 100 J=1,NP
      YODS(J)=(ETA(J)*X(NX)/SQRT(RX(NX)))/DELS
100   UP(J)=SQRT(UE(NX)/(X(NX)*CNU))*V(J,2)*DELS
       GO TO 600
400    THETA1=0.0
       F1=0.0
       DO 450 J=1,NP
       F2=(U(J,2)/U(NP,2))*(1.0-(U(J,2)/U(NP,2)))
       THETA1=THETA1+(F1+F2)*0.5*DETA(J-1)
450    F1=F2
       THETA=THETA1/SQRT(REYN)
       DELS=(ETA(NP)-(UREF/UE(NX))*F(NP,2))/SQRT(REYN)
       H=DELS/THETA
       CF=2.0*V(1,2)/SQRT(REYN)
       RTHETA(NX)=UREF*THETA/CNU



       RDELS=UREF*DELS/CNU
       DO 500 J=1,NP
       YODS(J)=(ETA(J)/SQRT(REYN))/DELS
500    UP(J)=SQRT(UREF/CNU)*V(J,2)*DELS
600    WRITE(6,4400)
      WRITE(6,4500)(J,ETA(J),F(J,2),U(J,2),V(J,2),B(J,2)
     1              ,UP(J),YODS(J),J=1,NP,4)
      WRITE(6,4500) NP,ETA(NP),F(NP,2),U(NP,2),V(NP,2),B(NP,2)
     1              ,UP(NP),YODS(NP)
      WRITE(6,9000) X(NX),THETA,DELS,H,CF,RX(NX),RTHETA(NX),RDELS,
     1              UE(NX),P2(NX)
210   NX=NX+1
      IF(NX.GT.NXT) GO TO 300
      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,2)=W(J,1)
250   B(J,1)=B(J,2)
      RETURN
300   CONTINUE

       STOP
4400   FORMAT(1H0,2X,1HJ,4X,3HETA,9X,1HF,13X,1HU,13X,1HV,13X,1HB,
     1        13X,7HOUDYODS,8X,4HYODS)
4500   FORMAT(1H ,I3,F10.3,6E14.6)
9000   FORMAT(1H0,6X,1HX,11X,5HTHETA,10X,4HDELS,11X,1HH,13X,2HCF/
     1        1H ,6X,1HR,13X,6HRTHETA,8X,5HRDELS,11X,2HUE,12X,2HP2/
     2        2H0,5E14.6/1H ,5E14.6/1X,"*********************************
     3*****************************************************************/
       END
```

7. REFERENCES

1. "Studies in Three-Dimensional Turbulent Boundary Layer Separation from Smooth Surfaces" Research Proposal submitted to the Lockheed-Georgia Company by the Georgia Tech Research Institute on March 24, 1981.

2. "An Improved Viscid/Inviscid Interaction Procedure for Transonic Flow over Airfoils" by Melnik, R.E., Chow, R.R., Mead, H.R. and Jameson A., Final Report for Contract No. NAS1-12426 (CR number waiting approval from NASA-Langley) 1981.

3. "Computation of Viscous/Inviscid Interactions" AGARD Conference Proceedings, No. 291, 1981.

4. "Analytic Investigation of Transonic Normal Shock-Boundary Layer Interaction" by W.H. Mason and G.R. Inger, VPI Engineering Report Aero-027, 1974.

5. "Experimental Data Base for Computer Program Assessment" AGARD Advisory Report No. 138, 1979.

6. "Validation of the GRUMFOIL Computer Code" by S. Ragab and S.G. Lekoudis, Lockheed-Georgia Engineering Report LG81-ER0093, 1981.

7. "Features of a Separating Turbulent Boundary Layer in the Vicinity of Separation" by Simpson, R.L., Strickland, J.H. and Barr, P.W., Journal of Fluid Mechanics, Vol. 79, Part 3, pp. 553-594, 1977.

8. "Analysis of Stalled Airfoils by Simultaneous Perturbations to Viscous and Inviscid Equations" by Gilmer, B.R. and Bristow, D.R., AIAA Paper 81-1239.

9. "Viscous-Inviscid Interaction Analyses of Transonic Turbulent Separated Flow" by J.E. Carter, AIAA Paper 81-1241.

10. "Transonic Viscous-Inviscid Interaction over Airfoils for Separated Laminar or Turbulent Flows" by Gordon, R. and Rom, J., AIAA Journal, Vol. 19, No. 5, pp. 545-552, 1981.

11. "Singularities in Three-Dimensional Turbulent Boundary-Layer Calculations and Separation Phenomena" by Cousteix, J. and Houdeville, R., AIAA Journal, Vol. 19, No. 8, pp. 976-985, 1981.

12. "Calculation of Turbulent-Inviscid Flow Interactions with Large Normal Pressure Gradients" by Mahgoub, H.E.H. and Bradshaw, P., AIAA Journal, Vol. 17, No. 10, pp. 1025-1029, 1979.

13. "Three-Dimensional Laminar boundary Layers and the OK of Accessibility" by Cebeci, T., A. Khattab and Stewartson, K., McDonnell-Douglas Report J 8716, 1980.

14. "Singularities in Three-Dimensional Turbulent Boundary Layer Calculations and Separation Phenomena" by J. Cousteix and R. Houdeville, AIAA J., Vol. 19, No. 8, pp. 976-985, 1981.

15. "Flow Separation on a Spheroid at Incidence" by T. Han and V.C. Patel, J. Fluid Mech., Vol. 92, Part 4, pp. 643-657, 1979.

16. "Mean Flow Measurements in the Three-Dimensional Boundary Layer over a Body of Revolution at Incidence" by B. Ramaprian and V.C. Patel and D.H. Choi, J. Fluid Mech., Vol. 103, pp. 479-504, 1981.

17. F. Thrasher of NSRDC. Personnel Communications, November 3, 1981.

18. Engineering Calculation Methods for Turbulent Flows, by P. Bradshaw, T. Cebeci and J. Whitelaw, Academic Press, 1981.