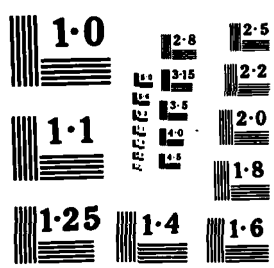
AD-A167 451 A COMPUTER PROGRAM FOR CALCULATING THE STEADY AND OSCILLATORY SUPERSONIC FLOW OVER THIN WINGS(U) AERONAUTICAL RESEARCH LABS MELBOURNE (AUSTRALIA) J A GEAR DEC 85 ARL-STRUC-TM-424 1/1

UNCLASSIFIED F/G 20/4 NL

END
DATE
FILMED
6 86


1·0
2·8
2·5
3·15
2·2
3·5
1·1
2·0
4·0
4·5
1·8
1·25
1·4
1·6

(12)

ARL-STRUC-TM-424 AR-004-037

AUSTRALIA

DEPARTMENT OF DEFENCE

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION

AERONAUTICAL RESEARCH LABORATORIES

MELBOURNE, VICTORIA

Structures Technical Memorandum 424

A COMPUTER PROGRAM FOR CALCULATING THE STEADY AND OSCILLATORY SUPERSONIC FLOW OVER THIN WINGS

by

J. A. GEAR

Approved for Public Release

DECEMBER 1985

AR-004-057

DEPARTMENT OF DEFENCE
DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION
AERONAUTICAL RESEARCH LABORATORIES

Structures Technical Memorandum 424

A COMPUTER PROGRAM FOR CALCULATING THE STEADY AND OSCILLATORY SUPERSONIC FLOW OVER THIN WINGS

by

J.A. GEAR

SUMMARY

A computer program for the calculation of linearised steady and oscillatory supersonic flow over thin wings has been implemented on the ELXSI. The program uses an explicit finite difference scheme on a characteristic grid to solve for the reduced potential and pressure coefficient on the wing surface.

POSTAL ADDRESS: Director, Aeronautical Research Laboratories,
P.O. Box 4331, Melbourne, Victoria, 3001, Australia.

CONTENTS

NOTATION

Symbol	Definition
AME	final Mach number
AMS	initial Mach number
AOME	final reduced frequency
AOMS	initial reduced frequency
BZ	semi span length
C(i,j,k)	f(i,j-1, k-1) + f(i,j-1, k+1) + f(i,j+1, k-1) + f(i,j+1, k+1)
C_p	steady pressure coefficient
DAM	delta change in Mach number
DAOM	delta change in reduced frequency
DX	characteristic grid spacing
f	reduced steady/oscillatory velocity potential
$\tilde{f}(i,j,k)$	$(1-p)\, f(i,j,k) + \frac{p}{2}\, [f(i+1, j, k) + f(i-1, j, k)]$
FXL (Y)	position of the leading edge
FXT (Y)	position of the trailing edge
h	vertical position of wing surface
h_o	oscillatory part of wing surface
$h_s^{\pm}$	steady upper/lower position of wing
H1 (X,Y)	$h_o/h_s{}^+{}_x$ for oscillatory/steady flow
H2 (X,T)	$h_{ox}/h_s{}^-{}_x$ for oscillatory/steady flow
IOUT	output parameter
ISW	option parameter
k	WM/α^2
K	k/zero for oscillatory/steady flow
l	typical chord length

NOTATION (CONT.)

Symbol	Definition
M	free stream Mach number
$M(i,j,k)$	$f(i,j-1,k) + f(i,j+1,k) + f(i,j,k-1) + f(i,j,k+1)$
p	constant/local pressure
P_∞	free stream pressure
Q	oscillatory pressure coefficient
U_∞	free stream velocity
x,y,z,t	Eulerian variables
α	$\sqrt{M^2 - 1}$
δ	thickness ratio
Δ	rectangular grid step length
η,ζ,τ	Prandtl-Glauert variables
λ	constant
$\rho\infty$	free stream density
ϕ	perturbation potential
ϕ_o	oscillatory perturbation potential
ϕ_s	steady perturbation potential
Φ	sealed oscillatory perturbation potential
ψ	velocity potential
ω	reduced frequency

1. INTRODUCTION

A computer program for the calculation of steady and oscillatory supersonic flow over thin wings has been implemented in Fortran on the ELXSI. As the basic differential equation is 'time-like' and the relevant part of the flow-field finite, the program uses an explicit finite-difference scheme to solve for the reduced potential on the wing.

In section 2 the governing differential equations and boundary conditions for steady and oscillatory supersonic flow over a thin wing are developed, and in section 3 the explicit finite difference formula applicable to linearised supersonic flow is presented. This formula has the advantage of using a characteristic grid rather than a rectangular grid thus halving the number of grid points. The method is restricted to boundary conditions defined on a plane. Section 4 discusses how to use the program, that is; setting up the data, compiling, binding and running. A listing of the program is given in Appendix A and finally a listing of a subroutine to read the output from the supersonic program is given in Appendix B, this subroutine is discussed in Section 4.

2. GOVERNING EQUATION AND BOUNDARY CONDITIONS

In the following, the isentropic inviscid flow of a perfect gas, initially irrotational, is considered. Thus we may assume the existence of a velocity potential ψ, such that the fluid velocity $\underline{v} = \underline{\nabla}\psi$. If U_∞ is the free stream velocity (which is assumed to be directed solely in the x- direction), then a perturbation potential ϕ can be defined such that

$$\psi\,(x,y,z,t) = U_\infty\,[x + \phi\,(x,y,z,t)], \qquad (1)$$

where x,y,z represent a rectangular carterian co-ordinate system and t is the time variable. Note that subsequent equations are expressed in non-dimensional co-ordinates based on a length scale ℓ, a typical value of the airfoil chord length, a velocity scale U_∞ and a density scale ℓ_∞, a typical value of the density at infinity. The time variable is then scaled with ℓ/U_∞ and the pressure with $\rho_\infty U_\infty^2$

For thin nearly planar wings of moderate aspect ratio the governing differential equation for the perturbation velocity potential can be written as

$$(M^2 - 1)\,\phi_{xx} - \phi_{yy} - \phi_{zz} + 2M^2\,\phi_{xt} + M^2\,\phi_{tt} = 0, \qquad (2)$$

where M is the free stream Mach number. It should be noted here that the procedure used to derive (2) is based upon the small parameter δ representing the ratio of airfoil thickness to chord length. It is assumed that as $\delta \to o$, ϕ/δ remains fixed. Then ignoring terms of second order in δ, we find that (2) is the appropriate governing differential equation for ϕ.

On a surface in an inviscid fluid the usual kinematic condition applies. If $Z = h(x,y,t)$ specifies the wing (where h is under δ by definition), then, to first order in δ, the boundary condition on the wing is

$$\phi_z /_{z=0} = \frac{\partial h}{\partial x} + \frac{\partial h}{\partial t} . \tag{3}$$

In many practical applications the unsteady motion of a wing may be assumed to consist of small infiniterimal perturbations around the steady state configuration. Thus the motion consists of a steady component plus a small harmonically oscillating unsteady component. Let

$$\phi\ (x,y,z,t) = \phi_s\ (x,y,z) + \phi_o\ (x,y,z)\ e^{inet}, \tag{4a}$$

$$h\ (x,y,t) = h_s^{\pm}\ (x,y) + h_o\ (x,y)\ e^{inet}, \tag{4b}$$

where, the subscript S/O denotes the steady/oscillatory part, $h_s^{\pm}$ represents the upper/lower surface of the wing and due to the sealings introduced above, ω represents the reduced frequency or Stroubal number. Note that the actual frequency is $\omega\ U_\infty / \ell$. Substituting (4a and b) into (2) and (3), and separating steady and oscillatory parts we find that

$$(M^2 - 1)\ \phi_{sxx} = \phi_{syy} + \phi_{szz}, \tag{5a}$$

$$\phi_{sz} /\ {z=0} = \frac{\partial}{\partial x}\ h_x^{\pm} \quad \text{on the wing,} \tag{5b}$$

and

$$(M^2 - 1)\ \phi_{oxx} + z_i\omega M^2\ \phi_{ox} - M^2\omega^2\ \phi_o$$

$$= \phi_{oyy} + \phi_{ozz}, \tag{6a}$$

$$\phi_{oz} /\ {z=0} = \frac{\partial h_o}{\partial x} + i\omega\ h_o \quad \text{on the wing.} \tag{6b}$$

At this stage it is convenient to introduce the generalised Prandtl-Glauert variables

$$\eta = \alpha y, \quad \zeta = az \quad \text{and} \quad \tau = a^2 t/m \tag{7a}$$

where $a^2 = M^2 - 1 > 0.$ (7b)

Also, in order to write (6a) in canonical form let

$$\phi_o = \Phi(x,\eta,\zeta) \quad e^{-ikmx}, \tag{8a}$$

where

$$k = \omega M/\alpha^2. \tag{8b}$$

Substituting (7a and b) into (5a and b), and, (7a and b) and (8a and b) into (6a and b) the steady and oscillatory problems become

$$\phi_{sxx} = \phi_{s\eta\eta} + \phi_{s\zeta\zeta}, \tag{9a}$$

$$\phi_{s\zeta/\zeta = 0\pm} = \frac{1}{\alpha} \quad h_{s\,x}^{\pm} \quad \text{on the wing} \tag{9b}$$

and

$$\Phi_{xx} + k^2 \Phi = \Phi_{\eta\eta} + \Phi_{\zeta\zeta}, \tag{10a}$$

$$\Phi_{\zeta/\zeta=0} = \frac{e^{ikmx}}{\alpha} \quad (h_{ox} + i\omega_{ho}) \quad \text{on the wing.} \tag{10b}$$

Finally note that if k is zero then the partial differential equation (10a) reduces to the same form as (9a). Also the boundary conditions (9b) and (10b) have similar forms. Due to these similarities both the steady and oscillatory problems can be solved by the same finite difference technique. In the next section we discuss the finite difference scheme that will be used to solve both (9a,b) and (10a,b).

3. FINITE DIFFERENCE SCHEME

The basic equation of linearised supersonic flow is

$$f_{xx} + K^2 f = f_{yy} + f_{zz} \tag{11}$$

where x is the flow direction, f the reduced perturbation velocity potential and K is a frequency parameter for unsteady flow, or zero for steady flow.

Let the potential be defined at grid points forming a cubic lattice at which it may be assumed that the co-ordinates take integer values (i,j,k). It has been shown by Sullivan [1] that a second order, consistent, finite difference scheme, with rotational symmetry about the x-axis and which reduces to the method of characteristics for two-dimensional flow in the x-y or x-z planes, is

$$\begin{aligned} &f(i+1,j,k) + f(i-1,j,k) + K^2\Delta^2 \tilde{f}(i,j,k) \\ &= -\lambda \; f(i,j,k) + \frac{\lambda}{2} \; M(i,j,k) + \left(\frac{1}{2} - \frac{\lambda}{4}\right) C(i,j,k) \end{aligned} \tag{12a}$$

where

$$\tilde{f}(i,j,k) = (1-p)\, f(i,j,k) + \frac{1}{2}\, P\, [f(i+1,j,k) + f(i-1,j,k)], \tag{12b}$$

$$\begin{aligned} M(i,j,k) &= f(i,j-1,k) + f(i,j+1,k) + f(i,j,k-1) \\ &\quad + f(i,j,K+1), \end{aligned} \tag{12c}$$

$$\begin{aligned} C(i,j,k) &= f(i,j-1,k-1) + f(i,j+1,k-1) \\ &\quad + f(i,j-1,k+1) + f(i,j+1,k+1), \end{aligned} \tag{12d}$$

where P and λ are constants and Δ is the grid step-length. It has also been shown by Sullivan [1] that the two-parameter family of finite difference schemes (12a,b,c and d) are strictly stable in the range

$$1 \geq \lambda\; 0 \quad , \quad p \geq \frac{1}{2} \, . \tag{13}$$

Following Sullivan [1] choose the simplest scheme $\lambda = 0$, $p = 1$ giving:

$$f(i+1,j,k) + f(i-1,j,k) = \frac{1}{(2+K^2\Delta^2)} \; C(i,j,k). \tag{14}$$

Note that (14) has the advantage that the terms f(i,j,k) and M(i,j,k) do not appear, giving a reduction in the number of grid points required. That is, the system (14) is solved on a characteristic grid rather than a rectangular grid.

Note that (14) is neutrally stable. This neutral stability will manifest itself as numerical instability as the number of grid points downstream becomes large.

Normal derivative boundary conditions can be implemented in a variety of ways. The technique used with (14) is to generate dummy potentials using the known normal derivates, with grid points on the boundary treated as interior points. That is

$$fz \,/_{\,z=0} = [f(i,j,2) - f(i,j,0)] \,/\, 2\Delta, \tag{15}$$

then on the boundary, level k = 1, (14) becomes

$$\begin{aligned} &f(i+1,j,1) + f(i-1,j,1) \\ &= \frac{2}{(2+K^2\Delta^2)} \left[f(i,j-1,2) + f(i,j+1,2) - \Delta f_z\,(i,j-1) \right. \\ &\qquad \left. - \Delta f_z\,(i,j+1) \right]. \end{aligned} \tag{16}$$

4. DISCUSSION

The supersonic program solves accurately the partial differential equations derived in Section 2. However, we should note that the equations (9a,b) and (10a,b) are approximate representations of the flow about a thin wing. Thus the numerical results should be viewed as approximations to the flow, though the results should be reasonably accurate, provided the thickness parameter δ, is small and the Mach number is not too close to unity.

The program offers a resolution of up to 291 x 291 grid points on the x-y plane. From experience with the program 50 x 50 grid points gives excellent agreement (i.e. four significant figures) with the assymptotic solutions for an infinite rectangular wing, at various Mach numbers and a sonic triangular wing.

It should be stated that the program uses very little central processing time. When the functions are given by Table 1 and the data by Table 2 (i.e. the sonic triangular wing) the running time is 7.2 c.p.u. seconds. If the grid step length is halved, that is put DX equal to 0.01, then the running time is 30.6 c.p.u seconds. Also, note that binding takes about six c.p.u. seconds.

Finally note that the program is capable of handling quite elaborate wing planforms. For instance; forward swept wings and wings with monotonically curved or convex leading and trailing edges; however, the program will in most cases be unable to determine the proper grid system for wings with more than one extreme leading or trailing point.

REFERENCE

[1] M.C.W. SULLIVAN, "Explicit finite difference methods for linearised oscillatory supersonic flow" British Aerospace, BAe - KSD - R - GEN - 1155, August 1983.

TABLE 1: Example of function subroutines FXL(Y), FXT(Y), HI(X,Y) and H2(X,Y)

FUNCT.F

```
      REAL FUNCTION FXL(Y)
      FXL=Y
      RETURN
C   Alternate entry for fxt.
      ENTRY FXT(Y)
      FXT=1.0
      RETURN
      END

      REAL FUNCTION H1(X,Y)
      PARAMETER (PI=3.1415926)
      DATA THETA/0.5/
      H1=-1.0*X*TAN(THETA*PI/180.0)
      RETURN
C   Entry for h2.
      ENTRY H2(X,Y)
      H2=-1.0*TAN(THETA*PI/180.0)
      RETURN
      END
```

TABLE 2: Example of input data for supersonic program. The first two parameters have Fortran format type I2, while, the rest have format type F12.8

TEST1.D

1	ISW
3	IOUT
0.02	DX
1.0	B2
1.4142196	AMS
1.4142196	AME
0.1	DAM
0.1	AOMS
0.1	AOME
0.1	DAOM

TABLE 3: Example of output when the functions in Figure 1 and the input data of Figure 2, are used

```
SONIC1.D

1     1
    Unsteady supersonic.

          Grid step size = .20000E-01          Semi span length = 1.0000
  Initial Mach no. = 1.41422   Final Mach no. = 1.41422   Mach no. step =   .10000
  Initial freq. =   .10000  Final freq. =   .10000  Freq. step =   .10000
1

      50    3
  Mach No. =1.41422     Frequency =   .10000


y =     .0100         Points =   50
x          .01000       .03000       .05000       .07000       .09000       .11000       .13000....
Phi'    .0000+000    .1745-003    .3054-003    .4254-003    .5400-003    .6545-003    .7670-003....
Phi"    .0000+000    .7958-012   -.1745-006   -.2618-006   -.3926-006   -.4908-006   -.6134-006....

"etc"
```

APPENDIX A

SUPERSONIC.F

```
      PROGRAM SUPERSONIC
      COMMON /SUPER/ AL,DX,B2,GS,AM,IM,JC
      EXTERNAL FXL,FXT,H1,H2
      INTEGER IS(291),IE(291),JS(291,291),JE(291,291)
      DIMENSION XL(581),XT(581),XX(291)
      COMPLEX G(581,146),F(291,291),FG(291),ADUM
      CHARACTER*4 CH1(4),CH2(4)
      DATA CH1/'Phi''','Q''  ','Phi ','Cp  '/
      DATA CH2/'Phi"','Q"  ','Phi-','Cp- '/
      PARAMETER (N=291,N2=581,N3=146)
 41   FORMAT(2(I2/),8(F12.8/))
 42   FORMAT('1',5X,'1',//,5X,'Unsteady supersonic.',//)
 43   FORMAT('1',5X,'2',//,5X,'Steady supersonic.',//)
 44   FORMAT('1',5X,'3',//,5X,'Steady supersonic (symmetric).',//)
 45   FORMAT(11X,'Grid step size = ',E10.5,10X,'Semi span length = ',
     #F6.4,//,2X,'Initial Mach no. = ',F7.5,3X,'Final Mach no. = ',
     #F7.5,3X,'Mach no. step = ',F7.5/)
 46   FORMAT(2X,'Initial freq. = ',F7.5,2X,'Final freq. = ',F7.5,2X,
     #'Freq. step = ',F7.5/)
 47   FORMAT('1',:,///,6X,I3,4X,I1,//,'  Mach No. =',F7.5,:,5X,
     #'Frequency = ',F7.5)
 48   FORMAT(//,' y = ',F9.4,10X,'Points =',I4)
 49   FORMAT(' x      ',3X,10(F8.5,4X))
 50   FORMAT(1X,A4,1X,10(E11.4,1X))
C
C    Main program for supersonic flow over a thin airfoil.
C
C    ISW = 1  for unsteady supersonic flow
C    ISW = 2  for steady supersonic flow (unsymmetric)
C    ISW = 3  for steady supersonic flow (symmetric)
C
C    Output para. IOUT determines type of output;
C    IOUT = 1 for pressure coefficient only,
C    IOUT = 2 for potential only,
C    IOUT = 3 for both potential and pressure.
C
C    DX = step size,  B2 = semi span length
C    AMS = initial Mach no.,  AME = final Mach no.
C    DAM = delta Mach no.
C    If ISW = 1 then AOMS = initial freq.,  AOME = final freq.
C    and DAOM = delta freq..
C    Leading /trailing edge data is in FXL/FXT.
C    H1 represents either ho or d/dx.hs+.
C    H2 represents either d/dx.ho or d/dx.hs-.
C
      READ(5,41,ERR=70,END=71)ISW,IOUT,DX,B2,AMS,AME,DAM,AOMS,AOME,DAOM
      GO TO 71
 70   CONTINUE
      STOP '  Error in input data.'
 71   CONTINUE
      IF  (AMS.LE.1.0.OR.AME.LE.1.0)  STOP '  Invalid Mach no. range.'
      IF  (ISW.LT.1.OR.ISW.GT.3) STOP '  Invalid switch parameter in
     # input data.'
      IF  (ISW.EQ.1)  PRINT 42
      IF  (ISW.EQ.2)  PRINT 43
      IF  (ISW.EQ.3)  PRINT 44
      PRINT 45,DX,B2,AMS,AME,DAM
      IF  (ISW.EQ.1)  PRINT 46,AOMS,AOME,DAOM
      DUM=AME-AMS
      IF  (DUM.NE.0.0)  DAM=SIGN(DAM,DUM)
      AM=AMS
 72   CONTINUE
      AL=SQRT(AM*AM-1.0)
      CALL GRIDXY(N,N2,XL,XT,IS,IE,JS,JE)
      IF  (ISW.EQ.1)  GO TO 73
      AK=0.0
      AOM=0.0
      PRINT 47,JC,IOUT,AM
      GO TO 75
 73   CONTINUE
      DUM2=AOME-AOMS
      IF  (DUM2.NE.0.0) DAOM=SIGN(DAOM,DUM2)
      AOM=AOMS
 74   CONTINUE
      AK=AOM*AM/AL/AL
      PRINT 47,JC,IOUT,AM,AOM
 75   CONTINUE
      CALL SOLVE(ISW,N,N2,N3,JS,JE,AK,XL,XT,F,G)
      CALL OUTPUT(N,N2,XL,XT,XX,F,IS,IE,AK)
```

SUPERSONIC.F

```
      IC=4-IABS(5-2*ISW)
      IF  (IOUT.EQ.1) GO TO 77
      DO 76 J=1,JC
        IF  (IE(J).LT.IS(J))  GO TO 76
        Y=FLOAT(2*J-1)*DX/AL/2.0
        ITT=IE(J)-IS(J)+1
        NUM=INT(FLOAT(ITT+8)/9.0)
        IF  (NUM.EQ.0)  GO TO 76
        PRINT 48,Y,ITT
        DO 76 I=1,NUM
          JB=(I-1)*9+IS(J)
          JT=MIN(IE(J),I*9+IS(J)-1)
          PRINT 49,(XX(K),K=JB,JT)
          PRINT 50,CH1(IC),(REAL(F(K,J)),K=JB,JT)
          IF (ISW.NE.3) PRINT 50,CH2(IC),(AIMAG(F(K,J)),K=JB,JT)
 76     CONTINUE
 77   CONTINUE
      IF  (IOUT.EQ.2) GO TO 80
      IC=IC+1
      IF  (IOUT.NE.1)  PRINT 47
      ADUM=CMPLX(0.0,-2.0*AOM*DX)
      DO 79  J=1,JC
        IF  (IE(J).LE.IS(J)+1)  GO TO 79
        DO 78 I=IS(J)+1,IE(J)-1
          FG(I)=(F(I-1,J)-F(I+1,J)+ADUM*F(I,J))/DX
 78     CONTINUE
        Y=FLOAT(2*J-1)*DX/AL/2.0
        ITT=IE(J)-IS(J)-1
        NUM=INT(FLOAT(ITT+8)/9.0)
        PRINT 48,Y,ITT
        DO 79 I=1,NUM
          JB=(I-1)*9+IS(J)+1
          JT=MIN(IE(J)-1,I*9+IS(J))
          PRINT 49,(XX(K),K=JB,JT)
          PRINT 50,CH1(IC),(REAL(FG(K)),K=JB,JT)
          IF (ISW.NE.3) PRINT 50,CH2(IC),(AIMAG(FG(K)),K=JB,JT)
 79   CONTINUE
 80   CONTINUE
      IF  (ISW.NE.1)  GO TO 81
      AOM=AOM+DAOM
      IF  (DUM2*(AOM-AOME).LE.0.0.AND.DUM2*DAOM.NE.0.0)  GO TO 74
 81   CONTINUE
      IF  (DAM.EQ.0.0)  STOP '  Mach no. step size is zero.'
      AM=AM+DAM
      IF  (DUM*(AM-AME).LE.0.0.AND.DUM.NE.0.0)  GO TO 72
      STOP '  Normal completion.'
      END

      SUBROUTINE GRIDXY(N,N2,XL,XT,IS,IE,JS,JE)
      COMMON/SUPER/ AL,DX,B2,GS,AM,IM,JC
      INTEGER IS(N),IE(N),JS(N,N),JE(N,N)
      DIMENSION XL(N2),XT(N2)
C
C     Subroutine to initialize grid system.
C
C     FIND EXTREME FORWARD AND TRAILING POINTS
C     Leading and trailing edge data is in function subroutines
C     FXL and FXT (in Eulerian co-ordinates).
C
      DO 900 J=1,N2
        XL(J)=0.0
        XT(J)=0.0
 900  CONTINUE
      D2=DX/AL/2.0
      JJS=1
      JJE=1
      GS=FXL(D2)
      GE=FXT(D2)
      XL(2)=GS
      XT(2)=GE
      J=2
 901  J=J+2
      Y=FLOAT(J-1)*D2
      IF  (Y.GT.B2)  GO TO 902
      XS=FXL(Y)
      XE=FXT(Y)
      XL(J)=XS
      XT(J)=XE
      IF  (XS.LT.GS)  THEN
```

SUPERSONIC.F

```
          GS=XS
          JJS=J/2
      END IF
      IF  (XE.GT.GE)  THEN
          GE=XE
          JJE=J/2
      END IF
      GO TO 901
 902  CONTINUE
      JC=-1+J/2
      DO 903 J=2,2*JC,2
        XL(J)=XL(J)-GS
        XT(J)=XT(J)-GS
 903  CONTINUE
      DO 904 J=1,2*JC-1,2
        Y=FLOAT(J-1)*D2
        XL(J)=FXL(Y)-GS
        XT(J)=FXT(Y)-GS
 904  CONTINUE
      Y=FLOAT(2*JC)*D2
      IF  (Y.LE.B2)  THEN
          XL(2*JC+1)=FXL(Y)-GS
          XT(2*JC+1)=FXT(Y)-GS
      END IF
C
C   Find leading edge/characteristic grid points.
C
      IS(JJS)=1
      IF  (JJS.NE.1)  THEN
          DO 906 J=JJS-1,1,-1
            IS(J)=IS(J+1)+1
            DUM=XL(2*J)/DX+2.0
 905        CONTINUE
            IF  (DUM.LE.FLOAT(IS(J)))  THEN
                IS(J)=IS(J)-1
                GO TO 905
            END IF
 906      CONTINUE
      END IF
      IF  (JJS.NE.JC)  THEN
          DO 908 J=JJS+1,JC
            IS(J)=IS(J-1)+1
            DUM=XL(2*J)/DX+2.0
 907        CONTINUE
            IF  (DUM.LE.FLOAT(IS(J)))  THEN
                IS(J)=IS(J)-1
                GO TO 907
            END IF
 908      CONTINUE
      END IF
C
C    Find trailing edge/characteristic grid points.
C
      IE(JJE)=1+INT((GE-GS)/DX)
      IF  (JJE.NE.1)  THEN
          DO 910 J=JJE-1,1,-1
            IE(J)=IE(J+1)-1
            DUM=XT(2*J)/DX
 909        CONTINUE
            IF  (DUM.GE.FLOAT(IE(J)))  THEN
                IE(J)=IE(J)+1
                GO TO 909
            END IF
 910      CONTINUE
      END IF
      IF  (JJE.NE.JC)  THEN
          DO 912 J=JJE+1,JC
            IE(J)=IE(J-1)-1
            DUM=XT(2*J)/DX
 911        CONTINUE
            IF  (DUM.GE.FLOAT(IE(J)))  THEN
                IE(J)=IE(J)+1
                GO TO 911
            END IF
 912      CONTINUE
      END IF
      J=JC+1
 919  J=J-1
      IF  (IE(J).LT.IS(J))  GO TO 919
```

SUPERSONIC.F

```
      JC=J
C
C     FIND TIP DIAPRAGHM REGION
C
      IJ=INT((IE(JC)-IS(JC))/2.0)
      JM=JC+IJ
      IF  (IJ.LE.0)  GO TO 918
      DO 913 J=JC+1,JM
        IS(J)=IS(J-1)+1
        IE(J)=IE(J-1)-1
 913  CONTINUE
C
C     Find j-starting and stopping grid points at all
C     vertical levels.
C
 918  IM=IE(JJE)
        DO 916 I=1,IM
          JSW=1
          J=0
 914      J=J+JSW
          IF  (J.GT.JM.OR.J.LT.1)  GO TO 916
          IF  (I.LT.IS(J).OR.I.GT.IE(J))  GO TO 914
          IF  (JSW.EQ.-1)  GO TO 915
          JS(1,I)=2*J
          JSW=-1
          J=JM+1
          GO TO 914
 915      JE(1,I)=2*J
 916  CONTINUE
      DO 917 K=2,IM
        K1=K-1
        IK=MOD(K,2)
        DO 917 I=1,IM-K+1
          JS(K,I)=JS(K1,I-IK+1)+2*IK-1
          JE(K,I)=MIN(JE(K1,I),JE(K1,I+1))+1
 917  CONTINUE
      RETURN
      END

      SUBROUTINE SOLVE(ISW,N,N2,N3,JS,JE,AK,XL,XT,F,G)
      COMMON/SUPER/ AL,DX,B2,GS,AM,IM,JC
      INTEGER JS(N,N),JE(N,N)
      COMPLEX F(N,N),G(N2,N3),DUM,DUM2
      DIMENSION XL(N2),XT(N2)
C
C      Subroutine to initialise wing boundary conditions, grid points
C      and integrate the unsteady/steady supersonic problem.
C
C      Initialize boundary conditions.
C
      AOM=AK*AL*AL/AM
      AKM=AK*AM
      DD=DX/AL
      COEF=4.0/(8.0+AK*AK*DX*DX)
      DUM2=CMPLX(0.0,0.0)
      DO 498 I=1,IM
        XX=FLOAT(2*I-3)*DX/2.0
        X=XX+GS
        DUM=CMPLX(COS(AKM*X),SIN(AKM*X))*DD
        JST=MAX(1,JS(1,I)-1)
        DO 498 J=JST,JE(1,I)+1,2
          Y=FLOAT(J-1)*DD/2.0
          JJ=(J+1)/2
          IF  (Y.GT.B2.OR.XX.LT.XL(J).OR.XX.GT.XT(J))  THEN
              F(I,JJ)=DUM2
          ELSE IF (ISW.EQ.1)  THEN
              F(I,JJ)=DUM*CMPLX(H2(X,Y),H1(X,Y)*AOM)
          ELSE IF (ISW.EQ.2)  THEN
              F(I,JJ)=DD*CMPLX(H1(X,Y),-1.0*H2(X,Y))
          ELSE
              F(I,JJ)=DD*CMPLX(H1(X,Y),0.0)
          END IF
 498  CONTINUE
      DO 499 I=1,IM
        DO 499 J=JS(1,I),JE(1,I),2
          JJ=J/2
          F(I,JJ)=F(I,JJ)+F(I,JJ+1)
 499  CONTINUE
C
```

SUPERSONIC.F

```
C     Initialize integration points to zero.
C
      DO 550 I=1,N2
        DO 550 J=1,N3
          G(I,J)=CMPLX(0.0,0.0)
 550  CONTINUE
C
C    Integrate the unsteady and steady supersonic problem.
C
      ISTOP=2*IM-1
      II=0
 551  II=II+1
      KE=IM-IABS(II-IM)
      IK=(II+1)/2
      DO 552 J=JS(1,IK),JE(1,IK),2
        J1=J+1
        J2=J-1
        JJ=J/2
        DUM=2.0*COEF*(G(J1,1)+G(J2,1))-G(J,1)-COEF*F(IK,JJ)
        IF  (ISW.EQ.2.AND.(J.GT.2*JC.OR.FLOAT((II+1)/2).LT.
     1  XL(J)/DX))   THEN
            DUM2=0.5*(AIMAG(DUM)+REAL(DUM))
            DUM=CMPLX(DUM2,DUM2)
        END IF
        F(IK,JJ)=DUM
        G(J,1)=DUM
 552  CONTINUE
      DO 553 K=3,KE,2
        IK=(II-K+2)/2
        K1=(K+1)/2
        K2=K1-1
        DO 553 J=JS(K,IK),JE(K,IK),2
          J1=J+1
          J2=J-1
          G(J,K1)=COEF*(G(J1,K1)+G(J2,K1)+G(J1,K2)+G(J2,K2))-G(J,K1)
 553  CONTINUE
      II=II+1
      IF  (II.GT.ISTOP)  RETURN
      KE=IM-IABS(II-IM)
      DO 554 K=2,KE,2
        K1=K/2
        K2=K1+1
        IK=(II-K+2)/2
        DO 554 J=JS(K,IK),JE(K,IK),2
          J1=J+1
          J2=J-1
          IF  (J.EQ.1) J2=2
          G(J,K1)=COEF*(G(J1,K1)+G(J2,K1)+G(J1,K2)+G(J2,K2))-G(J,K1)
 554  CONTINUE
      GO TO 551
      END

      SUBROUTINE OUTPUT(N,N2,XL,XT,XX,F,IS,IE,AK)
      COMMON /SUPER/ AL,DX,B2,GS,AM,IM,JC
      REAL XL(N2),XT(N2),XX(N)
      INTEGER IS(N),IE(N)
      COMPLEX F(N,N)
      DO 930 I=1,IM
        XX(I)=FLOAT(I-1)*DX+GS
 930  CONTINUE
      AKM=-1.0*AK*AM
      DO 931 J=1,JC
        ARG=XL(2*J)/DX
        I1=2+INT(ARG)
        IF  (MOD(ARG,1.0).EQ.0.0)  I1=I1-1
        I2=INT(XT(2*J)/DX)+1
        IS(J)=I1
        IE(J)=I2
        DO 931 I=I1,I2
          ARG=AKM*XX(I)
          F(I,J)=F(I,J)*CMPLX(COS(ARG),SIN(ARG))
 931  CONTINUE
      RETURN
      END
```

APPENDIX B

SUBREAD.F

```
      SUBROUTINE SUPDAT(M,N,DX,B2,ISW,IOUT,IC,AM,AOM,JC,JP,IP1,IP2,
     #Y1,Y2,X1,X2,F,G,*)
      DIMENSION Y1(M,N),Y2(M,N),X1(M,N,N),X2(M,N,N),AM(M),AOM(M)
      INTEGER IP1(M,N),IP2(M,N),JC(M),JP(M)
      COMPLEX F(M,N,N),G(M,N,N)
 40   FORMAT(6X,I1)
 41   FORMAT(4(/),28X,E10.5,29X,F6.4,9(/),7X,I3,3X,I1,//,12X,F7.5,
     #17X,F7.5)
 42   FORMAT(4(/),28X,E10.5,29X,F6.4,7(/),7X,I3,3X,I1,//,12X,F7.5)
 43   FORMAT(//,5X,F9.4,18X,I4)
 44   FORMAT(9X,9(F8.5,4X,:))
 45   FORMAT(6X,9(E11.4,1X,:))
 46   FORMAT()
 47   FORMAT(//)
 48   FORMAT(7X,I3,//,12X,F7.5,17X,F7.5)
 49   FORMAT(7X,I3,//,12X,F7.5)
      OPEN(1,FILE='SUPER.D',STATUS='OLD')
      IC=1
      READ(1,40,ERR=29)ISW
      IF  (ISW.EQ.1)  READ(1,41,ERR=29)DX,B2,JC(1),IOUT,AM(1),AOM(1)
      IF  (ISW.NE.1)  READ(1,42,ERR=29)DX,B2,JC(1),IOUT,AM(1)
 22   CONTINUE
      IF  (IOUT.EQ.1)  GO TO 24
      DO  23 J=1,JC(IC)
        READ(1,43,ERR=29)Y2(IC,J),ITT
        NUM=INT(FLOAT(ITT+8)/9.0)
        IP2(IC,J)=ITT
        DO 23 K=1,NUM
          IS=K*9-8
          IE=MIN(ITT,K*9)
          READ(1,44,ERR=29)(X2(IC,I,J),I=IS,IE)
          READ(1,45,ERR=29)(REAL(F(IC,I,J)),I=IS,IE)
          IF (ISW.NE.3) READ(1,45,ERR=29)(AIMAG(F(IC,I,J)),I=IS,IE)
 23   CONTINUE
      IF  (IOUT.EQ.2)  GO TO 26
      READ(1,46)
 24   CONTINUE
      DO  25 J=1,JC(IC)
        READ(1,43,END=28,ERR=29)Y1(IC,J),ITT
        IF  (ITT.LE.0)  GO TO 27
        NUM=INT(FLOAT(ITT+8)/9.0)
        IP1(IC,J)=ITT
        JP(IC)=J
        DO 25 K=1,NUM
          IS=K*9-8
          IE=MIN(ITT,K*9)
          READ(1,44,ERR=29)(X1(IC,I,J),I=IS,IE)
          READ(1,45,ERR=29)(REAL(G(IC,I,J)),I=IS,IE)
          IF (ISW.NE.3) READ(1,45,ERR=29)(AIMAG(G(IC,I,J)),I=IS,IE)
 25   CONTINUE
 26   CONTINUE
      READ(1,47)
 27   CONTINUE
      IC=IC+1
      IF  (IC.GT.M)  GO TO 30
      IF  (ISW.EQ.1)  READ(1,48,END=30,ERR=29)JC(IC),AM(IC),AOM(IC)
      IF  (ISW.NE.1)  READ(1,49,END=30,ERR=29)JC(IC),AM(IC)
      GO TO 22
 30   CONTINUE
      IC=IC-1
 28   CONTINUE
      CLOSE(1)
      RETURN
 29   CONTINUE
      CLOSE(1)
      RETURN 1
      END
```

APPENDIX C

Description of Computer Program

A computer program based on the finite difference scheme (14) and (16) has been implemented in Fortran on the ARL ELXSI computer. The program consists of a main program called SUPERSONIC and three subroutines, GRIDXY, SOLVE and OUTPUT. For completeness a listing is supplied in Appendix A. The position of the leading and trailing edge, and, either, h_o and h_{ox} or $h_s{}^+{}_x$ and $h_s{}^-{}_x$ must be supplied by the user in function subroutines called FXL(Y), FXT(Y), H1(X,Y) and H2(X,Y). For an example see Table 1. Note that X and Y in these functions are sealed Eulerian variables and not Prandtl - Glauert variables.

The program has three options; unsteady flow, steady flow and steady symmetric flow (i.e. hs^- = $-hs^+x$). . For unsteady flow H1(x,y) and H2(x,y) represent $h_o(x,y)$ and $h_{ox}(x,y)$ respectively. The output for the unsteady case is the real and imaginary parts of ϕ_o (x,y,o) (see (4a)) and Q(x,y,o), where the unsteady pressure coefficient is defined as

$$\frac{(P- P_\infty)}{\frac{1}{2}\ \rho_\infty\ U_\infty^2} = Q\ (x,y,o)\ e^{i\omega t}\ . \qquad (17)$$

For the steady case H1 and H2 represent $h_s{}^+x$ and $h_s{}^-x$. The output for this case is

$\phi_s\ (x,y,o^+)$, $\phi_s\ (x,y,o^-)$, $C_p\ (x,y,o^+)$ and $C_p\ (x,y,o^-)$,

where in the steady case,

$$\frac{(P- P_\infty)}{\frac{1}{2}\ \rho_\infty\ U_\infty^2} = C_p\ (x,y,o^\pm)\ . \qquad (18)$$

For the steady symmetric case, the flow below the x-y plane equals the flow above. The output for this case is just $\phi_s\ (x,y,o^+)$ and $C_p\ (x,y,o^+)$.

After FXL, FXT, H1 and H2 have been programmed, compile the functions with optimiser switch off, that is

```
FORTRAN MYFUNCTIONS          -OPT.
```

Then bind the programs in the following way:

```
BIND      /USER/ST. GEAR/SUPERSONIC,MYFUNCTIONS
BFILE  =  MY TITLE.
```

The executable file will then be called MYTITLE and it will exist in your area. To run SUPERSONIC, extra data must be supplied through an input file. Table 2 shows a typical example of the data required by SUPERSONIC.

The first parameter in Table 2 (ISW) has format type I2 and takes the values 1, 2 or 3. That is ISW equals one for unsteady flow, two for steady flow and three for steady symmetric flow. The second parameter IOUT also has format type I2 and it also takes the values 1,2 or 3. It represents the type of output. For instance if IOUT equals one then only the pressure coefficient is output. If IOUT equals two then only the potential is printed. If IOUT is three then both the potential and pressure coefficient are printed. The third parameter DX has format type

F12-8, it determines the distance between grid points on the characteristic grid. Note that DX is twice Δ (see (12a)). When FXL and FXT are given as in Figure 1, the Mach number is $\sqrt{2}$ and DX = 0.02 as in Table 2, then the grid system on the x-y plane has 50 x 50 points. Note that SUPERSONIC can take a grid size on the x-y plane of up to 291 x 291 points. Also note that if an "ACCESS VIOLATION" error occurs while running SUPERSONIC, then the grid size is greater than 291 x 291 points. To remedy this situation simply increase DX. The fourth parameter B2 also has format type F12.8, it represents the sealed Eulerian semi-span length. The fifth, sixth and seventh parameters determine the initial, final and delta change in Mach number. That is AMS is the initial Mach number, AME is the final Mach number and DAM is the delta change in Mach number. Thus with a suitable choice of AMS, AME and DAM, SUPERSONIC will produce data for several different Mach numbers, during one run. Note that AMS, AME and DAM have format type F12.8. Also note that if AMS and AME are less than or equal to one, then SUPERSONIC will not execute. The last three parameters are only used if ISW equals 1. AOMS represents the initial reduced frequency, AOME the final reduced frequency and DAOM the delta change in reduced frequency. AOMS, AOME and DAOM have format type

F12.8. Note that for each Mach number AOMS, AOME and DAOM determine a range of reduced frequencies. Also note that if ISW is not equal to one then these parameters need not be supplied.

Once the input data have been programmed into a file called, for instance, MYDATA, then SUPERSONIC can be executed by entering:

MYDATA > MYTITLE > MYOUTPUT,

where MYOUTPUT represents the file to which the output is printed. Note that if there is an error in the input data SUPERSONIC will stop execution and print one of three statements. For instance; "invalid switch parameter in input data", signifies that ISW is not in the range one to three, "invalid Mach no. range", signifies that AMS or AME is less than or equal to one, or "error in input data", signifies an error condition was encountered during the input operation. If SUPERSONIC executes normally then upon termination SUPERSONIC prints "normal completion" to your terminal or log file.

Table 3 shows the first part of the output produced when the functions of Table 1 and the data of Table 2 are used. The first character on the first line is a Fortran carriage control character the second character is ISW, in this case 1. DX and B2 are printed on the sixth line, AMS, AME and DAM are printed on the eighth line and if ISW = 1 then AOMS, AOME and DAOM are printed on the tenth line. On the twelfth line (or tenth if ISW F 1) the first character is a Fortran carriage control character. The next character is the number of grid points in the y direction, while the last character on this line is IOUT, in this case 3. SUPERSONIC then prints the Mach number and the reduced frequency for the first run, followed by the value of y at the grid points nearest the wing root and the number of points in the x- direction at that value of y, in this case 50. SUPERSONIC then prints the value of x, and the real and imaginary parts of ϕ_o (or ϕ_s^+ and ϕ_s^-) at each grid point for that y station. After all 50 points have been printed the program then prints the values of x and ϕ_o at the next value of y until the wing tip is reached. In this case as IOUT equals three the pressure coefficient at each grid point (where it is calculated) is printed in the same way as described above. Note that as the program uses central differences to calculate the x derivative of ϕ_o (or ϕ_s), the pressure coefficient is not calculated on the leading and trailing edge grid points.

To facilitate the use of this program Appendix B contains a listing of a subroutine that can be used to read the output of SUPERSONIC. When this subroutine is used the number of configurations read is transferred to the main program by the variable IC. The Mach number and reduced frequencies are stored in the arrays AM(K), AOM(K) where $1 \leq K \leq IC$.

APPENDIX C (CONT.)

The number of grid points in the spanwise direction for the pressure/potential is stored in JP(K)/JC(K) (1 ≤ K ≤ IC). The number of grid points in the streamwise direction at the Jth spanwise

point for the pressure/potential is stored in IP1(K,J)/IP2(K,J). The value of y at the Jth spanwise position for the pressure/potential is stored in Y1(K,J)/Y2(K,K). The value of x at the Jth spanwise and Ith streamwise position for the pressure/potential is stored in X1(K,I,J) / X2(K,I,J) and finally the real and imaginary parts of the pressure coefficient and potential at the Jth spanwise and Ith streamwise grid point are stored in G(K,I,J) and F(K,I,J), where I ≤ K ≤ IC. .

DISTRIBUTION

AUSTRALIA

DEPARTMENT OF DEFENCE

Defence Central

Chief Defence Scientist	(
Deputy Chief Defence Scientist	(
Superintendent, Science and Program	(1 copy
Administration	(
Controller, External Relations, Projects and	(
Analytical Studies	(

Defence Science Adviser (UK) (Doc Data sheet only)
Counsellor, Defence Science (USA) (Doc Data sheet only)
Defence Central Library
Document Exchange Centre DISB (18 copies)
Joint Intelligence Organisation
Librarian H Block, Victoria Barracks, Melbourne

Aeronautical Research Laboratories

Director
Library
Divisional File - Structures
Superintendent - Aerodynamics
R. Jones
P. Farrell
B. Emslie
M. Heller
T. Ryall
Author: J. Gear

Materials Research Laboratories

Director/Library

Defence Research Centre

Library

RAN Research Laboratory

Library

Navy Office

Navy Scientific Adviser
RAN Aircraft Maintenance and Flight Trials Unit
Directorate of Naval Aircraft Engineering
Directorate of Naval Aviation Policy
Superintendent, Aircraft Maintenance and Repair

Army Office

Royal Military College Library

Air Force Office

Air Force Scientific Adviser
Aircraft Research and Development Unit
Scientific Flight Group
Library
Technical Division Library
Director General Aircraft Engineering - Air Force
Director General Operational Requirements - Air Force
HQ Operational Command (SMAINTSO)
HQ Support Command (SLENGO)
RAAF Academy, Point Cook

Central Studies Establishment

Information Centre

Government Aircraft Factories

Manager
Library

DEPARTMENT OF AVIATION

Library
Flight Standards Division

INDUSTRY

Trans-Australia Airlines, Library
Qantas Airways Limited
Ansett Airlines of Australia, Library
Commonwealth Aircraft Corporation, Library
Hawker de Havilland Aust. Pty Ltd, Bankstown, Library
Rolls Royce of Australia Pty Ltd, Mr C.G.A. Bailey

UNIVERSITIES AND COLLEGES

Adelaide	Barr Smith Library
Flinders	Library
La Trobe	Library
Melbourne	Engineering Library
Monash	Hargrave Library

DISTRIBUTION (CONT.)

Sydney	Engineering Library
NSW	Physical Sciences Library
Queensland	Library
Tasmania	Engineering Library
Western Australia	Library
RMIT	Library

SPARES (5 copies)
TOTAL (77 copies)

Department of Defence

DOCUMENT CONTROL DATA

1. a AR No AR-004-057	1. b. Establishment No ARL-STRUC-TM-424	2. Document Date DECEMBER, 1985	3. Task No DST 82/053
4. Title A COMPUTER PROGRAM FOR CALCULATING THE STEADYAND OSCILLATORY SUPERSONIC FLOW OVER THIN WINGS		5. Security a. document UNCLASSIFIED b. title U c. abstract U	6. No Pages 24 7. No Refs 1
8. Author(s) J.A. GEAR		9. Downgrading Instructions	
10. Corporate Author and Address Aeronautical Research Laboratories P.O. Box 4331, Melbourne Vic 3001		11. Authority (as appropriate) a.Sponsor b.Security c.Downgrading d.Approval	

12. Secondary Distribution (of this document)
Approved for Public Release

Overseas enquirers outside stated limitations should be referred through ASDIS, Defence Information Services Branch, Department of Defence, Campbell Park, CANBERRA ACT 2601

13. a. This document may be ANNOUNCED in catalogues and awareness services available to ...
No limitations

13. b. Citation for other purposes (ie casual announcement) may be (select) unrestricted (or) as for 13 a.

14. Descriptors	15. COSATI Group
Supersonic flow Thin wings Computer programs	01010

16. Abstract

A computer program for the calculation of linearised steadyand oscillatory supersonic flow over thin wings has been implemented on an ELXSI. The program uses an explicit finite difference scheme on a characteristic grid to solve for the reduced potential and pressure coefficient on the wing surface.

PF 65

This page is to be used to record information which is required by the Establishment for its own use but which will not be added to the DISTIS data base unless specifically requested.

16. Abstract (Contd)		
17. Imprint Aeronautical Research Laboratories, Melbourne.		
18. Document Series and Number Structures Technical Memorandum 424	19. Cost Code 23-6925	20. Type of Report and Period Covered
21. Computer Programs Used		
22. Establishment File Ref(s)		

END

DATE

FILMED

6-86