N92-13868

Navier-Stokes Calculation of Transonic Flow Past the NTF 65-deg Delta Wing

by

Chivey C. Wu
Associate Professor of Mechanical Engineering
California State University, Los Angeles
Los Angeles, CA 90032

Models of four delta wings were built and tested in the 8 foot by 8 foot transonic wind tunnel at the National Transonic Facility, NASA Langley Research Center. The wings are identical in planform shape with a swept-back angle of 65 degrees, but bear different leading edge profiles. The models were tested under pressurized and cryogenic conditions to simulate true flight Reynold's numbers. Data on the aerodyanamic forces and pressure distributions at various locations on the wings were taken at various flight Mach numbers and angles of attack. Effects of high Reynold's number and leading edge radius on the aerodynamic characteristics of the wings are being accessed.

A preliminary analytical study was done at MIT, but it was limited to the case of low Reynold's number, thin layer laminar flow over one of the wings [1]. To thoroughly understand the turbulent, vortical flow around the wings, an effort to perform computational aerodynamic analysis is being committed. The objective of the analysis is to supplement and validate the experimental data and explain the high Reynold's number and leading edge effects. GRIDGEN, a software developed by General Dynamics, is being used to generate the grid topology for the flow field around the wings. The flow solver to be used is CFL3D, a computational fluid dynamics code developed by the Computational Aerodynamics Branch at Langley.

Based on the geometric description of the wings [2], a Fortran program called WINGSURF has been written to generate the databases defining the surface geometry of the wings. To match the true geometry of the models in the wind tunnel for realistic comparision with experimental results, databases defining the sting support for the wing models have also been created by two other Fortran programs, STINJOIN and STINREAR. Listing of the programs are attached and the geometry of a typical wing model with the sting support is shown in Fig. 1.

From the geometric databases, the computational grid for each wing both with and without the sting was created interactively on an IRIS Graphics Workstation employing the single block approach. A computational domain of 20 times the chord length of the wing in all directions was taken. Figs. 2 and 3 show the grid topology on a typical wing surface and in the entire computational domain, respectively, for free flight (no sting) condition. Figs. 4 and 5 show similar grid topology for a typical wing model attached to the sting. For ease of future comparision with experimental data, the grid on the wing surface is generated in such a way that the grid lines at 20%, 40%, 60% and 80% chord always coincide with the pressure orifices on the models.

With the single block approach, however, the grid lines extending from the sting tip to the wing apex have to collapse into a singular grid surface right on top of the centerline of the wing, resulting in a loss of a large number of computational grid points. Also, changing the leading edge profile would require a complete regeneration of the volume grid in the entire computational domain. This means excessive computing time.

For better computing efficiency, a multi-block approach is being employed to modify the grids involving the sting. The computational domain is divided into seven blocks, as shown in Fig. 6 (near field) and Fig. 7 (far field). Since the multi-block approach allows different computational dimensions among different blocks, the collapsed grid surface is easily eliminated by placing the sting in a separate block. The leading edge surface now only appears in two of the blocks, making it much more efficient to change leading edge and regenerate the grids in these two block while leaving the grids in the other blocks intact.

Surface grids for each block are in the process of being generated. Upon obtaining the surface grids, the GRIDGEN3D module in GRIDGEN can be run on the Cray to create the volume grid in each block, resulting in a complete volume grid for the entire computational domain. The flow solver can then be applied to compute flow properties in the domain. Results will be used to compare with wind tunnel data and to access effects of high Reynold's number and leading edge radius.

Due to lack of time and change of blocking strategy, the volume grids are not yet complete and calculation of flow properties has not been attempted. A research proposal is being prepared to request future support for continuation of the project at the university.

References

1. Becker, T. M., "Hybrid 3-D Euler/Navier-Stokes Calculations of Transonic Vortex Flows over the NTF Delta Wing", MIT CFDL-TR-89-7, August, 1987.

2. Luckring, J. M., "NTF Delta Wing Geometry", Model Description, NASA Langley, 1987.

```
      program wingsurf
      dimension a(4),b(4),c(4),d(4),x(22,22),z(22,22),y(22,22)
      pi = 4.*atan(1.)
      del = 65./180.*pi
      rc = 12.*tan(del)
      xle = .15*rc*cos(del)
      xte = .1*rc
      data bl/.510024/, cl/-.19819008/, dl/.025671542/
      data a/.4935262, .34897572, .20148123, 0.0/
      data b/.080048085, .29224873, .5087712, .804546399243/
      data c/-.19710469, -.28382228, -.37230603, .49317707317/
      data d/.046323876, .062270923, .078542758, .100770498643/
      do 900 k=1,4
      do 100 j=2,22
      xj = (22-j)/20.
      xn = xle*(exp(5.*xj)-1.)/(exp(5.)-1.)
      x(1,j) = xn/cos(del)
      y(1,j) = 0.
      z(1,j) = a(k)*xn**.5 + b(k)*xn + c(k)*xn**2 + d(k)*xn**3
  100 continue
      z(1,2) = bl*xte + cl*xte**2 + dl*xte**3
      x(1,1) = .15*rc
      x(2,1) = .2*rc
      x(3,1) = .4*rc
      x(4,1) = .6*rc
      x(5,1) = rc - 9.395
      x(6,1) = .8*rc
      x(7,1) = .9*rc
      do 110 i=1,7
      y(i,1) = 0.
      z(i,1) = z(1,2)
  110 continue
      do 130 i=8,22
      xi = (22-i)/15.
      xi = xte*(exp(4.*xi)-1.)/(exp(4.)-1.)
      x(i,1) = rc - xi
      y(i,1) = 0.
      z(i,1) = bl*xi + cl*xi**2 + dl*xi**3
  130 continue
      do 115 i=2,22
      do 115 j=2,22
      x(i,j) = x(i,1)
      y(i,j) = (x(i,j)-x(1,j))/tan(del)
  115 continue
      do 120 i=2,7
      do 120 j=2,22
      z(i,j) = z(1,j)
  120 continue
      do 125 i=8,22
      do 125 j=2,22
      z(i,j) = z(i,1)*z(7,j)/z(7,2)
  125 continue
      if (k.eq.1) open (7,file='bluntlo.dba')
      if (k.eq.2) open (7,file='interlo.dba')
      if (k.eq.3) open (7,file='sharplo.dba')
      if (k.eq.4) open (7,file='extralo.dba')
      write (7,*) 22,22
      write (7,*) ((x(i,j)/rc,i=1,22),j=1,22),
     1            ((y(i,j)/rc,i=1,22),j=1,22),
     2            ((-z(i,j)/rc,i=1,22),j=1,22)
      close (7)
  900 continue
      end
      program stinjoin
      dimension x(30,10),z(30,10),y(30,10)
      pi = 4.*atan(1.)
      del = 65./180.*pi
      rc = 12.*tan(del)
      xte = .1*rc
      data bl/.510024/, cl/-.19819008/, dl/.025671542/
      data as/.3094608214/, bs/.3327982281/, cs/-.04163681011/,
     1     ds/.001507294514/
      tmax = bl*xte + cl*xte**2 + dl*xte**3
      do 100 j=1,10
      x(1,j) = rc-10.022+.627
      y(1,j) = 0.
      z(1,j) = tmax
      do 120 i=2,15
      xi = (i-1)/14.
      xi = .627 + (9.395-xte)*(exp(3.*xi)-1.)/(exp(3.)-1.)
      r = as*xi**.5 + bs*xi + cs*xi**2 + ds*xi**3
      t = tmax
      ang = (pi/2.-asin(t/r))/9.*(j-1)
      x(i,j) = rc-10.022+xi
      y(i,j) = r*sin(ang)
      z(i,j) = r*cos(ang)
      if (j.eq.10) z(i,j) = tmax
  120 continue
      do 130 i=16,30
      xi = (i-15)/15.
      xi = 10.022-xte+xte*(exp(-4.*xi)-1.)/(exp(-4.)-1.)
      r = as*xi**.5 + bs*xi + cs*xi**2 + ds*xi**3
      if (xi.gt.9.5) r=1.65
      xt = 10.022-xi
      t = bl*xt + cl*xt**2 + dl*xt**3
      ang = (pi/2.-asin(t/r))/9.*(j-1)
      x(i,j) = rc - xt
      y(i,j) = r*sin(ang)
      z(i,j) = r*cos(ang)
  130 continue
  100 continue
      z(30,10) = 0.
      open (7,file='jointup.dba')
      write (7,*) 30,10
      write (7,*) ((x(i,j)/rc,i=1,30),j=1,10),
     1            ((y(i,j)/rc,i=1,30),j=1,10),
     2            ((z(i,j)/rc,i=1,30),j=1,10)
      close(7)
      end
      program sting
      dimension x(10),y(10,10),z(10,10)
      pi = 4.*atan(1.)
      open (7,file='stingup2.dba')
      del = 65./180.*pi
      rc = 12.*tan(del)
      x(1) = rc
      x(2) = rc + 4.5
      do 100 i=1,2
      do 100 j=1,10
      y(i,j) = 1.65*sin(pi/18.*(j-1))
      z(i,j) = 1.65*cos(pi/18.*(j-1))
  100 continue
      do 200 i=3,6
      x(i) = x(i-1)+.5
      r = 52.57-(50.92**2-(.5*(i-2))**2)**.5
      do 200 j=1,10
      y(i,j) = r*sin(pi/18.*(j-1))
      z(i,j) = r*cos(pi/18.*(j-1))
  200 continue
      x(7) = rc + 17.59
      do 300 j=1,10
      y(7,j) = 2.125*sin(pi/18.*(j-1))
      z(7,j) = 2.125*cos(pi/18.*(j-1))
  300 continue
      x(8) = rc + 19.5
      do 400 j=1,10
      y(8,j) = 2.875*sin(pi/18.*(j-1))
      z(8,j) = 2.875*cos(pi/18.*(j-1))
  400 continue
      x(9) = rc + 25.7
      r = 2.875 + 6.2*tan(2.*pi/180.)
      do 450 j=1,10
      y(9,j) = r*sin(pi/18.*(j-1))
      z(9,j) = r*cos(pi/18.*(j-1))
  450 continue
      x(10) = 20.*rc
      r = 2.875 + (19.*rc - 19.5)*tan(2.*pi/180.)
      do 500 j=1,10
      y(10,j) = r*sin(pi/18.*(j-1))
      z(10,j) = r*cos(pi/18.*(j-1))
  500 continue
      do 600 i=1,10
      z(i,10) = 0.
  600 continue
      write (7,*) 10,10
      write (7,*) ((x(i)/rc,i=1,10),j=1,10),




     1 ((y(i,j)/rc,i=1,10),j=1,10),((z(i,j)/rc,i=1,10),j=1,10)
      close (7)
      end
```

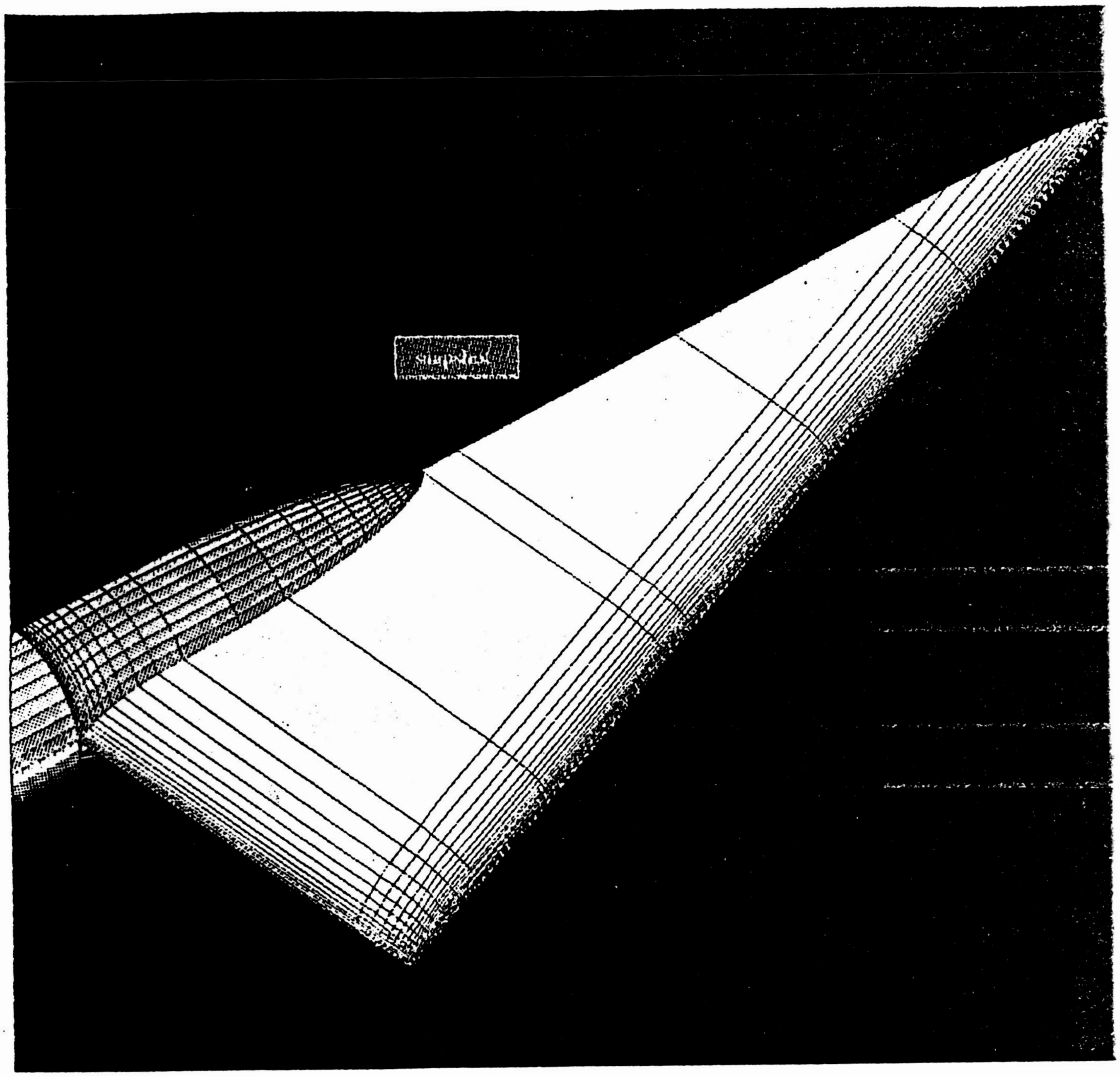

Fig. 1 Wing and Sting Geometry (Right Half)

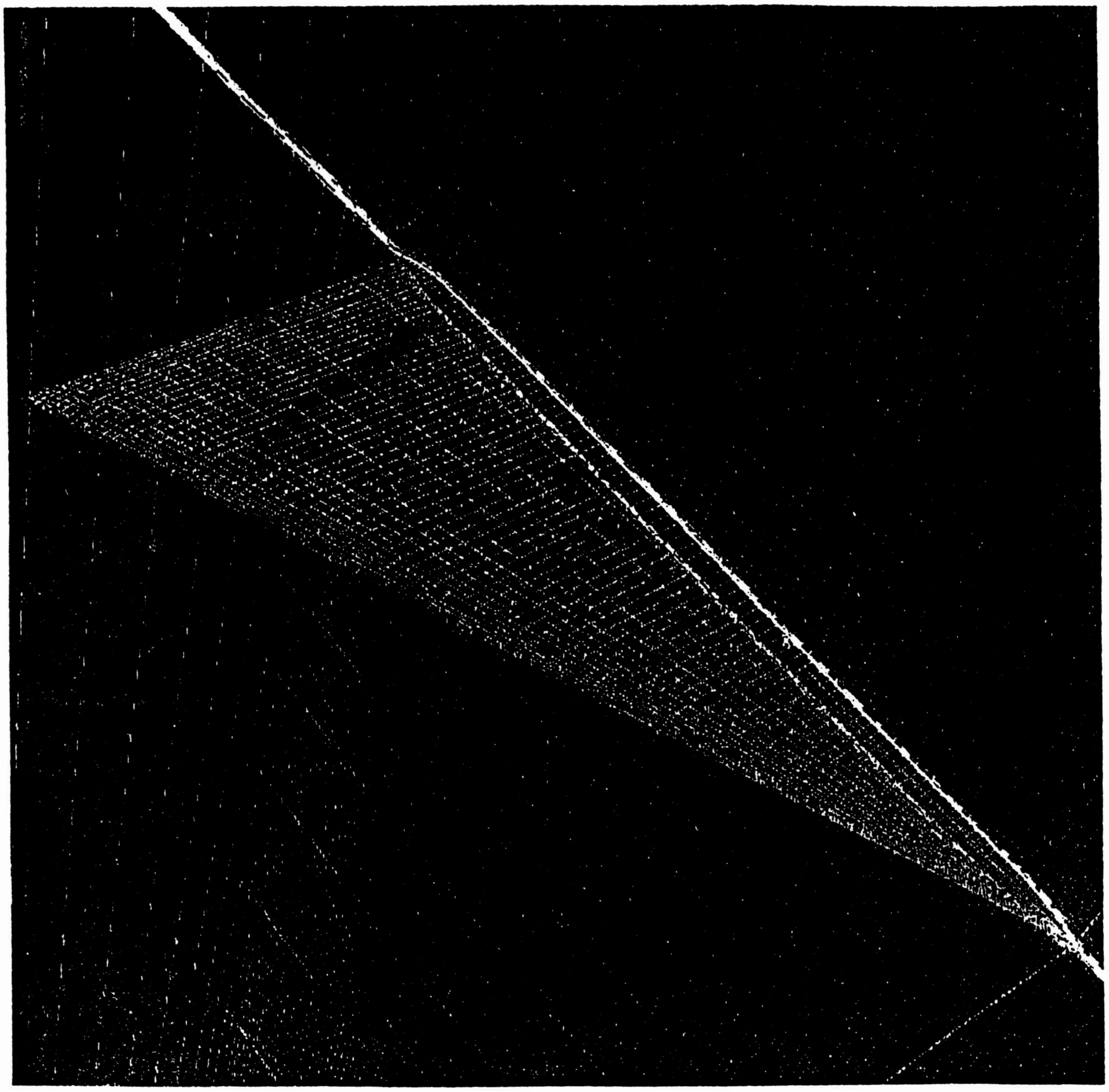

Fig. 2 Grid on Wing Surface

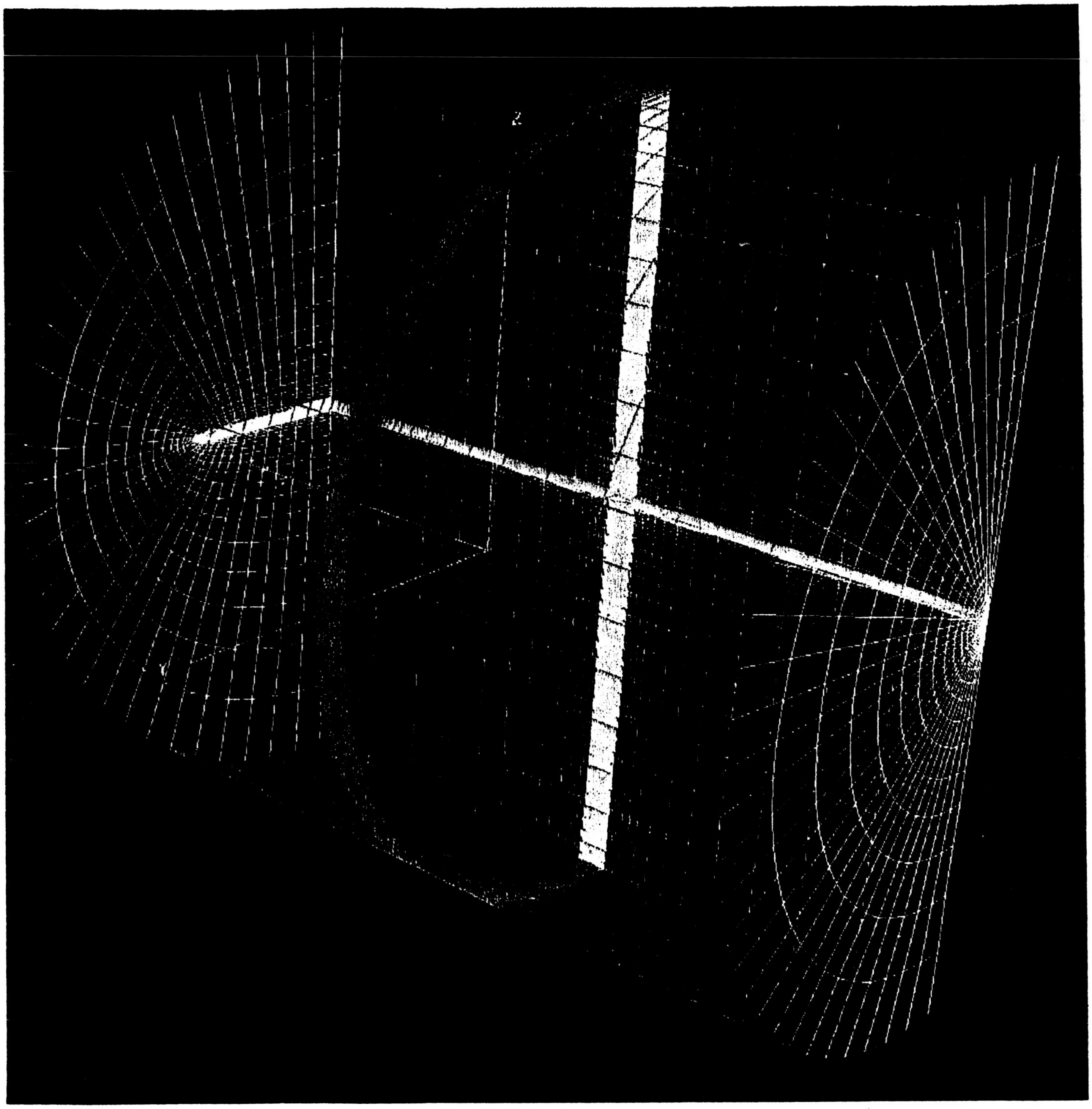

Fig. 3 Grid for Entire Computational Domain Around Wing Alone

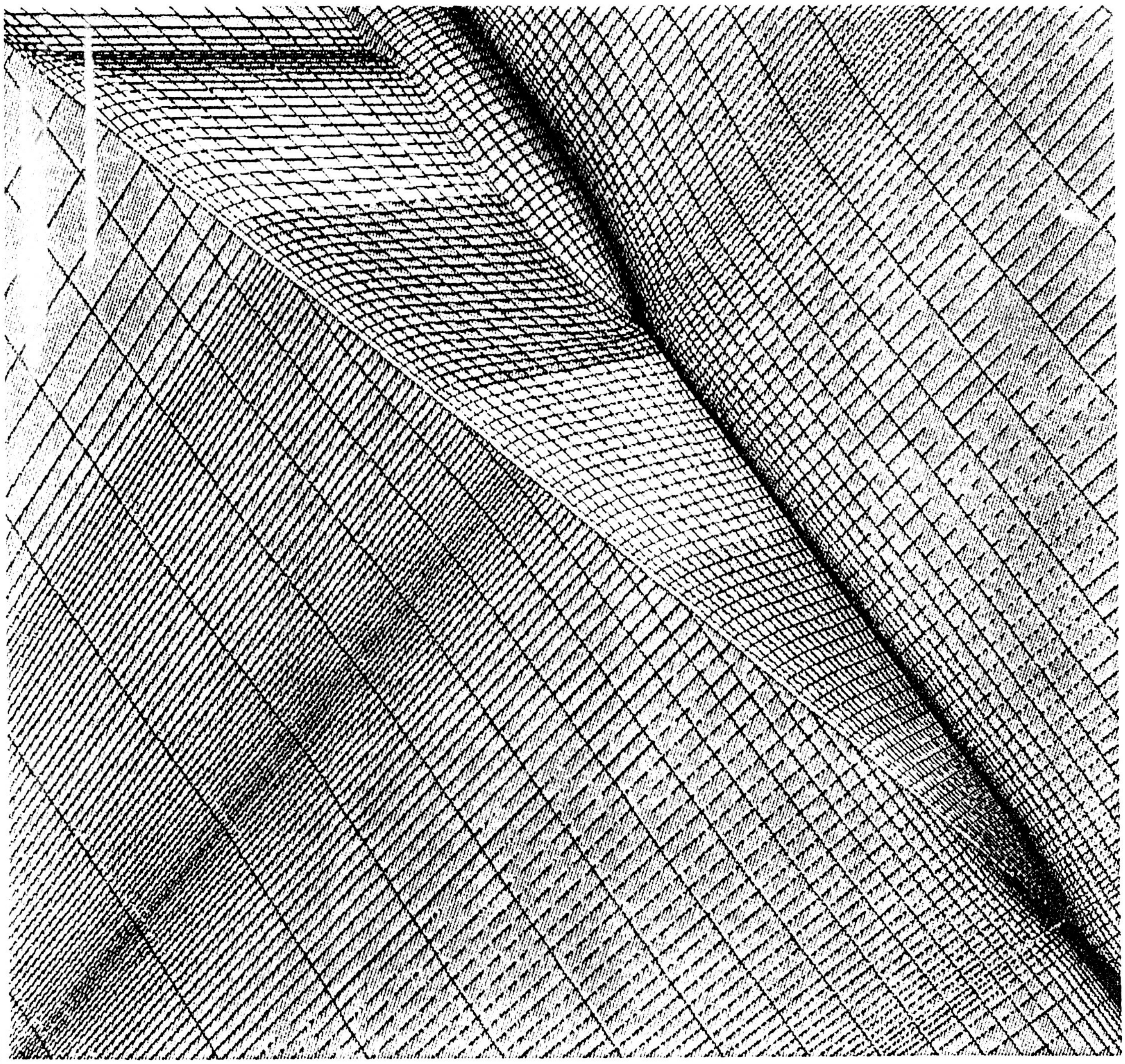

Fig. 4 Grid on Wing and Sting Surfaces

ORIGINAL PAGE IS
OF POOR QUALITY

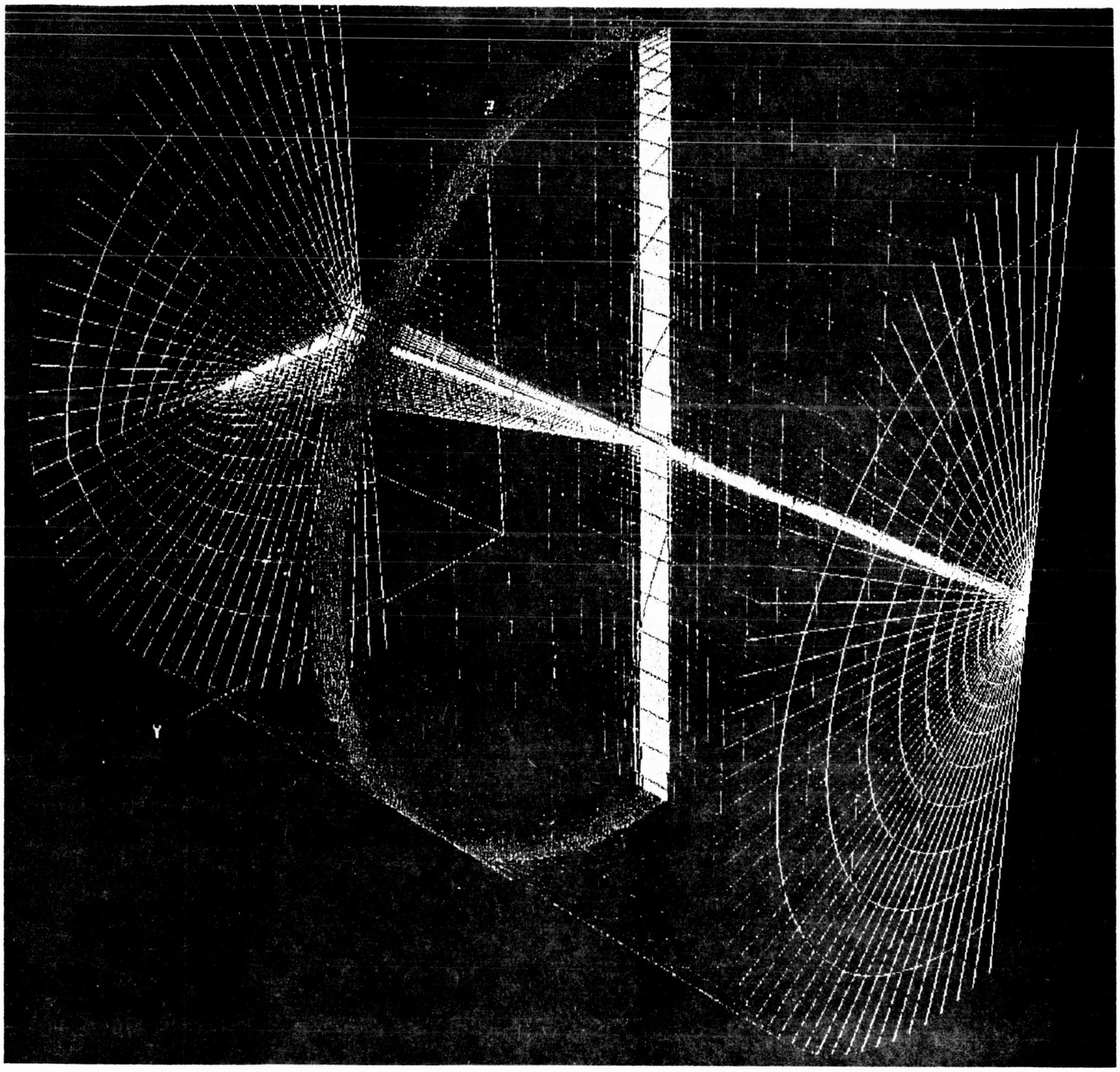

Fig. 5 Grid for Entire Computational Domain around Wing with Sting

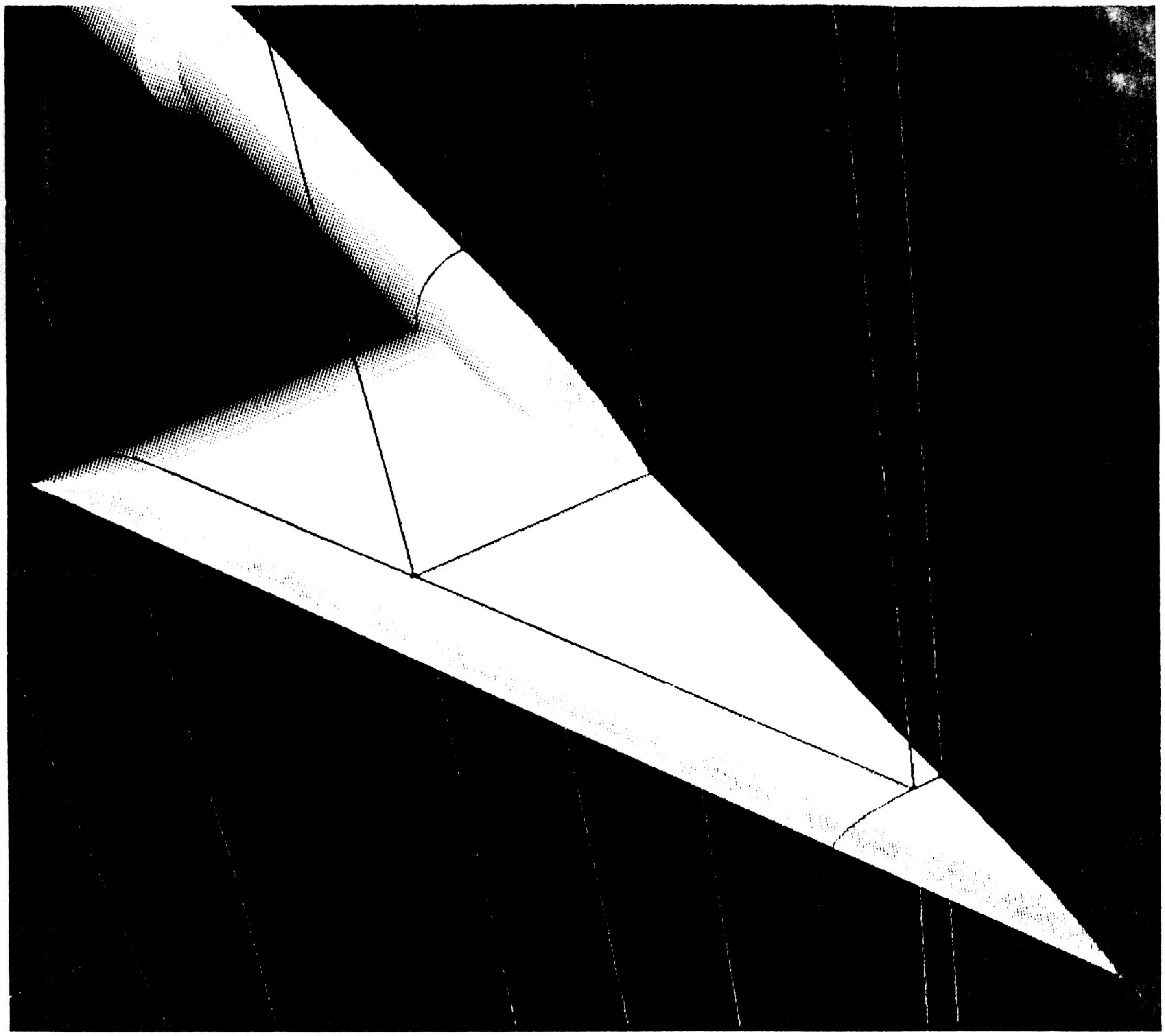

Fig. 6 Blocking Strategy for Wing Model with Sting

ORIGINAL PAGE IS
OF POOR QUALITY

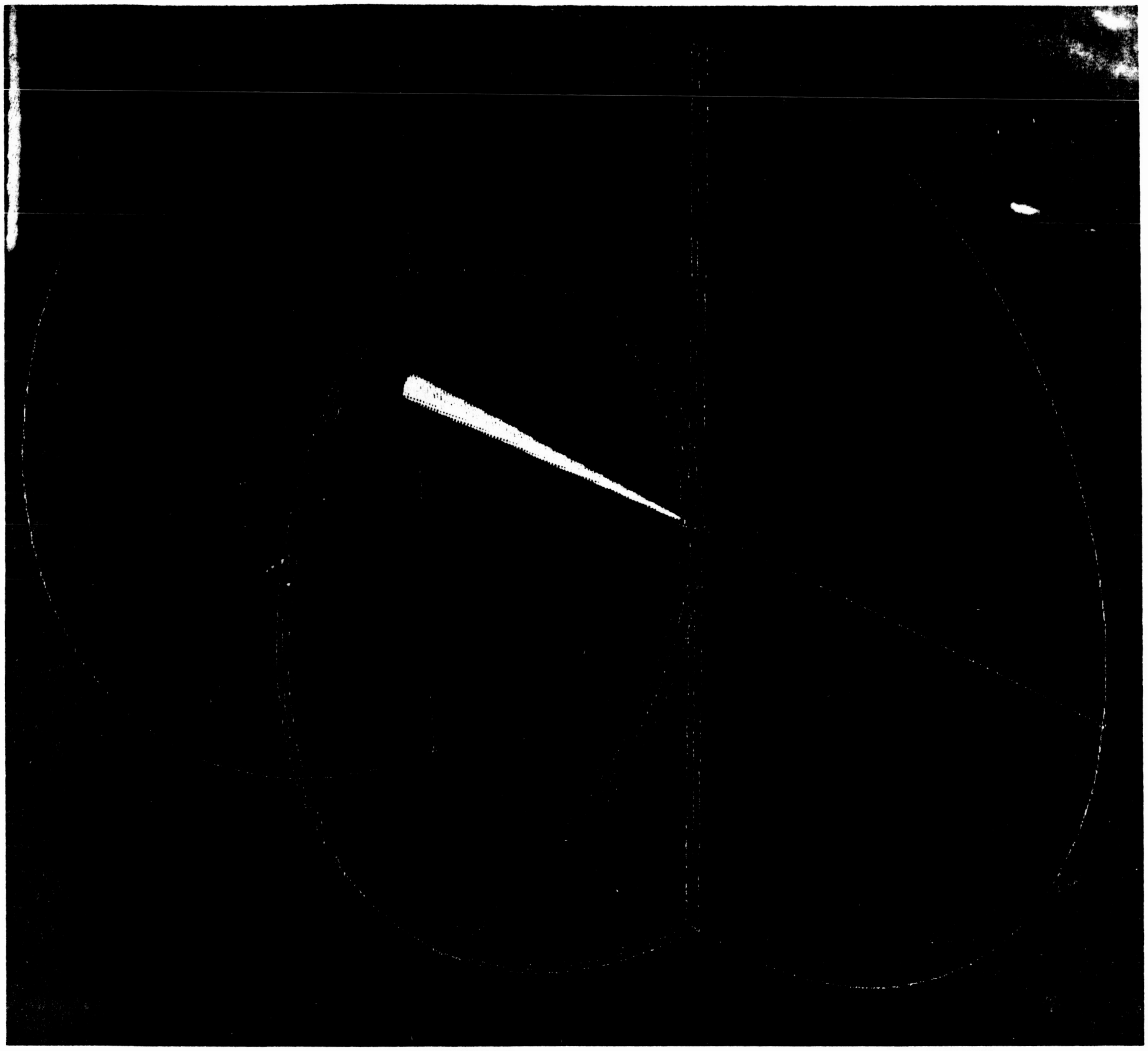

Fig. 7   Seven-Block System for Entire Computational Domain