

AMES GRANT
IN-64-CR

224938

P-126

FINAL REPORT

In response to the requirements of

NCC 2-515

EFFICIENT GRID GENERATION

Principal Investigator: Ryoichi Seki

Period of Study: August 1, 1987 - July 31, 1989

(NASA-CR-185034) EFFICIENT GRID GENERATION
Final Report, 1 Aug. 1987 - 31 Jul. 1989
(California State Univ.) 126 p CSCL 12A

W89-27434

Unclas
G3/64 0224988

CSUN

Northridge, CA 91330

Contents

Preface	iii
List of Figures	vi
Abstract	viii
1 Introduction	1
2 Classification of Partial Differential Equations	9
3 Transformation Metrics	11
4 Elliptic Grid Generation	13
5 Hyperbolic Grid Generation	23
5.1 Numerical technique	27
6 Combining Elliptic and Hyperbolic Equations	36
6.1 Solution Procedure	37
6.2 Testing Procedure	39
6.3 The Unmodified Algorithm	40
6.4 Floating the Boundary	41
6.5 Using Compressed-Cell Initial Grids	42
6.6 Curtailing Outer Boundary Movement within a Cavity	43
7 Grid Breakdown	94
8 Derivation of the Navier-Stokes Equations	99
9 Results and Conclusions	107

10 Notation and Definitions

113

Bibliography

116

List of Figures

Bibliography	118
1 Portion of boundary points and solution points in a 2-dimensional rectangular coordinate system with Δx and Δy constant. Boundary conforms to a coordinate line.	6
2 Variance in Δx and Δy to increase local solution accuracy and resolution with resultant additional number of solution points. . .	7
3 Consequent change in Δx and Δy near boundary that doesn't conform to a coordinate line.	8
4-7 The elliptic generator at work	19-22
8-14 The hyperbolic generator at work on bodies of various concavities	29-36
15-19 The combined elliptic-hyperbolic generator with a fixed outer boundary	46-50
20-33 The combined elliptic-hyperbolic generator with a floated outer boundary	51-64
34-45 The combined elliptic-hyperbolic generator with a floated outer boundary, using cell-volume growth.	65-76
46 The combined elliptic-hyperbolic generator with a floated outer boundary, using cell-volume growth. Failure of technique.	77
47-62 The combined elliptic-hyperbolic generator with a floated outer boundary, using cell-volume growth. Forming a cut.	78-93
63 Portion of boundary and first row of grid cells next to boundary. Here $k=2$	95
64 Grid breakdown at a strong convexity.	98

ORIGINAL PAGE IS
OF POOR QUALITY

65 Fixed control volume showing area vector \vec{A} and the three surface stresses at one of the two surfaces normal to the x axis. 69

66-69 Generating a grid about two bodies 110-112

70-73 The relation of the parameter cam to concavity 117

ABSTRACT
EFFICIENT GRID GENERATION

by

Bill Gutierrez

Master of Science in Physics

Because the governing equations in fluid dynamics contain partial differentials and are too difficult in most cases to solve analytically, these differentials are generally replaced by finite difference terms. These terms contain terms in the solution at nearby states. This procedure discretizes the field into a finite number of states. These states, when plotted, form a grid, or mesh, of points. It is at these states, or field points, that the solution is found.

The optimum choice of states, the x,y,z coordinate values, minimizes error and computational time. But the process of finding these states is made more difficult by complex boundaries, and by the need to control step size differences between the states, that is, the need to control the spacing of field points.

One solution technique uses a different set of state variables, which define a different coordinate system, to generate the grid more easily. A new method, developed by Dr. Joseph Steger, combines elliptic and hyperbolic partial differential equations into a mapping function between the the physical and computational coordinate systems. This system of equations offers more control than either equa-

tion provides alone. The author has modified Steger's algorithm in order to allow bodies with stronger concavities to be used, offering the possibility of generating a single grid about multiple bodies. Work has also been done on identifying areas where grid breakdown occurs. This work was supported by NASA under the NASA Minority Graduate Researcher's program.

1 Introduction

The governing equations in fluid dynamics, as in many other fields, contain terms in partial differentials. In the numerical solution of these equations the differentials are generally replaced by finite difference terms derived from Taylor series expansions. The solution is then found from given auxiliary data. For instance, if u is a function of n variables

$$u = u(x_1, x_2, \dots, x_n) \quad (1)$$

and the equation relating u to these variables is a partial differential equation, each m 'th partial derivative of u can be expressed as a sum of terms in u , and an error term

$$\frac{\partial^m u}{\partial x_i^m} = \sum_{j=1}^l a_j u(x_1, \dots, x_i + b_j \Delta x_i, \dots, x_n) + O(\Delta x)^p. \quad (2)$$

where a_j and b_j are constants and $O(\Delta x)^p$ is the error term. It is important to note that, similar to the exact differential on the left, all other x variables are held constant while x_i varies to produce the different u values. $O(\Delta x)^p$, the order of error, is a measure of difference between the exact differential and the finite difference term. The smaller Δx is, the smaller the error. Because Δx is generally less than one, higher values of p will also decrease this error.

A more specific example of the foregoing is a finite difference equation for the first partial derivative of u at $x_i = x'_i$. Dropping notation of x variables held constant, equation (2) becomes in this case

$$\frac{\partial u}{\partial x_i}(x'_i) \equiv u_{x_i}(x'_i) = \frac{u(x'_i + \Delta x_i) - u(x'_i - \Delta x_i)}{2\Delta x_i} + O(\Delta x_i)^2 \quad (3)$$

ORIGINAL PAGE IS
OF POOR QUALITY

here

$$\begin{aligned} u_1 = u_2 = \frac{1}{2\Delta x_i} \quad o_1 = 1 \quad o_2 = -1 \\ m = 1, \quad p = 2. \end{aligned} \quad (4)$$

Using equation (2), the solution of u at any $x_i = x'_i$ requires the solution of u at neighboring values of x_i . The solution for a given set of (x_1, x_2, \dots, x_n) values is thus dependent on the solution in some neighborhood of these x values. But therefore these neighboring solutions are dependent on their neighbors' solutions. For each x_i this dependency is generally continuous between one or two x_i values at which u is definitely known: the auxiliary data. Therefore, the solution process starts near values of the x variables for which u is known. One technique, a solution by "lines", is outlined as follows. One of the x variables, say x_1 is stepped through a set of values. At each value a finite difference equation equal, to some order of accuracy, to the original partial differential equation, is set up. The resultant set of equations is then solved simultaneously. x_2 may then be varied a small amount, Δx_2 , and the process of running through x_1 values is repeated to solve another set of equations. When x_2 reaches its maximum x_3 can then be varied a small amount, Δx_3 , and the previous processes repeated.

The fact that u changes only by changing one x variable makes it convenient to view the problem within an n -dimensional coordinate system where each x_i is represented by an axis. Within this system the set of x values in equation (2) lies on a coordinate line¹ of the system. In a two-dimensional system, where $x = x_1$ and $y = x_2$, u_x is calculated from values of x and u along a y coordinate line; u_y

¹a coordinate line is described by allowing x_i to vary while keeping all other coordinate variables constant. a coordinate surface is described by keeping x_i constant while varying the other variables.

is found from values of y (and u) along an x coordinate line. figure 1 shows the points used in these calculations and their relations to one another. It should be obvious from the figure that the field is now broken up into a set of points. It is at these points that the solution is obtained.

To facilitate numerical computations it is advantageous to keep Δx and Δy constant, for the following reasons.

- The derivation of the above finite difference equation, (3), is based on a constant Δx , allowing for a higher order of accuracy and simplicity of form than would otherwise be possible.
- A constant Δ leads to the symmetrical array of points in figure 1. These points can then be indexed as an array, with points in the x direction being indexed by j , and points in the y direction being indexed by k . When one boundary value of x , is indexed as $x(1)$, and the other boundary as $x(jmax)$, the following simple relations can then hold

$$x(j) = x(1) + (j - 1) \times \Delta x \quad (5)$$

$$x(jmax) = x(1) + (jmax - 1) \times \Delta x \quad (6)$$

Where $j = 1, 2, \dots, jmax$. A similar situation exists for y where the index runs from one to some maximum, $kmax$. One can then say that

$$\begin{aligned} x &= x(j) \equiv x_j \\ y &= y(k) \equiv y_k \\ u &= u(j, k) \equiv u_{j,k}. \end{aligned} \quad (7)$$

Further, since to each u there associates an x and a y , they can be indexed as

$$\begin{aligned} \varphi(j, k) &\equiv \varphi_{j,k} & j = 1, 2, \dots, j_{max} \\ y(j, k) &\equiv y_{j,k} & k = 1, 2, \dots, k_{max} \end{aligned} \quad (6)$$

Any point in the field can then be identified by its j,k indices.

There are three major reasons why the Δ 's are not usually constant.

- To reduce errors due to $O(\Delta x)^n$ and increase resolution of u, Δx can be made smaller.
- To increase efficiency Δx can be made larger in areas where u is changing more slowly.
- Varying y may vary the boundary values of x, so that equation (6) holds only if Δx is changed.

Figure 2 shows the result of varying Δx within the field, in the area labeled "increased resolution". Because of the constancy requirement for a finite difference expression, this new point requires additional points along the x and y coordinate lines intersecting this new point. u must be solved for at these new points, increasing computational time.

When the boundary does not lie on a coordinate line, as in figure 3, unless the boundary is a linear function of y and x, it is not possible to keep Δx 's or Δy 's constant in the field. Although interpolation can be employed or different finite difference terms used, accuracy tends to decrease, while computational time tends to increase.

One solution is to define a new coordinate system, a computational system, in which

- All boundaries coincide with coordinate lines (or surfaces in 3-d).

ORIGINAL PAGE IS
OF POOR QUALITY

- A minimum of coordinate lines are used to describe the boundary.
- If the new coordinate system's variables are denoted by ξ, η then $u = u(\xi, \eta)$,
and
- the finite difference approximations contain terms in ξ and η , and $\Delta\xi$ and $\Delta\eta$.
- $\Delta\xi$ and $\Delta\eta$, the point spacing on the boundary and interior, are constant.
- The mapping between coordinate systems must be one-to-one so that

$$\xi = \xi(x, y), \quad (9)$$

$$\eta = \eta(x, y), \quad (10)$$

$$x = x(\xi, \eta), \quad (11)$$

$$y = y(\xi, \eta). \quad (12)$$

- To ensure nonsingularity between corresponding points the Jacobian determinate (J) must be nonzero.
- When ξ, η are the coordinates of a cartesian system the grid is a rectangular mesh of points.

The requirements lend themselves rather well to using partial differential equations as the mapping functions. Existence of 2nd order partials also ensures smoothness and continuity. Although it is possible to use conformal mapping techniques to map between two-dimensional coordinate systems, conformal mapping does not extend to three-dimensional systems, whereas partial differential equations can map between these systems.

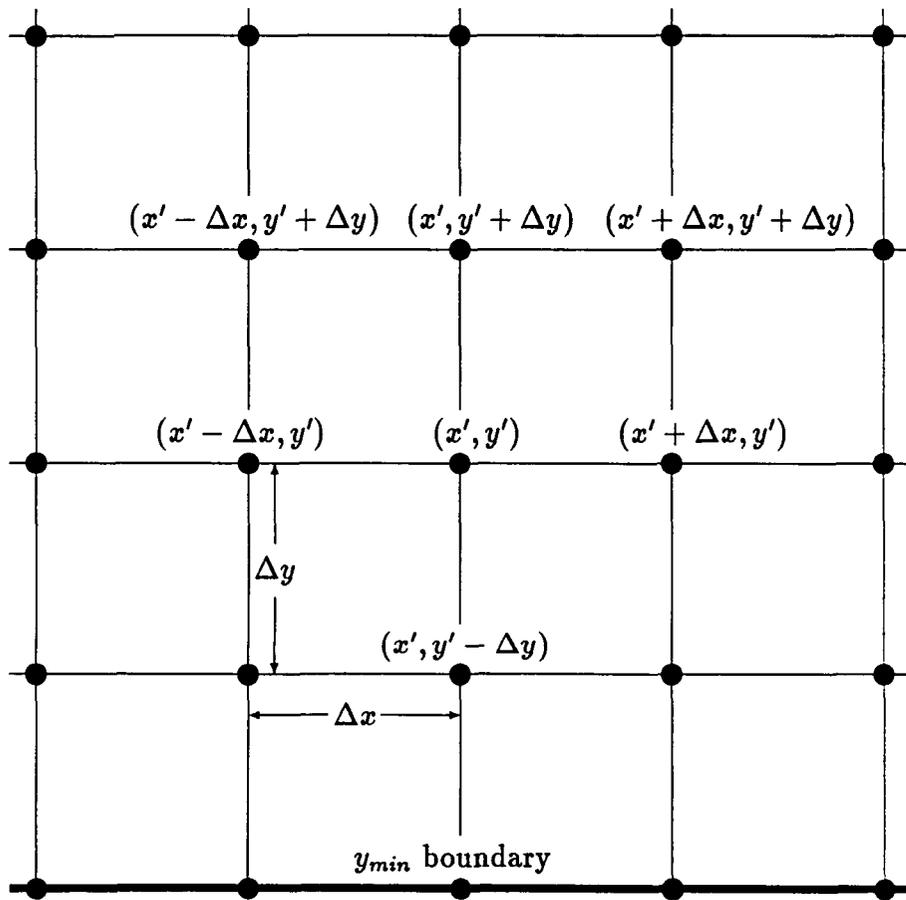


Figure 1: Portion of boundary points and solution points in a 2-dimensional rectangular coordinate system with Δx and Δy constant. Boundary conforms to a coordinate line.

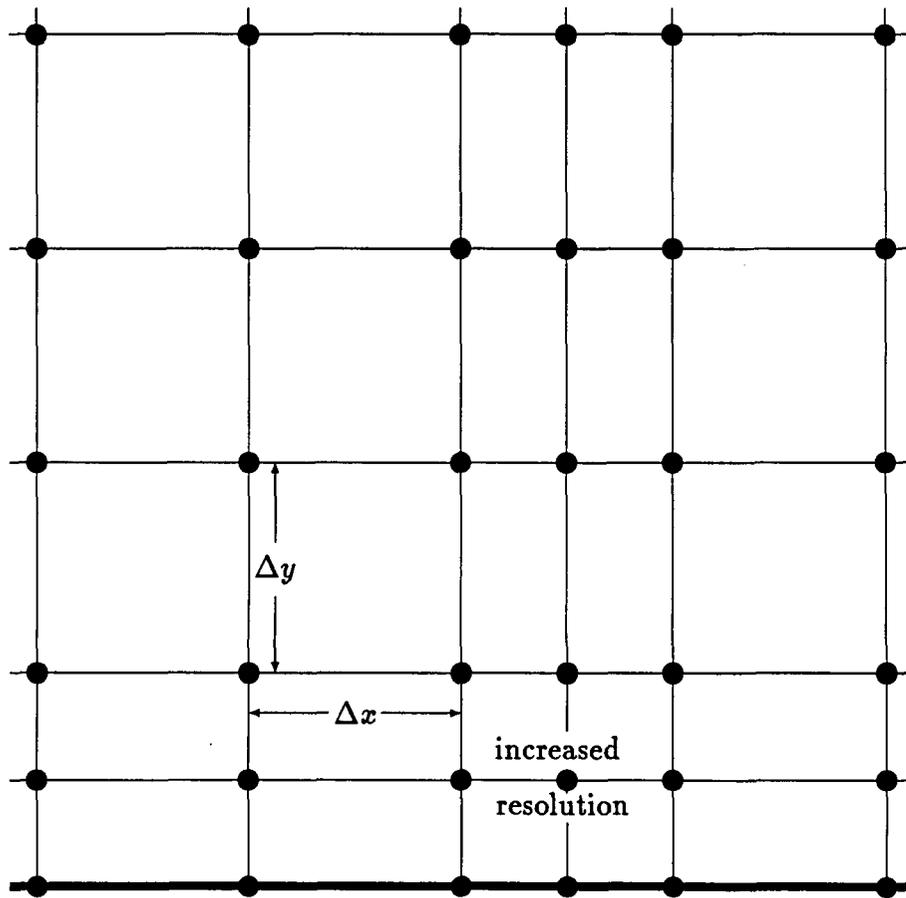


Figure 2: Variance in Δx and Δy to increase local solution accuracy and resolution with resultant additional number of solution points.

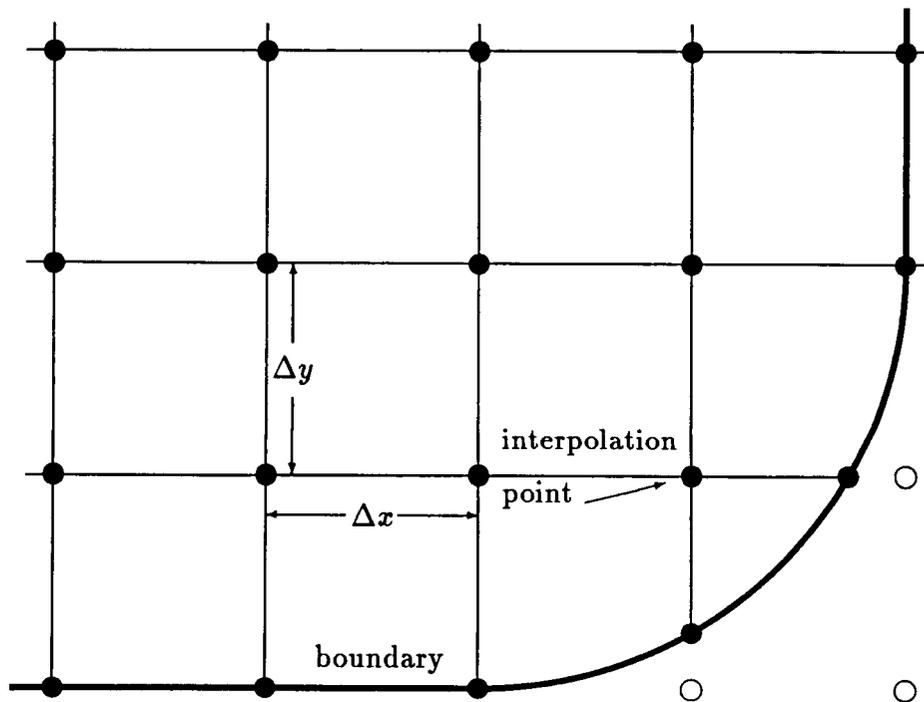


Figure 1: Consequent change in Δx and Δy near boundary that doesn't conform to a coordinate line.

2 Classification of Partial Differential Equations

The general form of a second-order partial differential equation in two variables is

$$au_{xx} + bu_{xy} + cu_{yy} + du_x + eu_y + fu = g \quad (13)$$

where $u = u(x, y)$ and $a, b, c, d, e, f,$ and g are functions of x and y for a linear partial differential equation. If these coefficients contain first partials of u (u_x or u_y) the equation is termed quasi-linear.

By an appropriate transformation from x, y to ξ, η (13) transforms into one of three canonical equations, depending on the value of $b^2 - 4ac$.

For $b^2 - 4ac > 0$ (13) is termed hyperbolic and the canonical equation is

$$u_{\xi\xi} - u_{\eta\eta} + \dots = 0 \quad (14)$$

For $b^2 - 4ac = 0$ (13) is termed parabolic and the canonical equation is

$$u_{\xi\xi} + \dots = 0 \quad (15)$$

For $b^2 - 4ac < 0$ (13) is termed elliptic and the canonical equation is

$$u_{\xi\xi} + u_{\eta\eta} + \dots = 0 \quad (16)$$

Where the \dots refers to terms in $x, y, u, u_\xi,$ and u_η . Each type of equation defines what auxiliary data in u , or one of its derivatives, is required for the problem to be "well-posed" for solution. This also directly influences what solution procedure may be used. If both x and y are spatial functions the auxiliary data is a boundary condition. If x or y represents time the auxiliary data represents initial conditions and possibly final conditions. Both hyperbolic and parabolic equations march out from the auxiliary data to an unspecified boundary (or final) condition of u . For hyperbolic equations, however, the solution at a point within the domain of

ORIGINAL PAGE IS
OF POOR QUALITY

integration is a function only of a certain portion of the given data. For parabolic equations, the solution at a point is dependent on the entire auxiliary data. The elliptic equation requires that the auxiliary data in u surround the domain of integration. That is, u for the entire boundary of the domain of integration must be given. The solution is then found for u within the domain.

3 Transformation Metrics

Given a general curvilinear coordinate system denoted by coordinates ξ and η , with a one-to-one point mapping between it and a cartesian system denoted by x and y , that is

$$\xi = \xi(x, y) \quad (17)$$

$$\eta = \eta(x, y) \quad (18)$$

as well as

$$x = x(\xi, \eta) \quad (19)$$

$$y = y(\xi, \eta), \quad (20)$$

then by the chain rule,

$$d\xi = \xi_x dx + \xi_y dy \quad (21)$$

$$dx = x_\xi d\xi + x_\eta d\eta \quad (22)$$

$$dy = y_\xi d\xi + y_\eta d\eta. \quad (23)$$

Substituting the values for dx and dy of equations (22) and (23) into equation (21) produces

$$d\xi = \xi_x(x_\xi d\xi + x_\eta d\eta) + \xi_y(y_\xi d\xi + y_\eta d\eta). \quad (24)$$

Rearranging,

Since ξ is not a function of η the terms within the second set of parentheses must equal zero and so the terms in the first set of parentheses must equal 1. The resulting set of equations

$$\xi_x x_\xi + \xi_y y_\xi = 1$$

$$\xi_x x_\eta + \xi_y y_\eta = 0$$

produces

$$\xi_x = \frac{y_\eta}{|J|} \quad (26)$$

$$\xi_y = \frac{-x_\eta}{|J|} \quad (27)$$

where

$$|J| = x_\xi y_\eta - x_\eta y_\xi$$

is called the Jacobian determinant. The same procedure can be applied to η and the results are

$$\eta_y = \frac{x_\xi}{|J|} \quad (28)$$

$$\eta_x = \frac{-y_\xi}{|J|} \quad (29)$$

4 Elliptic Grid Generation

The simplest, and most commonly used partial differential equation for grid generation is Laplace's equation [3],[2]

$$\nabla^2 u \equiv u_{xx} + u_{yy} = 0 \quad (30)$$

a homogeneous elliptic equation.

Replacing u in turn by the transformation coordinate variables ξ and η produces a system of equations

$$\nabla^2 \vec{C} = \begin{pmatrix} \xi_{xx} + \xi_{yy} \\ \eta_{xx} + \eta_{yy} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad (31)$$

where

$$\vec{C} \equiv \begin{pmatrix} \xi \\ \eta \end{pmatrix}.$$

When ξ, η are considered as the independent variables, equation (31) can be written, using the transformation metrics, as

$$\alpha I \vec{R}_{\xi\xi} - 2\beta I \vec{R}_{\xi\eta} + \gamma I \vec{R}_{\eta\eta} = 0, \quad (32)$$

where

$$\begin{aligned} \vec{R} &\equiv \begin{pmatrix} x \\ y \end{pmatrix}, \\ \alpha &\equiv x_\eta^2 + y_\eta^2, \\ \beta &\equiv x_\xi x_\eta + y_\xi y_\eta, \\ \gamma &\equiv x_\xi^2 + y_\xi^2, \\ I &\equiv \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \end{aligned}$$

Using central difference forms for the first and second derivatives (see appendix), noting that $\Delta\xi = \Delta\eta = 1$, and using indexing notation, one finite difference form of equation (32) is

$$a_{j,k}I\frac{\delta_\xi^2\vec{R}}{(2\Delta\xi)^2} - 2b_{j,k}I\frac{\bar{\delta}_\xi\bar{\delta}_\eta\vec{R}}{(2\Delta\xi)(2\Delta\eta)} + c_{j,k}I\frac{\delta_\eta^2\vec{R}}{(2\Delta\eta)^2} = 0, \quad (33)$$

where

$$a_{j,k} = \left(\frac{\bar{\delta}_\eta x_{j,k}}{2\Delta\eta}\right)^2 + \left(\frac{\bar{\delta}_\eta y_{j,k}}{2\Delta\eta}\right)^2 \quad (34)$$

$$b_{j,k} = \left(\frac{\bar{\delta}_\xi x_{j,k}}{2\Delta\xi}\right)\left(\frac{\bar{\delta}_\eta x_{j,k}}{2\Delta\eta}\right) + \left(\frac{\bar{\delta}_\xi y_{j,k}}{2\Delta\xi}\right)\left(\frac{\bar{\delta}_\eta y_{j,k}}{2\Delta\eta}\right) \quad (35)$$

$$c_{j,k} = \left(\frac{\bar{\delta}_\xi x_{j,k}}{2\Delta\xi}\right)^2 + \left(\frac{\bar{\delta}_\xi y_{j,k}}{2\Delta\xi}\right)^2 \quad (36)$$

After clearing the denominators and expanding, equation (33) becomes

$$\begin{aligned} & a_{j,k}I(\vec{R}_{j+1,k} - 2\vec{R}_{j,k} + \vec{R}_{j-1,k}) \\ & - 2b_{j,k}I(-\vec{R}_{j+1,k+1} + \vec{R}_{j-1,k+1}) \\ & - 2b_{j,k}I(\vec{R}_{j+1,k-1} + \vec{R}_{j-1,k-1}) \\ & + c_{j,k}I(\vec{R}_{j,k+1} - 2\vec{R}_{j,k} + \vec{R}_{j,k-1}) = 0. \end{aligned} \quad (37)$$

Gathering all \vec{R} terms in k to the left side, this now becomes

$$A_{j,k}I(\vec{R}_{j-1,k}) + B_{j,k}I(\vec{R}_{j,k}) + C_{j,k}I(\vec{R}_{j+1,k}) = G_{j,k} \quad (38)$$

where

$$\begin{aligned}
A_{j,k} &= C_{j,k} = a_{j,k} \\
B_{j,k} &= -2(a_{j,k} + c_{j,k}) \\
G_{j,k} &= 2b_{j,k}I(\vec{R}_{j+1,k+1} - \vec{R}_{j-1,k+1} - \vec{R}_{j+1,k-1} + \vec{R}_{j-1,k-1}) \\
&\quad - c_{j,k}I(\vec{R}_{j,k+1} + \vec{R}_{j,k-1})
\end{aligned}$$

This system can be solved by the "line" solution technique mentioned earlier: by keeping k constant (starting at k=1) an equation for each $\vec{R}_{j,1}$ (from j=1 to j=jmax) is generated. The resultant set of equations is then solved simultaneously for \vec{R} . The index variable k is then incremented and the process, of generating a set of equations for all j, is repeated. The overall process of incrementing k and solving a set of equations continues until a final k=kmax value is reached. Because the values of all x and y are initially unknown, except at boundary points, initial guesses for them must be made. These initial x and y values are then used to generate the values for A,B, and G. As new values of $\vec{R}_{j,k}$ ($x_{j,k}$ and $y_{j,k}$) are found, they are used to solve the succeeding set of equations for the next set of \vec{R} values.

equation (38) is a system of 2 equations, one for each unknown coordinate variable x, and y. Substituting x for each instance of the vector \vec{R} the equation becomes

$$\begin{aligned}
A_{j,k}x_{j-1,k} + B_{j,k}x_{j,k} + C_{j,k}x_{j+1,k} = \\
2b_{j,k}(x_{j+1,k+1} - x_{j-1,k+1} - x_{j+1,k-1} + x_{j-1,k-1}) \\
- c_{j,k}(x_{j,k+1} + x_{j,k-1}).
\end{aligned} \tag{39}$$

The complete set of equations, for constant k, from j=1 to jmax, can be put in matrix form as

$$\begin{pmatrix} B_{2,k} & C_{3,k} & 0 \dots & & & 0 \\ A_{2,k} & B_{3,k} & C_{4,k} & 0 & \dots & 0 \\ 0 \dots & & \ddots & & & \vdots \\ 0 & \dots 0 & A_{j-1,k} & B_{j,k} & C_{j+1,k} & 0 \dots 0 \\ \vdots & & & & \ddots & \vdots \\ 0 & \dots & 0 & A_{jmax-2,k} & B_{jmax-1,k} & \end{pmatrix} \begin{pmatrix} x_{2,k} \\ x_{3,k} \\ \vdots \\ x_{j,k} \\ \vdots \\ x_{jmax-1,k} \end{pmatrix} = \begin{pmatrix} Gx_{2,k} \\ Gx_{3,k} \\ \vdots \\ Gx_{j,k} \\ \vdots \\ Gx_{jmax-1,k} \end{pmatrix} \quad (40)$$

where each $Gx_{j,k}$ equals the right-hand-side of equation (39). A similar set of equations are solved for y at each k level.

Having run through all j and k values an improved set of x,y values is obtained. This improved set can then be used to obtain an even better set of values by repeating the procedure just described. This iterative process stops when at some iteration the change in x and y from a previous iteration is smaller than some given amount.

The number of iterations can be reduced significantly by using a technique known as successive line over-relaxation, [6] which is essentially a forecasting technique. The latest values of x and y are compared with their previous values and some fraction of their difference is added to the latest value, to predict a better value for the x and y's. For example, after solving the set of equations along a line using Thomas' algorithm, and obtaining new values for x(j,k) at each k level, where n refers to the iteration number, the following modification can be applied.

$$x_{j,k}^n = x_{j,k}^{n-1} + \omega(x_{j,k}^n - x_{j,k}^{n-1})$$

where n references the last iteration of x and n-1 references the previous value. ω controls the amount of change made. When ω lies between one and two the technique is known as over-relaxation. When ω is between zero and one the technique is known as under-relaxation. Under-relaxation is useful when values

of the dependent variable oscillate for a given set of values of the independent variables, that is, oscillation with respect to iteration

The main advantage of Laplace's equation in grid generation is that it serves to smooth grid lines. This is because it contains within it the maximum principle: there are no maximums or minimums of u within the domain of integration. That is, u attains its highest or lowest values on the boundary of the domain, $u(x^0, y^0)$ is an average of u in the neighborhood of x^0 and y^0 . This serves to reduce disparities in u for neighboring points. This can be shown by noting first that, within the domain of integration, the dependent variable cannot have a maximum or minimum. For there to be an extremum the first partial derivatives of the dependent variable must each go to zero; the second partials must have the same sign and each must be nonzero. This would mean $u_{xx} + u_{yy} \neq 0$ which is not the case here. Since u does not have an extremum within the boundary it must be bracketed by values of u on the boundary.

Figure 4 is an initial guess for a grid about a body. This grid was generated by placing points evenly on a circular outer boundary and joining them to points on the inner boundary, the body. Then points were evenly spaced along these connecting lines to generate the nodes. Figure 5 shows the results of 100 iterations of the elliptic solver. There has been a general smoothing of lines throughout. Figure 6 is a detail of the initial grid in figure 4 showing grid breakdown near one of the convex points. Figure 7 is this same area after 100 iterations of the elliptic generator

The main disadvantage to using this system of equations is that there is no control over grid point spacing. For each given set of boundary values there is

only one final distribution of interior points. An alternative is to use the nonhomogeneous form of the elliptic equation, known as Poisson's equation

$$\nabla^2 \xi = p \quad (41)$$

$$\nabla^2 \eta = q \quad (42)$$

In such equations p and q can be used to modify the strong tendency of the Laplacian to equalize all cell volumes [7]. Another alternative takes the fully converged grid and re-spaces the points along ξ lines in order to cluster points near the body. [4]

Initial Grid

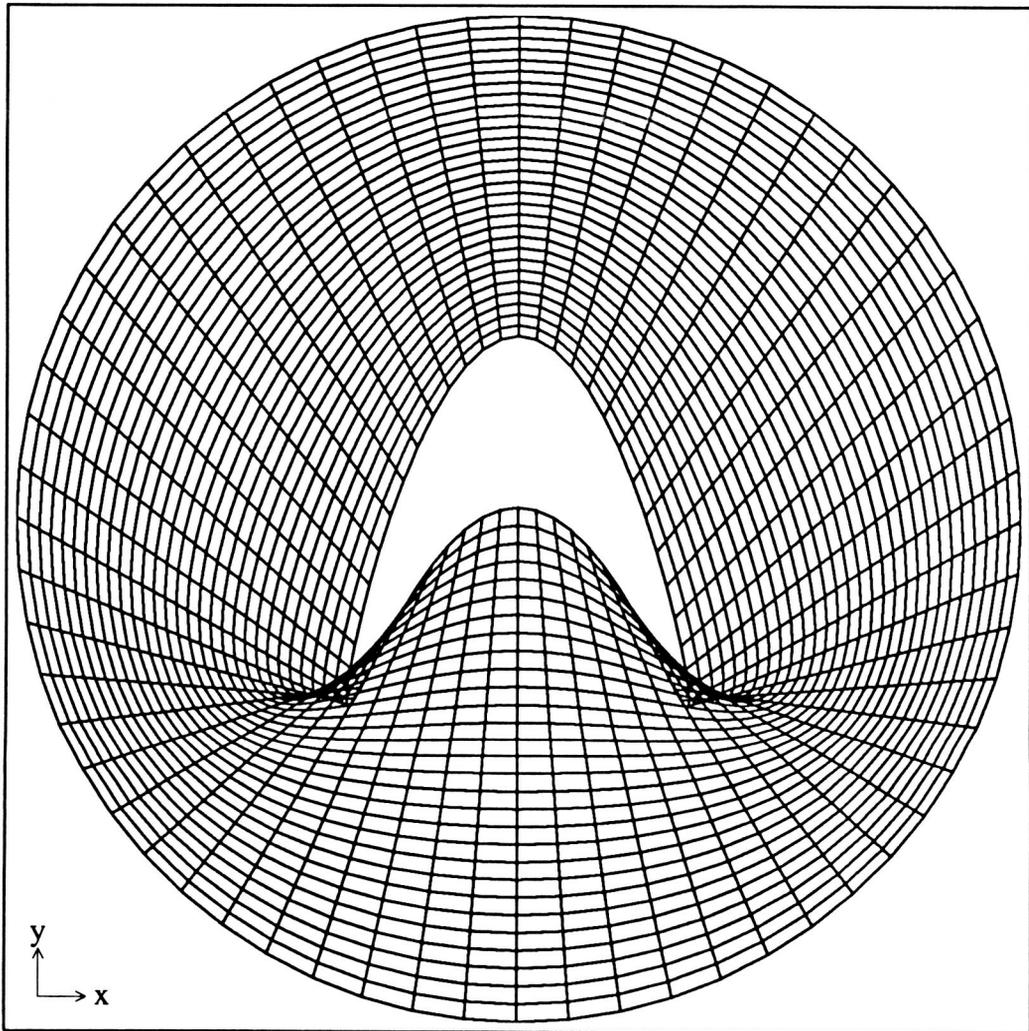


Figure 4: $\tau=0.5$, $\text{cam}=0.8$, Initial Grid
file: s5t8.00ps

The Effect of the Elliptic Solver

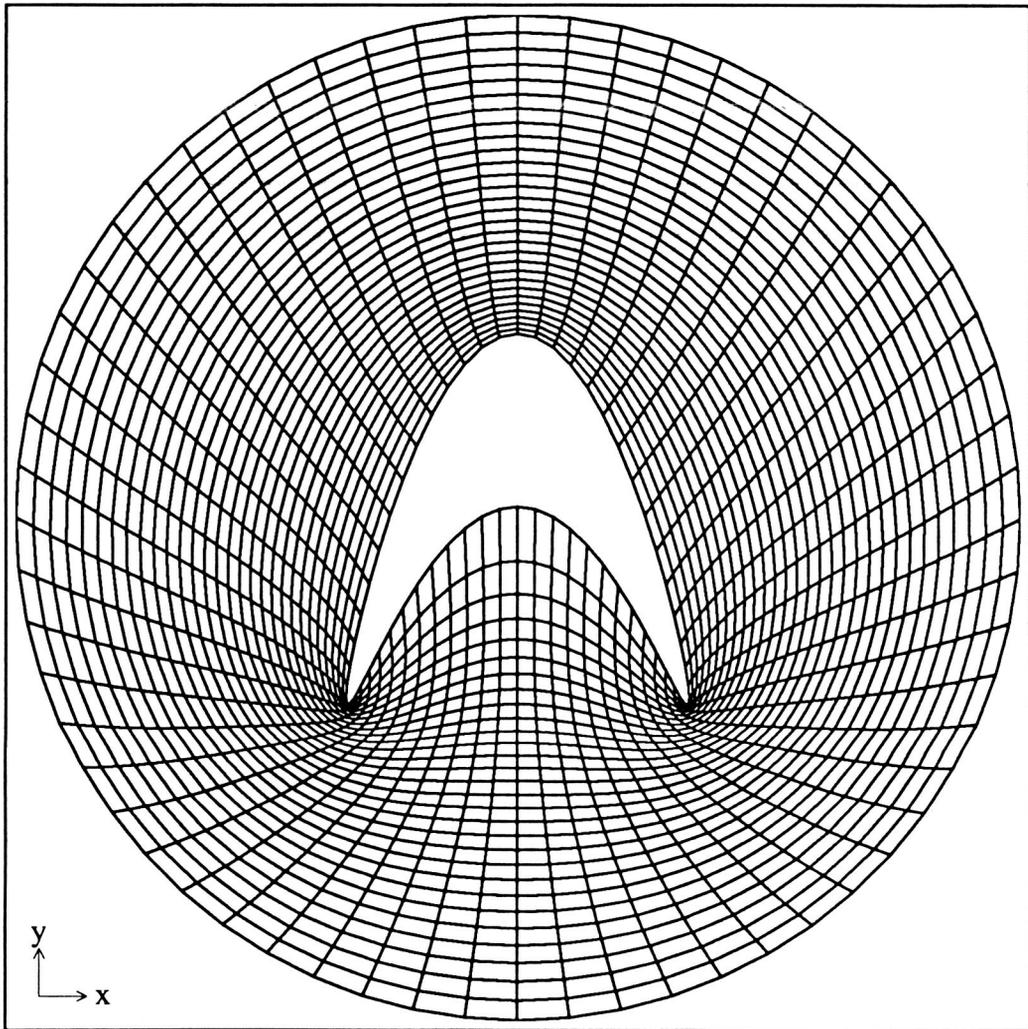


Figure 5: $\tau=0.5$, $\text{cam}=0.8$, pass 100

file: s5t8.10ps

Initial Grid

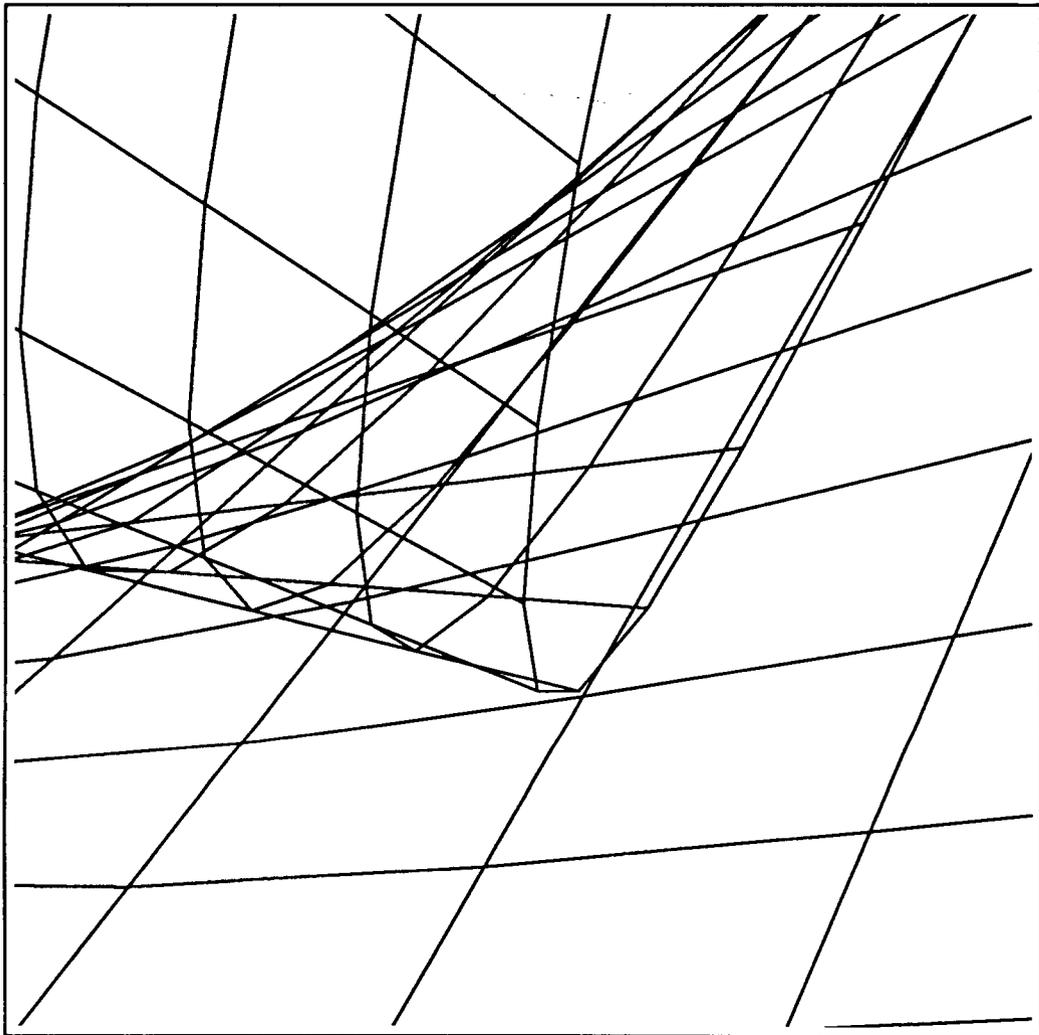


Figure 6: $\tau=0.5$, $\text{cam}=0.8$, detail of initial grid of figure 4
file: s5t8.00ptps

The Effect of the Elliptic Solver

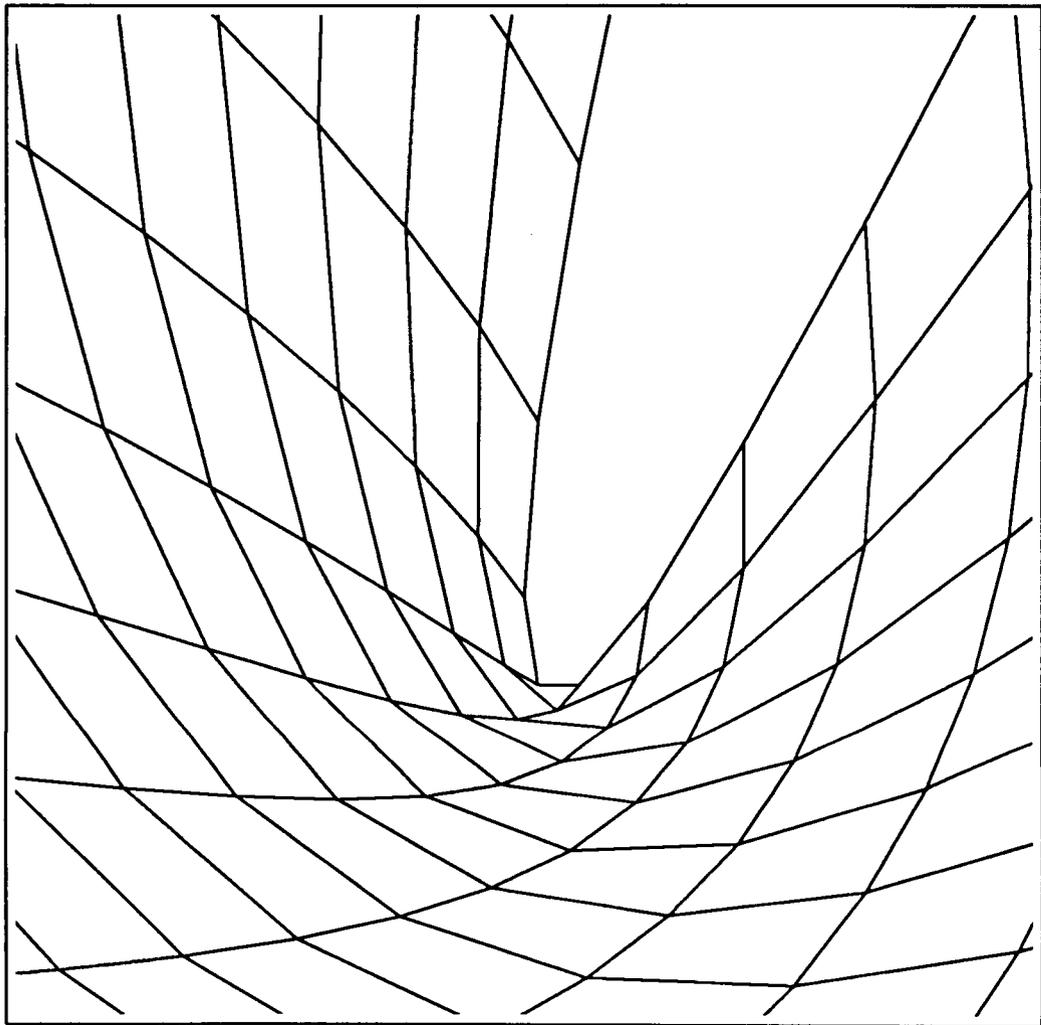


Figure 7: $\tau=0.5$, $\text{cam}=0.8$, pass 100, detail of final grid of figure 5
file: s5t8.10ptps

5 Hyperbolic Grid Generation

Steger [8] developed a grid generation scheme based on a set of equations which together form a hyperbolic system. These equations are expressions of constraints made on the grid generating system. These constraints are as follows:

- Orthogonality is explicitly specified,

$$\vec{\nabla}\xi \cdot \vec{\nabla}\eta \equiv \xi_x \eta_x + \xi_y \eta_y = 0. \quad (43)$$

- Grid cell volume (i.e. area in two dimensions), V , is user defined. Since, in the numerical implementation $\Delta\xi = \Delta\eta = 1$,

$$dx dy = (x_\xi y_\eta - x_\eta y_\xi) d\xi d\eta \equiv J^{-1} d\xi d\eta \quad (44)$$

$$d\xi d\eta = (\xi_x \eta_y - \xi_y \eta_x) dx dy \equiv J dx dy \quad (45)$$

Thus the Jacobian should be nonsingular and approximately equal to the physical cell volume, $\Delta x \Delta y$.

The resultant system is thus represented by

$$\xi_x \eta_x + \xi_y \eta_y = 0 \quad (46)$$

$$\xi_x \eta_y - \xi_y \eta_x = J \quad (47)$$

$$(48)$$

or, after a change of variables, again using the transformation metrics,

$$x_\xi x_\eta + y_\xi y_\eta = 0 \quad (49)$$

$$x_\xi y_\eta - x_\eta y_\xi = J^{-1} \equiv V \quad (50)$$

Equations (49) and (50) comprise a system of nonlinear partial differential equations with initial data in η . Through a process known as local linearization, shown below, it is found that this system is hyperbolic and thus can be marched in η from initial data along the body, $\eta = 0$.

If the substitutions $x = x^0 + \tilde{x}$ and $y = y^0 + \tilde{y}$ are made, where x^0 and y^0 represent a known nearby state, and \tilde{x} and \tilde{y} are of the order of Δx and Δy , local linearization can be performed by making these substitutions for x and y in equations (49) and (50). Starting with the first equation,

$$x_\xi x_\eta = (x^0 + \tilde{x})_\xi (x^0 + \tilde{x})_\eta \quad (51)$$

$$= x_\xi^0 x_\eta^0 + x_\xi^0 \tilde{x}_\eta + x_\eta^0 \tilde{x}_\xi + \tilde{x}_\xi \tilde{x}_\eta \quad (52)$$

$$\doteq x_\xi^0 x_\eta^0 + x_\xi^0 \tilde{x}_\eta + x_\eta^0 \tilde{x}_\xi \quad (53)$$

$$\doteq x_\xi^0 x_\eta + x_\eta^0 (x_\xi - x_\xi^0) \quad (54)$$

Similarly, the remaining terms in the two equations become

$$y_\xi y_\eta \doteq y_\xi^0 y_\eta + y_\eta^0 y_\xi - y_\xi^0 y_\eta^0 \quad (55)$$

$$x_\xi y_\eta \doteq y_\eta^0 x_\xi + x_\xi^0 y_\eta - x_\xi^0 y_\eta^0 \quad (56)$$

$$x_\eta y_\xi \doteq y_\xi^0 x_\eta + x_\eta^0 y_\xi - x_\eta^0 y_\xi^0 \quad (57)$$

Using these substitutions equation (49) becomes

$$(x_\xi^0 x_\eta + x_\eta^0 x_\xi - x_\xi^0 x_\eta^0) + (y_\xi^0 y_\eta + y_\eta^0 y_\xi - y_\xi^0 y_\eta^0) = 0 \quad (58)$$

But since $x_\xi x_\eta + y_\xi y_\eta = 0$ this reduces to

$$x_\xi^0 x_\eta + x_\eta^0 x_\xi + y_\xi^0 y_\eta + y_\eta^0 y_\xi = 0 \quad (59)$$

or, in matrix notation

$$\begin{pmatrix} x_\eta^0 & y_\eta^0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}_\xi + \begin{pmatrix} x_\xi^0 & y_\xi^0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}_\eta = 0 \quad (60)$$

Doing the same for equation (50)

$$(y_\eta^0 x_\xi + x_\xi^0 y_\eta - x_\xi^0 y_\eta^0) - (y_\xi^0 x_\eta + x_\eta^0 y_\xi - x_\eta^0 y_\xi^0) = V \quad (61)$$

Since by equation (50), $(x_\xi^0 y_\eta^0 - x_\eta^0 y_\xi^0) = V^0$ the equation becomes

$$y_\eta^0 x_\xi + x_\xi^0 y_\eta - y_\xi^0 x_\eta - x_\eta^0 y_\xi = V + V^0 \quad (62)$$

or

$$\begin{pmatrix} y_\eta^0 & -x_\eta^0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}_\xi + \begin{pmatrix} -y_\xi^0 & x_\xi^0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}_\eta = V + V^0 \quad (63)$$

Combining equations (60) and (63)

$$\begin{pmatrix} x_\eta^0 & y_\eta^0 \\ y_\eta^0 & -x_\eta^0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}_\xi + \begin{pmatrix} x_\xi^0 & y_\xi^0 \\ -y_\xi^0 & x_\xi^0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}_\eta = \begin{pmatrix} 0 \\ V + V^0 \end{pmatrix} \quad (64)$$

which can be represented as

$$A\vec{R}_\xi + B\vec{R}_\eta = f. \quad (65)$$

If B , whose determinant is the Jacobian transformation, is non-zero, all cell volumes are non-zero, and its inverse, B^{-1} , exists. Equation (64) can be then be transformed into

$$B^{-1}A\vec{R}_\xi + \vec{R}_\eta = B^{-1}f. \quad (66)$$

It has been shown [1] that this type of equation is hyperbolic and marches in the η direction if $B^{-1}A$ has real, distinct eigenvalues, or if $B^{-1}A$ is a symmetric matrix. Expanding $B^{-1}A$ and performing the indicated multiplication produces

$$B^{-1}A = \frac{\begin{pmatrix} x_\xi^0 & -y_\xi^0 \\ y_\xi^0 & x_\xi^0 \end{pmatrix}}{x_\xi^0 x_\xi^0 - (-y_\xi^0)y_\xi^0} \begin{pmatrix} x_\eta^0 & y_\eta^0 \\ y_\eta^0 & -x_\eta^0 \end{pmatrix} \quad (67)$$

$$= \frac{\begin{pmatrix} x_\xi^0 x_\eta^0 - y_\xi^0 y_\eta^0 & x_\xi^0 y_\eta^0 + y_\xi^0 x_\eta^0 \\ y_\xi^0 x_\eta^0 + x_\xi^0 y_\eta^0 & y_\xi^0 y_\eta^0 - x_\xi^0 x_\eta^0 \end{pmatrix}}{(x_\xi^0)^2 + (y_\xi^0)^2} \quad (68)$$

which is symmetric, fulfilling the latter requirement. If λ represents the eigenvalues the eigenvalue equation is

$$\left(\frac{(x_\xi^0 x_\eta^0 - y_\xi^0 y_\eta^0)}{(x_\xi^0)^2 + (y_\xi^0)^2} - \lambda \right)^2 - \left(\frac{y_\xi^0 x_\eta^0 + x_\xi^0 y_\eta^0}{(x_\xi^0)^2 + (y_\xi^0)^2} \right)^2 = 0 \quad (69)$$

or

$$\lambda_\pm = \frac{V^0 \pm (x_\xi^0 x_\eta^0 + y_\xi^0 y_\eta^0)}{(x_\xi^0)^2 + (y_\xi^0)^2} \quad (70)$$

Thus, since V^0 , the inverse Jacobian, is non-zero the eigenvalues are real, and distinct when $(x_\xi^0 x_\eta^0 + y_\xi^0 y_\eta^0)$ is non-zero

5.1 Numerical technique

The numerical solution of equation (66) can be carried out using the line inversion mentioned earlier in the introduction and in the section on the elliptic generator. The main difference here is that only one pass is performed in generating the grid. Unlike the elliptic generating system, which improved an initial grid, and therefore was dependent on the quality of that grid for convergence to a solution, the hyperbolic system generates a grid from the known boundary conditions.

Replacing the partial derivatives in equation (65) by finite difference operators produces

$$B^{-1}A \frac{\bar{\delta}_\xi \vec{R}}{2\Delta\xi} + \frac{\nabla_\eta \vec{R}}{\Delta\eta} = B^{-1}f \quad (71)$$

where all terms are second order accurate except for the first order replacement of \vec{R}_η .

Since, in the numerical implementation, $\Delta\xi = \Delta\eta = 1$, the expansion of equation (71) is

$$\frac{B^{-1}A}{2}(\vec{R}_{j+1,k+1} - \vec{R}_{j-1,k+1}) + \vec{R}_{j,k+1} - \vec{R}_{j,k} = B^{-1}f \quad (72)$$

Rearranging to gather terms involving $k + 1$ on one side produces

$$a_{j,k} \vec{R}_{j-1,k+1} + b_{j,k} \vec{R}_{j,k+1} + c_{j,k} \vec{R}_{j+1,k+1} = RHS \quad (73)$$

where

$$a_{j,k} = -\frac{B^{-1}A}{2} \quad (74)$$

$$b_{j,k} = 1 \quad (75)$$

$$c_{j,k} = \frac{B^{-1}A}{2} \quad (76)$$

$$RHS = B^{-1}f + \vec{R}_{j,k} \quad (77)$$

The coefficients are all evaluated at the same index values of j and k , where x and y are known. The partials with respect to ξ are straightforward central differences. The η partials can then be obtained by solving for them in equations (49) and (50)

$$x_\eta = -\frac{y_\xi V}{(x_\xi)^2 + (y_\xi)^2} \quad (78)$$

$$y_\eta = \frac{x_\xi V}{(x_\xi)^2 + (y_\xi)^2} \quad (79)$$

Equation (73) generates the set of equations in j , from $j=1$ to $j=jmax$. These equations are solved, as before, using Thomas' algorithm.

The results of using this algorithm, designated `hyg2d`, can be seen in figures 8 through 11 for bodies of increasing concavity². Note that as the bodies increase in concavity the radial grid lines start to cluster, and finally, the grid breaks down. figure 12 and 13 show the marching out of the grid just before and at this breakdown. Figure 14 is a detail view of one point of the body.

²See the section Notations and Definitions and figures 69-72 for more information on concavity and how it is meant here.

The Effect of Camber (concavity) on the Hyperbolic Solver

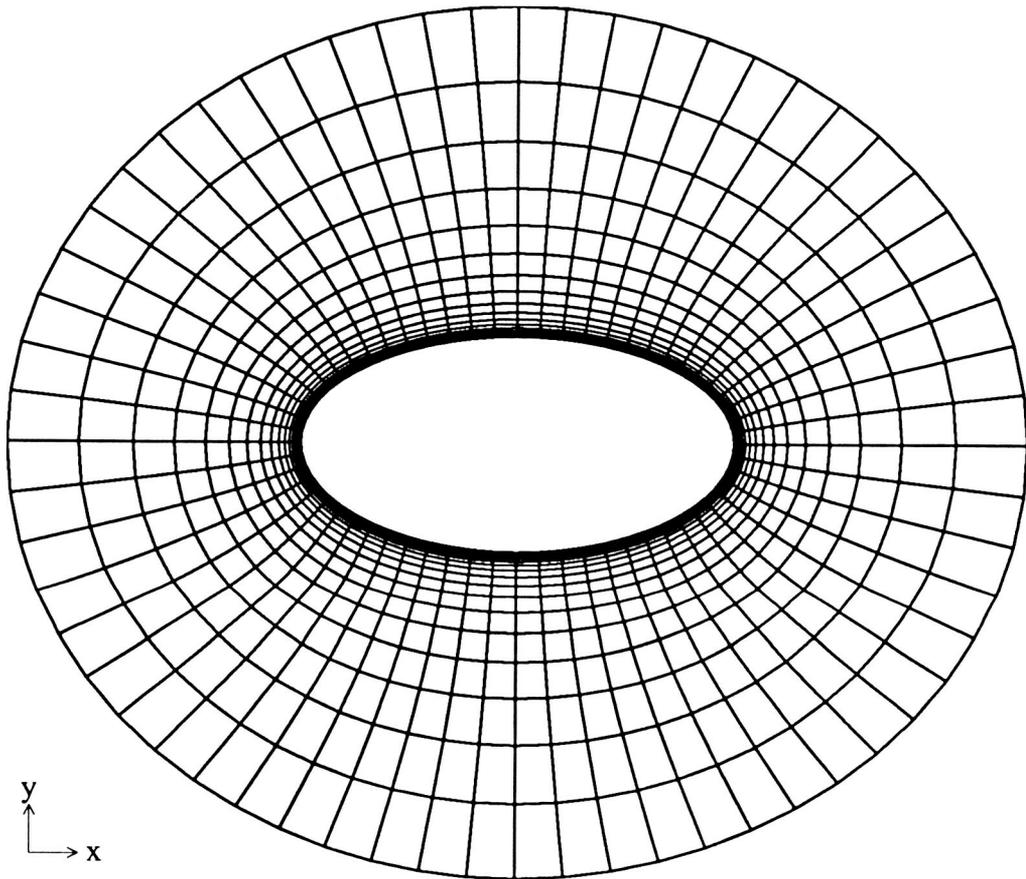


Figure 8: $\tau = .5$, $\text{cam} = 0$
file: h50y0.ps

The Effect of Camber (concavity) on the Hyperbolic Solver

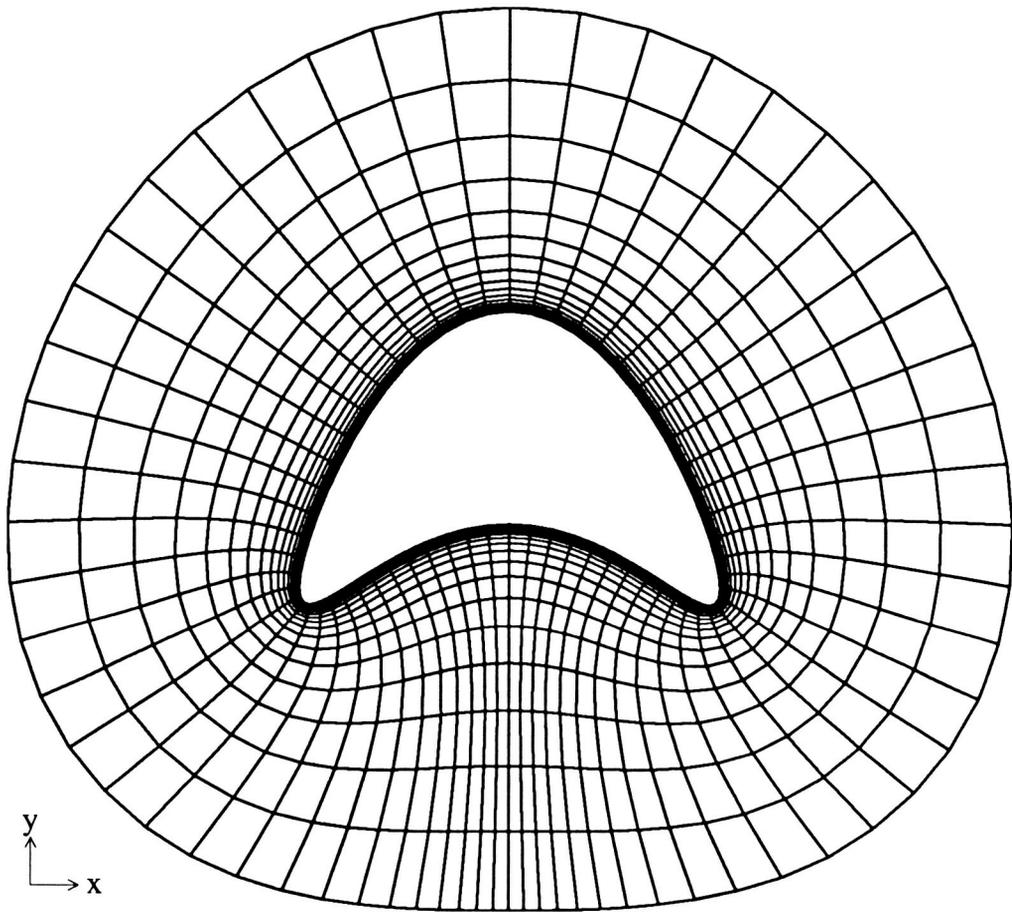


Figure 9: $\tau=0.5$, $\text{cam}=0.40$
file: h50y40.ps

The Effect of Camber (concavity) on the Hyperbolic Solver

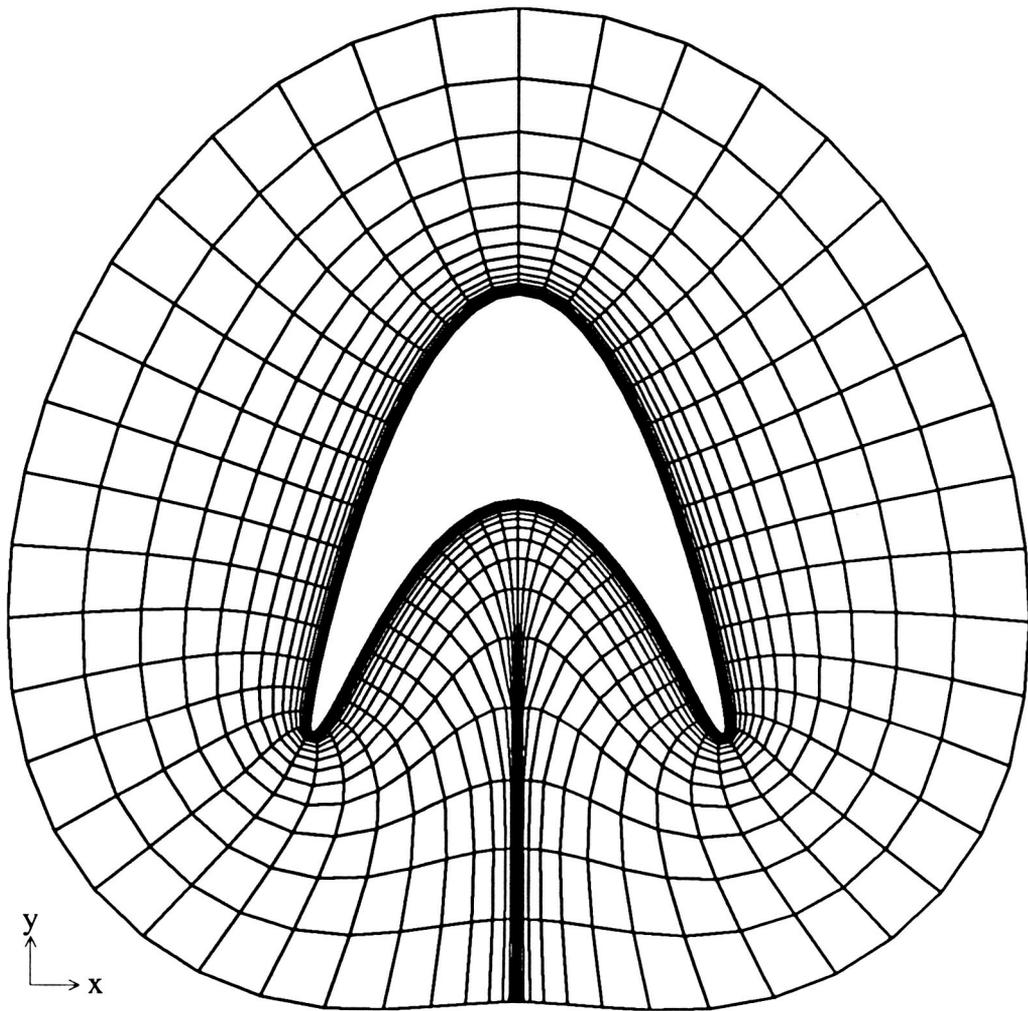


Figure 10: $\tau = .5$, $\text{cam} = 0.80$
file: h50y80.ps

The Effect of Camber (concavity) on the Hyperbolic Solver

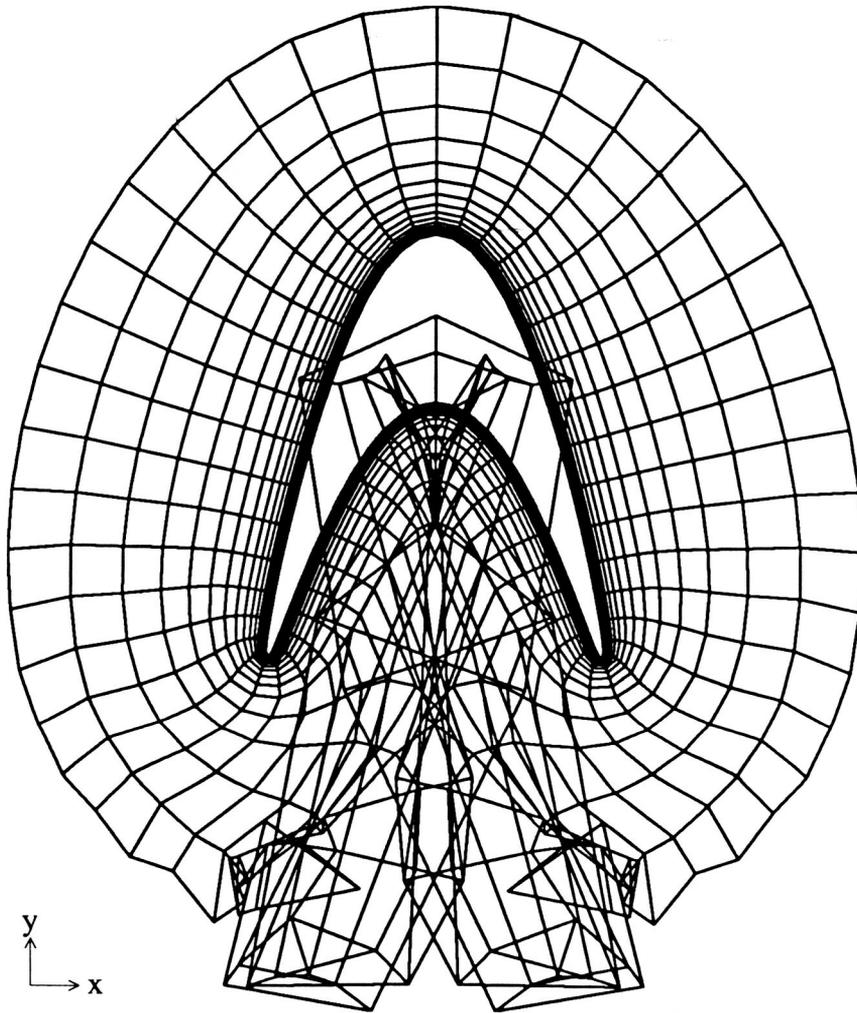


Figure 11 : $\tau = .5$, $\text{cam} = 1.0$
file: h5y10.ps

Marching Out of Hyperbolic Generator

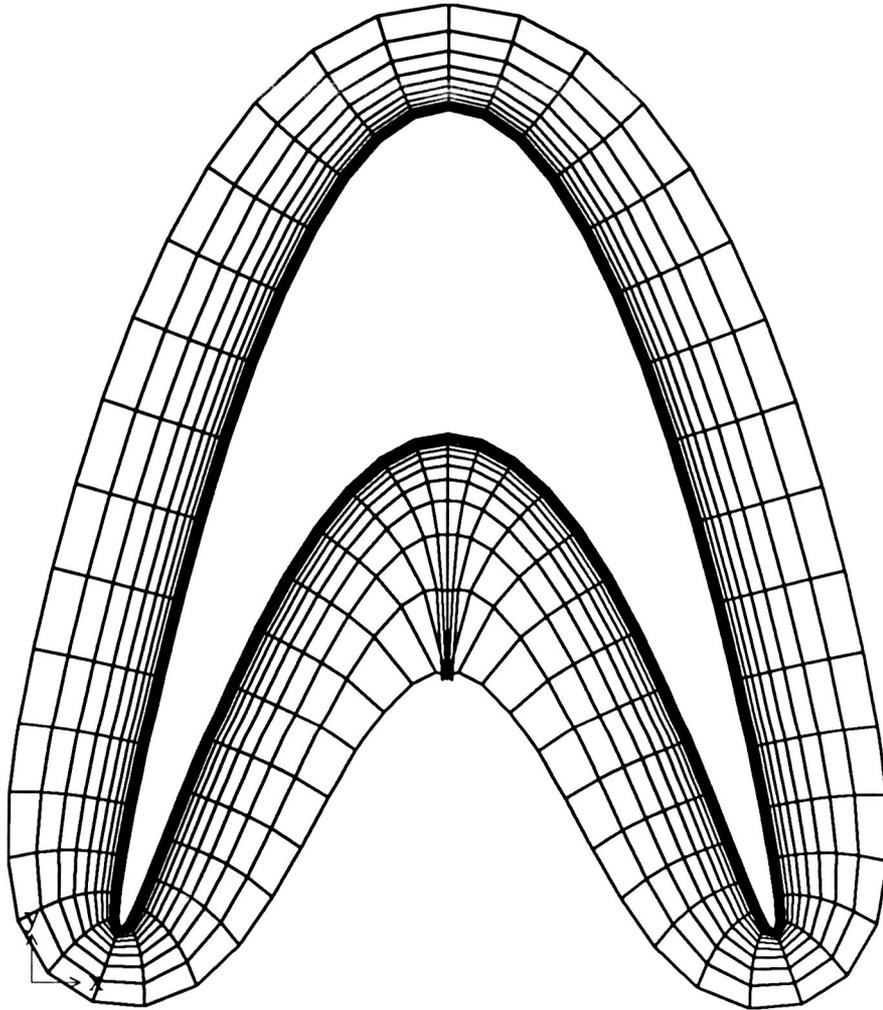


Figure 12: $\tau=0.5$, $\text{cam}=1.0$, just before cavity-caused breakdown
file: h5y10.k15ps

Marching Out of Hyperbolic Generator

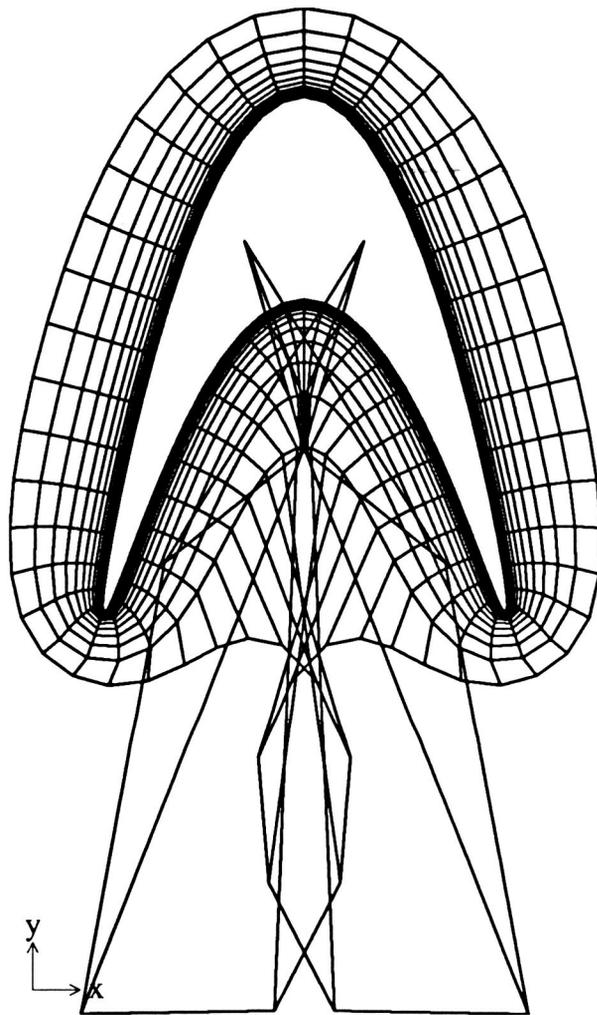


Figure 13: $\tau=0.5$, $\text{cam}=1.0$, breakdown due to strong concavity
file: h5y10.k16ps

The Effect of Camber (concavity) on the Hyperbolic Solver

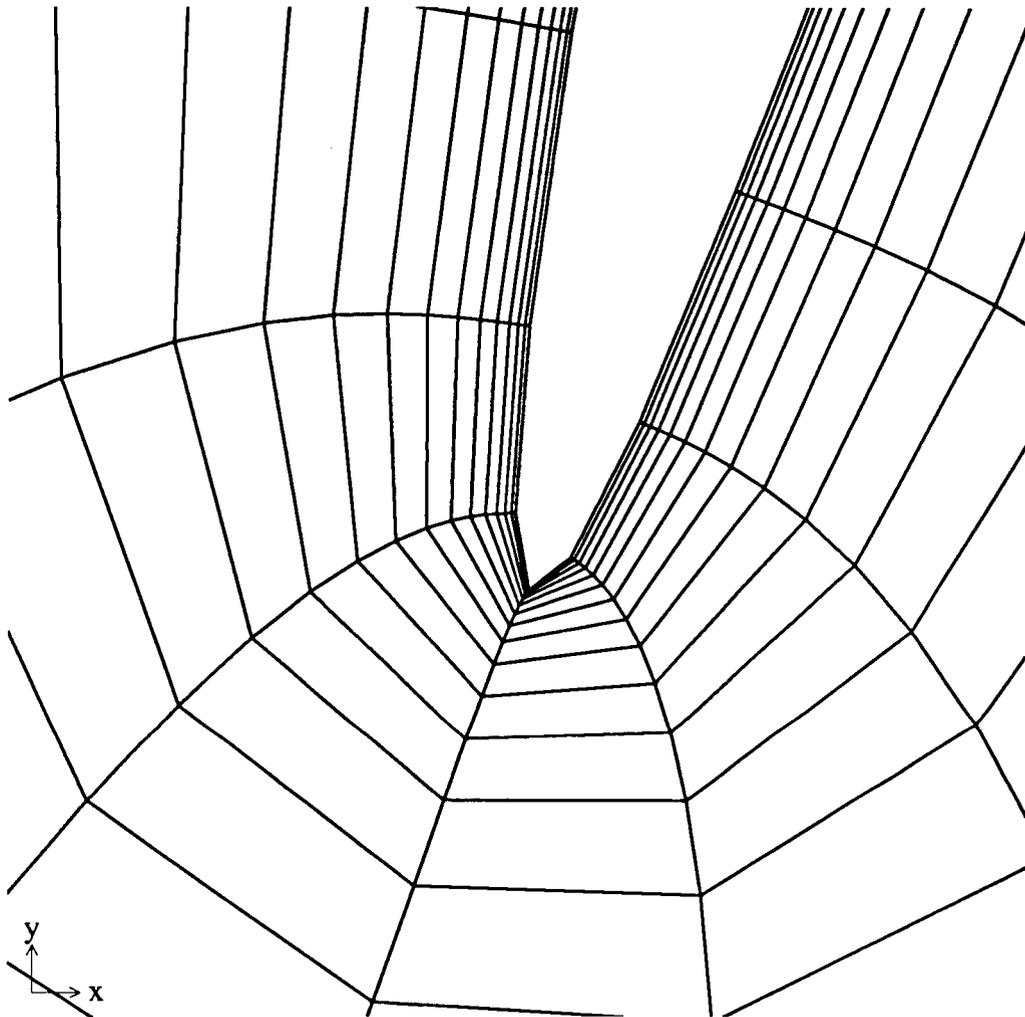


Figure 14: $\tau=0.5$, $\text{cam}=1.0$ detail
file: h5y10.ptps

6 Combining Elliptic and Hyperbolic Equations

Each of the grid generation schemes discussed, the elliptic and the hyperbolic, has a profile of advantages and disadvantages. The elliptic, through its smoothing action is able to generate a grid about more complex bodies than the hyperbolic. But it requires an initial grid to start from; it must iterate to convergence for a solution; Unless source terms are added, there is little control over point distribution; these source terms are not easy to use. The hyperbolic, on the other hand, creates a completed grid in the order of time it takes the elliptic to make one pass. This particular hyperbolic scheme has, additionally, orthogonality constraints, and user-control of grid cell volumes, providing a strong control over point distribution. Its chief disadvantage is that unlike the elliptic it breaks down sooner in areas of strong concavities or convexities.

A scheme that uses the best of both would be better than either. Steger [5] has developed such a grid generation scheme. His procedure is to add a scalar multiple of the elliptic system, equation (32), to the right side of the hyperbolic system, equation (66), producing

HYPERBOLIC :

$$B^{-1}A\vec{R}_\xi + \vec{R}_\eta = B^{-1}f \quad (66)$$

$\mu \times$ ELLIPTIC :

$$0 = \mu(\alpha\vec{R}_{\xi\xi} - 2\beta\vec{R}_{\xi\eta} + \gamma\vec{R}_{\eta\eta}) \quad (32)$$

COMBINED :

$$B^{-1}A\vec{R}_\xi + \vec{R}_\eta = B^{-1}f + \mu(\alpha\vec{R}_{\xi\xi} - 2\beta\vec{R}_{\xi\eta} + \gamma\vec{R}_{\eta\eta}) \quad (80)$$

The control parameter μ can then be used to emphasize the hyperbolic or elliptic system. For small μ the hyperbolic dominates and for large μ the elliptic dominates.

Viewed as a hyperbolic system the constraint of orthogonality is now relaxed by the strength of the elliptic terms when they sum to nonzero. Loosening this constraint helps avoid grid breakdown. In the volume equation the elliptic terms serve to smooth differences between cell volumes. Viewed as an elliptic system, the Laplace equation has been transformed into a Poisson equation with the hyperbolic terms serving as the source terms. These source terms allow concentration of ξ , and η lines. Which side dominates is controlled by μ . When it equals zero the system is purely hyperbolic. As it increases from zero the right side of the equations become more dominant until μ is over about 3,000, when the system is effectively elliptic. This parameter can be made a function of spatial position, in order to, for instance cluster lines near the body. It can also be made a function of gradients in the flow field so that, in an area of high gradients, μ can be set low, turning on the hyperbolic's volume control, to increase point density in this area. In areas of grid breakdown μ can be set high, turning on the elliptic, to repair the breakdown.

6.1 Solution Procedure

Replacing the partial derivatives in equation (6) by finite difference operators produces

$$B^{-1}A \frac{\bar{\delta}_\xi \vec{R}}{2\Delta\xi} + \frac{\nabla_\eta \vec{R}}{\Delta\eta} = B^{-1}f + \mu \left[\alpha \frac{\delta_\xi^2 \vec{R}}{(\Delta\xi)^2} - 2\beta \frac{\bar{\delta}_\xi \bar{\delta}_\eta \vec{R}}{(2\Delta\xi)(2\Delta\eta)} + \gamma \frac{\delta_\eta^2 \vec{R}}{(\Delta\eta)^2} \right] \quad (81)$$

where all terms are second order accurate except for the first order replacement of \vec{R}_η .

Since, in the numerical implementation, $\Delta\xi = \Delta\eta = 1$, the expansion of equation (81) is

$$\begin{aligned} \frac{B^{-1}A}{2}(\vec{R}_{j+1,k} - \vec{R}_{j-1,k}) + \vec{R}_{j,k} - \vec{R}_{j,k-1} = \\ B^{-1}f + \mu\alpha(\vec{R}_{j+1,k} - 2\vec{R}_{j,k} + \vec{R}_{j-1,k}) \\ + \frac{\mu\beta}{2}(\vec{R}_{j+1,k+1} - \vec{R}_{j-1,k+1} - \vec{R}_{j+1,k-1} + \vec{R}_{j+1,k-1}) \\ + \mu\gamma(\vec{R}_{j,k+1} - 2\vec{R}_{j,k} + \vec{R}_{j,k-1}) \end{aligned} \quad (82)$$

Rearranging to gather terms involving k on one side produces

$$a_{j,k}\vec{R}_{j-1,k} + b_{j,k}\vec{R}_{j,k} + c_{j+1,k}\vec{R}_{j,k} = RHS \quad (83)$$

where

$$a_{j,k} = -\frac{B^{-1}A}{2} - \mu\alpha \quad (84)$$

$$b_{j,k} = 1 + 2\mu(\alpha + \gamma) \quad (85)$$

$$c_{j,k} = -\frac{B^{-1}A}{2} - \mu\alpha \quad (86)$$

$$\begin{aligned} RHS = \vec{R}_{j,k-1} + \gamma\mu(\vec{R}_{j,k+1} + \vec{R}_{j,k-1}) \\ - \frac{\beta\mu}{2}(\vec{R}_{j+1,k+1} - \vec{R}_{j-1,k+1} - \vec{R}_{j+1,k-1} + \vec{R}_{j+1,k-1}) \end{aligned} \quad (87)$$

$$(88)$$

and all values in equations (84) – (88) not specifically indexed have index values of j and k , that is, they are evaluated at the points $x = x_{j,k}$, $y = y_{j,k}$.

Equation (83) can now be solved similar to the elliptic equations. Starting with $k=1$, letting the j index run from 1 to some j_{max} , and keeping the k index constant, a set of equations are generated. The left-hand coefficients form a tridiagonal matrix, each element of which is a two-by-two array. The set of equations is

solved using a modified form of Thomas' algorithm. The index variable k runs from 1 to some number k_{max} and therefore k_{max} sets of equations must be solved. Once these are solved the whole process is reiterated using the results of the last iteration, the resultant grid, as the source for generating new values for the right-hand side and the coefficients, of equation (83). The iteration process halts when grid changes are below some threshold value.

6.2 Testing Procedure

Since the elliptic can repair breakdowns in the grid (due to the hyperbolic), and since breakdown is a function of the number and strength of convexities and concavities, it was decided to test this new solver by using bodies of increasing concavity and/or convexity .

Although the hyperbolic generator breaks down in areas of either great concavity or great convexity, breakdowns due to the latter tend to be more localized and simpler to deal with. Uneven point spacing along the body on either side of the apex of the convexity causes the grid line radiating from the apex to swing over. This causes circumferential lines to cut through the body. But the number of nodes involved are generally limited to a few in either the circumferential (ξ) or radial (η) direction and can be handled by the fortified approach of Van Dalsem [9].

The breakdowns caused by a concavity tend to be more global as the strength of the concavity, and the percent body surface the concavity takes up, increase. Therefore, it was decided to approach the problem by examining bodies containing concavities of increasing strength.

6.3 The Unmodified Algorithm

Implementation of the elliptic-like algorithm just described requires an initial grid. The main requirement placed on this grid is that its outer boundary not self-intersect or intersect with interior grid lines. The hyperbolic generator was used to generate the initial grid of figure 15. This generator offers the advantage of being closely related to the hyperbolic part of the combined generator. The similarity can be used to avoid problems such as "collapsing" grid lines, η lines that tend to fall back towards the body rather than away from it with succeeding iterations. This defect can occur when the initial grid has different spacing in the η direction than the hyperbolic component provides. Here μ is set low near the body and increases geometrically in the radial direction. Thus the hyperbolic is enforced close to the body allowing clustering of points, while the elliptic is more in force deeper into the field out to the outer boundary, smoothing the sharper curvature of grid lines of the initial grids. Figure 16 shows the grid after 100 iterations. Note that because the outer boundary the clustering on it cannot be dealt with.

An alternate initial grid, having a strong concavity and the final grid drawn from it are shown in figures 17 and 18. Figure 19 shows detail of the latter grid. at a convex point. This initial grid was generated by specifying a circular outer boundary and spacing the points on it evenly. Lines are drawn connecting these points with points on the body. Points are then evenly spaced along these connecting lines to form the grid. Here the outer boundary point spacing was fixed to avoid the clustering of the hyperbolic system. For high concavities it seems to generate a reasonable grid. Because the outer boundary point spacing is not a strong function of the inner boundary, their connecting lines (ξ lines)

sometimes have to make large sweeps. As shown in figure 19 orthogonality can break down. Other drawbacks are that there is no way of knowing which points are best connected to each other and, eventually a body shape is reached at which this algorithm breaks down.

6.4 Floating the Boundary

The major problem with the original algorithm is that it found a solution in an elliptic way. That is, the boundaries were kept fixed while the inner field points were adjusted. The first modification to the algorithm above, and used in subsequent modifications, was to float the boundary. This was achieved by turning the hyperbolic on at the outer boundary to generate a new outer boundary. This technique is more amenable in areas where there is no physical outer boundary, such as for airfoils, areas that use hyperbolic generators to form grids.

Once the outer boundary is floated initial grids containing defects in their outer boundary, such as those generated by `hyg2d`, can be used. Such defects include boundaries that intersect other parts of the grid. The smoothing action of the elliptic, coupled with the boundary generating action of the hyperbolic can then repair defects in the initial grid. Figures 20 to 33 are a series of grids showing the effect of floating the boundary.

The drawback to this latest technique is that at some point a body is reached whose concavity makes it too difficult to generate a grid about. The generator then fails to improve the grid. Even before this point is reached, though, the range of values μ can take on narrows, making it difficult to choose a set of values for it. Here μ is just a function of η . For bodies with weak concavities it can take on any value at any η . But as concavity grows grid improvement becomes very sensitive

to the value of μ , to the point that it becomes very difficult to find values for μ that will enable a solution to converge.

6.5 Using Compressed-Cell Initial Grids

The problem with the previous technique was in the initial grids used. They contain too many defects and slow the convergence procedure. If cell volumes are extremely small the distance from inner to outer boundary makes intersections of grid lines more difficult. This led to modifying the algorithm to handle a different kind of initial grid, one with fewer defects. An initial grid of compacted cell volumes was used. During each iteration all cell volumes were then increased a fixed percent of their total volume. This had the benefit of starting with an initial grid which had very few, if any, defects and smoothing out defects as they developed, rather than starting with a grid containing many defects. Defects were generally isolated then, at or near the outer boundary. Each iteration was both a growth of the grid, somewhat like the hyperbolic, and also each iteration was a smoothing out in the interior, by the elliptic. Since the outer boundary is purely a product of the next lowest level, which was constantly being improved, this made for a very robust generator. Compared to the previous method this technique allows for a wider range of values for μ given the same body as before. Figures 34 to 45 are examples of this technique for bodies of high concavity. Cell volumes have been set to grow about fifteen percent per iteration. Figure 34 contains the initial grid but, since its cell volumes are compressed the cells can only be seen at several magnifications (figure 35). Figures 36 to 40 show the grid expanding during successive passes of the combined solver. Figure 41 is a close-up of the the grid at the 200th pass. Figures 42 to 45 shows that the technique can be applied

to even deeper cavities.

6.6 Curtailing Outer Boundary Movement within a Cavity

There are two main drawbacks to using the technique just discussed as it stands.

- A point is reached in which the cavity is too deep for this technique to work.
see figure 46
- Although a grid with no breakdown is possible for strong concavities, there tends to be less control over point distribution within these areas.

The technique in the previous section uses just the hyperbolic component to generate a new outer boundary with each iteration. The hyperbolic generates the points on this boundary, the η_{kmax} line from data on the next lower section of the grid, the η_{kmax-1} line. By the orthogonality constraint these points are normal to the η_{kmax-1} line. As the cell volumes increase this causes boundary self-intersection within a cavity with some depth. Because the interior is derived primarily elliptically and therefore a function of both boundaries there is a tendency therefore for grid breakdown near the outer boundary. But, since the inner boundary is okay, there is, by the maximum principle less breakdown of these interior grid lines than the boundary line. At each iteration these areas of breakdown are improved by the elliptic through communication with inner grid points. These improved areas are then used to generate the latest outer boundary. The fact that the numerical technique starts at the inner boundary may be part of the reason why the elliptic smoothes these areas so well.

The result of the above tendencies is that within the cavity the outer boundary moves, at each iteration, largely in the η direction and out of the cavity. Thus in strong concavities the ratio of cell side lengths can be very high. See figure? And, while volume control normally gives control over point distribution, in this case it is overridden by the smoothing action of the elliptic coupled with the hyperbolic's regeneration of the outer boundary, forcing the boundary out of the cavity. Within the cavity accuracy and resolution are thus lower in the η direction because of the resultant spacing. It was thought at first that point density was lower in the cavity, contributing to the increased cell aspect ratio, but analysis confirmed that for a given range of η point density is about the same within and without the cavity. This conforms with the fact that cell volumes had been functions of η only.

In and near an area of a concavity the gradients of the flow variables can be expected to be higher than average. Therefore Δ_ξ and Δ_η should actually be smaller than elsewhere to resolve the flow and provide a stable solution. There then appears to be a conflict between the need to maintain a small cell size and at the same time the need to cover the flow field out to several body diameters. It was therefore decided to modify how the outer boundary is generated within the cavity. A point on the boundary is chosen to serve as a hinge point. As the boundary points move away normal to the η_{kmax-1} line their movement is modified to some specified degree by a forcing function that brings corresponding points, on each side of the hinge point, together. When a set of points meet they are no longer allowed to float. Their corresponding cell volumes are then held constant through later iterations. Figures 47 to 62 show the results of using this technique

on the grid that caused breakdown of the previous technique. There is The result is that within the cavity the boundary overlaps itself along a line of symmetry of the cavity and it is now possible to generate a grid for deeper cavities also a higher point density within the grid.

Using the Combined Solver with a Fixed Outer Boundary

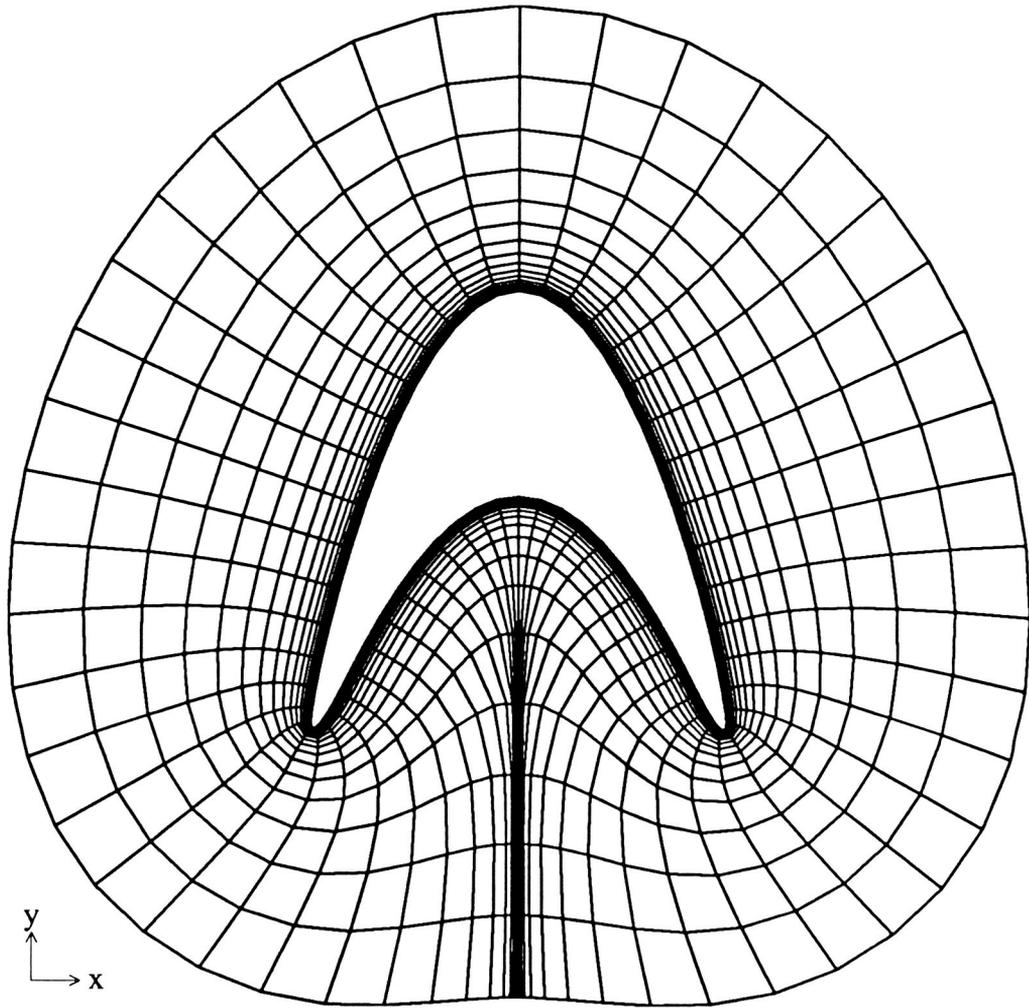


Figure 15: $\tau=0.5$, $\text{cam}=0.8$, initial grid
file: eh5y8.00ps

Using the Combined Solver with a Fixed Outer Boundary

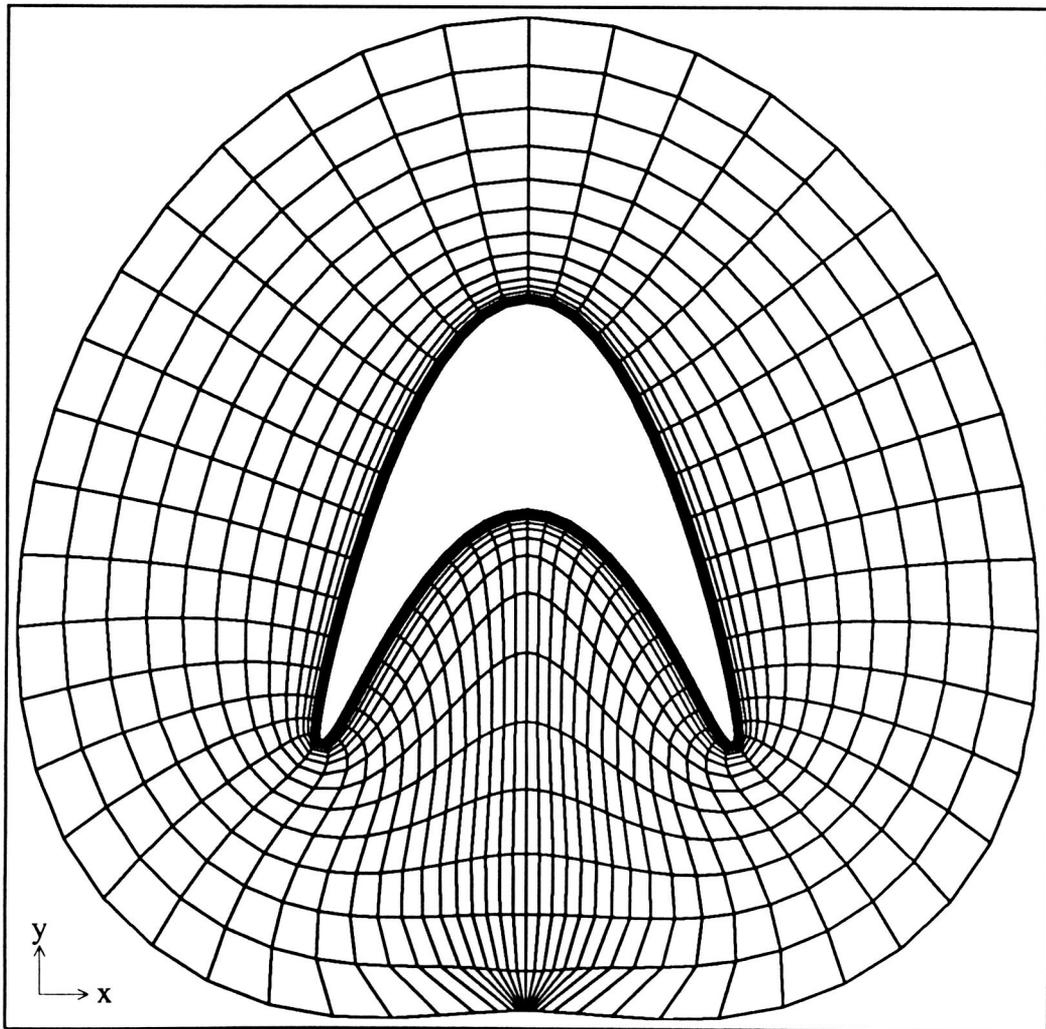


Figure 16: $\tau=0.5$, $\text{cam}=0.8$, pass number 100
file: eh5y8.10ps

The Effect of Camber (Concavity) on the Combined Elliptic-Hyperbolic Grid Generator

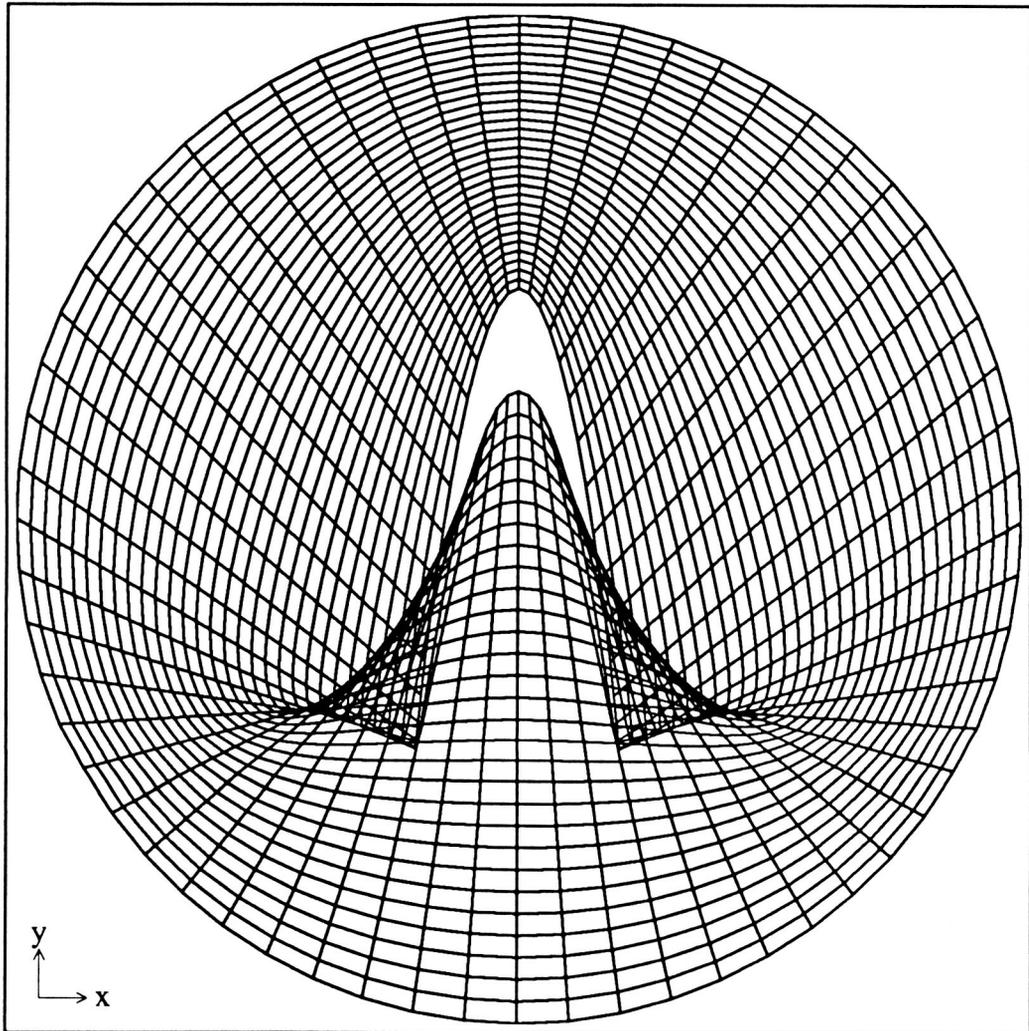


Figure 17: $\tau=0.5$, $\text{cam}=2.0$, Initial Grid

file : s5t20.p0pstmp

The Effect of Camber (Concavity) on the Combined Elliptic-Hyperbolic Grid Generator

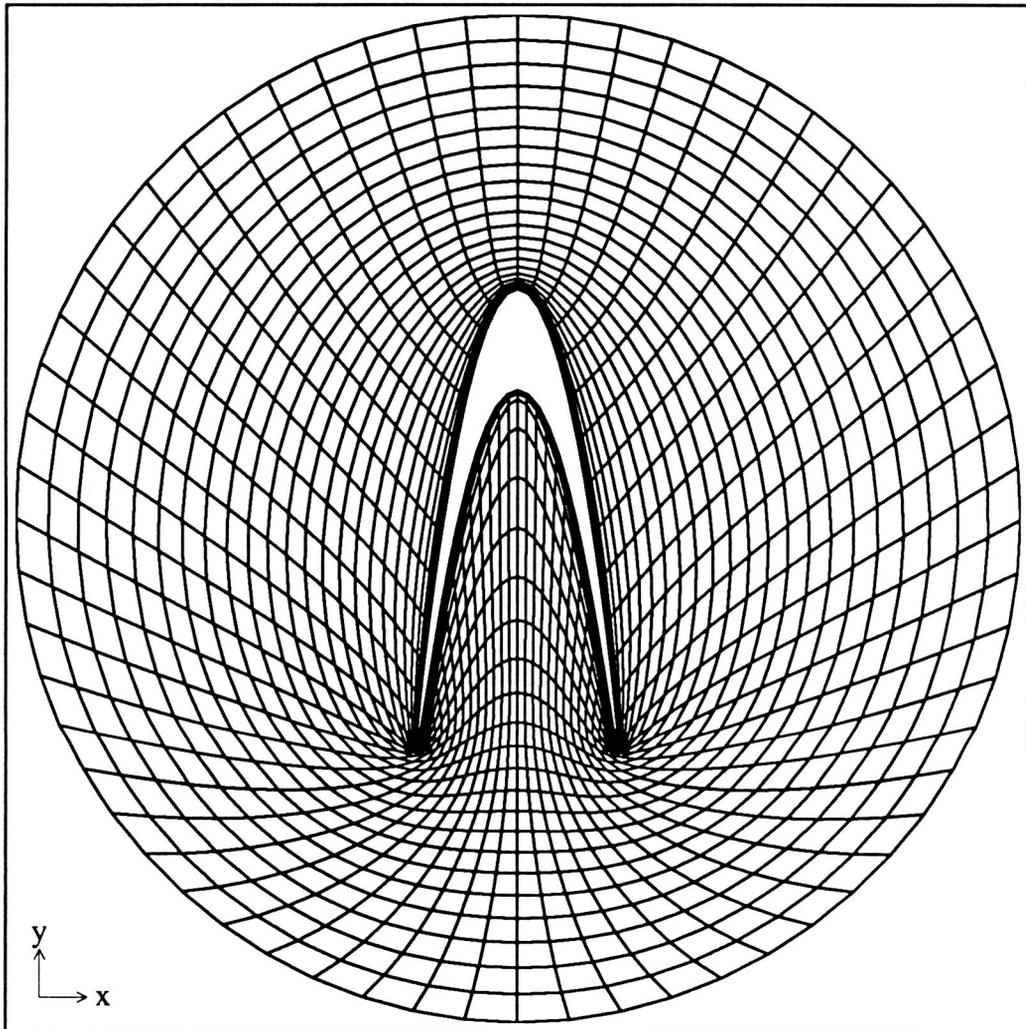


Figure 18: $\tau=0.5$, $\text{cam}=2.0$, pass 100
file: seh5t20.10ps

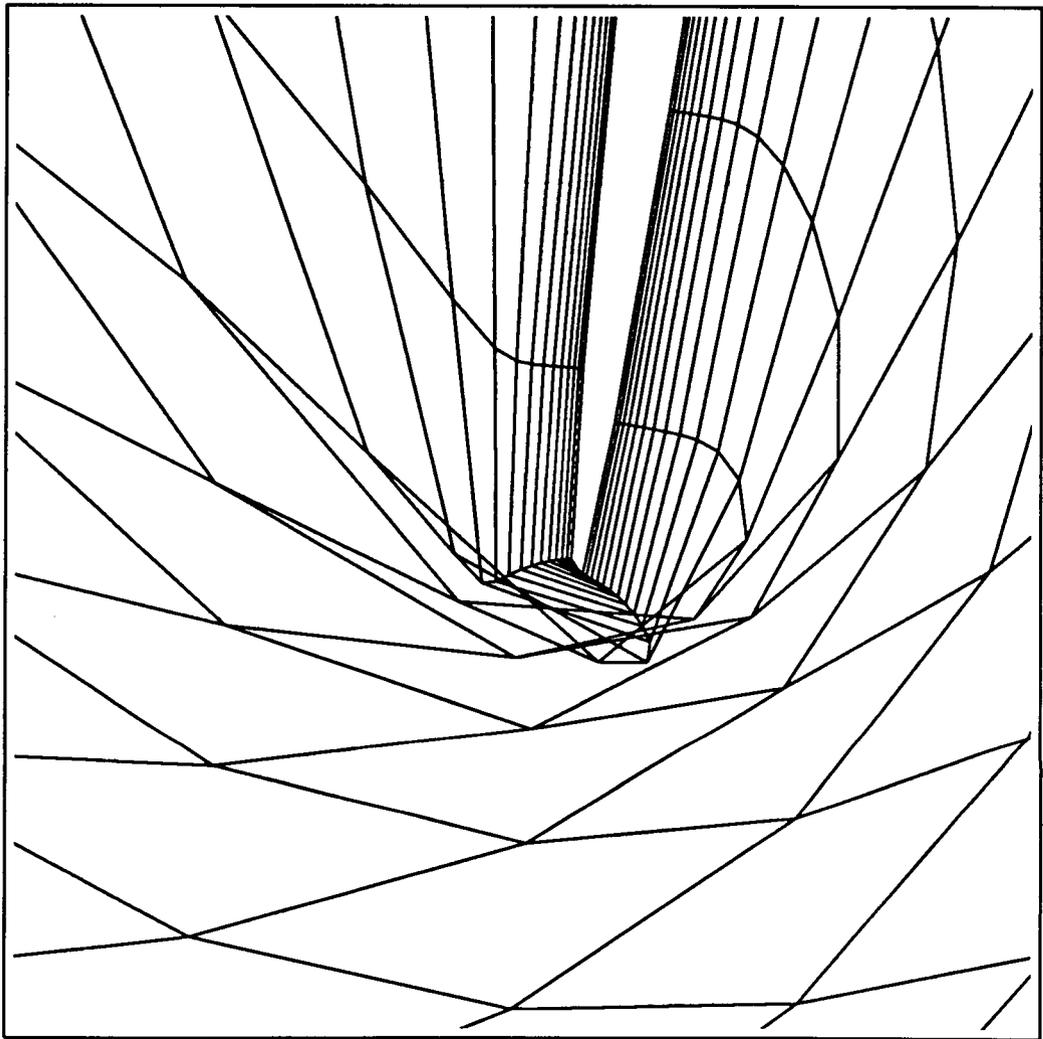


Figure 19: detail of figure 18

file :seh5t20.10ptps

Grid Improvement by Floating the outer Boundary

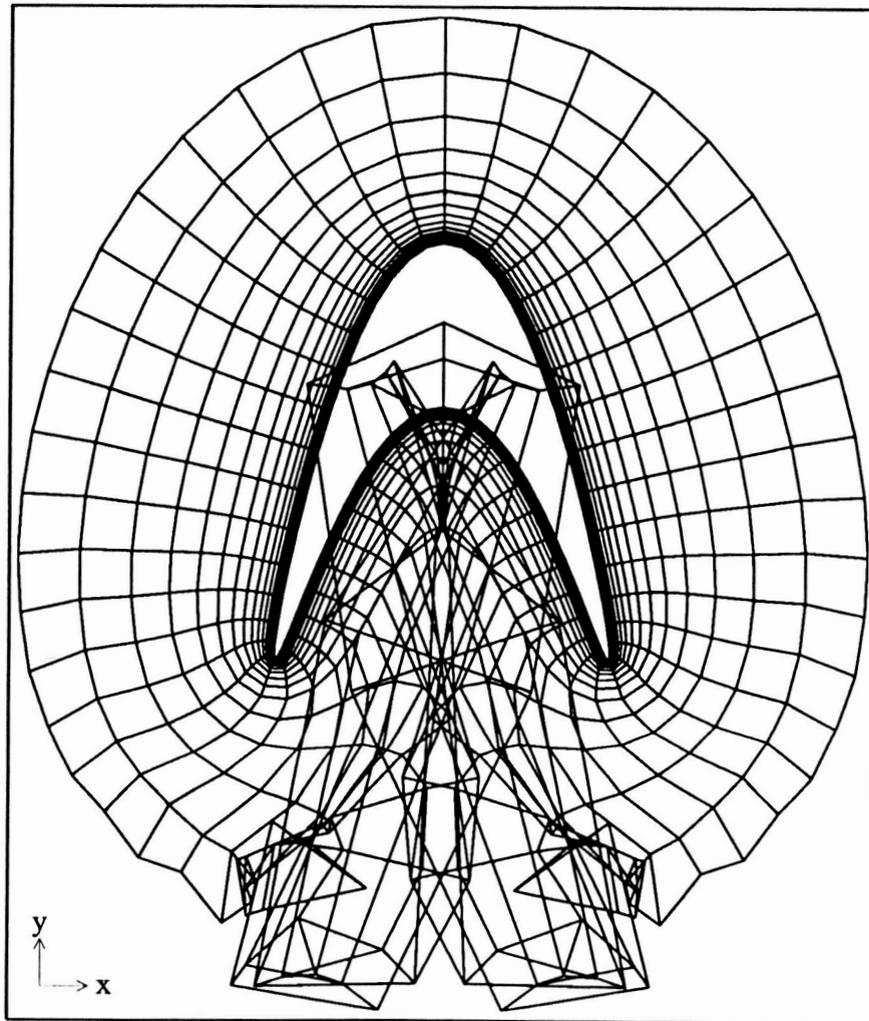


Figure 20: $\tau=0.5$, $\text{cam}=1.0$, initial grid
file: ehf510.00ps

Grid Improvement by Floating the outer Boundary

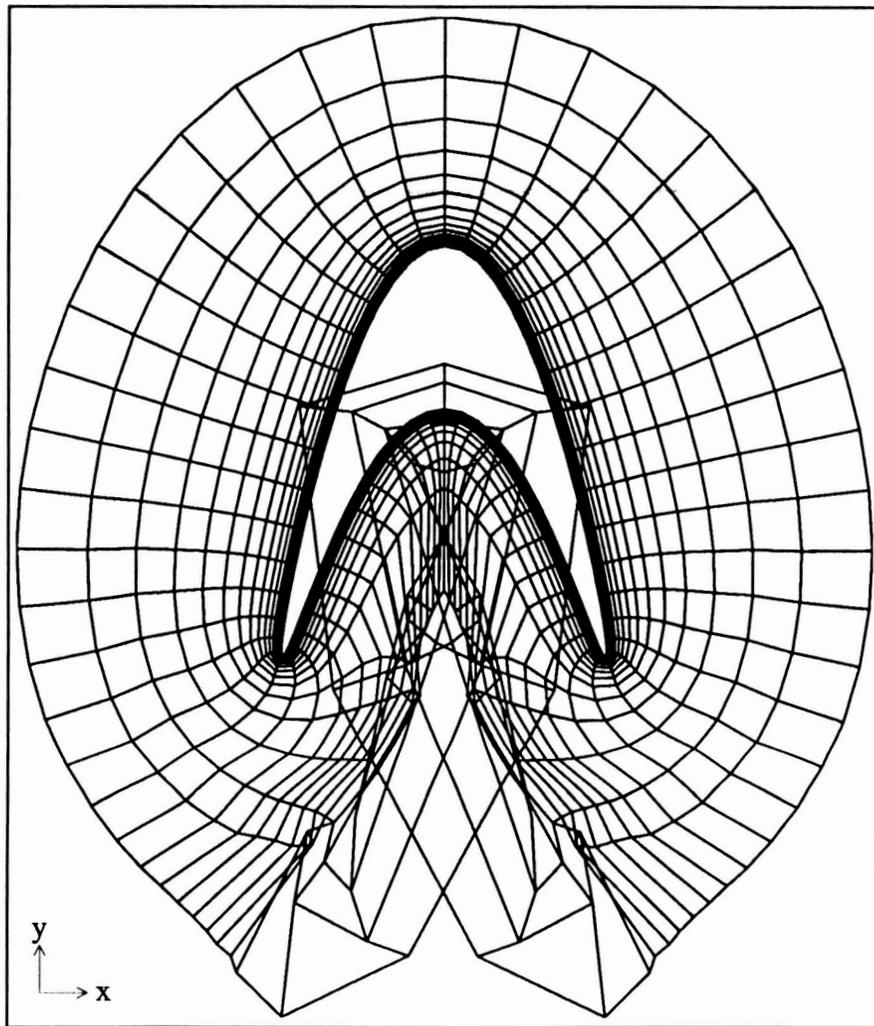


Figure 21: $\tau=0.5$, $\text{cam}=1.0$, pass number 1
file: ehf510.01ps

Grid Improvement by Floating the outer Boundary

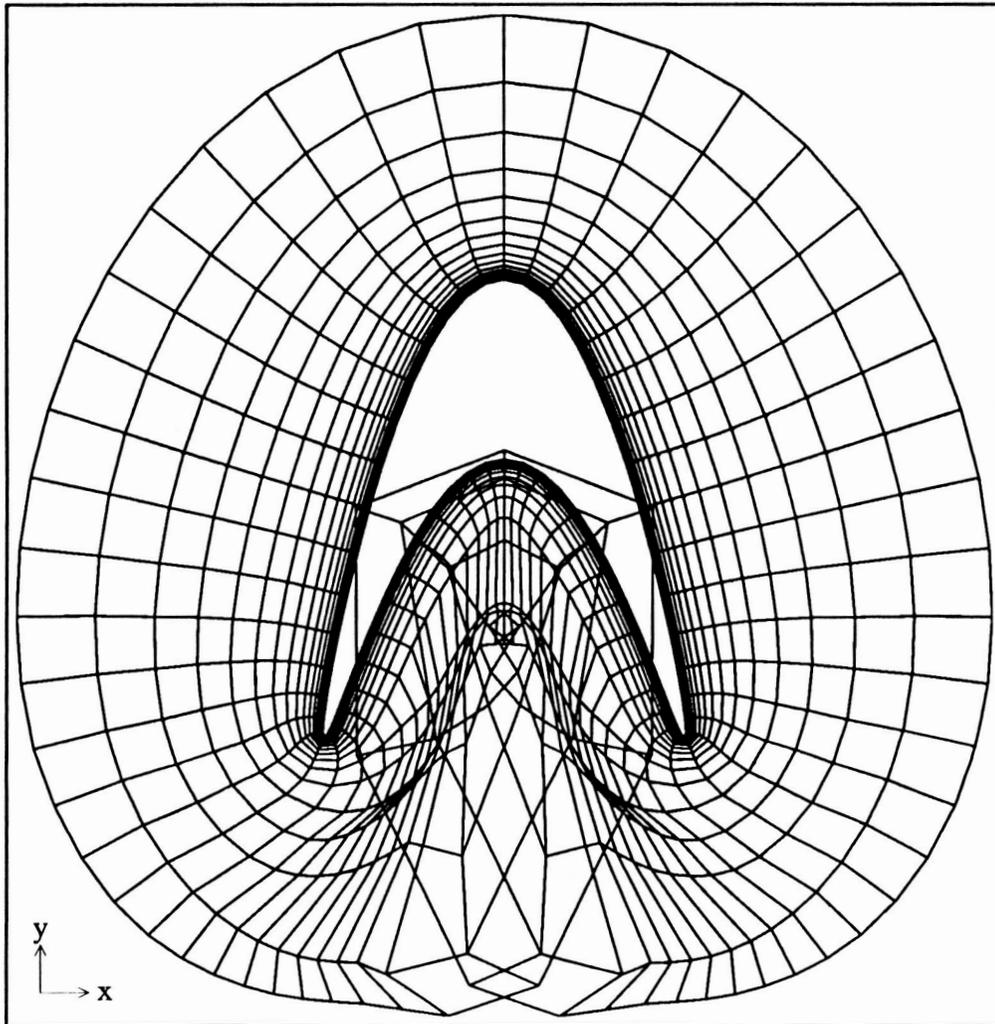


Figure 22: $\tau=0.5$, $\text{cam}=1.0$, pass number 2
file: ehf510.02ps

Grid Improvement by Floating the outer Boundary

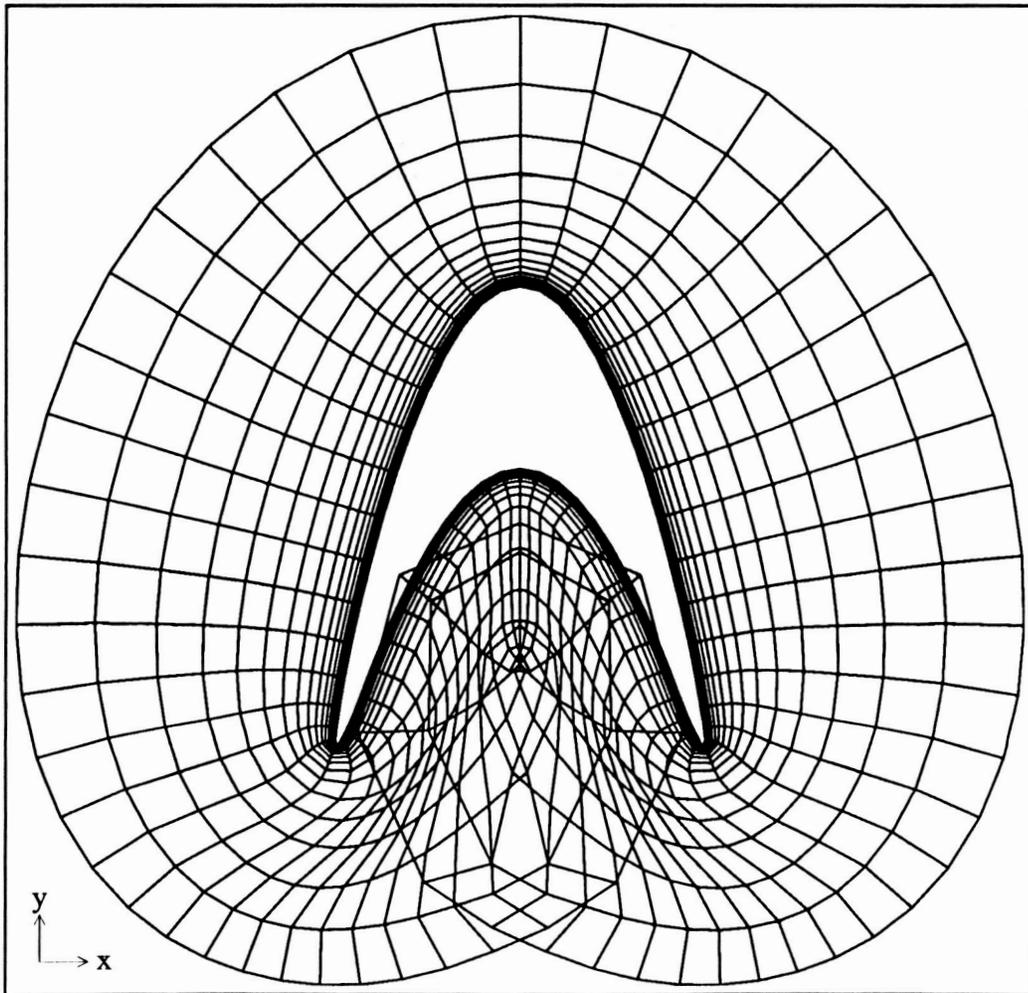


Figure 23: $\tau=0.5$, $\text{cam}=1.0$, pass number 3
file: ehf510.03ps

Grid Improvement by Floating the outer Boundary

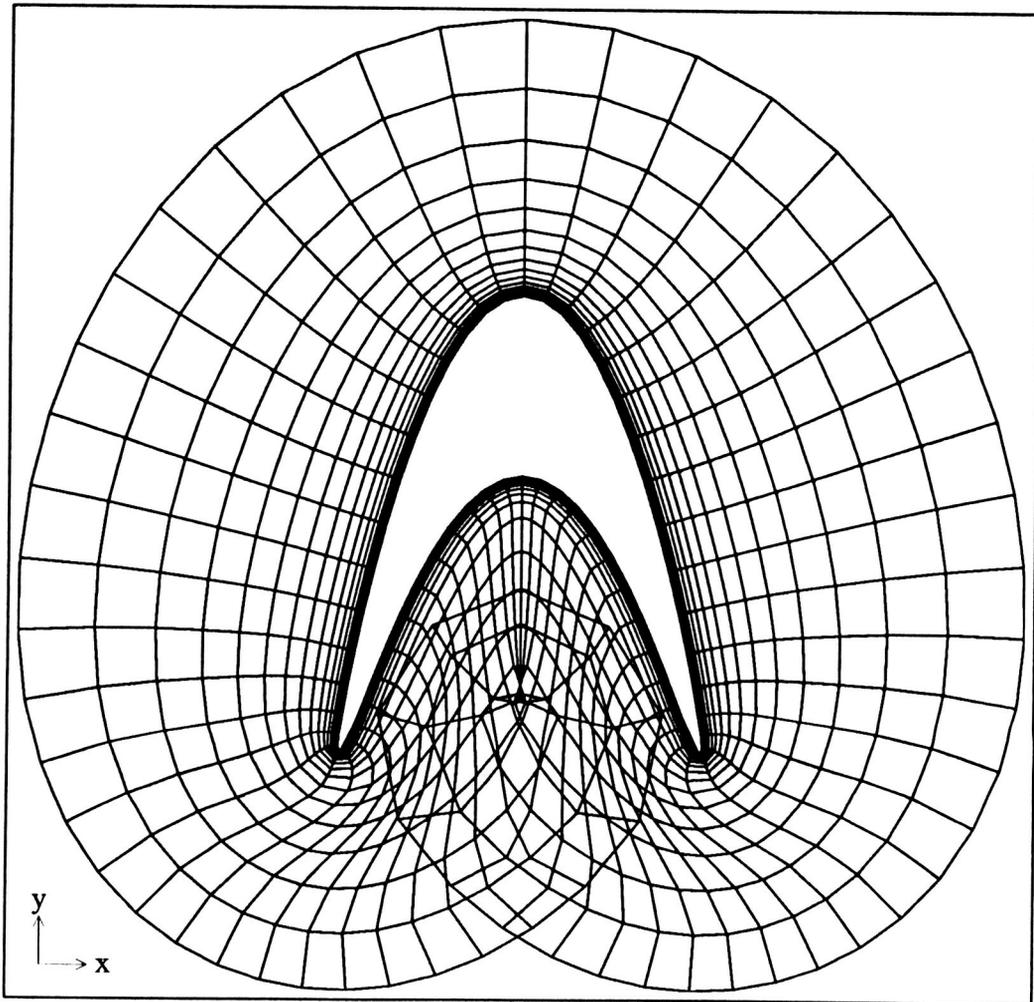


Figure 24: $\tau=0.5$, $\text{cam}=1.0$, pass number 4
file: ehf510.04ps

Grid Improvement by Floating the outer Boundary

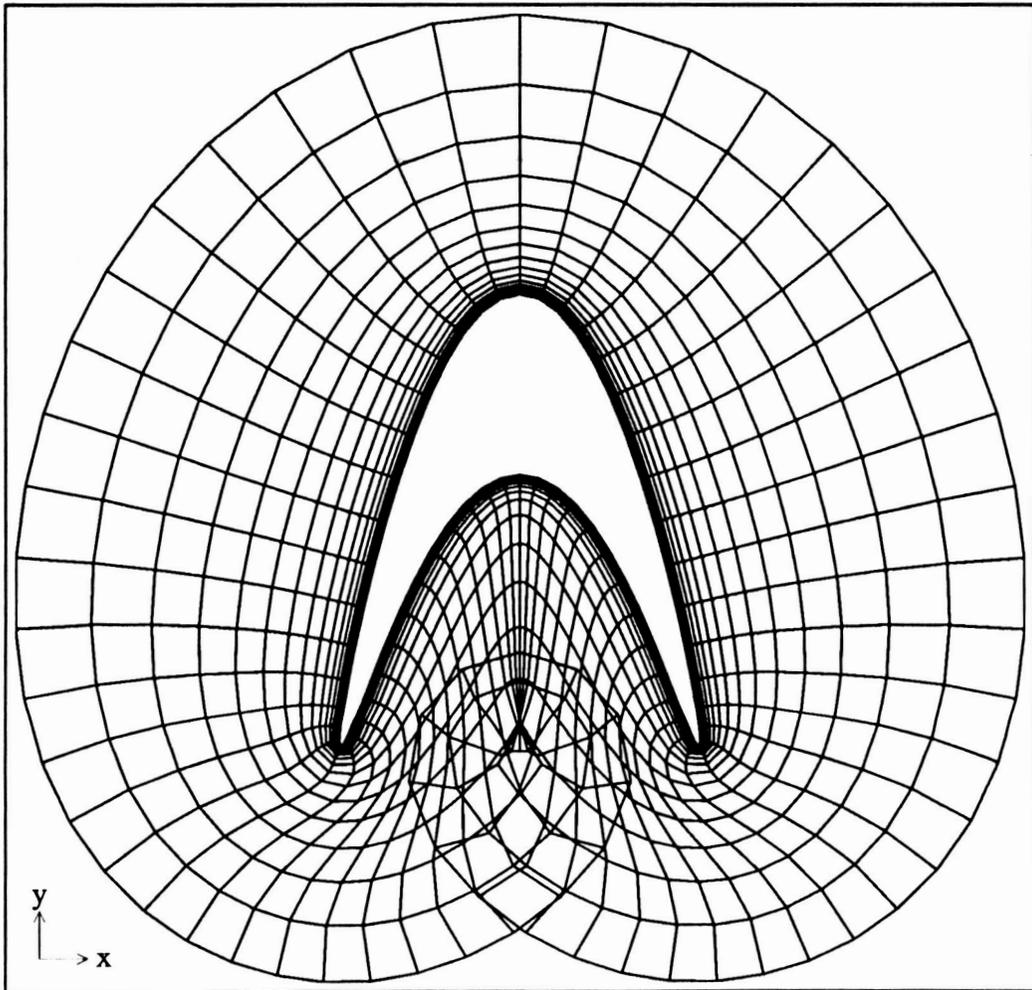


Figure 25: $\tau=0.5$, $\text{cam}=1.0$, pass number 5
file: ehf510.05ps

Grid Improvement by Floating the outer Boundary

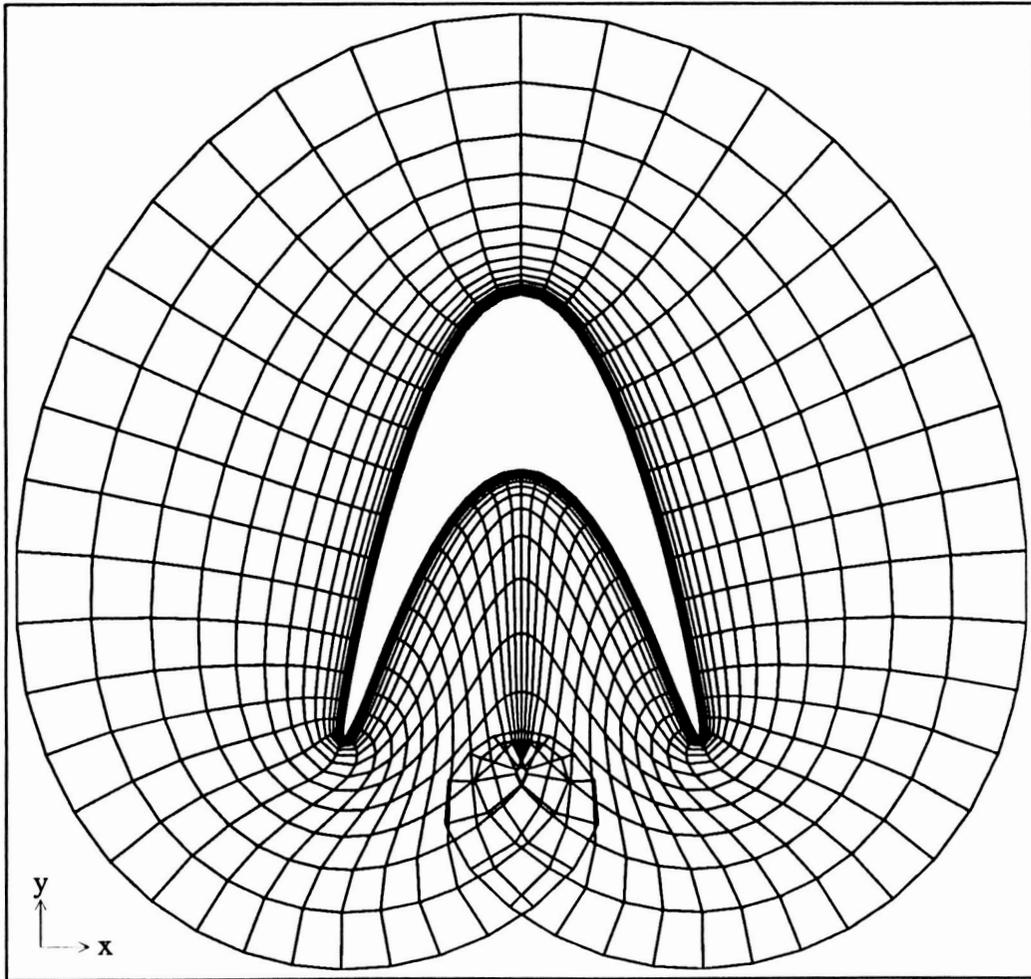


Figure 26: $\tau=0.5$, $\text{cam}=1.0$, pass number 6
file: ehf510.06ps

Grid Improvement by Floating the outer Boundary

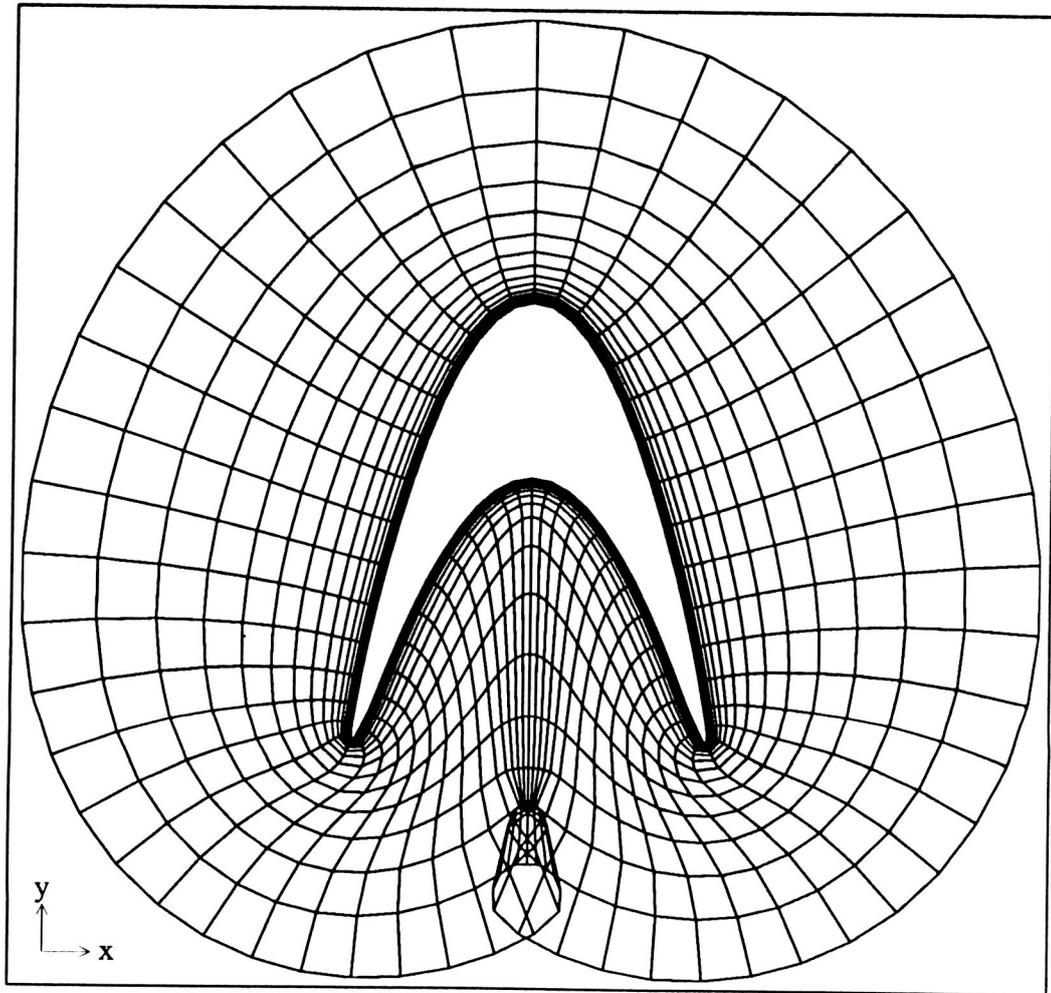


Figure 27: $\tau=0.5$, $\text{cam}=1.0$, pass number 7
file: ehf510.07ps

Grid Improvement by Floating the outer Boundary

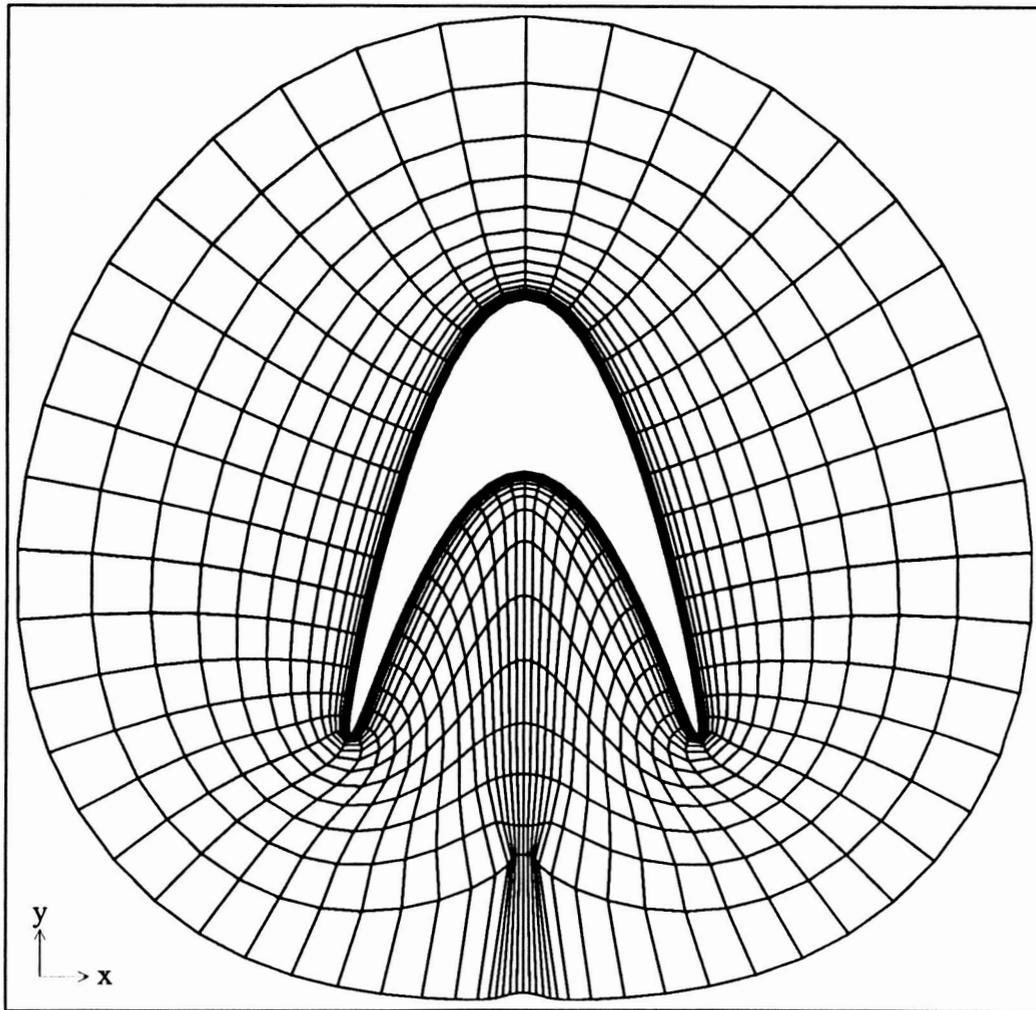


Figure 28: $\tau=0.5$, $\text{cam}=1.0$, pass number 8
file: ehf510.08ps

Grid Improvement by Floating the outer Boundary

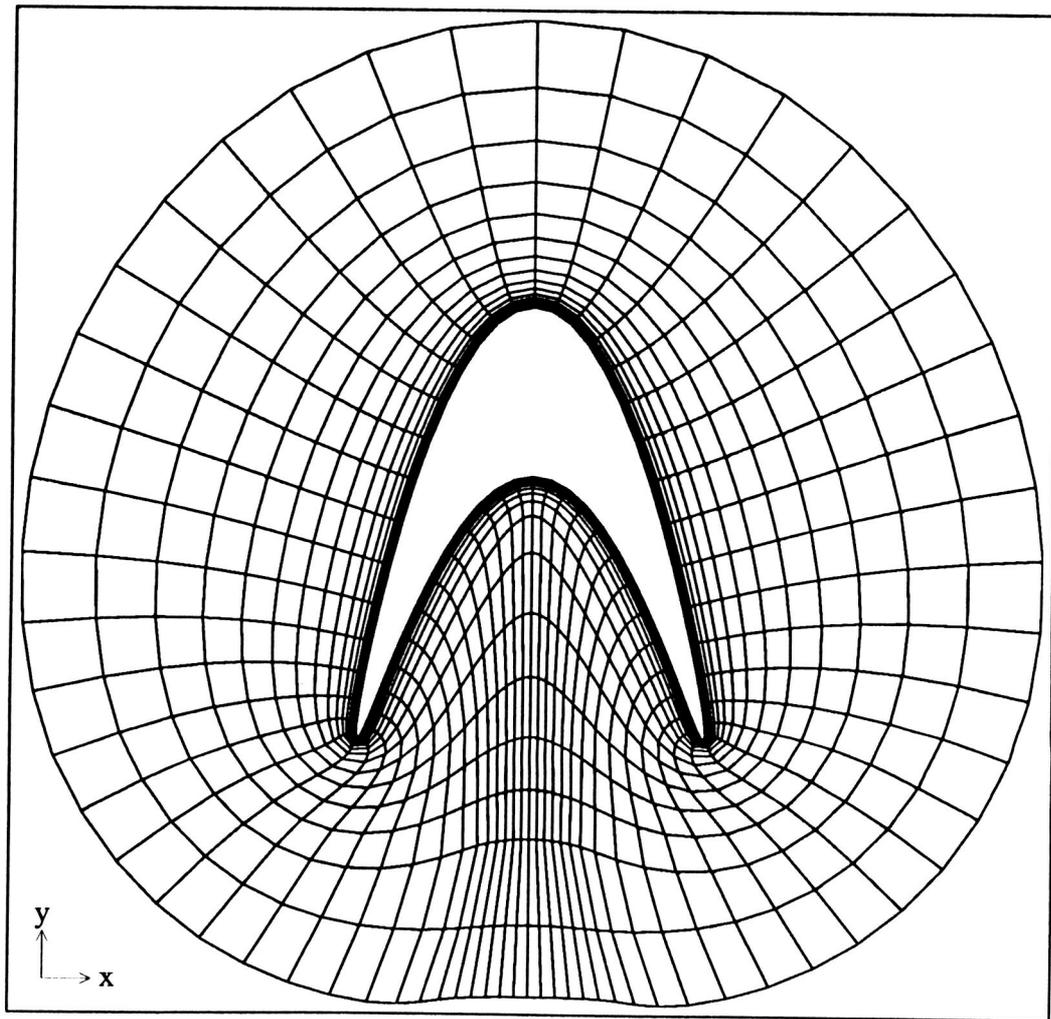


Figure 29: $\tau=0.5$, $\text{cam}=1.0$, pass number 9
file: ehf510.09ps

Grid Improvement by Floating the outer Boundary

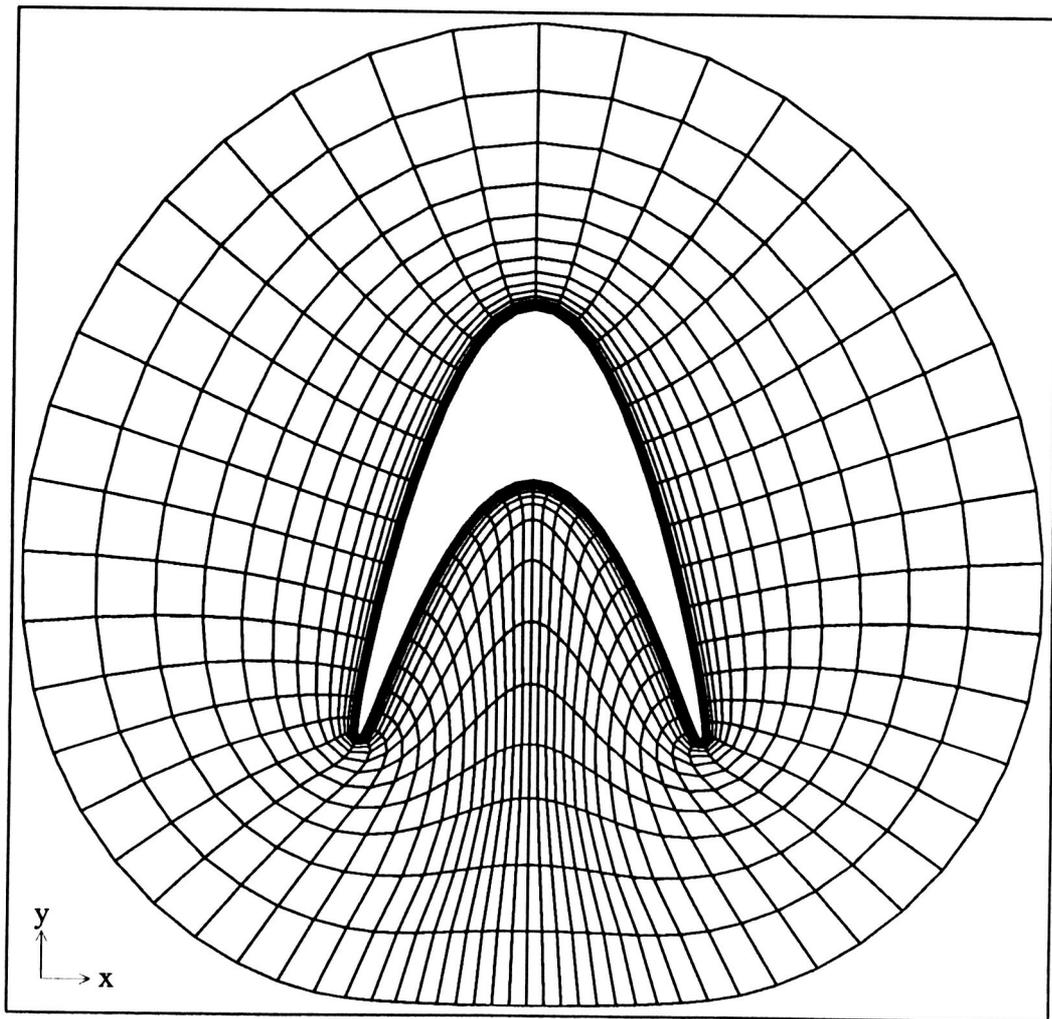


Figure 30: $\tau=0.5$, $\text{cam}=1.0$, pass number 10
file: ehf510.1ps

Grid Improvement by Floating the outer Boundary

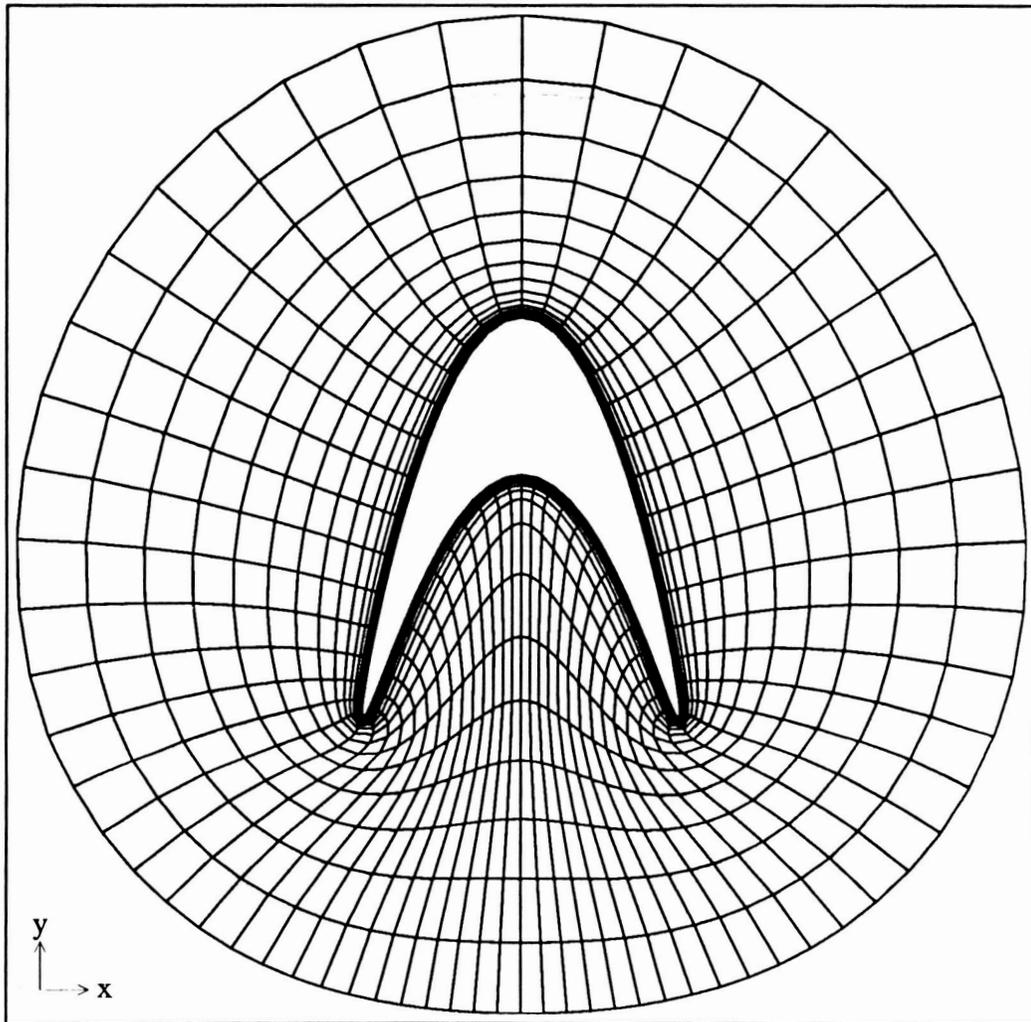


Figure 31: $\tau=0.5$, $\text{cam}=1.0$, pass number 20
file: ehf510.2ps

Grid Improvement by Floating the outer Boundary

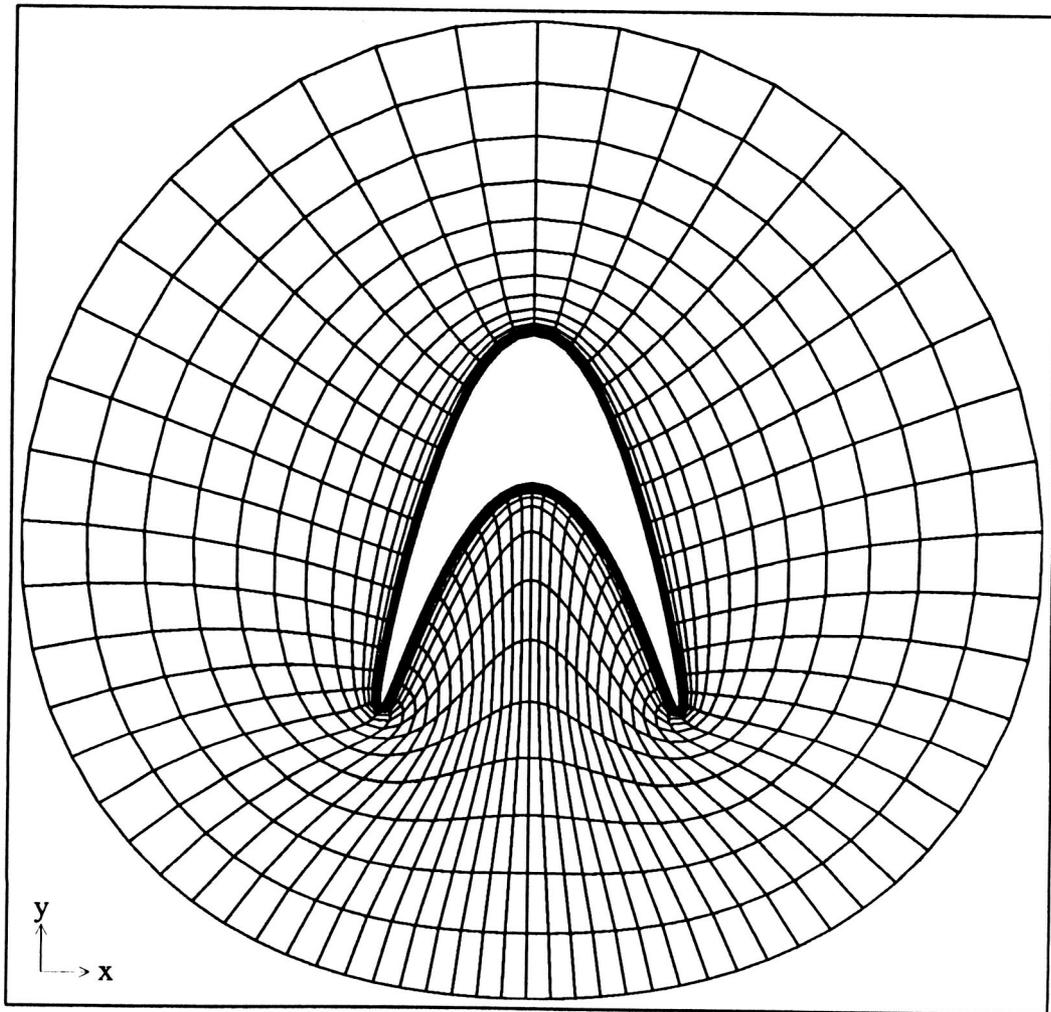


Figure 32: $\tau=0.5$, $\text{cam}=1.0$, pass number 50
file: ehf510.5ps

Grid Improvement by Floating the outer Boundary

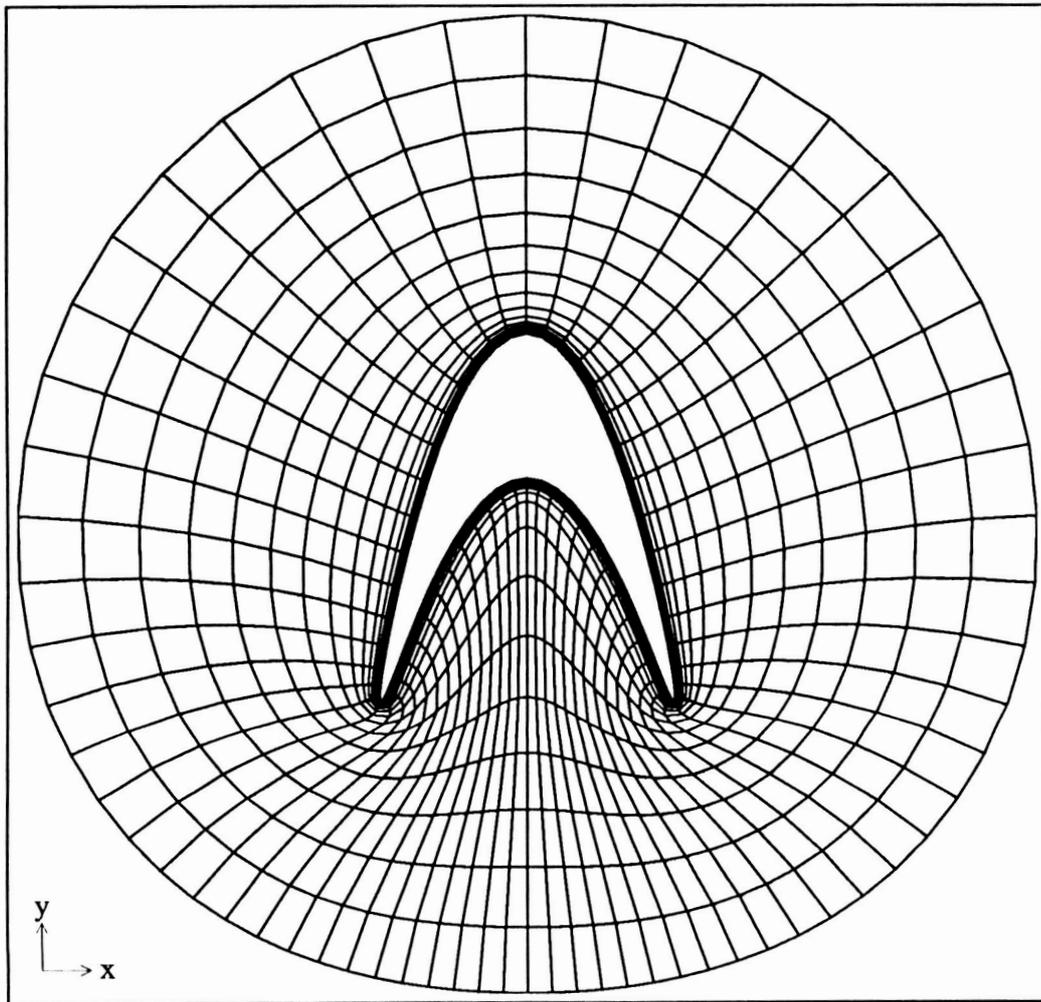


Figure 33: $\tau=0.5$, $\text{cam}=1.0$, pass number 100
file: ehf510.10ps

Initial Grid with Compressed Cells

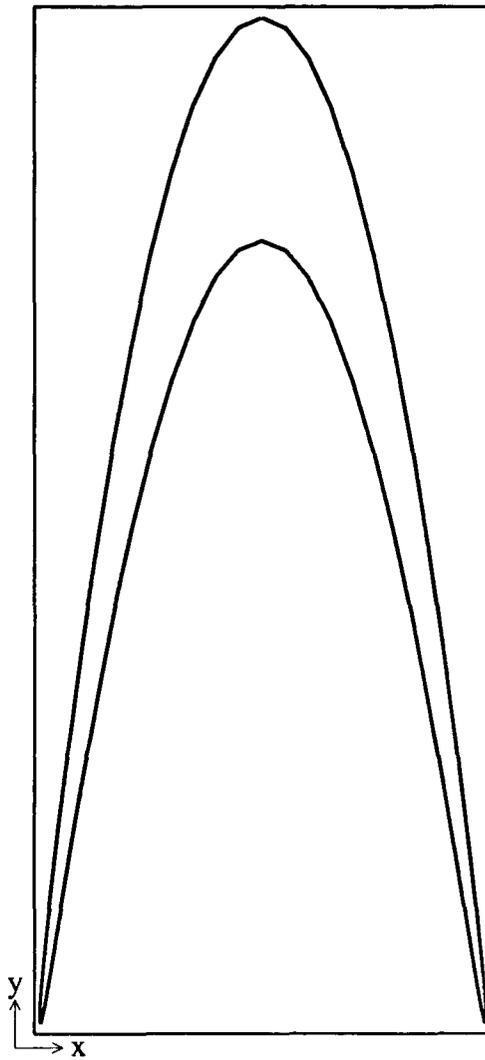


Figure 34: $\tau=0.5$, $\text{cam}=2.0$, full figure
file: hg5y20.ps, initial $d\text{seta}=1\text{e-}5$, final $d\text{seta}=1\text{e-}3$

Initial Grid with Compressed Cells

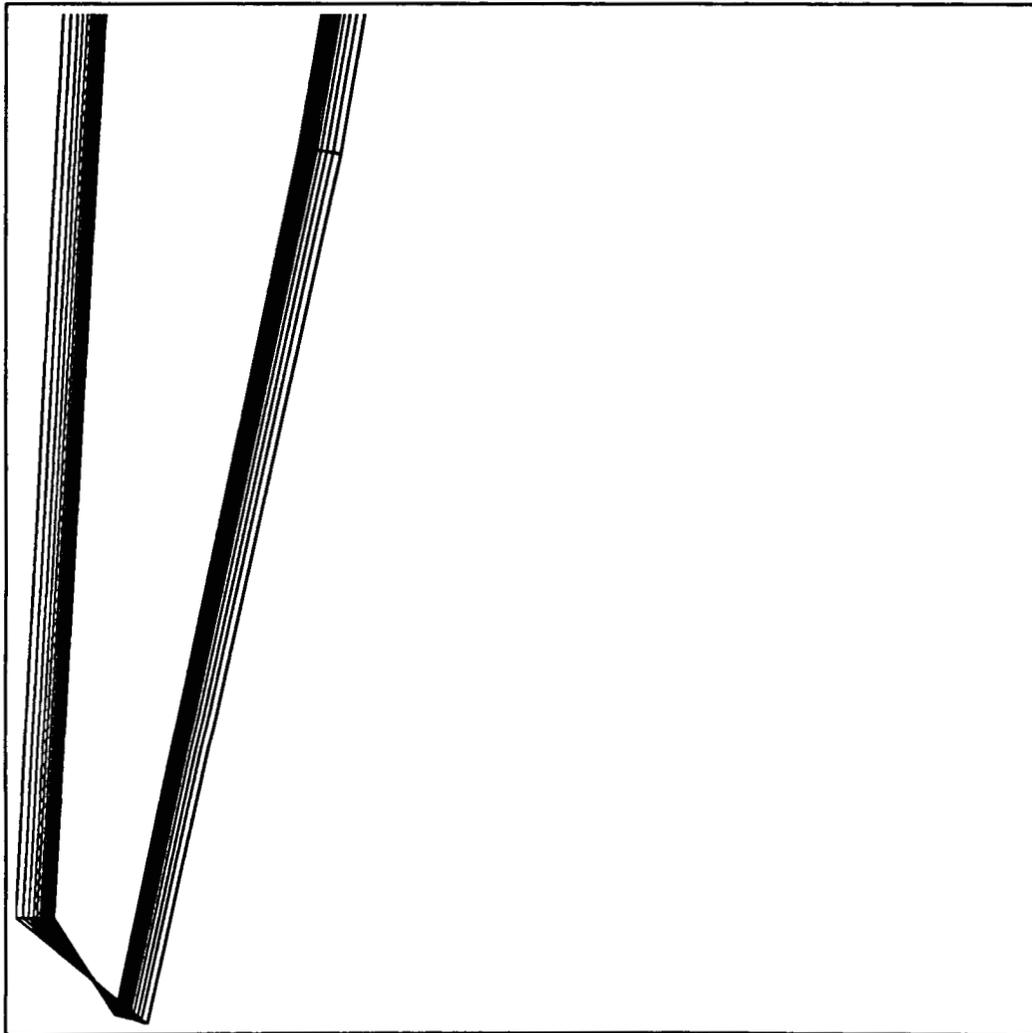


Figure 35: $\tau=0.5$, $\text{cam}=2.0$, detail of initial grid near convex point
file: hg5y20.ptps initial dseta= $1e-5$ final dseta= $1e-3$

Cell Volume Growth Coupled with Floating Boundary

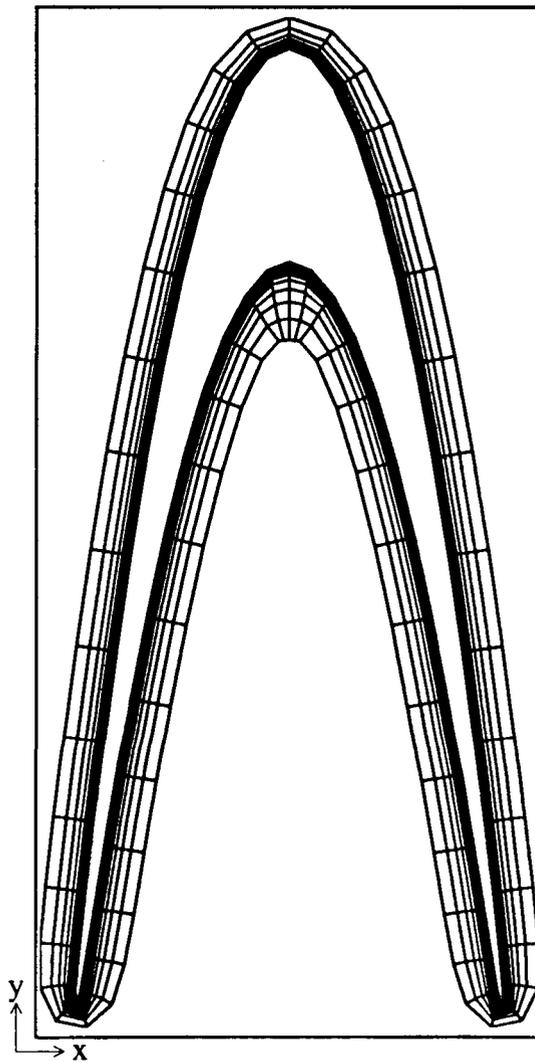


Figure 36: $\tau=0.5$, $\text{cam}=2.0$, 40th pass
file: fg520.4elxps

Cell Volume Growth Coupled with Floating Boundary

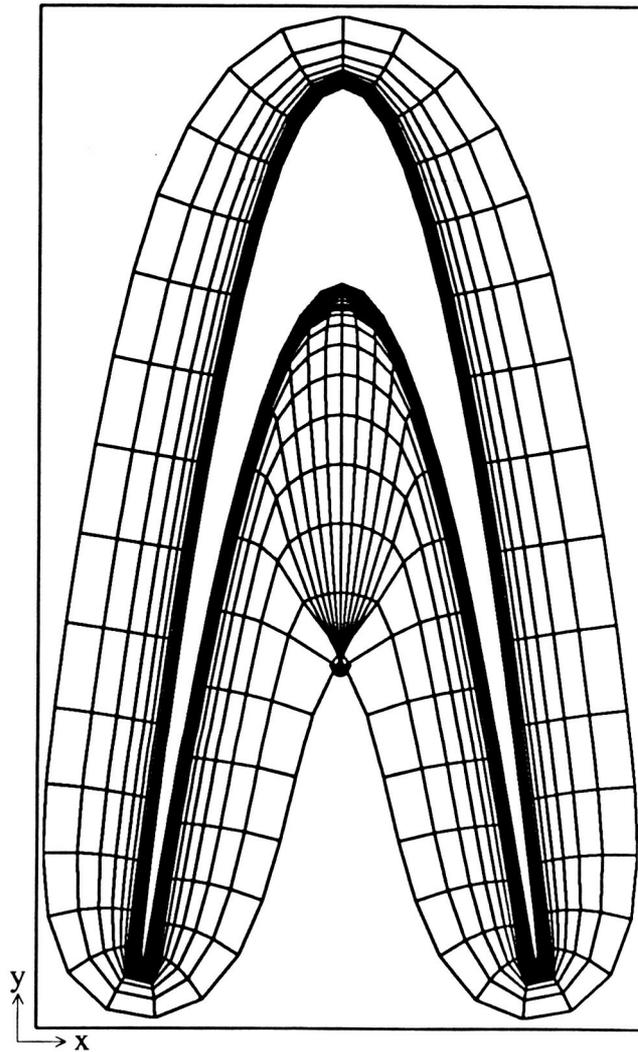


Figure 37: $\tau=0.5$, $\text{cam}=2.0$, 50th pass
file: hg520.5elxps

Cell Volume Growth Coupled with Floating Boundary

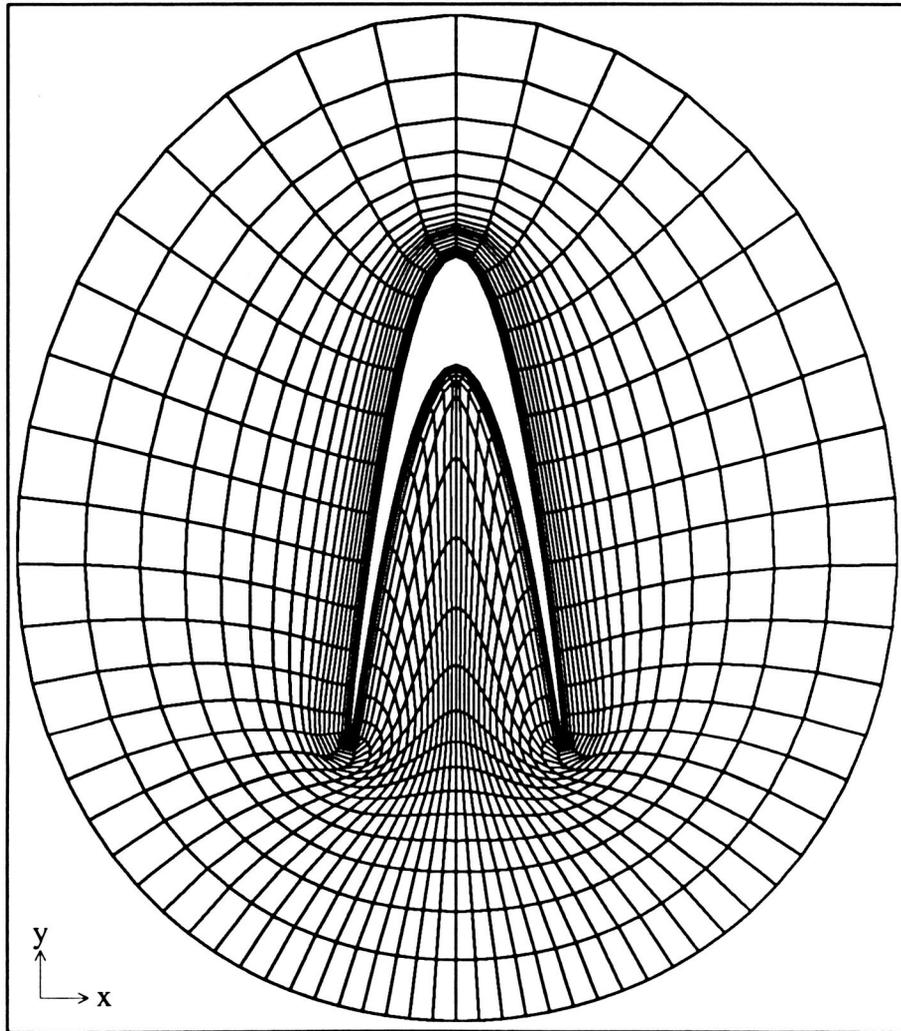


Figure 38: $\tau=0.5$, $\text{cam}=2.0$, 80th pass
file: fg520.8elxps

Cell Volume Growth Coupled with Floating Boundary

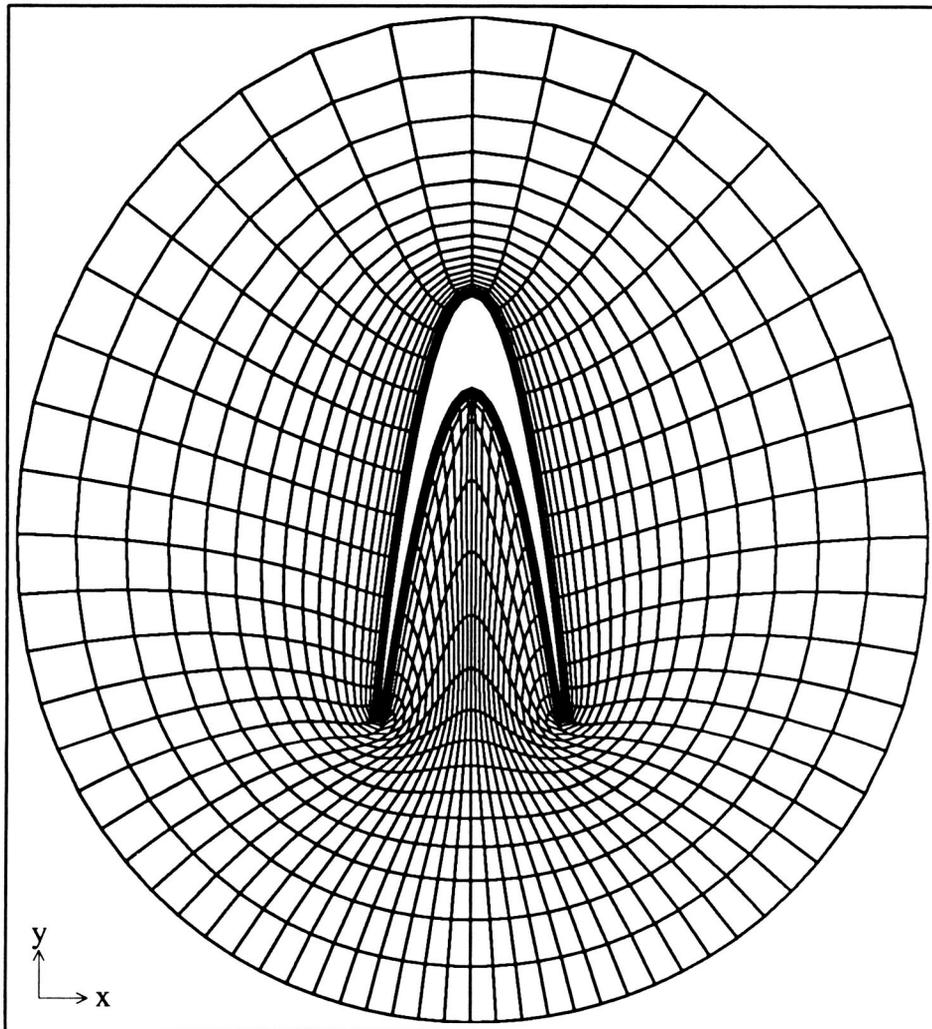


Figure 39: $\tau=0.5$, $\text{cam}=2.0$, 100th pass
file: fg520.10elxps

Cell Volume Growth Coupled with Floating Boundary

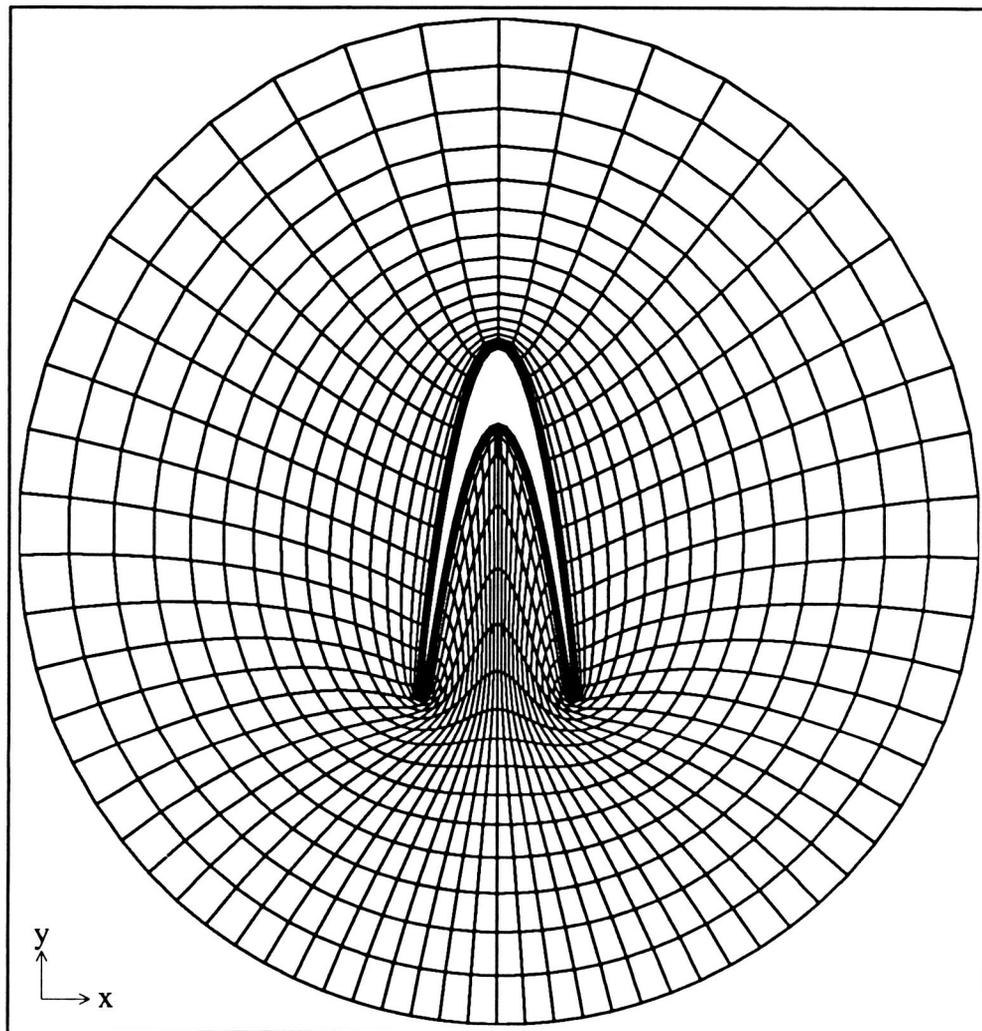


Figure 40: $\tau=0.5$, $\text{cam}=2.0$, 200th pass
file: fg520.20elxps

Cell Volume Growth Coupled with Floating Boundary

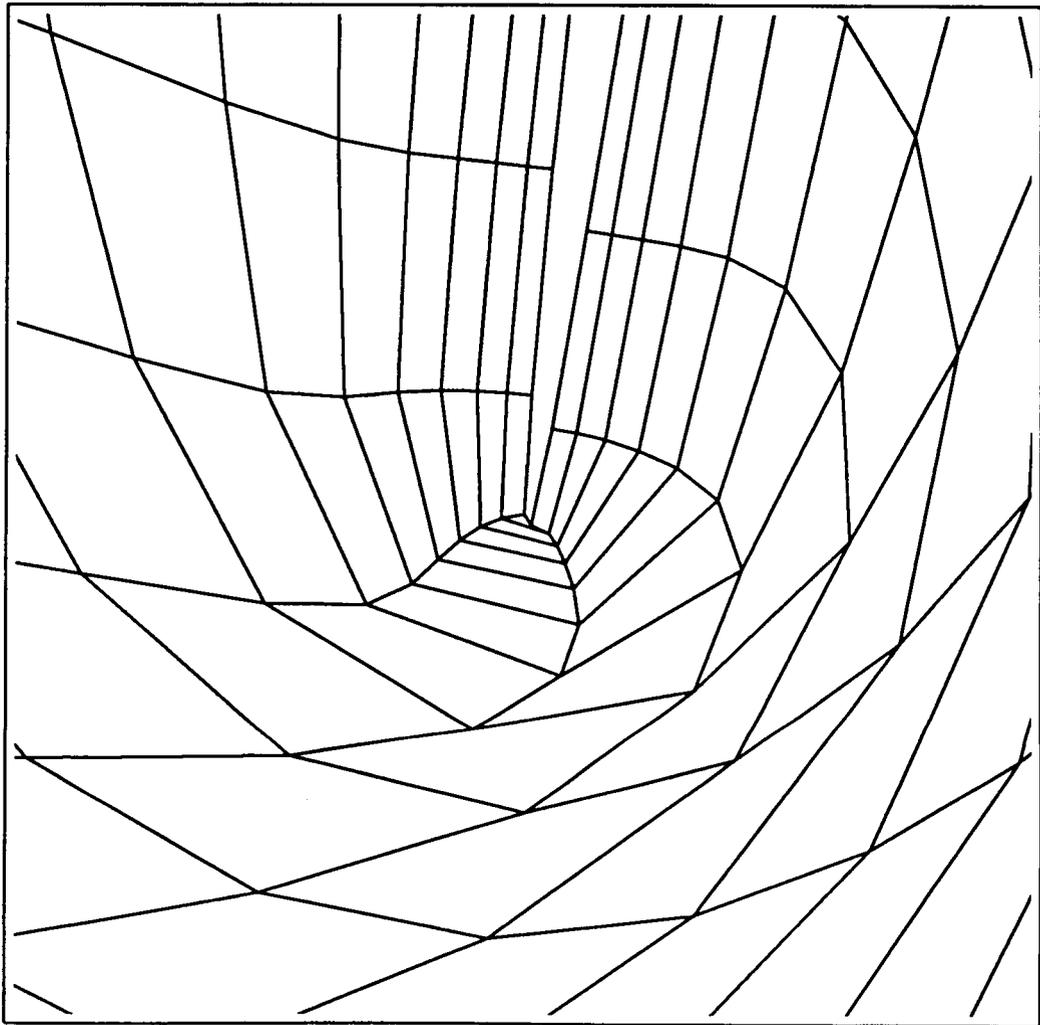


Figure 41: $\tau=0.5$, $\text{cam}=2.0$, 200th pass, detail near convex area
file: fg520.20elxpts

Initial Grid with Compressed Cells

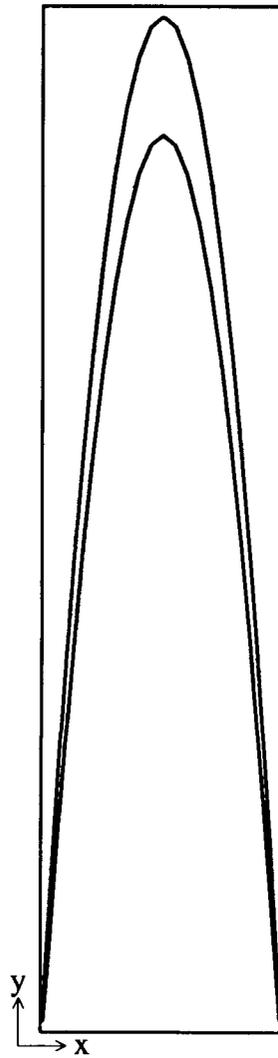


Figure 42: $\tau=0.5$, $\text{cam}=4.0$, Full Figure
file: hg5y40.ps initial $d\text{seta}=1e-5$ final $d\text{seta}=1e-3$

ORIGINAL PAGE IS
OF POOR QUALITY

Cell Volume Growth Coupled with Floating Boundary

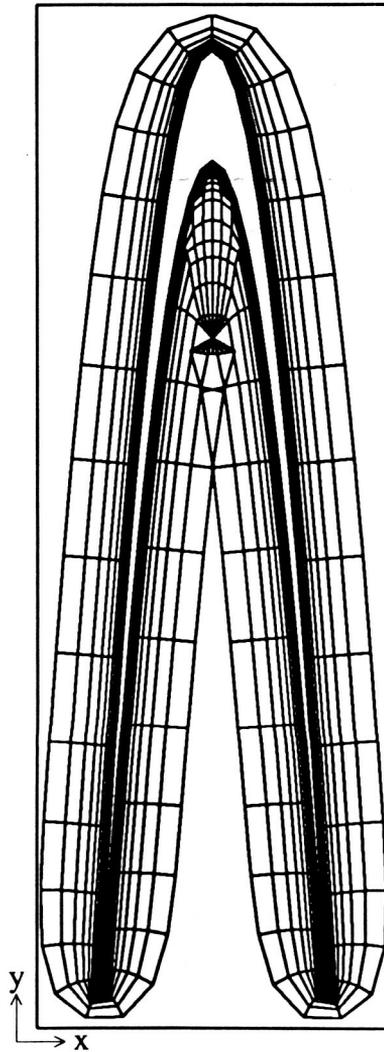


Figure 43: $\tau=0.5$, $\text{cam}=4.0$, 50th pass
file: fg540.5ps

Cell Volume Growth Coupled with Floating Boundary

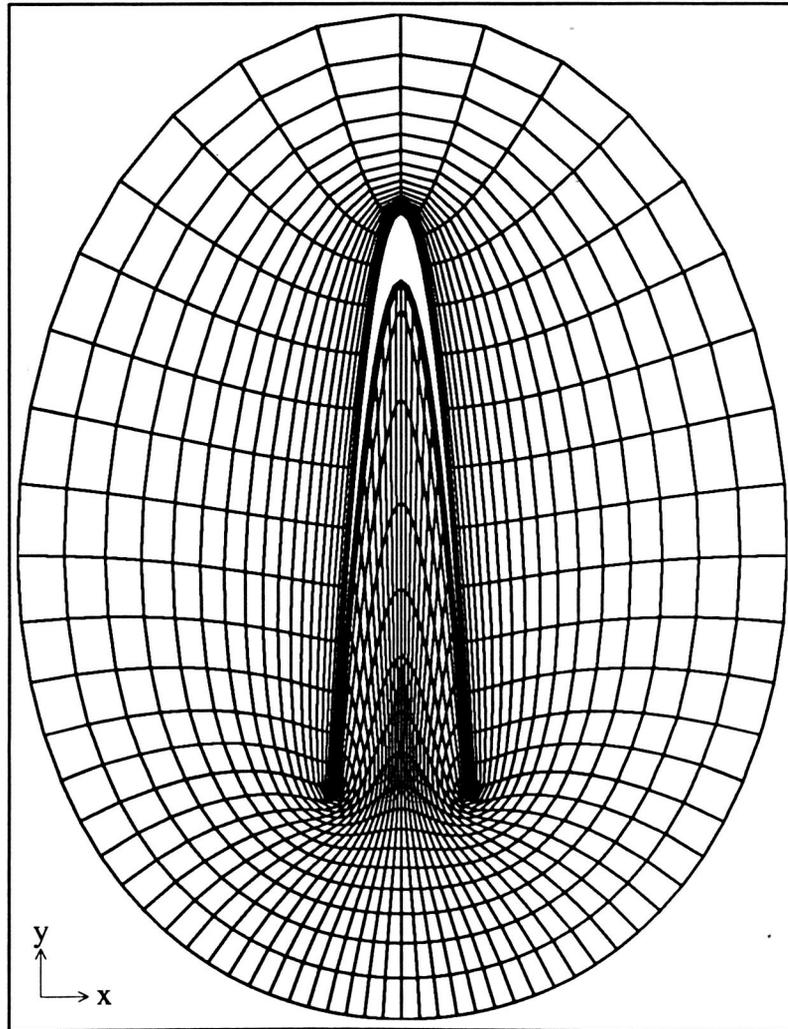


Figure 44: $\tau=0.5$, $\text{cam}=4.0$, 100th pass
file: fg540.10ps

Cell Volume Growth Coupled with Floating Boundary

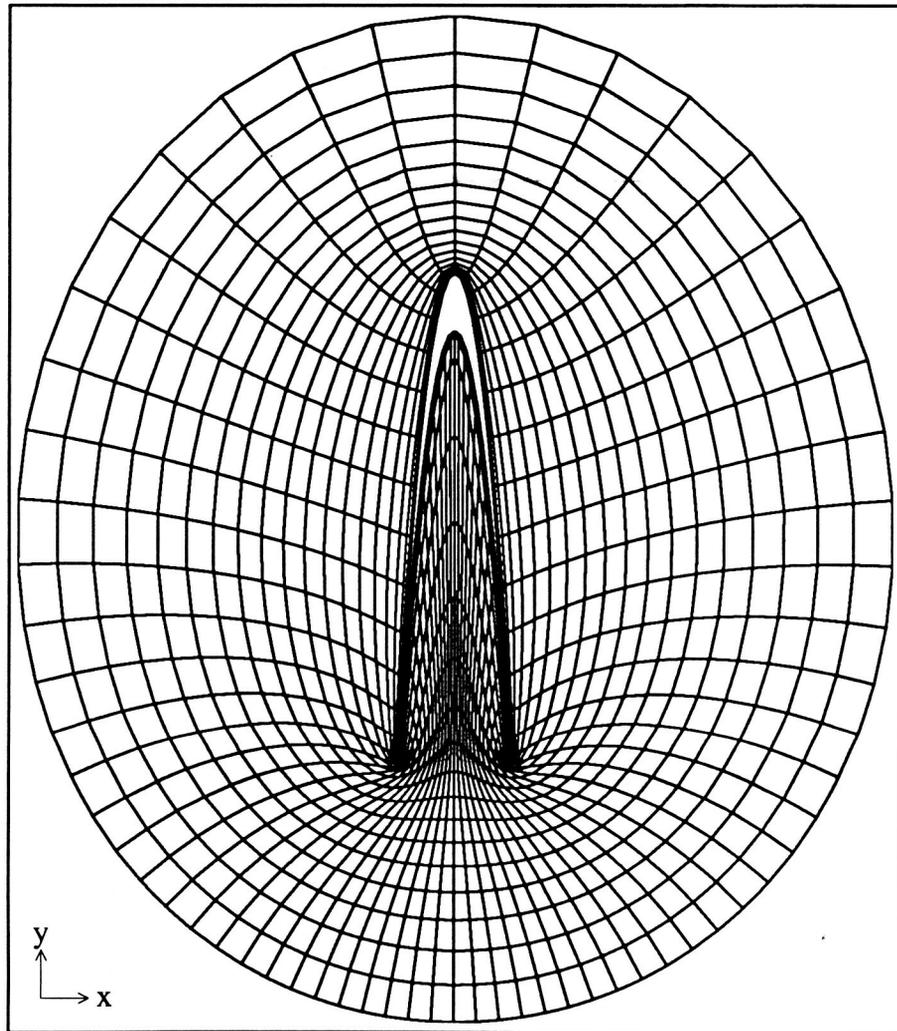


Figure 45: $\tau=0.5$, $\text{cam}=4.0$, 200th pass
file: fg540.20ps

ORIGINAL PAGE IS
OF POOR QUALITY

Cell Volume Growth Coupled with Floating Boundary Failure of Technique Due to Depth of Cavity

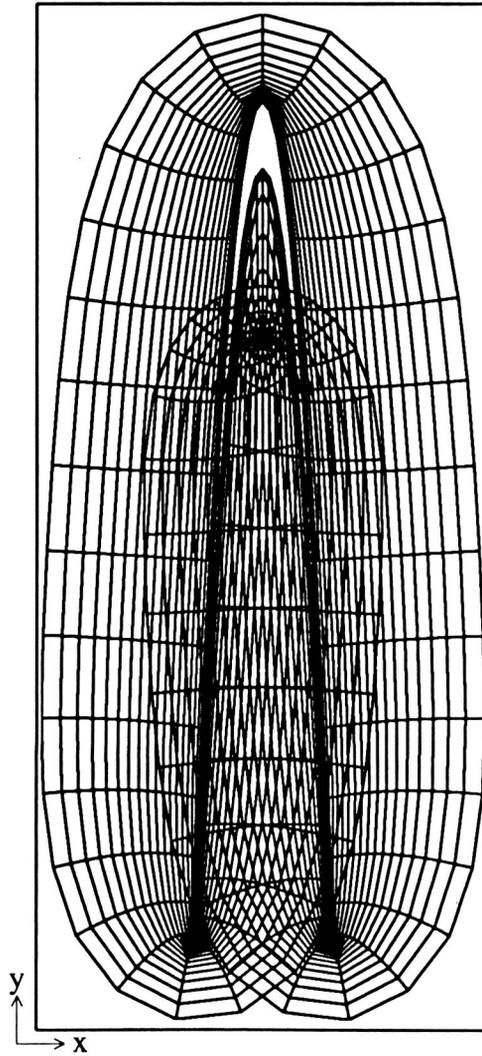


Figure 46: $\tau=0.5$, $\text{cam}=6.0$, 500th pass
file: fg560.50ps

Forming a Cut In a Cavity
Initial Grid

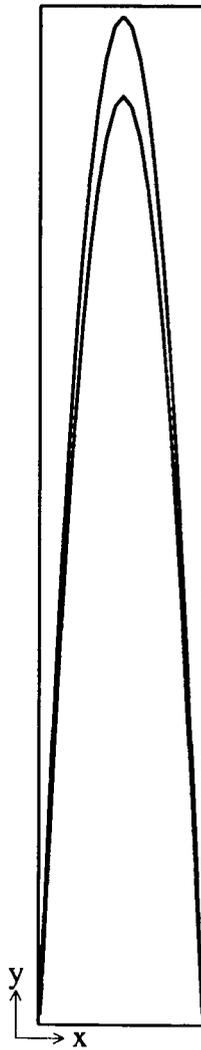


Figure 47: $\tau=0.5$, $\text{cam}=6.0$, initial grid
file: fgc560.00ps

Forming a Cut In a Cavity

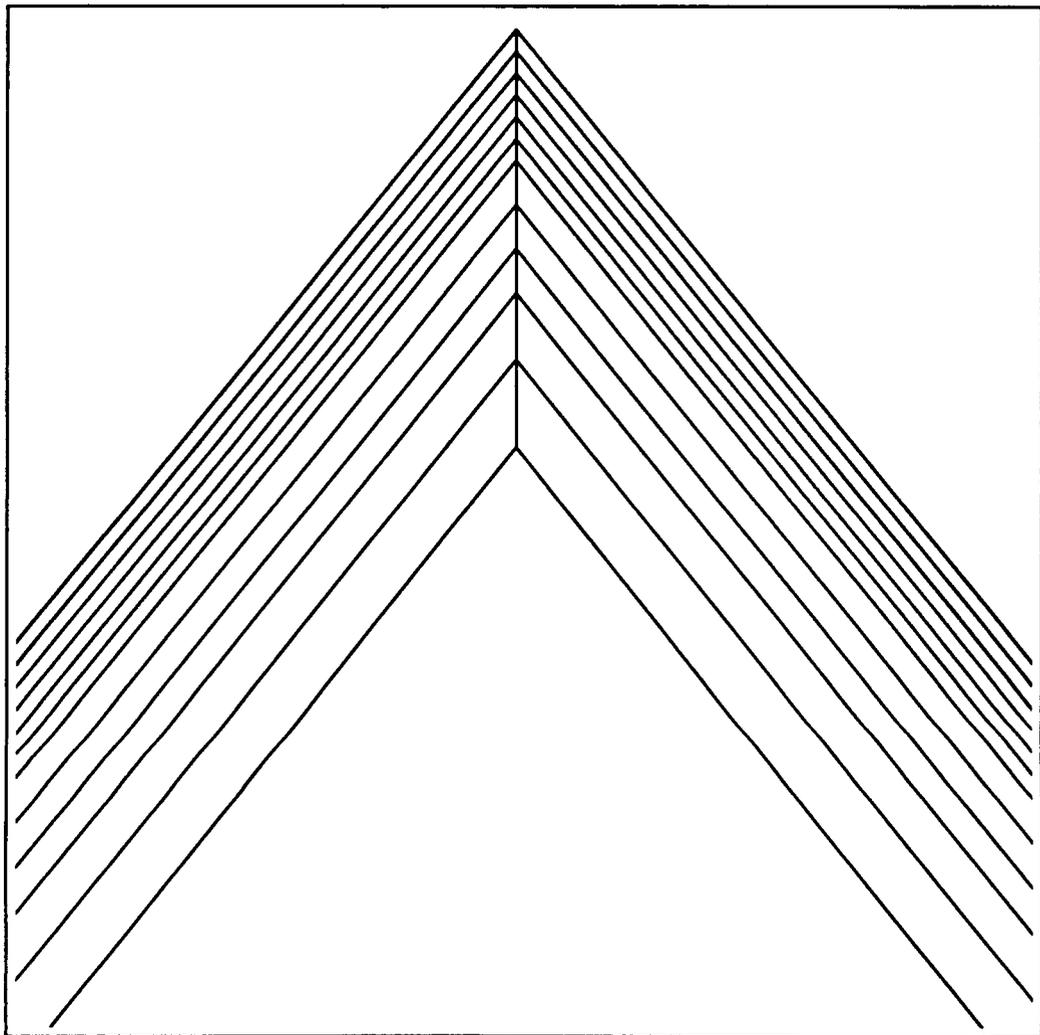


Figure 48: $\tau=0.5$, $\text{cam}=6.0$, initial grid, detail of cavity
file: fgc560.00pts

ORIGINAL PAGE IS
OF POOR QUALITY

Forming a Cut In a Cavity

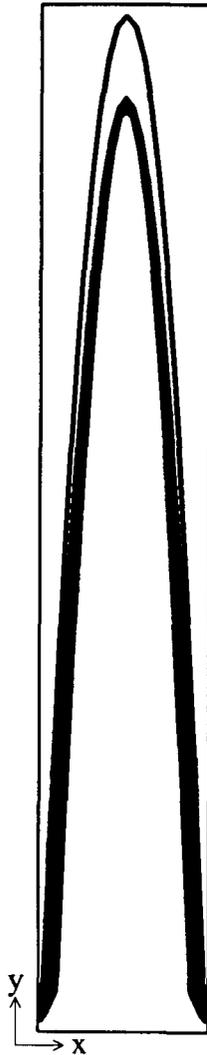


Figure 49: $\tau=0.5$, $\text{cam}=6.0$, pass 20
file: fgc560.2ps

Forming a Cut In a Cavity

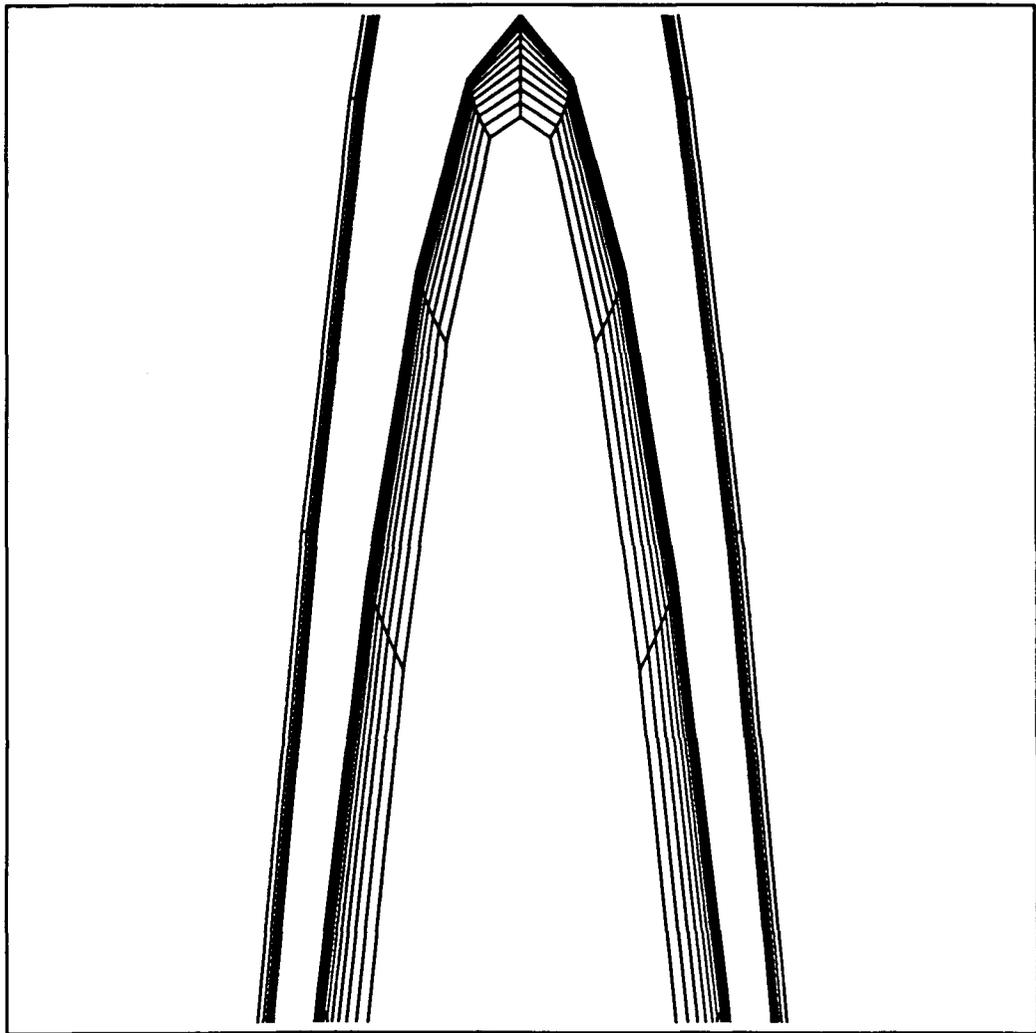


Figure 50: $\tau=0.5$, $\text{cam}=6.0$, pass 20, detail of cavity
file: fgc560.2ptps

ORIGINAL PAGE IS
OF POOR QUALITY

Forming a Cut In a Cavity

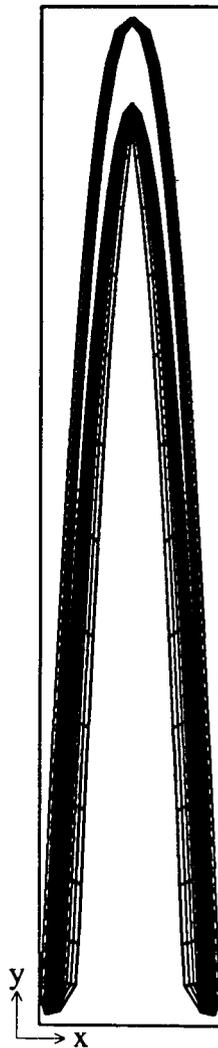


Figure 51: $\tau=0.5$, $\text{cam}=6.0$, pass 30
file: fgc560.3ps

Forming a Cut In a Cavity

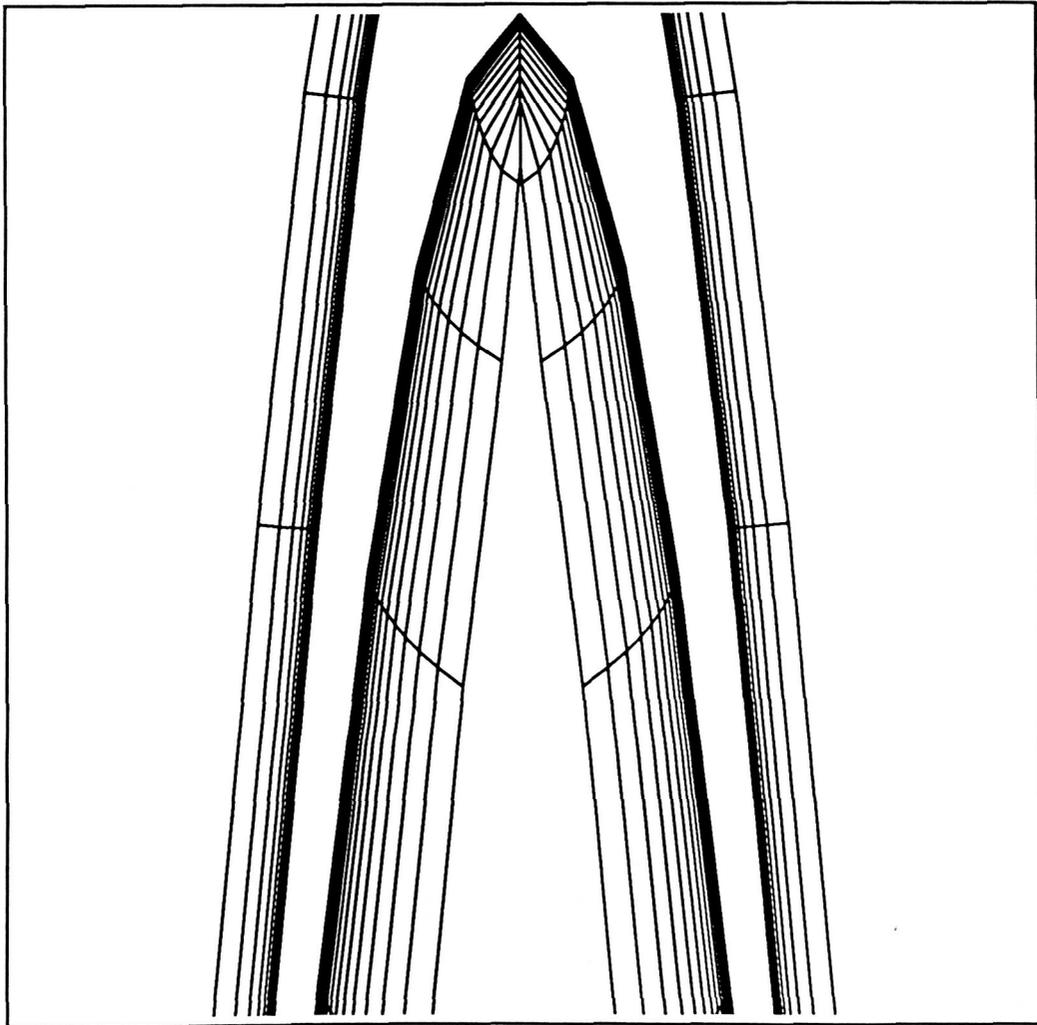


Figure 52: $\tau=0.5$, $\text{cam}=6.0$, pass 30 , detail of cavity
file: fgc560.3ptps

Forming a Cut In a Cavity

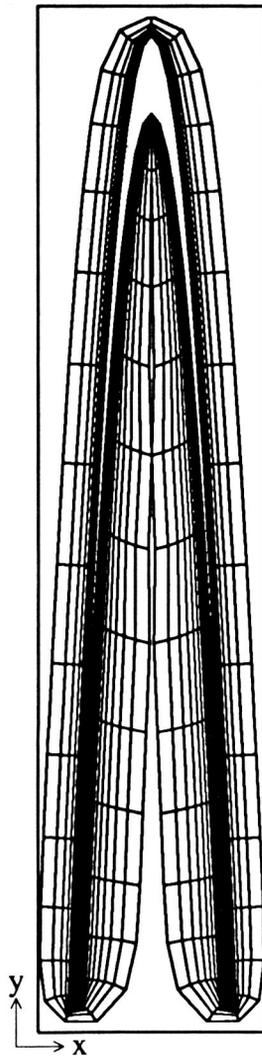


Figure 53: $\tau=0.5$, $\text{cam}=6.0$, pass 40
file: fgc560.4ps

Forming a Cut In a Cavity

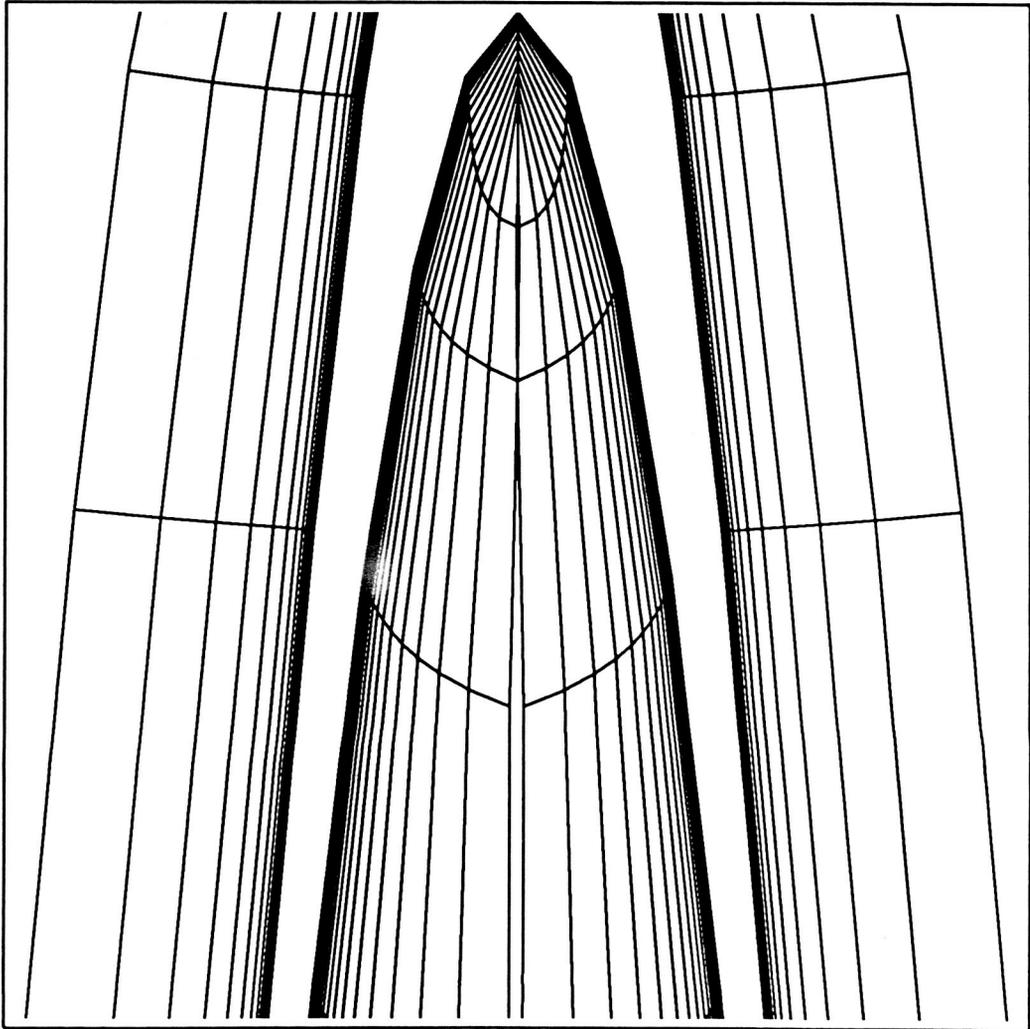


Figure 54: $\tau=0.5$, $\text{cam}=6.0$, pass 40, detail of cavity
file: fgc560.4ptps

Forming a Cut In a Cavity

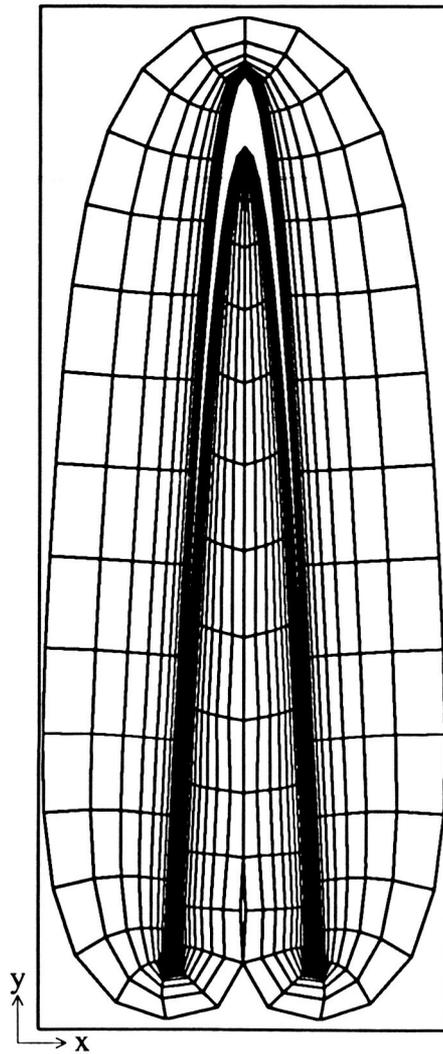


Figure 55: $\tau=0.5$, $\text{cam}=6.0$, pass 50
file: fgc560.5ps

Forming a Cut In a Cavity

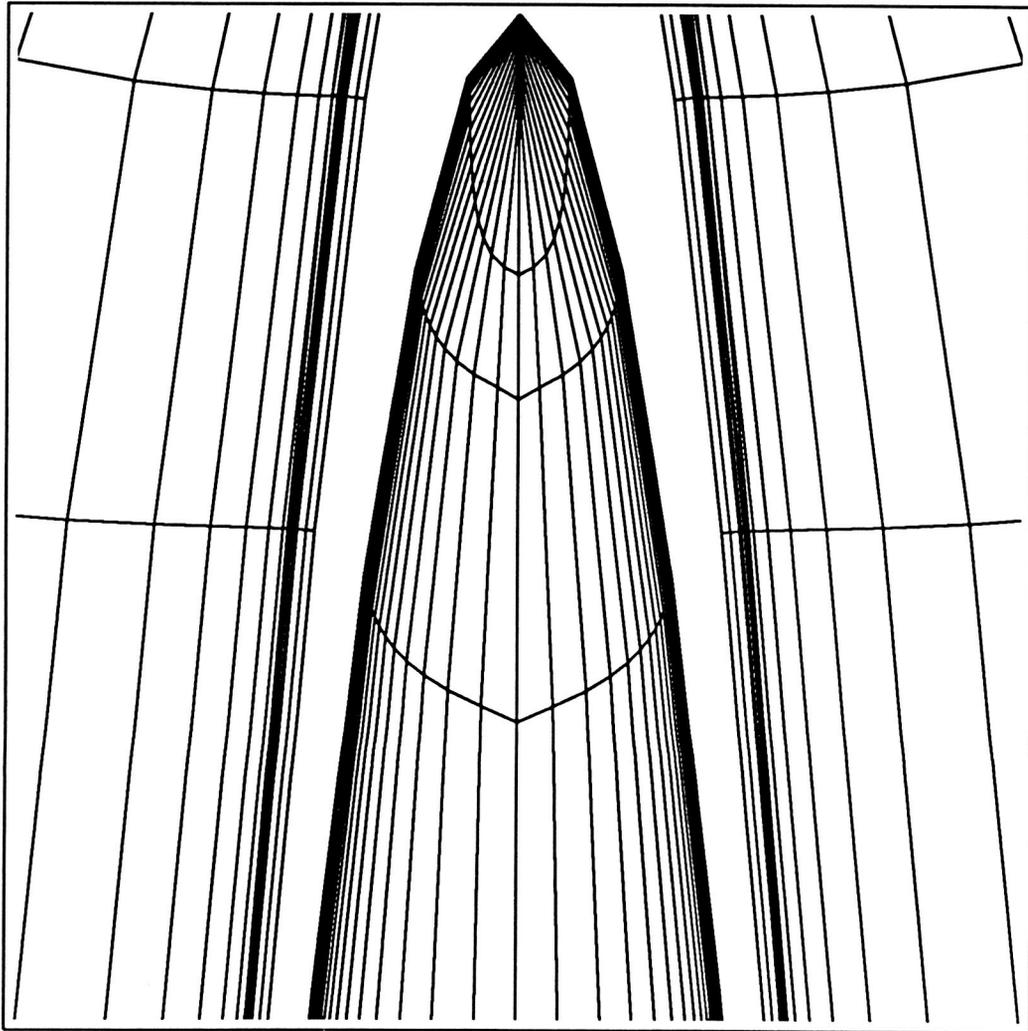


Figure 56: $\tau=0.5$, $\text{cam}=6.0$, pass 50, detail of cavity
file: fgc560.5ptps

Forming a Cut In a Cavity

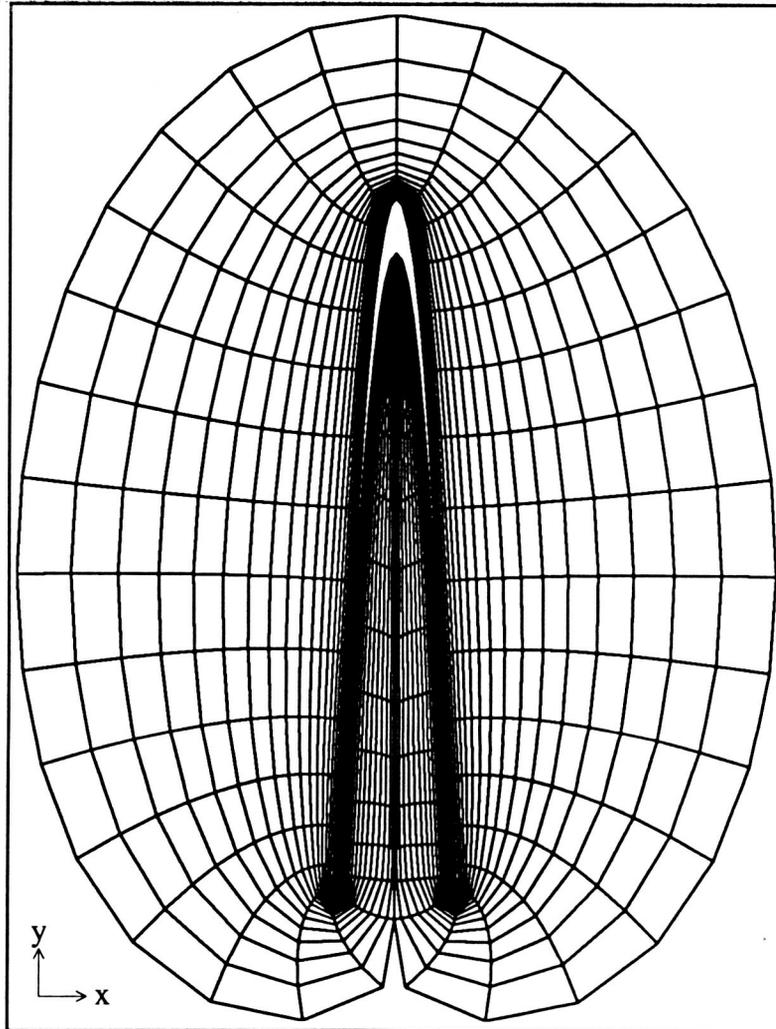


Figure 57: $\tau=.5$, $\text{cam}=6.0$, pass 80
file: fgc560.8ps

Forming a Cut In a Cavity

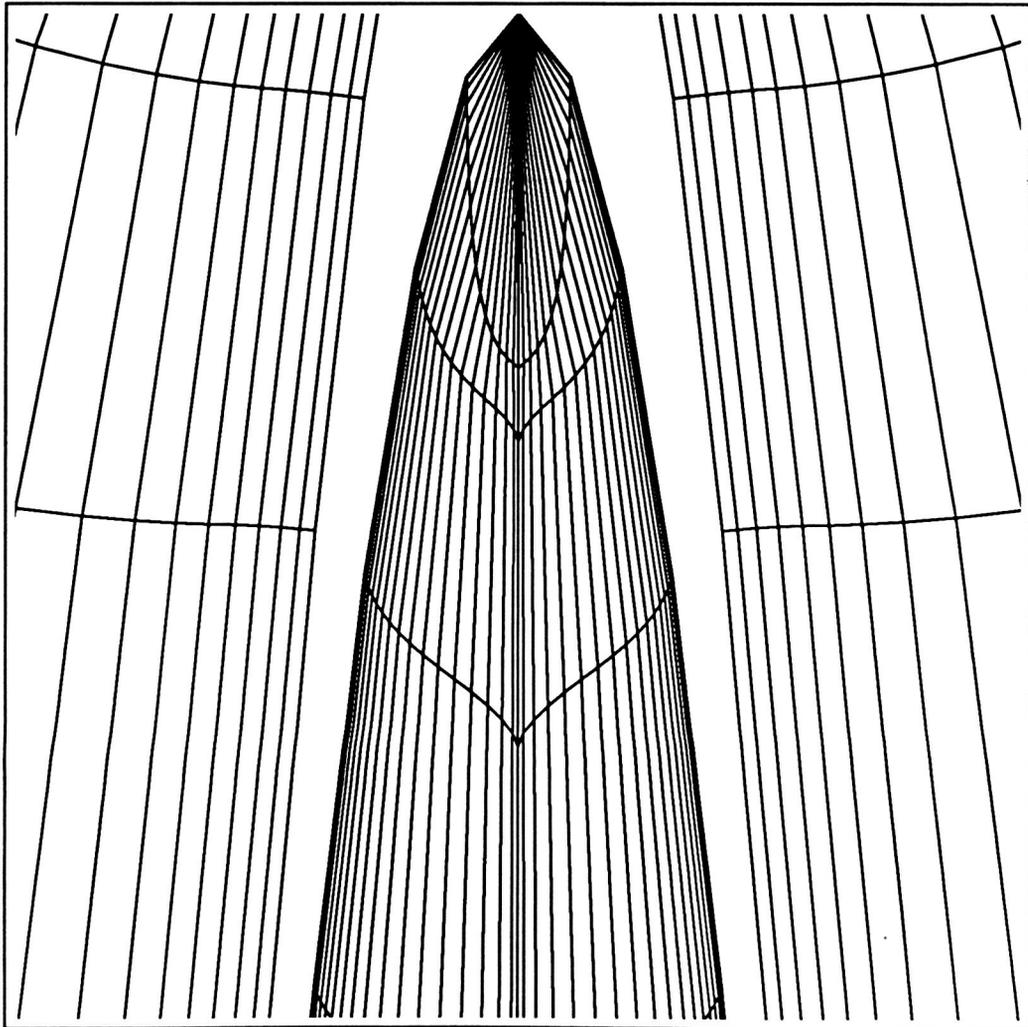


Figure 58: $\tau=0.5$, $\text{cam}=6.0$, pass 80 , detail of cavity
file: fgc560.8ptps

Forming a Cut In a Cavity

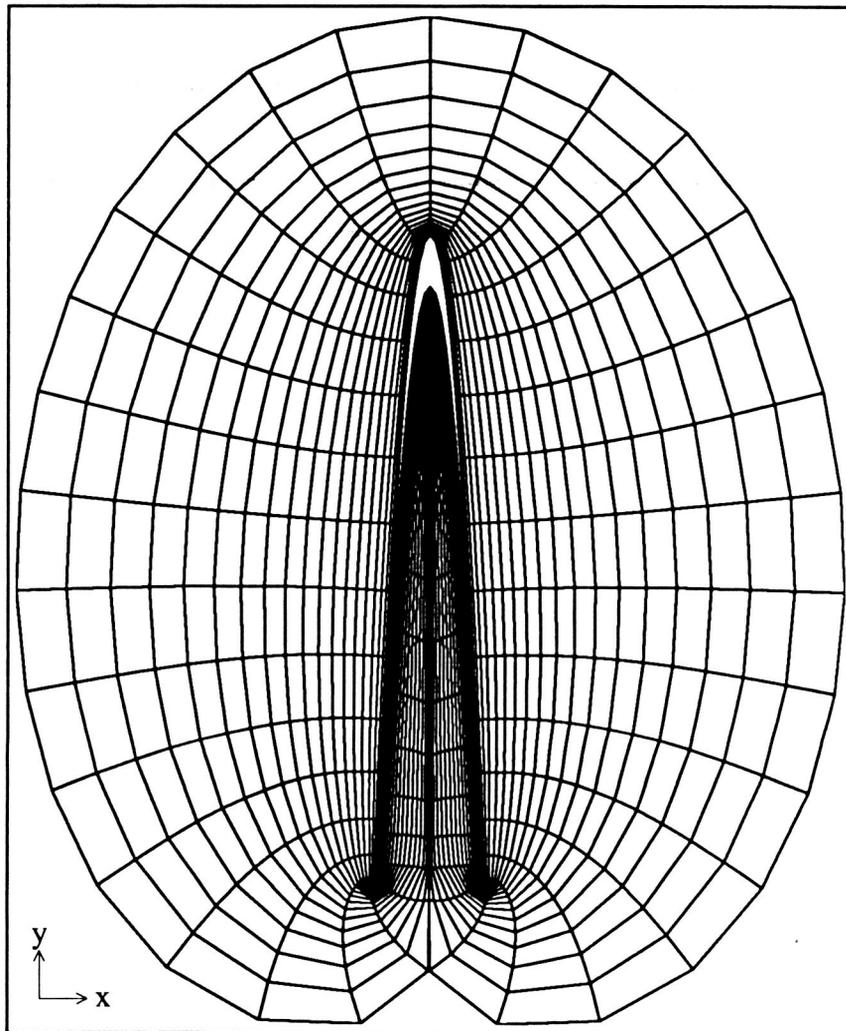


Figure 59: $\tau=0.5$, $\text{cam}=6.0$, pass 100
file: fgc560.10ps

Forming a Cut In a Cavity

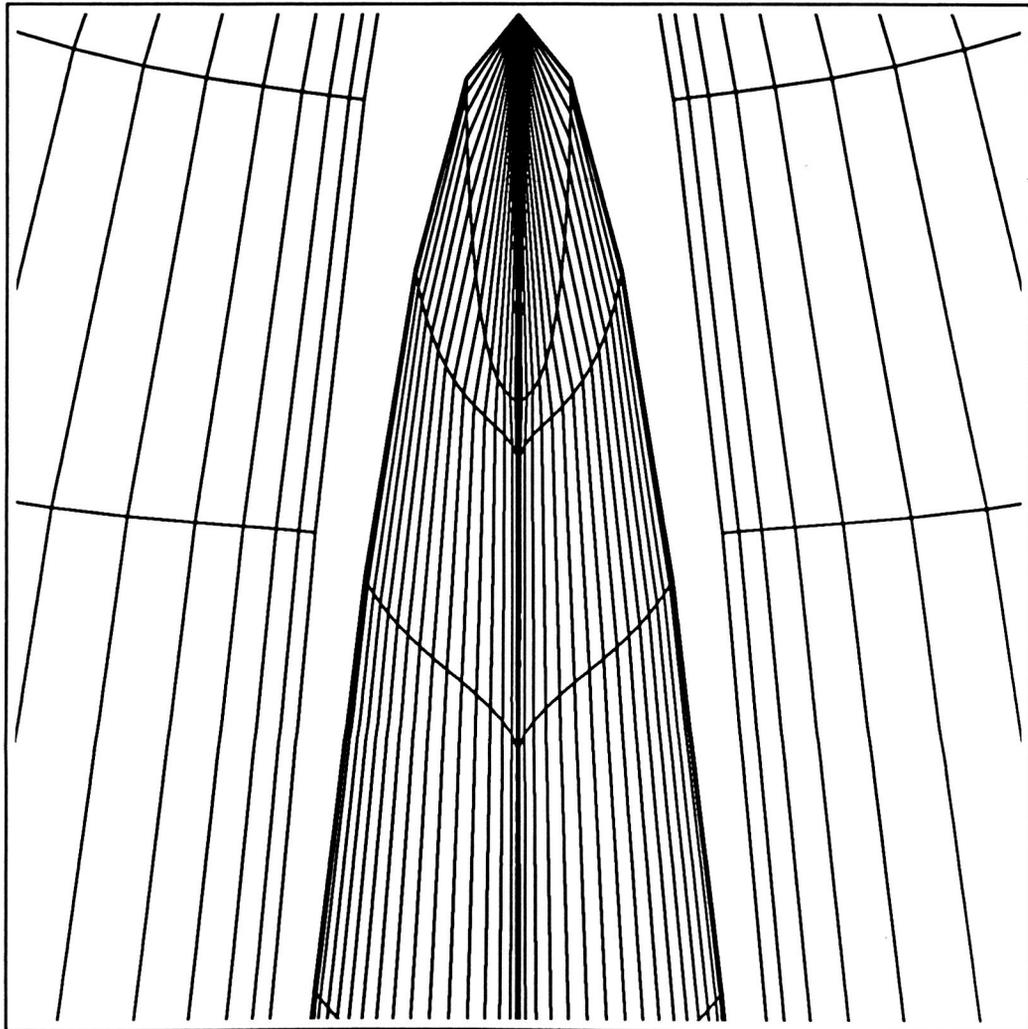


Figure 60: $\tau=0.5$, $\text{cam}=6.0$, pass 100, detail of cavity
file: fgc560.10ptps

Forming a Cut In a Cavity

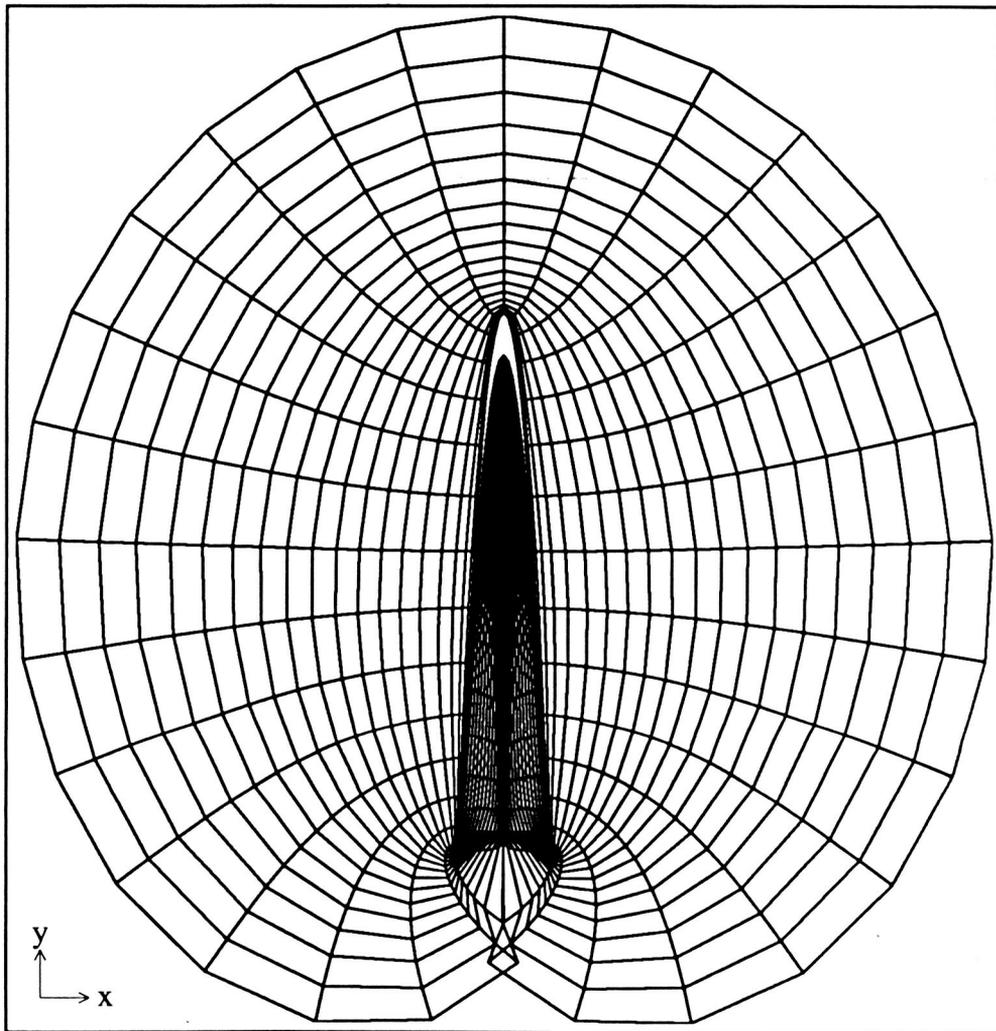


Figure 61: $\tau=0.5$, $\text{cam}=6.0$, pass 300
file: fgc560.30ps

Forming a Cut In a Cavity

ORIGINAL PAGE IS
OF POOR QUALITY

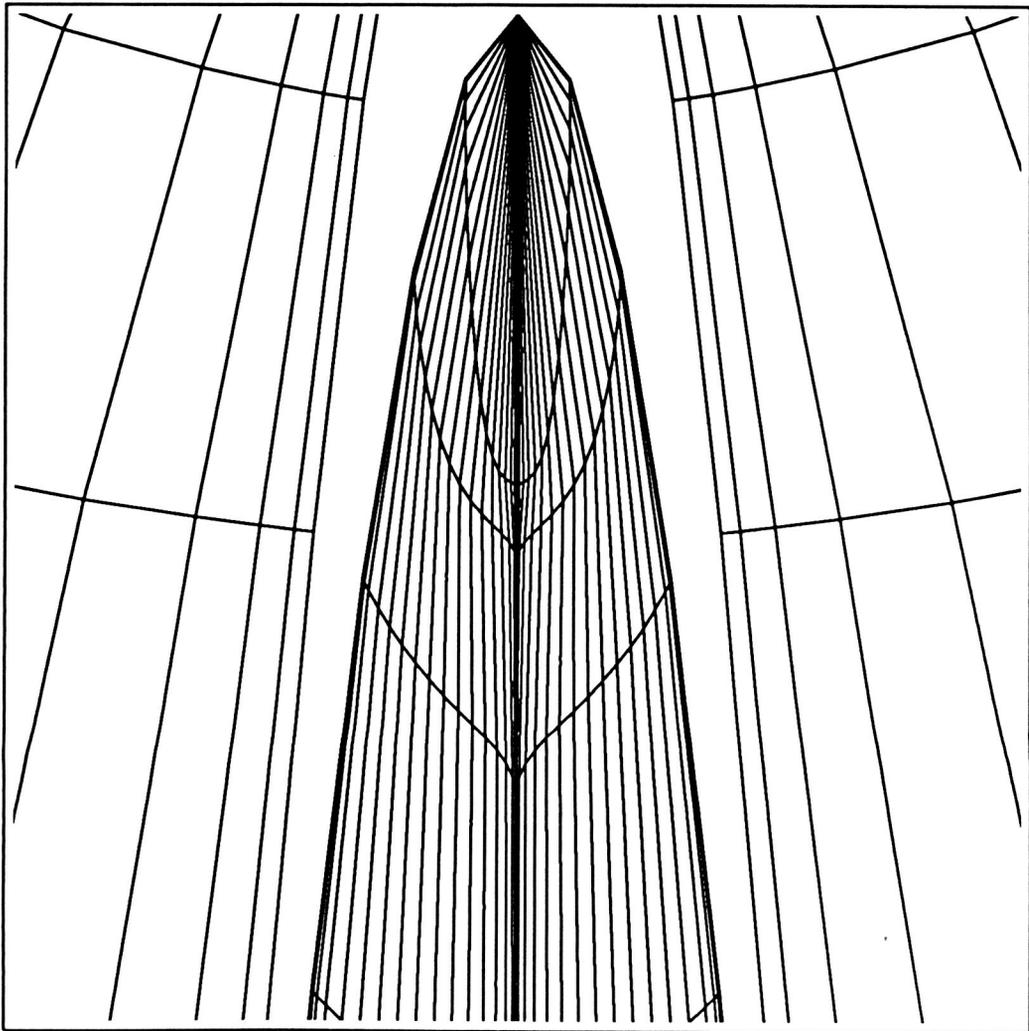


Figure 62: $\tau=0.5$, $\text{cam}=6.0$, pass 300, detail of cavity
file: fgc560.30ptps

7 Grid Breakdown

FORMING A CUT IN A CAVITY

Since it is required that the mapping between real space and computational space be one-to-one, that is

$$\vec{R} = \vec{R}(\vec{C}) \quad (89)$$

and

$$\vec{C} = \vec{C}(\vec{R}) \quad (90)$$

The intersection of grid line segments, other than at nodes, would violate this requirement for the following reason. The intersection of these grid lines is an intersection of lines connecting points which have adjoining index values. For instance one line segment may join $P(j,k)$ with $P(j+1,k)$ the other may join $P(i,l)$ with $P(i,l+1)$. Since ξ and η are linear functions of these indices and the indices are different in each set, the range of values of ξ and η are different between one set of points and the other set. Thus at the point of intersection in the r space coordinate system two values of \vec{C} exist for the one value of \vec{R} , violating the one-to-one requirement.

Since the only grid line that can be assumed satisfactory is a given boundary line, a check for line intersection should most logically start there and work towards the other boundary. But a more elementary check should be to first check whether the grid points nearest the boundary, and therefore the lines connecting them to the body, are outside the body, as they should be. If k indexes points in the radial or η direction, and j indexes points in the circumferential or ξ direction, then these first interior grid points can be noted as $P(j,k) = P(j,2) \equiv P_{j,2}$. $P_{j,2}$ joins the

body at the point $P_{j,1}$ which is between the points $P_{j\pm 1,1}$.

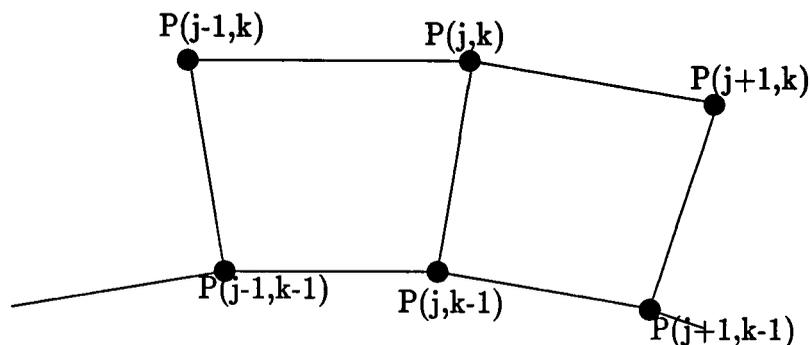


Figure 63: Portion of boundary and first row of grid cells next to boundary. Here $k=2$

If one defines following vectors

$$\vec{r}_t = \vec{r}_{j,2} - \vec{r}_{j,1} \quad (91)$$

$$\vec{r}_1 = \vec{r}_{j-1,1} - \vec{r}_{j,1} \quad (92)$$

$$\vec{r}_2 = \vec{r}_{j+1,2} - \vec{r}_{j,1} \quad (93)$$

then, for j increasing clockwise around the the body and k increasing radially away, it can be shown that if either of the following sets of conditions are met

$$\text{for } \vec{r}_1 \times \vec{r}_2 > 0 \quad (94a)$$

$$\vec{r}_1 \times \vec{r}_t > 0 \quad (94b)$$

and

$$\vec{r}_t \times \vec{r}_2 > 0 \quad (94c)$$

or

$$\text{for } \vec{r}_1 \times \vec{r}_2 < 0 \quad (95a)$$

either

$$\vec{r}_1 \times \vec{r}_t > 0 \quad (95b)$$

OR

$$\vec{r}_t \times \vec{r}_2 > 0 \quad (95c)$$

then \vec{r}_t and $P_{j,2}$ is on the wrong side of the boundary, the previous η line. If all the points at the $k=2$ line pass this test, the second test should be run before checking points at the next k level. This first test compares each point in the grid with points having neighboring j and k values. Thus, it uses relatively little computational time per point, but each point must be checked.

If the first test checks out the second test should be to check whether the k line intersects a previous k line. Figure 64 shows an example of this defect and the main reason to check for it. For strong convexities, when distances between the apex and its two adjoining points on the boundary are significantly different, the hyperbolic generator can cause the radial line emerging from the apex, the ξ line (the radial line near the bottom in figure 64), to swing far enough over such that intersections occur. This can be checked by testing the η line segment at each point for intersection with the next lower η line. This check, like the previous one, though it checks each point in the grid, makes only one or two checks for intersection per point.

A third breakdown, differing from the previous two in that there is no relation between the j values of the points compared, occurs when different parts of a boundary (the k_{\max} line), or boundaries of different grids, intersect. The simplest procedure is to test each boundary line segment for intersection with all other boundary line segments. An improvement on this procedure is the following. Since intersection occurs only for lines whose x and y values overlap for some portion of the lines, computational time can be reduced, if the the check for intersection

is made only after the simpler checks for x and y overlap are made. This can be done by first sorting either x or y into a monotonic sequence and examining the new arrangement of the old j values. When successive j values differ by more than one, indicating overlap, the other overlap check can be made. In other words, if the original sequence is $x(j)$, the new sequence can be denoted as $x(j(i))$. Then if $j(i)$ and $j(i+1)$ differ by more than one, there is overlap of x values of two separate line segments and the next test, to check for overlap of y values is made. If there is y overlap the check for intersection can be made. Because only points on the outer boundary are checked, the check for boundary intersection takes about the same order of magnitude of time as the previous two procedures the check for breakdown is of the same order of time as the previous two.

Grid Breakdown Due to Strong Convexity and Unequal Spacing Along Body

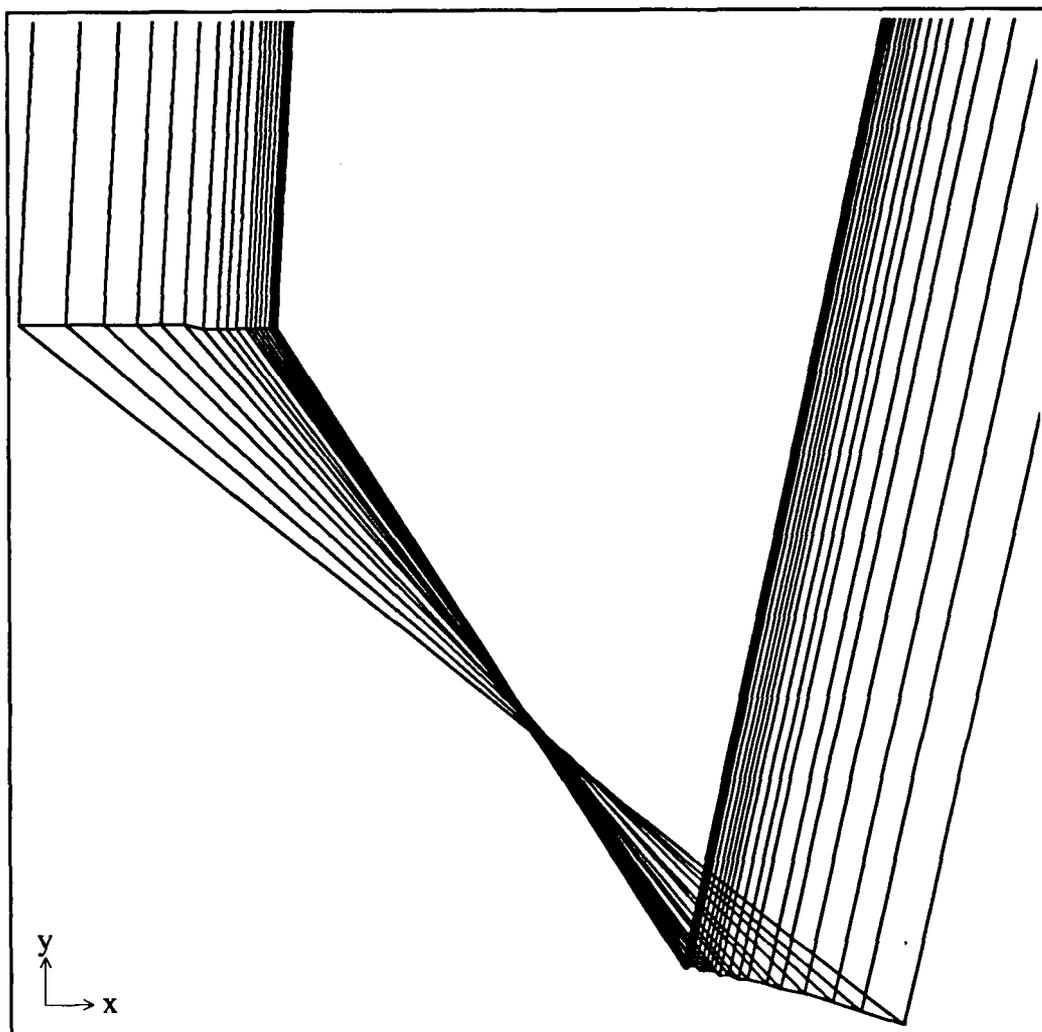


Figure 64: $\tau=0.5$, $\text{cam}=2.0$, detail of hg5y20.as used as initial grid
file : hg5y20.ptdps, bold line indicates body/boundary

8 Derivation of the Navier-Stokes Equations

The governing equations in fluid dynamics are based on the three conservation laws of mass, momentum, and energy. These equations can be derived in the following way [10], [6] using a control-volume of fixed dimensions within which the three laws are analyzed.

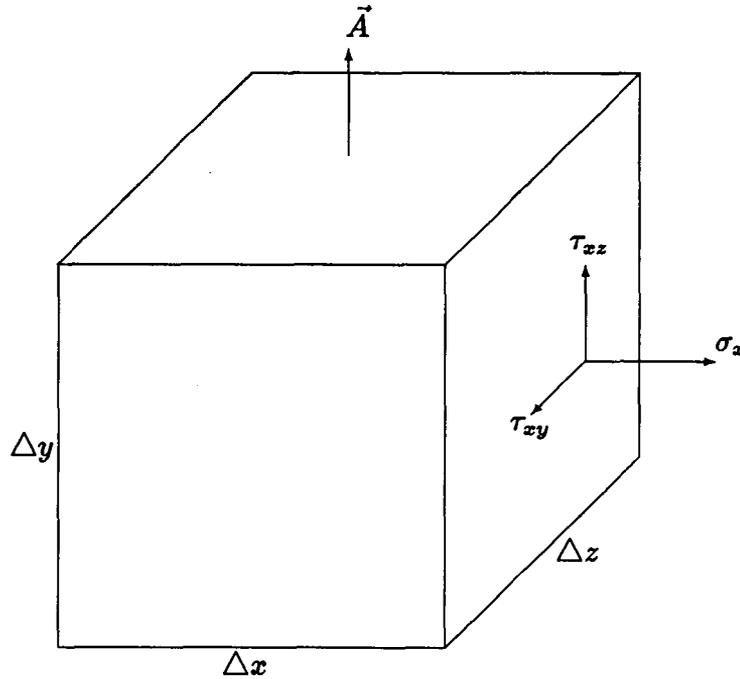


Figure 65: Fixed control volume showing area vector \vec{A} and the three surface stresses at one of the two surfaces normal to the x axis.

For the mass, or continuity, equation the change in mass per unit time, $\partial m / \partial t$ within the control volume (vol), is balanced by the net sum of mass entering and leaving the volume in unit time. Since density, ρ , equals $\partial m / \partial vol$, the rate of change in mass within the volume is

$$\left(\frac{\partial m}{\partial t} \right)_{\text{within}} = \frac{\partial \int dm}{\partial t} \quad (96)$$

$$\frac{dm}{dt} = \frac{\partial \int \rho \, dvol}{\partial t} \quad (97)$$

$$\frac{dm}{dt} = \int \frac{\partial \rho}{\partial t} \, dvol. \quad (98)$$

The mass entering the volume per unit time can be represented as

$$\left(\frac{dm}{dt}\right)_{\text{entering}} = - \oint_{\text{surface}} \rho \vec{u} \cdot d\vec{A}. \quad (99)$$

The minus sign is required because \vec{A} is directed away from the body. Combining equations (98) and (99) produces

$$\int \frac{\partial \rho}{\partial t} \, dvol = - \oint_{\text{surface}} \rho \vec{u} \cdot d\vec{A}. \quad (100)$$

Using Gauss' theorem, $\oint \vec{G} \cdot d\vec{A} = \int \vec{\nabla} \cdot \vec{G} \, dvol$, equation (100) becomes

$$\int \frac{\partial \rho}{\partial t} \, dvol = - \int \vec{\nabla} \cdot (\rho \vec{u}) \, dvol, \quad (101)$$

which leads to the point form, since integration is the same for all terms,

$$\frac{\partial \rho}{\partial t} + \vec{\nabla} \cdot (\rho \vec{u}) = 0 \quad (102)$$

The first term on the left exists when mass density within the control volume is unsteady. The second term indicates the rate of mass per unit volume building up within the control volume due to differences in density and/or flow rates at the various surfaces. When the first term is zero, indicating no change in mass density at a point, the second term must also be zero.

The momentum equation is a vector equation, containing three scalar equations, one for each direction. It is derived from Newton's second law: the rate of change in momentum for a body is equal to the forces acting on the body. For the fixed control volume, with mass flowing through it, the rate of change of momentum (of the mass within it), is equal to the net rate momentum is convected

across its entire surface by the flow, plus the sum of the forces acting on the mass within it

$$\frac{\partial \vec{P}_i}{\partial t} = \frac{d\vec{P}_c}{dt} + \vec{F}_b + \vec{F}_s \quad (103)$$

where P_i is the momentum within the control volume, P_c is the momentum convected across the surface, \vec{F}_b represents body forces, such as gravity, and \vec{F}_s represents surface forces such as pressure and friction.

Since $d\vec{P} = \vec{u} dm$, because \vec{u} may vary within the control volume, and $dm = \rho dvol$, the rate of change of momentum within the volume is

$$P_i = \int d\vec{P}_i = \int_{vol} \rho \vec{u} dvol \quad (104)$$

The momentum convected across the volume surface, per unit time, (covering a distance dr) is found by multiplying the differential of momentum at the surface with the dot product of the flow velocity \vec{u} and the surface element \vec{A} , and integrating over the whole surface

$$\frac{d\vec{P}_c}{dt} = \frac{d(\int_{vol} \rho \vec{u} dvol)}{dt} \quad (105)$$

$$= - \frac{d(\int_r \int_{area} \rho \vec{u} d\vec{r} \cdot d\vec{A})}{dt} \quad (106)$$

$$= - \int_{area} \rho \vec{u} \frac{d\vec{r}}{dt} \cdot d\vec{A} \quad (107)$$

$$= - \int_{area} \rho \vec{u} \vec{u} \cdot d\vec{A} \quad (108)$$

$$= - \int_{vol} \vec{\nabla} \cdot \rho \vec{u} \vec{u} dvol \quad (109)$$

The last change is by Gauss' theorem. As before, the negative sign in front of the integral is due to the direction of the area element.

When these momentum terms are on one side of the equation they then equal the sum of the forces acting on the mass in the control volume: body forces, such as gravity, and surface forces, or stresses, such as pressure and friction.

For the control volume the body forces are integrated over the entire volume. If \vec{F}_b is the total of body forces on the mass m in the volume then $\vec{f}_b = \vec{F}_b/m$ is the force per unit mass. And since density is defined as $\rho \equiv dm/dvol$, $\rho\vec{f}_b$ is the force per unit volume. Then the total force due to body forces is

$$\vec{F}_b = \int_{vol} \rho \vec{f}_b dvol. \quad (110)$$

Surfaces forces can be divided into two distinct types: normal stresses and tangential stresses. Normal stresses, denoted as $\vec{\sigma}_i = \sigma_i \hat{i}$, are pressure forces and act normal to the surface in question. The subscript indicates both the surface being acted upon and the direction of the force, which is normal to the surface. So $\vec{\sigma}_x$ acts on a surface area dA lying on a y - z plane (the line normal to a y - z plane lies in the x direction).

Tangential forces are shearing, or friction, stresses, such as those associated with viscosity. At a surface there are two independent tangential directions, therefore an indication of this must be made in denoting the symbol for this stress: $\vec{\tau}_{ik} = \tau_{ik} \hat{k}$. The first subscript, like that for the normal stress, denotes the plane the surface lies on. The second subscript indicates the direction of the force. So the two tangential forces acting at the y - z surface $dA\hat{x}$ are $\vec{\tau}_{xy} = \tau_{xy} \hat{y}$ acting in the y direction, and $\vec{\tau}_{xz} = \tau_{xz} \hat{z}$, acting in the z direction.

The normal and shearing stresses form a stress tensor which is symmetric

ORIGINAL PAGE IS
OF POOR QUALITY

$$S = \begin{pmatrix} \sigma_x & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \sigma_y & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \sigma_z \end{pmatrix} \quad (111)$$

$$= \begin{pmatrix} \sigma_x & \tau_{xy} & \tau_{xz} \\ \tau_{xy} & \sigma_y & \tau_{yz} \\ \tau_{xz} & \tau_{yz} & \sigma_z \end{pmatrix}. \quad (112)$$

The surface forces are integrated over the entire surface. Since the surface stresses form a tensor, its transformation by Gauss' theorem produces the following equation for the sum of the surface forces \vec{F}_s acting on the control volume

$$\vec{F}_s = \oint_{area} S \cdot d\vec{A} \quad (113)$$

$$= \int_{vol} \vec{\nabla} \cdot S \, dvol \quad (114)$$

where

$$\vec{\nabla} \cdot S = \frac{\partial S_{ij}}{\partial x_j} \quad i, j = 1, 2, 3. \quad (115)$$

The momentum equation is then, after removing the integration signs,

$$\frac{\partial(\rho\vec{u})}{\partial t} + \vec{\nabla} \cdot (\rho\vec{u}\vec{u}) = \vec{\nabla} \cdot S + \rho\vec{f}_b \quad (116)$$

and the x component is

$$\frac{\partial(\rho u_x)}{\partial t} + \vec{\nabla} \cdot (\rho u_x \vec{u}) = \frac{\partial \sigma_x}{\partial x} + \frac{\partial \tau_{xy}}{\partial y} + \frac{\partial \tau_{xz}}{\partial z} + \rho f_{xb} \quad (117)$$

Since σ is the negative of pressure this equation becomes

$$\frac{\partial(\rho u_x)}{\partial t} + \vec{\nabla} \cdot (\rho u_x \vec{u}) = -\frac{\partial p_x}{\partial x} + \frac{\partial \tau_{xy}}{\partial y} + \frac{\partial \tau_{xz}}{\partial z} + \rho f_b \quad (118)$$

The equations for the y and z components of momentum are found in a similar way.

$$\left(\sigma_x \quad \tau_{xy} \quad \tau_{xz} \right)$$

The energy equation derives from the first law of thermodynamics

$$dE = dQ - dW \quad (119)$$

In the case of a control volume with mass passing through it, the rate E_i , the energy within the control volume, is changing, equals the rate energy, E_c , is being convected in by the flow, plus the rate heat, $Q_{\text{conducted}}$, is conducted in (ignoring heat radiated in for flows of sufficient speed), plus the rate work W_b is being done on the mass within the volume due to body forces, plus the rate work, W_{sp} , is being done on the mass due to surface pressure forces, plus the rate work, W_{sv} , is being done on the mass due to surface viscous forces

$$\frac{\partial E_i}{\partial t} = \frac{dE_c}{dt} + \frac{dQ_{\text{conducted}}}{dt} + \frac{dW_b}{dt} + \frac{dW_{sp}}{dt} + \frac{dW_{sv}}{dt}. \quad (120)$$

The energy per unit mass is $\rho(e + 1/2u^2 + e_p)$, with e , $1/2u^2$ and e_p representing internal, kinetic, and potential energy respectively; $u^2 = \vec{u} \cdot \vec{u}$. Similar to the previous equations for mass and momentum, the energy terms are replaced by

$$\frac{\partial E_i}{\partial t} = \int_{\text{vol}} \rho(e_i + 1/2u^2 + e_{pi}) d\text{vol} \quad (121)$$

$$\frac{dE_c}{dt} = - \oint_{\text{area}} \rho(e + 1/2u^2 + e_p) \vec{u} \cdot d\vec{A} \quad (122)$$

$$= - \int_{\text{vol}} \vec{\nabla} \cdot \rho(e_c + 1/2u^2 + e_{pc}) \vec{u} d\text{vol}. \quad (123)$$

By Fourier's law, heat conduction per unit area is $q = -k\partial T/\partial n$, where k is a proportionality constant, T is temperature, n is the direction normal to the area, and the negative sign indicates that heat flux is from higher to lower temperatures.

In vector form, and integrating over the entire surface of the control volume the heat term in equation (120) becomes

$$\frac{dQ_{\text{conducted}}}{dt} = - \oint_{\text{area}} -k \frac{\partial T}{\partial n} \hat{n} \cdot d\vec{A} \quad (124)$$

$$= \oint_{\text{area}} k \frac{\partial T}{\partial n} \hat{n} \cdot d\vec{A} \quad (125)$$

$$= \int_{\text{vol}} \vec{\nabla} \cdot \left(k \frac{\partial T}{\partial n} \hat{n} \right) d\text{vol} \quad (126)$$

$$= \int_{\text{vol}} \vec{\nabla} \cdot (k \vec{\nabla} \cdot T) d\text{vol}. \quad (127)$$

The work W_b done by the body forces is found by integrating these forces over the entire volume. Since $W = \int dW = F dx$ and $F = \int \rho f d\text{vol}$, then ρf_{ib} is the body force per unit volume in the x_i direction and the energy due to this body force is

$$\frac{dW_b}{dt} = \frac{\int dW_b}{dt} \quad (128)$$

$$= \frac{\int \rho f_{ib} dx_i d\text{vol}}{dt} \quad (129)$$

$$= \int \frac{\rho f_{ib} dx_i d\text{vol}}{dt} \quad (130)$$

$$= \int \frac{\rho f_{ib} dx_i}{dt} d\text{vol} \quad (131)$$

$$= \int \rho f_{ib} u_i d\text{vol}. \quad (132)$$

This leads to

$$\frac{dW_b}{dt} = \int \rho \vec{f}_b \cdot \vec{u} d\text{vol} \quad (133)$$

Alternatively, this energy term (in fact all the energy terms associated with the momentum equation) can be obtained from the momentum equation by taking

the dot product of the momentum change due to the body forces with the velocity vector of the flow (129) becomes

$$\vec{F}_b \cdot \vec{u} = \int_{vol} \rho \vec{f}_b \cdot \vec{u} dvol \quad (134)$$

Applied to the surface force term in the momentum equation, the technique produces

$$\frac{dW_{sp}}{dt} + \frac{dW_{sv}}{dt} = (\vec{\nabla} \cdot S) \cdot \vec{u}. \quad (135)$$

and that due to the x component of change is

$$-\frac{\partial p_x}{\partial x} u_x + \frac{\partial \tau_{xy}}{\partial y} u_x + \frac{\partial \tau_{xz}}{\partial z} u_x + \quad (136)$$

The energy equation is then, after dropping the integration signs to get the differential form, and moving the convection energy term to the left,

$$\frac{\partial[\rho(e_i + \frac{u^2}{2} + e_{pi})]}{\partial t} + \vec{\nabla} \cdot [\rho(e_c + \frac{u^2}{2} + e_{pc})] \vec{u} = \vec{\nabla} \cdot (k \vec{\nabla} T) + \rho \vec{f}_b \cdot \vec{u} + (\vec{\nabla} \cdot S) \cdot \vec{u}. \quad (137)$$

9 Results and Conclusions

Steger's original algorithm for solving the combined elliptic-hyperbolic equations, using fixed boundaries, was shown to work well on certain bodies, for certain initial grids. As concavity increased, the initial grids generated by the hyperbolic generator, `hyg2d`, broke down, necessitating the use of some other initial-grid generator. This latter generator, not being a function of the body, spaced points on the boundary irrespective of body shape, causing loss of orthogonality in some areas.

The author then made modifications to the original algorithm to increase the range of bodies that the combined solver can handle. The first modification was to float the outer boundary. That is, at each iteration, after the interior of the grid was improved, the hyperbolic component, using the improved grid, generated a new and generally improved outer boundary. This allowed the use of initial grids with defects in the outer boundary. But, though an increased body complexity could now be dealt with, still, for strong enough concavities this algorithm breaks down. Additionally, as concavity increased, μ , the parameter used to control the amount of elliptic or hyperbolic, became more limited in the values it could take on. Only values within this range would allow for convergence to solution. Finding these values became more difficult as cavities grew in depth.

A second modification allows the use of a different kind of initial grid, one with few or no defects. This is a grid of compressed grid cells where the length of cells in the η direction is very small compared to the length in the ξ direction. By starting with such a defect-free grid and allowing grid cells to grow with each iteration, combined with the floating outer boundary it is possible to generate a

grid about more complex (higher concavity) bodies. Additionally μ can be more arbitrarily picked, affording more variability between hyperbolic and elliptic.

In the last modification to the algorithm, boundary movement within a cavity is modified. Instead of being generated by the hyperbolic alone it is modified to some degree by a forcing function that moves boundary points about a chosen boundary point so that corresponding boundary points merge. This causes a cut to form within the cavity, made up of the outer boundary. This modification is not complete. There still remains the problem of forcing the point at which the outer boundary joins itself to move far enough from the body such that the entire relevant flow field is covered.

When this is done a grid about a two-dimensional body with deep cavities may be achieved. Then, it should also be possible to generate a grid about multiple bodies. By attaching them with imaginary lines, such as to make one inner boundary, the cavities so formed may be manageable. This idea is demonstrated in figures 66 through 69. Figure 66 shows two bodies connected by a line, which forms a continuous inner boundary that outlines the two bodies. Figure 67 is a grid generated about this new body. Figures 68 and 69 are detailed views of the area surrounding the connection line. The main point here is that there is little if any breakdown of the grid, no intersections are noticeable.

Several methods of identifying grid breakdown have been described. Although the logic has been coded, and the first two techniques are in the grid generator program ehgrd, there remains yet the problem of using this information to control μ in order to eliminate the breakdown. One possibility is to simply turn the elliptic on (set μ high) wherever grid breakdown is detected. This, though, may

hamper using μ for resolution of flow variables.

Drawing One Grid About Two bodies by Connecting Them with a Line to Form One Body

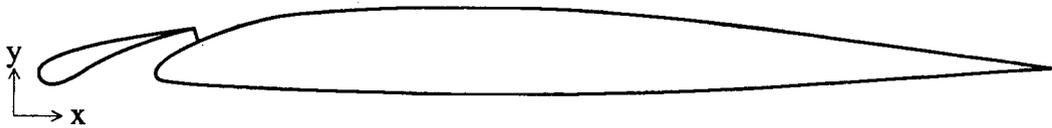


Figure 66: Airfoil and slat connected by a line
file : f14fp2.ps



Figure 67: 100th pass of the combined solver; boundary is
floated and grid cell volumes increase 5 percent per iteration
File : f14f3.10ps

Drawing One Grid About Two bodies by Connecting
Them with a Line to Form One Body

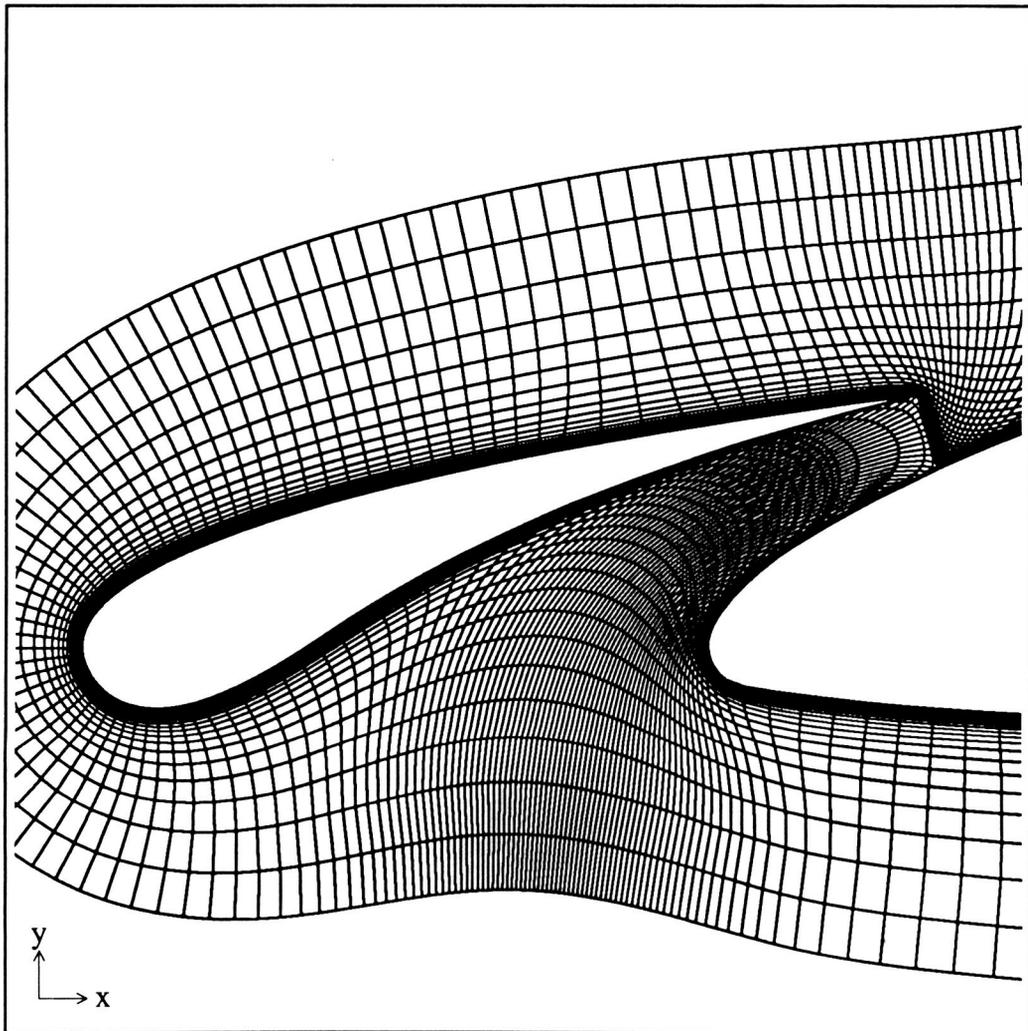


Figure 68: 100th pass, detail of final grid in figure 66
File : f14f3.10ptaps

Drawing One Grid About Two bodies by Connecting
Them with a Line to Form One Body

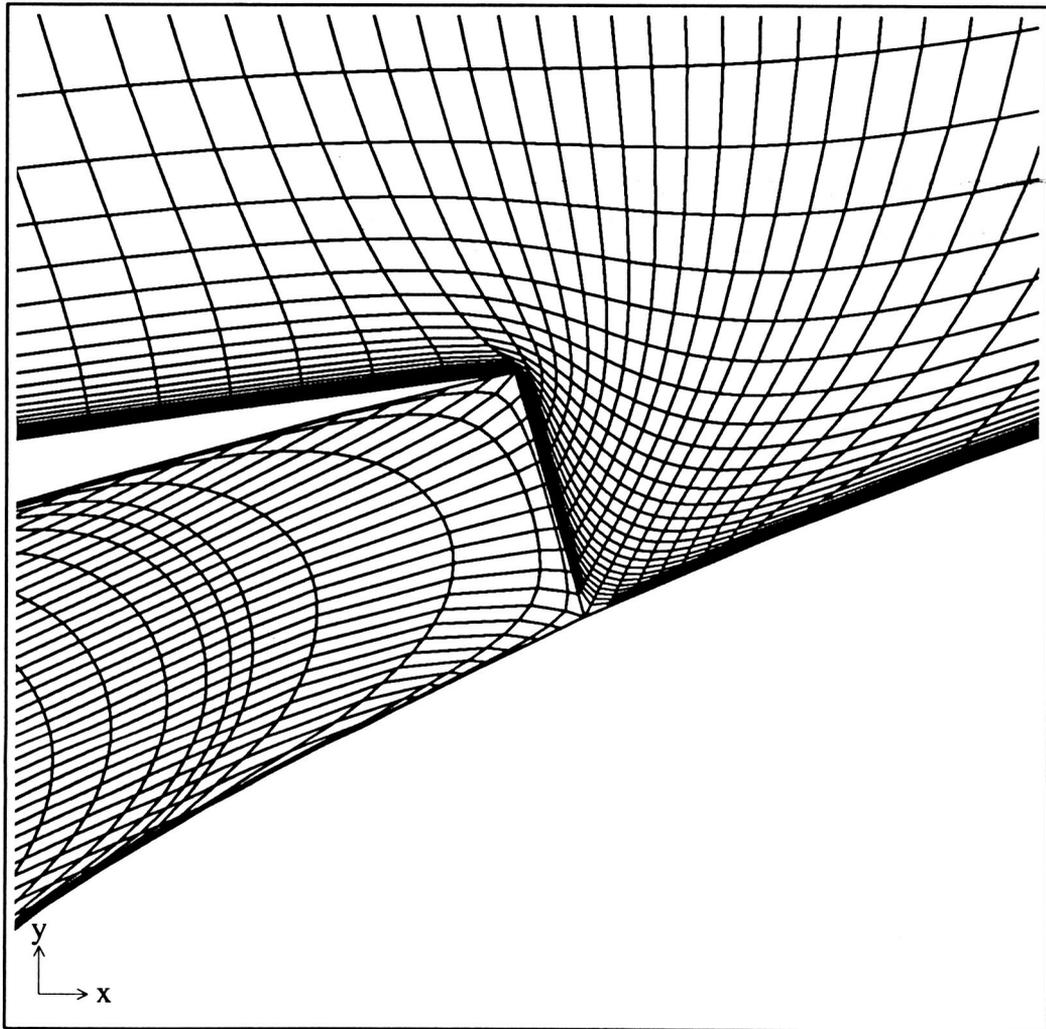


Figure 69: 100th pass, detail of final grid in figure 66
File : f14f3.10ptbps

10 Notation and Definitions

$$u_x \equiv \frac{\partial u}{\partial x} \quad (138)$$

$$u_{xx} \equiv \frac{\partial^2 u}{\partial x^2} \quad (139)$$

Linear Finite Difference Operators			
Operator	Symbol	Difference Representation	
Forward (Indexed)	Δ_x	$\Delta_x u = u(x + \Delta x) - u(x)$	(140a)
		$\Delta_x u = u(x_{i+1}) - u(x_i)$	(140b)
		$= u_{i+1} - u_i$	(140c)
Backward (Indexed)	∇_x	$\nabla_x u = u(x) - u(x - \Delta x)$	(141a)
		$\nabla_x u = u(x_i) - u(x_{i-1})$	(141b)
		$= u_i - u_{i-1}$	(141c)
Central (Indexed)	$\bar{\delta}_x$	$\bar{\delta}_x u = u(x + \Delta x) - u(x - \Delta x)$	(142a)
		$\bar{\delta}_x u = u(x_{i+1}) - u(x_{i-1})$	(142b)
		$= u_{i+1} - u_{i-1}$	(142c)
Central	δ_x	$\delta_x u = u(x + \frac{1}{2}\Delta x) - u(x - \frac{1}{2}\Delta x)$	(143)
Central (Indexed)	δ_x^2	$\delta_x^2 u = \delta_x(\delta_x u)$	(144a)
		$= u(x + \Delta x) - 2u(x) + u(x - \Delta x)$	(144b)
		$\delta_x^2 u = u(x_{i+1}) - 2u(x_i) + u(x_{i-1})$	(144c)
		$= u_{i+1} - 2u_i + u_{i-1}$	(144d)

2nd order accurate, 1st partial derivative

$$u_x = \frac{\bar{\delta}_x u}{2\Delta x} + O(\Delta x)^2 \quad (145)$$

$$u_x \doteq \frac{\bar{\delta}_x u}{2\Delta x} \quad (146)$$

2nd order accurate, 2nd partial derivative

$$u_{xx} = \frac{\delta_x^2 u}{(\Delta x)^2} + O(\Delta x)^2 \quad (147)$$

$$u_{xx} \doteq \frac{\delta_x^2 u}{(\Delta x)^2} \quad (148)$$

if $u = u(x, y)$, and if only a finite number of x and y values are required, as is generally the case in numerical analysis, these values can be indexed as

$$\begin{aligned} x &= x(l) \equiv x_l \\ y &= y(l) \equiv y_l \\ u &= u(l) \equiv u_l \end{aligned} \quad l = 1, 2, \dots, lmax. \quad (149)$$

Additionally, if the spacing in x and y values is constant a rectangular grid of points is formed, allowing double indexing of u

$$\begin{aligned} x &= x(m) \equiv x_m & m &= 1, 2, \dots, mmax. \\ y &= y(n) \equiv y_n & n &= 1, 2, \dots, nmax \\ u &= u(m, n) \equiv u_{m,n} \end{aligned} \quad (150)$$

But if this is not the case a transformation of variables may be performed that allows such a rectangular grid to exist. When $x = x(\xi, \eta)$ and $y = y(\xi, \eta)$, ξ and η can be indexed as

$$\xi = \xi(j) \equiv \xi_j \quad j = 1, 2, \dots, jmax, \quad (151)$$

$$\eta = \eta(k) \equiv \eta_k \quad k = 1, 2, \dots, kmax, \quad (152)$$

This leads to

$$x = x(j, k) \equiv x_{j,k}, \quad (153)$$

$$y = y(j, k) \equiv y_{j,k} \quad (154)$$

$$u = u(j, k) \equiv u_{j,k} \quad (155)$$

If u is some derivative with respect to ξ or η the finite difference equivalent can be stated as

$$v_\xi(j, k) \doteq \sum_{i=imin}^{imax} a_i v(j+i, k) \quad (156)$$

$$v_\eta(j, k) \doteq \sum_{i=imin}^{imax} b_i v(j, k+i) \quad (157)$$

Now if u happens to be a derivative with respect to x or y the chain rule can be used to transform into ξ, η variables so that

$$u_x = u_\xi \xi_x + u_\eta \eta_x \quad (158)$$

ORIGINAL PAGE IS
OF POOR QUALITY

Also by use of the metric transformation equations ξ_x and η_x can be changed to functions in ξ and η the finite difference representation is some combination of the u 's neighboring $u(j,k)$, so that if $v_x = u$, $v_x = v_x(j,k)$, and, for a given value of j and k , j^0, k^0

$$v_x(j, k) \doteq \sum_{i=i_{min}}^{i_{max}} a_i v(j^0 + i, k) \quad (159)$$

where a_i is some rational constant and i may take on negative values.

Convexity and Concavity

Convexity refers to the convex portions of a body. Concavity refers to the relative depth and curvature of the concave portions of a body. The algorithm used to generate the two-dimensional bodies used in this paper controlled the body shape by a parameter named *cam*. This parameter when set to zero, as in figure 70, produces a body with no concave portion. As *cam* increases the bodies produced have what can be called stronger concavities as shown in figures 71, 72 and 73.

The Relation of the Parameter Cam to Concavity

Parameter $\tau = 0.5$

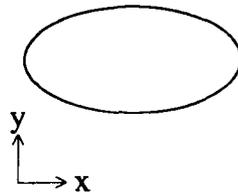


Figure 70: $\text{cam} = 0.0$

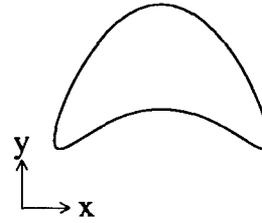


Figure 71: $\text{cam} = 0.40$

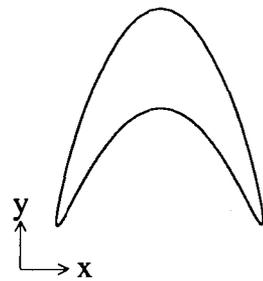


Figure 72: $\text{cam} = 0.80$

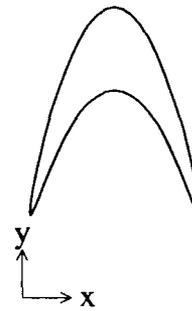


Figure 73: $\text{cam} = 1.0$

References

- [1] Garabedian P. R.: *Partial Differential Equations*, John Wiley, New York, 1964.
- [2] Thompson, J. F., Warsi, Z. U. A., Masten, C. W.: *Numerical Grid Generation Foundations and Applications*, North-Holland, New York, 1985.
- [3] Thompson, J. F., Thames, F. C., Masten, C. W.: Automatic Numerical Generation of Body-Fitted Curvilinear Coordinate System for Field Containing Any Number of Arbitrary Two-Dimensional Bodies. *J. Comp. Physics*, vol. 15, no. 3, July 1974, pp 299-319
- [4] Sorenson, R. L., Steger, J. L.: Simplified Clustering of Nonorthogonal Grids Generated by Elliptic Partial Differential Equations NASA TM-73252, 1977.
- [5] Steger, J. L.: Generation of Three-Dimensional Body-Fitted Grids by Solving Hyperbolic Partial Differential Equations NASA TM-101069, 1989.
- [6] Anderson, D. A., Tannehill, J. C., Pletcher R. H.: *Computational Fluid Mechanics*, Hemisphere Publishing Corporation, New York, 1984.
- [7] Sorenson, R. L.: A Computer Program to Generate Two-Dimensional Grids About Airfoils and Other Shapes by the Use of Poisson's Equation, NASA TM-81198, 1980.
- [8] Steger, J. L.: Generation of Body-Fitted Coordinates Using Hyperbolic Partial Differential Equations. *SIAM J. Sci. Stat. Comput.*, vol 1, no 4, December 1980.

- [9] Van Dalsem, W. R., Steger, J. L.: The Fortified Navier-Stokes Approach, proceedings of the Workshop on Computational Fluid Dynamics, Univ. of California, Davis, June 17-18, 1986.
- [10] Schlichting Herman : Boundary Layer Theory, McGraw-Hill Book Company, New York, 1979.