D15

GRID GENERATION USING COARSE, SMOOTH FINITE ELEMENTS

Lawrence J. Dickson
Department of Aeronautics
and Astronautics
University of Washington

I. Introduction

The grid generation problem lends itself to the use of finite elements and variational equations.

(1) Grids are usually generated as smooth solutions to "nice," elliptic differential equations--just the equations well suited to variational methods.

(2) The use of smooth finite elements gives the grid a functional expression, which can be examined, evaluated, manipulated, and modified naturally and cheaply.

(3) The "grid equations" are chosen for their qualitative character. Exactitude of solutions does not matter as long as this is preserved. As a result, extremely coarse (cheap) finite elements may generate a grid of high quality, if the boundary conditions are well parametrized.

I succeeded in demonstrating the following:

(1) Grid-quality solutions of a wide variety of equations--(direct) Laplace's, biharmonic, Helmholtz, even nonlinear--can be generated to fit reasonable functional boundary conditions in 2D using very coarse rectangular finite elements, often 6x3 C^2 bicubic. I even tried some "wavy" operators (with no natural variational expression) to demonstrate the method's versatility. I did not try the inverse Laplace equation, but I expect no problem but cost.

(2) The finite element grids can be refined, locally modified and "fine-tuned" using a simple, cheap composition-of-functions approach, without having to solve the differential equation repeatedly.

II. The Finite Elements

Smooth, rectangular finite elements were used; the ones here are C^2 bicubic in the interior. For the linear equations, an option of C^1 cubic boundary conditions with arbitrarily dense nodes was included. In the examples shown there are, unless otherwise mentioned, six patches "circumferentially" and three "radially," of which only the innermost ring of patches "radially" is plotted.

Where C^1 boundary conditions are used, second derivative discontinuity in the "circumferential" direction is confined to the ring(s) of elements meeting the C^1 boundary.

III. The Variational Expressions

Both linear and non-linear equations are solved by minimizing a variational integral. In the non-linear case there is iteration. Equations to fourth order (i.e., expressions squared in the variational integral to second order) are treated. In the case of the "wavy" Helmholtz equation $H(f) = \nabla^2 f + k^2 f = 0$, the variation integral for H^*H is used with Dirichlet ("underdetermined") boundary conditions.

A variational approach to solving the inverse Laplace's equation is known, but was not tried in this research. Other inverse equations (such as biharmonic) could also be used.

Equation	Variational Integrand
$\nabla^2 f = 0$	$\lvert\nabla f\rvert^2$
$\nabla^2 f - k^2 f = 0$	$\lvert\nabla f\rvert^2 + k^2 f^2$
$\nabla^2 f + k^2 f = 0$	$(\nabla^2 f + k^2 f)^2$
$\nabla^4 f = 0$	$(\nabla^2 f)^2$
$\nabla^2 (f^2) = 0$	$\lvert f\nabla f\rvert^2$

The boundary conditions are always "Dirichlet-like" (Dirichlet for second order, Dirichlet or Dirichlet + Neumann for 4th order), which allows simple minimization of the integral.

Since the finite element grids were so coarse, solution of the equations was by a one-step symmetric matrix solver.

IV. Fine-Tuning

The grids are defined by functional equations $x = x(i,j)$, i and j being "counting" variables. But the solutions to the differential equations are $x = x(s,t)$. It remains to define $s(i,j)$ and $t(i,j)$.

The simplest approach is to make them merely linear functions. As a matter of fact, I imposed the boundary conditions $s(i_L,j) = \text{const}$, $s(i_R,j) = \text{const}$, $t(i,j_B) = \text{const}$, $t(i,j_T) = \text{const}$, so that (s,t) lies in a box; then I manipulated the interior values through transfinite interpolation. If i is "circumferential" the maps are as follows:

$$\binom{i}{j} \xrightarrow[\substack{i = u \\ j = c(u,v) \\ \text{(unbroken cubic in v)}}]{} \binom{u}{v} \xrightarrow[\text{TFI}]{} \binom{s}{t} \xrightarrow[\text{FE}]{} \binom{x}{y}$$

Where the F.E. solution has high skewness or is nearly singular, the first map allows the requirement on the TFI boundary condition needed to mend this to be multiplied by a small constant, avoiding "grid folding." A price is paid; the grid comes out looking irregular. It is better to avoid the skewness in the FE solution itself, as by using the biharmonic equation with perpendicular boundary conditions.

V. Conclusions

The finite element approach to grid generation has proved eminently successful for linear grid equations. Its coarseness is a very desirable characteristic, most encouraging from the point of view of extending it to 3D. The wide choice of equations--controlled by a small "black box" determining the variational integrand--is another asset. So is the

method's usability, at little extra cost, with boundary conditions more finely specified than the interior finite element grid.

Iterative solution of nonlinear equations increases the expense by more than an order of magnitude, usually requires numerical integration of the variational expression (often convenient in the linear cases too), and makes it very difficult in general to apply finely-specified boundary conditions. I think there are better approaches (see below).

The "fine-tuning" needs further refinement itself, especially in handling variations in the normal velocity of the grid. Choosing the grid equation to yield perpendicularity of the finite element grid map (possible, for instance, with the biharmonic equation) is a help. Control of "circumferential" grid density at chosen locations worked well.

VI. Future Directions

Algorithms should be derived to parametrize boundary conditions in such a way as to yield good grids using direct (linear) equations, such as Laplace's or the biharmonic. The idea is to imitate the Riemann mapping, by slowing down where convex (avoiding boundary overlap), and speeding up where concave (avoiding "folds" in grid interior). This should make use of inverse equations less necessary, and if done right should be extendable to 3D.

The "fine-tuning" algorithms must be refined. They are in principle applicable to any grid that can be described as a function.

For the inverse equations (for instance, Thompson's method), I suggest use of linear paneling schemes in (x,y) space. The resulting solution can be approximated by (s,t) finite elements simply by solving for the (few) internal nodal values of (s,t) and using inverse function theorem derivative evaluations. If the solution needs to be iterated, linearized variational expressions using this as a starting point should be cheap. Such a method might even have uses in linearized flow simulation, as for cheap streamline tracking.

The method needs to be extended to 3D, adapted for "block" grids (with the equations, if desired, being valid across block boundaries), and adapted for vector computers.

VII. Figures

The figures are true representations of the functions they illustrate, although some "handwork" was done on some of them to circumvent bugs in the evaluation and plotting software.

All internal finite element grids are six circumferentially by three radially, with only the innermost radial layer plotted, unless otherwise mentioned.

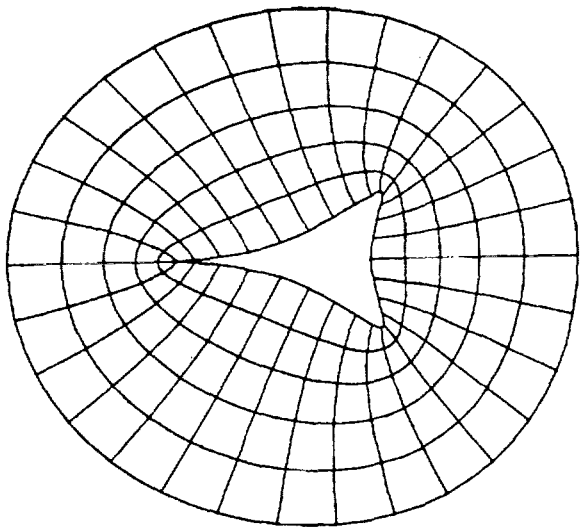

Figure 1.- "Heart" - Laplace's equation was approximated, with boundary conditions hand-parametrized to give a well-conditioned if not unskewed grid.

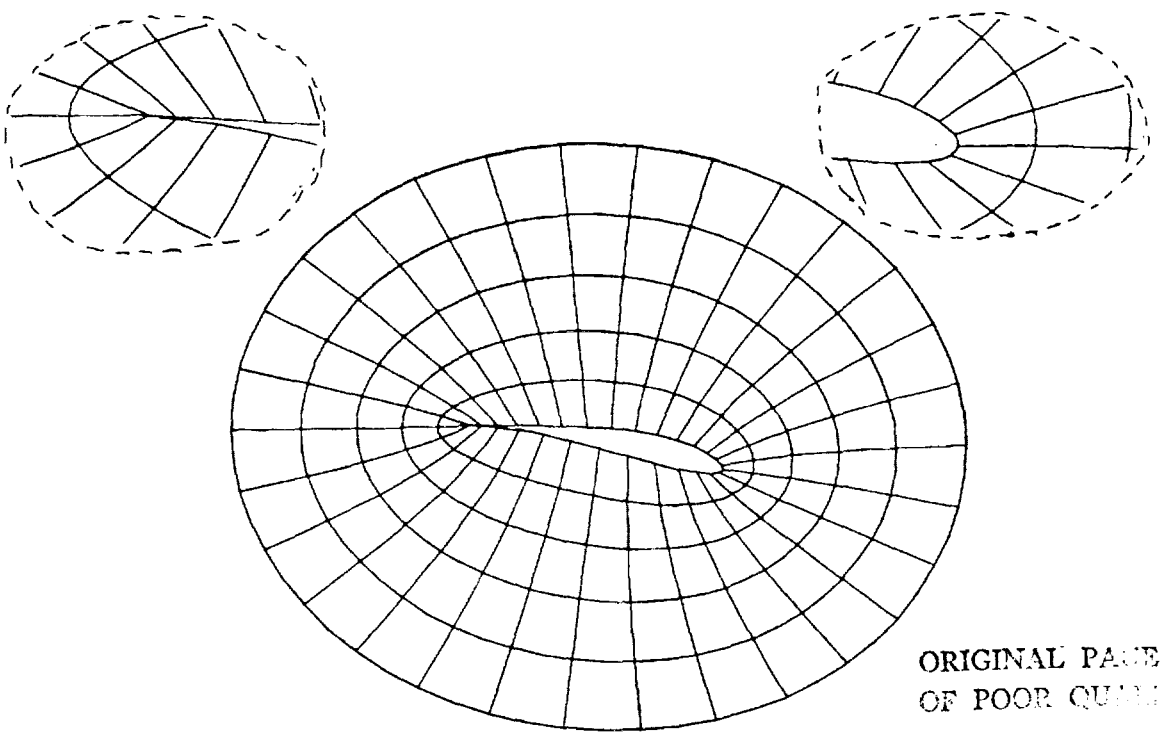

Figure 2.- "Joukowski" (Laplace's) - The natural, analytic parametrization of a Joukowski airfoil was imitated by a six-node C^2 cubic periodic spline. Success in avoiding skewness was middling, as the insets show. "Radial" velocity at trailing edge was non-zero.

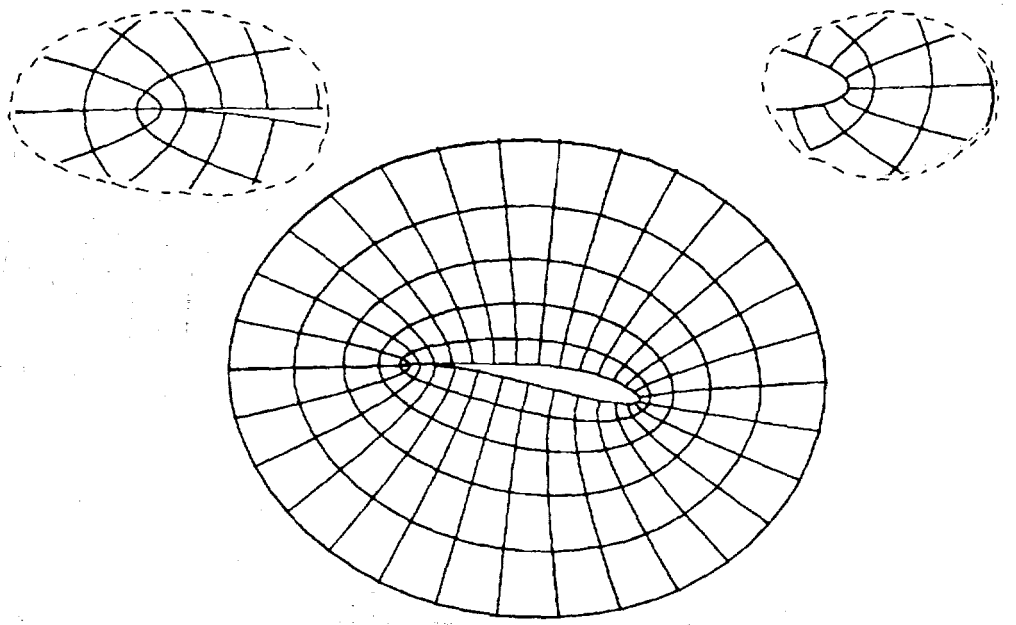

Figure 3.- "Joukowski" (Biharmonic) - The biharmonic equation's normal derivative condition was used to enforce conformality in the limit exactly at boundaries. (This makes the grid C^1 near the airfoil, and requires the "fine" boundary condition algorithm.) The insets show its success, and also that "normal" velocity at TE is zero.

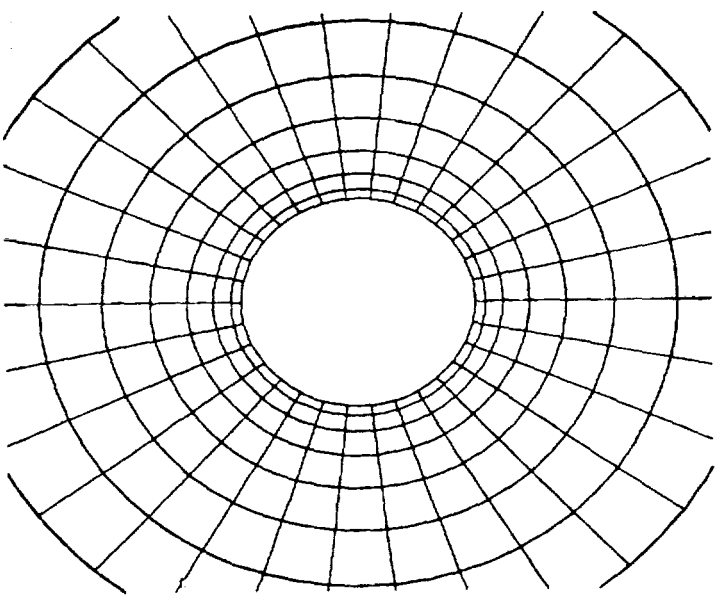

Figure 4.- "Helmholtz" - The equation is $\nabla^2 f - k^2 f = 0$, $k = .65$, with $0 \leq s = 3\theta/\pi \leq 6$ and $0 \leq t \leq 3$, radially symmetric boundary conditions, $r(t=0) = 1$, $r(t=3) = \exp(\pi)$. $0 \leq t \leq 1.4$ is plotted.

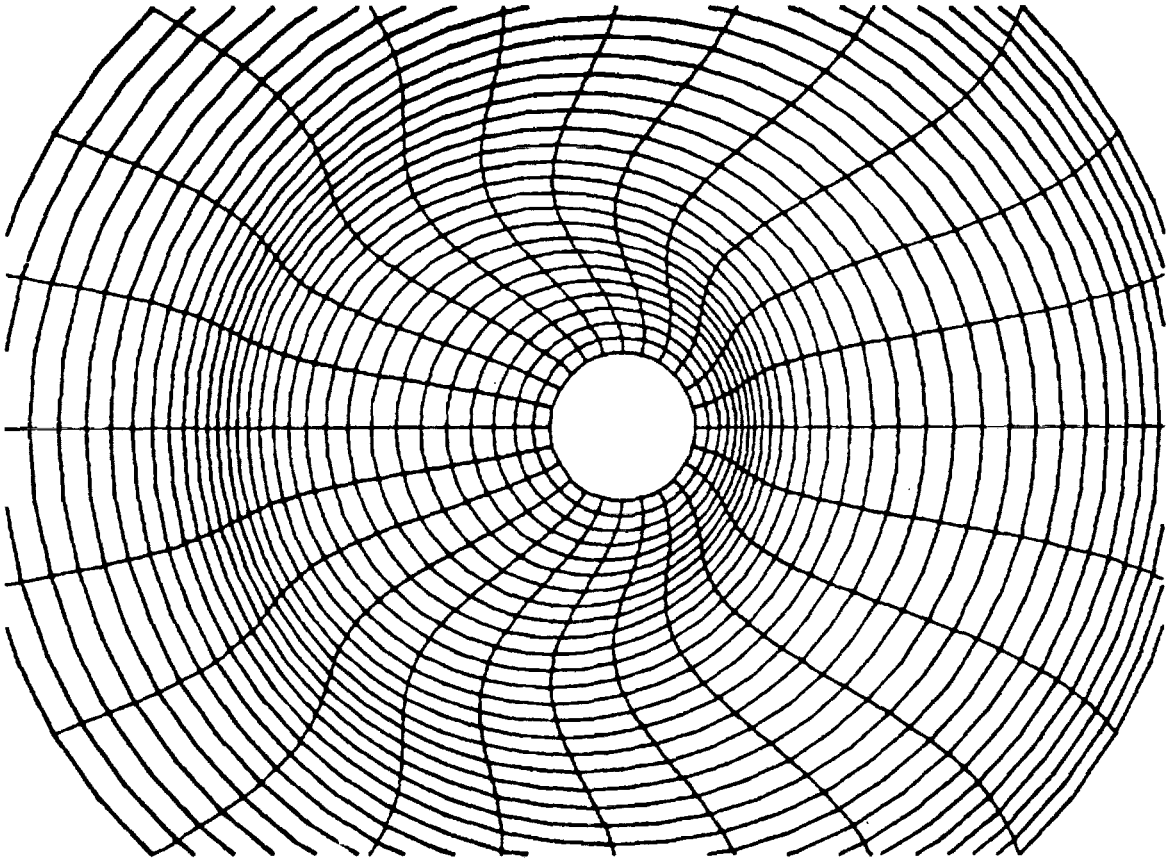

Figure 5.- "Wavy Helmholtz" - The equation is $\nabla^2 f + k^2 f = 0$, $k = \pi/3$, with $0 \leq s = 3\theta/\pi \leq 6$ and $0 \leq t \leq 6$. At $t = 0$, $(x,y) = (.5+\cos\theta, \sin\theta)$, and at $t = 6$, $(x,y) = \exp(\pi)\ (\cos\theta, \sin\theta)$. x includes a wave that traverses 1.25 cycle in joining these boundary conditions. The plot shows all the grid (except for screen cutoffs), which was solved on a 6×6 finite element mesh.

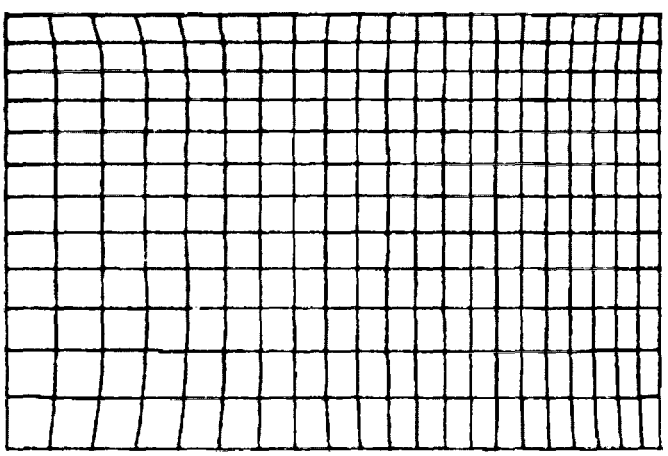

Figure 6.- Nonlinear - The equation is $\nabla^2(f^2) = 0$. This 3×3 grid, shown in its entirety, approximates $x = \sqrt{1.75t+1}$ and $y = \sqrt{s+1}$, as expected from equation and boundary conditions. Slight deviation is visible in x, due to the coarseness of the grid and the (2×2) Gaussian integration.

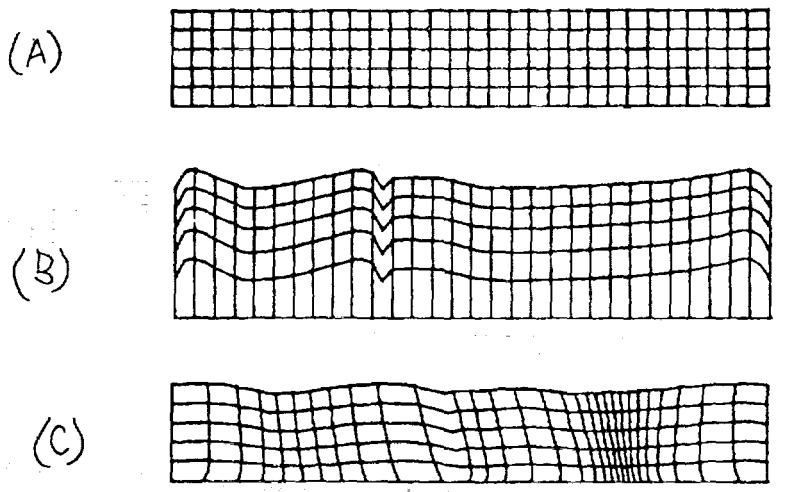

Figure 7.- "Fine tuning," Laplace - (A) = (i,j), (B) = (u,v), (C) = (s,t) in the discussion of Section IV. The wiggles in (B), due to skewness, are present only to second order in (C).

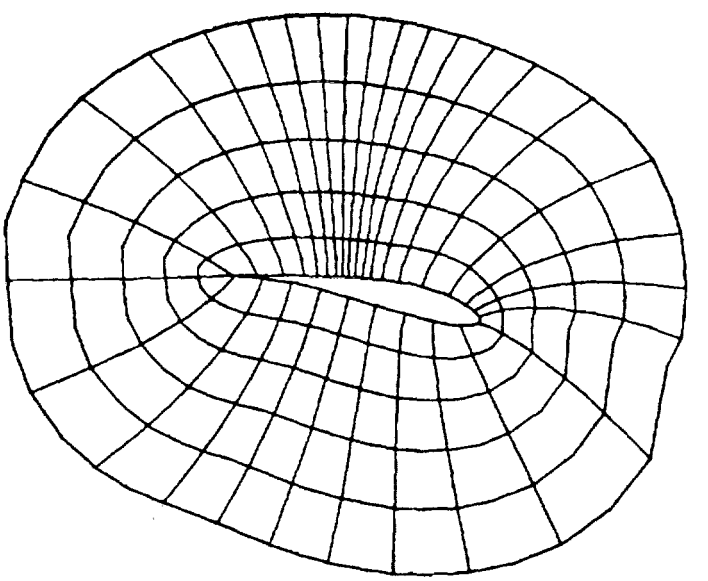

Figure 8.- "Fine tuned" Joukowski (Laplace) - The result of composition by the maps of Figure 7 is shown. (The "fine-tuning" specifications were deliberately clumsy.) The "shock densing" is good and the TE not too bad, but the LE shows a nasty glitch where skewness had to be corrected. The finite element map is that of Figure 2.

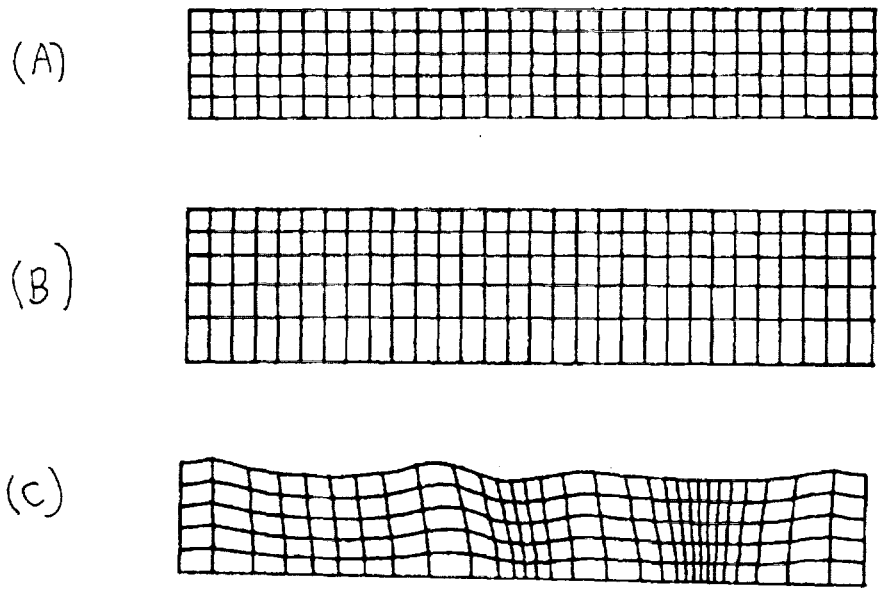

Figure 9.- "Fine tuning," Biharmonic - Smoother than Figure 7, since there is no skewness to be corrected at the boundary.

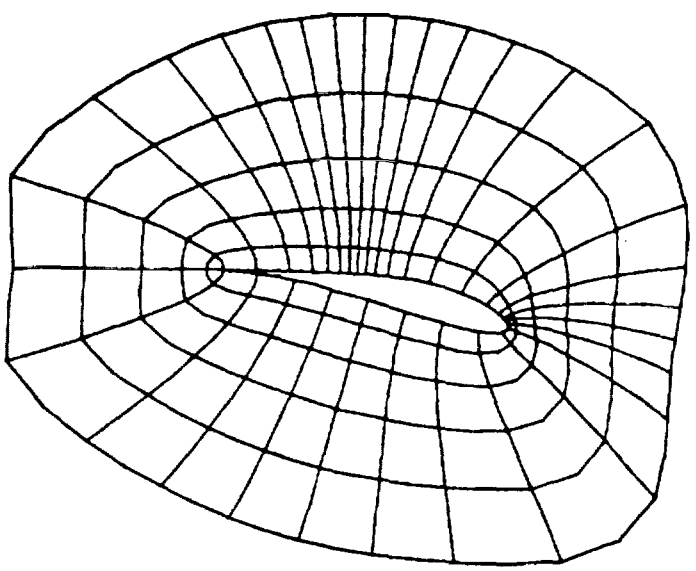

Figure 10.- "Fine tuned" Joukowski (Biharmonic) - Much better than Figure 8 at the LE, due to lack of skewness at the boundary in the map of Figure 3. The TE is not so good, due to adjusting to zero "normal" velocity there, a defect more easily correctable than the problem in Figure 8.