Linear stability analysis of hypersonic boundary layers

By E. Guilyardi, J. J. W. van der Vegt[1] AND J. H. Ferziger[2]

1. Motivation and objectives

Transition to turbulence is of great importance in the design of the new generation of hypersonic aircraft. Together with experiments, direct numerical simulations can provide valuable information on this phenomenon which is yet not well understood. In performing such simulations, linear stability analysis can be of enormous value in providing physical understanding and initial conditions.

The growth or decay of infinitesimal perturbations superposed on laminar solutions of the Navier-Stokes equations is the subject of the linear stability theory. The basic equations governing the linear stability of parallel-flow compressible boundary-layers are derived by linearizing the Navier-Stokes equations about the laminar flow. These initial perturbations are usually assumed to be of the form

$$u'(x,y,t) = \hat{u}(y)e^{i(\alpha x - \omega t)}. \tag{1.1}$$

For temporal stability analysis, α, the streamwise wavenumber, is fixed and real and ω, the frequency, is complex; for spatial analysis, α is complex and ω is fixed and real. In temporal analysis, $\omega = \omega_r + i\omega_i$, ω_r is the frequency and ω_i is the growth rate of the perturbation. These infinitesimal disturbances are imposed on the compressible Navier-Stokes equations linearized about a laminar boundary layer solution. If it is assumed that the mean flow is locally parallel, a set of five ordinary differential equations is obtained. Of these, three are the second order momentum equations, one is the second order energy equation, and one is the first order continuity equation; thus the complete system is ninth order. For a complete review of boundary-layer stability theory, see Reshotko (1976) or Mack (1984).

Mack (1984) showed that the second mode, which is active at supersonic speeds, has considerably higher growth rates than the first mode (which are the Tollmien-Schlichting waves). Mack further showed that the second mode is destabilized by wall cooling, unlike the first mode. The second mode also offers the advantage that two-dimensional waves are the most unstable second modes. We chose to begin by performing calculations of second mode instability of a flat plate boundary layer at Mach 4.5 as other computations are available for comparison (Erlebacher and Hussaini, 1990).

With the aid of the temporal linear stability code COSAL written by Malik (1982, 1990), our goal is to generate profiles of the most unstable waves to provide initial

1 Present address: ICOMP, NASA-Lewis Research Center, Cleveland, OH
2 Stanford University

data for the direct numerical simulation code written by van der Vegt (1990). The COSAL code was modified in several ways to meet this requirement. Improvements needed in the accuracy of the computation of the eigenfunctions were noted in van der Vegt and Ferziger (1990). Various numerical methods and grid adaptation have been applied in order to increase the accuracy of the results produced by the linear stability code.

2. Accomplishments

2.1. The linear stability code : COSAL

2.1.1 Introduction

COSAL is a compressible linear stability analysis code for two-dimensional boundary layers. It uses an iterative finite-difference method to compute the most unstable eigenvalue and requires an accurate estimate of the most-unstable eigenvalue. A local eigenvalue search procedure improves the accuracy of the eigenvalue and also yields eigenfunctions and group velocities. A global eigenvalue procedure was developed which may be used when no estimate of the most unstable eigenvalue is available. The elements of COSAL that were modified are first presented in the following lines.

2.1.2 Mean flow laminar profile

The mean flow is a similarity solution for the boundary layer on a flat plate obtained from the compressible form of the boundary layer equations; the resulting ordinary differential equations are solved using Keller's box method; see Cebeci and Smith (1974) for details. The mean flow profile for an adiabatic flat plate at Mach 4.5 is given in Figure 1.

The grid used for the numerical discretization of the boundary-layer equations was originally an exponentially stretched mesh which yielded high resolution near the wall and rapidly increasing mesh spacings away from it.

2.1.3 Global method

When no guess of the most-unstable eigenvalue is available, COSAL uses a global method that computes the whole eigenvalue spectrum. The finite differenced compressible stability equations can be reformulated as a matrix eigenvalue problem

$$\overline{A}\Phi = \omega\overline{B}\Phi, \tag{2.1}$$

where ω is the eigenvalue and Φ the discrete representation of the eigenfunctions. The eigenvalues are the roots of the determinant equation

$$Det\left|\overline{B}^{-1}\overline{A} - \omega I\right| = 0. \tag{2.2}$$

This is a standard matrix eigenvalue problem and is solved using the LR method. The most unstable eigenvalue is the one that satisfies the conditions

$$\omega_i > 0$$

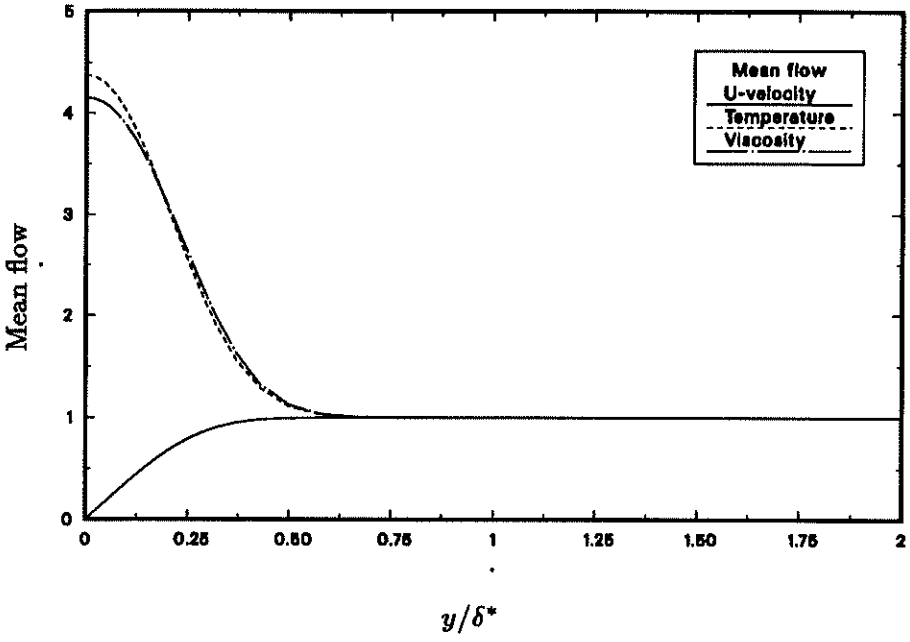

FIGURE 1. Mean Flow Profile for a Flat Plate, $M_\infty = 4.5$, $Pr = .7$, $Re_{\delta^*} = 8000$.

and

$$|\omega_i| = \max_{spectrum}(|\omega_i|).$$

2.1.4 Local method

When a guess for the most-unstable eigenvalue is available, it can be improved by a local method which also computes the corresponding eigenfunction. In the original version of COSAL, this was done with an inverse Rayleigh iteration procedure, for which the theory was presented in Wilkinson (1965). Generalization of this procedure to the compressible stability problem results in the following algorithm

$$(\overline{A} - \omega_k \overline{B})\Phi^{(k+1)} = \overline{B}\Phi^{(k)} \tag{2.4}$$

$$(\overline{A} - \omega_k \overline{B})^T \Psi^{(k+1)} = \overline{B}^T \Psi^{(k)} \tag{2.5}$$

$$\omega_{k+1} = \frac{(\Psi^{(k+1)}, \overline{A}\Phi^{(k+1)})}{(\Psi^{(k+1)}, \overline{B}\Phi^{(k+1)})}. \tag{2.6}$$

The iteration cycle is started with the guessed eigenvalue produced by the global method, ω_0, and an assumed but arbitrary smooth profile for the eigenfunction $\Phi(0)$ and its adjoint $\Psi(0)$. The algorithm converges cubically for the eigenvalue, but the eigenfunction converges at the square root of this value, as stated by Hackbusch (1985). In Figures 2 and 3, we present the temperature and velocity components of

the eigenfunction obtained after local computation for the adiabatic wall flat plate at Mach 4.5. The corresponding eigenvalue (for the most unstable second mode) is

$$\omega = 2.04706 + 0.02283i.$$

We see that the fluctuations are not restricted to the vicinity of the wall and that significant gradients appear up to a distance of several δ^* from the wall.

2.2. Modifications made to COSAL

2.2.1 General strategy

As noted in the previous section, the emphasis was originally on guaranteeing the accuracy of the computation of the *eigenvalue*. Concerns about accuracy in the *eigenfunctions* are rarely found in the literature. Most authors base their convergence criteria on the eigenvalue and present the eigenfunctions as a by-product of the eigenvalue calculation. However, to use the results of linear stability analysis as input to direct numerical simulations of transition, we need accurate eigenfunctions and must, therefore, be concerned with the convergence of the eigenfunctions as well the eigenvalue. Our assumption is that complete convergence of the eigenmode problem is only achieved when both quantities are converged.

The main goal of this work is to improve the accuracy of the calculation of the most unstable eigenfunction by the local procedure. This required improvement in three different areas:

- use of improved grids in the mean flow calculation;
- abandonment of inverse Rayleigh iteration in favor of a Newton method in the local calculations;
- implementation of an adaptive grid algorithm in local calculations.

Our objective is to keep the number of mesh points as low as possible (no more than a few hundred) as the profiles obtained from the local computation will be used in a simulation code that is much more expensive to run than the stability code itself. These refinements are not needed in the global calculation which are only required to provide a reasonable *estimate* of the most-unstable eigenvalue.

2.2.2 New grids for the mean flow computation

By considering the truncation error inherent in finite-difference approximations, Vinokur (1983) proposed new grid stretching functions based on the inverse hyperbolic sine. Using these stretching functions for the generation of mean flow grids yields better accuracy for a given number of grid points than is obtained with an exponentially-stretched mesh. We studied the effect of the new mesh generation scheme on profiles prior to using them in local calculations.

2.2.3 Newton method for local computations

The original local eigenvalue convergence process (inverse Rayleigh iteration) has been changed to Newton iteration which yields the same accuracy for both the eigenvalue and the eigenfunction.

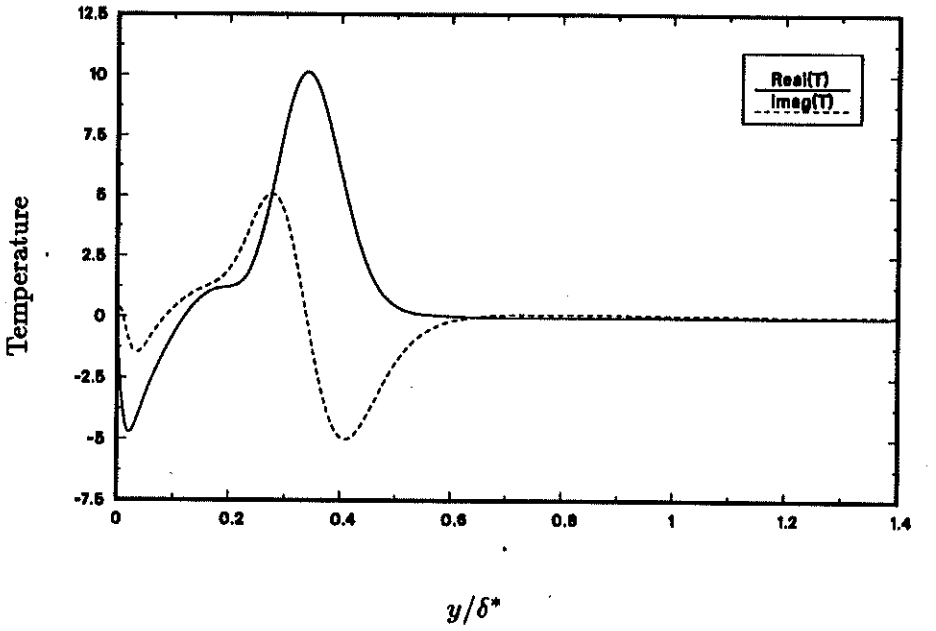

FIGURE 2. Temperature Eigenfunction for a Flat Plate, second mode $\alpha = 2.25$, $M_\infty = 4.5$, $Pr = .7$, $Re_{\delta^*} = 8000$.

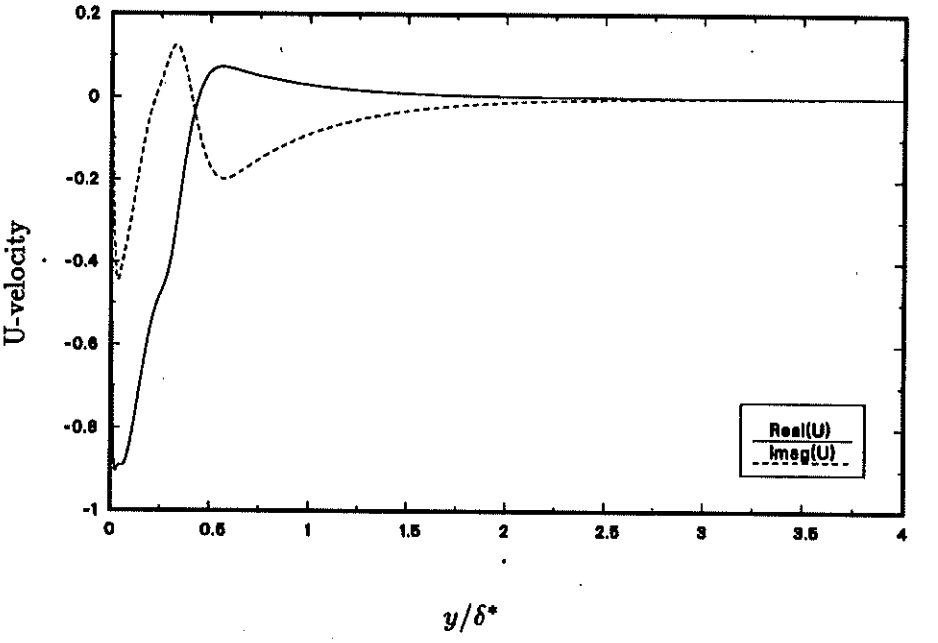

FIGURE 3. Streamwise Velocity Eigenfunction for a Flat Plate, second mode $\alpha = 2.25$, $M_\infty = 4.5$, $Pr = .7$, $Re_{\delta^*} = 8000$.

In symbolic form, the linear disturbances satisfy the system of ordinary differential equations which can be written

$$L\Phi = H, \tag{2.7}$$

where Φ is the five-component vector defined by

$$\{\hat{u}, \hat{v}, \hat{p}, \hat{T}, \hat{w}\}^T.$$

The boundary conditions for Eq. (2.7) are

$$\begin{cases} y = 0; & \phi_1 = \phi_2 = \phi_4 = \phi_5 = 0 \\ y \to \infty; & \phi_1, \phi_2, \phi_4, \phi_5 \to 0. \end{cases} \tag{2.8}$$

For the local eigenvalue problem, Eqs. (2.7) are a block-tridiagonal system which is solved using LU factorization. As Eq. (2.7) is homogeneous, in order to avoid a trivial solution, one inhomogeneous boundary conditions is imposed at the wall. Specifically, as proposed by Malik (1990), the boundary condition $\phi_1(0) = 0$ is replaced by $\phi_3(0) = 1$. This is equivalent to normalizing the eigenfunction so that the value of the pressure perturbation at the wall is unity. Since the pressure does not vanish at the wall, this condition is appropriate. See the discussion in Malik (1990) for other possible normalizations. A non-trivial solution may now be obtained if $\omega = \omega_0$, the correct eigenvalue. Newton's method is then used to iterate on ω so that the missing boundary condition $\phi_1(0) = 0$ is satisfied. After a solution Φ is obtained using the estimated value of ω_0, the correction $\Delta\omega$ is determined from the linearized equation

$$\phi_1(0) + \frac{\partial \phi_1(0)}{\partial \omega} \Delta\omega = 0, \tag{2.9}$$

where $\phi_1(0)$ is known from the solution Φ just computed; $\partial\phi_1(0)/\partial\omega$ is obtained by solving

$$L\frac{\partial \Phi}{\partial \omega} = -\frac{\partial L}{\partial \omega}\Phi. \tag{2.10}$$

The process is repeated until $\phi_1(0)$ vanishes within a preassigned tolerance.

We ran the local computation with this numerical method on the grid described above and the eigenvalue kept with 10 significant digits of accuracy. At the same time, the accuracy of both the eigenvalue and the eigenfunctions were checked by regularly doubling the number of mesh points. Figure 4 presents the absolute error in the real and imaginary parts of the frequency versus the number of grid points. The slopes of the two lines confirm the second order accuracy of the method. Now consider the eigenfunctions. Figure 5 presents the absolute error history for the imaginary part of the temperature eigenfunction as the number of mesh points is increased. After a minimum number of mesh points sufficient to capture the structure of the eigenfunction is reached, the local maxima in the error in the temperature eigenfunction decrease at the same rate as the eigenvalue error shown

in Figure 4. With one thousand mesh points, we now obtain an accuracy of 10^{-3} compared to 5.10^{-2} with the inverse Rayleigh iteration.

2.2.4 Grid works

The grids used in linear stability analysis have a significant effect on the results. As we are using a second order accurate numerical scheme, the grids need to be very smooth. Following the grid trials for the mean flow, we tried the Vinokur grids in the local calculations. Besides being smooth, they allow considerable control of the distribution of the grid points, which is not the case with exponentially stretched grids. The slopes at the wall and free-stream can be specified. It is also possible to join different Vinokur grids while retaining smoothness at the junctions. Finally, the number of grid points does not affect the shape of the grid. Several Vinokur grids were tried in the local computation. The resulting changes in the eigenvalue ω is an indicator of how sensitive these computations are to the grid. This naturally led us to implement an adaptive grid algorithm controlled by the error in the eigenfunctions.

Numerous adaptive grid methods are available. Most of them can be divided into two categories: displacement methods and refinement methods. The first type used a fixed number of mesh points and the adaptation consists of *moving* the mesh points from low-gradient regions to high-gradient regions. The second type starts with a coarse mesh and *adds* points in high-gradient regions. In an attempt to minimize the number of mesh-points, we first tried the displacement method based on a error equidistribution variational process proposed by Eiseman (1987). Although promising at low Mach number, this method was not able to handle the very steep gradients in the hypersonic second mode eigenfunctions, especially in the temperature eigenfunction. The main cause was the inability of the algorithm to maintain the proper smoothness of the grid.

It was then decided to develop a grid-refinement method that would maintain the required mesh smoothness. The algorithm is defined by the following steps :

(i) The initial grids, G_0 and G_1, are Vinokur grids of 41 and 81 points;

(ii) The eigenfunctions computed on G_i and G_{i-1} are compared and estimates of the error are constructed. Refinement intervals are introduced where the solution error is greater than a specified ϵ;

(iii) The number of points on each refinement interval is doubled by adding the mid-points of the old grid;

(iv) Smooth connection between new grids and the old ones is assured by a data-passing scheme;

(v) repeat steps *(ii)* - *(iv)* until no more refinement intervals are found.

The main difficulty was the choice of interpolation method. To avoid the wiggles that appear with B-spline interpolation, we used the interpolation method proposed by Akima (1970). This method is based on piecewise cubic polynomials. The slope is determined using a second-order geometric rule which leads to very "natural looking" and smooth grids. The junctions between the old parts of the grid and new the parts generated with Akima's method were made from fifth-order B-splines.

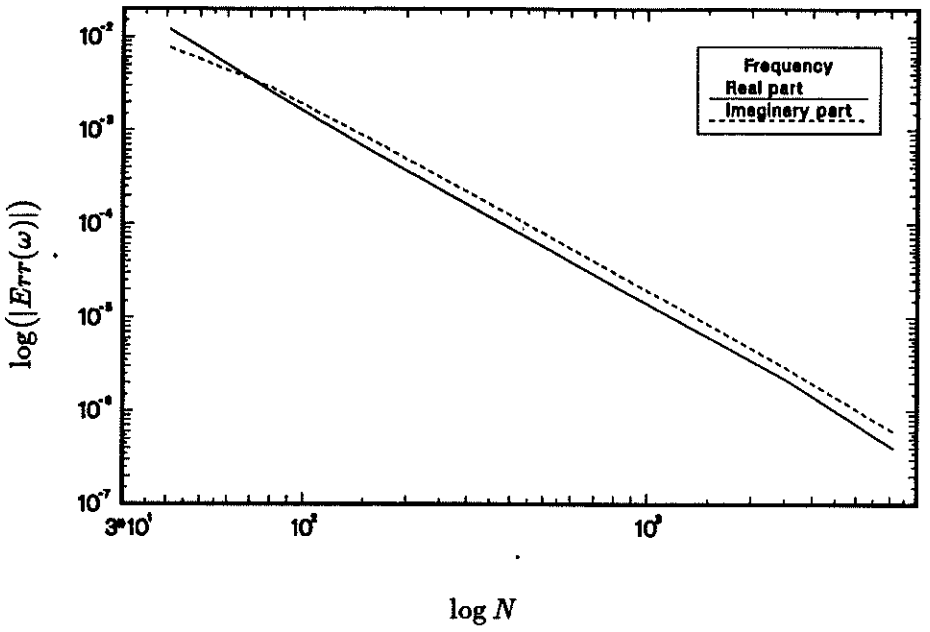

FIGURE 4. Convergence of the Most-Unstable Eigenvalue, second mode $\alpha = 2.25$, $M_\infty = 4.5$, $Pr = .7$, $Re_{\delta^*} = 8000$; $Err = |absolute \quad error|$.

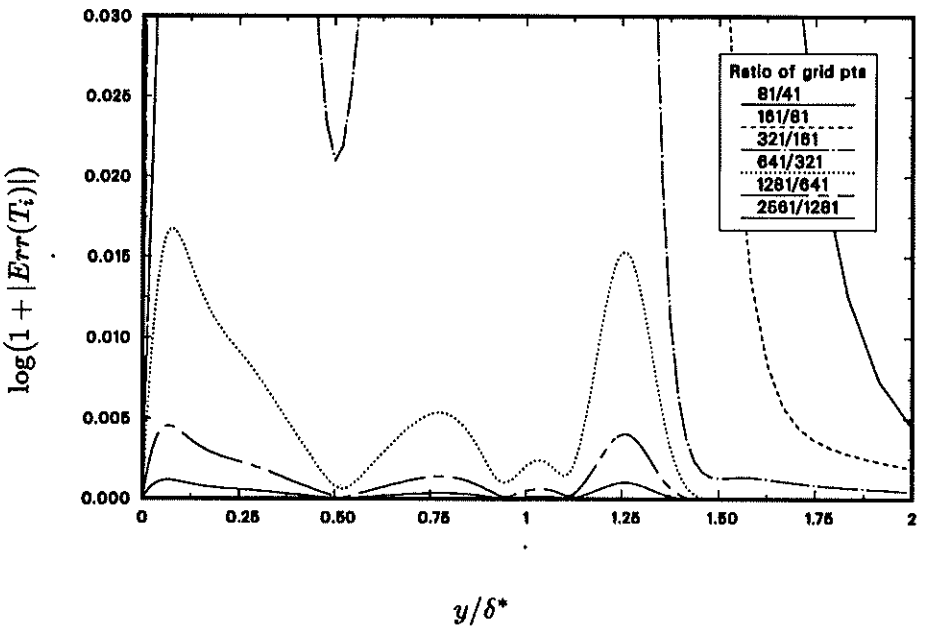

FIGURE 5. Convergence of the imaginary part of the second mode $\alpha = 2.25$, $M_\infty = 4.5$, $Pr = .7$, $Re_{\delta^*} = 8000$.

This new method was first tested on a Mach 0.5 case for which the eigenfunctions are smooth. The iteration process converged rapidly (in less than 10 iterations) for $\epsilon \approx 10^{-5}$. For lower values of ϵ, the first 10 iterations simply doubled the points everywhere thus producing a huge number of grid points. This demonstrated the need for *early capture of the significant gradients*. Consequently, to obtain very low error (say $\epsilon \approx 10^{-7}$), we first need to converge for an intermediate value of ϵ and then restart the process on the resultant grid with a lower value of ϵ.

This process is illustrated by Figure 6. We see the capture of three high-gradient regions during the refinement process. Convergence to $\epsilon = 10^{-4}$ is obtained on the sixth grid. Iteration continues with $\epsilon = 10^{-5}$ until convergence is obtained on the ninth grid. Figures 7 and 8 show the accelerated convergence of the eigenfunctions due to the capture of those layers.

In Figure 7 we plot the difference between the converged eigenfunction obtained with $\epsilon = 10^{-4}$ and the solutions on the intermediate grids. Note the big jump between the last two results. This should be compared to the convergence history in Figure 5. In Figure 8, we compare the converged solution for $\epsilon = 10^{-5}$ to the solutions obtained on all intermediate grids, including those of Figure 7. It is interesting to note that these "jumps" in the eigenfunction are accompanied by corresponding "jumps" in the eigenvalue. The eigenvalues obtained on the various grids of Figures 7 and 8 are given in Table 1.

Grid (# of points)	ω
1 (41)	$6.224E-2+2.727E-3i$
2 (81)	$6.224E-2+2.766E-3i$
3 (161)	$6.224E-2+2.775E-3i$
4 (321)	$6.224E-2+2.933E-3i$*
5 (467)	$6.224E-2+2.933E-3i$
6 (823)	$6.242E-2+4.627E-3i$*
7 (1081)	$6.242E-2+4.627E-3i$
8 (1485)	$6.242E-2+4.627E-3i$
9 (2791)	$6.342E-2+4.266E-3i$*

TABLE 1. Most Unstable Eigenvalues found by Local Grid Adaptation Method, (* denotes new structure captured).

To verify that the solution is not jumping from one eigenvalue to another, we ran the global calculation using the same adapted grid and found the same most unstable eigenvalue. This confirms our assumption that complete convergence of the eigenmode problem is achieved only when both the eigenvalue and the eigenfunction are converged.

Running this adaptive process for the Mach 4.5 case led to convergence in eleven iterations to $\epsilon = 10^{-4}$. The adaptive process was not able to continue because the B-spline caused increasing wiggles in the grid. A new adaptive method is being

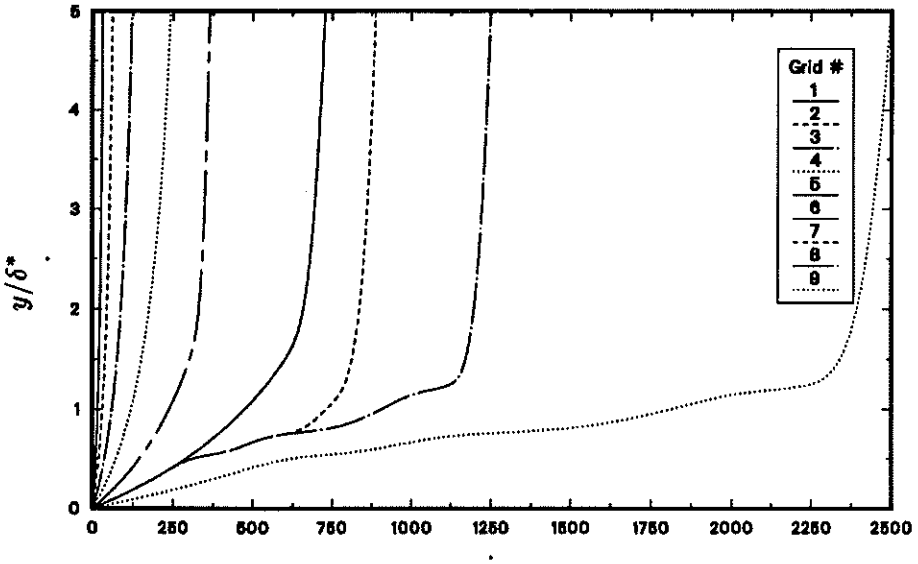

FIGURE 6. Grid Adaptation in Local Calculation, $M_\infty = 0.5$, $Re_{\delta^*} = 2000$.

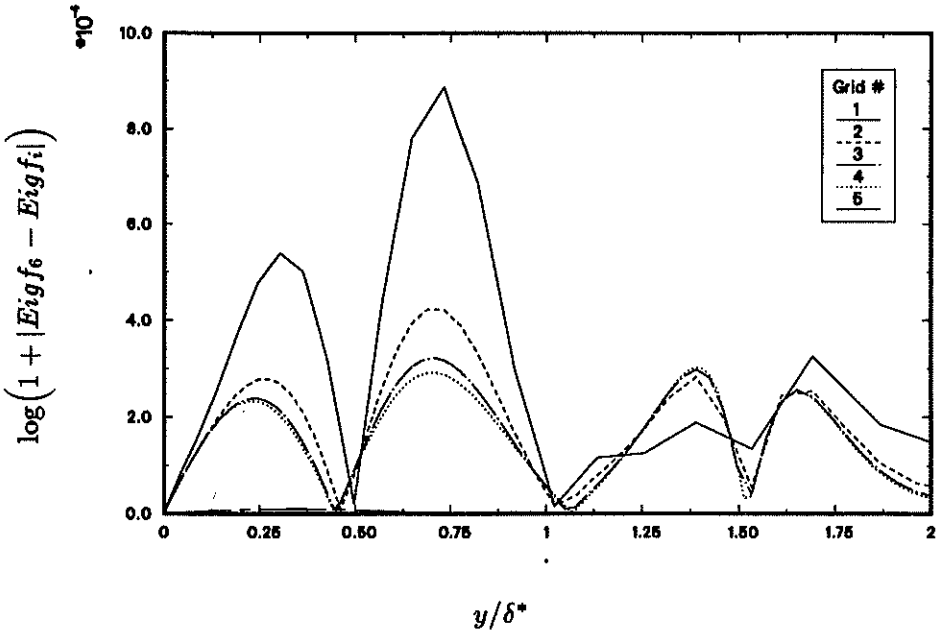

FIGURE 7. Eigenfunction Convergence on Adapted Grids 1 to 6, $M_\infty = 0.5$, $Re_{\delta^*} = 2000$.

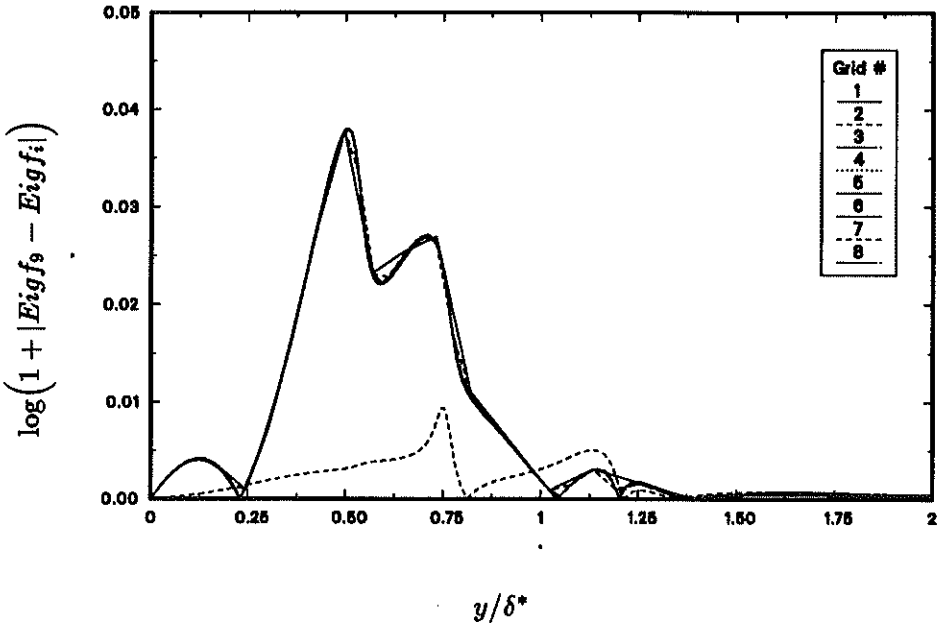

FIGURE 8. Eigenfunction Convergence on Adapted Grids 1 to 9, $M_\infty = 0.5$, $Re_{\delta^*} = 2000$.

developed to avoid that phenomenon (see future plans). The final number of grid-points was about 1500, which is too many to use in direct numerical simulation. We reduced this number by taking every fifth point to create a 300 point grid which has the *same structure* as the 1500 point one. Using these eigenfunctions in the simulation code improved its performance. Several CPU hours were saved on a CRAY Y-MP because the simulation code did not have to adjust the eigenfunction shape before being able to simulate their growth.

These encouraging results show that the eigenmode problem is now being solved accurately. During this study, it was found that accuracy requires capture of the relevant *physical layers* in the temperature and velocity profiles. Once this is realized, it becomes possible to ensure convergence of the eigenvalue and eigenfunction with a relatively small number of grid points. Perhaps just as importantly, the adaptive technique yields as a by-product a grid which is nearly ideal for the direct simulation of transition.

3. Future plans

• In order to decrease the number of points needed to obtain the required accuracy, a fourth order compact difference scheme will be implemented in the local calculations. This will also reduce the smoothing requirements for the grids used in these calculations.

• Because the resolution required for a given error is not uniform and is unknown

in advance, an adaptive grid will probably be required in the simulations of transition. Although they are not really needed in the instability calculations which are only one-dimensional, this is an ideal place to test them prior to their installation in the multi-dimensional code. The results reported here are an encouraging first step in that direction.

• As the results of the stability code are to be used as input to the three dimensional code and small changes in the eigenfunctions can result in large effects in the transition simulation, it is important that the grids used in the two calculations be as similar as possible. We, therefore, intend to allow the DNS code and the stability code to communicate directly to ensure consistency.

Acknowledgement

E. Guilyardi was supported in part by Dassault Aviation, France.

REFERENCES

AKIMA, H. 1970 A New Method of Interpolation and Smooth Curve Fitting Based on Local Procedures. *Journal of the Association for Computing Machinery.* **17.4**, 589,602.

CEBECI, T. & SMITH, A. M. O. 1974 *Analysis of turbulent boundary layers.* Academic Press, N. Y.

EISEMAN, P. R. 1987 Adaptive Grid Generation. *Computer Methods in Applied Mechanics and Engineering.* **64**, 321,376; North-Holland.

ERLEBACHER, G. & HUSSAINI, M. Y. 1990 Numerical Experiments in Supersonic Boundary-Layer Stability. *Physics of Fluids.* **A2**, 94-104.

HACKBUSCH, W. 1985 *Multi-grid methods and applications.* Springer - Verlag, Berlin.

MACK, L. M. 1984 Special Course on Stability and Transition of Laminar Flow. *AGARD Report 709.*

MALIK, M. R. 1982 Finite-Difference Solution of the Compressible Stability Eigenvalue Problem. *NASA Contractor Report NAS1-16572.*

MALIK, M. R. 1990 Numerical Methods for Hypersonic Boundary Layer Stability. *Journal of Computational Physics.* **86**, 376-413.

RESHOTKO, E. 1976 Boundary-Layer Stability and Transition. *Annual Review of Fluid Mechanics.* **8**, 311-349.

VAN DER VEGT, J. J. W. & FERZIGER, J. H. 1990 Methods for direct simulation of transition in hypersonic boundary layers. In *CTR Annual Research Briefs -1990.*

VINOKUR, M. 1983 On One-Dimensional Stretching Functions for Finite Difference Calculations. *Journal of Computational Physics.* **50**, 215-234.

WILKINSON, J. H. 1965 *The Algebraic Eigenvalue Problem.* Oxford University Press, London.