High Order Finite Element Discretization of the Compressible Euler and Navier-Stokes Equations

J.S. Wong, D.L. Darmofal, and J. Peraire
Department of Aeronautics and Astronautics,
Massachusetts Institute of Technology, Cambridge, MA 02139

Abstract

We present a high order accurate streamline-upwind/Petrov-Galerkin (SUPG) algorithm for the solution of the compressible Euler and Navier-Stokes equations. The flow equations are written in terms of entropy variables which result in symmetric flux Jacobian matrices and a dimensionally consistent Finite Element discretization. We show that solutions derived from quadratic element approximation are of superior quality next to their linear element counterparts. We demonstrate this through numerical solutions of both classical test cases as well as examples more practical in nature.

Key Words: Euler and Navier-Stokes equations, Petrov-Galerkin, entropy variables, symmetric flux jacobian matrices, high order accuracy

1 Introduction

This paper centers upon a high-order accurate, stabilized, finite element method for the numerical solution of the compressible Euler and Navier-Stokes equations. The SUPG finite element method for compressible flow simulations was initially developed and analyzed by Hughes *et al.* [3, 4, 5, 2] and has since gained significant popularity. Its relation to multidimensional upwinding was elucidated in [9, 10] and higher order implementations for inviscid flows were presented in [6]. In [1], the SUPG algorithm was extentded to cover the simulation of near-incompressible flows by employing a stabilization matrix which exhibits proper scaling over the entire range of Mach numbers. Here, we focus on the higher order implementation of the algorithm developed in [1] for invicid and viscous flows.

2 Compressible flow governing equations

We start from the time dependent two dimensional compressible Euler equations in conservation form

$$\mathbf{U}_{,t} + (\mathbf{F} - \mathbf{F}^v)_{1,1} + (\mathbf{F} - \mathbf{F}^v)_{2,2} = 0, \tag{1}$$

where

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho u_1 \\ \rho u_2 \\ \rho E \end{bmatrix}, \quad \mathbf{F}_1 = \begin{bmatrix} \rho u_1 \\ \rho u_1^2 + p \\ \rho u_1 u_2 \\ u_1(\rho E + p) \end{bmatrix}, \quad \mathbf{F}_2 = \begin{bmatrix} \rho u_2 \\ \rho u_1 u_2 \\ \rho u_2^2 + p \\ u_2(\rho E + p) \end{bmatrix}.$$

and

$$\mathbf{F}_1^v = \begin{bmatrix} 0 \\ \tau_{11} \\ \tau_{12} \\ u_1\tau_{11} + u_2\tau_{12} + q_1 \end{bmatrix}, \quad \mathbf{F}_2^v = \begin{bmatrix} 0 \\ \tau_{21} \\ \tau_{22} \\ u_1\tau_{21} + u_2\tau_{22} + q_2 \end{bmatrix}.$$

In the above expressions, ρ is the density; $\mathbf{u} = [u_1, u_2]^T$ is the velocity vector; E is the specific total energy; p is the pressure; and the comma denotes partial differentiation (e.g. $\mathbf{U}_{,t} = \partial\mathbf{U}/\partial t$, the partial derivative with respect to time, $\mathbf{F}_{i,j} = \partial\mathbf{F}_i/\partial x_j$, the partial derivative with respect to the j-th spatial coordinate). The system of equations is closed once the pressure is related to the problem variables through the equation of state, $p = (\gamma - 1)\rho e$, where $e = E - |\mathbf{u}|^2/2$, is the internal energy. Here, γ is the ratio of specific heats and μ is the absolute viscosity, both of which are assumed to be constant. Following the usual assumptions:

$$\tau_{11} = 2\frac{\mu}{Re}\frac{\partial u_1}{\partial x_1} - \frac{2}{3}\frac{\mu}{Re}\left(\frac{\partial u_1}{\partial x_1} + \frac{\partial u_2}{\partial x_2}\right)$$

$$\tau_{12} = \tau_{21} = \frac{\mu}{Re}\left(\frac{\partial u_1}{\partial x_2} + \frac{\partial u_2}{\partial x_1}\right)$$

$$\tau_{22} = 2\frac{\mu}{Re}\frac{\partial u_2}{\partial x_2} - \frac{2}{3}\frac{\mu}{Re}\left(\frac{\partial u_1}{\partial x_1} + \frac{\partial u_2}{\partial x_2}\right)$$

and

$$q_1 = -\frac{\mu}{RePr}\frac{\partial T}{\partial x_1}, \quad q_2 = -\frac{\mu}{RePr}\frac{\partial T}{\partial x_2}.$$

We will assume that all the above quantities have been non-dimensionalized using reference, or free stream, values for density ρ_*, velocity u_*, and length L. Thus, the dimensional variables, denoted with an overbar, are related to the non-dimensional variables introduced above as

$$\rho = \frac{\bar{\rho}}{\rho_*}, \quad u_i = \frac{\bar{u}_i}{u_*},\ i = 1,2, \quad p = \frac{\bar{p}}{\rho_* u_*^2}, \quad E = \frac{\bar{E}}{u_*^2}, \quad \mu = \frac{\bar{\mu}}{\mu_*}, \quad x_i = \frac{\bar{x}_i}{L},\ i = 1,2, \quad \text{and} \quad t = \frac{u_*\bar{t}}{L}.$$

We note that the equation system (1) can be written as

$$\mathbf{U}_{,t} + \mathbf{A}_1\mathbf{U}_{,1} + \mathbf{A}_2\mathbf{U}_{,2} = (\mathbf{K}_{11}\mathbf{U}_{,1})_{,1} + (\mathbf{K}_{12}\mathbf{U}_{,2})_{,1} + (\mathbf{K}_{21}\mathbf{U}_{,1})_{,2} + (\mathbf{K}_{22}\mathbf{U}_{,2})_{,2}, \tag{2}$$

where the Jacobian matrices $\mathbf{A}_i = \mathbf{F}_{i,\mathbf{U}}$, $i = 1, 2$, are unsymmetric but have real eigenvalues and a complete set of eigenvectors. $\mathbf{K}_{ij} = \mathbf{F}^v_{i,\mathbf{U}_{,j}}$ are the viscous flux jacobians. The above equation may be symmetrized through a change of variables, for details, we refer the reader to [5, 7].

2.1 Entropy variables

We seek a new set of variables $\mathbf{V}$, called entropy variables, such that the change $\mathbf{U} = \mathbf{U}(\mathbf{V})$ applied to (1) yields the transformed system

$$\mathbf{A}_0\mathbf{V}_{,t} + \tilde{\mathbf{A}}_1\mathbf{V}_{,1} + \tilde{\mathbf{A}}_2\mathbf{V}_{,2} = (\tilde{\mathbf{K}}_{11}\mathbf{V}_{,1})_{,1} + (\tilde{\mathbf{K}}_{12}\mathbf{V}_{,2})_{,1} + (\tilde{\mathbf{K}}_{21}\mathbf{V}_{,1})_{,2} + (\tilde{\mathbf{K}}_{22}\mathbf{V}_{,2})_{,2}, \tag{3}$$

where $\mathbf{A}_0 = \mathbf{U}_{,\mathbf{V}}$ is symmetric positive definite, and $\tilde{\mathbf{A}}_i = \mathbf{A}_i\mathbf{A}_0 = \mathbf{F}_{i,\mathbf{V}}$, $i = 1, 2$, are symmetric.

Following [7], we introduce a scalar entropy function $H(\mathbf{U}) = -\rho g(s)$, where s is the non-dimensional entropy $s = \ln(p/\rho^\gamma)$. The required change of variables is obtained by taking

$$\mathbf{V} = H^T_{,\mathbf{U}} = \frac{g'}{e}\begin{bmatrix} e(\gamma - g/g') - |\mathbf{u}|^2/2 \\ u_1 \\ u_2 \\ -1 \end{bmatrix}. \tag{4}$$

The conditions $g' > 0$ and $g''/g' < \gamma^{-1}$, ensure that $H(\mathbf{U})$ is a convex function and therefore $\mathbf{A}_0^{-1} = \mathbf{V}_{,\mathbf{U}} = H_{,\mathbf{UU}}$, and $\mathbf{A}_0$, are symmetric positive definite. Furthermore, if we chose $g(s) = \alpha s$ then we insure the matrix $\bar{\mathbf{K}}$

$$\bar{\mathbf{K}} = \begin{bmatrix} \tilde{\mathbf{K}}_{11} & \tilde{\mathbf{K}}_{12} \\ \tilde{\mathbf{K}}_{21} & \tilde{\mathbf{K}}_{22} \end{bmatrix}$$

is symmetric as well as positive semi-definite.

3 Variational formulation for the steady state problem

We now consider the compressible steady problem in conservation form expressed in terms of symmetrizing variables. The conservative form of the equations is taken to be the starting point because we are ultimately interested in an algorithm that can be used over the whole range of speed regimes, including situations were the solution may contain discontinuitites. The problem is defined in a domain Ω with boundary Γ by

$$(\mathbf{F} + \mathbf{F}^v)_1(\mathbf{V})_{,1} + (\mathbf{F} + \mathbf{F}^v)_2(\mathbf{V})_{,2} = 0 \quad \text{in} \quad \Omega, \tag{5}$$

$$\tilde{\mathbf{A}}_n^-\mathbf{V} = \tilde{\mathbf{A}}_n^-\mathbf{g} \quad \text{on} \quad \Gamma\backslash\Gamma_a, \tag{6}$$

$$\mathbf{F}^v \cdot \mathbf{n} = \mathbf{f} \quad \text{on} \quad \Gamma_a \tag{7}$$

For simplicity, the domain boundary is assumed to be made up of a solid wall Γ_a, and a computational far field boundary $\Gamma\backslash\Gamma_a$. In (6, 7), $\mathbf{n} = [n_1, n_2]^T$ is the outward unit normal vector to Γ, and $\tilde{\mathbf{A}}_n = \mathbf{A}_n\mathbf{A}_0$, $\mathbf{A}_n = \mathbf{A}_1 n_1 + \mathbf{A}_2 n_2$. Finally, $\tilde{\mathbf{A}}_n^- = \mathbf{A}_n^-\mathbf{A}_0$, and $\mathbf{A}_n^-$ denotes the negative definite part of $\mathbf{A}_n$.

Let the spatial domain Ω, be discretized into non-overlapping elements T_e, such that $\Omega = \bigcup T_e$, and $T_e \bigcap T_{e'} = \emptyset$, $e \neq e'$. We consider the space of functions $\mathcal{V}_h$, defined over the discretization and consisting of the continuous functions which are piecewise linear over each element

$$\mathcal{V}_h = \{\mathbf{W} \mid \mathbf{W} \in (C^0(\Omega))^4,\ \mathbf{W}|_{T_e} \in (\mathcal{P}_k(T_e))^4,\ \forall\, T_e \in \Omega\}.$$

The SUPG algorithm can then be written as: Find $\mathbf{V}_h \in \mathcal{V}^h$ such that for all $\mathbf{W} \in \mathcal{V}^h$,

$$B(\mathbf{V}_h, \mathbf{W})_{gal} + B(\mathbf{V}_h, \mathbf{W})_{supg|gls} + B(\mathbf{V}_h, \mathbf{W})_{bc} = 0, \tag{8}$$

where the forms $B(\cdot,\cdot)_{gal}$, $B(\cdot,\cdot)_{supg}$ and $B(\cdot,\cdot)_{bc}$ account for the Galerkin, SUPG stabilization, and boundary condition terms respectively, and are defined as

$$B(\mathbf{V}, \mathbf{W})_{gal} = \int_\Omega (-\mathbf{W}_{,1} \cdot (\mathbf{F} - \mathbf{F}^v)_1(\mathbf{V}) - \mathbf{W}_{,2} \cdot (\mathbf{F} - \mathbf{F}^v)_2(\mathbf{V}))\, d\Omega, \tag{9}$$

$$\begin{aligned} B(\mathbf{V}, \mathbf{W})_{supg} = \int_\Omega (\tilde{\mathbf{A}}_1 \mathbf{W}_{,1} + \tilde{\mathbf{A}}_2 \mathbf{W}_{,2}) \cdot \boldsymbol{\tau}\, (\tilde{\mathbf{A}}_1 \mathbf{V}_{,1} + \tilde{\mathbf{A}}_2 \mathbf{V}_{,2} - (\mathbf{K}_{11}\mathbf{V}_{,1})_{,1} - (\mathbf{K}_{12}\mathbf{V}_{,2})_{,1} - \\ (\mathbf{K}_{21}\mathbf{V}_{,1})_{,2} - (\mathbf{K}_{22}\mathbf{V}_{,2})_{,2})\, d\Omega, \end{aligned} \tag{10}$$

$$\begin{aligned} B(\mathbf{V}, \mathbf{W})_{gls} = \int_\Omega ((\tilde{\mathbf{A}}_1 \mathbf{W}_{,1} + \tilde{\mathbf{A}}_2 \mathbf{W}_{,2} - (\mathbf{K}_{11}\mathbf{W}_{,1})_{,1} - (\mathbf{K}_{12}\mathbf{W}_{,2})_{,1} - \\ (\mathbf{K}_{21}\mathbf{W}_{,1})_{,2} - (\mathbf{K}_{22}\mathbf{W}_{,2})_{,2}) \cdot \boldsymbol{\tau}\, (\tilde{\mathbf{A}}_1 \mathbf{V}_{,1} + \tilde{\mathbf{A}}_2 \mathbf{V}_{,2} - (\mathbf{K}_{11}\mathbf{V}_{,1})_{,1} - (\mathbf{K}_{12}\mathbf{V}_{,2})_{,1} - \\ (\mathbf{K}_{21}\mathbf{V}_{,1})_{,2} - (\mathbf{K}_{22}\mathbf{V}_{,2})_{,2})\, d\Omega, \end{aligned} \tag{11}$$

and

$$B(\mathbf{V}, \mathbf{W})_{bc} = \int_{\Gamma \setminus \Gamma_a} \mathbf{W} \cdot (\mathbf{F}_{ff} + \mathbf{F}^v)(\mathbf{V}, \mathbf{g}; \mathbf{n})\, ds. + \int_{\Gamma_a} \mathbf{W} \cdot \mathbf{F}^v(\mathbf{V}, \mathbf{f}; \mathbf{n})\, ds. \tag{12}$$

where $\boldsymbol{\tau}$ is the stabilization matrix. The numerical flux function on the far field boundary $\mathbf{F}_{ff}$, is defined by

$$\mathbf{F}_{ff}(\mathbf{V}_-, \mathbf{V}_+; \mathbf{n}) = \frac{1}{2}(\mathbf{F}_n(\mathbf{V}_-) + \mathbf{F}_n(\mathbf{V}_+)) - \frac{1}{2}|\mathbf{A}_n(\mathbf{V}^*(\mathbf{V}_-, \mathbf{V}_+))|(\mathbf{V}_+ - \mathbf{V}_-).$$

Here, $|\mathbf{A}_n(\mathbf{V})| = \mathbf{A}_n^+(\mathbf{V}) - \mathbf{A}^-(\mathbf{V})$ is the absolute value of $\mathbf{A}_n$ evaluated at $\mathbf{V}^*$, and $\mathbf{V}^*(\mathbf{V}_+, \mathbf{V}_-)$, is the an average between the states $\mathbf{V}^+$ and $\mathbf{V}^-$. The average state, $\mathbf{V}^*$, is chosen to ensure the global stability of the algorithm [6]. For inviscid compuations, the viscous terms in the expressions above would of course, vanish. For viscous simulations, Dirichlet boundary conditions may replace portions of the boundary integral.

4 Definition of $\boldsymbol{\tau}$

The definition of $\boldsymbol{\tau}$ follows that of [1]. The following modification is made for viscous simulations:

$$\boldsymbol{\tau}^{-1} = {\boldsymbol{\tau}_i}^{-1} + {\boldsymbol{\tau}_v}^{-1}$$

where $\boldsymbol{\tau}_i$ is the stabilization defined in [1] and $\boldsymbol{\tau}_v$ is defined as:

$${\boldsymbol{\tau}_v}^{-1} = 2(\tilde{\mathbf{K}}_{11} + \tilde{\mathbf{K}}_{22})/{h_e}^2$$

5 Numerical results

In this section we present some numerical results that illustrate the performance of the proposed algorithm. Test problems were solved employing both linear and quadratic element approximations. For comparative purposes, the meshes used for all linear element approximations were obtained by subdividing each element of the corresponding quadratic element mesh into four linear elements. In this way, comparisons between P_1 and P_2 solutions involving the same number of nodes can be made. h_c thus represents the distance between two nodes in the meshes used in the numerical simulations presented herein.

5.1 Example 1: Rinleb flow

In this example, we consider a ringleb test case (an exact solution of the Euler equations, [13]). The error is computed in the L_2 entropy norm [1]. Both P_1 and P_2 element approximation achieved their respective optimal convergence rate of $O(h^2)$ and $O(h^3)$ respectively, as can be seen in figure 1.

5.2 Example 2: Flow over an airfoil

In this example, the proposed scheme was used to simulate the flow over NACA 0012 airfoil at a Mach number of 0.6, and at an angle of attack of 2°. In figure 2, the L_2 entropy deviation for both P_1 and P_2 simulations are presented. The quadratic element approximation results in a much lower level of entropy error than its linear element counterpart. The geometric singularity at the trailing edge of the airfoil requires a much finer discretization around that point relative to the rest of the mesh for both the linear and quadratic element approximations to achieve their optimal convegence rate. This is paticularly important for the P_2 approximation since the error away from the geometric singularity vanishes far quicker, rendering the trailing edge error as the dominant source of error for the numerical approximation.

5.3 Example 3: Flow over flat plate

In this example, we consider flow over a flat plate of unit length. The computational domain is $[-1.5, 1]\times[0, 1]$, with the leading edge of the plate at $(0, 0)$. The free stream Mach number is 0.5. The Reynolds number is raised from 8000 to 64000 in successive simulations while keeping the mesh unchanged. The results in the form of boundary layer thickness, $\delta_{99}(x = L)$, are plotted in figure 4. The quadratic element approximation yields results very close to that of the Blasius solution while the linear element approximation shows increasing error with rising Reynolds number. Further numerical tests have shown that it is possible to resolve the Blasius boundary layer with only two elements when quadratic element approximation is used.

6 Conclusion

A high-order accurate, stabilized finite element method for the solution of compressible Euler and Navier-Stokes equations has been presented. The advantages of quadratic element approximation over linear representation of solution were demonstrated through a number of test problems. In particular, these numerical tests have shown that higher-order approximation is significantly more effective in resolving viscous boundary layers.

References

[1] J.S. Wong, D.L. Darmofal and J. Peraire, *The Solution of the Compressible Euler Equations at Low Mach Numbers using a Stabilized Finie Element Algorithm*, in preparation 2000.

[2] F. Shakib, T.J.R. Hughes and Z. Johan *A new finite element formulation for computational fluid dynamics : X. The compressible Euler and Navier-Stokes equations*, Comp. Meth. in Appl. Mech. and Engnr., 89, 1991, pp. 141-219.

[3] T.J.R. Hughes and M. Mallet, *A new finite element formulation for computational fluid dynamics : III. The generalized streamline operator for multidimensional advective diffusive systems*, Comp. Meth. in Appl. Mech. and Engnr., 58, 1986, pp. 305-328.

[4] T.J.R. Hughes, L.P. Franca and G.M. Hulbert, *A new finite element formulation for computational fluid dynamics : VIII. The Galerkin Least-Squares method for advective diffusive equations*, Comp. Meth. in Appl. Mech. and Engnr., 73, 1989, pp. 173-189.

[5] T.J.R. Hughes, L.P.Franca and M. Mallet, *A new finite element formulation for computational fluid dynamics : I. Symmetric forms of the compressible Euler and Navier-Stokes equations and the second law of thermodynamics*, Comp. Meth. in Appl. Mech. and Engnr., 54, 1986, pp. 223-234.

[6] T.J. Barth, *Numerical methods for gasdynamic systems on unstructured meshes*, Lecture Notes in Computational Science and Engineering, 1998, pp. 195-284

[7] A. Harten, *On the symmetric for of systems of conservation laws with entropy*, Journal of Computational Physics, 49, 1983, pp. 151-164.

[8] P.L. Roe, *Approximate Riemann solvers, parameter vectors, and difference schemes*, J. Comp. Phys., 43, 1981, pp. 357-72.

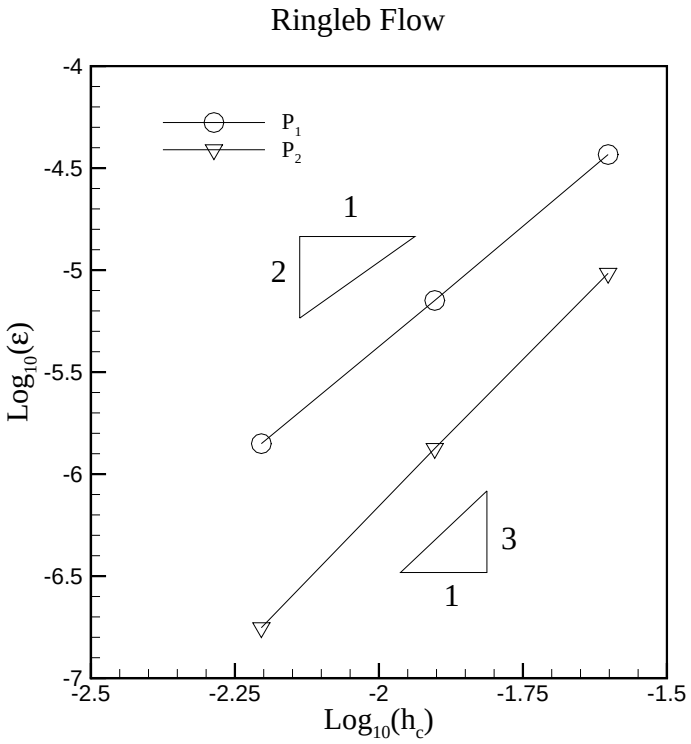
[9] J.C. Carette, *Adaptive Unstructured Mesh Algorithms and SUPG Finite Element Method for Compressible High Reynolds Number Flows*, PhD Thesis, von Karman Institute, 1997

[10] J.C. Carette, H. Deconinck, H. Paillere, and P.L. Roe, *Multidimensional upwinding: Its relation to finite elements*, Int. J. of Num Meth. Fluids, vol.20 pp935-955, 1995.

[11] D.R. Fokkema, *Subspace methods for linear, nonlinear and eigenproblems*, PhD Thesis, Utrech University, 1996.

[12] G.L.G. Sleijpen, H.A. van der Vorst and D.R. Fokkema, *BiCGstab(1) and other hybrid Bi-CG methods*, Numerical Algorithms, 7, 1994, pp.75-109.

[13] *Test Cases for Inviscid Flow Field Methods*, AGARD-AR-211, 7 Ancelle 92200, Neuilly sur Seine, France, 1985.



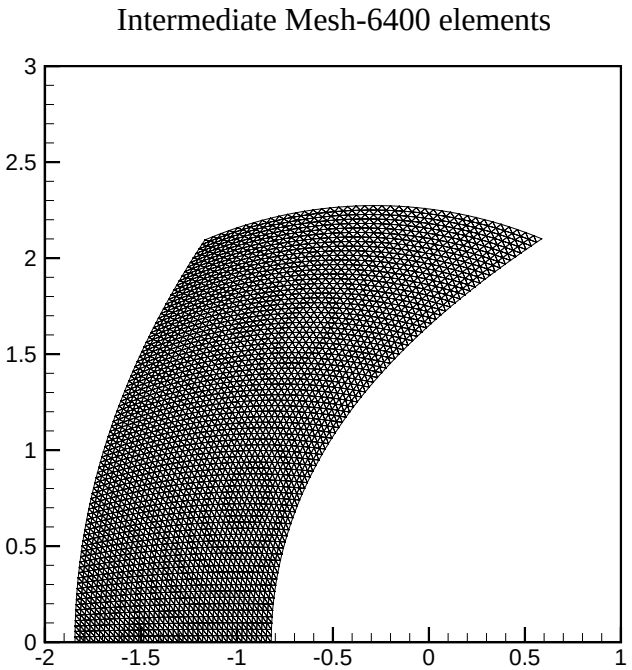
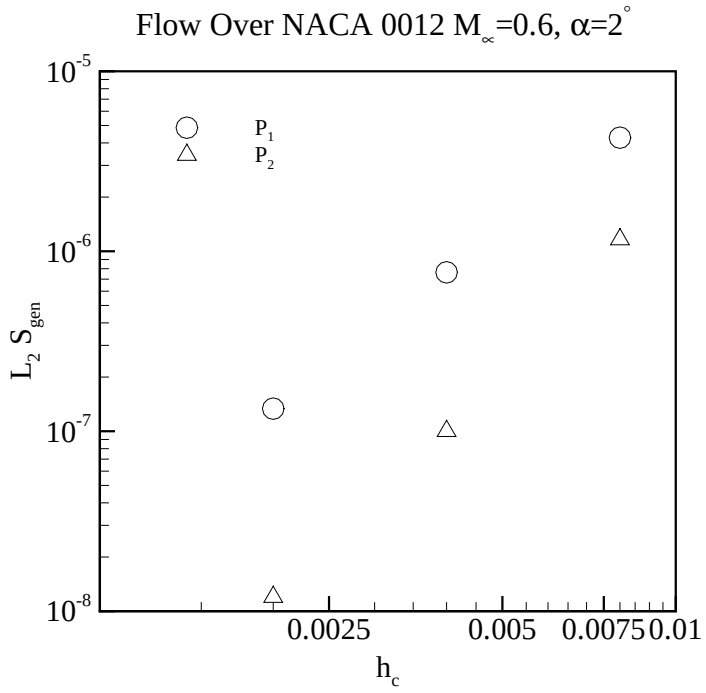


Figure 1: Top: L_2 entropy norm error of ringleb flow solution, Bottom: Computational mesh.



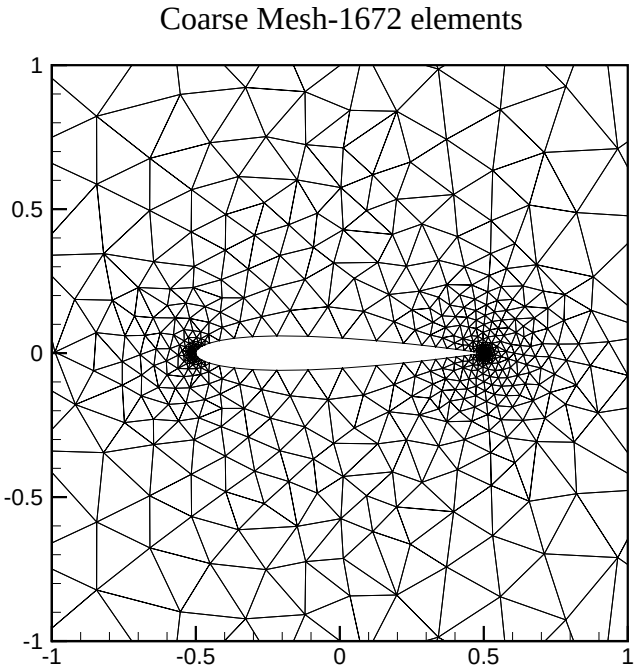


Figure 2: Top: L_2 entropy error of flow over NACA 0012 airfoil, Bottom: Computational mesh.

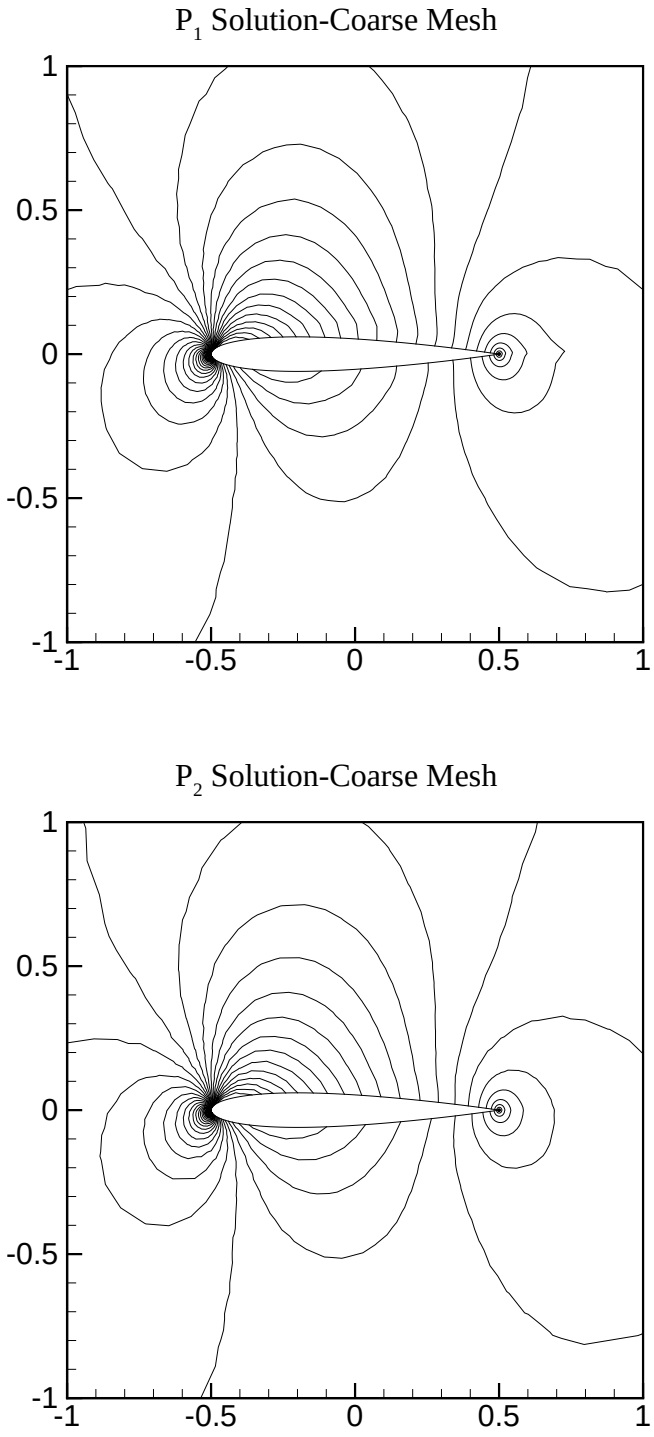
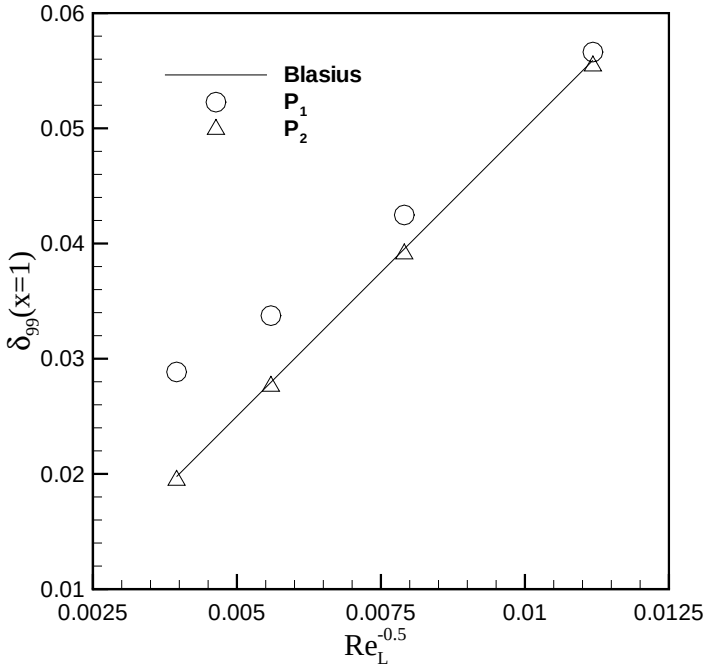


Figure 3: Mach contour of flow over NACA 0012 airfoil, $M_\infty = 0.6$, $\alpha = 2^\circ$



Mesh - Flow Over Flat Plate

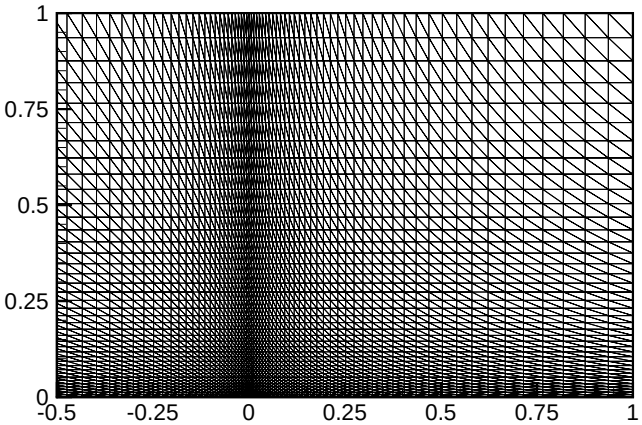


Figure 4: Top: Computed value of boundary layer thickness at $x = 1$, Bottom: Computational mesh.