Compressible flow SUPG parameters computed from element matrices

L. Catabriga\textsuperscript{1}, A. L. G. A. Coutinho\textsuperscript{2,\textsuperscript{*},\textsuperscript{†}} and T. E. Tezduyar\textsuperscript{3}

\textsuperscript{1}Department of Computer Science, Federal University of Espírito Santo, Brazil
\textsuperscript{2}Center for Parallel Computations and Department of Civil Engineering - COPPE
Federal University of Rio de Janeiro, Brazil
\textsuperscript{3}Team for Advanced Flow Simulation and Modeling (T\textsuperscript{*}AFSM) Mechanical Engineering and Materials Science, Rice University, U.S.A.

SUMMARY

We present, for the SUPG formulation of inviscid compressible flows with shocks, stabilization parameters defined based on the element-level matrices. These definitions are expressed in terms of the ratios of the norms of the matrices and take into account the flow field, the local length scales, and the time step size. Calculations of these stabilization parameters are straightforward and do not require explicit expressions for length or velocity scales. We compare the performance of these stabilization parameters, accompanied by a shock-capturing parameter introduced earlier, with the performance of a stabilization parameter introduced earlier, accompanied by the same shock-capturing parameter. We investigate the performance difference between updating the stabilization and shock-capturing parameters at the end of every time step and at the end of every non-linear iteration within a time step. We also investigate the influence of activating an algorithmic option that was introduced earlier, which is based on freezing the shock-capturing parameter at its current value when a convergence stagnation is detected. Copyright © 2005 John Wiley & Sons, Ltd.

KEY WORDS: Euler equations; compressible flow; finite elements; stabilization parameter

1. INTRODUCTION

The SUPG formulation for compressible flows was introduced, in the context of conservation variables, in Reference [1]. After that, several SUPG-like methods for compressible flows were developed, and a brief summary on that can be found in Reference [2]. One of such
approaches, introduced in Reference [3], is based on recasting the original SUPG formulation of compressible flows in entropy variables and supplementing that with a shock-capturing term. It was shown in Reference [4] that the SUPG formulation introduced in Reference [1], when supplemented with a similar shock-capturing term, is very comparable in accuracy to the one that was recast in entropy variables. In this paper we limit our attention to the SUPG formulation of inviscid compressible flows based on the conservation variables.

A stabilization parameter that is mostly known as ‘τ’ is embedded in the SUPG formulations. A brief reference to some of the τs proposed is also given in Reference [2]. One of the interesting ways of designing τs is based on, as it was proposed in Reference [5] and generalized in Reference [6], defining a separate τ for each degree-of-freedom (i.e. for each equation), leading to a matrix form of the τ. While we recognize that such an approach would lead to a more optimal definition of τ, in this paper we limit our attention, mainly for simplicity, to scalar definitions.

In this paper, we will call the SUPG formulation introduced in Reference [1] for compressible flows ‘(SUPG)$_{82}$’, and the set of τs introduced in conjunction with that ‘τ$_{82}$’. The τ used in Reference [4] with (SUPG)$_{82}$ is a slightly modified version of τ$_{82}$. A shock-capturing parameter, which we will call in this paper ‘δ$_{91}$’, was embedded in the shock-capturing term used in Reference [4]. Subsequent minor modifications of τ$_{82}$ took into account the interaction between the shock-capturing and the (SUPG)$_{82}$ terms in a fashion similar to how it was done in Reference [7] for advection–diffusion–reaction equations. All these slightly modified versions of τ$_{82}$ have always been used with the same δ$_{91}$, and we will categorize them in this paper all under the label ‘τ$_{82}$-MOD’.

New ways of computing the τs based on the element-level matrices and vectors were introduced in Reference [2] in the context of the advection–diffusion equation and the Navier–Stokes equations of incompressible flows. These definitions are expressed in terms of the ratios of the norms of the relevant matrices or vectors. While they automatically take into account the local length scales, advection field and the element-level Reynolds number, they do not require explicit definitions for length or velocity scales. Our experience shows that the performance of the τ is not much influenced by the type of matrix norm selected. This is because the τ calculations involve only the ratios of the norms of the matrices. Although based on these definitions a τ can also be calculated in an element for each degree-of-freedom, in this paper, for simplicity, we limit our attention to definitions based on a single τ for each element.

It was pointed out in Reference [2] that the τs to be used in advancing the solution from time level $n$ to $n+1$ (including the τ embedded in a stabilization term that resembles a discontinuity-capturing term) should be evaluated based on the flow field at time level $n$. That way we are spared from another level of non-linearity. Although this comes at the cost of additional storage, the computational simplicity makes it an attractive alternative.

Our objective in this paper is to have, for inviscid compressible flows with shocks, alternatives to τ$_{82}$-MOD that are simple and also free from the requirement for explicit definitions for length or velocity scales. To that end, we apply the τ definitions based on the element-level matrices to the (SUPG)$_{82}$ formulation supplemented with δ$_{91}$. Our studies here include performance comparisons between these τs (with δ$_{91}$) and a τ$_{82}$-MOD (also with δ$_{91}$) and comparisons between evaluating these τs and δ$_{91}$ at time level $n$ and at (every non-linear iteration of) time level $n+1$. We also test here the influence of activating an algorithmic option introduced in Reference [8], where if a convergence stagnation is detected, δ$_{91}$ is frozen at its current value.
2. EULER EQUATIONS

The system of conservation laws governing inviscid, compressible flows are the Euler equations. In two dimensions these equations can be written in terms of the conservation variables, $\mathbf{U} = (\rho, \rho u, \rho v, \rho e)$, as

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}_x}{\partial x} + \frac{\partial \mathbf{F}_y}{\partial y} = 0 \quad \text{on } \Omega \times [0, T]$$

(1)

Here $\rho$ is the fluid density, $\mathbf{u} = (u, v)$ is the velocity vector, $e$ is the total energy per unit mass, $\mathbf{F}_x$ and $\mathbf{F}_y$ are the Euler fluxes, $\Omega$ is a domain in $\mathbb{R}^2$, and $T$ is a positive real number. We denote the spatial and temporal co-ordinates, respectively, by $\mathbf{x} = (x, y) \in \overline{\Omega}$ and $t \in [0, T]$, where the superimposed bar indicates set closure, and $\Gamma$ is the boundary of domain $\Omega$. We consider ideal gases. Alternatively, Equation (1) can be written as

$$\frac{\partial \mathbf{U}}{\partial t} + \mathbf{A}_x \frac{\partial \mathbf{U}}{\partial x} + \mathbf{A}_y \frac{\partial \mathbf{U}}{\partial y} = 0 \quad \text{on } \Omega \times [0, T]$$

(2)

where $\mathbf{A}_x = \partial \mathbf{F}_x/\partial \mathbf{U}$ and $\mathbf{A}_y = \partial \mathbf{F}_y/\partial \mathbf{U}$. We assume that we have an appropriate set of boundary and initial conditions associated with Equation (2).

3. STABILIZED FORMULATION AND STABILIZATION PARAMETERS

Considering a standard discretization of $\Omega$ into finite elements, the (SUPG)$_{82}$ formulation for the Euler equations in conservation variables introduced in Reference [1] and supplemented with a shock-capturing term Reference [4] is written as

$$\int_{\Omega} \mathbf{W}^h \cdot \left( \frac{\partial \mathbf{U}^h}{\partial t} + \mathbf{A}_x^h \frac{\partial \mathbf{U}^h}{\partial x} + \mathbf{A}_y^h \frac{\partial \mathbf{U}^h}{\partial y} \right) \, d\Omega$$

$$+ \sum_{e=1}^{n_e} \int_{\Omega_e} \tau \left( \frac{\partial \mathbf{W}^h}{\partial x} \mathbf{A}_x^h + \frac{\partial \mathbf{W}^h}{\partial y} \mathbf{A}_y^h \right) \cdot \left[ \frac{\partial \mathbf{U}^h}{\partial t} + \mathbf{A}_x^h \frac{\partial \mathbf{U}^h}{\partial x} + \mathbf{A}_y^h \frac{\partial \mathbf{U}^h}{\partial y} \right] \, d\Omega$$

$$+ \sum_{e=1}^{n_e} \int_{\Gamma_e} \delta_{\gamma_1} \left( \frac{\partial \mathbf{W}^h}{\partial x} \cdot \frac{\partial \mathbf{U}^h}{\partial x} + \frac{\partial \mathbf{W}^h}{\partial y} \cdot \frac{\partial \mathbf{U}^h}{\partial y} \right) \, d\Omega = 0$$

(3)

Here $\mathbf{W}^h$ and $\mathbf{U}^h$ are the finite-dimensional test and trial functions that are defined on standard finite element spaces. In Equation (3), the first integral corresponds to the Galerkin formulation, the first series of element-level integrals are the SUPG stabilization terms, and the second series of element-level integrals are the shock-capturing terms added to the variational formulation to prevent spurious oscillations around shocks. The SUPG term has a strong influence on the convergence and accuracy of inviscid flow solutions. It was shown in Reference [9] that solutions with shock-capturing only converge slower and are less accurate than when both stabilization terms are present. We also note from Reference [10] that convergence estimates do not depend on the particular forms of $\tau s$ or $\delta s$. The version of $\tau_{82-MOD}$ used in this paper

for comparison is calculated based on the following expressions:

\[
\tau = \max[0, \tau_t + \zeta(\tau_a - \tau_{\delta_0})]
\]

\[
\tau_t = \frac{2}{3(1 + 2xCr)} \tau_a, \quad \tau_a = \frac{h}{2(c + |\mathbf{u} \cdot \mathbf{b}|)}, \quad \tau_{\delta_0} = \frac{\delta_0}{(c + |\mathbf{u} \cdot \mathbf{b}|)^2}
\]

(4)

Here \(c\) is the acoustic speed, \(Cr = (c + |\mathbf{u} \cdot \mathbf{b}|)\Delta t/h\) is the Courant (CFL) number, \(\Delta t\) is the time step, \(x\) is a parameter controlling stability and accuracy of the time-marching scheme, and the element length \(h = \sqrt{2A}\) for triangular elements, where \(A\) is the element area. The vector \(\mathbf{b}\) and the coefficient \(\zeta\) are defined as

\[
\mathbf{b} = \frac{\nabla \|\mathbf{u}\|^2}{\|\nabla \|\mathbf{u}\|^2}, \quad \zeta = \frac{2xCr}{1 + 2xCr}
\]

(6)

The shock-capturing parameter, \(\delta_{91}\), is calculated based on the expression

\[
\delta_{91} = \left[ \frac{\left| A_x^h \frac{\partial \mathbf{u}^h}{\partial x} + A_y^h \frac{\partial \mathbf{u}^h}{\partial y} \right|^2}{\left| \frac{\partial \xi}{\partial x} \frac{\partial \mathbf{u}^h}{\partial x} + \frac{\partial \xi}{\partial y} \frac{\partial \mathbf{u}^h}{\partial y} \right|^2 + \left| \frac{\partial \eta}{\partial x} \frac{\partial \mathbf{u}^h}{\partial x} + \frac{\partial \eta}{\partial y} \frac{\partial \mathbf{u}^h}{\partial y} \right|^2} \right]^{1/2}
\]

(7)

where \(\xi\) and \(\eta\) are the element co-ordinates, and \(\hat{A}_0\) is the Jacobian of the transformation between the entropy and conservation variables. We define the following element-level matrices, and the SUPG parameters from these element-level matrices, as given in Reference [2]:

\[
m : \int_{\Omega} W^h : \frac{\partial \mathbf{u}^h}{\partial t} \, d\Omega
\]

(8)

\[
\hat{c} : \int_{\Omega} \left( \frac{\partial W^h}{\partial x} \cdot A_x^h \frac{\partial \mathbf{u}^h}{\partial t} + \frac{\partial W^h}{\partial y} \cdot A_y^h \frac{\partial \mathbf{u}^h}{\partial t} \right) \, d\Omega
\]

(9)

\[
c : \int_{\Omega} \left( W^h : A_x^h \frac{\partial \mathbf{u}^h}{\partial x} + W^h : A_y^h \frac{\partial \mathbf{u}^h}{\partial y} \right) \, d\Omega
\]

(10)

\[
\hat{k} : \int_{\Omega} \left( \frac{\partial W^h}{\partial x} \cdot A_x^h \frac{\partial \mathbf{u}^h}{\partial x} + \frac{\partial W^h}{\partial y} \cdot A_y^h \frac{\partial \mathbf{u}^h}{\partial y} \right) \, d\Omega
\]

(11)

\[
\tau_{S1} = \frac{\|c\|}{\|k\|}, \quad \tau_{S2} = \frac{\Delta t}{2} \frac{\|c\|}{\|k\|}, \quad \tau = \left( \frac{1}{\tau_{S1}} + \frac{1}{\tau_{S2}} \right)^{-1/r}
\]

(12)

Here \(r\) is an integer parameter determining the smoothness of the transition between the two limits of the SUPG parameter and \(\|\mathbf{a}\| = \max_{1 \leq i \leq n_{ee}} \{ |a_{11}| + |a_{12}| + \cdots + |a_{i, n_{ee}}| \}\), where \(n_{ee}\) is
the number of element equations, that is, the number of element nodes times the number of degrees of freedom per node.

The solution is advanced in time by the implicit predictor-multicorrector algorithm given in Reference [1], with \(
\alpha = 0.5.
\) Although local time stepping can be used as shown in Reference [8], all solutions in this work are obtained using a fixed time step size with \(CFL = 1.\) The linear equation systems involved are solved by a nodal-block-diagonal element-by-element preconditioned GMRES method. This iterative solution technique, known as linear-GMRES, usually exhibits a residual stagnation after a decrease of two or three orders of magnitude [11]. The stagnation depends on factors such as the tolerance of the GMRES algorithm, dimension of the Krylov subspace, and the preconditioning type. Similar convergence problems were observed by Venkatakrishnan [12] in implicit unstructured higher-order finite volume upwind schemes. Motivated by the solution proposed in Reference [12], Catabriga and Coutinho [8] proposed an algorithm where the shock-capturing parameter is frozen when a convergence stagnation is detected. This idea was based on the observation that the shock-capturing parameter plays a role similar to the limiter functions in approximate Riemann solvers.

4. NUMERICAL RESULTS

4.1. Oblique shock

The first problem is a Mach 2 uniform flow over a wedge, at an angle of \(-10^\circ\) with respect to the horizontal wall. The solution involves an oblique shock at an angle of \(29.3^\circ\) emanating from the leading edge of the wedge, as shown in Figure 1. The computational domain is a square with \(0 \leq x \leq 1\) and \(0 \leq y \leq 1.\) Prescribing the following inflow data on the left and top boundaries results in a solution with the following outflow data:

\[
\begin{align*}
\text{Inflow} & : & \begin{cases} 
M = 2.0 \\
\rho = 1.0 \\
u = \cos 10^\circ \\
v = - \sin 10^\circ \\
p = 0.17857
\end{cases} \\
\text{Outflow} & : & \begin{cases} 
M = 1.64052 \\
\rho = 1.45843 \\
u = 0.88731 \\
v = 0.0 \\
p = 0.30475
\end{cases}
\end{align*}
\]

Here \(M\) is the Mach number and \(p\) is the pressure. Four Dirichlet boundary conditions are imposed at the left and top boundaries, the slip condition with \(v = 0\) is set at the bottom boundary, and no boundary condition is imposed at the outflow (right) boundary. A \(20 \times 20\) mesh with 800 linear triangles and 441 nodes is employed. The tolerance of the preconditioned GMRES algorithm is 0.1, the dimension of the Krylov subspace is 5, and the number of multi-corrections is 3. All computations are initialized with the inflow values.

Figure 2(a) shows the density along line \(x = 0.9,\) computed with the \(\tau\) based on the element-level matrices (with \(r = 1, 2, 3\) and 5) and \(\tau_{82-\text{MOD}}.\) We update \(\tau\) and \(\delta_{91}\) at every non-linear iteration of a time step (i.e. iteration update). Here, and in the rest of this paper, all results indicated by \(\tau_{82-\text{MOD}}\) have been obtained with updating \(\tau_{82-\text{MOD}}\) and \(\delta_{91}\) at every non-linear iteration (except for cases where \(\delta_{91}\) is frozen upon detection of convergence stagnation). Solutions with different values of \(r\) are very similar, and therefore in the rest of this paper
Figure 1. Oblique shock—problem description.

Figure 2. Oblique shock—solutions and residuals obtained with τ based on element-level matrices (with \( r = 1, 2, 3 \) and 5) and TAU_82-MOD: (a) density profile at \( x = 0.9 \); and (b) evolution of density residual.

we will use \( r = 2 \). Figure 2(b) shows the evolution of the density residual for 300 steps. We also tested updating \( \tau \) and \( \delta_{91} \) only at every time step, at the beginning of a time step (i.e., time step update). Figure 3(a) shows the density along line \( x = 0.9 \), with iteration update and time step update. The two solutions are virtually identical. Figure 3(b) shows the evolution of the density residual for 300 steps. The evolution of the density residual for the time step update
Figure 3. Oblique shock—solutions and residuals obtained with $\tau$ based on element-level matrices, with iteration update and time step update, and $\tau_{82\text{-MOD}}$: (a) density profile at $x = 0.9$; and (b) evolution of density residual.

is faster than that for the iteration update. It was reported in Reference [9] that in computation of this problem with only shock-capturing (without SUPG stabilization), the solutions were more diffusive and the convergence was slower. After 300 time steps of computation with only shock-capturing, the density residual decreased only two orders of magnitude, while in computation with shock-capturing and SUPG stabilization the decrease was approximately five orders of magnitude. It was also reported in Reference [9] that in computations with only shock-capturing, the solutions were more prone to mesh orientation effects. Figures 4(a) and (b) show the influence of a technique [8] that detects the convergence stagnation and freezes the shock-capturing parameter. Both time step update and iteration update are used in updating $\tau$ and $\delta_{91}$. The density and the evolution of its residual during the iterations are very similar to those obtained without freezing $\delta_{91}$. However, a few more GMRES iterations are needed in time step update compared to iteration update. The exact number of iterations for both cases are 4879 and 4798, respectively, and these are lower than the number of GMRES iterations we have in computation with $\tau_{82\text{-MOD}}$, which is 5226 GMRES iterations.

4.2. Reflected shock

This problem consists of three regions (R1, R2 and R3) separated by an oblique shock and its reflection from a wall, as shown in Figure 5. Prescribing the following Mach 2.9 inflow data in the first region on the left (R1), and requiring the incident shock to be at an angle...
of 29°, leads to the following exact solution at the other two regions (R2 and R3):

\[
\begin{align*}
\text{R1:} & \quad \begin{cases} 
M = 2.9 \\
\rho = 1.0 \\
u = 2.9 \\
v = 0.0 \\
p = 0.714286
\end{cases} \\
\text{R2:} & \quad \begin{cases} 
M = 2.3781 \\
\rho = 1.7 \\
u = 2.61934 \\
v = -0.50632 \\
p = 1.52819
\end{cases} \\
\text{R3:} & \quad \begin{cases} 
M = 1.94235 \\
\rho = 2.68728 \\
u = 2.40140 \\
v = 0.0 \\
p = 2.93407
\end{cases}
\end{align*}
\]

The computational domain is a rectangle with 0 \(\leq x \leq 4.1\) and 0 \(\leq y \leq 1\). We prescribe the density, velocities and pressure at the left and top boundaries, the slip condition with \(v = 0\) is imposed at the bottom boundary, and no boundary condition is imposed at the outflow (right)....
Figure 6. Reflected shock—structured mesh—solutions and residuals obtained with \( \tau \) based on element-level matrices, with iteration update and time step update, and \( \tau_{82\text{-MOD}} \): (a) density profile at \( y = 0.25 \); and (b) evolution of density residual.

boundary. We use a structured mesh with \( 60 \times 20 \) cells, where each cell is divided into two triangles (1281 nodes and 2400 elements), and an unstructured mesh with 1837 nodes and 3429 elements. The tolerance of the preconditioned GMRES algorithm is 0.1, the dimension of the Krylov subspace is 5, and the number of multi-corrections is 3. All computations are initialized with the inflow values.

Figure 6(a) shows the density along line \( y = 0.25 \) for the structured mesh, computed with the \( \tau \) based on element-level matrices and \( \tau_{82\text{-MOD}} \), for iteration update and time step update. The two solutions are very similar. Figure 6(b) shows the evolution of the density residual for 300 steps. We observe that the density residual computed with time step update is smaller than the one computed with the other alternative strategies. Figures 7(a) and (b) show the influence of freezing \( \delta_{91} \) when convergence stagnation is detected. Similar to the way it was in the oblique shock problem, the density profiles computed with freezing \( \delta_{91} \) are in close agreement with the results obtained without freezing \( \delta_{91} \). However, here \( \tau_{82\text{-MOD}} \) leads to a faster evolution of the density residual, needing 3226 GMRES iterations. In computation with \( \tau \) based on element-level matrices, on the other hand, we have 3627 GMRES iterations with time step update and 3787 GMRES iterations with iteration update.

Figure 8 shows, in computations with the unstructured mesh, the evolution of the density residual for 300 steps for iteration update and time step update. We observe that the density residual diminishes significantly faster for the case where \( \tau \) is based on element-level matrices and \( \tau \) and \( \delta_{91} \) are updated only at every time step. The influence of freezing \( \delta_{91} \) when convergence stagnation is detected can be seen in Figures 9 and 10. We see that the steady-state density distributions are in good agreement. Similar results were obtained for \( \tau_{82\text{-MOD}} \) and iteration update solution. With \( \tau_{82\text{-MOD}} \), the freezing of \( \delta_{91} \) occurs at time step 140. With
Figure 7. Reflected shock—structured mesh—solutions and residuals obtained with $\tau$ based on element-level matrices, with iteration update and time step update, and $\tau_{82}$-MOD. The shock-capturing parameter is frozen when convergence stagnation is detected: (a) density profile at $y = 0.25$; and (b) evolution of density residual.

Figure 8. Reflected shock—evolution of the density residual in the unstructured mesh, obtained with $\tau$ based on element-level matrices, with iteration update and time step update, and $\tau_{82}$-MOD.
Figure 9. Reflected shock—density contours obtained with the unstructured mesh using $\tau$ based on element-level matrices with time step update: (a) 300 steps; and (b) convergence towards machine precision with shock-capturing parameter frozen when convergence stagnation is detected.

Figure 10. Reflected shock—evolution of the density residual in the unstructured mesh, obtained with $\tau$ based on element-level matrices, with iteration update and time step update, and $\tau_{82\text{-MOD}}$. The shock-capturing parameter is frozen when convergence stagnation is detected.

$\tau$ based on element-level matrices, the freezing occurs at step 175 for iteration update and step 216 for time step update. Also in this problem the solution obtained with $\tau_{82\text{-MOD}}$ reaches steady-state faster, as shown in Figure 10. Note that the $\tau_{82\text{-MOD}}$ solution reaches steady-state in 836 steps, needing 9573 GMRES iterations. With $\tau$ based on element-level matrices, we have 9980 GMRES iterations with iteration update and 9485 GMRES iterations with time step update.

5. CONCLUDING REMARKS

We highlighted, for the SUPG formulation of inviscid compressible flows with shocks, stabilization parameters defined based on the element-level matrices. These definitions are expressed in terms of the ratios of the norms of the matrices. They take automatically into account the flow field, the local length scales, and the time step size, but without requiring
explicit definitions for length or velocity scales. Calculations of these stabilization parameters are straightforward. By inspecting the solution quality and convergence history, we compared the performance of these stabilization parameters, accompanied by a shock-capturing parameter introduced earlier, with the performance of a stabilization parameter introduced earlier, accompanied by the same shock-capturing parameter. Also by inspecting the solution quality and convergence history, we investigated the performance difference between updating the stabilization and shock-capturing parameters at the end of every time step and at the end of every non-linear iteration within a time step. We observed that the solution qualities were quite comparable. Updating the new stabilization parameters at the end of every time step gives better convergence than updating them at the end of every non-linear iteration. We also investigated the influence of activating an algorithmic option that was introduced earlier, which is based on freezing the shock-capturing parameter at its current value when a convergence stagnation is detected. From the convergence histories in that investigation, we observe that this freezing option benefits both the older and newer definitions of the stabilization parameters, with a slightly more benefit for the older definition.

ACKNOWLEDGEMENTS
This work is partly supported by grant CNPq/PROTEM-CC 68.0066/01-2 and CNPq/MCT 522692/95-8. The third author was supported by the US Army Natick Soldier Center and NASA.

REFERENCES