European Congress on Computational Methods in Applied Sciences and Engineering
ECCOMAS Computational Fluid Dynamics Conference 2001
Swansea, Wales, UK, 4-7 September 2001

ACCELERATION STRATEGIES FOR SOLUTION OF EULER EQUATIONS USING AN EDGE-BASED SUPG FORMULATION WITH SHOCK-CAPTURING

Andréa M. P. Valli*, Lucia Catabriga†, and Alvaro L. G. A. Coutinho‡

*Department of Computer Science (DI), Universidade Federal do Esp. Santo (UFES), Av. Fernando Ferrari, s/n, Goiabeiras, 29060-900, Vitoria, ES, Brazil Email: avalli@inf.ufes.br, Web page: http://www.inf.ufes.br/ avalli

†Department of Computer Science (DI), Universidade Federal do Esp. Santo (UFES), Av. Fernando Ferrari, s/n, Goiabeiras, 29060-900, Vitoria, ES, Brazil Email: luciac@inf.ufes.br, Web page: http://www.inf.ufes.br/ lucia

‡Department of Civil Engineering - COPPE, Federal University of Rio de Janeiro (UFRJ), Caixa Postal 68506, Rio de Janeiro, RJ, Brazil Email: alvaro@nacad.ufrj.br, Web page: http://www.coppe.ufrj.br/ alvaro

Key words: Acceleration Techniques, Compressible Flows, Stabilized Finite Element, Local-Time-Stepping Strategy, Edge-Based Data Structures, Adaptive Timestepping Strategies, Shock Capturing.

Abstract. *This work investigates the use of acceleration techniques for the compressible Euler equations. We introduce a local time-stepping strategy with adaptive feedback control selection of the CFL condition. We also examine the efficiency of this local time-stepping strategy combined with a procedure to freeze the shock-capturing term when convergence stagnates. The present solution method employs an implicit, edge-based implementation of the semi-discrete SUPG formulation with shock capturing for the Euler equations in conservative variables. By disassembling the resulting finite element matrices into their edge contributions, edge coefficients, residuals and matrix-vector products needed in Krylov-update techniques are computed based on edge data structures. Numerical results for standard problems are presented, and the efficiency of the acceleration schemes are examined.*

1 INTRODUCTION

Implicit finite element strategies for compressible flow problems in science and engineering often requires the repeated solution of nonlinear systems of equations involving millions of unknowns. To develop effective algorithms capable of high resolution, we need acceleration techniques towards steady-state, such as implicit methods, local-time-stepping strategies, edge-based data structures, adaptive timestepping strategies, shock capturing and other enhancements. Our main objective is the utilization these acceleration techniques to improve efficiency of computational codes developed within the framework of the SUPG finite element formulation with shock capturing.

Real life problems often present high variations in element size, thus the flow information propagates at considerably different rates on different parts of the domain. On some parts of the domain this rate could possibly be optimal and on the rest it could be highly suboptimal. Computing one time increment for the entire spatial domain at each time step slows down convergence and increases computational cost. Alternatively, we can locally determine time increments such that the flow information would propagate at nearly optimal rate throughout the domain [12, 9]. In this work we develop a local-time-stepping strategy with adaptive feedback control selection of the CFL condition.

Most timestep schemes are based on controlling accuracy as determined by truncation error estimates (e.g. Prediction-Modification-Correction). The objective of timestep selection is to minimize the computational effort to construct an approximate solution of a given problem in accordance with a desired accuracy. This strategy is motivated by the fact that the global error can be bounded in terms of the local truncation error per unit step. It can be shown that stepsize selection can be viewed as an automatic PID control problem [17, 15, 14, 8]. Here a PID controller was developed to select the CFL condition for the local-time-stepping strategy.

The two-dimensional Euler equations are solved using a semi-discrete implicit SUPG formulation in conservation variables with shock capturing [10, 3, 7]. Spatial discretization of the Euler equations gives rise to a system of nonlinear differential equation that is solved by an implicit predictor-multicorrector scheme. After linearization, the resulting systems are solved by the iterative preconditioned GMRES. By disassembling the resulting finite element matrices into their edge contributions, edge coefficients, residuals and matrix-vector products needed in GMRES are computed based on edge data structures. We consider here unstructured meshes composed by linear triangles.

The advantages of edge-based schemes with respect to the conventional element-based schemes are a major reduction in indirect addressing (i/a) operations and memory requirements [4, 13, 6]. It also enables a straightforward implementation of upwind-based schemes in the context of finite element methods. The edge-based data structure may be also viewed as a representation of the nodal graph of a grid composed by triangles and tetrahedra. Thus, the edge representation is an alternative data structure for computing the global matrix-vector products needed in Krylov space iterative techniques.

Several numerical experiments for the Euler equations show the superior performance of the edge-based implementation over standard element-based solutions [4].

The main objective of this work is to investigate the use of local time-stepping strategies with adaptive feedback control of the CFL condition in simulations of compressible flows by edge-based SUPG finite element solutions. We also examine the efficiency of this local time-stepping strategy combined with a procedure to freeze the shock-capturing term when convergence stagnates [2, 1].

The outline of the treatment is as follows. In the next section we briefly state the governing equations, the finite element formulation and solution approach. Section 3 shows our approach to local time-stepping. Then in section 4 we describe a simple PID control approach to select the CFL condition in the local time-stepping strategy. Following this, supporting numerical studies are presented to show the improvements obtained when our approach is employed. We solve the two-dimensional steady-state oblique shock and shock-reflection problems. Finally, we present some conclusions and future works.

2 GOVERNING EQUATIONS AND THE FINITE ELEMENT FORMULATION

The system of conservation laws governing inviscid, compressible fluid flow are the Euler equations. These equations, restricted to two spatial dimensions, may be written in terms of conservation variables $\mathbf{U} = (\rho, \rho u, \rho v, \rho e)$, as

$$\mathbf{U}_{,t} + \mathbf{F}_x + \mathbf{F}_y = \mathbf{0} \quad \text{on} \quad \Omega \times [0, T] \tag{1}$$

where $\mathbf{F}_x$ and $\mathbf{F}_y$ are the Euler fluxes given elsewhere [10], Ω is a domain in $\mathbb{R}^2$ and T is a positive real number. We denote the spatial and temporal coordinates respectively by $\mathbf{x} = (x, y) \in \overline{\Omega}$ and $t \in [0, T]$, where the superimposed bar indicates set closure, and the boundary of domain Ω by Γ. Here ρ is the fluid density; $\mathbf{u} = (u_x, u_y)^T$ is the velocity vector; e is the total energy per unit mass. We add to equation (1) the ideal gases assumption, relating pressure with the total energy per unit mass and kinetic energy. Alternatively, equation (1) may be written as,

$$\mathbf{U}_{,t} + \mathbf{A}_x \mathbf{U}_{,x} + \mathbf{A}_y \mathbf{U}_{,y} = \mathbf{0} \quad \text{on} \quad \Omega \times [0, T] \tag{2}$$

where $\mathbf{A}_i = \frac{\partial \mathbf{F}_i}{\partial \mathbf{U}}$. Associated to equation (2) we have proper boundary and initial conditions. Considering a standard discretization of Ω into finite elements, the semi-discrete finite element formulation for the Euler equations in conservation variables introduced by

Le Beau and Tezduyar [3] is written as,

$$\begin{aligned} \int_{\Omega} \mathbf{W}^h . \left(\frac{\partial \mathbf{U}^h}{\partial t} + \mathbf{A}_i^h \frac{\partial \mathbf{U}^h}{\partial x_i} \right) d\Omega \quad + \\ \sum_{e=1}^{n_{el}} \int_{\Omega^e} \boldsymbol{\tau} \mathbf{A}_k^h \left(\frac{\partial \mathbf{W}^h}{\partial x_k} \right) . \left[\frac{\partial \mathbf{U}^h}{\partial t} + \mathbf{A}_i^h \frac{\partial \mathbf{U}^h}{\partial x_i} \right] d\Omega \quad + \\ \sum_{e=1}^{n_{el}} \int_{\Omega^e} \delta \frac{\partial \mathbf{W}^h}{\partial x_i} . \frac{\partial \mathbf{U}^h}{\partial x_j} d\Omega \quad = \quad 0 \end{aligned} \tag{3}$$

where $\mathbf{W}^h$ and $\mathbf{U}^h$, respectively the discrete weighting and test functions, are defined on standard finite element spaces. In (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 particular form of the SUPG stabilization matrix employed here can be found in Aliabadi and Tezduyar [1]. The shock-capturing parameter, δ, is evaluated here using the approach proposed by Aliabadi and Tezduyar in [1].

The spatial discretization of (3) leads to a set of coupled non-linear ordinary differential equations,

$$\mathbf{M}\mathbf{a} + \mathbf{C}(\mathbf{v}) = \mathbf{0} \tag{4}$$

where $\mathbf{v}$ is the vector of nodal values of $\mathbf{U}$, $\mathbf{a}$ is its time derivative, $\mathbf{M}$ is the generalized "mass" matrix and $\mathbf{C}$ is a non-linear vector function of $\mathbf{v}$. To solve the system of non-linear ordinary differential equations (4) towards steady-state we employ here the implicit predictor/multicorrector scheme described in detail in Hughes and Tezduyar [11]. In this scheme at each non-linear iteration (or multicorrection) we have to solve the following non-symmetric algebraic system of equations,

$$\mathbf{M}^* \Delta \mathbf{a} = \mathbf{R}, \tag{5}$$

where

$$\mathbf{M}^* = \mathbf{M} + \alpha \Delta t \mathbf{K} \tag{6}$$

is a non-symmetric sparse matrix,

$$\mathbf{R} = -[\mathbf{M}\mathbf{a}^* + \mathbf{C}(\mathbf{v}^*)] \tag{7}$$

is the residual vector, function of the predicted values of $\mathbf{v}$ and $\mathbf{a}$, that is, $\mathbf{v}^*$, $\mathbf{a}^*$ and $\Delta \mathbf{a}$ is the correction in the nodal values of $\mathbf{a}$ from an iteration to the next, and $\mathbf{K}$ is a first-order approximation of $\mathbf{C}$. We adopt here $\alpha = 0.5$, which is second-order accurate in time.

The non-symmetric system of equations given by equation (5) is solved by a nodal block-diagonal left preconditioned GMRES algorithm. Thus, this solution scheme is similar to the linear GMRES scheme employed by Johan et al [12]. The edge implementation is described in detail by Catabriga and Coutinho in [4].

Catabriga and Coutinho presented in [4] a procedure to detect convergence stagnation in the computation of inviscid flows and afterwards to freeze the shock-capturing term, forcing the solution to converge to steady-state. A simple heuristic is used to detect convergence stagnation and then to stop updating the shock-capturing term [5]. As a consequence, the problem converges very fast towards a steady-state solution, eventually reaching machine zero.

The main idea is to compare the average L_2 norm of density residual (res_{avg}) computed for a certain number of time steps, entitled by i_{var} (15 to 50, for instance) with the maximum and minimum values of the same norm within the same interval, that is, res_{max} and res_{min}. Then, verify whether the maximum and minimum residuals are bounded within a narrow range around the average value, that is, if $res_{min} \geq \lambda_{min} res_{avg}$ and $res_{max} \leq \lambda_{max} res_{avg}$. If these conditions are satisfied, they consider that convergence has stagnated and freeze the shock-capturing operator. We adopt here this procedure in conjunction with a local-time-stepping strategy with adaptive feedback control selection of the CFL condition.

3 LOCAL-TIME-STEPPING STRATEGY

When one time increment, Δt, is computed for the entire spatial domain at the beginning of every timestep in an implicit predictor/multicorrector scheme, the technique is called the global-time-stepping strategy. Alternatively, we can locally determine time increments such that the flow information would propagate at nearly optimal rate throughout the domain. This approach is called the local-time-stepping strategy. This strategy is particularly important when a transient algorithm is used to obtain a steady solution and the flow information propagates at considerably different rates on different parts of the domain.

In our approach to local time-stepping, a different time increment is computed for every nodal point of the domain at the beginning of each time step to accelerate convergence towards steady-state. The local time-stepping strategy is introduced to the scheme described in the previous section by simply evaluating the effective matrix $\mathbf{M}^*$ in (6) as,

$$\begin{aligned} \mathbf{M}^* &= [\mathbf{m}^*_{ij}] \\ \mathbf{m}^*_{ij} &= \mathop{\mathsf{A}}_{e=1}^{n_e} (\mathbf{m}^{n_e}_{ij} + 0.5\,\Delta t(no_i)\,\mathbf{k}^{n_e}_{ij}) \quad \text{for} \quad i,j = 1,2 \end{aligned} \tag{8}$$

where n_e represents the number of edges in the mesh and $\Delta t(no_i)$ is the local time step, computed for each edge as,

$$\Delta t(no_i) = min_{e \in \mathcal{A}}\{\Delta t_e\} \quad \text{for} \quad i = 1, 2, \ldots, nnos \tag{9}$$

where $\mathcal{A}$ is the set of all edges containing node no_i, and $nnos$ is the total number of nodes. Each Δt_e is computed imposing a CFL condition to the mesh,

$$\Delta t_e = \frac{h_e}{c_e + |\boldsymbol{u}_e \boldsymbol{\beta}_e|} CFL \tag{10}$$

where h_e is a mesh parameter, $\boldsymbol{u}_e$ is the velocity field, c_e is the acoustic speed and $\boldsymbol{\beta}_e$ is an arbitrary vector described in detail in [1]. Note that the CFL condition may increase as the solution evolves to steady-state according to some heuristic rules, see for instance, Johan et al [12] for finite elements or Hager and Lee [9] for finite volumes. In this work, we use an automatic strategy based on the feedback control theory to choose the local CFL condition. This strategy is based on the timestep selection control algorithms presented by Valli, Coutinho and Carey in [17, 16, 15, 14].

One of the most widely used algorithms for closed-loop control is the three-term control, known as the Proportional-Integral-Differential (PID) control loop. The popularity of PID controllers can be attributed to their functional simplicity and to their robust performance in a large range of operating conditions. The objective in using PID control algorithms is to control the output along a smooth curve (vs. time) towards the set-point while minimizing overshoot, that is, the amount the system output response proceeds beyond the desired response. The following expression defines the PID feedback controller for the local CFL condition,

$$CFL_{n+1} = (\frac{e_{n-1}}{e_n})^{k_P} (\frac{tol}{e_n})^{k_I} (\frac{{e_{n-1}}^2}{e_n e_{n-2}})^{k_D} \; CFL_n, \tag{11}$$

where tol is some input tolerance, e_n is the measure of the change of the quantities of interest in timestep $\triangle t_n$, and k_P, k_I and k_D are the PID parameters. Three consecutive estimates of the solution are needed to calculate the local normalized truncation errors e_{n-2}, e_{n-1} and e_n in (11). Here, the measure of the change over a time step of the quantities of interest, e_n, is evaluated by the following norm,

$$e_n = \frac{e^*_\rho}{\text{tol}} \qquad\qquad e^*_\rho = \frac{\|\boldsymbol{\rho}^n - \boldsymbol{\rho}^{n-1}\|}{\|\boldsymbol{\rho}^n\|} \tag{12}$$

where $\boldsymbol{\rho}$ is the nodal density vector. The free parameters k_P, k_I and k_D should be tuned to minimize the computational effort to solve a given class of problem. The user should supply limiters, CFL_{min} and CFL_{max} to prevent an excessive growth (or reduction) of the timestep size (or the CFL parameter). In the next section, we present two problems to demonstrated the approach and verify the efficiency of this local time-stepping strategy combined with a procedure to freeze the shock-capturing term when convergence stagnates.

4 NUMERICAL EXAMPLES

4.1 OBLIQUE SHOCK

The first problem consists of a two-dimensional steady problem of a inviscid, Mach 2, uniform flow, over a wedge at an angle of -10° with respect to a horizontal wall, resulting in the occurrence of an oblique shock with an angle of 29.3° emanating from the leading edge of the wedge, as shown in Figure 1.

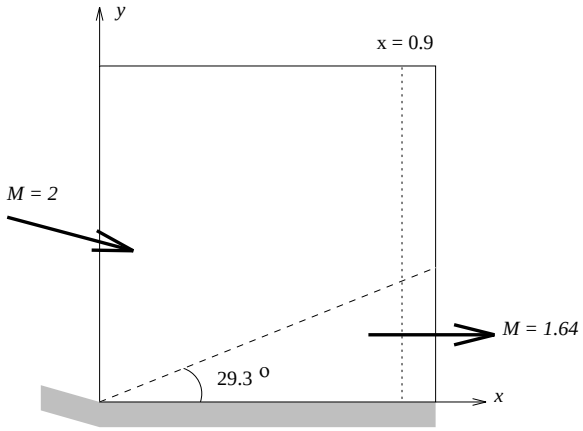


Figure 1: Oblique Shock (2D) - Problem Description.

The computational domain is the square $0 \leq x \leq 1$ and $0 \leq y \leq 1$. Prescribing the following flow data at the inflow, i.e., on the left and top sides of the shock, results in the exact solution with the flow data past the shock:

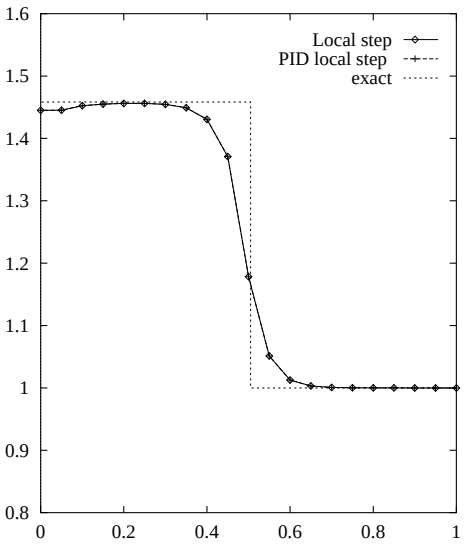
$$\text{Inflow} \begin{cases} M &= 2 \\ \rho &= 1 \\ u_1 &= cos10^0 \\ u_2 &= -sen10^0 \\ p &= 0.17857 \end{cases} \qquad \text{Outflow} \begin{cases} M &= 1.64052 \\ \rho &= 1.45843 \\ u_1 &= 0.88731 \\ u_2 &= 0 \\ p &= 0.30475 \end{cases} \tag{13}$$

where M is the Mach number, ρ is the flow density, u_1 and u_2 are the horizontal and vertical velocities respectively, and p is the pressure.

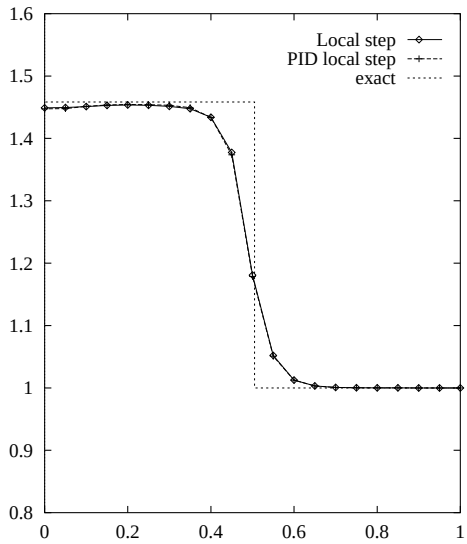
Four Dirichlet boundary conditions are imposed on the left and top boundaries; the slip condition $u_2 = 0$ is set at the bottom boundary; and no boundary conditions are imposed on the outflow (right) boundary. A 20×20 mesh with 800 linear triangles and 441 nodes and 1,240 edges is employed. Tolerance of preconditioned GMRES algorithm is set to 0.1, the dimension of the Krylov subspace to 5 and the number of multicorrections fixed to 3. All the solutions are initialized with free-stream values.

To test our PID feedback controller for the local CFL condition, we first find approximate steady-state solutions with a fixed CFL number equal to 0.5, and then we let the controller choose the local CFL number at each iteration. We solve the problem in two different ways: (a) with local-time-stepping and (b) with local-time-stepping and freezing. The objective here is to access the accuracy of the solution when the PID control strategy is applied to the problem with or without the procedure to freeze the shock-capturing term. We also want verify if the PID controller is able to produce a steady-state solution faster than the fixed CFL case, eventually reducing the residue to machine zero.

Computed densities along line $x = 0.9$ are plotted in Figure 2 for both Cases (a) and (b). After performing parametric studies for different values of PID parameters k_P, k_I, k_D, we consider $k_P = 0.18$, $k_I = 0.0$ and $k_D = 0.01$. We allowed a minimum and a maximum local CFL numbers of 0.1 and 1 respectively, and a tolerance of 0.1 for changes in nodal density. If we let the maximum value of the CFL condition grow to 1.5, we observe residual stagnation even using local-time-stepping and freezing. We can observe in both cases that the solution obtained using the PID controller is virtually identical to the solution obtained with a fixed CFL number. Note also that the procedure to freeze the shock-capturing term, Case (b), besides the fact of reducing the error in the beginning of the domain, produces better solutions than the approach used in Case (a). As a consequence, we restrict to Case (b) to demonstrate the efficiency of our PID controller in reducing computational costs.



(a) Density with local-time-stepping.

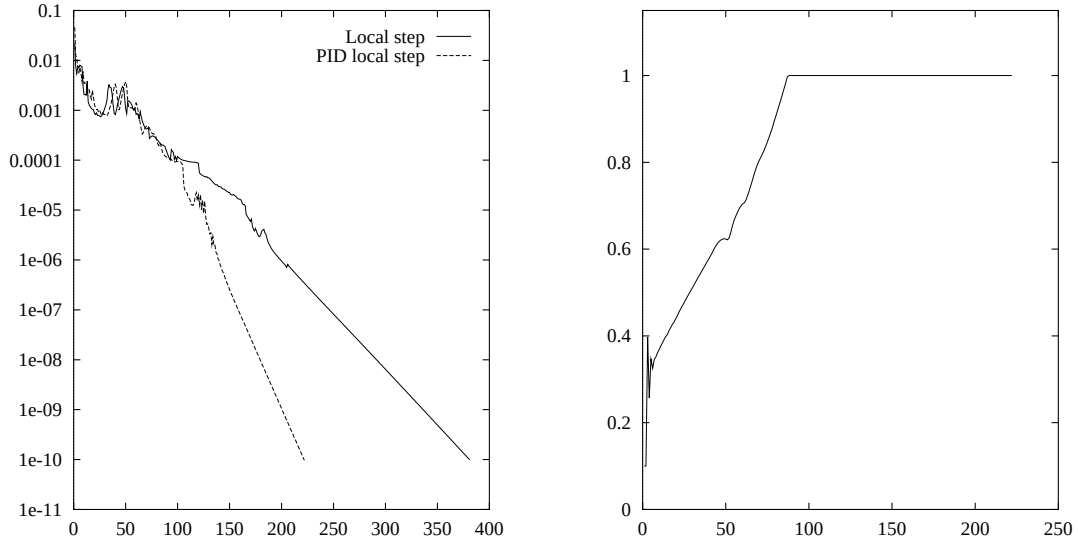


(b) Density with local-time-stepping and freezing.

Figure 2: Oblique shock problem - density profiles at $x = 0.9$.

We can measure the computational effort to calculate an approximate solutions by the total number of GMRES iterations needed to obtain the steady-state solution. We

need 2,197 GMRES iterations to obtain the steady-state solution against 1,440 iterations when we use the adaptive PID controller. Thus, we have obtained this solution 1.5 times faster using the PID control for CFL condition. With the PID control strategy we find approximate solutions with a smaller number of GMRES iterations without any significant loss of accuracy. Figure 3 shows (a) evolution of the L_2 norm of the density residual, and (b) CFL number variation. Using $\lambda_{min} = 0.8$, $\lambda_{max} = 1.2$ and $i_{var} = 20$, the freezing process starts at step 105 when we are using the PID controller and at step 120 with a fixed CFL condition. Both solutions converges to machine zero, 10^{-10}, in less than 400 steps. However, the steady-state is reached faster with the PID control process for the CFL condition. Observe also in Figure 3 that the PID control produces a smooth curve with small oscillations only in the beginning of the process.



(a) Evolution of density residual using the PID controller and fixed CFL condition.

(b) CFL number variation

Figure 3: Oblique shock problem with local-time-stepping and freezing.

4.2 REFLECTED SHOCK

This two-dimensional steady problem consists of three regions (R1, R2 and R3) separated by an oblique shock and its reflection from a wall, as shown in Figure 4. Prescribing the following Mach 2.9 flow data at the inflow, i.e., the first region on the left (R1), and requiring that the incident shock to be at an angle of 29°, leads to the exact solution (R2

and R3):

$$\text{R1}\begin{cases} M = 2.9 \\ \rho = 1.0 \\ u_1 = 2.9 \\ u_2 = 0.0 \\ p = 0.714286 \end{cases} \quad \text{R2}\begin{cases} M = 2.3781 \\ \rho = 1.7 \\ u_1 = 2.61934 \\ u_2 = -0.50632 \\ p = 1.52819 \end{cases} \quad \text{R3}\begin{cases} M = 1.94235 \\ \rho = 2.68728 \\ u_1 = 2.40140 \\ u_2 = 0.0 \\ p = 2.93407 \end{cases} \tag{14}$$

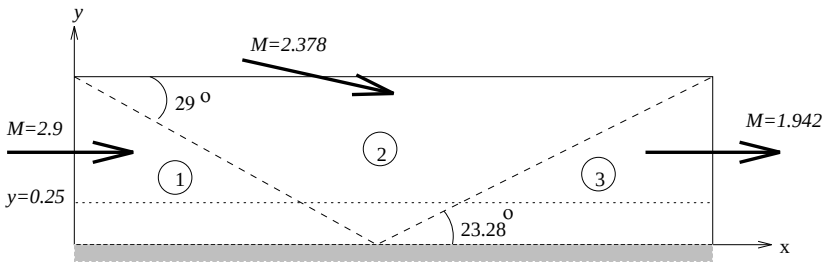
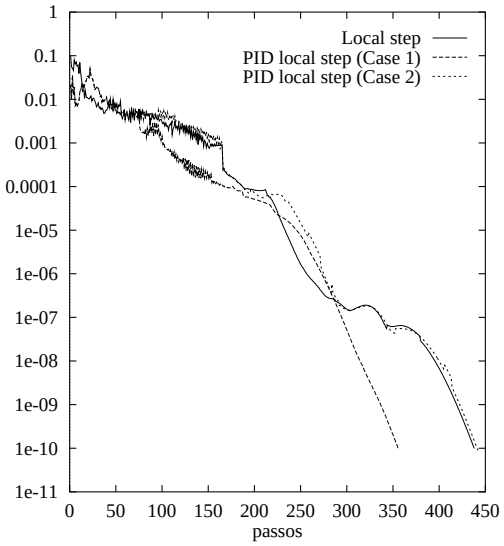


Figure 4: Reflected Shock - Problem Description.

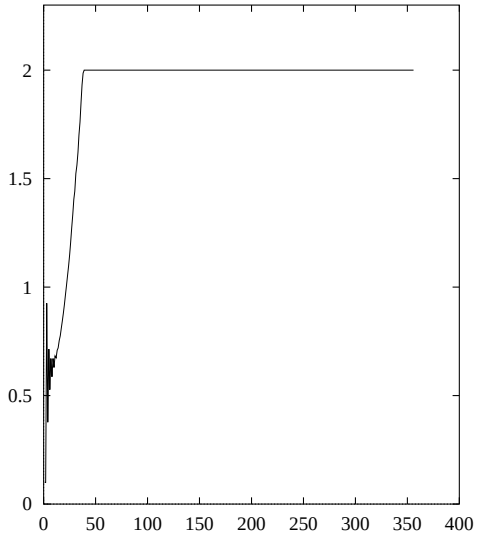
We prescribe density, velocities and pressure on the left and top boundaries; the slip condition is imposed on the wall (bottom boundary); and no boundary conditions are set on the outflow (right) boundary. The unstructured mesh consists of 1,837 nodes and 5,430 edges covering the domain $0 \leq x \leq 4.1$ and $0 \leq y \leq 1$. As before, the tolerance of the preconditioned GMRES algorithm is 0.1, the dimension of the Krylov subspace is 5, the number of multicorrections is 3, and all the solutions are initialized with free-stream values. Since the procedure to detect convergence stagnation and to freeze the shock-capturing term has been shown effective in accelerate convergence towards steady-state, we use our PID controller in conjunction with this approach to find approximate solutions of the problem. We compare the solutions using our PID controller for the CFL condition and a fixed CFL number equal to 2. We want to verify the behavior of the controller and his ability in reducing the residue to machine zero faster than the fixed CFL condition approach.

To evaluate the efficiency of our PID controller, we compare the number of GMRES iterations needed to obtain the steady-state solution with our approach and fixed CFL condition. We select the PID parameters equal to $k_P = 0.33$, $k_I = 0.003$ and $k_D = 0.01$, after a briefly parametric study. We allowed a minimum and a maximum local CFL numbers of 0.1 and 2 respectively, and a tolerance of 0.1 for changes in nodal density. We observe residual stagnation even using shock-capturing and freezing if we let the maximum value of the CFL condition be greater than 2. Figure 5 shows (a) evolution of the L_2 norm of the density residual, and (b) CFL number variation. Using $\lambda_{min} = 0.8$, $\lambda_{max} = 1.2$ and $i_{var} = 15$, the freezing process starts at step 75 when we are using the PID controller (Case 1) and at step 165 with a fixed CFL condition.

We need 4,185 GMRES iterations to obtain the steady-state solution against 3,322 iterations when we use the PID controller. Thus, we have obtained this solution 1.3 times faster using the PID control for adaptive CFL selection. We observe that if we impose that the freezing process starts at step 165 for the solution obtained using the PID controller (Case 2), we need 4,025 GMRES iterations to obtain the steady-state solution and the convergence occurs in a later step equal to 442. All the solutions converge to machine zero, 10^{-10}, in less than 500 steps. Observe in Figure 5 that the PID control produces a smooth curve with behavior similar to the one obtained in the first example.



(a) Evolution of density residual using the PID controller and fixed CFL condition. Case 1: freezing at step 75; Case 2: freezing at step 165.



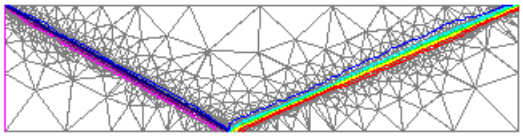
(b) CFL number variation

Figure 5: Reflected shock problem with local-time-stepping and freezing.

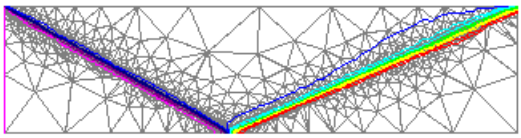
We show the density distribution obtained using a fixed CFL condition equal to 2, Figure 6(a), and the PID controller Cases 1 and 2, Figures 6(b) and 6(c) respectively. We observe in Figure 6 that the density distributions show good agreement. Consequently, in this example the PID controller also produces solutions without any significant loss of accuracy, improving convergence towards steady-state and reducing computational costs.

4.3 CONCLUSIONS

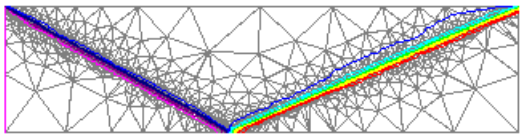
In this work we present acceleration techniques to improve efficiency of computational codes developed within the framework of the SUPG finite element formulation with shock capturing. We use edge-based data structures, shock capturing in the SUPG formulation, and a local-time-stepping strategy with an adaptive PID controller for the CFL condi-



(a) Density contours using local-time-stepping.



(b) Density contours using local-time-stepping with adaptive PID control (Case 1).



(c) Density contours using local-time-stepping with adaptive PID control (Case 2).

Figure 6: Reflected shock problem - density contours in an unstructured mesh.

tion. In particular, we verify the efficiency of a PID controller to improve convergence towards steady-state, reducing computational costs. We observe that the approach with the PID algorithm produces approximate solutions in a smaller number of steps without any significant loss of accuracy. Our results were obtained with a fixed set of GMRES parameters, a fixed number of nonlinear iterations and a given preconditioned. Although more experiments are needed to access the interplay of all these parameters, we believe that our approach may be decisive when solving complex, three-dimensional problems.

REFERENCES

[1] S.K. Aliabadi and T.E. Tezduyar. Parallel fluid dynamics computations in aerospace applications. *International Journal for Numerical Methods in Fluids*, 21:783–805, 1995.

[2] R.C. Almeida and A.C. Gale ao. An adaptative Petrov-Galerkin formulation for the compressible Euler and Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 129:157–176, 1996.

[3] G.J. Le Beau and T.E. Tezduyar. Finite element computation of compressible flows with the SUPG formulation. *ASME*, 123:21–27, 1991.

[4] L. Catabriga and A.L.G.A. Coutinho. Implicit SUPG solution of Euler equation using edge-based data structures. *submitted*, 2000.

[5] L. Catabriga and A.L.G.A. Coutinho. Improving convergence to steady-state of implicit SUPG solution of Euler equations. *submitted*, 2000.

[6] A.L.G.A. Coutinho, M.A.D. Martins, and J.L.D. Alves. Parallel iterative solution of finite element systems of equations employing edge-based data structures. In *7th SIAM Conference on Parallel Processing for Scientific Computing CD-ROM*, Philadelphia, USA, 1997.

[7] T.J.R. Hughes F. Shakib and Z. Johan. A new finite element formulation for computational fluid dynamics: X. the compressible Euler and Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 89:141–21, 1991.

[8] K. Gustafsson, M. Lundh, and G. Söderlind. A PI stepsize control for the numerical solution for ordinary differential equations. *BIT*, 28:270–287, 1988.

[9] J.O. Hager and K.D. Lee. Effects of implicit preconditioners on solution acceleration schemes in CFD. *International Journal for Numerical Methods in Fluids*, 22:1023–1035, 1996.

[10] C. Hirsh. *Numerical Computation of Internal and External Flows - Computational Methods for Inviscid and Viscous Flows*, volume 2. John Wiley and Sons Ltd, Chichester, 1992.

[11] T.J.R. Hughes and T.E. Tezduyar. Finite element methods for first-order hyperbolic systems with particular emphasis on the compressible Euler equations. *Compututer Methods in Applied Mechanics and Engineering*, 45:217–284, 1984.

[12] Z. Johan, T.J.R. Hughes, and F. Shakib. A globally convergent matrix-free algorithm for implicit time-marching schemes arising in finite element analysis in fluids. *Comput. Methods Appl. Mech. and Engrg.*, 87:281–304, 1991.

[13] L. Catabriga M.A.D. Martins, A.L.G.A. Coutinho, and J.L.D. Alves. Clustered edge-by-edge preconditioners for non-symmetric finite element equations. In *4th World Congress on Computational Mechanics CD-ROM*, Buenos Aires, Argentina, 1998.

[14] A.M.P. Valli, G.F. Carey, and A.L.G.A. Coutinho. Finite element simulation and control of nonlinear flow and reactive transport. In *Proc. 10th Int. Conf. Finite Element in Fluids*, pages 450–455, Tucson, Arizona, 1998.

[15] A.M.P. Valli, A.L.G.A. Coutinho, and G.F. Carey. Adaptive control for time step selection in finite element simulation of coupled viscous flow and heat transfer. In *European Conference on Computational Mechanics CD-ROM*, Munchen, Germany, August 1999.

[16] A.M.P. Valli, A.L.G.A. Coutinho, and G.F. Carey. Adaptive stepsize control strategies in finite element simulation of 2D Rayleigh-Benard-Marangoni flows. In *15th Brazilian Congress on Mechanical Sciences CD-ROM*, Águas de Lindóia, SP, Brazil, November 1999.

[17] A.M.P. Valli, A.L.G.A. Coutinho, and G.F. Carey. Control strategies for timestep selection and convergence rate of nonlinear iterations in simulation of Rayleigh-Benard-Marangoni flows. In *21st Iberian Latin American Congress on Computational Methods in Engineering CD-ROM*, Rio de Janeiro, RJ, Brazil, December 2000.