

SIMULTANEOUS SPACE-TIME ADAPTIVE FINITE ELEMENT SIMULATION OF 2D VISCOUS FLOWS

Andréa M. P. Valli^{1*}, José J. Camata¹, Lucia Catabriga¹, Alvaro L.G.A.
Coutinho² and Graham F. Carey³

1: Department of Computer Science
Federal University of Espírito Santo, Brazil
Av. F. Ferrari 514 - CEP 29075-910 - Vitória - ES, Brazil
e-mail: avalli,jcamata,luciac@inf.ufes.br

2: Civil Engineering Department
COPPE/Federal University of Rio de Janeiro, Brazil
P.O. Box 68506 - CEP 21945-970 - Rio de Janeiro, RJ, Brazil
e-mail: alvaro@coc.ufrj.br

3: Institute for Computational Engineering and Sciences
The University of Texas at Austin, USA
201 E. 24th St., ACES 6.430, 1 University Station C0200
78712-0227, Austin - TX - USA
e-mail: carey@cfdlab.ae.utexas.edu

Keywords: Adaptive mesh refinement, PID control for timestep selection, `libMesh` library, finite element

Abstract. *In this work we evaluate the performance of space-time adaptive finite element simulations to obtain to steady-state 2D viscous flows using the `libMesh` framework. We investigate a timestep selection technique based on feedback control theory in reducing the total computational effort when Adaptive Mesh Refinement (AMR) is utilized. First, we evaluate the efficiency of the timestep selection when fixed meshes are used. Then, we investigate the behavior of the AMR solution with the timestep selection algorithm. Numerical studies are conducted using the object oriented AMR software system `libMesh` with the `PETSc` library. Two standard test cases for transient Navier-Stokes computations are used for comparison purposes, the lid-driven cavity flow and the flow over a backward-facing step.*

1 INTRODUCTION

With the evolution of finite element methodology and its extension to more complex classes of coupled problems there has been an increasing need for improved algorithms and other enhancements such as adaptive grid refinement and coarsening (AMR) [2, 4, 11, 25, 24]. Several adaptive timestepping selection strategies have been studied as a means to provide stable accurate transient (and steady state) solutions more efficiently [19, 20, 17]. This adaptive timestepping selection process is usually approached by means of local truncation error analysis. In the same way, the adaptive grid schemes use feedback from the computed solution on a given intermediate grid to ascertain where the grid should be locally refined. This brings us to the main objective of the present work - the utilization of feedback control algorithms for timestep selection in conjunction with AMR process of finite element analysis in the simulations of steady-state 2D viscous flow.

There are many works in the literature that deal with spatial and time adaptivity [22, 18, 17]. Controlling the numerical error gives a better solution and also gives to the user the knowledge of the reliability of the results. In this work we use the same approach presented in [16, 13, 23] that splits the error in two parts: one part coming from the spatial discretization and another coming from the temporal discretization. An adaptive strategy is developed to drive the refinements, controlling the element size and distribution, and to control the timestep size. For the space discretization error we choose an *a-posteriori* error indicator similar to the classical indicator proposed by Kelly et al. [11], based on interface derivative jumps. There is an extensive literature devoted to obtaining more reliable *a-posteriori* estimates and accompanying errors indicators but, in practice, this jump indicators have proved to be broadly applicable. For the time discretization we use a proportional-integral-derivative (PID) control to select the timestep size, based on controlling normalized changes in the variables of interest [20]. Since in this work we are interested in the steady-state solution, this simple error indicator is very efficient and the computational overhead of the selection procedure is insignificant compared to solver operations [21].

Both timestep control and AMR offer means to accelerate simulation and analysis for design and rapid prototyping: timestep control reduces the CPU time to reach steady-state solution and likewise AMR permits a solution to be achieved to comparable accuracy on a coarser but better designed mesh than with standard fixed meshing. In this work, to reduce the amount of work involving in the implementation of an AMR process, we use the open-source, C++ finite element library, `libMesh` [12]. A major goal of `libMesh` is to provide a platform for parallel, adaptive, multiphysics finite element simulations in a reliable, reusable way [3]. Users can focus on the specifics of a given application without considering the additional complexities of adaptive mesh computing. In this way `libMesh` has proved a valuable testbed for a wide range of physical applications. Moreover, the available adaptive mesh refinement and coarsing scheme utilizes simple interface derivative jump (or flux jump) indicators that are essentially independent of the physics [11]. This

allows the library to be more flexibly applied in diverse applications.

In particular `libMesh` is suitable for testing our control algorithm for timestep selection [19, 20] when applied to an AMR scheme for finite element simulation. Computations of two standard test cases for transient Navier-Stokes computations are performed to compare the efficiency of the adaptive processes in reducing the total computational effort and to verify the behavior of AMR process with the timestep control. In the next section, we describe the governing equations and discrete formulation with the PID and AMR schemes and show the solution algorithm implemented. Then results are compared for fixed and variable timestep schemes, and with fixed and adaptive grid refinement and coarsening, for two benchmark problems, lid driven cavity and flow over a backward-facing step.

2 GOVERNING EQUATIONS AND DISCRETE FORMULATION

The system of equations considered is the unsteady Navier-Stokes equations for low-speed incompressible fluid flow, in the velocity-pressure formulation. The nondimensional form of the Navier-Stokes equations is

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} - \frac{1}{Re} \nabla^2 \mathbf{u} + \nabla p = \mathbf{f} \quad \text{in } \Omega \times (0, T) \quad (1)$$

$$\nabla \cdot \mathbf{u} = 0 \quad \text{in } \Omega \times (0, T) \quad (2)$$

where Ω is the flow domain, \mathbf{u} is the velocity vector, p is the pressure, Re is the Reynolds number and \mathbf{f} is an applied body force. In addition, we require Dirichlet boundary data on $\partial\Omega \times [0, T]$, $\mathbf{u} = \mathbf{u}_0$, and initial data at $t = 0$ to complete the specification of the evolution problem. Here we use a mixed finite element formulation as developed in [6].

Let us consider the spatial discretization of the viscous flow equation. Introducing a finite element discretization and basis for the velocity components and for the pressure on a discretization Ω_h , the semidiscrete projection of the variational formulation of the Navier-Stokes equations (1), (2) reduces to: find the pair (\mathbf{u}_h, p_h) with $\mathbf{u}_h \in V^h$ satisfying the initial condition with $\mathbf{u}_h = \mathbf{u}_0$ on $\partial\Omega_h$ and $p_h \in P^h$, such that

$$\int_{\Omega_h} \left(\left(\frac{\partial \mathbf{u}_h}{\partial t} + (\mathbf{u}_h \cdot \nabla) \mathbf{u}_h \right) \cdot \mathbf{v}_h + \frac{1}{Re} \nabla \mathbf{u}_h : \nabla \mathbf{v}_h - p_h \nabla \cdot \mathbf{v}_h \right) d\Omega = \int_{\Omega_h} \mathbf{f} \cdot \mathbf{v}_h d\Omega \quad (3)$$

$$\int_{\Omega_h} (\nabla \cdot \mathbf{u}_h) q_h d\Omega = 0 \quad (4)$$

for all admissible $\mathbf{v}_h \in V_h$, with $\mathbf{v}_h = 0$ on $\partial\Omega_h$, and $p_h \in P_h$. Here $\nabla \mathbf{u}_h : \nabla \mathbf{v}_h$ is the dyadic product. Introducing expansions for \mathbf{u}_h and p_h and finite element test bases for \mathbf{v}_h and q_h into the variational statements (3) and (4) and integrating, we obtain the following

nonlinear semidiscrete system of ordinary differential equations

$$\mathcal{M} \frac{d\mathbf{U}}{dt} + \mathcal{D}(\mathbf{U}) + \frac{1}{Re} \mathcal{A}\mathbf{U} - \mathcal{B}\mathbf{P} = \mathcal{F} \quad (5)$$

$$\mathcal{B}^T \mathbf{U} = \mathbf{0} \quad (6)$$

where $\mathbf{U}^T = [\mathbf{u}_1^T \mathbf{u}_2^T]$, $\mathbf{u}_i^T = [u_1^i \cdots u_N^i]$, $i = 1, 2$, for N nodal velocities and $\mathbf{P}^T = [p_1 \cdots p_M]$ for M nodal pressures. The matrices \mathcal{M} , \mathcal{A} and \mathcal{B} correspond to the respective mass, viscous and pressure terms on the left in (5), \mathcal{F} corresponds to the source term on the right and $\mathcal{D}(\mathbf{U})$ is a nonlinear function of the nodal velocities corresponding to the advective term. The resulting semidiscrete system (5) is integrated implicitly using a standard θ -method, $0 \leq \theta \leq 1$. At each timestep, we have a nonlinear system of the form

$$\begin{aligned} \mathcal{M} \frac{\mathbf{U}^n - \mathbf{U}^{n-1}}{\Delta t} + \theta \left[\mathcal{D}(\mathbf{U}^n) + \frac{1}{Re} \mathcal{A}\mathbf{U}^n - \mathcal{B}\mathbf{P}^n \right] \\ + (1 - \theta) \left[\mathcal{D}(\mathbf{U}^{n-1}) + \frac{1}{Re} \mathcal{A}\mathbf{U}^{n-1} - \mathcal{B}\mathbf{P}^{n-1} \right] \\ = \theta \mathcal{F}^n + (1 - \theta) \mathcal{F}^{n-1} \end{aligned} \quad (7)$$

$$\mathcal{B}^T \mathbf{U}^n = \mathbf{0} \quad (8)$$

where n denotes the timestep index. In the numerical studies, we are using the implicit Euler method ($\theta = 1$) even though it is only first-order accurate in time. The reason for this decision is that the second-order Crank-Nicolson method is notoriously oscillatory for problems with discontinuous initial data such as the lid-driven cavity problem shown later. We therefore, sacrifice accuracy in time for stability. Later we compare the steady-state solution with fixed and adaptive meshes for the Navier-Stokes equations (1)-(2). We assume that the steady-state occurs when the kinetic energy at two consecutive timesteps reaches a relative difference less than a specified tolerance, tol_{st} .

The nonlinear system (7),(8) is solved by Newton's method in the present study. Writing the nonlinear system formally as

$$\mathbf{g}(\mathbf{r}^n) = \mathbf{0}, \quad \text{with} \quad (\mathbf{r}^n)^T = [(\mathbf{U}^n)^T, (\mathbf{P}^n)^T] \quad (9)$$

and given \mathbf{r}_0^n , we solve the linear Jacobian system

$$\mathbf{J}(\mathbf{r}_{k+1}^n - \mathbf{r}_k^n) = -\mathbf{g}(\mathbf{r}_k^n), \quad \text{where} \quad \mathbf{J} = (\mathbf{J}_{ij}) = \left(\frac{\partial g_i}{\partial r_j} \right), \quad (10)$$

for $i, j = 1, 2, \dots, (2N + M)$ and iterate $k = 0, 1, 2, \dots$, at each timestep. The resulting linear system of equations is solved using GMRES method with the Incomplete LU-decomposition preconditioner ILU(1). In the numerical tests described later we consider 30 basis vectors for the GMRES method and a linear tolerance of 10^{-6} . For the Newton method, the non-linear tolerance is 10^{-4} .

2.1 Adaptive mesh refinement and coarsing (AMR/C)

Adaptive mesh refinement (h refinement) has been used to generate optimal grids and there is a extensive literature devoted to obtain a *posteriori* estimates and accompanying error indicators [11, 25, 2, 4]. In this work, we use the error indicator as implemented in the `libMesh` library [12], which employs a simple interface derivative jump (or flux jump) error indicator similar to the classical indicator proposed by Kelly et al. [11] to drive the h -refinement process. The adaptive technology utilizes element subdivision to locally refine the mesh and thereby resolve different scales such as boundary layers and interior shock layers. The focus in `libMesh` is on local subdivision (h refinement) with local coarsening by h restitution of subelements and the local indicators are essentially independent of the physics. The interface jump error indicator for a given element a is defined by

$$\eta_a^{FLUX} = \left(h \int_{\Omega_a} |(\nabla \mathbf{u}_{h_b} - \nabla \mathbf{u}_{h_a}) \cdot \eta|^2 ds \right)^{\frac{1}{2}}, \quad (11)$$

where element b shares an edge (face) with cell a in the finite element mesh, h is the length of this face and η is the outward unit normal for element a . In regions of rapidly changing solution gradients, the jump error (11) will be large and hence refinement will be triggered in such zones.

In this work, we employ a statistical approach to flag individual elements for refinement and coarsening, and the element error is treated as an approximately log-normal distribution. This kind of flagging scheme have been used with good success [1] and is an extension of one technique described in [14]. As shown in Figure 1, the solution error has an approximately normal distribution about its mean value μ with standard deviation σ . The user has to supply the refinement and coarsening fractions r_f and c_f . Observe that elements with errors greater than $\mu + \sigma r_f$ are flagged for refinement while elements with errors less than $\mu - \sigma c_f$ are flagged for coarsening. The final decision for refinement/coarsening can still be constrained by specifying a maximum-allowable refinement level and by compatibility conditions such as a gradual transition in cell size. The number of refinement levels per time step and the maximum-allowable refinement level are important AMR control parameters which influence the accuracy and efficiency of the method, and the choice of c_f and r_f is also problem-dependent and affects the overall outcome [1].

This scheme is beneficial in evolution problems where, at early times, the error is small and equidistributed and no elements are flagged for refinement. Later, as interesting features develop, the statistical distribution spreads and refinement and coarsing begins. As the steady-state solution is approached, the distribution of the error reaches a steady-state as well, effectively stopping the AMR/coarsening process. Here, for the statistical scheme in the AMR process, we consider the following parameters: the initial mesh (*initial-mesh*), the number of uniform refinement steps (*uniform-refinement*), the error percentage to refine (*refine-percentage*), the error percentage to coarsen (*coarsen-percentage*) and the number of maximum refinements levels (*refinement-levels*). In the numerical experi-

ments presented later, an AMR procedure is identified by the 5-tuple AMR (*initial-mesh, uniform-refinement, refine-percentage, coarsen-percentage, refinement-levels*).

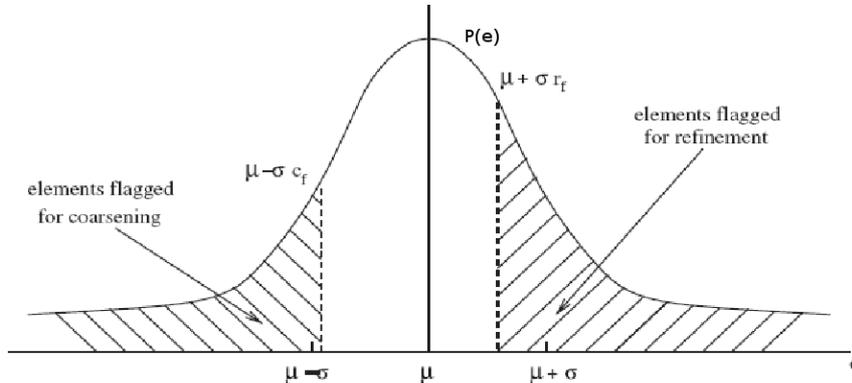


Figure 1: Graphical description of the refinement/coarsening scheme.

2.2 PID timestep selection

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. Gustafsson *at. al.* [9] and Hairer and Wanner [10] viewed the problem of automatic timestep selection as examples of a proportional-integral-derivative (PID) controller defined as

$$\Delta t_{n+1} = \left(\frac{e_{n-1}}{e_n}\right)^{k_P} \left(\frac{tol}{e_n}\right)^{k_I} \left(\frac{e_{n-1}^2}{e_n e_{n-2}}\right)^{k_D} \Delta t_n, \quad (12)$$

where tol is some input tolerance, e_n is the measure of the change of the quantities of interest in time t_n , and k_P , k_I and k_D are the PID parameters. In the present work, we use the changes in nodal velocities and pressure to compute e_n taking,

$$e_n = \frac{\|\mathbf{r}^n - \mathbf{r}^{n-1}\|}{\|\mathbf{r}^n\|}, \quad (\mathbf{r}^n)^T = [(\mathbf{U}^n)^T, (\mathbf{P}^n)^T] \quad (13)$$

The algorithm for controlling the timestep has two main parts. First, a step size is assumed, and using the newly computed solution, an *a posteriori* estimate is made of the error in the step. Second, this error measure is used to accept or reject the solution and modify the timestep accordingly. If the error is unacceptable, the new solution is discarded and we restart the time integration in the previous step with a reduced step size. If the error is acceptable, a new timestep is calculated using equation (12) and we proceed with the time integration. In the algorithm, we have to define the control data: the minimum

timestep size Δt_{min} , the maximum timestep size Δt_{max} , the PID parameters k_P , k_I , k_D , and the tolerance tol for changes in nodal velocities and pressure. The efficiency of the control was demonstrated by Valli et al. [19, 20] in numerical simulations of Rayleigh-Benard-Marangoni problems, flow over a backward-facing step and unsteady flow past a cylinder. Further, the computational overhead of the selection procedure is insignificant compared to solver operations, since timestep selection involves only storing a few extra vectors and computation of associated norms. As in [19, 20], we fix the values of the PID parameters equal to $k_P = 0.075$, $k_I = 0.175$ and $k_D = 0.01$ in all the numerical experiments performed subsequently.

2.3 Solution scheme

The solution scheme used is designed to arrive at faster steady-state solutions using the PID timestep control algorithm combined with adaptive mesh refinement (AMR). Figure 2 provides a schematic description of the main calculations to obtain the steady-state solution using PID and AMR schemes. As shown in the flowchart, after selecting initial values and generic parameters (Re , Δt , initial mesh, initial velocities, etc.), apply a fixed number of uniform refinement steps in the initial mesh. At each timestep (*Time Loop*), perform one (or more) h -adaptive refinement/coarsening steps (*AMR Loop*) using the flux-based indicator of (11) and the statistical flagging scheme mentioned in Section 2.1, calculate the new timestep using (12), test for the steady-state and exit or repeat for the next time step. Inside the *AMR Loop*, the PID error estimate (13) is calculated using two consecutive solutions at the same mesh to avoid further level of complexity. Note also that the mesh is refined inside the AMR loop only if the timestep is not rejected. In the numerical experiments presented later we consider four algorithm combinations for comparison: fixed mesh and constant timestep (case 1.1), fixed mesh and PID (case 1.2), AMR and constant timestep (case 2.1) and AMR and PID (case 2.2). In other words, we consider the cases of no adaptation, space adaptation only, time adaptation only and simultaneous space-time adaptation.

3 NUMERICAL RESULTS

3.1 Lid driven cavity problem

The lid-driven cavity flow is a standard test case for steady Navier-Stokes computations and there are numerous published results that can be used for comparison purposes [5, 7]. The domain of analysis is a unit square and both velocity components are prescribed to be zero, except at the top boundary (the lid) where the horizontal velocity component is prescribed as in [15]

$$u(x) = \begin{cases} \tanh(\beta x) & \text{for } 0 \leq x \leq 0.5, \\ -\tanh(\beta(x - 1)) & \text{for } 0.5 < x \leq 1.0. \end{cases} \quad (14)$$

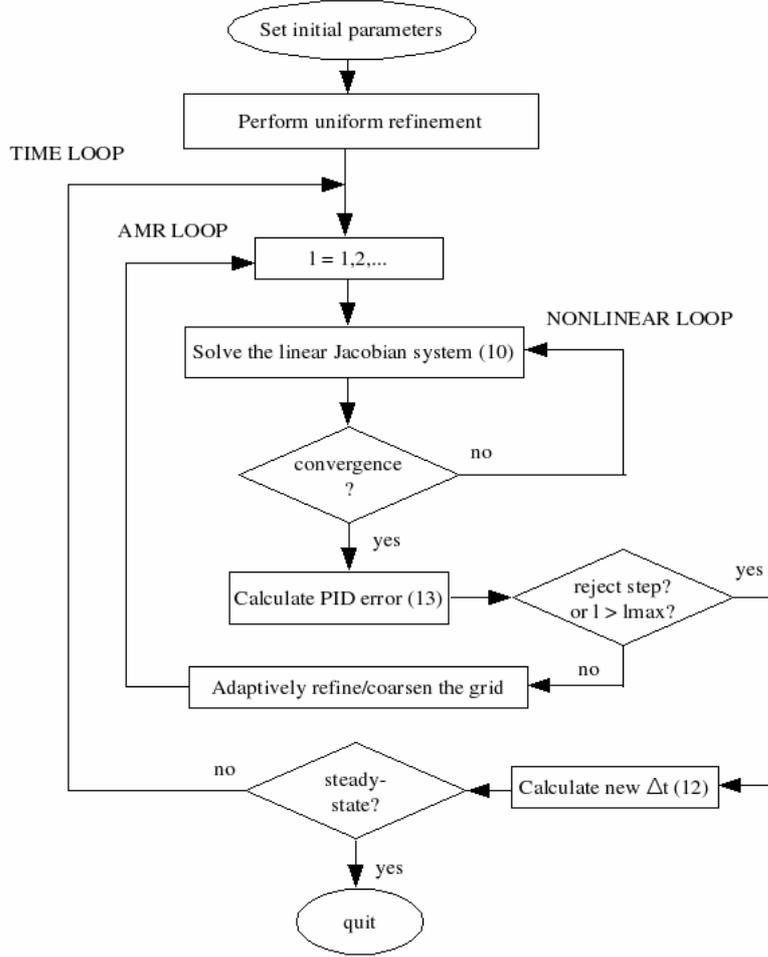


Figure 2: Flowchart for the adaptive solution scheme.

with $\beta > 0$. In this work we assume $\beta = 100$. The domain is discretized by quadratic triangular elements (TRI6) for velocity and by linear triangular elements (TRI3) for pressure. We infer experimentally that the steady-state is reached when the difference between the kinetic energy is less than 10^{-4} . We use a 80×80 fixed mesh and an adaptive mesh refinement defined by the 5-tuple AMR $(20 \times 20, 2, 0.3, 0.01, 2)$ with constant and adaptive timesteps. The numerical results will be compared with the result presented by Erturk et al. [7] for $Re = 1000$. Figure 3 shows the horizontal velocity u along the vertical centerline and the vertical velocity v along the horizontal centerline for the four algorithm combinations. The agreement for all cases is favorable when compared with the results in [7].

Next, we compare the computational effort to calculate the solution using the four algorithm combinations considered here. For the PID experiments, we define $\Delta t_{min} = 1.0$,

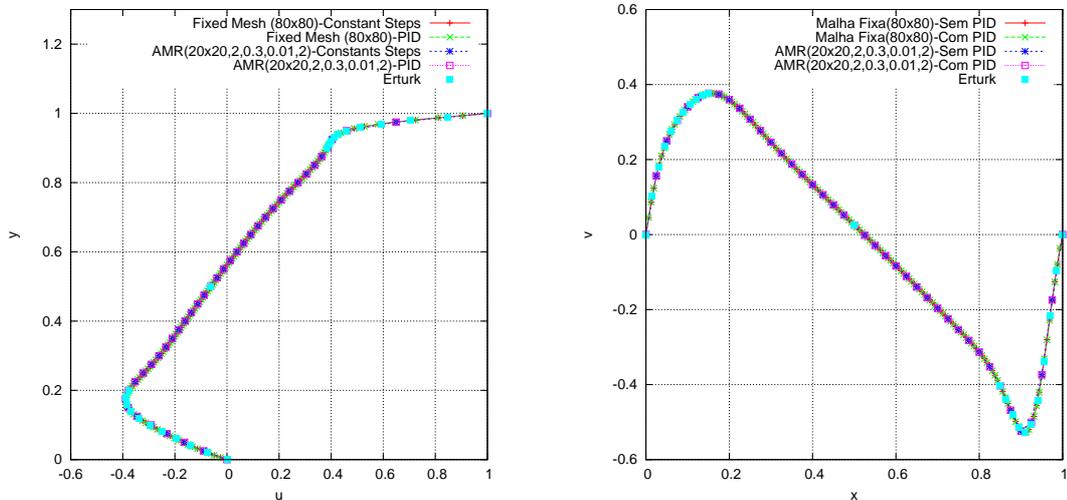


Figure 3: Horizontal velocity u along the vertical centerline (left) and the vertical velocity v along the horizontal centerline (right) for the lid driven cavity problem with $Re = 1000$.

$\Delta t_{max} = 5.0$ and $tol = 0.1$. Table 1 shows the number of nonlinear iterations (NLI), the number of linear iterations (LI), the CPU time in seconds to calculate the linear iterations (CPU_{gmres}), the CPU time in seconds to perform the refinement (CPU_{ref}), the computational effort (CPU_{effort}) and the maximum value of the streamlines (ψ_{max}). The computational effort is measured by the (CPU_{gmres}) using constant steps divided by the (CPU_{gmres}) using the PID controller for timestep selection. Observe that the computational effort to calculate the solutions in all cases is reduced using the PID controller. For both the fixed and adapted meshes, the solutions are obtained approximately 1.5 times faster using the PID controller and the maximum value of the streamlines are closer to the value calculated in [7], which corresponds to 0.118585 for a mesh with 401×401 cells using finite differences second order accuracy. The refinement CPU time (CPU_{ref}) corresponds to 0.34 % of the GMRES CPU time (CPU_{gmres}) using constant steps. However, using the PID controller this effort corresponds to only 0.25% of the GMRES CPU time. The evolution of timesteps is shown in Figure 4 for both fixed and AMR meshes. The behavior of the PID for both spatial meshes are similar for this example, stepping from the initial timestep size to the maximum timestep size of 5.0 after 40 time units.

The final AMR meshes using constant and adaptive steps are the same, as shown in Figure 5 together with the streamlines. The final number of nodes in both AMR meshes, cases 2.1 and 2.2, is equal to 6710. The steady-state using constant steps and the PID controller are achieved at approximately 80 and 120 time units respectively, as shown by the nondimensional kinetic energy versus time in Figure 6.

Table 1: Lid driven problem with $Re = 1000$.

Fixed mesh 80×80						
	NLI	LI	CPU_{gmres}	CPU_{ref}	CPU_{effort}	ψ_{max}
Constant Step (case 1.1)	218	6627	964.12	-	1.0	0.11877
PID (case 1.2)	102	6280	664.09	-	0.69	0.11869
AMR ($20 \times 20, 2, 0.3, 0.01, 2$)						
	NLI	LI	CPU_{gmres}	CPU_{ref}	CPU_{effort}	ψ_{max}
Constant Step (case 2.1)	292	2531	110.28	37.92	1.0	0.11877
PID (case 2.2)	152	2408	80.79	20.05	0.68	0.11879

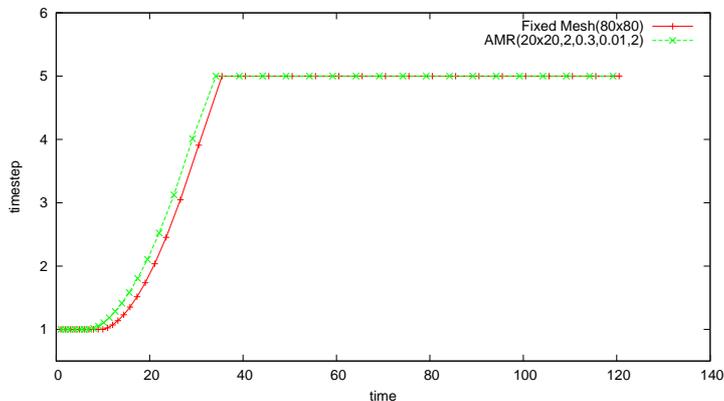


Figure 4: The lid driven cavity: timestep variation for fixed mesh and AMR.

3.2 Flow over a backward-facing step problem

The second numerical experiment is the two-dimensional backward-facing step problem. Here we compare our results with those presented by Gartling in [8]. The standard step geometry was simplified by excluding the channel upstream of the step (see Figure 7). The boundary conditions for the step geometry include the usual no-slip velocity specification for all solid surface walls as shown in Figure 7. The inlet velocity field is specified as a parallel flow given by $u(y) = 24y(0.5 - y)$ and $v(y) = 0$ for $0 \leq y \leq 0.5$. This produces a maximum inflow velocity of $u_{max} = 1.5$ and an average inflow velocity of $u_{avg} = 1.0$. We consider homogeneous natural outflow boundary conditions as shown in Figure 7. The problem is solved for a Reynolds number of $Re = 800$.

In this example the domain is also discretized by quadratic triangular elements (TRI6) for velocity and by linear triangular elements (TRI3) for pressure. The fixed mesh used comprises 128×16 cells with two elements per cell resulting in 8481 nodes and 2048 elements. For the adaptive mesh we consider the parameter tuple AMR ($64 \times 8, 1, 0.3, 0.01, 1$) and the PID parameters are: $\Delta t_{min} = 0.025$, $\Delta t_{max} = 0.5$ and $tol = 0.25 \times 10^{-3}$

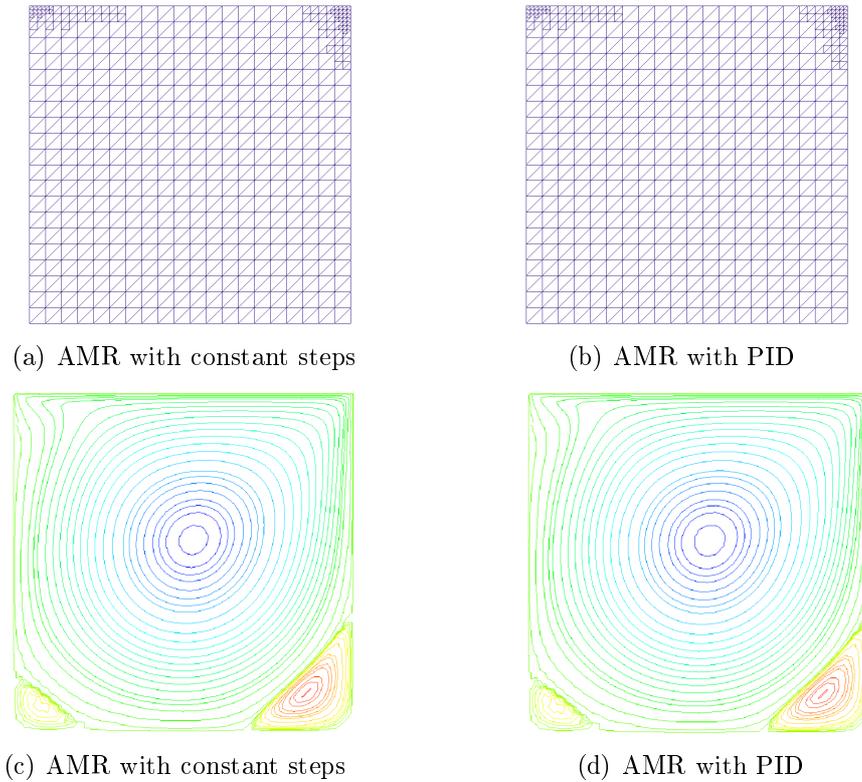


Figure 5: The lid driven cavity: final meshes (top) and streamlines (bottom).

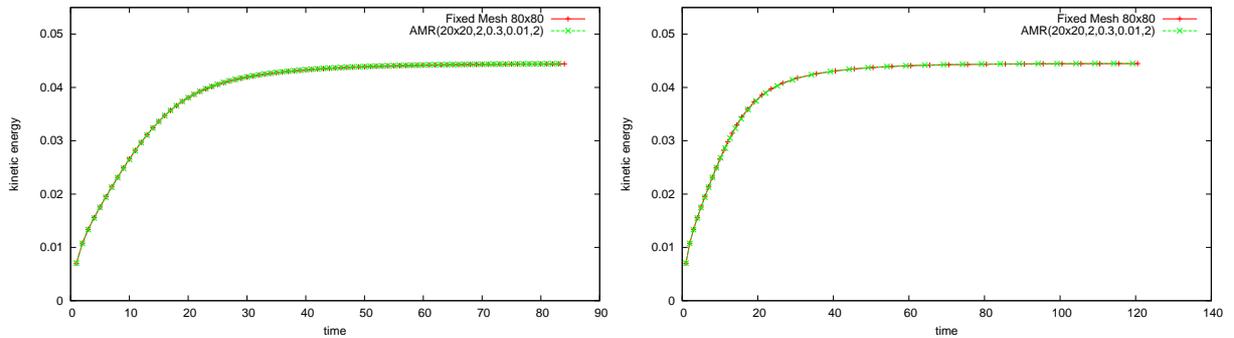


Figure 6: The lid driven cavity: kinetic energy using constant steps (left) and PID (right).

for changes in nodal velocities. Table 2 compares the PID performance for fixed and AMR meshes, using the same notation of the quantities defined in the last example. For the fixed mesh, the CPU time was reduced about 50% using the PID control even with a increased number of linear iterations. For the AMR mesh, with reference to CPU time we are able to calculate the solution 3.6 times faster using the PID controller and the total number of linear and non-linear iterations was reduced.

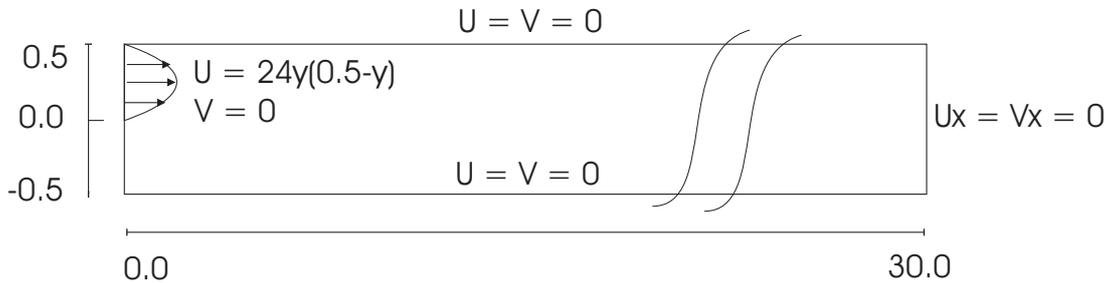


Figure 7: Backward-facing step geometry with channel dimensions and boundary conditions.

Table 2: Backward facing-step problem with $Re = 800$.

Fixed mesh 128×16					
	NLI	LI	CPU_{gmres}	CPU_{ref}	CPU_{effort}
Constant Step (case 1.1)	9474	34106	6432.78	-	1.0
PID (case 1.2)	1936	30650	4004.21	-	0.62
AMR($64 \times 8, 1, 0.3, 0.01, 1$)					
	NLI	LI	CPU_{gmres}	CPU_{ref}	CPU_{effort}
Constant Step (case 2.1)	13208	53608	7576.38	2803.11	1.0
PID (case 2.2)	3390	34649	2097.15	593.48	0.26

The timestep size variation given by the PID controller for cases 1.2 and 2.2 are presented in Figure 8. For the fixed mesh algorithm (case 1.2), the PID solution presents a smooth variation of the timesteps when confronted with the AMR algorithm (case 2.2). At the beginning of the process, the PID controller keeps the timestep value at minimum, but as soon as the solution evolves, the PID controller increases the timestep size towards its maximum value. For case 2.2, the stepsize size oscillates because of the changes in the mesh size and the limitation of the maximum timestep size. However, there was no rejected steps during the process.

Now we compare the final AMR meshes using constant timesteps and PID controller, Figures 9(a) and 9(b). Observe that the final meshes are different but they have similar features. The mesh is refined only on the upstream region of the channel and coarsened in the streamwise direction. The AMR mesh with constant steps begins with 8481 nodes and at the end of the adaptive process has 3745 nodes. In Figure 10 we show the evolution of the final number of nodes for both solutions. The final AMR mesh with PID has 3639 nodes, i.e., 106 nodes less than the final AMR mesh with constant steps. Note also that in the beginning of the solution process the number of nodes in the mesh oscillates but as the solution approaches steady state this number decreases. This is due to the coarsening process in the regions of the domain where the interface derivative jumps are small.

Although there is a difference between the two final AMR meshes, the final solution

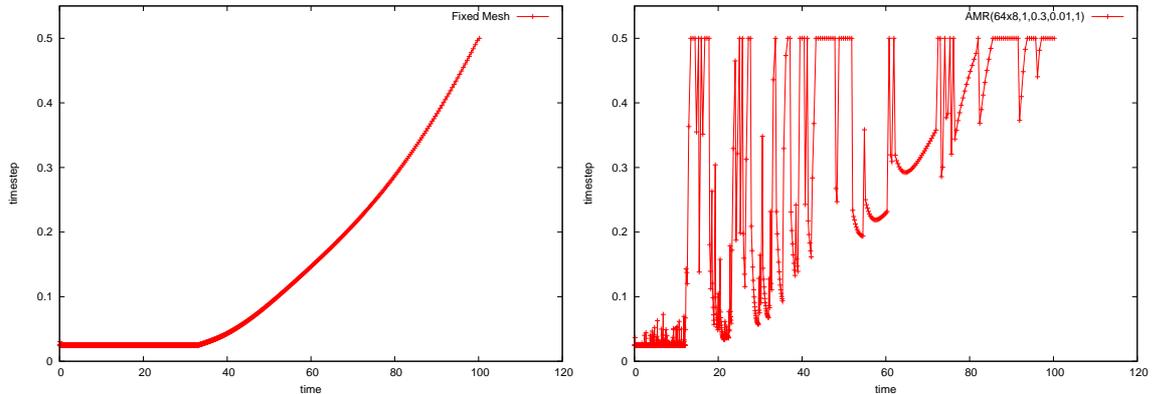
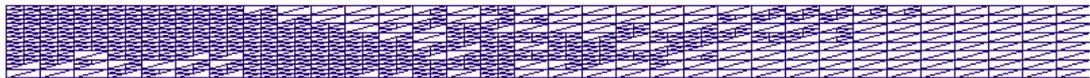
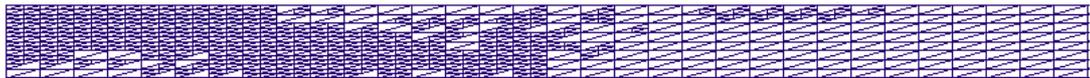


Figure 8: Backward facing-step problem - timestep variation for fixed mesh (left) and AMR (right).



(a) Constant step



(b) PID

Figure 9: Backward-facing problem for $Re = 800$ - mesh for $AMR(64 \times 8, 1, 0.3, 0.01, 1)$

obtained using the PID controller presents good accuracy as compared with the solution with fixed steps. This can be verified by the horizontal velocity components calculated at $x = 7.5$ and $x = 15.0$ using both constant and adaptive timesteps, shown in Figure 11. Note that the steady-state horizontal velocities are practically coincident. The basic features of the backward-facing step flow at $Re = 800$ is illustrated in the stream function contour plots of Figure 12 for cases 2.1 and 2.2. The plot shows only part of the channel since few phenomena of interest occur downstream of this point. The maximum streamline values obtained in the numerical experiments for cases 2.1 and 2.2 are 0.24668 and 0.24667 respectively. Our results are in good agreement with the results obtained by Gartling in [8]. In this example the steady-state was reached for all cases around time 100 time units.

4 CONCLUSIONS

We have investigated the use of a PID timestep control algorithm in conjunction with an AMR process for simulation of 2D viscous flow problems using the `libMesh` library. We have suggested an algorithm for the time-space adaptive process to obtain steady-state solutions of the problems. Uniform and adaptive finite element solutions have been computed for lid driven cavity and flow over a backward-facing step problems using fixed

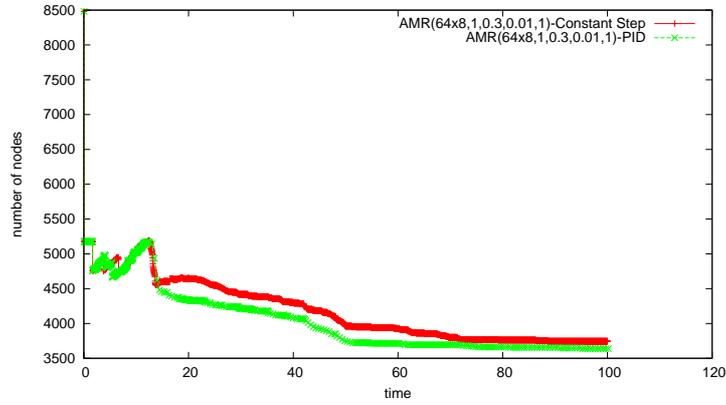


Figure 10: Backward-facing problem for $Re = 800$ - number of nodes.

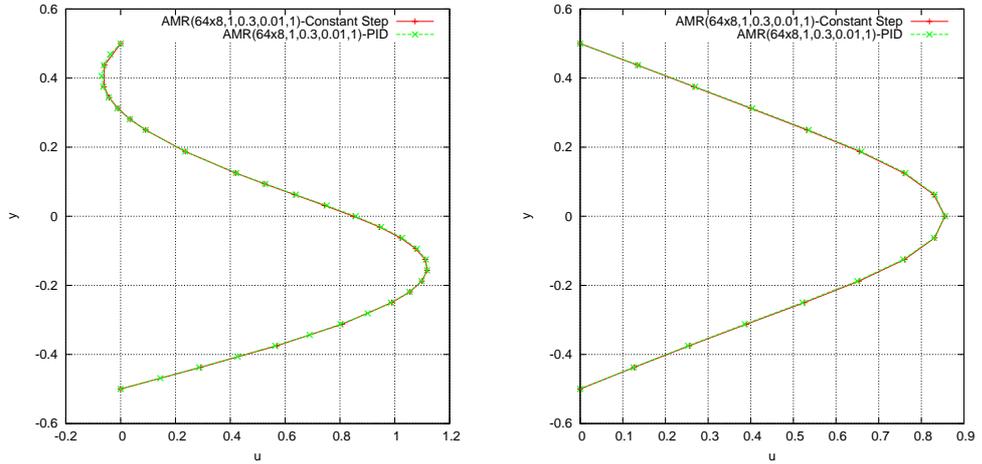


Figure 11: Horizontal velocity component at $x = 7.5$ (left) $x = 15$ (right) using AMR.

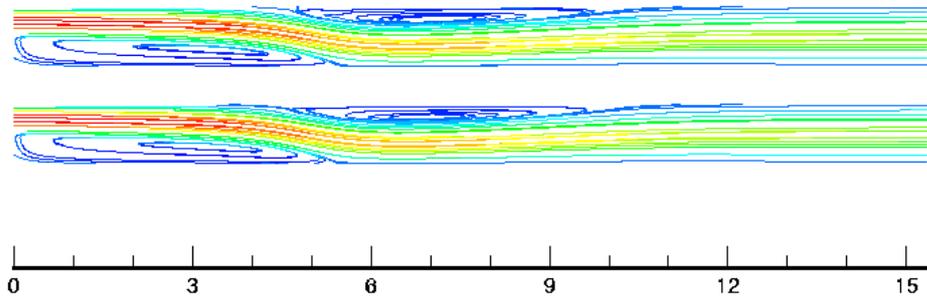


Figure 12: Streamlines using constant steps (top) and PID (bottom) using AMR.

and adaptive timesteps to test the approach. The computational effort was measured considering the cases of no adaptation, space adaptation only, time adaptation only and simultaneous space-time adaptation. Based on the numerical studies, it was concluded that the computational effort to calculate the solutions is reduced using the PID controller. For example, the solutions in the lid driven cavity problem are obtained approximately 1.5 times faster using the PID controller for both uniform and adaptive grids without any significant loss of accuracy. Moreover, the final adaptive meshes using fixed and adaptive timesteps are the same. For the second example, with reference to CPU time we are able to calculate the solution 3.6 times faster using the simultaneous adaptive of mesh and time steps (PID controller and AMR). We also observe that the final AMR mesh obtained with the PID control had less nodes than the the uniform mesh. However, the basic features of the mesh was not lost and the final solutions agree very well.

Acknowledgments

Dr. Valli was supported by a CAPES grant from the Ministry of Education, Brazil, as a visiting scholar at the University of Texas at Austin (CAPES, BEX0509/06-0). This work was also supported by CNPq (Proc. 620165/2006-5) under the Joint Cooperation between the Federal University of Esp rito Santo and the Federal University of Rio de Janeiro, Brazil.

REFERENCES

- [1] M.J. Aftosmis and M.J. Berger. Multilevel error estimation and adaptive h -refinement for cartesian meshes with embedded boundaries. *AIAA Paper 2002-0863*, 2002.
- [2] M. Ainsworth and J. T. Oden. *A Posteriori Error Estimation in Finite Element Analysis*. John Wiley & Sons, Inc., 2000.
- [3] R. H. Stogner B. S. Kirk, J. W. Peterson and G. F. Carey. libmesh: a c++ library for parallel adaptive mesh refinement/coarsening simulations. *Journal Engineering with Computers*, 22:237–254, 2006.
- [4] W. Bangerth and R. Rannacher. *Adaptive Finite Element Methods for Differential Equations*. Birkh user Verlag, Switzerland, 2003.
- [5] E. Barragy and G. F. Carey. Stream function-vorticity driven cavity flow using p finite elements. *Computers and Fluids*, 26:453–468, 1997.
- [6] G.F. Carey and J.T. Oden. *Finite Elements: Fluid Mechanics*, volume 6. Prentice–Hall, Inc., Englewood Cliffs, NJ, 1986.
- [7] E. Erturk, T.C. Corke, and C. Cokcol. Numerical solutions 2-d steady incompressible driven cavity flow at high reynolds numbers. *International Journal for Numerical Methods in Fluids*, 48:747–774, 2005.

- [8] D.K. Gartling. A test problem for outflow boundary conditions - flow over a backward-facing step. *Internat. J. Numer. Methods Fluids*, 11:953–967, 1990.
- [9] K. Gustafsson, M. Lundh, and G. Soderlind. A PI stepsize control for the numerical solution for ordinary differential equations. *BIT*, 28:270–287, 1988.
- [10] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*. Springer-Verlag, New York, NY, 1993.
- [11] D.W. Kelly, J. P. Gago, O. C. Zienkiewicz, and I. Babuska. A posteriori error analysis and adaptive processes in the finite element method: Part i error analysis. *Int. J. Numer. Meth. Engrg.*, 19:1593–1619, 1983.
- [12] LibMesh. <http://libmesh.sourceforge.net/>. visited 04-02-2007.
- [13] A. L. G. A. Coutinho P. R. M. Lyra, J. L. D. Alves and et al. Comparison of mesh refinement strategies for the h-version of the finite element method. In *X Congresso Ibero-Latino-Americano sobre Métodos Computacionais em Engenharia - MECOM*, volume 2, pages A595–A610, 1989.
- [14] J.W. Peterson, G.F. Carey, D.J. Knezevic, and B.T. Murray. Adaptive finite element methodology for tumor angiogenesis modeling. *Int. J. Numer. Meth. Eng.*, 69(6):1212–1238, 2007.
- [15] V. Prabhakar and J. N. Reddy. Spectral/hp penalty least-square finite element formulation for the steady incompressible navier-stokes equations. *Journal of Computational Physics*, 215:274–297, 2006.
- [16] R. S. Silva and R. C. Almeida. Space and time adaptivity for reaction-diffusion-convection problems. *European Congress on Computational methods in Applied Sciences and Engineering*, 2004.
- [17] R. S. Silva and R. C. Almeida. Using fuzzy controllers as adaptive procedures in the finite element method. *Commun. Numer. Meth. Engng (in press)*, 2007.
- [18] L.L. Thompson and D. He. Adaptive space-time element method for the wave equation on unbounded domains. *Comput. Meths. Appl. Mech. Engrg.*, 194:1947–2000, 2005.
- [19] A.M.P. Valli, G.F. Carey, and A.L.G.A. Coutinho. Control strategies for timestep selection in simulation of coupled viscous flow and heat transfer. *Communications in Numerical Methods in Engineering*, 18:131–139, 2002.

- [20] A.M.P. Valli, G.F. Carey, and A.L.G.A. Coutinho. Control strategies for timestep selection in finite element simulation of incompressible flows and coupled reaction-convection-diffusion processes. *International Journal for Numerical Methods in Fluids*, 47:201–231, 2005.
- [21] A.M.P. Valli, G.F. Carey, and A.L.G.A. Coutinho. On decoupled timestep/subcycling and iteration strategies for multiphysics problems. *Communications in Numerical Methods in Engineering*, 2007, to appear.
- [22] P. Ellsiepen W. Ehlers and M. Ammann. Time- and space-adaptive methods applied to localization phenomena in empty and saturated micropolar and standard porous materials. *Int. J. Numer. Meth. Engng*, 52:503–526, 2001.
- [23] S. O. Wille. Adaptive finite element simulations of the surface currents in the north sea. *Computer Methods in Applied Mechanics and Engineering*, 166:379–390, 1998.
- [24] O. C. Zienkiewicz. The background of error estimation and adaptivity in finite element computations. *Comput. Methods Appl. Mech. Engrg.*, 195:207–213, 2006.
- [25] O. C. Zienkiewicz and J.Z. Zhu. A simple error estimator and adaptive procedure for practical engineering analysis. *Int. J. Numer. Meth. Engrg.*, 24:337–357, 1987.