INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids (2009)
Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.1998

REVIEW

PID adaptive control of incremental and arclength continuation in nonlinear applications

A. M. P. Valli[1,*,†], R. N. Elias[2], G. F. Carey[3] and A. L. G. A. Coutinho[2]

[1]*Federal University of Espírito Santo, LCAD, Computer Science Department, Vitória, ES, Brazil*
[2]*Federal University of Rio de Janeiro, NACAD, COPPE, Department of Civil Engineering, Rio de Janeiro, Brazil*
[3]*The University of Texas at Austin, ICES, CFD Lab, Austin, TX, U.S.A.*

SUMMARY

A proportional-integral-derivative (PID) control approach is developed, implemented and investigated numerically in conjunction with continuation techniques for nonlinear problems. The associated algorithm uses PID control to adapt parameter stepsize for branch—following strategies such as those applicable to turning point and bifurcation problems. As representative continuation strategies, incremental Newton, Euler–Newton and pseudo-arclength continuation techniques are considered. Supporting numerical experiments are conducted for finite element simulation of the 'driven cavity' Navier–Stokes benchmark over a range in Reynolds number, the classical Bratu turning point problem over a reaction parameter range, and for coupled fluid flow and heat transfer over a range in Rayleigh number. Computational performance using PID stepsize control in conjunction with inexact Newton–Krylov solution for coupled flow and heat transfer is also examined for a 3D test case. Copyright © 2009 John Wiley & Sons, Ltd.

Received 21 August 2008; Revised 1 December 2008; Accepted 1 December 2008

KEY WORDS: PID control; incremental Newton; arclength continuation; finite element methods; branch-following strategies; nonlinear applications

1. INTRODUCTION

Continuation techniques are fundamental to successful numerical simulation of many nonlinear problems. Of particular interest are steady-state problems where one is seeking to construct approximate solutions of problems that may exhibit multiple solutions for given parameter values and

*Correspondence to: A. M. P. Valli, Rua Constante Sodre, 840/702, Praia do Canto, Vitória, ES 29055-420, Brazil.
†E-mail: avalli@inf.ufes.br

Contract/grant sponsor: CNPq/Brazil; contract/grant number: 620165/2006-5
Contract/grant sponsor: ICES
Contract/grant sponsor: The Curran Chair of Engineering at UT Austin

where the solution behavior is of interest across a wide parameter range. In such cases, it is a common practice to employ some form of incremental continuation, usually in a physical parameter of interest such as the Reynolds number (*Re*) in viscous flow problems, a reaction parameter (λ) in the Bratu problem or the Rayleigh number (*Ra*) in coupled fluid flow and heat transfer considered later. For example, in viscous incompressible Navier–Stokes simulations one may begin at $Re=0$ and compute the simpler linear problem solution for Stokes flow. Then, using this starting solution, the Reynolds number can be incremented and the nonlinear Navier–Stokes problem solved. In turn, this solution becomes the starting iterate for the next incremental step and, so on to determine solutions along a continuation path. Such a set of solutions is also said to define a 'solution branch'. At each step in *Re* the solution obtained at the end of the previous step provides the starting iterate for, say, Newton iterative solution at the next value of *Re*. Provided the step in *Re* is not too large, this procedure assists in providing a starting iterate, that is, in the domain of attraction of Newton's method for each successive nonlinear solve along the path. This type of approach is generally referred to as incremental continuation in the physical parameter. A standard scheme is to select a fixed step size in the parameter and a simple modification is to repeat the step at half the parameter increment if Newton iteration fails. Similarly, one can elect to double the step size if Newton converges in very few iterations. Another extension of this continuation algorithm is obtained by differentiating the nonlinear problem with respect to the parameter, and then integrating numerically with respect to this 'time-like' parameter using, for instance, an Euler integration scheme. This is referred to as an Euler–Newton variant of the continuation scheme. The Euler integration step provides a better starting iterate for Newton iteration at the next parameter level. Once again, the step size can be adjusted adaptively as integration proceeds. This concept is very similar to that used in time integration of evolution problems where the timestep during integration is varied similarly. Of course, there are strategies for estimating time truncation error to adjust the timestep or the order of the time integrator. Moreover, feedback control concepts may be applied to adjust the timestep [1–3]. A key goal in the present study is to extend this feedback control idea to investigate the use of PID control algorithms for parameter step size adaption in continuation approaches for nonlinear problems.

There are other complications, however, that also must be considered in parameterized branch-following algorithms. These include accommodation of multiple solutions for the same parameter values and the corresponding existence of stable and unstable solution branches. Along such branches certain singular turning and bifurcation points may arise and require special treatment. In the representative model Bratu problem considered later, there is a stable solution branch for increasing parameter value that passes through a turning point to become an unstable branch, which can be traversed as the parameter now decreases. The Jacobian matrix in the Newton solver becomes increasingly ill-conditioned as the singular turning point is approached. This has led numerical analysts to devise special algorithms for treating the singularity such as regularizing the problem by introducing a new parameter such as an abstract arclength [4–6] that restores Jacobian rank at the cost of solving a bordered system [7, 8]. Other special algorithms have been devised to improve performance at and near the singular point [7–10]. There are also algorithms that do not solve the bordered system but instead use this form to construct a pseudo-arclength algorithm that also enables monotone step size adjustment. An example of the latter is the algorithm of Keller [4]. This scheme is implemented here in later algorithms and then demonstrated in conjunction with PID control schemes for adaptive continuation of the Bratu turning point problem. This type of algorithm still involves the abstract arclength construction and, hence, the branch behavior is monotone in the new parameterization (distance along the branch). Our main algorithmic approach then entails the following: abstract parameter continuation in a pseudo-arclength parameterization to have a monotone increasing parameter as the path is traced out and use of PID control strategies to adjust step size along the path even at isolated points such as turning and bifurcation points where the step size is automatically reduced to enable passage over the singular point.

The outline of the paper is as follows. In the next section, we consider a representative nonlinear steady-state elliptic partial differential equation (PDE) problem with a parameter and introduce the incompressible viscous flow, reactive transport and coupled viscous flow and heat transfer problem classes to be considered in later numerical experiments. The weak integral formulation and associated Galerkin finite element approximation then are used to construct corresponding nonlinear parameterized sparse algebraic systems. Next, in Section 3, continuation schemes applied in this study are defined. PID feedback control approaches are introduced in Section 4 and algorithms combining PID control with parameterized continuation are developed. More specifically, incremental continuation in the physical parameter is first described with an accompanying algorithm for PID step size control. The arclength and pseudo-arclength parameterizations are then constructed and the PID control scheme for this form is also given. Numerical results for solutions obtained using incremental and pseudo-arclength control solutions of the application problems are presented and performance discussed in Section 5. The main ideas and results are summarized in the concluding remarks.

2. PARAMETERIZED NONLINEAR PROBLEMS

For generality, let us denote the nonlinear parameterized boundary value problem for the steady-state behavior of a physical system in operator form as

$$B(u,\lambda)=0 \tag{1}$$

where B corresponds to the operator for the nonlinear boundary value problem, $u\in V$ is the solution and $\lambda\in R^m$ is a finite dimensional vector of parameters. Since u depends on the parameters λ, there is a *continuum* of solutions parameterized by λ, which is usually referred to as a solution *branch*. Strategies that progressively trace out the solution along a branch as the parameters change are referred to as continuation techniques. Even when the solution is sought for a desired parameter value, continuation along the branch may be a preferable solution strategy because this approach may be more robust than iterating from some fixed starting iterate and multiple solution states may be more readily determined. Moreover, if the continuation step can be controlled reliably then such a step size selection algorithm will also enhance efficiency. Accordingly, here the focus is on automated step size control and PID control is investigated for parameter adaption applied to incremental continuation in branch-following algorithms. Three representative problem classes are selected as test cases: incompressible Navier–Stokes problems with Reynolds number as parameter, reaction–diffusion problems with reaction parameters and viscous flow coupled with heat transfer, where the parameter is the Rayleigh number.

For the first class, the steady Navier–Stokes equations for viscous flow of an incompressible fluid [11, 12] can be written as

$$B(\mathbf{u}(\lambda),\lambda)=\begin{pmatrix}\mathbf{u}\cdot\nabla\mathbf{u}-\dfrac{1}{\lambda}\nabla^2\mathbf{u}+\nabla p-\mathbf{f}\\ \nabla\cdot\mathbf{u}\end{pmatrix}=\mathbf{0} \tag{2}$$

where $\mathbf{u}$ is the velocity vector, p is the pressure, $\mathbf{f}$ is an applied body force and λ is the Reynolds number. Numerical approximations of (2) may be computed using a variety of discretization methods, including the finite element methods considered here. The continuation and control strategies are clearly independent of this choice. The systems of nonlinear equations arising from these approximation schemes are usually solved using an iterative method such as successive approximation or Newton iteration. These methods converge readily from a starting iterate obtained from the solution of the linear Stokes problem for flows at low Reynolds numbers (for which the nonlinear effects are not pronounced). For moderate and high Reynolds numbers, the Stokes solution may not lie in the domain of attraction of the iterative scheme and iteration fails with this starting iterate. The Reynolds number is a logical choice of parameter for formulating a simple incremental continuation method to alleviate this convergence problem. The essential idea is to compute solutions at progressively increasing Reynolds number. The solution at a given intermediate Reynolds number *Re* becomes the starting iterate at the next value of *Re*. Other parameterizations and incremental algorithms are possible, as considered later. In this way continuation techniques provide a means of extending the range of flows for which a solution can be computed.

The model transport equation simplifies in the absence of convection to the stationary reaction–diffusion problem subclass

$$B(u(\lambda),\lambda) = -\nabla\cdot(k\nabla u)+\lambda c(u)-f=0 \tag{3}$$

where k is the diffusion tensor, $c(u)$ is a nonlinear reaction source/sink term, f is a generic source term and λ is the reaction parameter. For example, λ is the Thiele modulus in the catalytic reactions considered in [13]. Such reaction–diffusion equations arise in models of many physical problems of interest in science and engineering, including chemical or biological catalysis, combustion and electrochemistry [8, 9, 14]. In this application class, the reaction functions $c(u)$ are often highly nonlinear, due to the nature of the heat transfer, adsorption, enzymatic or electrochemical effects. Moreover, they may be non-monotone and can have sharp gradients. As a result, multiple solutions often arise for certain ranges of the associated parameters.

The final problem class corresponds to coupled viscous fluid flow and heat transfer, the flow is modeled by the Navier–Stokes equations with temperature-dependent source term characterizing buoyancy effects through a Boussinesq assumption. This in turn is further coupled to the heat transfer equation with convection now entering due to the fluid motion. For example, the resulting coupled system in dimensionless form for the stationary problem becomes

$$B(\mathbf{u}(\lambda),T,\lambda)=\begin{pmatrix}\mathbf{u}\cdot\nabla\mathbf{u}-\nabla^2\mathbf{u}+\nabla p-\dfrac{\lambda}{Pr}T\mathbf{g}\\[2ex] \nabla\cdot\mathbf{u}\\[2ex] \mathbf{u}\cdot\nabla T-\dfrac{1}{Pr}\nabla^2 T\end{pmatrix}=\mathbf{0} \tag{4}$$

where $\mathbf{u}$ is the velocity vector, p is the pressure, λ is the Rayleigh number, Pr is the Prandtl number, T is the temperature and $\mathbf{g}$ is the gravity vector. No slip and no penetration are assumed at wall boundaries. At a free surface, thermocapillary effects generate surface shear stresses that contribute to flow and heat transfer [15–17]. Specified temperature, flux or mixed thermal boundary conditions are applied on appropriate segments of the boundary as in later 2D natural convection simulations. We also consider 3D Rayleigh–Benard simulations using transient computation to a steady state.

2.1. Finite element approximations and solution algorithm

In this work, numerical approximations are computed using the Galerkin finite element method. This finite-dimensional approximate problem can be conveniently expressed in the form

$$G(\mathbf{U},\lambda)=\mathbf{0} \tag{5}$$

where G denotes the discretized nonlinear algebraic residual and $\mathbf{U}$ is a vector of finite element solution coefficients. Certain smoothness properties of G are assumed in later discussion on nonlinear convergence of solver algorithms. For the viscous flow equation (2), a penalty finite element formulation is introduced to enforce incompressibility. Introducing a finite element discretization and basis on the approximate domain Ω_h, the reduced integration penalty finite element approximation problem becomes: for $0<\varepsilon\ll 1$, find $\mathbf{u}_h^\varepsilon \in V^h$ such that

$$\int_{\Omega_h}\left(\frac{1}{\lambda}\nabla\mathbf{u}_h^\varepsilon:\nabla\mathbf{v}_h+(\mathbf{u}_h^\varepsilon\cdot\nabla)\mathbf{u}_h^\varepsilon\cdot\mathbf{v}_h\right)\mathrm{d}\Omega+\frac{1}{\varepsilon}\mathbf{I}(\nabla\cdot\mathbf{u}_h^\varepsilon)(\nabla\cdot\mathbf{v}_h)\,\mathrm{d}\Omega=\int_{\Omega_h}\mathbf{f}\cdot\mathbf{v}_h\,\mathrm{d}\Omega \tag{6}$$

for all $\mathbf{v}_h\in V^h$, where $\mathbf{I}$ denotes reduced numerical integration and ε is the penalty parameter [11]. This leads to the following nonlinear system of algebraic equations:

$$G(\mathbf{U},\lambda)=\frac{1}{\lambda}\mathscr{A}\mathbf{U}+\mathscr{N}(\mathbf{U})+\frac{1}{\varepsilon}\mathscr{B}\mathbf{U}-\mathscr{F}=\mathbf{0} \tag{7}$$

where $\mathbf{U}$ is the vector of velocity finite element coefficients, $\mathscr{A}$, $\mathscr{N}$, $\mathscr{B}$ correspond to the respective viscous, inertial and penalty terms on the left in (6) and $\mathscr{F}$ corresponds to the source term.

The formulation for the model steady scalar transport problem (3) with linear diffusion and nonlinear reaction is constructed similarly: find $u_h\in W^h$ such that

$$\int_{\Omega_h}(k\nabla u_h\cdot\nabla w_h+\lambda c(u_h)w_h-f w_h)\,\mathrm{d}\Omega=0 \tag{8}$$

for all $w_h\in W^h$. The resulting nonlinear system of equation is given accordingly as

$$G(\mathbf{U},\lambda)=\mathscr{K}\mathbf{U}+\lambda\mathscr{C}(\mathbf{U})-\mathbf{F}=\mathbf{0} \tag{9}$$

where $\mathbf{U}$ is the vector of finite element solution coefficients. $\mathscr{K}$, $\mathscr{C}(\mathbf{U})$ corresponds to the respective diffusive and reactive terms on the left in (8) and $\mathbf{F}$ corresponds to the source term. In the above applications, the nonlinear systems (7) and (9) are solved using Newton iteration and one or more of the continuation techniques described in the next section.

The discretized finite element formulation for the coupled flow and transport problem (4) follows similarly from the above two representations and can be treated in the same manner as the above two cases. However, here a different iterative scheme is constructed to solve the stationary problem at each parameter continuation step by implicitly timestepping an associated transient problem to steady state. A real or artificial transient form may be used and Newton iteration is again used within each real or pseudo-timestep. This approach can be an effective alternative to the more common approach of solving the stationary equations by iterative algorithms or using Euler–Newton acceleration if the continuation step is large. In fact, algorithms that employ timestepping to the steady state may be interpreted as another type of continuation scheme in this context. In these cases time-accurate solutions are not sought and this implies further that adaptive timestepping that leads to quite large timesteps may be effective. For some related-work involving explicit time integration schemes that exploit large timesteps see, for instance [18, 19].

Within each timestep, a coupled nonlinear algebraic system associated with the discretized flow and transport equations is to be solved. The main coupling between the flow and transport subsystems enters weakly through the dependence of the source term in the flow equations on the temperature and the convective velocities in the temperature transport. Since the class of applications here does not involve high speed flow, a corresponding iterative block decoupling of the subsystems within each timestep will be effective [3, 20]. That is, the respective discretized flow and transport systems may be decoupled by a successive approximation scheme in which the source term is 'lagged' in the flow equation and the computed velocity iterate is then used in the discretized transport subsystems. This successive approximation iteration is then repeated until convergence. The resulting decoupled flow sub-problem is nonlinear due to the inertial term $\mathbf{u}\cdot\nabla\mathbf{u}$. Here, a penalty method is again implemented to enforce the incompressibility constraint in the 2D flow simulations and pressure stabilization is used in the 3D SUPG stabilized Rayleigh–Benard simulations [21]. The 2D nonlinear sparse systems are solved using successive approximation or Newton's method with sparse elimination of the linearized subsystems whereas the 3D nonlinear systems are solved using inexact Newton–Krylov iteration [22].

3. CONTINUATION TECHNIQUES

The implicit function theorem [23] specifies sufficient criteria guaranteeing that a branch can be parameterized by λ [24]. For a specific stationary solution $(\mathbf{U}^{\lambda_1}, \lambda_1)$ of (5) the criterion basically implies nonsingularity of the Jacobian matrix $G_u(\mathbf{U}^{\lambda_1}, \lambda_1)$. Then there is an interval around λ_1 such that for all λ in that interval Equation (5) has a solution $\mathbf{U}^{\lambda}$ close to $\mathbf{U}^{\lambda_1}$. However, in applications having the form (5) *turning points* and *bifurcations* may occur on the branches in question. At these points, the stability may be lost or the Jacobian G_u may be singular or the structures of the state may change drastically. More specifically, a *bifurcation with respect to* λ corresponds to a specific point (solution at some λ_0) on the branch where there is no neighborhood around λ_0 such that in this neighborhood the branch can be uniquely extended [4, 24, 25].

Obviously, only a finite selection of the infinitely many solutions $(\mathbf{U}, \lambda)$ of (5) can be calculated. That is, the continuous curves in the branching diagrams are approximated by sets of discrete points that are interpolated to approximate the actual curves. The process of tracing a branch by calculating representative solutions $(\mathbf{U}_j, \lambda_j)$ on that branch for $j=0,1,2,\ldots$ with certain distances apart is called *continuation* or *path following*. If a path consists of regular points, the simple continuation technique in the physical parameter described earlier may be applied with some confidence for determining nearby solutions on a branch. In the vicinity of such a solution, Newton's method converges quadratically and the method can produce results very efficiently, albeit at the cost of smaller parameter step size. A complication, however, is that the radius of convergence shrinks as the nonlinearity becomes more pronounced and the method deteriorates as singular points are approached. For instance, iteration may not converge or may converge to a solution on a different branch.

The incremental continuation process above may be improved in some situations as suggested in the Introduction. For example, differentiating the nonlinear system (5) with respect to the continuation parameter, leads to an associated 'initial-value' problem with the parameter as the 'time-like' variable. This may be numerically integrated to give an improved strategy and adaptive timestepping concepts can be applied to vary the step size and obtain a more robust algorithm. A simple approach is to apply explicit Euler forward difference integration to predict the starting iterate for Newton's method at the next parameter level. For example, consider the discretized nonlinear problem (5). Differentiating with respect to the parameter λ

$$\frac{\mathrm{d}G}{\mathrm{d}\lambda} = G_u \frac{\partial \mathbf{U}}{\partial \lambda} + G_\lambda = 0 \tag{10}$$

This defines the linear system

$$G_u \frac{\partial \mathbf{U}}{\partial \lambda} = -G_\lambda \tag{11}$$

to be solved for $\partial \mathbf{U}/\partial\lambda$. Equation (11) involves the calculation of Jacobian matrix G_u as for the original Newton's method, but has a different right-hand side. The benefit of solving this extra system of equations is an improved initial guess for $\mathbf{U}$. This solution can be used in the forward integration rule (Euler's method)

$$\mathbf{U}(\lambda+\Delta\lambda) = \mathbf{U}(\lambda) + \Delta\lambda \frac{\partial \mathbf{U}}{\partial \lambda}(\lambda) \tag{12}$$

to obtain a good starting vector in Newton iteration of the nonlinear system (1) for $\mathbf{U}(\lambda+\Delta\lambda)$. This scheme is termed as the *Euler–Newton* continuation in the physical parameter and is suitable provided that there is no limit points, that is, points where G_u in (11) is singular.

The problem of singular points and of reversal of the branch direction at a turning point leads to consideration of more abstract parameter transformations, the best known being the arclength along the branch. Since the distance along the branch curve monotonically increases, this implies the reversal problem through a turning point is circumvented at the cost of adding an additional arclength equation to the system. The effect of the additional equation is also to restore the rank deficiency in the Jacobian at the singular point. Pseudo-arclength forms such as the scheme due to Keller [4] are implemented here and preserve the monotonicity, but require step adjustment to cross the singular point.

In an arclength continuation method, (5) is augmented with a constraint equation to obtain the expanded nonlinear system

$$P(\mathbf{U}, \lambda, s) = \begin{bmatrix} G(\mathbf{U}, \lambda) \\ N(\mathbf{U}, \lambda, s) \end{bmatrix} = 0 \tag{13}$$

where the arclength s is defined as the distance along the solution branch. It follows that s is non-decreasing even if λ is decreasing along the branch.

In the present work, a linearized arclength form is used that has been termed a 'pseudo-arclength' scheme and is given by

$$N(\mathbf{U}, \lambda, s) = (\mathbf{U} - \mathbf{U}(s_i))^t \left.\frac{\partial \mathbf{U}}{\partial s}\right|_{s_i} + (\lambda - \lambda(s_i)) \left.\frac{\partial \lambda}{\partial s}\right|_{s_i} - (s - s_i) = 0 \tag{14}$$

where s_i corresponds to the pseudo-arclength distance from the starting point to the current step at point i along the path. The corresponding augmented Jacobian system for the expanded system (13) is then used to obtain the increment in the solution pair $(\mathbf{U}, \lambda)$ at each arclength step via Newton's method as follows: for iterate $k=0,1,2,\ldots$ until stopping criteria, solve

$$\begin{bmatrix} G_u & G_\lambda \\ N_u^t & N_\lambda \end{bmatrix}^k \begin{bmatrix} \Delta\mathbf{U} \\ \Delta\lambda \end{bmatrix}^{k+1} = \begin{bmatrix} -G \\ -N \end{bmatrix}^k \tag{15}$$

where $\Delta\mathbf{U}$, $\Delta\lambda$ are the solution increments corresponding to the specified step in pseudo-arclength. Several schemes are available in the literature for solving (15) efficiently by direct or iterative schemes as a sparse, bordered system. Here, for convenience (in terms of reuse of existing computational tools) the following two-solve procedure is applied. At each Newton step, the following computations are performed:

```
Algorithm:
1. Solve $G_u y = G_\lambda$ (first solve).
2. Solve $G_u z = -G$ (second solve).
3. Calculate $\Delta\lambda = \frac{-N - N_u^t z}{N_\lambda - N_u^t y}$.
4. Calculate $\Delta\mathbf{U} = z - (\Delta\lambda) y$.
```

The algorithm must be employed at each Newton step. To see how the algorithm arises, multiply the first 'row' of Equation (15) by G_u^{-1}

$$\Delta\mathbf{U} + (\Delta\lambda) G_u^{-1} G_\lambda = -G_u^{-1} G \tag{16}$$

Observe that the inverse Jacobian is not explicitly computed. Instead one computes its action. Using the definitions of y and z given in the algorithm (steps 1 and 2),

$$\Delta\mathbf{U} + (\Delta\lambda) y = z \tag{17}$$

or the expression of $\Delta\mathbf{U}$ (step 4). Now, from the second 'row' of Equation (15),

$$N_u^t (z - (\Delta\lambda) y) + (\Delta\lambda) N_\lambda = -N \tag{18}$$

which can be rearranged to obtain the scalar equation for $\Delta\lambda$ (step 3). Here the systems are solved with two different right-hand size vectors G_λ and $-G$. In this context the two-step algorithm is roughly as expensive as a normal Newton solve and, therefore, attractive to use near turning points or regions of rapid change in the solution relative to the parameter λ. Observe also that the initialization of the arclength continuation algorithm requires the solution and parameter derivatives with respect to s from the previous step, since the linearized arclength form (14) is used. Hence, it is assumed that two previous solutions, $\{\mathbf{U}_0, \lambda_0\}$ and $\{\mathbf{U}_1, \lambda_1\}$, have been computed using the incremental Newton scheme with small increments of the parameter λ.

An extension of the Euler–Newton approach may again be used to generate an initial guess for Newton iteration with the arclength parameterization. The main idea is to assume that the 'derivative' of $P(x(s))$, $x=[\mathbf{U}, \lambda]$, with respect to s is the zero vector. Chain-differentiation of P in (13) yields

$$\begin{bmatrix} G_u & G_\lambda \\ N_u^t & N_\lambda \end{bmatrix} \begin{bmatrix} \dfrac{\partial \mathbf{U}}{\partial s} \\ \dfrac{\partial \lambda}{\partial s} \end{bmatrix} = \begin{bmatrix} 0 \\ -N_s \end{bmatrix} \tag{19}$$

and determines the tangent vector. Then a predictor or initial guess for Newton's method can be obtained from

$$\mathbf{U}_2^* = \mathbf{U}_1 + \frac{\partial \mathbf{U}}{\partial s}\Delta s, \quad \lambda_2^* = \lambda_1 + \frac{\partial \lambda}{\partial s}\Delta s \tag{20}$$

where Δs is the step in arclength. Observe that (19) conveniently involves the same matrix as the augmented Jacobian system of (15). Therefore, to estimate the initial guess, the previous 'two-solve' algorithm is again applied. This retains monotonicity in the continuation variable and also simplifies the implementation, because of the close similarity to the incremental scheme for non-singular problems. To adapt the arclength step size for overall efficiency and to maintain convergence near the singular point, a PID feedback control algorithm is introduced. The step selection criterion is based on controlling accuracy using truncation error estimates for integration of (19) with respect to the continuation variable as discussed next.

4. PID STEP SIZE CONTROL

Feedback control strategies such as PID controllers are extensively used in traditional control applications. They have recently become a research topic of interest in timestep control for transient PDE integration and are introduced in the present work for parameter control in continuation algorithms. Such PID control methods provide a means of timestep selection in artificial transient algorithms [3, 26] and here for extending the range of steady parameterized flows for which a solution can be computed. A number of viable control forms are possible. For illustrative purposes here, a single PID control form is presented below for both timestep selection and adaptive continuation. More specifically, the step size can be defined as follows:

$$\Delta[\cdot]_{\text{stage}+1} = \left(\frac{e_{\text{stage}-1}}{e_{\text{stage}}}\right)^{k_P} \left(\frac{1}{e_{\text{stage}}}\right)^{k_I} \left(\frac{e_{\text{stage}-1}^2}{e_{\text{stage}} e_{\text{stage}-2}}\right)^{k_D} \Delta[\cdot]_{\text{stage}} \tag{21}$$

where

1. For timestep selection, $\Delta[\cdot]=\Delta t$ represents the timestep size (or a related quantity, such as Courant–Friedrichs–Lewy (CFL)) and $e_{\text{stage}}=e_n$ is the measure of the normalized change of the quantities of interest in time t_n. For example, in the natural convection problem considered later, the control uses the normalized changes in velocities ($\mathbf{U}$) and temperature ($\mathbf{T}$)

$$e_n = \max\left(\frac{e_u}{\text{tol}_u}, \frac{e_T}{\text{tol}_T}\right), \quad e_u = \frac{\|\mathbf{U}^n - \mathbf{U}^{n-1}\|}{\|\mathbf{U}^n\|}, \quad e_T = \frac{\|\mathbf{T}^n - \mathbf{T}^{n-1}\|}{\|\mathbf{T}^n\|} \tag{22}$$

and can be motivated based on the need to control accuracy with respect to time in the specific solution variables. Here the controller uses two supplied tolerances, tol_u and tol_T, corresponding to the normalized changes in velocities and temperature vectors, respectively.

2. For incremental continuation, $\Delta[\cdot]=\Delta\lambda$ represents the physical parameter increment and $e_{\text{stage}}=e_{\lambda_j}$ is the measure between two consecutive solutions on the branch with certain distance apart. For example, in the lid-driven cavity problem presented later, the Reynolds number ($\lambda=Re$) is incremented based on the changes of two consecutive steady-state velocities in the continuation process and the controller uses one supplied tolerance, tol_λ, as shown below

$$e_{\lambda_j}=\frac{e^*_{\lambda_j}}{\mathrm{tol}_\lambda},\quad e^*_{\lambda_j}=\frac{\|\mathbf{U}^{\lambda_j}-\mathbf{U}^{\lambda_{j-1}}\|}{\|\mathbf{U}^{\lambda_j}\|} \tag{23}$$

3. For arclength continuation, $\Delta[\cdot]=\Delta s$ represents the arclength parameter increment and $e_{\text{stage}}=e_{s_k}$ is the measure of the normalized changes of the most recent solution tangents with respect to the parameter λ,

$$e_{s_k}=\frac{e^*_{s_k}}{\mathrm{tol}_s},\quad e^*_{s_k}=\frac{\left\|\left.\frac{\partial u}{\mathrm{d}\lambda}\right|_{s_k}-\left.\frac{\partial u}{\mathrm{d}\lambda}\right|_{s_{k-1}}\right\|}{\left\|\left.\frac{\partial u}{\mathrm{d}\lambda}\right|_{s_k}\right\|} \tag{24}$$

where tol_s is a user-supplied tolerance. The motivation is to control the size of the arclength step size along the branch as the solution changes with respect to the parameter λ. In turn, controlling the arclength step size leads to a more robust nonlinear iterative behavior. This is particularly important near bifurcation and turning points, since the character of the solution typically changes rapidly in these regions and the transition from one solution regime to another is often of special interest.

The efficiency of this form of PID control for timestep selection was demonstrated in [1, 3, 20]. The computational overhead in (21)–(24) is insignificant compared with solver operations, since the step size selection involves only storing a few extra vectors and computation of associated norms. To prevent an excessive growth or reduction of the step in the controller, extreme values $[\cdot]_{\min}$ and $[\cdot]_{\max}$ are set, which limit the control signal (anti-windup effect [27]). The effect of the anti-windup is to reduce both overshoot and the control effort in the feedback system. Parametric studies were performed for different values of PID parameters ($k_{\mathrm{P}}, k_{\mathrm{I}}, k_{\mathrm{D}}$) for two test problems, to verify that the PID controller is robust [3]. Although feedback control theory provides techniques to choose PID parameters, robustness is required when a general method is used for a wide range of different situations. The controller was found to be very robust, allowing us to fix the values of the PID parameters, $k_{\mathrm{P}}=0.075$, $k_{\mathrm{I}}=0.175$ and $k_{\mathrm{D}}=0.01$, for all the numerical experiments described in the following section. Note that PID control in both timestep and parameter or arclength can be combined into a hybrid PID selection process as illustrated in the natural convection problem considered later. Moreover, one can combine such adaptive continuation procedures with inexact Newton–Krylov and other solver strategies to further enhance performance.

5. NUMERICAL STUDIES

5.1. The lid-driven cavity problem

The two-dimensional 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 [28–30]. The domain of analysis is a unit square. Both velocity components are prescribed to be zero, except at the top boundary (the lid) where the horizontal velocity component has a unit value. However, this problem is complicated by the presence of two corner singularities, which may be regularized in various ways (e.g. see [28, 31]). Here, the latter continuous hyperbolic tangent approximation

$$u(x)=\begin{cases}\tanh(\beta x), & 0\leqslant x\leqslant 0.5\\ -\tanh(\beta(x-1)), & 0.5<x\leqslant 1.0\end{cases} \tag{25}$$

is used for the horizontal velocity component on the upper boundary of the domain with regularization parameter $\beta=100$ chosen to give a sharp transition from $u=0.0$ to $u=1.0$ near the corners. For representative problems with low Reynolds numbers (e.g. 300), solutions can be achieved in few iterations using successive approximation or Newton–Raphson iteration starting from Stokes flow. Figure 1 (left) shows how these two nonlinear iterative schemes perform at several Reynolds numbers for solution with a coarse uniform mesh of bilinear elements ($h=\frac{1}{16}$). As anticipated, the schemes fail to converge as the convective nonlinearity becomes more pronounced with increasing *Re* ($Re\geqslant 1000$ or $\log(Re)\geqslant 3$ as shown in Figure 1 (left)). In the vicinity of a solution, Newton's method converges quadratically and the method can produce results very efficiently. However, the domain of attraction is small for the flows considered here. To obtain solutions at higher Reynolds numbers, an appropriate starting guess in the domain of attraction of the desired root can often be obtained using a continuation strategy as indicated in Figure 1 (right).

Here, as *Re* increases the convective term becomes more significant, thus the mesh size must be reduced accordingly to prevent oscillations and retain accuracy and convergence. The present studies at $Re=12\,500$ (or $\log(Re)\approx 4.1$) are computed on a 128×128 grid of bilinear elements in

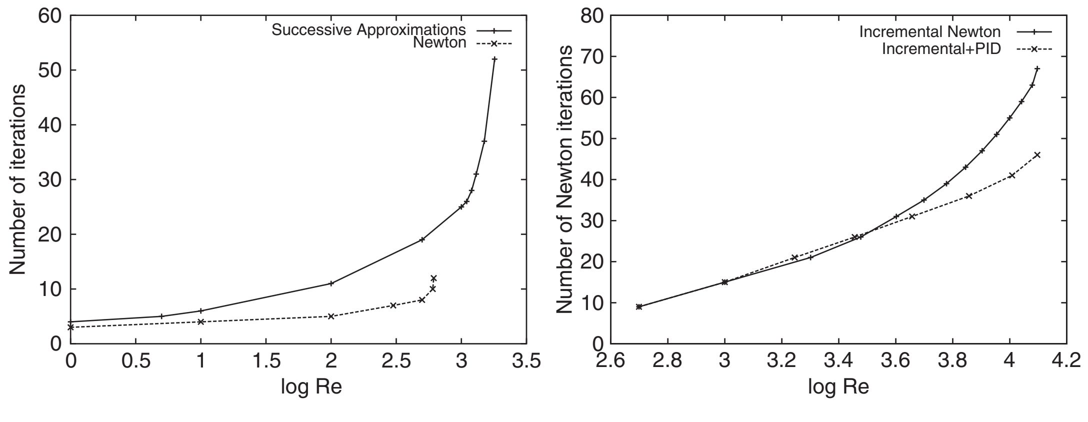

Figure 1. Total number of iterations from a Stokes flow initial iterate as *Re* increases using successive approximation and Newton iteration (left), total number of Newton iterations using uniform incremental and adaptive PID-incremental continuation (right).

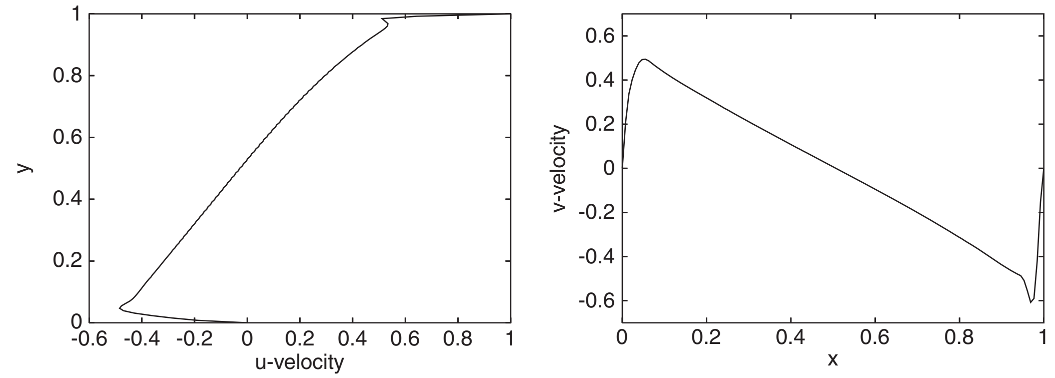

Figure 2. Computed horizontal velocity component along the vertical centerline (left) and vertical velocity component along the horizontal centerline (right) for $Re = 12\,500$.

all cases. For fixed continuation step sizes in Reynolds number, the first step *Re* is 500 (close to the maximum *Re* value for convergence from Stokes flow) and, subsequently, fixed increments of 1000 are taken. For the PID adaptive step size approach, the minimum and maximum allowable increments for *Re* are 500 and 3000, respectively, and the tolerance for allowable changes between two consecutive steady state velocities in the continuation process is $\text{tol}_\lambda = 1$. Since the size of the steady-state velocities increases as the Reynolds number increases in the continuation process, we can choose the tolerance in this example to be even bigger than 1. Figure 1 (right) shows the number of total iterations when the *Re* number increases using fixed and adaptive increments. For $Re > 1000$ standard Newton iteration from zero velocity fails to converge whereas PID continuation techniques permit more efficient solution for much higher Reynolds numbers. The computational effort here is measured by the total number of Newton iterations needed to calculate the target solution using the PID incremental approach divided by the number of Newton iterations obtained using a fixed increment size. Clearly this measure is essentially independent of the choice of iterative or elimination solver for the linearized Jacobian subsystem solves provided these subsystems are accurately solved. In the present 2D calculations, relatively small Jacobian subsystems are solved thus sparse elimination solvers are suitable. Here, the PID incremental solution for $Re = 12\,500$ is obtained in 46 Newton iterations, reducing the computational cost by about 31% when compared with fixed increments of $Re = 1000$. Figure 2 shows the 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 $Re = 12\,500$. The agreement is favorable when compared with results in [32, 33]. The value of the stream function at the primary vortex center differs by order 5×10^{-3} from that of the p-type finite element scheme with $h = \frac{1}{32}$, $p = 8$ in a fully coupled stream function-vorticity formulation used in [32].

5.2. Bratu problem

The Bratu reaction–diffusion problem [7, 8]

$$-\nabla^2 u - \lambda e^u = 0 \quad \text{on } (0,1)\times(0,1) \tag{26}$$

$$u = 0 \quad \text{on } \partial\Omega \tag{27}$$

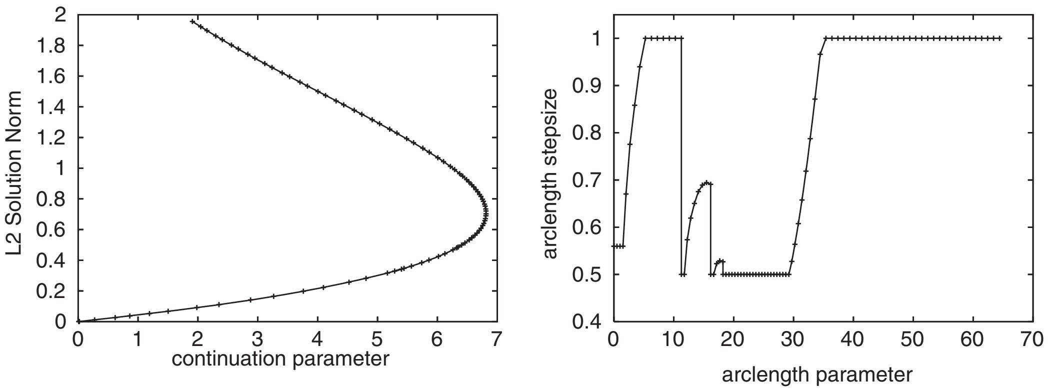

Figure 3. The main branch of the solution (left) and the arclength step size variation (right).

has a turning point near $\lambda = 6.81$. The arclength approach with PID control is applied here to follow the main branch of solutions with minimum and a maximum arclength step sizes of 0.5 and 1.0, respectively, and a tolerance of 0.1 in the change of the solution tangents with respect to the parameter λ. Isoparametric four-node quadrilateral elements on a uniform 32×32 mesh are used. The Euler–Newton approach is used with the maximum number of Newton iterations allowed set to 20 and a Newton tolerance of 10^{-10}. Figure 3 shows the main branch of solutions and the arclength step size variation. With PID feedback, the arclength step size rapidly grows from the initial value to the maximum value until λ reaches approximately 5 (corresponding to an arclength parameter greater than 10). Following this, the PID controller reduces the arclength step to the minimum value to follow the solution branch through the turning point and then allows the step size to increase again. Using the PID control, the turning point is estimated to be $\lambda = 6.81278399$.

5.3. Natural convection

The next case study involves natural convection in a unit square $\Omega = [0, 1] \times [0, 1]$ with temperatures $T = 1$, $T = 0$ on the left and right side walls, respectively, adiabatic top and bottom walls, Prandtl number $Pr = 0.71$. Incremental continuation in the Rayleigh number Ra is applied. The computed Nusselt number at the left wall, $Nu_0 = \int_0^1 q\,\mathrm{d}y$, where q is the heat flux and the stream function at the midpoint, ψ_{mid}, are compared with benchmark computations in [34, 35]. The benchmark case reports the quantities to four significant figures, and the reported accuracy is within 1% for all Rayleigh numbers used in the experiments. Here, the solutions are compared with $Ra = 100\,000$ for calculations with 9-node isoparametric quadrilateral elements on a uniform 32×32 mesh.

First, the associated transient formulation of the coupled problem (4) is solved to steady state for $Ra = 100\,000$ using fixed timestep sizes of 10^{-3}, $\mathrm{tol}_{st} = 10^{-4}$ from an initial state: $u(x, y, 0) = 0$, $T(x, y, 0) = 1 - x$, in domain Ω. This solution is time accurate and the transient response is oscillatory especially in the early stages. There is a slow final asymptotic approach to the steady state as transients decay. This scheme does not use the continuation techniques and will be denoted in the experiments as the *baseline* approach. The problem is next solved with parameter continuation

Table I. Computational effort for the natural convection problem.

	Baseline	PID-*Ra*	PID-*Ra*-time	
		$\text{tol}_\lambda = 1.0$	$\text{tol}_\lambda = 1.0$	$\text{tol}_\lambda = 1.5$
nsa	416	396	462	390
Nu_0 (error)	4.551 (0.93%)	4.552 (0.95%)	4.547 (0.84%)	4.552 (0.93%)
ψ_{mid} (error)	9.083 (0.31%)	9.081 (0.33%)	9.111 (0.00%)	9.080 (0.34%)

using the PID control algorithm to select *Ra* increment size, d*Ra* and with fixed timestep size. This yields a sequence of intermediate steady-state solutions in the continuation process, each solution initiated from the previous steady state in the sequence. This second solution strategy is termed the PID-*Ra* approach and can be viewed as a continuation iterative scheme for computing steady states at a sequence of *Ra* values of interest. Finally, the problem may be solved using PID control to select both the *Ra* increments for continuation and to adapt the timestep size. This last solution type is referred to as the PID-*Ra*-time approach. For brevity and convenience, in the following numerical results we invoke PID timestep adaption only in the last (largest) step of the *Ra*-continuation scheme. This allows us to examine use of *PID-Ra* through a sequence of steady states followed by a single PID timestep adaption stage to approach the final steady state at $Ra = 100\,000$. Obviously, other combined strategies to study acceleration to steady states may be considered.

Table I shows the total number of successive iterations, nsa, ψ_{mid}, Nu_0 and the percentage errors compared with the benchmark solution [34, 35] ($\psi_{\text{mid}} = 9.111$ and $Nu_0 = 4.509$), calculated using the three approaches for two parameter tolerances, $\text{tol}_\lambda = 1.0, 1.5$. In this application, the PID continuation process controls changes in two consecutive velocities and temperature. In both PID-*Ra* and PID-*Ra*-time approaches, the minimum and maximum parameter increments allowed are $dRa_{\min} = 1000$ and $dRa_{\max} = 50\,000$, respectively. Fixed timestep sizes of 10^{-2} are used to calculate the solution for $Ra < 10^4$ and then the timestep size is set equal to 10^{-3}. The respective steady-state tolerances are of the order of $\text{tol}_{st} = 10^{-1}$ and 10^{-4}. In the PID timestep selection, the minimum and maximum timestep sizes allowed are $ht_{\min} = 10^{-3}$ and $ht_{\max} = 10^{-1}$, respectively, and $\text{tol}_T = 10^{-3}$ and $\text{tol}_u = 5 \times 10^{-3}$ for $\text{tol}_\lambda = 1.0$ and $\text{tol}_T = 5 \times 10^{-2}$ and $\text{tol}_u = 10^{-1}$ for $\text{tol}_\lambda = 1.5$.

Observe from Table I that the PID continuation solutions are in good agreement with the target benchmark steady-state solution for all cases, with percentage errors no more than 1% in all quantities. For the PID-*Ra*-time approach with $\text{tol}_\lambda = 1.0$, the ψ_{mid} percentage error is zero but more successive approximation iterations are needed to obtain the steady-state solution. This is expected since smaller PID tolerances were prescribed for timestep selection in this particular experiment. Nevertheless, the PID performance is competitive. For the other cases, the target solution is obtained with almost the same precision as in the *baseline* approach, but using a smaller number of successive iterations. Of course, the goals for the respective strategies may be quite different—e.g. time accurate solution to a single target steady state versus rapid computation of a sequence of steady states.

This natural convection example also demonstrates the PID concepts for a familiar coupled physics problem of practical interest. The post-processed kinetic energy is a quantity of interest for this problem class [36] and allows convenient visualization of the approach to steady state.

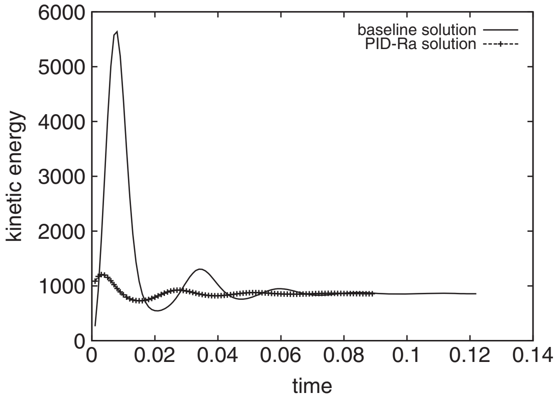

Figure 4. Transient kinetic energy for the target solution, at $Ra=10^5$ using the *baseline* approach and continuation path through 13 d*Ra* stages and intermediate 'timestep' iterates using the PID-*Ra* approach with $\text{tol}_\lambda=1.0$.

The kinetic energy is defined here as

$$K=\int_\Omega \frac{(u^{*2}+v^{*2})}{2}\,\mathrm{d}\Omega \tag{28}$$

where u^* and v^* are the nondimensional x-and y-components of the velocity. In particular, the associated transient dynamics can be characterized conveniently for different parameter regimes using this post-processed quantity [3, 20]. Figure 4 shows results from two approaches: (1) the oscillatory, time accurate, transient kinetic energy that eventually decays to the target solution value using the *baseline* approach and (2) a sequence of intermediate steady-state continuation solutions computed using the PID-*Ra* approach with $\text{tol}_\lambda=1.0$. The PID-*Ra* algorithm starts from initial data $u(x,y,0)=0$ and $T(x,y,0)=1-x$ and the PID control chooses *Ra* increments toward the target value $Ra=100\,000$. In the present calculation 13 monotonically increasing steps are taken with incremental values of d*Ra* selected by the PID control ranging from 1000 to 20 707. Both the 13 steady-state values and intermediate 'timestep' iterates are shown in the figure.

5.4. 3D Rayleigh–Benard problem

In the last test case we consider the combined use of PID control for both continuation step size and timestep adaption to steady state. The underlying sparse linearized subsystem solver for these 2D decoupled subsystems employed frontal elimination in serial computations. Clearly, the PID control strategies that are the focus of the present paper are equally applicable using elimination and iterative solution algorithms. If one computes using, say, a preconditioned Krylov iterative solver or an elimination solver to high accuracy in a step then the PID behavior will be approximately the same for both linear sub-solvers. Of course, the total central processing unit (CPU) time will depend on the number of solves resulting from the step size selection and the number of nonlinear iterations needed for each solve. There will be some tradeoff here since a larger step size will require more Newton iterations from the same starting iterate. However, near a root, Newton converges quadratically thus this enhances the adaptive step size selection strategy.

The cost of repeated Jacobian construction, relaxing the iterative tolerance on intermediate solutions and inexact Newton strategies with Krylov iterative subsystem solves can also be exploited to improve overall CPU efficiency. Again, these ideas are to some extent independent of the adaptive step size control aspect. The relative CPU cost would also depend on the problem class, preconditioner effectiveness, size of the discretized algebraic systems and computer hardware. For large-scale problems memory constraints suggest using the Krylov schemes with an appropriate preconditioner. E.g. one of our ongoing collaborations involves large-scale parallel AMR simulations with PID continuation using libMesh with Petsc [37] to provide the iterative solver library.

A related study of 3D Rayleigh–Benard convection with PID control of timestep selection has been carried out to investigate the performance of the PID strategy in conjunction with inexact Newton using Krylov subsystem solves. The test case corresponds to a rectangular 3D domain of aspect ratio 4:1:1 aligned with the Cartesian axes and subjected to a temperature gradient [38]. Simulations are made on a mesh of 93 925 tetrahedral elements using an edge-based Galerkin SUPG scheme with pressure stabilization. Further details of the finite element formulation are provided in [39, 40]. A converged stationary solution, shown in Figure 5, with four convective rolls is obtained at Rayleigh number $Ra=30000$ and Prandt number $Pr=0.71$ [41].

Performance results for this simulation with PID timestep control using inexact Newton–Krylov solution for simulation to steady state are given in Table II. We control here a CFL condition and investigate the interplay between choosing adaptively the timestep and the tolerance for an inexact Newton solution of flow equations. For more details on the inexact Newton method used here, see [42]. The inner iterative driver is an edge-based preconditioned GMRES method. A nodal block-diagonal preconditioner is used for the Navier–Stokes equations while a simple diagonal preconditioning is employed for the temperature equation. GMRES tolerance for the temperature is fixed at 10^{-3} while maximum tolerances for the inexact Newton method vary as indicated in Table II. For both flow and transport, the number of Krylov vectors is fixed in 25. We may observe in this table that the smallest CPU time (wall time) is achieved when we choose a maximum inexact Newton tolerance of 0.01 and CFL chosen adaptively between two and five. In this case

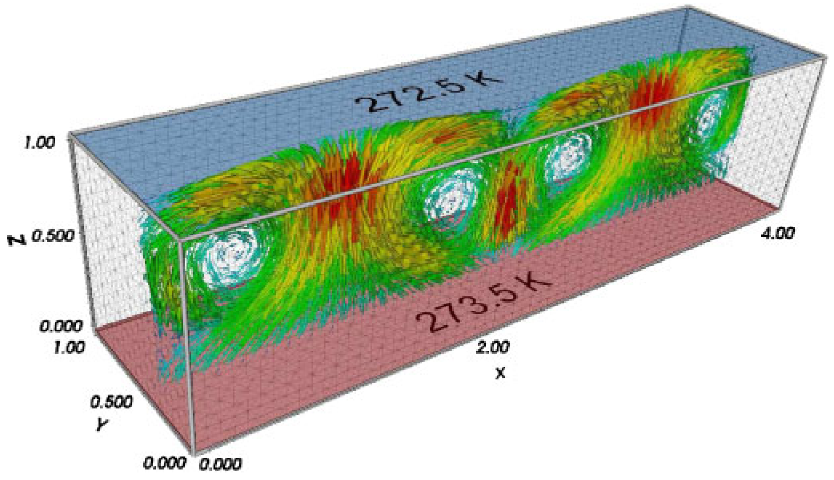

Figure 5. 3D Rayleigh–Benard solution at $Ra=30000$ and $Pr=0.71$.

Table II. Computational effort for the 3D Rayleigh–Benard problem.

Max IN tol	$\mathrm{CFL}_{\min}$	$\mathrm{CFL}_{\max}$	Timesteps	Wall time
0.001	2	10	109	1110.49
		5	90	855.42
		2	171	1184.81
0.01	2	10	108	803.89
		5	90	637.71
		2	172	1017.02
0.99	2	10	142	1652.06
		5	90	991.96
		2	171	1501.97

steady state is reached in 90 timesteps. We consider here that steady state is achieved when the relative velocity increment differs by less than 10^{-5}. Note that the best performances are obtained by setting the maximum tolerance for the inexact Newton method to moderate values, indicating that a too thigh value will result in over solving and too loose values will result in bigger timesteps leading in both cases to comparatively bigger computational efforts.

6. CONCLUDING REMARKS

Continuation techniques are fundamental to successful numerical simulation of many nonlinear problems. Of particular interest here are steady-state problems where one is seeking to construct approximate solutions of problems exhibiting multiple solutions at given parameter values and where the solution behavior is of interest across a wide parameter range. A continuation strategy in physical and arclength parameter form employing PID control for parameter step size adjustment is developed and applied to incompressible Navier–Stokes, a reaction–diffusion problem with a turning point and two coupled viscous flow and heat transfer problems. The advantageous performance of the PID strategy as compared with simpler incremental continuation is demonstrated. The PID continuation techniques permit more efficient solution for much higher *Re* numbers in the driven cavity problem, reducing the computational cost by about 31% when compared with fixed increments. In the natural convection problem, the adaptive PID continuation approach generates a sequence of intermediate steady-state solutions approaching the target steady-state solution at $Ra = 100\,000$, leading to a more robust and efficient strategy for this coupled system. The continuation solutions are in good agreement with the benchmark solution for all cases, with percentage errors no more than 1% in the computed Nusselt number and stream function at the midpoint. The PID feedback control algorithm is also seen to be a good technique for adapting the pseudo-arclength for overall efficiency and to maintain convergence near the singular point in the Bratu problem. The arclength approach with PID control is applied to automatically follow the main branch of solutions and to enable passage over the turning point.

Clearly, the PID control strategies are equally applicable using elimination and iterative solution algorithms. PID offers a different form of adaptive control along the branch or with respect to time in the present context and changes in the solution behavior influence continuation step size selection. The relative performance enhancement will depend on the reduction in the number of nonlinear solves and on any net increase in the number of nonlinear iterations for each solve. We have provided a comparison with uniform stepping to indicate representative performance. The total solution time will also depend on the efficiency of the linear subsystem solver. If one computes using, say, a preconditioned Krylov iterative solver [43–45] or an elimination solver to high accuracy in a step then the PID behavior will be approximately the same for both linear sub-solvers. The CPU time would be different of course. This aspect is not a focal point of the present study but clearly is important in practice. The relative CPU cost would depend on the problem class, preconditioner effectiveness, size of the discretized algebraic systems and computer hardware. Most of the test cases considered here are 2D and some involve decoupled physics. Hence, they are not large-scale and elimination solvers are competitive. For very large-scale problems, memory constraints suggest using Krylov schemes with an appropriate preconditioner. To illustrate this point, we have included some tabulated performance results from a 3D Rayleigh–Benard simulation using PID control with inexact Newton–Krylov solution. Other ongoing work involves further large-scale parallel AMR simulations with PID continuation using libMesh with Petsc [37] for the iterative solver library.

ACKNOWLEDGEMENTS

This work was developed under the Joint Cooperation Agreement between the University of Texas at Austin and COPPE, the Graduate School of Engineering of the Federal University of Rio de Janeiro, Brazil. This work has been supported in part by CNPq/Brazil, Proc. 620165/2006-5, ICES and the Curran Chair of Engineering at UT Austin.

REFERENCES

1. Valli AMP, Carey GF, Coutinho ALGA. Control strategies for timestep selection in simulation of coupled viscous flow and heat transfer. *Communications in Numerical Methods in Engineering* 2002; **18**:131–139.
2. Soderlind G. Digital filters in adaptive time-stepping. *ACM Transactions on Mathematical Software* 2003; **29**:1–26.
3. Valli AMP, Carey GF, Coutinho ALGA. 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* 2005; **47**:201–231.
4. Keller HB. Numerical solution of bifurcation and nonlinear eigenvalue problems. In *Applications of Bifurcation Theory*, Rabinowitz PH (ed.). Academic Press: New York, 1977; 359–384.
5. LOCA Library. Web site http://www.cs.sandia.gov/loca/. Visited 02-08-2008.
6. Salinger AG, Bou-Rabee NM, Pawlowsk RP, Wilkes ED, Burroughs EA, Lehoucq RB, Romero LA. Loca 1.0 library of continuation algorithms: theory and implementation manual. *Technical Report SAND2002-0396*, Sandia National Laboratories, Albuquerque, NM, Livermore, CA, 2002.
7. Barragy E, Carey GF. A partitioning scheme and iterative solution for sparse bordered systems. *Computer Methods in Applied Mechanics and Engineering* 1988; **70**:321–327.
8. Barragy E, Carey GF. Bifurcation detection using the Lanczos method and imbedded subspaces. *Impact of Computing in Science and Engineering* 1991; **3**:76–92.
9. Carnes BR, Carey GF. Estimating spatial and parameter error in parameterized nonlinear reaction–diffusion equations. *Communications in Numerical Methods in Engineering* 2007; **23**(9):835–854.
10. Griewank A, Reddien G. The calculation of HOPF points by a direct method. *IMA Journal of Numerical Analysis* 1983; **3**:295–303.
11. Carey GF, Oden JT. *Finite Elements*: *Fluid Mechanics*, vol. 6. Prentice-Hall Inc.: Englewood Cliffs, NJ, 1986.
12. Hughes TJR. *The Finite Element Method*. Prentice-Hall Inc.: Englewood Cliffs, NJ, 1987.
13. Finlayson BA, Carey GF. Orthogonal collocation on finite elements. *Journal of Chemical Engineering Science* 1975; **30**:587–596.
14. Finlayson BA. *The Method of Weighted Residuals and Variational Principles*, *Mathematics in Science and Engineering*, vol. 87. Academic Press: New York, NY, 1972.

15. Carey GF, Harlé C, McLay R, Swift S. MPP solution of Rayleigh–Benard–Marangoni flows. *Supercomputing 97*, San Jose, CA, 1997; 1–13.
16. Carey GF, McLay R, Bicken G, Barth B, Swift S, Ardelea A. Parallel finite element solution of three-dimensional Rayleigh–Benard–Marangoni flows. *International Journal for Numerical Methods in Engineering* 1999; **31**:37–52.
17. Davis MB, Carey GF. Parallel multilevel solution of Rayleigh–Benard–Marangoni problems. *International Journal for Numerical Methods of Heat and Fluid Flow* 2000; **10**(3):248–267.
18. Lorber AA, Carey GF, Joubert WD. ODE recursions and iterative solvers for linear equations. *SIAM Journal on Scientific Computing* 1996; **17**(1):65–77.
19. Bova SW, Lorber AA, Carey GF. *An RK Iterative Recursion and SUPG Method for the Stationary Shallow Water Equations*. Computational Mechanics Publications: Southampton, U.K., 1995; 47–54.
20. Valli AMP, Carey GF, Coutinho ALGA. On decoupled timestep/subcycling and iteration strategies for multiphysics problems. *Communications in Numerical Methods in Engineering* 2008; **24**(12):1941–1952.
21. Brooks AN, Hughes TJR. Streamline upwind/Petrov–Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier–Stokes equations. *Computer Methods in Applied Mechanics and Engineering* 1982; **32**:199–259.
22. Shadid JN, Tuminaro RS, Walaker HF. An inexact Newton method for fully-coupled solution of the Navier–Stokes equations with heat and mass transport. *Technical Report SAND97-0132*, Sandia National Laboratories, Albuquerque, NM, Livermore, CA, 1997.
23. Kumagai S. An implicit function theorem: comment. *Journal of Optimization Theory and Applications* 1980; **31**(2):285–288.
24. Seydel RU. *Practical Bifurcation and Stability Analysis*: *From Equilibrium to Chaos*. Springer: New York, NY, 1994.
25. Seydel RU. Numerical computation of branch points in ordinary differential equations. *Numerische Mathematik* 1979; **32**:51–68.
26. Gustafsson K, Lundh M, Soderlind G. A PI stepsize control for the numerical solution for ordinary differential equations. *BIT* 1988; **28**:270–287.
27. Franklin GF, Powell JD, Emami-Naeini A. *Feedback Control of Dynamic Systems*. Addison-Wesley Publishing Company: Reading, MA, 1994.
28. de Sampaio PAB, Lyra PRM, Morgan K, Weatherill NP. Petrov–Galerkin solutions of the incompressible Navier–Stokes equations in primitive variables with adaptive remeshing. *Computer Methods in Applied Mechanics and Engineering* 1993; **106**:143–178.
29. Spotz WF, Carey GF. High-order compact scheme for the esteady stream-function vorticity equations. *International Journal for Numerical Methods in Engineering* 1995; **38**:3497–3512.
30. Ghia U, Ghia KN, Shin CT. High resolutions for incompressible flow using Navier–Stokes equations and a multi-grid method. *Journal of Computational Physics* 1982; **48**:387–411.
31. Prabhakar V, Reddy JN. Spectral/hp penalty least-squares finite element formulation for the steady incompressible Navier–Stokes equations. *Journal of Computational Physics* 2006; **215**:274–297.
32. Barragy E, Carey GF. Stream function vorticity solution using high-p element-by-element techniques. *Communications in Numerical Methods in Engineering* 1993; **9**:387–395.
33. Erturk E, Corke TC, Cokcol C. Numerical solutions 2-D steady imcompressible driven-cavity flow at high Reynolds numbers. *International Journal for Numerical Methods in Fluids* 2005; **48**:747–774.
34. De Vahl Davis G. Natural convection of air in a square cavity: a benchmark numerical solution. *International Journal for Numerical Methods in Fluids* 1983; **3**:249–264.
35. De Vahl Davis G. Natural convection in a square cavity: a comparison exercise. *International Journal for Numerical Methods in Fluids* 1983; **3**:227–248.
36. Crochet MJ, Geyling FT, Van Schaftingen JJ. Finite element method for calculating the horizontal Bridgman growth of semiconductor crystals. In *Finite Elements and Fluids*, Gallagher RH *et al.* (eds), vol. 17(VI). Wiley: New York, 1985; 321–328.
37. Stogner RH, Kirk BS, Peterson JW, Carey GF. libMesh: a C++ library for parallel adaptive mesh refinement/coarsening simulations. *Engineering with Computers* 2006; **22**:237–254.
38. Griebel M, Dornseifer T, Neunhoeffer T. *Numerical Simulation in Fluid Dynamics—A Practical Introduction*. SIAM: Philadelphia, PA, 1998.
39. Elias RN, Coutinho ALGA. Stabilized edge-based finite element simulation of free-surface flows. *International Journal for Numerical Methods in Fluids* 2007; **54**(6):965–993.

40. Elias RN, Paraizo PLB, Coutinho ALGA. Stabilized edge-based finite element computation of gravity currents in lock-exchange configurations. *International Journal for Numerical Methods in Fluids* 2008; **57**(9):1137–1152.
41. Elias RN. Edge-based data structures for the parallel stabilized finite element simulation of incompressible flows. *Ph.D. Thesis*, COPPE/Federal University of Rio de Janeiro, 2007.
42. Elias RN, Coutinho ALGA, Martins MAD. Inexact Newton type methods for the solution of steady incompressible viscoplastic flows with the SUPG/PSPG finite element formulation. *Computer Methods in Applied Mechanics and Engineering* 2006; **195**:3145–3167.
43. Saad Y, Schultz MH. GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing* 1986; **7**:856–869.
44. Saad Y. *Iterative Methods for Sparse Linear Systems*. PWS Publishing: Boston, 1996.
45. Catabriga L, Valli AMP, Melotti BZ, Pessoa LM, Coutinho ALGA. Performance of LCD iterative method in the finite element and finite difference solution of convection-diffusion equations. *Communications in Numerical Methods in Engineering* 2005; **22**:643–656.