COMMUNICATIONS IN NUMERICAL METHODS IN ENGINEERING
Commun. Numer. Meth. Engng 2008; **24**:1941–1952
Published online 24 January 2008 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/cnm.1085

On decoupled time step/subcycling and iteration strategies for multiphysics problems

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

[1] *Department of Computer Science, Federal University of Espírito Santo, Vitoria, Brazil*
[2] *CFDLab, ICES, The University of Texas at Austin, Austin, TX, U.S.A.*
[3] *Center for Parallel Computations, COPPE, Federal University of Rio de Janeiro, Rio de Janeiro, Brazil*

SUMMARY

This work investigates partitioned iterative solution of coupled multiphysics systems including *subcycling time-stepping* strategies for decoupled subsystems in conjunction with a proportional-integral-derivative feedback control algorithm for adaptive time-step selection. Some basic algorithms are proposed and the total computational effort to integrate to steady state is compared for a representative coupled flow and heat transfer problem to illustrate the approach and assess performance efficiency. Copyright © 2008 John Wiley & Sons, Ltd.

Received 25 June 2006; Revised 4 October 2007; Accepted 23 October 2007

KEY WORDS: multiphysics decoupled iterative algorithms; subcycling and PID adaptive time steps; fluid–thermal

1. INTRODUCTION

There is a growing interest in analysis and efficient simulation of coupled multiphysics problems. This is enabled by continuing increases in computing power. As a result, both steady-state and transient calculations of coupled systems are increasingly sought. Algorithms may now exploit partitioning and iterative decoupling of the systems for various reasons. For instance,

*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: CAPES; contract/grant number: BEX 0509/06-0
Contract/grant sponsor: CNPq; contract/grant number: Proc. 620165/2006-5
Contract/grant sponsor: Sandia National Laboratory
Contract/grant sponsor: ICES/UT

this enables higher resolution simulations because a sequential computation may be less memory intensive than fully coupled analysis of a larger system. Furthermore, decoupled iterative solution may be less expensive than fully coupled solution, especially if the coupling between the subsystems is weak. In this weakly coupled situation, the nonlinear solution of the decoupled system and the use of related subsystem block preconditioning techniques can be more efficient and robust than the most standard fully coupled approaches. Finally, there is a considerable investment in software developed for complex single-physics applications in so-called 'legacy codes', and this provides a strong case for linking such software with more efficient decoupled algorithms rather than attempting to redesign and recode a new fully coupled multiphysics formulation.

In this work, we examine a class of algorithms for decoupled subsystem simulation and consider the following scenarios: (1) Decoupling of the steady-state (stationary) coupled subsystems; (2) decoupling of the time-dependent coupled systems within each time step; (3) subcycling physics subsystems based on transient behavior; (4) integration of the time-dependent problem to a steady state using subcycle schemes that are not necessarily time accurate; and (5) the extension of these ideas to parameter continuation techniques.

The scheme for decoupling steady-state subsystems is described first and the ideas here are then exploited for decoupling within a time step. Even when a solution to the stationary problem is the objective, transient schemes are still relevant. For example, the strategy of solving stationary problems by time stepping an associated transient problem to steady state can be an effective alternative to the more common approach of solving the stationary equations by iterative algorithms. This is particularly the case for nonlinear problems since convergence of such iterative algorithms is often an issue and may necessitate special continuation strategies. In fact, algorithms that employ time stepping to the steady state may be interpreted as a type of continuation scheme in this context. In these cases time-accurate solutions are not sought and this implies further that adaptive time stepping that leads to quite large time steps may be effective. For some related work involving explicit time integration schemes that exploit large time steps see, for instance, [1, 2].

For coupled multiphysics systems such as those encountered in fluid–thermal–structure interaction, the stability restrictions will differ for respective subsystems and one can accordingly use different respective time steps in a form of *subcycling* [3–5]. Even in the case where time-accurate solutions are needed, such adaptive time stepping and *subcycling* may be beneficial. Here we explore this idea further and in the context of combining *subcycling* with adaptive time stepping and proportional-integral-derivative (PID) feedback control algorithm strategies. Coupled fluid flow and heat transfer is selected as a representative dual-physics test case [6, 7] and some forms of the algorithms are implemented to assess the approach. The goal is to demonstrate the efficiency of the combined approach in reducing the total computational effort to obtain a steady-state numerical solution and to investigate some elementary variants of the algorithm. In this work, we will focus on implicit integrator schemes but remark that the decoupling strategies and especially the subcycling approach apply similarly for explicit integration schemes.

In the next section, we describe the decoupled partitioning approach and the basic strategy and algorithms for a generic multiphysics problem class. Then some variants of the algorithms are implemented, results verified and performance assessed for coupled fluid flow and heat transfer as a representative coupled dual-physics system. Results are compared for fixed and variable time-step schemes, and with subcycling. PID control feedback is used to determine step size.

DOI: 10.1002/cnm

2. MULTIPHYSICS DECOUPLED ITERATIVE ALGORITHMS

Let us consider typical coupling between multiphysics subsystems. This usually enters through the dependence of the source terms and coefficients in the respective subsystems on primary solution variables associated with other subsystems. As a specific example in the numerical tests described later we consider coupling via a Boussinesq heat source contribution in a viscous flow subsystem (identified as the *Physics 1* subsystem) together with coupling via the convective velocity in the heat transfer subsystem (identified as the *Physics 2* subsystem). The extension to include further subsystems with similar types of coupling such as additional reactive mass transport subsystems, thermal stress subsystems, etc. is evident. After discretization, this coupling persists in the coupled semidiscrete ordinary differential equation system for evolution problems as well as in the corresponding coupled steady-state problem. In the stationary case, one may seek to solve the coupled nonlinear algebraic systems by either a fully coupled or iteratively decoupled approach. Success here depends on the choice of the starting iterate, the strength of any nonlinearities and whether the coupling is weak or strong. Clearly, the decoupled strategy will be more appealing if the coupling is weak.

Similar reasoning applies in the transient problem. Here we will assume for the moment that an implicit integrator is implemented. Then in each time step, a coupled system must be solved and the same approaches as in the stationary problem apply. Synchronization occurs at the end of a time step. Clearly, adaptive time stepping with a common step for each subsystem is easy to implement in such a strategy. However, depending on the nonlinearities and computational costs of solving respective subsystems, one may elect to improve efficiency by subcycling with smaller time steps on one or more of the subsystems. For example, if some subsystems are more expensive to solve and have solutions that are relatively less sensitive to the other variables, then subcycling time steps on the subsystems with rapidly fluctuating solutions will generally improve the efficiency. Partial data transfers may be needed along the subcycling path and full synchronization with data transfer at the end of each set of subcycling steps.

In this work adaptive time-step selection is implemented via a PID control algorithm, but this is independent of the main ideas on decoupling algorithms discussed here. Adaptive time stepping can be invoked either with or without subcycling as indicated in the representative algorithms and numerical studies described later. Remark: Note that these decoupling/iteration/subcycling concepts all apply easily to explicit time integration in an analogous manner. The main distinction is that the time-step selection is limited by single-physics subsystem nonlinearity and stiffness. This may also dictate the subcycling synchronization. We will focus here on the implicit case, the explicit case being a straightforward modification.

To describe the main ideas and algorithms, it suffices that we consider a coupled dual-physics problem of the type indicated previously, where the source term of *Physics 1* subsystem depends weakly on the solution of *Physics 2* subsystem and a coefficient of *Physics 2* subsystem depends more strongly on the solution of *Physics 1* subsystem. This is precisely the type of coupled application considered in the coupled flow and heat transfer problem computed in subsequent tests. The extension to a larger number of subsystems is obvious. In the case where there is a rapidly fluctuating source term or similar feature in a subsystem, then subcycling may be appropriate in integrating this subsystem. A basic dual-physics scheme for the stationary problem is described first and then embedded in the transient cases. The familiar direct decoupled approach for the stationary case follows easily as shown below.

Basic Algorithm Cycle for dual-physics subsystems:

1. Select (or generate) starting iterates for solutions to both physics.
2. For each decoupled sweep, $s = 1, 2, \ldots$.
    - (a) Solve the nonlinear *Physics 1* subsystem numerically (lagging the *Physics 2* variable) to tolerance tol1.
    - (b) Solve the nonlinear *Physics 2* subsystem numerically (using the recently computed *Physics 1* variable) to tolerance tol2.
    - (c) Return to step (a) until convergence to a specified tolerance or iteration limit.

In this base algorithm there are two nonlinear subsystem solves per decoupled sweep. The scheme may be interpreted as a form of block Gauss–Seidel iteration on physics for the fully coupled system, since the most recently computed iterates are used in the subsystem solves. Block preconditioners can also be introduced for the subsystem solves to accelerate iterative convergence of these subsystems. Note that convergence of the coupled subsystem sweeps relies on the weak coupling assumption. Data transfers occur between respective subsystem solves and if different meshes are used for different physics, then this implies a further level of complexity for these projections. In the subsequent numerical work, the same mesh is used for both physics.

The scheme extends immediately to the transient problem with the same base algorithm applied within each time step. That is, we simply add a time stepping loop around the base algorithm. Two variants of the decoupled time stepping ‘pattern’ are described graphically in Figure 1 and incorporating *subcycling* with smaller time steps for *Physics 2* subsystem in Figure 2. The top diagram in Figure 1 depicts a basic decoupled solution scheme with constant fixed common time step. Here *Physics 1* subsystem is first solved lagging the dependence on the solution variable associated with *Physics 2* subsystem in the forcing term. Then *Physics 2* subsystem is solved with the recently computed *Physics 1* subsystem solution as coefficient input. This decoupled process-linking subsystem in a given time step can be characterized by a ‘Z’ pattern through a horizontal time step and across an embedded decoupled physics sweep. Iteration is usually repeated to convergence of both subsystem solution variables within the time step and then continued to successive time steps as shown in the figures by the repeated ‘Z’. Subsystem solves are represented by horizontal arrows and the decoupling by the diagonal arrow between physics subsystems. Remark: Note that each ‘Z’ in the pattern is usually repeated to convergence within the time step as described in the basic algorithm. However, in cases of time stepping to steady state where time-accurate solutions are not sought this is relaxed to yield a more efficient algorithm as seen later. The lower diagram in Figure 1 corresponds to a scheme that begins with a fixed time step and then adapts with a variable but again common time step across the coupled multiphysics. We refer to these schemes as algorithm 1.1 and algorithm 1.2, respectively. PID feedback controls the step size in algorithm 1.2 but any adaptive time step approach could be applied. Figure 2 illustrates the use of *subcycling* in *Physics 2*. The schemes again are initiated with a few constant fixed time steps and no *subcycling*. The first scheme, algorithm 2.1 in the upper diagram, shows two constant subcycle time steps applied to *Physics 2* subsystem integration for each fixed coarse time step in the *Physics 1* subsystem computation. Adaptive time stepping with the PID controller (for both subsystems) is applied in conjunction with similar *subcycling* for *Physics 2* subsystem in the lower diagram (algorithm 2.2).

The generalization of these schemes to more subsystems, to variable number of nonlinear iterations in each subsystem, and to variable number and size of subcycle steps is obvious. The

DOI: 10.1002/cnm

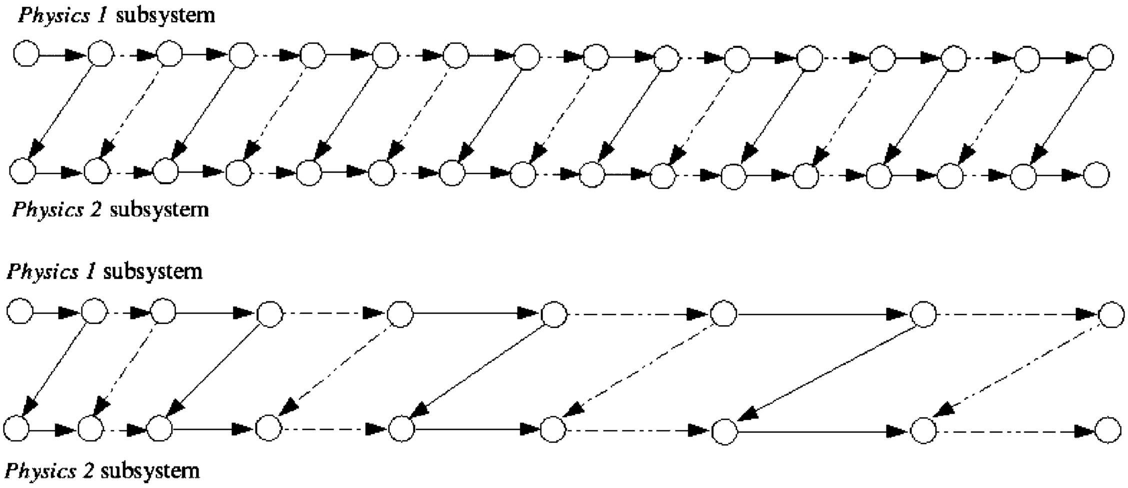

Figure 1. Solution sequence using fixed time-step sizes (algorithm 1.1, upper) and with adaptive PID time-step controller (algorithm 1.2, lower).

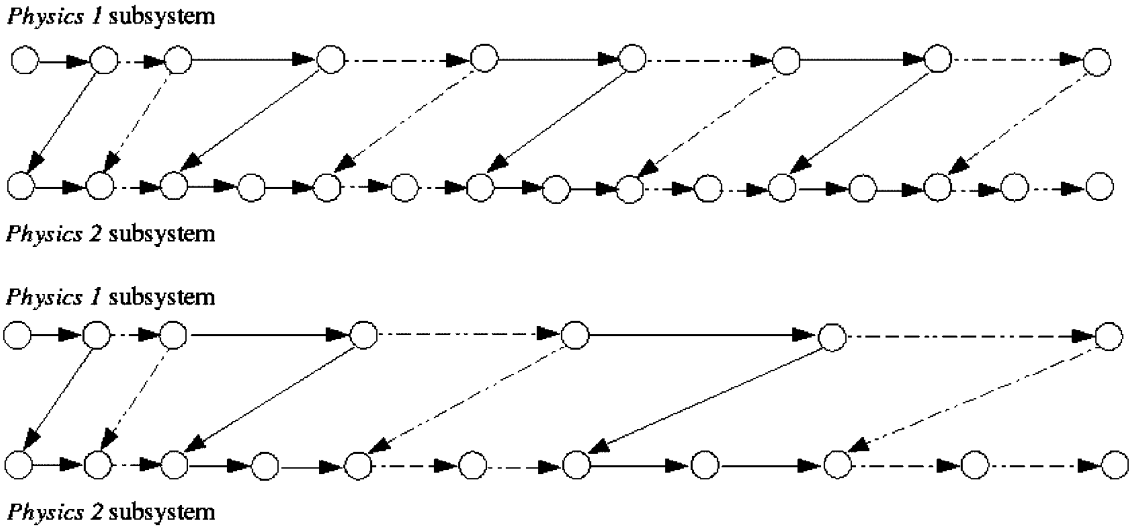

Figure 2. *Subcycling* in *Physics 2* subsystem (algorithm 2.1, upper) and with adaptive PID time stepping (algorithm 2.2, lower).

most effective strategy will depend on the strength of the coupling, the subsystem nonlinearities and subsystem stiffness. Hence, the best combination will be application dependent. For example, one may even elect to integrate one subsystem explicitly and another implicitly and use different meshes for each subsystem. Such ideas are common in practical large-scale applications employing sophisticated 'legacy' codes like those used in the DoE laboratories and for DoD applications such as fluid–structure interactions and analyzing the effects of blast waves on buildings. In the next

DOI: 10.1002/cnm

section, we consider coupled flow and heat transfer as a representative dual-physics application class to be implemented and used to assess the decoupled algorithm ideas proposed here.

3. FLUID–THERMAL SYSTEM AND ALGORITHM

The dimensionless equations describing the coupled fluid–thermal applications of interest have the form [8]

$$\frac{\partial \mathbf{u}}{\partial t}+\mathbf{u}\cdot\nabla\mathbf{u}-\nabla^2\mathbf{u}+\nabla p=\frac{Ra}{Pr}T\mathbf{g}+\mathbf{f}(t) \quad \text{in } \Omega\times I \tag{1}$$

$$\nabla\cdot\mathbf{u}=0 \quad \text{in } \Omega\times I \tag{2}$$

$$\frac{\partial T}{\partial t}+\mathbf{u}\cdot\nabla T-\frac{1}{Pr}\nabla^2 T=s(t) \quad \text{in } \Omega\times I \tag{3}$$

where Ω is the flow domain, $I=[0,\bar{t}]$ is the time interval, $\mathbf{u}$ is the velocity vector, p is the pressure, T is the temperature, $\mathbf{g}$ is the gravity vector, $\mathbf{f}(t)$ is a specified time-dependent body force, $s(t)$ is a time-dependent heat source that may be rapidly fluctuating, Ra is the Rayleigh number and Pr is the Prandtl number. Boundary conditions and initial conditions for temperature and velocity complete the mathematical statement of the problem. Note that we can decouple at the continuous operator level or at the algebraic level [6, 7].

The decoupled flow subproblem is obtained by iteratively 'lagging' temperature for the Boussinesq term in Equation (1). The resulting single-physics problem (*Physics 1* subsystem) is nonlinear due to the inertial term $\mathbf{u}\cdot\nabla\mathbf{u}$ and the subsystem for a time step can be solved by successive approximation, Newton iteration, or a similar strategy. Here, since we are only interested in the velocity solution and the associated coupled transport process, we use a penalty method to enforce the incompressibility constraint. The nonlinear systems are solved using successive approximations and integrated in time using the Crank–Nicolson mid-step method. The resulting approximate velocity field is then used to compute a convection coefficient for Equation (3). This linear equation is then discretized using a stream upwind Petrov–Galerkin (SUPG) scheme and integrated through the time step also using the Crank–Nicolson method. Remark: Note that (1)–(2) is a nonlinear vector subsystem and (3) is a linear scalar subsystem under the decoupling strategy. It follows that solving the decoupled viscous flow problem is computationally much more intensive than the heat transfer solve. Moreover, if the source term for the heat transfer equation is fluctuating rapidly, subcycling the second subsystem will be an efficient strategy since large steps can be taken on the flow solver with shorter subcycle steps to accommodate the source on the heat transfer calculation. This sequence of subsystem solves and data transfer is repeated to a desired convergence level.

A brief comment on PID control of time step size is warranted. To adapt the time step, a feedback control algorithm may be introduced based on controlling accuracy as determined by truncation error estimates. More specifically, Gustafsson [9, 10] and Gustafsson *et al.* [11] show that stepsize selection can be viewed as an automatic control problem with a PID controller defined as

$$\Delta t=\left(\frac{e_{n-1}}{e_n}\right)^{k_\mathrm{P}}\left(\frac{1}{e_n}\right)^{k_\mathrm{I}}\left(\frac{{e_{n-1}}^2}{e_n e_{n-2}}\right)^{k_\mathrm{D}}\Delta t_\mathrm{prev} \tag{4}$$

DOI: 10.1002/cnm

with

$$e_n = \max(e_u, e_T) \tag{5}$$

where

$$e_u = \frac{e_u^*}{\mathrm{tol}_u}, \quad e_u^* = \frac{\|\mathbf{U}^n - \mathbf{U}^{n-1}\|}{\|\mathbf{U}^n\|} \tag{6}$$

$$e_T = \frac{e_T^*}{\mathrm{tol}_T}, \quad e_T^* = \frac{\|\mathbf{T}^n - \mathbf{T}^{n-1}\|}{\|\mathbf{T}^n\|} \tag{7}$$

and Δt represents the new time-step size, Δt_{prev} is the time-step size at the previous step, e_n is the measure of the change of the quantities of interest in time t_n, k_{P}, k_{I}, and k_{D} are the PID parameters and tol_u and tol_T are user supplied tolerances corresponding to the normalized changes in velocities and temperature vectors, respectively. The efficiency of the control was demonstrated by Valli *et al.* [6, 7] 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 with solver operations, since time-step selection involves only storing a few extra vectors and computation of associated norms.

4. RESULTS

Several numerical experiments were conducted to confirm time accuracy of the PID variable time-step approach for representative model problems of increasing complexity and corresponding to:

1. scalar transport with specified convective velocity field;
2. incompressible Navier–Stokes;
3. coupled Navier–Stokes and transport.

The common strategy of constructing the source function to correspond to a pre-assigned solution function (more recently termed the 'method of manufactured solutions') was used to construct each problem. Tolerances for the time integration were selected together with an allowable range of step size. Results of numerical simulations with the PID strategy were compared for each of the above cases with those obtained using standard fixed time-step integration. The simulations with the present scheme confirm comparable accuracy at the final time and over the time period of integration with fewer time steps and therefore greater efficiency as anticipated. For instance, for the representative model problem 3 (coupled Navier–Stokes and transport), we obtain values for the velocity and temperature errors in the L_2-norm less than 10^{-8} and 10^{-4}, respectively, using fixed and adaptive time steps using PID selection. Using the PID controller, we can reduce the total computational cost by 13%. Time accuracy of the algorithm was maintained using an approach similar to that in [12] and based on a scheme in [13].

We remark that one can design problems that would be more demanding, but both integration strategies would fare similarly on these problems since the same integration schemes are used. Once again the PID control would be anticipated to be more accommodating through the variable time-step option. More sophisticated error indicators may be constructed and other time-step selection strategies based on these error indicators could be applied. Depending on the quality of these error indicators, time accuracy may be superior to the simple control strategy used here, but would no

DOI: 10.1002/cnm

doubt involve additional calculations. A better strategy may be to combine the present control approach with supplementary error indicator calculations that 'run in the background'. We are investigating the accuracy and efficiency of such hybrid strategies as part of our continuing work in this area. This will be part of a larger study that includes the details of the verification tests for the transport, Navier–Stokes, and coupled flow and transport problems mentioned above [14].

Following the previous time-accuracy tests on 'manufactured' problems with known analytic solutions, we next examine performance of the decoupled 'Z' algorithm on the familiar coupled problem of natural convection in a unit square $\Omega=[0,1]\times[0,1]$ with normalized temperatures $T=1, T=0$ on the left and right walls, respectively, adiabatic top and bottom walls (no free surface), with $Pr=0.71$ and Rayleigh numbers, Ra, of 10^3, 10^4 and 10^5. As a representative algorithm from the class of possible decoupled schemes described earlier, we implement the implicit mid-step transient formulation to step to steady state. This can be applied in various ways. For example, later we compare the steady-state results with various time-stepping combinations, including subcycling, and using one decoupled sweep per time step with successive approximation to convergence for the viscous flow problem (1)–(2). For these verification tests, the simulations repeating each decoupled cycle 'Z' in the pattern to convergence were also carried out for comparison purposes and gave the same steady-state solution with the same total number of successive approximations. Since time-accurate solutions are not sought in this fluid–thermal system example and the fluid system is more expensive to solve, the *subcycling* algorithm is implemented here in a manner similar to that described in Figure 2 previously.

Numerical results at steady state for Nusselt number at the left wall ($Nu_0=\int_0^1 q\,\mathrm{d}y$, where q is the heat flux) and for the stream function at the midpoint (ψ_{mid}) are compared with the results from [15–17]. The approximate velocities and temperatures are calculated with 9-node isoparametric quadrilateral elements on a uniform 16×16 mesh at $Ra=10^3$ and on a 32×32 mesh at $Ra=10^4, 10^5$. The initial time-step sizes in all cases is chosen small enough to allow convergence of the successive approximation solution iterations for viscous flow equations (1), (2) at the beginning of the process. For instance, we start with time-step size 0.01 at $Ra=10^3, 10^4$ and 0.001 at $Ra=10^5$. The kinetic energy of the system has been used to characterize global behavior such as rapid ramp-up in early time, oscillatory intermediate solutions with respect to time, and final steady-state behavior. In this work, we assume that the steady state occurs when the kinetic energy at two different time steps reaches a relative difference less than a specified tolerance, $\mathrm{tol}_{\mathrm{st}}$. By experiment, we infer that the steady state occurs when $\mathrm{tol}_{\mathrm{st}}=10^{-4}$ for all Reynolds numbers. For all Ra numbers, we set the ratio of fluid time step to *subcycling* thermal step as 1:3. The results using fixed time steps (algorithm 1.1), the PID controller (algorithm 1.2), *subcycling* without stepsize adaption (algorithm 2.1), and *subcycling* with the PID stepsize controller (algorithm 2.2) are shown in Table I. The agreement for all cases is favorable when compared with a benchmark

Table I. Comparison of specific results to benchmark case.

	Algorithm 1.1		Algorithm 1.2		Algorithm 2.1		Algorithm 2.2		Benchmark [17]	
Ra	Nu_0	ψ_{mid}	Nu_0	ψ_{mid}	Nu_0	ψ_{mid}	Nu_0	ψ_{mid}	Nu_0	ψ_{mid}
10^3	1.118	1.175	1.118	1.175	1.117	1.174	1.118	1.176	1.117	1.174
10^4	2.247	5.075	2.246	5.070	2.247	5.076	2.241	5.059	2.238	5.071
10^5	4.551	9.083	4.549	9.059	4.549	9.115	4.547	9.115	4.509	9.111

DOI: 10.1002/cnm

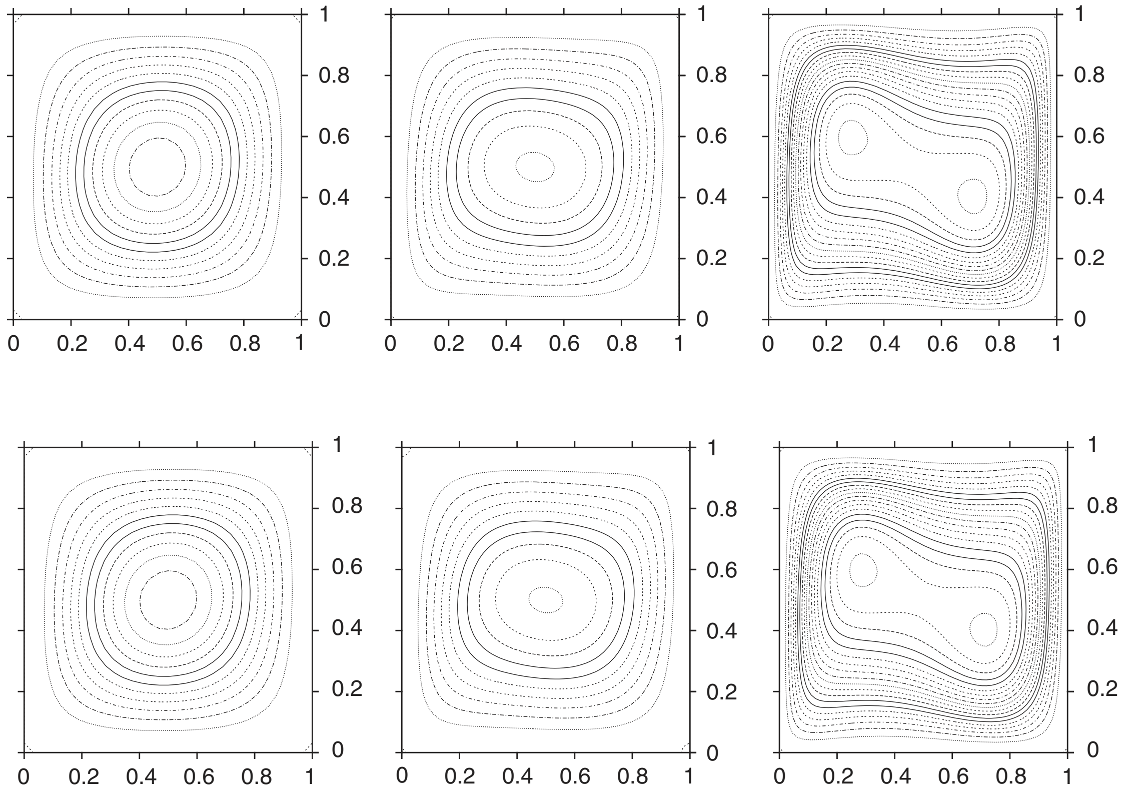

Figure 3. Stream functions contours for $Ra = 10^3$ (left), $Ra = 10^4$ (center) and $Ra = 10^5$ (right) using algorithm 1.1 (top) and algorithm 2.2 (bottom).

Table II. Computational effort for the natural convection problem.

	Algorithm 1.1		Algorithm 1.2		Algorithm 2.1		Algorithm 2.2	
Ra	nsa	c_{effort}	nsa	c_{effort}	nsa	c_{effort}	nsa	c_{effort}
10^3	58	1.0	32	0.55	25	0.43	21	0.36
10^4	91	1.0	80	0.88	41	0.45	41	0.45
10^5	416	1.0	388	0.93	143	0.34	118	0.28

result [17]. Corresponding quantities differ by less than 1.0%. Figure 3 shows the streamlines for $Ra = 10^3$, 10^4 and 10^5 obtained with fixed time steps (algorithm 1.1) and *subcycling* with the PID controller (algorithm 2.2). The streamlines obtained by the two algorithms agree very well and they also show good agreement with the results presented in [17].

Next, we compare the computational effort to calculate the solution using four variants of the algorithm strategies considered here. The computational effort, c_{effort}, is measured by the ratio of the total number of successive approximations, nsa, needed to calculate the velocity field to the number of successive approximations obtained using a fixed time-step size. Table II confirms that

Table III. Computational effort for the Rayleigh–Benard problem.

	Algorithm 1.1	Algorithm 1.2	Algorithm 2.1	Algorithm 2.2
nsa	1013	782	606	528
c_{effort}	1	0.77	0.60	0.52

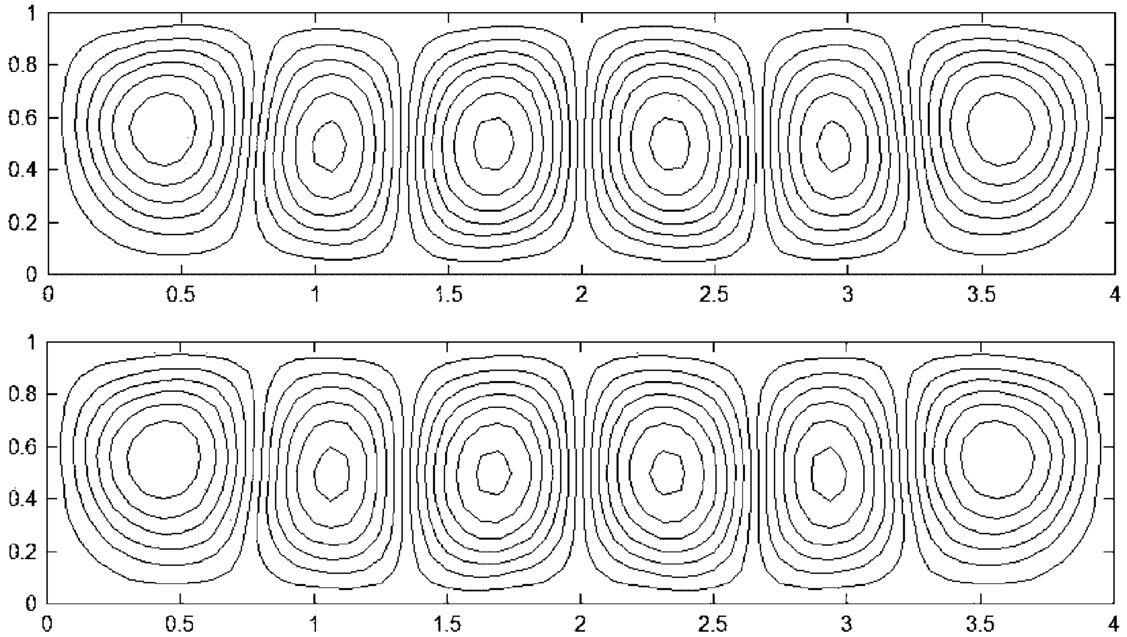

Figure 4. Streamlines for the flow in a container using fixed time-step sizes (algorithm 1.1, top) and the *subcycling* with PID controller (algorithm 2.2, bottom).

all approaches reduce the number of successive approximations needed to obtain the solution, but *subcycling* with the PID controller (algorithm 2.2) gives the best result in all cases. For example, with reference to CPU time we are able to calculate the solution 2.22 times faster than using algorithm 2.2 for $Ra = 10^4$.

The last test case is Rayleigh–Benard flow in a rectangular container of length 4 times the height with $Pr = 0.72$ and $Ra = 30\,000$. The normalized temperatures on the bottom surface and top surface are $T_h = 1$ and $T_c = 0$, respectively. The approximate velocity and temperature are calculated using a grid of 32×8 biquadratic elements. For algorithm 1.1, the steady state is assumed when the velocities and temperatures at two different time steps reach relative difference less than tolerance, $\text{tol}_{\text{st}} = 0.002$. For the other algorithms, the kinetic energy is used to characterize the final steady-state behavior. In the PID controller, a tolerance of 0.01 is set for changes in nodal velocities and temperatures. The initial time-step size is 0.001, and the ranges of minimum and maximum time steps are 0.001 and 0.5, respectively. This starting time step is the largest for which we obtained convergence in the successive approximation iterations for viscous flow from quiescent startup. The ratio of fluid to thermal time step is again taken to be 1:3.

DOI: 10.1002/cnm

As seen from Table III, using the PID controller with *subcycling* (algorithm 2.2) gives the best result. With a fixed time-step size of 0.001, the scheme requires 1013 iterations but only 528 iterations are needed when algorithm 2.2 is applied. In this case, the solution is obtained 1.9 times faster with respect to CPU time. Figure 4 compares the streamlines obtained with fixed time steps (algorithm 1.1) and *subcycling* with the PID controller (algorithm 2.2). There are six recirculation cells, and both solutions are in good agreement and are comparable to those in [18].

5. CONCLUSION

We investigate some concepts concerning decoupled algorithms for coupled multiphysics applications. The algorithms are described for coupled subsystems and diagrammed for a generic dual-physics pattern. The strategy is then applied to coupled fluid–thermal interaction to assess performance. The efficiency of *subcycling* and the PID time-step controller were investigated in this decoupled setting for the numerical simulation of natural convection and similar benchmark problems. The general approach covers the gamut of decoupled multiphysics strategies from one-way coupling where there is no feedback from *Physics 2* to two-way coupling for stationary and transient problems. Acceleration techniques including subcycling strategies, adjusting time stepping, and continuation approaches are included. Inevitably, these schemes must be considered in developing coupling strategies linking different physics in ‘legacy codes’. In this case, the meshes and discretization methods will usually also be different, so some care has to be taken here (e.g. see Carey *et al.* [19] for an example involving solution of a shallow-water subsystem on a finite element mesh with projection of the velocity field (in one-way coupling) to a different mesh using control volume approximation for transport simulation). We mention the relevance to grid computing, but conclude by noting that grid latency is the key factor here. Nevertheless, decoupled algorithms that involve (say) solving *Physics 1* on one computer system with data transfer to and from a *Physics 2* subsystem are practical, provided the communication is not excessive. For example, if only parameter extraction is returned then communication becomes negligible. In the viscous flow/heat transfer problem, the communication is reasonable for stationary problem or for implicit solves. The respective grid computer systems would influence the physics decompositions. Also note that in the parallel grid setting, the *Physics 1* and *Physics 2* simulations are not sequential, hence instead of the ‘Z’ pattern we imply both physics start in concert with lagged values and exchange data simultaneously at the end of the solve.

ACKNOWLEDGEMENTS

Dr Valli was supported by a CAPES grant (BEX 0509/06-0) from the Ministry of Education, Brazil, as a visiting scholar at the University of Texas at Austin. 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, Proc. 620165/2006-5, Sandia National Laboratory, and ICES/UT at Austin, U.S.

REFERENCES

1. Lorber AA, Carey GF, Joubert WD. ODE recursions and iterative solvers for linear equations. *SIAM Journal on Scientific Computing* 1996; **17**(1):65–77.
2. 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.

DOI: 10.1002/cnm

3. Belytschko T, Mullen R. Stability of explicit–implicit mesh partitions in time integration. *International Journal for Numerical Methods in Engineering* 1978; **12**:1575–1586.
4. Fellipa CA, Park KC, Farhat C. Partitioned analysis of coupled system. *Computer Methods in Applied Mechanics and Engineering* 2001; **190**:3247–3270.
5. Piperno S, Farhat C, Larrouturou B. Partitioned procedures for the transient solution of coupled aeroelastic problems. Part I: model problem, theory and two-dimensional application. *Computer Methods in Applied Mechanics and Engineering* 1995; **124**(34):79–112.
6. 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.
7. 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.
8. Davis MB, Carey GF. Parallel multilevel solution of Rayleigh–Benard–Marangoni problems. *International Journal of Numerical Methods for Heat and Fluid Flow* 2000; **10**(3):248–267.
9. Gustafsson K. Control theoretic techniques for stepsize selection in explicit Runge–Kutta methods. *ACM Transactions on Mathematical Software* 1991; **17**:533–554.
10. Gustafsson K. Control theoretic techniques for stepsize selection in implicit Runge–Kutta methods. *ACM Transactions on Mathematical Software* 1994; **20**:496–517.
11. Gustafsson K, Lundh M, Soderlind G. A PI stepsize control for the numerical solution for ordinary differential equations. *BIT* 1988; **28**:270–287.
12. Silva RS, Almeida RC. Space and time adaptivity for reaction–diffusion–convection problems. *European Congress on Computational Methods in Applied Science and Engineering*, Finland, July 2004.
13. Eriksson K, Johnson C. Error estimates and automatic time step control for nonlinear parabolic problem, I. *SIAM Journal on Numerical Analysis* 1987; **24**(1):12–23.
14. Valli AMP, Carey GF, Coutinho ALGA. *Technical Report*, The Institute for Computational Engineering and Sciences (ICES), Austin, TX, 2008; in preparation.
15. De Vahl Davis G. Laminar natural convection in a enclosed rectangular cavity. *International Journal of Heat and Mass Transfer* 1968; **11**:1675–1693.
16. De Vahl Davis G. Natural convection in a square cavity: a comparison exercise. *International Journal for Numerical Methods in Fluids* 1983; **3**:227–248.
17. 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.
18. Griebel M, Dornseifer T, Neunhoeffer T. *Numerical Simulation in Fluid Dynamics—A Practical Introduction*. SIAM: Philadelphia, PA, 1998.
19. Carey GF, Bicken G, Carey V, Berger C, Sanchez J. Locally constrained projections on grids. *International Journal for Numerical Methods in Engineering* 2001; **50**:549–577.

DOI: 10.1002/cnm