INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2002; **40**:1109–1119 (DOI: 10.1002/fld.397)

Fully coupled and operator-splitting approaches for natural convection flows in enclosures

S. Turek and R. Schmachtel

Institute for Applied Mathematics & Numerics, University of Dortmund, Vogelpothsweg 87, D-44227 Dortmund, Germany

SUMMARY

We present two different numerical approaches for the simulation of incompressible flow with heat transfer which is described via the Boussinesq model. Exemplarily, we describe a fully coupled approach, which allows both non-steady and steady simulations, while the decoupled approach is based on operator-splitting techniques as typical for fully non-steady incompressible problems. We discuss the numerical and computational characteristics and show results for the MIT2001 benchmark. All results are based on the Open Source CFD tool FEATFLOW (see www.featflow.de). Copyright © 2002 John Wiley & Sons, Ltd.

KEY WORDS: incompressible Navier–Stokes equations; Boussinesq; multigrid; FEM; CFD

1. INTRODUCTION

In Reference [1] we have shown that most of the existing solution schemes for the incompressible Navier–Stokes equations vary w.r.t. the different treatment of discrete non-linear saddle point problems of the following type:

$$S\mathbf{u} + kB\,p = \mathbf{g}, \quad B^{\mathrm{T}}\mathbf{u} = 0 \tag{1}$$

with matrices B and B^{T} being the discrete analogues of the operators ∇ and $-\nabla$, time step k and the (non-linear) velocity matrix S coming from the discretized momentum equations. Then, the various approaches can be mainly characterized through differences in the

- treatment of the non-linearity,
- treatment of the incompressibility,
- complete outer control.

*Correspondence to: S. Turek, Institute for Applied Mathematics and Numerics, University of Dortmund, Vogelpothsweg 87, D-44227 Dortmund, Germany.

Contract/grant sponsor: Publishing Arts Research Council; contract/grant number: 98-1846389

Received 21 January 2002
Revised 07 May 2002

The corresponding classification, particularly based on the aspects *treatment of the incompressibility* and *complete outer control* of the velocity–pressure coupling, leads to the *global multilevel pressure Schur complement* (MPSC) methods which contain SIMPLE-like or Uzawa methods as well as projection-like schemes (or fractional step, pressure correction, etc.) as proposed by Chorin, Van Kan or Turek and Gresho (see References [2, 1] for an overview). In contrast, there exist *local multilevel pressure Schur complement* methods which are related to block Jacobi- or Gauß–Seidel schemes for the velocity–pressure coupling and which include schemes as for instance the Vanka smoother [3] in a direct multigrid approach for stationary Navier–Stokes-like problems. In the following section, we will discuss the extension of these different approaches to the Boussinesq model which is the framework for the performed MIT benchmark calculations.

2. NUMERICAL CONCEPTS

The underlying Boussinesq model (see the given specifications of the model and the configurations for the MIT benchmark calculations in the summary articles [4, 5] by Christon *et al.*) can be written in the following form:

$$\partial_t \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla p + \nu_{\mathbf{u}} \Delta \mathbf{u} + \mathbf{j} T, \;\; \nabla \cdot \mathbf{u} = 0 \tag{2}$$

$$\partial_t T + \mathbf{u} \cdot \nabla T = \nu_T \Delta T \tag{3}$$

with $\nu_{\mathbf{u}} = \sqrt{Pr/Ra}$ and $\nu_T = \sqrt{1/Ra\,Pr}$. This set of PDEs is treated by a finite element approach (see References [1, 6]) which has been realized in the open source FEM software FEATFLOW (see http://www.featflow.de). The main characteristics of the numerical methods used are:

- Non-conforming Stokes finite elements (LBB-stable).
- Hybrid upwind/streamline-diffusion stabilization.
- Adaptive (fully/semi-)implicit time stepping (2nd order).
- Multilevel pressure Schur complement methods.
- Nonlinear quasi-Newton defect correction.
- Scalar multigrid solvers with adaptive grid transfer.

The performed mathematical steps for discretizing the Boussinesq equations and particularly for solving the resulting discrete non-linear systems in every time step can be described as follows:

Step 1: *Discretization in time and space.* We employ an implicit discretization in time (Implicit Euler, Crank–Nicolson, Fractional Step scheme [1]) and discretize in space using the above mentioned FEM techniques implemented in FEATFLOW. We denote the time step size (which may vary during the simulation) by k.

Then, the result is a fully coupled, discrete non-linear system of equations which can be described as follows:

$$S_{\mathbf{u}}(\mathbf{u}^{n+1})\mathbf{u}^{n+1} + kB\,p^{n+1} + kM_T T^{n+1} = \mathbf{f}(n+1,n), \;\; B^{\mathrm{T}}\mathbf{u}^{n+1} = 0 \tag{4}$$

$$S_T(\mathbf{u}^{n+1})T^{n+1} = g(n+1,n) \tag{5}$$

with the following specific matrices for the momentum equations w.r.t. velocity ($\mathbf{u}$) and temperature (T):

$$S_{\mathbf{u}}(\mathbf{v}) := \alpha M_{\mathbf{u}} + v_{\mathbf{u}} k L_{\mathbf{u}} + k K_{\mathbf{u}}(\mathbf{v}) \tag{6}$$

$$S_T(\mathbf{v}) := \alpha M_T + v_T k L_T + k K_T(\mathbf{v}) \tag{7}$$

M, L and $K(.)$ describe the reactive, diffusive and convective terms. The index $n+1$ denotes the $n+1$-th time step, that means the time stage t_{n+1}. Then, the resulting non-linear discrete saddle point problem can be rewritten in matrix–vector notation as

$$\begin{bmatrix} S_{\mathbf{u}}(\mathbf{u}^{n+1}) & kM_T & kB \\ 0 & S_T(\mathbf{u}^{n+1}) & 0 \\ B^{\mathrm{T}} & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}^{n+1} \\ T^{n+1} \\ p^{n+1} \end{bmatrix} = \begin{bmatrix} \mathbf{f}(n+1,n) \\ g(n+1,n) \\ 0 \end{bmatrix} \tag{8}$$

and will be treated in two different ways:

Step 2a: *Local MPSC* (*fully coupled*, *for* '*steady*' *configurations*).

Perform an outer Newton- or fixpoint-like nonlinear iteration scheme with a multigrid solver for the resulting linear coupled subproblems. The local MPSC approach solves 'exactly' on 'subsets/patches' and performs an outer Block–Gauß-Seidel/Vanka iteration as smoother (see Reference [1] for the details).

Consequently, the $(l+1)$th Newton-like step reads in corresponding algebraic notation

$$\begin{bmatrix} \mathbf{u}^{l+1} \\ T^{l+1} \\ p^{l+1} \end{bmatrix} = \begin{bmatrix} \mathbf{u}^{l} \\ T^{l} \\ p^{l} \end{bmatrix} - \omega^{l+1} \begin{bmatrix} N_{\mathbf{u}}(\mathbf{u}^l) & kM_T & kB \\ \gamma N_{\mathbf{u}}(T^l) & S_T(\mathbf{u}^l) & 0 \\ B^{\mathrm{T}} & 0 & 0 \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{def}_{\mathbf{u}}^l \\ \mathrm{def}_T^l \\ \mathrm{def}_p^l \end{bmatrix} \tag{9}$$

$$\begin{bmatrix} \mathbf{def}_{\mathbf{u}}^l \\ \mathrm{def}_T^l \\ \mathrm{def}_p^l \end{bmatrix} := \begin{bmatrix} S_{\mathbf{u}}(\mathbf{u}^l) & kM_T & kB \\ 0 & S_T(\mathbf{u}^l) & 0 \\ B^{\mathrm{T}} & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}^{l} \\ T^{l} \\ p^{l} \end{bmatrix} - \begin{bmatrix} \mathbf{f}(n+1,n) \\ g(n+1,n) \\ 0 \end{bmatrix} \tag{10}$$

where in each step a coupled linear problem with the following matrix has to be solved:

$$\begin{bmatrix} N_{\mathbf{u}}(\mathbf{u}^l) & kM_T & kB \\ \gamma N_{\mathbf{u}}(T^l) & S_T(\mathbf{u}^l) & 0 \\ B^{\mathrm{T}} & 0 & 0 \end{bmatrix} \tag{11}$$

The parameter $\gamma = 1$ yields the full Newton scheme which also includes the 'bad' reactive terms due to the nonlinear convection in the matrix $N_{\mathbf{u}}(\mathbf{u}^l)$, while $\gamma = 0$ corresponds to the fixed point approach (with $N_{\mathbf{u}}(\mathbf{u}^l) = S_{\mathbf{u}}(\mathbf{u}^l)$). Numerical details can be found in Reference [1].

To solve these Oseen-like problems, a fully coupled multigrid solver is used with a special smoother of local MPSC/'Vanka' type (see Reference [1]) which involves the following local

Table I. Total number of non-linear steps (left)/averaged number of multigrid iterations per non-linear sweep (right) for the fixed point approach vs the Newton scheme for a sequence of meshes.

	Non-linear solver ($\gamma = 0/1$)			
#NEL/#NEQ	Fixed point ($\gamma = 0$)		Newton ($\gamma = 1$)	
1.408/ $\approx$ 10.000	55/	2.1	11/	2.5
5.632/ $\approx$ 40.000	70/	1.9	8/	2.4
22.528/ $\approx$ 160.000	54/	1.8	5/	2.0

preconditioners on so-called 'patches' Ω_i:

$$\begin{bmatrix} N_{\mathbf{u}|\Omega_i}(\mathbf{u}^l) & kM_{T|\Omega_i} & kB_{|\Omega_i} \\ \gamma N_{\mathbf{u}|\Omega_i}(T^l) & S_{T|\Omega_i}(\mathbf{u}^l) & 0 \\ B^{\mathrm{T}}_{|\Omega_i} & 0 & 0 \end{bmatrix}^{-1} \tag{12}$$

The patch Ω_i may be a collection of one or several neighboured elements/cells which are clustered according to a prescribed pattern of blocked subdomains or, in an adaptive way, so as to get rid of (mesh) anisotropies.

The advantages of this (seemingly quite complicated) approach are the following: direct steady solvers become possible without any pseudo time-stepping to reach the steady-state solution, while for nonsteady configurations very large time steps can be used—without significant loss of robustness—due to the fully implicit treatment (see Reference [1]). On the other hand, the convergence behaviour of the multigrid solvers due to the full Newton linearization may lead to numerical problems and is still topic of ongoing research.

In Table I results are given for Rayleigh number $Ra = 3.0 \times 10^5$ and Prandtl number $Pr = 0.71$ which leads to a steady solution. We performed 16 smoothing steps with a maximum patchsize of 64 elements and applied streamline-diffusion stabilization for the convective terms. #NEL denotes the number of elements/cells while #NEQ is the total number of unknowns (including velocity, pressure and temperature).

Due to the excellent convergence behaviour of the Newton-like scheme, this complex non-linear problem can be solved in 5 non-linear steps leading to a total number of 10 linear multigrid steps only. As a conclusion, this approach constitutes a very efficient direct non-linear solver for stationary configurations (see Reference [1] for more examples).

Step 2b: *Global MPSC* (*operator-splitting*, *for* '*non-steady*' *configurations*)

> Perform an outer decoupling of the Navier–Stokes part (w.r.t. **u** and p) and the energy equation (for T). Then, apply a Newton-like non-linear solver for the momentum equation and employ multigrid schemes for all linear scalar subproblems (see Reference [1] for the details).

Then, the formulation as so-called 'discrete projection' scheme reads for time step t^{n+1}:

$$S_{\mathbf{u}}(\tilde{\mathbf{u}}^{n+1})\tilde{\mathbf{u}}^{n+1} = \mathbf{f}(n+1, n) - kB\,p^n + kM_T T^n \text{ (Burgers)} \tag{13}$$

$$\text{`}\Delta_h\text{'}q = f_p,\ \ f_p := \frac{1}{k} B^{\mathrm{T}} \tilde{\mathbf{u}}^{n+1} \ \text{(Pressure Poisson)} \tag{14}$$

$$p^{n+1} = p^n + q + \nu_{\mathbf{u}} k M_p^{-1} f_p,\ \ \mathbf{u}^{n+1} = \tilde{\mathbf{u}}^{n+1} - k M_{\mathbf{u}}^{-1} B q \ \text{(Update)} \tag{15}$$

$$S_T(\mathbf{u}^{n+1}) T^{n+1} = \mathbf{g}(n+1, n) \ \text{(Temperature)} \tag{16}$$

In contrast to the sequence of 'small and coupled' subproblems in the local MPSC approach, here we end up with a sequence of 'large but scalar' problems. However, due to the abundance of standard software packages, fast Numerical Linear Algebra tools are available for these classes of problems. Moreover, this approach—combined with our special choice of FEM spaces in FEATFLOW—is almost 'optimal' for fully non-steady problems, since in the Pressure-Poisson step a special preconditioner $\text{`}\Delta_h\text{'} = P = B^{\mathrm{T}} M_{\mathbf{u}}^{-1} B$ can be explicitly constructed which approaches an exact inverse for small time steps (in fully nonstationary configurations) and which leads to very favourable numerical properties w.r.t. optimized multigrid solvers.

Beside a short presentation of the results for the non-steady MIT benchmark configuration, which have been performed via this approach, we will concentrate on 'interesting results and experiences' which occured during our simulations and which should be reported in this paper.

3. SOME NUMERICAL DETAILS OF THE BENCHMARK CALCULATIONS

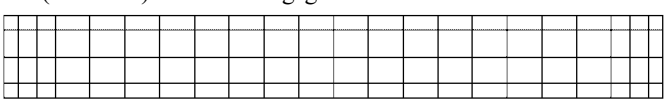
All tests have been performed for the following coarse mesh which has been successively refined by connecting opposite midpoints. The resulting geometrical details can be found in Table II.

Since the applied methodology is a semi-implicit approach, the question arises of how to select appropriate time steps: there is no CFL condition such that the time steps can be chosen by accuracy reasons, and not by robustness criteria!

In FEATFLOW (see Reference [1]) an implicit 'error indicator' based time step control is employed which performs (non-linear) calculations for time steps $\Delta T/3$ and ΔT to approximate the (local) truncation error for a user-defined functional $f(\mathbf{u}, p)$ in time:

$$|f(\mathbf{u}_h^k(t_n), p_h^k(t_n)) - f(\mathbf{u}_h(t_n), p_h(t_n))| \leqslant \mathrm{TOL} \tag{17}$$

Table II. Coarse mesh ('level 1') and resulting geometrical details for different levels of refinement.



Level	Vertices	Elements	Midpoints	Total unknowns
1	115	88	202	**694**
GRID0	22 945	22 528	45 472	**159 361**
GRID1	90 945	90 112	181 056	**634 113**
GRID2	362 113	360 448	722 560	**2 529 793**

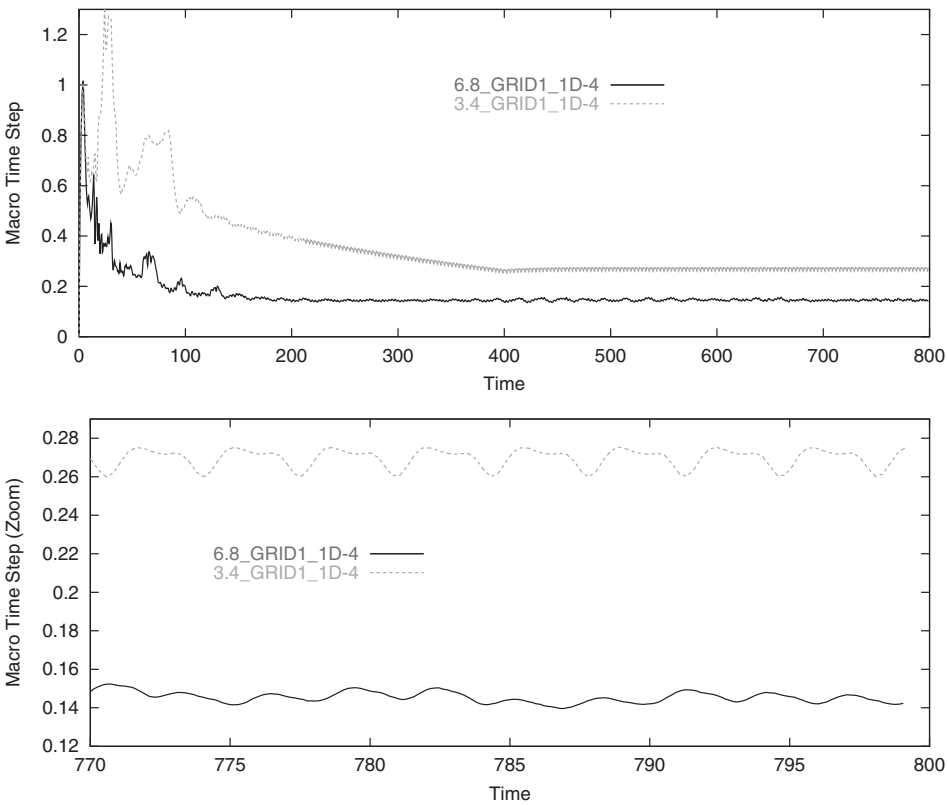


Figure 1. Development of the adaptive (macro) time steps.

Figure 1 shows the resulting behaviour of the adaptive 'macro-time steps' ($=3\cdot\Delta T$) for two cases: $Ra=3.4\times10^5$ (the benchmark configuration) and $Ra=6.8\times10^5$. The value $1D-4$ represents the exemplarily prescribed tolerance criterion $\mathrm{TOL}=10^{-4}$.

Our tests regarding the influence of the size of the tolerance criterion TOL demonstrate a typical behaviour of operator-splitting approaches—as prototypical representatives for schemes a la Chorin, Van Kan and other projection-like methods, see References [1, 2]. The results of the following tables and figures can be summarized as:

- *larger* amplitudes occur for *larger* time steps as TOL increases
- *mean values* and frequencies are *much less* influenced
- *nice graphical* results can be obtained via 'clever' combinations of coarse meshes and large time steps

What does this mean in practical calculations ?

Applying large time steps may allow a qualitatively good prediction of global flow quantities such as frequency and mean values, while the amplitudes may overshoot. This means that—based on such simulations—more accurate results for point-oriented quantities can only be obtained by taking smaller time steps while mean values can be approximated sufficiently well with such large time steps. However, these results also show that the combination of

Table III. Performance results for a sequence of refined spatial meshes and varying tolerance criterions for the symmetric periodical case $Ra = 3.4 \times 10^5$.

Time control TOL	GRID0	GRID1	GRID2
	Averaged macro-time-step size (adaptively chosen)		
10^{-3}	0.2525	0.2164	0.2064
10^{-4}	0.1572	0.0898	0.0803
10^{-5}	0.1518	0.0484	0.0297
	(Approximative) oscillation period (evaluated graphically)		
10^{-3}	3.977	3.475	3.582
10^{-4}	3.566	3.447	3.416
10^{-5}	3.564	3.438	3.422
	Averaged CPU per macro-time-step (in seconds) (≈ 3 small $(\Delta T) + 1$ large $(3\Delta T)$ substeps) (COMPAQ ES40/667 MHz, sequential, Standard FeatFlow)		
10^{-3}	6	31	159
10^{-4}	6	27	129
10^{-5}	5	24	108

'large' time steps and 'coarse' meshes may be used—if the exact solution is known—to 'produce' quantitatively nice benchmark results w.r.t. the amplitudes or peak values, while at the same time a qualitatively wrong behaviour of global characteristics such as frequency and similar quantities is observed (Table III):

> Time-dependent benchmark calculations are often useless unless one performs a rigorous error control in time—which no one seems to be able to do at the moment—or one shows the results for a sequence of successively refined temporal mesh configurations! Often, the results may get worse for increasingly fine temporal (and spatial) meshes.

Figure 2 demonstrates that the coarsest mesh GRID0 together with the weakest time stepping criterion $\mathrm{TOL} = 10^{-3}$ may give 'surprisingly' good results for the amplitudes, respectively, the differences of the peak values for the Nusselt number, which are even comparable with 'reference' values on GRID2 and $\mathrm{TOL} = 10^{-4}$. However, it is also obvious from this example that the frequency is completely wrong, and that a smaller time step (due to $\mathrm{TOL} = 10^{-4}$) will lead to much smaller oscillations. So, refining the temporal mesh leads to 'worse' results while calculations with prescribed large time step sizes—chosen in an appropriate way using some *a priori* knowledge about the expected oscillations—might produce 'reasonable' results, particularly according to a graphical evaluation (Table IV).

Finally, we show corresponding results for the case $Ra = 6.8 \times 10^5$ which leads to a non-symmetric and non-periodical behaviour as can be seen in Figure 3 (see also Figure 1 above showing the non-periodical behaviour of the adaptively chosen time steps). The symmetry (look at EPS12) is significantly broken, the Nusselt number oscillates in a non-periodical manner, and it is much harder—in contrast to the simpler case $Ra = 3.4 \times 10^5$—to obtain mesh independent results.

Nevertheless, the discussed numerical techniques (implicit time stepping with adaptive time step control, multigrid, FEM) work quite fine in this case, too, and the numerical cost seems

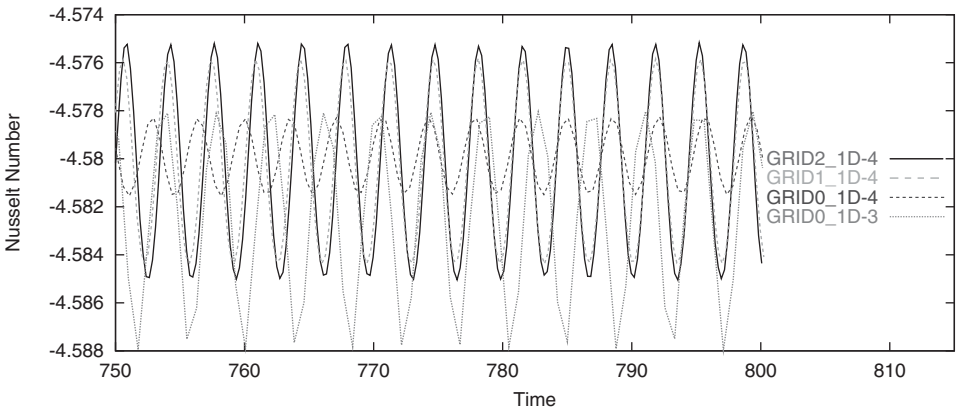


Figure 2. Nusselt number for a sequence of refined spatial meshes (GRD0-GRID2) and varying tolerance criterions (TOL $= 10^{-3}/10^{-4}$) for the symmetric periodical case $Ra = 3.4 \times 10^5$.

Table IV. Benchmark results for a sequence of refined spatial meshes and varying tolerance criterions for the symmetric periodical case $Ra = 3.4 \times 10^5$.

TOL (adaptiv)	GRID0		GRID1		GRID2	
	Mean	Ampl	Mean	Ampl	Mean	Ampl
Nusselt number						
10^{-3}	4.5824	0.0103	4.5773	0.0108	4.5766	0.0118
10^{-4}	4.5799	0.0036	4.5802	0.0085	4.5803	0.0099
10^{-5}	4.5797	0.0003	4.5795	0.0057	4.5791	0.0070
Temperature						
10^{-3}	0.2563	0.0833	0.2595	0.0826	0.2604	0.0861
10^{-4}	0.2589	0.0203	0.2637	0.0518	0.2651	0.0612
10^{-5}	0.2590	0.0022	0.2638	0.0336	0.2647	0.0442
U1 component						
10^{-3}	0.0668	0.0895	0.0628	0.1473	0.0631	0.1004
10^{-4}	0.0549	0.0250	0.0585	0.0655	0.0606	0.0789
10^{-5}	0.0542	0.0027	0.0552	0.0423	0.0572	0.0565
Pressure difference 14						
10^{-3}	0.0010	0.0238	0.0021	0.0276	0.0023	0.0287
10^{-4}	0.0013	0.0092	0.0016	0.0237	0.0016	0.0272
10^{-5}	0.0013	0.0019	0.0017	0.0162	0.0020	0.0209

to be scalable as can be seen in the following Table V. Moreover, even if the pointwise control (in space and time) of the flow quantities seems to be very hard, the prediction of mean values in space (compare the temperature plots with the Nusselt number which looks more 'smooth') and particularly in time for time-averaged Nusselt numbers—or for the time-

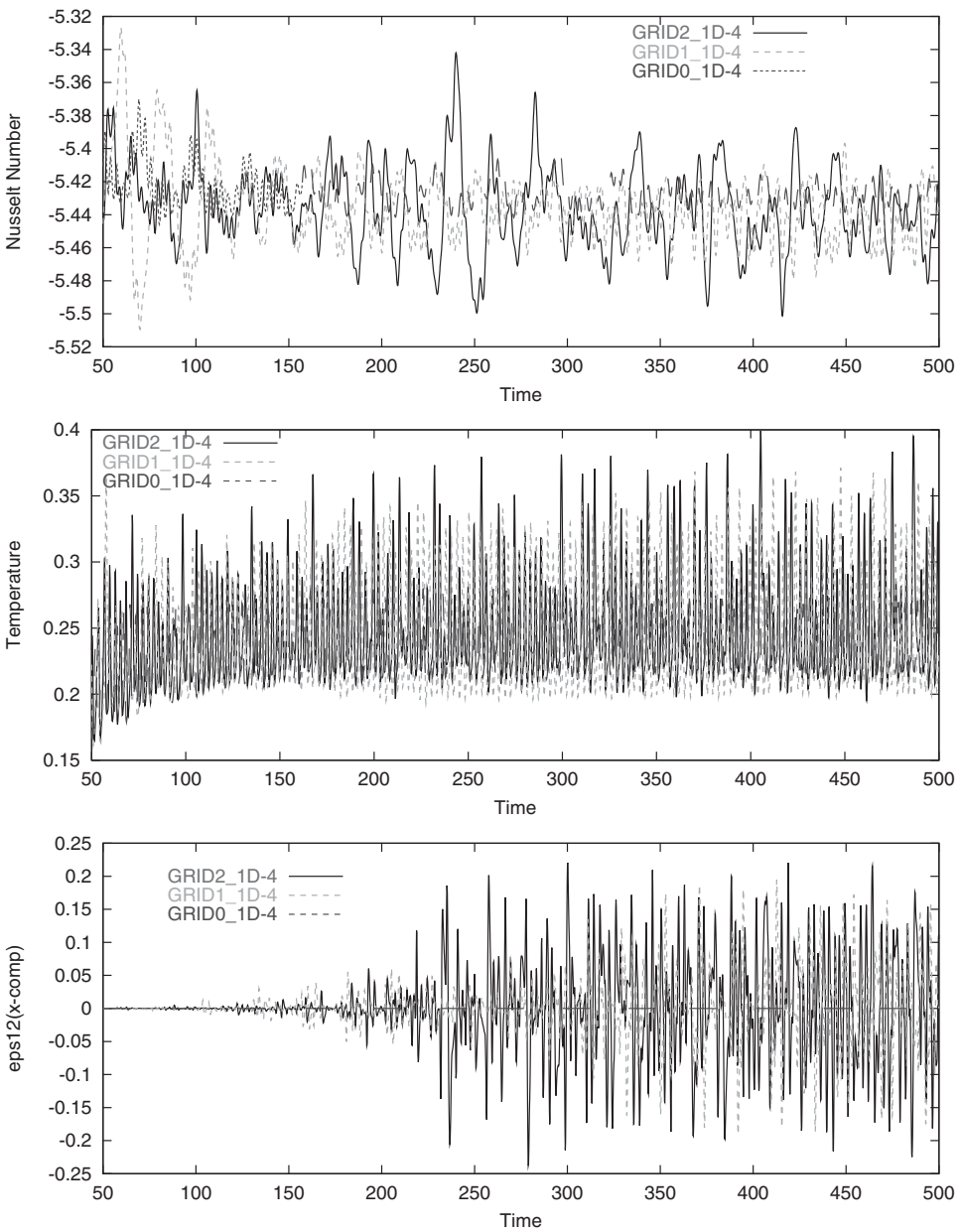


Figure 3. Results for a sequence of successively refined spatial meshes (GRID0-GRID2) for the non-symmetric non-periodical case $Ra = 6.8 \times 10^5$.

averaged drag and lift coefficient in automotive and aircraft design—might become feasible with such powerful mathematical techniques in the future.

Table V. Corresponding results for the non-symmetric non-periodical case $Ra = 6.8 \times 10^5$.

Time control TOL	GRID0	GRID1	GRID2
	Averaged time step (adaptively chosen, T = [0:500])		
10^{-4}	0.0911	0.0546	0.0534
	Averaged CPU per macro-time-step (in seconds) (≈ 3 small $(\Delta T) + 1$ large $(3\Delta T)$ substeps) (COMPAQ ES40/667 MHz, sequential, Standard FeatFlow)		
10^{-4}	5	27	108

4. CONCLUSIONS

The FEATFLOW software is an open source FEM package which is conceptualized and realized as general purpose CFD tool for incompressible flow in general domains. It is obvious that such a general approach is not 'optimal' for the proposed MIT benchmark since the underlying rectangular domain is very simple so that finite difference methods and particularly spectral methods seem to be favourable.

Nevertheless, the collection of benchmark results from the different groups shows that even such a very general tool, which however is based on very modern and sophisticated FEM techniques, adaptive implicit time stepping approaches and particularly rigorous multigrid realization, is at least comparable, if not superior. Moreover, one has to keep in mind that the implicit time stepping is controlled by accuracy reasons only, and not by CFL-like conditions such as the minimum mesh size or the magnitude of dimensionless numbers as *Re*, *Ra* and *Pr*. It is clear that higher values for the Rayleigh number will lead to smaller time steps, but the precise values are implicitly determined due to the more complex dynamic behaviour. Therefore, rigorous accuracy-oriented control mechanisms may get feasible in future (see Reference [1] for some information on this subject).

We finish with the following final remarks for the performed MIT benchmark calculations:

- For steady configurations ('small' *Ra* numbers), direct steady solution techniques are possible which are scalable w.r.t. the mesh size, because of the use of multigrid routines, and also w.r.t. the *Ra* number due to the (almost) quadratic convergence behaviour of the nonlinear Newton-like solvers. No pseudo time stepping is necessary!
- Scalability (w.r.t. mesh size, time steps, *Ra* number) of the solver for non-steady problems (higher *Ra* numbers) is achievable due to the implicit components and the optimized multigrid solvers.
- The adaptive and implicit time step control works well, even in the case of non-symmetric and non-periodical configurations.
- The choice of 'wrong' time steps may lead to very different results. However, there is no way to predict—by *a priori* arguments—the 'right' time step sizes (unless an explicit treatment does so via a CFL condition) so that methodologies for the selection and the resulting quality of the employed time step sizes should be developed in more detail, particularly for implicit approaches, in the future.

- Working with adaptive time stepping and implicit schemes, the resulting evaluation of certain benchmark flow quantities (oscillation period, peak values) is mostly based on graphical postprocessing such that due to the inequidistantly distributed data and the larger time steps, it might get hard to guarantee a 'postprocessing error' of less than 0.1–1% in some situations.

REFERENCES

1. Turek S. *Efficient Solvers for Incompressible Flow Problems: An Algorithmic and Computational Approach.* Lecture Notes in Computer Science, vol. 6, Springer: Berlin, 1999.
2. Gresho PM, Sani RL. *Incompressible Flow and the Finite Element Method.* Wiley: Chichester, 1998.
3. Vanka SP. Implicit multigrid solutions of Navier–Stokes equations in primitive variables. *Journal of Computational Physics* 1985; **65**:138–158.
4. Christon MA, Gresho PM, Sutton SB. Computational predictability of natural convection flows in enclosures. In *Computational Fluid and Solid Mechanics*, First MIT Conference on Computational Fluid and Solid Mechanics, Bathe KJ (ed.). Elsevier: Amsterdam, 2001, 1465–1468.
5. Christon MA, Gresho PM, Sutton SB. Computational predictability of natural convection flows in enclosures. *International Journal for Numerical Methods in Fluids* 2002; **40**:953–980.
6. Turek S. FEATFLOW. *Finite Element Software for the Incompressible Navier–Stokes Equations*: *User Manual.* University of Dortmund, Release 1.2, 1999 (see www.featflow.de).

