INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS, VOL. 25, 959–971 (1997)

ERROR ANALYSIS AND ITERATIVE METHODS IN PSEUDOSPECTRAL SEMI-IMPLICIT SIMULATIONS: AN APPLICATION TO COMPRESSIBLE CONVECTION

F. RUBINI,[1] V. TAVERNE[2] AND L. VALDETTARO[3]*

[1] *Dip. di Astronomia e Scienze dello Spazio, Università di Firenze, Largo E. Fermi 5, I-50125 Firenze, Italy*
[2] *Centre Européen de Recherche et de Formation Avancée en Calcul Scientifique, 42 Avenue G. Coriolis, F-31057 Toulouse Cedex, France*
[3] *Dip. Di Matematica, Politecnico di Milano, Piazza L. da Vinci 32, I-20133 Milano, Italy*

SUMMARY

Compressible convection is an interesting field for numerical experiments. Rapidly varying small-scale flow structures appear as the Rayleigh number *Ra* increases, demanding larger spatial resolution under more and more severe Courant stability conditions. Coupling a pseudospectral approximation in space to a semi-implicit scheme in time allows one to increase the size of Δt, though at each time step a system of algebraic equations, whose size increases with the spatial resolution, must be solved by means of direct or iterative methods. The former allows one to minimize the consumption of CPU time but leads to unacceptable demand of memory. The efficiency and cost of the latter, on the other hand, depend heavily on the choice of the preconditioning operator and on the allowed error tolerance. In this paper we check the capabilities of iterative-like methods and we achieve the main goal of drastically reducing the memory storage with respect to direct methods, without increasing the CPU time. © 1997 John Wiley & Sons, Ltd.

Int. J. Numer. Meth. Fluids, **25**: 959–971 (1997)

No. of Figures: 12. No. of Tables: 1. No. of References: 9.

KEY WORDS: error analysis; iterative methods; semi-implicit simulations

1. INTRODUCTION

Numerical simulation of compressible convection is a challenge of great physical and numerical interest. The equilibrium field stability in a viscous fluid depends essentially on the value of the Rayleigh number *Ra*. For *Ra* smaller than a critical value Ra_c the viscous forces are able to react to the buoyancy forces. In contrast, $Ra > Ra_c$, small perturbations will grow exponentially until non-linear saturation occurs.[1]

Numerical simulations of realistic turbulent convective flows must be able to span a wide range of both length and time scales in order to take into account the great variety of physical phenomena occurring in the fluid.[2] When *Ra* is not too large, a steady state pattern with large-scale convective cells is rapidly achieved. As *Ra* increases, small structures appear, leading to the need of high spatial resolution and more severe numerical temporal stability constraint. A comparison among all the time scales, due to diffusion, Dt_{diff}, sound wave propagation, Dt_{sw}, and convective motions, Dt_{conv}, gives

* Correspondence to: L. Valdettaro, Dip. di Matematica, Politecnico di Milano, Piazza L. da Vinci 32, I-20133 Milano, Italy.

CCC 0271–2091/97/080959–13 $17.50

Received 16 November 1995
Revised 19 December 1996

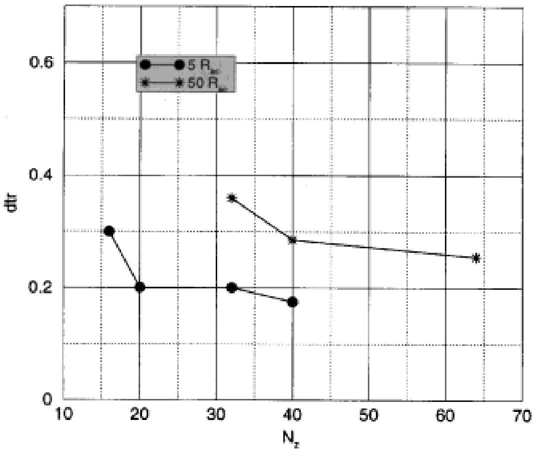

Figure 1. Ratio of maximum time step to *CFL* as a function of vertical resolution N_z

an upper bound to the numerical time step Dt_n of an explicit scheme:

$$Dt_\mathrm{n} < \min(Dt_\mathrm{sw}, Dt_\mathrm{diff}, Dt_\mathrm{conv}). \tag{1}$$

Larger time steps are allowed when a semi-implicit scheme is used. In the case where the diffusion and wave propagation terms are computed implicitly, the stability condition becomes

$$Dt_\mathrm{n} < Dt_\mathrm{conv} \tag{2}$$

and a gain in computational time is obtained, as shown in Figures 1 and 2. Such a gain implies, unfortunately, that a large system of equations must be solved at each time step and this operation, the most time-consuming one in the code, has to be handled with care.

The direct LU method is able to produce a very accurate solution but is expensive in terms of the number of operations. To reduce them, since in our application the matrices arising from the semi-implicit scheme do not depend on time (see Section 4.3), the factorization step can be performed only once at the beginning of the run and all the factorized matrices are stored. This approach minimizes the consumption of CPU time but requires a lot of memory storage.

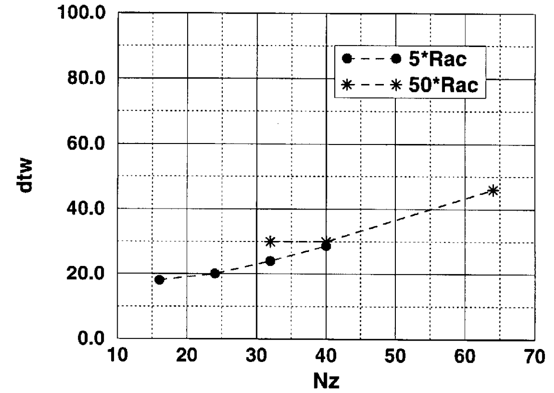

Figure 2. Ratio of maximum time step to fastest sound wave period as a function of vertical resolution N_z

Iterative-like methods are an alternative appealing choice in order to minimize the need of memory. In order to reduce the CPU time, it is convenient to stop the iterative solver as soon as the accuracy reaches that of the other parts of the numerical code, which is why the analysis of all the sources of truncation and round-off errors has been carried out in Section 4. Furthermore, the new approximate preconditioning technique presented in the same section permits us to reduce dramatically the total cost of the calculations, making this method as fast as the LU factorization and much less memory-consuming.

2. EQUATIONS AND PHYSICAL PARAMETERS

We consider the compressible Navier–Stokes equations for a perfect gas in a gravitational field,

$$\frac{\partial \vec{m}}{\partial t} = \vec{m} \times (\vec{\nabla} \times \vec{v}) - \tfrac{1}{2}\rho \vec{\nabla} v^2 - \vec{\nabla} p + \rho \vec{g} + \vec{F}_{\text{visc}}, \tag{3}$$

$$\frac{\partial T}{\partial t} = -\vec{v} \cdot \vec{\nabla} T - \frac{1}{c_{\text{v}}} T(\vec{\nabla} \cdot \vec{v}) + \frac{1}{\rho R c_{\text{v}}} \vec{\nabla} \cdot (\chi \vec{\nabla} T) + \frac{1}{C_{\text{v}} \rho} \Phi, \tag{4}$$

$$\frac{\partial \rho}{\partial t} = -\vec{\nabla} \cdot (\rho \vec{v}), \tag{5}$$

$$p = R \rho T, \tag{6}$$

where $\vec{m} = \rho \vec{v}$ is the momentum. In equation (3),

$$\vec{F}_{\text{visc}} = \mu\Big[\nabla^2 \vec{v} + \tfrac{1}{3} \vec{\nabla}(\vec{\nabla} \cdot \vec{v})\Big] + \vec{\nabla}(\vec{v} \cdot \vec{\nabla}\mu) + \vec{\nabla} \times (\vec{v} \times \vec{\nabla}\mu) - \vec{v}\nabla^2 \mu + \tfrac{1}{2}(\vec{\nabla}\mu)(\vec{\nabla} \cdot \vec{v}) \tag{7}$$

is the viscous force, with μ the dynamic viscosity coefficient. In equation (4) for the temperature,

$$\Phi = -v \cdot \vec{F}_{\text{visc}} + \vec{\nabla} \cdot \{\mu[\vec{\nabla}\vec{v}^2 - \vec{v} \times (\vec{\nabla} \times \vec{v}) - \tfrac{2}{3}\vec{v} \cdot (\vec{\nabla} \cdot \vec{v})]\}$$

is the dissipation function responsible for viscous heating, χ is the thermal conductivity, C_{v} and C_{p} are the specific heats, $c_{\text{v}} = C_{\text{v}}/R$ and $c_{\text{p}} = C_{\text{p}}/R$ are the rescaled values and R is the gas constant. The boundary conditions for the temperature are those of a thermally insulating wall. For the velocity we consider rigid boundary conditions.

All the tests that we shall present concern convective flows in a Cartesian geometry (a three-dimensional infinite horizontal layer, assuming periodicity in the two horizontal directions). The fluid is heated from below. We assume that the thermal conductivity χ and the dynamic viscosity coefficient μ are constant, that there are no heat sources inside the layer and that the gravity is vertical. We shall also assume a constant equilibrium density and a temperature gradient such that compressional effects are not negligible (i.e. Mach number of the order of 0·5). We shall not consider cases with shocks (Mach number larger than one), where spectral convergence is no longer exponential and the pseudospectral method is not very effective. Finally, the Prandtl number $Pr = \mu C_{\text{p}}/\chi$ will be taken equal to one.

The main parameter in our calculations is the Rayleigh number Ra_0 at the middle of the layer, which measures the importance of the destabilizing effect due to the buoyancy force compared with the stabilizing effects of the viscous and heat diffusion:

$$Ra_0 = \left.\frac{\mathrm{d}S}{\mathrm{d}z}\right|_0 \frac{g_0 L^4}{\chi \mu},$$

where z is the vertical co-ordinate, S is the entropy and L is the depth of the layer.

3. NUMERICAL METHOD

We use a pseudospectral method without removing aliasing. We use $k_{x_{\max}}$ and $k_{y_{\max}}$ Fourier modes in the horizontal periodic directions and $N_z + 1$ Chebyshev polynomials in the vertical direction.

The semi-implicit scheme used[3] allows us to relax the severe time step restrictions due to diffusive and wave-like terms.

For the momentum equation the numerical scheme is

$$\frac{\vec{m}^{n+1} - \vec{m}^n}{\delta t} = \tfrac{3}{2}\vec{F}^n_{\mathrm{AB}} - \tfrac{1}{2}\vec{F}^{n-1}_{\mathrm{AB}} + \tfrac{1}{2}\vec{F}^{n+1}_{\mathrm{SI1}} - \vec{F}_{\mathrm{SI1}} + \tfrac{1}{2}\vec{F}^{n-1}_{\mathrm{SI1}} + \vec{F}^{n+1}_{\mathrm{SI2}} - \vec{F}^n_{\mathrm{SI2}}, \tag{8}$$

where $\vec{F}_{\mathrm{AB}}$ represents the terms on the RHS of (3) that are treated using the Adams–Bashforth scheme, $\vec{F}_{\mathrm{SI1}}$ is the semi-implicit contribution for the viscous term and $\vec{F}_{\mathrm{SI2}}$ is the semi-implicit contribution for the wave-like terms:

$$\vec{F}_{\mathrm{AB}} = \vec{m} \times (\vec{\nabla} \times \vec{v}) - \tfrac{1}{2}\rho\vec{\nabla}v^2 - \vec{\nabla}p + \rho\vec{g} + \vec{F}_{\mathrm{visc}},$$

$$\vec{F}_{\mathrm{SI1}} = \tfrac{1}{2}\frac{\mu}{\rho_0}[\nabla^2\vec{m} + \tfrac{1}{3}\vec{\nabla}(\vec{\nabla}\cdot\vec{m})],$$

$$\vec{F}_{\mathrm{SI2}} = \delta t \left.\frac{\gamma P}{\rho}\right|_{\max} \vec{\nabla}(\vec{\nabla}\cdot\vec{m}).$$

A term similar to F_{SI2} has been proposed by Harned and Schnack[4] and has been tested by them in plasma physics computations. It has proven to be effective also in convective flows.[3]

The temperature equation is discretized as

$$\frac{T_{n+1} - T_n}{\delta t} = \tfrac{3}{2}G^n_{\mathrm{AB}} - \tfrac{1}{2}G^{n-1}_{\mathrm{AB}} + \tfrac{1}{2}G^{n+1}_{\mathrm{SI1}} - G^n_{\mathrm{SI1}} + \frac{1}{2}G^{n-1}_{\mathrm{SI1}},$$

with

$$G_{\mathrm{AB}} = -\vec{v}\cdot\vec{\nabla}T - \frac{1}{c_{\mathrm{v}}}T\vec{\nabla}\cdot\vec{v} + \frac{1}{\rho R c_{\mathrm{v}}}\vec{\nabla}\cdot(\chi\vec{\nabla}T) + \frac{1}{C_{\mathrm{v}}\rho}\Phi,$$

$$G_{\mathrm{SI1}} = \left.\frac{\chi}{\rho R c_{\mathrm{v}}}\right|_{\max} \nabla^2 T.$$

Finally, for the density we use the Crank–Nicolson scheme

$$\rho^{n+1} - \rho^n = -\frac{\delta t}{2}\vec{\nabla}\cdot(\rho^{n+1}\vec{v}^{n+1}) - \frac{\delta t}{2}\vec{\nabla}\cdot(\rho^n\vec{v}^n).$$

This equation is solved after the velocity equation, so that $\vec{v}^{n+1}$ is known. We solve this equation for ρ^{n+1} using a simple Richardson iterative scheme

$$\rho_0 = \rho^n, \qquad \rho_{i+1} = \rho^n - \frac{\delta t}{2}\vec{\nabla}\cdot(\rho^n\vec{v}^n) - \frac{\delta t}{2}\vec{\nabla}\cdot(\rho_i\vec{v}^n).$$

This scheme converges if the spectral radius of the linear operator L, such that $L\rho = \tfrac{1}{2}\delta t\vec{\nabla}\cdot(\rho\vec{v}^n)$, is less than one. This is true for δt sufficiently small; we have verified in all our computations that the maximum time step allowed by the semi-implicit scheme is indeed below this limit value.

The semi-implicit scheme gives rise to a set of linear problems that have to be solved at each time step. We shall give in Section 4.3 a description of the matrices involved and of the solution methods.

4. ERROR ANALYSIS

The goal of this section is both to develop a complete analysis of all the error sources in the code and to define a stopping criterion for the iterative procedure.

There are three independent sources of errors in time-dependent computations: (i) the truncation error, which comes from the fact that the solution at a given time is approximated with a finite number of Fourier and Chebyshev modes, (ii) the temporal discretization error, which arises from the discretization in time, and (iii) the round-off error, which is due to the fact that the calculations are performed with finite precision. The main sources of round-off errors come from the computation of pseudospectral derivatives and from the semi-implicit solver.

We analyse in the following these three sources of errors.

4.1. Spatial truncation error

The spatial absolute error has been measured by taking the absolute value of the z-component of the velocity, $|V_z(n, k_x, k_y)|$, and has been defined as

$$\varepsilon_{\mathrm{sa}}(N_z) = \max_{k_x,k_y} |V_z(N_z, k_x, k_y)|,$$

where N_z labels the last Chebyshev coefficient and k_x and k_y run over the Fourier mode numbers. These values become smaller and smaller as the resolution increases, because the spectrum decays exponentially near the viscous cut-off.

The relative truncation error is defined as

$$\varepsilon_{\mathrm{sr}}(n, N_z) = \frac{\varepsilon_{\mathrm{sa}}(N_z)}{\max_{k_x,k_y} |V_z(n, k_x, k_y)|}$$

and gives a measure of the relative precision of each mode. We have chosen this definition instead of the more usual one

$$\varepsilon_{\mathrm{sr}}(n, N_z) = \max_{k_x,k_y} \frac{|V_z(N_z, k_x, k_y)|}{|V_z(n, k_x, k_y)|},$$

because the denominator $|V_z(n, k_x, k_y)|$ approaches zero for some n, k_x and k_y, giving an artificially large relative error.

Large relative errors at small scales depend on the typical shape of the Chebyshev spectra, usually showing an accumulation of energy localized in the region of large wave numbers.

4.2. Temporal discretization error

This source of error can be estimated quite accurately by making use of Taylor expansion in time. The schemes used are Crank–Nicolson, Adams–Bashforth second-order and the two semi-implicit schemes in equation (8). These schemes introduce errors after one time step that are

$$\varepsilon_i^{\mathrm{S}}(n, k_x, k_y) = -C_{\mathrm{S}}\delta t^3 \frac{\mathrm{d}^2 f(u_i(n, k_x, k_y))}{\mathrm{d}t^2} + \mathcal{O}\left(\delta t^4 \frac{\mathrm{d}^3 f(u_i(n, k_x, k_y))}{\mathrm{d}t^3}\right),$$

where the index S labels the scheme: $C_{\mathrm{S}} = -\frac{5}{12}$ for Adams–Bashforth second-order, $C_{\mathrm{S}} = \frac{1}{12}$ for Crank–Nicolson and $C_{\mathrm{SI1}} = C_{\mathrm{SI2}} = \frac{1}{2}$ for both semi-implicit schemes of (8).

The total error $\varepsilon_i(n, k_x, k_y)$ is thus a weighted average of the second temporal derivative of the terms entering the RHS of the differential equation, multiplied by the cube of the time step. Generally, at the largest wave numbers the oscillations have larger frequencies. In fact, the large-scale motions have typical time scales that are the turnover time of the large eddies, L/V (L is the size of the system and V is the macroscopic velocity), the diffusion time of the large structures, L^2/ν, or the period of the large-scale compressional effects, L/c_s. On the other hand, small-scale amplitudes decrease with n but are subject to very rapid changes in time due to their rapid turnover, l/v, to the rapid viscous decay of their small structures, l^2/ν, or to the very rapid short period of sound waves, l/c_s. It turns out that the factor $d^2f(u_n)/dt^2$ is relatively less important at the larger scales than it is at the smaller scales; in other words, the temporal discretization induces a smaller relative error on the larger scales.

Starting from the total error, we can define the absolute temporal truncation error as

$$\varepsilon_{\text{dta}}(n) = \max_{k_x,k_y}(\varepsilon_{\text{dt}}(n, k_x, k_y)).$$

and the corresponding relative temporal truncation error as

$$\varepsilon_{\text{dtr}} = \max_{k_z,k_y} \frac{\varepsilon_{\text{dta}}(n)}{|V_z(n, k_x, k_y)|}.$$

If a purely explicit method were used, then one would be guaranteed that all the scales are well resolved in time, because that is precisely the condition for an explicit scheme to be stable. When one uses a semi-implicit scheme, one must be aware of the fact that some small-scale phenomena will not be correctly described. Semi-implicit operators introduce additional numerical dispersion which damps the high-frequency physical phenomena.[4] In our case we have chosen to handle the sound waves and the viscous operator implicitly. This means that the short sound waves and the small-scale viscous effects are not resolved at all in time. An estimate of the error introduced in this approximation is presented in the next section.

Finally we define the global temporal truncation error as $\varepsilon_{\text{dta}}(N_z)$. This number will be useful in the following to define the stopping criterion for the iterative solver.

4.3. Round-off error

The major contributions to the overall round-off error come from the computations of the derivatives and from the inversion of the linear system that arises from the semi-implicit part.

The Chebyshev pseudospectral algorithm to compute the pth derivative consists of three steps.[5]

1. Start from the value of the functions at the Gauss–Lobatto nodes $x_i = \cos(\pi j/N)$ and compute the coefficients a_k^0 of the Chebyshev expansion.
2. The coefficients a_k^p of the Chebyshev expansion of the pth derivative are expressed in terms of the coefficients of the $(p-1)$th derivative by the relation
$$\bar{c}_k a_k^p = a_{k+2}^p + 2(k+1)a_{k+1}^{p-1}, \tag{9}$$
with $a_N^p - p + k = 0$ for $k \geqslant 1$.
3. Reconstruct the value of the pth derivative at the Gauss–Lobatto nodes.

Steps 1 and 3 are achieved by making use of fast cosine transforms (FCTs). There are many algorithms that can be used to compute an FCT that have different properties with respect to rounding error. From numerical tests[6] and theoretical analysis[7] we derive that the 'good' (in terms of rounding error) algorithms produce an error in maximum norm that is uniformly distributed throughout the whole Fourier spectrum and that behaves like $\varepsilon c \log N$, c being a constant of order 10·7 and ε_0 the machine precision.

The 'bad' (in terms of rounding error) algorithms produce an additional amplification by a factor N of the rounding error produced during the FFT stage, thus bringing the total error of the FCT to

$\varepsilon c N \log N$. It is worth noting that the 'bad' algorithms are those generally used in the scientific community because they are the fastest.

During step 2 of the computation of the derivatives the error produced during step 1 is amplified, because the derivative operator is ill-conditioned. The condition number of the matrix for the pth derivative in spectral space scales like N^{2p}. The error is therefore amplified by this factor.

A second source of rounding error comes from the solution of the linear system due to the semi-implicit calculation. The matrix arising from the discretization of (8) using Fourier–Fourier–Chebyshev decomposition is block diagonal, each block of complex linear equations arising from the Chebyshev decomposition corresponding to a given couple k_x, k_y of Fourier wave numbers. The following set of linear systems has to be solved at each time step:

$$A(k_x, k_y)x(k_x, k_y) = b(k_x, k_y), \quad k_x = 0, \ldots, k_{x_{\max}}, \quad k_y = 0, \ldots, k_{y_{\max}},$$

where $A(k_x, k_y)$ is a $6(N_z + 1) \times 6(N_z + 1)$ matrix and $b(k_x, k_y)$ and $x(k_x, k_y)$ are vectors of dimension $6(N_z + 1)$.

The corresponding round-off error depends on the choice of the solver. When direct methods are used, the backward analysis provides an upper bound to the relative error[8]

$$\varepsilon_{\mathrm{r}}^{\mathrm{si}} = \frac{\|\Delta x\|_\infty}{\|x\|_\infty} \leqslant \frac{\omega\kappa_1}{1 - \omega\kappa_2}, \quad \text{with } \omega = \max_i \frac{|Ax - b|_i}{(|A||x| + |b|)_i}, \tag{10}$$

where

$$\kappa_1 = \frac{\||A^{-1}|(|A||x| + |b|)\|_\infty}{\|x\|_\infty}, \qquad \kappa_2 = \||A^{-1}||A|\|_\infty. \tag{11}$$

This analysis allows us to verify that the LU method produces unnecessarily accurate solutions. The mean value of $\varepsilon_{\mathrm{r}}^{\mathrm{si}}$ is actually $\mathcal{O}(10^{-20})$, much smaller than the other contributions to the global error. This also allows us to assume the LU solution as the *exact* solution, so that for iterative methods the residual error can be defined as

$$r(n, k_x, k_y) = x(n, k_x, k_y) - x_{\mathrm{LU}}(n, k_x, k_y) \tag{12}$$

and the absolute error as

$$\varepsilon_{\mathrm{abs}}^{\mathrm{si}} = \max_{k_x, k_y, n} |r(n, k_x, k_y)|. \tag{13}$$

5. RESULTS

In order to set the optimal working parameters of the iterative solver and of the preconditioning operator, we have started with 2D simulations, while 3D simulations have been performed later to check the algorithm in more realistic configurations. We have taken as initial conditions a random white noise in frequency space with a very small amplitude.

In the first 2D stage we have compared the performances of iterative-like solvers with the direct LU one for two values of Ra, namely $Ra = 5Ra_{\mathrm{c}}$ and $50Ra_{\mathrm{c}}$.

Our matrices are non-symmetric and ill-conditioned. Following Reference 9, we have tried the GMRES algorithm. As a preconditioner we have found that the incomplete LU factorization of A gives excellent results.

The 2D preconditioned problem is

$$\mathscr{L}^{-1}(k_y)A(k_y)x(k_y) = \mathscr{L}^{-1}(k_y)b(k_y), \quad k_y = 0, \ldots, k_{y_{\max}}, \tag{14}$$

but in this formulation the iterative solver would be extremely time-consuming, because for each k_y, computing $\mathscr{L}^{-1}(k_y)$ has almost the same cost as a complete factorization.

We have therefore considered the simplified formulation

$$\mathscr{L}^{-1}(\bar{k}_y)A(k_y)x(k_y) = \mathscr{L}^{-1}(\bar{k}_y)b(k_y), \quad k_y = 0, \ldots, k_{y_{\max}}, \tag{15}$$

where $\mathscr{L}^{-1}(\bar{k}_y)$ is the preconditioning matrix computed for a fixed $\bar{k}_y$. Indeed, as the matrix $A(k_y)$ does not depend so much on k_y, the same preconditioner computed for a particular wave number may be used for all the others. This property allows us to reduce dramatically the number of floating point operations and makes this technique competitive with respect to the LU one in terms of CPU time. The global memory storage, on the other hand, reduces from $k_{y_{\max}} + 1$ matrices to one.

Concerning the stopping criterion, the number of iterations required by the GMRES algorithm increases with the precision demanded, so that the proper choice of the tolerance is important to minimize the CPU time.

Figure 3 clearly shows that the main contribution to the global error arises from the truncation error in both space and time and that a good choice for the tolerance demanded by GMRES is $\mathcal{O}(10^{-7})$ for $Ra = 5Ra_c$ and 10^{-6} for $Ra = 50Ra_c$.

In the $Ra = 5Ra_c$ test both solvers have achieved the same steady state pattern after roughly 1000 time steps, corresponding to one diffusion time. The two simulations display the same transient

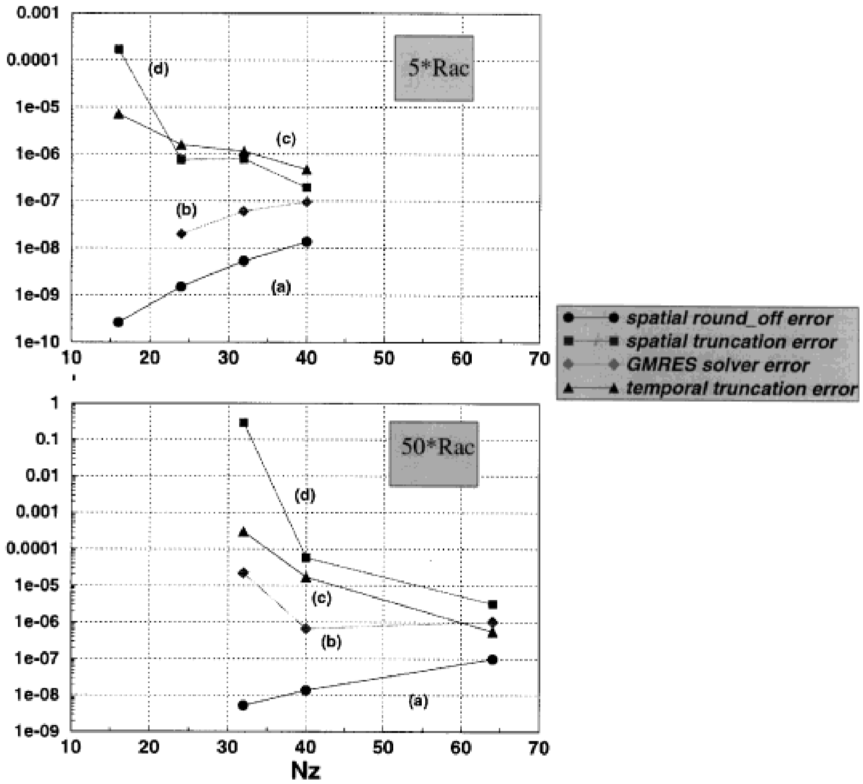

Figure 3. Global round-off and truncation errors for $Ra = 5Ra_c$ and $50Ra_c$. All runs have been performed in two dimensions with $k_{y_{\max}} = 24$. The errors are evaluated near the end of the simulation

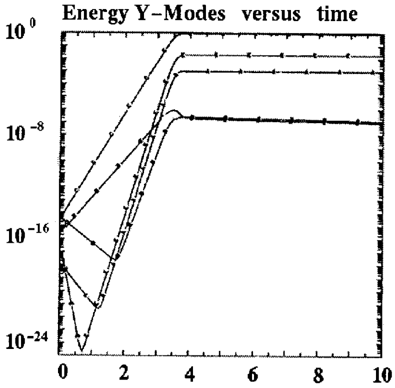

Figure 4. Kinetic energy of different k_y-modes as a function of time for 2D run at $Ra = 5Ra_c$. Time is in units of thermal diffusion time

behaviour and the same steady state pattern (Figures 4 and 5). In both cases we have used the maximum time step allowed by the semi-implicit scheme.

The absolute temporal truncation error is plotted in Figure 6 at two different times and the corresponding relative temporal truncation error is plotted in Figure 7. From Figure 6 we see that the absolute error is of the same order at all the scales, while from Figure 7 it is apparent that the relative error is much larger at small scales. The temporal error, due to the numerical dispersion introduced by the semi-implicit operator, is the dominating one. It is interesting to see in Figure 6 that at the beginning of the simulation, when the influence of the transient is still strong, each mode oscillates at its own frequency, curve (a), whereas a steady pattern is achieved at the steady state configuration, curve (b), after one diffusion time.

In Figure 7 one sees that the pattern of the temporal relative error (a) is not so different from the spatial error (b), both being determined by the shape of the Chebyshev spectrum.

Figure 8 shows the CPU time performances. For the LU curve we assume that the matrix has already been factorized, so it grows as N_z^2 independently of the *value* of the wave number k_y. This is not true for the GMRES case: increasing values of k_y leads to slower convergence. In the same figure we show three curves for the GMRES case. The lower one corresponds to the problem $A(1)x(1) = b(1)$, while the upper one is for $A(K_{y_{\max}})x(k_{y_{\max}}) = b(k_{y_{\max}})$. For a given k_y we observe that the GMRES time grows almost linearly with N_z, so that as the size of the problem increases, performances more and more favourable to the iterative solver are expected.

The total CPU times have been recorded in Table I and show that the GMRES code, though memory-saving, is about twice slower than the LU version.

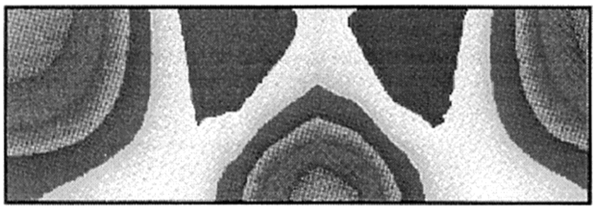

Figure 5. Pressure fluctuations of steady state solution of 2D convection at $Ra = 5Ra_c$

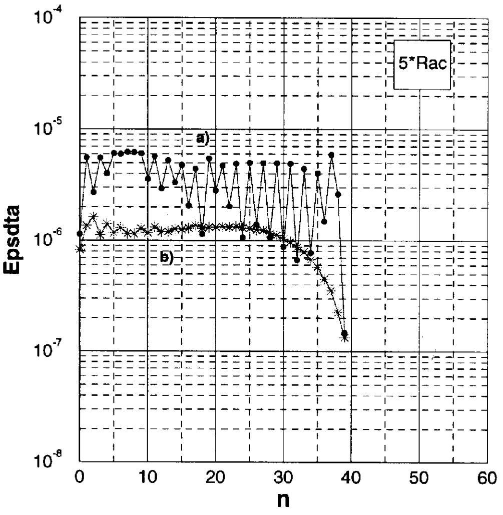

Figure 6. Absolute temporal truncation errors at two different times of integration process: (a) transient phase; (b) near converged stationary state. We have taken $N_z = 40, k_{y_{\max}} = 24$ and the run is in two dimensions

At larger *Ra* the flow becomes more complex and time-dependent (see Figure 9). In 3D calculations, moderate Rayleigh numbers ($Ra = 50Ra_c$) have been used starting from the configuration of axisymmetric rolls that is the stationary solution at $5Ra_c$. This configuration undergoes transition to a more elaborate and time-dependent pattern. In Figure 10 we show the pattern near the initial stage, in Figure 11 a sequence of states at different times and in Figure 12 the time evolution of the velocity components at a selected point. In all phases of the flow field we have controlled the absolute and relative errors and we have found that they do not differ qualitatively from those obtained in the 2D case at $Ra = 5Ra_c$ (Figures 6 and 7). For the 3D case we have used the approximate preconditioning technique based on the matrix $\mathscr{L}^{-1}(k_x = 1, k_y = 1)$. We have found that this preconditioner was able to produce very fast and accurate convergence for every k_x and k_y.

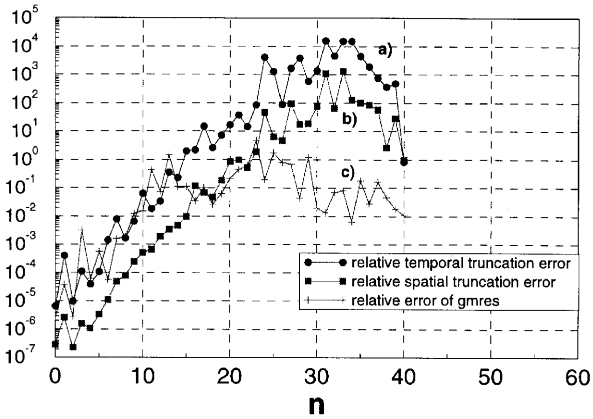

Figure 7. Relative errors due to (a) temporal and (b) spatial truncation and (c) relative error of iterative GMRES method at time near converged stationary state. We have taken $N_z = 40, k_{y_{\max}} = 24, Ra = 5Ra_c$ and the run is in two dimensions

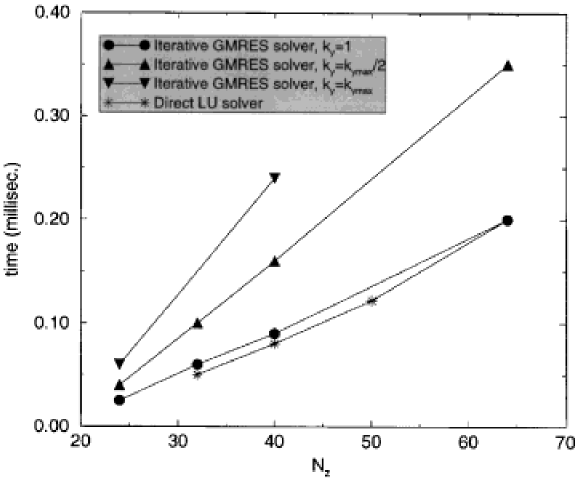

Figure 8. Time required on a Cray-2 to solve $4(N_z + 1) \times 4(N_z + 1)$ linear system arising from semi-implicit algorithm. The curves shown are those for the direct LU solver and for the iterative GMRES solver with the incomplete LU preconditioner described in Section 4 for a given k_y (and $k_x = 1$). The tolerance demanded for the iterative solver is $\varepsilon = 10^{-8}$. The horizontal truncation is $k_{y_{\max}} = 24$

Table I. CPU times (in seconds) required by LU and GMRES solvers for two runs with resolutions $N_z = 40$ and 64 on a Cray-2. In both cases $k_{y_{\max}} = 24$. The second and third columns give the total CPU times for the simulation. The fourth column gives the CPU time spent outside the linear solvers. The fifth column is the ratio between the second and third columns. The sixth column gives the ratio between the time spent in the direct LU solver and the time spent in the GMRES solver

N_z	Total CPU LU	Total CPU GMRES	Residual CPU	TOTAL $\mathrm{CPU_{LU}}/\mathrm{CPU_{GM\,RES}}$	$\mathrm{CPU_{LU}}/\mathrm{CPU_{GM\,RES}}$
40	17·2	22·0	15·2	0·78	0·29
64	25·50	49·0	20·5	0·52	0·18

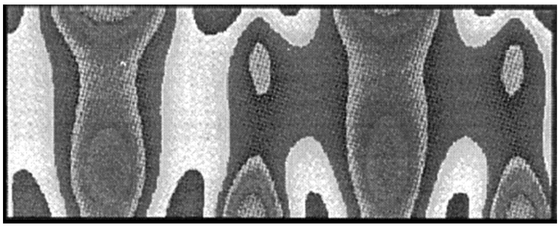

Figure 9. Snapshot of pressure fluctuations for 2D simulation at $Ra = 50Ra_c$

6. CONCLUSIONS

We have shown in this paper that the GMRES algorithm with the incomplete LU preconditioner is a very effective choice. In order to avoid the high cost of the preconditioner, one would have to compute all the preconditioning matrices once at the beginning of the code and to save them with an unacceptable memory occupancy. A clever modification of the preconditioners, namely taking them

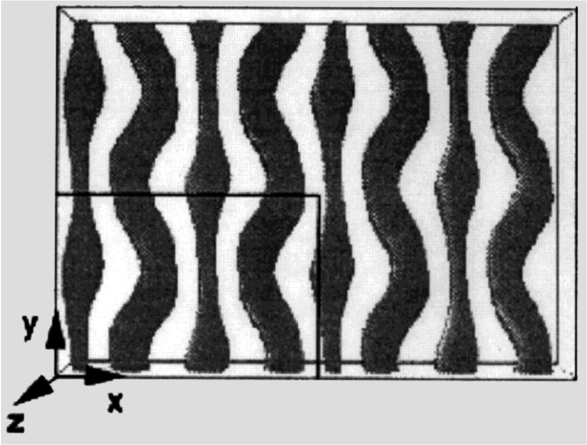

Figure 10. Pattern of convection near initial condition for 3D run at $Ra = 50Ra_c$

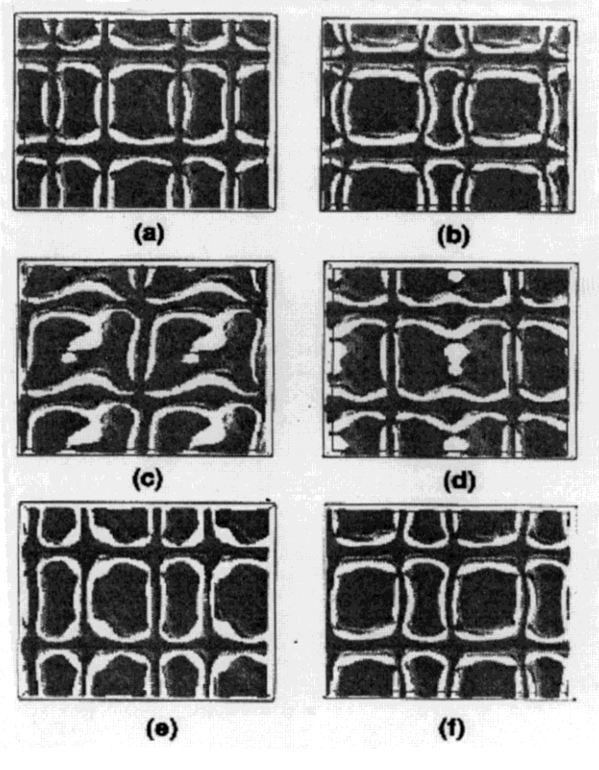

Figure 11. Iso-surface of constant vertical velocity at statistically stationary stage for 3D simulation at $Ra = 50Ra_c$. The interval between two successive figures is 0·4 thermal diffusion times

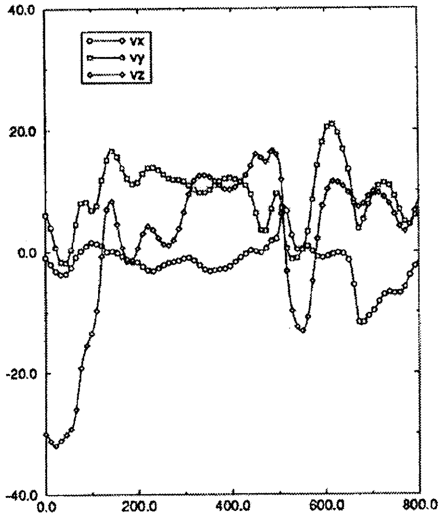

Figure 12. Time evolution of velocity components at one point. Time is in units of large-eddy turnover time. The total time corresponds to roughly three thermal diffusion times

as independent of the Fourier wave numbers, overcomes the limitation in memory and does not appreciably increase the CPU time cost.

In this paper we have also produced a complete error analysis of pseudospectral computations. This allows us to control the global error produced in a numerical simulation, thus giving more confidence in the results. Also, the application of the error analysis gives a proper stopping criterion for the iterative scheme, avoiding the production of unnecessarily accurate solutions.

ACKNOWLEDGEMENTS

We acknowledge Dr. Mario Arioli and Dr. Maurice Meneguzzi for very interesting discussions.

REFERENCES

1. S. Chandrasekhar, *Hydrodynamic and Hydromagnetic Stability*, Cambridge University Press, Cambridge, 1961.
2. F. Cattaneo, N. H. Brumell, J. Toomre, A. Malagoli and N. E. Hurlburt, 'Turbulent compressible convection', *Astrophys. J.*, **370**, (1991).
3. L. Valdettaro, 'Simulations numériques d'écoulements magnetohydrodynamiques compressibles, en géométrie sphérique', *Ph.D. Thesis*, Université Paul Sabatier, Toulouse, 1992.
4. D. S. Harned and D. D. Schnack, 'Semi-implicit method for long time scale magnetohydrodynamic computations in three dimensions', *J. Comput. Phys.*, **65**, 57 (1986).
5. D. Gottlieb and S. A. Orszag, *Numerical Analysis of Spectral Methods: Theory and Applications*, vol. 370, p. 1282, SIAM-CBMS, Philadelphia, PA, 1977.
6. E. E. Rothman, 'Reducing round-off error in Chebyshev pseudospectral computations', in M. Durand and F. El Dabaghi (eds), *High Performance Computing II*, North-Holland, Amsterdam, 1991, pp. 423–439.
7. M. Arioli and L. Valdettaro, 'Round-off error analysis of the fast cosine transform and its application to the Chebyshev pseudospectral method', *East–West J. Numer. Math.*, **3**, 43–58 (1995).
8. M. Arioli, I. Duff and D. Ruiz, 'Stopping criteria for iterative solvers', *SIAM J. Matrix Anal. Appl*, **13**, 138–144 (1992).
9. N. M. Nachtigal, S. C. Reddy and L. N. Trefethen, 'How fast are nonsymmetric matrix iterations?', *SIAM J. Matrix Anal. Appl.*, **13**, 778 (1992).