INT. J. NUM. METH. HEAT FLUID FLOW, VOL. 2, 195–213 (1992)

DIRECT SPECTRAL METHODS FOR THE LOW MACH NUMBER EQUATIONS

J. FRÖHLICH* AND R. PEYRET

Laboratoire de Mathématiques, Université de Nice, CNRS UA 168, Parc Valrose, F-06000 Nice, France

ABSTRACT

The low Mach number approximation of the Navier–Stokes equations is of similar nature to the equations for incompressible flow. A major difference, however, is the appearance of a space- and time-varying density that introduces a supplementary non-linearity. In order to solve these equations with spectral space discretization, an iterative solution method has been constructed and successfully applied in former work to two-dimensional natural convection and isobaric combustion with one direction of periodicity. For the extension to other geometries efficiency is an important point, and it is therefore desirable to devise a direct method which would have, in the best case, the same stability properties as the iterative method. The present paper discusses in a systematic way different approaches to this aim. It turns out that direct methods avoiding the diffusive time step limit are possible, indeed. Although we focus for discussion and numerical investigation on natural convection flows, the results carry over for other problems such as variable viscosity flows, isobaric combustion, or non-homogeneous flows.

KEY WORDS Low Mach number Rayleigh–Bénard problem Spectral space discretization

INTRODUCTION

Spectral paradise is the constant coefficient heat equation with periodic boundaries: the development of the unknown in a Fourier series leads to a set of ordinary differential equations in time, one for each mode, of which the solution is straightforward. In real life, however, several kinds of modifications introduce difficulties. The first is due to the boundary conditions that are rarely periodic and may even concern complex geometry. A second one (among others) arises from the presence of non-linearities in the equations which would prevent in the above example the uncoupling of different modes; non-linear operators are of course more difficult to handle than linear ones. In the present paper we shall focus on this aspect and discuss its consequences for the numerical solution of the equations governing low Mach number flows in two-dimensional geometry with one direction of periodicity.

Let us first recall some general features of current spectral methods in fluid dynamics[1]. One point is that, although based on spectral space discretization, these methods generally employ a time scheme of finite difference type. This is due to several reasons among which involved programming for spectral time-discretization plays the key role[2,3]. Explicit time discretization of the different terms may lead to corresponding limitations of the time step. These are restrictive in particular for terms that contain higher order derivatives. Due to the eigenvalues of the discrete differentiation operator the critical time step generated by a term of pth order derivative

* Present address: Fachvereich Mathematik, Universität Kaiserslautern, 6750 Kaiserslautern, Germany.

0961–5539/92/030195–19$2.00

Received April 1991
Revised September 1991

behaves, as a rule of thumb, like:

$$\Delta t_{\text{crit}} \sim \frac{1}{N^p} \qquad \text{Fourier series,} \quad |x_{i+1}-x_i| \sim 1/N \tag{1}$$

$$\Delta t_{\text{crit}} \sim \frac{1}{N^{2p}} \qquad \text{Chebychev series,} \quad \min\{|x_{i+1}-x_i|\} \sim 1/N^2 \tag{2}$$

where N is the degree of approximation and x_i the related collocation points[1]. The general strategy now is to apply implicit discretization to the most restrictive terms and to treat the remainder by an explicit scheme.

Let us note for illustration and comparison the non-dimensional equations of motion for a fluid with constant density and viscosity that read in primitive variables:

$$\partial_t \mathbf{v} + (\mathbf{v}.\nabla)\mathbf{v} + \nabla p = \frac{1}{Re}\nabla^2 \mathbf{v} \tag{3}$$

$$\nabla.\mathbf{v} = 0 \tag{4}$$

where $\mathbf{v} = (u, v)^t$ is the velocity, p the pressure, and Re the Reynolds number. A time scheme having become widely adopted for spectral methods that follows the above remarks is the three-level second order Euler-backward/Adams–Bashforth scheme[4]. It reads for the above equations:

$$\frac{3}{2\Delta t}\mathbf{v}^{n+1} - \frac{1}{Re}\nabla^2 \mathbf{v}^{n+1} + \nabla p^{n+1} = -\{(\mathbf{v}.\nabla)\mathbf{v}\}^{n,n-1} + \frac{1}{2\Delta t}(4\mathbf{v}^n - \mathbf{v}^{n-1}) \tag{5}$$

$$\nabla.\mathbf{v}^{n+1} = 0 \tag{6}$$

and has proven good stability properties in many cases. For small values of Re the scheme is unconditionally stable, contrarily to the Crank–Nicolson/Adams–Bashforth scheme[5]. When starting up a calculation, the scheme (5) is at the first time step replaced by a similar but two-level first order scheme. Observe that the diffusive term on the implicit side in (5) only contains linear operators with constant coefficients. This is favourable for the application of efficient techniques usually associated with spectral approximation like Tau method, influence matrix technique, and diagonalization. On the contrary, the non-linear convective term is treated explicitly, and we use the notation:

$$\{f\}^{n,n-1} = 2f^n - f^{n-1} \tag{7}$$

for the second order Adams–Bashforth extrapolation.

The particularity of the Navier–Stokes equations for incompressible fluids is the fact that the non-linearity is confined to the convective terms which contain first order spatial derivatives only, whereas the diffusion term is linear. The situation is no longer as simple as this if the diffusion term is non-linear as e.g. for a variable viscosity considered by Malik *et al.*[6]. This is similar when one considers the full Navier–Stokes equations for compressible fluids[7,8], or the low Mach number equations as in the present study. The difficulty arises from the fact that the derivatives in time and the highest order derivatives (diffusive terms) do not carry on the same variable (e.g. $\rho\mathbf{v}$ and $\mathbf{v}$ for the conservative form of the momentum equation). In this case, following the general procedure suggested by Gottlieb and Orszag[2], a semi-implicit scheme may be used as remedy. It is constructed by means of explicit discretization of the diffusion term and a stabilizing term which is added on both sides of the equation. On one side this stabilizing term is treated by an implicit scheme, on the other side it counterbalances a part of the diffusion term in the explicit part of the scheme. The choice of this supplementary term strongly influences the numerical stability. It is restricted *a priori* by the possibilities of the algorithm applied for inversion of the implicit operator and should take into account in some sense the major part of the diffusive term. This can be obtained for example by using a linearized expression of this term or other arguments based on the physical nature of the flow.

The above methodology applies in this form mainly to cases where the most restrictive stability condition arises from the diffusive term, as in low Reynolds number flows. In some cases of natural convection it leads even to unconditionally stable time schemes. On the other hand, in situations where the convective term determines the overall stability (CFL condition), the direct application of the time schemes presented below does not necessarily improve stability. In such a case, however, a semi-implicit treatment of the convection term in the same spirit as applied in this paper should be successful, especially in the presence of a strong basic flow, a situation that often appears in typical applications.

A fully non-linear implicit algorithm, as developed for a spectral method by Gerhold[9] and Fröhlich *et al.*[10] can be of advantage, of course. The unconditional stability is, however, obtained with additional computational effort at each time cycle, so that it is of interest in particular when one cannot get rid of too restrictive critical time steps by other means.

LOW MACH NUMBER EQUATIONS

The equations for natural convection under the classical Boussinesq approximation read like (3), (4) with a supplementary volume force that linearly depends on temperature being determined by an additional transport–diffusion equation. Consequently, the above scheme (5), (6) is generally applied to this kind of flow (similarly, when considering other formulations such as vorticity–streamfunction, etc.).

The set of equations that we will be concerned with in this paper are low Mach number (LM) equations that are of more general character than the Boussinesq equations. They have to be used for various problems of technical and theoretical interest, such as free and forced convection, crystal growth, isobaric combustion and others, since in these situations the hypotheses for the Boussinesq approximation do no longer hold. Recent developments in the cited fields justify the interest in devising efficient spectral solution algorithms, in particular for the study of stability problems where high accuracy is desirable (see Fröhlich and Peyret[11] for the stability of a plane flame front).

The LM equations are obtained when developing the complete Navier–Stokes equations in terms of γM^2, $M \ll 1$ being the characteristic Mach number of the flow, and γ the ratio of specific heats. Retaining lowest order terms gives in non-conservative form

$$\partial_t \rho + \nabla.(\rho \mathbf{v}) = 0 \tag{8}$$

$$\rho(\partial_t \mathbf{v} + (\mathbf{v}.\nabla)\mathbf{v}) + \nabla p = \mathbf{f}_g + Pr\,\nabla.\tau \tag{9}$$

$$\rho(\partial_t \theta + \mathbf{v}.\nabla\theta) - \frac{\gamma - 1}{\gamma} \mathrm{d}_t p_0 = \nabla^2 \theta \tag{10}$$

$$\rho = \frac{p_0}{\theta} \tag{11}$$

$$\tau = \nabla\mathbf{v} + (\nabla\mathbf{v})^t - 2/3(\nabla.\mathbf{v})\mathbf{I} \tag{12}$$

as indicated by Paolucci[12] and Chenoweth and Paolucci[13]. In (8)–(12) we have supposed an ideal gas with constant specific heats, viscosity and heat conductivity, but these assumptions are not essential. Similar equations can be deduced if they do not hold, and the exposed techniques carry over immediately. The notation of the dimensionless variables is the following: ρ is the density, $\mathbf{v}$ the velocity, θ the temperature and Pr the Prandtl number that fulfils here $Pr = 1/Re$ since a reference velocity based on heat diffusion has been chosen. The gravitational force $\mathbf{f}_g$ reads

$$\mathbf{f}_g = \left(0;\ -\frac{Ra\,Pr}{2\varepsilon}\rho\right)^t \tag{13}$$

where $\varepsilon=\Delta T/2T^*$ is the characteristic temperature difference in the flow field ΔT, non-dimensionalized by two times the reference temperature T^*, and Ra is the Rayleigh number. The underlying development of all dependent variables reads for the pressure:

$$p=p^{(0)}+\gamma M^2 p^{(1)}+\cdots \tag{14}$$

The superscript (0) has been suppressed in (8)–(13) for all quantities ($\mathbf{v}=\mathbf{v}^{(0)}$, $\theta=\theta^{(0)}$, $\rho=\rho^{(0)}$, $p_0=p^{(0)}$) and we denote $p=p^{(1)}$, which is the only quantity of first order in these equations. It plays the same role as the pressure in incompressible flows, (3)–(4). The pressure $p_0=p^{(0)}$ is constant in space and is determined by a supplementary scalar equation deduced from global conservation of mass. This equation reads:

$$p_0=\frac{M_A}{\int_{\Omega_A} 1/\theta \,\mathrm{d}\Omega_A} \tag{15}$$

and is obtained by integration of the state equation and the fact that p_0 is constant in space. It guarantees that the mass $M_A=\int_{\Omega_A}\rho|_{t=0}\,\mathrm{d}\Omega_A$ initially enclosed in the computational domain Ω_A remains constant, without being affected by accumulation of temporal or spatial truncation errors.

Instead of (8) we will occasionally use:

$$\nabla.\mathbf{v}=\frac{1}{p_0}\left(-\frac{1}{\gamma}\mathrm{d}_t p_0+\nabla^2\theta\right) \tag{16}$$

derived from (8), (10) and (11), for its resemblance to (4). The fundamental difference, however, is the non-vanishing rhs in (16) so that the problem of compatibility between velocity boundary conditions and thermodynamic variables arises, discussed e.g. by Majda[14]. For a closed domain, the boundary conditions on the velocity (rigid walls, periodicity in x) lead to

$$\int_{\Omega_A}\nabla.\mathbf{v}\,\mathrm{d}\Omega_A=0 \tag{17}$$

so that a solution to (16) only exists, if

$$\mathrm{d}_t p_0=\frac{\gamma}{\int_{\Omega_A}\mathrm{d}\Omega_A}\int_{\Omega_A}\nabla^2\theta\,\mathrm{d}\Omega_A \tag{18}$$

From a continuous point of view, (15) and (18) are equivalent. As with (15), (18) is obtained from (8), having multiplied by θ and using (10). A numerical method, however, introduces discretization errors and it is important to take care that these cannot deteriorate the calculation[15].

Owing to the structure of the equations that is similar to the one of those of an incompressible fluid, it is natural to use a temporal discretization that permits to split up the calculation at each time level $n+1$ in three parts as in Fröhlich and Peyret[15]: (1) determination of θ^{n+1} from the temperature equation by a direct scheme, (2) constant mass constraint in a closed domain giving p_0^{n+1}, state equation for ρ^{n+1}, (3) solution of momentum and continuity equation to determine $\mathbf{v}^{n+1}$ and p^{n+1}. In the present paper we will only deal with the third of these steps. It is the central part of the algorithm and, moreover, step (1) makes use of the same techniques as applied to the momentum equation in step (3) and has been described in the above reference. Consequently, from now on $\theta=\theta^{n+1}$, $p_0=p_0^{n+1}$, $\mathrm{d}_t p_0=\mathrm{d}_t p_0^{n+1}$, and $\rho=\rho^{n+1}$ are assumed to be known.

After these general remarks on the temporal discretization, we start with some brief indications on the technique applied for spatial discretization in order to motivate the way in which the implicit part of the time scheme is constructed. Then, in the main part of this paper, three direct methods for the solution of the LM equations are presented. Their discussion permits to point out the main difficulties associated with the temporal discretization of this type of equation and to develop adequate strategies. Two of the algorithms make use of the velocity–pressure

formulation. The most promising is the third one based on a momentum–pressure formulation. This is supported by the numerical results obtained for the Rayleigh-Bénard problem. We finally make some remarks on extensions to 2D fully non-periodic and 3D problems where computational efficiency is a crucial point.

RAYLEIGH–BÉNARD PROBLEM AND THE SPATIAL APPROXIMATION

Although the presented methods apply to a large class of flows, the discussion and the numerical results reported later on in this paper concern the two-dimensional Rayleigh–Bénard problem of a horizontal fluid layer heated from below. The boundary conditions are

$$\mathbf{v}=0 \quad \text{at} \quad y=\pm 1 \tag{19}$$

$$\theta=1+\varepsilon \quad \text{at} \quad y=-1 \qquad \theta=1-\varepsilon \quad \text{at} \quad y=1 \tag{20}$$

The solution is supposed to be periodic in the horizontal x-direction with period length $2A$ (the computational domain Ω_A is of aspect ratio A). The initial condition for a calculation with a given Rayleigh number is the hydrostatic solution, randomly perturbed, or the solution obtained for a different Rayleigh number.

The iterative algorithm presented in Frölich and Peyret[15] and briefly recalled in the following section has been applied to solve the LM equations (9)–(16) in the above geometry, and further results are to be found in Frölich *et al.*[16]. The spatial discretization is the same as in the above references, so that only some key points need to be recalled here: the present boundary conditions lead to an expansion of each dependent variable in a Fourier series with respect to x and Chebychev polynomials

$$T_n(y)=\cos(n \arccos(y)) \qquad n\geqslant 0 \tag{21}$$

in the non-periodic y-direction. (As the latter are defined on the interval $y\in[-1;1]$, we have introduced a scaling factor of 2 for the coordinates.) Spatially discrete equations are obtained using a Fourier–Galerkin/Chebychev-collocation method wherein the non-linear terms are evaluated by a pseudo-spectral technique[2]. The number of collocation points in x-direction is denoted N_F. In y-direction, we use N_C+1 Chebyshev Gauss–Lobatto collocation points that read

$$y_j=\cos\left(\frac{\pi j}{N_C}\right) \qquad j=0,\ldots,N_C \tag{22}$$

for all quantities, except the pressure p. Note that for the present problem the pressure is not involved in the physical boundary conditions, unlike e.g. in free surface flows. Hence, it is convenient to define the pressure on the Chebyshev–Gauss points:

$$\eta_j=\cos\left(\frac{\pi(j+1/2)}{N_C}\right) \qquad j=0,\ldots,N_C-1 \tag{23}$$

Moreover, the use of such a staggered mesh avoids spurious modes for this quantity as e.g. in Malik *et al.*[6]. The conservation of momentum is imposed on y_j and continuity on η_j. Consequently, the unknowns are the values of the Fourier coefficients at the Chebyshev collocation points.

The derivatives in x-direction are efficiently carried out in Fourier space. Derivations in y-direction can be implemented in a similar manner using FFT-algorithms, but on vector computers the use of matrix multiplications is competitive and for relatively small numbers of modes even faster, as shown e.g. by Canuto *et al.*[1], so that this latter is employed here.

Obviously, expressions of type $c\,\partial_{xx}u$, $c(y)\partial_{xx}u$ etc. lead to uncoupling of modes in Fourier space which is not the case when c depends on x as in $c(x,y)\partial_{xx}u$, for instance. Furthermore, since in a term like $c(y)\partial_{yy}u$ or $\partial_{yy}(c(y)u)$ the discrete differentiation with respect to y is carried out by multiplication with a differentiation matrix, the coefficient $c(y)$ can be accounted for by

simply modifying this matrix. This is of advantage especially when c does not depend on time, so that the modified differentiation matrix is calculated once and for all. Note that the uncoupling of different Fourier modes considerably reduces the complexity of the problem. It corresponds to the transition from a full matrix to a block diagonal matrix.

DIRECT SCHEME BASED ON A VELOCITY–PRESSURE FORMULATION

In this section we describe a direct algorithm for the solution of (9), (16). It is based on the scheme (5), (6) with some modifications due to the present equations. The density ρ throughout plays the role of a known coefficient, different at every time step. For the construction of the algorithm it is convenient to introduce a temporally constant state ρ_s destined to replace ρ in front of the time-derivative in (9). In the simplest case it may just be a constant, but together with Fourier space discretization in x we can easily set $\rho_s=\rho_s(y)$ and use for the case of natural convection the hydrostatic density distribution. As the following discussion will show, ρ_s should generally be chosen as close as possible to ρ and capture its characteristic variation. After multiplication of (9) by ρ_s/ρ the time-discretized version of (9) and (16) reads:

$$\sigma\rho_s\mathbf{v}^{n+1}+\mathbf{s}(\mathbf{v}^{n+1})+\nabla p^{n+1}=\mathbf{f}_g+\left\{-\rho_s(\mathbf{v}.\nabla)\mathbf{v}+Pr\frac{\rho_s}{\rho}\nabla.\tau\right\}^{n,n-1}+\left\{\left(1-\frac{\rho_s}{\rho}\right)\nabla p\right\}^{n,n-1}+\{\mathbf{s}(\mathbf{v})\}^{n,n-1}+\sigma\rho_s\left(\frac{4}{3}\mathbf{v}^n-\frac{1}{3}\mathbf{v}^{n-1}\right) \quad (24)$$

$$\nabla.\mathbf{v}^{n+1}=\frac{1}{p_0}\left(-\frac{1}{\gamma}\mathrm{d}_t p_0+\nabla^2\theta\right) \quad (25)$$

where $\sigma=3/2\Delta t$. The pressure has to be discretized at least partly implicitly as the mass conservation constraint is imposed on the level $n+1$ which is a usual feature of 'incompressible' algorithms. The notation in (24) emphasizes the use of a stabilizing term s. In fact, we choose:

$$\mathbf{s}(\mathbf{v})=-Pr\,\tilde{\nabla}^2\mathbf{v}=-Pr\left(\frac{4}{3}\partial_{xx}u+\partial_{yy}u;\ \partial_{xx}v+\frac{4}{3}\partial_{yy}v\right)^t \quad (26)$$

so that the explicit contribution of the diffusion term in the rhs of (24) reads:

$$\frac{\rho_s}{\rho}Pr\,\nabla.\tau+\mathbf{s}(\mathbf{v})=Pr\left(\frac{1}{3}\frac{\rho_s}{\rho}\partial_{xy}\mathbf{w}+\left(1-\frac{\rho_s}{\rho}\right)\tilde{\nabla}^2\mathbf{v}\right) \quad (27)$$

with $\mathbf{w}=(v;u)^t$. Note that, following the remarks made in the previous section, terms with second order derivatives in y-direction induce the most restrictive critical time step for low velocities. The severe stability condition has its origin in the diminishing spatial scale of the Chebyshev polynomials in the vicinity of the boundaries at $y=\pm1$, which is reflected by a decrease of the distance between the collocation points proportional to N^{-2}, see (2), (22), (23). The reason for the introduction of the factor ρ_s clearly appears in (27). First, if the density ρ is not too different from the hydrostatical distribution ρ_s, the effect of the diffusion term $(1-\rho_s/\rho)\tilde{\nabla}^2\mathbf{v}$ treated explicitly is weak with respect to the implicit term s. In particular from the present Dirichlet boundary conditions on θ and the fact that p_0 generally varies only little with respect to its initial value, we get that $\rho_s/\rho\approx1$ near $y=\pm1$. Consequently, the diffusion term $(1-\rho_s/\rho)\tilde{\nabla}^2\mathbf{v}$ is treated nearly fully implicitly in this region thus removing in most of the cases the related critical time step as experienced in Fröhlich[17]. The appearance of the mixed derivatives of the velocity is a particular feature of the LM equations with respect to constant density flow or Boussinesq approximation. Numerical experience shows that stabilizing with the second order derivatives is generally sufficient to allow explicit discretization of this part of the friction term. Its inclusion in the stabilizing term s would not be complicated, but the advantage of the expression (26) is

that u and v uncouple in s, which is thus represented by two $(N_C+1)\times(N_C+1)$ matrices instead of one $2(N_C+1)\times 2(N_C+1)$ matrix for each Fourier mode. Note that with no-slip conditions on the velocity (as well as slip conditions) the mixed derivatives vanish on the boundaries so that they contribute to the friction term only above a certain distance. Furthermore, as will be shown later, the mixed derivatives can be included in the pressure gradient term in the case of constant viscosity and have thus to be taken into account when considering variable fluid properties. This choice of s is well adapted to diffusion dominated situations. In cases where the non-linear convective term overwelms diffusion the operator s should also contain whenever possible a contribution $(\mathbf{v}_{app}.\nabla)\mathbf{v}$ with $\mathbf{v}_{app}$ being an approximation to v as suggested by Malik *et al.*[6] for a channel flow problem.

Note that in the forcing term (13) the hydrostatic density distribution can be substracted so that one actually considers a pressure p which represents the perturbation of the hydrostatic pressure. For the practical implementation of (24)–(25) a delta-formulation has been employed that uses $\mathbf{v}^{n+1}-\mathbf{v}^n$ and $p^{n+1}-p^n$ as unknowns at each time step. It is often applied to reduce the influence of round off errors and introduces no significant modification here due to the linearity of the implicit part of the scheme. For the sake of conciseness this point will not be apparent in the sequel.

Although focusing at present on the lhs of (24), (25), we would like to indicate that while calculating p_0^{n+1} in (25) from θ^{n+1} by (15), the use of (18) applied on the staggered Gauss points is recommended to determine $\mathrm{d}_t p_0^{n+1}$. This guarantees compatibility in the continuity equation (25) on a numerical level[15].

We now describe the procedure for solving (24), (25) at each time step using a symbolic notation in a spatially continuous sense, but we suppose that the corresponding discrete equivalents take into account the different types of collocation points and the boundary conditions as well. Defining the operator $\mathscr{L}$ by:

$$\mathscr{L}(\mathbf{v})=\sigma\rho_s\mathbf{v}+\mathbf{s}(\mathbf{v}) \tag{28}$$

the set of equations to be solved at each time step reads:

$$\mathscr{L}(\mathbf{v}^{n+1})+\nabla p^{n+1}=\mathbf{f}_v \tag{29}$$

$$\nabla.\mathbf{v}^{n+1}=g \tag{30}$$

where $\mathbf{f}_v$ and g replace the rhs of (24) and (25), respectively. With the definition (26) for s, $\mathscr{L}$ is represented after spatial discretization by a set of matrices, one for each Fourier mode, which are constant in time and can therefore be inverted once and for all. An equation for the pressure alone can be deduced solving (29) for $\mathbf{v}^{n+1}$ and inserting into (30):

$$\nabla.\mathscr{L}^{-1}(\nabla p^{n+1})=\nabla.\mathscr{L}^{-1}(\mathbf{f}_v)-g \tag{31}$$

The operator

$$\mathscr{A}=\nabla.\mathscr{L}^{-1}\nabla \tag{32}$$

once again corresponds to a set of 'one-dimensional' matrices in Fourier space. They are inverted at the beginning of the calculation and lead to the following direct algorithm for the determination of $\mathbf{v}^{n+1}$ and p^{n+1} including only matrix products:

$$\mathbf{v}_f=\mathscr{L}^{-1}(\mathbf{f}_v) \tag{33}$$

$$p^{n+1}=\mathscr{A}^{-1}(\nabla.\mathbf{v}_f-g) \tag{34}$$

$$\mathbf{v}_p=-\mathscr{L}^{-1}(\nabla p^{n+1}) \tag{35}$$

$$\mathbf{v}^{n+1}=\mathbf{v}_f+\mathbf{v}_p \tag{36}$$

For the inverse $\mathscr{A}^{-1}$ to exist, a supplementary condition is added by fixing the value of the pressure at one collocation point.

An important point related to the scheme (24), (25) is the choice of an initial state for the pressure. It is commonly known that an initial value problem for the Navier–Stokes equations for an incompressible fluid (and the present set of equations is of this type) is well posed without the initial pressure to be specified as to be found e.g. in Ladyzhenskaya[18]. When nevertheless using such a state, Heywood and Rannacher[19] indicate that it has to be determined from the initial velocity by solving a Poisson equation with Neumann boundary conditions. However, the regularity of the solution at subsequent time steps is strongly dependent on the way the initial velocity is defined[20]. Due to these difficulties, numerical algorithms requiring an initial state for the pressure should be avoided. In the present case, however, this problem is satisfactorily solved by starting from a true solution of the equations, either the hydrostatic state, or the solution of a different Rayleigh number which guarantees the initial pressure to be appropriately chosen.

When using this direct algorithm, it turns out that it is applicable only for very small perturbations of the static density distribution, more precisely if $|1-\rho_s/\rho|$ is smaller than about 0.3. If this factor is larger, the above direct scheme is unconditionally unstable.

We discuss this question of stability using the following Stokes problem as a model:

$$\sigma \mathbf{v}^{n+1} - \nabla^2 \mathbf{v}^{n+1} + \nabla p^{n+1} = -\{b\nabla p\}^{n,n-1} + \sigma\left(\frac{4}{3}\mathbf{v}^n - \frac{1}{3}\mathbf{v}^{n-1}\right) \tag{37}$$

$$\nabla . \mathbf{v}^{n+1} = 0 \tag{38}$$

with Fourier space discretization in both spatial directions. The scalar b here stands for the factor $(1-\rho_s/\rho)$ and is assumed to be positive. Let $\hat{\phi}_{km}$ be the Fourier coefficient of the function $\phi(x,y)$ and $\hat{U}_{km}^{n+1} = (\hat{u}_{km}^{n+1}, \hat{v}_{km}^{n+1}, \hat{p}_{km}^{n+1}, \hat{u}_{km}^{n}, \hat{v}_{km}^{n}, \hat{p}_{km}^{n})^t$. It can be shown that the amplification matrix E with:

$$\hat{U}_{km}^{n+1} = E\hat{U}_{km}^{n} \tag{39}$$

has the eigenvalues

$$\begin{aligned}
\lambda_1 &= -b + (b^2+b)^{1/2} \\
\lambda_2 &= -b - (b^2+b)^{1/2} \\
\lambda_3 &= \frac{2\sigma}{3\kappa}\left(1 + \left(1 - \frac{3\kappa}{4\sigma}\right)^{1/2}\right) \\
\lambda_4 &= \frac{2\sigma}{3\kappa}\left(1 - \left(1 - \frac{3\kappa}{4\sigma}\right)^{1/2}\right) \\
\lambda_5 &= 0 \\
\lambda_6 &= 0
\end{aligned} \tag{40}$$

with $\kappa = \sigma + k^2 + m^2$ for the wave numbers k and m in x- and y-direction, respectively. Note, that λ_1 and λ_2 are independent of $\sigma = 3/(2\Delta t)$, i.e. of the time step. A necessary condition for numerical stability is bounded spectral radius, i.e. $|\lambda| \leqslant 1$ for all eigenvalues of E. This condition is violated if $b > 1/3$, since in this case $|\lambda_2|$ becomes greater than unity and leads to unconditional numerical instability of the time scheme, an observation that corresponds nicely to the executed numerical calculations, even if the Fourier–Chebyshev space discretization is different from the Fourier–Fourier discretization for the model problem (37), (38). Since obviously the eigenvalues λ_1 and λ_2 are directly related to the pressure terms in (37), one may try to obtain them from exclusively considering the pressure terms in the scheme. A simple calculation shows that this is indeed the case. Note that if $0 < b < 1/3$ the above scheme is unconditionally stable.

A Crank–Nicolson scheme for the pressure in the constant density or LM equations leads to time schemes similar to (37), (38) in that the pressure enters on the level n. For the above Stokes problem approximated by Fourier series, the Crank–Nicolson scheme is exempt of numerical instability. For a channel flow problem using Fourier–Chebyshev approximation Zang and

Hussaini[21] report difficulties with such a scheme concerning the precision of the calculated pressure. They have been removed by implicit discretization of this quantity. Consequently, the use of a fully implicit scheme for the pressure is highly recommended, disregarding the additional problem of defining an initial value for this quantity.

The stability limit encountered for the use of the direct scheme (24), (25) restricts its application considerably. In the case of natural convection, the regime of extremely low Rayleigh numbers or very small departures from the Boussinesq case (characterized by a small value of ε) may be treated with this algorithm. Such conditions, however, are often far from the ones of interest. For these cases, the time scheme can be modified discretizing the pressure by an implicit and no longer by a semi-implicit scheme. We thus obtain the iterative algorithm described in Fröhlich and Peyret[15] which is briefly recalled here for completeness. The scheme reads:

$$\sigma\rho_s\mathbf{v}^{n+1}+\mathbf{s}(\mathbf{v})^{n+1}+\frac{\rho_s}{\rho}\nabla p^{n+1}=\mathbf{f}_g+\left\{-\rho_s(\mathbf{v}.\nabla)\mathbf{v}+Pr\frac{\rho_s}{\rho}\nabla.\tau\right\}^{n,n-1}+$$

$$\{\mathbf{s}(\mathbf{v})\}^{n,n-1}+\sigma\rho_s\left(\frac{4}{3}\mathbf{v}^n-\frac{1}{3}\mathbf{v}^{n-1}\right) \quad (41)$$

$$\nabla.\mathbf{v}^{n+1}=\frac{1}{p_0}\left(-\frac{1}{\gamma}\mathrm{d}_t p_0+\nabla^2\theta\right) \quad (42)$$

with s from (26). In the above way we deduce an equation for the pressure that reads:

$$\nabla.\mathscr{L}^{-1}\left(\frac{\rho_s}{\rho}\nabla p\right)=\nabla.\mathscr{L}^{-1}\left(\frac{\rho_s}{\rho}\mathbf{f}_v\right)-g \quad (43)$$

permitting to apply the algorithm (33)–(36) in a similar way. The pressure operator is now defined by:

$$\mathscr{A}^*(p)=\nabla.\mathscr{L}^{-1}\left(\frac{\rho_s}{\rho}\nabla p\right) \quad (44)$$

and contains the factor ρ_s/ρ which inhibits the uncoupling of modes and furthermore varies in time. This forbids the explicit use of $\mathscr{A}^{*-1}$, and therefore in the step analogous to (34) the pressure is now calculated iteratively. The procedure applied in Fröhlich and Peyret[15] is based on preconditioned conjugate residual iterations, detailed in Canuto *et al.*[1], using the spectral approximation of $\mathscr{A}$ as a preconditioner. Due to the fact that $\rho_s/\rho=0(1)$ this operator $\mathscr{A}$ is close to $\mathscr{A}^*$ and its use leads to very satisfactory convergence rates. Observe that in the present context, the time scheme is always chosen so that $\mathscr{L}$ can be inverted once and for all and applied with reasonable effort. Other temporal or spatial discretizations may impose an iterative solution of this equation by a true Uzawa algorithm.

Finally, concerning other numerical solutions of the LM equations in velocity–pressure formulation, we note the contribution of Caruso *et al.*[22] who have presented a finite element procedure based on Uzawa iterations for the solution of the steady state equations with a modified friction term and the method proposed by Le Quéré *et al.*[23] based on a finite difference technique.

A PSEUDO-UNSTEADY METHOD

As discussed in the previous section, the presence of the factor ρ_s/ρ in front of the pressure gradient in (41) requires the application of an iterative procedure. Recognizing that the origin of this factor is the time derivative $\rho\partial_t\mathbf{v}$ it is possible to avoid the need of an iterative procedure either by considering the derivative of the momentum $\partial_t\rho\mathbf{v}$ as it will be described later, or by modifying this term. In fact, if only the steady state solution is of interest, the time-derivative in the equations just serves to construct an iterative procedure in order to reach

this steady state. Without altering the final solution, this term may therefore be modified in order to set up a different algorithm permitting to obtain the final solution with less computational effort[24]. The true physical transition from the initial condition to the steady solution, of course, is no longer followed. In the present case, the simplest way is to just replace $\rho\partial_t \mathbf{v}$ by $\rho_s\partial_t \mathbf{v}$ or $\partial_t \mathbf{v}$ as in Peyret and Viviand[25] and Ouazzani and Peyret[26]. The resulting pseudo-unsteady algorithm reads:

$$\sigma \mathbf{v}^{n+1} + \mathbf{s}(\mathbf{v})^{n+1} + \nabla p^{n+1} = \mathbf{f}_g + \{-\rho(\nabla.\mathbf{v})\mathbf{v} + Pr\,\nabla.\tau\}^{n,n-1} + \{\mathbf{s}(\mathbf{v})\}^{n,n-1} + \sigma\left(\frac{4}{3}\mathbf{v}^n - \frac{1}{3}\mathbf{v}^{n-1}\right) \quad (45)$$

$$\nabla.\mathbf{v}^{n+1} = \frac{1}{p_0}\left(-\frac{1}{\gamma}\mathrm{d}_t p_0 + \nabla^2\theta\right) \quad (46)$$

with $\mathbf{s}$ given by (26) so that the part of the friction term discretized by the Adams–Bashforth scheme simply consists of the mixed derivatives.

The method has been used by Fröhlich and Peyret[11] for a study of isobaric combustion associated with a coordinate transform in y-direction to improve the numerical resolution of the flame. For the calculations reported in this reference, the use of the pseudo-unsteady method reduces the CPU time by a factor of about 4 with respect to the true unsteady iterative method. This concerns the complete set of equations wherein the calculation of temperature, species and density is kept unchanged.

A DIRECT METHOD BASED ON A MOMENTUM–PRESSURE FORMULATION

The pseudo-unsteady method just being described avoids iterations by modifying the physical equations that govern the transient solution. It is, however, possible to obtain the same feature while conserving the exact unsteady equations. This is achieved by using the conservative form of a momentum equation and equation (8) for the conservation of mass so that the dependent variables are the momentum $\mathbf{m}=\rho\mathbf{v}$ and the pressure p (Haldenwang, personal communication). The factor ρ_s/ρ in front of the pressure gradient now disappears, as ρ enters in the new variable. The hydrodynamical part of the LM equations then reads after temporal discretization:

$$\sigma \mathbf{m}^{n+1} + \mathbf{s}(\mathbf{m})^{n+1} + \nabla p^{n+1} = \mathbf{f}_g + \{-\nabla.(\mathbf{m}\mathbf{v}) + Pr\,\nabla.\tau\}^{n,n-1} + \{\mathbf{s}(\mathbf{m})\}^{n,n-1} + \sigma\left(\frac{4}{3}\mathbf{m}^n - \frac{1}{3}\mathbf{m}^{n-1}\right) \quad (47)$$

$$\nabla.\mathbf{m}^{n+1} = -\sigma\left(\rho^{n+1} - \frac{4}{3}\rho^n + \frac{1}{3}\rho^{n-1}\right) \quad (48)$$

so that the algorithm (33)–(36) can be applied with $\mathbf{v}^{n+1}$ being replaced by $\mathbf{m}^{n+1}$ and removing in $\mathscr{L}$ from (28) the factor ρ_s. The above scheme defines a true unsteady direct method for the hydrodynamical part of the LM equations that does not suffer either from semi-implicit pressure discretization and the related stability limit or from altering the physical equations.

In what concerns similar methods, we note that for the steady state LM equations, Caruso *et al.*[22] have developed a 2D finite element Uzawa algorithm based on a momentum–pressure formulation with a modified friction term and applied it to the calculation of free convection. Chenoweth and Paolucci[13] have devised a finite difference method for natural convection in square cavities that also employs the momentum as unknown. Finally, for the study of the Darrieus–Landau instability in isobaric combustion, Denet[27] has used the momentum as dependent variable but associated to an altered friction term $\rho^{-1}\nabla^2\mathbf{m}$ instead of $\nabla^2\mathbf{v}$. Moreover, the time derivatives in the temperature and species equation are modified in the same way as described in the previous section. The resulting equations are approximated with a Fourier/finite difference method.

The success of all the time schemes presented so far depends on the appropriate choice of the

stabilizing term s. If this term is not sufficient to get rid of the diffusive time step limitation, it would perhaps be better to use an explicit method (s=0) for reasons of simplicity. As previously, the operator s has to be linear with at most a y-depending factor if one wants to avoid iterations. Numerical experiments conducted with (47), (48) show that the use of:

$$\mathbf{s}(\mathbf{m}) = -Pr\,\tilde{\nabla}^2\mathbf{m} \tag{49}$$

similar to (26), does not stabilize the scheme in the same way as the iterative method based on (41), (42). The choice of

$$\mathbf{s}(\mathbf{m}) = -Pr\,\tilde{\nabla}^2\left(\frac{1}{\rho_s}\mathbf{m}\right) = -Pr\,\tilde{\nabla}^2\left(\frac{1}{\rho_s}\rho\mathbf{v}\right) \tag{50}$$

however, was successful. This is comprehensible, since, as already mentioned above, $\rho_s/\rho \approx 1$ near the boundaries, so that the friction term is treated in this region almost entirely by the second order Euler-backward scheme.

Let us now make a remark on the use of the continuity equation (48). The boundary conditions for a closed domain impose similarly to (17)

$$\int_{\Omega_A} \nabla.\mathbf{m}\,\mathrm{d}\Omega_A = 0 \tag{51}$$

For the continuous equations, the integral of $\partial_t\rho$ over the computational domain vanishes also due to the verification of the continuity equation. Consequently, a solution to the discrete equations does only exist up to a precision characterized by the difference existing between the discrete integral on either side of the continuity equation. Since in a time marching procedure, the error introduced in this way may sum up and lead to a breakdown of the calculation, the algorithm has to be constructed so as to safely avoid this phenomenon. For the velocity–pressure methods this should be used as a condition to determine $\mathrm{d}_t p_0$ in (25).

In the following we discuss this point in detail for the direct momentum–pressure algorithm (47) and (48) and define for convenience

$$R = \rho^{n+1} - \frac{4}{3}\rho^n + \frac{1}{3}\rho^{n-1} \tag{52}$$

appearing in the rhs of (48). Since the integral of R is influenced by truncation errors in space and time, we first concentrate on the time scheme assuming for the moment no discretization in space. Recall that the density at the level $n+1$ is determined from the equation of state (11) using θ^{n+1} and p_0^{n+1} given by (15). Consequently, at each time step the integral of ρ over the domain is equal to the total mass M_A, so that the integral of R vanishes exactly. It is obvious, that this would not be the case if p_0^{n+1} was calculated differently. For example the determination of this quantity by the Adams–Bashforth extrapolation (7) would lead to an error $O(\Delta t^2)$ in the integral of R.

Let us now consider the error introduced by the spatial discretization. Owing to the use of a staggered mesh of Gauss points to define the unknown pressure, the continuity equation is enforced on these points, whereas all the other equations are enforced on the Gauss–Lobatto points. With the spectral approximation the unknown temperature, velocity and density defined by their values at the Gauss–Lobatto points are represented in y-direction by polynomials of degree N_C. The projection from the Gauss–Lobatto to the Gauss points eliminates the highest order term in the Chebyshev expansion, since $T_{N_C}(\eta_j)=0$ due to (21) and (23). Consequently, integrating a dependent variable such as ρ with the help of the N_C+1 Gauss–Lobatto points gives the correct integral of this function over the domain. Integrating with the N_C Gauss points, the highest Chebyshev mode is disregarded.

It is now possible to show that the integral in (51) is (for the present geometry) not affected by the projection on the Gauss points, i.e. by the spatial discretization, if derivatives are evaluated

before projection (discrete differentiation and projection do not commute). The first part of the divergence term gives $\int_{\Omega_A} \partial_x m_1 \, d\Omega_A = 0$ due to the periodicity in x of $m_1 = \rho u$. Because of the differentiation with respect to y, the second part of the divergence term is a polynomial in y of degree $N_C - 1$, so that it's evaluation on the Gauss points does not remove the contribution from T_{N_C}. Consequently, the property (51) is true also on the spatially discrete level. On the other hand, integrating R over the domain gives different values when using the Gauss or the Gauss–Lobatto points, since ρ generally is a polynomial in y of degree N_C. However, when using the Gauss points for integration in (15), the integral of R on the staggered mesh vanishes. The total mass may now differ from M_A by an amount corresponding to the spatial truncation error, but compatibility in the discrete system is always verified up to machine precision.

We conclude this section by indicating a simplification that can be applied to all the presented methods but which has only been implemented for the momentum–pressure formulation and the pseudo-unsteady algorithm of the previous section. It is based on the particularity of the LM equation that the pressure p only appears in the momentum equation without thermodynamic relevance. Hence, it is possible to modify this pressure term by any expression in the form of a gradient as indicated by Denet[27]. For constant viscosity, where

$$\nabla . \tau = \nabla^2 \mathbf{v} + \frac{1}{3} \nabla (\nabla . \mathbf{v}) \tag{53}$$

this may be used to suppress the mixed derivatives in the friction term by defining a modified pressure:

$$\tilde{p} = p - Pr \frac{1}{3} (\nabla . \mathbf{v}) \tag{54}$$

so that the momentum equation (47) now reads:

$$\sigma \mathbf{m}^{n+1} + \mathbf{s}(\mathbf{m})^{n+1} + \nabla \tilde{p}^{n+1} = \mathbf{f}_g + \{ -\nabla . (\mathbf{mv}) + Pr \nabla^2 \mathbf{v} \}^{n,n-1} + \{ \mathbf{s}(\mathbf{m}) \}^{n,n-1} + \sigma \left(\frac{4}{3} \mathbf{m}^n - \frac{1}{3} \mathbf{m}^{n-1} \right) \tag{55}$$

With the stabilizing term

$$\mathbf{s}(\mathbf{m}) = -Pr \nabla^2 \left(\frac{1}{\rho_s} \mathbf{m} \right) \tag{56}$$

the explicit part of the friction term becomes simply

$$Pr \nabla . \tau - \mathbf{s}(\mathbf{m}) = Pr \nabla^2 \left(1 - \frac{\rho}{\rho_s} \right) \mathbf{v} \tag{57}$$

Note that the introduction of the modified pressure $\tilde{p}$ defined in (54) is possible, since the considered physical problem does not include boundary conditions for the pressure. Moreover, the prescription of $\tilde{p}$ on the boundaries for algorithmical reasons has been avoided by using the staggered mesh, so that the implementation is straightforward. Concerning the stability of the resulting algorithm, we note that the explicit term (57) behaves like the one in (27) with the mixed derivatives having disappeared here. As supported by the numerical results reported in the following section, the critical time step remains unchanged. An advantage of the formulation is the simplification of the equations to be solved (in the case of constant viscosity) which leads to a reduced number of operations. With this option coded for the momentum–pressure algorithm, we observed a gain of 7% in the overall CPU time when solving the complete Rayleigh–Bénard problem. It can finally be remarked that such a modification leads to a fully implicit treatment of the diffusion term when being applied to the pseudo-unsteady method.

NUMERICAL PROPERTIES OF THE ALGORITHMS

Having presented different schemes for the solution of the hydrodynamical part of the LM equations we shall now discuss some numerical properties related to these algorithms. All of them have been applied to the Rayleigh–Bénard problem defined earlier which serves as test case. We concentrate here on the numerical aspect and do not discuss the solution from a physical point of view. The reader is referred to Fröhlich[28] and Fröhlich *et al.*[16] where this is done in detail. It is, however, convenient to give an impression of a typical calculation for this problem. To this aim we begin with the temporal evolution of a sample calculation making some remarks on computation time. It should be noticed that the eventual physical solution of the problem is stationary up to high Rayleigh numbers. However, the analysis of the properties of the algorithms during the transient phase gives sufficient information on their behaviour for fully unsteady calculations. In the second part of this section we investigate the numerical stability of the different methods before treating the pseudo-unsteady method in a following part. Finally, some numerical experiences on the discrete compatibility condition for the momentum–pressure formulation are reported.

The calculations that are reported have been carried out in the following way. The aspect ratio of the domain is set to $A=2.017$ which approximatively corresponds to the critical wavelength of the Rayleigh–Bénard instability in the Boussinesq case, $Pr=0.71$ and $\gamma=1.4$ are values corresponding to air, and $\varepsilon=\Delta T/(2T^*)$ in (13) and (20) is generally chosen equal to $\varepsilon=0.5$. This value represents a large characteristic temperature difference related to a strong departure from the Boussinesq approximation. Note that by definition $0\leqslant\varepsilon\leqslant 1$ when choosing the mean temperature as reference. As the critical wavelength varies only slightly with ε, differing in the present case less than 1% from the Boussinesq value[16], the choice of the above aspect ratio is justified. The initial condition is the static state with a random perturbation in the temperature field of amplitude 10^{-5}. Then, for a given Rayleigh number, the different schemes described in the previous sections are applied to advance the solution in time (the first step being calculated by a first order scheme). The calculation is continued up to the steady state, characterized by $\max\{|\mathbf{v}^{n+1}-\mathbf{v}^n|/\Delta t\}\leqslant 10^{-10}$ on the collocation points. As indicated, the temperature equation is always solved by the same semi-implicit scheme, similar to (24).

Computation time

Let us first discuss the computation time for the direct momentum–pressure algorithm in comparison to the iterative velocity–pressure method. The evolution of the Nusselt number (ratio between the total heat flux and the heat flux of the purely conductive state) in *Figure 1* has been obtained with $Ra=6000$, $\Delta t=5\times 10^{-3}$, and $N_F=32$, $N_C=24$. It is characterized by a first phase of exponential growth of the initial perturbation leading to a transient phase with relatively high temporal derivatives which is followed by the final stabilization of the steady solution. In the present case the integration in time has been pursued sufficiently far so that all time-derivatives have decreased to a level corresponding to the high spatial accuracy.

In *Figure 1* the solution has first been obtained with the iterative algorithm (41), (42). The corresponding curve of the cumulative CPU time (in seconds, on CRAY 2) exhibits constant slope when temporal changes are very small due to the fact that only one or two iterations are executed at each time step. In the transient phase, the number of internal iterations increases so that the method becomes more time consuming. Note that this effect will be amplified for larger time steps and stronger perturbations of the density with respect to the reference state (here the hydrostatic state). The computation time of the direct method (47) and (48) is also reported in *Figure 1*. It exhibits a constant slope which is smaller than the limiting slope of the iterative method. This is due to the higher complexity of the iterative technique (necessity of supplementary calculations: residual, ...) and the difficulty of adjusting the convergence parameters for the inner iterations. Note that the time saving in the transient phase is important. In the present example, this latter, however, constitutes only a small part of the whole run

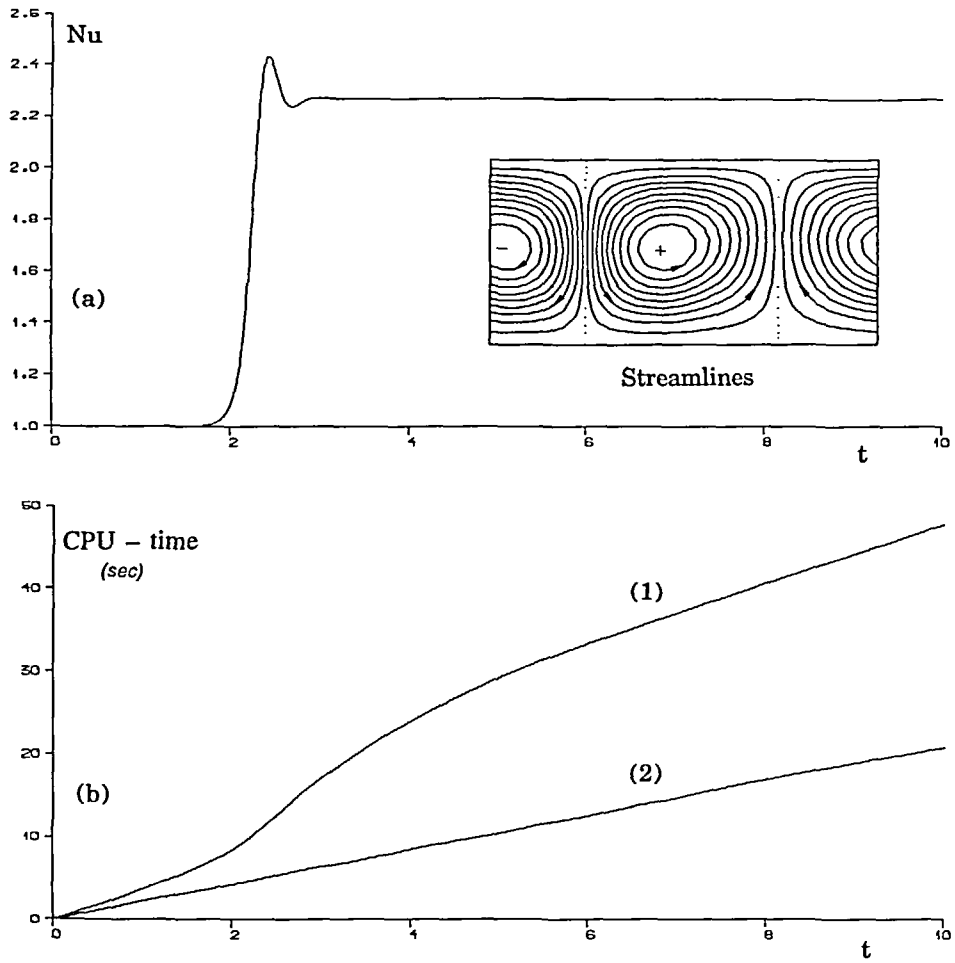

Figure 1 Temporal evolution for the calculation of the Rayleigh-Bénard flow field for $Ra=6000$, $\varepsilon=0.5$, $N_F=32$, $N_C=24$, $\Delta t=5\times10^{-3}$. The initial condition is the hydrostatic state with the temperature randomly perturbed by an amplitude of 10^{-5}. (a) Evolution of the Nusselt number on the bottom of the domain. The steady state solution is characterized by the streamlines based on the momentum. (b) Cumulative CPU time (CRAY 2) *versus* physical time for the iterative **v**–p method (1) and the direct **m**–p method (2). The calculated transient solution is the same in both cases

(roughly 1/10), so that a factor of about 2 is obtained in the end. For true unsteady solutions as they appear, e.g. in oscillatory convection, the acceleration will lead to a factor much higher than this, especially when many iterations would be necessary due to the use of relatively large time steps or strong density perturbations. Apart from considerations of CPU time, a direct method is often advocated because of greater simplicity and the fact that it avoids the specification and optimization of iteration parameters which is often a delicate task. Let us finally state that the other direct methods have been tested under the same conditions (see below). They need similar computation time as the momentum pressure method in *Figure 1*.

Numerical stability of the truly unsteady methods

We report first some results concerning the stability of the direct velocity–pressure algorithm (24) and (25). If the density ρ deviates too much from its reference state ρ_s, it is subject

to unconditional numerical instability due to the partly explicit temporal discretization of the pressure as indicated. This has been investigated above for a simplified model, and we report here in *Table 1* the corresponding results for the complete Rayleigh–Bénard flow.

We continue by considering the choice of the stabilizing term s in the momentum–pressure algorithm (47), (48) and compare its behaviour to the iterative velocity–pressure method (41), (42). For $N_F=16$ and $N_C=12$ we obtained with $Ra=2000$ the estimations of the critical time step that are reported in *Table 2*. Note that this Rayleigh number is only very slightly above the critical value which is for this case[16] $Ra_{crit}=1711$ so that the velocity remains very low as indicated in *Table 1*.

The results show (last line of *Table 2*) that in this case an appropriate choice of s can suppress the time step limit generated by the diffusive term, leading here to an unconditionally stable algorithm. Indeed, if the time step is enlarged further, e.g. to $\Delta t=1$ which is beyond any physically meaningful value, we observe that the numerical solution exhibits temporal oscillations due to the truncation error in time but does not diverge. If s is not chosen in an optimal way, this may lead to a less stable scheme as indicated in the third line of this Table. Nevertheless, a gain of roughly a factor 5 in the critical time step is to be observed when comparing to the explicit discretization of the diffusion term (s=0). With the present spatial resolution ($N_F=16$, $N_C=12$), the critical time step is about 10^{-3} for this latter case. The iterative v–p scheme with s from (26) also leads to an unconditionally stable scheme which is in accordance with the discussion in the corresponding section.

Table 3 presents results similar to those of *Table 2*, except that the Rayleigh number has been set to a higher value ($Ra=6000$) so that the convective flow becomes stronger, illustrated by the velocity values reported in *Table 1*. It can be observed that with the applied spatial resolution the explicit discretization of the diffusion term induces a limiting time step of about 5×10^{-5}, whereas one encounters only the restriction due to the convective term when chosing an

Table 1 Stability of the direct velocity–pressure method with $N_F=32$, $N_C=24$. In the last two lines the method is unstable and in this case the values for the velocity and the density factor have been obtained with a different method (momentum–pressure)

Ra	ε	$\lvert v\rvert_{max}$	$\left\lvert 1-\frac{\rho_s}{\rho}\right\rvert_{max}$	Δt
2000	0.5	3.93	0.232	5×10^{-2}
6000	0.01	10.28	0.00836	1×10^{-2}
6000	0.1	10.38	0.0858	5×10^{-3}
6000	0.2	10.52	0.177	5×10^{-3}
6000	0.3	10.70	0.275	5×10^{-3}
6000	0.4	10.90	0.384	unstable for 2×10^{-3}
6000	0.5	11.70	0.508	unstable for 5×10^{-6}

Table 2 Stability of the calculation for different choices of the stabilizing term s with $N_F=16$, $N_C=12$, $\varepsilon=0.5$, and $Ra=2000$, in parentheses the time step for divergence. The first line concerns the iterative algorithm (41), (42), the remainder is related to (47), (48)

Method	s(m)	Δt_{stable}
v–p iterative	$-Pr\,\tilde{\nabla}^2\mathbf{v}$	0.1
m–p	0	1×10^{-3} (2×10^{-3})
m–p	$-Pr\,\tilde{\nabla}^2\mathbf{m}$	5×10^{-3} (1×10^{-2})
m–p	$-Pr\,\tilde{\nabla}^2(1/\rho_s\mathbf{m})$	0.1

Table 3 Stability of the calculation for different choices of the stabilizing term s with $N_F=32$, $N_C=24$, $\varepsilon=0.5$, and $Ra=6000$, in parentheses the time step for divergence. The first line concerns the iterative algorithm (24) and (25), the remainder is related to (47) and (48) and for the last line the simplified friction term has been used

Method	s(m)	Δt_{stable}
v–p iterative	$-Pr\,\tilde{\nabla}^2\mathbf{v}$	1×10^{-3} (2×10^{-3})
m–p	0	5×10^{-5} (1×10^{-4})
m–p	$-Pr\,\tilde{\nabla}^2\mathbf{m}$	2×10^{-4} (4×10^{-4})
m–p	$-Pr\,\tilde{\nabla}^2(1/\rho_s\mathbf{m})$	8×10^{-3} (1×10^{-2})
m–$\tilde{p}$	$-Pr\,\tilde{\nabla}^2(1/\rho_s\mathbf{m})$	8×10^{-3} (1×10^{-2})

Table 4 Stability of the pseudo-unsteady method with $\varepsilon=0.5$ and $N_F=16$, $N_C=12$ in the first line and $N_F=32$, $N_C=24$ in the second line

Ra	s(m)	Δt_{stable}
2000	$-Pr\nabla^2\mathbf{v}$	0.1
6000	$-Pr\nabla^2\mathbf{v}$	5×10^{-3} (8×10^{-3})

appropriate expression for **s**. We also used the simplified friction term that avoids the mixed derivatives without modifying the physical solution as discussed in the previous section. The corresponding result in the last line of *Table 3* shows that this modification does not improve the numerical stability, at least in this case.

The pseudo-unsteady method

The pseudo-unsteady method has been implemented with the modification of the pressure so that the treatment of the diffusion term is fully implicit. *Table 4* gives values of the critical time step that is due to the non-linear convective term here. The Table shows that the stability is similar to the one experienced with the iterative **v**–p and the **m**–p (or **m**–$\tilde{p}$) method using the correct stabilizing term. Incidentally, this proves that with the above semi-implicit discretization the diffusive time step limitation has been fully eliminated, indeed. It is important to observe that the physical time needed to obtain the steady state is roughly the same as the one corresponding to the truly unsteady methods. Only the transient phase is somewhat different as to be seen on *Figure 2*. Hence, the number of time steps to be executed is similar. A slight decrease of the operation count in each time step leads to a small gain in the overall computation time (less than 10% in the present case).

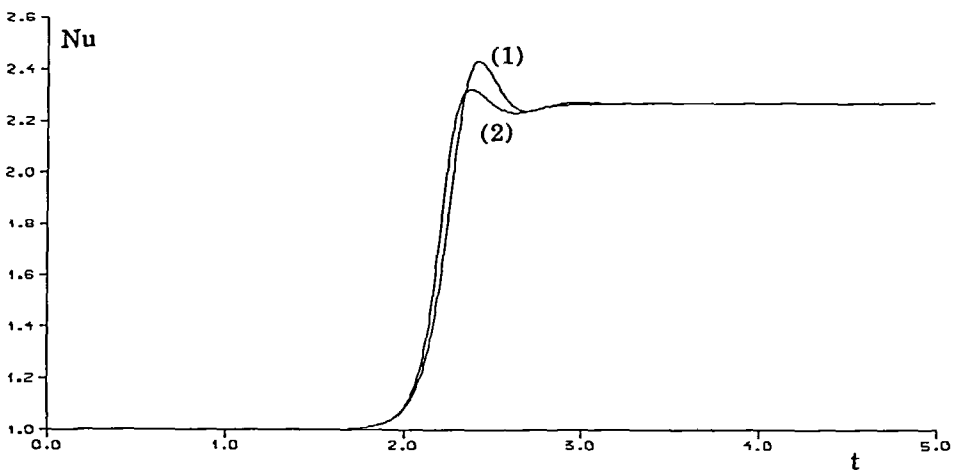

Figure 2 Temporal evolution of the Nusselt number on the bottom of the domain for the unsteady momentum–pressure method (1) and the pseudo-unsteady algorithm (2) with $Ra=6000$, $\varepsilon=0.5$, $N_F=32$, $N_C=24$. The transient solution obtained from the latter is modified with respect to the true evolution but leads to the same steady state

Numerical compatibility for the momentum–pressure formulation

In order to investigate the importance of the spatial truncation error for the discrete compatibility, we repeated the calculation with the direct momentum–pressure method using very low spatial resolution, $N_F=12$, $N_C=10$, which leads to a relatively high spatial truncation error of about 10^{-4} for the variables of order unity. In a first case, the integral in (15) is calculated on the Gauss–Lobatto points (22) evaluating at each time step the integral of the rhs in (31) for control. Its maximum absolute value reaches 7×10^{-4} during the transient phase and decreases below 10^{-10} in the final steady state (recall that the algorithms are implemented in a delta formulation for all variables). The solution does not change more than 10^{-8} when doubling the calculated laps of time, thus showing that no accumulation of error occurs in the steady state of this case. The error during the transient phase may, however, be trouble-causing in other applications, in particular for unsteady solutions. We therefore executed a second run evaluating now p_0^{n+1} with (15) applied on the Gauss points (23). The resulting integral of the rhs in (31) remained below 10^{-10} during the whole run. The final steady state solution was the same when taking into account the relatively coarse spatial resolution. This last method is to be preferred as it safely avoids the proliferation of round off errors, in particular for unsteady calculations. It should be indicated that large truncation errors are realistic when dealing with very stiff problems such as combustion phenomena, for example. When introducing a supplementary non-linear coordinate transformation in y-direction to improve the spatial resolution as in Fröhlich and Peyret[11], care has to be taken due to the fact that transformation and projection on a staggered mesh do not commute. It should, however, be stressed that the main point to preserve high accuracy is the calculation of p_0^{n+1} based on an unchanged reference quatity, here M_A.

CONCLUSIONS

From the present study of various spectral algorithms for the solution of the low Mach number equations, it can be concluded that the most efficient method is the direct method based on the following points: (1) momentum–pressure formulation (47), (48); (2) stabilizing term s from (50); (3) modified pressure $\tilde{p}$ following (54); (4) p_0 calculated by conservation of the total mass (15). This leads to an algorithm for the solution of the truly unsteady equations with good numerical stability, requiring only matrix products at each time step.

A first extension of the numerical schemes presented above is to take into account variable fluid properties such as temperature dependent viscosity μ and heat conductivity λ. We have experienced this for the present set up with the iterative velocity–pressure system (24)–(25), replacing τ from (2) by $\tau_\mu=\mu\tau$, where μ obeys the Sutherland law[28]. The behaviour of the scheme is essentially the same as described above provided the stabilizing term s takes into account the variation of the viscosity through a factor $\mu_s(y)$ (being here the viscosity in the hydrostatic state) similarly to $\rho_s(y)$ (see also Malik *et al.*[6]).

In a general way, the term s has to be adapted to the problem under consideration. It should lead to an implicit part of the scheme easy to invert inducing at the same time sufficient numerical stability. The choice of s is strongly related to the choice of ρ_s. In the present case of the Rayleigh–Bénard problem the hydrostatic solution furnishes a suitable expression, but for other situations its determination is perhaps less immediate. It can issue from physical considerations but may also be the result of some averaging procedure done once before starting the temporal evolution or periodically during the time integration.

We now make some remarks concerning the construction of schemes for different problems by the choice of s indicating which kind of coefficients are supported in each case. First, the introduction of a coordinate transform in the non-periodic direction is straightforward if the transformation does not depend on x^{11}. If it is variable in this latter direction, it leads to a variable coefficient that can be approximated by its mean value with respect to x.

For two-dimensional geometry without periodicity, the development of the dependent variables

in double Chebyshev series complicates the solution procedure. The diagonalization technique commonly used in spectral methods can be employed if the stabilizing term s is such that the operator $\mathscr{L}$ is of the form:

$$\mathscr{L}(\mathbf{v}) = \sigma \mathbf{v} - c(t)[a(x)\partial_{xx} + b(y)\partial_{yy}]\mathbf{v} \tag{58}$$

If $c = \text{const}$, $\mathscr{L}$ can be diagonalized once and for all[29,30] and if $c \neq \text{const.}$, the diagonalization is performed for the operator in square brackets as done by Orszag and Kells[31].

A second way to conserve direct inversion of $\mathscr{L}$ is to introduce an approximate factorization technique (generalized ADI), Peyret and Taylor[31], based on:

$$\mathscr{L}(\mathbf{v}) = \left[I - \frac{c(t)}{\sigma} a(x)\partial_{xx}\right]\left[I - \frac{d(t)}{\sigma} b(x)\partial_{yy}\right]\mathbf{v} \tag{59}$$

This leads to one-dimensional problems in each spatial direction with the corresponding operators being inverted once and for all.

A third technique for direct solution is often overlooked, when dealing with relatively few degrees of freedom, say 20×20. In former times it has been shocking to think of a direct inversion of the 400×400 matrix resulting from spatial discretization. With the progress in computer technology (vectorial and parallel processing) the use of direct inversion of such a matrix has become possible, provided it can be done once and for all. It leads to a matrix product to be performed at each time step with reasonable computational effort. This takes away any restriction on the coefficients that appear in the implicit part of the algorithm, except that they have to be constant in time.

In what concerns three-dimensional geometries, the above remarks for the two-dimensional case carry over (except direct inversion of the full 3-D operator will generally not be reasonable), in particular when dealing with cylindrical geometries with a Fourier–Chebyshev–Chebyshev development. In this case, the time scheme has to be chosen so that different Fourier modes uncouple, thus reducing the problem to a series of two-dimensional problems.

In the present paper we focused on direct methods. Without entering in the controversy on iterative or direct methods, it is our opinion that for time marching procedures one should first try to construct a direct algorithm in view of efficiency and practical reasons. If it turns out to exhibit too restrictive numerical stability criterion, an iterative procedure can still be added. In the present case of moderate non-linearities, in particular related to higher order derivatives, the present schemes are based on the consideration of a linear term which approximates sufficiently well the non-linear expression. With an iterative procedure, such a restriction does not appear. An example has been treated in Fröhlich *et al.*[16], where the stabilizing term s is non-linear as it contains diffusive and convective terms. Note that the recourse to preconditioning based on finite difference methods might be advantageous in highly non-linear problems.

Let us finally state that the presented methodology can also be applied to a large class of problems. For example, as already indicated, the equations that govern low speed combustion or non-homogeneous flows of miscible fluids with a large difference in molecular weight are nearly the same as the LM equations and can be solved in the same manner.

ACKNOWLEDGEMENTS

The CRAY 2 computation time for the numerical calculations has been supported by the Centre de Calcul Vectoriel pour la Recherche. The first author (JF) is indebted to the Conseil Regional PACA for financial support.

REFERENCES

1 Canuto, C., Hussaini, M. Y., Quarteroni, A. and Zang, T. A. *Spectral Methods in Fluid Dynamics*, Springer-Verlag, New York (1988)

2 Gottlieb, D. and Orszag, S. A. *Numerical Analysis of Spectral Methods: Theory and Applications*, SIAM, Philidelphia (1977)

3 Deville, M., Haldenwang, P. and Labrosse, G. Comparison of the integration (finite difference and spectral) for the non linear Burgers equation, *Proc. 4th GAMM Conf. Num. Meth. Fluid Mech.*, Vieweg, Braunschweig, pp. 64–76 (1982)
4 Vanel, J. M., Peyret, R. and Bontoux, P. A pseudo-spectral solution of vorticity-stream function equations using the influence matrix technique, *Numerical Methods for Fluid Dynamics II* (Morton, K. W. and Baines, M. J., eds.), pp. 477–488, Clarendon Press, Oxford (1986).
5 Ouazzani, J., Peyret, R. and Zakaria, A. Stability of collocation–Chebychev schemes with application to the Navier–Stokes equations, *Proc. 6th GAMM Conf. Num. Meth. Fluid Mech.*, pp. 287–294, Vieweg, Braunschweig (1986)
6 Malik, M. R., Zang, T. A. and Hussaini, M. Y. A spectral method for the Navier–Stokes equations, *J. Comp. Phys.*, **61**, 64–88 (1985)
7 Gauthier, S. A spectral method for two-dimensional compressible convection, *J. Comp. Phys.*, **75**, 217–235 (1988)
8 Guillard, H., Malé, J. M. and Peyret, R. Adaptive spectral methods with applications to mixing layer computations, *J. Comp. Phys.* (in press)
9 Gerhold, T. An implicit spectral method for the Navier–Stokes equations, applied to free convection flows, *Diploma of Engineering*, RWTH Aachen (1990)
10 Fröhlich, J., Gerhold, T., Lacroix, J. M. and Peyret, R. Fully implicit methods for natural convection, *High Performance Computing II* (M. Durand and F. El Dabaghi, eds), pp. 585–596, North-Holland, Amsterdam (1991)
11 Fröhlich, J. and Peyret, R. A spectral algorithm for low Mach number combustion, *Comp. Meth. ACPL. Mech. Eng.* **90**, 631–642 (1991)
12 Paolucci, S. On the filtering of sound from the Navier–Stokes equations, *SAND82-8257*, Sandia National Laboratories, Livermore (1982)
13 Chenoweth, D. R. and Paolucci, S. Natural convection in an enclosed vertical air layer with large horizontal temperature differences, *J. Fluid Mech.*, **169**, 173–210 (1986)
14 Majda, A. Compressible fluid flow and systems of conservation laws in several space variables, *Appl. Math. Sci.*, Vol. 53, Springer-Verlag, New York (1984)
15 Fröhlich, J. and Peyret, R. Calculations of non-Boussinesq convection by a pseudospectral method, *Comp. Meth. Appl. Mech. Eng.*, **80**, 425–433 (1990)
16 Fröhlich, J., Laure, P. and Peyret, R. Large departures from Boussinesq approximation in the Rayleigh–Bénard problem, *Phys. Fluids (A)* (in press)
17 Fröhlich, J. Comparison of solution algorithms for the variable coefficient heat equation applying a spectral collocation method, *Prépublication No. 212*, Laboratoire de Mathématiques, Université de Nice (1988)
18 Ladyzhenskaya, O. A. *The Mathematical Theory of Viscous Incompressable Flow*, Gordon and Breach, New York (1963)
19 Heywood, J. C. and Rannacher, R. Finite element approximation of the nonstationary Navier–Stokes problem. I Regularity of the solutions and second-order error estimates for spatial discretization, *SIAM J. Num. Anal.*, **19**, 275–311 (1982)
20 Montigny-Ranou, F. Influence of compatibility conditions in numerical simulation of inhomogeneous incompressible flow, *Proc. 5th GAMM Conf. Num. Meth. Fluid Mech.*, pp. 234–242, Vieweg, Braunschweig (1984)
21 Zang, T. A. and Hussaini, M. Y. Numerical experiments on subcritical transition mecanisms, *AIAA-paper No. 85-0296* (1985)
22 Caruso, A., Guyon, C., Metivet, B. and Thomas, B. Navier–Stokes equations with varying density: finite element implementation algorithms, *Second World Congr. Comp. Mech., Stuttgart* (1990)
23 Le Quéré, P., Humphrey, J. A. C. and Sherman, F. S. Numerical calculation of thermally driven two-dimensional unsteady laminar flow in cavities of rectangular cross section, *Num. Heat Transf.*, **14**, 249–283 (1981)
24 Peyret, R. and Viviand, H. Pseudo-unsteady methods for inviscid or viscous flow computation, in *Recent Advances in Aerospace Sciences* (Casci, C., ed.), pp. 41–71, Plenum Press, New York (1985)
25 Peyret, R. and Viviand, H. Calcul d'un écoulement d'un fluide visqueux compressible autour d'un obstacle de forme parabolique, *Proc. 3rd Int. Conf. Num. Meth. Fluid Mech.*, pp. 222–229, Springer-Verlag, New York (1973)
26 Ouazzani, J. and Peyret, R. A pseudo-spectral solution of binary gas mixture flows, *Proc. 5th GAMM Conf. Num. Meth. Fluid Mech.*, pp. 275–282, Vieweg, Braunschweig (1984)
27 Denet, B. Simulations numériques d'instabilités de fronts de flamme, *Thèse*, Université de Provence (1988)
28 Fröhlich, J. Résolution numérique des équations de Navier–Stokes à faible nombre de Mach par méthode spectrale, *PhD Thèse*, Université de Nice—Sophia-Antipolis (1990)
29 Haidvogel, D. B. and Zang, T. A. The accurate solution of Poisson's equation by expansion in Chebyshev polynomials, *J. Comp. Phys.*, **30**, 167–180 (1979)
30 Haldenwang, P., Labrosse, G., Aboudi, S. and Deville, M. Chebyshev 3-D spectral and 2-D pseudospectral solvers for the Helmholtz equation, *J. Comp. Phys.*, **55**, 115–128 (1984)
31 Orszag, S. A. and Kells, L. C. Transition to turbulence in plane Poiseuille flow and plane Couette flow, *J. Fluid Mech.*, **96**, 159–205 (1980)
32 Peyret, R. and Taylor, R. D. *Computational Methods for Fluid Flow*, Springer-Verlag, New York (1983)