Fast iterative solvers for buoyancy driven flow problems ☆

Howard Elman [a,*], Milan Mihajlović [b], David Silvester [c]

[a] Computer Science and Institute for Advanced Computer Studies, University of Maryland, College Park, MD 20742, USA
[b] School of Computer Science, University of Manchester, Manchester M13 9PL, UK
[c] School of Mathematics, University of Manchester, Manchester M13 9PL, UK

ARTICLE INFO

ABSTRACT

We outline a new class of robust and efficient methods for solving the Navier–Stokes equations with a Boussinesq model for buoyancy driven flow. We describe a general solution strategy that has two basic building blocks: an implicit time integrator using a stabilized trapezoid rule with an explicit Adams–Bashforth method for error control, and a robust Krylov subspace solver for the spatially discretized system. We present numerical experiments illustrating the efficiency of the chosen preconditioning schemes with respect to the discretization parameters.

© 2011 Elsevier Inc. All rights reserved.

1. Introduction

Even the simplest advection–diffusion problems can exhibit multiple time scales, see Gresho et al. [10]. This means that robust variable step time integrators are a prerequisite if such problems are to be efficiently solved computationally. The effectiveness of a stabilized TR–AB2 time-stepping (trapezoid rule with a second-order explicit Adams–Bashforth method for error control) algorithm has been demonstrated in the context of scalar convection–diffusion problems, and the *isothermal* Navier–Stokes equations in our previous publications, [10,13]. In this paper we assess the performance of the integrator in combination with a state-of-the-art iterative solver for problems with *coupled* physics. Specifically, we consider the simplest possible Boussinesq model of buoyancy driven flow represented by the following system of PDEs:

$$\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} - \boldsymbol{v}\nabla^2\vec{u} + \nabla p = \vec{j}T \quad \text{in } \mathcal{W} \equiv \Omega \times (0, \tau], \tag{1.1}$$

$$\nabla \cdot \vec{u} = 0 \quad \text{in } \mathcal{W}, \tag{1.2}$$

$$\frac{\partial T}{\partial t} + \vec{u} \cdot \nabla T - v\nabla^2 T = 0 \quad \text{in } \mathcal{W}. \tag{1.3}$$

Our notation is standard: the flow domain is Ω, $\vec{u}$ is the fluid velocity, p is the pressure and T is the temperature of the fluid. The energy Eq. (1.3) essentially determines T and is coupled to the momentum equation through the "buoyancy force" term on the right-hand side of (1.1).

The system (1.1)–(1.3) arises as a limiting case of modelling the flow of a fluid forced by gravity (acting in the vertical direction $\vec{j}$) where the typical fluid velocity is much smaller than the local sound speed, and where only small temperature deviations from the mean temperature are allowed. Further details are given in Straughan [19, Section 3.2]. Note that there are two different viscosity parameters in our model: $\boldsymbol{v}$ in (1.1) and v in (1.3). The precise definition of these viscosity parameters depends on the nondimensionalization used. In our case, following the approach of Christon et al. [2], we define $\boldsymbol{v}$ and v in terms of a Rayleigh number Ra and a Prandtl number Pr with

$$\boldsymbol{v} = \sqrt{\frac{\text{Pr}}{\text{Ra}}}, \quad v = \frac{1}{\sqrt{\text{Pr} \cdot \text{Ra}}}.$$

The Prandtl number Pr is a property of the fluid, essentially the ratio of momentum diffusivity and thermal diffusivity: typical values are Pr ≈ 0.71 for air and Pr ≈ 0.015 for mercury (much smaller since it is a much more effective heat conductor). The Rayleigh number is a dimensionless measure of the ratio of (destabilizing) buoyancy to (stabilizing) molecular diffusion of momentum and buoyancy so it characterizes the degree of instability of the system – it is proportional to the product of the thermal expansion coefficient of the fluid and the imposed temperature difference. Specific values of Ra that give rise to interesting physical behaviour lie in the range $10^3 < \text{Ra} < 10^6$. In this paper we consider two-dimensional flow models ($\Omega \subset \mathbb{R}^2$) and we focus on enclosed cavity flow configurations; the preconditioning methodology considered in this study is critically evaluated for three-dimensional problems in Smethurst [18].

Typical engineering applications are those that involve heat exchange systems – nuclear reactor cooling and internal cooling of computer central processor units are two important examples. The system (1.1)–(1.3) is supplemented by the generic boundary conditions:

$$\vec{u} = \vec{0} \quad \text{on } \Gamma \times [0, \tau], \quad \begin{cases} T = T_g & \text{on } \Gamma_D \times (0, \tau], \\ v \nabla T \cdot \vec{n} = 0 & \text{on } \Gamma_N \times (0, \tau], \end{cases} \tag{1.4}$$

where the boundary $\Gamma = \Gamma_D \cup \Gamma_N$ consists of non-overlapping pieces. A temperature field is specified on Γ_D to create a temperature gradient across the cavity and Γ_N models insulated parts of the boundary. The initial condition models a quiescent initial state:

$$\vec{u}(\vec{x}, 0) = \vec{0}, \quad T(\vec{x}, 0) = 0, \quad \vec{x} \in \Omega \cup \Gamma. \tag{1.5}$$

A conventional solution approach, see e.g. Xin and Le Quéré [22], for the initial-value problem (1.1)–(1.5) uses semi-implicit time integration. This leads to a relatively simple linear algebra (Poisson or Stokes-type) problem at every time step, but there is a CFL stability restriction on the maximum time step size. In contrast, there is no time step restriction in our fully implicit approach. This enables the possibility of self-adaptive time step control – with time steps automatically chosen to "follow the physics".

An outline of the paper is as follows. The temporal and spatial discretization of (1.1)–(1.5) is discussed in Section 2. The linear algebra aspects are discussed in Section 3, and the performance of our implicit time integrator methodology is assessed in Section 4. Results for three different flow problems are presented: the first two example problems are laterally heated cavity flows which ultimately attain a periodic state; the third problem is the classic problem of Rayleigh–Bénard convection in a horizontally stretched domain.

2. Discretization aspects

The "basic" time-stepping algorithm that we employ is the well-known, second-order accurate trapezoid rule (TR). Let the interval $[0, \tau]$ be divided into N steps, $\{t_i\}_{i=1}^N$, and let $\vec{v}^j$ denote the discrete (in time) approximation to $\vec{v}(\cdot, t_j)$. The semi-discretized problem is the following: given $(\vec{u}^n, p^n, T^n)$ at time t_n, and boundary data T_g^{n+1} at time t_{n+1}, compute $(\vec{u}^{n+1}, p^{n+1}, T^{n+1})$ via:

$$\frac{2}{k_{n+1}} \vec{u}^{n+1} - \boldsymbol{v} \nabla^2 \vec{u}^{n+1} + \vec{u}^{n+1} \cdot \nabla \vec{u}^{n+1} + \nabla p^{n+1} - \vec{j} T^{n+1} = \frac{2}{k_{n+1}} \vec{u}^n + \frac{\partial \vec{u}^n}{\partial t} \quad \text{in } \Omega, \tag{2.1}$$

$$-\nabla \cdot \vec{u}^{n+1} = 0 \quad \text{in } \Omega, \tag{2.2}$$

$$\vec{u}^{n+1} = \vec{0} \quad \text{on } \Gamma, \tag{2.3}$$

$$\frac{2}{k_{n+1}} T^{n+1} - v \nabla^2 T^{n+1} + \vec{u}^{n+1} \cdot \nabla T^{n+1} = \frac{2}{k_{n+1}} T^n + \frac{\partial T^n}{\partial t} \quad \text{in } \Omega, \tag{2.4}$$

$$T^{n+1} = T_g^{n+1} \quad \text{on } \Gamma_D, \tag{2.5}$$

$$v \nabla T^{n+1} \cdot \vec{n} = 0 \quad \text{on } \Gamma_N. \tag{2.6}$$

Here, $k_{n+1} := t_{n+1} - t_n$ is the current time step. The term $\frac{\partial \vec{u}^n}{\partial t} := \nu \nabla^2 \vec{u}^n - \vec{u}^n \cdot \nabla \vec{u}^n - \nabla p^n + \vec{j} T^n$ is just shorthand for the acceleration, and $\frac{\partial T^n}{\partial t} := \nu \nabla^2 T^n - \vec{u}^n \cdot \nabla T^n$ similarly represents the rate of change of temperature at time t_n.

The limited stability of TR time stepping for the incompressible Navier–Stokes equations has been extensively discussed in the literature. For example, results showing that the lack of numerical damping within TR can be problematic if the time step is kept fixed (and is not small enough) can be found in Dettmer and Perić [5]. Such problems can be circumvented if the basic TR method is *stabilized* using a simple time-step averaging technique. Full details are given in our papers [10,13]. The averaging is invoked periodically every n_* steps. For such a step, we save the values $t_* = t_n, \vec{u}^* = \vec{u}^n, T^* = T^n$ and having computed the scaled TR velocity update $\vec{d}^n := (\vec{u}^{n+1} - \vec{u}^n)/k_{n+1}$ and the temperature update $d^n := 2(T^{n+1} - T^n)/k_{n+1}$, we set $t_n = t_{n-1} + \frac{1}{2} k_n$ and $t_{n+1} = t_* + \frac{1}{2} k_{n+1}$ and define "shifted" solutions via the updates:

$$\vec{u}^n = \frac{1}{2}(\vec{u}^* + \vec{u}^{n-1}); \quad T^n = \frac{1}{2}(T^* + T^{n-1}); \tag{2.7}$$

$$\frac{\partial \vec{u}^n}{\partial t} = \frac{1}{2}\left(\frac{\partial \vec{u}^n}{\partial t} + \frac{\partial \vec{u}^{n-1}}{\partial t}\right); \quad \frac{\partial T^n}{\partial t} = \frac{1}{2}\left(\frac{\partial T^n}{\partial t} + \frac{\partial T^{n-1}}{\partial t}\right); \tag{2.8}$$

$$\vec{u}^{n+1} = \vec{u}^* + \frac{1}{2} k_{n+1} \vec{d}^n; \quad \frac{\partial \vec{u}^{n+1}}{\partial t} = \vec{d}^n; \tag{2.9}$$

$$T^{n+1} = T^* + \frac{1}{4} k_{n+1} d^n; \quad \frac{\partial T^{n+1}}{\partial t} = \frac{1}{2} d^n. \tag{2.10}$$

We let ε_t denote the user-specified accuracy tolerance. If the TR–AB2 velocity and temperature component L_2 error estimates are denoted by $\|\vec{e}_h^{n+1}\|$ and $\|e_h^{n+1}\|$, respectively, then with or without averaging, the time step k_{n+2} is computed using the heuristic formula:

$$k_{n+2} = k_{n+1}\left(\varepsilon_t / \sqrt{\left\|\vec{e}_h^{n+1}\right\|^2 + \left\|e_h^{n+1}\right\|^2}\right)^{1/3}, \tag{2.11}$$

before carrying on with the integration. In the code used in this study, the averaging frequency parameter n_* is typically fixed so that $n_* = 9$. Some of our results, however, are computed using unstabilized TR and correspond to setting $n_* = \infty$ in the code.

From (2.1) and (2.4), it is evident that a numerical scheme for handling the nonlinear terms $\vec{u}^{n+1} \cdot \nabla \vec{u}^{n+1}$ and $\vec{u}^{n+1} \cdot \nabla T^{n+1}$ is needed at every time step. A conventional algorithm, see Gresho and Sani [11, p. 800], would solve the nonlinear system (2.1)–(2.6) to a predefined accuracy using either fixed point iteration or some variant of Newton iteration. A recent paper by Damanik et al. [3] gives an efficient implementation of the latter approach. An alternative approach is adopted herein: we linearize (2.1) and (2.4) using the explicitly computable "second-order" convection field given by

$$\vec{w}^{n+1} = (1 + (k_{n+1}/k_n))\vec{u}^n - (k_{n+1}/k_n)\vec{u}^{n-1}. \tag{2.12}$$

This is much simpler! At every time step we need to solve a single linear system with $\vec{u}^{n+1} \cdot \nabla \vec{u}^{n+1} := \vec{w}^{n+1} \cdot \nabla \vec{u}^{n+1}$ in (2.1) and $\vec{u}^{n+1} \cdot \nabla T^{n+1} := \vec{w}^{n+1} \cdot \nabla T^{n+1}$ in (2.4). In contrast, the linear systems that arise using Newton's iteration are much more complicated (additional non-zero matrix blocks arise from the linearization of the term $\vec{u}^{n+1} \cdot \nabla T^{n+1}$) and iterative solution techniques typically take twice as many iterations to reach a given tolerance compared to systems arising from a fixed-point linearization, see [7, p. 374]. Computational experiments discussed later reinforce our assertion that temporal stability is not compromised and accuracy is (essentially) maintained using (2.12). This strategy for linearization is that proposed for constant time steps by Simo and Armero [17]. It is also mathematically equivalent to the TRLE algorithm discussed by Layton [14, p. 163].

Spatial discretization will, throughout this paper, be done using a method-of-lines approach based on finite element approximation on a fixed spatial grid of non-overlapping rectangles. We denote the grid subdivision by $\mathcal{T}_h$ and associate the parameter h with the length of the longest edge in $\mathcal{T}_h$. Then, given finite-dimensional approximation spaces $\mathbf{X}_h \subset (H_0^1(\Omega))^2$, $M_h \subset L^2(\Omega)$ and $X_h \subset H_{E_0}^1(\Omega)$, the fully discrete problem is to compute $\left(\vec{u}_h^{n+1}, p_h^{n+1}, T_h^{n+1}\right)$ satisfying:

$$\frac{2}{k_{n+1}}(\vec{u}_h^{n+1}, \vec{v}_h) + \nu(\nabla \vec{u}_h^{n+1}, \nabla \vec{v}_h) + (\vec{w}_h^{n+1} \cdot \nabla \vec{u}_h^{n+1}, \vec{v}_h) - (p_h^{n+1}, \nabla \cdot \vec{v}_h) - \left(T_h^{n+1}, \vec{j} \cdot \vec{v}_h\right)$$

$$= \frac{2}{k_{n+1}}(\vec{u}_h^n, \vec{v}_h) + \left(\frac{\partial \vec{u}_h^n}{\partial t}, \vec{v}_h\right), \tag{2.13}$$

$$(\nabla \cdot \vec{u}_h^{n+1}, q_h) = 0, \tag{2.14}$$

$$\frac{2}{k_{n+1}}(T_h^{n+1}, v_h) + \nu\left(\nabla T_h^{n+1}, \nabla v_h\right) + \left(\vec{w}_h^{n+1} \cdot \nabla T_h^{n+1}, v_h\right) = \frac{2}{k_{n+1}}(T_h^n, v_h) + \left(\frac{\partial T_h^n}{\partial t}, v_h\right), \tag{2.15}$$

for all $(\vec{v}_h, q_h, v_h) \in \mathbf{X}_h \times M_h \times X_h$, where $\vec{w}_h^{n+1} = (1 + (k_{n+1}/k_n))\vec{u}^n - (k_{n+1}/k_n)\vec{u}^{n-1}$. Note that the linearized system is *uncoupled* and can be solved by back-substitution; first, by computing T_h^{n+1} from (2.15), second, by computing $(\vec{u}_h^{n+1}, p_h^{n+1})$ satisfying (2.13) and (2.14).

Low-order mixed approximation methods for the velocity and pressure fields are not stable[1] in general. One mixed method that is stable is the "Taylor–Hood" Q_2–Q_1 method, using continuous piecewise quadratic approximation for the velocity components and continuous piecewise linear approximation for pressure. The Q_2–Q_1 method is combined with Q_2 approximation for the temperature, which leads to a well-balanced approximation, see Gunzburger [12, p. 221]. We also emphasise that the efficiency of the linear solver methodology described in the next section is independent of the mixed approximation employed.

3. A fast linear solver

To generate a linear system of algebraic equations we need to define specific basis sets for the approximation spaces:

$$\boldsymbol{X}_h = \mathrm{span}\left\{ \begin{bmatrix} \phi_i \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ \phi_i \end{bmatrix} \right\}_{i=1}^{n_u} \quad \text{for velocity,}$$

$$M_h = \mathrm{span}\{\psi_j\}_{j=1}^{n_p} \quad \text{for pressure, and}$$

$$X_h = \mathrm{span}\{\phi_k\}_{k=1}^{n_T} \quad \text{for temperature.}$$

As discussed in [10,13], our TR implementation explicitly computes discrete velocity and temperature *updates* $\vec{d}_h^n$ and d_h^n that are scaled by the time step so as to avoid underflow and inhibit subtractive cancellation. The current velocity and temperature solutions are then given by

$$\vec{u}_h^{n+1} = \vec{u}_h^n + k_{n+1}\vec{d}_h^n, \quad T_h^{n+1} = T_h^n + \frac{1}{2}k_{n+1}d_h^n. \tag{3.1}$$

To this end, given the expansions:

$$\vec{d}_h^{n+1} = \begin{bmatrix} \sum_{i=1}^{n_u} \alpha_i^{x,n+1}\phi_i \\ \sum_{i=1}^{n_u} \alpha_i^{y,n+1}\phi_i \end{bmatrix}, \quad p_h^{n+1} = \sum_{j=1}^{n_p} \alpha_j^{p,n+1}\psi_j, \quad d_h^{n+1} = \sum_{k=1}^{n_T} \alpha_k^{t,n+1}\phi_k,$$

the coefficient vectors $\boldsymbol{\alpha}^{u,n+1} = [\boldsymbol{\alpha}^{x,n+1}, \boldsymbol{\alpha}^{y,n+1}], \boldsymbol{\alpha}^{p,n+1}$ and $\boldsymbol{\alpha}^{t,n+1}$ are computed by solving the "saddle-point system" associated with (2.13)–(2.15) – that is

$$\begin{pmatrix} \boldsymbol{F}_v^{n+1} & B^T & -\frac{1}{2}k_{n+1}\overset{\circ}{M} \\ B & 0 & 0 \\ 0 & 0 & F_v^{n+1} \end{pmatrix} \begin{pmatrix} \boldsymbol{\alpha}^{u,n+1} \\ \boldsymbol{\alpha}^{p,n+1} \\ \boldsymbol{\alpha}^{t,n+1} \end{pmatrix} = \begin{pmatrix} \boldsymbol{f}^{u,n+1} \\ \boldsymbol{0} \\ \boldsymbol{f}^{t,n+1} \end{pmatrix}, \tag{3.2}$$

where we have divided the incompressibility constraint by the time step k_{n+1} to preserve the div–grad block symmetry. In (3.2), the matrix $\overset{\circ}{M}$ represents the buoyancy term, and is defined below. The matrix $B = [B_x, B_y]$ is the discrete divergence operator,

$$B_x := [B_x]_{ji} = -\left(\psi_j, \frac{\partial \phi_i}{\partial x}\right), \quad j = 1,\dots,n_p, \ i = 1,\dots,n_u, \tag{3.3}$$

$$B_y := [B_y]_{ji} = -\left(\psi_j, \frac{\partial \phi_i}{\partial y}\right), \quad j = 1,\dots,n_p, \ i = 1,\dots,n_u. \tag{3.4}$$

A consequence of the enclosed flow boundary conditions is that the matrix B^T has a one-dimensional null space spanning constant pressure vectors. This means that the system (3.2) is *singular*. Note that the zero vector in the right-hand side of (3.2) ensures that the system is consistent.

The matrices F_v^{n+1} and F_v^{n+1} in (3.2) are the discrete convection–diffusion-reaction operators for the velocity and temperature respectively:

$$\boldsymbol{F}_v^{n+1} := 2\boldsymbol{M} + \nu k_{n+1}\boldsymbol{A} + k_{n+1}\boldsymbol{N}(\vec{w}_h^{n+1}), \tag{3.5}$$

$$F_v^{n+1} := M + \frac{1}{2}\nu k_{n+1}A + \frac{1}{2}k_{n+1}N(\vec{w}_h^{n+1}), \tag{3.6}$$

with Scalar-analogue component matrices:

$$M := [M]_{ij} = (\phi_i, \phi_j), \quad i,j = 1,\dots,n_u, \tag{3.7}$$

$$A := [A]_{ij} = (\nabla\phi_i, \nabla\phi_j), \quad i,j = 1,\dots,n_u, \tag{3.8}$$

$$N(\vec{w}_h) := [N]_{ij} = (\vec{w}_h \cdot \nabla\phi_i, \phi_j) \quad i,j = 1,\dots,n_u. \tag{3.9}$$

---

[1] See Elman et al. [7, Ch. 5] for a full discussion of *inf-sup* stability.

The matrices $\boldsymbol{M}$, $\boldsymbol{A}$ and $\boldsymbol{N}$ in (3.5) are the block diagonal matrix analogues of M, A and N, respectively. The velocity-temperature coupling in (3.2) is represented by the matrix:

$$\overset{\circ}{M} := \left(\overset{\circ}{M}\right)_{ij} = ([0, \phi_i], \phi_j), \quad i = 1, \ldots, n_u,\, j = 1, \ldots, n_T. \tag{3.10}$$

Our solution algorithm is right-preconditioned GMRES with a preconditioner that is specially tailored to the structured matrix (3.2). The strategy we use is a natural extension of the block upper triangular methodology[2] discussed in [7, Section 8.1]. To illustrate the approach, we express the system (3.2) (omitting the matrix subscripts/superscripts) with a preconditioner $\mathcal{P}$ so that:

$$\begin{pmatrix} \boldsymbol{F} & B^T & -T \\ B & 0 & 0 \\ 0 & 0 & F \end{pmatrix} \mathcal{P}^{-1} \quad \mathcal{P} \begin{pmatrix} \boldsymbol{\alpha}^u \\ \boldsymbol{\alpha}^p \\ \boldsymbol{\alpha}^t \end{pmatrix} = \begin{pmatrix} \boldsymbol{i}^u \\ \boldsymbol{0} \\ \boldsymbol{f}^t \end{pmatrix}.$$

Introducing the velocity–pressure Schur complement matrix $S = B\boldsymbol{F}^{-1}B^T$, an *ideal* block triangular preconditioner:

$$\mathcal{P} \equiv \begin{pmatrix} \boldsymbol{F} & B^T & -T \\ 0 & -S & 0 \\ 0 & 0 & F \end{pmatrix}, \tag{3.11}$$

is motivated by the identity:

$$\begin{pmatrix} \boldsymbol{F} & B^T & -T \\ B & 0 & 0 \\ 0 & 0 & F \end{pmatrix} \underbrace{\begin{pmatrix} \boldsymbol{F}^{-1} & \boldsymbol{F}^{-1}B^TS^{-1} & \boldsymbol{F}^{-1}TF^{-1} \\ 0 & -S^{-1} & 0 \\ 0 & 0 & F^{-1} \end{pmatrix}}_{\mathcal{P}^{-1}} \equiv \begin{pmatrix} I & 0 & 0 \\ B\boldsymbol{F}^{-1} & I & B\boldsymbol{F}^{-1}TF^{-1} \\ 0 & 0 & I \end{pmatrix}.$$

A simple permutation of the second and third rows and columns of the matrix on the right here generates a lower triangular matrix with ones on the diagonal. Thus, the eigenvalues of this matrix are identically one. It can also be shown that this matrix has Jordan blocks of dimension two, which implies that GMRES applied to (3.2) with right preconditioner $\mathcal{P}$ will converge in precisely two iterations, independent of the convection field $\vec{w}_h^{n+1}$ and the values of the parameters h, k_{n+1}, ν and v.

It follows from (3.11) that the action of $\mathcal{P}^{-1}$ on a vector can be implemented as a three-step process. First, we solve systems associated with the matrix operator F and the Schur complement matrix S; second, we perform a matrix–vector multiply with $T = \frac{k_{n+1}}{2}\overset{\circ}{M}$ and with B^T, and finally we solve the two scalar systems associated with the matrix operator $\boldsymbol{F}$. For a practical algorithm, these matrix solves will be replaced by *inexact* solves associated with a fixed number (one or two, typically) of algebraic multigrid (AMG) V-cycles.

It is not practical to work with the Schur complement S, and we use two strategies that circumvent the use of this matrix. These two alternatives are identified in [7, Section 8.2]. The first approach, referred to as *Pressure Convection–Diffusion* (PCD) preconditioning, is a triple product approximation. The ingredients are a matrix–vector multiply with the operator $\boldsymbol{F}$ defined on the pressure space, together with solves with the pressure mass matrix Q and a pressure diffusion matrix $A_* := B\boldsymbol{M}_*^{-1}B^T$. In this work, we follow the suggestion of Elman & Tuminaro [8] and implement PCD via:

$$S^{-1} = \left(B\boldsymbol{F}^{-1}B^T\right)^{-1} \approx Q_*^{-1}F_*A_*^{-1}, \tag{3.12}$$

where Q_* is the diagonal of Q, $\boldsymbol{M}_*$ is the diagonal of $\boldsymbol{M}$, and F_* is constructed like $\boldsymbol{F}_v^{k+1}$ in (3.5) except that the velocity functions ϕ_i and ϕ_j are replaced by pressure basis functions ψ_i and ψ_j. The second approach, referred to as *Least Squares Commutator* (LSC) preconditioning, avoids the construction of the reaction-convection–diffusion operator on the pressure space and is given by

$$S^{-1} = \left(B\boldsymbol{F}^{-1}B^T\right)^{-1} \approx A_*^{-1}\left(B\boldsymbol{M}_*^{-1}\boldsymbol{F}\boldsymbol{M}_*^{-1}B^T\right)A_*^{-1}. \tag{3.13}$$

As discussed above, in a practical implementation, the action of A_*^{-1} in (3.12) and (3.13) will be performed inexactly using AMG. Note that the rank deficiency of B^T means that A_* inherits the same one-dimensional null space.[3] Preconditioning with the Schur complement approximations (3.12) and (3.13) and with the matrix operators F_v^{k+1}, $\boldsymbol{F}_v^{k+1}$ in $\mathcal{P}^{-1}$ will be referred to as *exact* PCD and LSC preconditioning, respectively. To show the effectiveness of these two preconditioning strategies, we describe their performance for solving two representative examples of the system (3.2).

First we consider a "snapshot" system that arises after 1200 time units when computing time accurate solutions of a model of natural convection in a tall (aspect ratio of 8:1) cavity—full details are given in Section 4.3. The spatial subdivision

---

[2] We did not explore the alternative approach of decoupling the block system (3.2) and solving the two component problems iteratively. The additional issue that doing so would introduce is that of designing a suitable stopping criterion for solving the temperature equation. Our fully-coupled preconditioner circumvents this issue.

[3] As discussed in [7, Section 2.3], there is no intrinsic difficulty in solving a consistent singular system using preconditioned iteration.

Fig. 3.1. Details of grid (to scale) showing the high aspect ratio rectangles in the two coordinate directions.
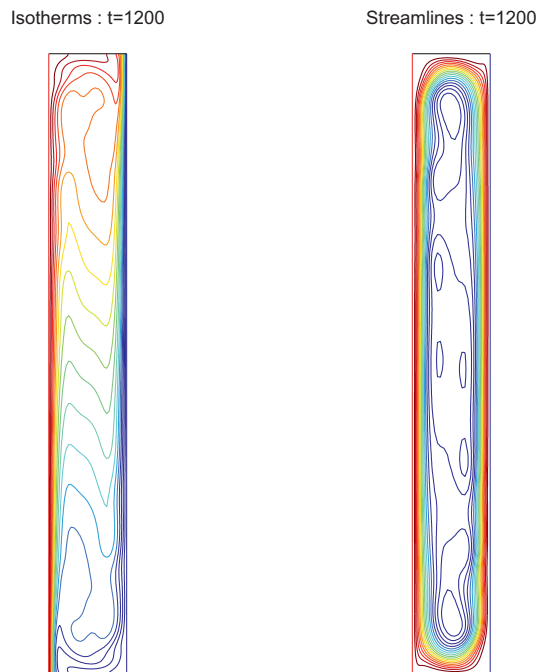

Fig. 3.2. Laterally heated 8:1 cavity Ra = 3.4×10^5, Pr = 0.71: snapshot of reference solution at t = 1200.

is a 31×248 nonuniform grid. The matrix dimensions are $n_u = n_T = 31311$, $n_p = 7968$, so the system dimension is 101901. As illustrated in Fig. 3.1, there are stretched elements (with aspect ratio of 1:16 in the vertical direction) next to the cavity walls to give better resolution of the shear layers that are generated in the early-time transient, see [2]. The "snapshot" flow solution is shown in Fig. 3.2. This is taken at a time when the system has reached a quasi-steady periodic cycle—temperature "hotspots" continuously circle clockwise around the cavity with a frequency of approximately 3.4 time units. Shown in Fig. 3.3 is the time step evolution of the adaptive integrator with the temporal tolerance set to 3×10^{-5}. It can be seen from this graph that at t = 1200, the time step k_{n+1} is 0.082(±0.003) time units. This gives ∼40 sample points per cycle – this is close to the minimum needed if our aim is to generate qualitatively accurate solutions, and it is slightly smaller than the empirically chosen constant time step Δt = 0.1 taken in [5].

To investigate the behaviour of our fast solver methodology we first consider solving the linearized system (3.2) using *exact* LSC and PCD preconditioning. GMRES convergence curves for a tight tolerance (residual reduction of 10^{-9}) are shown in Fig. 3.4. There are two potential sources of difficulty: first, grid stretching is known to be problematic for algebraic multigrid as well as to lead to a deterioration in the effectiveness of the LSC approach when solving steady problems, see [20], (the PCD methodology is generally robust however); and second, the highly recirculating convection field means that simple point smoothing is unlikely to be effective when solving the convection–diffusion sub-problems using geometric or algebraic multigrid.[4] Looking at the results for *exact* preconditioning (using direct solves for the subsidiary systems with $\boldsymbol{F}, F$ and A_*), we

---

[4] This issue is explored in some detail in [7, Section 4.3]. Alternating line relaxation is recommended instead.
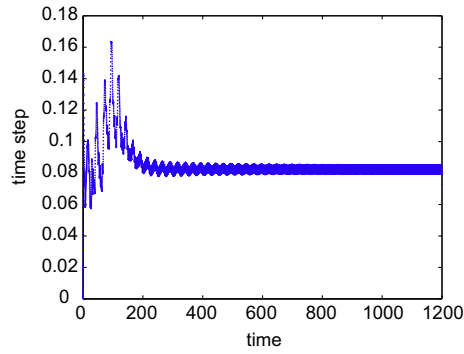
Fig. 3.3. Laterally heated 8:1 cavity Ra = 3.4×10^5, Pr = 0.71: evolution of the time step using the TR-AB2 integrator with $n_* = \infty$.
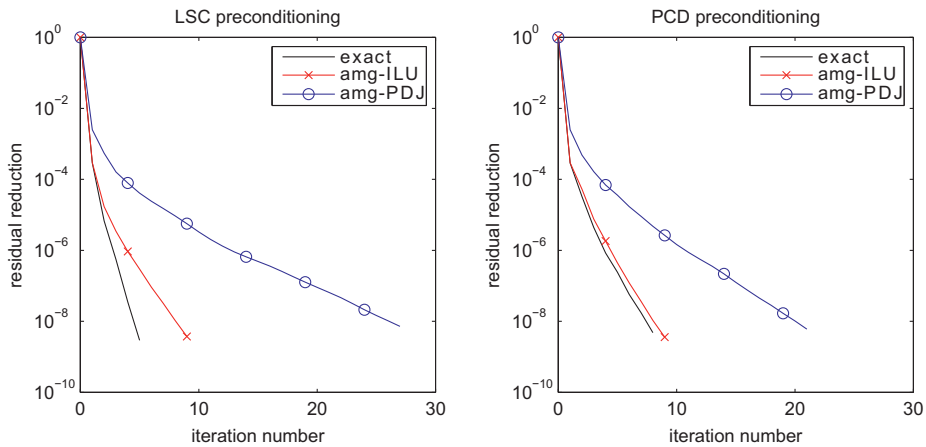

Fig. 3.4. GMRES convergence using Least Squares Commutator preconditioner (left) and Pressure Convection Diffusion preconditioner (right) for the snapshot flow solution illustrated in Fig. 3.2 with an automatically chosen time step of 0.082 units. (Note that $v = 0.00145$ and $v = 0.00203$ in this case.)
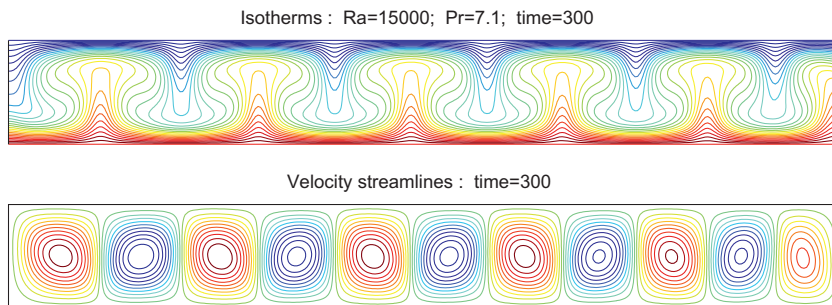

Fig. 3.5. Vertically heated 1:8 cavity: snapshot of pseudo-steady solution.

see that LSC and PCD are equally robust with regard to the issue of the stretched grid. Indeed, for a realistic residual reduction factor of 10^{-6} we obtain convergence in only three iterations for LSC and five iterations for PCD.

Achieving such good performance in the *inexact* case is hardly more challenging. Results obtained for AMG inexact LSC and PCD preconditioning are also shown in Fig. 3.4. We make use of a MATLAB version[5] of the code HSL_MI20 [1] and replace each of the solves with F_v^{n+1} ($v = 0.00145$ for this problem), F_v^{n+1} ($v = 0.00203$) and A_*^{-1} with a single V-cycle of AMG. Results are shown for two different smoothing strategies: a standard 2–2 (presmoothing–postsmoothing) V-cycle using point damped Jacobi (PDJ) with damping parameter $\omega = 0.5$, and, to enhance robustness, a hybrid approach which is a V-cycle with one sweep of ILU (0)

---

[5] The source code is freely available in release 3.1 of the IFISS software package [16].

with a lexicographic left → right and bottom → top ordering on the finest level and two sweeps of PDJ with $\omega = 0.5$ on all coarser levels. As might be expected, the ILU results are significantly better than those obtained using PDJ smoothing. The point smoothing approach appears to be viable nonetheless. For a realistic residual reduction factor of 10^{-6} we obtain convergence in thirteen iterations for LSC and eleven iterations for PCD. Similar results for three-dimensional problems can be found in [1].

We remark here that the large presence of the mass matrix as a component of the coefficient matrices $\boldsymbol{F}$ and F, which comes from time-stepping, is a crucial ingredient in the above examples. As the temporal error tolerance is tightened, the associated reduction in time step means that the effectiveness of the AMG solver is further increased. A more challenging test for our iterative solver is when the TR integrator generates large time steps, $k_n \sim O(1)$ say, which is typical behaviour when a system goes to a steady-state.

Our second "snapshot" system is an example of such a problem, arising from a model of Rayleigh–Bénard convection in a (thin) 1:8 aspect ratio cavity – full details are given in Section 4.2. For the specific values of Pr and Ra), the viscosity parameters are $\boldsymbol{\nu} = 0.0218$ and $\nu = 0.00306$. The "snapshot" solution is shown in Fig. 3.5. The spatial subdivision is a 160×20 uniform grid, and the matrix dimensions are $n_u = n_T = 13161$, $n_p = 3381$, so the overall system dimension is 42864. The Boussinesq problem models a classical example of instability: initially the flow solution tends to a quiescent equilibrium state ($\vec{u}_h = \vec{0}$) with the temperature T_h horizontally stratified. The equilibrium solution is not stable, however. This is illustrated by the evolution of the kinetic energy shown in Fig. 3.6. As can be seen, the time-step grows monotonically until it reaches a maximum at about 100 time units. At this stage the velocity solution is close to zero and the temperature solution is a linear function of the vertical coordinate. There is a time step rejection at this point, however, after which the TR integrator successively cuts back the time step and follows a rapid transient – ultimately settling on a quasi-steady solution which has the characteristic multiple "convective roll" pattern shown in Fig. 3.5. At the "final" time of 300 units the time step is approximately 4 time units, see Fig. 3.6.

As shown in Fig. 3.7, the iterative solver needs to work slightly harder in this case. There is little to choose between LSC and PCD preconditioning strategies. As in the first example, the ILU results are somewhat better than those obtained using PDJ smoothing. The point smoothing approach is still perfectly feasible: for residual reduction factor of 10^{-6} we obtain convergence in seventeen iterations for LSC and twenty-one iterations for PCD. Moreover, the fact that we are using standard Galerkin approximation without "tuning parameters" makes for a very clean discretization and, in our opinion, makes the preconditioning methodology look even more attractive.

4. Assessment of linearization and time-stepping

We present computational results for a selection of representative Boussinesq problems here. Our primary focus is on the behaviour of the linearized TR time-stepping algorithm. Thus, unless stated otherwise, the stabilized TR integrator in Section 2 is run with the saddle point system at each time level solved using the MATLAB "backslash" sparse solver. Some results generated using inexact AMG preconditioning will be presented in Section 4.3.

4.1. Laterally heated horizontal cavity

We consider the easiest problem first – that of a horizontally heated 1:4 aspect ratio cavity with small Prandtl number, Pr = 0.015, associated with gallium arsenide. This is a classical test problem and is a popular Boussinesq benchmark problem, see [15]. The physical problem is motivated by models of oscillatory instability in the so-called Bridgman technique for growing crystals for optoelectronic applications. For further motivation, see Winters [21] and the references therein. For

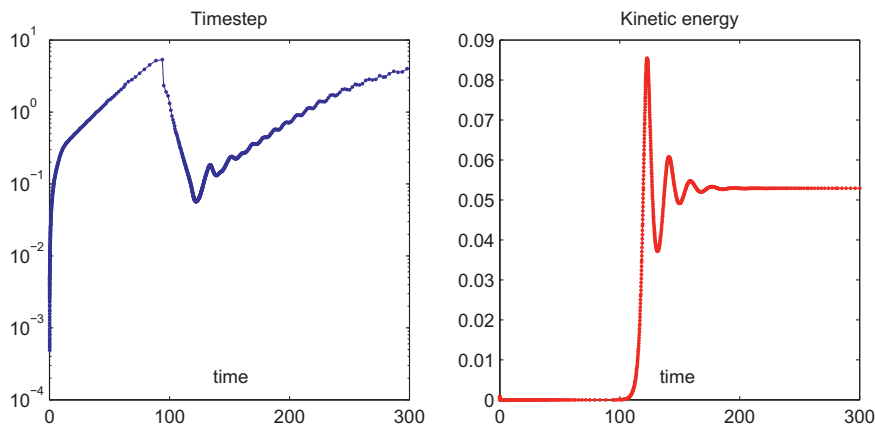**Fig. 3.6.** Vertically heated 1:8 cavity Ra = 1.5×10^4, Pr = 7.1: evolution of the time step using the TR-AB2 integrator with $n_* = 9$ and $\varepsilon_t = 10^{-6}$ (left) and kinetic energy evolution $\int_\Omega (\vec{u}_h^n)^2$ (right).
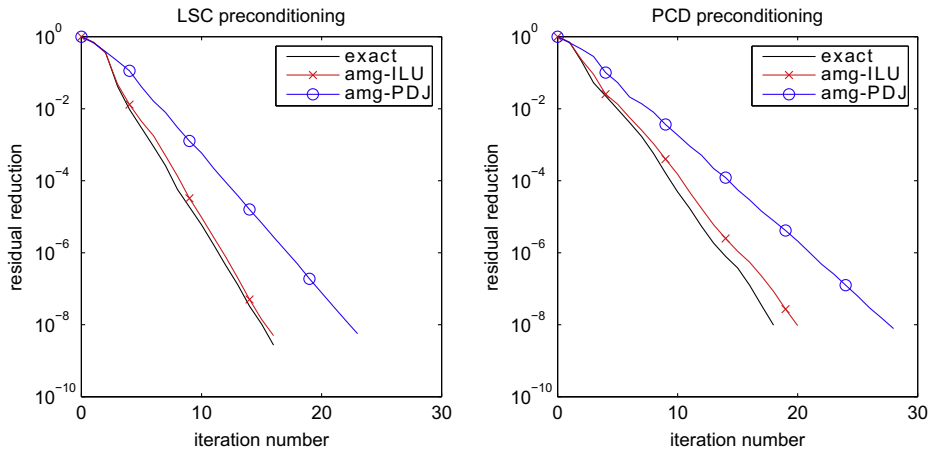
Fig. 3.7. GMRES convergence using Least Squares Commutator preconditioner (left) and Pressure Convection Diffusion preconditioner (right) for the snapshot flow solution illustrated in Fig. 3.5 with an associated time step of 4 units. (Note that $v = 0.0218$ and $v = 0.00306$ in this case.)

small Rayleigh number, the flow is steady, but if the temperature gradient is large enough then the system will undergo a transition from a stable to an oscillatory flow via a Hopf bifurcation – a detailed discussion can be found in Gelfgat et al. [9].

We take $\Omega = [0,4] \times [0,1]$ as the problem domain and we set the temperature difference on the vertical walls so that $T(x = 0, t \to \infty) = 0.5$, $T(x = 4, t \to \infty) = -0.5$. To produce a gradually emerging flow, the vertical walls are smoothly heated/cooled so that:

$$T(x = 0, t) = \frac{1}{2}(1 - e^{-10t}); \quad T(x = 4, t) = -\frac{1}{2}(1 - e^{-10t}), \tag{4.1}$$

and both horizontal walls are assumed to be insulated. We solve the flow problem with Ra = 3000. This corresponds to a Grashof number Gr = Ra/Pr = 2×10^5 which is greater than the critical value Gr* = 1.32×10^5 given in [9]. We take an 80×20 uniform grid initially and run stabilized TR to 300 time units with ε_t (the time-accuracy tolerance) set to 3×10^{-5}. As shown in Fig. 4.1, the computational solution quickly evolves into a stable cyclic configuration with a period of about 18 time units. The periodic solution is skew-symmetric (invariant with respect to a rotation through π about the cavity centre) and alternates in time between one and three vortices. This is visualised by the stationary streamline sequence shown in Fig. 4.2. The time step evolution is shown in Fig. 4.3. This plot makes it clear that the solution reaches the periodic configuration at approximately 50 time units, after which the TR integrator settles on a time step of 0.240(±0.033), about 75 points per cycle.

To provide quantitative evidence of the accuracy of these computational results, a set of solution statistics for the gridpoint $(0.1, 0.1)$ is given in Table 4.1. The reference results in the first column are those generated with the 20×80 grid and with $\varepsilon_t = 3 \times 10^{-5}$ and correspond to the results in Figs. 4.1, 4.2, 4.3. The extremal point values and the bottom-line estimate of the oscillation period are computed by averaging the results over the final four cycles (the interval $(220, 300)$ in Fig. 4.1). The
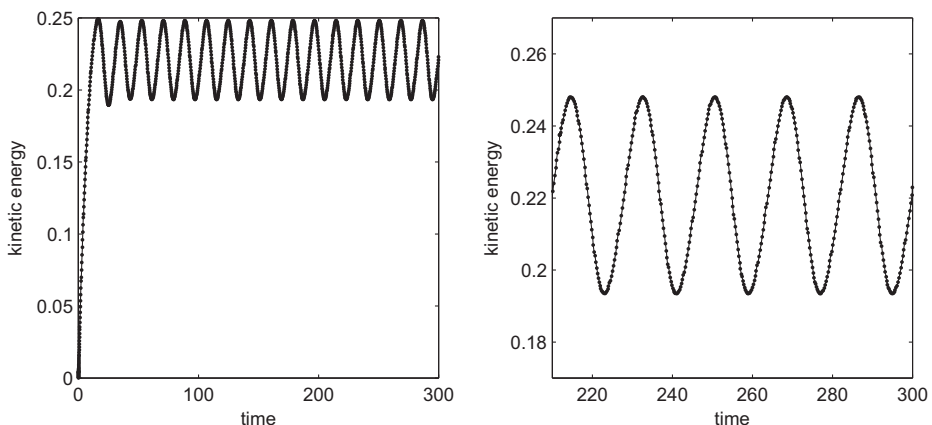

Fig. 4.1. Laterally heated 1:4 cavity Ra = 3000, Pr = 0.015: kinetic energy evolution.
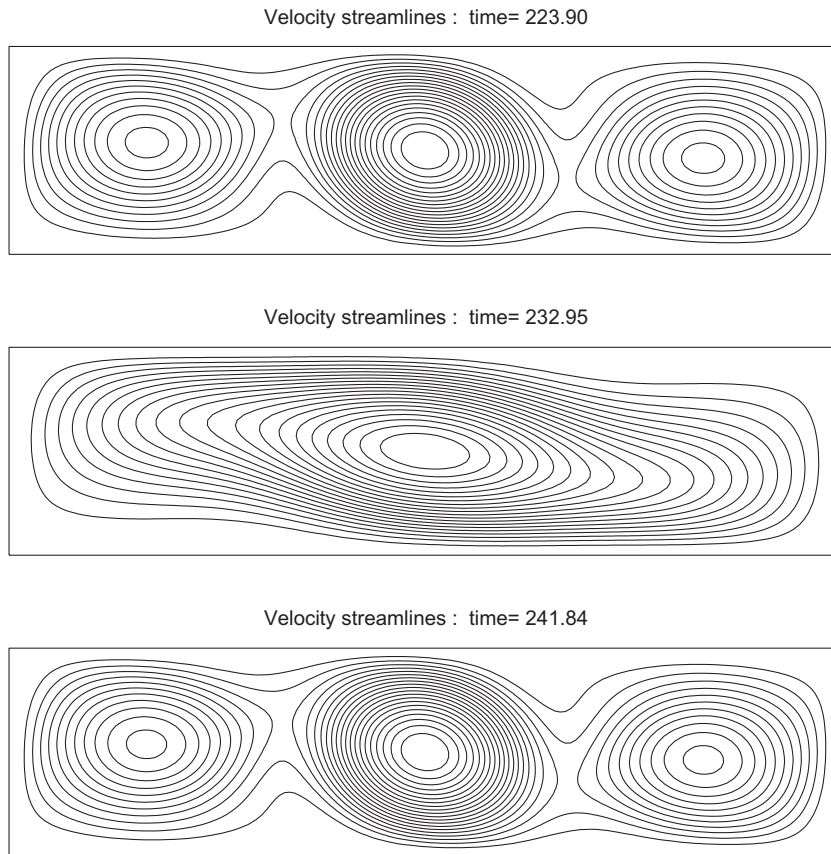
Velocity streamlines : time= 223.90

Velocity streamlines : time= 232.95

Velocity streamlines : time= 241.84

Fig. 4.2. Laterally heated 1:4 cavity Ra = 3000, Pr = 0.015: snapshots of the flow solution.

reference results show three significant–digit agreement with those obtained when the problem is solved on a finer spatial grid (second column) and also with the high accuracy results (third column) obtained using the refined grid in combination with a tight time accuracy tolerance.

4.2. Vertically heated horizontal cavity

The thin cavity configuration of the first test problem is retained in the second example but the heating direction is vertical instead of horizontal. The Prandtl number is set to Pr = 7.1, a typical value for water. Such a configuration is of huge
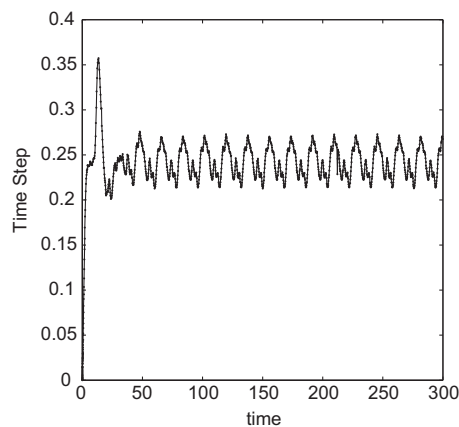

Fig. 4.3. Laterally heated 1:4 cavity Ra = 3000, Pr = 0.015: evolution of the time step using the TR-AB2 integrator with $n_* = 9$ and $\varepsilon_t = 3 \times 10^{-5}$.

Table 4.1
Point data statistics for laterally heated 1:4 cavity Ra = 3000, Pr = 0.015 at (0.1,0.1).

Subdivision ε_t	Reference 20×80 $3 \cdot 10^{-5}$	h-refined 30×120 $3 \cdot 10^{-5}$	ht-refined 30×120 $1 \cdot 10^{-6}$
$(u_x)_{min}$	−0.0425	−0.0425	−0.0423
$(u_x)_{max}$	−0.0191	−0.0192	−0.0194
Δu_x	0.0234	0.0233	0.0228
$\bar{u}_x$	−0.0308	−0.0308	−0.0308
$(u_y)_{min}$	0.0146	0.0146	0.0148
$(u_y)_{max}$	0.0338	0.0338	0.0336
Δu_y	0.0193	0.0192	0.0188
$\bar{u}_y$	0.0242	0.0242	0.0242
T_{min}	0.4622	0.4622	0.4623
T_{max}	0.4648	0.4648	0.4648
ΔT	0.0026	0.0026	0.0025
$\bar{T}$	0.4635	0.4635	0.4635
Period	17.983	17.969	17.972

importance because of the laboratory experiments of Bénard in 1900 and the subsequent stability analysis of Rayleigh in 1916 – for further motivation see Drazin [6, Chapter 6]. Here we take $\Omega = [0,8] \times [0,1]$ as the problem domain (1:8 cavity) and heat/cool the horizontal walls so that:

$$T(y = 0, t) = \frac{1}{2}(1 - e^{-10t}); \quad T(y = 1, t) = -\frac{1}{2}(1 - e^{-10t}). \tag{4.2}$$

The left and right cavity walls are assumed to be insulated. Our aim is to compute the transition from an unstable equilibrium flow to a stable configuration of convection rolls. The computation reported in Section 3 is done with a 160×20 uniform grid, a Rayleigh number of Ra = 1.5×10^4 and a temporal accuracy tolerance of $\varepsilon_t = 1 \times 10^{-6}$, see Figs. 3.5 and 3.6. Perhaps unsurprisingly, the system evolution is extremely sensitive to the choice of ε_t. To illustrate this point, Fig. 4.4 shows the evolution of the time step and kinetic energy when the problem in Fig. 3.6 is solved with a more relaxed tolerance of $\varepsilon_t = 1 \times 10^{-4}$. At first sight, comparison of Figs. 3.6 and 4.4 suggests that there is no agreement between these two runs!

The issue here is that the instability "trigger" for the equilibrium solution to lose stability is numerical round-off. In Fig. 4.4 the stabilized TR integrator is taking larger and larger time steps (the penultimate step is 94 time units) and the termination time of 300 units is not quite long enough for instability to set in.[6] The kinetic energy evolution in Fig. 4.4 shows that 100 time units is indeed the point at which stability is lost, which is in agreement with the results of Fig. 3.6. Moreover at the final time of 300 units the velocity solution is $O(10^{-6})$. If the integration were to be *restarted* from this solution then it would evolve as in Fig. 3.6 to the convection roll solution in Fig. 3.5.

4.3. Laterally heated tall cavity

The final problem – the MIT benchmark problem proposed by Christon et al. [2] – is a more challenging test of our solution strategy. The aim is to model a horizontally heated tall (8:1 aspect ratio) cavity with Prandtl number, Pr = 0.71 at a near-critical Rayleigh number of Ra* = 3.4×10^5. Although the geometry is simple, the solution has a complicated multiscale structure with a complex time behaviour resulting from travelling waves in the vertical boundary layers interacting with thermal instabilities along horizontal walls. As discussed in Xin & LeQuéré [22], the long-time solution dynamics are extremely sensitive to small changes in the benchmark problem statement. Steady solutions are not stable since there is a Hopf bifurcation to non-skew-symmetric periodic solutions at Ra = 3.06×10^5 and a second bifurcation to a branch of skew-symmetric periodic solutions at Ra = 3.11×10^5. As this is the only stable branch of solutions for Rayleigh numbers close to the benchmark value Ra*, comparisons of numerical results obtained using different approaches are meaningful. We will compare results obtained using our TR strategy with published reference results in [2]. Our aim at the outset is to see if the linearized TR methodology can solve such a demanding problem in "black-box" fashion—that is, without tuning parameters, and not knowing the structure, for example the period, of the long-time solution beforehand.

Motivated by the excellent results obtained by Davis & Bänsch [4], our initial attempt to solve the MIT problem was to run linearized TR–AB2 with $n_* = \infty$ on a stretched 31×248 spatial grid. To construct such a grid, our grid generator includes stretching parameters α_x and α_y to control the wall refinement in the x and y directions respectively. The grid shown in Fig. 3.1 corresponds to setting $\alpha_x = 2$ and $\alpha_y = 1$ and reproduces the specific mesh grading used in [4]. The top and bottom cavity walls are left insulated and the vertical walls are smoothly heated/cooled so that:

---

[6] We note that this issue is not seen using unstabilized TR because ringing makes the time step stagnate and prevents the integrator from taking large time steps when approaching a steady state.
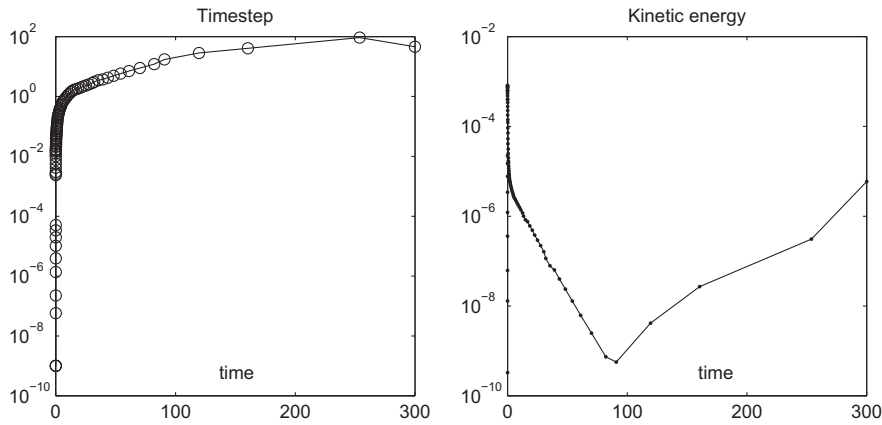
Fig. 4.4. Vertically heated 1:8 cavity Ra = 1.5×10^4, Pr = 7.1: evolution of the time step using the TR-AB2 integrator with $n_* = 9$ and $\varepsilon_t = 10^{-4}$ (left) and kinetic energy evolution (right).

$$T(x = 0, t) = \frac{1}{2}(1 - e^{-10t}); \quad T(x = 1, t) = -\frac{1}{2}(1 - e^{-10t}). \tag{4.3}$$

Running the unstabilized TR algorithm with an accuracy tolerance $\varepsilon_t = 3 \times 10^{-5}$ gives the results in Section 3 that were discussed earlier, see Figs. 3.1, 3.2, 3.3, 3.4. These results are qualitatively correct – after a long transient the solution evolves into a stable cycle with a period of ∼3.40 time units (the benchmark result in [22] is 3.411). Moreover, the solution snapshot in Fig. 3.2 is visibly skew-symmetric, and the kinetic energy (not shown here) tends to a value of ∼0.241 (the benchmark results are 0.240 ± 0.001).

A closer comparison with published results for the benchmark reveals, however, some deficiencies. The benchmark specification [2] examines data statistics at mesh points $P_1 = (0.181, 7.370)$ and $P_2 = (0.819, 7.370)$. We refer first to Fig. 4.5 and Table 4.2. The figure shows the temperature at P_1 computed using the TR-AB2 method with $n_* = 9$. (The results for $n_* = \infty$ are virtually identical in these cases.) The graph on the right shows an accurate solution; this is discussed in more detail below. The table summarises point data statistics for mesh points P_1 and P_2 computed using five different TR solution strategies. The highlighted results in the first column are of high accuracy, obtained using a stringent time-accuracy tolerance $\varepsilon_t = 5 \cdot 10^{-7}$ and a fully coupled TR, where the nonlinear equations at each time level are solved to a relative accuracy of 10^{-5}. These will be used as reference results in the sequel. (By way of comparison, $\Delta T = 0.0427$ in [22].) The other columns contain results, from left to right, for a fully coupled TR with milder tolerance $\varepsilon_t = 3 \cdot 10^{-5}$; a simple linearization of TR using a single Picard step with first order time accuracy; Picard linearization with two Picard steps; and an adaptive BDF2 integrator described in [11, p. 805]. The solvers are all run on the same stretched 31×248 grid to a time of 1200 units. The extremal point values and the estimated period are computed over the final twenty cycles.

What is most striking here is that (on the left side of Fig. 4.5) the amplitudes of the cyclic oscillations of the temperature are too high by about 50%. The velocity components (not shown) behave similarly. We attribute this inaccuracy to the
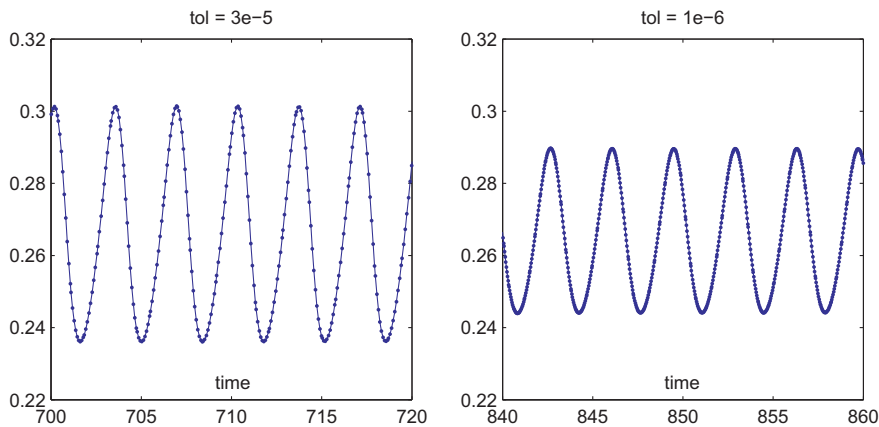

Fig. 4.5. Reference MIT problem: temperature at data point (0.181, 7.370) computed using the TR-AB2 integrator with $n_* = 9$ with relaxed tolerance $\varepsilon_t = 3 \times 10^{-5}$ (left) and a tight tolerance $\varepsilon_t = 10^{-6}$ (right).

Table 4.2
Computed MIT reference statistics computed using nonlinear TR time stepping ($n_* = \infty$).

Method timestepping linearisation	Reference TR-AB2 Picard	fullTR TR-AB2 Picard	linTR TR-AB2 Picard	TR-AB2 Picard linTR2	GL-BDF2 Picard BDF2
ε_t	**$5 \cdot 10^{-7}$**	$3 \cdot 10^{-5}$	$3 \cdot 10^{-5}$	$3 \cdot 10^{-5}$	$3 \cdot 10^{-5}$
Nonlinear tolerance	**$1 \cdot 10^{-5}$**	$1 \cdot 10^{-5}$	1 step	2 steps	$1 \cdot 10^{-5}$
# Nonlinear steps	$\sim$**4**	$\sim$**4**	1	2	$\sim$5
Total # timesteps	**39368**	9604	12820	10633	13977
Mean timestep	**0.028**	0.105	0.092	0.105	0.067
$(u_x)_{\min}$	**0.0346**	0.0345	0.0365	0.0333	0.0349
$(u_x)_{\max}$	**0.0868**	0.0871	0.1237	0.0926	0.0849
Δu_x	**0.0522**	0.0526	0.0872	0.0593	0.0501
$\overline{u}_x$	**0.0607**	0.0608	0.0801	0.0629	0.0599
$(\Delta p)_{\min}$	**-0.0125**	-0.0126	-0.0190	-0.0138	-0.0121
$(\Delta p)_{\max}$	**0.0074**	0.0074	0.0111	0.0084	0.0069
$\Delta(\Delta p)$	**0.0198**	0.0199	0.0301	0.0222	0.0191
$\overline{\Delta p}$	**-0.0026**	-0.0026	-0.0039	-0.0027	-0.0026
$T_{\min}$	**0.2461**	0.2459	0.2322	0.2438	0.2467
$T_{\max}$	**0.2872**	0.2872	0.2988	0.2898	0.2861
ΔT	**0.0411**	0.0414	0.0666	0.0461	0.0395
$\overline{T}$	**0.2666**	0.2665	0.2655	0.2668	0.2664
Period	**3.4135**	3.4211	4.0177	3.4248	3.4319

Table 4.3
Computed MIT reference statistics computed using linearized TR time stepping ($n_* = 9$).

	Reference	$\varepsilon_t = 3 \cdot 10^{-5}$	$\varepsilon_t = 1 \cdot 10^{-6}$
$(u_x)_{\min}$	**0.0346**	0.0329	0.0339
$(u_x)_{\max}$	**0.0868**	0.1170	0.0917
Δu_x	**0.0522**	0.0841	0.0578
$\overline{u}_x$	0.0607	0.0750	0.0628
$(\Delta p)_{\min}$	**-0.0125**	-0.0178	-0.0135
$(\Delta p)_{\max}$	**0.0074**	0.0116	0.0082
$\Delta(\Delta p)$	**0.0198**	0.0294	0.0218
$\overline{\Delta p}$	**-0.0026**	-0.0031	-0.0027
$T_{\min}$	**0.2461**	0.2362	0.2442
$T_{\max}$	**0.2872**	0.3012	0.2896
ΔT	**0.0411**	0.0650	0.0454
$\overline{T}$	**0.2666**	0.2687	0.2669
Period	**3.4135**	3.382	3.412

decoupling of the momentum and energy equations in the linearized system (2.13)–(2.15). Thus, in cases where the physics is "tightly coupled", we might expect that the linearization strategy is going to be ineffective.

This supposition is reinforced by the observation that essentially perfect agreement with the results in [22] can be obtained when we run our code using fully coupled TR (solving the nonlinear equations at each time level to a relative accuracy of 10^{-5}) with all other parameters unchanged; this is shown in the second column of Table 4.2. A simple linearization of TR using a single Picard step with first order time accuracy (the third column, "linTR", in Table 4.2) gives very poor results – both the period and the amplitude of oscillations differ in the leading digit from the reference values. The results obtained when two Picard steps are done at each nonlinear iteration (column "linTR2") are, in contrast, surprisingly good. This reinforces our view that the momentum–temperature (de-) coupling is very important in this problem. Note that over-diffusive results were obtained using the self-adaptive BDF2 integrator (column "GL–BDF2") described in [11, p. 805] – these results are much less accurate than the analogous results for TR–AB2 in the second column.

We return now to discussion of Fig. 4.5, which, again, shows results for the linearized TR method (2.13)–(2.15) with time-step stabilization. In this computation, our stabilized TR code is first run to 750 units with $\varepsilon_t = 3 \times 10^{-5}$. (This target time is long enough for the numerical solution to reach a stable periodic cycle.) A reference point temperature is shown in the left hand plot and we see that the integrator has settled on an approximate time step of 0.085 time units (approximately 40 points per period). The overshoot in the amplitude of the oscillations is self-evident, as can be seen from comparison with the reference data of Table 4.2. The computation is then restarted from this point with $\varepsilon_t = 10^{-6}$. This tighter tolerance leads to a smaller time step (approximately 0.034 time units) which generates about 100 points per period. After a smooth
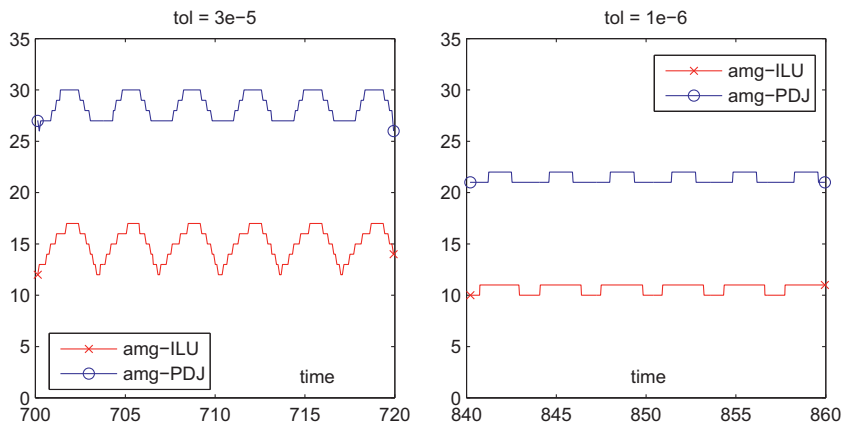
Fig. 4.6. GMRES iterations recorded using inexact PCD preconditioning when generating the solution in Fig. 4.5.

transient (not shown in Fig. 4.5) the TR integrator settles on the periodic solution illustrated in the right hand plot – with a significant decrease in the oscillation amplitude!

The fine-tolerance linearized TR results in Fig. 4.5 are visibly more accurate than the relaxed tolerance "linTR2" results discussed earlier. To quantify this, a set of point data statistics computed over the final seven cycles are tabulated in Table 4.3 together with the reference values from Table 4.2. Running with a tighter tolerance is especially appealing from a fast linear algebra perspective. This can be seen in Fig. 4.6 where we track the iteration counts for a GMRES residual reduction of 10^{-6} that were recorded in the process of computing the solution shown in Fig. 4.5. We use the PCD preconditioning strategy described in Section 3. Note that for increased efficiency (because the AMG setup phase is very slow MATLAB) the AMG grid sequence generated at the first time step is stored (at times $t = 700$ and $t = 840$ respectively), and successively (re-) used over the whole time integration interval shown. On a 2.0 GHz MacBook our strategy gives an average CPU time of 9.3 s/timestep when running amg–ILU with the relaxed tolerance (corresponding to the data labeled "×" in the left-hand plot of Fig. 4.6), but only 6.6 seconds/timestep for the amg–ILU results in the right-hand plot. The amg–PDJ preconditioning strategy (corresponding to "○") is a little less efficient – it requires about twice as many iterations to satisfy the GMRES tolerance, which translates to 17.5 and 13.2 s/timestep for the relaxed and the tight accuracy tolerance respectively.

5. Concluding remarks

We have developed a fast and robust computational strategy for finding the numerical solution of models of bouyancy driven flow using implicit methods. The algorithm is constructed using a time-stepping strategy built from a stabilized trapezoidal rule, in conjunction with a specialized linearization strategy that achieves second-order accuracy in time and has the feature that, despite the nonlinearity of the Navier–Stokes equations, requires only one linear system solution at each time step. Moreover, this system can be solved in essentially optimal order complexity using algebraic multigrid methods. We have demonstrated the effectness of this solution algorithm on a series of benchmark problems, where it is shown in particular that very little preliminary knowledge of problem structure or parameter tuning is needed to efficiently compute accurate solutions.

References

[1] J. Boyle, M. Mihajlović, J. Scott, HSL_MI20: an efficient AMG preconditioner for finite element problems in 3D, Int. J. Numer. Methods Eng. 82 (2010) 64–98.
[2] M.A. Christon, P.M. Gresho, S.B. Sutton, Computational predictability of time-dependent natural convection flows in enclosures (including a benchmark solution), Int. J. Numer. Methods Fluid. 40 (2002) 953–980.
[3] H. Damanik, J. Hron, A. Ouazzi, S. Turek, A monolithic FEM-multigrid solver for non-isothermal incompressible flow on general meshes, J. Comput. Phys. 228 (2009) 3869–3881.
[4] D. Davis, E. Bänsch, An operator-splitting finite-element approach to the 8:1 thermal-cavity problem, Int. J. Numer. Methods Fluid. 40 (2002) 1019–1030.
[5] W. Dettmer, D. Perić, An analysis of the time integration algorithms for the finite element solutions of incompressible Navier–Stokes equations based on a stabilised formulation, Comput. Methods Appl. Mech. Eng. 192 (2003) 1177–1226.
[6] P.G. Drazin, Introduction to Hydrodynamic Stability, Cambridge University Press, Cambridge, 2002, ISBN 0-521-00965-0.
[7] H.C. Elman, D.J. Silvester, A.J. Wathen, Finite Elements and Fast Iterative Solvers: with Applications in Incompressible Fluid Dynamics, Oxford University Press, Oxford, 2005, ISBN 978-0-19-852868-5 (xiv+400, 0-19-852868-X).
[8] H.C. Elman, R.S. Tuminaro, Boundary conditions in approximate commutator preconditioners for the Navier–Stokes equations, Electron. Trans. Numer. Anal. 35 (2009) 257–280.
[9] A.Y. Gelfgat, P.Z. Bar-Joseph, A.L. Yarin, Stability of multiple steady states of convection in laterally heated cavities, J. Fluid Mech. 388 (1999) 315–334.
[10] P.M. Gresho, D.F. Griffiths, D.J. Silvester, Adaptive time-stepping for incompressible flow; part I: scalar advection–diffusion, SIAM J. Sci. Comput. 30 (2008) 2018–2054.

[11] P.M. Gresho, R.L. Sani, Incompressible Flow and the Finite Element Method, Isothermal Laminar Flow, vol. 2, John Wiley, Chichester, 2000.
[12] M.D. Gunzburger, Finite Element Methods for Viscous Incompressible Flows, Academic Press, London, 1989, ISBN 0-12-307350-2.
[13] D.A. Kay, P.M. Gresho, D.F. Griffiths, D.J. Silvester, Adaptive time-stepping for incompressible flow; part II: Navier–Stokes equations, SIAM J. Sci. Comput. 32 (2010) 111–128.
[14] W. Layton, Introduction to the Numerical Analysis of Incompressible Viscous Flow, SIAM, Philadelphia, 2008, ISBN 978-0-898716-57-3.
[15] B. Roux (Ed.), Numerical Simulation of Oscillatory Convection in a Low Prandtl Number Fluid, Notes on Numerical Fluid Mechanics, vol. 27, Vieweg, 1990.
[16] D.J. Silvester, H.C. Elman, A. Ramage, Incompressible Flow and Iterative Solver Software (IFISS) version 3.1, January 2011. Available from: <http://www.manchester.ac.uk/ifiss>.
[17] J.C. Simo, F. Armero, Unconditional stability and long-term behaviour of transient algorithms for the incompressible Navier–Stokes and Euler equations, Comput. Methods Appl. Mech. Eng. 111 (1994) 111–154.
[18] C.A. Smethurst, A finite element solution of the natural convection problem in 3D, Ph.D. Thesis, University of Manchester, 2010.
[19] B. Straughan, The Energy Method, Stability, and Nonlinear Convection, second ed., Springer-Verlag, New York, 2004, ISBN 0-387-00453-X.
[20] M. ur Rehman, C. Vuik, G. Segal, SIMPLE-type preconditioners for the Oseen problem, Int. J. Numer. Methods Fluid 61 (2009) 432–452.
[21] K.H. Winters, Oscillatory convection in liquid metals in a horizontal temperature gradient, Int. J. Numer. Methods Eng. 25 (1988) 401–414.
[22] S. Xin, P. LeQuéré, An extended Chebyshev pseudo-spectral benchmark for the 8:1 differentially heated cavity, Int. J. Numer. Methods Fluid 40 (2002) 981–998.