

*A framework for the development of implicit  
solvers for incompressible flow problems*

Silvester, David and Bespalov, Alexei and Powell,  
Catherine E.

2011

MIMS EPrint: **2011.104**

Manchester Institute for Mathematical Sciences  
School of Mathematics

The University of Manchester

Reports available from: <http://eprints.maths.manchester.ac.uk/>

And by contacting: The MIMS Secretary  
School of Mathematics  
The University of Manchester  
Manchester, M13 9PL, UK

ISSN 1749-9097

## A FRAMEWORK FOR THE DEVELOPMENT OF IMPLICIT SOLVERS FOR INCOMPRESSIBLE FLOW PROBLEMS

DAVID SILVESTER

School of Mathematics, The University of Manchester  
Manchester M13 9PL, UK

ALEX BESPALOV AND CATHERINE POWELL

School of Mathematics, The University of Manchester  
Manchester M13 9PL, UK

**ABSTRACT.** This survey paper reviews some recent developments in the design of robust solution methods for the Navier–Stokes equations modelling incompressible fluid flow. There are two building blocks in our solution strategy. First, an implicit time integrator that uses a stabilized trapezoid rule with an explicit Adams–Bashforth method for error control, and second, a robust Krylov subspace solver for the spatially discretized system. Numerical experiments are presented that illustrate the effectiveness of our generic approach. It is further shown that the basic solution strategy can be readily extended to more complicated models, including unsteady flow problems with coupled physics and steady flow problems that are nondeterministic in the sense that they have uncertain input data.

**1. Background and context.** Our starting point is the Navier–Stokes equations for an incompressible fluid with a fixed viscosity parameter  $\nu > 0$ . The fluid moves inside a domain  $\Omega \subset \mathbb{R}^d$  and the velocity is nonzero on some portion of the boundary  $\Gamma$ . The goal is to compute the fluid velocity  $\vec{u} : \Omega \rightarrow \mathbb{R}^d$  and the pressure  $p : \Omega \rightarrow \mathbb{R}$  satisfying the Navier–Stokes equations,

$$\frac{\partial \vec{u}}{\partial t} - \nu \nabla^2 \vec{u} + \vec{u} \cdot \nabla \vec{u} + \nabla p = \vec{0} \quad \text{in } W \equiv \Omega \times (0, \tau], \quad (1)$$

$$\nabla \cdot \vec{u} = 0 \quad \text{in } W. \quad (2)$$

The initial condition will always be associated with the quiescent state,

$$\vec{u}(\vec{x}, 0) = \vec{u}_0(\vec{x}) \equiv \vec{0}, \quad \vec{x} \in \Omega \cup \Gamma = \overline{\Omega}, \quad (3)$$

and the system (1)–(2) is integrated up to the final time  $\tau$ . Note that  $\vec{u}_0$  trivially satisfies the incompressibility constraint; that is,  $\nabla \cdot \vec{u}_0 = 0$  everywhere in  $\overline{\Omega}$ .

For ease of exposition we will only consider two-dimensional flow models ( $\Omega \subset \mathbb{R}^2$ ) herein. Our solver methodology is unchanged in the case of three-dimensional flow models. We also focus exclusively on inflow–outflow configurations, where the boundary comprises two nonoverlapping segments  $\Gamma_D \cup \Gamma_N$  associated with a

---

2000 *Mathematics Subject Classification.* Primary: 65M22, 76D05, 76D05; Secondary: 76M10.

*Key words and phrases.* Navier–Stokes, Boussinesq, finite element approximation, stochastic Galerkin approximation, implicit time stepping, preconditioning, algebraic multigrid.

specified time-dependent inlet flow field  $\vec{g}$  (set to zero at a fixed wall) and a standard natural outflow boundary condition,

$$\vec{u}(\cdot, t) = \vec{g}(t) \quad \text{on } \Gamma_D, \quad t \in (0, \tau], \quad (4)$$

$$\nu \nabla \vec{u} \cdot \vec{n} - p \vec{n} = \vec{0} \quad \text{on } \Gamma_N \times (0, \tau]. \quad (5)$$

We will insist that  $\Gamma_N$  has a nonzero length. This ensures that the pressure  $p$  is uniquely specified (for all time  $t \in (0, \tau]$ ) by the outflow boundary condition (5). In the special case of fully-developed parallel flow, we have  $\vec{u} \cdot \vec{t} = 0$  and the pressure at the outflow boundary will automatically be set to zero (see Elman et al. [10, p. 216] for further discussion).

Even the simplest advection–diffusion problems exhibit multiple timescales; see Gresho et al. [14]. This means that variable step time integrators are essential if such problems are to be efficiently solved computationally. A conventional strategy for time stepping the initial-value problem (1)–(5) is to use a semi-implicit time integration scheme with the nonlinear convection term treated explicitly; see, e.g. Ascher et al. [1]. This leads to a relatively simple (symmetric indefinite) linear algebra system at every time level, but there is a CFL stability restriction on the maximum time step size. In contrast, there is no time-step restriction if one uses a linearized implicit approach. The latter choice enables the possibility of including self-adaptive time-step control: time steps can be automatically selected to efficiently follow the physics. There is, however, a price to pay for this improved robustness: the linear algebra associated with solving linearized Navier–Stokes systems is much more of a challenge. Our thesis is that this challenge can be met using the preconditioning techniques for linearized Navier–Stokes systems that have emerged over the last decade. Details can be found in the original papers of Elman, Silvester, Wathen and their collaborators, specifically [9], [17] and [21].

The effectiveness of a *stabilized* TR–AB2 time-stepping algorithm (linearized trapezoid rule with a second-order explicit Adams–Bashforth method for error control) algorithm is established by Kay et al. [16]. The (in)stability of unstabilized TR for the incompressible Navier–Stokes equations is extensively discussed in the literature; see, e.g. Simo & Armero [23]. The basic TR algorithm does have some attractive features though—given the simplest ODE model of convection–diffusion,

$$\dot{y} = - \left( \frac{1}{\tau} + \mathbf{i}\omega \right) y, \quad y(0) = 1,$$

where  $\tau$  corresponds to a decay time constant and  $\omega$  is a frequency parameter, it is well known (see [14]) that TR is unconditionally stable (A-stable) and nondissipative. This is important when modelling pure advection ( $\tau = \infty$ ), or even advection-dominated problems ( $\frac{1}{\tau} \ll \omega$ ).

The linearization at each time level is an important aspect of our solution strategy. It is also mildly contentious. A widely-held view is that robustness and accuracy can only be maintained by respecting the fully implicit coupling in the convection term  $\vec{u} \cdot \nabla \vec{u}$ : for example, by solving the time-discretized nonlinear system to a predefined accuracy using fixed point iteration or some variant of Newton iteration. A recent paper by Damanik et al. [5] gives an efficient implementation of the latter approach. Suppose that the interval  $[0, \tau]$  is divided into  $N$  steps,  $\{t_i\}_{i=1}^N$ , and let  $\vec{u}^j$  denote the discrete (in time) approximation to  $\vec{u}(\cdot, t_j)$ . An alternative perspective is that the quadratic term  $\vec{u}^{n+1} \cdot \nabla \vec{u}^{n+1}$  is well approximated by the

explicit “second-order” convection field  $\vec{w}^{n+1} \cdot \nabla \vec{u}^{n+1}$  with

$$\vec{w}^{n+1} = (1 + (k_{n+1}/k_n)) \vec{u}^n - (k_{n+1}/k_n) \vec{u}^{n-1}, \quad (6)$$

where  $k_{n+1} := t_{n+1} - t_n$  is the current time step and  $k_n$  is the previous one. Such a linearization was originally proposed in the context of constant trapezoid rule time-stepping by Simo & Armero [23] and is mathematically equivalent to the TRLE algorithm discussed by Layton [18, p. 163]. Using the strategy (6) gives a more straightforward algorithmic structure: there is no need for an “inner-outer” iteration since we only solve a *single* linear system at each time level.

Spatial discretization is equally important but it is not our primary focus. It will be accomplished herein using a method-of-lines approach based on a stable, conforming, mixed finite element approximation of (1)–(2), (see Elman et al. [10, Chapter 5] for further discussion of *inf-sup* stability). We use fixed grids of rectangular elements in this work and we present results obtained using Taylor–Hood  $Q_2-Q_1$  (biquadratic velocity, bilinear continuous pressure) approximation. We denote the grid subdivision by  $\mathcal{T}_h$  and we associate the parameter  $h$  with the length of the longest edge in the grid. Then defining finite element solution spaces  $\mathbf{X}_E^h$  (interpolating the velocity boundary data  $\vec{g}(t_k)$  on the inflow boundary  $\Gamma_D$  at discrete time steps,  $k = 1, 2, \dots$ ) and  $M^h \subset L^2(\Omega)$ , the fully discrete problem to be solved at each distinct time level is: given a time step  $k_{n+1} := t_{n+1} - t_n$  and approximations  $\vec{u}_h^n$  and  $p_h^n$  at time  $t_n$ , compute  $\vec{u}_h^{n+1} \in \mathbf{X}_E^h$  and  $p_h^{n+1} \in M^h$  satisfying the Galerkin formulation,

$$\begin{aligned} \frac{2}{k_{n+1}}(\vec{u}_h^{n+1}, \vec{v}_h) + \nu(\nabla \vec{u}_h^{n+1}, \nabla \vec{v}_h) + (\vec{w}_h^{n+1} \cdot \nabla \vec{u}_h^{n+1}, \vec{v}_h) - (p_h^{n+1}, \nabla \cdot \vec{v}_h) \\ = \frac{2}{k_{n+1}}(\vec{u}_h^n, \vec{v}_h) + \left(\frac{\partial \vec{u}_h^n}{\partial t}, \vec{v}_h\right), \end{aligned} \quad (7)$$

$$(\nabla \cdot \vec{u}_h^{n+1}, q_h) = 0, \quad (8)$$

for all test functions  $\vec{v}_h \in \mathbf{X}_0^h$ ,  $q_h \in M^h$ , where  $(\cdot, \cdot)$  represents the  $L^2(\Omega)$  inner product. Note that the term  $\frac{\partial \vec{u}_h^n}{\partial t} := \nu \nabla^2 \vec{u}_h^n - \vec{u}_h^n \cdot \nabla \vec{u}_h^n - \nabla p_h^n$  is just shorthand for the acceleration at time step  $t_n$ , and the convection term is linearized via (6), so that  $\vec{w}_h^{n+1} = (1 + (k_{n+1}/k_n)) \vec{u}_h^n - (k_{n+1}/k_n) \vec{u}_h^{n-1}$ .

The TR–AB2 algorithm is *stabilized* using a simple time-step averaging technique. Full details are given in [14] and [16]. The averaging is invoked periodically every  $n_*$  steps. For such a step, we save the values  $t_* = t_n$ ,  $\vec{u}_h^* = \vec{u}_h^n$  and having computed the scaled TR velocity update,  $\vec{d}_h^n := (\vec{u}_h^{n+1} - \vec{u}_h^n)/k_{n+1}$ , we set  $t_n = t_{n-1} + \frac{1}{2}k_n$ ,  $t_{n+1} = t_* + \frac{1}{2}k_{n+1}$  and define shifted solutions via the updates:

$$\vec{u}_h^n = \frac{1}{2}(\vec{u}_h^* + \vec{u}_h^{n-1}), \quad \frac{\partial \vec{u}_h^n}{\partial t} = \frac{1}{2} \left( \frac{\partial \vec{u}_h^n}{\partial t} + \frac{\partial \vec{u}_h^{n-1}}{\partial t} \right), \quad (9)$$

$$\vec{u}_h^{n+1} = \vec{u}_h^* + \frac{1}{2}k_{n+1} \vec{d}_h^n, \quad \frac{\partial \vec{u}_h^{n+1}}{\partial t} = \vec{d}_h^n. \quad (10)$$

Unless stated otherwise, the averaging frequency parameter is fixed:  $n_* = 10$ . We let  $\varepsilon_t$  denote the user-specified tolerance for the time accuracy. Once we have an error estimate  $\|\vec{e}_h^{n+1}\|$  (the  $L^2$  norm of the difference between the TR and AB2 approximations to  $\vec{u}(t_{n+1})$ ), the subsequent time step  $k_{n+2}$  is computed using the heuristic formula

$$k_{n+2} = k_{n+1} \left( \varepsilon_t / \|\vec{e}_h^{n+1}\| \right)^{1/3}. \quad (11)$$

In the remainder of this introductory section we will quickly review the concepts of pressure convection–diffusion preconditioning and least-squares commutator preconditioning in the context of the discrete formulation (7)–(8). First, to get to linear algebra, we need specific basis sets for the approximation spaces:

$$\begin{aligned} \mathbf{X}_0^h &= \text{span} \left\{ \left[ \begin{array}{c} \phi_i \\ 0 \end{array} \right], \left[ \begin{array}{c} 0 \\ \phi_i \end{array} \right] \right\}_{i=1}^{n_u} \text{ for velocity, and} \\ M^h &= \text{span}\{\psi_j\}_{j=1}^{n_p} \text{ for pressure.} \end{aligned}$$

Next, as discussed in [14] and [16], our TR–AB2 implementation explicitly computes discrete velocity updates  $\vec{d}_h^n$  that are scaled by the time step so as to avoid underflow and inhibit subtractive cancellation. The current velocity solution is thus given by

$$\vec{u}_h^{n+1} = \vec{u}_h^n + k_{n+1} \vec{d}_h^n. \quad (12)$$

Then, given the expansions

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

the coefficient vectors  $\boldsymbol{\alpha}^{u,n+1} = [\boldsymbol{\alpha}^{x,n+1}, \boldsymbol{\alpha}^{y,n+1}]$ ,  $\boldsymbol{\alpha}^{p,n+1}$  may be computed by solving the saddle-point system associated with (7)–(8); that is,

$$\begin{pmatrix} \mathbf{F}_\nu^{n+1} & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{\alpha}^{u,n+1} \\ \boldsymbol{\alpha}^{p,n+1} \end{pmatrix} = \begin{pmatrix} \mathbf{f}^{u,n+1} \\ \mathbf{0} \end{pmatrix}. \quad (13)$$

Note that we have divided the incompressibility constraint by the time step  $k_{n+1}$  to preserve the div–grad block symmetry. The matrix  $B = [B_x, B_y]$  is the discrete divergence operator,

$$B_x := [B_x]_{ji} = - \left( \psi_j, \frac{\partial \phi_i}{\partial x} \right), \quad j = 1, \dots, n_p, \quad i = 1, \dots, n_u, \quad (14)$$

$$B_y := [B_y]_{ji} = - \left( \psi_j, \frac{\partial \phi_i}{\partial y} \right), \quad j = 1, \dots, n_p, \quad i = 1, \dots, n_u. \quad (15)$$

A consequence of the inflow–outflow boundary conditions is that the matrix  $B^T$  has full rank. This means that the system (13) is *nonsingular*. The matrix  $\mathbf{F}_\nu^{n+1}$  in (13) is the discrete convection–diffusion–reaction operator:

$$\mathbf{F}_\nu^{n+1} := 2\mathbf{M} + \nu k_{n+1} \mathbf{A} + k_{n+1} \mathbf{N}(\vec{w}_h^{n+1}). \quad (16)$$

The matrices  $\mathbf{M}$ ,  $\mathbf{A}$  and  $\mathbf{N}$  in (16) are all block-diagonal matrices with (scalar) components,

$$M := [M]_{ij} = (\phi_i, \phi_j), \quad i, j = 1, \dots, n_u, \quad (17)$$

$$A := [A]_{ij} = (\nabla \phi_i, \nabla \phi_j), \quad i, j = 1, \dots, n_u, \quad (18)$$

$$N(\vec{w}_h) := [N]_{ij} = (\vec{w}_h \cdot \nabla \phi_i, \phi_j) \quad i, j = 1, \dots, n_u. \quad (19)$$

Our solution algorithm is right-preconditioned GMRES with a preconditioner that is specially tailored to the structured matrix (13). To illustrate the approach, we express system (13) (omitting the matrix subscripts/superscripts) with a preconditioner  $\mathcal{P}$  so that

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

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

$$\mathcal{P} := \begin{pmatrix} \mathbf{F} & B^T \\ 0 & -S \end{pmatrix} \approx \begin{pmatrix} \mathbf{F} & B^T \\ B & 0 \end{pmatrix}, \quad (20)$$

is motivated by the identity

$$\begin{pmatrix} \mathbf{F} & B^T \\ B & 0 \end{pmatrix} \underbrace{\begin{pmatrix} \mathbf{F}^{-1} & \mathbf{F}^{-1}B^TS^{-1} \\ 0 & -S^{-1} \end{pmatrix}}_{\mathcal{P}^{-1}} \equiv \begin{pmatrix} I & 0 \\ B\mathbf{F}^{-1} & I \end{pmatrix}. \quad (21)$$

This shows that the preconditioned matrix has all eigenvalues clustered at unity. It can be shown that the matrix on the right-hand side has Jordan blocks of dimension two, which implies that GMRES applied to (13) with right preconditioner  $\mathcal{P}$  will converge in precisely two iterations, independent of the convection field  $\bar{w}_h^{n+1}$  and the values of the parameters  $h$ ,  $k_{n+1}$  and  $\nu$ .

It follows from (21) 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 Schur complement matrix  $S$ , second, we perform a matrix–vector multiply with  $B^T$ , and finally we solve the two scalar systems associated with the matrix operator  $\mathbf{F}$ . 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 [10, 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 a matrix  $F_*$  (obtained by constructing the operator  $\mathbf{F}_\nu^{n+1}$  in (16) with velocity basis functions replaced by pressure basis functions), together with linear solves for a pressure diffusion matrix  $A_*$  and a pressure mass matrix  $Q_*$ . In this paper, we follow the suggestion of Elman & Tuminaro [11] and implement PCD via

$$S^{-1} = (B\mathbf{F}^{-1}B^T)^{-1} \approx Q_*^{-1} F_* A_*^{-1}, \quad (22)$$

where  $Q_*$  is the diagonal of the pressure mass matrix  $Q$ , and  $A_*$  is the operator  $BM_*^{-1}B^T$  where  $M_*$  is the diagonal of the velocity mass matrix  $M$ . 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

$$S^{-1} = (B\mathbf{F}^{-1}B^T)^{-1} \approx A_*^{-1}(BM_*^{-1}\mathbf{F}M_*^{-1}B^T)A_*^{-1}. \quad (23)$$

As discussed above, in a practical implementation, the action of  $A_*^{-1}$  in (22) and (23) will be performed inexactly using AMG. Preconditioning with the exact Schur complement approximations (22) and (23) and with the original matrix operator  $\mathbf{F}_\nu^{n+1}$  in  $\mathcal{P}^{-1}$  will be referred to as *exact* PCD and *exact* LSC preconditioning, respectively.

An outline of the rest of the paper is as follows. Section 2 introduces a model flow problem (flow in a channel over a backward-facing step) that will be used to demonstrate the effectiveness of our solver methodology. Our computational experiments will demonstrate that temporal stability is not compromised using the linearization (6). Traditional wisdom is that a coupled solver is mainly of use for computing steady flows, whereas projection-type schemes are usually preferred

when modelling unsteady flows. We want to challenge this perception. To this end, the basic flow model is extended to include buoyancy effects in Section 3. We will demonstrate that it is straightforward to generalize the solver methodology to the coupled Boussinesq model, and we will see that having an adaptive time integrator is crucial if we are to follow the coupled physics effectively and efficiently. This material complements and extends the computational results in the recent paper [7]. All the numerical results from Sections 2 and 3 can be reproduced by a diligent reader using our IFISS software toolbox [22]. The concluding section is more speculative. We discuss the extension of our preconditioning methodology to a modelling situation whereby the flow is steady but the volume of flow in the channel is not known precisely. We intend to provide a more comprehensive treatment of this problem in the follow-up publication [2].

**2. Steady flow over a step.** The flow inside a rectangular duct with a sudden expansion is a classical test problem. The domain geometry and boundary conditions are illustrated in Figure 1. The re-entrant corner is positioned at the origin  $(0, 0)$ . We set the viscosity parameter to be  $\nu = 1/50$  and we fix the dimensions so that  $d = 1$  and  $L = 5$ . A time-dependent, fully-developed (Poiseuille) flow profile,  $\vec{g} = (4y(d - y)(1 - e^{-10t})/d^2, 0)$ , is imposed on the inflow boundary,  $\Gamma_{\text{in}}$ . This models a smooth start-up from the quiescent state. A steady no-flow condition is imposed on  $\Gamma_{\text{wall}}$ . At the outflow boundary,  $\Gamma_{\text{out}}$ , the natural condition (5) forces the mean outflow pressure to be zero for all time  $t \in [0, \tau]$ .

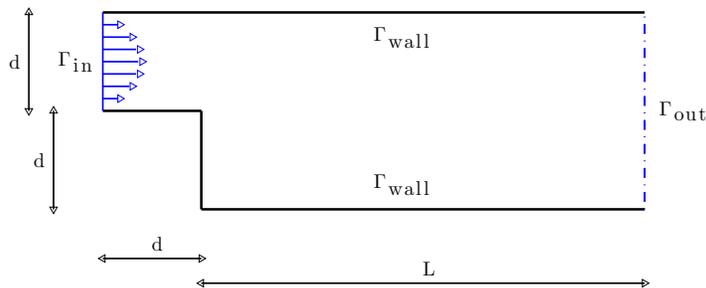


FIGURE 1. The backward-facing step domain.

The singularity in the solution at the re-entrant corner is a very important feature of this flow problem. A key point is that the Stokes equations are a good approximation to the Navier–Stokes equations in the neighbourhood of the corner. Moreover, small length-scale asymptotics show that the Stokes pressure solution behaves like  $r^{-\lambda}$ , where  $r$  is the radial distance from the corner and  $\lambda > 0$ . This means that derivatives of the (Stokes-flow) velocity—as well as derived quantities like vorticity—are *unbounded* at the corner point for all time  $t \in [0, \tau]$ . The upshot is that spatial discretization needs to be done carefully. To respect the singularity we will show some results that are computed using a nonuniform grid of 1536 elements.<sup>1</sup> The grid details are shown in Figure 2. The matrix dimensions in (13) are  $n_u = 6321$ ,  $n_p = 1625$ , so the overall system dimension is 14267. Note that there are highly stretched elements (with aspect ratios of 15:1) in both coordinate

<sup>1</sup>The grid is generated in IFISS 3.2 by running `newstep_domain.m` with the grid parameter set to 6 and the stretch parameter set to 1.2.

directions along the lines  $x = 0$  and  $y = 0$ . Such grids are known to provide a stiff test for solver strategies that employ algebraic multigrid.

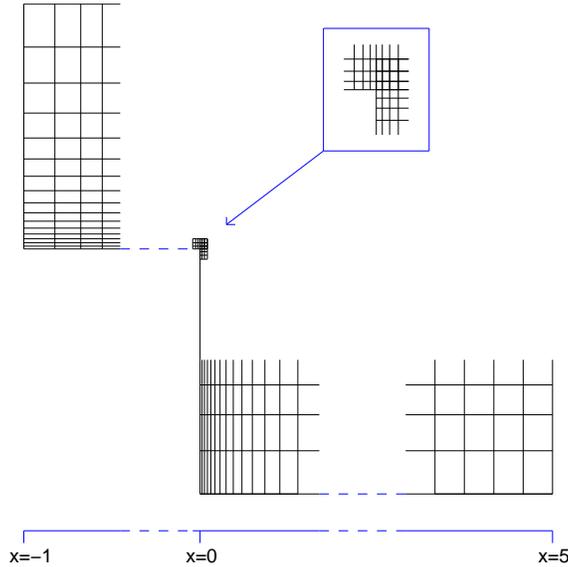


FIGURE 2. Grid details (plotted to scale) showing the refinement in the vicinity of the corner (magnified  $\times 3$  in the zoom).

The proposed flow problem is expected to have a stable steady-state solution. A standard nondimensionalization (see Gresho et al. [13]) of the configuration is to define the reference length  $\bar{L}$  to be the inlet length  $d$  (see Figure 1), and the reference velocity  $\bar{U}$  to be the average inlet velocity,  $V := \int_0^d g_x dy$ . Note that  $V$  equals the volume of fluid flowing into the channel in the case  $d = 1$ . With our scaling of the inflow,  $\bar{U}$  tends to  $2/3$  as  $t \rightarrow \infty$ , and the Reynolds number increases over time to a limiting value of  $\text{Re}^\infty = 3/(2\nu) = 75$ . This value is much smaller than the critical Reynolds number  $\text{Re}^* \sim 800$  (see Gresho et al. [13]), where the steady flow bifurcates to an unsteady (periodic) flow. Note that a longer outlet channel,  $L \gg 5$ , needs to be specified if we are to compute flows at higher Reynolds numbers. We will return later to the issue of choosing  $L$  appropriately.

The time-step evolution of the stabilized TR-AB2 integrator with a temporal tolerance,  $\varepsilon_t = 3 \times 10^{-5}$ , is shown in Figure 3. The initial time step is  $10^{-10}$  but is omitted from the left-hand plot to give a clearer picture of the scale of the increasing time step as the solution evolves to a steady state. The right-hand plot shows the first 79 time steps in better detail. There are no rejected time steps. Looking at the left-hand plot we see a rapid transient for the first 10–20 time steps, whereby the time step grows to a magnitude of  $10^{-2}$ . This is followed by a smooth evolution as the flow solution goes to a steady state. The final time,  $\tau = 200$ , is reached after only 150 time steps!

Computed streamlines and vorticity contours at three snapshot times are shown in Figures 4 and 5. We see that the flow develops into a single recirculation zone downstream of the step. The steady-state flow is visually identical to that shown in the bottom plot in Figure 4. At early times, the vorticity field illustrated in Figure 5

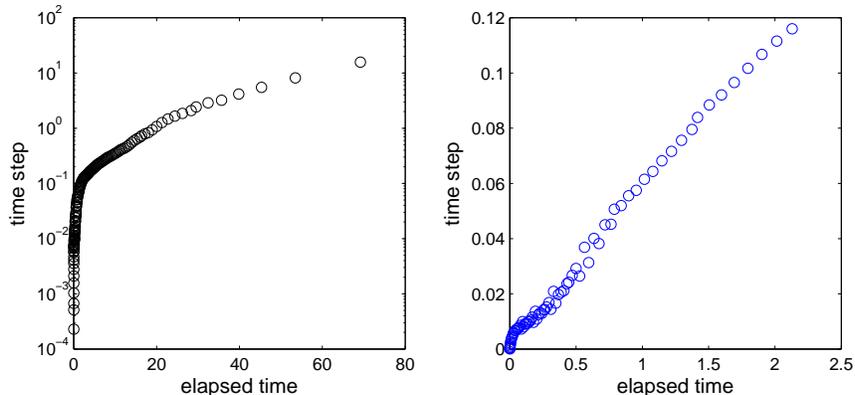


FIGURE 3. Time-step evolution (left) for flow over a step using stabilized TR-AB2 with zoom (right) of the first 79 time steps.

develops potential flow “sheets” along the rigid walls. This is an important and difficult-to-model feature of impulsively-started viscous flow. A detailed discussion can be found in the book by Gresho & Sani [15, Section 3.19]. As anticipated, the singularity at the origin is a prominent and persistent feature.

Turning now to linear algebra aspects, convergence history plots for LSC and PCD preconditioning are shown in Figure 6. We show GMRES convergence curves associated with the linear systems (13) that arise at three distinct time levels in the integration run. These three steps correspond to the flow solutions at approximate times  $t \sim 1$ ,  $t \sim 5$  and  $t \sim 100$  that are shown in Figure 4. The local time steps ( $k_{n+1}$  in (16)) are 0.058, 0.192 and 26.1 time units respectively. The trend is clear—the smaller the time step, the faster the convergence of preconditioned GMRES. There is little to choose between the performance of exact LSC and exact PCD preconditioning.

The inexact results are remarkable. We make use of the MATLAB version of the code HSL\_MI20 [4] and replace each of the solves involving  $\mathbf{F}_\nu^{n+1}$  and  $A_*$  in (22) and (23) with a single V-cycle of AMG. The key to the overall robustness is a clever *hybrid* smoothing strategy<sup>2</sup>: on the finest level, one sweep of ILU(0) with a left  $\rightarrow$  right and bottom  $\rightarrow$  top point ordering is performed; on all coarser levels a simple 2–2 (pre-smoothing–post-smoothing) V-cycle using point-damped Jacobi (PDJ) with damping parameter  $\omega = 0.5$  is used. For a realistic residual reduction factor of  $10^{-6}$  the inexact version of PCD always converges in 10–20 iterations. The strategy is extremely robust—it is effective even when arbitrarily large time steps are taken.

We can check the validity of our numerical results by rerunning the stabilized TR-AB2 integrator: first, keeping the same spatial resolution but taking a longer computational domain,  $L = 20$ ; and second, by keeping  $L = 5$  but using a much finer spatial grid. For the lengthened domain we take the grid in Figure 2 and extend it uniformly in the  $x$  direction as far as the repositioned outlet at  $L = 20$ . The extended grid has 4416 rectangular elements and the dimension of the discrete system (13) is increased to 40637. Retaining the temporal tolerance,  $\varepsilon_t = 3 \times 10^{-5}$ , the target time  $\tau = 200$  is reached after 173 time steps. The  $L = 20$  time steps are

<sup>2</sup>This is the default AMG smoothing strategy in the IFISS software.

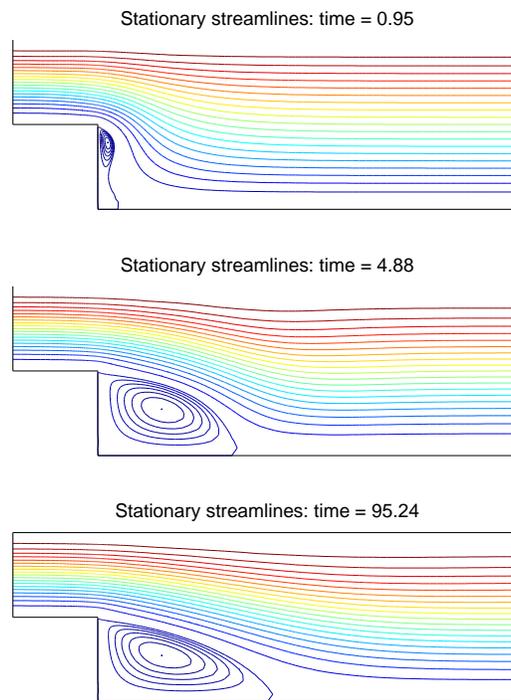


FIGURE 4. Flow over a step: snapshots of stationary streamlines.

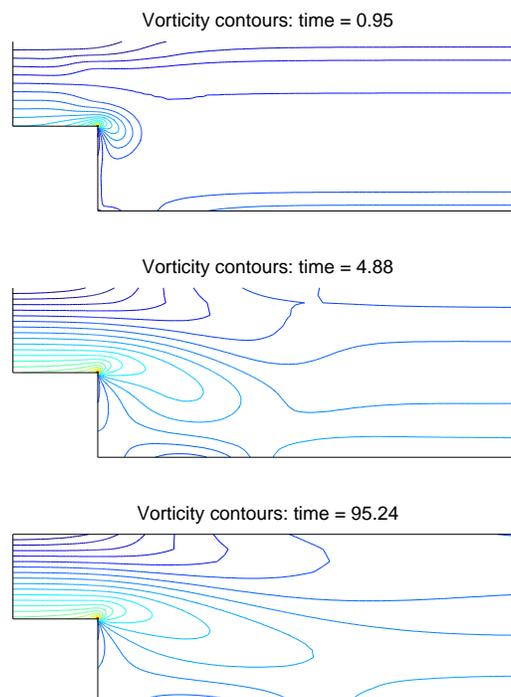


FIGURE 5. Flow over a step: snapshots of the vorticity evolution.

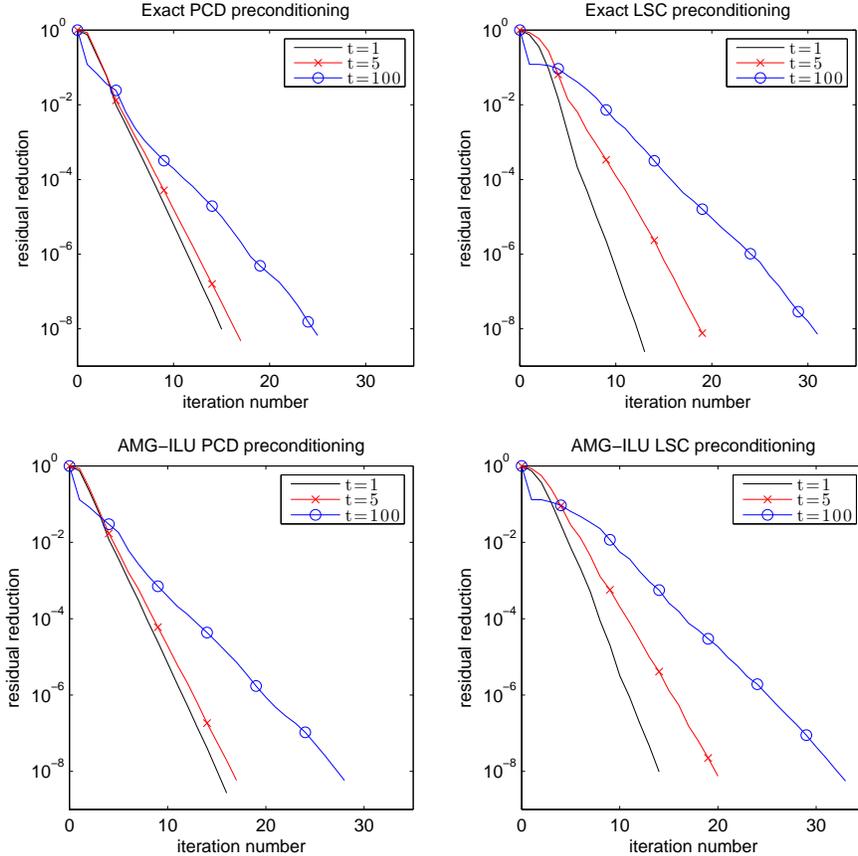


FIGURE 6. GMRES convergence using exact preconditioning (top) and inexact (AMG-ILU) preconditioning (bottom).

visually identical to the  $L = 5$  time steps shown in the right-hand plot of Figure 3, but, as might be expected, the evolution to the steady state for  $t > 10$  takes a little longer on the extended domain. If we compare the stationary streamlines plotted in Figure 7 with those in Figure 4, it is readily seen that the computed solutions on the extended domain are in very close agreement with those computed on the original domain.

We also observe grid-independent physics if we refine the original spatial grid. To show this, we fix  $L = 5$  and generate a new stretched grid consisting of 5888 elements.<sup>3</sup> The new dimensions are  $n_u = 23897$  and  $n_p = 6061$  and the overall system dimension in (13) is increased to 53855. When we run our integrator with  $\varepsilon_t = 3 \times 10^{-5}$ , the time step behaviour is essentially identical to that in Figure 3, and the snapshot streamline and vorticity solutions are qualitatively similar to those in Figures 4 and 5.

Finally, to quantify the effect that grid refinement has on the rate of convergence of the linear solver, we show convergence history plots in Figure 8 for inexact LSC

<sup>3</sup>The refined grid is generated in IFISS 3.2 by running `newstep_domain.m` with the grid parameter set to 7 and the stretch parameter set to 1.1.

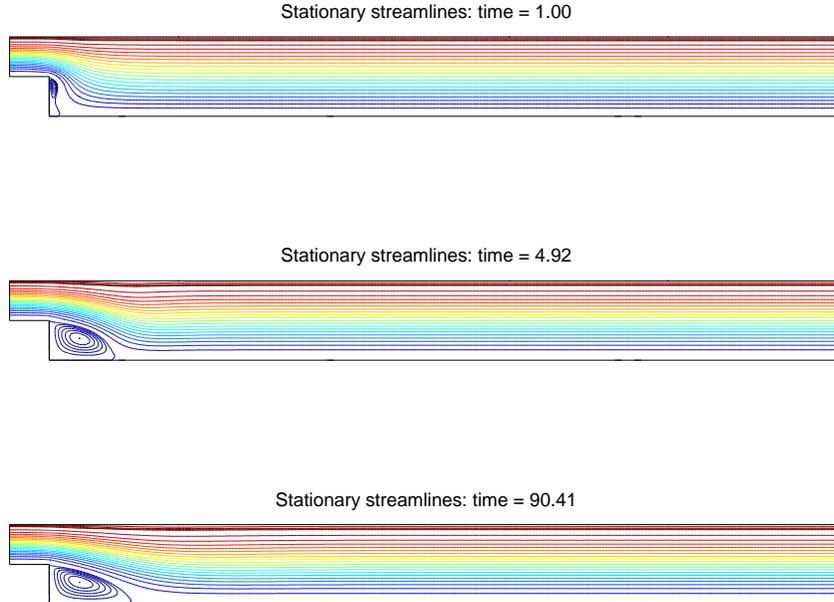


FIGURE 7. Flow over a step with an extended outlet: snapshots of stationary streamlines.

and PCD preconditioning of the linear systems that are solved at time  $t \sim 10$ . Note that local time steps  $k_{n+1}$  are not small at this snapshot time: 0.346 time units for the original grid, 0.343 for the lengthened grid and 0.340 for the refined grid, and it is not so surprising to see a small increase in the number of iterations when moving from the original grid to the refined grid. For a realistic residual reduction factor of  $10^{-6}$ , the inexact version of PCD converges in 15 iterations on the original and extended grids, and 18 iterations on the refined grid. We should also point out that the convergence curves in Figure 8 are not representative of the behaviour at earlier time levels. In particular, for the distinct time levels up to  $t \sim 1$ , the convergence of the alternative solver strategies is essentially independent of the specific grid.

**3. Flow over a step with buoyancy.** In this section of the paper, the flow domain is unchanged but the physical model is extended to accommodate an imposed (vertical) temperature gradient. Introducing the scalar temperature,  $T : \Omega \rightarrow \mathbb{R}$ , and including a transport equation, leads us to the model,

$$\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} - \nu \nabla^2 \vec{u} + \nabla p = \vec{j} T \quad \text{in } W \equiv \Omega \times (0, \tau], \quad (24)$$

$$\nabla \cdot \vec{u} = 0 \quad \text{in } W, \quad (25)$$

$$\frac{\partial T}{\partial t} + \vec{u} \cdot \nabla T - \epsilon \nabla^2 T = 0 \quad \text{in } W. \quad (26)$$

The Boussinesq system (24)–(26) arises as a limiting case of modelling the flow of a fluid forced by gravity (acting downwards 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 average value are allowed. Note that there are two different viscosity parameters in our model:  $\nu$  in (24) and  $\epsilon$  in (26). The

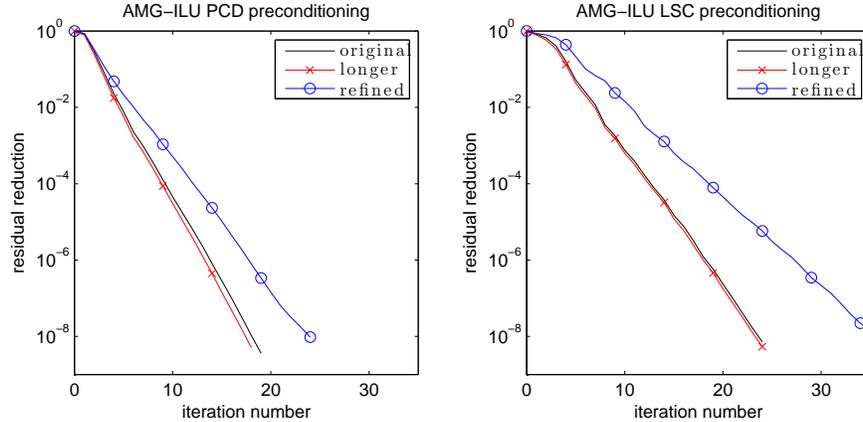


FIGURE 8. GMRES convergence for linear systems solved at time level  $t \sim 10$ . Results are shown for three different spatial discretizations of the flow domain.

precise definition of these viscosity parameters depends on the nondimensionalization used. In our case, we define  $\nu$  and  $\epsilon$  in terms of a Rayleigh number,  $Ra$ , and a Prandtl number,  $Pr$ , with

$$\nu = \sqrt{\frac{Pr}{Ra}}, \quad \epsilon = \frac{1}{\sqrt{Pr \cdot Ra}}.$$

The Prandtl number is a property of the fluid. Herein we will fix  $Pr = 7.1$ , which is a typical value for water at  $20^\circ\text{C}$ . The Rayleigh number 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. To provide a direct comparison with the isothermal model in the previous section we set the Rayleigh number to a fixed value,  $Ra = 17750$ . This implies that  $\nu = 1/50$  (exactly as in Section 2) and  $\epsilon \sim 0.0028$ .

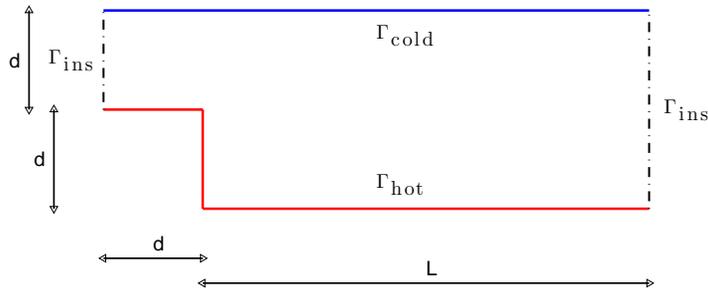


FIGURE 9. Boundary sections for temperature boundary conditions.

The fluid flow boundary conditions associated with the boundary sections in Figure 1 need to be supplemented by temperature boundary conditions. Three different

parts of the boundary are identified in Figure 9, and the associated conditions are

$$\left. \begin{aligned} T(\cdot, t) &= \frac{1}{2}(1 - e^{-10t}) && \text{on } \Gamma_{\text{hot}}, t \in (0, \tau], \\ T(\cdot, t) &= -\frac{1}{2}(1 - e^{-10t}) && \text{on } \Gamma_{\text{cold}}, t \in (0, \tau], \\ \epsilon \nabla T \cdot \vec{n} &= 0 && \text{on } \Gamma_{\text{ins}} \times (0, \tau], \end{aligned} \right\} \quad (27)$$

so that the temperature gradient across the channel tends to  $\pm \frac{1}{2}$  as  $t \rightarrow \infty$ . Note that the inflow and outflow temperature profiles are modelled with an insulating boundary condition. The initial condition is again the quiescent state:

$$\vec{u}(\vec{x}, 0) = \vec{0}, \quad T(\vec{x}, 0) = 0, \quad \vec{x} \in \overline{\Omega}. \quad (28)$$

We will solve the coupled flow problem (24)–(26) in the flow domain with length scales  $d = 1$  and  $L = 20$ . We note that the problem is not guaranteed to have a stable steady-state solution for these parameter choices. The thermal convection effects are significant because the value of  $\text{Ra} = 17750$  is bigger than the critical value for the onset of Rayleigh–Bénard convection rolls between parallel plates in an enclosed flow setting with separation distance  $2d = 2$ . (For further details, see Drazin [6, Chapter 6] and the numerical results in [7, Section 4.2]).

Full details of the Galerkin formulation with TR–AB2 time stepping can be found in Elman et al. [7]. We combine the basic  $Q_2$ – $Q_1$  velocity–pressure approximation used already, with standard  $Q_2$  approximation for the temperature; that is,  $X_0^h = \text{span}\{\phi_\ell\}_{\ell=1}^{n_T}$ . Using the standard linearization  $\vec{w}_h^{n+1} = (1 + (k_{n+1}/k_n))\vec{u}_h^n - (k_{n+1}/k_n)\vec{u}_h^{n-1}$  we are charged with computing  $(\vec{u}_h^{n+1}, p_h^{n+1}, T_h^{n+1}) \in \mathbf{X}_E^h \times M^h \times X_E^h$  satisfying the fully discrete formulation,

$$\begin{aligned} \frac{2}{k_{n+1}}(\vec{u}_h^{n+1}, \vec{v}_h) + \nu(\nabla \vec{u}_h^{n+1}, \nabla \vec{v}_h) + (\vec{w}_h^{n+1} \cdot \nabla \vec{u}_h^{n+1}, \vec{v}_h) - (p_h^{n+1}, \nabla \cdot \vec{v}_h) \\ - (T_h^{n+1}, \vec{j} \cdot \vec{v}_h) = \frac{2}{k_{n+1}}(\vec{u}_h^n, \vec{v}_h) + \left(\frac{\partial \vec{u}_h^n}{\partial t}, \vec{v}_h\right), \end{aligned} \quad (29)$$

$$(\nabla \cdot \vec{u}_h^{n+1}, q_h) = 0, \quad (30)$$

$$\begin{aligned} \frac{2}{k_{n+1}}(T_h^{n+1}, v_h) + \epsilon(\nabla T_h^{n+1}, \nabla v_h) + (\vec{w}_h^{n+1} \cdot \nabla T_h^{n+1}, v_h) \\ = \frac{2}{k_{n+1}}(T_h^n, v_h) + \left(\frac{\partial T_h^n}{\partial t}, v_h\right), \end{aligned} \quad (31)$$

for all  $(\vec{v}_h, q_h, v_h) \in \mathbf{X}_0^h \times M^h \times X_0^h$ . Note that the linearization *uncouples* the system so that it can, in principle, be solved by back-substitution: first, by computing  $T_h^{n+1}$  from (31), and second, by computing  $(\vec{u}_h^{n+1}, p_h^{n+1})$  satisfying (29)–(30).

Following [7], we will keep a fully-coupled approach and solve the “saddle-point system” associated with (29)–(31); that is,

$$\begin{pmatrix} \mathbf{F}_\nu^{n+1} & B^T & -\frac{1}{2}k_{n+1}\frac{0}{M} \\ B & 0 & 0 \\ 0 & 0 & F_\epsilon^{n+1} \end{pmatrix} \begin{pmatrix} \boldsymbol{\alpha}^{u,n+1} \\ \boldsymbol{\alpha}^{p,n+1} \\ \boldsymbol{\alpha}^{t,n+1} \end{pmatrix} = \begin{pmatrix} \mathbf{f}^{u,n+1} \\ \mathbf{0} \\ \mathbf{f}^{t,n+1} \end{pmatrix}, \quad (32)$$

where  $F_\epsilon^{n+1}$  is the scalar reaction–convection–diffusion operator defined on the temperature space and the matrix  $\frac{0}{M}$  represents the buoyancy term,

$$\frac{0}{M} := \left(\frac{0}{M}\right)_{ij} = ([0, \phi_i], \phi_j), \quad i = 1, \dots, n_u, \quad j = 1, \dots, n_T. \quad (33)$$

To facilitate a direct comparison with Section 2, we solve the coupled flow problem using the extended grid of 4416 rectangular elements. The component subspace dimensions are thus  $n_u = 36042$ ,  $n_p = 4595$ ,  $n_T = 18021$ , and the dimension of the discrete system (32) is 58658. As in Section 2, the stabilized TR–AB2 integrator

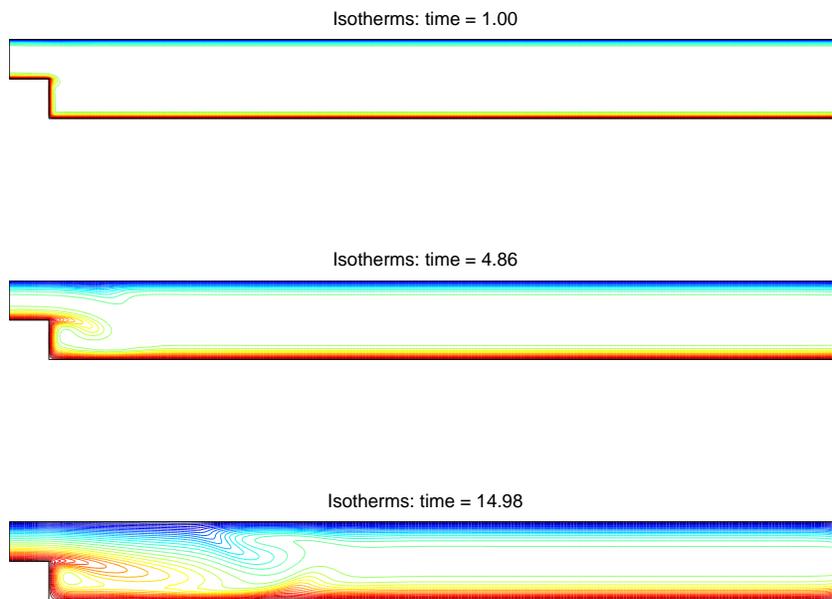


FIGURE 10. Flow over a heated step: isotherm snapshots at early times.

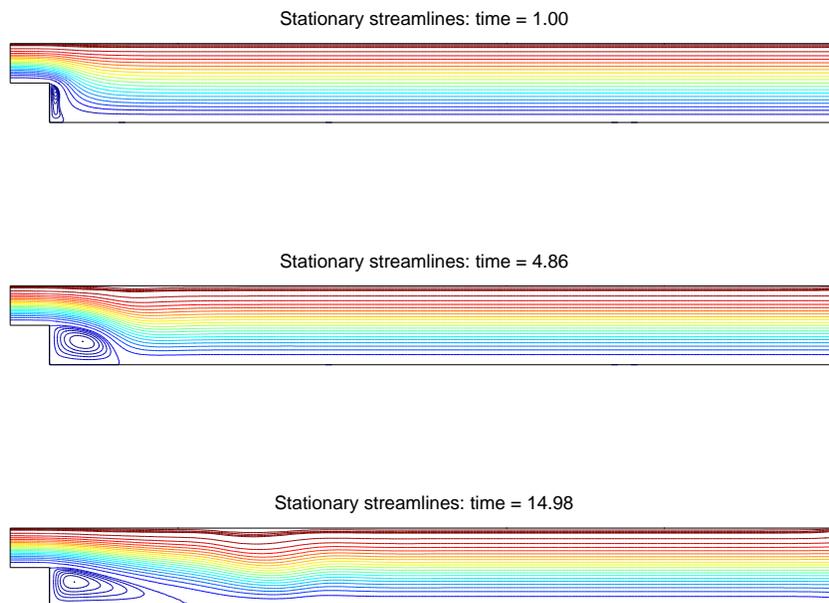


FIGURE 11. Flow over a heated step: snapshots of stationary streamlines at early times.

is run with the accuracy tolerance set to  $\varepsilon_t = 3 \times 10^{-5}$ . The evolution of the computed temperature field is shown in Figure 10. Initially, the vertical heating has little effect on the flow, but when  $t \sim 15$  we observe that the cold fluid is being convected from the top wall into the centre of the channel. This mixing of hot and cold fluid represents an unstable perturbation of the steady-state isothermal flow. If we compare the final-time plot in Figure 7 with that in Figure 11 we might suspect that a completely different dynamic has been set in motion.

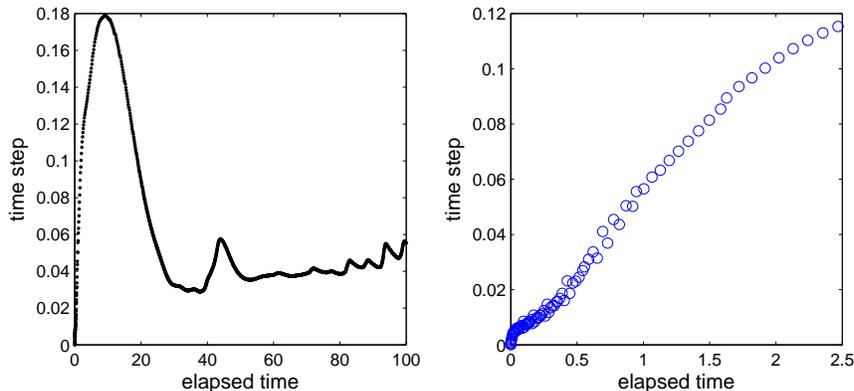


FIGURE 12. Time-step evolution (left) for flow over a heated step using stabilized TR–AB2 with zoom (right) of first 99 time steps.

The adaptive time-step selection is based on the coupled physics. Specifically,

$$k_{n+2} = k_{n+1} \left( \varepsilon_t / \sqrt{\|\bar{e}_h^{n+1}\|^2 + \|e_h^{n+1}\|^2} \right)^{1/3}, \quad (34)$$

where  $\|\bar{e}_h^{n+1}\|$  is the  $L^2$  norm of the difference between the TR and AB2 approximations to  $\bar{u}(t_{n+1})$  and  $\|e_h^{n+1}\|$  is the  $L^2$  norm of the difference between the TR and AB2 approximations to  $T(t_{n+1})$ . The resulting time-step evolution is shown in Figure 12. Comparing the right-hand plot with the right-hand plot in Figure 3 we can see that there is very close agreement initially. The two left-hand plots look very different however. Rather than exponentially increasing, in the nonisothermal case the time step decreases by an order of magnitude between  $t = 15$  and  $t = 25$ . Moreover, after a second transient, the time step stays almost constant ( $k_n \sim 0.04$  time units) between  $t = 50$  and  $t = 80$ . This behaviour is consistent with a travelling wave or periodic solution and this is further confirmed by the solution plots for  $t \sim 63$ ,  $t \sim 66$  and  $t \sim 70$  shown in Figure 13. An entrained “bubble” of fluid is convected along the bottom wall, creating further bubbles near the top wall as time evolves.

We can also see that there is a problem with our mathematical model for  $t > 90$ . The insulated boundary condition (27) is not compatible with the fluid temperature profile near the outflow and so the bubble is unable to exit gracefully. This manifests itself as a source of vorticity at the outlet and is illustrated in Figure 14. We speculate that a more appropriate condition would be for us to prescribe a “desired” temperature profile at the outflow—this is an issue for a future paper.

We conclude with a brief discussion of the robustness of our preconditioning methodology. Our strategy is to develop a natural extension of the block triangular

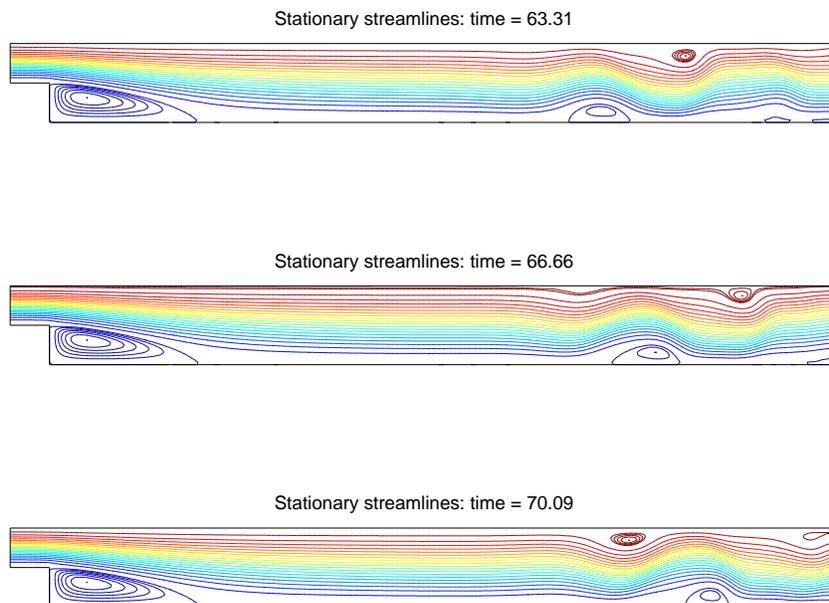


FIGURE 13. Flow over a heated step: snapshots of stationary streamlines at later times.

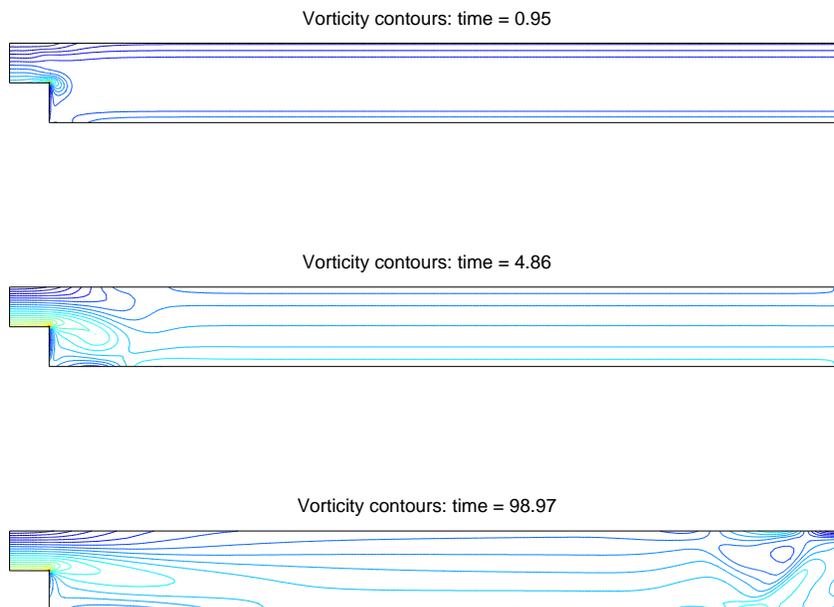


FIGURE 14. Flow over a heated step: snapshots of vorticity.

preconditioner discussed in Section 1. To illustrate the approach, let us express the system (32) (again omitting the matrix subscripts/superscripts) with a right preconditioner  $\mathcal{P}$  so that

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

where  $M := \frac{k_{n+1}}{2} \frac{0}{M}$ . Introducing the velocity-pressure Schur complement matrix  $S = B\mathbf{F}^{-1}B^T$ , an *ideal* block triangular preconditioner

$$\mathcal{P} \equiv \begin{pmatrix} \mathbf{F} & B^T & -M \\ 0 & -S & 0 \\ 0 & 0 & F \end{pmatrix} \approx \begin{pmatrix} \mathbf{F} & B^T & -M \\ B & 0 & 0 \\ 0 & 0 & F \end{pmatrix} \tag{35}$$

is motivated by the identity

$$\begin{pmatrix} \mathbf{F} & B^T & -M \\ B & 0 & 0 \\ 0 & 0 & F \end{pmatrix} \underbrace{\begin{pmatrix} \mathbf{F}^{-1} & \mathbf{F}^{-1}B^T S^{-1} & \mathbf{F}^{-1}MF^{-1} \\ 0 & -S^{-1} & 0 \\ 0 & 0 & F^{-1} \end{pmatrix}}_{\mathcal{P}^{-1}} \equiv \begin{pmatrix} I & 0 & 0 \\ B\mathbf{F}^{-1} & I & B\mathbf{F}^{-1}MF^{-1} \\ 0 & 0 & I \end{pmatrix}.$$

A permutation of the second and third rows and columns of the right-hand side matrix generates a lower triangular matrix with ones on the diagonal. We deduce that GMRES applied to (32) with right preconditioner  $\mathcal{P}$  will converge in two iterations, independent of the parameters  $h, k_{n+1}, \nu$  and  $\epsilon$ .

A practical implementation of the preconditioning strategy embodied in (35) is a three-step process. First, we follow the strategy in Section 2 and approximate the inverses of the matrix operator  $F$  and the Schur complement matrix  $S$  using a single V-cycle of AMG; second, we perform a matrix-vector multiply with  $M$  and with  $B^T$ ; the final step is to use an AMG V-cycle to approximate the inverse of the two scalar systems associated with the matrix operator  $\mathbf{F}$ .

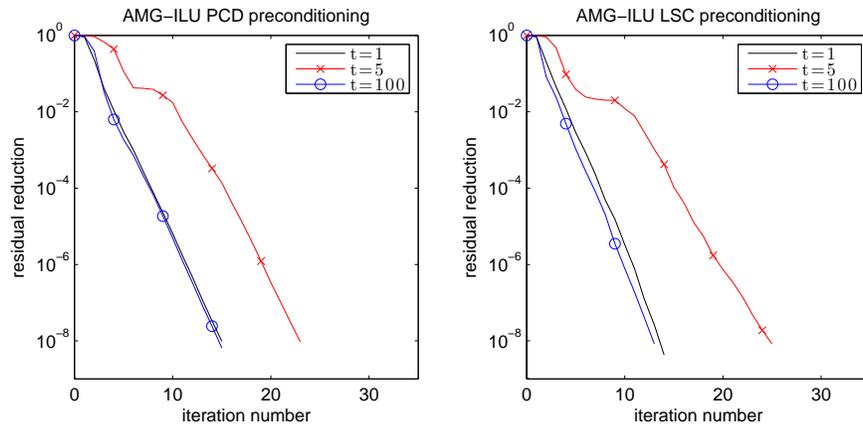


FIGURE 15. GMRES convergence when solving the discretized Boussinesq problem (32) using inexact AMG-ILU preconditioning.

Convergence histories for inexact LSC and PCD preconditioning of the linear systems that arise at approximate times  $t \sim 1, t \sim 5$  and  $t \sim 100$  are plotted in

Figure 15. The associated local time steps ( $k_{n+1}$ ) are 0.0555, 0.148 and 0.0536 time units, respectively. Comparing these results with the isothermal analogues in Figure 6 we see that the solver performance is essentially identical. For a realistic residual reduction factor of  $10^{-6}$ , the inexact version of PCD always converges in 10–20 iterations. The same trend noted earlier is also evident here—the faster the physics, the smaller the time step and the faster the convergence of our linear solver. A comprehensive assessment of the effectiveness of this strategy can be found in [7].

**4. Steady flow over a step with data uncertainty.** In this final section we consider a modelling situation in which the volume of flow in the channel is not known precisely. To model this uncertainty, we consider the stationary Navier–Stokes equations with a random parameter. More specifically, we consider the Navier–Stokes equations with fixed boundary conditions but with an uncertain viscosity. We show that this flow model is equivalent to a problem with a deterministic viscosity together with an uncertain inflow boundary condition: the (quadratic) shape of the inflow profile is exactly as in Section 2, but the *maximum* velocity is not known precisely. Numerical solutions will be generated using a “stochastic” Galerkin finite element method. We also demonstrate that the deterministic PCD and LSC preconditioning schemes in Sections 2 and 3 can be readily extended to provide an effective solution method for the linear systems of algebraic equations that need to be solved at every nonlinear iteration.

**4.1. Alternative stochastic formulations.** First, let us suppose that the fluid viscosity  $\nu$  is spatially uniform but that its value is not known precisely. In such a scenario, we can model  $\nu$  as a random variable of the form

$$\nu := \nu_0 + \nu_1 \xi_1, \quad (36)$$

where  $\nu_0, \nu_1 \in \mathbb{R}$  are given constants and  $\xi_1$  is a random variable on a probability space  $(\Xi, F, \mathbb{P})$ . Here, as usual,  $\Xi$  denotes the set of outcomes,  $F$  is a  $\sigma$ -algebra of events, and  $\mathbb{P} : F \rightarrow [0, 1]$  is a probability measure. If we know a range of possible values for  $\nu$ , it is natural to choose  $\xi_1$  to be a uniform random variable. If  $\xi_1 \sim U(-\sqrt{3}, \sqrt{3})$ , then  $\xi_1$  has mean zero and unit variance, and the expectation and variance of  $\nu$  are

$$\mathbb{E}[\nu] = \nu_0, \quad \text{Var}[\nu] = \nu_1^2, \quad (37)$$

respectively. We can then model different statistical scenarios by varying  $\nu_0$  and  $\nu_1$ . In the numerical experiments reported in the next section, we mirror the setup in Section 2 and set  $\nu_0 = 1/50$ . For a sufficiently small  $\nu_1$  and a fixed spatial grid, we might anticipate that there will be very close agreement between the mean flow solution and the steady-state solution generated previously; see Section 2.

If  $F$  is the  $\sigma$ -algebra generated by the random variable  $\xi_1$  in (36), then the solution of the steady-state Navier–Stokes equations,  $\vec{u}$  and  $p$ , are functions of both the spatial variables  $\vec{x}$  and the random variable  $\xi_1$  (i.e., they are random fields). Thus, given a (deterministic) vector function  $\vec{g} = \vec{g}(\vec{x})$  on  $\Gamma_D$ , the boundary value problem for the Navier–Stokes equations with a random viscosity reads as follows: find  $\vec{u} = \vec{u}(\vec{x}, \xi_1)$  and  $p = p(\vec{x}, \xi_1)$  such that  $\mathbb{P}$ -a.s.,

$$-\nu \nabla^2 \vec{u} + \vec{u} \cdot \nabla \vec{u} + \nabla p = \vec{0} \quad \text{in } \Omega, \quad (38)$$

$$\nabla \cdot \vec{u} = 0 \quad \text{in } \Omega, \quad (39)$$

$$\vec{u} = \vec{g} \quad \text{on } \Gamma_D, \quad (40)$$

$$\nu \nabla \vec{u} \cdot \vec{n} - p \vec{n} = \vec{0} \quad \text{on } \Gamma_N. \quad (41)$$

Now, let  $\vec{U}(\vec{x}, \xi_1) := \vec{u}(\vec{x}, \xi_1)/\nu(\xi_1)$  and  $P(\vec{x}, \xi_1) := p(\vec{x}, \xi_1)/\nu^2(\xi_1)$ . Then,  $\mathbb{P}$ -a.s., the pair  $\vec{U}(\vec{x}, \xi_1)$ ,  $P(\vec{x}, \xi_1)$  satisfies

$$-\nabla^2 \vec{U} + \vec{U} \cdot \nabla \vec{U} + \nabla P = \vec{0} \quad \text{in } \Omega, \quad (42)$$

$$\nabla \cdot \vec{U} = 0 \quad \text{in } \Omega, \quad (43)$$

$$\vec{U} = \vec{G} \quad \text{on } \Gamma_D, \quad (44)$$

$$\nabla \vec{U} \cdot \vec{n} - P \vec{n} = \vec{0} \quad \text{on } \Gamma_N, \quad (45)$$

where  $\vec{G} = \vec{G}(\vec{x}, \xi_1) := \vec{g}(\vec{x})/\nu(\xi_1)$ . The alternative boundary value problem (42)–(45) consists of the steady-state Navier–Stokes equations with unit viscosity, subject to the homogeneous Neumann boundary condition (45) and the stochastic Dirichlet boundary condition (44). For the inflow–outflow configuration shown in Figure 1, the quadratic inlet profile given by  $\vec{g}(\vec{x})$  is scaled by the random variable  $1/\nu(\xi_1)$ .

We focus on the random viscosity formulation in the sequel. There are two reasons for doing so: first, the problem (38)–(41) is more compatible with our existing (IFISS) software resource; and second, modelling the uncertainty via the viscosity parameter allows us to make a direct comparison with results in the literature, see, e.g. Le Maître & Knio [19, Chapter 6].

The problem nonlinearity is not the primary focus of this paper. To test out the linear solver strategy we take a simple Picard linearization of the convection term herein. This means that, at the  $n$ th step of the iterative procedure we have to solve a linearized problem with  $\vec{u} \cdot \nabla \vec{u}$  in (38) replaced by the lagged convection field  $\vec{u}^n \cdot \nabla \vec{u}^{n+1}$ . Note that we could equally well have solved the time-dependent version of (38)–(41) using linearized backward Euler time stepping with a large fixed time step. Discussion of this important aspect of the solution process is deferred to our follow-up publication [2].

Since the solution and test functions now depend on  $\xi_1$ , deriving a variational formulation from the linearized boundary value problem requires us to take expectations. For the particular choice  $\xi_1 \sim U(-\sqrt{3}, \sqrt{3})$  we have a constant probability density function,  $\rho(\lambda) = 1/(2\sqrt{3})$ . Moreover  $\Lambda := \xi_1(\Xi)$ , the range of  $\xi_1$ , is a finite interval  $\Lambda = [-\sqrt{3}, \sqrt{3}]$ . Now recall that for any measurable function  $f$  of  $\xi_1$ , the expectation is given by  $\mathbb{E}[f(\xi_1)] = \int_{\Lambda} \rho(\lambda) f(\lambda) d\lambda$ . Hence, at the  $n$ th Picard iteration, we have to solve the following variational problem: find  $\vec{u}^{n+1}(\vec{x}, \lambda) \in \mathbf{V}_E(\Lambda, \Omega)$  and  $p^{n+1}(\vec{x}, \lambda) \in W(\Lambda, \Omega)$  satisfying

$$\int_{\Lambda} \rho(\lambda) \left\{ \nu(\lambda) (\nabla \vec{u}^{n+1}, \nabla \vec{v}) + (\vec{u}^n \cdot \nabla \vec{u}^{n+1}, \vec{v}) - (p^{n+1}, \nabla \cdot \vec{v}) \right\} d\lambda = 0, \quad (46)$$

$$\int_{\Lambda} \rho(\lambda) (q, \nabla \cdot \vec{u}^{n+1}) d\lambda = 0, \quad (47)$$

for all  $\vec{v}(\vec{x}, \lambda) \in \mathbf{V}_0(\Lambda, \Omega)$  and  $q(\vec{x}, \lambda) \in W(\Lambda, \Omega)$ . Note that  $(\cdot, \cdot)$  in (46)–(47) represents the  $L^2(\Omega)$  inner product. The function spaces  $\mathbf{V}_0(\Lambda, \Omega) := L^2_{\rho}(\Lambda, \mathbf{H}_0^1(\Omega))$  and  $W(\Lambda, \Omega) := L^2_{\rho}(\Lambda, L^2(\Omega))$  contain functions of  $\lambda$  with finite second moment. These spaces will be formally defined in [2].

**4.2. A fully discrete formulation.** Full details of the Galerkin formulation with spectral approximation in the stochastic variable can also be found in [2]. A general overview of the approximation issues is given in [3]. The goal is to combine the usual spatial  $Q_2$ – $Q_1$  velocity–pressure approximation with standard polynomial approximation on  $\Lambda$ . To that end, let  $S^k = S^k(\Lambda) \subset L^2_{\rho}(\Lambda)$  be the set of univariate

polynomials in  $\lambda$  of degree  $\leq k$  on the interval  $\Lambda$ . Then the Picard linearization of the convection term leaves us with the task of computing  $\vec{u}_{hk}^{n+1} \in \mathbf{X}_E^h \otimes S^k$  and  $p_{hk}^{n+1} \in M^h \otimes S^k$  satisfying the fully discrete formulation,

$$\mathbb{E}\left[\nu(\nabla \vec{u}_{hk}^{n+1}, \nabla \vec{v}_{hk})\right] + \mathbb{E}\left[(\vec{u}_{hk}^n \cdot \nabla \vec{u}_{hk}^{n+1}, \vec{v}_{hk})\right] - \mathbb{E}\left[(p_{hk}^{n+1}, \nabla \cdot \vec{v}_{hk})\right] = 0, \quad (48)$$

$$\mathbb{E}\left[(\nabla \cdot \vec{u}_{hk}^{n+1}, q_{hk})\right] = 0, \quad (49)$$

for all  $\vec{v}_{hk} \in \mathbf{X}_0^h \otimes S^k$  and  $q_{hk} \in M^h \otimes S^k$ , where  $\mathbb{E}[\cdot]$  represents the expectation operator with respect to the probability density function  $\rho(\lambda)$ . Alternatively, given  $\vec{u}_{hk}^n \in \mathbf{X}_E^h \otimes S^k$  and  $p_{hk}^n \in M^h \otimes S^k$  we can compute the updates  $\delta \vec{u}_{hk}^n := \vec{u}_{hk}^{n+1} - \vec{u}_{hk}^n \in \mathbf{X}_0^h \otimes S^k$  and  $\delta p_{hk}^n := p_{hk}^{n+1} - p_{hk}^n \in M^h \otimes S^k$  by solving the system

$$\mathbb{E}\left[\nu(\nabla \delta \vec{u}_{hk}^n, \nabla \vec{v}_{hk})\right] + \mathbb{E}\left[(\vec{u}_{hk}^n \cdot \nabla \delta \vec{u}_{hk}^n, \vec{v}_{hk})\right] - \mathbb{E}\left[(\delta p_{hk}^n, \nabla \cdot \vec{v}_{hk})\right] = R_n(\vec{v}_{hk}), \quad (50)$$

$$\mathbb{E}\left[(\nabla \cdot \delta \vec{u}_{hk}^n, q_{hk})\right] = r_n(q_{hk}), \quad (51)$$

for all  $\vec{v}_{hk} \in \mathbf{X}_0^h \otimes S^k$  and  $q_{hk} \in M^h \otimes S^k$ , where the residuals are given by

$$R_n(\vec{v}_{hk}) := -\mathbb{E}\left[\nu(\nabla \vec{u}_{hk}^n, \nabla \vec{v}_{hk})\right] - \mathbb{E}\left[(\vec{u}_{hk}^n \cdot \nabla \vec{u}_{hk}^n, \vec{v}_{hk})\right] + \mathbb{E}\left[(p_{hk}^n, \nabla \cdot \vec{v}_{hk})\right],$$

$$r_n(q_{hk}) := -\mathbb{E}\left[(\nabla \cdot \vec{u}_{hk}^n, q_{hk})\right].$$

Given a basis set  $S^k = \{\varphi_\ell\}_{\ell=0}^k$ , we get to a linear algebra problem by expanding the updates in terms of the tensor product basis functions,

$$\delta \vec{u}_{hk}^n(\vec{x}, \lambda) = \begin{bmatrix} \sum_{\ell=0}^k \sum_{i=1}^{n_u} \alpha_{i\ell}^{x,n} \phi_i(\vec{x}) \varphi_\ell(\lambda) \\ \sum_{\ell=0}^k \sum_{i=1}^{n_u} \alpha_{i\ell}^{y,n} \phi_i(\vec{x}) \varphi_\ell(\lambda) \end{bmatrix}, \quad \delta p_{hk}^n = \sum_{\ell=0}^k \sum_{j=1}^{n_p} \beta_{j\ell}^n \psi_j \varphi_\ell.$$

The coefficient vectors  $\boldsymbol{\alpha}^n = [\boldsymbol{\alpha}^{x,n}, \boldsymbol{\alpha}^{y,n}]$  and  $\boldsymbol{\beta}^n$  may then be computed by solving the saddle-point system associated with (50)–(51); that is,

$$\begin{pmatrix} \mathbb{F}_\nu^n & \mathbb{B}^T \\ \mathbb{B} & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{\alpha}^n \\ \boldsymbol{\beta}^n \end{pmatrix} = \begin{pmatrix} \mathbf{f}^n \\ \mathbf{g}^n \end{pmatrix}, \quad (52)$$

where  $\mathbf{f}^n$  and  $\mathbf{g}^n$  are associated with the residual vectors  $R_n$  and  $r_n$  defined above.

The matrices  $\mathbb{F}_\nu^n$  and  $\mathbb{B}$  in (52) are defined in terms of Kronecker products of smaller matrices. To see this, let us order the degrees of freedom to run over the stochastic basis functions  $\varphi_\ell$  in turn: this means that the coefficient vectors  $\boldsymbol{\alpha}^{x,n}$  and  $\boldsymbol{\alpha}^{y,n}$  each have  $k+1$  consecutive blocks each of length  $n_u$ , and the vector  $\boldsymbol{\beta}^n$  has  $k+1$  consecutive blocks each of length  $n_p$ . With this numbering, the matrix  $\mathbb{B}$  in (52) can be simply written as  $\mathbb{B} = [I \otimes B_x, I \otimes B_y]$ , where  $B_x$  and  $B_y$  are the components of the discrete divergence operator (14)–(15) defined in Section 2.

The Kronecker product structure of the matrix  $\mathbb{F}_\nu^n$  looks quite complicated at first sight:

$$\mathbb{F}_\nu^n := (\nu_0 G_0 + \nu_1 G_1) \otimes \mathbf{A} + \sum_{\ell=0}^k H_\ell \otimes \mathbf{N}_\ell. \quad (53)$$

There is a lot of structure in (53) to exploit however. The Galerkin ‘‘G-matrices’’ take the form,

$$G_0 := [G_0]_{\ell s} = \mathbb{E}[\varphi_s \varphi_\ell], \quad \ell, s = 0, \dots, k, \quad (54)$$

$$G_1 := [G_1]_{\ell s} = \mathbb{E}[\lambda \varphi_s \varphi_\ell], \quad \ell, s = 0, \dots, k. \quad (55)$$

If we construct a basis set  $\{\varphi_\ell\}_{\ell=0}^k$  that is orthonormal with respect to the inner product associated with the expectation operator (scaled Legendre polynomials in

the case of the uniform distribution), then  $G_0$  is the identity matrix, and  $G_1$  is a permutation of a tridiagonal matrix, with at most two nonzero entries per row (see Powell & Elman [20]). This means that the diffusion part of the matrix  $\mathbb{F}_\nu^n$  is block sparse (inherited from  $G_1$ ), with at most three nonzero blocks in each block row. The convection part of the matrix  $\mathbb{F}_\nu^n$  involves a sum of Kronecker products of the Galerkin “H-matrices”,

$$H_\ell := [H_\ell]_{ms} = \mathbb{E}[\varphi_\ell \varphi_s \varphi_m], \quad m, s = 0, \dots, k, \quad (56)$$

with convection matrices  $N_\ell$  that have block diagonal components,

$$N_\ell(\vec{u}_{hk}^n) := [N_\ell]_{ij} = (\vec{u}_{h\ell}^n \cdot \nabla \phi_i, \phi_j), \quad i, j = 1, \dots, n_u. \quad (57)$$

Note that the “wind” components  $\vec{u}_{h\ell}^n$  correspond to the spatial coefficients in the stochastic expansion of the velocity field at the  $n$ th nonlinear iteration; that is,

$$\vec{u}_{hk}^n = \sum_{\ell=0}^k \underbrace{\sum_{i=1}^{n_u} \vec{u}_{i\ell}^n \phi_i(\vec{x})}_{\vec{u}_{h\ell}^n} \varphi_\ell(\lambda). \quad (58)$$

**4.3. Computational results.** As mentioned above, we mirror the setup in Section 2 and set  $\nu_0 = 1/50$  and  $L = 5$ . We show results for the specific case of  $\nu_1 = 1/500$  which corresponds to  $\nu$  being a uniform random variable on the interval  $[0.01654, 0.02346]$ . The spatial discretization is the reference grid shown in Figure 2 and the stochastic approximation is of polynomial degree  $k = 4$ . The resulting numerical approximation to (46)–(47) is generated by taking 15 Picard iteration steps (starting from the stochastic Stokes-flow solution). The statistical properties of the computed solution are illustrated in Figures 16 and 17.

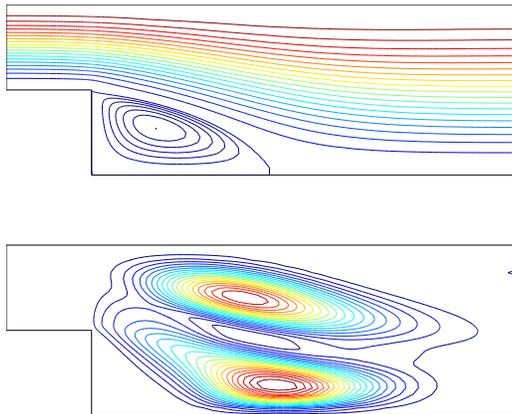


FIGURE 16. Uncertain flow over a step: streamlines of the mean flow field (top) and contours of the variance of the magnitude of the mean flow field (bottom).

As anticipated, the mean flow solution in Figure 16 looks exactly like the steady flow solution illustrated in the bottom plot of Figure 4. The mean pressure solution in Figure 17 is also indistinguishable from the deterministic steady-state analogue (not shown). The spatial singularity in the pressure solution is a prominent feature. The second order solution statistics shown in Figures 16 and 17 are also interesting. The uncertainty in the volume of flow at the inflow generates uncertainty in the

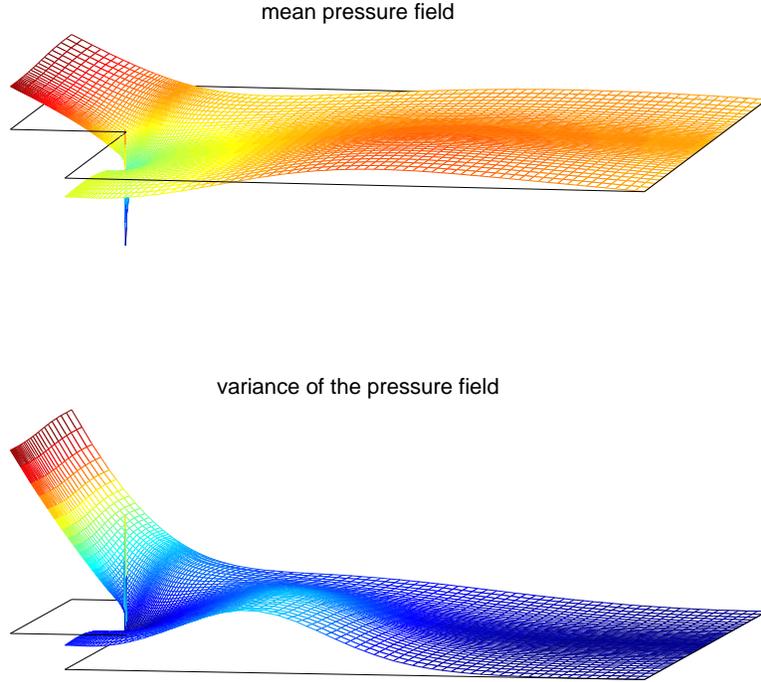


FIGURE 17. Uncertain flow over a step: mean (top) and variance (bottom) of the pressure field.

length of the recirculating eddy, so the variance of the velocity solution is concentrated in two areas downstream of the step. In contrast, the pressure uncertainty is concentrated at the inflow—the uncertainty in the volume of fluid at the inflow generates uncertainty in the pressure drop between the inflow and outflow (recall that the outflow boundary condition forces the pressure to have mean zero).

The influence of the degree of spectral approximation,  $k$ , on the computed solution can be inferred from the behaviour of the spatial coefficients  $\bar{u}_{h\ell}^*$  in the stochastic expansion (58) of the “converged” velocity field  $\bar{u}_{hk}^* := \bar{u}_{hk}^{15}$ . To illustrate this, the horizontal velocity component of the spatial coefficients (see (58)) are plotted in Figure 18, and the norms of the vector of coefficients  $\{u_{i\ell}^x\}_{i=1}^{n_u}$  are also recorded. The five polynomial components are ordered vertically with the mean horizontal velocity component  $u_{h0}^x$  at the top, the linear component  $u_{h1}^x$  just below, and the quartic component  $u_{h4}^x$  at the bottom. All five components can be seen to be spatially smooth. Also evident is the fact that the flow resolution monotonically increases with increasing polynomial degree (the linear horizontal velocity component has three “peaks” in Figure 18, whereas the quartic horizontal velocity component has six “peaks”). Another important point is that the norm of these components decreases rapidly with  $k$ —this suggests that an increase in the polynomial degree would have little effect on the numerical solution that is generated. We note that the issue of how to balance the spatial and spectral errors inside stochastic Galerkin approximation is a very active research topic; see Xiu [24, Chapter 6] and Le Maître & Knio [19, Chapter 9] for further discussion of this point.

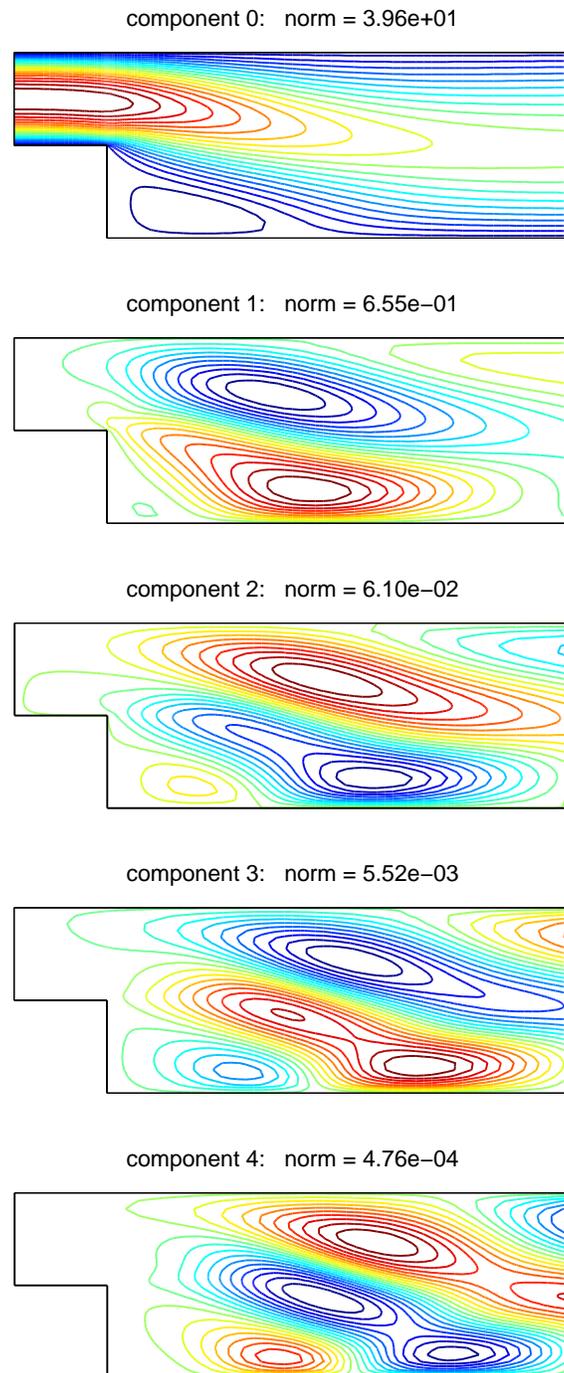


FIGURE 18. Uncertain flow over a step: contours of the spatial coefficients in the stochastic expansion of the horizontal velocity solution.

We conclude with a brief discussion of possible preconditioning strategies for the linear system (52). Note the increased dimensionality: the number of unknown coefficients is  $(k+1) \times n$  where  $k$  is the degree of the spectral approximation and  $n = 14267$  is the dimension of the spatially discretized system. Our immediate aim is to extend the block triangular preconditioner discussed in Section 1. We present results computed using three different strategies, all of which are based on approximating the spatial convection–diffusion operator in (53) by the mean component  $F_0 = \nu_0 I + N_0$ , where the convection matrix operator is simply linearized using the mean velocity field  $\bar{u}_{h0}^*$  defined in (57). Recalling (53), and noting that  $H_0 = I$ , we can see why this might be a good idea:

$$\begin{aligned}
\mathbb{F}_\nu &= (\nu_0 I + \nu_1 G_1) \otimes \mathbf{A} + \sum_{\ell=0}^k H_\ell \otimes \mathbf{N}_\ell \\
&= I \otimes (\nu_0 \mathbf{A} + \mathbf{N}_0) + \nu_1 G_1 \otimes \mathbf{A} + \sum_{\ell=1}^k H_\ell \otimes \mathbf{N}_\ell \\
&= I \otimes \mathbf{F}_0 + \nu_1 G_1 \otimes \mathbf{A} + \sum_{\ell=1}^k H_\ell \otimes \mathbf{N}_\ell \\
&\approx I \otimes \mathbf{F}_0 =: \mathbb{F}_0.
\end{aligned} \tag{59}$$

This approximation will be good and the preconditioning strategy will be effective whenever the ratio  $\nu_1/\nu_0$  is close to zero; see Powell & Elman [20]. (See Ernst et al. [12] for a detailed discussion of mean-based preconditioning in a mixed approximation context.) If the exact pressure Schur complement,  $\mathbb{S} = \mathbb{B}\mathbb{F}_\nu^{-1}\mathbb{B}^T$ , is also approximated using the construction (59) then we deduce that

$$\begin{aligned}
\mathbb{S} &= [I \otimes B_x, I \otimes B_y] \mathbb{F}_\nu^{-1} [I \otimes B_x, I \otimes B_y]^T \\
&\approx [I \otimes B_x, I \otimes B_y] (I \otimes \mathbf{F}_0)^{-1} [I \otimes B_x, I \otimes B_y]^T \\
&= [I \otimes B_x, I \otimes B_y] (I \otimes \mathbf{F}_0^{-1}) [I \otimes B_x, I \otimes B_y]^T \\
&= (I \otimes B_x)(I \otimes \mathbf{F}_0^{-1})(I \otimes B_x^T) + (I \otimes B_y)(I \otimes \mathbf{F}_0^{-1})(I \otimes B_y^T) \\
&= I \otimes (B_x \mathbf{F}_0^{-1} B_x^T) + I \otimes (B_y \mathbf{F}_0^{-1} B_y^T) \\
&= I \otimes (B \mathbf{F}_0^{-1} B^T) =: I \otimes \mathbb{S}_0 =: \mathbb{S}_0.
\end{aligned} \tag{60}$$

Combining (59) and (60), we can define an “ideal” preconditioner for (52) (analogous to (20)) via

$$\mathcal{P} := \begin{pmatrix} \mathbb{F}_0 & \mathbb{B}^T \\ 0 & -\mathbb{S}_0 \end{pmatrix} \approx \begin{pmatrix} \mathbb{F}_\nu & \mathbb{B}^T \\ \mathbb{B} & 0 \end{pmatrix}. \tag{61}$$

The ideal preconditioner is not practical (it requires the explicit construction of the deterministic problem Schur complement  $B\mathbf{F}_0^{-1}B^T$ ) but it does provide us with a reference point for evaluating performance. A practical approach is obtained by approximating the mean-based Schur complement using the PCD and LSC constructions that are described in Section 1.

GMRES convergence curves for the ideal preconditioner (61) and the *exact* LSC and PCD analogues are shown in Figure 19. The convergence curves in the right-hand plot are those obtained using the reference grid in Figure 2. These results should be compared with the large time step ( $t \sim 100$ ) results for the mean-value deterministic problem that are shown in Figure 6. The left-hand plot in Figure 19 is included to enable an assessment of the robustness of the preconditioning strategies.<sup>4</sup> The results in Figure 19, together with experiments not reported here, give

<sup>4</sup>The coarse grid is generated in IFISS 3.2 by running `newstep_domain.m` with the grid parameter set to 5 and the stretch parameter set to 1.3. The stochastic flow problem was rerun on this grid with all other problem parameters left unchanged.

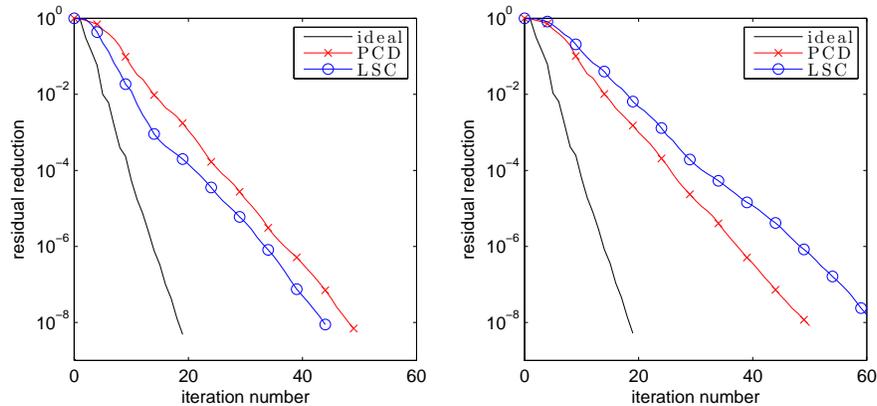


FIGURE 19. GMRES convergence when solving the discretized stochastic flow problem (52) using exact preconditioning for a coarse grid (left) and for the reference grid in Section 2 (right).

us confidence that the ideal preconditioner is perfectly robust; both with respect to the spatial approximation *and* with respect to the degree of spectral approximation. The LSC preconditioning results are less encouraging: looking at Figure 19 a deterioration in the convergence rate of LSC is apparent when the spatial grid is refined. There is no real surprise here—the grid dependence of LSC is also evident in the deterministic case; see Figure 8. In contrast, the PCD preconditioner results are very promising—the convergence rate stays within a factor of two of the *ideal* preconditioner and the approximation appears to be perfectly robust with respect to the parameters  $h$  and  $k$ . We aim to provide theoretical justification for this assertion in the future.

**5. Summary.** A general strategy for finding numerical solutions of incompressible flow models using implicit methods is outlined in this paper. The main ingredients are a self-adaptive time-stepping approach in conjunction with a linearization strategy that requires only one linear system solution at each time step. Moreover, it is shown that the linearized systems can be solved in essentially optimal order complexity using specialized algebraic multigrid solver components. The novel contribution is that the effectiveness of the solver methodology is demonstrated on a sequence of increasingly complex physical models. The computational results show that very little preliminary knowledge of problem structure or parameter tuning is needed to efficiently compute accurate solutions.

**Acknowledgements.** Some of the ideas in this paper were described in a series of four lectures given at the 12th School on Mathematical Theory in Fluid Mechanics held in Kacov, Czech Republic, May 27–June 3, 2011. We are grateful to the Organizing Committee of the School for the invitation and especially to Professors Josef Málek and Mirko Rokyta for their wonderful hospitality. We would also like to thank Dr Andrew Hazel for his advice and his careful reading of Section 3. Our research on uncertainty quantification is supported by EPSRC grant EP/H021205/1.

## REFERENCES

- [1] Uri M. Ascher, Steven J. Ruuth and Brian T.R. Wetton, *Implicit–explicit methods for time-dependent partial differential equations*, SIAM J. Numer. Anal., **32** (1995), 797–823.
- [2] Alexei Bespalov, Catherine E. Powell and David Silvester, *Fast iterative solvers for steady-state Navier–Stokes equations with random data*, preprint in preparation, see <http://eprints.ma.man.ac.uk/>.
- [3] Alexei Bespalov, Catherine E. Powell and David Silvester, *A priori error analysis of stochastic Galerkin mixed approximations of elliptic PDEs with random data.*, MIMS Eprint 2011.91, <http://eprints.ma.man.ac.uk/1696/>.
- [4] Jonathan Boyle, Milan Mihajlović and Jennifer Scott, *HSL\_MI20: an efficient AMG preconditioner for finite element problems in 3D*, Internat. J. Numer. Methods Engrg, **82** (2010), 64–98.
- [5] H. Damanik, J. Hron, A. Ouazzi and S. Turek, *A monolithic FEM–multigrid solver for non-isothermal incompressible flow on general meshes*, J. Comput. Phys., **228** (2009), 3869–3881.
- [6] Philip G. Drazin, “Introduction to Hydrodynamic Stability,” Cambridge University Press, Cambridge, 2002. xviii+258 pp. ISBN: 0-521-80427-2; 0-521-00965-0.
- [7] Howard Elman, Milan Mihajlović and David Silvester, *Fast iterative solvers for buoyancy driven flow problems*, J. Comput. Phys., **230** (2011), 3900–3914.
- [8] Howard C. Elman, Alison Ramage and David J. Silvester, *Algorithm 866: IFISS, a MATLAB toolbox for modelling incompressible flow*, ACM Trans. Math. Softw., **33**, (2007), 2–14.
- [9] Howard C. Elman, David J. Silvester and Andrew J. Wathen, *Performance and analysis of saddle point preconditioners for the discrete steady-state Navier–Stokes equations*, Numer. Math., **90** (2002), 665–688.
- [10] Howard C. Elman, David J. Silvester and Andrew J. Wathen, “Finite Elements and Fast Iterative Solvers: With Applications in Incompressible Fluid Dynamics,” Oxford University Press, New York, 2005. xiv+400 pp. ISBN: 978-0-19-852868-5; 0-19-852868-X.
- [11] Howard C. Elman and Ray S. Tuminaro, *Boundary conditions in approximate commutator preconditioners for the Navier–Stokes equations*, Electron. Trans. Numer. Anal., **35** (2009), 257–280.
- [12] Oliver G. Ernst, Catherine E. Powell, David J. Silvester and Elisabeth Ullmann, *Efficient solvers for a linear stochastic Galerkin mixed formulation of diffusion problems with random data*, SIAM J. Sci. Comput., **31** (2009), 1424–1447.
- [13] P.M. Gresho, D.K. Gartling, J.R. Torczynski, K.A. Cliffe, K.H. Winters, T.J. Garratt, A. Spence and J.W. Goodrich, *Is the steady viscous incompressible two-dimensional flow over a backward-facing step at  $Re=800$  stable?*, Internat. J. Numer. Methods Fluids, **17** (1993), 501–541.
- [14] Philip M. Gresho, David F. Griffiths and David J. Silvester, *Adaptive time-stepping for incompressible flow; Part I: Scalar advection–diffusion*, SIAM J. Sci. Comput., **30** (2008), 2018–2054.
- [15] P.M. Gresho and R.L. Sani, “Incompressible Flow and the Finite Element Method: Volume 2: Isothermal Laminar Flow,” John Wiley, Chichester, 1998.
- [16] David A. Kay, Philip M. Gresho, David F. Griffiths and David J. Silvester, *Adaptive time-stepping for incompressible flow; Part II: Navier–Stokes equations*, SIAM J. Sci. Comput., **32** (2010), 111–128.
- [17] David Kay, Daniel Loghin and Andrew Wathen, *A preconditioner for the steady-state Navier–Stokes equations*, SIAM J. Sci. Comput., **24** (2002), 237–256.
- [18] William Layton, “Introduction to the Numerical Analysis of Incompressible Viscous Flow,” SIAM, Philadelphia, 2008. xx+213 pp. ISBN: 978-0-898716-57-3.
- [19] O. P. Le Maître and O. M. Knio, “Spectral Methods for Uncertainty Quantification: With Applications to Computational Fluid Dynamics,” Springer, New York, 2010. xvi+536 pp. ISBN: 978-90-481-3519-6.
- [20] Catherine E. Powell and Howard C. Elman, *Block-diagonal preconditioning for spectral stochastic finite-element systems*, IMA J. Numer. Anal., **29** (2009), 350–375.
- [21] David Silvester, Howard Elman, David Kay and Andrew Wathen, *Efficient preconditioning of the linearised Navier–Stokes equations for incompressible flow*, J. Comput. Appl. Math., **128** (2001), 261–279.
- [22] David Silvester, Howard Elman and Alison Ramage, “Incompressible Flow and Iterative Solver Software (IFISS),” Version 3.2, 2012. Available from <http://www.manchester.ac.uk/ifiss/>.

- [23] J. C. Simo and F. Armero, *Unconditional stability and long-term behaviour of transient algorithms for the incompressible Navier–Stokes and Euler equations*, *Comput. Methods Appl. Mech. Engrg.*, **111** (1994), 111–154.
- [24] Dongbin Xiu, “Numerical Methods for Stochastic Computations: A spectral method approach,” Princeton University Press, Princeton, NJ, 2010. xiv+127 pp. ISBN: 978-0-691-14212-8.

Received xxxx 20xx; revised xxxx 20xx.

*E-mail address:* `d.silvester@manchester.ac.uk`

*E-mail address:* `alexey.bespalov@manchester.ac.uk`

*E-mail address:* `catherine.powell@manchester.ac.uk`