1. Introduction. Let Ω be an open bounded domain in $\mathrm{I\!R}^{\mathrm{d}}$ with boundary Γ. The steady-state Navier-Stokes equations are given by the system of partial differential equations

$$-\varepsilon\Delta\mathbf{u} + (\mathbf{u}\cdot\nabla)\mathbf{u} + \nabla p = \mathbf{f} \quad \text{in } \Omega, \tag{1.1a}$$

$$-\nabla\cdot\mathbf{u} = 0 \quad \text{in } \Omega, \tag{1.1b}$$

$$\mathcal{B}\mathbf{u} = \mathbf{g} \quad \text{on } \Gamma, \tag{1.1c}$$

where $(\mathbf{u})_i = u_i,\ i = 1, \ldots, d$, are the components of velocity, p is the pressure and ϵ is the viscosity parameter. The boundary conditions are represented by $\mathcal{B}$ which may be a differential operator to allow for derivative boundary conditions. In the following we concentrate on the case $d = 2$, since the 3 dimensional case can be treated similarly, as we show in section 2.

It is well-known (e.g. [4, pp. 255–257]) that an efficient numerical solution to (1.1) can be obtained via a fixed point (or Picard) iteration. More precisely, for $n = 1, 2, \ldots$, one constructs iterates $\{\mathbf{u}_n, p_n\}$ satisfying the linearised version of (1.1), known as the Oseen equations:

$$-\varepsilon\Delta\mathbf{u}_n + (\mathbf{u}_{n-1}\cdot\nabla)\mathbf{u}_n + \nabla p_n = \mathbf{f} \quad \text{in } \Omega, \tag{1.2a}$$

$$-\nabla\cdot\mathbf{u}_n = 0 \quad \text{in } \Omega, \tag{1.2b}$$

$$\mathcal{B}\mathbf{u}_n = \mathbf{g} \quad \text{on } \Gamma. \tag{1.2c}$$

The initial guess $\mathbf{u}_0$ can be taken to be the solution to the associated Stokes problem with boundary conditions (1.2c) (i.e., $\mathbf{u}_{-1} = \mathbf{0}$).

In this paper we are interested in the efficient solution to the sparse linear system

$$\mathcal{A}\begin{pmatrix}\mathbf{u}\\ p\end{pmatrix} = \begin{pmatrix}\mathbf{f}\\ 0\end{pmatrix} \tag{1.3}$$

arising from the discretization of the Oseen equations (1.2). For large problems, the only viable solution methods are Krylov subspace methods (see [20] for an overview). However, the performance of these methods is sensitive to the conditioning of $\mathcal{A}$ and thus the idea of preconditioning must be exploited. More precisely, we seek a preconditioning matrix $\mathcal{P} = \mathcal{P}_1\mathcal{P}_2$ which is easy to construct and invert, such that the equivalent system

$$\mathcal{P}_1^{-1}\mathcal{A}\mathcal{P}_2^{-1}\begin{pmatrix}\hat{\mathbf{u}}\\ \hat{p}\end{pmatrix} = \mathcal{P}_1^{-1}\begin{pmatrix}\mathbf{f}\\ 0\end{pmatrix}, \quad \begin{pmatrix}\hat{\mathbf{u}}\\ \hat{p}\end{pmatrix} = \mathcal{P}_2\begin{pmatrix}\mathbf{u}\\ p\end{pmatrix} \tag{1.4}$$

is solved efficiently via an appropriate iterative method. A sufficient condition for this to be the case is that $\mathcal{P}$ approximate in some sense the coefficient matrix: $\mathcal{P} \sim \mathcal{A}$. In the following we consider only the case $\mathcal{P} = \mathcal{P}_2$.

In general, mixed formulations for the discretization of the 2-dimensional Oseen problem give rise to a non-symmetric coefficient matrix of the form

$$\mathcal{A} = \begin{pmatrix} F & 0 & B_x^T \\ 0 & F & B_y^T \\ B_x & B_y & -\beta C \end{pmatrix} = \begin{pmatrix} \mathbf{F} & B^T \\ B & -\beta C \end{pmatrix}, \tag{1.5}$$

with

$$\mathbf{F} = \begin{pmatrix} F & 0 \\ 0 & F \end{pmatrix}, \quad B = \begin{pmatrix} B_x & B_y \end{pmatrix},$$

where $F = \varepsilon A + S$ is a discrete advection-diffusion operator, with A the discrete Laplacian and S the discrete advection operator; B is the negative discrete divergence operator. Since we are interested in both stable and stabilised formulations of the discrete problem, we include in (1.5) the stabilisation matrix C which is symmetric and positive semidefinite. Note that for stable methods $\beta = 0$; otherwise $\beta > 0$ ([9],[10]).

An ideal, though academic, preconditioner has already been suggested by Murphy and Wathen in [16],[17]. This is of the form

$$\mathcal{P} = \begin{pmatrix} \mathbf{F} & 0 \\ 0 & -X \end{pmatrix}, \tag{1.6}$$

where

$$X = B\mathbf{F}^{-1}B^T + \beta C. \tag{1.7}$$

It is shown in [16] and [17], that for this choice of X the preconditioned matrix $\mathcal{P}^{-1}\mathcal{A}$ has only three distinct eigenvalues. This guarantees convergence of an appropriate iterative solver in at most three iterations. The use of (1.6) requires the solution of separate linear systems with coefficient matrices $\mathbf{F}$ and X at each iteration. Inversion of $\mathbf{F}$ may well be practical (see section 4); however, the so-called Schur complement (1.7) is an impracticable choice and alternatives have been sought in the literature.

Elman and Silvester [6] suggested a triangular preconditioner

$$\mathcal{P} = \begin{pmatrix} \mathbf{F} & B^T \\ 0 & -X \end{pmatrix}, \tag{1.8}$$

which, in the case of X being the exact Schur complement (1.7), leads to a preconditioned matrix $\mathcal{A}\mathcal{P}^{-1}$ with a minimal polynomial of degree two, thus guaranteeing convergence in two iterations. As a practical alternative, they approximate the Schur complement X with a viscosity scaled pressure Grammian (or pressure mass matrix)

$$X_M = M_p/\varepsilon. \tag{1.9}$$

This choice leads to the eigenvalues of XX_M^{-1} being bounded independently of the mesh size. X_M is seen to perform well mostly for large values of viscosity, which is essentially the Stokes flow regime for which Wathen and Silvester ([24]) showed the optimality of X_M. However, the convergence rates for the choice (1.9) deteriorate roughly like $1/\varepsilon$ for the Oseen problem. Moreover, the bound on the eigenvalues guarantees asymptotic convergence of iterative solvers independently of the mesh refinement; however, the number of iterations is too large for any practical purpose.

In an attempt to reduce the dependence on viscosity, the choice

$$X_B = (BB^T)(B\mathbf{F}B^T)^{-1}(BB^T)$$

was suggested by Elman in ([5]). When applied to the finite difference MAC scheme [12] for an Oseen problem with periodic boundary conditions and constant wind, Elman showed [5, Theorem 3.3]

$$(B\mathbf{F}^{-1}B^T)X_B^{-1} = I. \tag{1.10}$$

Numerical experiments using the MAC scheme presented by Elman suggest that the dependence on viscosity is $O(\epsilon^{-1/2})$ with only mild dependence on h. However, in our experience (see section 4) X_B is not a robust choice when considering finite element discretizations, particularly when a stabilisation term is required. Moreover, for certain elements the dependence on mesh size seems to be at least $O(h^{-1/2})$ and the practical implementation of such a preconditioner to many mixed finite elements is not clear.

In this paper we introduce a preconditioner of the form (1.8) with a new choice of X which seems to have no dependence on the mesh size and only a slight dependence on viscosity. In section 2 we calculate Green's tensor for the Oseen operator. In section 3 we introduce our preconditioner and justify our choice via an argument involving the continuous inverse calculated in section 2. Finally, section 4 discusses the implementation and presents numerical results which show that the preconditioner is highly competitive and may be applied to a wide variety of discretizations.

2. Green's tensor and preconditioning. Our approach to devising a preconditioner is based on the form of $\mathcal{A}^{-1}$. If

$$\mathcal{A}^{-1} = \begin{pmatrix} M & N \\ P & Q \end{pmatrix},$$

it is easy to show that $Q = -X^{-1}$, $\forall \beta \geq 0$. In other words, the inverse of the Schur complement (1.7) is also the (3,3)-block of the inverse of (1.5). Thus, it is natural to turn our attention to the continuous form of the inverse, or the Green's tensor for the Oseen operator.

2.1. Green's tensor results. In this section we collect some definitions and properties of Green's functions which we will find useful in the derivation of our preconditioner.

Consider a system of m differential equations in m unknowns with homogeneous Dirichlet boundary conditions

$$\mathcal{L}\mathbf{u} = \mathbf{f} \qquad \text{in } \ \Omega \subseteq \mathrm{I\!R}^{\mathrm{d}} \tag{2.1a}$$

$$\mathbf{u} = 0 \qquad \text{on } \ \partial\Omega \tag{2.1b}$$

By analogy with the scalar definition, the Green's tensor for a tensor differential operator $\mathcal{L}$ is defined to be the solution to

$$\mathcal{L}_{\mathbf{x}}\mathbf{G}(\mathbf{x},\mathbf{y}) = \delta\ \delta(\mathbf{x},\mathbf{y}) \qquad \mathbf{x},\mathbf{y} \in \Omega \tag{2.2a}$$

$$\mathbf{G}(\mathbf{x},\mathbf{y}) = 0 \qquad \mathbf{x} \in \partial\Omega,\ \mathbf{y} \in \Omega. \tag{2.2b}$$

where $\delta(\mathbf{x},\mathbf{y})$ is the Dirac delta function and $(\delta)_{ij} = \delta_{ij},\ i,j = 1,\dots m$ is the Kronecker delta symbol. The subscript $\mathbf{x}$ denotes differentiation with respect to the $\mathbf{x}$ variables. When $\Omega \equiv \mathrm{I\!R}^{\mathrm{d}}$, (2.2a) together with decaying boundary conditions at infinity defines the **fundamental solution tensor**.

Alternatively, one can choose to define the Green's tensor as the solution of an adjoint problem with respect to the second variable $\mathbf{y}$

$$\mathcal{L}^*_{\mathbf{y}}\mathbf{G}(\mathbf{x},\mathbf{y}) = \delta\ \delta(\mathbf{x},\mathbf{y}) \qquad \mathbf{x},\mathbf{y} \in \Omega \tag{2.3a}$$

$$\mathbf{G}(\mathbf{x},\mathbf{y}) = 0 \qquad \mathbf{y} \in \partial\Omega,\ \ \mathbf{x} \in \Omega \tag{2.3b}$$

which one can use to show that $\mathbf{u}$ can be expressed in integral form

$$\mathbf{u} = \int_{\Omega} \mathbf{G}\ \mathbf{f}\ d\Omega. \tag{2.4}$$

Finally, the Green's tensor can be seen as a perturbation of the fundamental solution tensor (or free space Green's tensor) $\mathcal{G}$:

$$\mathbf{G}(\mathbf{x},\mathbf{y}) = \mathcal{G}(\mathbf{x}-\mathbf{y}) + \mathbf{g}(\mathbf{x},\mathbf{y}), \tag{2.5}$$

where $\mathbf{g}(\mathbf{x},\mathbf{y})$ is the solution of

$$\mathcal{L}\mathbf{g}(\mathbf{x},\mathbf{y}) = 0 \qquad \mathbf{x},\mathbf{y} \in \Omega, \tag{2.6a}$$

$$\mathbf{g}(\mathbf{x},\mathbf{y}) = -\mathcal{G}(\mathbf{x}-\mathbf{y}) \qquad \mathbf{x} \in \partial\Omega, \quad \mathbf{y} \in \Omega. \tag{2.6b}$$

In general, Green's tensor formulae are not available. However, since we are in search of an *approximation* to the inverse (i.e., a preconditioner) we can use (2.5) to devise a preconditioner. More precisely, we choose the fundamental solution tensor as our continuous approximation to the Green's tensor $\mathbf{G}(\mathbf{x},\mathbf{y})$. By (2.6), it is evident that for fixed $\mathbf{y}$ the perturbation $g_{\mathbf{y}}(\mathbf{x}) = g(\mathbf{x},\mathbf{y})$ introduced in our approximation is small everywhere inside the domain, except in the vicinity of the boundary. We discuss this issue in section 4.

2.2. Green's tensor for the Oseen operator. Let us assume that in (2.1) $\Omega \equiv \mathbb{R}^2$ and the tensor $\mathcal{L}$ is the Oseen operator with constant coefficients (i.e., $\mathbf{b} =$ constant)

$$\mathcal{L} = \begin{pmatrix} -\varepsilon\Delta + \mathbf{b}\cdot\nabla & 0 & \partial/\partial x \\ 0 & -\varepsilon\Delta + \mathbf{b}\cdot\nabla & \partial/\partial y \\ \partial/\partial x & \partial/\partial y & 0 \end{pmatrix}.$$

To calculate the fundamental solution tensor $\mathcal{G}$ for $\mathcal{L}$ consider again the integral representation (2.4) in the whole space

$$\mathbf{u}(\mathbf{x}) = \int_{\mathbb{R}^2} \mathcal{G}(\mathbf{x}-\mathbf{y})\ \mathbf{f}(\mathbf{y})\ d\mathbf{y}. \tag{2.7}$$

Let $\hat{v}$ denote the Fourier transform of v, i.e.,

$$\hat{v}(\mathbf{k}) = \int_{\mathbb{R}^2} v(\mathbf{x}) e^{-i\mathbf{k}\cdot\mathbf{x}} d\mathbf{k}\ .$$

The Fourier space representation of (2.1a) is

$$\widehat{\mathcal{L}\mathbf{u}} = \widehat{\mathcal{L}}\widehat{\mathbf{u}} = \widehat{\mathbf{f}},$$

where

$$\widehat{\mathcal{L}}(\mathbf{k}) = \begin{pmatrix} \varepsilon k^2 + i\mathbf{b}\cdot\mathbf{k} & 0 & ik_1 \\ 0 & \varepsilon k^2 + i\mathbf{b}\cdot\mathbf{k} & ik_2 \\ ik_1 & ik_2 & 0 \end{pmatrix},$$

with $\mathbf{k}\cdot\mathbf{k} = k^2$.
Thus,

$$\begin{aligned} \mathbf{u}(\mathbf{x}) &= \int_{\mathbb{R}^d} \widehat{\mathcal{L}}^{-1}(\mathbf{k})\widehat{\mathbf{f}}(\mathbf{k}) e^{i\mathbf{k}\cdot\mathbf{x}} d\mathbf{k} \\ &= \int_{\mathbb{R}^d}\int_{\mathbb{R}^d} \widehat{\mathcal{L}}^{-1}(\mathbf{k})\mathbf{f}(\mathbf{y}) e^{i\mathbf{k}\cdot(\mathbf{x}-\mathbf{y})} d\mathbf{k} d\mathbf{y} \end{aligned}$$

and by comparison with (2.7) we deduce that

$$\mathcal{G}(\mathbf{x}) = \int_{\mathbb{R}^d} \widehat{\mathcal{L}}^{-1}(\mathbf{k}) e^{i\mathbf{k}\cdot\mathbf{x}} d\mathbf{k}, \tag{2.8}$$

i.e.,

$$\widehat{\mathcal{G}}(\mathbf{k}) = \widehat{\mathcal{L}}^{-1}(\mathbf{k}) = \frac{1}{k^2\alpha(\mathbf{k})} \begin{pmatrix} k_2^2 & -k_1k_2 & -ik_1\alpha(\mathbf{k}) \\ -k_1k_2 & k_1^2 & -ik_2\alpha(\mathbf{k}) \\ -ik_1\alpha(\mathbf{k}) & -ik_2\alpha(\mathbf{k}) & \alpha(\mathbf{k})^2 \end{pmatrix},$$

where $\alpha(\mathbf{k}) = \varepsilon k^2 + i\mathbf{b}\cdot\mathbf{k}$.
Hence, using the properties of the Fourier transform, we find

$$\mathcal{G}(\mathbf{x}) = \begin{pmatrix} -\frac{\partial^2}{\partial y^2}\mathcal{G}_{\mathcal{Q}} & \frac{\partial^2}{\partial x\partial y}\mathcal{G}_{\mathcal{Q}} & -\frac{\partial}{\partial x}\mathcal{G}_{\Delta} \\ \frac{\partial^2}{\partial x\partial y}\mathcal{G}_{\mathcal{Q}} & -\frac{\partial^2}{\partial x^2}\mathcal{G}_{\mathcal{Q}} & -\frac{\partial}{\partial y}\mathcal{G}_{\Delta} \\ -\frac{\partial}{\partial x}\mathcal{G}_{\Delta} & -\frac{\partial}{\partial y}\mathcal{G}_{\Delta} & (-\varepsilon\Delta + \mathbf{b}\cdot\nabla)\mathcal{G}_{\Delta} \end{pmatrix},$$

where $\mathcal{G}_\Delta$ is the fundamental solution for the Laplacian and $\mathcal{G}_\mathcal{Q}$ is the fundamental solution for the singularly perturbed fourth order operator $\mathcal{Q} = \Delta(-\varepsilon\Delta + \mathbf{b}.\nabla)$.
If $\Omega \equiv \mathbb{R}^3$, we similarly find

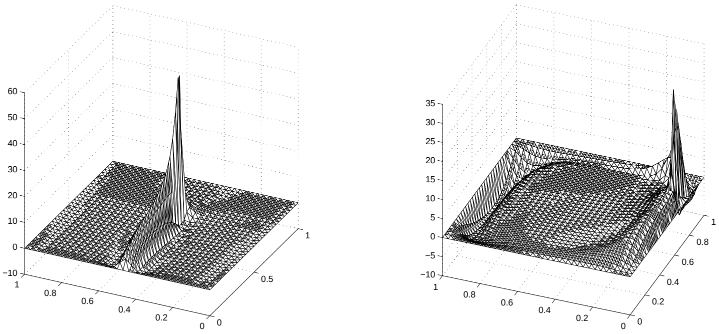
$$\mathcal{G}(\mathbf{x}) = \begin{pmatrix} -(\frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2})\mathcal{G}_{\mathcal{Q}} & \frac{\partial^2}{\partial x\partial y}\mathcal{G}_{\mathcal{Q}} & \frac{\partial^2}{\partial x\partial z}\mathcal{G}_{\mathcal{Q}} & -\frac{\partial}{\partial x}\mathcal{G}_{\Delta} \\ \frac{\partial^2}{\partial x\partial y}\mathcal{G}_{\mathcal{Q}} & -(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial z^2})\mathcal{G}_{\mathcal{Q}} & \frac{\partial^2}{\partial y\partial z}\mathcal{G}_{\mathcal{Q}} & -\frac{\partial}{\partial y}\mathcal{G}_{\Delta} \\ \frac{\partial^2}{\partial x\partial z}\mathcal{G}_{\mathcal{Q}} & \frac{\partial^2}{\partial y\partial z}\mathcal{G}_{\mathcal{Q}} & -(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2})\mathcal{G}_{\mathcal{Q}} & -\frac{\partial}{\partial z}\mathcal{G}_{\Delta} \\ -\frac{\partial}{\partial x}\mathcal{G}_{\Delta} & -\frac{\partial}{\partial y}\mathcal{G}_{\Delta} & -\frac{\partial}{\partial z}\mathcal{G}_{\Delta} & (-\varepsilon\Delta + \mathbf{b}\cdot\nabla)\mathcal{G}_{\Delta} \end{pmatrix},$$

where the Laplacian Δ and $\mathcal{Q}$ defined as above are now three-dimensional operators.

Remarks. (i) Peculiarly enough, the last column in the fundamental solution tensor is not necessary in order to write the solution in the form (2.7) (see [11, section VII.3]). This is due to the fact that the divergence free conditions (1.1b, 1.2b) lead to a right hand side of the form $(\mathbf{f},\ 0)^T$ which gives no contributions to the solution in the form of convolutions with the entries of the last column. This is equivalent to saying that the discrete operator can afford a **rectangular** preconditioner. Indeed, if one were to construct a rectangular matrix $\mathcal{P}$ such that

$$\mathcal{A}\mathcal{P} = \begin{pmatrix} I \\ 0 \end{pmatrix}, \tag{2.9}$$

then $\mathcal{P}$ will guarantee convergence in one iteration for any iterative solver. However, in practice, any rectangular matrix which approximates $\mathcal{P}$ in (2.9) is bound to have a different kernel and thus is certain to perform poorly with an iterative solver.
(ii) The continuous version of the Schur complement is the same in two and three dimensions: $(-\varepsilon\Delta + \mathbf{b}\cdot\nabla)\mathcal{G}_\Delta$. Though we will concentrate on the two-dimensional case in this paper, we believe that the three-dimensional choice of preconditioner is the natural multidimensional generalisation.
(iii) In the above, we considered only the case of constant wind $\mathbf{b}$, which seems to indicate directionality of the diagonal entries of $\mathcal{G}$ (see figure 2.1(a)). For the non-constant wind too, this directionality is preserved, as seen in the numerical plot of figure 2.1(b). This fact indicates that a preconditioner based on the analytic inverse



(a) $\mathbf{b} = (0, 1)$

(b) $\mathbf{b}$ = circular (as in (4.3))

Fig. 2.1: Numerical Green's functions (G_{11}) for the Oseen operator.

of a constant coefficient operator may well generalise to the case of non-constant coefficients, as is the case with other operators such as advection-diffusion [15].

Our motivation for looking at the fundamental solution tensor originated from the attempt to approximate the inverse of the Schur complement, which was shown to be also the (3,3) block of the inverse of system matrix (1.5). In the two-dimensional case the continuous counterpart of the Schur complement is

$$\mathcal{G}_{33}(\mathbf{x}) = (-\varepsilon\Delta + \mathbf{b}\cdot\nabla)\mathcal{G}_\Delta(\mathbf{x}) \tag{2.10}$$

In the next section we will be looking at the discrete version of this operator which, as we shall see, provides an efficient replacement for the Schur complement in a preconditioner of the form (1.8).

3. A preconditioner for the Oseen equations. The function (2.10) has no discrete counterpart arising from the discretization of the original Oseen problem. Its discretization should represent a convolution on the pressure space with a function (kernel) involving a convection-diffusion operator acting on the inverse Laplacian. Physically, this has no clear meaning. However, there is a precedent which can help us devise the preconditioner.

Consider the limiting case of the Oseen equations, the Stokes problem, with operator

$$\mathcal{L} = \begin{pmatrix} -\varepsilon\Delta & 0 & \partial/\partial x \\ 0 & -\varepsilon\Delta & \partial/\partial y \\ \partial/\partial x & \partial/\partial y & 0 \end{pmatrix}.$$

An analysis similar to that in the previous section leads to the Green's tensor for the Stokes operator:

$$\mathcal{G}(\mathbf{x}) = \begin{pmatrix} -\frac{\partial^2}{\partial y^2}\mathcal{G}_Q & \frac{\partial^2}{\partial x \partial y}\mathcal{G}_Q & -\frac{\partial}{\partial x}\mathcal{G}_\Delta \\ \frac{\partial^2}{\partial x \partial y}\mathcal{G}_Q & -\frac{\partial^2}{\partial x^2}\mathcal{G}_Q & -\frac{\partial}{\partial y}\mathcal{G}_\Delta \\ -\frac{\partial}{\partial x}\mathcal{G}_\Delta & -\frac{\partial}{\partial y}\mathcal{G}_\Delta & \varepsilon\delta \end{pmatrix},$$

where $\mathcal{G}_\mathcal{Q}$ is the fundamental solution for the scaled biharmonic operator $\mathcal{Q} = \varepsilon\Delta\Delta$ and δ is the Dirac delta function. By the same argument, if one were to search for a good replacement for the Schur complement, then a promising choice would be a viscosity-scaled discrete delta function. What is a discrete delta function? To see this, consider the 'equation'

$$u = f \quad \text{in } \mathbb{R}^2. \tag{3.1}$$

Green's function for the identity is the delta function and since the discrete version of (3.1) is

$$M\mathbf{u} = \mathbf{f},$$

where M is the mass matrix, we conclude that the discrete version of the delta function is an inverse mass matrix. Hence, the alternative for the Schur complement would be

$$X_M = M/\varepsilon. \tag{3.2}$$

This is the choice that Wathen and Silvester ([24]) investigated in the context of using block-diagonal preconditioners for the Stokes problem and for which they showed optimality in the sense that the eigenvalues of the preconditioned Stokes coefficient matrix are bounded above and below *and* away from the origin independently of the discretization.

Note that Green's tensor for the Oseen operator tends in the limit $\mathbf{b} \to 0$ to Green's tensor for the Stokes operator. Thus, one can view the discrete version of (2.10) as a generalization of (3.2). Hence, our preconditioner based on the fundamental solution should default to the optimal Stokes preconditioner in the limit as $\mathbf{b} \to 0$.

Note first that a straightforward interpretation of (2.10) would give

$$X^{-1} = F_p A_p^{-1},$$

where F_p and A_p are discrete versions of $-\varepsilon\Delta + \mathbf{b}\cdot\nabla$ and Δ respectively. The subscript p indicates that they act on the pressure space and that they should be discretized accordingly. However, this choice of X will not default to X_M for the Stokes problem but to εI. For this reason we include the mass matrix in our preconditioner, so that our choice becomes

$$X_\mathcal{G}^{-1} = M_p^{-1} F_p A_p^{-1}. \tag{3.3}$$

Remarks (i) There is another way we could introduce the mass matrix in our preconditioner, namely $X_\mathcal{G}^{-1} = F_p A_p^{-1} M_p^{-1}$. However, (3.3) performed marginally better in our tests and so we included numerical results only for this choice.
(ii) Note that $\mathcal{G}_{33}$ in (2.10) is a function and not a differential operator. Hence, the order in which we apply F_p and A_p^{-1} is as given in (3.3) and not otherwise.

4. Numerical results. In this section we present a comparative study of the performance of our preconditioner together with those suggested in the literature [5], [6] and in [7].

1. The test problems. Our test problems were chosen from the large body of literature on the Navier-Stokes equations.
(i) Pipe flow;
(ii) Driven cavity;

(iii) Backward facing step;
(iv) Flow past a cylinder.
The domains and boundary conditions are detailed in the following subsections. The viscosity varied from 1/10 to 1/1,000, according to the problem. Note that for some problems the onset of unsteady flow prevented us from decreasing the viscosity further.

2. The discretizations. The tests were conducted on 4 types of discretizations of the Oseen equations:
(i) the finite difference MAC scheme [12];
(ii) mixed finite element formulation on triangles using a quadratic approximation for the velocities and a linear approximation for the pressure (P2P1) [13];
(iii) mixed finite element formulation on triangles using a linear approximation for the velocities and pressure (P1P1) ;
(iv) mixed finite element formulation on rectangles using a bilinear approximation for the velocities and pressure (Q1Q1).
Note that the first two are stable discretizations of the Oseen equations in the LBB sense ([3, p. 205]), while the other two are stabilised formulations (see [9], [10], [18]).
Remarks. (i) For high Reynolds number flows, standard Galerkin discretization of the advection-diffusion operator is well known to be numerically unstable. For this reason, we chose the streamline diffusion method (SDM) originally suggested by Hughes and Brookes in [14] with parameter choice as in [8].
(ii) In general, to the best of our knowledge, we implemented our preconditioner with the most accurate representation of the continuous operator on a given mesh. This involved tuning various stabilisation/SDM parameters. Clearly, the performance of our preconditioner would deteriorate in the same way the actual solution would deteriorate, since an inaccurate discrete operator will lead to a poor representation of the discrete inverse.

3. The solvers. The iterative solver we used was full GMRES [21] with right preconditioning. The initial guess was the zero vector and the stopping criterion was the relative residual brought below a tolerance of 10^{-6}. We implemented preconditioner (1.8) with our approximation $X_{\mathcal{G}} = M_p F_p^{-1} A_p$ (cf. 3.3) of the Schur complement together with three other existing choices:
(i) $X_M = M_p/\varepsilon$ ([6])
(ii) $X_B = (BB^T)^{-1} B\mathbf{F}_p B^T (BB^T)^{-1}$ ([5]);
(iii) $X_R = (X_M + X_B)/2$ ([7]). In the following, the subscript of the preconditioner $\mathcal{P}$ will denote the respective choice of Schur complement approximation (e.g. $\mathcal{P}_M$ is preconditioner (1.8) with X replaced by X_M).

4. The implementation. From the factorisation of the inverse of (1.8)

$$\mathcal{P}^{-1} = \begin{pmatrix} \mathbf{F}^{-1} & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} I & B^T \\ 0 & -I \end{pmatrix} \begin{pmatrix} I & 0 \\ 0 & X^{-1} \end{pmatrix}, \tag{4.1}$$

we see that implementing the preconditioner involves computing the action of only two inverses : $\mathbf{F}^{-1}$ and X^{-1}. The first is a vector advection-diffusion solve for which efficient preconditioners have been devised ([1], [2], [15], [19]). The action of X^{-1} is also inexpensive computationally, with the possible exception of X_B (and thus X_R) as detailed below.
(i) X_M^{-1} can be approximated cheaply in a conjugate gradient code (CG), with a diagonal preconditioner [22], [23].
(ii) The implementation of X_B (and thus X_R) is not very clear for the case of most finite element discretizations (P2P1, P1P1, Q1Q1), due to the BB^T term which cannot be inverted in an inexpensive straightforward manner. However, for the MAC

finite difference scheme, BB^T turns out to be the common five-point finite difference Laplacian for which multigrid is an optimal solver.
(iii) The action of $X_{\mathcal{G}}$ requires the inverse of X_M as in (i) together with a solve for the Laplacian for which optimal multigrid codes exist [25].
To sum up, the cheapest preconditioner is $\mathcal{P}_M$, while the cost of $\mathcal{P}_{\mathcal{G}}$ is comparable with that of $\mathcal{P}_B$ (if the inverse of the latter exists).

In section 4.1 we present the results for the exact implementation of the preconditioners, which should give us an insight into their asymptotic behaviour. However, from a practical point of view we are interested in minimising the cost of each iteration. For this reason, further approximations were introduced, via multigrid and CG solutions to the inverses involved in the above preconditioners. These results are presented in section 4.2. Finally, section 4.3 considers the performance of our preconditioner when the issue of adaptivity is addressed.

4.1. Exact Preconditioning.

Problem I. The first problem to be considered is the Oseen equations (1.2), applied to pipe-flow on the domain $\Omega = (0,1) \times (0,1)$, with $\mathbf{w} = (1,0)$. Applying the constraint $\int_\Omega p \, \mathrm{d}\mathbf{x} = 0$ and the boundary conditions $u_1 = 4(y - y^2)$, $u_2 = 0$ on Γ gives the exact solution

$$\mathbf{u} = \begin{pmatrix} 4(y-y^2) \\ 0 \end{pmatrix}, \quad p = 4\epsilon - 8\epsilon x. \tag{4.2}$$

From tables 4.1, 4.2 and 4.3 we clearly see that the exact preconditioner seems to

h	1/32	1/64	1/128
$\epsilon = 1/100$	28	27	25
$\epsilon = 1/500$	54	46	42
$\epsilon = 1/1000$	73	61	55

Table 4.1: *GMRES iteration counts for the $P1P1$ solution to the pipe-flow problem using $\mathcal{P}_{\mathcal{G}}$ and uniform refinement .*

have a dependence on viscosity of order $O(\varepsilon^{-1/3})$. Furthermore, we see that as the mesh is refined the number of GMRES counts *decreases* up to the point where the mesh Péclet number becomes sufficiently small, i.e., when the finite dimensional operator becomes a good representation of the continuous operator. Note that this would be the mesh size where the discrete Green's function is a good approximation of the continuous Green's function. This feature of *reducing* iteration counts with an increasing dimension of the discrete problem (which is generally never seen for algebraically constructed preconditioners) is common to all the results we have computed with $\mathcal{P}_{\mathcal{G}}$.

Problem II. The second problem to be considered is the lid driven cavity problem on the domain $\Omega = (0,1) \times (0,1)$ with wind

$$\mathbf{b} = \begin{pmatrix} 2(2y-1)(1-(2x-1)^2) \\ -2(2x-1)(1-(2y-1)^2) \end{pmatrix}. \tag{4.3}$$

and boundary conditions $\mathbf{u}(x,1) = (1,0)^T$, $\mathbf{u}(x,y) = \mathbf{0}$ elsewhere. The iteration counts for the MAC scheme discretization of this problem are shown in table 4.4. Both $\mathcal{P}_B$ and $\mathcal{P}_{\mathcal{G}}$ exhibit similar viscosity dependence. However, the performance of

h	1/32	1/64	1/128
$\epsilon = 1/100$	36	36	35
$\epsilon = 1/500$	71	65	63
$\epsilon = 1/1000$	95	87	80

Table 4.2: *GMRES iteration counts for the $Q1Q1$ solution to the pipe-flow problem using $\mathcal{P}_{\mathcal{G}}$ and uniform refinement.*

h	1/16	1/32	1/64
$\epsilon = 1/100$	29	29	27
$\epsilon = 1/500$	56	52	48
$\epsilon = 1/1000$	69	70	67

Table 4.3: *GMRES iteration counts for the pipe-flow problem using the finite element discretization $P2P1$, preconditioner $\mathcal{P}_{\mathcal{G}}$ and uniform refinement.*

$\mathcal{P}_B$ seems to deteriorate with the refinement. Note that a MAC scheme discretization of a problem with periodic boundary conditions leads to $\mathcal{P}_B$ and $\mathcal{P}_{\mathcal{G}}$ being the same (see [5, pp. 5–6])

$$B\mathbf{F}^{-1}B^tX_B^{-1} = B\mathbf{F}^{-1}B^tX_{\mathcal{G}}^{-1} = I.$$

The results for $P2P1$ discretizations are presented in table 4.5. Note that preconditioner $\mathcal{P}_M$ gives a dependence on viscosity proportional to ϵ^{-1}. Furthermore, the iteration counts grow with refinement while the mesh Péclet number is large. For the preconditioners $\mathcal{P}_B$ and $\mathcal{P}_R$ we see viscosity dependence no worse than $O(\epsilon^{-1/2})$ and dependence on refinement proportional to $h^{-1/2}$. Finally, for the preconditioner $\mathcal{P}_{\mathcal{G}}$ we see a dependence on viscosity which is bounded above by $O(\epsilon^{-1/3})$ while no dependence on refinement is observed.

Problem III. We consider a stationary Navier-Stokes problem of flow past a backward facing step. The boundary conditions are detailed in figure 4.1. Table 4.6 shows that $\mathcal{P}_B$ fails and that the use of $\mathcal{P}_M$ or $\mathcal{P}_R$ would be in no way practical for problems of this nature. This is due to the fact that these preconditioners were devised algebraically, for a coefficient matrix (1.5) with a zero (3,3) block. Once stabilization is necessary, the preconditioner $\mathcal{P}_B$ fails to be a good approximation. However, $\mathcal{P}_{\mathcal{G}}$ was devised for *any* reasonable discretization of the continuous problem. The stabilization matrix C in (1.5) does not affect the preconditioner, since it is meant to improve the

ϵ	h	$\mathcal{P}_B$	$\mathcal{P}_{\mathcal{G}}$
1/10	1/16	11	14
	1/32	14	15
	1/64	18	14
1/100	1/16	20	25
	1/32	21	27
	1/64	24	28

Table 4.4: *GMRES iteration counts for the lid driven cavity using the MAC scheme.*

ϵ	h	$\mathcal{P}_M$	$\mathcal{P}_B$	$\mathcal{P}_R$	$\mathcal{P}_{\mathcal{G}}$
1/10	1/4	14	10	10	15
	1/8	17	16	16	15
	1/16	19	25	24	15
	1/32	17	38	35	15
1/20	1/4	17	11	11	18
	1/8	23	17	17	17
	1/16	27	28	26	16
	1/32	27	41	39	16
1/50	1/4	19	11	12	21
	1/8	30	18	19	20
	1/16	43	29	28	21
	1/32	55	43	43	21
1/100	1/4	20	13	13	22
	1/8	38	20	20	25
	1/16	62	32	32	26
	1/32	87	48	48	26

Table 4.5: *GMRES iteration counts for the lid driven cavity using the finite element discretization $P2P1$.*

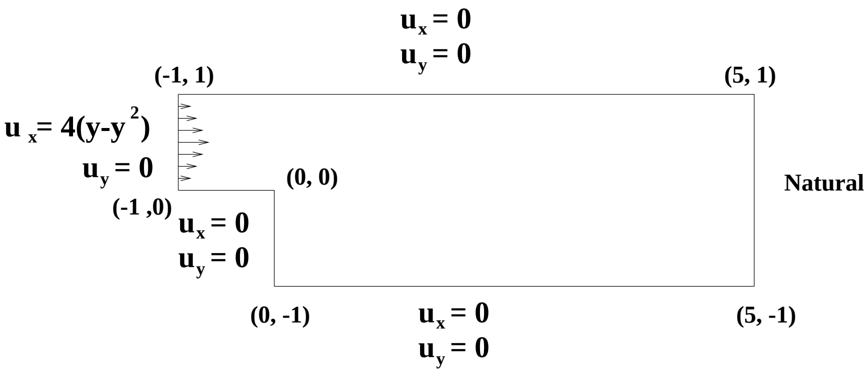


Fig. 4.1: *Backward facing step boundary conditions.*

discrete representation of the operator. Thus, in table 4.6 we see no h dependence for $\mathcal{P}_{\mathcal{G}}$ and from tables 4.7 and 4.8 we again see a viscosity dependence bounded above by $O(\epsilon^{-1/3})$ for both $P1P1$ and $P2P1$ discretizations.

4.2. Inexact preconditioning. The exact preconditioning experiments have shown $\mathcal{P}_{\mathcal{G}}$ consistently outperforming the other preconditioners. Henceforth we will only consider the performance of preconditioner $\mathcal{P}_{\mathcal{G}}$. For the preconditioner to be practical we must be able to cheaply obtain the inverses or good approximations to the inverses of the matrices M_p, A_p and $\mathbf{F}$. This is achieved as follows:

- M_p: *Preconditioned Conjugate Gradient*
  A good approximation to the inverse of this matrix can be obtained using the preconditioned conjugate gradient method. Jacobi preconditioning is suf-

h_{min} (DOF)	1/4 (333)	1/8 (1191)	1/16 (4491)	1/32 (17427)
$\mathcal{P}_M$	54	96	$\gg 100$	$\gg 100$
$\mathcal{P}_B$	$\gg 100$	$\gg 100$	$\gg 100$	$\gg 100$
$\mathcal{P}_R$	52	69	$\gg 100$	$\gg 100$
$\mathcal{P}_\mathcal{G}$	45	44	41	37

Table 4.6: *GMRES iteration counts for the $P1P1$ solution to the backward facing step problem; $\epsilon = 1/100$.*

h_{min} (DOF)	1/4 (333)	1/8 (1191)	1/16 (4491)	1/32 (17427)
$\epsilon = 2/25$	23	20	19	18
$\epsilon = 1/25$	24	21	21	19
$\epsilon = 1/50$	31	28	27	26
$\epsilon = 1/100$	45	44	41	37

Table 4.7: *GMRES iteration counts for the $P1P1$ solution to the backward facing step problem, using preconditioner $\mathcal{P}_\mathcal{G}$.*

h_{min} (DOF)	1/2 (256)	1/4 (905)	1/8 (3391)	1/16 (13115)
$\mathcal{P}_M$	37	97	$\gg 100$	$\gg 100$
$\mathcal{P}_B$	25	41	63	84
$\mathcal{P}_R$	25	41	56	71
$\mathcal{P}_\mathcal{G}$	34	47	42	41

Table 4.8: *GMRES iteration counts for the $P2P1$ solution to the backward facing step problem using preconditioner $\mathcal{P}_\mathcal{G}$; $\varepsilon = 1/100$.*

ficient for mesh-independent results [22],[23]. In the following we will apply 5 iterations of the preconditioned CG algorithm.

- A_p: *Multigrid*
  This matrix is a discrete Laplacian. A good approximation to the inverse may be obtained using multigrid, see [25]. In the following we will apply a very simple multigrid procedure with 2 V-cycles and simple point Gauss–Seidel smoothing.
- **F**: *Multigrid*
  Solving systems with this matrix is not as simple, due to the convection term. For reasonable viscosity it has been shown [19] that multigrid is a good preconditioner for convection-diffusion problems which are stabilised using streamline-diffusion. In the following we will apply multigrid with 2 V-cycles with forward and backward Gauss–Seidel smoothing.

We denote by $\mathcal{P}_{\mathcal{G},1}$ the preconditioner $\mathcal{P}_\mathcal{G}$ used with inexact solves applied to M_p and A_p and an exact solve used for $\mathbf{F}$. $\mathcal{P}_{\mathcal{G},2}$ will denote the use of inexact solves for M_p, A_p and $\mathbf{F}$.

For the lid driven cavity problem, table 4.9 shows iteration counts for $\mathcal{P}_{\mathcal{G},1}$ similar to those for $\mathcal{P}_\mathcal{G}$. This seems to indicate that the approximations introduced by multigrid and PCG are indeed reasonable.

h (DOF)	1/8 (187)	1/16 (659)	1/32 (2467)	1/64 (9539)
$\epsilon = 1/10$	15	16	17	17
$\epsilon = 1/20$	17	18	19	19
$\epsilon = 1/50$	20	23	25	25
$\epsilon = 1/100$	21	28	29	32

Table 4.9: *GMRES iteration counts for $P2P1$ solution to the lid driven cavity problem, with preconditioner $\mathcal{P}_{\mathcal{G},1}$.*

h_{min} (DOF)	1/4 (333)	1/8 (1191)	1/16 (4491)	1/32 (17427)
$\mathcal{P}_{\mathcal{G},1}$	44	45	41	39
$\mathcal{P}_{\mathcal{G},2}$	46	45	41	39

Table 4.10: *GMRES iteration counts for the $P1P1$ solution to the backward facing step problem; $\epsilon = 1/100$*

For the backward facing step problem, table 4.10 shows that the inexact solve on **F** does not affect the iteration counts. Furthermore, comparing tables 4.6 and 4.10 we see only a slight increase in the iteration counts when using $\mathcal{P}_{\mathcal{G},2}$ versus using $\mathcal{P}_{\mathcal{G}}$.

4.3. Adapted mesh refinement. In this section we investigate the performance of our preconditioner applied to a sequence of nested adaptively refined meshes. An error indicator is used to select the elements to be refined. The choice of refinement is such that the aspect ratio of the triangles can be no worse than twice that of the original aspect ratio.

Two problems are considered:
(i) flow past a backward facing step;
(ii) flow past a cylinder.

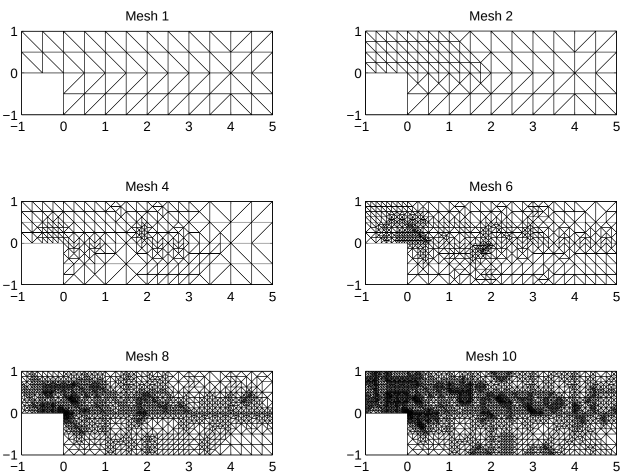


Fig. 4.2: *Adapted meshes for backward facing step.*

A selection of mesh refinements is given in figure 4.2. The numerical results are

Level	2 (306)	4 (720)	6 (1932)	8 (5427)	10 (9288)
$\mathcal{P}_{\mathcal{G}}$	46	46	41	41	39
$\mathcal{P}_{\mathcal{G},2}$	48	50	43	42	40

Table 4.11: *GMRES iteration counts for the $P1P1$ solution to the backward facing step using adaptive mesh refinement;* $\epsilon = 1/100$.

shown in table 4.11. Again, they seem to indicate no mesh dependence. Moreover, the use of multigrid on the adapted meshes does not seem to significantly affect the iteration counts.

For the problem of flow past a cylinder, the boundary conditions are detailed in figure 4.3; the sequence of mesh refinements is shown in figure 4.4. In table 4.12 iteration counts decrease with refinement and the use of $\mathcal{P}_{\mathcal{G},2}$ is as effective as using $\mathcal{P}_{\mathcal{G}}$.

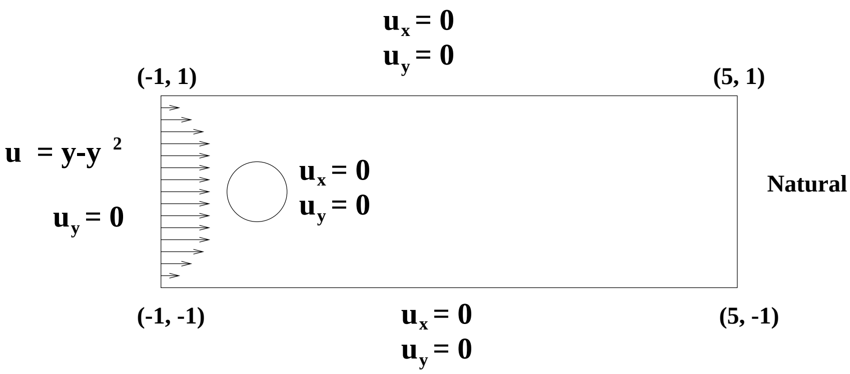


Fig. 4.3: *Boundary conditions for the flow past a cylinder problem*

Level	1 (264)	2 (540)	3 (1242)	4 (2709)	5 (5484)	6 (9003)
$\mathcal{P}_{\mathcal{G}}$	49	49	43	43	40	38
$\mathcal{P}_{\mathcal{G},2}$	49	51	44	44	41	40

Table 4.12: *GMRES iteration counts for the $P1P1$ solution to the flow past a cylinder problem using adaptive mesh refinement,* $\epsilon = 1/50$

5. Conclusion. In this paper we used a novel approach to preconditioning in order to devise a fast, efficient solver for the steady-state Navier-Stokes equations. Our starting point was the ideal, though impractical preconditioner (1.8), which required the inverse of a full matrix: the Schur complement. We circumvented this difficulty by constructing this inverse via inversion and multiplication of sparse matrices. The choice of these discrete operators was suggested by the continuous inverse of the Oseen operator, the fundamental solution tensor. The resulting preconditioner exhibits a mild dependence on the viscosity and no mesh dependence at all. This remarkable

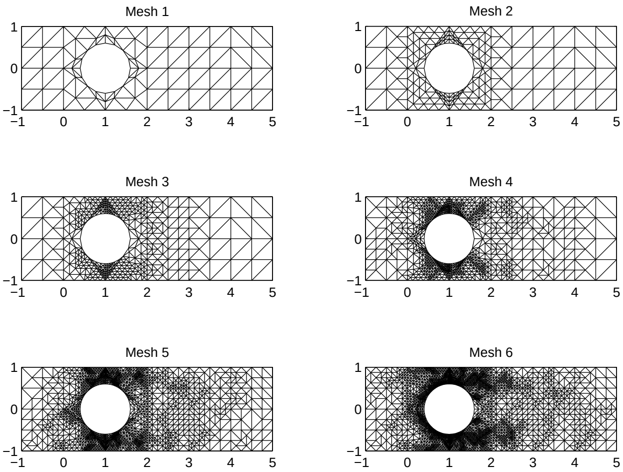


Fig. 4.4: *Adapted meshes for the flow past a cylinder problem*

property, namely improved performance under mesh refinement, is typical of preconditioners based on Green's functions [15]. Though the results in this paper cover various two-dimensional problems, the choice of preconditioner for 3D was exhibited in section 3 and we expect the results to be rather similar. Last, but not least, we must stress the generality of Green's function approach to preconditioning problems arising from the field of partial differential equations.

REFERENCES

[1] O. Axelsson, V. Eijkhout, B. Polman, and P. Vassilevski, *Incomplete block-matrix factorization iterative methods for convection-diffusion problems*, BIT, 29 (1989), pp. 867–889.

[2] J. Bey and G. Wittum, *Downwind numbering: Robust multigrid for convection-diffusion problems*, Appl. Num. Math., 23 (1997), pp. 177–192.

[3] F. Brezzi and M. Fortin, *Mixed and hybrid finite element methods*, Springer-Verlag, New York, 1991.

[4] C. Cuvelier, A. Segal, and A. van Steenhoven, *Finite Element Methods and the Navier-Stokes*, D. Reidel Publishing Co., 1986.

[5] H. C. Elman, *Preconditioning for the steady-state navier-stokes equations with low viscosity*, SIAM J. Sci. Comp., 20 (1999), pp. 1299–1316.

[6] H. C. Elman and D. J. Silvester, *Fast nonsymmetric iterations and preconditioning for Navier-Stokes equations*, SIAM J. Sci. Comp., 17 (1996), pp. 33–46.

[7] ———, *Private communication*, 1998.

[8] B. Fischer, A. Ramage, D. J. Silvester, and A. J. Wathen, *Towards parameter-free streamline upwinding for advection-diffusion problems*, to appear in Comp. Meths. Appl. Mech. Engrg., (1999).

[9] L. Franca and S. Frey, *Stabilised finite element methods: II. the incompressible Navier-Stokes equations*, Computer Methods in Applied Mechanical Engineering, 99 (1992), pp. 209–233.

[10] L. Franca, T. J. R. Hughes, and R. Stenberg, *Stabilised finite element methods*, in Incompressible computational fluid dynamics, M. Gunzburger and R. Nicolaides, eds., Cambridge University Press, 1993, pp. 87–107.

[11] G. Galdi, *An introduction to the mathematical theory of the Navier-Stokes equations*, vol. 38 of Springer Tracts in Natural Philosophy, Springer, 1994.

[12] F. H. Harlow and J. E. Welch, *MAC scheme*, Phys. Fluids, 8 (1965).

[13] P. Hood and C. Taylor, *A numerical solution of the Navier-Stokes equations using the finite*

element technique, Comp. and Fluids, (1973), pp. 73–100.
[14] T. J. R. Hughes and A. N. Brooks, *A multidimensional upwind scheme with no crosswind diffusion*, in Finite Element Methods for Convection Dominated Flows, T. J. R. Hughes, ed., vol. 34 of AMD, ASME, New York, 1979.
[15] D. Loghin and A. J. Wathen, *Preconditioning the advection-diffusion equation: the Green's function approach*, Tech. Rep. NA–15/97, Oxford University Computing Laboratory, 1997.
[16] M. F. Murphy, G. H. Golub, and A. J. Wathen, *A note on preconditioning for indefinite linear systems*, submitted to SIAM J. Sci. Comp., (1999).
[17] M. F. Murphy and A. J. Wathen., *On preconditioners for the Oseen equations.*, Tech. Rep. AM-95-07, Bristol University, July 1995.
[18] S. Norburn and D. J. Silvester, *Stabilised vs. stable mixed methods for incompressible flow*, Comp. Meths. Appl. Mech. Engrg., 166 (1998), pp. 131–141.
[19] A. Ramage, *A multigrid preconditioner for stabilised discretisations of advection-diffusion problems*, Tech. Rep. 33, Strathclyde University, August 1998.
[20] Y. Saad, *Iterative Methods for Sparse Linear Systems*, PWS Publishing Co., Boston, 1996.
[21] Y. Saad and M. Schultz, *GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Stat. Comp., 7 (1986), pp. 856–869.
[22] A. J. Wathen, *Realistic eigenvalue bounds for the Galerkin mass matrix*, IMA J. Numer. Anal., (1987), pp. 449–457.
[23] ———, *On relaxation of Jacobi iteration for consistent and generalized mass matrices*, Comm. Appl. Num. Meths., 7 (1991), pp. 93–102.
[24] A. J. Wathen and D. J. Silvester, *Fast iterative solution of stabilised Stokes systems.*, SIAM J. Num. Anal., 30 (1993), pp. 630–649.
[25] P. Wesseling, *An introduction to multigrid*, Wiley, 1991.