Journal of Computational Physics **164,** 258–282 (2000)
doi:10.1006/jcph.2000.6593, available online at http://www.idealibrary.com on IDEAL®

Parallel Computation of Flow in Heterogeneous Media Modelled by Mixed Finite Elements[1]

K. A. Cliffe,* I. G. Graham,† R. Scheichl,† and L. Stals‡

**AEA Technology, Harwell, Didcot, Oxon OX11 0RA, United Kingdom; †Department of Math. Sciences, University of Bath, Bath BA2 7AY, United Kingdom; and ‡Department of Computer Science, Old Dominion University, Norfolk, Virginia 23529-0162*
E-mail: andrew.cliffe@aeat.co.uk, igg@maths.bath.ac.uk, maprs@maths.bath.ac.uk, stals@icase.edu

Received August 2, 1999; revised July 12, 2000

In this paper we describe a fast parallel method for solving highly ill-conditioned saddle-point systems arising from mixed finite element simulations of stochastic partial differential equations (PDEs) modelling flow in heterogeneous media. Each realisation of these stochastic PDEs requires the solution of the linear first-order velocity–pressure system comprising Darcy's law coupled with an incompressibility constraint. The chief difficulty is that the permeability may be highly variable, especially when the statistical model has a large variance and a small correlation length. For reasonable accuracy, the discretisation has to be extremely fine. We solve these problems by first reducing the saddle-point formulation to a symmetric positive definite (SPD) problem using a suitable basis for the space of divergence-free velocities. The reduced problem is solved using parallel conjugate gradients preconditioned with an algebraically determined additive Schwarz domain decomposition preconditioner. The result is a solver which exhibits a good degree of robustness with respect to the mesh size as well as to the variance and to physically relevant values of the correlation length of the underlying permeability field. Numerical experiments exhibit almost optimal levels of parallel efficiency. The domain decomposition solver (DOUG, `http://www.maths.bath.ac.uk/~parsoft`) used here not only is applicable to this problem but can be used to solve general unstructured finite element systems on a wide range of parallel architectures. © 2000 Academic Press

Key Words: Raviart–Thomas mixed finite elements; second-order elliptic problems; divergence-free space; heterogeneous media; groundwater flow; domain decomposition; parallel computation.

[1] This work was supported in part by EPSRC Grant GR/L31715 and EPSRC CASE Award 97D00023.

1. INTRODUCTION

In this paper we describe, analyse, and implement a parallel algorithm for use in the numerical simulation of stochastic partial differential equations (PDEs) modelling groundwater flow in heterogeneous media. The classical equations governing this application are the first-order system consisting of *Darcy's law*,

$$\vec{q} + (k/\mu)\vec{\nabla} P_R = \vec{0}, \tag{1.1}$$

coupled with the *mass conservation law*,

$$\vec{\nabla} \cdot \vec{q} = 0, \tag{1.2}$$

subject to appropriate boundary conditions. Here $\vec{q}$ is the *velocity* (more precisely the specific discharge), and P_R is the *residual pressure*. The actual pressure is $P_R - \rho g z$, where z is the fluid height, ρ is the density, and g is the acceleration due to gravity. The functions $\vec{q}$ and P_R are both to be determined from (1.1) and (1.2), with k denoting permeability and μ denoting the dynamic viscosity.

In contrast to the classical determinisic models, k will, in the probabilistic case considered here, be modelled using a Gaussian random field. The numerical treatment of this problem then involves the solution of (1.1) and (1.2) for many different *realisations* of k and subsequent computation of statistical properties of the resulting velocity and/or pressure fields. In this paper we describe a fast parallel method for the solution of (1.1) and (1.2) when k is any typical realisation. Although we do not carrry out here any statistical analysis involving multiple simulations, a prime motivation of our work is to obtain a numerical method which is sufficiently accurate and efficient to make such a statistical analysis possible. For reasons described below, a typical simulation of (1.1) and (1.2) will involve the solution of very large, highly ill-conditioned indefinite linear systems and the fast method proposed here constitutes an essential tool which can be used in later statistical analyses. Because of applications in waste management and in the oil industry, there is a strong technological motivation for such a tool.

The Gaussian random fields which determine k are characterised by a pair of parameters (σ^2, λ), where σ^2 is the *variance* and λ is the *length scale* over which the field is correlated. In addition, the discrete model also depends on n, the number of degrees of freedom in the grid, although λ and n are typically related to each other. Extreme variations in these parameters contribute to the ill-conditioning of the discretisation of (1.1) and (1.2), and a key test for our algorithm is whether it behaves robustly when subjected to variations in these parameters.

It is known that any realisation of the Gaussian random field k is Hölder continuous, but not in general differentiable, and so the resulting velocity and pressure fields have only low regularity throughout the domain. Since this irregularity is global, it cannot be compensated for by local mesh refinement, and the only known way to achieve acceptable accuracy for these problems is to use low-order approximation on a mesh which is (uniformly) as fine as possible throughout the domain. In typical 2D simulations, the required number of degrees of freedom n for acceptable accuracy typically lies in the range 10^6 to 10^8. A key aim of the present paper is to provide usable methods for problems of this sort. The use of parallel computing power plays an essential rôle in achieving this aim.

Because the variable of prime interest here is the flow velocity $\vec{q}$, the discretisation schemes of most interest are those which preserve conservation of mass (1.2) in an appropriate way, with the prime candidates being mixed finite element or finite volume techniques.

Because of the lack of regularity in this problem, we discretise (1.1) and (1.2) using the lowest order mixed Raviart–Thomas elements on triangular meshes [3, 32]. Then $\vec{q}$ is approximated in an appropriate subspace of the vector-valued piecewise linear functions. The resulting discretisation enforces mass conservation on each element of the mesh and this in turn implies that the velocity field is in fact piecewise constant, making the computation of particle trajectories extremely simple.

This type of discretisation yields a symmetric indefinite system of *saddle-point* type which, in the presence of large variations in k and n, is very difficult to solve quickly enough for the multiple statistical simulations described above. We describe a fast parallel solver for these systems. The results in Section 5 show that, using our solver with a fixed number of processors, the time taken for a solve scales almost linearly (i.e., optimally) in n and is remarkably robust to σ^2 and λ. Moreover, almost 100% parallel efficiency is observed when the algorithm is tested on a machine with up to 10 processors. A modest decrease in efficiency is observed for higher numbers of processors.

Our solver is built on two essential steps. The first step decouples the velocity field in the saddle-point problem from the pressure field. This is done by writing the velocity as the curl of an appropriate discrete stream function, which automatically satisfies the discrete counterpart of the mass conservation law (1.2). The required discrete stream function turns out to be a standard finite element approximation of the solution of a related symmetric positive definite (SPD) problem which can be found independent of the pressure. Such SPD systems are easier to solve than indefinite ones and this one turns out to have the added advantage of being about five times smaller than the original saddle-point problem. A further advantage of this approach is that the approximate pressure (if it is required) can be retrieved by solving a triangular system by simple back substitutions.

The second step in the solver is the application of parallel preconditioned conjugate gradient methods to solve for the discrete stream function. This is done using additive Schwarz preconditioners (e.g., [5]) based on solves in overlapping subdomains together with a global coarse grid solve. It is known that this process can be used to build very efficient "black-box" parallel solvers which are remarkably robust for problems with highly discontinuous coefficients discretised on unstructured meshes [15, 16]. We have recently developed a general parallel package which implements this type of solution strategy and we use it here to solve the systems arising from the present application. This represents an extreme test for the package, and indeed for domain decomposition methods in general, and we are pleased to be able to report good performance of the solver under these circumstances.

Other iterative methods for related problems are reported, for example, in [2, 38], although there the emphasis is on finite volume/multigrid techniques.

The layout of this paper is as follows. In Section 2 we shall describe the background to the statistical models of heterogeneous media. In Section 3 we briefly review the discretisation of (1.1) and (1.2) and describe the reduction of the saddle-point system to an SPD system. In Section 4 we describe the additive Schwarz procedure and its resilience to discontinuous coefficients as well as our "black-box" package [19, 20]. In Section 5 we give a sequence of experiments which show the performance of the method.

2. STATISTICAL MODELLING OF HETEROGENEOUS MEDIA

The flow of fluids in the rocks composing the earth's crust is important in a number of technological and industrial fields, most notably the hydrocarbon and water resources industries. In the former, one is motivated to understand the underground flow of oil (and gas) in order to recover as much of this resource as possible. In the latter, a proper understanding of the flow of groundwater and of the transport of chemicals in it is essential not only for good resource management and quality control but also for applications in pollution modelling. One option for the long-term disposal of radioactive waste is storage in an underground repository. In order to scientifically assess the safety of this option it is necessary to model the transport of radionuclides in flowing groundwater. Thus, this topic is of general environmental importance.

From now on we restrict our attention to groundwater flow. One of the main characteristics of the rocks in the earth's crust affecting this flow is their heterogeneity (i.e., spatial variation). In particular the spatial variation of permeability k in (1.1) gives rise to variability in the flow velocity, which in turn affects the transport of dissolved chemicals or pollutants. This heterogeneity has two main aspects.

1. *Uncertainty.* Because the rock properties are varying in a complicated way and it is not possible to measure the permeability at each point in space, there is inevitably a degree of uncertainty concerning the values of the permeability. However, the permeability is in principle required at every point and simple-minded interpolation of measured values provides an insufficiently accurate approximation.
2. *Dispersion.* The heterogeneity means that the velocity field varies on a range of length scales and so particle paths that are initially close together can become progressively separated. This phenomenon—called *hydrodynamic dispersion*—is the primary mechanism for the spreading of a plume of pollutant as it is transported by the groundwater flow [11].

A widely used method of treating heterogeneity, capable of dealing with both these aspects, is stochastic modelling. The basic idea is to model the permeability field as a stochastic spatial process, assuming that a single realisation of this stochastic process is a reasonable representation of the permeability field and that any of the realisations are equally probable given the information available from measurements. This approach leads to the stochastic PDEs. (1.1) and (1.2), where $\vec{q}$ and P_R are now random variables. Given certain statistical properties of k (which we now describe), it is of interest to study statistical properties of those random variables. Thus, it is essential that we can quickly and efficiently solve (1.1) and (1.2) and, most importantly, compute the velocity $\vec{q}$ for each realisation of k. More detail on the following statistical background can be found in Refs. [1, 10].

A *random field* on an open domain $\Omega \subset \mathbb{R}^2$ (also called a *spatial process*) is a set of random variables $Z(\vec{x})$, each of which is associated with a point $\vec{x} \in \Omega$. This random field is called *Gaussian* if for each arbitrary n, each set of n random variables located at n arbitrarily chosen spatial points is Gaussian. Such fields can be completely specified by their (spatially varying) mean and covariance functions, denoted respectively by $m(\vec{x})$ and $\Sigma(\vec{x},\vec{y}) := E\{(Z(\vec{x}) - m(\vec{x}))\,(Z(\vec{y}) - m(\vec{y}))\}$, for $\vec{x},\vec{y} \in \Omega$.

In this paper we will be concerned only with *statistically homogeneous isotropic* Gaussian random fields whose mean and covariance have the particular forms

$$m(\vec{x}) := m, \qquad \Sigma(\vec{x},\vec{y}) := \sigma^2 \exp(-|\vec{x}-\vec{y}|/\lambda), \tag{2.1}$$

for positive constants m, σ, and λ. Elementary manipulations show that

$$\begin{aligned} E\{(Z(\vec{x}) - Z(\vec{y}))^2\} &= E\{Z(\vec{x})^2\} + E\{Z(\vec{y})^2\} - 2E\{Z(\vec{x})Z(\vec{y})\} \\ &= 2\sigma^2[1 - \exp(-|\vec{x}-\vec{y}|/\lambda)] = \gamma(|\vec{x}-\vec{y}|), \end{aligned} \tag{2.2}$$

where $\gamma(r) := 2\sigma^2(1 - \exp(-r/\lambda))$ is called the *variogram*; see, e.g., [10]. The parameter λ is called the *length scale*: (2.2) shows that $Z(\vec{x})$ and $Z(\vec{y})$ can only be expected to be close when $|\vec{x} - \vec{y}| < \lambda$. So, as $\lambda \to 0$, the random field Z becomes “rougher.”

We shall solve (1.1) and (1.2) in the case when $\log k$ is a realisation of such a Gaussian random field, usually called a “lognormal distribution.” There is some evidence from field data that this gives a reasonable representation of reality in certain cases [14, 24]. There are many good methods for generating realisations of Gaussian random fields, including those based on FFT [18, 33], direct simulation [12], and the turning bands method [28, 37]. Here we use the turning bands approach that represents the field as a superposition of one-dimensional fields, which are generated along lines radiating from the origin using a spectral technique.

Of particular importance to the accuracy of any discretisation is the question of regularity of this realisation. This question is thoroughly investigated in the statistical literature. In particular, it can be deduced from Adler [1] (see also [9]) that if X denotes any realisation of the Gaussian random field Z introduced above then, with probability 1, there exists a positive constant A such that $|X(\vec{x}) - X(\vec{y})| \leq A|\vec{x} - \vec{y}|^\alpha$, for all $\vec{x}, \vec{y} \in \Omega$, and $0 < \alpha < 1/2$. It then follows from the elliptic PDE theory that if k is any realisation of a lognormal distribution, then the velocity field $\vec{q}$ appearing in (1.1) will not in general exhibit better than $\mathcal{C}^\alpha$ regularity, thus motivating the use of low-order elements (see [9, Appendix] for more detail).

Once the system (1.1) and (1.2) has been solved for multiple realisations of k and the statistical properties of the velocity field have been found, the dispersion present in the system can (at least when the molecular diffusion is small) be studied by looking at the statistics of particle paths moving in the velocity field. In fact if X'_j denotes the jth coordinate of the particle displacement from its mean position then the spreading can be characterised by the second order moment of the particle paths: $X_{jl} := E\{X'_j X'_l\}$ for $j, l = 1, 2$. The variables X'_j satisfy the system of differential equations $dX'_j/dt = q_j$, where q_j is the jth component of the velocity field. This model highlights another advantage of the low-order mixed finite element method: Since the computed velocity field is constant on each element, the differential equation for X'_j is trivially integrated in an element by element fashion, thus allowing the efficient computation of the many particle paths which would be required in statistical analyses.

Before continuing we remark that there are many stochastic models for groundwater flow (see, e.g., [27] for a review), We have chosen the given model here because it is a relatively simple model which applies to fully saturated flows, but still has many of the features of some of the more complicated models. We remark also that more complicated models require more data to support them, and data are very often difficult to come by, especially in the case of deep geological waste disposal, our chief motivation for developing this algorithm.

3. DISCRETISATION AND REDUCTION TO SPD SYSTEM

For ease of notation, from now on we replace P_R by p in (1.1) and we consider (1.1) and (1.2) on a simply connected open domain $\Omega \subset \mathbb{R}^2$ with a polygonal boundary Γ, which is assumed partitioned into $\Gamma_D \cup \Gamma_N$. Each of Γ_D and Γ_N is assumed to consist of a finite union of intervals of Γ and each of the intervals in Γ_N is assumed to contain its end points. Throughout we shall assume that $\Gamma_D \neq \emptyset$, a condition which is generically satisfied in groundwater flow applications, where some inflow and outflow must occur. The extension to the case when $\Gamma_D = \emptyset$ (when p is non-unique) can easily be made by imposing an extra condition on p in the weak form below (e.g., that p should have a prescribed mean value—See [13]).

Let $\vec{\nu}(\vec{x})$ denote the outward unit normal from Ω at $\vec{x} \in \Gamma$. We describe the solution of (1.1) and (1.2) in the case where μ is a positive constant and k is a general symmetric uniformly positive definite 2×2 matrix-valued function on Ω. (In Section 5, k will be a scalar multiple of the identity.) This system is to be solved subject to mixed boundary conditions:

$$p = q \qquad \text{on } \Gamma_D, \qquad \text{and} \qquad \vec{q} \cdot \vec{\nu} = 0 \qquad \text{on } \Gamma_N. \tag{3.1}$$

Let $(\cdot, \cdot)_{L_2(\Omega)^d}$ denote the usual inner product in $L_2(\Omega)^d$, for $d = 1, 2$. Then introduce the space $H(\text{div}, \Omega) := \{\vec{v} \in L_2(\Omega)^2 \ : \ \text{div}\,\vec{v} \in L_2(\Omega)\}$, with the inner product $(\vec{u}, \vec{v})_{H(\text{div},\Omega)} := (\vec{u}, \vec{v})_{L_2(\Omega)^2} + (\text{div}\,\vec{u}, \text{div}\,\vec{v})_{L_2(\Omega)}$. Introduce also the subspace $H_{0,N}(\text{div}, \Omega) := \{\vec{v} \in H(\text{div}, \Omega) : \vec{v} \cdot \vec{\nu}|_N = 0\}$ and, for $\vec{u}, \vec{v} \in H_{0,N}(\text{div}, \Omega)$ and $w \in L_2(\Omega)$, define

$$m(\vec{u}, \vec{v}) := (\mu k^{-1}\vec{u}, \vec{v})_{L_2(\Omega)^2}, \qquad b(\vec{v}, w) := -(\text{div}\,\vec{v}, w)_{L_2(\Omega)}, \qquad \text{and}$$

$$G(\vec{v}) := -\int_{\Gamma_D} g\vec{v} \cdot \vec{\nu}\, ds.$$

Then the weak form of (1.1), (1.2) is to find $(\vec{q}, p) \in H_{0,N}(\text{div}, \Omega) \times L_2(\Omega)$ such that

$$\left.\begin{aligned} m(\vec{q}, \vec{v}) + b(\vec{v}, p) &= G(\vec{v}), && \text{for all } \vec{v} \in H_{0,N}(\text{div}, \Omega), \\ b(\vec{q}, w) &= 0, && \text{for all } w \in L_2(\Omega). \end{aligned}\right\} \tag{3.2}$$

The mixed finite element discretisation of (3.2) is obtained by choosing finite dimensional subspaces $\mathcal{V} \subset H_{0,N}(\text{div}, \Omega)$ and $\mathcal{W} \subset L_2(\Omega)$ and seeking $(\vec{Q}, P) \in \mathcal{V} \times \mathcal{W}$ to satisfy (3.2) for all $\vec{v} \in \mathcal{V}$ and $w \in \mathcal{W}$. By choosing bases $\{\vec{v}_i : i = 1, \ldots, n_\mathcal{V}\}$ and $\{w_j : j = 1, \ldots, n_\mathcal{W}\}$ for $\mathcal{V}$ and $\mathcal{W}$ and writing

$$\vec{Q} = \sum_{i=1}^{n_\mathcal{V}} q_i \vec{v}_i, \qquad P = \sum_{j=1}^{n_\mathcal{W}} p_j w_j,$$

the discrete problem reduces to the indefinite linear equation system

$$\begin{pmatrix} M & B \\ B^T & 0 \end{pmatrix} \begin{pmatrix} \mathbf{q} \\ \boldsymbol{p} \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ \mathbf{0} \end{pmatrix} \qquad \text{in } \mathbb{R}^{n_\mathcal{V}} \times \mathbb{R}^{n_\mathcal{W}}, \tag{3.3}$$

where $M_{i,i'} := m(\vec{v}_i, \vec{v}_{i'})$, $B_{i,j} := b(\vec{v}_i, w_j)$, and $f_i := G(\vec{v}_i)$.

To construct $\mathcal{V}$ and $\mathcal{W}$, let $\mathcal{T}$ denote a triangulation of Ω into conforming triangles $T \in \mathcal{T}$. We assume that the *collision points* (i.e., end-points of the intervals in Γ_N) are nodal points of this triangulation. Let $\mathcal{E}$ denote the set of all edges of the triangles in $\mathcal{T}$. It is convenient to think of these edges as open. For any $E \in \mathcal{E}$, we let $\vec{\nu}_E$ denote the unit normal to the edge E which, for convenience, is assumed to be oriented so that $\vec{\nu}_E \in \{\vec{x} \in \mathbb{R}^2 : x_1 > 0\} \cup \{(0, 1)^T\}$. Let $\mathcal{E}_I$, $\mathcal{E}_D$ and $\mathcal{E}_N$ denote the edges $E \in \mathcal{E}$ which lie, respectively, in Ω, Γ_D and Γ_N. The space $\mathcal{V}$ is defined to be the space of all functions $\vec{v} \in H_{0,N}(\text{div}, \Omega)$ such that for all $T \in \mathcal{T}$, there exist scalars α_T, β_T and γ_T such that

$$\vec{v}(\vec{x}) = \begin{pmatrix} \alpha_T \\ \beta_T \end{pmatrix} + \gamma_T \vec{x}, \qquad \text{for all } \vec{x} \in T. \tag{3.4}$$

Each $\vec{v} \in \mathcal{V}$ can be completely determined by specifying the constant value of $\vec{v} \cdot \vec{\nu}_E$ for each $E \in \mathcal{E}_I \cup \mathcal{E}_D$ and the standard basis for $\mathcal{V}$ is constructed by associating with each edge $E \in \mathcal{E}_I \cup \mathcal{E}_D$, a function $\vec{v}_E \in \mathcal{V}$ with the property that

$$\vec{v}_E \cdot \vec{\nu}_{E'} = \delta_{E,E'} \tag{3.5}$$

with δ denoting the Kronecker delta. The space $\mathcal{W}$ is chosen as the space of piecewise constant functions on Ω, with basis consisting of the characteristic functions w_T of each of the triangles $T \in \mathcal{T}$. Thus

$$n_{\mathcal{V}} = (\#\mathcal{E}_I + \#\mathcal{E}_D), \qquad n_{\mathcal{W}} = (\#\mathcal{T}), \tag{3.6}$$

where, throughout, $\#A$ denotes the number of elements of a (finite) set A.

Although the analysis below is given for system (3.3) with permeability k, in the practical implementation of (3.3) we shall replace k in m by $\tilde{k}$, the piecewise constant interpolation of k at the centroids of the triangles in the mesh $\mathcal{T}$. This is very useful practically since the generation of the Gaussian random field k is expensive and it is important to sample it at as few points as possible. It is shown in [9, Appendix] that this approximation does not degrade the estimate for the error $\|\vec{q} - \vec{Q}\|_{H(\text{div},\Omega)}$.

With the motivation given in Section 1, we now introduce our decoupling procedure for (3.3).

3.1. *General Decoupling Procedure*

The decoupling can be achieved by finding a basis $\{\mathbf{z}_1, \ldots, \mathbf{z}_{\mathring{n}}\}$ of $\ker B^T$. (Since B^T has full rank, $\mathring{n} = n_{\mathcal{V}} - n_{\mathcal{W}}$.) If we have such a basis, then the solution $\boldsymbol{q}$ of (3.3) can be written

$$\boldsymbol{q} = \sum_{j=1}^{\mathring{n}} \mathring{q}_j \mathbf{z}_j = Z^T \mathring{\boldsymbol{q}}, \tag{3.7}$$

for some $\mathring{\boldsymbol{q}} \in \mathbb{R}^{\mathring{n}}$ where Z denotes the $\mathring{n} \times n_{\mathcal{V}}$ matrix with rows $\mathbf{z}_1^T, \ldots, \mathbf{z}_{\mathring{n}}^T$. Also, since $ZB = (B^T Z^T)^T = 0$, multiplying the first (block) row of (3.3) by Z shows that $\mathring{\boldsymbol{q}}$ is a solution of the linear system

$$\mathring{A}\mathring{\boldsymbol{q}} = \mathring{\mathbf{f}} \tag{3.8}$$

where $\mathring{A} = ZMZ^T$ and $\mathring{\mathbf{f}} = Z\mathbf{f}$. Since M is SPD, so is $\mathring{A}$ and $\mathring{\boldsymbol{q}}$ is the unique solution of (3.8). Thus, if the basis $\{\mathbf{z}_1, \ldots, \mathbf{z}_{\mathring{n}}\}$ of $\ker B^T$ can be found, then the velocity $\boldsymbol{q}$ in (3.3) can be computed by solving the decoupled SPD system (3.8) rather than the indefinite coupled system (3.3). If the pressure p is also of interest, it may be found by computing a *complementary basis* $\{\mathbf{z}_{\mathring{n}+1}, \ldots, \mathbf{z}_{n_{\mathcal{V}}}\}$ with the property that span $\{\mathbf{z}_1, \ldots, \mathbf{z}_{\mathring{n}}, \mathbf{z}_{\mathring{n}+1}, \ldots, \mathbf{z}_{n_{\mathcal{V}}}\} = \mathbb{R}^{n_{\mathcal{V}}}$. If Z' then denotes the matrix with rows $\mathbf{z}_{\mathring{n}+1}^T, \ldots, \mathbf{z}_{n_{\mathcal{V}}}^T$, then $\boldsymbol{p}$ is the unique solution of the $n_{\mathcal{W}} \times n_{\mathcal{W}}$ system

$$(Z'B)\boldsymbol{p} = Z'(\mathbf{f} - M\boldsymbol{q}). \tag{3.9}$$

We show in the next two sections that, in the particular case (3.3),

(i) *It is always easy to find a basis* $\{\mathbf{z}_1, \ldots, \mathbf{z}_{\mathring{n}}\}$ *of* $\ker B^T$.

(ii) *The resulting matrix* $\mathring{A}$ *in the reduced problem (3.8) is in general a bordered matrix, whose main block consists of the stiffness matrix arising from a standard piecewise linear finite element approximation to an associated* H^1*-elliptic problem, and the number of borders is one less than the number of disconnected components in* Γ_N.

(iii) *The system (3.8) is about 5 times smaller than (3.3).*

Note that a choice of complimentary basis (in fact a subset of the fundamental basis $\{\mathbf{e}_1, \ldots, \mathbf{e}_{n_\mathcal{V}}\}$ of $\mathbb{R}^{n_\mathcal{V}}$) can be made so that the coefficient matrix $Z'B$ in (3.9) is lower triangular (see [9, 35] for a proof).

To establish conclusions (i)–(iii) we need to exploit the particular properties of (3.3). In particular note that finding a basis $\{\mathbf{z}_1, \ldots, \mathbf{z}_{\mathring{n}}\}$ of ker B^T is equivalent to finding a basis $\vec{\mathring{v}}_1, \ldots, \vec{\mathring{v}}_{\mathring{n}}$ of the finite element space

$$\mathring{\mathcal{V}} := \{\vec{V} \in \mathcal{V} : b(\vec{V}, W) = 0 \qquad \text{for all } W \in \mathcal{W}\}.$$

To see why let $Z = (Z_{i,j})$ be the matrix with rows $\mathbf{z}_1^T, \ldots, \mathbf{z}_{\mathring{n}}^T$. Then the formulae

$$\vec{\mathring{v}}_i = \sum_{j=1}^{n_\mathcal{V}} Z_{i,j} \vec{v}_j, \qquad i = 1, \ldots, \mathring{n} \tag{3.10}$$

(where $\{\vec{v}_j\}$ is the basis of $\mathcal{V}$), determine the basis $\{\vec{\mathring{v}}_i\}$. Conversely, if the basis $\{\vec{\mathring{v}}_i\}$ of $\mathring{\mathcal{V}}$ is known, then the matrix Z (and hence the basis $\mathbf{z}_1, \ldots, \mathbf{z}_{\mathring{n}}$ of ker B^T) is determined by (3.10).

3.2. *Basis of* $\mathring{\mathcal{V}}$

As a first step, recall that $\mathcal{T}$ is the set of all triangles in the mesh and that $\mathcal{E} = \mathcal{E}_I \cup \mathcal{E}_D \cup \mathcal{E}_N$ is the set of all edges of the mesh (assumed to be open intervals). Analogously we can write $\mathcal{N} = \mathcal{N}_I \cup \mathcal{N}_D \cup \mathcal{N}_N$, with $\mathcal{N}_I, \mathcal{N}_D$ and $\mathcal{N}_N$ denoting the nodes in Ω, Γ_D and Γ_N. Recall that the end points of each of the components of Γ_N belong to Γ_N and, since the collision points between Neumann and Dirichlet boundaries are mesh points, these end points lie in $\mathcal{N}_N$.

For $P \in \mathcal{N}$, let Φ_P denote the piecewise linear hat function satisfying $\Phi_P(P') = \delta_{P,P'}$. The basis for $\mathring{\mathcal{V}}$ will be constructed from the fundamental functions:

$$\vec{\Psi}_P = \vec{\text{curl}}\, \Phi_P = (\partial \Phi_P / \partial x_2, -\partial \Phi_P / \partial x_1)^T. \tag{3.11}$$

(i.e., Φ_P is the stream function for $\vec{\Psi}_P$). The functions (3.11) clearly satisfy div $\vec{\Psi}_P = 0$ on each triangle of the mesh, and a subset of them lie in $\mathring{\mathcal{V}}$ as the following proposition shows. (For a proof, see [3, Corollary III3.2].)

PROPOSITION 3.1. *For each* $P \in \mathcal{N}_I \cup \mathcal{N}_D$, $\vec{\Psi}_P \in \mathring{\mathcal{V}}$.

Note that each $\vec{\Psi}_P$ can be expressed as a local linear combination of the basis functions $\vec{v}_E$ of $\mathcal{V}$ satisfying (3.5); in fact only those $\vec{v}_E$ corresponding to edges E touching node P appear in the expansion of $\vec{\Psi}_P$ (see Fig. 1).

The functions introduced in Proposition 3.1 span a subset of $\mathring{\mathcal{V}}$, but for general boundary conditions, there are not enough of them to constitute a basis for $\mathring{\mathcal{V}}$. A small number of

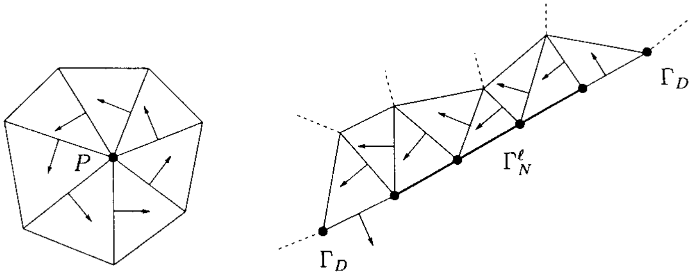

FIG. 1. Divergence-free basis functions $\vec{\Psi}_P$ (left) and $\sum_{P\in \mathbf{N}^{\ell}_{\mathcal{N}}} \vec{\Psi}_P$ (right).

additional basis functions may need to be added. Let $n_{\mathcal{C}}$ denote the number of connected components in Γ_N and write

$$\Gamma_N = \Gamma_N^1 \cup \Gamma_N^2 \cup \cdots \cup \Gamma_N^{n_{\mathcal{C}}}, \qquad \Gamma_N^{\ell} \cap \Gamma_N^{\ell'} = \emptyset \qquad \text{for all } 1 \le \ell, \ell' \le n_{\mathcal{C}}.$$

For $\ell = 1, \ldots, n_{\mathcal{C}}$, let $\mathcal{N}_N^{\ell} \subset \mathcal{N}$ denote the set of mesh nodes on Γ_N^{ℓ}. The following is proved using elementary arguments in [9].

PROPOSITION 3.2. *For each $\ell = 1, \ldots, n_{\mathcal{C}}$,*

$$\sum_{P\in\mathcal{N}_N^{\ell}} \vec{\Psi}_P \in \mathring{\mathcal{V}}. \tag{3.12}$$

The functions (3.12) are nonlocal linear combinations of the functions $\vec{v}_E$; however, the nonlocality of $\sum_{P\in\mathcal{N}_N^{\ell}} \vec{\Psi}_P$ is confined to the vicinity of Γ_N^{ℓ} (see Fig. 1). The number of such triangles is typically only $O((\#\mathcal{T})^{1/2})$.

From these elementary results we have our first theorem. It shows that when $\Gamma_N \neq \emptyset$, by combining all the functions found in Proposition 3.1 with all but one of the functions in Proposition 3.2, we have the required basis.

THEOREM 3.3. *Suppose $n_{\mathcal{C}} \neq 0$. Then a basis for $\mathring{\mathcal{V}}$ is*

$$\{\vec{\Psi}_P\colon P \in \mathcal{N}_I \cup \mathcal{N}_D\} \cup \left\{ \sum_{P\in\mathcal{N}_N^{\ell}} \vec{\Psi}_P : \ell = 1, \ldots, n_{\mathcal{C}} - 1 \right\}. \tag{3.13}$$

Proof. Suppose $n_c \neq 0$. Consider a typical Neumann boundary segment Γ_N^{ℓ}. Since $\Gamma_D \neq \emptyset$, this contains $\#\mathcal{N}_N^{\ell}$ nodes and $\#\mathcal{N}_N^{\ell} - 1$ edges. Summing over $\ell = 1, \ldots, n_{\mathcal{C}}$, we obtain $\#\mathcal{E}_N = \#\mathcal{N}_N - n_{\mathcal{C}}$. Therefore, the number of functions in (3.13) is

$$(\#\mathcal{N}_I + \#\mathcal{N}_D + n_{\mathcal{C}} - 1) = (\#\mathcal{N} - \#\mathcal{N}_N + n_{\mathcal{C}} - 1) = (\#\mathcal{N} - \#\mathcal{E}_N - 1). \tag{3.14}$$

Furthermore, since we assumed that Ω is simply connected, we can apply Euler's polyhedron theorem to obtain

$$(\#\mathcal{N} - \#\mathcal{E}_N - 1) = (\#\mathcal{E}_I + \#\mathcal{E}_D) - (\#\mathcal{E} - \#\mathcal{N} + 1) = (\#\mathcal{E}_I + \#\mathcal{E}_D) - \#\mathcal{T}, \tag{3.15}$$

Now recalling that $\mathring{n} = n_{\mathcal{V}} - n_{\mathcal{W}} = (\#\mathcal{E}_I + \#\mathcal{E}_D) - \#\mathcal{T}$, it follows from (3.14) and (3.15) that the number of functions in (3.13) is $\mathring{n} = \dim \mathring{\mathcal{V}}$, as required.

To complete the proof we merely need to show that the functions in (3.13) are linearly independent. So suppose $\{\alpha_P : P \in \mathcal{N}_I \cup \mathcal{N}_D\}$ and $\{\beta_\ell : \ell = 1, \ldots, n_{\mathcal{C}} - 1\}$ are scalars such that

$$\vec{0} = \sum_{P \in \mathcal{N}_I \cup \mathcal{N}_D} \alpha_P \vec{\Psi}_P + \sum_{\ell=1}^{n_{\mathcal{C}}-1} \beta_\ell \sum_{P \in \mathcal{N}_N^\ell} \vec{\Psi}_P. \tag{3.16}$$

Using the linearity of $\vec{\text{curl}}$, this may be rewritten

$$\vec{0} = \sum_{P \in \mathcal{N}} \alpha_P \vec{\Psi}_P = \sum_{P \in \mathcal{N}} \alpha_P \,\vec{\text{curl}}\, \Phi_P = \vec{\text{curl}} \left(\sum_{P \in \mathcal{N}} \alpha_P \Phi_P \right), \tag{3.17}$$

where $\alpha_P := \beta_\ell$ when $P \in \mathcal{N}_N^\ell$ for $\ell = 1, \ldots, n_{\mathcal{C}} - 1$ and $\alpha_P := 0$ for $P \in \mathcal{N}_N^{n_{\mathcal{C}}}$. But this implies that $\sum_{P \in \mathcal{N}} \alpha_P \Phi_P$ is constant on Ω and hence (since $\mathcal{N}_N^{n_{\mathcal{C}}} \neq \emptyset$), $\alpha_P = 0$ for all $P \in \mathcal{N}$. ■

Remark 3.4. In the pure Dirichlet case (i.e., $\Gamma_N = \emptyset$), a suitable basis is $\{\vec{\Psi}_P : P \in \mathcal{N}, P \neq P_0\}$ for any choice of $P_0 \in \mathcal{N}$. The proof follows exactly the same steps with a few changes in notation.

The construction in (3.11), in which divergence-free Raviart–Thomas elements are written as the curls of suitable stream functions, appears at several points in the literature, e.g., [7, 21] and later in the development of preconditioning strategies for the saddle point system (3.3), [13, 23, 29]. This construction is also found in the subsequent unpublished manuscript [22], although the principal subject of that paper is the solution of the Stokes problem.

In this paper we give for the first time an explicit basis for $\mathring{\mathcal{V}}$ in the case of general mixed boundary conditions and an algorithmic description of the use of this basis in a solver for (3.3). Recently, the same idea has been investigated in 3D [4], although in that paper the method is developed only for the case of uniform rectangular grids.

In the related but different case of the Stokes problem there is a large literature concerning divergence-free elements; see, for example, [17, 21, 30, 36]. However in the Stokes case, after decoupling, the discrete stream function appears underneath a (relatively ill-conditioned) fourth-order operator, whereas for groundwater flow only a second-order operator appears. For this reason, it is perhaps surprising that the divergence-free reduction has received more attention in the literature for the Stokes problem than for problems like groundwater flow.

The procedure given here has a non-trivial extension to 3D. Here the “stream functions” (or more precisely the vector potentials) are no longer the hat functions but rather the Nédélec edge elements [3, p. 117] and graph theoretic methods are needed to select an appropriate basis [34, 35].

3.3. *Implementation*

To implement the decoupled system (3.8) for determining $\mathring{\boldsymbol{q}}$ (and hence $\boldsymbol{q}$) we must work with the matrix $\mathring{A}$ and right hand side $\mathring{\mathbf{f}}$. A direct elementwise assembly of these from Raviart–Thomas basis functions can be easily given (see, e.g., [9]). Alternatively, $\mathring{A}$ can also be determined from a standard piecewise linear approximation of a related bilinear form, without the assembly of any Raviart–Thomas stiffness matrix entries.

First observe that $\mathring{A}$ is formally defined in terms of multiplications with the matrix Z which, through (3.10), represents the basis $\{\vec{\mathring{v}}_i\}$ of $\mathring{\mathcal{V}}$ in terms of the basis $\{\vec{v}_j\}$ of $\mathcal{V}$. In the specific system (3.3) the $\{\vec{v}_j\}$ are the Raviart–Thomas velocity basis functions $\{\vec{v}_E : E \in \mathcal{E}_I \cup \mathcal{E}_D\}$, whereas the $\{\vec{\mathring{v}}_i\}$ are the basis functions specified in Theorem 3.3. Thus, we can identify the rows of Z with the indices $P \in \mathcal{N}_I \cup \mathcal{N}_D$ and $\ell = 1, \ldots, n_{\mathcal{C}} - 1$, whereas the columns of Z correspond to $E \in \mathcal{E}_I \cup \mathcal{E}_D$. Then we can rewrite (3.10) as

$$\vec{\Psi}_P = \sum_{E \in \mathcal{E}_I \cup \mathcal{E}_D} Z_{P,E} \vec{v}_E, \qquad P \in \mathcal{N}_I \cup \mathcal{N}_D \tag{3.18}$$

and

$$\sum_{P \in \mathcal{N}_N^\ell} \vec{\Psi}_P = \sum_{E \in \mathcal{E}_I \cup \mathcal{E}_D} Z_{\ell,E} \vec{v}_E, \qquad \ell \in 1, \ldots, n_{\mathcal{C}} - 1. \tag{3.19}$$

With the same convention we can write the elements of the matrix M appearing in (3.3) as

$$M_{E,E'} = m(\vec{v}_E, \vec{v}_{E'}), \qquad E, E' \in \mathcal{E}_I \cup \mathcal{E}_D. \tag{3.20}$$

Now introduce the bilinear form

$$a(\zeta, \Phi) := (\mu K^{-1} \vec{\nabla} \zeta, \vec{\nabla} \Phi)_{L_2(\Omega)^2}, \qquad \zeta, \Phi \in H^1(\Omega), \tag{3.21}$$

where $K = S^T k S$, and $S = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$. Then, for $P, P' \in \mathcal{N}$, set

$$\mathcal{A}_{P,P'} = a(\Phi_P, \Phi_{P'}),$$

where $\{\Phi_P\}$ are the piecewise linear hat functions introduced at the beginning of Section 3.2. Thus, (after specifying an ordering of the nodes in $\mathcal{N}$), $\mathcal{A}$ is a standard finite element stiffness matrix corresponding to the bilinear form $a(\cdot, \cdot)$ with a natural boundary condition on all of Γ. Let A denote the minor of this matrix obtained by restricting to $P, P' \in \mathcal{N}_I \cup \mathcal{N}_D$. (This corresponds to imposing an essential boundary condition on Γ_N.) Moreover, define the matrices

$$C_{P,\ell'} := \sum_{P' \in \mathcal{N}_N^{\ell'}} \mathcal{A}_{P,P'}, \qquad \text{and} \qquad D_{\ell,\ell'} := \sum_{P \in \mathcal{N}_N^\ell} \sum_{P' \in \mathcal{N}_N^{\ell'}} \mathcal{A}_{P,P'},$$

for $P \in \mathcal{N}_I \cup \mathcal{N}_D$ and $\ell, \ell' = 1, \ldots, n_{\mathcal{C}} - 1$. The following result shows that $\mathring{A}$ can be obtained by applying a small number of elementary operations to $\mathcal{A}$.

THEOREM 3.5.

$$\mathring{A} = \begin{bmatrix} A & C \\ C^T & D \end{bmatrix}.$$

Remark 3.6. The rôle of bilinear form (3.21) in the theory of the Raviart–Thomas approximation of second-order elliptic problems was pointed out by [13, 29]. However, those references were not concerned with the construction of a basis for $\mathring{\mathcal{V}}$ and so (3.21) was not used there in the way it is used here.

Proof of Theorem 3.5. For $P, P' \in \mathcal{N}_I \cup \mathcal{N}_D$ we have

$$\begin{aligned}\mathring{A}_{P,P'} &= \sum_{E,E' \in \mathcal{E}_I \cup \mathcal{E}_D} Z_{P,E} M_{E,E'} Z_{P',E'} = m(\vec{\Psi}_P, \vec{\Psi}_{P'}) = (\mu k^{-1} \vec{\mathrm{curl}}\, \Phi_P, \vec{\mathrm{curl}}\, \Phi_{P'})_{L^2(\Omega)^2} \\ &= (\mu k^{-1} S \vec{\nabla} \Phi_P, S \vec{\nabla} \Phi_{P'})_{L^2(\Omega)^2} = a(\Phi_P, \Phi_{P'}) = \mathcal{A}_{P,P'} = \mathrm{A}_{P,P'}.\end{aligned}$$

Similarly, for $P \in \mathcal{N}_I \cup \mathcal{N}_D$ and $\ell = 1, \ldots, n_{\mathcal{C}} - 1$ we have $\mathring{A}_{P,\ell'} = C_{P,\ell'}$. Completely analogous arguments show that $\mathring{A}_{\ell,\ell'} = D_{\ell,\ell'}$, $\ell, \ell' = 1, \ldots, n_{\mathcal{C}} - 1$, and since $\mathring{A}$ is symmetric the theorem follows. ■

Remark 3.7. Observe that the decoupled system (3.8) is about five times smaller than the original indefinite system (3.3). More precisely, the dimension of (3.8) is smaller than that of (3.3) by a factor

$$F := \frac{\#\mathcal{E}_I + \#\mathcal{E}_D + \#\mathcal{T}}{\#\mathcal{E}_I + \#\mathcal{E}_D - \#\mathcal{T}}.$$

Note that $3(\#\mathcal{T}) = 2(\#\mathcal{E}_I) + \#\mathcal{E}_D + \#\mathcal{E}_N$ and that, under reasonable grid regularity assumptions, $\#\mathcal{E}_I$ is the dominant part of $\#\mathcal{E}$ as $\#\mathcal{T} \to \infty$. Thus, $F \to 5$ as $\#\mathcal{T} \to \infty$.

Remark 3.8. The coefficient matrix $\mathring{A}$ is a bordered matrix with major block consisting of the standard piecewise linear finite element stiffness matrix A, and with the width of the border $n_{\mathcal{C}} - 1$, where $n_{\mathcal{C}}$ is the number of disconnected components in the Neumann boundary Γ_N. If $n_{\mathcal{C}} = 1$, then $\mathring{A} = A$. In general, systems of this form can be solved by standard block elimination algorithms using $n_{\mathcal{C}}$ solves with A.

The average number of nonzero entries of A per row approaches 7 as the number of unknowns n in A tends to infinity and the bandwidth (which depends on the ordering of the degrees of freedom) is in general $O(n^{1/2})$. In comparison, for the matrix in the coupled system (3.3), the average number of nonzero entries per row approaches 5.4 and the bandwidth is still $O(n^{1/2})$ (see [35] for details).

On the other hand, under reasonable grid regularity assumptions, the condition number of A is $O(n)$ and the coupled matrix does have a better condition number ($O(n^{1/2})$ in fact). However, since A is SPD, we can apply preconditioned conjugate gradients, and a range of optimal preconditioners are available which ensure in theory that the number of iterations does not grow as n increases. (In Section 4 we present a parallel implementation where the number of iterations grows with $O(n^{1/6})$.) To solve system (3.3) on the other hand, we would have to fall back on other Krylov subspace methods such as MINRES [31]. Here (in the unpreconditioned case) the number of iterations can only be expected to grow no faster than the condition number of the matrix (i.e., $O(n^{1/2})$ and optimal preconditioners are not readily available (see [35] for details). So from several points of view, the reduction to SPD makes practical sense.

4. PARALLEL ITERATIVE METHOD

In this section we briefly describe our parallel solver for the velocity systems (3.8) arising in Section 3. Our method is based on the conjugate gradient algorithm with additive Schwarz preconditioner and uses the implementation provided by the `DOUG` package [20] for general unstructured systems. Although the applications in the present paper are on uniform meshes, this uniform structure is not exploited in the solver and so the computing times reported should be comparable to those required for more general unstructured applications. Also, although our application here is two-dimensional, we mention that the DOUG code handles quite general three-dimensional problems. Full details are given in [19, 20].

The first step in our parallelisation involves the partition of the domain Ω (in this case using the mesh partitioning software METIS [25]) into non-overlapping connected subdomains Ω_i, $i = 1, \ldots, s$, each consisting of a union of elements of the mesh. The METIS software strives to ensure that the Ω_i are of comparable size ("load-balancing") and the interfaces between them contain as few edges as possible (to minimise communication). These subdomains are then used for parallelisation of the vector-vector and matrix–vector operations required in the conjugate gradient algorithm. Good parallel efficiency is achieved for matrix–vector products by ensuring that the necessary communication of boundary data between neighbouring subdomains is overlapped with computations in the (independent) subdomain interiors.

For preconditioning we use the unstructured version of the classical two-level additive Schwarz method (e.g., [6]), which has the general form

$$\mathcal{P}^{-1} := R_H^T A_H^{-1} R_H + \sum_{i=1}^{s} R_i^T A_i^{-1} R_i. \tag{4.1}$$

In (4.1) the matrices A_i^{-1} represent local solves of the underlying PDE on overlapping extensions $\tilde{\Omega}_i$ of the Ω_i with homogeneous Dirichlet condition imposed on the parts of $\partial\tilde{\Omega}_i$ which do not intersect with $\partial\Omega$. The restriction operator R_i is taken to be the simple injection operator.

In our particular implementation of (4.1), $\tilde{\Omega}_i$ is constructed by adding to each Ω_i all the elements of the mesh which touch its boundary $\partial\Omega_i$. The resulting extended subdomains $\tilde{\Omega}_i$ then have overlap δ, say, with δ bounded above (respectively below) by the maximum (respectively minimum) diameter of all the elements of the mesh. This choice of overlap represents a compromise between the competing demands of condition number optimality and efficiency of the parallelisation (the former requiring, at least in theory, a reasonable overlap and the latter requiring that the overlap should be as small as possible). This choice also means that A_i is simply the minor of A obtained by removing all the rows and columns corresponding to nodes not on $\Omega_i \cup \partial\Omega_i$.

In the present version of the DOUG package the subdomain solves A_i^{-1} are done using a direct frontal solver and so, to achieve good efficiency, the underlying subdomains should not become too large. In DOUG the default size is 1000 degrees of freedom (and this is what we use in the present paper). Since the package is designed to run on any number of processors, we allow the possibility that each processor will handle several subdomains.

The preconditioner (4.1) also contains a coarse grid solve, A_H^{-1}, which handles the global interaction of the subdomains. This distinguishes (4.1) from block-Jacobi-like methods and is essential for the construction of optimal preconditioners. There is no need for the coarse mesh to be related directly to the fine mesh, but in principle it should be capable of representing the solution of the underlying PDE with appropriate accuracy. What this means in practice is that, if one has constructed a fine mesh which provides a sufficiently good resolution of the underlying problem, then one requires also a coarse mesh with the same qualitative properties at the coarser level. Such a coarsening may sometimes be available (e.g., from an earlier stage of a refinement process) but, since this is not always the case, the DOUG package produces a coarsening automatically. For this, an adaptive piecewise uniform strategy is used, the efficiency of which is discussed in detail in [19]. In (4.1) the operator R_H^T denotes piecewise linear interpolation from coarse to fine mesh, R_H denotes its transpose, and A_H is the Galerkin product $A_H = R_H^T A R_H$.

In the present version of DOUG the coarse mesh problem is assembled and solved directly using the frontal method on a master processor. In order to maintain efficient parallelisation, the time for this should not exceed the time which is being taken by the processors which are working on the subdomain solves. If n denotes the total number of degrees of freedom in the problem and n_p is the number of processors then (assuming load balancing) each processor has to solve $n/(1000 * n_p)$ problems, each with 1000 unknowns. The cost of a frontal solve for a finite element problem with N degrees of freedom (in 2D) is about $8N^{3/2}$ (see the references in [19]). Thus, for parallel efficiency the dimension of the coarse grid problem n_H is chosen in DOUG to satisfy

$$n_H^{3/2} = \text{cost of solving subproblems on processors} = \left(\frac{n}{(1000 * n_p)} \right) * 1000^{3/2}. \quad (4.2)$$

Note that for a fixed n_p this implies that $n_H = O(n^{2/3})$.

The asymptotic performance of the preconditioner (4.1) is analysed in [6], where it is shown that for general symmetric positive definite problems

$$\kappa(\mathcal{P}^{-1}A) = O((H/\delta)^2), \qquad \text{as } H, h \to 0 \quad (4.3)$$

where κ denotes the 2-norm condition number, h, H denote the fine and coarse mesh diameters, and δ denotes the overlap in the subdomains $\tilde{\Omega}_i$.

Then with the DOUG code as described above applied to a problem on a uniform fine mesh $n \approx h^{-2}$, the overlap is $\delta = h \approx n^{-1/2}$ and the coarse mesh (which will be almost uniform) has $n_H = O(n^{2/3})$ degrees of freedom and diameter $H \approx n_H^{-1/2} = O(n^{-1/3})$. The estimate (4.3) then reduces to $\kappa(\mathcal{P}^{-1}A) = O((n^{1/2}n^{-1/3})^2) = O(n^{1/3})$ and the number of iterations of the conjugate gradient method will grow no faster than $O(n^{1/6})$. We examine numerically in the following section the sharpness of this estimate.

We shall also discuss the performance of this method in the presence of very rough coefficients. A lot is known about this case provided the jumps occur on a coarser scale than the fine mesh being used to compute the solution. In the case of certain two-level domain decomposition methods on structured meshes, for example, the effect of the jumps can be removed completely provided the coarse mesh resolves the jumping regions. In the unstructured case this is no longer true, indeed the preconditioned problem may be just as ill-conditioned as the original matrix as the jumps worsen. An example showing this was given in [15, 16], where it is also shown that the condition number is not a very good guide in this case to the behaviour of the preconditioned conjugate gradient (PCG) method, since the preconditioned problem has only a small cluster of eigenvalues near the origin with the others lying in a bounded region away from the origin as the jumps get worse. The general proof of this phenomenon led in [15, 16] to the proof that the corresponding PCG method in fact is very resilient to jumping coefficients even in the unstructured case. Roughly speaking [15] shows that in the case of a piecewise constant coefficient k with respect to a fixed number of regions of the domain, the number of PCG iterations will grow only logarithmically in the quantity $\max|k|/\min|k|$, whereas the condition number itself generally grows linearly in $\max|k|/\min|k|$.

The results in [15, 16] apply when the jumping coefficient varies on a coarser scale than the fine mesh and so they do not strictly apply to the case of the heterogeneous media considered in the next section, where the coefficient varies on the fine mesh scale. However, interestingly, the numerical results given below indicate that in some sense the results of [15, 16] hold true even in this extreme case, although at the time of writing we know of no proof of this.

5. NUMERICAL RESULTS

In this section we report on a number of experiments on the parallel solution of (1.1), (1.2) in the special case when the domain Ω is $[0, 1] \times [0, 1]$, the viscosity μ is taken to be 1, and the permeability k is chosen so that $\log k$ is a realisation of a Gaussian random field on Ω (as described in Section 2) with zero mean, variance σ^2, and length scale λ.

We discretise this problem using the mixed finite element discretisation with lowest order Raviart–Thomas elements as described in Section 3 on a uniform mesh $\mathcal{T}$ on Ω obtained by first subdividing the mesh into N^2 equal squares $[(i-1)/N, i/N] \times [(j-1)/N, j/N]$ and then further subdividing each square into two triangles. This is done by colouring the squares in a red/black checkerboard pattern and then using a diagonal drawn from bottom left to top right for red squares and from top right to bottom left for black squares. We replace k on each element with its constant interpolant at the centroid of the element, an approach which allows an efficient implementation and maintains the accuracy of the discretisation (see [9, Appendix]). This approach only makes statistical sense when the length scale λ is of the order of the mesh diameter, equivalently

$$\lambda = C_\ell / N, \tag{5.1}$$

for some constant $C_\ell \geq 1$, as $N \to \infty$. However, since N must already be large enough to ensure acceptable accuracy (i.e., $N \sim 10^3$, or 10^4), fairly fine length scales are treatable by this choice, and it is widely used in hydrogeological modelling. Smaller length scales could be treated by an appropriate upscaling of k in each element, but this would be expensive and can be expected to have only minor effect on the dispersion in the velocity field, which is the main phenomenon of interest here. So, throughout this section, k is replaced by its piecewise constant interpolant $\tilde{k}$, which is computed using the turning bands algorithm [28, 37].

We assume that there is zero flow across the bottom and top of Ω and that the residual pressure p is required to have value 1 at the left-hand boundary and 0 at the right-hand boundary (corresponding to a prescribed pressure gradient across the domain). Thus, in (3.1) we have $\Gamma_D = \{0, 1\} \times (0, 1)$ and

$$g(\vec{x}) \equiv 1 \qquad \text{for } \vec{x} \in \{0\} \times [0, 1], \qquad g(\vec{x}) \equiv 0 \qquad \text{for } \vec{x} \in \{1\} \times (0, 1). \tag{5.2}$$

We shall give results here only for the computation of the velocity $\boldsymbol{q}$ in (3.3) by solving the decoupled system (3.8) for $\mathring{\boldsymbol{q}}$. In the case of the particular boundary conditions (5.2), the computation reduces to the solution of the linear system

$$\begin{bmatrix} A & \mathbf{c} \\ \mathbf{c}^T & d \end{bmatrix} \begin{bmatrix} \mathring{\boldsymbol{\eta}} \\ \mathring{\rho} \end{bmatrix} = \begin{bmatrix} \mathring{\boldsymbol{\varphi}} \\ \mathring{\xi} \end{bmatrix}, \tag{5.3}$$

where A is a square sparse matrix, and (since Γ_N here contains only two components) $\mathbf{c}$ is a single column vector and d is a scalar. All of these are obtained by elementary row and column operations on a standard piecewise linear finite element matrix (see Section 3.3). Block elimination in this case requires solutions of two systems of the form

$$A\boldsymbol{u} = \mathbf{b}. \tag{5.4}$$

In the special case here, where the Dirichlet data g is constant on each component of Γ_D, it turns out that $\mathring{\varphi} = \mathbf{0}$ and we only need to solve (5.4) once. The timings in Section 5.2 are for this task.

The sparse, SPD, and highly ill-conditioned problem (5.4) is solved by the parallel iterative method described in Section 4. There are three parameters which determine the difficulty of (5.4): the mesh parameter N, the variance σ^2, and the length scale λ. We are interested in the efficiency of this parallel method as well as its robustness with respect to these parameters. For our tests we allow σ^2 to vary independently, and λ to vary as in (5.1) for some constant C_ℓ to be specified below. From a numerical point of view these are particularly difficult problems, since the realisation of k varies from element to element and may take wildely differing values across the domain. As C_ℓ decreases, the probability of large jumps in k between neighbouring elements increases (see Section 2). On the other hand, to illustrate the effect of increasing σ^2, in Fig. 2 we give a gray scale plot of the values of $\log k$ for a single realisation in the case $N = 256$ and $\lambda = 10/N$ for two different values of σ^2. Observe that the pattern is independent of σ^2, but that the scale changes as σ^2 increases. In fact the numerical range of $\log k$ grows linearly in $\sqrt{\sigma^2}$, and so the condition number κ of the matrix A will grow like $\exp(2\sqrt{\sigma^2})$ as σ^2 increases. To emphasise the effect that this will have on the conditioning of (5.4) observe, for example, that $\max|k|/\min|k| \sim 10^9$ when $\sigma^2 = 8$.

5.1. *Selection of Stopping Criterion*

Since we have in mind here the solution of a range of problems of varying difficulty by an iterative method, it is important to design a stopping criterion which ensures reasonably uniform accuracy across all problems. This ensures that subsequent comparison of solution times will be meaningful. In this section we describe a heuristically based approach to designing such a stopping criterion.

The preconditioned conjugate gradient (PCG) method for (5.4) with SPD preconditioner $\mathcal{P}$ produces a sequence of iterates $\boldsymbol{u}^i$ and residuals $\mathbf{r}^i$ which satisfy $\mathbf{r}^i = \mathbf{b} - A\boldsymbol{u}^i = A\mathbf{e}^i$ where $\mathbf{e}^i = \boldsymbol{u} - \boldsymbol{u}^i$ is the error at the ith iterate. This algorithm also computes the *preconditioned residual* $\mathbf{z}^i = \mathcal{P}^{-1}\mathbf{r}^i = (\mathcal{P}^{-1}A)\mathbf{e}^i$. Typical stopping criteria for the PCG method involve requiring that $\mathbf{z}^i$ be small in some norm. More precisely we have the standard estimate for the relative error reduction

$$\|\mathbf{e}^i\|_2/\|\mathbf{e}^0\|_2 \leq \kappa\|\mathbf{z}^i\|_2/\|\mathbf{z}^0\|_2, \tag{5.5}$$

where κ denotes the condition number of $\mathcal{P}^{-1}A$ with respect to $\|\cdot\|_2$. From this it follows that the stopping criterion

$$\|\mathbf{z}^i\|_2/\|\mathbf{z}^0\|_2 \leq \epsilon/\kappa \tag{5.6}$$

is sufficient to ensure the required relative error reduction $\|\mathbf{e}^i\|_2/\|\mathbf{e}^0\|_2 \leq \epsilon$.

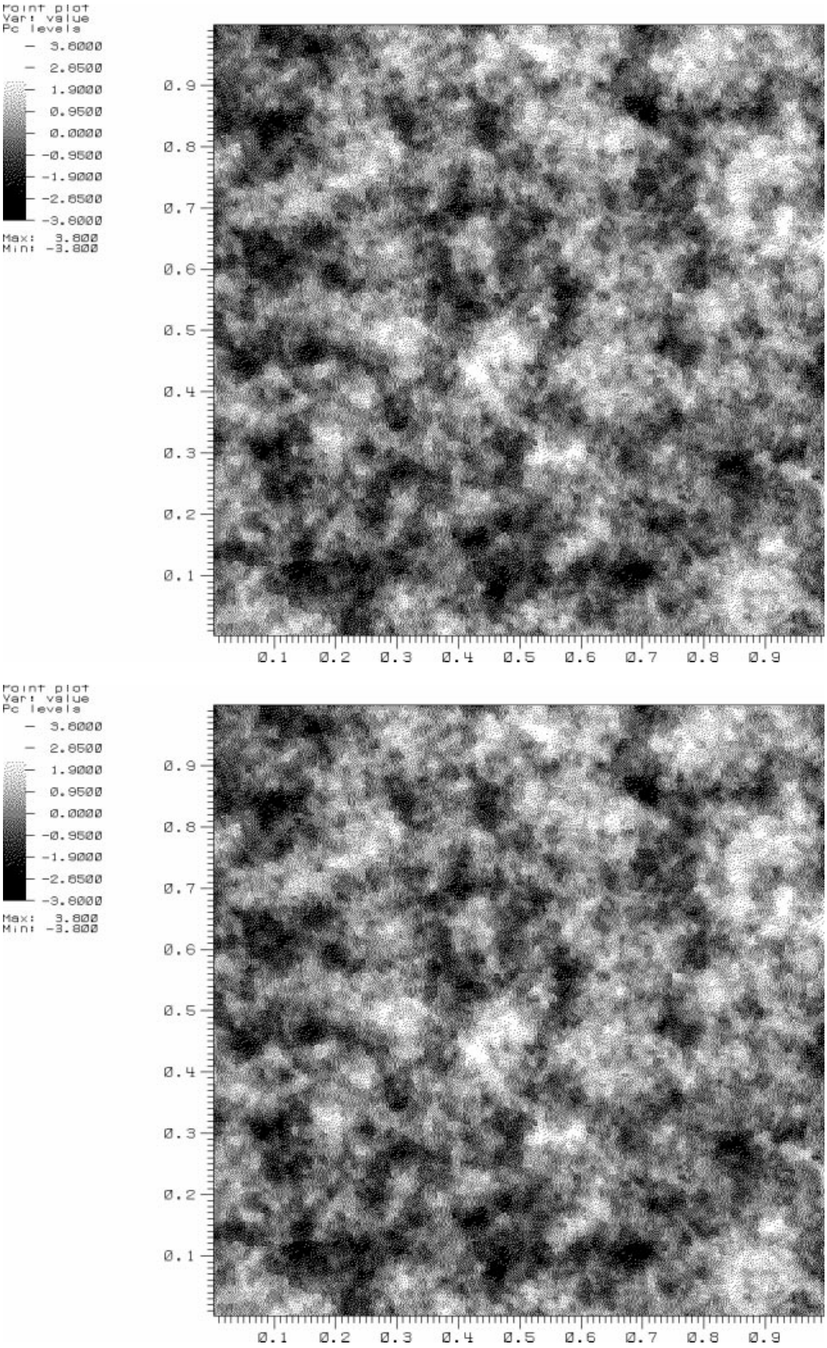

FIG. 2. Gray scale plot of log(k) for $\sigma^2 = 1$ (top) and $\sigma^2 = 8$ (bottom).

Of course κ is unknown and some authors (e.g., [26]) suggest estimating κ dynamically during the CG iteration. However, even if such a procedure is adopted, the resulting stopping criterion (5.6) is often over-pessimistic due to the fact that the smallest constant κ such that (5.5) holds for all i is often very much smaller than the true condition number of $\mathcal{P}^{-1}A$.

Here we are interested in a class of problems which depend on parameters σ^2, N, and λ. For a restricted range of problems (which are small enough so that the exact solution can be computed by a direct solver), we compute the *effective condition number*,

$$\tilde{\kappa} := \tilde{\kappa}(\sigma^2, N, \lambda) := \frac{\|\mathbf{e}^i\|_2}{\|\mathbf{e}^0\|_2} \frac{\|\mathbf{z}^0\|_2}{\|\mathbf{z}^i\|_2}, \tag{5.7}$$

for some specified i as the parameters σ^2, N, and λ change. Our practical stopping criterion is then to choose the first i such that

$$\frac{\|\mathbf{z}^i\|_2}{\|\mathbf{z}^0\|_2} \leq \epsilon/\tilde{\kappa}. \tag{5.8}$$

The result of this exercise is that $\tilde{\kappa}$ is found to vary only very mildly with these parameters (see (5.9) below).

To obtain $\tilde{\kappa}(\sigma^2, N, \lambda)$ experimentally, we solved the test problems using the conjugate gradient method with additive Schwarz preconditioner as described in Section 4, with initial guess $\boldsymbol{u}^0 = \mathbf{0}$, and we iterated until the relative error $\|\mathbf{e}^i\|_2/\|\mathbf{e}^0\|_2$ was less than $\epsilon = 10^{-4}$ (with the exact solution $\boldsymbol{u}$ found using a direct solver). From this solution we computed $\tilde{\kappa}$ above.

First, we studied the variation with respect to σ^2, and here we fixed $N = 32$ and $\lambda = 10/N$. In Fig. 3 (left) we plot computed values of $\tilde{\kappa}$ against σ^2 (solid line). The best least squares straight line fit to these points (dotted line) yields an empirical approximation for the variation of $\tilde{\kappa}$ with σ^2 as: $0.26 + 0.13\sigma^2$. To test the validity of this, we recomputed the above experiments using the stopping criterion (5.8) with $\tilde{\kappa} = 0.26 + 0.13\sigma^2$ and $\epsilon = 10^{-4}$. The relative error $\|\mathbf{e}^i\|_2/\|\mathbf{e}^0\|_2$ remained in the interval $[2 \times 10^{-5}, 1.4 \times 10^{-4}]$ as σ^2 ranged between 1 and 8, indicating that this is a reasonable approximation of how $\tilde{\kappa}$ varies with σ^2.

To study variation with respect to λ, we set $N = 32$ and $\sigma^2 = 4$ and computed $\tilde{\kappa}$ for $\lambda = 10/16, 10/32, \ldots, 10/1024$. A $\log_2 - \log_2$ plot of these results is given in Fig. 3 (right) (solid line). The dotted line shows the best computed straight line fit and suggests that $\tilde{\kappa}$ decreases with $O(\lambda^{-0.4})$ as λ increases. From this observation we propose the empirical model $\tilde{\kappa} = (0.26 + 0.13\sigma^2)(0.46\lambda^{-0.4})$. To demonstrate the validity of this we recomputed these experiments using this value of $\tilde{\kappa}$ in stopping criterion (5.8) where $\sigma^2 = 4$, $N = 32$, and $\epsilon = 10^{-4}$. We found that the resulting relative error lay in the range $[6 \times 10^{-5}, 1.4 \times 10^{-4}]$ indicating a stopping criterion which is robust to variations in λ.

Finally, to model variations with respect to N we computed $\tilde{\kappa}$ in (5.7) for $N = 16$, 32, 64, 128 in the case $\sigma^2 = 4$ and $\lambda = 10/16$. These experiments suggested that there is no

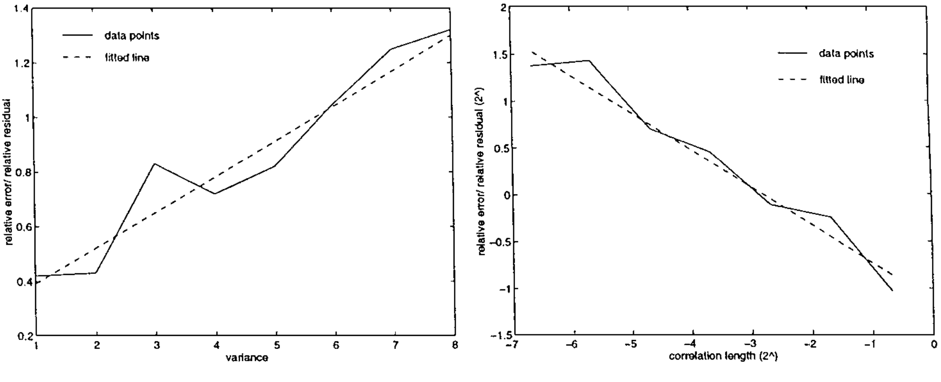

FIG. 3. Variation of $\tilde{\kappa}$ with σ^2 (left) and with λ for $\sigma^2 = 4$ (right) ($N = 32$).

TABLE I
Study of the Iteration Count ($\lambda = 10/N$)

			With coarse grid		Without coarse grid	
N	n	σ^2	No. It.	$\lVert\mathbf{z}^i\rVert_2/\lVert\mathbf{z}^0\rVert_2$	No. It.	$\lVert\mathbf{z}^i\rVert_2/\lVert\mathbf{z}^0\rVert_2$
128	16383	1	21	7.72×10^{-10}	123	1.80×10^{-9}
		2	22	1.20×10^{-9}	137	1.18×10^{-9}
		4	26	5.67×10^{-10}	174	8.14×10^{-10}
		6	29	6.98×10^{-10}	198	6.82×10^{-10}
		8	33	3.97×10^{-10}	223	4.22×10^{-10}
256	65535	1	23	9.52×10^{-10}	270	1.33×10^{-9}
		2	26	7.82×10^{-10}	320	1.09×10^{-9}
		4	31	5.47×10^{-10}	454	6.67×10^{-10}
		6	34	5.58×10^{-10}	602	5.57×10^{-10}
		8	41	3.81×10^{-10}	740	3.76×10^{-10}
512	262143	1	27	4.45×10^{-10}	593	1.10×10^{-9}
		2	29	7.87×10^{-10}	742	8.34×10^{-10}
		4	38	4.94×10^{-10}	1155	5.48×10^{-10}
		6	46	4.33×10^{-10}	1677	4.13×10^{-10}
		8	57	2.64×10^{-10}	>2000	
1024	1048575	1	33	3.44×10^{-10}	1059	8.74×10^{-9}
		2	35	6.40×10^{-10}	1598	6.46×10^{-9}
		4	45	4.15×10^{-10}	>2000	
		6	57	3.12×10^{-10}	>2000	
		8	70	1.74×10^{-10}	>2000	

noticeable increase in the value of $\tilde{\kappa}$ as N increases. Thus, we postulate that

$$\tilde{\kappa}(\sigma^2, N, \lambda) \approx (0.26 + 0.13\sigma^2)(0.46\lambda^{-0.4}) \tag{5.9}$$

as σ^2, λ, and N vary. In the experiments in the next section we use this formula for $\tilde{\kappa}$ in the stopping criterion (5.8).

5.2. *Performance of Iterative Method*

Our first set of results in Table I, illustrates the performance of the PCG method for (5.4) with an additive Schwarz preconditioner (4.1), with and without coarse grid solve for various values of N and σ^2, where the length scale λ varies as in (5.1) with $C_\ell = 10$, and n denotes the number of unknowns in system (5.4). The stopping criterion was (5.8) with $\epsilon = 10^{-9}$ and $\tilde{\kappa}$ given by (5.9). The value of $\lVert\mathbf{z}^i\rVert_2/\lVert\mathbf{z}^0\rVert_2$ given is the value of this quantity when the iteration stops (where $\mathbf{z}^i$ denotes the preconditioned residual, as described above).

TABLE II
Number of Iterations as N (and Therefore n) Increases ($\lambda = 10/N$, $\sigma^2 = 2$)

N	n	With coarse grid	Without coarse grid
256	65535	26	320
512	262143	29	742
1024	1048575	35	1598

TABLE III
Number of Iterations as σ^2 Increases (N = 256, λ = 10/N)

$\sqrt{\sigma^2}$	With coarse grid	Without coarse grid
1	23	270
1.4	26	320
2	31	454
2.4	34	602
2.8	41	740

The first thing to note is the observed success of the strategy for computing the coarse grid (see Section 4). Since this is constructed just from the *geometry* of the fine grid, ignoring the fact that the coefficient k is varying from element to element, on may be concerned that it may not model the underlying fine scales of the problem well enough to be effective. While there is clearly some dependence on the fine scale of the coefficient (the iteration numbers increase slightly as σ^2 increases) this dependence is mild (see below) and the addition of the coarse grid solve is clearly having a big effect on the robustness of the preconditioner. In the case $N = 512$ and $\sigma^2 = 8$, the addition of the coarse grid solve improved the computation time by a factor of about 30.

In the next two tables we investigate the robustness of the iterative method with respect to the various parameters in the problem in more detail. First, in Table II we investigate the behaviour of the method as N grows. We know from the discussion in Section 4 that, for a *fixed smooth coefficient function*, as N (and therefore n) increases, we expect the number of PCG iterations to grow at worst with $O(n^{1/2}) = O(N)$, when the coarse solve is not included in the preconditioner, and with $O(n^{1/6}) = O(N^{1/3})$, when the coarse solve is included. The results in Table II indicate a growth no worse than this, even though in this case the coefficient is extremely rough.

In Table III we illustrate how the iteration numbers are affected by growth in σ^2 for $N = 256$ and $\lambda = 10/N$. The rate of growth of the number of PCG iterations is approximately linear in $\sqrt{\sigma^2}$. This behaviour is observed both with and without a coarse grid solve, although with a considerably larger asymptotic constant in the latter case. This should be compared with the fact that the *condition number* of the stiffness matrix A in (5.4) grows like $\exp(2\sqrt{\sigma^2})$. This observed behaviour (where the growth of the number of iterations is logarithmic in the condition number) is exactly as proved in [15, 16] (see Section 4) for the special case when the number of regions in which the coefficient has a constant value

TABLE IV
Effect of C_ℓ on the Iteration Count (N = 512, with Coarse Grid)

	$C_\ell = 5$		$C_\ell = 20$	
σ^2	No. It.	$\Vert\mathbf{z}^i\Vert_2/\Vert\mathbf{z}^0\Vert_2$	No. It.	$\Vert\mathbf{z}^i\Vert_2/\Vert\mathbf{z}^0\Vert_2$
1	29	6.24×10^{-10}	27	5.87×10^{-10}
2	34	5.08×10^{-10}	27	5.69×10^{-10}
4	46	3.85×10^{-10}	30	6.51×10^{-10}
6	62	3.07×10^{-10}	35	4.19×10^{-10}
8	82	1.91×10^{-10}	41	3.37×10^{-10}

TABLE V
Effect of Aspect Ratio L on Iteration Count (with Coarse Grid)

L	No. It.	$\|\mathbf{z}^i\|_2/\|\mathbf{z}^0\|_2$
1	31	5.47×10^{-10}
4	42	6.01×10^{-10}
16	50	5.13×10^{-10}
64	65	7.56×10^{-10}

is small compared to the number of elements on the fine mesh. Here we have computed the harder problem where the coefficient has a different value on each element but we still observe the same good behaviour. It remains an open question to prove this.

Recall that for a physically realistic model we assume (see (5.1)) that the length scale λ decreases linearly in $1/N$. In the previous Tables I–III we took $C_\ell = 10$ in (5.1). In Table IV we illustrate the cases $C_\ell = 5$, 20. As expected, the smaller value of C_ℓ leads to neighbouring values of k being less well correlated, and thus a larger number of PCG iterations are needed to solve this "rougher" problem.

In groundwater flow calculations in practice it is often necessary to study flows in long thin regions. In Table V we repeat some of the above calculations for the case when the domain Ω is $[0, L] \times [0, 1]$ and we study the effect of varying the aspect ration $L \geq 1$ of the domain. In the absence of any additional information concerning anisotropy, in general for such problems we would need to take the same mesh diameter in both coordinate directions to ensure adequate accuracy. Thus, for each value of L, we here construct a uniform tensor product mesh with N_L subdivisions along the side [0, 1] and $L \times N_L$ subdivisions along the side $[0, L]$. However, in order to compare problems of the same dimension, N_L is chosen to ensure that the total number of degrees of freedom in the system is fixed at $n = (N+1) * (N-1)$, with $N = 256$. For these experiments $\sigma^2 = 4$ and $\lambda = 10/256$. The iteration count, as L increases, is given in Table V. A very modest growth with L is observed.

Our final table, Table VI, illustrates the parallel efficiency of the algorithm. To understand these results, recall that the DOUG solver described in Section 4 is organised on a master/slave model. In the construction of the preconditioner, the coarse mesh is assembled and solved on the master processor while the slaves handle the solves on the subdomains. Similarly, in

TABLE VI
Parallel Efficiency on SP2 ($\sigma^2 = 2$, $N = 256$, $\lambda = 1/N$)

	Without coarse grid		With coarse grid	
No. slaves	Time (s)	Efficiency	Time (s)	Efficiency
1	156.71	100%	12.65	100%
2	79.86	98%	6.25	101%
4	43.53	90%	3.15	100%
6	30.22	86%	2.04	103%
8	24.41	80%	1.62	98%
10	20.58	76%	1.34	94%
12	19.35	67%	1.19	89%
14	17.20	65%	1.10	82%

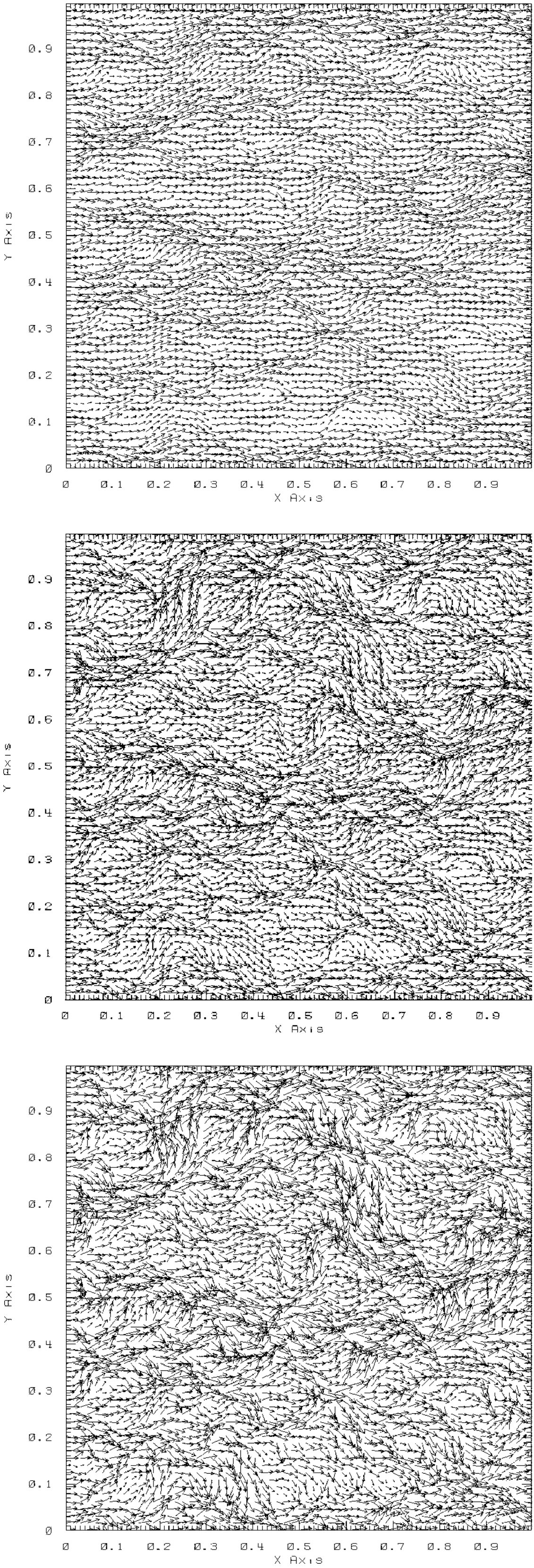

FIG. 4. Vector plot of the velocity for $\sigma^2 = 1$, 4, and 8 ($N = 256$, $\lambda = 10/N$).

the execution of dot and matrix–vector products, the slaves do the local calculations while the master is responsible for collating global information (see also [19]). In Table VI we give the parallel efficiency results as a function of the number of slave processors. The times recorded are those obtained on the 16 node IBM SP2 at the Daresbury Laboratory, United Kingdom (Peak performance: 20 GFlops/sec per processor). The efficiency column is computed in each case as $t(1)/(st(s))$, where $t(s)$ is the time required by the solver when s slaves are used. Because of the master-slave set up, it may be argued that to achieve the timing $t(s)$ we actually use $s+1$ processors. While this is strictly true, it is easily seen that if we recomputed the efficiencies using the formula $2t(1)/((s+1)t(s))$, then efficiencies of well over 100% will result. These figures indicate that there is no bottleneck present in communication between master and slave. In effect, the bulk of the computation is done on the slaves and so the figures in Table VI give an accurate impression of the parallel efficiency of the algorithm.

In Table VI, note especially the improved parallel efficiency of the method with the coarse grid compared to that without. This indicates the success of the the parallelization strategy implemented in DOUG: the coarse solve is not only necessary to obtain good theoretical results, it also gives much improved timings and efficiency even though in principle much more communication is needed. The key is the overlapping of communication with computation implemented in DOUG [19, 20].

Efficiencies of greater than 100% for small numbers of processors are not unusual, due to cache effects as well as small differences in the actual quality of the solution produced at the end of the PCG iteration (see also [19]).

Finally, in Fig. 4 we plot the velocity fields corresponding to the problem (1.1) and (1.2) with boundary conditions (5.2) in the case $N = 256$, $\lambda = 10/N$, and $\sigma^2 = 1, 4, 8$ respectively. Note the increased dispersion in the flow paths as σ^2 increases.

ACKNOWLEDGMENTS

We thank Professor Thomas Russell for kindly sending us a copy of Hecht's unpublished manuscript [22]. We also thank Dr. Andy Wood for useful discussions.

Note added in proof. After this work was completed, the paper "Mixed Finite Element Methods and Tree–Cotree Implicit Condensation," by P. Alotto and I. Perugia [*Calcolo* **36**, 233 (1999)], came to our attention. This paper uses related algebraic techniques to speed up the iterative solution of saddle-point problems. However, the reduced systems which result there are of Schur-complement type and therefore entirely different from ours.

REFERENCES

1. R. J. Adler, *The Geometry of Random Fields* (Wiley, Chichester, 1980).
2. S. F. Ashby, R. D. Falgout, S. G. Smith, and T. W. Fogwell, Multigrid preconditioned conjugate gradients for the numerical solution of groundwater flow on the Cray T3D, in *Proc. ANS Confer. on Mathematics and Computations, Reactor Physics, and Environmental Analysis* (Portland, OR, 1995), Vol. 1, p. 405.
3. F. Brezzi and M. Fortin, *Mixed and Hybrid Finite Element Methods* (Springer-Verlag, New York, 1991).
4. Z. Cai, R. R. Parashkevov, T. F. Russell, and X. Ye, Domain decomposition for a mixed finite element method in three dimensions, *SIAM J. Numer. Anal.* (1998), to appear.
5. T. F. Chan and T. Mathew, Domain decomposition methods, in *Acta Numerica 1994* (Cambridge Univ. Press, Cambridge, UK, 1994).
6. T. F. Chan, B. F. Smith, and J. Zou, Overlapping Schwarz methods on unstructured meshes using non-matching course grids, *Numer. Math.* **73**, 149 (1996).

7. G. Chavent, G. Cohen, J. Jaffre, M. Dupuy, and I. Ribera, Simulation of two-dimensional water flooding by using mixed finite elements, *Soc. Pet. Eng. J.* **24**, 382 (1984).

8. P. G. Ciarlet, *The Finite Element Method for Elliptic Problems* (North–Holland, Amsterdam, 1978).

9. K. A. Cliffe, I. G. Graham, R. Scheichl, and L. Stals, Parallel Computation of Flow in Heterogeneous Media Using Mixed Finite Elements, Bath Mathematics Preprint 99/16 (University of Bath, 1999).

10. N. A. C. Cressie, *Statistics for Spatial Data* (Wiley, London, 1993).

11. G. Dagan, *Flow and Transport in Porous Formations* (Springer-Verlag, New York, 1989).

12. C. R. Dietrich and G. N. Newsam, A fast and exact method for multidimensional Gaussian stochastic simulations. *Water Resour. Res.* **29**(8), 2861 (1993).

13. R. E. Ewing and J. Wang, Analysis of the Schwarz algorithm for mixed finite element methods, *RAIRO Model. Math. Anal. Num.* **26**(6), 739 (1992).

14. L. W. Gelhar, A stochastic conceptual analysis of one-dimensional groundwater flow in nonuiform homogeneous media, *Water Resour. Res.* **11**, 725 (1975).

15. I. G. Graham and M. J. Hagger, Unstructured additive Schwarz–CG method for elliptic problems with highly discontinuous coefficients, *SIAM J. Sci. Comput.* **20**, 2041 (1999).

16. I. G. Graham and M. J. Hagger, Additive Schwarz, CG and discontinuous coefficients, in *Domain Decomposition Methods in Science and Engineering*, edited by P. E. Bjørstad, M. S. Espedal, and D. E. Keyes (Domain Decomposition Press, Bergen, 1998).

17. D. F. Griffiths, The construction of approximately divergence-free finite elements, in *The Mathematics of Finite Elements and Applications* (Academic Press, New York, 1979), Vol. 3.

18. A. L. Gutjahr, D. McKay, and J. L. Wilson, Fast Fourier transform methods for random field generation, *Eos Trans. AGU* **68**(44), 1265 (1987).

19. M. J. Hagger, Automatic domain decomposition on unstructured grids (DOUG), *Adv. Comput. Math.* **9**, 281 (1998).

20. M. J. Hagger and L. Stals, DOUG User Guide, Version 1.98, Technical Report, University of Bath (1998), *available at* http://www.maths.bath.ac.uk/~parsoft.

21. F. Hecht, Construction d'une base de fonctions P1 non-conformes à divergence nulle dans $\mathbb{R}^3$, *RAIRO Model. Math. Anal. Num.* **15**, 119 (1981).

22. F. Hecht, Construction d'une base pour des éléments finis mixtes à divergence faiblement nulle, unpublished report (1988).

23. R. Hiptmair, T. Schiekofer, and B. Wohlmuth, Multilevel preconditioned augmented Lagrangian techniques for 2nd order mixed problems, *Computing* **57**, 25 (1996).

24. R. J. Hoeksema and P. K. Kitanidis, Analysis of the spatial structure of properties of selected aquifers, *Water Resour. Res.* **21**, 825 (1985).

25. G. Karypis and V. Kumar, A fast and high quality multilevel scheme for partitioning irregular graphs, *SIAM J. Sci. Comput.* **20**, 359 (1999).

26. E. F. Kaasschieter, A practical termination criterion for the conjugate gradient method, *BIT* **28**, 308 (1988).

27. C. E. Kolterman and S. M. Gorelick, Heterogeneity in sedimentary deposits: A review of structure-imitating, process-imitating and descriptive approaches, *Water Resour. Res.* **32**(9), 2617 (1996).

28. A. Mantoglou, Digital simulation of multivariate two- and three-dimensional stochastic processes with a spectral Turning Bands method, *Math. Geol.* **19**(2), 129 (1987).

29. T. P. Mathew, Schwarz alternating and iterative refinement methods for mixed formulations of elliptic problems. 1. Algorithms and numerical results, *Numer. Math.* **65**, 445 (1993).

30. J. C. Nédélec, Éléments finis mixtes incompressibles pour l'équation de Stokes dans $\mathbb{R}^3$, *Numer. Math.* **39**, 97 (1982).

31. C. C. Paige and M. A. Saunders, Solution of sparse indefinite systems of linear equations, *SIAM J. Num. Anal.* **12**(4), 617 (1975).

32. P. A. Raviart and J. M. Thomas, A mixed finite element method for 2-nd order elliptic problems, in *Mathematical Aspects of the Finite Element Method*, edited by I. Galligani and E. Magenes, Lecture Notes in Mathematics (Springer-Verlag, New York, 1977), Vol. 606, p. 292.

33. M. L. Robin, A. L. Gutjahr, E. A. Sudicky, and J. L. Wilson, Cross-correlated random field generation with the direct Fourier transform method, *Water Resour. Res.* **29**, 2395 (1993).

34. R. Scheichl, A decoupled iterative method for mixed problems using divergence-free finite elements, Bath Mathematics Preprint 00/11 (University of Bath, 2000).

35. R. Scheichl, *Iterative Solution of Saddle-Point Problems Using Divergence-Free Finite Elements with Applications to Groundwater Flow*, Ph.D. thesis, in preparation (University of Bath, 2000).

36. F. Thomasset, *Implementation of Finite Element Methods for Navier–Stokes Euations* (Springer-Verlag, New York, 1981).

37. A. F. B. Tompson, R. Ababou, and L. W. Gelhar, Implementation of the three-dimensional Turning Bands random field generator, *Water Resour. Res.* **25**(10), 2227 (1989).

38. C. Wagner, W. Kinzelbach, and G. Wittum, Schur-complement multigrid, a robust method for groundwater flow and transport problems, *Numer. Math.* **75**, 523 (1997).