Parallel Computation of Flow in Heterogeneous Media Modelled by Mixed Finite Elements*

K.A. Cliffe[†] I.G. Graham[‡] R. Scheichl[§] L. Stals[¶]

Abstract

In this paper we describe a fast parallel method for 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 is applicable not only to this problem but can be used to solve general unstructured finite element systems on a wide range of parallel architectures.

Keywords: Raviart-Thomas Mixed Finite Elements, Second-order Elliptic Problems, Divergence-free Space, Heterogeneous Media, Groundwater Flow, Domain Decomposition, Parallel Computation

Mathematics Subject Classification: 65N55, 65N30, 65N22, 65F10, 65Y05, 76S05

*This work was supported by EPSRC grant GR/L31715 and EPSRC CASE Award 97D00023

[†]AEA Technology, Harwell, Oxon OX11 0RA, UK. `andrew.cliffe@aeat.co.uk`

[‡]Dept of Mathematical Sciences, University of Bath, Bath BA2 7AY, UK. `igg@maths.bath.ac.uk`

[§]Dept of Mathematical Sciences, University of Bath, Bath BA2 7AY, UK. `maprs@bath.ac.uk`

[¶]Dept of Computer Science, Old Dominion University, Norfolk, VA 23529-0162, USA. `stals@icase.edu`

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 acceleration due to gravity. The functions $\vec{q}$ and P_R are both to be determined from (1.1), (1.2), with k denoting permeability and μ denoting the dynamic viscosity.

In contrast to the classical deterministic 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), (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), (1.2) when k is any typical realisation. Although we do not carry 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), (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), (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 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. Systems of this size and difficulty are beyond the limit of what has previously been done in the groundwater flow literature and 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 in this computation 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, high order elements are inefficient and so we discretise (1.1), (1.2) using the lowest order mixed Raviart-Thomas elements on triangular meshes ([36, 4]). Here P_R is approximated in the space of piecewise constant functions and $\vec{q}$ is approximated in an appropriate subspace of the vector-valued piecewise linear functions, where the normal component of $\vec{q}$ is required to be continuous across the element boundaries. 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.

As is well-known this type of discretisation yields a symmetric indefinite system of *saddle-point* type. Especially in combination with the large variations in k and N, this makes it very difficult to solve quickly enough for the multiple statistical simulations described above. In this paper we describe a fast parallel solver for these systems. The results in §6 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, with a modest decrease in efficiency thereafter.

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, automatically satisfying 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 solved independently 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 5 times smaller than the original saddle-point problem. A further side 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. [6]) 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 even in the case of problems with highly discontinuous coefficients when discretised on unstructured meshes ([20, 19]). 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] and [41], although there the emphasis is on finite volume/multigrid techniques.

The layout of this paper is as follows. In §2 we shall describe the background to the statistical models of heterogeneous media. In §3 we briefly review the discretisation of (1.1), (1.2) using the Raviart-Thomas elements, leading to the saddle-point system. In a brief appendix we explain why it is sufficient in terms of accuracy to sample the permeability k at the centroids of the finite elements. In §4 we describe the reduction of the saddle-point system to an SPD system and in §5 we describe briefly the additive Schwarz procedure and its resilience to discontinuous coefficients as well as our "black-box" package ([24],[23]). In §6 we give a sequence of experiments which show the performance of the method.

Acknowledgement. We would like to thank Professor Thomas Russell for kindly sending us a copy of Hecht's unpublished manuscript [26]. We also thank Dr. Andy Wood for useful discussions.

2 Statistical Modelling of Heterogeneous Media

The flow of fluids in the rocks comprising 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 in order to model the flow of groundwater. Simple-minded interpolation of measured values yields rather inaccurate permeability fields that do not reflect the heterogeneity which is known to be present.

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. [12]

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), (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), (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 [1] or [10].

We recall the notion of a *Gaussian random variable* Z which is specified by its mean (or expectation) $m = E\{Z\}$ and variance σ^2. More generally, a vector $Z = (Z_1, \ldots, Z_n)^T$ of n Gaussian random variables is completely specified by the vector $m \in \mathbb{R}^n$ (containing the mean values of each of these variables) together with its $n \times n$ *covariance matrix* $\Sigma = E\{(Z-m)(Z-m)^T\}$.

Generalising this to the infinite dimensional setting, 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

$$\Sigma(\vec{x},\vec{y}) := E\{(Z(\vec{x}) - m(\vec{x}))\,(Z(\vec{y}) - m(\vec{y}))\}\ , \quad \vec{x},\vec{y} \in \Omega. \tag{2.1}$$

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

$$m(\vec{x}) := m\ , \quad \Sigma(\vec{x},\vec{y}) := \sigma^2 \exp(-|\vec{x}-\vec{y}|/\lambda)\ , \tag{2.2}$$

for positive constants m, σ and λ. Note that evaluating (2.1) at $\vec{y} = \vec{x}$ and combining with (2.2) shows that for each $\vec{x} \in \Omega$, the random variable $Z(\vec{x})$ is normally distributed with mean m and variance σ^2 (independent of $\vec{x}$). Moreover, expanding (2.1), rearranging and using (2.2) shows

$$E\{Z(\vec{x})Z(\vec{y})\} = \Sigma(\vec{x},\vec{y}) + m^2 = \sigma^2 \exp(-|\vec{x}-\vec{y}|/\lambda) + m^2\ , \tag{2.3}$$

from which we can deduce

$$\begin{aligned} E\{(Z(\vec{x}) - Z(\vec{y}))^2\} &= E\{Z(\vec{x})^2\} + E\{Z(\vec{y})^2\} - 2E\{Z(\vec{x})Z(\vec{y})\} \\ &= 2\sigma^2\,[1 - \exp(-|\vec{x}-\vec{y}|/\lambda)] \\ &= \gamma(|\vec{x}-\vec{y}|)\ , \end{aligned} \tag{2.4}$$

where

$$\gamma(r) := 2\sigma^2(1 - \exp(-r/\lambda))\ . \tag{2.5}$$

The function γ is often called a *variogram* - see, e.g. [10].

We shall solve (1.1), (1.2) in the case when $\log k$ is a realisation of such a random field. There is some evidence from field data that this gives a reasonable representation of reality in certain cases ([17, 28]). There are many good methods for generating realisations of Gaussian random fields, including those based on FFT [22, 37], direct simulation [13], and the Turning Bands method [31, 32, 40]. Here we use the Turning Bands approach which represents the field as a superposition of one-dimensional fields 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 and the following theorem can be deduced from Adler [1].

Theorem 2.1 *Let $0 < \alpha < 1/2$ and let X denote any realisation of the Gaussian random field Z introduced above. Then with probability* 1,

$$|X(\vec{x}) - X(\vec{y})| \leq A|\vec{x} - \vec{y}|^{\alpha} \ , \quad \vec{x}, \vec{y} \in \Omega$$

for some positive constant A.

Proof. Without loss of generality we can assume that the polygonal domain Ω is a subset of $[0,1] \times [0,1]$. (If this is not the case, choose $a \in \mathbb{R}$ and $\mathbf{b} \in \mathbb{R}^2$ such that the affine map $\tilde{\mathbf{x}} \mapsto a\tilde{\mathbf{x}} + \mathbf{b}$ provides a bijection between a polygon $\tilde{\Omega} \subset [0,1] \times [0,1]$ and Ω. Then $\tilde{Z}(\tilde{\mathbf{x}}) := Z(a\tilde{\mathbf{x}} + \tilde{\mathbf{b}})$ defines a statistically homogeneous Gaussian random field on $\tilde{\Omega}$, with variogram $\tilde{\gamma}(r) = 2\sigma^2(1 - \exp(-ar/\lambda))$ and Hölder continuity of realisations of $\tilde{Z}$ will imply the analogous result for Z.)

So, assuming that $\Omega \subset [0,1] \times [0,1]$, we first note that the random field Z is just the restriction to Ω of the Gaussian random field on $[0,1] \times [0,1]$ with the same variogram. The regularity of any realisation of Z can then be deduced from the asymptotics of the variogram $\gamma(r)$ as $r \to 0$. Because $\sqrt{\gamma(r)} = O(r^{1/2})$ it follows that $Z - m$ is an index-1/2 (2,1) Gaussian field in the sense of ([1, Definition 8.3.1]) and the result follows from Theorem 8.3.2 of the same reference.

□

Once system (1.1), (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:

$$X_{jl} := E\{X'_j X'_l\} \ , \quad j, l = 1, 2 \ . \tag{2.6}$$

X'_j satisfies the differential equation

$$\frac{dX'_j}{dt} = q_j \quad , \quad j = 1, 2. \tag{2.7}$$

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 possibility of computation of the many particle paths which would be required in statistical analyses.

3 Mixed Finite Element Discretisation

In this section we describe the mathematical setting for (1.1),(1.2) together with its discretisation by mixed finite elements. Since this is a standard procedure (see, e.g.

[4]), we shall be brief. For ease of notation, from now on we replace P_R by p in (1.1) and so the system to be solved is

$$\vec{q} + (k/\mu)\,\vec{\nabla} p = \vec{0}\,, \tag{3.1}$$
$$\vec{\nabla}\cdot\vec{q} = 0\,. \tag{3.2}$$

Let Ω denote a simply connected open domain in $\mathbb{R}^2$ 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 [15]).

Let $\vec{\nu}(\vec{x})$ denote the outward unit normal from Ω at $\vec{x} \in \Gamma$. We describe the solution of (3.1),(3.2) in the case when μ is a positive constant and k is a general symmetric uniformly positive definite 2×2 matrix-valued function on Ω. (In §6 k will be a scalar multiple of the identity.) This system is to be solved subject to mixed boundary conditions:

$$p = g \quad \text{on} \quad \Gamma_D, \qquad \text{and} \quad \vec{q}\cdot\vec{\nu} = 0 \quad \text{on} \quad \Gamma_N\,. \tag{3.3}$$

To discretise (3.1), (3.2) we put it in weak form. Let $(\cdot,\cdot)_{L_2(\Omega)^d}$ denote the usual inner product in $L_2(\Omega)^d$, for $d = 1, 2$. Then introduce the Hilbert space

$$H(\mathrm{div},\Omega) := \{\vec{v} \in L_2(\Omega)^2 : \mathrm{div}\,\vec{v} \in L_2(\Omega)\},$$

with the inner product

$$(\vec{u},\vec{v})_{H(\mathrm{div},\Omega)} := (\vec{u},\vec{v})_{L_2(\Omega)^2} + (\mathrm{div}\,\vec{u}, \mathrm{div}\,\vec{v})_{L_2(\Omega)},$$

and its subspace

$$H_{0,N}(\mathrm{div},\Omega) := \{\vec{v} \in H(\mathrm{div},\Omega) : \vec{v}\cdot\vec{\nu}|_{\Gamma_N} = 0\}$$

(see [4] for details). Introduce the bilinear forms

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

and the linear functional

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

Then the weak form of (3.1),(3.2) is to find $(\vec{q},p) \in H_{0,N}(\mathrm{div},\Omega) \times L_2(\Omega)$ such that

$$\left.\begin{array}{lcll} m(\vec{q},\vec{v}) + b(\vec{v},p) & = & G(\vec{v})\,, & \text{for all } \vec{v} \in H_{0,N}(\mathrm{div},\Omega), \\ b(\vec{q},w) & = & 0\,, & \text{for all } w \in L_2(\Omega). \end{array}\right\} \tag{3.4}$$

The mixed finite element discretisation of (3.4) is obtained by choosing finite dimensional subspaces $\mathcal{V} \subset H_{0,N}(\mathrm{div},\Omega)$ and $\mathcal{W} \subset L_2(\Omega)$ and seeking $(\vec{Q},P) \in \mathcal{V} \times \mathcal{W}$ such that

$$\left.\begin{array}{lcll} m(\vec{Q},\vec{V}) + b(\vec{V},P) & = & G(\vec{V})\,, & \text{for all } \vec{V} \in \mathcal{V}, \\ b(\vec{Q},W) & = & 0\,, & \text{for all } W \in \mathcal{W}. \end{array}\right\} \tag{3.5}$$

In practice this is implemented 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 $\mathcal{W}$. By writing

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

problem (3.4) is then reduced to the indefinite linear equation system

$$\begin{pmatrix} M & B \\ B^T & 0 \end{pmatrix} \begin{pmatrix} \mathbf{q} \\ \boldsymbol{p} \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ \mathbf{0} \end{pmatrix} \qquad \text{in} \quad \mathbb{R}^{n_{\mathcal{V}}} \times \mathbb{R}^{n_{\mathcal{W}}} \ , \tag{3.6}$$

where $M_{i,i'} := m(\vec{v}_i, \vec{v}_{i'})$ is the "mass matrix", $B_{i,j} := b(\vec{v}_i, w_j)$ is the "discrete gradient", and $f_i := G(\vec{v}_i)$.

Here we restrict attention to the case when $\mathcal{V}$ is the lowest order Raviart-Thomas space on triangles ([36]). To define this, 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, so that, for $E \in \mathcal{E}$, $\overline{E}$ denotes the closure of E (including its end-points). For any $E \in \mathcal{E}$, we let $\vec{\nu}_E$ denote the unit normal to the edge E which, for convenience, is assumed orientated 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 $\vec{v} \in H_{0,N}(\mathrm{div}, \Omega)$ such that for all $T \in \mathcal{T}$, there exist scalars α_T, β_T and γ_T such that

$$\vec{v}(\vec{x}) = \begin{pmatrix} \alpha_T \\ \beta_T \end{pmatrix} + \gamma_T \vec{x}, \qquad \text{for all} \quad \vec{x} \in T. \tag{3.7}$$

Equivalently we can define $\mathcal{V}$ to be the space of all $\vec{v} : \Omega \to \mathbb{R}^2$ which satisfy (3.7) for each $T \in \mathcal{T}$, such that

$$\text{and} \quad \begin{array}{ll} \text{(i)} & \vec{v} \cdot \vec{\nu}_E \quad \text{is continuous across each edge } E \in \mathcal{E}_I \\ \text{(ii)} & \vec{v} \cdot \vec{\nu}_E = 0 \quad \text{for all} \quad E \in \mathcal{E}_N. \end{array} \tag{3.8}$$

Because of the special form of (3.7) it is easily shown that $\vec{v} \cdot \vec{\nu}_E$ is constant on any edge E of T. Thus $\vec{v} \in \mathcal{V}$ can be completely determined by specifying the constant value of $\vec{v} \cdot \vec{\nu}_E$ for each $E \in \mathcal{E}_I \cup \mathcal{E}_D$. This leads us to introduce the standard basis for $\mathcal{V}$ which 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

$$\vec{v}_E \cdot \vec{\nu}_{E'} = \delta_{E,E'} \tag{3.9}$$

with δ denoting the Kronecker delta.

We also have to specify the space $\mathcal{W}$. To fulfill the discrete inf-sup condition (see [4]), necessary for existence and uniqueness, $\mathcal{W}$ has to be 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

$$n_{\mathcal{V}} = (\#\mathcal{E}_I + \#\mathcal{E}_D), \qquad n_{\mathcal{W}} = \#\mathcal{T} \ , \tag{3.10}$$

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

Although the analysis below is given for system (3.6) with permeability k, in the practical implementation of (3.6) 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. We show in the appendix that this approximation does not degrade the estimate for the error $\vec{q} - \vec{Q}$ (measured in an appropriate norm). In the next section we begin our consideration of fast methods for solving (3.6).

4 Solenoidal Velocity Elements

In this section we formulate our method for decoupling the vector of velocities $\boldsymbol{q}$ from the vector of pressures $\boldsymbol{p}$ in system (3.6). Recall ([4]) that (3.6) has a unique solution $(\boldsymbol{q}, \boldsymbol{p}) \in \mathbb{R}^{n_\mathcal{V}} \times \mathbb{R}^{n_\mathcal{W}}$ for all $\mathbf{f} \in \mathbb{R}^{n_\mathcal{V}}$, and clearly $\boldsymbol{q}$ is in $\ker B^T$. To describe our decoupling procedure, consider first (3.6) as an abstract system.

4.1 General Procedure

The decoupling can be achieved by finding

$$\text{a basis} \quad \{\mathbf{z}_1, \ldots, \mathbf{z}_{\mathring{n}}\} \quad \text{of} \quad \ker B^T \ . \tag{4.1}$$

(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 (3.6) can be written

$$\boldsymbol{q} = \sum_{j=1}^{\mathring{n}} \mathring{q}_j \mathbf{z}_j = Z^T \mathring{\boldsymbol{q}} \ , \tag{4.2}$$

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 (3.6) by Z shows that $\mathring{\boldsymbol{q}}$ is a solution of the linear system

$$\mathring{A} \mathring{\boldsymbol{q}} = \mathring{\mathbf{f}} \tag{4.3}$$

where

$$\mathring{A} = ZMZ^T \quad \text{and} \quad \mathring{\mathbf{f}} = Z\mathbf{f} \, . \tag{4.4}$$

Since M is SPD, so is $\mathring{A}$ and $\mathring{\boldsymbol{q}}$ is the unique solution of (4.3). Thus if the basis (4.1) can be found then the velocity $\boldsymbol{q}$ in (3.6) can be computed by solving the decoupled SPD system (4.3) rather than the indefinite coupled system (3.6).

In applications to porous media flow, where one is primarily interested in the velocity $\vec{q}$ in (3.1), (3.2), the method described above is of great relevance. Even when the pressure p is also of interest our method may still be highly competitive, provided we can also compute a *complementary basis* $\{\mathbf{z}_{\mathring{n}+1}, \ldots, \mathbf{z}_{n_\mathcal{V}}\}$ with the property that

$$\mathrm{span}\{\mathbf{z}_1, \ldots, \mathbf{z}_{\mathring{n}}, \mathbf{z}_{\mathring{n}+1}, \ldots, \mathbf{z}_{n_\mathcal{V}}\} = \mathbb{R}^{n_\mathcal{V}} \ . \tag{4.5}$$

If this is known and if Z' denotes the matrix with rows $\mathbf{z}_{\mathring{n}+1}^T, \ldots, \mathbf{z}_{n_\mathcal{V}}^T$, then multiplying the first (block) row of (3.6) by Z' shows that $\boldsymbol{p}$ is the solution of the $n_\mathcal{W} \times n_\mathcal{W}$ system

$$(Z'B)\boldsymbol{p} = Z'(\mathbf{f} - M\boldsymbol{q}) \ . \tag{4.6}$$

An elementary argument shows that $Z'B$ is non-singular and so the unique solution $\boldsymbol{p}$ of (4.6) also determines the pressure in (3.6) once the velocity $\boldsymbol{q}$ is known.

We show in the next two subsections that in the particular case of the mixed finite element system (3.6)

(i) It is always easy to find the basis (4.1).

(ii) The resulting matrix $\mathring{A}$ in the reduced problem (4.3) 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 (4.3) is about 5 times smaller than (3.6) .

We also point out in §4.4 that a choice of complimentary basis can be made so that the coefficient matrix $Z'B$ in the system (4.6) is lower triangular. However, since our main concern in this paper is the computation of the velocity $\boldsymbol{q}$, the details of the pressure computation are left for later work.

To establish conclusions (i) – (iii) we need to exploit the particular properties of (3.6). In particular note that finding the basis $\mathbf{z}_1, \ldots, \mathbf{z}_{\mathring{n}}$ in (4.1) is equivalent to finding a basis $\vec{\mathring{v}}_1, \ldots, \vec{\mathring{v}}_{\mathring{n}}$ of the finite element space

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

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

$$\vec{\mathring{v}}_i = \sum_{j=1}^{n_{\mathcal{V}}} Z_{i,j} \vec{v}_j \ , \quad i = 1, \ldots, \mathring{n}, \tag{4.7}$$

(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) can be found from (4.7).

We now turn our attention to finding a basis of $\mathring{\mathcal{V}}$.

4.2 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:

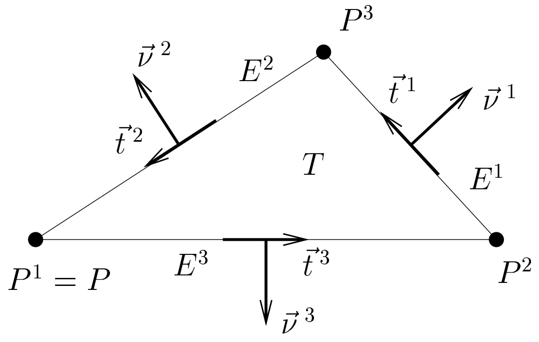
$$\vec{\Psi}_P = \vec{\mathrm{curl}} \Phi_P \tag{4.8}$$

(i.e. Φ_P is the stream function for $\vec{\Psi}_P$).[1] The functions (4.8) clearly satisfy $\mathrm{div}\vec{\Psi}_P = 0$ on each triangle of the mesh, and a subset of them lie in $\mathring{\mathcal{V}}$ as the following proposition shows.

Proposition 4.1 *For each $P \in \mathcal{N}_I \cup \mathcal{N}_D$, $\vec{\Psi}_P \in \mathring{\mathcal{V}}$.*

Proof. This fact is observed at various points in the literature (e.g. [15]). We give a brief proof for completeness, and also because some of the details are useful later on.

First consider the general case of $P \in \mathcal{N}$. Then clearly supp $\vec{\Psi}_P$ consists only of the triangles touching node P. A typical such triangle with nodes $P^1 := P$, P^2, and P^3 is depicted below,



with $\vec{t}^{\,\alpha}$ and $\vec{\nu}^{\,\alpha}$ denoting unit vectors in the directions shown, and $P^\alpha = (p_1^\alpha, p_2^\alpha)^T$, $\alpha = 1, 2, 3$. Then

$$\Phi_P(\vec{x}) := \frac{1}{2|T|}\begin{vmatrix} 1 & x_1 & x_2 \\ 1 & p_1^2 & p_2^2 \\ 1 & p_1^3 & p_2^3 \end{vmatrix}, \quad \vec{x} \in T$$

and it follows easily that

$$\vec{\Psi}_P(\vec{x}) = \frac{|E^1|}{2|T|}\vec{t}^{\,1}, \quad \vec{x} \in T,$$

which is easily seen to be of the form (3.7). Furthermore

$$\left.\begin{aligned} \vec{\Psi}_P(\vec{x}) \cdot \vec{\nu}^{\,1} &= \frac{|E^1|}{2|T|}\,\vec{t}^{\,1} \cdot \vec{\nu}^{\,1} = 0 \\ \vec{\Psi}_P(\vec{x}) \cdot \vec{\nu}^{\,2} &= \frac{|E^1|}{2|T|}\,\vec{t}^{\,1} \cdot \vec{\nu}^{\,2} = \frac{|E^1|}{2|T|}\,\vec{t}^{\,1} \times \vec{t}^{\,2} = \frac{1}{|E^2|} \\ \vec{\Psi}_P(\vec{x}) \cdot \vec{\nu}^{\,3} &= \frac{|E^1|}{2|T|}\,\vec{t}^{\,1} \cdot \vec{\nu}^{\,3} = \frac{|E^1|}{2|T|}\,\vec{t}^{\,1} \times \vec{t}^{\,3} = -\frac{1}{|E^3|} \end{aligned}\right\} \vec{x} \in T\,. \tag{4.9}$$

Now to obtain the result observe that, since $\mathrm{div}\vec{\Psi}_P = 0$ trianglewise, it is sufficient to show that

$$\vec{\Psi}_P \in \mathcal{V}, \quad \text{for all } P \in \mathcal{N}_I \cup \mathcal{N}_D\,. \tag{4.10}$$

To obtain (4.10), consider first $P \in \mathcal{N}_I$. Let $E \in \mathcal{E}_I$. If $E \not\subset \mathrm{supp}\,\vec{\Psi}_P$ we have trivially

$$\vec{\Psi}_P \cdot \vec{\nu}_E \quad \text{is continuous across } E. \tag{4.11}$$

[1]Throughout, if $\vec{x}$, $\vec{y}$ are vectors in $\mathbb{R}^2$, then $\vec{x} \times \vec{y}$ denotes the x_3 component of the cross product of the vectors $(x_1, x_2, 0)^T$ and $(y_1, y_2, 0)^T$, and for any $\Phi : \mathbb{R}^2 \to \mathbb{R}$, $\vec{\mathrm{curl}}\Phi$ denotes the x_1 and x_2 components of the curl of the function $(0, 0, \Phi)^T$ which can be expressed $\vec{\mathrm{curl}}\;\Phi = (\partial\Phi/\partial x_2, -\partial\Phi/\partial x_1)^T$.

Now take a general triangle $T \subset \operatorname{supp} \vec{\Psi}_P$, as pictured above. If $E = E^3$, then performing the computation (4.9) in the other triangle adjoining E and combining with (4.9) establishes (4.11). Similarly (4.11) holds if $E = E^2$. On the other hand, when $E = E^1$, (4.11) also holds since $\vec{\Psi}_P \cdot \vec{\nu}_E|_T = 0$ and since the other triangle adjoining E lies outside $\operatorname{supp} \vec{\Psi}_P$. Altogether we have established that $\vec{\Psi}_P$ satisfies criterion (3.8)(i).

To establish (3.8)(ii), let $E \in \mathcal{E}_N$. If $E \not\subset \operatorname{supp} \vec{\Psi}_P$, then $\vec{\Psi}_P \cdot \vec{\nu}_E = 0$ trivially. If $E \subset T \subset \operatorname{supp} \vec{\Psi}_P$, then with the above notation E has to be E^1 and again $\vec{\Psi}_P \cdot \vec{\nu}_E = 0$, proving (3.8)(ii).

Thus we have shown that $\vec{\Psi}_P \in \mathcal{V}$ for all $P \in \mathcal{N}_I$. Similar arguments establish that $\vec{\Psi}_P \in \mathcal{V}$ for all $P \in \mathcal{N}_D$, proving (4.10). □

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 (3.9); in fact only those $\vec{v}_E$ corresponding to edges E touching node P appear in the expansion of $\vec{\Psi}_P$ (see Figure 1 (left)).

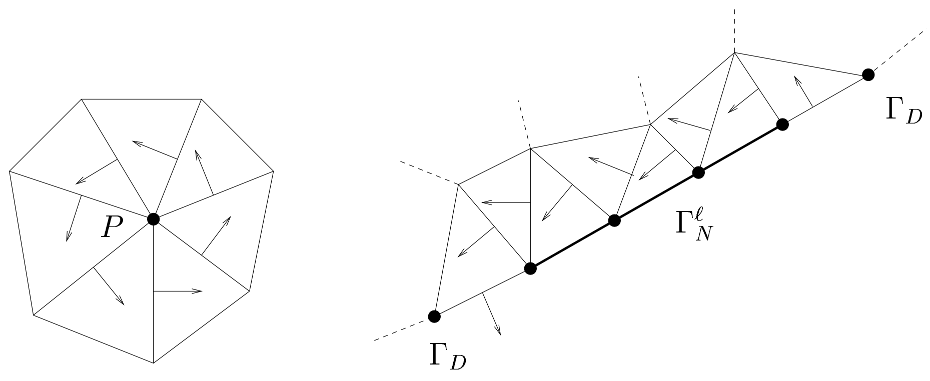


Figure 1: Solenoidal basis functions $\vec{\Psi}_P$ (left) and $\sum_{P \in \mathcal{N}_N^\ell} \vec{\Psi}_P$ (right)

The functions introduced in Proposition 4.1 span a subset of $\overset{\circ}{\mathcal{V}}$, but for general boundary conditions, there are not enough of them to constitute a basis for $\overset{\circ}{\mathcal{V}}$. A small number of additional basis functions may need to be added. These are introduced in Proposition 4.2. In this we let $n_{\mathcal{C}}$ denote the number of connected components in Γ_N and write

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

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.

Proposition 4.2 *For each* $\ell = 1, \ldots, n_{\mathcal{C}}$,

$$\sum_{P \in \mathcal{N}_N^\ell} \vec{\Psi}_P \in \overset{\circ}{\mathcal{V}} \ . \tag{4.12}$$

Proof. Let $\ell \in \{1, \ldots, n_{\mathcal{C}}\}$. Analogously to the proof of Proposition 4.1 it is sufficient to show that $\left(\sum_{P \in \mathcal{N}_N^\ell} \vec{\Psi}_P \right) \in \mathcal{V}$. Criterion (3.8)(i) is a mere consequence of (4.9) again, and to establish (3.8)(ii) observe first that the components $\Gamma_N^{\ell'}$, $\ell' = 1, \ldots, n_{\mathcal{C}}$, are disjoint, and therefore $\left(\sum_{P\in\mathcal{N}_N^\ell}\vec{\Psi}_P\right)\cdot\vec{\nu}_E = 0$ for all $E\subset\Gamma_N^{\ell'}$, $\ell'\neq\ell$, by definition. Now let $E\subset\Gamma_N^\ell$, and let $P^1,P^2\in\mathcal{N}_N^\ell$ be the two end points of E. Then, using (4.9),

$$\Big|\sum_{P\in\mathcal{N}_N^\ell}\vec{\Psi}_P\cdot\vec{\nu}_E\Big| = \left|\vec{\Psi}_{P^1}\cdot\vec{\nu}_E + \vec{\Psi}_{P^2}\cdot\vec{\nu}_E\right| = \left|\frac{1}{|E|}-\frac{1}{|E|}\right| = 0\,.$$

Thus $\sum_{P\in\mathcal{N}_N^\ell}\vec{\Psi}_P\cdot\vec{\nu}_E = 0$ for all $E\in\mathcal{E}_N$, and the proof is complete. □

In contrast to the functions in Proposition 4.1, the functions introduced in Proposition 4.2 are non-local linear combinations of the functions $\vec{v}_E$; however, the non-locality of $\sum_{P\in\mathcal{N}_N^\ell}\vec{\Psi}_P$ is confined to the vicinity of Γ_N^ℓ (see Figure 1 (right)). The number of such triangles is typically $O((\#\mathcal{T})^{1/2})$, so they are only modestly non-local.

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 4.1 with all but one of the functions in Proposition 4.2 we have the required basis.

Theorem 4.3 *Suppose $n_{\mathcal{C}}\neq 0$. Then the functions*

$$\{\vec{\Psi}_P : P\in\mathcal{N}_I\cup\mathcal{N}_D\}\cup\{\sum_{P\in\mathcal{N}_N^\ell}\vec{\Psi}_P : \ell=1,\ldots,n_{\mathcal{C}}-1\} \tag{4.13}$$

are a basis for $\mathring{\mathcal{V}}$.

Proof. Suppose $n_{\mathcal{C}}\neq 0$. First we shall check that the number of basis functions in (4.13) coincides with $\mathring{n}=\dim\mathring{\mathcal{V}}$. Consider a typical Neumann boundary segment Γ_N^ℓ. This contains $\#\mathcal{N}_N^\ell$ nodes, two of which are the end points of Γ_N^ℓ, and so the number of edges in Γ_N^ℓ is $\#\mathcal{N}_N^\ell-1$. Summing over $\ell=1,\ldots,n_{\mathcal{C}}$, we obtain

$$\#\mathcal{E}_N = \#\mathcal{N}_N - n_{\mathcal{C}}\ . \tag{4.14}$$

Now the number of functions in (4.13) is $\#\mathcal{N}_I+\#\mathcal{N}_D+n_{\mathcal{C}}-1$ and, using the fact that $\mathcal{N}_I$, $\mathcal{N}_D$ and $\mathcal{N}_N$ partition $\mathcal{N}$, together with (4.14), we have

$$(\#\mathcal{N}_I+\#\mathcal{N}_D+n_{\mathcal{C}}-1) = (\#\mathcal{N}-\#\mathcal{N}_N+n_{\mathcal{C}}-1) = (\#\mathcal{N}-\#\mathcal{E}_N-1)\ . \tag{4.15}$$

Furthermore, using the fact that $\mathcal{E}_I$, $\mathcal{E}_D$ and $\mathcal{E}_N$ partition $\mathcal{E}$, we have

$$(\#\mathcal{N}-\#\mathcal{E}_N-1) = (\#\mathcal{E}_I+\#\mathcal{E}_D)-(\#\mathcal{E}-\#\mathcal{N}+1) = (\#\mathcal{E}_I+\#\mathcal{E}_D)-\#\mathcal{T}\ , \tag{4.16}$$

where in the last step we have used Euler's Polyhedron Theorem and the assumption that Ω is simply connected. Now recalling that $\mathring{n}=n_{\mathcal{V}}-n_{\mathcal{W}}$ (and from §3 we have $n_{\mathcal{V}}=\#\mathcal{E}_I+\#\mathcal{E}_D$ and $n_{\mathcal{W}}=\#\mathcal{T}$), it follows from (4.15) and (4.16) that the number of functions in (4.13) is $\mathring{n}$, as required.

To complete the proof we merely need to show that the functions in (4.13) 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

$$\vec{0} = \sum_{P\in\mathcal{N}_I\cup\mathcal{N}_D}\alpha_P\vec{\Psi}_P + \sum_{\ell=1}^{n_{\mathcal{C}}-1}\beta_\ell\sum_{P\in\mathcal{N}_N^\ell}\vec{\Psi}_P\ . \tag{4.17}$$

This may be rewritten

$$\vec{0} = \sum_{P \in \mathcal{N}} \alpha_P \vec{\Psi}_P \ , \qquad (4.18)$$

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 $P \in \mathcal{N}_N^{n_\mathcal{C}}$. Now consider any edge $E \in \mathcal{E}$ with end points P' and P''.

If $E \in \mathcal{E}_I \cup \mathcal{E}_D$, combining (4.18) with (4.9) we get

$$0 = \Big| \sum_{P \in \mathcal{N}} \alpha_P \vec{\Psi}_P \cdot \vec{\nu}_E \Big| = \left| \alpha_{P'} \frac{1}{|E|} - \alpha_{P''} \frac{1}{|E|} \right| = \frac{1}{|E|} |\alpha_{P'} - \alpha_{P''}|$$

which shows $\alpha_{P'} = \alpha_{P''}$.

On the other hand if $E \in \mathcal{E}_N$, $\alpha_{P'} = \alpha_{P''}$ by definition. Since every two vertices can be connected by a series of edges, it follows that α_P is independent of P. Recalling that $\mathcal{N}_N^{n_\mathcal{C}} \neq \emptyset$, we have $\alpha_P = 0$ for all $P \in \mathcal{N}$. $\square$

Remark 4.4 *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 (4.8), 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. [8, 25] and later in the development of preconditioning strategies for the saddle point system (3.6) [15, 16, 33, 34, 27], This construction is also found in the subsequent unpublished manuscript [26], 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 (3.6). Recently the same idea has been investigated in 3D [5], although here the method is restricted to 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, [11, 21, 39, 25, 35]. However in the Stokes case, after decoupling, the discrete stream function appears underneath a (relatively ill-conditioned) fourth-order operator, whereas for groundwater 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.

4.3 Implementation

To implement the decoupled system (4.3) for determining $\mathring{\boldsymbol{q}}$ (and hence $\boldsymbol{q}$) we must work with the matrix $\mathring{A}$ and right hand side $\mathring{\mathbf{f}}$ specified in (4.4). We observe that these are formally defined in terms of multiplications with the matrix Z which, through (4.7), 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 (3.6) the $\{\vec{v}_j\}$ are the Raviart-Thomas velocity basis functions $\{\vec{v}_E : E \in \mathcal{E}_I \cup \mathcal{E}_D\}$ given in §3, whereas the $\{\vec{\mathring{v}}_i\}$ are the basis functions specified in Theorem 4.3. 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 $E \in \mathcal{E}_I \cup \mathcal{E}_D$.

Using this identification we can rewrite (4.7) as

$$\vec{\Psi}_P = \sum_{E\in\mathcal{E}_I\cup\mathcal{E}_D} Z_{P,E}\,\vec{v}_E, \qquad P \in \mathcal{N}_I \cup \mathcal{N}_D \tag{4.19}$$

and

$$\sum_{P\in\mathcal{N}_N^\ell} \vec{\Psi}_P = \sum_{E\in\mathcal{E}_I\cup\mathcal{E}_D} Z_{\ell,E}\,\vec{v}_E, \qquad \ell \in 1,\ldots,n_{\mathcal{C}}-1\,. \tag{4.20}$$

Note that the matrix Z is sparse, in fact $Z_{P,E} \neq 0$ only when node P is contained in edge E (Fig. 1, left), whereas $Z_{\ell,E} \neq 0$ only when the Neumann boundary segment Γ_N^ℓ touches the edge E (Fig. 1, right).

With the same convention we can write the elements of the matrix M appearing in (3.6) as

$$M_{E,E'} = m(\vec{v}_E, \vec{v}_{E'}), \qquad E, E' \in \mathcal{E}_I \cup \mathcal{E}_D\,. \tag{4.21}$$

With these observations, it is simple to write $\mathring{A}$ as a sum of element matrices. To be precise, if we write

$$M = \sum_{T\in\mathcal{T}} M_T, \qquad \text{where } (M_T)_{E,E'} = \int_T \mu k^{-1} \vec{v}_E \cdot \vec{v}_{E'}\,dx,$$

then

$$\mathring{A} = \sum_{T\in\mathcal{T}} \mathring{A}_T \tag{4.22}$$

where $\mathring{A}_T = Z_T M_T Z_T^T$, and Z_T denotes the matrix whose entries equal the entries of Z for columns corresponding to edges $E \subset T$ and rows corresponding to $P \in T$ and ℓ such that Γ_N^ℓ touches T and are zero elsewhere. The representation (4.22) may be important if iterative methods are being used to solve (4.3). A similar element-wise representation can be given for the computation of $\mathring{\mathbf{f}}$ in (4.4).

Alternatively (and this is used in the computations in §6) $\mathring{A}$ can be determined (element-wise or globally) from a standard piecewise linear approximation of a related bilinear form, without the assembly of any Raviart-Thomas stiffness matrix entries, as the following calculation shows.

Introduce the bilinear form

$$a(\zeta,\Phi) := (\mu K^{-1}\,\vec{\nabla}\zeta, \vec{\nabla}\Phi)_{L_2(\Omega)^2}, \qquad \zeta,\Phi \in H^1(\Omega), \tag{4.23}$$

where

$$K = S^T k S, \qquad \text{and} \quad S = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}. \tag{4.24}$$

(Note that $K = k$ when k is a scalar multiple of the identity.)

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 in §4.2. 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

$$C_{P,\ell'} := \sum_{P' \in \mathcal{N}_N^{\ell'}} \mathcal{A}_{P,P'}, \qquad P \in \mathcal{N}_I \cup \mathcal{N}_D, \quad \ell' = 1, \ldots, n_{\mathcal{C}} - 1$$

and

$$D_{\ell,\ell'} := \sum_{P \in \mathcal{N}_N^{\ell}} \sum_{P' \in \mathcal{N}_N^{\ell'}} \mathcal{A}_{P,P'}, \qquad \ell, \ell' = 1, \ldots, n_{\mathcal{C}} - 1 \ .$$

The following result identifies a simple formula for the matrix $\mathring{A}$. It shows that $\mathring{A}$ can be obtained from $\mathcal{A}$ using a small number of elementary operations on rows and columns corresponding to certain boundary nodes $P \in \mathcal{N}_N$.

Theorem 4.5

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

Remark 4.6 *The rôle of bilinear form (4.23) in the theory of the Raviart-Thomas approximation of second-order elliptic problems was pointed out by [15, 16] and [33, 34]. However, those references were not concerned with the construction of a basis for $\mathring{\mathcal{V}}$ and so (4.23) was not used there in the way it is used here.*

Proof of Theorem 4.5. Recall that from (4.4) we have $\mathring{A} = ZMZ^T$, and recall the notation for the entries of Z and M defined in (4.19)–(4.21).

Then, for $P, P' \in \mathcal{N}_I \cup \mathcal{N}_D$ we have

$$\begin{aligned} \mathring{A}_{P,P'} = \sum_{E,E' \in \mathcal{E}_I \cup \mathcal{E}_D} Z_{P,E} M_{E,E'} Z_{P',E'} = m(\vec{\Psi}_P, \vec{\Psi}_{P'}) &= (\mu k^{-1} \vec{\Psi}_P, \vec{\Psi}_{P'})_{L_2(\Omega)^2} \\ &= (\mu k^{-1} \vec{\mathrm{curl}} \Phi_P, \vec{\mathrm{curl}} \Phi_{P'})_{L_2(\Omega)^2}, \end{aligned}$$

where in the last step we used (4.8). Since $\vec{\mathrm{curl}}\Phi_P = S\vec{\nabla}\Phi_P$, and since $S^T k^{-1} S = K^{-1}$, we have

$$\mathring{A}_{P,P'} = (\mu k^{-1} S \vec{\nabla} \Phi_P, S \vec{\nabla} \Phi_{P'})_{L_2(\Omega)^2} = a(\Phi_P, \Phi_{P'}) = \mathcal{A}_{P,P'} = A_{P,P'} \ . \tag{4.25}$$

Similarly, for $P \in \mathcal{N}_I \cup \mathcal{N}_D$ and $\ell = 1, \ldots, n_{\mathcal{C}} - 1$ we have

$$\begin{aligned} \mathring{A}_{P,\ell'} &= \sum_{E,E' \in \mathcal{E}_I \cup \mathcal{E}_D} Z_{P,E} M_{E,E'} Z_{\ell',E'} = m(\vec{\Psi}_P, \sum_{E' \in \mathcal{E}_I \cup \mathcal{E}_D} Z_{\ell',E'} \vec{v}_{E'}) \\ &= m(\vec{\Psi}_P, \sum_{P' \in \mathcal{N}_N^{\ell'}} \vec{\Psi}_{P'}) = \sum_{P' \in \mathcal{N}_N^{\ell'}} m(\vec{\Psi}_P, \vec{\Psi}_{P'}) = \sum_{P' \in \mathcal{N}_N^{\ell'}} \mathcal{A}_{P,P'} = C_{P,\ell'} \ . \end{aligned}$$

Completely analogous arguments show that

$$\mathring{A}_{\ell,\ell'} = D_{\ell,\ell'}, \qquad \ell, \ell' = 1, \ldots, n_{\mathcal{C}} - 1 \ ,$$

and since $\mathring{A}$ is symmetric the theorem follows. □

Remark 4.7

(i) *Observe that the decoupled system (4.3) is about 5 times smaller than the original indefinite system (3.6). More precisely the dimension is reduced by a factor*

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

Since $3(\#\mathcal{T}) = 2(\#\mathcal{E}_I) + \#\mathcal{E}_D + \#\mathcal{E}_N$ *we have*

$$F = 5 \left\{ \frac{\#\mathcal{E}_I + \frac{4}{5}(\#\mathcal{E}_D) + \frac{1}{5}(\#\mathcal{E}_N)}{\#\mathcal{E}_I + 2(\#\mathcal{E}_D) - \#\mathcal{E}_N} \right\} .$$

Under reasonable grid regularity assumptions $\#\mathcal{E}_I$ *is the dominant part of* $\#\mathcal{E}$ *as* $\#\mathcal{T} \to \infty$, *and so* $F \to 5$ *as* $\#\mathcal{T} \to \infty$.

(ii) *The coefficient matrix* $\mathring{A}$ *is a bordered matrix. The major block is the standard piecewise linear finite element stiffness matrix* A, *and the width of the border is* $n_{\mathcal{C}} - 1$ *where* $n_{\mathcal{C}}$ *is the number of disconnected components in the Neumann boundary* Γ_N. *Note that if the Neumann boundary is connected (i.e.* $n_{\mathcal{C}} = 1$*),* $\mathring{A} = A$ *and (4.3) reduces to a standard piecewise finite element system.*

For most applications the number of borders (i.e. the number of columns in C*) will be small and in particular does not increase as the mesh is refined. Thus it is reasonable to consider solving (4.3) by block elimination (as we shall do in §6). More precisely, if we write (4.3) as*

$$\begin{bmatrix} A & C \\ C^T & D \end{bmatrix} \begin{bmatrix} \mathring{\boldsymbol{\eta}} \\ \mathring{\boldsymbol{\rho}} \end{bmatrix} = \begin{bmatrix} \mathring{\boldsymbol{\varphi}} \\ \mathring{\boldsymbol{\xi}} \end{bmatrix} ,$$

first we compute a matrix X *and a vector* $\mathbf{y}$ *satisfying*

$$AX = C, \qquad A\mathbf{y} = \mathring{\boldsymbol{\varphi}} . \tag{4.26}$$

Then we solve

$$(D - C^T X)\mathring{\boldsymbol{\rho}} = (\mathring{\boldsymbol{\xi}} - C^T\mathbf{y}) , \tag{4.27}$$

from which we compute

$$\mathring{\boldsymbol{\eta}} = \mathbf{y} - X\mathring{\boldsymbol{\rho}} .$$

When C *has a small number of columns, (4.26) constitutes a small number of systems with SPD coefficient matrix* A, *which can be efficiently solved iteratively, and (4.27) is a small system to be solved for* $\mathring{\boldsymbol{\rho}}$.

4.4 Pressure computations

In this subsection we briefly indicate a procedure for the efficient handling of the decoupled system (4.6) for the pressure $\boldsymbol{p}$. This procedure is not applied in the present paper – where the velocity $\boldsymbol{q}$ is of chief interest – but it may be useful in other contents.

In the general situation described in §4.1, the assembly of (4.6) requires the computation of a complementary basis $\{\mathbf{z}_{\mathring{n}+1}, \ldots, \mathbf{z}_{n_\mathcal{V}}\}$ satisfying (4.5). This is again equivalent to finding a complementary basis $\vec{v}^c_{\mathring{n}+1}, \ldots, \vec{v}^c_{n_\mathcal{V}}$ such that

$$\mathrm{span}\{\vec{\mathring{v}}_1, \ldots, \vec{\mathring{v}}_{\mathring{n}}, \vec{v}^c_{\mathring{n}+1}, \ldots, \vec{v}^c_{n_\mathcal{V}}\} = \mathcal{V} . \tag{4.28}$$

In the context of the specific system (3.6) (using lowest-order Raviart-Thomas elements) this can be done by finding a distinguished subset of edges

$$\mathcal{E}^c \subset \mathcal{E}_I \cup \mathcal{E}_D$$

such that the corresponding subset of Raviart-Thomas basis functions, $\{\vec{v}_E : E \in \mathcal{E}^c\}$, constitutes a complementary basis. Note that this set must contain $n_{\mathcal{V}} - \mathring{n} = \#\mathcal{T}$ elements. The following simple algorithm chooses $\#\mathcal{T}$ appropriate edges, yielding a complementary basis, in such a way that the system (4.6) has a particularly simple form.

Algorithm 4.8

1. *Choose $T_1 \in \mathcal{T}$ to be any triangle with an edge $E_1 \in \mathcal{E}_D$ and set $\mathcal{E}^c = \{E_1\}$.*
2. *For $j = 2, \ldots, \#\mathcal{T}$,*
   - *choose $T_j \in \mathcal{T} \backslash \{T_\ell : \ell = 1, \ldots, j-1\}$ with the property that there exists $E_j \in \mathcal{E}_I \cup \mathcal{E}_D$ such that*
   $$E_j \subset T_j \cap \left\{ \bigcup_{\ell=1}^{j-1} T_\ell \right\}$$
   - *update $\mathcal{E}^c = \mathcal{E}^c \cup \{E_j\}$.*

   End of loop over j.
3. *Assemble $Z'B$ as*
$$(Z'B)_{i,j} = b(\vec{v}_{E_i}, w_{T_j}), \qquad i, j = 1, \ldots, \#\mathcal{T}\,.$$

Notes

(i) Step 1 is always possible since we assumed that the Dirichlet boundary is non-empty (i.e. $\mathcal{E}_D \neq \emptyset$).

(ii) It can be proved inductively that the procedure in Step 2 is always possible when Ω is a simply connected open domain.

(iii) In Step 2 the triangles in $\mathcal{T}$ are assumed to be closed, so each triangle contains its edges.

(iv) The matrix $Z'B$ in Step 3 is simply obtained from the original matrix B in (3.6) by deleting some rows and reordering the rows and columns.

In [38] it will be shown that this procedure does yield a complementary basis and that $Z'B$ given in Step 4 is lower triangular. Similar observations but in a simpler context have been made in [8].

5 Parallel Iterative Method

In this section we briefly describe our parallel solver for the velocity systems (4.26) arising in §4. Our method is based on the conjugate gradient algorithm with additive Schwarz preconditioner and uses the implementation provided by the `DOUG` package [24] 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 [23, 24].

The first step in our parallelisation involves the partition of the domain Ω (in this case using the mesh partitioning software `METIS` - [29]) into non-overlapping connected subdomains $\Omega_i, \ i = 1, \ldots, s$, 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. [7]) which has the general form

$$P^{-1} := R_H^T A_H^{-1} R_H + \sum_{i=1}^{s} R_i^T A_i^{-1} R_i \ . \tag{5.1}$$

In (5.1) 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 (5.1), $\tilde{\Omega}_i$ 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 (5.1) also contains a coarse grid solve, A_H^{-1}, which handles the global interaction of the subdomains. This distinguishes (5.1) from block-Jacobi-like methods and is essential for the construction of optimal preconditioners [14]. 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 [23]. In (5.1) 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 [23]). Thus for parallel efficiency the dimension of the coarse grid problem n_H is chosen in DOUG to satisfy

$$n_H^{3/2} = \text{ cost of solving subproblems on processors } = \left(\frac{n}{(1000 * n_p)}\right) * 1000^{3/2}. \quad (5.2)$$

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

The asymptotic performance of the preconditioner (5.1) is analysed in [7], where it is shown that for general symmetric positive definite problems

$$\kappa(P^{-1}A) = O((H/\delta)^2), \quad \text{as } H,\ h \to 0 \quad (5.3)$$

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 (5.3) then reduces to $\kappa(P^{-1}A) = O((n^{1/2}n^{-1/3})^2) = O(n^{1/3})$ 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 [20, 19], 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 ([20, 19]) to the proof that the corresponding PCG method in fact is very resilient to the existence of jumping coefficients even in the unstructured case. Roughly speaking [19] 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 logarithmically in the quantity $\max|k|/\min|k|$, whereas the condition number itself generally grows linearly in $\max|k|/\min|k|$.

The results in [20, 19] 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 [20, 19] hold true even in this extreme case, although at the time of writing we know of no proof of this fact.

6 Numerical Results

In this section we report on a number of experiments on the parallel solution of (3.1), (3.2) 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 §2) with zero mean, variance σ^2 and length scale λ.

We discretise this problem using the mixed finite element discretisation (3.5) with lowest order Raviart-Thomas elements on a uniform mesh $\mathcal{T}$ on Ω obtained by firstly 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.

As is shown in the Appendix, the mixed finite element method of §3 applied to this problem yields a velocity field which is no more than $O(h^\alpha)$ accurate (with $\alpha < 1/2$). In practice we shall replace k on each element by its constant interpolant at the centroid of the element, an approach which allows an efficient implementation and still maintains the $O(h^\alpha)$ accuracy (see Appendix). This approach only makes statistical sense when the length scale λ is of the order of the mesh diameter, equivalently

$$\lambda = C_\ell/N \tag{6.1}$$

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 (i.e. $\lambda\sim 10^{-2}$ or 10^{-3}) 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 ([31, 32, 40]).

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 (3.3) we have

$$\Gamma_D = \{0,1\} \times [0,1] \tag{6.2}$$

and

$$g(\vec{x}) \equiv 1 \ \text{ for } \vec{x} \in \{0\} \times [0,1], \quad g(\vec{x}) \equiv 0 \ \text{ for } \vec{x} \in \{1\} \times [0,1] \, . \tag{6.3}$$

We shall give results here only for the computation of the velocity $\boldsymbol{q}$ in (3.6) by solving the decoupled system (4.3) for $\mathring{\boldsymbol{q}}$. In the case of the particular boundary conditions (6.3), the computation reduces to the solution of the linear system

$$\begin{bmatrix} A & \mathbf{c} \\ \mathbf{c}^T & d \end{bmatrix} \begin{bmatrix} \mathring{\boldsymbol{\eta}} \\ \mathring{\rho} \end{bmatrix} = \begin{bmatrix} \mathring{\boldsymbol{\varphi}} \\ \mathring{\xi} \end{bmatrix} , \tag{6.4}$$

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 §4.3).

The block elimination procedure in Remark 4.7(ii) in this case requires solutions of two systems of the form

$$A\boldsymbol{u} = \mathbf{b} \, , \tag{6.5}$$

In the special case here, where the Dirichlet data g is constant on each component of Γ_D, it turns out that $\mathring{\boldsymbol{\varphi}} = \mathbf{0}$ and we only need to solve (6.5) once. The timings in §6.2 are for this task.

The sparse, SPD and highly ill-conditioned problem (6.5) is solved by the parallel iterative method described in §5. As described in §2, there are 3 parameters which determine the difficulty of (6.5): 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 (6.1), 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 wildly differing values across the domain. Moreover as C_ℓ decreases the probability of large jumps in k between neighbouring elements increases. On the other hand, to illustrate the effect of increasing σ^2, in Figure 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. This illustrates the fact that 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.

6.1 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.

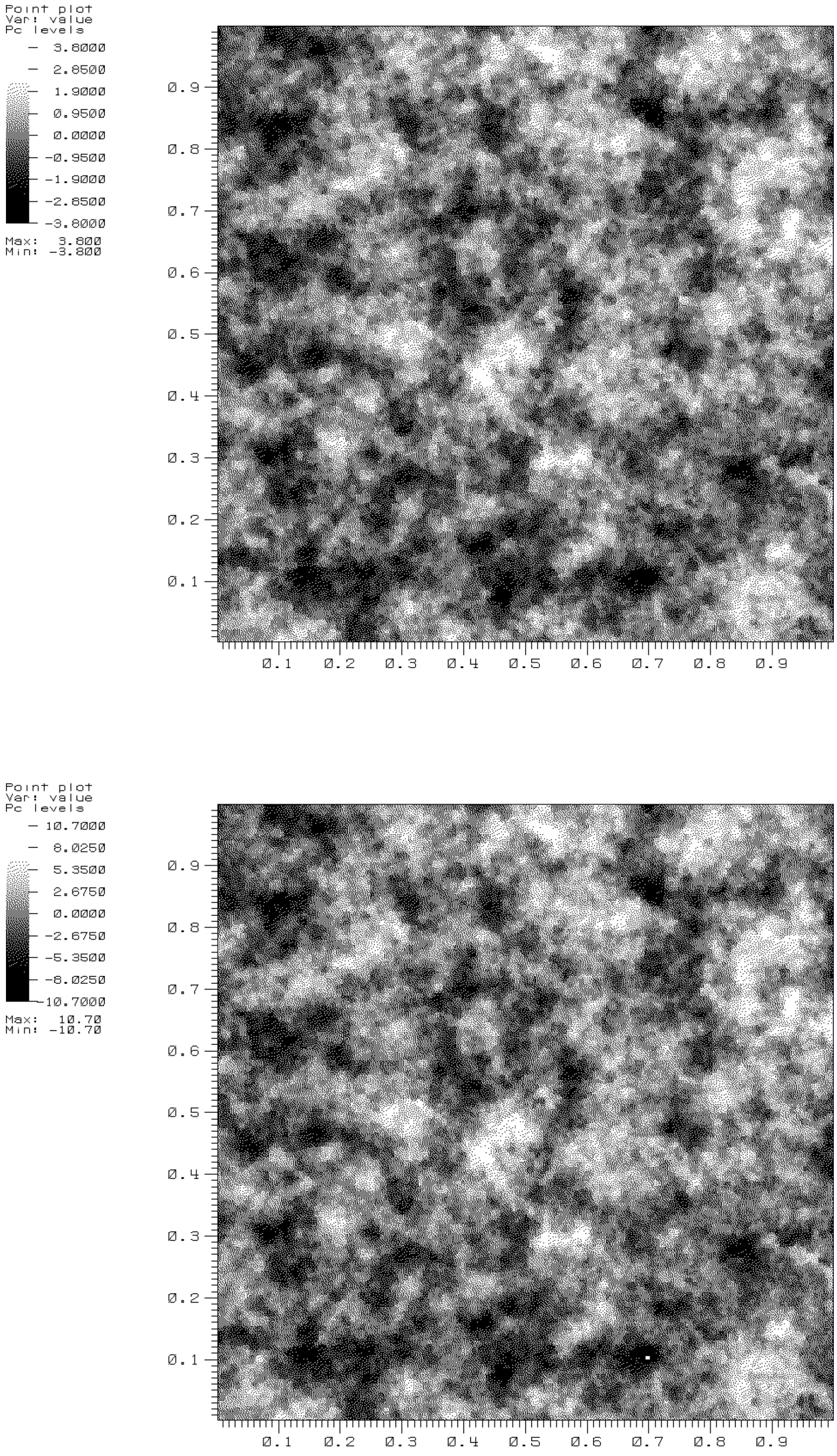


Figure 2: Greyscale plot of log(k) for $\sigma^2 = 1$ (top) and $\sigma^2 = 8$ (bottom).

The preconditioned conjugate gradient (PCG) method for (6.5) with SPD preconditioner 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* $\mathbf{z}^i = P^{-1}\mathbf{r}^i = (P^{-1}A)\mathbf{e}^i$.

Typical stopping criteria for the PCG method involve requiring that $\mathbf{z}^i$ is small in some norm. More precisely we have the standard estimate for the relative error reduction

$$\|\mathbf{e}^i\|_2/\|\mathbf{e}^0\|_2 \leq \kappa \|\mathbf{z}^i\|_2/\|\mathbf{z}^0\|_2 \ , \tag{6.6}$$

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

$$\|\mathbf{z}^i\|_2/\|\mathbf{z}^0\|_2 \leq \epsilon/\kappa \tag{6.7}$$

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

The difficulty with implementing (6.7) is the problem of computing κ. In [30] (for the unpreconditioned case $P = I$) it is proposed to estimate κ dynamically using the Lanczos procedure (for which some of the data is already computed during the CG iteration). However, even if such a procedure is adopted, the resulting stopping criterion (6.7) is often over-pessimistic due to the fact that the smallest constant κ such that (6.6) holds for all i is often very much smaller than the true condition number of $P^{-1}A$.

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*:

$$\tilde{\kappa} := \tilde{\kappa}(\sigma^2, N, \lambda) := \frac{\|\mathbf{e}^i\|_2}{\|\mathbf{e}^0\|_2} \frac{\|\mathbf{z}^0\|_2}{\|\mathbf{z}^i\|_2} \ , \tag{6.8}$$

for some specified i as the parameters σ^2, N and λ change, and our practical stopping criterion is then

$$\frac{\|\mathbf{z}^i\|_2}{\|\mathbf{z}^0\|_2} \leq \epsilon/\tilde{\kappa} \ . \tag{6.9}$$

The end result is that $\tilde{\kappa}$ is found to vary only very mildly with these parameters (see (6.10) 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 §5, 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. In this experiment we fixed $N = 32$ and $\lambda = 10/N$. In Figure 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 (6.9) 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.

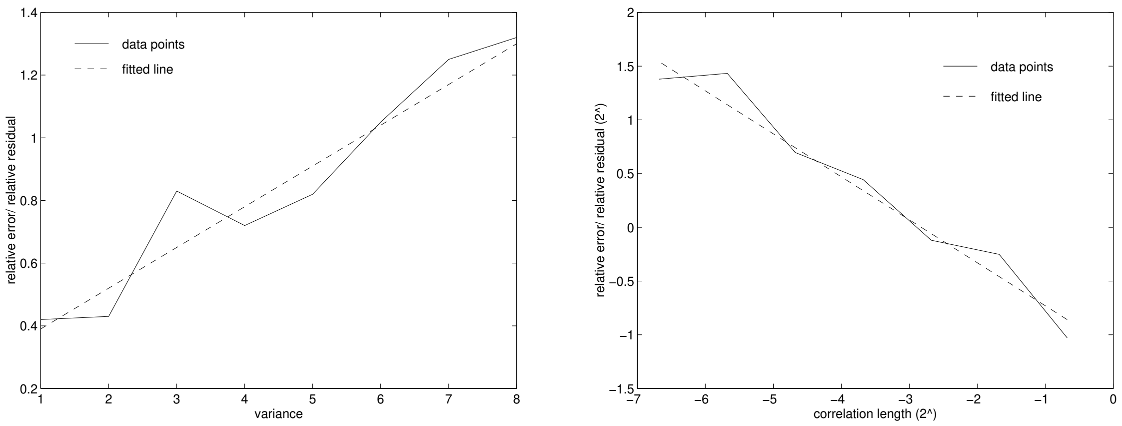


Figure 3: Variation of $\tilde{\kappa}$ with σ^2 (left), and with λ for $\sigma^2 = 4$ (right) ($N = 32$).

To study variation with respect to λ, we set $N = 32$ and $\sigma^2 = 4$ and computed $\tilde{\kappa}$ for $\lambda = 10/16, 10/32, \cdots, 10/1024$. A $\log_2 - \log_2$ plot of these results is given in Figure 3 (right) (solid line). The dotted line shows the best computed straight line fit which leads to the empirical approximation for the variation of $\tilde{\kappa}$ with λ as $\lambda^{-0.4}$.

To demonstrate the validity of this, we recomputed these experiments using stopping criterion (6.9) with $\tilde{\kappa} = (0.26 + 0.13\sigma^2)(0.46\lambda^{-0.4})$, where $\sigma^2 = 4$, $N = 32$ and $\epsilon = 10^{-4}$ and 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 (6.8) for $N = 16, 32, 64, 128$ in the case $\sigma^2 = 4$ and $\lambda = 10/16$. These experiments suggested that there is no noticeable increase in the value of $\tilde{\kappa}$ as N increases. Thus we postulate that

$$\tilde{\kappa}(\sigma^2, N, \lambda) \approx (0.26 + 0.13\sigma^2)(0.46\lambda^{-0.4}) \tag{6.10}$$

as σ^2, λ and N vary. In the experiments in the next subsection we use this formula for $\tilde{\kappa}$ in the stopping criterion (6.9).

6.2 Performance of iterative method

Our first set of results - Table 1 - illustrates the performance of the PCG method for (6.5) with additive Schwarz preconditioner (5.1), with and without coarse grid solve for various values of N and σ^2, where the length scale λ varies as in (6.1) with $C_\ell = 10$, and n denotes the number of unknowns in system (6.5). The stopping criterion was (6.9) with $\epsilon = 10^{-9}$ and $\tilde{\kappa}$ given by (6.10). The value of $\|\mathbf{z}^i\|_2/\|\mathbf{z}^0\|_2$ given is the value of this quantity when the iteration stops (where $\mathbf{z}^i$ denotes the preconditioned residual, as described above).

The first thing to note is the observed success of the strategy for computing the coarse grid as outlined in §5. Since the coarse grid is constructed just from the geometry of the fine grid, ignoring the fact that the coefficient k is varying from element to element, one may be concerned that the coarse grid may not model the underlying fine scales of the problem (at the fine grid level) 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.

N	n	σ^2	With Coarse Grid		Without Coarse Grid	
			No. It.	$\|\|\mathbf{z}^i\|\|_2/\|\|\mathbf{z}^0\|\|_2$	No. It.	$\|\|\mathbf{z}^i\|\|_2/\|\|\mathbf{z}^0\|\|_2$
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	

Table 1: Study of the iteration count ($\lambda = 10/N$).

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 2 we investigate the behaviour of the method as N grows.

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

Table 2: Number of iterations as N (and therefore n) increases ($\lambda = 10/N$, $\sigma^2 = 2$).

We know from the discussion in §5 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. Interestingly the results in

Table 2 indicate that exactly the same growth is observed in this case, *even though* in this case the coefficient is extremely rough.

In Table 3 we illustrate how the iteration numbers are affected by growth in σ^2 for $N = 256$ and $\lambda = 10/N$.

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

Table 3: Number of iterations as σ^2 increases ($N = 256$, $\lambda = 10/N$).

We see from Table 3 that the rate of growth of the number of PCG iterations is 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 (6.5) 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 [20, 19] (see §5) for the special case when the number of regions in which the coefficient has a constant value 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 as predicted in [20, 19]. It remains an open question to give a proof of this observation.

Recall that for a physically realistic model we assume (see (6.1)) that the length scale λ decreases linearly in $1/N$. In the previous Tables 1 - 3 we took $C_\ell = 10$ in (6.1). In Table 4 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.

	$C_\ell = 5$		$C_\ell = 20$	
σ^2	No. It.	$\|\|\mathbf{z}^i\|\|_2/\|\|\mathbf{z}^0\|\|_2$	No. It.	$\|\|\mathbf{z}^i\|\|_2/\|\|\mathbf{z}^0\|\|_2$
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 4: Affect of C_ℓ on the iteration count ($N = 512$, with coarse grid).

In groundwater flow calculations in practice it is often necessary to study flows in long thin regions. In Table 5 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 aspect ratio $L \geq 1$. 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 nodes in the mesh is fixed at $n = (N+1)*(N-1)$, for $N = 256$. For these experiments $\sigma^2 = 4$ and $\lambda = 10/256$. The iteration count, as L increases, is given in Table 5. A very modest growth with L is observed.

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}

Table 5: Effect of aspect ratio L on iteration count (with coarse grid).

Our final table - Table 6 - illustrates the parallel efficiency of the algorithm. To understand these results, recall that the `DOUG` solver described in §5 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 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 [23]). In Table 6 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, UK (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 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

	Without coarse grid		With coarse grid	
No. Slaves	Time (sec)	Efficiency	Time (sec)	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%

Table 6: Parallel Efficiency on SP2 ($\sigma^2 = 2$, $N = 256$, $\lambda = 1/N$).

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 6 give an accurate impression of the parallel efficiency of the algorithm.

In Table 5 note especially the improved parallel efficiency of the method with the coarse grid compared to that without. This indicates the success of the the parallelisation 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` [23, 24].

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 [23]).

Finally, in Figure 4 we plot the velocity fields corresponding to the problem (3.1), (3.2) with boundary conditions (6.2) 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.

Appendix: Accuracy considerations

As we mentioned at the end of §3, in the practical implementation of (3.6) (equivalently (3.5)), using lowest-order Raviart-Thomas elements, it turns out to be sufficient in terms of accuracy to replace the bilinear form m with the approximation

$$\widetilde{m}(\vec{u}, \vec{v}) := ((\mu/\tilde{k})\vec{u}, \vec{v})_{(L_2(\Omega))^2}, \tag{A.1}$$

where $\tilde{k}$ is the piecewise constant approximation of k obtained by interpolation at the centroids of the triangles T of the mesh $\mathcal{T}$. In this appendix we explain why this is the case.

First observe that, by substituting equation (3.1) into (3.2), we obtain a second order elliptic PDE

$$-\vec{\nabla} \cdot (k/\mu)\vec{\nabla} p = 0 \tag{A.2}$$

to be solved for p subject to mixed boundary conditions. Recalling that μ is constant, and that $\log k$ is a realisation of a Gaussian random field, Theorem 2.1 implies that (with probability 1) $k/\mu \in \mathcal{C}^\alpha(\Omega)$, the space of Lipschitz continuous functions with exponent $\alpha < 1/2$. The solution p has in general no better regularity than $\mathcal{C}^{1,\alpha}(\Omega)$, i.e. the first derivatives of p are in $\mathcal{C}^\alpha(\Omega)$ (see [18, Thm.8.34]), and so the velocity $\vec{q}$ in (3.1) in general has no better regularity than $(C^\alpha(\Omega))^2$. From now on we assume that $\vec{q} \in (C^\alpha(\Omega))^2$.

Next recall (see e.g. [3]) that solving (3.4) is equivalent to seeking $\vec{q} \in \mathring{H}_{0,N}(\mathrm{div}, \Omega)$ such that

$$m(\vec{q}, \vec{v}) = G(\vec{v}), \qquad \text{for all } \vec{v} \in \mathring{H}_{0,N}(\mathrm{div}, \Omega), \tag{A.3}$$

where $\mathring{H}_{0,N}(\mathrm{div}, \Omega) := \{\vec{v} \in H_{0,N}(\mathrm{div}, \Omega) : b(\vec{v}, w) = 0 \text{ for all } w \in L_2(\Omega)\}$. Moreover, setting $\mathring{\mathcal{V}} := \{\vec{V} \in \mathcal{V} : b(\vec{V}, W) = 0 \text{ for all } W \in \mathcal{W}\}$, as in §4, it follows that (3.5) is equivalent to seeking $\vec{Q} \in \mathring{\mathcal{V}}$ such that

$$m(\vec{Q}, \vec{V}) = G(\vec{V}), \qquad \vec{V} \in \mathring{\mathcal{V}}. \tag{A.4}$$

Since

$$\|\vec{v}\|_{H(\mathrm{div},\Omega)} = \|\vec{v}\|_{L_2(\Omega)} \tag{A.5}$$

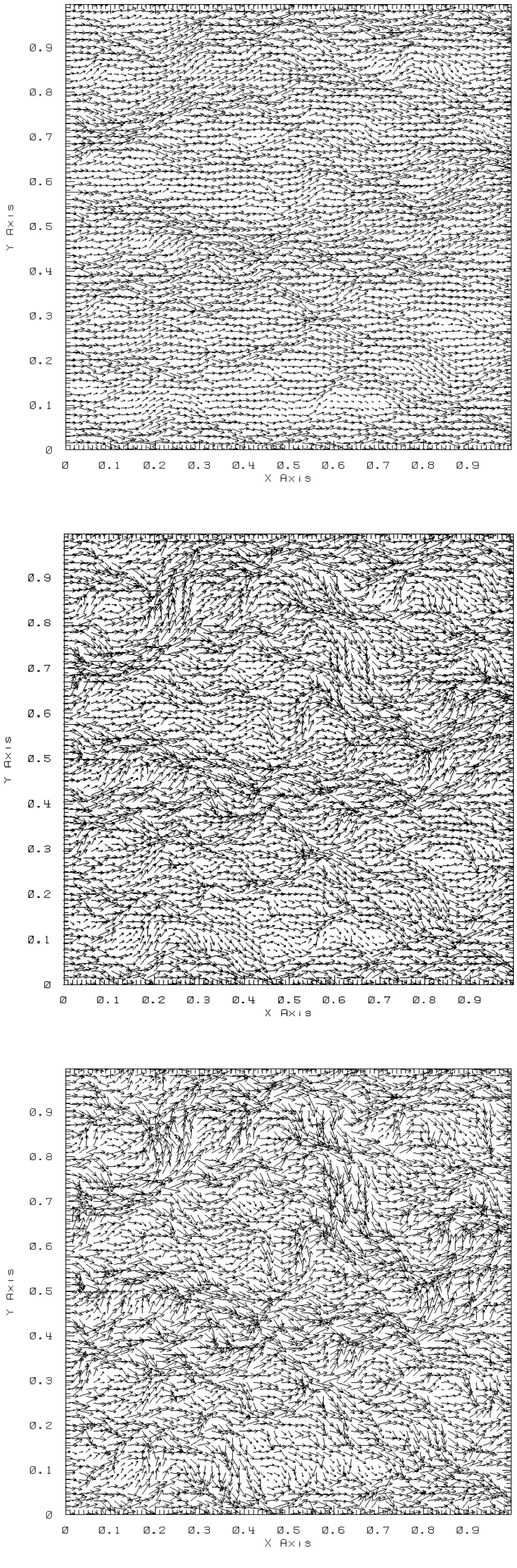


Figure 4: Vector plot of the velocity for $\sigma^2 = 1$, 4, and 8 ($N = 256$, $\lambda = 10/N$).

when $\vec{v} \in \mathring{H}_{0,N}(\text{div}, \Omega)$, it can easily be shown that m is continuous and elliptic on $\mathring{H}_{0,N}(\text{div}, \Omega)$ which ensures the well-posedness of (A.3) and the stability of the approximation scheme (A.4).

Now suppose that m in (A.4) is replaced by $\widetilde{m}$ defined in (A.1) and, for simplicity, denote the resulting solution again by $\vec{Q}$. Then, from the First Strang Lemma [9, Thm.4.1.1], we obtain the estimate:

$$\begin{aligned} \|\vec{q} - \vec{Q}\|_{H(\text{div},\Omega)} &\leq \inf_{\vec{V} \in \mathring{\mathcal{V}}} \left\{ \|\vec{q} - \vec{V}\|_{H(\text{div},\Omega)} + \sup_{\vec{V}' \in \mathring{\mathcal{V}}} \frac{|m(\vec{V}, \vec{V}') - \tilde{m}(\vec{V}, \vec{V}')|}{\|\vec{V}'\|_{H(\text{div},\Omega)}} \right\} \\ &\leq \inf_{\vec{V} \in \mathring{\mathcal{V}}} \left\{ \|\vec{q} - \vec{V}\|_{L_2(\Omega)} + C\|k - \tilde{k}\|_{\infty,\Omega} \|\vec{V}\|_{L_2(\Omega)} \right\} \qquad \text{(A.6)} \end{aligned}$$

where C depends on μ and $\min|k|$ and we have again used (A.5). Now, because of the regularity of k we have $\|k - \tilde{k}\|_{\infty,\Omega} = O(h^\alpha)$ with $\alpha < 1/2$ where h is the maximum diameter of the triangles in $\mathcal{T}$. In Lemma A.1 below we prove that there exists a choice of $\vec{V} \in \mathring{\mathcal{V}}$ with $\|\vec{V}\|_{L_2(\Omega)}$ bounded and $\|\vec{q} - \vec{V}\|_{L_2(\Omega)} \leq O(h^\alpha)$, as $h \to 0$. Because of the reduced regularity we know of no better estimate than this, and so we conclude that the perturbation term $\|k - \tilde{k}\|_{\infty,\Omega}$ in (A.6) has no effect on the overall convergence estimate for the method. To finish the argument we state and prove the required lemma.

Lemma A.1 *Suppose that as $h \to 0$ the sequence of meshes on Ω (parametrised by h) is non-degenerate (see, e.g. [3, p.106]). Then for each h, there exists $\vec{V} \in \mathring{\mathcal{V}}$ such that*

$$\|\vec{q} - \vec{V}\|_{L_2(\Omega)} \leq Ch^\alpha \|\vec{q}\|_{C^\alpha(\Omega)} \qquad \text{(A.7)}$$

$$\text{and} \qquad \|\vec{V}\|_{L_2(\Omega)} \leq C' \|\vec{q}\|_{\infty,\Omega} \qquad \text{(A.8)}$$

where $\|\vec{q}\|_{C^\alpha(\Omega)} = \max_{k=1,2} \|q_k\|_{C^\alpha(\Omega)}$, $\|\vec{q}\|_{\infty,\Omega} = \max_{k=1,2} \|q_k\|_{\infty,\Omega}$, and C, C' are constants independent of h.

Proof. Since in most convergence theories (e.g. [4]) one is dealing with functions $\vec{q}$ which are regular in appropriate Sobolev spaces, estimates like (A.7), (A.8) are not so commonly found in the literature. Nevertheless we can obtain them by employing the projection operator

$$\Pi_h : H_{0,N}(\text{div}, \Omega) \to \mathcal{V} \qquad \text{(A.9)}$$

defined in [4, §III.3.4]. This has the property that it maps $\mathring{H}_{0,N}(\text{div}, \Omega)$ into $\mathring{\mathcal{V}}$. Moreover, for any $\vec{v} \in \mathring{H}_{0,N}(\text{div}, \Omega)$ we have, on each $T \in \mathcal{T}$, $(\Pi_h \vec{v})|_T = \rho_T \vec{v}$ where $\rho_T \vec{v}$ is locally determined on T by the equations

$$(\rho_T \vec{v}) \cdot \vec{\nu}_E = \frac{1}{|E|} \int_E \vec{v} \cdot \vec{\nu}_E, \qquad \text{(A.10)}$$

with E ranging over the three edges of T. From this we see that $\rho_T \vec{v} = \vec{v}$ when $\vec{v}$ is a constant vector on T. Now consider $\vec{V} = \Pi_h \vec{q}$ as a candidate for the approximation to $\vec{q}$ in (A.7), and let C denote a generic constant independent of h. Then

$$\begin{aligned} \|\vec{q} - \vec{V}\|_{L_2(\Omega)} &= \left\{ \sum_{T \in \mathcal{T}} \|\vec{q} - \rho_T \vec{q}\|^2_{L_2(T)} \right\}^{1/2} \\ &\leq C \left\{ \sum_{T \in \mathcal{T}} \left(\|\vec{q} - \vec{q}_T\|^2_{L_2(T)} + \|\rho_T(\vec{q} - \vec{q}_T)\|^2_{L_2(T)} \right) \right\}^{1/2} \qquad \text{(A.11)} \end{aligned}$$

where $\vec{q}_T$ denotes the constant vector on T which coincides with $\vec{q}$ at the centroid of T.

We show below that for each T and each $\vec{v} \in (L_\infty(T))^2$ with $\vec{\nabla} \cdot \vec{v} \equiv 0$ on T we have

$$\|\rho_T \vec{v}\|_{L_2(T)} \leq C|T|^{1/2}\|\vec{v}\|_{\infty,T}. \tag{A.12}$$

Assuming (A.12) for the moment and using it in (A.11), we obtain

$$\|\vec{q} - \vec{V}\|_{L_2(\Omega)} \leq C\left\{\sum_{T\in\mathcal{T}} |T|\|\vec{q} - \vec{q}_T\|^2_{\infty,T}\right\}^{1/2} \leq C\left\{\sum_{T\in\mathcal{T}} |T|\right\}^{1/2} h^\alpha \|\vec{q}\|_{C^\alpha(\Omega)}$$

where, in the last step, we have used the regularity of $\vec{q}$. This yields (A.7) whereas (A.8) follows easily from writing

$$\|\vec{V}\|_{L_2(\Omega)} = \left\{\sum_{T\in\mathcal{T}} \|\rho_T \vec{q}\|^2_{L_2(T)}\right\}^{1/2}$$

and then using (A.12).

To complete the proof we now obtain (A.12). Suppose $\vec{\nabla} \cdot \vec{v} \equiv 0$ on T. Then by [4, Remark 3.3], we have $\vec{\nabla} \cdot \rho_T \vec{v} \equiv 0$ on T, i.e. $\rho_T \vec{v}$ is constant on T. Thus, if E_1 and E_2 are two edges of T we can write

$$\rho_T \vec{v} = \alpha_1 \vec{\nu}_{E_1} + \alpha_2 \vec{\nu}_{E_2}$$

for some $\boldsymbol{\alpha} := (\alpha_1, \alpha_2)$, and (A.10) is equivalent to the 2×2 linear system $M\boldsymbol{\alpha} = \mathbf{b}$, where for $i, j = 1, 2$, $M_{ij} := \vec{\nu}_{E_i} \cdot \vec{\nu}_{E_j}$, $b_i := \frac{1}{|E_i|}\int_{E_i} \vec{v} \cdot \vec{\nu}_{E_i}$. Hence

$$\|\rho_T \vec{v}\|^2_{L_2(T)} = |T|\ \boldsymbol{\alpha}^T M \boldsymbol{\alpha} = |T|\ \mathbf{b}^T M^{-1} \mathbf{b}$$

When the mesh sequence is non-degenerate M^{-1} is bounded as $h \to 0$ and so

$$\|\rho_T \vec{v}\|^2_{L_2(T)} \leq C|T|\mathbf{b}^T\mathbf{b} \leq C|T| \max_{i=1,2} |b_i|^2. \tag{A.13}$$

But

$$|b_i| \leq \frac{1}{|E_i|}\int_{E_i} |\vec{v}| \leq \max_{\vec{x}\in E_i} |\vec{v}(\vec{x})| \leq C\|\vec{v}\|_{\infty,T},$$

and combining this with (A.13) yields (A.12). □

References

[1] R.J. Adler, *The Geometry of Random Fields* (John 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* **1**, *Portland, OR, 1995* (1995), p. 405.

[3] S.C. Brenner and L.R. Scott, *The Mathematical Theory of Finite Element Methods* (Springer, Berlin, 1994).

[4] F. Brezzi and M. Fortin, *Mixed and hybrid finite element methods* (Springer, New York, 1991).

[5] Z. Cai, R.R. Parashkevov, T.F. Russell, and X. Ye, Domain decomposition for a mixed finite element method in three dimensions, to appear in *SIAM J Num. Anal.* (1998).

[6] T.F. Chan and T. Mathew, Domain Decomposition Methods, *Acta Numerica* 1994 (Cambridge University Press, 1994).

[7] T.F. Chan, B.F. Smith, and J. Zou, Overlapping Schwarz methods on unstructured meshes using non-matching course grids, *Numer. Math.* **73** (1996) p. 149.

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

[9] P.G. Ciarlet, *The Finite Element Method for Elliptic Problems* (North Holland, Amsterdam, 1978).

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

[11] M. Crouzeix, In *Proc. Journées éléments finis* (Université de Rennes, 1976).

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

[13] C.R. Dietrich and G.N. Newsam, A Fast and Exact Method for Multidimensional Gaussian Stochastic Simulations, *Water Resour. Res.* **29**, 8 (1993) p. 2861.

[14] M. Dryja and O.B. Widlund, Some domain decomposition methods for for elliptic problems, in *Iterative Methods for Large Linear Systems*, edited by Hayes and Kincaid, (Academic Press, San Diego, 1989) p. 273.

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

[16] R.E. Ewing and J. Wang, Analysis of multilevel decomposition iterative methods for mixed finite element methods, *RAIRO Model. Math. Anal. Num.* **28**, 4 (1994), p. 377.

[17] L.W. Gelhar, A Stochastic Conceptual Analysis of One-Dimensional Groundwater Flow in Nonuniform Homogeneous Media, *Water Resour. Res.* **11** (1975), p. 725.

[18] D. Gilborg and N.S. Trudinger, *Elliptic Partial Differential Equations of Second Order* (Springer, Berlin, 1983).

[19] I.G. Graham and M.J. Hagger, Unstructured additive Schwarz-CG method for elliptic problems with highly discontinuous coefficients, *SIAM J Sci. Comp.* **20** (1999), p. 2041.

[20] I.G. Graham and M.J. Hagger, Additive Schwarz, CG and discontinuous coefficients, in *Proc. Ninth Intern. Confer. on Domain Decomposition Methods, Bergen, 1996*, edited by Bjorstad, Espedal and Keyes (Domain Decomposition Press, Bergen, 1998).

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

[22] A.L. Gutjahr, D. McKay, and J.L. Wilson, Fast Fourier Transform Methods for Random Field Generation, *Eos Trans. AGU* **68**, 44 (1987) p. 1265.

[23] M.J. Hagger, Automatic domain decomposition on unstructured grids (`DOUG`), *Advances in Computational Mathematics* **9** (1998), p. 281.

[24] 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`.

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

[26] F. Hecht, Construction d'une base pour des éléments finis mixtes à divergence faiblement nulle, Unpublished report (1988).

[27] R. Hiptmair, T. Schiekofer, and B. Wohlmuth, Multilevel preconditioned Augmented Lagrangian techniques for 2^{nd} order mixed problems, *Computing* **57** (1996), p.25.

[28] R.J. Hoeksema and P.K. Kitanidis, Analysis of the Spatial Structure of Properties of Selected Aquifers, *Water Resour. Res.* **21** (1985), p. 825.

[29] G. Karypis and V. Kumar, A fast and high quality multilevel scheme for partitioning irregular graphs, *SIAM J Sci. Comp.* **20** (1999), p. 359.

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

[31] A. Mantoglou and J.L. Wilson, The Turning Bands Method for Simulation of Random Fields Using Line Generation by a Spectral Method, *Water Resour. Res.* **18**, 5 (1982), p. 1379.

[32] A. Mantoglou, Digital Simulation of Multivariate Two- and Three-dimensional Stochastic Processes with a Spectral Turning Bands Method *Math. Geol.* **19**, 2 (1987), p. 129.

[33] T.P. Mathew, Schwarz alternating and iterative refinement methods for mixed formulations of elliptic problems, part 1: algorithms and numerical results, *Numer. Math.* **65** (1993), p. 445.

[34] T.P. Mathew, Schwarz alternating and iterative refinement methods for mixed formulations of elliptic problems, part 2: convergence theory, *Numer. Math.* **65** (1993), p. 469.

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

[36] 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 Galligani and Magenes, Lect. Notes in Math. (Springer, New York, 1977), Vol. 606, p. 292.

[37] 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** (1993), p. 2395.

[38] R. Scheichl, Ph.D. Thesis, in preparation (University of Bath, 2000).

[39] F. Thomasset, *Implementation of finite element methods for Navier-Stokes equations* (Springer, New York, 1981).

[40] 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 (1989), p. 2227.

[41] C. Wagner, W. Kinzelbach, and G. Wittum, Schur-Complement Multigrid, a robust method for groundwater flow and transport problems, *Numer. Math.* **75** (1997), p. 523.