RAL-TR-2001-006

A null space algorithm for mixed finite element approximation of Darcy's equation

Mario Arioli[1] and Gianmarco Manzini[2]

ABSTRACT
A null space algorithm is considered to solve the augmented system produced by the mixed finite element approximation of Darcy's Law. The method is based on the combination of an orthogonal factorisation technique for sparse matrices with an iterative Krylov solver. The computational efficiency of the method relies on a suitable stopping criterion for the iterative solver. We experimentally investigate its performance on a realistic set of selected application problems.

Keywords: Augmented systems, sparse matrices, mixed finite elements.

AMS(MOS) subject classifications: 65F05, 65F50.

Current reports available by anonymous ftp to ftp.numerical.rl.ac.uk in directory pub/reports.
[1] M.Arioli@rl.ac.uk, Rutherford Appleton Laboratory,
[2] marco@ian.pv.cnr.it, IAN - CNR, via Ferrata 1, 27100 Pavia, Italy

Computational Science and Engineering Department
Atlas Centre
Rutherford Appleton Laboratory
Oxon OX11 0QX

January 4, 2001

Contents

1 INTRODUCTION **1**

2 PROBLEM FORMULATION **1**

3 NUMERICAL ALGORITHM **2**

4 STOPPING CRITERION **3**

5 NUMERICAL EXPERIMENTS **4**

5.1 Test Problems 5

5.2 Numerical Results 5

6 CONCLUSIONS **6**

1 INTRODUCTION

The Mixed Finite Element approximation of Darcy's law produces a finite dimensional problem, which is defined by an augmented system. In order to solve the latter, we present in this paper a null space method which combines a direct Householder factorisation solver and a conjugate gradient iterative one. The properties of this method have been studied in Arioli (2000*b*) where its backward stability when using finite-precision arithmetic is proved, and where a review of the bibliography on the topic is also presented.

The efficiency of this kind of approach is strongly affected by the criterion used to stop the iterative solver. The one adopted in this paper is based on an estimate of the approximation error in the energy norm of the problem. See Arioli (2000*a*) for a more detailed theoretical presentation.

We remark that this method can be applied to any diffusion equation. For the sake of simplicity, we will focus on Darcy's Law, which is a significant example among saddle point problems.

The performancs of the final algorithm is experimentally investigated on a representative set of problems.

In the presentation of the algorithm, we will denote by E_1 and E_2 the $n \times m$ ($m \leq n$) matrix

$$E_1 = \begin{bmatrix} I_m \\ 0_{n-m,m} \end{bmatrix} \tag{1}$$

and the $n \times (n-m)$ matrix

$$E_2 = \begin{bmatrix} 0_{n-m,m} \\ I_{n-m} \end{bmatrix}. \tag{2}$$

2 PROBLEM FORMULATION

Let Ω be a simply connected, bounded, polygonal domain in $\mathbb{R}^2$, defined by a closed curve Γ. Γ is usually the union of two parts Γ_D and Γ_N, where different Dirichlet and Neumann type boundary conditions are imposed, that is $\Gamma = \Gamma_D \cup \Gamma_N$.

Darcy's laws can be formulated as follows:

$$\begin{cases} \mathbf{u}(\mathbf{x}) &= -K(\mathbf{x})\mathrm{grad}\, p(\mathbf{x}), & \mathbf{x} \in \Omega \\ \mathrm{div}\, \mathbf{u}(\mathbf{x}) &= f(\mathbf{x}), & \mathbf{x} \in \Omega \end{cases} \tag{3}$$

with a set of boundary conditions for both $\mathbf{u}$ and p:

$$\begin{cases} p(\mathbf{x}) &= g_D(\mathbf{x}), & \mathbf{x} \in \Gamma_D \\ \mathbf{u} \cdot \mathbf{n} &= g_N(\mathbf{x}), & \mathbf{x} \in \Gamma_N \end{cases} \tag{4}$$

using two regular functions g_D and g_N for Dirichlet and Neumann conditions, and where $\mathbf{n}$ denotes the external normal to Γ. In the following, we will assume that $g_N = 0$.

Darcy's law describes the relationship between the pressure $p(\mathbf{x})$ (the total head) and the velocity field $\mathbf{u}(\mathbf{x})$ (the visible effect) in groundwater flow. In system (3) $K(\mathbf{x})$ is the hydraulic conductivity tensor and $f(\mathbf{x})$ is a source-sink term.

The former equation in (3) relates the vector field $\mathbf{u}$ to the scalar field p via the permeability tensor K, which accounts for the soil characteristics. The latter equation in (3) relates the divergence of $\mathbf{u}$ to the source-sink term $f(\mathbf{x})$.

Let $\mathcal{T}_h$ be a family of triangulations of Ω, i.e. each $\mathcal{T}_h$ is a set of disjoint triangles τ which cover Ω in such a way that no vertex of any triangle lies in the interior of an edge of another triangle. Let $h = \max\{\text{diam}\,(\tau) \,:\, \tau \in \mathcal{T}_h\}$. We assume that $\mathcal{T}_h$ is regular in the sense of Ciarlet (1978), i.e. triangles do not degenerate as $h \to 0$. Moreover, we assume that no triangle has one vertex on g_D and another vertex on g_N, and that each triangle cannot have more than one edge lying on Γ.

The mixed finite element approximation of system (3) with the usual Raviart-Thomas space (we refer to Brezzi and Fortin (1992) for a detailed analysis), leads to the solution of the following system of linear equations:

$$\begin{bmatrix} M & A \\ A^T & 0 \end{bmatrix} \begin{bmatrix} u \\ p \end{bmatrix} = \begin{bmatrix} q \\ b \end{bmatrix}, \tag{5}$$

where, denoting by n the number of edges and by m the number of triangles, we have that $M \in \mathbb{R}^{n\times n}$ is a symmetric and positive definite matrix, and that $A \in \mathbb{R}^{n\times m}$ is a sub matrix of a totally unimodular matrix with $m+1$ columns. Therefore, A is full rank and its entries are equal to either 1, -1, or 0.

The augmented system (5) is nonsingular because $\text{Ker}\left(A^T\right) \cap \text{Ker}(M) = 0$.

3 NUMERICAL ALGORITHM

In this section, we illustrate the classical null space algorithm, which is described in Gill, Murray and Wright (1981), for the minimisation of linearly constrained quadratic forms. Let $Y \in \mathbb{R}^{n\times m}$ and $Z \in \mathbb{R}^{n\times(n-m)}$ be two matrices such that

$$Y^T A = I_m \qquad \text{and} \qquad Z^T A = 0_{n-m,m}. \tag{6}$$

The algorithm can be formulated as follows:

Null Space Algorithm:

$$\begin{aligned} &1. \quad u_0 = Yb, \\ &2. \quad Z^T M Z\, w = Z^T q - Z^T M u_0 = s, \\ &3. \quad u = u_0 + Zw, \\ &4. \quad p = Y^T q - Y^T M u. \end{aligned} \tag{7}$$

The matrices Y and Z can be computed using either an orthogonal factorisation or a Gaussian factorisation of the matrix A. In Amestoy, Duff and Puglisi (1996), it is shown that, by a sparse

version of the Householder algorithm, the matrix A can be factored as follows:

$$A = H \begin{bmatrix} R \\ 0 \end{bmatrix} \tag{8}$$

where R is an $m \times m$ sparse, non singular, upper triangular matrix and H is an $n \times n$ orthonormal matrix. The matrix H can be stored implicitly as the set of sparse vectors that generate the elementary Householder transformations, see Amestoy et al. (1996) for details on the sparsity of R and on the sparse storage of H. With respect to the previous choice, we have

$$Y = H E_1 R^{-T} \qquad \text{and} \qquad Z = H E_2. \tag{9}$$

The matrix Z is an orthonormal basis of the kernel of A^T. It is very important to observe that we never need to explicitly compute either Y or Z. Indeed the Householder factorisation gives the possibility of using its sparse result for implicitly computing all the matrix-vector products required by the algorithm. We can perform the product of the projected Hessian matrix $Z^T M Z$ by a vector and the product of Y or Y^T by a vector in the following ways:

$$\begin{aligned} Z^T M Z\, w &= E_2^T\,(H^T\,(M\,(H\,(E_2 w)\,)\,)\,), \\ Y^T y &= R^{-1}\,(E_1^T\,(H^T y)\,), \\ Y x &= H\,(E_1\,(R^{-T} x)\,). \end{aligned} \tag{10}$$

This approach has the advantage of performing backward and forward substitutions for triangular matrices, and of using the sparse Householder elementary matrices to perform matrix-vector products.

In Step 2 of the Null Space algorithm, we also need to solve a system involving $Z^T M Z$, the projected Hessian matrix. We have two alternative ways to proceed. If $n - m$ is small (the number of constraints is very close to the number of unknowns), or the projected Hessian matrix is still sparse, we can explicitly compute $Z^T M Z$, and then solve the system $Z^T M Z\, w = s$ using the Cholesky factorisation. Otherwise, when the product $Z^T M Z$ cannot be performed directly because both the complexity would be too high – $\mathcal{O}(n^3)$ – or the resulting matrix would be fairly dense, despite the sparsity of M. we can solve the linear system $Z^T M Z\, w = s$ using a conjugate gradient algorithm, and implicitly computing the matrix by vector products.

4 STOPPING CRITERION

If we use the conjugate gradient method, it is quite natural to have a stopping criterion which takes advantage of the minimisation property of this method. At each step j the conjugate gradient minimises the energy norm of the error $\delta w = w - w^{(j)}$ on a Krylov space. The space $\mathbb{R}^{n-m}$ with the norm

$$||\, y \,||_{Z^T M Z} = \left(y^T Z^T M Z y \right)^{\frac{1}{2}} \tag{11}$$

induces on its dual space the dual norm

$$||\, f \,||_{(Z^T M Z)^{-1}} = \left(f^T \left(Z^T M Z \right)^{-1} f \right)^{\frac{1}{2}} \tag{12}$$

A stopping criterion such as

$$\textbf{if } \| Z^T M Z\, w^{(j)} - s \|_{(Z^T M Z)^{-1}} \leq \eta \| s \|_{(Z^T M Z)^{-1}} \textbf{ then STOP}, \tag{13}$$

will guarantee that the computed solution $w^{(j)}$ satisfies the perturbed linear system

$$\begin{aligned} & Z^T M Z\, w^{(j)} = s + f, \\ & \| f \|_{(Z^T M Z)^{-1}} \leq \eta \| s \|_{(Z^T M Z)^{-1}}. \end{aligned} \tag{14}$$

The choice of η will depend on the properties of the problem that we want to solve, and, in the practical cases $\eta >> \epsilon$, where ϵ is the rounding unit. Therefore, it is appropriate to analyse the influence of the perturbations on the error between the exact solutions u and p and the computed u^* and p^* neglecting the part depending on ϵ. Using the results of Arioli (2000*b*), Arioli and Baldini (1999), we can prove that

$$\begin{aligned} \| u - u^* \|_M & \leq \eta \| u - u_0 \|_M \\ \| p - p^* \|_{A^T M^{-1} A} & \leq \eta \sqrt{\zeta} \| u - u_0 \|_M \end{aligned} \tag{15}$$

where ζ is the spectral radius of $Z^T M Z E_2 M^{-1} E_2$.

Furthermore, we need to add some tool within the conjugate gradient algorithm for estimating the values in (13). The estimate ξ_j of $\| Z^T M Z\, w^{(j)} - s \|_{(Z^T M Z)^{-1}}$ can be computed using a Gauss quadrature rule as proposed in Golub and Meurant (1997) and tested in (Arioli 2000*a*, Meurant 1997). At step j of the conjugate gradient, ξ_j estimates the true value of the error at step $j - d$, where d is an a priori selected integer value.

Finally, we must estimate $\| s \|_{(Z^T M Z)^{-1}}$. Taking into account that

$$\| s \|_{(Z^T M Z)^{-1}} = \| u \|_{Z^T M Z},$$

we could replace $\| s \|_{(Z^T M Z)^{-1}}$ with $\| w^{(j)} \|_{Z^T M Z}$ at the step j of the conjugate gradient if the current estimate ξ_j is less than or equal to $\eta \| s \|_2$. Therefore, we can only use (13) after an additional check:

$$\begin{aligned} & \textbf{if} \quad \xi_j \leq \eta \| s \|_2 \quad \textbf{then} \\ & \qquad \textbf{if} \quad \xi_j \leq \eta \| w^{(j)} \|_{Z^T M Z} \quad \textbf{then STOP} \\ & \textbf{endif} \ . \end{aligned} \tag{16}$$

Moreover, using (16) we can avoid too many additional matrix-vector products, and the additional floating-point operations needed are negligible in comparison with the total number of floating point operations performed by the conjugate gradient.

5 NUMERICAL EXPERIMENTS

All our experiments were performed on a SUN workstation using the sparse code **MA49** of the HSL (HSL 2000), described in Amestoy et al. (1996). In **MA49** the Householder factorisation is implemented using a multifrontal approach for sparse matrices: in the symbolic part, the

code computes the column ordering of A following the minimum degree reordering of $A^T A$ in order to minimise the fill-in in the upper triangular matrix R, and then a row ordering making A ordered by the leading entries.

In our experiments, we compare the exact solution u and p of (5) with the values u^* and p^* computed by the null space algorithm where, in step 2, we used the conjugate gradient method with (16), and we chose $w^{(0)} = 0$. We assume that the values computed by the HSL routine **MA47** (Duff and Reid 1983) which implements a sparse Gaussian factorisation applied to (5) are exact. We also report on the performances of the algorithm when a symmetric scaling is performed on the problem. The entries of the diagonal scaling matrix D are equal to $(\sqrt{M_{ii}})_{i=1,\ldots,n}$ and the system (5) is scaled as follows:

$$\begin{bmatrix} D^{-1} & 0 \\ 0 & I_m \end{bmatrix} \begin{bmatrix} M & A \\ A^T & 0 \end{bmatrix} \begin{bmatrix} D^{-1} & 0 \\ 0 & I_m \end{bmatrix} \begin{bmatrix} Du \\ p \end{bmatrix} = \begin{bmatrix} D^{-1}q \\ b \end{bmatrix}. \tag{17}$$

5.1 Test Problems

In Figure 1, we illustrate the geometry of the domain and the boundary conditions. We generated three regular meshes on Ω with an increasing number n of triangles (see Table 1). For each mesh, we assembled M, A, q, and b corresponding to $K_0 = 1$ and to four values of K_1 (see Figure 1): $10^{-2}, 10^{-4}, 10^{-6}, 10^{-8}$.

In Table 1, we report on the values n, m (number of degrees of freedom of the problem), and the values of the number of nonzero entries in M and A. Moreover, in Table 1, we report on the values of the storage (measured in words) needed for H and R, and on the values of η.

5.2 Numerical Results

In Table 2 and in Table 3, we summarise the numerical performance of the algorithm for each test problem and each mesh respectively without and with scaling. In Fig 3, 4, and 5, we plot the convergence history of $||Z^T M Z\, w^{(j)} - s||_2/||s||_2$, and of $||\, f\, ||_{(Z^T MZ)^{-1}}/||\, s\, ||_{(Z^T MZ)^{-1}}$ and its estimate, for each mesh and each value of K_1 respectively. We choose $\eta = h$ (see the values of $\eta = h$ in Table 1) in equ:Mstop as suggested by Arioli (2000*a*). Arioli (2000*a*) proved, in the framework of a classical finite-element approximation, that the energy norm of the difference between the function corresponding to the computed algebraic solution and the true solution of the continuous problem is of order h. Our numerical experiments support the same conclusion for the velocity field in the framework of the mixed finite-element approximation. We denote by N_{iter} the iteration on which (16), with η chosen as in Table 1, stops the conjugate gradient. We point out that the error estimate is relative to the errors at the step $N_{iter} - d$. Nevertheless, because the conjugate gradient energy norm convergence is monotone, we can safely use the final values of $w^{(N_{iter})}$. In Table 2, we report the errors of the final step, the values chosen for the parameter d used in computing the estimate ξ_j, and the errors between u and u^*, p and p^*. For the sake of simplicity, we used the same value of d within each mesh. We point out that for the biggest values of K, we obtained similar results using smaller values of d.

The condition number $\kappa(Z^T MZ) = ||Z^T MZ||\, ||(Z^T MZ)^{-1}||$ of the projected Hessian matrix is uniformly bounded by a values which depends on K but not on h (Brezzi and Fortin 1992). Therefore, for decreasing values of h, the convergence of the conjugate gradient, measured in error energy norm, will depends only on K. In our test problems, the convergence

has a staircase behaviour where the length, the depth, and the steepness of the steps increases when K_1 decreases. The choice of d depends on the length and the steepness of these steps and moderately on h. If we choose a smaller value for d then we can have phenomena similar to the one of Fig 5 relative to $K_1 = 10^{-8}$. When scaling is performed, the convergence is faster and without steps because the condition number of the scaled projected hessian matrix depends neither on K nor on h (see Table 3).

In Fig 2, we plot the velocity field $\mathbf{u}_h(\mathbf{x})$, which is obtained solving (5) by **MA47**, and the velocity field $\mathbf{u}^*(\mathbf{x})$, which is computed by the null space algorithm using (16), for specific choices of the mesh and permeability. From Table 2, the error between the two fields is $\approx 10^{-2}$ while $h = 5.8\ 10^{-2}$. We point out that p^* is computed with less accuracy (see Table 2).

In Fig 6, we show that the energy norm of the solution at each step converges quite fast to the energy norm of u.

Finally, we observe that the symmetric scaling (17) normalising to 1 the entries M_{ij}, increases the rate of convergence of the conjugate gradient for 2-D problems. Nevertheless, this improvement is much less dramatic when the permeability is not isotropic, and when we solve 3-D problems. Moreover, we point out that this null space algorithm can be easily generalised to solve non-linear problems in which the permeability tensor depends on the solution. Within this framework, the scaling process imposes a new factorisation at each step of the Newton algorithm, making the total cost prohibitive.

6 CONCLUSIONS

We can see from the numerical experiments that the null space algorithm based on the Householder factorisation gives a projected Hessian matrix asymptotically independent of n and m and, thus, of the parameter h. This follows from the inf-sup condition (Brezzi and Fortin 1992) and the choice of the Raviart-Thomas finite-element approximation functions.

Nevertheless, when the physical problem is very ill-conditioned owing to the presence of very low permeability regions in the domain, the rate of convergence of the conjugate gradient does not deteriorate dramatically.

The Householder decomposition-based algorithm stores a number of non-zero entries for R and H, which increases with n and m. In Arioli, Maryška, Rozložník and Tůma (2001), the authors point out that, for 3-D problems, the computer memory requirement for storing H can become prohibitive.

Table 1: Parameters of the runs.

	n	m	nnz(A)	nnz(M)	nnz(R)	stg(H)	η
Mesh 1	237	147	416	1119	940	3023	0.16
Mesh 2	2309	1497	4396	11291	16058	47120	$5.8\,10^{-2}$
Mesh 3	22919	15136	45084	113735	250944	649793	$1.92\,10^{-2}$

Table 2: Stopping iteration and final residual (without Scaling)

		$K_1 = 10^{-2}$	$K_1 = 10^{-4}$	$K_1 = 10^{-6}$	$K_1 = 10^{-8}$
Mesh 1 d = 15	N_{iter}	21	25	29	31
	$\frac{\|\|u-u^*\|\|_2}{\|\|u\|\|_2}$	$2.8\,10^{-3}$	$2.8\,10^{-2}$	$2.8\,10^{-2}$	$3.5\,10^{-2}$
	$\frac{\|\|p-p^*\|\|_2}{\|\|p\|\|_2}$	$5.1\,10^{-4}$	$6.6\,10^{-3}$	$3.4\,10^{-2}$	22.7
Mesh 2 d = 25	N_{iter}	42	66	80	93
	$\frac{\|\|u-u^*\|\|_2}{\|\|u\|\|_2}$	$9.0\,10^{-4}$	$2.4\,10^{-3}$	$5.1\,10^{-3}$	$1.7\,10^{-2}$
	$\frac{\|\|p-p^*\|\|_2}{\|\|p\|\|_2}$	$4.0\,10^{-4}$	$5.2\,10^{-3}$	$6.5\,10^{-1}$	2.08
Mesh 3 d = 30	N_{iter}	59	96	119	143
	$\frac{\|\|u-u^*\|\|_2}{\|\|u\|\|_2}$	$4.2\,10^{-4}$	$1.2\,10^{-3}$	$3.1\,10^{-3}$	$7.8\,10^{-3}$
	$\frac{\|\|p-p^*\|\|_2}{\|\|p\|\|_2}$	$1.8\,10^{-5}$	$1.4\,10^{-3}$	$2.0\,10^{-4}$	1.66

Table 3: Stopping iteration and final residual (with Scaling)

		$K_1 = 10^{-2}$	$K_1 = 10^{-4}$	$K_1 = 10^{-6}$	$K_1 = 10^{-8}$
Mesh 1 d = 5	N_{iter}	6	6	6	6
	$\frac{\|\| u-u^* \|\|_2}{\|\| u \|\|_2}$	$1.1\,10^{-5}$	$4.0\,10^{-6}$	$2.0\,10^{-6}$	$1.9\,10^{-6}$
	$\frac{\|\| p-p^* \|\|_2}{\|\| p \|\|_2}$	$4.4\,10^{-7}$	$1.5\,10^{-6}$	$1.6\,10^{-6}$	$1.6\,10^{-6}$
Mesh 2 d = 5	N_{iter}	7	7	7	7
	$\frac{\|\| u-u^* \|\|_2}{\|\| u \|\|_2}$	$1.4\,10^{-5}$	$5.4\,10^{-6}$	$4.5\,10^{-6}$	$4.5\,10^{-6}$
	$\frac{\|\| p-p^* \|\|_2}{\|\| p \|\|_2}$	$2.8\,10^{-7}$	$1.0\,10^{-6}$	$1.1\,10^{-6}$	$1.1\,10^{-6}$
Mesh 3 d = 5	N_{iter}	8	8	8	8
	$\frac{\|\| u-u^* \|\|_2}{\|\| u \|\|_2}$	$4.5\,10^{-6}$	$4.4\,10^{-6}$	$4.4\,10^{-6}$	$4.4\,10^{-6}$
	$\frac{\|\| p-p^* \|\|_2}{\|\| p \|\|_2}$	$1.8\,10^{-8}$	$1.9\,10^{-8}$	$1.9\,10^{-8}$	$1.9\,10^{-8}$

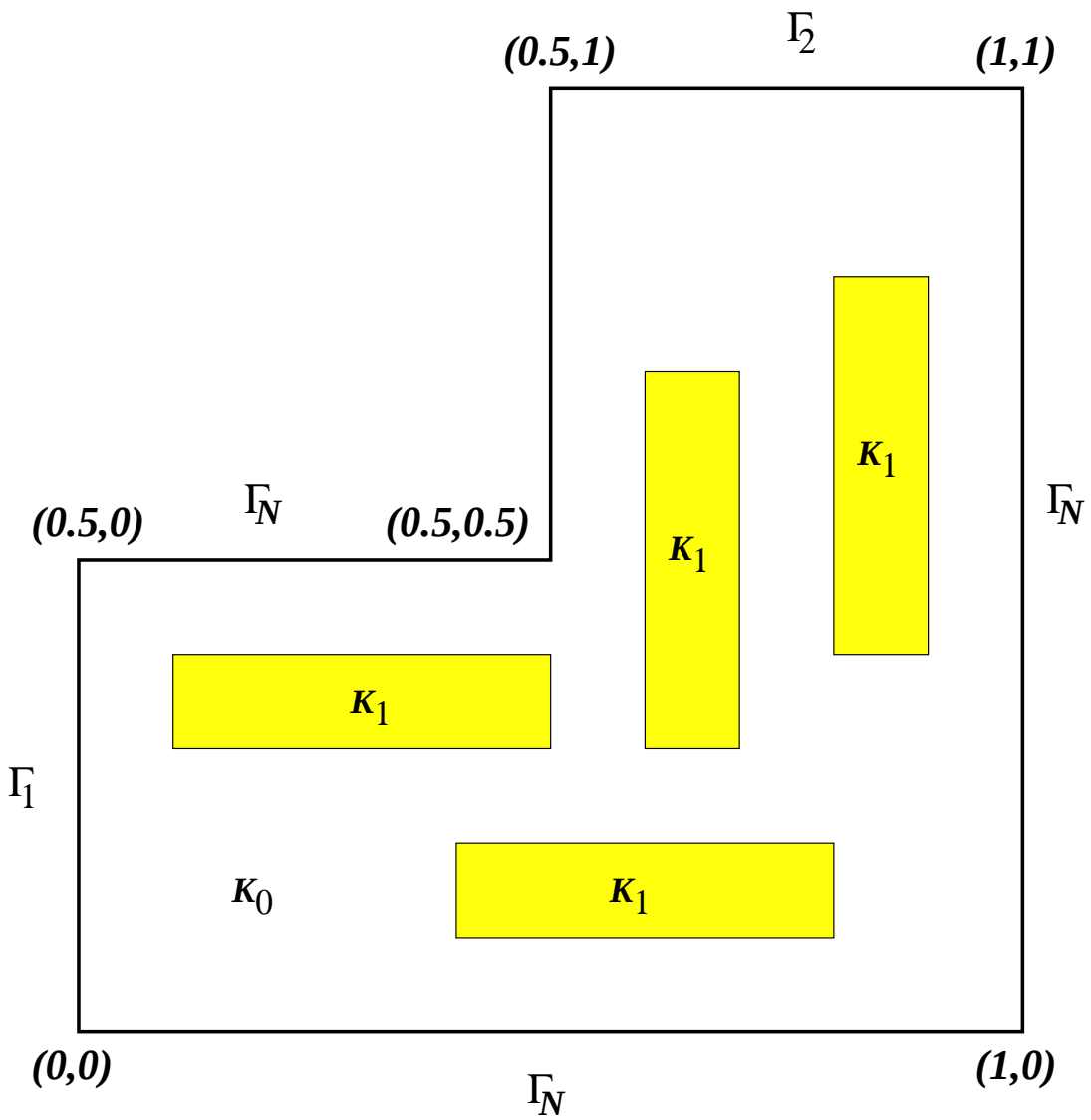

Figure 1: Sketch of the test problem

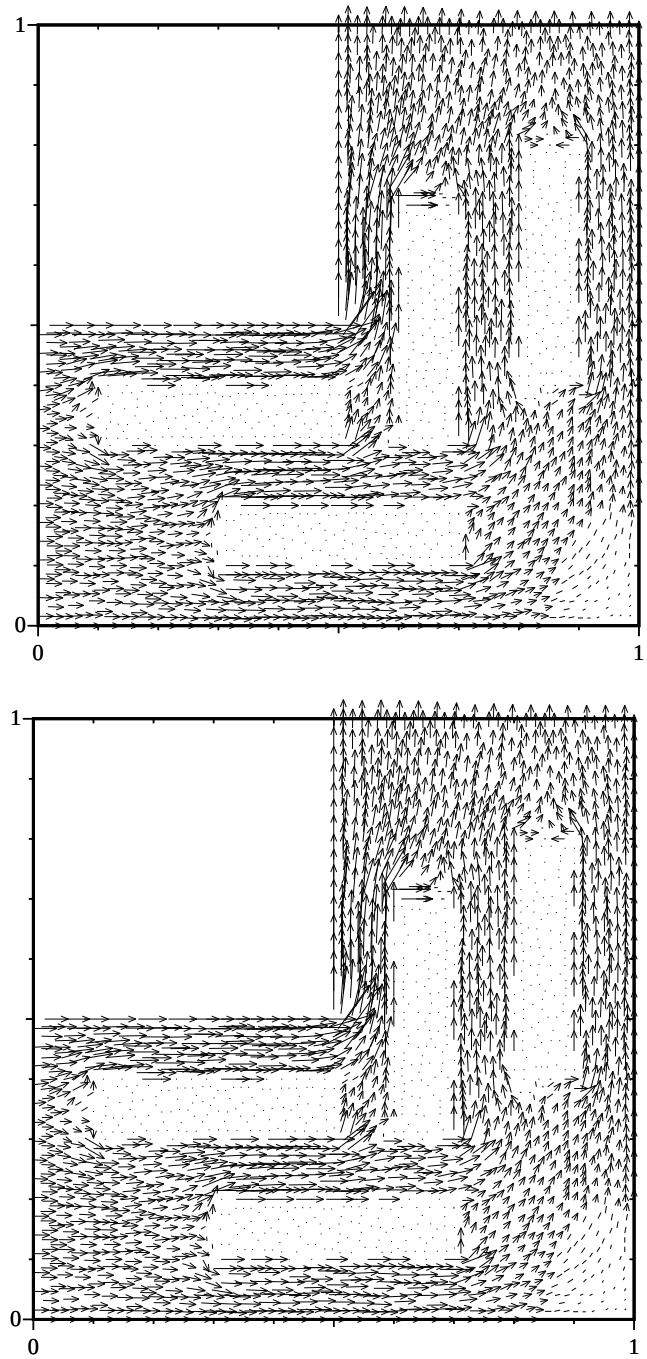

Figure 2: Velocity fields produced by the direct (top) and iterative (bottom) solvers on Mesh 2 with $K_1 = 10^{-8}$.

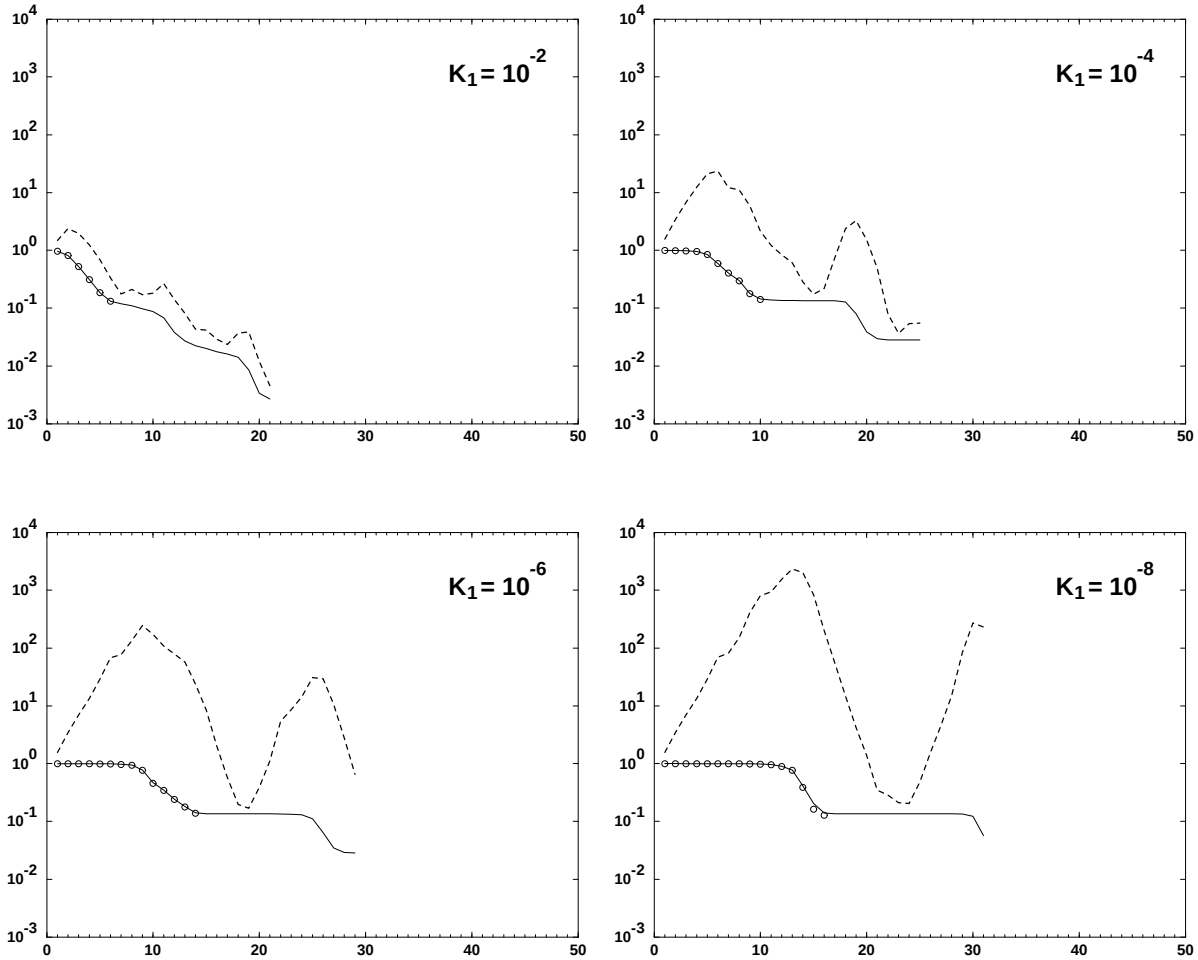

Figure 3: Mesh 1 calculations for different relative permeabilities: "exact" error (solid), estimated error (circles), residual norm (dotted).

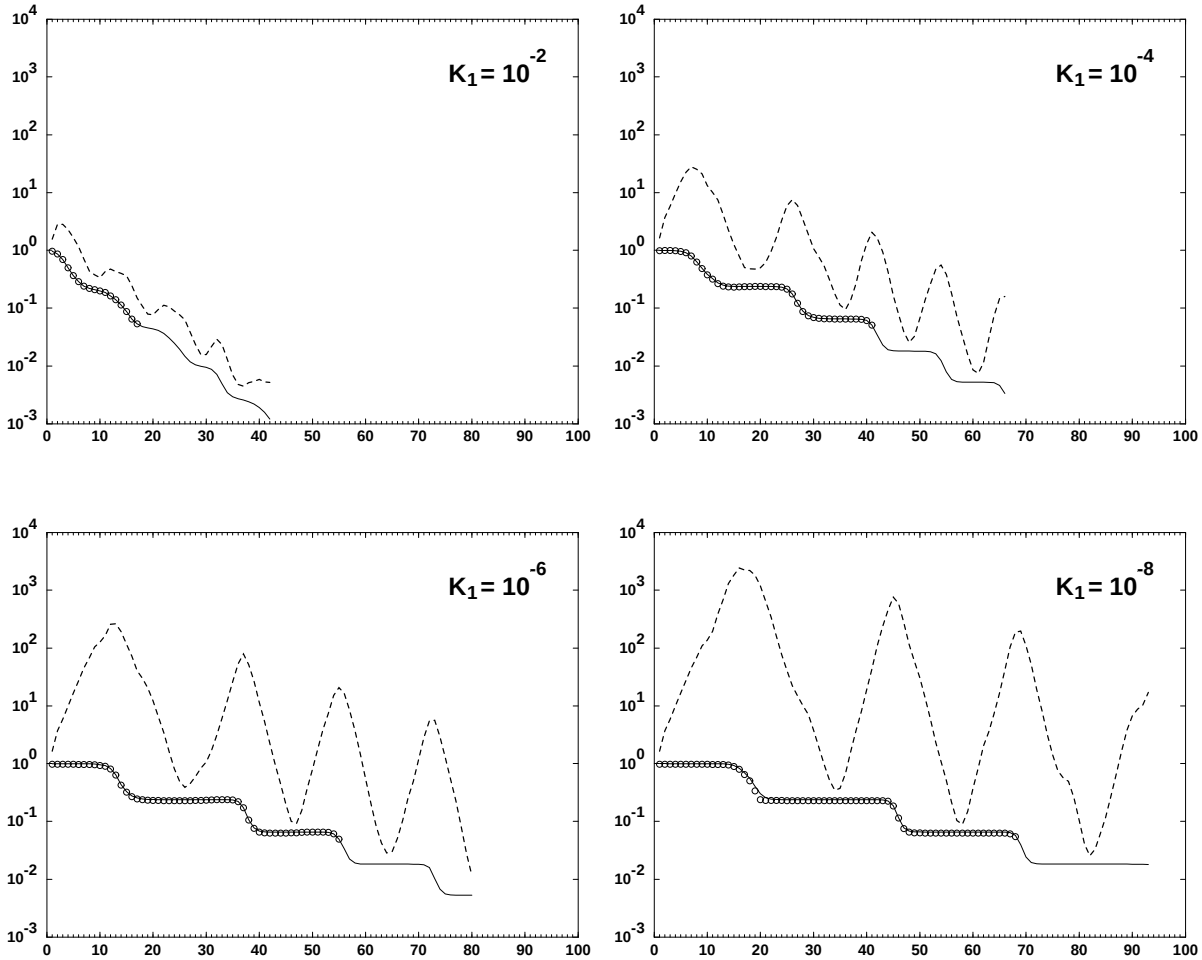

Figure 4: Mesh 2 calculations for different relative permeabilities: "exact" error (solid), estimated error (circles), residual norm (dotted).

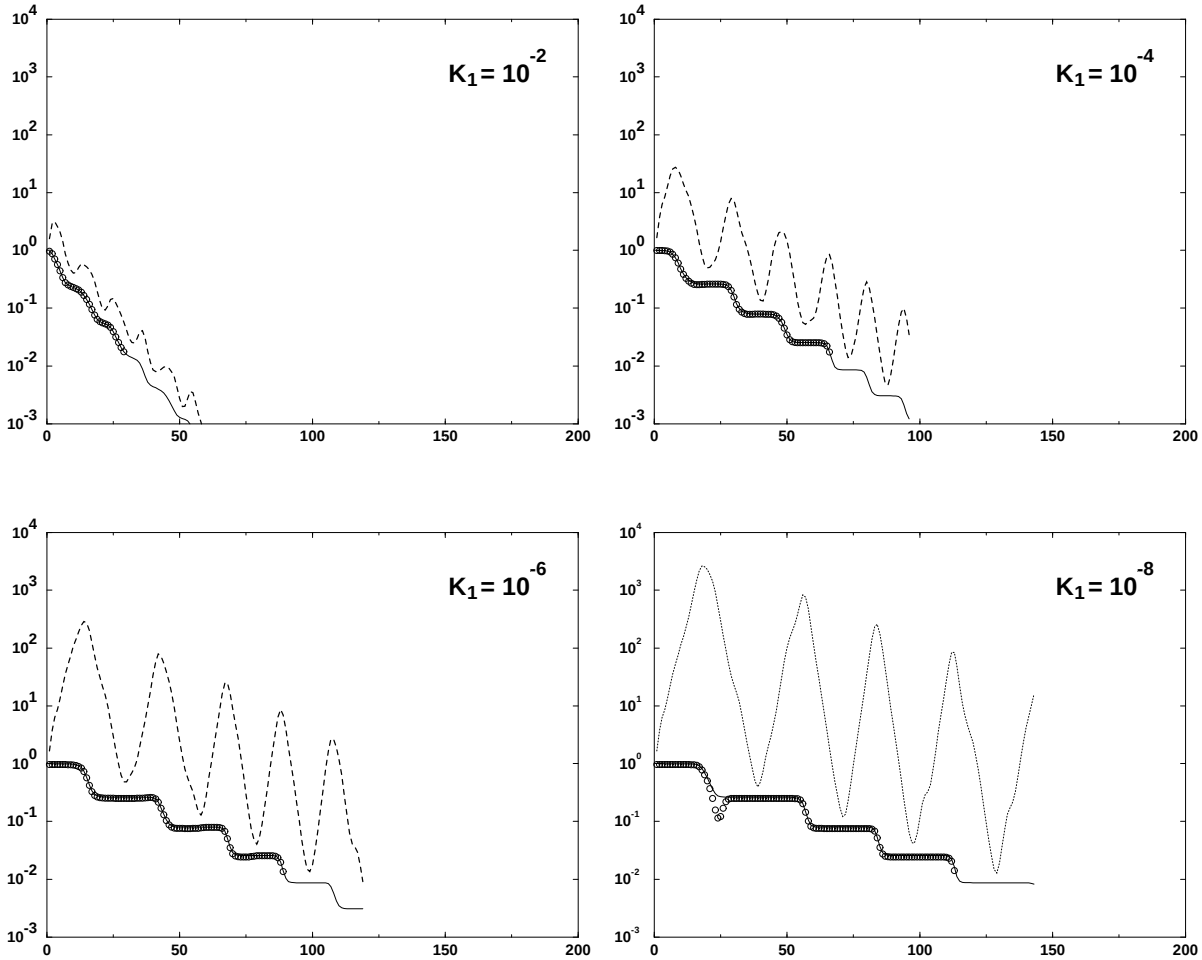

Figure 5: Mesh 3 calculations for different relative permeabilities: "exact" error (solid), estimated error (circles), residual norm (dotted).

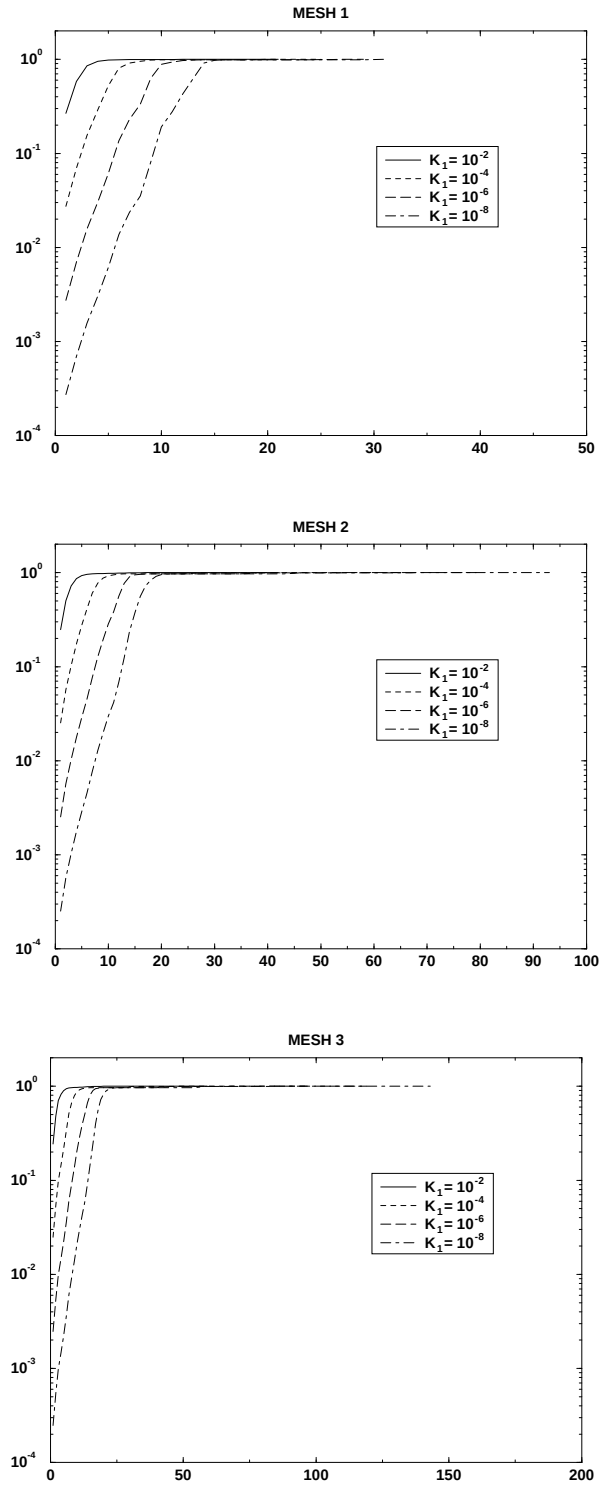

Figure 6: Ratio of the energy norm of the "exact" and the iterative solution for the three calculations.

References

Amestoy, P., Duff, I. and Puglisi, C. (1996), 'Multifrontal QR factorization in a multiprocessor environment', *Numerical Linear Algebra with Appl.* **3**, 275–300.

Arioli, M. (2000*a*), A stopping criterion for the conjugate gradient algorithm in a finite element method framework, Technical Report Tech. Report IAN-1179, IAN.

Arioli, M. (2000*b*), 'The use of QR factorization in sparse quadratic programming and backward error issues', *SIAM J. Matrix Anal. and Applics.* **21**, 825–839.

Arioli, M. and Baldini, L. (1999), A backward error analysis of a null space algorithm in sparse quadratic programming, Technical Report Tech. Report IAN-1150, IAN.

Arioli, M., Maryška, J., Rozložník, M. and Tůma, M. (2001), Dual variable methods for mixed-hybrid finite element approximation of the potential fluid flow problem in porous media, in preparation.

Brezzi, F. and Fortin, M. (1992), *Mixed and Hybrid Finite Element Methods*, Springer-Verlag, Berlin.

Ciarlet, P. G. (1978), *The finite element method for elliptic problems*, North-Holland Publishing Company, Amsterdam, Holland.

Duff, I. and Reid, J. (1983), 'The multifrontal solution of indefinite sparse linear systems', *ACM Trans. Math. Softw.* **9**, 302–325.

Gill, P. H., Murray, W. and Wright, M. H. (1981), *Practical Optimization*, Academic Press, London, UK.

Golub, G. and Meurant, G. (1997), 'Matrices, moments and quadrature II: how to compute the norm of the error in iterative methods', *BIT* **37**, 687–705.

HSL (2000), 'A collection of Fortran codes for large scale scientific computation'. http://www.cse.clrc.ac.uk/Activity/HSL.

Meurant, G. (1997), 'The computation of bounds for the norm of the error in the conjugate gradient algorithm', *Numerical Algorithms* **16**, 77–87.