Journal of Engineering Mathematics **29:** 473–496, 1995.

Operator weighing in a multigrid method for locally refined grids

G.H. SCHMIDT†
Shell Research B.V., P.O. Box 60, 2280 AB Rijswijk, The Netherlands

Received 14 July 1992; accepted in revised form 23 May 1995

Abstract. A multigrid method for an elliptic linear boundary value problem is presented. A consistent lowest-order mixed-finite-element discretisation is used on Cartesian locally refined grids in up to three space dimensions. The solution of the indefinite system of equations is reformulated as a problem of constrained minimisation. The constraint is satisfied exactly after one multigrid cycle; the functional is reduced iteratively by smoothing and coarse-grid corrections. By a suitable choice of prolongation and restriction operators, all corrections on coarser levels also reduce the functional within the constraint. This approach leads to a non-standard convergence proof which also holds for the variable-coefficient case. The proof does not predict the actual convergence rate, but shows that the functional will never increase after a multigrid cycle, while the constraint is satisfied exactly after the first multigrid cycle. The conditions required for convergence allow some freedom in choosing the restriction and prolongation operators, which can be used to improve the convergence rate. This leads to operator weighed restriction and prolongation operators in a novel manner. Some numerical examples are presented to demonstrate the robustness of the method.

1. Introduction

This paper describes a multigrid method for the solution of an elliptic set of linear equations, which are discretised by means of mixed finite elements on a locally refined grid. It is well-known that the mixed-finite-element discretisation is not positive definite, even when this discretisation is applied to an elliptic system of equations. It is therefore difficult to construct a multigrid method for the solution of this indefinite system which is as robust as multigrid applied to a positive definite discretisation of the same equations. The indefinite system requires a careful choice of prolongation and restriction operators.

It is the typical feature of mixed-finite-element discretisation, that the unknown quantities (here: one scalar field and one vector field) are approximated by a *system of first-order differential equations*. This is in contrast with the more usual finite-element discretisation, where approximation is applied to the unknown quantities in one second-order differential equation. The multigrid method discussed here is stated fully in terms of first-order equations, also on the coarse grids. This feature occurs also in the multigrid methods presented in [1], [2] and [3]. The approach reported here is different, a.o., with respect to the equations to be solved, the prolongation and restriction operators, and the type of smoothing. It extends work reported in [4].

Prolongations and restrictions are introduced for scalar fields and for vector fields separately. Since the equations are of first order, the order of the prolongations and restrictions can be lower than in the usual finite-element discretisation. This low order makes it easier to define the prolongations and restrictions for locally refined grids. The method described here

† deceased (1992).
Revisions of this paper were taken care of by W.A. Mulder of Shell Research B.V.

is very well suited to dynamic regridding, possibly triggered by the time-dependent solution of evolution equations.

The construction of the multigrid computational process is by transforming the problem into a problem of *minimising an energy-like functional*, within kinematic constraints. Hence it is along the lines of the consistency theorems in [5] for the mixed-finite-element discretisation.

It is the principle of multigrid that equations are solved by means of corrections applied on a number of levels of discretisation. These corrections converge only when a correction on one level does not annul (or even invert) the effect of corrections on other levels. To avoid such conflicting corrections, they are such that they all reduce the same (fine-level) functional. It will be shown rigorously, that when prolongations and restrictions satisfy certain constraints, then the corrections on all levels do reduce the functional.

Numerical results for a variety of problems have shown that this multigrid method is very robust. This must probably be attributed to the minimisation principle: Whenever the prolongations and restrictions violate the constraints, numerical results show divergence for some extreme values of the coefficients in some part of the computational domain.

A simple construction of prolongations and restrictions, satisfying the constraints for constant values of the coefficients in the equations, was given in [4]. In this paper the coefficients have arbitrary space dependence. We give a construction of the prolongations and restrictions using local values of the coefficients.

When interpolating discontinuous solutions to problems with discontinuous coefficients, better results are obtained if the coefficients of the discrete operator are used to construct the interpolant (cf. Eq. (10.3.6) in [6]). This observation has led to the idea of operator weighing for the prolongation (interpolation) and restriction operators. Here, the operator weighing arises as a natural consequence of the above-mentioned requirement of a decreasing fine-grid functional. The requirement does not, however, completely specify the prolongation and restriction operators. This leaves a number of degrees of freedom that can be chosen to improve the convergence rate. The specific choice will depend on the application. Two examples are given in section 11.

The grids used here are constructed from topologically regular grids by (local) refinement and the grid generation requires in principle only booleans (refinement or not). It requires no computation of the position of grid points, as in more complex (curvilinear) grids. Therefore this type of grid generation is computationally cheap, which makes it suitable for dynamic regridding. The multigrid method given here is used at the numerical solution of initial-boundary-value problems, which occur e.g. in oil-reservoir simulation, see [7], [8] and [9]. The elliptic equations in these papers arise from implicit discretisation of the time derivatives. These equations have to be solved every time-step, with varying values of the coefficients and on varying grids. This requires extreme robustness of the multigrid method.

This paper presents the construction of the multigrid method and the constraints on the prolongation and restriction operators. Numerical examples in the form of computed spectral radii of the error amplification matrix are given in section 12. Other examples can be found in, e.g., [7], [8], [10].

2. Elliptic set of equations

This paper deals with the linear equations:

$$c\varphi + \nabla \cdot \mathbf{u} = q, \tag{2.1a}$$

$$\nabla\varphi + W\mathbf{u} = \mathbf{g}, \tag{2.1b}$$

in a region Ω in N space dimensions ($N = 1$, 2 or 3). At each point of the boundary there is one of the boundary conditions:

$$\mathbf{u} \cdot \nu = u_n, \quad \mathbf{x} \in \partial\Omega_N; \qquad \varphi - \alpha\mathbf{u} \cdot \nu = \beta, \quad \mathbf{x} \in \partial\Omega_D, \tag{2.2a,b}$$

where ν is the unit outward normal of the boundary $\partial\Omega = \partial\Omega_N \cup \partial\Omega_D$.

Equations (1) and (2) constitute an elliptic boundary value problem for *potential* φ and *flux* $\mathbf{u}$. Elimination of $\mathbf{u}$ results in (the spatial part of) the inhomogeneous, anisotropic diffusion equation for φ. The coefficients are known functions of the space-coordinates. We assume the following point-wise conditions to hold:

- the scalars c and α are non-negative,
- the tensor W is positive definite, bounded away from zero.

The equations (2.1) and (2.2) are equivalent to minimisation of the "energy-functional":

$$f = \frac{1}{2}\int_\Omega \{c\varphi^2 + (W\mathbf{u} - 2\mathbf{g}) \cdot \mathbf{u}\} + \frac{1}{2}\int_{\partial\Omega_D} (\alpha\mathbf{u} \cdot \nu + 2\beta)(\mathbf{u} \cdot \nu), \tag{2.3}$$

within the constraints (2.1a) and (2.2a), as follows by means of Lagrange multipliers.

3. Topological structure of locally refined grids

We assume that the computational domain Ω is covered by a Cartesian (possibly non-equidistant) grid, which is called the *base grid*. The grid divides space in *blocks* Ω_i (dimension N). The intersection of the closure of two blocks is a *face*, provided that the dimension of that intersection equals $N - 1$. The boundary $\partial\Omega$ is also composed of faces. Blocks will always be denoted by a *subscript* i (or i') and faces are denoted by a *subscript* j (or j').

A block Ω_i can be refined by the *basic refinement*, which means that it is divided into 2^N identical smaller blocks $\Omega_{i'}$, called its *sons*. The set of 2^N sons of block i is denoted by $I(i)$. The basic refinement creates $N2^{N-1}$ faces in the interior of Ω_i, which are called *block-born faces*. The refinement also divides each face in the boundary of Ω_i in 2^{N-1} identical smaller faces; these faces are also called sons and they are *face-born*. The set of 2^{N-1} sons of face j is denoted by $J(j)$. All blocks and faces in the base grid have *level* 1. Refinement applied to a block of level l gives blocks and faces of level $l + 1$. The levels are denoted by l_i for blocks and l_j for faces.

The equations of the preceding sections are discretised on the *discretisation grid* Ω^L, which is obtained from the base grid by applying (locally) the basic refinement an arbitrary number of times up to an arbitrary level. L is the maximum level of blocks and faces in the grid. In a *regular* grid all blocks are refined up to the same level; otherwise the grid is *irregular*.

We say that a block and a face are *adjacent*, when the intersection of the boundary of the block and the face has dimension $N-1$. Faces in the boundary of the computational domain are called *boundary faces*; the others are called *interior faces*.

When a block in Ω^L has been refined, we call it *coarse*; otherwise it is called *fine*. Note that an irregular grid can have fine blocks of level 1 up to L.

Let face j be an interior face of Ω^L and let it have level l. When all fine blocks adjacent to face j have level less than or equal to l, then it has just two adjacent fine blocks and the face is called *fine*. When both adjacent fine blocks of a fine face have level l, the face is called *regular*. When all fine blocks adjacent to face j have level greater than l, the face is called *coarse*. When face j is not fine, but it has one adjacent fine block with level l, it is a *refinement face*. The sons of a refinement (sub-)face are *refinement subfaces*.

In the multigrid process we use a *series of grids* $\Omega^1, \ldots, \Omega^L$. The grid Ω^{l-1} is obtained from Ω^l by deleting all blocks and faces of level l. All grids in the series, except for Ω^1 (the base grid), can be irregular. Figure 1 gives an example of this series for $N=2$ and $L=6$. Definitions of fine and coarse blocks and faces etc. were given with respect to Ω^L. These definitions can also be stated with respect to other grids in the series.

4. Mixed-finite-element discretisation

In this section the discretisation of the equations (1) and (2) on a grid Ω^L is stated. Here fine blocks, regular faces, fine boundary faces and refinement faces only play a role. Coarse blocks and faces occur in the multigrid process of the next sections. Refinement subfaces are convenient for coding, but they occur neither in the discretisation nor in the multigrid process.

The lowest-order Raviart-Thomas approximation of φ and $\mathbf{u}$ is:

$$\varphi(\mathbf{x}) = \sum_i \varphi_i \psi_i(x); \qquad \mathbf{u}(\mathbf{x}) = \sum_j u_j \mathbf{v}_j(\mathbf{x}). \tag{4.1a,b}$$

Here the summation over i is over all fine blocks and the summation over j is over all regular faces, fine boundary faces and refinement faces. The approximation function $\psi_i(x)$ is the indicator function of Ω_i. The vector valued approximation function $\mathbf{v}_j(\mathbf{x})$ is defined on Ω as follows: Its direction is that of the coordinate perpendicular to face j. Its magnitude is $1/a_j$ on face j, where a_j represents the area of face j ($N=3$), the length of face j ($N=2$) or 1 ($N=1$). Its magnitude is zero on all other faces. In the interior of blocks the magnitude is obtained by linear interpolation in the direction perpendicular to face j. Observe that the support of $\mathbf{v}_j$ is the union of all fine blocks adjacent to face j.

The coefficients φ_i and u_j are the unknowns to be computed, except for the u_j in $\partial\Omega_N$, which have known values. They represent respectively an average of φ over Ω_i and an integral of the normal component of $\mathbf{u}$ over face j.

A well-known condition for consistent approximation by mixed finite elements is that the range of the approximations of $\nabla \cdot \mathbf{u}$ is included in the range of the approximations of φ : $\mathcal{R}(\nabla \cdot \mathbf{u}) \subset \mathcal{R}(\varphi)$. The approximation (4.1) satisfies this condition, as may be verified by the reader.

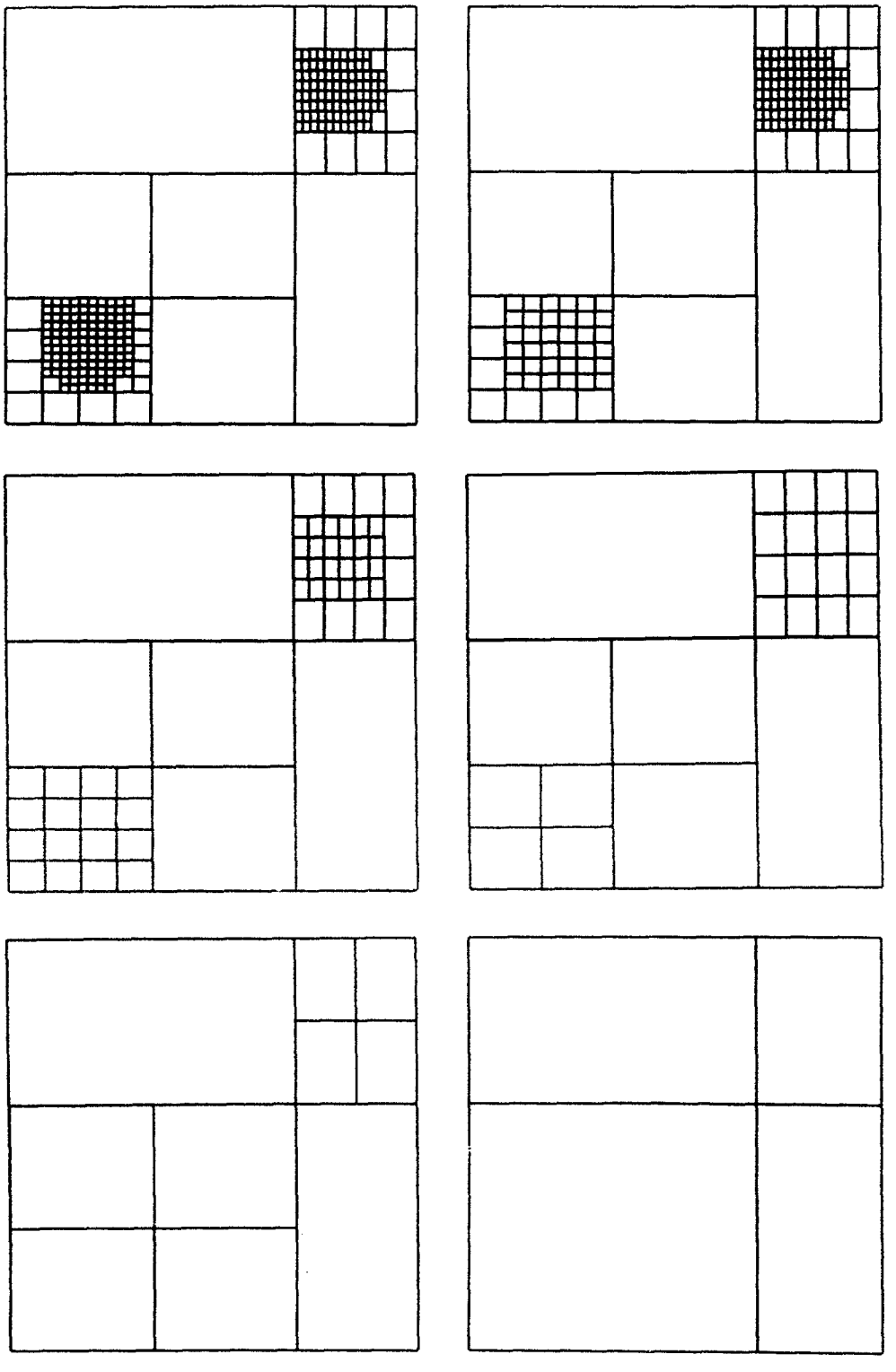


Fig. 1. The multigrid series of grids $\Omega^6, \ldots, \Omega^1$.

The discretisation of (2.1) and (2.2) is:

$$c_i\varphi_i + \sum_j d_{ij}u_j = q_i; \qquad -\sum_i d_{ij}\varphi_i + \sum_{j'} w_{jj'}u_{j'} = g_j, \tag{4.2a,b}$$

where:

$$c_i = \int_{\Omega_i} c; \qquad q_i = \int_{\Omega_i} q; \tag{4.3a,b}$$

$$w_{jj'} = \int_{\Omega}(W\mathbf{v}_j \cdot \mathbf{v}_{j'}) + \int_{\Omega_D} \alpha(\mathbf{v}_j \cdot \nu)(\mathbf{v}_{j'} \cdot \nu); \tag{4.4a}$$

$$g_j = \int_{\Omega}(\mathbf{g} \cdot \mathbf{v}_j) - \int_{\partial\Omega_D} \beta(\mathbf{v}_j \cdot \nu); \tag{4.4b}$$

$$d_{ij} = \begin{cases} 0, & \text{if } i \text{ and } j \text{ are not adjacent;} \\ 2^{-(N_1)(l_i - l_j)}, & \text{if face } j \text{ is at the positive side of block } i; \\ -2^{-(N-1)(l_i - l_j)}, & \text{if face } j \text{ is at the negative side of block } i. \end{cases} \tag{4.5}$$

The indices i run through all fine blocks. The corresponding φ_i are the entries of the vector Φ^f. The indices j and j' run through all regular faces, fine boundary faces and refinements faces. The corresponding u_j are the entries of the vector U^f. The *superscript* f refers to the "fine" grid, or discretisation grid, to mark it off from matrix equations on coarser grids to be state in next sections. The equations (4.2) can be written as matrix equations:

$$C^f\Phi^f + D^fU^f = Q^f; \qquad -D^{f*}\Phi^f + W^fU^f = G^f, \tag{4.6a,b}$$

where * denotes the transpose of a matrix. The matrix C^f is a non-negative definite diagonal matrix; W^f is positive definite. The complete system (4.6) is indefinite.

The equations (4.6) have a unique solution, except for the so called incompressible case. In the *incompressible case* all c_i are zero and $\partial\Omega = \partial\Omega_N$. Then a solution exists only if source terms q_i and boundary fluxes satisfy a linear relation, which is assumed to hold. There is a one-parameter family of solutions, characterised by a shift of the φ_i over a constant value.

Observe that (2.1a) holds point-wise in the classical sense, when the functions c and q are block-wise constant and when the coefficients satisfy (4.6a). Equation (4.6b) is equivalent with minimisation of the functional (2.3) with the constraints (2.1a) and (2.2a) and within the approximation (4.1). In the computational process to solve (4.6) we also use this minimisation formulation. We state the *minimisation problem* as: Find the minimum of the functional:

$$F^f(\Phi^f, U^f) = \frac{1}{2}\langle C^f\Phi^f, \Phi^f\rangle + \frac{1}{2}\langle W^fU^f - 2G^f, U^f\rangle, \tag{4.7}$$

within the constraint (4.6a). The brackets $\langle\cdot,\cdot\rangle$ in (4.7) represent the l_2-inner-product. The minimisation problem and the problem of solving (4.6) are solved by the same values of Φ^f and U^f.

5. Smoothing

This paper describes a multigrid method to solve (4.6) for irregular grids. The computational process consists of smoothing and coarse-grid corrections. The *smoothing acts on regular grids or regular parts of the grid only.* In the next sections we will show that smoothing and coarse-grid corrections will also converge to the solution on irregular grids.

The full process to solve (4.6) consists of an initialisation and an iterative part. The *initialisation* makes the unknown satisfy (4.6a). This means that we have *exact* "mass conservation" after one multigrid cycle. The *iterative part* applies iterative corrections to Φ^f and U^f within the constraints (4.6a), thereby reducing F^f. This means that we iterate on "Darcy's law" in porous media applications or "Ohm's law" in electrostatic problems such as occur in well logging.

The convergence proof presented here shows that the functional will never increase. This is different from the usual multigrid convergence proofs based on the "approximation" and "smoothing" properties [6]. Strictly speaking, we do not prove convergence but only non-divergence.

Smoothing consists of making sweeps of local updates as in a Gauss-Seidel process. Each *local update* consists of the following correction applied to the current value of Φ^f and U^f: A connected *local region* is composed by choosing a subset of the (fine) blocks Ω_i. On this local region a Neumann boundary value problem is formulated. The current values of u_j on the boundary of the local region serve as boundary values for the local boundary value problem. New values are computed for the φ_i and u_j in the local region by solving the local boundary value problem. These new values update the current values of φ_i and u_j. Updated values are denoted by a *tilde* ($\tilde{}$), to distinguish them from previous values.

The minimisation principle stated for the boundary value problem on the whole grid Ω^L holds also for the local boundary value problem. As a consequence of this principle, the functional decreases due to each local update:

$$F^f(\tilde{\Phi}^f, \tilde{U}^f) - F^f(\Phi^f, U^f) \leqslant 0, \tag{5.1}$$

whenever (4.6a) did hold before applying the update. Inequality (5.1) holds also for the result of a sweep of sequential local updates. In the iterative part of the computational process, the equations (4.6a) do hold before applying the local update, so (5.1) holds in the iterative part.

We can use local regions of different forms, e.g. line-regions in case of strong anisotropy in the tensor W, but they always are composed of blocks of one level. The boundary conditions are always of Neumann type, except when the boundary of the local region and $\partial\Omega_D$ have a non-empty intersection. In the latter case we have boundary conditions of the type (2.2b). These have been included in the equations (4.6b), as follows from (4.4).

For the numerical examples in section 12, lexicographical Gauss-Seidel relaxation has been used on 2^N blocks simultaneously. That is, we scan over all blocks on a given level l. If a block has neighbouring blocks to the "right" that live on the same level l for all N coordinates, then a local linear system is formed and solved. It should be noted that the lexicographical ordering is determined by the adaptive gridding and may be irregular, except for the fact that block-born blocks are always grouped together. No under- or over-relaxation is applied.

In two space dimensions, the linear system is formed for a square configuration of 2×2 blocks, labelled by the left lower one. The unknowns are the 4 values of $\phi_{i'}$ in the blocks and the 4 values of $u_{j'}$ living at the faces between these 4 blocks. The values of u_j on the boundary of the 2×2 configuration are kept fixed. Equations (4.6a,b) provide an 8×8 linear system that is solved by Gauss-elimination with pivoting. In the "incompressible" case ($c_{i'} = 0$), this linear system is singular. In that case, one of the equations of (4.6a) is replaced by the condition that the sum of the $\phi_{i'}$ does not change.

6. Prolongations and restrictions

In this section we start the description of the multigrid process by specifying the prolongations and restrictions that will be used. These prolongations and restrictions are linear operators acting between two consecutive grids $\Omega^c = \Omega^l$ and $\Omega^f = \Omega^{l+1}$ in the series of grids of section 3. The Φ^c and U^c are defined on Ω^c analogously as the Φ^f and U^f are defined on Φ^f. We use five different linear operators:

$$P^q : \Phi^c \to \Phi^f; \quad P^\varphi : \Phi^c \to \Phi^f; \quad R^\varphi : \Phi^f \to \Phi^c; \quad P^u : U^c \to U^f; \quad R^u : U^f \to U^c.$$

The linear operators are defined by matrices and we will use the same symbols for the matrices. All fine blocks and all fine faces and refinement faces in Ω^f of level l or less occur also in Ω^c. For these blocks and faces each of the operators acts as the identity operator.

The prolongation P^q is defined by:

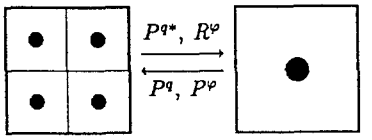
$$p^q_{i'i} = \begin{cases} 1, & i' \in I(i), \\ 0, & \text{otherwise.} \end{cases} \tag{6.1}$$

Fig. 2a indicates the sparsity pattern of P^q by marking all non-zero entries of $P^q E^c_i$, where the entries of "unit-vector" E^c_i are all zero, with the exception of the entry for block i having the value one.

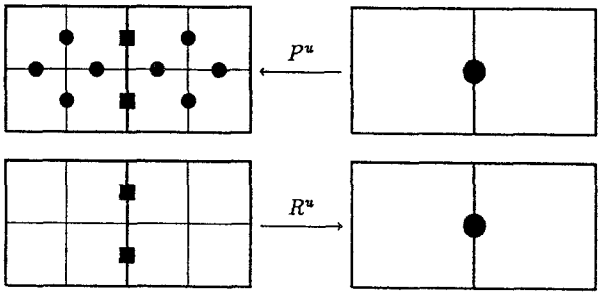
For the operators P^φ, R^φ, P^u and R^u we give here only the sparsity pattern. The non-zero entries of these matrices are parameters to be determined in section 11. The sparsity pattern of P^φ and $R^{\varphi *}$ are given in Fig. 2a for the two-dimensional case. In Fig. 2b, the sparsity patterns of R^u and P^u are given. All non-zero entries of $P^u E^c_j$ are marked. Here the entries of E^c_j are zero for all faces in Ω_c, with the exception of a value 1 for face j. In these figures, both coarse-grid blocks are assumed to be refined. If one of them is not, say the right one, the block-born faces, marked by filled circles, on the right do not occur. The face-born faces are marked by black squares. If, again, the block on the right side is not refined (j is a refinement face), the face-born faces are not involved in the discretisation. In the actual implementation, however, it is convenient to add the refinement faces to the data structure and set the fluxes on the refinement face equal to $1/2^{N-1}$ times the flux on the parent face.

7. Coarse-grid equations

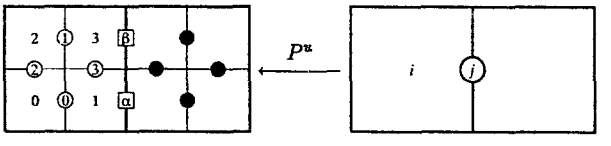
In section 4 we stated equations of the form (4.6) on the discretisation grid. Here we derive equations of the same form on each grid in the series of section 3. The Full-Approximation-Storage (FAS) scheme [11] is used.



(a)



(b)



(c)

Fig. 2. (a) Stencil for the restriction operators P^{q*} and R^{φ} and the prolongation operators P^{q} and P^{φ} in two dimensions. (b) Sparsity pattern of the prolongation operator P^{u} and the restriction operator R^{u} in two dimensions. (c) Labelling for the example in section 11.

Let Ω^c and Ω^f be two consecutive grids in the series, an let (4.6) be stated on Ω^f. Then the coarse-grid equations, defined on Ω^c, are:

$$C^c\Phi^c + D^cU^c = Q^c; \qquad -D^{c*}\Phi^c + W^cU^c = G^c. \tag{7.1a,b}$$

Here we have *coarse-grid matrices*:

$$C^c = P^{\varphi*}C^fP^{\varphi}; \qquad D^c = P^{q*}D^fP^u; \qquad W^c = P^{u*}W^fP^u, \tag{7.2a,b,c}$$

and *coarse-grid right-hand sides*:

$$Q^c = P^{q*}(Q^f - C^f\Phi^f) + C^cR^{\varphi}\Phi^f; \tag{7.3a}$$

$$G^c = P^{u*}(G^f - W^fU^f) + W^cR^uU^f. \tag{7.3b}$$

The sparsity pattern of the matrices C^c, D^c and W^c is equal to the sparsity pattern of C^f, D^f and W^f, as is easily verified.

8. Constraints for the prolongations and restrictions

In this section we state matrix equations to be satisfied by the prolongation and restriction matrices. We will use these equations in the next sections to show that the multigrid process solves the minimisation problem. In Section 11 we will show how the prolongation- and restriction-matrices can be chosen appropriately to solve the equations stated here.

The equations are:

$$P^{\varphi *} C^f P^{\varphi} = P^{q*} C^f P^{\varphi}; \qquad C^c R^{\varphi} = P^{\varphi *} C^f; \qquad D^c R^u = P^{q*} D^f, \tag{8.1a,b,c}$$

and

$$\mathcal{N}\{P^{q*}(C^f P^{\varphi} \quad D^f P^u)\} = \mathcal{N}\{(C^f P^{\varphi} \quad D^f P^u)\}, \tag{8.2}$$

where $\mathcal{N}$ denotes the null space of a matrix. This is equivalent to:

$$P^{q*}(C^f P^{\varphi} \Phi^c + D^f P^u U^c) = 0 \Rightarrow C^f P^{\varphi} \Phi^c + D^f P^u U^c = 0, \quad \text{for all } \Phi^c \text{ and } U^c. \tag{8.2a,b}$$

In Section 10 and the Appendix it will be shown that these constraints are *sufficient* to guarantee that the fine-grid functional does not increase after each multigrid cycle.

9. Initialisation of the multigrid process

As already stated in section 5, the multigrid process consists of an initialisation (this section) and an iterative part (next section).

The initialisation changes Φ^f and U^f (which can have arbitrary values before the multigrid process) so that they satisfy (4.6a). This correction can easily be made by solving Φ^f from (4.6a), when the matrix C^f (a diagonal matrix) is non-singular. For many important applications C^f is zero or small and we give here a process which works for all non-negative C^f. The structure of the initialisation is as follows:

- In a *sweep down* over all grids $l = L-1, \ldots, 1$ we compute restrictions:

  $$\Phi^c = R^{\varphi} \Phi^f; \qquad U^c = R^u U^f, \tag{9.1a,b}$$

  and right-hand sides (7.3) for the coarse-grid equations.
- On the coarsest grid we compute new values $\tilde{\Phi}^c$ and $\tilde{U}^c$ from (7.1) by a direct method.
- In a *sweep up* over all grids $l = 2, \ldots, L$ we compute new values for the solutions by prolongation:

  $$\tilde{\Phi}^f = \Phi^f + P^{\varphi}(\tilde{\Phi}^c - \Phi^c); \qquad \tilde{U}^f = U^f + P^u(\tilde{U}^c - U^c). \tag{9.2a,b}$$

  Moreover, after prolongation to grid Ω^l we apply a *reduced smoothing sweep* on that grid, which consists of a number of local updates (see section 5). The local region in each update is the combination of the 2^N sons of one block in Ω^{l-1}. The reduced sweep runs over all combinations of this type in Ω^l.

This initialisation makes the solution satisfy (4.6a), as follows from the following arguments. On the coarse grid we have:

$$C^c \tilde{\Phi}^c + D^c \tilde{U}^c = Q^c. \tag{9.3}$$

From (9.2), (7.2a,b), (8.1a), (9.3), (7.3a), (9.1a,b) and (8.1c) we get (see appendix A):

$$P^{q*}(C^f\tilde{\Phi}^f + D^f\tilde{U}^f - Q^f) = 0. \tag{9.4}$$

This implies, by the specific choice (6.1) for P^q, that the residuals of the 2^N equations in the local update have sum zero. By the sparsity of the matrix D^f, these 2^N equations can be solved by giving appropriate values to the u_j for the block-born faces in the local update. If one reduced smoothing sweep is subsequently carried out, the updated solution will satisfy (4.6a).

10. Iterative part of the multigrid process

The iterative part of the multigrid process consists of iterative application of multigrid corrections till convergence. A *multigrid correction* consists of:

- In a *sweep down* over all grids $l = L-1, \ldots, 1$ we compute restrictions (9.1) and right-hand sides (7.3) for the coarse-grid equations.
- On the coarsest grid we compute new values $\tilde{\Phi}^c$ and $\tilde{U}^c$ from (7.1) by a direct method.
- In a *sweep up* over all grids $l = 2, \ldots, L$ we compute new values for the approximations by prolongation (9.2). Moreover, after prolongation to grid Ω^l we apply a *full smoothing sweep* on that grid, which consists of a number of local updates. The local region in each update is again a combination of 2^N blocks all having level l. The only difference with the reduced sweep is that the 2^N blocks do not have to be sons of the same father in Ω^{l-1}.

The effect of a multigrid correction is a decrease of the functional (4.7), within the constraints (4.6a). This is shown as follows. the equations (9.1a,b), (7.3a), (8.1c) and (4.6a) imply (7.1a), see appendix A. Then the equations (4.6a), (9.2a,b), (8.2), (7.2a,b), (8.1a), (7.1a) and (9.3) imply, see appendix A:

$$C^f\tilde{\Phi}^f + D^f\tilde{U}^f = Q^f \tag{10.1}$$

Hence the correction is within the constraint (4.6a). Smoothing on the coarse grid makes the functional decrease, since (7.1a) holds; see (5.1). The equations (7.2a,c), (7.3b), (8.1b), (9.1a) and (9.2a,b) imply, see appendix A:

$$F^f(\tilde{\Phi}^f, \tilde{U}^f) - F^f(\Phi^f, U^f) = F^c(\tilde{\Phi}^c\tilde{U}^c) - F^c(\Phi^c, U^c) \tag{10.2}$$

This means that decrease on a coarse grid induces *the same decrease* on the discretisation grid.

11. Operator weighing

The preceding sections state a multigrid method which converges to the solution of (4.6), provided that the prolongations P^φ and P^u and restrictions R^φ and R^u satisfy (8.1) and (8.2). In this section we give a construction of the prolongations and restrictions with the prescribed sparsity patterns, such that (8.1) and (8.2) are satisfied. It will appear that there are many degrees of freedom, which are used to introduce operator weighing.

Equation (8.1a) holds if

$$\sum_{i' \in I(i)} \{c_{i'} p^{\varphi}_{i'i}(p^{\varphi}_{i'i} - 1)\} = 0. \tag{11.1}$$

This leaves $2^N - 1$ degrees of freedom in $p^{\varphi}_{i'i}$ for each coarse-grid block i. The matrix C^c is given in (7.2a) and is a diagonal matrix. Hence we can introduce R^{φ} as: $R^{\varphi} = (C^c)^{-1} P^{\varphi *} C^f$, which implies:

$$r^{\varphi}_{ii'} = p^{\varphi}_{i'i} c_{i'} / c_i, \qquad i' \in I(i), \tag{11.2}$$

so that (8.1b) holds. When a diagonal element c_i of C^c equals zero, we set $r^{\varphi}_{ii'} = 2^{-N}$, and (8.1b) is also satisfied.

The sparsity pattern of P^u was given in section 6. When $p^u_{j'j}$ is non-zero, then either $j' \in J(j)$, or face j' is a block-born face in one of the two blocks adjacent to face j, see Fig. 2b. The $p^u_{j'j}$ with $j' \in J(j)$, are computed such that:

$$\sum_{j' \in J(j)} p^u_{j'j} = 1. \tag{11.3}$$

This leaves $2^{N-1} - 1$ degrees of freedom for each coarse face j. It follows from (7.2b), (6.1) and (11.3) that the entries of D^c are also given by (4.5). The restriction R^u is chosen as:

$$r^u_{jj'} = 1, \qquad j' \in J(j). \tag{11.4}$$

It follows from (11.3) and (11.4) that (8.1c) holds.

The values of $p^u_{j'j}$, when j' is a block-born face in one of the two blocks i adjacent to j, are computed so that (8.2) holds. Each equation in matrix equation (8.2a) must imply 2^N equations in matrix equation (8.2b); i.e., the 2^N equations in (8.2b) must be all dependent on one equation in (8.2a). It follows from the structure of D^f that the entries of P^u can be chosen such that this dependence holds. The computation of these entries is the same as computing internal flows in an incompressible local boundary value problem on Ω_i. The values of $p^u_{j'j}$, $j' \in J(j)$ serve as Neumann boundary values on one edge of the local region; the boundary values on the other edges are zero. The entries of $\lambda C^f P^{\varphi} E^c_i$ serve as right-hand sides, where λ is chosen such that the incompressible local boundary value problem has a solution.

With this construction of the prolongations and restrictions we have a solution of (8.1) and (8.2). This implies decrease of the functional (4.7) in every coarse-grid correction. There are degrees of freedom in $p^{\varphi}_{i'i}$ and in $p^u_{j'j}$ with $j' \in J(j)$. These degrees of freedom are chosen to speed up convergence. The choice is based on heuristics and numerical experiments, and is as follows: The simplest choice of $p^{\varphi}_{i'i}$ is $p^{\varphi}_{i'i} = 1$, which clearly satisfies (11.1). For the numerical experiments in section 12 we have used:

$$p^{\varphi}_{i'i} \propto \frac{1}{c_{i'}}, \qquad i' \in I(i). \tag{11.5a}$$

If all $c_{i'}$ are zero for $i' \in I(i)$, we set $p^{\varphi}_{i'i} = 1$. If a few but not all $c_{i'}$ are zero, we set $p^{\varphi}_{i'i} = 0$ if $c_{i'} > 0$ and $p^{\varphi}_{i'i} = 2^N / n_z$ if $c_{i'} = 0$, where n_z is the number of zero $c_{i'}$ $(i' \in I(i))$.

The values of $p^u_{j'j}$ with $j' \in J(j)$ are computed so that they are proportional to "permeabilities":

$$p^u_{j'j} \sim \kappa_{j'}; \qquad \kappa_{j'} = \int (W^{-1}\nu_{j'} \cdot \nu_{j'}); \qquad j' \in J(j). \tag{11.5b}$$

Here the integral is over face j' and $\nu_{j'}$ is the unit normal on that face. The proportionality constant follows from (11.3).

To clarify the above, an example of this procedure for the two-dimensional case is given. Consider a coarse-grid face j as in Fig. 2c. We will concentrate on the left block i. The values $p^u_{j'j}$ on the face-born faces are p^u_α and p^u_β. If these are computed from (11.5), we obtain

$$p^u_\alpha = \frac{\kappa_\alpha}{\kappa_\alpha + \kappa_\beta}, \qquad p^u_\beta = \frac{\kappa_\beta}{\kappa_\alpha + \kappa_\beta}.$$

Condition (8.2) boils down to

$$\begin{aligned} (p^u_0 - 0) + (p^u_2 - 0) &= h_0, \\ (p^u_\alpha - p^u_0) + (p^u_3 - 0) &= h_1, \\ (p^u_1 - 0) + (0 - p^u_2) &= h_2, \\ (p^u_\beta - p^u_1) + (0 - p^u_3) &= h_3, \end{aligned} \tag{11.6a}$$

Here $h_i = \lambda c_i p^\varphi_i$. Applying P^{q*} to these expressions is the same as adding them up. The result is

$$\lambda = \frac{p^u_\alpha + p^u_\beta}{\sum_{i=1}^4 c_i p^\varphi_i} = \frac{1}{\sum_{i=1}^4 c_i p^\varphi_i}. \tag{11.6b}$$

If (11.5a) is used for p^φ_i, then $h_i = 2^{-N} = \frac{1}{4}$. We now want to minimise the local contribution to $\langle P^u, WP^u\rangle$ subject to (11.6), which can be written in the form $D^f P_u = Q^f$. This problem is identical to performing a relaxation step for an incompressible problem, but now the *operator* values instead of the solution values have to be found. Denoting the Lagrange multipliers by μ_i, we obtain

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 \\ -1 & 1 & 0 & 0 & w_{00} & 0 & w_{02} & w_{03} \\ 0 & 0 & -1 & 1 & 0 & w_{11} & w_{12} & w_{13} \\ -1 & 0 & 1 & 0 & w_{20} & w_{21} & w_{22} & 0 \\ 0 & -1 & 0 & 1 & w_{30} & w_{31} & 0 & w_{33} \end{pmatrix} \begin{pmatrix} \mu_0 \\ \mu_1 \\ \mu_2 \\ \mu_3 \\ p^u_0 \\ p^u_1 \\ p^u_2 \\ p^u_3 \end{pmatrix} = \begin{pmatrix} \frac{1}{4} \\ \frac{1}{4} - p^u_\alpha \\ \frac{1}{4} \\ \frac{1}{4} - p^u_\beta \\ -w_{0\alpha} p^u_\alpha \\ -w_{1\beta} p^u_\beta \\ 0 \\ -w_{3\alpha} p^u_\alpha - w_{3\beta} p^u_\beta \end{pmatrix}. \tag{11.7}$$

As the matrix on the left-hand side is singular, its first row is replaced by (1 1 1 1 0 0 0 0) and the first entry on the right-hand side is replaced by 0.

If all coefficients $c_{i'}$ and $w_{jj'}$ are constant, and W is diagonal and isotropic, we obtain the standard choice, as in [4]: $p^\varphi_{i'i} = 1$ and $p^u_\alpha = p^u_\beta = \frac{1}{2}$, $p^u_0 = p^u_1 = \frac{1}{4}$, $p^u_2 = p^u_3 = 0$.

Another, more costly approach is the following *nonlinear* method. Let the superscripts pr and pr' refer to values preceding the current value by a number (possibly zero) of multigrid corrections. The heuristics for the choice of P^φ and P^u involve two requirements:

- the prolongations should be a multiple of the total correction made between pr and pr';
- the entries $p^{\varphi}_{i'i}$ of P^{φ} and $p^{u}_{j'j}$ of P^{u} should be non-negative.

Considering these two conflicting requirements, we have constructed the following computational scheme. For P^{φ} introduce

$$\tilde{\lambda} = \sum_{i' \in I(i)} c_{i'}(\varphi^{pr}_{i'} - \varphi^{pr'}_{i'}) / \sum_{i' \in I(i)} c_{i'}(\varphi^{pr}_{i'} - \varphi^{pr'}_{i'})^2,$$

$$\tilde{p}^{\varphi}_{i'i} = \max(0, \tilde{\lambda}(\varphi^{pr}_{i'} - \varphi^{pr'}_{i'})), \tag{11.8a}$$

and the entries $p^{\varphi}_{i'i}$ are given by

$$\lambda = \sum_{i' \in I(i)} c_{i'} \tilde{p}^{\varphi}_{i'i} / \sum_{i' \in I(i)} c_{i'}(\tilde{p}^{\varphi}_{i'i})^2, \qquad p^{\varphi}_{i'i} = \lambda \tilde{p}^{\varphi}_{i'i}. \tag{11.8b}$$

Note that if all of the $c_{i'}$, $i' \in I(i)$, are zero, we set $p^{\varphi}_{i'i} = 1$.

For P^{u} we introduce, in the *two-dimensional* case,

$$\gamma = (u^{pr}_{\alpha} - u^{pr'}_{\alpha})/(u^{pr}_{\beta} - u^{pr'}_{\beta}), \qquad \alpha, \beta \in J(j). \tag{11.9}$$

If $\gamma \geqslant 0$, p^{u}_{α} and p^{u}_{β} follow from (11.3) and $p^{u}_{\alpha} = \gamma p^{u}_{\beta}$. If $-1 \leqslant \gamma < 0$, we set $p^{u}_{\alpha} = 0$ and $p^{u}_{\beta} = 1$. If $\gamma < -1$, we set $p^{u}_{\alpha} = 1$ and $p^{u}_{\beta} = 0$.

12. Numerical convergence results

Quantitative measurements of convergence are indispensable for appreciation of the efficiency of an iterative solution process. The numerical results are given here in the form of spectral radii of the error amplification matrix. For a full appreciation more information is needed, such as aspects on efficient coding, accessibility for vectorisation and/or parallelisation. The latter aspects are not discussed in this paper.

The multigrid performance of the linear and nonlinear operator-weighed methods are evaluated by the spectral radius and the pseudo-spectral radius, respectively, see section 12b. A good computational method should work efficiently in "normal" circumstances and should not deteriorate too much in "exceptional" circumstances. Accordingly, this method was tested on an extensive class of grids, permeabilities and compressibilities. The class for which numerical results are reported is given in section 12a. Numerical values for the radii are in the sections 12c and 12d.

12A. Classes of grids, permeabilities and compressibilities

The class of grids is given in Fig. 3 and the grids are denoted by Roman numerals. The class of permeabilities (denoted by lower-case letters) and the class of compressibilities (denoted by capital letters) use the partitions of the computational domain given in Fig. 4.

The only regular grid in the class is grid I. In grids III and V neighbouring blocks occur with a difference in level of two or more. Grids IV and VI are refinements of these grids such that these differences do not exceed one.

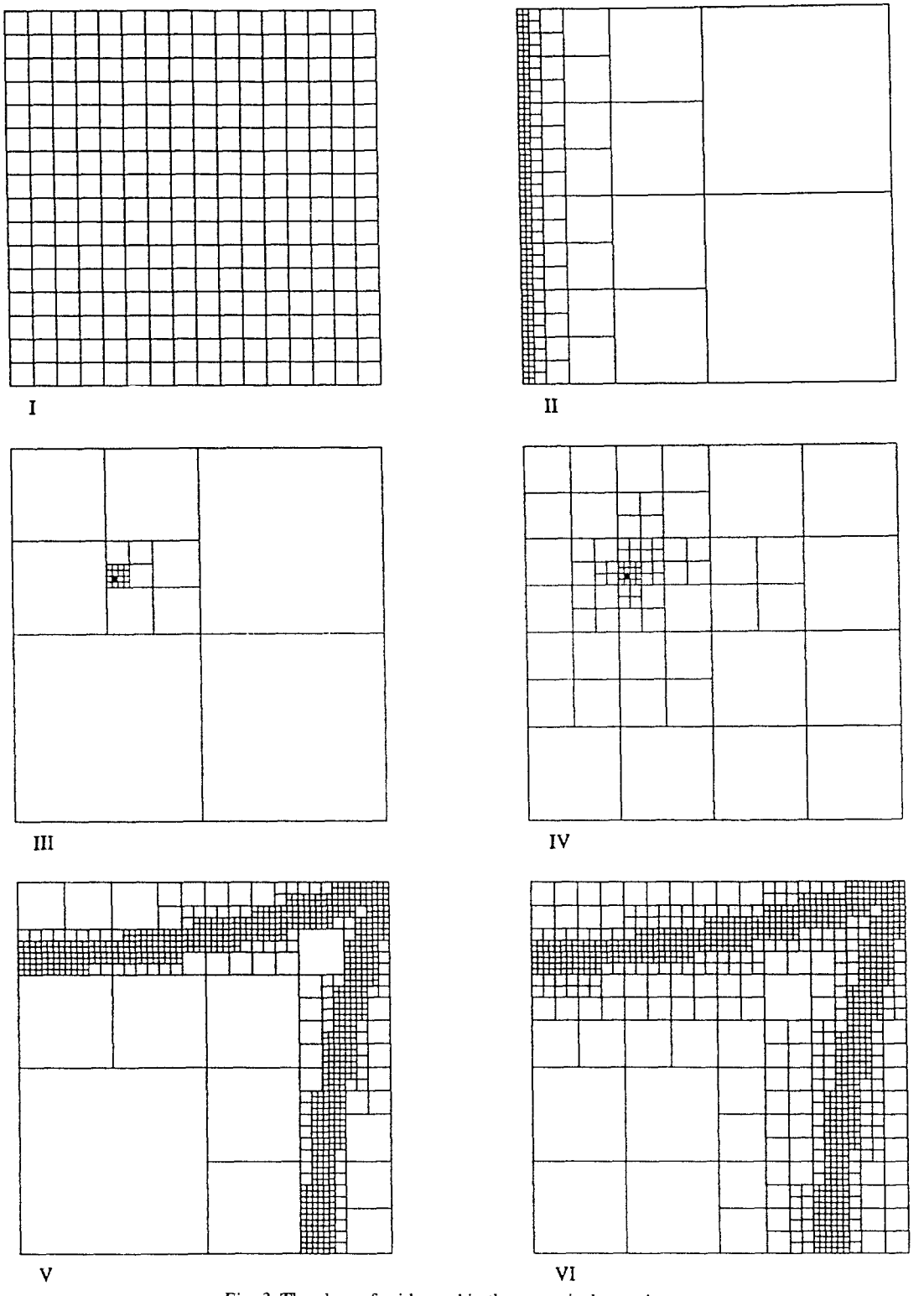


Fig. 3. The class of grids used in the numerical experiments.

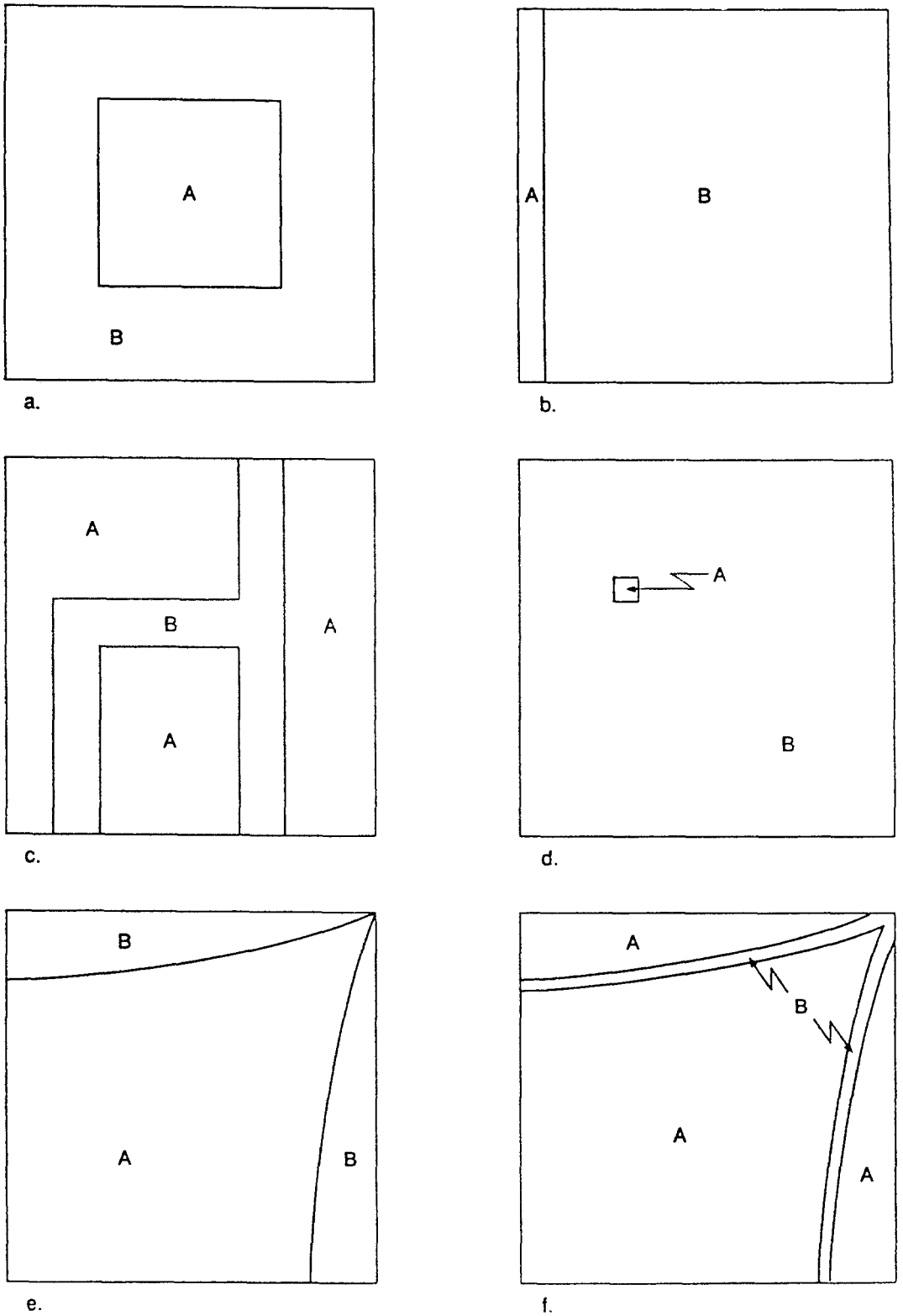


Fig. 4. The class of regions used in the numerical experiments.

The permeabilities ($K = W^{-1}$) are as follows (see Fig. 4):

(a) $K = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ in A, and $K = \lambda \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ in B;

(b) $K = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}$ in A, and $K = \lambda \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}$ in B;

(c) $K = \begin{pmatrix} 4 & 1 \\ 1 & 4 \end{pmatrix}$ in A, and $K = \lambda \begin{pmatrix} 4 & -1 \\ -1 & 4 \end{pmatrix}$ in B;

(d) $K = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ in A, and $K = \lambda \begin{pmatrix} 4 & 1 \\ 1 & 4 \end{pmatrix}$ in B;

(e) $K = \begin{pmatrix} 6 & 1 \\ 1 & 4 \end{pmatrix}$ in A, and $K = \lambda \begin{pmatrix} 3 & 1 \\ 1 & 5 \end{pmatrix}$ in B;

(f) $K = \begin{pmatrix} 6 & 1 \\ 1 & 6 \end{pmatrix}$ in A, and $K = \lambda \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ in B;

Note that each permeability involves a parameter λ. With extreme values of λ, extreme inhomogeneity is covered. Isotropic and anisotropic permeabilities with arbitrary orientations occur, but the anisotropy is moderate.

Figure 4 gives the representation of the partitions of the computational domain $0 \leqslant x \leqslant 1$, $0 \leqslant y \leqslant 1$, except for Fig. 4b: the width of strip A in Fig. 4b is $1/16$ or $1/64$, when the region is covered by grid I or grid II respectively. (This partition is not combined with grids III through VI.) The small blocks of grids III and IV precisely cover part A of Fig. 4d. The boundary between parts A and B in Fig. 4e has been opened up to a strip with width $1/32$ in Fig. 3f. The strip in Fig. 4f is fully covered by the smallest blocks in grids V and VI.

We also consider six compressibility functions, denoted A, ..., F. Compressibility A is defined as:

$c = \sigma$ in part A, and $c = \tau$ in part B,

where parts A and B are given in Fig. 4a. Compressibilities B, D, E, and F are defined similarly, using Eqs. 4b,d,e, and f respectively. The compressibility C is defined as

$$c(x, y) = \sigma + \tau\delta(x - x_0, y - y_0), \tag{12.1}$$

where the position (x_0, y_0) of the δ-function is indicated by a dot in grids III and IV. Note that each compressibility involves the two parameters σ and τ.

12B. THE (PSEUDO-)SPECTRAL RADIUS

An approximation of the spectral radius of the error amplification matrix associated with a multigrid-correction cycle (see section 3e) is computed as follows.

The right-hand sides in (4.6) are set to zero and a random initial guess $x^0 = (\Phi^f, U^f)$ is made. A multigrid correction applied to x^n gives x^{n+1}, and the spectral radius is the limit for $n \to \infty$ of

$$\rho^n = \|x^{n+1}\| / \|x^n\|. \tag{12.2}$$

In many cases the ρ^n have converged for $n = 30$; in other cases the ρ^n are oscillating with a period usually less than 5. To eliminate the oscillations we introduce

$$\tilde{\rho}^0 = 1, \qquad \tilde{\rho}^n = \frac{1}{5}\rho^n + \frac{4}{5}\tilde{\rho}^{n-1}, \tag{12.3}$$

which also converge to the spectral radius, but with fewer oscillations. All numerical values in section 12c are $\tilde{\rho}^{30}$.

For the nonlinear approach of section 11, we introduce the pseudo-spectral radius as follows. Let a field (Φ_e^f, U_e^f) be given and the related right-hand sides be computed from (4.6). The multigrid process is started with initial values (Φ^f, U^f) equal to zero, and continued until

$$\{\langle C^f(\Phi^f - \Phi_e^f), \Phi^f - \Phi_e^f\rangle + \langle W^f(U^f - U_e^f), U^f - U_e^f\rangle\}^{\frac{1}{2}} \leqslant \varepsilon\{\langle C^f \Phi_e^f, \Phi_e^f\rangle + \langle W^f U_e^f, U_e^f\rangle\}^{\frac{1}{2}}. \tag{12.4}$$

Note that the terms in (12.4) can be interpreted as energy norms; the inequality states that the energy norm of the error is at most a fraction ε of the energy norm of the solution. If n multigrid corrections are needed to reach (12.4), this is related to a pseudo-spectral radius

$$\rho_{ps} = (\varepsilon)^{\frac{1}{n}}. \tag{12.5}$$

The fields Φ_e^f and U_e^f have been computed as follows. In the right-hand side of (4.6) we take $G^f = 0$ and Q^f as related to a point source $+1$ at $(x, y) = (0^+, 0^+)$ and a point source -1 at $(x, y) = (1^-, 1^-)$. We take random initial values for Φ^f and U^f, and do a few (say 5) multigrid corrections. The resulting field is (Φ_e^f, U_e^f).

12c. Linear method

Here we give numerical values for $\tilde{\rho}^{30}$, introduced in section 12b. The choice of the free parameters in the prolongations is the one in (11.5). Just one smoothing sweep is made after each prolongation.

Each combination of grid, permeability and compressibility, e.g. IIIdC, involves the three parameters λ, σ, and τ. These parameters run through the values

$$\lambda = 10^{-4}, 10^{-2}, 1, 10^2, 10^4; \qquad \sigma, \tau = 0, 1, 10^3. \tag{12.6}$$

$\tilde{\rho}^{30}$ has been computed for each combination of these parameter values. Table 1 gives the average and the maximum of these 45 values for a number of combinations of grids, permeabilities and compressibilities. The average values of the spectral radii in Table 1 allow the conclusion that the method gives, on the whole, rapid convergence. Cases in which the method

Table 1. Spectral radii

Combination			Average	Maximum	Max. attained for		
					λ	σ	τ
I	a	A	0.22	0.75	10^{-4}	0	10^3
		B	0.16	0.39	10^{+4}	0	10^3
		C	0.15	0.36	10^{+4}	10^3	0
	b	A	0.24	0.86	10^{-4}	1	1
		B	0.24	0.91	10^{-4}	1	10^3
		C	0.22	0.86	10^{-4}	1	0
	c	C	0.21	0.65	10^{-4}	1	0
	d	C	0.13	0.20	10^{-4}	10^3	0
	e	E	0.22	0.68	10^{-4}	0	10^3
II	b	B	0.38	0.92	10^{-4}	1	10^3
		C	0.38	0.87	10^{-4}	1	0
III	d	C	0.17	0.44	10^{-4}	10^3	0
		D	0.19	0.59	10^{-4}	10^3	1
IV	d	C	0.43	0.54	10^{+4}	0	0
		D	0.44	0.67	10^{-4}	0	10^3
V	e	E	0.52	0.995	10^{-4}	10^3	10^3
		F	0.54	0.995	10^{-4}	10^3	0
	f	E	0.36	0.39	10^{-4}	0	10^3
		F	0.36	0.37	10^{-4}	0	0
VI	e	E	0.34	0.91	10^{-4}	0	10^3
		F	0.31	0.53	10^{-4}	10^3	0
	f	E	0.23	0.24	10^{-4}	0	1
		F	0.23	0.24	10^{-4}	0	0

deteriorates involve unfavourable contrasts in both the permeability and the compressibility; all computed spectral radii for the incompressible case ($\sigma = \tau = 0$) are less than 0.62.

Spectral radii for the same computational problems, but corresponding to the method of [4] (no operator weighing), are to be found in Table 2. Note that the operator weighing introduced here improves the average spectral radii considerably and eliminates the many cases with inadmissibly slow convergence in the previous method. The exceptional cases with slow convergence in the new method can be improved using the nonlinear method, as will be seen in the next section.

12D. NONLINEAR METHOD

In the nonlinear method, we first perform n_{sk} multigrid corrections with prolongation and restriction operators based on the linear method. Next, the nonlinear approach is used. The superscript pr now refers to the current value and the superscript pr' refers to the value preceding that value by n_{sk} multigrid corrections.

Table 2. Spectral radii for the method in [4]

Combination			Average	Maximum	Max. attained for		
					λ	σ	τ
I	a	A	0.50	0.9994	10^{+4}	1	0
		B	0.57	0.9994	10^{+4}	1	1
		C	0.43	0.9994	10^{+4}	1	0
	b	A	0.54	0.9419	10^{-4}	1	1
		B	0.65	0.9505	10^{-4}	0	10^3
		C	0.51	0.9419	10^{-4}	1	0
	c	C	0.51	0.8776	10^{+4}	10^3	0
	d	C	0.44	0.9967	10^{+4}	1	0
	e	E	0.65	0.9888	1	0	1
II	b	B	0.90	0.9999	10^{-2}	0	10^3
		C	0.78	0.9988	10^{+4}	0	0
III	d	C	0.46	0.9929	10^{+4}	10^3	0
		D	0.78	0.9999	10^{-2}	0	1
IV	d	C	0.61	0.9963	10^{+4}	0	0
		D	0.73	0.9963	10^{+4}	0	0
V	e	E	0.95	0.9999	1	1	0
		F	0.86	0.9965	10^{-4}	10^3	0
	f	E	0.78	0.9999	10^{-4}	1	0
		F	0.36	0.3742	10^{-4}	0	0
VI	e	E	0.86	0.9972	1	0	1
		F	0.75	0.9919	10^{+4}	1	0
	f	E	0.65	0.9973	10^{-4}	0	1
		F	0.24	0.2409	10^{-4}	0	0

Pseudo-spectral radii have been computed with $\varepsilon = 10^{-4}$ and $n_{sk} = 5$. The grids, permeabilities and compressibilities, and also the parameters λ, σ and τ, run through the classes, or values, respectively, as in section 12c. Table 3 gives, for each combination of grid, permeability and compressibility, the average and the maximum over λ, σ and τ of the value of ρ_{ps}.

Comparison of these results with Table 1 shows that the rapid convergence on the whole is slightly improved, though this is probably not worth the additional computational work associated with the extra sweeps for the coarse-grid equations. The cases with very slow convergence in Table 1, e.g. VeE and VeF, are much improved in Table 3; the maximum value of all pseudo-spectral radii in Table 3 is 0.78.

The results of sections 12c and 12d point to the following method. The computational solution of (4.6) starts with prolongations and coarse-grid equations based on a priori information (permeabilities and compressibilities). When, according to some criterion, the process has converged within a few (say, 5) multigrid corrections, the solution has been found and the process stops. Otherwise, the prolongations and coarse-grid equations are updated with

Table 3. Pseudo-spectral radii for the nonlinear method

Combination			Average	Maximum	Max. attained for		
					λ	σ	τ
I	a	A	0.16	0.63	10^{-4}	0	10^3
		B	0.10	0.41	10^{+4}	10^3	10^3
		C	0.09	0.39	10^{+4}	10^3	0
	b	A	0.13	0.49	10^{-4}	1	1
		B	0.13	0.63	10^{-4}	1	10^3
		C	0.11	0.49	10^{-4}	1	10^3
		C	0.15	0.48	10^{-4}	1	0
	d	C	0.08	0.09	10^{+2}	10^3	0
	e	E	0.13	0.46	10^{-4}	1	10^3
II	b	B	0.25	0.74	10^{-4}	1	1
		C	0.26	0.71	10^{-4}	1	0
III	d	C	0.11	0.52	10^{-4}	10^3	10^3
		D	0.12	0.51	10^{-4}	10^3	10^3
IV	d	C	0.32	0.58	10^{-4}	10^3	1
		D	0.34	0.67	10^{-4}	0	10^3
V	e	E	0.31	0.78	10^{-4}	1	10^3
		F	0.30	0.74	10^{-4}	10^3	0
	f	E	0.14	0.14	10^{-4}	10^3	10^3
		F	0.14	0.14	10^{-4}	10^3	0
VI	e	E	0.20	0.70	10^{-4}	1	10^3
		F	0.16	0.43	10^{-4}	1	0
	f	E	0.09	0.09	10^{-4}	0	10^3
		F	0.09	0.09	10^{-4}	10^3	0

the use of the current and previous approximations of the solution and the process continues, with occasional updates of prolongations and coarse-grid equations, until a sufficient level of convergence has been reached.

13. Concluding remarks

The results of this paper can be summarised as follows:

1. The multigrid method has been formulated in terms of discretisations of first-order equations on all levels. Prolongations and restrictions are introduced for scalar- and vector-fields separately.
2. Constraints for prolongations and restrictions have been presented, that lead to a very robust multigrid method. Examples for prolongations and restrictions that satisfy these constraints have been given.
3. It may be expected that these constraints are also necessary conditions for convergence for all possible grids and all possible values of the coefficient functions. This is suggested

by numerical experiments (not given here) which show divergence on violation of the constraints.

4. Local refinement is embedded naturally in the method. Smoothing on regular parts of the grid only is sufficient.
5. The emphasis on robustness in "general" circumstances, a necessity for industrial applications, has resulted in a rather complicated computational scheme. Consequently, in "moderate" circumstances (smooth or mildly discontinuous coefficient functions) the equations are solved with probably less computational work by use of the simple prolongations in [4] or by other methods, e.g. [12], [13], for regular grids.
6. The case of strong anisotropy in the permeability has not been discussed here. Line relaxation may be necessary in that case.
7. The set of numerical experiments shows that operator weighing as introduced here improves the convergence rate considerably. Acceptable convergence rates are obtained in many cases for which the non-weighed method displays inadmissibly slow convergence.

Application of this method to problems in two-phase porous media flow presented elsewhere [7], [8], shows that a reduction of the residual by 5 orders of magnitude is typically reached within a few (3 to 6) V-cycles. During time stepping, typically 1 or 2 cycles are needed to update the total flow field from earlier time.

Appendix A: Some derivations

We have to prove (9.4). By (9.2):

$$P^{q*}(C^f\tilde{\Phi}^f + D^f\tilde{U}^f - Q^f)$$
$$= P^{q*}C^f(\Phi^f + P^{\varphi}(\tilde{\Phi}^c - \Phi^c)) + P^{q*}D^f(U^f + P^u(\tilde{U}^c - U^c)) - P^{q*}Q^f$$

By (7.2a,b) and (8.1a) this equals:

$$P^{q*}C^f\Phi^f + C^c(\tilde{\Phi}^c - \Phi^c) + P^{q*}D^fU^f + D^c(\tilde{U}^c - U^c) - P^{q*}Q^f$$

By (9.3) this equals:

$$P^{q*}C^f\Phi^f - C^c\Phi^c + P^{q*}D^fU^f - D^cU^c + Q^c - P^{q*}Q^f$$

By (7.3a) this equals:

$$-C^c\Phi^c + C^cR^{\varphi}\Phi^f + P^{q*}D^fU^f - D^cU^c$$

By (9.1a,b) this equals:

$$P^{q*}D^fU^f - D^cR^uU^f = 0$$

by (8.1c).

Proof of (7.1a). By (9.1a,b) and (7.3a) we have:

$$C^c\Phi^c + D^cU^c - Q^c = C^cR^{\varphi}\Phi^f + D^cR^uU^f - P^{q*}(Q^f - C^f\Phi^f) - C^cR^{\varphi}\Phi^f$$

By (8.1c) this equals:

$$P^{q*}(D^fU^f - Q^f + C^f\Phi^f) = 0$$

by (4.6a).

Proof of (10.1). By (4.6a) we have to prove:

$$C^f(\tilde{\Phi}^f - \Phi^f) + D^f(\tilde{U}^f - U^f) = 0$$

Hence we have to prove by (9.2a,b):

$$C^f P^{\varphi}(\tilde{\Phi}^c - \Phi^c) + D^f P^u(\tilde{U}^c - U^c) = 0$$

Hence we have to prove by (8.2):

$$P^{q*}(C^f P^{\varphi}(\tilde{\Phi}^c - \Phi^c) + D^f P^u(\tilde{U}^c - U^c)) = 0$$

This follows from (7.2a,b), (8.1a), (7.1a) and (9.3).

Proof of (10.2). We have to show that:

$$\begin{aligned}&\langle C^f\tilde{\Phi}^f, \tilde{\Phi}^f\rangle + \langle W^f\tilde{U}^f - 2G^f, \tilde{U}^f\rangle - \langle C^f\Phi^f, \Phi^f\rangle - \langle W^fU^f - 2G^f, U^f\rangle\\ &\quad -\langle C^c\tilde{\Phi}^c, \tilde{\Phi}^c\rangle - \langle W^c\tilde{U}^c - 2G^c, \tilde{U}^c\rangle + \langle C^c\Phi^c, \Phi^c\rangle + \langle W^cU^c - 2G^c, U^c\rangle\end{aligned}$$

equals zero. By (7.3b) this equals:

$$\begin{aligned}&\langle C^f\tilde{\Phi}^f, \tilde{\Phi}^f\rangle + \langle W^f\tilde{U}^f, \tilde{U}^f\rangle - \langle C^f\Phi^f, \Phi^f\rangle - \langle W^fU^f, U^f\rangle\\ &\quad -\langle C^c\tilde{\Phi}^c, \tilde{\Phi}^c\rangle - \langle W^c\tilde{U}^c, \tilde{U}^c\rangle + \langle C^c\Phi^c, \Phi^c\rangle + \langle W^cU^c, U^c\rangle\\ &\quad +2\langle G^f, U^f - \tilde{U}^f + P^u(\tilde{U}^c - U^c)\rangle + 2\langle P^{u*}W^fU^f - W^cU^c, U^c - \tilde{U}^c\rangle\end{aligned}$$

By (9.2b) this equals:

$$\begin{aligned}&\langle W^fP^u(U^c - \tilde{U}^c), P^u(U^c - \tilde{U}^c)\rangle - \langle W^c(U^c - \tilde{U}^c), (U^c - \tilde{U}^c)\rangle\\ &\langle C^f\tilde{\Phi}^f, \tilde{\Phi}^f\rangle - \langle C^f\Phi^f, \Phi^f\rangle - \langle C^c\tilde{\Phi}^c, \tilde{\Phi}^c\rangle + \langle C^c\Phi^c, \Phi^c\rangle\end{aligned}$$

The first line is zero due to (7.2c). By (9.2a) this equals:

$$\langle C^fP^{\varphi}(\tilde{\Phi}^c - \Phi^c), P^{\varphi}(\tilde{\Phi}^c - \Phi^c)\rangle + 2\langle C^f\Phi^f, P^{\varphi}(\tilde{\Phi}^c - \Phi^c)\rangle - \langle C^c\tilde{\Phi}^c, \tilde{\Phi}^c\rangle + \langle C^c\Phi^c, \Phi^c\rangle$$

By (7.2a) this equals:

$$2\langle C^c\Phi^c - P^{\varphi*}C^f\Phi^f, \Phi^c - \tilde{\Phi}^c\rangle$$

By (9.1a) and (8.1b) this equals zero.

References

1. P.W. Hemker and J. Molenaar, An adaptive multigrid approach for the solution of the 2D semiconductor equations. In: W. Hackbuch and U. Trottenberg (eds.), *Multigrid Methods III*. Basel/Boston/Berlin, Birkhäuser Verlag (1991) 41–60.
2. J. Molenaar, A two-grid analysis of the combination of mixed finite elements and Vanka-type relaxation. In: W. Hackbuch and U. Trottenberg (eds.), *Multigrid Methods III*. Basel/Boston/Berlin, Birkhäuser Verlag (1991) 313–324.
3. S.P. Vanka, Block-implicit multigrid solution of Navier-Stokes equations in primitive variables. *J. Comput. Phys.* 65 (1986) 138–158.
4. G.H. Schmidt and F.J. Jacobs, Adaptive local grid refinement and multigrid in numerical reservoir simulation. *J. Comput. Phys.* 77 (1988) 140–165.

5. P.A. Raviart and J.M. Thomas, A mixed finite element method for second orderd elliptic problems. In: *Mathematical Aspects of Finite Element Methods*, Lecture Notes in Mathematics Vol. 606, Berlin/Heidelberg/New York, Springer-Verlag (1977) 292–302.
6. W. Hackbusch, *Multi-Grid Methods and Applications*, Computational Mathematics Vol. 4, Berlin/Heidelberg/New York, Springer-Verlag (1985).
7. R.H.J. Gmelig Meyling, W.A. Mulder and G.H. Schmidt, Porous media flow on locally refined grids. In: T. Verheggen (ed.), *Proceedings of the Workshop on Numerical Methods for the Simulation of Multi-Phase and Complex Flow*. Berlin/Heidelberg/New York, Springer-Verlag (1992).
8. W.A. Mulder and R.H.J. Gmelig Meyling, Numerical simulation of two-phase flow using locally refined grids in three-space dimensions, SPE 21230. In: *11th SPE Symposium on Reservoir Simulation*, Anaheim, California, Feb. 17–20, 1991. Also published as: SPE Advanced Technology Series 1 (1993) 36-41.
9. G.H. Schmidt, A dynamic grid generator and multigrid for numerical fluid dynamics. In: P. Wesseling (ed.), *Proceedings of the 8th GAMM Conference on Numerical Methods in Fluid Mechanics*, Notes in Numerical Fluid Mechanics Vol. 29, Braunschweig, Vieweg, (1990) 493–502.
10. G.H. Schmidt and E. de Sturler, Multigrid for porous media flow on locally refined three dimensional grids, GMD-Studien Nr. 189, GMD St. Augustin, (1991).
11. A. Brandt, Multi-level adaptive solutions to boundary value problems, *Math. Comput.* 31 (1977) 333–390.
12. R. Kettler, Analysis and comparison of relaxation schemes in robust multigrid and preconditioned conjugate gradient methods, Lecture Notes in Math. Vol. 960, Berlin/Heidelberg/New York, Springer-Verlag (1982) 502–534.
13. P. Wesseling, Cell-centered multigrid for interface problems, *J. Comput. Phys.* 79 (1989) 85–91.