JOURNAL OF COMPUTATIONAL PHYSICS 77, 140–165 (1988)

Adaptive Local Grid Refinement and Multi-grid in Numerical Reservoir Simulation

G. H. SCHMIDT AND F. J. JACOBS

*Koninklijke/Shell Exploratie en Produktie Laboratorium, Rijswijk, The Netherlands**

Received May 9, 1986; revised August 5, 1987

Flexible gridding has the potential of increasing the accuracy in numerical simulation of flow through hydrocarbon reservoirs within limitations in computing time and memory space. The adaptive gridding method presented here follows general concepts outlined by Brandt (*Math. Comput.* **31**, 333 (1977)). The pressure equation is discretised on locally refined grids with mixed finite elements. A multi-grid solution method is stated for the (indefinite) set of equations in the pressure and its gradient (or flow). A proof of convergence of the multi-grid process is given and convergence rates are discussed. The method has been tested on a set of equations representative for reservoir simulation. Results obtained for a five spot example are shown. © 1988 Academic Press, Inc.

1. INTRODUCTION

The computational work in numerical simulation of fluid flow in hydrocarbon-reservoirs is composed of two parts:

(a) *The flash calculations.* These are evaluations of thermodynamic equilibrium between fluid components and require the solution of a set of (strongly) non-linear algebraic equations.

(b) *The transport calculations.* These consist of the solution of the conservation equations which govern the motion of fluid through the reservoir and form a coupled system of partial differential equations; where elliptic (or, for compressible fluids, parabolic) and hyperbolic types occur simultaneously.

The choice of a computational method is strongly influenced by the possible occurrence of (saturation) shocks propagating through the reservoir. Combining the need for an accurate approximation of these moving features with the need to economise on computational work points to a numerical method based on adaptive

* Acknowledgment is made to Shell Research B.V. Neither Royal Dutch Petroleum Company nor The "SHELL" Transport and Trading Company p.l.c. nor any company of the Royal Dutch/Shell Group will accept any liability for loss or damage originating from the use of the information containing therein. Although SHELL companies have their own separate identities the expressions "SHELL" and "GROUP" are used for convenience to refer to companies of the Royal Dutch/Shell Group in general, or to one or more such companies as the context may require.

0021-9991/88 $3.00

Copyright © 1988 by Academic Press, Inc.
All rights of reproduction in any form reserved.

local mesh refinement. Local refinement clearly reduces the work in (a), as compared with regular grids. On the other hand, the work in (b), which consists mainly of the solution (at each timestep) of the large set of (linear) equations representing the discretisation of the elliptic part, can be performed more efficiently on regular grids than on locally refined grids.

Regularity and local refinement are optimally balanced in the method outlined by Brandt [5]. Here locally refined grids are effected by dynamically "switching on and off" the individual grid points in a regular grid structure. On such grids the equations in (b) can be solved efficiently by the multi-grid technique.

Use of the IMPEC (implicit in pressure, explicit in components) time discretisation of the transport equations allows, to some degree, separate treatment of the elliptic or parabolic (pressure) equation and the hyperbolic (component) equations. The moving features that require dynamic local refinement are dictated by the hyperbolic part. Since we use one discretisation grid for all fields, the locally refined grid is used in the elliptic part too. In this paper we describe a solution method for the elliptic part, which suits the locally refined grid with respect to datastructure and computational efficiency.

In order to keep the data structure on the locally refined grid manageable, the approximation of the fields are taken of low order. Low-order approximation can be consistent only if the differential equations that are discretised are of low order too. Therefore we write the elliptic part as a set of first-order differential equations and approximate with the lowest order mixed finite elements of Raviart and Thomas [13]. The advantage of the mixed finite element approximation, notably the direct approximation of the flow field, has already been demonstrated for reservoir simulation in [7–10]. The combination of mixed finite elements with local refinement distinguishes the approach in this paper from these applications and from other methods with local refinement [2, 3, 11].

The set of equations resulting from the mixed finite element approximation is not positive-definite. We present a multi-grid method for its solution. Convergence of this multi-grid process is proved. Numerical results show convergence rates comparable with rates of multi-grid methods for the positive-definite discretisation of the pressure equation, as reported in, e.g., [1, 4, 6, 12].

The elliptic and hyperbolic equations give rise to different strategies for (un)refining the grid. Numerical results show that, for the type of grids used here, relatively simple strategies suffice.

All numerical results were obtained with a prototype reservoir simulator, based on very simple flash calculations. The paper is restricted to two space dimensions, but extending the ideas to three dimensions is straightforward.

2. The Equations of Reservoir Simulation

In this section we state the basic equations underlying the prototype simulator, for which numerical results are given in the final section. The equations model

incompressible multi-phase flow. We also use them to illustrate the IMPEC scheme and the split into an hyperbolic and an elliptic part. The equations are

$$\frac{\partial s_n}{\partial t} = q_n - \nabla \mathbf{u}^n \qquad n = 1, ..., N \tag{2.1a}$$

$$\mathbf{u}^n = -k_n \nabla p \qquad n = 1, ..., N \tag{2.1b}$$

$$\sum_{n=1}^{N} s_n = 1 \tag{2.1c}$$

for $\mathbf{x} = (x, y)$ in a fixed domain Ω and $t \geqslant 0$. The primary unknowns are: saturations s_n, flows $\mathbf{u}^n$ $(n = 1, ..., N)$, and pressure p. The source terms q_n will be assumed to be known (rate-constrained wells) and the permeabilities k_n are given functions of the saturations and the space-coordinates. Though this approach allows the permeabilities to be tensors, here we assume them to be scalars. Initial saturations are given

$$s_n = s_n^0 \qquad t = 0, n = 1, ..., N \tag{2.2}$$

and at the boundary no-normal-flow conditions hold:

$$\mathbf{u}^n \cdot \mathbf{v} = 0 \qquad \mathbf{x} \in \partial\Omega, n = 1, ..., N, \tag{2.3}$$

where $\mathbf{v}$ is the unit outward normal.

Introducing

$$\mathbf{u} = \sum_{n=1}^{N} \mathbf{u}^n; \qquad k = \sum_{n=1}^{N} k_n; \qquad q = \sum_{n=1}^{N} q_n, \tag{2.4}$$

where k is supposed to be positive-definite, we can write (2.1) as:

$$\frac{\partial s_n}{\partial t} = q_n - \nabla \mathbf{u}^n \qquad n = 1, ..., N, \tag{2.5a}$$

$$\mathbf{u}^n = k_n k^{-1} \mathbf{u} \qquad n = 1, ..., N, \tag{2.5b}$$

$$\nabla \mathbf{u} = q \tag{2.5c}$$

$$\nabla p + k^{-1} \mathbf{u} = 0. \tag{2.5d}$$

Equations (2.5a), (2.5b) form the *hyperbolic part*, and (2.5c), (2.5d) form the *elliptic part*. There is coupling only through the occurrence of the total flow $\mathbf{u}$ in the hyperbolic part, and of the saturations (via the total permeability k) in the elliptic part.

The numerical discretisation of system (2.5) starts with the introduction of a class of grids in Section 3. Discretisation of the elliptic part on a fixed grid and its solution by a multi-grid method are the subjects of Sections 4 and 5. The dis-

cretisation of the hyperbolic part is "one-point-upstream." It is given in Section 6, where the grid-updating is analysed.

For compressible flow, Eqs. (2.5c), (2.5d) become of parabolic character, since a term $c\,\partial p/\partial t$ (with $c>0$) appears in the left-hand side of (2.5c). We include compressibility in the discussion of the multi-grid method, although numerical results will be presented only for $c=0$.

3. Grids and Data Structure

In this section we define the class of grids that will be used, outline how field quantities discretised on these grids are stored, and introduce pointers needed to code a computer program.

The coarsest grid in the class is fixed during simulation time and consists of a rectangular, possibly not equidistant, grid with interfaces parallel to the x- and y-axes of a Cartesian coordinate system. The total class of grids consists of this coarsest grid and all grids that can be obtained from it by the *basic refinement*, which refines one block into four identical smaller ones (see Fig. 1).

All blocks and interfaces in the coarsest grid have *level* one and the basic refinement creates blocks and interfaces of one level higher. When at some time during the simulation the grid for discretisation of (2.5) is as shown in Fig. 2a, a series of coarser grids for the multi-grid process is created by unrefining sequentially all blocks of the highest level, see Figs. 2b–f. The level of each of these grids is defined as the highest level occurring among its composing blocks. Storage is effected by numbering linearly all blocks of level 1, of level 2, etc. All x-interfaces (i.e., interfaces perpendicular to the x-axis) are also numbered linearly as are the y-interfaces. Though required for consistent notation, subscripts x or y for the interfaces are suppressed in this paper.

Blocks and interfaces in the discretisation grid are called *fine*; all blocks and interfaces in the other grids, which do not occur in the discretisation grid, are called *coarse*. Note that there can be fine blocks of all levels.

When the side of a fine block consists of two or more fine interfaces, the side and interfaces are called *green* [2]; e.g., in Fig. 3b there are two green interfaces and one

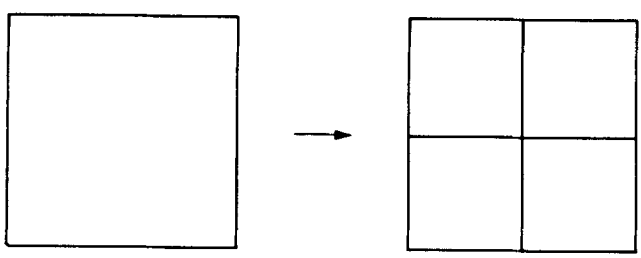

FIG. 1. The basic refinement.

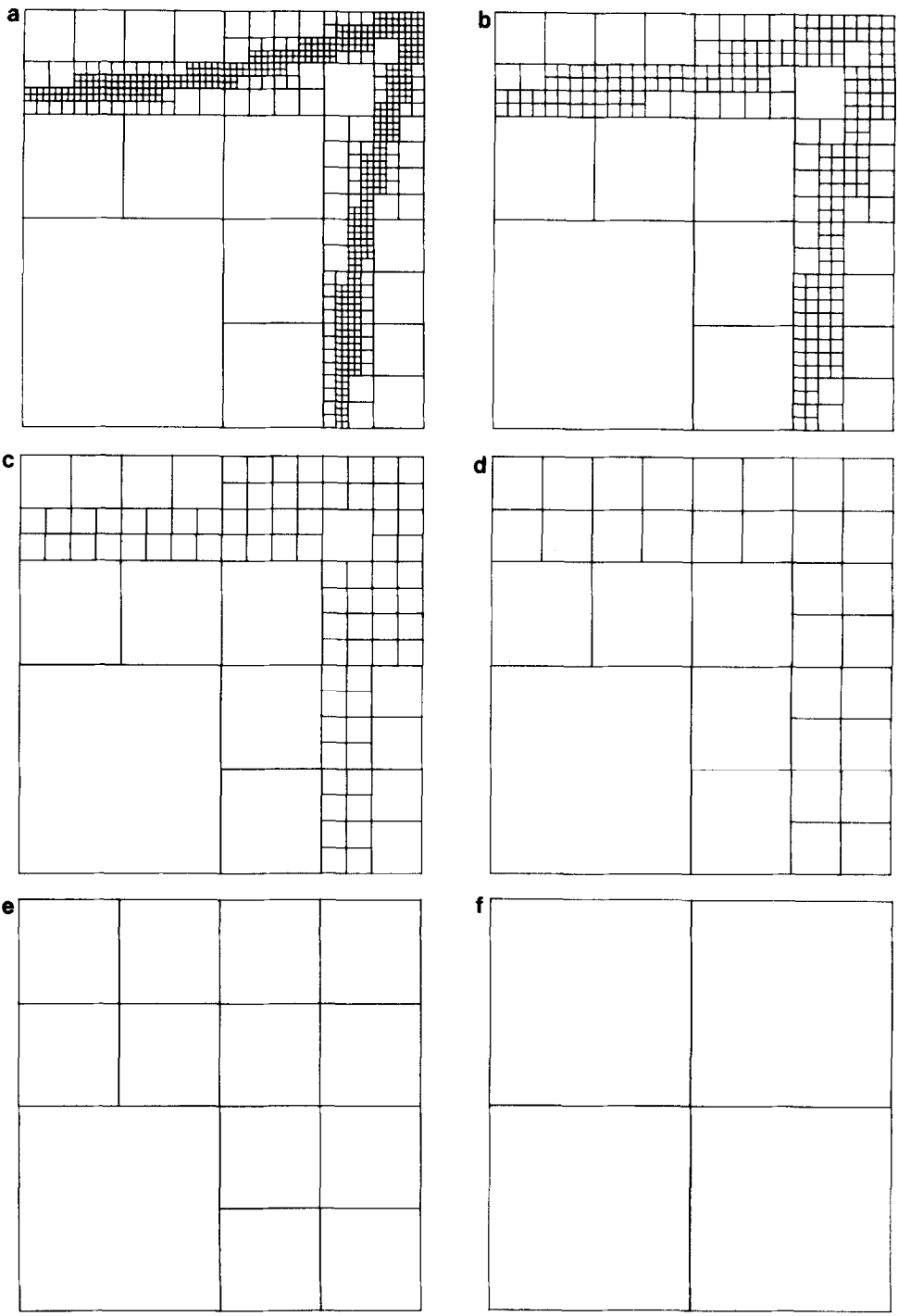

FIG. 2. The discretisation grid (a) and the coarser grids to be used in the multi-grid process.

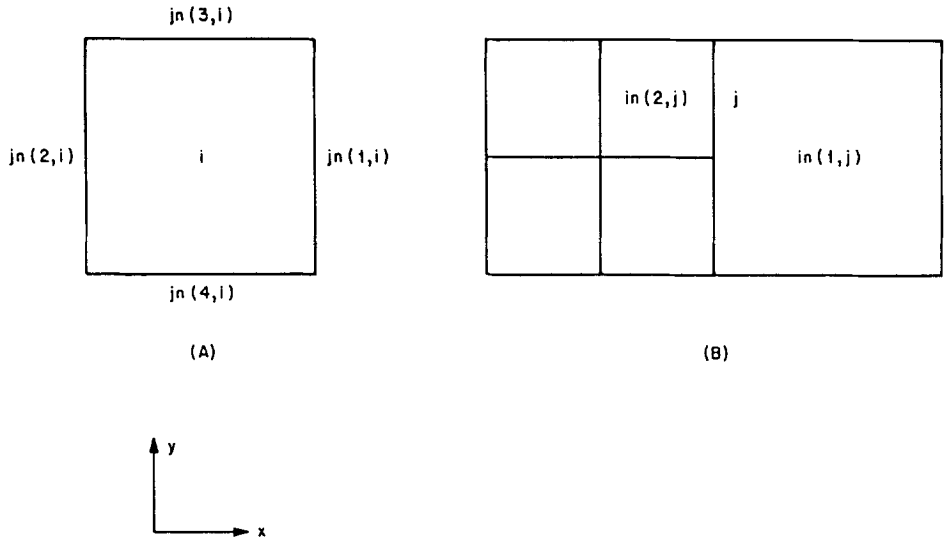

FIG. 3. Neighbour pointers *jn* and *in*.

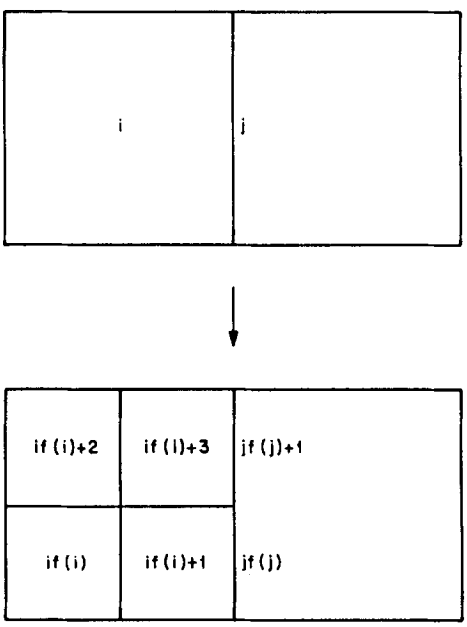

FIG. 4. Descendant pointers *if* and *jf*.

green side. A point common to four blocks of the same level is called a *regular node.* If the four adjoining blocks are not of the same level, we call the node *green.*

The pointers are: *Neighbour pointers*:

— $jn(1, i), ..., jn(4, i)$ are the numbers of the four interfaces, of the same level, neighbouring to block i, in the order indicated in Fig. 3a.

— $in(1, j), in(2, j)$ are the numbers of the two blocks neighbouring to interface j, in the order indicated in Fig. 3b.

The neighbours $jn(1, i)$, $jn(3, i)$, and $in(1, j)$ are called *positive* neighbours, the others are *negative* ones.

Descendant pointers:

— $if(i), ..., if(i)+3$ are the four sons of block i.

— $jf(j), jf(j)+1$ are the two sons of interface j.

The order of the sons is indicated in Fig. 4. For fine blocks and interfaces we have, respectively, $if(i)=0$ and $jf(j)=0$. An interface, created by refinement, is part of a coarse interface (i.e., "interface-born"), or it is in the middle of a coarse block (i.e., "block-born"). Only the interface-born interfaces are in the range of the pointer jf.

4. Mixed Finite Element Discretisation

For discretisation of the elliptic part on grids in the class of Section 3 (e.g., Fig. 2a) we introduce the lowest order mixed finite element discretisation by Raviart and Thomas [13].

The elliptic part consists of

$$c\phi + \nabla \mathbf{u} = q \tag{4.1}$$

$$\nabla \phi + w\mathbf{u} = 0 \tag{4.2}$$

for $x \in \Omega$, with

$$\mathbf{u} \cdot \mathbf{v} = 0 \tag{4.3}$$

for $x \in \partial\Omega$. Here w (k^{-1} of (2.5)) is assumed to be positive and bounded away from zero. The compressibility c is either positive and bounded away from zero or $c \equiv 0$. For $c \equiv 0$, q satisfies $\int_\Omega q = 0$ and $\int_\Omega \phi = 0$ is assumed to ensure uniqueness.

With introduction of the spaces $L_2(\Omega)$,

$$H = \{\mathbf{u} \mid u_x, u_y, \nabla \mathbf{u} \in L_2(\Omega), \mathbf{u} \cdot \mathbf{v} = 0 \text{ on } \partial\Omega\} \tag{4.4}$$

with norm

$$\|\mathbf{u}\|_H^2 = |\mathbf{u}|_{L_2}^2 + |\nabla \mathbf{u}|_{L_2}^2, \tag{4.5}$$

and the product space $X = L_2 * H$, and under the assumption that q is sufficiently smooth, equivalent variational formulations of (4.1)–(4.3) hold.

For $c > 0$, the unique minimum of

$$f(\phi, \mathbf{u}) = \int_\Omega (c\phi^2 + w\mathbf{u} \cdot \mathbf{u}) \tag{4.6}$$

in the closed affine subspace of X

$$X_q = \{(\phi, \mathbf{u}) \in X \mid (4.1) \text{ holds}\} \tag{4.7}$$

is attained as the unique solution of (4.1)–(4.3).

For $c = 0$, the unique minimum of $f(\mathbf{u})$ is the closed affine subspace of H

$$H_q = \{\mathbf{u} \in H \mid (4.1) \text{ holds}\} \tag{4.8}$$

is attained in a point $\mathbf{u}$ to which a unique ϕ satisfying (4.2) and $\int_\Omega \phi = 0$ may be associated, so that $(\phi, \mathbf{u})$ is the unique solution of (4.1)–(4.3).

In the sequel we will switch from the differential formulation (or later, after discretisation, from the difference formulation) to the minimisation formulation and vice versa as need prescribes.

For the discretisation of scalar quantities we introduce $\Psi_i(x, y)$ as the indicator function of block i. For vector quantities we use $\Theta_j(x, y)$: the direction of $\Theta_j(x, y)$ is that of the x- or y-axis (perpendicular to interface j); its length is 1 on interface j, 0 on the others, and is obtained by linear interpolation in the x- or y-direction in the blocks, see Fig. 5. The known quantities are discretised by

$$c_i = a_i^2 \left(\int_\Omega \frac{1}{c} \Psi_i \right)^{-1} \qquad c > 0 \tag{4.9a}$$

$$= 0 \qquad c = 0 \tag{4.9b}$$

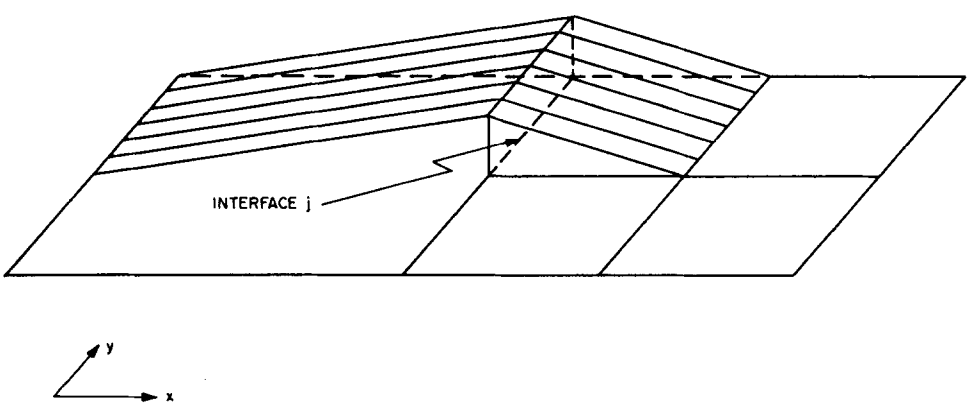

FIG. 5. The x-component of Θ_j; its y-component is identically zero.

and

$$q_i = \int_\Omega q\Psi_i, \qquad w_{jk} = \int_\Omega w\Theta_j \cdot \Theta_k, \tag{4.9c}$$

where i runs through all blocks and where j and k run through all interfaces. The a_i is the area of block i.

The discrete approximation of the unknowns is:

$$\phi^d(x, y) = \sum_i \phi_i c_i \Psi_i(x, y)/(a_i c(x, y)), \qquad c > 0 \tag{4.10a}$$

$$= \sum_i \phi_i \Psi_i(x, y), \qquad c = 0 \tag{4.10b}$$

and

$$\mathbf{u}^d(x, y) = \sum_j u_j \Theta_j(x, y), \text{ with } u_j = 0 \text{ if interface } j \text{ lies on } \partial\Omega \text{ and}$$
$$u_j = u_k \text{ if interfaces } j \text{ and } k \text{ are parts of the same green side } g. \tag{4.11}$$

The summation in (4.10) is over all blocks and in (4.11) over all interfaces.

In order to discretise (4.1), we substitute (4.10)–(4.11), multiply by Ψ_i, and integrate over Ω. This results in

$$c_i\phi_i + \sum_j (\pm) h_j u_j = q_i, \tag{4.12}$$

where i runs through all blocks, and where the summation over j is over all interfaces neighbouring to block i. The h_j is the mesh size of interface j and $(\pm)$ represents $+$ for positive neighbours and $-$ for negative ones. Equations (4.12) are equivalent to

$$c\phi^d + \nabla \mathbf{u}^d = \tilde{q} \tag{4.13}$$

with

$$\tilde{q}(x, y) = \sum_i q_i \Psi_i(x, y)/a_i, \tag{4.14}$$

since all terms in (4.13) are blockwise constant, as follows from the construction of ϕ^d and from the constraints in (4.11).

Let X^d and H^d be the finite dimensional subspaces of X and H of functions of the form (4.10)–(4.11). $X^d_{\tilde{q}}$ and $H^d_{\tilde{q}}$ are the subspaces of X^d and H^d constrained by (4.12). Then $X^d_{\tilde{q}}$ and $H^d_{\tilde{q}}$ are subspaces of $X_{\tilde{q}}$ and $H_{\tilde{q}}$.

To define discrete approximations to (4.1)–(4.3) we minimise f in $X^d_{\tilde{q}}$ $(c > 0)$ or

$H_{\tilde{q}}^d$ ($c=0$). Introducing the vectors $\Phi=\{\phi_i\}$, $U=\{u_i\}$, $Q=\{q_i\}$, the diagonal matrix $C=\text{diag}\{c_i\}$, and the sparse matrices $W=\{w_{jk}\}$ and D with elements

$$\begin{aligned} d_{ij} &= +h_j, && \text{if interface } j \text{ is a positive neighbour of block } i,\\ &= -h_j, && \text{if interface } j \text{ is a negative neighbour of block } i,\\ &= 0, && \text{otherwise,} \end{aligned} \tag{4.15}$$

we may write the constraint (4.12) as

$$C\Phi + DU = Q \tag{4.16}$$

and the functional $f(\phi^d, \mathbf{u}^d)$ becomes

$$F(\Phi, U) = (C\Phi \cdot \Phi) + (WU \cdot U). \tag{4.17}$$

Denoting

$$\begin{aligned} \mathbf{H} &= \{U \mid U \text{ satisfies the constraints in (4.11)}\}\\ \mathbf{X} &= \{\Phi\} * \mathbf{H} \end{aligned} \tag{4.18}$$

and the affine subspaces of $\mathbf{H}$ and $\mathbf{X}$ defined by (4.16) as $\mathbf{H}_Q$ and $\mathbf{X}_Q$, the minimisation is formulated as the minimisation of F in (4.17) within $\mathbf{H}_Q$ ($c=0$) or $\mathbf{X}_Q$ ($c>0$). The unique minimum is attained at the solution of the system consisting of the constraints in (4.11), of (4.16), and of

$$-\sum_i d_{ij}\phi_i + \sum_k w_{jk}u_k = 0 \tag{4.19a}$$

for each non-green interface j and

$$\sum_{j\in g}\left\{-\sum_i d_{ij}\phi_i + \sum_k w_{jk}u_k\right\} = 0 \tag{4.19b}$$

for each green side g. Equations (4.19) are to be considered as the discretisation of (4.2). The sparsity pattern follows from (4.15).

Let h be the maximal mesh size in the grid and let superscript h denote discretisation on the regular grid with mesh size h in the whole region Ω. From standard finite element error estimates (for $c=0$, see [13]), we have $O(h)$-approximation in the L_2 and H-norms for the solution $(\phi^h, \mathbf{u}^h)$ to the solution $(\tilde{\phi}, \tilde{\mathbf{u}})$ of the continuous problem with right-hand side $\tilde{q}$. Because of the minimisation principle and the inclusions $X_{\tilde{q}}^h \subset X_{\tilde{q}}^d \subset X_{\tilde{q}}$ and $H_{\tilde{q}}^h \subset H_{\tilde{q}}^d \subset H_{\tilde{q}}$, the approximations $(\phi^d, \mathbf{u}^d)$ ($c>0$) and $\mathbf{u}^d$ ($c=0$) are as good at least. For $c=0$, $O(h)$-approximation for ϕ^d holds as well, because the Brezzi condition between L_2^d and H^d holds uniformly for all regular and locally refined grids (according to extension of the proofs in [13]). As the additional error caused by the shift from q to $\tilde{q}$ can be estimated to be $O(h)$ too, consistency of $O(h)$ holds for the discretisation method.

5. Multi-Grid Solution

In the previous section the computation of the discrete fields (Φ, U) has been formulated as the minimisation of F in (4.17) within $\mathbf{X}_Q$. Here we describe an iterative solution process which

1. transforms a guess $(\Phi^{(0)}, U^{(0)}) \in \mathbf{X}$ into an admissible initial value

$$(\Phi^{(1)}, U^{(1)}) \in \mathbf{X}_Q.$$

2. constructs updates

$$(\Phi^{(n+1)}, U^{(n+1)}) = (\Phi^{(n)}, U^{(n)}) + (\Delta\Phi^{(n)}, \Delta U^{(n)}) \in \mathbf{X}_Q$$

by reducing F in affine subspaces

$$(\Phi^{(n)}, U^{(n)}) + \mathbf{X}_0^{(n)} \subset \mathbf{X}_Q.$$

In order for this iterative process to converge it is sufficient that the $\mathbf{X}_0^{(n)}$ sweep cyclically through a finite number of relaxation subspaces $\mathbf{X}_{0,m}$ with the property $\oplus_m \mathbf{X}_{0,m} = \mathbf{X}_0$, as follows from the theorem in Appendix A, which is essentially the Ostrowski–Reich theorem as formulated by Varga [14].

As already begun in the notations above, throughout the first part of this section we will consider the compressible case $c > 0$ in detail. The non-trivial modifications required for $c = 0$ are postponed till Subsection 5.2. In Subsection 5.1 we define the relaxation subspaces $\mathbf{X}_{0,m}$ which constitute the multi-grid method proper. In Subsection 5.3 convergence rates for $c = 0$ are presented. To accelerate convergence special relaxation subspaces of $\mathbf{H}_0$ are added to the set of $\mathbf{H}_{0,m}$ (the analoges of $\mathbf{X}_{0,m}$ for $c = 0$).

5.1. *Multi-Grid for Compressible Fluids* $(c > 0)$

5.1.1. *Multi-Grid Definitions*

In order to specify a multi-grid choice for the relaxation subspaces $\mathbf{X}_{0,m}$, we will first introduce multi-grid operators and matrices.

For each coarse grid l, $1 \leqslant l < k$, the prolongation is a mapping from functions on the coarse grid to functions on the discretisation grid. Since the discretisation grid is obtained from the coarsest one by basic refinement only, it is sufficient to define prolongation for this basic refinement. The functions Ψ and Θ, introduced in Section 4, each induce a prolongation:

Prolongation P_ψ for vectors associated with blocks,

$$\phi_{i'} = \phi_i, \tag{5.1}$$

where i is the number of the coarse block and $i' = if(i), ..., if(i) + 3$ runs through its sons, see also Fig. 4.

Prolongation P_u for vectors associated with interfaces: For interface-born interfaces,

$$u_{j'} = u_j, \tag{5.2}$$

where j is the number of the coarse interface and $j' = jf(j), jf(j)+1$ runs through its sons. For a block-born interface the value is the average of the two neighbouring parallel interfaces of the same level.

For fine blocks and interfaces in grid l the prolongations are the identity mapping. The prolongations can be represented by matrices P_ψ^{kl} and P_u^{kl}, where the superscripts k and l are suppressed when there is no risk of confusion.

The restriction is a mapping from the discretisation grid to the coarse grid l. It is (an approximation of) the generalised inverse of the prolongation:

Restriction R_ψ for vectors associated with blocks:

$$\phi_i = \tfrac{1}{4} \sum_{i'} \phi_{i'}. \tag{5.3}$$

Restriction R_u for vectors associated with interfaces:

$$u_j = \tfrac{1}{2} \sum_{j'} u_{j'}. \tag{5.4}$$

We again use matrix notation R_ψ^{lk} and R_u^{lk}, where the superscripts are suppressed whenever possible. The products $R_\psi P_\psi$ and $R_u P_u$ equal the identity mapping and $R_\psi = \frac{1}{4} P_\psi^*$.

Consistent with (4.10), which makes $c\phi$ blockwise constant, we introduce for the potential Φ a pologation P_ϕ and a restriction R_ϕ, which depend on c. For each grid l we have a diagonal matrix C^l with elements given by (4.9). Then

$$P_\phi^{kl} = (C^k)^{-1}(R_\psi^{lk})^* C^l, \qquad R_\phi^{kl} = (C^l)^{-1}(P_\psi^{kl})^* C^k. \tag{5.5}$$

The matrix C^k determines the matrix C^l by

$$c_i = 16 \left(\sum_{i'} 1/c_{i'} \right)^{-1}, \tag{5.6}$$

as follows from (4.9) (summation is over the four sons of block i). The matrix C^l can be written as

$$C^l = P_\psi^* C^k P_\phi. \tag{5.7}$$

For coarse grid l we form the matrices

$$D^l = P_\psi^* D^k P_u, \tag{5.8}$$

$$W^l = P_u^* W^k P_u, \tag{5.9}$$

and the vector

$$Q^l = P_\psi^* Q^k. \tag{5.10}$$

The functional F of (4.17) is indexed for its level

$$F^l(\Phi^l, U^l) = (C^l \Phi^l \cdot \Phi^l) + (W^l U^l \cdot U^l) \tag{5.11}$$

and also the spaces $\mathbf{X}^l$ and $\mathbf{X}^l_{Q^l}$. Note that the quantities in (5.7)–(5.10) are equal to the quantities obtained by direct application of the discretisation of Section 4 to the grid l. This is a very favourable feature for a program with dynamic adaptation of the grid, see Section 6.

For later use we mention the equalities

$$C^l R_\phi = P_\psi^* C^k, \qquad D^l R_u = P_\psi^* D^k \tag{5.12}$$

$$C^k P_\phi = \tfrac{1}{4} P_\psi C^l, \qquad D^k P_u = \tfrac{1}{4} P_\psi D^l. \tag{5.13}$$

5.1.2. *Complete Coarse Grid Correction* ($l < k$)

Let (Φ^k, U^k) be in $\mathbf{X}^k_{Q^k}$. Then $(\Phi^l = R_\phi \Phi^k,\ U^l = R_u U^k)$ is in $\mathbf{X}^l_{Q^l}$, as follows from (5.12). To improve on (Φ^k, U^k) as a solution of the system (4.11), (4.16), (4.19), we minimise F^k within

$$\{(\Phi^k, U^k) + (P_\phi \Delta\Phi^l, P_u \Delta U^l) \mid (\Delta\Phi^l, \Delta U^l) \in \mathbf{x}_0^l\}, \tag{5.14}$$

which indeed defines an affine subspace of $\mathbf{X}^k_{Q^k}$, as follows from (5.13). The minimisation of F^k is equivalent to the minimisation of

$$F^l(\tilde{\Phi}^l, \tilde{U}^l) - 2(\Delta T^l, \tilde{U}^l), \tag{5.15}$$

where

$$\Delta T^l = P_u^*(D^k{}^* \Phi^k - W^k U^k) - (D^l{}^* \Phi^l - W^l U^l) \tag{5.16}$$

with $(\tilde{\Phi}^l, \tilde{U}^l)$ in the affine subspace $\mathbf{X}^l_{Q^l}$. The solution satisfies system (4.11), (4.16), and (4.19) for level l with a right-hand member ΔT^l in (4.19).

The update $(\tilde{\Phi}^k, \tilde{U}^k)$ of (Φ^k, U^k), and its restriction on the coarse grids is given by

$$\tilde{\Phi}^m = \Phi^m + P_\phi(\tilde{\Phi}^l - \Phi^l), \qquad \tilde{U}^m = U^m + P_u(\tilde{U}^l - U^l) \tag{5.17a}$$

for $l \leqslant m < k$, by $(\tilde{\Phi}^l, \tilde{U}^l)$ on grid l, and by

$$\tilde{\Phi}^m = R_\phi \tilde{\Phi}^l, \qquad \tilde{U}^m = R_u \tilde{U}^l \tag{5.17b}$$

for $1 \leqslant m < l$.

5.1.3. *Relaxation on level l ($l \leqslant k$)*

It is obvious that with the procedures of Section 5.1.2 we will also obtain improvements of (Φ^k, U^k) and representations of the improved solution on the coarser levels according to (5.17), if we restrict the space for $(\tilde{\Phi}^l, \tilde{U}^l)$ to a proper affine subspace of $\mathbf{X}^l_{Q^l}$ (also for $l=k$). Let Ω^l_{loc} be a connected subdomain of Ω consisting of blocks in grid l. Let $\mathbf{X}^l_{0,\Omega^l_{\text{loc}}}$ be the subspace of $\mathbf{X}^l_0$ defined as

$$\{(\Phi^l, \Delta U^l) \in \mathbf{X}^l_0 \mid (\Delta\Phi^l, \Delta U^l) = 0 \text{ in } \Omega/\Omega^l_{\text{loc}}\}. \tag{5.18}$$

The determination of the update $(\tilde{\Phi}^l, \tilde{U}^l)$ is now reduced to relaxation in Ω^l_{loc}. This means to solving the equations of the level l discretisation (4.11), (4.16), and (4.19) (with right-hand member ΔT^l) which apply to the interior of Ω^l_{loc}, while keeping (Φ^l, U^l) fixed in $\Omega/\Omega^l_{\text{loc}}$ (in particular, U^l on the boundary $\partial\Omega^l_{\text{loc}}$ is fixed). It should be noticed that it is not necessary to recompute ΔT^l after each relaxation on level l, because the update to ΔT^l in (5.16)

$$P^*_u(D^{k*} P_\phi \Delta\Phi^l - W^k P_u \Delta U^l) - (D^{l*} \Delta\Phi^l - W^l \Delta U^l) = 0, \tag{5.19}$$

according to the definitions (5.8)–(5.9) of D^l and W^l. (Actually, none of the ΔT^m with $l \leqslant m \leqslant k$ changes.) Therefore, in a sequence of level l relaxations we may accumulate the corrections in $(\tilde{\Phi}^l, \tilde{U}^l)$ and have only to perform the update (5.17) when leaving level l.

5.1.4. *Relaxation Subspaces*

As relaxation subspaces $\mathbf{X}^k_{0,m}$ for the iterative process we choose the spaces $P^{kl}(\mathbf{X}^l_{0,\Omega^l_{\text{loc}}})$ which are related to the Ω^l_{loc} consisting of four blocks of level l with one common regular node (denoted as Ω^l_{node}), for $l = 2(1)k$, and to $\Omega^1_{\text{loc}} = \Omega$ for $l = 1$. We cyclically sweep through them in the order $l = 1(1)k$, while the order at level l is determined by the adaptive data structure of the grid and is rather arbitrary. This is called point relaxation. The sum of these relaxation spaces is $\mathbf{X}^k_0$, because the Φ^k can be eliminated by (4.16) so that the minimisation becomes minimisation over U^k only, and because each green side and non-green interface appears as an internal interface of one of the Ω^l_{node} at least. In 3D the Ω^l_{loc} should be either the union of four blocks having a common edge, or the union of eight blocks having a common node; both sets span $X^k_{0,m}$.

5.1.5. *Initialisation*

A guess $(\Phi^{(0)}, U^{(0)}) \in \mathbf{X}^k$ (so satisfying the constraints in (4.11); usually we take the fields at the previous time level adapted to the present grid) must be transformed into a value $(\Phi^{(1)}, U^{(1)}) \in \mathbf{X}^k_{Q^k}$ (so satisfying all constraints). For $c > 0$ this first update does not differ from the relaxation updates described in 5.1.3 and 5.1.4: after one sweep over $l = 1(1)k$, the solution and its restrictions on coarse grids given by (5.17b) satisfy

$$C^l\Phi^l + D^l U^l = Q^l, \qquad 1 \leqslant l \leqslant k \tag{5.20}$$

as follows from (5.12). The value of the functional F may increase at the transition from $(\Phi^{(0)}, U^{(0)})$ to $(\Phi^{(1)}, U^{(1)})$, but is reduced in the subsequent iterative process.

5.2. *Multi-Grid for Incompressible Fluids* ($c=0$)

The multi-grid relaxation for incompressible fluids is obtained from the relaxation in subsection 5.1 by taking the limit c constant $\to 0$. Then $P_\phi = P_\psi$ and $R_\phi = R_\psi$. Existence of a solution of the set of equations in Ω^l_{loc} (see 5.1.3) is a consequence of $U^l \in \mathbf{H}^l_{Q^l}$. For uniqueness of the $\tilde{\Phi}^l$ we prescribe

$$\sum_i a_i(\tilde{\phi}_i - \phi_i) = 0 \tag{5.21}$$

as follows from taking c constant $\to 0$. In (5.21) the summation is over the blocks in Ω^l_{loc}.

To ensure convergence of the process we will show in 5.2.1 that $\bigoplus_m \mathbf{H}_{0,m} = \mathbf{H}_0$ for the subspace associated with the series of Ω^l_{node} formed in 5.1.4. (Note that for $c=0$ we cannot eliminate Φ^k by (4.16), and (4.16) is indeed a constraint on U^k.) In 5.2.2 we give necessary modifications for the initialisation.

5.2.1. *Completeness of the Relaxation Subspaces*

For $c=0$, the regions Ω^l_{node} as introduced in 5.1.4 are associated with the subspaces

$$\{P_u \Delta U^l \mid \Delta U^l \in \mathbf{H}^l_0,\ \Delta U^l = 0 \text{ in } \Omega/\Omega^l_{\text{node}}\} \tag{5.22}$$

of $\mathbf{H}^k_0$, and we have to show that all spaces of the form (5.22) span $\mathbf{H}^k_0$.

For each $U \in \mathbf{H}^k_0$ there is a stream-function $\psi^d(x, y)$, zero on $\partial\Omega$, so that

$$\mathbf{u}^d(x, y) = \nabla \times \psi^d(x, y)\mathbf{z}, \tag{5.23}$$

where $\mathbf{u}^d$ is the approximation (4.11) and $\mathbf{z}$ is the unit vector in the z-direction. The ψ^d follow from the $\mathbf{u}^d$ by integration from $\partial\Omega$ into Ω of the component of $\mathbf{u}^d$ normal to the path of integration. The ψ^d is a first-order polynomial in x or y on each (non-green) interface and on each green side, and is bilinear in each block. This implies a bijection between the solenoidal fields $\mathbf{u}^d$ of the form (4.11) and these bilinear stream functions ψ^d vanishing on $\partial\Omega$.

Each ψ^d can be written as

$$\psi^d(x, y) = \sum_i \psi_i e_i(x, y), \tag{5.24}$$

where the sum is over all interior nodes and where the bilinear approximation functions e_i are one in node i and zero in the others. The flow field $\nabla \times e_i\mathbf{z}$ is called an eddy current. If the support of an eddy current and associated stream function equals the closure of one of the Ω^l_{node}, the eddy current and stream function are

called elementary. All other eddy currents and their stream functions are called green and their support is denoted by $\Omega^l_{\text{green node}}$, where l is the maximum of the levels neighbouring to node i, see Fig. 6.

The elements of the spaces (5.22) represent elementary eddy currents. It follows from (5.24) that they span $\mathbf{H}^k_0$, if we show that each green e_i is the sum of elementary stream functions.

Let the support of a green e_i be the union of blocks of levels k and $k-1$. Then this support is contained in the closure of some $\Omega^{k-1}_{\text{node}}$. Let $\tilde{e}_i$ be the elementary stream function on $\Omega^{k-1}_{\text{node}}$. Then

$$e_i = \tilde{e}_i - \sum_{i'} \tilde{e}_i(x_{i'}, y_{i'})\, e_{i'}, \tag{5.25}$$

where the summation is over all nodes in $\Omega^{k-1}_{\text{node}}$. Equation (5.25) shows that e_i is the sum of elementary stream functions. By induction the proof follows for green e_i with a support involving more levels.

5.2.2. *Initialisation*

For $c=0$, the initialisation of Section 5.1.5 would be wrong, since the set of equations encountered on an Ω^l_{node} would generally not admit a solution. Therefore we present a modified initialisation.

Instead of sweeping through all Ω^l_{node}, we restrict the initialisation sweep to Ω^1 and the Ω^l_{node} composed of blocks with a common father block. For $l=1$ the constraint equations have a solution, since the sum of the entries of Q^1 equals that of Q^k, which is zero. If on level l, U^l satisfies the constaints (5.20), then on the next level we have

$$(P^{l+1,l}_{\psi})^*(Q^{l+1} - D^{l+1}U^{l+1}) = 0 \tag{5.26}$$

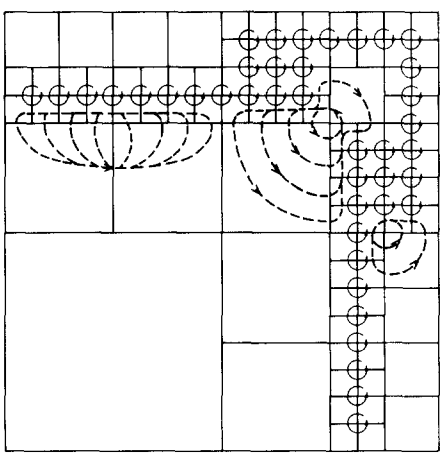

FIG. 6. Solid lines: elementary eddy current in Ω^4_{node}; dashed lines: green eddy current in $\Omega^4_{\text{green node}}$.

as follows from (5.17a) and (5.12). Equation (5.26) implies that the set of equations for each $\Omega_{\text{node}}^{l+1}$ proposed has a solution. By induction it follows that one sweep results in a $U^{(1)}$ satisfying the constraints.

5.3. *Convergence Rates*

In this section a summary of numerical results for the convergence rate in a number of examples with $c = 0$ is given.

To measure the convergence rate we use the spectral radius of the error amplification matrix associated with one cycle of the multi-grid process. To compute the spectral radius numerically we put Q equal to zero, take an initial guess $U^{(1)} \neq 0$, and compute iterative approximations $U^{(n)}$ to the solution $U = 0$. The spectral radius is recovered from

$$\rho = \lim_{n \to \infty} \rho_n; \qquad \rho_n = \|U^{n+1}\|/\|U^n\|, \tag{5.27}$$

where $\|U\| = (WU \cdot U)^{1/2}$. All tables give values of $\tilde{\rho} = -^{10} \log \rho$, which represents the number of significant decimal digits of the solution computed per multi-grid sweep.

In most cases ρ_n had become steady at $n = 50$; then we display $\rho = \rho_{50}$. In some cases ρ_n was fluctuating with a typical period near 2 or 3; then we display $\rho = \max_{40 \leqslant n \leqslant 50} \rho_n$.

The computational domain Ω for each example is $\{0 < x < 8 \wedge 0 < y < 8\}$ and the permeability $k = w^{-1}$ is a diagonal tensor with diagonal elements k_x and k_y given as functions of x and y. We consider four examples, each containing a parameter λ.

(a) Two subregions, $\Omega_1 = \{x < 2 \vee y > 6\}$, $\Omega_2 = \Omega/\Omega_1$ in $\Omega_1 : k_x = k_y = 1$; in $\Omega_2 : k_x = k_y = \lambda$.

TABLE I

Convergence Rate of Point Relaxation for Constant Isotropic permeability on Regular Grids

n_s	$k=2$	$k=3$	$k=4$	$k=5$
1	0.65	0.73	0.75	0.73
2	2.0	1.3	1.2	1.1
3	2.6	1.6	1.4	1.2
4	3.0	1.8	1.6	1.5
5	3.4	1.9	1.7	1.6

Note. n_s: number of sweeps through each grid; k: level of discretisation grid.

TABLE II

Convergence Rate of Point Relaxation on Regular Grid

λ	a	b	c
10^{-5}	0.49	0.27	0.13
10^{-3}	0.49	0.27	0.13
10^{-1}	0.63	0.50	0.25
1	0.75	0.75	0.75
10	0.75	0.44	0.44
10^{3}	0.74	0.27	0.23
10^{5}	0.63	0.26	0.23

Note. $n_s = 1$, $k = 4$.

(b) Three subregions, $\Omega_1 = \{x > 3 \wedge y < 5\}$, $\Omega_2 = \{(x < 2 \wedge y > 1) \vee (x < 7 \wedge y > 6)\}$, $\Omega_3 = \Omega/(\Omega_1 \cup \Omega_2)$; in Ω_1 and Ω_2: $k_x = k_y = 1$; in Ω_3: $k_x = k_y = \lambda$. (Note that for $\lambda > 1$, Ω_3 represents a "channel" for fluid flow, and for $\lambda < 1$ it represents a "barrier.")

(c) Three subregions, $\Omega_1 = \{x > 6.4 + 1.6 * (y/8)^2\}$, $\Omega_2 = \{y > 6.4 + 1.6 * (x/8)^2\}$, $\Omega_3 = \Omega/(\Omega_1 \cup \Omega_2)$; in Ω_1 and Ω_2: $k_x = k_y = 1$; in Ω_3: $k_x = k_y = \lambda$. (Note that Ω_3 is a domain as occupied by a water tongue at breakthrough in certain five-spot displacements.)

(d) $k_x = 1$, $k_y = \lambda$.

The rate of convergence for constant isotropic permeability ($\lambda = 1$ in all the examples) can be read from Table I, if the coarsest grid is $2 * 2$ and the discretisation grid is regular of level k. The number of sweeps n_s on each grid l varies from 1 to 5 and k varies from 2 to 5.

TABLE III

Convergence Rate for Anisotropic Permeability (Ex. d) on Regular Grid

λ	Point relaxation	Line relaxation
10^{-5}	0.012	0.65
10^{-3}	0.014	0.78
10^{-1}	0.15	1.2
1	0.75	1.4
10	0.14	1.2
10^{3}	0.013	0.78
10^{5}	0.012	0.65

Note. $n_s = 1$, $k = 4$.

TABLE IV

Convergence Rate on a Locally Refined Grid

λ	a	b Point relaxation	c	c Green eddy currents
10^{-5}	0.41	0.43	0.000089	0.052
10^{-3}	0.41	0.43	0.0027	0.053
10^{-1}	0.48	0.45	0.090	0.14
1	0.45	0.45	0.45	0.71
0	0.28	0.42	0.10	0.59
10^{3}	0.13	0.36	0.0027	0.30
10^{5}	0.13	0.36	0.00013	0.25

Note. $n_s = 1$, $[k = 6]$.

Rapid convergence on regular grids is obtained also when strong contrasts are present in the permeability, as in examples a, b, and c, see Table II. Here we have set $n_s = 1$ and $k = 4$.

Strong anisotropy requires line-relaxation. We form large relaxation spaces on level l as in Section 5.1, by combining all Ω^l_{node} which have their node on some (x- or y-)grid line into an Ω^l_{line}. With line-relaxation rapid convergence is retained, as appears from Table III. Here also $n_s = 1$ and $k = 4$.

Lastly we consider locally refined grids. Table IV gives results for the grid of Fig. 2a, with $n_s = 1$ and permeability as in (a), (b), and (c). In the first three columns we show the results of point relaxation.

The grid in Fig. 2a is "adapted" to the permeability of example c. It is a grid which appears in the adaptive grid solution of a five-spot displacement which features a water tongue, see Section 6. Note that large permeability gradients coincide with large "mesh-width gradients." From the third column of Table IV we see that the convergence rate of point-relaxation deteriorates for extreme λ.

To restore the convergence rate we add the relaxation spaces associated with $\Omega^l_{\text{green node}}$, introduced in 5.2.1. For arbitrary grids these regions may become quite complex as may be seen in Fig. 6, where only three eddy currents of such type are shown. When the grids are constrained by the condition that the levels of neighbouring blocks differ by not more than one, this complexity is very much moderated. With the above extension of the set of relaxation spaces we obtain the acceptable convergence rates of the last column in Table IV.

6. Dynamic Grid Refinement

Good representation of the solution requires updating the grid at certain moments during the simulation process. The updates are executed at fixed

simulation time. In this section we describe decision strategies for (un)refinement. The method is assessed on the basis of experimental results.

The split of the equations of reservoir simulation into a hyperbolic part (Eqs. (2.5a), (2.5b)) and an elliptic part (Eqs. (2.5c), (2.5d)) entails two different grid updates: for better approximation of quantities associated with the hyperbolic part (s_n, $\mathbf{u}^n$) and with the elliptic part (p, $\mathbf{u}$), respectively.

Hyperbolic Grid-Update

The hyperbolic grid-update aims at better approximation of the s_n. It refines and unrefines the grid according to local indicators, while the overall accuracy is determined by the total number of fine blocks, which is the given input.

At the computation of the indicators we use the explicit discretisation of (2.5a), (2.5b):

$$\frac{a_i}{\Delta t}(s_{n,i}^{\text{new}} - s_{n,i}) = q_{n,i} - \sum_j (\pm)\, h_j u_{n,j} \tag{6.1}$$

$$u_{n,j} = (k_n k^{-1})_j u_j. \tag{6.2}$$

Here i runs through all fine blocks; $s_{n,i}$ is the average and $q_{n,i}$ the integral over block i and a_i is the area. The $u_{n,j}$ is the average normal value of $\mathbf{u}^n$ on interface j. The summation in (6.1) is over all fine interfaces neighbouring to block i, see also (4.12). The evaluation of the coefficient preceding u_j in (6.2) is one-point upstream, u_j is known from the elliptic part. The time step Δt is the residence-time of fluid in a block, minimised over all blocks, and multiplied by a Courant–Friedrichs number.

The indicator E_i, belonging to block i, is computed as

$$E_i = \max_n [\max_j l(s_{n,j}, s_{n,i}) - \min_j l(s_{n,j}, s_{n,i})]. \tag{6.3}$$

Here j runs through the value i and the numbers of all upstream neighbouring blocks of block i, as determined by the sign of expression (6.2). The value of $l(s_{n,j}, s_{n,i})$ is $s_{n,i}$ for $j = i$ and is otherwise computed as follows:

— associate the value of $s_{n,j}$ (resp. $s_{n,i}$) with the centre of block j (resp. i);

— interpolate linearly between these points;

— evaluate at the edge of block i.

For blocks with injection wells, the extremes within the square brackets in (6.3) also run over the injection "saturations."

Next, we compute tolerances tol_s^+ and tol_s^-. Blocks with $E_i > \text{tol}_s^+$ and of level less than a given ml_h will be refined and blocks with $E_i < \text{tol}_s^-$ will be unrefined. The values of tol_s^+ and tol_s^- follow from:

— A given value of $\text{tol}_s^+/\text{tol}_s^-$ ($=2$, typically);

— A lower bound tol_s^{2-} for tol_s^-;

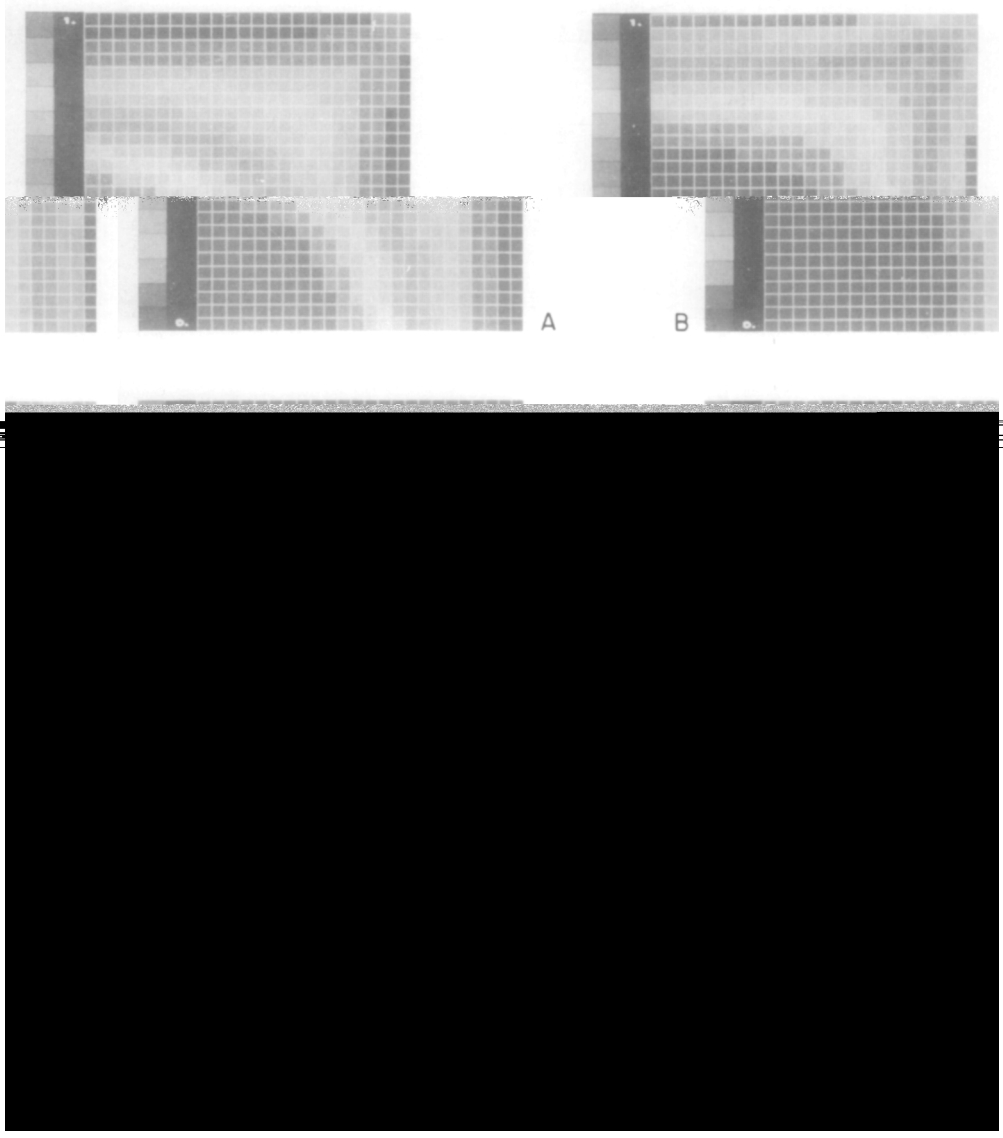

FIG. 8. Same as Fig. 7, but with a regular grid with $576 = 24^2$ blocks.

For $M = 2$ (Figs. 7a, b and 8a, b), the saturations in the exact solution are continuous functions of the space coordinates.

For $M = 1$ (Figs. 7c, d and 8c, d), the saturations in the exact solution adopt the values 0 and 1 only. The front is a contact discontinuity, the width of which increases owing to numerical diffusion. For model equations, discretised (with small

Courant–Friedrichs number) on regular grids, the width of the front is known to be proportional to $h^{1/2}$. The smallest mesh width in Fig. 7 is 0.375 times the meshwidth in Fig. 8, predicting a reduction of the front-with by 0.6. Comparison of Figs. 7c and 8c shows that the results reach this reduction. Observe that the water tongue near the production well is much better resolved in Fig. 7d than in Fig. 8d.

For $M = 0.5$ (Figs. 7e, f and 8e, f), the exact solution again displays 0 and 1 only. Now the front is a stable shock. Numerical diffusion is cancelled by the stability of the shock, resulting in an effective front-width not more than two gridblocks, as can be seen from Figs. 7e, f and 8e, f. Hence the front-width is proportional to h. Observe that the tongue is again seen better in Fig. 7f than in Fig. 8f.

APPENDIX A: A THEOREM FOR SECTION 5

THEOREM. *Let H be a finite dimensional Hilbert space (norm: $\|\cdot\|$, inner product $(\cdot,\cdot)$ and let $V_0, \ldots, V_{M-1}$ be subspaces, the sum of which equals H. Let $x_0 \in H$ be arbitrary and*

$$x_{n+1} = x_n - v_n, \tag{A.1}$$

where v_n is such that

$$v_n \in V_m, \qquad n = m \bmod(M) \tag{A.2}$$

$$\|x_{n+1}\| < \|x_n + v\|, \qquad v \in V_m. \tag{A.3}$$

Then we have

$$\lim_{n \to \infty} x_n = 0. \tag{A.4}$$

Proof. The $\|x_n\|$ form a decreasing bounded series and, hence, form a convergent series. The definition of v_n implies

$$(x_n, v) = (v_n, v), \qquad v \in V_m. \tag{A.5}$$

It follows that

$$(v_n, v_n) = 2(x_n, v_n) - (v_n, v_n) = \|x_n\|^2 - \|x_{n+1}\|^2 \to 0 \qquad \text{for} \quad n \to \infty. \tag{A.6}$$

For fixed $v \in V_m$ we have from (A.5)

$$|(x_n, v)| = |(v_n, v)| \leqslant \|v_n\| \cdot \|v\|, \qquad n = m \bmod(M) \tag{A.7}$$

and from (A.6)

$$(x_n, v) \to 0, \qquad n \to \infty, n = m \bmod(M). \tag{A.8}$$

(A.1) and (A.6) imply

$$\|x_n - x_{n'}\| \to 0, \qquad n, n' \to \infty; |n - n'| < M, \tag{A.9}$$

with which (A.8) gives

$$(x_n, v) \to 0, \qquad n \to \infty. \tag{A.10}$$

Since each vector in H can be written as a finite sum of vectors in the V_m, (A.10) holds for all vectors $v \in H$, and this implies (A.4), since H is finite dimensional.

REFERENCES

1. R. E. ALCOOUFFE, A. BRANDT, J. E. DENDY, JR., AND J. W. PAINTER, *SIAM J. Sci. Statist. Comput.* **2**, 430 (1981).
2. R. E. BANK, "The efficient implementation of local mesh refinement algorithms," in *Adaptive Computational Methods for Partial Differential Equations*, edited by I. Babuska *et al.* (SIAM, Philadelphia, 1983).
3. R. E. BANK AND T. F. CHAN, *SIAM J. Sci. Statist. Comput.* **7**, 540 (1986).
4. A. BEHIE AND P. A. FORSYTH, *SPE J.* **23**, 623 (1985).
5. A. BRANDT, *Math. Comput.* **31**, 333 (1977).
6. A. BRANDT, J. E. DENDY, JR., AND H. RUPPEL, *J. Comput. Phys.* **34**, 348 (1980).
7. G. CHAVENT, J. JAFFRÉ, G. COHEN, AND I. DIESTE, "Simulation of Two-Dimensional Waterflooding Using Mixed Finite Elements," in *Sixth SPE Symposium on Reservoir Simulation, New Orleans, LA, Jan. 31, 1982.*
8. G. CHAVENT, G. COHEN, J. JAFFRÉ, M. DUPUY, AND I. RIBERA, *SPE J.* **24**, 382 (1984).
9. B. L. DARLOW, R. E. EWING, AND M. F. WHEELER, "Mixed Finite Element Methods for Miscible Displacement Problems in Porous Media", in *Sixth SPE Symposium on Reservoir Simulation, New Orleans, LA, Jan. 31, 1982.*
10. R. E. EWING AND R. F. HEINEMANN, "Incorporation of Mixed Finite Element Methods in Compositional Simulation for Reduction of Numerical Dispersion", in *Seventh SPE Symposium on Reservoir Simulation, San Francisco, CA, Nov. 16, 1983.*
11. R. E. HEINEMANN, G. GERKEN, AND G. VON HANTELMAN, "Using Local Grid Refinement in a Multiple-Application Reservoir Simulator", in *Seventh SPE Symposium on Reservoir Simulation, San Francisco, CA, Nov. 16, 1983.*
12. R. KETTLER, Lecture Notes in Math. Vol. 960, p. 502, Springer-Verlag, New York/Berlin, 1982.
13. P. A. RAVIART AND J. M. THOMAS, Lecture notes in Math. Vol. 606, p. 292, Springer-Verlag, New York/Berlin, 1977.
14. R. S. VARGA, *Matrix Iterative Analysis* (Prentice-Hall, Englewood Cliffs, NJ, 1962), p. 78.