Electronic Transactions on Numerical Analysis.
Volume 10, 2000, pp. 56-73.
Copyright © 2000, Kent State University.
ISSN 1068-9613.

MULTILEVEL PROJECTION METHODS FOR NONLINEAR LEAST-SQUARES FINITE ELEMENT COMPUTATIONS[*]

JOHANNES KORSAWE AND GERHARD STARKE[†]

Abstract. The purpose of this paper is to describe and study several algorithms for implementing multilevel projection methods for nonlinear least-squares finite element computations. These algorithms are variants of the full approximation storage (FAS) scheme which is widely used in nonlinear multilevel computations. The methods are derived in the framework of the least-squares mixed formulation of nonlinear second-order elliptic problems. The nonlinear variational problems on each level are handled by smoothers of Gauss-Seidel type based on a space decomposition of the finite element spaces. Finally, the different algorithms are tested and compared for a nonlinear elliptic problem arising from an implicit time discretization of a variably saturated subsurface flow model.

Key words. nonlinear elliptic problems, least-squares finite element method, nonlinear multilevel methods, multilevel projection methods, FAS scheme.

AMS subject classifications. 65M55, 65M60.

1. Introduction.

In recent years, least-squares mixed finite element methods have received much attention and have been applied to a number of problems which can be stated as first-order systems (see e.g. the recent survey paper [3]). One of the advantages that contributed to the popularity of the least-squares approach is the fact that the resulting algebraic problems are positive definite and may be easier to attack by solution techniques like multilevel methods. Another advantage is the accessability of a "built-in" a posteriori error estimator that does not require additional computations and estimates nonlinear problems directly without linearization.

The focus of this paper will be on the iterative solution of the nonlinear algebraic least-squares problems arising from the first-order system least-squares approach. The methods will be presented in the framework of the least-squares mixed finite element formulation of nonlinear second-order elliptic problems. We study several algorithms for the implementation of nonlinear multilevel projection methods for least-squares mixed finite element discretizations. These algorithms are different variants of the full approximation storage (FAS) scheme which is widely used in nonlinear multilevel computations. The nonlinear variational problems on each level are handled by smoothers of Gauss-Seidel type based on a space decomposition of the finite element spaces. Since the local evaluation of the least-squares functional provides an easily accessible a posteriori error estimator, it is natural to use adaptive mesh refinement strategies in the least-squares finite element framework. We therefore also describe an adaptive version of the multilevel projection schemes utilizing the smoothing iteration only in those parts of the domain where it is needed.

Although the FAS scheme can be applied to the least-squares formulation of general nonlinear first-order systems, this work is motivated from its application to variably saturated subsurface flow. A widely used model in this context is based on two first-order differential equations which stand for mass conservation (scalar equation) and a generalization of Darcy's law (vector equation) for the two process variables, namely, the hydraulic potential (scalar unknown) and volumetric flux (vector unknown). This model is often used in combination with a parametrization proposed by Mualem [13] and Van Genuchten [16]. Mathematically, this is equivalent to a nonlinear parabolic initial-boundary value problem, which is possibly degenerate, for the scalar variable (hydraulic potential). Since one is also interested in accurate

[*] Received May 25, 1999. Accepted for publication February 1, 2000. Recommended by K. Jordan.

[†] Fachbereich Mathematik, Universität-GH Essen, 45117 Essen, Germany ({jkorsawe,starke}@ing-math.uni-essen.de). This work was sponsored by a grant of the Deutsche Forschungsgemeinschaft.

approximations to the vector variable (flux), it is appropriate to work directly with the first-order system as a starting point for the numerical method. After an implicit discretization of the time derivative, we obtain the first-order system formulation of a nonlinear elliptic boundary value problem. The solution of nonlinear variably saturated subsurface flow problems by least-squares finite element methods has been studied in [15] and [14]. The resulting systems of nonlinear algebraic equations are solved by a damped Gauss-Newton method in [15] and an inexact Gauss-Newton method with a multilevel V-cycle as inner iteration in [14]. In the context of variably saturated subsurface flow, the use of nonlinear multilevel methods is motivated by the fact that the nonlinearity of the problem is usually pronounced only in relatively small areas of the computational domain. This lets us expect that a nonlinear multilevel approach is more effective since it is more flexible in handling local nonlinear effects.

The following section presents the background on the least-squares mixed finite element formulation of nonlinear second-order elliptic problems. Three different FAS-type schemes for the implementation of nonlinear multilevel projection methods are derived in Section 3. Section 4 is concerned with the Gauss-Seidel-type smoothing iterations for the solution of the nonlinear variational problems on each level. Implementational issues are discussed in Section 5. In Section 6, the different algorithms are tested and compared for a nonlinear elliptic problem arising from an implicit time discretization of a variably saturated subsurface flow model. Finally, we draw some conclusions in Section 7.

2. Nonlinear Least-Squares Finite Element Formulation. We consider nonlinear second-order elliptic boundary value problems re-written as a first-order system

$$\mathcal{R}(p,u) := \begin{pmatrix} \operatorname{div} u + b(p) + f \\ u + a(p)\,\nabla p \end{pmatrix} = 0 \tag{2.1}$$

in $\Omega \subset \mathbb{R}^2$ subject to boundary conditions $p = p^D$ on $\Gamma_D \subset \partial\Omega$ and $n \cdot u = n \cdot u^N$ on $\Gamma_N \subset \partial\Omega$ ($\Gamma_D \cup \Gamma_N = \partial\Omega$). For example, modelling variably saturated subsurface flow leads, after an implicit discretization with respect to time, to (2.1) with

$$a(p) = K(\theta(p))\ ,\ b(p) = \frac{\theta(p)}{\tau}\ ,\ f = -\frac{\theta(p^{\text{old}})}{\tau}$$

(see, for example, [11, Sect. 3.3]). Here, p and p^{old} denote the hydraulic potential at the current and previous time-steps, respectively, $K(\theta(p))$ is the permeability, $\theta(p)$ the water content and τ the time-step. Existence and uniqueness of the underlying parabolic initial-boundary value problem is discussed in [1]. In particular, for the parametrization used in Section 6, the functions $K(\theta)$ and $\theta(p)$ are such that the assumptions in [1] are fulfilled.

We compute approximate solutions to (2.1) using the least-squares finite element method. To this end, let

$$\begin{aligned} Q_h \subset Q &:= \{q \in H^1(\Omega) : q = 0 \text{ on } \Gamma_D\}\ , \\ V_h \subset V &:= \{v \in H(\text{div},\Omega) : n \cdot v = 0 \text{ on } \Gamma_N\} \end{aligned}$$

be suitable finite-dimensional spaces and let $p^D \in H^1(\Omega)$ and $u^N \in H(\text{div},\Omega)$ be extensions of the boundary values on Γ_D and Γ_N, respectively. The least-squares finite element approximation is then defined as $(p_h, u_h) = (p^D + \hat{p}_h, u^N + \hat{u}_h)$ with $(\hat{p}_h, \hat{u}_h) \in Q_h \times V_h$ such that

$$\|\mathcal{R}(p^D + \hat{p}_h, u^N + \hat{u}_h)\|_{0,\Omega}^2 = \min_{\hat{q}_h \in Q_h, \hat{v}_h \in V_h} \|\mathcal{R}(p^D + \hat{q}_h, u^N + \hat{v}_h)\|_{0,\Omega}^2\ . \tag{2.2}$$

In the sequel, we will concentrate on standard piecewise linear continuous functions for Q_h and the lowest-order Raviart-Thomas spaces for V_h. The approximation properties of this approach are discussed in [15] for the above-mentioned variably saturated subsurface flow model and in [6] for the linear case (i.e., with $a(p)$ constant and $b(p) = cp$).

By $\mathcal{J}(p_h, u_h)$ we denote the Fréchet derivative of $\mathcal{R}(\cdot,\cdot)$ with respect to Q_h and V_h at (p_h, u_h). For the first-order operator $\mathcal{R}(\cdot,\cdot)$ of (2.1) this is given by

$$\mathcal{J}(p_h, u_h)\begin{pmatrix}\hat{q}_h\\ \hat{v}_h\end{pmatrix} = \begin{pmatrix}\text{div}\,\hat{v}_h + b'(p_h)\,\hat{q}_h\\ \hat{v}_h + a'(p_h)\,\nabla p_h\,\hat{q}_h + a(p_h)\,\nabla\hat{q}_h\end{pmatrix}\ . \tag{2.3}$$

The nonlinear least-squares problem (2.2) is equivalent to the variational problem

$$\left(\mathcal{R}(p^D + \hat{p}_h, u^N + \hat{u}_h), \mathcal{J}(p^D + \hat{p}_h, u^N + \hat{u}_h)\begin{pmatrix}\hat{q}_h\\ \hat{v}_h\end{pmatrix}\right)_{0,\Omega} = 0 \tag{2.4}$$

for all $(\hat{q}_h, \hat{v}_h) \in Q_h \times V_h$.

With respect to bases for the spaces Q_h and V_h, (2.2) becomes a nonlinear algebraic least-squares problem which may be solved using the Gauss-Newton method [15]. In [14], an inexact Gauss-Newton method for nonlinear least-squares computations is studied which uses a multilevel method for the linear least-squares problems arising in each step. An alternative approach for least-squares finite element computations is to use a nonlinear multilevel method which is based on successive minimization of the (nonlinear) least-squares functional with respect to low-dimensional subspaces on different levels.

3. FAS-type Nonlinear Multilevel Projection Schemes.

Multilevel methods are based on a sequence of triangulations $\mathcal{T}_0, \mathcal{T}_1, \ldots, \mathcal{T}_L$ which are assumed to be shape-regular but may be the result of an adaptive refinement process. Associated with these triangulations are sequences of subspaces $Q_0, Q_1, \ldots, Q_L$ and $V_0, V_1, \ldots, V_L$. Similarly to the linear case, nonlinear multilevel methods are also based on the idea of coarse space corrections (for treatises of nonlinear multilevel methods see e.g. [10, Ch. 9] or [4, Sect. V.5]). If (p_l, u_l) is an approximation on level l, our aim is to work with the least-squares problem of finding the correction $(\hat{p}_{l-1}, \hat{u}_{l-1}) \in Q_{l-1} \times V_{l-1}$ on level $l-1$ such that

$$\begin{aligned}&\|\mathcal{R}(p_l + \hat{p}_{l-1}, u_l + \hat{u}_{l-1})\|^2_{0,\Omega}\\ &\quad= \min_{\hat{q}_{l-1}\in Q_{l-1}, \hat{v}_{l-1}\in V_{l-1}} \|\mathcal{R}(p_l + \hat{q}_{l-1}, u_l + \hat{v}_{l-1})\|^2_{0,\Omega}\ .\end{aligned} \tag{3.1}$$

This is exactly the multilevel projection method, as described in [12], applied to our nonlinear least-squares formulation. As above, the minimizer $(\hat{p}_{l-1}, \hat{u}_{l-1}) \in Q_{l-1} \times V_{l-1}$ also satisfies the variational formulation

$$(\mathcal{R}(p_l + \hat{p}_{l-1}, u_l + \hat{u}_{l-1}), \mathcal{J}(p_l + \hat{p}_{l-1}, u_l + \hat{u}_{l-1})\begin{pmatrix}\hat{q}_{l-1}\\ \hat{v}_{l-1}\end{pmatrix})_{0,\Omega} = 0 \tag{3.2}$$

for all $(\hat{q}_{l-1}, \hat{v}_{l-1}) \in Q_{l-1} \times V_{l-1}$. In the linear case, we may simply use

$$\mathcal{R}(p_l + \hat{p}_{l-1}, u_l + \hat{u}_{l-1}) = \mathcal{R}(p_l, u_l) + \mathcal{R}(\hat{p}_{l-1}, \hat{u}_{l-1})$$

and the fact that $\mathcal{J}(\cdot,\cdot)$ does not depend at all on $(p_l + \hat{p}_{l-1}, u_l + \hat{u}_{l-1})$. The correction equation (3.2) then turns into

$$(\mathcal{R}(\hat{p}_{l-1}, \hat{u}_{l-1}), \mathcal{J}\begin{pmatrix}\hat{q}_{l-1}\\ \hat{v}_{l-1}\end{pmatrix})_{0,\Omega} = -(\mathcal{R}(p_l, u_l), \mathcal{J}\begin{pmatrix}\hat{q}_{l-1}\\ \hat{v}_{l-1}\end{pmatrix})_{0,\Omega} \tag{3.3}$$

for all $(\hat{q}_{l-1}, \hat{v}_{l-1}) \in Q_{l-1} \times V_{l-1}$ which is a variational problem that can be set up completely on level $l-1$. With this approach corrections on successively coarser levels can be computed recursively.

Obviously, this approach can not be used for nonlinear problems. Instead, we also need a coarse level approximation to the solution of the nonlinear problem at which the coarse level problem may be linearized. To this end, a projection

$$I_{l-1}: \quad Q_l \times V_l \to Q_{l-1} \times V_{l-1}$$

is needed as proposed in the original FAS scheme by Brandt [5]. A natural choice is to use standard interpolation at the coarse nodes for Q_{l-1} and averaging $n \cdot u_l$ along coarse edges for V_{l-1}. Below we present three implementations of multilevel projection methods for nonlinear least-squares formulations which resemble the original FAS idea. In all cases, if $(\hat{p}^\star_{l-1}, \hat{u}^\star_{l-1})$ denotes an approximation to $(\hat{p}_{l-1}, \hat{u}_{l-1})$, then the coarse grid correction for (p_l, u_l) is done by

$$(p_l^{\mathrm{new}}, u_l^{\mathrm{new}}) \;=\; (p_l + \hat{p}^\star_{l-1}, u_l + \hat{u}^\star_{l-1})$$

which is implemented using the standard interpolation operator which represents the embedding of (Q_{l-1}, V_{l-1}) in (Q_l, V_l) with respect to the nodal basis on level l.

3.1. FAS for the Normal Equations. The original FAS scheme in [5] was introduced in the context of nonlinear equations. As a first approach, we apply this FAS scheme to the nonlinear variational formulation (3.2). The corresponding variational problem for the correction on the coarse level $l-1$ consists in finding $(\hat{p}_{l-1}, \hat{u}_{l-1}) \in Q_{l-1} \times V_{l-1}$ such that

$$\begin{aligned}
&(\mathcal{R}(I_{l-1}(p_l, u_l) + (\hat{p}_{l-1}, \hat{u}_{l-1}))\,,\, \mathcal{J}(I_{l-1}(p_l, u_l) + (\hat{p}_{l-1}, \hat{u}_{l-1}))\begin{pmatrix} \hat{q}_{l-1} \\ \hat{v}_{l-1} \end{pmatrix})_{0,\Omega} \\
&= (\mathcal{R}(I_{l-1}(p_l, u_l))\,,\, \mathcal{J}(I_{l-1}(p_l, u_l))\begin{pmatrix} \hat{q}_{l-1} \\ \hat{v}_{l-1} \end{pmatrix})_{0,\Omega} \qquad (3.4)\\
&- (\mathcal{R}(p_l, u_l)\,,\, \mathcal{J}(p_l, u_l)\begin{pmatrix} \hat{q}_{l-1} \\ \hat{v}_{l-1} \end{pmatrix})_{0,\Omega}
\end{aligned}$$

holds for all $(\hat{q}_{l-1}, \hat{v}_{l-1}) \in Q_{l-1} \times V_{l-1}$. This means that during the computation on the coarse level, an approximation

$$I_{l-1}(p_l, u_l) + (\hat{p}^\star_{l-1}, \hat{u}^\star_{l-1}) \in Q_{l-1} \times V_{l-1}$$

to the solution of the nonlinear problem is stored rather than the correction $(\hat{p}^\star_{l-1}, \hat{u}^\star_{l-1})$ itself. Note that in the linear case, (3.4) is equivalent to the standard coarse level correction problem (3.3).

In general, the variational formulation (3.4) is no longer of the form (3.2), i.e., it does not come from a least-squares problem. Consequently, the variational problem for the coarse level correction does not have the form (3.4) for $l < L$ in a recursive call of the multilevel algorithm anymore. We may formally write (3.4) as

$$\begin{aligned}
&(\mathcal{R}(I_{l-1}(p_l, u_l) + (\hat{p}_{l-1}, \hat{u}_{l-1}))\,,\, \mathcal{J}(I_{l-1}(p_l, u_l) + (\hat{p}_{l-1}, \hat{u}_{l-1}))\begin{pmatrix} \hat{q}_{l-1} \\ \hat{v}_{l-1} \end{pmatrix})_{0,\Omega} \qquad (3.5)\\
&= f_{l-1}(\hat{q}_{l-1}, \hat{v}_{l-1})
\end{aligned}$$

for all $(\hat{q}_{l-1}, \hat{v}_{l-1}) \in Q_{l-1} \times V_{l-1}$ where f_{l-1} is some linear functional on $Q_{l-1} \times V_{l-1}$. This variational formulation allows the recursive application of the FAS coarse level correction scheme of the form (3.5) where

$$f_L(\hat{q}_L,\hat{v}_L) = (\mathcal{R}(p_L,u_L)\,,\,\mathcal{J}(p_L,u_L)\begin{pmatrix}\hat{q}_L\\ \hat{v}_L\end{pmatrix})_{0,\Omega}$$

and recursively, for $l = L, L-1, \dots, 1$,

$$f_{l-1}(\hat{q}_{l-1},\hat{v}_{l-1}) = (\mathcal{R}(I_{l-1}(p_l,u_l))\,,\,\mathcal{J}(I_{l-1}(p_l,u_l))\begin{pmatrix}\hat{q}_{l-1}\\ \hat{v}_{l-1}\end{pmatrix})_{0,\Omega} - f_l(\hat{q}_{l-1},\hat{v}_{l-1})\,.$$

3.2. A Least-Squares FAS-Scheme. This formulation starts from the FAS correction equation in analogy to [5] for the first-order system,

$$\mathcal{R}(I_{l-1}(p_l,u_l)+(\hat{p}_{l-1},\hat{u}_{l-1})) = \mathcal{R}(I_{l-1}(p_l,u_l)) - \mathcal{R}(p_l,u_l)$$

which is then handled by least-squares minimization. This leads to the problem of finding $(\hat{p}_{l-1},\hat{u}_{l-1}) \in Q_{l-1}\times V_{l-1}$ such that

$$\begin{aligned}&\|\mathcal{R}(I_{l-1}(p_l,u_l)+(\hat{p}_{l-1},\hat{u}_{l-1})) - \mathcal{R}(I_{l-1}(p_l,u_l)) + \mathcal{R}(p_l,u_l)\|_{0,\Omega}^2\\ &= \min_{\hat{q}_{l-1}\in Q_{l-1},\hat{v}_{l-1}\in V_{l-1}} \|\mathcal{R}(I_{l-1}(p_l,u_l)+(\hat{q}_{l-1},\hat{v}_{l-1})) - \mathcal{R}(I_{l-1}(p_l,u_l)) + \mathcal{R}(p_l,u_l)\|_{0,\Omega}^2\,.\end{aligned} \tag{3.6}$$

Taking variations in (3.6), we get that $(\hat{p}_{l-1},\hat{u}_{l-1}) \in Q_{l-1}\times V_{l-1}$ satisfies

$$\begin{aligned}&(\mathcal{R}(I_{l-1}(p_l,u_l)+(\hat{p}_{l-1},\hat{u}_{l-1}))\,,\,\mathcal{J}(I_{l-1}(p_l,u_l)+(\hat{p}_{l-1},\hat{u}_{l-1}))\begin{pmatrix}\hat{q}_{l-1}\\ \hat{v}_{l-1}\end{pmatrix})_{0,\Omega}\\ =&(\mathcal{R}(I_{l-1}(p_l,u_l))\,,\,\mathcal{J}(I_{l-1}(p_l,u_l)+(\hat{p}_{l-1},\hat{u}_{l-1}))\begin{pmatrix}\hat{q}_{l-1}\\ \hat{v}_{l-1}\end{pmatrix})_{0,\Omega}\\ &-(\mathcal{R}(p_l,u_l)\,,\,\mathcal{J}(I_{l-1}(p_l,u_l)+(\hat{p}_{l-1},\hat{u}_{l-1}))\begin{pmatrix}\hat{q}_{l-1}\\ \hat{v}_{l-1}\end{pmatrix})_{0,\Omega}\,,\end{aligned} \tag{3.7}$$

for all $(\hat{q}_{l-1},\hat{v}_{l-1}) \in Q_{l-1}\times V_{l-1}$. A comparison of (3.4) and (3.7) shows that FAS for the nonlinear variational formula (2.4) is not equivalent to FAS applied to the minimization problem (3.6), in general. Nevertheless, in the linear case the two formulations are both equivalent to the standard correction problem (3.3). This is due to the fact, that only the point of linearization in $\mathcal{J}$ differs in these two approaches. This scheme can also be written in the form (3.5) with recursively defined right-hand side

$$\begin{aligned}f_L(\hat{q}_L,\hat{v}_L) &= (\mathcal{R}(p_L,u_L)\,,\,\mathcal{J}(p_L+\hat{p}_L,u_L+\hat{u}_L)\begin{pmatrix}\hat{q}_L\\ \hat{v}_L\end{pmatrix})_{0,\Omega}\\ f_{l-1}(\hat{q}_{l-1},\hat{v}_{l-1}) &= (\mathcal{R}(I_{l-1}(p_l,u_l))\,,\,\mathcal{J}(I_{l-1}(p_l,u_l)+(\hat{p}_{l-1},\hat{u}_{l-1}))\begin{pmatrix}\hat{q}_{l-1}\\ \hat{v}_{l-1}\end{pmatrix})_{0,\Omega}\\ &\quad - f_l(\hat{q}_{l-1},\hat{v}_{l-1})\,,\; l = L, L-1, \dots, 1\,.\end{aligned}$$

The implementational effort is bigger than for the first approach as the right-hand side depends on the solution $I_{l-1}(p_l,u_l)+(\hat{p}_{l-1},\hat{u}_{l-1})$ of the coarse level problem. This means that during the solution process on the coarse level the right-hand side for the (local) Gauss-Newton iteration needs to be updated with the current iterate $I_{l-1}(p_l,u_l)+(\hat{p}^\star_{l-1},\hat{u}^\star_{l-1})$. This normally interdicts recursive calls of the multilevel routines, as the computations have to be done on different levels if more than two levels are involved. It is possible, however, to keep the recursive structure of the algorithm by interpolating the current iterate to the finest level L and assemble the right-hand side for the Gauss-Newton iterations there.

3.3. A Hybrid FAS Scheme. Finally, we propose a hybrid of the two approaches (3.5) and (3.7) presented above which combines advantages from both versions. It consists in computing $(\hat{p}_{l-1}, \hat{u}_{l-1}) \in Q_{l-1} \times V_{l-1}$ such that

$$\begin{aligned}
&(\mathcal{R}(I_{l-1}(p_l, u_l) + (\hat{p}_{l-1}, \hat{u}_{l-1}))\,,\, \mathcal{J}(I_{l-1}(p_l, u_l) + (\hat{p}_{l-1}, \hat{u}_{l-1}))\begin{pmatrix} \hat{q}_{l-1} \\ \hat{v}_{l-1} \end{pmatrix})_{0,\Omega} \\
=&(\mathcal{R}(I_{l-1}(p_l, u_l))\,,\, \mathcal{J}(I_{l-1}(p_l, u_l) + (\hat{p}_{l-1}, \hat{u}_{l-1}))\begin{pmatrix} \hat{q}_{l-1} \\ \hat{v}_{l-1} \end{pmatrix})_{0,\Omega} \qquad (3.8)\\
&-(\mathcal{R}(p_l, u_l)\,,\, \mathcal{J}(p_l, u_l)\begin{pmatrix} \hat{q}_{l-1} \\ \hat{v}_{l-1} \end{pmatrix})_{0,\Omega}\,,
\end{aligned}$$

holds for all $(\hat{q}_{l-1}, \hat{v}_{l-1}) \in Q_{l-1} \times V_{l-1}$. This scheme avoids the costly interpolation to the finest level by using $\mathcal{J}(p_l, u_l)$ in the second term on the right hand side. On the other hand, it is closer to the least-squares FAS scheme (3.7) than (3.5) since it updates the right-hand side with respect to the current coarse level iterate where this is possible without much computational cost.

4. Nonlinear Smoothing Iterations. For each of the three multilevel approaches presented in the previous section the situation on a given level l is as follows. Starting from an initial guess (p_l°, u_l°), we have to construct an approximation to the solution $(\hat{p}_l, \hat{u}_l) \in Q_l \times V_l$ of a variational problem of the form

$$(\mathcal{R}(p_l^\circ + \hat{p}_l, u_l^\circ + \hat{u}_l)\,,\, \mathcal{J}(p_l^\circ + \hat{p}_l, u_l^\circ + \hat{u}_l)\begin{pmatrix} \hat{q}_l \\ \hat{v}_l \end{pmatrix})_{0,\Omega} = f_l(\hat{q}_l, \hat{v}_l) \qquad (4.1)$$

for all $(\hat{q}_l, \hat{v}_l) \in Q_l \times V_l$. On level l, we want to employ Gauss-Seidel type relaxation methods based on a decomposition of the underlying finite element spaces

$$Q_l = \sum_{\nu=1}^{N_{l,p}} Q_{l,\nu} \quad , \quad V_l = \sum_{\nu=1}^{N_{l,u}} V_{l,\nu} \qquad (4.2)$$

(w.l.o.g. $N_{l,p} = N_{l,u} =: N_l$). Of course, this decomposition only makes sense if variational problems with respect to $Q_{l,\nu} \times V_{l,\nu}$ are easy to solve, i.e., these subspaces are low-dimensional with local support.

A fully nonlinear version of this Gauss-Seidel relaxation scheme consists in successively, for $\nu = 1, \ldots, N_l$, updating the approximation $(\hat{p}_l^\star, \hat{u}_l^\star) \in Q_l \times V_l$ by $(\hat{p}_l^\star + \delta p_l^{(\nu)}, \hat{u}_l^\star + \delta u_l^{(\nu)})$ where $(\delta p_l^{(\nu)}, \delta u_l^{(\nu)}) \in Q_{l,\nu} \times V_{l,\nu}$ solves the variational problem

$$\begin{aligned}
&(\mathcal{R}(p_l^\circ + \hat{p}_l^\star + \delta p_l^{(\nu)}, u_l^\circ + \hat{u}_l^\star + \delta u_l^{(\nu)})\,,\, \mathcal{J}(p_l^\circ + \hat{p}_l^\star + \delta p_l^{(\nu)}, u_l^\circ + \hat{u}_l^\star + \delta u_l^{(\nu)})\begin{pmatrix} \hat{q}_l \\ \hat{v}_l \end{pmatrix})_{0,\Omega} \\
&= f_l(\hat{q}_l, \hat{v}_l)
\end{aligned}$$

for all $(\hat{q}_l, \hat{v}_l) \in Q_{l,\nu} \times V_{l,\nu}$. However, the computational cost for solving the nonlinear variational problems associated with the subspaces $Q_{l,\nu} \times V_{l,\nu}, \nu = 1, \ldots, N_l$ to great accuracy is much too high. Therefore, these problems are replaced by the linearized variational problems

$$\begin{aligned}
&(\mathcal{R}(p_l^\circ + \hat{p}_l^\star, u_l^\circ + \hat{u}_l^\star) + \mathcal{J}(p_l^\circ + \hat{p}_l^\star, u_l^\circ + \hat{u}_l^\star)\begin{pmatrix} \delta p_l^{(\nu)} \\ \delta u_l^{(\nu)} \end{pmatrix}\,, \\
&\qquad \mathcal{J}(p_l^\circ + \hat{p}_l^\star, u_l^\circ + \hat{u}_l^\star)\begin{pmatrix} \hat{q}_l \\ \hat{v}_l \end{pmatrix})_{0,\Omega} = f_l(\hat{q}_l, \hat{v}_l)
\end{aligned}$$

for all $(q_l, v_l) \in Q_{l,\nu} \times V_{l,\nu}$. In the context of least-squares problems this constitutes one Gauss-Newton step for each of the local nonlinear problems. In contrast, applying Newton's method directly to these nonlinear variational problems would require the Fréchet derivative of $\mathcal{J}(\cdot,\cdot)$ with respect to $Q_{l,\nu} \times V_{l,\nu}$ which is costly and complicated (cf. [8, Sec. 10.1]). This relaxation scheme still requires that the Jacobian $\mathcal{J}(p_l^\circ + \hat{p}_l^\star, u_l^\circ + \hat{u}_l^\star)$ and the right-hand side be updated after each relaxation step. In order to avoid the computational cost associated with this, we collect "non-overlapping" local subspaces into clusters

$$\bigcup_{\nu=1}^{N_l^{(1)}} Q_{l,\nu} \quad , \quad \bigcup_{\nu=N_l^{(1)}+1}^{N_l^{(2)}} Q_{l,\nu} \quad , \quad \cdots \quad , \quad \bigcup_{\nu=N_l^{(k-1)}+1}^{N_l^{(k)}} Q_{l,\nu}$$

$(N_l^{(k)} = N_l)$. These k clusters are constructed such that a modification of the approximation in any of the local subspaces $Q_{l,\nu} \times V_{l,\nu}$ has no effect on the subproblems associated with the other local subspaces in this cluster. In other words, the basis functions belonging to the subspaces of each cluster have non-overlapping support. Consequently, updates to the Jacobian and the right-hand side need to be done only after a cluster has been completely traversed. With the specific space decomposition described in the next section, the number of Gauss-Newton linearizations is reduced to at most six.

The stiffness matrices for these linear variational problems may be assembled in the usual way by testing against (e.g. the canonical) basis functions of Q_l and V_l. This enables us to formulate the corresponding linearized variational formulations for the nonlinear problem on each level, and we may therefore use the FAS multilevel scheme. In order to make use of this scheme, we need a hierarchical decomposition of Q_L, V_L:

$$Q_L \;=\; \sum_{\nu=1}^{N_{L,0}+\ldots+N_{L,l}} Q_{l,\nu} \;,\; V_L \;=\; \sum_{\nu=1}^{N_{L,0}+\ldots+N_{L,l}} V_{l,\nu} \tag{4.3}$$

with $N_{L,l}\,, l = 0, \ldots, L$. A decomposition like this is easily gained during the refinement process, and the transfer between levels is given canonically by the sequence of spaces $Q_l\,,\, V_l$. The nonlinear relaxation method is embedded into a multilevel scheme like FAS by applying the projection method to the nonlinear least-squares correction formulation like (4.1). We specifically use the decomposition (4.3) where relaxation sweeps are only done on components which exclusively lie in the current space. Formulating the FAS correction equations according to (3.5),(3.7), (3.8), we end up with three variants of FAS multilevel projection methods.

5. Implementational Issues.

We assumed (2.1) to be uniquely solvable where the solution

$$(p, u) = (p^D + \hat{p}, u^N + \hat{u}) \;,\; (\hat{p}, \hat{u}) \in Q \times V$$

is at the same time the unique minimizer of the associated least-squares functional with zeros as its minimal value. However, the discrete solution

$$(p_h, u_h) = (p^D + \hat{p}_h, u^N + \hat{u}_h) \;,\; (\hat{p}_h, \hat{u}_h) \in Q_h \times V_h$$

of the variational formulation (2.4) will not result in a vanishing least-squares functional, in general, since the minimization of the least-squares functional is only done with respect to subspaces of finite dimension. In the course of the Gauss-Newton iteration, the least-squares functional contains components of both the discretization and the algebraic error. A reasonable stopping criterion for the Gauss-Newton method should cause the method to terminate as soon as the total error reaches the size of the discretization error as depicted in figure 5.1. The stagnation in reduction of the least-squares functional may serve as an indicator for reduction of the least-squares functional to the order of the size of the discretization error.

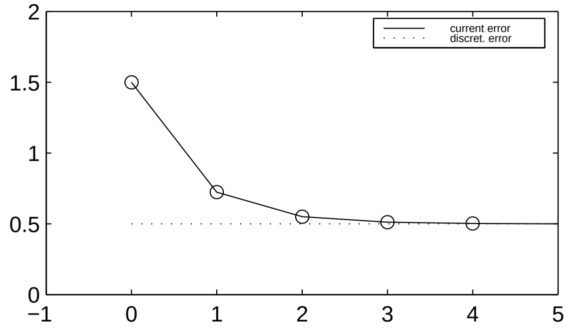

FIG. 5.1. *Discrete error reduction*

The computational experiments in the next section will use a criterion which compares the reduction of two consecutive iterations. If the product of these reduction ratios is below ϵ_s the iteration will be continued. We use an ϵ_s of 0.999 in our experiments. Another approach would be to use the reduction in the norm of the linearized right-hand side of the Gauss-Newton-step as a stopping criterion. The adaptive relaxation as described in the previous section had been implemented by using a colouring algorithm to determine non-overlapping domains in order to apply simultaneous smoothing steps on those domains during the projection relaxation method. It turned out that it is sufficient to use at most six independent sets of domains per level.

A closer look to the implementation of the different variants of FAS methods reveals that the computational effort in assembling the right-hand side of the linear system arising in each Gauss-Newton-iteration differs. A comparison of the number of flops for each method will therefore also be included in the numerical results in the next section.

We still have to find a suitable space decomposition according to (4.2). As we have to deal with subspaces of $\mathrm{H}(\mathrm{div},\Omega)$, we have to handle the div-free components in this space. These components have a deteriorating effect on the condition numbers of the matrices of the linearized system and therefore cause poor convergence. A decomposition described in [2] suggests a block decomposition which may be indicated using the vertices of the triangulation. For each vertex, the decomposition of $Q_h \times V_h$ then consists of the associated nodal basis function in Q_h and of those basis functions in V_h which are associated with edges starting at this vertex. This decomposition may be regarded as a domain decomposition which fulfills a finite overlap condition. We divide the vertices into independent subsets such that we have no overlap within each of them in order to be able to compute simultaneously on all domains in each subset. This targets on solving the linear problems using a block Gauss-Seidel scheme. Moreover, in view of the adaptive setting in the previous section, we only include a subspace on level l if it differs from the subspace associated with the same vertex on the next coarser level $l-1$.

6. Computational Experiments. In this section, we present numerical results with the different adaptive FAS-type multilevel algorithms described above for a nonlinear least-squares problem arising from the time discretization of a variably saturated subsurface flow problem. Given a domain $\Omega \subset \mathbb{R}^2$, the mass conservation equation, expressed in terms of the pressure variable p, reads

$$\partial_t \, \theta(p) \; - \; \text{div}(K(\theta(p))\nabla p) \; = \; 0. \tag{6.1}$$

Introducing the flux variable u using the Darcy-Buckingham law

$$u + K(\theta(p))\nabla p \; = \; 0, \tag{6.2}$$

which is assumed to be valid in the porous media context, we end up with a time-dependent system of first-order equations. θ and K are water content and permeability as described in Section 2. We use an implicit Euler discretization with steplength τ to approximate the derivative in time $\partial_t\theta$:

$$\begin{aligned} \frac{\theta(p) - \theta(p^{\text{old}})}{\tau} + \text{div}\, u \; &= \; 0 \\ u + K(\theta(p))\nabla p \; &= \; 0 \end{aligned} \tag{6.3}$$

As an example for suitable tests, we use a water table recharge problem as described in [17] which models infiltration into initially dry sand. The computational domain is a box of 3 m length and 2 m depth as sketched in Figure 6.1.

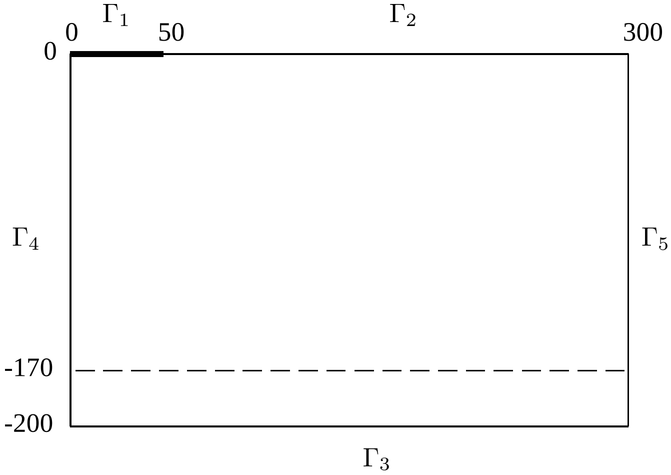

FIG. 6.1. *Test example of a two-dimensional water table recharge problem*

At time $t = 0$ we assume a constant hydraulic potential at -1.7 m (which is the position of the groundwater table). For $t > 0$ we have constant infiltration of 0.148 m/h through the boundary segment Γ_1. Zero flux boundary conditions are prescribed at the boundary segments Γ_2, Γ_3 and Γ_4, while the hydraulic potential is held constant at -1.7 m at the right boundary Γ_5.

We use the model by van Genuchten [16] with the parameters listed in [7] for the functions $K(p)$ and $\theta(p)$. For some types of soil (e.g. sand), this parametrization leads to unbounded $K'(p)$ in the saturated limit which causes problems for the Newton iteration. Following a suggestion in [9, Chap. 6], we replace $K(p)$ by a cubic spline interpolant in a neighborhood of the saturated limit (for $0 \leq z - p \leq 10^{-2}$ where z is the height). We also found it useful to limit the permeability from below by $10^{-3}K_s$.

Our interest is in the performance of the different nonlinear multilevel solvers. For fixed $\tau = 0.1$, we pick the step from 2.9 hours to 3 hours from the simulation and concentrate on the solution of the corresponding nonlinear elliptic problem (6.3) at this time step. All multilevel methods will be compared to a single grid method, where only the Gauss-Seidel block relaxation is done as described before for a space decomposition according to all points belonging to the finest grid. That is, the single-grid method does not use any kind of multigrid correction. In order to make use of the least-squares functional as an error estimator, we first observe that the least-squares functional may be computed by

$$||\mathcal{R}(p^D + \hat{p}_h, u^N + \hat{u}_h)||^2_{0,\Omega} = \sum_{T \in \mathcal{T}_h} ||\mathcal{R}(p^D + \hat{p}_h, u^N + \hat{u}_h)||^2_{0,T} =: \sum_{T \in \mathcal{T}_h} \eta_T,$$

where $T \in \mathcal{T}_h$ is one single element in the triangulation $\mathcal{T}_h$. As the total error in the least-squares functional therefore consists of local contributions η_T, we decide to refine those triangles with

$$\eta_T \geq \frac{\epsilon_e}{\#T},$$

where $\#T$ equals the number of triangles in the triangulation, in order to gain an overall accuracy of ϵ_e.

In our experiments we used five levels of adaptive refinement within a full multigrid scheme to provide suitable initial iterates for the next finer grid. Therefore, all calculations start with a "single-grid" computation on level $l = 0$. Table 6.1 shows the number of degrees of freedom (for u_h and for p_h) and the minimum of the least-squares functional on the different levels.

TABLE 6.1
Degrees of freedom and least-squares functional on different levels after 3 hours

	$\dim V_h$	$\dim Q_h$	$\|\mathcal{R}(p^D + \hat{p}_h, u^N + \hat{u}_h)\|^2_{0,\Omega}$
$l = 0$	343	127	$2.14 * 10^{-4}$
$l = 1$	645	231	$8.79 * 10^{-5}$
$l = 2$	1246	435	$3.79 * 10^{-5}$
$l = 3$	2494	854	$1.38 * 10^{-5}$
$l = 4$	3996	1358	$5.57 * 10^{-6}$
$l = 5$	6128	2072	$2.81 * 10^{-6}$

Figure 6.2 shows the adaptively refined triangulation for this time where the infiltration front can easily be recognized on the left of the domain. In the rest of the domain no refinement happens as the soil is dry in this area and flux is negligible there. We first turn our attention to the number of relaxation sweeps needed for convergence of the single grid algorithm and the three multilevel variants (FAS for the normal equations $\approx$ ML-1, hybrid FAS $\approx$ ML-2, least-squares FAS $\approx$ ML-3).

While the number of single grid iterations obviously grows as the mesh is refined, the three multilevel methods seem to have a bounded number of sweeps. Recalling that the stopping criterion for the Gauss-Newton iteration was based on the stagnation in the reduction in

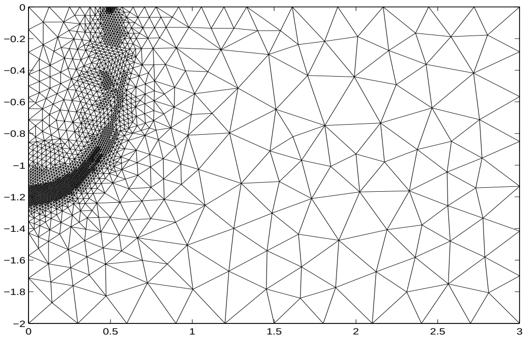

FIG. 6.2. *Adaptively refined triangulation*

TABLE 6.2
Number of relaxation sweeps for the different solving methods

	single-grid	ML-1	ML-2	ML-3
$l = 0$	16	-	-	-
$l = 1$	9	6	6	6
$l = 2$	13	6	6	6
$l = 3$	31	10	10	10
$l = 4$	61	13	13	13
$l = 5$	62	12	12	12

the least-squares functional, we observe that while this reduction is disturbed for the single-grid case, we have a fast reduction to the discretization error for the multilevel algorithms. We may conclude that the single-grid convergence is obstructed by coarse-grid parts of the remaining error, while these parts are well represented and well resolved on the coarser grids. A first weak point of the applied stopping criterion is revealed by looking at the achieved minimal least-squares functional of each method (in percent of achieved single grid functional) shown in Table 6.3.

TABLE 6.3
Percentage of achieved single-grid functional

	single-grid	ML-1	ML-2	ML-3
$l = 1$	100	99.89	99.89	99.91
$l = 2$	100	100.21	100.21	100.2
$l = 3$	100	99.71	99.71	99.72
$l = 4$	100	98.83	98.83	98.82
$l = 5$	100	97.89	97.89	97.87

Ignoring the results for $l = 2$ for the moment, we observe that the iteration for the single-grid algorithm is stopped before the discretization error is reached. This means that there exist coarse-grid parts of the error, which are not smoothed out during the projective relaxation or at best damped at a rate greater than $\epsilon_s = 0.999$. On the other hand, these components should be slashed by multilevel methods and of course the ML-x-algorithms result in a smaller functional here using less relaxation sweeps (see Table 6.2). Another conclusion is that the stagnation in reduction of the functional does not necessarily lead to a small algebraical error, which may explain level-2 results, and so this stopping criterion needs to be modified. Taking Table 6.3 into account, we now compare the number of relaxation sweeps that the multilevel algorithms required in order to end up with a functional of the same size as the one that the single-grid algorithm produced.

TABLE 6.4
Number of relaxation sweeps to reach single-grid ls-functional

	single-grid	ML-x
$l = 1$	9	4
$l = 3$	31	6
$l = 4$	61	6
$l = 5$	62	4

With Table 6.4, we are ready to compare the amount of work done by the computer. As we already mentioned above, the computational cost widely differs for the four solving algorithms, so that a comparison of the reduction in the functional to the amount of work in flops is indispensable. Let us shortly recall that reduction in the least-squares functional only means reduction of the algebraic error, not reduction of the discretization error (as given in Table 6.1), but the transfer to a finer grid of the same solution will result in a larger starting least-squares functional as this discretization provides a more accurate measure of the true error. This is due to the fact that the functions for saturation and permeability are linearly interpolated to different resolutions on different levels. The three graphs in Figure 6.3 show the reduction of the least-squares functional versus the computational cost (multilevel methods dashed).

The jumps in the graphs represent the stronger measure for the least-squares functional on the next finer grid. So we concentrate on the achieved reduction in the functional (y-axis) compared to the computational work (x-axis). According to the considerations in Section 3, we observe the high cost of the ML-3 algorithm against the cheap ML-1 algorithm, where only one point of linearization has to be updated for the right-hand side of the Gauss-Newton system. Obviously, the computational cost is smaller in the multilevel algorithms ML-1 and ML-2, whereas the ML-3 algorithm has a hard stand against the cost of the many single-grid relaxations. Nevertheless, if we have the last table in mind, the ML-3 ansatz is still competitive. However, the first time that the ML-1 algorithm does pay off is on the third level and therefore this one looks much more effective. On the other hand, the ML-2 algorithm does not give significant advantage for the additional cost. If we restricted the number of iterations to the numbers in the last table, all multilevel methods show a satisfying behaviour and only differ in the computational cost per iteration. Nevertheless we should keep in mind that the stopping criterion not only (indirectly) depends on the remaining error but also on the actual error reduction. So we need another criterion to compare the performance of the multilevel algorithms. In Section 5 we mentioned that another stopping criterion could be given using the Euclidean norm of the right-hand side of the Gauss-Newton system. Consequently, we should now compare this “residual quantity” during the iterative process of our solving methods.

The graphs in Figure 6.4 show a comparison of the three variants in terms of residual norm vs. number of iterations.

The full multigrid scheme seems to provide good estimates for the next levels as the starting norm of the right-hand side is reduced on each level. Again the multilevel methods show a very good performance, whereas the single-grid convergence slows down during the iteration. This confirms the former speculation that remaining coarse grid components of the error, here summed up in form of the norm of the right-hand side of the correction system, are

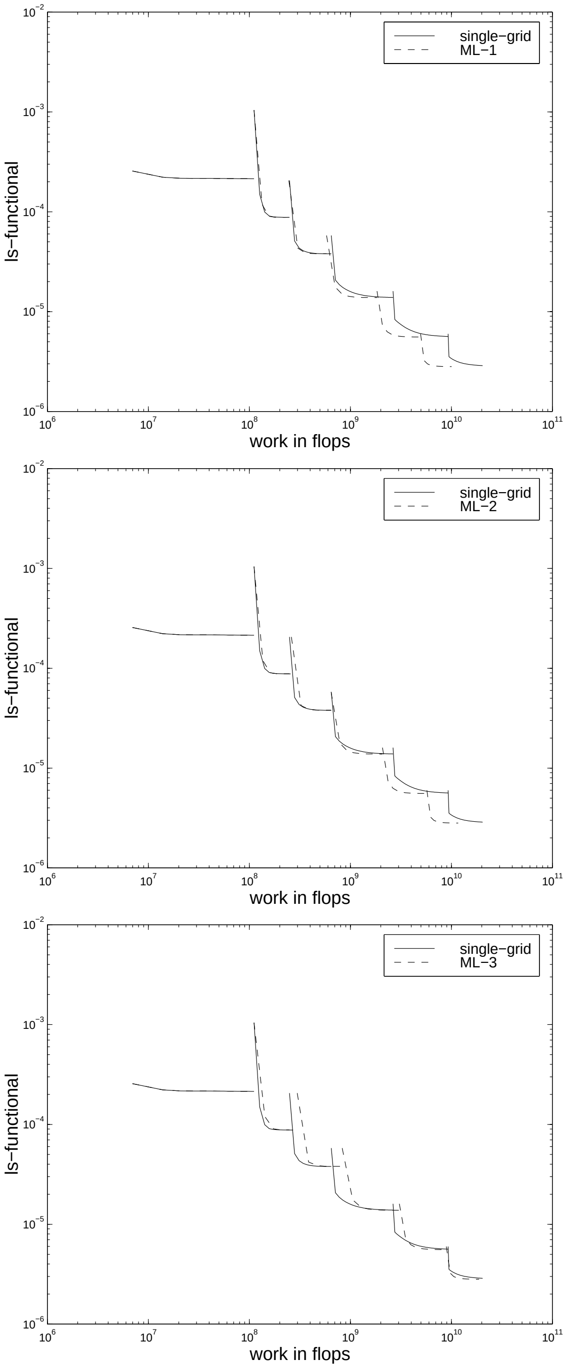

FIG. 6.3. *Functional vs. cost*

the explanation for retarding effects in convergence. Using a stopping criterion based on this reduction with a fixed maximal error ϵ_r of about 10^{-4}, we found that the multilevel methods

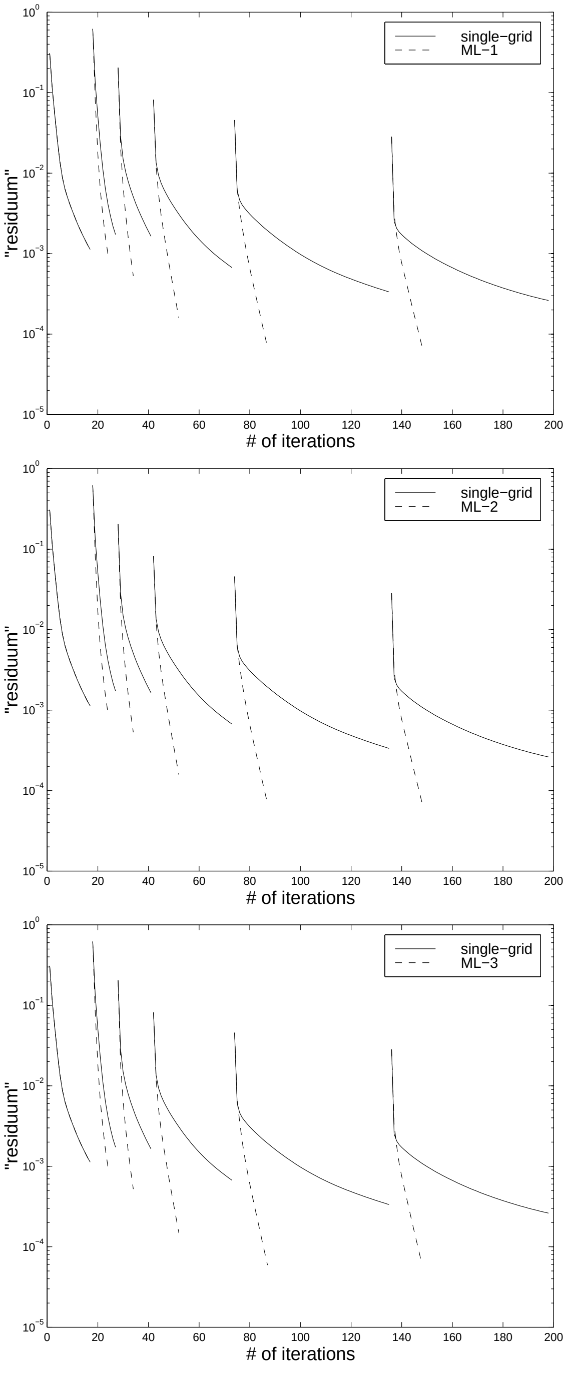

FIG. 6.4. *Residual norm vs. iterations*

still presented a good convergence performance, while the single-grid algorithm resulted in unaffordably many relaxation sweeps. The graphs in Figure 6.5 show that the nonlinear coarse-level correction really has a major effect in convergence.

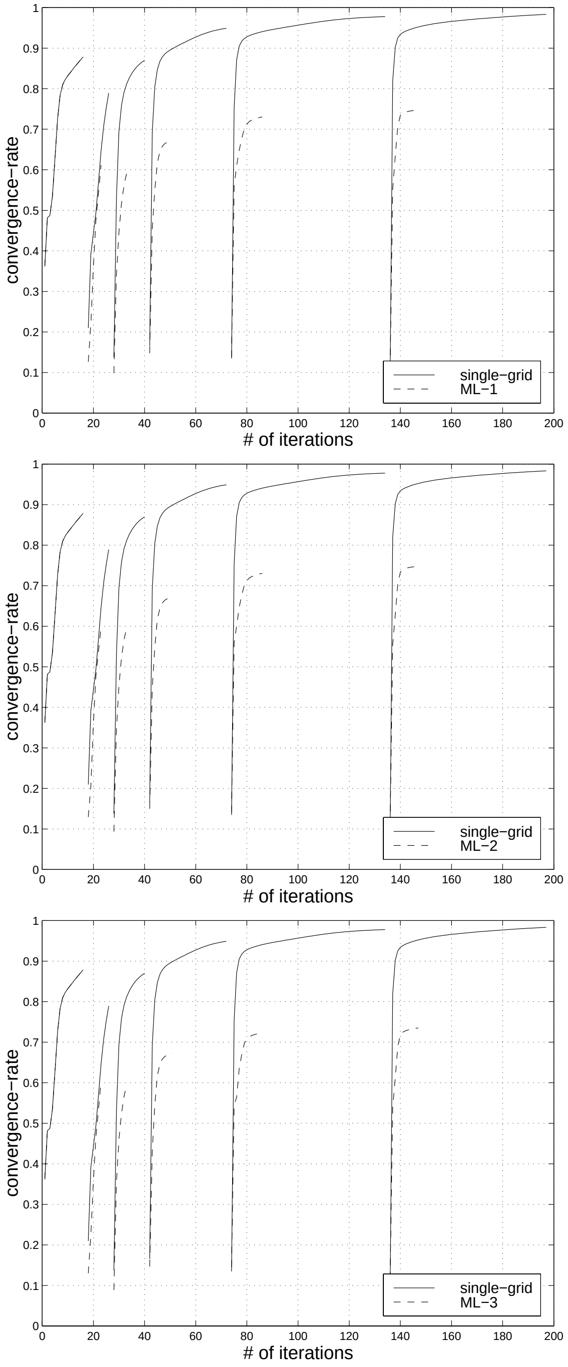

FIG. 6.5. *Convergence rate*

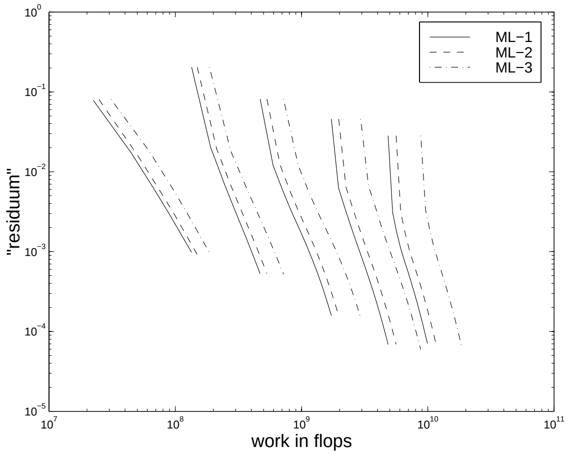

FIG. 6.6. *Comparison of residuals*

TABLE 6.5
Number of relaxation sweeps, second stopping criterion

	single-grid	ML-1	ML-2	ML-3
$l = 0$	6	-	-	-
$l = 1$	23	19	19	19
$l = 2$	55	15	15	15
$l = 3$	122	16	16	16
$l = 4$	$>$200	18	18	17
$l = 5$	$>$200	19	19	19

Although the single-grid convergence factors tend to 1, the multilevel variants show pretty good rates which seem - as is typical for multilevel methods - to be bounded away from 1 independently of the mesh size. Discrepancies between the ML-x variants first appear on the last two levels, so the convergence rates for these levels are given in this picture. We observe only slight differences between the ML-1 and ML-2 method, whereas the ML-3 variant shows the best rates. Figure 6.6, however, which additionally takes into account the computational cost, shows that the ML-1 algorithm is the cheapest algorithm to achieve small “residuals” ($l = 0$ left out for simplicity). We can obviously draw the conclusion from this figure that on all levels the ML-1 algorithm is the cheapest way to reduce the “residual”. For the next results, we use the reduction of the linearized right-hand side as a stopping criterion with a minimal tolerance of $\epsilon_r = 0.00001$. To limit the computation time, we bound the maximum number of iterations by 200. In Table 6.5 we have a look on the number of relaxation sweeps until this new stopping criterion is fulfilled. These results confirm the favourable performance of the multilevel methods in contrast to the number of single-grid relaxations which about doubles from level to level. We finally compare the computational costs of the single-grid and the ML-1 method in Figure 6.7.

Two conclusions can be drawn from Figures 6.7 and 6.3. First, the second stopping criterion also provides the reduction of the least-squares functional nearly to the size of the discretization error. Second, using this criterion the computational costs for applying nonlinear coarse-grid corrections already pay out on the second level and the improvement becomes

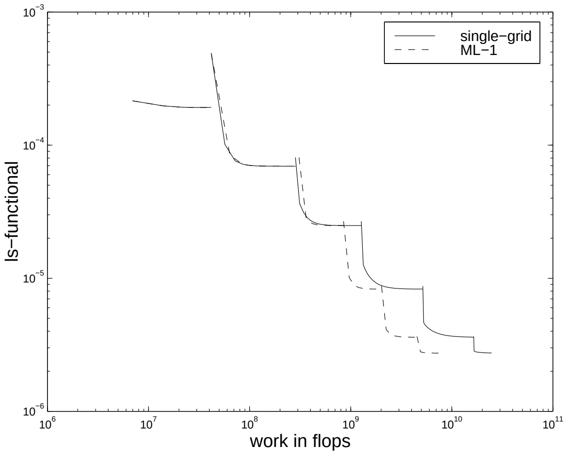

FIG. 6.7. *Computational costs, ML-1 algorithm, second stopping criterion*

obvious on the third level.

7. Concluding Remarks. The main result of this paper is the applicability of nonlinear multilevel methods for systems arising from least-squares reformulations of partial differential equations by achieving typical multilevel behaviour of the convergence history (level independence, well bounded away form one). As pointed out in Section 3 the different approaches of using the FAS-scheme on the normal equations of the arising linear Gauss-Newton system and the ansatz to apply least-squares on the FAS equations for the first-order system only differ in a varying point of linearization in calculating the right-hand side of the variational formulation. So the good and nearly equal performance of either method — in terms of iterations — results in a need for the comparison of the computational cost. It turns out that the update of the linearization point does not improve the convergence rates by much. The first stopping criterion — using the reduction of the discretized least-squares functional — is detected to be not suitable for single-grid computations. Therefore a second stopping criterion is introduced which takes into account the reduction of the norm of the right-hand side of the Gauss-Newton correction system. Using this criterion, the good convergence results for the nonlinear multilevel methods can be confirmed and the obstacles for single-grid-convergence quantified. Using this criterion it will also be easier to compare the results in this article to the results from [14].

REFERENCES

[1] H. W. ALT AND S. LUCKHAUS, *Quasilinear elliptic-parabolic differential equations*, Math. Z., 183 (1983), pp. 311–341.

[2] D. N. ARNOLD, R. S. FALK, AND R. WINTHER, *Preconditioning in $H(div)$ with applications*, Math. Comp., 66 (1997), pp. 957–984.

[3] P. B. BOCHEV AND M. D. GUNZBURGER, *Finite element methods of least-squares type*, SIAM Review, 40 (1998), pp. 789–837.

[4] D. BRAESS, *Finite Elements*, Cambridge University Press, Cambridge, 1997.

[5] A. BRANDT, *Multi-level adaptive solutions to boundary-value problems*, Math. Comp., 31 (1977), pp. 333–390.

[6] Z. CAI, R. LAZAROV, T. A. MANTEUFFEL, AND S. F. MCCORMICK, *First-order system least squares for second-order partial differential equations: Part I*, SIAM J. Numer. Anal., 31 (1994), pp. 1785–1799.

[7] R. F. Carsel and R. S. Parrish, *Developing joint probability distributions of soil water retention characteristics*, Water Resources Research, 24 (1988), pp. 755–769.

[8] J. E. Dennis and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, SIAM, Philadelphia, 1996.

[9] J. Fuhrmann, *Zur Verwendung von Mehrgitterverfahren bei der numerischen Behandlung elliptischer partieller Differentialgleichungen mit variablen Koeffizienten*, PhD thesis, Technische Universität Chemnitz-Zwickau, Aachen, 1995.

[10] W. Hackbusch, *Multi-Grid Methods and Applications*, Springer, Berlin, 1985.

[11] R. Helmig, *Multiphase Flow and Transport Processes in the Subsurface*, Springer, Berlin, 1997.

[12] S. F. McCormick, *Multilevel Projection Methods for Partial Differential Equations*, SIAM, Philadelphia, 1992.

[13] Y. Mualem, *A new model for predicting the hydraulic conductivity of unsaturated porous media*, Water Resources Research, 12 (1976), pp. 513–522.

[14] G. Starke, *Gauss-Newton multilevel methods for least-squares finite element computations of variably saturated subsurface flow*, Computing, (1999). In Press.

[15] ———, *Least-squares mixed finite element solution of variably saturated subsurface flow problems*, SIAM J. Sci. Comput., 20 (1999). In Press.

[16] M. T. van Genuchten, *A closed-form equation for predicting the hydraulic conductivity of unsaturated soils*, Soil Sci. Soc. Am. J., 44 (1980), pp. 892–898.

[17] M. Vauclin, D. Khanji, and G. Vachaud, *Experimental and numerical study of a transient, two-dimensional unsaturated-saturated water table recharge problem*, Water Resources Research, 15 (1979), pp. 1089–1101.