Preprint
UCRL-JC- 150513

A Multigrid Method for Nonlinear Unstructured Finite Element Elliptic Equations

Miguel A. Dumett, Panayot Vassilevski, and Carol S. Woodward.

This article was submitted to
SIAM Journal on Scientific Computing

U.S. Department of Energy

Lawrence
Livermore
National
Laboratory

October 2002

DISCLAIMER

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

This is a preprint of a paper intended for publication in a journal or proceedings. Since changes may be made before publication, this preprint is made available with the understanding that it will not be cited or reproduced without the permission of the author.

A MULTIGRID METHOD FOR NONLINEAR UNSTRUCTURED FINITE ELEMENT ELLIPTIC EQUATIONS[*]

MIGUEL A. DUMETT, PANAYOT S. VASSILEVSKI, AND CAROL S. WOODWARD [†]

Abstract. This paper presents an application of the element agglomeration-based coarsening procedure (agglomeration AMGe) proposed in [10], to build the components of a multigrid method for solving nonlinear finite element elliptic equations on general unstructured meshes. The agglomeration-based AMGe offers the ability to define coarse elements and element matrices, provided access to elements and element matrices on the fine grid is available. We focus on the performance of the classical full approximation scheme (FAS). In the present context the coarse nodes are constructed algebraically based on the element agglomeration, and the interpolation rules are based on the (linear) AMGe exploiting element matrices of Laplace operator and L_2-mass element matrices. The AMGe provides the coarse counterparts on all levels. The nonlinear coefficients are averaged over the coarse elements, which leads to non-inherited forms and hence to non-inherited multigrid methods. Numerical results show that the resulting nonlinear multigrid gives mesh independent convergence on model problems. In addition, the nonlinear multigrid scheme appears to be more efficient and robust for poor initial guesses than repeated applications of the nonlinear system smoother (i.e., single level method). Finally, our numerical results indicate that handling nonlinearities on coarse grids can provide an advantage over nonlinear solvers that handle nonlinearities only on the original problem grid.

Key words. : nonlinear elliptic equations, unstructured meshes, finite elements, FAS, algebraic multigrid, inexact Newton, Picard, AMGe.

AMS subject classifications. : 65N30, 65N55.

1. Introduction.

This paper addresses nonlinear elliptic equations discretized on generally unstructured meshes using finite elements. The unstructured finite elements are widely used in practice because of their better adaptation to geometrical or coefficient irregularities of the elliptic operator. Therefore, unstructured meshes are not generally obtained by successive steps of refinement; coarse meshes, if needed, must be generated algebraically. We intend to investigate the performance of the classical full approximation scheme (FAS) [1] applied to the specified class of nonlinear elliptic equations. We will compare FAS with some standard nonlinear schemes, like inexact Newton and Picard. This paper should be viewed as a preliminary first study on this subject to assess the potential of the developed nonlinear multigrid method. More detailed and sophisticated comparison is yet to be performed. The approach of generating coarse nonlinear problems taken in this paper can be viewed as an extension of the element agglomeration AMGe proposed in [10]. This extension of the (linear) agglomeration-based AMGe provides all components for a nonlinear multigrid (coarse grids, interpolation rules, and coarse nonlinear operators). The resulting FAS algorithm will be referred to herein as the FAS-AMGe method. A preliminary version of the method was reported in [11]. Note that our methods of extending the AMGe framework are also relevant for the nonlinear multigrid method (NMGM) of Hackbusch [8]. Extending our work to this method is a possible topic of future work.

On structured grids, the coarsening strategy, the grid transfer operators and the coarse grid operator (needed by any multigrid method) can be defined in a straightforward geometric way for both linear and nonlinear multigrid algorithms. The hierarchy

[*] This work was performed under the auspices of the U.S. Department of Energy by University of California Lawrence Livermore National Laboratory under contract number W-7405-Eng-48.

[†] Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, Livermore, CA 94551; (dumett1@llnl.gov, panayot@llnl.gov, cswoodward@llnl.gov).

of grids, the grid transfer and the coarse grid operators can be constructed independently of the (linear or nonlinear) problem considered. This hierarchy is actually naturally (and uniquely) defined by the sequence of nested finite element spaces used. However, in certain applications (like strongly anisotropic elliptic PDEs) it has been found that in order to obtain a reasonable multigrid rate of convergence it is helpful to introduce operator-dependent interpolation and to define the coarse operators algebraically through Galerkin relations. This argument applies to nonlinear problems as well. On unstructured grids the selection of the hierarchy of grids demands special algorithms like nested-mesh subdivision, overset meshes, or agglomeration methods [12]. An adequate definition of grid transfer operators and coarse grid operators is a serious issue because it depends on the hierarchy of grids.

Other nonlinear multigrid methods have been introduced (for instance, [13], [14], [16]) for solving nonlinear problems defined on unstructured meshes. Both Mavripilis [13], [14] and Stals [16] use an FAS algorithm defined on generated grids obtained using geometrical information. Furthermore, the particular definition of their grids triggers the characterization of their intergrid transfer operators. The algorithm we present in this paper can also be seen as an FAS algorithm defined on a particular hierarchy of grids. These three nonlinear multigrid algorithms differ on what is assumed from the discretization and the amount of geometric information taken from the grid.

For instance, Mavriplis [13] assumes an unstructured triangular mesh of control volumes and uses, in the coarsening strategy, an agglomeration method introduced in [12]. Agglomeration is based on fusing together neighboring fine grid control volumes to form large coarse grid control volumes. Prolongation operators are taken as piecewise constant, and restriction operators are taken as the transposes of interpolation operators. These grid transfer operators are used because the coarse cells have complex shapes, and more sophisticated interpolation operators are not easily constructed. These interpolation operators, as is well known, trigger lack of accuracy and degradation of the multigrid performance [12]. To improve this less-accurate interpolation strategy, an implicit prolongation operator is proposed in [12]. This new approach for interpolation resembles a fine node iteration, as in classical algebraic multigrid (AMG) [17], while fixing values at coarse nodes.

In [16], Stals considers triangular finite element discretization and geometric refinement. More specifically, the construction of the hierarchy of grids is based on the newest vertex bisection algorithm done adaptively; that is, the triangles are divided into two by bisecting the edge opposite to the newest vertex. Since grids are recursively refined, the prolongation operators are defined appropriately according to geometric information. In particular, every time the value of a given node changes, the local stiffness matrices have to be reevaluated (this involves a search through the data structure to find the appropriate triangles, the formulation of the basis functions and the evaluation of the integral).

In this paper, an AMGe-based FAS algorithm is presented where the intergrid transfer operators are not defined using any geometrical information. Instead, the intergrid transfer operators require access to the local element matrices. When the finite element method (FEM) is used to discretize a linear elliptic partial differential equation (PDE), not only is the relation between elements and nodes available, but the local stiffness matrices are also available for each element. Several AMGe methods have been introduced in [2], [10] and [5]. We focus on the simplest one, which provides a systematic generation of coarse element information, namely, the element agglomeration AMGe proposed in [10]. This algorithm creates coarse grids from vertices of agglomerated elements, which is the minimal coarse grid one can use. Richer coarse grids are possible but more expensive to construct (see [5]). The interpolation is defined by energy minimization; that is, a quadratic functional is minimized related to a local neighborhood matrix assembled from element matrices forming the neighborhood of interest (for details, see [10]).

The main goal of this paper is to apply the AMGe based on element agglomeration, developed for finite element discretizations on unstructured grids, in building effective components for nonlinear multigrid methods. Our objective is to show how to construct coarse versions of the nonlinear finite element problems and respective intergrid transfer operators so that nonlinearities may be handled directly on coarse grids (without visiting the fine grid). Our numerical tests indicate that the studied FAS-AMGe method exhibits a mesh-independent convergence rate for model problems. The FAS-AMGe method offers a reasonable alternative to the more standard inexact Newton multigrid or inexact Picard multigrid methods (in which coarse nonlinear problems are not required on coarser grids). Note that the inexact Newton method applied to nonlinear elliptic PDEs requires solving generally non-symmetric linear systems, and this requirement may cause difficulties for a linear solver. In addition, the initial nonlinear iterate may be too far away from the exact solution for efficient convergence. In this respect, inexact Picard is more appealing since the linear systems that need to be solved (even inexactly) are symmetric positive definite (for the model class of nonlinear PDEs considered in this paper).

The remainder of this paper is structured as follows: Section 2 contains a detailed description of the components of FAS and the FAS algorithm itself. In Section 3 we specify the model nonlinear problem of interest and its finite element discretization. Section 4 summarizes the agglomeration-based AMGe algorithm from [10] and its extension to nonlinear finite element elliptic problems. Section 5 contains numerical results illustrating the performance of the FAS-AMGe for several relaxation methods and shows a comparison with some other nonlinear solvers. Finally, some conclusions are drawn in Section 6.

2. The FAS Algorithm.

The FAS algorithm has been developed to solve nonlinear discrete problems of the form

$$F(u) = f, \quad \text{on } \Omega, \tag{2.1}$$

where Ω is a given mesh (in vector notation Ω is just a set of integer indices) [1]. In our application, the above discrete problem will represent a set of nonlinear equations coming from the finite element discretization of a nonlinear elliptic PDE on a general unstructured mesh. The more standard notation, $\mathcal{F}(u) = 0$, is sometimes used where $\mathcal{F}(u) = f - F(u)$.

The following components are needed to define a nonlinear multigrid method for solving (2.1). Because a hierarchy of grids may not be available (as in the unstructured mesh case), the definition of this hierarchy must be included as a multigrid component.

These components are:

1. **The coarsening strategy:** Given the original mesh Ω, the coarsening strategy constructs a hierarchy of grids

$$\Omega^l \subset \Omega^{l-1} \subset \cdots \subset \Omega^1 \subset \Omega^0 = \Omega. \tag{2.2}$$

2. **The grid transfer operators (prolongation and restriction):** Given the hierarchy of grids (2.2), it is necessary to define three collections of operators

$$\begin{array}{llcll} \textit{interpolation operators:} & P^j_{j+1} : \Omega^{j+1} & \longrightarrow & \Omega^j, & 0 \le j \le l-1, \\ \textit{restriction operators:} & R^{j+1}_j : \Omega^j & \longrightarrow & \Omega^{j+1}, & 0 \le j \le l-1, \\ \textit{injection operators:} & \tilde{R}^{j+1}_j : \Omega^j & \longrightarrow & \Omega^{j+1}, & 0 \le j \le l-1. \end{array} \tag{2.3}$$

   These operators transfer functions (vectors) defined on a particular grid Ω^j to a neighboring one. There are two restriction operators: the first one R^{j+1}_j is used to transfer residuals, whereas $\tilde{R}^{j+1}_j$ is used to transfer iterates.
3. **The cycle form:** The spirit of multigrid is to solve the nonlinear problem $F(u) = F^0(u^0) = f^0 = f$ in Ω^0 using a sequence of approximate solutions of nonlinear problems

$$F^j(u^j) = f^j, \ 1 \le j \le l, \tag{2.4}$$

   on consecutive grids of (2.2). Once the hierarchy of grids in (2.2) is constructed, it is necessary to indicate the order in which the sequence of grids is visited and how many times a grid is visited. Also, once reaching a given grid it should be determined whether or not the current level nonlinear problems will be relaxed (by few steps of simple iterations).
4. **The sequence of nonlinear problems:** Given the nonlinear operator $F(u)$ and the right hand side f, coarse nonlinear operators and the respective right-hand sides in (2.4) are needed. Very often (for linear problems this is standard) one defines the coarse operators variationally via the Galerkin relation,

$$R^{j-1}_j \dots R^1_0 F^0(P^0_1 \dots P^{j-1}_j u_j).$$

   In the nonlinear case, though, such a definition implies computing coarse residuals by visiting the finest grid Ω^0, and this definition will not lead to an optimal complexity method. If geometric information is available one can construct nonlinear operators directly by rediscretizing the PDE on the coarse grid. Such a procedure, however, does not generally lead to coarse nonlinear operators that are variationally obtained from the fine grid operators.
5. **The smoothing procedure:** Given the nonlinear equations (2.4), a relatively simple nonlinear iterative method needs to be applied to relax them. In practice, one may use any adequate iterative method applied to the nonlinear problem.

Having the multigrid components in place, a standard FAS V-cycle algorithm reads:

Algorithm 2.1 (FAS V-cycle). *Given an initial* 0*th level iterate* v_0^0 *for the exact solution* u^0 *of* $F^0(u^0) = f^0$ *on* Ω^0.

```
• Apply a relaxation method to $F^0(u^0) = f^0$ (with initial iterate $v_0^0$), $\nu_1$ times
  on $\Omega^0$ to get the new $0$th level iterate $v^0 = v^0_{\nu_1}$.
• Given $v^0$, compute the residual $r^0 = f^0 - F^0(v^0)$.
    • Restrict the residual $r^1 = R^1_0 r^0$.
    • Restrict (inject) the iterate $v^0$ to get $v^1 = \tilde{R}^1_0 v^0$.
    • Compute $f^1 = F^1(v^1) + r^1$.
    • Relax on $F^1(u^1) = f^1$ (with initial iterate $v^1$), $\nu_1$ times to get $v^1 = v^1_{\nu_1}$.
    • Given $v^1$, compute the residual $r^1 = f^1 - F^1(v^1)$.
        • .
          .
          .
            • Solve the coarsest nonlinear equation $F^l(u^l) = f^l = r^l + F^l(v^l)$
              for $u^l$ (with initial iterate $v^l = \tilde{R}^l_{l-1} v^{l-1}$).
            • Compute the $l$-correction $e^l = u^l - v^l$.
        • Correct iterate $v^{l-1}$ by $v^{l-1} = v^{l-1} + P^{l-1}_l e^l$.
        • Relax on $F^{l-1}(u^{l-1}) = f^{l-1}$ (with initial iterate $v^{l-1}$), $\nu_2$ times to
          get $v^{l-1}_{\nu_2}$.
        • Compute the correction $e^{l-1} = v^{l-1}_{\nu_2} - v^{l-1}$.
    • .
      .
      .
• Correct iterate $v^0$ by $v^0 = v^0 + P^0_1 e^1$.
• Relax on $F^0(u^0) = f^0$ (with initial iterate $v^0$), $\nu_2$ times to get $v^0_{\nu_2}$.
```

A specific FAS construction of the f^j, $1 \leq j \leq l$ is given in the above steps of the FAS algorithm. The definition of the multigrid components is open to the user according to the nonlinear problem at hand. For clarity in the following we assume a particular family of nonlinear problems often arising in practice. Next, we specify a particular AMGe choice of the multigrid components that lead to our FAS-AMGe method, and we present results of numerical tests illustrating the performance of the resulting FAS-AMGe algorithm on this family of nonlinear equations.

3. A family of model nonlinear problems.

3. A family of model nonlinear problems. To show a particular choice of multigrid components, we introduce a family of nonlinear problems that appear frequently in applications.

Suppose the nonlinear equation

$$L(u) = -\nabla \cdot a(u)\nabla u + g(u)u = f(x) \tag{3.1}$$

has been posed on a 2D or 3D domain with $u = 0$ on $\partial\Omega$. The coefficient functions $a(u)$ and $g(u)$ are nonlinear and $a(u),\ g(u) \geq 0$. In addition, $f(x)$ is given. Assume that $u_0 \approx u$. Then the following second order self-adjoint elliptic (linear) problem $-\nabla \cdot a(u_0)\nabla u + g(u_0)u = f(x)$ approximates the nonlinear problem (3.1). This class of linear elliptic problems is well handled by multigrid or various algebraic multigrid (AMG). Our hope is to see similar behavior of FAS applied to this model class of nonlinear PDEs.

We assume that a and g can be analytically evaluated for any value of their arguments. Otherwise, interpolated values should be used. Our method does not take special advantage of the homogeneous Dirichlet boundary conditions; those are assumed for simplicity. The above model class of nonlinear diffusion problems arises in a number of applications, including flow-through porous media, radiation transport, phase transition and biochemistry, and dynamics of biological groups. One could treat more general nonlinearities such as $a(u, \nabla u)$, but this would require a more sophisticated choice of the coarse nonlinear problems to be constructed in what follows, which is generally possible provided that additional fine grid information is assumed. The more general case, though, is not considered in this paper.

For any two admissible functions u and v of some Hilbert space V, the variational formulation of this problem is given by: Find $u \in V$ such that

$$(L(u), v) = (f, v), \text{ for all } v \in V, \tag{3.2}$$

where

$$(L(u), v) \equiv \int_\Omega a(u) \nabla u \cdot \nabla v \; dx + \int_\Omega g(u) uv \; dx,$$

and

$$(f, v) \equiv \int_\Omega fv \; dx.$$

Existence and uniqueness of a solution for nonlinear elliptic PDEs are found in [7], [4], and also [15]. Typically, strong monotonicity of the elliptic operator is assumed, which imposes some additional restrictions on the nonlinear coefficients a and g. For our purposes it is sufficient to assume that a unique solution u exists and has some smoothness. Then, a finite element discretization will have similar behavior, and the discrete solution will approach the exact solution when the mesh size tends to zero. Details about finite element error estimates for nonlinear elliptic PDEs are found, e.g., in [20].

We now specify a finite element discretization to (3.2). Let τ be a partition of Ω into quasi-uniform triangular elements T. Let V_h be the finite element space of piecewise linear functions associated with τ and vanishing on $\partial\Omega$. Then the Galerkin method for approximating the solution of (3.2) consists of finding $u^h \in V^h$ such that

$$(L(u^h), v^h) = (f, v^h),$$

where the left- and right-hand side of this variational equation are defined by

$$(L(u^h), v^h) \equiv \sum_{T \in \tau} \int_T a(u^h) \nabla u^h \cdot \nabla v^h \; dx \; + \sum_{T \in \tau} \int_T g(u^h) u^h v^h \; dx, \tag{3.3}$$

and

$$(f, v^h) \equiv \sum_{T \in \tau} \int_T f v^h \; dx. \tag{3.4}$$

If in any triangle $T \in \tau$ with vertices i, j, k, we approximate the nonlinear coefficients $a(u)$ and $g(u)$ inside the integrals by their averaged values $a_T(u) \equiv a(\bar{u}_T)$ and $g_T(u) \equiv g(\bar{u}_T)$ (the lower index $_T$ indicates averaging over the triangle T) where

$$\bar{u}_T = \frac{1}{3}(u(i) + u(j) + u(k)),$$

then (3.3) can be approximated by

$$(L_h(u^h), v^h) \equiv \sum_{T\in\tau} a_T(u^h) \int_T \nabla u^h \cdot \nabla v^h \; dx + \sum_{T\in\tau} g_T(u^h) \int_T u^h v^h \; dx. \tag{3.5}$$

The latter expression can be seen as a form obtained from (3.3) where the exact integrals are replaced by a simple quadrature rule.

If we define $\mathbf{u} = (u_1, \cdots, u_N)$, and $\mathbf{v} = (v_1, \cdots, v_N)$, where N is the total number of degrees of freedom (dofs) or vertices, we can introduce the Laplace element matrix A_T and the L_2-mass matrices G_T. Then, (3.5) can be rewritten in matrix vector form as

$$\mathbf{v}^t L_h(\mathbf{u}) = \sum_{T\in\tau} \mathbf{v}_T^t \left(a_T(\mathbf{u})A_T + g_T(\mathbf{u})G_T\right) \mathbf{u}_T, \tag{3.6}$$

where for any vector $\mathbf{q}$, $\mathbf{q}_T$ indicates the restriction of $\mathbf{q}$ to the element T and $\mathbf{q}^t$ stands for the transpose of $\mathbf{q}$. That is, if one assembles the matrix $M(\mathbf{u}_0)$ for any given $\mathbf{u}_0$ from the element matrices

$$a_T(\mathbf{u}_0)\; A_T + g_T(\mathbf{u}_0)\; G_T, \tag{3.7}$$

then the actions of the nonlinear operator defined in (3.6) are computed via the matrix-vector product $F(\mathbf{u}_0) = M(\mathbf{u}_0)\mathbf{u}_0$. Alternatively, the nonlinear operator actions can be computed element-wise. Assuming that the coefficients $a = a(u)$ and $g = g(u)$ are differentiable, that is, the derivatives $a^{'}(u)$ and $g^{'}(u)$ exist, it is straightforward to show that the Fréchet derivative $J(\mathbf{u}) \equiv F^{'}(\mathbf{u})$ for $F(\mathbf{u})$ is given variationally as follows. Note that for any direction $\mathbf{h}$, using Taylor expansion $a_T(\mathbf{u}+\mathbf{h}) = a_T(\mathbf{u}) + a_T^{'}(\mathbf{u})\overline{\mathbf{h}}_T + \dots$ and similarly for g_T, one gets for any $\mathbf{v}$ and any direction $\mathbf{h}$, the representation

$$\begin{aligned} \mathbf{v}^T J(\mathbf{u})\; \mathbf{h} &= \sum_T \left(a_T(\mathbf{u})\; \mathbf{v}_T^t A_T \mathbf{h}_T + g_T(\mathbf{u})\; \mathbf{v}_T^t G_T \mathbf{h}_T\right) \\ &+ \sum_T \left(a_T^{'}(\mathbf{u})\overline{\mathbf{h}}_T\; \mathbf{v}_T^t A_T \mathbf{u}_T + g_T^{'}(\mathbf{u})\overline{\mathbf{h}}_T\; \mathbf{v}_T^t G_T \mathbf{u}_T\right). \end{aligned} \tag{3.8}$$

We recall that $\overline{\mathbf{h}}_T$ stands for the averaged value of $\mathbf{h}$ over a given element T. It is clear that the above expression (for a fixed $\mathbf{u}$) leads to a nonsymmetric matrix (linear operator) $J(\mathbf{u})$ (the expression (3.8) is not symmetric with respect to $\mathbf{v}$ and $\mathbf{h}$).

4. The multigrid components of the FAS-AMGe method.

In this section we specify the multigrid components for the FAS method described in Section 2 in the case when (linear) element agglomeration AMGe makes sense. This is the case of finite element second-order elliptic equations discretized on general unstructured meshes. In the following we summarize the agglomeration-based AMGe coarsening, paying particular attention to the selection of coarse nodes (degrees of freedom), the construction of the AMGe-based intergrid transfer operators, and the construction of coarse non-linear problems. Finally, a general smoothing iteration is outlined.

4.1. The coarsening strategy.

The coarsening strategy is the same as in the linear agglomeration AMGe method (for details see [10] or [19]). We briefly outline the principal steps. Having the original grid $\Omega^0 = \Omega$, which by definition is a set of fine grid elements, the AMGe method assumes that the elements are represented as certain relation tables (lists). A minimal assumption is that one has access to the fine grid relation table "element_node", which specifies for every element a list of nodes (degrees of freedom) that belong to that element. A proper storage of this relation table is a boolean sparse matrix (rows being the element number and columns being the node numbers). For the coarsening to proceed, one needs some additional topological information such as the relation tables "element_face" and "face_face". The "element_face" table specifies for each element a list of faces (which are obtained as a maximal intersection set of the collection of elements). The "face_face" table specifies the connectivity of the faces. All this is assumed only on the fine grid (available to the user).

The goal is to construct similar relation tables on coarse levels by recursion. A main step of the coarsening procedure is the agglomeration of fine grid elements. Based on the assumed fine grid topological relations, one forms agglomerated elements (unions of connected fine grid elements) such that the resulting set of agglomerated elements provides a partition of the original set of fine grid elements. Having the agglomerated elements (which serve as coarse elements) constructed, one then appropriately defines their faces. Finally, the connectivity relation "face_face" of the new faces is created. Detailed constructions are found in [19]. Since the original information is created on a coarse level, a recursion is feasible, and a sequence of coarse elements can be created. The algorithm at this stage does not make use of nodal information and in particular of the fine grid relation table "element_node". The next step is to actually form the list of coarse elements in terms of coarse nodes. To form this list one has to identify coarse nodes at every level. The procedure to construct the first coarse grid Ω^1 from Ω^0 focuses on finding a subset Ω^1 of nodes of the fine grid. The simplest choice (which we use in this paper) is to label a node as coarse if it belongs to more than two agglomerated elements (hence if it belongs to at least two coarse faces). Then one can form the lists (coarse elements) _ (coarse nodes) (the coarse nodes which are contained in the fine elements forming an agglomerated element). In other words, the coarse nodes are the vertices of the agglomerated elements. This is the minimal coarse grid one can use. One may label more nodes as coarse giving richer coarse grids, or one can even choose coarse "nodes" (or rather degrees of freedom) that do not have a nodal meaning (such as certain eigenvectors of small matrices). But this more sophisticated choice is not considered in this paper.

For our purpose, it is enough that we have a recursive procedure which provides nested sets of coarse grids (in terms of nodes) $\{\Omega^k\}_{k=0}^{l}$ such that $\Omega^{k+1} \subset \Omega^k$.

4.2. The grid transfer operators. The next step of the FAS-AMGe algorithm is to construct the grid transfer operators between two consecutive grids $\Omega^f = \Omega^k$, a fine grid, and $\Omega^c = \Omega^{k+1}$, a coarse grid, of the hierarchy of grids already constructed. Restriction operators R from Ω^f to Ω^c are the transpose of interpolation operators P from Ω^c to Ω^f. The injection operator for any fine grid vector $\mathbf{v}$ is $\tilde{R}\mathbf{v} = \mathbf{v}|_{\Omega^c}$ (this definition makes sense if $\Omega^c \subset \Omega^f$).

In AMGe the place where the element matrices are used is when the interpolation operators P are constructed. The procedure (originating in [2]) requires creation for each fine node $d^f \in \Omega^f$ an element neighborhood $\Omega(d^f)$ (consisting of fine grid elements that share d^f) and a set N^c of coarse nodes $N^c = \{d^c\}$ which belong to the elements from $\Omega(d^f)$. The numerical solution at each fine node d^f will be interpolated (or 'averaged') from its associated neighborhood of coarse nodes N^c. The interpolation is the identity operator for every fine node which is labeled as coarse. For the remaining fine nodes one uses a energy minimization principle leading to solving a small system of equations, which determines the respective row of the interpolation matrix corresponding to the fine node d^f. The procedure is summarized as follows. From the given fine grid element matrices one assembles the local matrix $A_{\Omega(d^f)}$. Then one partitions this matrix into a two-by-two block form; the second block corresponds to the set of nodes N^c, i.e.,

$$A_{\Omega(d^f)} = \left[\begin{array}{cc} A_{ff} & A_{fc} \\ A_{cf} & A_{cc} \end{array} \right] \begin{array}{ll} \} & \Omega(d^f) \setminus N^c \\ \} & N^c \end{array} .$$

The row of $-\left(A_{ff}\right)^{-1} A_{fc}$ corresponding to $d^f \in \Omega(d^f) \setminus N^c$ gives the interpolation weights $w_{d^f,\, d^c}$, $d^c \in N^c$, which form the non-zero entries of the d^fth row of the interpolation matrix $P = P^k_{k+1}$.

The agglomeration AMGe selects the sets N^c such that the resulting P has the property to create coarse element matrices without conflicts (note that shared dofs by two agglomerated elements should be uniquely interpolated only by coarse dofs that are shared by the same agglomerated elements). Details are found in [10]. Having constructed coarse element matrices one can proceed by recursion. Other interpolation procedures are possible (without access to the element matrices, see [9]), but in our model case of nonlinear elliptic problems the element matrices are naturally available. More specifically, in the interpolation procedure one can use the matrices from (3.7) for a given approximation $\mathbf{u}_0$.

4.3. Nonlinearities and coarse equations. The nonlinear operator F^0 in the fine grid is defined by the right-hand side of (3.6). The coarse nonlinear operator F^j is defined by (3.6) where the sum runs over the macroelements given at the grid Ω^j.

This choice, F^j, of coarse nonlinearity is simple. It just requires the coarse stiffness matrices (for the Laplacian) and the coarse mass matrices (weighted by averaged nonlinearities) for each macroelement. As already mentioned, the agglomeration-based AMGe has the ability to provide coarse element matrices.

A second choice of computing the nonlinear operator is the nonlinear Galerkin operator F^j_G given by

$$F^j_G(u^j) = {Q^j}^t F^0(Q^j u^j), \tag{4.1}$$

where $Q^j = P^0_1 P^1_2 \cdots P^{j-1}_j$. Galerkin coarse nonlinearities (4.1) are more expensive to compute than nonlinear operators F^j because they are implicit and (as already mentioned in the introduction) their actions are computed via the fine grid nonlinear operator. This will not lead to a method of optimal complexity per iteration. However, according to variational theory, Galerkin nonlinearities in the coarse equations are expected to generate a better correction for the actual iterate in the finest grid than non-Galerkin ones. Moreover, variational theory could be used to study convergence of some nonlinear multigrid algorithms when using Galerkin coarse nonlinear operators.

4.4. The nonlinear relaxation methods. By accumulating elements in (3.6) it is not difficult to see that nonlinearity $F(u)$ has the form

$$F(u) = M(u)u. \tag{4.2}$$

This special form of the nonlinearity suggests that besides a Newton solver to iterate according to

$$\mathcal{F}^{j^{'}}(u^j_m)(u^j_{m+1} - u^j_m) = -\mathcal{F}^j(u^j_m), \tag{4.3}$$

where $\mathcal{F}^j(u) = f^j - F^j(u)$ and $\mathcal{F}^{j'}$ is its Jacobian, a fixed-point (Picard) iterative method could be used instead for relaxing the nonlinear problems $F^j(u^j) = f^j$, $0 \leq j \leq l$ by iterating on

$$u^j_{m+1} = \left(M^j(u^j_m)\right)^{-1} f^j,\ m \geq 0. \tag{4.4}$$

In practice all inverses involved (the Jacobian inverse and the Picard matrix inverse) are computed inexactly, for example, by a few (linear) multigrid cycles, generally as preconditioners in GMRES or CG.

5. Numerical results.

Our main goal in this section is to verify that the speed of convergence of the FAS-AMGe algorithm is independent of the mesh size of the nonlinear problem. A second goal is to make a fair comparison between the FAS-AMGe algorithm using some relaxation method and the respective relaxation method used as a nonlinear iterative method on the finest grid.

Two smoothers have been chosen to relax the nonlinear equations: the Newton nonlinear solver and the Picard fixed-point iterative method. The Picard nonlinear smoother iterates on (4.4) and uses a V-cycle with minimal overlapping multiplicative Schwarz smoother to solve linear problems [3]. On the other hand, the Newton nonlinear method iterates on (4.3). The linear system in (4.3) is solved using a GMRES algorithm preconditioned by the same V-cycle as in the Picard smoother. The V-cycle is an agglomeration AMGe constructed based on (the symmetric positive definite matrix) $M(u)$ in (4.2). This Newton implementation does not include any kind of line search strategy (see [6]).

In the numerical results given below, we have constructed three variants of this FAS-AMGe algorithm. The first, FAS-Picard, uses the Picard smoother in all grids. The second, FAS-Newton, uses the Newton smoother in all grids. Our numerical experiments have suggested the implementation of a third variant of the FAS-AMGe algorithm, which we call FAS-Hybrid. This method uses the Picard smoother in the finest grid and the Newton smoother in all the others.

We have used a set of four grids obtained by successive factor-of-two mesh refinement. These grids have been constructed from a Cartesian mesh on a unit square. The smallest of these grids is shown in the top left panel in Figure 5.1. The other five panels show the coarse macroelements obtained by successive application of the agglomeration of elements algorithm.

Interpolation operators are constructed once per V-cycle. We have selected a collection of nonlinear scalar functions, $a(u)$ and $g(u)$ in (3.3), which are given in Table 5.1. Homogeneous Dirichlet boundary conditions are imposed on the rectangular domain. The exact solution to all these problems is the function $u(x, y) = x(1 - x)y(1 - y)$ unless otherwise stated. The initial iterate on the interior of the rectangle is either a constant or a multiple of the exact solution as specified.

5.1. Scalability of the FAS-AMGe algorithm.

In this subsection we present two studies that show scalability of the FAS-AMGe algorithm for nonlinear problems.

The first study is shown in Tables 5.2 and 5.3. These tables show results for different initial guesses. The linear solver tolerances of this study have been set to 10^{-2}, and the maximum number of linear iterations set to 1000 (to allow convergence). Each of the four columns has two numbers. In the first column, the number outside parenthesis indicates the number of original dofs and the number inside parenthesis shows the number of coarse grids corresponding to each fine mesh. In the other columns, the number outside parenthesis indicates the number of FAS-Picard $V(1, 1)$ cycles (Table

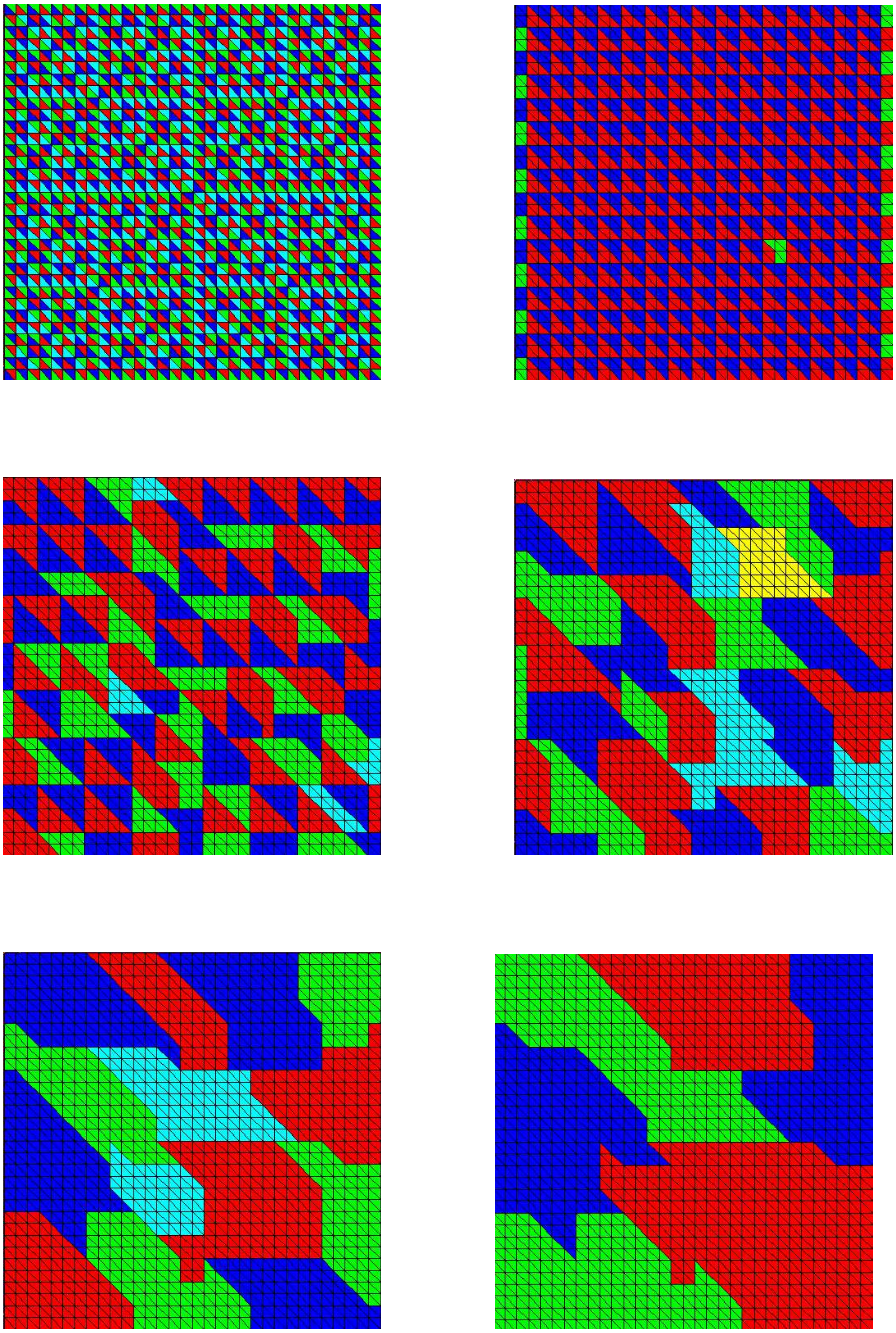

Fig. 5.1. *The left top panel shows the smallest finite element triangularization of the square used for the convergence speed scalability studies. The vertices of this triangles are the fine dofs. The other panels show part of the agglomeration of elements procedure used in the coarsening strategy.*

Table 5.1

Set of nonlinearities in numerical simulations.

Nonlinearity case	$a(u)$	$g(u)$
1	$\frac{1}{\sqrt{u^2+10^{-3}}}$	1
2	$\sqrt{u^2+u+1}$	0
3	u^2+10^{-3}	u
4	$\frac{1}{\sqrt{u^2+10^{-3}}}$	u^2+10^{-3}

Table 5.2

The number of FAS-Picard $V(1,1)$ cycles and linear iterations on the finest grid for different initial constant guesses and nonlinearity case 1*. Linear tolerances have been set to* 10^{-2} *and the maximum number of linear iterations is* 1000*.*

Degrees of freedom (levels)	Number of V-cycles (Number of linear iterations)		
	Initial constant guess		
	10	100	1000
1,089 (7)	14 (40)	14 (39)	14 (38)
4,225 (8)	8 (22)	9 (24)	9 (24)
16,641 (9)	5 (13)	6 (13)	8 (17)
66,049 (11)	7 (20)	9 (21)	9 (21)

Table 5.3

The number of Picard iterations and linear iterations on the finest grid for different initial constant guesses and nonlinearity case 1*. Linear tolerances have been set to* 10^{-2} *and the maximum number of linear iterations is* 1000*.*

Degrees of freedom (levels)	Number of nonlinear iterations (Number of linear iterations)		
	Initial constant guess		
	10	100	1000
1,089 (7)	13 (106)	29 (110)	31 (112)
4,225 (8)	18 (126)	19 (86)	21 (131)
16,641 (9)	16 (154)	17 (201)	18 (199)
66,049 (11)	15 (724)	16 (718)	16 (80)

Table 5.4

The number of FAS-Newton $V(1,1)$ cycles and linear iterations on the finest grid for different initial guesses and nonlinearity case 2*. These guesses are multiples of the exact solution. Linear tolerances have been set to* 10^{-1} *and the maximum number of linear iterations is* 1000*.*

Degrees of freedom (levels)	Number of V-cycles (Number of linear iterations)		
	Initial multiple guess		
	13	26	39
1,089 (7)	2 (1)	4 (6)	4 (4)
4,225 (8)	2 (1)	2 (2)	3 (1)
16,641 (9)	2 (2)	2 (2)	3 (4)
66,049 (11)	2 (3)	2 (2)	3 (4)

Table 5.5

The number of Newton iterations and linear iterations on the finest grid for different initial guesses and nonlinearity case 2 *(multiples of the exact solution). Linear tolerances have been set to* 10^{-1} *and the maximum number of linear iterations is* 1000*.*

Degrees of freedom (levels)	Number of nonlinear iterations (Number of linear iterations)		
	Initial multiple guess		
	13	26	39
1,089 (7)	8 (7)	10 (4)	11 (8)
4,225 (8)	8 (13)	10 (13)	11 (15)
16,641 (9)	8 (22)	9 (14)	11 (25)
66,049 (11)	7 (31)	9 (30)	10 (34)

5.2) or the number of Picard nonlinear iterations (Table 5.3). Or shortly, this is the number of outer nonlinear iterations. The number inside parenthesis shows the total number of linear (or inner) iterations in the finest grid for both the FAS-Picard (Table 5.2) and the Picard fixed-point iterative method (Table 5.3). The nonlinear tolerance is 10^{-6}, and the initial guess is constant on interior nodes. Nonlinearity case 1 (see Table 5.1) is used.

It is clear that the convergence rate in terms of not only $\mathrm{V}(1,1)$-cycles but also in terms of linear iterations for the FAS-Picard algorithm is fairly insensitive with respect to the fine-grid mesh size. Furthermore, both numbers of V-cycles and linear iterations on a given fine grid required by FAS-Picard are also independent of the initial constant guess. The same is not always true for Picard iteration. In addition, the total number of linear iterations on the finest grid performed by FAS-Picard is much less than the number of linear iterations performed by Picard in this particular example.

A similar study that compares FAS-Newton and Newton with analogous conclusions is shown in Tables 5.4 and 5.5. The parameters used are the same although the tolerances for the linear iterations have been set to 10^{-1} and the initial interior node guesses have been set as a multiple of the exact solution. The number of Schwarz $V(1,1)$-cycles for the preconditioner used in GMRES is 1. Nonlinearity case 2 (see Table 5.1) is used.

Note that for this nonlinear problem the total number of linear iterations on the finest grid for FAS-Newton is small when compared to the number of $V(1,1)$ cycles. Therefore, using methods based on the hierarchy of meshes leads to better performance than just a single-level method (Picard or Newton) in which a linear multigrid preconditioner is incorporated to inexactly solve the corresponding linear problem.

5.2. A comparison between nonlinear solvers and FAS-AMGe. We next study the performance of each method with respect to a spectrum of constant interior initial guesses for a couple of nonlinear problems. In these studies all tolerances (linear and nonlinear) have been set to 10^{-6}. A maximum number of 3 linear iterations per nonlinear iteration has been imposed for Picard and Newton methods. Only 3 Schwarz V$(1,1)$ cycles for FAS-Picard and FAS-Hybrid (finest grid) per smoother step were applied. Similarly, only 3 GMRES iterations (without restarting) for FAS-Newton and FAS-Hybrid per smoother step were applied.

Tables 5.6 and 5.7 show the performance of Picard and Inexact Newton nonlinear solvers and the three versions of the FAS-AMGe algorithm for nonlinearities 3 and 4 (see Table 5.1). These two nonlinearities exhibit different behaviors for the nonlinear diffusion $a(u)$. While in nonlinearity case 3 the diffusion coefficient is growing quadratically with respect to the numerical solution, in nonlinearity 4 it is decreasing monotonically. Both nonlinear problems have been computed on a structured mesh with $66,049$ dofs and 11 levels. Constant initial guesses are displayed in the first column of these tables. In the second and third columns, two numbers are shown. The number outside parenthesis indicates the number of nonlinear iterations required to converge for Picard and Newton solvers, respectively. The number inside parenthesis shows the total number of linear iterations. Columns four to six display the number of FAS-AMGe $V(1,1)$ cycles for FAS-Picard, FAS-Newton, and FAS-Hybrid, respectively, outside the parentheses, and inside parenthesis the total number of linear iterations on the finest grid is shown.

Table 5.6 shows that the number of iterations of Newton and the number of FAS-Newton V$(1,1)$ cycles depends on the initial guess. FAS-Picard and FAS-Hybrid are robust with respect to the constant initial guess requiring roughly the same number of V$(1,1)$ cycles. Picard is also robust with respect to the constant initial guess, but it requires much more computational work than FAS-Picard and FAS-Hybrid.

Table 5.7 shows that the convergence region for Newton's method and FAS-Newton is much smaller for this problem than for the previous example. The number of Newton iterations and FAS-Newton V$(1,1)$ cycles increases exponentially as soon as the constant initial guess reaches the boundary of the basin of convergence of these methods. The other three methods again show robustness with respect to the initial guess even though Picard requires more computational work than FAS-Picard and FAS-Hybrid.

The results shown in Tables 5.6 and 5.7 bear out theory that indicates that Picard should have a wider basin of convergence than Newton. This suggested why the FAS-Hybrid method was developed the way it was.

6. Conclusions. We have developed a nonlinear algebraic algorithm that extends an AMGe linear method to nonlinear elliptic problems. The current work is

Table 5.6

The number of Picard and Newton nonlinear iterations and FAS-AMGe $V(1,1)$ cycles for nonlinearity case 3. *FAS-AMGe algorithms are used in a mesh with* $66,049$ *dofs and* 11 *levels. Linear tolerances have been set to* 10^{-6} *and the maximum number of linear iterations per nonlinear iteration is* 3.

	Nonlinear iterations		Number of V-cycles		
Initial	(linear iterations)		(Number of finest grid linear iterations)		
guess	Picard	Newton	FAS-Picard	FAS-Newton	FAS-Hybrid
1.0	15 (45)	10 (30)	1 (6)	5 (30)	2 (12)
5.0	15 (45)	14 (42)	1 (6)	7 (42)	2 (12)
10.0	15 (45)	16 (48)	1 (6)	4 (24)	3 (18)
20.0	13 (39)	18 (54)	1 (6)	17 (102)	3 (18)
50.0	9 (27)	20 (60)	1 (6)	10 (60)	2 (12)
100.0	10 (30)	22 (66)	2 (12)	12 (72)	2 (12)
1000.0	16 (48)	27 (81)	2 (12)	15 (90)	2 (12)
10000.0	8 (24)	33 (99)	4 (24)	diverges	3 (18)
100000.0	17 (51)	39 (117)	diverges	22 (132)	2 (12)

Table 5.7

The number of Picard and Newton nonlinear iterations and FAS-AMGe $V(1,1)$ cycles for nonlinearity case 4. *FAS-AMGe algorithms are used in a mesh with* $66,049$ *dofs and* 11 *levels. Linear tolerances have been set to* 10^{-6} *and the maximum number of linear iterations per nonlinear iteration is* 3.

	Nonlinear iterations		Number of V-cycles		
Initial	(linear iterations)		(Number of finest grid linear iterations)		
guess	Picard	Newton	FAS-Picard	FAS-Newton	FAS-Hybrid
0.01	6 (18)	4 (12)	2 (12)	2 (12)	2 (12)
0.03	7 (21)	4 (12)	2 (12)	2 (12)	2 (12)
0.05	8 (24)	4 (12)	2 (12)	2 (12)	2 (12)
0.06	8 (24)	5 (15)	2 (12)	2 (12)	2 (12)
0.07	8 (24)	7 (21)	2 (12)	$>$ 25 (150)	2 (12)
0.08	8 (24)	$>$ 50 (150)	2 (12)	$>$ 25 (150)	2 (12)
1.0	10 (30)	$>$ 50 (150)	2 (12)	$>$ 25 (150)	4 (24)
15.0	9 (27)	$>$ 50 (150)	2 (12)	$>$ 25 (150)	4 (24)
50.0	7 (21)	$>$ 50 (150)	2 (12)	$>$ 25 (150)	2 (12)
100.0	8 (24)	$>$ 50 (150)	2 (12)	$>$ 25 (150)	2 (12)

a much wider study (involving in particular Newton method) than the preliminary results reported in [11]. The proposed nonlinear algebraic algorithm can be seen as an FAS algorithm defined on an unstructured mesh, provided the usual multigrid components (grid hierarchy, cycle form, coarsening algorithm, intergrid transfer operators, coarse nonlinearities and relaxation methods) have been identified. This algorithm has been implemented to solve discretizations of nonlinear operators coming from variational formulations of reaction-diffusion equations with nonlinear diffusion and reaction terms.

Numerical simulations show scalability of the computational work of this algorithm. When used to compute a numerical solution of some nonlinear elliptic PDEs, the FAS-AMGe approach requires much less computational work than nonlinear solvers applied to the fine grid.

An additional study comparing several smoothers in FAS-AMGe is performed. A hybrid method that uses a global nonlinear iterative relaxation method on the finest grid with a local nonlinear iterative relaxation method on coarse grids is introduced, and initial studies show that this hybrid method can perform better than a method with either smoother on all grids.

REFERENCES

[1] A. Brandt, *Multigrid techniques: 1984 guide with applications to fluid dynamics*, Weizmann Institute of Science Technical Report, 1984.

[2] M. Brezina, A. J. Cleary, R. D. Falgout, V. E. Henson, J. E. Jones, T. A. Manteuffel, S. F. McCormick, & J. W. Ruge, *Algebraic multigrid based on element interpolation (AMGe)*, SIAM J. Scientific Computing, 22(2000), pp. 1570-1592.

[3] X-C. Cai, & O. B. Widlund, *Multiplicative Schwarz algorithms for some nonsymmetric and indefinite algorithms*, SIAM J. Numer. Anal., 30 (1993), pp. 936-952.

[4] P.G. Ciarlet, *The Finite Element Method for Elliptic problems*, North Holland, Amsterdam, 1978.

[5] T. Chartier, R. Falgout, V. E. Henson, J. Jones, T. Manteuffel, S. McCormick, J. Ruge, and P. S. Vassilevski, "*Spectral AMGe (ρAMGe)*", 2001 (submitted).

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

[7] D. Gilbarg and N. S. Trudinger, *Elliptic Partial Differential Equations of Second Order*, Springer-Verlag, Berlin Heidelberg, 2001.

[8] W. Hackbusch, *Multi-Grid methods and Applications*, Springer, Berlin, 1985.

[9] V. E. Henson, & P. S. Vassilevski, *Element-free AMGe: general algorithms for computing interpolation weights*, SIAM J. Scientific Computing, 23(2001), pp. 629-650.

[10] J. E. Jones, & P. S. Vassilevski, *AMGe based on element agglomeration*, SIAM J. Scientific Computing, 23(2001), pp. 109-133.

[11] J. E. Jones, P. S. Vassilevski, and C. S. Woodward, *Computational Issues in the Application of Nonlinear Multigrid to Nonlinear Diffusion Problems*, talk presented at Tenth Copper Mountain Conference on Multigrid Methods, April 1–6, 2001.

[12] D. J. Mavriplis, *Multigrid techniques for unstructured meshes*, ICASE Report No. 95-27, NASA Langley research center, 1995. Also available as Lecture notes for the 26th CFD Lecture Series of von Karman Institute; AGARD Publication.

[13] D. J. Mavriplis, *An assessment of linear versus non-linear multigrid methods for unstructured meshes*, ICASE Report No. 2001-12 also available as J. of Comp. Phys., vol 175, pp. 302-325, Jan. 2002.

[14] D. J. Mavriplis, *Multigrid approaches to non-linear diffusion problems on unstructured meshes*, Numer. Linear Algebra Appl., 8(2001), pp. 499-512.

[15] V. V. Shaidurov, *Multigrid Methods for Finite Elements*, Kluwer Academic Publishers, Dordrecht-Boston-London, 1995.

[16] L. Stals, *The solution of radiation transport equations with adaptive finite elements*, ICASE Report No. 2001-29, NASA Langley research center, 2001.

[17] U. Trottenberg, C. Oosterlee, & A. Schüller, *Multigrid*, Academic Press, San Francisco, 2001.

[18] J.W. Ruge, & K. Stuben, *Algebraic Multigrid (AMG), Multigrid Methods*, Frontiers in Applied Mathematics (ed. S.F. McCormick), pp 73-130, SIAM, Philadelphia, 1987.

[19] P. S. Vassilevski, *Sparse matrix element topology with application to AMG(e) and preconditioning*, Numerical Linear Algebra with Applications (in press), 2002.

[20] J. Xu, *Two-grid discretization techniques for linear and nonlinear problems*, SIAM Journal Numerical Analysis 33(1996), pp. 1759-1777.

University of California
Lawrence Livermore National Laboratory
Technical Information Department
Livermore, CA 94551