Article:

Jones, A.C. and Jimack, P.K. (2005) An adaptive multigrid tool for elliptic and parabolic systems. International Journal for Numerical Methods in Fluids, 47 (10-11). pp. 1123-1128.

ISSN 0271-2091

https://doi.org/10.1002/fld.845

INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2000; **00**:1–6 *Prepared using* fldauth.cls *[Version: 2002/09/18 v1.01]*

An adaptive multigrid tool for elliptic and parabolic systems

A. C. Jones[1,*] and P. K. Jimack[1]

[1] *CPDE Unit, School of Computing, University of Leeds, Leeds, LS2 9JT, U.K.*

SUMMARY

This paper describes a multigrid tool for use with adaptively refined meshes in two dimensions. A standard Full Approximation Storage (FAS) multigrid scheme is modified to maintain efficient performance on non-uniform grids arising as a results of local mesh refinement. This is achieved by allowing hanging nodes to exist but ensuring that at each stage of the FAS algorithm the solution is projected into the continuous space in which the hanging node values are interpolants of their coarse grid parent values. A nonlinear, elliptic, finite element example is provided to demonstrate the efficiency of the proposed algorithm. However, the method is also suitable for linear and nonlinear parabolic problems. Copyright © 2000 John Wiley & Sons, Ltd.

KEY WORDS: mesh adaptivity, FAS multigrid, finite element method

1. INTRODUCTION

Many CFD problems feature regions of high activity which are small relative to the computational domain. In such cases it is natural to use some form of adaptivity to ensure that the mesh resolution is sufficiently fine in these regions to obtain an accurate solution, whilst being sufficiently coarse elsewhere to maximise efficiency. Examples of this type are numerous but include phase field simulations of rapid solidification, as described in [1, 2] for example. Whilst obtaining a suitably refined mesh is undoubtedly a necessary condition for achieving an efficient solution it is by no means sufficient. When the adaptive algorithm is based upon a hierarchy of nested meshes the multigrid approach naturally suggests itself as a candidate for an efficient solver. In [3], for example, a nonlinear time-dependent problem, modelling the spreading of a thin drop over a rough surface, is solved very efficiently using an implicit, adaptive, time-stepping algorithm built around an FAS multigrid solver. In that work however the different grid levels are obtained through uniform global refinement, rather than local, adaptive refinement. In this paper we propose a simple framework that allows the efficiency of local hierarchical refinement (and coarsening) to be combined with multigrid to

*Correspondence to: Alison Jones, CPDE Unit, School of Computing, University of Leeds, Leeds, LS2 9JT, U.K.

allow the optimal solution of nonlinear elliptic problems and nonlinear parabolic problems with fully implicit time discretisation.

The 'disconnected' or 'hanging nodes' that are found in non-uniform meshes need special treatment in order to avoid discontinuities in the solution. This has been tackled in many ways including adding refinement to eliminate the hanging nodes [2], modifying the finite element basis [4, 5] or adding constraint equations for these nodes [6]. The nature of multigrid opens up other options for dealing with hanging nodes. The MLAT approach of Brandt [7] sets temporary artificial Dirichlet conditions at the boundaries of any locally refined areas and smoothing takes place on these uniform subdomains. The FAC approach of McCormick *et al* [8], however, uses special implicit finite-volume-element stensils at the refinement boundaries to resolve the fine and course grid residuals in this area. We propose to use a projection method, originally suggested in the context of linear problems in [6], to impose constraints on the value of the hanging nodes. We have adapted the basic FAS multigrid method so that smoothing can be performed over non-uniform grids at each level. This method allows the use of a straight forward continuous finite element discretisation.

2. ADAPTIVE MESH METHODS

The adaptive mesh algorithm that is used in this work is based around quadrilateral elements that are stored in a quadtree data structure, as in [2]. The use of a quadtree allows easy access to all levels within the grid using tree traversal functions, making it an appropriate choice for use with a multigrid solver. At any given time the current mesh is made up of those elements that correspond to leaf nodes in the quadtree. The refinement of an element involves the creation of four children for the corresponding leaf node in the quadtree. To coarsen, four elements with a shared parent, and which all correspond to leaf nodes of the tree, may be replaced by their parent. Numerous constraints are placed on these operations however to ensure that:

- There is never more than one level of refinement difference between any two neighbouring elements. It follows that there is only ever one hanging node per element edge.
- A specified number of "safety layers" are added to the refinement at each level. This ensures that there will always be at least this number of elements at each level between the interface with the next finer level and the interface with the next coarser level.

Any element-based refinement criteria may be used to control the adaptivity, such as local error estimates or simple error indicators based upon solution gradients, for example.

3. MULTIGRID METHOD

The multigrid method that we propose is based upon the FAS scheme and the use of non-uniform, locally refined grids as described in the previous section. Each level of the multigrid algorithm is based upon a composite grid covering the whole domain. Figure 1 shows a small section of a composite grid at three consecutive levels. Hence a suitable smoother, as well as suitable restriction and prolongation operators are required for such composite grids. These are described in this section in the context of a typical nonlinear model problem.

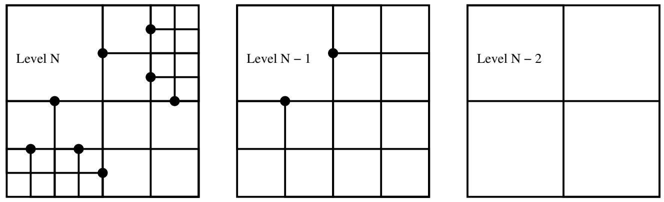

Figure 1. Three levels of a composite grid with hanging nodes marked with •

3.1. Model Problem

For the purposes of clarity, in this paper we focus on a single model problem only, however the ideas presented extend naturally to more general elliptic problems and to the solution of parabolic problems using implicit time stepping. The model problem considered here is:

$$-\nabla^2 u + \gamma u e^u = f \tag{1}$$

on the domain $\Omega = (-1,1) \times (-1,1)$ and subject to $u = 0$ on the Dirichlet boundary. For the purposes of testing the source term, f, is chosen to permit the solution: $u = 1 - tanh(k(x^2 + y^2 - \alpha^2))$. This solution has a relatively small area of high curvature in the region of $x^2 + y^2 = \alpha^2$, making it an appropriate problem to evaluate the multigrid solver on highly refined grids. For the results in this paper $\gamma = 1$, $k = 25$ and $\alpha = \frac{1}{2}$. A Galerkin finite element discretisation of this problem gives a nonlinear algebraic system of the form:

$$\hat{K}(\underline{u})\underline{u} = \underline{\hat{b}}, \tag{2}$$

where $\hat{K}(\underline{u})$ is the matrix obtained from the assembly of the element stiffness matrices, without regard for the hanging nodes, marked with • in Figure 1. These hanging nodes allow discontinuities in the finite element space along the edges where they lie. A general solution of this system therefore lies in this non-conforming space. The FAS multigrid method employed here seeks to solve a modified version of this problem by only ever working in specific subspaces of these non-conforming spaces at each refinement level.

3.2. Projected Jacobi Smoother

The multigrid smoother that is used deals naturally with hanging nodes and is based on the Preconditioned Projected Conjugate Gradient scheme due to Meyer, [6]. At each level we smooth the nonlinear system:

$$P^T \hat{K}(\underline{u}) P \underline{u} = P^T \underline{\hat{b}}, \tag{3}$$

where $\hat{K}(\underline{u})$ is as in Equation (2) and P is a matrix which projects the solution vector into the continuous subspace. The effect of P is to overwrite the value of each hanging node with the linear interpolant of its parent nodes. It is important to note that $P^T P \underline{u} \neq P P^T \underline{u} \neq \underline{u}$.

Applying the standard nonlinear damped Jacobi correction scheme to this new projected system will not produce a continuous solution. The correction δ_j, for node j, must be formed using vectors contained in $Image(P)$ in order to guarantee that they will preserve the continuity of the solution. An extra projection P is therefore added to the numerator and denominator of the standard formula to make δ_j a valid correction. This Projected Jacobi (PJ) update maintains the solution continuity despite the fact that the finite element assembly is performed in the non-conforming space:

$$\delta_j = \omega \frac{[P(P^T\hat{\underline{b}} - P^T\hat{K}\underline{u})]_j}{[P(\frac{\partial}{\partial u_j}P^T\hat{K}\underline{u})]_j}. \tag{4}$$

3.3. The Multigrid Algorithm

Algorithm 1 details the algorithm for the Projected FAS multigrid that is proposed. For simplicity a basic two grid version is presented which can easily be extended to a full V-cycle. The restriction operators, I_l^L and $\tilde{I}_l^L$, denote simple injection whilst full weighting is used for the prolongation operator, I_L^l. The restriction, I_l^L, is a point-wise operator for the solution while $\tilde{I}_l^L$ resolves the difference in the element support area for the residual between the two grid levels. These operators do not interfere with the projection method. The fine grid residual $\underline{r}^l \in Image(P^T)$ before it is restricted so all values corresponding to hanging nodes are zero. Therefore no information is lost at the boundaries between levels by using injection for the residual restriction. Injection is also appropriate for the solution restriction, on *line 8*, since it cannot create a non-continuous coarse grid solution from a continuous fine grid solution, i.e. if $\underline{v}^l \in Image(P)$ then $\underline{v}^L \in Image(P)$. Likewise the full weighting prolongation operator, used on *line 13*, will create a fine grid error whose fine level node values are linear interpolants of their parent node values. Hence, providing $\underline{e}^L \in Image(P)$ the result of the prolongation $\underline{e}^l \in Image(P)$. Although it is not highlighted explicitly in this notation the operators, P and P^T, are clearly different for each grid level.

The measure used as a stopping criterion must be an accurate representation of the system being solved. For example, the residual norm must be calculated using a residual vector within $Image(P)$:

$$\underline{r} = P[P^T\hat{\underline{b}} - P^T\hat{K}\underline{u}].$$

Algorithm 1 Projected FAS Multigrid - Two Grid Scheme

1: l = level number of current grid
2: $L = l - 1$ (level number of grid one coarser than l)
3: Choose initial fine grid solution $\underline{v}^l$ so that $P\underline{v}^l = \underline{v}^l$ i.e. $\underline{v}^l \in Image(P)$
4: Calculate the right-hand side $\hat{\underline{b}}^l$ and set $\underline{b}^l = P^T\hat{\underline{b}}^l$
5: Update $\underline{v}^l$ by smoothing ρ_1 times with the system $P^T\hat{K}^l(\underline{u}^l)\underline{u}^l = \underline{b}^l$ using PJ
6: Find residual $\underline{r}^l := \underline{b}^l - P^T\hat{K}^l(\underline{v}^l)\underline{v}^l$
7: Restrict residual $\underline{r}^L := \tilde{I}_l^L(\underline{r}^l)$
8: Restrict solution $\underline{v}^L := I_l^L(\underline{v}^l)$
9: Calculate coarse grid right-hand side $\underline{b}^L := \underline{r}^L + P^T\hat{K}^L(\underline{v}^L)\underline{v}^L$
10: Save the initial coarse grid solution $\underline{v}^L_{init} = \underline{v}^L$
11: Solve the system $P^T\hat{K}^L(\underline{v}^L)\underline{v}^L = \underline{b}^L$ exactly
12: Calculate the coarse grid correction $\underline{e}^L = \underline{v}^L - \underline{v}^L_{init}$
13: Prolongate the correction $\underline{e}^l := I_L^l(\underline{e}^L)$
14: Correct the fine grid solution $\underline{v}^l = \underline{v}^l + \underline{e}^l$
15: Update $\underline{v}^l$ by smoothing ρ_2 times with the system $P^T\hat{K}^l(\underline{u}^l)\underline{u}^l = \underline{b}^l$ using PJ

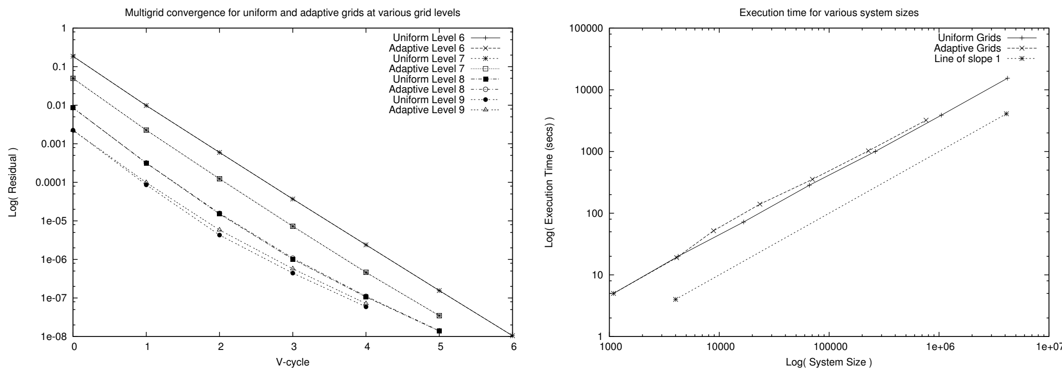

Figure 2. Graphs showing the mutligrid convergence at several grid levels and the execution time for various system sizes

4. RESULTS

This section presents typical results demonstrating the efficiency and accuracy of the proposed method. For the steady state test problem considered here a full multigrid scheme is used, with local mesh refinement undertaken after the solution is obtained at the previous level. A simple solution gradient monitor is used to trigger local refinement. No de-refinement is used here since the problem is steady state. At each level of the full multigrid the residual on this grid is used as the stopping criterion. The first graph of Figure 2 shows that the residual is reduced by an almost constant factor at each V-cycle on all uniform and non-uniform grids.

The second graph of Figure 2 shows that the execution time is linearly dependent on the system size. This relationship is different by a constant amount between the uniform and non-uniform cases since there is a slightly greater overhead in the adaptive case for refinement and hanging node operations. Nevertheless the time savings of adaptivity far outweighs this minor overhead.

Finally, note that for both uniformly and locally refined grids the error achieves the expected $O(h^2)$ behaviour. Table I illustrates this, as well as showing the difference in system size between the uniform and adaptive cases.

5. DISCUSSION

We have presented a new multigrid tool for adaptively refined meshes in two dimensions and given evidence of its satisfactory convergence and accuracy. The main advantage of the proposed method over alternatives such as [7] is its simplicity in dealing with arbitrary local refinement patterns. It should also be noted that, although smoothing is undertaken over the entire domain at each level, the cost per V-cycle is still optimal when refining into any finite sub-domain or any shock or boundary layer. In the case of a one-dimensional boundary layer, for example, the number of unknowns approximately doubles at each grid level and the efficiency achieved is therefore similar to that obtained when applying multigrid to a one-dimensonal problem using uniform refinement. Although results are only presented here for a single elliptic

Table I. Table showing the solution accuracy and system size for various grid levels, in the uniform and adaptive cases

Level	Uniform No. Nodes	L2-norm of Error	Adaptive No. Nodes	L2-norm of Error
4	289	2.0455651179e-01	289	2.0455651179e-01
5	1089	6.7681314537e-02	1089	6.7681314537e-02
6	4225	1.6865284766e-02	4093	1.6865280871e-02
7	16641	4.2772402758e-03	8873	4.2772482800e-03
8	66049	1.0738744046e-03	23425	1.0739181585e-03
9	263169	2.6877823655e-04	70485	2.6890580843e-04
10	1050625	6.7230013977e-05	227681	6.7730354577e-05

problem the method has been applied with equal success to a wide range of nonlinear elliptic and parabolic flow problems. Examples of the latter include fully implicit discretisations of the Porous Medium Equation [9], convection-dominated convection-diffusion problems (both single equations and systems), and nonlinear phase-field systems used to model rapid solidification [1, 2]. Note that these last two examples yield non-symmetric algebraic systems of equations for which the projected multigrid algorithm works equally well.

ACKNOWLEDGEMENT

We are very grateful to Dr. Mark Walkley for his valuable ideas and suggestions received during the course of this work.

REFERENCES

1. Wheeler AA, Murray BT, Schaefer RJ. Computation of dendrites using a phase field model. *Physica D* 1993; **66**:243–262.
2. Provatas N, Goldenfeld N, Dantzig J. Adaptive mesh refinement computation of solidification mircostructures using dynamic data structures. *Journal of Computational Physics* 1999; **148**:265–290.
3. Gaskell PH, Jimack PK, Sellier M, Thompson HM. Efficient and accurate time adaptive multigrid simulations of droplet spreading. *International Journal for Numerical Methods in Fluids* 2004; **45**:1161–1186.
4. Gupta A K. A finite element for transition from a fine to a coarse grid. *International Journal for Numerical Methods in Engineering* 1978; **12**:35-45.
5. Wang W. Special bilinear quadrilateral elements for locally Refined finite element grids. *SIAM Journal on Scientific Computing* 2001; **22**:2029–2050.
6. Meyer A. Projection techniques embedded in the PCGM for handling hanging nodes and boundary restrictions. In *Engineering Computational Technology,* Topping B.H.V, Bittnar Z (eds). Saxe-Coburg Publications, Stirling, Scotland, 2002; 147–165.
7. Brandt A. Multi-level adaptive solutions to boundary-value problems. *Mathematics of Computation* 1977; **31**(138):333–390.
8. Hart L, McCormick S, O'Gallagher A. The fast adaptive composite-grid method (FAC): algorithms for advanced computers. *Applied Mathematics and Computation* 1986; **19**:103–125.
9. Baines MJ, Hubbard ME, Jimack PK, Jones AC. Scale invariant moving finite elements for nonlinear partial differential equations. Submitted to *Applied Numerical Mathematics* 2004.