

Multigrid Algorithms for Optimal Engineering Computations

Peter K. Jimack

School of Computing, University of Leeds, Leeds LS2 9JT, UK

Abstract

Multigrid methods are becoming increasingly popular within engineering software and this chapter provides a general introduction to the topic, including a short survey and selected case studies. The main concepts behind the multigrid approach are introduced and the key multigrid algorithms are described. A discussion of some of the most significant applications of multigrid for the solution of computational engineering problems is provided and two specific case studies are considered in more detail. These include the solution of nonlinear problems and the incorporation of both mesh adaptivity and parallelism within the solution strategy. The chapter concludes with a brief summary and some conclusions.

Keywords: computational engineering, multigrid, computational efficiency, optimal computational complexity.

1 Introduction

Computational engineering research relies heavily on the numerical solution of mathematical models that are based upon systems of partial differential equations (PDEs). These models arise in disciplines ranging from structural mechanics through to fluid dynamics, acoustics, electromagnetics, etc. Applications include stress analysis in structures, aerodynamic design, multi-phase flow simulation, fluid-structure interaction, and many more.

Solution techniques for PDE-based models are numerous and their suitability depends upon a range of factors. Typically however approaches consist of two distinct stages: discretization and algebraic system solves. Discretization schemes include finite element, finite volume or finite difference methods and lead to the reduction of the problem to the solution of one or more systems of linear or nonlinear algebraic equations. It is the solution of such systems that lies at the heart of most computational

engineering software, and it is this part of the code where the most computational work is typically expended.

In recent years there have been significant advances in computer hardware, which have allowed the number of unknowns that are used by the discretization scheme to be increased dramatically. This has the major advantage of allowing far greater accuracy in the numerical simulations but comes at the expense of needing to solve far larger systems of algebraic equations. Traditional solution methods for linear systems, for example, do not scale well with the problem size and so the computational work grows super-linearly as the number of unknowns, N say, is increased. For example, Gaussian elimination for a banded system, of band-width \sqrt{N} , has a computational complexity of $O(N^2)$. Similar complexities are observed for more general sparse solvers, whether they are direct ([51]) or iterative ([67]) in nature.

The multigrid approach, where it is applicable, seeks to solve the sparse algebraic systems that arise from local discretizations of PDEs (e.g. based upon finite differences/elements/volumes) at a computational cost that is close to optimal, that is $O(N)$. This makes such methods extremely attractive as the size of N , the number of unknowns, increases in the search for ever-greater numerical resolution and accuracy. For this reason there has been a significant growth in the popularity of multigrid in recent years, and its application to the solution of computational engineering problems is now becoming common place. The aim of this chapter is to introduce some of the key concepts behind multigrid and to explore how they have been used in recent years across a range of engineering applications.

The paper begins with a discussion of the main features that define multigrid methods. We then present a short survey of some of the engineering applications that have exploited this technique over recent years. This is followed by some more detailed case studies that are taken from the author's own collaborations. The chapter concludes with a short discussion and summary.

2 Background

It is beyond the scope of this chapter to provide a full introduction to all aspects of the multigrid method, so the aim of this section is to briefly outline some of the main ideas behind multigrid. The particular techniques that are covered include the standard geometric approach for linear problems, and then generalizations to: nonlinear problems (full approximation storage (FAS)); problems with adaptive mesh refinement (the multilevel adaptive technique (MLAT)); and problems for which a geometric mesh hierarchy is not available or not convenient (the algebraic multigrid (AMG) approach). Many further details of these and related techniques can be found in texts such as [34, 76, 77] and references therein.

2.1 The standard multigrid V-cycle

Suppose one wishes to solve a linear elliptic PDE of the form

$$\mathcal{A}u = f \quad (1)$$

on a given domain and with prescribed boundary conditions (where \mathcal{A} denotes the differential operator). Consider the use of a finite difference discretization of (1) on a grid of a desired resolution (which will be referred to here as “the fine grid”), to produce a system of discrete equations of the form

$$A^f \underline{u}^f = \underline{f}^f. \quad (2)$$

It is necessary to solve this system of equations for the finite-dimensional set of values \underline{u}^f corresponding to the solution u approximated at the points of the fine grid. Suppose an approximation to \underline{u}^f has been found: $\hat{\underline{u}}^f$ say. Then the error in the solution of (2) is given by

$$\underline{e}^f = \underline{u}^f - \hat{\underline{u}}^f. \quad (3)$$

Let \underline{r}^f be the residual, given by

$$\underline{r}^f = \underline{f}^f - A^f \hat{\underline{u}}^f. \quad (4)$$

From (3) and (4) it follows that

$$\begin{aligned} A^f \underline{e}^f &= A \underline{u}^f - A \hat{\underline{u}}^f \\ &= \underline{f}^f - (\underline{f}^f - \underline{r}^f) \\ &= \underline{r}^f. \end{aligned} \quad (5)$$

Hence, if we can solve the error equation (5) exactly we can improve upon $\hat{\underline{u}}^f$ to obtain:

$$\underline{u}^f = \hat{\underline{u}}^f + \underline{e}^f. \quad (6)$$

In the multigrid approach we do not solve (5) exactly but instead we approximate the solution by discretizing the PDE (1) on a coarse grid and then solving the corresponding error equation exactly on this coarse grid:

$$A^c \underline{e}^c = \underline{r}^c. \quad (7)$$

In (7) \underline{r}^c is obtained by restricting the residual \underline{r}^f (from (4)) to the coarse grid. The solution of (7) is cheaper to obtain than the solution of either (2) or (5) since the corresponding grid is coarser and the number of unknowns is therefore fewer. Once \underline{e}^c has been found from (7) it may be used to approximate \underline{e}^f in (6) by interpolating values of \underline{e}^c from the coarse grid to the fine grid. Let this interpolated vector be denoted as $\hat{\underline{e}}^f$, then

$$\underline{u}^f \approx \hat{\underline{u}}^f + \hat{\underline{e}}^f. \quad (8)$$

This improved approximation can then be used to continue the iterative solution process on the fine grid.

Function TwoGrid(\underline{v}^f)

1. Let the initial fine grid solution be \underline{v}^f .
2. Update \underline{v}^f applying ρ_1 iterations of the selected smoother.
3. Find the residual: $\underline{r}^f := A^f \underline{v}^f - \underline{f}^f$.
4. Restrict the residual to the coarse grid: $\underline{r}^f \rightarrow \underline{r}^c$.
5. Solve the error equation on the coarse grid: $A^c \underline{e}^c = \underline{r}^c$.
6. Interpolate the correction to the fine grid: $\underline{e}^c \rightarrow \underline{e}^f$.
7. Correct the fine grid solution: $\underline{v}^f = \underline{v}^f + \underline{e}^f$.
8. Update \underline{v}^f by applying ρ_2 iterations of the selected smoother.
9. Return

Figure 1: Two-grid solution algorithm that forms the basis for the multigrid solution of the linear system $A^f \underline{v}^f = \underline{f}^f$.

For the above approach to be successful two properties must hold. The iterative solver that is used on the fine grid should possess a “smoothing property”, which implies that it damps out the highest frequency components of the error the most quickly. Hence, the coarse grid correction, equation (7) should successfully remove the low frequency components of the error. An illustration of this two grid algorithm is shown in Fig. 1. Note that this process of “smooth-correct-smooth” is known as a single cycle, and should be repeated until convergence.

The multigrid method does not just use two grids, as in Fig. 1. Instead, the coarse grid correction problem (the error equation at step 5 in Fig. 1) is solved using a similar approach to that used for the original problem. That is, a small number of smoothing iterations are taken, then an even coarser grid correction is made, followed by a few more smoothing iterations. This coarser grid correction problem may also be “solved” in the same way, using an even coarser grid, and so on, until the coarsest possible grid is reached. At this point an exact solve is possible at negligible cost. This complete multigrid algorithm is illustrated in Fig. 2. At grid level k the discretization of (1) is assumed to take the form

$$A^{(k)} \underline{u}^{(k)} = \underline{f}^{(k)} \quad (9)$$

for $k = 0, \dots, K$. Furthermore, $k = K$ is the grid level at which the solution is

Function $\text{MultiGrid}(\underline{v}^{(k)}, \underline{f}^{(k)}, k)$

1. Let the initial grid k solution be $\underline{v}^{(k)}$.
2. Update $\underline{v}^{(k)}$ applying ρ_1 iterations of the selected smoother.
3. Find the residual: $\underline{r}^{(k)} := A^{(k)}\underline{v}^{(k)} - \underline{f}^{(k)}$.
4. Restrict the residual to the next coarsest grid: $\underline{r}^{(k)} \rightarrow \underline{r}^{(k-1)}$.
5. Solve the error equation on the next coarsest grid:
 if $(k = 1)$ then
 Solve $A^{(0)}\underline{e}^{(0)} = \underline{r}^{(0)}$ directly
 else
 Initialize $\underline{e}^{(k-1)} = \underline{0}$
 MultiGrid($\underline{e}^{(k-1)}, \underline{r}^{(k-1)}, k - 1$).
6. Interpolate the correction to the fine grid: $\underline{e}^{(k-1)} \rightarrow \underline{e}^{(k)}$.
7. Correct the fine grid solution: $\underline{v}^{(k)} = \underline{v}^{(k)} + \underline{e}^{(k)}$.
8. Update $\underline{v}^{(k)}$ by applying ρ_2 iterations of the selected smoother.
9. Return

Figure 2: Multigrid solution algorithm for the solution of the linear system $A^{(K)}\underline{v}^{(K)} = \underline{f}^{(K)}$.

required, whilst $k = 0$ represents the coarsest grid in the hierarchy. Note that the algorithm in Fig. 2 is presented in the form of a function which calls itself recursively until the coarsest grid is reached.

The mathematical justification for this algorithm is beyond the scope of this chapter (see, for example, [77]) however it should be noted that for self-adjoint linear elliptic problems Gauss-Seidel iteration, or block Gauss-Seidel iteration for PDE systems, is generally a satisfactory smoother. The optimality criterion, which requires the total solution time to grow as $O(N)$ as the number of degrees of freedom, N , is increased, will be achieved provided that the work per iteration of the smoother is proportional to N and the number of full cycles required to converge is independent of N . The former property is relatively easy to achieve using Gauss-Seidel iteration (say) for a finite difference/element/volume discretization provided that the number of unknowns in each grid forms a geometric series. The latter requires that the smoothing property

and the coarse grid correction are both achieved. This performance is widely observed in many practical implementations, [13, 43, 64].

2.2 Nonlinear problems

The description in the previous subsection is applicable to linear PDEs only. Discretization of a nonlinear PDE leads to a nonlinear system of algebraic equations and so a different solution strategy is required. There are two main classes of method for using multigrid to solve these nonlinear systems of equations. The first of these is to use a Newton or quasi-Newton iteration in order to obtain a sequence of linear algebraic systems which can each be solved using a linear multigrid approach, similar to that outlined above. This technique has been used successfully across a variety of applications, as in [17, 40, 71] for example. The alternative strategy is to develop a multigrid algorithm that may be applied directly to nonlinear problems, and this is the approach that is described in the remainder of this subsection.

The full approximation scheme (FAS) was proposed by Brandt in his original 1977 paper [12]. This algorithm requires the use of a nonlinear iteration, such as a nonlinear version of Gauss-Seidel for example, to act as a smoother on each grid. Furthermore, the coarse grid correction is obtained not by solving an error equation but by solving a modified version of the discrete PDE on the coarser grid. This modification takes the form of an additional term on the right-hand side which is based upon the difference between the restricted fine grid residual and the coarse grid residual of the restricted fine grid solution.

To illustrate these steps, suppose that equation (1) now represents a nonlinear elliptic PDE. Let the discretization of this PDE yield a nonlinear system of algebraic equations of the form

$$\underline{F}^{(K)}(\underline{u}^{(K)}) = \underline{f}^{(K)} \quad (10)$$

on grid level K , where $\underline{F}^{(K)}$ is a nonlinear vector-valued function. The nonlinear Gauss-Seidel smoother for this system takes the form:

$$u_i^{(K)} = u_i^{(K)} - \frac{[\underline{F}^{(K)}(\underline{u}^{(K)}) - \underline{f}^{(K)}]_i}{\frac{\partial F_i^{(K)}}{\partial u_i^{(K)}}(\underline{u}^{(K)})}, \quad (11)$$

for $i = 1, \dots, N$ (where N is the number of unknowns on grid level K). Furthermore, the coarse grid correction is obtained by solving the following system at grid level $K - 1$:

$$\begin{aligned} \underline{F}^{(K-1)}(\underline{u}^{(K-1)}) &= \underline{f}^{(K-1)} - [\underline{r}^{(K-1)} - (F^{(K-1)}(\hat{\underline{u}}^{(K-1)}) - \underline{f}^{(K-1)})] \\ &= \underline{F}^{(K-1)}(\hat{\underline{u}}^{(K-1)}) - \underline{r}^{(K-1)}, \end{aligned} \quad (12)$$

where $\hat{\underline{u}}^{(K-1)}$ is the restriction of $\underline{u}^{(K)}$ and $\underline{r}^{(K-1)}$ is the restriction of $\underline{r}^{(K)}$ to grid level $K - 1$. Once (12) is solved for $\underline{u}^{(K-1)}$ we may prescribe

$$\underline{e}^{(K-1)} = \underline{u}^{(K-1)} - \hat{\underline{u}}^{(K-1)} \quad (13)$$

and then interpolate this up to the fine grid (level K) to obtain the coarse grid correction $\underline{e}^{(K)}$. A summary of this FAS multigrid algorithm is shown in Fig. 3.

Note that the nonlinear iteration is not guaranteed to converge and so the choice of initial guess for $\underline{u}^{(K)}$ becomes highly significant. Often this is obtained based upon a so-called full multigrid (FMG) approach. This begins by finding a solution on the coarsest grid and interpolating this to the next grid as an initial guess. A fixed number of multigrid cycles (typically one or two) are taken on this grid to obtain a good approximation to the solution (on this grid), which is then interpolated to the next grid. This process is repeated until the finest grid is reached, whereupon we expect the initial guess to be sufficiently good to ensure convergence of the method. For further details see, for example, [34, 76].

2.3 Multigrid with non-uniform mesh refinement

In the previous subsections nothing has been said explicitly about the relationships between the fine and the coarse grids that are required for the multigrid approach to be successful. In fact, the ways in which these grids are related has a direct bearing on the interpolation and the restriction operations that can be used. In the simplest case the grids are fully nested, as illustrated in Fig. 4 for example, which means that the grid at level k is always a uniform refinement of the grid at level $k - 1$ (for $k = 1, \dots, K$). In this situation, straightforward linear interpolation is generally suitable and this operation may be represented by a rectangular matrix I_{k-1}^k such that

$$\underline{e}^{(k)} = I_{k-1}^k \underline{e}^{(k-1)}. \quad (14)$$

The restriction operator may also be expressed through a rectangular matrix, I_k^{k-1} say, to give

$$\underline{r}^{(k-1)} = I_k^{k-1} \underline{r}^{(k)}, \quad (15)$$

for example. For fully nested grids co-called “full-weighting” is typically used to undertake restriction which, for a finite difference discretization, means that

$$I_k^{(k-1)} = (I_{k-1}^k)^T, \quad (16)$$

where $(I_{k-1}^k)^T$ is the transpose of the linear interpolation matrix.

In many practical engineering applications it is not sensible for the finest grid, on which we are computing the desired solution, to be of uniform density throughout the entire domain, as illustrated in Fig. 4. Instead, greater resolution is frequently desired in some regions compared to others: to provide the required accuracy without using an excessive number of degrees of freedom. One way of achieving this is to use local, rather than uniform, mesh refinement. This is illustrated in Fig. 5, where the greatest resolution is required toward the bottom right of the domain. The grids are still nested, but not fully nested as in Fig. 4.

Multigrid may still be applied to solve problems using a sequence of non-uniformly refined grids, such as that illustrated in Fig. 5. In fact, there are numerous possible ways in which this may be achieved: only one of which is described here. The

Function FASMultiGrid($\underline{v}^{(k)}, \underline{f}^{(k)}, k$)

1. Let the initial grid k solution be $\underline{v}^{(k)}$.
2. Update $\underline{v}^{(k)}$ applying ρ_1 iterations of the selected smoother.
3. Find the residual: $\underline{r}^{(k)} := \underline{F}^{(k)}(\underline{v}^{(k)}) - \underline{f}^{(k)}$.
4. Restrict the residual to the next coarsest grid: $\underline{r}^{(k)} \rightarrow \underline{r}^{(k-1)}$.
5. Restrict the solution to the next coarsest grid: $\underline{v}^{(k)} \rightarrow \underline{v}^{(k-1)}$.
6. Calculate the coarse grid right-hand side: $\underline{f}^{(k-1)} = \underline{F}^{(k-1)}(\underline{v}^{(k-1)}) - \underline{r}^{(k-1)}$.
7. Save the initial coarse grid solution: $\hat{\underline{v}}^{(k-1)} = \underline{v}^{(k-1)}$.
8. Solve the coarse grid system:
 - if ($k = 1$) then
 - Solve $\underline{F}^{(0)}(\underline{v}^{(0)}) = \underline{f}^{(0)}$ directly
 - else
 - Initialize $\underline{v}^{(k-1)} = \hat{\underline{v}}^{(k-1)}$
 - FASMultiGrid($\underline{v}^{(k-1)}, \underline{f}^{(k-1)}, k - 1$).
9. Calculate the coarse grid correction: $\underline{e}^{(k-1)} = \underline{v}^{(k-1)} - \hat{\underline{v}}^{(k-1)}$.
10. Interpolate the correction to the fine grid: $\underline{e}^{(k-1)} \rightarrow \underline{e}^{(k)}$.
11. Correct the fine grid solution: $\underline{v}^{(k)} = \underline{v}^{(k)} + \underline{e}^{(k)}$.
12. Update $\underline{v}^{(k)}$ by applying ρ_2 iterations of the selected smoother.
13. Return

Figure 3: FAS multigrid solution algorithm for the solution of the nonlinear system $\underline{F}^{(K)}(\underline{v}^{(K)}) = \underline{f}^{(K)}$.

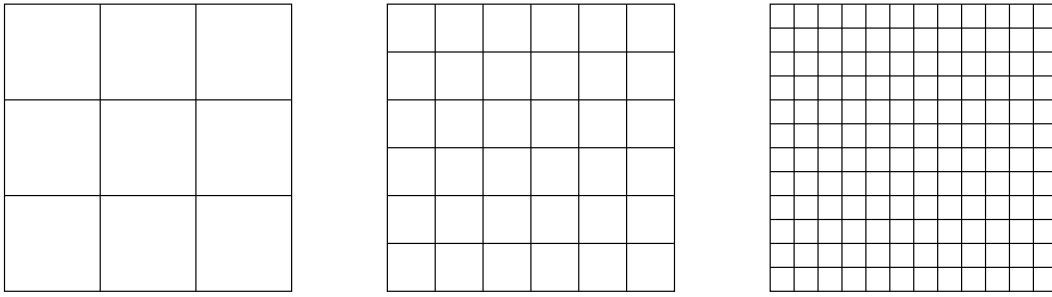


Figure 4: Sequence of three fully-nested grids in two dimensions.

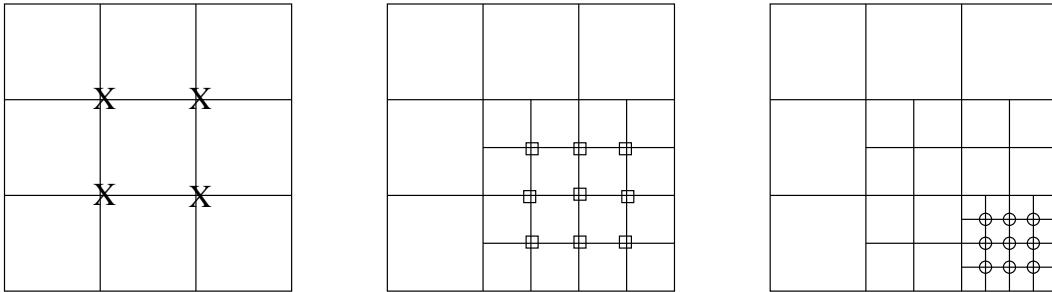


Figure 5: Sequence of three locally-nested grids in two dimensions.

approach first proposed by Brandt in [12] is known as MLAT (multilevel adaptive technique) but more recent alternatives are described in [6, 32, 43] for example.

The MLAT scheme has the advantages of applying directly to nonlinear problems (of which linear systems are an obvious special case) and of not requiring any modification of the underlying discretization scheme in order to take account of the “hanging” nodes at the interface between levels. This is achieved by treating the fine grid as only the refined region of the coarse grid and applying Dirichlet boundary conditions throughout the perimeter of this region. Referring to Fig. 5 the circles therefore represent the unknowns on the finest grid (assuming, for simplicity, that the domain boundary is subject to Dirichlet conditions everywhere). The coarse grid correction is then obtained by using the FAS scheme and solving at the next level using the unknowns shown by the square symbols. The right-hand side for this problem is only modified for those nodes which lie in, or at the interface with, the locally refined region. Finally, in this three grid example, the coarsest grid solution covers the whole domain and is calculated at those nodes denoted by the crosses. In this particular example the right-hand side will be modified for each of these nodes since they all overlap with the locally refined region.

2.4 Algebraic multigrid methods

The previous examples described in this section all assume that a nested sequence of grids is available in order to compute corrections on coarser and coarser grids. This nesting requirement puts a significant constraint on the way in which the finest grid may be generated since it must be the result of a sequence of refinements from the coarsest grid. Often however the grid that we wish to use to obtain a numerical solution is not generated in such a manner. This is particularly true when finite element or finite volume schemes are used on unstructured grids. These may be generated directly from a mesh generation tool such as [26, 38, 68] for example. In this situation no hierarchy of meshes is available and so the standard multigrid approach cannot be followed.

Algebraic multigrid (AMG) methods seek to overcome this problem by generating the coarse grid problem from the fine grid problem in, as the name suggests, an algebraic way. In the linear case, for example, this requires the matrix $A^{(k-1)}$ to be generated from $A^{(k)}$ without explicitly discretizing the problem on a coarser grid. When using uniformly-refined sequences of grids, as in Fig. 4 for example, it is relatively straightforward to devise a set of algebraic operations on the matrix $A^{(k)}$ that allow the smaller matrix $A^{(k-1)}$ to be recovered exactly. Such a sequence of operations may be used as the basis for a purely algebraic method applied to a different matrix, resulting from a discretization of (1) on an arbitrary grid. This is the principle that drives the AMG approach. The main challenge is to produce a robust technique for coarsening the discretization matrix and for undertaking the corresponding restriction and interpolation operations. This important issue will not be covered further in this chapter however the reader is referred to articles such as [11, 20, 35, 49, 73] for further details and examples.

A final point that it is important to note within this background section is that both the standard linear multigrid method and the AMG technique are frequently used in the form of preconditioners for other iterative solvers, rather than as solvers in their own right. This is particularly so for AMG. The typical approach is to apply a Krylov subspace solver (GMRES for example, [67]) either to the discrete equations directly (in the case of a linear PDE) or to the Jacobian of the discrete equations (in the case of a quasi-Newton method for nonlinear problems), and employ a single cycle of the multigrid algorithm as a preconditioner at each iteration. Such an approach is also capable of delivering optimal computational complexity for suitable problems and extends very naturally to the AMG case [9].

3 Selected Contributions

In this section we provide a short description of some of the many published applications of multigrid methods in computational engineering. The volume of the literature is so vast that we make no attempt to provide a systematic coverage of the many techniques and applications that have been developed, however the section is organised around three broad topics: fluid dynamics, solid mechanics and electromagnetics.

3.1 Fluid dynamics

Fluid flow problems may be divided into a large number of sub-classes, such as Newtonian or non-Newtonian, compressible or incompressible, viscous or inviscid, etc. The resulting governing equations can have very different characteristics and so there is no single multigrid algorithm that is best, or even entirely appropriate, for all such problems. For a discussion of techniques that are used in order to achieve optimal, or close to optimal, performance see, for example, [75]. This review article focuses particularly on so-called “factorizability” properties for compressible Navier-Stokes systems, so that the elliptic and hyperbolic components may be treated separately, and optimally, in order to attain “textbook multigrid efficiency” for aerodynamic simulations.

Other papers which consider aerodynamic configurations include [23, 71]. In the former case particular care is again taken to treat the convection part of the equations separately, and appropriately, via an upwind-based multigrid smoother. Complex geometric configurations are modelled through a multiblock technique in which each block contains a structured mesh hierarchy that is suitable for geometric multigrid. In [71] a very different approach is taken to dealing with complex geometries, based upon the use of a discontinuous Galerkin (DG) discretization on an unstructured grid. The authors are able to obtain optimal performance for both a nonlinear multigrid solver and a Newton-Krylov solver that uses multigrid as a preconditioner. In one of the few papers to explicitly contrast both approaches they conclude that the second of these (multigrid preconditioned Newton-Krylov) is the more efficient and the more robust for the problems that were considered.

Each of the papers cited above considers compressible flow models. For a detailed description of optimal preconditioning methods for incompressible flow problems the book [22] is highly recommended. In addition to this however there have been numerous articles written on the application of multigrid to incompressible flow problems. Given the challenging computational scale of these problems a popular theme has been the parallel implementation of such techniques, e.g. [19, 21, 74]. Each of these papers considers time-dependent problems through the use of implicit time-stepping, leading to nonlinear algebraic systems of equations at each time step. Different discretization schemes are used however: second order finite differences on a structured staggered grid, finite volumes on a block-structured grid and finite volumes on an unstructured tetrahedral non-nested grid, respectively. In each case good parallel performance is achieved but only on moderate numbers of processors. This reflects a common difficulty with parallel implementations of multigrid since the coarse grid operations, which are an essential component of any multigrid procedure, are inherently difficult to implement efficiently in parallel. Despite, or perhaps because of, this there is a vast literature on the development of parallel multigrid algorithms, that goes well beyond the scope of this chapter. See [1, 2, 5, 14, 35, 39, 49] for just some of the many examples.

Finally, we note some further contributions to multigrid algorithms for incompressible flow problems. In particular, the existence of boundary layers and simi-

lar features has led authors to consider the use of either stretched, [56], or locally-adapted, [31, 52], grids in combination with multigrid solvers. This can lead to highly anisotropic meshes which can, in turn, result in some drop in the multigrid performance if appropriate action is not taken. Indeed, the ill-conditioned nature of systems arising from the use of highly anisotropic grids, and the resulting need for efficient solvers, has been reported widely in the literature, e.g. [3]. These issues are considered in detail in [31, 56].

3.2 Solid mechanics

Multigrid methods have also been widely applied to the solution of computational models in solid mechanics. Indeed, such problems are generally very well suited to multigrid and algebraic multigrid solvers due to their elliptic nature. Perhaps the simplest example of this is the linear elastostatic problem, which has been solved optimally using a variety of discretizations including least squares finite element methods [16, 15], mixed finite element methods [50, 59] and pure displacement models [54, 29].

The latter two papers provide examples of the efficient use of algebraic multigrid preconditioners for linear elasticity problems. The key to success, as described in [54, 55] and in more recent developments such as [9], is to order the system in a blockwise fashion (based upon the displacement in coordinate direction at a time) and then apply a block-diagonal AMG preconditioner with a Krylov solver such that each block consists of the action of a single AMG V-cycle. Other applications of AMG to these problems include [10], where parallel algebraic multigrid preconditioners are successfully applied to a range of structural mechanics problems.

Nonlinear problems have also been considered, including multibody contact problems [18, 78], large deformation elasticity [1, 53] and fluid-structure interaction problems [33, 36]. These last two papers both address the fully-coupled nonlinear system directly (although [36] also considers decoupled approaches) through their different discretization schemes. The least squares system approach of [36] uses an algebraic multigrid preconditioner for the conjugate gradient solution of the linearized systems, whereas [33] uses a standard finite element discretization of each set of equations and a block-triangular approximation to the Jacobian matrix as a preconditioner at each Newton iteration. The action of a multigrid cycle is used for the diagonal blocks of the preconditioner which correspond to the elastic deformation problems.

3.3 Electromagnetics

Just as for applications in fluid dynamics and solid mechanics, multigrid has been applied widely, and in various forms, in electromagnetics and associated applications. In this subsection we focus primarily on the solution of Maxwell's equations and Helmholtz problems.

Perhaps the most significant contribution to the development of multigrid methods

for Maxwell's equations is due to [37] in which a rigorous demonstration of the optimality of a suitably constructed multigrid solver is first presented. This is based upon a mixed variational formulation of the problem which is discretized with appropriately conforming finite elements (the Nedelec edge elements, [57]) to yield a discrete algebraic system for which an appropriate multigrid scheme is developed. A similar discretization is used in [30], however they eliminate a block of the mixed finite element system and solve the resulting Schur complement system, which is symmetric and positive-definite, by a parallel multigrid preconditioned conjugate gradient iteration. Good efficiency is observed however optimality is not proved. An algebraic multigrid version of the mixed finite element approach is presented in [62]. This again assumes a suitable conforming finite element discretization and shows good performance on certain test problems. Subsequently [7, 41] have generalized this method so as to prove h -independent convergence under certain conditions, and to improve execution times. Finally, we note that other discretization schemes are possible. For example, [4] use a finite volume discretization of a vector-potential formulation of the problem. Their multigrid preconditioning approach yields a system whose condition number is bounded independently of the mesh size.

The Helmholtz equation has also received considerable recent attention from the multigrid community. The problem is hard to solve when the wave number is real and large since the coarse grid problem (without modification) requires multiple points per wavelength to maintain stability. In [46] therefore the emphasis is on the efficient solution of this coarse grid problem. More recently, [25] introduce a multigrid preconditioner which involves shifting certain eigenvalues through at the coarse grid correction step. This idea is further developed in [24], with both papers demonstrating effective multigrid performance for high wave number problems. AMG is very challenging for these equations due to the fact that the discrete operator is non-self-adjoint and, as noted by [58], it permits oscillatory error components which yield relatively small residuals. Nevertheless, provided sufficient *a priori* knowledge is included, successful algebraic multigrid algorithms have begun to be developed [8, 58].

4 Case Studies

This section briefly describes two case studies that are taken from the author's own work. The first of these demonstrates the combination of multigrid techniques with mesh adaptivity, whilst the second illustrates a typical parallel implementation of multigrid. In both cases the nonlinear, FAS, version of multigrid is used, as introduced in Section 2.2 above.

4.1 Multigrid and adaptive mesh refinement

The work described here is based primarily on the PhD thesis [63] and the subsequent publications that have arisen from it [64, 65, 66]. One particular application is pre-

sented, however the need for combining adaptive spatial discretizations with multigrid solution techniques arises across a wide range of computational engineering problems. In particular, when solutions exhibit local features, such as steep fronts, discontinuities or boundary layers. Perhaps the most popular technique for undertaking the adaptivity is to make use of local mesh refinement in the regions that require the greatest resolution, and to use a coarser mesh elsewhere. This is most frequently achieved through a hierarchical approach in which one starts with a coarse mesh and then refines this more in some regions than in others. In two dimensions this may be undertaken using a quad-tree data structure (e.g. [60]) and in three dimensions an oct-tree structure is used (e.g. [42]). Here we demonstrate how such an adaptive technique may be successfully combined with multigrid for the solution of nonlinear systems of algebraic equations obtained from the discretization of a particular set of partial differential equations (PDEs) arising in the modelling of phase change problems.

One of the most popular techniques to have emerged in recent years for the modelling and simulation of phase change problems is the phase field method. Details of this technique can be found in numerous articles, e.g. [44, 45, 61], however the basic idea is quite simple. Instead of treating the evolving boundary between the solid and liquid regions as being discrete, it is treated as being diffuse and an additional (smooth) dependent variable is introduced to represent the phase as a function of space and time. Evolution of this phase variable is prescribed by a PDE that is derived from a governing free energy functional, and there is typically a parameter present within the formulation that allows the width of the diffuse interface to be controlled. This type of model is able to benefit significantly from the use of mesh adaptivity around the interface in order to capture this aspect of the solution as accurately as possible.

A large number of different phase-field models have been presented in the literature and we present just one of these here. This is taken from [61] and provides a model for the solidification of a dilute binary alloy, in which both the concentration of the minor alloy component (U) and the variation of temperature (T) are important drivers. These are both represented as non-dimensional variables in the governing equations, below, which also feature the phase variable ϕ :

$$A(\nabla\phi)^2 \frac{\partial\phi}{\partial t} = \nabla \cdot (A^2(\nabla\phi)\nabla\phi) + \frac{\partial}{\partial x} \left(|\nabla\phi|^2 A(\nabla\phi) \frac{\partial A(\nabla\phi)}{\partial(\phi_x)} \right) + \frac{\partial}{\partial y} \left(|\nabla\phi|^2 A(\nabla\phi) \frac{\partial A(\nabla\phi)}{\partial(\phi_y)} \right) + \phi - \phi^3 - \lambda(1 - \phi^2)^2 (T + Mc_\infty U), \quad (17)$$

$$\frac{1+k}{2} \frac{\partial U}{\partial t} = \nabla \cdot \left(D \frac{1-\phi}{2} \nabla U + \frac{1}{2\sqrt{2}} [1 + (1-k)U] \frac{\partial\phi}{\partial t} \frac{\nabla\phi}{|\nabla\phi|} \right) + \frac{1}{2} \frac{\partial}{\partial t} \{ \phi [1 + (1-k)U] \}, \quad (18)$$

$$\frac{\partial T}{\partial t} = \alpha \nabla^2 T + \frac{1}{2} \frac{\partial\phi}{\partial t}. \quad (19)$$

Note that the above system is highly nonlinear and is made stiff by the fact that the diffusion coefficients D and α typically differ by a number of orders of magnitude.

The terms $A(\nabla\phi)$ are nonlinear functions of the derivatives of ϕ which are used to define the anisotropy within the solidification.

The spatial discretization that we use is based upon a second order finite difference scheme, whilst temporal discretization is undertaken using the fully implicit BDF2 scheme which is well-suited to stiff problems. Mesh adaptivity is controlled using a simple error indicator which is based upon a linear combination of the gradients of the three dependent variables. The resulting systems of nonlinear algebraic equations that occur at each time step are solved using the MLAT scheme described in Section 2.3. Figure 6 illustrates a typical adaptive mesh (where the different grey levels represent different levels of refinement) and Figure 7 contains graphs that demonstrate the optimality of the multigrid implementation. Each of these figures is courtesy of J. Rosam.

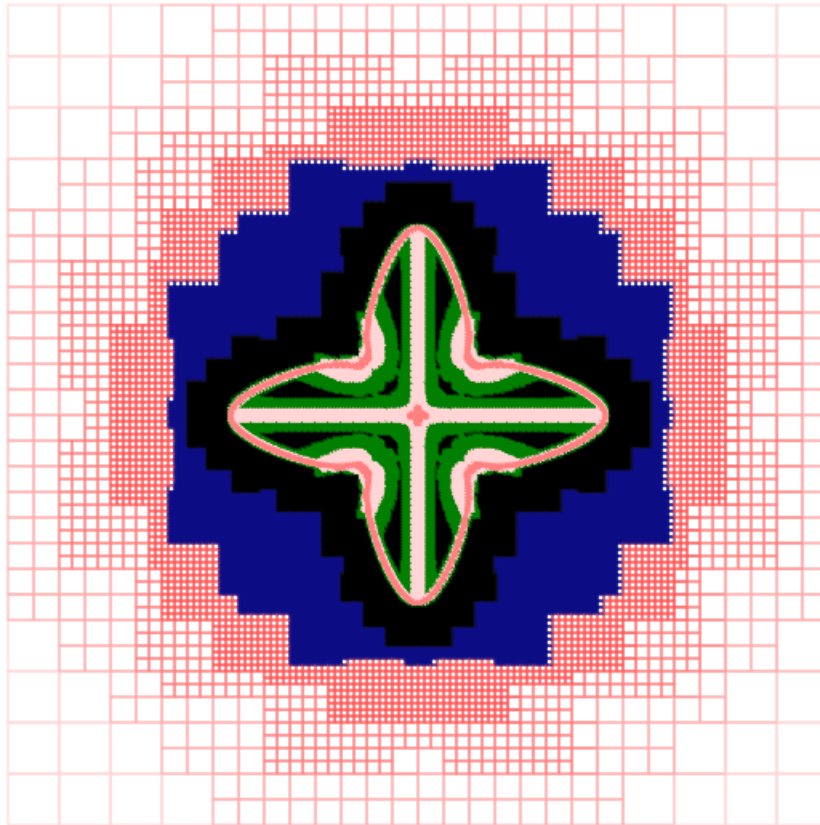


Figure 6: A typical adapted mesh (courtesy of J. Rosam).

4.2 A parallel multigrid implementation

The work described in this sub-section is based primarily on the PhD thesis [47] and the subsequent publications arising from it [27, 48]. As in the previous sub-section just

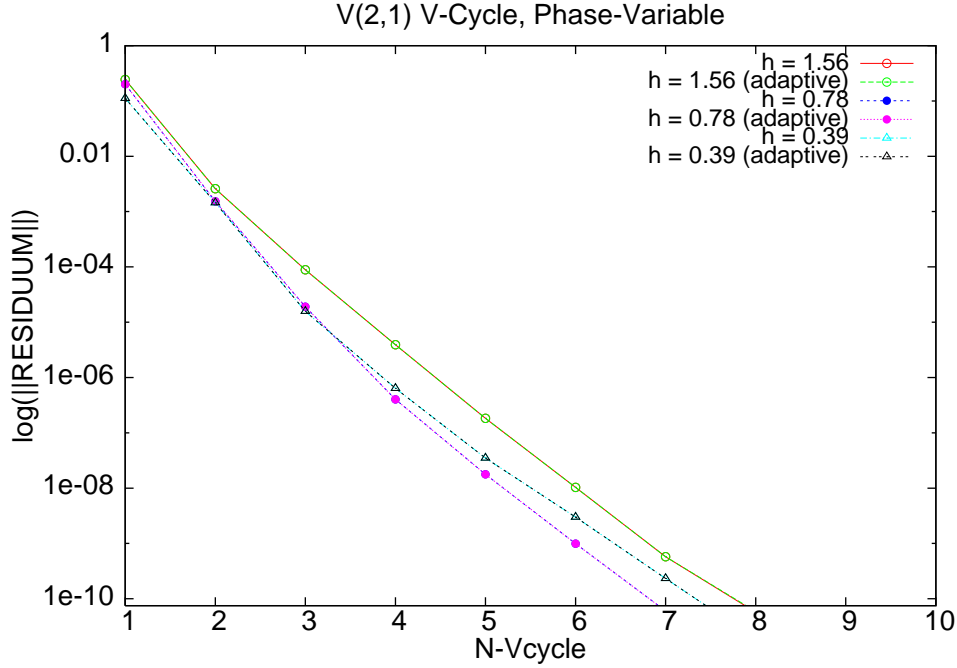


Figure 7: Multigrid convergence for different levels of mesh (courtesy of J. Rosam).

one application is presented here but this is typical of a much larger set of problems for which parallel implementations of multigrid have been developed, e.g. [19, 21, 28, 74].

In this example we consider the flow of a viscous droplet of fluid spreading down an inclined plane subject to gravitational and surface tension forces. The model used is based upon a long wave approximation, in which it is assumed that the flow perpendicular to the plane is negligible compared to the flow parallel with the inclined plane. A small parameter, ϵ , may be defined as the ratio of typical length scales, H_0 and L_0 , in the perpendicular and parallel directions respectively. It is then possible to non-dimensionalize the Stokes equations and collect together the leading order terms in this small parameter in order to yield a simplified system of equations for this thin film flow. Further details are provided in [69] for example, but the resulting PDEs take the following form:

$$\frac{\partial h}{\partial t} = \frac{\partial}{\partial x} \left[\frac{h^3}{3} \left(\frac{\partial p}{\partial x} - \frac{Bo}{\epsilon} \sin \alpha \right) \right] + \frac{\partial}{\partial y} \left[\frac{h^3}{3} \left(\frac{\partial p}{\partial y} \right) \right], \quad (20)$$

$$p = -\nabla^2(h + s) - \Pi(h) + Bo \cos \alpha (h + s - z). \quad (21)$$

In the above, h and p are the dependent variables (film height and pressure respectively), Bo is the dimensionless Bond number (which represents the ratio of gravitational to surface tension forces) and α is the angle made by the inclined plane. Figure 8 (courtesy of Y.-Y. Koh [47]) provides an illustration of the problem being modelled. The function s is a known function describing surface roughness on the plane whilst

the term $\Pi(h)$ is a so-called *disjoining pressure* term that is used to model slip of the moving contact line. This term is based upon the assumption of the presence a thin precursor film, of non-dimensional thickness h^* , and takes the general non-dimensional form:

$$\Pi(h) = \frac{(n-1)(m-1)}{h^*(n-m)} \sigma (1 - \cos \theta_c) \left[\left(\frac{h^*}{h} \right)^n - \left(\frac{h^*}{h} \right)^m \right]. \quad (22)$$

Here, θ_c represents the contact angle between the droplet and the surface (which may differ for an advancing or retreating fluid) and the pair (n, m) are the exponents of the chosen interaction potential ((3, 2) in this example). For further details see, for example, [69, 72].

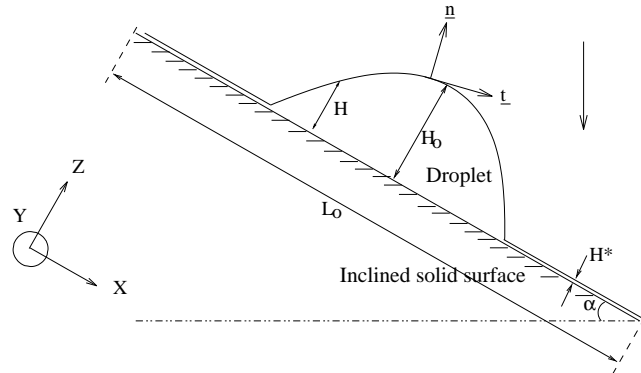


Figure 8: A schematic of the droplet problem (courtesy of Koh [47]).

The solution method that is employed is based upon a parallel implementation of the nonlinear multigrid solver described in detail in [70]. This is based upon the use of finite differences in space with implicit time-stepping and an FAS multigrid solver for the nonlinear algebraic equations that arise at each step. In order for the precursor film model to be physically realistic it is essential that the precursor film thickness, h^* , be sufficiently small (for example, [72] suggests that the dimensional size should be in the range 1-100nm). Furthermore, to capture correspondingly small values of h^* in the above model it is essential that the spatial resolution be on a similar length scale: this is achieved here through the use of very fine grids and parallel computation.

The parallelism is achieved via a geometric decomposition of the domain, which is enforced across all levels of the multigrid hierarchy, and a stripwise partition is used to ensure that each subdomain/processor has at most two immediate neighbours (with a one-to-one mapping between processors and subdomains). Each processor is responsible for implementing the FAS algorithm on its own subdomain, making use of additional columns of “ghost nodes” immediately to either side of this subdomain. These ghost nodes are necessary for neighbour-to-neighbour communications at the following stages of the multigrid algorithm:

1. after each red and each black sweep of the nonlinear Red-Black Gauss-Seidel smoother at each grid level,

2. prior to restriction of the residual and solution to each coarser level,
3. to obtain the exact solution at the coarsest level,
4. prior to interpolation of the error to each finer level.

In addition, a single global communication is required after each multigrid V-cycle in order to decide if convergence has been achieved. Note also that the size of the coarsest possible grid is determined by the number of processors being used since, for the simplicity of this implementation, there should be at least one row of the coarsest grid on each processor.

The scalability of this parallel implementation is illustrated by Tables 1 and 2 (courtesy of Koh, [47]) which show typical solutions times for problems with fine grids of size 1025×1025 and 8193×8193 respectively. It is clear that for the smaller of these problems there is no advantage to be gained by using 128 processors for example. On the other hand the larger problem is solved on 128 processors in only a little over half of the time required to obtain a solution on 64 processors, this demonstrating excellent speed-up. An alternative indication of the scalability of this implementation comes from inspecting the underlined figures in each table: these compare the solution times when the amount of work per processor is almost identical (a fine grid of 1025×1025 across 2 processors versus a fine grid of 8193×8193 across 128 processors) and thus give an indication of the loss of efficiency due to the additional interprocessor communication on 128 processors.

Multigrid Levels	Coarse Grid	Number of Processors							
		1	2	4	8	16	32	64	128
3	257	–	119.54	62.21	30.24	13.84	8.10	6.23	5.31
4	129	–	<u>56.88</u>	28.52	14.20	7.96	3.94	2.91	3.14
5	65	–	57.10	28.89	13.89	7.21	3.37	2.61	–
6	33	–	65.32	28.91	15.66	7.68	4.65	–	–

Table 1: Parallel multigrid timings on a 1025×1025 fine grid (courtesy of Koh, [47]).

Multigrid Levels	Coarse Grid	Number of Processors		
		32	64	128
5	513	250.72	154.85	<u>73.57</u>
6	257	247.54	151.04	74.17
7	129	255.96	156.39	77.94

Table 2: Parallel multigrid timings on a 8193×8193 fine grid (courtesy of Koh, [47]).

5 Discussion

This chapter has discussed the application of multigrid methods to the solution of computational engineering problems arising in a wide range of application areas. The goal of all multigrid solvers is to obtain an accurate solution in $O(N)$ operations, or close to this, where N is the number of degrees of freedom used in the numerical approximation. In order to achieve this a number of components are required.

1. The discretization of the governing equations should result in a sparse algebraic system. This is typically achieved through the use of finite difference, finite element or finite volume methods.
2. An iterative solver must be developed which has the smoothing property that ensures that the high frequency components of the error, relative to the grid that is being used for the discretization, are damped quickly.
3. It should be possible to restrict the problem either geometrically (to a coarser grid) or algebraically (to produce a discrete system which is representative of an approximation of the continuous operator on a coarser grid). This restricted problem may either represent the error equation, as in the conventional linear multigrid approach, or the original equation (with a modified source term), as in the FAS approach. In either case its solution must allow the lower frequency components of the error on the original grid to be eliminated: the coarse grid correction.
4. A recursive formulation is typically used to allow the coarse grid correction to be obtained using the smoother followed by a correction on an even coarser grid. Only the problem at the coarsest level need be solved exactly.
5. Optimal performance will be achieved if the cost of applying the smoother to eliminate the highest frequency error components is $O(N)$, the number of unknowns on each grid reduces geometrically, and the number of cycles through all of the grids required for convergence is independent of the size of the finest grid.

Having introduced the fundamental concepts of multigrid, the remainder of the chapter is divided into two more parts. The first of these provides a short review of the application of multigrid to a range of engineering problems in fluid flow, solid mechanics and electromagnetics. Common themes that arise include the widespread use of AMG, rather than geometric multigrid, and frequent attempts to employ multigrid in parallel, despite the inherent difficulties associated with solving coarse grid problems efficiently on multiple processors.

Finally, the chapter provides two case studies taken from the author's own research. The first of these illustrates the application of multigrid with local mesh refinement. This is in the context of the simulation of a phase change problem using second order finite differences in space a fully implicit, stiff, time integrator. The second study

shows an application of multigrid on a parallel computing system. Here the chosen application is the spreading of a thin liquid drop, which is also governed by a time-dependent nonlinear parabolic system of partial differential equations. Again a second order finite difference scheme is used in space and an implicit second order time-stepping rule is applied. The parallelism is achieved through a geometric decomposition of the problem and the use of columns of “ghost nodes” at the edge of each subdomain so that the inter-processor communication is minimized.

Current challenges in multigrid research lay predominantly in the development of highly scalable implementations that will perform efficiently on modern, and future, multicore architectures. In order to scale to many thousands of cores, for example, it is likely that new variants will need to be developed, perhaps relying on asynchronous communications and more efficient ways of incorporating corrections from the coarsest grids. The additional complexity of combining adaptivity and parallelism in an efficient manner is also a topic of much current research.

Acknowledgements

The author gratefully acknowledges the substantial contributions of his numerous research collaborators in the development and application of multigrid methods. Specific collaborative research cited in this chapter has been in collaboration with P.H. Gaskell, Y.-Y. Koh, A.M. Mullis, J. Rosam and H.M. Thompson. Particular thanks to Y.-Y. Koh and J. Rosam for giving permission for selected Figures to be reproduced from [47] and [63] respectively.

References

- [1] M.F. Adams. Parallel multigrid solvers for 3D unstructured finite element problems in large deformation elasticity and plasticity. *Int. J. Numer. Meth. Engrg.* 48:1241–1262, 2000.
- [2] M.F. Adams. Evaluation of three unstructured multigrid methods on 3D finite element problems in solid mechanics. *Int. J. Numer. Meth. Engrg.* 55:519–534, 2002.
- [3] T. Apel, S. Grosman, P.K. Jimack and A.Meyer. A new methodology for anisotropic mesh refinement based upon error gradients. *Appl. Numer. Math.* 50:329–341, 2004.
- [4] D.A. Aruliah and U.M. Ascher. Multigrid preconditioning for Krylov methods for time-harmonic Maxwell’s equations in three dimensions. *SIAM J. Sci. Comput.* 24:702–718, 2002.
- [5] R.E. Bank and P.K. Jimack. A new parallel domain decomposition method for the adaptive finite element solution of elliptic partial differential equations. *Concurrency and Computation: Practice and Experience* 13:327–350, 2001.

- [6] R. Becker and M. Braack. Multigrid techniques for finite elements on locally refined meshes. *Num. Lin. Alg. Applic.* 7:363–379, 2000.
- [7] P.B. Bochev, C.J. Garasi, J.J. Hu, A.C. Robinson and R.S. Tuminaro. An improved algebraic multigrid method for solving Maxwell’s equations. *SIAM J. Sci. Comput.* 25:623–642, 2003.
- [8] M. Bollhofer, M.J. Grote and O. Schenk. Algebraic multilevel preconditioner for the Helmholtz equation in heterogeneous media. *SIAM J. Sci. Comput.* 31:3781–3805, 2009.
- [9] J. Boyle, M. Mihajlovic and J. Scott. HSL MI20: An efficient AMG preconditioner for finite element problems in 3D. *Int. J. Numer. Meth. Engrg.* 82:64–98, 2010.
- [10] M. Brezina, C. Tong and R. Becker. Parallel algebraic multigrids for structural mechanics. *SIAM J. Sci. Comput.* 27:1534–1554, 2006.
- [11] D. Braess. Towards algebraic multigrid for elliptic problems of second order. *Computing* 55:379–393, 1995.
- [12] A. Brandt. Multi-level adaptive solutions to boundary-value problems. *Math. Comp.* 31:333–390, 1977.
- [13] S. Brenner. A non-conforming mixed multigrid method for the pure displacement problem in planar linear elasticity. *SIAM J. Numer. Anal.* 30:116–135, 1993.
- [14] P.N. Brown, R.D. Falgout and J.E. Jones. Semicoarsening multigrid on distributed memory machines. *SIAM J. Sci. Comput.* 21:1823–1834, 2000.
- [15] Z.Q. Cai, T.A. Manteuffel, S.F. McCormick and S.V. Parter. *SIAM J. Numer. Anal.* 35:320–335, 1998.
- [16] Z.Q. Cai and G. Starke. Least-squares methods for linear elasticity. *SIAM J. Numer. Anal.* 42:826–842, 2004.
- [17] H. Damanik, J. Hron, A. Ouazzi and S. Turek. A monolithic FEM-multigrid solver for non-isothermal incompressible flow on general meshes. *J. Comput. Phys.* 228:3869–3881, 2009.
- [18] W.G. Davids and G.M. Turkiyyah. Multigrid preconditioner for unstructured nonlinear 3D FE models. *J. Engrg. Mech. - ASCE* 125:186–196, 1999.
- [19] A.T. Degani and G.C. Fox. Parallel multigrid computation of the unsteady incompressible Navier-Stokes equations. *J. Comput. Phys.* 128:223–236, 1996.
- [20] J.E. Dendy. Black box multigrid. *J. Comput. Phys.* 48:366–386, 1982.
- [21] F. Durst and M. Schafer. A parallel block-structured multigrid method for the prediction of incompressible flows. *Int. J. Numer. Meth. Fluids* 22:549–565, 1996.
- [22] H.C. Elman, D.J. Silvester and A.J. Wathen. *Finite Elements and Fast Iterative Solvers with applications in incompressible fluid dynamics*, OUP, 2005.
- [23] B. Epstein, T. Rubin and S. Seror. Accurate multiblock Navier-Stokes solver for complex aerodynamic configurations. *AIAA Journal* 41:582–594, 2003.
- [24] Y.A. Erlangga and R. Nabben. On a multilevel Krylov method for the Helmholtz equation preconditioned by shifted Laplacian. *Elec. Trans. Numer. Anal.* 31:403–424, 2008.

- [25] Y. Erlangga, C. Oosterlee and C. Vuik. A novel multigrid based preconditioner for heterogeneous Helmholtz problems. *SIAM J. Sci. Comput.* 27:1471–1492, 2006.
- [26] L.A. Freitag and C. Ollivier Gooch. Tetrahedral mesh improvement using swapping and smoothing. *Int. J. Numer. Meth. Engrg.* 40:3979–4002, 1997.
- [27] P.H. Gaskell, P.K. Jimack, Y.-Y. Koh and H.M. Thompson. Development and application of a parallel multigrid solver for the simulation of spreading droplets. *Int. J. Num. Meth. Fluids* 56:979–1002, 2008.
- [28] C.E. Goodyer and M. Berzins. Parallelisation and scalability issues of a multi-level elasto-hydrodynamic lubrication solver. *Concurrency and Comp.: Practice and Exp.* 19:369–396, 2007.
- [29] M. Griebel, D. Oeltz and M.A. Schweitzer. An algebraic multigrid method for linear elasticity *SIAM J. Sci. Comput.* 25:385–407, 2003.
- [30] G. Haase, M. Kuhn and U. Langer. Parallel multigrid 3D Maxwell solvers. *Parallel Comput.* 27:761–775, 2001.
- [31] F.E. Ham, F.S. Lien and A.B. Strong. A Cartesian grid method with transient anisotropic adaption. *J. Comput. Phys.* 179:469–494, 2002.
- [32] L. Hart, S.F. McCormick and A. O’Gallagher. The fast adaptive composite-grid method (FAC): Algorithms for advanced computers. *Appl. Math. Comp.* 19:103–125, 1986.
- [33] M. Heil. An efficient solver for the fully coupled solution of large-displacement fluid-structure interaction problems. *Comput. Meth. Appl. Mech. Engrg.* 193:1–23, 2004.
- [34] V.E. Henson, W.L. Briggs and S.F. McCormick. *A Multigrid Tutorial*, SIAM, 2000.
- [35] V.E. Henson and U.M. Yang. BoomerAMG: A parallel algebraic multigrid solver and preconditioner. *Appl. Numer. Math.* 41:155–177, 2002.
- [36] J.J. Heys, T.A. Manteuffel, S.F. McCormick and J.W. Ruge. First-order system least squares (FOSLS) for coupled fluid-elastic problems *J. Comput. Phys.* 195:560–575, 2004.
- [37] R. Hiptmair. Multigrid method for Maxwell’s equations. *SIAM J. Numer. Anal.* 36:204–225, 1998.
- [38] D.C. Hodgson and P.K. Jimack. Efficient parallel generation of partitioned, unstructured meshes. *Advances in Engineering Software* 27:59–70, 1996.
- [39] D.C. Hodgson and P.K. Jimack. A domain decomposition preconditioner for a parallel finite element solver on distributed unstructured grids. *Parallel Computing* 23:1157–1181, 1997.
- [40] M.J. Holst and F. Saied. Numerical solution of the nonlinear Poisson-Boltzmann equation - developing more robust and efficient methods. *J. Comput. Chem.* 16:337–364, 1995.
- [41] J.J. Hu, R.S. Tuminaro, P.B. Bochev, C.J. Garasi and A.C. Robinson. Toward an h-independent algebraic multigrid method for Maxwell’s equations. *SIAM J. Sci. Comput.* 27:1669–1688, 2006.
- [42] J.-H. Jeong, N. Goldenfeld and J.A. Dantzig. Phase field model for three-

- dimensional dendritic growth with fluid flow. *Phys. Rev. E* 64:041692, 2001.
- [43] A.C. Jones and P.K. Jimack. An adaptive multigrid tool for elliptic and parabolic systems. *Int. J. Num. Meth. Fluids* 47:1123–1128, 2005.
- [44] A. Karma. Phase-field formulation for quantitative modelling of alloy solidification. *Phys. Rev. Lett.* 87:115701, 2001.
- [45] A. Karma and E.-J. Rappel. Quantitative phase-field modeling of dendritic growth in two and three dimensions. *Phys. Rev. E* 57:4323–4349, 1998.
- [46] S. Kim and S. Kim. Multigrid simulation or high-frequency solutions of the Helmholtz problem in heterogeneous media. *SIAM J. Sci. Comput.* 24:684–701, 2002.
- [47] Y.-Y. Koh. *Efficient Numerical Solutions of Droplet Spreading Flows*, PhD Thesis, University of Leeds, UK, 2007.
- [48] Y.-Y. Koh, Y.C. Lee, P.H. Gaskell, P.K. Jimack and H.M. Thompson. Droplet migration: quantitative comparisons with experiment. *Eur. Phys. J.* 166:117–120, 2009.
- [49] A. Krechel and K. Stuben. Parallel algebraic multigrid based on subdomain blocking. *Parallel Computing* 27:1009–1031, 2001.
- [50] C.O. Lee. Multigrid methods for the pure traction problem of linear elasticity: Mixed formulation. *SIAM J. Numer. Anal.* 35:121–145, 1998.
- [51] X.Y.S. Li. An overview of SuperLU: Algorithms, implementation, and user interface. *ACM Trans. on Math. Software* 31:302–325, 2005.
- [52] C. Liu, Z. Liu and S. McCormick. Multilevel adaptive methods for incompressible-flow in grooved channels. *J. Comput. Appl. Math.* 38: 283–295, 1991.
- [53] T.A. Manteuffel, S.F. McCormick, J.G. Schmidt and C.R. Westphal. First-order system least squares for geometrically nonlinear elasticity. *SIAM J. Numer. Anal.* 44:2057–2081, 2006.
- [54] S. Mijalkovic and M.D. Mihajlovic. Component-wise algebraic multigrid preconditioning for the iterative solution of stress analysis problems from microfabrication technology. *Comm. Numer. Meth. Engrg.* 17:737–747, 2001.
- [55] M.D. Mihajlovic and S. Mijalkovic. A component decomposition preconditioning for 3D stress analysis problems. *Numer. Lin. Alg. Applic.* 9:567–583, 2002.
- [56] R.S. Montero, I.M. Llorente and M.D. Salas. Robust multigrid algorithms for the Navier-Stokes equations. *J. Comput. Phys.* 173:412–432, 2001.
- [57] J.C. Nedelec. A new family of mixed finite elements in R^3 . *Numer. Math.* 50:57–81, 1986.
- [58] L.N. Olson and J.B. Schroder. Smoothed aggregation for Helmholtz problems. *Numer. Lin. Alg. Applic.* 17:361–386, 2010.
- [59] J.E. Pasciak and Y. Wang. A multigrid preconditioner for the mixed formulation of linear plane elasticity. *SIAM J. Numer. Anal.* 44:478–493, 2006.
- [60] N. Provatas, N. Goldenfeld and J. Dantzig. Adaptive mesh refinement computation of solidification microstructures using dynamic data structures. *J. Comput. Phys.* 148:265–290. 1999.
- [61] J.C. Ramirez, C. Beckermann, A. Karma and H.J. Diepers. Phase-field modelling

- of binary alloy solidification with coupled heat and solute diffusion. *Phys. Rev. E* 69:051607, 2004.
- [62] S. Reitzinger and J. Schoberl. An algebraic multigrid method for finite element discretizations with edge elements. *Numer. Lin. Alg. Applic.* 9:223–238, 2002.
- [63] J. Rosam. *A Fully Implicit, Fully Adaptive Multigrid Method for Multiscale Phase-Field Modelling* PhD Thesis, University of Leeds, UK, 2007.
- [64] J. Rosam, P.K. Jimack and A.M. Mullis. A fully implicit, fully adaptive time and space discretisation method for phase-field simulation of binary alloy solidification. *J. Comput. Phys.* 225:1271–1287, 2007.
- [65] J. Rosam, P.K. Jimack and A.M. Mullis. An adaptive, fully implicit, multigrid phase-field model for quantitative simulation of non-isothermal binary alloy solidification. *Acta Materialia* 56:4559–4569, 2008.
- [66] J. Rosam, P.K. Jimack and A.M. Mullis. Quantitative phase-field modelling of solidification at high Lewis number. *Phys. Rev. E* 79:030601(R), 2009.
- [67] Y. Saad and M.H. Schultz. GMRES - A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.* 7:856–869, 1986.
- [68] J. Schöberl. NETGEN - An advancing front 2D/3D-mesh generator based on abstract rules. *Comput. Visual. Sci.* 1:41–52, 1997.
- [69] L.W. Schwartz. Hysteretic effects in droplet motion on heterogeneous substrates: direct numerical simulation. *Langmuir* 14:3440–3453, 1998.
- [70] M. Sellier, P.H. Gaskell, P.K. Jimack and H.M. Thompson. Efficient and accurate time-adaptive multigrid simulations of droplet spreading. *Int. J. Numer. Meth. Fluids* 45:1161–1186, 2004.
- [71] K. Shahbazi, D.J. Mavriplis and N.K. Burgess. Multigrid algorithms for high-order discontinuous Galerkin discretizations of the compressible Navier-Stokes equations *J. Comput. Phys.* 228:7917–7940, 2009.
- [72] V.M. Starov, V.V. Kalinin and J.D. Chen. Spreading of liquid drops over dry surfaces. *Adv. Coll. Interface Sci.* 50:187–221, 1994.
- [73] P.S. Summant, A.C. Cangellaris and N.R. Aluru. A node-based agglomeration AMG solver for linear elasticity in thin bodies. *Comm. Numer. Meth. Engrg.* 25:219–236, 2009.
- [74] C.H. Tai, Y. Zhao and K.M. Liew. Parallel computation of unsteady three-dimensional incompressible viscous flow using an unstructured multigrid method. *Computers and Structures* 28: 2425–2436, 2004.
- [75] J.L. Thomas, B. Diskin and A. Brandt. Textbook multigrid efficiency for fluid simulations. *Ann. Rev. Fluid Mech.* 35:317–340, 2003.
- [76] U. Trottenberg, C.W. Oosterlee and A. Schüller. *Multigrid*, Academic Press, 2000.
- [77] P. Wesseling. *Introduction to Multigrid Methods*, Wiley, 1992.
- [78] B.I. Wohlmuth and R.H. Krause. Monotone multigrid methods on nonmatching grids for nonlinear multibody contact problems. *SIAM J. Sci. Comput.* 25:324–347, 2003.