Title: Multigrid Methods

Name: Luke Olson[1]

Affil./Addr.: Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, IL 61801
email: lukeo@illinois.edu
url: http://www.cs.uiuc.edu/homes/lukeo/

Multigrid Methods

Synonyms

Geometric Multigrid, MG, GMG

Definition

Multigrid (MG) methods are used to approximate solutions to elliptic partial differential equations (PDEs) by iteratively improving the solution through a sequence of coarser discretizations or grids. The methodology has been developed and extended since the 1970's to also target more general PDEs and systems of algebraic equations. A typical approach consists of a series of refinements or grids, where an approximate solution is iteratively improved through a combination relaxation — e.g. Gauss-Seidel — and defect corrections — e.g. using projections to coarser, smaller grids.

Overview

Multigrid methods were formalized by the late 1970's in the works of Brandt [4; 3] and Hackbusch [11], but were also studied earlier by Fedorenko [9; 10]. Over the next decade, multigrid development focused on, among other directions, the design and analysis of different relaxation techniques, the construction of coarse discretizations, and the theory of a framework toward a more robust *geometric* multigrid framework — e.g., see McCormick [12]. Through this early development, operator-based strategies and an *algebraic* approach to multigrid emerged, which culminated in the work of Ruge and Stüben [13]. More recently, multigrid methods have grown in popularity and in robustness, being used in a vast number of areas of science and on a variety of computing architectures. Several texts on the subject give a more complete historical overview and description [5; 15].

Since there are many ways to set up a multigrid approach and each with a number of setup decisions and tunable parameters in each method, multigrid is best viewed as a framework rather than a specific method. Here, we present a representative approach based in the context of a matrix problem resulting from a discretization of an elliptic PDE. An alternative approach to presenting a *geometric* multigrid method is to formulate of the problem in a weak context at each grid level — e.g., a finite element formulation. Likewise, an entirely *algebraic* approach may be taken wherein only the matrix A is considered — e.g., recent versions of the *algebraic* multigrid.

Terminology

The goal is to solve a matrix problem

$$A^h \mathbf{u}^h = f^h \tag{1}$$

associated with grid Ω^h. In the following we construct a sequence of symmetric, positive-definite (matrix) problems, $A^h \mathbf{u}^h = \mathbf{f}^h$, associated with grid Ω^h (see Figure 1). We assume that grids Ω^h, with $h = h_0 < h_1 < \cdots < h_m$, are nested — i.e., $\Omega^{h_{k+1}} \subset \Omega^{h_k}$. A grid spacing of $h = h_0$ is referred to as the *fine* grid, while the coarsest grid is represented with $h = h_m$. In addition, when considering only two grids, h and H are used to simplify notation for fine and coarse grids.

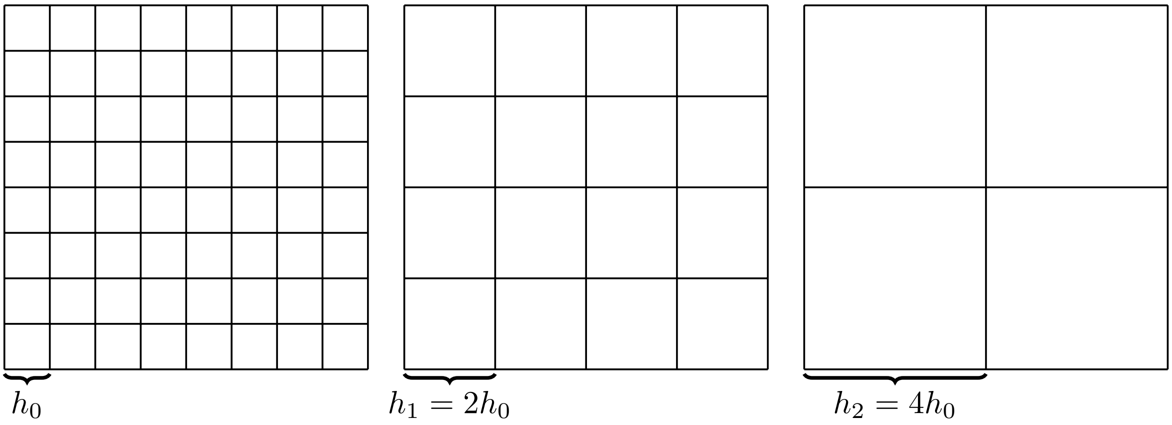


Fig. 1: Hierarchy of grids with spacing h, $2h$, and $4h$.

Central to the multigrid process is the ability to adequately represent certain grid functions $\mathbf{u}^h \in \Omega^h$ on a coarser grid, Ω^H. We denote the *restriction* operator as $R_h^H : \Omega^h \to \Omega^H$ and the *prolongation* or interpolation operator as $P_H^h : \Omega^H \to \Omega^h$, both of which are assumed to be full rank.

In the following, the standard Euclidean and energy norms are denoted $\| \cdot \|$ and $\| \cdot \|_A$, with respective inner products $\langle \cdot, \cdot \rangle$ and $\langle \cdot, \cdot \rangle_A$. For an initial guess, $\mathbf{u}_0^h$, the objective is to construct and multilevel iterative process that reduces the energy norm of the error. This is accomplished by exposing the error in $\mathbf{u}_0^h$ as oscillatory error on different grid levels. A useful observation is that the error $\mathbf{e}_0^h = \mathbf{u}_0^h - \mathbf{u}_*^h$, satisfies the *error equation* $A^h \mathbf{e}_0^h = \mathbf{r}_0^h$, where $\mathbf{r}_0^h$ is the residual, $\mathbf{r}_0^h = \mathbf{f}^h - A^h \mathbf{u}_0^h$.

Basic Methodology

Consider the elliptic partial differential equation

$$-u_{xx} = f(x), \tag{2}$$

with zero boundary conditions on the unit interval. Using second-order finite differences on $\Omega^h = \{x_i^h\}$ with nodes $x_i^h = ih$, where $h = 1/(n+1)$ and $i = 0, \ldots, n+1$, results in the matrix problem

$$\underbrace{\frac{1}{h^2}\begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & & \ddots & & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{bmatrix}}_{A^h} \underbrace{\begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{n-1} \\ u_n \end{bmatrix}}_{\mathbf{u}^h} = \underbrace{\begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_{n-1} \\ f_n \end{bmatrix}}_{\mathbf{f}^h}. \tag{3}$$

Given an initial guess $\mathbf{u}_0^h$ to the solution $\mathbf{u}_*^h$, a stationary iterative method computes an update of the form

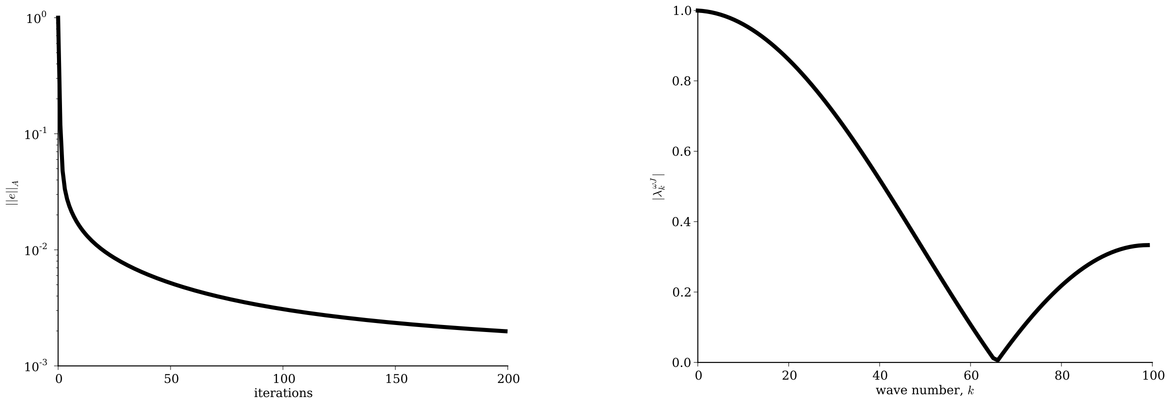
$$\mathbf{u}_1^h = \mathbf{u}_0^h + M^{-1}(\mathbf{f}^h - A^h\mathbf{u}_0^h) = \mathbf{u}_0^h + M^{-1}\mathbf{r}_0^h. \tag{4}$$

Notice that if $M = A$, then the iteration is exact. The error in a stationary iteration (4) satisfies

$$\mathbf{e}_1^h = \left(I - M^{-1}A^h\right)\mathbf{e}_0^h = G\mathbf{e}_0^h, \tag{5}$$

which implies that a sufficient condition on the *error propagation* matrix for this problem, G, is that $\rho(G) < 1$. For a symmetric, positive-definite M-matrix — i.e., weakly diagonally dominant with positive diagonals and negative off-diagonals — a common stationary method is weighted Jacobi with $M = (1/\omega)D$ for some weight $\omega \in (0, 1)$ and with D as the diagonal of A. As an example, consider (3) with $\omega = 2/3$ and $h = 0.01$, a random initial guess, and $\mathbf{f}^h \equiv 0$. As shown in Figure 2a, weighted Jacobi is very effective at reducing the error for the first few iterations, but quickly stagnates.

The ability of a *relaxation* or *smoothing* method, such as weighted Jacobi, to rapidly reduce the error in the first few iterations is central to a multigrid method. To see this, we note that the eigenvectors of A^h are Fourier modes, and the eigenvalue-eigenvector pairs $(\lambda, \mathbf{v})$ are



(a) Error history.

(b) Asymptotic convergence factors.

Fig. 2: Energy norm of the error and convergence factors in a weighted Jacobi iteration.

$$\lambda_k = 4\sin^2\left(\frac{\pi}{2n}\cdot k\right) \quad \mathbf{v}_{k,i} = \sin\left(\frac{j\pi}{2n}\cdot k\right) \quad \text{for } k = 1,\ldots,n. \tag{6}$$

Correspondingly, the eigenvalue-eigenvectors pairs $(\lambda^{\omega J}, \mathbf{v}^{\omega J})$ of the weighted Jacobi iteration matrix, G in (5) become

$$\lambda^{\omega J} = 1 - \frac{\omega}{2}\lambda \quad \mathbf{v}^{\omega J} = \mathbf{v}. \tag{7}$$

Thus, eigenvalues of the weighted Jacobi iteration matrix that approach 1.0 (thus leading to stagnation) correspond to low k and are associated with Fourier modes that are smooth. Consequently, weighted Jacobi is effective for highly oscillatory error — i.e., error with large energy norm — and is ineffective for smooth error — i.e., error that corresponds to low Fourier modes. This is depicted in Figure 2b where the weighted Jacobi convergence factor is shown for each Fourier wavenumber.

Prior to relaxation, the error $\mathbf{e}_0^h$, is likely to have representation of both low and high frequency Fourier modes. After relaxation, the high frequency modes no longer dominate and the remaining error is largely comprised of low frequency Fourier modes. To eliminate these smooth errors, a multigrid method constructs a coarse-grid *correction* step as part of the iteration. That is, consider k-steps of a weighted Jacobi relaxation method:

$$\mathbf{u}_k^h \leftarrow \mathbf{u}_{k-1}^h + \omega D^{-1}\mathbf{r}_{k-1}^h = \mathcal{G}(\mathbf{u}^h_{,}\mathbf{f}^h, k). \tag{8}$$

Since $\mathbf{e}_k^h$ is expected to be smooth, it can be represented with a coarser vector $\mathbf{e}_k^H$ and reconstructed through low-order (linear) interpolation. For example, halving the fine-grid problem results in a coarse grid Ω^H with $H = h/2$ and $n_c = (n+1)/2$ coarse points. Then, we define an interpolation operator $P_H^h : \mathbb{R}^{(n+1)/2} \to \mathbb{R}^n$ using linear interpolation,

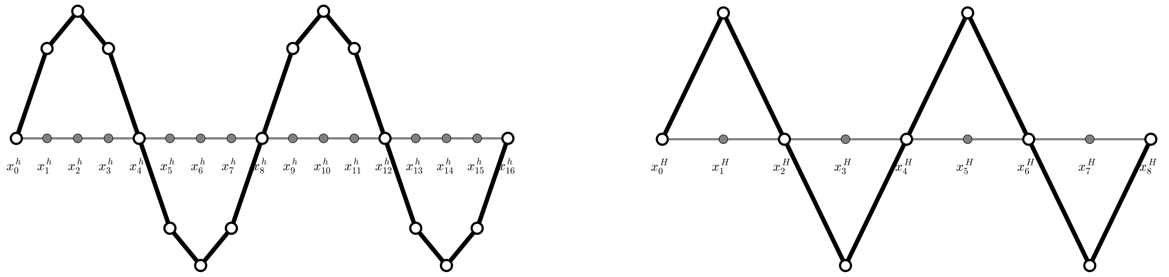
$$P_H^h = \frac{1}{2}\begin{bmatrix} 1\,2\,1 & & & & \\ & 1\,2\;\,1 & & & \\ & & \ddots & & \\ & & & 1\;\,2\,1 & \\ & & & & 1\,2\,1 \end{bmatrix}^T, \tag{9}$$

and restriction given by $R_h^H = (P_H^h)^T$. Then the two-level multigrid algorithm is given in Algorithm 1.

Algorithm 1: Two-level Multigrid

```
$\mathbf{u}^h = \mathcal{G}(\mathbf{u}_0^h, \mathbf{f}^h, k_{\text{pre}})$      {relax $k_{\text{pre}}$ times on the fine grid, $\Omega^h$}
$\mathbf{r}^h = \mathbf{f}^h - A^h\mathbf{u}^h$      {form residual}
$\mathbf{r}^H = R_h^H\mathbf{r}^h$      {restrict residual to coarse grid $\Omega^H$}
$\mathbf{e}^H = (A^H)^{-1}\mathbf{r}^H$      {solve the coarse-grid error problem}
$\mathbf{e}^h = P_H^h\mathbf{e}^H$      {interpolate coarse error approximation}
$\bar{\mathbf{u}}^h = \mathbf{u}^h + \mathbf{e}^h$      {correct the (relaxed) solution}
$\mathbf{u}_1^h = \mathcal{G}(\bar{\mathbf{u}}^h, \mathbf{f}^h, k_{\text{post}})$      {relax $k_{\text{post}}$ times on the fine grid, $\Omega^h$}
```

A mulilevel algorithm follows by observing the effect of restricing a low Fourier mode to a coarser grid. For example, consider the case of a fine grid with $n = 15$, which results in a coarse grid of $n = 7$. A lower Fourier mode with wave number $k = 5$ (see (6)) results in high Fourier mode on the coarse grid if sampled at every other point. That is, a mode that is slow to converge with relaxation on the fine grid is more effectively reduced when restricted to a coarse grid. This is illustrated in Figure 3. In this particular example, the convergence factor of the mode on the fine grid is 0.8 while the convergence factor of the same mode on the coarse grid is 0.3.



(a) Fine grid.

(b) Coarse grid.

Fig. 3: Mode $k = 5$ on a fine grid ($n = 15$) and coarse grid ($n = 7$).

With this observation we arrive at a multilevel variant of Algorithm 1, where the coarse-level solve is replaced with relaxation, thereby postponing the inversion of a coarse matrix to the coarsest grid level. The process is shown in Figure 4.

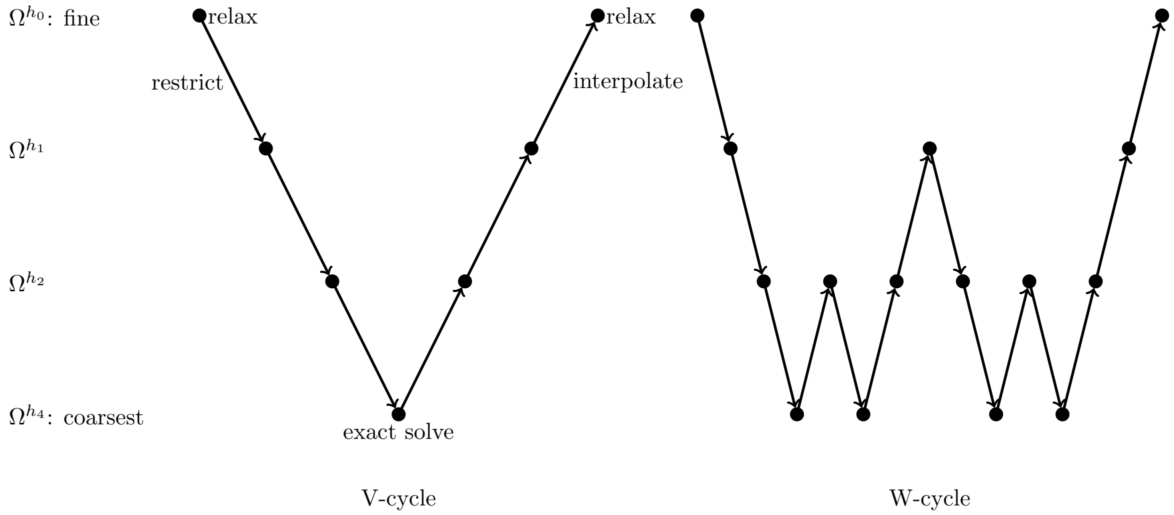


Fig. 4: V and W multigrid cycling. The down and up arrows represent restriction of the residual and interpolation of the error between grids. A circle (●) represents relaxation.

Higher dimensions

The mechanics of the algorithm extend directly to higher dimensions. In particular, if the matrix problems $A^h\mathbf{u}^h = \mathbf{f}^h$ are defined on a sequence of grids where even indexed grid points become coarse grid points in each coordinate direction — for example, as shown in Figure 1 — then the 1D definition of linear interpolation extends through tensor definitions. That is, the 2D form for bilinear and the 3D form for trilinear interpolation are define as

$$P_H^h = P \otimes P \quad \text{and} \quad P_H^h = P \otimes P \otimes P, \tag{10}$$

respectively, where P is 1D linear interpolation as defined by (9).

Theoretical Observations and Extensions

The multigrid process defined by Algorithm 1 immediately yields several theoretical conclusions. In turn, these theoretical observations lead to extensions to the basic form of geometric multigrid and ultimately to a more *algebraic* form of the method, where the rigid assumptions on grid structure and interpolation definitions are relieved and made more general. To this end, we consider the operator form of the error propagation in the multigrid cycle. Following Algorithm 1 for an initial guess $\mathbf{u}_0^h$, we arrive at the following operation on the error (using G as pre/post relaxation as in Algorithm 1):

$$E = G\left(I - P_H^h(A^H)^{-1}R_h^H A^h\right)G = GTG, \tag{11}$$

where we T is called two-grid correction matrix. From right to left, we see that relaxation, forming the residual (with $A^h\mathbf{e}_0^h$), restriction, the coarse-solve, interpolation, corrections, and additional relaxation are all represented in the operator. If $R_h^H = c(P_H^h)^T$, for some constant c, then T simplifies to

$$T = I - cP(A^H)^{-1}RA^h, \tag{12}$$

where we have dropped the sub and superscripts on P. Notice, that if

$$A^H = RA^hP, \tag{13}$$

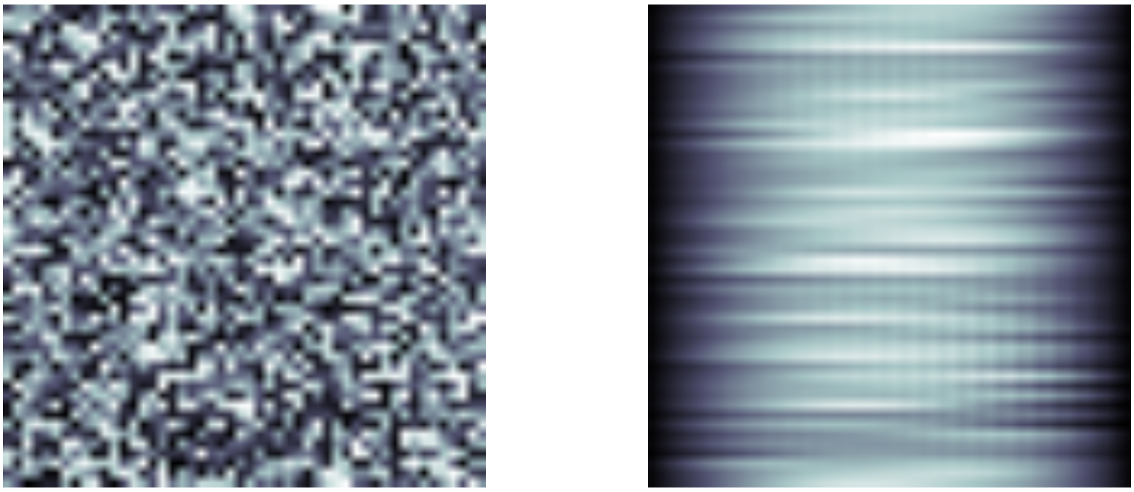
then T is an A-orthogonal projection and, importantly, $I - T$ is the A-orthogonal projection onto the range of P, interpolation. This form of the coarse grid operator A^H is the *Galerkin* form, which also follows from a variational formulation of multigrid. It suggests that the coarse grid operator can be constructed solely from A^h using P. Moreover, the form of T yields an important theoretical property: *if* $G\mathbf{e}_0^h \in \mathcal{R}(P)$*, the range of* P*, then the V-cycle is exact.* This highlights the complementary nature of relaxation and coarse-grid correction in the multigrid process, and has been used as the basis for the design of new methods and the development of new multigrid theory over the last several decades. Indeed, if an efficient relaxation process can be defined and a sparse interpolation operator can be constructed so that error not eliminated by relaxation is accurately represented through interpolation, then the multigrid cycle will be highly accurate and efficient.

Beyond basic multigrid

If error components not reduced by relaxation are not geometrically smooth, as motivated in the previous sections with Fourier modes, then coarse-grid correction based on uniform coarsening and linear interpolation may not adequately complement the relaxation process. As an example, consider the 2D model problem on a unit square with anisotropy:

$$-u_{xx} - \varepsilon u_{yy} = f(x, y). \tag{14}$$

Figure 5 depicts an oscillatory error before and after 100 weighted Jacobi iterations for the case of $\varepsilon = 0.001$. Notice that contrary to the isotropic example, where the error after relaxation is well represented by the lowest Fourier mode and is smooth in every direction, in this example the error is not geometrically smooth in the y-direction.



(a) Initial error.

(b) Error after 100 iterations.

Fig. 5: The effect of relaxation for anisotropic problems.

Since the error is geometrically smooth in the x-direction, one approach is to coarsen only in the x-direction, which is called *semi*-coarsening. Likewise, relaxation could be modified to perform block relaxation sweeps using y-slices in the domain, while still using uniform coarsening. Both methods work well for anisotropy aligned in the coordinate direction, yet the effect is limited for more complicated scenarios — e.g., rotated anisotropy.

As an alternative, the practitioner could develop an improved interpolation operator to directly target error components not reduced by relaxation. This approach is called *operator induced* interpolation and points toward a more algebraic approach to constructing more robust multigrid methods. In algebraic multigrid, the relaxation method is fixed, while coarse grids and interpolation operators are automatically constructed in order to define a complementary coarse-grid correction.

Advantages and Limitations of Geometric Multigrid

While traditional forms of geometric-based multigrid are limited to problems with structure and problems that have a strong geometric association, there are a number of notable advantages of this methodology in contrast to more general, robust multigrid methods. For one, structured problems often admit a stencil-based approach in defining operators such as A^h and P_H^h. This often results in lower storage, less communication in a parallel setting, and increased locality. Furthermore, setup costs for geometric multigrid, particularly if the stencils are known *a priori*, can be much less than in algebraic methods. As a result, if a problem is inherently structured, then geometric multigrid is a clear advantage if appropriate relaxation methods can be formed.

There are several packages that implement geometric multigrid methods at scale. The parallel semicoarsening multigrid solvers SMG [14; 6] and PFMG [1] are both implemented in the *hypre* package [8]. Both offer stencil-based multigrid solvers for semi-structured problems, with SMG leaning toward robustness and PFMG toward efficiency [7]. Other methods such as hierarchical hybrid grids (HHG) [2] explicitly build structure into the problem in order to take advantage of the efficiencies in geometric multigrid.

The limitation of a purely geometric approach to multigrid is squarely in the direction of robustness. Graph and data problems, as well as unstructured mesh problems, do not have a natural structure for which to build a hierarchy of grids. Even more, for many complex physics applications that are structured, the design of an effective relaxation process with a grid hierarchy is often elusive. On the other hand, the push toward more algebraic theory and design is also contributing to the development of more robust geometric approaches in order to take advantage of its efficiencies.

Cross-references

iterative methods, Krylov methods, preconditioning, algebraic multigrid, domain decomposition

References

1. Ashby SF, Falgout RD (1996) A parallel multigrid preconditioned conjugate gradient algorithm for groundwater flow simulations. Nuclear Science and Engineering 124:145–159
2. Bergen B, Gradl T, Rude U, Hulsemann F (2006) A massively parallel multigrid method for finite elements. Computing in science & engineering 8(6):56–62
3. Brandt A (1973) Multi-level adaptive technique (MLAT) for fast numerical solution to boundary value problems. In: Cabannes H, Temam R (eds) Proceedings of the Third International Conference on Numerical Methods in Fluid Mechanics, Lecture Notes in Physics, vol 18, Springer Berlin Heidelberg, pp 82–89, DOI 10.1007/BFb0118663, URL http://dx.doi.org/10.1007/BFb0118663
4. Brandt A (1977) Multi-level adaptive solutions to boundary-value problems. Math Comp 31(138):333–390
5. Briggs WL, McCormick SF, et al (2000) A multigrid tutorial, vol 72. Siam
6. Brown PN, Falgout RD, Jones JE (2000) Semicoarsening multigrid on distributed memory machines. SIAM Journal on Scientific Computing 21(5):1823–1834
7. Falgout RD, Jones JE (2000) Multigrid on massively parallel architectures. In: Dick E, Riemslagh K, Vierendeels J (eds) Multigrid Methods VI, Lecture Notes in Computational Science and Engineering, vol 14, Springer Berlin Heidelberg, pp 101–107
8. Falgout RD, Yang UM (2002) hypre: A library of high performance preconditioners. In: Computational ScienceICCS 2002, Springer, pp 632–641
9. Fedorenko R (1962) A relaxation method for solving elliptic difference equations. {USSR} Computational Mathematics and Mathematical Physics 1(4):1092–1096, DOI 10.1016/0041-5553(62)90031-9
10. Fedorenko R (1964) The speed of convergence of one iterative process. {USSR} Computational Mathematics and Mathematical Physics 4(3):227 – 235
11. Hackbusch W (1976) Ein iteratives verfahren zur schnellen auflösung elliptischer randwertprobleme. Tech. Rep. 76-12, Institute for Applied Mathematics, University of Cologne
12. McCormick S (1985) Multigrid methods for variational problems: general theory for the v-cycle. SIAM journal on numerical analysis 22(4):634–643
13. Ruge JW, Stüben K (1987) Algebraic multigrid. In: Multigrid methods, Frontiers Appl. Math., vol 3, SIAM, Philadelphia, PA, pp 73–130

14. Schaffer S (1998) A semicoarsening multigrid method for elliptic partial differential equations with highly discontinuous and anisotropic coefficients. SIAM Journal on Scientific Computing 20(1):228–242

15. Trottenberg U, Oosterlee CW, Schuller A (2000) Multigrid. Academic press