

Iterative Methods:
Multigrid Techniques
Lecture 7

1 Background

SLIDE 1

- Developed over the last 25 years — Brandt (1973) published first paper with practical results.
- Offers the possibility of solving a problem with work and storage proportional to the number of unknowns.
- Well developed for linear elliptic problems — application to other equations is still an active area of research.

Good Introductory Reference: *A Multigrid Tutorial*, W.L. Briggs, V.E. Henson, and S.F. McCormick, SIAM Monograph, 2000.

2 Basic Principles

2.1 Some ideas

SLIDE 2

We will use, once again, the one dimensional problem to illustrate the basic principles of multigrid methods. The ideas presented can be readily extended to multiple dimensions. First, however, we will review some simple facts about iterative methods seen in the last lecture and develop some ideas to accelerate the iterative process.

1. Multigrid is an iterative method \rightarrow a *good initial guess* will reduce the number of iterations:

to solve $A_h \mathbf{u}_h = \mathbf{f}_h$ by iteration, we could take

$$\mathbf{u}_h^0 \sim \mathbf{u}_{2h}, \quad \text{where } A_{2h} \mathbf{u}_{2h} = \mathbf{f}_{2h} \quad \dots$$

Here A_h , \mathbf{u}_h and \mathbf{f}_h denote the matrix, vector of unknowns, and force vector, respectively, that results from discretizing our model problem on a grid of size h . $\mathbf{u}_h^0 \sim \mathbf{u}_{2h}$ means that the initial guess for the iterative process on grid h is “approximated” from the solution of the problem on a grid of size $2h$. We use the word approximate because \mathbf{u}_h and \mathbf{u}_{2h} are vectors that have different lengths. We shall see later how this approximation is carried out. It is clear that the procedure outlined is recursive; that is, the problem on the grid $2h$ can also be solved by iteration, with an initial guess provided by solving $A_{4h} \mathbf{u}_{4h} = \mathbf{f}_{4h}$, etc We point out that iteration on coarser meshes is cheaper because n is smaller, and therefore

there is less work per iteration; and because fewer iterations are required, i.e. n is smaller.

but ... the number of iterations needed to solve $A_h \mathbf{u}_h = \mathbf{f}_h$ still $O(n^2)$.

$$h = \frac{1}{n+1}$$

Since some smooth components of the error will still remain.

SLIDE 3

2. If after a few iterations, the error is smooth, we could solve for the error on a coarser mesh, e.g. $A_{2h} \mathbf{e}_{2h} = \mathbf{r}_{2h}$.

A good idea because:

- Smooth functions can be represented on coarser grids;
- Coarse grid solutions are cheaper.

This idea is in fact the central idea of multigrid techniques. In order to turn this idea into a practical algorithm, several ingredients will be required.

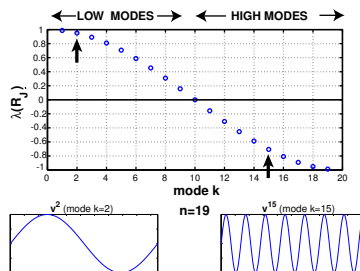
2.2 Smoother

SLIDE 4

If the high frequency components of the error decay faster than the low frequency components, we say that the iterative method is a smoother.

2.2.1 Jacobi

SLIDE 5



Is Jacobi a smoother?

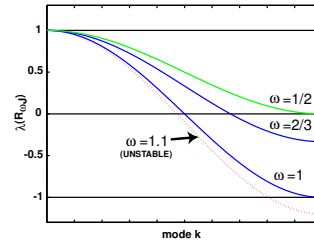
... → NO

We see that $\rho(R_J) = |\lambda^n(R_J)| = |\lambda^1(R_J)|$ and, since n is the highest frequency mode, it is clear that Jacobi is not a smoother.

2.2.2 Under-Relaxed Jacobi

SLIDE 6

$$R_{\omega J} = \omega R_J + (1 - \omega) I$$



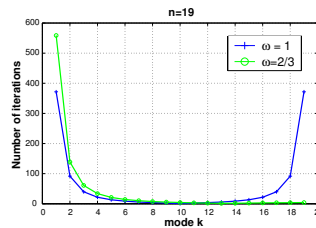
$$\lambda^k(R_{\omega J}) = \omega \lambda^k(R_J) + (1 - \omega) = 1 - \omega(1 - \lambda^k(R_J)) ,$$

$$k = 1, \dots, n$$

We observe that for $\omega < 1$, Jacobi can in fact be a good smoother. If we set the condition $|\lambda^{(n+1)/2}(R_{\omega J})| = |\lambda^n(R_{\omega J})|$, we obtain $\omega = 2/3$. We also note that for $\omega > 1$ the method becomes unstable (does not converge) since for some k , $|\lambda^k(R_{\omega J})| > 1$. In some sense, the price to be paid for Jacobi to be a good smoother is a slow down in convergence of the low frequency modes.

SLIDE 7

Iterations required to reduce an error mode by a factor of 100

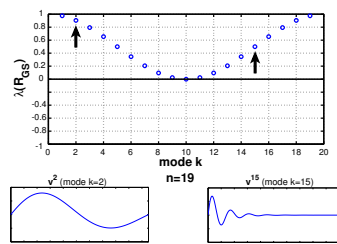


The graph shows the number of iterations required to reduce the the amplitude of each error mode by a factor of 100. We see that the standard Jacobi ($\omega = 1$) algorithm, requires many iterations to eliminate the highest frequency modes. On the other hand, the under-relaxed Jacobi scheme, eliminates the high modes very quickly, but on the other hand, the low frequency modes take longer, than with standard Jacobi, to disappear. We shall see that this slow down in the convergence of the low frequency modes is not really a problem and that, by using coarser meshes, we will be able to speed up the convergence of these modes.

2.2.3 Gauss-Seidel

SLIDE 8

Recall,

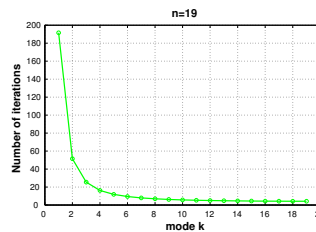


Is Gauss-Seidel a good smoother? ...

Since the eigenvectors of R_{GS} and A do not coincide, there is little we can say about the smoothing properties of Gauss-Seidel by looking at the eigenvalues of the iteration matrix.

SLIDE 9

Iterations required to reduce an A error mode by a factor of 100



... GS is a good smoother.

By looking at the number of iterations required to reduce the amplitude of each mode, of the A matrix, by a factor of 100, we can determine the smoothing properties of the Gauss-Seidel scheme. It turns out that based on the above graph, the high frequency modes do in fact decay at a much faster rate than the low frequency ones.

2.3 Restriction

SLIDE 10

We shall require procedures for transferring information between grids. The process of transferring a vector from a fine to a coarse mesh is called restriction.

Given w_h we obtain w_{2h} by **restriction**

$$w_{2h} = I_{2h}^h w_h$$

I_{2h}^h : restriction operator (matrix).

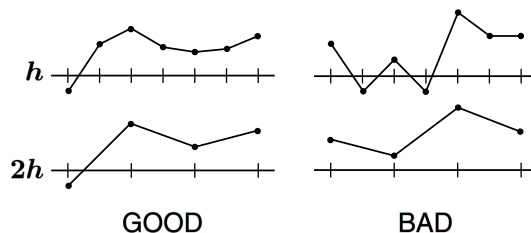
Simplest procedure is *injection*

$$w_{2h,i} = w_{h,2i} \quad \text{for } i = 1, \dots, \frac{n-1}{2}$$

We shall assume, for simplicity, that $n + 1$ is an even number. Restriction by injection reduces to taking the components of w_{2h} to be the components of w_h at every other point. We will see later that other forms of restriction can also be used.

SLIDE 11

Intuitively,



If the solution is “smooth”, the restricted function is a good approximation to the original grid function on the fine mesh. On the other hand, we see that for an “oscillatory” function, a lot of information is lost during the restriction operation.

SLIDE 12

The concept of “smooth” or “oscillatory” function can be made more precise.

If we write

v^k : eigenvectors of A

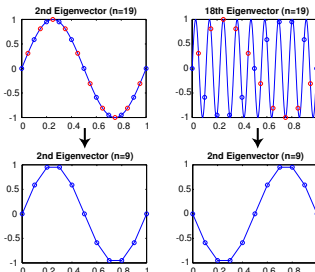
$$w_h = \sum_{k=1}^n c_k v^k$$

Only the modes $k = 1, \dots, \frac{n-1}{2}$ are “visible” by grid $2h$.

$$\underbrace{1, 2, \dots, \frac{n-1}{2}}_{\text{“visible” by grid } 2h}, \quad \underbrace{\frac{n+1}{2}, \dots, n-1, n}_{\text{aliased}}$$

2.3.1 Aliasing

SLIDE 13



Mode $k > (n - 1)/2$ on grid h becomes $(n - k)$ mode on grid $2h$.

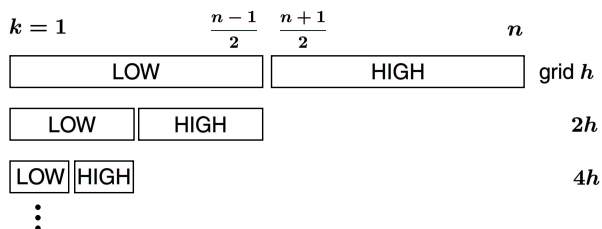
The effect of restricting a solution which has significant high frequencies can have very negative effects, since high modes on h may appear as low modes on $2h$ and hence exhibit slow convergence.

2.3.2 Summary

SLIDE 14

- Only low modes in h can be represented well in $2h$.
- Low modes on h become higher modes in $2h$.

Hence having a faster convergence rate.



2.4 Prolongation

SLIDE 15

The process of transferring a vector between a fine and a coarse mesh is known as prolongation.

Given w_{2h} we obtain w_h by **prolongation**

$$w_h = I_h^{2h} w_{2h}$$

I_h^{2h} : prolongation operator (matrix).

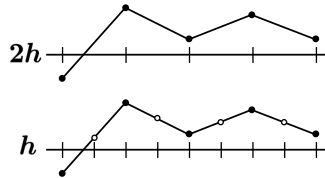
N1

Typically, we use *interpolation*.

$$i = 1, \dots, \frac{n-1}{2}$$

$$\begin{aligned} w_{h,2i} &= w_{2h,i} \\ w_{h,2i+1} &= \frac{1}{2}(w_{2h,i} + w_{2h,i+1}) \end{aligned}$$

SLIDE 16



Note 1

Galerkin Formulation

If the restriction and prolongation matrices satisfy

$$I_h^{2h} = c(I_{2h}^h)^T$$

for a constant c , the method is said to have the **variational property**. This is a property that facilitates some of the theoretical convergence proofs for multigrid methods. However, other simpler choices, like the ones considered above, can also work well in practice.

Consider, for illustration purposes, that the fine grid h has $n = 7$ points, and the coarse grid $2h$ has $n = 3$ points. The prolongation operator can be written as

$$I_h^{2h} \mathbf{w}_{2h} = \frac{1}{2} \begin{bmatrix} 1 & & & & & & \\ 2 & & & & & & \\ 1 & 1 & & & & & \\ & 2 & & & & & \\ & 1 & 1 & & & & \\ & & 2 & & & & \\ & & 1 & & & & \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix}_{2h} = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \\ w_5 \\ w_6 \\ w_7 \end{bmatrix} = \mathbf{w}_h$$

The corresponding restriction operator is usually referred to as a full weighting operator and has the form

$$I_{2h}^h \mathbf{w}_h = \frac{1}{4} \begin{bmatrix} 1 & 2 & 1 & & & & \\ & 1 & 2 & 1 & & & \\ & & 1 & 2 & 1 & & \\ & & & 1 & 2 & 1 & \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \\ w_5 \\ w_6 \\ w_7 \end{bmatrix}_h = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix}_{2h} = \mathbf{w}_{2h}$$

Just for completeness we give here the Galerkin prolongation and full weighting restriction operators in two dimensions. If $\mathbf{w}_{2h} = I_{2h}^h \mathbf{w}_h$, the components of

w_{2h} are given by

$$\left. \begin{aligned} w_{h\ 2i,2j} &= w_{2h\ i,j} \\ w_{h\ 2i+1,2j} &= \frac{1}{2}(w_{2h\ i,j} + w_{2h\ i+1,j}) \\ w_{h\ 2i,2j+1} &= \frac{1}{2}(w_{2h\ i,j} + w_{2h\ i,j+1}) \\ w_{h\ 2i+1,2j+1} &= \frac{1}{4}(w_{2h\ i,j} + w_{2h\ i+1,j} + w_{2h\ i,j+1} + w_{2h\ i+1,j+1}) \end{aligned} \right\} 1 \leq i, j \leq \frac{n-1}{2}$$

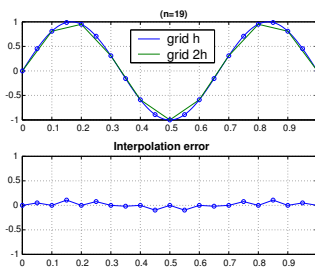
and for $w_h = I_h^{2h} w_{2h}$ the components of w_{2h} are

$$\begin{aligned} w_{2h\ i,j} &= \frac{1}{16} [w_{h\ 2i-1,2j-1} + w_{h\ 2i-1,2j+1} + w_{h\ 2i+1,2j-1} + w_{h\ 2i+1,2j+1} \\ &\quad + 2(w_{h\ 2i,2j-1} + w_{h\ 2i,2j+1} + w_{h\ 2i-1,2j} + w_{h\ 2i+1,2j}) + 4w_{h\ 2i,2j}], \\ &\quad 1 \leq i, j \leq \frac{n-1}{2}. \end{aligned}$$

2.5 Interpolation Error

SLIDE 17

When direct injection is used, the restriction of a low mode introduces no error. That is, a low mode k , on grid h , becomes mode k on grid $2h$. On the other hand, the prolongation operator introduces errors which contain high frequencies.



Interpolation introduces high frequency errors.

We observe from the figure that, the mode $k = 3$ on grid $2h$, produces after prolongation, the mode $k = 3$ on grid h plus a small, but important, high frequency error.

3 Two Grid (Correction) Scheme

SLIDE 18

One cycle

$$\mathbf{u}_h^{r+1} \leftarrow MG(\mathbf{u}_h^r, \mathbf{f}_h)$$

- Relax ν_1 iterations of $A_h \mathbf{u}_h = \mathbf{f}_h$ with initial guess $\mathbf{u}_h^r \rightarrow \mathbf{u}_h^{r+1/3}$.
- Compute $\mathbf{r}_h = \mathbf{f}_h - A_h \mathbf{u}_h^{r+1/3}$, and restrict $\mathbf{r}_{2h} = I_{2h}^h \mathbf{r}_h$.
- Solve $A_{2h} \mathbf{e}_{2h} = \mathbf{r}_{2h}$ on $2h$.
- Prolongate $\mathbf{e}_h = I_h^{2h} \mathbf{e}_{2h}$, and correct $\mathbf{u}_h^{r+2/3} = \mathbf{u}_h^{r+1/3} + \mathbf{e}_h$.
- Relax ν_2 iterations of $A_h \mathbf{u}_h = \mathbf{f}_h$ with initial guess $\mathbf{u}_h^{r+2/3} \rightarrow \mathbf{u}_h^{r+1}$.

Above we describe one cycle of a two grid correction scheme. The inputs are an initial guess \mathbf{u}_h^r , and a forcing vector \mathbf{f}_h . The output is the new approximation to the solution \mathbf{u}_h^{r+1} . Here, any of the relaxation, restriction, and prolongation schemes described, can be used.

We recall the equivalence between solving for $A \mathbf{u}_h = \mathbf{f}_h$ or $A \mathbf{e}_h = \mathbf{r}_h$. It turns out that writing the coarse grid correction in terms of the error leads to a simpler and more straightforward formulation.

ν_1 and ν_2 are usually referred to as the number of pre-and post-smoothing iterations, respectively.

3.1 Example

SLIDE 19

We solve

$$u(0) = u(1) = 0$$

$$-u_{xx} = -25(\sin(5\pi x) + 9 \sin(15\pi x)) .$$

Initial guess: $\mathbf{u}^0 = 0$

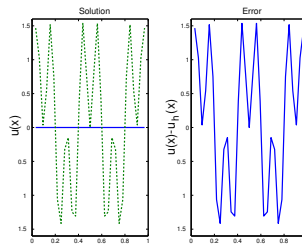
Solution: $u = \sin(5\pi x) + \sin(15\pi x)$

Two grid scheme: $h = \frac{1}{32}$, $2h = \frac{1}{16}$

Solve using under-relaxed Jacobi with $\omega = \frac{2}{3}$

SLIDE 20

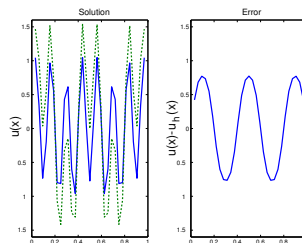
Initial condition



The error is made up of two components: $\sin(5\pi x)$ which is a low mode on the fine mesh, and a low (but not so low) mode on the coarse mesh. On the other hand, $\sin(15\pi x)$ is a low mode on the fine mesh and a high mode on the coarse mesh. The next figures, show the evolution of the error and the solution through the multigrid process.

SLIDE 21

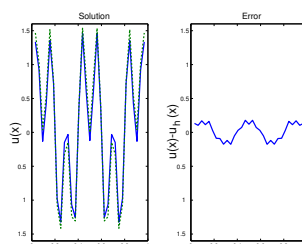
After $\nu_1 = 2$ iterations on the fine mesh



After three smoothing iterations the high error mode has been eliminated. The low error mode has suffered little change.

SLIDE 22

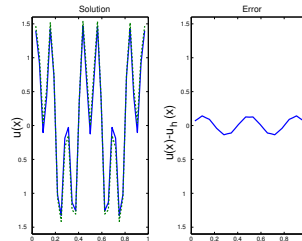
After coarse grid correction (4 iterations)



After the coarse grid correction, the low mode has been substantially reduced, but now, some new high modes have been introduced through the interpolation process.

SLIDE 23

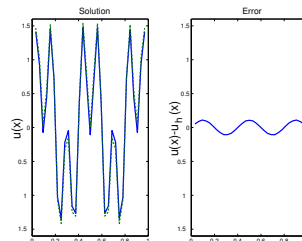
After $\nu_2 = 2$ post smoothing iterations (end of cycle 1)



The high modes have now been reduced.

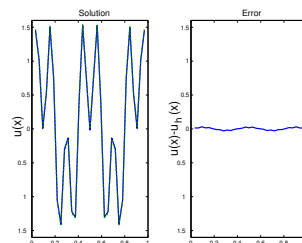
SLIDE 24

After $\nu_1 = 2$ iterations



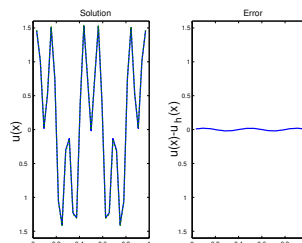
SLIDE 25

After coarse grid correction



SLIDE 26

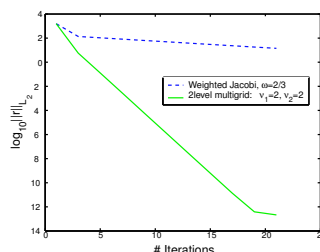
After $\nu_2 = 2$ iterations (end of cycle 2)



We see that the second cycle is analogous to the first cycle, and that, after only two cycles, the solution is almost converged.

SLIDE 27

Multigrid convergence vs. single grid



This figure illustrates the much faster convergence obtained using the multigrid procedure. The convergence level is shown versus the number of iterations on the fine mesh. One can think of the number of iterations in the fine mesh, as being approximately proportional to the amount of computational work involved.

4 Multiple Grids

4.1 V-Cycle

SLIDE 28

The two grid scheme presented above leaves unresolved the question of how to solve for e_{2h} in the coarse mesh. The answer comes from realizing that the problems $A_h \mathbf{u}_h = \mathbf{f}_h$ and $A_{2h} \mathbf{e}_{2h} = \mathbf{r}_{2h}$ look exactly the same and hence the same procedure can be used to solve them.

One cycle

$$\mathbf{u}_h^{r+1} \leftarrow VG_h(\mathbf{u}_h^r, \mathbf{f}_h)$$

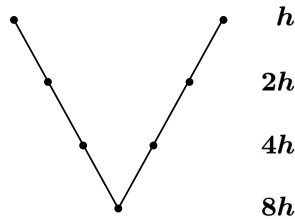
- Relax ν_1 times on $A_h \mathbf{u}_h = \mathbf{f}_h$ with initial guess $\mathbf{u}_h^r \rightarrow \mathbf{u}_h^{r+1/3}$.
- If $h \equiv$ coarsest grid, go to (SKIP)
- Else

$$\begin{aligned} \mathbf{r}_{2h} &\leftarrow I_{2h}^h(\mathbf{f}_h - A_h \mathbf{u}_h^{r+1/3}) \\ \mathbf{e}_{2h} &\leftarrow VG_{2h}(\mathbf{0}, \mathbf{r}_{2h}) . \end{aligned}$$
- Correct $\mathbf{u}_h^{r+2/3} = \mathbf{u}_h^{r+1/3} + I_h^{2h} \mathbf{e}_{2h}$.
- (SKIP) Relax ν_2 times on $A_h \mathbf{u}_h = \mathbf{f}_h$ with initial guess $\mathbf{u}_h^{r+2/3} \rightarrow \mathbf{u}_h^{r+1}$.

The step $\mathbf{e}_{2h} \leftarrow VG_{2h}(\mathbf{0}, \mathbf{r}_{2h})$, means that we recursively invoke the procedure VG , but now to solve $A_{2h}\mathbf{e}_{2h} = \mathbf{r}_{2h}$ with an initial guess $\mathbf{e}^0 = \mathbf{0}$. Within V_{2h} we will invoke V_{4h} to solve $A_{4h}\mathbf{e}_{4h} = \mathbf{r}_{4h} = I_{4h}^{2h}(\mathbf{r}_{2h} - A_{2h}\mathbf{e}_{2h})$, etc.. This process will continue until we reach the coarsest mesh, at which point we will only do $\nu_1 + \nu_2$ smoothing iterations and prolongate down the resulting solution. This V-cycle is illustrated below for four grids.

4.1.1 Schematically

SLIDE 29



4.1.2 2D Example

SLIDE 30

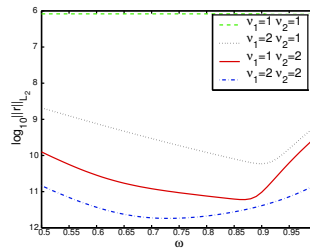
Solve

$$-(u_{xx} + u_{yy}) = 1, \quad \in \Omega \equiv \text{unit square}$$

$$u = 0 \quad \text{on the boundary}$$

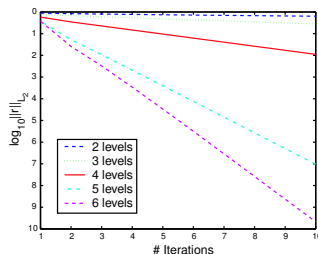
SLIDE 31

Parameter dependence



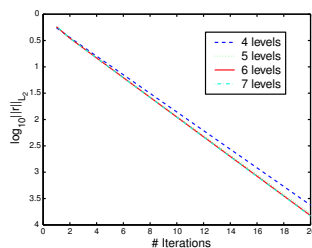
In order to determine the optimum combination of parameters some experimentation is often needed. The figure shows the residual reduction after 10 complete V-cycles, using 6 grids. The finest grid has 64×64 intervals. We see that the optimum value of ω depends on the number of pre- and post-smoothing iterations. We also observe that for $\nu_1 = \nu_2 = 1$, the algorithm does not converge. Can you explain why?

Convergence as a function of grid levels (same fine mesh)



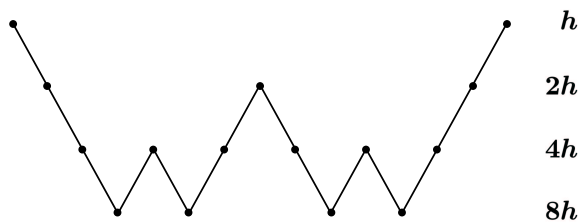
In this plot we show the convergence for 10 V-cycles of the multigrid algorithm, keeping a constant fine mesh of 64×64 . We see that the more grids we consider the faster the convergence, as expected. We are using $\omega = 0.8$ and $\nu_1 = 2$, $\nu_2 = 1$.

Convergence as a function of grid levels (same coarse mesh)



This figure illustrates the fact that the convergence rate is determined by the coarsest mesh. Essentially this observed behaviour proves that the number of iterations is independent of the number of grid points, provided sufficient coarse meshes are considered, and hence the overall algorithm has a cost which is $O(n)$. For this example, we are using $\omega = 0.8$ and $\nu_1 = 2$, $\nu_2 = 1$.

4.2 W-Cycles



Other types of cycles are also possible. The objective of W -cycles is to concentrate much of the work on the coarse meshes.

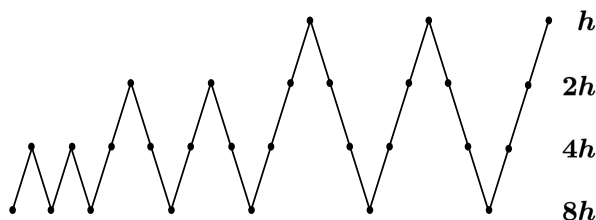
5 Full Multigrid Scheme

5.1 Schematically

SLIDE 35

Putting it all together ...

So far we have only used one of the main ideas involved in multigrid. The idea of starting with a good initial guess has not been implemented yet. This is accomplished with the full multigrid scheme.



6 More Advanced Topics

SLIDE 36

- Anisotropic grids/equations.
- Algebraic multigrid.
- Convergence theory.
- How to deal with other operators.

The above are some topics which are the subject of active research and which are beyond the scope of this lecture. In particular, algebraic multigrid methods are gaining increased popularity because of their potential as a general equation solver. Here, we have assumed that our matrix was the result of discretizing a PDE on a given grid of size h . In this setting, it is straightforward to define the coarse grid problems. If our equation system, $Au = f$, does not come from the discretization of a PDE, or the discretization process is not available, the procedures given in this lecture can not be used directly. Dealing with such general systems, without requiring information about the discretization process, is the subject of algebraic multigrid methods.

See [BHM] for further details.

References

[BHM] W.L. Briggs, V.E. Henson and S.F. McCormick, “A Multigrid Tutorial”, SIAM Monograph, 2000.