Multigrid

A Basic Tutorial

Babak Gholami

CSE

TUM

JASS2009

Contents

- Introduction
- Model Problems
- Basic Iterative Methods
- Elements of Multigrid
- Implementation
- Nonlinear Problem

Introduction

Origins...

- Multigird is an entire approach to computational problem solving, a collection of ideas...
- Originally, for solving BVP's.
- But later:
  - Parabolic and Hyperbolic PDE's
  - Purely algebraic problems with no physical grid
  - Optimization
  - Integral Equations
  - ...

Capabalities

- A wide variety of problems
- Arbitrary regions and boundary conditions
- An optimal solver (with the right setup)
- Can be efficiently parallelized
- Effectively treats local demands (Multilevel Adaptive Methods)

Model Problems

What problem?

- Multigrid methods were originally applied to simple BVPs.
- For simplicity we do the same here (for now!)
- Let’s call this “Model Problem”:

$$-\nabla^2 u + \sigma u = f\left(x, y, z, \ldots\right), \qquad 0 < x, y, z, \ldots < 1, \quad \sigma \geq 0$$

- Zero at boundaries.

Discretization

- Two dimensional case:

$$\frac{-v_{i-1,j}+2v_{ij}-v_{i+1,j}}{h_x^{2}}+\frac{-v_{i,j-1}+2v_{ij}-v_{i,j+!}}{h_y^{2}}+\sigma v_{ij}=f_{ij},$$

$$v_{i0}=v_{in}=v_{0j}=v_{mj}=0, \qquad 1\le i\le m-1, \quad 1\le j\le n-1.$$

- Or more compactly:

$$A\mathbf{v}=\mathbf{f}$$

- Direct and Relaxation methods are the two large categories for solving such a problem.

Direct Methods

- Determine a solution up to the machine’s precision in a finite number of arithmetic steps.
- Gaussian Elimination is a prototype.
- But we are interested in relaxation methods here…

Basic Iterative Methods

Notation

- $\mathbf{u}$ denotes the exact solution and $\mathbf{v}$ the current approximation.
- To associate $\mathbf{u}$ with a grid Ω^h, $\mathbf{u}^h$ is used.
- $\mathbf{e}$ is used for error which is given by:

$$\mathbf{e} = \mathbf{u} - \mathbf{v}$$

- The residual is shown by $\mathbf{r}$ and is defined this way:

$$\mathbf{r} = \mathbf{f} - A\mathbf{v}$$

More on the Residual

- Knowing that $A\mathbf{u} = \mathbf{f}$, we can write the residual equation:

$$A\mathbf{e} = \mathbf{r}$$

- It says the error will satisfy the same set of equations as the unknown when the right hand side is replaced by the residual.
- The residual is simply the amount by which the current approximation fails to satisfy our problem.

A Simple Scheme: Jacobi

- Jacobi is solving the jth equation of system of equations for the jth unknown using current approximation for all other variables.
- We can express the relaxation scheme in matrix form:

$$A = D - L - U$$

$$\Rightarrow \quad \mathbf{u} = R_J\mathbf{u} + D^{-1}\mathbf{f}, \qquad R_J = D^{-1}(L+U).$$

- So we write: $\mathbf{v}^{(1)} = R_J\mathbf{v}^{(0)} + D^{-1}\mathbf{f}.$

Weighted Jacobi

- We make a modification and introduce an entire family of iterations called weighted Jacobi (damped Jacobi).
- In matrix form:

$$\mathbf{v}^{(1)} = R_\omega \mathbf{v}^{(0)} + \omega D^{-1}\mathbf{f}, \qquad R_\omega = (1-\omega)I + \omega R_J$$

- Using the definition of the residual:

$$\mathbf{v}^{(1)} = \mathbf{v}^{(0)} + \omega D^{-1}\mathbf{r}^{(0)}.$$

- It can be shown that the closer ωD^{-1} is to A^{-1} the more effective is our iteration scheme.

More Relaxation Schemes(1)

- Gauss-Seidel: Components of the new approximation are used as soon as they are computed
- In matrix form:

$$\mathbf{v} \leftarrow R_G \mathbf{v} + (D-L)^{-1}\mathbf{f}, \qquad R_G = (D-L)^{-1}U.$$

- The order in which the components of $\mathbf{v}$ are updated is important.
- An effective alternative to just sweeping in ascending order is red-black Gauss-Seidel.

More Relaxation Schemes(2)

- Red-black Gauss-Seidel updates all even components first, and then goes through the odd components.
- This scheme is very well suited for parallel computation. (each sweep can be done by several independent processors)

More Investigation(1)

- Because these methods are stationary linear it is enough to work with the homogeneous linear system $A\mathbf{v} = 0$ and use arbitrary initial guesses.
- To study the one-dimensional case of the model problem we use Fourier modes for the initial guess:

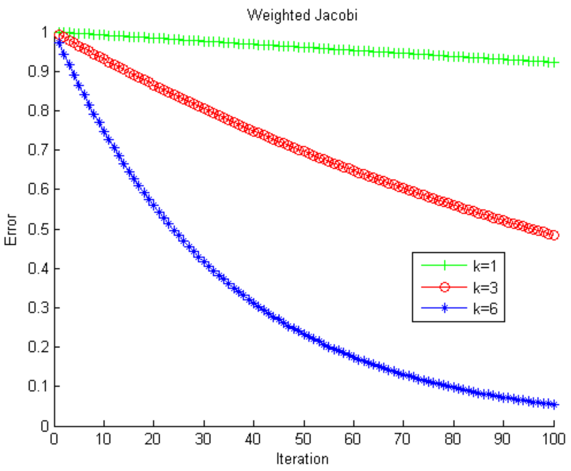
$$v_j = \sin\left(\frac{jk\pi}{n}\right), \qquad 0 \le j \le n, \quad 1 \le k \le n-1.$$

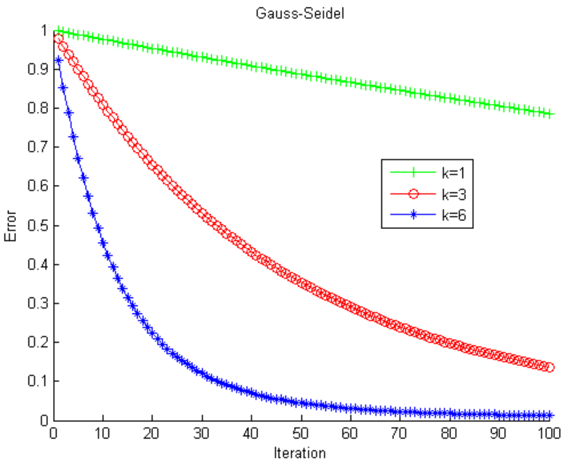
More Investigation(2)

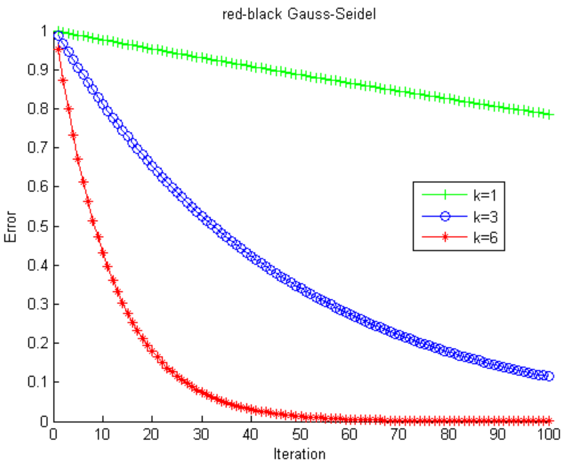
- Introducing the wavenumber (frequency), $\mathbf{v}_k$ designates the entire vector $\mathbf{v}$ with wavenumber k.
- To compare the introduced methods first the weighed Jacobi with $\omega = {}^{2}\!/_{3}$ is applied to the one-dimensional version of the model problem with $\mathbf{f} = 0$ on a grid with 64 points.
- Beginning with $\mathbf{v}_1$, $\mathbf{v}_3$ and $\mathbf{v}_6$ the iteration is applied 100 times.

Results(1)

- Here the norm of the error is plotted against the number of iterations for all the three methods. What is important now is the qualitative behavior.





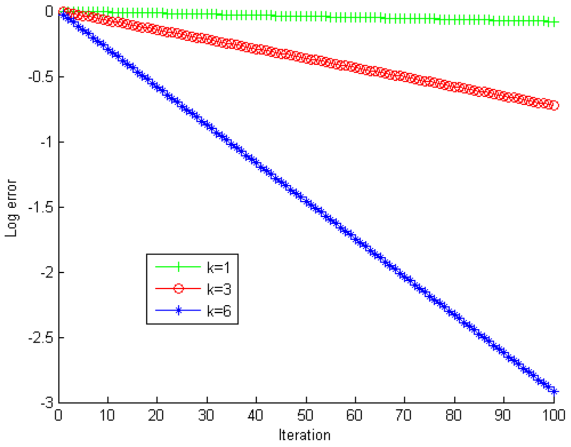


Results(2)

- Here the logarithm of the norm of the Jacobi error is plotted against the number of iterations.
- This clear linear behavior indicates that the error itself decreases geometrically with each iteration:

$$\left\|e^{(m)}\right\|_{\infty} = c_k^m \left\|e^{(0)}\right\|_{\infty}$$

- This constant depends on the wavenumber.

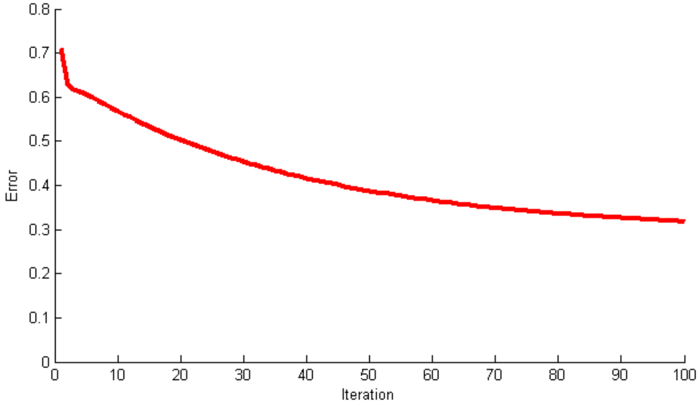


More Realistic Situation(1)

- In general most initial guesses (or equivalently the right hand side) consist of different modes.
- This is been simulated with an initial guess consisting of the average of one low frequency mode ($k=1$), a medium ($k=6$) and a high frequency ($k=32$).
- The figure in the next slide shows the result for weighted Jacobi applied to the problem.

More Realistic Situation(2)

- The standard iteration converges very quickly only as long as the error has high-frequency components. The slower elimination of the low-frequency components desgrades the performance of the relaxation methods.



What to Conclude?

- This drawback of the relaxation schemes can be analytically proven.
- Moreover, one can find an optimum for ω to reduce the smooth components of the error more effectively.
- This is not enough.

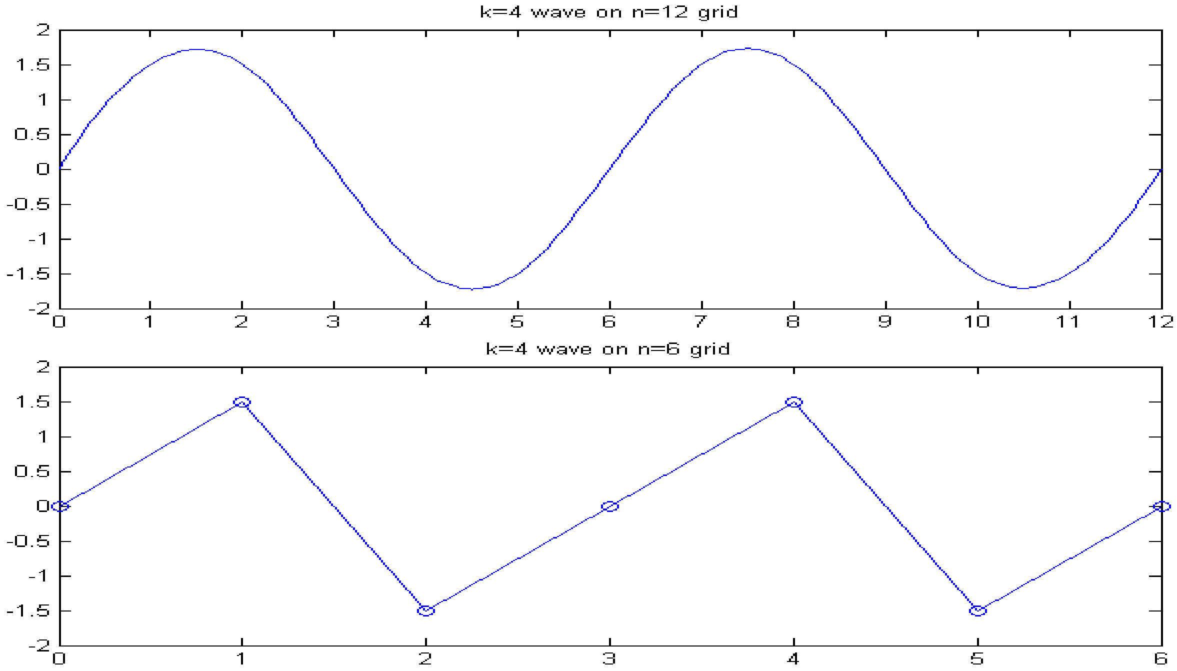
Elements of Multigrid

Why Considering Coarser Grids? (1)

- So far we have seen that most of basic iterative methods possess the smoothing property.
- One idea is to take advantage of a good initial guess.
- A well-known technique to obtain one is perform some iterations on a coarse grid.
- It is cheaper due to fewer unknowns and faster while convergence behaves like $1-O\left(h^{2}.\right)$

Why Considering Coarser Grids? (2)

- A smooth wave with $k = 4$ on Ω^h with $n = 12$ points has been projected directly to the grid Ω^{2h} with $n = 6$.



Why Considering Coarser Grids? (3)

- To state this more precisely, note that the grid points of the coarse grid are the even-numbered grid points of the fine grid.
- For the kth mode:

$$w^h_{k,2j} = \sin\left(\frac{2jk\pi}{n}\right) = \sin\left(\frac{jk\pi}{n/2}\right) = w^{2h}_{k,j}, \qquad \mathbf{1} \le k < \frac{n}{2}$$

- So the kth mode on the fine grid is the kth mode on the coarse grid. But there are half as many modes on Ω^{2h} as there are on Ω^h.

The Starting Point of Multigrid

- Smooth modes on a fine grid look less smooth on a coarse one.
- This suggests that when relaxation begins to stall, move to a coarser grid.
- But how do we move?
- Relaxation on the original equation $A\mathbf{u} = \mathbf{f}$ with a arbitrary initial guess is equivalent to relaxing on the residual equation $A\mathbf{e} = \mathbf{r}$ with the specific initial guess $\mathbf{e} = 0$.

Putting the Ideas Together (1)

- Using the coarse grid to obtain better initial guess:
  - Relax on $A\mathbf{u} = \mathbf{f}$ on a very coarse grid to obtain an initial guess for the next finer grid.

    .

    .

    .
  - Relax on $A\mathbf{u} = \mathbf{f}$ on Ω^{2h} to obtain an initial guess for Ω^{h} .
  - Relax on $A\mathbf{u} = \mathbf{f}$ on Ω^{h} to obtain a final approx.
- The basis of the strategy “nested iteration.”

Putting the Ideas Together (2)

- Using the residual equation to relax on error:
  - Relax on $A\mathbf{u} = \mathbf{f}$ on Ω^h to obtain an approximation.
  - Compute the residual $\mathbf{r} = \mathbf{f} - A\mathbf{v}^h$.
    - Relax on the residual equation $A\mathbf{e} = \mathbf{r}$ on Ω^{2h} to obtain an approximation to the error $\mathbf{e}^{2h}$.
  - Correct the approximation obtained on Ω^h with the error estimate obtained on Ω^{2h} : $\mathbf{v}^h = \mathbf{v}^h + \mathbf{e}^{2h}$.
- Basis of the strategy “correction scheme.”
- How do we transfer errors between grids?

Intergrid Transfers; Interpolation(1)

- There is usually no advantage in using grid spacing with ratio other than 2.
- Transferring information form a coarse grid to a fine one is called interpolation or prolongation.
- There are, of course, many methods!
- Linear interpolation: $I_{2h}^{h}\mathbf{v}^{2h} = \mathbf{v}^{h}$

$$v_{2j}^{h} = v_{j}^{2h},$$

$$v_{2j+1}^{h} = \left(v_{j}^{2h} + v_{2j+1}^{2h}\right)\Big/2, \qquad 0 \le j \le {}^{n}\!/_{2} - 1.$$

Intergrid Transfers; Interpolation(2)

- How well does this work?
- We can show that interpolation is most effective when the error is smooth.
- Nested iteration and Correction scheme both use interpolation, so are they only effective when the error is smooth?
- Happily, these processes provide a complement to relaxation that is most effective when the error is oscillatory.

Intergrid Transfers; Restriction

- Form fine to coarse $\left(I_h^{2h}\right)$.
- One obvious restriction operator is injection:

$$v_j^{2h} = v_{2j}^h .$$

- An alternative is called full weighting:

$$v_j^{2h} = \frac{1}{4}\left(v_{2j-1}^h + 2v_{2j}^h + v_{2j+1}^h\right), \qquad 1 \le j \le \frac{n}{2} - 1.$$

- Now we are ready to introduce our first multilevel scheme.

Two-Grid Correction Scheme(1)

$$\mathbf{v}^h \leftarrow MG\left(\mathbf{v}^h, \mathbf{f}^h\right).$$

- Relax ν_1 times on $A^h\mathbf{u}^h = \mathbf{f}^h$ on Ω^h with initial guess $\mathbf{v}^h$.
- Compute the fine-grid residual $\mathbf{r}^h = \mathbf{f}^h - A^h\mathbf{v}^h$ and restrict it to the coarse grid.
- Solve $A^{2h}\mathbf{e}^{2h} = \mathbf{r}^{2h}$ on Ω^{2h}.
- Interpolate the coarse-grid error and correct the fine grid approximation by $\mathbf{v}^h \leftarrow \mathbf{v}^h + \mathbf{e}^h$.
- Relax ν_2 times on $A^h\mathbf{u}^h = \mathbf{f}^h$ on Ω^h with initial guess $\mathbf{v}^h$.

Two-Grid Correction Scheme(2)

- In practice v_1 is often 1,2, or 3.
- Here is the fortunate complement that the scheme brings: Relaxation on the fine grid eliminates the oscillatory components of the error, leaving a relatively smooth error.

Numerical Example(1)

- Weighted Jacobi with $\omega = {}^{2}\!/_{3}$ is applied to the one-dimensional model problem on a grid with 64 points.
- The initial guess contains 16th and 40th modes. (one quite smooth and one quite oscillatory)
- The aforementioned two-grid correction scheme with $v_1 = v_2 = 3$ is used.

Numerical Example(2); Results

- After one relaxation sweep the 2-norm of the error has been diminished to 57% of the norm of the initial guess.
- After three sweeps the reduction is 36%.
- By one relaxation sweep on the coarse grid the error will be reduced to 26%.
- After three coarse-grid sweeps: 8%.
- Correcting the fine grid approximation and performing three relaxation sweeps: 3%.
- Once again to the coarse grid and then three relaxation sweeps: 1%.

More Advanced Schemes; V-Cycle(1)

$$\mathbf{v}^h \leftarrow V^h\left(\mathbf{v}^h, \mathbf{f}^h\right).$$

1. Relax ν_1 times on $A^h \mathbf{u}^h = \mathbf{f}^h$ on Ω^h with a given initial guess $\mathbf{v}^h$.
2. If Ω^h=coarsest grid, then go to step 4.

   Else

$$\mathbf{f}^{2h} \leftarrow I_h^{2h}\left(\mathbf{f}^h - A^h \mathbf{v}^h\right),$$

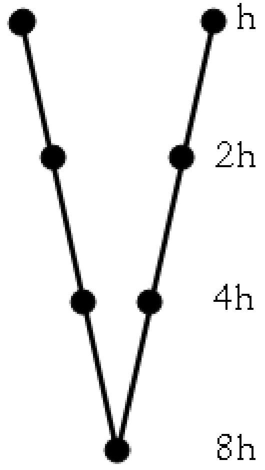
$$\mathbf{v}^{2h} \leftarrow 0,$$

$$\mathbf{v}^{2h} \leftarrow V^{2h}\left(\mathbf{v}^{2h}, \mathbf{f}^{2h}\right).$$

7. Correct $\mathbf{v}^h \leftarrow \mathbf{v}^h + I_{2h}^h \mathbf{v}^{2h}$.
8. Relax ν_2 times on $A^h \mathbf{u}^h = \mathbf{f}^h$ on Ω^h.

More Advanced Schemes; V-Cycle(2)

- In a V-Cycle scheme we use the two-grid scheme within itself.
- Here is the schedule of grids for V-Cycle:



More Advanced Schemes; μ-Cycle(1)

$$\mathbf{v}^h \leftarrow M\mu^h\left(\mathbf{v}^h, \mathbf{f}^h\right).$$

1. Relax ν_1 times on $A^h\mathbf{u}^h = \mathbf{f}^h$ on Ω^h with a given initial guess $\mathbf{v}^h$.
2. If Ω^h =coarsest grid, then go to step 4.

   Else

$$\mathrm{f}^{2h} \leftarrow I_h^{2h}\left(\mathrm{f}^h - A^h\mathrm{v}^h\right),$$

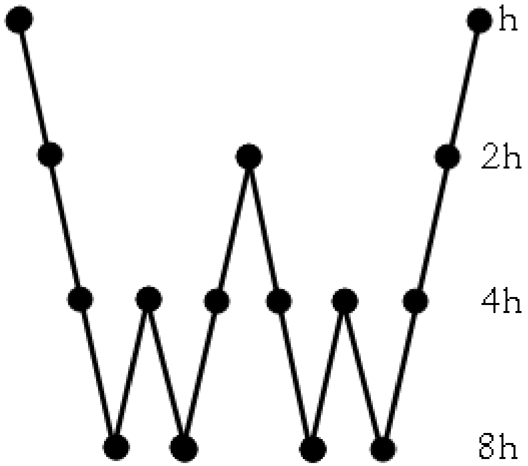
$$\mathrm{v}^{2h} \leftarrow 0,$$

$$\mathrm{v}^{2h} \leftarrow M\mu^{2h}\left(\mathrm{v}^{2h}, \mathrm{f}^{2h}\right) \ \mu \text{ times} .$$

7. Correct $\mathbf{v}^h \leftarrow \mathbf{v}^h + I_{2h}^h\mathbf{v}^{2h}$.
8. Relax ν_2 times on $A^h\mathbf{u}^h = \mathbf{f}^h$ on Ω^h.

More Advanced Schemes; μ-Cycle(2)

- μ -cycle is an entire multigrid cycling scheme family and V-cycle is just one member of this family ($\mu = 1$).
- W-cycle, shown below, is the case with $\mu = 2$:



Full Multigrid V-Cycle(1)

- So far we have developed only the correction scheme.
- The nested iteration idea has yet to be explored.
- Nested iteration suggests solving a problem on Ω^{2h} to obtain an initial guess for solving the problem on Ω^{h}.
- But where does the initial guess for Ω^{2h} come from?

Full Multigrid V-Cycle(2)

$$\mathrm{v}^{h} \leftarrow FMG^{h}\left(\mathrm{f}^{h}\right).$$

2. If Ω^h=coarsest grid, set $\mathbf{v}^h \leftarrow 0$ and go to step 3.

   Else
   $$\mathbf{f}^{2h} \leftarrow I_h^{2h}\left(\mathbf{f}^{h}\right),$$
   $$\mathbf{v}^{2h} \leftarrow FMG^{2h}\left(\mathbf{f}^{2h}\right).$$

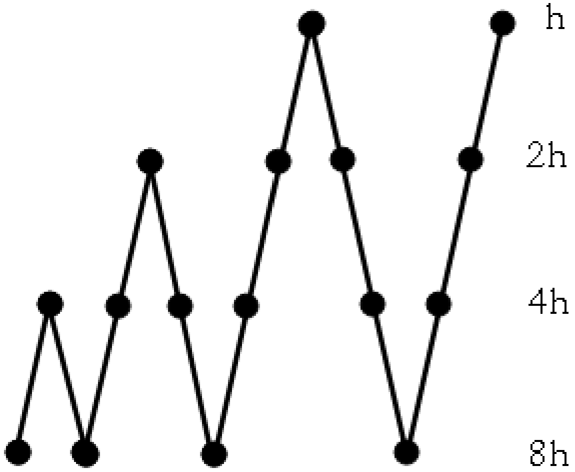
5. Correct $\mathbf{v}^h \leftarrow I_{2h}^{h}\mathbf{v}^{2h}$.

6. $\mathbf{v}^h \leftarrow V^h\left(\mathbf{v}^h,\mathbf{f}^h\right)$ ν_0 times.

Full Multigrid V-Cycle(3)

- We do some extra work to find the best initial guess possible. But this is not only inexpensive, but easily pays for itself. The result is a very powerful algorithm.
- Here is the schedule

  for FMG scheme in 4 levels:



Implementation

Writing a Code

- We now want to turn to more practical issues of writing a multigird program.
- The experiment of many practitioners suggests that such a program should be highly modular.
- This way, besides its simplicity, one can change different components of his code.
- We will see how effective that can be.

Data Structure(1)

- Choosing an appropriate data structure is also of great importance.
- One way is to declare a new structure that groups together all the associated information for each grid level.
- Going further in this area without pointing a specific language seems pointless since every discussion would soon be outdated!

Data Structure(2)

- In languages with more restrictive data structures, like MATLAB and FORTRAN, there seems to be a general agreement.
- That is the solutions and the right-side vectors on the various grids should be stored contiguously in single arrays (so two arrays).
- Actually, mutligrid codes started to “grow up” in such an environment.

Complexity(1)

- How much do the multigrid scheme cost in terms of storage and computation?
- Consider a d-dimensional grid with n^d points.
- For simplicity n is a power of two.
- On the finest grid we need n^d storage locations for each array (two arrays).
- From now on each grid needs 2^{-d} times as much storage as the finer grid before it.
- Adding these term give a geometric series.

Complexity(2)

- Storage= $2n^d\left(1+2^{-d}+2^{-2d}+\ldots+2^{-nd}\right)<\dfrac{2n^d}{1-2^{-d}}$
- The storage costs of multigrid algorithm decreases relatively as the dimension of the problem increases.
- Same analysis gives an estimation of the computational costs of multigird mehtods.
- Note that in the results nest page the cost of intergrid transfers, typically 10-20% of the cost of the entire cycle, is neglected.

Complexity(3)

- In a V-cycle with $v_1 = v_2 = 1$ each level is visited twice and grid Ω^{ph} requires p^{-d} work units, so the computational cast for a V-cycle is:

$$2\left(1 + 2^{-d} + 2^{-2d} + \ldots + 2^{-nd}\right) < \frac{2}{1 - 2^{-d}} WU.$$

- With a slight modification for the FMG we can obtain:

$$\left(\frac{2}{1 - 2^{-d}}\right)\left(1 + 2^{-d} + 2^{-2d} + \ldots + 2^{-nd}\right) < \frac{2}{\left(1 - 2^{-d}\right)^2} WU.$$

Complexity(4); final points

- We again can see with increasing the dimension of the problem the relative computational cost decreases in both scheme.
- As expected, a single FMG cycle costs more than a single V-cycle (the difference is less in higher dimensions).
- To reach a final idea which one is more suitable we need to know how many cycles they need to give satisfactory results.

Numerical Example(1)

- We solve the two-dimensional problem

$$-u_{xx}-u_{yy}=2\left[\left(1-6x^2\right)y^2\left(1-y^2\right)+\left(1-6y^2\right)x^2\left(1-x^2\right)\right]$$

in unit square with zero on the boundaries.

- We aim to
  - Compare results using different relaxation, interpolation and restriction operators
  - Make a conclusion about the effectiveness of the V-cycle and FMG schemes

Numerical Example(2)

- In comparing different operators more than what introduced before here we add half-injection and cubic interpolation.
- Half-injection in one-dimension is simply half of the injection operator.
- Cubic interpolation in one-dimension is defined this way:

$$v_{2j}^{h} = v_{j}^{2h},$$

$$v_{2j+1}^{h} = \frac{1}{16}\left(-v_{j-1}^{2h} + 9v_{j}^{2h} + 9v_{j+1}^{2h} - v_{j+2}^{2h}\right)$$

Results(1); Different Operators

	Relaxation	Injection		Full Weighting		Half-Injection	
(v_1, v_2)	Scheme	Linear	Cubic	Linear	Cubic	Linear	Cubic
$(1,0)$	Jacobi	-	-	0.49	0.49	0.55	0.62
	GS	0.89	0.66	0.33	0.34	0.38	0.37
	RBGS	-	-	0.21	0.23	0.45	0.42
	Cost	1.00	1.25	1.13	1.39	1.01	1.26
$(1,1)$	Jacobi	0.94	0.56	0.35	0.34	0.54	0.52
	GS	0.16	0.16	0.14	0.14	0.45	0.43
	RBGS	-	-	0.06	0.05	0.12	0.16
	Cost	1.49	1.75	1.63	1.88	1.51	1.76
$(2,1)$	Jacobi	0.46	0.31	0.24	0.24	0.46	0.45
	GS	0.07	0.07	0.08	0.07	0.40	0.39
	RBGS	-	-	0.04	0.03	0.03	0.07
	Cost	1.99	2.24	2.12	3.37	1.51	1.76

Results(2); Different Operators

- At least in this problem cubic interpolation is noticeably more effective than linear one only when injection is used for restriction.
- Not surprisingly, you get what you pay for: better convergence factor comes with higher cost.
- Parameter selection largely depends on what we want: cost or performance?

Results(3); V-cycle or FMG?

- Here, comparing FMG(1,1) and V(1,2), for all grids FMG is less expensive than V-cycle.
- It confirms the fact that for converging to the level of discretization error, full multigrid methods are generally preferable to simple V-cycles.

	FMG(1,1)	V(2,1)	V(2,1)
	WU	cycles	WU
4	7/2	3	12
8	7/2	4	16
16	7/2	4	16
32	7/2	5	20
64	7/2	5	20
128	7/2	6	24
256	7/2	7	28
512	7/2	7	28
1024	7/2	8	32
2048	7/2	9	36

Diagnostic Tools(1)

- Debugging can be the most difficult part of creating a successful program. Here is a short list of useful debugging techniques in evaluating a mutigrid code:

2. Methodical Plan: modular! Focus should be firstly on the solver.
3. Starting Simply: basic methods, simple problems.
4. Homogenous Problem: norms of the residual and the error.

Diagnostic Tools(2)

1. Residual Printout: on each level the norm of the residual should decline to machine zero at a steady rate.
2. Error Graph: is it oscillatory after coarse-grid correction? Effectively smoothed by relaxation? Any unusual behavior?
3. Two-Level Cycles: it is necessary that the two-level scheme work, test it!

There is no end to this list.

Nonlinear Problem

Nonlinearity

- Up to now everything was linear!
- Do we need to make any changes to treat nonlinear problems?
- Let's take a look at the most significant difference between linear and nonlinear systems.
- Consider the nonlinear system below (note the new notation):

$$A(\mathbf{u}) = \mathbf{f}$$

Residual Equation

- From the definition of the residual we find the new residual equation:

$$A(\mathbf{u}) - A(\mathbf{v}) = \mathbf{r}.$$

- Even though $\mathbf{u} - \mathbf{v} = \mathbf{e},$ we cannot conclude that $A(\mathbf{u}) - A(\mathbf{v}) = A(\mathbf{e}).$
- Since the solver now needs to solve a nonlinear equation, it makes sense to take a look at a classical relaxation method for such a system.

Nonlinear Gauss-Seidel Relaxation

- The same as linear GS, we form the j th equation and update the corresponding component:

$$v_j \leftarrow M_j\left(v_1, v_2, \dots, v_{j-1}, v_{j+1}, \dots, v_n\right), \quad 1 \le j \le n.$$

- In cases where we cannot form the equation explicitly, the following system should be solved using a few steps of Newton method:

$$\left(A\left(\mathbf{v} + s\varepsilon_j\right)\right)_j = \mathbf{f}_j, \quad 1 \le j \le n.$$

- When s is found: $\mathbf{v} \leftarrow \mathbf{v} + s\varepsilon_j$

Newton-Multigrid

- The new residual equation can be written:

$$A(\mathbf{v}+\mathbf{e})-A(\mathbf{v})=\mathbf{r}.$$

- Expanding the first term in Taylor series about $\mathbf{v}$ and truncating the series after two terms, we have a linear equation:

$$J(\mathbf{v})\mathbf{e}=\mathbf{r},$$

- This system is an approximation to the nonlinear system. One highly recommended option to solve it is multigrid.

Going Further…

- Newton-multigrid can be effective, but it does not use mutilgrid ideas to treat the nonlinearity directly.
- In a two-grid setting, the residual on the coarser grid appears as:

$$A^{2h}\left(\mathbf{v}^{2h}+\mathbf{e}^{2h}\right)-A^{2h}\left(\mathbf{v}^{2h}\right)=\mathbf{r}^{2h}.$$

- We choose the coarse-grid residual to be the restriction of the fine-grid residual:

$$\mathbf{r}^{2h}=I_h^{2h}\mathbf{r}^h=I_h^{2h}\left(\mathbf{f}^h-A^h\left(\mathbf{v}^h\right)\right).$$

The Nonlinear Version of Multigrid

- But what about the current approximation?
- Using the same operator for the residual, we restrict the current approximation on the fine gird to the coarser grid.

$$\mathbf{v}^{2h} = I_h^{2h}\mathbf{v}^h.$$

- Putting everything together in the residual equation and solving it gives the coarse-grid approximation for the error which can be interpolated to fine grid to correct the current approximation. This is FAS.

Full Approximation Scheme (FAS)

- FAS steps can be summarized this way:
  - Restrict the current approximation and its fien-grid residual to the coarse grid.
  - Solve this coarse-grid problem:

$$A^{2h}\left(\mathbf{u}^{2h}\right) = A^{2h}\left(\mathbf{v}^{2h}\right) + \mathbf{r}^{2h}.$$

  - Compute the coarse-grid approximation to the error :

$$\mathbf{e}^{2h} = \mathbf{u}^{2h} - \mathbf{v}^{2h}.$$

  - Interpolate the error approximation up to the fine grid and correct the current fine-grid approximation.

$$\mathbf{v}^{h} \leftarrow \mathbf{v}^{h} + I_{2h}^{h}\mathbf{e}^{2h}.$$

FAS Comments(1)

- If the operator is linear FAS reduces to the linear two-grid scheme.
- The process stalls at and only at the exact solution.
- The second step of the FAS procedure involves a nonlinear problem itself. In a two-level scheme it is solved with standard relaxation method such as nonlinear GS.
- A true FAS process would be done recursively.

FAS Comments(2)

- Thus, like its linear counterpart, FAS is usually implemented as a V-cycle or W-cycle scheme.
- The convergence of nonlinear iterations depend critically on a good initial guess. Using one FMG cycle can provide accuracy to the level of discretization (whether we use Newton-multigrid or FAS V-cycles).

Thank you!!