Adaptive Algebraic Multigrid

Scott MacLachlan

maclachl@colorado.edu

Department of Applied Mathematics, University of Colorado at Boulder

Support and Collaboration

- This work has been supported by the DOE SciDAC TOPS program, the Center for Applied Scientific Computing at Lawrence Livermore National Lab, and Los Alamos National Laboratory.
- This work has been performed in collaboration with Steve McCormick, Tom Manteuffel, John Ruge, Marian Brezina, James Brannick at CU-Boulder, and Rob Falgout at CASC.

Multiscale/Multiphysics problems

- Significant interest in simulating complex physical systems with features, and hence solutions, that vary on multiple scales
- Accuracy constraints are often driven by motivating applications, requiring efficient iterative methods to solve the resulting linear (and non-linear) systems
- Multiscale solution techniques, such as multigrid, are often most efficient approach
- Need simulation tools that can accurately detect non-standard behavior in the model and adapt to account for it

Multigrid

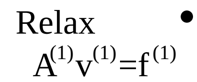
Multigrid Methods achieve optimality through complementarity

Multigrid

Multigrid Methods achieve optimality through complementarity

Multigrid Components

- Relaxation



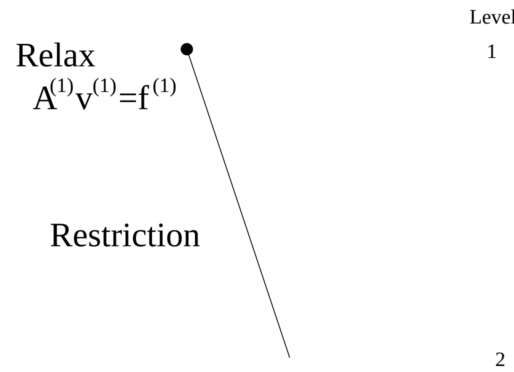
- Use a smoothing process (such as Gauss-Seidel) to eliminate oscillatory errors
- Remaining error satisfies $Ae = r \equiv f - Av$

Multigrid

Multigrid Methods achieve optimality through complementarity

Multigrid Components

- Relaxation
- Restriction



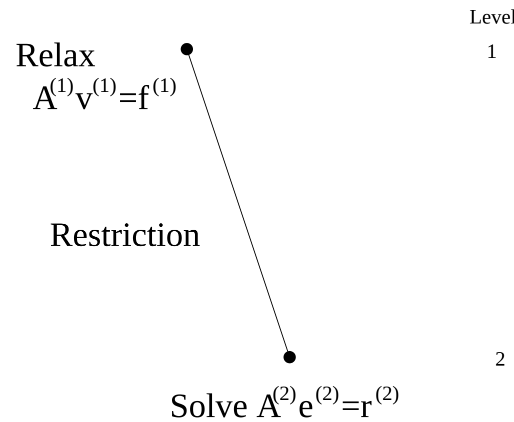
- Transfer residual to coarse grid

Multigrid

Multigrid Methods achieve optimality through complementarity

Multigrid Components

- Relaxation
- Restriction
- Coarse-Grid Correction



- Use coarse-grid correction to eliminate smooth errors
- To solve for error on coarse grid, use residual equation

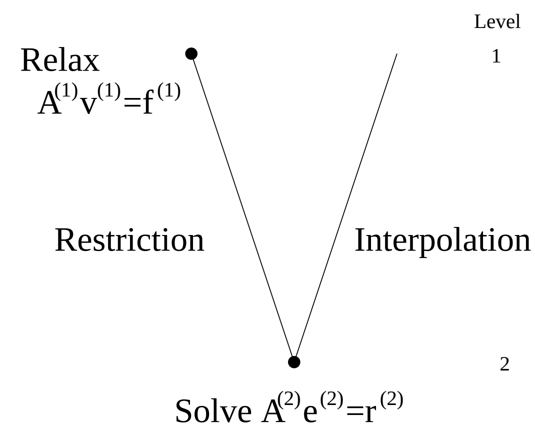
$$A^{(2)}e^{(2)} = r^{(2)}$$

Multigrid

Multigrid Methods achieve optimality through complementarity

Multigrid Components

- Relaxation
- Restriction
- Coarse-Grid Correction
- Interpolation



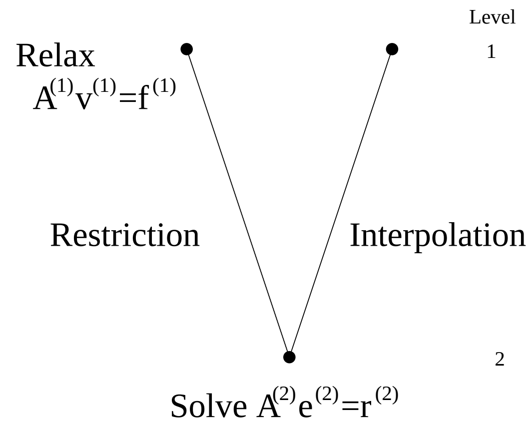
- Transfer correction to fine grid

Multigrid

Multigrid Methods achieve optimality through complementarity

Multigrid Components

- Relaxation
- Restriction
- Coarse-Grid Correction
- Interpolation
- Relaxation



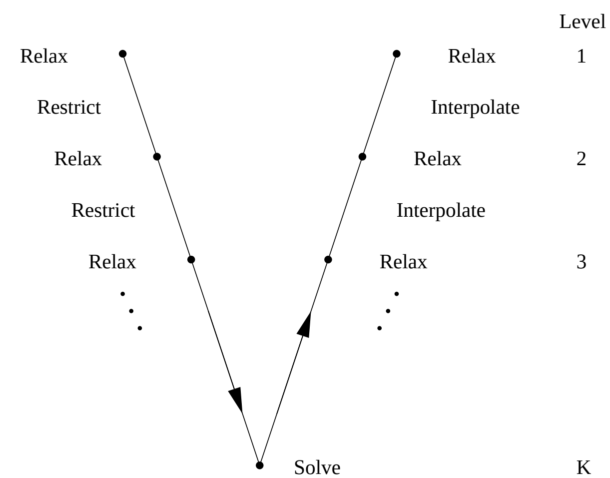
- Relax once again to remove oscillatory error introduced in coarse-grid correction

Multigrid

Multigrid Methods achieve optimality through complementarity

Multigrid Components

- Relaxation
- Restriction
- Coarse-Grid Correction
- Interpolation
- Relaxation



Obtain optimal efficiency through recursion

Algebraically Smooth Error

- Multigrid methods reduce error through
  - Relaxation (Jacobi, Gauss-Seidel)
  - Coarse-grid correction (variational)
- Error which is not efficiently reduced by relaxation is called *algebraically smooth* and must be reduced by coarse-grid correction
- Pointwise relaxation implies that algebraically smooth error, $\mathbf{e}$, satisfies $A\mathbf{e} \approx \mathbf{0}$, relative to $\mathbf{e}$
- If the origins of the matrix are known, so is character of algebraically smooth error

Algebraic Multigrid

- Assume no knowledge of grid geometry
- Interpolation and coarse grids chosen based only on the entries of the matrix
- Primary goal is to interpolate suitable corrections from the coarse grids
- Assume algebraically smooth error is locally constant
- Equivalently, assume global near null space is the constant vector

Adaptive Multigrid

- If we don’t know what algebraically smooth error looks like, can we still develop an effective multigrid method?

Adaptive Multigrid

- If we don't know what algebraically smooth error looks like, can we still develop an effective multigrid method? Yes!
- Use relaxation on $Av = 0$ to expose algebraic smoothness
- Fine-grid relaxation quickly exposes local character of algebraic smoothness
- Use this representation to determine interpolation
- Interpolation weights are chosen through a local collapsing of the operator done to fit the prototypical algebraically smooth error
- Apply these ideas recursively, using relaxation to expose appropriate components of the error on each level of the multigrid hierarchy

Adaptive Cycling

- Suppose the resulting cycle is ineffective - this indicates that relaxation and coarse-grid correction are not yet sufficiently complementary
- A representative of components that the current cycle does not quickly resolve can be found by applying it to the homogeneous problem
- This representative can then be used in conjunction with the existing prototype of algebraically smooth error to determine a better multigrid hierarchy
- Keep adapting the multigrid cycle until acceptable performance is achieved

Cost of Adaptivity

- Adaptation of multigrid components is quite expensive
- Recomputing interpolation at any level in the V-cycle requires recomputing operators at all coarser levels
- Added cost for each prototype to be fit is significant
- Need strategy and measures to ensure adaptations are performed as efficiently as possible

Choosing to Adapt

- Only want to adapt if further improvement in convergence is needed
- Need to measure performance of current solver, $\|I - BA\|$
  $\Rightarrow$ power method: iterate on $Av = 0$ with a random initial guess
- A few iterations on the homogeneous problem quickly exposes poor performance

How to Adapt

- Adding additional prototypes results in a more expensive method
- Want to ensure that each prototype is as good as possible before choosing to add more
- Measure strength of prototype as a representative of slowly-converging error: slowest converging error is

$$\operatorname*{argmax}_v \frac{\|(I - BA)v\|}{\|v\|}$$

- Estimate of solver performance also gives measure of how to adapt

Algorithm Overview

- while $\|I - BA\|_{\text{est}}$ is large
  - if $\|I - B^{\text{(old)}}A\|_{\text{est}}$ is increasing
    - iterate on $Av = 0$ with old solver, $v \leftarrow (I - B^{\text{(old)}}A)v$
    - recalibrate interpolation based on new v
    - recompute coarse-grid operator
    - restrict v to coarse grid and cycle there
    - interpolate further improved v after coarse-grid cycle
  - else
    - $B^{\text{(old)}} \leftarrow B$

Numerical Results

- 2-D Finite Element Shifted Laplacian, Dirichlet BCs, 512×512 grid

$$-\Delta u - 2\pi^2(1 - 2^{-15})u = 0$$

- $\lambda_{\text{min}} = 6.64 \times 10^{-4}$, random $v^{(0)}$, RQ$(v^{(0)}) = 2.06 \times 10^5$

Iteration	$\|I - B^{(\text{old})}A\|_{\text{est}}$	$RQ(v)$	$\|I - BA\|_{\text{est}}$
1	0.87	1.18×10^4	0.9999998
2	0.996	1.07×10^3	0.999985
3	0.99988	3.62×10^1	0.9996
4	0.999997	8.21×10^{-1}	0.986
5	0.99999993	1.72×10^{-2}	0.622
6	0.999999997	1.02×10^{-3}	0.078
7	0.999999998	6.72×10^{-4}	0.071

Multiple Prototypes

- In principle, need prototypes of slowly converging modes of relaxation
- Difficult to distinguish when there are multiple modes converging at same rate
- Instead, always look for slowest converging mode of current solver
  - It represents exactly the error that this solver is missing
  - Must also be a slowly converging mode of relaxation
- When developed carefully, each prototype will represent distinct slowly converging error types, and all must be accounted for in interpolation

Conclusions

- Efficient multigrid performance can be recovered, even if character of algebraically smooth error is not known
- Needed adaptivity is, however, quite expensive, and so should be performed with care
- Improve each prototype as much as possible, as it is exposed, before introducing more
- Estimates of performance key in making informed decisions about adaptivity
- Many open questions in how best to design adaptive process, answers are often objective-dependent