A Mini-course on Multigrid Method: Introduction

Hardik Kothari

Euler Institute,
Università della Svizzera italiana,
Switzerland

May 16, 2023
Research school on Iterative Methods for Partial Differential Equations
Paris, France
Outline

1. Overview
2. Model problem
3. Basic iterative methods
4. Smoothers
5. Two-grid method
6. Multigrid method
7. Numerical Results
8. Outlook and extensions
Motivation

- Effective in solving large-scale problems from applied mathematics, engineering, physics, computational sciences
- More of a framework rather than a solution method, hence it requires a detailed knowledge of the algorithm for effective usage
- Properties of multigrid method
  - optimal time complexity
  - optimal memory complexity
  - level independent convergence
  - stable asymptotic convergence rates
- Included in libraries for solving numerical solutions of PDEs, e.g., PETSc, Trilinos, etc.
Historical developments

- Highly effective iterative numerical framework for solving linear system of equations arising from discretization of the partial differential equations (PDEs).

- Historical developments
  - Federenko (1964) - multigrid algorithm for standard 5-point stencil for solving Laplace problem
  - Bachvalov (1966) - generalization of the MG for central-difference scheme for linear elliptic PDE with variable smooth coefficients
  - Theoretical work estimates were pessimistic and method was not put in practice
  - First practical results were reported by Brandt, outlined the main principles and demonstrated practical utility of the method (1973, 1977)
  - The method was also discovered independently by Hackbusch (1978, 1980, 1981), who laid mathematical foundations
  - numerous contributions since then...
Main monographs

- Multi-Grid Methods and Applications, - by W. Hackbusch (1985)
- An Introduction to Multigrid Methods, - by P. Wesseling (1992)
- Multigrid Methods for Finite Elements, - by V. V. Shaidurov (1995)
- Multigrid, - by U. Trottenberg, C. W. Oosterlee, A. Schüler (2001)
Chapters in other monographs

- The mathematical theory of Finite Element methods - Chapter 6, - by S. C. Brenner, L. R. Scott (2008)
- Finite Difference and Spectral Methods for Ordinary and Partial differential equations - Chapter 9, - by L. N. Trefethen (1996)
Overview

Multigrid methods

- An effective iterative numerical method for solving linear system of equations arising from discretization of the partial differential equations.
- An optimal solution method
  - convergence rate of the method is independent of problem size
  - iteration till convergence do not grow with increasing problem size
  - only arithmetic operations grow with problem size
- It employs hierarchy of meshes with different resolutions
- Essential components:
  - Smoothers of each level
  - Transfer operators
  - A direct solver on the coarsest level
Model problem

Table of Contents

1 Overview

2 Model problem

3 Basic iterative methods

4 Smoothers

5 Two-grid method

6 Multigrid method

7 Numerical Results

8 Outlook and extensions
A model problem

- **Strong form:**
  \[-\nabla^2 u = f, \quad \text{in } \Omega,\]
  \[u = 0, \quad \text{on } \partial \Omega,\]

  where \(f \in L^2(\Omega)\) regular, \(\partial \Omega\) is Lipschitz

- **Weak form:** Find \(u \in H^1_0(\Omega)\) such that

  \[
  \int_{\Omega} \nabla u \cdot \nabla v \, dx = \int_{\Omega} f v \, dx, \quad \forall \; v \in H^1_0(\Omega),
  \]

  \[a(u, v) = F(v), \quad \forall \; v \in H^1_0(\Omega).\]

- **Triangulation of \(\Omega\) with elements**
- **Choosing basis** \(\{\phi_i\}_{i=1}^N\), where \(N = \text{number of nodes}\)
- **Discretized weak form can be given as a linear system**

  \[Ax = b, \quad \text{where } \begin{cases} 
  A_{ij} = \int_{\Omega} \nabla \phi_i \cdot \nabla \phi_j \, dx \\
  b_i = \int_{\Omega} f \cdot \phi_i \, dx
  \end{cases}\]
Matrix stencil and Eigenvalues/Eigenvectors

- 1D problem

\[ A = \frac{1}{h^2} \begin{bmatrix}
2 & -1 & & & \\
-1 & 2 & -1 & & \\
& -1 & 2 & -1 & \\
& & \ddots & \ddots & \ddots \\
& & & -1 & 2 \\
\end{bmatrix} \]

- \( A \in \mathbb{R}^{n \times n} \) symmetric positive definite

- Diagonally dominant, i.e.,

\[ |a_{ii}| \geq \sum_{j \neq i} |a_{ij}| \quad \text{for } i = 1, 2, \ldots, n \]

- Eigenvalues in range \([0, 4]\)

\[ \lambda_k = 4 \sin^2 \left( \frac{k\pi}{2n} \right) \]

- Eigenvectors are Fourier modes

\[ (v_k)_j = \sin \left( \frac{jk\pi}{n} \right) \]
# Table of Contents

1. Overview
2. Model problem
3. Basic iterative methods
4. Smoothers
5. Two-grid method
6. Multigrid method
7. Numerical Results
8. Outlook and extensions
Iterative methods based on Matrix splitting

- Solve linear system of equation:

\[ Ax = b \]

- Split the matrix \( A \) to construct iterative method

\[ A = M - N \]

\[ Mx = Nx + b \]

- Iterative method can be written as

\[ Mx^{(k+1)} = Nx^{(k)} + b \]

\[ x^{(k+1)} = M^{-1}(Nx^{(k)} + b) \]

\[ x^{(k+1)} = M^{-1}((M - A)x^{(k)} + b) \]

\[ x^{(k+1)} = x^{(k)} + M^{-1}(b - Ax^{(k)}) \]

- Alternatively, we can also write

\[ x^{(k+1)} = (I - M^{-1}A)x^{(k)} + M^{-1}b \]

\[ x^{(k+1)} = Gx^{(k)} + d \]
Basic iterative methods

Fixed point iteration

- Given an initial guess \( x^{(0)} \), an iterative method can be given as

\[
x^{(k+1)} = Gx^{(k)} + d, \quad k = 0, 1, 2, \ldots
\]

where \( G = I - M^{-1}A \) and \( d = M^{-1}b \)

- The solution \( x^* \) of the linear system \( Ax = b \) is a fixed point of above iteration, i.e.,

\[
x^* = Gx^* + d
\]

- Error for a given iterate \( x^{(k)} \)

\[
e^{(k)} = x^* - x^{(k)}
\]

- Residual at a given iterate \( x^{(k)} \)

\[
r^{(k)} = b - Ax^{(k)}
\]

- Error equation

\[
r^{(k)} = b - Ax^{(k)} = Ax^* - Ax^{(k)} = Ae^{(k)}
\]
Error propagation

- Utilizing the fixed point and iterative method

\[ x^{(k+1)} - x^* = G(x^{(k)} - x^*) \]
\[ x^{(k+1)} - x^* = G^2(x^{(k-1)} - x^*) \]
\[ \vdots \]
\[ x^{(k+1)} - x^* = G^{k+1}(x^{(0)} - x^*) \]

- Using the error

\[ e^{(k+1)} = G^{k+1}e^{(0)} \]

**Theorem:**
A fixed-point iterative method is called convergent, if for an arbitrary \( x^{(0)} \), we have \( \lim_{k \to \infty} x^{(k)} = x^* \).
We can equivalently write,

\[ \lim_{k \to \infty} G^k = 0. \]

Iteration error can be bounded by

\[ \|e^{(k)}\| \leq \|G\|^k \|e^{(0)}\| \]
Basic iterative methods

Fundamental theorem of iterative method

For an iteration matrix

\[ G = I - M^{-1}A \]

- Iterative method is convergent if \( \lim_{k \to \infty} G^k = 0 \) if and only if \( \rho(G) < 1 \) or \( ||G|| < 1 \)
- \( \rho(G) \) denotes spectral radius, i.e.,
  \[ \rho(G) = \max\{|\lambda_1(G)|, |\lambda_2(G)|, \ldots, |\lambda_n(G)|\} \]

Speed of convergence:
- Using the bounds on iteration error
  \[ \frac{||e^{(k)}||}{||e^{(0)}||} \leq ||G^k|| \leq ||G||^k \leq 10^{-d} \]
- Using spectral radius
  \[ \rho(G^k) \leq 10^{-d} \]
  \[ \log(\rho(G^k)) \leq \log(10^{-d}) \]
  \[ k \log(\rho(G)) \leq -d \]
  \[ k \geq \frac{d}{-\log(\rho(G))}, \quad \text{where convergence rate} = -\log(\rho(G)) \]
Basic iterative methods

Classical iterative methods

- Richardson method:
  \[ M = \omega I, \quad N = \omega I - A \]
  \[ x^{(k+1)} = x^{(k)} + \omega (b - Ax^{(k)}), \quad k = 0, 1, 2, \ldots \]

- Jacobi method:
  \[ M = D, \quad N = (D - A) \]
  \[ x^{(k+1)} = x^{(k)} + D^{-1} (b - Ax^{(k)}), \quad k = 0, 1, 2, \ldots \]

- Damped-Jacobi method:
  \[ M = \omega D, \quad N = \omega D - A \]
  \[ x^{(k+1)} = x^{(k)} + \omega D^{-1} (b - Ax^{(k)}), \quad k = 0, 1, 2, \ldots \]

- Gauss-Seidel method:
  \[ M = D + L, \quad N = D + L - A \]
  \[ x^{(k+1)} = x^{(k)} + (D + L)^{-1} (b - Ax^{(k)}), \quad k = 0, 1, 2, \ldots \]
Basic iterative methods

Error v/s iterations

Observations:
- Iterative methods are very slow to converge
- Error reduced rapidly in few first iterations, but then it stagnates
Table of Contents

1 Overview
2 Model problem
3 Basic iterative methods
4 Smoothers
5 Two-grid method
6 Multigrid method
7 Numerical Results
8 Outlook and extensions
Eigenvectors of $A$

$$(v_k)_j = \sin\left( \frac{jk\pi}{n} \right)$$
Convergence of different error components

damped-Jacobi method with $\omega = \frac{2}{3}$

Error reduces rapidly for high frequency component
Convergence analysis for damped Jacobi method

- Iteration matrix

\[ G_{\omega J} = I - \omega D^{-1}A \]

- Eigenvalues of the iteration matrix

\[ \lambda(G_{\omega J}) = 1 - \frac{\omega}{2} \lambda(A) \]

- Recall: \( \lambda_i(A) = 4 \sin^2 \left( \frac{i\pi}{2n} \right) \)

\[ \lambda_i(G_{\omega J}) = 1 - \frac{\omega}{2} 4 \sin^2 \left( \frac{i\pi}{2n} \right) \]

- Convergence rate:

\[ \rho(G_{\omega J}) = \lambda_1 = 1 - 2\omega \sin^2 \left( \frac{h\pi}{2} \right) = 1 - O(h^2) < 1 \]

- Eigenvectors of \( G \) and \( A \) are same for Jacobi/damped-Jacobi method, which is not true for Gauss-Seidel method, as Gauss-Seidel mixes the eigenvectors of \( A \)
Expanding on eigenvalues of iteration matrix

- Expanding initial error using eigenvectors
  \[ e^{(0)} = \sum_{i=1}^{n} c_i v_i \]

- After \( \nu \) iterations
  \[ e^{(\nu)} = Ge^{(\nu-1)} \]

- Error equation for damped Jacobi iteration
  \[ e^{(\nu)} = (I - \omega D^{-1}A)e^{(\nu-1)} = G_{\omega J}^{\nu} e^{(0)} \]

- For \( \nu \) iterations, we get
  \[ G_{\omega J}^{\nu} e^{(0)} = \sum_{i=1}^{n} c_i G_{\omega J}^{\nu} v_i = \sum_{i=1}^{n} c_i \lambda_i^{\nu} v_i \]

- The \( i^{th} \) mode of the error is reduced by magnitude of \( \lambda_i \) at each iteration
Relaxation smooths different eigenmodes

\[ \lambda_k(G_{\omega J}) = 1 - 2\omega \sin^2 \left( \frac{i \pi}{2n} \right) \]

- Eigenvalues of iteration matrix: \( \lambda_i(G_{\omega J}) = 1 - 2\omega \sin^2 \left( \frac{i \pi}{2n} \right) \)
- Which value of \( \omega \) provides best smoothing of high frequency error?
Smoothers

Smoothing factor
The smoothing factor of a relaxation method is the largest absolute value of the eigenvalues in the upper half of the spectrum of the iteration matrix:

$$\text{smoothing factor} = \max\{ |\lambda_{n/2}(G)|, |\lambda_{n/2+1}(G)|, |\lambda_{n/2+2}(G)|, \ldots, |\lambda_n(G)| \}$$

Smoothing factors for damped-Jacobi method:

- for $\omega = 2/3$
  $$|\lambda_{n/2}(G_{\omega J})| = |\lambda_n(G_{\omega J})| = \frac{1}{3}$$

- for $\omega = 1/2$
  $$|\lambda_{n/2}(G_{\omega J})| = \frac{1}{2}, \quad |\lambda_n(G_{\omega J})| = 0$$

- for $\omega = 1/3$
  $$|\lambda_{n/2}(G_{\omega J})| = \frac{2}{3}, \quad |\lambda_n(G_{\omega J})| = \frac{1}{3}$$

- for $\omega = 1$
  $$|\lambda_{n/2}(G_{\omega J})| = 0 \quad |\lambda_n(G_{\omega J})| = 1$$

- Jacobi is not a smoother, but damped-Jacobi is a smoother
Smoothing error for Jacobi v/s damped-Jacobi methods

iteration = 1
Smoothers

Smoothing error for Jacobi v/s damped-Jacobi methods

\[ \text{iteration} = 2 \]

Jacobi (\( \omega = 0 \))

Jacobi (\( \omega = \frac{2}{3} \))
Smoothers

**Smoothing error for Jacobi v/s damped-Jacobi methods**

For Jacobi methods:
- Jacobi ($\omega = 0$)
- Jacobi ($\omega = \frac{2}{3}$)

Graphs showing the smoothing error for different iterations and values of $\omega$.
Smoothing error for Jacobi v/s damped-Jacobi methods

Iteration = 4

Jacobi (\(\omega = 0\))

Jacobi (\(\omega = \frac{2}{3}\))
Smoothers

Smoothing error for Jacobi v/s damped-Jacobi methods

Jacobi ($\omega = 0$)

iteration = 5

Jacobi ($\omega = \frac{2}{3}$)

H. Kothari

A Mini-course on Multigrid Method: Introduction
Smoothers

Smoothing error for Jacobi v/s damped-Jacobi methods

iteration = 6
Smoothers

Smoothing error for Jacobi v/s damped-Jacobi methods

iteration = 7
Smoothers

Smoothing error for Jacobi v/s damped-Jacobi methods

iteration = 8

Jacobi ($\omega = 0$)

Jacobi ($\omega = 2/3$)
Smoothers

Smoothing error for Jacobi v/s damped-Jacobi methods

iteration = 9

Jacobi (ω = 0)

Jacobi (ω = 2/3)
Smoothers

Smoothing error for Jacobi v/s damped-Jacobi methods

Jacobi ($\omega = 0$)  
iteration = 10  
Jacobi ($\omega = \frac{2}{3}$)

H. Kothari  
A Mini-course on Multigrid Method: Introduction
Smoothers

Smoothing error for Jacobi v/s damped-Jacobi methods

iteration = 50
Towards projection method

Observation:

- Classical iterative schemes have **smoothing property**, i.e., the oscillatory modes of the error are damped very quickly, the smooth modes are damped very slowly.

Idea:

- Do $\nu$ smoothing iterations to improve on the initial guess:
  \[ x^{(k)} = S^\nu(x^{(k-1)}) \]

- Look for an update $\delta x$ that provides the "best" improvement for iterate $x^* = x^{(k)} + \delta x$.

- Construct $\delta x \approx e^{(k)}$ by least square minimization on a smaller space $V$
  \[ \min_{\delta x \in \text{span}\{V\}} \|\delta x - e^{(k)}\| \]

- Then $\delta x = Vw$

- Update from the projection method:
  \[ x^{(k+1)} = x^{(k)} + V(V^TV)^{-1}V^Te^{(k)} \]
Smoothers

**A-orthogonal projection**

- Instead of standard Euclidean norm, we can employ $A$-norm
- Thus employing $A$-orthogonal projection onto the range of $V$
- Construct $\delta x \approx e^{(k)}$ by minimization in $A$-norm

$$
\min_{\delta x \in \text{span}\{V\}} \| \delta x - e^{(k)} \|_A
$$

- Then $\delta x = Vw$
  $$V^T AVw = V^T A e^{(k)}$$

- Update from the projection method
  $$
x^{(k+1)} = x^{(k)} + V(V^T AV)^{-1} V^T A e^{(k)}
$$
  $$
x^{(k+1)} = x^{(k)} + V(V^T AV)^{-1} V^T r^{(k)}
$$
Table of Contents

1. Overview
2. Model problem
3. Basic iterative methods
4. Smoothers
5. Two-grid method
6. Multigrid method
7. Numerical Results
8. Outlook and extensions
Two-grid method

Coarse grids

- Smooth error modes on a fine grid appear oscillatory on a coarse grid
- Thus, smoothing iteration on coarse grid can damp the oscillatory errors
- Relaxation on coarse grid is much cheaper, as the mesh size grows with the coarsening factor of $\frac{1}{2}$
- Relaxation on the coarse grid has better convergence rate
  \[ \rho(G) = 1 - O(4h^2) \]
- The A-orthogonal projection property ensure that for a given subspace $V$, we get optimal coarse-grid correction
Two-grid method

Smooth error on coarse grid

- Smooth modes appear oscillatory on coarse grid
Coarse modes

- For $k \in [1, n/2]$ the $k^{th}$ mode is preserved on the coarse grid

$$ (v^h_k)_{2j} = \sin \frac{jk\pi}{n/2} = \sin \frac{2jk\pi}{n} = (v^h_k)_j $$

- For $k \in (n/2, n]$ the $k^{th}$ mode is invisible on the coarse grid (Aliasing)

$$ (v^h_k)_{2j} = \sin \frac{2jk\pi}{n} = -\sin \frac{2j(n-k)\pi}{n} $$

$$ = -\sin \frac{j(n-k)\pi}{n/2} = -(v^{2h}_{n-k})_j $$

- It is necessary to damp the oscillating error modes on fine grid before a problem on coarse grid is considered. Otherwise, one would get additional smooth error modes on the coarser grid.
Incorporating coarse grid

Recall: Projection iteration

\[ x^{(k+1)} = x^{(k)} + V(V^T AV)^{-1} V^T r^{(k)} \]

Questions:

- How to approximate \( V \) to transfer information between a fine level and a coarse level?
- How to solve the problem on coarse space?
- Which problem do we solve on coarse grid?
Prolongation operator

- Construct a prolongation operator $P : \Omega_H \to \Omega_h$
- We employ linear interpolation as a prolongation operator: the simplest approach!

```
1 1/2 1/2 1 1/2 1/2 1 1/2 1/2 1 1/2 1/2 1 1/2 1/2 1
```

- Prolongation gives the best results, if the error on the fine grid is smooth
- Hence, it is an appropriate complement to the smoother which works most efficiently if the error is oscillating

Prolongation operator

$$P = \begin{bmatrix} 1 & 1/2 & 1/2 \\ 1/2 & 1 & 1/2 \\ & 1/2 & 1/2 \\ & & & \ddots \\ & & & & 1/2 \\ & & & & 1 \end{bmatrix}$$
Two-grid method

Restriction operators

- Construct a restriction operator \( R : \Omega_h \rightarrow \Omega_H \)
- Injection is the simplest restriction

Not an efficient approach, as the errors on the fine node is not corrected on the coarse level

Weighted restriction: \( R = cP^\top \) (used restrict primal quantities)

In FE framework \( R = P^\top \) (used to restrict residuals/dual quantities)
Further observations on transfer operators

- Columns of interpolation matrix represent basis functions
- Thus the basis functions on coarse level are constructed as linear combination of the basis functions from the finest level
- Hierarchy of nested spaces
- Full rank for of the transfer operator
- The constant functions are preserved when going to the fine level
Two-grid method

Two-level method

Solve: $A_h x_h = b_h$

Coarse grid correction scheme:

1. Perform $\nu_{pre}$ pre-smoothing steps:
   \[ x_h \leftarrow x_h + S^\nu_{pre} (b_h - A_h x_h) \]
2. Restrict residual to the coarse grid:
   \[ r_H \leftarrow R(b - A_h x_h) \]
3. Solve coarse grid equation:
   \[ A_H e_H = r_H \]
4. Interpolate correction to the fine grid:
   \[ x_h \leftarrow x_h + P e_H \]
5. Perform $\nu_{post}$ post-smoothing steps:
   \[ x_h \leftarrow x_h + S^\nu_{post} (b_h - A_h x_h) \]
Two-grid method

Coarse grid matrix

Question: How to construct $A_H$?

- A straightforward approach consists of constructing $A_H$ by discretizing the PDE on $\Omega_H$
- Galerkin projection approach can be given as:

$$A_h e_h = r_h$$
$$A_h P e_H = r_h$$
$$R A_h P e_H = R r_h$$
$$A_H e_H = r_H$$

- Galerkin projection based on the assumption that the error $e_h$ is in the range of the prolongation. This property is in general not given.
- For simple cases, $A_H$ constructed using rediscretization and Galerkin projection coincides
- Although, it does not hold in general
Two-grid method

Analysis of two-grid method

- Recall iterative method
  \[ x_h \leftarrow (I - M^{-1}A)x_h + M^{-1}b_h \]

- Two grid iteration process
  \[ I - M_{TG}^{-1}A_h = (I - M^{-T}A_h)^{\nu_{post}} (I - P(P^TA_hP)^{-1}P^TA_h)(I - M^{-1}A_h)^{\nu_{pre}} \]

- Error propagation
  \[ e \leftarrow (I - M_{TG}^{-1}A)^k e \]

- The two-grid method converges, if and only if
  \[ \rho(I - M_{TG}^{-1}A) < 1 \]
Two-grid method

Analysis of two-grid method

- Two-grid iteration matrix without post-smoothing steps:
  \[ G_{TG} = I - M_{TG}^{-1}A_h = (I - P(P^T A_h P)^{-1}P^T A_h)(I - M^{-1}A_h)'' \]

\[
\|G_{TG}\|_2 = \|(I - P(P^T A_h P)^{-1}P^T A_h)G''_s\|_2 \\
= \|(I - P(P^T A P)^{-1}P^T A)A_h^{-1}A_h G''_s\|_2 \\
\leq \|A_h^{-1} - P(P^T A_h P)^{-1}P^T)\|_2 \|A_h G''_s\|_2 \\
= \underbrace{\|A_h^{-1} - PA_c^{-1}P^T\|_2}_{\text{approximation property}} \underbrace{\|A_h G''_s\|_2}_{\text{smoothing property}}
\]

- Smoothing property and approximation property
  \[ \|A_h G''_s\|_2 \leq \frac{c}{\nu} h^{-2} \quad \|A_h^{-1} - PA_c^{-1}P^T\|_2 \leq ch^2 \]

- Contraction property for sufficiently large \( \nu \)
  \[ \|G_{TG}\|_2 \leq \frac{c}{\nu} < 1 \]
# Table of Contents

1. **Overview**
2. **Model problem**
3. **Basic iterative methods**
4. **Smoothers**
5. **Two-grid method**
6. **Multigrid method**
7. **Numerical Results**
8. **Outlook and extensions**
Sketch of multigrid method

\[
x_2 \leftarrow S_{2}^{\nu_1, \nu_2}(f_2 - A_2 x_2)
\]

\[
c_1 \leftarrow 0
\]

\[
c_1 \leftarrow S_{1}^{\nu_1, \nu_2}(r_1 - A_1 c_1)
\]

\[
r_2 \leftarrow f_2 - A_2 x_2
\]

\[
r_1 \leftarrow (\Pi_1^2)^T r_2
\]

\[
r_0 \leftarrow (\Pi_0^1)^T (r_1 - A_1 c_1)
\]

\[
c_1 \leftarrow c_1 + \Pi_0^1 c_0
\]

\[
c_0 \leftarrow A_0^{-1} r_0
\]

\[
u_2 \leftarrow x_2 + \Pi_1^2 c_2
\]
Type of multigrid cycles

- **V-cycle**
- **W-cycle**
- **F-cycle**

[Diagram showing the three types of multigrid cycles: V-cycle, W-cycle, and F-cycle.]
Multigrid extension

- **V-cycle**
  - Recursive formulation of the two-grid method

- **W-cycle**
  - Useful when coarse grid is not optimal
  - Realized by calling V-cycle twice on each level

- **F-cycle**
  - Requires discretization on grid
  - Motivated by nested iteration
  - Eliminates errors from coarse to fine level
# Numerical Results

## Table of Contents

1. Overview
2. Model problem
3. Basic iterative methods
4. Smoothers
5. Two-grid method
6. Multigrid method
7. Numerical Results
8. Outlook and extensions
Numerical Results

Numerical experiments

Poisson problem
### Numerical Results

#### Comparing various MG-cycles

<table>
<thead>
<tr>
<th>mesh size</th>
<th># levels</th>
<th>V(3,3)</th>
<th>V(5,5)</th>
<th>W(3,3)</th>
<th>W(5,5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>20 × 20</td>
<td>2</td>
<td>7</td>
<td>6</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>40 × 40</td>
<td>3</td>
<td>8</td>
<td>8</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>80 × 80</td>
<td>4</td>
<td>9</td>
<td>8</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>160 × 160</td>
<td>5</td>
<td>10</td>
<td>9</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>320 × 320</td>
<td>6</td>
<td>10</td>
<td>9</td>
<td>7</td>
<td>6</td>
</tr>
</tbody>
</table>

#### # Iterations

<table>
<thead>
<tr>
<th>mesh size</th>
<th># levels</th>
<th>V(3,3)</th>
<th>V(5,5)</th>
<th>W(3,3)</th>
<th>W(5,5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>20 × 20</td>
<td>2</td>
<td>0.0139</td>
<td>0.0057</td>
<td>0.0139</td>
<td>0.0057</td>
</tr>
<tr>
<td>40 × 40</td>
<td>3</td>
<td>0.0372</td>
<td>0.0185</td>
<td>0.0182</td>
<td>0.0093</td>
</tr>
<tr>
<td>80 × 80</td>
<td>4</td>
<td>0.0460</td>
<td>0.0276</td>
<td>0.0184</td>
<td>0.0106</td>
</tr>
<tr>
<td>160 × 160</td>
<td>5</td>
<td>0.0467</td>
<td>0.0299</td>
<td>0.0186</td>
<td>0.0108</td>
</tr>
<tr>
<td>320 × 320</td>
<td>6</td>
<td>0.0551</td>
<td>0.0312</td>
<td>0.0179</td>
<td>0.0108</td>
</tr>
</tbody>
</table>

Asymptotic convergence rate \( \rho^* = \frac{\|x^{(k+1)} - x^{(k)}\|_A}{\|x^{(k)} - x^{(k-1)}\|_A} \)
Convergence of V(3,3) cycle
Numerical Results

Convergence of V(5,5) cycle

![Graph showing convergence of V(5,5) cycle](image)
Numerical Results

Convergence of W(3,3) cycle
Numerical Results

Convergence of W(5,5) cycle
Outlook and extensions

Table of Contents

1 Overview
2 Model problem
3 Basic iterative methods
4 Smoothers
5 Two-grid method
6 Multigrid method
7 Numerical Results
8 Outlook and extensions
Outlook and extensions

Still many topics remain

Algebraic multigrid methods

- Aims to construct better transfer operators
- When mesh is not available
- Utilizes only graph information from matrix
- Compute strength between the edges of graph

AMG libraries:

- Hypre - BoomerAMG: http://acts.nersc.gov/hypre/
- Trilinos - muLU: https://trilinos.org/
- PyAMG: http://pyamg.org/
- PETSc: https://petsc.org
Questions?

Contact: hardik.kothari@usi.ch