Introduction to Multigrid Methods

Jaap van der Vegt

Numerical Analysis and Computational Science Group
Department of Applied Mathematics
Universiteit Twente

Enschede, The Netherlands

Part 1.

USTC Summer School, August 29 – September 2, 2022

Introduction

- Multigrid methods are very efficient iterative solvers for large systems of linear and nonlinear algebraic equations.
- Optimal multigrid methods can solve linear systems in $O(N)$ number of operations, with N the number of unknowns.
- The main focus in this course will be on multigrid methods suitable for finite difference, finite element, finite volume and spectral discretizations of partial differential equations.

Introduction

- The two main types of multigrid methods are:
  - *h*-Multigrid methods are well suited for partial differential equations using a computational mesh.

    h-Multigrid methods require specific optimizations based on the type of partial differential equations under consideration to obtain optimal convergence rates.

    h-Multigrid methods are suitable to many types of partial differential equations on complicated domains and various types of boundary conditions.

  - Algebraic multigrid methods are well suited as black box multigrid solver, but will generally not reach optimal convergence rates for discretizations of partial differential equations.

    Algebraic methods are very useful if one does not have the time or interest to develop and implement a dedicated multigrid solver.

Introduction

- In this course we will discuss the basis aspects of multigrid aspects.

  The main topics in this course are:

  - What are the basic principles behind a multigrid method and why can multigrid methods converge in $O(N)$ number of operations.
  - What are the main components of a multigrid method.
  - How to theoretically analyze and optimize multigrid performance and to choose suitable parameters in a multigrid algorithm.
  - The focus in these lectures will be on h-multigrid methods.

Introduction

- The course will be at the introductory level. More information can be found in
  - U. Trottenberg, C. Oosterlee, A. Schüller, Multigrid, Elsevier 2001.
  - W.L. Briggs, Van Emden-Henson, S.F. McCormick, A multigrid tutorial, 2nd ed., SIAM 2000.

Introduction

Multigrid methods for Partial Differential Equations

- The performance and complexity of multigrid methods depends on the type of partial differential equation.

- Consider the second order scalar partial differential equation

$$Lu = f \qquad \text{in } \Omega \subseteq \mathbb{R}^n,$$

where

$$Lu = a_{11}u_{xx} + a_{12}u_{xy} + a_{22}u_{yy} + a_1 u_x + a_2 u_y + a_0 u$$

with coefficients a_{ij}, a_i, f depending in general on x, y, u, u_x, u_y.

Introduction

Classical types of linear second order Partial Differential Equations for

$$Lu = a_{11}u_{xx} + a_{12}u_{xy} + a_{22}u_{yy} + a_1 u_x + a_2 u_y + a_0 u$$

with coefficients only depending on (x, y)

- Elliptic if $4a_{11}a_{22} - a_{12}^2 > 0$.
- Hyperbolic if $4a_{11}a_{22} - a_{12}^2 < 0$.
- Parabolic if $4a_{11}a_{22} - a_{12}^2 = 0$.

Examples

- Poisson equation

$$-\Delta u := -u_{xx} - u_{yy} = f$$

- Wave equation

$$u_{xx} - u_{yy} = f$$

- Heat equation

$$u_{xx} - u_{y} = f$$

Introduction

More complicated cases

- Anisotropic model equation

$$-\epsilon u_{xx} - u_{yy} = f \qquad \text{with } \epsilon << 1$$

- Convection-diffusion equation

$$-\epsilon \Delta u + a_1 u_x + a_2 u_y = f \qquad \text{with } \epsilon << 1$$

Introduction

- We will only consider steady problems.
- For time-dependent problems (semi)-implicit methods also yield a system of algebraic equations that can be solved using multigrid methods.

Meshes

The discretization of a PDE requires a mesh (or grid).

Types of grids

- Cartesian grids
- Boundary fitted (block structured) grids
- Unstructured grid (can contain different types of elements).

Notation

- Discrete boundary value problems

$$L_h^\Omega u_h(x,y) = f_h^\Omega(x,y), \qquad (x,y) \in \Omega_h,$$
$$L_h^\Gamma u_h(x,y) = f_h^\Gamma(x,y), \qquad (x,y) \in \Omega_h,$$

  with h a mesh parameter.

- Infinite grid

$$G_h := \{(x,y) \ : \ x = x_i = ih_x, y = y_j = jh_y, i,j \in \mathbb{Z}\}$$

  with h_x, h_y the mesh size in respectively the x- and y-direction.

- Computational grid $\Omega_h := \Omega \cap G_h$.

  The variable u_h is defined on Ω_h for example as

$$u_h(x,y) = u_h(x_i,y_j) = u_h(ih_x, jh_y) \ \text{ or } \ u_h = u_{i,j}$$

Notation

- L_h^Ω, L_h^Γ are grid operators approximating L^Ω and L^Γ.
- L_h^Ω, L_h^Γ can be obtained with e.g. a finite difference, finite volume, finite element or spectral method.
- Discrete equations

$$L_h u_h = f_h \qquad \text{on } \Omega_h$$

$$L_h : \mathcal{G}(\Omega_h) \to \mathcal{G}(\Omega_h)$$

with $\mathcal{G}(\Omega_h)$ the linear space of grid functions on Ω_h.

Inner Products and Norms

- Euclidian inner product

$$\langle u_h, v_h \rangle_2 := \frac{1}{\sharp \Omega_h} \sum_{x \in \Omega_h} u_h(x) \overline{v_h(x)}$$

  with $\sharp \Omega_h$ the number of grid points in Ω_h.

- Euclidian norm

$$\|u_h\|_2 := \sqrt{\langle u_h, v_h \rangle_2}$$

- Infinity norm

$$\|u_h\|_\infty := \max\{|u_h(x)| \ : \ x \in \Omega_h\}$$

Inner Products and Norms

- Spectral norm

$$\|B_h\|_S := \sqrt{\rho(B_h B_h^*)}$$

with $B_h : \mathcal{G}(\Omega_h) \to \mathcal{G}(\Omega_h)$ any linear operator, with conjugate transposed B_h^* and spectral radius ρ.

- If L_h is symmetric and positive definite, we can define the energy inner product

$$\langle u_h, v_h \rangle_E := \langle L_h u_h, v_h \rangle_2$$

with energy norm

$$\|B_h\|_E := \|L_h^{\frac{1}{2}} B_h L_h^{-\frac{1}{2}}\|_S$$

$$= \sqrt{\rho(L_h B_h L_h^{-1} B_h^*)}$$

Relation between Energy Norm and Spectral Radius

- Note

$$\|B_h\|_E = \|L_h^{\frac{1}{2}} B_h L_h^{-\frac{1}{2}}\|_S$$

$$= \sqrt{\rho((L_h^{\frac{1}{2}} B_h L_h^{-\frac{1}{2}})(L_h^{\frac{1}{2}} B_h L_h^{-\frac{1}{2}})^*)}$$

Use

$$(L_h^{\frac{1}{2}} B_h L_h^{-\frac{1}{2}})(L_h^{\frac{1}{2}} B_h L_h^{-\frac{1}{2}})^* = L_h^{\frac{1}{2}} B_h L_h^{-\frac{1}{2}} L_h^{-\frac{1}{2}} B_h^* L_h^{\frac{1}{2}} \quad \text{since } L_h^* = L_h$$

$$= L_h^{\frac{1}{2}} B_h L_h^{-1} B_h^* L_h^{\frac{1}{2}}$$

The spectral radius ρ is the maximum modulus of the eigenvalues of $L_h^{\frac{1}{2}} B_h L_h^{-1} B_h^* L_h^{\frac{1}{2}}$.

Relation between Energy Norm and Spectral Radius

- Consider

$$L_h^{\frac{1}{2}} B_h L_h^{-1} B_h^* L_h^{\frac{1}{2}} x = \lambda x \tag{1}$$

Define $y = L_h^{\frac{1}{2}} x$, hence $x = L_h^{-\frac{1}{2}} y$, then (1) is equal to

$$L_h^{\frac{1}{2}} B_h L_h^{-1} B_h^* y = \lambda L_h^{-\frac{1}{2}} y$$

which is equivalent with

$$L_h B_h L_h^{-1} B_h^* y = \lambda y$$

Thus

$$\rho(L_h^{\frac{1}{2}} B_h L_h^{-1} B_h^* L_h^{\frac{1}{2}}) = \rho(L_h B_h L_h^{-1} B_h^*)$$

hence

$$\|B_h\|_E = \sqrt{\rho(L_h B_h L_h^{-1} B_h^*)}$$

Stencil Notation

- On G_h with a Cartesian or logically Cartesian structure, we consider the grid functions

$$w_h : G_h \to \mathbb{R} \qquad (\text{or } \mathbb{C})$$

$$(x, y) \mapsto w_h(x, y)$$

A general stencil $\left[s_{k_1,k_2}\right]_h$ is given by

$$\left[s_{k_1,k_2}\right]_h = \begin{bmatrix} & \vdots & \vdots & \vdots & \\ \cdots & s_{-1,1} & s_{0,1} & s_{1,1} & \cdots \\ \cdots & s_{-1,0} & s_{0,0} & s_{1,0} & \cdots \\ \cdots & s_{-1,-1} & s_{0,-1} & s_{1,-1} & \cdots \\ & \vdots & \vdots & \vdots & \end{bmatrix}, \qquad s_{k_1,k_2} \in \mathbb{R}$$

which defines an operator on the set of grid functions by

$$\left[s_{k_1,k_2}\right]_h w_h(x, y) = \sum_{(k_1,k_2)} s_{k_1,k_2}\, w_h(x + k_1 h_x, y + k_2 h_y)$$

Stencil Notation

Examples

- Five point stencil

$$\begin{bmatrix} & s_{0,1} & \\ s_{-1,0} & s_{0,0} & s_{1,0} \\ & s_{0,-1} & \end{bmatrix}$$

- Compact nine point stencil

$$\begin{bmatrix} s_{-1,1} & s_{0,1} & s_{1,1} \\ s_{-1,0} & s_{0,0} & s_{1,0} \\ s_{-1,-1} & s_{0,-1} & s_{1,-1} \end{bmatrix}$$

- Stencil notation is very useful to investigate the properties of a multigrid algorithm using Fourier analysis.

Example

- Model problem: Poisson equation

$$-\Delta u(x,y) = f^{\Omega}(x,y)\,, \qquad (x,y) \in \Omega = (0,1)^2 \subset \mathbb{R}^n$$

$$u(x,y) = f^{\Gamma}(x,y), \qquad (x,y) \in \Gamma = \partial\Omega,$$

with $\Delta u = u_{xx} + u_{yy}$.

Example

- Second order accurate finite difference discretization Poisson equation

$$Lu - L_h u = O(h^2) \qquad \text{as } h \to 0$$

with

$$\begin{aligned} L_h u_h(x,y) &= -\Delta_h u_h(x,y) \\ &= \frac{1}{h^2}\big(4u_h(x,y) - u_h(x-h,y) - u_h(x+h,y) \\ &\quad - u_h(x,y-h) - u_h(x,y+h)\big) \\ &= \frac{1}{h^2}\begin{bmatrix} & -1 & \\ -1 & 4 & -1 \\ & -1 & \end{bmatrix}_h u_h(x,y) \end{aligned}$$

Near the domain boundary the stencil has to be adjusted.

Matrix Terminology

- The discrete operator L_h for linear problems can be represented by matrices using various orderings of the mesh points.

- Lexicographic ordering

$$\begin{array}{rrrr} 13 & 14 & 15 & 16 \\ 9 & 10 & 11 & 12 \\ 5 & 6 & 7 & 8 \\ 1 & 2 & 3 & 4 \end{array}$$

- Red-black ordering

$$\begin{array}{rrrr} 15 & 7 & 16 & 8 \\ 5 & 13 & 6 & 14 \\ 11 & 3 & 12 & 4 \\ 1 & 9 & 2 & 10 \end{array}$$

Discretization Poisson Equation I

- Matrix for the Poisson equation with homogeneous Dirichlet boundary condition on a 4×4 mesh with lexicographic ordering

$$A_h = \frac{1}{h^2} \begin{pmatrix} T & -I & & \\ -I & T & -I & \\ & -I & T & -I \\ & & -I & T \end{pmatrix} \in \mathbb{R}^{16 \times 16}$$

with

$$T = \begin{pmatrix} 4 & -1 & & \\ -1 & 4 & -1 & \\ & -1 & 4 & -1 \\ & & -1 & 4 \end{pmatrix} \in \mathbb{R}^{4 \times 4} \qquad I = \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{pmatrix} \in \mathbb{R}^{4 \times 4}$$

Discretization Poisson Equation II

- Matrix for the Poisson equation with homogeneous Dirichlet boundary condition on a 4×4 mesh with red-black ordering

$$A_h = \begin{pmatrix} A_{rr} & A_{rb} \\ A_{br} & A_{bb} \end{pmatrix}$$

with

$$A_{rb} = \frac{1}{h^2} \begin{pmatrix} -1 & 0 & -1 & & & & & \\ -1 & -1 & 0 & -1 & & & & \\ -1 & 0 & -1 & -1 & -1 & & & \\ & -1 & 0 & -1 & 0 & -1 & & \\ & & -1 & 0 & -1 & 0 & -1 & \\ & & & -1 & -1 & -1 & 0 & -1 \\ & & & & -1 & 0 & -1 & -1 \\ & & & & & -1 & 0 & -1 \end{pmatrix}$$

$$A_{rb} = (A_{br})^T$$

$$A_{rr} = A_{bb} = \frac{4}{h^2} \mathrm{diag}(1, 1, 1, 1, 1, 1, 1, 1)$$

Discretization Poisson Equation II

- For the multigrid algorithm it is generally not necessary to build the matrix A_h.
- Most of the operations in the multigrid method are "local" operations, such as multiplications and additions carried out grid point by grid point.

  Generally, only the solution vectors, defects ($d_h^m = f_h - L_h u_h^m$) and right-hand side on all grid levels need to be stored.

Solvers for Poisson Equation

Overview of solvers for the Poisson equation

Gaussian elimination (band version)	$O(N^2)$
Jacobi iteration	$O(N^2 \log(\epsilon))$
Gauss-Seidel iteration	$O(N^2 \log(\epsilon))$
Successive Overrelation (SOR)	$O(N^{\frac{3}{2}} \log(\epsilon))$
Conjugate Gradient (CG)	$O(N^{\frac{3}{2}} \log(\epsilon))$
Alternating Direction Implicit (ADI)	$O(N \log N \log(\epsilon))$
Fast Fourier Transform (FFT)	$O(N \log(N))$
Multigrid (iterative)	$O(N \log(N))$
Full Multigrid (FMG)	$O(N)$

Here ϵ depends on the accuracy of the stopping criteria for the iteration.

A First Glance at Multigrid

- Solve the Poisson equation on a grid with lexicographical ordering with Gauss-Seidel iteration

$$u_h^{m+1}(x_i, y_j) = \frac{1}{4}(h^2 f_h(x_i, y_j) + u_h^{m+1}(x_i - h, y_j) + u_h^m(x_i + h, y_j) + u_h^{m+1}(x_i, y_j - h) + u_h^m(x_i, y_j + h))$$

where $(x_i, y_j) \in \Omega_h$ or equivalently

$$u_{i,j}^{m+1} = \frac{1}{4}(h^2 f_{ij} + u_{i-1,j}^{m+1} + u_{i+1,j}^m + u_{i,j-1}^{m+1} + u_{i,j+1}^m)$$

Here u_h^m and u_h^{m+1} are the approximation of $u_h(x_i, y_j)$ before and after a Gauss-Seidel iteration.

A First Glance at Multigrid

- Define the iteration error v_h^m as

$$v_h^m(x_i, y_j) = u_h(x_i, y_j) - u_h^m(x_i, y_j)$$

- For the Poisson equation this gives

$$v_h^{m+1}(x_i, y_j) = \frac{1}{4}\left(v_h^{m+1}(x_i - h, y_j) + v_h^m(x_i + h, y_j) + v_h^{m+1}(x_i, y_j - h) + v_h^m(x_i, y_j + h)\right)$$

Note, this resembles an error averaging process.

A First Glance at Multigrid

Two basic principles of a multigrid algorithm

- Smoothing Principle

  Many classical iterative methods (e.g. Gauss-Seidel, but not Jacobi) have a strong smoothing effect.

- Coarse Grid Principle

  A smooth error term is well approximated on a coarse grid, which is computationally substantially less expensive than a fine grid.

A First Glance at Multigrid

Heuristic Fourier analysis (Laplace equation with homogeneous Dirichlet bc.)

- Expand the error in a Fourier series

$$v_h(x,y) = \sum_{k,l=1}^{n-1} \alpha_{k,l} \sin k\pi x \sin l\pi y$$

- Define for $(x,y) \in \Omega_h$ the functions

$$\phi_h^{k,l}(x,y) = \sin k\pi x \sin l\pi y \qquad (k,l = 1,\cdots,n-1)$$

- The $\phi_h^{k,l}(x,y)$ are the discrete eigenfunctions of the discrete operator L_h and satisfy

$$L_h \phi_h^{k,l}(x,y) = \lambda \phi_h^{k,l}(x,y)$$

A First Glance at Multigrid

- The smoothing property of the iterative solver implies that the high frequency components of the error, namely

$$\alpha_{k,l} \sin k\pi x \sin l\pi y \qquad \text{with } k \text{ or } l \text{ large,}$$

  become small after a few iterations.

- The low frequency components of the error hardly change after a few iterations since they are well represented on the mesh.

- The distinction between high and low frequencies is important in the multigrid context and a crucial part of the algorithm.

High and Low Frequencies and Coarse Meshes

- Consider the finite difference approximation of the Poisson equation with homogeneous Dirichlet boundary condition on $\Omega = (0,1)^2$.
- Assume for the fine mesh that $h = \frac{1}{n}$ and use a square mesh with $n \times n$ grid points.
- Assume for the coarse mesh that $H = 2h$.

High and Low Frequencies and Coarse Meshes

- Consider the eigenfunctions $\phi^{k,l}(x,y) = \sin(k\pi x)\sin(l\pi y)$ on Ω_{2h} with $k, l = 1, \cdots, n-1$.
- Observe that for $(x,y) \in \Omega_{2h}$ implies that $x = \frac{2jh}{nh}$, $y = \frac{2mh}{nh}$ with $j, m = 1, \cdots \frac{n}{2}$, then

$$
\begin{aligned}
\phi^{k,l}(x,y) &= \sin k\pi x \sin l\pi y \\
&= \sin(\frac{2\pi kjh}{nh}) \sin(\frac{2\pi mlh}{nh}) \\
&= \sin(\frac{2\pi kj}{n}) \sin(\frac{2\pi ml}{n}) \\
&= -\sin(2\pi j - \frac{2\pi jk}{n}) \sin(\frac{2\pi ml}{n}) \\
&= -\sin((n-k)\frac{2\pi j}{n}) \sin(\frac{2\pi ml}{n}) \\
&= -\sin((n-k)\pi x) \sin(l\pi y) \\
&= -\phi^{n-k,l}(x,y) \qquad\qquad (x,y) \in \Omega_{2h}
\end{aligned}
$$

High and Low Frequencies and Coarse Meshes

- On the coarse mesh with $(x, y) \in \Omega_{2h}$ we obtain then

$$\begin{aligned}
\phi^{k,l}(x,y) &= \sin(k\pi x)\sin(l\pi y) \\
&= -\sin((n-k)\pi x)\sin(l\pi y) = -\phi^{n-k,l}(x,y) \\
&= -\sin(k\pi x)\sin((n-l)\pi y) = -\phi^{k,n-l}(x,y) \\
&= \sin((n-k)\pi x)\sin((n-l)\pi y) = \phi^{n-k,n-l}(x,y) \qquad (x,y) \in \Omega_{2h}
\end{aligned}$$

- On the coarse mesh Ω_{2h} the four eigenfunctions

$$\phi^{k,l}, \phi^{n-k,l}, \phi^{k,n-l}, \phi^{n-k,n-l}$$

cannot be distinguished (aliasing).

High and Low Frequencies and Coarse Meshes

- For the eigenfunctions $\phi^{k,l}$ of the Poisson equation with homogeneous Dirichlet boundary conditions we consider the frequencies:

$$\text{low frequency if } \max(k,l) < \frac{n}{2}$$

$$\text{high frequency if } \frac{n}{2} \leq \max(k,l) < n$$

- The low frequency components can be represented on the coarse mesh Ω_{2h}, the high frequency components not.

High and Low Frequencies and Coarse Meshes

- Split the iteration error $v_h(x, y)$ into low and high frequency components

$$
\begin{aligned}
v_h(x,y) &= \sum_{k,l=1}^{n-1} \alpha_{k,l} \phi_h^{k,l}(x,y) \\
&= \sum_{\text{low}} \alpha_{k,l} \phi_h^{k,l}(x,y) + \sum_{\text{high}} \alpha_{k,l} \phi_h^{k,l}(x,y) \\
&= \sum_{k,l=1}^{\frac{n}{2}-1} \alpha_{k,l} \phi_h^{k,l}(x,y) + \sum_{\substack{k,l=1 \\ \frac{n}{2} \leq \max(k,l)}}^{n-1} \alpha_{k,l} \phi_h^{k,l}(x,y)
\end{aligned}
$$

- Note, other types of coarsening, e.g. $H = 4h$, or semi-coarsening are also possible, but in general $H = 4h$ does not give an optimal multigrid method.

From Two Grids to Multigrid

- Take meshes $\Omega_h, \Omega_{2h}, \Omega_{4h}, \ldots$.
- Extend the idea that the smoother only acts on the high frequency part of the error.
- On each subsequent mesh the next part of the high frequency error is damped and the work on each coarser mesh is considerably smaller.
- For a successful multigrid method we need:
  - Good smoothers
  - A sequence of meshes
  - Grid transfer operators (restriction and prolongation operators)

Multigrid Features

- Multigrid as iterative solver:

  Convergence speed is independent of h for discrete strongly elliptic partial differential equations.

- Multigrid as solver for pde's:

  Full Multigrid Method (FMG), including self-adaptive methods

- Efficient: $O(N)$ operations for FMG

- Generality

  General types of pde's and boundary conditions and structured, unstructured, adaptively locally refined meshes and complex domains are possible.

Multigrid Features

- Optimization versus Robustness:

  Geometric Multigrid $\longleftrightarrow$ Algebraic Multigrid

  Optimal geometric multigrid requires well designed smoothers and coarse grid corrections (restriction, prolongation, coarse grid smoothers), which is often not trivial.

  Algebraic multigrid is more robust, but generally not optimal.

- Adaptivity combines well with multigrid methods.

- Multigrid has good parallel features, but the scalability on the coarse meshes can be a problem. Also, stability on coarse meshes needs to be ensured.

Iterative Solvers, Splitting and Preconditioners

Consider the linear system with a general invertible matrix A

$$Au = f.$$

- The simplest iterative scheme is the Richardson iteration

$$\begin{aligned} u^{n+1} &= u^m + \tau(f - Au^m) \\ &= (I - \tau A)u^m + \tau f. \end{aligned}$$

- General iteration

$$u^{m+1} = Mu^m + s$$

with iteration matrix M.

Iterative Solvers, Splitting and Preconditioners

- The asymptotic convergence of a general iteration method is characterized by the spectral radius $\rho(M)$

$$\rho(M) = \max\{|\lambda| \ : \ \lambda \text{ eigenvalue of } M\}.$$

- The spectral radius gives the asymptotic convergence rate

$$\|u - u^{m+1}\| \leq \rho(M)\|u - u^m\| \qquad \text{as } m \to \infty.$$

  Hence $\rho(M) < 1$ for convergence.

Iterative Solvers, Splitting and Preconditioners

- Three different, but equivalent ways, to formulate general iterative methods

1. Approximate solution of defect equation

   If u^m is an approximation of u and $d^m = f - Au^m$ (defect) then the defect equation is

   $$Av^m = d^m$$

   with $v^m = u - u^m$.

   After solving v^m in the defect equation the solution is

   $$u = u^m + v^m$$

   This approach is, however, just as expensive as solving the original problem $Au = f$.

Iterative Solvers, Splitting and Preconditioners

- Approximate A with a matrix $\widehat{A}$, where $\widehat{A}\hat{v}^m = d^m$ is easier to solve.

  Iterative process

$$d^m = f - Au^m$$

$$\widehat{A}\,\hat{v}^m = d^m$$

$$u^{m+1} = u^m + \hat{v}^m$$

  Combining the steps in the iterative process gives

$$\widehat{A}(u^{m+1} - u^m) = f - Au^m$$

  and we obtain

$$u^{m+1} = \widehat{A}^{-1} f + (I - \widehat{A}^{-1} A)u^m.$$

  The iteration matrix is equal to

$$M = I - \widehat{A}^{-1} A.$$

2. Splitting

Split the matrix A as

$$A = \widehat{A} - R$$

and use the iteration

$$\widehat{A}u^{m+1} = Ru^m + f.$$

The iteration matrix M is then equal to

$$M = \widehat{A}^{-1}R = \widehat{A}^{-1}(\widehat{A}^{-1} - A) = I - \widehat{A}^{-1}A$$

3. Preconditioning

Consider

$$CAu = Cf$$

with C an invertible matrix, called left preconditioner. Then

$$\begin{aligned} u^{m+1} &= u^m + C(f - Au^m) \quad \text{(Preconditioned Richardson iteration with } \tau = 1\text{)} \\ &= (I - CA)u^m + Cf \end{aligned}$$

with iteration matrix $M = I - CA$. Take the left preconditioner as $C = \widehat{A}^{-1}$ then

$$u^{m+1} = (I - \widehat{A}^{-1}A)u^m + Cf,$$

which gives the same iteration matrix as for the splitting and defect equation.

Iterative Solvers, Splitting and Preconditioners

- A right preconditioner C uses

$$ACz = f$$

with C an invertible matrix and $u = Cz$.

- The Richardson method ($\tau = 1$) for the left preconditioner then is

$$\begin{aligned} u^{m+1} &= u^m + C(f - Au^m) \\ &= (I - CA)u^m + Cf. \end{aligned}$$

- The Richardson method ($\tau = 1$) for the right preconditioner then is

$$z^{m+1} = (I - AC)z^m + f.$$

Iterative Solvers, Splitting and Preconditioners

- Use the relations

$$I - AC = AA^{-1} - ACAA^{-1}$$
$$= A(I - CA)A^{-1}.$$

Then the spectral radius of the right-preconditioner is equal to

$$\rho(I - AC) = \rho(A(I - CA)A^{-1})$$

Next, since

$$A(I - CA)A^{-1}z = \lambda z$$

is equal to

$$(I - CA)A^{-1}z = \lambda A^{-1}z$$

or equivalently with $y = A^{-1}z$

$$(I - CA)y = \lambda y$$

- We obtain thus the relation

$$\rho(I - AC) = \rho(A(I - CA)A^{-1}) = \rho(I - CA).$$

- The asymptotic convergence rate for a left and right preconditioner are thus the same.

Iterative Solvers, Splitting and Preconditioners

- The rate of convergence of the iterative method strongly depends on the condition number

$$\kappa(A) = \|A\|\|A^{-1}\|$$

- Assume that A is a symmetric positive definite matrix with maximum and minimum eigenvalues $\lambda_{\text{min}}, \lambda_{\text{max}} > 0$

- The spectral norm for a symmetric positive definite matrix A is

$$\|A\|_S = \lambda_{\text{max}}$$

The Richardson iteration then converges for

$$0 < \tau < 2\|A\|_s^{-1}.$$

Iterative Solvers, Splitting and Preconditioners

- The optimal τ for an SPD matrix A, which gives the minimal spectral radius in the Richardson iteration method $\rho(I - \tau A)$, is

$$\tau_{\text{opt}} = \frac{2}{(\lambda_{\min} + \lambda_{\max})}$$

and the spectral radius is equal to

$$\rho(I - \tau_{\text{opt}} A) = \|I - \tau_{\text{opt}} A\|_S = \frac{\lambda_{\max} - \lambda_{\min}}{\lambda_{\max} + \lambda_{\min}} = \frac{\kappa_s(A) - 1}{\kappa_s(A) + 1}$$

- If we use a left preconditioner C, which is also SPD, then

$$\rho(I - \tau_{\text{opt}} CA) = \frac{\kappa_s(CA) - 1}{\kappa_s(CA) + 1}$$

- Choosing a good preconditioner can thus significantly improve the convergence rate.