Parallel Numerics, WT 2012/2013

5 Iterative Methods for Sparse Linear Systems of Equations

Contents

1 Introduction
- **1.1** Computer Science Aspects
- **1.2** Numerical Problems
- **1.3** Graphs
- **1.4** Loop Manipulations

2 Elementary Linear Algebra Problems
- **2.1** BLAS: Basic Linear Algebra Subroutines
- **2.2** Matrix-Vector Operations
- **2.3** Matrix-Matrix-Product

3 Linear Systems of Equations with Dense Matrices
- **3.1** Gaussian Elimination
- **3.2** Parallelization
- **3.3** QR-Decomposition with Householder matrices

4 Sparse Matrices
- **4.1** General Properties, Storage
- **4.2** Sparse Matrices and Graphs
- **4.3** Reordering
- **4.4** Gaussian Elimination for Sparse Matrices

5 Iterative Methods for Sparse Linear Systems of Equations
- **5.1** Stationary Methods
- **5.2** Nonstationary Methods
- **5.3** Preconditioning

6 Domain Decomposition

- Disadvantages of direct methods (in parallel):
  - strongly sequential
  - may lead to dense matrices
  - sparsity pattern changes, additional entries necessary
  - indirect addressing
  - storage
  - computational effort
- Iterative solver:
  - choose initial guess = starting vector $x^{(0)}$, e.g., $x^{(0)} = \mathbf{0}$
  - iteration function $x^{(k+1)} := \Phi(x^{(k)})$
- Applied on solving a linear system:
  - Main part of Φ should be a matrix-vector multiplication Ax (matrix-free!?)
  - Easy to parallelize, no change in the pattern of A.

$$x^{(k)} \stackrel{k\to\infty}{\longrightarrow} \bar{x} = A^{-1}b$$

  - Main problem: Fast convergence!

5.1. Stationary Methods

5.1.1. Richardson Iteration

- Construct from $Ax = b$ an iteration process:

$$b = Ax = (\underbrace{A - I + I}_{\text{(artificial) splitting of } A})x = x - (I - A)x \quad \Rightarrow \quad x = b + (I - A)x$$

$$= b + Nx$$

- Leads to equation $x = \Phi(x)$ with $\Phi(x) := b + Nx$:

$$\text{start: } x^{(0)};$$

$$x^{(k+1)} := \Phi(x^{(k)}) = b + Nx^{(k)} = b + (I - A)x^{(k)}$$

Richardson Iteration (cont.)

$$\text{start: } x^{(0)};$$
$$x^{(k+1)} := \Phi(x^{(k)}) = b + Nx^{(k)} = b + (I - A)x^{(k)}$$

If $x^{(k)}$ is convergent, $x^{(k)} \to \tilde{x}$,

then

$$\tilde{x} = \Phi(\tilde{x}) = b + N\tilde{x} = b + (I - A)\tilde{x} \Rightarrow A\tilde{x} = b$$

and therefore it holds

$$x^{(k)} \to \tilde{x} = \bar{x} := A^{-1}b$$

Residual-based formulation:

$$\begin{aligned} x^{(k+1)} &= \Phi(x^{(k)}) = b + (I - A)x^{(k)} = b + x^{(k)} - Ax^{(k)} \\ &= x^{(k)} + \underbrace{(b - Ax^{(k)})}_{r(x) \,=\, \text{residual}} = x^{(k)} + r(x^{(k)}) \end{aligned}$$

Convergence Analysis via Neumann Series

$$\begin{aligned}
x^{(k)} &= b + Nx^{(k-1)} = b + N(b + Nx^{(k-2)}) = b + Nb + N^2 x^{(k-2)} = \\
\dots &= b + Nb + N^2 b + \cdots + N^{k-1} b + N^k x^{(0)} = \\
&= \sum_{j=0}^{k-1} N^j b + N^k x^{(0)} = \left(\sum_{j=0}^{k-1} N^j\right) b + N^k x^{(0)}
\end{aligned}$$

Special case $x^{(0)} = 0$:

$$x^{(k)} = \left(\sum_{j=0}^{k-1} N^j\right) b$$

$$\begin{aligned}
\Rightarrow x^{(k)} \in \mathrm{span}\{b, Nb, N^2 b, \dots, N^{k-1} b\} &= \mathrm{span}\{b, Ab, A^2 b, \dots, A^{k-1} b\} \\
&= K_k(A, b)
\end{aligned}$$

which is called the Krylov space to A and b.

For $\|N\| < 1$ holds:

$$\sum_{j=0}^{k-1} N^j \to \sum_{j=0}^{\infty} N^j = (I - N)^{-1} = (I - (I - A))^{-1} = A^{-1}$$

Convergence Analysis via Neumann Series (cont.)

$$x^{(k)} \to \left(\sum_{j=0}^{\infty} N^j\right) b = (I-N)^{-1} b = A^{-1} b = \bar{x}$$

Richardson iteration is convergent for $\|N\| < 1$ or $A \approx I$.

Error analysis for $e^{(k)} := x^{(k)} - \bar{x}$:

$$\begin{aligned} e^{(k+1)} &= x^{(k+1)} - \bar{x} = \Phi(x^{(k)}) - \Phi(\bar{x}) = (b + Nx^{(k)}) - (b + N\bar{x}) = \\ &= N(x^{(k)} - \bar{x}) = Ne^{(k)} \end{aligned}$$

$$\|e^{(k)}\| \leq \|N\|\|e^{(k-1)}\| \leq \|N\|^2\|e^{(k-2)}\| \leq \cdots \leq \|N\|^k\|e^{(0)}\|$$

$$\|N\| < 1 \Rightarrow \|N\|^k \stackrel{k\to\infty}{\longrightarrow} 0 \Rightarrow \|e^{(k)}\| \stackrel{k\to\infty}{\longrightarrow} 0$$

- Convergence, if $\rho(N) = \rho(I - A) < 1$, where ρ is spectral radius

$$\rho(N) = |\lambda_{\max}| = \max_i(|\lambda_i|) \qquad (\lambda_i \text{ is eigenvalue of } N)$$

- Eigenvalues of A have to be all in circle around 1 with radius 1.

Splittings of A

- Convergence of Richardson only in very special cases!
  Try to improve the iteration for better convergence!
- Write A in form $A := M - N$

$$b = Ax = (M - N)x = Mx - Nx \Leftrightarrow x = M^{-1}b + M^{-1}Nx = \Phi(x)$$

$$\begin{aligned}\Phi(x) &= M^{-1}b + M^{-1}Nx = M^{-1}b + M^{-1}(M - A)x = \\ &= M^{-1}(b - Ax) + x = x + M^{-1}r(x)\end{aligned}$$

- N should be such that Ny can be evaluated efficiently.
- M should be such that $M^{-1}y$ can be evaluated efficiently.

$$x^{(k+1)} = x^{(k)} + M^{-1}r^{(k)}$$

- Iteration with splitting $M - N$ is equivalent to Richardson on

$$M^{-1}Ax = M^{-1}b$$

Convergence

- Iteration with splitting $A = M - N$ is convergent if

$$\rho(M^{-1}N) = \rho(I - M^{-1}A) < 1$$

- For fast convergence it should hold
  - $M^{-1}A \approx I$
  - $M^{-1}A$ should be better conditioned than A itself

- Such a matrix M is called a preconditioner for A.
  Is used in other iterative methods to accelerate convergence.

- Condition number:

$$\kappa(A) = \|A^{-1}\|\,\|A\|, \quad \left|\frac{\lambda_{\max}}{\lambda_{\min}}\right|, \quad \text{or} \quad \frac{\sigma_{\max}}{\sigma_{\min}}$$

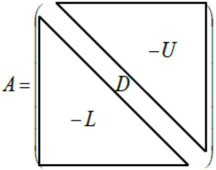
5.1.2. Jacobi (Diagonal) Splitting

Choose $A = M - N = D - (L + U)$ with

$D = \text{diag}(A)$

L the lower triangular part of A, and

U the upper triangular part.



$$\begin{aligned} x^{(k+1)} &= D^{-1}b + D^{-1}(L+U)x^{(k)} = \\ &= D^{-1}b + D^{-1}(D-A)x^{(k)} = x^{(k)} + D^{-1}r^{(k)} \end{aligned}$$

Convergent for $A \approx \text{diag}(A)$ or diagonal dominant matrices:

$$\rho(D^{-1}N) = \rho(I - D^{-1}A) < 1$$

Jacobi (Diagonal) Splitting (cont.)

Iteration process written elementwise:

$$x^{(k+1)} = D^{-1}(b - (A - D)x^{(k)}) \Rightarrow x_j^{(k+1)} = \frac{1}{a_{jj}} \left(b_j - \sum_{m=1, m \neq j}^{n} a_{j,m} x_m^{(k)} \right)$$

$$a_{jj} x_j^{(k+1)} = b_j - \sum_{m=1}^{j-1} a_{j,m} x_m^{(k)} - \sum_{m=j+1}^{n} a_{j,m} x_m^{(k)}$$

- Damping or relaxation for improving convergence
- Idea: Iterative method as correction of last iterate in search direction.
- Introduce step length for this correction step:

$$x^{(k+1)} = x^{(k)} + D^{-1} r^{(k)} \qquad \rightarrow \qquad x^{(k+1)} = x^{(k)} + \omega D^{-1} r^{(k)}$$

  with additional damping parameter ω.
- Damped Jacobi iteration:

$$x_{\text{damped}}^{(k+1)} = (\omega + 1 - \omega) x^{(k)} + \omega D^{-1} r^{(k)} = \omega x^{(k+1)} + (1 - \omega) x^{(k)}$$

Damped Jacobi Iteration

$$\begin{aligned} x^{(k+1)} &= x^{(k)} + \omega D^{-1} r^{(k)} = x^{(k)} + \omega D^{-1}(b - Ax^{(k)}) = \\ &= \dots \\ &= \omega D^{-1} b + [(1-\omega)I + \omega D^{-1}(L+U)]x^{(k)} \end{aligned}$$

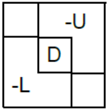
is convergent for

$$\rho(\underbrace{[(1-\omega)I + \omega D^{-1}(L+U)]}_{\stackrel{\omega \to 0}{\longrightarrow} I}) < 1$$

Look for optimal ω with best convergence (add. degree of freedom).

Parallelism in the Jacobi Iteration

- Jacobi method is easy to parallelize: only Ax and $D^{-1}x$.
- But often too slow convergence!
- Improvement: block Jacobi iteration



5.1.3. Gauss-Seidel Iteration

Always use newest information available!

Jacobi iteration:

$$a_{jj}x_j^{(k+1)} = b_j - \sum_{m=1}^{j-1} a_{j,m} \underbrace{x_m^{(k)}}_{\text{already computed}} - \sum_{m=j+1}^{n} a_{j,m}x_m^{(k)}$$

Gauss-Seidel iteration:

$$a_{jj}x_j^{(k+1)} = b_j - \sum_{m=1}^{j-1} a_{j,m} \underbrace{x_m^{(k+1)}}_{\text{already computed}} - \sum_{m=j+1}^{n} a_{j,m}x_m^{(k)}$$

Gauss-Seidel Iteration (cont.)

- Compare dependancy graphs for general iterative algorithms. Here:

$$x = f(x) = D^{-1}(b + (D - A)x) = D^{-1}(b - (L + U)x)$$

to splitting $A = (D - L) - U = M - N$

$$\begin{aligned} x^{(k+1)} &= (D-L)^{-1}b + (D-L)^{-1}Ux^{(k)} = \\ &= (D-L)^{-1}b + (D-L)^{-1}(D-L-A)x^{(k)} = \\ &= x^{(k)} + (D-L)^{-1}r^{(k)} \end{aligned}$$

- Convergence depends on spectral radius $\rho(I - (D - L)^{-1}A) < 1$

Parallelism in the Gauss-Seidel Iteration

- Linear system in $D - L$ is easy to solve because $D - L$ is lower triangular but
- strongly sequential!
- Use red-black ordering or graph colouring for compromise:

$$\text{parallel} \leftrightarrow \text{convergence}$$

Successive Over Relaxation (SOR)

- Damping or relaxation:

$$x^{(k+1)} = x^{(k)} + \omega(D-L)^{-1}r^{(k)} = \omega(D-L)^{-1}b + [(1-\omega) + \omega(D-L)^{-1}U]x^{(k)}$$

- Convergence depends on spectral radius of iteration matrix

$$(1-\omega) + \omega(D-L)^{-1}U$$

- Parallelization of SOR == parallelization of GS

Stationary Methods (in General)

- Can always be written in the two normal forms

  $$x^{(k+1)} = c + Bx^{(k)}$$

  with convergence depending on $\rho(B)$ and

  $$x^{(k+1)} = x^{(k)} + Fr^{(k)}$$

  with preconditioner F, $B = I - FA$

- For $x^{(0)} = \mathbf{0}$:

  $$x^{(k+1)} \subseteq K_k(B, c),$$

  which is the Krylov space with respect to matrix B and vector c.

- Slow convergence (but good smoothing properties! $\rightarrow$ multigrid)