Solvers

Matthias Bolten
University of Wuppertal

2011/06/22 and 2011/06/23

Part I

Solvers for linear sytsems

Outline

Introduction

Direct methods

Iterative methods

Krylov subspace methods

Preconditioning

Introduction

- Sought for is the solution of

$$Ax = b, \tag{1}$$

where $A \in \mathbb{C}^{n \times n}$ non-singular and $x, b \in \mathbb{C}^n$

- Often A is sparse (as in QCD...)
- How to solve (1)?
  1. Directly
  2. Iteratively

Direct methods

- Usually based on *factorization* of system matrix A
- Well-known methods:
  - $A = LU$: LU factorization (Gaussian elimination)
  - $A = LDL^T$: LDL factorization
  - $A = LL^*$: Cholesky factorization (A hermitian, positive (semi-)definite)
- Usually expensive (cf. $\mathcal{O}(n^3)$ for dense matrices)
- Methods exploiting sparsity exist, reducing complexity
- Inexact methods used as preconditioners
- Also include special solvers, e.g. using FFTs
- Too expensive for many applications. . .

Iterative methods

- Start with initial guess $x^{(0)}$
- Construct sequence $\{x^{(k)}\}_{k=0}^{\infty}$ of approximate solutions
- Two classes:
  - Stationary methods
  - Non-stationary methods
- Stationary methods characterized by iteration matrix M
- Examples for stationary methods:
  - Jacobi method ($M = I - D^{-1}A$)
  - Gauss-Seidel methods ($M = I - (D - L)^{-1}A$)
  - Successive over-relaxation (SOR, $M(\omega) = I - \omega(D - \omega L)^{-1}A$)
  - Symmetric SOR (SSOR, $M(\omega) = \dots$)
- Non-stationary methods: Krylov-subspace methods

Krylov subspace methods

- Non-stationary iterative methods
- Approximation of solution in *Krylov subspace*:

$$\mathcal{K}_k(A, b) = \langle b, Ab, A^2 b, \dots, A^{k-1} b \rangle \subseteq \mathbb{C}^n$$

- *Krylov matrix* defined by

$$K_k = \left[b \middle| Ab \middle| A^2 b \middle| \dots \middle| A^{k-1} b \right]$$

- K_k has reduced QR factorization

$$K_k = Q_k R_k$$

- Basis Q_k of K_k created by *Arnoldi iteration*

Arnoldi iteration I

> *Suppose, to pass the time while marooned on a desert island, you challenged yourself to devise an algorithm to reduce a nonhermitian matrix to Hessenberg form by orthogonal similarity transformations, proceeding column by column from a prescribed first column q_1. To your surprise you could solve this problem in an hour and still have time to gather coconuts for dinner. The method you would come up with goes by the name of the Arnoldi iteration.*
>
> (Lloyd N. Trefethen, Numerical Linear Algebra, SIAM, Philadelphia, 1997)

Arnoldi iteration II

- Complete orthogonal similarity transform given by

$$A = QHQ^* \Leftrightarrow AQ = QH$$

- Let $Q_k \in \mathbb{C}^{n \times k}$ consist of the first k columns of Q
- Define $\tilde{H}_k$ as upper-left section of H

$$\tilde{H}_k = \begin{bmatrix} h_{1,1} & & \cdots & h_{1,k} \\ h_{2,1} & h_{2,2} & & \\ & \ddots & \ddots & \vdots \\ & & h_{k,k-1} & h_{k,k} \\ & & & h_{k+1,k} \end{bmatrix}$$

- Then $AQ_k = Q_{k+1}\tilde{H}_k$ (Arnoldi relation)

Arnoldi iteration III

- k-th column given by $(n+1)$-term recurrence:

$$Aq_k = h_{1,k}q_1 + \cdots + h_{k,k}q_k + h_{k+1,k}q_{k+1}$$

Algorithm (Arnoldi iteration)

```
$b = \text{arbitrary}, q_1 = b/\|b\|$
for $k = 1, 2, \dots$ do
    $v = Aq_k$
    for $j = 1, \dots, i$ do
        $h_{j,k} = q_j^* v$
        $v = v - h_{j,k}q_j$
    end for
    $h_{k+1,k} = \|v\|$
    $q_{k+1} = v/h_{k+1,k}$
end for
```

GMRES

- GMRES computes $x^{(k)} = K_k c^{(k)} \in \mathcal{K}_k, c^{(k)} \in \mathbb{C}^k$ s.t.

$$\|AK_k c^{(k)} - b\| = \text{minimum}$$

- Set $x^{(k)} = Q_k y^{(k)}$ and use Arnoldi relation to reduce (1) to

$$\begin{aligned} \|AQ_k y^{(k)} - b\| &= \text{minimum} \\ \Leftrightarrow \|Q_{k+1}\tilde{H}_k y^{(k)} - b\| &= \text{minimum} \end{aligned}$$

- Properties of Q_{k+1} finally yield

$$\|\tilde{H}_k y - \|b\| e_1\| = \text{minimum}$$

- Problem is reduced to $(k+1) \times k$ matrix least squares problem
- Work further reduced by updating QR factorization

Restarted GMRES

- Size of least squares problem grows with number of iterations
- Size of Q_k also grows, increased memory requirement
- Approach to limit both grows: Restarting after m iterations
- Downside: Method is not guaranteed to converge anymore

Algorithm (restarted GMRES, GMRES(m))

```
for $j = 0, 1, \dots$ do
    $r^{(j \cdot m)} = b - Ax^{(j \cdot m)}$
    $q_1 = r^{(j \cdot m)} / \|r^{(j \cdot m)}\|$
    for $k = 1, \dots, m$ do
        Step $k$ of Arnoldi iteration
        Find $y$ to minimize $\|\tilde{H}_k y - \|r\| e_1\|$
        $x^{(j \cdot m + k)} = x^{(j \cdot m)} + Q_k y$
    end for
end for
```

Lanzcos iteration I

- For hermitian matrices Hessenberg matrix H_k becomes tridiagonal matrix T_k:

$$T_k = \begin{bmatrix} \alpha_1 & \beta_1 & & & \\ \beta_1 & \alpha_2 & \beta_2 & & \\ & \beta_2 & \alpha_3 & \ddots & \\ & & \ddots & \ddots & \beta_{k-1} \\ & & & \beta_{k-1} & \alpha_k \end{bmatrix}$$

- $(k+1)$-term recurrence reduces to 3-term recurrence
- Arnoldi iteration reduces to Lanczos iteration
- GMRES reduces to MINRES

Lanczos iteration II

Algorithm (Lanczos iteration)

```
$\beta_0 = 0, q_0 = 0, b =$ arbitrary, $q_1 = b/\|b\|$
for $k = 1, 2, \dots$ do
    $v = Aq_k$
    $\alpha_k = q_k^* v$
    $v = v - \beta_{k-1} q_{k-1} - \alpha_k q_k$
    $\beta_k = \|v\|$
    $q_{k+1} = v/\beta_k$
end for
```

Conjugate gradients (CG)

- Can be interpreted as an optimization of the functional

$$\varphi(x) = 1/2x^*Ax - x^*b$$

- Minimizes A-norm of the error (instead of 2-norm of residual)
- Error satisfies $\|e_k\|_A/\|e_0\|_A \leq 2((\sqrt{\kappa}-1)/(\sqrt{\kappa}+1))^k$

Algorithm (CG iteration)

```
$x^{(0)} = 0, r^{(0)} = 0, p^{(0)} = 0$
for $k = 1, 2, \dots$ do
    $\alpha_k = ((r^{(k-1)})^* r^{(k-1)})/((p^{(k-1)})^* A p^{(k-1)})$
    $x^{(k)} = x^{(k-1)} + \alpha_k p^{(k-1)}$
    $r^{(k)} = r^{(k-1)} - \alpha_k p^{(k-1)}$
    $\beta_k = ((r^{(k)})^* r^{(k)})/((r^{(k-1)})^* r^{(k-1)})$
    $p^{(k)} = r^{(k)} + \beta_k p^{(k-1)}$
end for
```

Biorthogonalization

- CG for nonhermitian matrices using normal equation

$$Ax = b \Leftrightarrow A^*Ax = A^*b$$

- Squared condition number leads to slow convergence
- Alterative: Tridiagonal biorthogonalization

$$A = VTV^{-1} \text{ and } A^* = \underbrace{V^{-*}}_{=:W} T^*(V^{-*})^{-1} = WT^*W^{-1}$$

where $W^*V = V^*W = I$

- Columns of V and W span $\mathcal{K}(A, v_1)$ and $\mathcal{K}(A^*, w_1)$
- Leads to BiCG and variants (including BiCGstab, QMR,...)

Summary

- Krylov subspace methods are fast solvers
- They are easy to implement
- They are easy to parallelize
- Efficiency depends on efficiency of matrix vector multiplication
- Methods with 3-term recurrence need fixed amount of memory
- In exact arithmetic solution is obtained in n steps
- Convergence rate depends on the eigenvalues
- Preconditioning necessary in many cases

Preconditioning

- Given a nonsingular $M \in \mathbb{C}^{n \times n}$ (1) is equivalent to
$$M^{-1}Ax = M^{-1}b$$
- Convergence of iterative solver now depends on $M^{-1}A$
- M should be chosen such that convergence rate is improved (optimal: $M = A$)
- Linear systems with system matrix M should be easy to solve
- This is called (left) *preconditioning*
- Right preconditioning: Solve $AM^{-1}y = b$, then $Mx = y$
- Hermitian preconditioning: $M = CC^*$, (1) transformed to
$$(C^{-1}AC^{-*})C^*x = C^{-1}b$$

Diagonal scaling, Jacobi and Gauss-Seidel-type

- Simple preconditioner: Scale A by $M = diag(A)$ like in Jacobi
- Easy to implement
- Often very effective
- More general: Scaling with $M = diag(c)$ for some vector $c \in \mathbb{C}^n$
- Extension: Apply Gauss-Seidel, SOR or SSOR

Incomplete factorizations

- Gaussian elimination produces LU factorization

$$A = LU$$

- Even for sparse A L and U are usually not as sparse
- Incomplete factorization obtained by allowing nonzeros only, where A was nonzero
- Same is possible for Cholesky factorization
- Extension possible by introducing a drop tolerance, multiple levels, ...
- Not as easy to implement
- Hard to parallelize

Deflation

- Convergence often decelerated by a few eigenvalues
- Given that the eigenvectors are known, they can be *deflated*
- For ℓ (left) orthonormal eigenvectors u_j corresponding to eigenvalues λ_j this leads to left preconditioner

$$M = I - \sum_{j=1}^{\ell} \lambda_j v_j v_j^*$$

- Can be combined with other preconditioners
- Can be combined with subspace receycling
- Efficiency limited by growing number of "critical" eigenvalues

Outlook

- Other notable preconditioners:
  - Domain decomposition (local solution on subdomains)
  - Multigrid (see K. Kahl's lecture on friday)
  - Mixed-precision
  - ...
- Choice of right preconditioner difficult
- Some need *flexible* Krylov methods

Part II

Solvers for linear systems in Lattice QCD

Outline

The Wilson fermion matrix

Solvers

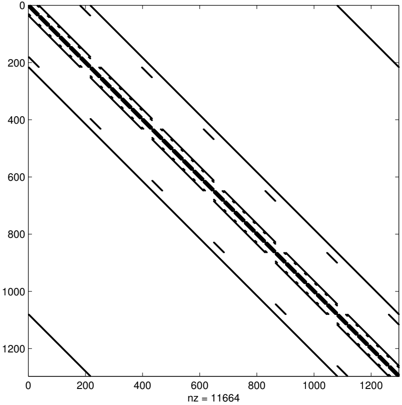
Preconditioners

The overlap operator

Outlook

The Wilson fermion matrix I

- $M = I - \kappa D$
- $M \in \mathbb{C}^{n \times n}$
- Nearest neighbor coupling on 4-dimensional torus
- 12 variables per grid point
- $n = 12 \cdot n_1 \cdot n_2 \cdot n_3 \cdot n_4$



The Wilson fermion matrix II

We have

$$(M\psi)_x = \psi_x - \kappa \left(\sum_{\mu=1}^{4} ((I - \gamma_\mu) \otimes U_\mu(x))\, \psi_{x+e_\mu} + \sum_{\mu=1}^{4} \Big((I + \gamma_\mu) \otimes U_\mu^H(x - e_\mu)\Big)\, \psi_{x-e_\mu} \right) .$$

Here:

- $U_\mu(x) \in SU(3)$
- $\gamma_\mu \in \mathbb{C}^{4\times 4}$
- $I \pm \gamma_\mu$ is projector on 2-dimensional subspace

γ_5-symmetry of the Wilson fermion matrix

$$\Gamma_5 M = M^H \Gamma_5,$$

where Γ_5 is a simple permutation,

$$\Gamma_5 = I \otimes (\gamma_5 \otimes I_3),$$

$$\gamma_5 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \text{ or } \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}.$$

Consequences:

- $\lambda \in \text{spec}(M) \Rightarrow \bar{\lambda} \in \text{spec}(M)$
- unsymmetric Lanczos process with Γ_5 instead of M^H
- $Q = \Gamma_5 M$ is hermitian (and *maximally indefinite*)

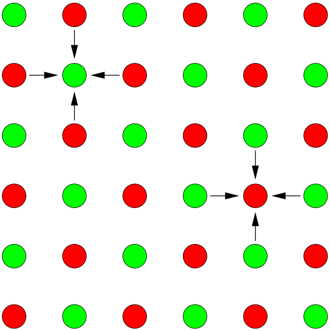
Odd-even symmetry of D

- Grid points x are even or odd (= red or green).
- odd-even-ordering yields

$$D = \begin{bmatrix} 0 & D_{oe} \\ D_{eo} & 0 \end{bmatrix}$$

- Consequence:

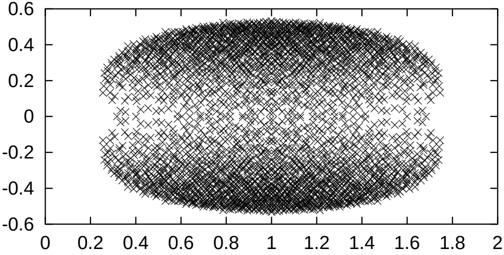
$$\mu \in \text{spec}(D) \Rightarrow -\mu \in \text{spec}(D),$$
$$\lambda \in \text{spec}(M) \Rightarrow 2 - \lambda \in \text{spec}(M)$$



Spectrum of the Wilson fermion matrix

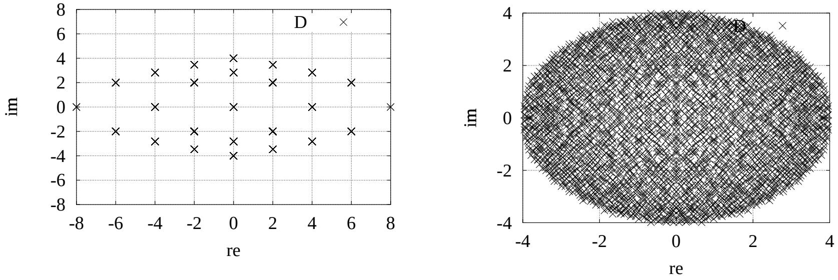
- M is positive real for $0 \leq \kappa < \kappa_{\text{crit}}$
- κ close to κ_{crit} is interesting: relative quark mass m_q becomes small,

$$m_q = \frac{1}{2}\left(\frac{1}{\kappa} - \frac{1}{\kappa_{\text{crit}}}\right)$$



spec(M) for 4^4 grid (realistic configuration)

Spectrum of D



spec(D) for *cold* ($U_\mu(x) = I$) and *hot* ($U_\mu(x)$ random) configurations

Solvers

The whole "zoo" of solvers is used:

- "minimal residual" $=$ GMRES(1)
- CGN
- BiCG
- QMR
- BiCGstab
- GMRES(k)
- MINRES for Q

A recurring theme: Shifted systems

Often solution of

$$(I - \kappa D)\,\psi = \varphi$$

is sought for several values of κ.

Observation: Krylov subspaces independent of κ.

Potential: Solve

- for several κ at the same time with
- just one matrix vector multiplication per step for all systems.

Shifted methods

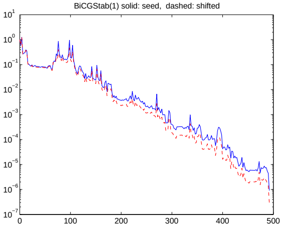
- Shifted CG
- Shifted QMR
- Shifted Chebyshev
- Shifted BiCG
- Shifted FOM
- Shifted GMRES(k)
- Shifted BiCGstab
- Shifted BiCGstab(ℓ)

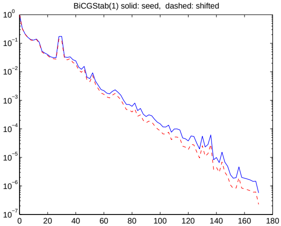
Sample Theorem [Frommer, Glässner 98, Frommer 03]:
Perform true GMRES(k) for largest $\kappa < \kappa_c$

$\rightarrow$

shifted method converges faster for all other values of κ.

Example: Shifted BiCGstab





$\kappa_1 = 0.180, \kappa_2 = 0.176$

$\kappa_1 = 0.176, \kappa_2 = 0.170$

Preconditioners

- Preconditioning accepted technique and widely used
- Used techniques:
  - Odd-even preconditioning
  - SSOR
  - Domain decomposition
  - Multi-level domain decomposition
  - Multigrid
  - . . .
- Preconditioners vary widely in
  - Ease of implementation
  - Presence of setup phase
  - Computational cost
  - Efficiency

(as it is always the case. . .)

Odd-even preconditioning

For odd-even ordering we obtain

$$\begin{bmatrix} I & -\kappa D_{oe} \\ -\kappa D_{eo} & I \end{bmatrix} \begin{bmatrix} \psi_o \\ \psi_e \end{bmatrix} = \begin{bmatrix} \varphi_o \\ \varphi_e \end{bmatrix}$$

$$\iff \begin{bmatrix} I & -\kappa D_{oe} \\ 0 & I - \kappa^2 D_{eo} D_{oe} \end{bmatrix} \begin{bmatrix} \psi_o \\ \psi_e \end{bmatrix} = \begin{bmatrix} \varphi_o \\ \varphi_e + \kappa D_{eo} \varphi_o \end{bmatrix},$$

so

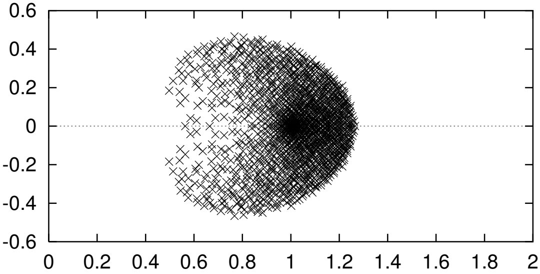
$$M_e \psi_e = \tilde{\varphi}_e, \ \psi_o = \varphi_o + \kappa D_{oe} \psi_e$$

where

$$M_e = I - \kappa^2 D_{eo} D_{oe}, \ \tilde{\varphi}_e = \varphi_e + \kappa D_{eo} \varphi_o.$$

The odd-even reduced system

- Odd-even reduced system $M_e \psi_e = \tilde{\varphi}_e$ is γ_5-symmetric
- Improves convergence speed by a factor of 2



spec(M_e) for a 4^4 lattice with $\beta = 5.0$ and $\kappa = .150$.

Block SSOR preconditioning

The system

$$M\psi = \varphi$$

is preconditioned as

$$V_1^{-1} M V_2^{-1} \tilde{\psi} = \tilde{\varphi},\ \tilde{\varphi} = V_1^{-1}\varphi,\ \tilde{\psi} = V_2 \psi.$$

Let $M = I - L - U$ be the decomposition of M into its diagonal, strictly lower triangular matrix $-L$ and strictly upper triangular matrix $-U$. For $\omega \neq 0$ the *block SSOR preconditioner* is defined by

$$V_1 = \left(\frac{1}{\omega} I - L\right)\left(\frac{1}{\omega} I\right)^{-1}, V_2 = \frac{1}{\omega} I - U.$$

Domain decomposition

The overlap operator

Approximation for sign$(Q)b$ in Krylov subspace

Numerical results

Stopping criterion

Rational approximations

Outlook

-