Scientific Computing WS 2020/2021

Slide lecture 4

Jürgen Fuhrmann

juergen.fuhrmann@wias-berlin.de

Generalization of iteration schemes

- So far we considered simple iterative schemes, perhaps with preconditioners
- Here, we introduce Krylov subspace methods which indeed in many cases yield faster convergence than simple iterative schemes
- Material after
  - M. Gutknecht A Brief Introduction to Krylov Space Methods for Solving Linear Systems
  - J. Shewchuk: An Introduction to the Conjugate Gradient Method Without the Agonizing Pain“
- Extended reading: J.Liesen, Z. Strakoš: Krylov Subspace Methods: Principles and Analysis
- Extended coverage of the topic available at TU: Prof. Jörg Liesen, Prof. Reinhard Nabben are active researchers in the field.

Simple iterative method I

Solve $Au = b$, assume exact solution $\hat{u}$.

$$u_{k+1} = u_k - \alpha(Au_k - b) \quad (k = 0, 1 \dots)$$

1. Choose initial value u_0, tolerance ε, set $k = 0$
2. Calculate *residuum* $r_k = Au_k - b$
3. Test convergence: if $||r_k|| < \varepsilon$ set $u = u_k$, finish
4. Update solution: $u_{k+1} = u_k - \alpha r_k$, set $k = k + 1$, repeat with step 2.

Simple iterative method II

- From step 4:

$$\begin{aligned} Au_{k+1} &= Au_k - \alpha Ar_k \\ Au_{k+1} - b &= Au_k - b - \alpha Ar_k \\ r_{k+1} &= r_k - \alpha Ar_k \end{aligned}$$

- Therefore $r_k = p_k(A)r_0 \in \text{span}\{r_0, Ar_0, \ldots, A^k r_0\}$ where $p(\xi) = (1 - \alpha\xi)^k$ is a polynomial of degre k
- For the iterate u_k, we have

$$\begin{aligned} u_k &= u_{k-1} - \alpha r_{k-1} = u_{k-2} - \alpha r_{k-2} - \alpha r_{k-1} \\ &= u_0 - \alpha(r_0 + r_1 + \cdots + r_{k-1}) \\ &= u_0 + q_{k-1}(A)r_0 \end{aligned}$$

  where q_{k-1} is a polynomial of degree $n - 1$.
- From $r_k = Au_k - b = Au_0 - b + Aq_{k-1}(A)r_0 = (I + Aq_{k-1}(A))r_0$ one obtains $p_k(\xi) = 1 + \xi q_k(\xi)$.
- Consequently, $u_k \in u_0 + \text{span}\{r_0, Ar_0, \ldots, A^{k-1} r_0\}$

The Krylov subspace

Definition: Let $A \in \mathbb{R}^{N\times N}$ be nonsingular, let $0 \neq y \in \mathbb{R}^n$. The k-th *Krylov subspace* generated from A by y is defined as $\mathcal{K}_k(A, y) = \text{span}\{y, Ay, \ldots, A^{k-1}y\}$.

- $\mathcal{K}_1 \subseteq \mathcal{K}_2 \subseteq \cdots \subseteq \mathcal{K}_k$
- $\dim \mathcal{K}_i \leq \dim \mathcal{K}_{i-1} + 1$
- For the simple iteration,

$$\begin{aligned} u_k &= u_0 + q_{k-1}(A)r_0 \in \mathcal{K}_k(A, r_0) \\ r_k &= p_k(A)r_0 \in \mathcal{K}_{k+1}(A, r_0) \\ p_k(\xi) &= 1 + \xi q_k(\xi) \\ p_k(0) &= 1 \end{aligned}$$

with particular polynomials p, q.

Are these the best ones possible ? - Not necessarily: we can try to find other ones which yield better convergence...

Krylov subspace methods

Definition: Let $A \in \mathbb{R}^{N\times N}$ be nonsingular, let $0 \neq y \in \mathbb{R}^N$. An iterative method such that

$$u_k = u_0 + q_{k-1}(A)r_0 \in \mathcal{K}_k(A, r_0)$$

where q_{k-1} is a polynomial of degree k is called *Krylov subspace method*.

- For the residuals of the method, we have $r_k = p_k(A)r_0 \in \mathcal{K}_{k+1}(A, r_0)$ with $p_k(\xi) = 1 + \xi q_k(\xi)$
- Preconditioned form: use the same ansatz for $M^{-1}Ax = M^{-1}b$ and define Krylov subspace for $M^{-1}A$

The case of symmetric positive definite matrices

Assume A is spd (symmetric, positive definite)

$$a(u,v) = (Au,v) = v^T Au = \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij} v_i u_j$$

As A is SPD, for all $u \neq 0$ we have $(Au,u) > 0$.

For a given vector b, regard the function

$$f(u) = \frac{1}{2} a(u,u) - b^T u$$

What is the minimizer of f ?

$$f'(u) = Au - b = 0$$

- Solution of SPD system $\equiv$ minimization of f.

Method of steepest descent

- Given some vector u_i, look for a new iterate u_{i+1}.
- The direction of steepest descend is given by $-f'(u_i)$.
- So look for u_{i+1} in the direction of $-f'(u_i) = r_i = b - Au_i$ such that it minimizes f in this direction, i.e. set $u_{i+1} = u_i + \alpha r_i$ with α choosen from

$$\begin{aligned} 0 &= \frac{d}{d\alpha} f(u_i + \alpha r_i) = f'(u_i + \alpha r_i) \cdot r_i \\ &= (b - A(u_i + \alpha r_i), r_i) \\ &= (b - Au_i, r_i) - \alpha(Ar_i, r_i) \\ &= (r_i, r_i) - \alpha(Ar_i, r_i) \\ \alpha &= \frac{(r_i, r_i)}{(Ar_i, r_i)} \end{aligned}$$

Method of steepest descent: iteration scheme

$$r_i = b - Au_i$$
$$\alpha_i = \frac{(r_i, r_i)}{(Ar_i, r_i)}$$
$$u_{i+1} = u_i + \alpha_i r_i$$

Let $\hat{u}$ the exact solution. Define $e_i = u_i - \hat{u}$, then $r_i = -Ae_i$

Let $||u||_A = (Au, u)^{\frac{1}{2}}$ be the *energy norm* wrt. A.

Theorem The convergence rate of the method is

$$||e_i||_A \leq \left(\frac{\kappa - 1}{\kappa + 1}\right)^i ||e_0||_A$$

where $\kappa = \frac{\lambda_{max}(A)}{\lambda_{min}(A)}$ is the spectral condition number.

Method of steepest descent: advantages

- Simple Richardson iteration $u_{k+1} = u_k - \alpha(Au_k - f)$ needs good eigenvalue estimate to be optimal with $\alpha = \frac{2}{\lambda_{max}+\lambda_{min}}$
- In this case, asymptotic convergence rate is $\rho = \frac{\kappa-1}{\kappa+1}$
- Steepest descent has the same rate without need for spectral estimate

Conjugate directions

For steepest descent, there is no guarantee that a search direction $d_i = r_i = -Ae_i$ is not used several times. If all search directions would be orthogonal, or, indeed, A-orthogonal, one could control this situation.

So, let $d_0, d_1 \dots d_{n-1}$ be a series of A-orthogonal (or conjugate) search directions, i.e. $(Ad_i, d_j) = 0$, $i \neq j$.

- Look for u_{i+1} in the direction of d_i such that it minimizes f in this direction, i.e. set $u_{i+1} = u_i + \alpha_i d_i$ with α choosen from

$$\begin{aligned} 0 &= \frac{d}{d\alpha} f(u_i + \alpha d_i) = f'(u_i + \alpha d_i) \cdot d_i \\ &= (b - A(u_i + \alpha d_i), d_i) \\ &= (b - Au_i, d_i) - \alpha(Ad_i, d_i) \\ &= (r_i, d_i) - \alpha(Ad_i, d_i) \\ \alpha_i &= \frac{(r_i, d_i)}{(Ad_i, d_i)} \end{aligned}$$

- $u_{i+1} \in \text{span}\{d_0 \dots d_i\}$

Conjugate gradients

- Choose $d_0 \ldots d_i$ such that $\text{span}\{d_0 \ldots d_i\} = \mathcal{K}_i(A, r_0)$.
- Orthogonalize by Gram-Schmidt
- Result: short recursions!
- $u_i \in u_0 + \mathcal{K}_i(A, r_0)$ minimizes the energy norm of the error e_i: $||e_i||_A = (Ae_i, e_i)$.
- $r_{i+1} \perp \mathcal{K}_i(A, r_0)$
- There are at most N directions, so the method yields the exact solution after at most N iteration steps.

Conjugate gradients - the algorithm

Given initial value u_0, spd matrix A, right hand side b.

$$
\begin{aligned}
d_0 &= r_0 = b - Au_0 \\
\alpha_i &= \frac{(r_i, r_i)}{(Ad_i, d_i)} \\
u_{i+1} &= u_i + \alpha_i d_i \\
r_{i+1} &= r_i - \alpha_i A d_i \\
\beta_{i+1} &= \frac{(r_{i+1}, r_{i+1})}{(r_i, r_i)} \\
d_{i+1} &= r_{i+1} + \beta_{i+1} d_i
\end{aligned}
$$

At the i-th step, the algorithm yields the element from $e_0 + \mathcal{K}_i$ with the minimum energy error.

Theorem The convergence rate of the method is

$$
||e_i||_A \leq 2 \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^i ||e_0||_A
$$

where $\kappa = \frac{\lambda_{max}(A)}{\lambda_{min}(A)}$ is the spectral condition number of A.

Preconditioning

Let M be spd, and spectrally equivalent to A, and assume that $\kappa(M^{-1}A) << \kappa(A)$.

Let E be such that $M = EE^T$, e.g. its Cholesky factorization. Then, $\sigma(M^{-1}A) = \sigma(E^{-1}AE^{-T})$:

Assume $M^{-1}Au = \lambda u$. We have

$$(E^{-1}AE^{-T})(E^T u) = (E^T E^{-T})E^{-1}Au = E^T M^{-1}Au = \lambda E^T u$$

$\Leftrightarrow$ $E^T u$ is an eigenvector of $E^{-1}AE^{-T}$ with eigenvalue λ.

Preconditioned CG I

Now we can use the CG algorithm for the preconditioned system

$$E^{-1}AE^{-T}\tilde{x} = E^{-1}b$$

with $\tilde{u} = E^T u$

$$\begin{aligned}
\tilde{d}_0 &= \tilde{r}_0 = E^{-1}b - E^{-1}AE^{-T}u_0 \\
\alpha_i &= \frac{(\tilde{r}_i, \tilde{r}_i)}{(E^{-1}AE^{-T}\tilde{d}_i, \tilde{d}_i)} \\
\tilde{u}_{i+1} &= \tilde{u}_i + \alpha_i \tilde{d}_i \\
\tilde{r}_{i+1} &= \tilde{r}_i - \alpha_i E^{-1}AE^{-T}\tilde{d}_i \\
\beta_{i+1} &= \frac{(\tilde{r}_{i+1}, \tilde{r}_{i+1})}{(\tilde{r}_i, \tilde{r}_i)} \\
\tilde{d}_{i+1} &= \tilde{r}_{i+1} + \beta_{i+1}\tilde{d}_i
\end{aligned}$$

Not very practical as we need E

Preconditioned CG II

Assume $\tilde{r}_i = E^{-1} r_i$, $\tilde{d}_i = E^T d_i$, we get the equivalent algorithm

$$
\begin{aligned}
r_0 &= b - Au_0 \\
d_0 &= M^{-1} r_0 \\
\alpha_i &= \frac{(M^{-1} r_i, r_i)}{(Ad_i, d_i)} \\
u_{i+1} &= u_i + \alpha_i d_i \\
r_{i+1} &= r_i - \alpha_i A d_i \\
\beta_{i+1} &= \frac{(M^{-1} r_{i+1}, r_{i+1})}{(r_i, r_i)} \\
d_{i+1} &= M^{-1} r_{i+1} + \beta_{i+1} d_i
\end{aligned}
$$

It relies on the solution of the preconditioning system, the calculation of the matrix vector product and the calculation of the scalar product.

The convergence rate of the method is

$$
||e_i||_{E^{-1}AE^{-T}} \leq 2 \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^i ||e_0||_{E^{-1}AE^{-T}}
$$

where $\kappa = \frac{\gamma_{max}}{\gamma_{min}}$ comes from $\gamma_{min}(Mu, u) \leq (Au, u) \leq \gamma_{max}(Mu, u)$.

Issues and consequences

- Usually we stop the iteration when the residual r becomes small. However during the iteration, floating point errors occur which distort the calculations and lead to the fact that the accumulated residuals

$$r_{i+1} = r_i - \alpha_i A d_i$$

  give a much more optimistic picture on the state of the iteration than the real residual

$$r_{i+1} = b - A u_{i+1}$$

- The convergence rate estimate in terms of $\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}$ indeed provides a qualitatively better complexity estimate for the solution algorithm
- Always consider CG when solving symmetric positive definite linear systems iteratively

Unsymmetric problems

- By definition, CG is only applicable to symmetric problems.
- The biconjugate gradient (BICG) method provides a generalization:

Choose initial guess x_0, perform

$$
\begin{aligned}
r_0 &= b - A x_0 & \tilde{r}_0 &\neq 0 \\
p_0 &= r_0 & \tilde{p}_0 &= \tilde{r}_0 \\
\alpha_i &= \frac{(\tilde{r}_i, r_i)}{(\tilde{p}_i, A p_i)} & & \\
x_{i+1} &= x_i + \alpha_i p_i & \tilde{x}_{i+1} &= \tilde{x}_i + \alpha_i \tilde{p}_i \\
r_{i+1} &= r_i - \alpha_i A p_i & \tilde{r}_{i+1} &= \tilde{r}_i - \alpha_i \tilde{p}_i A^T \\
\beta_i &= \frac{(\tilde{r}_{i+1}, r_{i+1})}{(\tilde{r}_i, r_i)} & & \\
p_{i+1} &= r_{i+1} + \beta_i p_i & \tilde{p}_{i+1} &= \tilde{r}_{i+1} + \beta_i \tilde{p}_i
\end{aligned}
$$

- The two sequences produced by the algorithm are biorthogonal, i.e., $(\tilde{p}_i, A p_j) = (\tilde{r}_i, r_j) = 0$ for $i \neq j$.
- We have $r_i \in \mathcal{K}_i(A, r_0)$ and $\tilde{r}_i \in \mathcal{K}(A^T, \tilde{r}_0)$

Unsymmetric problems II

- BiCG is very unstable and additionally needs the transposed matrix vector product, it is seldomly used in practice
- There is as well a preconditioned variant of BiCG which also needs the transposed preconditioner.
- Main practical approaches to fix the situation:
  - "Conjugate gradients squared" (CGS, Sonneveld, 1989): Replace multiplication by A^T in BICG with multiplication by A, residual polynomial $p_{CGS} = p_{BICG}^2$.
  - "Stabilize" BiCG $\rightarrow$ BiCGstab (H. Van der Vorst, 1992), BiCGstab(l) (Sleijpen/Fokkema 1993)
  - Error minimization in Krylov subspace $\rightarrow$ "Generalized Minimum Residual" (GMRES, Saad/Schulz, 1986)
- Both CGS and BiCGstab can show erratic convergence behavior $\Rightarrow$ always try to stop iteration after residual check
- For GMRES one has to keep the full Krylov subspace, which is not possible in practice $\Rightarrow$ restart strategy.
- As in the case of CG, always combine preconditioners with Krylov subspace methods
- From my experience, BiCGstab is a good first guess