$\sim$

Orthogonalization methods

Scientific Computing Winter 2016/2017

Lecture 11

Jürgen Fuhrmann

juergen.fuhrmann@wias-berlin.de

made wit pandoc

$\sim$

Recap

The Gershgorin Circle Theorem

(everywhere, we assume $n \geq 2$)

Theorem Let A be an $n \times n$ (complex) matrix. Let

$$\Lambda_i = \sum_{\substack{j=1\dots n \\ j \neq i}} |a_{ij}|$$

If λ is an eigenvalue of A then there is r, $1 \leq r \leq n$ such that

$$|\lambda - a_{rr}| \leq \Lambda_r$$

Corollary: Any eigenvalue of A lies in the union of the disks defined by the Gershgorin cicles

$$\lambda \in \bigcup_{i=1\dots n} \{\mu \in \mathbb{C} : |\mu - |a_{ii}|| \leq \Lambda_i\}$$

Gershgorin Circle Theorem Corolary

Corollary:

$$\rho(A) \leq \max_{i=1\dots n} \sum_{j=1}^{n} |a_{ij}| = ||A||_\infty$$

$$\rho(A) \leq \max_{j=1\dots n} \sum_{i=1}^{n} |a_{ij}| = ||A||_1$$

Reducible and irreducible matrices

Definition A is *reducible* if there exists a permutation matrix P such that

$$PAP^T = \begin{pmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{pmatrix}$$

A is *irreducible* if it is not reducible.

Directed matrix graph:

- Nodes: $\mathcal{N} = \{N_i\}_{i=1\dots n}$
- Directed edges: $\mathcal{E} = \{\vec{N_k N_l} | a_{kl} \neq 0\}$

A is irreducible $\Leftrightarrow$ the matrix graph is connected, i.e. for each *ordered* pair N_i, N_j there is a path consisting of directed edges, connecting them.

Equivalently, for each i, j there is a sequence of nonzero matrix entries $a_{ik_1}, a_{k_1k_2}, \dots, a_{k_rj}$.

Taussky theorem

Theorem Let A be irreducible. Assume that the eigenvalue λ is a boundary point of the union of all the disks

$$\lambda \in \partial \bigcup_{i=1\dots n} \{\mu \in \mathbb{C} : |\mu - a_{ii}| \leq \Lambda_i\}$$

Then, all n Gershgorin circles pass through λ, i.e. for $i = 1 \dots n$,

$$|\lambda - a_{ii}| = \Lambda_i$$

Diagonally dominant matrices

Definition

- A is *diagonally dominant* if for $i = 1 \dots n$,

$$|a_{ii}| \geq \sum_{\substack{j=1\dots n \\ j\neq i}} |a_{ij}|$$

- A is *strictly diagonally dominant* (sdd) if for $i = 1 \dots n$,

$$|a_{ii}| > \sum_{\substack{j=1\dots n \\ j\neq i}} |a_{ij}|$$

- A is *irreducibly diagonally dominant* (idd) if A is irreducible, for $i = 1 \dots n$,

$$|a_{ii}| \geq \sum_{\substack{j=1\dots n \\ j\neq i}} |a_{ij}|$$

  and for at least one r, $1 \leq r \leq n$,

$$|a_{rr}| > \sum_{\substack{j=1\dots n \\ j\neq r}} |a_{rj}|$$

A very practical nonsingularity criterion

Theorem: Let A be strictly diagonally dominant or irreducibly diagonally dominant. Then A is nonsingular.

If in addition, if $a_{ii} > 0$ for $i = 1 \dots n$, then all real parts of the eigenvalues of A are positive:

$$\mathrm{Re}\lambda_i > 0, \quad i = 1 \dots n$$

Corollary: If A is symmetric, sdd or idd, with positive diagonal entries, it is positive definite.

Jacobi method convergence

Theorem: Let A be sdd or idd, and D its diagonal. Then

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

Corollary: Let A be sdd or idd, and D its diagonal. Assume that $a_{ii} > 0$ and $a_{ij} \leq 0$ for $i \neq j$. Then $\rho(I - D^{-1}A) < 1$, i.e. the Jacobi method converges.

Main Practical M-Matrix Criterion

Corollary: If

- A is strictly or irreducibly diagonally dominat
- $a_{ii} > 0$
- $a_{ij} \leq 0$ for $i \neq j$.

Then A is an M-Matrix, i.e. in addition to the sign pattern,

- A is nonsingular
- $A^{-1} \geq 0$

Regular splittings

- $A = M - N$ is a regular splitting if
  - M is nonsingular
  - M^{-1}, N are nonnegative, i.e. have nonnegative entries
- Regard the iteration $u_{k+1} = M^{-1}Nu_k + M^{-1}b$.
- We have $I - M^{-1}A = M^{-1}N$.

Theorem: Assume A is nonsingular, $A^{-1} \geq 0$, and $A = M - N$ is a regular splitting. Then $\rho(M^{-1}N) < 1$.

Corollary: $\rho(M^{-1}N) = \frac{\tau}{1+\tau}$ where $\tau = \rho(A^{-1}N)$.

Corollary: Let $A \geq 0$, $A = M_1 - N_1$ and $A = M_2 - N_2$ be regular splittings. If $N_2 \geq N_1 \geq 0$, then $1 > \rho(M_2^{-1}N_2) \geq \rho(M_1^{-1}N_1)$.

Application

Let A be an M-Matrix. Assume $A = D - E - F$, $D > 0$ diagonal, $E \geq 0$ lower triangular part, $F \geq 0$ upper triangular part.

- Jacobi method: $M = D$ is nonsingular, $M^{-1} \geq 0$. $N = E + F$ nonnegative $\Rightarrow$ convergence
- Gauss-Seidel: $M = D - E$ is an M-Matrix as $A \leq M$ and M has non-positive off-digonal entries. $N = F \geq 0$. $\Rightarrow$ convergence
- Comparison: $N_J \geq N_{GS} \Rightarrow$ Gauss-Seidel converges faster.

Intermediate Summary

- Given some matrix, we now have some nice recipies to establish nonsingularity and iterative method convergence:
- **Check if the matrix is irreducible.**
  This is mostly the case for elliptic and parabolic PDEs.
- **Check for if matrix is strictly or irreducibly diagonally dominant**.
  If yes, it is in addition nonsingular.
- **Check if main diagonal entries are positive and off-diagonal entries are nonpositive.**
  If yes, in addition, the matrix is an M-Matrix, its inverse is nonnegative, and elementary iterative methods converge.

Example: 1D finite volume matrix:

We assume $\alpha > 0$.

$$\begin{pmatrix} \alpha+\frac{1}{h} & -\frac{1}{h} & & & & & \\ -\frac{1}{h} & \frac{2}{h} & -\frac{1}{h} & & & & \\ & -\frac{1}{h} & \frac{2}{h} & -\frac{1}{h} & & & \\ & \ddots & \ddots & \ddots & \ddots & & \\ & & -\frac{1}{h} & \frac{2}{h} & -\frac{1}{h} & & \\ & & & -\frac{1}{h} & \frac{2}{h} & -\frac{1}{h} & \\ & & & & -\frac{1}{h} & \frac{1}{h}+\alpha \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_{N-2} \\ u_{N-1} \\ u_N \end{pmatrix} = \begin{pmatrix} \frac{h}{2}f_1+\alpha v_1 \\ hf_2 \\ hf_3 \\ \vdots \\ hf_{N-2} \\ hf_{N-1} \\ \frac{h}{2}f_N+\alpha v_n \end{pmatrix}$$

- idd
- main diagonal entries are positive and off-diagonal entries are nonpositive

So this matrix is nonsingular, has the M-property, and we can e.g. apply the Jacobi iterative method to solve it.

Moreover, due to $A^{-1} \geq 0$, for $f \geq 0$ and $v \geq 0$ it follows that $u \geq 0$.

Incomplete LU factorizations (ILU)

Idea (Varga, Buleev, 1960):

- fix a predefined zero pattern
- apply the standard LU factorization method, but calculate only those elements, which do not correspond to the given zero pattern
- Result: incomplete lower and upper triangular factors $\tilde{L}$, $\tilde{U}$, remainder R:

$$A = \tilde{L}\tilde{U} - R$$

- Problem: with complete LU factorization procedure, for any nonsingular matrix, the method is stable, i.e. zero pivots never occur. Is this true for the incomplete LU Factorization as well ?

Stability of ILU

Theorem (Saad, Th. 10.2): If A is an M-Matrix, then the algorithm to compute the incomplete LU factorization by omitting all entries except those belonging to a a given nonzero pattern resulting

$$A = \tilde{L}\tilde{U} - R$$

is stable. Moreover, $A = \tilde{L}\tilde{U} - R$ is a regular splitting.

ILU(0)

- Special case of ILU: ignore any fill-in.
- Representation:

$$M = \tilde{L}\tilde{U} = (\tilde{D} - E)\tilde{D}^{-1}(\tilde{D} - F)$$

- $\tilde{D}$ is a diagonal matrix (wich can be stored in one vector) which is calculated by the incomplete factorization algorithm.
- Setup in two loops of $O(n)$ complexity:

```
for i=1...n do
    d(i)=a(i,i)
end

for i=1...n do
     d(i)=1.0/d(i)
     for j=i+1 ... n do
         d(j)=d(j)-a(i,j)*d(i)*a(j,i)
     end
end
```

ILU(0)

Solve $Mu = v$ in one forward and one backward sweep.

```
for i=1...n do
     x=0
     for j=1 ... i-1 do
        x=x+a(i,j)*u(j)
     end
     u(i)=d(i)*(v(i)-x)
end

for i=n...1 do
     x=0
     for j=i+1...n do
        x=x+a(i,j)*u(j)
     end
     u(i)=u(i)-d(i)*x
```

ILU(0)

- Generally better convergence properties than Jacobi, Gauss-Seidel
- One can develop block variants
- Alternatives:
  - ILUM: ("modified"): add ignored off-diagonal entries to $\tilde{D}$
  - ILUT: zero pattern calculated dynamically based on drop tolerance
- Dependence on ordering
- Can be parallelized using graph coloring
- Not much theory: experiment for particular systems
- I recommend it as the default initial guess for a sensible preconditioner
- Incomplete Cholesky: symmetric variant of ILU

Preconditioners

- Leave this topic for a while now
- Hopefully, we well be able to discuss
  - Multigrid: gives $O(n)$ complexity in optimal situations
  - Domain decomposition: Structurally well suited for large scale parallelization

∼

More general iteration schemes

Generalization of iteration schemes

- Simple iterations converge slowly
- For most practical purposes, Krylov subspace methods are used.
- We will introduce one special case and give hints on practically useful more general cases
- Material after J. Shewchuk: An Introduction to the Conjugate Gradient Method Without the Agonizing Pain“

Solution of SPD system as a minimization procedure

Regard $Au = f$,where A is symmetric, positive definite. Then it defines a bilinear form $a : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$

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

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

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

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.

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

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

Conjugate directions II

$e_0 = u_0 - \hat{u}$ (such that $Ae_0 = -r_0$) can be represented in the basis of the search directions:

$$e_0 = \sum_{i=0}^{n-1} \delta_j d_j$$

Projecting onto d_k in the A scalar product gives

$$\begin{aligned}
(Ae_0, d_k) &= \sum_{i=0}^{n-1} \delta_j (Ad_j, d_k) \\
(Ae_0, d_k) &= \delta_k (Ad_k, d_k) \\
\delta_k &= \frac{(Ae_0, d_k)}{(Ad_k, d_k)} = \frac{(Ae_0 + \sum_{i<k} \alpha_i d_i, d_k)}{(Ad_k, d_k)} = \frac{(Ae_k, d_k}{(Ad_k, d_k)} \\
&= \frac{(r_k, d_k)}{(Ad_k, d_k)} \\
&= -\alpha_k
\end{aligned}$$

Conjugate directions III

Then,

$$
\begin{aligned}
e_i = e_0 + \sum_{j=0}^{i-1} \alpha_j d_j &= -\sum_{j=0}^{n-1} \alpha_j d_j + \sum_{j=0}^{i-1} \alpha_j d_j \\
&= -\sum_{j=i}^{n-1} \alpha_j d_j
\end{aligned}
$$

So, the iteration consists in component-wise suppression of the error, and it must converge after n steps.

Let $k \leq i$. A-projection on d_k gives

$$(Ae_i, d_k) = -\sum_{j=i}^{n-1} \alpha_j (Ad_j, d_k) = 0$$

Therefore, $r_i = Ae_i$ is orthogonal to $d_0 \ldots d_{i-1}$.

Conjugate directions IV

Looking at the error norm $||e_i||A$, the method yields the element with the minimum energy norm from all elements of the affine space $e_0 + \mathcal{K}_i$ where $\mathcal{K}_i = \mathrm{span}\{d_0, d_1 \dots d_{i-1}\}$

$$
\begin{aligned}
(Ae_i, e_i) &= \left(\sum_{j=i}^{n-1} \delta_j d_j, \sum_{j=i}^{n-1} \delta_j d_j \right) = \sum_{j=i}^{n-1} \sum_{k=i}^{n-1} \delta_j \delta_k (d_j, d_k) \\
&= \sum_{j=i}^{n-1} \delta_j^2 (d_j, d_j) \\
& \min_{e \in e_0 + \mathcal{K}_i} ||e||_A
\end{aligned}
$$

By what magic we can obtain these d_i?

Conjugate directions V

Furthermore, we have

$$
\begin{aligned}
u_{i+1} &= u_i + \alpha_i d_i \\
e_{i+1} &= e_i + \alpha_i d_i \\
Ae_{i+1} &= Ae_i + \alpha_i Ad_i \\
r_{i+1} &= r_i - \alpha_i Ad_i
\end{aligned}
$$

Gram-Schmidt Orthogonalization

- Assume we have been given some linearly independent vectors $v_0, v_1 \dots v_{n-1}$.
- Set $d_0 = v_0$
- Define

$$d_i = v_i + \sum_{k=0}^{i-1} \beta_{ik} d_k$$

- For $j < i$, A-project onto d_j and require orthogonality:

$$\begin{aligned}
(Ad_i, d_j) &= (Av_i, d_j) + \sum_{k=0}^{i-1} \beta_{ik}(Ad_k, d_j) \\
0 &= (Av_i, d_j) + \beta_{ij}(Ad_j, d_j) \\
\beta_{ij} &= -\frac{(Av_i, d_j)}{(Ad_j, d_j)}
\end{aligned}$$

- If v_i are the coordinate unit vectors, this is Gaussian elimination!
- If v_i are arbitrary, they all must be kept in the memory

Conjugate gradients (Hestenes, Stiefel, 1952)

As Gram-Schmidt builds up d_i from d_j, $j < i$, we can choose $v_i = r_i$ – the residuals built up during the conjugate direction process.

Let $\mathcal{K}_i = \mathrm{span}\{d_0 \dots d_{i-1}\}$. Then, $r_i \perp \mathcal{K}_i$

But d_i are built by Gram-Schmidt from the residuals, so we also have $\mathcal{K}_i = \mathrm{span}\{r_0 \dots r_{i-1}\}$ and $(r_i, r_j) = 0$ for $j < i$.

From $r_i = r_{i-1} - \alpha_{i-1} A d_{i-1}$ we obtain

$\mathcal{K}_i = \mathcal{K}_{i-1} \cup \mathrm{span}\{A d_{i-1}\}$

This gives two other representations of $\mathcal{K}_i$:

$$\begin{aligned}\mathcal{K}_i &= \mathrm{span}\{d_0, Ad_0, A^2 d_0, \dots, A^{i-1} d_0\} \\ &= \mathrm{span}\{r_0, Ar_0, A^2 r_0, \dots, A^{i-1} r_0\}\end{aligned}$$

Such type of subspace of $\mathbb{R}^n$ is called *Krylov subspace*, and orthogonalization methods are more often called *Krylov subspace methods*.

Conjugate gradients II

Look at Gram-Schmidt under these conditions. The essential data are (setting $v_i = r_i$ and using $j < i$) $\beta_{ij} = -\frac{(Ar_i, d_j)}{(Ad_j, d_j)} = -\frac{(Ad_j, r_i)}{(Ad_j, d_j)}$.

Then, for $j < i$:

$$\begin{aligned}
r_{j+1} &= r_j - \alpha_j A d_j \\
(r_{j+1}, r_i) &= (r_j, r_i) - \alpha_j (Ad_j, r_i) \\
\alpha_j (Ad_j, r_i) &= (r_j, r_i) - (r_{j+1}, r_i) \\
(Ad_j, r_i) &= \begin{cases} -\frac{1}{\alpha_j}(r_{j+1}, r_i), & j+1 = i \\ \frac{1}{\alpha_j}(r_j, r_i), & j = i \\ 0, & \text{else} \end{cases} = \begin{cases} -\frac{1}{\alpha_{i-1}}(r_i, r_i), & j+1 = i \\ \frac{1}{\alpha_i}(r_i, r_i), & j = i \\ 0, & \text{else} \end{cases} \\
\beta_{ij} &= \begin{cases} \frac{1}{\alpha_{i-1}} \frac{(r_i, r_i)}{(Ad_{i-1}, d_{i-1})}, & j+1 = i \\ 0, & \text{else} \end{cases}
\end{aligned}$$

Conjugate gradients III

For Gram-Schmidt we defined (replacing v_i by r_i):

$$\begin{aligned} d_i &= r_i + \sum_{k=0}^{i-1} \beta_{ik} d_k \\ &= r_i + \beta_{i,i-1} d_{i-1} \end{aligned}$$

So, the new orthogonal direction depends only on the previous orthogonal direction and the current residual. We don't have to store old residuals or search directions. In the sequel, set $\beta_i := \beta_{i,i-1}$.

We have

$$\begin{aligned} d_{i-1} &= r_{i-1} + \beta_{i-1} d_{i-2} \\ (d_{i-1}, r_{i-1}) &= (r_{i-1}, r_{i-1}) + \beta_{i-1}(d_{i-2}, r_{i-1}) \\ &= (r_{i-1}, r_{i-1}) \\ \beta_i &= \frac{1}{\alpha_{i-1}} \frac{(r_i, r_i)}{(Ad_{i-1}, d_{i-1})} = \frac{(r_i, r_i)}{(d_{i-1}, r_{i-1})} \\ &= \frac{(r_i, r_i)}{(r_{i-1}, r_{i-1})} \end{aligned}$$

Conjugate gradients IV - 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.

Preconditioning

We discussed all these nice preconditioners - GS, Jacobi, ILU, may be there are more of them. Are they of any help here ?

Let M be spd. We can try to solve $M^{-1}Au = M^{-1}b$ instead of the original system.

But in general, $M^{-1}A$ is neither symmetric, nor definite. But there is a trick:

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

$$(E^{-1}AE^{-T})(E^Tu) = (E^TE^{-T})E^{-1}Au = E^TM^{-1}Au = \lambda E^Tu$$

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

Good preconditioner: $M \approx A$ in the sense that $\kappa(M^{-1}A) << \kappa(A)$.

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.

A few issues

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

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

C++ implementation

```
template < class Matrix, class Vector, class Preconditioner, class Real >
int  CG(const Matrix &A, Vector &x, const Vector &b,
   const Preconditioner &M, int &max_iter, Real &tol)
{ Real resid;
  Vector p, z, q;
  Vector alpha(1), beta(1), rho(1), rho_1(1);
  Real normb = norm(b);
  Vector r = b - A*x;
  if (normb == 0.0)   normb = 1;
  if ((resid = norm(r) / normb) <= tol) {
    tol = resid;
    max_iter = 0;
    return 0;
  }
  for (int i = 1; i <= max_iter; i++) {
    z = M.solve(r);
    rho(0) = dot(r, z);
    if (i == 1)
      p = z;
    else {
      beta(0) = rho(0) / rho_1(0);
      p = z + beta(0) * p;
    }
    q = A*p;
    alpha(0) = rho(0) / dot(p, q);
    x += alpha(0) * p;
    r -= alpha(0) * q;
    if ((resid = norm(r) / normb) <= tol) {
      tol = resid;
      max_iter = i;
      return 0;
    }
    rho_1(0) = rho(0);
  }
  tol = resid;   return 1;
}
```

C++ implementation II

- Available from `http://www.netlib.org/templates/cpp//cg.h`
- Slightly adapted for numcxx
- Available in numxx in the namespace netlib.

Unsymmetric problems

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

Choose initial guess x_0, perform

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

The two sequences produced by the algorithm are biorthogonal, i.e., $(\hat{p}_i, A p_j) = (\hat{r}_i, r_j) = 0$ for $i \neq j$.

Unsymmetric problems II

- BiCG is very unstable an 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:
  - "Stabilize" BiCG $\rightarrow$ BiCGstab
  - tweak CG $\rightarrow$ Conjugate gradients squared (CGS)
  - Error minimization in Krylov subspace $\rightarrow$ Generalized Minimum Residual (GMRES)
- Both CGS and BiCGstab can show rather erratic convergence behavior
- For GMRES one has to keep the full Krylov subspace, which is not possible in practice $\Rightarrow$ restart strategy.
- From my experience, BiCGstab is a good first guess

Plan for next lectures

- Move on to higher dimensional (2D) discretiztion methods:
  - Domain triangulation
  - Partial differential equations
  - Finite volume method
  - Finite element method
- Aim: working with the methods introduced on 2D systems.

Next time

Special Guest: 斯杭 (Hang Si) from Weierstrass Institute, author of the tetrahedral mesh generator TetGen.