Numerical Methods for
Hyperbolic and Parabolic Conservation Laws
Linear System Solver
Part I: Conjugate Gradients Method

Andreas Meister

UMBC, Department of Mathematics and Statistics
Outline

- Methods for symmetric, positive definite Matrices
  - Method of steepest descent
  - Method of conjugate directions
  - CG-scheme

- Methods for non-singular Matrices
  - GMRES
  - BiCG, CGS and BiCGSTAB

- Preconditioning
  - ILU, IC, GS, SGS, ...
We consider

\[ Ax = b \]

with given data \( A \in \mathbb{R}^{n \times n} \), \( b \in \mathbb{R}^n \).

### Splitting methods

- Looking for approximations
  \[ x_m \in \mathbb{R}^n \]

### Projection methods

- Looking for approximations
  \[ x_m \in x_0 + K_m \subset \mathbb{R}^n \]
  \[ \dim K_m = m \leq n \]

### Numerical algorithm

- Numerical algorithm (orthogonality constraint)
  \[ x_{m+1} = Mx_m + Nb \]
  \[ b - Ax_m \perp L_m \subset \mathbb{R}^n \]
  \[ \dim L_m = m \leq n \]
Projection method & Krylov subspace approach

Example

\[ A = I \in \mathbb{R}^{2\times2}, \quad x_0 = 0 \in \mathbb{R}^2 \]

- \( K_1 = L_1 \) (Orthogonal projection method)
Projection method & Krylov subspace approach

Example

\( A = I \in \mathbb{R}^{2\times2} \), \( x_0 = 0 \in \mathbb{R}^2 \)

- \( K_1 = L_1 \) (Orthogonal projection method)
- \( K_1 \neq L_1 \) (Skew projection method)
Krylov subspace approach:

Projection method based on

\[ K_m = K_m(A, r_0) = \text{span}\{r_0, Ar_0, \ldots, A^{m-1}r_0\}, \]

with \( r_0 = b - Ax_0 \) is called Krylov subspace method.
Methods for symmetric, positive definite matrices

Basic idea:
Minimize the function

$$F(x) = \frac{1}{2} (Ax, x) - (b, x)$$

with respect to specific search directions

$$p_0, p_1, \ldots \in \mathbb{R}^n \setminus \{0\}.$$ 

Procedure:

- Choose $$x_0 \in \mathbb{R}^n$$ and $$p_0, p_1, \ldots \in \mathbb{R}^n \setminus \{0\}.$$ 
- For $$m = 0, 1, \ldots$$ we calculate $$x_{m+1}$$ such that
  $$F(x_{m+1}) \leq F(y) \quad \forall y \in x_m + \text{span}\{p_m\}$$
  $$\implies x_{m+1} = \arg \min_{\lambda \in \mathbb{R}} F(x_m + \lambda p_m)$$
  $$= f_{x_m,p_m}(\lambda)$$

Andreas Meister (UMBC) Methods for Conservation Laws Fall 2013 7 / 43
Methods for symmetric, positive definite matrices

Questions:

1. Does \( x^* = A^{-1}b \) represent the global minimum of \( F \)?  Yes

2. How do we calculate \( \lambda \in \mathbb{R} \)?

Concerning 1)

\[
F(x) = \frac{1}{2}(Ax, x) - (b, x)
\]

\[
\nabla F(x) = \frac{1}{2}(A + A^T)x - b
\]

A symm. \( \Rightarrow \)

\[
Ax - b
\]

\[
\nabla^2 F(x) = A \quad \text{A pos.def.} \quad \text{F is a convex mapping}
\]

\[
\nabla F(x) = 0 \iff x = A^{-1}b
\]
Questions:

1. Does $x^* = A^{-1} b$ represent the global minimum of $F$?  Yes
2. How do we calculate $\lambda \in \mathbb{R}$?

Concerning 1)

\[
F(x) = \frac{1}{2} (Ax, x) - (b, x)
\]

\[
\nabla F(x) = \frac{1}{2} (A + A^T)x - b
\]

$A$ symm. \quad \Rightarrow \quad Ax - b$

\[
\nabla^2 F(x) = A \quad \text{A pos.def.} \quad \Rightarrow \quad F \text{ is a convex mapping}
\]

\[
\nabla F(x) = 0 \quad \iff \quad x = A^{-1} b
\]
Methods for symmetric, positive definite matrices

Questions:

1. Does \( x^* = A^{-1}b \) represent the global minimum of \( F \)?  Yes

2. How do we calculate \( \lambda \in \mathbb{R} \)?

Concerning 1)

\[
F(x) = \frac{1}{2} (Ax, x) - (b, x)
\]

\[\Rightarrow \nabla F(x) = \frac{1}{2}(A + A^T)x - b\]

A symm. \[\Rightarrow Ax - b\]

\[
\nabla^2 F(x) = A \quad \text{A pos.def.} \quad F \text{ is a convex mapping}
\]

\[
\nabla F(x) = 0 \quad \iff \quad x = A^{-1}b
\]

Andreas Meister (UMBC)
Methods for symmetric, positive definite matrices

Questions:
1. Does \(x^* = A^{-1}b\) represent the global minimum of \(F\)? Yes
2. How do we calculate \(\lambda \in \mathbb{R}\) ?

Conc. 2)  
\[
f_{x,p}(\lambda) = \frac{1}{2}(Ax + \lambda p, x + \lambda p) - (b, x + \lambda p)
\]
\[
= F(x) + \lambda(Ax - b, p) + \frac{1}{2}\lambda^2(Ap, p)
\]
\[
f'_{x,p}(\lambda) = (Ax - b, p) + \lambda(Ap, p)
\]
\[
f''_{x,p}(\lambda) = (Ap, p) > 0 \quad \text{für} \quad p \neq 0
\]

Thus, \(f_{x,p}\) is convex and the optimal \(\lambda\) is given in the form
\[
f'_{x,p}(\lambda) = 0 \iff \lambda = \frac{(b - Ax, p)}{(Ap, p)}.
\]
Methods for symmetric, positive definite matrices

Questions:

1. Does $x^* = A^{-1}b$ represent the global minimum of $F$? Yes

2. How do we calculate $\lambda \in \mathbb{R}$?

$$\lambda = \frac{(b - Ax_m, p_m)}{(Ap_m, p_m)}$$

Conc. 2)

$$f_{x,p}(\lambda) = \frac{1}{2}(A(x + \lambda p), x + \lambda p) - (b, x + \lambda p)$$

$$= F(x) + \lambda(Ax - b, p) + \frac{1}{2}\lambda^2(Ap, p)$$

$$f'_{x,p}(\lambda) = (Ax - b, p) + \lambda(Ap, p)$$

$$f''_{x,p}(\lambda) = (Ap, p) > 0 \text{ für } p \neq 0$$

Thus, $f_{x,p}$ is convex and the optimal $\lambda$ is given in the form

$$f'_{x,p}(\lambda) = 0 \iff \lambda = \frac{(b - Ax, p)}{(Ap, p)}.$$
Questions:

1. Does \( x^* = A^{-1} b \) represent the global minimum of \( F \)?  Yes

2. How do we calculate \( \lambda \in \mathbb{R} \)?

\[
\lambda = \frac{(b - Ax_m, p_m)}{(Ap_m, p_m)}
\]

Conc. 2)

\[
f_{x, p}(\lambda) = \frac{1}{2}(A(x + \lambda p), x + \lambda p) - (b, x + \lambda p)
\]

\[
= F(x) + \lambda(Ax - b, p) + \frac{1}{2} \lambda^2(Ap, p)
\]

\[
f'_{x, p}(\lambda) = (Ax - b, p) + \lambda(Ap, p)
\]

\[
f''_{x, p}(\lambda) = (Ap, p) > 0 \quad \text{für} \quad p \neq 0
\]

Thus, \( f_{x, p} \) is convex and the optimal \( \lambda \) is given in the form

\[
f'_{x, p}(\lambda) = 0 \iff \lambda = \frac{(b - Ax, p)}{(Ap, p)}.
\]
Residual

The vector \( r = b - Ax \) is called residual (vector).

Algorithm:
- Choose \( x_0 \in \mathbb{R}^n \) and \( p_0, p_1, \ldots \in \mathbb{R}^n \setminus \{0\} \)
- For \( m = 0, 1, \ldots \)
  \[
  r_m = b - Ax_m \\
  \lambda_m = \frac{(r_m, p_m)}{(Ap_m, p_m)} \\
  x_{m+1} = x_m + \lambda_m p_m
  \]

Problem:
- Specification of the search direction \( p_0, p_1, \ldots \)
Method of steepest descent

Basic idea:
Choose the **optimal** search direction in the **local** sense

\[ \tilde{\mathbf{p}}_m = -\nabla F(x_m) = -(A\mathbf{x}_m - \mathbf{b}) = r_m \]

Normalizing the search direction:

\[ \mathbf{p}_m = \frac{\tilde{\mathbf{p}}_m}{\|\tilde{\mathbf{p}}_m\|_2} = \frac{r_m}{\|r_m\|_2} \]

Stopping criterion: \( r_m = 0 \)
Method of steepest descent

Algorithm:

- Choose $x_0 \in \mathbb{R}^n$
- For $m = 0, 1, \ldots$
  
  \[ r_m = b - Ax_m \]

  If $r_m \neq 0$
  
  \[ \lambda_m = \frac{\|r_m\|_2^2}{(Ar_m, r_m)} \]

  \[ x_{m+1} = x_m + \lambda_m r_m \]

Example

\[ A = \begin{pmatrix} 2 & 0 \\ 0 & 10 \end{pmatrix}, \quad b = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad x_0 = \begin{pmatrix} 4 \\ \sqrt{1.8} \end{pmatrix} \]
### Method of steepest descent

<table>
<thead>
<tr>
<th>$m$</th>
<th>$x_{m,1}$</th>
<th>$x_{m,2}$</th>
<th>$\varepsilon_m := | x_m - x^* |_A$</th>
<th>$\varepsilon_m / \varepsilon_{m-1}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>4.000000e+00</td>
<td>1.341641e+00</td>
<td>7.071068e+00</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2.987552e+00</td>
<td>-3.562863e-01</td>
<td>4.372680e+00</td>
<td>6.183904e-01</td>
</tr>
<tr>
<td>2</td>
<td>1.529627e+00</td>
<td>5.130523e-01</td>
<td>2.704023e+00</td>
<td>6.183904e-01</td>
</tr>
<tr>
<td>3</td>
<td>1.142460e+00</td>
<td>-1.362463e-01</td>
<td>1.672142e+00</td>
<td>6.183904e-01</td>
</tr>
<tr>
<td>4</td>
<td>5.849394e-01</td>
<td>1.961946e-01</td>
<td>1.034036e+00</td>
<td>6.183904e-01</td>
</tr>
<tr>
<td>5</td>
<td>4.368842e-01</td>
<td>-5.210148e-02</td>
<td>6.394382e-01</td>
<td>6.183904e-01</td>
</tr>
<tr>
<td>6</td>
<td>2.236847e-01</td>
<td>7.502613e-02</td>
<td>3.954224e-01</td>
<td>6.183904e-01</td>
</tr>
<tr>
<td>7</td>
<td>1.670674e-01</td>
<td>-1.992395e-02</td>
<td>2.445254e-01</td>
<td>6.183904e-01</td>
</tr>
<tr>
<td>8</td>
<td>8.553851e-02</td>
<td>2.869049e-02</td>
<td>1.512122e-01</td>
<td>6.183904e-01</td>
</tr>
<tr>
<td>9</td>
<td>6.388768e-02</td>
<td>-7.619051e-03</td>
<td>9.350814e-02</td>
<td>6.183904e-01</td>
</tr>
<tr>
<td>10</td>
<td>3.271049e-02</td>
<td>1.097143e-02</td>
<td>5.782453e-02</td>
<td>6.183904e-01</td>
</tr>
<tr>
<td>20</td>
<td>2.674941e-04</td>
<td>8.972025e-05</td>
<td>4.728673e-04</td>
<td>6.183904e-01</td>
</tr>
<tr>
<td>30</td>
<td>2.187466e-06</td>
<td>7.336985e-07</td>
<td>3.866930e-06</td>
<td>6.183904e-01</td>
</tr>
<tr>
<td>40</td>
<td>1.788827e-08</td>
<td>5.999910e-09</td>
<td>3.162230e-08</td>
<td>6.183904e-01</td>
</tr>
<tr>
<td>50</td>
<td>1.462836e-10</td>
<td>4.906500e-11</td>
<td>2.585953e-10</td>
<td>6.183904e-01</td>
</tr>
<tr>
<td>60</td>
<td>1.196252e-12</td>
<td>4.012351e-13</td>
<td>2.114695e-12</td>
<td>6.183904e-01</td>
</tr>
</tbody>
</table>
Contour lines (level curves) of $F(x) = \frac{1}{2}(Ax, x) - (b, x)$ w.r.t. the example

$$A = \begin{pmatrix} 2 & 0 \\ 0 & 10 \end{pmatrix}, \quad b = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

are determined by $F(x) = \frac{1}{2}(Ax, x) - (b, x) = x_1^2 + 5x_2^2$. 
Method of steepest descent

Problem:

Forgetfulness

- During the calculation of the new search direction $p_m$ we do not take into account any old search direction $p_0, \ldots, p_{m-1}$.

Aim:

- Choose linear independent $p_0, p_1, \ldots \in \mathbb{R}^n$
- Search into the direction $p_m$ and find the optimal approximation $x_{m+1} \in \mathbb{R}^n \text{ w.r.t. } x_0 + \text{span}\{p_0, \ldots, p_m\}$

Effect:

- At least for $m = n$ we obtain $x_m = A^{-1}b$
Method of steepest descent

**Problem:**

Forgetfulness

- During the calculation of the new search direction $p_m$ we do not take into account any old search direction $p_0, \ldots, p_{m-1}$.

**Aim:**

- Choose linear independent $p_0, p_1, \ldots \in \mathbb{R}^n$
- Search into the direction $p_m$ and find the optimal approximation $x_{m+1} \in \mathbb{R}^n$ w.r.t. $x_0 + \text{span}\{p_0, \ldots, p_m\}$

**Effect:**

- At least for $m = n$ we obtain $x_m = A^{-1}b$
Method of steepest descent

Problem:

Forgetfulness

- During the calculation of the new search direction $p_m$ we do not take into account any old search direction $p_0, \ldots, p_{m-1}$.

Aim:

- Choose linear independent $p_0, p_1, \ldots \in \mathbb{R}^n$
- Search into the direction $p_m$ and find the optimal approximation $x_{m+1} \in \mathbb{R}^n$ w.r.t. $x_0 + \text{span}\{p_0, \ldots, p_m\}$

Effect:

- At least for $m = n$ we obtain $x_m = A^{-1}b$
Method of steepest descent

Optimality:

Let $F : \mathbb{R}^n \rightarrow \mathbb{R}$. A vector $x \in \mathbb{R}^n$ is called

1. optimal w.r.t. $p \in \mathbb{R}^n \setminus \{0\}$, if

$$F(x) \leq F(x + \lambda p) \quad \forall \lambda \in \mathbb{R}.$$

2. optimal w.r.t. $U \subset \mathbb{R}^n$, if

$$F(x) \leq F(x + \xi) \quad \forall \xi \in U.$$
Method of steepest descent

How to investigate the optimality of \( x \in \mathbb{R}^n \) w.r.t. \( U \subset \mathbb{R}^n \)?

Consider

\[
\begin{align*}
  f_{x,\xi}(\lambda) &= F(x + \lambda \xi) \\
  f'_{x,\xi}(\lambda) &= (Ax - b, \xi) + \lambda (A\xi, \xi)
\end{align*}
\]

\( x \) is optimal w.r.t. \( U \ni \xi \neq 0 \)

\[\iff f'_{x,\xi}(0) = 0 \iff (Ax - b, \xi) = 0 \iff r \perp U \]
Method of steepest descent

How to maintain optimality?

Let $x_m \in x_0 + \text{span}\{p_0, \ldots, p_{m-1}\}$. 

$U_m :=$

If $x_m$ is optimal w.r.t. $U_m$ and

$x_{m+1} = x_m + \lambda_m p_m \quad , \quad \xi \in U_m$

$\implies \quad (b - Ax_{m+1}, \xi) = (b - Ax_m, \xi) - \lambda_m (Ap_m, \xi) = 0$

Condition:

$(Ap_m, p_i) = 0 \quad \text{für} \ i = 0, \ldots, m - 1$
Conjugate vectors

The vectors $p_0, \ldots, p_m \in \mathbb{R}^n \setminus \{0\}$ are called pairwise conjugated or A-orthogonal, if

$$(Ap_j, p_i) = 0 \text{ for all } i \neq j.$$
Method of steepest descent

How to obtain optimality w.r.t. $U_{m+1}$?

Optimality w.r.t. $U_m$:

$$(Ap_m, p_i) = 0, \quad i = 0, \ldots, m - 1$$

Optimality w.r.t. $p_m$:

$$U_{m+1} = \{U_m, p_m\} := \text{span}\{p_0, \ldots, p_{m-1}, p_m\}$$

$$0 = (b - Ax_{m+1}, p_m) = (b - Ax_m, p_m) - \lambda_m (Ap_m, p_m)$$

$$\implies \lambda_m = \frac{(b - Ax_m, p_m)}{(Ap_m, p_m)}$$
Method of steepest descent

Eastern and Christmas simultaneously?

or in other words

Are pairwise conjugated vectors always linear independent?

Proof by contradiction: Assume:
\[ p_0, \ldots, p_m \in \mathbb{R}^n \setminus \{0\} \text{ pairwise conjugated, } p_m \in \text{span}\{p_0, \ldots, p_{m-1}\} \]

\[ \implies p_m = \sum_{j=0}^{m-1} \alpha_j p_j \]

\[ \implies 0 = (A p_m, p_i) = \left( A \sum_{j=0}^{m-1} \alpha_j p_j, p_i \right) = \sum_{j=0}^{m-1} \alpha_j (A p_j, p_i) = \alpha_i (A p_i, p_i) \neq 0 \]

holds for \( i = 0, \ldots, m-1 \) \( \implies p_m = 0 \) Contradiction!!!

Answer: Yes, in the case that the matrix \( A \) is positive definite!
Method of steepest descent

Eastern and Christmas simultaneously?

or in other words

Are pairwise conjugated vectors always linear independent?

Proof by contradiction: Assume:

\( p_0, \ldots, p_m \in \mathbb{R}^n \setminus \{0\} \) pairw. conjugated, \( p_m \in \text{span}\{p_0, \ldots, p_{m-1}\} \)

\[ \implies p_m = \sum_{j=0}^{m-1} \alpha_j p_j \]

\[ \implies 0 = (Ap_m, p_i) = \left( A \sum_{j=0}^{m-1} \alpha_j p_j, p_i \right) = \sum_{j=0}^{m-1} \alpha_j (Ap_j, p_i) = \alpha_i (Ap_i, p_i) \]

holds for \( i = 0, \ldots, m-1 \) \( \implies p_m = 0 \) Contradiction!!

Answer: Yes, in the case that the matrix \( A \) is positive definite!
Method of steepest descent

Eastern and Christmas simultaneously?

or in other words

Are pairwise conjugated vectors always linear independent?

Proof by contradiction: Assume:
\[ p_0, \ldots, p_m \in \mathbb{R}^n \setminus \{0\} \text{ pairw. conjugated, } p_m \in \text{span}\{p_0, \ldots, p_{m-1}\} \]

\[ \implies p_m = \sum_{j=0}^{m-1} \alpha_j p_j \]

\[ \implies 0 = (Ap_m, p_i) = \left( A \sum_{j=0}^{m-1} \alpha_j p_j, p_i \right) = \sum_{j=0}^{m-1} \alpha_j (Ap_j, p_i) = \alpha_i (Ap_i, p_i) \]

holds for \( i=0, \ldots, m-1 \implies p_m = 0 \quad \text{Contradiction!!} \)

Answer: Yes, in the case that the matrix \( A \) is positive definite!
Summary:

- Choose **pairwise conjugated** search directions $p_0, \ldots, p_{n-1}$
- Calculate

\[
\lambda_m = \frac{(b - Ax_m, p_m)}{(Ap_m, p_m)} \quad m = 0, \ldots, n - 1
\]

⇒ Hence, one obtains at least $x_n = A^{-1}b$. 

Andreas Meister (UMBC)  
Methods for Conservation Laws  
Fall 2013  22 / 43
Method of conjugate directions

Algorithm (Method of conjugate directions)

- Choose $x_0 \in \mathbb{R}^n$ and $p_0, \ldots, p_{n-1}$ pairwise conjugated
- $r_0 = b - Ax_0$
- For $m = 0, \ldots, n - 1$

\[
\lambda_m = \frac{(r_m, p_m)}{(Ap_m, p_m)}
\]

\[
x_{m+1} = x_m + \lambda_m p_m
\]

\[
r_{m+1} = r_m - \lambda_m Ap_m
\]

Problems

- Calculation of $p_0, \ldots, p_{n-1}$
- error reduction (convergence history)
Method of conjugate directions

Algorithm (Method of conjugate directions)

- Choose $x_0 \in \mathbb{R}^n$ and $p_0, \ldots, p_{n-1}$ pairwise conjugated
- $r_0 = b - Ax_0$
- For $m = 0, \ldots, n - 1$

$$
\lambda_m = \frac{(r_m, p_m)}{(Ap_m, p_m)}
$$

$$
x_{m+1} = x_m + \lambda_m p_m
$$

$$
r_{m+1} = r_m - \lambda_m Ap_m
$$

Problems

- Calculation of $p_0, \ldots, p_{n-1}$
- error reduction (convergence history)
Method of conjugate gradients (CG)

<table>
<thead>
<tr>
<th>Method of steepest descent</th>
<th>Method of conjugate directions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basis: Gradients as search directions</td>
<td>Basis: Search directions are conjugated</td>
</tr>
<tr>
<td>Advantage: Associated w.t. problem</td>
<td>Advantage: Optimality</td>
</tr>
<tr>
<td>Disadvant.: Forgetfulness</td>
<td>Disadvant.: Convergence history</td>
</tr>
</tbody>
</table>

Method of conjugate gradients

| Basis: Use gradients for the calculation of conjugated search directions | Advantage: Associated with the problem Optimality |
Method of conjugate gradients (CG)

Ansatz:

\[ p_0 = r_0 \]

\[ p_m = r_m + \sum_{j=0}^{m-1} \alpha_j p_j \quad , \quad m = 1, \ldots, n-1 \]

- \( m \) degrees of freedom
- Calculation of \( \alpha_0, \ldots, \alpha_{m-1} \)

\[ 0 = (Ap_m, p_i) = (Ar_m, p_i) + \sum_{j=0}^{m-1} \alpha_j (Ap_j, p_i) \]

\[ \alpha_j = -\frac{(Ar_m, p_i)}{(Ap_i, p_i)} \quad , \quad i = 0, \ldots, m - 1 \]
Method of conjugate gradients (CG)

Algorithm

Choose $x_0 \in \mathbb{R}^n$ and define $p_0 = r_0 = b - Ax_0$

For $m = 0, \ldots, n - 1$

$$
\lambda_m = \frac{(r_m, p_m)}{(Ap_m, p_m)}
$$

$$
x_{m+1} = x_m + \lambda_m p_m
$$

$$
r_{m+1} = r_m - \lambda_m Ap_m
$$

$$
p_{m+1} = r_{m+1} - \sum_{j=0}^{m} \frac{(Ar_{m+1}, p_j)}{(Ap_j, p_j)} p_j
$$

Disadvantages

- Break down for $p_m = 0$
- Inapplicable in the case of large, sparse matrices
- Computational effort increasing from iteration step to iteration step
Method of conjugate gradients (CG)

Algorithm

- Choose $x_0 \in \mathbb{R}^n$ and define $p_0 = r_0 = b - Ax_0$
- For $m = 0, \ldots, n - 1$

\[
\lambda_m = \frac{(r_m, p_m)}{(Ap_m, p_m)}
\]
\[
x_{m+1} = x_m + \lambda_m p_m
\]
\[
r_{m+1} = r_m - \lambda_m Ap_m
\]
\[
p_{m+1} = r_{m+1} - \sum_{j=0}^{m} \frac{(Ar_{m+1}, p_j)}{(Ap_j, p_j)} p_j
\]

Disadvantages

- Break down for $p_m = 0$
- Inapplicable in the case of large, sparse matrices
- Computational effort increasing from iteration step to iteration step
Method of conjugate gradients (CG)

Algorithm

- Choose $x_0 \in \mathbb{R}^n$ and define $p_0 = r_0 = b - Ax_0$
- For $m = 0, \ldots, n - 1$
  
  $$\lambda_m = \frac{(r_m, p_m)}{(Ap_m, p_m)}$$
  
  $$x_{m+1} = x_m + \lambda_m p_m$$
  
  $$r_{m+1} = r_m - \lambda_m Ap_m$$
  
  $$p_{m+1} = r_{m+1} - \sum_{j=0}^{m} \frac{(Ar_{m+1}, p_j)}{(Ap_j, p_j)} p_j$$

Disadvantages

- Break down for $p_m = 0$
- Inapplicable in the case of large, sparse matrices
- Computational effort increasing from iteration step to iteration step
Method of conjugate gradients (CG)

Algorithm

- Choose $x_0 \in \mathbb{R}^n$ and define $p_0 = r_0 = b - Ax_0$
- For $m = 0, \ldots, n - 1$
  
  $\lambda_m = \frac{(r_m, p_m)}{(Ap_m, p_m)}$
  
  $x_{m+1} = x_m + \lambda_m p_m$
  
  $r_{m+1} = r_m - \lambda_m Ap_m$
  
  $p_{m+1} = r_{m+1} - \sum_{j=0}^{m} \frac{(Ar_{m+1}, p_j)}{(Ap_j, p_j)} p_j$

Disadvantages

- Break down for $p_m = 0$
- Inapplicable in the case of large, sparse matrices
- Computational effort increasing from iteration step to iteration step
Method of conjugate gradients (CG)

Algorithm

- Choose $x_0 \in \mathbb{R}^n$ and define $p_0 = r_0 = b - Ax_0$
- For $m = 0, \ldots, n - 1$
  \[
  \lambda_m = \frac{(r_m, p_m)}{(Ap_m, p_m)}
  \]
  \[
  x_{m+1} = x_m + \lambda_m p_m
  \]
  \[
  r_{m+1} = r_m - \lambda_m Ap_m
  \]
  \[
  p_{m+1} = r_{m+1} - \sum_{j=0}^{m} \frac{(Ar_{m+1}, p_j)}{(Ap_j, p_j)} p_j
  \]

Disadvantages

- Break down for $p_m = 0$
- Inapplicable in the case of large, sparse matrices
- Computational effort increasing from iteration step to iteration step
Properties and consequences

1. \( U_m := \text{span}\{p_0, \ldots, p_{m-1}\} = \text{span}\{r_0, Ar_0, \ldots, A^{m-1}r_0\} \)

\[
x_m = x_{m-1} + \lambda_{m-1}p_{m-1} = \ldots = x_0 + \sum_{j=0}^{m-1} \lambda_j p_j
\]

\( \implies x_m \in x_0 + U_m = x_0 + K_m \)

2. \( r_m \perp U_m \)

\[
r_m = b - Ax_m \perp U_m = K_m
\]

\( \implies \text{Orthogonal Krylov subspace method} \)
Method of conjugate gradients (CG)

**Properties and consequences**

1. \( x_m = A^{-1} b \iff r_m = 0 \iff p_m = 0 \)
   - \( p_m = 0 \iff \) Stopping criterion

2. \( (A r_{m+1}, p_j) = 0 \), \( j = 0, \ldots, m - 1 \)
   - \( p_{m+1} = r_{m+1} - \sum_{j=0}^{m} \frac{(A r_{m+1}, p_j)}{(A p_j, p_j)} p_j \)
     - \( = r_{m+1} - \frac{(A r_{m+1}, p_m)}{(A p_m, p_m)} p_m \)
   - Applicable for large sparse systems
   - Low computational effort
Method of conjugate gradients (CG)

**Algorithm**

- Choose $x_0 \in \mathbb{R}^n$ and define $p_0 = r_0 = b - Ax_0$
- For $m = 0, \ldots, n - 1$
  
  If $p_m \neq 0$ then

  $$
  \lambda_m = \frac{(r_m, p_m)}{(Ap_m, p_m)}
  $$

  $$
  x_{m+1} = x_m + \lambda_m p_m
  $$

  $$
  r_{m+1} = r_m - \lambda_m Ap_m
  $$

  $$
  p_{m+1} = r_{m+1} - \frac{(Ar_{m+1}, p_m)}{(Ap_m, p_m)} p_m
  $$

  else STOP
Method of conjugate gradients (CG)

Example: 1-D Poisson-Equation \( x'' = b \)

\[ D = [0, 1] \quad , \quad h = 1/8 \quad , \quad N = 7 \]

\[ \mathbb{R}^{7 \times 7} \ni A = \text{tridiag} \{-64, 128, -64\} \]

\[ \mathbb{R}^{7} \ni b = (128, -448, 704, -832, 512, 128, 320)^T, \quad x_0 = 0 \]

<table>
<thead>
<tr>
<th>( m )</th>
<th>( x_{m,1} )</th>
<th>( x_{m,2} )</th>
<th>( x_{m,3} )</th>
<th>( x_{m,4} )</th>
<th>( x_{m,5} )</th>
<th>( x_{m,6} )</th>
<th>( x_{m,7} )</th>
<th>( |\vec{r}_m|_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>1336.36</td>
</tr>
<tr>
<td>1</td>
<td>0.58</td>
<td>-2.04</td>
<td>3.21</td>
<td>-3.79</td>
<td>2.33</td>
<td>0.58</td>
<td>1.46</td>
<td>363.57</td>
</tr>
<tr>
<td>2</td>
<td>-0.39</td>
<td>-1.72</td>
<td>2.81</td>
<td>-4.57</td>
<td>3.00</td>
<td>4.99</td>
<td>4.26</td>
<td>252.76</td>
</tr>
<tr>
<td>3</td>
<td>-0.01</td>
<td>-2.38</td>
<td>2.06</td>
<td>-3.53</td>
<td>4.87</td>
<td>6.07</td>
<td>6.25</td>
<td>153.30</td>
</tr>
<tr>
<td>4</td>
<td>-0.14</td>
<td>-2.88</td>
<td>2.57</td>
<td>-2.13</td>
<td>6.50</td>
<td>7.48</td>
<td>5.93</td>
<td>117.64</td>
</tr>
<tr>
<td>5</td>
<td>-0.70</td>
<td>-2.18</td>
<td>3.53</td>
<td>-1.12</td>
<td>7.65</td>
<td>7.81</td>
<td>6.27</td>
<td>103.52</td>
</tr>
<tr>
<td>6</td>
<td>0.13</td>
<td>-1.14</td>
<td>5.40</td>
<td>0.54</td>
<td>8.23</td>
<td>8.54</td>
<td>6.98</td>
<td>89.70</td>
</tr>
<tr>
<td>7</td>
<td>1.00</td>
<td>0.00</td>
<td>6.00</td>
<td>1.00</td>
<td>9.00</td>
<td>9.00</td>
<td>7.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>
Convection-Diffusion Equation

**Governing Equation**

\[ \beta \cdot \nabla u(x, y) - \epsilon \Delta u(x, y) = 0 \quad \text{on} \quad D = (0, 1) \times (0, 1) \]

with

\[ \beta = \alpha \begin{pmatrix} \cos \frac{\pi}{4} \\ \sin \frac{\pi}{4} \end{pmatrix} \quad \alpha, \epsilon \in \mathbb{R}_0^+ \]

**Boundary Conditions**

\[ u(x, y) = x^2 + y^2 \quad \text{for} \quad (x, y) \in \partial D \]

**Mesh**

\[ x_i = i \cdot h \quad \text{and} \quad y_j = j \cdot h \quad \text{for} \quad j = 0, \ldots, N + 1, \quad h = \frac{1}{N + 1} \]
Convection-Diffusion Equation

Discretization of Laplacian (Central Difference)

\[
\frac{\partial^2 u}{\partial x^2}(x_i, y_j) \approx \frac{1}{h^2}(u_{i+1,j} - 2u_{ij} + u_{i-1,j}) \\
\frac{\partial^2 u}{\partial y^2}(x_i, y_j) \approx \frac{1}{h^2}(u_{i,j+1} - 2u_{ij} + u_{i,j-1})
\]

Discretization of convective part (Backward Difference)

\[
\frac{\partial u}{\partial x}(x_i, y_j) \approx \frac{1}{h}(u_{i,j} - u_{i-1,j}) \\
\frac{\partial u}{\partial y}(x_i, y_j) \approx \frac{1}{h}(u_{i,j} - u_{i,j-1})
\]
## Testcases

<table>
<thead>
<tr>
<th></th>
<th>$\alpha$</th>
<th>$\epsilon$</th>
<th>Matrix properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test 1</td>
<td>0</td>
<td>1</td>
<td>Symmetric, positive definite</td>
</tr>
<tr>
<td>Test 2</td>
<td>0.1</td>
<td>1</td>
<td>Non-symmetric, non-singular</td>
</tr>
<tr>
<td>Test 3</td>
<td>1</td>
<td>0.1</td>
<td>Non-symmetric, non-singular</td>
</tr>
</tbody>
</table>

- **Number of unknowns:** $100 \times 100 = 10000$ $(N = 100)$
- **Stopping criterion:** $\|r_j\|_2 < 10^{-12}\|b\|$
Convection-Diffusion Equation

Numerical Solution of Test 3
Steepest Descent vs. Conjugate Gradient method

**Test 1: Pure Diffusion (\(\alpha = 0, \epsilon = 1\))**

<table>
<thead>
<tr>
<th></th>
<th>Number of Iterations</th>
<th>CPU Time (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Steepest Descent</td>
<td>47300</td>
<td>12506</td>
</tr>
<tr>
<td>Conjugate Gradient</td>
<td>344</td>
<td>100</td>
</tr>
</tbody>
</table>

![Graph showing the comparison between Steepest descent and Conjugate Gradient methods showing the convergence of the residual norm squared over the number of iterations](image.png)
Conjugate Gradients for Non-SPD Systems

Comparison of CG method for all three test case

<table>
<thead>
<tr>
<th>Test</th>
<th>$\alpha$</th>
<th>$\epsilon$</th>
<th>Number of Iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test 1</td>
<td>0</td>
<td>1</td>
<td>344</td>
</tr>
<tr>
<td>Test 2</td>
<td>0.1</td>
<td>1</td>
<td>631</td>
</tr>
<tr>
<td>Test 3</td>
<td>1</td>
<td>0.1</td>
<td>Convergence failed</td>
</tr>
</tbody>
</table>

Andreas Meister (UMBC)
Theorem

Let $A \in \mathbb{R}^{n \times n}$ be symmetric and positive definite. Then the error estimate

$$\|e_m\|_A \leq 2 \left( \frac{\sqrt{\text{cond}_2(A)} - 1}{\sqrt{\text{cond}_2(A)} + 1} \right)^m \|e_0\|_A,$$

holds with $e_m = x_m - A^{-1}b$, $x_m =$ approximate solution.

Properties: (A symm., positive definite)

1. **Eigenvalues:** $\lambda_n \geq \ldots \geq \lambda_1 > 0$,
   Eigenvectors: $\{v_1, \ldots, v_n\}$ ONB of $\mathbb{R}^n$

2. **Condition number:** $c := \text{cond}_2(A) = \|A\|_2 \|A^{-1}\|_2 = \frac{\lambda_n}{\lambda_1}$

3. **Weighted vector norm (energy norm):**

$$\|x\|_A = \sqrt{(Ax, x)}$$
Convergence properties of the CG-method

Relation between error- and residual vector:

Due to the relation

\[ r_m = b - Ax_m = -A(x_m - A^{-1}b) = -Ae_m \]

one obtains

\[ K_m = \text{span}\{r_0, Ar_0, \ldots, A^{m-1}r_0\} = \text{span}\{Ae_0, \ldots, A^me_0\}, \]

such that the approximate solution \( x_m \in x_0 + K_m \) reads

\[ x_m = x_0 + \sum_{i=1}^{m} c_i A^i e_0. \]
Convergence properties of the CG-method

Formulation of the error vector

\[ e_m = x_m - A^{-1}b = x_0 - A^{-1}b + \sum_{i=1}^{m} c_i A^i e_0 = p_m(A) e_0 \]

with \( p_m \in P_m^1 = \{ p \in P_m | p(O) = I \} \)

Equivalence to the minimization of the functional:

\[
F(x) = \frac{1}{2} (Ax, x) - (b, x) = \frac{1}{2} (Ax - b, x - A^{-1}b) - \frac{1}{2} (b, A^{-1}b) = A e - e + const.
\]

\[
= \frac{1}{2} ||e||_A^2 + const
\]

\[
x_m = \arg \min_{x \in x_0 + K_m} F(x) \iff ||e_m||_A = \min_{p \in P_m} ||p(A)e_0||_A
\]
Convergence properties of the CG-method

**Error estimate:**
\[ e_0 = \sum_{i=1}^{n} \alpha_i v_i \text{ with the ONB } v_1, \ldots, v_n \]

1. \[ \|e_0\|_A^2 = (Ae_0, e_0) = \left( \sum_{i=1}^{n} \lambda_i \alpha_i v_i, \sum_{i=1}^{n} \alpha_i v_i \right) = \sum_{i=1}^{n} \alpha_i^2 \lambda_i \]

2. \[ \|p(A)e_0\|_A^2 = \left( \sum_{i=1}^{n} p(\lambda_i)^2 \alpha_i^2 \lambda_i \right) \]

3. \[ \|e_m\|_A = \min_{p \in P_m^1} \left( \sum_{i=1}^{n} p(\lambda_i)^2 \alpha_i^2 \lambda_i \right)^{\frac{1}{2}} \leq \min_{p \in P_m^1} \max_{\lambda \in \{\lambda_1 \ldots \lambda_n\}} |p(\lambda)| \left( \sum_{i=1}^{n} \alpha_i^2 \lambda_i \right)^{\frac{1}{2}} \leq \min_{p \in P_m^1} \max_{\lambda \in [\lambda_1, \lambda_n]} |p(\lambda)| \|e_0\|_A \]
Convergence properties of the CG-method

Consideration of a suitable polynomial:

\[ \|e_m\|_A \leq \min_{p \in P_m^1} \max_{\lambda \in [\lambda_1, \lambda_n]} |p(\lambda)| \|e_0\|_A \]

**Case 1:** \( \lambda_1 \neq \lambda_n \)  
Tschebyscheff-Polynomials

\[ T_m(\lambda) = \cos(m \arccos \lambda), \ m \in \mathbb{N}_0 \quad (\lambda \in [-1, 1]) \]

1. \( |T_m(\lambda)| \leq 1 \)
2. \[ T_m(\lambda) = 2\lambda T_{m-1}(\lambda) + T_{m-2}(\lambda), \quad T_0(\lambda) = 1 \implies T_m \in P_m \]
3. \[ T_m \left( \frac{1}{2} \left( \lambda + \frac{1}{\lambda} \right) \right) = \left( \frac{1}{2} \left( \lambda^m + \frac{1}{\lambda^m} \right) \right) \]

\[ p_m(\lambda) := \frac{T_m \left( \frac{2\lambda - (\lambda_n + \lambda_1)}{\lambda_1 - \lambda_n} \right)}{T_m \left( \frac{\lambda_n + \lambda_1}{\lambda_n - \lambda_1} \right)} \quad (\text{2}) \in P_m^1 \]
Convergence properties of the CG-method

Utilizing

\[
\frac{\lambda_n + \lambda_1}{\lambda_n - \lambda_1} = \frac{\lambda_n}{\lambda_1} + 1 = \frac{c + 1}{c - 1} = \frac{1}{2} \left( \frac{\sqrt{c} + 1}{\sqrt{c} - 1} + \frac{\sqrt{c} - 1}{\sqrt{c} + 1} \right)
\]

one gets

\[
\frac{\|e_m\|_A}{\|e_0\|_A} \leq \max_{\lambda \in [\lambda_1, \lambda_n]} |p_m(\lambda)| \overset{(1)}{\leq} \max_{\lambda \in [\lambda_1, \lambda_n]} \left| \frac{1}{T_m\left(\frac{\lambda_n + \lambda_1}{\lambda_n - \lambda_1}\right)} \right|
\]

\[
= \left| T_m\left(\frac{\lambda_n + \lambda_1}{\lambda_n - \lambda_1}\right) \right|^{-1} \overset{(3)}{=} \left| 2 \left( \frac{\sqrt{c} - 1}{\sqrt{c} + 1} \right)^m \right|
\]

\[
\leq 2 \left( \frac{\sqrt{c} - 1}{\sqrt{c} + 1} \right)^m
\]
Consideration of a suitable polynomial:

\[ \|e_m\|_A \leq \min_{p \in P_m^1} \max_{\lambda \in [\lambda_1, \lambda_n]} |p(\lambda)| \|e_0\|_A \]

Case 2: \( \lambda_1 = \lambda_n \)

Taking account of

\[ c = \frac{\lambda_n}{\lambda_1} = 1. \]

we simply define

\[ p_m(\lambda) = 1 - \frac{\lambda}{\lambda_n} \in P_m^1. \]

Thus,

\[ \frac{\|e_m\|_A}{\|e_0\|_A} \leq \max_{\lambda \in [\lambda_1, \lambda_n]} |p_m(\lambda)| = 0 = 2 \left( \frac{\sqrt{c} - 1}{\sqrt{c} + 1} \right)^m \]