The Conjugate Gradient Method

Tom Lyche

University of Oslo

Norway

Plan for the day

- The method
- Algorithm
- Implementation of test problems
- Complexity
- Derivation of the method
- Convergence

The Conjugate gradient method

- Restricted to positive definite systems: $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, $\boldsymbol{A} \in \mathbb{R}^{n,n}$ positive definite.
- Generate $\{\boldsymbol{x}_k\}$ by $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k$,
- $\boldsymbol{p}_k$ is a vector, the search direction,
- α_k is a scalar determining the step length.
- In general we find the exact solution in at most n iterations.
- For many problems the error becomes small after a few iterations.
- Both a direct method and an iterative method.
- Rate of convergence depends on the square root of the condition number

The name of the game

- Conjugate means orthogonal; orthogonal gradients.
- But why gradients?
- Consider minimizing the quadratic function $Q : \mathbb{R}^n \to \mathbb{R}$ given by $Q(\boldsymbol{x}) := \frac{1}{2}\boldsymbol{x}^T \boldsymbol{A}\boldsymbol{x} - \boldsymbol{x}^T \boldsymbol{b}$.
- The minimum is obtained by setting the gradient equal to zero.
- $\nabla Q(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b} = \boldsymbol{0}$ linear system $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$
- Find the solution by solving $\boldsymbol{r} = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x} = \boldsymbol{0}$.
- The sequence $\{\boldsymbol{x}_k\}$ is such that $\{\boldsymbol{r}_k\} := \{\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_k\}$ is orthogonal with respect to the usual inner product in $\mathbb{R}^n$.
- The search directions are also orthogonal, but with respect to a different inner product.

The algorithm

- Start with some $\boldsymbol{x}_0$. Set $\boldsymbol{p}_0 = \boldsymbol{r}_0 = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_0$.
- For $k = 0, 1, 2, \ldots$
- $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k, \quad \alpha_k = \frac{\boldsymbol{r}_k^T \boldsymbol{r}_k}{\boldsymbol{p}_k^T \boldsymbol{A} \boldsymbol{p}_k}$
- $\boldsymbol{r}_{k+1} = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_{k+1} = \boldsymbol{r}_k - \alpha_k \boldsymbol{A}\boldsymbol{p}_k$
- $\boldsymbol{p}_{k+1} = \boldsymbol{r}_{k+1} + \beta_k \boldsymbol{p}_k, \quad \beta_k = \frac{\boldsymbol{r}_{k+1}^T \boldsymbol{r}_{k+1}}{\boldsymbol{r}_k^T \boldsymbol{r}_k}$

Example

$$\begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

- Start with $\boldsymbol{x}_0 = \boldsymbol{0}$.
- $\boldsymbol{p}_0 = \boldsymbol{r}_0 = \boldsymbol{b} = [1, 0]^T$
- $\alpha_0 = \frac{\boldsymbol{r}_0^T \boldsymbol{r}_0}{\boldsymbol{p}_0^T \boldsymbol{A} \boldsymbol{p}_0} = \frac{1}{2}$, $\boldsymbol{x}_1 = \boldsymbol{x}_0 + \alpha_0 \boldsymbol{p}_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \frac{1}{2} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1/2 \\ 0 \end{bmatrix}$
- $\boldsymbol{r}_1 = \boldsymbol{r}_0 - \alpha_0 \boldsymbol{A} \boldsymbol{p}_0 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} - \frac{1}{2} \begin{bmatrix} 2 \\ -1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1/2 \end{bmatrix}$, $\boldsymbol{r}_1^T \boldsymbol{r}_0 = 0$
- $\beta_0 = \frac{\boldsymbol{r}_1^T \boldsymbol{r}_1}{\boldsymbol{r}_0^T \boldsymbol{r}_0} = \frac{1}{4}$, $\boldsymbol{p}_1 = \boldsymbol{r}_1 + \beta_0 \boldsymbol{p}_0 = \begin{bmatrix} 0 \\ 1/2 \end{bmatrix} + \frac{1}{4} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1/4 \\ 1/2 \end{bmatrix}$,
- $\alpha_1 = \frac{\boldsymbol{r}_1^T \boldsymbol{r}_1}{\boldsymbol{p}_1^T \boldsymbol{A} \boldsymbol{p}_1} = \frac{2}{3}$,

  $\boldsymbol{x}_2 = \boldsymbol{x}_1 + \alpha_1 \boldsymbol{p}_1 = \begin{bmatrix} 1/2 \\ 0 \end{bmatrix} + \frac{2}{3} \begin{bmatrix} 1/4 \\ 1/2 \end{bmatrix} = \begin{bmatrix} 2/3 \\ 1/3 \end{bmatrix}$
- $\boldsymbol{r}_2 = \boldsymbol{0}$, exact solution.

Exact method and iterative method

- Orthogonality of the residuals implies that $\boldsymbol{x}_m$ is equal to the solution $\boldsymbol{x}$ of $A\boldsymbol{x} = \boldsymbol{b}$ for some $m \leq n$.
- For if $\boldsymbol{x}_k \neq \boldsymbol{x}$ for all $k = 0, 1, \ldots, n-1$ then $\boldsymbol{r}_k \neq \boldsymbol{0}$ for $k = 0, 1, \ldots, n-1$ is an orthogonal basis for $\mathbb{R}^n$. But then $\boldsymbol{r}_n \in \mathbb{R}^n$ is orthogonal to all vectors in $\mathbb{R}^n$ so $\boldsymbol{r}_n = \boldsymbol{0}$ and hence $\boldsymbol{x}_n = \boldsymbol{x}$.
- So the conjugate gradient method finds the exact solution in at most n iterations.
- The convergence analysis shows that $\|x - \boldsymbol{x}_k\|_A$ typically becomes small quite rapidly and we can stop the iteration with k much smaller that n.
- It is this rapid convergence which makes the method interesting and in practice an iterative method.

Conjugate Gradient Algorithm

[Conjugate Gradient Iteration] The positive definite linear system $\boldsymbol{Ax} = \boldsymbol{b}$ is solved by the conjugate gradient method. $\boldsymbol{x}$ is a starting vector for the iteration. The iteration is stopped when $||\boldsymbol{r}_k||_2/||\boldsymbol{r}_0||_2 \leq$ tol or $k >$ itmax. itm is the number of iterations used.

```
function [x,itm]=cg(A,b,x,tol,itmax) r=b-A*x; p=r; rho=r'*r;
rho0=rho; for k=0:itmax
    if sqrt(rho/rho0)<= tol^2
        itm=k; return
    end
    t=A*p; a=rho/(p'*t);
    x=x+a*p; r=r-a*t;
    rhos=rho; rho=r'*r;
    p=r+(rho/rhos)*p;
end itm=itmax+1;
```

A family of test problems

We can test the methods on the Kronecker sum matrix

$$\boldsymbol{A} = \boldsymbol{C}_1 \otimes \boldsymbol{I} + \boldsymbol{I} \otimes \boldsymbol{C}_2 = \begin{bmatrix} \boldsymbol{C}_1 & & & & \\ \boldsymbol{C}_1 & & & & \\ & & \ddots & & \\ & & & \boldsymbol{C}_1 & \\ & & & & \boldsymbol{C}_1 \end{bmatrix} + \begin{bmatrix} c\boldsymbol{I} & b\boldsymbol{I} & & & \\ b\boldsymbol{I} & c\boldsymbol{I} & b\boldsymbol{I} & & \\ & \ddots & \ddots & \ddots & \\ & & b\boldsymbol{I} & c\boldsymbol{I} & b\boldsymbol{I} \\ & & & b\boldsymbol{I} & c\boldsymbol{I} \end{bmatrix},$$

where $\boldsymbol{C}_1 = \text{tridiag}_m(a, c, a)$ and $\boldsymbol{C}_2 = \text{tridiag}_m(b, c, b)$.
Positive definite if $c > 0$ and $c \geq |a| + |b|$.

$m = 3, n = 9$

$$\boldsymbol{A} = \left[\begin{array}{ccc|ccc|ccc} \mathbf{2c} & a & 0 & b & 0 & 0 & 0 & 0 & 0 \\ a & \mathbf{2c} & a & 0 & b & 0 & 0 & 0 & 0 \\ 0 & a & \mathbf{2c} & 0 & 0 & b & 0 & 0 & 0 \\ \hline b & 0 & 0 & \mathbf{2c} & a & 0 & b & 0 & 0 \\ 0 & b & 0 & a & \mathbf{2c} & a & 0 & b & 0 \\ 0 & 0 & b & 0 & a & \mathbf{2c} & 0 & 0 & b \\ \hline 0 & 0 & 0 & b & 0 & 0 & \mathbf{2c} & a & 0 \\ 0 & 0 & 0 & 0 & b & 0 & a & \mathbf{2c} & a \\ 0 & 0 & 0 & 0 & 0 & b & 0 & a & \mathbf{2c} \end{array}\right]$$

- $b = a = -1, c = 2$: Poisson matrix
- $b = a = 1/9, c = 5/18$: Averaging matrix

Averaging problem

- $\lambda_{jk} = 2c + 2a\cos(j\pi h) + 2b\cos(k\pi h),\ \ j,k = 1,2,\ldots,m.$
- $a = b = 1/9$, $c = 5/18$
- $\lambda_{max} = \frac{5}{9} + \frac{4}{9}\cos(\pi h)$, $\lambda_{min} = \frac{5}{9} - \frac{4}{9}\cos(\pi h)$
- $\text{cond}_2(\boldsymbol{A}) = \frac{\lambda_{max}}{\lambda_{min}} = \frac{5+4\cos(\pi h)}{5-4\cos(\pi h)} \leq 9$.

2D formulation for test problems

- $\boldsymbol{V} = \text{vec}(\boldsymbol{x})$. $\boldsymbol{R} = \text{vec}(\boldsymbol{r})$, $\boldsymbol{P} = \text{vec}(\boldsymbol{p})$
- $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \Longleftrightarrow \boldsymbol{D}\boldsymbol{V} + \boldsymbol{V}\boldsymbol{E} = h^2\boldsymbol{F}$,
- $\boldsymbol{D} = \text{tridiag}(a, c, a) \in \mathbb{R}^{m,m}$, $\boldsymbol{E} = \text{tridiag}(b, c, b) \in \mathbb{R}^{m,m}$
- $\text{vec}(\boldsymbol{A}\boldsymbol{p}) = \boldsymbol{D}\boldsymbol{P} + \boldsymbol{P}\boldsymbol{E}$

Testing

[Testing Conjugate Gradient] $\boldsymbol{A} = \text{trid}(a,c,a,m) \otimes \boldsymbol{I}_m + \boldsymbol{I}_m \otimes \text{trid}(b,c,b,m) \in \mathbb{R}^{m^2,m^2}$

```
function [V,it]=cgtest(m,a,b,c,tol,itmax)
h=1/(m+1); R=h*h*ones(m);
D=sparse(tridiagonal(a,c,a,m)); E=sparse(tridiagonal(b,c,b,m));
V=zeros(m,m); P=R; rho=sum(sum(R.*R)); rho0=rho;
for k=1:itmax
    if sqrt(rho/rho0)<= tol
        it=k; return
    end
    T=D*P+P*E; a=rho/sum(sum(P.*T)); V=V+a*P; R=R-a*T;
    rhos=rho; rho=sum(sum(R.*R)); P=R+(rho/rhos)*P;
end;
it=itmax+1;
```

The Averaging Problem

n	2 500	10 000	40 000	1 000 000	4 000 000
K	22	22	21	21	20

Table 1: The number of iterations K for the averaging problem on a $\sqrt{n} \times \sqrt{n}$ grid. $\boldsymbol{x}_0 = \mathbf{0}$ $tol = 10^{-8}$

- Both the condition number and the required number of iterations are independent of the size of the problem
- The convergence is quite rapid.

Poisson Problem

- $\lambda_{jk} = 2c + 2a\cos(j\pi h) + 2b\cos(k\pi h),\ j,k = 1,2,\ldots,m.$
- $a = b = -1$, $c = 2$
- $\lambda_{max} = 4 + 4\cos(\pi h)$, $\lambda_{min} = 4 - 4\cos(\pi h)$
- $\mathrm{cond}_2(\boldsymbol{A}) = \frac{\lambda_{max}}{\lambda_{min}} = \frac{1+\cos(\pi h)}{1-\cos(\pi h)} = \mathrm{cond}_(\boldsymbol{T})_2.$
- $\mathrm{cond}_2(\boldsymbol{A}) = O(n)$.

The Poisson problem

n	2 500	10 000	40 000	160 000
K	140	294	587	1168
$K/\sqrt{n}$	1.86	1.87	1.86	1.85

- Using CG in the form of Algorithm 8 with $\epsilon = 10^{-8}$ and $\boldsymbol{x}_0 = 0$ we list K, the required number of iterations and $K/\sqrt{n}$.
- The results show that K is much smaller than n and appears to be proportional to $\sqrt{n}$
- This is the same speed as for SOR and we don't have to estimate any acceleration parameter!
- $\sqrt{n}$ is essentially the square root of the condition number of A.

Complexity

The work involved in each iteration is

1. one matrix times vector ($\boldsymbol{t} = \boldsymbol{A}\boldsymbol{p}$),
2. two inner products ($\boldsymbol{p}^T\boldsymbol{t}$ and $\boldsymbol{r}^T\boldsymbol{r}$),
3. three vector-plus-scalar-times-vector ($\boldsymbol{x} = \boldsymbol{x} + a\boldsymbol{p}$, $\boldsymbol{r} = \boldsymbol{r} - a\boldsymbol{t}$ and $\boldsymbol{p} = \boldsymbol{r} + (rho/rhos)\boldsymbol{p}$),

The dominating part of the computation is statement 1. Note that for our test problems $\boldsymbol{A}$ only has $O(5n)$ nonzero elements. Therefore, taking advantage of the sparseness of $\boldsymbol{A}$ we can compute $\boldsymbol{t}$ in $O(n)$ flops. With such an implementation the total number of flops in one iteration is $O(n)$.

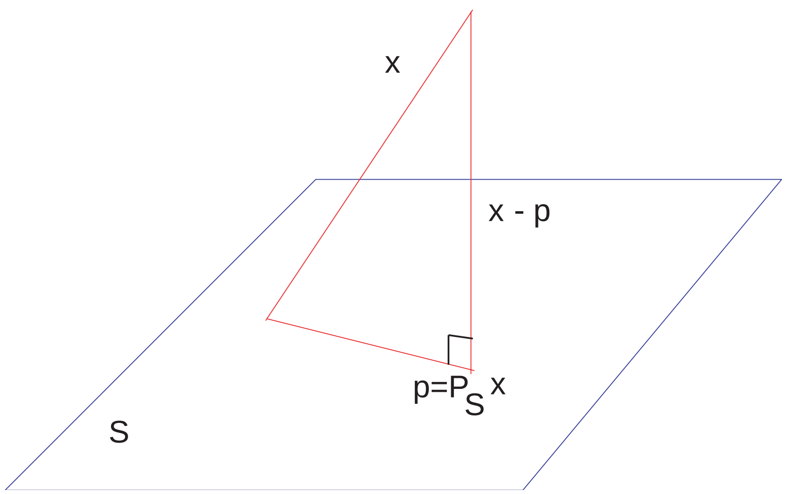
More Complexity

- How many flops do we need to solve the test problems by the conjugate gradient method to within a given tolerance?
- Average problem. $O(n)$ flops. Optimal for a problem with n unknowns.
- Same as SOR and better than the fast method based on FFT.
- Discrete Poisson problem: $O(n^{3/2})$ flops.
- same as SOR and fast method.
- Cholesky Algorithm: $O(n^2)$ flops both for averaging and Poisson.

Analysis and Derivation of the Method

Theorem 3 (Orthogonal Projection)**.** *Let* $\mathcal{S}$ *be a subspace of a finite dimensional real or complex inner product space* $(\mathcal{V}, \mathbb{F}, \langle\cdot,\cdot,\rangle)$*. To each* $\boldsymbol{x} \in \mathcal{V}$ *there is a unique vector* $\boldsymbol{p} \in \mathcal{S}$ *such that*

$$\langle \boldsymbol{x} - \boldsymbol{p}, \boldsymbol{s} \rangle = 0, \quad \textit{for all } \boldsymbol{s} \in \mathcal{S}. \tag{1}$$



Best Approximation

Theorem 4 (Best Approximation)**.** *Let $\mathcal{S}$ be a subspace of a finite dimensional real or complex inner product space $(\mathcal{V}, \mathbb{F}, \langle\cdot,\cdot,\rangle)$. Let $x \in \mathcal{V}$, and $\boldsymbol{p} \in \mathcal{S}$. The following statements are equivalent*

1. $\langle \boldsymbol{x} - \boldsymbol{p}, \boldsymbol{s}\rangle = 0, \quad$ *for all $\boldsymbol{s} \in \boldsymbol{S}$.*
2. $\|\boldsymbol{x} - \boldsymbol{s}\| > \|\boldsymbol{x} - \boldsymbol{p}\|$ *for all $\boldsymbol{s} \in \mathcal{S}$ with $\boldsymbol{s} \neq \boldsymbol{p}$.*

If $(\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k)$ is an orthogonal basis for $\boldsymbol{S}$ then

$$\boldsymbol{p} = \sum_{i=1}^{k} \frac{\langle \boldsymbol{x}, \boldsymbol{v}_i\rangle}{\langle \boldsymbol{v}_i, \boldsymbol{v}_i\rangle} \boldsymbol{v}_i. \tag{2}$$

Derivation of CG

- $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, $\boldsymbol{A} \in \mathbb{R}^{n,n}$ is pos. def., $\boldsymbol{x}, \boldsymbol{b} \in \mathbb{R}^n$
- $(\boldsymbol{x}, \boldsymbol{y}) := \boldsymbol{x}^T \boldsymbol{y}, \quad \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$
- $\langle \boldsymbol{x}, \boldsymbol{y} \rangle := \boldsymbol{x}^T \boldsymbol{A}\boldsymbol{y} = (\boldsymbol{x}, \boldsymbol{A}\boldsymbol{y}) = (\boldsymbol{A}\boldsymbol{x}, \boldsymbol{y})$
- $\|\boldsymbol{x}\|_{\boldsymbol{A}} = \sqrt{\boldsymbol{x}^T \boldsymbol{A}\boldsymbol{x}}$
- $\mathbb{W}_0 = \{\boldsymbol{0}\}$, $\mathbb{W}_1 = \operatorname{span}\{\boldsymbol{b}\}$, $\mathbb{W}_2 = \operatorname{span}\{\boldsymbol{b}, \boldsymbol{A}\boldsymbol{b}\}$,
  $\mathbb{W}_k = \operatorname{span}\{\boldsymbol{b}, \boldsymbol{A}\boldsymbol{b}, \boldsymbol{A}^2\boldsymbol{b}, \ldots, \boldsymbol{A}^{k-1}\boldsymbol{b}\}$
- $\mathbb{W}_0 \subset \mathbb{W}_1 \subset \mathbb{W}_2 \subset \mathbb{W}_k \subset \cdots$
- $\dim(\mathbb{W}_k) \le k, \quad \boldsymbol{w} \in \mathbb{W}_k \Rightarrow \boldsymbol{A}\boldsymbol{w} \in \mathbb{W}_{k+1}$
- $\boldsymbol{x}_k \in \mathbb{W}_k$, $\langle \boldsymbol{x}_k - \boldsymbol{x}, \boldsymbol{w} \rangle = 0$ for all $\boldsymbol{w} \in \mathbb{W}_k$
- $\boldsymbol{p}_0 = \boldsymbol{r}_0 := \boldsymbol{b}$, $\boldsymbol{p}_j = \boldsymbol{r}_j - \sum_{i=0}^{j-1} \frac{\langle \boldsymbol{r}_j, \boldsymbol{p}_i \rangle}{\langle \boldsymbol{p}_i, \boldsymbol{p}_i \rangle} \boldsymbol{p}_i, \quad j = 1, \ldots, k.$

Convergence

Theorem 5. *Suppose we apply the conjugate gradient method to a positive definite system* $\boldsymbol{Ax} = \boldsymbol{b}$*. Then the* $\boldsymbol{A}$*-norms of the errors satisfy*

$$\frac{||\boldsymbol{x} - \boldsymbol{x}_k||_{\boldsymbol{A}}}{||\boldsymbol{x} - \boldsymbol{x}_0||_{\boldsymbol{A}}} \leq 2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^k, \quad \textit{for} \quad k \geq 0,$$

where $\kappa = cond_2(\boldsymbol{A}) = \lambda_{max}/\lambda_{min}$ *is the 2-norm condition number of* $\boldsymbol{A}$*.*

This theorem explains what we observed in the previous section. Namely that the number of iterations is linked to $\sqrt{\kappa}$, the square root of the condition number of $\boldsymbol{A}$. Indeed, the following corollary gives an upper bound for the number of iterations in terms of $\sqrt{\kappa}$.

Corollary 6. *If for some* $\epsilon > 0$ *we have* $k \geq \frac{1}{2}\ln(\frac{2}{\epsilon})\sqrt{\kappa}$ *then* $||\boldsymbol{x} - \boldsymbol{x}_k||_{\boldsymbol{A}}/||\boldsymbol{x} - \boldsymbol{x}_0||_{\boldsymbol{A}} \leq \epsilon$.