Convergence bound for Conjugate Gradients:

$$\frac{\|x - x_k\|_A}{\|x - x_0\|_A} \le \min_{p \in \Pi_k, p(0)=1} \max_j |p(\lambda_j)|$$

If $\lambda_j \in [a, b], a > 0$ for all $j$ then

$$\frac{\|x - x_k\|_A}{\|x - x_0\|_A} \le \min_{p \in \Pi_k, p(0)=1} \max_{t \in [a,b]} |p(t)|$$

and the minimum here is achieved by the shifted and scaled Chebyshev polynomial

$$p(t) = T_k\left(\frac{2t - b - a}{b - a}\right) \Big/ T_k\left(\frac{-b - a}{b - a}\right)$$

similarly to before (but note different normalisation).

Now for

$$t \in [a,b] \quad , \frac{2t-b-a}{b-a} \in [-1,1]$$

so

$$\max_{t \in [a,b]} |p(t)| = p(b) = \frac{\frac{1}{2^{k-1}} 1}{\frac{1}{2^{k-1}} \cosh k\theta} \quad , \cosh\theta = \frac{b+a}{b-a}.$$

Note: $e^{\theta} + e^{-\theta} = 2\left(\frac{b+a}{b-a}\right) \Leftrightarrow \left(e^{\theta}\right)^2 - 2\left(\frac{a+b}{b-a}\right)\left(e^{\theta}\right) + 1 = 0$

$$\Leftrightarrow e^{\theta} = -\left(\frac{\sqrt{b}-\sqrt{a}}{\sqrt{b}+\sqrt{a}}\right) \text{ or } \left(\frac{-\sqrt{b}-\sqrt{a}}{\sqrt{b}-\sqrt{a}}\right),$$

So

$$\min_{p\in\Pi_k, p_0=1} \quad \max_{t\in[a,b]} |p(t)| =$$

$$2 \quad \left[\left(\frac{\sqrt{b}-\sqrt{a}}{\sqrt{b}+\sqrt{a}}\right)^k + \left(\frac{\sqrt{b}+\sqrt{a}}{\sqrt{b}-\sqrt{a}}\right)^k\right]^{-1}$$

$$\leq \quad 2\left(\frac{\sqrt{b}-\sqrt{a}}{\sqrt{b}+\sqrt{a}}\right)^k = 2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^k$$

when $b = \lambda_{\text{max}}(A), a = \lambda_{\text{min}}(A)$ and $\kappa = \frac{\lambda_{\max}(A)}{\lambda_{\min}(A)}$
$\kappa$ the $\|\cdot\|_2$ condition number again.

$$CG: \quad \frac{\|x - x_k\|_A}{\|x - x_0\|_A} \leq \min_{p \in \Pi_k, p(0)=1} \max_j |p(\lambda_j)|$$

$$\leq 2 \left( \frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^k, \quad \kappa = \frac{\lambda_{\max}(A)}{\lambda_{\min}(A)}.$$

For fast convergence:

$$CG: \quad \frac{\|x - x_k\|_A}{\|x - x_0\|_A} \leq \min_{p \in \Pi_k, p(0)=1} \max_j |p(\lambda_j)|$$

$$\leq 2 \left( \frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^k, \quad \kappa = \frac{\lambda_{\max}(A)}{\lambda_{\min}(A)}.$$

For fast convergence:

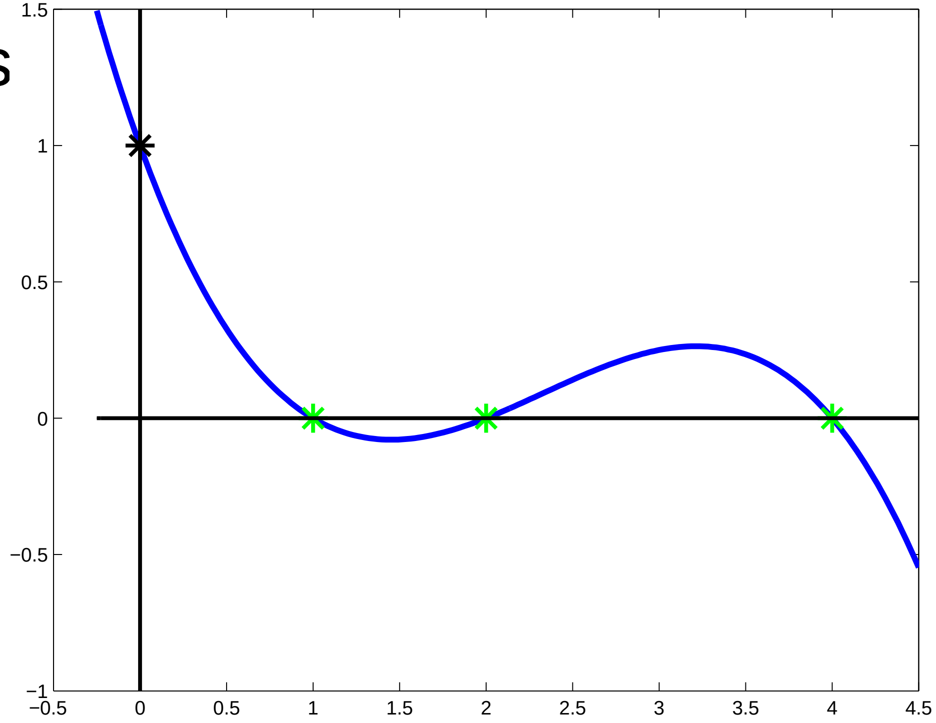
- few distinct eigenvalues
- cluster eigenvalues
- reduce $\kappa$

$$CG: \quad \frac{\|x - x_k\|_A}{\|x - x_0\|_A} \leq \min_{p \in \Pi_k, p(0)=1} \max_j |p(\lambda_j)|$$

$$\leq 2 \left( \frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^k, \quad \kappa = \frac{\lambda_{\max}(A)}{\lambda_{\min}(A)}.$$

For fast convergence:

- few distinct eigenvalues
- cluster eigenvalues
- reduce $\kappa$

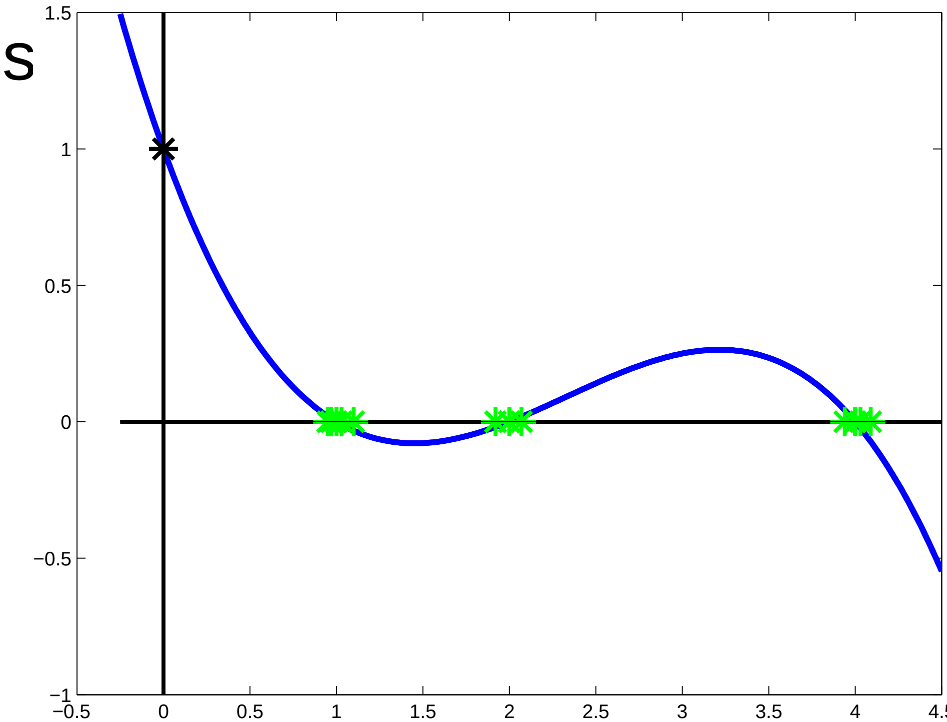


$$\boldsymbol{CG:}\quad \frac{\|x-x_k\|_A}{\|x-x_0\|_A} \le \min_{p\in\Pi_k,\,p(0)=1} \max_j |p(\lambda_j)|$$

$$\le 2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^k, \quad \kappa = \frac{\lambda_{\max}(A)}{\lambda_{\min}(A)}.$$

For fast convergence:

- few distinct eigenvalues
- cluster eigenvalues
- reduce $\kappa$

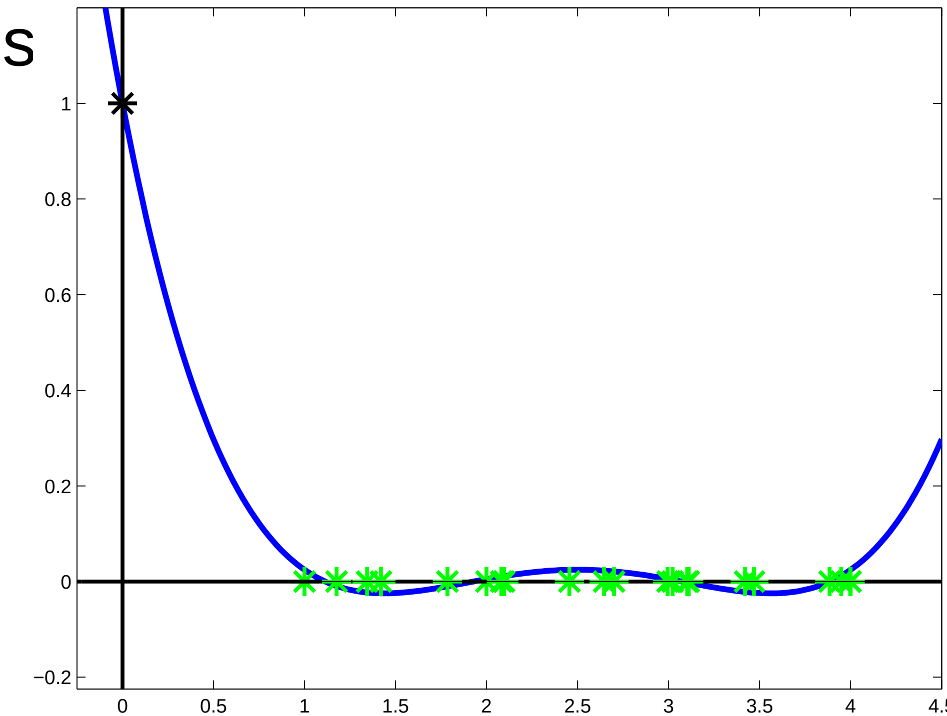


$$CG: \quad \frac{\|x - x_k\|_A}{\|x - x_0\|_A} \leq \min_{p \in \Pi_k, p(0)=1} \max_j |p(\lambda_j)|$$

$$\leq 2 \left( \frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^k, \quad \kappa = \frac{\lambda_{\max}(A)}{\lambda_{\min}(A)}.$$

For fast convergence:

- few distinct eigenvalues
- cluster eigenvalues
- reduce $\kappa$



$$\frac{\|x - x_k\|_A}{\|x - x_0\|_A} \leq \min_{p \in \Pi_k, p(0)=1} \max_j |p(\lambda_j)|$$
$$\leq \min_{p \in \Pi_k, p(0)=1} \max_{t \in [a,b]} |p(t)|$$
$$\leq 2 \left( \frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^k$$

Hence Conjugate Gradients guaranteed to converge fast if

**(i)** $A$ has few distant eigenvalues (or few clusters)

**(ii)** $\kappa$ is small, e.g. if $\kappa = 9$

$$\frac{\|x - x_k\|_A}{\|x - x_0\|_A} \leq 2 \left( \frac{3 - 1}{3 + 1} \right)^k = \frac{2}{2^k}$$

error halving at each iteration.

BUT $\boldsymbol{A}, \boldsymbol{b}$ given not chosen! So how can fast convergence be achieved for $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ when none of (i), (ii) are true?

**Preconditioning**: choose symmetric positive definite matrix $\boldsymbol{P}$ and for the mathematical analysis only (not for the algorithm - see exercises) let $\boldsymbol{P} = \boldsymbol{H}\boldsymbol{H}^T$ for example be a Cholesky factorisation. Solve

$$\underbrace{(\boldsymbol{H}^{-1}\boldsymbol{A}\boldsymbol{H}^{-T})}_{\text{symmetric}}(\boldsymbol{H}^T\boldsymbol{x}) = \boldsymbol{H}^{-1}\boldsymbol{b} \qquad (\star)$$

by Conjugate Gradients.

It computes iterates $\{x_k\}$ for $(\star)$ and hence for $Ax = b$ such that

$$\frac{\|x - x_k\|_A}{\|x - x_0\|_A} \leq 2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^k$$

where

$$\begin{aligned}\kappa &= \frac{\lambda_{\max}(H^{-1}AH^{-T})}{\lambda_{\min}(H^{-1}AH^{-T})} \\ &= \frac{\lambda_{\max}(P^{-1}A)}{\lambda_{\min}(P^{-1}A)}\end{aligned}$$

because of the similarity transformation

$$H^{-T}(H^{-1}AH^{-T})H^{T} = P^{-1}A.$$

This CG method with preconditioner $P$ is called the Preconditioned Conjugate Gradient method.

The preconditioned Conjugate Gradient method requires the solution for $z_k$ given $r_k$ of

$$Pz_k = r_k$$

at each iteration $\Rightarrow$ the (contradictory) requirements:

- $\kappa(P^{-1}A) << \kappa(A) \quad \Rightarrow$ fast convergence
- $Pz_k = r_k$ easy to solve

**(1)** $P = A \Rightarrow$ convergence in 1 iteration, but solution of linear system with $A$ at that iteration!

**(2)** $P = I \Rightarrow$ unpreconditioned CG

(2)leads to simple ideas like $P = diag(A)$ or $P = tridiag(A)$ or even $P =$part of $A$ with bandwidth $b$ $\Rightarrow O(nb^2)$ work for $Pz = r$ solve by banded elimination.
(1)leads to consideration of approximate or incomplete triangular factorizations of $A$

Incomplete Cholesky(0) factorization:

```
for i = 1, 2, ... n
    m = min{k : a_{i,k} ≠ 0}
    for j = m, ..., i − 1
        if a_{i,j} ≠ 0
            l_{i,j} ← (a_{i,j} − Σ_{k=m}^{j−1} l_{i,k} l_{j,k}) / l_{j,j}
        endif
    enddo
    l_{i,i} = (a_{i,i} − Σ_{k=m}^{i−1} l_{i,k} l_{i,k})^{1/2}
enddo
```

$$\texttt{for}\ i = 1, 2, \ldots n$$
$$m = \min\{k : a_{i,k} \neq 0\}$$
$$\texttt{for}\ j = m, \ldots, i-1$$
$$\texttt{if}\ a_{i,j} \neq 0$$
$$l_{i,j} \leftarrow \left(a_{i,j} - \sum_{k=m}^{j-1} l_{i,k} l_{j,k}\right) / l_{j,j}$$
$$\texttt{endif}$$
$$\texttt{enddo}$$
$$l_{i,i} = \left(a_{i,i} - \sum_{k=m}^{i-1} l_{i,k} l_{i,k}\right)^{\frac{1}{2}}$$
$$\texttt{enddo}$$

Incomplete Cholesky(0) factorization computes entries of a lower triangular matrix $\boldsymbol{L}$ such that $\boldsymbol{L}\boldsymbol{L}^T = \boldsymbol{A} + \boldsymbol{R}$ where

sparsity pattern of $\boldsymbol{L}$=sparsity pattern of lower triangle of $\boldsymbol{A}$

$\boldsymbol{R}$ is the remainder.

Effect of preconditioning in this case: replace $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ by

$$\boldsymbol{L}^{-1}\boldsymbol{A}\boldsymbol{L}^{-T}(\boldsymbol{L}\boldsymbol{x}) = \boldsymbol{L}^{-1}\boldsymbol{b}$$

and note $\boldsymbol{L}^{-1}\boldsymbol{A}\boldsymbol{L}^{-T} = \boldsymbol{I} - \boldsymbol{L}^{-1}\boldsymbol{R}\boldsymbol{L}^{-T}$ so if $\boldsymbol{R}$ small can expect eigenvalues of $\boldsymbol{L}^{-1}\boldsymbol{A}\boldsymbol{L}^{-T}$ to be clustered around 1.

Note: solve $\boldsymbol{P}\boldsymbol{z} = \boldsymbol{r}$ is solve $\boldsymbol{L}\boldsymbol{L}^T\boldsymbol{z} = \boldsymbol{r}$ is easily achieved by forwards and back substitution.

Preconditioning can also be applied with GMRES and MINRES: for GMRES no symmetry to preserve so just solve eg. $\boldsymbol{P^{-1}Ax = P^{-1}b}$.

In the algorithm just replace $\boldsymbol{A}*$vector multiplies by

$$\boldsymbol{z = Av}, \quad \text{solve } \boldsymbol{Ps = z} \qquad \Rightarrow \quad \boldsymbol{s = P^{-1}Az}$$