MA 580; Iterative Methods for Linear Equations

C. T. Kelley
NC State University
tim_kelley@ncsu.edu

Version of October 21, 2016

Read Chapters 2 and 3 of the Red book.

NCSU, Fall 2016

Part VId: Krylov Methods for Linear Equations: Conjugate Gradient

CG's Minimization Principle

Solve $\mathbf{Ax} = \mathbf{b}$ where $\mathbf{A}$ is spd.
For CG, $\mathbf{x}_k$ minimizes the $\mathbf{A}$-norm of the error

$$\|\mathbf{x}^* - \mathbf{x}\|_{\mathbf{A}} = \min_{\mathbf{x} \in \mathbf{x}_0 + \mathcal{K}_k} \|\mathbf{x}^* - \mathbf{x}\|_{\mathbf{A}}$$

over $\mathbf{x}_0 + \mathcal{K}_k$, where

$$\|\mathbf{v}\|_{\mathbf{A}}^2 = \mathbf{v}^T \mathbf{A} \mathbf{v}.$$

CG and Residual Polymonials

As with GMRES, any $\mathbf{x} \in \mathbf{x}_0 + \mathcal{K}_k$ can be written

$$\mathbf{x} = \mathbf{x}_0 + \sum_{j=0}^{k-1} \gamma_j \mathbf{A}^j \mathbf{r}_0$$

Let $\mathbf{x}^* = \mathbf{A}^{-1}\mathbf{b}$ and $\mathbf{e} = \mathbf{x}^* - \mathbf{x}$. Since $\mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{x}^* = \mathbf{A}\mathbf{e}$,

$$\begin{aligned} \mathbf{x}^* - \mathbf{x} &= \mathbf{e} = \mathbf{x}^* - \mathbf{x}_0 - \textstyle\sum_{j=0}^{k-1} \gamma_j \mathbf{A}^j \mathbf{r}_0 \\ &= \mathbf{e}_0 - \textstyle\sum_{j=1}^{k} \gamma_{j-1} \mathbf{A}^j \mathbf{e}_0 = p(\mathbf{A})\mathbf{e}_0 \end{aligned}$$

for some $p \in \mathcal{P}_k$.

Minimization Principle

So, if $\mathbf{x}_k$ is the kth CG iteration

$$\|\mathbf{e}_k\|_{\mathbf{A}} \le \|p(\mathbf{A})\mathbf{e}_0\|_{\mathbf{A}} \le \|p(\mathbf{A})\|_{\mathbf{A}} \|\mathbf{e}_0\|_{\mathbf{A}}$$

for all $p \in \mathcal{P}_k$.
So what does this mean?

What is the **A**-norm of $p(\mathbf{A})$

Since **A** is spd, **A** has a unique spd square root,

$$A = \mathbf{U} \Lambda \mathbf{U}^T \text{ and } \sqrt{\mathbf{A}} = \mathbf{U} \sqrt{\Lambda} \mathbf{U}^T$$

so

$$\|\mathbf{x}\|_{\mathbf{A}}^2 = \mathbf{x}^T \mathbf{A} \mathbf{x} = (\sqrt{\mathbf{A}}\mathbf{x})^T (\sqrt{\mathbf{A}}\mathbf{x}) = \|\sqrt{\mathbf{A}}\mathbf{x}\|_2^2$$

which means

$$\|p(\mathbf{A})\mathbf{x}\|_{\mathbf{A}}^2 = \|\sqrt{\mathbf{A}} p(\mathbf{A})\mathbf{x}\|_2^2 = \|p(\mathbf{A})(\sqrt{\mathbf{A}}\mathbf{x})\|_2^2$$

$\|p(\mathbf{A})\|_{\mathbf{A}} = \|p(\mathbf{A})\|_2$

Hence

$$\begin{aligned}
\|p(\mathbf{A})\|_{\mathbf{A}} &= \max_{\mathbf{x}\neq 0} \frac{\|p(\mathbf{A})\mathbf{x}\|_A}{\|\mathbf{x}\|_{\mathbf{A}}} = \max_{\mathbf{x}\neq 0} \frac{\|\sqrt{\mathbf{A}}p(\mathbf{A})\mathbf{x}\|_2}{\|\sqrt{\mathbf{A}}\mathbf{x}\|_2} \\
&= \max_{\sqrt{\mathbf{A}}\mathbf{x}\neq 0} \frac{\|p(\mathbf{A})\sqrt{\mathbf{A}}\mathbf{x}\|_2}{\|\sqrt{\mathbf{A}}\mathbf{x}\|_2} = \max_{\mathbf{z}\neq 0} \frac{\|p(\mathbf{A})\mathbf{z}\|_2}{\|\mathbf{z}\|_2} \\
&= \|p(\mathbf{A})\|_2 = \max_{\lambda\in\sigma(\mathbf{A})} |p(\lambda)|
\end{aligned}$$

Residual Polynomial Analysis

As with GMRES

$$\|\mathbf{e}_k\|_{\mathbf{A}} \le \max_{\lambda \in \sigma(\mathbf{A})} |p(\lambda)| \|\mathbf{e}_0\|_{\mathbf{A}}$$

So, for example, if $\sigma(\mathbf{A}) \subset (.9, 1.1)$ then

$$\|\mathbf{e}_k\|_{\mathbf{A}} \le \|\mathbf{e}_0\| 10^{-k}$$

which we get by using $p(z) = (1 - z)^k$.

Convergence within N Iterations

Theorem: Let $\mathbf{A}$ be spd. Then the CG algorithm will find the solution within N iterations.

Proof: Use

$$p(z) = \prod \left(\frac{\lambda_i - z}{\lambda_i} \right)$$

The Concus-Golub-O'Leary Estimate

Theorem: Let $0 < \lambda_1 \leq \lambda_2 \leq \lambda_N$ be the eigenvalues of $\mathbf{A}$ (so $\kappa(\mathbf{A}) = \lambda_N/\lambda_1$). Let $\mathbf{x}_k^*$ be the kth CG iteration. Then

$$\frac{\|\mathbf{e}_k\|_{\mathbf{A}}}{\|\mathbf{e}_0\|_{\mathbf{A}}} \leq \left[\frac{\sqrt{\kappa(\mathbf{A})} - 1}{\sqrt{\kappa(\mathbf{A})} + 1}\right]^k.$$

This can be pessimistic if the eigenvalues are clustered.

Termination

It's standard to terminate the iteration when the residual is small

$$\|\mathbf{r}_k\|_2 \le \|\mathbf{b} - \mathbf{A}\mathbf{x}_k\|_2 \le \eta \|\mathbf{b}\|_2.$$

How is $\|\mathbf{r}\|_2$ connected to the $\mathbf{A}$-norm of e?
Since

$$\sqrt{\lambda_1}\|\mathbf{x}^*\|_2 \le \|\mathbf{x}^*\|_{\mathbf{A}} \le \sqrt{\lambda_N}\|\mathbf{x}^*\|_2$$

we have

$$\frac{\|\mathbf{r}_k\|_2}{\|\mathbf{r}_0\|_2} = \frac{\|\mathbf{A}\mathbf{e}_k\|_2}{\|\mathbf{A}\mathbf{e}_0\|_2} \le \sqrt{\kappa(\mathbf{A})}\frac{\|\sqrt{\mathbf{A}}\mathbf{e}_k\|_2}{\|\sqrt{\mathbf{A}}\mathbf{e}_0\|_2} = \sqrt{\kappa(\mathbf{A})}\frac{\|\mathbf{e}_k\|_{\mathbf{A}}}{\|\mathbf{e}_0\|_{\mathbf{A}}}$$

Example

Let $\mathbf{x}_0 = 0$ and assume $\sigma(\mathbf{A}) \subset (9, 11)$. Using $p(z) = (10 - z)^k / 10^k$ we see

$$\|\mathbf{e}_k\|_{\mathbf{A}} / \|\mathbf{e}_0\|_{\mathbf{A}} \le 10^{-k}.$$

So the **A**-norm of the error will be reduced by a factor of 10^{-3} after 3 iterations.
What about the residual? All we know is that $\kappa(\mathbf{A}) \le 11/9$, so

$$\frac{\|\mathbf{r}_k\|_2}{\|\mathbf{r}_0\|_2} \le 10^{-k} \sqrt{11/9}$$

and we need 4 iterations to guarantee a residual reduction of 10^{-3}.

Alternative Minimization Principle

Theorem: The kth iterate $\mathbf{x}_k$ of CG minimizes

$$\phi(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T \mathbf{A}\mathbf{x} - \mathbf{x}^T b$$

over $\mathbf{x}_0 + \mathcal{K}_k$

Remark: Note that if $\tilde{\mathbf{x}}$ is any a stationary point,

$$\nabla\phi(\tilde{\mathbf{x}}) = \mathbf{A}\tilde{\mathbf{x}} - \mathbf{b} = 0$$

then $\tilde{\mathbf{x}} = \mathbf{x}^*$.

Proof

Note that, since **A** is symmetric

$$\|\mathbf{x}^* - \mathbf{x}\|_{\mathbf{A}}^2 = (\mathbf{x}^* - \mathbf{x})^T A(\mathbf{x}^* - \mathbf{x}) = \mathbf{x}^{*T}\mathbf{A}\mathbf{x}^* - 2\mathbf{x}^{*T}\mathbf{A}\mathbf{x} + \mathbf{x}^T\mathbf{A}\mathbf{x}.$$

Now $\mathbf{A}\mathbf{x}^* = b$ implies

$$-2\mathbf{x}^{*T}\mathbf{A}\mathbf{x} + \mathbf{x}^T\mathbf{A}\mathbf{x} = \mathbf{x}^T\mathbf{A}\mathbf{x} - 2\mathbf{x}^T\mathbf{b} = 2\phi(\mathbf{x}).$$

Therefore

$$\|\mathbf{e}\|_A^2 = \|\mathbf{x}^* - \mathbf{x}\|_{\mathbf{A}}^2 = 2\phi(\mathbf{x}) + (\mathbf{x}^*)^T\mathbf{A}\mathbf{x}^*.$$

So $\mathbf{x}$ minimizes ϕ over any set if and only if $\mathbf{x}$ minimizes $\|\mathbf{x}^* - \mathbf{x}\|_{\mathbf{A}}^2$.

CG Implementation

cg($\mathbf{x}, \mathbf{b}, \mathbf{A}, \epsilon, kmax$)

$\mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{x}$, $\rho_0 = \|\mathbf{r}\|_2^2$, $k = 1$.
while $\sqrt{\rho_{k-1}} > \epsilon \|\mathbf{b}\|$ and $k < kmax$ **do**
  if $k = 1$ **then**
    $\mathbf{p} = \mathbf{r}$
  else
    $\beta = \rho_{k-1}/\rho_{k-2}$ and $\mathbf{p} = \mathbf{r} + \beta \mathbf{p}$
  end if
  $\mathbf{w} = \mathbf{A}\mathbf{p}$
  $\alpha = \rho_{k-1}/\mathbf{p}^T\mathbf{w}$
  $\mathbf{x} = \mathbf{x} + \alpha \mathbf{p}$
  $\mathbf{r} = \mathbf{r} - \alpha \mathbf{w}$
  $\rho_k = \|\mathbf{r}\|^2$
  $k = k + 1$
end while

CG Implementation: Cost I, two scalar products

cg($\mathbf{x}^*, \mathbf{b}, \mathbf{A}, \epsilon, kmax$)

$\mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{x}^*$, $\rho_0 = \|r\|_2^2$, $k = 1$.

while $\sqrt{\rho_{k-1}} > \epsilon\|b\|$ and $k < kma\mathbf{x}^*$ **do**

if $k = 1$ **then**

$\mathbf{p} = \mathbf{r}$

else

$\beta = \rho_{k-1}/\rho_{k-2}$ and $p = r + \beta p$

end if

$\mathbf{w} = \mathbf{A}\mathbf{p}$

$\alpha = \rho_{k-1}/\mathbf{p}^T\mathbf{w}$

$\mathbf{x} = \mathbf{x} + \alpha\mathbf{p}$

$\mathbf{r} = \mathbf{r} - \alpha\mathbf{w}$

$\rho_k = \|r\|^2$

$k = k + 1$

end while

CG Implementation: Cost II, three daxpys

cg($\mathbf{x}, \mathbf{b}, \mathbf{A}, \epsilon, kmax$)

$\mathbf{r} = \mathbf{b} - \mathbf{Ax}$, $\rho_0 = \|\mathbf{r}\|_2^2$, $k = 1$.

while $\sqrt{\rho_{k-1}} > \epsilon\|\mathbf{b}\|$ and $k < kmax$ **do**

  if $k = 1$ **then**

    $p = r$

  else

    $\beta = \rho_{k-1}/\rho_{k-2}$ and $p = r + \beta p$

  end if

  $w = \mathbf{Ap}$

  $\alpha = \rho_{k-1}/\mathbf{p}^T\mathbf{w}$

  $\mathbf{x} = \mathbf{x} + \alpha\mathbf{p}$

  $\mathbf{r} = \mathbf{r} - \alpha\mathbf{w}$

  $\rho_k = \|\mathbf{r}\|^2$

  $k = k + 1$

end while

Cost of CG

Each iteration requires

- one matrix-vector product,
- two scalar products,
- three daxpys,

and the storage of $\mathbf{x}, \mathbf{b}, \mathbf{r}, \mathbf{p}, \mathbf{w}$ five vectors!
Compare to GMRES (k vectors and $O(k)$ scalar products).

Preconditioned CG (PCG)

Right (or left) preconditioning is a problem because

$$\mathbf{BA} \text{ or } \mathbf{AB}$$

need not be spd.
The correct way to precondition CG is symmetrically

$$\mathbf{SASy} = \mathbf{S}b$$

and then $\mathbf{x} = \mathbf{Sy}$. This means that $\mathbf{S}^2 = \mathbf{B}$ is the preconditioner.
So do you have to compute $\mathbf{S} = \sqrt{\mathbf{B}}$?

PCG

pcg($\mathbf{x}, \mathbf{b}, \mathbf{A}, \mathbf{B}, \epsilon, kmax$)

$\mathbf{r} = \mathbf{b} - \mathbf{Ax}$, $\rho_0 = \|\mathbf{r}\|^2$, $k = 1$

while $\sqrt{\rho_{k-1}} > \epsilon \|\mathbf{b}\|$ and $k < kmax$ **do**

$\mathbf{z} = \mathbf{Br}$

$\tau_{k-1} = \mathbf{z}^T \mathbf{r}$

if if $k = 1$ **then**

$\beta = 0$ and $\mathbf{p} = \mathbf{z}$

else

$\beta = \tau_{k-1}/\tau_{k-2}$, $\mathbf{p} = \mathbf{z} + \beta \mathbf{p}$

end if

$w = \mathbf{Ap}$

$\alpha = \tau_{k-1}/\mathbf{p}^T \mathbf{w}$

$\mathbf{x} = \mathbf{x} + \alpha \mathbf{p}$; $\mathbf{r} = \mathbf{r} - \alpha \mathbf{w}$; $\rho_k = \mathbf{r}^T \mathbf{r}$

$k = k + 1$

end while

Cost of PCG

Each iteration requires

- one matrivx-vector product,
- one preconditioner-vector product,
- three scalar products,
- three daxpys,

and the storage of $\mathbf{x}, \mathbf{b}, \mathbf{r}, \mathbf{z}, \mathbf{p}, \mathbf{w}$ six vectors.

CG Examples

We'll use a couple tricks.

Theorem: Trick 1 Suppose $\mathbf{A}$ is spd with eigenvalues

$$0 < \lambda_1^{\mathbf{A}} \le \lambda_2^{\mathbf{A}} \le \cdots \le \lambda_N^{\mathbf{A}},$$

$\mathbf{B}$ is symmetric and $\rho(\mathbf{B}) < \lambda_1^{\mathbf{A}}$. Then $\mathbf{A} + \mathbf{B}$ is spd.

Proof

The sum of symmetric matrices is symmetric. Since

$$\mathbf{x}^T(\mathbf{A}+\mathbf{B})\mathbf{x} = \mathbf{x}^T\mathbf{A}\mathbf{x} + \mathbf{x}^T\mathbf{B}\mathbf{x} \geq \lambda_1^{\mathbf{A}}\|\mathbf{x}\|_2^2 - \rho(\mathbf{B})\|\mathbf{x}\|_2^2 > 0,$$

because $\rho(\mathbf{B}) < \lambda_1^{\mathbf{A}}$, $\mathbf{A}+\mathbf{B}$ is spd.

Application to integro-differential equation

Discretize

$$-u''(x) - \int_0^1 \frac{\cos(x+y)}{1+x+y} u(y)\, dy = e^x, \; u(0) = u(1) = 0,$$

as we have done several times to get

$$\mathbf{D}_2\mathbf{u} + \mathbf{K}\mathbf{u} = \mathbf{b}$$

where $b_i = e^{x_i}$, $\mathbf{D}_2$ is the finite-difference negative Laplacian, and

. . .

K and its norm

$$k_{ij} = -\frac{h\cos(h(i+j))}{1+h(i+j)}$$

$\|\mathbf{K}\|_\infty$ is the maximum absolute row sum so

$$\|\mathbf{K}\|_\infty = \max_i \sum_{j=1}^{N} \left| \frac{h\cos(h(i+j))}{1+h(i+j)} \right| \le \sum_{j=1}^{N} h = \frac{N}{N+1} \le 1.$$

Since $\lambda_1^{\mathbf{D}_2} \approx \pi^2$ and

$$\rho(\mathbf{K}) \le \|\mathbf{K}\|_\infty \le 1 < \lambda_1^{\mathbf{D}_2},$$

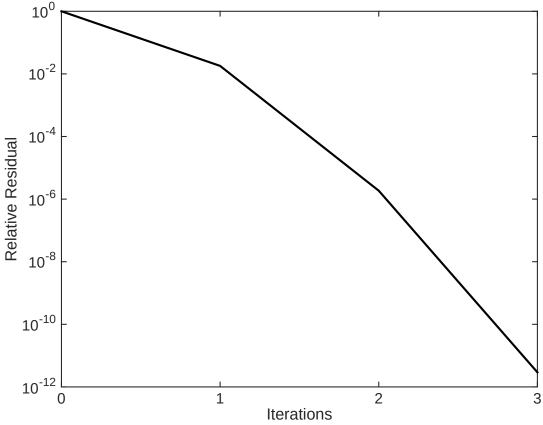
the result is applicable and $D_2 + \mathbf{K}$ is spd.

Setting up kl.m; cg_kde.m

No real difference from GMRES.

```
A=D2+K;
[L,U]=lu(D2);
p_data=struct('L',L,'U',U);
options=kl_optset('ltol',1.d-8,'matvec_data
    ',A,...
      'p_data',p_data,'ptv',@d2inv,'lmethod
         ','cg');
[u,hist]=kl(u0,b,@ide,options);
```

Residual History, $N = 99$, PCG



Trick 2

Suppose

$$\mathbf{B} = \mathbf{I} + \mathbf{MC}$$

where

- $\mathbf{M}$ is symmetric,
- $\mathbf{C}$ is diagonal and $0 < c_{ii} < 1$, and
- $\rho(\mathbf{M}) < 1$.

Then

$$\mathbf{A} = \mathbf{C}^{1/2}\mathbf{B}\mathbf{C}^{-1/2} = \mathbf{I} + \mathbf{C}^{1/2}\mathbf{M}\mathbf{C}^{1/2}$$

is spd.

Proof

We use trick #1 and

$$\rho(\mathbf{C}^{1/2}\mathbf{M}\mathbf{C}^{1/2}) \le \|\mathbf{M}\| \|\mathbf{C}^{1/2}\|^2 \le \|\mathbf{M}\|$$

for any norm.
In practice, you'd replace

$$\mathbf{u} - \mathbf{M}\mathbf{C}\mathbf{u} = \mathbf{f} \text{ with } \mathbf{v} - \mathbf{C}^{1/2}\mathbf{M}\mathbf{C}^{1/2}\mathbf{v} = \mathbf{b}$$

where $\mathbf{v} = \mathbf{C}^{1/2}\mathbf{u}$ and $\mathbf{b} = \mathbf{C}^{1/2}\mathbf{f}$.
After the solve: $\mathbf{u} \leftarrow \mathbf{C}^{-1/2}\mathbf{v}$.

Transport: Continuous form

The integral equation for the scalar flux is

$$\phi(x) - \int_0^\tau E_1(|x-y|)c(y)\phi(y)\,dy = g(x)$$

The integral operator is the product of a symmetric operator

$$Mu(x) = \int_0^\tau E_1(|x-y|)u(y)\,dy$$

and a multiplication operator $u(y) \rightarrow c(y)u(y)$.
So we can symmetrize by setting $\sqrt{c}\phi = v$ to get ...

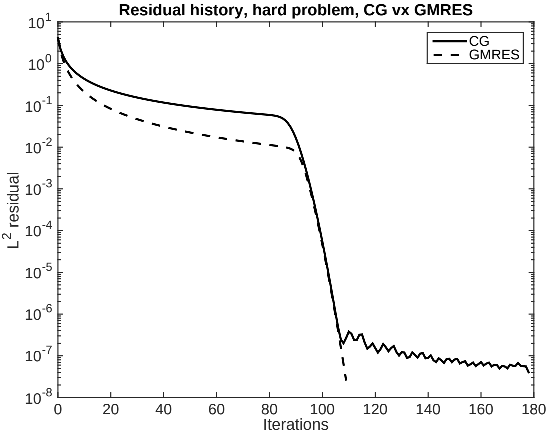
Symmetric integral equation

$$v(x) - \sqrt{c(x)} \int_0^\tau E_1(|x-y|)\sqrt{c(y)} v(y) \, dy = \sqrt{c(x)} g(x).$$

In the discrete case, we have to work on the matvec directly.
I'll show you the results, but not the code.
Yes, I did make sure that I got the same results both ways.

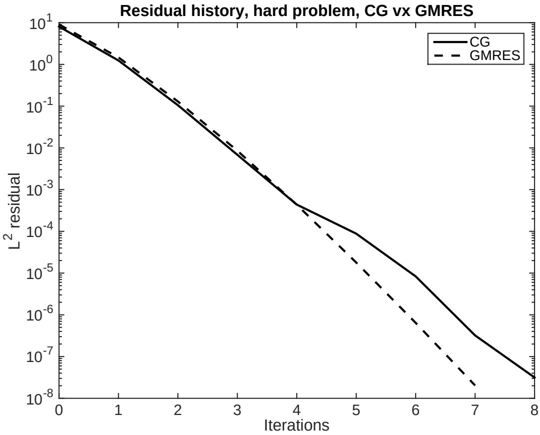
Residual History CG, Hard Problem, $na = 100, nx = 8001$

Already spd. CG=23 secs; GMRES=14 secs



Residual History CG, Easy Problem, $na = 40, nx = 2001$

CG=.3 secs; GMRES=.2 secs



A few subtleties

- You only have the $\mathbf{u} \leftarrow (\mathbf{MC})\mathbf{u}$ product
- This means you get $\mathbf{C}^{1/2}\mathbf{MC}^{1/2}\mathbf{v}$ by

$$\mathbf{C}^{1/2}(\mathbf{MC})(\mathbf{C}^{-1/2}\mathbf{v})$$

- $\rho(\mathbf{MC})$ is very near 1 for the hard problem.
- CG and GMRES minimize different things. Do not expect the iterations to be the same.
- Why is my CG for the hard problem looking bad at the end?

CGNR and CGNE

Conjugate gradient on the normal equations.

Two low-storage + provably convergent methods for nonsymmetric problems.

CGNR: Apply CG to

$$\mathbf{A}^T\mathbf{A}\mathbf{x} = \mathbf{A}^T\mathbf{b}$$

CGNE: Apply CG to

$$\mathbf{A}\mathbf{A}^T\mathbf{y} = \mathbf{b} \text{ and set } \mathbf{x} = \mathbf{A}^T\mathbf{y}.$$

This is not preconditioning!

Analysis of CGNR

Apply the minimization property. You miminize

$$\begin{aligned}\|\mathbf{x}^* - \mathbf{x}\|^2_{\mathbf{A}^T\mathbf{A}} &= (\mathbf{x}^* - \mathbf{x})^T\mathbf{A}^T\mathbf{A}(\mathbf{x}^* - \mathbf{x}) = (\mathbf{A}\mathbf{x}^* - \mathbf{A}\mathbf{x})^T(\mathbf{A}\mathbf{x}^* - \mathbf{A}\mathbf{x}) \\ &= (\mathbf{b} - \mathbf{A}\mathbf{x})^T(\mathbf{b} - \mathbf{A}\mathbf{x})^T = \|\mathbf{r}\|^2\end{aligned}$$

over $\mathbf{x}_0 + \mathcal{K}_k(\mathbf{A}^T\mathbf{A})$. Hence the name **C**onjugate **G**radient on the **N**ormal equations to minimize the **R**esidual.

Analysis of CGNE

Same story,

$$\begin{aligned} \|\mathbf{y}^* - \mathbf{y}\|^2_{\mathbf{A}\mathbf{A}^T} &= (\mathbf{y}^* - \mathbf{y})^T(\mathbf{A}\mathbf{A}^T)(\mathbf{y}^* - \mathbf{y}) \\ &= (\mathbf{A}^T\mathbf{y}^* - \mathbf{A}^T\mathbf{y})^T(\mathbf{A}^T\mathbf{y}^* - \mathbf{A}^T\mathbf{y}) = \|\mathbf{x}^* - \mathbf{x}\|^2 \end{aligned}$$

is minimized over $\mathbf{y}_0 + \mathcal{K}_k(\mathbf{A}\mathbf{A}^T)$ at each iterate. **C**onjugate **G**radient on the **N**ormal equations to minimize the **E**rror.

Using Residual Polynomials

For CGNR we look at the $\mathbf{A}^T\mathbf{A}$ norm and get

$$\|\mathbf{x}^* - \mathbf{x}\|^2_{\mathbf{A}^T\mathbf{A}} = (\mathbf{x}^* - \mathbf{x})^T \mathbf{A}^T \mathbf{A} (\mathbf{x}^* - \mathbf{x}) = \|\mathbf{A}(\mathbf{x}^* - \mathbf{x})\|_2^2 = \|\mathbf{r}\|_2^2.$$

Hence, for any residual polynomial $\bar{p}_k \in \mathcal{P}_k$,

$$\|\mathbf{r}_k\|_2 \le \|\bar{p}_k(\mathbf{A}^T\mathbf{A})\mathbf{r}_0\|_2 \le \|\mathbf{r}_0\|_2 \max_{z \in \sigma(\mathbf{A}^T\mathbf{A})} |\bar{p}_k(z)|. \qquad (1)$$

So, we want to cluster the squared singular values of $\mathbf{A}$.
CGNE is very, very, similar to this.

Example

Suppose the singular values of $\mathbf{A}$ lie in $(.9, 1.1)$. What is the smallest k so that

$$\|\mathbf{r}_k\|_2 \le 10^{-4}\|\mathbf{b}\|_2$$

where $\mathbf{r}$ is the CGNR residual.
The squared singular values like in $(.81, 1.21)$. So if I use $p_k(z) = (1 - z)^k$ I need

$$.21^k < 10^{-4}$$

and $k = 6$ will do the job.
Note: that'll be 12 matvecs.

Observations

- CGNR and CGNE need two matrix-vector products
- one is a transpose-vector product
  hard to do in a matrix-free way
- Condition number is squared, so more iterations are needed
- Classic time-for-storage trade-off.

GMRES(m) and other transpose-free methods

Give GMRES m iterations and then restart with

$$\mathbf{x}_0^{new} = \mathbf{x}_m^{old}$$

- Good news: you control storage, works great if $\|\mathbf{I} - \mathbf{A}\| < 1$
- Bad news: no theory, can and does completely fail

Other things: BiCGStab, TFQMR