Lecture 8: Fast Linear Solvers
(Part 5)
Conjugate Gradient (CG) Method

• Solve $Ax = b$ with $A$ being an $n \times n$ symmetric positive definite matrix.
  – proposed by Hestenes and Stiefel in 1951

• Define the quadratic function
  \[ \phi(x) = \frac{1}{2} x^T Ax - x^T b \]
  Suppose $x$ minimizes $\phi(x)$, $x$ is the solution to $Ax = b$.

• $\nabla \phi(x) = \left( \frac{\partial \phi}{\partial x_1}, ..., \frac{\partial \phi}{\partial x_n} \right) = Ax - b$

• The iteration takes form $x^{(k+1)} = x^{(k)} + \alpha_k v^{(k)}$
  where $v^{(k)}$ is the search direction and $\alpha_k$ is the step size.

• Define $r^{(k)} = b - Ax^{(k)}$ to be the residual vector.
Let \( x \) and \( v \neq 0 \) \( \phi(x + \alpha v) \) be fixed vectors and \( \alpha \) a real number variable.

Define:

\[
h(\alpha) = \phi(x + \alpha v) = \phi(x) + \alpha < v, Ax - b > + \frac{1}{2} \alpha^2 < v, Av >
\]

\( h(\alpha) \) has a minimum when \( h'(\alpha) = 0 \). This occurs when

\[
\hat{\alpha} = \frac{v^T(b - Ax)}{v^TAv}.
\]

So \( h(\hat{\alpha}) = \phi(x) - \frac{1}{2} \frac{(v^T(b-Ax))^2}{v^TAv} \).

Suppose \( x^* \) is a vector that minimizes \( \phi(x) \). So \( \phi(x + \hat{\alpha} v) \geq \phi(x^*) \).

This implies \( v^T(b - Ax^*) = 0 \). Therefore \( b - Ax^* = 0 \).
• For any \( \mathbf{v} \neq \mathbf{0} \), \( \phi(\mathbf{x} + \alpha \mathbf{v}) > \phi(\mathbf{x}) \) unless \( \mathbf{v}^T(\mathbf{b} - A\mathbf{x}) = 0 \) with \( \alpha = \frac{\mathbf{v}^T(\mathbf{b} - A\mathbf{x})}{\mathbf{v}^T A\mathbf{v}} \).

• How to choose the search direction \( \mathbf{v} \)?
  – **Method of steepest descent**: \( \mathbf{v} = \mathbf{r} = -\nabla \phi(\mathbf{x}) \)
    • Remark: Slow convergence for linear systems

*Algorithm.*

Let \( \mathbf{x}^{(0)} \) be initial guess.

**for** \( k = 1, 2, \ldots \)
  \( \mathbf{v}^{(k)} = \mathbf{b} - A\mathbf{x}^{(k-1)} \)
  \( \alpha_k = \frac{\langle \mathbf{v}^{(k)}, (\mathbf{b} - A\mathbf{x}^{(k-1)}) \rangle}{\langle \mathbf{v}^{(k)}, A\mathbf{v}^{(k)} \rangle} \)
  \( \mathbf{x}^{(k)} = \mathbf{x}^{(k-1)} + \alpha_k \mathbf{v}^{(k)} \)

**end**
Steepest descent method when $\frac{\lambda_{\text{max}}}{\lambda_{\text{min}}}$ is large

- Consider to solve $Ax = b$ with $A = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$, $b = \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix}$ and the start vector $v = \begin{bmatrix} -9 \\ -1 \end{bmatrix}$.

Reduction of $||Ax^{(k)} - b||_2 < 10^{-4}$.

- With $\lambda_1 = 1$, $\lambda_2 = 2$, it takes about 10 iterations
- With $\lambda_1 = 1$, $\lambda_2 = 10$, it takes about 40 iterations
• Second approach to choose the search direction \( \mathbf{v} \)?
  – *A-orthogonal approach*: use a set of nonzero direction vectors \( \{ \mathbf{v}^{(1)}, \ldots, \mathbf{v}^{(n)} \} \) that satisfy \( < \mathbf{v}^{(i)}, A\mathbf{v}^{(j)} > = 0 \), if \( i \neq j \). The set \( \{ \mathbf{v}^{(1)}, \ldots, \mathbf{v}^{(n)} \} \) is called A-orthogonal.

• **Theorem.** Let \( \{ \mathbf{v}^{(1)}, \ldots, \mathbf{v}^{(n)} \} \) be an A-orthogonal set of nonzero vectors associated with the symmetric, positive definite matrix \( A \), and let \( \mathbf{x}^{(0)} \) be arbitrary. Define \( \alpha_k = \frac{< \mathbf{v}^{(k)}, (\mathbf{b} - A\mathbf{x}^{(k-1)}) >}{< \mathbf{v}^{(k)}, A\mathbf{v}^{(k)} >} \) and \( \mathbf{x}^{(k)} = \mathbf{x}^{(k-1)} + \alpha_k \mathbf{v}^{(k)} \) for \( k = 1, 2 \ldots n \). Then \( A\mathbf{x}^{(n)} = \mathbf{b} \) when arithmetic is exact.
Conjugate Gradient Method

• The conjugate gradient method of Hestenes and Stiefel.

• Main idea: Construct \( \{ \mathbf{v}^{(1)}, \mathbf{v}^{(2)} \ldots \} \) during iteration so that \( \{ \mathbf{v}^{(1)}, \mathbf{v}^{(2)} \ldots \} \) are A-orthogonal.

• Define:
\[
K_k(A, \mathbf{r}^{(0)}) = \text{span}\{\mathbf{r}^{(0)}, A\mathbf{r}^{(0)}, A^2\mathbf{r}^{(0)}, \ldots, A^{k-1}\mathbf{r}^{(0)}\}.
\]

• Lemma (Kelly). Let A be spd and let \( \{ \mathbf{x}^{(k)} \} \) be CG iterates, then \( \mathbf{r}_k^T \mathbf{r}_l = 0 \) for all \( 0 \leq l < k \).
  – Remark: let \( \{ \mathbf{x}^{(k)} \} \) be CG iterates. \( \mathbf{r}_l \in K_k \) for all \( l < k \).

• Lemma (Kelly). Let A be spd and let \( \{ \mathbf{x}^{(k)} \} \) be CG iterates. If \( \mathbf{x}^{(k)} \neq \mathbf{x}^* \), then \( \mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_{k+1} \mathbf{v}^{(k+1)} \) and \( \mathbf{v}^{(k+1)} \) is determined up to a scalar multiple by the conditions \( \mathbf{v}^{(k+1)} \in K_{k+1}, (\mathbf{v}^{(k+1)})^T A \xi = 0 \) for all \( \xi \in K_k \).
  – Remark: This implies \( \mathbf{v}^{(k+1)} = \mathbf{r}^{(k)} + \mathbf{w}^{(k)} \) with \( \mathbf{w}^{(k)} \in K_k \).
• **Theorem** (Kelly). Let $A$ be spd and assume that $r^{(k)} \neq 0$. Define $v^{(0)} = 0$. Then $v^{(k+1)} = r^{(k)} + \beta_{k+1} v^{(k)}$ for some $\beta_{k+1}$ and $k \geq 0$.

  - Remark (1): $v^{(k+1)} \cdot A r^{(k-1)} = 0 = r^{(k)} \cdot A r^{(k-1)} + \beta_{k+1} v^{(k)} \cdot A r^{(k-1)}$

  - Remark (2): $x^{(k+1)} = x^{(k)} + \alpha_{k+1} v^{(k+1)}$ implies $r^{(k+1)} = r^{(k)} - \alpha_{k+1} A v^{(k+1)}$, which leads to $r^{(k)} \cdot A v^{(k+1)} = r^{(k)} \cdot r^{(k)}/\alpha_{k+1} \neq 0$
• **Lemma** (Kelly). Let $A$ be spd and assume that $r^{(k)} \neq 0$. Then

$$\alpha_k = \frac{\langle r^{(k-1)}, r^{(k-1)} \rangle}{\langle v^{(k)}, Av^{(k)} \rangle}$$

And

$$\beta_k = \frac{\langle r^{(k)}, r^{(k)} \rangle}{\langle r^{(k-1)}, r^{(k-1)} \rangle}$$

• **Fact**: Since $x^{(k+1)} = x^{(k)} + \alpha_{k+1} v^{(k+1)}$, $r^{(k+1)} = r^{(k)} - \alpha_{k+1} Av^{(k+1)}$. 
Algorithm of CG Method

Let $x^{(0)}$ be initial guess.
Set $r^{(0)} = b - Ax^{(0)}; v^{(1)} = r^{(0)}$.

\textbf{for} $k = 1, 2, ...$

\begin{align*}
\alpha_k &= \frac{\langle r^{(k-1)}, r^{(k-1)} \rangle}{\langle v^{(k)}, Av^{(k)} \rangle} \\
&= \frac{\langle r^{(k)}, r^{(k)} \rangle}{\langle r^{(k-1)}, r^{(k-1)} \rangle} \\
&= \frac{\langle r^{(k)}, r^{(k)} \rangle}{\langle r^{(k-1)}, r^{(k-1)} \rangle} \\
&= \frac{\langle r^{(k)}, r^{(k)} \rangle}{\langle r^{(k-1)}, r^{(k-1)} \rangle}
\end{align*}

$x^{(k)} = x^{(k-1)} + \alpha_k v^{(k)}$

$r^{(k)} = r^{(k-1)} - \alpha_k Av^{(k)}$ \quad \text{// construct residual}

$\rho_k = \langle r^{(k)}, r^{(k)} \rangle$

\textbf{if} $\sqrt{\rho_k} < \varepsilon$ \textbf{exit.} \quad \text{// convergence test}

\begin{align*}
S_k &= \frac{\langle r^{(k)}, r^{(k)} \rangle}{\langle r^{(k-1)}, r^{(k-1)} \rangle} \\
&= \frac{\langle r^{(k)}, r^{(k)} \rangle}{\langle r^{(k-1)}, r^{(k-1)} \rangle} \\
&= \frac{\langle r^{(k)}, r^{(k)} \rangle}{\langle r^{(k-1)}, r^{(k-1)} \rangle}
\end{align*}

$v^{(k+1)} = r^{(k)} + S_k v^{(k)}$ \quad \text{// construct new search direction}

\textbf{end}
Remarks

• Constructed \( \{v^{(1)}, v^{(2)} \ldots \} \) are pair-wise A-orthogonal.

• Each iteration, there are one matrix-vector multiplication, two dot products and three scalar multiplications.

• Due to round-off errors, in practice, we need more than \( n \) iterations to get the solution.

• If the matrix \( A \) is ill-conditioned, the CG method is sensitive to round-off errors (CG is not good as Gaussian elimination with pivoting).

• Main usage of CG is as iterative method applied to bettered conditioned system.
CG as Krylov Subspace Method

**Theorem.** $x^{(k)}$ of the CG method minimizes the function $\phi(x)$ with respect to the subspace

$$K_k(A, r^{(0)}) = \text{span}\{r^{(0)}, Ar^{(0)}, A^2r^{(0)}, \ldots, A^{k-1}r^{(0)}\}.$$ 

I.e.

$$\phi(x^{(k)}) = \min_{c_i} \phi(x^{(0)} + \sum_{i=0}^{k-1} c_i A^i r^{(0)})$$

The subspace $K_k(A, r^{(0)})$ is called Krylov subspace.
Error Estimate

• Define an energy norm $\| \cdot \|_A$ of vector $\mathbf{u}$ with respect to matrix $A$: $\| \mathbf{u} \|_A = (\mathbf{u}^T A \mathbf{u})^{1/2}$

• Define the (algebraic) error $e^{(k)} = x^{(k)} - x^*$ where $x^*$ is the exact solution.

• Theorem.

$$\| x^{(k)} - x^* \|_A \leq 2 \left( \frac{\sqrt{\kappa(A)} - 1}{\sqrt{\kappa(A)} + 1} \right)^k \| x^{(0)} - x^* \|_A$$

with

$$\kappa(A) = \text{cond}(A) = \frac{\lambda_{\text{max}}(A)}{\lambda_{\text{min}}(A)} \geq 1.$$ 

Remark: Convergence is fast if matrix $A$ is well-conditioned.
Preconditioning

Let the symmetric positive definite matrix $M$ be a preconditioner for $A$ and $LL^T = M$ be its Cholesky factorization. $M^{-1}A$ is better conditioned than $A$ (and not necessarily symmetric).

The preconditioned system of equations is

$$M^{-1}Ax = M^{-1}b$$

or

$$L^{-T}L^{-1}Ax = L^{-T}L^{-1}b$$

where $L^{-T} = (L^T)^{-1}$.

Multiply with $L^T$ to obtain

$$L^{-1}AL^{-T}L^Tx = L^{-1}b$$

Define: $\tilde{A} = L^{-1}AL^{-T}$; $\tilde{x} = L^Tx$; $\tilde{b} = L^{-1}b$

Now apply CG to $\tilde{A}\tilde{x} = \tilde{b}$. 
Preconditioned CG for $M^{-1}Ax = M^{-1}b$

- **Definition:** Let $A, M$ be spd. The M-inner product $<\cdot,\cdot>_M$ is said to be $<x,y>_M = <Mx,y> = x^TMy$.

**Fact:**

1. $M^{-1}A$ is symmetric with respect to $<\cdot,\cdot>_M$, i.e.,
   $<M^{-1}Ax,y>_M = <x,M^{-1}Ay>_M$

2. $M^{-1}A$ is positive definite with respect to $<\cdot,\cdot>_M$, i.e.,
   $<M^{-1}Ax,x>_M > 0$ for all $x \neq 0$.

- We can apply the CG algorithm to $M^{-1}Ax = M^{-1}b$, replacing the standard inner product by the M-inner product.

  - Let $r = b - Ax$, $z = M^{-1}r$. Then $<z,z>_M = <r,z>$ and $<M^{-1}Av,v>_M = <Av,v>$

- Reference. Y. Saad. Iterative Methods for Sparse Linear Systems
Preconditioned CG Method

- Define $z^{(k)} = M^{-1}r^{(k)}$ to be the preconditioned residual.

Let $x^{(0)}$ be initial guess.

Set $r^{(0)} = b - Ax^{(0)}$; Solve $Mz^{(0)} = r^{(0)}$ for $z^{(0)}$

Set $v^{(1)} = z^{(0)}$

for $k = 1, 2, ...$

$$
\alpha_k = \frac{\langle z^{(k-1)}, r^{(k-1)} \rangle}{\langle v^{(k)}, Av^{(k)} \rangle}
$$

$$
x^{(k)} = x^{(k-1)} + \alpha_k v^{(k)}
$$

$$
r^{(k)} = r^{(k-1)} - \alpha_k Av^{(k)}
$$

solve $Mz^{(k)} = r^{(k)}$ for $z^{(k)}$

$$
\rho_k = \langle r^{(k)}, r^{(k)} \rangle
$$

if $\sqrt{\rho_k} < \varepsilon$ exit. //convergence test

$$
S_k = \frac{\langle z^{(k)}, r^{(k)} \rangle}{\langle z^{(k-1)}, r^{(k-1)} \rangle}
$$

$$
v^{(k+1)} = z^{(k)} + S_k v^{(k)}
$$
end
Split Preconditioner CG for $\tilde{A}\tilde{x} = \tilde{b}$

- $M$ is a Cholesky product.
- Define $\tilde{v}^{(k)} = L^T v^{(k)}$, $\tilde{x} = L^T x$, $\tilde{r}^{(k)} = L^T z^{(k)} = L^{-1}r^{(k)}$, $\tilde{A} = L^{-1}AL^{-T}$.
- Fact:
  - $< r^{(k)}, z^{(k)} > = < r^{(k)}, L^{-T}L^{-1}r^{(k)} > = < \tilde{r}^{(k)}, \tilde{r}^{(k)} >$.
  - $< A v^{(k)}, v^{(k)} > = < AL^{-T}\tilde{v}^{(k)}, L^{-T}\tilde{v}^{(k)} > = < \tilde{A}\tilde{v}^{(k)}, \tilde{v}^{(k)} >$.
  - With new variables, the preconditioned CG method solves $\tilde{A}\tilde{x} = \tilde{b}$. 


Split Preconditioner CG

Let $x^{(0)}$ be initial guess.
Set $r^{(0)} = b - Ax^{(0)}$; $\hat{r}^{(0)} = L^{-1}r^{(0)}$ and $v^{(1)} = L^{-T}\hat{r}^{(0)}$

for $k = 1, 2, ...$

$$\alpha_k = \frac{\langle \hat{r}^{(k-1)}, \hat{r}^{(k-1)} \rangle}{\langle v^{(k)}, Av^{(k)} \rangle}$$

$$x^{(k)} = x^{(k-1)} + \alpha_k v^{(k)}$$

$$\hat{r}^{(k)} = \hat{r}^{(k-1)} - \alpha_k L^{-1}Av^{(k)}$$

$$\rho_k = \langle r^{(k)}, r^{(k)} \rangle$$

if $\sqrt{\rho_k} < \varepsilon$ exit. //convergence test

$$s_k = \frac{\langle \hat{r}^{(k)}, \hat{r}^{(k)} \rangle}{\langle \hat{r}^{(k-1)}, \hat{r}^{(k-1)} \rangle}$$

$$v^{(k+1)} = L^{-T}\hat{r}^{(k)} + s_k v^{(k)}$$

day
Incomplete Cholesky Factorization

• Assume $A$ is symmetric and positive definite. $A$ is sparse.
• Factor $A = LL^T + R$, $R \neq 0$. $L$ has similar sparse structure as $A$.

```plaintext
for $k = 1, \ldots, n$
    $l_{kk} = \sqrt{a_{kk}}$
    for $i = k + 1, \ldots, n$
        $l_{ik} = \frac{a_{ik}}{l_{kk}}$
        for $j = k + 1, \ldots, n$
            if $a_{ij} = 0$ then
                $l_{ij} = 0$
            else
                $a_{ij} = a_{ij} - l_{ik}l_{kj}$
            endif
        endfor
    endfor
endfor
```
In diagonal or Jacobi preconditioning

\[ M = \text{diag}(A) \]

• Jacobi preconditioning is cheap if it works, i.e. solving \( Mz^{(k)} = r^{(k)} \) for \( z^{(k)} \) almost cost nothing.

References

• CT. Kelley. Iterative Methods for Linear and Nonlinear Equations
Parallel CG Algorithm

- Assume a row-wise block-striped decomposition of matrix $A$ and partition all vectors uniformly among tasks.

Let $x^{(0)}$ be initial guess.
Set $r^{(0)} = b - Ax^{(0)}$; Solve $Mz^{(0)} = r^{(0)}$ for $z^{(0)}$
Set $v^{(1)} = z^{(0)}$

for $k = 1, 2, ...$

\[ g = Av^{(k)} \] // parallel matrix-vector multiplication
\[ zr = \langle z^{(k-1)}, r^{(k-1)} \rangle \] // parallel dot product by MPI_Allreduce
\[ \alpha_k = \frac{zr}{\langle v^{(k)}, g \rangle} \] // parallel dot product by MPI_Allreduce
\[ x^{(k)} = x^{(k-1)} + \alpha_k v^{(k)} \] //
\[ r^{(k)} = r^{(k-1)} - \alpha_k g \] //
solve $Mz^{(k)} = r^{(k)}$ for $z^{(k)}$ // Solve matrix system, can involve additional complexity
\[ \rho_k = \langle r^{(k)}, r^{(k)} \rangle \] // MPI_Allreduce
if $\sqrt{\rho_k} < \epsilon$ exit. // convergence test
\[ zr_n = \langle z^{(k)}, r^{(k)} \rangle \] // parallel dot product
\[ s_k = \frac{zr_n}{zr} \]
\[ v^{(k+1)} = r^{(k)} + s_k v^{(k)} \]
end