Carnegie Mellon

18-660: Numerical Methods for Engineering Design and Optimization

Xin Li
Department of ECE
Carnegie Mellon University
Pittsburgh, PA 15213

Electrical & Computer ENGINEERING

Overview

- **Conjugate Gradient Method (Part 3)**
  - Conjugate gradient method
  - Convergence rate

Conjugate Search Direction

- **Important equations about conjugate search direction**

$$AX = B \longrightarrow \text{Linear equation}$$

$$\min_X \quad f(X) = \frac{1}{2} X^T A X - B^T X + C \longrightarrow \text{Equivalent optimization}$$

$$R^{(k)} = B - AX^{(k)} \longrightarrow \text{Residual definition}$$

$$\nabla f\left[X^{(k)}\right] = AX^{(k)} - B = -R^{(k)} \longrightarrow \text{Residual vs. gradient}$$

Conjugate Search Direction

- **Important equations about conjugate search direction**

$$\left.\begin{aligned} X^{(k+1)} &= X^{(k)} + \mu^{(k)} D^{(k)} \\ \mu^{(k)} &= \frac{D^{(k)T} R^{(k)}}{D^{(k)T} A D^{(k)}} \end{aligned}\right\} \longrightarrow \text{Iteration scheme}$$

$$D^{(i)T} A D^{(j)} = 0 \longrightarrow \text{Conjugate search directions}$$

$$D^{(k)T} R^{(k+1)} = 0 \longrightarrow \text{Orthogonal residual}$$

Conjugate Gradient Method

- **Step 1: start from an initial guess $X^{(0)}$, and set k = 0**
- **Step 2: calculate**

$$D^{(0)} = R^{(0)} = B - AX^{(0)}$$

- **Step 3: update solution**

$$X^{(k+1)} = X^{(k)} + \mu^{(k)} D^{(k)} \quad \text{where} \quad \mu^{(k)} = \frac{D^{(k)T} R^{(k)}}{D^{(k)T} A D^{(k)}}$$

- **Step 4: calculate residual**

$$R^{(k+1)} = B - AX^{(k+1)}$$

- **Step 5: determine search direction**

$$D^{(k+1)} = R^{(k+1)} + \sum_{i=0}^{k} \beta_{k+1,i} D^{(i)} \quad \text{where} \quad \beta_{k+1,i} = -\frac{D^{(i)T} A R^{(k+1)}}{D^{(i)T} A D^{(i)}}$$

- **Step 6: set k = k + 1 and go to Step 3**

Conjugate Gradient Method

- **Step 1: start from an initial guess $X^{(0)}$, and set k = 0**
- **Step 2: calculate**

$$D^{(0)} = R^{(0)} = B - AX^{(0)}$$

- **Step 3: update solution**

$$X^{(k+1)} = X^{(k)} + \mu^{(k)} D^{(k)} \quad \text{where} \quad \mu^{(k)} = \frac{D^{(k)T} R^{(k)}}{D^{(k)T} A D^{(k)}}$$

- **Step 4: calculate residual**

$$\boxed{R^{(k+1)} = B - AX^{(k+1)}}$$

- **Step 5: determine search direction**

$$D^{(k+1)} = R^{(k+1)} + \sum_{i=0}^{k} \beta_{k+1,i} D^{(i)} \quad \text{where} \quad \beta_{k+1,i} = -\frac{D^{(i)T} A R^{(k+1)}}{D^{(i)T} A D^{(i)}}$$

- **Step 6: set k = k + 1 and go to Step 3**

Conjugate Gradient Method

$$R^{(k)} = B - AX^{(k)} \qquad X^{(k+1)} = X^{(k)} + \mu^{(k)} D^{(k)}$$

$$R^{(k+1)} = B - AX^{(k+1)}$$

$$R^{(k+1)} = B - A\left[X^{(k)} + \mu^{(k)} D^{(k)}\right]$$

$$= B - Ax^{(k)} - \mu^{(k)} AD^{(k)}$$

$$= R^{(k)} - \mu^{(k)} A D^{(k)}$$

Conjugate Gradient Method

- Step 1: start from an initial guess $X^{(0)}$, and set k = 0
- Step 2: calculate

$$D^{(0)} = R^{(0)} = B - AX^{(0)}$$

- Step 3: update solution

$$X^{(k+1)} = X^{(k)} + \mu^{(k)} D^{(k)} \quad \text{where} \quad \mu^{(k)} = \frac{D^{(k)T} R^{(k)}}{D^{(k)T} \underline{AD^{(k)}}}$$

- Step 4: calculate residual $B - AX^{(k)}$

$$R^{(k+1)} = R^{(k)} - \mu^{(k)} \underline{AD^{(k)}}$$

- Step 5: determine search direction

$$D^{(k+1)} = R^{(k+1)} + \sum_{i=0}^{k} \beta_{k+1,i} D^{(i)} \quad \text{where} \quad \beta_{k+1,i} = -\frac{D^{(i)T} AR^{(k+1)}}{D^{(i)T} AD^{(i)}}$$

- Step 6: set k = k + 1 and go to Step 3

Conjugate Gradient Method

- **Step 1: start from an initial guess $X^{(0)}$, and set k = 0**
- **Step 2: calculate**

$$D^{(0)} = R^{(0)} = B - AX^{(0)}$$

- **Step 3: update solution**

$$X^{(k+1)} = X^{(k)} + \mu^{(k)} D^{(k)} \quad \text{where} \quad \mu^{(k)} = \frac{D^{(k)T} R^{(k)}}{D^{(k)T} A D^{(k)}}$$

- **Step 4: calculate residual**

$$R^{(k+1)} = R^{(k)} - \mu^{(k)} A D^{(k)}$$

- **Step 5: determine search direction**

$$\boxed{D^{(k+1)} = R^{(k+1)} + \sum_{i=0}^{k} \beta_{k+1,i} D^{(i)} \quad \text{where} \quad \beta_{k+1,i} = -\frac{D^{(i)T} A R^{(k+1)}}{D^{(i)T} A D^{(i)}}}$$

- **Step 6: set k = k + 1 and go to Step 3**

Conjugate Gradient Method

- **Orthogonal residuals**

$$R^{(k+1)} = R^{(k)} - \mu^{(k)} A D^{(k)} \qquad D^{(i)T} A D^{(j)} = 0 \qquad D^{(k)T} R^{(k+1)} = 0$$

$D^{(k)T} R^{(k+1)} = 0$

$D^{(k)T} R^{(k+2)} = D^{(k)T} [R^{(k+1)} - \mu^{(k+1)} A D^{(k+1)}]$

$= D^{(k)T} R^{(k+1)} - \mu^{(k+1)} D^{(k)T} A D^{(k+1)}$

$= 0$

$D^{(k)T} R^{(k+3)} = 0$

$\vdots$

$D^{(i)} R^{(j)} = 0 \quad (i < j)$

Conjugate Gradient Method

- **Orthogonal residuals**

$$D^{(i)T}R^{(j)} = 0 \quad (i < j) \qquad span\{D^{(0)}, D^{(1)}, \cdots, D^{(k)}\} = span\{R^{(0)}, R^{(1)}, \cdots, R^{(k)}\}$$

$R^{(k)} \perp span\{D^{(0)}, D^{(1)}, \cdots, D^{(k-1)}\}$

$R^{(k)} \perp span\{R^{(0)}, \cdots, R^{(k-1)}\}$

$R^{(i)T} R^{(j)} = 0 \quad (i \neq j)$

Conjugate Gradient Method

$$D^{(k+1)} = R^{(k+1)} + \sum_{i=0}^{k} \beta_{k+1,i} D^{(i)} \quad \text{where} \quad \beta_{k+1,i} = -\frac{D^{(i)T} A R^{(k+1)}}{D^{(i)T} A D^{(i)}}$$

$$R^{(k+1)} = R^{(k)} - \mu^{(k)} A D^{(k)}$$

$R^{(i+1)T} R^{(k+1)} = [R^{(i)} - \mu^{(i)} A D^{(i)}]^T R^{(k+1)}$

$= R^{(i)T} R^{(k+1)} - \underline{\mu^{(i)} D^{(i)T} A R^{(k+1)}}$

$\mu^{(i)} D^{(i)T} A R^{(k+1)} = R^{(i)T} R^{(k+1)} - R^{(i+1)T} R^{(k+1)}$

Conjugate Gradient Method

$$D^{(k+1)} = R^{(k+1)} + \sum_{i=0}^{k} \beta_{k+1,i} D^{(i)} \quad \text{where} \quad \beta_{k+1,i} = -\frac{D^{(i)T} A R^{(k+1)}}{D^{(i)T} A D^{(i)}}$$

$$\mu^{(i)} D^{(i)T} A R^{(k+1)} = R^{(i)T} R^{(k+1)} - R^{(i+1)T} R^{(k+1)} \qquad R^{(i)T} R^{(j)} = 0$$

- $i = k$

$$\cancel{\mu^{(i)}} D^{(i)T} A R^{(k+1)} = R^{(i)T} R^{(k+1)} - R^{(k+1)T} R^{(k+1)}$$

$$= \frac{-R^{(k+1)T} R^{(k+1)}}{\mu^{(k)}}$$

- $i < k$

$$\mu^{(i)} D^{(i)T} A R^{(k+1)} = R^{(i)T} R^{(k+1)} - R^{(i+1)T} R^{(k+1)}$$

$$= 0 - 0 = 0$$

$$D^{(k+1)} = R^{(k+1)} + \beta_{k+1,k} D^{(k)}$$

$$\beta_{k+1,k} = \frac{R^{(k+1)T} R^{(k+1)}}{\mu^{(k)} \cdot D^{(k)T} A D^{(k)}}$$

Conjugate Gradient Method

- **Step 1: start from an initial guess $X^{(0)}$, and set k = 0**
- **Step 2: calculate**

$$D^{(0)} = R^{(0)} = B - AX^{(0)}$$

- **Step 3: update solution**

$$X^{(k+1)} = X^{(k)} + \mu^{(k)} D^{(k)} \quad \text{where} \quad \mu^{(k)} = \frac{D^{(k)T} R^{(k)}}{D^{(k)T} A D^{(k)}}$$

- **Step 4: calculate residual**

$$R^{(k+1)} = R^{(k)} - \mu^{(k)} A D^{(k)}$$

- **Step 5: determine search direction**

$$D^{(k+1)} = R^{(k+1)} + \beta_{k+1,k} D^{(k)} \quad \text{where} \quad \beta_{k+1,k} = \frac{R^{(k+1)T} R^{(k+1)}}{\mu^{(k)} D^{(k)T} A D^{(k)}}$$

- **Step 6: set k = k + 1 and go to Step 3**

Conjugate Gradient Method

- **Step 1: start from an initial guess $X^{(0)}$, and set k = 0**
- **Step 2: calculate**

$$D^{(0)} = R^{(0)} = B - AX^{(0)}$$

- **Step 3: update solution**

$$X^{(k+1)} = X^{(k)} + \mu^{(k)} D^{(k)} \quad \text{where} \quad \mu^{(k)} = \frac{D^{(k)T} R^{(k)}}{D^{(k)T} A D^{(k)}}$$

- **Step 4: calculate residual**

$$R^{(k+1)} = R^{(k)} - \mu^{(k)} A D^{(k)}$$

- **Step 5: determine search direction**

$$\boxed{D^{(k+1)} = R^{(k+1)} + \beta_{k+1,k} D^{(k)} \quad \text{where} \quad \beta_{k+1,k} = \frac{R^{(k+1)T} R^{(k+1)}}{\mu^{(k)} D^{(k)T} A D^{(k)}}}$$

- **Step 6: set k = k + 1 and go to Step 3**

Conjugate Gradient Method

$$D^{(k+1)} = R^{(k+1)} + \beta_{k+1,k} D^{(k)} \quad \text{where} \quad \beta_{k+1,k} = \frac{R^{(k+1)T} R^{(k+1)}}{\mu^{(k)} D^{(k)T} A D^{(k)}}$$

$$\mu^{(k)} = \frac{D^{(k)T} R^{(k)}}{D^{(k)T} A D^{(k)}}$$

$$\mu^{(k)} D^{(k)T} A D^{(k)} = D^{(k)T} R^{(k)}$$

$$\beta_{k+1,k} = \frac{R^{(k+1)T} R^{(k+1)}}{D^{(k)T} R^{(k)}}$$

Conjugate Gradient Method

- **Step 1: start from an initial guess $X^{(0)}$, and set k = 0**
- **Step 2: calculate**

$$D^{(0)} = R^{(0)} = B - AX^{(0)}$$

- **Step 3: update solution**

$$X^{(k+1)} = X^{(k)} + \mu^{(k)} D^{(k)} \quad \text{where} \quad \mu^{(k)} = \frac{D^{(k)T} R^{(k)}}{D^{(k)T} A D^{(k)}}$$

- **Step 4: calculate residual**

$$R^{(k+1)} = R^{(k)} - \mu^{(k)} A D^{(k)}$$

- **Step 5: determine search direction**

$$D^{(k+1)} = R^{(k+1)} + \beta_{k+1,k} D^{(k)} \quad \text{where} \quad \beta_{k+1,k} = \frac{R^{(k+1)T} R^{(k+1)}}{D^{(k)T} R^{(k)}}$$

- **Step 6: set k = k + 1 and go to Step 3**

Convergence Rate

- **Mathematical analysis of convergence is quite complex**
  - We will directly show the results, but not detailed analysis

- **Convergence rate of conjugate gradient method**

$$AX = B$$

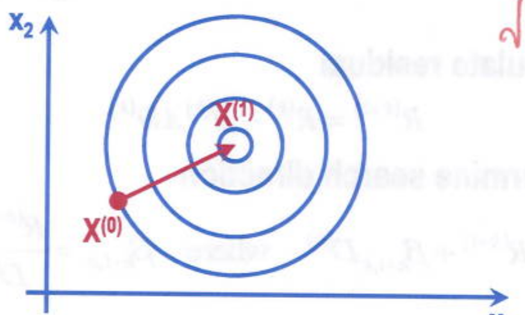
$$\Big\|\underbrace{X^{(k+1)} - X}_{\text{Exact solution}}\Big\| \le \left[\frac{\sqrt{\kappa(A)} - 1}{\sqrt{\underbrace{\kappa(A)}_{\text{Condition number}}} + 1}\right]^k \cdot \left\|X^{(0)} - X\right\|$$

Convergence Rate

$$\left\|X^{(k+1)} - X\right\| \le \left[\frac{\sqrt{\kappa(A)}-1}{\sqrt{\kappa(A)}+1}\right]^k \cdot \left\|X^{(0)} - X\right\|$$

- **Property 1: Converge by one iteration if κ(A) = 1**
  - E.g., A is an identity matrix

$\sqrt{\kappa(A)} - 1 = 0$



Convergence Rate

$$\left\|X^{(k+1)} - X\right\| \le \left[\frac{\sqrt{\kappa(A)}-1}{\sqrt{\kappa(A)}+1}\right]^k \cdot \left\|X^{(0)} - X\right\|$$

- **Property 2: slowly converge if κ(A) is large**
  - I.e., AX = B is ill-conditioned

- **In this case, we want to improve convergence rate by pre-conditioning**
  - Scale the matrix A to reduce its condition number

Summary

- **Conjugate gradient method (Part 3)**
  - Conjugate gradient method
  - Convergence rate

18-660: Numerical Methods for Engineering Design and Optimization

Xin Li
Department of ECE
Carnegie Mellon University
Pittsburgh, PA 15213

Electrical & Computer ENGINEERING

Overview

- Conjugate Gradient Method (Part 4)
  - Pre-conditioning
  - Nonlinear conjugate gradient method

Conjugate Gradient Method

- **Step 1: start from an initial guess $X^{(0)}$, and set k = 0**
- **Step 2: calculate**

$$D^{(0)} = R^{(0)} = B - AX^{(0)}$$

- **Step 3: update solution**

$$X^{(k+1)} = X^{(k)} + \mu^{(k)} D^{(k)} \quad \text{where} \quad \mu^{(k)} = \frac{D^{(k)T} R^{(k)}}{D^{(k)T} A D^{(k)}}$$

- **Step 4: calculate residual**

$$R^{(k+1)} = R^{(k)} - \mu^{(k)} A D^{(k)}$$

- **Step 5: determine search direction**

$$D^{(k+1)} = R^{(k+1)} + \beta_{k+1,k} D^{(k)} \quad \text{where} \quad \beta_{k+1,k} = \frac{R^{(k+1)T} R^{(k+1)}}{D^{(k)T} R^{(k)}}$$

- **Step 6: set k = k + 1 and go to Step 3**

Convergence Rate

$$\left\| X^{(k+1)} - X \right\| \le \left[\frac{\sqrt{k(A)} - 1}{\sqrt{k(A)} + 1} \right]^k \cdot \left\| X^{(0)} - X \right\|$$

- **Conjugate gradient method has slow convergence if k(A) is large**
  - I.e., AX = B is ill-conditioned

- **In this case, we want to improve convergence rate by pre-conditioning**

Pre-Conditioning

- **Key idea**
  - Convert AX = B to another equivalent equation $\tilde{A}\tilde{X} = \tilde{B}$
  - Solve $\tilde{A}\tilde{X} = \tilde{B}$ by conjugate gradient method

- **Important constraints to construct $\tilde{A}\tilde{X} = \tilde{B}$**
  - $\tilde{A}$ is symmetric and positive definite – so that we can solve it by conjugate gradient method
  - $\tilde{A}$ has a small condition number – so that we can achieve fast convergence

Pre-Conditioning

$$AX = B$$

$$L^{-1}AX = L^{-1}B$$

$$\underbrace{L^{-1}AL^{-T}}_{\tilde{A}}\underbrace{L^{T}X}_{\tilde{X}} = \underbrace{L^{-1}B}_{\tilde{B}}$$

$$\tilde{A}\cdot\tilde{X} = \tilde{B}$$

$\tilde{A}$ is symmetric and positive definite

$$(L^{-1}AL^{-T})^{T} = (L^{-T})^{T}A^{T}(L^{-1})^{T} = L^{-1}A\cdot L^{-T}$$

$$x^{T}L^{-1}AL^{-T}\cdot x = \underbrace{(L^{-T}x)^{T}}_{y^{T}}\cdot A\cdot\underbrace{(L^{-T}x)}_{y} > 0$$

Pre-Conditioning

$$\underbrace{L^{-1}AL^{-T}}_{\tilde{A}} \cdot \underbrace{L^{T}X}_{\tilde{X}} = \underbrace{L^{-1}B}_{\tilde{B}}$$

- **$L^{-1}AL^{-T}$ has a small condition number, if L is properly selected**

- **In theory, L can be optimally found by Cholesky decomposition**

$A = LL^T$

$\tilde{A} = L^{-1}AL^{-T} = \underbrace{L^{-1}L}_{I}\,\underbrace{L^{T}L^{-T}}_{I} = I$

  - However, Cholesky decomposition is not efficient for large, sparse problems
  - If we know Cholesky decomposition, we almost solve the equation – no need to use conjugate gradient method

Pre-Conditioning

$$\underbrace{L^{-1}AL^{-T}}_{\tilde{A}} \cdot \underbrace{L^{T}X}_{\tilde{X}} = \underbrace{L^{-1}B}_{\tilde{B}}$$

- **In practice, L can be constructed in many possible ways**

- **Diagonal pre-conditioning (or Jacobi pre-conditioning)**
  - Scale A along coordinate axes

$$L = \begin{bmatrix} \sqrt{a_{11}} & & & \\ & \sqrt{a_{22}} & & \\ & & \ddots & \\ & & & \ddots \end{bmatrix}$$

Pre-Conditioning

$$\underbrace{L^{-1}AL^{-T}}_{\tilde{A}} \cdot \underbrace{L^{T}X}_{\tilde{X}} = \underbrace{L^{-1}B}_{\tilde{B}}$$

- **Incomplete Cholesky pre-conditioning**

$$L = \begin{bmatrix} \times & & & 0 \\ \times & \times & & \\ \times & \times & \ddots & \\ \times & \times & & \ddots \end{bmatrix}$$

  - L is lower-triangular
  - Few or no fill-ins are allowed
  - $A \approx LL^T$ (not exactly equal)

Pre-Conditioning

- **Step 1: start from an initial guess $\tilde{X}^{(0)}$, and set k = 0**
- **Step 2: calculate**

$$\widetilde{D}^{(0)} = \widetilde{R}^{(0)} = L^{-1}B - L^{-1}AL^{-T}\widetilde{X}^{(0)}$$

- **Step 3: update solution**

$$\widetilde{X}^{(k+1)} = \widetilde{X}^{(k)} + \widetilde{\mu}^{(k)}\widetilde{D}^{(k)} \quad \text{where} \quad \widetilde{\mu}^{(k)} = \frac{\widetilde{D}^{(k)T}\widetilde{R}^{(k)}}{\widetilde{D}^{(k)T}L^{-1}AL^{-T}\widetilde{D}^{(k)}}$$

- **Step 4: calculate residual**

$$\widetilde{R}^{(k+1)} = \widetilde{R}^{(k)} - \widetilde{\mu}^{(k)}L^{-1}AL^{-T}\widetilde{D}^{(k)}$$

- **Step 5: determine search direction**

$$\widetilde{D}^{(k+1)} = \widetilde{R}^{(k+1)} + \widetilde{\beta}_{k+1,k}\widetilde{D}^{(k)} \quad \text{where} \quad \widetilde{\beta}_{k+1,k} = \frac{\widetilde{R}^{(k+1)T}\widetilde{R}^{(k+1)}}{\widetilde{D}^{(k)T}\widetilde{R}^{(k)}}$$

- **Step 6: set k = k + 1 and go to Step 3**

Pre-Conditioning

$$\underbrace{L^{-1}AL^{-T}}_{\tilde{A}} \cdot \underbrace{L^{T}X}_{\tilde{X}} = \underbrace{L^{-1}B}_{\tilde{B}}$$

$$\widetilde{D}^{(0)} = \widetilde{R}^{(0)} = L^{-1}B - L^{-1}AL^{-T}\widetilde{X}^{(0)}$$

$$\widetilde{X}^{(k+1)} = \widetilde{X}^{(k)} + \widetilde{\mu}^{(k)}\widetilde{D}^{(k)} \quad \text{where} \quad \widetilde{\mu}^{(k)} = \frac{\widetilde{D}^{(k)T}\widetilde{R}^{(k)}}{\widetilde{D}^{(k)T}L^{-1}AL^{-T}\widetilde{D}^{(k)}}$$

$$\widetilde{R}^{(k+1)} = \widetilde{R}^{(k)} - \widetilde{\mu}^{(k)}L^{-1}AL^{-T}\widetilde{D}^{(k)}$$

$$\widetilde{D}^{(k+1)} = \widetilde{R}^{(k+1)} + \widetilde{\beta}_{k+1,k}\widetilde{D}^{(k)} \quad \text{where} \quad \widetilde{\beta}_{k+1,k} = \frac{\widetilde{R}^{(k+1)T}\widetilde{R}^{(k+1)}}{\widetilde{D}^{(k)T}\widetilde{R}^{(k)}}$$

- **L^{-1} should not be explicitly computed**
  - Instead, $Y = L^{-1}W$ or $Y = L^{-T}W$ (where W is a vector) should be computed by solving linear equation $LY = W$ or $L^{T}Y = W$

Pre-Conditioning

- **Diagonal pre-conditioning**
  - L is a diagonal matrix
  - $Y = L^{-1}W$ or $Y = L^{-T}W$ can be found by simply scaling

$$\begin{bmatrix} \sqrt{a_{11}} & & \\ & \sqrt{a_{22}} & \\ & & \ddots \end{bmatrix} \cdot \begin{bmatrix} y_1 \\ y_2 \\ Y \end{bmatrix} = \begin{bmatrix} w_1 \\ w_2 \\ W \end{bmatrix}$$

$$y_1 = \frac{w_1}{\sqrt{a_{11}}}$$

$$y_2 = \frac{w_2}{\sqrt{a_{22}}}$$

Pre-Conditioning

- **Incomplete Cholesky pre-conditioning**
  - L is lower-triangular
  - Y = L^{-1}W or Y = L^{-T}W can be found by backward substitution

$$\begin{bmatrix} l_{11} & & \\ l_{21} & l_{22} & \\ l_{31} & l_{32} & \ddots \end{bmatrix} \cdot \begin{bmatrix} y_1 \\ y_2 \\ Y \\ \; \end{bmatrix} = \begin{bmatrix} w_1 \\ w_2 \\ W \\ \; \end{bmatrix}$$

$$y_1 = \frac{w_1}{l_{11}}$$

$$l_{21} y_1 + l_{22} y_2 = w_2$$

$$y_2 = \frac{w_2 - l_{21} y_1}{l_{22}}$$

Pre-Conditioning

$$\underbrace{L^{-1}AL^{-T}}_{\tilde{A}} \cdot \underbrace{L^{T}X}_{\tilde{X}} = \underbrace{L^{-1}B}_{\tilde{B}}$$

- **Once X̃ is known, X is calculated as X = L^{-T}X̃**
  - Solve linear equation L^{-T}X = X̃ by backward substitution

$$\begin{bmatrix} \sqrt{a_{11}} & 0 & 0 \\ & \sqrt{a_{22}} & 0 \\ & & \ddots \end{bmatrix} \cdot \begin{bmatrix} X \end{bmatrix} = \begin{bmatrix} \tilde{X} \end{bmatrix}$$

$$x_1 = \tilde{x}_1 / \sqrt{a_{11}}$$
$$x_2 = \tilde{x}_2 / \sqrt{a_{22}}$$
$$\vdots$$

Diagonal pre-conditioning

$$\begin{bmatrix} l_{11} & l_{21} & l_{31} \\ & l_{22} & l_{32} \\ & & \ddots \end{bmatrix} \cdot \begin{bmatrix} X \end{bmatrix} = \begin{bmatrix} \tilde{X} \end{bmatrix}$$

$$x_N = \tilde{x}_N / l_{NN}$$
$$x_{N-1} = \left(\tilde{x}_{N-1} - l_{N,N-1} x_N\right) / l_{N-1,N-1}$$
$$\vdots$$

Incomplete Cholesky pre-conditioning

Nonlinear Conjugate Gradient Method

- Conjugate gradient method can be extended to general (i.e., non-quadratic) unconstrained nonlinear optimization

$$\min_X \quad \frac{1}{2}X^T AX - B^T X + C \qquad\qquad \min_X \quad f(X)$$

$$\Downarrow$$

$$X = A^{-1}B$$

Quadratic programming

Nonlinear programming

- A number of changes must be made to solve nonlinear optimization problems

Nonlinear Conjugate Gradient Method

- Step 1: start from an initial guess $X^{(0)}$, and set k = 0
- Step 2: calculate

$$\boxed{D^{(0)} = R^{(0)} = B - AX^{(0)}}$$

- Step 3: update solution

$$X^{(k+1)} = X^{(k)} + \mu^{(k)}D^{(k)} \quad \text{where} \quad \boxed{\mu^{(k)} = \frac{D^{(k)T}R^{(k)}}{D^{(k)T}AD^{(k)}}}$$

- Step 4: calculate residual

$$\boxed{R^{(k+1)} = R^{(k)} - \mu^{(k)}AD^{(k)}}$$

- Step 5: determine search direction

$$\boxed{D^{(k+1)} = R^{(k+1)} + \beta_{k+1,k}D^{(k)} \quad \text{where} \quad \beta_{k+1,k} = \frac{R^{(k+1)T}R^{(k+1)}}{D^{(k)T}R^{(k)}}}$$

- Step 6: set k = k + 1 and go to Step 3

Nonlinear Conjugate Gradient Method

- **New definition of residual**

$$D^{(0)} = R^{(0)} = B - AX^{(0)}$$
$$R^{(k+1)} = R^{(k)} - \mu^{(k)} AD^{(k)}$$

Quadratic programming

Nonlinear programming

$$R^{(k)} = -\nabla f[x^{(k)}]$$

- **"Residual" is defined by the gradient of f(X)**
  - If X* is optimal, ∇f(X*) = 0
  - −∇f(X*) = B − AX for quadratic programming

Nonlinear Conjugate Gradient Method

- **New formula for conjugate search directions**

$$D^{(k+1)} = R^{(k+1)} + \beta_{k+1,k} D^{(k)} \quad \text{where} \quad \beta_{k+1,k} = \frac{R^{(k+1)T} R^{(k+1)}}{D^{(k)T} R^{(k)}}$$

Quadratic programming

- **Ideally, search directions should be computed by Gram-Schmidt conjugation of residues**
  - In practice, we often use approximate formulas

$$\beta_{k+1,k} = \frac{R^{(k+1)T} R^{(k+1)}}{R^{(k)T} R^{(k)}}$$

Fletcher-Reeves formula

$$\beta_{k+1,k} = \frac{R^{(k+1)T} \cdot \left[R^{(k+1)} - R^{(k)}\right]}{R^{(k)T} R^{(k)}}$$

Polak-Ribiere formula

Nonlinear Conjugate Gradient Method

- **Optimal step size calculated by one-dimensional search**

$$X^{(k+1)} = X^{(k)} + \mu^{(k)} D^{(k)} \quad \text{where} \quad \mu^{(k)} = \frac{D^{(k)T} R^{(k)}}{D^{(k)T} A D^{(k)}}$$

Quadratic programming

- **$\mu^{(k)}$ cannot be calculated analytically**
  - Optimize $\mu^{(k)}$ by one-dimensional search

$$\min_{\mu(k)} f[X^{(k+1)}] = f[X^{(k)} + \mu^{(k)} D^{(k)}]$$

Nonlinear Conjugate Gradient Method

- **Step 1: start from an initial guess $X^{(0)}$, and set k = 0**
- **Step 2: calculate**

$$D^{(0)} = R^{(0)} = -\nabla f[X^{(0)}]$$

- **Step 3: update solution**

$$\min_{\mu^{(k)}} f[X^{(k)} + \mu^{(k)} D^{(k)}] \qquad X^{(k+1)} = X^{(k)} + \mu^{(k)} D^{(k)}$$

- **Step 4: calculate residual**

$$R^{(k+1)} = -\nabla f[X^{(k+1)}]$$

- **Step 5: determine search direction (Fletcher-Reeves formula)**

$$\beta_{k+1,k} = \frac{R^{(k+1)T} R^{(k+1)}}{R^{(k)T} R^{(k)}} \qquad D^{(k+1)} = R^{(k+1)} + \beta_{k+1,k} D^{(k)}$$

- **Step 6: set k = k + 1 and go to Step 3**

Nonlinear Conjugate Gradient Method

- **Gradient method, conjugate gradient method and Newton method**
  - Conjugate gradient method is often preferred for many practical large-scale engineering problems

	Gradient	Conjugate Gradient	Newton
1st-Order Derivative	Yes	Yes	Yes
2nd-Order Derivative	No	No	Yes
Pre-conditioning	No	Yes	No
Cost per Iteration	Low	Low	High
Convergence Rate	Slow	Fast	Fast
Preferred Problem Size	Large	Large	Small

Summary

- **Conjugate gradient method (Part 4)**
  - Pre-conditioning
  - Nonlinear conjugate gradient method