

Métodos Iterativos para Solução de Sistemas Lineares

INF1608 – Análise Numérica

Waldemar Celes
celes@inf.puc-rio.br

Departamento de Informática, PUC-Rio



Sistemas Lineares

Resolução de sistemas lineares

- ▶ Eliminação de Gauss
 - ▶ Método direto
 - ▶ Solução exata (teoricamente)
 - ▶ Complexidade computacional: $O(n^3)$



Sistemas Lineares

Resolução de sistemas lineares

- ▶ Eliminação de Gauss
 - ▶ Método direto
 - ▶ Solução exata (teoricamente)
 - ▶ Complexidade computacional: $O(n^3)$

Métodos iterativos

- ▶ Solução, em geral, aproximada
- ▶ Complexidade computacional: $n\delta$
 - ▶ n é o número de iterações
 - ▶ δ é a complexidade em cada iteração



Sistemas Lineares

Resolução de sistemas lineares

- ▶ Eliminação de Gauss
 - ▶ Método direto
 - ▶ Solução exata (teoricamente)
 - ▶ Complexidade computacional: $O(n^3)$

Métodos iterativos

- ▶ Solução, em geral, aproximada
- ▶ Complexidade computacional: $n\delta$
 - ▶ n é o número de iterações
 - ▶ δ é a complexidade em cada iteração
- ▶ Em condições especiais
 - ▶ **Sistemas esparsos**
 - ▶ Complexidade de cada interação: $O(n)$
 - ▶ **Sistemas coerentes** (boas estimativas iniciais)
 - ▶ Número de iterações: $O(1)$
 - ▶ Complexidade computacional total: $O(n)$



Métodos Iterativos

Sistemas lineares

$$Ax = b$$

Método de Jacobi

- ▶ Forma de iteração de ponto fixo para sistemas de equações
- ▶ Aplica IPF na i -ésima equação para achar x_i



Métodos Iterativos

Sistemas lineares

$$Ax = b$$

Método de Jacobi

- ▶ Forma de iteração de ponto fixo para sistemas de equações
- ▶ Aplica IPF na i -ésima equação para achar x_i

Exemplo:

$$\begin{cases} 3x + y = 5 \\ x + 2y = 5 \end{cases}$$



Métodos Iterativos

Sistemas lineares

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

Método de Jacobi

- ▶ Forma de iteração de ponto fixo para sistemas de equações
- ▶ Aplica IPF na i -ésima equação para achar x_i

Exemplo:

$$\begin{cases} 3x + y = 5 \\ x + 2y = 5 \end{cases}$$

Reescrevendo:

$$\begin{cases} x = \frac{5-y}{3} \\ y = \frac{5-x}{2} \end{cases}$$



Métodos Iterativos

Método de Jacobi

$$\begin{cases} x = \frac{5-y}{3} \\ y = \frac{5-x}{2} \end{cases}$$

Iterando:

$$\begin{bmatrix} x_0 \\ y_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \rightarrow \begin{bmatrix} \frac{5}{3} \\ \frac{5}{2} \end{bmatrix} \rightarrow \begin{bmatrix} \frac{5}{6} \\ \frac{5}{3} \end{bmatrix} \rightarrow \begin{bmatrix} \frac{10}{9} \\ \frac{25}{12} \end{bmatrix} \rightarrow \dots$$



Métodos Iterativos

Método de Jacobi

$$\begin{cases} x = \frac{5-y}{3} \\ y = \frac{5-x}{2} \end{cases}$$

Iterando:

$$\begin{bmatrix} x_0 \\ y_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \rightarrow \begin{bmatrix} \frac{5}{3} \\ \frac{5}{2} \end{bmatrix} \rightarrow \begin{bmatrix} \frac{5}{6} \\ \frac{5}{3} \end{bmatrix} \rightarrow \begin{bmatrix} \frac{10}{9} \\ \frac{25}{12} \end{bmatrix} \rightarrow \dots$$

- ▶ Converge!
- ▶ Solução exata:

$$\begin{bmatrix} 1 \\ 2 \end{bmatrix}$$



Métodos Iterativos

Método de Jacobi

- Invertendo a ordem das equações

Exemplo:

$$\begin{cases} x + 2y = 5 \\ 3x + y = 5 \end{cases}$$



Métodos Iterativos

Método de Jacobi

- Invertendo a ordem das equações

Exemplo:

$$\begin{cases} x + 2y = 5 \\ 3x + y = 5 \end{cases}$$

Reescrevendo:

$$\begin{cases} x = 5 - 2y \\ y = 5 - 3x \end{cases}$$



Métodos Iterativos

Método de Jacobi

- Invertendo a ordem das equações

Exemplo:

$$\begin{cases} x + 2y = 5 \\ 3x + y = 5 \end{cases}$$

Reescrevendo:

$$\begin{cases} x = 5 - 2y \\ y = 5 - 3x \end{cases}$$

Iterando:

$$\begin{bmatrix} x_0 \\ y_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \rightarrow \begin{bmatrix} 5 \\ 5 \end{bmatrix} \rightarrow \begin{bmatrix} -5 \\ 10 \end{bmatrix} \rightarrow \begin{bmatrix} 25 \\ 20 \end{bmatrix} \rightarrow \dots$$



Métodos Iterativos

Método de Jacobi

- Invertendo a ordem das equações

Exemplo:

$$\begin{cases} x + 2y = 5 \\ 3x + y = 5 \end{cases}$$

Reescrevendo:

$$\begin{cases} x = 5 - 2y \\ y = 5 - 3x \end{cases}$$

Iterando:

$$\begin{bmatrix} x_0 \\ y_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \rightarrow \begin{bmatrix} 5 \\ 5 \end{bmatrix} \rightarrow \begin{bmatrix} -5 \\ 10 \end{bmatrix} \rightarrow \begin{bmatrix} 25 \\ 20 \end{bmatrix} \rightarrow \dots$$

- Não converge!



Métodos Iterativos

Definição: **Matriz Estritamente Diagonal Dominante**

- ▶ Uma matriz $A_{n \times n}$ é estritamente diagonal dominante se:

$$|a_{ii}| > \sum_{j \neq i} |a_{ij}|$$

- ▶ Isto é, o valor absoluto da diagonal é maior que a soma dos valores absolutos dos demais elementos da linha

No exemplo:

- ▶ 1o caso:

$$\begin{bmatrix} 3 & 1 \\ 1 & 2 \end{bmatrix}$$

- ▶ 2o caso:

$$\begin{bmatrix} 1 & 2 \\ 3 & 1 \end{bmatrix}$$



Métodos Iterativos

Formulação do Método de Jacobi

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

- ▶ Decomposição da matriz

$$A = L + D + U$$

- ▶ L : matriz com os elementos abaixo da diagonal
- ▶ D : matriz com os elementos da diagonal
- ▶ U : matriz com os elementos acima da diagonal



Métodos Iterativos

Formulação do Método de Jacobi

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

- Decomposição da matriz

$$A = L + D + U$$

- L : matriz com os elementos abaixo da diagonal
- D : matriz com os elementos da diagonal
- U : matriz com os elementos acima da diagonal
- Reescrevendo algebricamente:

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

$$(D + L + U)\mathbf{x} = \mathbf{b}$$

$$D\mathbf{x} = \mathbf{b} - (L + U)\mathbf{x}$$

$$\mathbf{x} = D^{-1}(\mathbf{b} - (L + U)\mathbf{x})$$

- notando que D tem inversa trivial



Métodos Iterativos

Algoritmo: Método de Jacobi

$\mathbf{x}_0 =$ estimativa inicial

$$\mathbf{x}_{k+1} = D^{-1}(\mathbf{b} - (L + U)\mathbf{x}_k), \quad k = 1, 2, 3, \dots$$



Métodos Iterativos

Algoritmo: Método de Jacobi

$\mathbf{x}_0 =$ estimativa inicial

$$\mathbf{x}_{k+1} = D^{-1}(\mathbf{b} - (L + U)\mathbf{x}_k), \quad k = 1, 2, 3, \dots$$

No exemplo:

$$\begin{bmatrix} 3 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 5 \\ 5 \end{bmatrix}$$



Métodos Iterativos

Algoritmo: Método de Jacobi

\mathbf{x}_0 = estimativa inicial

$$\mathbf{x}_{k+1} = D^{-1}(\mathbf{b} - (L + U)\mathbf{x}_k), \quad k = 1, 2, 3, \dots$$

No exemplo:

$$\begin{bmatrix} 3 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 5 \\ 5 \end{bmatrix}$$

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \end{bmatrix} = \begin{bmatrix} \frac{1}{3} & 0 \\ 0 & \frac{1}{2} \end{bmatrix} \left(\begin{bmatrix} 5 \\ 5 \end{bmatrix} - \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_k \\ y_k \end{bmatrix} \right) = \begin{bmatrix} \frac{5-y_k}{3} \\ \frac{5-x_k}{2} \end{bmatrix}$$



Métodos Iterativos

Método de Gauss-Seidel

- ▶ Iteração de Jacobi que usa valores mais recentes dentro da mesma iteração
 - ▶ Impede computação em paralelo
- ▶ Convergência mais rápida, em geral
 - ▶ Ainda condicionada a matriz estritamente diagonal dominante



Métodos Iterativos

Método de Gauss-Seidel

- ▶ Iteração de Jacobi que usa valores mais recentes dentro da mesma iteração
 - ▶ Impede computação em paralelo
- ▶ Convergência mais rápida, em geral
 - ▶ Ainda condicionada a matriz estritamente diagonal dominante

Exemplo:

$$\begin{cases} 3x + y = 5 \\ x + 2y = 5 \end{cases}$$



Métodos Iterativos

Método de Gauss-Seidel

- ▶ Iteração de Jacobi que usa valores mais recentes dentro da mesma iteração
 - ▶ Impede computação em paralelo
- ▶ Convergência mais rápida, em geral
 - ▶ Ainda condicionada a matriz estritamente diagonal dominante

Exemplo:

$$\begin{cases} 3x + y = 5 \\ x + 2y = 5 \end{cases}$$

Jacobi:

$$\begin{cases} x_{i+1} = \frac{5-y_i}{3} \\ y_{i+1} = \frac{5-x_i}{2} \end{cases}$$



Métodos Iterativos

Método de Gauss-Seidel

- ▶ Iteração de Jacobi que usa valores mais recentes dentro da mesma iteração
 - ▶ Impede computação em paralelo
- ▶ Convergência mais rápida, em geral
 - ▶ Ainda condicionada a matriz estritamente diagonal dominante

Exemplo:

$$\begin{cases} 3x + y = 5 \\ x + 2y = 5 \end{cases}$$

Jacobi:

$$\begin{cases} x_{i+1} = \frac{5-y_i}{3} \\ y_{i+1} = \frac{5-x_i}{2} \end{cases}$$

Gauss-Seidel:

$$\begin{cases} x_{i+1} = \frac{5-y_i}{3} \\ y_{i+1} = \frac{5-x_{i+1}}{2} \end{cases}$$



Métodos Iterativos

Exemplo:

$$\begin{cases} x = \frac{5-y}{3} \\ y = \frac{5-x}{2} \end{cases}$$

Iterações de Jacobi:

$$\begin{bmatrix} x_0 \\ y_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \rightarrow \begin{bmatrix} \frac{5}{3} \\ \frac{5}{2} \end{bmatrix} \rightarrow \begin{bmatrix} \frac{5}{6} \\ \frac{5}{3} \end{bmatrix} \rightarrow \begin{bmatrix} \frac{10}{9} \\ \frac{25}{12} \end{bmatrix} = \begin{bmatrix} 1.1111 \\ 2.0833 \end{bmatrix}$$



Métodos Iterativos

Exemplo:

$$\begin{cases} x = \frac{5-y}{3} \\ y = \frac{5-x}{2} \end{cases}$$

Iterações de Jacobi:

$$\begin{bmatrix} x_0 \\ y_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \rightarrow \begin{bmatrix} \frac{5}{3} \\ \frac{5}{2} \end{bmatrix} \rightarrow \begin{bmatrix} \frac{5}{6} \\ \frac{5}{3} \end{bmatrix} \rightarrow \begin{bmatrix} \frac{10}{9} \\ \frac{25}{12} \end{bmatrix} = \begin{bmatrix} 1.1111 \\ 2.0833 \end{bmatrix}$$

Iterações de Gauss-Seidel:

$$\begin{bmatrix} x_0 \\ y_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \rightarrow \begin{bmatrix} \frac{5}{3} \\ \frac{5}{3} \end{bmatrix} \rightarrow \begin{bmatrix} \frac{10}{9} \\ \frac{35}{18} \end{bmatrix} \rightarrow \begin{bmatrix} \frac{55}{54} \\ \frac{215}{108} \end{bmatrix} = \begin{bmatrix} 1.0185 \\ 1.9907 \end{bmatrix}$$



Métodos Iterativos

Algoritmo: Método de Gauss-Seidel

$\mathbf{x}_0 =$ estimativa inicial

$$\mathbf{x}_{k+1} = D^{-1}(\mathbf{b} - U\mathbf{x}_k - L\mathbf{x}_{k+1}), \quad k = 1, 2, 3, \dots$$

$$- \left[\begin{array}{c} \text{Upper triangular matrix } U \\ \text{Vector } \mathbf{x}_k \end{array} - \begin{array}{c} \text{Lower triangular matrix } L \\ \text{Vector } \mathbf{x}_{k+1} \end{array} \right]$$

- Na prática, trabalha-se com um único vetor (*in place*)

Métodos Iterativos

Relaxação sucessiva

$$\mathbf{x}_{k+1} = (1 - w)\mathbf{x}_k + w [\text{fórmula para determinar } \mathbf{x}_{k+1}]$$



Métodos Iterativos

Relaxação sucessiva

$$\mathbf{x}_{k+1} = (1 - w)\mathbf{x}_k + w [\text{fórmula para determinar } \mathbf{x}_{k+1}]$$

- ▶ Se $w < 1$, tem-se sub-relaxação (interpolação)
- ▶ Se $w > 1$, tem-se sobre-relaxação (extrapolação)



Métodos Iterativos

Relaxação sucessiva

$$\mathbf{x}_{k+1} = (1 - w)\mathbf{x}_k + w [\text{fórmula para determinar } \mathbf{x}_{k+1}]$$

- ▶ Se $w < 1$, tem-se sub-relaxação (interpolação)
 - ▶ Usado para convergir sistemas não convergentes ou acelerar a convergência amortecendo oscilações
- ▶ Se $w > 1$, tem-se sobre-relaxação (extrapolação)



Métodos Iterativos

Relaxação sucessiva

$$\mathbf{x}_{k+1} = (1 - w)\mathbf{x}_k + w [\text{fórmula para determinar } \mathbf{x}_{k+1}]$$

- ▶ Se $w < 1$, tem-se sub-relaxação (interpolação)
 - ▶ Usado para convergir sistemas não convergentes ou acelerar a convergência amortecendo oscilações
- ▶ Se $w > 1$, tem-se sobre-relaxação (extrapolação)
 - ▶ Usado para acelerar convergência de um sistema convergente
 - ▶ Esta estratégia é muito usada!



Métodos Iterativos

Custo computacional por iteração: $O(n^2)$

- ▶ Operação dominante: multiplicação de matriz por vetor



Métodos Iterativos

Custo computacional por iteração: $O(n^2)$

- ▶ Operação dominante: multiplicação de matriz por vetor

Sistemas esparsos

- ▶ Matriz com a grande maioria dos elementos nulos
- ▶ Armazenar e processar apenas os elementos não nulos

Exemplo

$$\begin{bmatrix} 3 & -1 & 0 & \cdots & 0 & 0 & 0.5 \\ -1 & 3 & -1 & \cdots & 0 & 0.5 & 0 \\ 0 & -1 & 3 & \cdots & 0.5 & 0 & 0 \\ & & & \ddots & & & \\ 0 & 0 & 0.5 & \cdots & 3 & -1 & 0 \\ 0 & 0.5 & 0 & \cdots & -1 & 3 & -1 \\ 0.5 & 0 & 0 & \cdots & 0 & -1 & 3 \end{bmatrix} \mathbf{x} = \begin{bmatrix} 2.5 \\ 1.5 \\ 1.5 \\ \vdots \\ 1.5 \\ 1.5 \\ 2.5 \end{bmatrix}, \text{ solução } \mathbf{x} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \\ 1 \\ 1 \end{bmatrix}$$



Métodos Iterativos

Sistemas esparsos

Exemplo: $n = 100\,000$

- ▶ Matriz cheia: $n^2 = 10^{10}$ valores
 - ▶ Memória: $8 \times 10^{10} = 80$ Gbytes!
 - ▶ Processamento por Eliminação de Gauss $O(n^3)$
 - ▶ 10^{15} operações



Métodos Iterativos

Sistemas esparsos

Exemplo: $n = 100\,000$

- ▶ Matriz cheia: $n^2 = 10^{10}$ valores
 - ▶ Memória: $8 \times 10^{10} = 80$ Gbytes!
 - ▶ Processamento por Eliminação de Gauss $O(n^3)$
 - ▶ 10^{15} operações
 - ▶ Suponha máquina de 10^8 flops
 - ▶ Tempo de processamento: $\frac{10^{15}}{10^8} = 10^7$ segundos!
 - ▶ Um ano tem 3×10^7 segundos



Métodos Iterativos

Sistemas esparsos

Exemplo: $n = 100\,000$

- ▶ Matriz cheia: $n^2 = 10^{10}$ valores
 - ▶ Memória: $8 \times 10^{10} = 80$ Gbytes!
 - ▶ Processamento por Eliminação de Gauss $O(n^3)$
 - ▶ 10^{15} operações
 - ▶ Suponha máquina de 10^8 flops
 - ▶ Tempo de processamento: $\frac{10^{15}}{10^8} = 10^7$ segundos!
 - ▶ Um ano tem 3×10^7 segundos
- ▶ Considerando esparsidade da matriz
 - ▶ Memória: $4n$
 - ▶ Número de operações: $2 \times 4n = 8 \times 10^5$
 - ▶ 1 segundo por iteração!



Reformulando o Problema

Sistema linear

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

Exemplo:

$$\begin{bmatrix} 3 & 2 \\ 2 & 6 \end{bmatrix} \mathbf{x} = \begin{bmatrix} 2 \\ -8 \end{bmatrix}$$



Reformulando o Problema

Sistema linear

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

Exemplo:

$$\begin{bmatrix} 3 & 2 \\ 2 & 6 \end{bmatrix} \mathbf{x} = \begin{bmatrix} 2 \\ -8 \end{bmatrix}$$

► Solução:

$$\mathbf{x} = \begin{bmatrix} 2 \\ -2 \end{bmatrix}$$



Reformulando o Problema

Sistema linear

$$Ax = b$$

Exemplo:

$$\begin{bmatrix} 3 & 2 \\ 2 & 6 \end{bmatrix} \mathbf{x} = \begin{bmatrix} 2 \\ -8 \end{bmatrix}$$

► Solução:

$$\mathbf{x} = \begin{bmatrix} 2 \\ -2 \end{bmatrix}$$

- Interpretação geométrica
 - Interseção dos hiperplanos

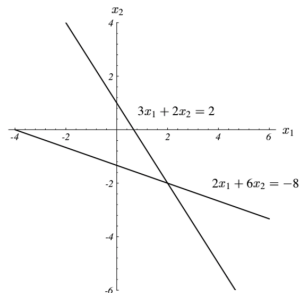


Figura extraída de "An Introduction to the Conjugate Gradient Method Without the Agonizing Pain", Shewchuk, 1994



Reformulando o Problema

Forma Quadrática

- ▶ Equação escalar na forma:

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

- ▶ onde c é um valor escalar (p.e. $c = 0$)



Reformulando o Problema

Forma Quadrática

- Equação escalar na forma:

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

- onde c é um valor escalar (p.e. $c = 0$)

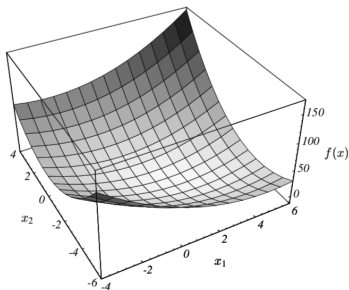


Figura extraída de "An Introduction to the Conjugate Gradient Method Without the Agonizing Pain", Shewchuk, 1994

Reformulando o Problema

Gradiente da forma quadrática

$$f'(x) = A\mathbf{x} - b$$



Reformulando o Problema

Gradiente da forma quadrática

$$f'(x) = A\mathbf{x} - \mathbf{b}$$

- ▶ Ponto crítico da forma quadrática

$$f'(x) = 0$$

- ▶ que equivale a achar a solução de $A\mathbf{x} = \mathbf{b}$



Reformulando o Problema

Gradiente da forma quadrática

$$f'(x) = A\mathbf{x} - \mathbf{b}$$

- ▶ Ponto crítico da forma quadrática

$$f'(x) = 0$$

- ▶ que equivale a achar a solução de $A\mathbf{x} = \mathbf{b}$
- ▶ Sendo A positiva definida, será sempre um **ponto de mínimo**



Reformulando o Problema

Gradiente da forma quadrática

$$f'(x) = Ax - b$$

- ▶ Ponto crítico da forma quadrática

$$f'(x) = 0$$

- ▶ que equivale a achar a solução de $Ax = b$
- ▶ Sendo A positiva definida, será sempre um **ponto de mínimo**

Determinar a solução de $Ax = b$ é equivalente a encontrar o ponto mínimo de $f(x)$

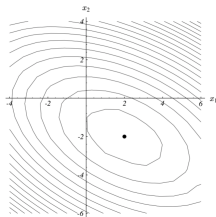


Figura extraída de "An Introduction to the Conjugate Gradient Method Without the Agonizing Pain", Shewchuk, 1994

Determinação do Mínimo da Função

Erro da solução

$$\mathbf{e}_k = \mathbf{x}_k - \mathbf{x}_s$$

Resíduo

$$\mathbf{r}_k = \mathbf{b} - A\mathbf{x}_k$$

Logo:

$$\mathbf{r}_k = \mathbf{b} - A(\mathbf{e}_k + \mathbf{x}_s)$$

$$\mathbf{r}_k = -A\mathbf{e}_k$$

- O resíduo é o erro transformado por A no espaço de \mathbf{b}



Determinação do Mínimo da Função

Tomar a direção inversa do gradiente

- Converge, mas resulta em muitas iterações

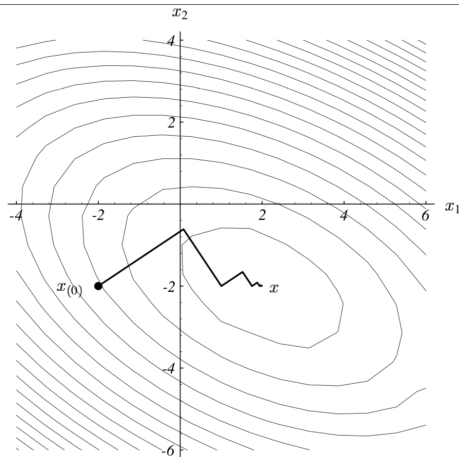


Figura extraída de "An Introduction to the Conjugate Gradient Method Without the Agonizing Pain", Shewchuk, 1994

Determinação do Mínimo da Função

Buscar as direções ortogonais $\mathbf{d}_0, \mathbf{d}_1, \dots, \mathbf{d}_{n-1}$

- ▶ Erro expresso como combinação linear de \mathbf{d}'_k s
- ▶ Um passo em cada direção
 - ▶ Elimina cada componente do erro na direção \mathbf{d}_k



Determinação do Mínimo da Função

Buscar as direções ortogonais $\mathbf{d}_0, \mathbf{d}_1, \dots, \mathbf{d}_{n-1}$

- ▶ Erro expresso como combinação linear de \mathbf{d}'_k s
- ▶ Um passo em cada direção
 - ▶ Elimina cada componente do erro na direção \mathbf{d}_k

Procedimento

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k$$

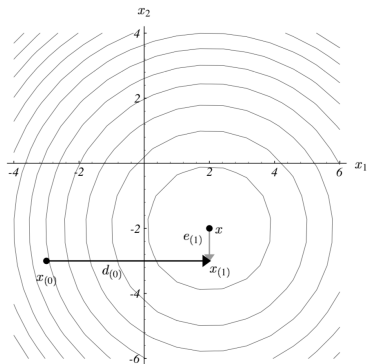


Figura extraída de "An Introduction to the Conjugate Gradient Method Without the Agonizing Pain", Shewchuk, 1994

Determinação do Mínimo da Função

Buscar as direções ortogonais $\mathbf{d}_0, \mathbf{d}_1, \dots, \mathbf{d}_{n-1}$

- ▶ Erro expresso como combinação linear de \mathbf{d}'_k s
- ▶ Um passo em cada direção
 - ▶ Elimina cada componente do erro na direção \mathbf{d}_k

Procedimento

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k$$

- ▶ Determinação de α_k
 - ▶ \mathbf{e}_{k+1} ortogonal a \mathbf{d}_k

$$\mathbf{d}_k^T \mathbf{e}_{k+1} = 0$$

$$\mathbf{d}_k^T (\mathbf{e}_k + \alpha_k \mathbf{d}_k) = 0$$

$$\alpha_k = -\frac{\mathbf{d}_k^T \mathbf{e}_k}{\mathbf{d}_k^T \mathbf{d}_k}$$

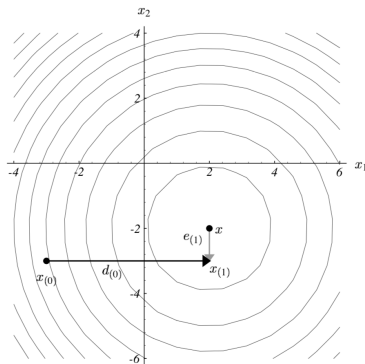


Figura extraída de "An Introduction to the Conjugate Gradient Method Without the Agonizing Pain", Shewchuk, 1994

Determinação do Mínimo da Função

Buscar as direções ortogonais $\mathbf{d}_0, \mathbf{d}_1, \dots, \mathbf{d}_{n-1}$

- ▶ Erro expresso como combinação linear de \mathbf{d}'_k s
- ▶ Um passo em cada direção
 - ▶ Elimina cada componente do erro na direção \mathbf{d}_k

Procedimento

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k$$

- ▶ Determinação de α_k
 - ▶ \mathbf{e}_{k+1} ortogonal a \mathbf{d}_k

$$\mathbf{d}_k^T \mathbf{e}_{k+1} = 0$$

$$\mathbf{d}_k^T (\mathbf{e}_k + \alpha_k \mathbf{d}_k) = 0$$

$$\alpha_k = -\frac{\mathbf{d}_k^T \mathbf{e}_k}{\mathbf{d}_k^T \mathbf{d}_k}$$

- ▶ Mas não conhecemos \mathbf{e}_k

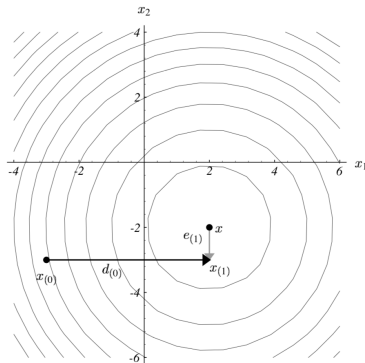


Figura extraída de "An Introduction to the Conjugate Gradient Method Without the Agonizing Pain", Shewchuk, 1994

Determinação do Mínimo da Função

Solução

- ▶ Busca em direções ortogonais em A
 - ▶ Conjugados em A

$$\mathbf{d}_i^T \mathbf{A} \mathbf{d}_j = 0$$

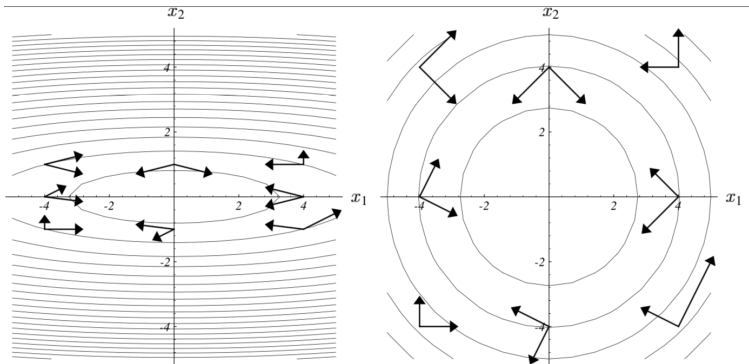


Figura extraída de "An Introduction to the Conjugate Gradient Method Without the Agonizing Pain", Shewchuk, 1994

Método dos Gradientes Conjugados

Método Gradiente Conjugado

- ▶ Aplicado a sistema cuja matriz é simétrica positiva definida



Método dos Gradientes Conjugados

Método Gradiente Conjugado

- ▶ Aplicado a sistema cuja matriz é simétrica positiva definida
- ▶ Dada uma estimativa inicial \mathbf{x}_0
 - ▶ Elimina, um a um, os n componentes ortogonais do erro



Método dos Gradientes Conjugados

Método Gradiente Conjugado

- ▶ Aplicado a sistema cuja matriz é simétrica positiva definida
- ▶ Dada uma estimativa inicial \mathbf{x}_0
 - ▶ Elimina, um a um, os n componentes ortogonais do erro
- ▶ Método direto *versus* iterativo
 - ▶ Converge para solução exata após n iterações
 - ▶ Resultado pode ser satisfatório antes de n iterações
 - ▶ Considerando uma determinada tolerância numérica



Método Gradiente Conjugado

Definição: **Produto escalar em A**

Sendo A uma matriz simétrica positiva definida, dois vetores \mathbf{v} e \mathbf{w} de dimensão n definem um produto escalar em A :

$$(\mathbf{v}, \mathbf{w})_A = \mathbf{v}^T A \mathbf{w}$$

- \mathbf{v} e \mathbf{w} são **ortogonais em A (conjugados)** se $\mathbf{v}^T A \mathbf{w} = 0$.



Método Gradiente Conjugado

Definição: **Produto escalar em A**

Sendo A uma matriz simétrica positiva definida, dois vetores \mathbf{v} e \mathbf{w} de dimensão n definem um produto escalar em A :

$$(\mathbf{v}, \mathbf{w})_A = \mathbf{v}^T A \mathbf{w}$$

- \mathbf{v} e \mathbf{w} são **ortogonais em A (conjugados)** se $\mathbf{v}^T A \mathbf{w} = 0$.

Lembrando que um produto interno entre dois vetores é zero se eles forem ortogonais



Método Gradiente Conjugado

Definição: **Produto escalar em A**

Sendo A uma matriz simétrica positiva definida, dois vetores \mathbf{v} e \mathbf{w} de dimensão n definem um produto escalar em A :

$$(\mathbf{v}, \mathbf{w})_A = \mathbf{v}^T A \mathbf{w}$$

- ▶ \mathbf{v} e \mathbf{w} são **ortogonais em A (conjugados)** se $\mathbf{v}^T A \mathbf{w} = 0$.

Lembrando que um produto interno entre dois vetores é zero se eles forem ortogonais

- ▶ Como A é simétrica: $(\mathbf{v}, \mathbf{w})_A = (\mathbf{w}, \mathbf{v})_A$
- ▶ Como A é positiva definida: $(\mathbf{v}, \mathbf{v})_A > 0 \quad \forall \mathbf{v} \neq 0$



Método Gradiente Conjugado

Atualização do vetor solução

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k$$

Atualização do vetor resíduo

► Sabe-se:

$$A\mathbf{x}_{k+1} + \mathbf{r}_{k+1} = A\mathbf{x}_k + \mathbf{r}_k$$

► Então:

$$A(\mathbf{x}_k + \alpha_k \mathbf{d}_k) + \mathbf{r}_{k+1} = A\mathbf{x}_k + \mathbf{r}_k$$

$$A\mathbf{x}_k + \alpha_k A\mathbf{d}_k + \mathbf{r}_{k+1} = A\mathbf{x}_k + \mathbf{r}_k$$

$$\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k A\mathbf{d}_k$$



Método Gradiente Conjugado

Escolha de α_k

- ▶ Como \mathbf{d}_k deve ser ortogonal a \mathbf{r}_{k+1}

$$\mathbf{d}_k^T \mathbf{r}_{k+1} = 0$$

$$\mathbf{d}_k^T (\mathbf{r}_k - \alpha_k A \mathbf{d}_k) = 0$$

$$\mathbf{d}_k^T \mathbf{r}_k = \alpha_k \mathbf{d}_k^T A \mathbf{d}_k$$

$$\therefore \alpha_k = \frac{\mathbf{d}_k^T \mathbf{r}_k}{\mathbf{d}_k^T A \mathbf{d}_k}$$

- ▶ Que pode ser re-escrito como:

$$\alpha_k = \frac{\mathbf{r}_k^T \mathbf{r}_k}{\mathbf{d}_k^T A \mathbf{d}_k}$$



Método Gradiente Conjugado

Escolha da nova direção de busca

- Assumindo que \mathbf{d}_k é perpendicular a \mathbf{r}_{k+1}

$$\mathbf{d}_{k+1} = \mathbf{r}_{k+1} + \beta_k \mathbf{d}_k$$

- Pré-multiplicando por $\mathbf{d}_k^T A$:

$$\mathbf{d}_k^T A \mathbf{d}_{k+1} = \mathbf{d}_k^T A \mathbf{r}_{k+1} + \beta_k \mathbf{d}_k^T A \mathbf{d}_k = 0$$

$$\therefore \beta_k = \frac{\mathbf{d}_k^T A \mathbf{r}_{k+1}}{\mathbf{d}_k^T A \mathbf{d}_k}$$

- Que pode ser re-escrito como:

$$\beta_k = \frac{\mathbf{r}_{k+1}^T \mathbf{r}_{k+1}}{\mathbf{r}_k^T \mathbf{r}_k}$$



Método Gradiente Conjugado

Algoritmo: Gradiente Conjugado

$x_0 =$ estimativa inicial

$d_0 = r_0 = b - Ax$

for $k = 0, 1, \dots, n - 1$ **do**

if $\|r_k\|_2 < tol$ **then**

stop

$$\alpha_k = \frac{r_k^T r_k}{d_k^T A d_k}$$

$$x_{k+1} = x_k + \alpha_k d_k$$

$$r_{k+1} = r_k - \alpha_k A d_k$$

$$\beta_k = \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}$$

$$d_{k+1} = r_{k+1} + \beta_k d_k$$



Método Gradiente Conjugado

Uso de preconditionadores

- ▶ Diminuir o número de condicionamento de A
- ▶ Acelerar convergência

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

$$M^{-1}A\mathbf{x} = M^{-1}\mathbf{b}$$

- ▶ onde M é o preconditionador



Método Gradiente Conjugado

Uso de preconditionadores

- ▶ Diminuir o número de condicionamento de A
- ▶ Acelerar convergência

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

$$M^{-1}A\mathbf{x} = M^{-1}\mathbf{b}$$

- ▶ onde M é o preconditionador

Escolha do preconditionador

- ▶ O mais próximo de A possível
 - ▶ Para $M^{-1}A$ resultar numa matriz bem condicionada
- ▶ Fácil de inverter



Método Gradiente Conjugado

Uso de preconditionadores

- ▶ Diminuir o número de condicionamento de A
- ▶ Acelerar convergência

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

$$M^{-1}A\mathbf{x} = M^{-1}\mathbf{b}$$

- ▶ onde M é o preconditionador

Escolha do preconditionador

- ▶ O mais próximo de A possível
 - ▶ Para $M^{-1}A$ resultar numa matriz bem condicionada
- ▶ Fácil de inverter

Escolhas extremas

- ▶ $M = A$, resultaria na identidade
- ▶ $M = I$, teria inversa trivial



Método Gradiente Conjugado

Algoritmo: Gradiente Conjugado com Precondicionador

$x_0 =$ estimativa inicial

$r_0 = b - Ax$

$d_0 = z_0 = M^{-1}r_0$

for $k = 0, 1, \dots, n - 1$ **do**

if $\|r_k\|_2 < tol$ **then**

stop

$$\alpha_k = \frac{r_k^T z_k}{d_k^T A d_k}$$

$$x_{k+1} = x_k + \alpha_k d_k$$

$$r_{k+1} = r_k - \alpha_k A d_k$$

$$z_{k+1} = M^{-1} r_{k+1}$$

$$\beta_k = r_{k+1}^T z_{k+1} / r_k^T z_k$$

$$d_{k+1} = z_{k+1} + \beta_k d_k$$



Método Gradiente Conjugado

Precondicionador de Jacobi

$$M = D$$

- Bom quando a matriz é estritamente diagonal dominante



Método Gradiente Conjugado

Precondicionador de Jacobi

$$M = D$$

- ▶ Bom quando a matriz é estritamente diagonal dominante

Precondicionador simétrico sucessivo com sobrerelaxação
(*Symmetrix Successive Over Relaxation* – SSOR)

$$M = (D + wL)D^{-1}(D + wU)$$

- ▶ com $w \in [0, 2]$
 - ▶ Se $w = 0$, tem-se Jacobi
 - ▶ Se $w = 1$, tem-se Gauss-Seidel



Método Gradiente Conjugado

Precondicionador de Jacobi

$$M = D$$

- ▶ Bom quando a matriz é estritamente diagonal dominante

Precondicionador simétrico sucessivo com sobrerelaxação
(*Symmetrix Successive Over Relaxation* – SSOR)

$$M = (D + wL)D^{-1}(D + wU)$$

- ▶ com $w \in [0, 2]$
 - ▶ Se $w = 0$, tem-se Jacobi
 - ▶ Se $w = 1$, tem-se Gauss-Seidel

Resolução de $M\mathbf{z} = \mathbf{r}$

- ▶ Precondicionador SSOR é um produto de matrizes triangulares

$$(I + wLD^{-1})(D + wU)$$

