

## Métodos Numéricos em ALgebra Linear (MAC0300)

---

**Por**

Caio Vinícius Dadauto 7994808

02/11/2014

As implementações deste trabalho foram executadas em uma máquina com as seguintes configurações:

<b>OS</b>	Arch Linux
<b>Kernel</b>	x86_64 Linux 4.1.5-1-ARCH
<b>CPU</b>	Intel Core i5-4200 CPU @ 2.6GHz
<b>RAM</b>	3862MiB

## 1 Gradientes Conjugados

Gradientes conjugados é um método iterativo para resolução de problemas lineares,

$$Ax = b \quad A \in \mathbb{R}^{n \times n} \quad x, b \in \mathbb{R}^n$$

onde  $A$  é simétrica e positiva definida. Este método é, normalmente, utilizado para resolver sistemas lineares que possuem dimensão grande o suficiente para tornar os métodos diretos impraticáveis devido a sua complexidade de  $\mathcal{O}(n^3)$ .

**Definicao 1.** *Seja a matriz  $A$  simétrica. Dois vetores  $a$  e  $b$  são ditos  $A$ -ortogonais se,*

$$\langle a, b \rangle_A := \langle Aa, b \rangle = \langle a, A^T b \rangle = a^T A b = 0$$

**Proposição 1.** *Seja  $A$  positiva definida e um conjunto  $d_0, \dots, d_n$  de  $n$  vetores não nulos  $A$ -ortogonal, ou seja,  $\langle d_i, d_j \rangle_A = 0$  para  $i \neq j \in 0, \dots, n$ . Então este conjunto é linearmente independente.*

**Prova 1.** *Seja um conjunto de coeficientes  $\alpha_i$ , tais que:*

$$\alpha_0 d_0 + \dots + \alpha_n d_n = 0$$

*então, multiplicando por  $A$  e  $d_i^T$  qualquer, tem-se que:*

$$\alpha_i d_i^T A d_i = 0$$

*pois, o conjunto dos vetores  $d$ 's é  $A$ -ortogonal. Como  $A$  é definida positiva, então  $d_i^T A d_i > 0$ . Assim,  $\alpha_i = 0$  para qualquer  $i$ , ou seja, o conjunto dos vetores  $A$ -ortogonal é linearmente independente.* ■

Dessa forma, voltando ao problema do sistema linear  $Ax = b$ , seja um conjunto  $\mathcal{B} = [d_0, \dots, d_{n-1}]$   $A$ -ortogonal e  $x^*$  a solução de  $Ax = b$ , temos que:

$$x^* = \alpha_0 d_0 + \dots + \alpha_{n-1} d_{n-1}$$

pois  $\mathcal{B}$  gera  $\mathbb{R}^n$ . Multiplicando por  $A$  e  $d_i^T$  qualquer, temos que:

$$\alpha_i = \frac{d_i^T A x^*}{d_i^T A d_i} = \frac{d_i^T b}{d_i^T A d_i} = \frac{\langle d_i, b \rangle}{\langle d_i, d_i \rangle_A}$$

dessa forma, tem-se ainda que:

$$x^* = \sum_{i=0}^{n-1} \frac{\langle d_i, b \rangle}{\langle d_i, d_i \rangle_A} d_i \quad (1)$$

Como o método dos gradientes conjugados é um método iterativo, é possível inferir que podemos gerar novos valores de  $x$  a partir das direções em  $\mathcal{B}$ , ou seja, suponha-se que:

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

essa inferência traz a tona o sentido iterativo do método. Porém, ainda devemos determinar como  $\alpha_k$  deve se comportar. A partir da igualdade (1) é possível supor que  $\alpha_k$  se comporte da seguinte maneira,

$$\alpha_k = \frac{\langle b - A x_k, d_k \rangle}{\langle d_k, d_k \rangle_A} = - \frac{\langle A x_k - b, d_k \rangle}{\langle d_k, d_k \rangle_A}$$

Pois, se considerarmos que  $\|e_k\| := \|x^* - x_k\| \geq \|x^* - x_{k+1}\| := \|e_{k+1}\|$  e que  $x^* = x_n$ , então

$$x^* = x_0 + \sum_{i=0}^{n-1} \frac{\langle A x_i - b, b \rangle}{\langle d_i, d_i \rangle_A} d_i$$

já que  $\alpha_n$  deve ser nulo. E ainda, essa igualdade é equivalente a (1), devido a  $\mathcal{B}$  ser uma base.

É importante notar que o método dos gradientes conjugados é equivalente ao problema,

$$\min f(x) = \frac{1}{2}x^T Ax - b^T x$$

pois, como essa função é convexa, basta verificar a condição de otimalidade de primeira ordem, ou seja,  $\nabla f = 0$  para determinar o ponto de mínimo de  $f(x)$ . O que é justamente resolver o problema  $Ax = b$ , já que  $\nabla f = Ax - b$ . Portanto, desta última observação, denotamos  $g_k = Ax_k - b$ , onde  $g$  é o gradiente de  $f$ , e então:

$$x^* = x_0 - \sum_{i=0}^{n-1} \frac{\langle g_i, b \rangle}{\langle d_i, d_i \rangle_A} d_i$$

**Teorema 1.** *Seja  $\mathcal{B}$  um conjunto de  $n$  vetores  $A$ -ortogonal. Para qualquer  $x_0$  a sequência  $x_k$  gerada por*

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

com

$$\alpha_k = -\frac{\langle g_k, d_k \rangle}{\langle d_k, d_k \rangle_A}; \quad g_k = Ax_k - b$$

converge para a única solução de  $Ax = b$ ,  $x^* = x_n$ .

**Prova 2.** *Da recorrência de  $x_k$  e do fato de que  $x^* = x_n$ , temos que:*

$$x^* - x_0 = \alpha_0 d_0 + \cdots + \alpha_{n-1} d_{n-1}$$

como sempre, multiplicando por  $A$  e fazendo o produto escalar por  $d_k$  qualquer, temos:

$$\alpha_k = \frac{\langle d_k, x^* - x_0 \rangle_A}{\langle d_k, d_k \rangle_A}$$

Dessa forma, basta provar que  $\alpha_k$  é da forma  $-\frac{\langle g_k, d_k \rangle}{\langle d_k, d_k \rangle_A}$ .

Para isso tomemos a seguinte igualdade para  $x_k$  qualquer:

$$x_k - x_0 = \alpha_0 d_0 + \cdots + \alpha_{k-1} d_{k-1}$$

de forma análoga, porém fazendo o produto escalar por  $d_k$ , tem-se que:

$$\langle d_k, x_k - x_0 \rangle_A = 0$$

Assim, temos que:

$$\begin{aligned}
\alpha_k &= \frac{\langle d_k, (x^* - x_k) + (x_k - x_0) \rangle_A}{\langle d_k, d_k \rangle_A} \\
&= \frac{\langle d_k, x^* - x_k \rangle_A + \langle d_k, x_k - x_0 \rangle_A}{\langle d_k, d_k \rangle_A} \\
&= \frac{\langle d_k, x^* - x_k \rangle_A}{\langle d_k, d_k \rangle_A} \\
&= \frac{d_k^T (Ax^* - Ax_k)}{\langle d_k, d_k \rangle_A} \\
&= \frac{d_k^T (b - Ax_k)}{\langle d_k, d_k \rangle_A} \\
&= -\frac{(Ax_k - b)^T d_k}{\langle d_k, d_k \rangle_A} \\
&= -\frac{g_k^T d_k}{\langle d_k, d_k \rangle_A} \\
\alpha_k &= -\frac{\langle g_k, d_k \rangle}{\langle d_k, d_k \rangle_A}
\end{aligned}$$

O que demonstra a convergência dados a recorrência e o  $\alpha$  apresentados pelo teorema. ■

## 1.1 Algoritmo

Do último Teorema da seção anterior, podemos inferir o seguinte algoritmo:

---

**Algorithm 1:** Algoritmo para o método dos gradientes conjugados.

---

```

 $x_k = 0, g_k = -b, d_k = b$ 
while  $\|g_k\| > \epsilon$  do
     $\alpha_k = -\frac{\langle g_k, d_k \rangle}{\langle d_k, d_k \rangle_A}$ 
     $x_k = x_k + \alpha_k d_k$ 
     $g_k = Ax_k - b$ 
     $\beta_k = \frac{\langle g_k, d_k \rangle_A}{\langle d_k, d_k \rangle_A}$ 
     $d_k = -g_k + \beta_k d_k$ 

```

---

Para verificar a corretude desse algoritmo, notemos o seguinte teorema.

**Teorema 2.** *O algoritmo é um método de gradientes conjugados, pois as seguintes afirmações são verdadeiras:*

- (i)  $g_0, g_1, \dots, g_k = g_0, Ag_0, \dots, A^k g_0$
- (ii)  $d_0, d_1, \dots, d_k = g_0, Ag_0, \dots, A^k g_0$
- (iii)  $d_k \perp A d_i$  para qualquer  $i < k$
- (iv)  $\alpha_k = \frac{\langle g_k, g_k \rangle}{\langle d_k, d_k \rangle_A}$
- (v)  $\beta_k = \frac{\langle g_{k+1}, g_{k+1} \rangle}{\langle g_k, g_k \rangle}$

Assim, a partir dos itens (iv) e (v) do teorema acima, é possível escrever outro algoritmo para o método dos gradientes conjugados com a vantagem deste ser menos custoso que o primeiro, como segue a baixo:

---

**Algorithm 2:** Algoritmo para o método dos gradientes conjugados otimizado.

---

```

 $x_k = 0, g_k = b, d_k = b$ 
while  $\|g_k\| > \epsilon$  do
     $g_{k-1} = g_k$ 
     $\alpha_k = \frac{\langle g_k, g_k \rangle}{\langle d_k, d_k \rangle_A}$ 
     $x_k = x_k + \alpha_k d_k$ 
     $g_k = g_k - \alpha_k A d_k$ 
     $\beta_k = \frac{\langle g_k, g_k \rangle}{\langle g_{k-1}, g_{k-1} \rangle}$ 
     $d_k = g_k + \beta_k d_{k-1}$ 

```

---

Apesar da privisão teórica impor que o método atinge  $x^*$  após  $n$  iterações, isso não é normalmente necessário para que o método atinga um valor satisfatório de  $x^*$ , ou seja,  $g_k < \epsilon$ , onde  $\epsilon$  é da ordem da precisão de máquina.

## 2 Implementação

Para a implementação do método, buscou-se uma forma mais adequada para o armazenamento da matriz esparsa  $A$  (caso onde o uso método iterativo faz sentido) de forma que o produto  $Ad$  fosse de complexidade  $\mathcal{O}(n)$ . Assim, a matriz esparsa  $A$  foi armazenada segundo o padrão *Yale*. Esse padrão consiste em armazenar três vetores distintos denominados  $A$ ,  $IA$  e  $JA$ . Em  $A$  são armazenados os valores não nulos da matriz esparsa e em  $JA$  são armazenados os índices das colunas onde os respectivos valores não nulos de  $A$  se encontram na matriz esparsa. Já os índices  $i$ 's do vetor  $IA$  representam os índices das linhas da matriz esparsa, enquanto que cada posição de  $IA$  armazena um índice do vetor  $A$ , onde esse índice representa o primeiro valor não nulo da linha  $i$  na matriz esparsa. A leitura da matriz esparsa para o armazenamento no padrão *Yale* é feita da esquerda para direita de cima para baixo, ou seja, ordenada por linha.

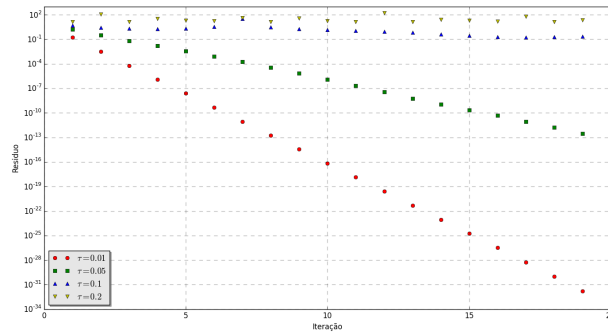
Para testar o algoritmo apresentado na seção anterior, implementou-se um gerador de matrizes esparsas simétricas e definidas positivas. Esse gerador foi desenvolvido seguindo o mesmo raciocínio empregado no exemplo da página 300 da referência [1], este raciocínio se baseia em criar uma matriz esparsa com diagonal principal unitária e com os demais valores fora da diagonal inicializados com um valor aleatório segundo uma distribuição uniforme em  $[-1, 1]$ , mantendo a simetria da matriz. Porém, para garantir a característica de esparsidade da matriz  $A$ , cada valor gerado nessa distribuição uniforme é comparado em módulo

com um valor  $\tau \in (0, 1)$ , se o valor gerado pela distribuição for maior que  $\tau$ , então a entrada na matriz será nula. Além disso, o vetor  $b$  do sistema  $Ax = b$  é gerado pela mesma distribuição uniforme.

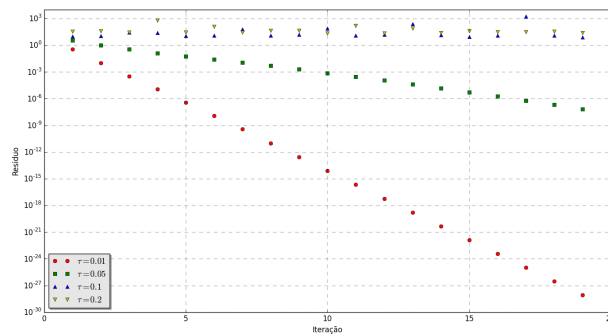
A implementação segue junto com esse relatório, o arquivo *gerador.c* contém o gerador de matriz esparsa segundo o que foi explicado acima. Já o arquivo *gradienteConjugado.c* contém a implementação do método explicado na seção anterior, bem como um conjunto de função auxiliares para tratar de operações com matrizes e vetores.

## 2.1 Exemplo

O exemplo da referência [1] é reproduzido a seguir. Foram geradas matriz esparsas para os valores 0.01, 0.05, 0.1 e 0.2 de  $\tau$ . As dimensões ( $n$ ) das matrizes utilizadas neste exemplo foram de 500 e 1000. Os gráficos a seguir apresentam o resultado obtido para as 20 primeiras iterações do método.



(a) Matriz de dimensão  $n = 500$ .



(b) Matriz de dimensão  $n = 1000$ .

Figura 1: Gráficos para o desenvolvimento do método durante as 20 primeiras iterações em função do módulo do resíduo.

## 2.2 Velocidade de Convergência

A medição de velocidade é algo complicado de se tratar neste caso, pois a esparsidade da matriz  $A$  depende do parâmetro  $\tau$  e o condicionamento da matriz  $A$  não é garantido, pois os valores da matriz são aleatoriamente gerados. Entretanto, buscou-se determinar sem muita precisão a complexidade do método. Para isso, gerou-se matrizes esparsas para  $\tau = 0.01$  de dimensões variadas, a saber 100, 500, 700, 1000, 1200, 1400, 1600, 2000, 5000, 6000, 8000 e 10000. Não foram tratados valores maiores do que estes pois o método para gerar estas matrizes passa a consumir muito tempo de processamento. A cada matriz gerada aplicou-se o método dos gradientes conjugados e registou-se o tempo para que o método atingisse um resíduo com módulo menor que  $10^{-19}$ . Assim, plotou-se os pontos gerados do tempo em função da dimensão  $n$  e ajustou-se aos pontos uma função da forma  $y = [0]x^{[1]}$ , obtendo um expoente de 1.89. O que implica em uma complexidade aproximadamente  $\mathcal{O}(n^2)$ . A figura abaixo apresenta o gráfico plotado.

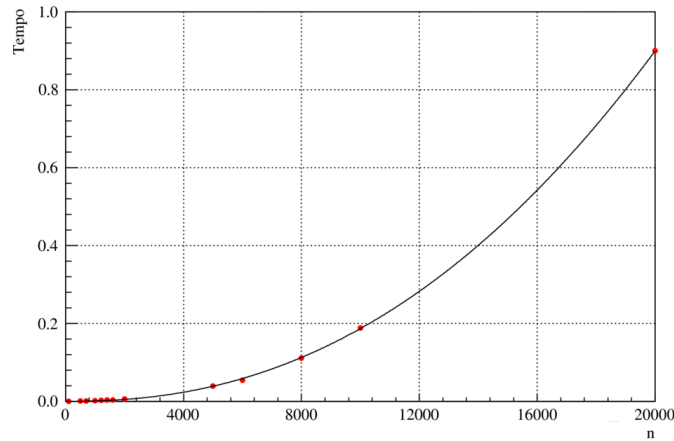


Figura 2: Estimativa da complexidade para o método iterativo.

## Referências

- [1] *Lloyd N. Trefethen, David Bau*, Numerical Linear Algebra, Society for Industrial and Applied Mathematics, Philadelphia, 1997.

- [2] *Luenberger D.G., Ye Y.*, Linear and nonlinear programming-Springer, 2008.