

Computação científica

Métodos Iterativos Não Estacionários

Lucia Catabriga

LCAD - Laboratório de Computação de Alto Desempenho
Departamento de Informática - CT/UFES





Referências

www.inf.ufes.br/~luciac

- Barret, R, et al., ``Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods'', SIAM, 1994.
- Shewchuk, J. R., ``An Introduction to the Conjugate Gradient Method Withuot the Agonizing Pain'', 1994.
- Dongarra, J.J., Duff, I.S., Sorasen, D.C., Van der Vorst, H.A., Numerical LInear Algebra for High-Performance Computers, SIAM, 1998.
- Golub, G. and Van Loan, C., "Matrix Computations", The John Hopkins University Press, 1993.
- Kelley C.T., "Iterative Methods for Linear and Nonlinear Equations", SIAM, 1995.
- Saad, Y., "Iterative Methods for Sparse Linear Systems", PWS Publishing Company, 1996.
- White, R.E., "Computational Modeling with Methods and Analysis", 2003.



UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
DOCERE OMNES GENTES

Métodos baseados nos Espaços de Krylov

- Introdução
- Método dos Gradientes Conjugados (CG)
- Método do Resíduo Mínimo Generalizado (GMRES)
- Método das Direções Conjugadas à Esquerda (LCD)

Introdução

- **Características:**

- São métodos diretos na sua essência, mas na prática são usados com métodos iterativos. Portanto a solução aproximada depende de uma tolerância pré-fixada.
- A matriz dos coeficientes A e o vetor dos termos independentes b não são alterados durante o processo iterativo
- depende de critérios de convergência relacionados a matriz dos coeficientes A .

- **Objetivo:**

- transformar o sistema $Ax=b$ em um problema de minimização do resíduo $||r_k|| = \min ||b-Ax_k||$

- **Complexidade:**

- em torno de n^2 por iteração.

Introdução

- Os Métodos baseados nos espaços de Krylov representam a classe de métodos mais usadas em Dinâmica dos Fluidos Computacionais
- Idéias Gerais:
 - $x_k = x_0 + y_k$ é uma solução aproximada de $Ax = b$, onde:
 - $y_k \in K_m = \text{span}[r_0, Ar_0, A^2r_0, \dots, A^{k-1}r_0]$
 - $r_0 = b - Ax_0$
- Principais Operações:
 - Produto Matriz-vetor: $y = Ax$ (matvec)
 - Produto Escalar $r = y \cdot x$ (dot)
 - Adição de uma constante vezes um vetor com um vetor $y = y + ax$ (SAXPY)

Método dos Gradientes Conjugados

- A simétrica, definida positiva:
 - $a_{ij}=a_{ji}$ e para todo $v \in \mathbb{R}^n$ $v^T A v > 0$ e $v=0$, $v^T A v = 0$
- $Ax=b \rightarrow$ Encontrar o único ponto de mínimo de uma "função erro"
- x solução aproximada
- $r = b - Ax$ resíduo da aproximação x
- **Função erro:** $f(x) = \frac{1}{2}r^T A^{-1}r$
 - Para todo r , $f \geq 0$ e se x sol. exata $f(x)=0$
 - $f(x) = \frac{1}{2}(b-Ax)^T A^{-1}(b-Ax)$
 - $f(x) = \frac{1}{2}x^T A x - b^T x + \frac{1}{2}b^T A^{-1}b$

Método dos Gradientes Conjugados

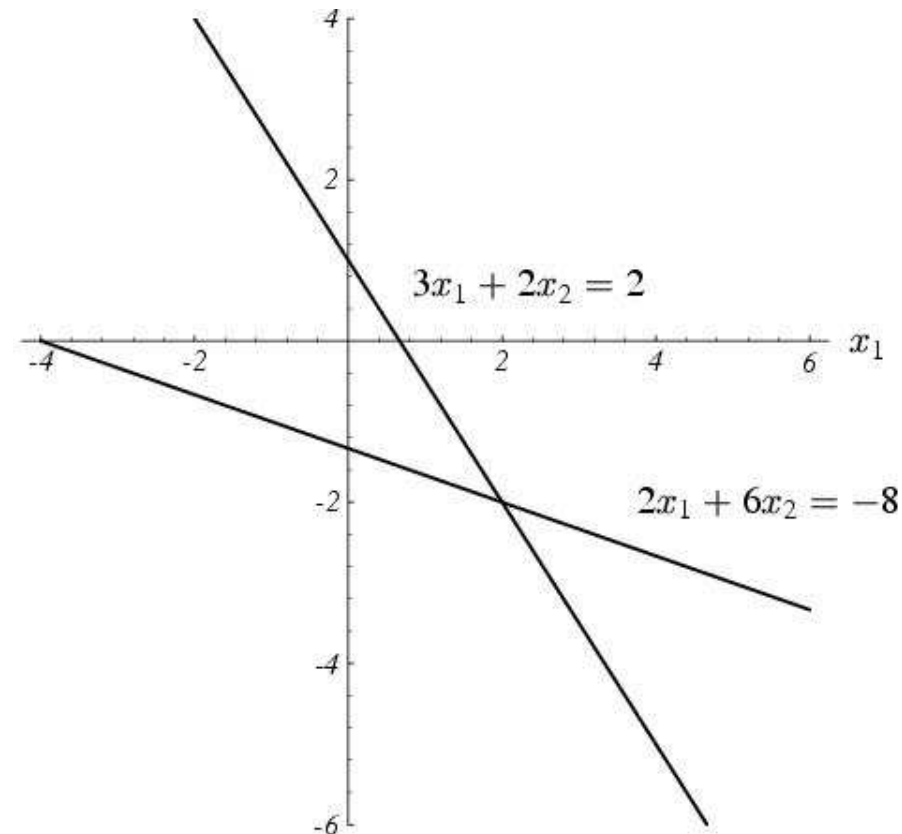
$$x_e = \min_{x \in \mathbb{R}^n} f(x) \quad \text{tq}$$

$$Ax_e = b$$

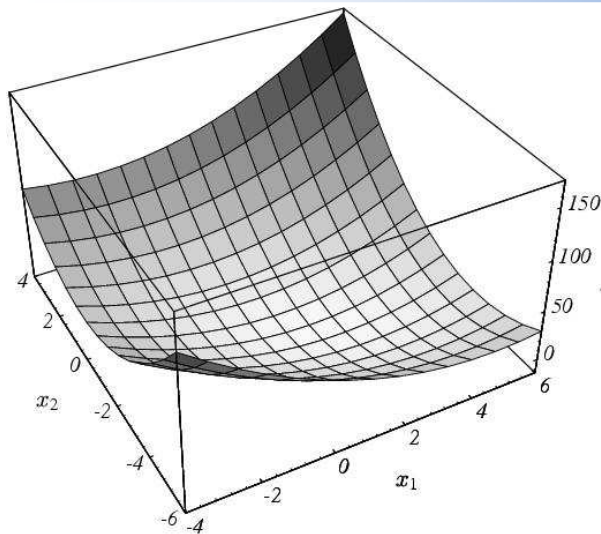
$$f(x) = \frac{1}{2} x^t A x - b^t x + \frac{1}{2} b^t A^{-1} b$$

Exemplo:

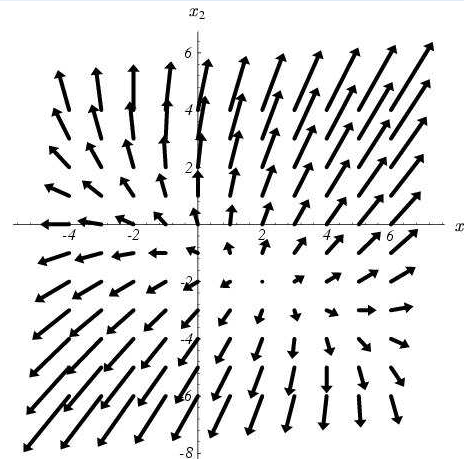
$$\begin{cases} 3x_1 + 2x_2 = 2 \\ 2x_1 + 6x_2 = -8 \end{cases} \Rightarrow A = \begin{bmatrix} 3 & 2 \\ 2 & 6 \end{bmatrix} \quad b = \begin{bmatrix} 2 \\ -8 \end{bmatrix} \quad x = \begin{bmatrix} 2 \\ -2 \end{bmatrix}$$



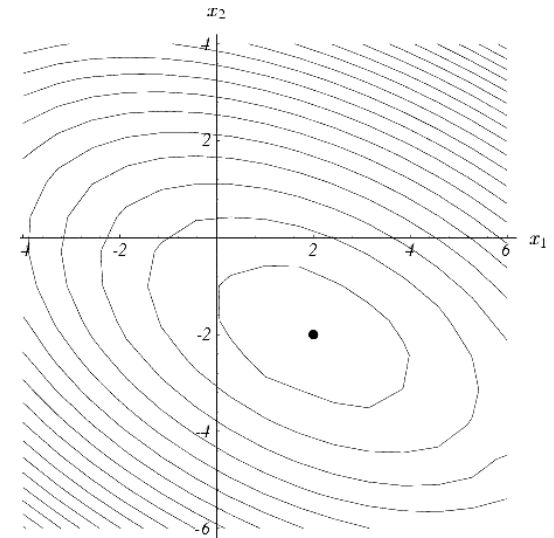
Método dos Gradientes Conjugados



$f(x)$



$f'(x)$



Ponto de mínimo de $f(x)$

Solução $Ax=b$

$$\nabla f(x) = Ax - b = -r$$



r representa a direção de maior descida

Método da Menor Descida ou Método dos Gradientes

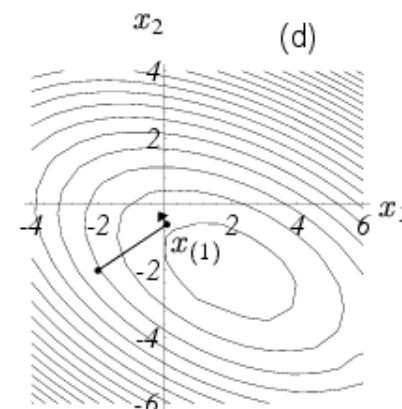
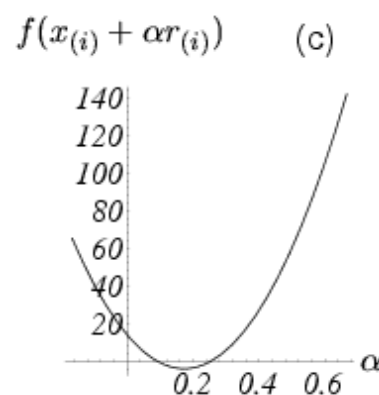
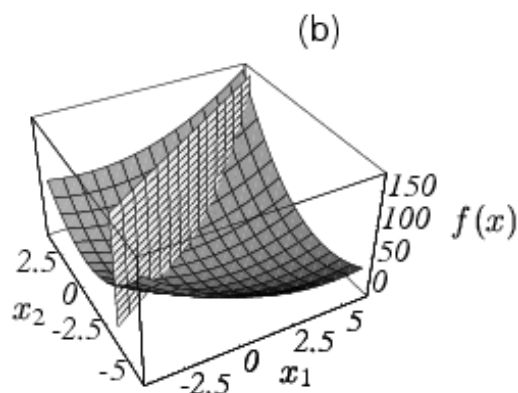
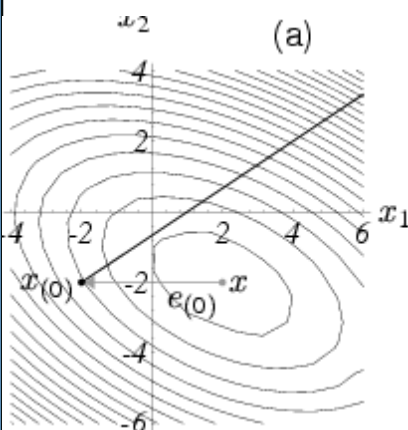
$$x_1 = x_0 + \alpha r_0 \quad (\text{mesma direção de } r_0)$$

$$f(x_1) = \frac{1}{2} (x_0 + \alpha r_0)^t A (x_0 + \alpha r_0) - b^t (x_0 + \alpha r_0) + \frac{1}{2} b^t A^{-1} b$$

$$\frac{df(x_1)}{d\alpha} = 0 \quad (\text{ponto de mínimo})$$

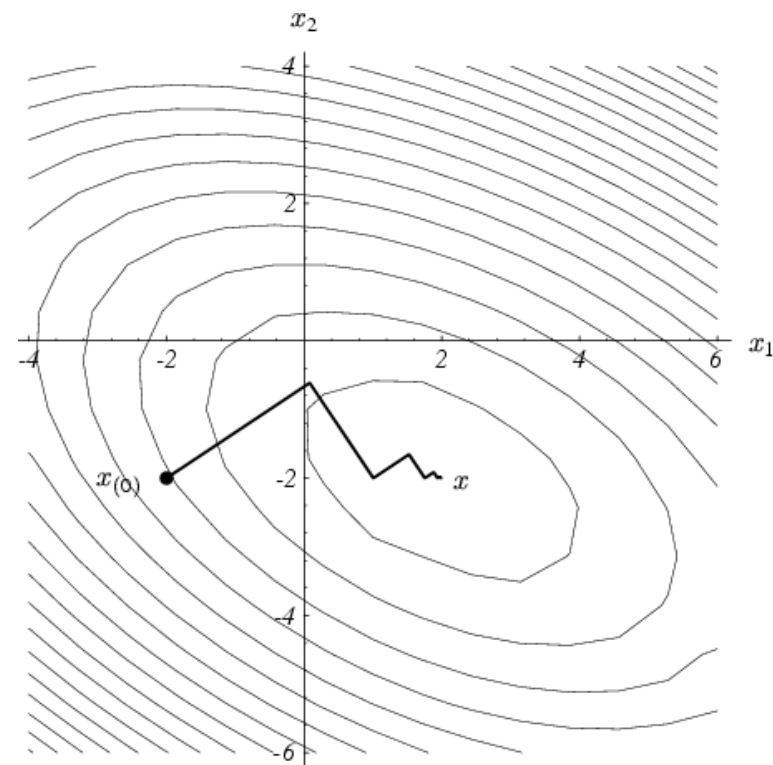
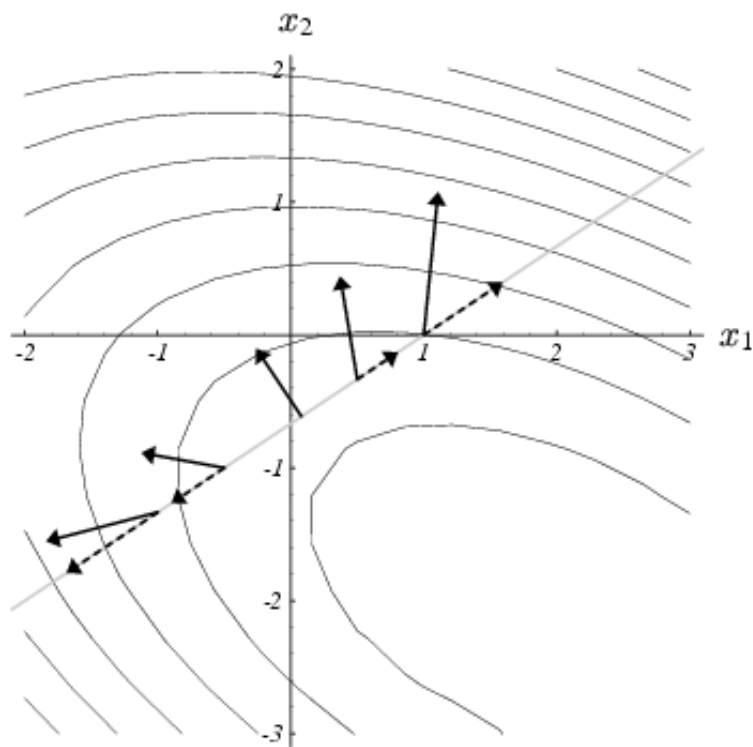
$$\frac{df(x_1)}{d\alpha} = \nabla f(x_1) \cdot \frac{dx_1}{d\alpha} = \nabla f(x_1) \cdot r_0 = 0$$

$$(r_1)^t r_0 = 0$$



- (a) $x_0 = (-2, -2)^t$ calculando o próximo passo na direção de maior descida
- (b) encontrar o ponto que intercepta as duas superfícies que minimiza f
- (c) Parábola que intercepta as duas superfícies
- (d) r_1 (Gradiente em x_1) é ortogonal a r_0

Método da Menor Descida ou Método dos Gradientes



Detalhes do Processo iterativo

Método da Menor Descida ou Método dos Gradientes

Cálculo de α :

$$r_{i+1}^t r_i = 0$$

$$(b - Ax_{i+1})^t r_i = 0$$

$$(b - A(x_i + \alpha r_i))^t r_i = 0$$

$$r_i^t r_i - \alpha r_i^t A r_i = 0$$

$$\alpha = \frac{r_i^t r_i}{r_i^t A r_i}$$

Atualização do Resíduo:

$$x_{i+1} = x_i + \alpha r_i$$

$$b - Ax_{i+1} = b - A(x_i + \alpha r_i)$$

$$r_{i+1} = r_i - \alpha A r_i$$

Algoritmo do Método dos
Gradientes:

Dados $x_0 = 0$, A , b , $Nmax$, tol conhecidos

$$r_0 = b$$

Repita para $i = 0, 1, 2, \dots$

$$v = A r_i$$

$$\alpha = \frac{r_i^t r_i}{r_i^t A r_i}$$

$$x_{i+1} = x_i + \alpha r_i$$

$$r_{i+1} = r_i - \alpha A r_i$$

Até $r_i^t r_i < tol$ ou $i \geq Nmax$

Método Gradientes Conjugados

- É possível escolher n direções ortogonais e linearmente independentes $v_0, v_1, \dots, v_{n-1}$ e, por meio da minimização da função $f(x_k + \alpha v_k)$, em cada uma das direções separadamente, construir uma sequência de aproximações que forneça o mínimo de $f(x)$ após n passos.

- Por indução:

$$\nabla f(x_{j+1}) \cdot v^{(i)} = 0 \text{ para } i = 0, 1, 2, \dots, j \quad (r_{j+1} \perp v_i)$$

- Assim, quando $j=n-1$:

$$\nabla f(x_n) \cdot v_i = 0 \text{ para } i = 0, 1, 2, \dots, n-1$$

- Portanto x_n é o mínimo de $f(x)$

Método Gradientes Conjugados

- Encontrar o conjunto de vetores LI $v_0, v_1, \dots, v_{n-1}$.
- Em cada passo j , encontrar α que minimiza $f(x_j + \alpha v_j)$

$$x_{j+1} = x_j + \alpha v_j$$

- As direções v_j são LI e ortogonais com relação a A ($v_j^t A v_i = 0$ para $i \neq j$, $v_j^t r_j = 0$ para todo j)
- Pode-se mostrar existe uma combinação linear entre v_j e r_j :

$$v_{j+1} = r_j + \beta v_j$$



$$v_{j+1}^t r_j = r_j^t r_j = \|r_j\|_2^2 \Rightarrow \alpha = \frac{r_j^t r_j}{v_j^t A v_j}$$
$$r_{j+1} = r_j - \alpha A v_{j+1}$$

Método Gradientes Conjugados

$$v_{j+1}^t A v_j = 0$$

$$v_{j+1} = r_j + \beta v_j$$

$$\Downarrow$$

$$\beta = -\frac{r_j^t A v_j}{v_j^t A v_j}$$

$$\Downarrow$$

$$\beta = \frac{r_j^t r_j}{r_{j-1}^t r_{j-1}} = \frac{\|r_{j-1}\|_2^2}{\|r_{j-1}\|_2^2}$$

Algoritmo do CG (Mét. Direto)

$$v_0 = r_0 = b - A x_0$$

Para $j = 0, n-1$ faça

$$\alpha = \frac{\|r_j\|_2^2}{v_j^t A v_j}$$

$$x_{j+1} = x_j + \alpha v_j$$

$$r_{j+1} = r_j - \alpha A v_j$$

$$\beta = \frac{\|r_{j+1}\|_2^2}{\|r_j\|_2^2}$$

$$v_{j+1} = r_{j+1} + \beta v_j$$

Fim

x_n solução exata

Método Gradientes Conjugados

Exemplo

Considere o sistema $Ax=b$ onde:

$$\begin{bmatrix} 4 & 1 & & & \\ 1 & 4 & 1 & & \\ & & \ddots & & \\ & & & 1 & 4 & 1 \\ & & & & 1 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} 5 \\ 6 \\ \vdots \\ 6 \\ 5 \end{bmatrix} \Rightarrow x = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \\ 1 \end{bmatrix} \rightarrow$$

- Para um código em Octave (ou MatLab) a solução perde precisão para valores de $n \geq 250$

Critério de Parada

$$\|r_j\|_2 \leq tol \quad \text{ou} \quad \frac{\|r_j\|_2}{\|b\|_2} \leq tol$$

$\Downarrow$

$$\|r_j\|_2^2 \leq tol^2 \|b\|_2^2$$

Método Gradientes Conjugados

Algoritmo Iterativo do CG

Dados $x_0 = 0$, A, b , $Nmax$ e tol

$$v_0 = r_0 = b$$

$$\delta_{novo} = \|r_0\|_2^2, \delta_0 = \delta_{novo}, j = 0$$

Enquanto $j < Nmax$ e $\delta_{novo} > tol^2 \delta_0$ faça

$$p_j = Av_j$$

$$\alpha = \frac{\delta_{novo}}{v_j^t p_j}$$

$$x_{j+1} = x_j + \alpha v_j$$

$$r_{j+1} = r_j - \alpha p_j$$

$$\delta_{velho} = \delta_{novo}$$

$$\delta_{novo} = \|r_{j+1}\|_2^2$$

$$\beta = \frac{\delta_{novo}}{\delta_{velho}}$$

$$v_{j+1} = r_{j+1} + \beta v_j$$

$$j = j + 1$$

Fim

- Número de operações:

- 1 matvec
- 3 saxpy
- 2 dot

(complexidade $\approx n^2$)

- Armazenamento:

- 3 vetores
- 1 Matriz

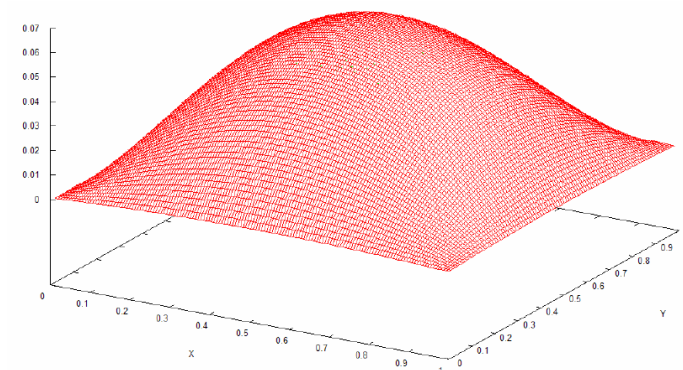
Exemplo Numérico (SOR x CG)

$$\frac{\partial u}{\partial t} - \left(\frac{\partial^2 u}{\partial^2 x} + \frac{\partial^2 u}{\partial^2 y} \right) = f(x, y, t) \quad (x, y) \in (0, 1) \times (0, 1), t \in (0, 1)$$

$$u|_{\Gamma} = 0 \quad (\text{condição de contorno})$$

$$u(x, y, 0) = 0 \quad (\text{condição inicial})$$

$$\text{Supor } f(x, y, t) \text{ tq } u_e(x, y, t) = t^2 x(x-1)y(y-1)$$



Considerando MDF no espaço e no tempo :

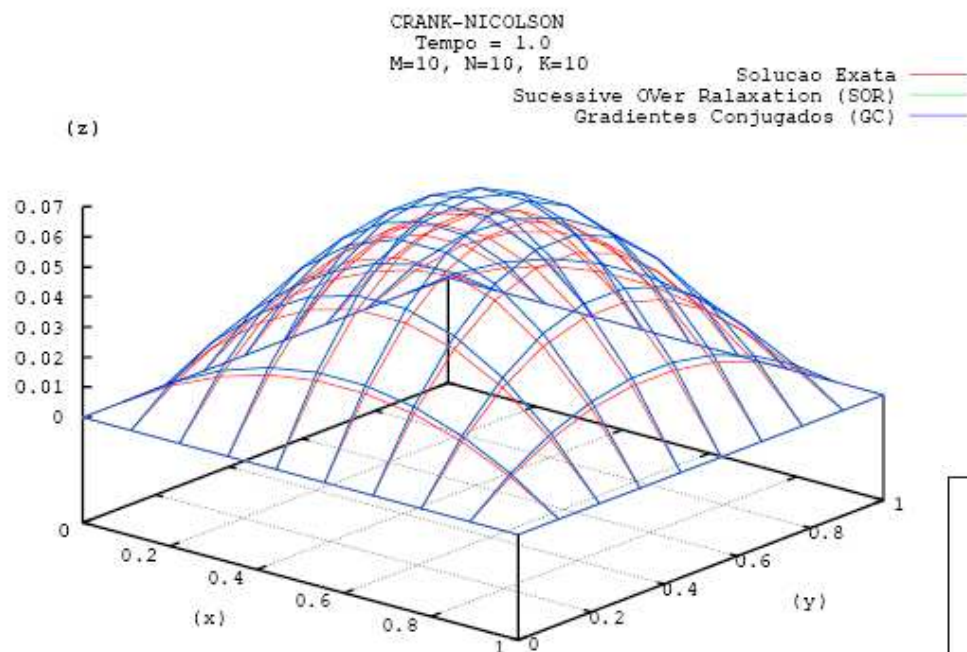
$$Au^{k+1} = Bu^k + F \quad k = 0, 1, 2, \dots$$

A : matriz pentadiagonal (CDS)

Fonte: Relatório do aluno Vicente Bissoli Sessa (Métodos Numéricos I - 2006/2)



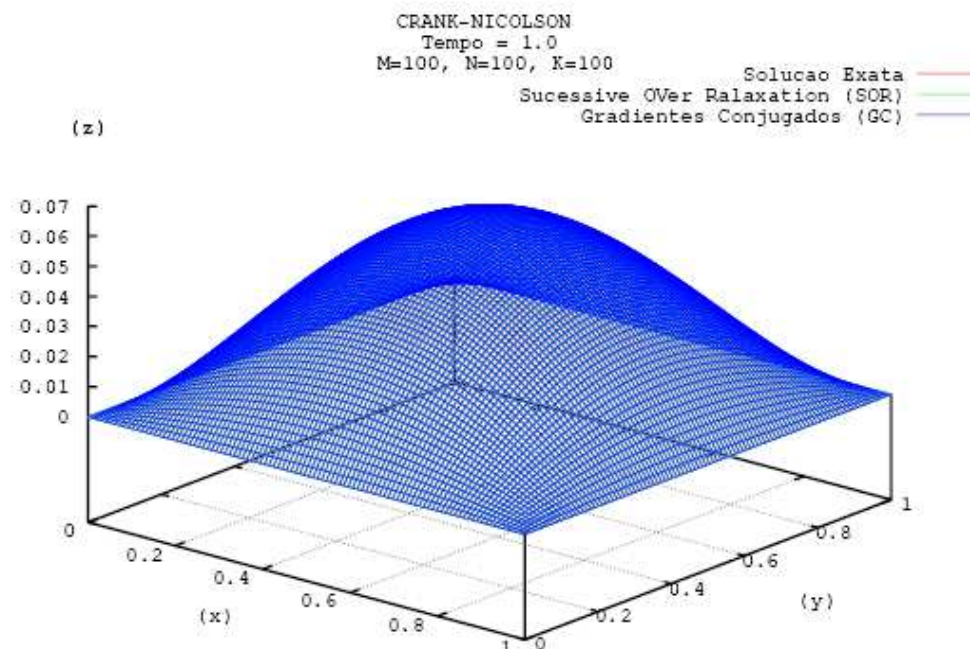
Exemplo Numérico (SOR x CG)



$N=10, M=10, k=10$

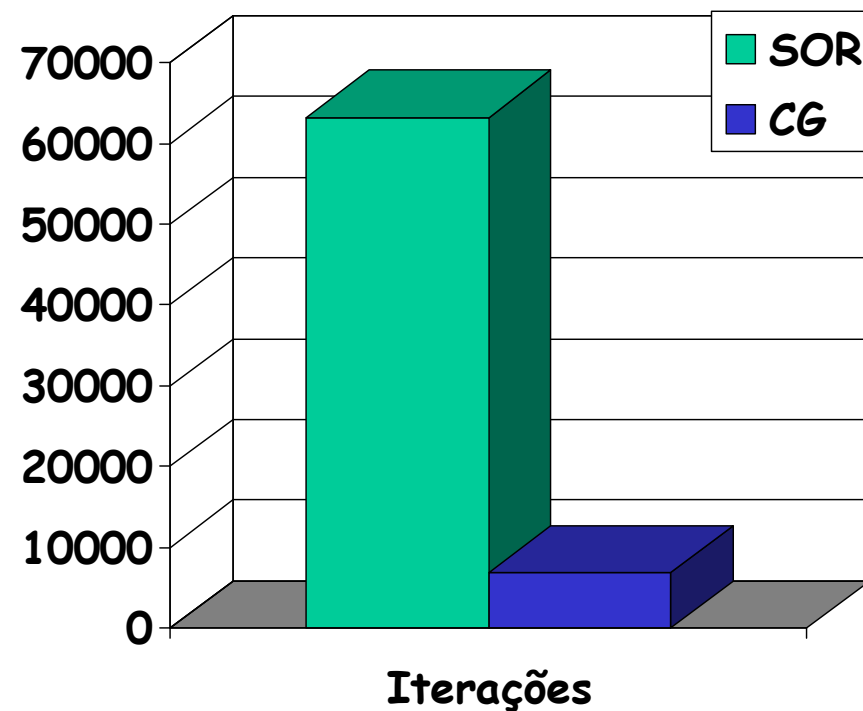
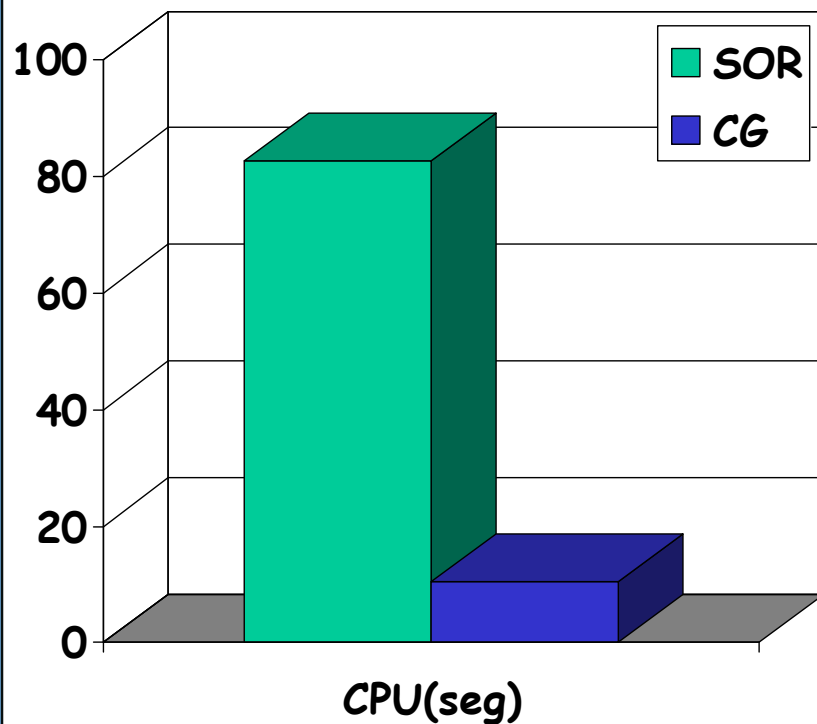
$$W_{otimo}=1.3$$

$$\text{Tolerância} = 10^{-7}$$



$N=100, M=100, k=100$

Exemplo Numérico (SOR x CG)



$N=100, M=100, k=100$

Métodos de Projeção (Breve Definição)

Definições:

$$x_{i+1} = x_i + s_i \quad \text{Solução aprox. } Ax = b$$

$$\mathcal{K}^m = \text{span}\{v_0, v_1, \dots, v_{m-1}\} \quad \text{Espaço de busca}$$

$$\mathcal{L}^m = \text{span}\{w_0, w_1, \dots, w_{m-1}\} \quad \text{Espaço de restrição}$$

$$V = [v_0, v_1, \dots, v_{m-1}] \quad \text{base de } \mathcal{K}^m$$

$$s_i \in \mathcal{K}^m \Rightarrow s_i = [v_0, v_1, \dots, v_{m-1}] \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_{m-1} \end{bmatrix} \quad \text{onde } y \in \mathbb{R}^m$$

$$x_{i+1} = x_i + Vy \Rightarrow y \in \mathcal{L}^m$$

- A escolha do vetor y define as restrições impostas para avançar no processo iterativo

Métodos de Projeção (Breve Definição)

Espaços de Krylov

$$\mathcal{K}^m(A, v) = \text{span}\{v, Av, A^2v, \dots, A^{m-1}v\}$$

$$x_m = x_0 + q_{m-1}(A)v$$

$$q_{m-1}(A) = c_0 + c_1A + c_1A^2 + \dots + c_{m-1}A^{m-1}$$

$$q_{m-1}(A)v = (c_0 + c_1A + c_1A^2 + \dots + c_{m-1}A^{m-1})v$$

$$= c_0v + c_1Av + c_1A^2v + \dots + c_{m-1}A^{m-1}v$$

$$= \begin{bmatrix} v, Av, A^2v, \dots, A^{m-1}v \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_{m-1} \end{bmatrix} = VC \Rightarrow x_m = x_0 + VC \Rightarrow c \in \mathcal{L}^m$$

- Os métodos baseados nos espaços de Krylov se diferenciam pela escolha do espaço de restrições

Métodos de Projeção (Breve Definição)

- Métodos de Projeção Ortogonal $\mathcal{L}^m = K^m$
 - Matriz dos coeficientes **simétrica**
 - Exemplo:
 - Método dos Gradientes Conjugados
- Métodos de Projeção Oblíqua $\mathcal{L}^m = AK^m$
 - Matriz dos coeficientes **não-simétrica**
 - Exemplos:
 - Método do Resíduo Mínimo Generalizado (GMRES)
 - Método dos gradientes bi-conjugados estabilizados (Bi-CGSTAB)
 - Método das direções conjugadas à esquerda (LCD)

Método do Resíduo Mínimo Generalizado (GMRES)

- Proposto por Yousef Saad (1986)
- Objetivo:
 - Minimizar a norma do resíduo do sistema $x = \min ||b - Ax_k||$. Considerando x_0 solução inicial uma solução aproximada x_k é dada por:

$$x_k = x_0 + z$$

onde z é um vetor do espaço de Krylov

$$K^k = \text{span}\{r_0, Ar_0, A^2r_0, \dots, A^{k-1}r_0\}$$

$$\min_{x \in x_0 + K^k} \|b - Ax\|_2$$

GMRES – Montagem da base ortonormal de K^k

Processo de Gram-Schmidt

$$u_1 = \frac{r_0}{\|r_0\|}$$

Para $i = 1, \dots, k$ faça

$$\tilde{u}_{i+1} = Au_i$$

Para $j = 1, \dots, i$ faça

$$\beta_{i+1,j} = \tilde{u}_{i+1}^t u_j$$

$$\tilde{u}_{i+1} = \tilde{u}_{i+1} - \beta_{i+1,j} u_j$$

Fim j

$$u_{i+1} = \frac{\tilde{u}_{i+1}}{\|\tilde{u}_{i+1}\|}$$

Fim i

$$U_k = [u_1, u_2, \dots, u_k]$$

$$AU_k = U_{k+1} H_k$$

$$H_k = \begin{bmatrix} \beta_{21} & \beta_{31} & \cdots & \beta_{k1} & \beta_{k+1,1} \\ \|\tilde{u}_2\| & \beta_{32} & \cdots & \beta_{k2} & \beta_{k+1,2} \\ & \|\tilde{u}_3\| & \cdots & \beta_{k3} & \beta_{k+1,3} \\ & & \ddots & \vdots & \vdots \\ & & & \beta_{k,k-1} & \beta_{k+1,k-1} \\ & & & \|\tilde{u}_k\| & \beta_{k+1,k} \\ & & & & \|\tilde{u}_{k+1}\| \end{bmatrix}$$

H_k : matriz de Hessemberg superior
ordem $k \times (k + 1)$

GMRES – O problema de Minimização em K^k

$$z = \sum_{j=1}^k y_j u_j \quad \text{para } y \in \mathbb{R}^k \quad (z \in K^k) \quad \|b - A(x_0 + z)\| = \left\| b - Ax_0 - A \sum_{j=1}^k y_j u_j \right\|$$

$$e = \{\|r_0\|, 0, \dots, 0\}^t \quad \text{com } (k+1) \text{ termos}$$

$$r_0 = U_{k+1} e \quad \text{pois}$$

$$u_1 = \frac{r_0}{\|r_0\|} \text{ e } U_{k+1} \text{ é uma base de } K^{k+1}$$

$$= \|r_0 - AU_k y\|$$

$$= \|U_{k+1} e - U_{k+1} H_k y\|$$

$$= \|U_{k+1} (e - H_k y)\|$$

$$\|U_{k+1}\| = 1 \Rightarrow \|b - Ax\| = \|e - H_k y\|$$

↓

$$\min_{z \in K^k} \|b - A(x_0 + z)\| = \min_{y \in \mathbb{R}^k} \|e - H_k y\|$$

O problema de minimização em K^k foi transformado em um problema de minimização em $\mathbb{R}^k$.

GMRES – Algoritmo Inicial

Dados $x_0, A, b, Kmax, tol$

$$r = b - Ax_0$$

$$\rho = \|r\|_2 \quad u_1 = r/\rho \quad k = 0$$

Enquanto $\rho > tol \|b\|_2$ e $k < Kmax$ faça

$$k = k + 1$$

$$\tilde{u}_{k+1} = Au_k$$

Para $j = 1, \dots, k$ faça

$$h_{jk} = \tilde{u}_{k+1}^t u_j$$

$$\tilde{u}_{k+1} = \tilde{u}_{k+1} - h_{jk} u_j$$

Fim j

$$h_{k+1,k} = \|\tilde{u}_{k+1}\|_2$$

$$u_{k+1} = \tilde{u}_{k+1} / \|\tilde{u}_{k+1}\|_2$$

$$e = (\|r\|, 0, \dots, 0)^t$$

$$\min_{y \in \mathbb{R}^k} \|e - H_k y\|_2$$

$$\rho = \|e - H_k y\|_2$$

Fim k

$$x_k = x_0 + U_k y$$

$$\min_{y \in \mathbb{R}^k} \|e - H_k y\|_2$$



$H_k y = e$, onde :

$$H_k = \begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1,k-1} & h_{1k} \\ \|\tilde{u}_2\| & h_{22} & \cdots & h_{2,k-1} & h_{2k} \\ & \|\tilde{u}_3\| & \cdots & h_{3,k-1} & h_{3,k} \\ & & \ddots & \vdots & \vdots \\ & & & h_{k-1,k-1} & h_{k-1,k} \\ & & & \|\tilde{u}_k\| & h_{kk} \\ & & & & \|\tilde{u}_{k+1}\| \end{bmatrix} \quad e = \begin{bmatrix} r \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \end{bmatrix}$$



$\bar{H}_k y = \bar{e}$, onde :

$$\bar{H}_k = \begin{bmatrix} \bar{h}_{11} & \bar{h}_{12} & \cdots & \bar{h}_{1k} \\ & \bar{h}_{21} & \cdots & \bar{h}_{2k} \\ & & \ddots & \vdots \\ & & & \bar{h}_{kk} \end{bmatrix} \quad \bar{e} = \begin{bmatrix} \bar{e}_1 \\ \bar{e}_2 \\ \vdots \\ \bar{e}_k \end{bmatrix}$$

GMRES – Cálculo de $\hat{H}_k$

Algoritmo QR – Rotação de Givens

$$\bar{H}_k = G_k G_{k-1} \cdots G_2 G_1 H_k$$

$$\bar{e} = G_k G_{k-1} \cdots G_2 G_1 e$$

$$G_j = \begin{bmatrix} 1 & \cdots & \overset{j}{0} & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & c & s & \cdots & 0 \\ 0 & \cdots & -s & c & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 1 \end{bmatrix}^j$$

$$H_k = \begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1,k-1} & h_{1k} \\ \boxed{h_{21}} & h_{22} & \cdots & h_{2,k-1} & h_{2k} \\ & \boxed{h_{32}} & \cdots & h_{3,k-1} & h_{3k} \\ & & \ddots & \vdots & \vdots \\ & & & h_{k-1,k-1} & h_{k-1,k} \\ & & & \boxed{h_{k,k-1}} & h_{kk} \\ & & & & \boxed{h_{k+1,k}} \end{bmatrix}$$

$$H_k^2 = G_1 H_k, \text{ onde:}$$

$$G_1 = G_1(c_1, s_1)$$

$$r_1 = \sqrt{h_{11}^2 + h_{21}^2}$$

$$c_1 = h_{11}/r_1 \quad s_1 = h_{21}/r_1$$

$$G_1 H_k = \begin{bmatrix} c_1 & s_1 & & & \\ -s_1 & c_1 & & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & 1 \end{bmatrix} H_k = H_k^2$$

$$H_k^2 = \begin{bmatrix} \bar{h}_{11} & \bar{h}_{12} & \cdots & \bar{h}_{1k} \\ & \boxed{\bar{h}_{22}} & \cdots & \bar{h}_{2k} \\ & & \boxed{\bar{h}_{32}} & \cdots & h_{3k} \\ & & & \ddots & \vdots \\ & & & & h_{k+1,k} \end{bmatrix}$$

Algoritmo GMRES

Dados $A, b, lmax, tol, x = 0$

$\varepsilon = tol \|b\|_2 \quad i = 1$

$u_i = b - Ax \quad e_i = \|u_i\| \quad u_i = u_i / e_i \quad \rho = e_i$

Enquanto $\rho > \varepsilon$ e $i < lmax$ faça

$u_{i+1} = Au_i$

(Ortogonalização de Gram – Schmidt)

Para $j = 1, \dots, i$ faça

$h_{ji} = u_{i+1}^t u_j$

$u_{i+1} = u_{i+1} - h_{ji} u_j$

Fim j

$h_{i+1,i} = \|u_{i+1}\|_2$

$u_{i+1} = u_{i+1} / h_{i+1,i}$

(Algoritmo QR)

Para $j = 1, \dots, i-1$

$h_{ji} = c_j h_{ji} + s_j h_{j+1,i}$

$h_{j+1,i} = -s_j h_{ji} + c_j h_{j+1,i}$

Fim j

$r = \sqrt{h_{ii}^2 + h_{i+1,i}^2}$

$c_i = h_{ii} / r$

$s_i = h_{i+1,i} / r$

$h_{ii} = r \quad (h_{i+1,i} = 0.0)$

$e_{i+1} = -s_i e_i$

$e_i = c_i e_i$

$\rho = |e_{i+1}|$

$i = i + 1$

Fim i

$i = i - 1$

Para $j = i, \dots, 1$

$y_j = \frac{e_j - \sum_{l=j+1}^i h_{jl} y_l}{h_{jj}}$

Fim j

$x = \sum_{j=1}^i y_j u_j$

i-ésima iteração:

• n° de operações:

- 1 matvec
- i saxpy
- i dot

• Armazenamento:

- i vetores
- 1 Matriz A
- 1 Matrix H_k $(i+1) \times i$

• Complexidade:

- $O(i n) + O(i^3) +$
 $O(\text{matrix-vetor}) \approx$
 $O(n^2)$

GMRES - Restart

- Cada iteração do Método GMRES é muito cara.
- Para torná-lo viável computacionalmente é adotado o processo de reinicializar a formação dos vetores da base a cada k iterações.
- Supor $k=20$:

Dados A, b, x_0

$$V = [u_1, u_2, \dots, u_{20}]$$

$$y = (y_1, y_2, \dots, y_{20})$$

$$x_{20} = x_0 + \sum_{j=1}^{20} y_j u_j$$

Dados A, b, x_{20}

$$V = [u_1, u_2, \dots, u_{20}]$$

$$y = (y_1, y_2, \dots, y_{20})$$

$$x_{40} = x_{20} + \sum_{j=1}^{20} y_j u_j$$

$\vdots$

Algoritmo GMRES com Restart

Dados $A, b, lmax, tol, x = 0, k$

$\varepsilon = tol \|b\|_2 \quad l = 1$

Repita

$i = 1 \quad u_i = b - Ax \quad e_i = \|u_i\| \quad u_i = u_i / e_i \quad \rho = e_i$

Enquanto $\rho > \varepsilon$ e $i < k$ faça

$u_{i+1} = Au_i$

(Ortogonalização de Gram – Schmidt)

Para $j = 1, \dots, i$ faça

$h_{ji} = u_{i+1}^t u_j$

$u_{i+1} = u_{i+1} - h_{ji} u_j$

Fim j

$h_{i+1,i} = \|u_{i+1}\|_2$

$u_{i+1} = u_{i+1} / h_{i+1,i}$

(Algoritmo QR)

Para $j = 1, \dots, i - 1$

$h_{ji} = c_j h_{ji} + s_j h_{j+1,i}$

$h_{j+1,i} = -s_j h_{ji} + c_j h_{j+1,i}$

Fim j

$r = \sqrt{h_{ii}^2 + h_{i+1,i}^2}$

$c_i = h_{ii} / r$

$s_i = h_{i+1,i} / r$

$h_{ii} = r \quad (h_{i+1,i} = 0.0)$

$e_{i+1} = -s_i e_i$

$e_i = c_i e_i$

$\rho = |e_{i+1}|$

$i = i + 1$

Fim i

$i = i - 1$

Para $j = i, \dots, 1$

$y_j = \frac{e_j - \sum_{l=j+1}^i h_{jl} y_l}{h_{jj}}$

Fim j

$x = \sum_{j=1}^i y_j u_j$

$l = l + 1$

Até $\rho < \varepsilon$ e $l \geq lmax$

i -ésima iteração:

• nº de operações:

- 1 matvec
- k saxpy
- k dot

• Armazenamento:

- k vetores
- 1 Mat A
- 1 Mat H_k $(K+1) \times K$

• Complexidade:

- $O(kn) + O(k^3) +$
 $O(\text{matrix-vetor}) \approx$
 $O(n^2)$

Método das Direções Conjugadas à Esquerda (LCD)

- Inicialmente apresentado por **Yuan, Golub, Plemmons** and **Cecílio** (2002→2004).
- **Catabriga, Coutinho** and **Franca** (2004) introduziram restart apresentando comparações com GMRES, Bi-CGSTAB e TFQMR para formulações de elementos finitos.
- **Dai** and **Yuan** (2004) introduziram um novo algoritmo (limitação de memória e 1 produto matriz-vetor) usando formulações de diferenças finitas.
- **Catabriga, Valli, Melotti, Pessoa** and **Coutinho** (2006) avaliaram o novo algoritmo para equações lineares e não lineares usando o método de Newton inexato.

LCD - Introdução

- Definição: Os vetores $p_1, p_2, \dots, p_n \in \mathbb{R}^n$ são chamados vetores de direções conjugadas à esquerda (LCD) de uma matriz real não singular A se:

$$\begin{cases} p_i^t A p_j = 0 & \text{para } i < j \\ p_i^t A p_j \neq 0 & \text{para } i = j \end{cases}$$

isto é,

$$P^t A P = L \quad (\text{matriz triangular inferior})$$

onde $P = [p_1, p_2, \dots, p_n]$

- P é Linearmente Independente

- $x \in \mathbb{R}^n \Rightarrow x = x_0 + \sum_{i=1}^n \alpha_i p_i$

LCD – Descrição do Método

- x^* solução de $Ax=b$ e $P = [p_1, p_2, \dots, p_n]$ vetores de direções conjugadas à esquerda,

$$x \in \mathbb{R}^n \Rightarrow x = x_0 + \sum_{i=1}^n \alpha_i p_i$$

- seja x_0 aproximação inicial

- $Ax^* = b$

$$x^* = x_0 + \sum_{i=1}^n \alpha_i p_i$$

$$r = b - Ax^*$$

$$r = b - Ax_0 - \sum_{i=1}^n \alpha_i Ap_i$$

$$A \left(x_0 + \sum_{i=1}^n \alpha_i p_i \right) = b$$

$$r = r_0 - \sum_{i=1}^n \alpha_i Ap_i$$

$$Ax^* = Ax_0 + \sum_{i=1}^n \alpha_i Ap_i$$

Como calcular α_i ?

LCD – Descrição do Método

- x^* solução exata $r=0$, portanto r é ortogonal a todo p_i :

$$p_i^t r = 0 \quad \forall i = 1, \dots, n$$

$$p_i^t \left(r_0 - \sum_{j=1}^n \alpha_j A p_j \right) = 0$$

$$p_i^t (r_0 - \alpha_1 A p_1 - \alpha_2 A p_2 - \dots - \alpha_i A p_i - \dots - \alpha_n A p_n) = 0$$

$$p_i^t \left(r_0 - \underbrace{\sum_{j=1}^{i-1} \alpha_j A p_j}_{r_{i-1}} - \alpha_i A p_i - \alpha_{i+1} A p_{i+1} - \dots - \alpha_n A p_n \right) = 0$$

$$p_i^t r_{i-1} - \alpha_i p_i^t A p_i - \underbrace{\alpha_{i+1} p_i^t A p_{i+1}}_0 - \dots - \underbrace{\alpha_n p_i^t A p_n}_0 = 0$$

$$\alpha_i = \frac{p_i^t r_{i-1}}{p_i^t A p_i} \Rightarrow x_i = x_0 + \sum_{k=1}^i \alpha_k p_k \Rightarrow x^* = x_0 + \sum_{k=1}^n \alpha_k p_k$$

LCD – Algoritmo inicial

Dados $A, b, x_0, \varepsilon, Kmax$

"Escolha" $\{p_1, p_2, \dots, p_n\}$

$$r_0 = b - Ax_0$$

$$\rho = \|r_0\|_2 \quad k = 1$$

Enquanto $\rho > \|b\|_2$ e $k < Kmax$ faça

$$\alpha_k = \frac{p_k^t r_{k-1}}{p_k^t A p_k}$$

$$x_k = x_{k-1} + \alpha_k p_k$$

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

$$\rho = \|r_k\|_2$$

$$k = k + 1$$

Fim k

- Como obter o conjunto de direções conjugadas à esquerda?

LCD – Cálculo do Conjunto $\{p_i\}$

- É possível demonstrar a existência de uma relação de recorrência entre $\{p_1, p_2, \dots, p_k\}$ e r_k para calcular p_{k+1} :

$$p_{k+1} = r_k + \sum_{i=1}^k \beta_i p_i$$

- Sabendo que

$$p_j^t A p_{k+1} = 0 \quad \forall \quad j = 1, \dots, k$$

$$p_{k+1} = r_k + \beta_1 p_1 + \dots + \beta_k p_k$$

$$p_1^t A p_{k+1} = p_1^t A r_k + \beta_1 p_1^t A p_1 + \beta_2 \underbrace{p_1^t A p_2}_{=0} + \dots + \beta_k \underbrace{p_1^t A p_k}_{=0} = 0 \Rightarrow \beta_1 p_1^t A p_1 = -p_1^t A r_k$$

$$p_2^t A p_{k+1} = p_2^t A r_k + \beta_1 p_2^t A p_1 + \beta_2 p_2^t A p_2 + \dots + \beta_k \underbrace{p_2^t A p_k}_{=0} = 0 \Rightarrow \beta_1 p_2^t A p_1 + \beta_2 p_2^t A p_2 = -p_2^t A r_k$$

$\vdots$

$$p_k^t A p_{k+1} = p_k^t A r_k + \beta_1 p_k^t A p_1 + \dots + \beta_k p_k^t A p_k = 0 \Rightarrow \beta_1 p_k^t A p_1 + \dots + \beta_k p_k^t A p_k = -p_k^t A r_k$$

LCD – Cálculo do Conjunto $\{p_i\}$

Sistema Triangular Inferior

$$\begin{bmatrix} p_1^t A p_1 & & & \\ p_2^t A p_1 & p_2^t A p_2 & & \\ \vdots & \vdots & \ddots & \\ p_k^t A p_1 & p_k^t A p_2 & \cdots & p_k^t A p_k \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_k \end{bmatrix} = - \begin{bmatrix} p_1^t A r_k \\ p_2^t A r_k \\ \vdots \\ p_k^t A r_k \end{bmatrix}$$



Supor $\{p_1, p_2, \dots, p_k\}$ e inicializando $p_{k+1} = r_k$

Para $i = 1, \dots, k$ faça

$$\beta_i = - \frac{p_i^t A p_{k+1}}{p_i^t A p_i}$$

$$p_{k+1} = p_{k+1} + \beta_i p_i$$

Fim i

LCD – Algoritmo (versão p)

Dados $A, b, x_0, \varepsilon, Kmax$

$$r_0 = b - Ax_0 \quad \rho = \|r_0\|_2$$

Escolha p_1 tq $p_1^t A p_1 \neq 0$

$k = 1$

Enquanto $\rho > \varepsilon \|b\|_2$ e $k < Kmax$ faça

$$\alpha_k = \frac{p_k^t r_{k-1}}{p_k^t A p_k}$$

$$x_k = x_{k-1} + \alpha_k p_k$$

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

$$p_{k+1} = r_k$$

Para $i = 1, \dots, k$ faça

$$\beta_i = -\frac{p_i^t A p_{k+1}}{p_i^t A p_i}$$

$$p_{k+1} = p_{k+1} + \beta_i p_i$$

Fim i

$$\rho = \|r_k\|_2$$

$$k = k + 1$$

Fim k

- Em cada iteração k são realizados **$k+1$ produtos matriz-vetor.**

Algoritmo Ineficiente!

Supor

$$q_k = A p_k$$

$$\alpha_k = \frac{p_k^t r_{k-1}}{p_k^t q_k}$$

Inicilizando $q_{k+1} = A p_{k+1}$, no loop i

$$\beta_i = -\frac{p_i^t q_{k+1}}{p_i^t q_i}$$

$$p_{k+1} = p_{k+1} + \beta_i p_i$$

$$A p_{k+1} = A p_{k+1} + \beta_i A p_i$$

$$q_{k+1} = q_{k+1} + \beta_i q_i$$

LCD – Algoritmo (versão pq)

Dados $A, b, x_0, \varepsilon, Kmax$

$$r_0 = b - Ax_0 \quad \rho = \|r_0\|_2$$

Escolha p_1 tq $p_1^t A p_1 \neq 0$

$$q_1 = A p_1 \quad k = 1$$

Enquanto $\rho > \varepsilon \|b\|_2$ e $k < Kmax$ faça

$$\alpha_k = \frac{p_k^t r_{k-1}}{p_k^t q_k}$$

$$x_k = x_{k-1} + \alpha_k p_k$$

$$r_k = r_{k-1} - \alpha_k q_k$$

$$p_{k+1} = r_k \quad q_{k+1} = A p_{k+1}$$

Para $i = 1, \dots, k$ faça

$$\beta_i = -\frac{p_i^t q_{k+1}}{p_i^t q_i}$$

$$p_{k+1} = p_{k+1} + \beta_i p_i$$

$$q_{k+1} = q_{k+1} + \beta_i q_i$$

Fim i

$$\rho = \|r_k\|_2$$

$$k = k + 1$$

Fim k

k-ésima iteração:

- n° de operações:

- 1 matvec
- K+2 saxpy
- k+2 dot

- Armazenamento:

- 2k vetores
- 1 Matriz A

- Complexidade:

- $O(k n) + O(k^2) +$
 $O(\text{matrix-vetor}) + \dots \approx$
 $O(n^2)$

LCD – Algoritmo (versão pq com restart)

Dados $A, b, x_0, \varepsilon, lmax, k$

$$r = b - Ax \quad \rho = \|r\|_2$$

Escolha p_1 tq $p_1^t A p_1 \neq 0$ ($p_1 = r$)

$l = 1$

Repita

$$q_l = A p_l \quad i = 1$$

Enquanto $\rho > \varepsilon \|b\|_2$ e $i < k$ faça

$$\alpha_i = \frac{p_i^t r}{p_i^t q_i}$$

$$x = x + \alpha_i p_i$$

$$r = r - \alpha_i q_i$$

$$p_{i+1} = r \quad q_{k+1} = A p_{k+1}$$

Para $j = 1, \dots, i$ faça

$$\beta_j = -\frac{p_j^t q_{i+1}}{p_j^t q_j}$$

$$p_{i+1} = p_{i+1} + \beta_j p_j$$

$$q_{i+1} = q_{i+1} + \beta_j q_j$$

Fim j

$$\rho = \|r\|_2$$

$$i = i + 1$$

Fim i

$$p_1 = r \quad (\text{ou } p_1 = p_{i+1})$$

$$l = l + 1$$

até $\rho \leq \varepsilon \|b\|_2$ ou $l > lmax$

- cada iteração:

• n° de operações:

- 1 matvec
- $K+2$ saxpy
- $k+2$ dot

• Armazenamento:

- $2k$ vetores
- 1 Matriz A

• Complexidade:

- $O((k+2)n) + O(k^2) + O(\text{matrix-vector}) + \dots \approx O(n^2)$

GMRESxLCD

Por iteração	matvec	saxpy	dot	memory
GMRES(k)	1	k	k	K vet 1 Mat $H(k+1) \times k$
LCD(k)	1	k+2	k+2	2k vet 1 Mat

Experimentos Numéricos (GMRESxLCD)

Equação Convecção-Difusão

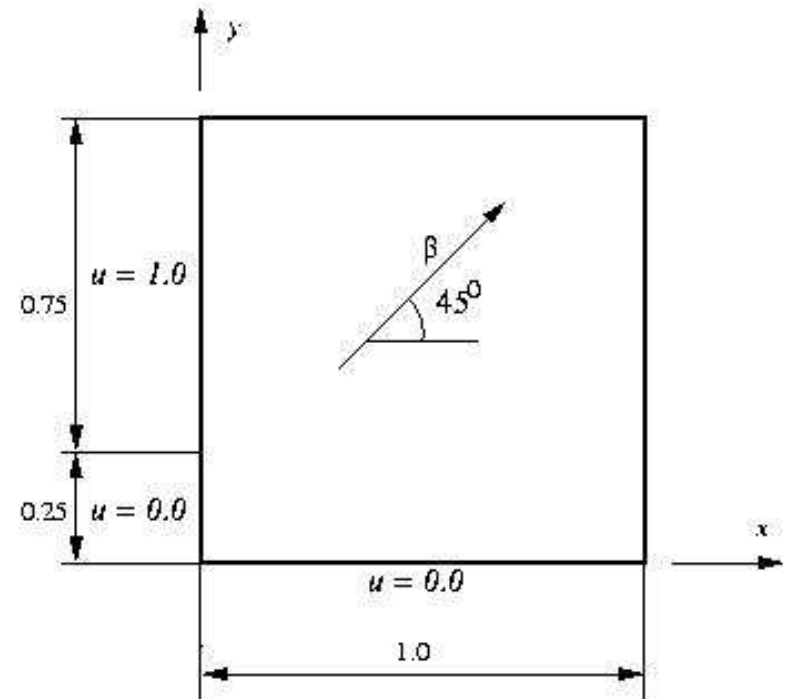
$$\beta \cdot \nabla u - \nabla \cdot (k \nabla u) = f$$

$$\beta = \begin{bmatrix} \beta_x \\ \beta_y \end{bmatrix} \quad k = \begin{bmatrix} k_x & 0 \\ 0 & k_y \end{bmatrix}$$

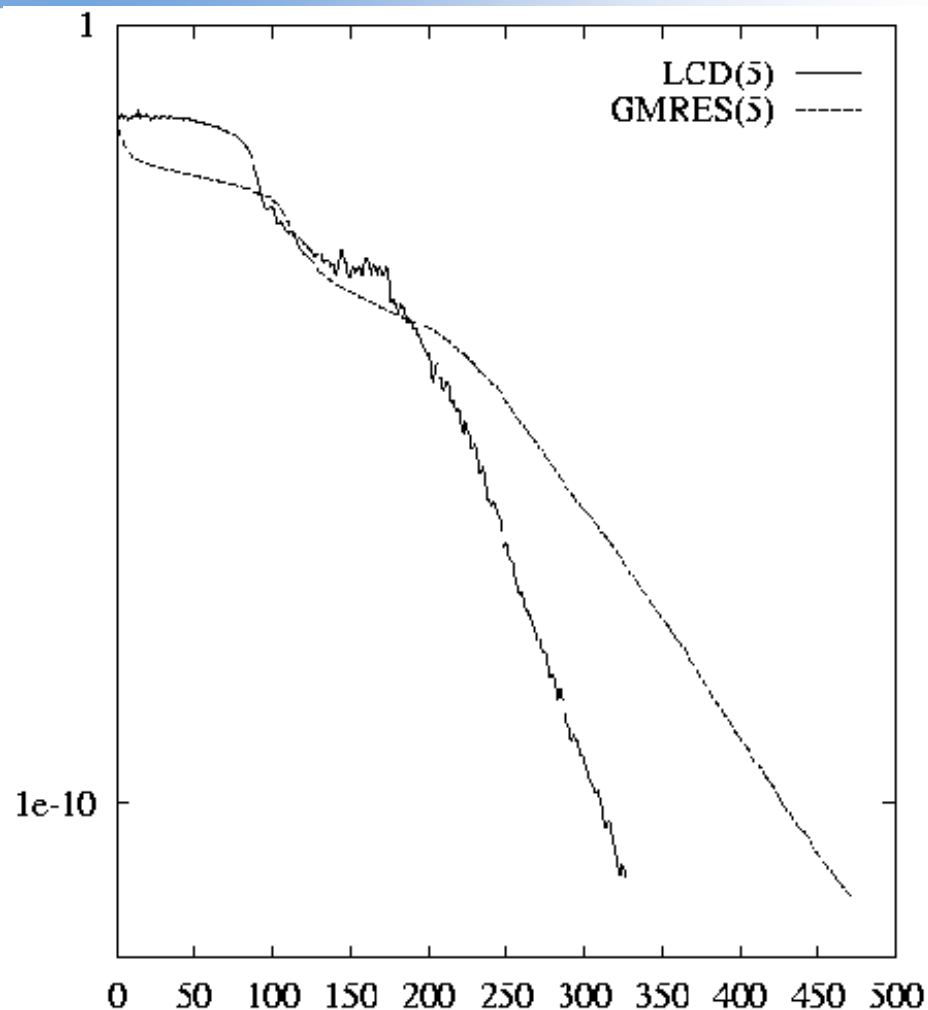
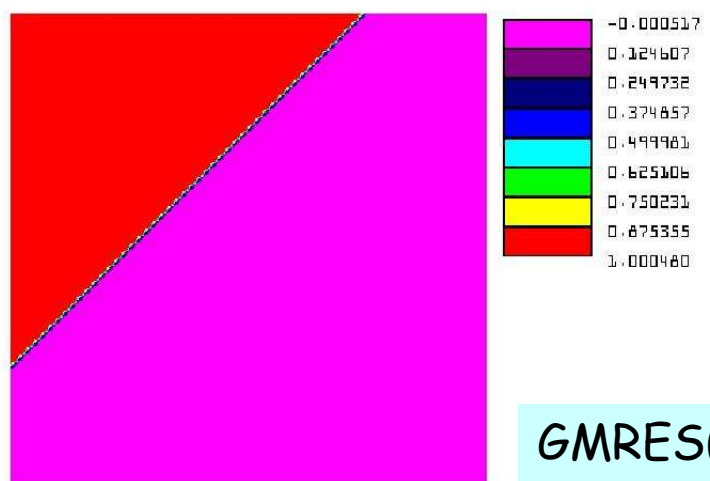
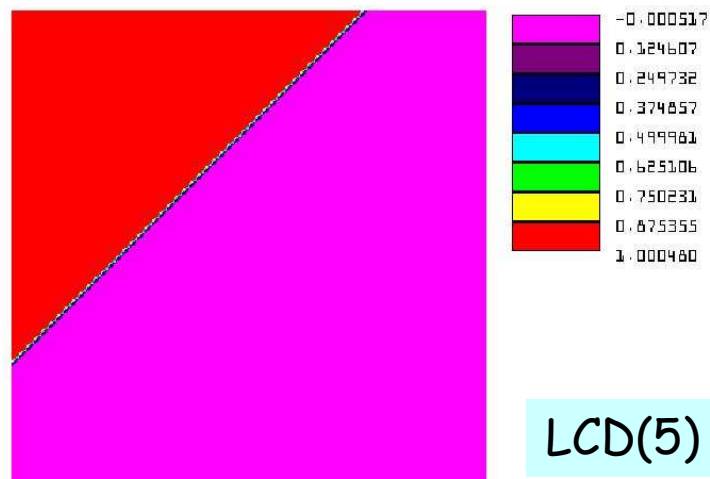
Formulação
Elementos Finitos
EBE

$$Kv = F$$

- $k = 10^{(-7)}$
- $||\beta|| = 1$
- $64 \times 64, 128 \times 128, 256 \times 256$ e 512×512 células considerando 4 triângulos em cada célula



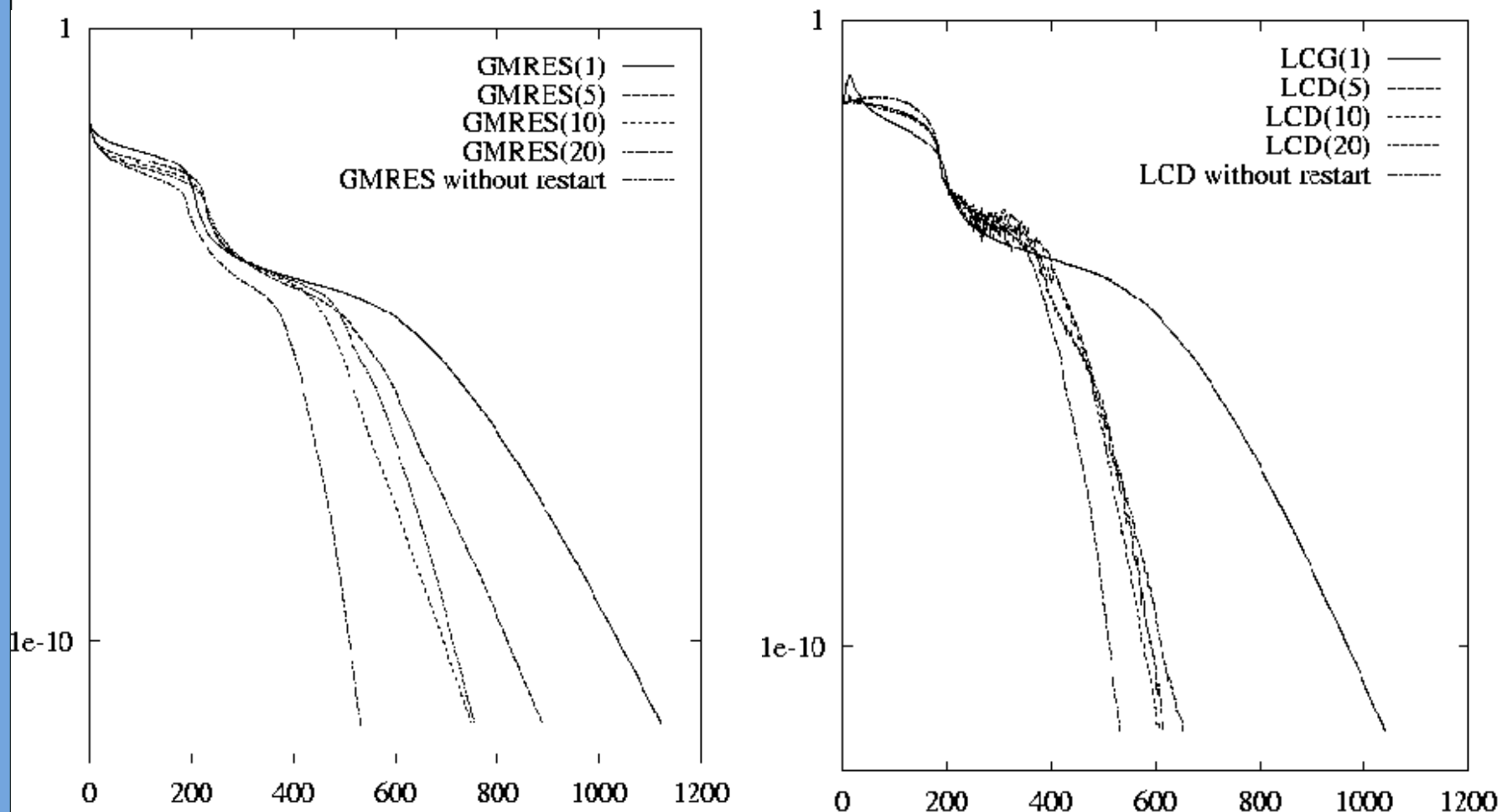
Experimentos Numéricos (GMRESxLCD)



Resíduo Relativo - malha 64x64

Fonte: Catabriga, Coutinho, Franca (IJNME, 2004)

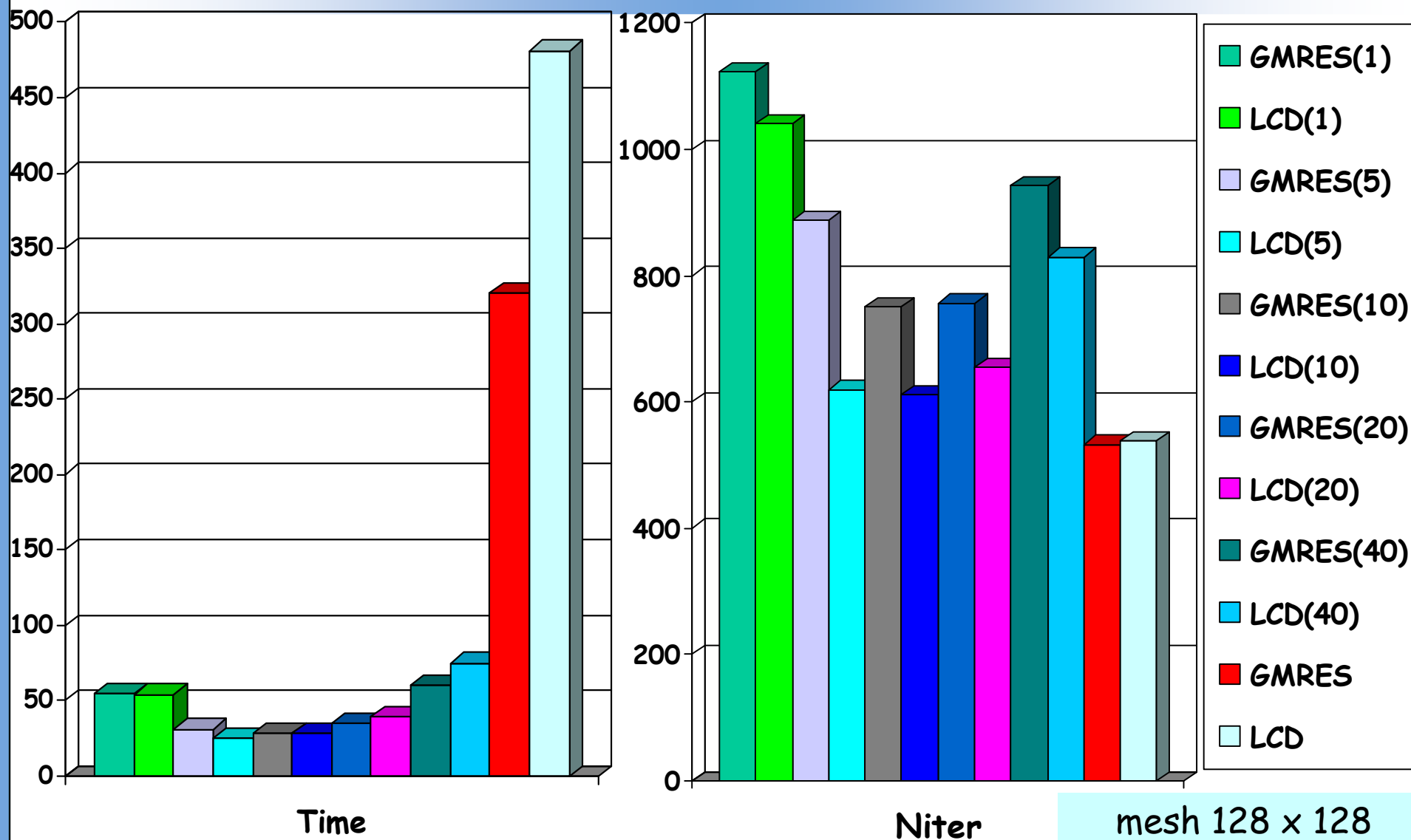
Experimentos Numéricos (GMRESxLCD)



Relative residual - mesh 128 x 128

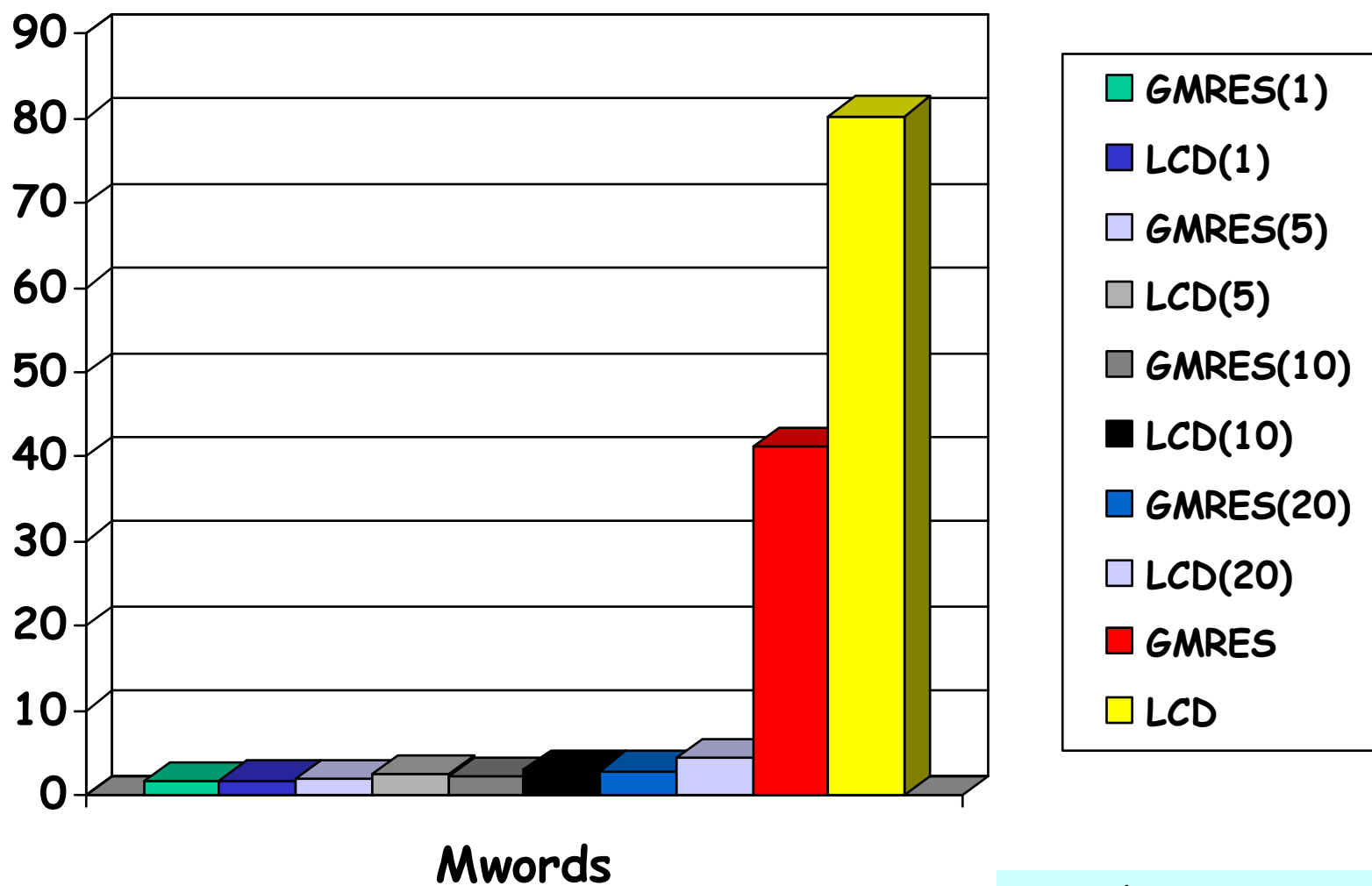
Fonte: Catabriga, Coutinho, Franca (CMAME, 2004)

Experimentos Numéricos (GMRESxLCD)



Fonte: Catabriga, Valli, Melotti, Pessoa and Coutinho, CNME, 2006

Experimentos Numéricos (GMRESxLCD)



mesh 128 x 128

Fonte: Catabriga, Coutinho, Franca (CMAME, 2004)