

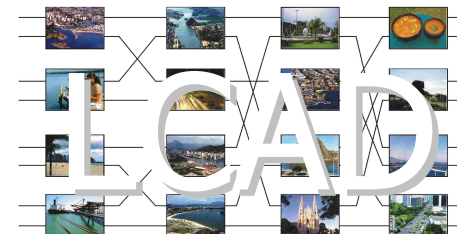


# LNCC - Programa de Verão 2008

## Minicurso M16 Estrutura de Dados e Solvers

Lucia Catabriga

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



# Ementa do Curso

- Introdução
- Estudo de Armazenamento de Matrizes Esparsas:
  - armazenamentos globais e armazenamentos locais aplicados ao método das diferenças finitas e ao método dos elementos finitos;
- Métodos de solução de sistemas lineares: Diretos, iterativos;
- Métodos Iterativos: Estacionários, Não-estacionários;
- Métodos Não-estacionários:
  - Método dos Gradientes Conjugados;
  - Método do Resíduo Mínimo Generalizado (GMRES);
  - Métodos das direções conjugadas à esquerda (LCD);
- Estudo de Precondicionadores:
  - Fatorações tipo Gauss-Seidel e fatorações tipo ILU.
- Solução de sistemas não-lineares:
  - Método de Newton inexato;
  - Critérios de parada.



# Precondicionamento de Sistemas Lineares

- Técnica para acelerar métodos existentes
- Idéia principal é solucionar:

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

- $M$  aproxima  $M^{-1}$  e é escolhida de tal forma que:
  - $M$  é simples de se implementar
  - $M$  diminui sensivelmente o número de iterações
- Formas possíveis
  - Direita:  $AM^{-1}y = b$
  - Esquerda:  $M^{-1}Ax = M^{-1}b$
  - Esquerda-Direita:  $M^{-1}AN^{-1}x = M^{-1}b$

# Precondicionamento do Método dos Gradientes Conjugados

- $M^{-1}A$  deve ser simétrica definida positiva
- Condição: existir  $E$  tq  $M=EE^t$  ( $E$  pode ser obtida pela fatoração de Cholesky)  $\rightarrow M$  simétrica definida positiva:

$$\underbrace{E^{-1}AE^{-t}}_{\tilde{A}} \underbrace{E^t x}_{\tilde{x}} = \underbrace{E^{-1}b}_{\tilde{b}}$$

- $E^{-1}AE^{-t}$  é simétrica definida positiva
- Observando que:

$$\tilde{r}_i = E^{-1}r$$

$$\tilde{v}_i = E^t v$$

$$\tilde{x}_i = E^t x$$

Não é necessário fatorar  $M$ !

$$E^{-t}E^{-1} = M^{-1}$$

# Algoritmo Gradiente Conjugado Precondicionado

Dados  $x_0 = 0$ ,  $A, b, M$ ,  $N_{max}$  e  $tol$

$$\rightarrow r_0 = b \quad v_0 = M^{-1}r_0$$

$$\rightarrow \delta_{novo} = r_0^t v_0, \delta_0 = \delta_{novo}, j = 0$$

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

$$p_j = A v_j$$

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

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

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

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

$$\rightarrow s = M^{-1} r_{j+1}$$

$$\rightarrow \delta_{novo} = r_{j+1}^t s$$

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

$$\rightarrow v_{j+1} = s + \beta v_j$$

$$j = j + 1$$

Fim

Como efetuar operação

$$p = M^{-1}v?$$

Resolvendo o sistema  $Mp = v$

Obs:

- A ação do preconditionador se dá durante o processo iterativo.
- Não é necessário calcular  $M^{-1}$
- O preconditionador será tão mais eficiente quanto mais simples for resolver o sistema  $Mp = v$  desde que a convergência seja acelerada.

# Precondicionando dos métodos baseado nos espaços de Krylov

- A ação do preconditionador se dá durante o processo iterativo no momento que for necessário executar operações que envolvem  $A$  e  $b$ :

⋮

$$b = M^{-1}b \quad (\Rightarrow Mb = b)$$

⋮ *(No loop de iterações)*

$$p = M^{-1}Av \quad (\Rightarrow q = Av)$$

$$Mp = q )$$

⋮

# Precondicionadores Globais

- Precondicionador Diagonal ou Jacobi

$$M = D$$

$$p = D^{-1}v$$

- Precondicionadores baseados na fatoração  $A=L+D+U$  (Gauss-Seidel)

– Não há necessidade de armazenamento extra

$$M = (U + D)(U^t + D)$$

$$Mp = v$$

$$(L + D) \underbrace{(U + D)p}_q = v$$

$$1) (L + D)q = v$$

$$2) (U + D)p = q$$

$$M = (L + D)D^{-1}(U + D)$$

$$Mp = v$$

$$(L + D) \underbrace{D^{-1} \overbrace{(U + D)p}^w}_q = v$$

$$1) (L + D)q = v$$

$$2) D^{-1}w = q$$

$$3) (U + D)p = w$$

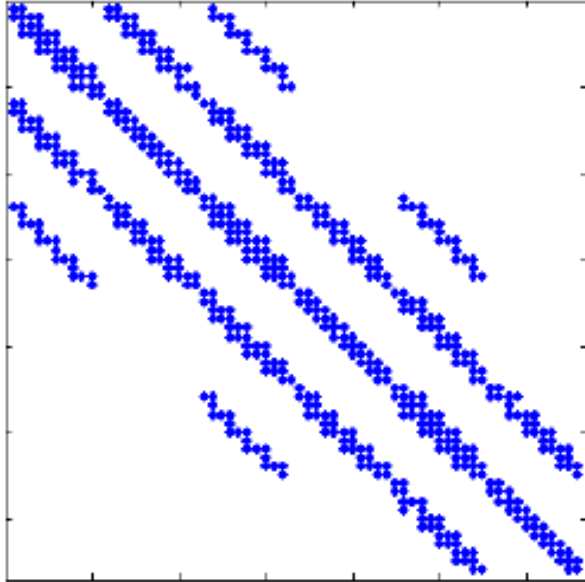
# Precondicionadores Globais

- Precondicionadores baseados na fatoração incompleta LU (ILU)
  - É necessário armazenamento extra para L e U (lembrar que a fatoração LU prevê preenchimento).
  - Definição de níveis de fill-in → fatoração incompleta
  - ILU(0): descarta todos os preenchimentos por coeficientes não nulos em posições originalmente nulas.
  - ILU(1): descarta os preenchimentos por coeficientes não nulos em posições originalmente nulas a partir da segunda etapa de decomposição.
  - ILU(k): descarta os preenchimentos por coeficientes não nulos em posições originalmente nulas a partir da k+1-ésima etapa de decomposição.

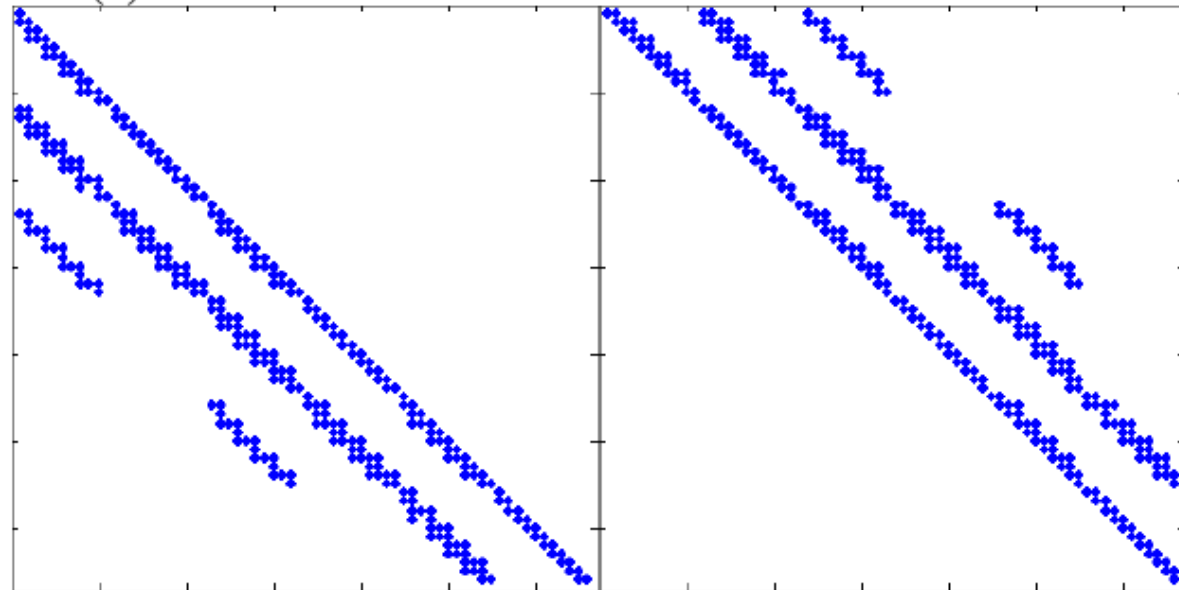


# Precondicionador Global ILU(k)

Matrix  $A$ : 582 nonzero entries.



ILU(0):  $A \approx \tilde{L}\tilde{U}$ : 648 total nonzero entries.



ILU(k):  $A \approx \tilde{L}\tilde{U}$

$$M = \tilde{L}\tilde{U}$$

$$\tilde{U}^{-1}\tilde{L}^{-1}Ax = \tilde{U}^{-1}\tilde{L}^{-1}b$$

No algoritmo

⋮

$$b = \tilde{U}^{-1}\tilde{L}^{-1}b \quad (\Rightarrow \tilde{L}\tilde{U}b = b)$$

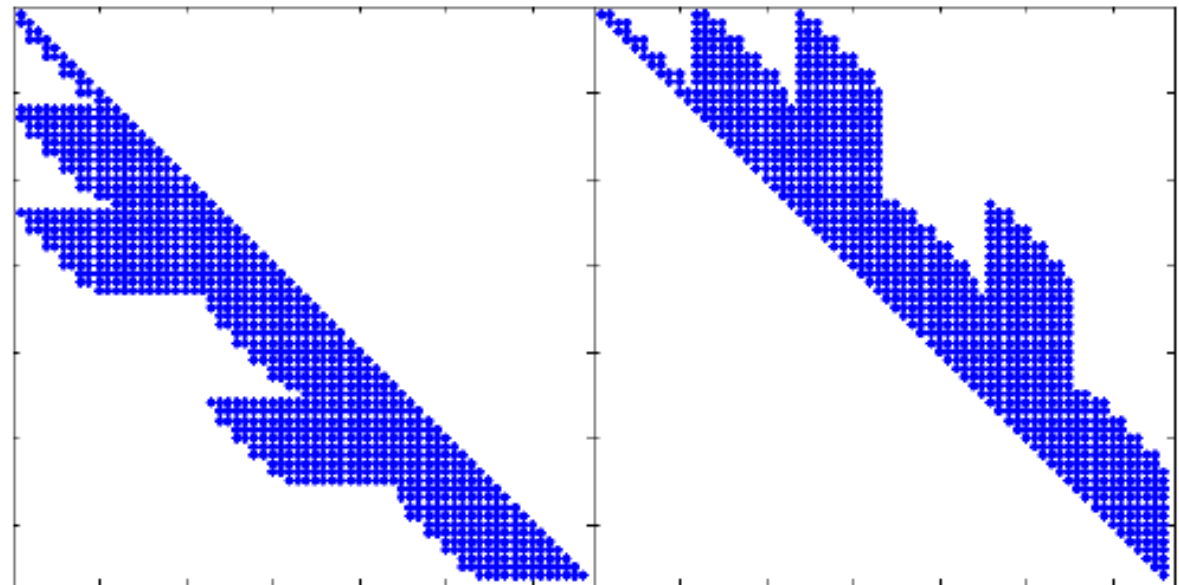
⋮ (No loop de iterações)

$$p = \tilde{U}^{-1}\tilde{L}^{-1}Av \quad (\Rightarrow q = Av)$$

$$\tilde{L}\tilde{U}p = q$$

⋮

$A = LU$ : 1950 total nonzero entries.



# Precondicionador Local EBE

$$(L^{-1}AU^{-1})(Ux) = (L^{-1}b)$$

$$L = \mathbf{A} \prod_{e=1}^{nel} \tilde{L}^e \quad U = \mathbf{A} \prod_{e=1}^{nel} \tilde{U}^e$$

$$\begin{aligned} \tilde{L}^e + \tilde{U}^e &= A^e - \text{diag}(A^e) + 2I^e \\ (\tilde{L}^e + I^e) + (\tilde{U}^e + I^e) &= A^e - \text{diag}(A^e) \end{aligned}$$

Fatoração  
Gauss-Seidel  
Local

ou

$$\tilde{L}^e \tilde{U}^e = A^e$$

Fatoração LU  
Local

# Precondicionador Local EDS

$$(L^{-1}AU^{-1})(Ux) = (L^{-1}b)$$

$$L = \prod_{s=1}^{nedges} \tilde{L}^s \quad U = \prod_{s=1}^{nedges} \tilde{U}^s$$

$$\tilde{L}^s + \tilde{U}^s = A^s - \text{diag}(A^s) + 2I^s$$

$$(\tilde{L}^s + I^s) + (\tilde{U}^s + I^s) = A^s - \text{diag}(A^s)$$

Fatoração  
Gauss-Seidel  
Local

ou

$$\tilde{L}^s \tilde{U}^s = A^s$$

Fatoração LU  
Local

# Pré-precondicionamento Local (EBE ou EDS)

$$\tilde{A}\tilde{x} = \tilde{b},$$

where  $\tilde{A} = W^{-1/2}AW^{-1/2}$ ,  $\tilde{x} = W^{1/2}x$ ,  $\tilde{b} = W^{-1/2}b$  and  $W = \text{diag}(A)$

$$\bar{A}\bar{x} = \bar{b},$$

where  $\bar{A} = L^{-1}\tilde{A}U^{-1}$ ,  $\bar{x} = U\tilde{x}$  and  $\bar{b} = U^{-1}\tilde{b}$

$$L = \mathbf{A}_{e=1}^{nel} L_e$$

$$U = \mathbf{A}_{e=1}^{nel} U_e$$

where  $L_e + U_e = \tilde{A}_e - \text{diag}(\tilde{A}_e) + 2I$

# Experimentos Numéricos

## Equação Convecção-Difusão

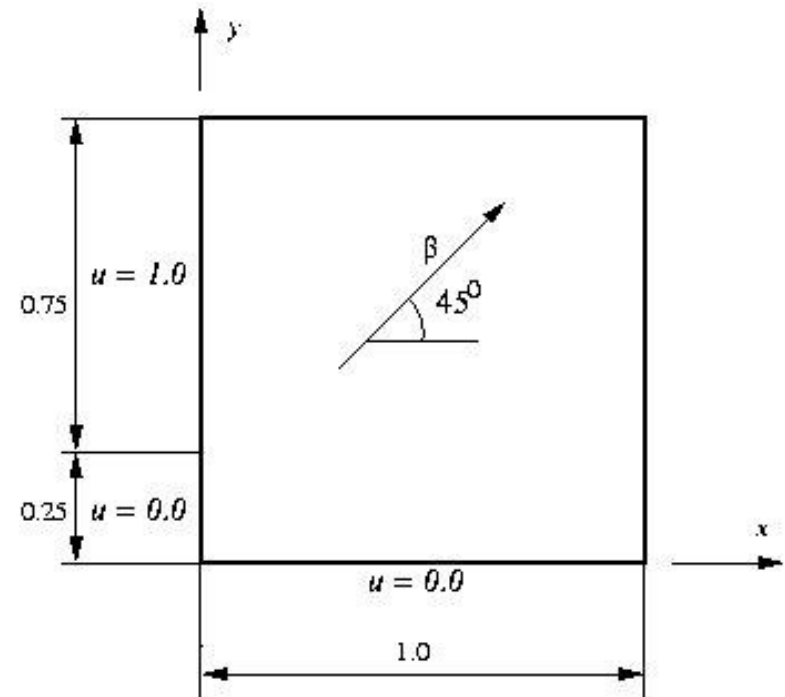
$$\beta \cdot \nabla u - \nabla \cdot (k \nabla u) = f$$

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

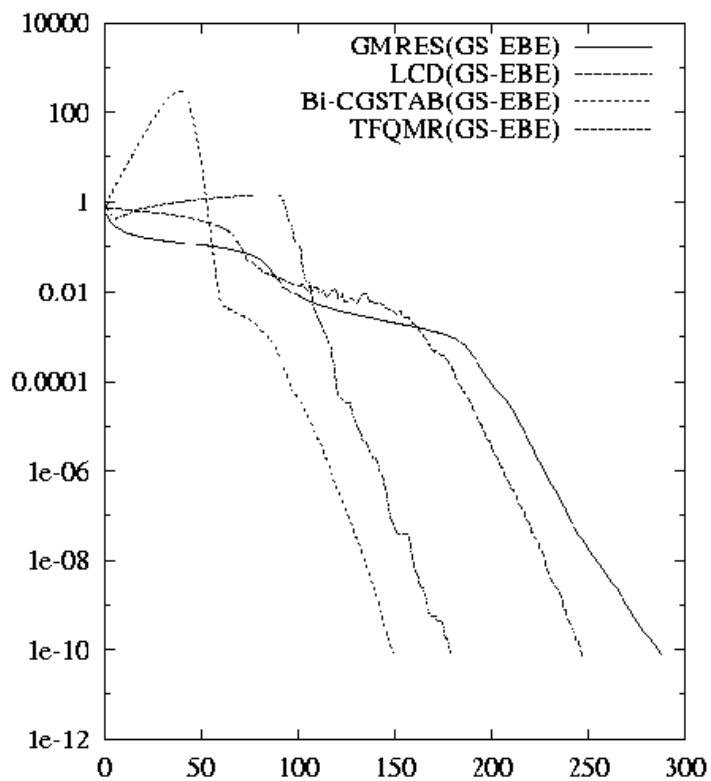
Formulação  
Elementos Finitos  
EBE

$$Kv = F$$

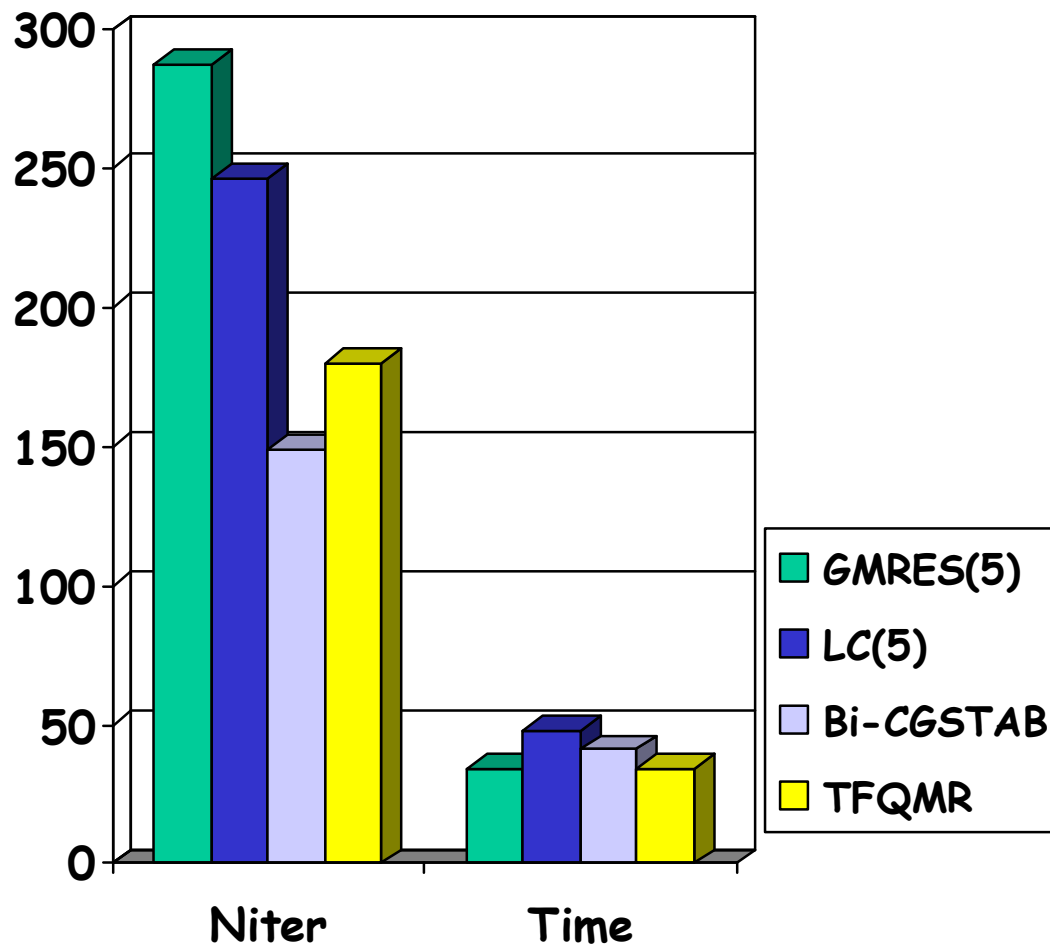
- $k = 10^{(-7)}$
- $\|\beta\| = 1$
- 128x128 células considerando 4 triângulos em cada célula
- Armazenamento EBE
- Precondicionador EBE Gauss-Seidel



# Experimentos Numéricos



Relative residual - mesh 128 x 128





# PETSc - Portable, Extensible Toolkit for Scientific Computation

- Developing parallel, non-trivial PDE solvers that deliver high performance is still difficult and requires months (or even years) of concentrated effort.
- PETSc is a toolkit that can ease these difficulties and reduce the development time, but it is not a black-box PDE solver nor a silver bullet.
- Mathematics and Computer Science Division - Argonne National Laboratory

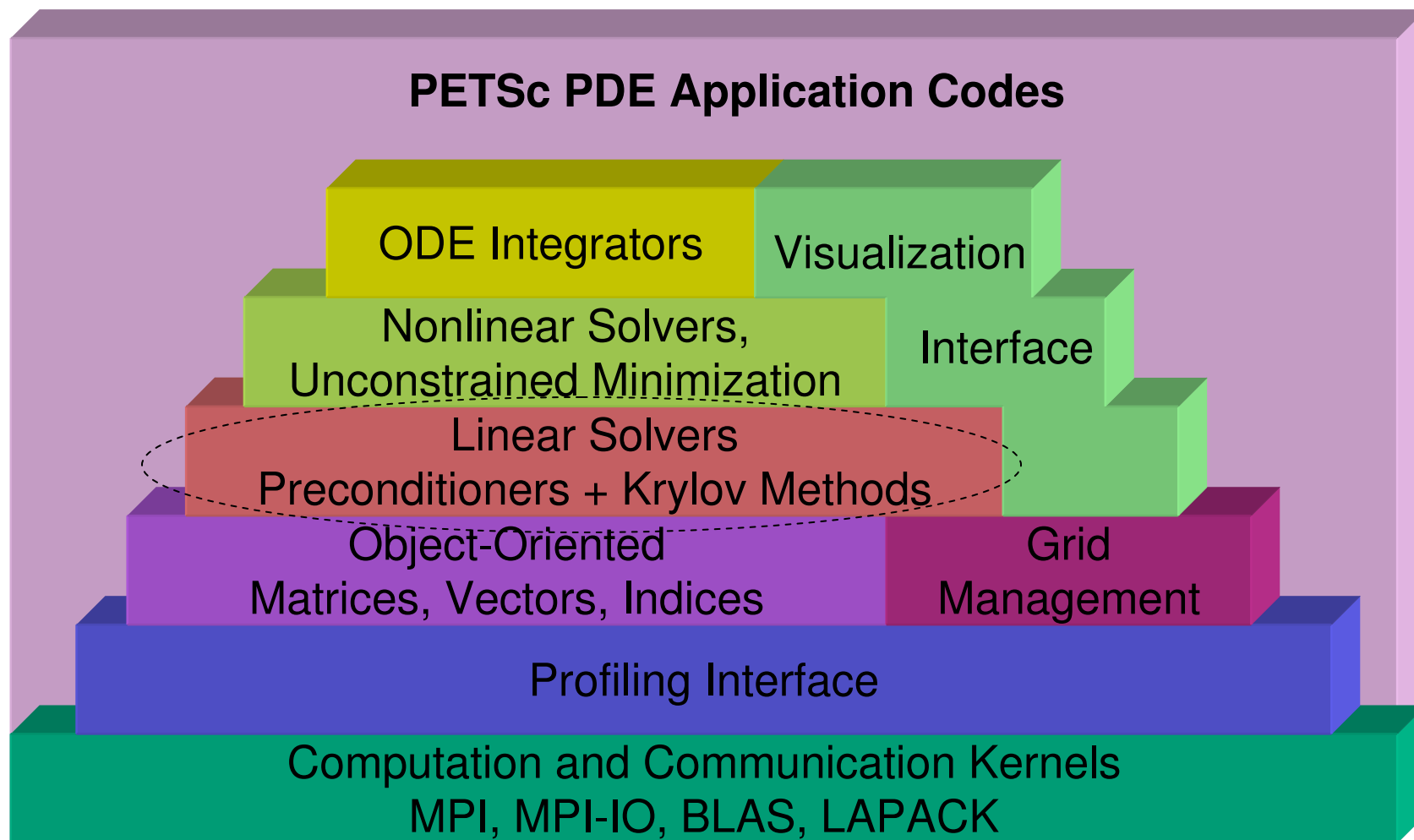


# What is PETSc?

- A freely available and supported research code
  - Available via <http://www.mcs.anl.gov/petsc>
  - Free for everyone, including industrial users (Open Source, non-GPL)
  - Hyperlinked documentation and manual pages for all routines
  - Many tutorial-style examples
  - Support via email: [petsc-maint@mcs.anl.gov](mailto:petsc-maint@mcs.anl.gov)
  - Usable from Fortran 77/90, C, and C++
- Portable to ANY parallel system supporting MPI
- PETSc history
  - Begun in September 1991
  - Now: over 8,500 downloads since 1995 (versions 2.0 and 2.1)
- PETSc funding and support
  - Department of Energy: MICS Program, DOE2000, SciDAC
  - National Science Foundation, Multidisciplinary Challenge Program, CISE



# Structure of PETSc





# PETSc Numerical Components

| Nonlinear Solvers    |              |       |
|----------------------|--------------|-------|
| Newton-based Methods |              | Other |
| Line Search          | Trust Region |       |

| Time Steppers |                |                      |       |
|---------------|----------------|----------------------|-------|
| Euler         | Backward Euler | Pseudo Time Stepping | Other |

| Krylov Subspace Methods |    |     |            |       |            |           |       |
|-------------------------|----|-----|------------|-------|------------|-----------|-------|
| GMRES                   | CG | CGS | Bi-CG-STAB | TFQMR | Richardson | Chebyshev | Other |

| Preconditioners   |              |        |     |     |                      |        |
|-------------------|--------------|--------|-----|-----|----------------------|--------|
| Additive Schwartz | Block Jacobi | Jacobi | ILU | ICC | LU (Sequential only) | Others |

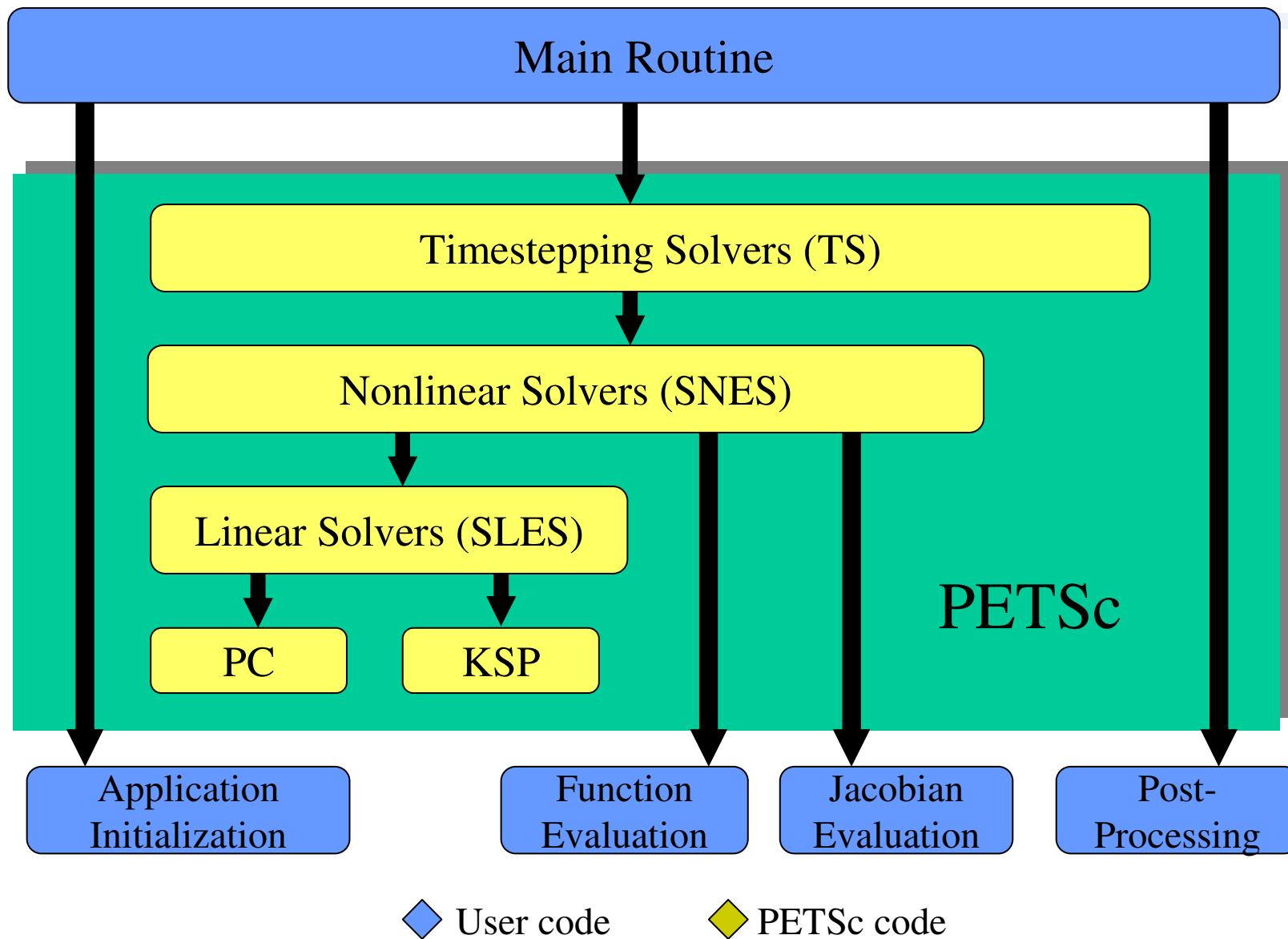
| Matrices                    |                                      |                        |       |             |       |
|-----------------------------|--------------------------------------|------------------------|-------|-------------|-------|
| Compressed Sparse Row (AIJ) | Blocked Compressed Sparse Row (BAIJ) | Block Diagonal (BDIAG) | Dense | Matrix-free | Other |

**Distributed Arrays**

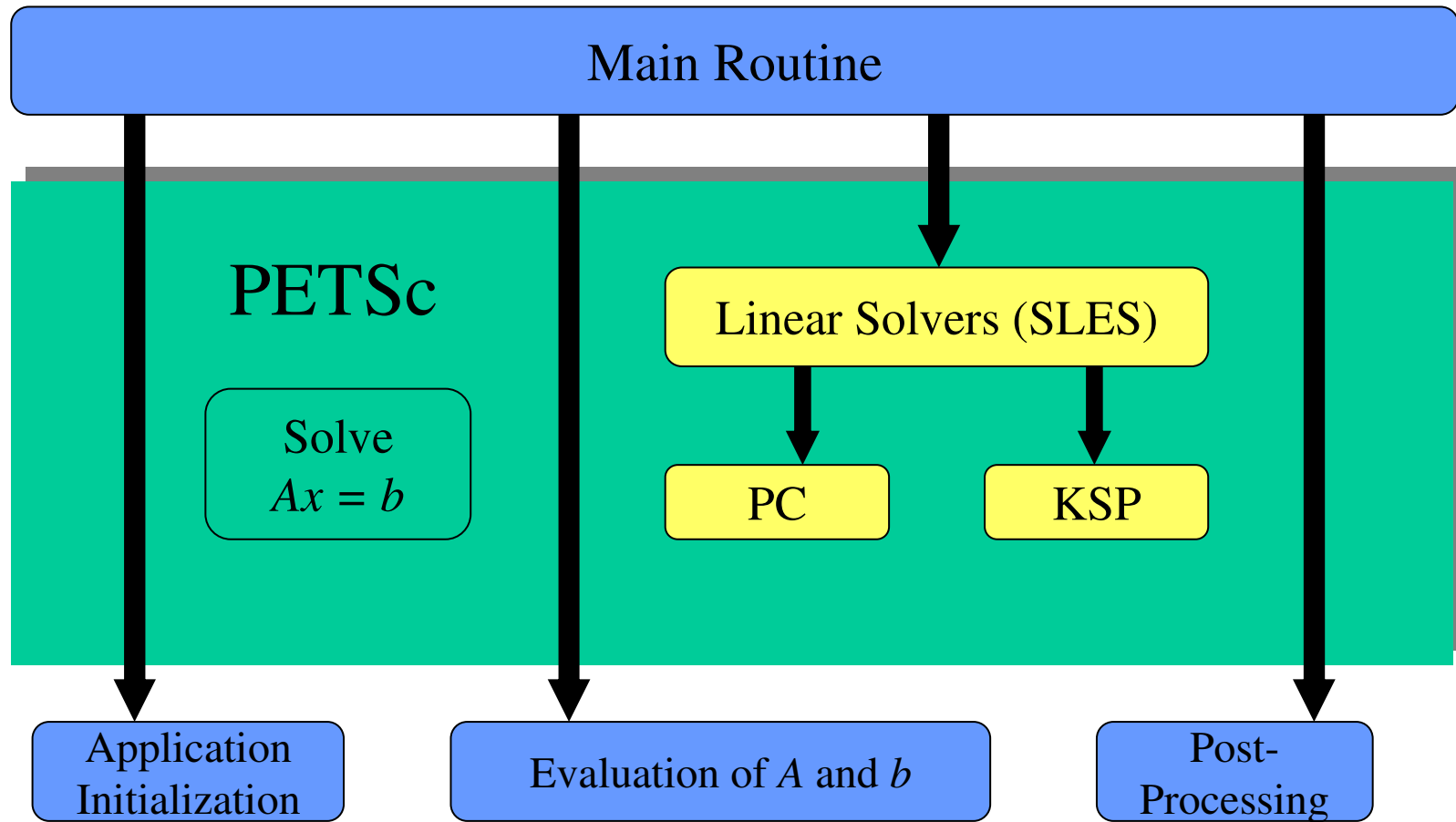
| Index Sets |               |        |       |
|------------|---------------|--------|-------|
| Indices    | Block Indices | Stride | Other |

**Vectors**

# Flow of Control for PDE Solution



# Linear PDE Solution



◆ User code

◆ PETSc code

beginne

solvers:  
linear

# Experimentos Numéricos

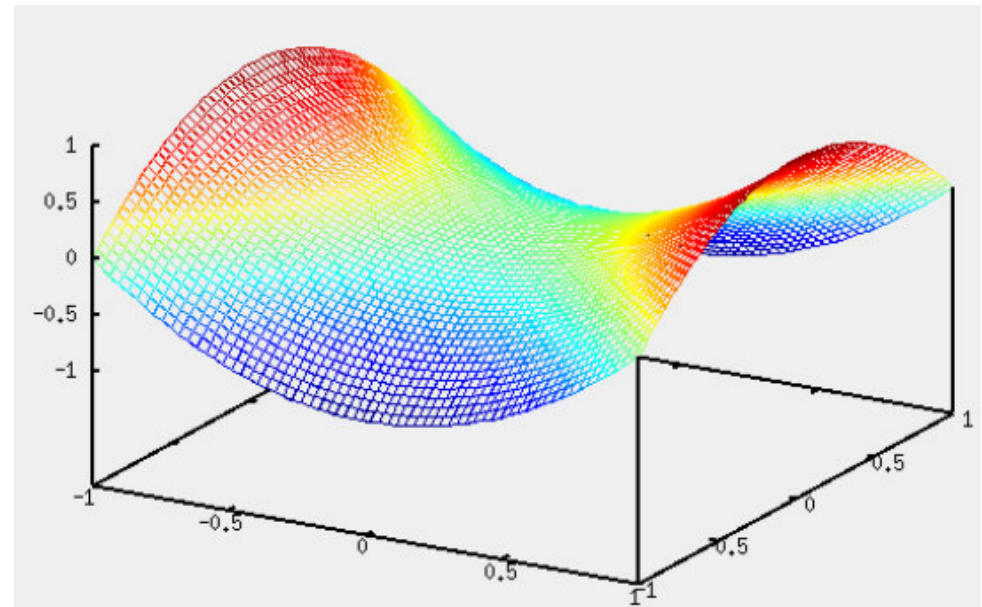
$$-\left(\frac{\partial^2 u}{\partial^2 x} + \frac{\partial^2 u}{\partial^2 y}\right) = 0 \quad (x, y) \in (-1, 1) \times (-1, 1)$$

$$\begin{cases} u(x, y) = x^2 - 1, & y = -1 \text{ e } y = 1 \quad -1 \leq x \leq 1 \\ u(1, y) = 1 - y^2, & x = -1 \text{ e } x = 1 \quad -1 \leq y \leq 1 \end{cases}$$

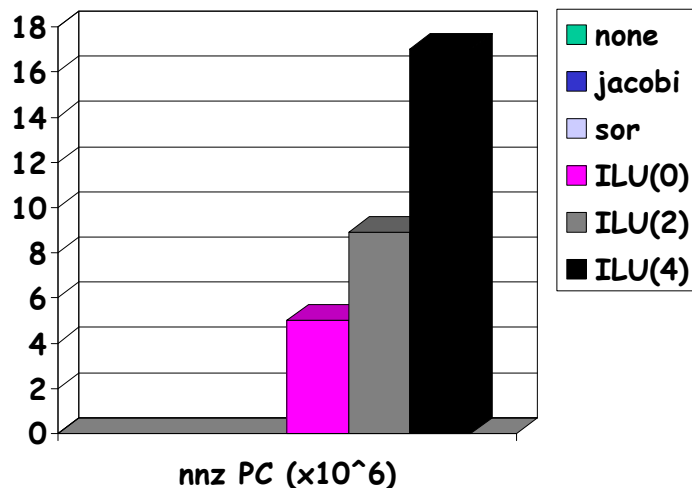
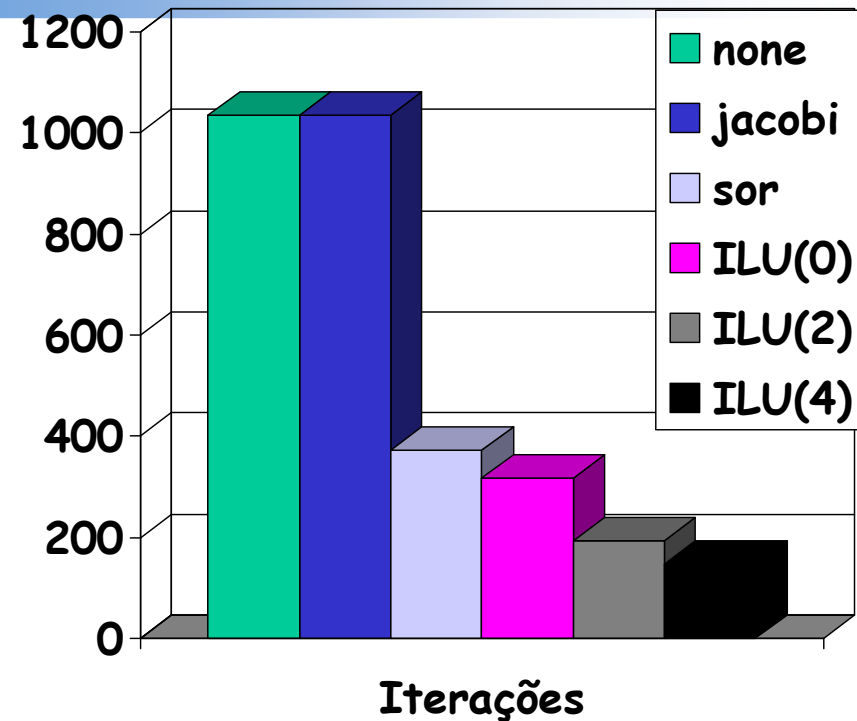
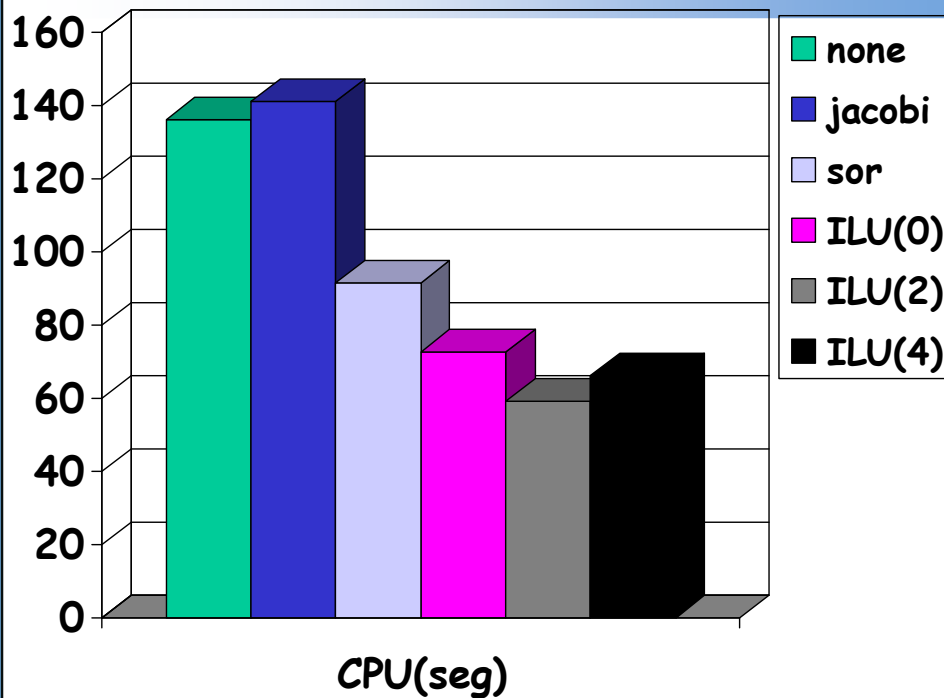
$$u_e(x, y) = x^2 - y^2$$

Considerando  
discretização pelo MDF e  
utilizando a Biblioteca  
PETSc

- Matriz simétrica (CG)
- Armazenamento CSR
- Malha 1000x1000
- $\text{tol} = 10^{-7}$



# Gradientes Conjugados x Precondicionadores



Qual é o preconditionador mais eficiente?

# Experimentos Numéricos

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

$$a(x, y) = 60$$

$$b(x, y) = 80$$

$$c(x, y) = -40$$

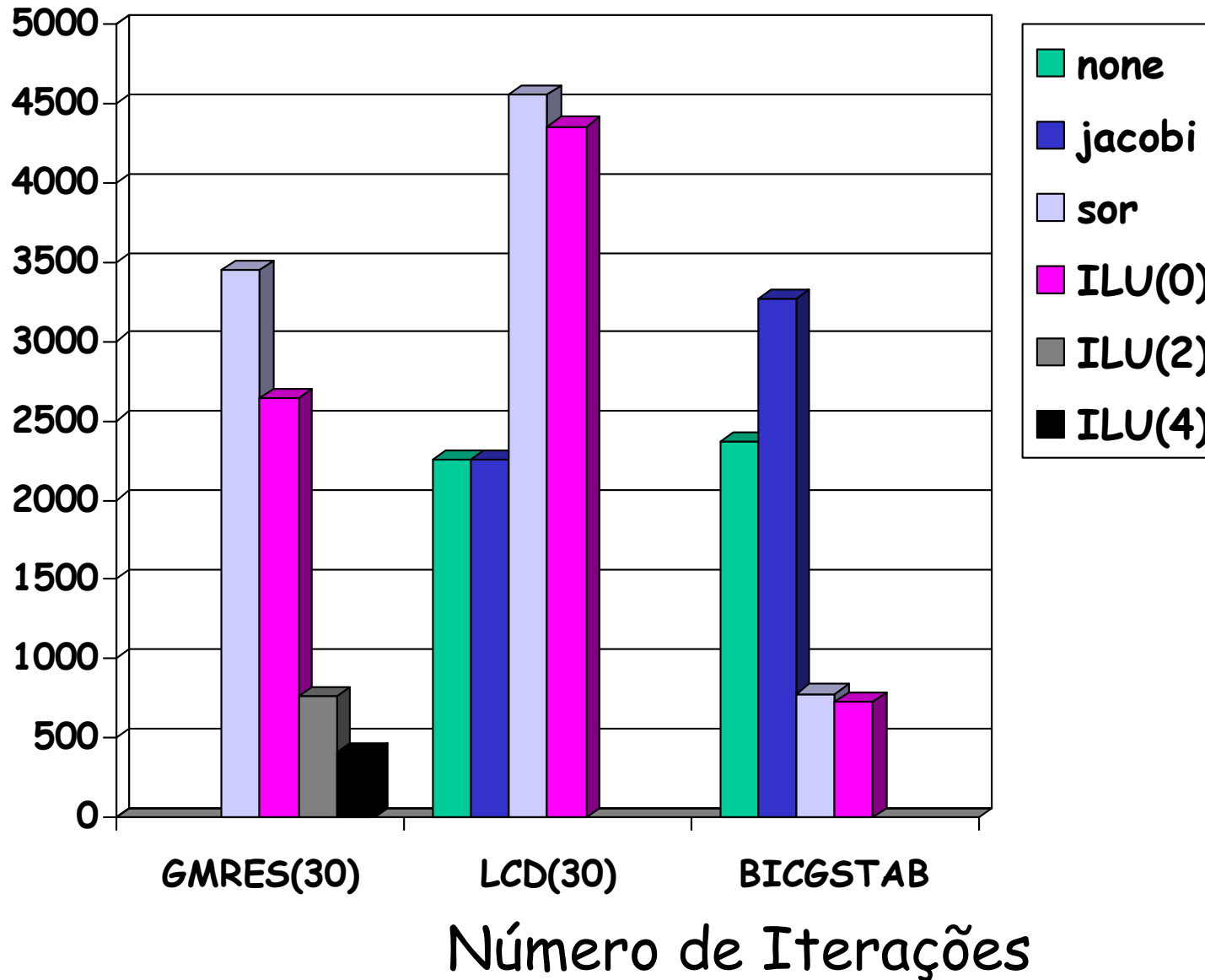
$$u_e(x, y) = x e^{xy} \text{sen}(\pi x) \cos(\pi y)$$

Considerando discretização pelo MDF e utilizando a Biblioteca PETSc

- GMRES, LCD, Bi-CGSTAB
- Armazenamento CSR
- Malha 1000x1000
- tol=10<sup>-7</sup>

Fonte: Relatório do aluno Rodrigo Sarmiento (Computação Científica - 2006/2)

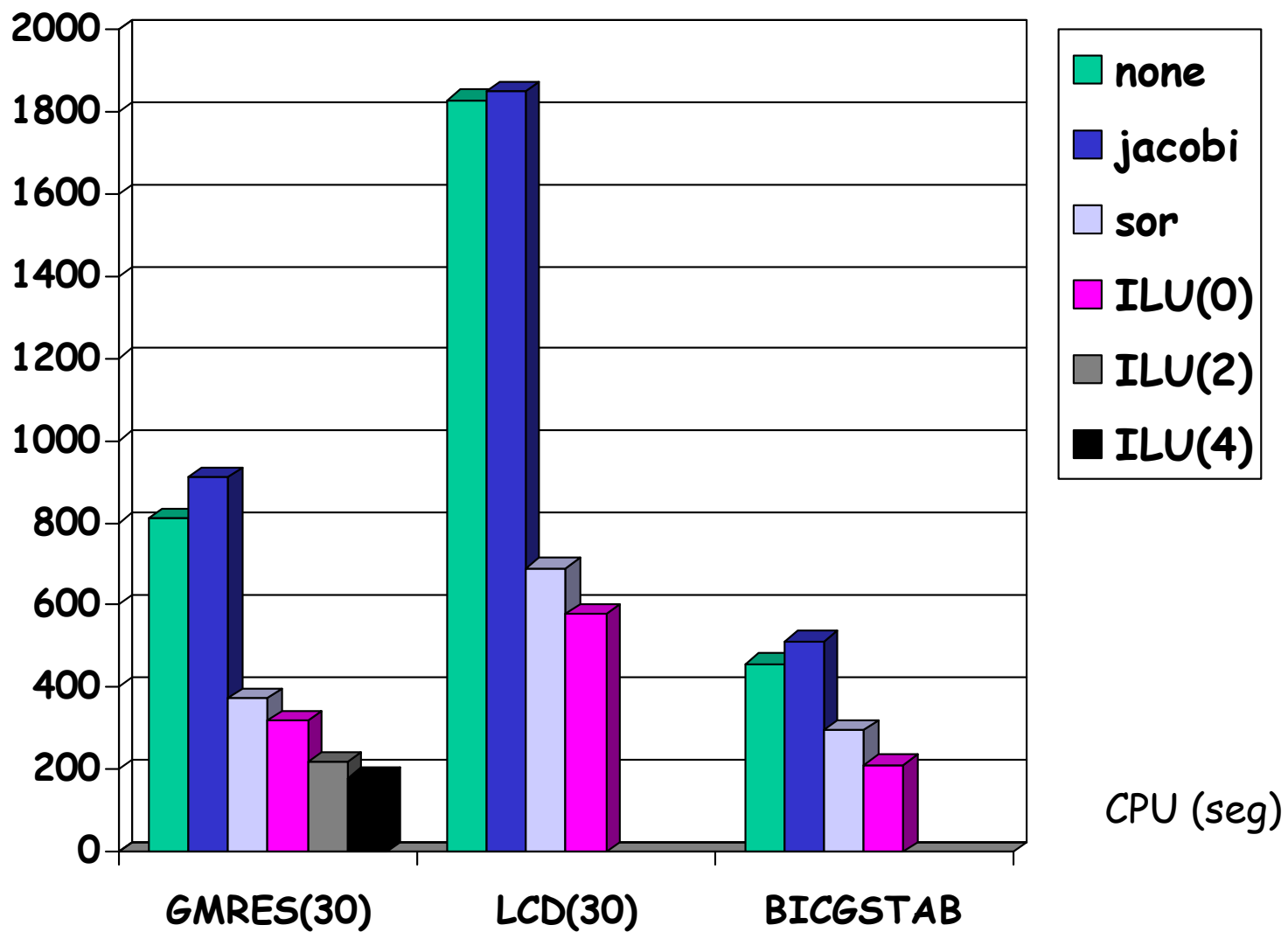
# GMRES, LCD, BICGSTAB x Precondicionadores







# GMRES, LCD, BI-CGSTAB x Precondicionadores



# Experimentos Numéricos

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

$$a(x, y) = 1$$

$$b(x, y) = 20y$$

$$c(x, y) = 1$$

$$u_e(x, y) = 10xy(1-x)(1-y)e^{x^{4.5}}$$

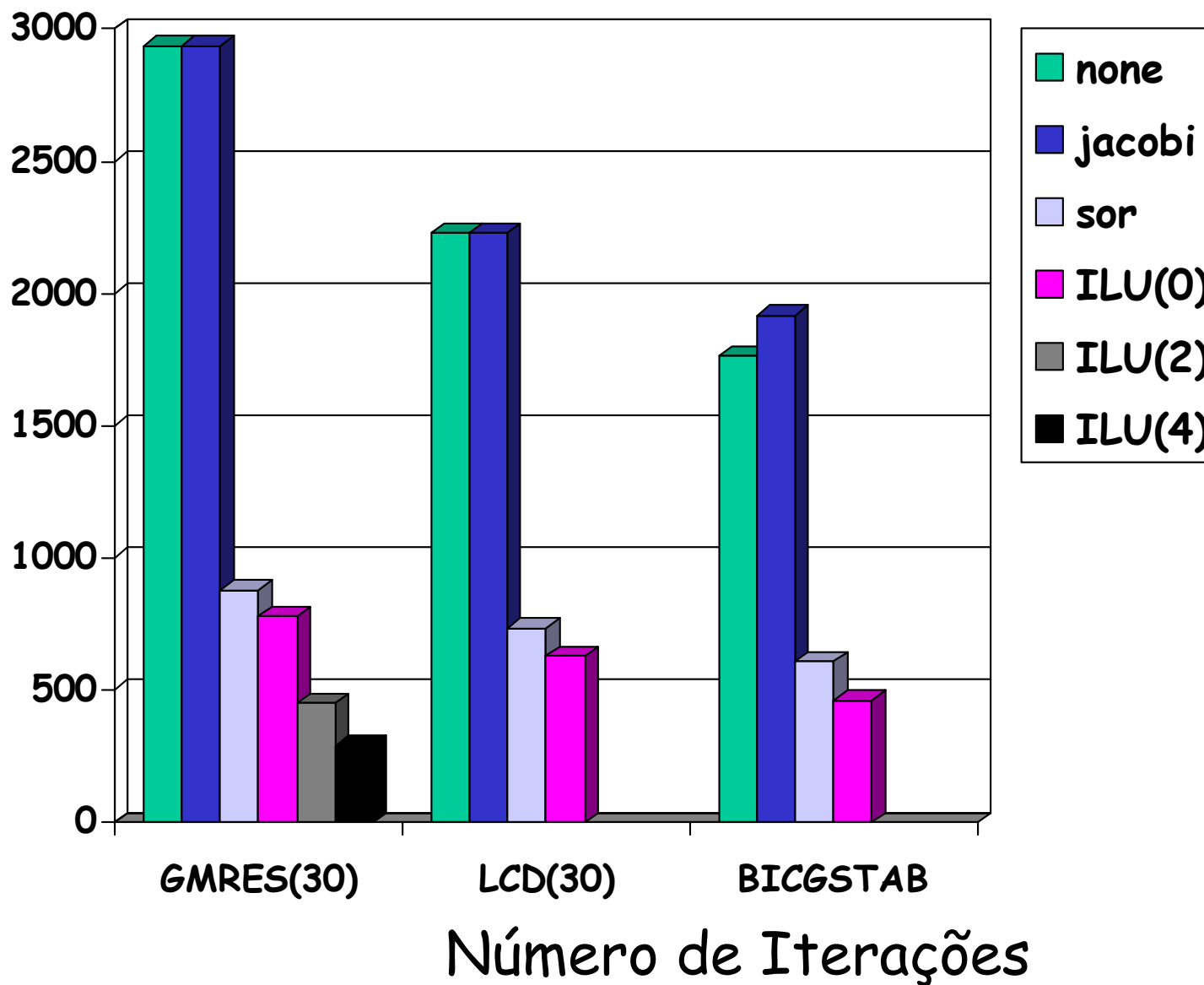
Considerando discretização pelo MDF e utilizando a Biblioteca PETSc

- GMRES, LCD, Bi-CGSTAB
- Armazenamento CSR
- Malha 1000x1000
- tol=10<sup>-7</sup>

Fonte: Relatório do aluno Rodrigo Sarmiento (Computação Científica - 2006/2)

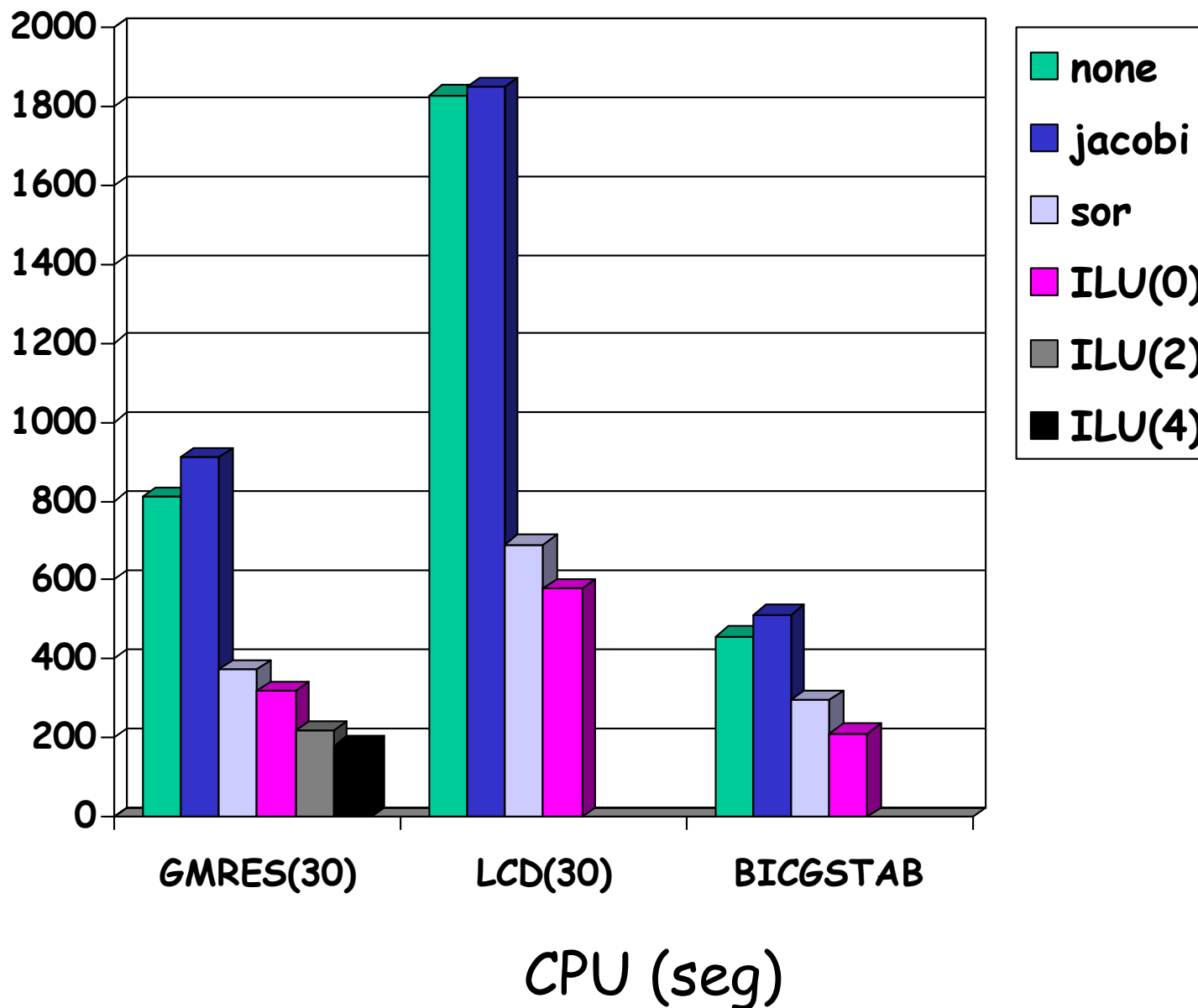


# GMRES, LCD, BI-CGSTAB x Precondicionadores





# GMRES, LCD, BI-CGSTAB x Precondicionadores



# Sistemas Não-Lineares

Equação de transferência  
de calor

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

Equação de convecção  
e difusão

$$Cu\nabla \cdot u - \nabla^2 u = f$$



Formulação de  
Diferenças  
Finitas

$$F(u_1, u_2, \dots, u_N) = \begin{bmatrix} f_1(u_1, u_2, \dots, u_N) \\ \vdots \\ f_N(u_1, u_2, \dots, u_N) \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}$$

# Método de Newton Inexato - Algoritmo

1. dados  $F, J, u^0, \tau_a, \tau_r, \eta_{max}, g$  ou  $t$
2.  $r^0 = ||F(u^0)||$
3. enquanto  $||F(u^k)|| > \tau_r r^0 + \tau_a$  faça
  - 3.1. escolher critério de parada  $\eta_k$
  - 3.2. calcular  $J(u^k)$
  - 3.3. resolver  $J(u^k)s^k = -F(u^k)$  com toler.  $\eta_k$
  - 3.4.  $u^{k+1} = u^k + s^k$
  - 3.5. avaliar  $F(u^k)$

Critério de Papadrakakis

$$\eta_k = \min \left\{ \eta_{\max}, \left( \frac{\|F(u^k)\|}{r_0} \right)^t \right\}$$

Critério de Kelley

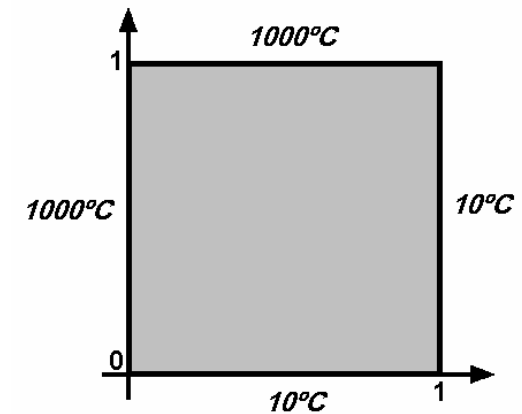
$$\eta_k = \begin{cases} \eta_{\max}, & k = 0 \\ \min(\eta_{\max}, \gamma \frac{\|F(u^k)\|^2}{\|F(u^{k-1})\|^2}), & k > 0 \text{ and } \gamma \eta_{k-1}^2 < 0.1 \\ \min(\eta_{\max}, \max(\frac{\|F(u^k)\|^2}{\|F(u^{k-1})\|^2}, \gamma \eta_{k-1}^2)), & k > 0 \text{ and } \gamma \eta_{k-1}^2 \geq 0.1 \end{cases}$$

# Exemplo Numérico: transferência de calor com condutividade quadrática

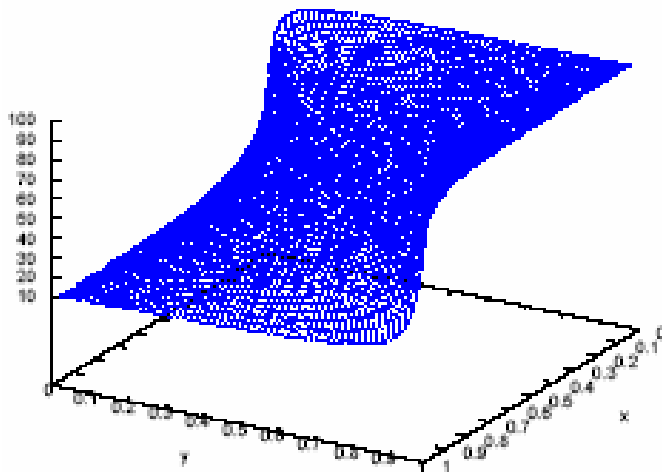
$$-\nabla(k(u) \cdot \nabla u) = 0 \quad \text{em} \quad (0,1) \times (0,1)$$

$$\text{bc: } u(x,0) = u(1,y) = 10 \quad \text{e} \quad u(x,1) = u(0,y) = 1000$$

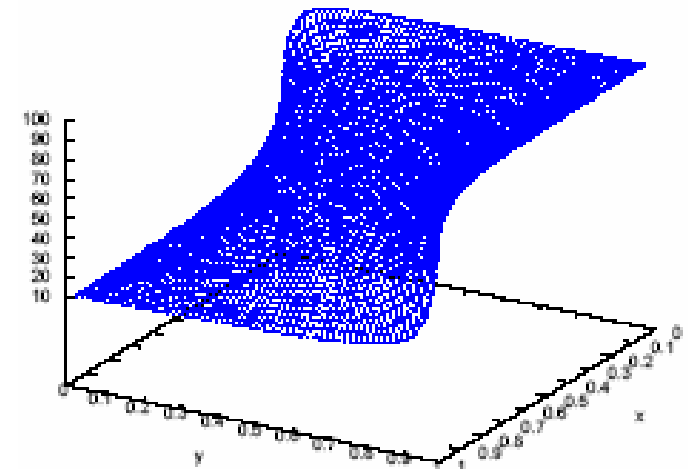
$$\text{onde: } k(u) = 0.001(1 + 0.01u + 0.0002u^2)$$



Mesh 64x64



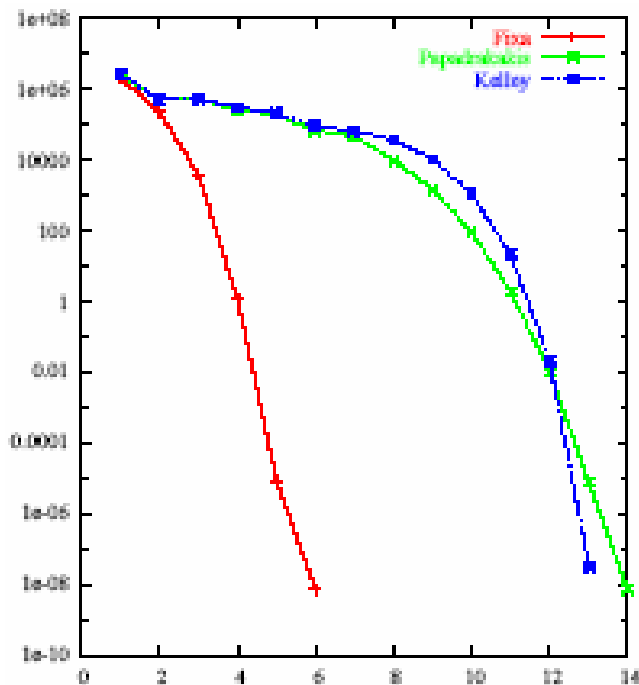
LCD(10)



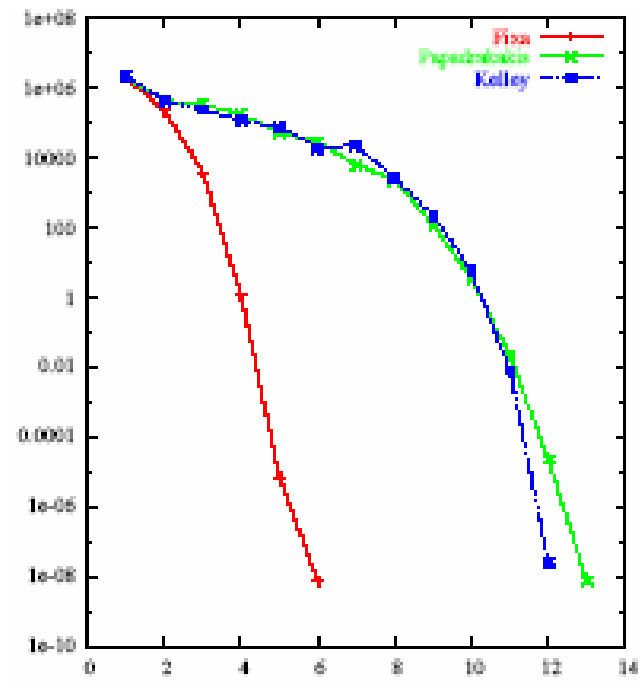
GMRES(10)

# Exemplo Numérico: transferência de calor com condutividade quadrática

## Convergência



LCD(10)



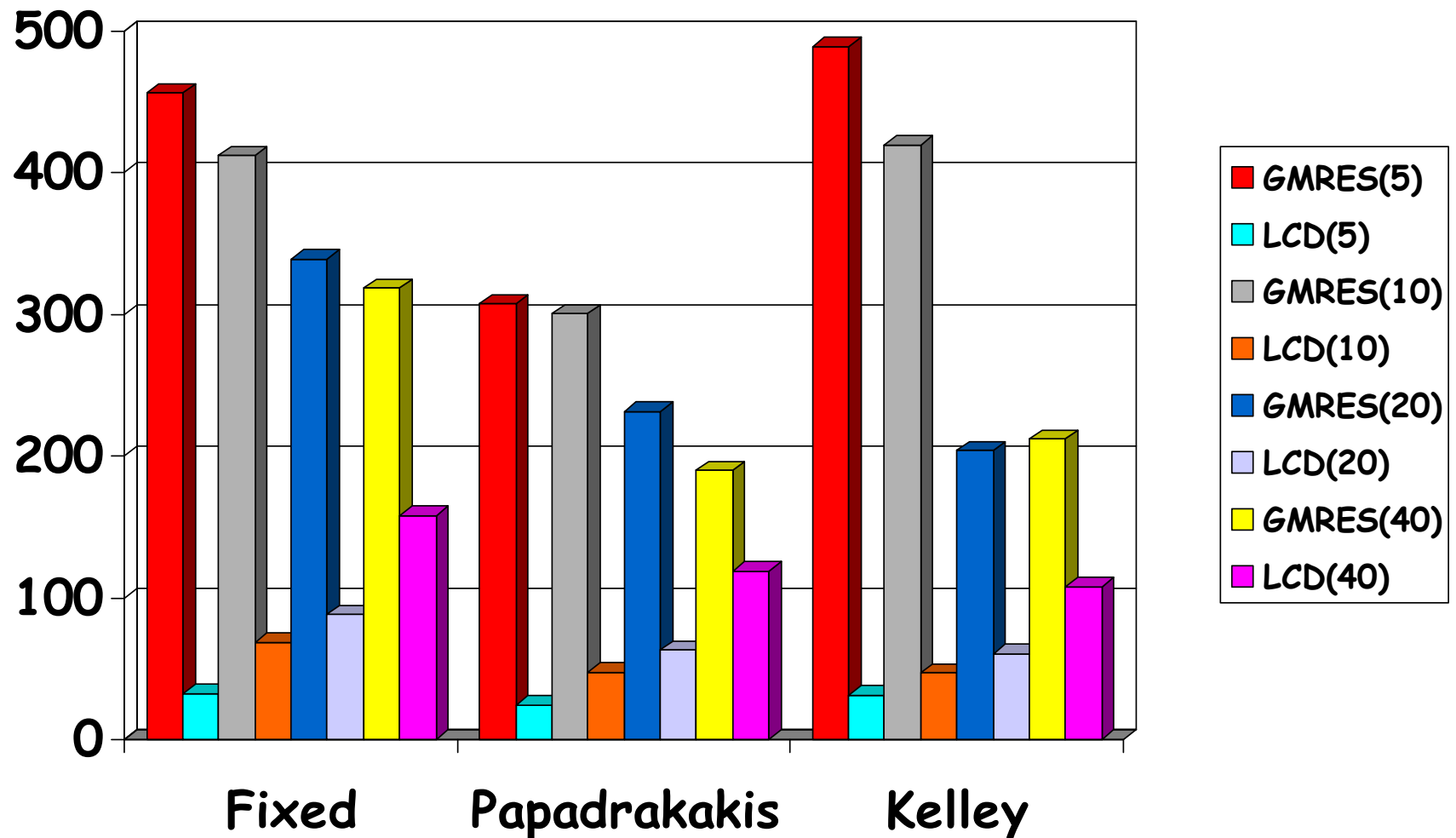
GMRES(10)

Iterações Não-Lineares  $\times \|F(u^k)\|$   
 Malha  $512 \times 512$      $\eta_{\text{fixo}} = 10^{-5}$



# Exemplo Numérico: transferência de calor com condutividade quadrática

Tempo computacional - malha 512x512

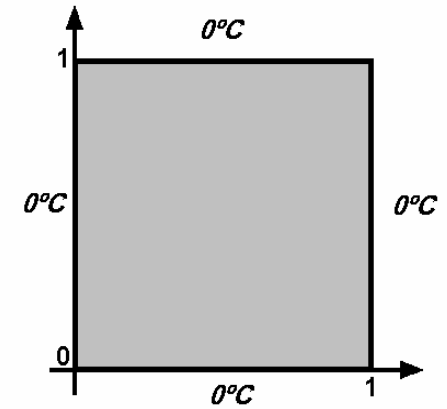


# Exemplo Numérico: convecção e difusão não-linear

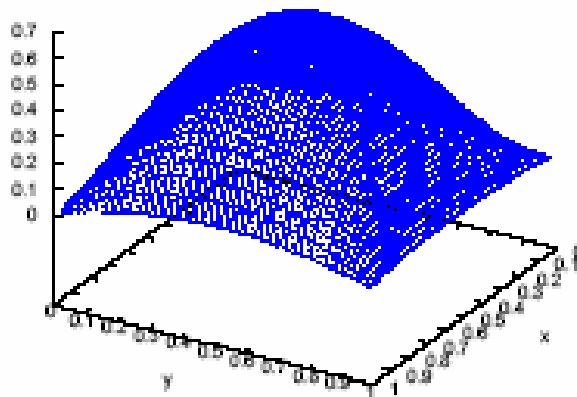
$$-\nabla^2 u + 20u \nabla \cdot u = g \quad \text{em } (0,1) \times (0,1)$$

$$\text{bc: } u(x,0) = u(1,y) = u(x,1) = u(0,y) = 0$$

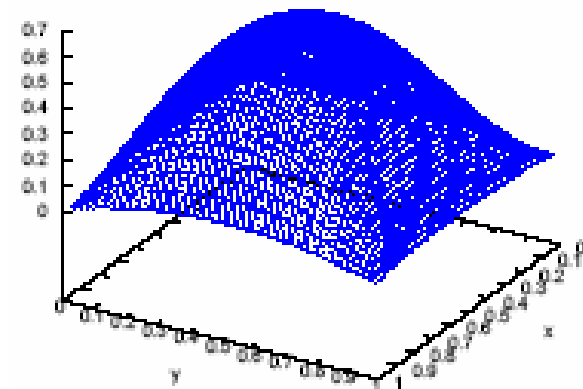
$$g \text{ tal que } u_{\text{exata}} = 10xy(1-x)(1-y)\exp(x4.5)$$



Malha  $64 \times 64$



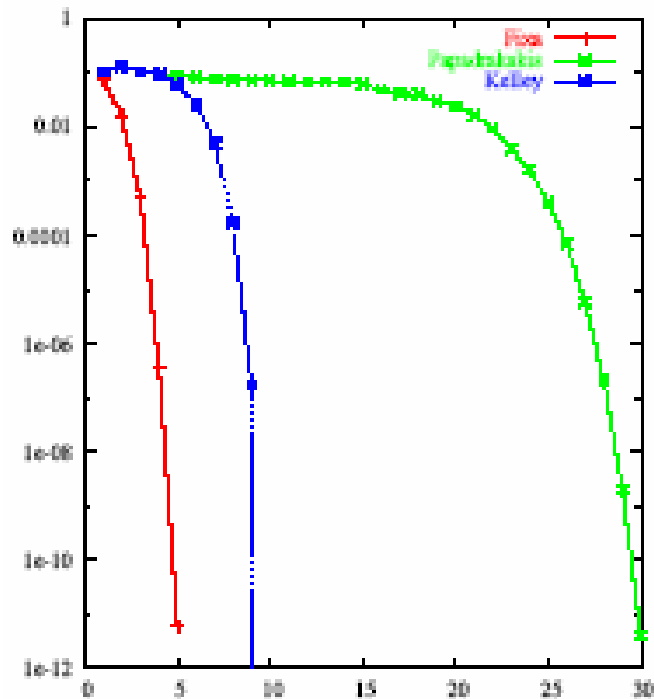
LCD(10)



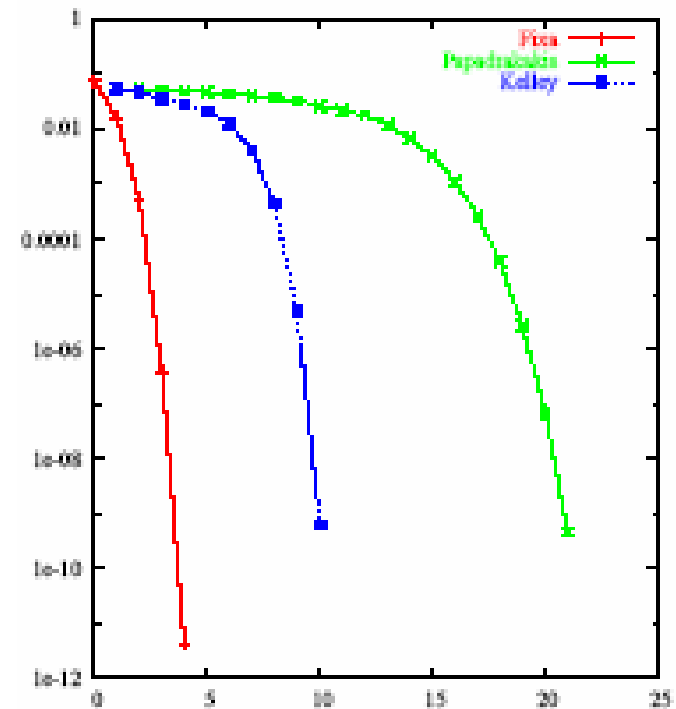
GMRES(10)

# Exemplo Numérico: convecção e difusão não-linear

## Convergência



LCD(10)



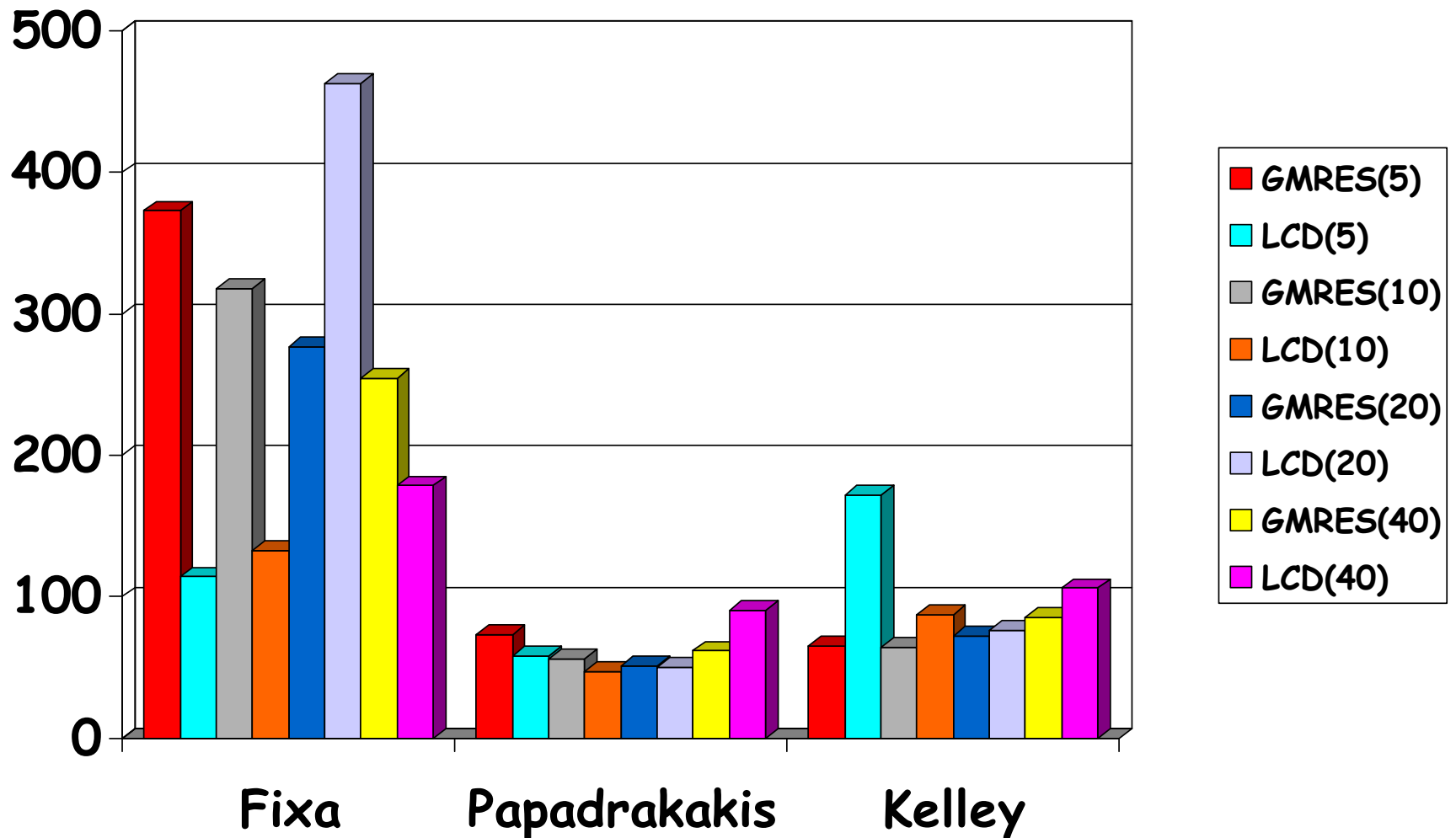
GMRES(10)

Iterações Não-Lineares  $\times \|F(u^k)\|$

Malha  $512 \times 512$   $\eta_{\text{fixo}} = 10^{-5}$

# Exemplo Numérico: covencção e difusão não-linear

Tempo computacional - malha 512x512





# Conclusões – Estrutura de Dados e Solvers

- O caminho do “alto desempenho” ...
  - o que está tentando implementar? ... características dos métodos, condições de convergência, esparsidade da matriz dos coeficientes,
  - Como implementar? ... cuidado com operações simples! ... lembrem-se o código serial deve ser bom!
  - É muito importante observar o número de operações de ponto flutuante do seu código!
  - Não existe método “melhor” ... pode existir método que funciona melhor sob condições específicas, mas ... nunca generalize!



**Lucia Catabriga**

luciac@inf.ufes.br

[www.inf.ufes.br/~luciac](http://www.inf.ufes.br/~luciac)

**Mestrado em Informática**

**Área: Computação de Alto Desempenho**

Departamento de Informática

Laboratório de Computação de Alto Desempenho - LCAD

Universidade Federal do Espírito Santo - UFES