

# TEI: Computação Científica Combinatorial

## Introdução a Computação Científica

Lucia Catabriga

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



# Sumário

---

- Processos de Solução
- Método das Diferenças Finitas
- Método dos Elementos Finitos
- Estrutura de Dados envolvidas

# Processo de Solução

- Fenômeno Natural
- Modelo Matemático - Equações Governantes
- Métodos de Aproximação

Diferenças Finitas  
Volumes Finitos  
Elementos Finitos  
Elementos de Contorno

# Etapas de Solução

## **Pré-processamento dos dados:**

- Condições de Contorno
- Condições Iniciais
- Definição do domínio discretizado

## **Processamento de solução:**

- Para cada ponto de interesse do domínio discretizado da malha montar estrutura de solução
- Obter solução aproximada ou solução no tempo corrente

## **Pós-processamento dos Resultados:**

- Visualização e análise dos resultados obtidos

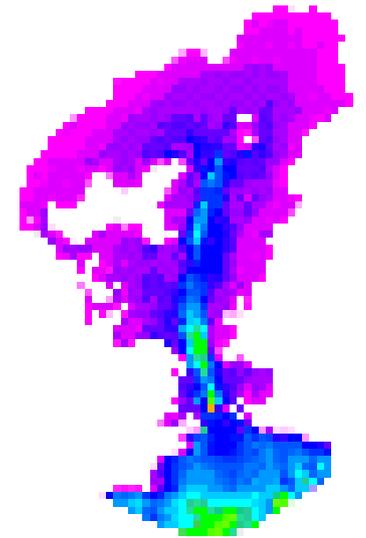
# Exemplo do Processo de Solução



Domínio Real



Domínio Discretizado

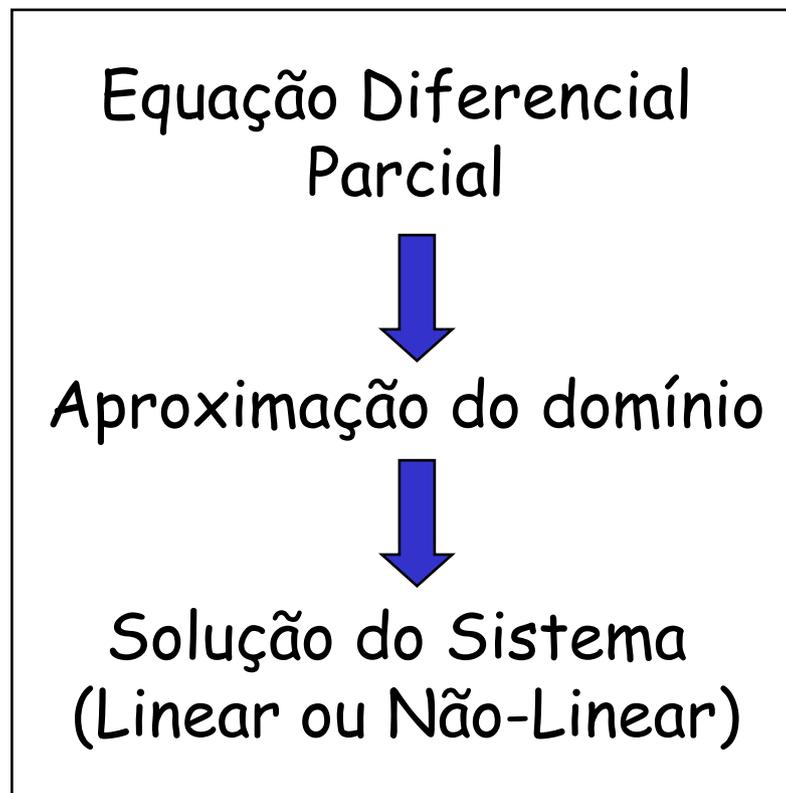


Solução Aproximada

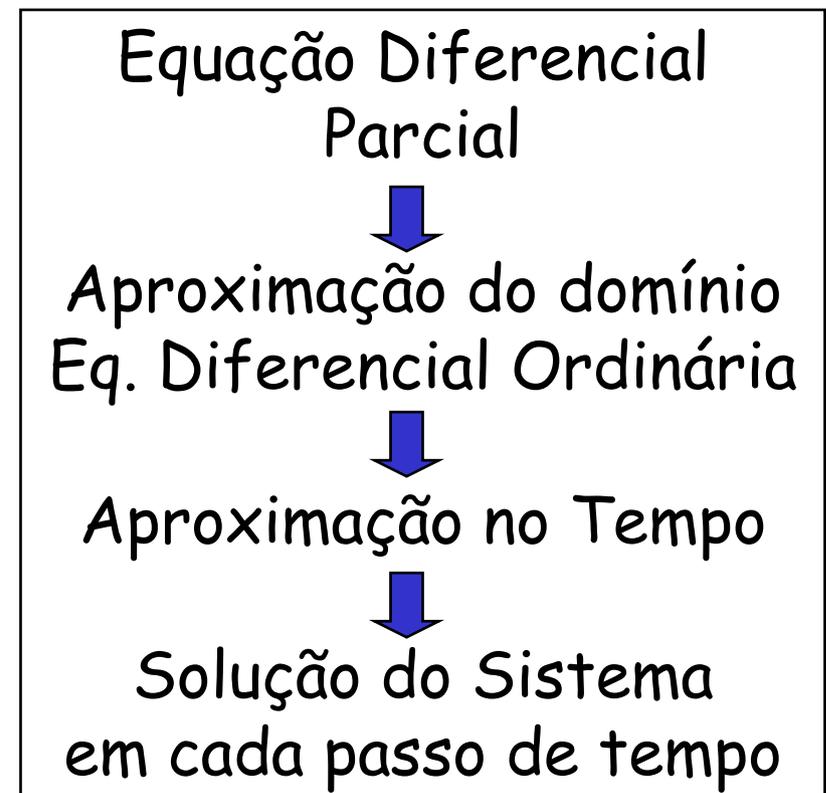
Dispersão de Poluentes na Baía de Guanabara

# Processo de Solução

Não dependem do Tempo



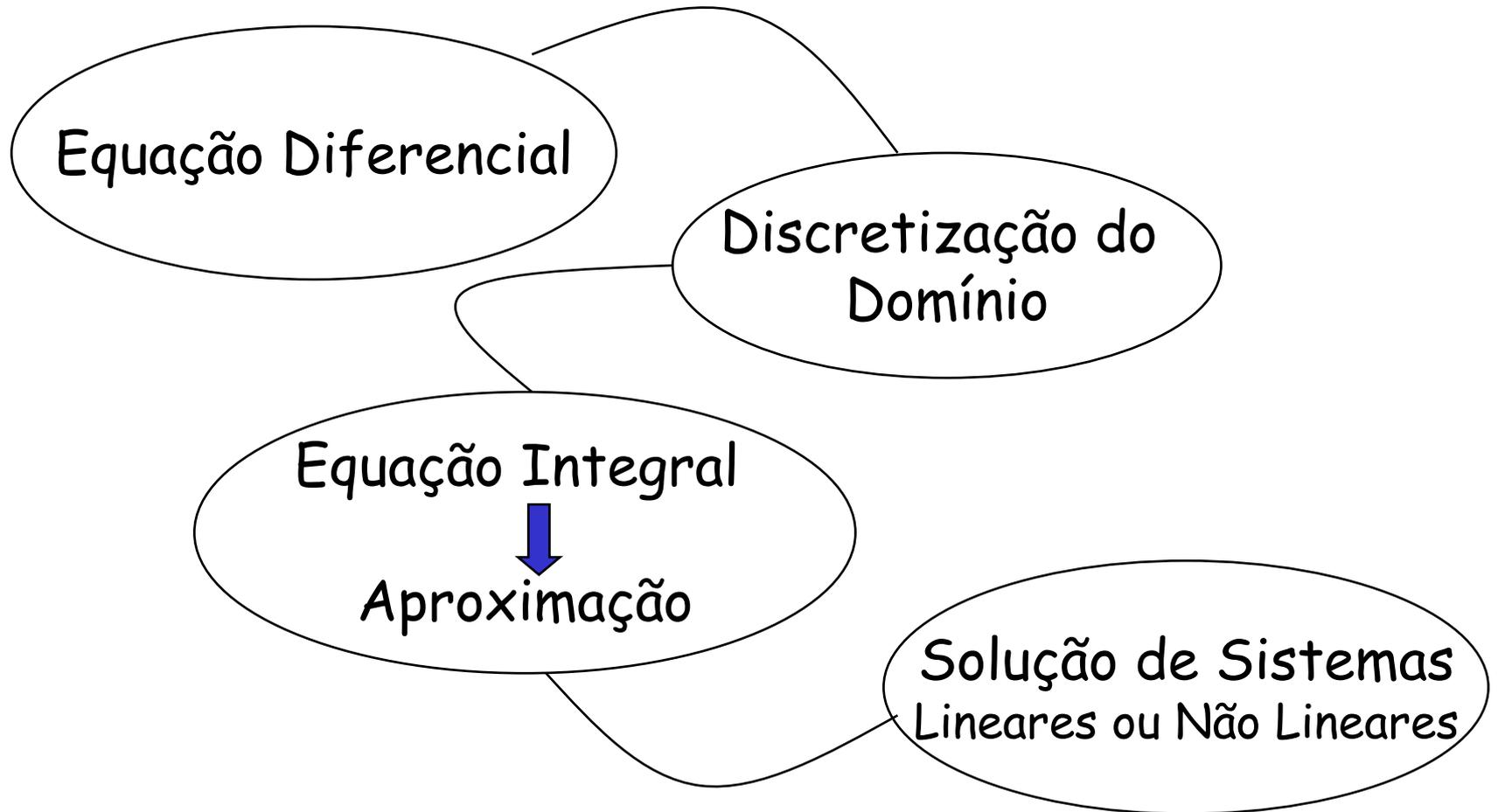
Dependem do Tempo



# Método das Diferenças Finitas (MDF)



# Método dos Elementos Finitos



# MDF: exemplo de solução para equação 1D

Equação diferencial contínua

$$\begin{cases} u'' = f & \text{em } (0,1) \\ u(0) = u(1) = 0 \end{cases}$$

1- Discretização do Domínio:

$$0 = x_0 \leq x_1 \leq \dots x_{i-1} \leq x_i \leq x_{i+1} \leq \dots x_{n+1} = 1$$

$$x_i = x_0 + i\Delta x$$

2 - Aproximação das derivada por "diferenças":

$$u' = \frac{u_i - u_{i-1}}{\Delta x}, O(\Delta x)$$

$$u' = \frac{u_{i+1} - u_{i-1}}{2\Delta x}, O(\Delta x^2)$$

$$u' = \frac{u_{i+1} - u_i}{\Delta x}, O(\Delta x)$$

$$u'' = \frac{u_{i-1} - 2u_i + u_{i+1}}{\Delta x^2}, O(\Delta x^2)$$

# MDF: exemplo de solução para equação 1D

2 - Aproximação das derivada por "diferenças":

$$\underbrace{\left(\frac{1}{\Delta x^2}\right)}_b u_{i-1} + \underbrace{\left(-\frac{2}{\Delta x^2}\right)}_a u_i + \underbrace{\left(\frac{1}{\Delta x^2}\right)}_b u_{i+1} = f_i, \quad i = 1, \dots, n$$



$$\begin{bmatrix} a & b & & & & \\ b & a & b & & & \\ & \ddots & \ddots & \ddots & & \\ & & b & a & b & \\ & & & b & a & \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{n-1} \\ u_n \end{bmatrix} = \begin{bmatrix} f_1 - bu_0 \\ f_2 \\ \vdots \\ f_{n-1} \\ f_n - bu_{n+1} \end{bmatrix} \quad A \text{ é tridiagonal} \quad \begin{bmatrix} 0 & a & b \\ b & a & b \\ \vdots & \vdots & \vdots \\ b & a & b \\ b & a & 0 \end{bmatrix}$$

# MDF: exemplo de solução para equação 2D

Equação diferencial contínua

$$\left\{ \begin{array}{l} \nabla^2 u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f \text{ em } (0,1) \times (0,1) \subset \square^2 \\ u = g(x, y) \text{ e / ou } \frac{\partial u}{\partial \mathbf{n}} = h(x, y) \text{ em } \Gamma \end{array} \right.$$

1- Discretização do Domínio:

$$0 = x_0 \leq x_1 \leq \dots x_{i-1} \leq x_i \leq x_{i+1} \leq \dots x_{n+1} = 1, \quad x_i = x_0 + i\Delta x$$

$$0 = y_0 \leq y_1 \leq \dots y_{j-1} \leq y_j \leq y_{j+1} \leq \dots y_{m+1} = 1, \quad y_j = y_0 + j\Delta y$$

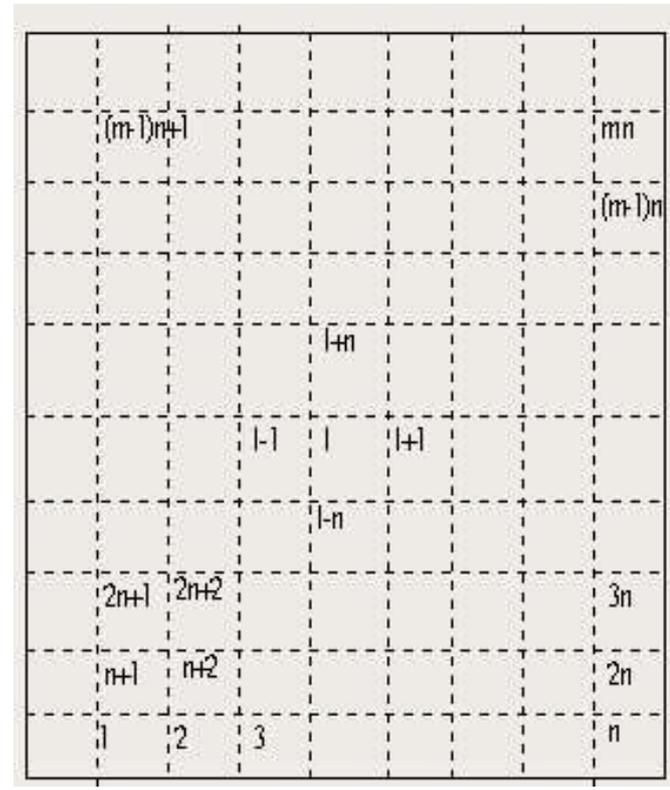
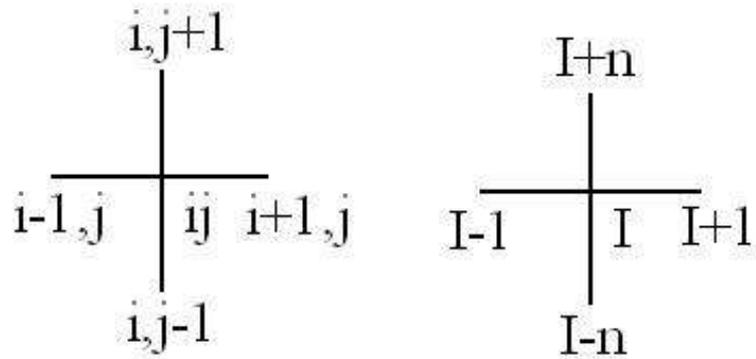
2 - Aproximação das derivadas por "diferenças":

$$\frac{\partial u}{\partial x} = \frac{u_{i+1,j} - u_{i-1,j}}{2\Delta x}, \quad O(\Delta x^2)$$

$$\frac{\partial^2 u}{\partial x^2} = \frac{u_{i-1,j} - 2u_{ij} + u_{i+1,j}}{\Delta x^2}, \quad O(\Delta x^2)$$

# MDF: exemplo de solução para equação 2D

Numeração das incógnitas:



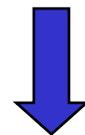
$$\frac{u_{i-1,j} - 2u_{ij} + u_{i+1,j}}{\Delta x^2} + \frac{u_{i,j-1} - 2u_{ij} + u_{i,j+1}}{\Delta y^2} = f_{ij}, \quad i = 1, \dots, n \quad j = 1, \dots, m$$

$$\frac{u_{I-1} - 2u_I + u_{I+1}}{\Delta x^2} + \frac{u_{I-n} - 2u_I + u_{I+n}}{\Delta y^2} = f_I, \quad I = 1, \dots, N$$

# MDF: exemplo de solução para equação 2D

2 - Aproximação das derivada por "diferenças":

$$\underbrace{\left(\frac{1}{\Delta y^2}\right)}_c u_{I-n} + \underbrace{\left(\frac{1}{\Delta x^2}\right)}_b u_{I-1} - \underbrace{2\left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2}\right)}_a u_I + \underbrace{\left(\frac{1}{\Delta x^2}\right)}_b u_{I+1} + \underbrace{\left(\frac{1}{\Delta y^2}\right)}_c u_{I+n} = f_I, \quad I = 1, \dots, n \times m$$



$$\begin{bmatrix} a & b & c & & & & & & & & \\ b & a & b & c & & & & & & & \\ & b & a & 0 & c & & & & & & \\ c & & 0 & a & b & c & & & & & \\ & c & & b & a & b & c & & & & \\ & & c & & b & a & 0 & c & & & \\ & & & c & & 0 & a & b & & & \\ & & & & c & & b & a & b & & \\ & & & & & c & & b & a & & \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \\ u_8 \\ u_9 \end{bmatrix} = \begin{bmatrix} f_1 - cu_s - bu_o \\ f_2 - cu_s \\ f_3 - cu_s - bu_L \\ f_4 - bu_o \\ f_5 \\ f_6 - bu_L \\ f_7 - bu_o - cu_N \\ f_8 - cu_N \\ f_9 - cu_N - bu_L \end{bmatrix}$$

*A é Pentadiagonal*



# MEF: exemplo de solução para a equação 1D

Formulação Forte

$$\begin{cases} u'' = f & \text{em } (0,1) \\ u(0) = u(1) = 0 \end{cases}$$

$$\int_0^1 u''(x)v(x)dx = \int_0^1 f(x)v(x)dx$$

$$u'(x)v(x)\Big|_0^1 - \int_0^1 u'(x)v'(x)dx = \int_0^1 f(x)v(x)dx$$

Formulação Variacional ou Formulação Fraca

$$\text{Encontrar } u \in V \text{ tal que } \forall v \in V, -\int_0^1 u'v' dx = \int_0^1 fvdx$$

# MEF: exemplo de solução para a equação 1D

## Formulação Variacional Aproximada

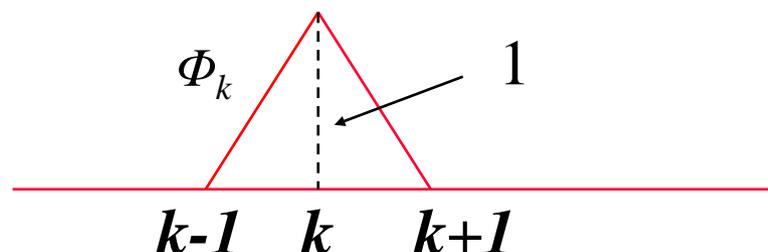
$$0 = x_0 \leq x_1 \leq \dots \leq x_{i-1} \leq x_i \leq x_{i+1} \leq \dots \leq x_{n+1} = 1$$

$$x_i = x_0 + i\Delta x$$

$$V = \{u : [0,1] \rightarrow \mathbb{R}, u \text{ é contínua, } u|_{[x_k, x_{k+1}]} \text{ é linear, } k = 0, \dots, n \text{ e } u(0) = u(1) = 0\}$$

$$\phi(x) = \begin{cases} \frac{x - x_{k-1}}{x_k - x_{k-1}} & \text{se } x \in [x_{k-1}, x_k] \\ \frac{x - x_{k+1}}{x_k - x_{k+1}} & \text{se } x \in [x_k, x_{k+1}] \\ 0 & \text{se } x \notin [x_{k-1}, x_{k+1}] \end{cases}$$

Escolher  $\Phi_k$  com suporte compacto, isto é:



# MEF: exemplo de solução para a equação 1D

$$\text{Encontrar } u \in V \text{ tal que } \forall v \in V, -\int_0^1 u'v' dx = \int_0^1 fvdx$$

$$u^h(x) = \sum_{k=1}^n u_k \phi_k(x) \quad f^h(x) = \sum_{k=1}^n f_k \phi_k(x)$$

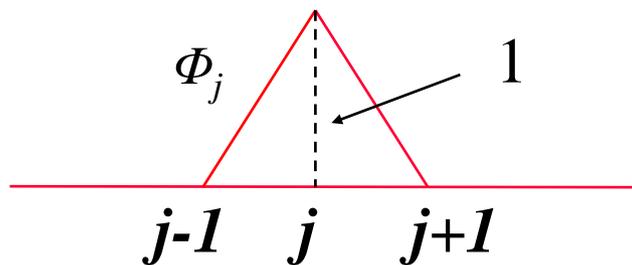
$$\text{Encontrar } u^h \in V^h \text{ tal que } \forall v^h \in V^h, -\int_0^1 u^{h'}v^{h'} dx = \int_0^1 f^h v^h dx$$

$$-\int_0^1 \sum_{j=1}^n (u_j \phi_j'(x)) \phi_i'(x) dx = \int_0^1 f^h \phi_i(x) dx \quad \forall \phi_i, i = 1, \dots, n$$

$$\sum_{j=1}^n \left( -\int_0^1 \phi_j'(x) \phi_i'(x) dx \right) u_j = \int_0^1 f^h \phi_i(x) dx \quad \forall \phi_i, i = 1, \dots, n$$

# MEF: exemplo de solução para a equação 1D

Como  $\Phi_j$  possui suporte compacto,  $\Phi_j'(x)$  é diferente de zero somente para  $i = j-1, j$  e  $j+1$



$$Au = b$$

$A$  é tridiagonal

# MEF: exemplo de solução para a equação 1D

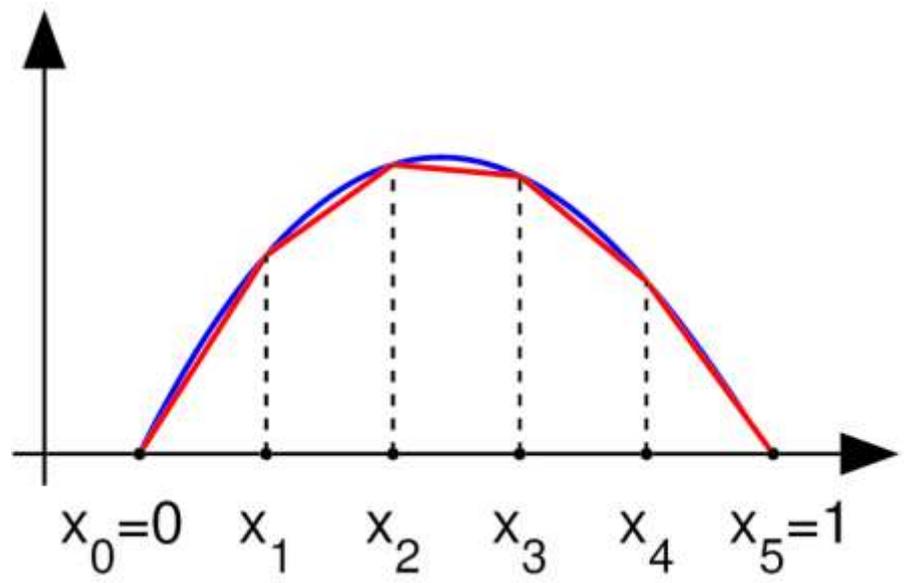
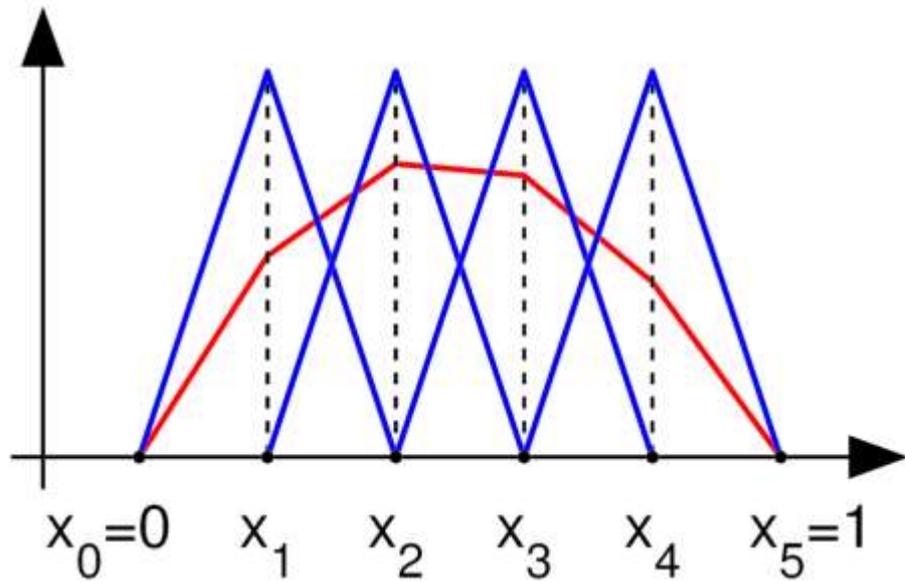
$$A = \sum_{e=1}^{Nel} \mathbf{A}^e k^e$$

$$b = \sum_{e=1}^{Nel} \mathbf{A}^e f^e$$

$$Au = b$$

$$k^e = \begin{bmatrix} k_{11}^e & k_{12}^e \\ k_{21}^e & k_{22}^e \end{bmatrix}$$

$$u^h(x) = \sum_{k=1}^n u_k \phi_k(x)$$



# MEF: exemplo de solução para equação 2D

Equação diferencial contínua

$$\begin{cases} \nabla^2 u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f & \text{em } (0,1) \times (0,1) \subset \square^2 \\ u = 0 & \text{em } \Gamma \end{cases}$$

Formulação Variacional ou Formulação Fraca

$$\text{Encontrar } u \in V \text{ tal que } \forall v \in V, \quad - \int_{\Omega} \nabla u \cdot \nabla v \, d\Omega = \int_{\Omega} f v \, d\Omega$$

# MEF: exemplo de solução para equação 2D

$$u^h(x, y) = \sum_{k=1}^n u_k \phi_k(x, y) \quad f^h(x, y) = \sum_{k=1}^n f_k \phi_k(x, y)$$

Formulação Variacional Aproximada

$$\text{Encontrar } u^h \in V^h \text{ tal que } \forall v^h \in V^h, -\int_{\Omega} \nabla u^h \cdot \nabla v^h d\Omega = \int_{\Omega} f v^h d\Omega$$

$$-\int_{\Omega} \sum_{j=1}^n (u_j \nabla \phi_j) \cdot \nabla \phi_i d\Omega = \int_{\Omega} f^h \phi_i d\Omega \quad \forall \phi_i, i = 1, \dots, n$$

$$Au = b$$

$$\sum_{j=1}^n \left( -\int_{\Omega} \nabla \phi_j \cdot \nabla \phi_i d\Omega \right) u_j = \int_{\Omega} f^h \phi_i d\Omega \quad \forall \phi_i, i = 1, \dots, n$$

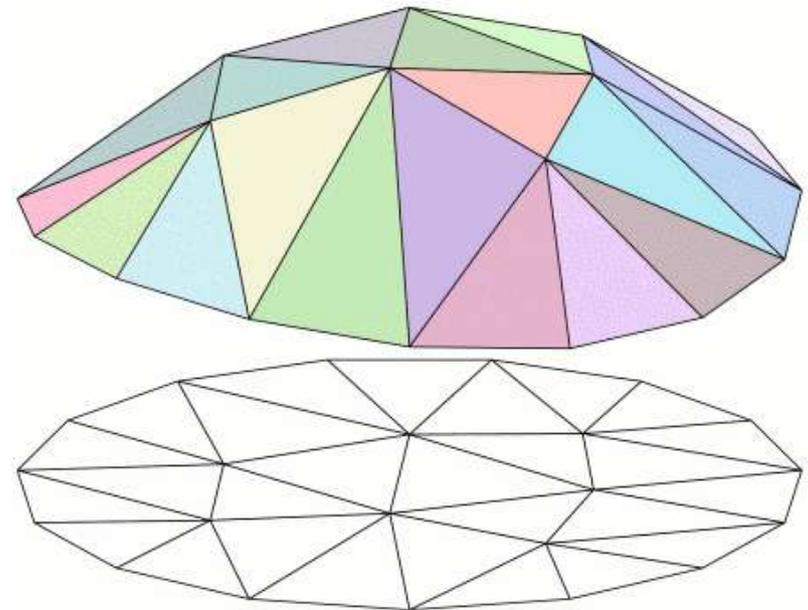
*A* é esparsa

# MEF: exemplo de solução para equação 2D

$$A = \sum_{e=1}^{Nel} \mathbf{A} k^e$$

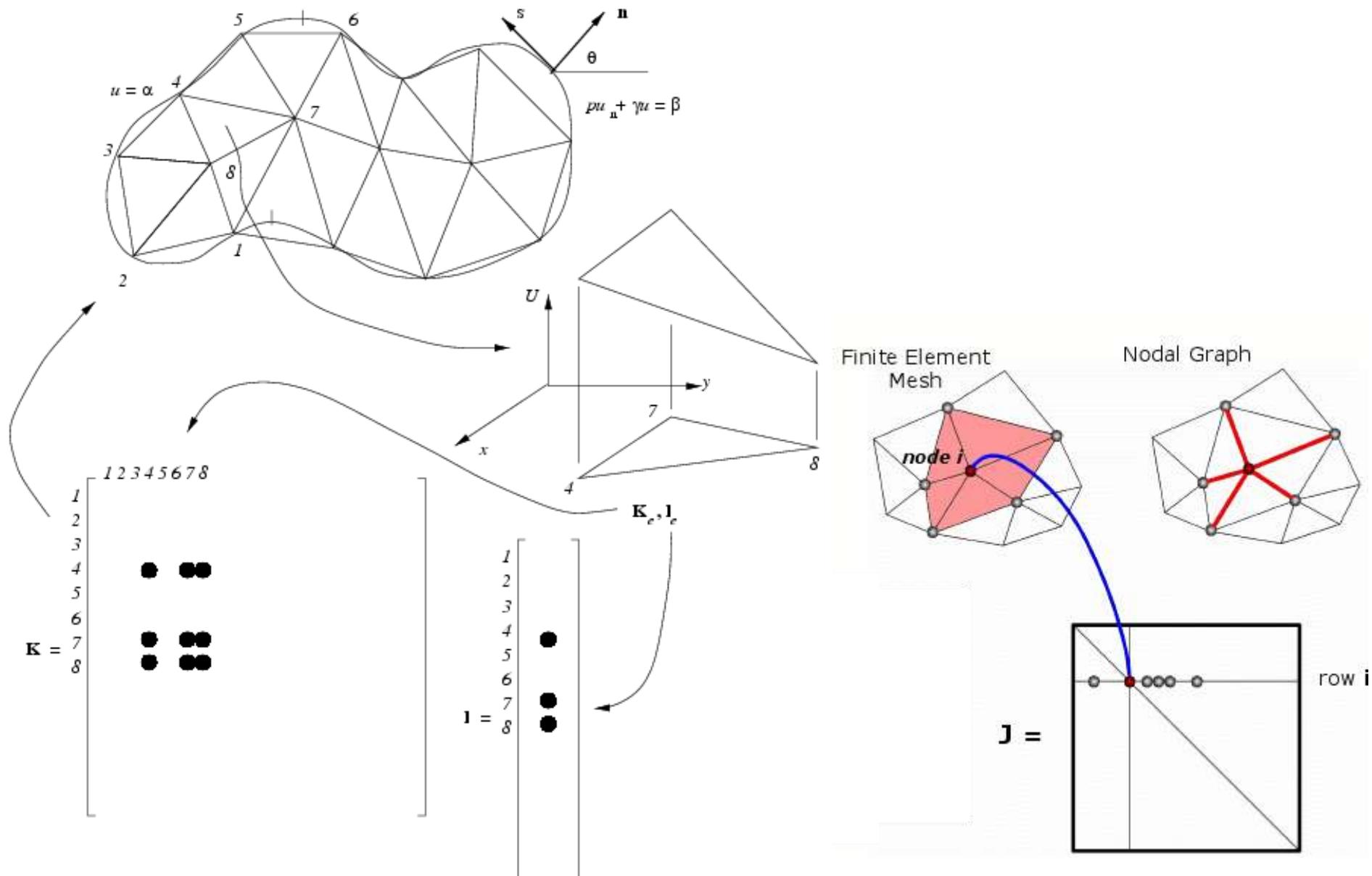
$$b = \sum_{e=1}^{Nel} \mathbf{A} f^e$$

$$k^e = \begin{bmatrix} k_{11} & k_{12} & \cdots & k_{1,nen} \\ k_{21} & k_{22} & \cdots & k_{2,nen} \\ \vdots & \vdots & \ddots & \vdots \\ k_{nen1} & k_{nen2} & \cdots & k_{nen,nen} \end{bmatrix}$$



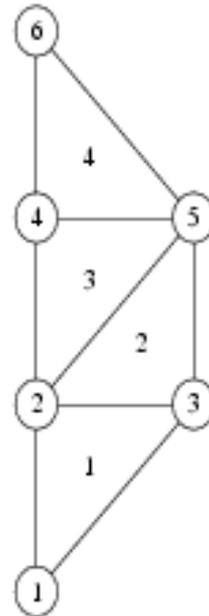
$$u^e = \sum_{k=1}^{nen} u_k \varphi_k$$

# MEF: Montagem da Matriz Esparsa

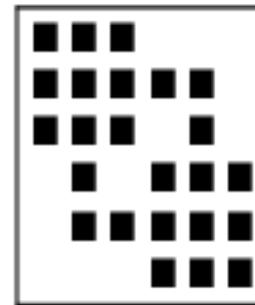


# MEF: Montagem da Matriz Esparsa

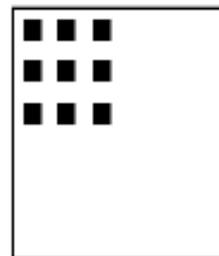
Finite element mesh



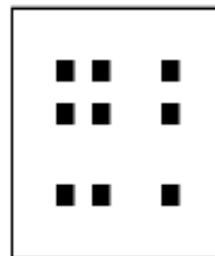
Assembled matrix



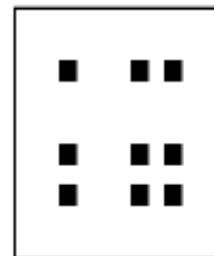
$A^{[1]}$



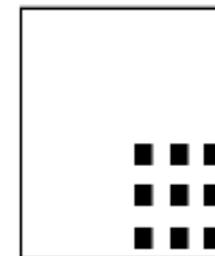
$A^{[2]}$



$A^{[3]}$

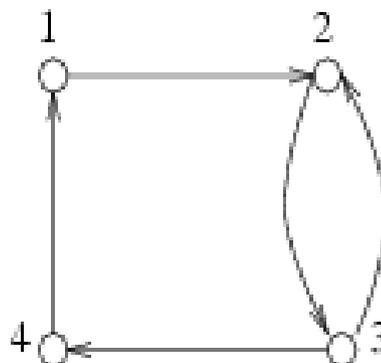
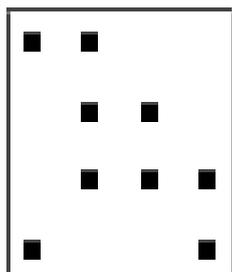


$A^{[4]}$

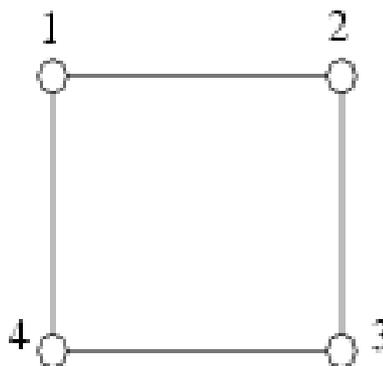
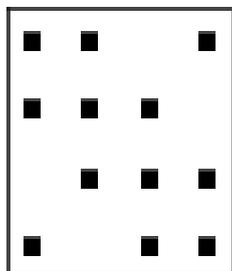




# Matrizes Esparsas x Grafo Associado



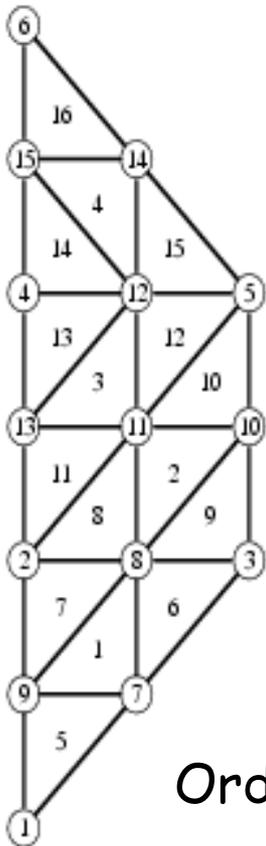
Grafo Ordenado



Grafo não-ordenado

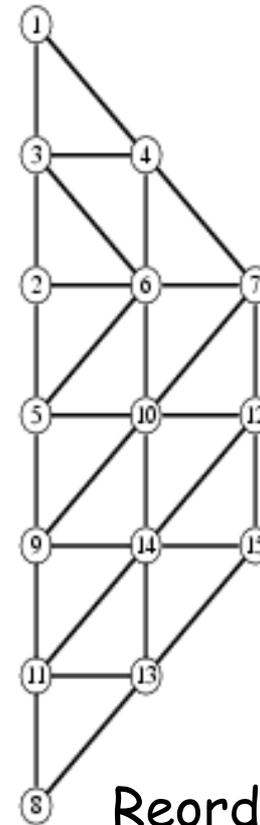
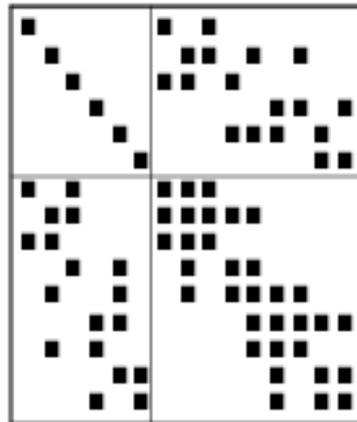
# Matrizes Esparsas Resultantes x Reordenamento de Grafos

Finite element mesh

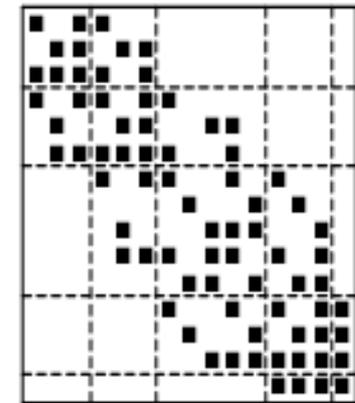


Ordenamento Natural

Assembled matrix

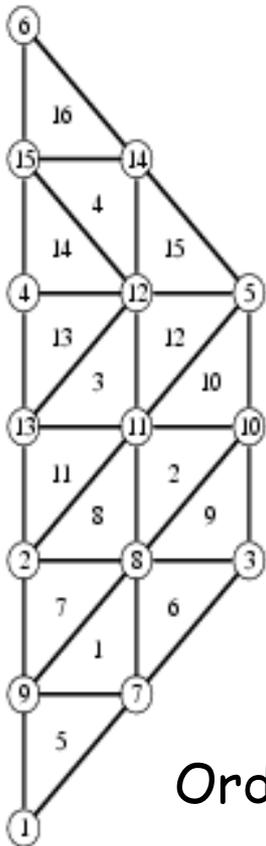


Reordenamento Cuthill-McKee



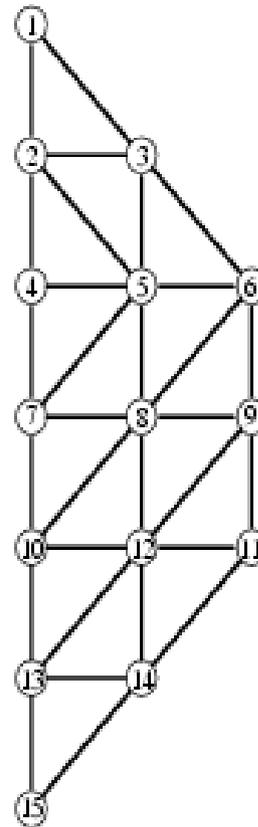
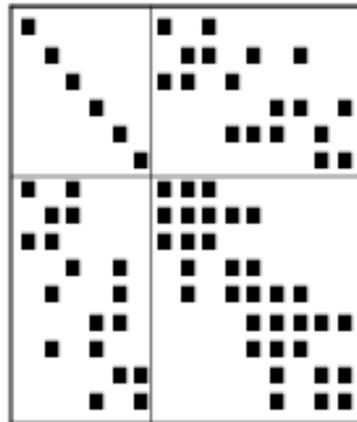
# Matrizes Esparsas Resultantes x Reordenamento de Grafos

Finite element mesh

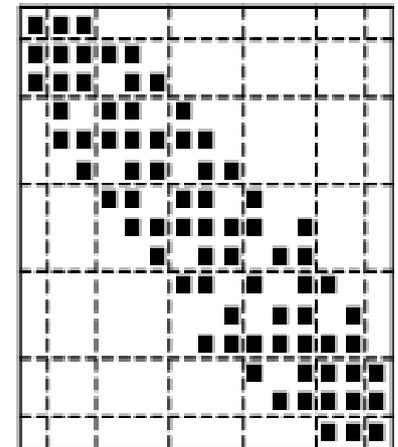


Ordenamento Natural

Assembled matrix



Reordenamento Reverse Cuthill-McKee





# Matrizes Esparsas Resultantes

- Matrizes esparsas x Solução de sistemas Lineares
- Armazenamentos Globais
- Armazenamentos Locais

# Matrizes Esparsas x Métodos de Solução

- Métodos diretos:

- Solução exata a menos de erros de arredondamento. Transformação do sistema em sistemas triviais modificando os coeficientes da matriz esparsa e alterando a esparsidade.

- Métodos Iterativos:

- Solução aproximada com tolerância pré-fixada. Não há alteração dos coeficientes nem da esparsidade da matriz
- Dependem de condições de convergência
- Necessidade do produto matriz-vetor









# Armazenamento de Matrizes Esparsas

## Estratégias Globais

### Algoritmo Produto Matriz-vetor CSR

```
para i=1,2,...,n
    k1 = IA(i)
    k2 = IA(i+1)-1
    para j = k1,..., K2
        y(i) = y(i) + AA(j)*v(JA(j))
    fim_para ! j
Fim_para ! i
```

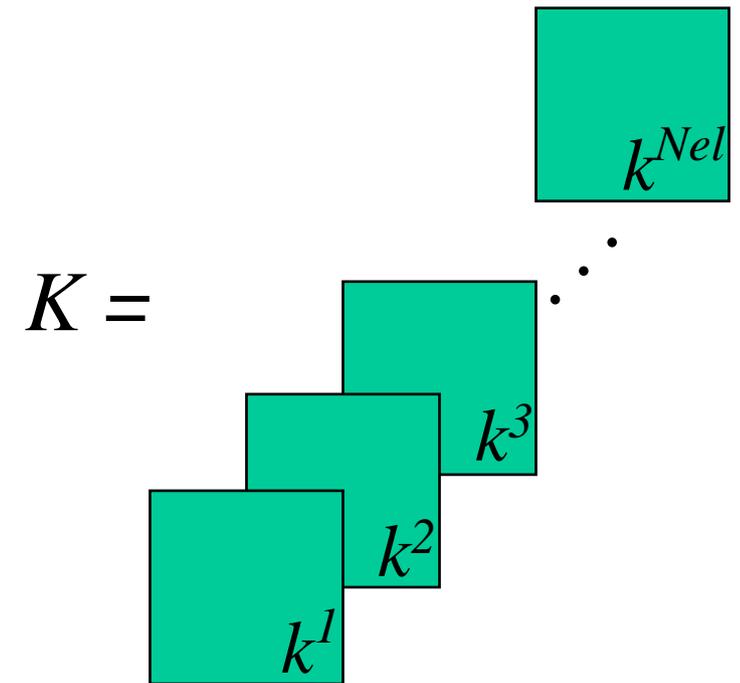
# Armazenamento de Matrizes Esparsas

## Estratégias Locais

### Elemento por Elemento (EBE)

$$K = \sum_{e=1}^{nel} \mathbf{A} k^e$$

$$k^e = \begin{bmatrix} k_{11} & k_{12} & \cdots & k_{1,nen} \\ k_{21} & k_{22} & \cdots & k_{2,nen} \\ \vdots & \vdots & \ddots & \vdots \\ k_{nen1} & k_{nen2} & \cdots & k_{nen,nen} \end{bmatrix}$$





# Armazenamento de Matrizes Esparsas

## Estratégias Locais

### Algoritmo Produto Matriz-Vetor EBE

$$\mathbf{A}\mathbf{v} = \left( \mathbf{A} \prod_{e=1}^{nel} A^e \right) \mathbf{v} = \mathbf{A} \prod_{e=1}^{nel} A^e \mathbf{v}^e$$

```
para e=1,2,...,nel
  localize: ve ← v(e)
  produto: ave ← ke*ve
  espalhe e acumule: v(e) ← v(e) + ave
fim_para ! e
```

# Armazenamento de Matrizes Esparsas Estratégias Locais

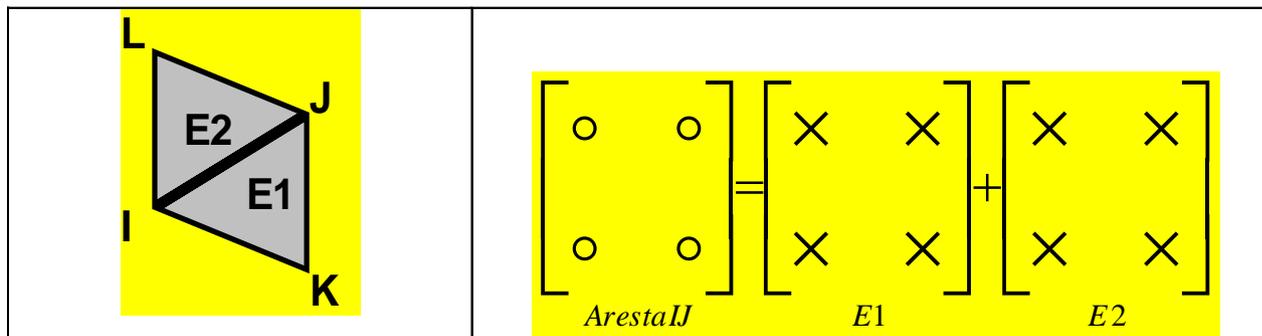
## Aresta por Aresta (EDS)

- Desmembramento da matriz elemento nas componentes das arestas

$$\begin{bmatrix} \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet \end{bmatrix} = \begin{bmatrix} \times & \times & 0 \\ \times & \times & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & \times & \times \\ 0 & \times & \times \end{bmatrix} + \begin{bmatrix} \times & 0 & \times \\ 0 & 0 & 0 \\ \times & 0 & \times \end{bmatrix}$$

*Elemento*
*Aresta1*
*Aresta2*
*Aresta3*

- Soma das contribuições das arestas dos elementos adjacentes



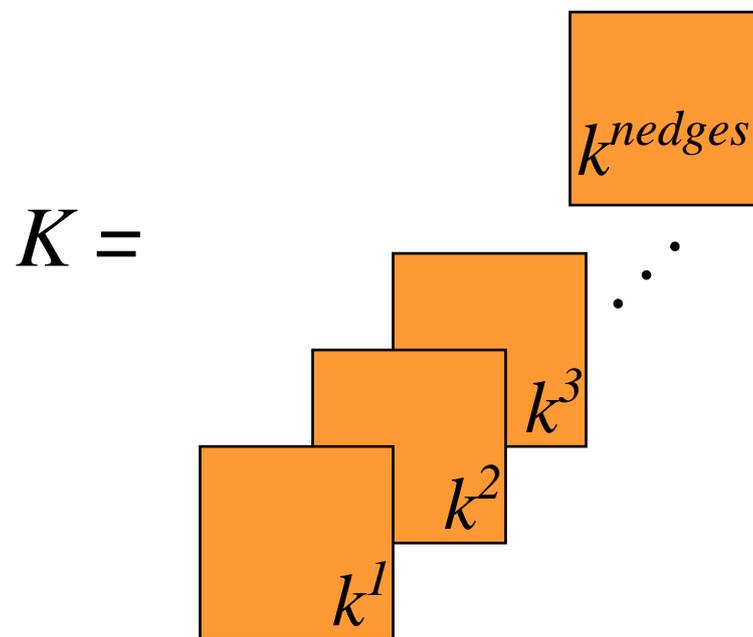
# Armazenamento de Matrizes Esparsas

## Estratégias Locais

### Aresta por Aresta (EDS)

$$K = \sum_{s=1}^{nedges} \mathbf{A} k^s$$

$$k^s = \begin{bmatrix} k_{11} & k_{12} & \cdots & k_{1,ned} \\ k_{21} & k_{22} & \cdots & k_{2,ned} \\ \vdots & \vdots & \ddots & \vdots \\ k_{ned1} & k_{ned2} & \cdots & k_{ned,ned} \end{bmatrix}$$





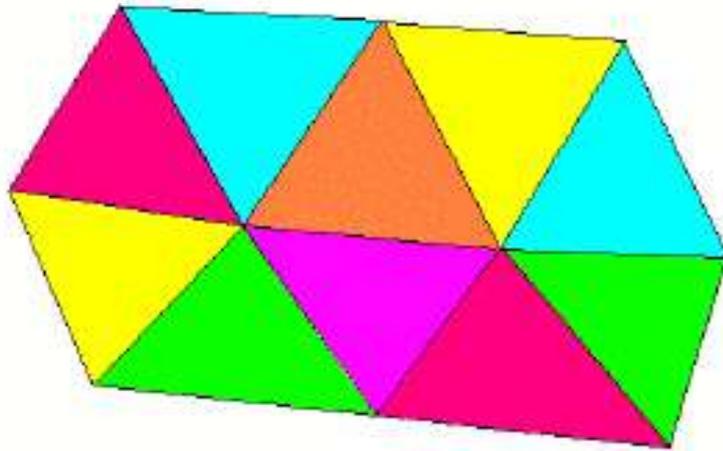
# Armazenamento de Matrizes Esparsas Estratégias Locais

## Algoritmo Produto Matriz-Vetor EDS

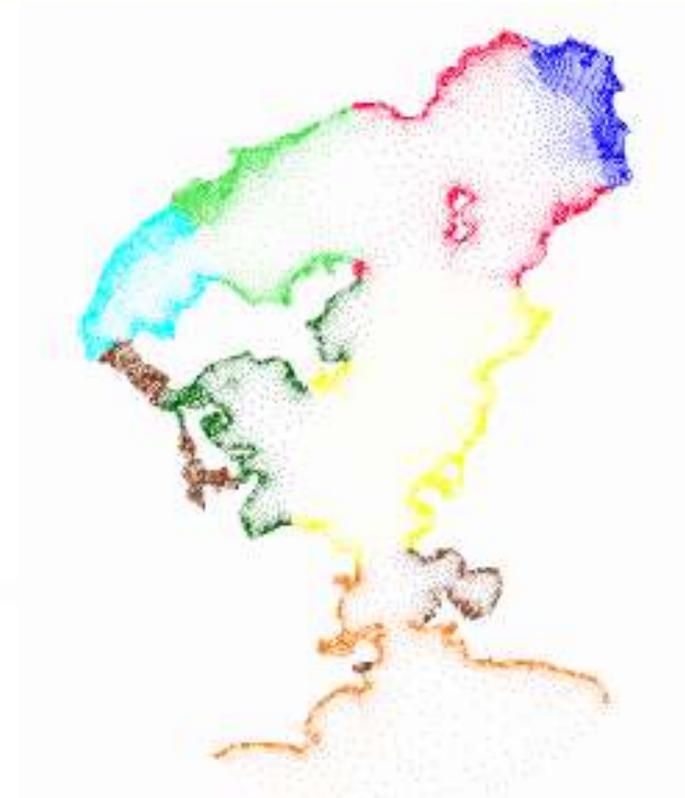
$$\mathbf{A}\mathbf{v} = \left( \underset{e=1}{\overset{nedges}{\mathbf{A}}} \ A^e \right) \mathbf{v} = \underset{e=1}{\overset{nedges}{\mathbf{A}}} k^e \mathbf{v}^e$$

```
para s=1,2,...,nedges
  localize: vs ← v(s)
  produto: avs ← ks*vs
  espalhe e acumule: v(s) ← v(s) + avs
fim_para ! s
```

# Processamento Paralelo



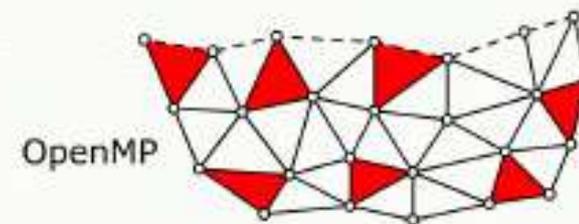
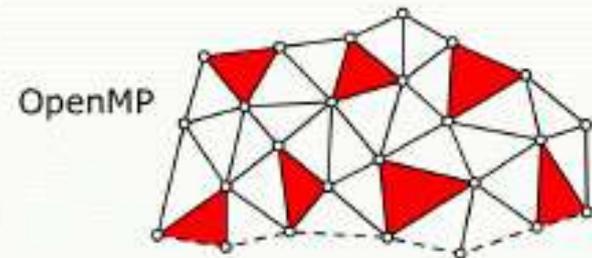
Shared Memory: Mesh Coloring



Distributed Memory: Mesh Partitioning

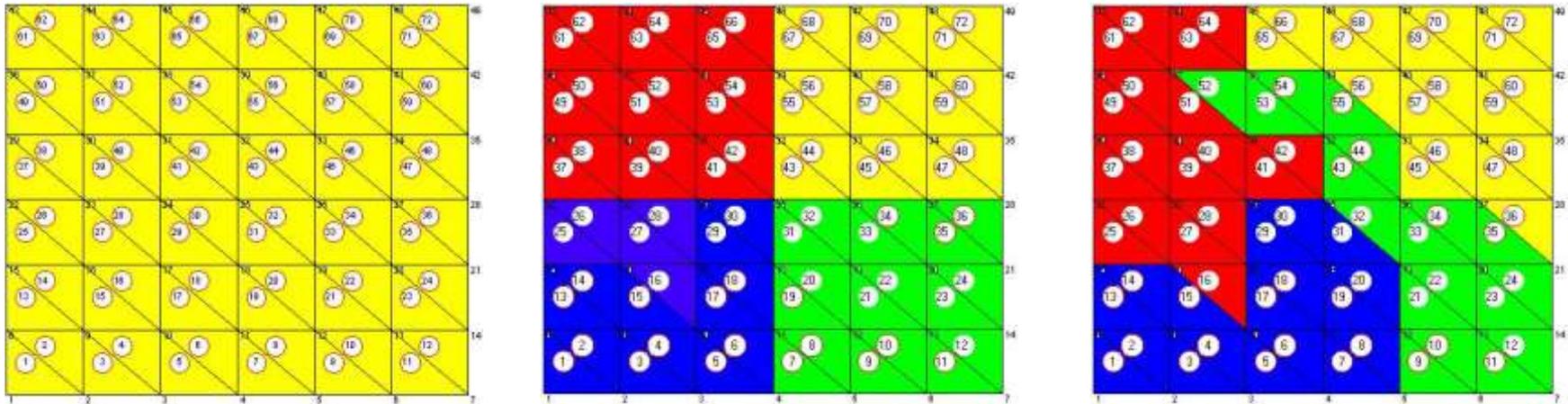
# Armazenamento Local x Coloração de Grafos

```
iside = 0
DO iblk = 1, nedblk
nvec = iedblk(iblk)
!dir$ ivdep
!$OMP PARALLEL DO
    DO i = iside+1, iside+nvec, 1
        ...MATVEC computations...
    ENDDO
!$OMP END PARALLEL DO
iside = iside + nvec
ENDDO
...over interface nodes...
#ifdef MPICODE
call MPI_AllReduce
#endif
```



Necessitamos de um Algoritmo de Coloração de Grafos !!  
Atualmente é usado um algoritmo "Guloso"

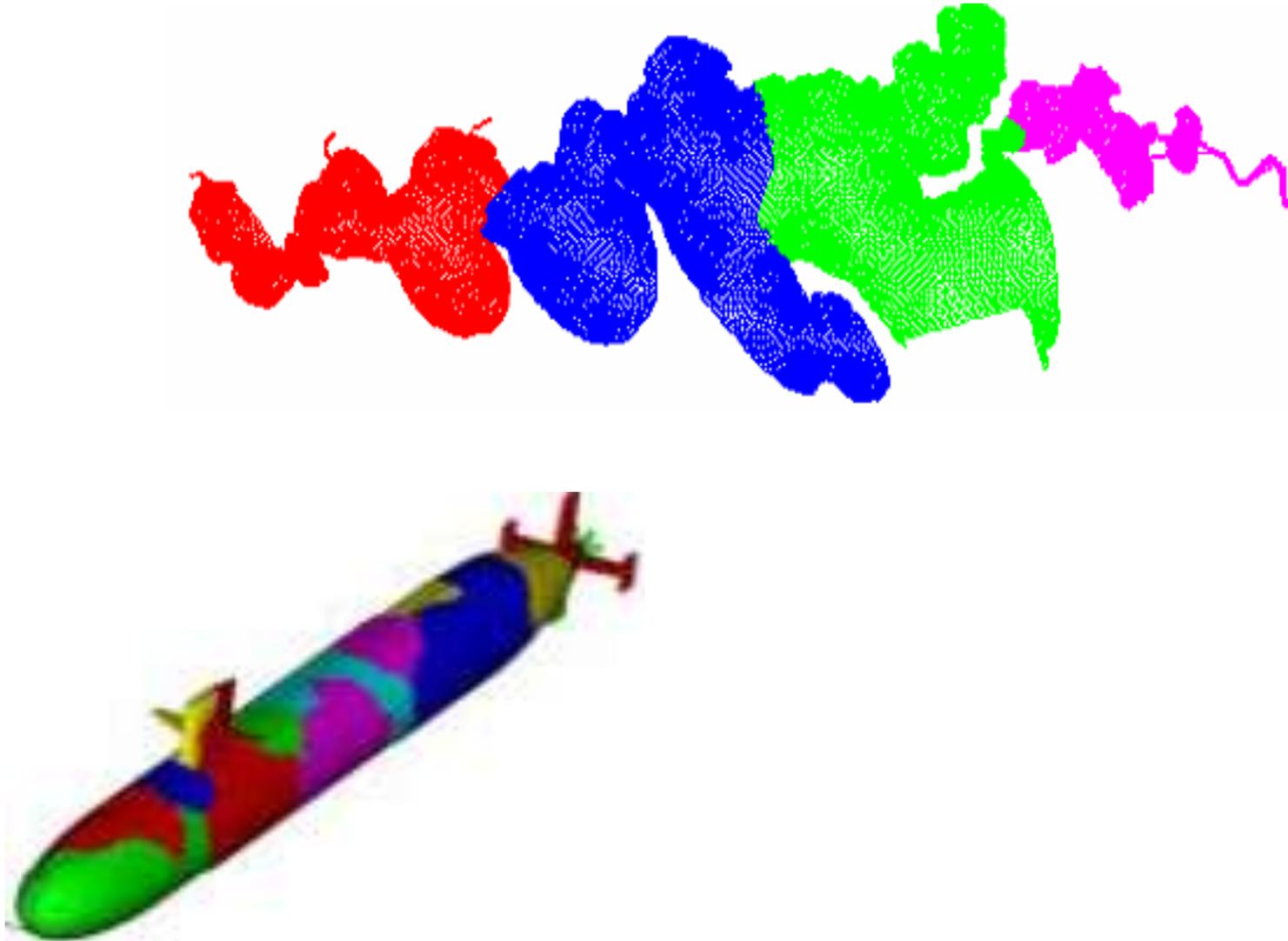
# Partição de Grafos



## Algoritmos de Partição de Grafos

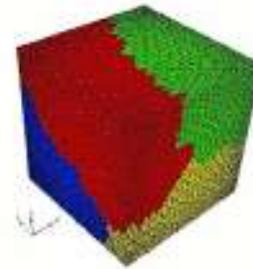
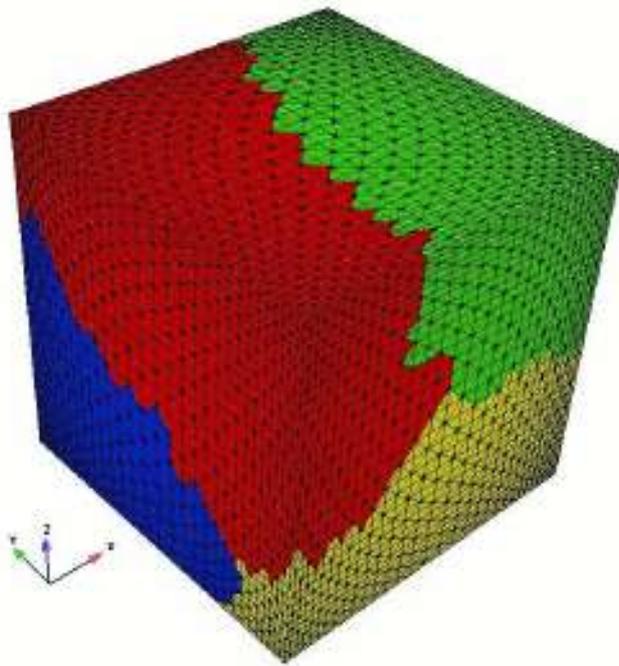
Metis ([http://people.sc.fsu.edu/~burkardt/c\\_src/metis/metis.html](http://people.sc.fsu.edu/~burkardt/c_src/metis/metis.html)): família de particionamento de grafos não estruturados considerando redução de banda da matriz associada ([www.cs.umn.edu/~metis](http://www.cs.umn.edu/~metis))

# Processamento Paralelo



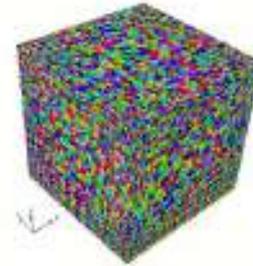
# Processamento Paralelo x Partição de Grafos

- How do we work with data partitioning and memory dependency?



## Data partitioning (Metis)

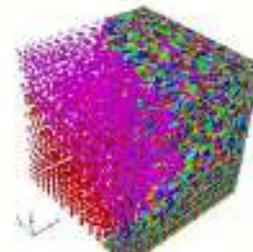
Distributed memory (MPI)



## Mesh coloring

Vectorization (ivdep) and/or  
shared memory parallelism  
(OpenMP)

Pipelines in Itanium-2 (ivdep)



## Data partitioning + Mesh coloring and vectorization