Universidade Federal do Paraná
Programa de Pós-Graduação em Geologia

GEOL7048: Tópicos Especiais em Geologia Exploratória II

Métodos semiquantitativos

Saulo P. Oliveira

Departamento de Matemática, Universidade Federal do Paraná

Aula 25

- Deconvolução de Euler 3D

Deconvolução de Euler

$$x_0 \frac{\partial T}{\partial x} + z_0 \frac{\partial T}{\partial z} + NB = x \frac{\partial T}{\partial x} + NT$$

Equação 2D (Thompson, 1982)

$$x_0 \frac{\partial T}{\partial x} + y_0 \frac{\partial T}{\partial y} + z_0 \frac{\partial T}{\partial z} + NB = x \frac{\partial T}{\partial x} + y \frac{\partial T}{\partial y} + z \frac{\partial T}{\partial z} + NT$$

Equação 3D (Reid et al, 1990)

- (x_0, y_0, z_0): localização da fonte causadora da anomalia
- N: índice estrutural
- B: campo regional

Deconvolução de Euler

$$\boxed{x_0 \frac{\partial T}{\partial x} + z_0 \frac{\partial T}{\partial z} + NB = x \frac{\partial T}{\partial x} + NT}$$

Equação 2D (Thompson, 1982)

$$\boxed{x_0 \frac{\partial T}{\partial x} + y_0 \frac{\partial T}{\partial y} + z_0 \frac{\partial T}{\partial z} + NB = x \frac{\partial T}{\partial x} + y \frac{\partial T}{\partial y} + z \frac{\partial T}{\partial z} + NT}$$

Equação 3D (Reid et al, 1990)

- (x_0, y_0, z_0): localização da fonte causadora da anomalia
- N: índice estrutural
- B: campo regional

OBS: Thompson (1982) não leva em conta a altura de voo

Altura de voo

Euler 2D: para a altura de voo constante $z = -H$,

$$x_0 \frac{\partial T}{\partial x} + z_0 \frac{\partial T}{\partial z} + NB = x \frac{\partial T}{\partial x} + z \frac{\partial T}{\partial z} + NT$$

Altura de voo

Euler 2D: para a altura de voo constante $z = -H$,

$$x_0 \frac{\partial T}{\partial x} + z_0 \frac{\partial T}{\partial z} + NB = x \frac{\partial T}{\partial x} - H \frac{\partial T}{\partial z} + NT$$

Altura de voo

Euler 2D: para a altura de voo constante $z = -H$,

$$x_0 \frac{\partial T}{\partial x} + z_0 \frac{\partial T}{\partial z} + NB = x \frac{\partial T}{\partial x} - H \frac{\partial T}{\partial z} + NT$$

Definindo $\tilde{z}_0$ tal que $z_0 = \tilde{z}_0 - H$, temos:

$$x_0 \frac{\partial T}{\partial x} + (\tilde{z}_0 - H) \frac{\partial T}{\partial z} + NB = x \frac{\partial T}{\partial x} - H \frac{\partial T}{\partial z} + NT$$

Altura de voo

Euler 2D: para a altura de voo constante $z = -H$,

$$x_0 \frac{\partial T}{\partial x} + z_0 \frac{\partial T}{\partial z} + NB = x \frac{\partial T}{\partial x} - H \frac{\partial T}{\partial z} + NT$$

Definindo $\tilde{z}_0$ tal que $z_0 = \tilde{z}_0 - H$, temos:

$$x_0 \frac{\partial T}{\partial x} + \tilde{z}_0 \frac{\partial T}{\partial z} - H \frac{\partial T}{\partial z} + NB = x \frac{\partial T}{\partial x} - H \frac{\partial T}{\partial z} + NT$$

Altura de voo

Euler 2D: para a altura de voo constante $z = -H$,

$$x_0 \frac{\partial T}{\partial x} + z_0 \frac{\partial T}{\partial z} + NB = x \frac{\partial T}{\partial x} - H \frac{\partial T}{\partial z} + NT$$

Definindo $\tilde{z}_0$ tal que $z_0 = \tilde{z}_0 - H$, temos:

$$x_0 \frac{\partial T}{\partial x} + \tilde{z}_0 \frac{\partial T}{\partial z} + NB = x \frac{\partial T}{\partial x} + NT$$

Altura de voo

Euler 2D: para a altura de voo constante $z = -H$,

$$x_0 \frac{\partial T}{\partial x} + z_0 \frac{\partial T}{\partial z} + NB = x \frac{\partial T}{\partial x} - H \frac{\partial T}{\partial z} + NT$$

Definindo $\tilde{z}_0$ tal que $z_0 = \tilde{z}_0 - H$, temos:

$$x_0 \frac{\partial T}{\partial x} + \tilde{z}_0 \frac{\partial T}{\partial z} + NB = x \frac{\partial T}{\partial x} + NT$$

Analogamente, para Euler 3D,

$$x_0 \frac{\partial T}{\partial x} + y_0 \frac{\partial T}{\partial y} + z_0 \frac{\partial T}{\partial z} + NB = x \frac{\partial T}{\partial x} + y \frac{\partial T}{\partial y} - H \frac{\partial T}{\partial z} + NT$$

Altura de voo

Euler 2D: para a altura de voo constante $z = -H$,

$$x_0 \frac{\partial T}{\partial x} + z_0 \frac{\partial T}{\partial z} + NB = x \frac{\partial T}{\partial x} - H \frac{\partial T}{\partial z} + NT$$

Definindo $\tilde{z}_0$ tal que $z_0 = \tilde{z}_0 - H$, temos:

$$x_0 \frac{\partial T}{\partial x} + \tilde{z}_0 \frac{\partial T}{\partial z} + NB = x \frac{\partial T}{\partial x} + NT$$

Analogamente, para Euler 3D,

$$x_0 \frac{\partial T}{\partial x} + y_0 \frac{\partial T}{\partial y} + (\tilde{z}_0 - H) \frac{\partial T}{\partial z} + NB = x \frac{\partial T}{\partial x} + y \frac{\partial T}{\partial y} - H \frac{\partial T}{\partial z} + NT$$

Altura de voo

Euler 2D: para a altura de voo constante $z = -H$,

$$x_0 \frac{\partial T}{\partial x} + z_0 \frac{\partial T}{\partial z} + NB = x \frac{\partial T}{\partial x} - H \frac{\partial T}{\partial z} + NT$$

Definindo $\tilde{z}_0$ tal que $z_0 = \tilde{z}_0 - H$, temos:

$$x_0 \frac{\partial T}{\partial x} + \tilde{z}_0 \frac{\partial T}{\partial z} + NB = x \frac{\partial T}{\partial x} + NT$$

Analogamente, para Euler 3D,

$$x_0 \frac{\partial T}{\partial x} + y_0 \frac{\partial T}{\partial y} + \tilde{z}_0 \frac{\partial T}{\partial z} - H \frac{\partial T}{\partial z} + NB = x \frac{\partial T}{\partial x} + y \frac{\partial T}{\partial y} - H \frac{\partial T}{\partial z} + NT$$

Altura de voo

Euler 2D: para a altura de voo constante $z = -H$,

$$x_0 \frac{\partial T}{\partial x} + z_0 \frac{\partial T}{\partial z} + NB = x \frac{\partial T}{\partial x} - H \frac{\partial T}{\partial z} + NT$$

Definindo $\tilde{z}_0$ tal que $z_0 = \tilde{z}_0 - H$, temos:

$$x_0 \frac{\partial T}{\partial x} + \tilde{z}_0 \frac{\partial T}{\partial z} + NB = x \frac{\partial T}{\partial x} + NT$$

Analogamente, para Euler 3D,

$$x_0 \frac{\partial T}{\partial x} + y_0 \frac{\partial T}{\partial y} + \tilde{z}_0 \frac{\partial T}{\partial z} + NB = x \frac{\partial T}{\partial x} + y \frac{\partial T}{\partial y} + NT$$

Conclusão: basta resolver para $\tilde{z}_0$ e calcular $z_0 = \tilde{z}_0 - H$.

Altura de voo

Euler 2D: para a altura de voo constante $z = -H$,

$$x_0 \frac{\partial T}{\partial x} + z_0 \frac{\partial T}{\partial z} + NB = x \frac{\partial T}{\partial x} - H \frac{\partial T}{\partial z} + NT$$

Definindo $\tilde{z}_0$ tal que $z_0 = \tilde{z}_0 - H$, temos:

$$x_0 \frac{\partial T}{\partial x} + \tilde{z}_0 \frac{\partial T}{\partial z} + NB = x \frac{\partial T}{\partial x} + NT$$

Analogamente, para Euler 3D,

$$x_0 \frac{\partial T}{\partial x} + y_0 \frac{\partial T}{\partial y} + \tilde{z}_0 \frac{\partial T}{\partial z} + NB = x \frac{\partial T}{\partial x} + y \frac{\partial T}{\partial y} + NT$$

Conclusão: basta resolver para $\tilde{z}_0$ e calcular $z_0 = \tilde{z}_0 - H$.
(daqui em diante vamos denotar $\tilde{z}_0$ por z_0)

Deconvolução de Euler 3D

Janela móvel de de $M = M_x \times M_y$ pontos (x_i, y_i, z), com $z = -H$:

$$
\begin{cases}
x_0 \, \frac{\partial T}{\partial x}(x_1, y_1, z) + y_0 \, \frac{\partial T}{\partial y}(x_1, y_1, z) + z_0 \frac{\partial T}{\partial z}(x_1, y_2, z) + NB = \\
x_1 \, \frac{\partial T}{\partial x}(x_1, y_1, z) + y_1 \, \frac{\partial T}{\partial y}(x_1, y_1, z) + NT(x_1, y_1, z) \\
x_0 \, \frac{\partial T}{\partial x}(x_2, y_2, z) + y_0 \, \frac{\partial T}{\partial y}(x_2, y_2, z) + z_0 \frac{\partial T}{\partial z}(x_2, y_2, z) + NB = \\
x_2 \, \frac{\partial T}{\partial x}(x_2, y_2, z) + y_2 \, \frac{\partial T}{\partial y}(x_2, y_2, z) + NT(x_2, y_2, z) \\
\vdots \\
x_0 \, \frac{\partial T}{\partial x}(x_M, y_M, z) + y_0 \, \frac{\partial T}{\partial y}(x_M, y_M, z) + z_0 \frac{\partial T}{\partial z}(x_M, y_M, z) + NB = \\
x_M \, \frac{\partial T}{\partial x}(x_M, y_M, z) + y_M \, \frac{\partial T}{\partial y}(x_M, y_M, z) + NT(x_M, y_M, z)
\end{cases}
$$

Deconvolução de Euler 3D

Janela móvel de de $M = M_x \times M_y$ pontos (x_i, y_i, z), com $z = -H$:

$$
\begin{bmatrix}
\frac{\partial T}{\partial x}(x_1, y_1, z) & \frac{\partial T}{\partial y}(x_1, y_1, z) & \frac{\partial T}{\partial z}(x_1, y_1, z) & N \\
\frac{\partial T}{\partial x}(x_2, y_2, z) & \frac{\partial T}{\partial y}(x_2, y_2, z) & \frac{\partial T}{\partial z}(x_2, y_2, z) & N \\
\vdots & \vdots & \vdots & \vdots \\
\frac{\partial T}{\partial x}(x_M, y_M, z) & \frac{\partial T}{\partial y}(x_M, y_M, z) & \frac{\partial T}{\partial z}(x_M, y_M, z) & N
\end{bmatrix}
\begin{bmatrix} x_0 \\ y_0 \\ z_0 \\ B \end{bmatrix}
$$

$$
= \begin{bmatrix}
x_1 \frac{\partial T}{\partial x}(x_1, y_1, z) + y_1 \frac{\partial T}{\partial y}(x_1, y_1, z) + NT(x_1, y_1, z) \\
x_2 \frac{\partial T}{\partial x}(x_2, y_2, z) + y_2 \frac{\partial T}{\partial y}(x_2, y_2, z) + NT(x_2, y_2, z) \\
\vdots \\
x_M \frac{\partial T}{\partial x}(x_M, y_M, z) + y_M \frac{\partial T}{\partial y}(x_M, y_M, z) + NT(x_M, y_M, z)
\end{bmatrix}
$$

Deconvolução de Euler 3D

Janela móvel de de $M = M_x \times M_y$ pontos (x_i, y_i, z), com $z = -H$:

$$A = \begin{bmatrix} \frac{\partial T}{\partial x}(x_1, y_1, z) & \frac{\partial T}{\partial y}(x_1, y_1, z) & \frac{\partial T}{\partial z}(x_1, y_1, z) & N \\ \frac{\partial T}{\partial x}(x_2, y_2, z) & \frac{\partial T}{\partial y}(x_2, y_2, z) & \frac{\partial T}{\partial z}(x_2, y_2, z) & N \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial T}{\partial x}(x_M, y_M, z) & \frac{\partial T}{\partial y}(x_M, y_M, z) & \frac{\partial T}{\partial z}(x_M, y_M, z) & N \end{bmatrix}$$

$$\boldsymbol{b} = \begin{bmatrix} x_1 \frac{\partial T}{\partial x}(x_1, y_1, z) + y_1 \frac{\partial T}{\partial y}(x_1, y_1, z) + NT(x_1, y_1, z) \\ x_2 \frac{\partial T}{\partial x}(x_2, y_2, z) + y_2 \frac{\partial T}{\partial y}(x_2, y_2, z) + NT(x_2, y_2, z) \\ \vdots \\ x_M \frac{\partial T}{\partial x}(x_M, y_M, z) + y_M \frac{\partial T}{\partial y}(x_M, y_M, z) + NT(x_M, y_M, z) \end{bmatrix}$$

Algoritmo

Dados de entrada: `x,y,T,M,N`
Dados de saída: `xo,yo,zo`
Calcule `dxT,dyT,dzT`
Para todas as posições de janela possíveis:

  Construa o sistema linear para os pontos da janela;

  Resolva o sistema linear para `xo,yo,zo,B`;

  Armazene as coordenadas `xo,yo,zo`;

Algoritmo

```
function [xo,yo,zo] = euler3D(x,y,T,M,N)
[dxT,dyT,dzT] = DF_2D(x,y,T);
```

Para todas as posições de janela possíveis:

- Construa o sistema linear para os pontos da janela;
- Resolva o sistema linear para `xo,yo,zo,B`;
- Armazene as coordenadas `xo,yo,zo`;

Algoritmo

```
function [xo,yo,zo] = euler3D(x,y,T,M,N)
[dxT,dyT,dzT] = DF_2D(x,y,T);
```

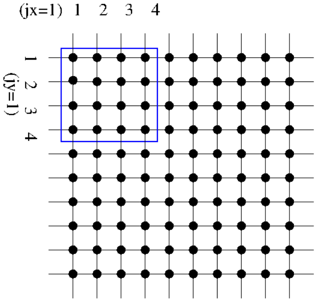
Para todas as posições de janela possíveis:

Construa o sistema linear para os pontos da janela;

Resolva o sistema linear para `xo,yo,zo,B`;

Armazene as coordenadas `xo,yo,zo`;

Posições de janela:
se `length(x)`=10 e
$M_x = M_y = 4$,



Algoritmo

```
function [xo,yo,zo] = euler3D(x,y,T,M,N)
[dxT,dyT,dzT] = DF_2D(x,y,T);
```

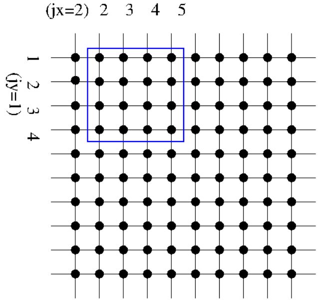
Para todas as posições de janela possíveis:

Construa o sistema linear para os pontos da janela;

Resolva o sistema linear para `xo,yo,zo,B`;

Armazene as coordenadas `xo,yo,zo`;

Posições de janela:
se `length(x)`=10 e
$M_x = M_y = 4$,



Algoritmo

```
function [xo,yo,zo] = euler3D(x,y,T,M,N)
[dxT,dyT,dzT] = DF_2D(x,y,T);
```

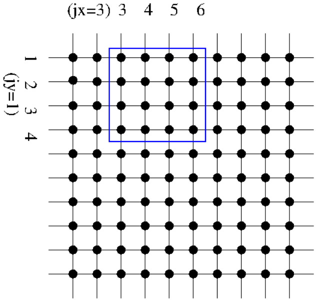
Para todas as posições de janela possíveis:

Construa o sistema linear para os pontos da janela;

Resolva o sistema linear para `xo,yo,zo,B`;

Armazene as coordenadas `xo,yo,zo`;

Posições de janela:
se `length(x)`=10 e
$M_x = M_y = 4$,



Algoritmo

```
function [xo,yo,zo] = euler3D(x,y,T,M,N)
[dxT,dyT,dzT] = DF_2D(x,y,T);
```

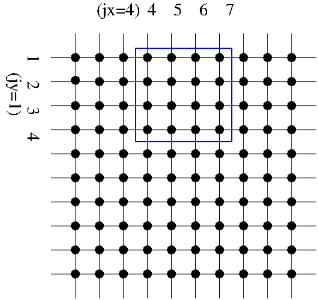
Para todas as posições de janela possíveis:

Construa o sistema linear para os pontos da janela;

Resolva o sistema linear para `xo,yo,zo,B`;

Armazene as coordenadas `xo,yo,zo`;

Posições de janela:
se `length(x)`=10 e
$M_x = M_y = 4$,



Algoritmo

```
function [xo,yo,zo] = euler3D(x,y,T,M,N)
[dxT,dyT,dzT] = DF_2D(x,y,T);
```

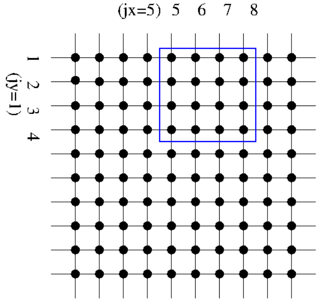
Para todas as posições de janela possíveis:

Construa o sistema linear para os pontos da janela;

Resolva o sistema linear para `xo,yo,zo,B`;

Armazene as coordenadas `xo,yo,zo`;

Posições de janela:
se `length(x)`=10 e
$M_x = M_y = 4$,



Algoritmo

```
function [xo,yo,zo] = euler3D(x,y,T,M,N)
[dxT,dyT,dzT] = DF_2D(x,y,T);
```

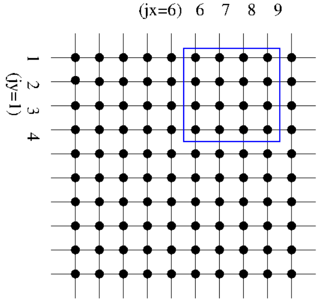
Para todas as posições de janela possíveis:

Construa o sistema linear para os pontos da janela;

Resolva o sistema linear para `xo,yo,zo,B`;

Armazene as coordenadas `xo,yo,zo`;

Posições de janela:
se `length(x)`=10 e
$M_x = M_y = 4$,



Algoritmo

```
function [xo,yo,zo] = euler3D(x,y,T,M,N)
[dxT,dyT,dzT] = DF_2D(x,y,T);
```

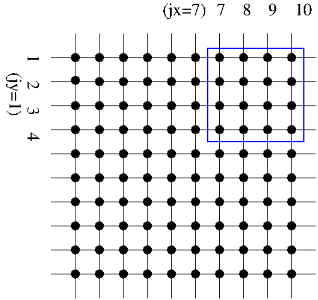
Para todas as posições de janela possíveis:

Construa o sistema linear para os pontos da janela;

Resolva o sistema linear para `xo,yo,zo,B`;

Armazene as coordenadas `xo,yo,zo`;

Posições de janela:
se `length(x)`=10 e
$M_x = M_y = 4$,



Algoritmo

```
function [xo,yo,zo] = euler3D(x,y,T,M,N)
[dxT,dyT,dzT] = DF_2D(x,y,T);
```

Para todas as posições de janela possíveis:

Construa o sistema linear para os pontos da janela;

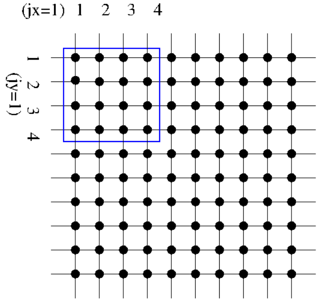
Resolva o sistema linear para `xo,yo,zo,B`;

Armazene as coordenadas `xo,yo,zo`;

Posições de janela:

se `length(x)`=10 e

$M_x = M_y = 4$,



Algoritmo

```
function [xo,yo,zo] = euler3D(x,y,T,M,N)
[dxT,dyT,dzT] = DF_2D(x,y,T);
```

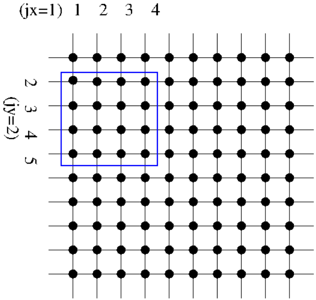
Para todas as posições de janela possíveis:

Construa o sistema linear para os pontos da janela;

Resolva o sistema linear para `xo,yo,zo,B`;

Armazene as coordenadas `xo,yo,zo`;

Posições de janela:
se `length(x)`=10 e
$M_x = M_y = 4,$



Algoritmo

```
function [xo,yo,zo] = euler3D(x,y,T,M,N)
[dxT,dyT,dzT] = DF_2D(x,y,T);
```

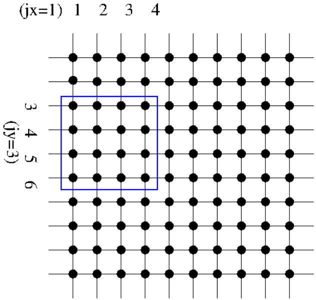
Para todas as posições de janela possíveis:

Construa o sistema linear para os pontos da janela;

Resolva o sistema linear para `xo,yo,zo,B`;

Armazene as coordenadas `xo,yo,zo`;

Posições de janela:
se `length(x)`=10 e
$M_x = M_y = 4$,



Algoritmo

```
function [xo,yo,zo] = euler3D(x,y,T,M,N)
[dxT,dyT,dzT] = DF_2D(x,y,T);
```

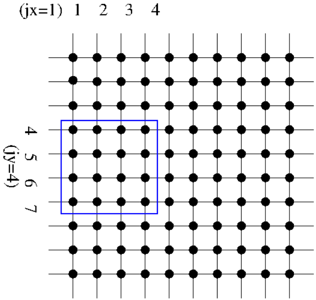
Para todas as posições de janela possíveis:

Construa o sistema linear para os pontos da janela;

Resolva o sistema linear para `xo,yo,zo,B`;

Armazene as coordenadas `xo,yo,zo`;

Posições de janela:
se `length(x)`=10 e
$M_x = M_y = 4$,



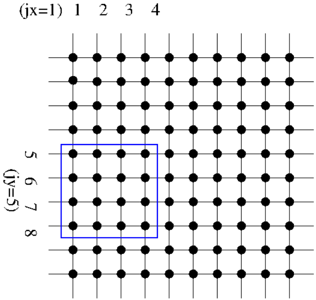
Algoritmo

```
function [xo,yo,zo] = euler3D(x,y,T,M,N)
[dxT,dyT,dzT] = DF_2D(x,y,T);
```

Para todas as posições de janela possíveis:

> Construa o sistema linear para os pontos da janela;
>
> Resolva o sistema linear para `xo,yo,zo,B`;
>
> Armazene as coordenadas `xo,yo,zo`;

Posições de janela:
se `length(x)`=10 e
$M_x = M_y = 4$,



Algoritmo

```
function [xo,yo,zo] = euler3D(x,y,T,M,N)
[dxT,dyT,dzT] = DF_2D(x,y,T);
```

Para todas as posições de janela possíveis:

Construa o sistema linear para os pontos da janela;

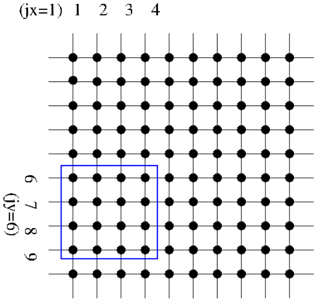
Resolva o sistema linear para `xo,yo,zo,B`;

Armazene as coordenadas `xo,yo,zo`;

Posições de janela:

se `length(x)`=10 e

$M_x = M_y = 4$,



Algoritmo

```
function [xo,yo,zo] = euler3D(x,y,T,M,N)
[dxT,dyT,dzT] = DF_2D(x,y,T);
```

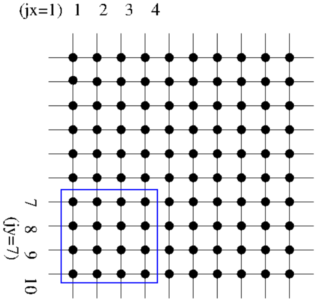
Para todas as posições de janela possíveis:

Construa o sistema linear para os pontos da janela;

Resolva o sistema linear para `xo,yo,zo,B`;

Armazene as coordenadas `xo,yo,zo`;

Posições de janela:
se `length(x)`=10 e
$M_x = M_y = 4$,



Algoritmo

```
function [xo,yo,zo] = euler3D(x,y,T,M,N)
[dxT,dyT,dzT] = DF_2D(x,y,T);
```

Para todas as posições de janela possíveis:

Construa o sistema linear para os pontos da janela;

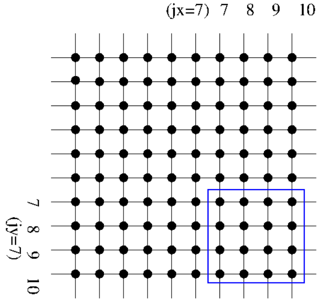
Resolva o sistema linear para `xo,yo,zo,B`;

Armazene as coordenadas `xo,yo,zo`;

Posições de janela:

se `length(x)`=10 e

$M_x = M_y = 4$,

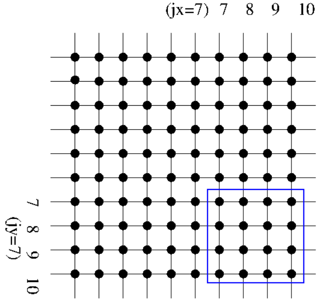


Algoritmo

```
function [xo,yo,zo] = euler3D(x,y,T,M,N)
[dxT,dyT,dzT] = DF_2D(x,y,T);
[ny,nx] = size(x);
for jx = 1:nx-M+1
  for jy = 1:ny-M+1
```

Construa o sistema linear para ($x_{i,j}$,$y_{i,j}$), $\begin{cases} j_y \leq i \leq j_y + M - 1 \\ j_x \leq j \leq j_y + M - 1; \end{cases}$

```
    sol = A\b;
    xo(jy,jx) = sol(1);
    yo(jy,jx) = sol(2);
    zo(jy,jx) = sol(3);
  end
end
```



Detalhes do sistema linear

Sistema linear para $(\mathrm{x}_{i,j},\mathrm{y}_{i,j})$, $\begin{cases} j_y \leq i \leq j_y^F = j_y + M - 1 \\ j_x \leq j \leq j_x^F = j_y + M - 1 \end{cases}$

$$A = \begin{bmatrix} \frac{\partial T}{\partial x}(\boldsymbol{p}_{j_x,j_y}, z) & \frac{\partial T}{\partial y}(\boldsymbol{p}_{j_x,j_y}, z) & \frac{\partial T}{\partial z}(\boldsymbol{p}_{j_x,j_y}, z) & N \\ \frac{\partial T}{\partial x}(\boldsymbol{p}_{j_x+1,j_y}, z) & \frac{\partial T}{\partial y}(\boldsymbol{p}_{j_x+1,j_y}, z) & \frac{\partial T}{\partial z}(\boldsymbol{p}_{j_x+1,j_y}, z) & N \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial T}{\partial x}(\boldsymbol{p}_{j_x^F,j_y^F}, z) & \frac{\partial T}{\partial y}(\boldsymbol{p}_{j_x^F,j_y^F}, z) & \frac{\partial T}{\partial z}(\boldsymbol{p}_{j_x^F,j_y^F}, z) & N \end{bmatrix}$$

$$\boldsymbol{b} = \begin{bmatrix} x_{j_x,j_y}\frac{\partial T}{\partial x}(\boldsymbol{p}_{j_x,j_y}, z) + y_{j_x,j_y}\frac{\partial T}{\partial y}(\boldsymbol{p}_{j_x,j_y}, z) + NT(\boldsymbol{p}_{j_x,j_y}, z) \\ x_{j_x+1,j_y}\frac{\partial T}{\partial x}(\boldsymbol{p}_{j_x+1,j_y}, z) + y_{j_x+1,j_y}\frac{\partial T}{\partial y}(\boldsymbol{p}_{j_x+1,j_y}, z) + NT(\boldsymbol{p}_{j_x+1,j_y}, z) \\ \vdots \\ x_{j_x^F,j_y^F}\frac{\partial T}{\partial x}(\boldsymbol{p}_{j_x^F,j_y^F}, z) + y_{j_x^F,j_y^F}\frac{\partial T}{\partial y}(\boldsymbol{p}_{j_x^F,j_y^F}, z) + NT(\boldsymbol{p}_{j_x^F,j_y^F}, z) \end{bmatrix}$$

Detalhes do sistema linear

Sistema linear para ($\mathtt{x}_{i,j}$,$\mathtt{y}_{i,j}$), $\begin{cases} j_y \leq i \leq j_y^F = j_y + M - 1 \\ j_x \leq j \leq j_x^F = j_y + M - 1 \end{cases}$

$$A = \begin{bmatrix} \mathtt{dxT}_{j_x,j_y} & \mathtt{dyT}_{j_x,j_y} & \mathtt{dzT}_{j_x,j_y} & \mathtt{N} \\ \mathtt{dxT}_{j_x+1,j_y} & \mathtt{dyT}_{j_x+1,j_y} & \mathtt{dzT}_{j_x+1,j_y} & \mathtt{N} \\ \vdots & \vdots & \vdots & \vdots \\ \mathtt{dxT}_{j_x^F,j_y^F} & \mathtt{dyT}_{j_x^F,j_y^F} & \mathtt{dzT}_{j_x^F,j_y^F} & \mathtt{N} \end{bmatrix}$$

$$\boldsymbol{b} = \begin{bmatrix} x_{j_x,j_y}\mathtt{dxT}_{j_x,j_y} + y_{j_x,j_y}\mathtt{dyT}_{j_x,j_y} + \mathtt{NT}_{j_x,j_y} \\ x_{j_x+1,j_y}\mathtt{dxT}_{j_x+1,j_y} + y_{j_x+1,j_y}\mathtt{dyT}_{j_x+1,j_y} + \mathtt{NT}_{j_x+1,j_y} \\ \vdots \\ x_{j_x^F,j_y^F}\mathtt{dxT}_{j_x^F,j_y^F} + y_{j_x^F,j_y^F}\mathtt{dyT}_{j_x^F,j_y^F} + \mathtt{NT}_{j_x^F,j_y^F} \end{bmatrix}$$

Detalhes do sistema linear

Sistema linear para ($\mathtt{x}_{i,j}$,$\mathtt{y}_{i,j}$), $\begin{cases} j_y \le i \le j_y^F = j_y + M - 1 \\ j_x \le j \le j_x^F = j_y + M - 1 \end{cases}$

$$A = \begin{bmatrix} \mathtt{dxT}_{j_x,j_y} & \mathtt{dyT}_{j_x,j_y} & \mathtt{dzT}_{j_x,j_y} & \mathtt{N} \\ \mathtt{dxT}_{j_x+1,j_y} & \mathtt{dyT}_{j_x+1,j_y} & \mathtt{dzT}_{j_x+1,j_y} & \mathtt{N} \\ \vdots & \vdots & \vdots & \vdots \\ \mathtt{dxT}_{j_x^F,j_y^F} & \mathtt{dyT}_{j_x^F,j_y^F} & \mathtt{dzT}_{j_x^F,j_y^F} & \mathtt{N} \end{bmatrix}$$

$$\boldsymbol{b} = \begin{bmatrix} x_{j_x,j_y}\mathtt{dxT}_{j_x,j_y} + y_{j_x,j_y}\mathtt{dyT}_{j_x,j_y} + \mathtt{NT}_{j_x,j_y} \\ x_{j_x+1,j_y}\mathtt{dxT}_{j_x+1,j_y} + y_{j_x+1,j_y}\mathtt{dyT}_{j_x+1,j_y} + \mathtt{NT}_{j_x+1,j_y} \\ \vdots \\ x_{j_x^F,j_y^F}\mathtt{dxT}_{j_x^F,j_y^F} + y_{j_x^F,j_y^F}\mathtt{dyT}_{j_x^F,j_y^F} + \mathtt{NT}_{j_x^F,j_y^F} \end{bmatrix}$$

Algoritmo

function [xo,yo,zo] = euler3D(x,y,T,M,N)
[dxT,dyT,dzT] = DF_2D(x,y,T);
[ny,nx] = size(x);
for jx = 1:nx-M+1
  for jy = 1:ny-M+1
    Construa o sistema linear para ($\mathtt{x}_{i,j}$,$\mathtt{y}_{i,j}$), $\begin{cases} j_y \leq i \leq j_y + M - 1 \\ j_x \leq j \leq j_y + M - 1; \end{cases}$
    sol = A\b;
    xo(jy,jx)=sol(1); yo(jy,jx)=sol(2); zo(jy,jx)=sol(3);
  end
end

Algoritmo

```
function [xo,yo,zo] = euler3D(x,y,T,M,N)
[dxT,dyT,dzT] = DF_2D(x,y,T);
[ny,nx] = size(x);
for jx = 1:nx-M+1
  for jy = 1:ny-M+1
    k = 0;
    for i = jy:jy+M-1
      for j = jx:jx+M-1
        k = k + 1;
        A(k,:)  = [ dxT(i,j),dyT(i,j),dzT(i,j),N ];
        b(k) = x(i,j)*dxT(i,j)+y(i,j)*dyT(i,j)+N*T(i,j);
      end
    end
    sol = A\b;
    xo(jy,jx)=sol(1); yo(jy,jx)=sol(2); zo(jy,jx)=sol(3);
  end
end
```