

Tópicos de Transferência de Massa com uso do Matlab

Prof. Diomar Cesar Lobão

UFF – Volta Redonda, RJ

Outubro, 2009



Uso de diferenças finitas para discretização na solução de:

- ***Problema advectivo-difusivo 1D***
 - a) método explícito de Euler
 - b) método implícito de Euler
- ***Problema advectivo-difusivo 2D***
 - a) método explícito de Euler
 - b) método implícito de Euler

A solução numérica é calculada utilizando os recursos da linguagem e ambiente computacional do Matlab.

Convecção-Difusão

- No transporte de calor, a energia é transportada por:
convecção (dirigida pela circulação do fluido) e
difusão (transferência de calor condutivo dirigida pelo movimento Browniano “Brownian”)
- No transporte de soluto, o soluto é movido por:
advecção (isto é, o transporte desse fluido através da velocidade),
dispersão e *difusão* (mistura do soluto como um resultado do gradiente de concentração e “*difusão molecular*”).

Convecção (advecção) e difusão (dispersão) ocorre simultaneamente em muitas situações físicas e o modelamento preciso destes processos tem mostrado ser de difícil trato.

=> Dispersion in fluid flow is defined as Taylors (1921) “Diffusion by continuous movements”. Transport phenomena due to combined *advection* and *diffusion* is known as hydrodynamic *dispersion*. This so-called Taylor dispersion was experimentally [2] and theoretically [2,3] studied for flow through tubes. <=

Difusão: Tinta de caneta



Diffusion is a macroscopic process, which results in concentrations "smoothing out" or "diffusing" through time. For example, putting a drop of ink into a bowl of still water, the ink will eventually diffuse to evenly color the whole bowl. The macroscopic process of diffusion is driven by the microscopic process of Brownian motion. As a mental example of Brownian motion (BM), think of the water particles as tennis balls, and the molecules of ink as beach balls; the tennis balls are bouncing all around in the bowl (due to their internal energy, which is measured by temperature), striking and jostling the larger beach balls.

In the bowl, all we see are the ink "beach balls", the water "tennis balls" are invisible, so it appears as though the beach balls are moving around randomly by themselves. Since the motion of the smaller water molecules is random, it tends to average things out through time (second law of thermodynamics). *Kris Kuhlman. September 25, 2007*

The **second law of thermodynamics** is an expression of the universal law of increasing **entropy**, stating that the entropy of an isolated system which is not in equilibrium will tend to increase over time, approaching a maximum value at equilibrium. In simple terms, the second law is an expression of the fact that over time, ignoring the effects of self-gravity, differences in temperature, pressure, and density tend to even out in a physical system that is isolated from the outside world. **Entropy** is a measure of how far along this evening-out process has progressed.

Observação

A influência da velocidade no fenômeno **dispersivo** pode ser analisado à luz do número de Peclet, P . Como exemplo, veja a situação de escoamento subterrâneo: para valores baixos de $P < 1$, o processo **dispersivo longitudinal** é controlado pela **difusão molecular**; para valores elevados $P > 10$ o processo **dispersivo longitudinal** é controlado pela **advecção**.

Também para a **dispersão transversal** se pode estabelecer uma relação semelhante: para valores de $P < 1$ o processo **dispersivo transversal** é controlado pela **difusão molecular**; para valores de $P > 100$, o processo **dispersivo transversal** é controlado pela **advecção**.

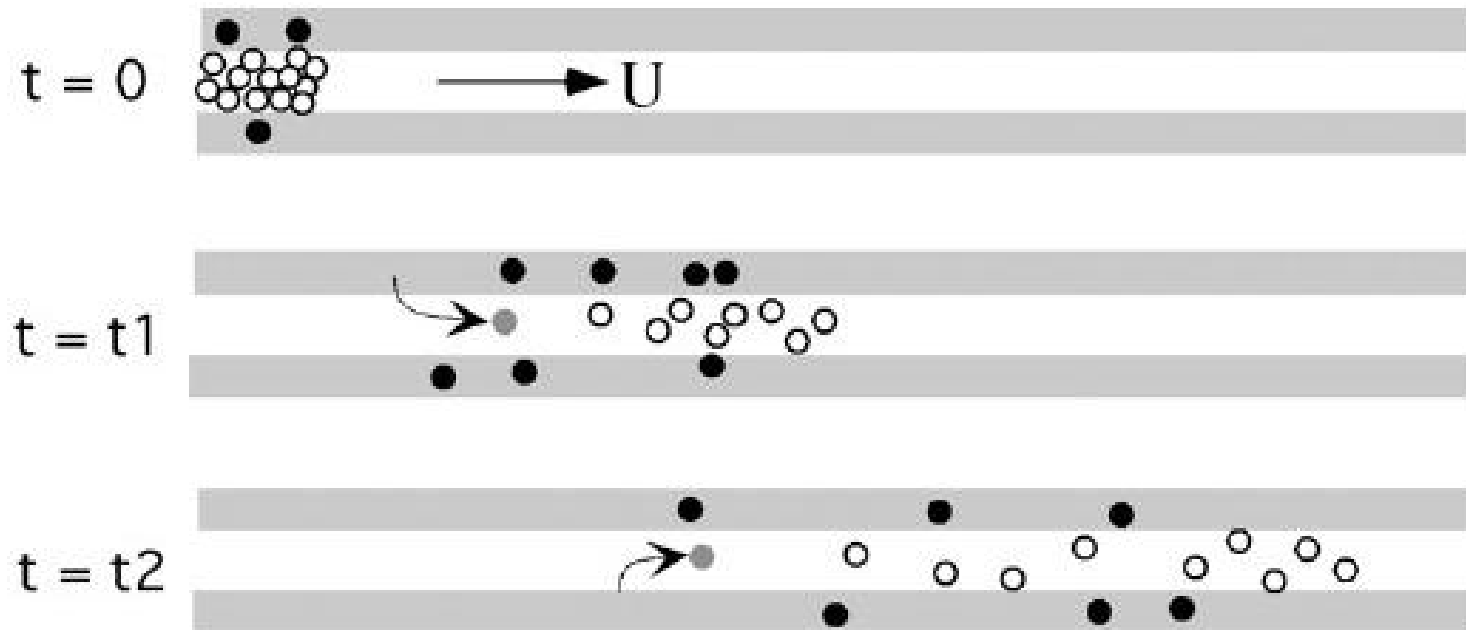
$$P = Vx d / Dd$$

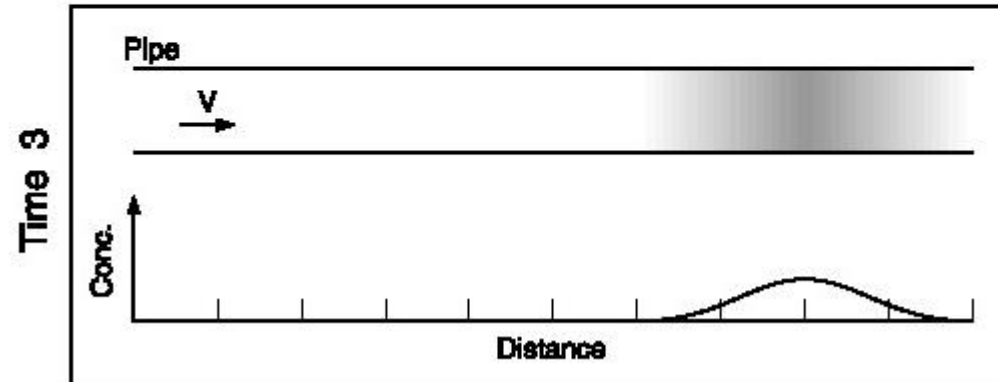
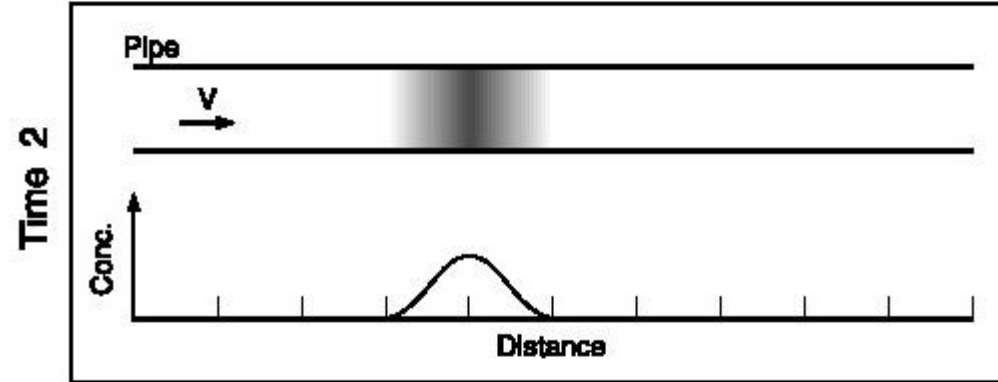
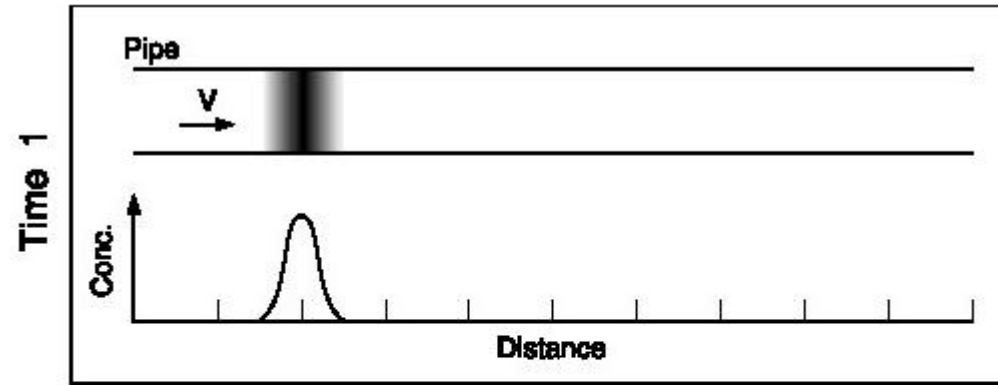
Dd é o coeficiente efetivo de difusão molecular ($\text{cm}^2 \text{s}^{-1}$)

Vx é a velocidade média do soluto (cm s^{-1}); e

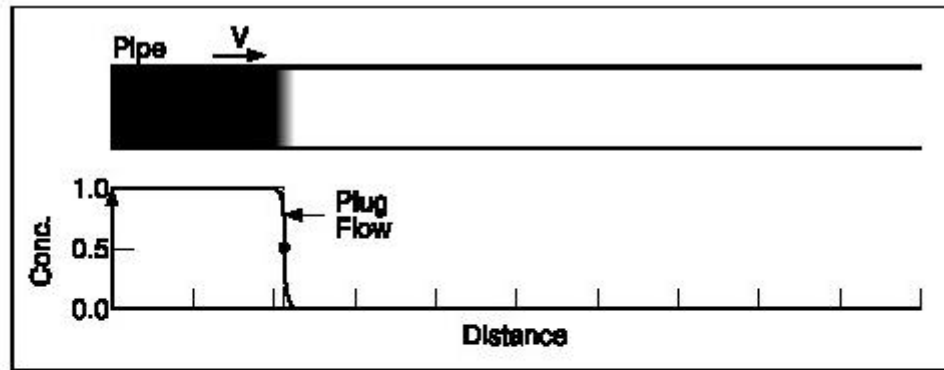
d é o diâmetro médio das partículas (cm).

Advecção - Difusão

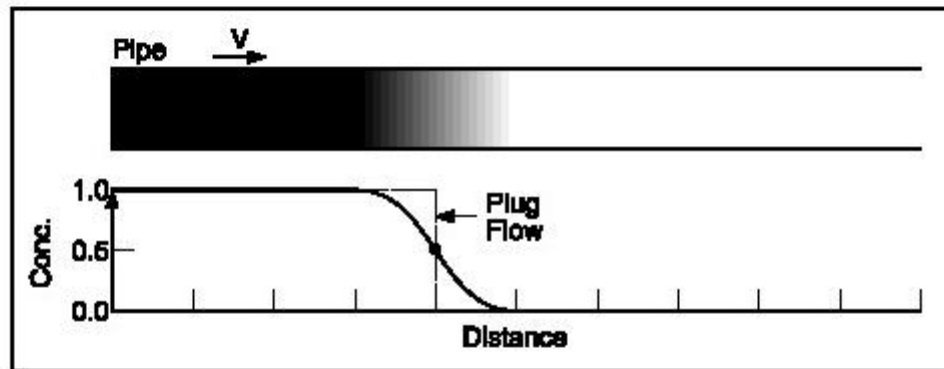




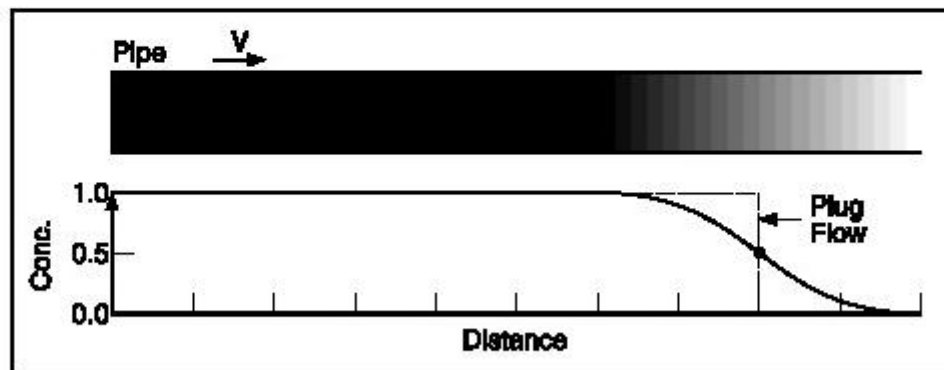
Time 1



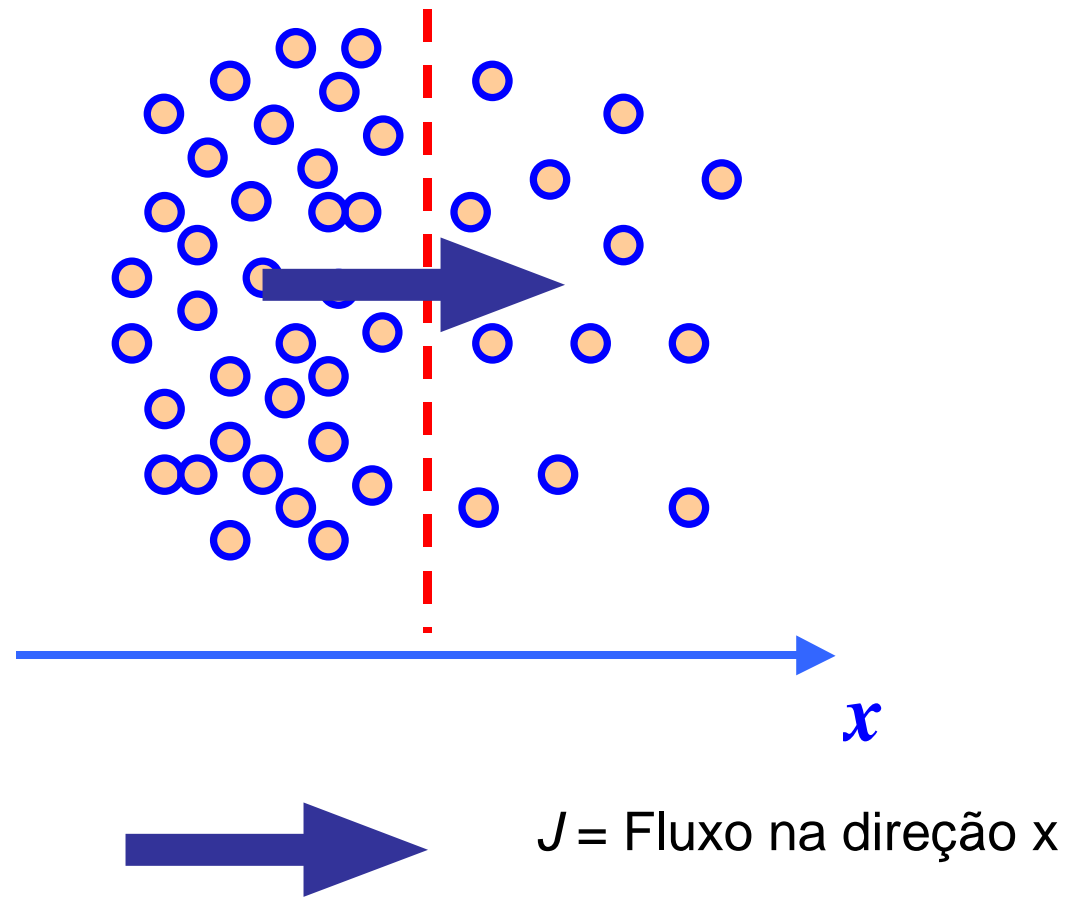
Time 2



Time 3



Primeira Lei da Difusão de Fick



O fluxo na direção x é dado por: $J_x = -D \frac{\partial C}{\partial x}$

D → coeficiente de difusão

Substância	Equação	Unidade de J	Unidade da cte
Partículas	$J_s = -\frac{D\partial C}{\partial x}$	$m^{-2}s^{-1}$	m^2s^{-1}
Massa	$J_m = -D \frac{\partial C}{\partial x}$	$Kg m^{-2}s^{-1}$	m^2s^{-1}
Calor	$J_H = -\lambda \frac{\partial T}{\partial x}$	$Jm^{-2}s^{-1}$	$Jk^{-1}m^{-1}s^{-1}$
Carga elétrica	$J_e = -\tau \frac{\partial v}{\partial x}$	$Cm^{-2}S^{-1}$	$Cm^{-1}S^{-1}V^{-1}$
Lei de Newton da Viscosidade	$\frac{F}{S} = -\eta \frac{\partial V_x}{\partial y}$	$Nm^{-2} (Pa)$	$Kg m^{-1}s^{-1}$

Segunda Lei da Difusão de Fick

Diferenciando a 1ª Lei de Fick temos:

$$\left. \begin{aligned} \vec{\nabla} \cdot (\vec{J}) &= \vec{\nabla} \cdot (-D \vec{\nabla} C(x, y, z)) \\ \vec{\nabla} \cdot \vec{J} &= -D \vec{\nabla}^2 C(x, y, z) \end{aligned} \right\}$$

Mas pela equação da continuidade

$$\vec{\nabla} \cdot \vec{J} = -\frac{\partial C}{\partial t}$$

$$\frac{\partial C}{\partial t} = D \vec{\nabla}^2 C(x, y, z)$$

A segunda lei *prevê a variação temporal da concentração* ou seja como $C(x, y, z, t)$ varia no tempo e no espaço. Uma solução *analítica* para esta equação pode ser encontrada:

$$C(x, t) = \frac{C_o}{\sqrt{2\pi\sigma(t)}} e^{-x^2/2\sigma^2(t)}$$

Veja,

$$\frac{\partial C}{\partial t} = C_o \left[-\frac{1}{\sqrt{2\pi\sigma^2(t)}} e^{-x^2/2\sigma^2(t)} + \frac{1}{\sqrt{2\pi\sigma(t)}} \left(\frac{x^2}{\sigma^3} \right) e^{-x^2/2\sigma^2(t)} \right] \frac{d\sigma}{dt}$$

$$\frac{\partial C}{\partial x} = -\frac{C_o}{\sqrt{2\pi\sigma(t)}} \left[e^{-x^2/2\sigma^2(t)} \right] \left(\frac{x}{\sigma^2(t)} \right)$$

$$\frac{\partial^2 C}{\partial x^2} = -\left(\frac{C_o}{\sqrt{2\pi\sigma^3(t)}} \right) e^{-x^2/2\sigma^2(t)} + \frac{C_o}{\sqrt{2\pi\sigma(t)}} \left(\frac{x}{\sigma^2(t)} \right) e^{-x^2/2\sigma^2(t)} \left(\frac{x}{\sigma^2(t)} \right)$$

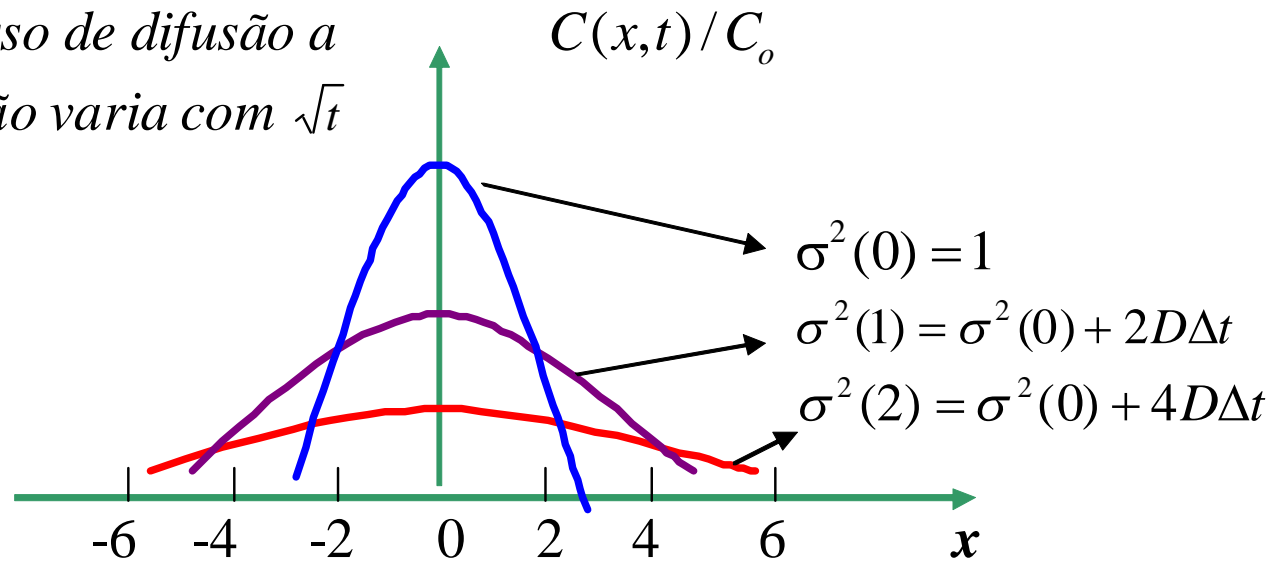
Substituindo na equação original, tem-se:

$$\frac{C_o}{\sqrt{2\pi\sigma^2}} e^{-x^2/2\sigma^2} \left(-1 + \frac{x^2}{\sigma^2} \right) \frac{d\sigma}{dt} = D \left(\frac{C_o}{\sqrt{2\pi\sigma^3}} \right) e^{-x^2/2\sigma^2} \left(-1 + \frac{x^2}{\sigma^2} \right) \quad \Rightarrow \quad \sigma \frac{d\sigma}{dt} = D \quad \text{Integrando:}$$

$$\sigma^2(t) = 2Dt + \sigma^2(0)$$

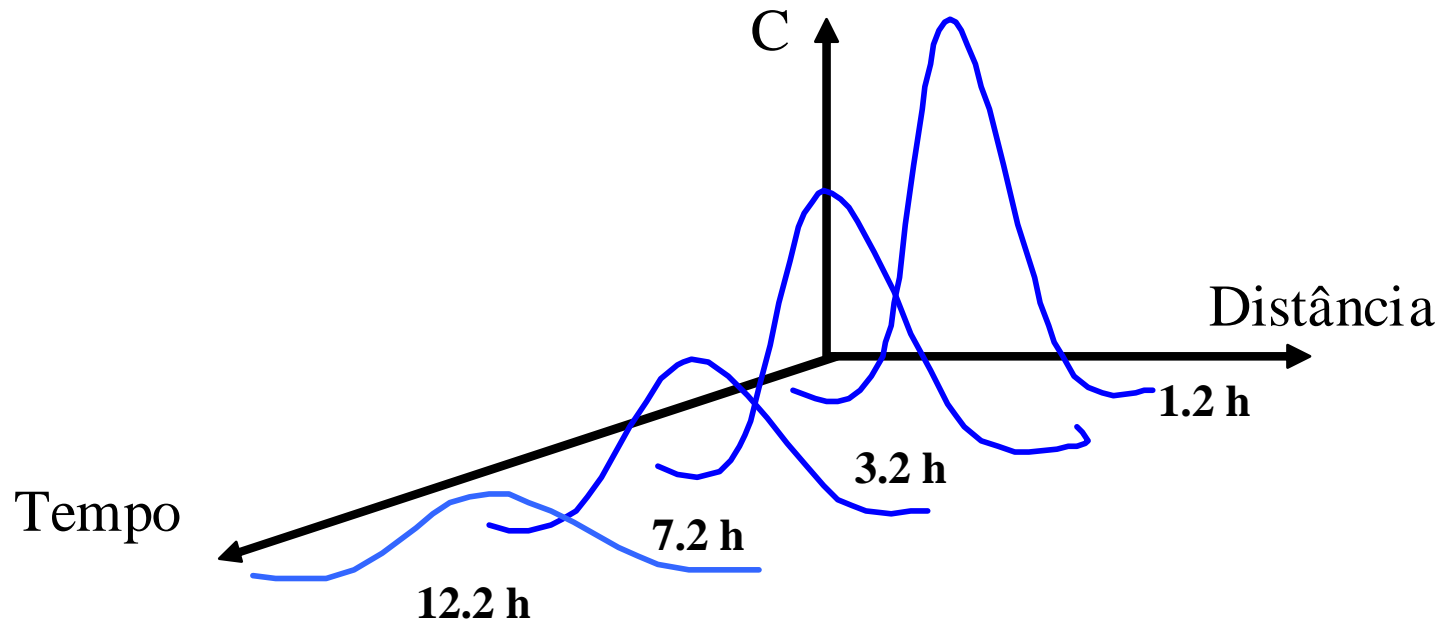
$$\sigma^2(t) = 2Dt + \sigma^2(0)$$

*Num processo de difusão a
concentração varia com \sqrt{t}*



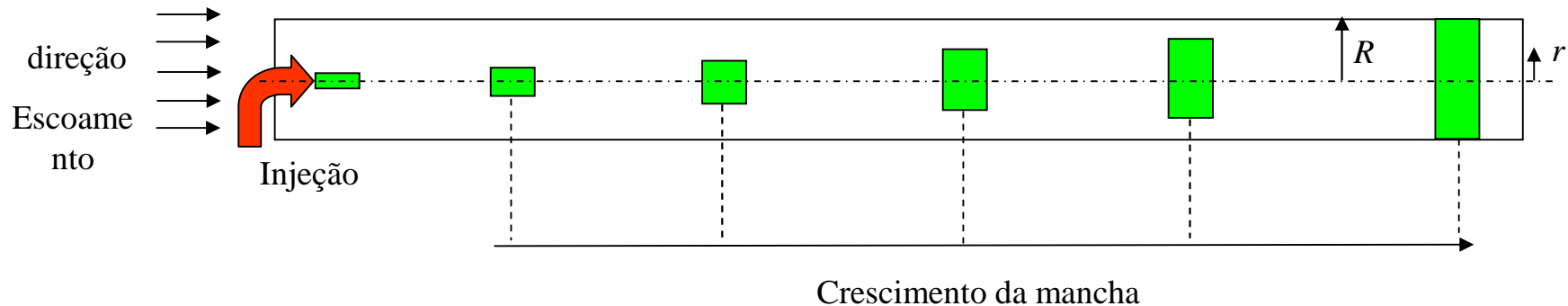
Difusão

Concentração de Fe criados por radiação



Simulação 1D

Advecção e Difusão de uma substância em um meio 1D



O crescimento e transporte da mancha é um processo *advectivo-difusivo*, o qual pode ser formulado como:

$$\frac{\partial c}{\partial t} + u \frac{\partial c}{\partial x} = D \frac{\partial^2 c}{\partial x^2}$$

Onde C é a concentração da substância, D é a difusividade. Para valores suficientemente grandes de R é possível assumir as seguintes condições de contorno e condições iniciais:

Condição inicial: C_0 é a concentração inicial na origem $x=0$.

Condição contorno 1: condição de simetria na origem.

Condição contorno 2: concentração zero no infinito ou extrapolação

A solução analítica deste problema 1D é da seguinte forma:

Calcula-se e plota-se a concentração de C contra a posição x em diferentes tempos t para a ***advecção-difusão*** da mancha. A ***distribuição inicial é dada por:***

$$C(x, t) = \frac{M}{A \sqrt{4 \pi D t}} e^{-\frac{(x-ut)^2}{4 D t}}$$

x varia de 0 à 1 com incrementos de 0.01, ou seja, $x=0, 0.01, 0.02, 0.03, \dots, 1$.

Assume: $M=250(\text{kg})$, $u=0.25(\text{m/s})$, $C \Rightarrow (\text{kg/m}^3/\text{s})$ e $D=25.4(\text{m}^2/\text{s})$.

A solução numérica adotada é o método ***Explícito de Euler***: Veja o ***script Matlab*** adiante.

O método Explícito de Euler 1D

Utilizando diferenças finitas para aproximar as derivadas contínuas, de primeira ordem no tempo e segunda ordem no espaço, obtém-se:

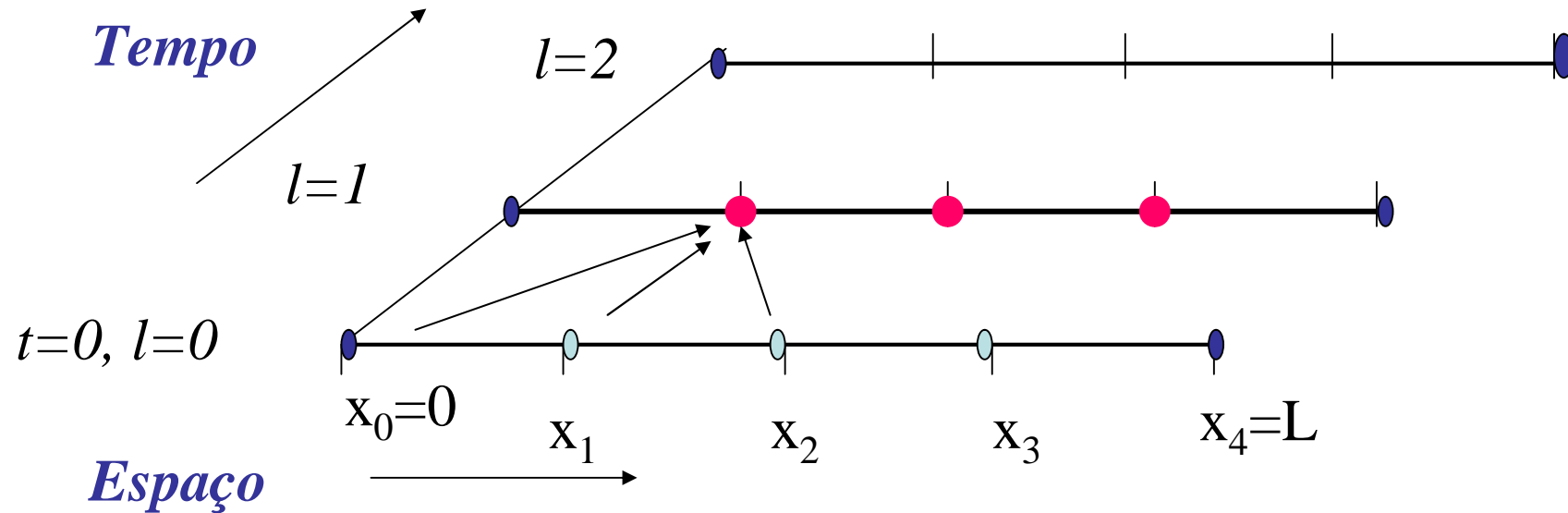
$$c_i^{m+1} = c_i^m - \frac{u\Delta t}{2\Delta x} (c_{i+1}^m - c_{i-1}^m) + \frac{D\Delta t}{\Delta x^2} (c_{i-1}^m - 2c_i^m + c_{i+1}^m)$$

Condições de contorno:

$$c(1)=c(2)$$

$$c(Nx)=c(Nx-1);$$

Método Explícito de Euler 1D



$$\begin{array}{l}
 l = 0, \quad i = 1 \\
 \quad \quad i = 2 \\
 \quad \quad i = 3
 \end{array}
 \left\{ \begin{array}{l}
 c_1^1 = c_1^0 + u \frac{\Delta t}{2 \Delta x} (c_2^0 - c_0^0) + \frac{D \Delta t}{\Delta x^2} (c_2^0 - 2c_1^0 + c_0^0) \\
 c_2^1 = c_2^0 + u \frac{\Delta t}{2 \Delta x} (c_3^0 - c_1^0) + \frac{D \Delta t}{\Delta x^2} (c_3^0 - 2c_2^0 + c_1^0) \\
 c_3^1 = c_3^0 + u \frac{\Delta t}{2 \Delta x} (c_4^0 - c_2^0) + \frac{D \Delta t}{\Delta x^2} (c_4^0 - 2c_3^0 + c_2^0)
 \end{array} \right.$$

As variáveis do RHS são conhecidas.
 $i=0$ e $i=4$ são conhecidos como: C.C.

$i=1..3$ são C.I.

Convergência & Estabilidade dos Métodos Explícitos

Convergência: Quando $\Delta x \rightarrow 0$ & $\Delta t \rightarrow 0$,
A solução numérica aproxima da solução exata.

• **Estabilidade:**

$$\text{se } \left(\lambda = \frac{D\Delta t}{(\Delta x)^2} \right) > \frac{1}{2}, \text{ ou } \Delta t > \frac{(\Delta x)^2}{2D},$$

-Instável

o erro irá aumentar.

$$\text{se } \frac{1}{4} < \lambda \leq \frac{1}{2}, \text{ ou } \frac{(\Delta x)^2}{4D} < \Delta t \leq \frac{(\Delta x)^2}{2D},$$

-Estável

o erro não irá aumentar porém irá oscilar.

$$\text{se } \lambda \leq \frac{1}{4}, \text{ ou } \Delta t \leq \frac{(\Delta x)^2}{4D},$$

o erro irá diminuir.

“Portanto, para ser preciso, os passos no tempo e espaço devem ser pequenos !!!”

O que isso implica?

Δx e Δt – pequenos implica em grande tempo de CPU !!!.

Matlab script para método *explícito de Euler 1D*

```
Nx=100; % number of grid
Diff_C=25.4; % Diffusibility coefficient (m2/s) It is also
% possible substitute it by the
Longitudinal
% Dispersion coeficiente K
% (see page 588 Merle C. Porter)
Cour=0.00008; % Courant-friedrich-lewy number
%
xmax=1.0;
xmin=0.0;
t=0.0; % elapsed time (s)
u0 = 1.0; % velocity (m/s)
Area_a = 20.0; % Area (m2)
Mass_t = 250.0; % Total mass (kg)
%
x = zeros(Nx,1); % arrangement for x coordinate
dx=(xmax-xmin)/(Nx-1); % incremental distance
in x-direction
for(i=1:Nx)
    x(i)=dx*(i-1);
end;
%
t=0.0; % initialization of elapsed time
dt=Cour*dx/u0; % time step to calculate
```

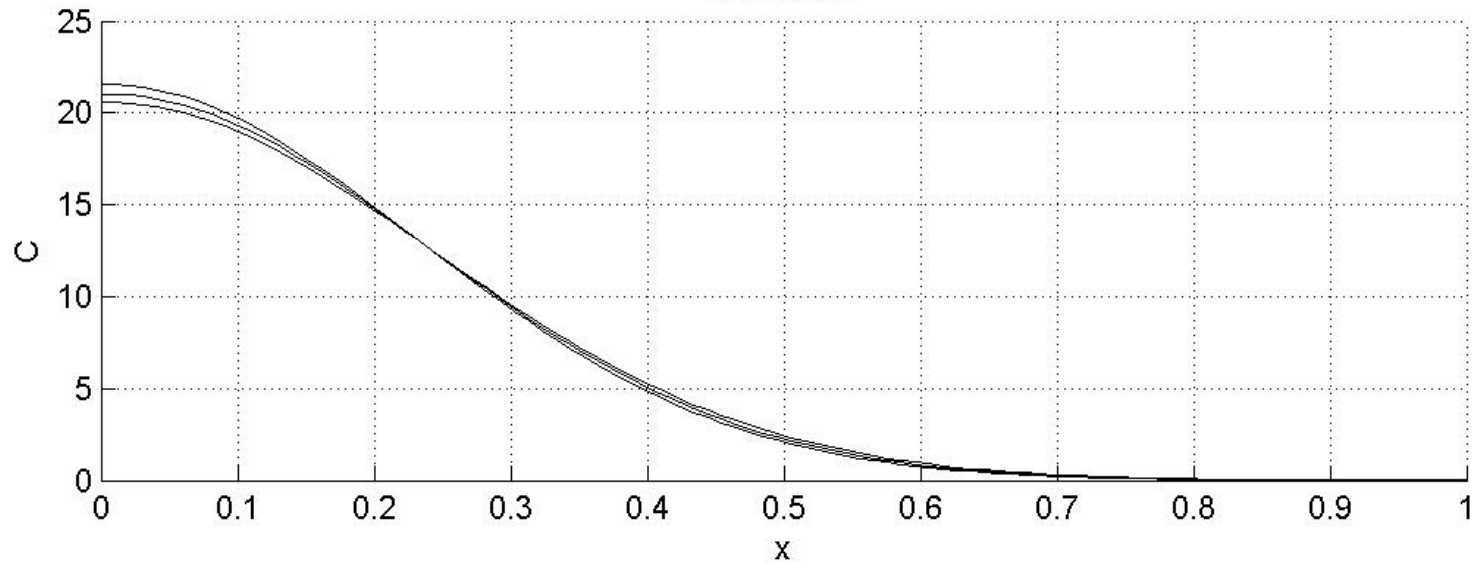
```

for k=1:100
%
  for(i=2:Nx-1) % install initial values
    f(i)=Mass_t/(Area_a*sqrt(4.0*pi*Diff_C*(t+0.001)));
    f(i)=f(i)*exp(-(x(i)-u0*(t+0.001))^2/(4.0*Diff_C*(t+0.001)));
  end
  %boundary condition
  f(1)=f(2); f(Nx)=f(Nx-1);
  %
  if (k == 1) for(i=1:Nx) f0(i) = f(i); end; end;
%   Exp_Euler scheme - Explicit Euler Scheme

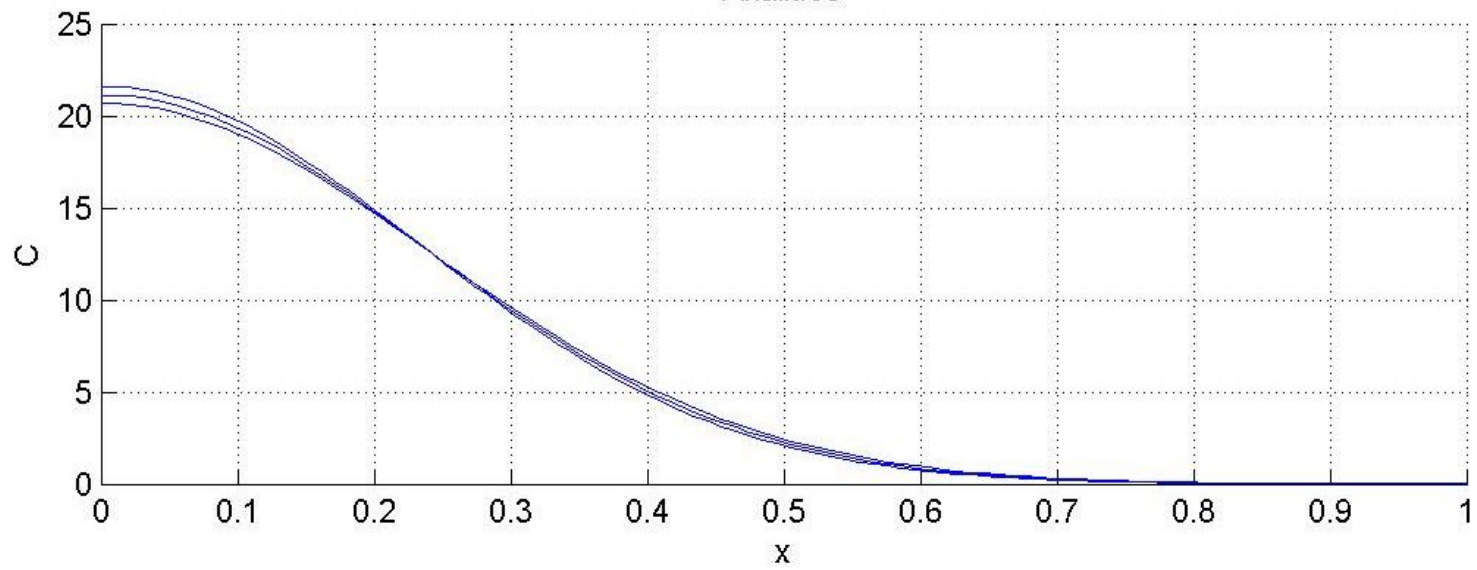
  for(i=2:Nx-1)
    ff(i)=f0(i)-dt/(2*dx)*u0*(f0(i+1)-f0(i-1)) +
dt/(dx*dx)*Diff_C*(f0(i+1) - 2.0*f0(i)+f0(i-1));
  end
  % install boundary condition
  ff(1)=ff(2);
  ff(Nx)=ff(Nx-1);
  f0=ff;
  %
  t = t + dt;
  if(mod(k,10) == 0)
  % Plotting...
  figure(1);
  subplot(2,1,1), hold on, plot(x,ff(:)),
  xlabel('x'),ylabel('C'),title('Numerico');
  hold off
  subplot(2,1,2), hold on, plot(x,f0(:)),
  xlabel('x'),ylabel('C'),title('Analitico');
  hold off
  pause(0.01);
  end
%
end %Iterative time cycle ends here...

```

Numerico



Analitico



O método adotado: *Implícito de Euler 1D*

Usa-se o fato de que as derivadas são aproximadas pelas variáveis tomadas no nível de tempo (m+1)

$$\boxed{-A c_{i-1}^{m+1} + (1 + 2p)c_i^{m+1} + B c_{i+1}^{m+1} = c_i^m}$$

Sistema Tri-diagonal

$$p = \frac{D \Delta t}{\Delta x^2} \quad A = \frac{\Delta t u}{2\Delta x} + \frac{D \Delta t}{(\Delta x)^2} \quad B = \frac{\Delta t u}{2\Delta x} - \frac{D \Delta t}{(\Delta x)^2}$$

Para as Condições de contorno adota é possível escrever:

$$\boxed{A \cdot c^{m+1} = c^m}$$

Condições de contorno: $C(1)=C(2)+A \cdot C(1)$

$C(Nx-2)=C(Nx-1)-B \cdot C(Nx)$

$$\mathbf{A} \cdot \mathbf{c}^{m+1} = \mathbf{c}^m$$

$$\begin{bmatrix}
 1+2p & +B & 0 & \dots & & & & 0 \\
 -A & 1+2p & B & 0 & \dots & & & 0 \\
 0 & -A & 1+2p & +B & 0 & \dots & & 0 \\
 & & & \ddots & \ddots & \ddots & & \\
 0 & \dots & & 0 & -A & 1+2p & +B & 0 \\
 0 & & \dots & & 0 & -A & 1+2p & +B \\
 0 & & & \dots & & 0 & -A & 1+2p
 \end{bmatrix}_{(N-2) \times (N-2)}
 \begin{bmatrix}
 c_2 \\
 c_3 \\
 c_4 \\
 \vdots \\
 c_{N-2} \\
 c_{N-1}
 \end{bmatrix}^{m+1}
 =
 \begin{bmatrix}
 c_2 \\
 c_3 \\
 c_4 \\
 \vdots \\
 c_{N-2} \\
 c_{N-1}
 \end{bmatrix}^m$$

De maneira clássica utiliza-se o método de Thomas para resolver este sistema tri-diagonal, com o Matlab é possível usar a função **INV**

Passos adotados na implementação do método *Implícito de Euler 1D*.

- 1) Entrada de parâmetros e inicialização
- 2) Construir a matriz $A(N_x-2) \times (N_x-2)$ e calcular a sua inversa.
- 3) Aplicar as condições de contorno e atualizar a solução em cada passo de tempo.

```
Nx=100; % number of grid
Niter=180; % number of iterations
f0=zeros(Nx,1);
%
Diff_C=25.4; % Diffusibility coefficient (m2/s).(see page 588 Merle C. Porter)
Cour=0.00008; % Courant-friedrichs-lewy number
%
xmax=1.0;
xmin=0.0;
t=0.0; % elapsed time (s)
u0 = 1.0; % velocity (m/s)
Area_a = 20.0; % Area (m2)
Mass_t = 250.0; % Total mass (kg)
%
x = zeros(Nx,1); % arrangement for x coordinate
dx=(xmax-xmin)/(Nx-1); % incremental distance in x-direction
for(i=1:Nx)
    x(i)=dx*(i-1);
end;
%
t=0.0; % initialization of elapsed time
dt=Cour*dx/u0; % time step to calculate
```

```

% Matrix of coefficients
%
aa=dt*u0/(2.*dx) + Diff_C*dt/dx^2;
p=2.*Diff_C*dt/dx^2;
bb=1. + p;
cc=dt*u0/(2.*dx) - Diff_C*dt/dx^2;
A=zeros(Nx-2,Nx-2);
A(1,1)=bb; A(1,2)=cc; %first line of A
A(Nx-2,Nx-3)=-aa; A(Nx-2,Nx-2)=bb;
%last line of A
%
for i=2:Nx-3 % Matrix A from i=2...Nx-3
A(i,i)=bb; A(i,i-1)=-aa; A(i,i+1)=cc;
end

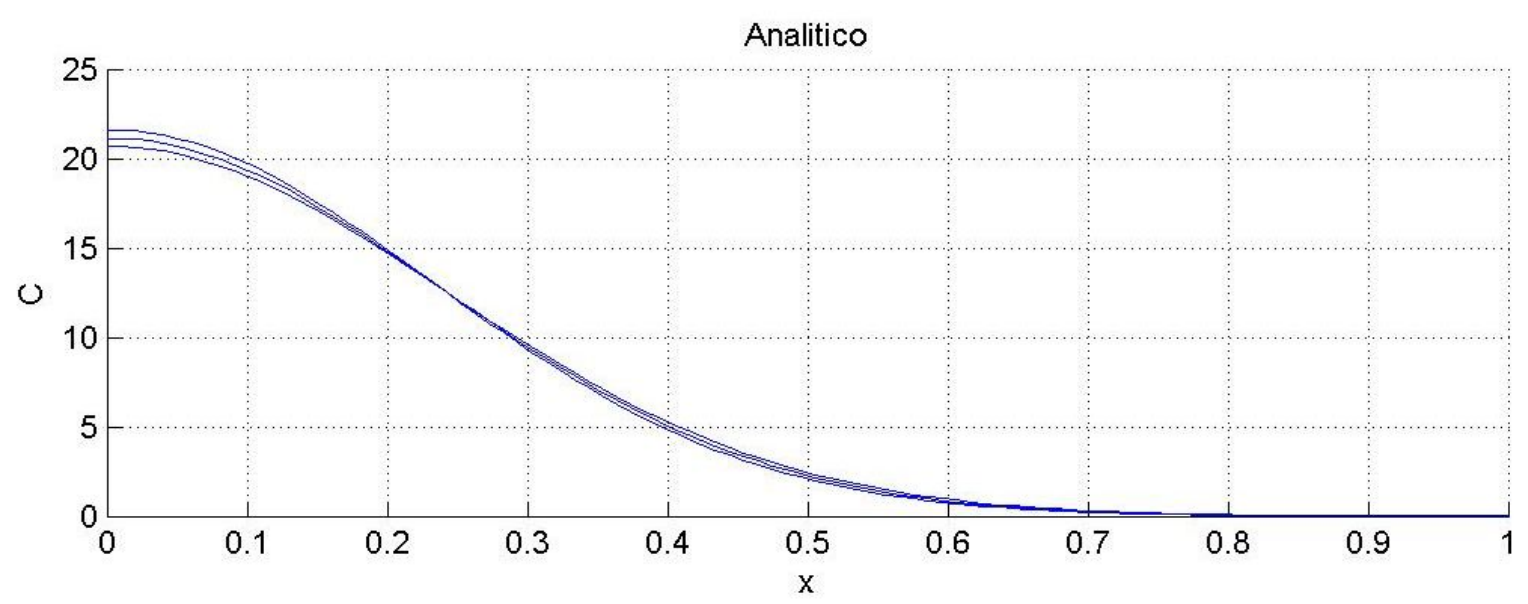
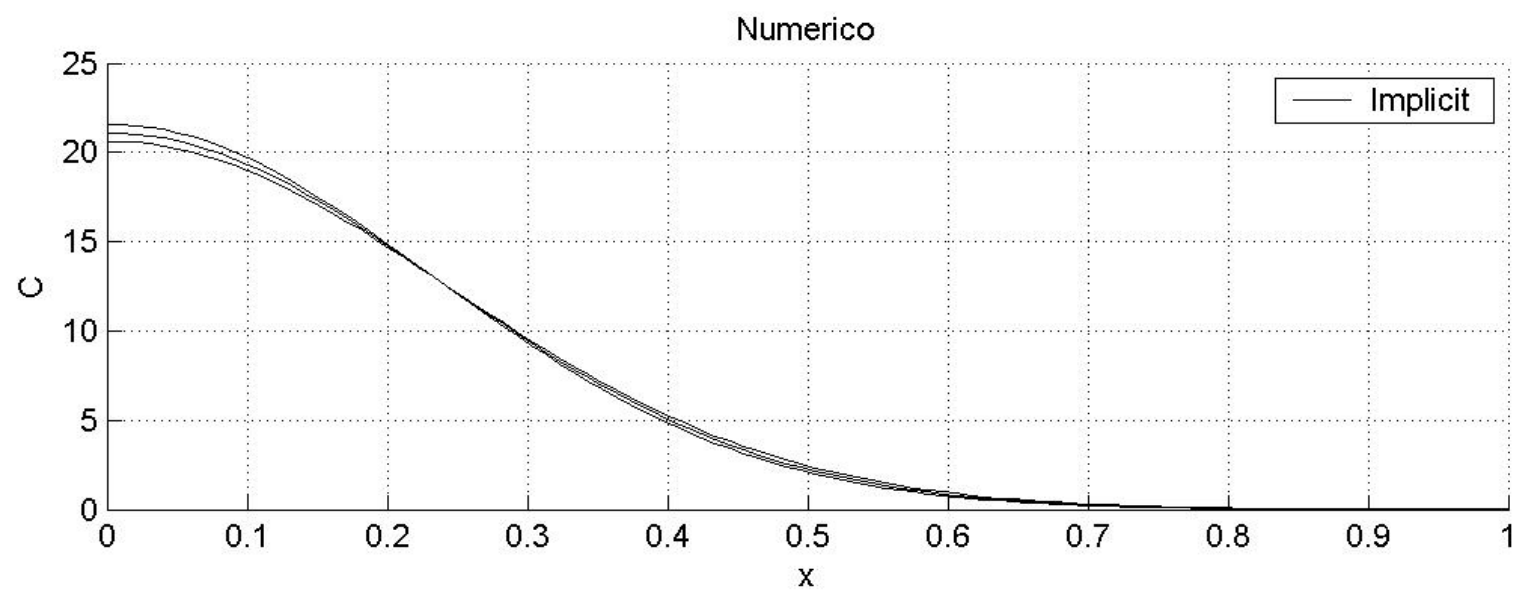
```

```

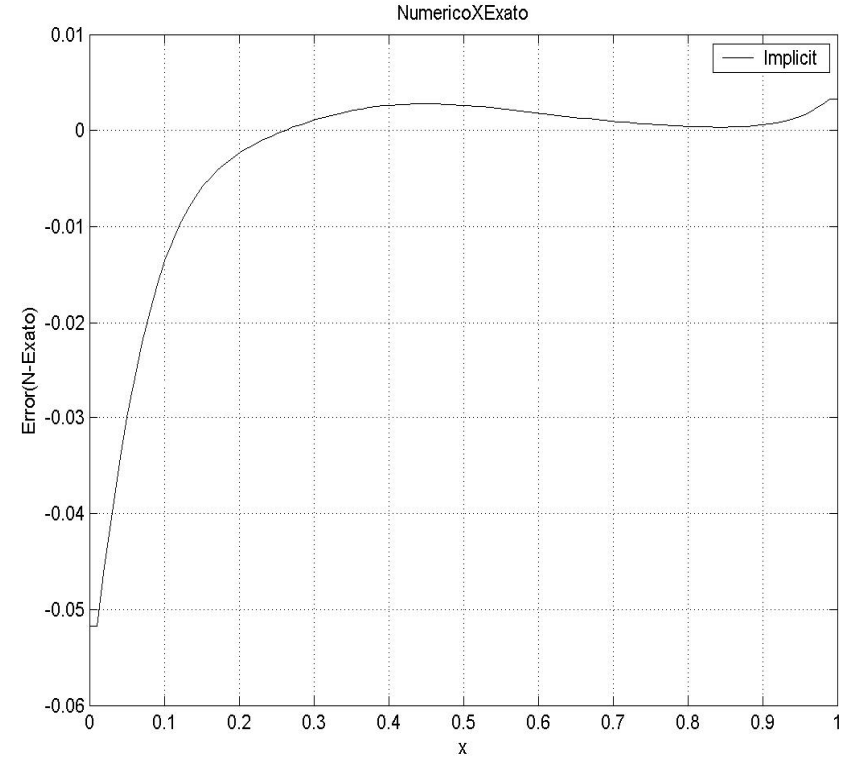
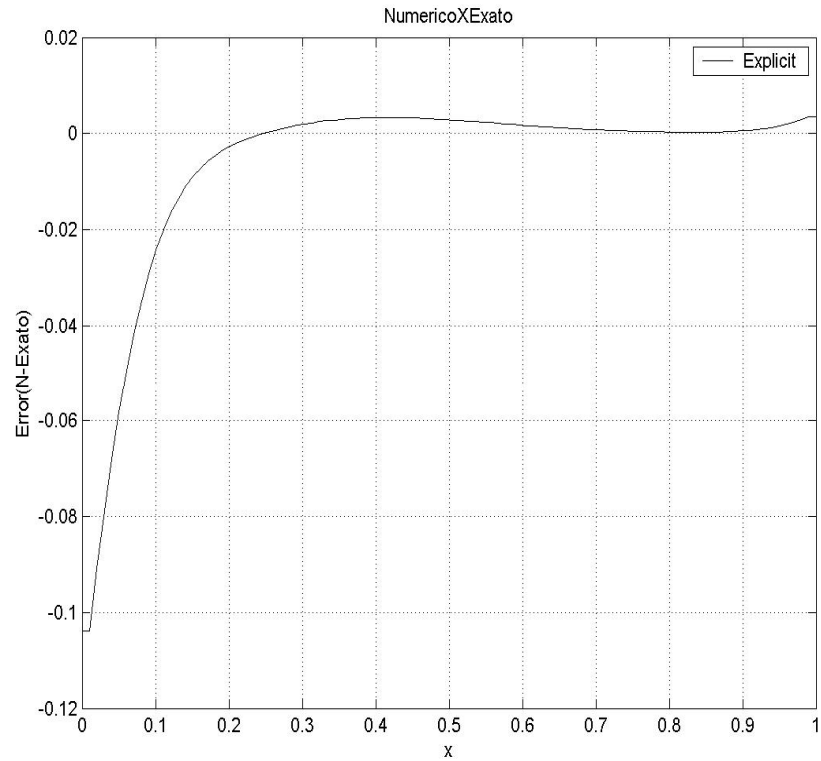
for k=1:Niter
%
for(i=2:Nx-1) % install initial values
f(i)=Mass_t/(Area_a*sqrt(4.0*pi*Diff_C*(t+0.001)));
f(i)=f(i)*exp(-(x(i)-u0*(t+0.001))^2/(4.0*Diff_C*(t+0.001)));
end
%boundary condition
f(1)=f(2); f(Nx)=f(Nx-1);
%
if (k == 1) for(i=1:Nx) f0(i) = f(i); end; end;
%
% Implicit Euler scheme - Implicit Euler Scheme
% Vector B on the right side
B(1)=f0(2) + aa*f0(1);
B(Nx-2)=f0(Nx-1) - cc*f0(Nx) ;
for ii=2:Nx-3
B(ii)=f0(ii+1);
end
% Matrix ...Inverstion
AA=inv(A);
C2=AA*(B');
% install boundary condition
for ii=1:Nx-2
f0(ii+1) = C2(ii);
end
f0(1)=f0(2);
f0(Nx)=f0(Nx-1);
%
t = t + dt;
if(mod(k,60) == 0)
% Plotting...
figure(1);
subplot(2,1,1), hold on, plot(x,f0(:),'k-'), grid on,
xlabel('x'),ylabel('C'),title('Numerico'); legend('Implicit');
hold off
subplot(2,1,2), hold on, plot(x,f(:),'b-'), grid on,
xlabel('x'),ylabel('C'),title('Analitico');
hold off
pause(0.01);
end
%
end %Iterative time cycle ends here...

```

← INV

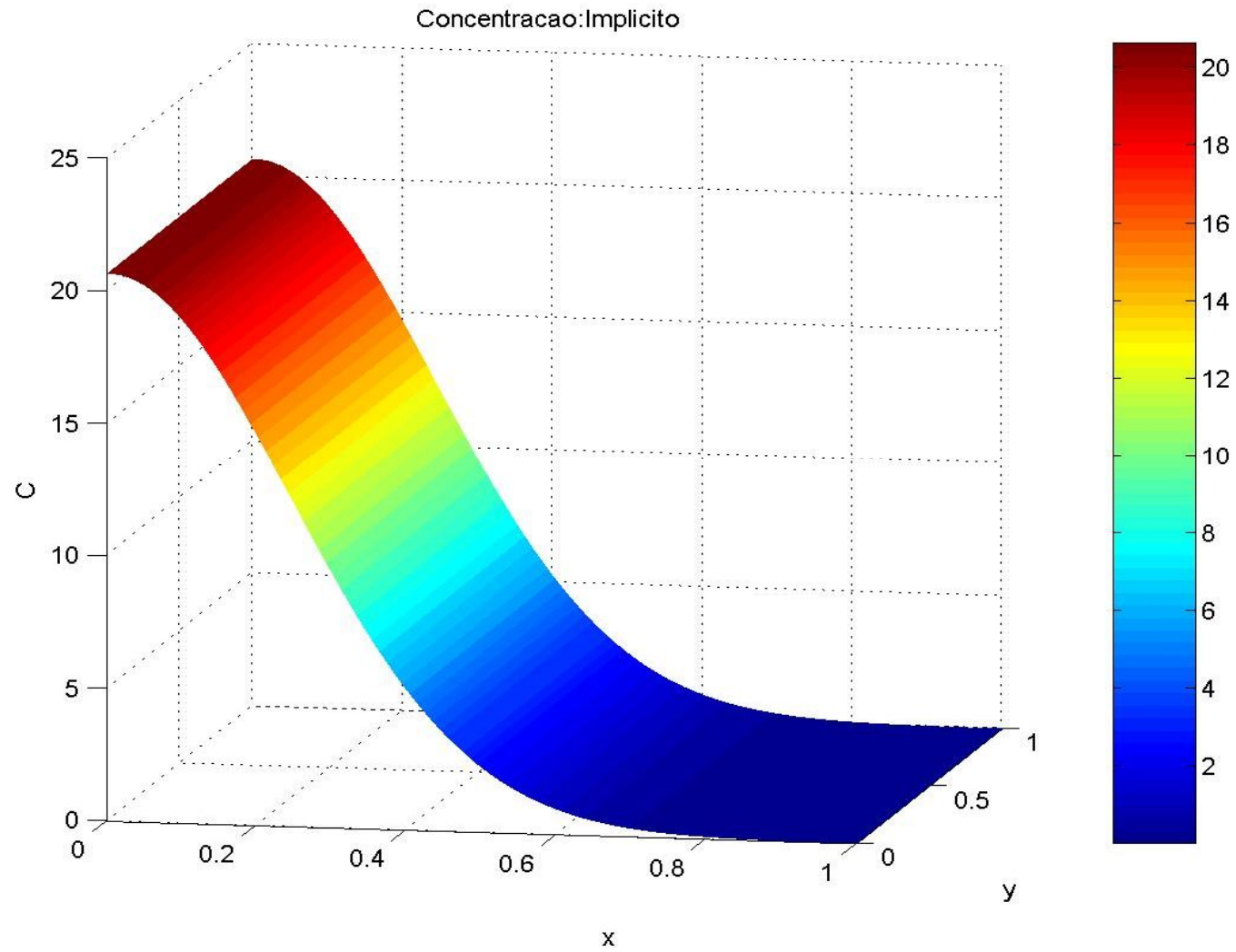


Avaliação do erro: Método *Explícito* e *Implícito* de Euler 1D



Erro = Solução(Numérica – Exata)

O método adotado: *implícito de Euler 1D*



Simulação em 2D

O método adotado: *explícito de Euler 2D*

$$\frac{\partial c}{\partial t} + u \frac{\partial c}{\partial x} + v \frac{\partial c}{\partial y} = D_x \frac{\partial^2 c}{\partial x^2} + D_y \frac{\partial^2 c}{\partial y^2}$$

Utilizando diferenças finitas para aproximar as derivadas contínuas, obtém-se:

$$c_{i,j}^{m+1} = c_{i,j}^m - \frac{u \Delta t}{2 \Delta x} (c_{i+1,j}^m - c_{i-1,j}^m) + \frac{D \Delta t}{\Delta x^2} (c_{i-1,j}^m - 2c_{i,j}^m + c_{i+1,j}^m) \\ + \frac{v \Delta t}{2 \Delta y} (c_{i,j+1}^m - c_{i,j-1}^m) + \frac{D \Delta t}{\Delta y^2} (c_{i,j-1}^m - 2c_{i,j}^m + c_{i,j+1}^m)$$

Condições de contorno:

$$c(1,:) = c(2,:)$$

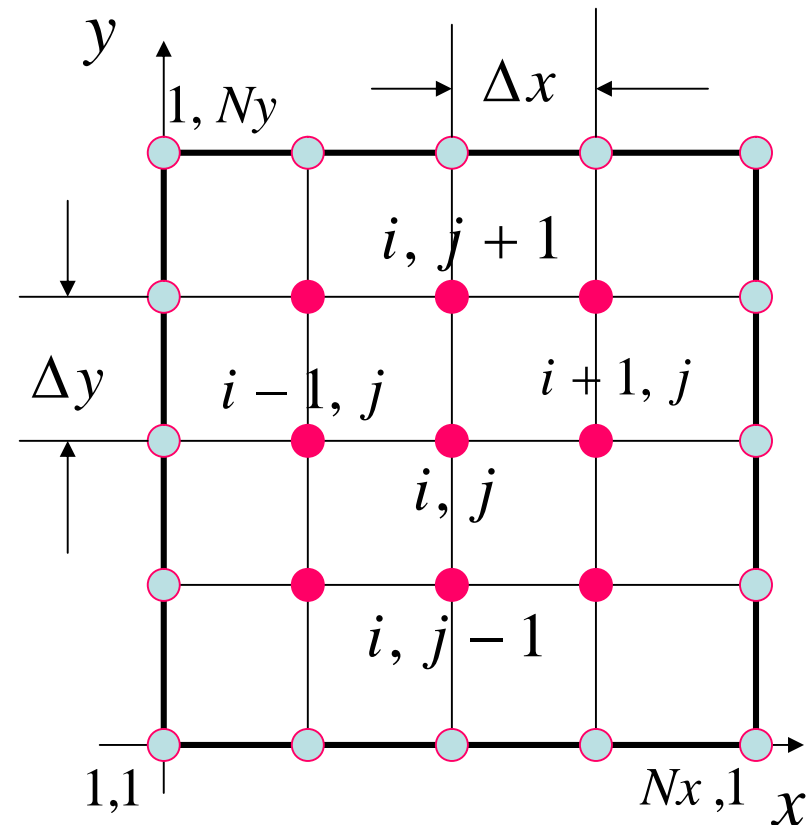
$$c(:,Ny) = c(:,Nx-1)$$

Discretização

Malha, e o esquema de Diferenças Finitas.

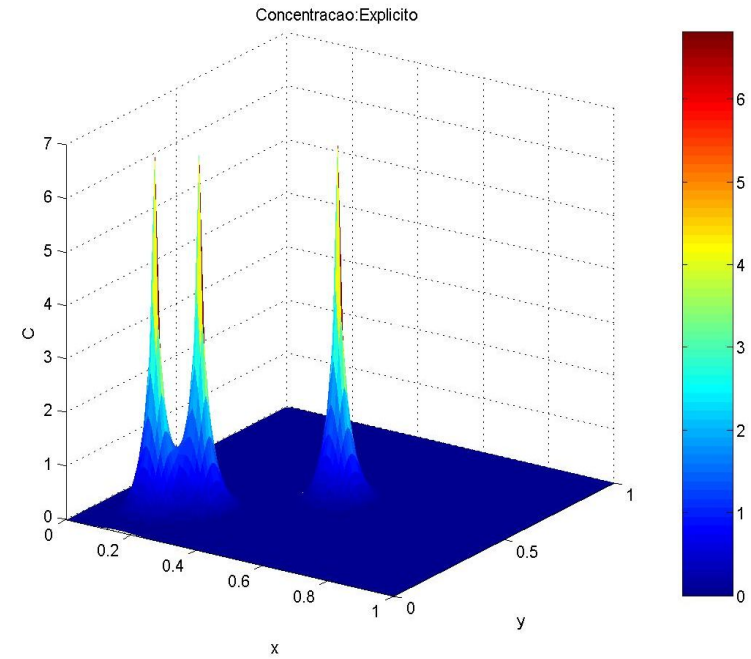
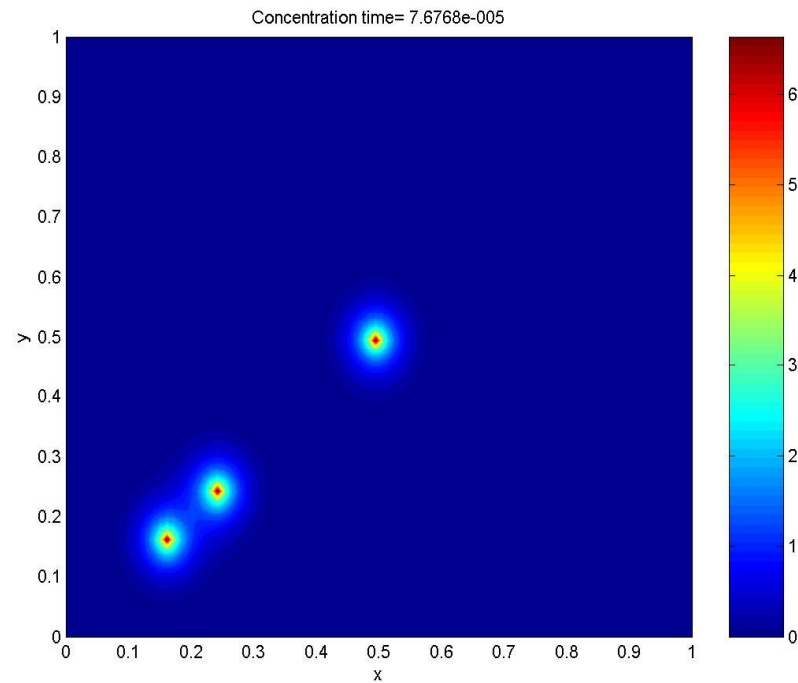
- N_x intervalos na direção x .
- N_y intervalos na direção y .
- pontos Interior $(N_x-2)*(N_y-2)$.
- Total de pontos $(N_x)*(N_y)$.

O eixo perpendicular ao plano é o eixo do tempo

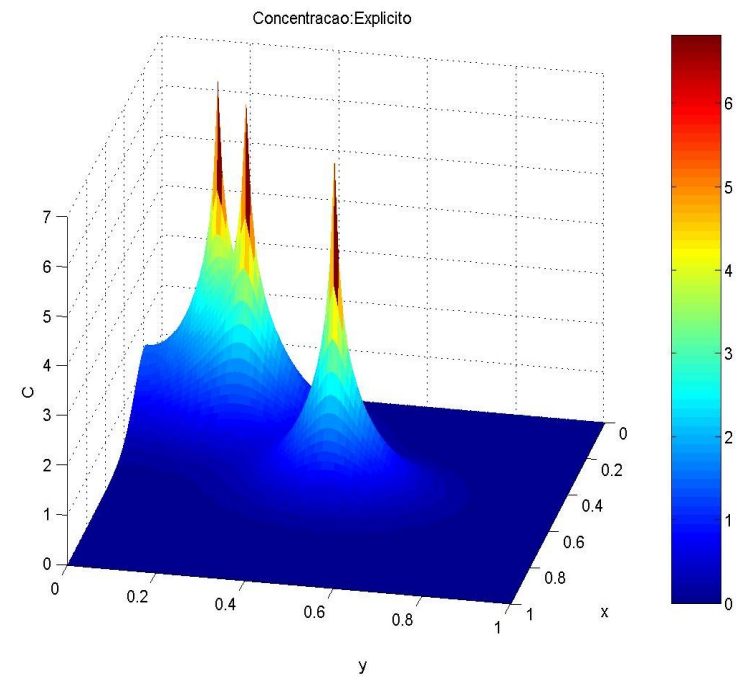
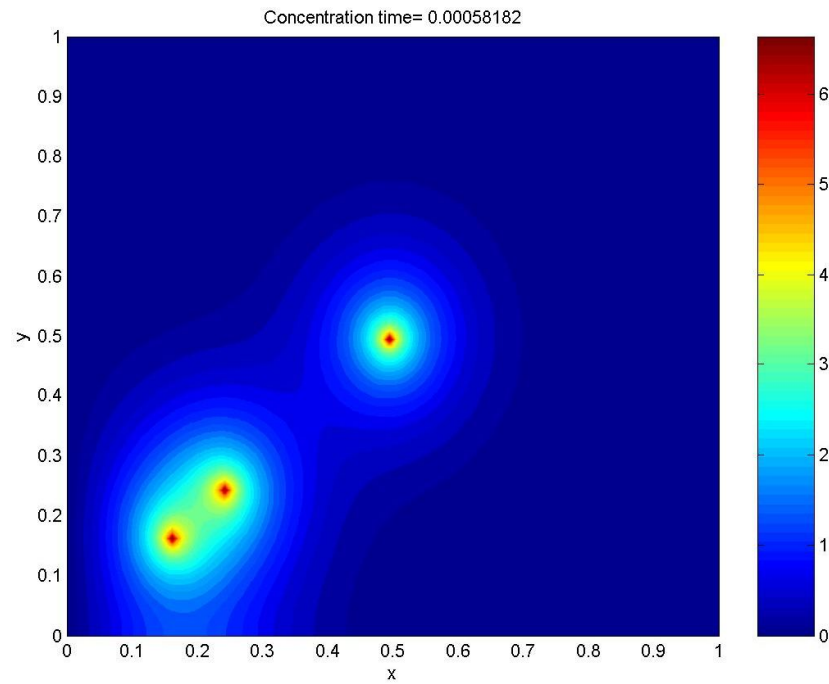


$$\Delta x = \frac{L}{(N_x - 1)}, \quad \Delta y = \frac{B}{(N_y - 1)}$$

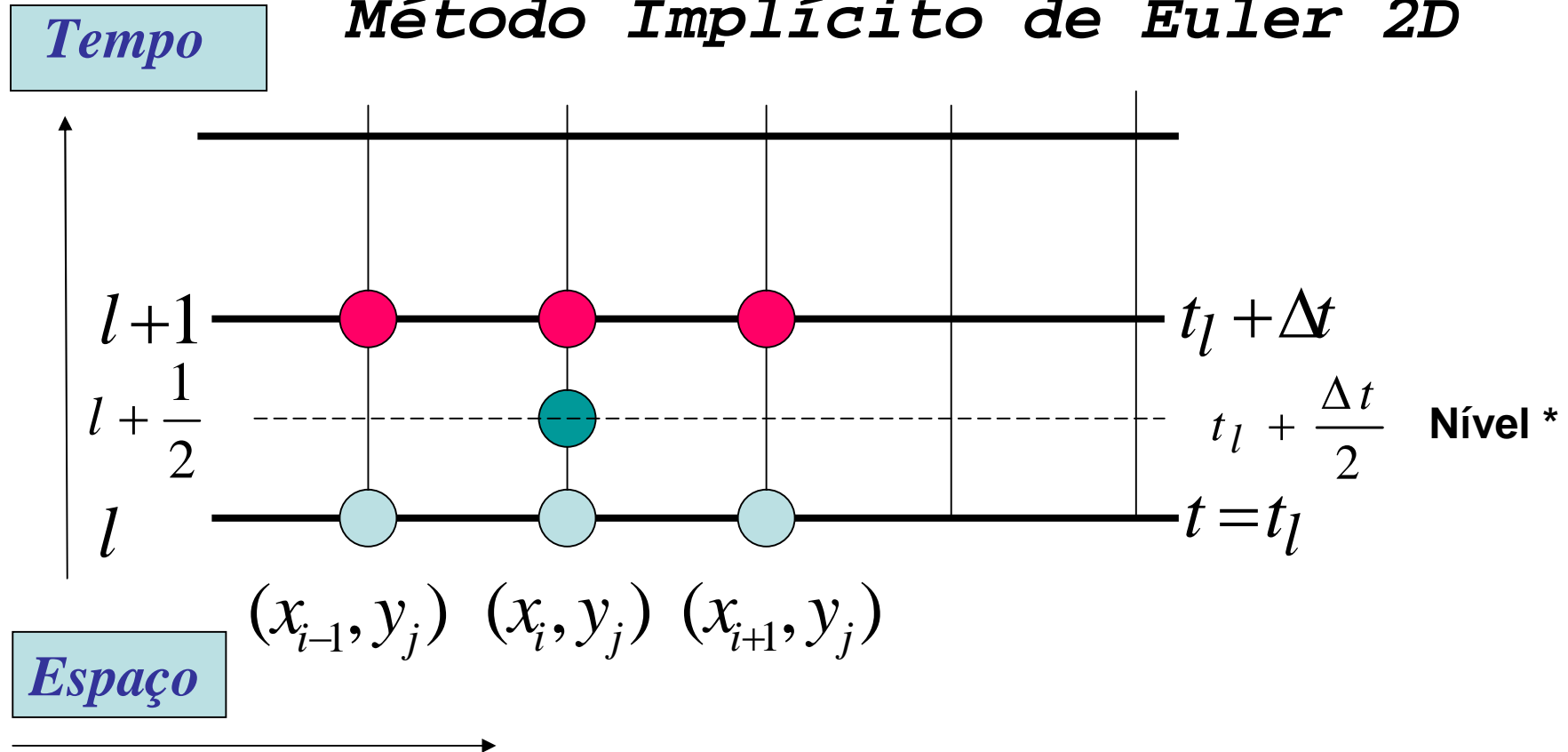
Resultados método de *Euler explícito 2D*



Resultados método de *Euler explícito 2D*



Método Implícito de Euler 2D



Esquema para equação de diferença finita no passo *intermediário* de tempo de $l+1/2$ e nódo (i,j) .

Nível * => Usa-se o conceito de passo no tempo intermediário, para varrer as direções x e y

O método adotado: *implícito de Euler 2D*

1° Sweep na direção x

$$-A c_{i-1,j}^{*m+1} + (1+p) c_{i,j}^{*m+1} + B c_{i+1,j}^{*m+1} = c_i^m + rhs_{i,j}^m$$

$$p = \frac{D_x \Delta t}{\Delta x^2} \quad A = \frac{\Delta t u}{4\Delta x} + \frac{D_x \Delta t}{2(\Delta x)^2} \quad B = \frac{\Delta t u}{4\Delta x} - \frac{D_x \Delta t}{2(\Delta x)^2}$$

Para as Condições de contorno adota é possível escrever:

$$A \cdot c^{*m+1} = c^m + rhs_{i,j}^m$$

Condições de contorno: $C(1)=C(2)+A \cdot C(1)+rhs(1)$

$C(Nx-2)=C(Nx-1)-B \cdot C(Nx)+rhs(Nx-1)$

Lado direito – rhs para direção x

$$rhs(i, j) = c_{i,j}^n - \frac{v\Delta t}{4\Delta y} (c_{i,j+1}^n - c_{i,j-1}^n) + \frac{D_y \Delta t}{2\Delta y^2} (c_{i,j-1}^n - 2c_{i,j}^n + c_{i,j+1}^n)$$

O método adotado: *implícito de Euler 2D*

2° Sweep na direção y

$$-A c_{i,j-1}^{m+1} + (1+p)c_{i,j}^{m+1} + B c_{i,j+1}^{m+1} = c_{i,j}^{*m+1} + rhs_{i,j}^{*m+1}$$

$$p = \frac{D_y \Delta t}{\Delta y^2} \quad A = \frac{\Delta t v}{4\Delta y} + \frac{D_y \Delta t}{2(\Delta y)^2} \quad B = \frac{\Delta t v}{4\Delta y} - \frac{D_y \Delta t}{2(\Delta y)^2}$$

Para as Condições de contorno adota é possível escrever:

$$A.c^{m+1} = c^{*m+1} + rhs_{i,j}^{*m+1}$$

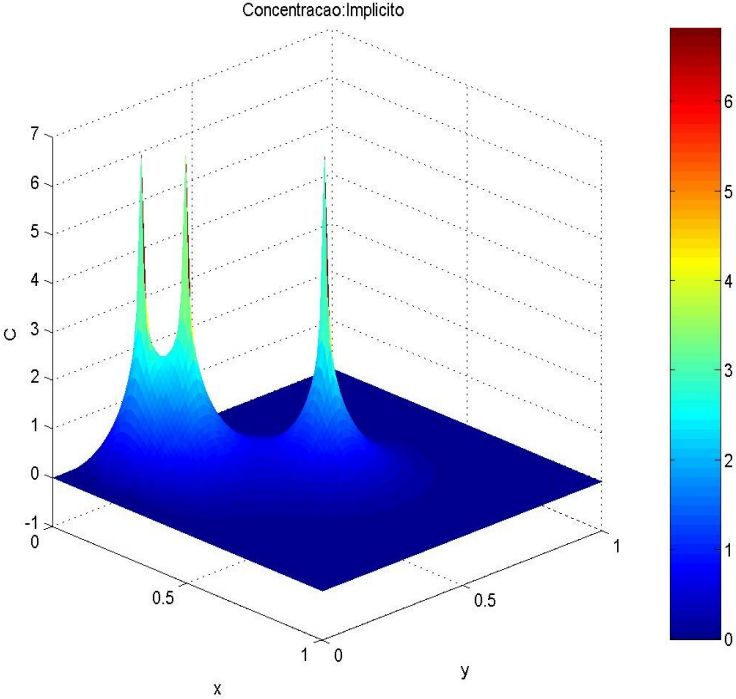
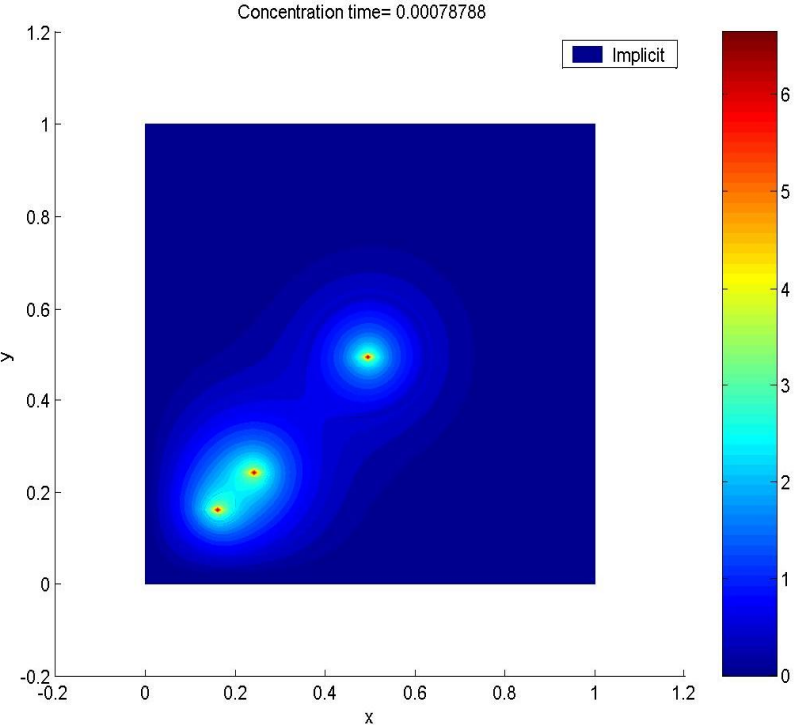
Condições de contorno: $C(1)=C(2)+A*C(1)+rhs(1)$

$C(Ny-2)=C(Ny-1)-B*C(Ny)+rhs(Ny-1)$

Lado direito – rhs para direção y

$$rhs(i, j) = c_{i,j}^{*m+1} - \frac{u\Delta t}{4\Delta x} (c_{i+1,j}^{*m+1} - c_{i-1,j}^{*m+1}) + \frac{D_x \Delta t}{2\Delta x^2} (c_{i-1,j}^{*m+1} - 2c_{i,j}^{*m+1} + c_{i+1,j}^{*m+1})$$

Resultados método de *Euler implícito 2D*



Conclusão

- Os métodos explícitos são mais restritivos em termos de estabilidade e portanto requer mais CPU
- Os métodos Implícitos são menos restritivos em termos de estabilidade e portanto requer menos CPU

REFERÊNCIA BIBLIOGRÁFICA

- [1] - Merle C. Potter, David C. Wiggert, página 588.
Mecânica dos Fluidos, Editora Thomson, Brasil. ISN 85-221-0309-7.
- [2] - G. I. Taylor, Proc. Roy. Soc. Lond. A **219**, 186 (1953).
- [3] - R. Aris, Proc. Roy. Soc. Lond. A **235**, 67 (1956).