

# Modelação Numérica 2017

## Aula 14, 4/Abr

- Equação de Fourier da condução de calor/ Lei de Fick da difusão
- Solução estacionária: Equação de Laplace
- Equação de Poisson
- Método de Jacobi
- Condições fronteira

<http://modnum.ucs.ciencias.ulisboa.pt>

# Lei de Fourier da condução

- Na ausência de fontes e sumidouros de calor, a temperatura num meio contínuo satisfaz a equação:

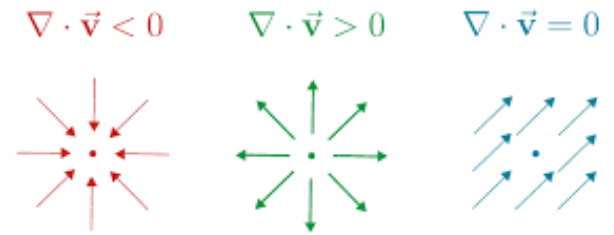
$$\frac{\partial T}{\partial t} = \lambda \left[ \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} \right] = \lambda \nabla^2 T$$

- Onde  $\lambda$  é a **condutividade térmica**. A equação de Fourier pode escrever-se na forma de fluxo:

$$\frac{\partial T}{\partial t} = -\nabla \cdot \vec{F} = -\nabla \cdot (-\lambda \nabla T)$$

- Onde  $F = -\lambda \nabla T$  é o fluxo de calor. Num sólido homogéneo,  $\lambda$  é aproximadamente constante (ou função da temperatura), mas depende do material constitutivo. Num fluído,  $\lambda$  depende do escoamento (para além de condução existe convecção).

# Lei de Fourier da condução



- Na ausência de fontes e sumidouros de calor, a temperatura num meio contínuo satisfaz a equação:

$$\frac{\partial T}{\partial t} = \lambda \left[ \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} \right] = \lambda \nabla^2 T$$

- Onde  $\lambda$  é a **condutividade térmica**. A equação de Fourier pode escrever-se na forma de fluxo:

$$\frac{\partial T}{\partial t} = -\nabla \cdot \vec{F} = -\nabla \cdot (-\lambda \nabla T)$$

- Onde  $F = -\lambda \nabla T$  é o fluxo de calor. Num sólido homogêneo,  $\lambda$  é aproximadamente constante (ou função da temperatura), mas depende do material constitutivo. Num fluido,  $\lambda$  depende do escoamento (para além de condução existe convecção).

# Lei de Fick da difusão

- Num fluido em que se dissolveu um soluto, o seu **fluxo**  $J$  é proporcional ao gradiente da **concentração** ( $\phi$ ):

$$\vec{J} = -D\nabla\phi$$

- Onde  $D$  é a **difusividade**. A concentração satisfaz a lei de Fick:

$$\frac{\partial\phi}{\partial t} = -\nabla\cdot\vec{J} = -\nabla\cdot(-D\nabla\phi) = D \left[ \frac{\partial^2\phi}{\partial x^2} + \frac{\partial^2\phi}{\partial y^2} + \frac{\partial^2\phi}{\partial z^2} \right] = D\nabla^2\phi$$

- A última passagem só é válida com  $D$  constante.

## Limite estacionário

- As leis de Fourier e de Fick são **idênticas** do ponto de vista da sua formulação matemática. Em ambos os casos, o campo da variável de controlo ( $T$  ou  $\phi$ , a seguir designada por  $V$ ) evoluirá enquanto o seu laplaciano não for nulo.
- No limite estacionário (não há evolução no tempo), pode-se calcular a distribuição do campo resolvendo a equação de Laplace:

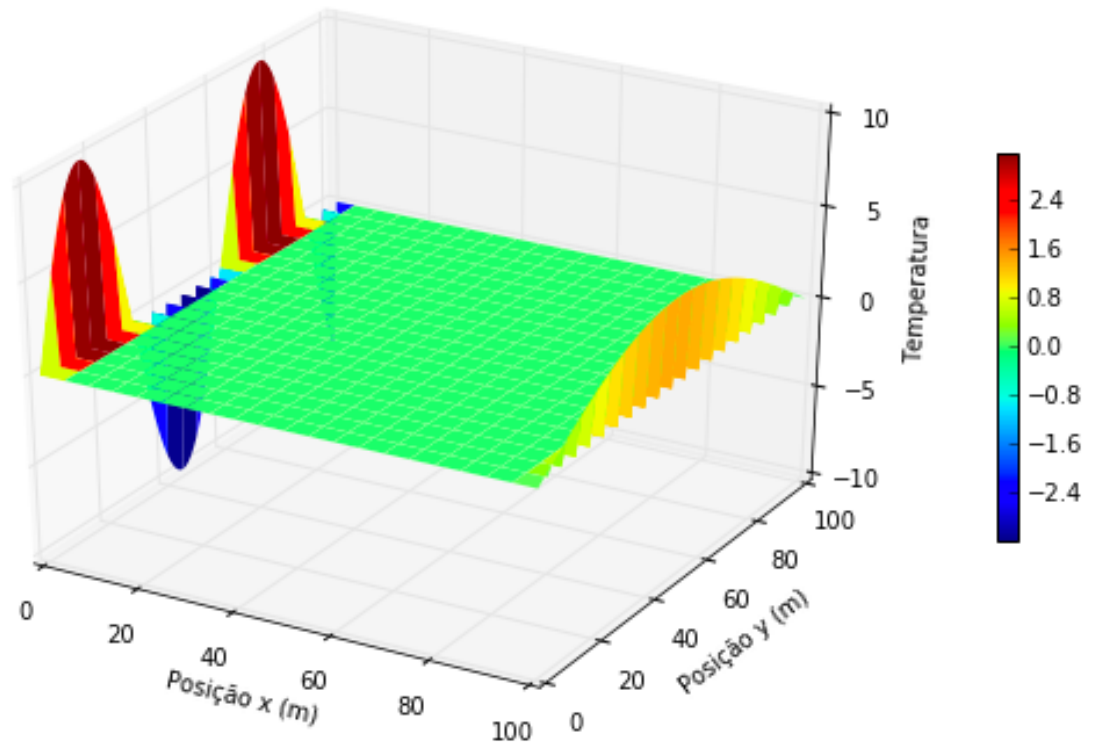
$$\frac{\partial V}{\partial t} = 0 \implies \nabla^2 V = 0 \Leftrightarrow \frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2} + \frac{\partial^2 V}{\partial z^2} = 0$$

- A equação de Laplace é um exemplo de uma equação diferencial às derivadas parciais cuja solução é um problema de condições fronteira espaciais. Na nomenclatura das equações diferenciais às derivadas parciais de segunda ordem, é uma equação elíptica.

# Exemplo: Temperatura de equilíbrio de uma placa

$$\frac{\partial V}{\partial t} = 0 \implies \nabla^2 V = 0 \Leftrightarrow \frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2} + \frac{\partial^2 V}{\partial z^2} = 0$$

$$\frac{\partial T}{\partial t} = \lambda \nabla^2 T = 0$$



# Equação de Poisson

- A solução numérica da equação de Laplace pode ser obtida como um caso particular de uma equação mais geral, a [equação de Poisson](#):

$$\nabla^2 V = f \Leftrightarrow \frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2} + \frac{\partial^2 V}{\partial z^2} = f$$

- Em que  $f(x, y, z)$  é um campo forçador. Tal como a equação de Laplace, a equação de Poisson também é elíptica, sendo a sua solução um problema de condições fronteira espaciais. No caso da equação de Laplace,  $f = 0$ .

# Teoria do potencial

- Potencial eléctrico:

$$\frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2} + \frac{\partial^2 V}{\partial z^2} = -\frac{\rho}{\epsilon}$$

Onde  $\rho(x, y, z)$  é a densidade de carga eléctrica (pode ser positiva ou negativa) e  $\epsilon$  é a permitividade eléctrica do meio.

- Potencial gravítico:

$$\frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2} + \frac{\partial^2 V}{\partial z^2} = 4\pi G\rho$$

Onde  $\rho(x, y, z)$  é a massa volúmica e  $G$  é a constante gravitacional



# Discretização

- Vamos considerar o problema bidimensional. Na vizinhança do ponto  $(x, y)$  podemos escrever a série de Taylor:

$$\begin{aligned} f(x_0 + \Delta x) &= f(x_0) + \left(\frac{\partial f}{\partial x}\right)_{x=x_0} \Delta x + \frac{1}{2} \left(\frac{\partial^2 f}{\partial x^2}\right)_{x=x_0} \Delta x^2 + \frac{1}{3!} \left(\frac{\partial^3 f}{\partial x^3}\right)_{x=x_0} \Delta x^3 + \dots \\ + f(x_0 - \Delta x) &= f(x_0) - \left(\frac{\partial f}{\partial x}\right)_{x=x_0} \Delta x + \frac{1}{2} \left(\frac{\partial^2 f}{\partial x^2}\right)_{x=x_0} \Delta x^2 - \frac{1}{3!} \left(\frac{\partial^3 f}{\partial x^3}\right)_{x=x_0} \Delta x^3 + \dots \end{aligned}$$

---

$$f(x_0 + \Delta x) + f(x_0 - \Delta x) = 2f(x_0) + \left(\frac{\partial^2 f}{\partial x^2}\right)_{x=x_0} \Delta x^2 + \frac{2}{4!} \left(\frac{\partial^4 f}{\partial x^4}\right)_{x=x_0} \Delta x^4 + \dots$$

# Discretização

- Vamos considerar o problema bidimensional. Na vizinhança do ponto  $(x, y)$  podemos escrever a série de Taylor:

$$\begin{aligned} f(x_0 + \Delta x) &= f(x_0) + \cancel{\left(\frac{\partial f}{\partial x}\right)_{x=x_0} \Delta x} + \frac{1}{2} \left(\frac{\partial^2 f}{\partial x^2}\right)_{x=x_0} \Delta x^2 + \cancel{\frac{1}{3!} \left(\frac{\partial^3 f}{\partial x^3}\right)_{x=x_0} \Delta x^3} + \dots \\ + f(x_0 - \Delta x) &= f(x_0) - \cancel{\left(\frac{\partial f}{\partial x}\right)_{x=x_0} \Delta x} + \frac{1}{2} \left(\frac{\partial^2 f}{\partial x^2}\right)_{x=x_0} \Delta x^2 - \cancel{\frac{1}{3!} \left(\frac{\partial^3 f}{\partial x^3}\right)_{x=x_0} \Delta x^3} + \dots \end{aligned}$$

---

$$f(x_0 + \Delta x) + f(x_0 - \Delta x) = 2f(x_0) + \left(\frac{\partial^2 f}{\partial x^2}\right)_{x=x_0} \Delta x^2 + \frac{2}{4!} \left(\frac{\partial^4 f}{\partial x^4}\right)_{x=x_0} \Delta x^4 + \dots$$

# Discretização

- Vamos considerar o problema bidimensional. Na vizinhança do ponto  $(x, y)$  podemos escrever a série de Taylor:

$$\begin{aligned} f(x_0 + \Delta x) &= f(x_0) + \left(\frac{\partial f}{\partial x}\right)_{x=x_0} \Delta x + \frac{1}{2} \left(\frac{\partial^2 f}{\partial x^2}\right)_{x=x_0} \Delta x^2 + \frac{1}{3!} \left(\frac{\partial^3 f}{\partial x^3}\right)_{x=x_0} \Delta x^3 + \dots \\ + f(x_0 - \Delta x) &= f(x_0) - \left(\frac{\partial f}{\partial x}\right)_{x=x_0} \Delta x + \frac{1}{2} \left(\frac{\partial^2 f}{\partial x^2}\right)_{x=x_0} \Delta x^2 - \frac{1}{3!} \left(\frac{\partial^3 f}{\partial x^3}\right)_{x=x_0} \Delta x^3 + \dots \end{aligned}$$

---

$$f(x_0 + \Delta x) + f(x_0 - \Delta x) = 2f(x_0) + \left(\frac{\partial^2 f}{\partial x^2}\right)_{x=x_0} \Delta x^2 + \frac{2}{4!} \left(\frac{\partial^4 f}{\partial x^4}\right)_{x=x_0} \Delta x^4 + \dots$$

$$\begin{aligned} \left(\frac{\partial^2 f}{\partial x^2}\right)_{x=x_0} &= \frac{f(x_0 + \Delta x) + f(x_0 - \Delta x) - 2f(x_0) - \frac{2}{4!} \left(\frac{\partial^4 f}{\partial x^4}\right)_{x=x_0} \Delta x^4 - \dots}{\Delta x^2} \\ &= \frac{f(x_0 + \Delta x) + f(x_0 - \Delta x) - 2f(x_0)}{\Delta x^2} - \frac{2}{4!} \left(\frac{\partial^4 f}{\partial x^4}\right)_{x=x_0} \Delta x^2 - \dots \\ &= \frac{f(x_0 + \Delta x) - 2f(x_0) + f(x_0 - \Delta x)}{\Delta x^2} - \frac{2}{4!} \left(\frac{\partial^4 f}{\partial x^4}\right)_{x=x_0} \Delta x^2 - \dots \end{aligned}$$

# Discretização

- Justificando a aproximação:

$$\frac{\partial^2 V}{\partial x^2} = \frac{V_{i+1,j} - 2V_{i,j} + V_{i-1,j}}{\Delta x^2} - \frac{2}{4!} \frac{\partial^4 f}{\partial x^4} \Delta x^2 - \dots = \frac{V_{i+1,j} - 2V_{i,j} + V_{i-1,j}}{\Delta x^2} + \mathcal{O}(\Delta x^2)$$

$$\begin{aligned} \left( \frac{\partial^2 f}{\partial x^2} \right)_{x=x_0} &= \frac{f(x_0 + \Delta x) + f(x_0 - \Delta x) - 2f(x_0) - \frac{2}{4!} \left( \frac{\partial^4 f}{\partial x^4} \right)_{x=x_0} \Delta x^4 - \dots}{\Delta x^2} \\ &= \frac{f(x_0 + \Delta x) + f(x_0 - \Delta x) - 2f(x_0)}{\Delta x^2} - \frac{2}{4!} \left( \frac{\partial^4 f}{\partial x^4} \right)_{x=x_0} \Delta x^2 - \dots \\ &= \frac{f(x_0 + \Delta x) - 2f(x_0) + f(x_0 - \Delta x)}{\Delta x^2} - \frac{2}{4!} \left( \frac{\partial^4 f}{\partial x^4} \right)_{x=x_0} \Delta x^2 - \dots \end{aligned}$$

# Discretização

- Justificando a aproximação:

$$\frac{\partial^2 V}{\partial x^2} = \frac{V_{i+1,j} - 2V_{i,j} + V_{i-1,j}}{\Delta x^2} - \frac{2}{4!} \frac{\partial^4 f}{\partial x^4} \Delta x^2 - \dots = \frac{V_{i+1,j} - 2V_{i,j} + V_{i-1,j}}{\Delta x^2} + \mathcal{O}(\Delta x^2)$$

$$\left\{ \begin{array}{l} \frac{\partial^2 V}{\partial x^2} \approx \frac{V_{i+1,j} - 2V_{i,j} + V_{i-1,j}}{\Delta x^2} \\ \frac{\partial^2 V}{\partial y^2} \approx \frac{V_{i,j+1} - 2V_{i,j} + V_{i,j-1}}{\Delta x^2} \end{array} \right.$$

Para:

$$i=2, \dots, nx-1$$

$$j=2, \dots, ny-1$$

$$\left\{ \begin{array}{l} V_{i,j} = V(x_i, y_i) \\ x_i = (i-1)\Delta x; \quad i = 1, \dots, nx \\ y_i = (i-1)\Delta y; \quad j = 1, \dots, ny \end{array} \right.$$

Os pontos de **fronteira** terão de ser tratados à parte.

# Solução iterativa pelo Método Iterativo de Jacobi

- O objectivo é calcular  $V$ . Vamos considerar que as condições de fronteira são de Dirichlet, isto é, o campo está especificado ao longo da fronteira:

$$V(x = 0) = V_W(y)$$

$$V(x = L_x) = V_E(y)$$

$$V(y = 0) = V_S(y)$$

$$V(y = L_y) = V_N(y)$$

- Sendo os valores interiores arbitrários.

# Solução iterativa pelo Método Iterativo de Jacobi

- Começamos por discretizar a equação de Poisson:

$$\frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2} + \cancel{\frac{\partial^2 V}{\partial z^2}} = f \quad \text{Problema 2D}$$

$$\frac{V_{i+1,j} - 2V_{i,j} + V_{i-1,j}}{\Delta x^2} + \frac{V_{i,j+1} - 2V_{i,j} + V_{i,j-1}}{\Delta y^2} = f_{i,j}$$

$$\left[ \frac{V_{i+1,j} - 2V_{i,j} + V_{i-1,j}}{\Delta x^2} + \frac{V_{i,j+1} - 2V_{i,j} + V_{i,j-1}}{\Delta y^2} = f_{i,j} \right] \times \Delta x^2$$

$$V_{i+1,j} - 2V_{i,j} + V_{i-1,j} + \frac{\Delta x^2}{\Delta y^2} (V_{i,j+1} - 2V_{i,j} + V_{i,j-1}) = \Delta x^2 f_{i,j}$$

$$V_{i+1,j} + V_{i-1,j} + \frac{\Delta x^2}{\Delta y^2} (V_{i,j+1} + V_{i,j-1}) - 2V_{i,j} \left( 1 + \frac{\Delta x^2}{\Delta y^2} \right) = \Delta x^2 f_{i,j}$$

$$V_{i,j} = \frac{V_{i+1,j} + V_{i-1,j} + \frac{\Delta x^2}{\Delta y^2} (V_{i,j+1} + V_{i,j-1}) - \Delta x^2 f_{i,j}}{2 \left( 1 + \frac{\Delta x^2}{\Delta y^2} \right)}$$

# Solução iterativa pelo Método Iterativo de Jacobi

$$V_{i,j} = \frac{V_{i+1,j} + V_{i-1,j} + \frac{\Delta x^2}{\Delta y^2}(V_{i,j+1} + V_{i,j-1}) - \Delta x^2 f_{i,j}}{2(1 + \frac{\Delta x^2}{\Delta y^2})}$$

- Se soubessemos calcular o lado direito, poderíamos calcular o lado esquerdo da equação. Como não sabemos, temos de resolver simultaneamente a equação para todos os nodos. O problema é um problema de álgebra linear.
- Podemos utilizar o método iterativo de Jacobi para resolver simultaneamente todas as equações. Começamos por considerar uma solução arbitrária para o problema ( $V=V^0$ ), para todos os pontos no interior do domínio. Colocamos estes valores no lado direito da equação:

$$V_{i,j}^{n+1} = \frac{V_{i+1,j}^n + V_{i-1,j}^n + \frac{\Delta x^2}{\Delta y^2}(V_{i,j+1}^n + V_{i,j-1}^n) - \Delta x^2 f_{i,j}}{2(1 + \frac{\Delta x^2}{\Delta y^2})}$$



# Solução iterativa pelo Método Iterativo de Jacobi

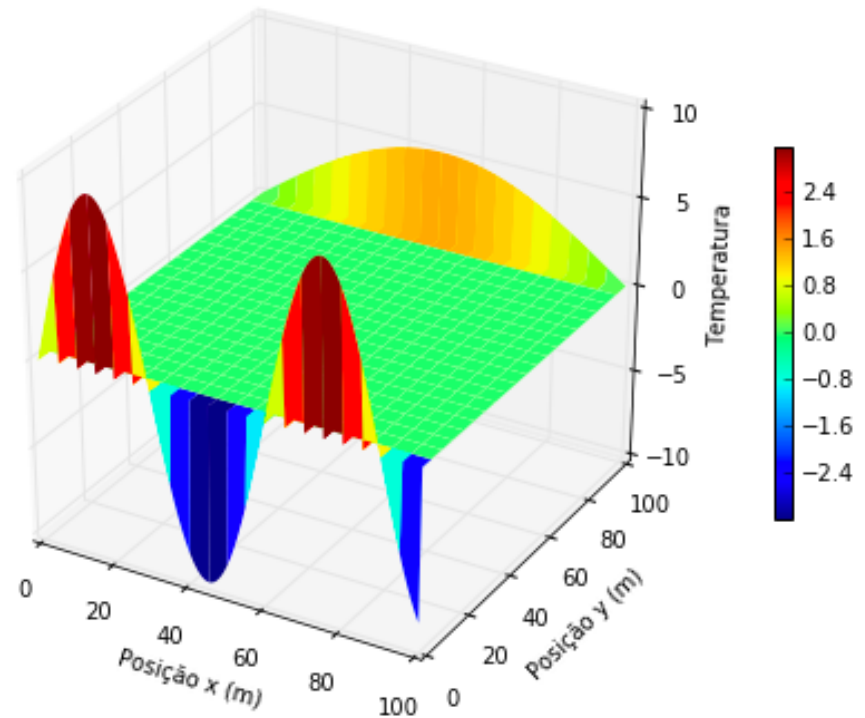
$$V_{i,j}^{n+1} = \frac{V_{i+1,j}^n + V_{i-1,j}^n + \frac{\Delta x^2}{\Delta y^2} (V_{i,j+1}^n + V_{i,j-1}^n) - \Delta x^2 f_{i,j}}{2(1 + \frac{\Delta x^2}{\Delta y^2})}$$

- O procedimento repete-se até se obter convergência (quando a solução entre dois passos consecutivos deixa de variar significativamente).

# Exemplo: Temperatura de equilíbrio de uma placa

$$\frac{\partial V}{\partial t} = 0 \implies \nabla^2 V = 0 \Leftrightarrow \frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2} + \frac{\partial^2 V}{\partial z^2} = 0$$

$$\frac{\partial T}{\partial t} = \lambda \nabla^2 T = 0$$

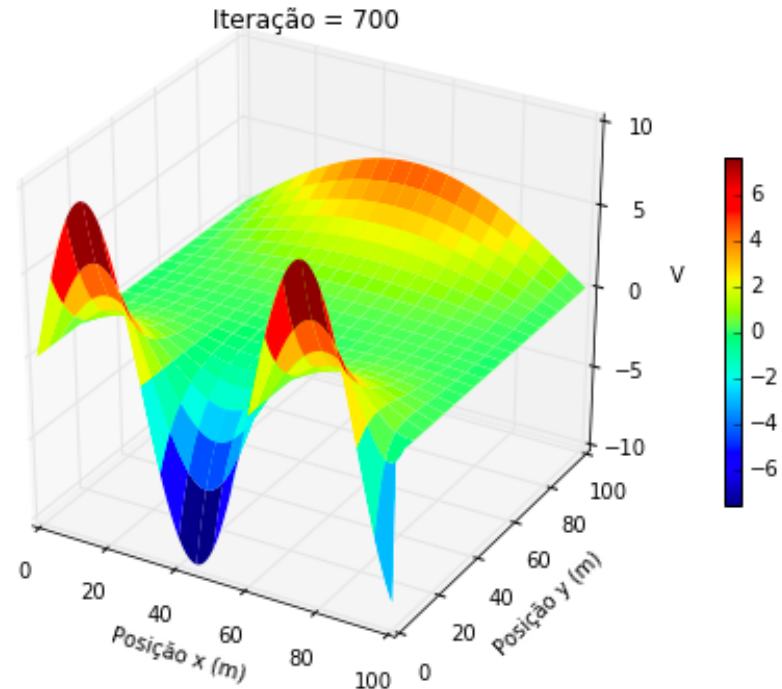


$$\begin{matrix} V=T \\ f=0 \end{matrix} \rightarrow \frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2} + \frac{\partial^2 V}{\partial z^2} = f \rightarrow V_{i,j}^{n+1} = \frac{V_{i+1,j}^n + V_{i-1,j}^n + \frac{\Delta x^2}{\Delta y^2}(V_{i,j+1}^n + V_{i,j-1}^n) - \Delta x^2 f_{i,j}}{2(1 + \frac{\Delta x^2}{\Delta y^2})}$$

# Exemplo: Temperatura de equilíbrio de uma placa

$$\frac{\partial V}{\partial t} = 0 \implies \nabla^2 V = 0 \Leftrightarrow \frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2} + \frac{\partial^2 V}{\partial z^2} = 0$$

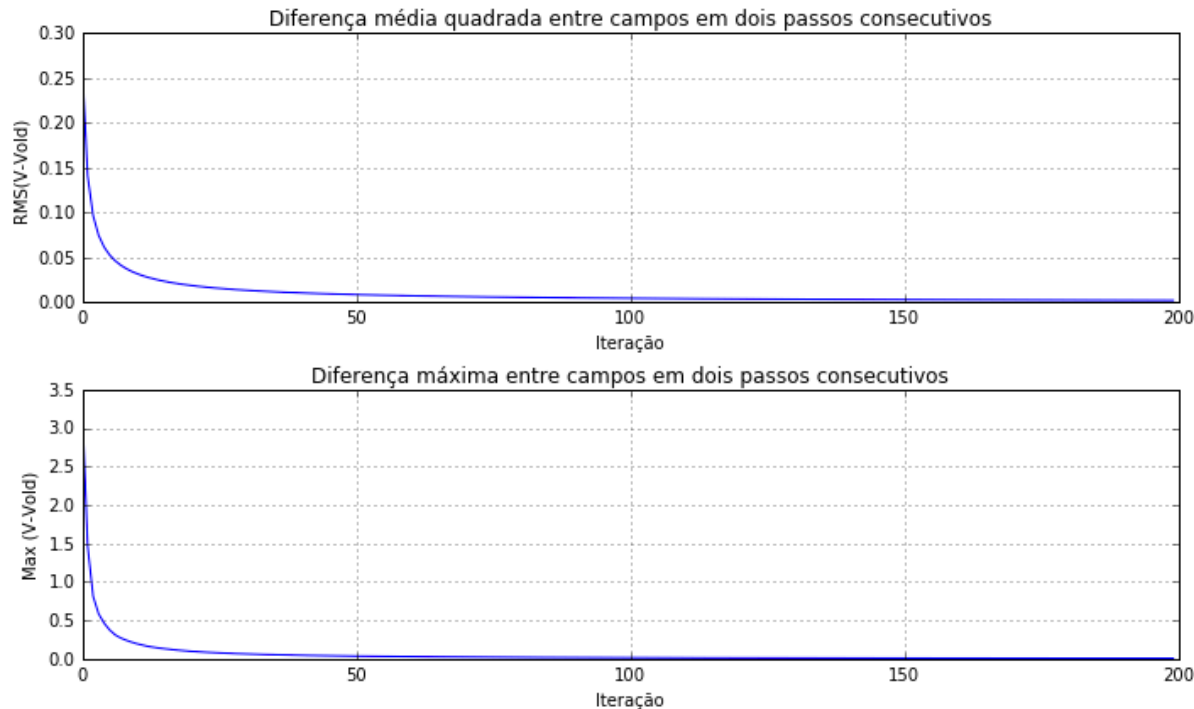
$$\frac{\partial T}{\partial t} = \lambda \nabla^2 T = 0$$



$$\begin{matrix} V=T \\ f=0 \end{matrix} \rightarrow \frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2} + \frac{\partial^2 V}{\partial z^2} = f \rightarrow V_{i,j}^{n+1} = \frac{V_{i+1,j}^n + V_{i-1,j}^n + \frac{\Delta x^2}{\Delta y^2} (V_{i,j+1}^n + V_{i,j-1}^n) - \Delta x^2 f_{i,j}}{2(1 + \frac{\Delta x^2}{\Delta y^2})}$$

# Exemplo: Temperatura de equilíbrio de uma placa

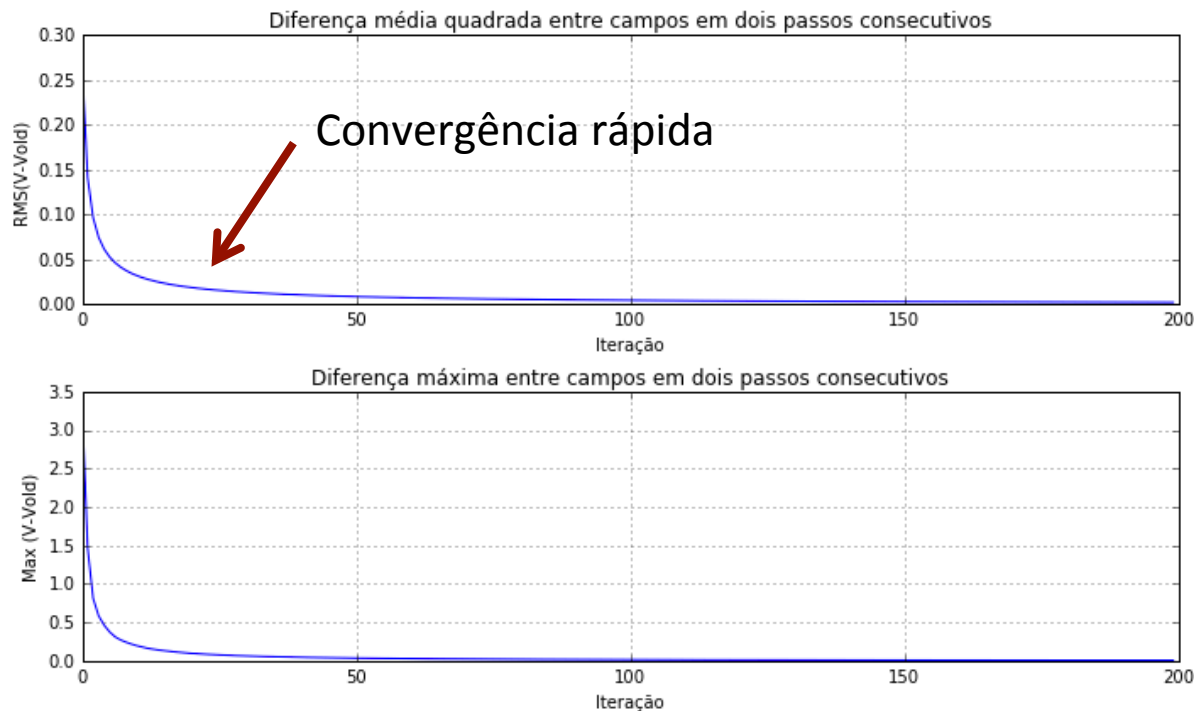
$$\frac{\partial V}{\partial t} = 0 \implies \nabla^2 V = 0 \Leftrightarrow \frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2} + \frac{\partial^2 V}{\partial z^2} = 0$$



$$\begin{matrix} V=T \\ f=0 \end{matrix} \rightarrow \frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2} + \frac{\partial^2 V}{\partial z^2} = f \rightarrow V_{i,j}^{n+1} = \frac{V_{i+1,j}^n + V_{i-1,j}^n + \frac{\Delta x^2}{\Delta y^2} (V_{i,j+1}^n + V_{i,j-1}^n) - \Delta x^2 f_{i,j}}{2(1 + \frac{\Delta x^2}{\Delta y^2})}$$

# Exemplo: Temperatura de equilíbrio de uma placa

$$\frac{\partial V}{\partial t} = 0 \implies \nabla^2 V = 0 \Leftrightarrow \frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2} + \frac{\partial^2 V}{\partial z^2} = 0$$



$$\begin{matrix} V=T \\ f=0 \end{matrix} \rightarrow \frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2} + \frac{\partial^2 V}{\partial z^2} = f \rightarrow V_{i,j}^{n+1} = \frac{V_{i+1,j}^n + V_{i-1,j}^n + \frac{\Delta x^2}{\Delta y^2}(V_{i,j+1}^n + V_{i,j-1}^n) - \Delta x^2 f_{i,j}}{2(1 + \frac{\Delta x^2}{\Delta y^2})}$$

```

import matplotlib.pyplot as plt
import numpy as np
from mpl_toolkits.mplot3d import Axes3D
from matplotlib import cm
from math import pi as pi

plt.rcParams['figure.figsize'] = 10, 6

### Parâmetros

dx=1.          # espaçamento entre nodos na direcção x
dy=1.          # espaçamento entre nodos na direcção x
dx2dy2=dx**2/dy**2
a=2.*(1+dx2dy2)

x=np.arange(0.,101.,dx); nx=len(x)      # vector x
y=np.arange(0.,101.,dy); ny=len(y)      # vector y

V=np.zeros([nx,ny])                    # matriz do campo a determinar
f=np.zeros([nx,ny])                    # matriz do forçamento

it=0                                     # iteração

Vold=np.zeros([nx,ny])                 # campo no passo anterior
DifAll=np.array([])                    # diferença entre campos em dois passos consecutivos, considerando tod
DifMax=np.array([])                    # diferença entre campos em dois passos consecutivos, considerando ape

### Condições fronteira

T1=60.          # período do sinal na fronteira 1
T2=200.         # período do sinal na fronteira 2

V[0,:]=0.
V[nx-1,:]=0
V[:,0]=10.*np.sin(2*pi/T1*x)
V[:,ny-1]=5.*np.sin(2*pi/T2*x)

```

```

# %% Plot 3D

fig = plt.figure()
ax = plt.axes(projection='3d')

xx, yy = np.meshgrid(x, y)
surf=ax.plot_surface(xx,yy, V, rstride=5, cstride=5, cmap=cm.jet, linewidth=0)
#ax.set_zlim(0,10)
fig.colorbar(surf, shrink=0.5)

ax.set_xlabel(u'Posição x (m)')
ax.set_ylabel(u'Posição y (m)')
ax.set_zlabel(u'Temperatura')

# %% Método de Jacobi

niter = 200          # nr de iterações

for i in range(niter):
    it += 1
    Vold[:]=V[:]     # actualizar o campo no passo anterior
    ms=0             # média quadrada
    for ix in range(1,nx-1):
        for iy in range(1,ny-1):
            V[ix,iy] = (V[ix+1,iy] + V[ix-1,iy] + dx2dy2*(V[ix,iy+1] + V[ix,iy-1]) \
                - dx**2*f[ix,iy]) / a
            ms = ms + (V[ix,iy] - Vold[ix,iy])**2          # diferença quadrada entre campos para cad

ms = np.sqrt(ms/(nx*ny))          # raíz quadrada da média quadrada
DifAll = np.append(DifAll,ms)     # vector com a diferença média quadrada entre campos em do

dif = V-Vold                      # diferença entre o campo em dois passos consecutivos
dm = np.max(np.max(dif))          # diferença máxima entre o campo em dois passos consecutiv
DifMax = np.append(DifMax,dm)     # vector com a diferença máxima entre campos em dois passo

```

```

# %% Plot 3D

fig = plt.figure()
ax = plt.axes(projection='3d')

xx, yy = np.meshgrid(x, y)
surf=ax.plot_surface(xx,yy, V, rstride=5, cstride=5, cmap=cm.jet, linewidth=0)
#ax.set_zlim(0,10)
fig.colorbar(surf, shrink=0.5)

ax.set_xlabel(u'Posição x (m)')
ax.set_ylabel(u'Posição y (m)')
ax.set_zlabel(u'V')
ax.set_title(u'Iteração = '+str(it))

#plt.savefig('figura.png')

# %% Plot convergence

plt.close()
plt.rcParams['figure.figsize'] = 10, 6

plt.subplot(2,1,1)
plt.plot(range(it), DifAll)

plt.xlabel(u'Iteração')
plt.ylabel(u'RMS(V-Vold)')
plt.title(u'Diferença média quadrada entre campos em dois passos consecutivos')
plt.grid()

plt.subplot(2,1,2)
plt.plot(range(it), DifMax)

plt.xlabel(u'Iteração')
plt.ylabel(u'Max (V-Vold)')
plt.title(u'Diferença máxima entre campos em dois passos consecutivos')
plt.grid()

plt.tight_layout()

```



# Exemplo: Potencial eléctrico de 3 cargas pontuais

$$\frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2} + \frac{\partial^2 V}{\partial z^2} = -\frac{\rho}{\epsilon}$$

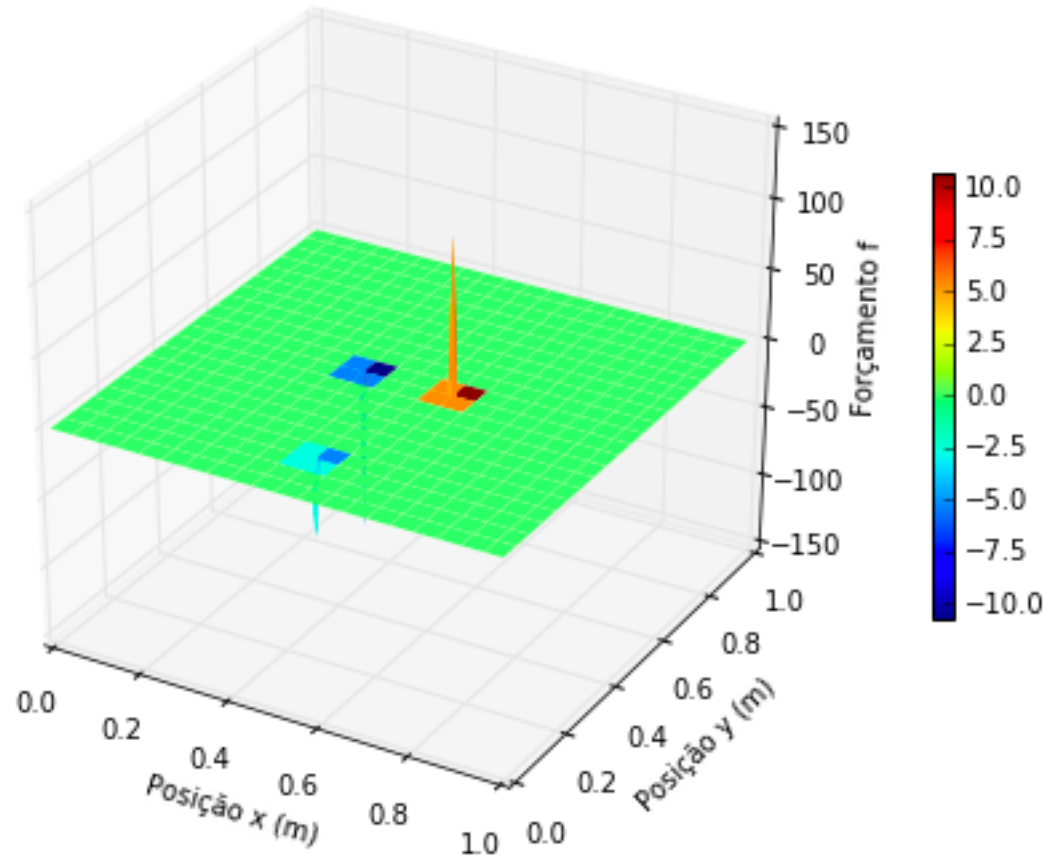
- Três cargas pontuais:

$$\rho = 10e-9, -10e-9, 5e-10 \text{ (C/m}^2\text{)}$$

- Condições fronteira:

$$V_{y=L_y} = 0$$

$$\left(\frac{\partial V}{\partial x}\right)_{x=0, x=L_x} = \left(\frac{\partial V}{\partial y}\right)_{y=0} = 0$$



$$V=V_E \quad f=-\rho/\epsilon \quad \rightarrow \quad \frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2} + \frac{\partial^2 V}{\partial z^2} = f \quad \rightarrow \quad V_{i,j}^{n+1} = \frac{V_{i+1,j}^n + V_{i-1,j}^n + \frac{\Delta x^2}{\Delta y^2}(V_{i,j+1}^n + V_{i,j-1}^n) - \Delta x^2 f_{i,j}}{2(1 + \frac{\Delta x^2}{\Delta y^2})}$$

# Exemplo: Potencial eléctrico de 3 cargas pontuais

$$\frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2} + \frac{\partial^2 V}{\partial z^2} = -\frac{\rho}{\epsilon}$$

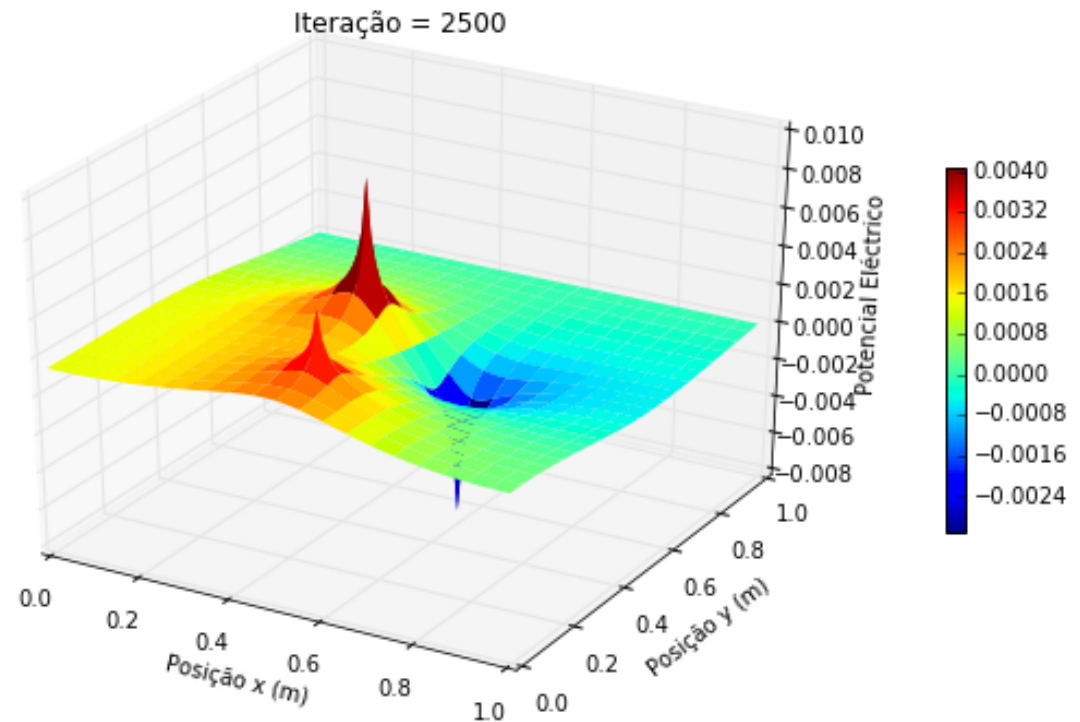
- Três cargas pontuais:

$$\rho = 10e-9, -10e-9, 5e-10 \text{ (C/m}^2\text{)}$$

- Condições fronteira:

$$V_{y=L_y} = 0$$

$$\left(\frac{\partial V}{\partial x}\right)_{x=0, x=L_x} = \left(\frac{\partial V}{\partial y}\right)_{y=0} = 0$$



$$V=V_E \quad f=-\rho/\epsilon \quad \rightarrow \quad \frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2} + \frac{\partial^2 V}{\partial z^2} = f \quad \rightarrow \quad V_{i,j}^{n+1} = \frac{V_{i+1,j}^n + V_{i-1,j}^n + \frac{\Delta x^2}{\Delta y^2}(V_{i,j+1}^n + V_{i,j-1}^n) - \Delta x^2 f_{i,j}}{2(1 + \frac{\Delta x^2}{\Delta y^2})}$$

# Exemplo: Potencial eléctrico de 3 cargas pontuais

$$\frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2} + \frac{\partial^2 V}{\partial z^2} = -\frac{\rho}{\epsilon}$$

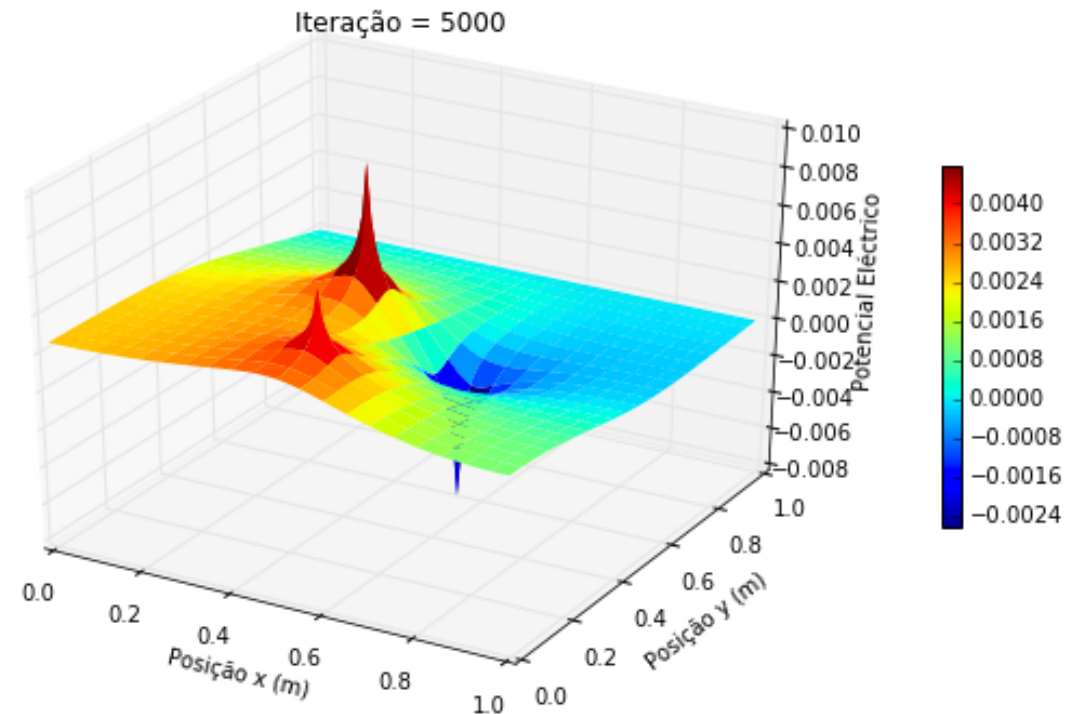
- Três cargas pontuais:

$$\rho = 10e-9, -10e-9, 5e-10 \text{ (C/m}^2\text{)}$$

- Condições fronteira:

$$V_{y=L_y} = 0$$

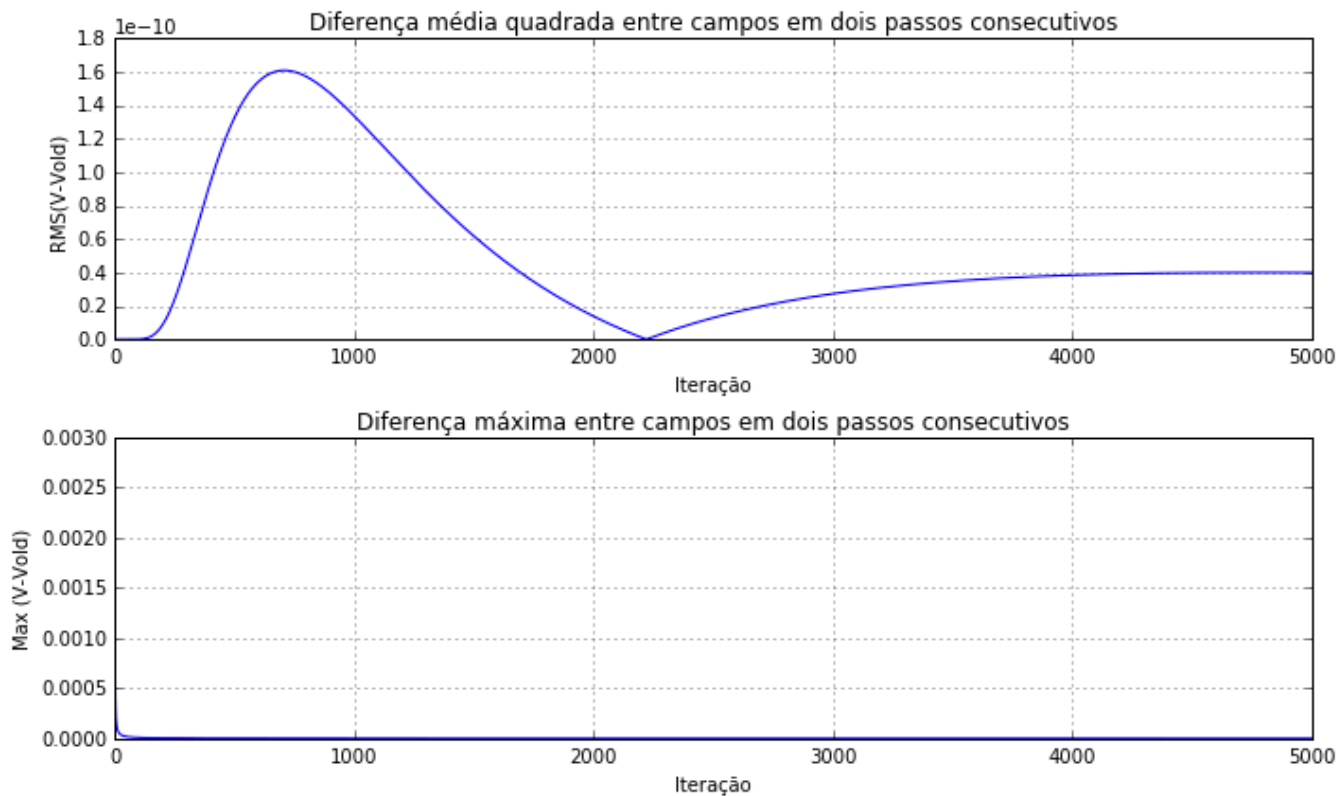
$$\left(\frac{\partial V}{\partial x}\right)_{x=0, x=L_x} = \left(\frac{\partial V}{\partial y}\right)_{y=0} = 0$$



$$V=V_E \quad f=-\rho/\epsilon \quad \rightarrow \quad \frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2} + \frac{\partial^2 V}{\partial z^2} = f \quad \rightarrow \quad V_{i,j}^{n+1} = \frac{V_{i+1,j}^n + V_{i-1,j}^n + \frac{\Delta x^2}{\Delta y^2}(V_{i,j+1}^n + V_{i,j-1}^n) - \Delta x^2 f_{i,j}}{2(1 + \frac{\Delta x^2}{\Delta y^2})}$$

# Exemplo: Potencial eléctrico de 3 cargas pontuais

$$\frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2} + \frac{\partial^2 V}{\partial z^2} = -\frac{\rho}{\epsilon}$$



$$V=V_E \quad f=-\rho/\epsilon \quad \Rightarrow \quad \frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2} + \frac{\partial^2 V}{\partial z^2} = f \quad \Rightarrow \quad V_{i,j}^{n+1} = \frac{V_{i+1,j}^n + V_{i-1,j}^n + \frac{\Delta x^2}{\Delta y^2}(V_{i,j+1}^n + V_{i,j-1}^n) - \Delta x^2 f_{i,j}}{2(1 + \frac{\Delta x^2}{\Delta y^2})}$$

# Exemplo: Potencial eléctrico de 3 cargas pontuais

$$\frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2} + \frac{\partial^2 V}{\partial z^2} = -\frac{\rho}{\epsilon}$$

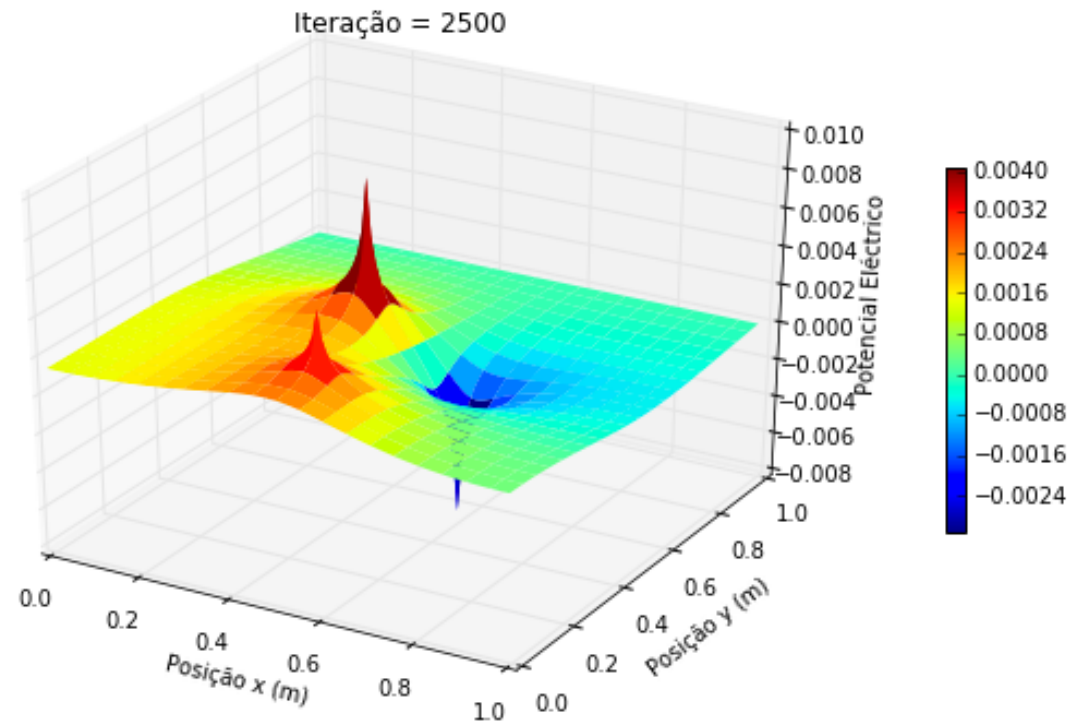
- Três cargas pontuais:

$$\rho = 10e-9, -10e-9, 5e-10 \text{ (C/m}^2\text{)}$$

- Condições fronteira:

$$V_{y=L_y} = 0$$

$$\left(\frac{\partial V}{\partial x}\right)_{x=0, x=L_x} = \left(\frac{\partial V}{\partial y}\right)_{y=0} = 0$$



$$V=V_E \quad f=-\rho/\epsilon \quad \rightarrow \quad \frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2} + \frac{\partial^2 V}{\partial z^2} = f \quad \rightarrow \quad V_{i,j}^{n+1} = \frac{V_{i+1,j}^n + V_{i-1,j}^n + \frac{\Delta x^2}{\Delta y^2}(V_{i,j+1}^n + V_{i,j-1}^n) - \Delta x^2 f_{i,j}}{2(1 + \frac{\Delta x^2}{\Delta y^2})}$$

# Exemplo: Potencial eléctrico de 3 cargas pontuais

$$\frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2} + \frac{\partial^2 V}{\partial z^2} = -\frac{\rho}{\epsilon}$$

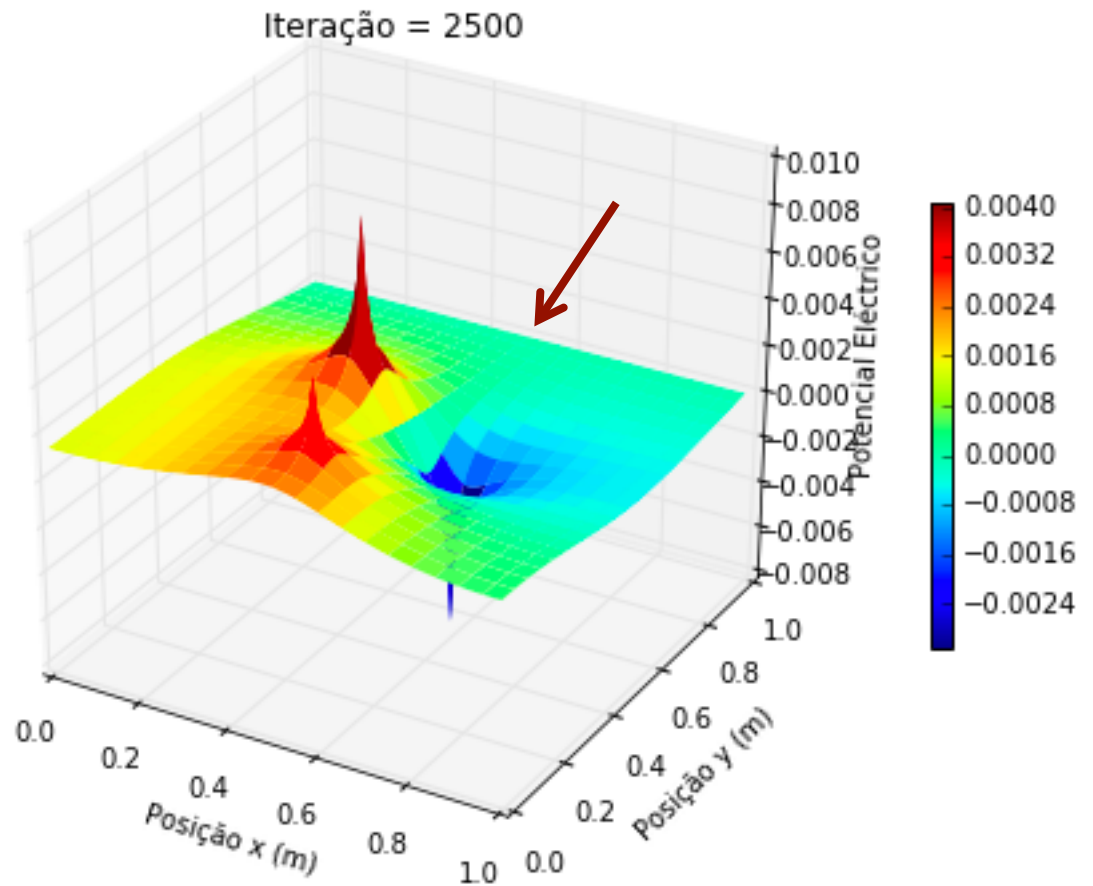
- Três cargas pontuais:

$$\rho = 10e-9, -10e-9, 5e-10 \text{ (C/m}^2\text{)}$$

- Condições fronteira:

$$V_{y=L_y} = 0$$

$$\left(\frac{\partial V}{\partial x}\right)_{x=0, x=L_x} = \left(\frac{\partial V}{\partial y}\right)_{y=0} = 0$$



$$V=V_E \quad f=-\rho/\epsilon \quad \rightarrow \quad \frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2} + \frac{\partial^2 V}{\partial z^2} = f \quad \rightarrow \quad V_{i,j}^{n+1} = \frac{V_{i+1,j}^n + V_{i-1,j}^n + \frac{\Delta x^2}{\Delta y^2}(V_{i,j+1}^n + V_{i,j-1}^n) - \Delta x^2 f_{i,j}}{2(1 + \frac{\Delta x^2}{\Delta y^2})}$$

# Exemplo: Potencial eléctrico de 3 cargas pontuais

$$\frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2} + \frac{\partial^2 V}{\partial z^2} = -\frac{\rho}{\epsilon}$$

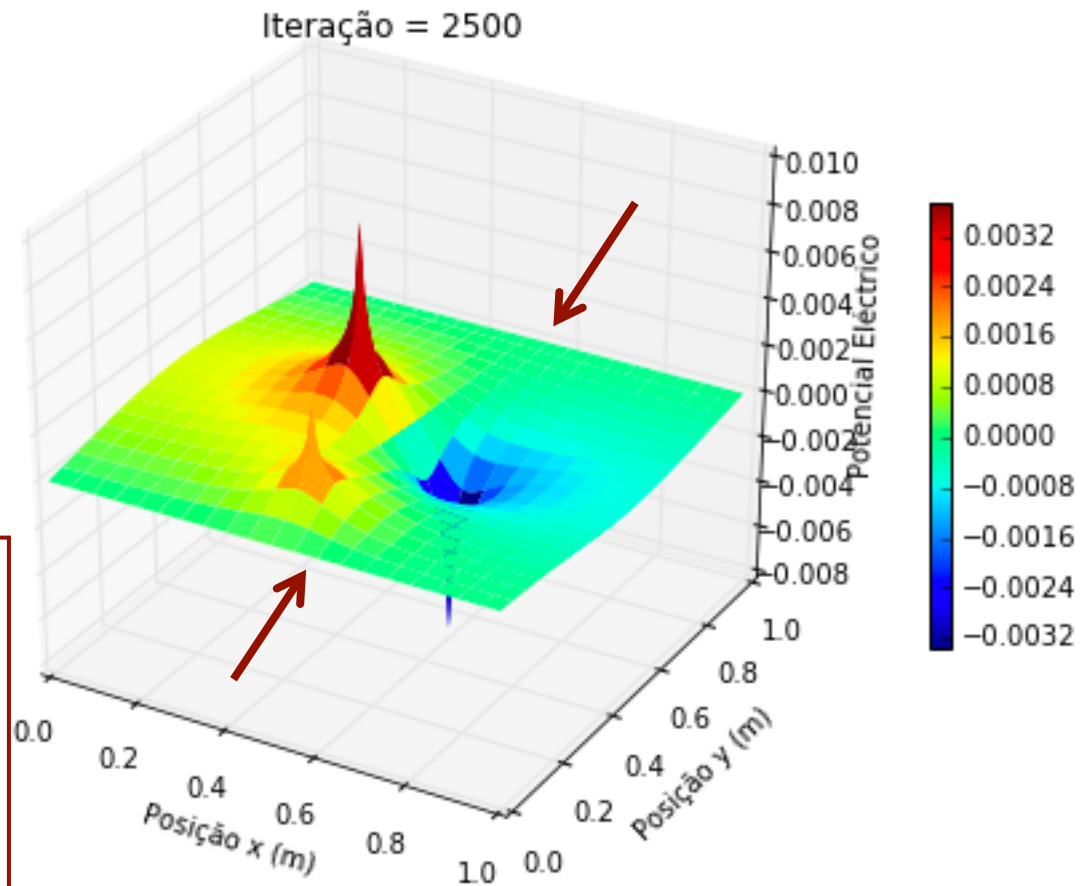
- Três cargas pontuais:

$$\rho = 10e-9, -10e-9, 5e-10 \text{ (C/m}^2\text{)}$$

- Condições fronteira:

$$V_{y=0; y=L_y} = 0$$

$$\left(\frac{\partial V}{\partial x}\right)_{x=0; x=L_x} = 0$$



$$V=V_E \quad f=-\rho/\epsilon \quad \rightarrow \quad \frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2} + \frac{\partial^2 V}{\partial z^2} = f \quad \rightarrow \quad V_{i,j}^{n+1} = \frac{V_{i+1,j}^n + V_{i-1,j}^n + \frac{\Delta x^2}{\Delta y^2}(V_{i,j+1}^n + V_{i,j-1}^n) - \Delta x^2 f_{i,j}}{2(1 + \frac{\Delta x^2}{\Delta y^2})}$$

```

import matplotlib.pyplot as plt
import numpy as np
from mpl_toolkits.mplot3d import Axes3D
from matplotlib import cm
from math import pi as pi

plt.rcParams['figure.figsize'] = 10, 6

# %% Parâmetros

dx=.01          # espaçamento entre nodos na direcção x
dy=.01          # espaçamento entre nodos na direcção y
dx2dy2=dx**2/dy**2
a=2.*(1+dx2dy2)

x=np.arange(0.,1.,dx); nx=len(x)      # vector x
y=np.arange(0.,1.,dy); ny=len(y)      # vector y

V=np.zeros([nx,ny])                   # matriz do campo a determinar
f=np.zeros([nx,ny])                   # matriz do forçamento

c1=1e-9
c2=-1e-9
c3=5e-10

eps=8.88541878172e-12

f[40,50]= -c1/eps
f[60,50]= -c2/eps
f[50,15]= -c3/eps

it=0          # iteração

Vold=np.zeros([nx,ny])                # campo no passo anterior
DifAll=np.array([])                   # diferença entre campos em dois passos consecutivos, considerando to
DifMax=np.array([])                   # diferença entre campos em dois passos consecutivos, considerando ap

```



```
#%% Condições fronteira
```

```
V[0,:]=V[1,:]  
V[nx-1,:]=V[nx-2,:]  
V[:,0]=V[:,1]  
V[:,ny-1]=0
```

```
#%% Plot 3D - potencial
```

```
fig = plt.figure()  
ax = plt.axes(projection='3d')  
  
yy, xx = np.meshgrid(x, y)  
surf=ax.plot_surface(xx,yy, V, rstride=5, cstride=5, cmap=cm.jet, linewidth=0)  
#ax.set_zlim(0,10)  
fig.colorbar(surf, shrink=0.5)  
  
ax.set_xlabel(u'Posição x (m)')  
ax.set_ylabel(u'Posição y (m)')  
ax.set_zlabel(u'Potencial Eléctrico')
```

```
#%% Plot 3D - forçamento
```

```
fig = plt.figure()  
ax = plt.axes(projection='3d')  
  
surf=ax.plot_surface(xx,yy, f, rstride=5, cstride=5, cmap=cm.jet, linewidth=0)  
#ax.set_zlim(0,10)  
fig.colorbar(surf, shrink=0.5)  
  
ax.set_xlabel(u'Posição x (m)')  
ax.set_ylabel(u'Posição y (m)')  
ax.set_zlabel(u'Forçamento f')
```

```
#%% Método de Jacobi
```

```
niter = 2500          # nr de iterações
```

```
for i in range(niter):
```

```
    it += 1
```

```
    Vold[:]=V[:]      # actualizar o campo no passo anterior
```

```
    ms=0             # média quadrada
```

```
    for ix in range(1,nx-1):
```

```
        for iy in range(1,ny-1):
```

```
            V[ix,iy] = (V[ix+1,iy] + V[ix-1,iy] + dx2dy2*(V[ix,iy+1] + V[ix,iy-1]) \
                        - dx**2*f[ix,iy]) / a
```

```
# Condições fronteira
```

```
V[0,:]=V[1,:]
```

```
V[nx-1,:]=V[nx-2,:]
```

```
V[:,0]=V[:,1]
```

```
V[:,ny-1]=0
```

```
ms = ms + (V[ix,iy] - Vold[ix,iy])**2          # diferença quadrada entre campos para cada ponto
```

```
ms = np.sqrt(ms/(nx*ny))
```

```
# raiz quadrada da média quadrada
```

```
DifAll = np.append(DifAll,ms)
```

```
# vector com a diferença média quadrada entre campos em d
```

```
dif = V-Vold
```

```
# diferença entre o campo em dois passos consecutivos
```

```
dm = np.max(np.max(dif))
```

```
# diferença máxima entre o campo em dois passos consecuti
```

```
DifMax = np.append(DifMax,dm)
```

```
# vector com a diferença máxima entre campos em dois pass
```

```

# %% Plot 3D

fig = plt.figure()
ax = plt.axes(projection='3d')

surf=ax.plot_surface(xx,yy, V, rstride=5, cstride=5, cmap=cm.jet, linewidth=0)
#ax.set_zlim(0,10)
fig.colorbar(surf, shrink=0.5)

ax.set_xlabel(u'Posição x (m)')
ax.set_ylabel(u'Posição y (m)')
ax.set_zlabel(u'Potencial Eléctrico')
ax.set_title(u'Iteração = '+str(it))

#plt.savefig('figura.png')

# %% Plot 2D
plt.close()
plt.rcParams['figure.figsize'] = 8, 6

# plt.contourf(xx,yy,h)
plt.pcolor(xx,yy,V) # figura 2D do sinal inicial
#plt.clim(0,10) # limites da barra de cores
cb=plt.colorbar()
cb.set_label(u'Potencial Eléctrico') # legenda da barra de cores

plt.xlabel(u'Posição x (m)')
plt.ylabel(u'Posição y (m)')
plt.title(u'Iteração = '+str(it))

```

```
## Plot convergence
```

```
plt.close()  
plt.rcParams['figure.figsize'] = 10, 6
```

```
plt.subplot(2,1,1)  
plt.plot(range(it), DifAll)
```

```
plt.xlabel(u'Iteração')  
plt.ylabel(u'RMS(V-Vold)')  
plt.title(u'Diferença média quadrada entre campos em dois passos consecutivos')  
plt.grid()
```

```
plt.subplot(2,1,2)  
plt.plot(range(it), DifMax)
```

```
plt.xlabel(u'Iteração')  
plt.ylabel(u'Max (V-Vold)')  
plt.title(u'Diferença máxima entre campos em dois passos consecutivos')  
plt.grid()
```

```
plt.tight_layout()
```