UdeMM - Ingeniería de Sistemas

Simulación de modelos en derivadas parciales

Trabajo Práctico 4

Año 2020

Módulo 5

Problemas Elípticos

1. Métodos Iterativos

El operador discreto de Laplace de 5 puntos aplicado a la ecuación de Laplace 2D $\nabla^2 p = 0$ en una grilla cartesiana se escribe:

$$\frac{p_{i+1,j} - 2p_{i,j} + p_{i-1,j}}{\Delta x^2} + \frac{p_{i,j+1} - 2p_{i,j} + p_{i,j-1}}{\Delta y^2} = 0 \tag{1}$$

Cuando $\Delta x = \Delta y$, resulta la expresión siguiente:

$$p_{i+1,j} + p_{i-1,j} + p_{i,j+1} + p_{i,j-1} - 4p_{i,j} = 0 \tag{2}$$

Esta ecuación discreta es válida para cada punto interior del dominio. Si escribimos las ecuaciones para *todos* los puntos interiores, tendremos un sistema lineal de ecuaciones algebraicas, con una matriz de coeficientes *dispersa* (esto es, la mayor parte de sus elementos son cero). Tales sistemas se resuelven empleando **métodos iterativos**.
Asocie cada uno de los siguientes fragmentos de código a su correspondiente método iterativo:

(A)

```
for j in range(1,ny-1):
    for i in range(1,nx-1):
        p[j,i] = .25*(pn[j,i-1]+pn[j,i+1]+pn[j-1,i]+
        pn[j+1,i])
```

(B)

```
for j in range(1,ny-1):
    for i in range(1,nx-1):
        p[j,i] = (1-omega)*p[j,i]+omega*.25*(p[j,i-1]+
        p[j,i+1]+p[j-1,i]+p[j+1,i])
```

(C)

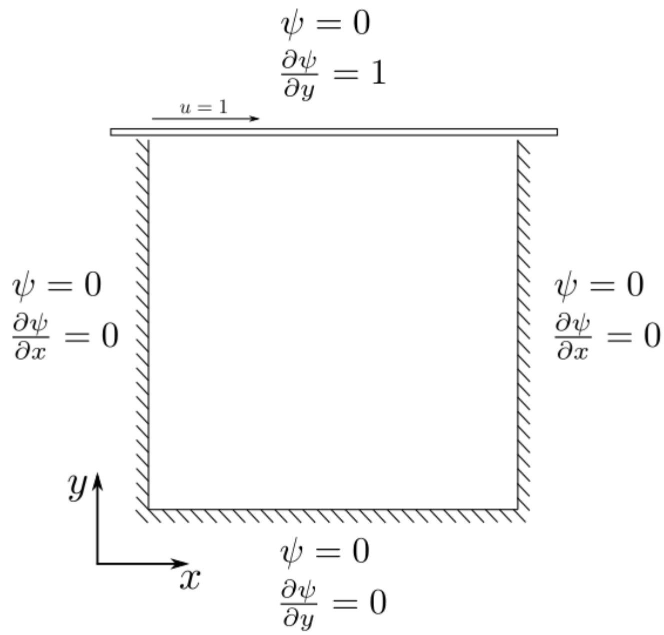
```
for j in range(1,ny-1):
    for i in range(1,nx-1):
        p[j,i] = .25*(p[j,i-1]+p[j,i+1]+p[j-1,i]+
        p[j+1,i])
```

Método Iterativo

1. Jacobi
2. Gauss-Seidel
3. Sobre-relajación sucesiva (SOR)

2. Trabajo de código: Flujo de Stokes

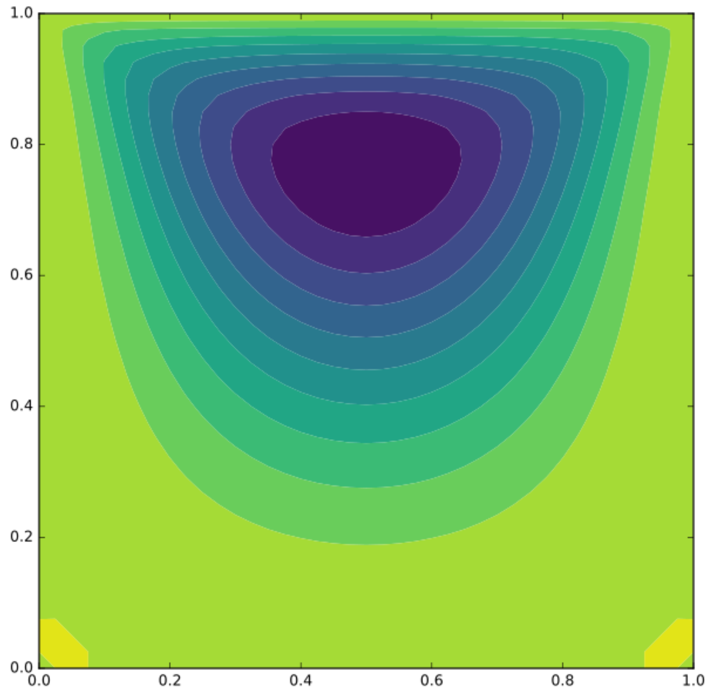
Resolveremos el problema del **flujo en una cavidad con pared móvil**. Asumimos que la pared superior de una cavidad cuadrada se desplaza a una velocidad constante $u = 1$, y no hay salida de fluido a través de la misma. Queremos visualizar el campo de flujo dentro de la cavidad en estado estacionario. Suponemos además condición de no deslizamiento en todas las superficies. Las condiciones de borde se escriben en términos de la función de corriente ψ,



NOTA: La solución del sistema acoplado para ambas variables ω y ψ se llevará a cabo en forma iterativa hasta que la norma L1 de la diferencia entre iteraciones sucesivas sea menor que 1E^{-}6 para ambas cantidades. Utilize los siguientes parámetros en su simulación:

```
import numpy

# Set parameters
nx, ny = 41, 41  # number of points in each direction
L = 1.0  # length of the square cavity
dx = L/(nx-1)  # grid spacing in the x direction
dy = L/(ny-1)  # grid spacing in the y direction

def l1_norm(u, u_ref):
    """
    Computes and returns the L1-norm of the difference
    between a solution u and a reference solution u_ref.

    Parameters
    ----------
    u : numpy.ndarray
        The solution as an array of floats.
    u_ref : numpy.ndarray
        The reference solution as an array of floats.

    Returns
    -------
    diff : float
        The L2-norm of the difference.
    """
    diff = numpy.sum(numpy.abs(u - u_ref))
    return diff
```

Corrobore que el resultado final para la función de corriente en estado estacionario se asemeje al siguiente campo de flujo:


1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0