Trabalho Prático #1:
Solução de Sistemas Lineares
Métodos Iterativos

EXERCÍCIO 1

Resolva a solução do problema do valor contorno elíptico que está sendo dado pela equação diferencial

$$u_{xx} + u_{yy} = -10(x^2 + y^2 + 5)$$

com condições de contorno *u(0,y) = u (1,y) = u (x, 0) = 0* e *u(x,1) = 1*. Aplique o Método iterativo de Jacobi, em uma grade ortogonal com limites (0, 1) × (0, 1) com N × M = 50 × 50 pontos discretos. Como o valor inicial defina u(x, y) = 0 em toda a grade. Após cada iteração, verifique sua média da soma do erro absoluto de uma iteração para a próxima (usando apenas pontos internos)

$$\text{tolerance} = \frac{1}{(N-2)(M-2)} \sum_{i,j} |u_{i,j}^{\text{new}} - u_{i,j}^{\text{old}}|$$

O método iterativo termina quando a tolerância ≤ 10E−7. Escreva um programa que resolva o problema acima. Além disso, faça o que se pede abaixo:

a. Plote a solução u(x, y). Dica: use a ferramenta gnuplot.

EXERCÍCIO 2

Refaça o Exercício 1, mas desta vez use o método SOR, com fatores de aceleração ω = 1.0, ω = 1.95 e ω = 1.99.

a. Compare o número de iterações, o tempo de execução de cada valores de ω.
b. Crie um diagrama log-log para a tolerância em função ao número de iterações (no mesmo diagrama mostram curvas para o Método de Jacobi e para o SOR
c. Comente sobre os resultados acima.

Para Entrega:
Envie um relatorio com os resultados obtidos juntamente com o código-fonte (octave ou python).

TEORIA

Dada a equação de Poisson em duas dimensões:

$$-(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}) = S(x, y)$$

e aproximando as derivadas parciais por diferencas finitas

$$\frac{\partial^2 u}{\partial x^2}|_i = \frac{u_{i-1,j} - 2u_{ij} + u_{i+1,j}}{h^2}$$ e $$\frac{\partial^2 u}{\partial y^2}|_i = \frac{u_{i,j-1} - 2u_{ij} + u_{i,j-1}}{h^2}$$

Considere que o dominio (0, 1) × (0, 1) é discretizado por um grid cartesiano com (N x M) pontos. Supondo a mesma quantidade de pontos em cada direção (N == M), os pontos serão igualmente espaçados por h = (1- 0)/(N-1). Assim, pontos com posião (i,j) neste grid, terão coordenadas dado por (i*h,j*j*h), 1<= i <=N e1 <= j <= N. Aplicado, aproximação de diferenças finitas em todos pontos interno do grid, podemos resolver o problema plicando um dos métodos iteratvos abaixo.

- Pelo método de Jacobi:

$$u_{i,j}^{n+1} = \frac{1}{4}[u_{i-1,j}^{n} + u_{i+1,j}^{n} + u_{i,j-1}^{n} + u_{i,j+1}^{n} + h^2 S_{i,j}]$$

- Método de Gauss-Seidel

$$u_{i,j}^{n+1} = \frac{1}{4}[u_{i-1,j}^{n+1} + u_{i+1,j}^{n} + u_{i,j-1}^{n+1} + u_{i,j+1}^{n} + h^2 S_{i,j}]$$

- Método SOR:

$$u_{i,j}^{n+1} = (1-\omega)u_{i,j}^{n} + \frac{1}{4}\omega[u_{i-1,j}^{n+1} + u_{i+1,j}^{n} + u_{i,j-1}^{n+1} + u_{i,j+1}^{n} + h^2 S_{i,j}]$$

Observação:

- Considere $u_{i,j}= u(x_i,y_j)$
- $S_{i,j} = 10(x_i^2+y_i^2 + 5)$

Algoritmo:

- Dados
    - n, o numeoro de pontos em cada direção
    - O intervalo a e b, tal que a <= x <= b e a <=y <= b
    - tol: a tolerância da aproximação
    - kmax: numero maximo de iterações,

Aloque duas matrizes u (equivale ao u^{n+1}) e u_old(equivale ao u^n) de e dimensão n x n

h ← (b-a)/(n-1)

e ← 1.0

k ← 0

```
u ← 0
//Aplicando a condições de contorn no topo, ou seja u(:,n)
para i =1:n faça
    u(i,n) = 1.0
fim_para
Enquanto (e >= tol e k <= kmax)
   k ← k + 1
   e ← 0
    u_old k ← u
    para j=2:n-1 faça
       y = j*h
       para i=2:n faça
          x = i*h
          u(i,j) = // Insere aqui cálculo correspondente do método
          e ← e + |u(i,j) - u_old(i,j)|
       fim_para
    fim_para
     e ← e/((n-2)*(n-2))
  FimEnquando
```

Para mais detalhes sobre SOR, veja:
http://users.math.uoc.gr/~vagelis/Courses/Scientific_Computing/Papers/Parallel_SOR_Evans_84.pdf