

# Teste Computacional 1 (TC1)

Turma Engenharia Elétrica

Resolvendo Sistemas Lineares pelo Método Iterativo de Gauss Jacobi

## Objetivos:

Entender e implementar o método iterativo de Gauss Jacobi e verificar a velocidade de convergência da sequência gerada usando diversos chutes iniciais, para um conjunto de problemas específicos (serão descritos).

## Implementações

Implementar um código para resolver vários sistemas lineares via método iterativo de Gauss Jacobi. A implementação do método iterativo deve ser feita em rotina (função) própria e deve ser desenvolvida de forma que haja um critério de parada e limitante para a quantidade de iterações para evitar loop infinito nas situações onde não houver convergência. O critério de parada deve ser:

Parar quando a distância relativa (em módulo) entre os 2 últimos vetores < dada tolerância

Usar a norma do máximo para calcular o tamanho (a norma) da distância entre os vetores e, também, usar a norma do máximo para calcular a norma do vetor mais atual. (obs: definição desta norma está nas notas de aulas).

A função deve ter como dados de entrada os elementos mencionados acima, isto é:

a matriz A e o vetor b (os dados do sistema linear  $Ax=b$ )

a tolerância

e a quantidade máxima de iterações

A função deve retornar, pelo menos, a quantidade de iterações realizadas (mas pode haver mais variáveis de retorno se quiserem).

Por exemplo, se a função implementada fosse denominada de “`resolve_via_Jacobi`”, a chamada da função para um sistema linear dado (com a A e b já previamente definidas), com tolerância  $tol=0.0001$  e uma quantidade máxima de iterações de 500 iterações, seria:

`> qte_iter = resolve_via_Jacobi(A, b, 0.0001, 500)`

De forma a ser possível fazer vários experimentos e poder fazer comparações para sistemas lineares quadrados, de várias dimensões, que geram sequências convergentes, funções que geram sistemas lineares (matrizes A e vetores b) específicos, automaticamente, estão sendo disponibilizadas. Serão descritos mais adiante.

## TAREFAS

### Experimento 1

Obter a solução do sistema linear  $Ax=b$ , com A e b dados abaixo, pelo método de Gauss Jacobi

A = [ 5.0 0.5 0.5  
8.0 1.5 5.0  
7.0 6.0 5.0 ]

b = [ 10.0  
15.0  
18.0 ]

Aplicar o método, ou seja, obter a solução usando, os seguintes chutes iniciais:

$x_0 = [0.0 \ 0.0 \ 0.0]$

$x_0 = [1.0 \ 1.0 \ 1.0]$

$x_0 = [20.0 \ 30.0 \ 40.0]$

e mais um chute distinto, à sua escolha.

Para cada chute, nos casos onde houver convergência em menos de 1000 iterações, relatar a solução obtida e o número de iterações necessárias para se obter a solução. Quando não houver convergência, colocar a quantidade máxima de iterações.

|                  | Chute 1 | Chute 2 | Chute 3 | Chute 4 | Média | desvio padrão |
|------------------|---------|---------|---------|---------|-------|---------------|
| Qte de iterações | +1000   | +1000   | +1000   | +1000   | +1000 | N/A           |

TABELA 1 Quantidade de iterações necessárias para atingir a tolerância, para o problema do exemplo 1

Todos os chutes, inclusive o adicional, definido como [2.0, 3.0, 4.0] excederam o limite de 1000 iterações

## Experimento 2

De forma a ser possível fazer vários experimentos e poder fazer comparações para sistemas lineares de várias dimensões, que geram sequências convergentes, foi criada uma função chamada “gera\_matriz\_DIAGDOM(n)” que gera uma matriz de dimensão n (valor a ser fornecido na chamada da função) cujos elementos são quaisquer, gerados aleatoriamente entre 0 e 1 e diagonalmente dominante (ver definição nas notas de aulas). Além disso, com o objetivo de criar sistemas cuja solução seja previamente conhecida (para fins de comparações didáticas) foi criada a função “gera\_b\_para\_Sistema\_linear\_com\_solucão\_unitaria” de forma a gerar um vetor b fazendo o produto da matriz A com um vetor unitário (com todos os seus elementos iguais a 1).

Usando os códigos fornecidos, obtenha a solução de um sistema linear, de dimensão 3, gerado pelas funções descritas acima, com 4 chutes (vetores iniciais) distintos. Estes chutes devem ser vetores gerados no código, tendo com elementos valores aleatórios entre 0 e 10.

Para cada chute, anotar e relatar, a solução obtida e o número de iterações necessárias para se obter a solução com tolerância dada. Calcular a média aritmética da quantidade de iterações observadas. Relatar os dados do sistema linear, ou seja, copiar a matriz A gerada e o vetor b gerado.

Usando os códigos fornecidos, repetir o mesmo processo mais 2 vezes, isto é, gerar mais 2 sistemas lineares e obter, para cada sistema, a solução com 4 chutes iniciais distintos.

Ao final, portanto, o experimento terá sido realizado para 3 sistemas distintos.

Relatar os resultados em uma tabela similar à tabela dada abaixo.

| dim=3     | Chute 1 | Chute 2 | Chute 3 | Chute 4 | Média  | desvio padrão |
|-----------|---------|---------|---------|---------|--------|---------------|
| Sistema 1 | 34      | 34      | 34      | 33      | 33.75  | 0.433         |
| Sistema 2 | 40      | 40      | 40      | 42      | 40.5   | 0.87          |
| Sistema 3 | 108     | 107     | 108     | 106     | 107.25 | 0.83          |

TABELA 2 Quantidade de iterações necessárias para atingir a tolerância, para problemas de dimensão 3

### Experimento 3

Repetir o mesmo experimento feito em (2) mas agora com problemas de dimensão 20.

Relatar os resultados da mesma forma, em uma tabela similar à tabela dada abaixo.

| dim=20    | Chute 1 | Chute 2 | Chute 3 | Chute | Média  | desvio padrão |
|-----------|---------|---------|---------|-------|--------|---------------|
| Sistema 1 | 299     | 304     | 303     | 303   | 302.25 | 1.92          |
| Sistema 2 | 323     | 321     | 326     | 325   | 323.75 | 1.92          |
| Sistema 3 | 354     | 360     | 357     | 359   | 357.5  | 2.29          |

TABELA 3 Quantidade de iterações necessárias para atingir a tolerância, para problemas de dimensão 20

Para todos os experimentos usar a tolerância =  $10^{-12}$  e fazer no máximo 1000 iterações (Qtde Máxima de iterações).

### Informações adicionais sobre trabalho e a entrega

**Componentes:** O trabalho deve ser feito por, no máximo, duas pessoas.

**Linguagem:** Em python

(quem preferir fazer em outra linguagem, entrar em contato comigo para combinar)

**Data de entrega:** 06/05/2024 às 23h59

### O que entregar:

- o código implementado. O código deve ter o primeiro nome do(s)a(s) componentes, escritos em minúsculas, separados por underscore)

Exemplos:

Feito por Lucas e João\_Pedro -> lucas\_joao\_pedro.py

Feito por Ana e Gabriel -> ana\_gabriel.py

Colocar o(s) nome(s) completos do(os)a(as) componente(s) nas primeiras linhas do código.

- o relatório contendo o que foi solicitado nas tarefas. Gere um arquivo em pdf.  
Colocar o(s) nome(s) completo(s) também no relatório.