

Obs1.: antes de resolver os exercícios abaixo, teste cada um dos métodos implementados resolvendo sistemas lineares simples de dimensão 3×3 (use um exemplo dos Slides).

Obs2.: utilize precisão dupla.

1. Resolva o sistema gerado pela questão 2 da 1ª Lista utilizando métodos diretos (Thomas, Gauss, LU e Cholesky) e métodos iterativos (Jacobi e Gauss-Seidel). Para este estudo, considere:
 - Sistemas lineares de diferentes dimensões (ex. 1000, 5000, 10000) a partir do problema resolvido na 1ª Lista variando o número de elementos;
 - A aplicação do critério das linhas para determinar se os métodos iterativos convergem neste caso;
 - O tempo de processamento de cada método, direto ou iterativo, para a resolução do sistema e monte uma tabela;
 - Apresente o critério de parada e a tolerância utilizada para os métodos iterativos;
 - Comente os resultados obtidos.

O critério de Parada Utilizado nas implementações dos métodos iterativos está no código abaixo. Na geração da matriz, foi usado o erro 0,01.
E a tolerância dos métodos iterativos usada foi de 0,0001.

Python

```
1  def normaMaximo(self, x):
2      size = len(x)
3      maximo = abs(x[0])
4
5      for i in range(size):
6          temp = abs(x[i])
7          if(temp > maximo):
8              maximo = temp
9
10     return maximo
11
12  def distanciaMaximo(self, x1, x2):
13      if(len(x1) != len(x2)):
14          print("O tamanho dos vetores x1 e x2 precisa ser o mesmo")
15          return 0
16
17      size = len(x1)
18      dist = abs(x1[0] - x2[0])
19
20      for i in range(size):
21          temp = abs(x1[i] - x2[i])
22          if(temp > dist):
23              dist = temp
24
25      return dist
26
27  def calculaErro(self, x_prox, x_atual):
28      return self.distanciaMaximo(x_prox, x_atual) / self.normaMaximo(x_prox)
```

Antes da execução dos métodos iterativos foi feita a verificação do critério das linhas com a implementação abaixo

Python

```
1 def checarCritérioDasLinhas(self, M):
3     ordem = len(M)
5     for i in range(ordem):
6         valores = []
7         div = M[i][i]
9         #se algum elemento da diagonal principal for zero
10        #a matriz não satisfaz o critério das linhas
11        if(div == 0):
12            return False
13
14        for j in range(ordem):
15            if(i != j):
16                valores.append(M[i][j] / div)
17
18        #um elemento dividido pelo valor da diagonal principal deu maior ou
19        #igual que 1
20        #a matriz não satisfaz o critério das linhas
21        if(max(valores) >= 1):
22            return False
23
24    return True
```

O método que calcula o erro residual da questão 2 também está mostrado abaixo.

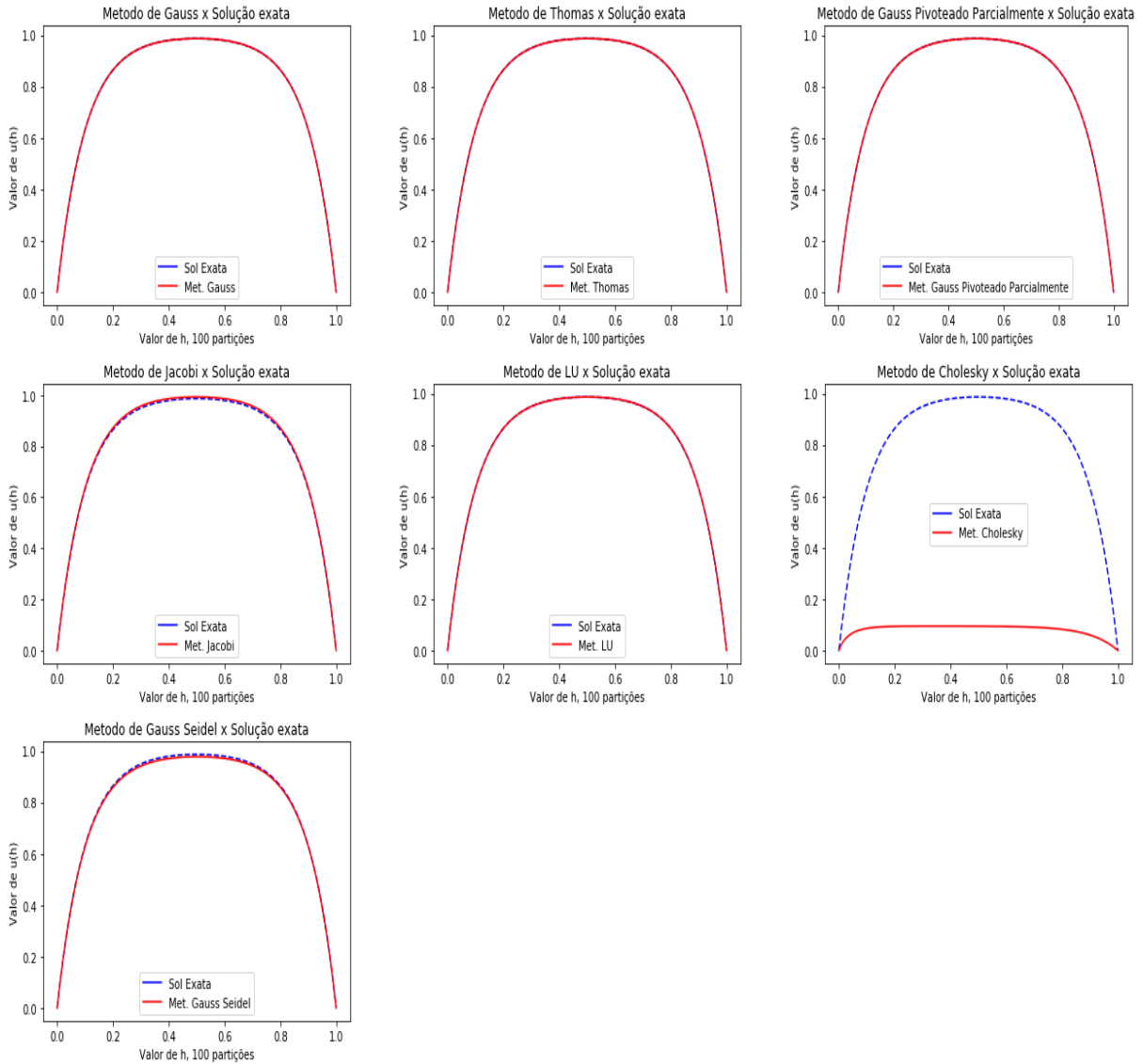
Python

```
1 def erroResidual(self, M, X, B):
2     size = len(M[0])
3     erroRet = []
4     for i in range(size):
5         valor = 0
6         for j in range(size):
7             valor += M[i][j] * X[j]
9
10        erroRet.append(abs(valor - B[i]))
11
12    return [erroRet, max(erroRet)]
```

Lembrando que a contagem de passos conta os realizados pelo pivoteamento
 Para uma Matriz 100x100:

<i>Metodo</i>	<i>Passos</i>	<i>Tempo de Execução</i>
Thomas (direto)	9997	0:00:00.002952
Gauss (direto)	328251	0:00:00.107486
Gauss com Pivoteamento Parcial	338052	0:00:00.104647
Gauss-Seidel (iterativo)	4361445	0:00:01.201678
LU	333300	0:00:00.324635
Choleski	24354	0:00:00.074590
Jacobi (iterativo)	46629	0:00:00.785732

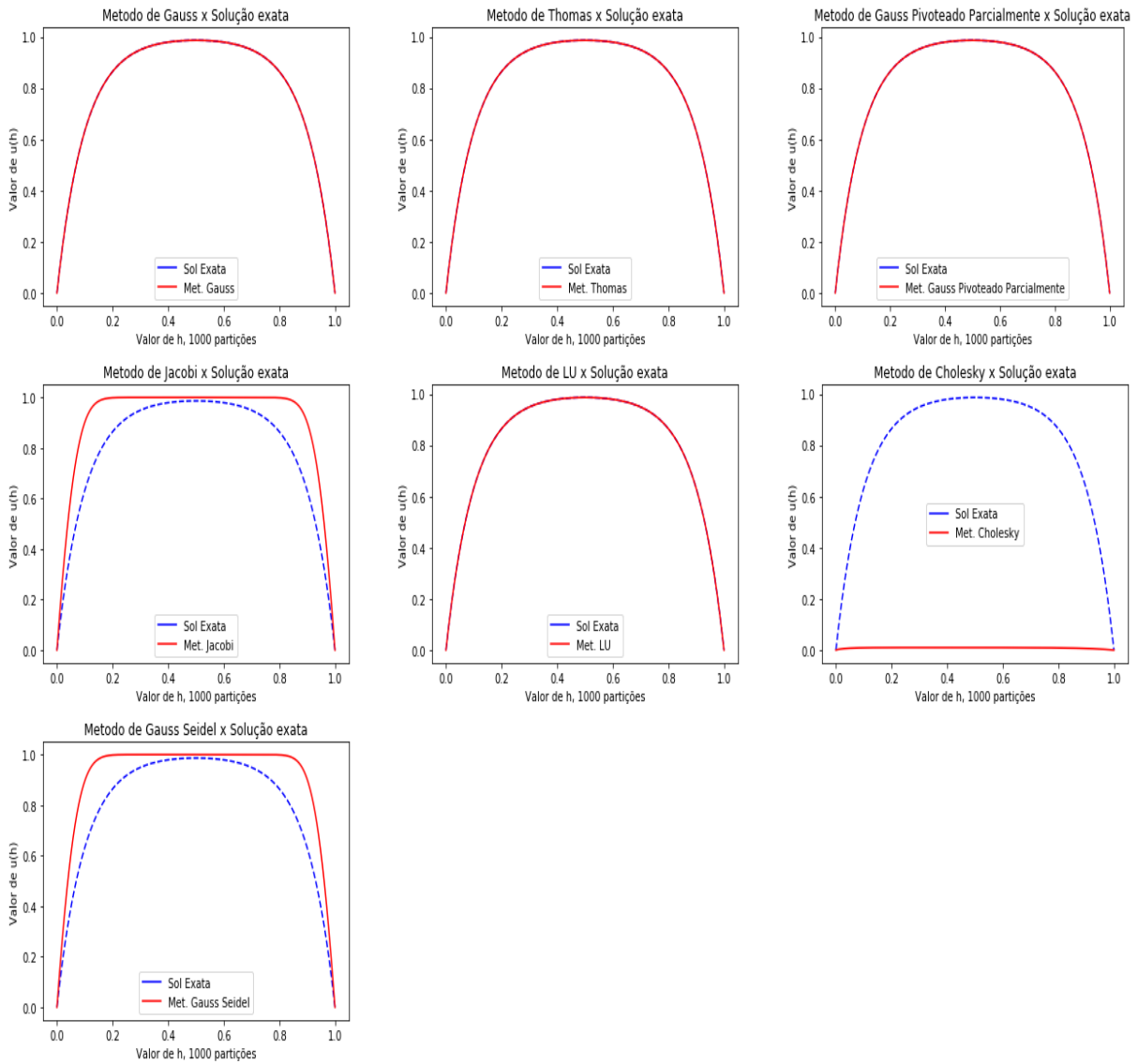
Tabela 1: Matriz 100x100



Para uma matriz 1000x1000:

<i>Metodo</i>	<i>Passos</i>	<i>Tempo de Execução</i>
Thomas (direto)	999997	0:00:00.235122
Gauss (direto)	332832501	0:01:39.306903
Gauss com Pivoteamento Parcial	333830502	0:01:39.587652
Gauss-Seidel (iterativo)	2011970016	0:09:22.138811
LU	333333000	0:05:06.996908
Choleski	2493504	0:00:34.001452
Jacobi (iterativo)	3964032	0:06:03.981231

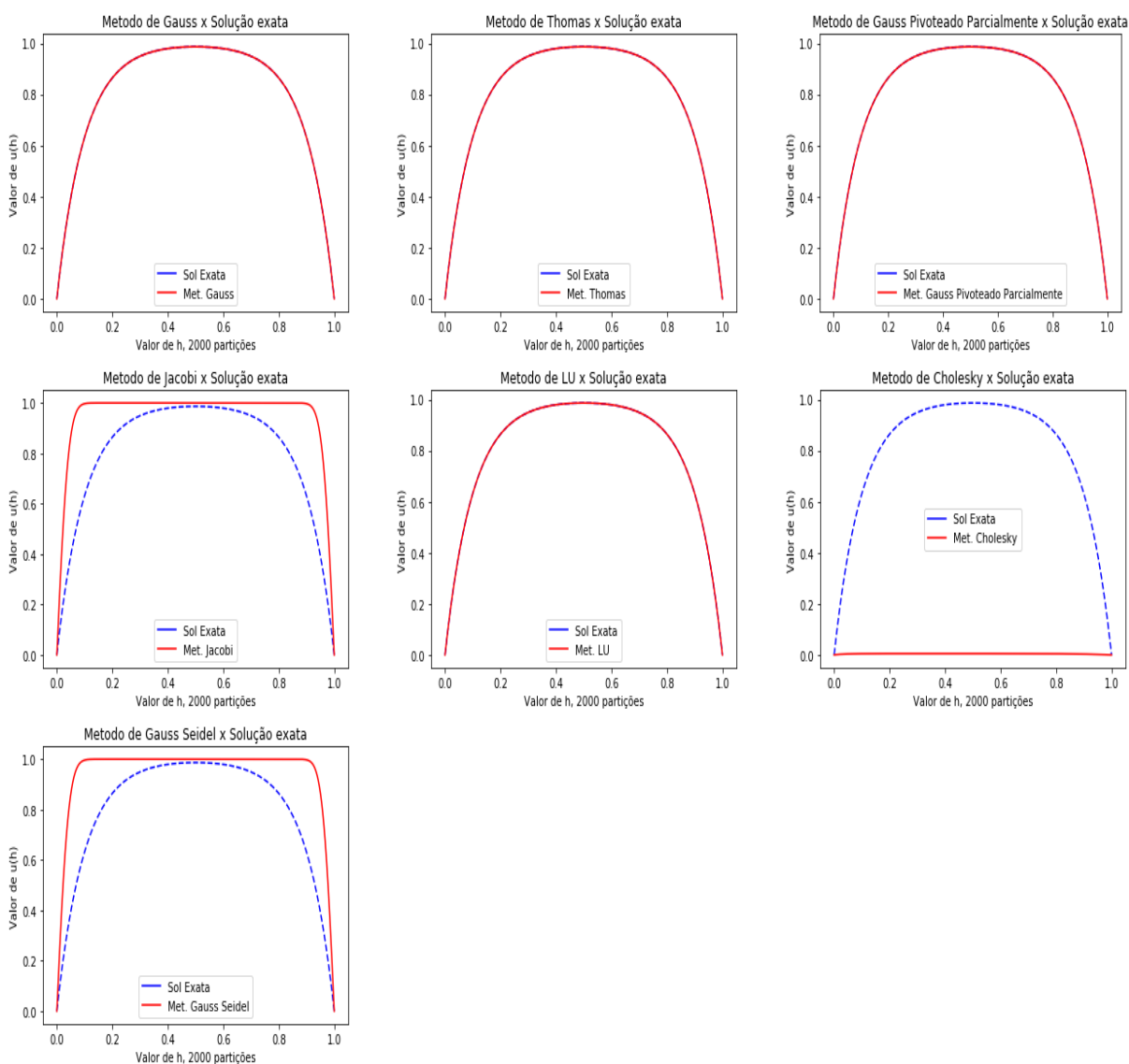
Tabela 2: Matriz 1000x1000



Para uma matriz 2000x2000:

Metodo	Passos	Tempo de Execução
Thomas (direto)	3999997	0:00:01.438566
Gauss (direto)	2664665001	0:12:35.660406
Gauss com Pivoteamento Parcial (direto)	2668661002	0:13:36.051002
Gauss-Seidel (iterativo)	9270722320	0:45:47.694755
LU	2666666000	0:44:35.352543
Choleski	9987004	0:04:44.301479
Jacobi (iterativo)	9137429	0:26:51.788595

Tabela 3: Matriz 2000x2000



2. Seja o sistema linear:

$$\mathbf{Ax} = \mathbf{b}$$

onde,

$$\mathbf{A} = A_{i,j} = \frac{1}{i+j+1} \quad \text{e} \quad \mathbf{b} = b_i = \frac{1}{i+n+1}.$$

Supondo a matriz $\mathbf{A}_{n \times n}$, com diferentes dimensões n (ex. $n = 10, 100, 1000$), faça:

- Resolva utilizando o método de eliminação de Gauss **sem** e **com** pivoteamento e a decomposição LU.
- Determine o erro cometido, por cada um dos métodos utilizados, através do resíduo calculado na norma do máximo, dado por:

$$\|\mathbf{Ax} - \mathbf{b}\|_{\infty} = \max_{1 \leq i \leq n} |A_{i,j}x_i - b_i|, \quad \forall j \in [1, n]$$

onde x_i é o vetor solução. Compare e discuta os resultados.

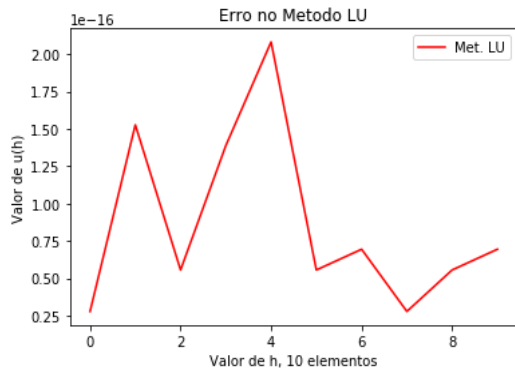
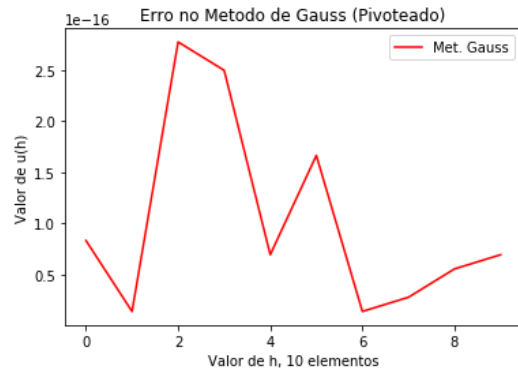
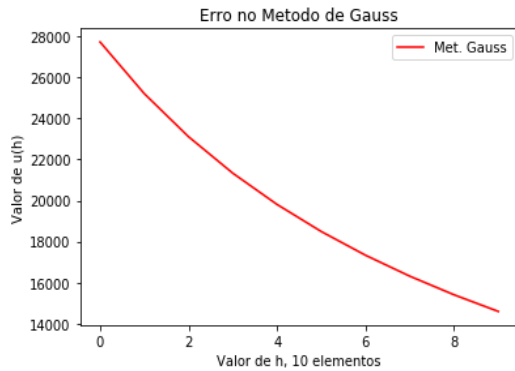
Supondo matrizes de dimensões 10,100 e 1000; Para 10 elementos:

	<i>Gauss</i>	<i>Gauss com Pivoteamento Parcial</i>	<i>LU</i>
PASSOS	375	475	440
TEMPO DE EXECUÇÃO	0:00:00.000989	0:00:00.000997	0:00:00.000990

Tabela 4: Comparação de resultados de diferentes dimensões

	<i>Gauss</i>	<i>Gauss com Pivoteamento Parcial</i>	<i>LU</i>
Erro Max	27717.293488028645	2.7755575615628914e-16	2.0816681711721685e-16

Tabela 5: Comparação de erros



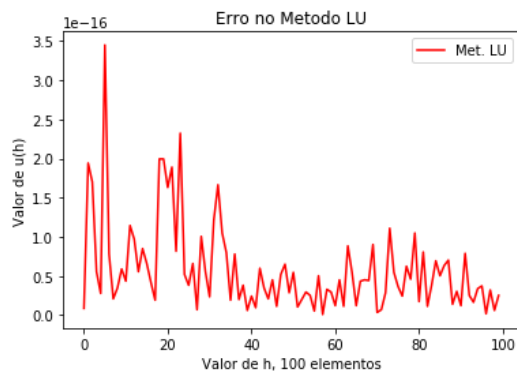
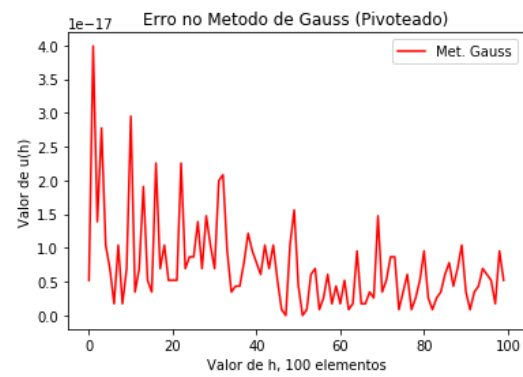
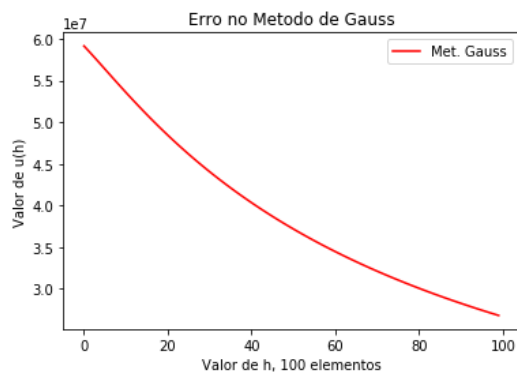
Para 100:

	<i>Gauss</i>	<i>Gauss com Pivoteamento Parcial</i>	<i>LU</i>
PASSOS	338250	348250	343400
TEMPO DE EXECUÇÃO	0:00:00.156041	0:00:00.100983	0:00:00.085986

Tabela 6: Comparação de resultados de diferentes dimensões

	<i>Gauss</i>	<i>Gauss com Pivoteamento Parcial</i>	<i>LU</i>
Erro Max	59153715.99267997	3.9898639947466563e-17	3.452099717193846e-16

Tabela 7: Comparação de erros



Para 1000 elementos:

	<i>Gauss</i>	<i>Gauss com Pivoteamento Parcial</i>	<i>LU</i>
PASSOS	333832500	334832500	334334000
TEMPO DE EXECUÇÃO	0:01:38.251980	0:01:40.540858	0:05:04.649882

Tabela 8: Comparação de resultados de diferentes dimensões

	<i>Gauss</i>	<i>Gauss com Pivoteamento Parcial</i>	<i>LU</i>
Erro Max	140669356.34659576	1.9081958235744878e-17	3.3881317890172013563e-20

Tabela 9: Comparação de erros

