Relatório do Trabalho Computacional 2

Erik Davino Vincent - BMAC - Turma 54
NUSP: 10736584

June 2, 2019

IME - USP

Contents

1 Introdução

O seguinte relatório tem como objetivo a analise dos resultados dos algorítimos criados no *Python 3.7*, usados para o **Trabalho Computacional 2**, de acordo com o que foi solicitado.

De antemão, aviso que para a melhor verificação dos resultados obtidos, deve-se observar os arquivos "resultados1" e "resultados2", com os resultados dos algorítimos criados.

2 Método SOR

Foram utilizadas cinco matrizes esparsas, simétricas positivas-definidas para o estudo. Todas possuem determinante diferente de 0, logo possuem solução. Consideramos o vetor inicial como $\vec{0}$, um máximo de 2000 iterações e uma tolerância máxima de $1 \cdot 10^{-6}$. Para a verificação de convergência, foi observada a $\|r\|_\infty$, isso é, a norma infinito do resíduo, para cada iteração.

2.1 Resultados obtidos

- Matriz 1:

  Tamanho: 27x27

  $\omega = 1$ [Gauss-Seidel]

  $\|erro\|_2 = 5.6705572624014735$.
  Numero de iterações: 2001 (máximo atingido).
  Tempo de computação: 1.318568229675293 segundos.

Obs.: erro = diferença entre resultado real (vetor unitário) e resultado obtido.

  $\omega = 1.25$

  $\|erro\|_2 = 5.1944567842822 62$.
  Numero de iterações: 2001 (máximo atingido).
  Tempo de computação: 1.5247633457183838 segundos.

  $\omega = 1.5$

  $\|erro\|_2 = 5.708433662722788$.
  Numero de iterações: 2001 (máximo atingido).
  Tempo de computação: 1.27396559715271 segundos.

  $\omega = 1.75$

  $\|erro\|_2 = 5.701455068207202$.
  Numero de iterações: 2001 (máximo atingido).
  Tempo de computação: 1.3555736541748047 segundos.

$\omega = 2$

$\|erro\|_2 = 5.687485015225028$.
Numero de iterações: 2001 (máximo atingido).
Tempo de computação: 1.6231069564819336 segundos.

Sobre os resultados para a matriz 1: Devido aos grandes valores nessa matriz, muito erro é gerado nas operações, o que faz com que o trabalho para encontrar o $\vec{x}$ seja muito grande também. Como a matriz é bem pequena, em relação as outras, o tempo de computação não é elevado. Provavelmente se manteve estável, pois o número máximo de iterações foi sempre atingido. Usando o tempo de computação como critério, eu diria que o ω ideal é ~ 1.5.

- Matriz 2:

  Tamanho: 199x199

  $\omega = 1$ [Gauss-Seidel]

  $\|erro\|_2 = 6.932321114199427 \cdot 10^{-10}$.
  Numero de iterações: 4.
  Tempo de computação: 1.7564196586608887 segundos.

  $\omega = 1.25$

  $\|erro\|_2 = 2.6329301929475523 \cdot 10^{-05}$.
  Numero de iterações: 15.
  Tempo de computação: 2.081052303314209 segundos.

  $\omega = 1.5$

  $\|erro\|_2 = 0.005131208622683244$.
  Numero de iterações: 31.
  Tempo de computação: 2.4052183628082275 segundos.

  $\omega = 1.75$

  $\|erro\|_2 = 0.07163245509322742$.
  Numero de iterações: 76.
  Tempo de computação: 3.7591421604156494 segundos.

  $\omega = 2$

  $\|erro\|_2 = 13.707743047733151$.
  Numero de iterações: 2001 (máximo atingido).
  Tempo de computação: 50.86198973655701 segundos.

Sobre os resultados para a matriz 2: podemos concluir que o ω ideal para essa matriz é 1. Observe que as matrizes 3 e 4 são semelhantes a essa, então veremos resultados parecidos.

- Matriz 3:

  Tamanho: 300x300

  $\omega = 1$ [Gauss-Seidel]

  $\|erro\|_2 = 1.2980869169884327 \cdot 10^{-10}$.
  Numero de iterações: 4.
  Tempo de computação: 1.570756435394287 segundos.

  $\omega = 1.25$

  $\|erro\|_2 = 1.139336312316622 \cdot 10^{-05}$.
  Numero de iterações: 15.
  Tempo de computação: 2.1385648250579834 segundos.

  $\omega = 1.5$

  $\|erro\|_2 = 0.0033754056235056756$.
  Numero de iterações: 31.
  Tempo de computação: 3.0635197162628174 segundos.

  $\omega = 1.75$

  $\|erro\|_2 = 0.05809824113951881$.
  Numero de iterações: 76.
  Tempo de computação: 5.676847457885742 segundos.

  $\omega = 2$

  $\|erro\|_2 = 17.017309405586831$.
  Numero de iterações: 2001 (máximo atingido).
  Tempo de computação: 117.70887660980225 segundos.

Sobre os resultados da matriz 3: podemos concluir que o ω ideal é 1. Essa matriz vem do mesmo grupo de matrizes da matriz 2 (uma "versão maior" da matriz anterior?), logo observamos resultados muito semelhantes, apenas com maior tempo de computação, devido a escala do problema.

- Matriz 4:

  Tamanho: 500x500

  $\omega = 1$ [Gauss-Seidel]

  $\|erro\|_2 = 3.970685725352563 \cdot 10^{-10}$.
  Numero de iterações: 3.
  Tempo de computação: 2.9093031883239746 segundos.

  $\omega = 1.25$

  $\|erro\|_2 = 6.30133773729189 \cdot 10^{-05}$.
  Numero de iterações: 16.
  Tempo de computação: 5.0129125118255615 segundos.

$\omega = 1.5$

$\|erro\|_2 = 0.007938096583749001$.
Numero de iterações: 32.
Tempo de computação: 7.5498363971710205 segundos.

$\omega = 1.75$

$\|erro\|_2 = 0.08909599645185523$.
Numero de iterações: 77.
Tempo de computação: 15.183024644851685 segundos.

$\omega = 2$

$\|erro\|_2 = 21.58550286841684$.
Numero de iterações: 2001 (máximo atingido).
Tempo de computação: 334.4933285713196 segundos.

Sobre os resultados da matriz 4: podemos concluir que o ω ideal é 1. Essa matriz vem do mesmo grupo de matrizes da matriz 2, logo observamos resultados muito semelhantes, apenas com maior tempo de computação, devido a escala do problema.

- Matriz 5:

  Tamanho: 900x900

  $\omega = 1$ [Gauss-Seidel]

  $\|erro\|_2 = 0.0002488221925603945$.
  Numero de iterações: 747.
  Tempo de computação: 419.3978316783905 segundos.

  $\omega = 1.25$

  $\|erro\|_2 = 0.015775950315693545$.
  Numero de iterações: 447.
  Tempo de computação: 250.73916149139404 segundos.

  $\omega = 1.5$

  $\|erro\|_2 = 0.12560256204567447$.
  Numero de iterações: 245.
  Tempo de computação: 134.9454402923584 segundos.

  $\omega = 1.75$

  $\|erro\|_2 = 0.3544045452522077$.
  Numero de iterações: 95.
  Tempo de computação: 56.77784085273743 segundos.

$\omega = 2$

$\|erro\|_2 = 9.003498043683594$.
Numero de iterações: 2001 (máximo atingido).
Tempo de computação: 1058.8030331134796 segundos.

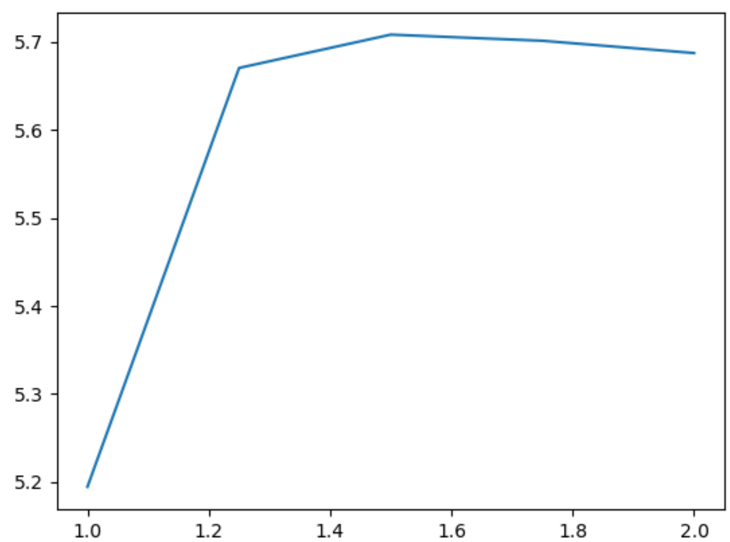
Sobre os resultados da matriz 5: podemos concluir que o ω ideal é ~ 1.75. Essa matriz vem de outro grupo de problemas e matrizes, logo os resultados divergem dos três anteriores. Provavelmente, devido ao uso da $\|\cdot\|_\infty$ como verificação de convergência, o erro que obtemos é maior para ω maior, pois o programa considera a convergência, mesmo que somente um dos valores tenha chegado até a tolerância exigida.

Observação geral sobre ω: de todos esses cinco resultados, em nenhum deles $\omega = 2$ foi uma escolha minimamente razoável. Conclui-se que não se deve utilizar $\omega = 2$. Isso está de acordo com o teorema de Kahan, de que o método SOR converge somente se $0 < \omega < 2$.

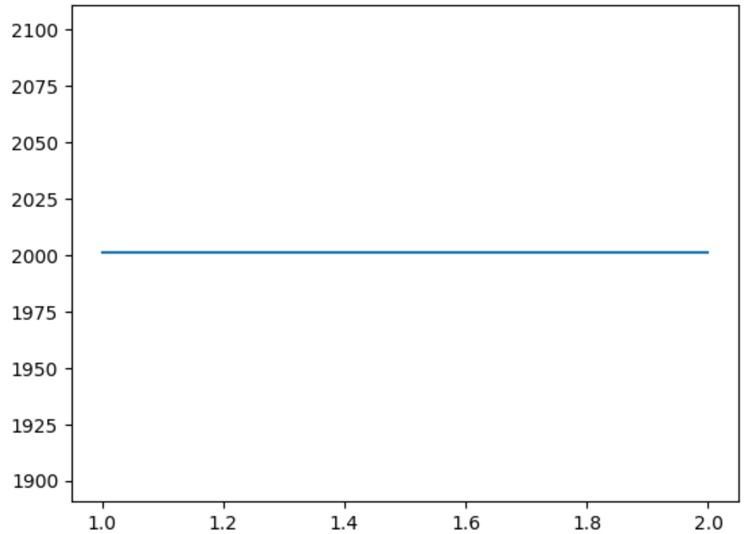
2.2 Gráficos dos resultados

2.2.1 Gráficos Matriz 1

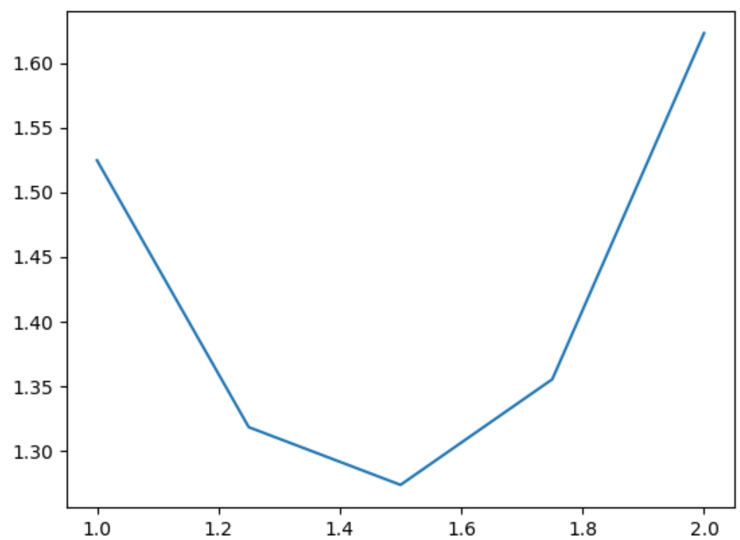
Erro x ω:



Iterações x ω:

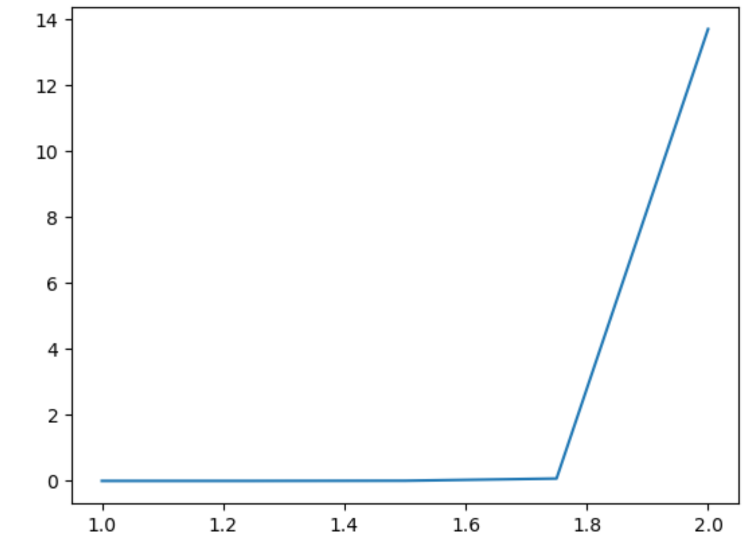


Tempo x ω:

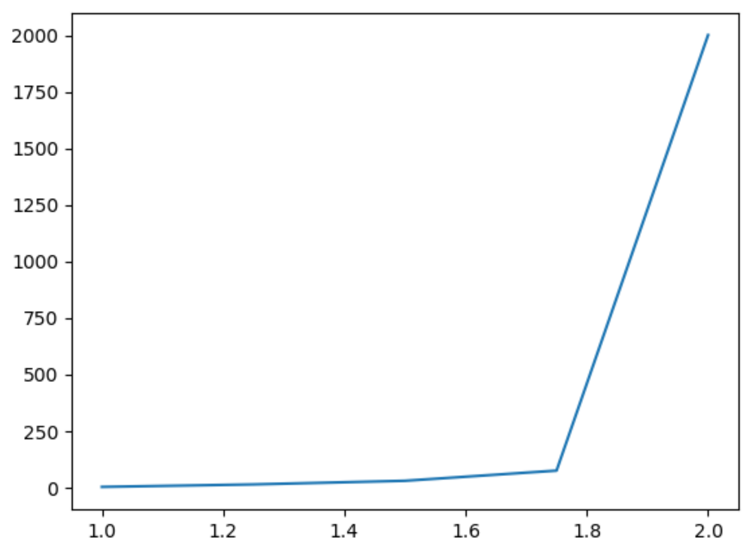


2.2.2 Gráficos Matriz 2

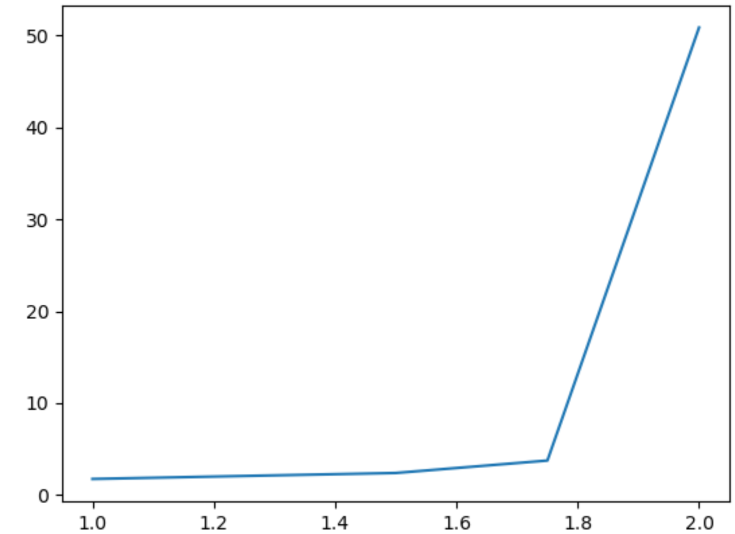
Erro x ω:



Iterações x ω:

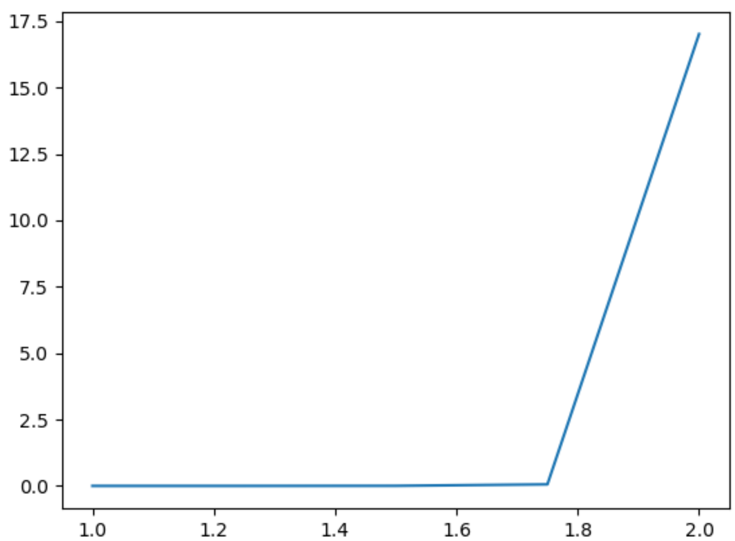


Tempo x ω:

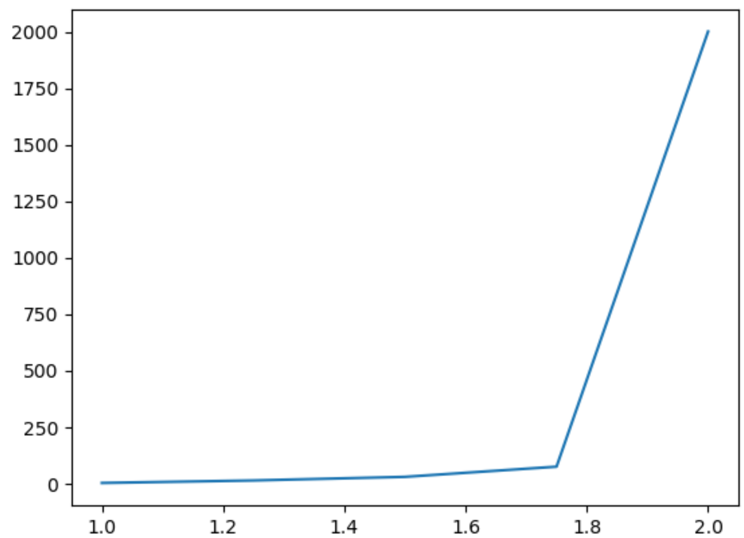


2.2.3 Gráficos Matriz 3

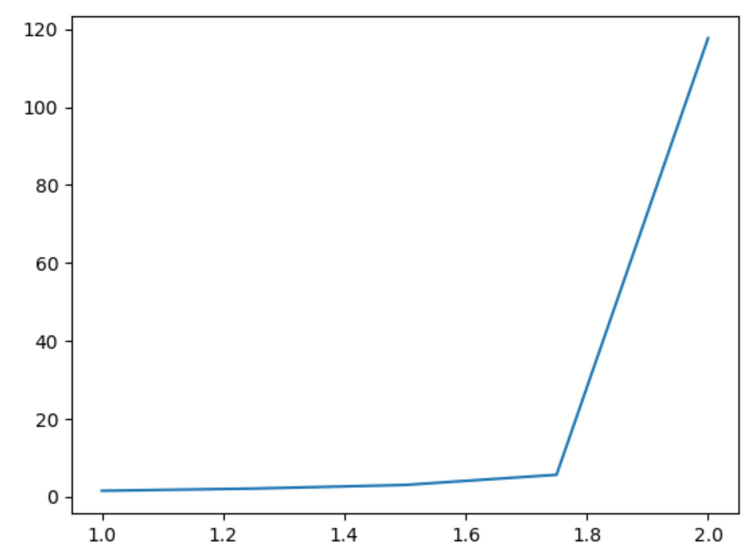
Erro x ω:



Iterações x ω:

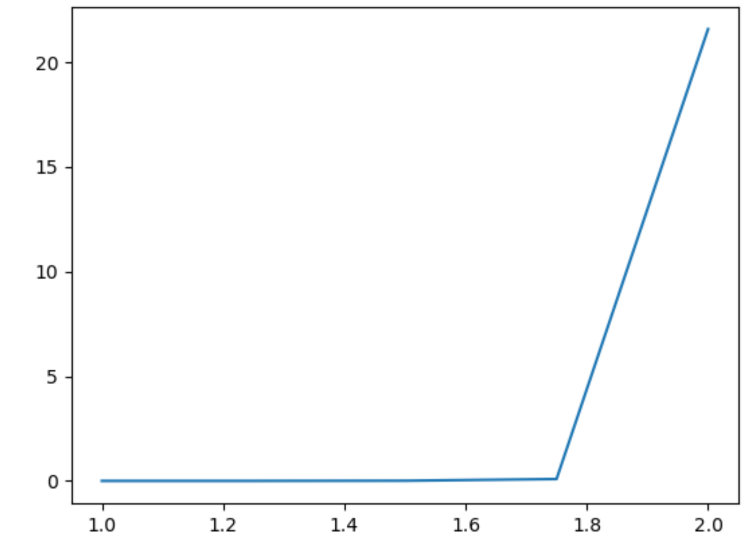


Tempo x ω:

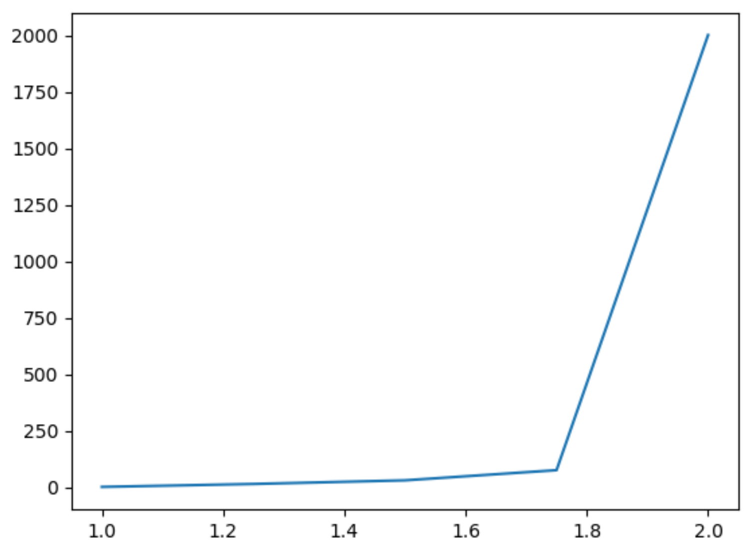


2.2.4 Gráficos Matriz 4

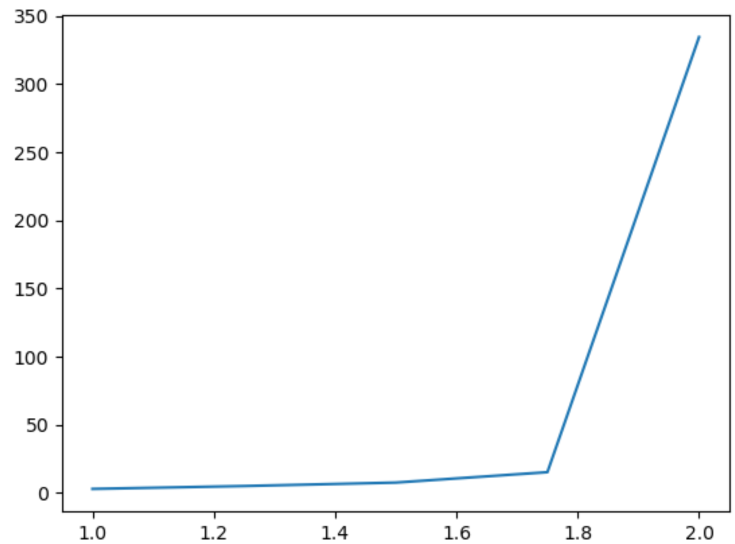
Erro x ω:



Iterações x ω:

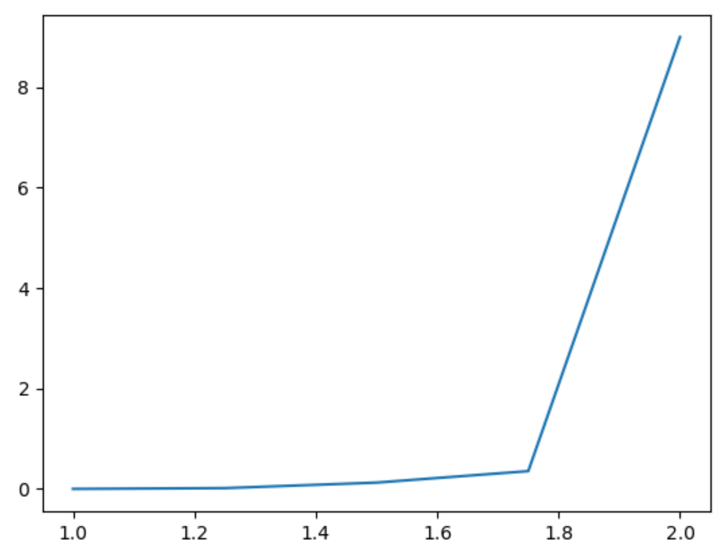


Tempo x ω:

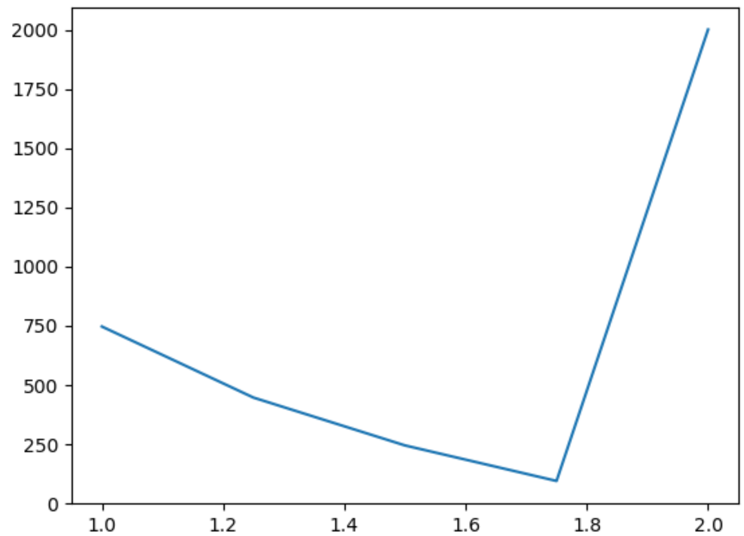


2.2.5 Gráficos Matriz 5

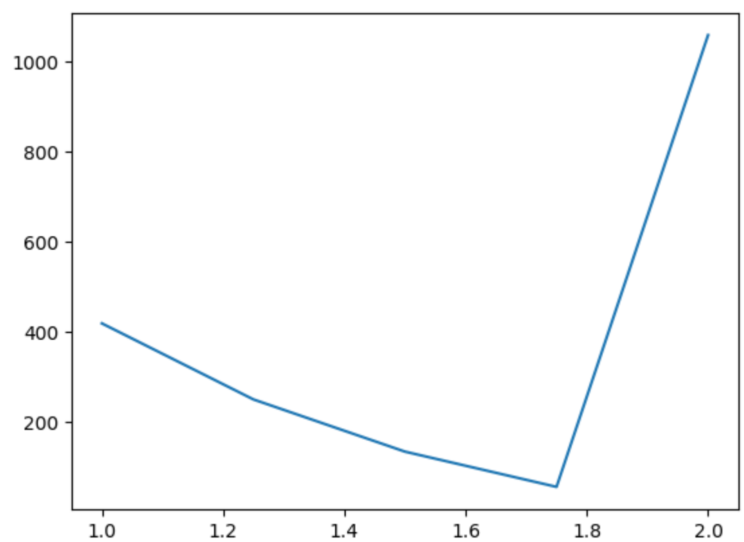
Erro x ω:



Iterações x ω:



Tempo x ω:



2.2.6 Resultados extras

Utilizando a $\|\cdot\|_2$ do resíduo como critério de convergência do método para a matriz 5, obtive os resultados em "resultados_extra". O que observei é que a única diferença obtida foi o número de iterações, o qual aumentou. Além disso, a precisão melhorou.

Fiz também algumas observações para outras matrizes, menores dessa vez. Um problema comum encontrado foi atingir o limite do tamanho do float, e obter erros de *overflow*. Isso pois, apesar de pequenas, operavam com valores muito grandes. Para as matrizes que não apresentaram esses erros, os resultados foram muito semelhantes aos vistos acima.

3 Método do Gradiente Conjugado

Foram utilizadas as matrizes 1, 2, 3, 4 e 5 para se extrair resultados do método do Gradiente Conjugado. Isso, pois, mesmo que as matrizes 1 a 4 tenham sido as que mais demoraram para convergir, vi que o método trouxe uma mudança de resultado para todas as cinco matrizes. Consideramos o vetor inicial como $\vec{0}$, um máximo de 2000 iterações e uma tolerância máxima de $1 \cdot 10^{-6}$. Para a verificação de convergência, foi observada a $\|r\|_2$, isso é, a norma infinito do resíduo, para cada iteração.

3.1 Resultados obtidos

- Matriz 1:

  Tamanho: 27x27

  $\|erro\|_2 = 9.874220931163642 \cdot 10^{-08}$.
  Numero de iterações: 71.
  Tempo de computação: 0.203078031539917 segundos.

- Matriz 2:

  Tamanho: 199x199

  $\|erro\|_2 = 5.151619552914506 \cdot 10^{-10}$.
  Numero de iterações: 14.
  Tempo de computação: 1.8311574459075928 segundos.

- Matriz 3:

  Tamanho: 300x300

  $\|erro\|_2 = 7.527812024440836 \cdot 10^{-10}$.
  Numero de iterações: 14.
  Tempo de computação: 2.7717161178588867 segundos.

- Matriz 4:

  Tamanho: 500x500

  $\|erro\|_2 = 1.6030647960685775 \cdot 10^{-10}$.
  Numero de iterações: 15.
  Tempo de computação: 5.231234788894653 segundos.

- Matriz 5:

  Tamanho: 900x900

  $\|erro\|_2 = 1.3229364899598372 \cdot 10^{-10}$.
  Numero de iterações: 39.
  Tempo de computação: 17.96347212791443 segundos.

Sobre os resultados obtidos: vemos para as matrizes 2 a 4 que os resultados são semelhantes. Isso decorre do mesmo fato considerado para o método SOR, que essas matrizes são parte de uma mesma família de problemas. Para essas matrizes houve um declínio de resultado, se comparado ao método anterior com $\omega = 1$, porém ainda é um resultado muito bom e pouco custoso em termos de tempo. Além disso, exceto para a matriz 3, houve uma melhoria de precisão, o que deve decorrer do uso da $\| \cdot \|_2$ como critério de convergência. Tendo em vista que o método é mais eficiente para "sistemas grandes e esparsos com valores não zeros aparecendo em padrões previsíveis", o resultado parece estar de acordo.

As matrizes 1 e 5 sofreram melhoras significativas de eficiência e de precisão. Especialmente a matriz 1, que sempre utilizava o limite de iterações para fazer o cálculo.

Considerando que o número de iterações do método caso a matriz esteja bem pré-condicionada (no caso, não está) é da ordem de $\sqrt{n}$, nxn o tamanho da matriz, vemos que os resultados obtidos estão bem condizentes com os teóricos:

Matriz 1: $\sqrt{27} = 5.19$
Matriz 2: $\sqrt{199} = 14.10$
Matriz 3: $\sqrt{300} = 17.32$
Matriz 4: $\sqrt{500} = 22.36$
Matriz 5: $\sqrt{900} = 30$

`FIM DO RELATÓRIO`