# Trabalho 2

Gustavo Giro Resende - 11340571

# Introdução

O estudo tem como objetivo analisar e comparar a aplicação de métodos de relaxação (SOR) e Gradiente Conjugado sem pré-condicionamento, para solução de sistemas lineares.

Para todos os exemplos expostos aqui foram consideradas as seguintes características:

- As matrizes imputadas são simétricas positivas definidas
- As matrizes estão na configuração COO
- A solução de cada sistema linear Ax = b é dada por x_sol = [1,1,…,1]
- O número máximo de interações considerado para o método de relaxação SOR foi 300
- A tolerância foi variada entre 10^(-1) e 10^(-10) para as comparações
- O erro foi calculado como sum( (x_aprox – x_sol) ^ (2) )

Conforme apresentado na teoria, *Kahan (Teorema 7.24 - Numerical Analysis*, Burden e Faires*)* afirma que "*se* $a_{ii}$ *!= 0, para todo i = 1,2,…,n então o raio espectral de* $T_w >= | w – 1 |$*. Isso implica que o método pode convergir somente se 0 < w < 2*". Para o caso de a matriz ser positiva definida, *Ostrowski-Reich (Teorema 7.5 - Numerical Analysis*, Burden e Faires,*)* afirma que: "*se A for positiva definida e 0 < w < 2, então o método SOR converge para qualquer escolha de aproximação inicial de* $x_0$".

Para o método do Gradiente Conjugado é possível mostrar que, para uma matriz A (n x n), ele converge em ate "n" etapas (*Numerical Analysis*, Burden e Faires, pag: 481).

O objetivo do trabalho é testar matrizes encontradas em problemas reais em ambos os métodos e comparar o numero de interações necessárias para convergência, bem como erro cometido nas aproximações das respostas. As matrizes utilizadas para o estudo foram:

- Trefethen_20 → dimensão: 20x20
- bcsstm22 → dimensão 138x138
- bcsstm34 → dimensão: 588x588
- bcsstm12 → dimensão: 1473x1473
- Trefethen_2000 → dimensão: 2000x2000

# Desenvolvimento

Todos os casos apresentados seguem uma mesma estrutura:

- Gráfico-1 → numero de interações X coeficiente de ajuste (w), tentando aproximar, visualmente, o coeficiente ótimo.
- Tabela-1 → Dados do método SOR
- Tabela-2 → Dados do método do Gradiente Conjugado

## JGD_Trefethen/Trefethen_20

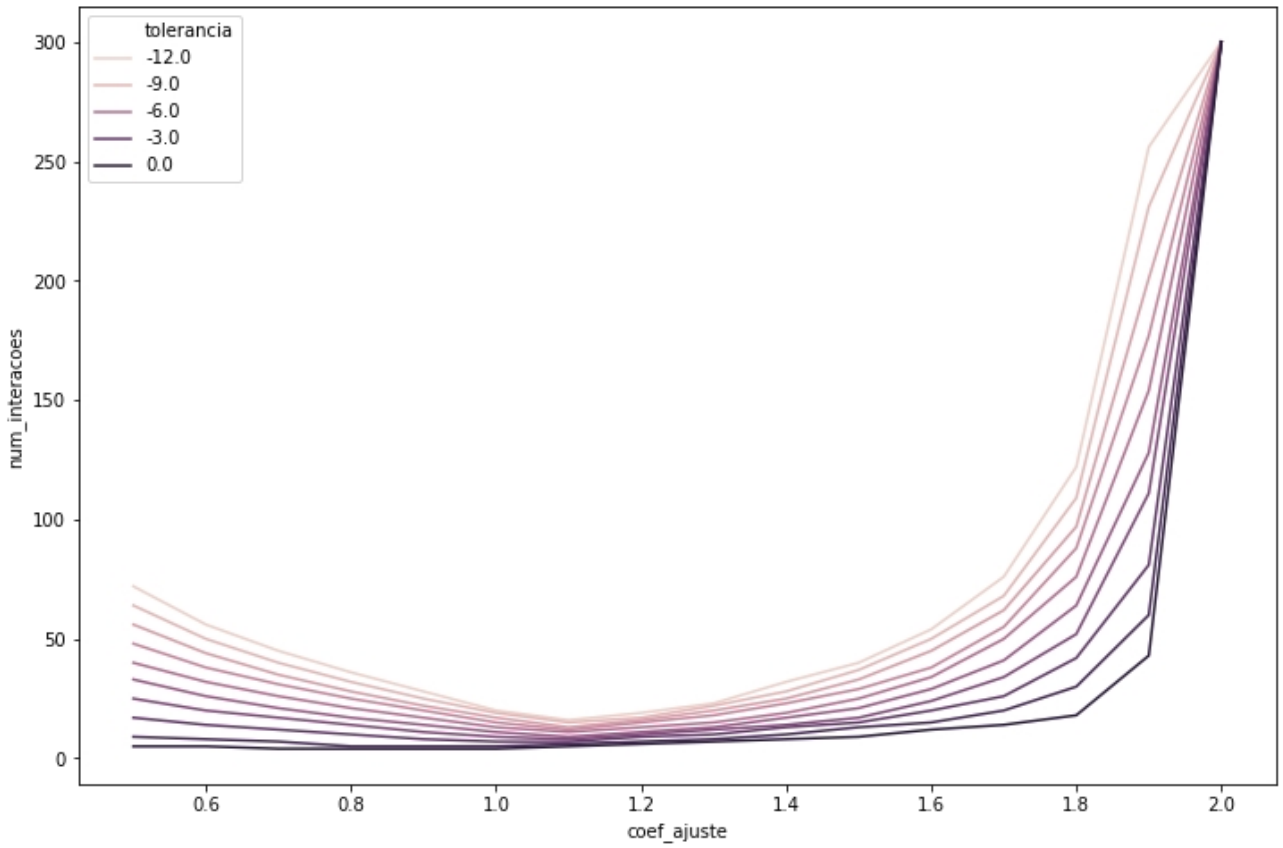
### Característica da matriz:

- Dimensão = 20x20
- Tipo = "*Combinatorial Problems*"
- Não nulos = 158

### Comparação dos métodos:

Implementando o algoritmo do método SOR, obteve-se o seguinte gráfico que mostram a relação entre coeficiente de ajuste (w) e numero de interações, sendo que cada cor representa um nível de tolerância (10^(-1) à 10^(-10)). Observa-se que o coeficiente w que produz o menor numero de interações para essa matriz está entre 1 e 1,2 para todas as tolerâncias.

Gráfico-1) coeficiente de ajuste (w) X numero de interações




Essa observação é confirmada quando analisamos os dados tabulados abaixo. Nota-se que, conforme a tolerância exigida para a convergência vai ficando menor, o erro de aproximação também vai diminuindo (o que era esperado).

Tabela-1) Método SOR

| | num_interações | erros | coeficiente_min (w) | tolerância |
|---|---|---|---|---|
| **0** | 4.0 | 6.879738e-05 | 1.0 | -1 |
| **1** | 5.0 | 6.731120e-06 | 1.0 | -2 |
| **2** | 7.0 | 3.901677e-10 | 1.1 | -3 |
| **3** | 8.0 | 6.001905e-12 | 1.1 | -4 |
| **4** | 9.0 | 3.120524e-12 | 1.1 | -5 |
| **5** | 11.0 | 3.150961e-16 | 1.1 | -6 |
| **6** | 12.0 | 3.073696e-18 | 1.1 | -7 |
| **7** | 13.0 | 1.454958e-18 | 1.1 | -8 |
| **8** | 15.0 | 6.629141e-23 | 1.1 | -9 |
| **9** | 16.0 | 2.518104e-24 | 1.1 | -10 |

Utilizando então o método do Gradiente Conjugado, observa-se que este necessita de um numero de interações maior do que o método de relaxação. Mesmo assim, para todas as tolerâncias, o método necessitou no máximo de 19 interações (menor do que a dimensão da matriz) e, a partir da tolerância de 10^(-3), seu erro de aproximação passa a ser de apenas 2.41e^(-26). Isso torna a aproximação mais precisa que todas as encontrada pelo método SOR.

Tabela-2) Método do Gradiente Conjugado

| | num_interações | tempos | erros | tolerância |
|---|---|---|---|---|
| **0** | 12 | 0.007109 | 8.667292e-03 | -1 |
| **1** | 18 | 0.016944 | 1.626809e-05 | -2 |
| **2** | 19 | 0.009425 | 2.416801e-26 | -3 |
| **3** | 19 | 0.009748 | 2.416801e-26 | -4 |
| **4** | 19 | 0.008686 | 2.416801e-26 | -5 |
| **5** | 19 | 0.008348 | 2.416801e-26 | -6 |
| **6** | 19 | 0.009314 | 2.416801e-26 | -7 |
| **7** | 19 | 0.008366 | 2.416801e-26 | -8 |
| **8** | 19 | 0.009776 | 2.416801e-26 | -9 |
| **9** | 19 | 0.008425 | 2.416801e-26 | -10 |

# HB/bcsstm22

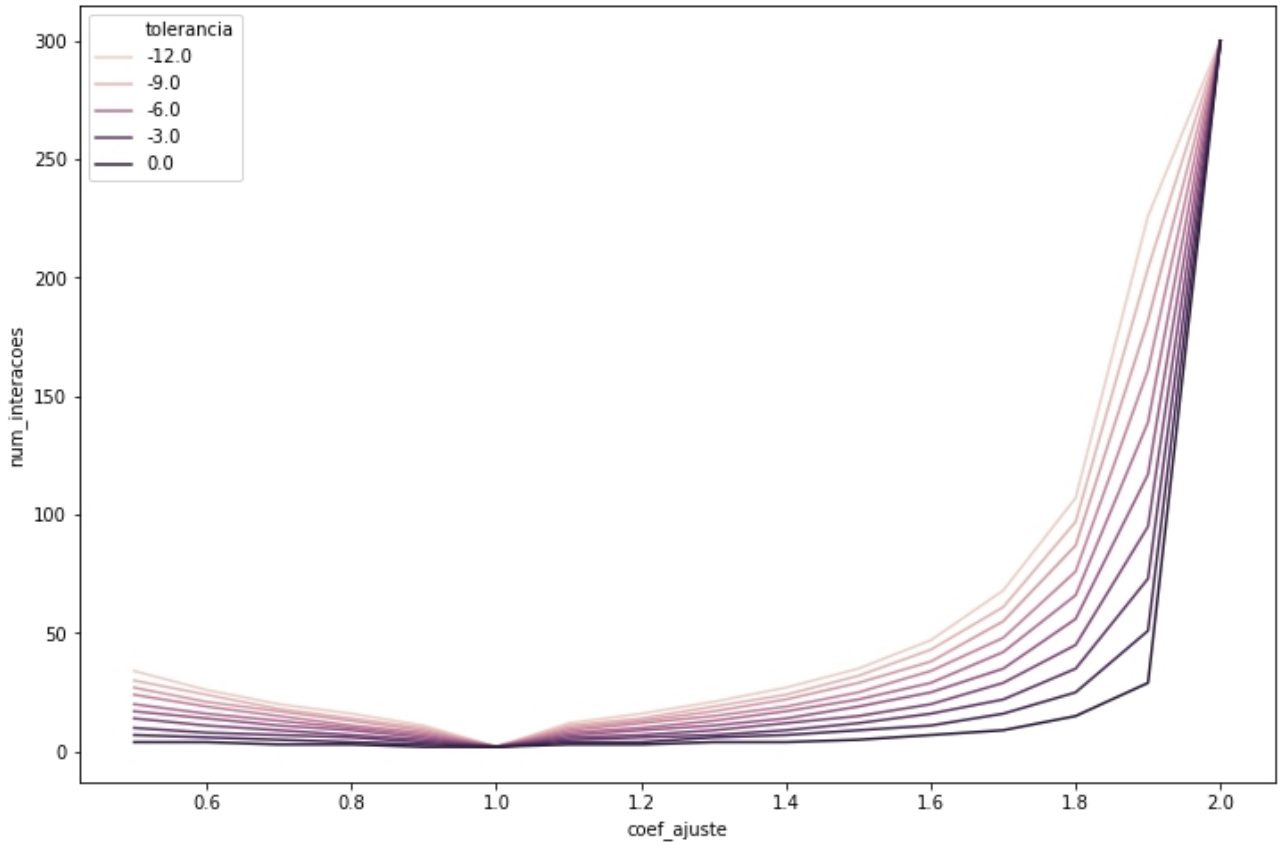
## Característica da matriz:

- Dimensão = 138x138
- Tipo = “*Structural Problem*”
- Não nulos = 138

## Comparação dos métodos:

Essa matriz é interessante de ser analisada, tendo em vista que ela possui apenas a diagonal principal com elementos não nulos. Pela forma como são realizadas as operações no método SOR, onde $x_i = (1-w)x0_i + (1/a_{ii})*(w)*(b_i - \Sigma^{j=i-1} a_{ij}x_j - \Sigma_{j=i+1} a_{ij}x_j)$ é possível demonstrar que ele deve convergir, para w=1, em 2 interações apenas. Isso pode ser observado nas tabelas abaixo.

Gráfico -1)  coeficiente de ajuste (w) X numero de interações




Como pode ser visto nas tabelas, o método do Gradiente Conjugado tem uma aproximação com um erro maior que o método SOR, mesmo para uma tolerância de 10^(-10).

Tabela-1) Método SOR

| | num_interações | erros | coeficiente_min (w) | tolerância |
|---|---|---|---|---|
| **0** | 2.0 | 7.888609e-31 | 1.0 | -1 |
| **1** | 2.0 | 7.888609e-31 | 1.0 | -2 |
| **2** | 2.0 | 7.888609e-31 | 1.0 | -3 |
| **3** | 2.0 | 7.888609e-31 | 1.0 | -4 |
| **4** | 2.0 | 7.888609e-31 | 1.0 | -5 |
| **5** | 2.0 | 7.888609e-31 | 1.0 | -6 |
| **6** | 2.0 | 7.888609e-31 | 1.0 | -7 |
| **7** | 2.0 | 7.888609e-31 | 1.0 | -8 |
| **8** | 2.0 | 7.888609e-31 | 1.0 | -9 |
| **9** | 2.0 | 7.888609e-31 | 1.0 | -10 |

Tabela-2) Método do Gradiente Conjugado

| | num_interações | tempos | erros | Tolerância |
|---|---|---|---|---|
| **0** | 0 | 0.000459 | 138.000000 | -1 |
| **1** | 0 | 0.001709 | 138.000000 | -2 |
| **2** | 0 | 0.000664 | 99.787534 | -3 |
| **3** | 1 | 0.000970 | 56.693736 | -4 |
| **4** | 3 | 0.001532 | 30.124477 | -5 |
| **5** | 7 | 0.002508 | 6.430305 | -6 |
| **6** | 11 | 0.003018 | 3.323497 | -7 |
| **7** | 13 | 0.004809 | 2.926047 | -8 |
| **8** | 22 | 0.004935 | 0.232913 | -9 |
| **9** | 25 | 0.004760 | 0.126674 | -10 |

Como a ordem das matrizes a seguir passou de 500x500, foi interessante analisar o tempo gasto na aplicação de cada algoritmo. Esses dados encontraram-se também nas tabelas de cada método.

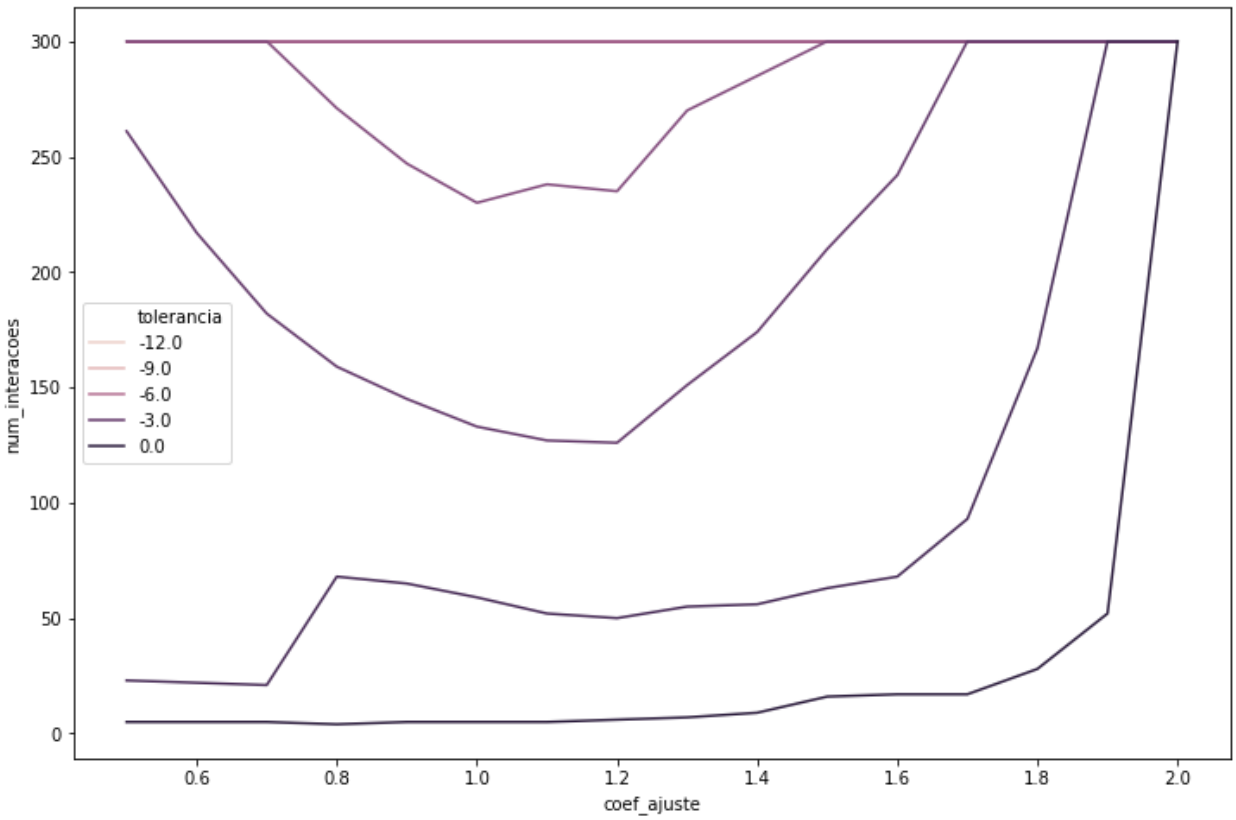
# Boeing/bcsstk34

## Característica da matriz:

- Dimensão = 588x588
- Tipo = “*Structural Problem*”
- Não nulos = 21.418

## Comparação dos métodos:

Gráfico -1)  coeficiente de ajuste (w) X numero de interações




Para essa matriz, o coeficiente w com o menor numero de interações foi se aproximando de 1,2, que produziu erros na casa de 10^(-5), enquanto que para todos os níveis de tolerância o método do Gradiente Conjugado produziu erros na casa de e^(-8).

Tabela-1) Método SOR

| | num_interações | tempos | erros | coeficiente_min (w) | tolerância |
|---|---|---|---|---|---|
| **0** | 4.0 | 0.068535 | 211.324536 | 0.8 | -1 |
| **1** | 21.0 | 0.360143 | 63.055212 | 0.7 | -2 |
| **2** | 126.0 | 2.362883 | 0.028098 | 1.2 | -3 |
| **3** | 230.0 | 4.293833 | 0.000441 | 1.0 | -4 |
| **4** | 300.0 | 5.483312 | 0.000013 | 1.2 | -5 |
| **5** | 300.0 | 5.521762 | 0.000013 | 1.2 | -6 |
| **6** | 300.0 | 5.204806 | 0.000013 | 1.2 | -7 |
| **7** | 300.0 | 5.119932 | 0.000013 | 1.2 | -8 |
| **8** | 300.0 | 5.552236 | 0.000013 | 1.2 | -9 |
| **9** | 300.0 | 5.899644 | 0.000013 | 1.2 | -10 |

Tabela-2) Método do Gradiente Conjugado

| | num_interações | tempos | erros | tolerância |
|---|---|---|---|---|
| **0** | 589 | 10.282191 | 6.378913e-08 | -1 |
| **1** | 589 | 10.774627 | 6.378913e-08 | -2 |
| **2** | 589 | 10.729409 | 6.378913e-08 | -3 |
| **3** | 589 | 11.057798 | 6.378913e-08 | -4 |
| **4** | 589 | 10.579430 | 6.378913e-08 | -5 |
| **5** | 589 | 11.534456 | 6.378913e-08 | -6 |
| **6** | 589 | 12.088627 | 6.378913e-08 | -7 |
| **7** | 589 | 11.577661 | 6.378913e-08 | -8 |
| **8** | 589 | 11.457863 | 6.378913e-08 | -9 |
| **9** | 589 | 11.124917 | 6.378913e-08 | -10 |

Como esperado, o tempo do Gradiente Conjugado foi superior ao do SOR por conta do maior numero de interações necessárias em cada nível de tolerância. Ambos os métodos atingiram o limite de interações do algoritmo, sendo que o SOR apenas a partir da tolerância de 10^(-5).

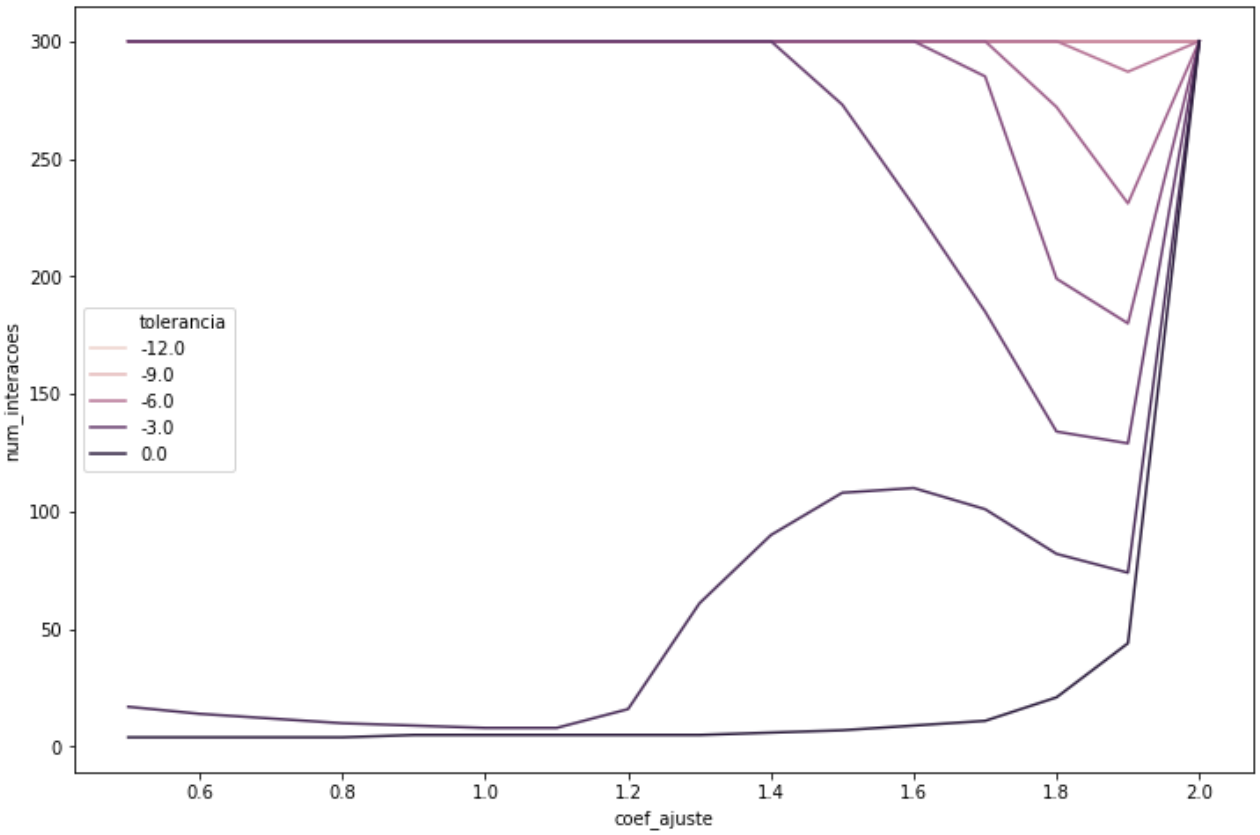
# HB/bcsstm12

## Característica da matriz:

- Dimensão = 1.473x1.473
- Tipo = “*Structural Problem*”
- Não nulos = 19.659

## Comparação dos métodos:

Gráfico -1)  coeficiente de ajuste (w) X numero de interações




O gráfico acima indica que a convergência do método SOR acontece para um coeficiente de ajuste w de aproximadamente 1,9. Isso pode ser observado para todos os níveis de tolerância acima de 10^(-3).

Tabela-1) Método SOR

| | num_interações | tempos | erros | coeficiente_min (w) | tolerância |
|---|---|---|---|---|---|
| **0** | 4.0 | 0.085582 | 1.153394e+03 | 0.5 | -1 |
| **1** | 8.0 | 0.132977 | 2.082075e+03 | 1.0 | -2 |
| **2** | 129.0 | 2.532660 | 2.569614e-03 | 1.9 | -3 |
| **3** | 180.0 | 3.320672 | 2.536563e-05 | 1.9 | -4 |
| **4** | 231.0 | 3.953764 | 2.399294e-07 | 1.9 | -5 |
| **5** | 287.0 | 4.978725 | 2.859114e-09 | 1.9 | -6 |
| **6** | 300.0 | 4.922331 | 1.319146e-09 | 1.9 | -7 |
| **7** | 300.0 | 5.761239 | 1.319146e-09 | 1.9 | -8 |
| **8** | 300.0 | 5.488956 | 1.319146e-09 | 1.9 | -9 |
| **9** | 300.0 | 5.495706 | 1.319146e-09 | 1.9 | -10 |

Tabela-2) Método do Gradiente Conjugado

| | num_interações | tempos | erros | tolerância |
|---|---|---|---|---|
| **0** | 27 | 0.446690 | 326.767791 | -1 |
| **1** | 37 | 0.592352 | 308.882130 | -2 |
| **2** | 55 | 0.857057 | 270.005068 | -3 |
| **3** | 119 | 1.712726 | 132.666683 | -4 |
| **4** | 276 | 4.239316 | 23.227879 | -5 |
| **5** | 420 | 6.145239 | 3.260564 | -6 |
| **6** | 601 | 9.482658 | 0.287149 | -7 |
| **7** | 749 | 11.891365 | 0.094499 | -8 |
| **8** | 1006 | 15.922458 | 0.008215 | -9 |
| **9** | 1239 | 19.376286 | 0.001036 | -10 |

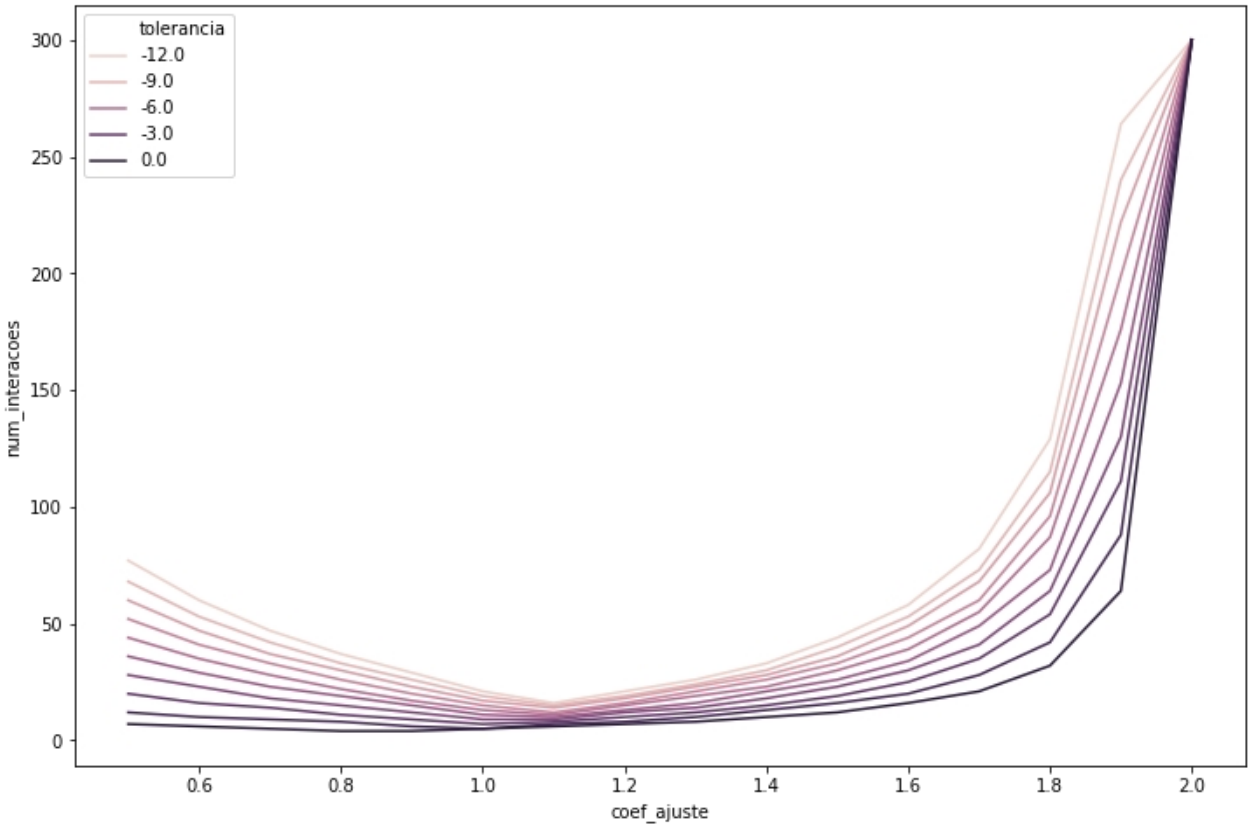
O método SOR nesse caso convergiu, com um erro abaixo de e^(-9), a partir da tolerância de 10^(-6), necessitando apenas de 300 interações (limite estipulado para o estudo) e consequentemente um tempo bem menor do que o Gradiente Conjugado.

# JGD_Trefethen/Trefethen_2000

## Característica da matriz:

- Dimensão = 2.000x2.000
- Tipo = “*Combinatorial Problem*”
- Não nulos = 41.906

## Comparação dos métodos:




O gráfico acima indica que a convergência do método SOR, para essa matriz, acontece para um coeficiente de ajuste w de aproximadamente 1,9. Isso pode ser observado para todos os níveis de tolerância acima de 10^(-3).

Como nos caso 1 e 3 apresentados, a convergência da matriz é mais comportada no método SOR, o que pode ser observado também no erro do método do Gradiente Conjugado.

Tabela-1) Método SOR

| | **num_interações** | **tempos** | **erros** | **coeficiente_min (w)** | **tolerância** |
|---|---|---|---|---|---|
| **0** | 4.0 | 0.398627 | 1.764047e-03 | 0.9 | -1 |
| **1** | 5.0 | 0.512050 | 5.070766e-06 | 1.0 | -2 |
| **2** | 7.0 | 0.744521 | 5.443794e-08 | 1.0 | -3 |
| **3** | 9.0 | 0.998746 | 1.746393e-11 | 1.1 | -4 |
| **4** | 10.0 | 1.055825 | 1.515257e-13 | 1.1 | -5 |
| **5** | 11.0 | 1.132477 | 2.347338e-15 | 1.1 | -6 |
| **6** | 12.0 | 1.306251 | 1.826483e-16 | 1.1 | -7 |
| **7** | 14.0 | 1.514774 | 1.585296e-19 | 1.1 | -8 |

| | num_interações | tempos | erros | coeficiente_min (w) | tolerância |
|---|---|---|---|---|---|
| **8** | 15.0 | 1.625636 | 8.952489e-22 | 1.1 | -9 |
| **9** | 16.0 | 1.732528 | 2.016797e-23 | 1.1 | -10 |

Tabela-2) Método do Gradiente Conjugado

| | num_interações | tempos | erros | tolerância |
|---|---|---|---|---|
| **0** | 343 | 36.167318 | 8.861893e-05 | -1 |
| **1** | 377 | 39.253077 | 1.029809e-06 | -2 |
| **2** | 399 | 42.118171 | 8.671782e-08 | -3 |
| **3** | 420 | 43.842074 | 4.552042e-09 | -4 |
| **4** | 437 | 46.039737 | 3.411034e-10 | -5 |
| **5** | 453 | 47.797522 | 1.263255e-11 | -6 |
| **6** | 464 | 48.770559 | 1.304409e-12 | -7 |
| **7** | 476 | 50.320380 | 1.829840e-13 | -8 |
| **8** | 490 | 51.817790 | 1.474546e-14 | -9 |
| **9** | 502 | 54.270889 | 1.239023e-15 | -10 |

Ambos os algoritmos convergiram com um erro baixo, mas o método de relaxação superou o método do Gradiente Conjugado tanto em numero de interações, quanto no erro.

## Representação COO da matriz  X representação normal

A ideia do estudo dos métodos mostrados utilizou uma representação diferente da usual das matrizes, a representação COO. Os algoritmos desenvolvidos visaram se beneficiar dessa estrutura, realizando um numero menor de operações, sendo que grande parte dos elementos das matrizes eram nulos.

A tabela abaixo, mostra o beneficio de trabalhar com essa representação, quanto ao tempo gasto pelo algoritmo conforme a dimensão das matrizes foi aumentando. Olhando a esparsidade da matriz, podemos ver quantas vezes o tempo da representação normal superou o tempo da representação COO.

Considerou-se apenas o método de relaxação SOR, com uma média dos tempos gastos para cada coeficiente w, e a tolerância de 10^(-3), tendo em vista que a análise envolveu diferentes níveis de tolerância, como exposto nos casos acima.

Tabela-1) Tempo (COO) médio X Tempo (Normal) médio

| **Matrizes** | **Trefethen_20** | **bcsstm22** | **bcsstk34** | **bcsstm12** | **Trefethen_2000** |
|---|---|---|---|---|---|
| **Esparsidade** | 60,5% | 99,2% | 93,8% | 99% | 98,9% |
| **Tempo (COO)** | 0,01923 | 0,0073 | 3,9188 | 5,0563 | 4,2885 |
| **Tempo (Normal)** | 0,01331 | 0,2691 | 35,1854 | 199,2312 | 73,5716 |
| **Tempo (Normal) = coef * Tempo (COO)** | 1,44x | 36.86x | 8.98x | 39.40x | 17.15x |

# Resultados encontrados

Em todos os casos apresentados, tanto o método de relaxação SOR, quanto o método do Gradiente Conjugado sem pré-processamento, mostraram-se eficientes, com erros cada vez menores conforme diminuiu-se a tolerância. Esse fato esta de acordo com o que foi apresentado na teoria e mostra a eficiência deles diante de problemas de métodos iterativos.

Um ponto importante a ser destacado é que, como o numero de interações máximo foi fixado para o método SOR, pode ser que a aproximação do x ficasse melhor caso fossem realizadas mais interações. Isso pode fazer com que este método seja inferior ao método do Gradiente Conjugado, para matrizes de ordens maiores.

Outro ponto relevante foi que o coeficiente de ajuste (w) do método SOR se aproximou de um único valor, conforme diminuiu-se a tolerância exigida. Isso mostra que o w ótimo mantém-se o mesmo a partir de determinado nível de tolerância.

Por fim, o tempo computacional mostrou-se diretamente ligado a quantidade de passagens necessárias em cada método e a estrutura da matriz. Matrizes mais esparsas tendem a levar um tempo menor, por realizarem menos operações que matrizes mais cheias (considerando ambas convergindo em números semelhantes de interações). Isso pode ser visto comparando os tempos médios da “*Trefethen_20*” e da “*bcsstm22*” que possuem, respectivamente, ordem 20x20 e 138x183.

# Verificação dos dados

Para verificação de todas informações expostas acima, é possível utilizar o programa T2.py que possui as funções “*GradienteConjugadoNormalCOO*” e “SOR_*SparseMatrix*”. O produto entre matriz vetor, considerando a matriz no formato COO, é feito pela função “*ProdutoVetorMatrizCOO*” no mesmo arquivo.

Um jupyter notebook “a*nalise_T2*”, que pode ser encontrado no mesmo diretório, possui uma sequencia de passos para geração dos dados, tabelas e gráficos expostos acima. Todos os dados, para cada matriz, estão salvos em pastas nomeadas por “analise_” + “nome da matriz”, que possuem os seguintes arquivos:

- **comparacao_SOR.csv** → Tabela com os dados para os coeficiente de ajuste (w) ótimos
- **gradienteConjugado.csv** → Tabela com os dados do método do Gradiente Conjugado, sem trabalhar com a configuração esparsa da matriz.
- **gradienteConjugado_COO.csv** → Idem do arquvio acima, mas trabalhando com a configuração COO
- **SOR.csv** → Tabela com os dados do método do Gradiente Conjugado, sem trabalhar com a configuração esparsa da matriz
- **SOR_COO.csv** → Idem do arquivo acima, mas trabalhando com a configuração esparsa.