

# Resultados obtidos no 1º exercício computacional

## Algoritmos numéricos I

Leandro Furlam Turi

05 de setembro de 2019

### 1 Preparativos

A escolha das matrizes esparsas utilizadas neste exercício computacional se deu única e exclusivamente observando o fato de elas serem quadradas e inversíveis, com exceção da matriz de ordem 10E5, onde foi necessário buscar uma matriz que houvesse menor quantidade de termos não nulos para que todas as operações de ponto flutuante ocorressem de maneira viável. Ressalta-se ainda que o cálculo de condicionamento (K) para matrizes de ordem maiores que 10E4 não ocorreu de maneira adequada, através da utilização da função *cond()*, gerando o aviso de que a saída da memória ou a dimensão da matriz são muito grandes para o *Octave*. Diante disto, foi calculada de maneira teórica, através das normas de infinito das matrizes, através da equação a seguir:

$$K = \|A^{-1}\|_{\text{inf}} * \|A\|_{\text{inf}}$$

Ainda que pela formulação teórica seja possível apresentar resultados, após dois dias de execução (até a entrega do trabalho), não foi terminado o cálculo do coeficiente de condicionamento para a matriz de ordem 10E5. Isto pode ser explicado pelo grande número de operações necessárias para o cálculo da inversa da matriz.

### 2 Resultados

A Tabela 1 apresenta os resultados obtidos para as simulações.

Dimensão	taxa [%]	dist_x	dist_A	dist_b	norm_res	K
48	7.3839e+01	1.6631e-13	1.4692e+00	4.0056e-16	1.4305e-06	1.5976e+06
662	7.7282e+01	9.3259e-14	1.2856e+00	3.1525e-16	4.5475e-13	8.2712e+05
2003	9.1077e+01	1.5854e-11	1.2806e+00	1.5724e-15	9.6703e-04	4.5698e+10
15575	8.5971e+01	1.0958e-13	5.1826e-01	2.9984e-16	1.1369e-13	5.5549e+05
109460	9.8998e+01	1.9335e-06	1.3548e-01	1.8691e-16	5.6843e-14	...

Tabela 1: Resultados obtidos

Nota-se, através da Tabela 1, a alta taxa de preenchimento das matrizes resultantes da fatoração LU, em relação à matriz original. Isto sugere que o método, para matrizes esparsas de ordem como a dos testes realizados, não é uma boa alternativa para resolução de sistemas lineares, uma vez que não faz proveito da esparsidade da matriz, tampouco resulta em outras matrizes que sejam esparsas, aumentando, assim, os gastos com memória e tempo de processamento.

Ainda, através da ordem elevada do condicionamento das matrizes, pode-se indicar que elas são mau condicionadas. Diante disto, pequenas perturbações nas entradas, que podem ser geradas pelos erros de ponto flutuante durante as operações realizadas na fatoração LU ou até mesmo erros de representação, geram erros consideráveis que vão se acumulando na solução de um sistema linear.

Em todos os casos analisados, quando observada unicamente a distância relativa entre a matriz original e a matriz resultante da decomposição LU, não se pode concluir que as matrizes são mau condicionadas, uma vez que a ordem de  $dist_A$  é baixa, quando comparada às dimensões das matrizes originais.

Para as matrizes de ordem inferior a  $10E4$ , não ocorreram erros consideráveis entre a solução exata e a solução aproximada ( $dist_x$ ), e entre o vetor dos termos independentes original e o vetor resultante da decomposição LU ( $dist_b$ ) (quando comparado ao  $eps$  do *Octave*) uma vez que suas distâncias relativas possuem ordens baixas. Contudo, observando a matriz de ordem  $10E5$ , nota-se que houve algum acúmulo dos erros durante as operações de pontos flutuantes. Isto pode ser explicado pelo valor de condicionamento  $K$ , que possui ordem elevada nesta matriz e em todas as demais.

### 3 Conclusões

O fato de o método LU apresentar soluções aproximadas para matrizes de ordens baixas demonstra que um mau condicionamento não afeta diretamente estas matrizes, fato este que não pode ser observado na matriz de ordem  $10E5$ .

Logo, o método LU, quando aplicado a matrizes esparsas, não mostrou ser uma boa alternativa para a solução de sistemas lineares. Embora apresente boa aproximação da solução, ele gera um gasto excessivo de memória, como pode ser observado pelas taxas de preenchimento, o que em matrizes de ordens elevadas pode ser prejudicial ao desempenho do algoritmo e à saúde da máquina.