

Projeto II: Comparação de diferentes abordagens para o problema de quadrados mínimos lineares

Bryan Alves Do Prado Raimundo - RA 195171

Jean-Yves Roulet - RA 170315

Thiago Felipe Castro Carrenho - RA 224831

Vinicius Jameli Cabrera - RA 225414

08 de Junho de 2020



UNICAMP

Universidade Estadual de Campinas

Instituto de Matemática, Estatística e Computação Científica

MS512A Análise Numérica I 1S2020

Prof.: Sandra Augusta Santos

Sumário

| | | |
|----------|--|-----------|
| 1 | Introdução | 3 |
| 2 | Desenvolvimento | 3 |
| 2.1 | Planejamento Inicial | 3 |
| 2.2 | Parte I: Comparando diferentes abordagens | 3 |
| 2.2.1 | Problema Teste | 4 |
| 2.2.2 | Preparação | 5 |
| 2.2.3 | Procedimento pelo método (a) Solução das equações normais via fatoração de Cholesky | 5 |
| 2.2.4 | Procedimento pelo método (b) Solução via fatoração QR condensada obtida pelo processo de Gram-Schmidt modificado | 5 |
| 2.2.5 | Procedimento pelo método (c) Solução via fatoração QR completa usando transformações de Householder | 6 |
| 2.2.6 | Análise e Discussão | 6 |
| 2.2.7 | Conclusão | 10 |
| 2.3 | Parte II: Algo Mais | 11 |
| 2.3.1 | Introdução | 11 |
| 2.3.2 | Método Utilizado | 11 |
| 2.3.3 | Análise e Resultados | 13 |
| 2.3.4 | Conclusão | 13 |
| 3 | Conclusão | 14 |
| 4 | Apêndice | 15 |
| 4.1 | Script que gera os dados do problema | 15 |
| 4.2 | Script que computa a solução para o método (a) | 16 |
| 4.3 | Script que computa a solução para o método (b) | 17 |
| 4.4 | Script que computa a solução para o método (c) | 18 |
| 4.5 | Script que computa o número de condição das matrizes necessárias | 19 |
| 4.6 | Script que monta a tabela dos coeficientes e plota o gráfico | 20 |
| 4.7 | Script adaptado para a montagem da matriz A da parte 2 | 21 |
| 4.8 | Tabela de casos acumulados de COVID-19 por dia | 22 |
| 4.9 | Vetor obtido na Parte 2 | 23 |
| 4.10 | Gráfico oficial do ministério da saúde | 24 |
| | Referências Bibliográficas | 25 |

1 Introdução

Este projeto tem a finalidade de testar a qualidade da solução obtida por diferentes métodos para resolver o problema de quadrados mínimos (QM) lineares, dado por

$$\min_{x \in \mathbb{R}^n} \frac{1}{2} \|Ax - b\|_2^2 \quad (1)$$

em que $A \in \mathbb{R}^{m \times n}$ e $b \in \mathbb{R}^m$.

Para isto, três métodos foram testados e implementados no sistema algébrico matricial (SAM) OCTAVE, escolhido pelo grupo. Os métodos explorados se baseiam na fatoração da matriz A a fim de resolver o sistema de equações normais decorrente do problema de minimização (1).

Para este projeto, usamos como referência o livro-base da matéria Watkins [2010], e o livro Shores [2007], que baseou o enunciado do Projeto.

2 Desenvolvimento

2.1 Planejamento Inicial

Dado a simplicidade desse projeto em relação ao projeto anterior, foi mais tranquilo a divisão de afazeres entre os componentes, de acordo com o fluxograma abaixo:

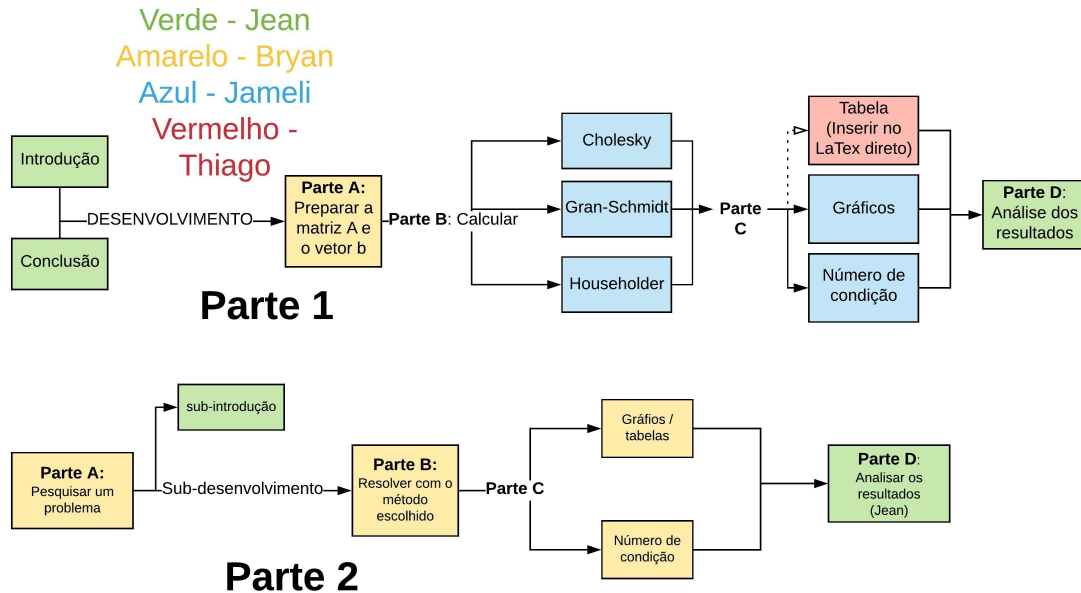


Figura 1: Divisão de Afazeres

2.2 Parte I: Comparando diferentes abordagens

O problema de quadrados mínimos ilustrado por (1) incorpora, no caso em que $m > n$, a ideia da minimização do resíduo definido como $r = Ax - b$, ou equivalentemente, do quadrado de sua norma-2. É

conveniente trabalhar com a expressão $\frac{1}{2}\|Ax - b\|_2^2$ a fim de simplificar o cálculo de gradiente, já que a busca pelo valor minimizador x leva intuitivamente a considerar os pontos estacionários da expressão de interesse. Neste caso o gradiente é dado por $A^t Ax - A^t b$ e se anula quando a seguinte igualdade é satisfeita:

$$A^t Ax = A^t b \quad (2)$$

Refere-se comumente à esta equivalência como "Sistema de equações normais", associada ao problema de quadrados mínimos. Se A for uma matriz de posto completo, $\text{posto}(A) = n$, então $A^t A$ será positiva definida e existirá solução única para (2), esta podendo ser didaticamente explicitada por:

$$x = (A^t A)^{-1} A^t b \quad (3)$$

Além disso, a quadrática que se deseja minimizar $\frac{1}{2}\|Ax - b\|_2^2$ será um paraboloide elíptico com um único ponto de mínimo (*concavidade para cima*), o que garante que (2) possuirá uma solução única bem definida. Se b não pertencer à imagem de A , o minimizador x encontrado não anulará o resíduo, o que só acontecerá caso b pertença à imagem de A .

Vale salientar que se $m \leq n$ então $Ax = b$ apresentará uma única solução ou infinitas soluções e portanto o problema (1) é despropositado, justificando a atenção dada somente aos casos em que $m > n$.

A forma explícita (3) do problema de quadrados mínimos (1) não apresenta utilidade na prática, já que o cálculo da inversa de $A^t A$ é excessivamente caro computacionalmente. Neste cenário técnicas inteligentes foram desenvolvidas a fim de resolver diretamente o sistema de equações normais (2). Neste trabalho se analisará o desempenho de três métodos para resolução de tal sistema baseados na fatoração da matriz A , especificamente via:

- (a) Fatoração de Cholesky
- (b) Fatoração QR condensada através do processo de Gram-Schmidt modificado
- (c) Fatoração QR completa através de transformações de Householder

2.2.1 Problema Teste

Visando aproximar a curva $f(x) = \exp(\sin(6x))$ no intervalo $[0, 1]$ por um polinômio de grau 10, considere como dados de entrada os valores computados para $f(x)$ nos pontos igualmente espaçados $x_k = kh$, $k = 0, 1, \dots, 20$ (logo, $h = 0.05$). Dessa forma, haverão 21 equações lineares:

$$f(x_k) = c_0 + c_1 x_k + c_2 x_{2k} + \dots + c_{10} x_{10k}, \text{ com } k = 0, 1, \dots, 20 \quad (4)$$

em que as 11 incógnitas serão os coeficientes c_0, c_1, \dots, c_{10} do polinômio aproximador. A matriz dos coeficientes desse problema é conhecida como *matriz de Vandermonde* e o OCTAVE possui comandos pré-programados para construção automática de tais matrizes.

É importante destacar que uma matriz de Vandermonde formada a partir de pontos distintos entre si, sendo os x_k 's na notação estabelecida acima, tem determinante não nulo já que seu valor é dado por $\prod_{i < j} (x_j - x_i)$ (ver Skoufranis [2012] para uma dedução simples desta propriedade). Desta forma $A^t A$ é positiva definida e existe solução única para o problema de quadrados mínimos.

2.2.2 Preparação

O algoritmo em OCTAVE que gera a matriz dos coeficientes de A e o vetor do sistema linear está no Apêndice 4.1.

2.2.3 Procedimento pelo método (a) Solução das equações normais via fatoração de Cholesky

Como $A^t A$ é simétrica e positiva definida então $A^t A$ admite fatoração de Cholesky e esta pode ser utilizada para resolver o sistema (2). Tem-se que $A^t A x = R^t R x = A^t b$. Nomeando $y = R x$ se obtém os sistemas equivalentes seguintes:

$$R^t y = A^t b \quad (5)$$

$$R x = y \quad (6)$$

Tais sistemas são simples de serem resolvidos sequencialmente já que R é uma matriz triangular superior.

O algoritmo em OCTAVE que computa a fatoração de Cholesky e resolve o sistema de equações normais do problema teste está no Apêndice 4.2.

2.2.4 Procedimento pelo método (b) Solução via fatoração QR condensada obtida pelo processo de Gram-Schmidt modificado

É fundamental perceber a validade da utilização da fatoração QR no problema abordado. Isto se deve novamente ao fato da matriz A ter posto completo e por consequência $A^t A$ ser simétrica e positiva definida. Embora a fatoração de Cholesky apresenta-se como um método viável para a resolução do sistema de equações normais num primeiro momento, é preciso trabalhar diretamente com a matriz $A^t A$, o que se torna computacionalmente ineficiente à medida que o número de elementos de A aumenta.

A fatoração QR contorna tal obstáculo trabalhando diretamente com a matriz A e permite a obtenção da matriz R da fatoração de Cholesky da matriz $A^t A$, significativa melhoria em relação ao método anterior. Quando tal fatoração é realizada através do processo de Gram-Schmidt somente parte da matriz Q é obtida, a associada à imagem da matriz A e por isso tal procedimento é denominado de fatoração condensada.

Realizando a fatoração QR da matriz $A_{m \times n}$ por Gram-Schmidt obtém-se uma matriz $Q_1_{m \times n}$, com colunas ortonormais, e a matriz triangular superior $R_{n \times n}$, fator de Cholesky de $A^t A$. A matriz Q completa é escrita como $\begin{pmatrix} Q_1 & Q_2 \end{pmatrix}$, onde $Q_2_{m \times (m-n)}$ é associada ao núcleo de A .

A partir desta fatoração escreve-se $A = Q_1 R$ e o sistema normal (2) se torna $A^t A x = R^t Q_1^t Q_1 R x = R^t R x = A^t b = R^t Q_1^t b$. Como R é fator de Cholesky (não singular) podemos pré-multiplicar ambos os lados da igualdade por R^{-t} e obter a seguinte igualdade:

$$R x = Q_1^t b \quad (7)$$

A equação acima equivale à um simples sistema triangular superior, podendo ser eficientemente resolvido computacionalmente.

Há diversas maneiras de implementar o processo de Gram-Schmidt, sendo que escolheu-se o método de Gram-Schmidt modificado a fim de aumentar a estabilidade numérica durante a decomposição.

Foi desenvolvido um algoritmo em OCTAVE que calcula fatoração QR condensada pelo processo de Gram-Schmidt modificado (perceba que só há um *loop* em i) e resolve o sistema de equações normais do problema teste. Este algoritmo está disponível no Apêndice 4.3.

2.2.5 Procedimento pelo método (c) Solução via fatoração QR completa usando transformações de Householder

Uma outra abordagem para a decomposição QR da matriz A é a sua fatoração através de transformações de Householder. A ideia base para este método é a busca da triangularização da matriz A via reflexões. Neste método obtém-se $A = QR$ sendo Q $m \times m$ e R $m \times n$. Como neste caso a matriz Q possui m colunas ortonormais a fatoração é chamada de completa.

Além disso vale explicitar que a matriz R não é o fator de Cholesky de $A^t A$ porém vale $A^t A = R^t R$. Aproveitando a ortogonalidade de Q escreve-se o sistema normal (2) como $A^t A x = R^t Q^t Q R x = R^t R x = A^t b = R^t Q^t b$. Recorrendo ao fato de que se trabalhando com $m > n$ as n primeiras linhas de R são linearmente independentes, formando a matriz \hat{R} $n \times n$, e as outras $(m - n)$ linhas são nulas, então a igualdade $R^t R x = R^t Q^t b$ pode-ser pré-multiplicada por \hat{R}^{-t} $n \times n$ e obtém-se:

$$R x = Q^t b \quad (8)$$

Este é um sistema fácil de ser resolvido justamente devido a estas características da matriz R .

O algoritmo em OCTAVE que computa a fatoração QR completa com comandos nativos do próprio OCTAVE, os quais implementam as transformações de Householder, e resolve o sistema de equações normais está no Apêndice 4.4.

2.2.6 Análise e Discussão

O algoritmo que calcula o número de condição das matrizes $A^t A$ e das matrizes R dos métodos (a), (b) e (c) está no Apêndice 4.5, e o algoritmo que une todas as funções acima e plota os gráficos está no Apêndice 4.6.

Obteve-se a tabela de coeficientes, que, no caso do algoritmo, é representado por T_x. A seguir mostra-se tal tabela. Em primeiro a tabela gerada através do OCTAVE:

| | 1 | 2 | 3 |
|----|---------|---------|---------|
| 1 | 0.99926 | 0.99926 | 0.99926 |
| 2 | 7.5067 | 7.5069 | 7.5069 |
| 3 | -33.856 | -33.865 | -33.865 |
| 4 | 622.11 | 622.22 | 622.22 |
| 5 | -3523.1 | -3523.8 | -3523.8 |
| 6 | 7500.6 | 7503.3 | 7503.3 |
| 7 | -4118.8 | -4124.8 | -4124.8 |
| 8 | -8956.2 | -8947.7 | -8947.7 |
| 9 | 17038 | 17031 | 17031 |
| 10 | -11256 | -11252 | -11252 |
| 11 | 2719.4 | 2718.8 | 2718.8 |

Figura 2: Tabela do OCTAVE: Coeficientes computados

Em seguida a tabela montada da maneira requisitada no enunciado do projeto:

| | método (a) | método (b) | método (c) |
|----------|------------|------------|------------|
| c_0 | 0.99926 | 0.99926 | 0.99926 |
| c_1 | 7.5067 | 7.5069 | 7.5069 |
| c_2 | -33.856 | -33.865 | -33.865 |
| c_3 | 622.11 | 622.22 | 622.22 |
| c_4 | -3523.1 | -3523.8 | -3523.8 |
| c_5 | 7500.6 | 7503.3 | 7503.3 |
| c_6 | -4118.8 | -4124.8 | -4124.8 |
| c_7 | -8956.2 | -8947.7 | -8947.7 |
| c_8 | 17038 | 17031 | 17031 |
| c_9 | -11256 | -11252 | -11252 |
| c_{10} | 2719.4 | 2718.8 | 2718.8 |

Tabela 1: Tabela dos coeficientes computados

Analisando os dados expostos na tabela acima verifica-se grande compatibilidade entre os resultados. A utilização dos métodos (b) e (c) propiciaram os mesmos valores dos coeficientes da função $f(x)$ estudada. Além disso, os valores gerados através do método (a) confirmam a intuição de que apresentaria um desempenho inferior aos outros dois métodos, já que trabalha diretamente com a matriz $A^t A$ acarretando em significativas discrepâncias nos coeficientes computados pois tal matriz é consideravelmente mal condicionada, como pode ser visto pelo seu número de condição exposto nas duas tabelas apresentadas na sequência sendo respectivamente a tabela obtida diretamente do OCTAVE e a tabela montada evidenciando a norma utilizada para os números de condição.

| | | | |
|----------|--------|-----|----------------|
| cond_AtA | double | 1x1 | 5.3849e+14 |
| cond_R_a | double | 1x1 | 23207466.39114 |
| cond_R_b | double | 1x1 | 23175496.81216 |
| cond_R_c | double | 1x1 | 23175496.81089 |

Figura 3: Tabela do OCTAVE: Números de Condição-2

| | |
|-----------------|-------------------|
| $cond_2(A^t A)$ | 5.3849e+14 |
| $cond_2(R_a)$ | 2.320746639114e+7 |
| $cond_2(R_b)$ | 2.317549681216e+7 |
| $cond_2(R_c)$ | 2.317549681089e+7 |

Tabela 2: Tabela dos números de condição-2

Na tabela acima também são expostos os números de condição para as matrizes R : R_a , R_b e R_c respectivamente montadas durante os métodos (a), (b) e (c). A análise do número de condição destas matrizes reforça o fenômeno percebido durante a análise anterior de que processo (a) apresenta desempenho inferior aos outros processos, R_a possui o maior número de condição o que está associado à menor estabilidade e precisão do processo quando comparado aos outros métodos.

Fato interessante é que os dados desta tabela também permitem diferenciar os métodos (a) e (b) pois o número de condição de R_c é levemente inferior ao de R_b . Isto é indicativo de melhor desempenho da fatoração QR quando realizada usando transformações de Householder do que pelo processo de Gram-Schmidt.

É válido acrescentar que o ganho de estabilidade e precisão adquirido ao migrar da resolução do sistema normal a partir da fatoração de Cholesky para a fatoração QR é bem superior ao ganho análogo quando se passa do uso da fatoração QR via Householder para Gram-Schmidt, como se verifica facilmente comparando-se os números de condição das matrizes R da tabela citada.

Abaixo são apresentados gráficos relativos aos valores aproximados obtidos para a função $f(x)$ através dos métodos utilizados em comparação com o resultado $\exp(\sin(6x))$. Os três primeiros gráficos expõem separadamente os resultados adquiridos (Figuras 4, 5, 6). O último gráfico apresenta uma sobreposição de todos os resultados (Figura 7).

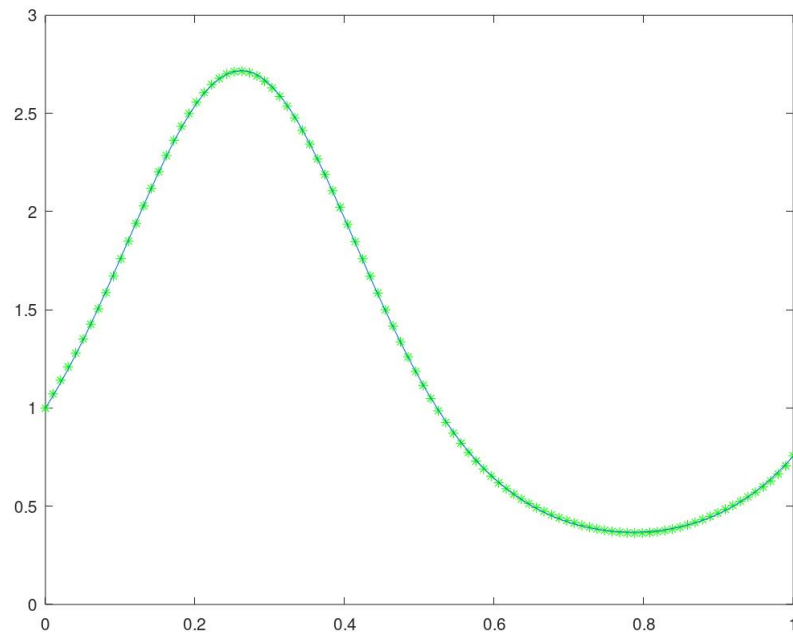


Figura 4: 100 pontos igualmente espaçados de $f(x)$ no intervalo $[0,1]$ obtidos da aproximação polinomial com os coeficientes calculados pelo método (a) sobrepostos à curva suave $\exp(\sin(6x))$

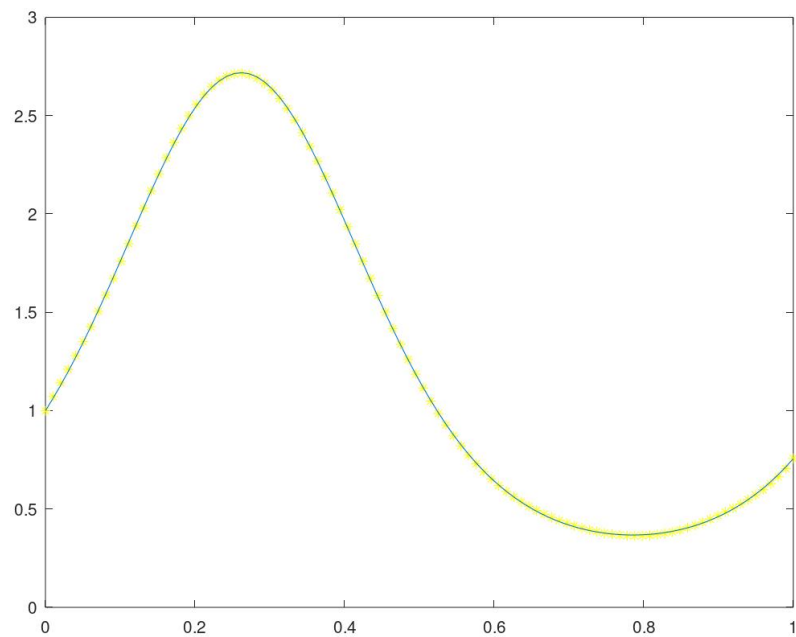


Figura 5: 100 pontos igualmente espaçados de $f(x)$ no intervalo $[0,1]$ obtidos da aproximação polinomial com os coeficientes calculados pelo método (b) sobrepostos à curva suave $\exp(\sin(6x))$

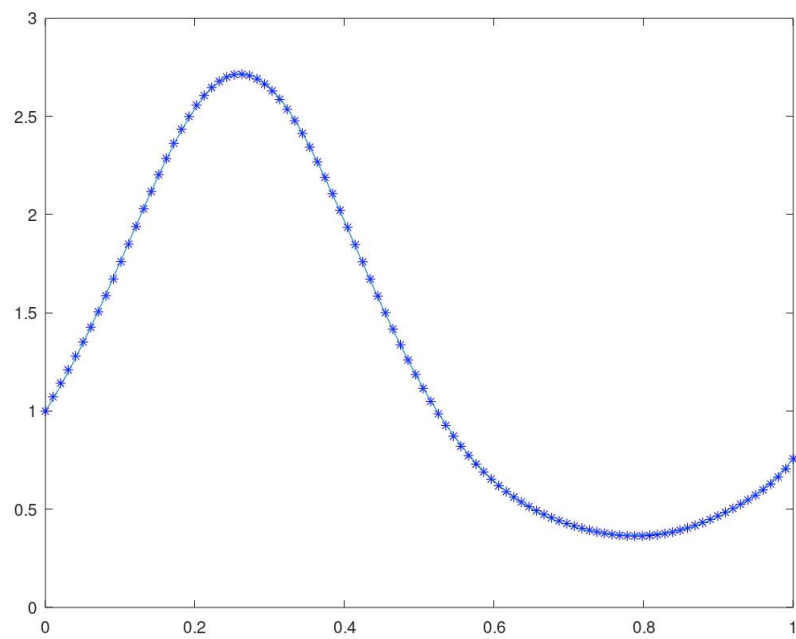


Figura 6: 100 pontos igualmente espaçados de $f(x)$ no intervalo $[0,1]$ obtidos da aproximação polinomial com os coeficientes calculados pelo método (c) sobrepostos à curva suave $\exp(\sin(6x))$

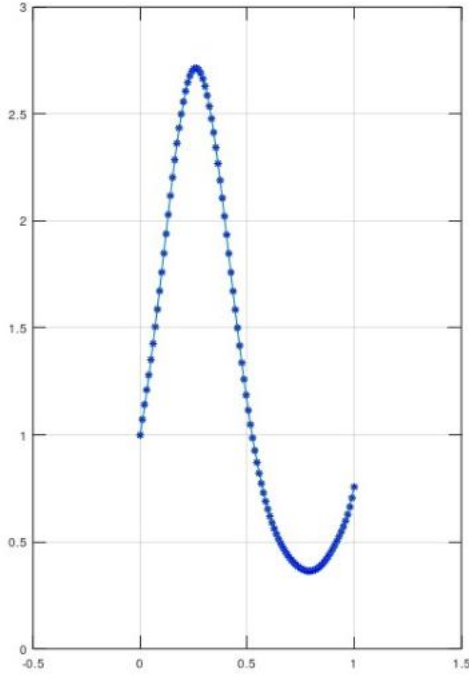


Figura 7: 100 pontos igualmente espaçados de $f(x)$ no intervalo $[0,1]$ obtidos da aproximação polinomial com os coeficientes calculados pelo método (a), (b) e (c) sobrepostos à curva suave $\exp(\sin(6x))$

A diferença entre os valores obtidos é sutil demais para ser visível no gráfico logo acima, mas foi possível percebê-la aplicando diretamente *zoom's* sucessivos nesta imagem utilizando os recursos do OCTAVE.

Em termos de custo computacional, a decomposição de $A^t A$ via fatoração de Choleky demanda $\frac{n^3}{3}$ operações ou $O(n^3)$, já a fatoração QR de A pelo processo de Gram-Schmidt exige $2(n^2 - n)m$ operações ou $O(m, n^2)$ e por fim a fatoração QR de A utilizando transformações de Householder precisa de $2n^2(m - \frac{n}{3})$ operações ou $O(m, n^2)$. Além disso, para resolver o sistema de equações normais é preciso resolver mais dois sistemas triangulares no caso (a) e mais um sistema triangular nos casos (b) e (c), o que não altera o custo computacional requerido considerando-se somente as fatorações já que a resolução de um sistema triangular via eliminação Gaussiana tem custo $O(n^2)$.

2.2.7 Conclusão

Conclui-se portanto que em relação aos três métodos utilizados, quanto maior é a precisão e estabilidade do algoritmo utilizado, decorrentes do tipo de fatoração efetuada, maior é o custo computacional necessário. Isto é direto de ser percebido considerando que o algoritmo da fatoração de Cholesky tem custo de menor ordem que da fatoração QR (lembrando que $m > n$); ademais a fatoração QR por transformações de Householder torna-se mais cara que a fatoração QR via Gram-Schmidt conforme se trabalha com problemas mais complexos pois $nm \gg \frac{n^3}{3}$ a medida que $m \gg n$.

Entretanto para m não muito maior que n , caso do problema abordado neste trabalho, a resolução das equações normais via fatoração QR por transformações de Householder é a alternativa mais estável e precisa, embora o processo de Gram-Schmidt seja levemente mais caro computacionalmente.

2.3 Parte II: Algo Mais

2.3.1 Introdução

Dada a atual situação que passamos, considerando o contexto de pandemia pela COVID-19, optamos por buscar aplicações em modelos epidemiológicos que focassem principalmente no número acumulado de contaminados no decorrer dos dias.

Para tal, obtivemos os números de contaminados nos 31 primeiros dias da pandemia no Brasil, ou seja, do dia 26 de fevereiro de 2020 ao dia 27 de março de 2020, com esses dados, nos foi possível obter uma curva via quadrados mínimos que comparamos com a curva real. Fizemos também uma comparação com um período posterior a 27 de março de 2020; tal comparação visa constatar a efetividade do ajuste para a previsão de tendências a curto prazo; a longo prazo a modelagem necessária é extremamente mais complexa dado que se faz necessário a inclusão de fatores como a adesão ou não a quarentena, lockdown em alguns casos, princípios de distanciamento social, utilização de máscaras, taxa de testagem por milhão, entre outros.

2.3.2 Método Utilizado

Dada a análise feita anteriormente avaliando os três métodos propostos, decidimos por utilizar o método (c), fatoração QR completa usando transformações de Householder, tal método se mostrou o mais viável para a aplicação que propomos, considerando a precisão e estabilidade que o mesmo proporciona.

Os dados utilizados foram obtidos principalmente no site do ministério da saúde (Brasil, Ministério da Saúde [2020]) dedicado aos números referentes a pandemia da COVID-19 (antes do governo censurar grande parte dos dados no dia 06/06/2020).

Com esses dados em mãos, fizemos pequenas alterações no algoritmo para a montagem da matriz A , a matriz B foi montada manualmente. Com isso, chegamos em uma matriz A , montada utilizando-se os dias, com dimensões 31×11 , ou seja, um sistema composto de 31 equações com potências de x indo de 0 a 10 (mesma aproximação utilizada na primeira parte) e uma matriz B , montada utilizando-se o número de contaminados acumulados em dado dia, com dimensões 31×1 . Com isso, obtivemos o seguinte resultado.

| | |
|----------------|-------------|
| $cond_{A^t A}$ | 2.39514e+30 |
| $cond_{R_c}$ | 1.7637e+16 |

Tabela 3: Tabela dos Números de Condição

| | |
|----------|------------------|
| c_0 | 2.7820505948067 |
| c_1 | 1.4799449386002 |
| c_2 | -0.5823807993020 |
| c_3 | -1.1578832093610 |
| c_4 | 0.6510439858730 |
| c_5 | -0.1388841575011 |
| c_6 | 0.0154163473504 |
| c_7 | -0.0009704007333 |
| c_8 | 0.0000348328073 |
| c_9 | -0.0000006627346 |
| c_{10} | 0.0000000051772 |

Tabela 4: Tabela dos coeficientes da equação da parte 2

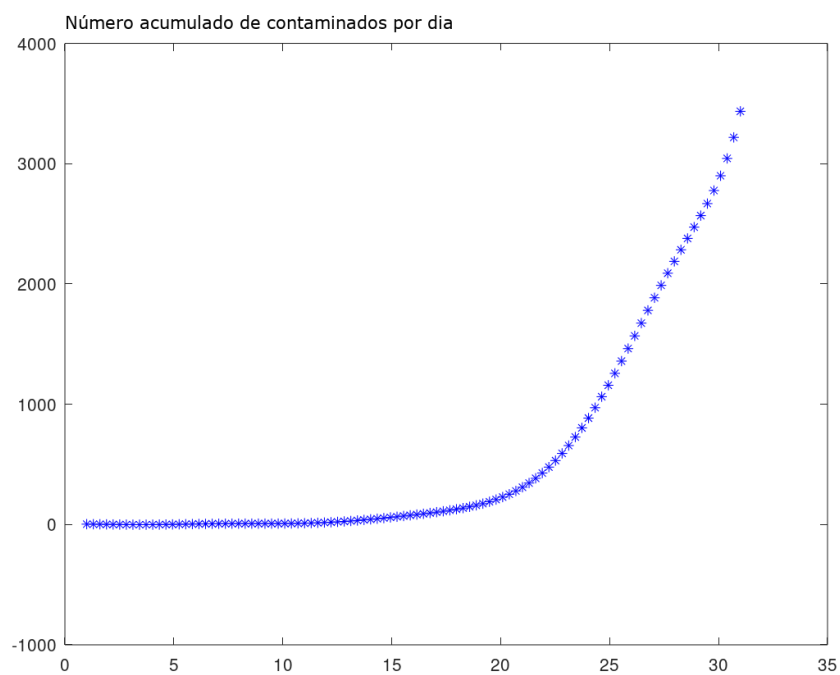


Figura 8: Gráfico obtido via QM para 31 dias

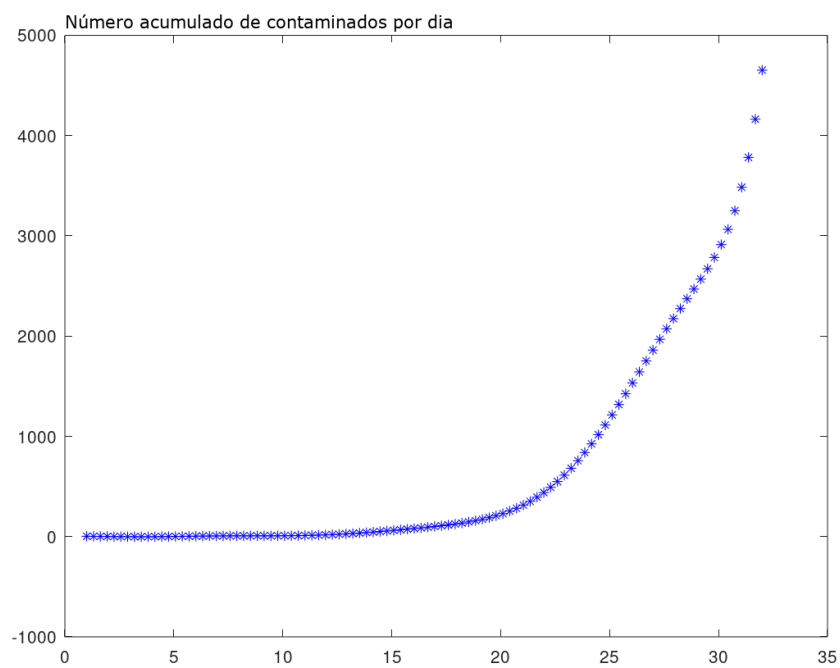


Figura 9: Gráfico obtido via QM para 32 dias

2.3.3 Análise e Resultados

Ao compararmos o gráfico obtido via QM e o gráfico oficial do ministério da saúde (Figura 10), observamos logo de início um grande diferencial entre os dois, a curvatura muito mais acentuada no primeiro em relação ao segundo, a explicação para isto possui inúmeros fatores relevantes, entre eles, podemos citar a adesão a condutas de distanciamento social e quarentena por boa parte das cidades brasileiras (Nussbaumer-Streit u. a. [2020]), fator que em um geral demonstra um efeito de achatamento na curva pelo menos 15 dias depois de iniciada a adesão (o tempo de incubação geral do vírus gira em torno de 8-14 dias) e a ausência de testagem em massa, o foco em testes de casos graves, o que considerando a alta porcentagem de assintomáticos e pessoas com sintomas fracos gera uma subnotificação enorme (Temos de ressaltar aqui o fato do Brasil ser um dos países mais afetados que tem menor testagem por milhão de habitantes) (Gaete [2020]) e o funcionamento parcial de secretarias da saúde de alguns Estados em finais de semana e segundas-feiras, o que gera uma queda brusca no número de contaminados diários nesses dias.

Como já citado na introdução, um modelo para previsão a longo prazo necessitaria de uma construção muito mais robusta, levando em consideração inúmeros fatores já citados, porém, para uma estimativa a curto prazo (entre 1 e 3 dias), o método por QM pode se mostrar válido, principalmente considerando um espaço de tempo, e por consequência, dados, maior sendo avaliado (com um espaço de tempo maior, se perceberia nos próprios dados os fatores de "achatamento" aqui constatados), o valor que obtivemos para o dia 32 via QM por exemplo, foi de 4688, enquanto que o número oficial foi 3992, considerando os pontos já levantadas anteriormente, consideramos que é um erro aceitável para um modelo simplificado que por questões técnicas não consegue considerar o efeito destes pontos. Podemos ressaltar também que a curvatura acentuada nos 31 primeiros dias nos mostra um caminho que seria tenebroso se tivesse se mantido um cenário de abertura completa, dada a tendência caótica de proliferação (Manchein [2020]), o número de casos explodiria em pouquíssimo tempo e teríamos um colapso da saúde extremamente rápido e com uma recuperação para um estado padrão lenta, o que ocasionaria um número de óbitos extremamente alto pela falta de atendimento de emergência.

2.3.4 Conclusão

Assim, considerando as devidas restrições impostas pela complexidade da realidade, consideramos os resultados obtidos pelo modelo satisfatórios, sendo uma ferramenta interessante para a previsão de números a um curto prazo utilizando-se de dados de dias anteriores, gostaríamos de ressaltar também a possibilidade de utilização para a análise de tendências, principalmente para a possibilidade de se prever o caminho da curva, ou seja, se a mesma está em uma fase de crescimento, platô ou decrescimento; considerando a discussão de reabertura ou não, tal análise é de extrema importância.

3 Conclusão

Na primeira parte deste trabalho foi possível perceber o desempenho diferenciado dos três métodos utilizados para a resolução do sistema de equações normais associado ao problema de quadrados mínimos proposto. Constatou-se que a fatoração QR representa um avanço substancial em termos de precisão e estabilidade numérica para a resolução de tal sistema em relação à abordagem dada pela fatoração de Cholesky, ressaltando o fato de que a análise de condicionamento de matrizes representa uma ferramenta poderosa para computações matemáticas eficientes. Ademais percebeu-se que o processo pelo qual uma fatoração QR é realizada também é de grande relevância quando estudadas as performances dos algoritmos de decomposição QR via processo de Gram-Schmidt e através de transformações de Householder. Embora nenhuma diferença em termos de precisão numérica foi detectada entre os resultados obtidos por estes dois processos, a implementação do tipo Householder apresentou maior estabilidade, evidenciada pelo menor número de condição da matriz R obtida. Em termos computacionais o uso da fatoração de Cholesky levou à um menor custo computacional para a resolução objetivada do que a utilização da fatoração QR, sendo que a qualidade dos resultados obtidos via Cholesky são intensamente inferiores aos obtidos via fatoração QR. Já a diferença entre os esforços computacionais necessários para os algoritmos desenvolvidos baseados na fatoração QR é insignificante devido ao tamanho e proporção do problema tratado (valores de m, n e proporção entre estas dimensões).

Já na segunda parte deste projeto aplicou-se a técnica de confirmado melhor desempenho para a modelagem matemática de um problema atual complexo. Embora as ferramentas utilizadas podem ser consideradas simples em termos matemáticos, é agradável perceber que os resultados e estimativas alcançadas foram bons considerando-se os inúmeros parâmetros envolvidos. O tratamento da situação problema escolhido revelou-se como sendo um meio confiável para a efetuação de previsões de curto prazo.

As principais adversidades que surgiram no decorrer deste projeto foram a dificuldade de obter informações precisas de implementações realizadas pelo MATLAB, o que deve como consequência a migração do desenvolvimento de algoritmos para OCATVE, e de coletar dados sobre a COVID-19, principalmente devido à censura efetuada pelo governo brasileiro. Felizmente não foram encontrados impasses nas implementações propriamente ditas, já que são referentes à procedimentos recorrentemente vistos ao longo da graduação!

Por fim é possível notar uma leve e ignorável distinção nos comentários dos scripts do anexo e dos arquivos enviados, pois o pacote de algoritmos usado na escrita do texto em \LaTeX não aceita acentos e cedilha, então os substituímos nos arquivos do apêndice pelas letras sem acento.

4 Apêndice

4.1 Script que gera os dados do problema

```
1 %Algoritmo que gera a matriz dos coeficientes (A) e o vetor b do sistema
2 %linear
3 function [A,b] = Ab()
4
5
6 %Preparando a matriz A
7 x = 0:0.05:1;
8 A = zeros(21,11);
9 for i = 1:21
10     for j = 1:11
11         if j == 1
12             A(i,j) = 1;
13         else
14             A(i,j) = x(i)^(j-1);
15         end
16     end
17 end
18
19 %Preparando o vetor b
20 b = zeros(21,1);
21 for i = 1:21
22     b(i) = exp(sin(6*x(i)));
23 end
24 end
```

4.2 Script que computa a solução para o método (a)

```
1 %Algoritmo que computa a fatoracao de Cholesky e resolve o sistema de
2 %equacoes normais. Como A (mxn) = (21x11), o fator de Cholesky R_a triang.
3 %sup. tem dimensao (nxn) = (11x11)
4 function [x_a, R_a] = questao_1_a(A,b)
5 %Aplicando a fatoracao de Cholesky no sistema normal, temos:
6 %A^t*A*x = A^t*b ; A^t*A = R_a^t*R_a --> R_a^t*R_a*x = A^t*b
7
8 %Calcula o fator de Cholesky de A^t*A
9 R_a = chol(A'*A);
10
11 %Reescrevendo, tenho R_a^t*R_a*x = A^t*b
12 %Faco y = R_a*x
13 %Resolvo |R_a^t*y = A^t*b
14 %      |R_a*x = y
15
16 %Utilizando o comando "barra invertida", o OCTAVE reconhece a estrutura da
17 %matriz R_a e R_a^t (triang. sup. e inf. respectivamente) e se aproveita dela
18 %na resolucao do sistema.
19
20 y = R_a'\A'*b;
21 x_a = R_a \ y;
22 end
```


4.3 Script que computa a solução para o método (b)

```
1 %Algoritmo que computa a fatoracao QR condensada pelo processo de
2 %Gram-Schmidt e resolve o sistema de equacoes normais. A = Q1*R_b, onde
3 %Q1(mxn) e R_b(nxn) triang. sup.
4 %Como A(21x11), Q1 tem dimensao (21x11) e R_b(11x11) (fator de Cholesky)
5 %triang. sup.
6
7 %Com o objetivo de nao fazer alteracoes na matriz A, neste codigo, vamos
8 %criar uma matriz auxiliar V=A, pagando o preco pelo espaco ocupado na
9 %memoria (como no algoritmo feito em aula).
10
11 function [x_b, R_b] = questao_1_b(A,b)
12 %Indexando o numero de colunas da matriz A
13 [m,n]=size(A);
14
15 %Criando as matrizes R_b e Q1 com zeros, para que o OCTAVE nao tenha que
16 %alterar a dimensao das matrizes a cada iteracao; aumentando o custo
17 %computacional.
18
19 R_b = zeros(n,n);
20 Q1 = zeros(m,n);
21
22 %Criando a matriz auxiliar
23 V = A;
24
25 %Comecando a fatoracao
26 for j = 1:n
27     for i = 1:j-1
28         R_b(i,j) = Q1(:,i)'*V(:,j);
29         V(:,j) = V(:,j) - R_b(i,j)*Q1(:,i);
30     end
31     R_b(j,j) = norm(V(:,j));
32     if R_b(j,j) == 0
33         disp('Conjunto LD')
34         break
35     else
36         Q1(:,j) = V(:,j)/R_b(j,j);
37     end
38 end
39
40 %Encerrada a fatoracao, resolveremos o sistema de equacoes normais.
41 %Com A^t*A*x = A^t*b (1), teremos da fatoracao A = Q1*R_b (2).
42 %Substituindo (2) em (1) e fazendo as devidas manipulacoes (aqui, o Jean
43 %faz as manipulacoes no trab) chegamos em R_b*x = Q1'*b
44
45 %Utilizando o comando "barra invertida", o OCTAVE reconhece a estrutura da
46 %matriz R_b (triang. sup0 e se aproveita dela na resolucao do sistema.
47
48
49 x_b = R_b\Q1'*b;
50 end
```

4.4 Script que computa a solução para o método (c)

```
1  %Algoritmo que computa a fatoracao QR completa com comandos nativos do
2  %OCTAVE, os quais implementam as transformacoes de Householder no calculo,
3  %e resolve o sistema de equacoes normais.
4  %%Como A(21x11), Q tem dimensao (21x21) e R_c(21x11).
5
6
7  function [x_c, R_c] = questao_1_c(A,b)
8  %Efetuando a fatoracao
9  [Q,R_c] = qr(A);
10
11 %Encerrada a fatoracao, resolveremos o sistema de equacoes normais.
12 %Com A^t*AX = A^t*b (1), teremos da fatoracao A = Q*R_c (2).
13 %Substituindo (2) em (1) e fazendo as devidas manipulacoes (aqui, o Jean
14 %faz as manipulacoes no trab) chegamos em R_c*x = Q'*b
15
16 %Utilizando o comando "barra invertida", o OCTAVE reconhece a estrutura da
17 %matriz R_c (triang. sup) e se aproveita dela na resolucao do sistema.
18
19 x_c = R_c\Q'*b;
20 end
```

4.5 Script que computa o número de condição das matrizes necessárias

```
1 %Calcula o numero de condicao das matrizes A^t*A, R_a, R_b e R_c.
2 %O uso da funcao nativa cond e justificada pois o Octave utiliza da SVD
3 %para computar o numero de condicao K2. Que se da pelo maior valor singular
4 %dividido pelo menor.
5 function [cond_AtA, cond_R_a, cond_R_b, cond_R_c] = n_cond(A, R_a, R_b, R_c)
6     cond_AtA = cond(A'*A);
7     cond_R_a = cond(R_a);
8     cond_R_b = cond(R_b);
9     cond_R_c = cond(R_c);
10 end
```

4.6 Script que monta a tabela dos coeficientes e plota o gráfico

```
1  %Algoritmo que invoca todas as funcoes e plota os graficos
2
3  %Invoca a funcao que prepara a matriz A e o vetor b
4  [A,b]=Ab();
5
6  %Invoca as funcoes que resolvem o sistema normal das 3 diferentes formas
7  [x_a, R_a]=questao_1_a(A,b);
8  [x_b, R_b]=questao_1_b(A,b);
9  [x_c, R_c]=questao_1_c(A,b);
10
11 %Invoca a funcao que computa o numero de condicao das matrizes A^t*A, R_a,
12 %R_b e R_c
13 [cond_AtA,cond_R_a,cond_R_b,cond_R_c] = n_cond(A,R_a,R_b,R_c);
14
15 %Criando a tabela com os numeros de condicao e com os coeficientes
16 T_c = [cond_AtA, cond_R_a, cond_R_b, cond_R_c]
17
18 coeficientes_do_metodo_a = x_a;
19 coeficientes_do_metodo_b = x_b;
20 coeficientes_do_metodo_c = x_c;
21 T_x = [coeficientes_do_metodo_a, coeficientes_do_metodo_b, coeficientes_do_metodo_c];
22
23 %plotando os graficos dos tres polinomios:
24
25 %criando os intervalos do plot
26 e = linspace(0,1);
27
28 %calculando os pontos de cada polinomio
29 %tomando cuidado pra inverter o vetor x_ e se encaixar
30 %corretamente na funcao "polyval"
31
32 y_a = polyval(flipud(x_a),e);
33 y_b = polyval(flipud(x_b),e);
34 y_c = polyval(flipud(x_c),e);
35
36 %calcula a funcao
37 f = exp(sin(6*e));
38
39 %Plota o grafico.
40 %No grafico, a curva em azul claro representa a funcao original,
41 %os pontos em verde representam o polinomio gerado com Cholesky
42 %os pontos em amarelo representam o polinomio gerado por Gram-Schmidt
43 %os pontos em azul escuro representam o polinomio gerado por Householder
44
45 %Nesta escala, parecem sobrepostos, mas com bastante zoom, da para ver que
46 %nao.
47 plot(e,y_a,'g*',e,y_b,'y*',e,y_c,'b*',e,f)
48
49 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%FIM%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

4.7 Script adaptado para a montagem da matriz A da parte 2

```
1 %Algoritmo que gera a matriz dos coeficientes (A) do problema pesquisado
2 x = 1:1:31;
3 A = zeros(31,11);
4 for i = 1:31
5     for j = 1:11
6         if j == 1
7             A(i,j) = 1;
8         else
9             A(i,j) = x(i)^(j-1);
10        end
11    end
12 end
```

4.8 Tabela de casos acumulados de COVID-19 por dia

Tabela construída com base nos dados de Taiti [2020], Mamoon und Rasskin [2020] e Brasil, Ministério da Saúde [2020].

| Dias passados | Números de casos acumulados |
|---------------|-----------------------------|
| 1 | 1 |
| 2 | 1 |
| 3 | 1 |
| 4 | 2 |
| 5 | 2 |
| 6 | 2 |
| 7 | 2 |
| 8 | 3 |
| 9 | 7 |
| 10 | 13 |
| 11 | 19 |
| 12 | 25 |
| 13 | 25 |
| 14 | 34 |
| 15 | 52 |
| 16 | 77 |
| 17 | 98 |
| 18 | 121 |
| 19 | 200 |
| 20 | 234 |
| 21 | 291 |
| 22 | 428 |
| 23 | 621 |
| 24 | 904 |
| 25 | 1128 |
| 26 | 1546 |
| 27 | 1891 |
| 28 | 2201 |
| 29 | 2433 |
| 30 | 2915 |
| 31 | 3417 |

Tabela 5: Número de casos acumulados por dia

4.9 Vetor obtido na Parte 2

$$x_c = \begin{bmatrix} 2.7820505948067 \\ 1.4799449386002 \\ -0.5823807993020 \\ -1.1578832093610 \\ 0.6510439858730 \\ -0.1388841575011 \\ 0.0154163473504 \\ -0.0009704007333 \\ 0.0000348328073 \\ -0.0000006627346 \\ 0.0000000051772 \end{bmatrix}$$

De cima para baixo, da menor potência de x (x^0) para a maior (x^{10})

4.10 Gráfico oficial do ministério da saúde

Obs: Obtido antes do governo censurar boa parte dos dados sobre a COVID-19 no Brasil, incluindo este gráfico.

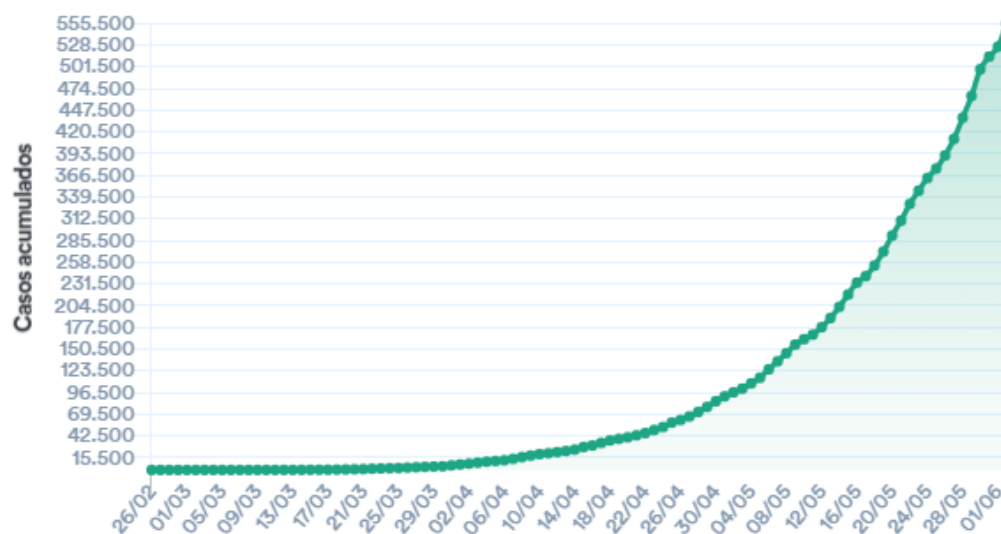


Figura 10: Gráfico Oficial Ministério da Saúde

Referências Bibliográficas

- [Brasil, Ministério da Saúde 2020] BRASIL, MINISTÉRIO DA SAÚDE: *Painel Coronavírus*. 2020. – URL <https://covid.saude.gov.br/>
- [Gaete 2020] GAETE, Rodrigo: *Análise Subnotificação*. 2020. – URL <https://ciis.fmrp.usp.br/covid19/analise-subnotificacao/>
- [Mamoon und Rasskin 2020] MAMOON, Navid ; RASSKIN, Gabriel: *COVID-19 Visualizer*. 2020. – URL <https://www.covidvisualizer.com/>
- [Manchein 2020] MANCHEIN, Cesar: *Teoria do Caos e a Pandemia*. 2020. – URL <https://revistagalileu.globo.com/Ciencia/noticia/2020/04/teoria-do-caos-explica-importancia-da-quarentena-na-pandemia-de-covid-19.html>
- [Nussbaumer-Streit u. a. 2020] NUSSBAUMER-STREIT ; MAYR ; ALULIA, Dobrescu ; CHAPMAN ; KLERINGS, Persad ; WAGNER ; SIEBERT ; CHRISTOF ; ZACHARIAH ; GARTLEHNER: *A quarentena isolada ou em combinação com outras medidas de saúde pública controla a COVID-2019?* 2020. – URL https://www.cochrane.org/pt/CD013574/INFECTN_quarentena-isolada-ou-em-combinacao-com-outras-medidas-de-saude-publica-controla-covid-2019
- [Shores 2007] SHORES, Thomas S.: *Applied Linear Algebra and Matrix Analysis*. 1. ed. New York : Springer, 2007. – ISBN 978-0-387-48947-6
- [Skoufranis 2012] SKOUFRANIS, Paul: *MATH 115A - Determinant Of the Vandermonde Matrix*. 2012. – URL <http://math.uga.edu/~pete/Skoufranis12.pdf>
- [Taiti 2020] TAITI, Adam: *Covid-19 Accelerometer*. 2020. – URL <https://theguarani.com.br/covid-19/>
- [Watkins 2010] WATKINS, David S.: *Fundamentals of Matrix Computations*. 3. ed. New Jersey : John Wiley & Sons, 2010. – ISBN 978-0-470-52833-4