Relatório do Lab 5 de CCI-22

Trabalho 5 – Mínimos Quadrados

Aluno:

Bruno Costa Alves Freire

Turma:

T 21.4

Professor:

Luiz Gustavo Bizarro Mirisola

Data:

23/05/2018

Instituto Tecnológico de Aeronáutica – ITA
Departamento de Computação

1. Análise: Complexidade de uma função desconhecida

Medindo-se o tempo de execução da função coringa `fmagicexp` para diferentes tamanhos de entrada, foi construído o gráfico da Figura 1. Para construir o gráfico, foram considerados os tamanhos de entrada variando de 300 a 1700, pois é o trecho em que o comportamento exponencial começa a se pronunciar de maneira relevante, e, naturalmente, a escolha da quantidade de ponto levou em conta o tempo de execução total das medições.

Para cada tamanho de entrada foram realizadas 10 medidas e tomou-se o valor médio entre elas, visando atenuar o ruído das medições.

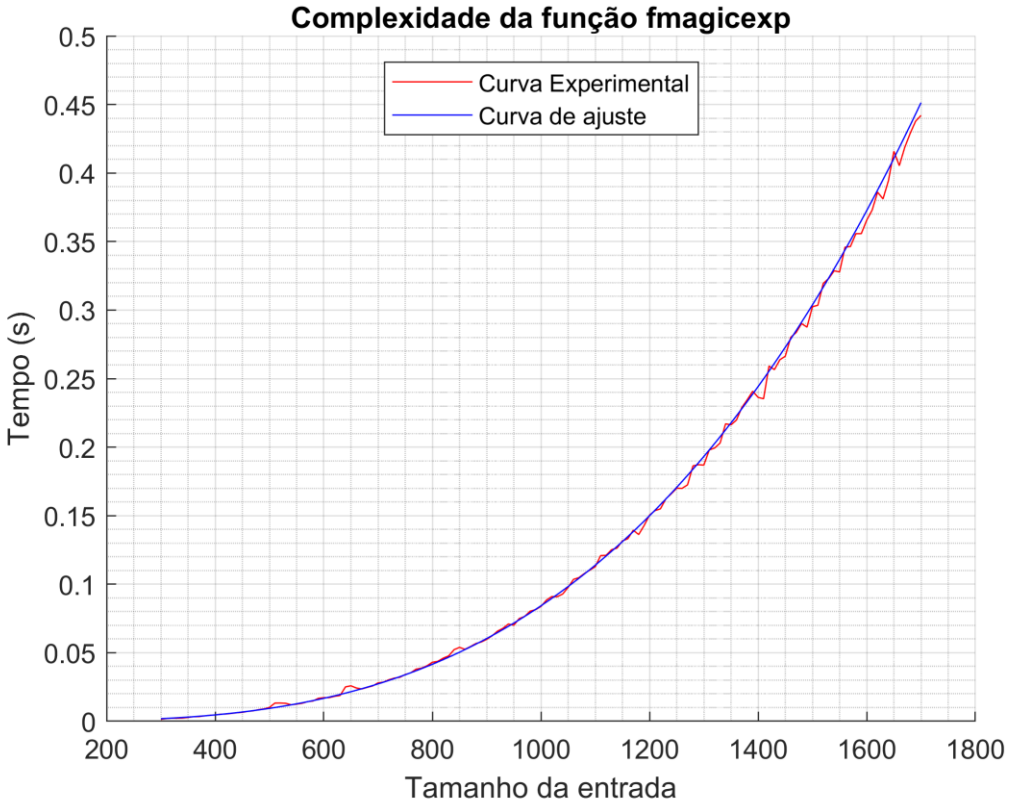

Figura 1: Medidas experimentais do tempo de execução

Com base nos dados medidos, foi realizado o ajuste da curva pelo método dos Mínimos Quadrados, por meio do script `ComplexityFinder.m`. A curva de ajuste resultando foi inserida no mesmo gráfico juntamente com a curva experimental. Pode-se verificar a coerência entre as duas.

Para realizar o ajuste da curva, foi assumido que a complexidade se comporta aproximadamente conforme a função $f(x) = \beta \cdot x^{\alpha}$, com α não inteiro. O procedimento para a determinação das constantes α e β foi de aplicar o logaritmo natural sobre os dados, e então aplicar o método dos mínimos quadrados com uma função do tipo $f^*(x) = c_0 + c_1 \cdot ln(x)$, encontrando assim os coeficientes c_0 e c_1, que correspondem respectivamente a $ln(\beta)$ e α.

Os valores encontrados foram:

- α = 3.1635
- β = 2,7215·10^{-11}

Uma vez encontrado o valor de α, foi plotado o gráfico da Figura 2, o qual apresenta a curva dos tempos medidos, elevados a 1/α, de forma que se deveria esperar algo próximo de uma reta, a menos de eventuais ruídos.

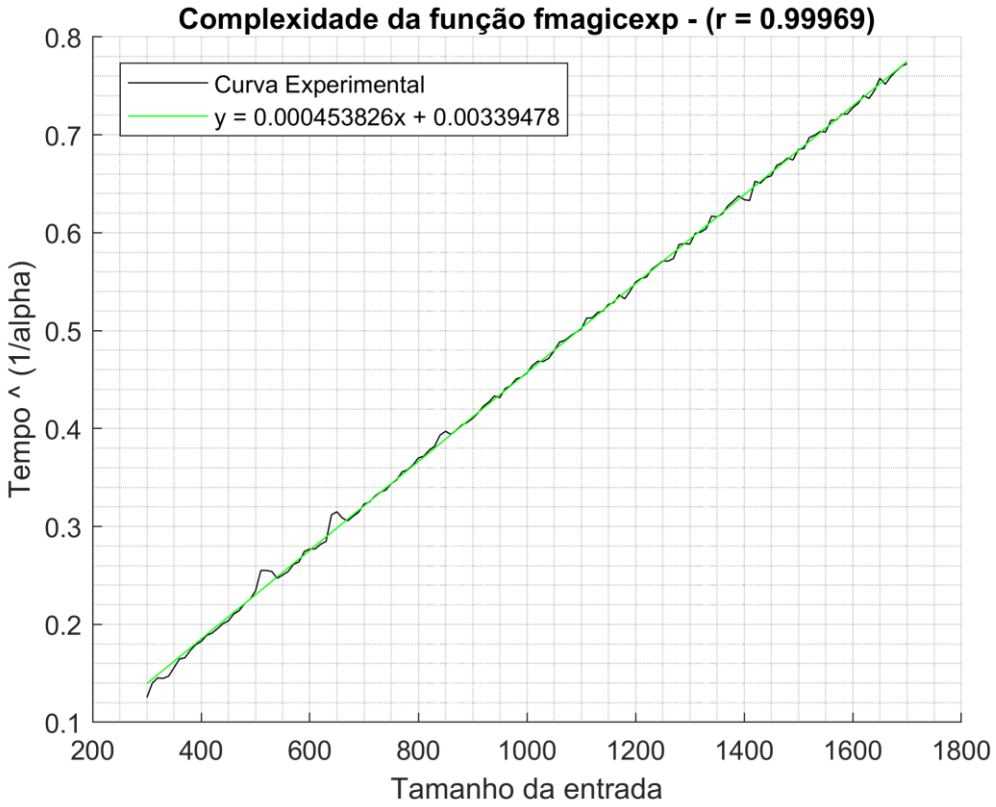

Figura 2: Gráfico dos tempos linearizados

Juntamente com a curva dos dados linearizados, foi inserida a reta de ajuste, obtida por meio da Regressão Linear destes dados, e cuja equação é apresentada na forma *y* = *ax* + *b* no próprio gráfico. Foi calculado também o valor do coeficiente de correlação, *r*, também incluído no gráfico.

- Equação da reta de ajuste: *y* = 4,53826·10^{-4}·x + 3,39478·10^{-3};
- Coeficiente de correlação: *r* = 0,99969.

2. Análise: Trilateração

2.1. Derivando as equações normais

Para o problema da trilateração, teremos um sistema de *n* equações quadráticas com duas incógnitas, sendo este um sistema bastante sobredeterminado. Podemos utilizar a n-ésima equação deste sistema para linearizar as demais, obtendo o seguinte:

$$A\,u = -2 \cdot \begin{bmatrix} (x_1 - x_n) & (y_1 - y_n) \\ (x_2 - x_n) & (y_2 - y_n) \\ \vdots & \vdots \\ (x_{n-1} - x_n) & (y_{n-1} - y_n) \end{bmatrix} \begin{bmatrix} u_x \\ u_y \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{n-1} \end{bmatrix} + B^{\perp} = b + B^{\perp}$$

Resolver este problema consiste em encontrar o vetor *u* que minimiza a norma do resíduo $B^{\perp} = Au - b$. Em termos práticos, minimizar a norma de uma vetor equivale a minimizar o quadrado da mesma, que é denotado por *R(u)*. Calculando a norma quadrada do resíduo e igualando seu gradiente a zero, chegaremos a um sistema de duas equações e duas incógnitas. Contudo, uma outra abordagem, mais algébrica permite chegar ao mesmo resultado por outro caminho.

Essa equação indica que o vetor *u* é transformado pela matriz A na soma do vetor *b* com um resíduo ortogonal à imagem de A. Tendo isto em mente, e lembrando-se do teorema do núcleo e da imagem, sabemos que a imagem de A é o complemento ortogonal do núcleo da transposta de A. Logo, ao multiplicar pela esquerda por A^T, cancelamos a parcela residual e obtemos a equação:

$$\frac{A^T A\,u}{4} = \begin{bmatrix} (x_1 - x_n) & (x_2 - x_n) & \cdots & (x_{n-1} - x_n) \\ (y_1 - y_n) & (y_2 - y_n) & \cdots & (y_{n-1} - y_n) \end{bmatrix} \begin{bmatrix} (x_1 - x_n) & (y_1 - y_n) \\ (x_2 - x_n) & (y_2 - y_n) \\ \vdots & \vdots \\ (x_{n-1} - x_n) & (y_{n-1} - y_n) \end{bmatrix} \begin{bmatrix} u_x \\ u_y \end{bmatrix}$$

$$= \frac{-2}{4} \begin{bmatrix} (x_1 - x_n) & (x_2 - x_n) & \cdots & (x_{n-1} - x_n) \\ (y_1 - y_n) & (y_2 - y_n) & \cdots & (y_{n-1} - y_n) \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{n-1} \end{bmatrix} + A^T B^{\perp} = A^T b$$

Que é conhecida como sistema normal, pois sua origem está no problema de minimizar a *norma* do resíduo. Explicitando os cálculos, temos:

$$2 \begin{bmatrix} \sum_{k=1}^{n-1} (x_k - x_n)^2 & \sum_{k=1}^{n-1} (x_k - x_n)(y_k - y_n) \\ \sum_{k=1}^{n-1} (x_k - x_n)(y_k - y_n) & \sum_{k=1}^{n-1} (y_k - y_n)^2 \end{bmatrix} \begin{bmatrix} u_x \\ u_y \end{bmatrix} = - \begin{bmatrix} \sum_{k=1}^{n-1} (x_k - x_n) b_k \\ \sum_{k=1}^{n-1} (y_k - y_n) b_k \end{bmatrix}$$

Agora podemos encontrar *u* simplesmente resolvendo o sistema linear.

2.2. Implementando as equações normais

A implementação da solução do problema da trilateração por meio das equações normais é realizada no script `calcAandBforTrilateration.m`.

2.3. Comparando os métodos de solução

O problema de mínimos quadrados pode ser reduzido à solução do sistema normal, ou seja, $\hat{A}x = \hat{b}$, no entanto, pode-se chegar a este por dois caminhos: um utilizando cálculo diferencial e extremização de uma função de várias variáveis, e outro utilizando álgebra linear, e a teoria de projeções ortogonais.

Queremos comparar a complexidade da implementação dos dois métodos, em relação ao número de incógnitas (coordenadas da posição a se determinar, outros parâmetros desconhecidos, etc.), que será denotado por *n*, e em relação ao número de *beacons*, que será indicado por *m*.

Em ambos os métodos, iniciaremos linearizando as equações de distância, obtendo um sistema linear *m*-1 por *n*, *Ax* = *b*. A partir daqui, os caminhos se separam. No método do cálculo, consideramos a função resíduo quadrático:

$$R(\boldsymbol{x}) \;=\; \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 = \sum_{k=1}^{m-1} (\boldsymbol{A}_{k,*}^T \boldsymbol{x} - b_k)^2$$

Sendo esta uma função de *n* variáveis, queremos extremizá-la, e para isso, impomos a condição de que seu gradiente seja o vetor nulo. Derivando *R* parcialmente em relação a cada variável x_k, temos:

$$\nabla R(\boldsymbol{x}) = 0 \implies \begin{bmatrix} \frac{\partial R}{\partial x_1} \\ \frac{\partial R}{\partial x_2} \\ \vdots \\ \frac{\partial R}{\partial x_j} \\ \vdots \\ \frac{\partial R}{\partial x_n} \end{bmatrix} = \begin{bmatrix} \sum_{k=1}^{m-1} 2A_{k1}(\boldsymbol{A}_{k,*}^T \boldsymbol{x} - b_k) \\ \sum_{k=1}^{m-1} 2A_{k2}(\boldsymbol{A}_{k,*}^T \boldsymbol{x} - b_k) \\ \vdots \\ \sum_{k=1}^{m-1} 2A_{kj}(\boldsymbol{A}_{k,*}^T \boldsymbol{x} - b_k) \\ \vdots \\ \sum_{k=1}^{m-1} 2A_{kn}(\boldsymbol{A}_{k,*}^T \boldsymbol{x} - b_k) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ \vdots \\ 0 \end{bmatrix} \implies$$

$$\begin{bmatrix} \sum_{k=1}^{m-1} A_{k1}\boldsymbol{A}_{k,*}^T \boldsymbol{x} \\ \sum_{k=1}^{m-1} A_{k2}\boldsymbol{A}_{k,*}^T \boldsymbol{x} \\ \vdots \\ \sum_{k=1}^{m-1} A_{kj}\boldsymbol{A}_{k,*}^T \boldsymbol{x} \\ \vdots \\ \sum_{k=1}^{m-1} A_{kn}\boldsymbol{A}_{k,*}^T \boldsymbol{x} \end{bmatrix} = \begin{bmatrix} \sum_{k=1}^{m-1} A_{k1} b_k \\ \sum_{k=1}^{m-1} A_{k2} b_k \\ \vdots \\ \sum_{k=1}^{m-1} A_{kj} b_k \\ \vdots \\ \sum_{k=1}^{m-1} A_{kn} b_k \end{bmatrix} \implies$$

$$\begin{bmatrix} \boldsymbol{A_{1,*}^T Ax} \\ \boldsymbol{A_{2,*}^T Ax} \\ \vdots \\ \boldsymbol{A_{j,*}^T Ax} \\ \vdots \\ \boldsymbol{A_{n,*}^T Ax} \end{bmatrix} = \begin{bmatrix} \boldsymbol{A_{1,*}^T b} \\ \boldsymbol{A_{2,*}^T b} \\ \vdots \\ \boldsymbol{A_{3,*}^T b} \\ \vdots \\ \boldsymbol{A_{4,*}^T b} \end{bmatrix} \Longrightarrow \boldsymbol{A^T Ax = A^T b}$$

Vemos então que o caminho do cálculo nos leva exatamente ao mesmo caminho da álgebra linear. Uma vez que se chega à equação acima, basta resolvê-lo por qualquer método, ou até mesmo invertendo a matriz A^TA. Isso mostra que *os dois métodos são analiticamente equivalentes*.

Quanto à complexidade, a derivação não conta nenhuma operação pois já conhecemos a função, então basta implementar o sistema. Cada coeficiente da matriz do lado esquerdo leva O(m-1), e como são n^2 elementos, a complexidade total será $O(mn^2)$. Para calcular o vetor do lado direito, cada elemento leva O(m-1), e são n elementos, logo, O(mn). Somando as duas coisas, temos $O(mn^2+mn) = O(mn^2)$. Por fim, devemos resolver o sistema linear, que em geral tem complexidade cúbica no número de incógnitas, sendo assim, fica: $O(n^3+mn^2)$.

Quanto ao caminho algébrico, não há nenhuma diferença em termos de contas, os procedimentos são os mesmos, apenas a justificação deles é diferente.

	Diferenciação das equações de resíduo	**Aplicação de $A^TAx = A^Tb$**
Complexidade	$O(n^3+mn^2)$	$O(n^3+mn^2)$