Projeto II: ajustamento

Danrley Tavares, Filipe Bessa, Ielane Nogueira , Jean de Oliveira, Ruben Esteche, Tertius Ferraz

Alunos da disciplina Cálculo Numérico, turma T8.

Resumo: *Este trabalho constitui um estudo sobre os efeitos do ajustamento de dados tabelados e da linearização de funções.*

Palavras-chave: ajustamento; linearização de funções; sistemas lineares.

1. Introdução

Ao se realizar experimentos, é comum haver valores tabelados para o fenômeno sendo estudado. Entretanto, nem sempre se conhece a relação existente entre as variáveis de seu experimento e os resultados observados – uma vez que, pelo fato de serem obtidos experimentalmente, não há certeza sobre a corretude dos mesmos. O método de ajustamento permite que seja obtida a função que melhor se aproxima dos dados levantados.

Este trabalho visa analisar os resultados obtidos em um programa elaborado em linguagem C capaz de identificar, dentre uma coleção de funções pré-determinadas, qual tipo de função representa melhor a relação entre dados x e $f(x)$.

2. Desenvolvimento

A. Ajustamento

Sejam x e $f(x)$ dados representados em de uma tabela, é possível deduzir o tipo de curva que melhor se ajusta (se aproxima) a eles. Este processo, chamado de ajustamento, é de grande importância em experimentos, pois é possível que se desconheça a função que relaciona x e $f(x)$. Além disso, através da função obtida, pode-se "prever" os resultados de experimentos para variáveis estejam fora do intervalo tabelado. De maneira geral, o ajustamento ($P(x)$) é representado como a combinação linear de funções "elementares" – como, por exemplo, 1, x, x^2, $\ln x$.

Dada uma coleção de funções elementares, determina-se qual P é o melhor ajustamento verificando qual delas tem o menor valor para o somatório dos quadrados de seus resíduos – que é definido como:

$$\sum_{i=0}^{n} R^2(x_i) = \sum_{i=0}^{n} (P(x_i) - f(x_i))^2$$

Escolhe-se o quadrado dos resíduos de modo a evitar problemas com sinais de R, cujo somatório poderia ser igual a zero e dar a falsa impressão de que $P(x)$ é uma ótima aproximação quando, na verdade, seus valores estão distantes de $f(x)$.

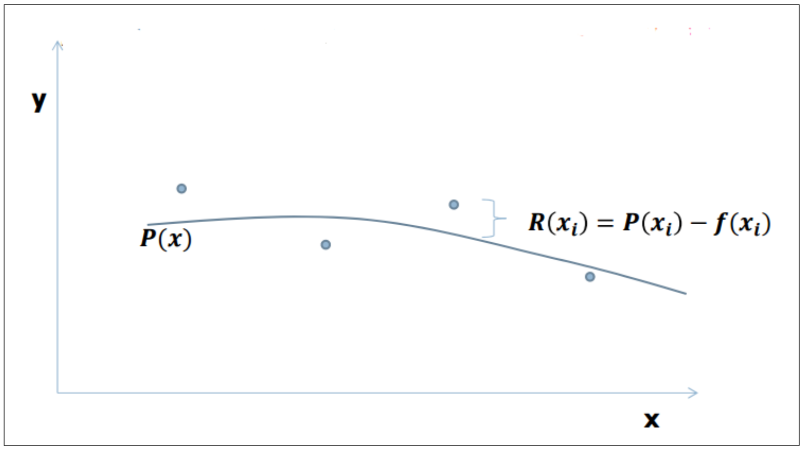

Figura 1 – Ideia geométrica do ajustamento e seu resíduo

B. Método dos Mínimos Quadrados (MMQ)

Um dos métodos para se ajustar dados tabelados é o dos mínimos quadrados. Este método consiste em, através da montagem de um sistema normal – sistema linear cujas equações tem solução zero -, encontrar os coeficientes de $P(x)$.

Seja $P(x)$ a combinação linear do tipo $P(x) = a_0 \times G_0(x) + a_1 \times G_1(x) + ... + a_m \times G_m(x)$ e uma função $f(x)$, tem-se que: $R(x_i) = a_0 \times g_0(x_i) + a_1 \times g_1(x_i) + ... + a_m \times g_m(x_i) - f(x_i)$. Após manipular algebricamente $\sum R^2(x_i)$ – calculando sua derivada em relação a a_j e igualando a zero – obtém-se o seguinte sistema linear

$$\sum_{k=0}^{m} a_k \sum_{i=0}^{n} G_k(x_i)G_j(x_i) = \sum_{i=0}^{n} f(x_i)G_j(x_i),$$

com a_k representando os coeficientes e $G_k(x_i)$ e $G_j(x_i)$ as funções elementares. Substituindo os valores da tabela nos somatórios, é possível resolver o sistema linear através de diversos métodos (como o de eliminação de Gauss, por exemplo) e descobrir os coeficientes do ajustamento $P(x)$.

É importante salientar que o MMQ só pode ser aplicado em casos lineares. Quando tivermos funções não-lineares – como em ae^{bx} – deve utilizar métodos para linear a função em questão e, só então, aplicar o MMQ.

C. Método de Eliminação de Gauss

Ao fim do MMQ, teremos um conjunto de n equações lineares com n incógnitas (descrito como o sistema Ax = b) para ser solucionado a fim de se obter os coeficientes de $P(x)$. Para solucionar esse sistema, pode-se utilizar diversos métodos, mas o escolhido neste estudo foi o método de eliminação de Gauss.

Este método consiste em transformar um sistema $Ax = b$ em um sistema $Tx = c$. Onde T é uma matriz triangular superior, e é escolhido pois sabe-se que as matrizes triangulares (aquelas onde todos os elementos abaixo da diagonal principal são iguais a zero) são mais simples de solucionar. Essa transformação é feita através de algumas operações básicas que não alteram a solução do sistema – como permutação de linhas, multiplicação de linha por número real diferente de zero e adição de uma linha a outra multiplicada por um número real diferente de zero –, além de utilizar o conceito de pivotação parcial.

A pivotação parcial é um processo no qual permuta-se, quando necessário, linhas da matriz de modo que o primeiro elemento diferente de zero (chamado de pivô) seja o maior, em módulo, dentro todos os pivôs da coluna.

Os processos acima são executados até que se obtenha um sistema triangular superior e seja possível obter a sua solução.

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \dots + a_{1n}x_n = b_1 \\ \qquad\quad a_{22}x_2 + a_{23}x_3 + \dots + a_{2n}x_n = b_2 \\ \qquad\qquad \cdot \qquad\qquad\qquad\qquad\qquad\qquad \cdot \\ \qquad\qquad\qquad \cdot \qquad\qquad\qquad\qquad\quad \cdot \\ \qquad\qquad\qquad\qquad \cdot \qquad\qquad\qquad \cdot \\ \qquad\qquad\qquad\qquad\qquad\qquad a_{nn}x_n = b_n \end{cases}$$

Figura 2 – Sistema linear triangular superior

D. Programa

Foi elaborado um programa em linguagem C capaz de receber valores para *x* e *f*(*x*) de um arquivo de texto e, a partir deles, encontrar qual função (dentre uma coleção pré-definida) se ajusta melhor aos dados lidos.

Assim que é iniciado, o programa executa a função *main* e exibe um menu perguntando se o usuário pretende digitar o nome de um arquivo de entrada ou fechar o programa. Essa função foi implementada com laços do tipo *while* e funções *scanf*. Para realizar os comandos do menu foram utilizadas as funções *if* e *scanf* (para identificar a escolha do usuário) e *fopen* (para abrir o arquivo “entrada.txt”).

Logo após, o programa recebe (através do comando *fscanf*) do arquivo o número de pares ordenados (*x*, *f*(*x*)) a serem lidos (*n*), aloca *n* espaços consecutivos em vetores do tipo *double* (com a função *malloc*), que serão modificados posteriormente por outras funções, e imprime o valor de *n*. Então, os valores dos pares ordenados são lidos no arquivo através de um laço *for* as variáveis px[0] a px[9] (responsáveis por armazenar os valores do resíduo de cada função da coleção), que chamam as funções do programa referentes a cada item da coleção de funções, tendo como parâmetro os valores lidos no arquivo.

Para que fosse implementado o método dos mínimos quadrados no programa, foram utilizadas as versões linearizadas de todas as funções pré-estabelecidas na coleção, de modo a se obter os termos a_0, a_1 (coeficientes) e G_0, G_1 (funções elementares). A lista de linearizações segue na tabela abaixo:

Funções	**Linearização**
$f_1(x) = ax + b$	$P(x) = ax + b$
$f_2(x) = ax^2 + bx$	$P(x) = ax^2 + bx$
$f_3(x) = ax^3 + bx$	$P(x) = ax^3 + bx$
$f_4(x) = ax^3 + bx^2$	$P(x) = ax^3 + bx^2$
$f_5(x) = ae^{bx}$	$P(x) = \ln(f(x)) = \ln(a)+ bx$
$f_6(x) = ax^b$	$P(x) = \ln(f(x)) = \ln(a)+ b\ln(x)$

$f_7(x) = a\ln(x) + {}^{b}/_{x}$	$P(x) = a \ln(x) + b\ {}^{1}/_{x}$
$f_8(x) = ax + {}^{b}/_{x}$	$P(x) = ax + {}^{b\ 1}/_{x}$
$f_9(x) = a\cos(x) + bx$	$P(x) = a \cos(x) + bx$
$f_{10}(x) = {}^{1}/_{a\mathrm{sen}(x)+be^{x}}$	$P(x) = ({}^{1}/_{f(x)}) = a\ \mathrm{sen}(x) + be^{x}$

Tabela 1 – Linearização das funções f_1 a f_{10}

Para cada função da coleção, foi criada uma função do programa tendo os parâmetros específicos de cada linearização. Por exemplo, no caso de "funcao1", os elementos dos vetores g0 e g1 recebem, através de um laço *for*, os valores de 1 e *x*, respectivamente; enquanto que em "funcao10", a g0 e g1 são atribuídos os valores de sen(x) e exp(x). Quando todos os elementos dos vetores são preenchidos, é chamada a função "tabela".

O objetivo de "tabela" é armar as matrizes do sistema linear com os coeficientes e resultados de $P(x)$ e solucioná-lo. Através de um laço *for*, duas matrizes ("matriz", uma 2x2, e "resultado", uma 2x1) são criadas, e com outro laço *for*, preenche suas posições com o somatório dos produtos de g0, g1 e num2 (variável referente ao valor de $f(x)$). Após isso, é criado um laço com *if*, onde é realizado o método de eliminação de Gauss, através de operações aritméticas, e, por fim, se obtém os valores de a_0 e a_1 (que são atribuídos a outros dois vetores após as conversões necessárias).

Em seguida, é a vez de calcular o resíduo da função. Através de um laço *for*, é calculado o quadrado da diferença $P(x) - f(x)$ e o somatório é armazenado em "px". O próximo passo é normalizar os valores dos vetores usando laços condicionais e imprimir a função com os coeficientes encontrados e o valor do somatório dos seus resíduos.

A função *main*, então, compara todos os valores de "px" com o uso de laços condicionais (*for* e *if*) para então imprimir a função com menor resíduo, fechar o arquivo (fclose) e encerrar o programa.

E. Exemplos

Sejam *a* e *b* constantes das equações, as funções que vão de $f1(x)$ até $f10(x)$ são os ajustamentos dos pontos escritos em forma de funções e os Ri(x), os resíduos associados a esses ajustamentos, respectivamente. Todos os ajustamentos e resíduos presentes nos exemplos a seguir foram obtidos com uso do programa elaborado para o projeto.

Exemplo 1: Temos dois pontos a serem avaliados para achar a curva que melhor se ajusta a todos esses pontos, sendo eles x_1=2.0000E+00; $f(x_1)$=9.9104E-01; x_2=3.0008E-01 e $f(x_2)$=8.7316E+00. Para esse caso, obtivemos o seguinte resultado:

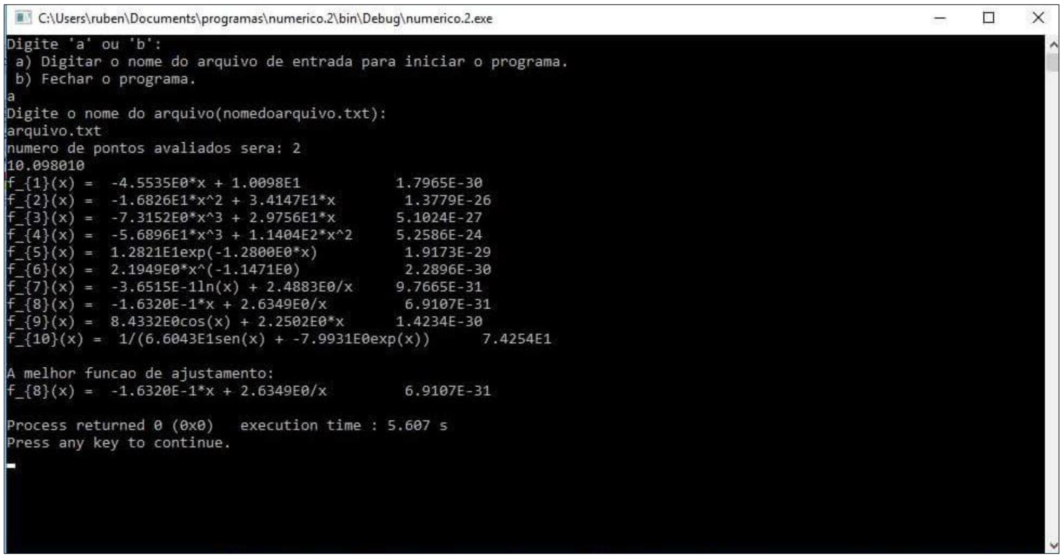

Figura 3 – Exemplo 1

Exemplo 2: Temos três pontos a serem avaliados para achar a curva que melhor se ajusta a todos esses pontos, sendo eles x_1=3.5578E+01, $f(x_1)$=4.5729E+01; x_2=2.6324E+00, $f(x_2)$=1.2000E-03; x_3=3.4738E-01, $f(x_3)$=8.7216E+00. Para esse caso, obtivemos o seguinte resultado:

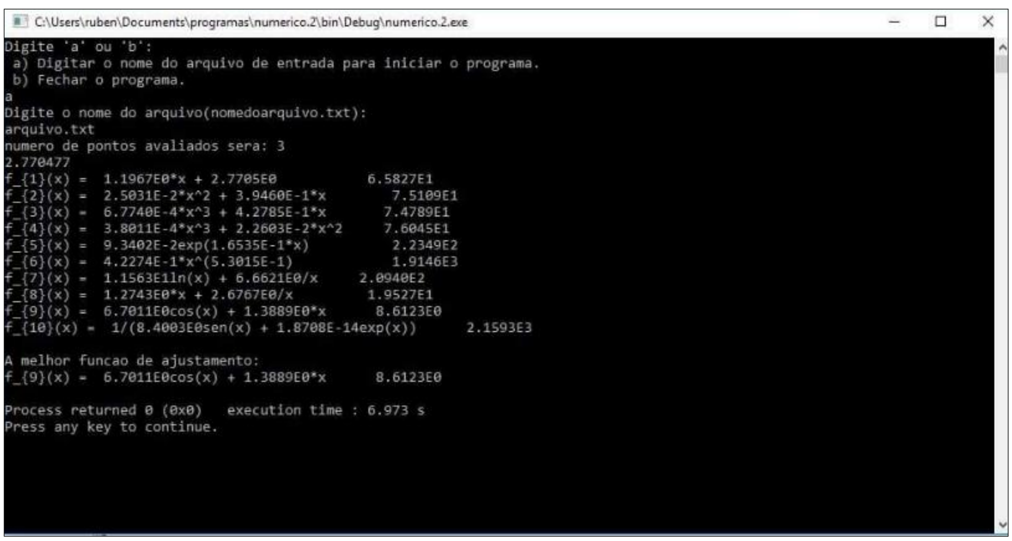

Figura 4 – Exemplo 2

Exemplo 3: Temos quatro pontos a serem avaliados para achar a curva que melhor se ajusta a eles, sendo x_1=1.0000E+00, $f(x_1)$=1.1059E+00; x_2=2.0000E+00, $f(x_2)$=9.9104E-01; x_3=3.1000E+00, $f(x_3)$=1.5515E-01; x_4=3.5000E+00; $f(x_4)$=8.6590E-02. Para esse caso, obtivemos o seguinte resultado:

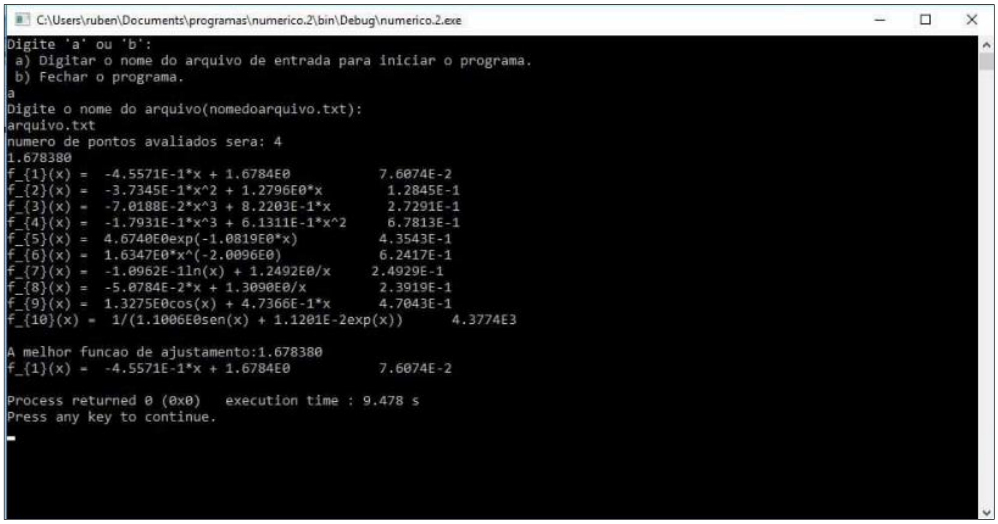

Figura 5 – Exemplo 3

3. Limitações

Dentre as limitações encontradas durante a elaboração do programa, uma delas foi a de identificar como as funções seriam linearizadas, embora tenha sido relativamente simples a sua implementação. Além disso, descobriu-se que não era possível solucionar satisfatoriamente o problema quando os valores escolhidos para x eram muito pequenos (com expoentes menores que -5) pois o número explodia no programa.

4. Conclusões

Através dos resultados obtidos, observou-se que o programa cumpre seu papel em encontrar a função que mais se adapta aos pontos dados. Entretanto, esta coleção pré definida pode ser insuficiente em alguns casos se levarmos em conta o alto valor dos resíduos gerados (o que nos levou a crer que a função que melhor descrevia o ponto não pertencia à coleção).

Referências

[l] AMORIM, Guilherme. Aula 13 – Ajustamento. Disponível em: <http://www.cin.ufpe.br/~if215/slides/2016-2/lecture13.pdf>.

[2] DOS SANTOS, José Dias; DA SILVA, Zanoni Carvalho. Métodos Numéricos. 3ª ed. Pernambuco: Editora Universitária UFPE, 2010.