```
__/\\\\\\\\\\\\\\\__/\\\\\\\\\\\\\_________________/\\\\\\\\\\___
 _\/\\\///////////__\/\\\/////////\\\_____________/\\\///////\\\_
  _\/\\\_____________\/\\\_______\/\\\____________\///______/\\\__
   _\/\\\\\\\\\\\_____\/\\\\\\\\\\\\\/____________________/\\\//___
    _\/\\\///////______\/\\\/////////_____________________\////\\\__
     _\/\\\_____________\/\\\_________________________________\//\\\_
      _\/\\\_____________\/\\\________________________/\\\______/\\\__
       _\/\\\\\\\\\\\\\\\_\/\\\________________________\///\\\\\\\\\/___
        _\///////////////__\///___________________________\/////////_____

                Métodos Numéricos em ALgebra Linear (MAC0300)
```


Por

Caio Vinícius Dadauto 7994808

22/11/2014

As implementações deste trabalho foram executadas em uma máquina com as seguites configurações:

OS	Arch Linux
Kernel	x86_64 Linux 4.1.5-1-ARCH
CPU	Intel Core i5-4200 CPU @ 2.6GHz
RAM	3862MiB

1 Implementação

A implementação para o problema de quadrados mínimos utilizando refletores foi feita basicamente seguindo a rigor os passos sugeridos pelo livro [1]. Grande parte das explicações concernentes a implementação estão indicadas em forma de comentario no proprio código. Para facilitar a compreensão da implementação, ao declarar as funções utilizadas durante o código foi indicado a qual parte do problema cada conjunto de função está relacionado. Os comentários se encontram sempre ao início de cada função e explica alguns detalhes da implementação e os objetivos de cada uma edstas funções.

Para poder testar o programa, foi feita duas abordagens do problema de quadrados mínimos. A primeira é uma aborgem mais prática que consiste na seguinte situação, um certo usuário toma um conjunto de dados contendo pontos (a, b) e a este conjunto quer ajustar um polinômio de grau $m-1$. Dessa forma o usuário fornece um arquivo texto contendo o tamanho do conjunto, o grau do polinômio e o conjunto de pontos (a, b). A partir deste arquivo texto o programa gera o sistema $Ax = b$, tomando cuidado na escolher da base de polinômios para gerar A bem condicionada. Esta escolha foi feita baseada em [2], o que é basicamente determinar os valores máximo e mínimo entre os valores de a e tomar a base como sendo potências da seguinte variável:

$$s = \frac{a - \left(\frac{\max+\min}{2}\right)}{\max - \left(\frac{\max+\min}{2}\right)} \tag{1}$$

A segunda aborgem está relacionada exclusivamente com os teste do programa. Esta abordagem consistem em fornecer ao programa um arquivo texto contendo explicitamente A e b do sistema $Ax = b$. A ídeia é fornecer A com posto defeituoso para constatar se o programa de fato encontra tal problema. Neste caso de posto defeituoso, o porblema de quadrados mínimos assume infinitas soluções. Assim, ao resolver o MMQ, optou-se por tomar y_2 como vetor nulo em [1]:

$$R_{11}y_1 = c - R_{12}y_2$$

esta escolha foi feita unicamente para fins práticos.

Ao término do programa é impresso na tela o posto de A e a solução x do porblema de quadrados mínimos. Caso o usuário optou por inserir o conjunto de pontos ao invés de A e b, o progrma imprime o polinômio que se ajusta aos pontos fornecidos. A impressão da solução x é feita somente se x possuir um tamanho "legível", no caso com tamanho maior ou igual a 20.

2 Exemplos

2.1 Constatação de Ajuste

Suponha que o usuário meça um conjunto de pontos e a este quer ajustar um polinômio de grau 2. Para exemplificar tal situação, tomemos o conjunto de pontos especificado no arquivo `exemplo1` como entrada do progrma. Assim, após resolver o problema de quadrados mínimos, obteve-se o seguinte ajuste ao conjunto de pontos:

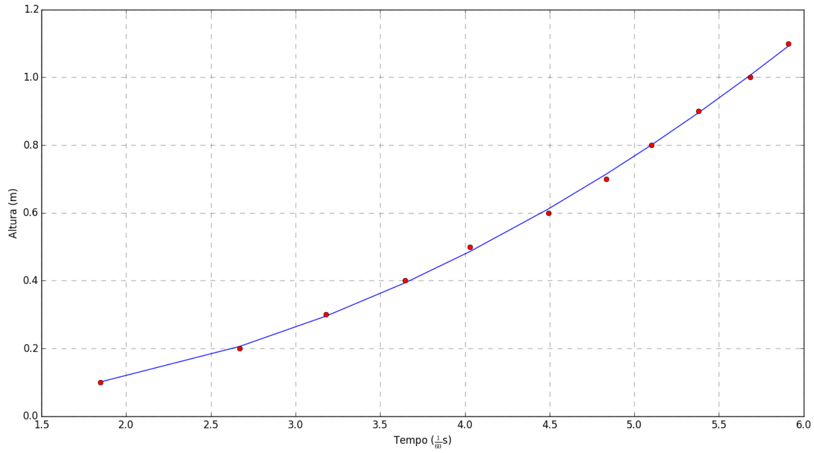

Figura 1: Ajuste obtido para o conjunto de dados fornecido pelo arquivo `exemplo1`.

2.2 Estimativa da Gravidade

Suponha um experimento em que o usuário quer determinar a gravidade local. Para isso ele decide abandonar um objeto com boa aerodinâmica a uma determinada autura do solo. Durante a queda ele anota a posição (em cm) com relação ao solo a cada $\frac{1}{60}s$, obtendo dessa forma um conjunto de pontos de posição contra o tempo. Assim, quer ajustar a estes dados a seguinte função:

$$x = x_0 + v_0 t + \frac{1}{2} g t^2 \tag{2}$$

e com o valor de ajuste determinar g. Como está função nada mais é que um polinômio de segundo grau, basta fornecer ao porgrama um arquivo ((exemplo2)) contendo esse conjunto de dados.

O coeficiente obtido para s^2 foi de 32.4073, a partir disso é possível determinar o coeficiente de t^2 através da expressão (1). Coeficiente, este, que vale 0.1349. Sabendo que a posição foi medida em centímetros, os instântes de tempo são da ordem de $\frac{1}{60}$s e, ainda, notando a constante $\frac{1}{2}$ na expressão (2), temos que:

$$0.1349 = \frac{1}{72} g$$

o que resulta em $g = 9.7128$, resultado que, por sua vez, é acurado. O ajuste obtido é apresentado na figura logo abaixo.

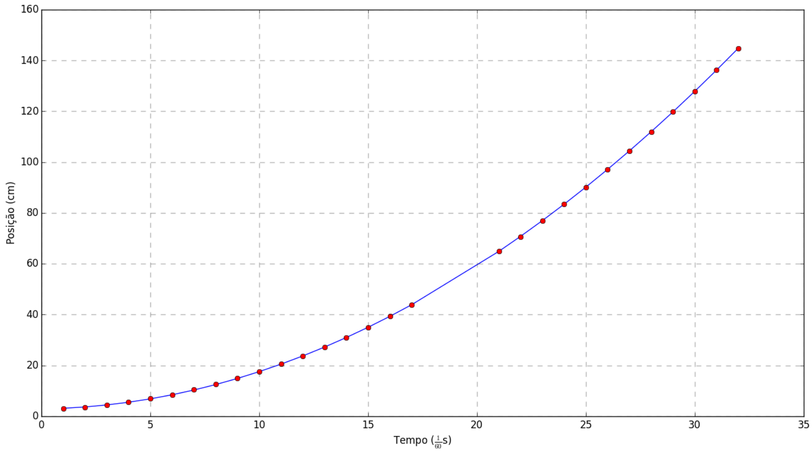

Figura 2: Ajuste obtido para o conjunto de dados fornecido pelo arquivo `exemplo1`.

2.3 Posto Incompleto

Para gerar A com posto incompleto, utilizou-se o progrma do `ep1` que gera A e b aleatóriamente com A singular, ou seja, com posto imcompleto. A partir deste gerador, criou-se uma matriz A quadrada de 700 linhas e singular, com posto 699. Esta A e b de $Ax = b$ estão salvos no arquivo `teste`. Tomando este arquivo como entrada do programa, constata-se que de fato o programa encontra o posto incompleto de A.

Referências

[1] *David S. Watkins,* Fundamentals of Matrix Compputation, Wiley, Third Edition.

[2] *Math Works,* `http://www.mathworks.com/moler/leastsquares.pdf`