Lab 3 – Sistemas Lineares (*Big Bang*)

Material necessário:

-Celular com câmera

- 3 bolas

Experiência:

Para decifrar os enigmas do início do universo, matemáticos utilizam dados de observações atuais para prever como era anteriormente. Extrapolando essa técnica, seria possível determinar o início do universo - *Big Bang.*

Este experimento visa determinar o momento exato e a posição de início de um universo ligeiramente simplificado (composto por 3 partículas). O modelo do universo simulado irá gerar um sistema de equações lineares que poderá ser resolvido utilizando as técnicas vistas em sala de aula.

Um componente do grupo deverá segurar 3 bolas de forma empilhada enquanto o outro filma de cima. A câmera deve permanecer parada e as 3 bolas devem ser soltas simultaneamente.

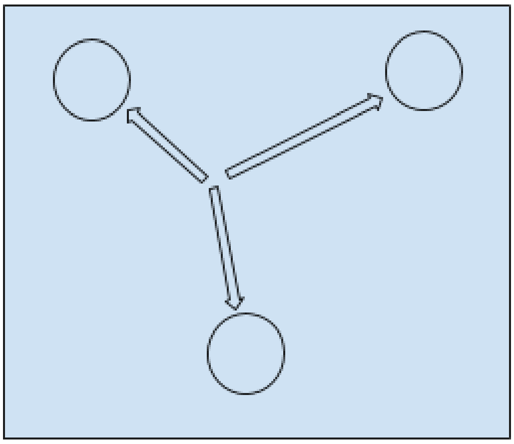

Figura 1: Universo simulado.

As bolas irão quicar em uma posição em comum antes de se afastarem.

O grupo deve capturar 3 quadros e seus respectivos tempos com precisão de milissegundos:

Quadro 1: Momento que as bolas tocam juntas no chão;

Quadro 2: Dois ou três quadros antes de uma das bolas sair da imagem;

Quadro 3: Último quadro antes de uma das bolas sair da imagem;

Para cada imagem deve descobrir as posições x e y de cada partícula (bola). Lembre que na primeira imagem as posições de x e y de todas as bolas devem supostamente ser iguais. Plote a primeira imagem com todas as posições coletadas.

O movimento de cada uma das partículas pode ser modelado, aproximadamente, por uma equação de movimento retilíneo uniforme (planar):

$$p = p_0 + V.(t - t_0)$$

em que p é a posição da bola no tempo t, V é a velocidade da partícula e p0 é a posição inicial no tempo t0.

Decompondo o sistema nas componentes x e y, teremos:

$$x_i = x_0 + sen(\theta_i).V_i.(t - t_0)$$

$$y_i = y_0 + sen(\theta_i).V_i.(t - t_0)$$

em que x_i e y_i são a posição, V_i é a velocidade e θ_i o ângulo formado pela i-ésima partícula em relação ao eixo x.

O sistema formado para as três partículas será:

$$\begin{pmatrix} 0 & -\cos(\theta_1) & V_1 sen(\theta_1)\cos(\theta_1) \\ sen(\theta_2) & -\cos(\theta_2) & 0 \\ sen(\theta_3) & 0 & -V_3 sen(\theta_3)\cos(\theta_3) \end{pmatrix} \begin{pmatrix} x_0 \\ y_0 \\ t_0 \end{pmatrix} = \begin{pmatrix} -cos(\theta_1)y_{1f} + V_1 sen(\theta_1)cos(\theta_1)t_f \\ sen(\theta_2)x_{2f} - cos(\theta_2)y_{2f} \\ sen(\theta_3)x_{3f} - V_3 sen(\theta_3)cos(\theta_3)t_f \end{pmatrix}$$

em que x_{if} e y_{if} são a posição e t_f o tempo da i-ésima partícula no quadro 3.

Utilize os quadros 2 e 3 para encontrar a velocidade e o ângulo (utilize a função atan() do Scilab com 2 argumentos) de cada partícula.

Implemente o método de resolução por LU e resolva o sistema. Compare os valores obtidos com o primeiro quadro (erro relativo)

Justifique, utilizando as técnicas vistas em sala de aula, se a convergência para os métodos iterativos é garantida. Independente da convergência ser garantida, utilize o método de Gauss-Seidel para encontrar a solução do sistema.

Programa:

Os algoritmos abaixo implementam os métodos de Resolução Retroativa, Eliminação de Gauss, e Jacobi, respectivamente.

Algoritmo de resolução retroativa:
Entrada: matriz de coeficientes triangular superior A e vetor de termos independentes b
Saída: solução x do sistema

```
[L,c] = tamanho(A); // no scilab: size(A)
para i=L:-1:1
        soma = 0;
        para j=i+1:c
                soma = soma + x(j)*A(i,j);
        fim_para
        x(i) = (b(i) - soma)/A(i,i);
fim_para
```

Teste do código:

A=[2 3 -1; 0 -2 -1; 0 0 5];

b=[5; -7; 15]

Chamada: x=ResolucaoRetroativa(A,b)

Saída: x=[1 2 3]^t

Algoritmo do método de Gauss:
Entrada: matriz de coeficientes A e vetor de termos independentes b
Saída: solução x do sistema

```
[L,c] = tamanho(A);
Aa = [A , b];          //matriz aumentada
para i=1 ate L-1
        pivo=Aa(i,i);
        para j=i+1 ate c
                m = Aa(j,i)/pivo;
                Aa(j,:) = Aa(j,:) - m* Aa(i,:);
        fim_para
fim_para
x = ResolucaoRetroativa(Aa(1:L,1:c), Aa(:,c+1));
```

Teste do código:

A=[2 3 -1; 4 4 -3; 2 -3 1];

b=[5; 3; -1]

Chamada: x=Gauss(A,b)

Saída: x=[1 2 3]^t

Algoritmo do método de Jacobi:
Entrada: matriz de coeficientes A, vetor de termos independentes b, solução inicial x0, tolerancia T e número máximo de iterações N.
Saída: solução xn do sistema

```
[L,c] = tamanho(A);
cont = 0;
xa = xn = x0;
faça
        xa=xn;
        para i=1 ate L
                soma=0
                para j= 1 ate L, j<>i
                        soma = soma + A(i,j)*xa(j)
                fim_se
                xn(i)=(b(i)-soma)/A(i,i);
        fim_para
        cont = cont +1;
Enquanto max(abs(xn-xa)) > T & cont < N
```

Teste do código:

A= [-4 0 -0.5 1; 2 3 3 4; 0 1 -2 0; 0 0 1 -1];

b= [0; 20; 0; -1];

Chamada:
[x,iter]=Jacobi(A,b,zeros(4,1),10^-3,300)

Saída:
iter = 275;

x=[0.3964404
2.3383478
1.1690189
2.1698738];