

## SISTEMAS DE EQUAÇÕES DISCRETOS

Os sistemas de equações discretas auxiliam na solução de muitos tipos de problema através do uso do computador, além de ajudar na criação de ferramentas interessantes na sociedade, por exemplo, criptografar dados.

Em este trabalho – prova se apresenta um sistema de equações discretas que apresenta um comportamento caótico, usado para criptografar dados. Pretende-se que **três estudantes passem um código feito na linguagem Matlab a uma outra linguagem.**

As tarefas serão feitas em duas etapas, a primeira é exclusiva de uma pessoa, denominada líder. **Os líderes serão os estudantes cujos números de matrícula terminarem em 5, 7, 8 ou 9.** Na segunda etapa estarão envolvidos um líder e uma dupla de estudantes. A dupla será conformada por dois estudantes, que não sejam líder, de forma livre.

O líder é responsável por criar uma estratégia, sem saber especificamente a linguagem na que vá ser desenvolvido o projeto, só sabe que deve suportar orientação a objetos. Pelo que deve gerar uma estratégia geral, funcional a qualquer linguagem e usar os conceitos básicos da modularidade. Para isso deve desenvolver um fluxograma descendente ou diagrama em UML se o preferir. Além, deve entender o problema e a codificação em Matlab, para assim poder passar instruções a dupla encarregada da codificação.

A dupla escolherá uma linguagem onde possa ser usada a orientação a objetos. Eles serão encarregados de seguir as explicações do líder e fazer a codificação e realizar os testes necessários para o correto funcionamento do programa.

**Problema:** Tem-se o seguinte sistema de equações de diferenças:

$$\begin{aligned}x_{n+1} &= x_{n-1} + h \cdot a \cdot (y_{n-1} - x_{n-1}) \cdot b_1^{-1} - f(x_{n-1}) \cdot b_1^{-1} \\y_{n+1} &= y_{n-1} + h \cdot a \cdot (y_{n-1} - x_{n-1}) \cdot b_2^{-1} + z_{n-1} \cdot b_2^{-1} \\z_{n+1} &= z_{n-1} + h \cdot (y_{n-1} - R_L z_{n-1}) \cdot g^{-1}\end{aligned}$$

Onde  $f(x_n) = m_0 \cdot x_n + 0.5 \cdot (m_1 - m_0) \cdot (|x_n + B_p| - |x_n - B_p|)$

$a, b, g, R_L, m_1 - m_0, B_p$ , são constantes e em todos os trabalhos são iguais.

Além, sabe-se os valores para  $x_0, y_0, z_0$ , que são diferentes para cada grupo. Para o grupo do líder 3, **deseja-se que se comparem as soluções para “y” na posição 100 para dos sistemas de equações um com os valores iniciais :  $x_0 = 0,2$ ;  $y_0 = 0,1$ ;  $z_0 = 0,1$ ; e um segundo sistema com valores iniciais:  $x_0 = 0,2$ ;  $y_0 = 0,1$ ;  $z_0 = 0,1$ ;**

A continuação a solução para um único sistema em matlab. O sistema de equações em Matlab fica:

$$\begin{aligned}x(n+1) &= x(n-1) + h * X( x(n-1), y(n-1), a, b1); \\y(n+1) &= y(n-1) + h * Y( x(n-1), y(n-1), z(n-1), a, b2); \\z(n+1) &= z(n-1) + h * Z( y(n-1), z(n-1), g, RL); \end{aligned}$$

onde “X”, “Y”, “Z” e “f” são funções definidas em arquivos individuais.  
As funções em Matlab são:

```

para "X":
    function x=X(xn,yn,a,b1)
        x=(a*(yn-xn)*b1^-1)-(f(xn)*b1^-1);
    end
para "Y":
    function y=Y(xn,yn,zn,a,b2)
        y=(a*(-yn+xn)*b2^-1)+(zn*b2^-1);
    end

para "Z":
    function z=Z(yn,zn,g, RL)
        z=(-yn/g)-(RL*zn*g^-1);
    end

para "f":
function F=f(xn)
    m0=-0.7879;m1=-1.4357;Bp=1;
    F=m0*xn+0.5*(m1-m0)*(abs(xn+Bp)-abs(xn-Bp));
end

```

Agora bem, o programa principal, onde encontra-se a solução para o vetores x, y e z é o seguinte:

```

% o símbolo "%" em matlab serve para fazer comentários e a documentação do código
% Clear limpa a memoria, clc limpa a tela, close all fecha as janelas abertas
clear, clc, close all

% Valores para as constantes
a=0.9259; b1=0.15; b2=1.5;g=0.1;RL=0.002;

%Valores iniciais das equações
x0=0.10;y0=0.1;z0=0.1;t0=0;

% h é o tamanho em que se avança no tempo, deve ser pequeno, muito menor que 1
h=0.02;

%Inicialização das equações
x1=x0+h*X(x0,y0,a,b1);
y1=y0+h*Y(x0,y0,z0,a,b2);
z1=z0+h*Z(y0,z0,g,RL);

% n indica o número de dados a serem encontrados, é dizer as vezes que se soluciona as equações,
%enquanto mais vezes se tem um comportamento mais completo.
n=10000;

% A função zeros de Matlab cria um vetor/matriz de zeros. Zeros (m,n), seria uma matriz de zeros
% de m filas e n colunas, no caso aqui, tem-se uma fila e n colunas. Ou seja, criam-se os vetores x,
% y e z.
xn=zeros(1,n); yn=zeros(1,n); zn=zeros(1,n);

% Inicializem os vetores x, y e z com os valores para n=0 e n=1;
xn(1)=x0;x(2)=x1;

```

```

yn(1)=y0;yn(2)=y1;
zn(1)=z0;zn(2)=z1;

```

```

% Encontram-se as n soluções do sistema de equações
for i=2:n-1
% Aqui estão as equações de diferencias no formato de matlab, equivalentes as equações de
% diferencias expressas em (1).
    xn(i+1)=xn(i-1)+h*X(xn(i-1),yn(i-1),a,b1);
    yn(i+1)=yn(i-1)+h*Y(xn(i-1),yn(i-1),zn(i-1),a,b2);
    zn(i+1)=zn(i-1)+h*Z(yn(i-1),zn(i-1),g,RL);
end % fim do for

```

```

%cria-se um vetor com a metade do tempo. Isto devido a questões de velocidade, é dizer só
% graficarm-se a metade dos pontos. O valor de 3*h indica que se começa a contar desde o ponto 3,
% devido que os dois primeiros são os pontos iniciais. 2*h indica o tamanho do passo do tempo,
% como só se desejam gráficar a metade dos pontos, por isso vai o dois, caso quera o total dos
% pontos basta tirar o 2. O n*h é o tempo final, onde n é o nuemro de soluções e h o passo do
% tempo.
H=(3*h:2*h:n*h);

```

```

figure(1)% cria uma janela
% Gráfica x vs h, que equivale a x vs t. '-' indica uma linha sólida, enquanto 'LineWidth',0.25 é o
tamanho dessa linha.

```

```

    plot([1,H], xn(1,1:2:end), '-', 'LineWidth',0.25, 'color',[0.1 0.1 0.8])

```

```

figure(2)

```

```

% Gráfica y vs h, que equivale a y vs t.

```

```

    plot([1, H], yn(1,1:2:end), '-', 'LineWidth',0.25, 'color',[0.2 0.2 1])

```

```

figure(3)

```

```

% Gráfica z vs h, que equivale a z vs t.

```

```

    plot([1, H], zn(1,1:2:end), '-', 'LineWidth',0.25, 'color',[0.2 0.2 0.8])

```

```

% Até aqui vai o código, fim.

```

A continuação se apresentam as gráficas obtidas em Matlab, só como informação. Neste projeto não se deseja necessariamente as gráficas, e sim as soluções para x, y e z.

