Atividade 2 – Métodos numéricos para SED de Problema de valor inicial

Análise Matemática 2

Docente: Arménio Correia

Autores:

Daniel Duarte Dias Ferreira Albino - a2020134077 – LEI

Miguel Ferreira Neves - a2020146521 - LEI

Nuno Alexandre Domingues - a2020109910 - LEI

Engenharia Informática

Coimbra, maio de 2021

Índice

1. Introdução

Este trabalho foi realizado no âmbito da unidade curricular de Análise Matemática 2, do curso de Licenciatura Informática do Instituto Superior de Engenharia de Coimbra.

O principal objetivo consiste, na aplicação de conhecimentos sobre a resolução de Equações Diferenciais Ordinárias (EDOs) de 2ª ordem, aplicadas e transformadas em sistemas de duas equações diferenciais e aplicações de Métodos numéricos, que permitem uma aproximação a solução exata.

Com este trabalho procura-se desenvolver o sentido de responsabilidade, investigação, competências algorítmicas e claro, a aprendizagem da linguagem de programação *Matlab*.

Este trabalho foi desenvolvido com interface gráfica *(APP)*, onde o utilizador pode usar no intuito de resolver problemas dinâmicos. Importante referir que esta *APP*, já tem alguns problemas dinâmicos predefinidos, em que, é só necessário o utilizador selecionar qual destes problemas quer analisar.

<u>Este relatório subdivide-se em três partes:</u>

1º Parte:

- Explicitação dos vários Métodos numéricos.

3º Parte:

- Aplicação de equações diferenciais a problemas dinâmicos

1. Métodos Numéricos para resolução de PVI

1.1. Método de Euler para SED

O Método de Euler é um método numérico para aproximar a solução da equação diferencial, sendo que é o método com mais erro, em relação aos outros métodos que iremos analisar.

1.1.1. Fórmula de Euler

$$u_{i+1} = u_i + hf\left(t_i, u_{i,} v_i\right)$$

$$v_{i+1} = v_i + hg\left(t_i, u_{i,} v_i\right)$$

$$i = 0,1, \dots, n-1$$

<u>Onde:</u>

$u_{i+1}/v_{i+1} \rightarrow Próximo\ valor\ a\ calcular$

$u_i/v_i \rightarrow Valor\ aproximado\ na\ abcissa\ atual$

$f(t_i, u_i, v_i) \rightarrow Valor\ da\ equação\ f\ em\ (t_i, u_i, v_i)$

$g(t_i, u_i, v_i) \rightarrow Valor\ da\ equação\ g\ em\ (t_i, u_i, v_i)$

$h = \frac{b-a}{n} \rightarrow Amplitude\ de\ cada\ subintervalo\ (PASSO)$

$[a, b] \rightarrow Intervalo\ pretendido$

$n \rightarrow número\ de\ vezes, que\ o\ intervalo\ vai\ ser\ dividido$

1.1.2. Algoritmo / Função

Algoritmo:

1º Calcular o *h* (passo).

Nota: O *h* irá dar a amplitude da cada partição do intervalo [a, b], quanto menor o *h* menor o erro.

$$h = \frac{b-a}{n}$$

2º Criar um vetor, por hipótese u e atribuir ao u(1) a condição inicial u0.

$$u(1) = u0$$

3º Criar um vetor, por hipótese v e atribuir ao v(1) a condição inicial v0.

$$v(1) = v0$$

4º Calcular os u(n) e v(n) seguintes, seguindo o método de Euler, até à n iteração.

$$u_{i+1} = u_i + hf(t_i, u_{i,} v_i)$$

$$v_{i+1} = v_i + hg(t_i, u_{i,} v_i)$$

Função Euler para SED (MatLab):

```
function [t,u,v] = NEulerSED(f,g,a,b,n,u0,v0)

h = (b-a)/n;          % Cálculo do passo
t = a:h:b;            % Alocação de memória
u = zeros(1,n+1);     % Alocação de memória
v = zeros(1,n+1);     % Alocação de memória
u(1) = u0;            % Condição inicial
v(1) = v0;            % Condição inicial

    for i = 1:n                                 %Ciclo com n iterações
        u(i+1) = u(i)+h*f(t(i),u(i),v(i));  %Cálculo com o método de Euler até n
        v(i+1) = v(i)+h*g(t(i),u(i),v(i));  %Cálculo com o método de Euler até n
    end
end
```

1.2. Método de Euler Melhorado para SED

O Método de Euler Melhorado é um também método numérico para aproximar a solução da equação diferencial, mas como o nome indica é um melhoramento do método de Euler original, pois consegue melhores aproximações. **Nota:** As suas aproximações coincidem às do RK2.

1.2.1. Fórmula

$$u_{i+1} = u_i + hf\left(t_i, u_{i,} v_i\right)$$

$$v_{i+1} = v_i + hg\left(t_i, u_{i,} v_i\right),$$

$$u_{i+1} = u_i + \frac{h}{2}\left(f(t_i, u_i, v_i) + f(t_{i+1}, u_{i+1}, v_{i+1})\right)$$

$$v_{i+1} = v_i + \frac{h}{2}\left(g(t_i, u_i, v_i) + g(t_{i+1}, u_{i+1}, v_{i+1})\right)$$

$$i = 0,1, \dots, n-1$$

Onde:

$u_{i+1}/v_{i+1} \rightarrow Próximo\ valor\ a\ calcular$

$u_i/v_i \rightarrow Valor\ aproximado\ na\ abcissa\ atual$

$f(t_i, u_i, v_i) \rightarrow Valor\ da\ equação\ f\ em\ (t_i, u_i, v_i)$

$g(t_i, u_i, v_i) \rightarrow Valor\ da\ equação\ g\ em\ (t_i, u_i, v_i)$

$h = \frac{b-a}{n} \rightarrow Amplitude\ de\ cada\ subintervalo\ (PASSO)$

$[a, b] \rightarrow Intervalo\ pretendido$

$n \rightarrow número\ de\ vezes, que\ o\ intervalo\ vai\ ser\ dividido$

1.2.2. Algoritmo/Função

Algoritmo:

1º Calcular o *h* (passo).

Nota: O *h* irá dar a amplitude da cada partição do intervalo [a, b], quanto menor o *h* menor o erro.

$$h = \frac{b-a}{n}$$

2º Criar um vetor, por hipótese u e atribuir ao u(1) a condição inicial u0.

$$u(1) = u0$$

3º Criar um vetor, por hipótese v e atribuir ao v(1) a condição inicial v0.

$$v(1) = v0$$

4º Calcular os u(n) e v(n) seguintes, seguindo o método de Euler, até à n iteração.

$$u_{i+1} = u_i + hf(t_i, u_{i,} v_i)$$

$$v_{i+1} = v_i + hg(t_i, u_{i,} v_i)$$

$$u_{i+1} = u_i + \frac{h}{2}\left(f(t_i, u_i, v_i) + f(t_{i+1}, u_{i+1}, v_{i+1})\right)$$

$$v_{i+1} = v_i + \frac{h}{2}\left(g(t_i, u_i, v_i) + g(t_{i+1}, u_{i+1}, v_{i+1})\right)$$

Função Euler Melhorado para SED (MatLab):

```
function [t,u,v] = NEulerMSED(f,g,a,b,n,u0,v0)

h = (b-a)/n;        % Cálculo do passo
t = a:h:b;          % Alocação de memória
u = zeros(1,n+1);   % Alocação de memória
v = zeros(1,n+1);   % Alocação de memória
u(1) = u0;          % Condição inicial
v(1) = v0;          % Condição inicial
h2 = h/2;           % Atribuição da divisão h/2, para eficiência da função

    for i = 1:n                                                     %Ciclo com n iterações
        u(i+1) = u(i)+h*f(t(i),u(i),v(i));
        v(i+1) = v(i)+h*g(t(i),u(i),v(i));
        u(i+1) = u(i)+h2*(f(t(i),u(i),v(i))+f(t(i+1),u(i+1),v(i+1))); % Cálculo do método de Euler Melhorado até n
        v(i+1) = v(i)+h2*(g(t(i),u(i),v(i))+g(t(i+1),u(i+1),v(i+1))); % Cálculo do método de Euler Melhorado até n
    end
end
```

1.3. Método de RK2 para SED

O Método de Runge-Kutta de ordem 2 (RK2) é método numérico de passo simples, onde utiliza derivadas para aproximar a solução da equação diferencial. **Nota:** As suas aproximações coincidem às do Euler melhorado.

1.3.1. Fórmula

$$u_{i+1} = u_i + \frac{1}{2}(k_1u + k_2u)$$

$$v_{i+1} = v_i + \frac{1}{2}(k_1v + k_2v)$$

$$i = 0,1, \dots, n-1$$

Onde:

$u_{i+1}/v_{i+1} \rightarrow Próximo\ valor\ a\ calcular$

$u_i/v_i \rightarrow Valor\ aproximado\ na\ abcissa\ atual$

$f(t_i, u_i, v_i) \rightarrow Valor\ da\ equação\ f\ em\ (t_i, u_i, v_i)$

$g(t_i, u_i, v_i) \rightarrow Valor\ da\ equação\ g\ em\ (t_i, u_i, v_i)$

$k_1u = hf(t_i, u_i, v_i)$

$k_1v = hg(t_i, u_i, v_i)$

$k_2u = hf(t_{i+1}, u_i + k_1u, v_i + k_1v)$

$k_2v = hg(t_{i+1}, u_i + k_1u, v_i + k_1v)$

$h = \frac{b-a}{n} \rightarrow Amplitude\ de\ cada\ subintervalo\ (PASSO)$

$[a, b] \rightarrow Intervalo\ pretendido$

$n \rightarrow número\ de\ vezes, que\ o\ intervalo\ vai\ ser\ dividido$

1.3.2. Algoritmo/Função

Algoritmo:

1º Calcular o *h* (passo).

Nota: O *h* irá dar a amplitude da cada partição do intervalo [a, b], quanto menor o *h* menor o erro.

$$h = \frac{b-a}{n}$$

2º Criar um vetor, por hipótese u e atribuir ao u(1) a condição inicial u0.

$$u(1) = u0$$

3º Criar um vetor, por hipótese v e atribuir ao v(1) a condição inicial v0.

$$v(1) = v0$$

4º Cálculo dos parâmetros k1 e k2.

$$k_1u = hf(t_i, u_i, v_i)$$

$$k_1v = hg(t_i, u_i, v_i)$$

$$k_2u = hf(t_{i+1}, u_i + k_1u, v_i + k_1v)$$

$$k_2v = hg(t_{i+1}, u_i + k_1u, v_i + k_1v)$$

5º Calcular os u(n) e v(n) seguintes, seguindo o método de Euler, até à n iteração.

$$u_{i+1} = u_i + \frac{1}{2}(k_1u + k_2u)$$

$$v_{i+1} = v_i + \frac{1}{2}(k_1v + k_2v)$$

Função RK2 (MatLab):

```
function [t,u,v] = NRK2SED(f,g,a,b,n,u0,v0)

h = (b-a)/n;              % Cálculo do passo
t = a:h:b;                % Alocação de memória
u = zeros(1,n+1);         % Alocação de memória
v = zeros(1,n+1);         % Alocação de memória
u(1) = u0;                % Condição inicial
v(1) = v0;                % Condição inicial

for i = 1:n                                  %Ciclo com n iterações
    k1u = h*f(t(i),u(i),v(i));               %Parâmetro K1 da variavel u
    k1v = h*g(t(i),u(i),v(i));               %Parâmetro K1 da variavel v
    k2u = h*f(t(i+1),u(i)+k1u,v(i)+k1v);%Parâmetro K2 da variavel u
    k2v = h*g(t(i+1),u(i)+k1u,v(i)+k1v);%Parâmetro K2 da variavel v
    u(i+1) = u(i)+(k1u+k2u)/2;               %Cálculo da variável u com o método RK2 até n
    v(i+1) = v(i)+(k1v+k2v)/2;               %Cálculo da variável u com o método RK2 até n
end
end
```

1.4. Método de RK4

O Método de Runge-Kutta de ordem 4 (RK4) é método numérico de passo simples, que depende de uma função calculada em diferentes pontos.

1.4.1. Fórmula

$$u_{i+1} = u_i + \frac{1}{6}(k_1u + 2k_2u + 2k_3u + k_4u)$$

$$v_{i+1} = v_i + \frac{1}{6}(k_1v + 2k_2v + 2k_3v + k_4v)$$

$$i = 0,1, \dots, n-1$$

<u>Onde:</u>

$u_{i+1}/v_{i+1} \rightarrow Próximo\ valor\ a\ calcular$

$u_i/v_i \rightarrow Valor\ aproximado\ na\ abcissa\ atual$

$f(t_i, u_i, v_i) \rightarrow Valor\ da\ equação\ f\ em\ (t_i, u_i, v_i)$

$g(t_i, u_i, v_i) \rightarrow Valor\ da\ equação\ g\ em\ (t_i, u_i, v_i)$

$k_1u = hf(t_i, u_i, v_i)$

$k_1v = hg(t_i, u_i, v_i)$

$k_2u = hf\left(t_i + \frac{h}{2}\ ,\ u_i + \frac{1}{2}k1u\ ,\ v_i + \frac{1}{2}k1v\right)$

$k_2v = hg\left(t_i + \frac{h}{2}\ ,\ u_i + \frac{1}{2}k1u\ ,\ v_i + \frac{1}{2}k1v\right)$

$k_3u = hf\left(t_i + \frac{h}{2}\ ,\ u_i + \frac{1}{2}k2u\ ,\ v_i + \frac{1}{2}k2v\right)$

$k_3v = hg\left(t_i + \frac{h}{2}\ ,\ u_i + \frac{1}{2}k2u\ ,\ v_i + \frac{1}{2}k2v\right)$

$$k_4u = hf(t_{i+1}, u_i + k3u\ , v_i + k3v)$$

$$k_4v = hg(t_{i+1}, u_i + k3u\ , v_i + k3v)$$

$$h = \frac{b-a}{n} \rightarrow Amplitude\ de\ cada\ subintervalo\ (PASSO)$$

$$[a, b] \rightarrow Intervalo\ pretendido$$

$$n \rightarrow número\ de\ vezes, que\ o\ intervalo\ vai\ ser\ dividido$$

1.4.2. Algoritmo/Função

<u>Algoritmo:</u>

1º Calcular o h (passo).

Nota: O h irá dar a amplitude da cada partição do intervalo [a, b], quanto menor o h menor o erro.

$$h = \frac{b-a}{n}$$

2º Criar um vetor, por hipótese u e atribuir ao u(1) a condição inicial u0.

$$u(1) = u0$$

3º Criar um vetor, por hipótese v e atribuir ao v(1) a condição inicial v0.

$$v(1) = v0$$

4º Calcular os diferentes parâmetros k (k1, k2, k3, k4), pela ordem indicada.

$$k_1u = hf(t_i, u_i, v_i)$$

$$k_1v = hg(t_i, u_i, v_i)$$

$$k_2u = hf\left(t_i + \frac{h}{2}\ ,\ u_i + \frac{1}{2}k1u\ ,\ v_i + \frac{1}{2}k1v\right)$$

$$k_2v = hg\left(t_i + \frac{h}{2}\ ,\ u_i + \frac{1}{2}k1u\ ,\ v_i + \frac{1}{2}k1v\right)$$

$$k_3u = hf\left(t_i + \frac{h}{2}\ ,\ u_i + \frac{1}{2}k2u\ ,\ v_i + \frac{1}{2}k2v\right)$$

$$k_3v = hg\left(t_i + \frac{h}{2}\ ,\ u_i + \frac{1}{2}k2u\ ,\ v_i + \frac{1}{2}k2v\right)$$

$$k_4u = hf(t_{i+1}, u_i + k3u\ , v_i + k3v)$$

$$k_4v = hg(t_{i+1}, u_i + k3u\ , v_i + k3v)$$

5º Calcular os u(n) e v(n) seguintes, seguindo o método de Euler, até à n iteração.

$$u_{i+1} = u_i + \frac{1}{6}(k_1u + 2k_2u + 2k_3u + k_4u)$$

$$v_{i+1} = v_i + \frac{1}{6}(k_1v + 2k_2v + 2k_3v + k_4v)$$

Função RK4 (MatLab):

```
function [t,u,v] = NRK4SED(f,g,a,b,n,u0,v0)

h = (b-a)/n;          % Cálculo do passo
t = a:h:b;            % Alocação de memória
u = zeros(1,n+1);     % Alocação de memória
v = zeros(1,n+1);     % Alocação de memória
u(1) = u0;            % Condição inicial
v(1) = v0;            % Condição inicial

    for i = 1:n                                          %Ciclo com n iterações
        k1u=h*f(t(i),u(i),v(i));                         %Parâmetro K1 da variavel u
        k1v=h*g(t(i),u(i),v(i));                         %Parâmetro K1 da variavel v
        k2u=h*f(t(i)+(h/2),u(i)+(k1u/2),v(i)+(k1v/2));%Parâmetro K2 da variavel u
        k2v=h*g(t(i)+(h/2),u(i)+(k1u/2),v(i)+(k1v/2));%Parâmetro K2 da variavel v
        k3u=h*f(t(i)+(h/2),u(i)+(k2u/2),v(i)+(k2v/2));%Parâmetro K3 da variavel u
        k3v=h*g(t(i)+(h/2),u(i)+(k2u/2),v(i)+(k2v/2));%Parâmetro K3 da variavel v
        k4u=h*f(t(i+1),u(i)+k3u,v(i)+k3v);               %Parâmetro K4 da variavel u
        k4v=h*g(t(i+1),u(i)+k3u,v(i)+k3v);               %Parâmetro K4 da variavel v

        u(i+1) = u(i)+1/6*(k1u+(2*k2u)+(2*k3u)+k4u);  %Cálculo da variável u com o método RK4 até n
        v(i+1) = v(i)+1/6*(k1v+(2*k2v)+(2*k3v)+k4v);  %Cálculo da variável v com o método RK4 até n
    end
end
```

2. Problemas dinâmicos com EDO's de 2ª ordem

2.1. Problema do Pêndulo

O movimento do pêndulo é descrito por uma equação diferencial de 2ª ordem, que se apresenta de seguida:

$$\theta'' + \frac{c}{mL}\theta' + \frac{g}{L}\sin{(\theta)} = 0$$

Onde:

$\theta = \hat{A}ngulo\ do\ objeto\ com\ a\ vertical$

$m = Massa\ do\ objeto$

$L = Comprimento\ da\ corda$

$g = Constante\ de\ força\ de\ gravidade$

$c = Coeficiente\ de\ amortecimento$

Ao substituirmos as variáveis por valores, temos que:

$$\left.\begin{array}{l} g = 9.8 \\ L = 9.8 \\ c = 2.94 \\ m = 1 \\ y = \theta \\ y(0) = \frac{\pi}{2} \rightarrow (posição\ inicial) \\ y'(0) = 0 \rightarrow (velocidade\ inicial) \end{array}\right\} \quad y'' + 0.3y' + \sin{(\theta)} = 0$$

Então, ficamos com o seguinte PVI:

$$\begin{cases} y'' + 0{,}3y' + \sin(y) = 0 \\ y(0) = \dfrac{\pi}{2} \\ y'(0) = 0 \end{cases}$$

Agora transformamos, a equação diferencial ordinária de 2ª ordem num sistema de duas equações diferenciais ordinárias lineares de 1ª ordem.

Para tal, isolamos a derivada de maior ordem de fazemos uma mudança de variável, que se demonstra de seguida:

$$y'' = -0{,}3y' - \sin(y)$$

Mudança de variável:

$u = y$

$v = y'$

$u' = v$

$v' = y''$

Com isto, ficamos:

$$\begin{cases} u' = v \\ v' = -0{,}3v - \sin(u) \end{cases}$$

$y(0) = u(0) = \frac{\pi}{2}$ →Posição inicial

$y'(0) = v(0) = 0$ →Velocidade inicial

2.2. Problema mola-massa sem amortecimento

O movimento da mola-massa sem amortecimento é descrito por uma equação diferencial de 2ª ordem, que se apresenta de seguida:

$$mx'' + \ bx' + kx = f(t)$$

Onde:

$x = Deslocamento\ ao\ longo\ do\ tempo$

$m = Massa\ do\ objeto$

$k = Constante\ da\ mola$

$f(t) = força\ externa\ aplicada$

$b = Coeficiente\ de\ amortecimento$

Ao substituirmos as variáveis por valores, temos que:

$$\left.\begin{array}{l} k = 16 \\ m = 1 \\ f(t) = 0 \rightarrow não\ tem\ força\ externa\ aplicada \\ b = 0 \rightarrow não\ tem\ amortecimento \\ x(0) = \ 9 \ \rightarrow (posição\ inicial) \\ x'(0) = 0 \rightarrow (velocidade\ inicial) \end{array}\right\} \quad x'' + 16x = 0$$

Então, ficamos com o seguinte PVI:

$$\begin{cases} x'' + 16x = 0 \\ x(0) = 9 \\ x'(0) = 0 \end{cases}$$

Agora transformamos, a equação diferencial ordinária de 2ª ordem num sistema de duas equações diferenciais ordinárias lineares de 1ª ordem.

Para tal, isolamos a derivada de maior ordem de fazemos uma mudança de variável, que se demonstra de seguida:

$$x'' = -16x$$

Mudança de variável:

$u = x$

$v = x'$

$u' = v$

$v' = x''$

Com isto, ficamos:

$$\begin{cases} u' = v \\ v' = -16\mathrm{u} \end{cases}$$

$y(0) = u(0) = 9$ →Posição inicial

$y'(0) = v(0) = 0$ →Velocidade inicial

2.3. Problema mola-massa com amortecimento

O movimento da mola-massa com amortecimento é descrito por uma equação diferencial de 2ª ordem, que se apresenta de seguida:

$$mx'' + \ bx' + kx = f(t)$$

Onde:

$x = Deslocamento\ ao\ longo\ do\ tempo$

$m = Massa\ do\ objeto$

$k = Constante\ da\ mola$

$f(t) = força\ externa\ aplicada$

$b = Coeficiente\ de\ amortecimento$

Ao substituirmos as variáveis por valores, temos que:

$$\left.\begin{array}{l} k = 16 \\ m = 1 \\ f(t) = 0 \rightarrow não\ tem\ força\ externa\ aplicada \\ b = 1 \rightarrow tem\ amortecimento \\ x(0) = \ 9 \ \rightarrow (posição\ inicial) \\ x'(0) = 0 \rightarrow (velocidade\ inicial) \end{array}\right\} \quad x'' + x' + 16x = 0$$

Então, ficamos com o seguinte PVI:

$$\begin{cases} x'' + x' + 16x = 0 \\ x(0) = 9 \\ x'(0) = 0 \end{cases}$$

Agora transformamos, a equação diferencial ordinária de 2ª ordem num sistema de duas equações diferenciais ordinárias lineares de 1ª ordem.

Para tal, isolamos a derivada de maior ordem de fazemos uma mudança de variável, que se demonstra de seguida:

$$x'' = -x' - 16x$$

Mudança de variável:

$u = x$

$v = x'$

$u' = v$

$v' = x''$

Com isto, ficamos:

$$\begin{cases} u' = v \\ v' = -v - 16\mathrm{u} \end{cases}$$

$y(0) = u(0) = 9$ →Posição inicial

$y'(0) = v(0) = 0$ →Velocidade inicial

2.4. Circuito elétrico (RLC)

A carga acumulada num circuito elétrico (RLC) é descrito por uma equação diferencial de 2ª ordem, que se apresenta de seguida:

$$Lq'' + rq' + \frac{1}{c}q = e(t)$$

Onde:

$q = carga\ acumulada$

$L = Indutância$

$r = Resistência$

$e(t) = força\ eletromotriz$

$c = Capacitância$

Ao substituirmos as variáveis por valores, temos que:

$L = 10$

$r = 200$

$e(t) = 0 \rightarrow não\ tem\ força\ eletromotriz$

$c = 0.0005$

$x(0) = 1 \rightarrow (posição\ inicial)$

$x'(0) = 0 \rightarrow (velocidade\ inicial)$

$$10x'' + 200x' + \frac{1}{0.0005}x = 0$$

Então, ficamos com o seguinte PVI:

$$\begin{cases} 10x'' + 200x' + 2000x = 0 \\ x(0) = 1 \\ x'(0) = 0 \end{cases}$$

Agora transformamos, a equação diferencial ordinária de 2ª ordem num sistema de duas equações diferenciais ordinárias lineares de 1ª ordem.

Para tal, isolamos a derivada de maior ordem de fazemos uma mudança de variável, que se demonstra de seguida:

$$x'' = \frac{-200x' - 2000x}{10}$$

Mudança de variável:

$u = x$

$v = x'$

$u' = v$

$v' = x''$

Com isto, ficamos:

$$\begin{cases} u' = v \\ v' = \dfrac{-200v - 2000u}{10} \end{cases}$$

$y(0) = u(0) = 1$ →Posição inicial

$y'(0) = v(0) = 0$ →Velocidade inicial

2.5. Modelo de preços de mercado com expectativa de preços aplicado à Economia

A evolução dos preços de mercado, assumindo em todos os instantes de tempo t, se tem equilíbrio de mercado, no sentido em que as funções de oferta e procura são iguais, é descrito por uma equação diferencial de 2ª ordem, que se apresenta de seguida:

$$dP'' + \ cP' + bP + a = 0$$

Onde:

$P = Preço\ corrente\ da\ mercadoria$

$a = Constante\ positiva$

$b = Constante\ positiva$

$c = Constante\ real$

$d = Constante\ real$

Ao substituirmos as variáveis por valores, temos que:

$$\left.\begin{array}{l} a = 80 \\ b = 40 \\ c = 80 \\ d = 100 \\ P(0) = \ 0 \\ P'(0) = 3 \end{array}\right\} \quad 100P'' + 80P' + 40P + 80 = 0$$

Então, ficamos com o seguinte PVI:

$$\begin{cases} 100P'' + 80P' + 40P + 80 = 0 \\ x(0) = 0 \\ x'(0) = 3 \end{cases}$$

Agora transformamos, a equação diferencial ordinária de 2ª ordem num sistema de duas equações diferenciais ordinárias lineares de 1ª ordem.

Para tal, isolamos a derivada de maior ordem de fazemos uma mudança de variável, que se demonstra de seguida:

$$P'' = \frac{-80P' - 40P - 80}{100}$$

Mudança de variável:

$u = P$

$v = P'$

$u' = v$

$v' = P''$

Com isto, ficamos:

$$\begin{cases} u' = v \\ v' = \dfrac{-80v - 40u - 80}{100} \end{cases}$$

$y(0) = u(0) = 0$

$y'(0) = v(0) = 3$

3. Conclusão

Em suma, as equações diferenciais de qualquer ordem têm inúmeras aplicações para modelação de problemas dinâmicos que acontecem frequentemente na vida real, em muitas áreas, como por exemplo: Engenharia, Biologia, Economia, Física, Química e muitas mais.

A aplicação de sistemas de equações diferenciais, revela-se uma forma, mais rápida e fácil de calcular equações diferenciais de 2ª ordem, através de simplificação e mudança de variáveis.

É importante referir também que o Matlab, ainda não consegue resolver equações não lineares pela função dsolve(), pelo que no problema dinâmico do pêndulo, que por sua vez é uma equação não-linear, só foi possível calcular aproximações à solução exata, através dos métodos numéricos, sendo que conseguimos valores mesmo muito próximos á solução exata, se utilizarmos métodos como o RK2 e RK4 ou até mesmo o Euler Melhorado.

4. Referências Bibliográficas

http://www.mat.uc.pt/~amca/MPII0607/folha3.pdf

https://ifspmatematica.files.wordpress.com/2015/07/tcc_arnaldo.pdf

https://repositorioaberto.uab.pt/bitstream/10400.2/9299/1/ED_em_economia.pdf