

Modelagem de Índice Quantitativo para previsão de “crash” em bolsa de valores

I. Introdução

Esse estudo tem como objetivo a construção de um índice para previsão de *crashes* nos índices de bolsa de valores. Através do estudo da volatilidade dos principais papéis que compõem os índices ao longo do tempo e de um modelo baseado em equações diferenciais, serão feitas análises das previsões teóricas comparando-se os resultados com os dados reais dos papéis que compõem o índice em questão. No primeiro instante desse estudo serão introduzidas noções das ferramentas básicas que serão usadas nessa avaliação parcial, assim como a exposição de programações realizadas através de VBA, componente disponível no Microsoft Excel®.

Serão introduzidas nesse relatório noções de equações diferenciais ordinárias, métodos numéricos para a solução de sistemas de equações diferenciais, assim como suas variantes para soluções mais complexas e noções de estabilidade para o estudo de pontos críticos de um sistema de equações diferenciais. Serão apresentados exemplos como forma de aplicação do estudo de sistemas dinâmicos.

II. Noções de Equações Diferenciais

Muitos dos fenômenos físicos existentes são explicados por algumas relações, algumas vezes com influência de outros acontecimentos externos, outras simplesmente através de uma taxa de crescimento ou de transformação. Essas transformações podem ser explicadas por equações diferenciais, onde, para explicar taxas e reações para as mudanças temporais, essas equações contêm derivadas. Na área de ciências econômicas não é diferente, sendo que alguns dos acontecimentos podem ser de certa forma modelados através de equações diferenciais, como um modelo monetário para a determinação da taxa de câmbio de um país ou um modelo para o ajuste das transações correntes dessa economia.

Uma equação diferencial ordinária, por exemplo, é sempre do tipo:

$$\frac{dy}{dt} = f(t; y)$$

Um exemplo de equação diferencial ordinária é a equação diferencial linear conforme a forma explicitada abaixo.

$$\frac{dy}{dt} = ay + b$$

Nessa equação, “a” representa a taxa de crescimento e “b” um termo adicional constante nas taxas futuras.

Para a solução analítica de uma equação diferencial de primeira ordem, o primeiro e mais simples método, conhecido como o método de separação de variáveis, leva em conta as partes da equação que dizem respeito à mesma incógnita. Elas devem ser isoladas em torno da igualdade, e depois, essa igualdade deve ser integrada em ambos os lados da equação, determinando assim, qual é a solução em termos da incógnita principal. Exemplificando a seguir, dado uma equação do tipo:

$$\frac{dy}{dt} = y$$

Deve-se agora realizar o processo para a solução analítica dessa equação diferencial:

$$\frac{dy}{dt} = y \Rightarrow$$

$$\Rightarrow \frac{dy}{y} = dt \Rightarrow$$

$$\Rightarrow \int \frac{dy}{y} = \int dt \Rightarrow$$

$$\Rightarrow \ln|y| = t + c \Rightarrow$$

$$\Rightarrow y = e^t$$

O trabalho de iniciação científica em questão utilizará equações diferenciais mais complexas e de ordens superiores. Essa forma de resolução se torna extremamente complicada de ser utilizada em alguns desses modelos. É necessária, nesse caso, a utilização dos métodos numéricos de integração, propostos nas seções seguintes.

III. Métodos Numéricos para Integração

O início desse estágio consiste em realizar o estudo e a programação em VBA de alguns métodos numéricos para obter soluções para problemas de valor inicial. Inicialmente foram abordados dois métodos: o método de Euler e o método de Runge-Kutta de quarta ordem e passo fixo.

O método de Euler é o mais antigo e mais simples método de aproximação numérica conhecido, ele é expresso pela equação:

$$y_{n+1} = y_n + f(t_n, y_n)(t_{n+1} - t_n)$$

Onde $n = 0, 1, 2, \dots$ e $f(t_n, y_n)$ é a equação diferencial a ser resolvida. Nessa equação da aproximação é assumido que há um passo de integração de valor constante, e consiste em calcular diversas vezes essa equação usando o resultado obtido para o próximo valor, e assim sucessivamente, obtendo uma seqüência de valores que o aproximam do valor da solução para a equação diferencial utilizada em cada um de seus pontos.

No caso de uma programação computacional (em específico aqui o VBA) para esse método, temos o seguinte algoritmo:

```
For i = 1 To n
    y(i + 1) = y(i) + h * f(t, y(i))
    t = t + h
Next i
```

Onde h é o tamanho do passo, n é o número de passos e $f(t, y(i))$ é a função para a equação diferencial atrelada a um contador i .

Outro método existente é o método de Runge-Kutta de quarta ordem. Esse método parte das mesmas premissas do método de Euler, onde há uma equação que, calculada quatro vezes através de um valor inicial, realiza uma aproximação para a solução analítica. Porém, esse

método é três vezes mais preciso que o método de Euler, devido ao seu erro de truncamento local proporcional a h^5 . A fórmula desse método envolve uma média ponderada de valores da função em pontos diferentes dentro do intervalo de tempo de cada passo. A fórmula é dada por:

$$y_{n+1} = y_n + h \left(\frac{k_{n1} + 2k_{n2} + 2k_{n3} + k_{n4}}{6} \right)$$

onde:

$$\begin{aligned} k_{n1} &= f(t_n, y_n) \\ k_{n2} &= f\left(t_n + \frac{1}{2}h, y_n + \frac{1}{2}hk_{n1}\right) \\ k_{n3} &= f\left(t_n + \frac{1}{2}h, y_n + \frac{1}{2}hk_{n2}\right) \\ k_{n4} &= f(t_n + h, y_n + hk_{n3}). \end{aligned}$$

Aqui, como feito para o método de Euler, tem-se a programação computacional em VBA para esse método. Segue abaixo o algoritmo:

```

For i = 1 To n
    k1 = f(t, (y(i)))
    k2 = f((t + (h / 2)), (y(i) + ((h * k1) / 2)))
    k3 = k2
    k4 = f((t + h), (y(i) + (h * k3)))
    y(i + 1) = y(i) + (h / 6) * (k1 + 2 * k2 + 2 * k3 + k4)
    t = t + h
Next i

```

Onde mais uma vez n representa o número de passos que serão dados, h representa o tamanho do passo e f(t,y(i)) é também a função para a equação diferencial atrelada a um contador i. As variáveis k1, k2, k3 e k4 são as avaliações da equação diferencial ordinária para espaços de tempo diferentes.

A seguir alguns exemplos que demonstram ambos os métodos comparados às soluções analíticas, assim como o erro percentual de cada um deles comparados às soluções analíticas.

Dada a equação diferencial $\frac{dy}{dt} = y$, apresenta-se abaixo o gráfico da solução por ambos os métodos de integração (Euler e Runge-Kutta) e também pela solução analítica, que já foi mostrado anteriormente que é: $y = e^t$.

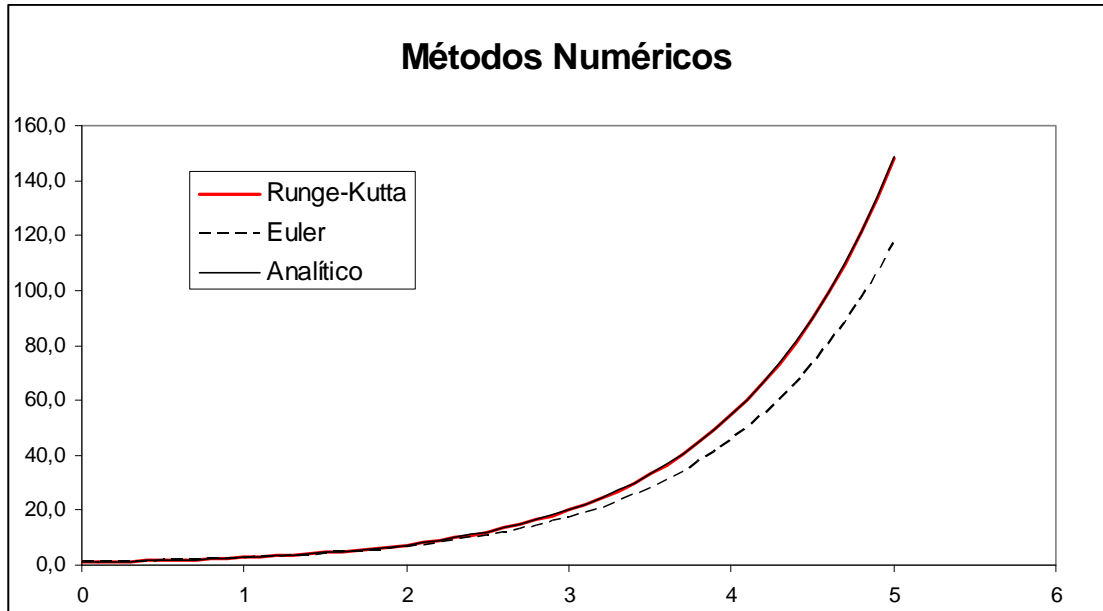


Figura 1 – Comparação entre os métodos de Euler e Runge-Kutta

Foi observado após a obtenção dos resultados (como mostrado na [Figura 1](#)) que a solução pelo método de Runge-Kutta, como citado anteriormente, por ter um erro de truncamento menor, foi muito mais precisa que a solução pelo método de Euler. Na [Figura 2](#) e na [Figura 3](#) são mostrados os cálculos dos erros percentuais de cada um dos métodos em relação à solução analítica.

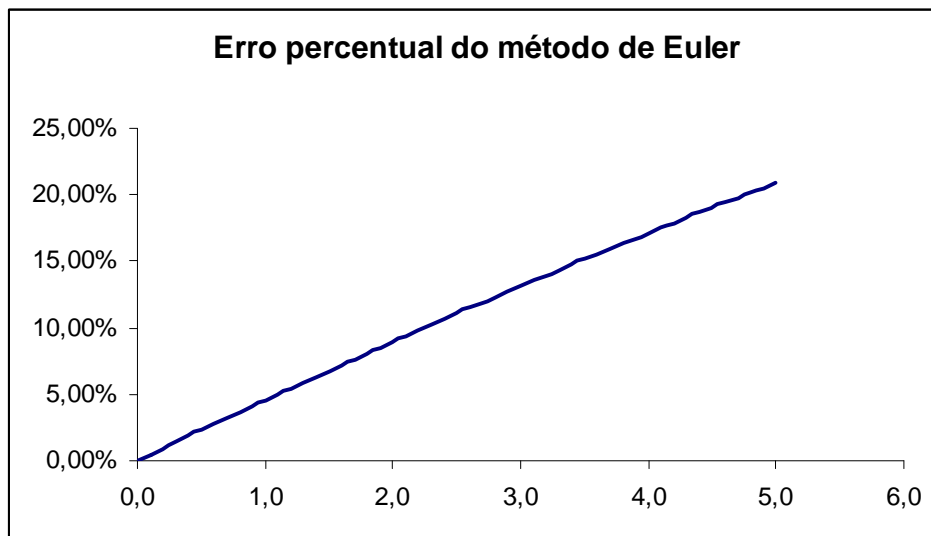


Figura 2 – Erro percentual do Método de Euler em relação à solução analítica

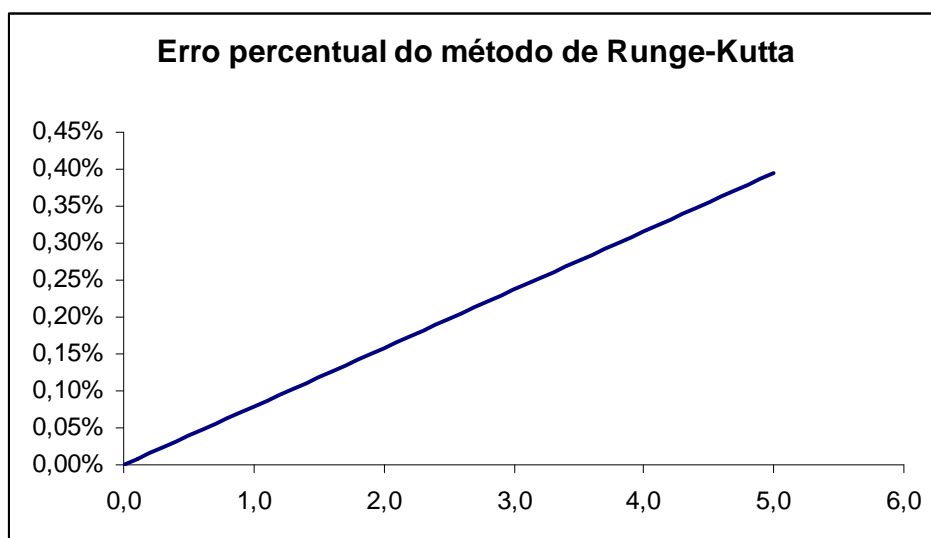


Figura 3 – Erro percentual do Método de Runge-Kutta em relação à solução analítica

O que foi observado após a programação do método e a obtenção de resultados para a equação diferencial proposta, foi que o método de Euler divergiu em 26,43% da solução analítica (Figura 2) da equação para uma seqüência de 50 valores com passo igual a 0,1, enquanto o método de Runge-Kutta (Figura 3) não chega sequer a um erro de 0,5% quando é solução para a mesma equação diferencial para a mesma seqüência e passo.

IV. Equações Diferenciais Lineares de Ordens Superiores

Equações Diferenciais Lineares de Ordens Superiores possuem uma grande importância para o estudo pois são mais representativas para fenômenos científicos. Da mesma forma que as equações diferenciais de primeira ordem, as equações diferenciais de ordens superiores traduzem geralmente algum fenômeno que sofre mudanças temporais. Porém, a solução de problemas que tratem da oscilação e dos movimentos cíclicos e ondulatórios somente será possível utilizando-se o método de Runge-Kutta de quarta ordem com passo fixo.

Uma equação diferencial de segunda ordem, por exemplo, é expressa por uma função do tipo:

$$\frac{d^2 y}{dt^2} = f\left(t, y, \frac{dy}{dt}\right)$$

Ou, sendo outra maneira de representação, também pode ser expressa como:

$$y'' = f(t; y; y')$$

Onde:

$$y' = \frac{dy}{dt}$$
$$y'' = \frac{d^2 y}{dt^2}$$

A equação diferencial ordinária de ordem superior é dita linear se a função f tem a forma:

$$f = g(t) - p(t)\frac{dy}{dt} - q(t)y$$

Para a solução de uma equação diferencial desse tipo, é necessário propor como hipótese uma solução do tipo $y = e^{rt}$, onde r é um parâmetro a ser determinado. Com isso tem-se que $y' = re^{rt}$ e $y'' = r^2 e^{rt}$.

Dada uma equação diferencial genérica:

$$ay''+by'+cy = 0$$

Tendo essa equação coeficientes constantes arbitrários, deve-se substituir a solução proposta nessa equação, onde se tem:

$$(ar^2 + br + c)e^{rt} = 0$$

Determinando as raízes desta equação e admitindo que sejam reais e distintas, temos r_1 e r_2 como raízes desta equação. Essa equação é conhecida como polinômio característico. Sendo como soluções do sistema:

$$y_1(t) = e^{r_1 t} \text{ e } y_2(t) = e^{r_2 t}$$

Utiliza-se do teorema que afirma que uma combinação das duas soluções com algum coeficiente atrelado também é solução da equação diferencial proposta. Logo, a solução a seguir também é solução da equação diferencial:

$$y(t) = K_1 e^{r_1 t} + K_2 e^{r_2 t}$$

E os coeficientes (K_1 e K_2) são obtidos através de um problema de valor inicial.

V. Escrevendo uma Equação Diferencial Linear de ordem “n” em um sistema de “n” equações diferenciais de primeira ordem

Será abordado nessa seção o método de transformação de uma equação diferencial linear de ordens superiores em um sistema de “n” equações diferenciais de primeira ordem. Esse tópico é muito importante para a proposta no que diz respeito à programação computacional para esse trabalho. O fato de transformar as equações diferenciais de ordens superiores em um sistema de equações mais simples faz com que seja possível a adesão dessa solução ao método de Runge-Kutta de quarta ordem e passo fixo abordado nesse trabalho.

Suponha como exemplo a equação diferencial de segundo grau que será utilizada posteriormente, que descreve o movimento harmônico simples com amortecimento.

Considere a seguinte equação diferencial de segunda ordem com a representação de Leibniz:

$$y'' + y' + 9y = 0$$

Definimos assim as variáveis que serão utilizadas para a montagem do sistema de acordo com o método de transformação da equação diferencial em um sistema. Primeiramente, devem-se renomear as variáveis do problema da seguinte forma:

$$\begin{cases} x_1 = y \\ x_2 = y' \end{cases}$$

Então, deriva-se as duas variáveis renomeadas:

$$\begin{cases} x_1' = y' \\ x_2' = y'' \end{cases}$$

Posteriormente, substituindo-se y' , tem-se:

$$\begin{cases} x_1' = x_2 \\ x_2' = y'' \end{cases}$$

Mas y'' é a própria equação diferencial, ou seja:

$$\begin{cases} x_1' = x_2 \\ x_2' = -y' - 9y \end{cases}$$

Logo, pode ser montado o sistema que possibilita a adaptação ao método de Runge-Kutta de quarta ordem e passo fixo para a solução da equação diferencial:

$$\begin{cases} x_1' = x_2 \\ x_2' = -x_2 - 9x_1 \end{cases}$$

VI. Método de Runge-Kutta na solução do sistema de Equações Diferenciais de Primeira Ordem

Esse tópico do estudo consiste em utilizar-se do método numérico de Runge-Kutta para a solução de problemas de Equações Diferenciais Lineares de ordens superiores quando transformadas em um sistema de "n" equações diferenciais de primeira ordem, onde "n" é o tamanho da ordem da Equação Diferencial Linear a qual é transformada. Com isso, o método de Runge-Kutta, opera da mesma forma que utilizado anteriormente no estudo, e isso o possibilita resolver sistemas de ordem superior.

Através da programação em VBA para resolver sistemas de equações diferenciais, pode-se obter soluções para esses sistemas de equações de maneira mais rápida. Segue abaixo o algoritmo da programação para o método de Runge-Kutta de quarta ordem aplicado à solução de um sistema de equações diferenciais de primeira ordem:

Do While t <= tf

$$k11(1) = fx1(t, x1(i), x2(i))$$

$$k12(2) = fx2(t, x1(i), x2(i))$$

$$dx(1) = (x1(i) + ((h * k11(1)) / 2))$$

$$dx(2) = (x2(i) + ((h * k12(2)) / 2))$$

$$k21(1) = fx1((t + (h / 2)), dx(1), dx(2))$$

$$k22(2) = fx2((t + (h / 2)), dx(1), dx(2))$$

$$dx(1) = (x1(i) + ((h * k21(1)) / 2))$$

$$dx(2) = (x2(i) + ((h * k22(2)) / 2))$$

$$k31(1) = fx1((t + (h / 2)), dx(1), dx(2))$$

$$k32(2) = fx2((t + (h / 2)), dx(1), dx(2))$$

$$dx(1) = (x1(i) + (h * k31(1)))$$

$$dx(2) = (x2(i) + (h * k32(2)))$$

$$k41(1) = fx1((t + h), dx(1), dx(2))$$

$$k42(2) = fx2((t + h), dx(1), dx(2))$$

$$x1(i + 1) = x1(i) + (h / 6) * (k11(1) + 2 * k21(1) + 2 * k31(1) + k41(1))$$

$$x2(i + 1) = x2(i) + (h / 6) * (k12(2) + 2 * k22(2) + 2 * k32(2) + k42(2))$$

Loop

Aqui h é o tamanho do passo, t_f é o tempo final, $x1(i)$ e $x2(i)$ denotam os diferentes valores das soluções e $fx1$ e $fx2$ são as funções do sistema de equações diferenciais, onde nesse caso tem ordem igual a dois. Em termos computacionais $fx1$ e $fx2$ são as *functions* da programação. Uma *function* é um instrumento para a programação computacional em VBA o qual é usado para recuperar a função matemática a ser utilizada no cálculo das soluções das equações diferenciais através dos métodos numéricos.

O exemplo abaixo resolvido com a utilização do método numérico é um Movimento Livre Não-Amortecido, onde se trata, por exemplo, do ciclo de um corpo acoplado a uma mola, dado que não haja forças externas e amortecimento. Nesse exemplo, o problema de valor inicial proposto é o seguinte:

Exemplo 1:

É proposta a seguinte equação diferencial:

$$y'' + 9y = 0$$

Onde as condições iniciais dadas são: $y(0) = \frac{10}{3}$ e $y'(0) = 4$.

Após o desmembramento em duas equações de primeira ordem dessa equação diferencial de segunda ordem:

$$\begin{cases} x_1 = y \\ x_2 = y' \end{cases}$$

$$\begin{cases} x_1' = y' = x_2 \\ x_2' = y'' = -9x_1 \end{cases}$$

Onde as condições iniciais agora se referem ao novo sistema, sendo elas definidas agora como: $x_1(0) = \frac{10}{3}$ e $x_2(0) = 4$.

A solução do sistema através do método de Runge-Kutta é mostrada na [figura 4](#), onde é observado que a massa oscila de um lado para o outro em torno de sua posição de equilíbrio. Esse movimento é conhecido como Movimento Harmônico Simples.

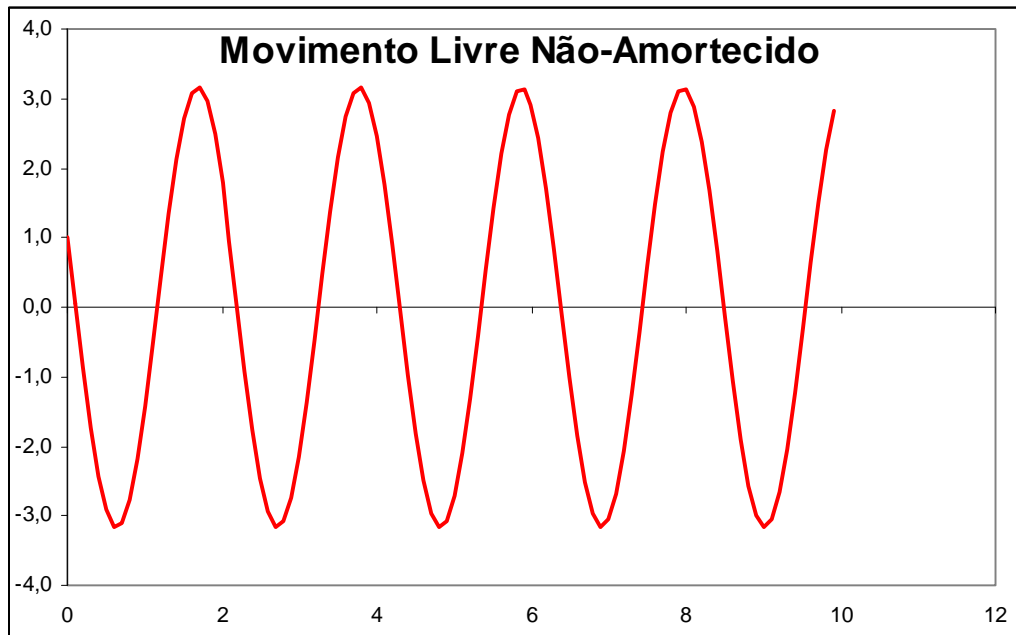


Figura 4 – Movimento Harmônico Simples

Exemplo 2:

Esse segundo exemplo trata-se do mesmo problema acima, todavia, agora, há um amortecimento na mola, que faz com que ela oscile ao longo do tempo parando finalmente na sua posição de equilíbrio. Dada a equação diferencial:

$$y'' + y' + 9y = 0$$

Onde as condições iniciais dadas são: $y(0) = \frac{10}{3}$ e $y'(0) = 4$.

Após o desmembramento em duas equações de primeira ordem dessa equação diferencial de segunda ordem:

$$\begin{cases} x_1 = y \\ x_2 = y' \end{cases}$$

$$\begin{cases} x_1' = y' = x_2 \\ x_2' = y'' = -9x_1 - x_2 \end{cases}$$

Onde as condições iniciais agora se referem ao novo sistema, sendo elas definidas agora como: $x_1(0) = \frac{10}{3}$ e $x_2(0) = 4$.

A solução do sistema através do método de Runge-Kutta é mostrada na [figura 5](#), onde é observado que a massa oscila de um lado para o outro em torno de sua posição de equilíbrio, porém, agora, com essa oscilação sendo reduzida por um freio, o que faz com que a massa chegue ao seu ponto de equilíbrio.

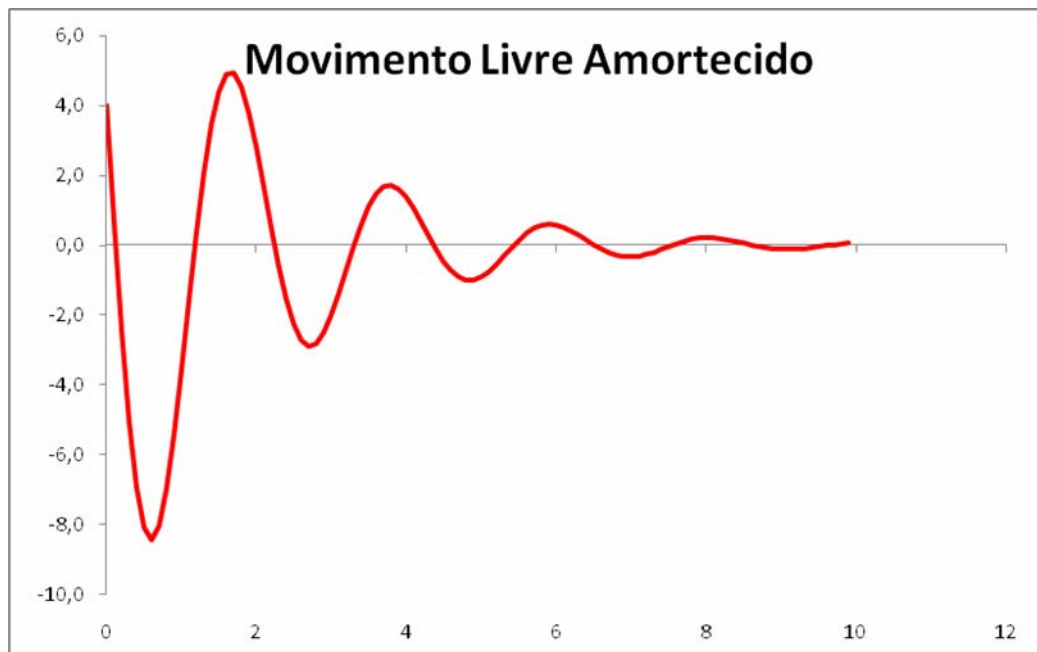


Figura 5 – Movimento Livre Amortecido

Nesse caso é observado que o corpo demora a chegar ao seu valor de posição de equilíbrio. Isso se deve provavelmente ao fato de o movimento estar sub-amortecido, ou seja, trata-se de uma fraca resistência em comparação com uma mola relativamente forte ou uma grande massa.

Exemplo 3:

Esse modelo trata-se de descrever a relação entre duas espécies que vivem em um mesmo meio-ambiente. Uma delas (o predador) se alimenta da outra (a presa), enquanto a presa se alimenta de qualquer outro tipo de comida. Há algumas hipóteses que devem ser feitas para a construção desse modelo de interação entre as duas espécies: A população de presas aumenta a uma taxa proporcional à taxa atual na ausência de predadores. A população de predadores vai à extinção na falta de presas e, a última delas, o encontro corriqueiro entre as duas espécies tende a frear o crescimento da população de presas e a acelerar o crescimento da população de predadores.

Dadas as equações diferenciais:

$$\begin{cases} \frac{dx}{dt} = x(1 - 0,5y) \\ \frac{dy}{dt} = y(-0,75 + 0,25x) \end{cases}$$

onde a primeira equação representa a dinâmica da população de presas e a segunda equação, a dinâmica da população de predadores. Nesse caso não é necessário o desmembramento das duas equações em ordens inferiores, havendo assim a possibilidade de já serem inseridas na programação computacional como dadas no problema (denotadas a seguir pelas “functions” da programação, tendo sua função já citada anteriormente):

```
Function fx1(t As Single, x As Single, y As Single) As Single
```

```
    fx1 = x * (1 - 0.5 * y)
```

```
End Function
```

```
Function fx2(t As Single, x As Single, y As Single) As Single
```

```
    fx2 = y * (-0.75 + 0.25 * x)
```

```
End Function
```

Após a solução através do método de Runge-Kutta ([Figura 6](#)), é possível observar que as hipóteses feitas anteriormente sobre as variáveis representativas de presas e predadores são verdadeiras.

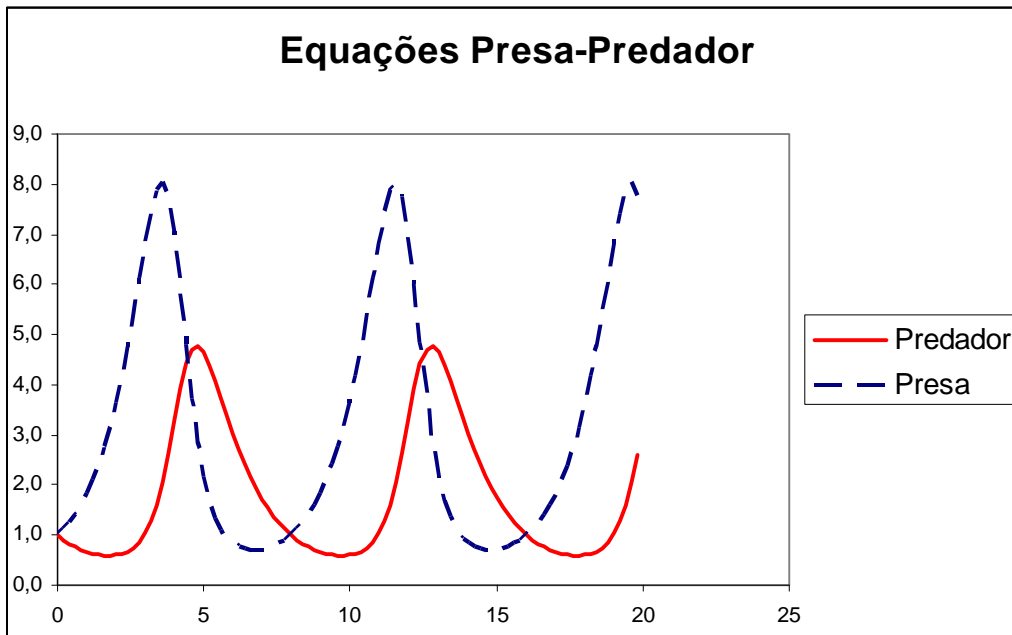


Figura 6 – Modelo Presa-Predador

A seguir a Tabela 1 mostra parte do trajeto de ambos, presa e predador, em função do tempo:

Tabela 1 – Trajeto de Presa e Predador em função do tempo

Tempo	Presa	Predador
1,00	1,816841	0,661877
1,20	2,080684	0,627875
1,40	2,389728	0,604198
1,60	2,749725	0,591207
1,80	3,166262	0,589822
2,00	3,644045	0,601735
2,20	4,185685	0,629732
2,40	4,789705	0,678194
2,60	5,447358	0,753833
2,80	6,137806	0,866734
3,00	6,821379	1,031590
3,20	7,431697	1,268599
3,40	7,870513	1,602271
3,60	8,015124	2,054517
3,80	7,753444	2,627659
4,00	7,050317	3,280248
4,20	6,004845	3,917047
4,40	4,826629	4,420272
4,60	3,727599	4,708632

VII. Estabilidade

A idéia da estabilidade de uma equação ou um sistema de equações diferenciais vem do fato de que muitas equações diferenciais não podem ser resolvidas de forma analítica, precisando assim de algum método numérico de integração para a sua solução. A estabilidade é uma informação qualitativa sobre o sistema dinâmico, podendo reproduzir diversas soluções apenas com a informação das condições iniciais. Para a verificação da estabilidade devem ser seguidos os processos listados no exemplo abaixo:

Dado o sistema de equações diferenciais: $\frac{dx}{dt} = x - xy$, $\frac{dy}{dt} = y + 2xy$:

O primeiro passo que deve ser tomado é a linearização desse sistema não-linear. Isso se dá através do encontro dos pontos críticos desse sistema (igualando cada uma das derivadas a zero) e a formulação da Matriz Jacobiana (que será aqui chamada de matriz A). Através da formulação sistema que iguala as duas derivadas a zero, como mostrado abaixo, obtemos os dois pontos críticos, que são as soluções possíveis do sistema:

$$\begin{cases} x - xy = 0 \\ y + 2xy = 0 \end{cases}$$

Os pontos críticos desse sistema são os pontos $\left(-\frac{1}{2}; 1\right)$ e $(0; 0)$.

Para a linearização desse problema, deve-se montar a matriz A, que tem a seguinte constituição:

$$A = \begin{pmatrix} f_{xx} & f_{xy} \\ f_{yx} & f_{yy} \end{pmatrix}$$

Após isso, os pontos críticos de cada uma das soluções do sistema são substituídos nas incógnitas da matriz A, tendo assim a matriz resultante da linearização do sistema. Logo, para o exemplo, tem-se:

$$A = \begin{pmatrix} 1-y & -x \\ 2y & 1+2x \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 2 & 0 \end{pmatrix}$$

para o ponto $\left(-\frac{1}{2}; 1\right)$, e

$$A = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

para o ponto $(0;0)$.

Os autovalores da matriz Jacobiana são os determinantes da propriedade de Estabilidade do sistema. Sendo assim, entre outras formas, eles podem ser calculados da seguinte forma:

$$\text{Det}(A - \lambda I) = 0$$

$$\text{Det} \begin{pmatrix} 1 - y - \lambda & -x \\ 2y & 1 + 2x - \lambda \end{pmatrix} = 0$$

$$(1 - y - \lambda)(1 + 2x - \lambda) + 2xy = 0$$

Assim, para o ponto $\left(-\frac{1}{2}; 1\right)$ são obtidos os seguintes autovalores:

$$(1 - 1 - \lambda) \left(1 + 2 \left(-\frac{1}{2} \right) - \lambda \right) + 2 \left(-\frac{1}{2} \right) (1) = 0$$

$$\lambda^2 - 1 = 0$$

$$\therefore \lambda_1 = 1 \text{ e } \lambda_2 = -1$$

Esses autovalores classificam o ponto como um ponto de sela instável, pois um deles é menor que zero e o outro, por sua vez, maior que zero. Para o outro ponto, $(0;0)$, temos os seguintes autovalores:

$$(1 - \lambda)(1 - \lambda) = 0$$

$$\lambda^2 - 2\lambda + 1 = 0$$

$$\therefore \lambda_1 = \lambda_2 = 1$$

Esses autovalores, diferentemente dos autovalores calculados anteriormente, classificam o ponto como um nó próprio instável, pois ambos os autovalores são iguais e maiores que zero.

Segue abaixo a tabela com as propriedades de Estabilidade para os sistemas de equações diferenciais:

Tabela 2 – Definições de Estabilidade

Autovalores	Tipo de Ponto Crítico	Estabilidade
$\lambda_1 > \lambda_2 > 0$	Nó	Instável
$\lambda_1 < \lambda_2 < 0$	Nó	Assintoticamente estável
$\lambda_2 < 0 < \lambda_1$	Ponto de sela	Instável
$\lambda_1 = \lambda_2 > 0$	Nó próprio ou impróprio	Instável
$\lambda_1 = \lambda_2 < 0$	Nó próprio ou impróprio	Assintoticamente estável
$\lambda_1, \lambda_2 = a \pm i\mu$	Ponto espiral	Se $a > 0$, Instável Se $a < 0$, Assintoticamente estável
$\lambda_1 = i\mu; \lambda_2 = -i\mu$	Centro	Estável

Para o exemplo anterior é apresentado o plano de fase (Figura 7). É possível observar através da análise do plano de fase que ambos os pontos estudados são instáveis, tendo como característica a não-convergência da solução do problema para o ponto crítico. Além disso, é possível observar através do gráfico em função do tempo (Figura 8) que a solução de ambas as equações do sistema divergirá partindo de um ponto inicial, o que corrobora a instabilidade do sistema.

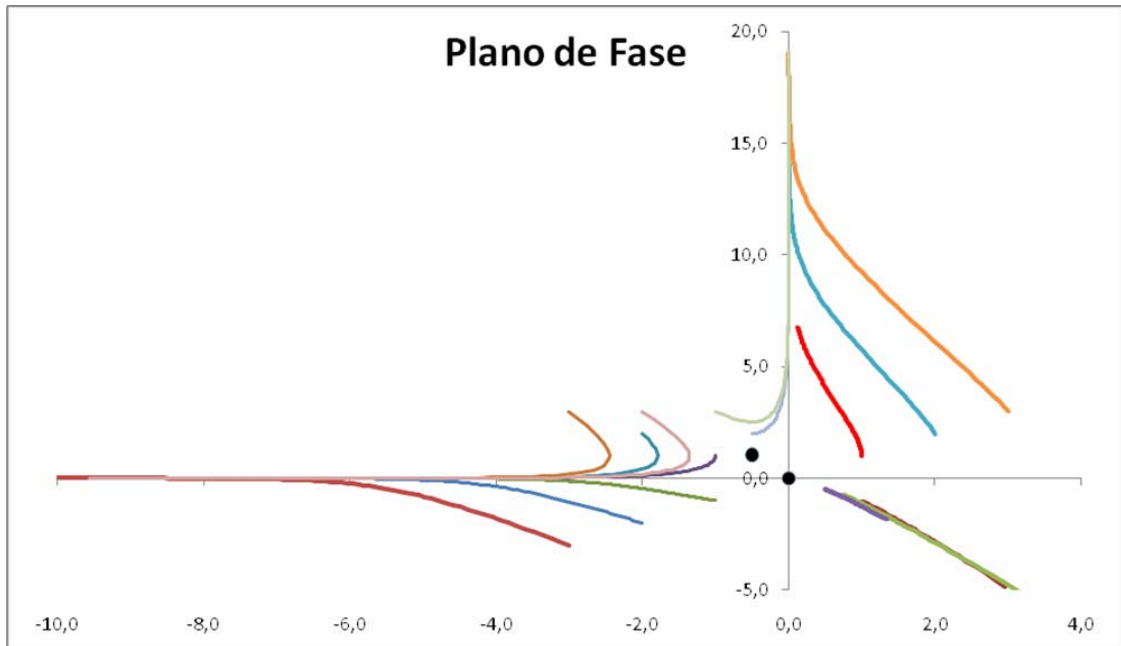


Figura 7 – Plano de Fase



Figura 8 – Solução em Função do Tempo

Exemplo 4:

Esse exemplo consiste na verificação da estabilidade para o sistema obtido no exemplo 1. De acordo, foi proposta a seguinte equação diferencial:

$$y''+9y = 0$$

Após o desmembramento em duas equações de primeira ordem dessa equação diferencial de segunda ordem:

$$\begin{cases} x_1 = y \\ x_2 = y' \end{cases}$$

$$\begin{cases} x_1' = y' = x_2 \\ x_2' = y'' = -9x_1 \end{cases}$$

Igualando cada uma das equações do sistema à zero, podem ser obtidos os pontos críticos:

$$\begin{cases} x_2 = 0 \\ -9x_1 = 0 \end{cases}$$

Portanto, esse sistema possui um ponto crítico o qual é o ponto $(0;0)$.

Para o cálculo dos autovalores que dizem respeito a cada ponto, deve ser montada a matriz A. Essa matriz tem a seguinte constituição:

$$A = \begin{pmatrix} f_{x_1 x_1} & f_{x_1 x_2} \\ f_{x_2 x_1} & f_{x_2 x_2} \end{pmatrix}$$

Sem a necessidade aqui de substituir os pontos críticos nas incógnitas da Matriz A, é obtido que, para esse exemplo:

$$A = \begin{pmatrix} 0 & 1 \\ -9 & 0 \end{pmatrix}$$

Os autovalores da matriz A são os determinantes da propriedade de Estabilidade do sistema. Sendo assim, entre outras formas, eles podem ser calculados da seguinte forma:

$$\text{Det}(A - \lambda I) = 0$$

$$\text{Det}\begin{pmatrix} -\lambda & 1 \\ -9 & -\lambda \end{pmatrix} = 0$$

$$\lambda^2 + 9 = 0$$

Assim, para esse sistema, são obtidos os seguintes autovalores:

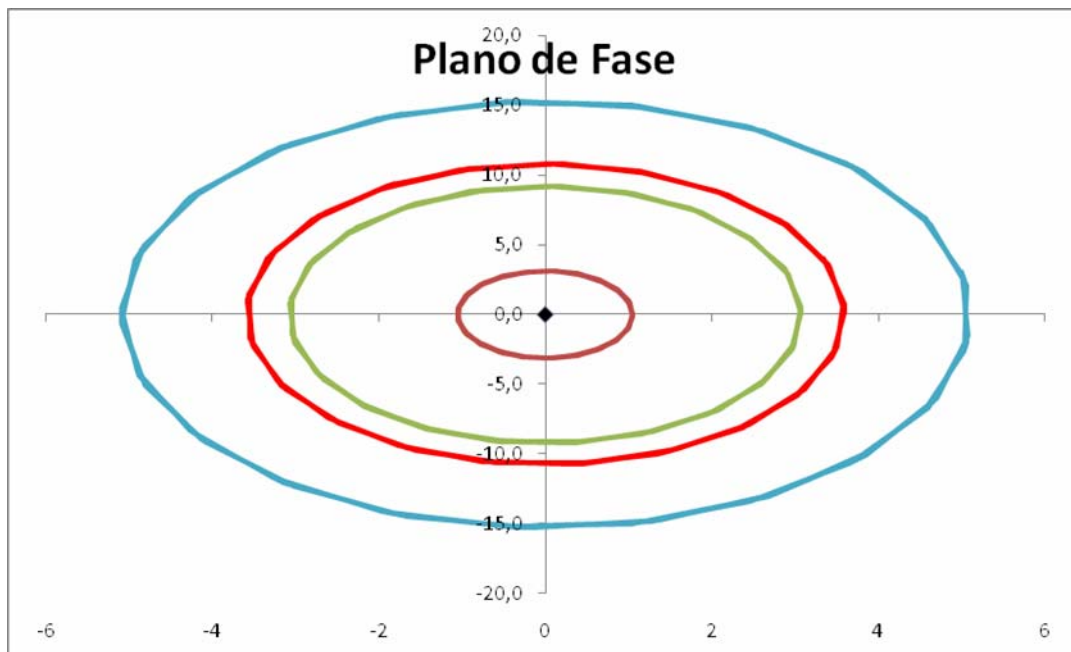
$$\lambda^2 = -9$$

$$\lambda = \pm\sqrt{-9}$$

$$\lambda = \pm 3i$$

$$\therefore \lambda_1 = 3i \text{ e } \lambda_2 = -3i$$

Esses autovalores, de acordo com a [tabela 2](#) classificam o ponto como um centro estável. Observando a [Figura 9](#) pode ser ratificada essa classificação.



[Figura 9](#) – Plano de Fase com ponto crítico estável

Exemplo 5:

Esse exemplo consiste na verificação da estabilidade para o sistema obtido no exemplo 2. De acordo, foi proposta a seguinte equação diferencial:

$$y'' + y' + 9y = 0$$

Após o desmembramento em duas equações de primeira ordem dessa equação diferencial de segunda ordem:

$$\begin{cases} x_1 = y \\ x_2 = y' \end{cases}$$

$$\begin{cases} x_1' = y' = x_2 \\ x_2' = y'' = -9x_1 - x_2 \end{cases}$$

Igualando cada uma das equações do sistema à zero, podem ser obtidos os pontos críticos:

$$\begin{cases} x_2 = 0 \\ -9x_1 - x_2 = 0 \end{cases}$$

Portanto, esse sistema possui um ponto crítico o qual é o ponto $(0;0)$.

Para o cálculo dos autovalores que dizem respeito a cada ponto, deve ser montada a matriz A. Essa matriz tem a seguinte constituição:

$$A = \begin{pmatrix} f_{x_1 x_1} & f_{x_1 x_2} \\ f_{x_2 x_1} & f_{x_2 x_2} \end{pmatrix}$$

Sem a necessidade aqui de substituir os pontos críticos nas incógnitas da Matriz A, é obtido que, para esse exemplo:

$$A = \begin{pmatrix} 0 & 1 \\ -9 & -1 \end{pmatrix}$$

Os autovalores da matriz A são os determinantes da propriedade de Estabilidade do sistema. Sendo assim, entre outras formas, eles podem ser calculados da seguinte forma:

$$\text{Det}(A - \lambda I) = 0$$

$$\text{Det} \begin{pmatrix} -\lambda & 1 \\ -9 & -1-\lambda \end{pmatrix} = 0$$

$$\lambda^2 + \lambda + 9 = 0$$

Assim, para esse sistema, são obtidos os seguintes autovalores:

$$\lambda = \frac{-1 \pm \sqrt{-36}}{2}$$

$$\lambda = \frac{-1 \pm 6i}{2}$$

$$\lambda = -\frac{1}{2} \pm 3i$$

$$\therefore \lambda_1 = -\frac{1}{2} + 3i \text{ e } \lambda_2 = -\frac{1}{2} - 3i$$

Esses autovalores, de acordo com a [tabela 2](#) classificam o ponto como um ponto espiral assintoticamente estável. Observando a [Figura 10](#) pode ser ratificada essa classificação.

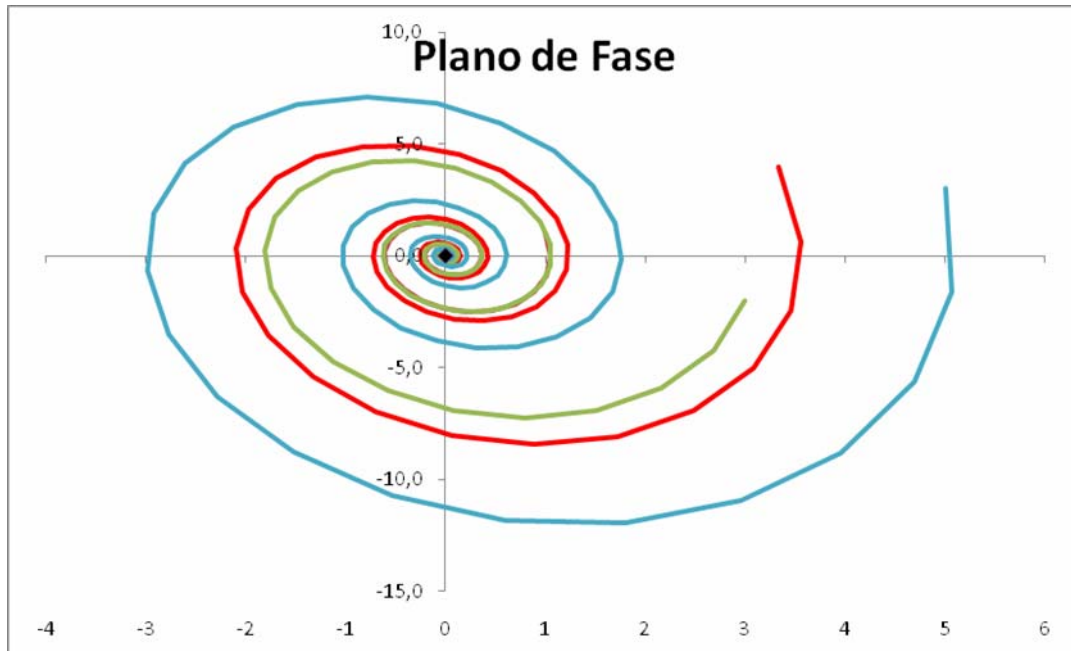


Figura 10 – Plano de Fase

VIII. Conclusão Parcial

Nessa primeira fase do estudo foram observadas algumas ferramentas que serão utilizadas posteriormente para as previsões de *crashes* na bolsa de valores, o qual é o objetivo principal do estudo. Primeiramente foram observadas as principais utilidades que as equações diferenciais ordinárias têm para a descrição de fenômenos físicos e as suas utilidades na área de ciências econômicas.

Foram estudados também os métodos de integração, os quais são alternativas ao método analítico. Eles são os métodos de Euler e de Runge-Kutta, sendo os dois métodos que facilitam a resolução de uma equação diferencial mais complexa a qual é inviável a sua solução através do método analítico. No estudo desses dois métodos foi observado que o método de Runge-Kutta é um método mais eficiente para a solução de uma equação diferencial. Como visto na figura 3, para o exemplo da seção, o erro não foi sequer de 0,5% em relação à solução analítica, enquanto o erro do método de Euler foi maior que 25% para a mesma equação diferencial, mesmo tamanho de passo e mesmo número de passos (figura 2).

Após a verificação da maior eficiência do método de Runge-Kutta, e o estudo da transformação de uma equação diferencial de ordem superior em um sistema de equações diferenciais, foi realizada a programação computacional para o método de Runge-Kutta de

quarta ordem com passo fixo o qual requer esse desmembramento da solução em um sistema de equações diferenciais.

Também foi feito o estudo de Estabilidade para um sistema de equações diferenciais. Em muitas equações diferenciais não há a possibilidade de se realizar a solução através da forma analítica, por isso deve ser considerado o estudo de estabilidade em cada um dos pontos críticos do sistema, que é uma informação qualitativa que mostra se o ponto crítico que é solução para o sistema é instável ou não, daí a idéia de saber se é possível a resolução da equação diferencial pela forma analítica ou se a solução, sendo um ponto instável, diverge para uma solução inexistente.

IX. Referências Bibliográficas

Boyce, W.E. e DiPrima, R.C. – *“Equações Diferenciais Elementares e Problemas de Valores de Contorno”*, Sétima Edição, Livros Técnicos e Científicos Editora S.A., 2001.

Shone, R. – *“Economic Dynamics”*, 1999 Edition, Cambridge University Press, 1997.

Araújo, E. – Nota de Aula: *“Equações Diferenciais Ordinárias de Primeira Ordem”*, Ibmec São Paulo, 2007.

Araújo, E. – Nota de Aula: *“Equações Diferenciais Ordinárias de Segunda Ordem com Coeficientes Constantes”*, Ibmec São Paulo, 2007.

Araújo, E. – Nota de Aula: *“Sistemas de Equações Diferenciais Lineares de Primeira Ordem com Coeficientes”*, Ibmec São Paulo, 2007.

Simon, C.P. e Blume, L. – *“Matemática para Economistas”*, Edição do ano de 2004, Bookman Editora, 1994.

X. Apêndice I – Programa em VBA para o Método de Euler

```
Private Sub Image1_Click()

End Sub

Private Sub TextBox1_Change()
t = TextBox1.Value

End Sub

Private Sub TextBox2_Change()
h = TextBox2.Value

End Sub

Private Sub TextBox3_Change()

n = TextBox3.Value

End Sub
Private Sub textbox4_change()

y = TextBox4.Text
End Sub

Private Sub commandbutton1_click()

Dim t As Single
Dim h As Single
Dim n As Single
Dim y(0 To 500) As Single

Dim grafsmi As Chart

Dim myChtObj As ChartObject
Dim dados As Range
Dim xval As Range
Dim iColumn As Long

t = TextBox1.Value
h = TextBox2.Value
n = TextBox3.Value

y(1) = 1
Cells(1, 1) = t
Cells(1, 2) = y(1)
For i = 1 To n
y(i + 1) = y(i) + h * f(t, y(i))
t = t + h
Cells(i + 1, 1) = t
Cells(i + 1, 2) = y(i + 1)
```

Next i

```
'deleta o gráfico passado (ignora)  
On Error Resume Next  
Worksheets("sheet1").ChartObjects(1).Delete
```

```
Set entrada = Worksheets("sheet1").Range(Cells(1, 2), Cells(50, 2))  
Set entradax = Worksheets("sheet1").Range(Cells(1, 1), Cells(50, 1))  
'o título é o nome do gráfico selecionado ou, caso nada esteja selecionado, SMI, que é o gráfico padrão
```

```
titulo1 = " Euler "
```

```
'define qual será o tamanho da escala do eixo X  
qntas = entrada.Count  
If qntas > n Then  
qntas2 = qntas / n  
Else  
qntas2 = 5  
End If
```

```
'-----grafico -2 -----
```

```
'deleta o gráfico passado ou ignora, caso não haja gráfico passado  
On Error Resume Next  
Worksheets("sheet1").ChartObjects(1).Delete
```

```
'adiciona um gráfico na sheet1 com os dados definidos  
Set grafsmi = ActiveWorkbook.Charts.Add
```

```
With grafsmi  
.SetSourceData Source:=entrada, PlotBy:=xlColumns  
.Location Where:=xlLocationAsObject, Name:="sheet1"  
End With
```

```
'define o tamanho do gráfico  
Worksheets("sheet1").ChartObjects(1).Height = 250  
Worksheets("sheet1").ChartObjects(1).Width = 400
```

```
'define algumas características do gráfico, como o título e os valores do eixo X  
With Worksheets("sheet1").ChartObjects(1).Chart  
.ChartType = xlLine  
.HasLegend = False  
.HasTitle = True  
.ChartTitle.Text = titulo1  
.SeriesCollection(1).XValues = entradax  
End With
```

```
'define o tamanho e a cor da área de plotagem  
ActiveChart.PlotArea.Select  
Selection.Top = 1  
Selection.Height = 250
```

```
Selection.Width = 400
Selection.Left = 1
With Selection.Interior
    .ColorIndex = 2
End With
```

```
'apaga a linha de grade
ActiveChart.Axes(xlValue).MajorGridlines.Select
Selection.Delete
```

```
'define que o eixo X cruz o Y no ponto 0
ActiveChart.Axes(xlValue).Select
With ActiveChart.Axes(xlValue)
    .CrossesAt = 0
End With
```

```
'define a formatação do eixo Y
Selection.TickLabels.AutoScaleFont = True
With Selection.TickLabels.Font
    .Size = 8.5
End With
Selection.TickLabels.NumberFormat = "0.0"
```

```
'define a escala do eixo X
ActiveChart.Axes(xlCategory).Select
With ActiveChart.Axes(xlCategory)
    .TickLabelSpacing = qntas2
    .TickMarkSpacing = qntas2
    .TickLabels.NumberFormat = "0"
End With
```

```
'formatação do eixo X
Selection.TickLabels.AutoScaleFont = True
With Selection.TickLabels.Font
    .Size = 8.5
End With
With Selection.TickLabels
    .Alignment = xlCenter
    .Offset = 100
    .Orientation = xlHorizontal
End With
With Selection
    .TickLabelPosition = xlLow
End With
```

```
'formatação da linha do gráfico da variavel 1
ActiveChart.SeriesCollection(1).Select
With Selection.Border
    .ColorIndex = 3
    .Weight = xlMedium
    .LineStyle = xlContinuous
End With
With Selection
    .MarkerStyle = xlNone
End With
```

```
'localização do título no gráfico  
  
ActiveChart.ChartTitle.Select  
Selection.Left = 29  
Selection.Top = 6  
'-----fim dos graficos -----
```

```
Unload Me
```

```
End Sub
```

```
Function f(t As Single, y As Single) As Single  
f = y
```

```
End Function
```

XI. Apêndice 2 – Programa em VBA para o Método de Runge-Kutta de quarta ordem e passo fixo

```
Sub rk()  
  
Dim k11(100) As Single  
Dim k21(100) As Single  
Dim k31(100) As Single  
Dim k41(100) As Single  
Dim k12(100) As Single  
Dim k22(100) As Single  
Dim k32(100) As Single  
Dim k42(100) As Single  
Dim t As Single  
Dim tf As Single  
Dim i As Single  
Dim x1(100) As Single  
Dim x2(100) As Single  
Dim h As Single  
Dim dx(100) As Single  
  
Dim grafico As Chart  
Dim entradax As Range  
Dim entraday As Range  
Dim entraday2 As Range  
Dim j As Single  
Dim n As Single  
  
On Error Resume Next  
Worksheets("sheet1").ChartObjects(1).Delete  
  
For j = 1 To 500
```

```
Worksheets("Sheet1").Cells(j + 1, 1).ClearContents
Worksheets("Sheet1").Cells(j + 1, 2).ClearContents
Worksheets("Sheet1").Cells(j + 1, 3).ClearContents
Next j
```

```
t = 0
h = 0.05
x1(1) = -2
x2(1) = 3
tf = 5
linha = 1
i = 1
```

```
Do While t <= tf
```

```
Cells(linha, 1) = t
Cells(linha, 2) = x1(i)
Cells(linha, 3) = x2(i)
```

```
k11(1) = fx1(t, x1(i), x2(i))
k12(2) = fx2(t, x1(i), x2(i))
dx(1) = (x1(i) + ((h * k11(1)) / 2))
dx(2) = (x2(i) + ((h * k12(2)) / 2))
```

```
k21(1) = fx1((t + (h / 2)), dx(1), dx(2))
k22(2) = fx2((t + (h / 2)), dx(1), dx(2))
dx(1) = (x1(i) + ((h * k21(1)) / 2))
dx(2) = (x2(i) + ((h * k22(2)) / 2))
```

```
k31(1) = fx1((t + (h / 2)), dx(1), dx(2))
k32(2) = fx2((t + (h / 2)), dx(1), dx(2))
dx(1) = (x1(i) + (h * k31(1)))
dx(2) = (x2(i) + (h * k32(2)))
```

```
k41(1) = fx1((t + h), dx(1), dx(2))
k42(2) = fx2((t + h), dx(1), dx(2))
```

```
x1(i + 1) = x1(i) + (h / 6) * (k11(1) + 2 * k21(1) + 2 * k31(1) + k41(1))
x2(i + 1) = x2(i) + (h / 6) * (k12(2) + 2 * k22(2) + 2 * k32(2) + k42(2))
```

```
linha = linha + 1
i = i + 1
t = t + h
```

```
Loop
```

```
n = tf / h
```

```
Set entradax = Worksheets("sheet1").Range(Cells(1, 1), Cells(n, 1))
Set entraday = Worksheets("sheet1").Range(Cells(1, 2), Cells(n, 2))
Set entraday2 = Worksheets("sheet1").Range(Cells(1, 3), Cells(n, 3))
```

```
Set grafico = ActiveWorkbook.Charts.Add
```

```
With grafico
.SetSourceData Source:=entraday2, PlotBy:=xlColumns
```

```
.Location Where:=xlLocationAsObject, Name:="sheet1"  
End With
```

```
Worksheets("sheet1").ChartObjects(1).Height = 250  
Worksheets("sheet1").ChartObjects(1).Width = 400
```

```
With Worksheets("sheet1").ChartObjects(1).Chart  
    .ChartType = xlXYScatterLinesNoMarkers  
    .HasLegend = False  
    .HasTitle = True  
    .ChartTitle.Text = "Eq. Diferencias de Ordem 2"  
    .SeriesCollection(1).XValues = entradax  
End With
```

```
ActiveChart.PlotArea.Select  
Selection.Top = 1  
Selection.Height = 250  
Selection.Width = 400  
Selection.Left = 1  
With Selection.Interior  
    .ColorIndex = 2  
End With
```

```
ActiveChart.Axes(xlValue).MajorGridlines.Select  
Selection.Delete
```

```
ActiveChart.Axes(xlValue).Select  
With ActiveChart.Axes(xlValue)  
    .CrossesAt = 0  
End With
```

```
tamanho = n
```

```
Selection.TickLabels.AutoScaleFont = True  
With Selection.TickLabels.Font  
    .Size = 8.5  
End With  
Selection.TickLabels.NumberFormat = "0.0"
```

```
ActiveChart.Axes(xlCategory).Select  
With ActiveChart.Axes(xlCategory)  
    .TickLabelSpacing = tamanho  
    .TickMarkSpacing = tamanho  
    .TickLabels.NumberFormat = "0"  
End With
```

```
Selection.TickLabels.AutoScaleFont = True  
With Selection.TickLabels.Font  
    .Size = 8.5  
End With  
With Selection.TickLabels  
    .Alignment = xlCenter  
    .Offset = 100  
    .Orientation = xlHorizontal
```

```
End With
With Selection
    .TickLabelPosition = xlLow
End With
```

```
ActiveChart.SeriesCollection(1).Select
With Selection.Border
    .ColorIndex = 3
    .Weight = xlMedium
    .LineStyle = xlContinuous
End With
With Selection
    .MarkerStyle = xlNone
End With
```

```
Unload Me
```

```
End Sub
```

```
Function fx1(t As Single, x As Single, y As Single) As Single
fx1 = x - x * y
End Function
```

```
Function fx2(t As Single, x As Single, y As Single) As Single
fx2 = y + 2 * x * y
End Function
```