Universidade de São Paulo

IAG - USP

Projeto de Cálculo Numérico

Resolução de E.D.O’s pelos métodos de Euler e Runge-Kutta e análise do erro associado a cada método empregado.

Prof. Dr. Alexandre M. Roma

São Paulo
2015

Universidade de São Paulo

IAG - USP

Projeto de Cálculo Numérico

Resolução de E.D.O’s pelos métodos de Euler e Runge-Kutta e análise do erro associado a cada método empregado.

Fábio Oliveira 7978417

Jessé Stenico 9051932

São Paulo
2015

Sumário

1.Introdução 4
2.Resumo 5
3.Motivação 6
4.Discretização do Domínio 7
5.Métodos numéricos empregados 8
5.a.Método de Euler Explícito 8
5.b.Método de Euler Implícito 9
5.c.Ordem de precisão dos métodos de Euler Implícito e Explícito 10
5.d.Ordem do erro para método de Euler 11
5.e.Comentário sobre a Ordem do método 12
5.f.Método de Runge-Kutta de quarta ordem (RK4) 13
5.f.1Discretização do Método Numérico de RK4 14
6.Apresentação do Problema 15
7.Considerações a respeito do código-fonte 17
8.Resultados 18
8.a. Valores para a solução analítica 18
8.b.Para o método de Euler 18
8.c.Para método de Euler implícito 19
8.d.Para método de Runke-Kutta 21
9.Verificação de implementação do Método (teste de convergência) 23
9.a.Teste de convergência para o método de Euler Explícito 23
9.b.Teste de convergência para o método de Euler Implícito 24
9.c.Teste de convergência para o método de Runge-Kutta 25
10. Análise da Velocidade de Convergência dos métodos apresentados à Solução Analítica. .. 27
10.a.Métodos de Interpolação (Spline) 27
10.b.Condições para implementação e cálculo do Spline. 28
11.Verificação do objetivo principal (alteração da função) 29
11.a.Resultados 29
12.Conclusão 30
13.Referências bibliográficas 31
14.Apêndice 31

1.Introdução

Uma ciência exata é qualquer campo da ciência capaz de expressões quantitativas, predições precisas e métodos rigorosos de testar hipóteses, especialmente os experimentos reprodutíveis envolvendo predições e medições quantificáveis. Podemos citar como exemplo Matemática, Física, Engenharia, Química, Estatística e Computação.

As ciências exatas nasceram no início do século XVII na Europa Ocidental. Baseiam-se na observação aprofundada dentro de um quadro temático restrito ou previamente definido, uma abordagem simples e progressiva da modelização e, sobretudo, do uso sistemático de uma lógica reducionista no sentido de manter apenas os dados e leis necessários e suficientes para explicar os fenômenos observados.

No que se diz respeito à matemática, é a ciência do raciocínio lógico e abstrato, que estuda quantidades, medidas, espaços, estruturas, variações e estatísticas. Um trabalho matemático consiste em procurar por padrões, formular conjecturas e, por meio de deduções rigorosas a partir de axiomas e definições, estabelecer novos resultados.

Em meio a esses estudos podemos citar métodos que foram criados e adaptados para resoluções de problemas que envolvem, além de fenômenos físicos e matemáticos, também computacionais. Dentre eles podemos citar os estudos e aplicações das Equações Diferenciais Ordinárias.

As E.D.O.s possuem diversas aplicações, dentre elas: Decaimento Radioativo, Crescimento Populacional: Malthus (1766-1834), Crescimento Populacional: Verhulst (Modelo Logistico ou modelo de Verhulst-Pearl), Lei do resfriamento de Newton, Circuitos Elétricos RLC[4]. Sendo assim, neste projeto serão exemplificados e discretizados métodos numéricos, explícitos e implícitos para resolução das E.D.O.s.

2.Resumo

Este trabalho tem como objetivo a resolução numérica de diferentes E.D.O.s de primeiro grau, com a utilização de métodos numéricos ministrados no Curso de Cálculo Numérico- MAP0214 - pelo Prof. Dr. Alexandre Megiorin Roma. Foram empregados métodos de Euler Explícito, Euler Implícito e Runge-Kutta (4,4) visando observar as diferenças no que se diz respeito à velocidade de convergência, ordem e precisão. Foram também utilizados recursos gráficos com o objetivo de visualizar a convergência dos métodos quando comparados à solução analítica.

3.Motivação

Os Cursos de Bacharelado em Geofísica e Meteorologia possuem uma base muito forte em cálculo, física e programação. É natural que um estudante da área de exatas se depare com equações e fórmulas descrevendo fenômenos físicos e muitas vezes essas equações não possuem solução analítica, como é o caso de determinadas E.D.O.s utilizadas para tal descrição.

Assim, dada a funcionalidade das Equações Diferenciais Ordinárias, muito frequentemente utilizadas para descrever processos nos quais a mudança de uma medida ou dimensão é causada pelo próprio processo, o motivo principal deste projeto é encontrar soluções aproximadas para qualquer que seja a E.D.O. (de primeiro grau) apresentada e minimizar o esforço que o estudante (ou pesquisador) dedicaria na resolução dessas equações por outros meios, além de estimular o conhecimento em programação.

4.Discretização do Domínio

Trata-se de se tomar um intervalo contínuo $I = [t_f - t_0]$qualquer e amostrá-lo com uma certa densidade de pontos. É importante notar que há infinitos pontos em um intervalo contínuo mesmo que finito. Assim, se I é o domínio de uma função contínua $f(t)$, torna-se impraticável seu cálculo em cada ponto de seu domínio, por dois principais motivos: i). dados dois pontos consecutivos t_k e t_{k+1}, a precisão ε de qualquer máquina é tal que $|t_k - t_{k+1}| < \varepsilon$, sendo impossível representar todos os pontos do intervalo em questão; ii). a imagem dessa função teria infinitos pontos mesmo que finita.

Como o mesmo modo de discretização foi utilizado nos três métodos estudados neste trabalho, a fim de evitar redundâncias, explicamos aqui o modo empregado.

Supondo que I sejao intervalo de interesse para o ajuste numérico, podemos definir o incremento h para cada iteraçãoda seguinte maneira (discretização uniforme):

$$h = \frac{t_f - t_0}{n}, \qquad (4.0)$$

onde$n$ é o número total de passos desejado para o ajuste. Tomamos como um novo intervalo, agora discreto, o conjunto de pontos definidos pela expressão

$$t_k = t_0 + hk, \qquad t_0 < t_k \leq t_f \qquad (4.1)$$

onde $0 \leq k < n$ é um inteiro e diz respeito à iteração.

Temos, assim, um intervalo contínuo $[t_f - t_0]$ agorasubdividido em k partes de mesmo comprimento h. Há outras maneiras de se discretizar um intervalo. No entanto, essa foi a empregada neste trabalho. Assim, quando posteriormente neste trabalho nos referirmos a discretização, estaremos nos referindo a essa apresentada aqui.

5.Métodos numéricos empregados

Os métodos numéricos utilizados a seguir baseiam-se na expansão em Série de Taylor da função $y(x)$ em torno do ponto x, ou seja:

$$y\,(x+h) = y(x) + \; hf\big(x, y(x)\big) + \frac{h^2}{2!} f_1(x, y(x)) + \frac{h^3}{3!} f_2(x, (y(x)) + \ldots + \frac{h^n}{n!} f_{n-1}(x, (y(x))$$

, sendo $f_n(x, (y(x))$ a n-ésima derivada da função $f\big(x, y(x)\big)$ (5.0)

Para estes métodos, o cálculo da aproximação de y_{k+1} depende somente de y_k , x_k e h, e por isso são normalmente chamados de “métodos de passo simples”.

5.a.Método de Euler Explícito

É um método numérico de primeira ordem para solucionar uma equação diferencial ordinária com valor inicial $y_0$conhecido. O método de Euler Explícito é dado através de uma aproximação pelos dois primeiros termos (aproximação linear) da expansão em Série de Taylor:

$$y_0 = y(x_0) \quad \text{(5.a.1) (Condição de Contorno)}$$

$$y\,(x+h) = y(x) + \; hf\big(x, y(x)\big) \quad \text{(5.a.2)}$$

e sua discretização é discutida logo abaixo.

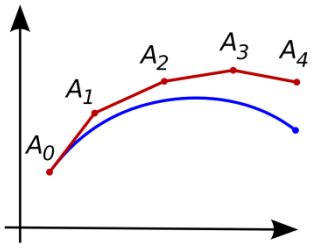

Figura 5.a.f1 - Método de Euler Explícito - A curva da expressão analítica está em azul e sua aproximação polinomial está em vermelho.

Sendo uma E.D.O. do tipo

$$y'(t) = f(t,(y(t)) \quad \text{(5.a.3)}$$

fazemos uso da equação (5.a.2) e a discretizamos (ver tópico 4). Temos, então, o algoritmo para Euler Explícito:

$$h \leftarrow \frac{t_f - t_0}{n}$$

$$y_k \leftarrow y_0$$

$$t_k \leftarrow t_0$$

$$para\ k = 0\ até\ n-1:$$

$$\begin{cases} y_{k+1} \leftarrow y_k + hf(t_k, y_k) \\ t_k \leftarrow t_0 + hk \\ k \leftarrow k+1 \end{cases}$$

(Algoritmo 5.a.A – Euler Explícito)

Nota: Em uma situação ideal, sem erros associados à máquina ou às operações matemáticas, temos que quanto mais fina a discretização (menor h) e maior o número de passos de integraçãok, melhor aproximada é a função. No entanto, em uma situação real, contamos com erros que são oriundos dos processos de mudança de base e a precisão limitada inerente à máquina. Nesse caso, vale ressaltar que valores muito pequenos de Δt podem exceder o limite inferior de precisão do computador e, ao contrário do desejado, adicionar ruídos e valores sem significado ao ajuste, condenando todo o processo. Essa situação se aplica a todos os métodos empregados neste trabalho.

5.b.Método de Euler Implícito

Análogo ao método de Euler Explícito, é um método numérico de primeira ordem para solucionar equações diferenciais ordinárias. No entanto, busca calcular o valor de y_{k+1} ,ou seja, o passo à frente do passo atual de integração. Assim como o método de Euler Explícito, o método de Euler Implícito também é de primeira ordem.

Considerando ainda a E.D.O. (5.a.3) e $y_0 = y(t_0)$, sendo (t_0, y_0)valores conhecidos, o método de Euler Implícito é expressado da seguinte maneira:

$$y_{k+1} \leftarrow y_k + hf(t_{k+1}, y_{k+1}) \qquad (5.b.1)$$

$$h \leftarrow \frac{t_f - t_0}{n}$$

$$y_k \leftarrow y_0$$

$$t_k \leftarrow t_0$$

$$para\ k = 0\ até\ n - 1:$$

$$\begin{cases} y_{k+1} \leftarrow y_k + hf(t_{k+1}, y_{k+1}) \\ t_k \leftarrow t_0 + hk \\ k \leftarrow k + 1 \end{cases}$$

(Algoritmo 5.b.A – Euler Implícito)

Note que, diferentemente do Euler Explícito, aqui se empregaf(t_{k+1}, y_{k+1}) em vez de $f(t_k, y_k)$. Por esse motivo o método é implícito: para calcularmos o valor de y_{k+1}, aparentemente, precisamos conhecer o valor do próprio y_{k+1}. Para contornar essa situação, utilizamos a aproximação de Newton-Rhapson, que será descrita posteriormente.

5.c.Ordem de precisão dos métodos de Euler Implícito e Explícito

Os dois métodos apresentados acima têm a mesma ordem de precisão, descrita da seguinte forma:

Chamaremos por A e B as soluções analítica e numérica, respectivamente, da E.D.O. em questão. Ainda, denotaremos por $y(t)$ a solução analítica no ponto t e y_t a solução numérica nesse mesmo ponto. Temos que:

$$A \equiv y(t_{k+1}) = y(t_k) + hy'(t_k) + \frac{h^2}{2!}y''(\zeta_k) \quad (5.c.1)$$

$$B \equiv y_{k+1} = y_k + hf(t_k, y_k) \quad (5.c.2)$$

A diferença $A - B$ entre os dois resultados será:

$$y(t_{k+1}) - y_{k+1} = y(t_k) - y_k + h[f(t_k, y(t_k)) - f(t_k, y_k)] + \frac{h^2}{2!}y''(\zeta_t) \quad (5.c.3)$$

Podemos ainda fazer as seguintes definições:

$$y(t_{k+1}) - y_{k+1} = e_{k+1} \quad (5.c.4)$$

$$y(t_k) - y_k = e_k \quad (5.c.5)$$

Temos então, a partir de (5.c.3):

$$e_{k+1} = e_k + h\left[\frac{\partial f}{\partial x}(t_{k,}\theta_k)[y(t_k) - y_k]\right] + \frac{h^2}{2!}y''(\zeta_k) \quad (2.c.6)$$

sendo $\frac{\partial f}{\partial x}(t_{k,}\theta_k) \approx \lambda = cte$, a equação acima passa a ser

$$e_{k+1} = e_k + h[\lambda e_k] + \frac{h^2}{2!}y''(\zeta_k) \rightarrow$$

$$\rightarrow e_{k+1} = e_k(1 + h\lambda) + \frac{h^2}{2!}y''(\zeta_k) \quad (5.c.7)$$

onde $e_k(1 + h\lambda)$ é o coeficiente de amplificação.

5.d.Ordem do erro para método de Euler

Métodos numéricos nos dão soluções aproximadas e trazem em si certo erro, seja ele por conta do computador ou dos próprios dados. Assim, devemos levar em conta nesses processos sua propagação. Utilizaremos aqui a mesma notação adotada no tópico anterior.

Considere a seguinte equação:

$$e_m = y(t_m) - y_m \quad (5.d.1)$$

onde e_mé o erro para um ponto t_m.Temos, para as três primeiras iterações ($k = 0,1,2$):

$$k = 0 \ : e_1 = (1 + \lambda h)e_0 + y''(\zeta_0)h^2$$

$$k = 1 \ : e_2 = (1 + \lambda h)^2 e_0 + (1 + \lambda h)y''(\zeta_0)h^2 + \frac{y'''(\zeta_1)}{2!}h^2$$

$$k = 2 \ : e_3 = (1 + \lambda h)^3 e_0 + (1 + \lambda h)y''(\zeta_0)h^2 + (1 + \lambda h)\frac{y'''(\zeta_1)}{2!}h^2 + \frac{y'''(\zeta_2)}{2!}h^2$$

Para um k genérico:

$$|e_k| \leq |1 + \lambda h|^k e_0 + \left(1 + |1 + \lambda h| + |1 + \lambda h^2| + \cdots + |1 + \lambda h|^{k-1}\right) * \max_{(to,tf)} |y''| \frac{h^2}{2!}$$

Resolvendo:

$$|e_k \leq |1 + \lambda h|^k |e_0 + \frac{|1 - e^{|\lambda k \Delta t|}|}{|\lambda|} * \frac{\Delta t}{2} * \max_{(to,tf)} |\ddot{y}| \qquad (5.d.2)$$

Assim, temos que o erro do método de Euler é da ordem de h, ou seja,

$$|e_k| = O(h) \ ,$$

para h< $h_0 \ll 1$, existe $c > 0$, tal que

$$0 < h < h_k \ \rightarrow |e_k| \ll ch$$

5.e.Comentário sobre a Ordem do método

A ordem de um método mede o quanto rapidamente este converge para a solução analítica quando se diminui os passos na integração numérica. Infelizmente, devido a limitações computacionais, erros de arredondamento crescem quando se diminui o tamanho dos passos, ocorrendo até mesmo divergência ou mesmo valores errados. Uma forma de resolver este problema é aumentar a ordem do método numérico.

Com isso, o próximo método numérico a ser abordado é o Método de Runge-Kuta de 4ª ordem (Runge-Kutta 4x4, ou RK4), que é um método de ordem maior, ou seja, espera-se que a resolução numérica convirja para solução analítica mais rapidamente e com menores erros.

5.f.Método de Runge-Kutta de quarta ordem (RK4)

O método de Runge-Kutta é provavelmente um dos métodos mais populares. O método de Runge-Kutta de quarta ordem também é um dos mais preciosos para obter soluções aproximadas de um valor inicial.

Cada método de Runge-Kutta consiste em comparar um polinômio de Taylor apropriado para eliminar o cálculo das derivadas.

Fazendo-se várias avaliações da função f a cada passo. Estes métodos podem ser construídos para qualquer ordem

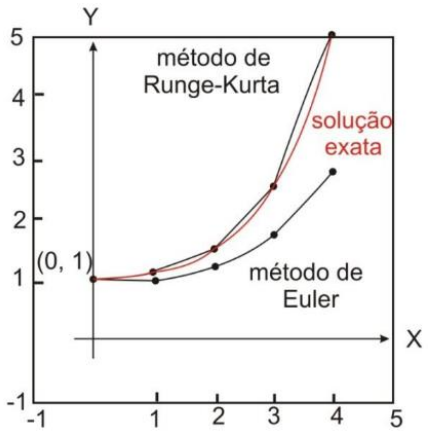

Figura 5.f.f1-Método de Runge-Kutta. Observa-se uma melhor aproximação à solução analítica.

O RK4 é um método iterativo implícito e explícito para a resolução numérica da E.D.O, segue várias etapas, estas que podem ser generalizadas da seguinte maneira

$$u^{n+1} = u^n + h\sum_{i=1}^{e} b_i k_i$$

onde,

$$k_i = F(u^n + \Delta t \sum_{j=1}^{e} a_{ij} k_j, t_n + c_i \Delta t) \quad , i = 1, \dots, e$$

com a_i, b_i, c_iconstantes próprias do método numérico. Os métodos Runge-Kutta podem ser explícitos ou implícitos dependendo das constantes a_{ij} do

esquema. Se esta matriz é triangular inferior com todos os elementos da diagonal principal iguais a zero; quer dizer, $a_{ij} = 0$ para $j = 1, \dots, e$, os esquemas são explícitos.

Os métodos de RK de ordem m fornecem valores aproximados da EDO ($\frac{dx}{dt} = -\lambda x$) que coincidem com valores obtidos através da expressão em série de Taylor de y, em torno do ponto x, até o termo que inclui h^m., ou seja,

$$y\,(x+h) = y(x) + \; hf\big(x, y(x)\big) + \frac{h^2}{2!} f'\big(x, y(x)\big) + \cdots + f^{(m-1)}(x, y(x)) \frac{h^m}{m!}$$

Nestes métodos o cálculo do valor das derivadas da função $f(x, y(x))$ nos pontos (x_k, y_k) será substituído pelo cálculo da função $f(x, y)$ em pontos convenientes, produzindo resultados equivalentes.

5.f.1Discretização do Método Numérico de RK4

A família de métodos Runge–Kutta explícitos é uma generalização do método RK4 mencionado acima.

Ela é dada por

$$y_{k+1} = y_k + h \sum_{i=1}^{n} b_n \mathrm{k}_n \qquad \text{(5.f*.1)}$$

Assim,

$$\begin{cases} k = f(t_k, y_k) \\ \mathrm{k}_2 = f(t_k + \frac{h}{2}, y_k + \frac{h}{2}\mathrm{k}_1 \\ \mathrm{k}_3 = f(t_k + \frac{h}{2}, y_k + \frac{h}{2}\mathrm{k}_2) \\ \mathrm{k}_4 = f\left(t_k + \frac{h}{2}, y_k + \frac{h}{2}\mathrm{k}_3\right) \\ \downarrow \\ y_{k+1} = y_k + h[\mathrm{k}_1 + 2\mathrm{k}_3 + 2\mathrm{k}_3 + \mathrm{k}_4] \\ \downarrow \\ t_{k+1} = t_k + h \\ \_ \\ \circ \end{cases}$$

(Algoritmo 5.f.A – Runge-Kutta)

6.Apresentação do Problema

Primeiramente, antes de elaborar um código para resolver uma E.D.O, foi escolhida uma equação de solução analítica já conhecida, e seu comportamento gráfico. Com isso, pode-se "criar" o código a partir dos resultados dessa solução. A E.D.O escolhida foi de decaimento radioativo.

Numa substância radioativa, cada átomo tem uma certa probabilidade, por unidade de tempo de se transformar num átomo mais leve emitindo radiação nuclear no processo. Se λ representa essa probabilidade, o número médio de átomos que se transmutam, por unidade de tempo, é $\lambda X'$, em que X' é o número de átomos existentes em cada instante. O número de átomos transmutados por unidade de tempo é também igual a menos a derivada temporal da função x'.

$$\frac{dx\prime}{dt} = -\lambda x' \quad \text{(6.a.0)}$$

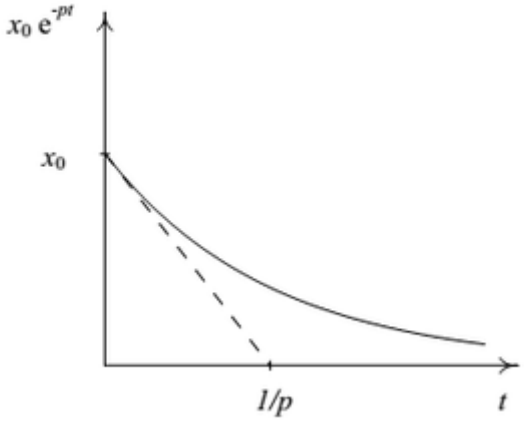

Figura 6.f1–Decaimento exponencial de uma substância radioativa.....

com , $\mathbf{p} = \boldsymbol{\lambda}$

A massa dos correspondentes átomos, x, é diretamente proporcional a x' assim obtemos a seguinte equação diferencial

$$\frac{dx}{dt} = -\lambda x \qquad (6.a.1)$$

onde λé uma constante, designada de constante de decaimento. A solução geral desta equação é uma função que diminui exponencialmente até zero

$$x = Ce^{-\lambda t}$$

e a solução única para a condição inicial $x = x_0$no instante inicial é (figura 3.f1)

$$x = x_0 e^{-\lambda t}$$

Quanto maior for a constante de decaimento λ, mais rápido diminuirá a massa da substância.

Com isso podemos apresentar meios que utilizamos para resolver a EDO acima apresentada, sua solução analítica e computacional, juntamente com resultados obtidos.

7.Considerações a respeito do código-fonte

O projeto foi implementado em C, sendo dividido em dois arquivos a fim de se evitar a escrita de um código-fonte poluído e de difícil entendimento. A saber, os arquivos são: i) *main.c* e ii) *funcoes.h.*

O arquivo “main.c”, como o nome sugere, é o arquivo principal e que deve ser compilado e executado para uso do programa. Nele, estão declaradas as chamadas principais de funções das bibliotecas padrão do C bem como as funções declaradas no cabeçalho “*funcoes.h*” (ii).

O usuário deve ficar atento ao fato de que a E.D.O. de interesse deverá ser declarada no cabeçalho do arquivo “*main.c” (apêndice Figura 7.f.1)*. Vale ressaltar ainda que os dois arquivos citados acima devem estar no mesmo diretório para a compilação do programa.

8.Resultados

8.a. Valores para a solução analítica

Após “rodar” o código, obtivemos os seguintes valores para a solução analítica.

Tabela 1 - Valores encontrados para solução analítica através do código		
Número de passos	Solução analítica	
	Valores de x_{tf}	Valores de f(x_{tf})
2	5.0000000000	0.0820849986
4	5.0000000000	0.0820849986
8	5.0000000000	0.0820849986
16	5.0000000000	0.0820849986
32	5.0000000000	0.0820849986
64	5.0000000000	0.0820849986
128	5.0000000000	0.0820849986
256	5.0000000000	0.0820849986
512	5.0000000000	0.0820849986
1024	5.0000000000	0.0820849986
2048	5.0000000000	0.0820849986

Os resultados acima foram coerentes com o esperado. Mesmo com um número pequeno ou grande de passos de integração, a solução analítica vai devolver o mesmo valor do último passo, pois é uma solução exata. O número de passos só faz com que os valores entre t_0 e t_f sejam calculados em pequenos Δt's.

8.b.Para o método de Euler

Porém, se tabelarmos (tabela 2) os valores obtidos através dos métodos numéricos, observamos uma discrepância entre estes e o valor da solução analítica em t_f .

Tabela 2 - Valores encontrados com a implementação do método numérico		
Número de passos	Euler explícito	
	Valores de x_{tf}	Valores de f(x_{tf})
2	5.000000000	0.0625000000
4	5.0000000000	0.0197753906
8	5.0000000000	0.0499093162
16	5.0000000000	0.0659812552
32	5.0000000000	0.0740466142
64	5.0000000000	0.0780708800
128	5.0000000000	0.0800793962
256	5.0000000000	0.0810825838
512	5.0000000000	0.0815838905
1024	5.0000000000	0.0818344697
2048	5.0000000000	0.0819597405

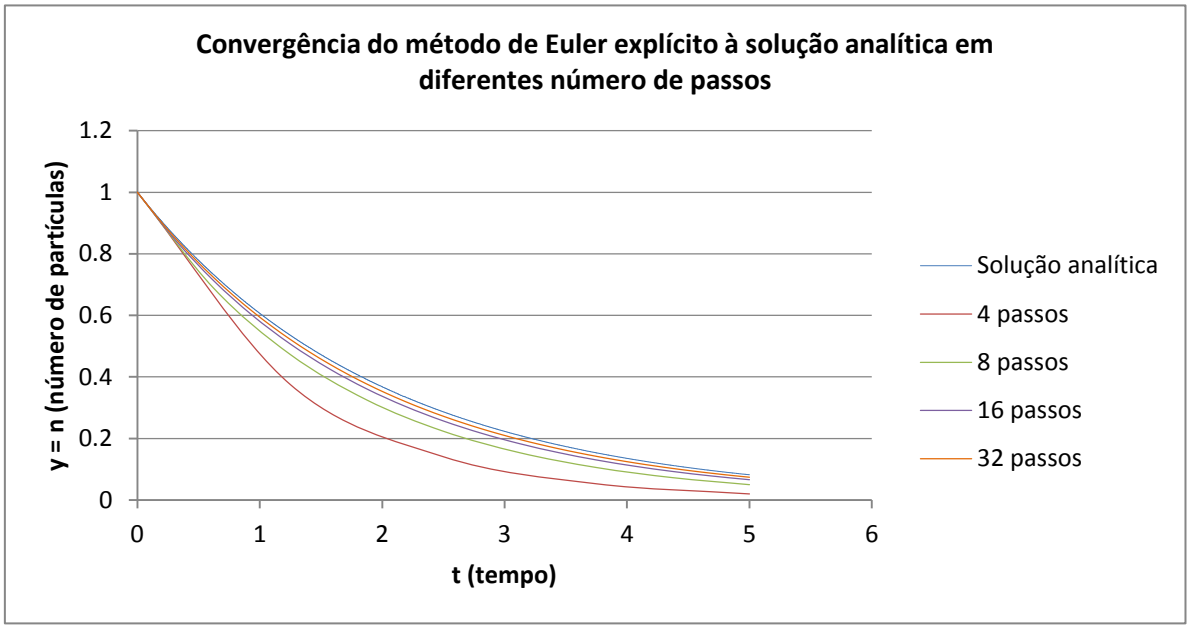

Gráfico 1- ***Convergência do método de Euler explícito à soluçao analítica. Observa-se que, com 4 passos, a solução é bem mais distante do que com 32 passos.***

8.c.Para método de Euler implícito

O método de Euler Implícito converge mais rapidamente ara solução analítica, apesar do erro ser da $O(h)$ de Δt. A explicação disso vem da forma como o próximo passo (y_{k+1}) é calculado, ou seja, o valor de y_{k+1} é encontrado através da melhor aproximação da raiz no ponto x_{k+1}. Assim, o resultado de $f(x)$ é mais "preciso" para este método numérico.

Vejamos abaixo (tabela 3) os valores para o Método de Euler Implícito.

Tabela 3 - Valores encontrados com a implementação do método numérico		
Número de passos	Euler implícito	
	Valores de x_{tf}	Valores de f(x_{tf})
2	5.00000	0.1975308642
4	5.00000	0.1434123455
8	5.00000	0.1135548017
16	5.00000	0.0979878138
32	5.00000	0.0900716801
64	5.00000	0.0860861005
128	5.00000	0.0840873412
256	5.00000	0.0830865981
512	5.00000	0.0825859029
1024	5.00000	0.0823354766
2048	5.00000	0.9339637196

Também podemos comparar graficamente os resultados de Euler explicito e implícito com a solução analítica.

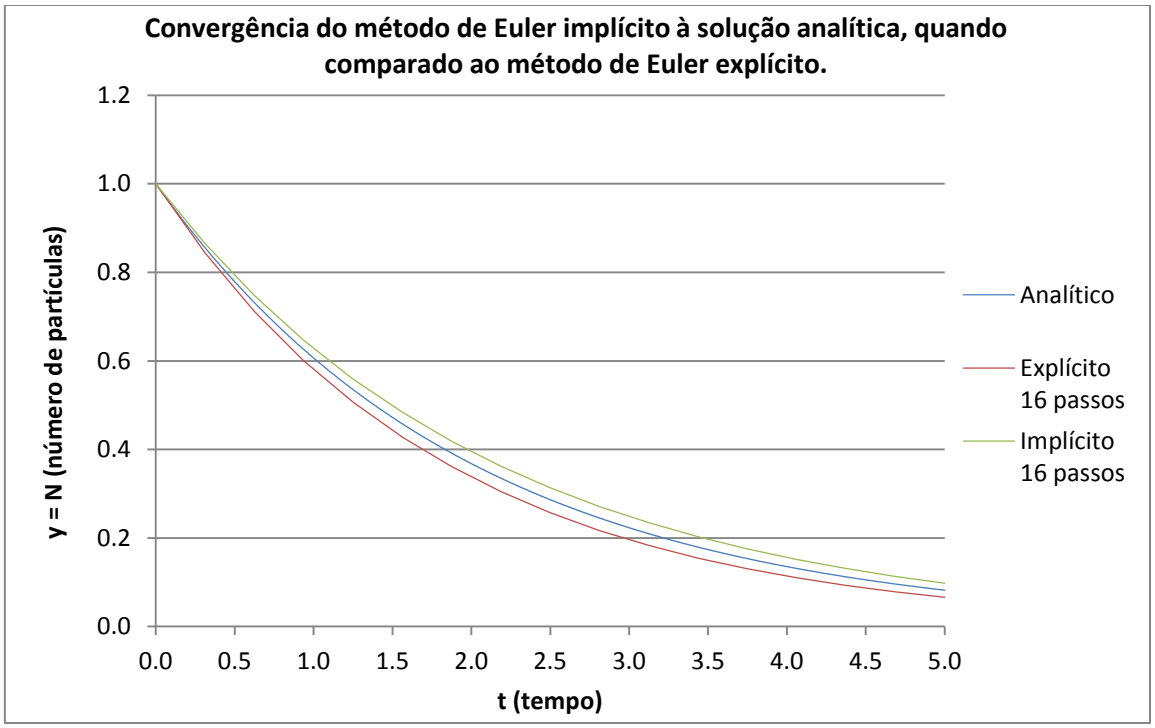

Gráfico 2 - Convergência do método de Euler explícito e implícito à solução analítica. Observa-se que o método de Euler implícito converge mais rapidamente para solução do que o Euler Explícito, para uma mesma quantidade de passos.

8.d.Para método de Runke-Kutta

O Método de Runge-Kutta converge ainda mais rapidamente para solução analítica. Apesar de ser um método explícito, o erro $O(h^4)$, ou seja, há necessidade de muito menos passos de integração para convergir para solução analítica, comparado com métodos de Euler.

Na tabela abaixo podemos observar (tabela 4) que o valor de $f(x)$ e t_f é bem próximo da solução analítica.

Tabela 4 - Valores encontrados com a implementação do método numérico

Número de passos	Runge-Kutta 4ª Ordem (Rk4)	
	Valores de x_{tf}	Valores de f(x_{tf})
2	5.00000000	0.0945282247
4	5.000000000	0.0825268114
8	5.000000000	0.0821061814
16	5.000000000	0.0820861600
32	5.000000000	0.0820850666
64	5.000000000	0.0820850027
128	5.000000000	0.0820849989
256	5.000000000	0.0820849986
512	5.000000000	0.0820849986
1024	5.000000000	0.0820849986
2048	5.000000000	0.0820849986

Novamente, pode-se observar graficamente como a velocidade de convergência do método de método de RK4 é bem mais rápida que os demais métodos de Euler até agora abordados.

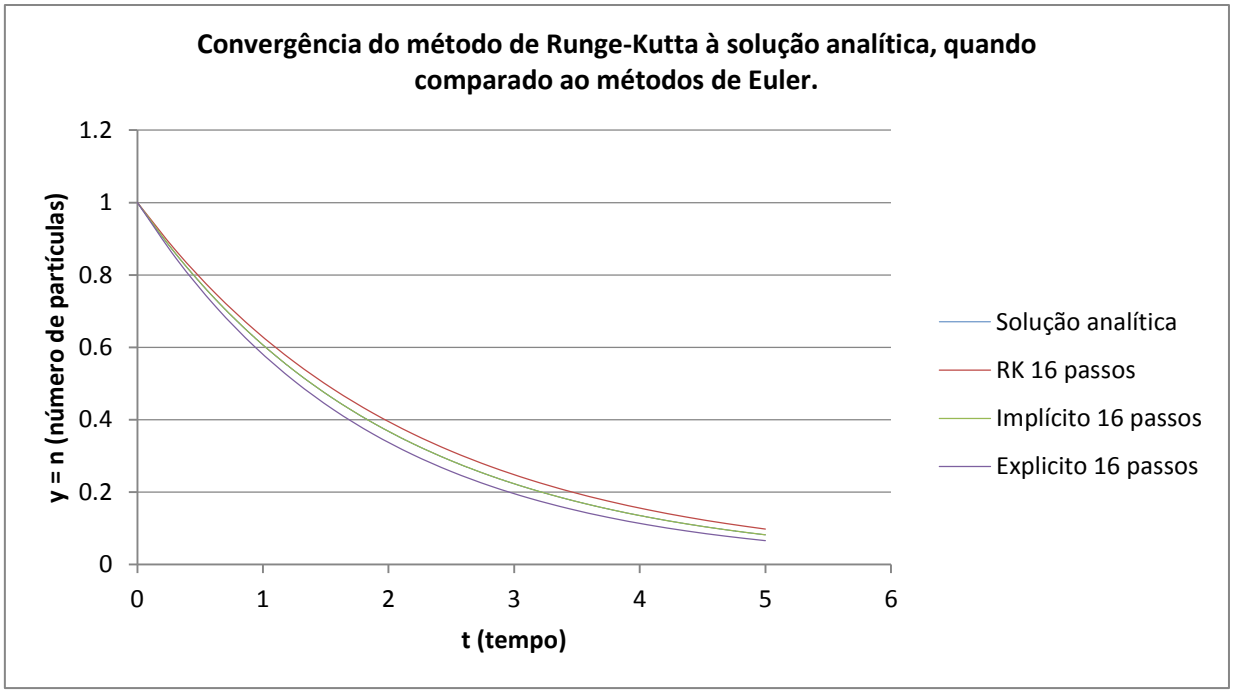

Gráfico 3 - Convergência do método de Euler explícito, implícito e Runge-Kutta à solução analítica. Observa-se que o método de Runge-Kutta converge mais rapidamente para solução (sobposta à linha do RK4) do que os métodos de Euler, para uma mesma quantidade de passos.

9.Verificação de implementação do Método (teste de convergência)

Como o objetivo deste projeto é a resolução de qualquer EDO de primeiro grau, temos que ter a certeza de que o código implementado esteja correto.

Dada uma EDO de solução conhecida, é natural através da visualização gráfica saber se está convergindo conforme o esperado. Porém nada se pode afirmar quanto à velocidade de convergência. Também, em muitos casos, o gráfico da solução analítica não é conhecido, sendo assim, temos que utilizar artifícios matemáticos para ter certeza da convergência do método.

Assim, para verificação do código, temos que o erro cometido por aproximar uma função φ de um$x$ numérico é dado por:

$$\varphi(h, t_f) \approx x(t_f) \rightarrow e(h, t_k) = x(t_f) - \varphi(h, t_f) \quad (9.1)$$

Para verificação da velocidade de convergência, temos que o valor da divisão (5.a.1) de $e(h, t_k)$ em 2^k passos sobre 2^{k+1} deve ser 2^p, sendo p a ordem do método.

$$\frac{2^k\ passos}{2^{k+1}\ passos} = 2^p \quad (9.2)$$

9.a.Teste de convergência para o método de Euler Explícito

Sabendo que o método de Euler Explícito é de Ordem 1, logo, o valor da divisão dos erros deve convergir para 2. Observa-se claramente essa convergência nos instantes iniciais de integração (Tabela 5).

Tabela 5 – Teste de convergência para o método de Euler Explícito	
Número de passos	Método de Euler Explícito
	Valores dos erros
$\frac{e_2}{e_4}\ passos$	0.3143174745
$\frac{e_4}{e_8}\ passos$	1.9365434818
$\frac{e_8}{e_{16}}\ passos$	1.9980250325
$\frac{e_{16}}{e_{64}}\ passos$	2.0033557145
$\frac{e_{64}}{e_{128}}\ passos$	2.0025278708
$\frac{e_{128}}{e_{256}}\ passos$	2.0014527759
$\frac{e_{256}}{e_{512}}\ passos$	2.0007709332
$\frac{e_{512}}{e_{1024}}\ passos$	2.0003962864
$\frac{e_{1024}}{e_{2048}}\ passos$	2.0002008095

Podemos observar claramente uma convergência para 2.

Sendo assim, está verificada a implementação do método de Euler Explícito.

9.b.Teste de convergência para o método de Euler Implícito

Sabendo que o método de Euler implícito é de Ordem 1, logo, o valor da divisão dos erros deve convergir para 2. Observa-se claramente essa convergência nos instantes iniciais de integração (Tabela 6).

Tabela 6 - Teste de convergência para o método de Euler Implícito	
Número de passos	Método de Euler implícito
	Valores dos erros
$\frac{e_2}{e_4}\ passos$	1.8824532846
$\frac{e_4}{e_8}\ passos$	1.9487680509
$\frac{e_8}{e_{16}}\ passos$	1.9788825351
$\frac{e_{16}}{e_{64}}\ passos$	1.9911668054
$\frac{e_{32}}{e_{64}}\ passos$	1.9961205269
$\frac{e_{64}}{e_{128}}\ passos$	1.9982104233
$\frac{e_{128}}{e_{256}}\ passos$	1.9991449351
$\frac{e_{256}}{e_{512}}\ passos$	1.9995826850
$\frac{e_{512}}{e_{1024}}\ passos$	1.9997939336
$\frac{e_{1024}}{e_{2048}}\ passos$	1.9998976192

Podemos observar claramente uma convergência para 2.

Sendo assim, está verificada a implementação do método de Euler implícito.

9.c.Teste de convergência para o método de Runge-Kutta

Sabendo que o método de Runge-Kutta é de Ordem 4, logo, o valor da divisão dos erros deve convergir para 16. Observa-se claramente essa convergência nos instantes iniciais de integração (Tabela 7).

Tabela 7 - Teste de convergência para o método de Runge-Kutta	
Número de passos	Método de Runge-Kutta
	Valores dos erros
$\frac{e_2}{e_4}\ passos$	28.1640258127
$\frac{e_4}{e_8}\ passos$	20.8571776088
$\frac{e_8}{e_{16}}\ passos$	18.2399707526
$\frac{e_{16}}{e_{64}}\ passos$	17.0795004959
$\frac{e_{64}}{e_{128}}\ passos$	16.5301564698
$\frac{e_{128}}{e_{256}}\ passos$	16.2627366180
$\frac{e_{256}}{e_{512}}\ passos$	16.1307332710
$\frac{e_{512}}{e_{1024}}\ passos$	16.0670041522
$\frac{e_{1024}}{e_{2048}}\ passos$	16.0447182295

Podemos observar claramente uma convergência para 16.

Sendo assim, está verificada a implementação do método de Runge-Kutta.

10. Análise da Velocidade de Convergência dos métodos apresentados à Solução Analítica.

Métodos de maior ordem convergem mais rapidamente para solução analítica, esta velocidade de convergência pode ser observada abaixo (tabela 8):

Tabela 8 - Análise da Velocidade de Convergência dos métodos apresentados à Solução Analítica.

Número de passos	Métodos numéricos e solução analítica			
	Valores de f(x_{tf}) para Solução analítica discretizada	Valores de f(x_{tf}) para Euler Explícito	Valores de f(x_{tf}) para Euler implícito	Valores de f(x_{tf}) para Runge-Kutta
2	0.0820849986	0.0625000000	0.1975308642	0.0945282247
4	0.0820849986	0.0197753906	0.1434123455	0.0825268114
8	0.0820849986	0.0499093162	0.1135548017	0.0821061814
16	0.0820849986	0.0659812552	0.0979878138	0.0820861600
32	0.0820849986	0.0740466142	0.0900716801	0.0820850666
64	0.0820849986	0.0780708800	0.0860861005	0.0820850027
128	0.0820849986	0.0800793962	0.0840873412	0.0820849989
256	0.0820849986	0.0810825838	0.0830865981	0.0820849986
512	0.0820849986	0.0815838905	0.0825859029	0.0820849986
1024	0.0820849986	0.0818344697	0.0823354766	0.0820849986
2048	0.0820849986	0.0819597405	0.0822102440	0.0820849986

Esses dados também podem ser utilizados em uma interpolação do tipo Spline. A seguir descrevemos suas principais condições para implementação e verificação

10.a.Métodos de Interpolação (Spline)

É uma técnica de aproximação que consiste em se dividir o intervalo de interesse em vários subintervalos e interpolar, da forma mais suave possível, nestes subintervalos com polinômios de grau pequeno.

Sejam uma subdivisão do intervalo $[a, b]$. Uma função spline de grau p com nós nos pontos (x_i, f_i) para $(i = 0,\ m - 1)$ é uma função $sp(x)$ com as propriedades:

a) em cada subintervalo (x_i, x_{i+1}) $(i = 0, m-1)$, $sp(x)$ é um polinômio de grau p.

b) $sp(x)$ é contínuo em $[a, b]$ e tem derivada contínua em $[a, b]$ até ordem p.

A spline interpolante é a função $sp(x)$ tal que $sp(x_i) = f(x_i)$ $(i = 0, m)$.

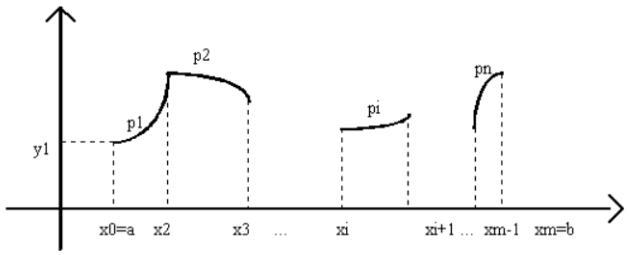

Figura 11 - O spline calcula uma função cúbida a cada intervalo formado por 2 pontos.

10.b.Condições para implementação e cálculo do Spline.

- $S(x), s'(x) e\ s"(x)$ são funções contínuas no intervalo $[x_i, x_{i+1}]$;
- $S(x)$ é um polinômio cúbico tal que $S(x_i) = f_i = f(x_i)$, $i = 1,2,3, \dots, n$;

Para garantir continuidade e suavidade:

- $S_i(x_i) = f_i$;
- $S_i(x_{i+1}) = f_{i+1}$ (continuidade de $s's$) ;
- $S'_i(x_{i+1}) = S'_{i+1}(x_{i+1})$;
- $S"_i(x_{i+1}) = S"_{i+1}(x_{1+1})$;

É composto por $n-1$ polinômios cúbicos, onde encontramos os coeficientes $a_i, b_i, c_i\ e\ d_i$ que serão encontrados através da resolução de um sistema, este que pode ser resolvido através da implementação do Algoritmo de Thomas, Gauss-Seidel, por exemplo.

11.Verificação do objetivo principal (alteração da função)

Apesar da verificação da implementação dos métodos numéricos (capítulo 9), há necessidade de alterar a EDO (6.a.1) e analisá-la, assim, cumpre-se o objetivo principal deste trabalho: resolução de qualquer EDO de primeiro grau através dos métodos de Euler e Runge-Kutta. Alterando a EDO (6.a.1) para

$$y' = sin(t) - y \qquad (11.0)$$

E a solução analítica,

$$y = \lambda \cdot e^{-t} + \frac{1}{2} \cdot sin(t) - \frac{1}{2} \cdot cos(t) \quad (11.1)$$

E resolvê-la pelos mesmos métodos acima descritos.

11.a.Resultados

Tabela 9 – Teste de convergência dos métodos implementados, agora com a EDO (11.0) a ser analisada.			
-	RK 4	Explícito	Implícito
$\frac{e_{64}}{e_{128}}$ passos	15.9171016328	1.9934058177	1.9408779016
$\frac{e_{128}}{e_{256}}$ passos	15.9655065021	1.9968350952	1.9703138022
$\frac{e_{256}}{e_{512}}$ passos	15.9844354819	1.9984497020	1.9851287796
$\frac{e_{512}}{e_{1024}}$ passos	15.9910700656	1.9992327826	1.9925570432

Tabela 9 – Nota-se claramente que o RK 4 converge para 16, e os demais métodos de Euler para 2, conforme prescrito na literatura..

12.Conclusão

Após análise dos resultados obtidos, nota-se ao utilizar quantidades de passos pequenas, há uma melhor aproximação do método numérico à solução analítica, porém dependente da ordem do método. Métodos de maior ordem, como por exemplo, RK4, convergem mais rapidamente para solução do que métodos de 1ª ordem. Como se observa no problema inicial, o valor encontrado para solução analítica discretizada com 1024 passos foi 0.0820849986, o que para Euler Explícito, este de 1ª ordem, foi 0.0819597405, e no caso do RK4, o valor encontrado com a mesma quantidade de passos foi de 0.0820849986, o que comprova uma maior velocidade de convergência.

Para verificação da implementação do código foi utilizado o teste de convergência (capítulo.9). Neste caso, métodos de Euler explícito e implícito convergiram para 2, enquanto RK4 convergiu para 16, o que está de acordo com a literatura.

13.Referências bibliográficas

https://pt.wikipedia.org/wiki/Ci%C3%AAncias_exatas

https://pt.wikipedia.org/wiki/Matem%C3%A1tica

http://www.mat.uc.pt/~jaimecs/uaberta/eqdif.html

Burden, Richard., Faires, Douglas.,2001.Numerical Analysis, Books/Cole, Boston, EUA.

Humes,M.,Yoshida,M.,1984.Noções de Cálculo Numérico,2ª Edição, McGraw-Hill,São Paulo.

http://www.mat.ufmg.br/~espec/Monografias_Noturna/Monografia_KarineNayara.pdf (acesso 04/11/2015 14h33)

http://www2.ic.uff.br/~aconci/splineatual.html

14.Apêndice

```
onde "Lambda" depende do material radioativo em questão.
*/
#include <stdio.h>
#include <math.h>

double SOL_ANALITICA (double t){

    return exp(-0.5*t);
}

double F_XY(double t, double y){

    return -0.5*y;
}

double dFdY(double t, double y){

    return -0.5;
}
```

Figura 7.f.1 **– O usuário deve alterar apenas o *return* das funções SOL_ANALITICA, F_XY e dFdY para qual EDO de primeiro grau desejar resolver.**