

Capítulo 9 – Resolução Numérica de Equações Diferenciais Ordinárias

9.1 – Introdução:

No Capítulo 3 abordamos a solução numérica de uma equação **algébrica** $f(x)=0$ envolvendo uma única variável desconhecida x . Aqui trataremos de soluções numéricas de outros tipos de equações, que ampliam a capacidade de modelagem de fenômenos.

Definição 1 - Uma **equação diferencial** é toda equação que envolve uma, ou mais, derivada(s) de uma função desconhecida $y=y(x)$. Se nesta equação estiverem envolvidas derivadas ordinárias até a ordem n , ela é denominada de equação diferencial ordinária de ordem n , denotada pela sigla EDO, e genericamente representada por: **equação diferencial**

$$y^{(n)} = f(x, y, y', \dots, y^{(n-1)}), \text{ onde } y = y(x) \text{ e } x \in [a, b] \quad (1)$$

Definição 2- **Solução de uma EDO** de ordem n é toda função $y = y(x)$, definida em $[a, b]$, que possui as n derivadas ordinárias neste domínio e que satisfaz à equação diferencial (1).

Como uma EDO pode ter mais de uma solução, por exemplo, a equação $y' = x^2$ possui a família de soluções $y(x) = (x^3 / 3) + C$, onde C é uma constante, temos que impor condições adicionais para obter numericamente uma solução específica. Tais condições complementares podem ser de dois tipos:

i). Na eq. (1) impondo que $y(a) = u_1$, $y'(a) = u_2$, ..., $y^{(n-1)}(a) = u_n$ (condições em um único ponto $x = a$) a EDO é denominada de **Problema de Valor Inicial - PVI**. Por exemplo o PVI $y' = -y * \operatorname{tg}(x)$, com $y(0) = 1$ e $x \in [0, \pi]$ tem solução $y(x) = \cos(x)$;

ii). Se além de condições do tipo (i) também forem impostas condições na outra extremidade do intervalo, $y(b) = v_1$, $y'(b) = v_2$, ..., $y^{(n-1)}(b) = v_n$ a EDO é denominada

de Problema de Valor no Contorno - PVC.

Equações Diferenciais Ordinárias são usadas com muita frequência na modelação de fenômenos da natureza, como a taxa de variação da posição x de um móvel em função do tempo t e de sua aceleração a . No caso de aceleração a constante, temos o movimento uniformemente variado, amplamente conhecido das disciplinas introdutórias de Física, mas que também pode ser modelado com o seguinte PVI de 1ª ordem:

$$\frac{d^2x(t)}{dt^2} = a, \text{ onde } a = \text{aceleração do móvel.}$$

Condições iniciais:

$$\frac{dx(t=0)}{dt} = v_0, \text{ onde } v_0 = \text{valor da sua velocidade inicial.}$$

$$x(t=0) = x_0, \text{ onde } x_0 = \text{valor da sua posição inicial.}$$

Neste caso obtendo a solução da EDO acima por integração analítica direta resulta:

$$\frac{d}{dt} \left(\frac{dx(t)}{dt} \right) = a \Rightarrow \int d \left(\frac{dx(t)}{dt} \right) = \int a * dt \Rightarrow \frac{dx(t)}{dt} = a * t + C_1$$

Integrando novamente teremos:

$$dx(t) = (a * t + C_1) dt \Rightarrow \int dx(t) = \int (a * t + C_1) dt \Rightarrow x(t) = a * t^2 / 2 + C_1 * t + C_2$$

Aplicando as condições iniciais $\frac{dx}{dt}(t=0) = v_0$ e $x(t=0) = x_0$, teremos a conhecida equação do movimento uniformemente variado:

$$x(t) = a * t^2 / 2 + v_0 * t + x_0$$

Existem vários métodos analíticos de resolução de equações diferenciais ordinárias, mas na maioria dos casos não é possível representar a solução em termos de funções elementares, como feito no exemplo anterior. Mesmo uma

equação diferencial com aspecto simples, como:

$$y' = x^2 + y^2$$

não pode ser resolvida em termos de funções elementares.

Nestes casos, os métodos numéricos permitem encontrar soluções discretas e aproximadas. Inicialmente, abordaremos alguns métodos numéricos para resolver PVI's de 1ª ordem com fundamentação teórica no desenvolvimento da solução $y = y(x)$ em série de Taylor. Por este motivo são chamados de métodos baseados na série de Taylor.

9.2. – Método de Euler Simples

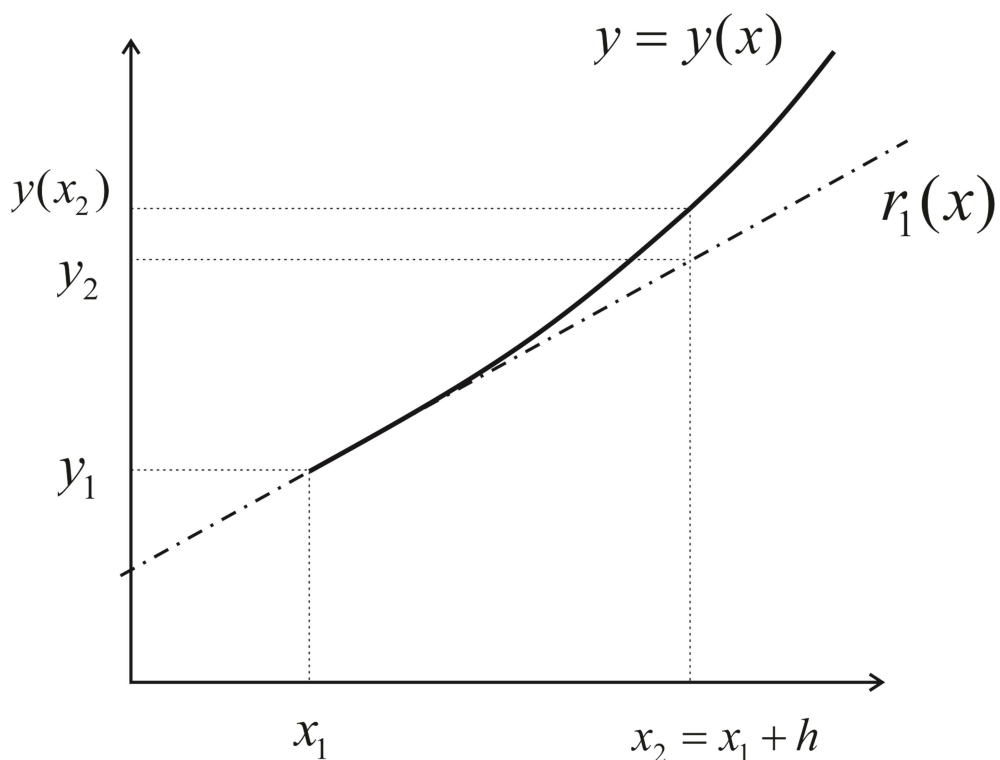
Seja um PVI de primeira ordem:

$$y' = f(x, y) \text{ com } x \in [a, b] \text{ e condição inicial } y(x_1) = y_1 \quad (2)$$

onde $f(x, y)$ representa a derivada da função solução $y(x)$.

Suponhamos que a solução exata da equação diferencial (2) seja dada por $y = y(x)$ conforme Gráfico 1 a seguir.

Gráfico 9.1 – Representação de uma solução $y = y(x)$ e da reta tangente $r_1(x)$ no ponto definido pela condição inicial $y(x_1) = y_1$.



Fonte: Elaboração própria

Como conhecemos o ponto inicial da solução desejada (x_1, y_1) , podemos obter o coeficiente angular $y'(x_1) = f(x_1, y_1)$ da reta $r_1(x)$ tangente à esta solução $y(x)$ neste ponto (x_1, y_1) , e assim obter a expressão algébrica desta reta:

$$r_1(x) = y_1 + (x - x_1)y'(x_1) = y_1 + (x - x_1)f(x_1, y_1) \quad (3)$$

Na Gráfico 9.1, considerando $h = (x_2 - x_1)$ como o espaçamento entre os dois primeiros valores da variável independente x da solução numérica desejada e aplicando na eq. (3) resulta:

$$y_2 = y_1 + (x_2 - x_1)f(x_1, y_1) = y_1 + h * f(x_1, y_1), \text{ onde } x_2 = x_1 + h.$$

Dividindo o domínio $[a, b]$ em n partes iguais, resulta o passo constante $h = (b - a) / n$ e generalizando a eq.(3), para todo $k = 1, 2, \dots, n$, temos:

$$y_{k+1} = y_k + h * f(x_k, y_k) \quad (4a)$$

$$x_{k+1} = x_k + h \quad (4b)$$

Assim podemos obter n aproximações na forma discreta da solução $y = y(x)$

do PVI. Este é o denominado **Método de Euler Simples**.

Note-se que $y_2 = y_1 + (x_2 - x_1)f(x_1, y_1) = y_1 + h * f(x_1, y_1)$ é exatamente o valor da solução $y(x)$, avaliada em $x = x_2$, desenvolvida em série de Taylor, a partir de $x = x_1$ e truncada a partir do terceiro termo.

Exemplo 9.1: Resolva por Euler Simples o Problema de Valor Inicial:

$$y' = x - y + 2, \text{ com } y(x=0) = 2 \text{ no domínio } [0, 1].$$

Inicie com $n = 8$ divisões do domínio e aumente o n até que o $y(x=1)$ tenha erro estimado da ordem de 10^{-6} .

Solução:

Temos que $f(x, y) = x - y + 2$ e $y(x=0) = 2$.

Tomando $n = 8$ para o número de subdivisões e aplicando as eqs. (4), resulta:

$$h = (1 - 0) / 8 = 0.125$$

$$x_1 = 0 \quad \text{e} \quad y_1 = 2$$

$$x_2 = x_1 + h \quad \text{e} \quad y_2 = y_1 + h * f(x_1, y_1)$$

$$x_2 = 0 + 0.125 = 0.125 \quad y_2 = 2 + 0.125 * (0 - 2 + 2) = 2$$

$$x_3 = x_2 + h \quad \text{e} \quad y_3 = y_2 + h * f(x_2, y_2)$$

$$x_3 = 0.125 + 0.125 = 0.250 \quad y_3 = 2 + 0.125 * (0.125 - 2 + 2) = 2.015625$$

Assim sucessivamente, resultando em:

ponto inicial			ponto novo		
k	x_k	y_k	$K_1 = f(x_k, y_k)$	x_{k+1}	y_{k+1}
1	0	2	0.000000000	0.125	2.000000000
2	0.12500	2.000000000	0.125000000	0.250	2.015625000
3	0.25000	2.015625000	0.234375000	0.375	2.044921875
4	0.37500	2.044921875	0.330078125	0.500	2.086181641
5	0.50000	2.086181641	0.413818359	0.625	2.137908936

6	0.62500	2.137908936	0.487091064	0.750	2.198795319
7	0.75000	2.198795319	0.551204681	0.875	2.267695904
8	0.87500	2.267695904	0.607304096	1.000	2.343608916
9	1.00000	2.343608916			

Portanto, para $n = 8$, $y(x=1) = y_9 \cong \mathbf{2.343608916}$.

Qual o erro associado ao último ponto da solução obtida?

Se repetirmos todo o cálculo com o dobro de sub-divisões, $n = 16$, teremos um valor aproximado $y(x=1)$ com erros de truncamento inferiores, que pode ser considerado um valor estimativo mais próximo do exato que o valor obtido com $n = 8$.

Tomando $n = 16$ divisões do domínio e aplicando as eqs. 4, temos:

$$h = (1 - 0) / 16 = 0.0625$$

$$x_2 = 0.0625 \quad \text{e} \quad y_2 = y_1 + h * f(x_1, y_1) = 2 + 0.0625 * (0 - 2 + 2) = 2$$

$$x_3 = 0.1250 \quad \text{e}$$

$$y_3 = y_2 + h * f(x_2, y_2) = 2 + 0.0625 * (0.0625 - 2 + 2) = 2.003906250$$

Assim sucessivamente, resultando em:

ponto inicial			ponto novo		
k	x_k	y_k	$K_1 = f(x_k, y_k)$	x_{k+1}	y_{k+1}
1	0	2	0.000000000	0.06250	2.000000000
2	0.06250	2.000000000	0.062500000	0.12500	2.003906250
3	0.12500	2.003906250	0.121093750	0.18750	2.011474609
4	0.18750	2.011474609	0.176025391	0.25000	2.022476196
5	0.25000	2.022476196	0.227523804	0.31250	2.036696434
6	0.31250	2.036696434	0.275803566	0.37500	2.053934157
7	0.37500	2.053934157	0.321065843	0.43750	2.074000772
8	0.43750	2.074000772	0.363499228	0.50000	2.096719474
9	0.50000	2.096719474	0.403280526	0.56250	2.121924507
10	0.56250	2.121924507	0.440575493	0.62500	2.149460475
11	0.62500	2.149460475	0.475539525	0.68750	2.179181695
12	0.68750	2.179181695	0.508318305	0.75000	2.210951589

13	0.75000	2.210951589	0.539048411	0.81250	2.244642115
14	0.81250	2.244642115	0.567857885	0.87500	2.280133233
15	0.87500	2.280133233	0.594866767	0.93750	2.317312406
16	0.93750	2.317312406	0.620187594	1.00000	2.356074130
17	1.00000	2.356074130			

Assim, temos que

$$y(x=1) \cong y_9^{(n=8)} = \mathbf{2.343608916} \quad (n=8)$$

$$y(x=1) \cong y_{17}^{(n=16)} = \mathbf{2.356074130} \quad (n=16) \text{ (estimativa do valor exato)}$$

$$\text{Erro estimado Euler} = \left| y_9^{(n=8)} - y_{17}^{(n=16)} \right| = 0.012465214 > 10^{-6}.$$

Que está muito distante ainda da precisão requerida. Por simulação em computador, teremos que aumentar o n até $n=2^{15}=32768$ para chegamos a um erro estimado $2.8068e-06$, da ordem de 10^{-6} .

A seguir apresentaremos outros métodos com maior precisão, para evitar o aumento exagerado no número n de subdivisões.

9.3 – Métodos de Runge-Kutta:

Os métodos de Runge-Kutta são obtidos tomando mais termos nas séries de Taylor para aproximar as soluções dos PVI's. Se forem cancelados na série os termos que contém derivadas de ordens maiores que p , o método é considerado de ordem p . O método de Euler Simples estudado anteriormente é de primeira ordem ($p=1$), porque despreza os termos com derivadas de ordem 2 e superiores. Na sequência apresentaremos alguns métodos desta categoria.

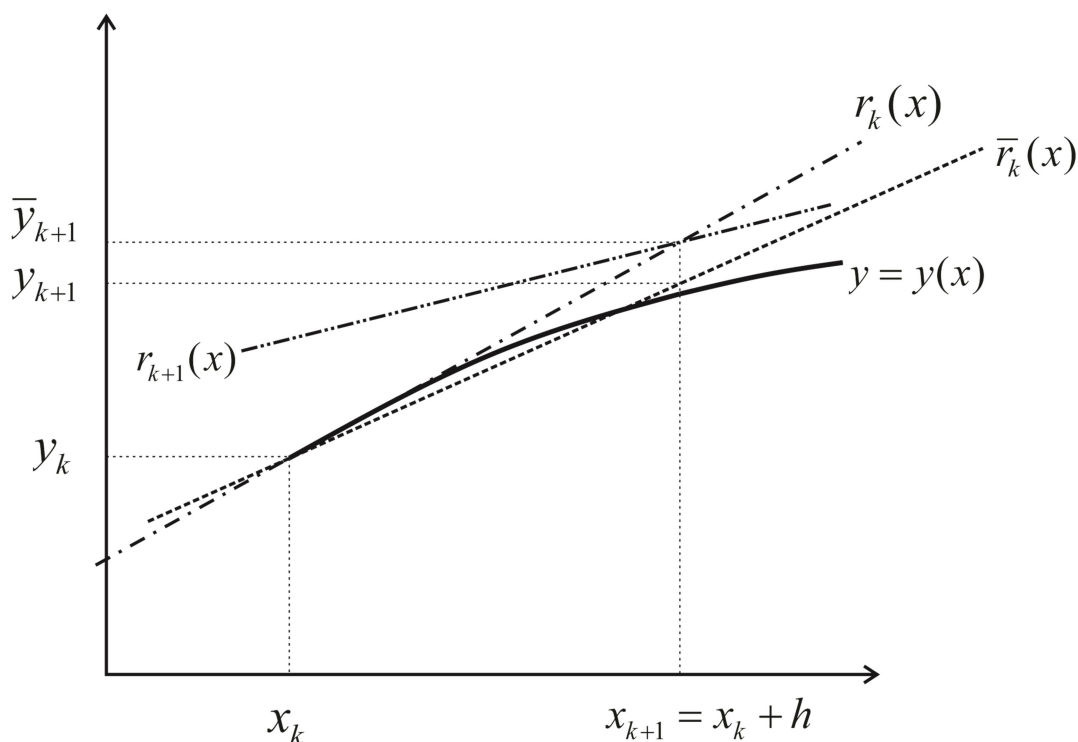
9.3.1. Método de Euler Aperfeiçoado

Exploramos inicialmente um método particular que é o método de Euler aperfeiçoado, pois ele tem uma interpretação geométrica bastante simples.

Conforme o próprio nome indica, este método consiste em fazer mudanças no método de Euler Simples para assim conseguir um método de ordem mais elevada.

Graficamente o Método de Euler Aperfeiçoado, pode ser interpretado conforme Gráfico 9.2 a seguir.

Gráfico 9.2 – Método de Euler Aperfeiçoado com inclinações representadas pelas retas $r_k(x)$ e $r_{k+1}(x)$.



Fonte: Elaboração própria

Dado o ponto (x_k, y_k) , supondo que a curva desenhada com linha cheia seja a solução exata $y = y(x)$ da nossa equação diferencial (as soluções numérica e exata só coincidem no ponto inicial (x_1, y_1)).

Por (x_k, y_k) traçamos a reta $r_k(x)$, cujo coeficiente angular é

$$y'(x_k) = f(x_k, y_k) \quad (6)$$

e

$$r_k(x) = y_k + (x - x_k)y'(x_k) = y_k + (x - x_k)f(x_k, y_k) \quad (7)$$

Assim, com o passo h do método de Euler, determinamos,

$$\bar{y}_{k+1} = y_k + h * f(x_k, y_k) \quad (8)$$

que chamamos aqui de \bar{y}_{k+1} .

Com os valores de x_{k+1} e de \bar{y}_{k+1} podemos determinar o coeficiente angular da reta $r_{k+1}(x)$ que passa pelo ponto (x_{k+1}, \bar{y}_{k+1}) , conforme segue

$$y'(x_{k+1}) = f(x_{k+1}, \bar{y}_{k+1}) \quad (9)$$

e a respectiva reta,

$$r_{k+1}(x) = y_{k+1} + (x - x_{k+1}) * f(x_{k+1}, \bar{y}_{k+1}) \quad (10)$$

Tomando como coeficiente angular a média aritmética simples entre os valores dados pelas eqs. (6) e (9), definimos uma nova reta, de inclinação média entre $r_k(x)$ e $r_{k+1}(x)$, e que passe pelo primeiro ponto (x_k, y_k) , conforme segue,

$$\bar{r}_k(x) = y_k + (x - x_k) * (f(x_k, y_k) + f(x_{k+1}, \bar{y}_{k+1})) / 2 \quad (11)$$

Assim, podemos recalcular o segundo ponto (x_{k+1}, y_{k+1}) , como

$$y_{k+1} = y_k + h * (f(x_k, y_k) + f(x_{k+1}, \bar{y}_{k+1})) / 2 \quad (12a)$$

ou usando apenas os valores do primeiro ponto (x_k, y_k) , temos

$$y_{k+1} = y_k + h * (f(x_k, y_k) + f(x_k + h, y_k + h * f(x_k, y_k))) / 2 \quad (12b)$$

O método de Euler aperfeiçoado consiste em obter o valor y_{k+1} conforme eqs. (12a) ou (12b).

Este método também é conhecido como método de Runge-Kutta de 2ª ordem (RK₂) e podemos comprovar que sua fórmula equivale à expansão da $y(x)$ em série de Taylor até o termo com derivada de 2ª ordem ($O(h^2)$). Tradicionalmente a eq. (12b) é expressa na literatura, para todo $k = 1, 2, \dots, n$ na seguinte forma:

$$y_{k+1} = y_k + h(K_1 + K_2)/2 \quad (13a)$$

$$x_{k+1} = x_k + h \quad (13b)$$

onde

$$K_1 = f(x_k, y_k) \quad (13c)$$

$$K_2 = f(x_k + h, y_k + h * K_1) \quad (13d)$$

Observe que a parcela $(K_1 + K_2)/2$ é uma inclinação média da função solução $y(x)$, considerando avaliações no ponto inicial e aproximações no ponto final do subintervalo $[x_k, x_{k+1}]$.

Exemplo 9.2 Resolva pelo Método de Runge-Kutta de 2ª ordem o PVI do Exemplo 9.1, isto é, $y' = x - y + 2$, com $y(x=0) = 2$ no domínio $[0, 1]$.

Inicie com $n = 8$ divisões do domínio e aumente o n até que o $y(x=1)$ tenha erro estimado da ordem de 10^{-6} .

Solução:

Temos que $f(x, y) = x - y + 2$ e $y(x=0) = 2$.

Tomando $n = 8$ para o número de subdivisões e aplicando as eqs. (13), resulta:

$$h = (1 - 0) / 8 = 0.125$$

ponto inicial				ponto novo		
k	x_k	y_k	$K_1 = f(x_k, y_k)$	$K_2 = f(x_k + h, y_k + h * K_1)$	x_{k+1}	y_{k+1}
1	0	2	0.000000000	0.125000000	0.125	2.007812500
2	0.125	2.007812500	0.117187500	0.227539063	0.250	2.029357910
3	0.250	2.029357910	0.220642090	0.318061829	0.375	2.063026905
4	0.375	2.063026905	0.311973095	0.397976458	0.500	2.107398752
5	0.500	2.107398752	0.392601248	0.468526092	0.625	2.161219211
6	0.625	2.161219211	0.463780789	0.530808190	0.750	2.223381022
7	0.750	2.223381022	0.526618978	0.585791606	0.875	2.292906684
8	0.875	2.292906684	0.582093316	0.634331652	1.000	2.368933244
9	1.000	2.368933244				

Agora com $n = 16$, $h = (1-0)/16 = 0.0625$ as eqs. (13) fornecem:

ponto inicial			ponto novo			
k	x_k	y_k	$K_1 = f(x_k, y_k)$	$K_2 = f(x_k + h, y_k + h * K_1)$	x_{k+1}	y_{k+1}
1	0	2	0.000000000	0.062500000	0.063	2.001953125
2	0.063	2.001953125	0.060546875	0.119262695	0.125	2.007572174
3	0.125	2.007572174	0.117427826	0.172588587	0.188	2.016635187
4	0.188	2.016635187	0.170864813	0.222685762	0.250	2.028933642
5	0.250	2.028933642	0.221066358	0.269749710	0.313	2.044271645
6	0.313	2.044271645	0.268228355	0.313964083	0.375	2.062465158
7	0.375	2.062465158	0.312534842	0.355501414	0.438	2.083341291
8	0.438	2.083341291	0.354158709	0.394523789	0.500	2.106737619
9	0.500	2.106737619	0.393262381	0.431183482	0.563	2.132501553
10	0.563	2.132501553	0.429998447	0.465623544	0.625	2.160489740
11	0.625	2.160489740	0.464510260	0.497978369	0.688	2.190567509
12	0.688	2.190567509	0.496932491	0.528374210	0.750	2.222608344
13	0.750	2.222608344	0.527391656	0.556929678	0.813	2.256493386
14	0.813	2.256493386	0.556006614	0.583756201	0.875	2.292110974
15	0.875	2.292110974	0.582889026	0.608958462	0.938	2.329356208
16	0.938	2.329356208	0.608143792	0.632634805	1.000	2.368130539
17	1.000	2.368130539				

$$y(x=1) \cong y_9^{(n=8)} = \mathbf{2.368933244} \quad (n=8)$$

$$y(x=1) \cong y_{17}^{(n=16)} = \mathbf{2.368130539} \quad (n=16) \text{ (estimativa do valor exato)}$$

$$\text{Erro estimado RK}_2 = \left| y_9^{(n=8)} - y_{17}^{(n=16)} \right| = 8.0271e-04 > 10^{-6}$$

Portanto a exatidão almejada ainda não foi obtida com $n = 8$, apesar de a precisão ser bem melhor que a do método de Euler Simples. Por simulação em computador, teremos que aumentar n somente até $n = 2^7 = 128$ para chegamos a um erro estimado $2.8260e-06$ que é da ordem de 10^{-6} , mas vamos buscar um método ainda mais preciso.

9.3.2. Método de Runge-Kutta de 4ª ordem:

Aqui o truncamento na série de Taylor representativa da solução $y(x)$ ocorre a partir do termo de 5ª ordem ($O(h^5)$) e conduz a múltiplas possibilidades de expressões, mas a mais popular, $\forall k=1,2,\dots,n$, é a seguinte:

$$y_{k+1} = y_k + h(K_1 + 2K_2 + 2K_3 + K_4)/6 \quad (14a)$$

$$x_{k+1} = x_k + h \quad (14b)$$

onde

$$K_1 = f(x_k, y_k) \quad (14c)$$

$$K_2 = f(x_k + h/2, y_k + (h/2)K_1) \quad (14d)$$

$$K_3 = f(x_k + h/2, y_k + (h/2)K_2) \quad (14e)$$

$$K_4 = f(x_k + h, y_k + hK_3) \quad (14f)$$

Observe que o termo $(K_1 + 2K_2 + 2K_3 + K_4)/6$ é uma inclinação média ponderada da função solução $y(x)$, considerando uma avaliação no ponto inicial $K_1 = f(x_k, y_k)$, duas no ponto intermediário, $K_2 = f(x_k + h/2, y_k + (h/2)K_1)$ e $K_3 = f(x_k + h/2, y_k + (h/2)K_2)$, e uma no ponto final $K_4 = f(x_k + h, y_k + hK_3)$ do subintervalo $[x_k, x_{k+1}]$.

Exemplo 9.3: Resolva o PVI do Exemplo 9.1 considerando agora o Método de Runge-Kutta de 4ª ordem, de modo que o $y(x=1)$ tenha erro estimado da ordem de 10^{-6} .

Solução:

Temos que $f(x, y) = x - y + 2$ com $y(x=0) = 2$.

Tomando inicialmente $n = 8$, $h = (1-0)/8 = 0.125$, para o número de subdivisões e aplicando as eqs. (14), resulta:

$$h = (1-0)/8 = 0.125$$

ponto inicial				ponto novo				
k	x_k	y_k	K_1	K_2	K_3	K_4	x_{k+1}	y_{k+1}
1	0	2	0.000000000	0.062500000	0.058593750	0.117675781	0.125	2.007497152
2	0.125	2.007497152	0.117502848	0.172658920	0.169211666	0.221351390	0.250	2.028801223
3	0.250	2.028801223	0.221198777	0.269873854	0.266831661	0.312844820	0.375	2.062289861
4	0.375	2.062289861	0.312710139	0.355665755	0.352981029	0.393587511	0.500	2.106531345
5	0.500	2.106531345	0.393468655	0.431376864	0.429007601	0.464842705	0.625	2.160262184
6	0.625	2.160262184	0.464737816	0.498191703	0.496100835	0.527725212	0.750	2.222367353
7	0.750	2.222367353	0.527632647	0.557155607	0.555310422	0.583218845	0.875	2.291862843
8	0.875	2.291862843	0.583137157	0.609191084	0.607562714	0.632191817	1.000	2.367880272
9	1.000	2.367880272						

Agora com $n = 16$, $h = (1 - 0) / 16 = 0.0625$ e reaplicando as eqs. (14), resulta:

ponto inicial				ponto novo				
k	x_k	y_k	K_1	K_2	K_3	K_4	x_{k+1}	y_{k+1}
1	0	2	0.000000000	0.031250000	0.030273438	0.060607910	0.0625	2.001913071
2	0.0625	2.001913071	0.060586929	0.089943588	0.089026192	0.117522792	0.1250	2.007496917
3	0.1250	2.007496917	0.117503083	0.145081111	0.144219298	0.170989377	0.1875	2.016529139
4	0.1875	2.016529139	0.170970861	0.196878022	0.196068423	0.221216585	0.2500	2.028800809
5	0.2500	2.028800809	0.221199191	0.245536716	0.244776168	0.268400680	0.3125	2.044115660
6	0.3125	2.044115660	0.268384340	0.291247330	0.290532861	0.312726037	0.3750	2.062289313
7	0.3750	2.062289313	0.312710687	0.334188478	0.333517297	0.354365856	0.4375	2.083148564
8	0.4375	2.083148564	0.354351436	0.374527953	0.373897437	0.393482846	0.5000	2.106530700
9	0.5000	2.106530700	0.393469300	0.412423384	0.411831069	0.430229858	0.5625	2.132282868
10	0.5625	2.132282868	0.430217132	0.448022847	0.447466418	0.464750481	0.6250	2.160261473
11	0.6250	2.160261473	0.464738527	0.481465448	0.480942731	0.497179606	0.6875	2.190331624
12	0.6875	2.190331624	0.497168376	0.512881864	0.512390817	0.527643950	0.7500	2.222366600
13	0.7500	2.222366600	0.527633400	0.542394856	0.541933561	0.556262552	0.8125	2.256247358
14	0.8125	2.256247358	0.556252642	0.570119747	0.569686400	0.583147242	0.8750	2.291862069
15	0.8750	2.291862069	0.583137931	0.596164871	0.595757779	0.608403070	0.9375	2.329105676
16	0.9375	2.329105676	0.608394324	0.620632002	0.620249574	0.632128726	1.0000	2.367879490
17	1.0000	2.367879490						

Os valores da tabela foram calculados com precisão *double*.

$$y(x=1) \cong y_9^{(n=8)} = \mathbf{2.367880272} \quad (n=8)$$

$$y(x=1) \cong y_{17}^{(n=16)} = \mathbf{2.367879490} \quad (n=16) \text{ (estimativa do valor exato)}$$

$$\text{Erro estimado RK}_4 = |y_9^{(n=8)} - y_{17}^{(n=16)}| = 7.8147e-07 < 10^{-6}$$

Logo, temos o problema proposto no exemplo 9.1 resolvido com a precisão desejada, Erro estimado $< 10^{-6}$, em apenas $n = 8$ divisões, usando o método de Runge-Kutta de 4ª ordem.

Apenas para efeito didático, tomando a solução analítica do PVI obtida através de fator integrante, $y(x) = (x+1) + e^{-x}$, conforme sessão Complementando...

Logo, o valor exato de y em $x=1$ é

$$y(x=1) = (1+1) + e^{-1} = 2.36787944117144$$

Erros exatos dos valores aproximados por Runge-Kutta de 4ª ordem:

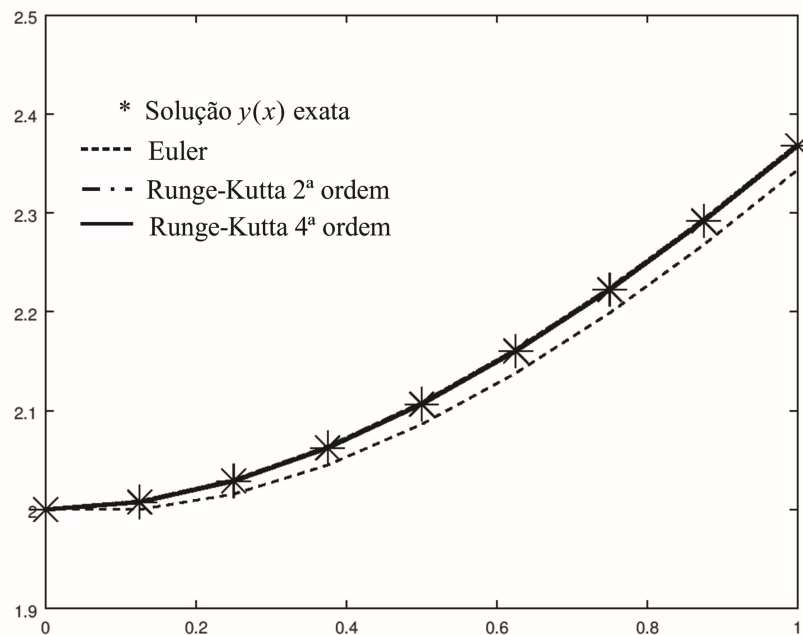
$$y(x=1) \cong y_9^{(n=8)} = 2.36788027192195 \Rightarrow \text{Erro exato } y_9^{(n=8)} = 8.3075 \cdot 10^{-7}$$

$$y(x=1) \cong y_{17}^{(n=16)} = 2.36787949045257 \Rightarrow \text{Erro exato } y_{17}^{(n=16)} = 4.9281 \cdot 10^{-8}$$

Observe que com $n=16$ temos um resultado mais preciso, pois o erro com $n=16$ é de ordem inferior ao erro de $n=8$.

Observe que com Método de Runge-Kutta de 4ª ordem, com $n=8$ subdivisões, temos em $y(x=1)$ o Erro estimado $\text{RK}_4 = 7.8147 \cdot 10^{-7}$ (obtido comparativamente com o resultado de $n=16$ subdivisões), enquanto o seu erro exato, comparando com a solução exata é $\text{Erro exato RK}_4 = 8.3075 \cdot 10^{-7}$, ou seja, o erro estimado é da mesma ordem do erro exato.

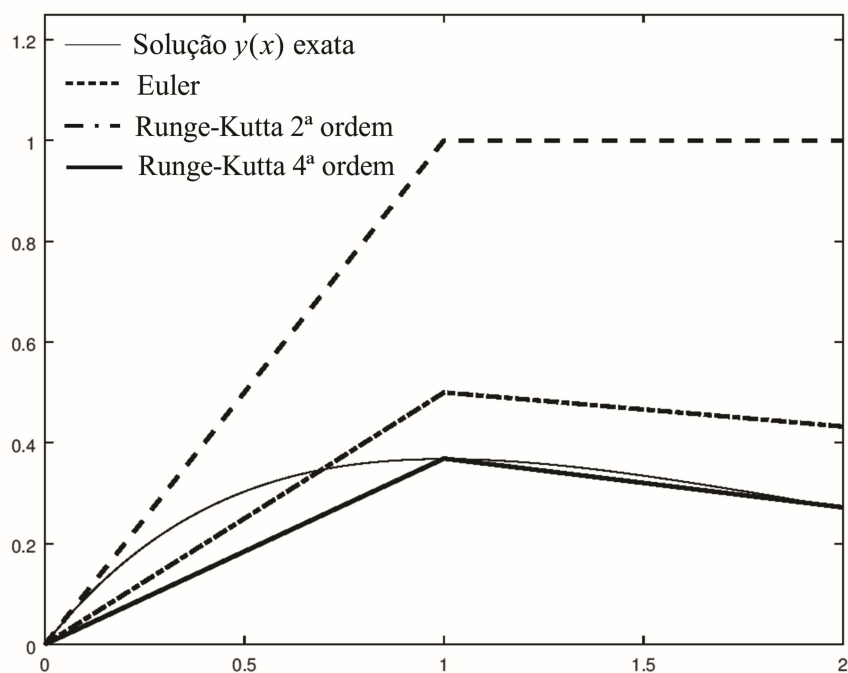
Gráfico 9.3 - Resultados aproximados obtidos executando os algoritmos Euler Simples, Runge-Kutta de 2ª e 4ª ordem para $n=8$ para o exemplo 9.1.



Fonte: Elaboração própria

DESTAQUE Para visualizar o comportamento de cada um dos métodos apresentados, usamos o PVI exemplo $y' = \exp(-x) - x \cdot \exp(-x)$, $y(x=0) = 0$ com $x \in [0, 2]$ (solução exata $y(x) = x \cdot \exp(-x)$), e passo $h = 1$, conforme Gráfico 9.4. Nesse exemplo, observamos uma redução crescente nos erros com as soluções de Euler, Runge-Kutta de 2ª ordem e de 4ª ordem, respectivamente. Percebe-se visualmente que o Método de Runge-Kutta de 4ª ordem gera praticamente os resultados exatos.

Gráfico 9.4 – Comparativo entre soluções aproximadas e a exata.



Fonte: Elaboração própria

FIM DESTAQUE.

Os Algoritmos de Euler Simples, Runge-Kutta de 2ª e 4ª ordem estão no Caderno de Algoritmos no link <http://sergiopeters.prof.ufsc.br/algoritmos-livro/> no arquivo **Cap9exem9.1,9.2,9.3EulerRungeKutta.m** (com cálculos dos erros de forma exata e estimada conforme exemplos 9.1, 9.2 e 9.3).

Considerações:

Sobre os métodos de Runge-Kutta de ordem p concluímos que se caracterizam pelas três propriedades a seguir:

- i) são de passo simples, ou seja, a sua aplicação é autosuficiente não necessitando de correções iterativas ou de outros métodos complementares;
- ii) não exigem o cálculo de qualquer derivada da função inclinação $f(x, y)$, mas precisam calcular $f(x, y)$ em vários pontos;
- iii) a expansão da função solução $y(x)$ diretamente por série de Taylor em torno de (x_k, y_k) , que envolve as derivadas da função derivada $f(x, y)$ (não apresentado neste livro), gera expressões de ordem equivalente as obtidas por Euler, Runge-Kutta de 2ª e 4ª Ordem.

Existem outras famílias de método numéricos de solução de PVI's, além dos fundamentados em série de Taylor apresentados neste livro. Por exemplo, para o PVI $y' = f(x, y)$ com $y(x_1) = y_1$, temos que:

$$y' = \frac{dy}{dx} = f(x, y) \Rightarrow dy = f(x, y)dx \Rightarrow \int_{x_k}^{x_{k+1}} dy = \int_{x_k}^{x_{k+1}} f(x, y)dx$$

Logo,

$$y_{k+1} = y_k + \int_{x_k}^{x_{k+1}} f(x, y)dx \quad (15)$$

Agora efetuando a integral $I = \int_{x_k}^{x_{k+1}} f(x, y)dx$ numericamente por Trapézios com

$n = 1$, por exemplo, resulta em:

$$I = \int_{x_k}^{x_{k+1}} f(x, y)dx \cong \frac{h}{2} (f(x_k, y_k) + f(x_{k+1}, y_{k+1}))$$

Que substituída na eq. (15) resulta em,

$$y_{k+1} = y_k + \frac{h}{2} (f(x_k, y_k) + f(x_{k+1}, y_{k+1})) \quad (16)$$

Assim, geramos um método de solução com uma característica completamente distinta dos já abordados, pois a incógnita y_{k+1} aparece em ambos os lados da eq. (16). Este tipo de método é denominado de multipassos, pois ele não é autoinicializável, isto é, temos que atribuir um valor inicial para o y_{k+1} da direita da eq. (16), conforme fizemos com os métodos iterativos dos capítulos 2 e 3. Na literatura o leitor encontrará uma família destes métodos desenvolvidos por Adams-Bashforth que excluem algebricamente os valores de y_{k+1} da direita da eq. (16). Por exemplo, o primeiro método desta família proveniente da eq. (16), baseado na extrapolação linear de y_{k+1} a partir dos dois pontos anteriores y_{k-1} e y_k , torna-se

$$y_{k+1} = y_k + \frac{h}{2} (3f(x_k, y_k) - f(x_{k-1}, y_{k-1})) \quad \forall k = 2, 3, \dots, n \quad (17)$$

com erro local da ordem de $O(h^3)$. A eq. (17) calcula uma solução aproximada a

partir de y_3 usando os dois pontos anteriores: o valor inicial y_1 conhecido e necessitando de um método complementar, de outra família, para gerar o valor de y_2 .

Exemplo 9.4: Resolva o mesmo PVI do Exemplo 9.1 pela eq.(17), com o y_2 gerado pela eq.(13) do Método de Runge-Kutta de 2ª ordem e com $n = 8$.

Solução:

Para $f(x, y) = x - y + 2$, $y(x = 0) = 2$, temos:

k	x_k	y_k
1	0	2
2	0.125	2.00781250000000
3	0.250	2.02978515625000
4	0.375	2.06375122070312
5	0.500	2.10834693908691
6	0.625	2.16232883930206
7	0.750	2.22460136562586
8	0.875	2.29419666202739
9	1.000	2.37025987324887

$$y(x = 1) \cong y_9^{(n=8)} = \mathbf{2.37025987324887} \quad (n = 8)$$

$$y(x = 1) \cong y_{17}^{(n=16)} = \mathbf{2.36847724634937} \quad (n = 16) \text{ (estimativa do valor exato)}$$

Erro estimado Adams Bashforth com $Tn = |y_9^{(n=8)} - y_{17}^{(n=16)}| = 0.00178262689949493$

Erro exato Adams Bashforth com $Tn = 0.00238043207742722$.

Observe que estes erros ficaram na mesma ordem de grandeza dos erros do método de Runge-Kutta de 2ª ordem. Porém, Adams-Bashforth com Trapézios utiliza uma única chamada na $f(x, y)$ em cada ponto a ser obtido, uma no ponto anterior (que pode ser reaproveitada para o próximo) e outra no início do subintervalo de cálculo, enquanto que o Runge-Kutta de 2ª ordem utiliza duas chamadas na $f(x, y)$ para cada valor a ser obtido.

O Algoritmo de Adams-Bashforth com integração por Trapézios está no Caderno de Algoritmos (<<http://sergiopeters.prof.ufsc.br/algoritmos-livro/>>) no arquivo

A seguir vamos abordar a solução numérica de EDO's de ordem superior, através da sua transformação em um sistema de EDO's de primeira ordem.

9.4. Solução numérica de sistemas de PVI's de 1ª ordem e de PVI's de ordem superior

Aqui vamos estender as expressões dos métodos de solução numérica de um PVI de 1ª ordem para a solução de um PVI de ordem n através da solução de um sistema de n EDO's de 1ª ordem. Sem perda de generalidade, vamos estender os métodos para os PVI's de ordem $n = 2$.

Definição 3: Um **sistema de duas EDO's** de 1ª ordem com $x \in [a, b]$ é toda expressão do tipo:

$$\begin{cases} y_1' = f_1(x, y_1, y_2) \\ y_2' = f_2(x, y_1, y_2) \end{cases} \quad (18)$$

Na eq. (18), se as condições iniciais $y_1(x_1 = a) = u_1$ e $y_2(x_1 = a) = u_2$ forem valores conhecidos, teremos um sistema de dois PVI's de 1ª ordem. Para resolver este sistema dividimos o domínio $x \in [a, b]$ em n partes, de comprimento $h = (b - a)/n$ por exemplo, e adaptando o método de Euler simples, eq. (8), para todo $k = 1, 2, \dots, n$, resulta,

$$\begin{cases} y_{1,k+1} = y_{1,k} + f_1(x_k, y_{1,k}, y_{2,k}) \\ y_{2,k+1} = y_{2,k} + f_2(x_k, y_{1,k}, y_{2,k}) \\ x_{k+1} = x_k + h \end{cases} \quad (19)$$

Adaptando o método de Runge-Kutta de 2ª ordem (eq.13) para resolver o sistema (18), resulta,

$$\begin{cases} y_{1,k+1} = y_{1,k} + 0.5h(K_1^{f_1} + K_2^{f_1}) \\ y_{2,k+1} = y_{2,k} + 0.5h(K_1^{f_2} + K_2^{f_2}) \\ x_{k+1} = x_k + h \end{cases} \quad (20a)$$

onde os valores de $K_1^{f_1}$ e $K_1^{f_2}$ devem ser calculados primeiro, para depois

calculamos $K_2^{f_1}$ e $K_2^{f_2}$ que dependem de $K_1^{f_1}$ e $K_1^{f_2}$:

$$\begin{aligned} K_1^{f_1} &= f_1(x_k, y_{1,k}, y_{2,k}) \\ K_1^{f_2} &= f_2(x_k, y_{1,k}, y_{2,k}) \\ K_2^{f_1} &= f_1(x_k + h, y_{1,k} + h * K_1^{f_1}, y_{2,k} + h * K_1^{f_2}) \\ K_2^{f_2} &= f_2(x_k + h, y_{1,k} + h * K_1^{f_1}, y_{2,k} + h * K_1^{f_2}). \end{aligned} \quad (20b)$$

Dáí, para resolver um PVI de segunda ordem:

$y'' = f(x, y, y')$ com $x \in [a, b]$, $y(x_1 = a) = u_1$ e $y'(x_1 = a) = u_2$ conhecidos,

basta tomar uma variável auxiliar $y_2 = y'$ e definir a solução como $y_1 = y$ (para manter a notação de sistema), de seguinte forma:

$$\begin{cases} y_1 = y \\ y_2 = y' \end{cases} \quad (21)$$

Derivando ambas as funções das eqs. (21), o PVI de 2ª ordem torna-se o sistema:

$$\begin{cases} y_1' = y' = y_2 \\ y_2' = y'' = f(x, y, y') = f(x, y_1, y_2) \end{cases} \quad (22)$$

Com $y_1(x_1 = a) = u_1$ e $y_2(x_1 = a) = u_2$ conhecidos. Podemos reescrever as eqs. (22), como:

$$\begin{cases} y_1' = f_1(x, y_1, y_2) = y_2 & \rightarrow y_1(x_1 = a) = u_1 \\ y_2' = f_2(x, y_1, y_2) = f(x, y_1, y_2) & \rightarrow y_2(x_1 = a) = u_2 \end{cases} \quad (23)$$

Temos justamente um sistema de duas EDO's de 1ª ordem, como a das eqs. (18), o qual pode ser resolvido numericamente pelas eqs. (19), ou (20), ou por outro método, como o Runge-Kutta de 4ª ordem.

A seguir vamos apresentar um exemplo de solução de PVI de 2ª ordem.

Exemplo 9.5: Determine a solução do PVI $y'' - 4y' + 4y = 8x^2 - 16x + 4$, com $x \in [0, 1]$, condições iniciais, $y(x = 0) = 0$ e $y'(x = 0) = 1$, e erro máximo menor que 10^{-6} .

Solução:

Com os métodos apresentados, podemos resolver apenas uma EDO de 1ª ordem por vez, então podemos transformar uma EDO de 2ª ordem em um sistema de 2 EDO's de 1ª ordem. Para EDO's de ordem 2, como neste exemplo, redefinimos $y(x)$ e uma variável auxiliar para $y'(x)$, conforme segue:

$$\begin{cases} y_1(x) = y(x) \\ y_2(x) = y'(x) \end{cases}$$

Derivamos as 2 variáveis auxiliares, $y_1(x)$ e $y_2(x)$, para gerar 2 equações diferenciais:

$$y_1(x) = y(x) \Rightarrow y_1'(x) = y'(x) = y_2(x)$$

$$y_1'(x) = z_1(x, y_1, y_2) = y_2(x)$$

$$y_1(x=0) = y(x=0) = 0 \text{ (condição inicial)}$$

$$y_2(x) = y'(x) \Rightarrow y_2'(x) = y''(x) = +4y' - 4y + 8x^2 - 16x + 4 \text{ (EDO original)}$$

$$y_2'(x) = z_2(x, y_1, y_2) = +4y_2 - 4y_1 + 8x^2 - 16x + 4$$

$$y_2(x=0) = y'(x=0) = 1 \text{ (condição inicial)}$$

Resultando em 2 EDOs de 1ª ordem, para $y_1(x)$ e $y_2(x)$, com 2 condições iniciais:

$$\begin{cases} y_1'(x) = z_1(x, y_1, y_2) = y_2, & \text{com } y_1(x=0) = y(x=0) = 0 \\ y_2'(x) = z_2(x, y_1, y_2) = +4y_2 - 4y_1 + 8x^2 - 16x + 4, & \text{com } y_2(x=0) = y'(x=0) = 1 \end{cases}$$

Então, resolvemos as 2 EDO's simultaneamente, pois a incógnita $y_1(x)$ depende de $y_2(x)$, e vice versa, conforme arquivo **Cap9exem9.5sist2PVI.m** do Caderno de Algoritmos (<<http://sergiopeters.prof.ufsc.br/algoritmos-livro/>>).

Resultados obtidos para $n = 43$, para que o erro máximo seja menor que 10^{-6} :

t	y_1	y_2
0.00000	0.00000	1.0000
0.02326	0.02544	1.1894
0.04651	0.05537	1.3856
0.06977	0.08995	1.5892
0.09302	0.12935	1.8007

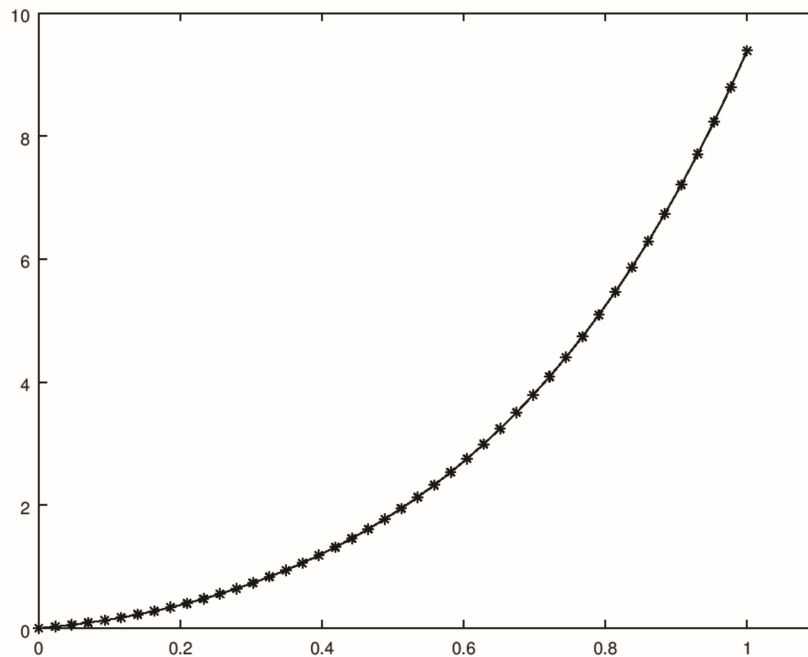
0.11628	0.17377	2.0204
0.13953	0.22339	2.2489
0.16279	0.27844	2.4869
0.18605	0.33914	2.7348
0.20930	0.40572	2.9933
0.23256	0.47845	3.2630
0.25581	0.55758	3.5447
0.27907	0.64341	3.8390
0.30233	0.73624	4.1468
0.32558	0.83640	4.4689
0.34884	0.94421	4.8061
0.37209	1.06006	5.1594
0.39535	1.18432	5.5298
0.41860	1.31740	5.9182
0.44186	1.45973	6.3258
0.46512	1.61178	6.7538
0.48837	1.77403	7.2033
0.51163	1.94700	7.6757
0.53488	2.13123	8.1723
0.55814	2.32730	8.6946
0.58140	2.53584	9.2440
0.60465	2.75748	9.8223
0.62791	2.99292	10.4311
0.65116	3.24290	11.0722
0.67442	3.50818	11.7476
0.69767	3.78958	12.4592
0.72093	4.08797	13.2093
0.74419	4.40428	13.9999
0.76744	4.73947	14.8336
0.79070	5.09457	15.7128
0.81395	5.47067	16.6403
0.83721	5.86893	17.6187
0.86047	6.29056	18.6511
0.88372	6.73687	19.7406
0.90698	7.20920	20.8905
0.93023	7.70901	22.1044
0.95349	8.23783	23.3858
0.97674	8.79728	24.7387
1.00000	9.38906	26.1672

Erro Exato máximo $y = y_1 = 9.28874230865517e-07$ (em todo $x \in [0, 1]$)

Erro Exato máximo $y = y_1 = 9.28874230865517e-07$ (em $x=1$)

Erro estimado máximo $y = y_1 = 8.69081802079563e-07$ (em $x=1$)

Gráfico 9.5 – Gráfico da solução discreta $y_1(x)$ obtida por Runge-Kutta de 4ª ordem marcados com * e a solução exata $y(x)$ em linha contínua.



Fonte: Elaboração própria

A seguir, apresentaremos uma introdução à solução numérica de problemas com valor no contorno (PVC's).

9.5 – Solução Numérica de EDO's com Valor no Contorno (PVC's):

Para uma EDO de ordem $n > 1$ se o valor da sua solução $y(x)$, ou de alguma(s) de suas derivadas até ordem $n-1$, for conhecido em mais de um ponto do domínio $[a,b]$, normalmente nos extremos deste intervalo, ela é denominada de Problema de Valor no Contorno – PVC e sua solução não pode ser obtida diretamente através dos métodos de resolução de PVI's. Por exemplo, para EDO de ordem $n = 2$, $y'' = f(x, y, y')$ com domínio $x \in [a, b]$, a especificação dos valores conhecidos nos extremos pode ser sintetizada pelo sistema:

$$\begin{cases} \alpha_1 y(a) + \beta_1 y'(a) = v_1 \\ \alpha_2 y(b) + \beta_2 y'(b) = v_2 \end{cases}$$

onde α_i , β_i e v_i com $i=1,2$, são constantes Reais conhecidas, bem como os α_i e

β_i não podem ser simultaneamente nulos.

Os métodos de solução de PVC's podem ser agrupados na categoria dos fundamentados nas diferenças finitas, nos quais as derivadas envolvidas na EDO são aproximadas numericamente por operadores diferenças gerando sistemas de equações algébricas, cuja solução fornece aproximações da solução do PVC, conforme (Faires; Burden, 2011). Uma segunda maneira de se resolver um PVC consiste em determinar a condição inicial desconhecida através de uma função entre esta condição inicial desconhecida e a condição de contorno conhecida, que será desenvolvido a seguir.

9.5.1. Métodos de determinação de uma condição inicial em função de uma condição de contorno

Para o PVC $y'' = f(x, y, y')$, $x \in [a, b]$ com condições de contorno $y(a) = v_1$ e $y(b) = D$ conhecidos, a idéia é tentar **adivinhar** um valor inicial C para o $y'(a)$ desconhecido do PVI a seguir

$$y'' = f(x, y, y'), \quad x \in [a, b] \quad \text{com} \quad y(a) = y_1(a) = v_1 \quad \text{e} \quad y'(a) = y_2(a) = C = ?$$

de modo que após resolvido este PVI, pelos métodos abordados na sessão anterior a partir das condições iniciais (CI) atribuídas, se atinja a condição de contorno (CC) $y(b) = D$ conhecida. É o mesmo que determinar C da equação $\text{erro}(C) = Z(C) - D = 0$, onde $y_1(b) = y(b) = Z(C)$ é o alvo atingido pela solução numérica da EDO, obtida a partir de $y_2(a) = y'(a) = C$ estabelecido, e $y_1(b) = y(b) = D$ é o alvo correto. A função $Z(C)$ representa o cálculo da condição de contorno $y_1(x=b)$ (alvo) em função de uma condição inicial atribuída (mira) $y_2(x=a) = C$, através da solução numérica de um sistema de PVI's para cada C atribuído.

Através do exemplo 9.6 a seguir, vamos obter a solução de um PVC de 2ª ordem aplicando o chamado *Shooting Method*, onde calculamos a condição inicial (mira) $y_2(x=0) = C$, através de **correções** baseadas no valor da condição de contorno conhecida (alvo) $y_1(x=1) = D$. Por isso este método de tentativas é chamado de *shooting* (tiro). Alternativamente, vamos determinar diretamente a raiz

da equação $erro(C) = Z(C) - D = 0$ pelo Método de Newton com derivadas numéricas (Secante).

Exemplo 9.6: Determine a solução $y(x)$ da EDO de 2ª ordem $y'' - 4y' + 4y = 8x^2 - 16x + 4$, com as duas condições de contorno, $y(x=0) = 0$ e $y(x=1) = 9.389056098930666$, com erro máximo da ordem 10^{-6} . Use o *Shooting Method* e o Método de Newton numérico para determinar a condição inicial desconhecida $y'(x=0) = C$.

Solução:

Para esta EDO de 2ª ordem, definimos as variáveis auxiliares, conforme segue:

$$y_1(x) = y(x) \Rightarrow y_1'(x) = y'(x) = y_2(x) \text{ (própria solução } y(x) \text{)}$$

$$y_1'(x) = z_1(x, y_1, y_2) = y_2(x)$$

$$y_1(x=0) = y(x=0) = 0 \text{ (condição inicial conhecida)}$$

$$y_2(x) = y'(x) \Rightarrow y_2'(x) = y''(x) = +4y' - 4y + 8x^2 - 16x + 4 \text{ (} y'' \text{ é própria EDO)}$$

$$y_2'(x) = z_2(x, y_1, y_2) = +4y_2 - 4y_1 + 8x^2 - 16x + 4$$

$$y_2(x=0) = y'(x=0) = C \text{ (} C \text{ é uma condição inicial de valor desconhecido)}$$

Resultando em 2 EDOs de 1ª ordem que precisam de 2 condições iniciais:

$$\begin{cases} y_1'(x) = z_1(x, y_1, y_2) = y_2, & \text{com } y_1(x=0) = y(x=0) = 0 \\ y_2'(x) = z_2(x, y_1, y_2) = +4y_2 - 4y_1 + 8x^2 - 16x + 4, & \text{com } y_2(x=0) = y'(x=0) = C \end{cases}$$

Como a condição inicial para $y_2(x=0)$ é desconhecida, é atribuída uma incógnita C , que precisa ser determinada, mas conhecemos uma condição adicional, que é o valor da condição de contorno no outro extremo do domínio de cálculo, em $x=1$, $y_1(x=1) = y(x=1) = 9.389056098930666$, que será denominada de D .

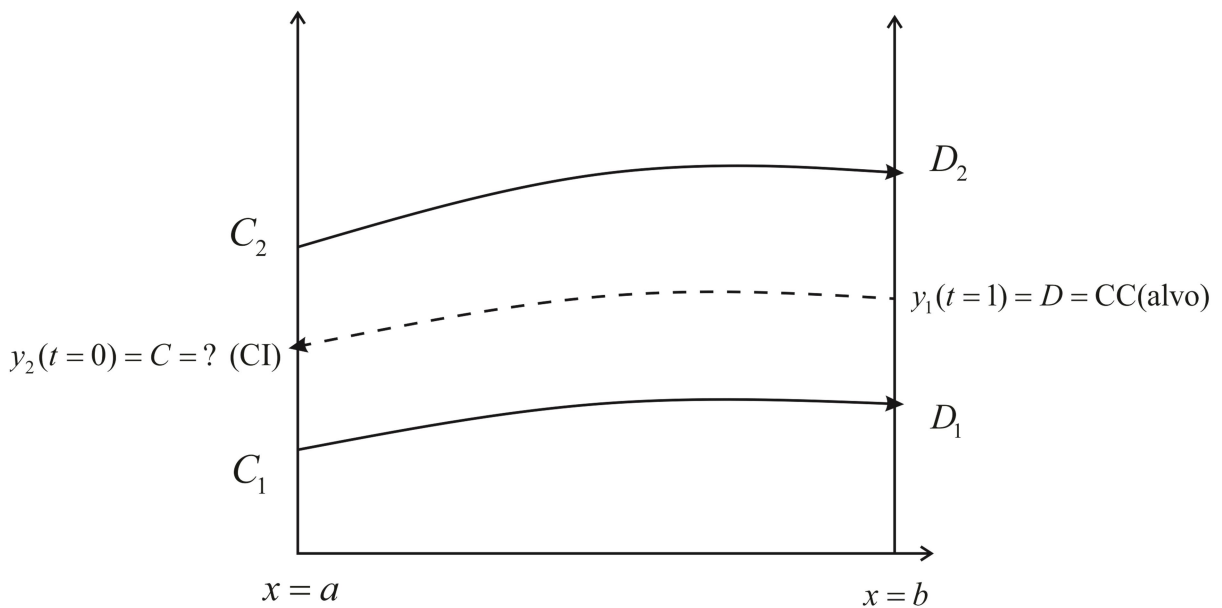
Para determinar o C podemos:

a). Determinando C pelo *Shooting method*

Vamos atribuir 2 valores estimados para C (miras), resolver 2 vezes as EDO's, e obter 2 valores finais para $y_1(x=1)$ em $x=1$ (alvos atingidos), que a princípio estarão errados, mas o valor correto em $x=1$ é conhecido, $y_1(x=1) = D = 9.389056098930666$ (alvo correto).

Então, este método considera estimativas da condição inicial ausente, como se fosse a mira de tiro. Atribuindo dois valores iniciais de mira, C_1 e C_2 , resolvendo as equações entre o ponto inicial a e o ponto final b , determinando 2 resultados para os alvos $y_1(x=1)$ atingidos no outro extremo b (contorno), como D_1 e D_2 , cujo valor correto é o alvo com o valor D da condição de contorno conhecida em b , conforme Gráfico 9.6 a seguir.

Gráfico 9.6 - *Shooting method* para determinar uma condição inicial (CI) C a partir de uma condição de contorno (CC) D conhecida.



Fonte: Elaboração própria

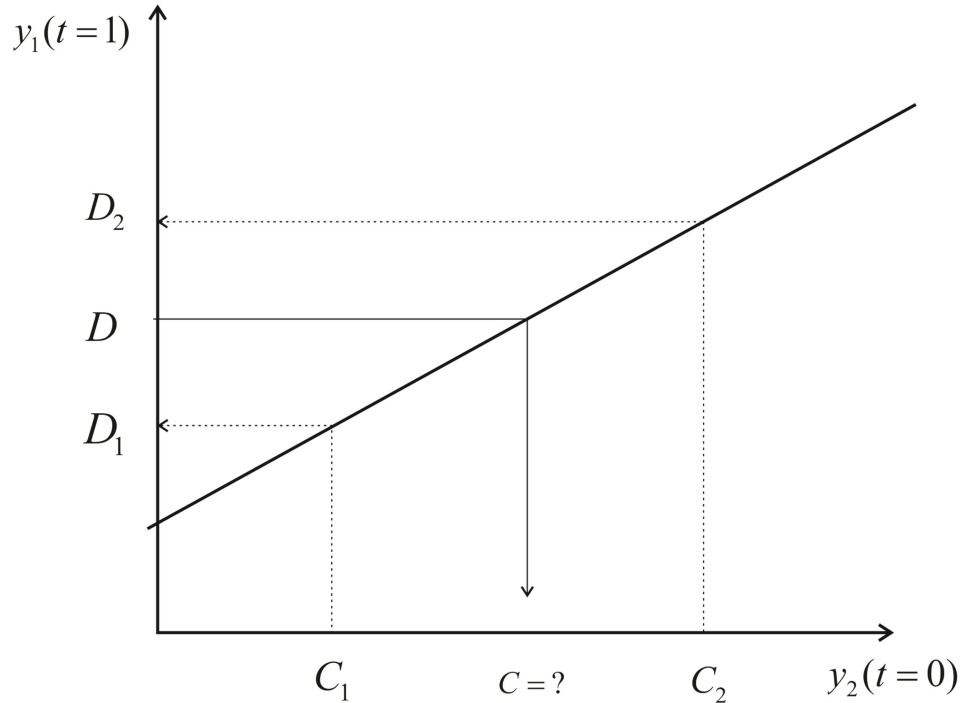
Então, podemos relacionar $y_2(x=0) = C$ com $y_1(x=1) = D$, supondo uma relação de proporcionalidade linear entre C e D , que é uma relação válida para valores próximos da solução, conforme mostrado na Gráfico 9.7:

Para $y_2(x=0) = y'(x=0) = C_1$, obtemos $y_1(x=1) = y(x=1) = D_1$;

Para $y_2(x=0) = y'(x=0) = C_2$, obtemos $y_1(x=1) = y(x=1) = D_2$;

Para $y_2(x=0) = y'(x=0) = C = ?$, obter $y_1(x=1) = y(x=1) = D = 9.389056098930666$.

Gráfico 9.7 - Relação linear entre a condição de contorno conhecida $y_1(x=1) = D$ (alvo) e a condição inicial $y_2(x=0) = C$ desconhecida (mira).



Fonte: Elaboração própria

Logo, estabelecendo uma relação de proporcionalidade entre as condições iniciais C_1 e C_2 (miras propostas) com as condições de contorno D_1 e D_2 (alvos atingidos), calculadas na outra extremidade, podemos estabelecer uma relação de linear entre C e D , conforme segue,

$$\frac{C - C_1}{C_2 - C_1} = \frac{D - D_1}{D_2 - D_1}$$

$$C = C_1 + (C_2 - C_1) \left(\frac{D - D_1}{D_2 - D_1} \right) \quad (24)$$

Obtido o valor de C , via eq. (24), teremos um resultado melhor que os valores iniciais C_1 e C_2 , então podemos escolher 2 destes valores iniciais e refazer o cálculo de C iterativamente.

Vamos tomar C e C_2 como os 2 melhores valores, descartando C_1 , conforme a seguinte seleção:

$C_1 = C_2$ (atualiza C_1 , substitui por C_2)

$C_2 = C$ (atualiza C_2 , substitui por C)

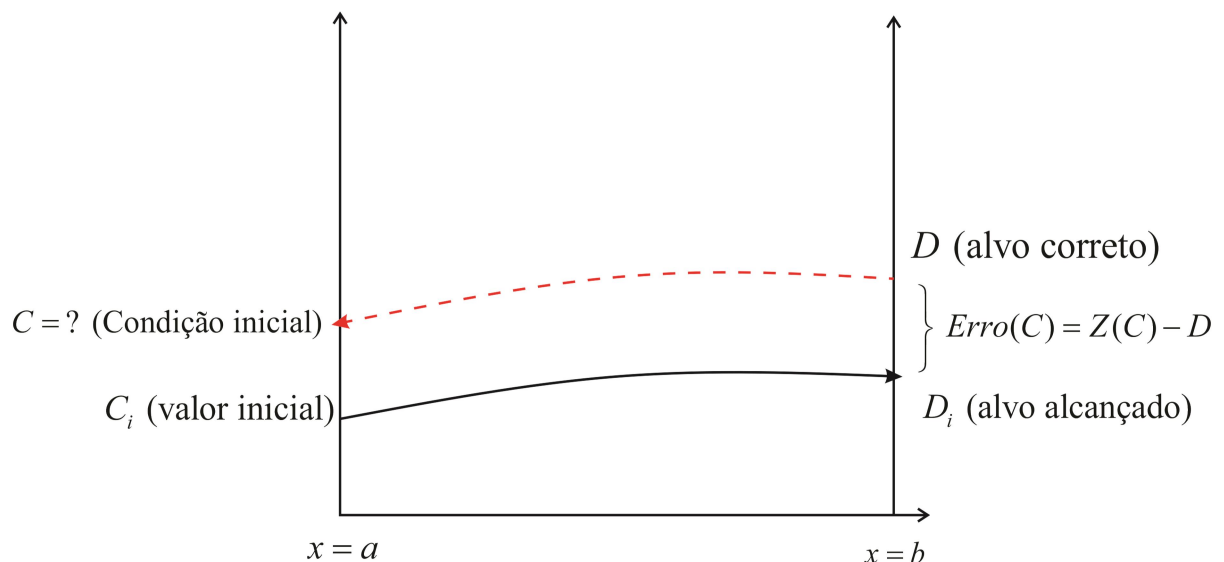
Com novos C_1 e C_2 , recalculamos D_1 e D_2 (como D_1 é o valor de D_2 no passo anterior, então $D_1 = D_2$ não precisa ser recalculado), refazemos o cálculo de C , via eq. (24), até que algum critério de parada seja satisfeito. Podemos avaliar se $|C_2 - C_1| < 10^{-6}$, $|D_2 - D| < 10^{-6}$ ou $|D_2 - D_1| < 10^{-6}$, como possíveis critérios de parada.

b). Determinando C pelo método de Newton:

Vamos determinar C diretamente através da equação do erro $erro(C) = Z(C) - D = 0$, em função da condição C desconhecida, aplicando o método de Newton com derivadas numéricas (Secante).

Observe no Gráfico 9.8, o Erro entre o alvo atingido D_i , via resolução da EDO, e o alvo correto D conhecido. A mira correta C será atingida quando o $erro(C) = Z(C) - D = 0$.

Gráfico 9.8 – Erro entre o alvo atingido D_i o alvo correto D



Fonte: Elaboração própria

Observe que para cada valor inicial de C_i (mira), fica definida a condição inicial $y_2(x=0) = C_i$ do sistema de PVI's, e pode-se resolver as EDO's pelos métodos disponíveis, obter o valor final para $y_1(x=1) = D_i$ em $x=1$ (alvo atingido), que seria

equivalente a uma função $Z(C_i) = D_i$ (condição de contorno calculada). A princípio o valor D_i está errado, mas o valor correto em $x=1$ está disponível, dado por $y_1(x=1) = D = 9.389056098930666$ (alvo correto). Logo, o valor correto de C deve satisfazer a equação $\text{erro}(C) = Z(C) - D = 0$, que pode ser resolvida pelo método de Newton com derivadas numéricas. LINK Este mesmo método de Newton pode ser estendido para problemas de valor no contorno com 2 ou mais condições iniciais desconhecidas, bastando gerar e resolver 2 ou mais funções de erro entre os alvos atingidos, pela própria solução numérica do sistema de PVI's, e o alvo correto, conforme [exercício 9.7.FIMLINK](#)

Após determinar o valor da condição inicial desconhecida, $C=1$, conforme algoritmo apresentado, devemos estimar o erro máximo da solução em função do número n de subintervalos estabelecido, mas como vamos medir o erro máximo da solução $y(x)$, uma vez nos 2 contornos (extremos) os erros são nulos, então onde o erro deverá ser máximo?

Possivelmente o maior erro estará no meio no intervalo $x \in [0, 1]$, então poderemos estimar o erro máximo comparando a solução aproximada com n subdivisões com outra solução aproximada com $2n$ subdivisões no ponto médio, por exemplo, ou calcula-lo em vários pontos e tomar o seu valor máximo. Vamos também conferir esta estimativa de erro com o máximo erro exato calculado em vários pontos do domínio (disponível neste exemplo 9.6), conforme algoritmo a seguir.

Neste exemplo 9.6, vamos determinar o n mínimo por sucessivas tentativas (ou pode-se aplicar o tradicional algoritmo de busca do n mínimo, enquanto algum erro seja maior que a tolerância).

Apresentamos no Caderno de Algoritmos os arquivos Cap9exep9.6ShootingNewton2.m e Cap9exem9.6NR2.m, com algoritmos que resolvem o exemplo 9.6, determinando C via *shooting method* e via Método de Newton, respectivamente.

Resultados:

Com $n=18$ obtemos:

$C = 1.00000377206141$ pelo Shooting, obtido em 2 iterações

$C = 1.00000377206141$ pelo método de Newton, obtido também em 2 iterações

Erro exato máximo de $y_1 = 2.81176553951212e-06$ ocorre em $x = 0.666666666666667$

Erro estimado máximo calculado no meio do domínio $x = 0.5$ é $y_1 = 2.59399765134560e-06$. Neste caso na mesma ordem.

Conclusão

Neste capítulo apresentamos uma abordagem superficial da solução numérica das equações diferenciais ordinárias EDO's visando suprir os tópicos constantes nos programas de disciplinas dos cursos de graduação e resolver modelos matemáticos básicos que utilizam esta ferramenta. Existem inúmeros tipos de EDO's, e poucos são os métodos analíticos conhecidos que fornecem sua solução. Por isso recorre-se com frequência aos métodos numéricos para a solução de uma EDO. Os métodos de passo simples da família Runge-Kutta, aqui detalhados, tem sido amplamente usados para resolver EDO's por fornecerem resultados precisos e com baixo esforço computacional.

O próximo passo nesta área é a solução numérica das equações diferenciais parciais EDP's através dos métodos clássicos das Diferenças Finitas, Elementos Finitos ou Volumes Finitos. Para o método dos Volumes Finitos, indicamos a referência (Maliska;C.R., 2004).

COMPLEMENTANDO...

Solução exata de $y' = x - y + 2$ (para exemplos 9.1, 9.2, 9.3):

Reescrevendo:

$y' + y = x + 2 \Rightarrow$ multiplicando um fator integrante $u(x)$ desconhecido em ambos os lados temos:

$$\left(\frac{dy}{dx} + y\right)u(x) = (x + 2)u(x) \Rightarrow u(x)\frac{dy}{dx} + u(x)y(x) = (x + 2)u(x)$$

Se no lado esquerdo tivéssemos apenas a derivada do produto $u(x)y(x)$, para depois facilitar a integração.

Ou seja

$$\text{Se } \frac{d(u(x) * y(x))}{dx} = u(x) * \frac{dy(x)}{dx} + \frac{du(x)}{dx} * y(x)$$

$$\text{For igual a } u(x) * \frac{dy(x)}{dx} + u(x) * y(x)$$

Comparando termo a termo, o fator integrante $u(x)$, caso exista, deve satisfazer:

$$\frac{du(x)}{dx} = u(x)$$

Resolvendo essa equação acima, de variáveis separáveis, temos:

$$\frac{du(x)}{u(x)} = dx \Rightarrow \int \frac{du(x)}{u(x)} = \int dx \Rightarrow \ln(u(x)) = x + C, \text{ escolhendo um valor}$$

adequado $C = 0$ teremos:

$$u(x) = e^x,$$

Substituindo o nosso fator integrante $u(x)$, temos que

$$\left(\frac{dy}{dx} + y\right)u(x) = \left(\frac{dy}{dx} + y\right) * e^x = e^x * \frac{dy}{dx} + e^x * y = \frac{d(e^x * y(x))}{dx}, \quad \text{assim podemos}$$

trocar o termo $\left(\frac{dy}{dx} + y\right) * e^x$ por $\frac{d(e^x * y(x))}{dx}$ e a nossa EDO fica:

$$\left(\frac{dy}{dx} + y\right) * e^x = \frac{d(e^x * y(x))}{dx} = (x + 2) * e^x \Rightarrow$$

$$\frac{d(e^x * y(x))}{dx} = (x + 2) * e^x$$

Integrando,

$$\int d(e^x * y(x)) = \int (x + 2) * e^x dx \Rightarrow e^x * y(x) = \int x * e^x dx + \int 2 * e^x dx$$

Integrando $\int x * (e^x * dx) = x * e^x - \int e^x * 1 * dx$ por partes $\int u dv = u * v - \int v du$, obtemos

$$\int x * e^x * dx = x * e^x - e^x \Rightarrow \int x * e^x dx = (x - 1) * e^x + c$$

Integrando $\int 2 * e^x * dx = 2 * e^x$ diretamente, temos

$$e^x * y(x) = (x - 1) * e^x + 2 * e^x + c$$

$$e^x * y(x) = (x + 1) * e^x + c$$

$$y(x) = (x + 1) + c * e^{-x}$$

Aplicando a condição inicial $y(0) = 2$, temos $y(x = 0) = (0 + 1) + c * e^{-0} = 2 \Rightarrow c = 1$

Logo a solução é dada por:

$$y(x) = (x + 1) + e^{-x}$$

FIM COMPLEMENTANDO