

MS211 - Cálculo Numérico
Turma K - 2o semestre de 2018 - UNICAMP

Rafael Matheus Garcia - RA 121295

Projeto 02 - *Aproximação Numérica para a Solução de EDOs*

Objetivo

Comparar os métodos de aproximação numérica para equações diferenciais ordinárias. No caso em específico, a análise é referente à equação logística de *Pierre-François Verhulst* tal qual da forma:

$$y'(t) = r y(t) (1 - (y(t) / K)) \quad (I^*)$$

Esta descreve a taxa de crescimento populacional dado que $y(t)$ seja o número de indivíduos no instante t ; r seja a taxa de reprodução da população e K a constante de capacidade do meio. K é o maior valor populacional quando $t \rightarrow \infty$.

A solução analítica para o problema de valor inicial é dada por:

$$y(t) = Ky_0 e^{rt} / (K + y_0(e^{rt} - 1)) \quad ,$$

Na qual y_0 é o número de indivíduos inicialmente ($t = 0$).

Execução do Código

O projeto foi desenvolvido na linguagem Kotlin, que é compatível com a JVM (ambiente Java). Existem algumas opções para executar o código:

1. Entrar em <http://try.kotlinlang.org/> e colar o código fonte localizado em *main.kt*
2. Executar o jar anexado, que possui a versão compilada do código
3. Compilar o código *proj02-ms211k.kt*. Para tal, usar ferramentas como o IntelliJ (<https://www.jetbrains.com/idea/>) ou baixando o CLI do Kotlin (<https://kotlinlang.org/docs/tutorials/command-line.html>)

A função **fun** `main(args: Array<String>)` possui as diversas chamadas com diferentes entradas para os três métodos: **fun** `euler`, **fun** `analytical` e **fun** `rungeKutta4th`.

Customizações de valores dos parâmetros podem ser feitas ali. Todas as funções aceitam todos os parâmetros, e já possuem valores padrão para serem executadas conforme valores do **item a** a seguir.

A precisão escolhida (tolerância) foi **10⁻⁵**

Observação: dependendo do tamanho do passo **h** escolhido, a saída do programa pode ser muito grande. É recomendável que a direcione para um arquivo a seu critério.

Item a - Aproximação por método de Euler

Considere $r = 0.5$; $K = 10$; $h = 0.05$; $y_0 = 1$ e $t \in [0, 4]$.

A aproximação de Euler para equações diferenciais ordinárias (EDOs) é tal que:

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

onde $y'(t) = f(t_n, y_n)$, ou seja, é o lado direito da equação **(I*)**.

Segue então para o objeto de estudo que:

$$y_{n+1} = y_n + h(r y(t) (1 - (y(t) / K))) ,$$

No método de Euler h é chamado de passo. Ele será sempre acrescido ao t_n atual , de tal forma que $t_{n+1} = t_n + h$.

Tanto t_{n+1} e y_{n+1} serão calculados iterativamente $(t_{nMax} - t_0) / h$ vezes, ou seja,

$$(4 - 0) / 0.05 = 80 \text{ iterações calculadas.}$$

É importante ressaltar que o método de Euler, na realidade, é um método da série de Taylor truncado de ordem **1** (ou 2, denominado de método de *Euler aprimorado*). Também pode ser escrito como o método de Runge-Kutta de ordem 1, conforme será dito posteriormente.

Considerando os valores dados, segue o gráfico plotado da EDO e de $y(t)$ respectivamente.

A ferramenta utilizada para a geração dos gráficos foram o **Wolfram Alpha Pro for Students**.

Gráfico 1.0: $y(t)$ para o intervalo $t \in [0, 4]$

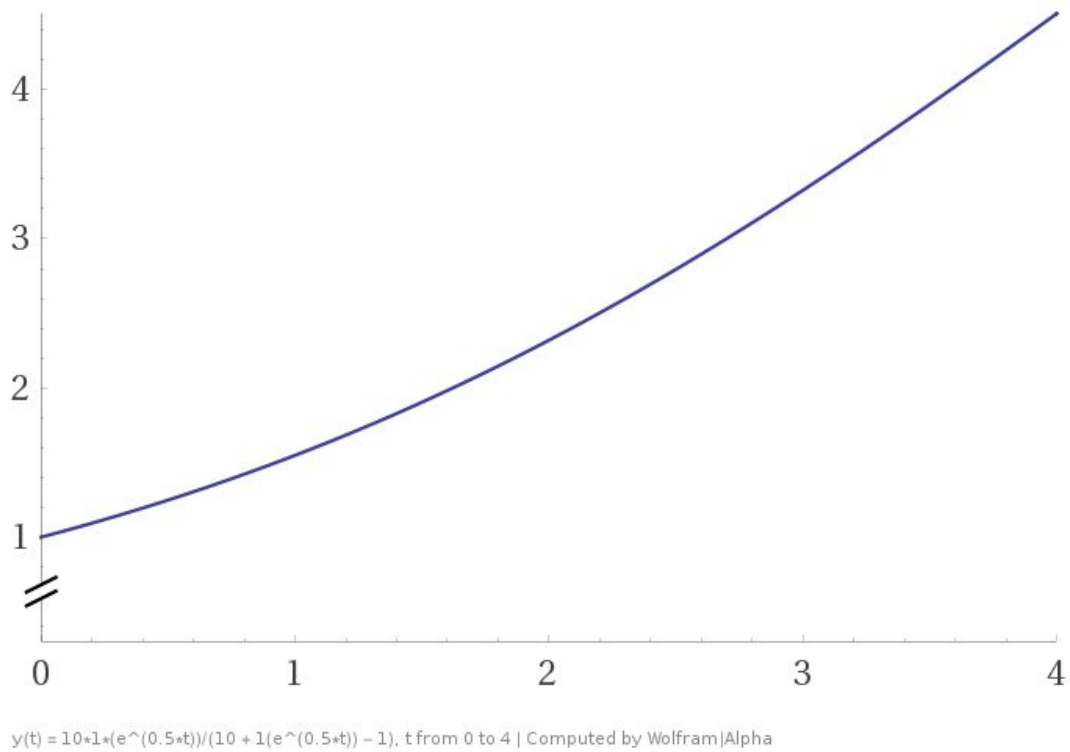
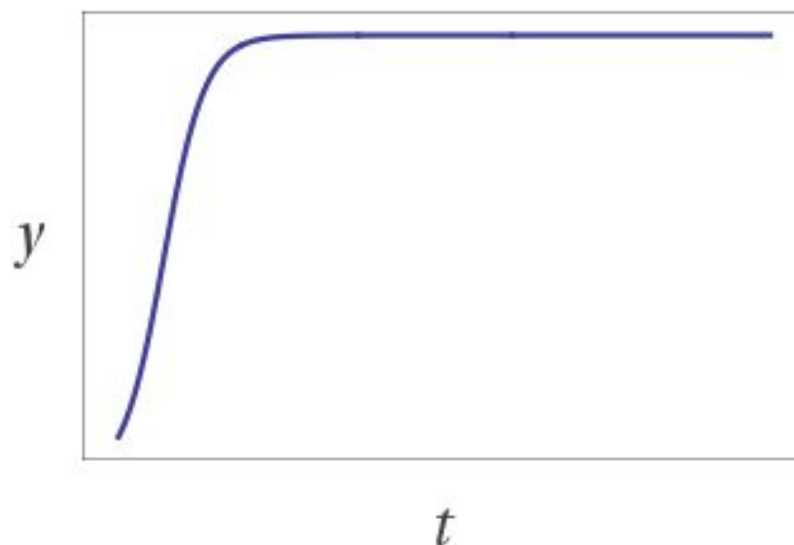
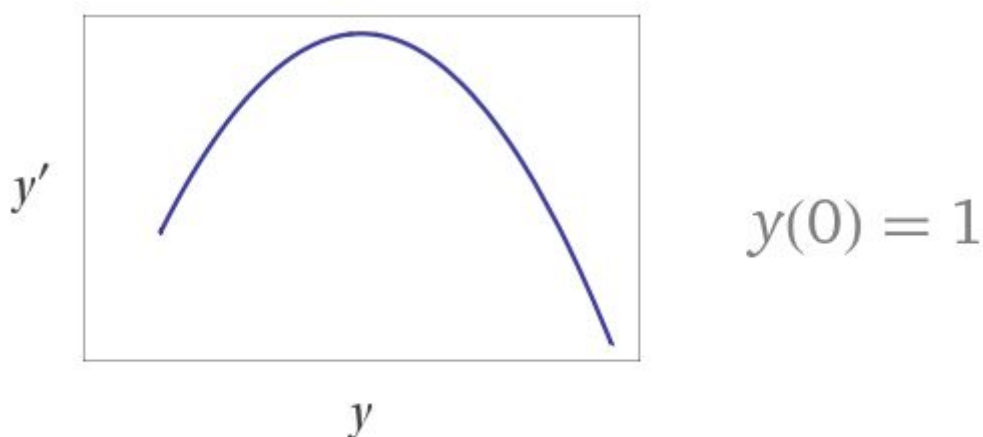


Gráfico 1.1: Uma visão mais ampla da curva $y(t)$.



Nota-se que a função tende rapidamente ao $y(t) = K$, o limite do ambiente.

Gráfico 1.2: Gráfico com os valores de $y'(t)$ e $y(t)$ para $y_0 = 1$.



A tabela 1.1 abaixo mostra a comparação dos resultados de $y(t)$ analítico com as aproximações obtidas pelo método de Euler.

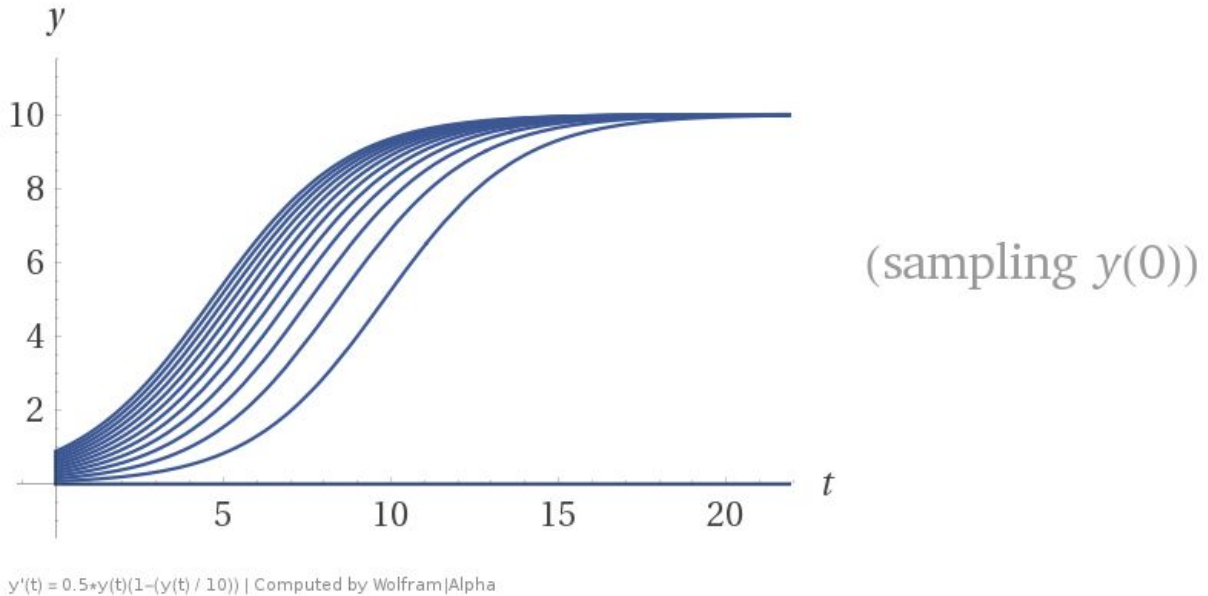
Tabela 1.1: Analítico e aproximação pelo método de Euler:

t	y(t)	$\sim y(t)$ por Euler	Erro Local
0	1	1.02250	0.02250
0.05	1.02273	1.04545	0.02272
0.10	1.04591	1.06885	0.02294
0.15	1.06955	1.09272	0.02316
$\dots t_k = t_{k-1} + h \dots$	$\dots y(t_k) \dots$	$\dots y_k \dots$	$\dots y(t_k) - y_k \dots$
4	4.50853	4.53908	0.03055

Observação: Os dados da Tabela 1.1 foram gerados a partir da chamada da função `fun_euler()` que está com sua implementação presente no arquivo `proj02-ms211k.kt`.

Item b - Variação nos parâmetros fornecidos

Gráfico 1.3: Gráfico com variações de $y(0)$ mostrando que a convergência é sempre exponencial até alcançar $y(t) = K$.



No método de Euler, por ser iterativo, quanto menor a granularidade do passo h , melhor será o resultado (mais próximo do resultado analítico).

Essa afirmação é verdadeira de maneira intuitiva, já que a cada iteração a tendência do y_n corrente é de aproximar do valor original.

Assim, segue demonstração das variações de h e as consequências na precisão da aproximação. Note que por se tratar de um número muito grande de iterações, não convém colocar todos os resultados parciais do método. Esses podem ser verificados executando o código. A tabela tem o número de linhas baseado em $(t_{nMax} - t_0) / h_{max}$, onde $h_{max} = 1.0$

Tabela 1.2: Valores de y_k no intervalo $[t_0...t_{nMax}]$ para outros valores de passo h

t	y(t)	$\sim y(t), h = 1.0$	$\sim y(t), h = 0.5$	$\sim y(t), h = 0.05$	$\sim y(t), h = 0.0005$	$\sim y(t), h = 0.00005$
0.00	1	1.45000	1.22500	1.02250	1.00023	1.00000
1.00	1.54828	2.06987	1.81139	1.57484	1.54855	1.54829
2.00	2.31969	2.89059	2.60871	2.34898	2.31999	2.31970
3.00	3.32428	3.91811	3.62462	3.35468	3.32458	3.32428
4.00	4.50853	5.10959	4.81142	4.53908	4.50884	4.50854

É fácil observar que quanto menor o valor de h , a aproximação fica mais próxima da solução analítica. Na última coluna, temos um erro absoluto $|EA| \leq 0.00001 = 5.0E-4$

Item c - Runge-Kutta de 4ª ordem

A aproximação por Runge-Kutta de 4ª ordem é notada por:

$$y_{n+1} = y_n + (1/6) * (k_1 + 2 * k_2 + 2 * k_3 + k_4),$$

Tal que:

$$k_1 = h * (f(t_n, y_n))$$

$$k_2 = h * (f(t_n + h/2, y_n + k_1/2))$$

$$k_3 = h * (f(t_n + h/2, y_n + k_2/2))$$

$$k_4 = h * (f(t_n + h, y_n + k_3))$$

$$y'(t) = f(t_n, y_n), \text{ ou seja, é o lado direito da equação (I*)}.$$

A idéia da iteração é a mesma que de Euler, onde cada passagem mostrada acima, será calculada iterativamente $(t_{nMax} - t_0) / h$ vezes, ou seja, $(4 - 0) / h$ iterações calculadas.

É importante ressaltar que a EDO em questão **não** possui t_n , como visto em (I*).

Dessa forma, o único parâmetro variando nos cálculos dos k_i é y_n

Item d - Comparação de métodos

Tabela 1.3: Analítico e aproximação pelo método de Runge-Kutta ordem 4

t	y(t)	~y(t) por Runge-Kutta	Erro Local	Diferença com Euler
0	1	1.02273	0.02273	0.00023
0.05	1.02273	1.04591	0.02318	0.00046
0.10	1.04591	1.06955	0.02365	0.00070
0.15	1.06955	1.09367	0.02411	0.00095
... $t_k = t_{k-1} + h$ $y(t_k)$ y_k $ y(t_k) - y_k $...	$ EA_e - EA_{RK4th} $
4	4.50853	4.57050	0.06197	0.03142

Podemos observar que para $h = 0.05$ o método obteve resultados piores se comparado à aproximação de Euler com o mesmo h .

Tabela 1.4: Comparação dos Δ (deltas) de resultados entre a aproximação de Euler e de Runge-Kutta ordem 4 com os mesmos h apresentados no item b

t	$y(t)$	$\Delta p/h = 1.0$	$\Delta p/h = 0.5$	$\Delta p/h = 0.05$	$\Delta p/h = 0.0005$	$\Delta p/h = 0.00005$
0.00	1	0.09815	0.02356	1.02273	0.00000	0.00000
1.00	1.54828	0.24952	0.09289	1.58128	0.00015	0.00000
2.00	2.31969	0.43320	0.18570	2.34898	0.00025	0.00001
3.00	3.32428	0.58982	0.27555	3.35468	0.00031	0.00001
4.00	4.50853	0.64099	0.32045	4.53908	0.00001	0.00001

Analisando os resultados obtidos nas Tabelas 1.3 e 1.4, pode-se concluir que o método de **Euler** foi mais preciso nas entradas com um passo h grande (pequeno espaço amostral) e que elas são semelhantes em precisão quando h fica na ordem de grandeza **5.0E-4**.

Custo Computacional

Considerando que as operações aritméticas todas sejam da ordem $O(1)$, constantes, então é possível concluir que a ordem de grandeza de ambos os métodos de aproximação estão atrelados ao número de iterações que executarão. Isso por sua vez é inversamente proporcional ao tamanho do passo h .

Assim sendo, segue que:

Euler: $O(t_n - t_0/h)$

Runge-Kutta: $O(t_n - t_0/h/h)$

Referências

[1] <https://www.ime.unicamp.br/~valle/Teaching/MS211/Aula21.pdf>

[2] <https://www.ime.unicamp.br/~valle/Teaching/2015/MS211/Aula12.pdf>

[3] https://www.ufrgs.br/reamat/CalculoNumerico/livro-sci/pdvi-metodo_de_euler.html

[4] <https://pt.khanacademy.org/science/biology/ecology/population-growth-and-regulation/a/exponential-logistic-growth>