RELATÓRIO SOBRE AS IMPLEMENTAÇÕES

Relatório expondo as implementações dos métodos numéricos solicitadas pelo Professor Doutor Ricardo Martins de Abreu Silva

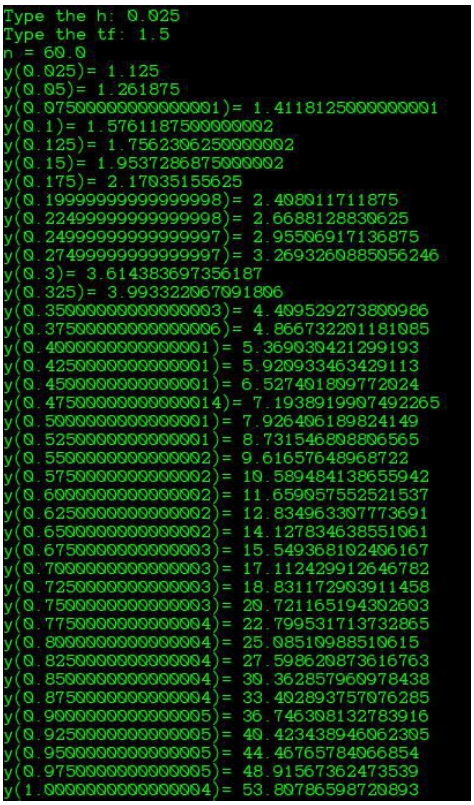
Vitor Matheus de Azevedo Martins

09.11.2017

IF816 - Métodos Numéricos Computacionais

INTRODUÇÃO

Na implementação dos métodos foram usadas as bibliotecas **matplotlib.pyplot** para fazer a plotagem dos dados no gráfico, assim como para exibi-lo, a biblioteca **math** foi utilizada para abranger funções como logarítmica, trigonométricas, entre outras e a biblioteca **decimal** para trabalhar com os números decimais, visto que possui uma precisão melhor que o float (inclusive erros de precisão no float que me levaram a usar esta biblioteca).



Na imagem ao lado temos cada termo das iterações no método de euler para uma devida função, onde o t_n deveria ser somado em h = 0.025 a cada iteração. O uso de float levou a esta imprecisão que acabaria trazendo resultados insatisfatórios para um número de iterações muito grande. O problema foi sanado ao utilizar Decimal() no lugar de float.

Entretanto operações entre float e Decimal não são suportadas, então se torna necessário que o usuário digite qualquer decimal ou uma função de **math** como argumento de Decimal(). Além disso, para as funções de **math** é necessário utilizar o prefixo math.. É preciso que o usuário digite explicitamente os operadores. Para o número de Euler deve-se utilizar math.exp(). Qualquer divergência da entrada com o que foi citado neste documento poderá causar um erro.

Documentação:
https://docs.python.org/2/library/math.html
https://docs.python.org/2/library/decimal.html
https://matplotlib.org/users/

As implementações também podem ser encontradas no meu repositório do Github.

EXEMPLOS

Expressão	Input
0.2y	Decimal(0.2)*y
sen(4t) + y	Decimal(math.sin(4*t)) + y
e^{3t} + 0.5y	Decimal(math.exp(3*t)) + Decimal(0.5)*y
1-t+4y	1 - t + 4*y
ln(t+1)	Decimal(math.log(t+1))

MÉTODO DE EULER

Para o método de Euler (euler.py), é pedido que o usuário digite inicialmente a função y'. Além disso é pedido um valor inicial de y, ou seja, t_0 e y_0 de forma que $y(t_0) = y_0$.

É dada a opção do usuário escolher entre dar como entrada o **h** (height) e o **número de passos** (n_steps), dar o **h** e o $\mathbf{t_f}$ (tf) desejado, ou ainda o **número de passos** e o $\mathbf{t_f}$ desejado. A terceira variável que o usuário optar por não digitar será computada posteriormente.

Considere a função $y' = 1 - t + 4y; y(0) = 1$ como exemplo para o nosso caso teste. Usaremos também um $h = 0{,}001$ e iremos até $y(2)$, ou seja, $t_f = 2$.

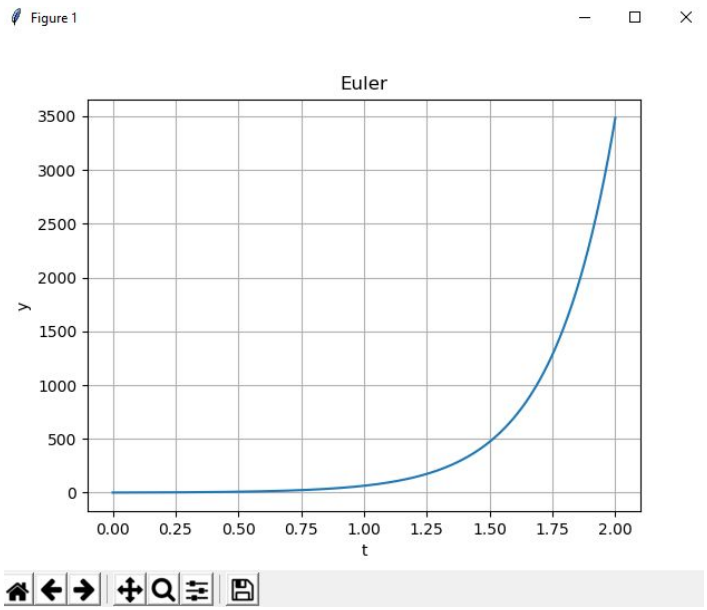
INPUT:

```
Type the function (y'): 1 - t +4*y
Type the y0: 1
Type the t0: 0
1 - Enter height and number of steps(divisions)
2 - Enter height and the tf
3 - Enter number of steps(divisions) and the tf
2
Type the h: 0.001
Type the tf: 2
```

OUTPUT:

```
y(2) = 3484.160803076250053082217390
```

E a janela com o gráfico da plotagem é aberta:



MÉTODO DE EULER INVERSO

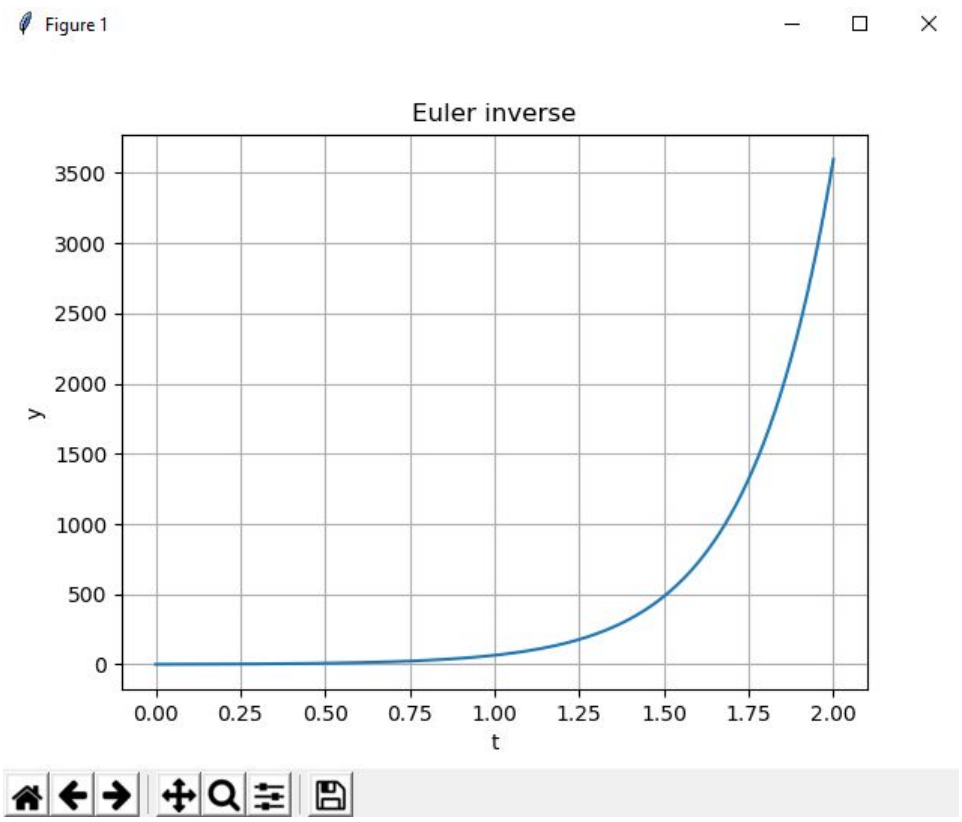
O método de Euler inverso (euler_inverse.py) é feito de forma semelhante ao método de Euler. A entrada é dada da mesma forma. É necessário frisar que o y_{n+1} não foi feito implicitamente, e sim como uma aproximação do método de Euler.

Considere a função $y' = 1 - t + 4y$; $y(0) = 1$ como exemplo para o nosso caso teste. Usaremos também um $h = 0{,}001$ e iremos até $y(2)$, ou seja, $t_f = 2$.

```
Type the function (y'): 1 -t + 4*y
Type the y0: 1
Type the t0: 0
1 - Enter height and number of steps(divisions)
2 - Enter height and the tf
3 - Enter number of steps(divisions) and the tf
2
Type the h: 0.001
Type the tf: 2
```

```
y(2) = 3596.9873757660465526495482 41
```

E a janela com o gráfico da plotagem é aberta:



MÉTODO DE EULER MODIFICADO

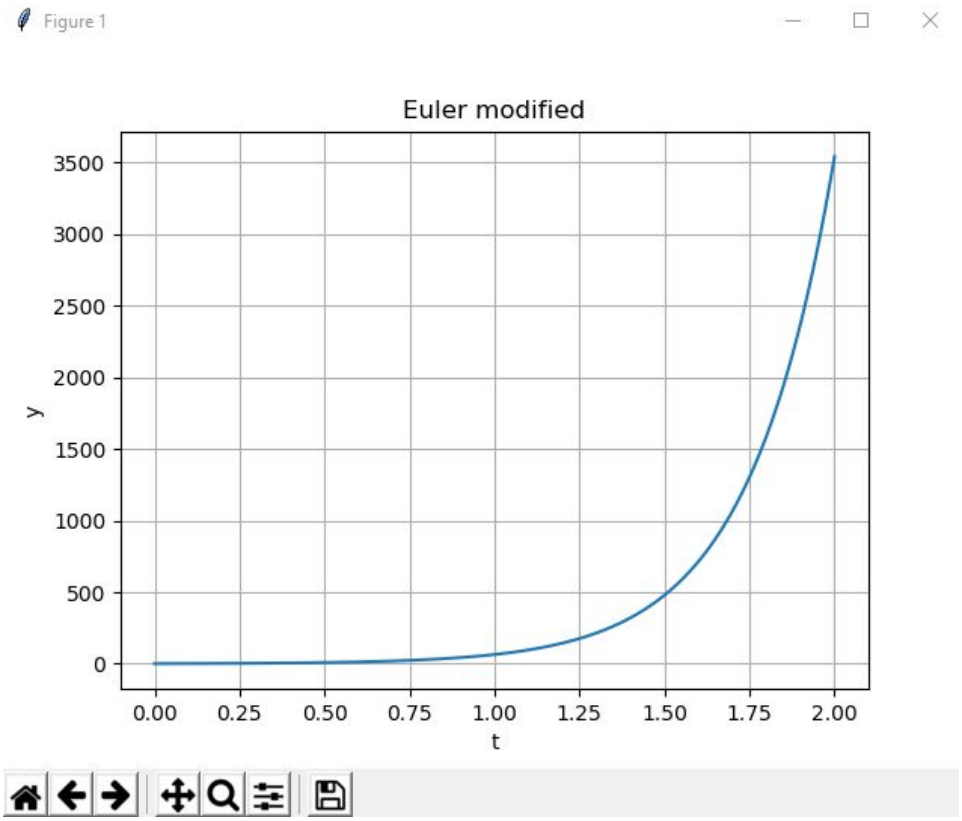
O método de Euler modificado (euler_modified.py) também foi feito de forma semelhante ao método de Euler. A entrada se dá da mesma forma. Cada y_{n+1} é dado com base na média entre o y_n, encontrado anteriormente, e o y_{n+1} do método de Euler.

Considere a função $y' = 1 - t + 4y$; $y(0) = 1$ como exemplo para o nosso caso teste. Usaremos também um $h = 0{,}001$ e iremos até $y(2)$, ou seja, $t_f = 2$.

```
Type the function (y'): 1-t+4*y
Type the y0: 1
Type the t0: 0
Choose one:
1 - Enter height and number of steps(divisions)
2 - Enter height and the tf
3 - Enter number of steps(divisions) and the tf
2
Type the h: 0.001
Type the tf: 2
```

```
y(2) = 3540.124819000735771583460563
```

E a janela com o gráfico da plotagem é aberta:



RUNGE-KUTTA

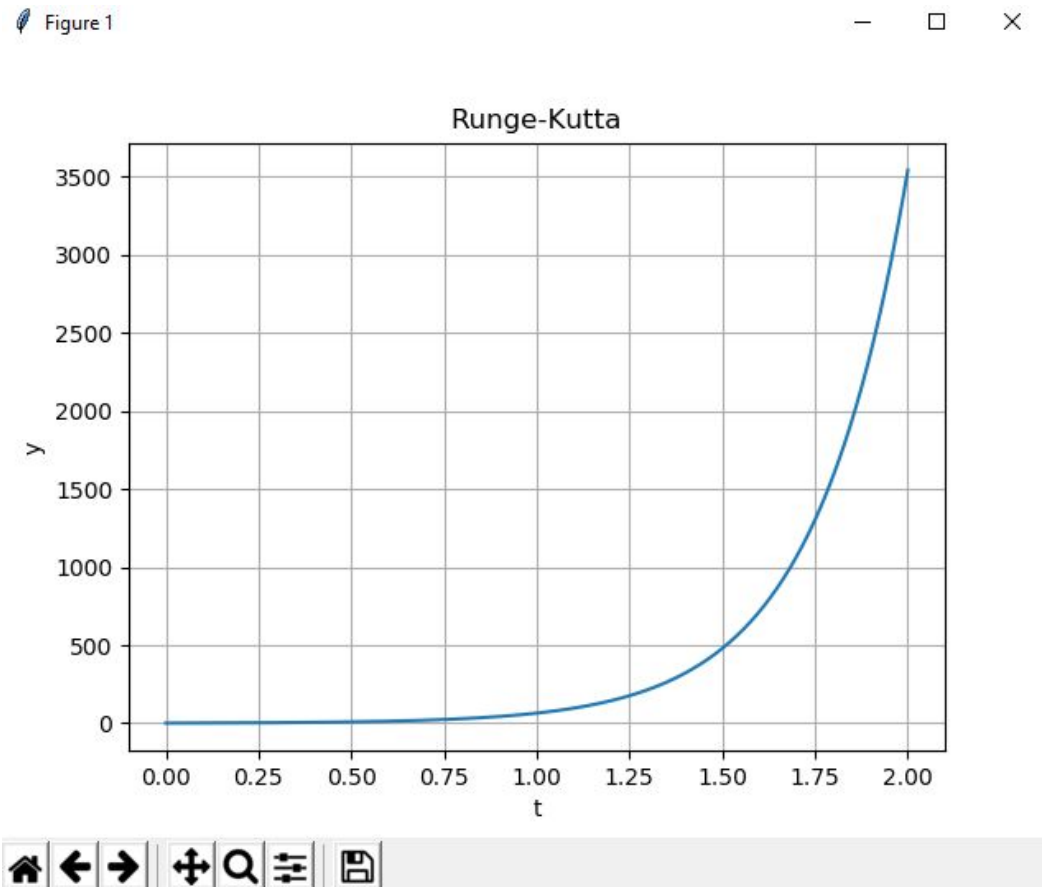
O entrada para o método de Runge-Kutta (runge_kutta.py) também é feita da mesma forma que o método de Euler.

Considere a função $y' = 1 - t + 4y$; $y(0) = 1$ como exemplo para o nosso caso teste. Usaremos também um $h = 0{,}001$ e iremos até $y(2)$, ou seja, $t_f = 2$.

```
Type the function (y'): 1-t+4*y
Type the y0: 1
Type the t0: 0
1 - Enter height and number of steps(divisions)
2 - Enter height and the tf
3 - Enter number of steps(divisions) and the tf
2
Type the h: 0.001
Type the tf: 2
```

```
y(2) = 3540.200109551839279839795312
```

E a janela com o gráfico da plotagem é aberta:



MÉTODO DE ADAMS-BASHFORTH

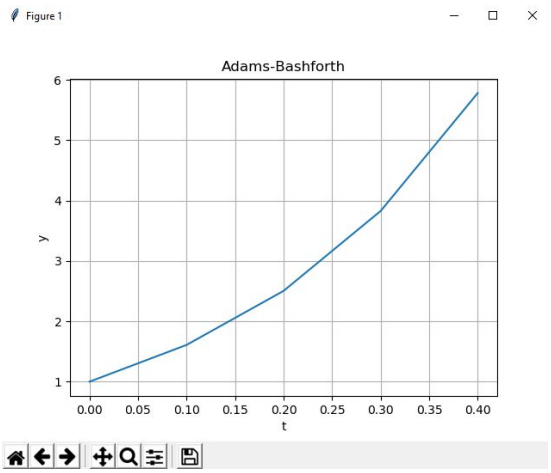
Para o método de Adams-Bashforth, além da função y', do ponto inicial e das variáveis **height**, **n_steps** e/ou **tf**, é pedido que o usuário entre com mais alguns pontos, com a quantidade dependendo da ordem escolhida para o método. Por exemplo, para o método na segunda ordem, é necessário a entrada de mais um ponto, totalizando dois pontos. Para o método na sexta ordem, é necessário a entrada de mais cinco pontos, totalizando seis pontos. É dada ao usuário a opção de escolher qual ordem deseja.

Considere a função $y' = 1 - t + 4y; y(0) = 1$ como exemplo para o nosso caso teste. Usaremos também um $h = 0,1$ e iremos até $y(0,4)$, ou seja, $t_f = 0,4$.

```
Type the function (y'): 1-t+4*y
Type the y0: 1
Type the t0: 0
1 - Enter height and number of steps(divisions)
2 - Enter height and the tf
3 - Enter number of steps(divisions) and the tf
2
Type the h: 0.1
Type the tf: 0.4
Choose the order of Adams-Bashforth
1 - First order
2 - Second order
3 - Third order
4 - Fourth order
5 - Fifth order
6 - Sixth order
4
About the point (y1, f1):
Type the 'y' of the point: 1.6089333
About the point (y2, f2):
Type the 'y' of the point: 2.5050062
About the point (y3, f3):
Type the 'y' of the point: 3.8294145
```

```
y(0.4) = 5.7836305633333333333333333
```

E a janela com o gráfico da plotagem é aberta:



MÉTODO DE ADAMS-MOULTON

Para o método de Adams-Bashforth, além da função y', do ponto inicial e das variáveis **height**, **n_steps** e/ou **tf**, é pedido que o usuário entre com mais alguns pontos, com a quantidade dependendo da ordem escolhida para o método. Por exemplo, para o método na terceira ordem, é necessário a entrada de mais um ponto, totalizando dois pontos, o terceiro ponto é calculado utilizando o método de Euler. Para o método na sexta ordem, é necessário a entrada de mais quatro pontos, totalizando cinco pontos, da mesma forma, o sexto ponto será calculado pelo método de Euler. É dada ao usuário a opção de escolher qual ordem deseja.

```
Type the function (y'): 1-t+4*y
Type the y0: 1
Type the t0: 0
1 - Enter height and number of steps(divisions)
2 - Enter height and the tf
3 - Enter number of steps(divisions) and the tf
2
Type the h: 0.1
Type the tf: 0.4
Choose the order of Adams-Moulton
1 - First order
2 - Second order
3 - Third order
4 - Fourth order
5 - Fifth order
6 - Sixth order
5
About the point (y1, f1):
Type the 'y' of the point: 1.6089333
About the point (y2, f2):
Type the 'y' of the point: 2.5050062
About the point (y3, f3):
Type the 'y' of the point: 3.8294145
```

```
y(0.4) = 5.74288856405555555555555556
```

E a janela com o gráfico da plotagem é aberta: