Relatório Projeto Metódos Numéricos - Parte 1

Autor: Thiago Augusto dos Santos Martins - tasm2@cin.ufpe.br
Monitor Chefe: Victor Crisóstomo Mellia - vcm@cin.ufpe.br
Professor: Ricardo Martins de Abreu Silva - rmas@cin.ufpe.br
Disciplina: Métodos Numéricos Computacionais - IF816
Período: 2018.2

21 de Outubro de 2018

Resumo

Este relatório tem como objetivo analisar os métodos numéricos discutidos em sala de aula e implementados em um programa python. Além de calcular os pontos retornados por cada um desses, explictando as informações das implementações requisitadas e uma análise qualitativa entre os métodos e os valores retornados em si. Para executar o projeto ler o arquivo *README.md*.

1 Métodos Numéricos

Do mesmo modo que temos uma sequência de passos para resolver um sistema de equações linear, como:

$$\begin{cases} 3x + 5y + z = 3 \\ 7x – 2y + 4z = 2 \\ -6x + 3y + 2z = 0 \end{cases}$$

também temos uma sequência de passos para solucionar equações diferenciais ordinárias, EDO.

$$y'(t) = f(t, y(t)) \tag{1}$$

Eq. 1: Exemplo de uma EDO do primeiro grau

Existem passos diferentes para achar a solução de cada tipo de EDO. Que pode depender da sua linearidade, do seu grau, se ela é implícita ou explícita, homogênea ou não. A solução para uma EDO é uma função $y(t)$ cujas derivadas satisfazem a equação, podendo ter mais de uma solução. Se for definido um problema de valor inicial(PVI), teremos uma única solução para a EDO.

Mas existem procedimentos numéricos que permitem estimar o valor de pontos da solução dessas EDOs. Nas próximas secções vamos analisar os metódos de,

1. Euler
2. Euler Inverso
3. Euler Aprimorado
4. Runge-Kutta
5. Adam-Bashforth
6. Adam-Multon
7. Fórmula Inversa

como eles são calculados, exemplos de entrada e saída dos métodos e uma comparação entre eles.

1.1 Implementação

Para calcular os métodos foi utilizado a linguagem de programação Python e a biblioteca de matemática simbólica, sympy, que permite um fácil entendimento e resolução de problemas de álgebra computacional.

2 Análise dos Métodos

Para as análises do próximos métodos, levaremos em consideração a equação 2 escrita no modelo da equação 1.

$$y'(t) = 1 - t + 4y(t) \tag{2}$$

Eq. 2: Exemplo de EDO para cálculo das instâncias dos métodos numéricos.

que tem como solução:

$$y(t) = c_1 e^{4t} + \frac{t}{4} - \frac{3}{16} \tag{3}$$

Eq. 3: Solução da equação 2.

e para um PVI de $y(0) = 0$, a equação 3 torna-se:

$$y(t) = \frac{3}{16}e^{4t} + \frac{t}{4} - \frac{3}{16} \tag{4}$$

Eq. 4: Solução da equação 2 para o PVI: $y(0) = 0$.

Todos os códigos referentes aos métodos se encontram com o arquivo *main.py* que vai junto com este relatório. E cada implementação escrita na função com o nome referente ao método.

2.1 Euler

O método de Euler é o mais simples de todos eles, que necessita de um PVI, e seus valores são gerados a partir desse primeiro PVI e utilizando da própria equação 2 para estimar os próximos pontos, com o valor da taxa de variação deles, ou seja, a própria derivada. O método de euler explicitado é:

$$y_{n+1} = y_n + hf(t_n, y_n) \tag{5}$$

Eq. 5: Método de Euler

Onde h é o passo dado pelo método, o intervalo em que a solução é estimada; $f(t_n, y_n)$ é o valor calculado pela própria equação 2 e y_n o valor atual, que no caso inicial é o próprio PVI, e y_{n+1} é o próximo valor estimado. As explicações aqui feitas também servirão para os outros métodos.

2.1.1 Exemplo de Entrada

Método	y0	t0	h	Quantidade Passos	Função
euler	0	0	0.1	20	1 - t + 4y

2.1.2 Método de Aproximação

Com o auxílio da biblioteca sympy, e suas funções *subs e sympify* é possível reconhecer a expressão albégrica como um objeto a ser iterado por python, e assim iterar pelos valores da função.

2.1.3 Exemplo de Saída

Os valores de saída são gerados num arquivo texto. Aqui se encontram os valores em um gráfico. Na figura 1.

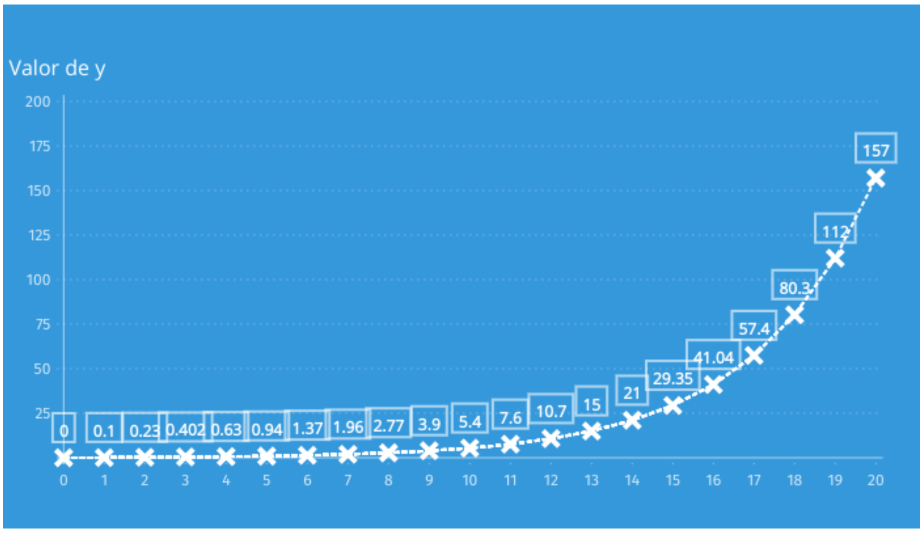

Figura 1: Saída dos valores aproximados com o método de euler.

2.2 Euler Inverso

O método de Euler Inverso é uma variante do método de Euler, calculando o próximo valor utlizando de uma previsão a partir da próxima derivada. O método de euler inverso explicitado é:

$$y_{n+1} = y_n + hf(t_{n+1}, y_{n+1}) \tag{6}$$

Esta equação está na forma implícita, mas que pode ser reformulada para uma fórmula explícita, se ajustada algebricamente. Mas em com a biblioteca sympy é possível utilizar a ferramenta Solve e resolver equações diretamente sem os ajustes para uma expressão explícita.

2.2.1 Exemplo de Entrada

Método	y0	t0	h	Quantidade Passos	Função
euler_inverso	0	0	0.1	20	1 - t + 4y

2.2.2 Método de Aproximação

Com o auxílio da biblioteca sympy, e suas funções *subs e sympify* é possível reconhecer a expressão albégrica como um objeto a ser iterado por python, e assim iterar pelos valores da função. Resolvendo a equação implícita.

2.2.3 Exemplo de Saída

Os valores de saída são gerados num arquivo texto. Aqui se encontram os valores em um gráfico. Na figura 2.

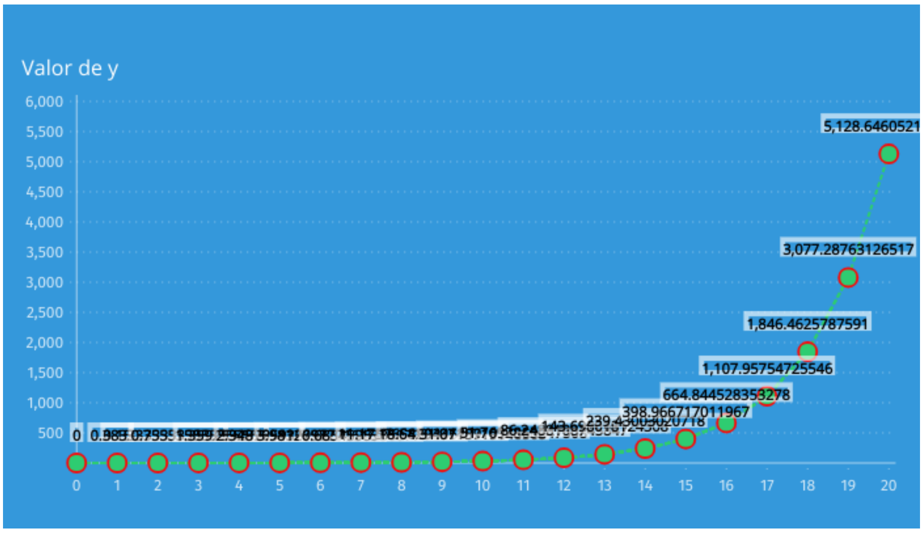

Figura 2: Saída dos valores aproximados com o método de Euler Inverso.

2.3 Euler Aprimorado

O método de Euler Aprimorado é uma variante do método de Euler, que utiliza de um ajuste do próximo passo a partir do cálculo da previsão do próximo ponto, utilizando o método de Euler. O método de euler Aprimorado explicitado é:

$$y_{n+1} = y_n + \frac{h(f(t_n, y_n) + f(t_{n+1}, y_{n+1}))}{2} \tag{7}$$

2.3.1 Exemplo de Entrada

Método	y0	t0	h	Quantidade Passos	Função
euler_aprimorado	0	0	0.1	20	1 - t + 4y

2.3.2 Método de Aproximação

O cálculo é feito no mesmo modelo que o Método de Euler.

2.3.3 Exemplo de Saída

Os valores de saída são gerados num arquivo texto. Aqui se encontram os valores em um gráfico. Na figura 3.

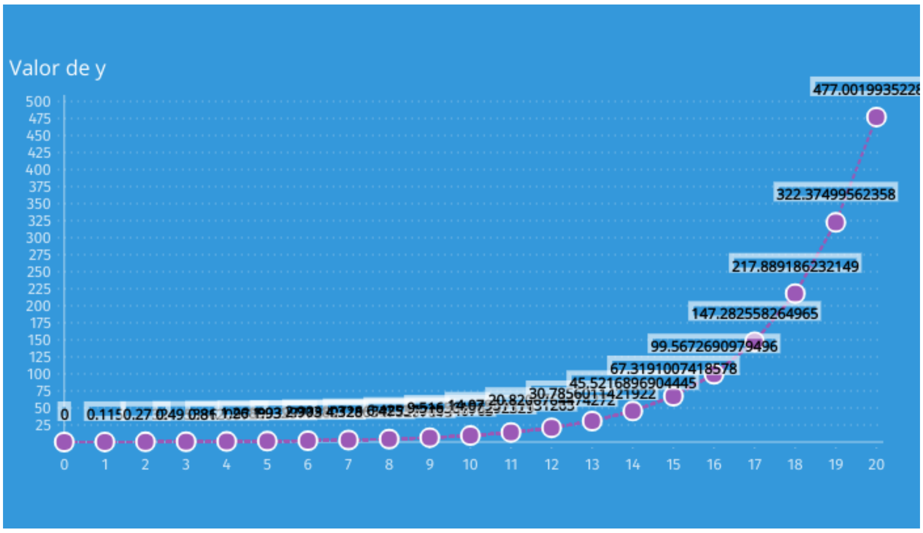

Figura 3: Saída dos valores aproximados com o método de Euler Aprimorado.

2.4 Runge Kutta

O método de Runga Kutta estima o próximo valor a partir de uma média ponderada de alguns pontos em relação ao intervalo analisado. No caso estaremos levando em consideração o método de Runge-Kutta para o 4º grau. O método de Runge Kutta explicitado é:

$$
\begin{aligned}
y_{n+1} = y_n + \frac{h(k_1 + 2k_2 + 2k_3 + k4)}{6} \\
k_1 = f(t_n, y_n) \\
k_2 = f(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_1) \\
k_3 = f(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_2) \\
k_4 = f(t_n + h, y_n + hk_3)
\end{aligned}
\tag{8}
$$

2.4.1 Exemplo de Entrada

Método	y0	t0	h	Quantidade Passos	Função
runge_kutta	0	0	0.1	20	1 - t + 4y

2.4.2 Método de Aproximação

O cálculo é feito no mesmo modelo que o Método de Euler.

2.4.3 Exemplo de Saída

Os valores de saída são gerados num arquivo texto. Aqui se encontram os valores em um gráfico. Na figura 4.

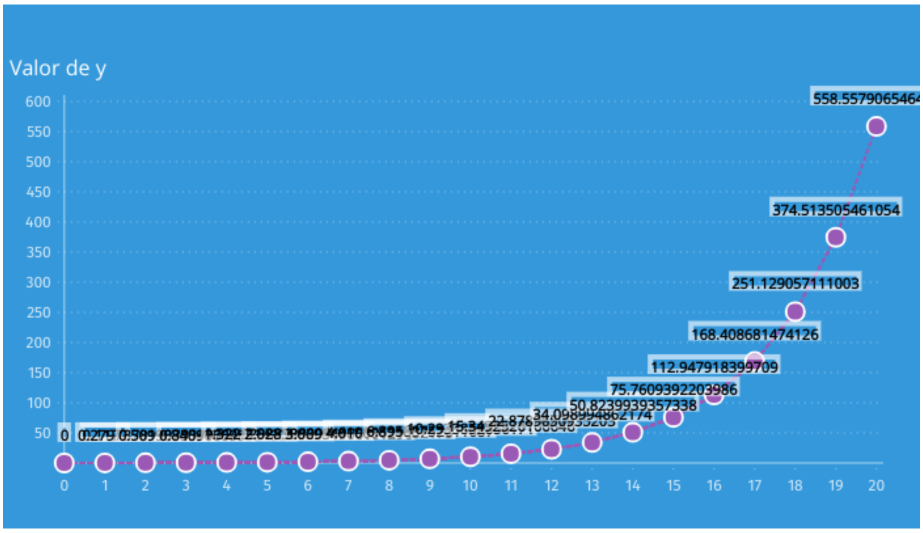

Figura 4: Saída dos valores aproximados com o método de Runge Kutta.

2.5 Adams-Bashforth

O método de Adams-Bashforth é um método de passo múltiplos explícito que estima o próximo valor a partir do conhecimento de etapas anteriores já geradas, ao invés do método de Euler, que descarta as informações anteriores. Este método tem vários graus, e coeficientes multiplicativos para cada um destes, analisando a tabela deste método nos requisitos do projeto, podemos escrever os métodos de Adams Bashforth no modelo:

$$\begin{aligned} SegundaOrdem : y_{n+1} = y_n + h(\beta_1 f(t_n, y_n) + \beta_2 f(t_{n-1}, y_{n-1})) \\ TerceiraOrdem : y_{n+1} = y_n + h(\beta_1 f(t_n, y_n) + \beta_2 f(t_{n-1}, y_{n-1} + \beta_3 f(t_{n-2}, y_{n-2})) \end{aligned} \tag{9}$$

Se manter o padrão chega-se até o Adams-Bashforth de grau 8.

2.5.1 Exemplo de Entrada

Para o método de Adams-Bashforth existiam dois modelos de se executado: Utilizando uma lista de valores pre-estabelecidade ou utilizando um dos métodos anteriores para gerar os pontos de PVI.

Entrada com a lista dos elementos:

Método	y(Lista)	t0	h	Quantidade Passos	Função	ordem
adam_bashforth	[...]	0	0.1	20	1 - t + 4y	5

Método	y0)	t0	h	Quantidade Passos	Função	ordem
adam_bashforth_euler	0	0	0.1	20	1 - t + 4y	5

2.5.2 Método de Aproximação

O cálculo é feito no mesmo modelo que o Método de Euler.

2.5.3 Exemplo de Saída

Os valores de saída são gerados num arquivo texto. Aqui se encontram os valores em um gráfico. Na figura 5.

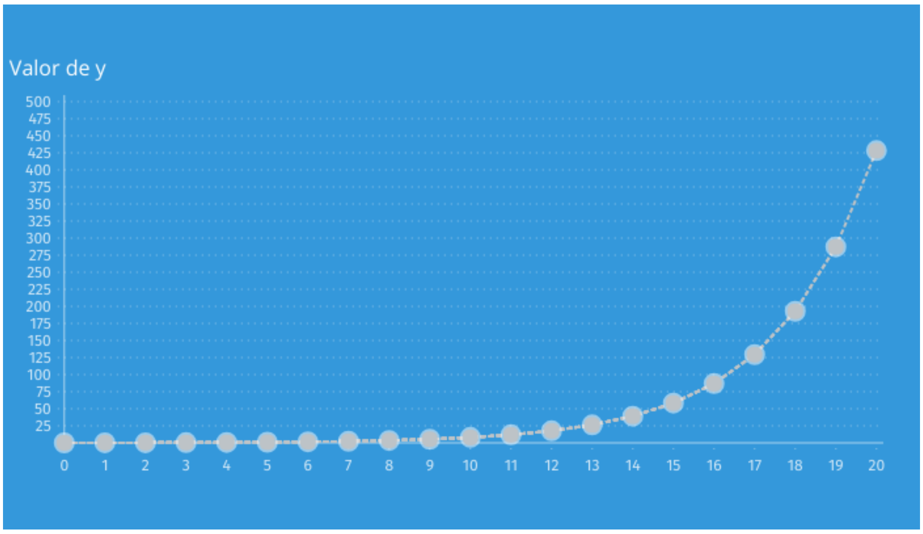

Figura 5: Saída dos valores aproximados com o método de Adam Bashforth.

2.6 Adams-Multon

O método de Adams-Multon é um método de passo múltiplos implícito que estima o próximo valor a partir do conhecimento de etapas anteriores já geradas, ao invés do método de Euler, que descarta as informações anteriores. Este método tem vários graus, e coeficientes multiplicativos para cada um destes, analisando a tabela deste método nos requisitos do projeto, podemos escrever os métodos de Adams Multon no modelo:

$$\begin{aligned} SegundaOrdem: y_{n+1} = y_n + h(\beta_1 f(t_{n+1}, y_{n+1}) + \beta_2 f(t_n, y_n)) \\ TerceiraOrdem: y_{n+1} = y_n + h(\beta_1 f(t_{n+1}, y_{n+1}) + \beta_2 f(t_n, y_n + \beta_3 f(t_{n-1}, y_{n-1})) \end{aligned} \tag{10}$$

Se manter o padrão chega-se até o Adams-Multon de grau 8.

2.6.1 Exemplo de Entrada

Para o método de Adams-Multon existiam dois modelos de se executado: Utilizando uma lista de valores pre-estabelecidade ou utilizando um dos métodos anteriores para gerar os pontos de PVI.

Entrada com a lista dos elementos:

Método	y(Lista)	t0	h	Quantidade Passos	Função	ordem
adam_multon	[...]	0	0.1	20	1 - t + 4y	6

Método	y0)	t0	h	Quantidade Passos	Função	ordem
adam_multon_euler	0	0	0.1	20	1 - t + 4y	6

2.6.2 Método de Aproximação

O cálculo é feito no mesmo modelo que o Método de Euler Inverso.

2.6.3 Exemplo de Saída

Os valores de saída são gerados num arquivo texto. Aqui se encontram os valores em um gráfico. Na figura 6.

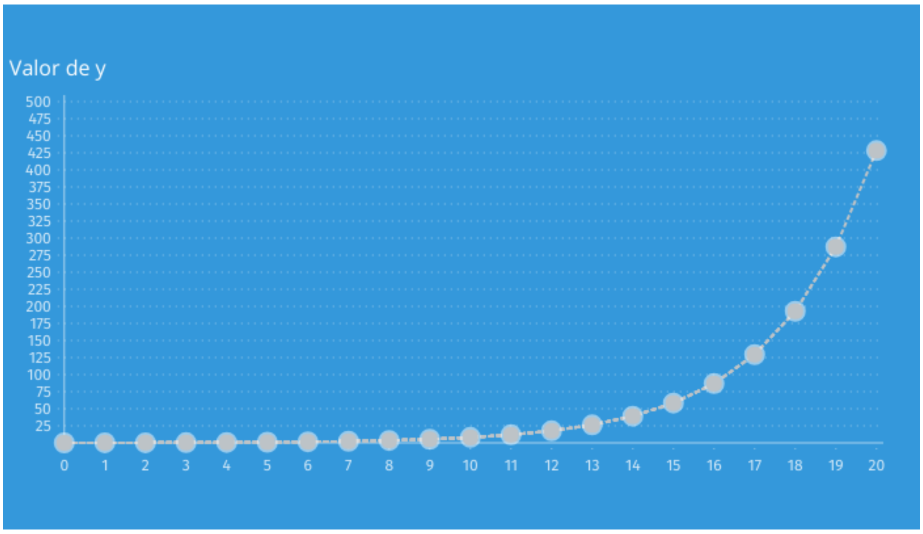

Figura 6: Saída dos valores aproximados com o método de Adam Multon.

3 Análise dos Resultados

Para analisar os resultados, se faz necessário olhar os pontos que deveriam ser gerados pela solução exata, vista na equação 4. Utilizando a mesma quantidade de passos e o intervalos. Foi gerada a curva:

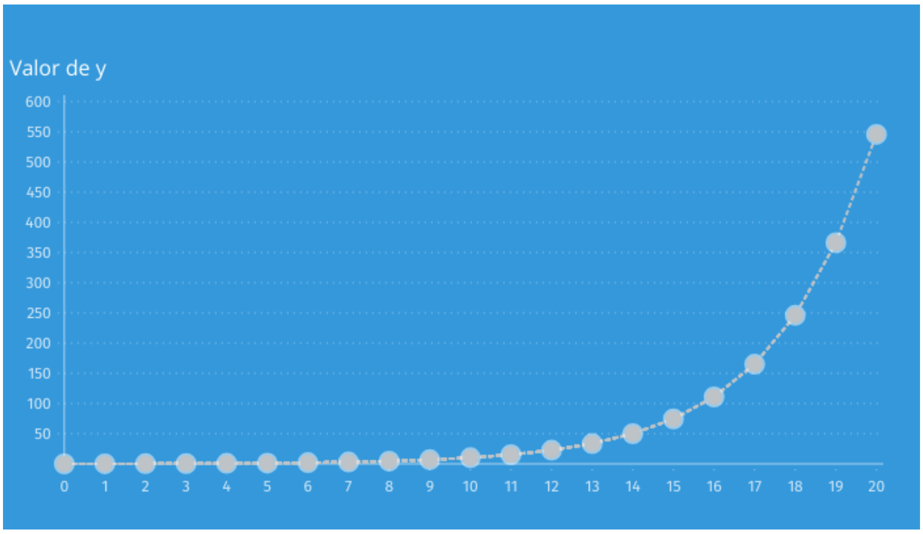

Figura 7: Pontos gerados pela solução exata.

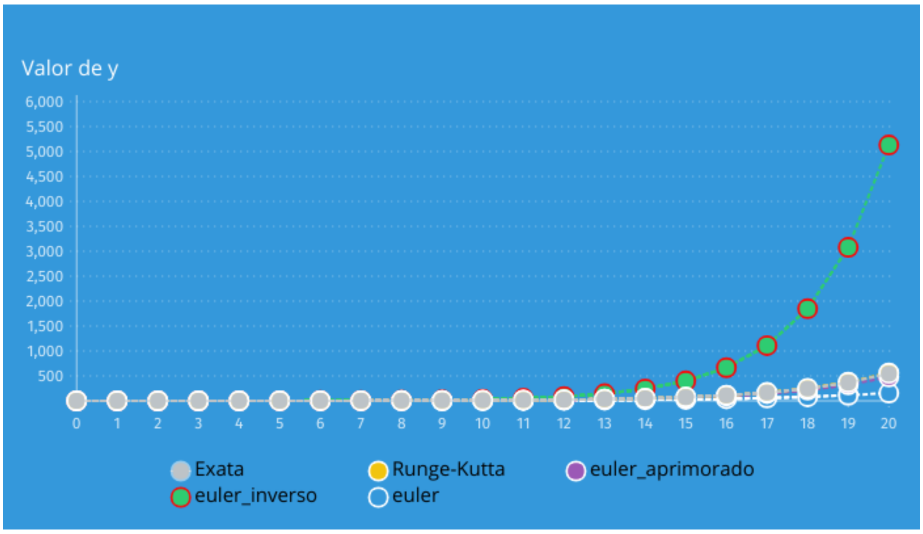

Figura 8: Comparação dos métodos de passos únicos e a solução exata.

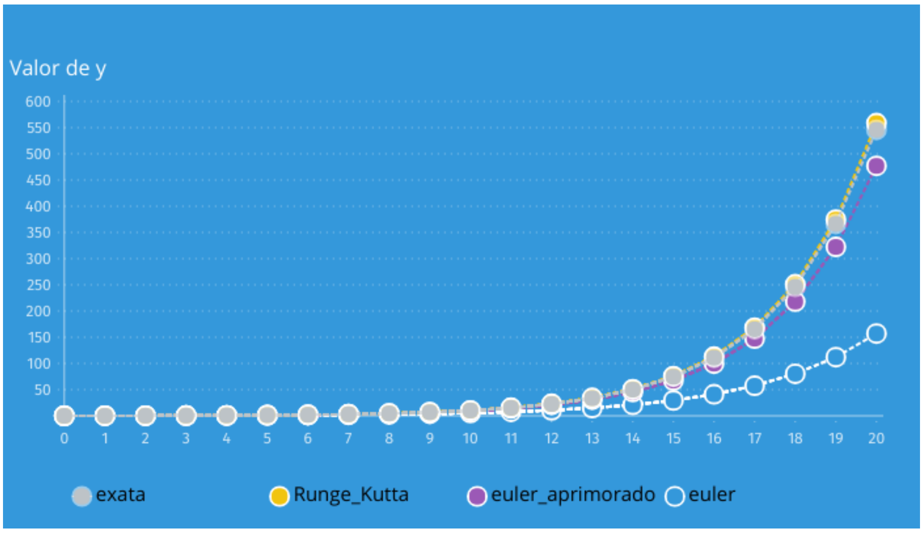

Figura 9: Comparação dos métodos de passos únicos, sem euler inverso, e a solução exata.

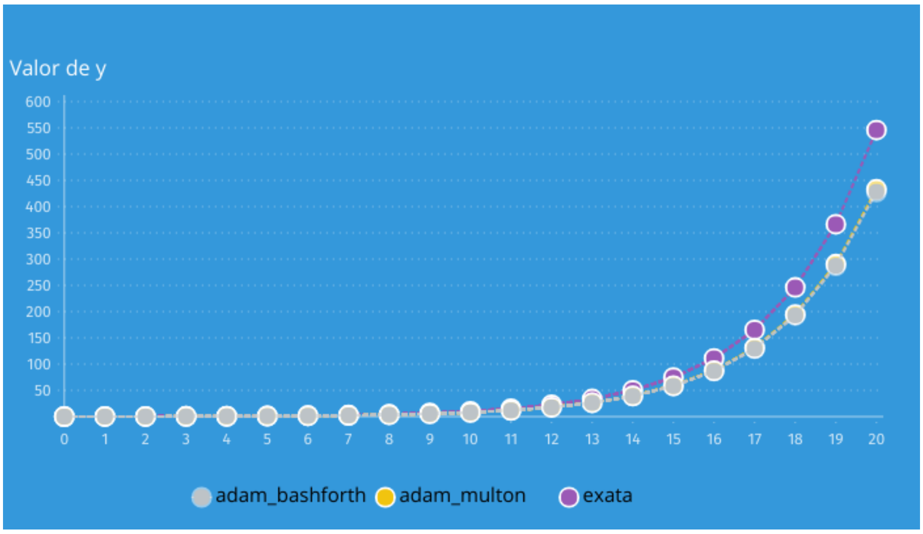

Figura 10: Comparação dos métodos de passos múltiplos e a solução exata.

Foi gerado um gráfico de comparação sem o euler inverso, para que fosse possível uma análise mais próxima entre os outros métodos, já que o euler inverso se distanciou da solução exata.

Dentre os métodos de passos únicos, Runge Kutta se destacou, com erros bem menores, o que era de se esperar, com uma maior quantidade de previsão e correção. Foi perceptível que o método de adams-bashforth foi bem mais preciso do que o multon, e foi o que mais se aproximou dos resultados da solução exata.