Projeto da disciplina de Métodos Numéricos Computacionais

Mikael Vidal da Silva
Centro de Informática UFPE 2018.2
Disciplina IF816-EC

Recife
Outubro 2018

Sumario

0.1 Introdução

A realização do projeto da disciplina de métodos numericos computacionais do Cin-UFPE ministrada pelo professor Ricardo Martins visa implementar os métodos numéricos estudados em sala de aula, fazendo assim com que seja possível uma análise prática de seus resultados e métodos. Nesta primeira parte do projeto da disciplina, foi preciso apenas realizar as implementações destes métodos e uma descrição detalhada de suas implementações que serão encontradas neste relatório.

0.2 Implementação

0.2.1 Considerações iniciais

Como foi dito anteriormente, o objetivo deste relatório é demonstrar como foi realizado o projeto, com isso, é importante ressaltar quais foram os métodos requeridos na especificação do projeto, descritos abaixo:

1. Euler
2. Euler inverso
3. Euler aprimorado
4. Runge-Kutta (4 Ordem)
5. Adam-Bashforth (2 a 8 Ordem)
6. Adam-Bashforth por Euler
7. Adam-Bashforth por Euler inverso
8. Adam-Bashforth por Euler aprimorado
9. Adam-Bashforth por Runge-Kutta
10. Adam-Moulton (2 a 8 Ordem)
11. Adam-Moulton por Euler
12. Adam-Moulton por Euler inverso
13. Adam-Moulton por Euler aprimorado
14. Adam-Moulton por Runge Kutta
15. Formula inversa (2 a 6 Ordem)
16. Formula inversa por Euler
17. Formula inversa por Euler inverso
18. Formula inversa por Euler aprimorado
19. Formula inversa por Runge-Kutta

Dentre estes métodos também foi oferecido a oportunidade de implementar quesitos bônus, mas todos dentro do uso dos métodos citados acima.

0.2.2 métodos implementados

Infelizmente não foi possivel implementar todos os quesitos bônus. Os quesitos que não foram possiveis de implementar foram os de Adam-Moulton sem o método da previsão com Adam-Bashforth, o método de Adam-Bashforth sem o método da previsão, formula inversa sem o método da previsão, uma função que calcula para qualquer ordem N nos métodos de Adam-Moulton, Adam-Bashforth e da formula inversa, uma função que pudesse calcular para todos os métodos e uma função que compara todos os métodos com seus valores exatos, os outros requisitos necessarios e bônus indicados na especificação do projeto foram implementados da melhor maneira possivel dentro de meus conhecimentos teóricos em relação ao assunto abordado, e práticos, com a utilização da linguagem de programação Python e as bibliotecas Sympy e Matplotlib, o código foi comentando com suas devidas indicações de passo a passo com o intuito de melhorar a legibilidade.

0.2.3 Exemplos de leitura

Os exemplos fornecidos para as análises consistem do seguinte padrão:

1. euler 0 0 0.1 20 1-t+4*y *(Para métodos de passos simples)*
2. adam bashforth 0.0 0.1 0.23 0.402 0.6328 0.94592 0 0.1 20 1-t+4*y 6 *(Para métodos de passos múltiplos)*
3. adam bashforth by euler 0 0 0.1 20 1-t+4*y 6 *(Para métodos de passos méltiplos com um método de passo simples para descobrir seus valores iniciais de* ***y*** *e* ***t*** *iniciais)*

Sendo o seu primeiro valor o nome do método a ser executado, seguido de seu valor em y(t0) para o caso de apenas um ponto formecido, ou para N valores de $y(t_n)$, o próprio valor de t, este sim é fornecido apenas uma vez, o valor de h, o número de passos que se deseja verificar o método, sua função e no caso dos métodos de passos múltiplos é fornecido a ordem que se deseja executar o método.

0.3 Métodos

0.3.1 Método de Euler

Explicando o método

O método de Euler e todos os outros métodos implementados possuem algo em comum, todos visam aproximar através de suas equações ao valor do P.V.I (Problema do Valor Inicial) dada por dy/dt = f(t,y).

Neste tópico iremos abordar o método de Euler dado pela Eq. $\mathbf{y}_{n+1} = y_n + hf_n$

O objetivo do método é calcular quantas vezes seja necessario n = 0,1,2.., utilizando-se do resultado do passo anterior y_n para obter f_n e encontrar o valor para o próximo passo em $\mathbf{y}_{n+1}$, onde f_n é a função que está sendo analisada.

Exemplo de leitura

O exemplo de leitura aqui citado é o mesmo disponibilizado na especificação do projeto [euler 0 0 0.1 20 1-t+4*y]

Pontos gerados

Os pontos gerados para $\mathbf{y}_{n+1}$ foram salvos no arquivo salvar.txt e foram gerados com os resultados a seguir, organizado de forma que seja visualizado primeiro o nome do método, o primeiro valor de y e t, o valor de h e o numero do passo e seu valor de y_n.

1. Método de euler
2. y(0) = 0
3. h = 0.1
4. 0 0.0
5. 1 0.100000000000000
6. 2 0.230000000000000
7. 3 0.402000000000000
8. 4 0.632800000000000
9. 5 0.945920000000000
10. 6 1.37428800000000
11. 7 1.96400320000000
12. 8 2.77960448000000
13. 9 3.91144627200000
14. 10 5.48602478080000
15. 11 7.68043469312000
16. 12 10.7426085703680
17. 13 15.0196519985152
18. 14 20.9975127979213
19. 15 29.3565179170898

20. 16 41.0491250839257

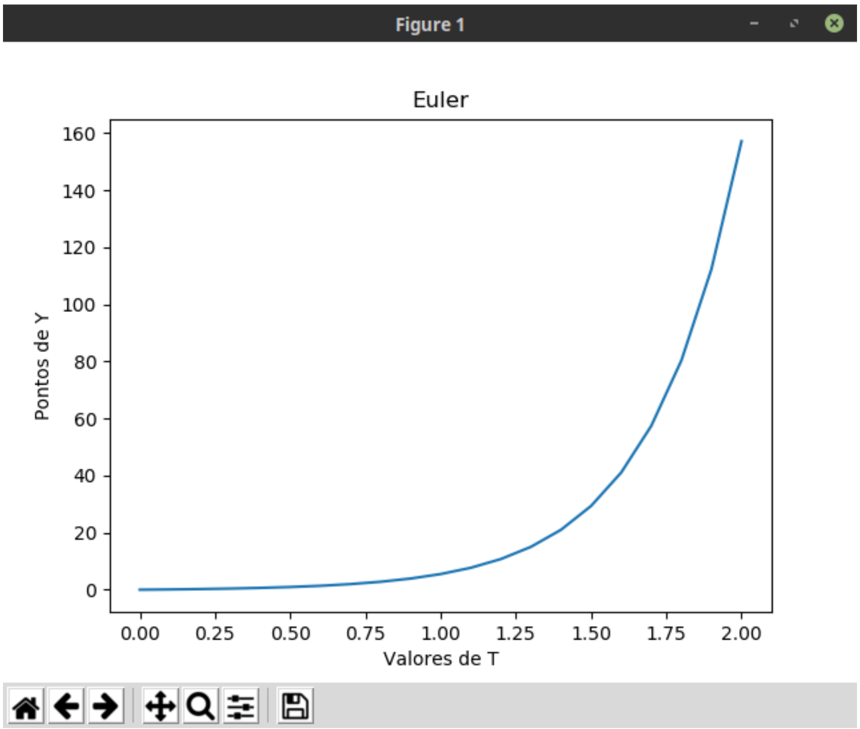
21. 17 57.4087751174960

22. 18 80.3022851644944

23. 19 112.343199230292

24. 20 157.190478922409

Grafico



Método 1: Euler

OBS: *Este foi o gráfico gerado no método para as condições já especificadas anteriormente utilizando-se de algumas funções da biblioteca matplotlib, isso vale para todos os gráficos dos métodos especificados neste projeto.*

0.3.2 Método de Euler inverso

Explicando o método com previsão

O método de Euler inverso é uma adaptação do método de Euler, este método é dado pela Eq. $\mathbf{y}_{n+1} = y_n + hf_{n+1}$

Primeiro, é executado o método de euler para poder prever um $\mathbf{y}_{n+1}$, quando esse valor for encontrado, ele será utilizado para calcular o f_{n+1} junto com o t_{n+1}, ou seja $f_{n+1} = (t_{n+1}, y_{n+1})$, encontrado esse valor, então finalmente o método fornece o valor definitivo para a soma com o y_n, obtendo assim o resultado de $\mathbf{y}_{n+1}$.

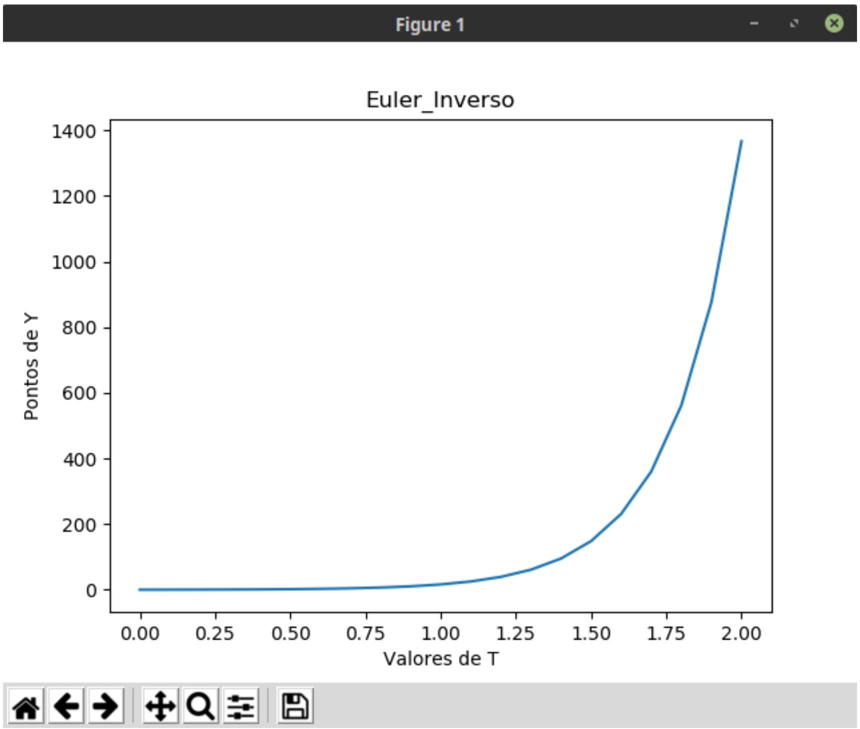
Exemplo de leitura

O exemplo de leitura foi o mesmo fornecido na especificação do projeto [euler inverso 0 0 0.1 20 1-t+4*y]

Pontos gerados

1. Método de Euler Inverso
2. y(0) = 0
3. h = 0.1
4. 0 0.0
5. 1 0.130000000000000
6. 2 0.318800000000000
7. 3 0.599328000000000
8. 4 1.02295168000000
9. 5 1.66980462080000
10. 6 2.66489520844800
11. 7 4.20323652517888
12. 8 6.58904897927905
13. 9 10.2969164076753
14. 10 16.0671895959735
15. 11 25.0548157697187
16. 12 39.0615126007611
17. 13 60.8979596571874
18. 14 94.9488170652123
19. 15 148.054154621731
20. 16 230.884481209901
21. 17 360.085790687445
22. 18 561.625833472414
23. 19 876.014300216966
24. 20 1366.44630833847

Grafico



Método 2: Euler inverso

Explicando o método sem previsão

O método de euler inverso sem o método da previsao com o método de Euler foi realizada de forma a igualar a zero a Eq. $\underline{y_n + hf_{n+1} \mathbf{-y}_{n+1}} = 0$, dessa forma, utilizando-se de uma função fornecida pela biblioteca sympy é possivel resolver a equação sem o método da previsao.

Exemplo de leitura

Como este quesito é bônus, não tendo nenhum exemplo na especificação do projeto, foi decidido que a forma que seria feita a leitura para está função seria da seguinte maneira [euler inverso sem previsao 0 0 0.1 20 1-t+4*y]

Pontos gerados

1. Método de Euler Inverso Sem Previsao
2. y(0) = 0
3. h = 0.1
4. 0 0.0
5. 1 0.150000000000000
6. 2 0.383333333333333
7. 3 0.755555555555556
8. 4 1.35925925925926
9. 5 2.34876543209877

10. 6 3.98127572016462

11. 7 6.68545953360770

12. 8 11.1757658893462

13. 9 18.6429431489103

14. 10 31.0715719148505

15. 11 51.7692865247508

16. 12 86.2488108745847

17. 13 143.698018124308

18. 14 239.430030207180

19. 15 398.966717011967

20. 16 664.844528353278

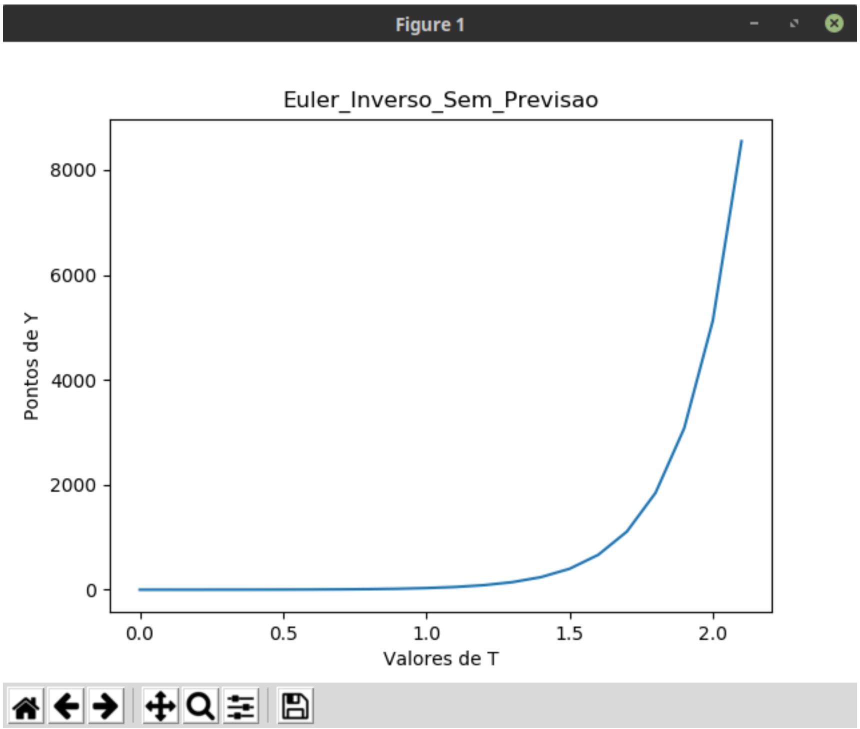
21. 17 1107.95754725546

22. 18 1846.46257875910

23. 19 3077.28763126517

24. 20 5128.64605210862

Grafico



Método 3: Euler inverso sem previsao

0.3.3 Método de Euler aprimorado

Explicando o método com previsao

Este método é um melhoramento do método de Euler. Nele, é preciso calcular o $\mathbf{y}_{n+1}$ pelo método de Euler, com isso, vamos utilizar seu valor e o de t_{n+1} para podermos calcular $f(t_{n+1},y_{n+1})$. Após realizar esse calculo, podemos utilizar do ponto y_n anterior e seu t_n para descobrirmos o $f(t_n,y_n)$, com isso a ideia do melhoramento do método de Euler é colocada em prática, pois tendo os extremos dos valores que se poderia ter apenas prevendo o ft_{n+1},y_{n+1} e os valores do ponto anterior $f(t_n,y_n)$ tiramos uma media entre seus valores, fazendo-se então $(f_n + f_{n+1})/2)$, a Eq. final é $\underline{\mathbf{y}_{n+1} = y_n + (h/2)(f_n + f_{n+1})}$

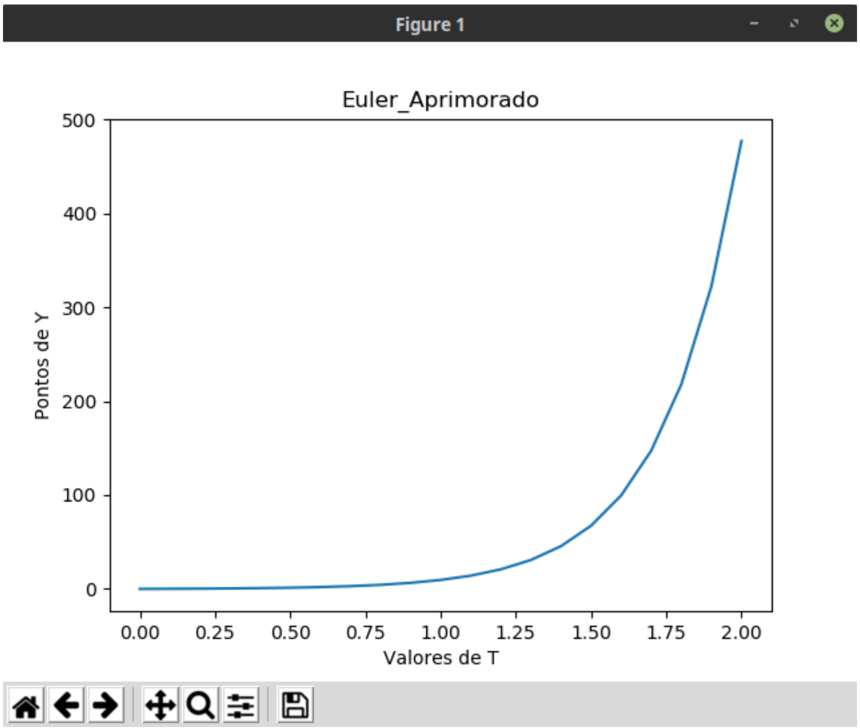
Exemplo de leitura

O exemplo para este método foi fornecido na especificação da seguinte forma [euler aprimorado 0 0 0.1 20 1-t+4*y]

Pontos gerados

1. Método de Euler Aprimorado
2. y(0) = 0
3. h = 0.1
4. 0 0.0
5. 1 0.115000000000000
6. 2 0.273200000000000
7. 3 0.495336000000000
8. 4 0.812097280000000
9. 5 1.26890397440000
10. 6 1.93297788211200
11. 7 2.90380726552576
12. 8 4.32863475297813
13. 9 6.42537943440763
14. 10 9.51656156292329
15. 11 14.0795111131265
16. 12 20.8206764474272
17. 13 30.7856011421922
18. 14 45.5216896904445
19. 15 67.3191007418578
20. 16 99.5672690979496
21. 17 147.282558264965
22. 18 217.889186232149
23. 19 322.374995623580
24. 20 477.001993522899

Grafico



Método 4: Euler aprimorado

Explicando o método sem previsao

O mesmo método utilizado para se obter o método de euler inverso sem a previsao por euler ocorre neste método, precisao fazer com que a quação seja comparada a zero, e novamente, utilizar-se da função solve() da biblioteca sysmpy para resolver a equação, desta forma conseguimos o método de Euler aprimorado sem a previsao por Euler, obtemos então $y_n + (h/2)(f_n + f_{n+1})$ - $\mathbf{y}_{n+1} = 0$

Exemplo de leitura

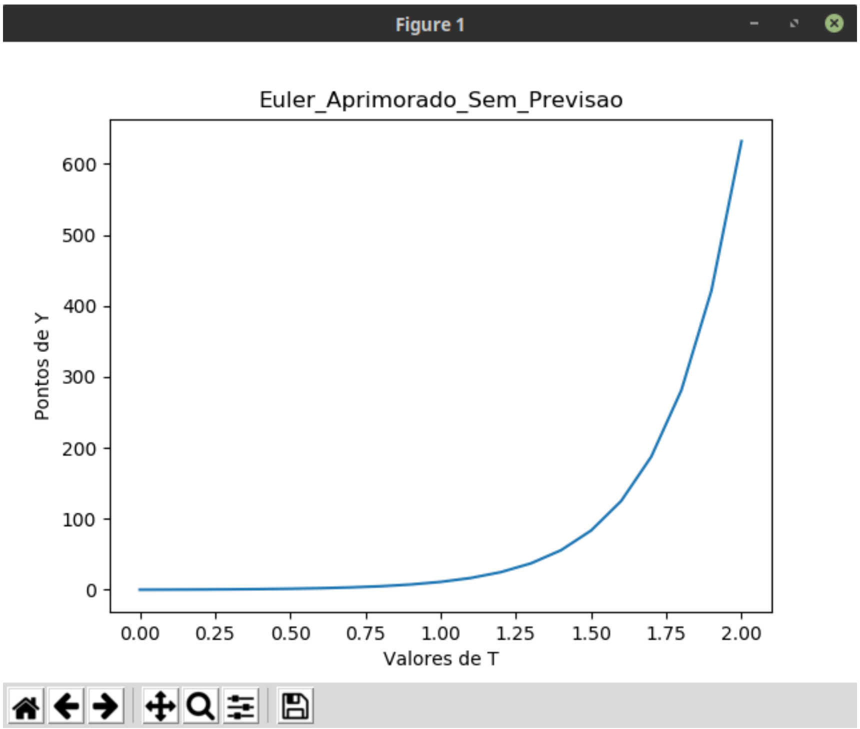
Novamente, este quesito era um bõnus, sendo assim, não foi fornecido nenhum valor de base na especificação do projeto, então ficou decidido que a leitura no arquivo seria feita da seguinte forma [euler aprimorado sem previsao 0 0 0.1 20 1-t+4*y]

Pontos gerados

1. Método de Euler Aprimorado Sem Previsao
2. y(0) = 0
3. h= 0.1
4. 0 0.0
5. 1 0.120000000000000
6. 2 0.287500000000000
7. 3 0.526250000000000
8. 4 0.871875000000000
9. 5 1.37781250000000

10. 6 2.12421875000000
11. 7 3.23132812500000
12. 8 4.87949218750000
13. 9 7.33923828125000
14. 10 11.0163574218750
15. 11 16.5195361328125
16. 12 24.7618041992188
17. 13 37.1127062988282
18. 14 55.6265594482423
19. 15 83.3848391723634
20. 16 125.009758758545
21. 17 187.434638137818
22. 18 281.059457206727
23. 19 421.484185810091
24. 20 632.108778715135

Grafico



Método 5: Euler aprimorado sem previsao

0.3.4 Método de Runge Kutta

Explicando o método

No método anterior nos verificamos que o método tira uma media dos extremos de pontos de y_n e y_{n+1}, se tornando um pouco mais preciso que o método de Euler. O método que iremos analisar neste tópico também tira uma media, mas desta vez é uma média ponderada para valores de $f(t_n,y_n)$ em pontos distintos dentro do intervalor de $t_n \leq t \leq t_{n+1}$.

A Eq. $\underline{\mathbf{y}_{n+1} = y_n + (h/6)(k_{n1} + 2k_{n2} + 2k_{n3} + k_{n4})}$ é distribuida em alguns valores de k que será definido abaixo.

1. $k_{n1} = f(t_n,y_n)$,
2. $k_{n2} = f(t_n + (0,5)h, y_n + (0,5)hk_{n1})$,
3. $k_{n3} = f(t_n + (0,5)h, y_n + (0,5)hk_{n2})$,
4. $k_{n4} = f(t_n + h, y_n + hk_{n1})$,

Exemplo de leitura

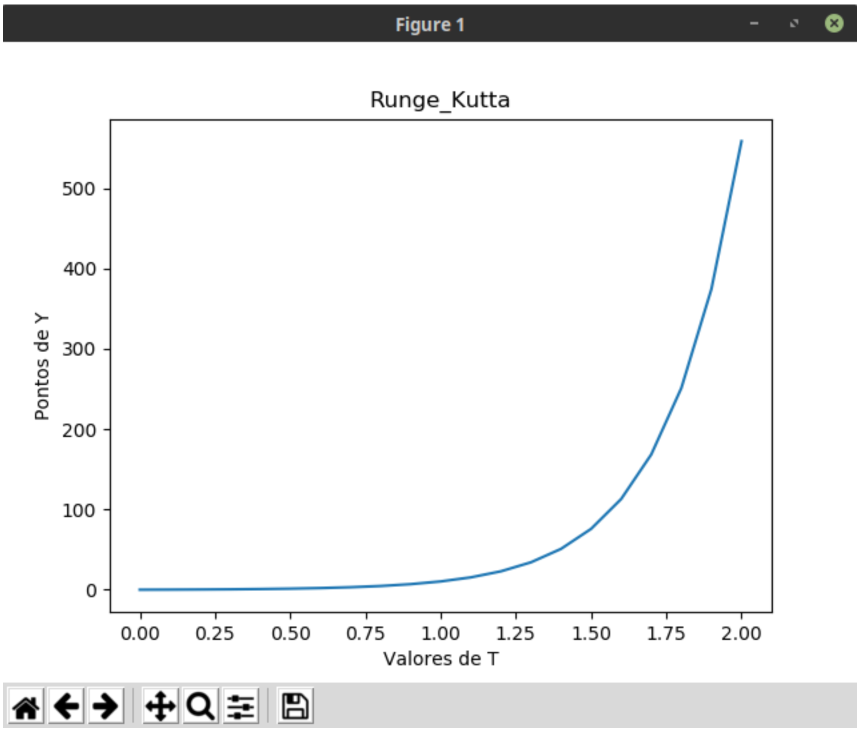
O exemplo de leitura fornecido na especificação do projeto foi [runge kutta 0 0 0.1 20 1-t+4*y]

Pontos gerados

1. Método de Runge Kutta
2. y(0) = 0
3. h = 0.1
4. 0 0.0
5. 1 0.117200000000000
6. 2 0.279737813333333
7. 3 0.509907554076445
8. 4 0.840966095334301
9. 5 1.32252382328002
10. 6 2.02858620464758
11. 7 3.06954966101296
12. 8 4.61009621432173
13. 9 6.89588752611087
14. 10 10.2933852856171
15. 11 15.3492526100646
16. 12 22.8789650935203
17. 13 34.0989948621740
18. 14 50.8239939357338
19. 15 75.7609392203986
20. 16 112.947918399709
21. 17 168.408681474126

22. 18 251.129057111003
23. 19 374.513505461054
24. 20 558.557906546436

Grafico



Método 6: Runge Kutta

Explicando o método de Runge-Kutta para 5 ordem

O método é análogo ao de 4 ordem, o que altera são os seus coeficientes e as formas com que os k_n são calculados,sendo sua Eq. $\underline{\mathbf{y}_{n+1} = y_n + (h/90)(7k_1 + 32k_3 + 12k_4 + 32k_5 + 7k_6)}$ o método foi calculado a partir desses valores:

1. $k_1 = f(t,y)$
2. $k_2 = f(t + (h/4), y + (k_1 h)/4)$
3. $k_3 = f(t + (h/4), y + (k_1 h)/8 + (k_2 h)/8)$
4. $k_4 = f(t + (h/2), y - (k_2 h)/2 + hk_3)$
5. $k_5 = f(t + (3h/2), y + (3k_1 h)/16 + (9k_3 h)/16)$
6. $k_6 = f(t + h, y - (3k_1 h)/7 + (2k_2 h)/7 + (12k_3 h)/7 - (12k_4 h)/7 + (8k_5 h)/7)$

Exemplo de leitura

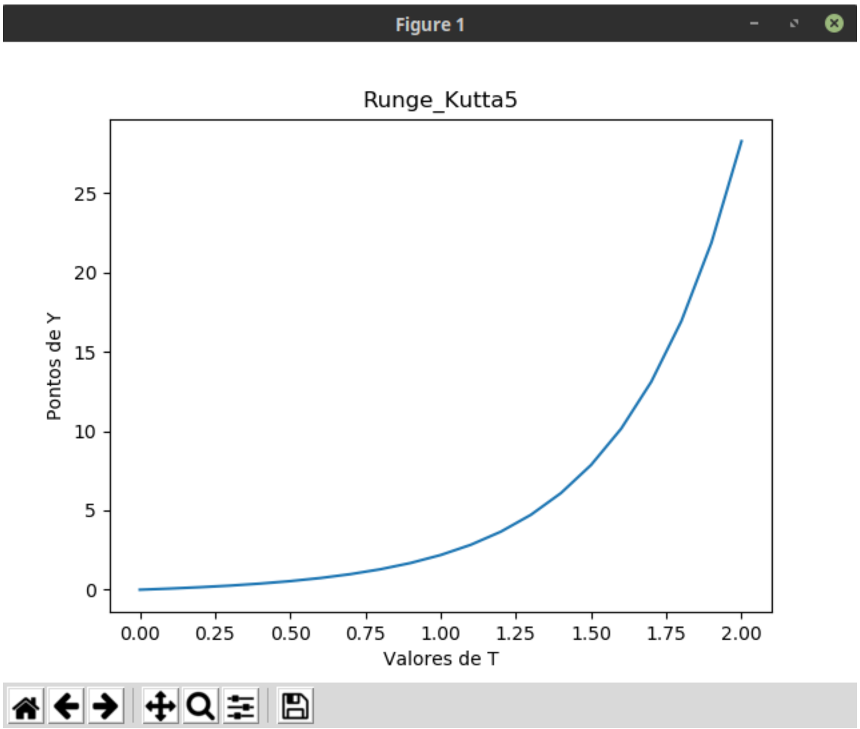
Como este tópico se apresenta como um quesito bõnus, a forma de leitura foi decidida durante o desenvolvimento do problema, alterando pouca coisa em relação as outras leituras disponibilizadas na especificação, segue um exemplo:

1. runge kutta5 0 0 0.1 20 1-t+4*y *Indicando sua ordem, se fosse apenas runge kutta, estaria indicando a de 4 ordem, que já é implicita e não precisa ser especificada.*

Pontos gerados

1. Metodo de Runge Kutta5
2. y(0) = 0
3. h = 0.1
4. 0 0.0
5. 1 0.0713957250000000
6. 2 0.156514184740487
7. 3 0.259414232677416
8. 4 0.385355233075154
9. 5 0.541152143104598
10. 6 0.735635619648726
11. 7 0.980248214587074
12. 8 1.28981691024395
13. 9 1.68355415216118
14. 10 2.18635496318461
15. 11 2.83047771256473
16. 12 3.65772201595752
17. 13 4.72225080563706
18. 14 6.09424710090837
19. 15 7.86465236287710
20. 16 10.1513063401024
21. 17 13.1069029322826
22. 18 16.9292992062410
23. 19 21.8748735698196
24. 20 28.2758349709374

Grafico



Método 7: Runge Kutta de 5 Ordem
Verificar tópico da conclusão

0.3.5 Método de Adam-Bashforth (2 a 8 Ordem)

Explicando o método

Deste tópico em diante iremos analisar os métodos de passos multiplos, esses métodos podem conseguir diversos tipos de aproximações dependendo do numero de pontos que se usa para criar uma estimativa melhor para somarmos com $\mathbf{y}_{n+1}$

Neste tópico iremos analisar o método de Adam-Bashforth, como na especificação do projeto foi solicitado que fossem de ordem 2 até a ordem 8, será explicado de uma forma geral os objetivos do método, e depois, mostrado os resultados gerados para o método de 2 até 8 ordem.

Segue abaixo as Eq. utilizadas para implementar o método de Adam-Bashforth:

1. $\mathbf{y}_{n+1} = y_n + (h/2)(3f_n - f_{n-1})$
2. $\mathbf{y}_{n+1}=y_n+(h/12)(23f_n - 16f_{n-1} + 5f_{n-2})$
3. $\mathbf{y}_{n+1} = y_n + (h/24)(55F_n - 59f_{n-1} + 37f_{n-2} - 9f_{n-3})$
4. $\mathbf{y}_{n+1}=y_n+(h/720)(1901f_n - 2774f_{n-1} + 2616f_{n-2} - 1274f_{n-3} + 251f_{n-4})$
5. $\mathbf{y}_{n+1}=y_n+(h/1440)(4277f_n - 7923f_{n-1} + 9982f_{n-2} - 7298fn-3 + 2877f_{n-4} - 475f_{n-5})$
6. $\mathbf{y}_{n+1}=y_n+(h/60480)(198721f_n - 447288f_{n-1} + 705549f_{n-2} - 688256f_{n-3} + 407139f_{n-4} - 134472f_{n-5} + 19087f_{n-6)}$
7. $\mathbf{y}_{n+1} = y_n+(H/120960)(120960f_n - 1152169f_{n-1} + 2183877f_{n-2} - 26644777f_{n-3} + 21022430f_{n-4} - 1041723f_{n-5} + 295767f_{n-6} - 36799f_{n-7})$

Exemplo de leitura

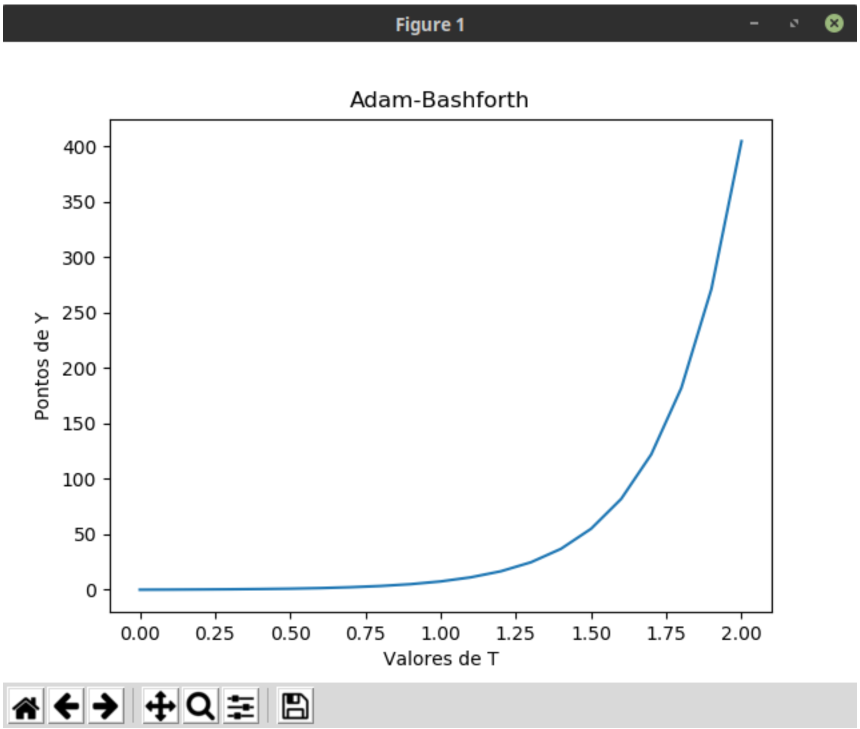
O exemplo demonstrado foi o utilizado para a demonstração dos pontos e do grafico, esse valor de leitura foi disponibilizado na especificação do projeto, [adam bashforth 0.0 0.1 0.23 0.402 0.6328 0.94592 0 0.1 20 1-t+4*y 6]

Pontos gerados

1. Método de Adam Bashforth
2. y(0) = 0.0
3. h = 0.1
4. 0 0.0
5. 1 0.1
6. 2 0.23
7. 3 0.402
8. 4 0.6328
9. 5 0.94592
10. 6 1.45035040000000
11. 7 2.23691931688889
12. 8 3.35844174442605
13. 9 4.99695083498148
14. 10 7.48198639667956

15. 11 11.1802888333590
16. 12 16.6249202050360
17. 13 24.7500581304216
18. 14 36.9138997900254
19. 15 55.0062241322630
20. 16 81.9320443028140
21. 17 122.138411354335
22. 18 182.125068886493
23. 19 271.512719182194
24. 20 404.819759848820

Grafico



Método 8: Adam-Bashforth

Método de Adam-Bashforth por outros métodos

Neste tópico será demonstrado os resultados obtidos pela implementação do método de Adam-Bashforth obtendo os valores iniciais de y e t, ou seja, utilizamos do método de Euler, Euler inverso, Euler aprimorado ou Runge-Kutta para calcular os N valores iniciais, para podermos então utilizar o método de Adam-Bashforth.

Exemplos de leitura

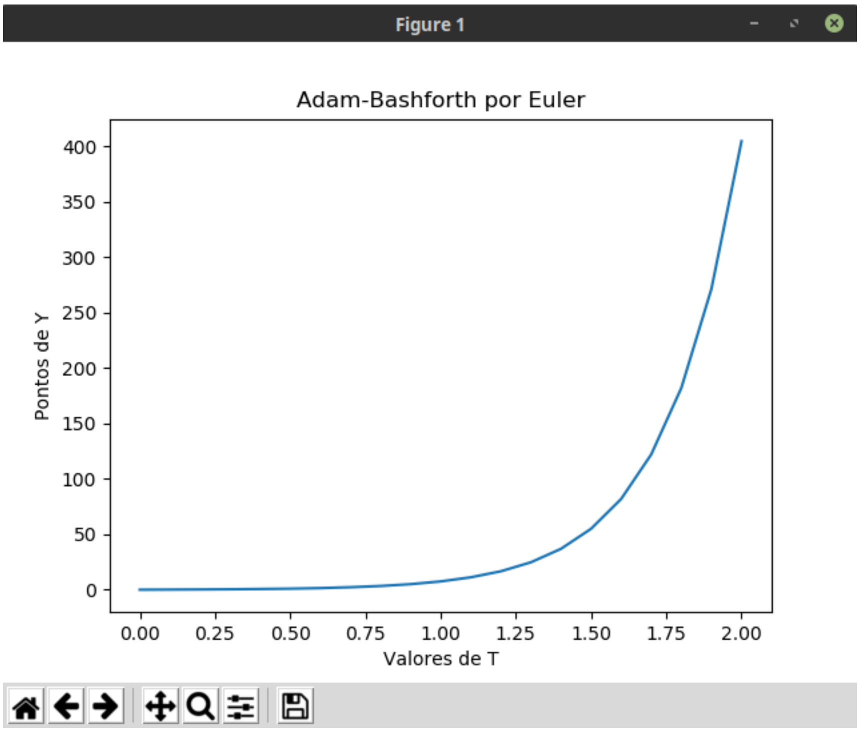
Os exemplos citados aqui foram disponibilizados na especificação do projeto, os que serão demonstrados seus resultados e gráficos aqui são:

1. adam bashforth by euler 0 0 0.1 20 1-t+4*y 6
2. adam bashforth by euler inverso 0 0 0.1 20 1-t+4*y 6
3. adam bashforth by euler inverso sem previsao 0 0 0.1 20 1-t+4*y 6
4. adam bashforth by euler aprimorado 0 0 0.1 20 1-t+4*y 6
5. adam bashforth by runge kutta 0 0 0.1 20 1-t+4*y 6

Pontos gerados

1. Método de Adam Bashforth By Euler
2. y(0) = 0
3. h = 0.1
4. 0 0.0
5. 1 0.100000000000000
6. 2 0.230000000000000
7. 3 0.402000000000000
8. 4 0.632800000000000
9. 5 0.945920000000000
10. 6 1.45035040000000
11. 7 2.23691931688889
12. 8 3.35844174442605
13. 9 4.99695083498148
14. 10 7.48198639667956
15. 11 11.1802888333590
16. 12 16.6249202050360
17. 13 24.7500581304216
18. 14 36.9138997900254
19. 15 55.0062241322630
20. 16 81.9320443028140
21. 17 122.138411354335
22. 18 182.125068886494
23. 19 271.512719182194
24. 20 404.819759848820

Grafico



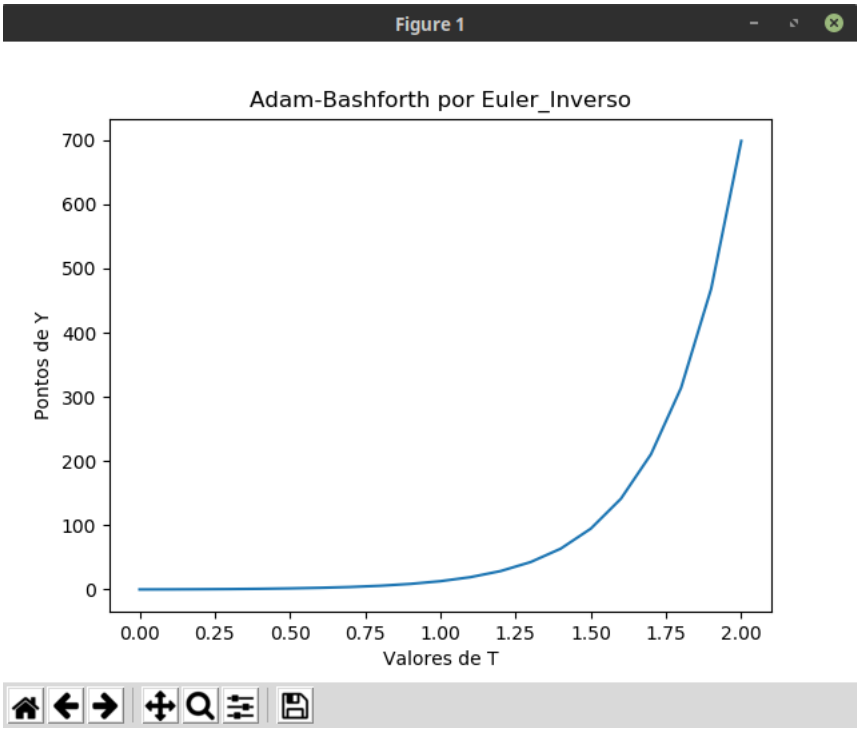
Método 9: Adam-Bashforth Por Euler

Pontos gerados

1. Método de Adam Bashforth By Euler Inverso
2. y(0) = 0
3. h = 0.1
4. 0 0.0
5. 1 0.130000000000000
6. 2 0.318800000000000
7. 3 0.599328000000000
8. 4 1.02295168000000
9. 5 1.66980462080000
10. 6 2.56669570372267
11. 7 3.83517932594029
12. 8 5.76086180833961
13. 9 8.64973784449650
14. 10 12.8827270903983
15. 11 19.1762162622181
16. 12 28.6224313633973

17. 13 42.6826707427550
18. 14 63.5715579421872
19. 15 94.7590071017958
20. 16 141.323288735407
21. 17 210.695867974306
22. 18 314.116250900087
23. 19 468.445356751208
24. 20 698.631384884924

Grafico



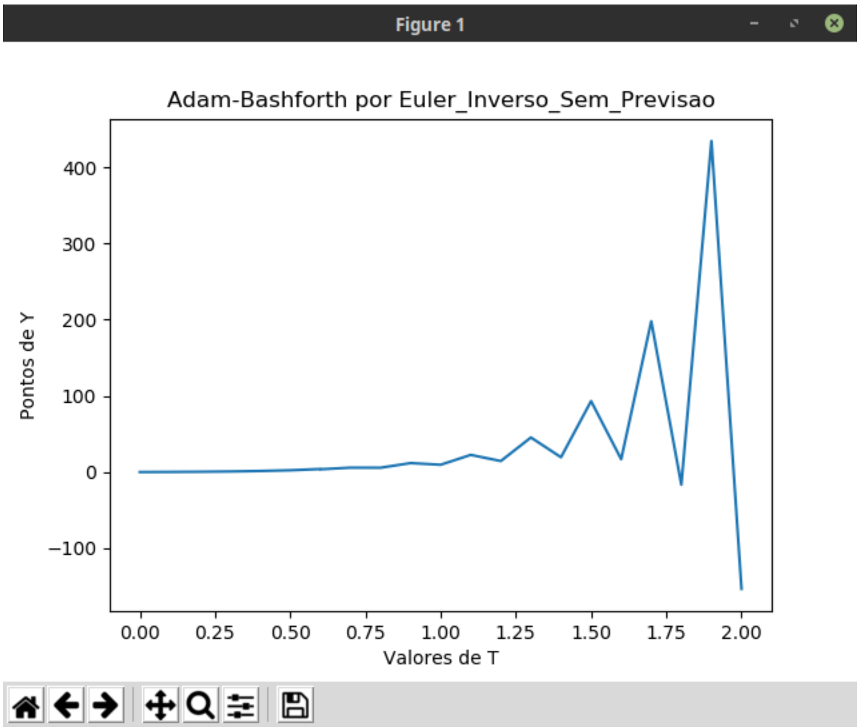
Método 10: Adam-Bashforth Por Euler inverso

Pontos gerados

1. Método de Adam Bashforth By Euler Inverso Sem Previsao
2. y(0) = 0
3. h = 0.1
4. 0 0.0
5. 1 0.150000000000000
6. 2 0.383333333333333
7. 3 0.755555555555556
8. 4 1.35925925925926

9. 5 2.34876543209877
10. 6 3.98127572016462
11. 7 3.63048696844994
12. 8 5.75002286236855
13. 9 5.66614027968298
14. 10 11.7613774037240
15. 11 9.58053616788071
16. 12 22.5273230297922
17. 13 14.4534301085984
18. 14 45.3583424664234
19. 15 19.4172159858294
20. 16 93.1539258931419
21. 17 16.8643618850167
22. 18 197.954621338766
23. 19 -16.6243154488087
24. 20 434.640071836189

Grafico

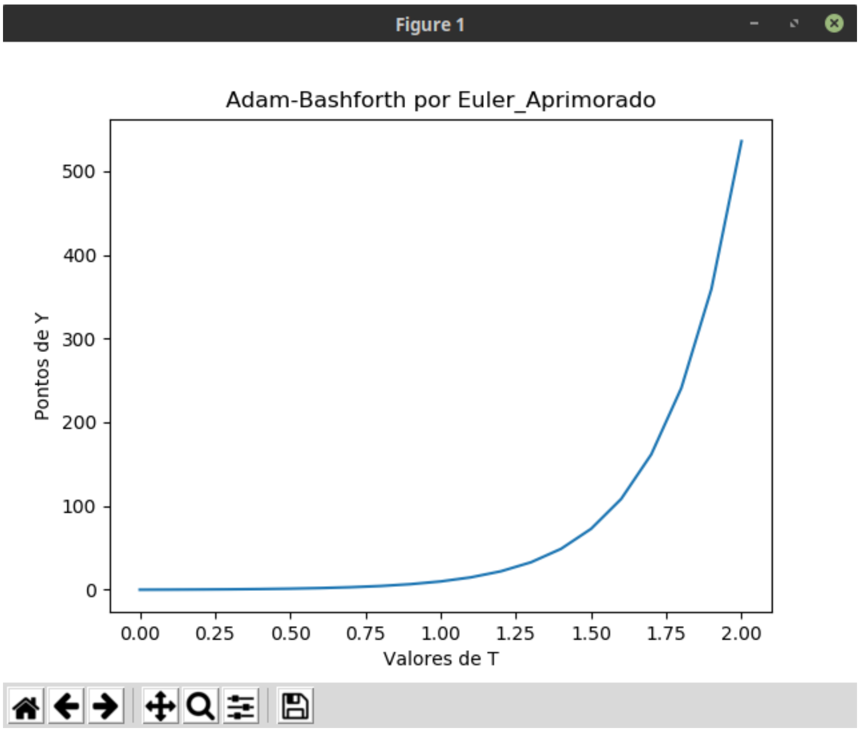


Método 11: Adam-Bashforth Por Euler inverso sem previsao
Verificar tópico da conclusão

Pontos gerados

1. Método de Adam Bashforth By Euler Aprimorado
2. y(0) = 0
3. h = 0.1
4. 0 0.0
5. 1 0.115000000000000
6. 2 0.273200000000000
7. 3 0.495336000000000
8. 4 0.812097280000000
9. 5 1.26890397440000
10. 6 1.94566533580800
11. 7 2.95034555054123
12. 8 4.43033073361868
13. 9 6.62178245644458
14. 10 9.88667894391046
15. 11 14.7449658269955
16. 12 21.9694362353666
17. 13 32.7351647816092
18. 14 48.7896601869607
19. 15 72.7172842441209
20. 16 108.386148255521
21. 17 161.584447620543
22. 18 240.924394284387
23. 19 359.238976622275
24. 20 535.696063138739

Grafico



Método 12: Adam-Bashforth Por Euler aprimorado

Pontos gerados

1. Método de Adam Bashforth By Runge Kutta (ordem=6)
2. y(0) = 0
3. h = 0.1
4. 0 0.0
5. 1 0.117200000000000
6. 2 0.279737813333333
7. 3 0.509907554076445
8. 4 0.840966095334301
9. 5 1.32252382328002
10. 6 2.02836134047985
11. 7 3.06872220834702
12. 8 4.60828020736213
13. 9 6.89236515086689
14. 10 10.2867228821220
15. 11 15.3372283794531
16. 12 22.8581235247808
17. 13 34.0634846111728
18. 14 50.7642408683034

19. 15 75.6618503726720

20. 16 112.785401547090

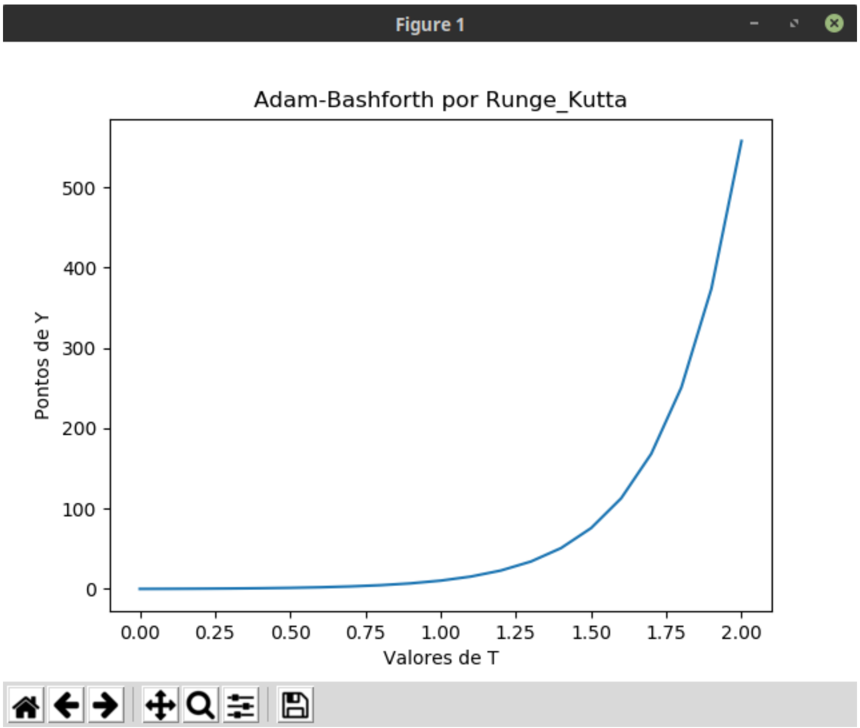
21. 17 168.144096399825

22. 18 250.701252221953

23. 19 373.826207774336

24. 20 557.459348935009

Grafico



Método 13: Adam-Bashforth Por Runge Kutta

0.3.6 Método de Adam-Moulton (2 a 8 Ordem)

Explicando o método

Este método é um variante do método de Adam-Bashforth, que tem por objetivo utilizar-se de polimonios para conseguir aproximar o seu valor ao valor do P.V.I (Problema do Valor Inicial). Neste método porém, utilizamos de valores para pontos calculados com o método da previsão para encontrarmos f_{n+1},podemos então descreve-lá de forma implicita, utilizando-se do método da previsão para podermos encontrar $\mathbf{y}_{n+1}$ para f_{n+1} e depois utilizamos os pontos anteriores (de acordo com a ordem do método que foi solicitado) para encontrarmos o valor final de $\mathbf{y}_{n+1}$ através do método de Adam-Moulton, a seguir, podemos identificar quais os coeficientes e suas ordem que foram utilizados.

1. $\mathbf{y}_{n+1} = y_n + (h/2)(f_n + f_{n+1})$
2. $\mathbf{y}_{n+1} = y_n + (h/12)(5f(_{n+1} + 8f_n - f_{n-1})$
3. $\mathbf{y}_{n+1} = y_n + (h/24)(9f(_{n+1} + 19f_n - 5f_{n-1} + f_{n-2})$
4. $\mathbf{y}_{n+1} = y_n + (h/720)(251f_{n+1} + 646f_n - 264f_{n-1} + 106f_{n-2} - 19f_{n-3})$
5. $\mathbf{y}_{n+1} = y_n + (h/1440)(475f_{n+1} + 1427f_n - 798f_{n-1} + 482f_{n-2} - 173f_{n-3} + 27f_{n-4})$
6. $\mathbf{y}_{n+1} = y_n + (h/60480)(19087f_{n+1} + 64872f_n - 46461f_{n-1} + 37504f_{n-2} - 20208f_{n-3} + 6312f_{n-4} - 863f_{n-5})$
7. $\mathbf{y}_{n+1} = y_n + (h/120960)(36799f_{n+1} + 139849f_n - 121797f_{n-1} + 123133f_{n-2} - 88547f_{n-3} + 41499f_{n-4} - 11351f_{n-5} + 1375f_{n-6})$

Exemplo de leitura

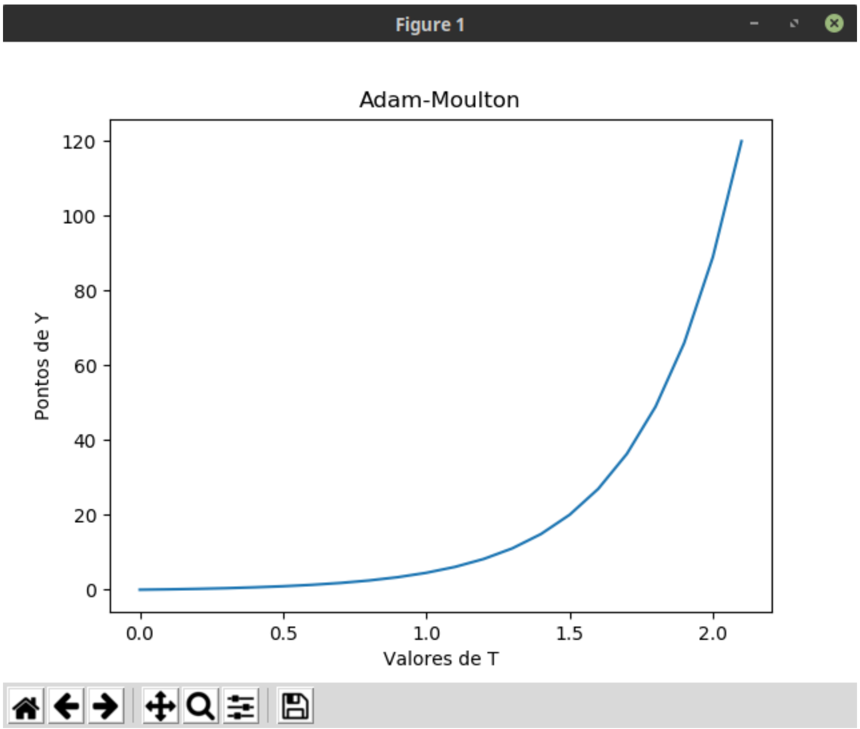
Os exemplos demonstrados aqui foram disponibilizados na especificação do projeto, no caso, a única diferença, é que a ultima leitura é realizada indicando a ordem da equação [adam multon 0.0 0.1 0.23 0.402 0.6328 0 0.1 20 1-t+4*y 6], no caso, de 6 ordem, foi dado 5 pontos para encontrarmos f_{n-1},f_{n-2}..... f_{n-5} e utilizarmos além de utilizar no método de Adam-Moulton, utilizarmos para encontrar o y_{n+1} para utilizarmos no sexto coeficiente (equação de 6 ordem portanto de Adam-Moulton no quesito 5 demonstrado anteriormente) e encontrarmos por fim $\mathbf{y}_{n+1}$.

Pontos gerados

1. Método de Adam Moulton
2. y(0.1) = 0.0
3. h = 0.1
4. 0 0.0
5. 1 0.1
6. 2 0.23
7. 3 0.402
8. 4 0.6328
9. 5 0.923972925925926
10. 6 1.30194049850637
11. 7 1.80152732866933
12. 8 2.46343218737344
13. 9 3.34417375697991
14. 10 4.52089671316776

15. 11 6.09706618299341
16. 12 8.21234086754611
17. 13 11.0554082163102
18. 14 14.8809319122879
19. 15 20.0326578976015
20. 16 26.9746270852402
21. 17 36.3332488085889
22. 18 48.9541086764570
23. 19 65.9786640506211
24. 20 88.9477690882309

Grafico



Método 14: Adam-Moulton
Verificar tópico da conclusão

Método de Adam-Moulton por outros métodos

Novamente nós nos deparamos com a utilziação dos métodos de passos simples (Euler, Euler inverso, Euler aprimorado e Runge-Kutta) para obtermos os valores iniciais de y e t para utilizarmos o método de Adam-Moulton.

Exemplo de leitura

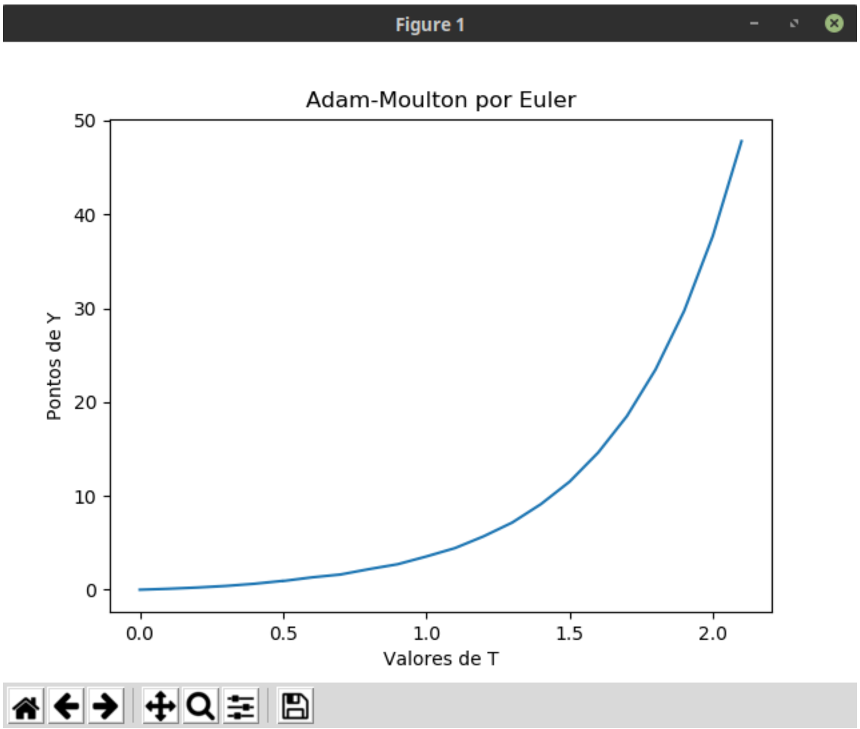
Os exemplos de leitura foram disponibilizados na especificação do projeto e como já foi mencionado anteriomente, alguns quesitos foram bônus, suas leituras foram definidas durante a realização do projeto e tambéms e encontram a seguir:

1. adam multon by euler 0 0 0.1 20 1-t+4*y 6
2. adam multon by euler inverso 0 0 0.1 20 1-t+4*y 6
3. adam multon by eulera primorado 0 0 0.1 20 1-t+4*y 6
4. adam multon by euler aprimorado sem previsao 0 0 0.1 20 1-t+4*y 6
5. adam multon by runge kutta 0 0 0.1 20 1-t+4*y 6

Pontos gerados

1. Método de Adam Multon By Euler
2. y(0) = 0
3. h = 0.1
4. 0 0.0
5. 1 0.100000000000000
6. 2 0.230000000000000
7. 3 0.402000000000000
8. 4 0.632800000000000
9. 5 0.945920000000000
10. 6 0.923972925925926
11. 7 1.31369835555556
12. 8 1.61815837396934
13. 9 2.19256825838932
14. 10 2.71581589392693
15. 11 3.54066980834415
16. 12 4.43450071562832
17. 13 5.69061783521868
18. 14 7.16073802463788
19. 15 9.12208108873918
20. 16 11.5137990452944
21. 17 14.6241379549302
22. 18 18.4951253289344
23. 19 23.4679196924167
24. 20 29.7208222790681

Grafico



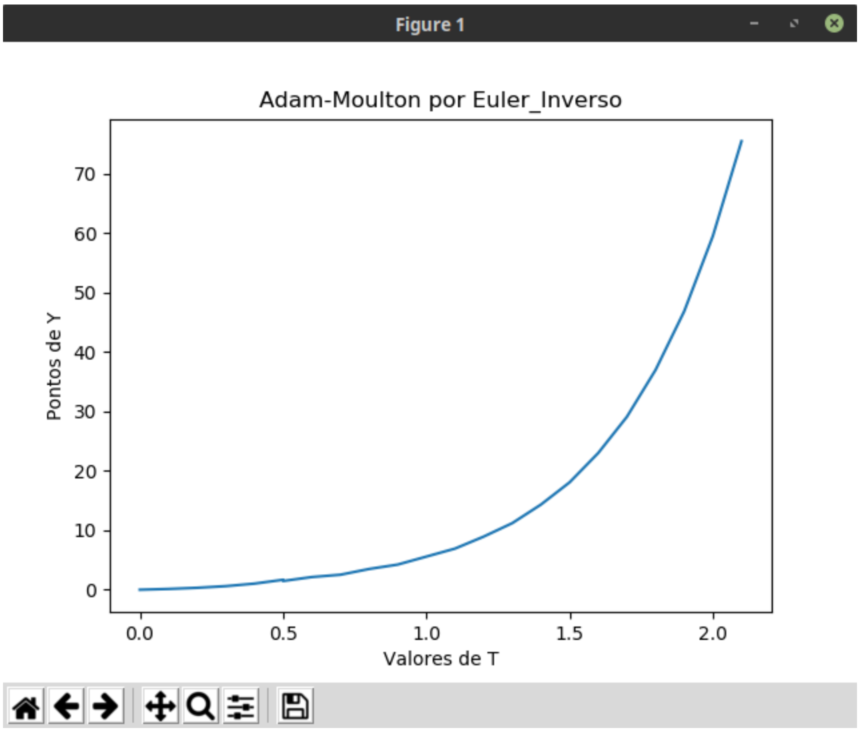
Método 15: Adam-Moulton Por Euler
Verificar tópico da conclusão

Pontos gerados

1. Método de Adam Multon By Inversor
2. y(0) = 0
3. h = 0.1
4. 0 0.0
5. 1 0.130000000000000
6. 2 0.318800000000000
7. 3 0.599328000000000
8. 4 1.02295168000000
9. 5 1.66980462080000
10. 6 1.46395159668148
11. 7 2.12814757781570
12. 8 2.51567888801189
13. 9 3.48258224105557
14. 10 4.22407673546692
15. 11 5.56618113673504
16. 12 6.91083980582428

17. 13 8.92531166909023
18. 14 11.1931839411277
19. 15 14.3146897679553
20. 16 18.0509512577189
21. 22.9849115520839
22. 18 29.0709622912434
23. 19 36.9489689344638
24. 20 46.8131340252578

Grafico



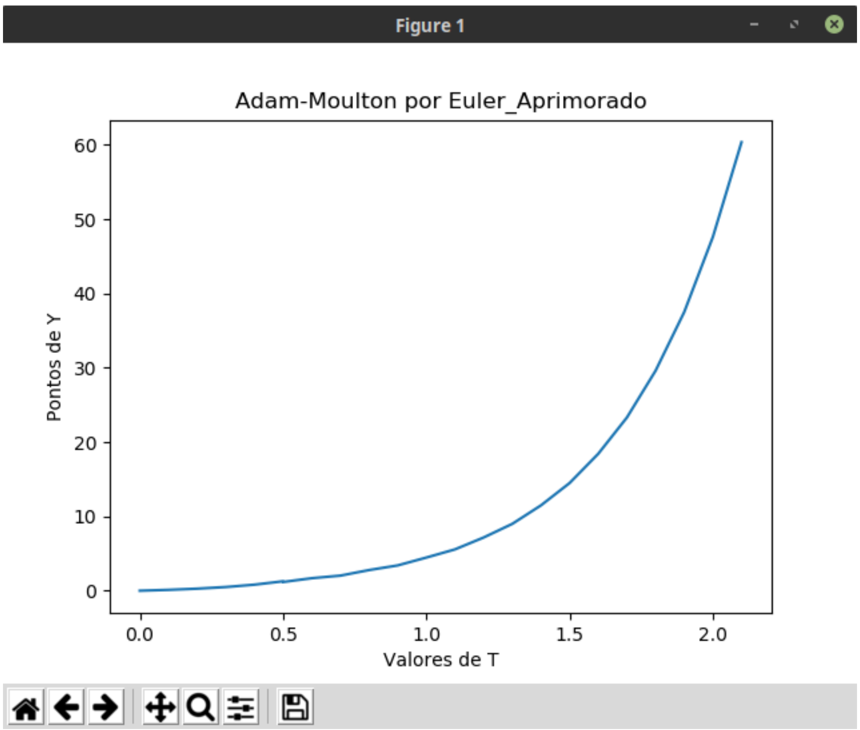
Método 16: Adam-Moulton Por Euler inverso
Verificar tópico da conclusão

Pontos gerados

1. Método de Adam Multon By Aprimorado
2. y(0) = 0
3. h = 0.1
4. 0 0.0
5. 1 0.115000000000000
6. 2 0.273200000000000
7. 3 0.495336000000000

8. 4 0.812097280000000
9. 5 1.26890397440000
10. 6 1.17160420808889
11. 7 1.68283418388267
12. 8 2.02899367808882
13. 9 2.77815414207341
14. 10 3.40421631137557
15. 11 4.46172965536276
16. 12 5.56347303355323
17. 13 7.16260583568798
18. 14 8.99809013884822
19. 15 11.4859476424624
20. 16 14.4915958298441
21. 17 18.4309759139434
22. 18 23.3119845532059
23. 19 29.6066870587619
24. 20 37.5051705843006

Grafico

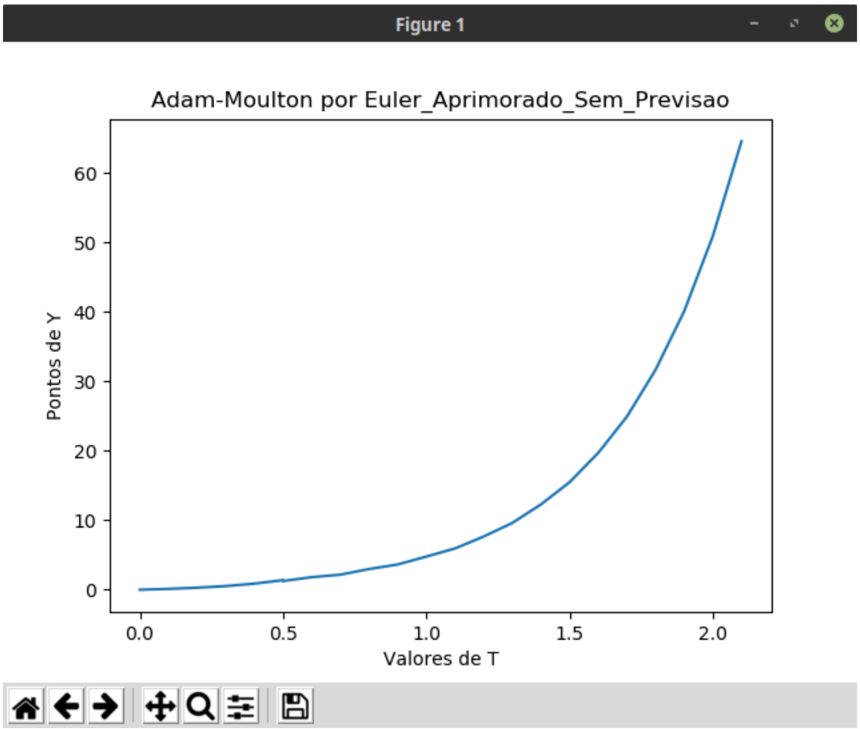


Método 17: Adam-Moulton Por Euler aprimorado

Pontos gerados

1. Método de Adam Multon By Aprimorado Sem Previsao
2. y(0) = 0
3. h = 0.1
4. 0 0.0
5. 1 0.120000000000000
6. 2 0.287500000000000
7. 3 0.526250000000000
8. 4 0.871875000000000
9. 5 1.37781250000000
10. 6 1.25421792293596
11. 7 1.80656216845100
12. 8 2.16615820551758
13. 9 2.97430421273648
14. 10 3.63431197404461
15. 11 4.77004144161187
16. 12 5.94099679453056
17. 13 7.65519053957930
18. 14 9.61263196195393
19. 15 12.2768705869916
20. 16 15.4876898242362
21. 17 19.7046076672529
22. 18 24.9233414983601
23. 19 31.6604253011767
24. 20 40.1092905489826

Grafico



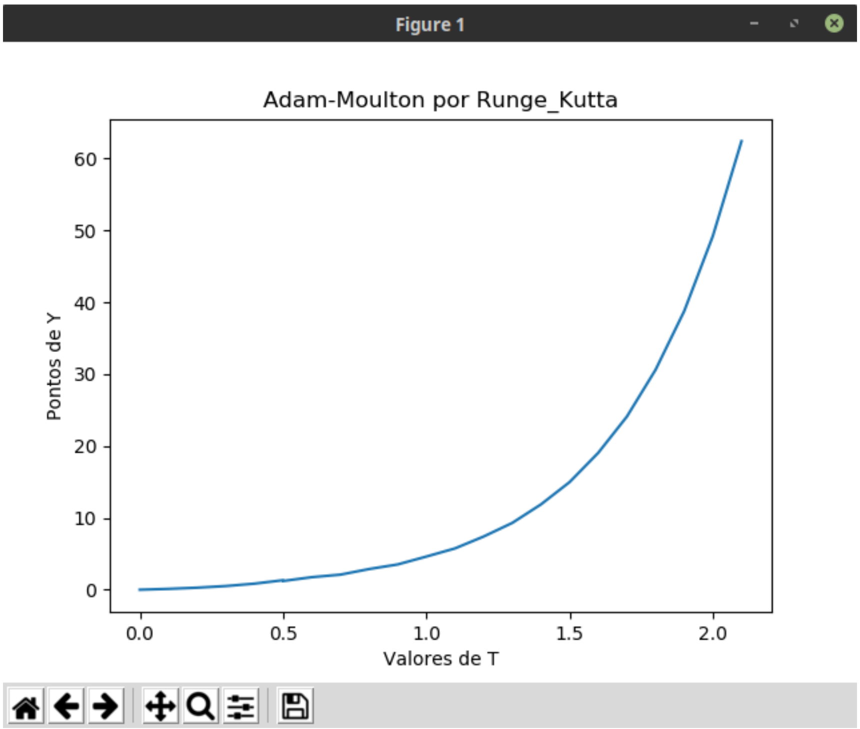
Método 18: Adam-Moulton Por Euler aprimorado sem previsao
Verificar tópico da conclusão

Pontos gerados

1. Método de Adam Multon By Runge Kutta (ordem = 6)
2. y(0) = 0
3. h = 0.1
4. 0 0.0
5. 1 0.117200000000000
6. 2 0.279737813333333
7. 3 0.509907554076445
8. 4 0.840966095334301
9. 5 1.32252382328002
10. 6 1.21156394206300
11. 7 1.74312802921568
12. 8 2.09541593289953
13. 9 2.87365014136489
14. 10 3.51584815041701
15. 11 4.61166350709940
16. 12 5.74676270400706

17. 13 7.40204089189516
18. 14 9.29656292296261
19. 15 11.8703044861881
20. 16 14.9754658129424
21. 17 19.0498364827652
22. 18 24.0947952189918
23. 19 30.6045450273122
24. 20 38.7703255213266

Grafico



Método 19: Adam-Moulton Por Runge-Kutta
Verificar tópico da conclusão

0.3.7 Formula inversa (2 a 6 Ordem)

Explicando o método

Este método também visa por meio de polinomios aproximar seu valor ao valor de P.V.I (Problema do Valor Inicial), porém, seu método tem por objetivo obter uma formula implicita para $\mathbf{y}_{n+1}$. Na especificação do projeto foi solicitado até a 6 ordem, que podemos verificar a seguir:

1. $\mathbf{y}_{n+1}$ = (1/3)(4y_n - y_{n-1} + 2f_{n+1}
2. $\mathbf{y}_{n+1}$ = (1/11)(18y_n - 9y_{n-1} + 2y_{n-2} + 6hf_{n+1})
3. $\mathbf{y}_{n+1}$ = (1/25)(48y_n - 36y_{n-1} + 16y_{n-2} - 3y_{n-3} + 12hf_{n+1})
4. $\mathbf{y}_{n+1}$ =(1/137)(300y_n - 300y_{n-1} + 200y_{n-2} - 75y_{n-3} - 12y_{n-4} + 60hf_{n+1})
5. $\mathbf{y}_{n+1}$ = (1/147)(360y_n - 450y_{n-1} + 400y_{n-2} - 225y_{n-3}+72y_{n-4}-10Y(n-5) + 60hf_{n+1})

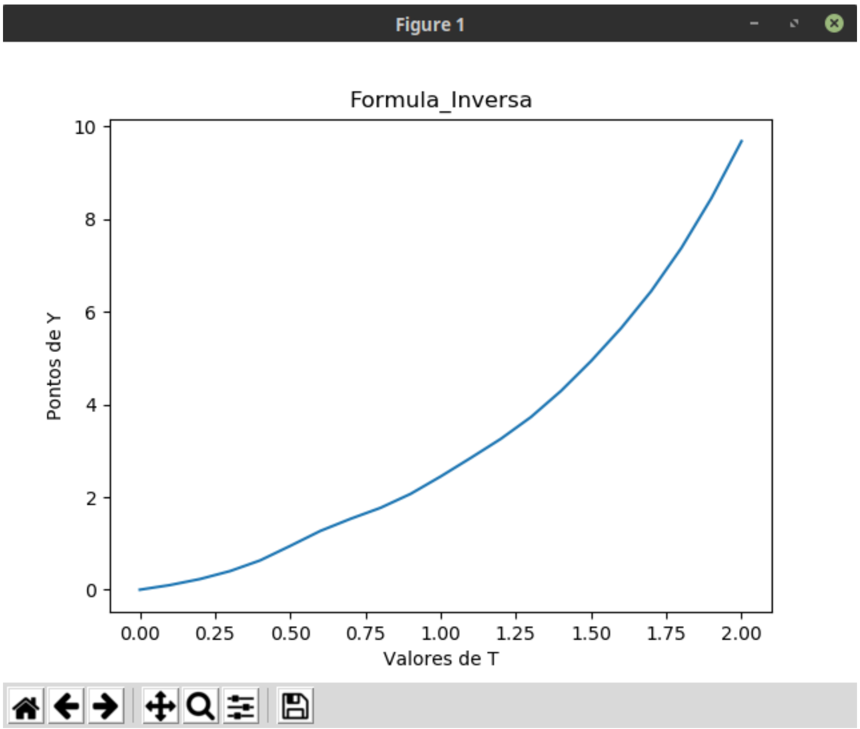
Exemplo de leitura

Os exemplos de leitura foram retirados da especificação do projeto. Seu ultimo valor de leitura indica qual a ordem que se está solicitando para execução, [formula inversa 0.0 0.1 0.23 0.402 0.6328 0.94592 0 0.1 20 1-t+4*y 6] foi solicitado de 6 ordem.

Pontos gerados

1. Método de Formula Inversa
2. y(0.1) = 0.0
3. h = 0.1
4. 0 0.0
5. 1 0.1
6. 2 0.23
7. 3 0.402
8. 4 0.6328
9. 5 0.94592
10. 6 1.26889475918367
11. 7 1.52974055877310
12. 8 1.76784327263991
13. 9 2.06909318423963
14. 10 2.44529614034927
15. 11 2.84581437673247
16. 12 3.25755086967631
17. 13 3.72755404078940
18. 14 4.29202586631765
19. 15 4.93804424903362
20. 16 5.64886258154591
21. 17 6.44814553427314
22. 18 7.37767089961126
23. 19 8.45605559073216
24. 20 9.68462444625757

Grafico



Método 20: Formula inversa
Verificar tópico da conclusão

Método de Formula Inversa por outros métodos

Como já vimos com os métodos de Adam-Bashforth e com o de Adam-Moulton, utilizamos os métodos de passos simples (Euler, Euler inverso, Euler aprimorado e Runge-Kutta) para termos os valores inicias de y e t.

Exemplo de leitura

Como já foi especificado anteriormente os exemplos aqui demonstrados foramos disponibilizados na especificação, estes exemplos são os declarados a seguir:

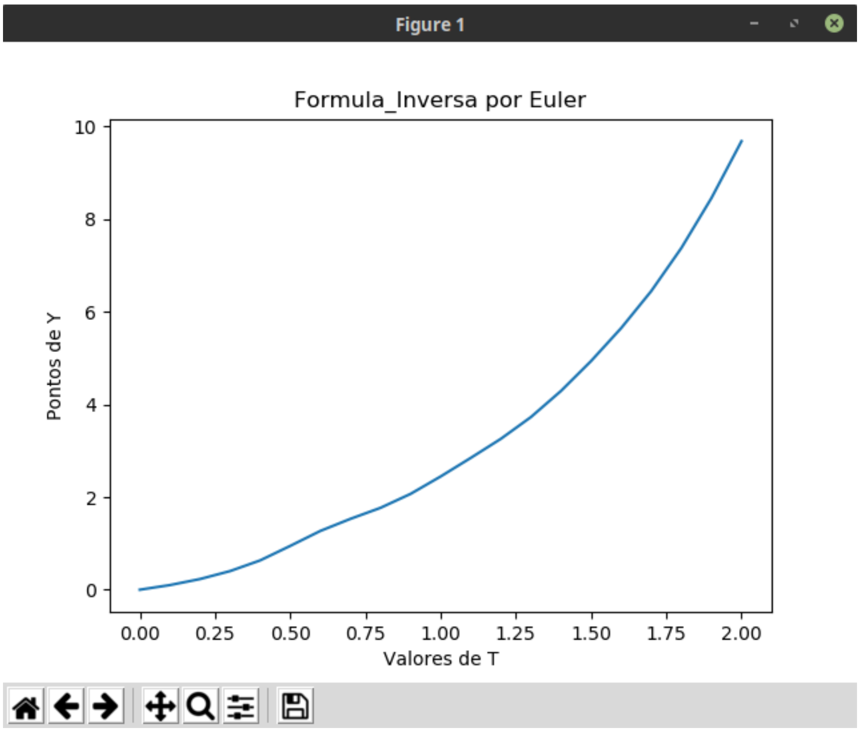
1. formula inversa by euler 0 0 0.1 20 1-t+4*y 6
2. formula inversa by euler inverso 0 0 0.1 20 1-t+4*y 6
3. formula inversa by euler aprimorado 0 0 0.1 20 1-t+4*y 6
4. formula inversa by runge kutta 0 0 0.1 20 1-t+4*y 6

Pontos gerados

1. Método de Formula Inversa By Euler (0) = 0 = 0.1
2. 0 0.0
3. 1 0.100000000000000
4. 2 0.230000000000000
5. 3 0.402000000000000

6. 4 0.632800000000000
7. 5 0.945920000000000
8. 6 1.26889475918367
9. 7 1.52974055877310
10. 8 1.76784327263991
11. 9 2.06909318423963
12. 10 2.44529614034927
13. 11 2.84581437673247
14. 12 3.25755086967631
15. 13 3.72755404078940
16. 14 4.29202586631764
17. 15 4.93804424903362
18. 16 5.64886258154591
19. 17 6.44814553427313
20. 18 7.37767089961126
21. 19 8.45605559073215
22. 20 9.68462444625756

Grafico

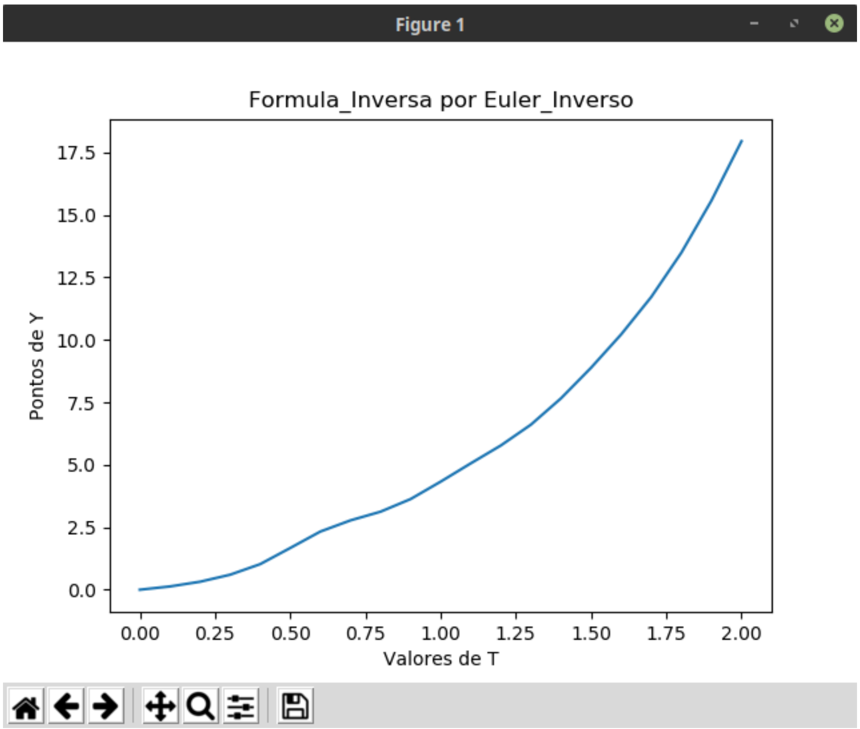


Método 21: Formula inversa Por Euler
Verificar tópico da conclusão

Pontos gerados

1. Método de Formula Inversa By Euler Inverso (0) = 0
2. h = 0.1
3. 0 0.0
4. 1 0.130000000000000
5. 2 0.318800000000000
6. 3 0.599328000000000
7. 4 1.02295168000000
8. 5 1.66980462080000
9. 6 2.32712920733431
10. 7 2.77382772004460
11. 8 3.12176042197046
12. 9 3.62733230941861
13. 10 4.32503894430359
14. 11 5.05647389680171
15. 12 5.77243657174414
16. 13 6.60607717371875
17. 14 7.65965445464305
18. 15 8.88515265115031
19. 16 10.2187247648224
20. 17 11.7162138764366
21. 18 13.4878426391958
22. 19 15.5721609209695
23. 20 17.9511812803398

Grafico



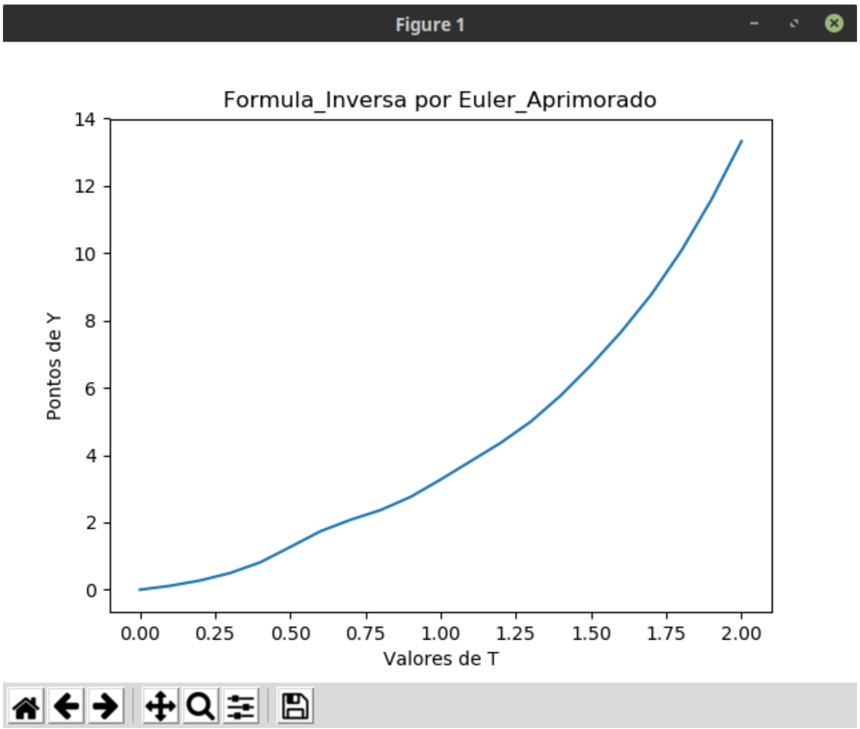
Método 22: Formula inversa Por Euler inverso
Verificar tópico da conclusão

Pontos gerados

1. Método de Formula Inversa By Euler Aprimorado
2. y(0) = 0
3. h = 0.1
4. 0 0.0
5. 1 0.115000000000000
6. 2 0.273200000000000
7. 3 0.495336000000000
8. 4 0.812097280000000
9. 5 1.26890397440000
10. 6 1.73434236365845
11. 7 2.07586379078060
12. 8 2.36446346032338
13. 9 2.75671225560794
14. 10 3.27295757573324
15. 11 3.81791294063466
16. 12 4.36445519536140

17. 13 4.99584743108431
18. 14 5.77538574014353
19. 15 6.67545129871931
20. 16 7.66036729236073
21. 17 8.76791095502858
22. 18 10.0685473770586
23. 19 11.5892806606545
24. 20 13.3239641113316

Grafico



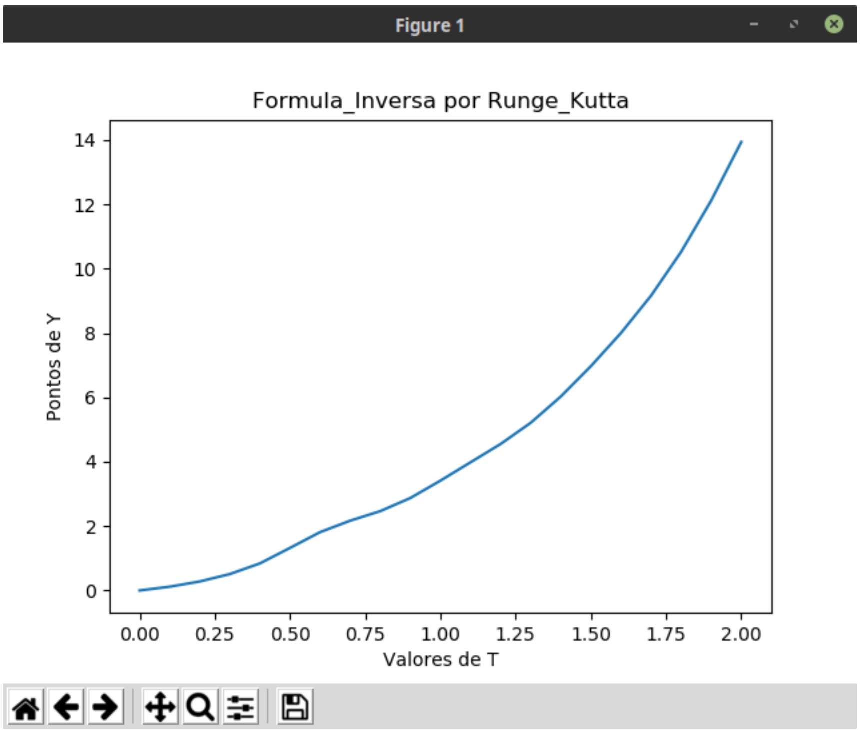
Método 23: Formula inversa Por Euler aprimorado
Verificar tópico da conclusão

Pontos gerados

1. Método de Formula Inversa By Runge Kutta (ordem=6)
2. y(0) = 0
3. h = 0.1
4. 0 0.0
5. 1 0.117200000000000
6. 2 0.279737813333333
7. 3 0.509907554076445

8. 4 0.840966095334301
9. 5 1.32252382328002
10. 6 1.81275066342684
11. 7 2.16803919121053
12. 8 2.46476807318837
13. 9 2.87215664304092
14. 10 3.41222902685146
15. 11 3.98170219317539
16. 12 4.55077844001977
17. 13 5.20910984313364
18. 14 6.02488921984639
19. 15 6.96789068301866
20. 16 7.99894357868839
21. 17 9.15821348684247
22. 18 10.5212411258758
23. 19 12.1165053950755
24. 20 13.9364245057843

Grafico



Método 24: Formula inversa Por Runge-Kutta
Verificar tópico da conclusão

0.4 Conclusão

Pordemos concluir que além de ter tido alguns quesitos não elaborados no projeto, os quais já foram especificados no tópico *Implementações*, houveram também alguns casos que ocorreram erro durante o desenvolvimento do projeto, casos esses que não foram corrigidos a tempo da entrega do mesmo. Durante a implementação dos métodos foi possível fixar melhor o assunto e entender melhor como estes métodos funcionam.

Os métodos que não obtiveram um resultado satisfatório em relação aos casos disponibilizados na especificação do projeto estão indicados ao final de seu gráfico.

Houveram diversar tentativas com o objetivo de reparar os erros obtidos, como foi visto, não houve exito nessas tentativas, mesmo assim foi considerado de grande importancia demonstrar todos os métodos que foram implementados durante o desenvolvimento do projeto, dando ênfase nos métodos que obtiveram os resultados satisfatórios são eles:

1. Método de Runge-Kutta 5 Ordem *Sem parâmetros para identicar o porque esta incorreto, mas fica a atenção*par
2. Método de Adam-Bashforth por Euler inverso sem previsão *Sem parâmetros para identicar o porque esta incorreto, mas fica a atenção*
3. Método de Adam-Moulton *(valores ficaram muito pequenos em relação aos disponibilizados na especificação)*
4. Método de Formula Inversa *(novamento os valores ficaram muito pequenos em relação aos disponibiliados na especificação)*