



Relatório do Lab 8 de CCI-22 (Exame)

Trabalho 8 – Equações Diferenciais Ordinárias

Alunos:

Bruno Costa Alves Freire

Eduardo Soares e Silva Britto

Turma:

T 21.4

Professor:

Luiz Gustavo Bizarro Mirisola

Data:

28/06/2018

Instituto Tecnológico de Aeronáutica – ITA
Departamento de Computação

1. Descrição do Problema

O problema escolhido para ser tratado neste trabalho foi uma classe de equações diferenciais que generalizam a chamada Equação de Euler, estudada na disciplina de MAT-32:

$$x^2 y'' + \alpha x y' + \beta y = 0, \alpha, \beta \in \mathbb{R}$$

Enquanto a equação de Euler é uma equação de segunda ordem, a generalização da ideia seria uma equação do tipo:

$$\sum_{k=0}^n \alpha_k x^k y^{(k)} = 0$$

Com α_1 não nulo, de modo que sempre é possível dividir toda a equação por esse coeficiente, ficando com apenas n coeficientes livres.

Trata-se de um problema puramente matemático, para o qual existem soluções analíticas conhecidas e de fácil obtenção. No entanto, a depender do problema de valor inicial (PVI), o método de solução pode variar.

1.1. Equação Diferencial a ser resolvida

O exemplo escolhido para ser desenvolvido aqui é da equação de quarta ordem:

$$x^4 y^{(4)} + 6x^3 y''' + 7x^2 y'' + xy' - y = 0$$

Com os seguintes valores iniciais:

$$\begin{cases} y(1) = 2 \\ y'(1) = 0 \\ y''(1) = -2 \\ y'''(1) = 10 \end{cases}$$

1.2. Transformação em sistema de primeira ordem

O método padrão para reduzir uma equação diferencial de ordem n a um sistema de equações de primeira ordem, é nomear as derivadas de y como as componentes de um vetor de funções. Ou seja:

$$\begin{cases} y(x) = u_1(x) \\ y'(x) = u_2(x) \\ y''(x) = u_3(x) \\ y'''(x) = u_4(x) \end{cases}$$

Dessa forma, podemos identificar essas componentes na equação original, e isolar a derivada da última componente:

$$x^4 u_4' + 6x^3 u_4 + 7x^2 u_3 + x u_2 - u_1 = 0 \Rightarrow$$

$$u_4' = -6x^{-1} u_4 - 7x^{-2} u_3 - x^{-3} u_2 + x^{-4} u_1$$

E com isso somos capazes de escrever o sistema:

$$\begin{cases} u_1' = u_2 \\ u_2' = u_3 \\ u_3' = u_4 \\ u_4' = x^{-4} u_1 - x^{-3} u_2 - 7x^{-2} u_3 - 6x^{-1} u_4 \end{cases}$$

Com $u_1(1) = 2, u_2(1) = 0, u_3(1) = -2, u_4(1) = 10$.

1.3. Solução Analítica

A solução analítica de equações desse tipo é feita supondo uma solução do tipo $y = x^r$:

$$r(r-1)(r-2)(r-3)x^r + \alpha r(r-1)(r-2)x^r + \beta r(r-1)x^r + \gamma r x^r + \delta x^r = 0$$

Dividindo tudo por x^r e rearranjando os termos:

$$r^4 + (\alpha - 6)r^3 + (11 - 3\alpha + \beta)r^2 + (2\alpha - \beta + \gamma - 6)r + \delta = 0$$

No caso geral, precisamos resolver essa equação polinomial em r para encontrar as possíveis soluções, que terá o grau igual à ordem da equação diferencial. Como equações polinomiais de graus altos são difíceis de se resolver analiticamente, optamos por um caso em que a equação fosse simples o suficiente para ser resolvida manualmente, e nesse caso, a ordem da equação é 4, e os coeficientes são $\alpha = 6, \beta = 7, \gamma = 1, \delta = -1$. Assim ficamos com:

$$r^4 - 1 = 0$$

Cujas raízes são trivialmente iguais a 1, i , -1 e $-i$.

Por termos obtido raízes distintas, temos que as soluções da equação diferencial, $y_1(x) = x^1, y_2(x) = x^{-1}, y_3(x) = x^i, y_4(x) = x^{-i}$, são todas linearmente independentes, formando assim um conjunto fundamental de soluções. Isso significa que a solução geral da equação é:

$$y(x) = ax + bx^{-1} + cx^i + dx^{-i}$$

Como estamos interessados em soluções reais, devemos combinar os termos com potências complexas de modo a tomar suas partes real e imaginária como soluções linearmente independentes. Usando a identidade de Euler:

$$x^i = e^{i \ln x} = \cos \ln x + i \sin \ln x$$

Vamos tomar a parte real e a parte imaginária como soluções em vez de x^i e x^{-i} :

$$y(x) = ax + bx^{-1} + c \cos \ln x + d \sin \ln x$$

Agora, calculamos as derivadas de y para aplicar as condições iniciais:

$$\begin{cases} y(x) = ax + bx^{-1} + c \cos \ln x + d \sin \ln x \\ y'(x) = a - bx^{-2} - cx^{-1} \sin \ln x + dx^{-1} \cos \ln x \\ y''(x) = 2bx^{-3} + cx^{-2}(\sin \ln x - \cos \ln x) - dx^{-2}(\sin \ln x + \cos \ln x) \\ y'''(x) = -6bx^{-4} + cx^{-3}(3 \cos \ln x - \sin \ln x) + dx^{-3}(\cos \ln x + 3 \sin \ln x) \end{cases}$$

Aplicando $x = 1$ acima, temos:

$$\begin{cases} y(1) = a + b + c = 2 \\ y'(1) = a - b + d = 0 \\ y''(1) = 2b - c - d = -2 \\ y'''(1) = 3c + d - 6b = 10 \end{cases}$$

Resolvendo o sistema linear em a, b, c e d , chegamos em:

$$a = 1, b = -1, c = 2, d = -2$$

E com isso, nossa solução analítica para a equação com o PVI dado será

$$y(x) = x - x^{-1} + 2 \cos \ln x - 2 \sin \ln x$$

1.4. Casos e comportamentos da solução

A classe de equações referida na descrição problema tem seu método de resolução atrelado às características do PVI.

Caso sejam dadas informações sobre a função e suas derivadas num ponto $x > 0$, procedemos como feito acima, supondo soluções do tipo x^r . Contudo, se o ponto inicial é $x < 0$, devemos fazer uma mudança de variável do tipo $x = -z$, para que possamos aplicar o mesmo método à nova equação. Isso é necessário pois algumas soluções da equação só estão definidas para $x > 0$.

Para o ponto $x = 0$, temos algumas restrições impostas pela própria equação. Ao substituirmos $x = 0$ na equação, notamos que necessariamente, $y(0) = 0$. Além disso, não somos capazes de obter informação sobre as derivadas de y nesse ponto, de forma que um PVI com ponto inicial em $x = 0$ é mal formulado.

Agora, supondo não haver problemas com a formulação do PVI, ou seja, supondo que o ponto inicial seja em $x > 0$, quais são as características das soluções? Vimos que pelo procedimento de supor $y = x^r$, chegamos em equações polinomiais em r do tipo:

$$\sum_{k=0}^n \alpha_k \binom{r}{k} k! = 0$$

A natureza das soluções vai ser totalmente determinada pelas raízes dessa equação e suas multiplicidades.

Para as raízes reais, teremos soluções do tipo $y = x^r$, as quais serão linearmente dependentes caso não haja raízes repetidas. Para as raízes complexas, sabemos que estas vêm aos pares de conjugados, portanto, basta tomar as partes real e imaginária de x^r quando r for complexo.

Caso alguma raiz seja repetida, deve-se usar o método de redução de ordem, que consiste em supor uma solução do tipo $f(x) \cdot x^r$ para encontrar uma nova solução associada a essa raiz, que seja linearmente independente das demais.

O que se observa é que, quando as raízes possuem parte real positiva, temos uma solução que cresce com a potência dessa parte real, se tornando um termo mais relevante na solução. Para partes reais negativas, as soluções tendem a zero quando x cresce e explodem para x próximo de zero. Quanto às partes imaginárias, estas irão se manifestar na forma de uma oscilação senoidal em escala logarítmica, de modo seu comportamento oscilatório é evidente quando se toma o eixo x em escala logarítmica, mas pouco pronunciado numa escala normal. As soluções provenientes de raízes repetidas terão um termo logaritmo acoplado que fará com que explodam perto de $x = 0$, e cresçam de modo que é pouco perceptível a diferença entre estas e as demais soluções.

1.5. Medidas Realizadas e Erro

Por se tratar de um problema puramente matemático, não houve nenhuma medição a ser realizada. Uma vez que existe uma solução analítica, e nenhum dado experimental, não há nenhum erro a ser calculado.

Quanto ao parâmetro de erro aceitável para as soluções numéricas, decidimos que um erro de 10^{-4} é aceitável.

2. Soluções Numéricas

2.1. Via *software* de prateleira (MATLAB)

Para implementar a solução através de comandos MATLAB, basta transformar a equação num sistema de equações de primeira ordem, como feito acima, definir os valores iniciais, e fornecer à função `ode45` do MATLAB um intervalo de integração. O intervalo escolhido foi de $x = 1$ até $x = 15$.

Foram gerados os gráficos das figuras 1 e 2 comparando a solução do MATLAB com a solução analítica conhecida.

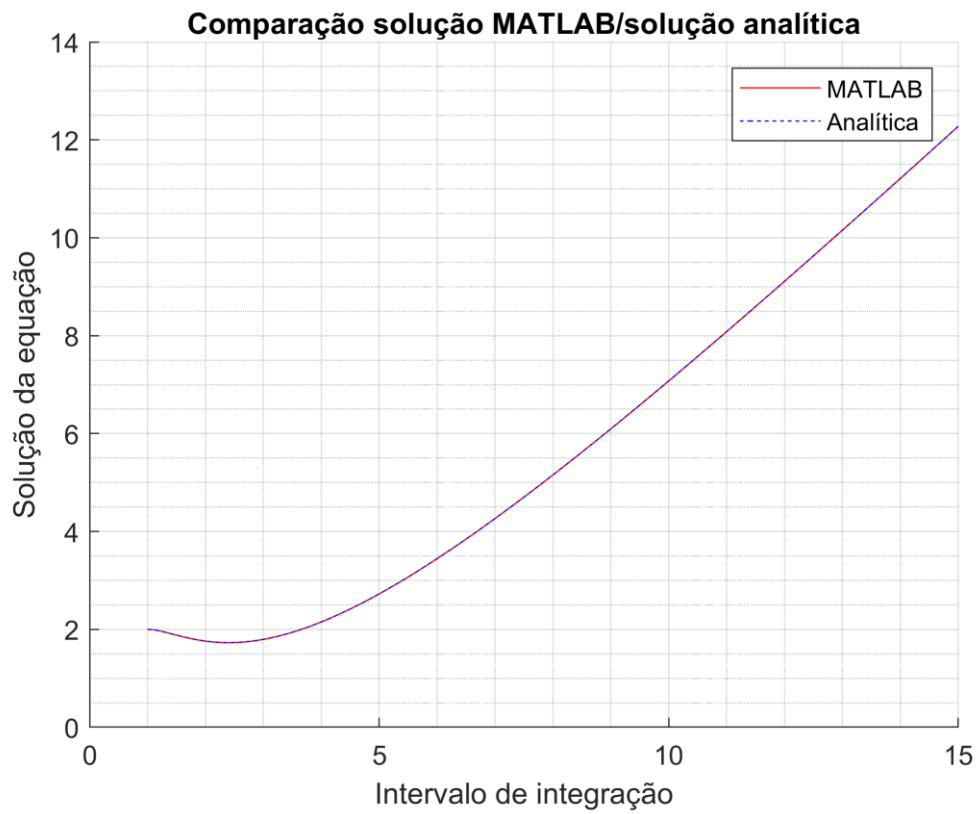


Figura 1

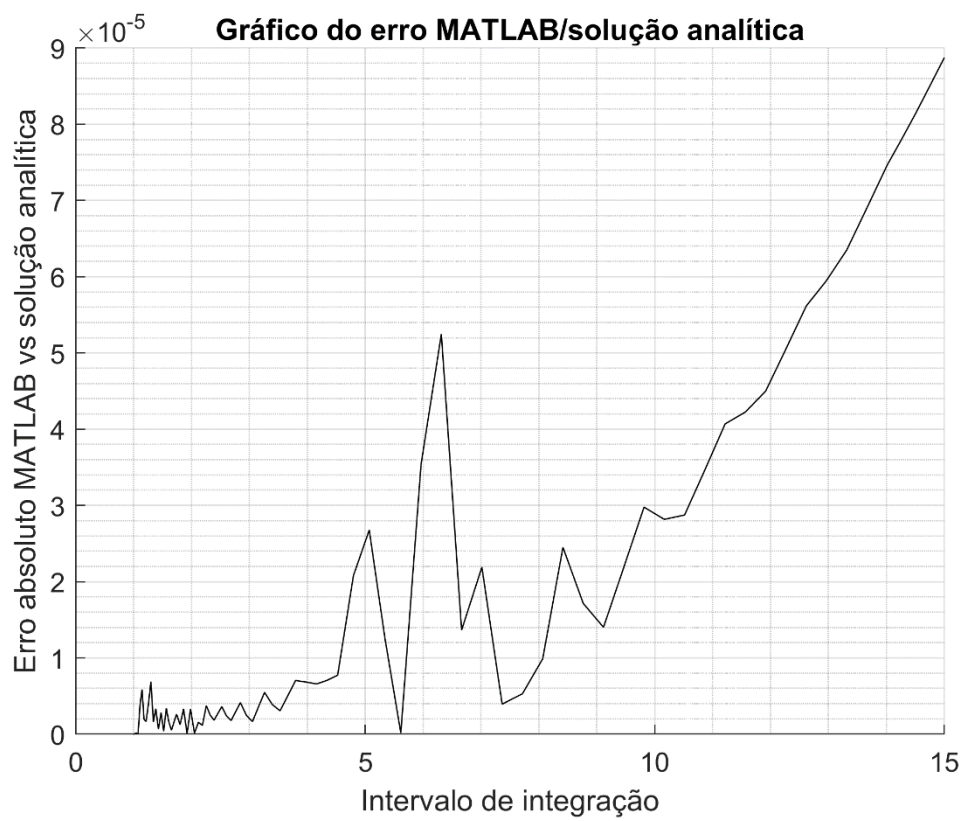


Figura 2

Podemos observar que, com um erro mantido sempre menor que 10^{-4} , é imperceptível na escala normal qualquer discrepância entre as duas curvas, razão pela qual foram plotadas uma com linha sólida e outra pontilhada, para que fosse possível visualizar ambas.

2.2. Via algoritmos implementados para o Lab

Foram implementados os métodos de Heun, de Runge-Kutta de 4ª ordem, e de Previsão-Correção de 4ª ordem, com passos de Adams-Bashforth e Adams-Moulton como códigos predictor e corretor respectivamente. Os scripts seguem em anexo, e estão nomeados conforme os cabeçalhos fornecidos.

Uma vez implementados, esses métodos foram comparados com a solução analítica e com a solução do MATLAB, gerando uma série de gráficos e tabelas, que se encontram na seção de resultados.

Para que todos satisfizessem o critério de erro aceitável, seus passos tiveram que ser escolhidos da seguinte forma:

	<i>Heun</i>	<i>Runge-Kutta</i>	<i>Previsor-Corretor</i>
<i>Passo</i>	0,0028	0,0565	0,0220
<i>Tempo de execução (s)</i>	0,1228	0,0120	0,0763

Tabela 1: Passos e tempos de execução dos métodos numéricos

O script completo do lab que produz todos os gráficos e estatísticas e resolve a equação por todos os métodos segue em anexo e está nomeado como `scriptcompleto.m`.

3. Resultados

Todos os métodos foram executados utilizando-se os passos conforme constam na tabela 1, de modo que estes satisfizessem o critério de erro aceitável. Em seguida, os resultados por estes obtidos foram comparados com a solução analítica (gabarito) e com a solução do MATLAB (quase gabarito). Os gráficos das comparações constam nas figuras 3 a 14.

3.1. Método de Heun

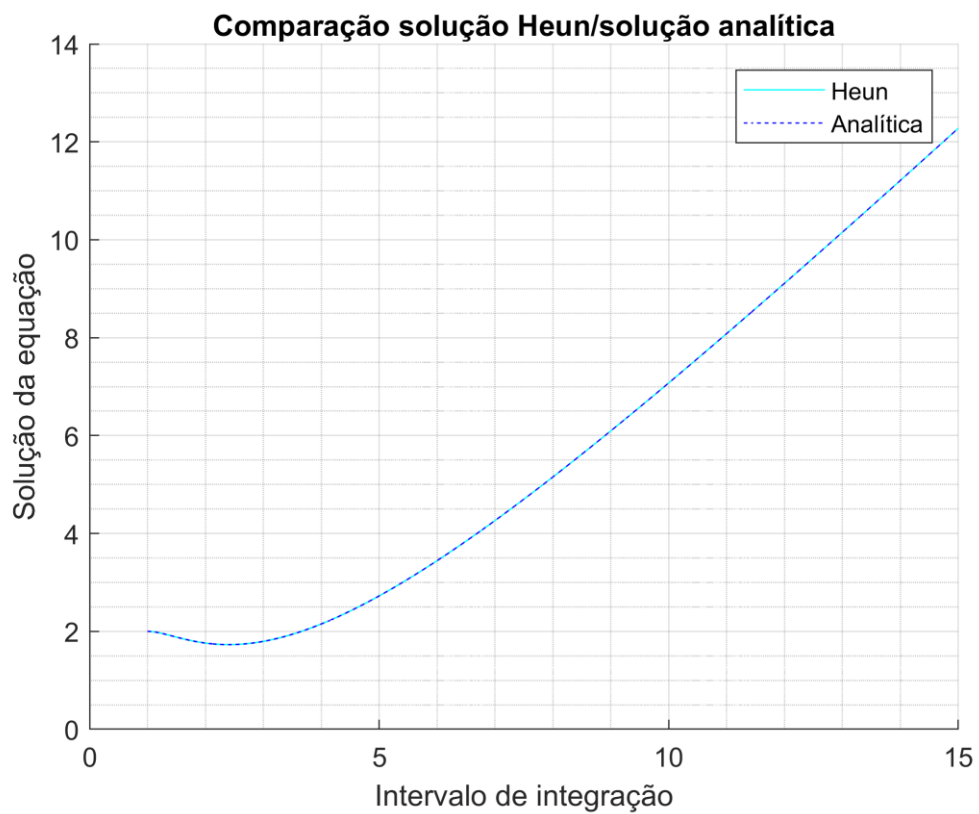


Figura 3

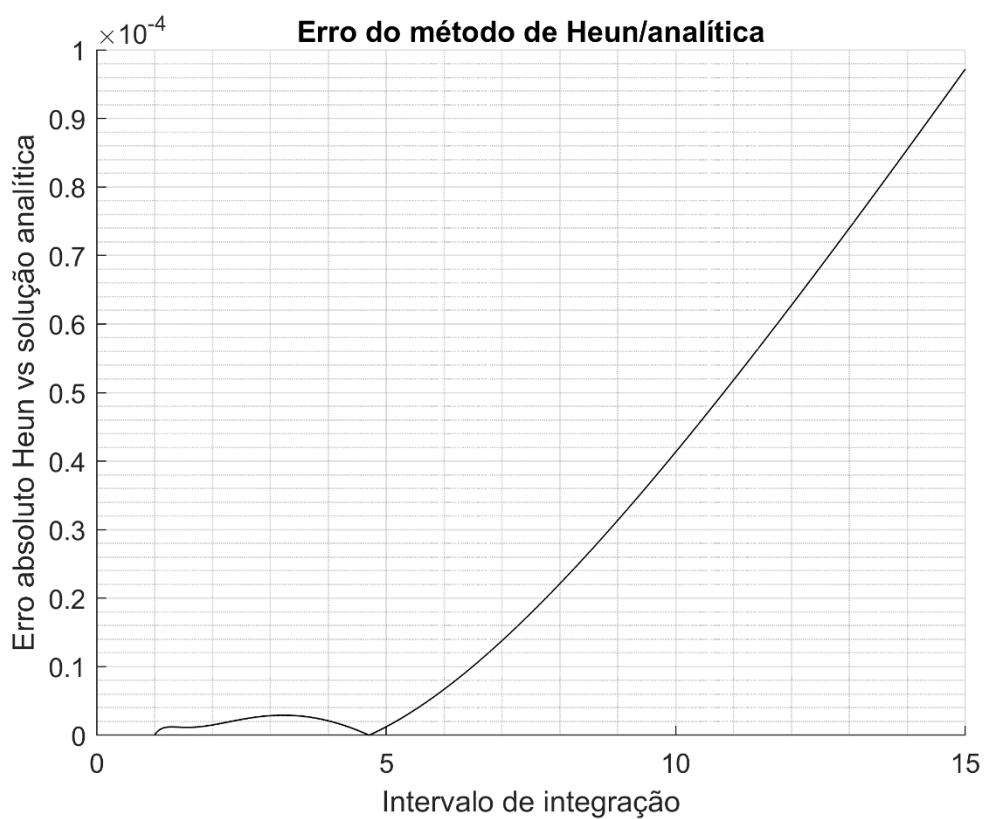


Figura 4

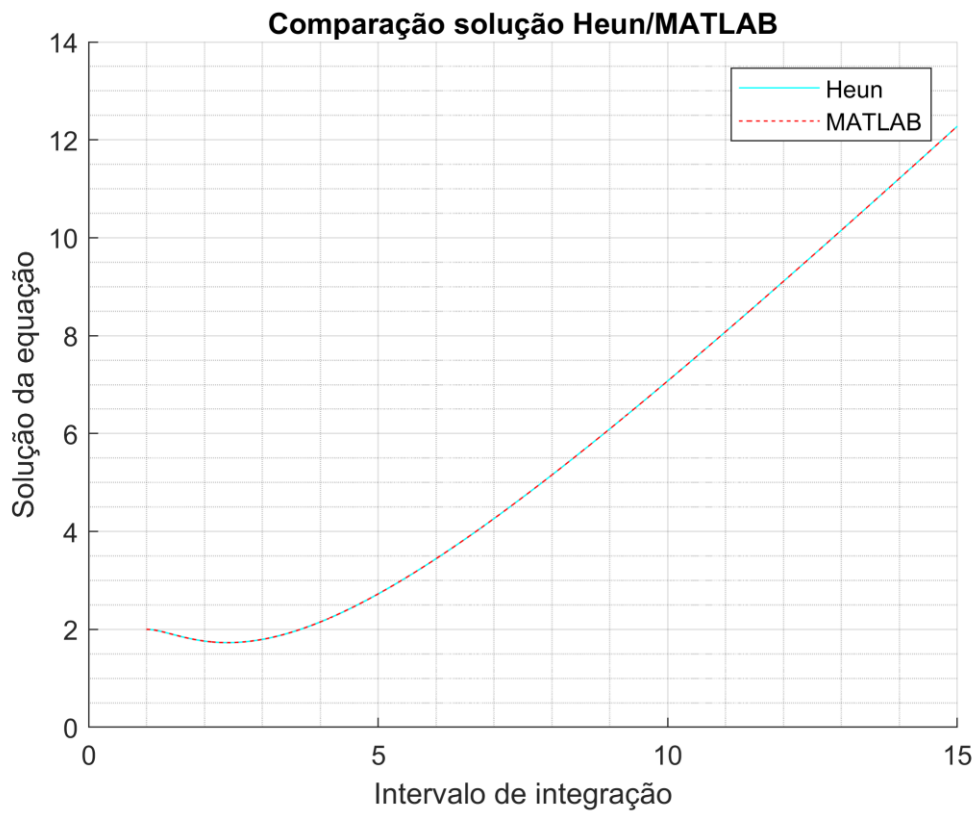


Figura 5

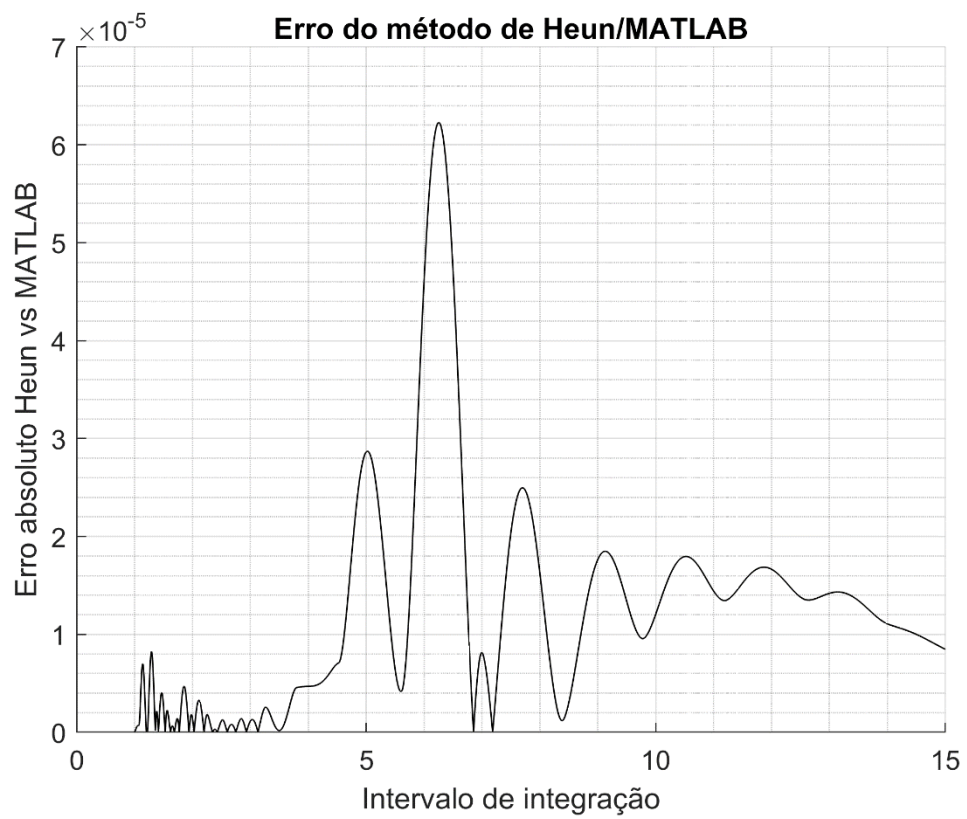


Figura 6

3.2. Método de Runge-Kutta

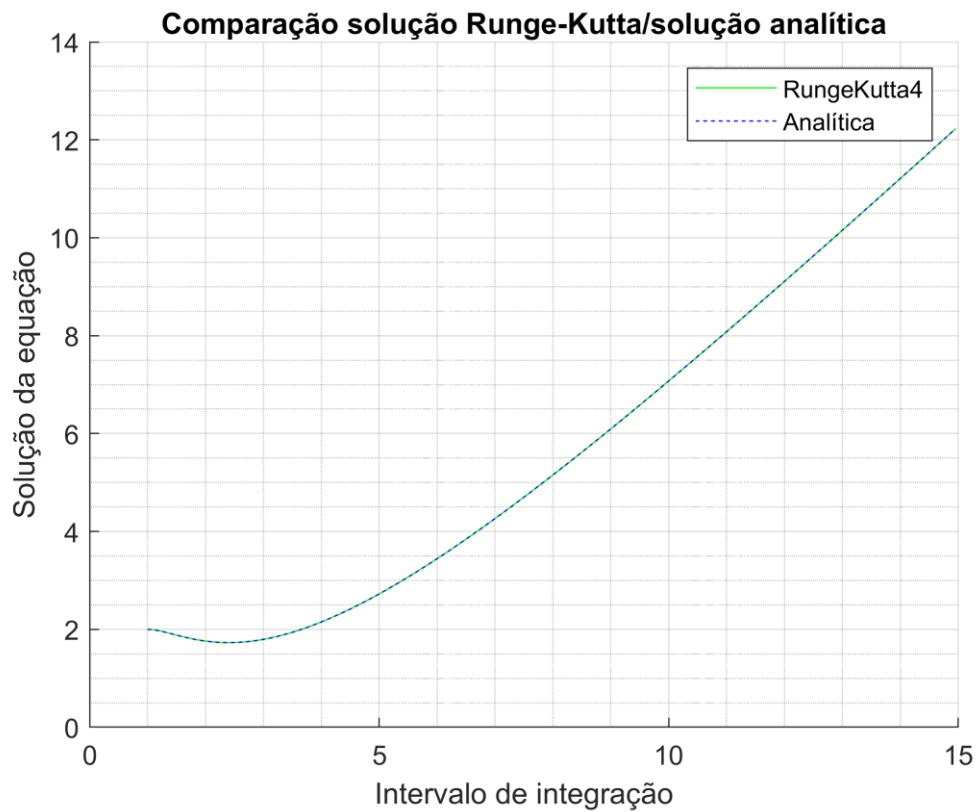


Figura 7

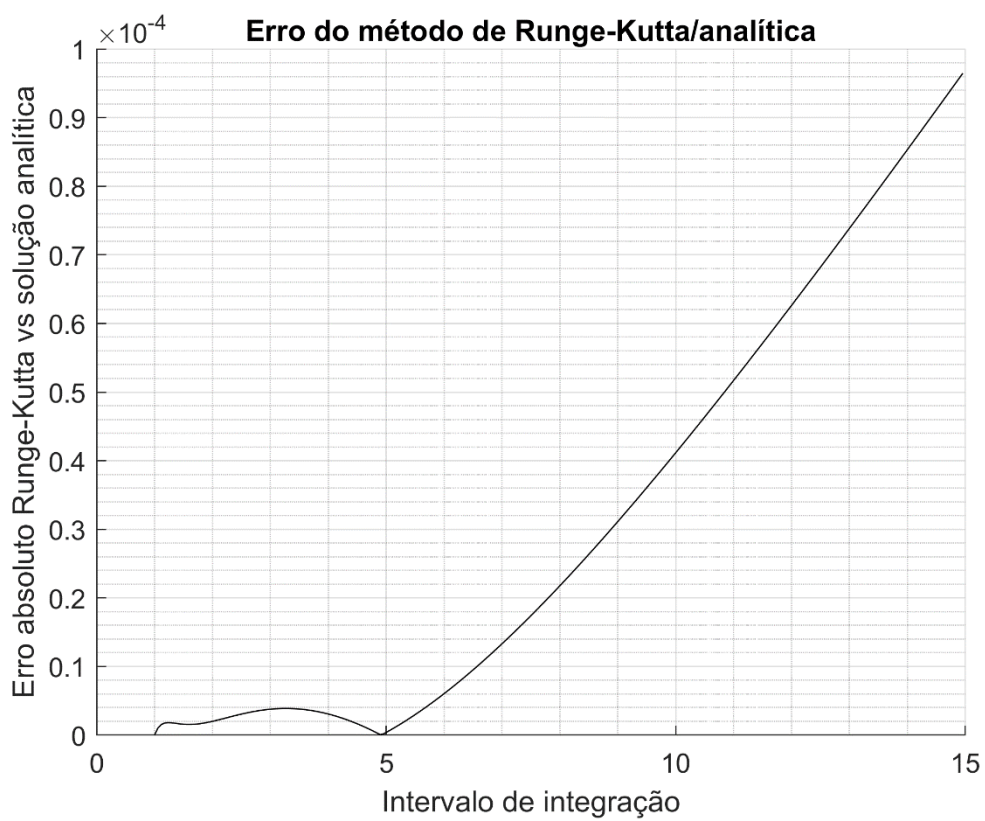


Figura 8

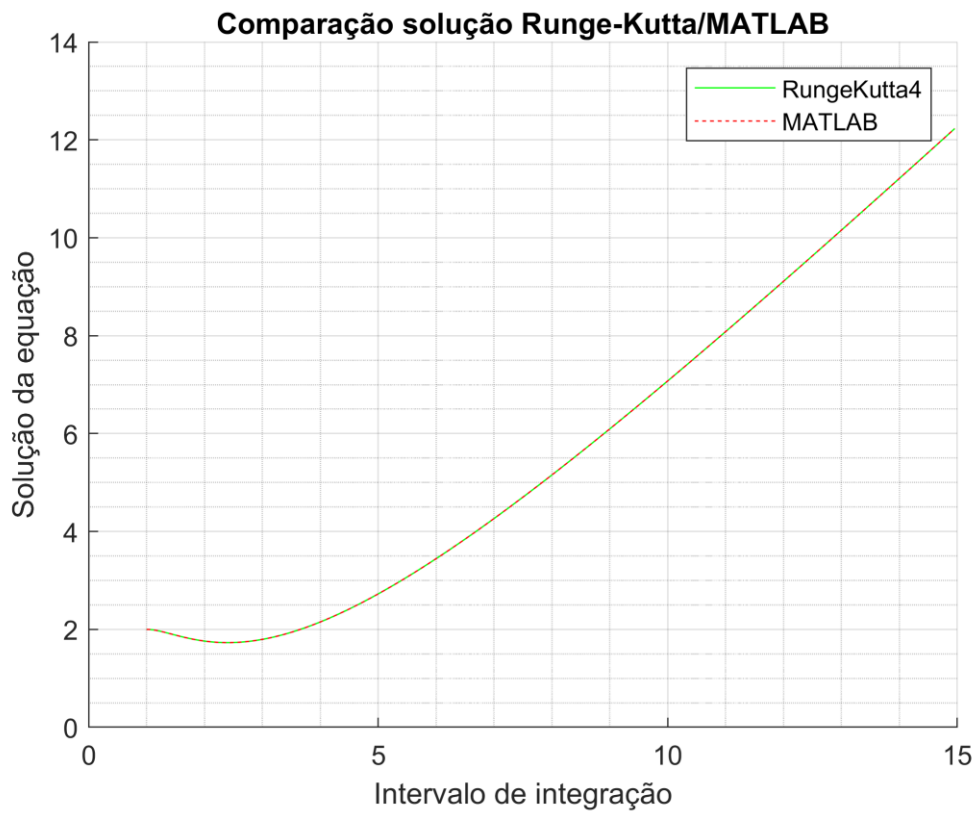


Figura 9

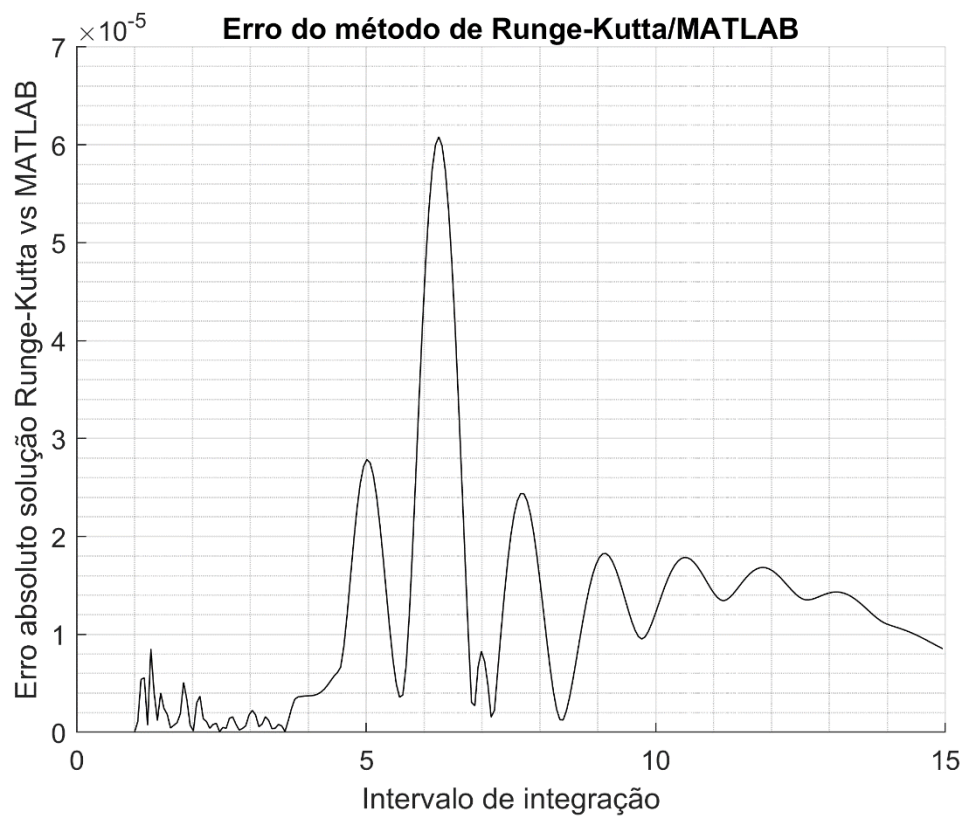


Figura 10

3.3. Método de Predição-Correção

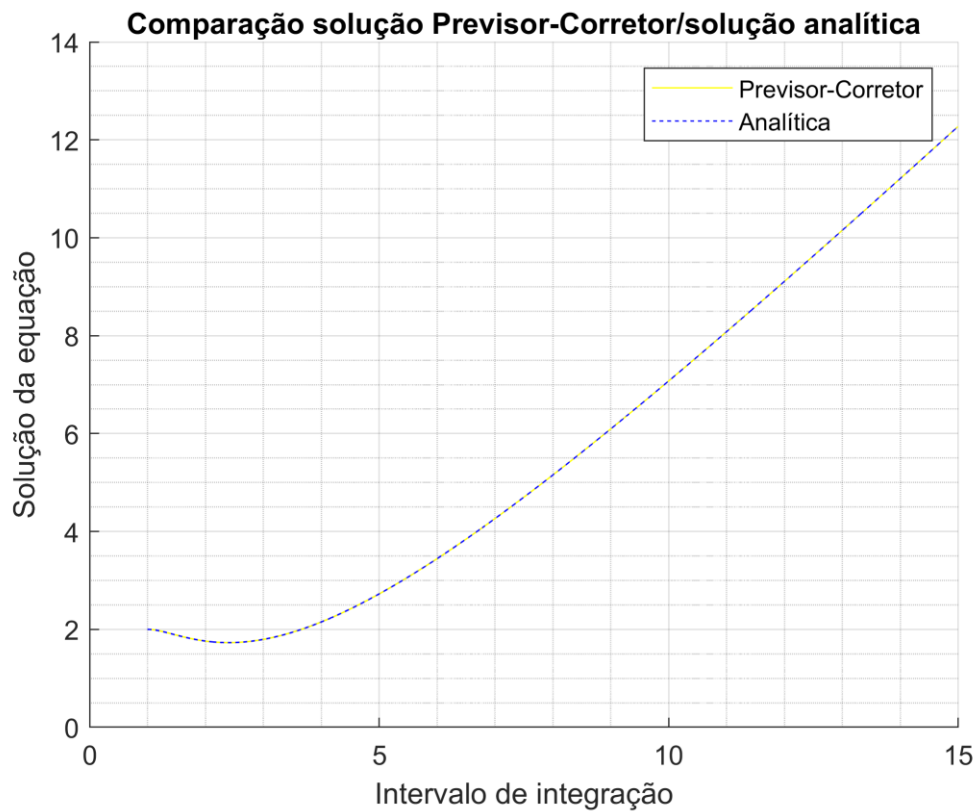


Figura 11

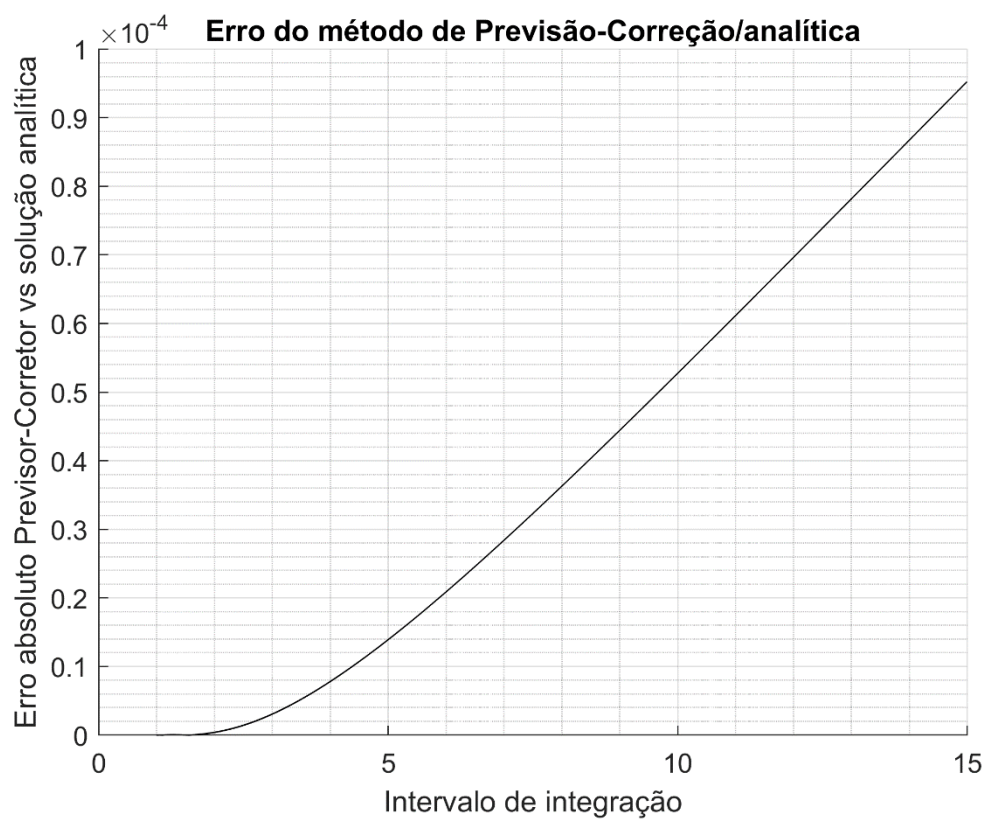


Figura 12

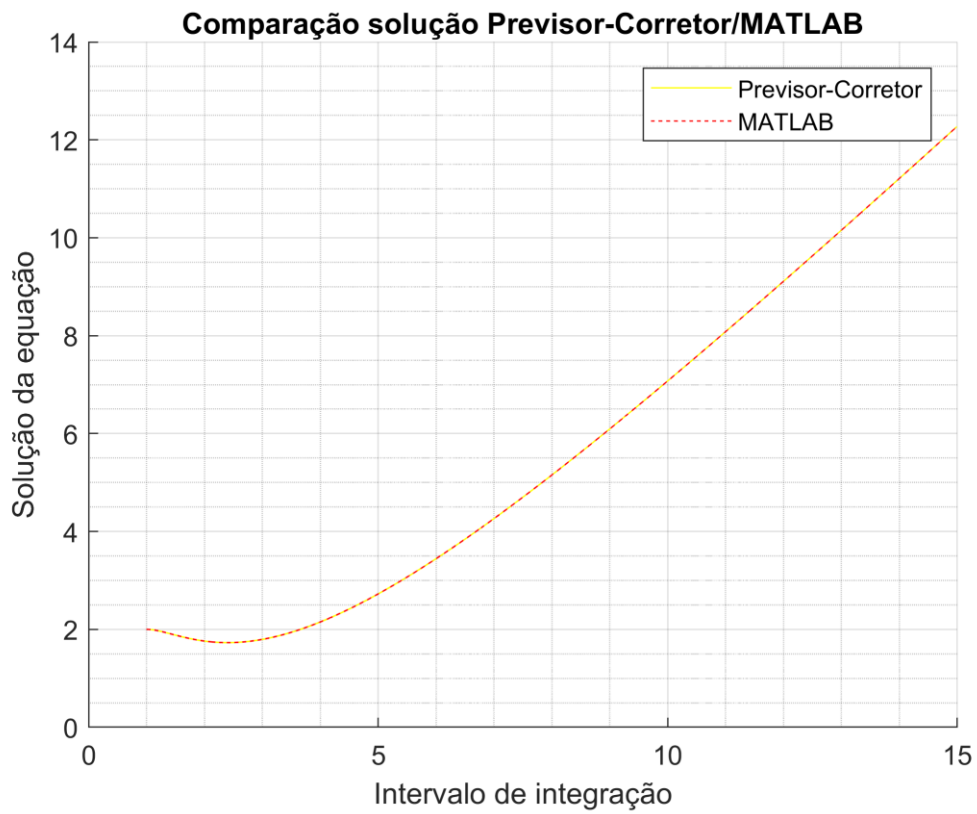


Figura 13

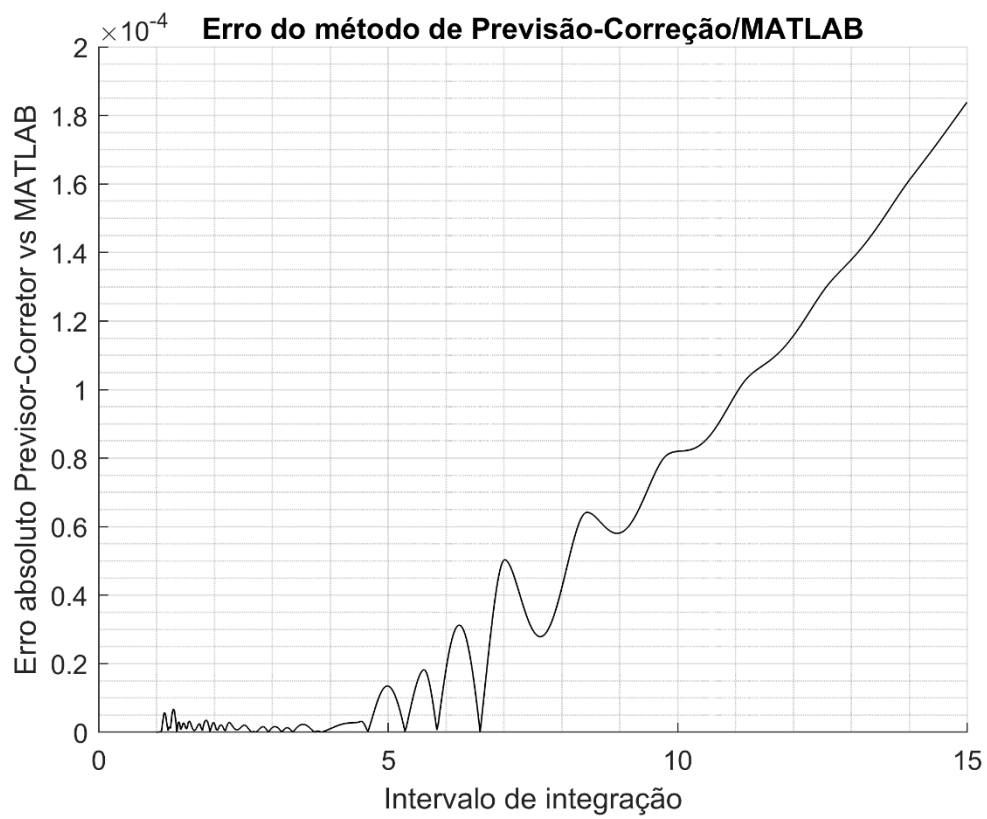


Figura 14

Calibração do método de Predição-Correção

O método de predição-correção incorpora em sua etapa de correção uma tolerância para servir de critério de parada para a correção. Essa tolerância diz respeito ao erro relativo entre duas correções. Além de calcular as correções, a implementação utilizada desse método mediu a quantidade de iterações corretivas e realizou a média dessas iterações, de modo a calibrar o parâmetro de tolerância de modo que não fossem necessárias várias iterações de correção, mas que também não fosse o caso de só haver uma iteração sempre.

Tolerância	Média do número de iterações corretivas
10^{-6}	1,101105845181675

Tabela 2: Calibração do método de Predição-Correção

Comparação Geral

	Heun	Runge-Kutta	Previsor-Corretor	MATLAB (ode45)
<i>Erro máximo (gabarito)</i>	$9,72 \cdot 10^{-5}$	$9,65 \cdot 10^{-5}$	$9,52 \cdot 10^{-5}$	$8,87 \cdot 10^{-5}$
<i>Erro médio (gabarito)</i>	$3,18 \cdot 10^{-5}$	$3,16 \cdot 10^{-5}$	$3,93 \cdot 10^{-5}$	$1,42 \cdot 10^{-5}$
<i>Erro máximo (MATLAB)</i>	$6,23 \cdot 10^{-5}$	$6,08 \cdot 10^{-5}$	$1,838 \cdot 10^{-4}$	-
<i>Erro médio (MATLAB)</i>	$1,29 \cdot 10^{-5}$	$1,26 \cdot 10^{-5}$	$6,10 \cdot 10^{-5}$	-
<i>Tamanho do passo</i>	0,0028	0,0565	0,0220	-
<i>Número de passos</i>	5000	248	636	-
<i>Tempo de execução (s)</i>	0,1228	0,0120	0,0763	0,0084
<i>Tempo médio por passo (s)</i>	$2,46 \cdot 10^{-5}$	$4,82 \cdot 10^{-5}$	$1,199 \cdot 10^{-4}$	-

Tabela 3: Comparação dos três métodos em termos de erro, tempo e passos.

Interpretação dos resultados

Por se tratar de um problema puramente matemático, a única constatação que pode ser feita acerca do resultado é a predominância do aspecto linear da solução conforme x cresce, pois as demais partes da solução ou tendem a zero, ou são oscilatórias logarítmicas e limitadas.

Conclusões

Todos os métodos implementados foram capazes de satisfazer o critério de erro aceitável com tempos de execução e tamanhos de passo razoáveis. Para este problema, todos eles são suficientes.

No entanto, pode-se constatar que dentre os três, o método de Runge-Kutta é certamente o mais eficiente, por ser superior tanto no critério do passo (ou seja, consegue melhor precisão mesmo com passos maiores) tanto no critério do tempo de execução (apesar de seu tempo por passo ser maior que o de Heun, a quantidade de passos necessária é muito menor, de modo que o tempo total é muito menor).

O *trade-off* entre tempo e precisão existe para cada método individualmente, mas nesse caso, comparando os três, o método de Runge-Kutta elimina o *trade-off* ao ser superior em ambos os critérios simultaneamente. Dessa forma, é seguro afirmar que é melhor utilizar um método mais preciso com um passo maior a um método mais simples com passo menor.

A solução MATLAB apresenta uma complicação que dificulta sua comparação direta com as demais. Por utilizar um passo adaptativo, a solução MATLAB trabalha de forma a atingir um erro total máximo, e dessa forma, limita sua precisão ao valor configurado. Contudo, quando configurados para obter o mesmo erro aceitável, a solução MATLAB e a solução Runge-Kutta tiveram desempenhos muito próximos, tanto em precisão quanto em tempo.