

INSTITUTO DE EDUCAÇÃO TECNOLÓGICA

Marco Aurélio Amarante Ribeiro

**SOLUÇÃO DE PROBLEMAS DE VALOR INICIAL USANDO OS
MÉTODOS DE RUNGE-KUTTA, DORMAND-PRINCE E DE
BULIRSCH-STOER**

Belo Horizonte

2021

Marco Aurélio Amarante Ribeiro

**SOLUÇÃO DE PROBLEMAS DE VALOR INICIAL USANDO OS
MÉTODOS DE RUNGE-KUTTA, DORMAND-PRINCE E DE
BULIRSCH-STOER**

Dissertação apresentada ao Programa de Mestrado do Instituto de Educação Tecnológica, como requisito parcial à obtenção do título de Mestre em Engenharia e Gestão de Processos e Sistemas.

Área de concentração: Engenharia e Gestão de Processos e Sistemas

Linha de pesquisa: Engenharia de processos e sistemas

Orientador: Prof. Dr. José Helvécio Martins
Instituto de Educação Tecnológica

Belo Horizonte
Ietec
2021

Ribeiro, Marco Aurélio Amarante.

R484s Solução de problemas de valor inicial usando os métodos de Runge-Kutta, Dormand-Prince e de Bulirsch-Stoer / Marco Aurélio Amarante Ribeiro. - Belo Horizonte, 2021.
1 recurso online (92 f., il., algumas color.): pdf.

Orientador: José Helvécio Martins.

Dissertação (mestrado) – Instituto de Educação Tecnológica.

Bibliografia: f. 79-80

1. Métodos numéricos - Teses. 2. Equações diferenciais ordinárias - Teses. 3. Runge-Kutta (Fórmulas) - Teses. 4. Dormand-Prince - Teses. 5. Bulirsch-Stoer - Teses. I. Martins, José Helvécio. II. Instituto de Educação Tecnológica. Mestrado em Engenharia e Gestão de Processos e Sistemas. III. Título.

CDU: 517.91



Marco Aurélio Amarante Ribeiro. *Solução de problemas de valor inicial usando os métodos de Runge-Kutta, Dormand-Prince e de Bulirsch-Stoer.*

Dissertação apresentada ao Programa de Mestrado em Engenharia e Gestão de Processos e Sistemas do Instituto de Educação Tecnológica, como requisito parcial à obtenção do título de Mestre em Engenharia e Gestão de Processos e Sistemas.

Área de concentração: Engenharia e Gestão de Processos e Sistemas

Linha de Pesquisa: Engenharia de processos e sistemas

Orientador: Prof. Dr. José Helvécio Martins
Faculdade Ietec

Aprovada pela banca examinadora constituída pelos membros:

Prof. Dr. Eder Marinho Martins – UFOP (Examinador Externo)

Profa. Dra. Wanyr Romero Ferreira – IETEC

Prof. Dr. José Helvécio Martins – Orientador

Belo Horizonte, 13 de maio de 2021

Dedico este trabalho, em primeiro lugar, à minha mãe, Jane Amaral Amarante Ribeiro, salientando que aqui não há espaço suficiente para agradecê-la e exaltá-la o quanto ela merece.

Ao meu pai e aos seus pais, por toda a influência que exerceram sobre mim.

AGRADECIMENTOS

Ao meu orientador, Prof. José Helvécio Martins, por sua amizade, trabalho e dedicação.

A todos os meus Professores, colegas e funcionários do IETEC e da UFMG.

Aos meus irmãos Almir Amarante Ribeiro e Cinthia Amarante Ribeiro, pela leitura e pelas sugestões de correção deste trabalho.

Ao meu filho Mateus por toda a torcida e acompanhamento deste trabalho

À Pessoa que *iluminou* minha volta ao mestrado, mas quis se manter anônima e assim como apareceu, desapareceu.

A todos os deuses do universo, existentes ou não.

*“Os filósofos se limitaram a interpretar o mundo.
Cabe a nós transformá-lo.”*

Karl Marx

RESUMO

Vários problemas encontrados nas ciências e, particularmente, em engenharia, podem ser resolvidos pela modelagem matemática que resulta, na maior parte, em equações diferenciais. Equações diferenciais ordinárias são subconjunto desse universo, que constituem a formulação de problemas de valor inicial de uma variedade de processos e sistemas. A solução desses problemas pode ser obtida pela solução das equações diferenciais que descrevem a dinâmica do sistema. Uma das ferramentas fundamentais para obter essas soluções são os métodos numéricos, vários deles já incorporados a alguma plataforma computacional, como, por exemplo, o Matlab. Nesta dissertação, o objetivo foi utilizar os métodos de Runge-Kutta de passo fixo e o método de Bulirsch-Stoer, para resolver três problemas típicos envolvendo equações diferenciais ordinárias e comparar conjuntamente os resultados com o método de Dormand-Prince, incorporado ao Matlab pela sub-rotina ode45. Para os métodos de Runge-Kutta e de Bulirsch-Stoer foram escritas duas sub-rotinas em linguagem do Matlab. Foram resolvidos três problemas: 1) uma equação diferencial ordinária simples com solução analítica, usada como referência; 2) modelagem de um sistema de suspensão de um veículo; e 3) a equação de Page. Para o problema 1, todos os métodos forneceram solução adequada com tamanho do passo de integração apropriado. Para o problema 2, o método de Dormand-Prince foi inflexível em manter a solução estável mesmo para tamanhos do passo maior. Os métodos de Runge-Kutta e Bulirsch-Stoer, neste caso, funcionaram bem para passos reduzidos. No problema 3, o método que forneceu uma solução estável adequada foi Bulirsch-Stoer. Os outros também funcionam, para tamanhos de passo muito reduzidos. Portanto, concluiu-se que o método adequado para solução de problemas de valor inicial depende da natureza do problema e da escolha adequada do tamanho do passo de integração.

Palavras-chave: Métodos numéricos. Equações diferenciais ordinárias. Runge-Kutta. Dormand-Prince. Bulirsch-Stoer.

ABSTRACT

Several problems encountered in sciences and, particularly engineering, can be solved by mathematical modeling that results, in general, in differential equations. Ordinary differential equations are subsets of this universe, which constitute the formulation of initial value problems for a variety of processes and systems. The solution of these problems can be obtained by solving the differential equations that describe the dynamics of the system. One of the fundamental tools to obtain these solutions are the numerical methods, several of which have already been incorporated into some computational platform, such as, for example, Matlab. In this dissertation, the objective was to use the fixed-step Runge-Kutta methods and the Bulirsch-Stoer method, to solve three typical problems involving ordinary differential equations and to jointly compare the results with the Dormand-Prince method, incorporated into Matlab by ode45 subroutine. For the Runge-Kutta and Bulirsch-Stoer methods, two subroutines were written in Matlab language. Three problems were solved: 1) a simple ordinary differential equation with an analytical solution, used as a reference; 2) modeling of a vehicle suspension system; and 3) the Page equation. For problem 1, all methods provided an adequate solution with the appropriate integration step size. For problem 2, the Dormand-Prince method was relentless in keeping the solution stable even for larger step sizes. The Runge-Kutta and Bulirsch-Stoer methods, in this case, worked well for reduced steps. In problem 3, the method that provided an adequate stable solution was Bulirsch-Stoer. The others also work, for very small step sizes. Therefore, it was concluded that the appropriate method for solving initial value problems depends on the nature of the problem and the appropriate choice of the integration step-size.

Keywords: Numerical methods. Ordinary differential equations. Runge-Kutta. Dormand-Prince. Bulirsch-Stoer.

LISTA DE FIGURAS

Figura 1 - Ilustração esquemática do método de Euler	24
Figura 2 - Ilustração esquemática do método do ponto médio ou Runge-Kutta de segunda ordem.....	27
Figura 3 - Ilustração esquemática do método de Runge-Kutta de quarta ordem.....	29
Figura 4 - Região de estabilidade absoluta do par Dormand-Prince	32
Figura 5 - Duplicação do passo como um meio para o controle adaptativo do tamanho do passo no método de Runge-Kutta de quarta ordem.....	35
Figura 6 - Solução analítica e numérica da equação diferencial, usando $dy/dt + a.y = b$ do método de Runge-Kutta de primeira ordem, $k = 1$, com passo fixo, $h = 0,1$	47
Figura 7 - Variação do erro relativo (%) da equação diferencial $dy/dt + a.y = b$, usando o método de Runge-Kutta de primeira ordem, $k = 1$, com passo fixo, $h = 0,1$	47
Figura 8 - Solução analítica e numérica da equação diferencial $dy/dt + a.y = b$, usando o método de Runge-Kutta de segunda ordem, $k = 2$, com passo fixo, $h = 0,1$	48
Figura 9 - Variação do erro relativo (%) da equação diferencial $dy/dt + a.y = b$, usando o método de Runge-Kutta de segunda ordem, $k = 2$, com passo fixo, $h = 0,1$	48
Figura 10 - Solução analítica e numérica da equação diferencial $dy/dt + a.y = b$, usando o método de Runge-Kutta de terceira ordem, $k = 3$, com passo fixo, $h = 0,1$	49
Figura 11 - Variação do erro relativo (%) da equação diferencial $dy/dt + a.y = b$, usando o método de Runge-Kutta de terceira ordem, $k = 3$, com passo fixo, $h = 0,1$	49
Figura 12 - Solução analítica e numérica da equação diferencial $dy/dt + a.y = b$, usando o método de Runge-Kutta de quarta ordem, $k = 4$, com passo fixo, $h = 0,1$	50
Figura 13 - Variação do erro relativo (%) da equação diferencial $dy/dt + a.y = b$, usando o método de Runge-Kutta de quarta ordem, $k = 4$, com passo fixo, $h = 0,1$	50

Figura 14 - Solução analítica e numérica da equação diferencial $dy/dt + a.y = b$, usando o método de Runge-Kutta de quinta ordem, $k = 5$, com passo fixo, $h = 0,1$	51
Figura 15 - Variação do erro relativo (%) da equação diferencial $dy/dt + a.y = b$, usando o método de Runge-Kutta de segunda ordem, $k = 5$, com passo fixo, $h = 0,1$	51
Figura 16 - Solução analítica e numérica da equação diferencial $dy/dt + a.y = b$, usando o método de Runge-Kutta de primeira ordem, $k = 1$, com passo fixo, $h = 0,01$	52
Figura 17 - Variação do erro relativo (%) da equação diferencial $dy/dt + a.y = b$, usando o método de Runge-Kutta de segunda ordem, $k = 1$, com passo fixo, $h = 0,01$	52
Figura 18 - Solução analítica e numérica da equação diferencial $dy/dt + a.y = b$, usando o método de Runge-Kutta de segunda ordem, $k = 2$, com passo fixo, $h = 0,01$	53
Figura 19 - Variação do erro relativo (%) da equação diferencial $dy/dt + a.y = b$, usando o método de Runge-Kutta de segunda ordem, $k = 2$, com passo fixo, $h = 0,01$	53
Figura 20 - Solução analítica e numérica da equação diferencial $dy/dt + a.y = b$, usando o método de Runge-Kutta de terceira ordem, $k = 3$, com passo fixo, $h = 0,01$	54
Figura 21 - Variação do erro relativo (%) da equação diferencial $dy/dt + a.y = b$, usando o método de Runge-Kutta de segunda ordem, $k = 3$, com passo fixo, $h = 0,01$	54
Figura 22 - Solução analítica e numérica da equação diferencial $dy/dt + a.y = b$, usando o método de Runge-Kutta de quarta ordem, $k = 4$, com passo fixo, $h = 0,01$	55
Figura 23 - Variação do erro relativo (%) da equação diferencial $dy/dt + a.y = b$, usando o método de Runge-Kutta de segunda ordem, $k = 4$, com passo fixo, $h = 0,01$	55
Figura 24 - Solução analítica e numérica da equação diferencial $dy/dt + a.y = b$, usando o método de Runge-Kutta de quinta ordem, $k=5$, com passo fixo, $h = 0,01$	56

Figura 25 - Variação do erro relativo (%) da equação diferencial $dy/dt + a.y = b$, usando o método de Runge-Kutta de segunda ordem, $k = 5$, com passo fixo, $h = 0,01$	56
Figura 26 - Solução analítica e numérica da equação diferencial $dy/dt + a.y = b$, usando o método de Dormand-Prince, com passo $h = 0,1$	59
Figura 27 - Variação do erro relativo (%) da equação diferencial $dy/dt + a.y = b$, usando o método de Dormand-Prince, com passo $h = 0,1$	60
Figura 28 - Solução analítica e numérica da equação diferencial $y' + a y = b$, usando o método de Runge-Kutta de quinta ordem, $k = 5$, com passo fixo, $h = 0,1$	60
Figura 29 - Variação do erro relativo (%) da equação diferencial $dy/dt + a.y = b$, usando o método de Runge-Kutta de quinta ordem, $k = 5$, com passo fixo, $h = 0,1$	61
Figura 30 - Solução analítica e numérica da equação diferencial $dy/dt + a.y = b$, usando o método de Bulirsch-Stoer, com passo $h = 0,1$	61
Figura 31 - Variação do erro relativo (%) da equação diferencial $dy/dt + a.y = b$, usando o método de Bulirsch-Stoer, com passo $h = 0,1$	62
Figura 32 - Solução analítica e numérica da equação diferencial $dy/dt + a.y = b$, usando o método de Dormand-Prince, com passo $h = 0,01$	62
Figura 33 - Variação do erro relativo (%) da equação diferencial $dy/dt + a.y = b$, usando o método de Dormand-Prince, com passo $h = 0,01$	63
Figura 34 - Solução analítica e numérica da equação diferencial $dy/dt + a.y = b$, usando o método de Runge-Kutta de quinta ordem, $k = 5$, com passo fixo, $h = 0,01$	63
Figura 35 - Variação do erro relativo (%) da equação diferencial $dy/dt + a.y = b$, usando o método de Runge-Kutta de quinta ordem, $k = 5$, com passo fixo, $h = 0,01$	64
Figura 36 - Solução analítica e numérica da equação diferencial $dy/dt + a.y = b$, usando o método de Bulirsch-Stoer, com passo $h = 0,01$	64
Figura 37 - Variação do erro relativo (%) da equação diferencial $dy/dt + a.y = b$, usando o método de Bulirsch-Stoer, com passo $h = 0,01$	65
Figura 38 - Modelo do sistema de suspensão de um veículo (um quarto de veículo)	66

Figura 39 - Diagrama de corpo livre, assumindo que $\dot{y} > \dot{x}$ e $y > x$. Somente a força dinâmica é mostrada porque a força estática é cancelada pela força gravitacional mg	66
Figura 40 - Tempo computacional da solução do problema do sistema de suspensão de um veículo em função do tamanho do passo de integração	68
Figura 41 - Soluções numéricas do problema de sistema de suspensão de um veículo, usando o método de Runge-Kutta de quarta ordem ($k = 4$), com passo fixo, e os métodos de passo adaptativo de Dormand-Prince e de Bulirsch-Stoer, com $h = 0,05$	69
Figura 42 - Soluções numéricas do problema de sistema de suspensão de um veículo, usando o método de Runge-Kutta de quarta ordem ($k = 4$), com passo fixo, e os métodos de passo adaptativo de Dormand-Prince e de Bulirsch-Stoer, com $h = 0,025$	69
Figura 43 - Soluções numéricas do problema de sistema de suspensão de um veículo, usando o método de Runge-Kutta de quarta ordem ($k = 4$), com passo fixo, e os métodos de passo adaptativo de Dormand-Prince e de Bulirsch-Stoer, com $h = 0,0125$	70
Figura 44 - Soluções numéricas do problema de sistema de suspensão de um veículo, usando o método de Runge-Kutta de quarta ordem ($k = 4$), com passo fixo, e os métodos de passo adaptativo de Dormand-Prince e de Bulirsch-Stoer, com $h = 0,00125$	70
Figura 45 - Soluções numéricas da equação de Page, usando os métodos de Runge-Kutta de quarta ordem ($k = 4$), Dormand-Prince e Bulirsch-Stoer. $h = 10$ [min]	74
Figura 46 - Variação do erro relativo de aproximação na solução numérica da equação de Page, usando o método de Bulirsch-Stoer. $h = 10$ [min 5]	74
Figura 47 - Soluções numéricas da equação de Page, usando os métodos de Runge-Kutta de quarta ordem ($k = 4$), Dormand-Prince e Bulirsch-Stoer. $h = 5$ [min]	75
Figura 48 - Variação do erro relativo de aproximação na solução numérica da equação de Page, usando o método de Bulirsch-Stoer. $h = 5$ [min]	75

LISTA DE QUADROS

Quadro 1 - Quadro de Butcher para o método de Runge-Kutta de Segunda ordem.	26
Quadro 2 - Quadro de Butcher para o par Dormand-Prince de Sete Estágios com Interpelador.	31
Quadro 3 - Quadro de Butcher para o par Runge-Kutta-Verner de Oito Estágios.....	34

LISTA DE TABELAS

Tabela 1 - Tempo computacional e erro máximo na solução da Equação (24) pelos métodos de Runge-Kutta explícitos de um passo.	46
Tabela 2 - Tempo computacional e erro máximo na solução da Equação (24) usando os métodos de Runge-Kutta de quinta ordem, Dormand-Prince e Bulirsch-Stoer.....	58
Tabela 3 - Tempo computacional e estimativas finais da solução do problema do sistema de suspensão de um veículo.	68
Tabela 4 - Tempo computacional da solução da equação de secagem de Page.....	73

SUMÁRIO

1	INTRODUÇÃO	18
2	OBJETIVOS	20
2.1	Geral.....	20
2.2	Específicos	20
3	BASES TEÓRICAS.....	21
3.1	Problemas envolvendo equações diferenciais.....	21
3.2	Métodos numéricos para problemas de valor inicial.....	23
3.2.1	Métodos de Euler e Euler Melhorado	24
3.2.1.1	Método de Euler	24
3.2.1.2	Método de Euler Melhorado	25
3.2.1.3	Métodos de Runge-Kutta Explícitos	25
3.2.1.4	Método de Runge-Kutta de Primeira Ordem	25
3.2.1.5	Método de Runge-Kutta de Segunda Ordem	26
3.2.1.6	Método de Runge-Kutta de Terceira Ordem.....	28
3.2.1.7	Método de Runge-Kutta de Quarta Ordem.....	28
3.2.1.8	Pares Integrados de Métodos de Runge-Kutta.....	29
3.2.1.9	Par de Métodos de Runge-Kutta Integrados de Quatro Estágios.....	29
3.2.1.10	Par Dormand-Prince de Sete Estágios com Interpolador	30
3.2.1.11	Método de Runge-Kutta-Fehlberg de Seis Estágios.....	32
3.2.1.12	Par de Métodos de Runge-Kutta-Verner de Oito Estágios	33
3.2.2	Controle Adaptativo de Passos nos Métodos de Runge-Kutta.....	34
3.2.3	Método de Bulirsch-Stoer	37
3.2.3.1	Método do Ponto Médio Modificado	37
3.2.3.2	Extrapolção de Richardson.....	39
3.2.3.3	O Método de Bulirsch-Stoer	39
3.2.3.4	Controle do Tamanho de Passo para o Método de Bulirsch-Stoer.....	41
3.2.3.5	Observações Finais.....	43
4	METODOLOGIA	44
4.1	Implementação e teste de uma rotina em Matlab que incorpora os métodos de Runge-Kutta de ordem 1, 2, 3, 4 e 5	44

4.1.1	Implementação da Rotina FixedRK.....	44
4.1.2	Teste da Sub-Rotina.....	44
4.2	Implementação e teste da sub-rotina BulirschStoer junto com FixedRK e ode45.....	45
4.2.1	Implementação da Rotina BulirschStoer	45
4.2.2	Teste da Sub-Rotina BulirschStoer	45
4.3	Solução de dois problemas usando os métodos Dormand-Prince, Runge-Kutta de ordem 5 E Bulirsch-Stoer	45
5	RESULTADOS E DISCUSSÃO	46
5.1	Implementação e teste de uma rotina em Matlab que implementa os métodos de Runge-Kutta de ordem 1, 2, 3, 4 e 5	46
5.1.1	Implementação da Rotina FixedRK.....	46
5.1.2	Teste da Sub-Rotina.....	46
5.2	Implementação e teste da sub-rotina BulirschStoer junto com FixedRK e ode45.....	57
5.2.1	Implementação da Rotina BulirschStoer	57
5.2.2	Teste da Sub-Rotina BulirschStoer	57
5.3	Solução de dois problemas usando os métodos Dormand-Prince, Runge-Kutta de ordem 5 e Bulirsch-Stoer.....	65
5.3.1	Modelagem do sistema de suspensão de um veículo	65
5.3.2	Solução da equação de secagem de Page	71
5.4	Considerações finais	76

6	CONCLUSÕES	78
	REFERÊNCIAS	79
	APÊNDICE A - Sub-Rotina FixedRK.....	81
	APÊNDICE B - Sub-Rotina BulirschStoer	83
	APÊNDICE C - Amostras de Resultados do Problema 1.....	88
	APÊNDICE D - Amostras de Resultados do Problema 2.....	89
	APÊNDICE E - Amostras de Resultados do Problema 3.....	90
	APÊNDICE F - Solução Analítica do Problema 2	91
	APÊNDICE G - Solução Analítica do Problema 3.....	92

1 INTRODUÇÃO

Equações Diferenciais Ordinárias (EDOs) para cientistas e engenheiros constituem uma ferramenta essencial na solução de vários problemas importantes, sem a necessidade de focalizar profundamente nos aspectos teóricos dos modelos matemáticos utilizados.

Os métodos de resolução de equações diferenciais ordinárias se iniciaram de maneira natural com a constatação simples da relação inversa entre diferenciação e integração. Inicialmente, constatou-se que a maioria das equações diferenciais não podia ser resolvida apenas por integração simples ou por soluções analíticas (BRONSON; COSTA, 2014).

Quando se iniciaram os estudos de métodos de resolução de EDOs, rapidamente constatou-se que a maioria delas não podia ser resolvida apenas por integração simples ou por soluções analíticas (BRONSON; COSTA, 2014). Em muitos casos, ou em sua maioria, tornou-se necessária a utilização de métodos numéricos para obter uma solução aproximada, com acurácia desejada. Isto só foi possível com o advento, inicialmente de calculadoras eletrônicas, e de computadores com capacidade de processamento muito maior do que essas calculadoras, com a utilização de algoritmos computacionais mais eficientes, possibilitando a solução de problemas complexos envolvendo equações diferenciais (BASSANEZI; FERREIRA Jr., 1988).

Os Métodos de resolução das EDOs podem ser vistos como originados de modelos diferentes, e essas diferentes visões, em geral, lançam mais luz sobre o problema; e a computação está, ou pelo menos deveria estar intimamente ligada à origem do problema e ao uso que será feito das respostas, não sendo, portanto, um passo a ser dado separadamente da realidade e que está inserido em um contexto mais amplo que envolve a análise numérica. Neste contexto, um método numérico é uma ferramenta matemática projetada para resolver problemas numéricos e a sua implementação, com uma verificação de convergência apropriada em uma linguagem de programação, é criado o algoritmo numérico e por fim implementado em uma linguagem de programação ou em uma plataforma (QUARTERONI; SACCO; SALERI, 2017).

Dentre os precursores no desenvolvimento de métodos numéricos, destaca-se Leonhard Paul Euler (1707-1783), importante matemático e cientista suíço, considerado um dos maiores estudiosos da matemática, em sua época (BOYER, 1996). Euler talvez tenha sido o maior responsável pelos métodos de resoluções numéricas usados na solução de problemas envolvendo equações diferenciais ordinárias. O método de Euler, devido à sua simplicidade e facilidade de aplicação, é utilizado para facilitar a compreensão dos conceitos envolvidos nos métodos de integração numérica, além de fornecer resultados satisfatórios para alguns problemas, sob certas condições e restrições (BISWAS *et al.*, 2013).

Vários outros métodos foram desenvolvidos, que fornecem resultados mais precisos para problemas mais complexos, embora necessitem de algoritmos mais sofisticados e maior tempo computacional. Dentre os métodos numéricos clássicos mais modernos citam-se os métodos de Runge-Kutta, e seus aprimoramentos, e o método de Bulirsch-Stoer (PRESS *et al.*, 2007).

Encontram-se na literatura vários trabalhos com aplicações de métodos numéricos para a solução dos mais variados tipos de problemas. Entretanto, detalhamentos sobre o desenvolvimento e a fundamentação teórica desses métodos são encontrados em livros-textos e/ou técnicos sobre o assunto (GEAR, 1971; NAGLE; SAFF; SNIDER, 2012; ZILL, 2016).

Artigos técnico-científicos sobre o tema são encontrados de maneira geral e em abundância na abordagem em dissertações e teses e não contém detalhamento teórico sobre os métodos utilizados na obtenção dos resultados. Entretanto, esses artigos servem como referência para comparação de resultados de validação de novas metodologias ou técnicas de abordagem para solução de novos problemas (DUTTA *et al.*, 2020; GHANBARI; BALEANU, 2020; HOSSEINI *et al.*, 2020; OBERLEITHNERA; PASCHEREITA; SORIAB, 2015; VANANI; AMINATAEI, 2011).

Nesta dissertação, o objetivo foi utilizar os métodos de Runge-Kutta de passo fixo e o método de Bulirsch-Stoer, para resolver três problemas típicos envolvendo equações diferenciais ordinárias e comparar, conjuntamente, os resultados com o método de Dormand-Prince, incorporado ao Matlab pela sub-rotina ode45.

2 OBJETIVOS

2.1 Geral

Comparar a solução de problemas de valor inicial usando os métodos de Runge-Kutta de passo fixo, o método de Dormand-Prince e o método de Bulirsch-Stoer.

2.2 Específicos

- a) Escrever uma sub-rotina computacional para implementar os métodos de Runge-Kutta de passo fixo;
- b) Escrever uma sub-rotina computacional para implementar o método de Bulirsch-Stoer;
- c) Testar as duas sub-rotinas implementadas na solução de três problemas de valor inicial típicos;
- d) Comparar os resultados conjuntamente com as soluções obtidas usando o método de Dormand-Prince já incorporado na sub-rotina ode45 do MATLAB.

3 BASES TEÓRICAS

Nesta seção apresentam-se uma abordagem sobre dois aspetos:

- a) problemas envolvendo equações diferenciais ordinárias;
- b) métodos numéricos para solução de problemas de valor inicial.

3.1 Problemas envolvendo equações diferenciais

Os problemas analisados neste trabalho são restritos àqueles envolvendo equações diferenciais ordinárias. Se y é uma função de x e y' , y'' , ..., $y^{(n)}$ denotam suas derivadas, uma equação diferencial ordinária (EDO) é uma equação que envolve a variável independente, x , a variável dependente, y , e suas derivadas, y' , y'' , ..., $y^{(n)}$, escrita na sua forma implícita como (NAGLE; SAFF; SNIDER, 2012; ZILL, 2016):

$$F\left(x, y, \frac{dy}{dx}, \frac{d^2y}{dx^2}, \dots, \frac{d^ny}{dx^n}\right) = 0 \quad (1a)$$

Consideram-se que a equação diferencial se aplica para todo x no intervalo aberto $a < x < b$, em que a ou b podem ser infinitos. Em muitos casos, é possível isolar o termo de mais alta ordem e reescrever a equação (1a) explícita como:

$$\frac{d^ny}{dx^n} = f\left(x, y, \frac{dy}{dx}, \frac{d^2y}{dx^2}, \dots, \frac{d^{(n-1)}y}{dx^{(n-1)}}\right) \quad (1b)$$

Uma solução da EDO é uma função $y(x)$ cujas derivadas satisfazem à equação (1), embora não esteja garantido que tal função exista, e caso exista, normalmente ela não é única.

As equações diferenciais ordinárias (EDOs) surgem em muitos casos quando se usam técnicas de modelagem matemática para descrever fenômenos na ciência, engenharia, economia, dentre outras. Na maioria dos casos, o modelo é muito complexo, tornando-se difícil ou impossível encontrar uma solução exata, ou mesmo uma solução aproximada manualmente, sendo necessária uma simulação computacional confiável e eficiente.

Problemas envolvendo equações diferenciais ordinárias (EDOs) sempre podem ser reduzidos ao estudo de conjuntos (ou sistemas) de equações diferenciais de primeira ordem (NAGLE; SAFF; SNIDER, 2012; ZILL, 2016). Por exemplo, considere a equação de segunda ordem, expressa pela equação (2):

$$\frac{d^2y}{dx^2} + q(x) \frac{dy}{dx} = r(x) \quad (2)$$

A equação (2) pode ser reescrita como a equação (3), introduzindo uma nova variável, z , constituindo-se um exemplo de procedimento para uma equação diferencial ordinária arbitrária.

$$\begin{cases} \frac{dy}{dx} = z(x) \\ \frac{dz}{dx} = r(x) - q(x)z(x) \end{cases} \quad (3)$$

Uma escolha usual para as novas variáveis é defini-las como sendo as derivadas umas das outras, e da variável original. Às vezes, é útil incorporar na definição dessas variáveis alguns fatores presentes na equação, ou algumas potências da variável independente, com o intuito de mitigar comportamento singular que possa resultar em divisões por valores próximos de zero ou aumento de erro de arredondamento. Então, o problema genérico envolvendo equações diferenciais ordinárias é reduzido ao estudo de um conjunto de N equações diferenciais acopladas de primeira ordem para a função y_i , $i = 1, 2, \dots, N$, expressa pela seguinte forma geral:

$$\frac{dy_i(x)}{dx} = f_i(x, y_1, \dots, y_N), \quad i = 1, \dots, N \quad (4)$$

A função f_i do lado direito dessa equação são incógnitas e representa as derivadas de $y_i(x)$.

Um problema envolvendo equações diferenciais ordinárias não é completamente definido pelas suas equações, porque depende da natureza das condições iniciais, que é crucial para determinar a melhor maneira de abordá-lo. Condições iniciais são condições algébricas dos valores da função $y_i(x)$ na equação (4) que, em geral,

podem ser satisfeitas em pontos discretos específicos, mas não são mantidas entre eles, ou seja, não são automaticamente preservadas pelas equações diferenciais (NAGLE; SAFF; SNIDER, 2012; PRESS *et al.*, 2007; ZILL, 2016).

Ao invés de serem fornecidas as condições iniciais, podem ser apresentadas as condições de contorno junto com a equação, que podem ser tão simples quanto exigir que certas variáveis tenham determinados valores numéricos, ou tão complicadas como um conjunto de equações algébricas não-lineares em suas variáveis. Normalmente, são as condições de contorno que determinam quais métodos numéricos são viáveis, e são divididas em duas grandes categorias (NAGLE; SAFF; SNIDER, 2012; PRESS *et al.*, 2007; ZILL, 2016):

- a) **problemas de valor inicial** - neste caso, todos os $y_i(x)$ são dados para algum valor inicial x_0 , e deseja-se obter os y_i em algum ponto para algum valor final de x , x_f , ou em alguma lista de pontos discretos;
- b) **problemas de valor no contorno em dois pontos** - aqui as condições de contorno são especificadas em mais de um valor de x . Tipicamente, algumas das condições são especificadas em x_0 e as restantes em x_f .

3.2 Métodos numéricos para problemas de valor inicial

A solução de um problema de valor inicial envolve a ideia implícita de sempre reescrever as derivadas dy_i e dx_i na equação (4) como passos finitos Δy e Δx e multiplicar as equações por Δx . Este procedimento fornece fórmulas algébricas para mudanças nas funções, quando a variável independente x é acrescida de um incremento Δx .

No limite, quando o tamanho do passo for muito pequeno, uma boa aproximação da equação diferencial é obtida e, basicamente, todos os métodos de integração numérica se baseiam neste princípio, diferindo na forma de cálculo da inclinação da curva em cada ponto. Este trabalho se limita a apresentar apenas os detalhes necessários para a compreensão prática dos métodos numéricos utilizados, haja vista que os desenvolvimentos teóricos detalhados se encontram em abundância na literatura (BUTCHER, 2000; HULL *et al.*, 2006).

3.2.1 Métodos de Euler e Euler Melhorado

3.2.1.1 Método de Euler

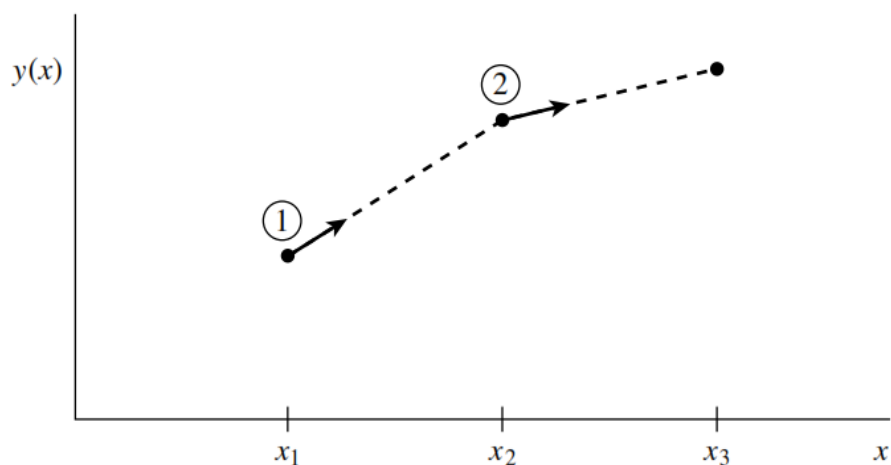
O método explícito mais simples é o método de Euler que consiste em escolher N pontos, de abscissas $x_n = x_0 + n \cdot h$, em que $h = (x_f - x_0)/N$. Usando expansão em série de Taylor da função e truncando no termo da derivada segunda, obtém-se um método com erro de aproximação de primeira ordem, $o(h)$, descrito pela equação (5), e o algoritmo de iteração pela equação (6).

$$y_{n+1} = y_n + h \cdot f(x_n, y_n) \quad (5)$$

$$y_{n+1} = y_n + h \cdot f(x_0 + n \cdot h, y_n) \quad (6)$$

O método de Euler é, conceitualmente, importante porque todos os métodos práticos originam-se da ideia de adicionar pequenos incrementos às funções correspondentes às derivadas (lado direito das equações) multiplicadas pelos tamanhos dos passos (NAGLE; SAFF; SNIDER, 2012; PRESS *et al.*, 2007; ZILL, 2016). Essa fórmula é assimétrica porque avança a solução ao longo de um intervalo h , mas usa informações derivadas apenas no início desse intervalo (FIGURA 1).

Figura 1 - Ilustração esquemática do método de Euler



Fonte: PRESS *et al.*, 2007.

3.2.1.2 Método de Euler Melhorado

O método de Euler melhorado usa a média se a inclinação da curva à esquerda e à direita termina cada passo. Neste caso, o método é formulado como preditor - corretor, com erro de aproximação de segunda ordem, $O(h^2)$, em que os sobrescritos P e C indicam preditor e corretor, respectivamente:

$$y_{n+1}^P = y_n^C + h \cdot f(x_n, y_n^C) \quad (7)$$

$$y_{n+1}^C = y_n^C + \frac{1}{2}h [f(x_n, y_n^C) + f(x_{n+1}, y_{n+1}^P)] \quad (8)$$

3.2.1.3 Métodos de Runge-Kutta Explícitos

Os métodos de Runge-Kutta são métodos de um passo com múltiplos estágios que propagam uma solução em um intervalo combinando as informações de vários passos como no método de Euler (cada um envolvendo uma avaliação das funções f' do lado direito) e, em seguida, usam estas informações para desenvolver uma expansão em série de Taylor, até uma ordem superior (NAGLE; SAFF; SNIDER, 2012; PRESS *et al.*, 2007; ZILL, 2016).

Avanços nos métodos de Runge-Kutta, particularmente no desenvolvimento de métodos de ordem superior, tornaram esses métodos competitivos com os outros métodos, em muitos casos. Os métodos de Runge-Kutta quase sempre têm sucesso e, geralmente, é o método mais rápido quando a avaliação de f'_i não for dispendiosa e o requisito de precisão não for rigoroso ($\approx 10^{-10}$), ou, em geral, quando a precisão exigida for moderada ($\approx 10^{-5}$) (NAGLE; SAFF; SNIDER, 2012; PRESS *et al.*, 2007; ZILL, 2016).

3.2.1.4 Método de Runge-Kutta de Primeira Ordem

O desenvolvimento desse método é trivial, conforme descrito pela equação (9):

$$\begin{aligned}
 k_1 &= h f(x_n, y_n) \\
 y_{n+1} &= y_n + k_1
 \end{aligned}
 \tag{9}$$

3.2.1.5 Método de Runge-Kutta de Segunda Ordem

Os métodos de Runge-Kutta explícitos de dois estágios são dados pela equação (10) e, convenientemente, na forma de um quadro de Butcher (QUADRO 1):

$$\begin{aligned}
 k_1 &= h f(x_n, y_n) \\
 k_2 &= h f(x_n + c_2 h, y_n + a_{21} k_1) \\
 y_{n+1} &= y_n + b_1 k_1 + b_2 k_2
 \end{aligned}
 \tag{10}$$

Em um quadro de Butcher, os componentes do vetor c são os incrementos de x_n e as entradas da matriz A são os multiplicadores das inclinações aproximadas que, depois da multiplicação pelo tamanho do passo h , incrementa y_n . Os componentes do o vetor b são os pesos na combinação dos valores intermediários, k_j .

Quadro 1 - Quadro de Butcher para o método de Runge-Kutta de Segunda ordem

	c	A
k_1	0	0
k_2	c_2	a_{21} 0
y_{n+1}	b^T	b_1 b_2

Fonte: ASHINO; NAGASE; VAILLANCOURT, 2000.

Para atingir a segunda ordem, c , A e b devem ser escolhidos criteriosamente. Por meio dessa escolha, feita usando o Teorema de Taylor para expandir a função, pode-se obter diversas fórmulas de Runge-Kutta de segunda ordem. Algumas delas são clássicas, assumindo outros nomes, e são apresentadas a seguir.

a) Método de Runge-Kutta de dois Estágios

O método de Euler aprimorado pode ser escrito na forma de um método de Runge-Kutta explícito de dois estágios, pela escolha conveniente dos coeficientes, conforme equação (11). Esta equação é idêntica ao método de Heun de segunda ordem.

$$\begin{aligned} k_1 &= h \cdot f(x_n, y_n) \\ k_2 &= h \cdot f(x_n + h, y_n + k_1) \\ y_{n+1} &= y_n + \frac{1}{2}(k_1 + k_2) \end{aligned} \quad (11)$$

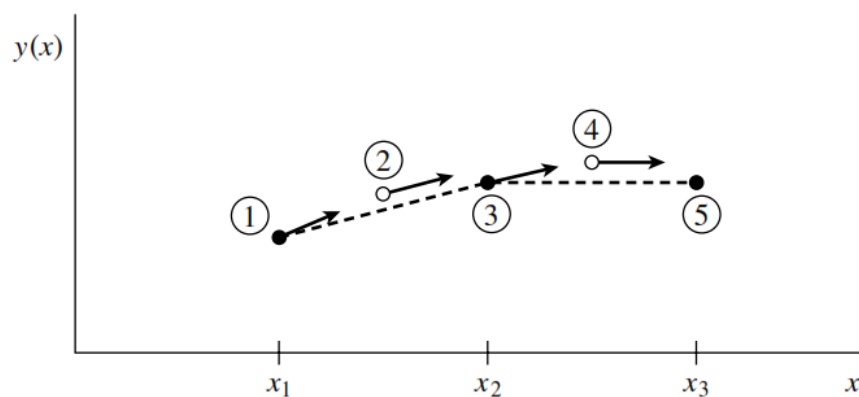
b) Outros métodos de segunda ordem de dois estágios

Método do ponto médio:

A fórmula matemática é descrita pela equação (12) e a Figura 2 contém uma representação esquemática desse método.

$$\begin{aligned} k_1 &= h \cdot f(x_n, y_n) \\ k_2 &= h \cdot f\left(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_1\right) \\ y_{n+1} &= y_n + k_2 \end{aligned} \quad (12)$$

Figura 2 - Ilustração esquemática do método do ponto médio ou Runge-Kutta de segunda ordem



Fonte: PRESS *et al.*, 2007.

Método de Heun:

$$\begin{aligned}
 k_1 &= h \cdot f(x_n, y_n) \\
 k_2 &= h \cdot f\left(x_n + \frac{2}{3}h, y_n + \frac{2}{3}k_1\right) \\
 y_{n+1} &= y_n + \frac{1}{4}k_1 + \frac{3}{4}k_2
 \end{aligned}
 \tag{13}$$

3.2.1.6 Método de Runge-Kutta de Terceira Ordem

Apresentam-se, a seguir, dois métodos comuns de Runge-Kutta de terceira ordem. Um deles é conhecido como fórmula de Heun.

Fórmula de Heun de terceira ordem:

$$\begin{aligned}
 k_1 &= h \cdot f(x_n, y_n) \\
 k_2 &= h \cdot f\left(x_n + \frac{1}{3}h, y_n + \frac{1}{3}k_1\right) \\
 k_3 &= h \cdot f\left(x_n + \frac{2}{3}h, y_n + \frac{2}{3}k_2\right) \\
 y_{n+1} &= y_n + \frac{1}{4}k_1 + \frac{3}{4}k_3
 \end{aligned}
 \tag{14}$$

Fórmula de Runge-Kutta de terceira ordem:

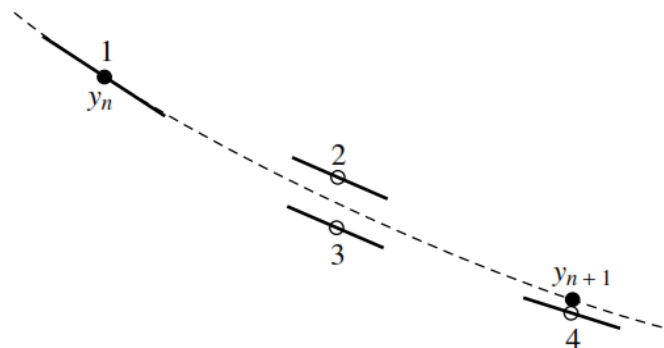
$$\begin{aligned}
 k_1 &= h \cdot f(x_n, y_n) \\
 k_2 &= h \cdot f\left(x_n + \frac{1}{2}h, y_n - k_1\right) \\
 k_3 &= h \cdot f\left(x_n + \frac{2}{3}h, y_n + 2k_2\right) \\
 y_{n+1} &= y_n + \frac{1}{6}(k_1 + 4k_2 + k_3)
 \end{aligned}
 \tag{15}$$

3.2.1.7 Método de Runge-Kutta de Quarta Ordem

O método de Runge-Kutta de quarta ordem (também conhecido como método de Runge-Kutta clássico) é o mais popular dentre os métodos explícitos de um passo, descrito pela equação (15). Sem entrar em detalhes, apresenta-se na Figura 3 uma ilustração esquemática desse método.

$$\begin{aligned}
 k_1 &= h f(x_n, y_n) \\
 k_2 &= h f\left(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_1\right) \\
 k_3 &= h f\left(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_2\right) \\
 k_4 &= h f(x_n + h, y_n + k_3) \\
 y_{n+1} &= y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)
 \end{aligned}
 \tag{16}$$

Figura 3 - Ilustração esquemática do método de Runge-Kutta de quarta ordem



Fonte: PRESS *et al.*, 2007.

3.2.1.8 Pares Integrados de Métodos de Runge-Kutta

Os métodos apresentados nas seções anteriores consideram apenas um tamanho de passo constante, h . Na prática, é vantajoso deixar h variar de modo que h seja considerado maior quando $y(x)$ não variar rapidamente e menor quando $y(x)$ muda rapidamente.

Pares incorporados de métodos Runge-Kutta de ordens p e $p + 1$ têm erros locais e controles de tamanho do passo, monitorando a diferença entre as soluções de ordem superior e inferior, $y_{n+1} - \hat{y}_{n+1}$. Alguns pares incluem um interpolante, usado para interpolar a solução numérica entre os nós da solução numérica e, em alguns casos, para controlar o tamanho do passo.

3.2.1.9 Par de Métodos de Runge-Kutta Integrados de Quatro Estágios

Esse processo consiste em um par de quatro estágios de métodos Runge–Kutta explícitos de ordens 2 e 3 incorporados, com controle de erro. Ele avança de y_n para y_{n+1} com o método de terceira ordem (denominado extrapolação local) e controla o erro local usando a diferença entre as soluções numéricas de terceira e de segunda ordem. O algoritmo de cálculo é realizado em quatro etapas, cujo código está implementado na plataforma Matlab em uma sub-rotina com o nome de ode23. As quatro etapas são:

$$\begin{aligned}
 k_1 &= h \cdot f(x_n, y_n) \\
 k_2 &= h \cdot f(x_n + (1/2)h, y_n + (1/2)k_1) \\
 k_3 &= h \cdot f(x_n + (3/4)h, y_n + (3/4)k_2) \\
 k_4 &= h \cdot f(x_n + h, y_n + (2/9)k_1 + (1/3)k_2 + (4/9)k_3)
 \end{aligned} \tag{17}$$

Os três primeiros estágios produzem a solução na próxima etapa:

$$y_{n+1} = y_n + \frac{2}{9}k_1 + \frac{1}{3}k_2 + \frac{4}{9}k_3 \tag{18}$$

Todos os quatro estágios fornecem a estimativa de erro local:

$$E = -\frac{5}{72}k_1 + \frac{1}{12}k_2 + \frac{1}{9}k_3 - \frac{1}{8}k_4 \tag{19}$$

Este é, realmente, um método de três estágios, pois a primeira etapa em x_{n+1} é igual à última etapa em x_n , ou seja, $k_1^{[n+1]} = k_4^{[n]}$.

O interpolador natural usado na sub-rotina ode23 é o polinômio de Hermite de dois pontos de grau 3, que interpola y_n e $f(x_n, y_n)$ em $x = x_n$, e y_{n+1} e $f(x_{n+1}, y_{n+1})$ em $x = x_{n+1}$.

3.2.1.10 Par Dormand-Prince de Sete Estágios com Interpolador

O par Dormand-Prince de sete estágios, com estimativa de erro local e interpolador é apresentado em um quadro de Butcher (QUADRO 2). A solução é avançada com y_{n+1} de ordem cinco (procedimento denominado extrapolação local), e a solução \hat{y}_{n+1} de

ordem quatro é usada para obter a estimativa de erro local por meio da diferença $y_{n+1} - \hat{y}_{n+1}$. De fato, \hat{y}_{n+1} não é calculado; ao invés disso, os coeficientes na linha $b^T - \hat{b}^T$ no quadro de Butcher são usados para obter a estimativa de erro local. A sexta constante (C_6) no termo de erro de ordem superior é minimizada, mantendo a estabilidade. Seis estágios são necessários para o método de ordem 5 e para o sétimo estágio é necessário ter um interpolador, que é a última linha do Quadro 2 (ASHINO; NAGASE; VAILLANCOURT, 2000).

Quadro 2 - Quadro de Butcher para o par Dormand-Prince de Sete Estágios com Interpolador

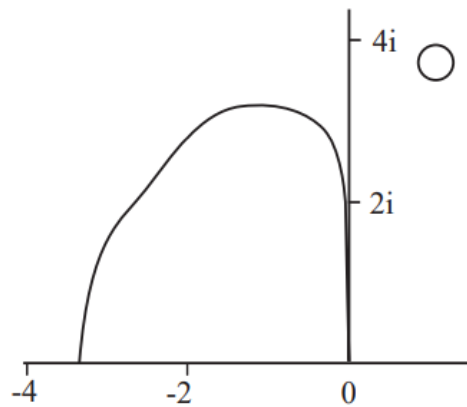
c		A					
k_1	0	0					
k_2	$\frac{1}{5}$	$\frac{1}{5}$	0				
k_3	$\frac{3}{10}$	$\frac{3}{40}$	$\frac{9}{40}$	0			
k_4	$\frac{4}{5}$	$\frac{44}{45}$	$-\frac{56}{15}$	$\frac{32}{9}$	0		
k_5	$\frac{8}{9}$	$\frac{19372}{6561}$	$-\frac{25360}{2187}$	$\frac{64448}{6561}$	$-\frac{212}{729}$	0	
k_6	1	$\frac{9017}{3168}$	$-\frac{355}{33}$	$\frac{46732}{5247}$	$\frac{49}{176}$	$-\frac{5103}{18656}$	0
k_7	1	$\frac{35}{384}$	0	$\frac{500}{1113}$	$\frac{125}{192}$	$-\frac{2187}{6784}$	$\frac{11}{84}$
\hat{y}_{n+1}	\hat{b}^T	$\frac{5179}{57600}$	0	$\frac{7571}{16695}$	$\frac{393}{640}$	$-\frac{92097}{339200}$	$\frac{1}{40}$
y_{n+1}	b^T	$\frac{35}{384}$	0	$\frac{500}{1113}$	$\frac{125}{192}$	$-\frac{2187}{6784}$	0
$b^T - \hat{b}^T$		$\frac{71}{57600}$	0	$-\frac{71}{16695}$	$\frac{71}{1920}$	$-\frac{17253}{339200}$	$-\frac{1}{40}$
$y_{n+0.5}$		$\frac{5783653}{57600000}$	0	$\frac{466123}{1192500}$	$-\frac{41347}{1920000}$	$\frac{16122321}{339200000}$	$\frac{183}{10000}$

Fonte: ASHINO; NAGASE; VAILLANCOURT, 2000.

Este método de sete estágios, na prática, se reduz a seis estágios porque $k_1^{[n+1]} = k_7^{[n]}$, e o vetor linha b^T é igual à sétima linha correspondente a k_7 .

O intervalo de estabilidade absoluta desse método é, aproximadamente, $(-3,3, 0)$, como mostrado na Figura 4. Esse método está implementado no solucionador de EDO's no subprograma ode45 do Matlab.

Figura 4 - Região de estabilidade absoluta do par Dormand-Prince



Fonte: ASHINO; NAGASE; VAILLANCOURT, 2000.

3.2.1.11 Método de Runge-Kutta-Fehlberg de Seis Estágios

O método de Runge-Kutta-Fehlberg de seis estágios, com estimativa de erro local, usa um método de ordem 4 para avançar o valor numérico de y_n para y_{n+1} , e um método de ordem 5 para obter o valor auxiliar \hat{y}_{n+1} , que serve para calcular o erro local por meio da diferença $y_{n+1} - \hat{y}_{n+1}$. O método é apresentado a seguir, sem discussão.

$$\begin{aligned}
 k_1 &= h f(x_n, y_n) \\
 k_2 &= h f(x_n + h/4, y_n + k_1/4) \\
 k_3 &= h f(x_n + 3h/8, y_n + 3k_1/32 + 9k_2/32) \\
 k_4 &= h f(x_n + 12h/13, y_n + 1932k_1/2197 - 7200k_2/2197 + 7296k_3/2197) \\
 k_5 &= h f(x_n + h, y_n + 439k_1/216 - 8k_2 + 3680k_3/513 + 845k_4/4104) \\
 k_6 &= h f(x_n + h/2, y_n - 8k_1/27 + 2k_2 + 3544k_3/2565 + 1859k_4/4104 - 11k_5/40)
 \end{aligned} \tag{20}$$

Método de quarta ordem:

$$y_{n+1} = y_n + \left(\frac{25}{216} k_1 + \frac{1408}{2565} k_3 + \frac{2197}{4104} k_4 - \frac{1}{5} k_5 \right) \tag{21}$$

Método de quinta ordem:

$$\hat{y}_{n+1} = y_n + \left(\frac{16}{35} k_1 + \frac{6556}{12825} k_3 + \frac{28561}{56430} k_4 - \frac{9}{50} k_5 + \frac{2}{55} k_6 \right) \tag{22}$$

Algoritmo:

1. Calcular duas aproximações para y_{n+1} , uma usando a Equação (21), e outra para \hat{y}_{n+1} , usando a Equação (22).
2. Se $|y_{n+1} - \hat{y}_{n+1}| < \epsilon \cdot h$, aceita-se y_{n+1} como aproximação de $y(x_{n+1})$.

Substitui-se h por $q \cdot h$, em que q é igual a:

$$q = \left[\frac{\epsilon \cdot h}{(2 \cdot |\hat{y}_{n+1} - y_{n+1}|)} \right]^{1/4}$$

Retorna-se para o passo 1 para calcular uma aproximação para y_{j+2} .

3. Se $|y_{n+1} - \hat{y}_{n+1}| \geq \epsilon \cdot h$, substitui-se h por $q \cdot h$, em que q é igual a:

$$q = \left[\frac{\epsilon \cdot h}{(2 \cdot |\hat{y}_{n+1} - y_{n+1}|)} \right]^{1/4}$$

Retorna-se para o passo 1 para calcular a próxima aproximação para y_{n+1} .

Pode ser mostrado que o erro de truncamento local para a Equação (21) é, aproximadamente:

$$E = \frac{|\hat{y}_{n+1} - y_{n+1}|}{h} \quad (22)$$

3.2.1.12 Par de Métodos de Runge-Kutta-Verner de Oito Estágios

O par Runge-Kutta-Verner de oito estágios de ordem 5 e 6 é apresentado em um Quadro de Butcher (QUADRO 3). Observe que são necessários 8 estágios para obter a ordem de aproximação 6. O método tenta manter o erro global proporcional a uma tolerância especificada pelo usuário. Isto é eficiente para sistemas não rígidos, nos quais as avaliações da derivada não são computacionalmente dispendiosas e quando a solução não é necessária em um grande número de pontos finamente espaçados.

Quadro 3 - Quadro de Butcher para o par Runge-Kutta-Verner de Oito Estágios.

c		A							
k_1	0	0							
k_2	$\frac{1}{6}$	$\frac{1}{6}$	0						
k_3	$\frac{4}{15}$	$\frac{4}{75}$	$\frac{16}{75}$	0					
k_4	$\frac{2}{3}$	$\frac{5}{6}$	$-\frac{8}{3}$	$\frac{5}{2}$	0				
k_5	$\frac{5}{6}$	$-\frac{165}{64}$	$\frac{55}{6}$	$-\frac{425}{64}$	$\frac{85}{96}$	0			
k_6	1	$\frac{12}{5}$	-8	$\frac{4015}{612}$	$-\frac{11}{36}$	$\frac{88}{255}$	0		
k_7	$\frac{1}{15}$	$-\frac{8263}{15000}$	$\frac{124}{75}$	$-\frac{643}{680}$	$-\frac{81}{250}$	$\frac{2484}{10625}$	0		
k_8	1	$\frac{3501}{1720}$	$-\frac{300}{43}$	$\frac{297275}{52632}$	$-\frac{319}{2322}$	$\frac{24068}{84065}$	0	$\frac{3850}{26703}$	
y_{n+1}	b^T	$\frac{13}{160}$	0	$\frac{2375}{5984}$	$\frac{5}{16}$	$\frac{12}{85}$	$\frac{3}{44}$		
\hat{y}_{n+1}	\hat{b}^T	$\frac{3}{40}$	0	$\frac{875}{2244}$	$\frac{23}{72}$	$\frac{264}{1955}$	0	$\frac{125}{11592}$	$\frac{43}{616}$

Fonte: ASHINO; NAGASE; VAILLANCOURT, 2000.

3.2.2 Controle Adaptativo de Passos nos Métodos de Runge-Kutta

O objetivo desse controle adaptativo do tamanho de passo é obter uma precisão pré-determinada na solução com o mínimo de esforço computacional. Muitos passos pequenos devem ser dados com extremo cuidado quando se percorre determinados intervalos, enquanto alguns passos grandes devem acelerar através de intervalos suaves e sem interrupções. Os ganhos resultantes em eficiência nesse processo não são apenas dezenas de percentagens ou fatores de dois, mas podem ser fatores de dez, cem ou mais. Às vezes, a acurácia pode ser exigida não diretamente na própria solução, mas em uma quantidade conservada relacionada que pode ser monitorada.

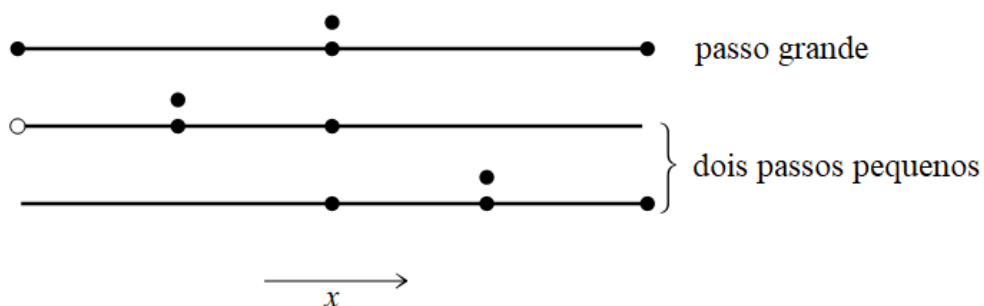
A implementação do controle adaptativo do tamanho do passo requer que o algoritmo de incremento de passo sinalize informações sobre seu desempenho e, mais importante, forneça uma estimativa do seu erro de truncamento. Obviamente, o cálculo dessas informações aumentará a sobrecarga computacional, mas o investimento, geralmente, será bem recompensado.

Especificamente no método de Runge-Kutta de quarta ordem, a técnica mais direta é,

sem dúvida, a duplicação do tamanho dos passos (GEAR, 1971). Cada passo é executado duas vezes, uma vez como um passo completo e, de forma independente, como duas metades do passo (FIGURA 4). A sobrecarga, por exemplo, em termos do número de avaliações do lado direito das equações, para cada uma das três etapas separadas do procedimento de Runge-Kutta, requer quatro avaliações, mas as sequências simples e duplas compartilham um ponto de partida, portanto, um total de 11. Isso deve ser comparado não a quatro, mas a oito (as duas etapas), já que, independentemente do controle de tamanho do passo, atinge-se a precisão do menor passo (metade), com uma sobrecarga indireta multiplicada por um fator de 1,375 e não 2.

Na Figura 4, os pontos onde a derivada é avaliada são mostrados como círculos preenchidos e o círculo aberto representa as mesmas derivadas como no círculo preenchido imediatamente acima dele, portanto, o número total de avaliações é 11 para as duas etapas. A comparação da precisão do passo grande com os dois passos pequenos fornece um critério para ajustar o tamanho do passo na próxima etapa, ou para rejeitar a etapa atual como imprecisa.

Figura 5 - Duplicação do passo como um meio para o controle adaptativo do tamanho do passo no método de Runge-Kutta de quarta ordem



Fonte: PRESS *et al.*, 2007.

Definindo-se a solução exata para um avanço de x para $x + 2h$ por $y(x + 2h)$ e as duas soluções aproximadas por y_1 (um passo de $2h$) e y_2 (dois passos de tamanho h cada um), como o método básico é de quarta ordem, a solução verdadeira e as duas aproximações numéricas são relacionadas por:

$$\begin{aligned} y(x + 2h) &= y_1 + 1(2h)^5\phi + O(h^6) + \dots \\ y(x + 2h) &= y_2 + 2(h)^5\phi + O(h^6) + \dots \end{aligned} \quad (23)$$

A expansão da série Taylor mostra que ϕ é um número cuja ordem de magnitude é $y^{(5)}(x)/5!$. A primeira expressão na equação (7) envolve $(2h)^5$, pois o tamanho do passo é $2h$, enquanto a segunda expressão envolve $2(h^5)$, pois o erro em cada etapa é $h^5\phi$. A diferença entre as duas estimativas numéricas é um indicador conveniente de erro de truncamento:

$$\Delta = y_2 - y_1 \quad (24)$$

É essa diferença que se deve procurar manter no grau desejado de precisão, nem muito grande nem muito pequeno, ajustando-se o tamanho do passo, h . Além disso, pode ocorrer que, ignorando-se os termos de ordem h^6 e superiores, pode-se resolver a equação (23) para melhorar a estimativa numérica da solução verdadeira $y(x + 2h)$:

$$y(x + 2h) = y_2 + \frac{\Delta}{15} + O(h^6) \quad (25)$$

A equação (25) fornece uma estimativa de quinta ordem, uma ordem superior ao método original de quarta ordem (obtida usando extrapolação de Richardson!). Entretanto, não é possível monitorar o erro de truncamento nesse método, confirmando que uma ordem mais alta do erro não garante maior precisão. O método de quinta ordem, raramente, causa transtornos, mas não há uma forma direta de saber se o método está apresentando um bom desempenho, ou não (PRESS *et al.*, 2007). Portanto, deve-se usar Δ , equação (24), como estimativa de erro e tomar como benefício qualquer ganho de precisão adicional derivado da equação (9). Na literatura técnica, o uso de procedimento como esse é chamado de extrapolação local.

A duplicação de etapas foi substituída por um algoritmo de ajuste de passos mais eficiente, com base nas fórmulas de Runge-Kutta incorporadas, originalmente inventadas por Merson e popularizadas no método de Fehlberg. Um fato interessante sobre as fórmulas de Runge-Kutta é que, para ordens de magnitude do erro, M , maiores que quatro, são necessárias mais do que M avaliações de funções. Isso

explica a popularidade do método clássico de quarta ordem, que parece dar o maior retorno possível.

No entanto, Fehlberg descobriu um método de quinta ordem com seis avaliações de funções, no qual outra combinação das seis funções fornece um método de quarta ordem. A diferença entre as duas estimativas de $y(x + 2h)$ pode ser usada como estimativa do erro de truncamento para ajustar o tamanho do passo. Desde a fórmula original de Fehlberg, muitas outras fórmulas incorporadas de Runge-Kutta foram encontradas.

O método de quinta ordem requer seis avaliações de funções, o de ordem seis requer sete, o de ordem sete requer nove, e o de ordem oito requer onze. Sabe-se que para ordem $M \geq 8$ são necessárias, pelo menos, $M + 3$ avaliações. O método de ordem mais alta, construído explicitamente, já conhecido é de ordem 10, com 17 avaliações. O cálculo dos coeficientes desses métodos de alta ordem é muito complicado.

3.2.3 Método de Bulirsch-Stoer

Antes de prosseguir é preciso discutir o método do ponto médio modificado, que tem sua aplicação mais importante no Método de Bulirsch-Stoer. Outra técnica que pode ser combinada com o método de Bulirsch-Stoer é a Extrapolação de Richardson. Uma breve discussão desses métodos é apresentada nas duas próximas seções.

3.2.3.1 Método do Ponto Médio Modificado

O método do ponto médio modificado consiste em avançar um vetor de variáveis dependentes, $y(x)$, de um ponto cuja abscissa é x para $x + H$ por uma sequência de n subpassos de tamanho h cada:

$$h = \frac{H}{n} \quad (26)$$

O número de avaliações do lado direito exigidas pelo método do ponto médio modificado é $n + 1$. As fórmulas para esse método são:

$$\begin{aligned}
z_0 &= y(x) \\
z_1 &= z_0 + h \cdot f(x, z_0) \\
z_{m+1} &= z_{m-1} + 2h \cdot f(x + m \cdot h, z_m); & \text{para } m = 1, 2, \dots, n-1 & \quad (27) \\
y(x + H) &\approx y_n \equiv \frac{1}{2} [z_n + z_{n-1} + h \cdot f(x + H, z_n)]
\end{aligned}$$

As variáveis z são aproximações intermediárias que marcham ao longo dos passos h , enquanto y_n é a aproximação final de $y(x + H)$. Esse é um método basicamente de *diferença centralizada* ou *ponto médio*, exceto no primeiro e no último ponto, que conferem ao método o qualificador *modificado*.

O método do ponto médio modificado é de segunda ordem com a vantagem de exigir, assintoticamente, para n grande, apenas uma avaliação da derivada por passo h , em vez das duas exigidas pelo método de Runge-Kutta de segunda ordem. Talvez, haja aplicações nas quais a simplicidade do método de Runge-Kutta de segunda ordem ou do ponto médio, equação (10), facilmente codificada em linha em algum programa, possa recomendá-lo. Em geral, o uso do método do ponto médio modificado, por si só, é superado pelo método de Runge-Kutta com controle adaptativo de tamanho do passo incorporado.

A utilidade do método do ponto médio modificado para a técnica de Bulirsch-Stoer deriva de um resultado detalhado da equação (27), devido ao trabalho de Gragg (STOER; BULIRSCH, 1980). Ocorre que o erro da equação (27), expresso como uma série de potências de h , contém apenas potências pares de h :

$$y_n - y(x + H) = \sum_{i=1}^{\infty} \alpha_i \cdot h^{2i} \quad (28)$$

O valor de H é mantido constante, mas h muda quando n varia na equação (26). A importância dessas séries de potências pares é que, se forem praticados os truques habituais de combinar etapas e eliminar termos de erro de ordem superior, podem-se obter mais duas ordens de precisão de cada vez. Por exemplo, suponha-se que n seja par e $y_{n/2}$ seja o resultado da aplicação das equações (26) e (29), com metade do número de passos, $n = n/2$. Então estima-se:

$$y(x + H) \approx \frac{4y_n - y_{n/2}}{3} \quad (29)$$

Essa fórmula tem precisão de quarta ordem, a mesma que o método de Runge-Kutta de quarta ordem, mas requer apenas cerca de 1,5 avaliações das derivadas por passo h , em vez das 4 avaliações de Runge-Kutta.

3.2.3.2 Extrapolação de Richardson

A extrapolação de Richardson usa a poderosa ideia de extrapolar um resultado computado para o valor que teria sido obtido se o tamanho do passo tivesse sido muito menor do que realmente era. Em particular, a extrapolação para passo de tamanho zero é o objetivo desejado. O primeiro integrador prático de EDOs que implementou essa ideia foi desenvolvido por Bulirsch e Stoer e, portanto, os métodos de extrapolação são, frequentemente, chamados de Métodos de Bulirsch-Stoer (PRESS *et al.*, 2007).

3.2.3.3 O Método de Bulirsch-Stoer

A sequência de tentativas separadas para cruzar o intervalo H é feita com o aumento dos valores de n , o número de subpassos. Bulirsch e Stoer propuseram, originalmente, a seguinte sequência (PRESS *et al.*, 2007):

$$n = 2, 4, 6, 8, 12, 16, 24, 32, 48, 64, 96, \dots, [n_j = 2n_{j-2}], \dots \quad (30)$$

Deuffhard (1983, 1985) sugere que a seguinte sequência é mais eficiente:

$$n = 2, 4, 6, 8, 10, 12, 14, \dots, [n_j = 2j], \dots \quad (31)$$

Para cada etapa, não se sabe, antecipadamente, até onde irá essa sequência. Depois de cada n sucessivo, uma extrapolação polinomial é tentada. Essa extrapolação fornece valores extrapolados e estimativas de erro. Se os valores dos erros não forem satisfatórios, aumenta-se o valor de n . Se forem satisfatórios, prossegue-se para o próximo passo e começa-se novamente com $n = 2$.

Para cada etapa, não se sabe, antecipadamente, até onde irá essa sequência. Depois de cada n sucessivo, uma extrapolação polinomial é tentada. Essa extrapolação fornece valores extrapolados e estimativas de erro. Se os valores dos erros não forem satisfatórios, aumenta-se o valor de n . Se forem satisfatórios, prossegue-se para o próximo passo e começa-se novamente com $n = 2$.

Certamente, existe algum limite superior, além do qual conclui-se que haja algum obstáculo no caminho ao longo do intervalo H , de modo que devesse reduzir H em vez de apenas subdividi-lo mais finamente. Além disso, a perda de acurácia ocorre se for escolhido uma subdivisão muito fina. Nas implementações de algoritmo usando esse método, geralmente, considera n igual a oito. O oitavo valor na sequência expressa pela equação (31) é 16, então esse é o número máximo de subdivisões de H permitido (PRESS *et al.*, 2007).

O controle de erros é reforçado, como no método de Runge-Kutta, monitorando-se a consistência interna e adaptando o tamanho do passo para corresponder a um limite prescrito do erro de truncamento local. Considere a tabela expressa na equação (32), em que P representa uma “filha” e x representa os seus “pais”.

$$\begin{array}{rccccccc}
 x_0: & y_0 = P_0 & & & & & & \\
 & & & P_{01} & & & & \\
 x_1: & y_1 = P_1 & & & P_{012} & & & \\
 & & & P_{12} & & P_{0123} & & (32) \\
 x_2: & y_2 = P_2 & & & P_{123} & & & \\
 & & & P_{23} & & & & \\
 x_3: & y_3 = P_3 & & & & & &
 \end{array}$$

O algoritmo de Neville é uma maneira recursiva de preencher os números no quadro expresso pela equação (32), uma coluna de cada vez, da esquerda para a direita. É baseado no relacionamento entre uma “filha” P e seus “pais”, conforme equação (33), que essa recorrência funciona, porque os dois “pais” já concordam nos pontos $x_{i+1} \cdots x_{i+m-1}$.

$$P_{i(i+1)\dots(i+m)} = \frac{(x - x_{i+m})P_{i(i+1)\dots(i+m-1)} + (x_i - x)P_{(i+1)(i+2)\dots(i+m)}}{(x_i - x_{i+m})} \quad (33)$$

Voltando ao problema abordado, cada novo resultado da sequência de integrações modificadas do ponto médio permite que um quadro como o da equação (32) seja estendido por um conjunto adicional de diagonais, escrito como uma matriz triangular inferior:

$$\begin{array}{ccc} T_{00} & & \\ T_{10} & T_{11} & \\ T_{20} & T_{21} & T_{22} \\ \dots & \dots & \dots \end{array} \quad (34)$$

Aqui, $T_{k0} = y_k$, em que y_k é $y(x_n + H)$ calculado com o tamanho do passo $h_k = H/n_k$. Substituindo-se P por T , $x_i = h_i^2$, e $x = 0$ na equação (33), o algoritmo de Neville pode ser reescrito como:

$$T_{k,j+1} = T_{kj} + \frac{T_{kj} - T_{k-1,j}}{(n_k/n_{k-j})^2 - 1}, \quad \forall j = 0, 1, \dots, k-1 \quad (35)$$

A cada novo tamanho de passo h_i inicia-se uma nova linha na tabela e, em seguida, a extrapolação polinomial preenche o restante da linha. Cada novo elemento na tabela vem dos dois elementos mais próximos da coluna anterior. Os elementos na mesma coluna têm a mesma ordem, e T_{kk} , o último elemento de cada linha, é a aproximação de mais alta ordem com esse tamanho de passo. A diferença entre os dois últimos elementos consecutivos é tomada como estimativa de erro (conservadora). Essa estimativa de erro pode ser usada para ajustar o tamanho do passo. Uma boa estratégia foi, originalmente, proposta por Deuffhard (1983,1985). Uma versão modificada é apresentada a seguir (HAIRER; NØRSETT; WANNER, 1993; LAPIDUS; SEINFELD, 1971; PRESS *et al.*, 2007; STOER; BULIRSCH, 2002).

3.2.3.4 Controle do Tamanho de Passo para o Método de Bulirsch-Stoer

Os elementos na tabela mostrada na equação (32) são vetores correspondentes ao vetor y de variáveis dependentes. Consequentemente, pode-se definir:

$$\Delta_k = T_{kk} - T_{k,k-1} \quad (36)$$

Essa norma é a mesma norma escalonada usada nos métodos de Runge-Kutta, definida pela equação (24). O controle de erros é imposto exigindo que $\Delta_k \leq 1$. Agora T_{kk} é de ordem $2k + 2$ e $T_{k,k-1}$ é de ordem $2k$, que é, a ordem de Δ_k . Dessa forma, tem-se:

$$\Delta_k \sim H^{2k+1} \quad (37)$$

Portanto, uma estimativa simples de um novo tamanho escalonado H_k , para obter convergência em uma coluna fixa k , pode ser dada pela equação (38), em que S_1 e S_2 são fatores de segurança menores que um:

$$H_k = H \cdot S_1 \left(\frac{S_2}{\Delta_k} \right)^{1/(2k+1)} \quad (38)$$

Para determinar em qual coluna k deve-se buscar a convergência, pode-se comparar o trabalho necessário para diferentes valores de k . Suponha-se que A_k seja o trabalho para obter a linha k da tabela de extrapolação. Suponha-se que o trabalho seja dominado pelo custo de avaliar as funções que definem os lados direito das equações diferenciais. Para n_k subdivisões em H , o número de avaliações de funções pode ser encontrado a partir da fórmula de recorrência:

$$\begin{aligned} A_0 &= n_0 + 1 \\ A_{k+1} &= A_k + n_{k+1} \end{aligned} \quad (39)$$

O trabalho por unidade de passo para obter a coluna k é, portanto:

$$W_k = \frac{A_k}{H_k} \quad (40)$$

O índice ideal de colunas é o que minimiza W_k . A estratégia é definir um alvo k para a próxima etapa e escolher o tamanho do passo usando a equação (38), para tentar obter convergência (ou seja, $\Delta_k \leq 1$) para o valor de k na etapa seguinte.

Na prática, calcula-se a tabela de extrapolação da equação (34) linha por linha, mas apenas testa convergência em uma janela para ordem entre $k - 1$ e $k + 1$. A justificativa para a janela de ordem é que, se a convergência parece ocorrer antes da

coluna $k - 1$, ela é, frequentemente, espúria, resultante de uma estimativa de erro fortuitamente pequena na extrapolação. Todavia, se for preciso ir além de $k + 1$ para obter convergência, o modelo de comportamento de convergência local, obviamente, não é muito bom e será preciso reduzir o tamanho do passo e restabelecê-lo.

3.2.3.5 Observações finais

Cada um dos dois métodos de interpolação (ponto médio modificado e extrapolação de Richardson) apresentados pode ser organizado para monitorar a consistência interna. Isso permite que erros numéricos, que são inevitavelmente introduzidos na solução, sejam controlados pela mudança automática (adaptativa) do tamanho do passo fundamental. Sempre é recomendado que o controle adaptativo do tamanho do passo seja implementado. Em geral, todos os tipos de métodos podem ser aplicados a qualquer problema de valor inicial. Cada um vem com seu próprio conjunto de *prós* e *contras*, que devem ser entendidos antes de serem usados (PRESS *et al.*, 2007).

4 METODOLOGIA

Os métodos de Runge-Kutta de um passo de ordem 1, 2, 3, 4 e 5 foram testados na solução de problemas de valor inicial, juntamente com os métodos mais precisos de Bulirsch-Stoer e Dormand-Prince. Este último faz parte das sub-rotinas já implementadas no Matlab, com a denominação de ode45. O trabalho foi realizado em três etapas, como apresentadas a seguir.

4.1 Implementação e teste de uma rotina em Matlab que incorpora os métodos de Runge-Kutta de ordem 1, 2, 3, 4 e 5

4.1.1 Implementação da Rotina FixedRK

A sub-rotina foi implementada na linguagem do Matlab e incorpora os métodos explícitos de Runge-Kutta de um passo de ordem 1, 2, 3, 4 e 5, portanto, não incorpora a função de adaptação de passo.

4.1.2 Teste da Sub-Rotina

Para testar a sub-rotina FixedRK foi desenvolvido um programa em Matlab, para resolver o seguinte problema de valor inicial:

$$\frac{dy}{dt} + a y = b; \quad y(0) = 5 \quad (41)$$

Solução analítica:

$$y(t) = \frac{(5 a - b) \cdot \exp(-a t)}{a} \quad (42)$$

As soluções aproximadas foram comparadas com a solução analítica, para fins de teste e validação da sub-rotina.

4.2 Implementação e teste da sub-rotina BulirschStoer junto com FixedRK e ode45

4.2.1 Implementação da Rotina BulirschStoer

A sub-rotina foi implementada na linguagem do Matlab e incorpora o método de Bulirsch-Stoer, que contém uma estratégia de adaptação do passo de integração. Por isso apresenta, geralmente, maior exatidão nas soluções numéricas.

4.2.2 Teste da Sub-Rotina BulirschStoer

Para testar a sub-rotina Bulirsch-Stoer foi desenvolvido um programa em Matlab, para resolver o mesmo problema de valor inicial definido pelas equações (41) e (42), da subseção 3.1. Foram incluídas no programa as sub-rotinas FixedRK e ode45 como referências para comparação dos resultados.

4.3 Solução de dois problemas usando os métodos Dormand-Prince, Runge-Kutta de ordem 5 e Bulirsch-Stoer

O método de Dormand-Prince pertence às classes de variações dos métodos de Runge-Kutta, possui adaptação de passo e está implementado no Matlab pela sub-rotina ode45. Os métodos de Runge-Kutta e Bulirsch-Stoer foram implementados nesse trabalho.

5 RESULTADOS E DISCUSSÃO

5.1 Implementação e teste de uma rotina em Matlab que implementa os métodos de Runge-Kutta de ordem 1, 2, 3, 4 e 5

5.1.1 Implementação da Rotina FixedRK

Essa sub-rotina implementa os métodos explícitos de Runge-Kutta de um passo, portanto, não incorpora a função de adaptação de passo. O código em Matlab encontra-se no Apêndice A.

5.1.2 Teste da Sub-Rotina

Assumindo-se $a = 4$ e $b = 15a$, a equação (41) foi resolvida usando-se os métodos de Runge-Kutta, Dormand-Prince e Bulirsch-Stoer, para $0 \leq t \leq 1$. Os resultados são apresentados a seguir, para vários valores do passo de integração e ordem de aproximação dos métodos de Runge-Kutta. Uma amostra dos resultados encontra-se no Apêndice C.

Os resultados do tempo computacional e do erro máximo para resolver o problema, para cada ordem de aproximação do método, encontra-se na Tabela 1, para valores do passo de integração $h = 0,1$ e $h = 0,01$. As representações gráficas estão nas Figuras 6 a 15, para $h = 0,1$, e 16 a 25, para $h = 0,01$, respectivamente.

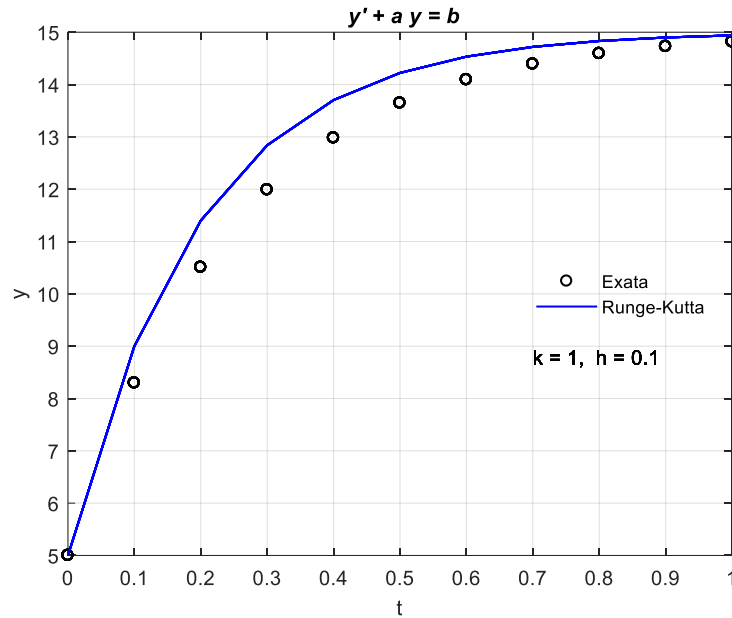
Tabela 1 - Tempo computacional e erro máximo na solução da Equação (24) pelos métodos de Runge-Kutta explícitos de um passo.

Ordem	Tempo [s]		Erro Máximo [%]	
	$h = 0,1$	$h = 0,01$	$h = 0,1$	$h = 0,01$
Exata	0	0	0	0
k = 1	0,00110	0,00163	$8,5021 \times 10^0$	$7,0828 \times 10^{-1}$
k = 2	0,00109	0,00175	$1,2441 \times 10^0$	$9,5344 \times 10^{-3}$
k = 3	0,00111	0,00410	$1,2581 \times 10^0$	$9,5527 \times 10^{-5}$
k = 4	0,00065	0,00178	$1,0203 \times 10^{-2}$	$7,6523 \times 10^{-7}$
k = 5	0,00062	0,00184	$4,3513 \times 10^{-4}$	$3,3287 \times 10^{-9}$

Fonte: Elaborado pelo autor, 2021.

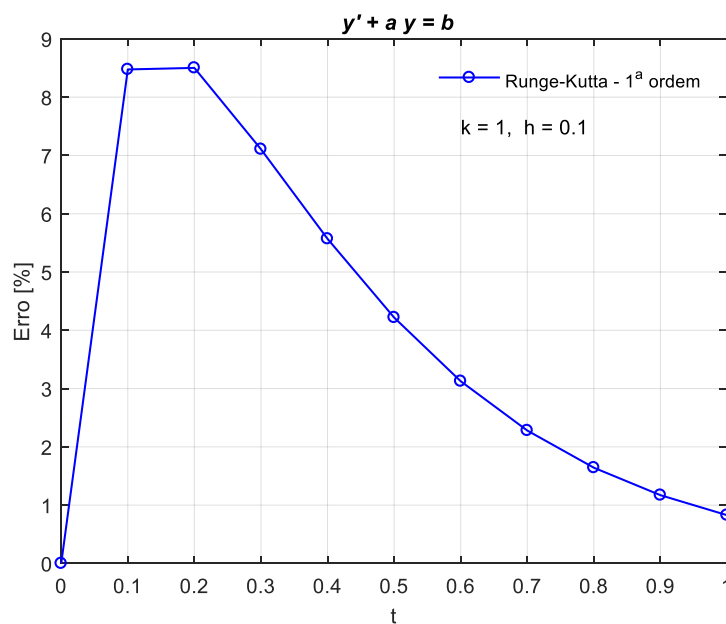
Tamanho do passo: $h = 0,1$

Figura 6 - Solução analítica e numérica da equação diferencial, usando $dy/dt + a \cdot y = b$ do método de Runge-Kutta de primeira ordem, $k = 1$, com passo fixo, $h = 0,1$



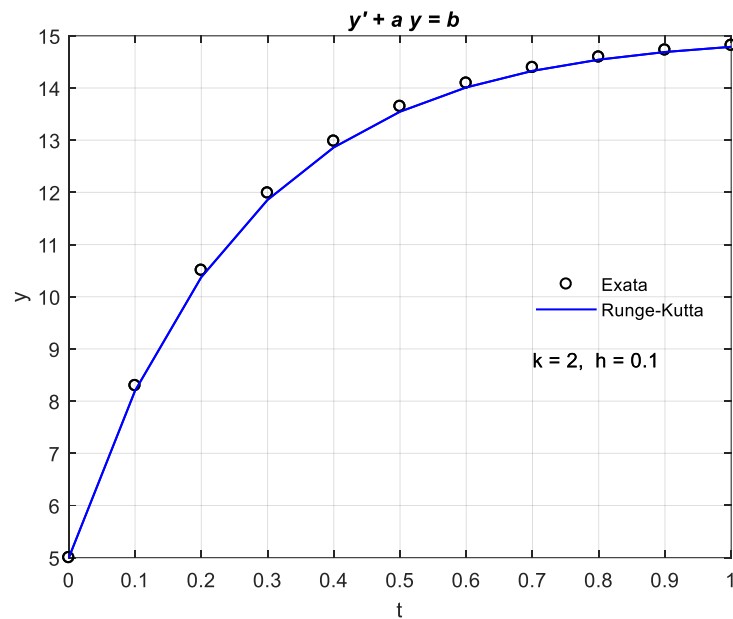
Fonte: Elaborado pelo autor, 2021.

Figura 7 - Variação do erro relativo (%) da equação diferencial $dy/dt + a \cdot y = b$, usando o método de Runge-Kutta de primeira ordem, $k = 1$, com passo fixo, $h = 0,1$



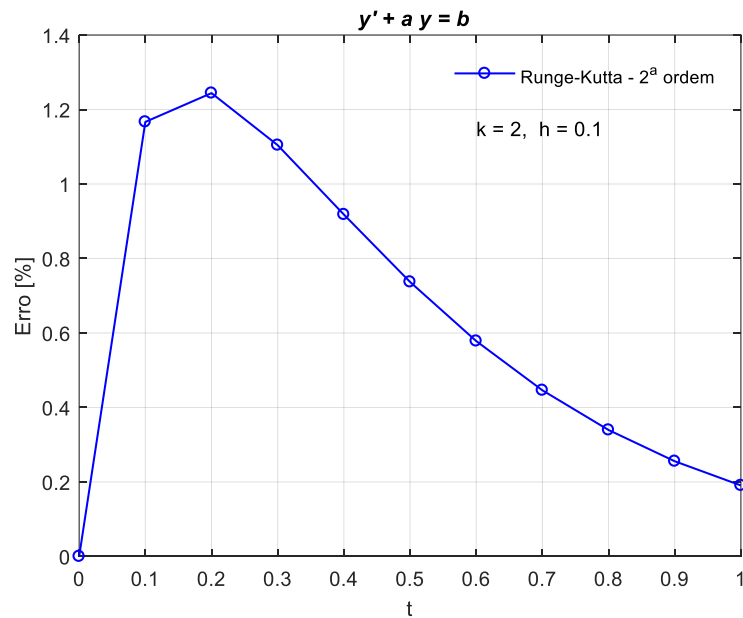
Fonte: Elaborado pelo autor, 2021.

Figura 8 - Solução analítica e numérica da equação diferencial $dy/dt + a.y = b$, usando o método de Runge-Kutta de segunda ordem, $k = 2$, com passo fixo, $h = 0,1$



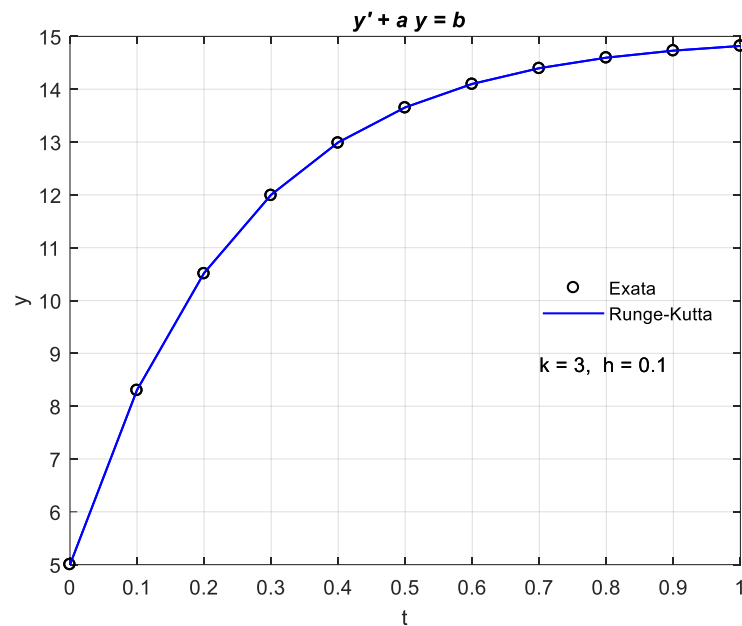
Fonte: Elaborado pelo autor, 2021.

Figura 9 - Variação do erro relativo (%) da equação diferencial $dy/dt + a.y = b$, usando o método de Runge-Kutta de segunda ordem, $k = 2$, com passo fixo, $h = 0,1$



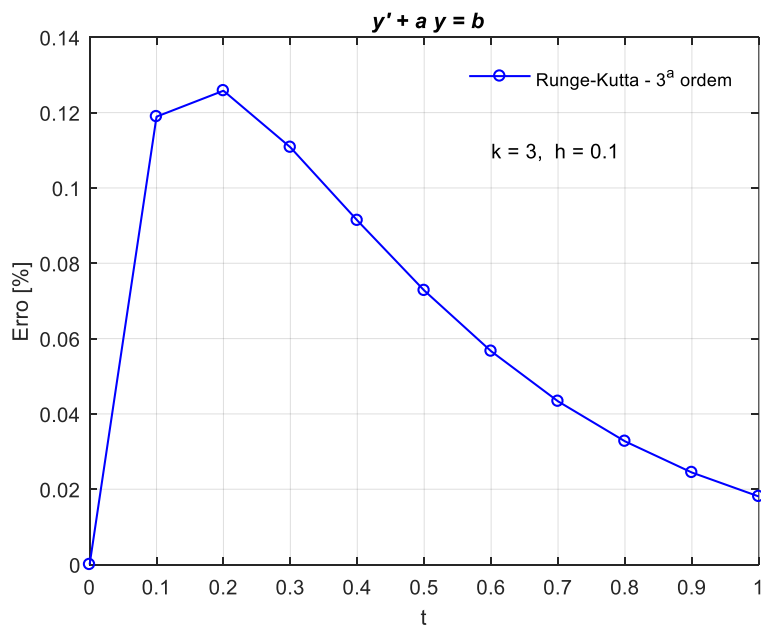
Fonte: Elaborado pelo autor, 2021.

Figura 10 - Solução analítica e numérica da equação diferencial $dy/dt + a.y = b$, usando o método de Runge-Kutta de terceira ordem, $k = 3$, com passo fixo, $h = 0,1$



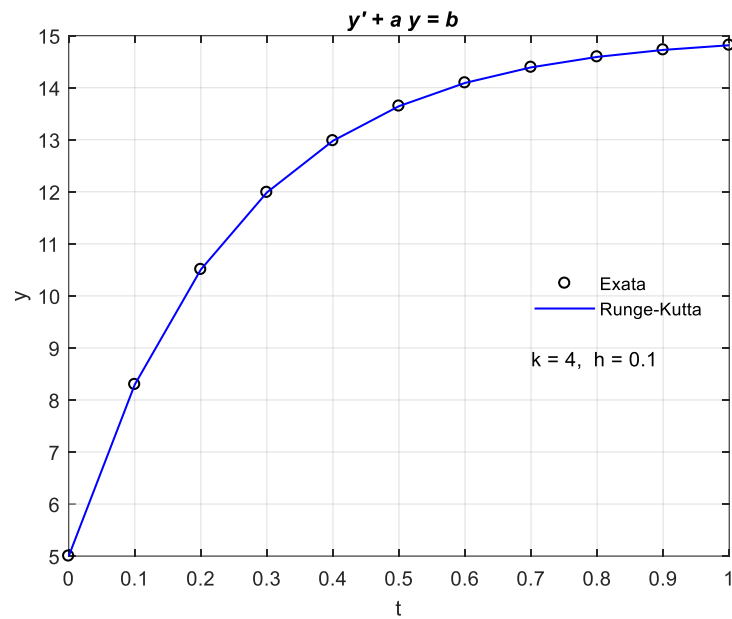
Fonte: Elaborado pelo autor, 2021.

Figura 11 - Variação do erro relativo (%) da equação diferencial $dy/dt + a.y = b$, usando o método de Runge-Kutta de segunda ordem, $k = 3$, com passo fixo, $h = 0,1$



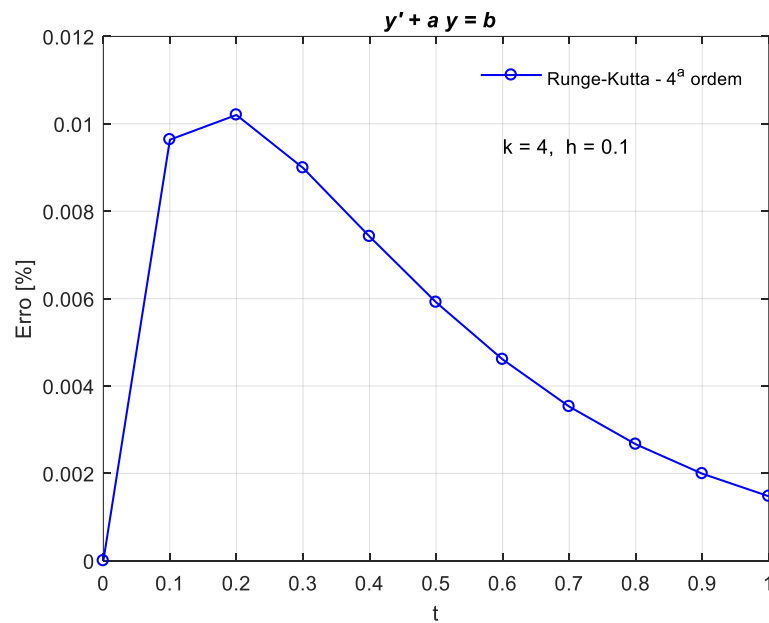
Fonte: Elaborado pelo autor, 2021.

Figura 12 - Solução analítica e numérica da equação diferencial $dy/dt + a.y = b$, usando o método de Runge-Kutta de quarta ordem, $k = 4$, com passo fixo, $h = 0,1$



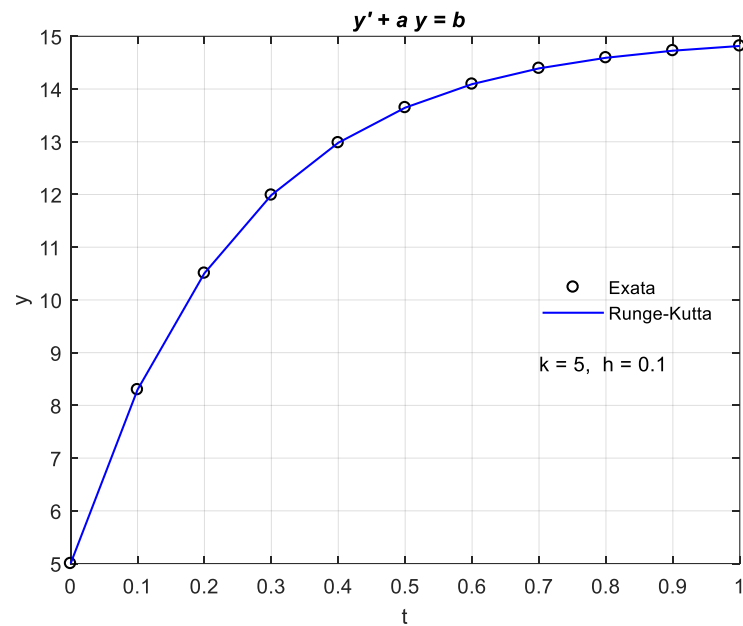
Fonte: Elaborado pelo autor, 2021.

Figura 13 - Variação do erro relativo (%) da equação diferencial $dy/dt + a.y = b$, usando o método de Runge-Kutta de segunda ordem, $k = 4$, com passo fixo, $h = 0,1$



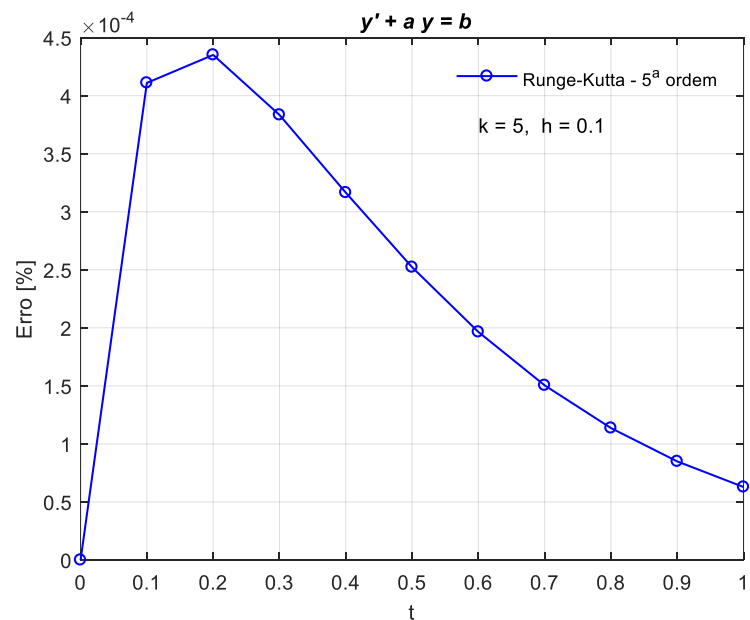
Fonte: Elaborado pelo autor, 2021.

Figura 14 - Solução analítica e numérica da equação diferencial $dy/dt + a.y = b$, usando o método de Runge-Kutta de quinta ordem, $k = 5$, com passo fixo, $h = 0,1$



Fonte: Elaborado pelo autor, 2021.

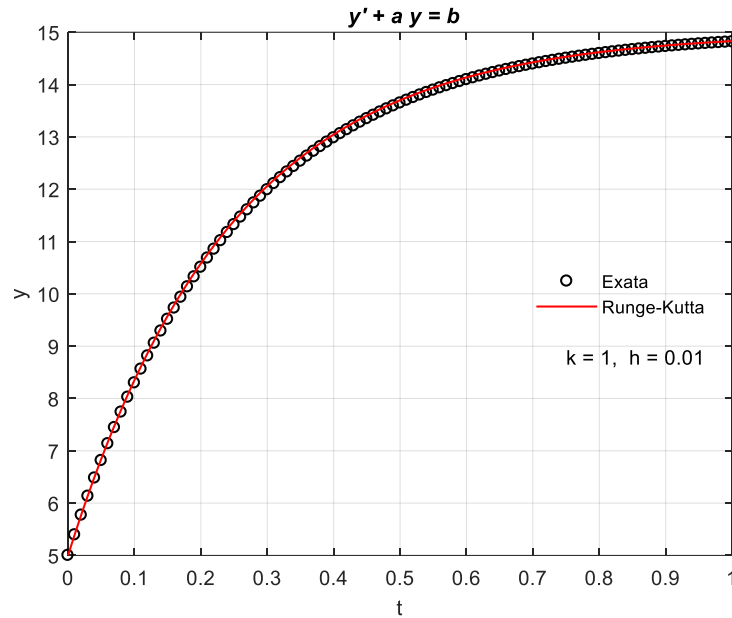
Figura 15 - Variação do erro relativo (%) da equação diferencial $dy/dt + a.y = b$, usando o método de Runge-Kutta de segunda ordem, $k = 5$, com passo fixo, $h = 0,1$



Fonte: Elaborado pelo autor, 2021.

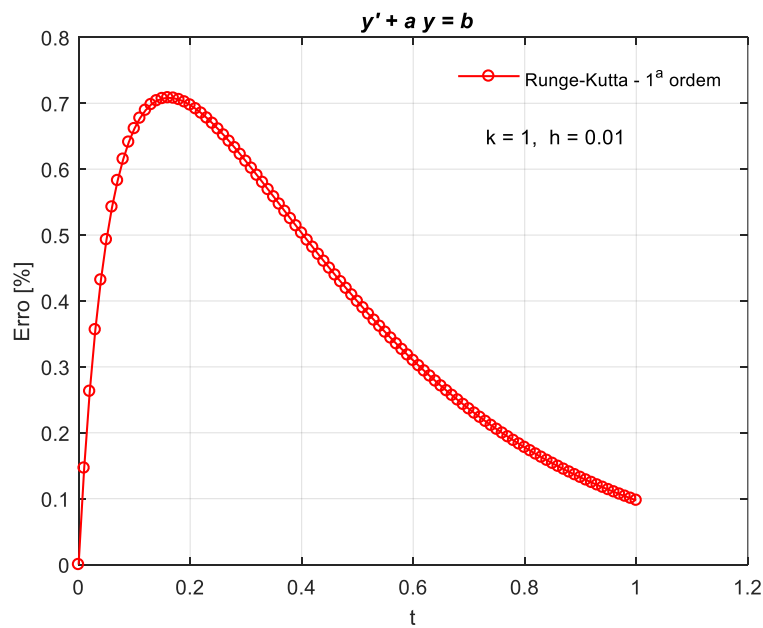
Tamanho do passo: $h = 0,01$

Figura 16 - Solução analítica e numérica da equação diferencial $dy/dt + a \cdot y = b$, usando o método de Runge-Kutta de primeira ordem, $k = 1$, com passo fixo, $h = 0,01$



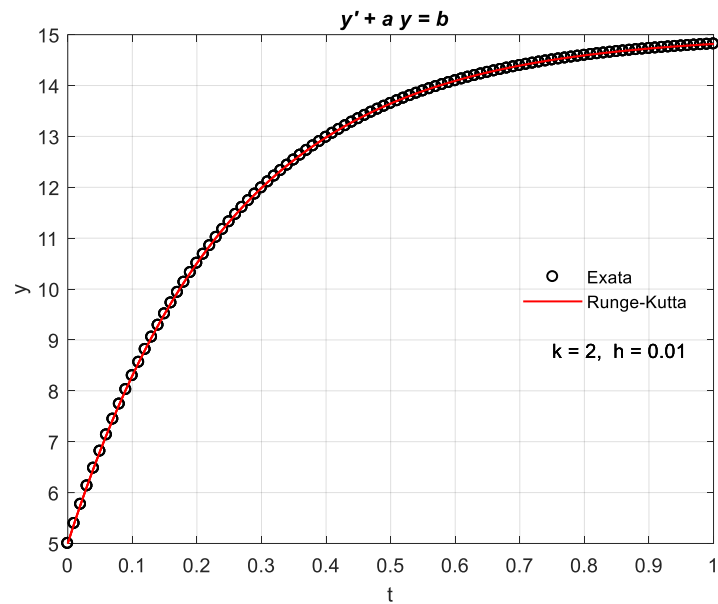
Fonte: Elaborado pelo autor, 2021.

Figura 17 - Variação do erro relativo (%) da equação diferencial $dy/dt + a \cdot y = b$, usando o método de Runge-Kutta de segunda ordem, $k = 1$, com passo fixo, $h = 0,01$.



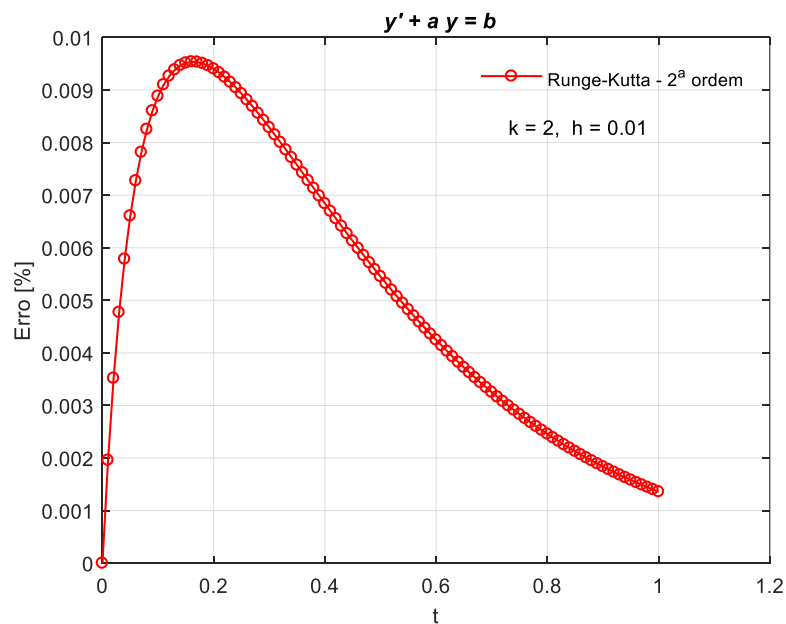
Fonte: Elaborado pelo autor, 2021.

Figura 18 - Solução analítica e numérica da equação diferencial $dy/dt + a \cdot y = b$, usando o método de Runge-Kutta de segunda ordem, $k = 2$, com passo fixo, $h = 0,01$



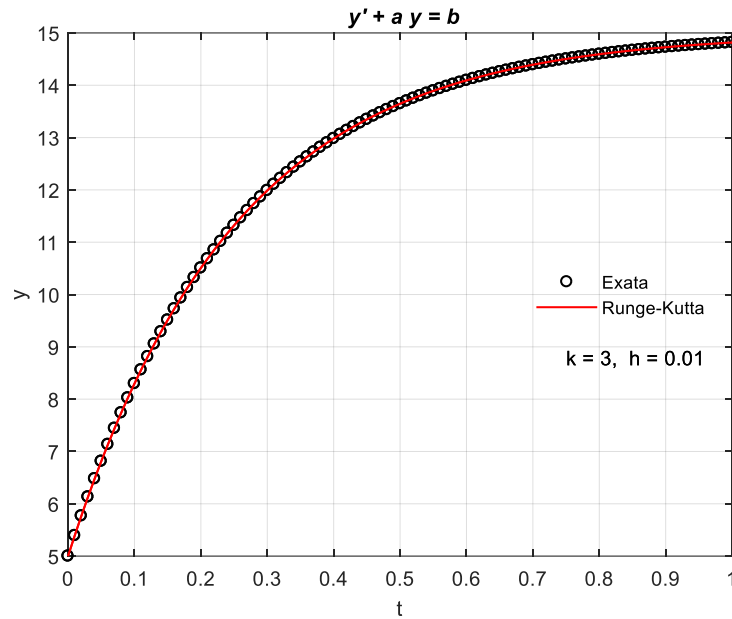
Fonte: Elaborado pelo autor, 2021.

Figura 19 - Figura 18 - Variação do erro relativo (%) da equação diferencial $dy/dt + a \cdot y = b$, usando o método de Runge-Kutta de segunda ordem, $k = 2$, com passo fixo, $h = 0,01$



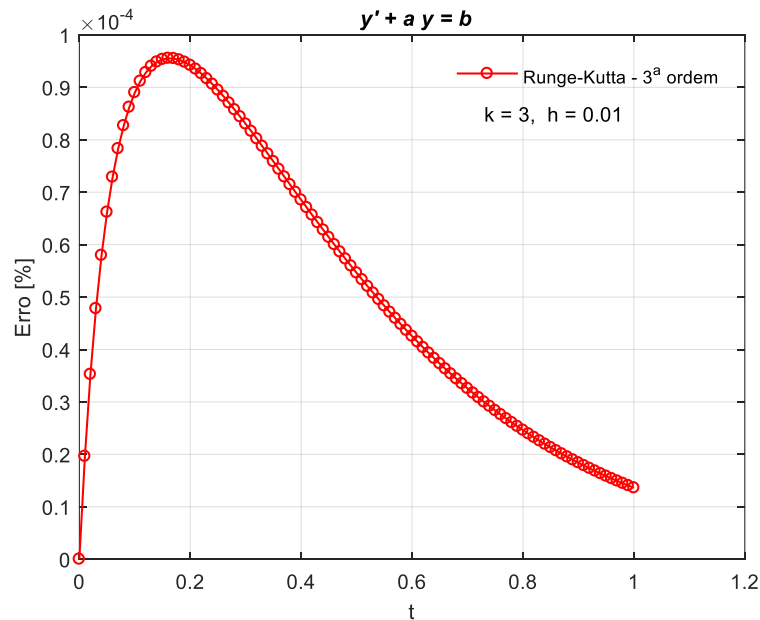
Fonte: Elaborado pelo autor, 2021.

Figura 20 - Solução analítica e numérica da equação diferencial $dy/dt + a.y = b$, usando o método de Runge-Kutta de terceira ordem, $k = 3$, com passo fixo, $h = 0,01$



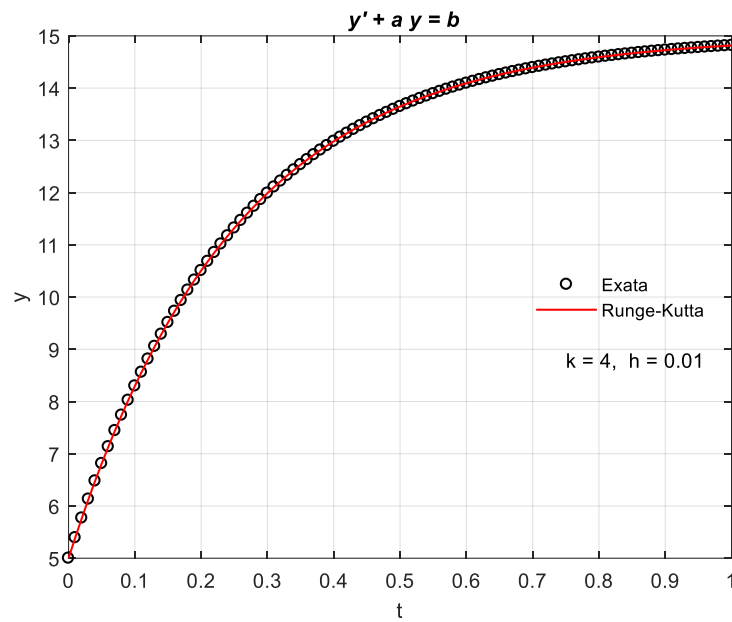
Fonte: Elaborado pelo autor, 2021.

Figura 21 - Variação do erro relativo (%) da equação diferencial $dy/dt + a.y = b$, usando o método de Runge-Kutta de segunda ordem, $k = 3$, com passo fixo, $h = 0,01$



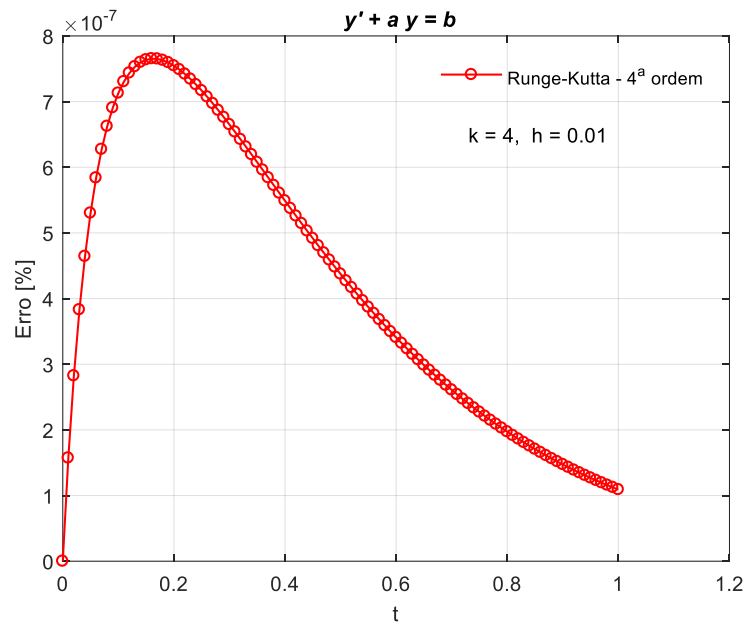
Fonte: Elaborado pelo autor, 2021.

Figura 22 - Solução analítica e numérica da equação diferencial $dy/dt + a \cdot y = b$, usando o método de Runge-Kutta de quarta ordem, $k = 4$, com passo fixo, $h = 0,01$



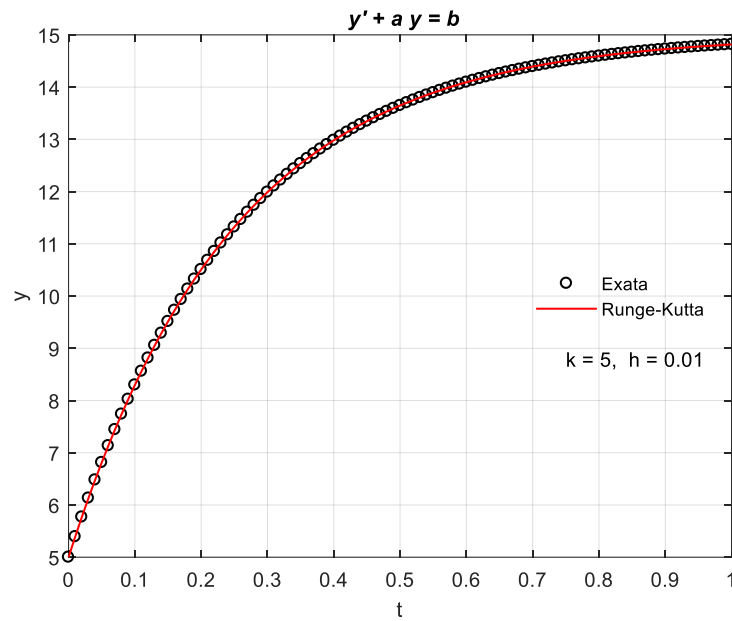
Fonte: Elaborado pelo autor, 2021.

Figura 23 - Variação do erro relativo (%) da equação diferencial $dy/dt + a \cdot y = b$, usando o método de Runge-Kutta de segunda ordem, $k = 4$, com passo fixo, $h = 0,01$



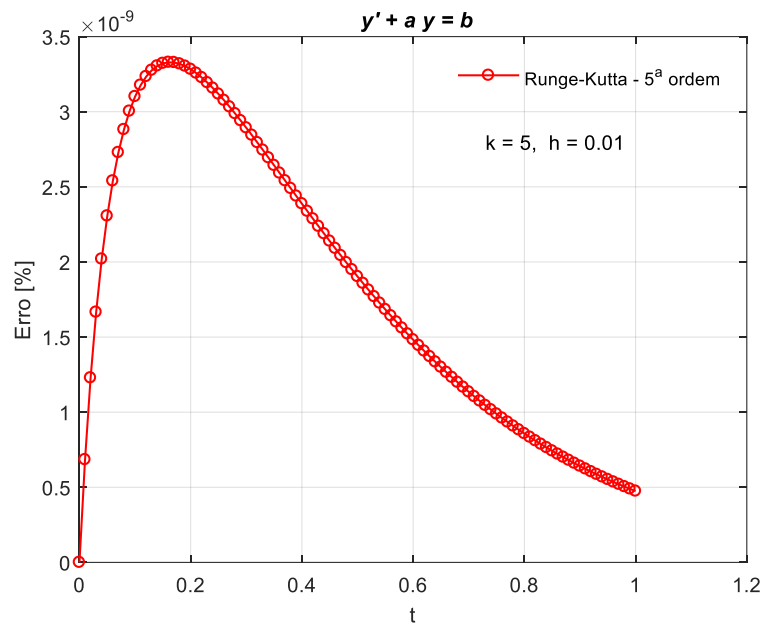
Fonte: Elaborado pelo autor, 2021.

Figura 24 - Solução analítica e numérica da equação diferencial $dy/dt + a \cdot y = b$, usando o método de Runge-Kutta de quinta ordem, $k=5$, com passo fixo, $h = 0,01$



Fonte: Elaborado pelo autor, 2021.

Figura 25 - Variação do erro relativo (%) da equação diferencial $dy/dt + a \cdot y = b$, usando o método de Runge-Kutta de segunda ordem, $k = 5$, com passo fixo, $h = 0,01$



Fonte: Elaborado pelo autor, 2021.

Observa-se na Tabela 1 e nas Figuras 6 a 25 que a precisão das soluções numéricas nos métodos de Runge-Kutta de passo fixo depende do tamanho do passo e, em certo grau, do tipo de problema que está sendo resolvido. Observa-se que, principalmente, para os métodos de Runge-Kutta de primeira e de segunda ordem (métodos de Euler aperfeiçoado), a precisão dos resultados é altamente sensível ao tamanho do passo (FIGURAS 6 a 9). Portanto, esses métodos podem ser usados para resolver problemas simples, desde que o tamanho do passo de integração seja adequado.

O método de Runge-Kutta de terceira ordem deve ser utilizado com cautela, ou combinado com outras técnicas, para resolver problemas mais complexos e fornecer resultados satisfatórios, com tamanho do passo adequado. Entretanto, o método de Runge-Kutta clássico de quarta ordem é o mais utilizado para a solução de problemas que não apresentam rigidez na sua solução, devido sua simplicidade e boa precisão. É interessante notar que o comportamento do erro relativo permanece o mesmo, independentemente da ordem do método, variando sistematicamente de magnitude em função do tamanho do passo (TABELA 1), com valores máximos atingindo a ordem de magnitude de 10^{-7} e 10^{-9} , respectivamente, para os métodos de ordem 4 e 5, para tamanho do passo de $h = 0,01$ (FIGURAS 23 e 25).

5.2 Implementação e teste da sub-rotina BulirschStoer junto com FixedRK e ode45

5.2.1 Implementação da Rotina BulirschStoer

O método de Bulirsch-Stoer foi implementado na sub-rotina BulirschStoer, na linguagem do Matlab, e incorpora uma estratégia de adaptação do passo de integração por meio do Método do Ponto Médio Modificado. Por isso apresenta, geralmente, maior exatidão nas soluções numéricas. O código de programação dessa sub-rotina encontra-se no Apêndice B.

5.2.2 Teste da Sub-Rotina BulirschStoer

Assumindo-se $a = 4$ e $b = 15 a$, a equação (41) foi resolvida usando-se o método de Bulirsch-Stoer, para $0 \leq t \leq 1$ e passo de integração $h = 0,1$ e $h = 0,01$. Uma amostra

dos resultados encontra-se no Apêndice D. Para possibilitar a comparação dos resultados, as sub-rotinas FixedRK e ode45 foram inclusas no programa. Os resultados do tempo computacional e do erro máximo para resolver o problema, encontram-se na Tabela 2. Os resultados na forma gráfica estão apresentados nas Figuras 26 a 36.

Tabela 2 - Tempo computacional e erro máximo na solução da Equação (24) usando os métodos de Runge-Kutta de quinta ordem, Dormand-Prince e Bulirsch-Stoer.

Método	Tempo [s]		Erro Máximo [%]	
	$h = 0,1$	$h = 0,01$	$h = 0,1$	$h = 0,01$
Exata	0	0	0	0
Dormand-Prince	0,00113	0,00179	$2,9939 \times 10^{-4}$	$7,0125 \times 10^{-4}$
RK: k = 5	0,00076	0,00151	$4,3513 \times 10^{-4}$	$3,3287 \times 10^{-9}$
Bulirsch-Stoer	0,00203	0,00856	$4,5377 \times 10^{-13}$	$3,2463 \times 10^{-13}$

Fonte: Elaborado pelo autor, 2021.

A sub-rotina FixedRK foi implementada e testada na Seção 3.1 para os métodos de Runge-Kutta de ordem 1, 2, 3, 4 e 5, resolvendo um problema de valor inicial de primeira ordem, comparando-se as soluções numéricas com a analítica. Foram utilizados dois tamanhos de passo de integração, $h = 0,1$ e $h = 0,01$. Observou-se que todos os métodos forneceram resultados coerentes, mostrando que a precisão das soluções numéricas depende do tamanho de passo e aumenta com o decréscimo deste, demonstrado pela diminuição do erro de truncamento.

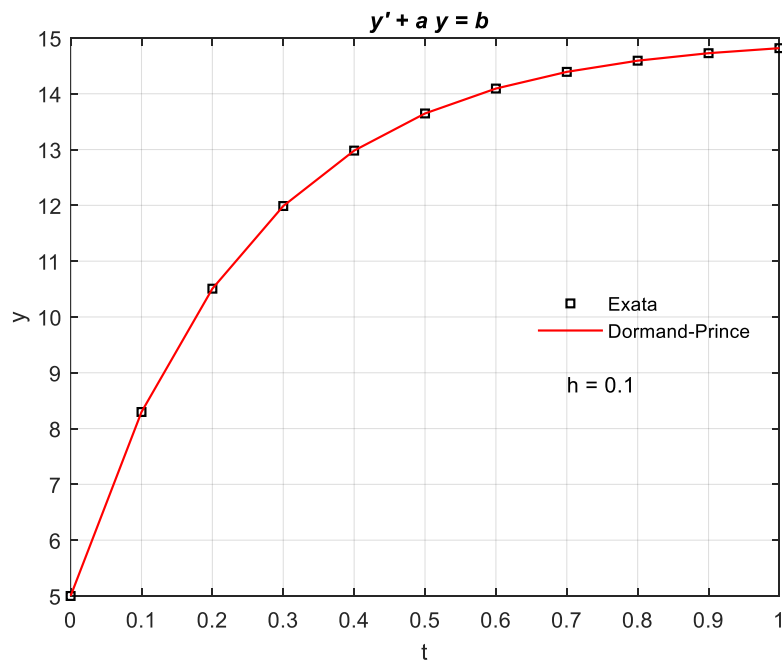
Para comparação dos resultados obtidos por meio de métodos com ordem de aproximação mais alta, utilizaram-se na Seção 3.2 três métodos: Runge-Kutta de ordem 5, Bulirsch-Stoer e Dormand-Prince. Os três métodos forneceram ótimas aproximações da solução, mas a precisão do método de Runge-Kutta diminui com o aumento do tamanho do passo, pois os valores do erro de aproximação não são conservadores, como mostrado na Tabela 2.

Nos métodos de Bulirsch-Stoer e de Dormand-Prince os valores dos erros são conservadores, ou seja, independentemente do tamanho do passo, eles mantêm os erros dentro de uma faixa especificada. Todavia, a ordem de magnitude dos erros no

método de Dormand-Prince é menor do que no de Bulirsch-Stoer. Entretanto, o método de Runge-Kutta de quinta ordem pode fornecer erros com ordem de magnitude menor que o de Dormand-Prince, desde que o tamanho do passo seja suficientemente pequeno, o que é uma vantagem quando o problema não apresenta característica de rigidez, pois o método é mais simples, fácil de programar, embora o esforço computacional seja discutível.

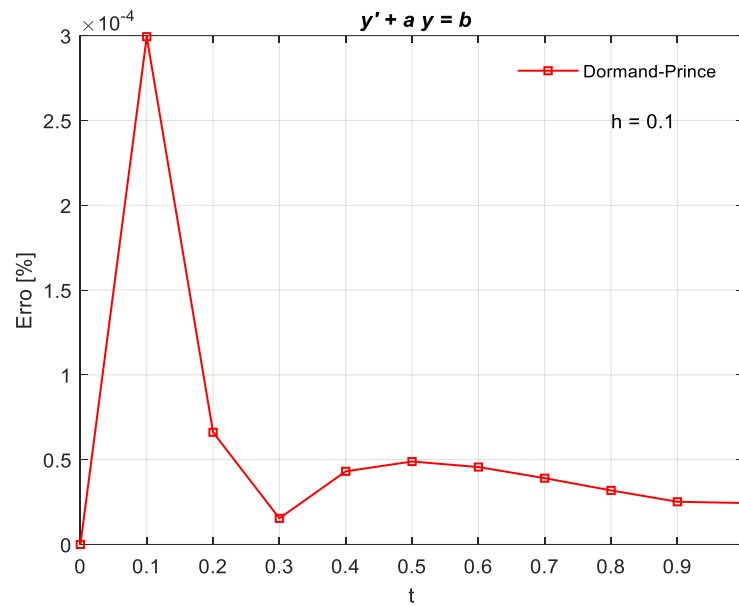
Métodos de Dormand-Prince, Runge-Kutta e Bulirsch-Stoer: $h = 0,1$

Figura 26 - Solução analítica e numérica da equação diferencial $dy/dt + a.y = b$, usando o método de Dormand-Prince, com passo $h = 0,1$



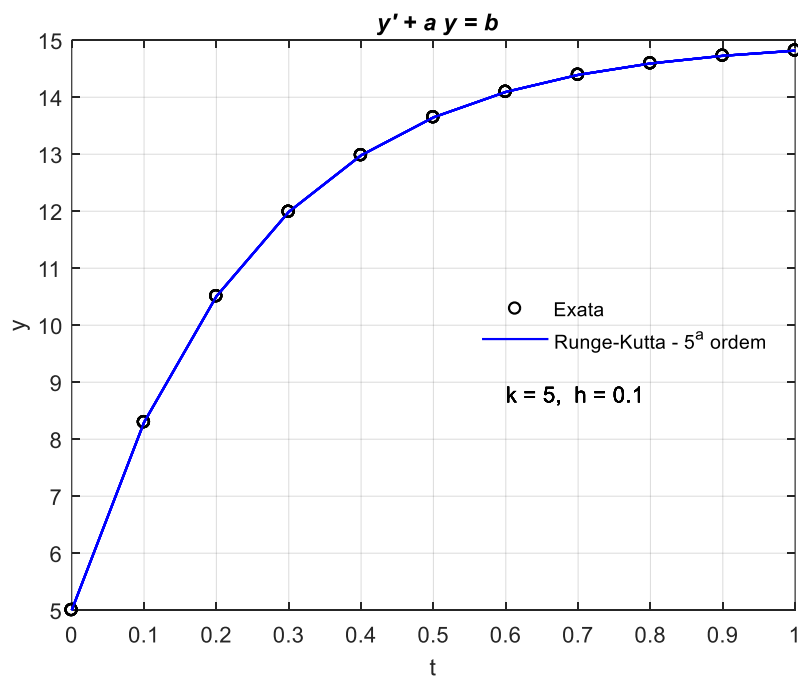
Fonte: Elaborado pelo autor, 2021.

Figura 27 - Variação do erro relativo (%) da equação diferencial $dy/dt + a y = b$, usando o método de Dormand-Prince, com passo $h = 0,1$



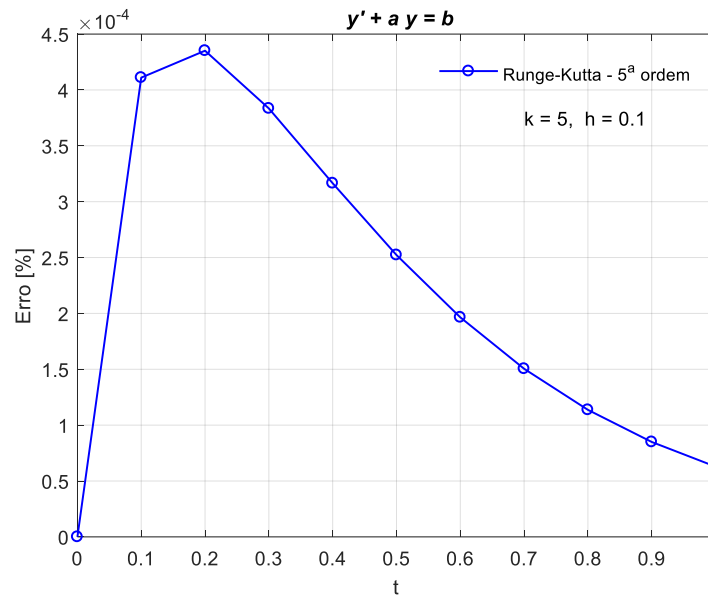
Fonte: Elaborado pelo autor, 2021.

Figura 28- Solução analítica e numérica da equação diferencial $y' + a y = b$, usando o método de Runge-Kutta de quinta ordem, $k = 5$, com passo fixo, $h = 0,1$.



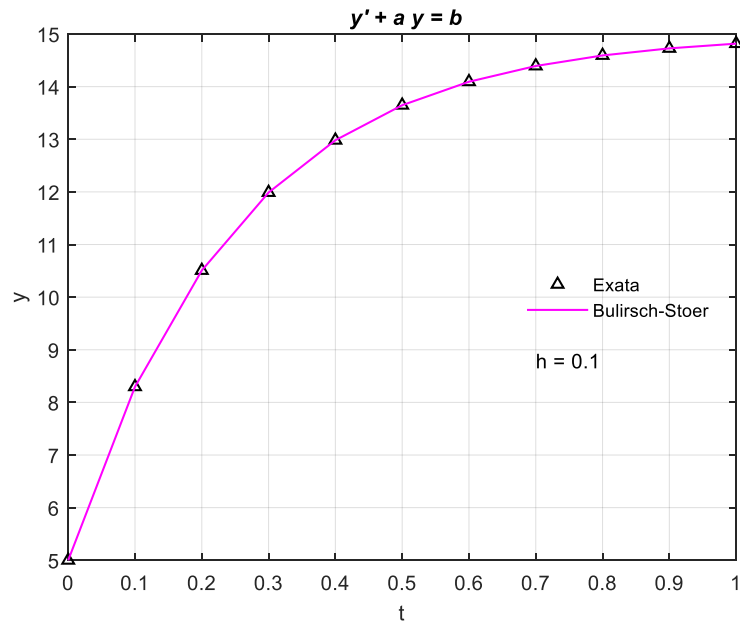
Fonte: Elaborado pelo autor, 2021.

Figura 29 - Variação do erro relativo (%) da equação diferencial $dy/dt + a.y = b$, usando o método de Runge-Kutta de quinta ordem, $k = 5$, com passo fixo, $h = 0,1$



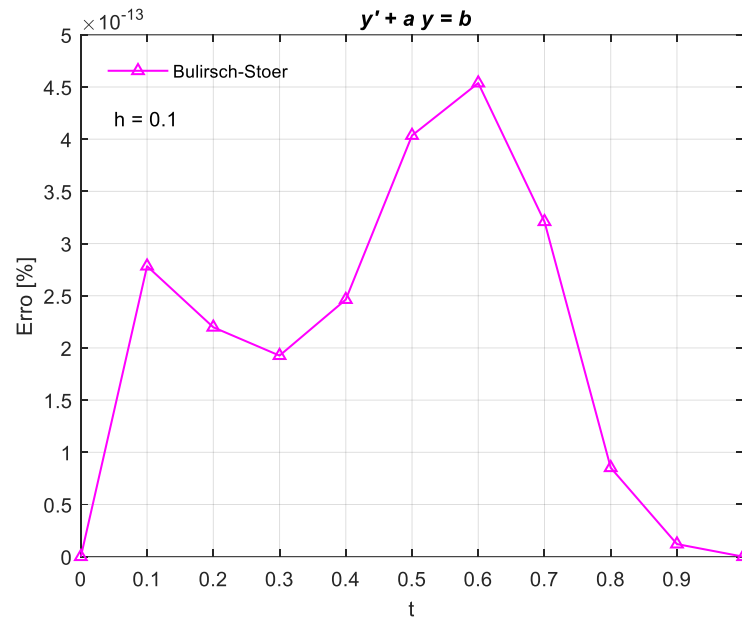
Fonte: Elaborado pelo autor, 2021.

Figura 30 - Solução analítica e numérica da equação diferencial $dy/dt + a.y = b$, usando o método de Bulirsch-Stoer, com passo $h = 0,1$



Fonte: Elaborado pelo autor, 2021.

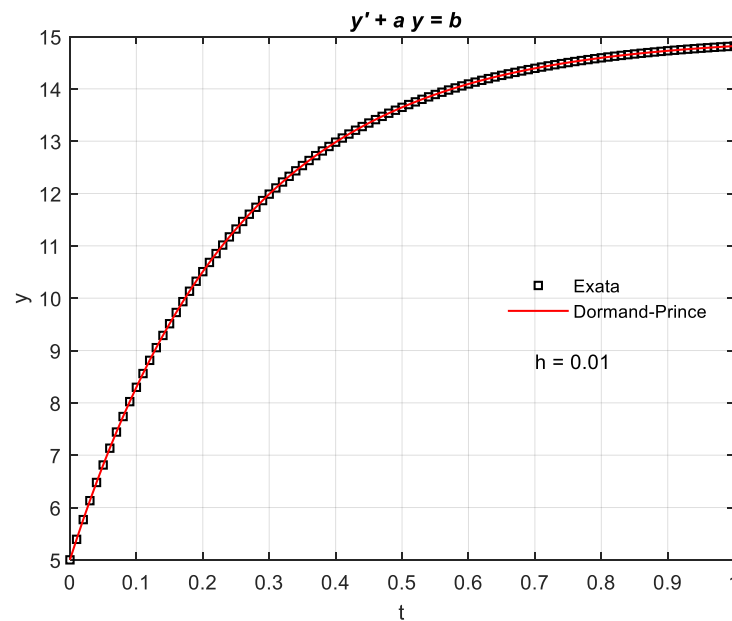
Figura 31 - Variação do erro relativo (%) da equação diferencial $dy/dt + a.y = b$, usando o método de Bulirsch-Stoer, com passo $h = 0,1$



Fonte: Elaborado pelo autor, 2021.

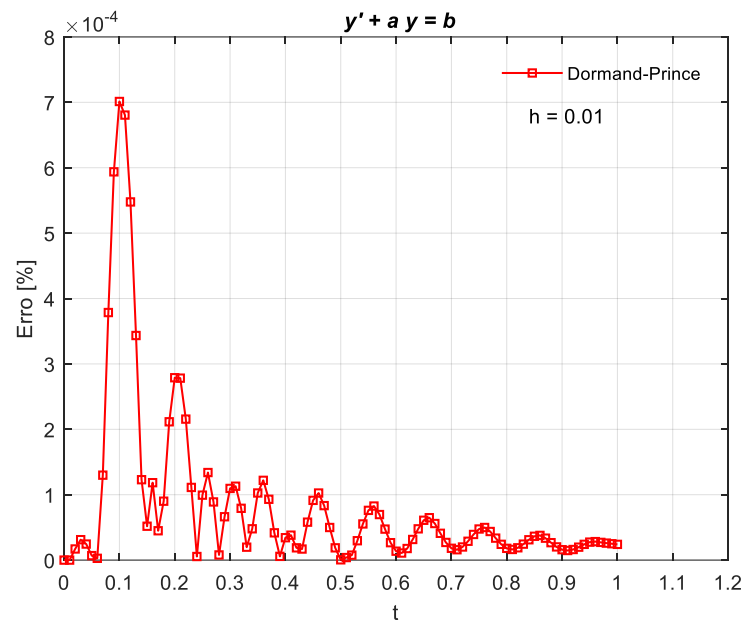
Métodos de Dormand-Prince, Runge-Kutta e Bulirsch-Stoer: $h = 0,01$

Figura 32 - Solução analítica e numérica da equação diferencial $dy/dt + a.y = b$, usando o método de Dormand-Prince, com passo $h = 0,01$



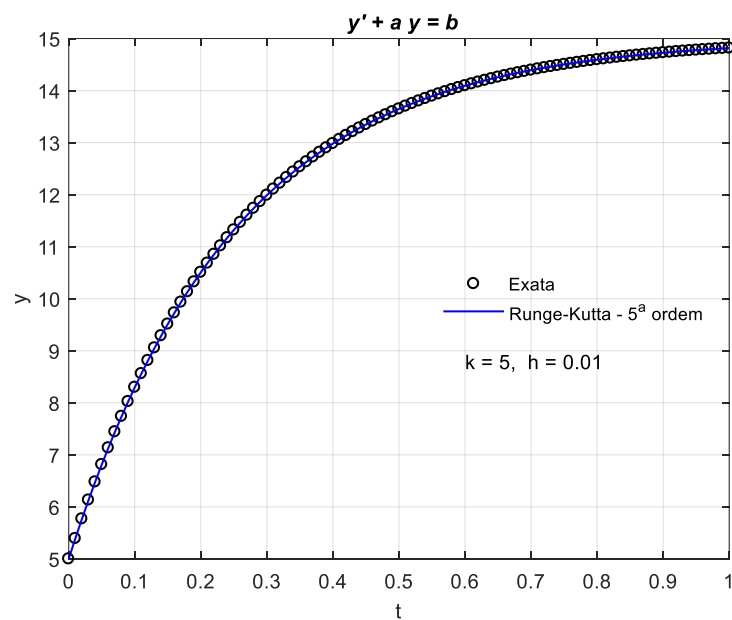
Fonte: Elaborado pelo autor, 2021.

Figura 33 - Variação do erro relativo (%) da equação diferencial $dy/dt + a \cdot y = b$, usando o método de Dormand-Prince, com passo $h = 0,01$



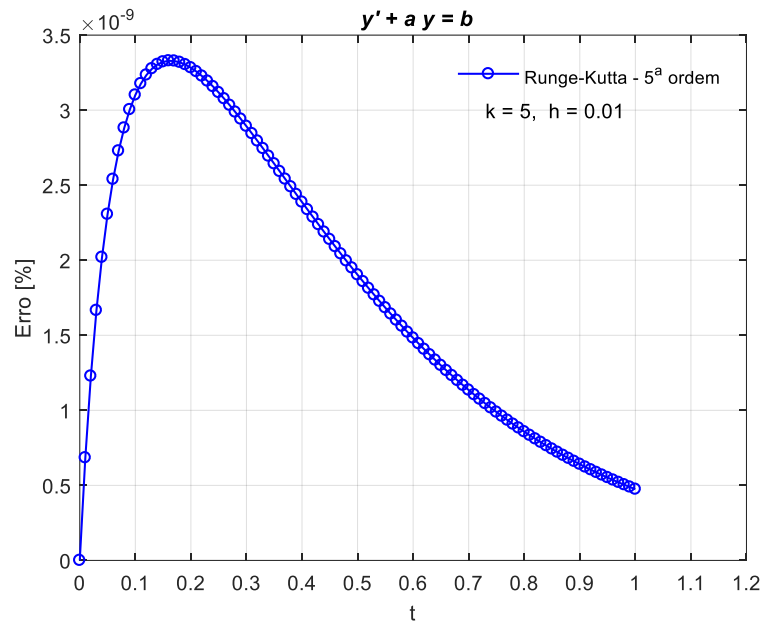
Fonte: Elaborado pelo autor, 2021.

Figura 34 - Solução analítica e numérica da equação diferencial $dy/dt + a \cdot y = b$, usando o método de Runge-Kutta de quinta ordem, $k = 5$, com passo fixo, $h = 0,01$



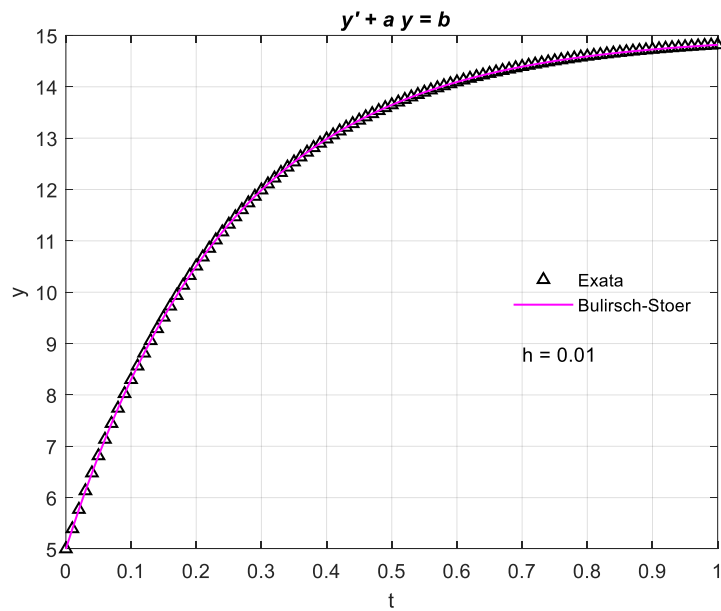
Fonte: Elaborado pelo autor, 2021.

Figura 35 - Variação do erro relativo (%) da equação diferencial $dy/dt + a.y = b$, usando o método de Runge-Kutta de quinta ordem, $k = 5$, com passo fixo, $h = 0,01$



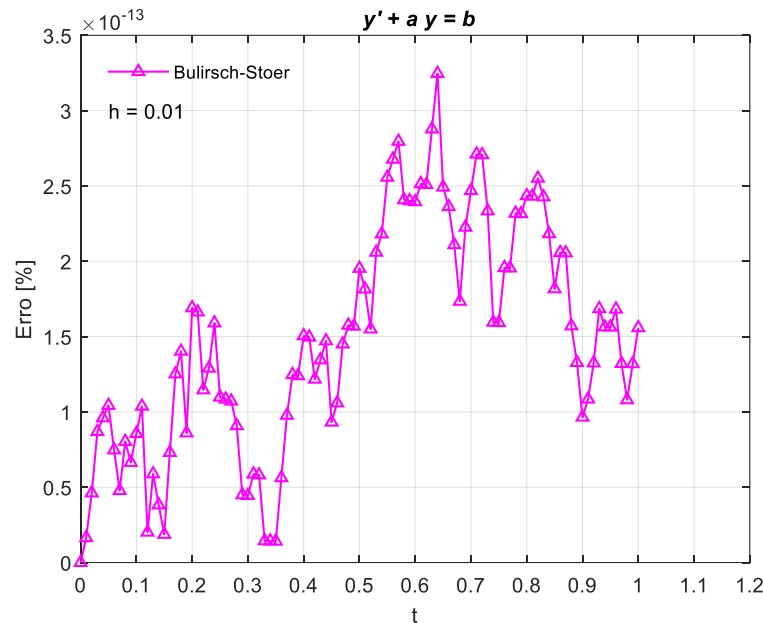
Fonte: Elaborado pelo autor, 2021.

Figura 36 - Solução analítica e numérica da equação diferencial $dy/dt + a.y = b$, usando o método de Bulirsch-Stoer, com passo $h = 0,01$



Fonte: Elaborado pelo autor, 2021.

Figura 37 - Variação do erro relativo (%) da equação diferencial $dy/dt + a.y = b$, usando o método de Bulirsch-Stoer, com passo $h = 0,01$



Fonte: Elaborado pelo autor, 2021.

5.3 Solução de dois problemas usando os métodos Dormand-Prince, Runge-Kutta de ordem 5 e Bulirsch-Stoer

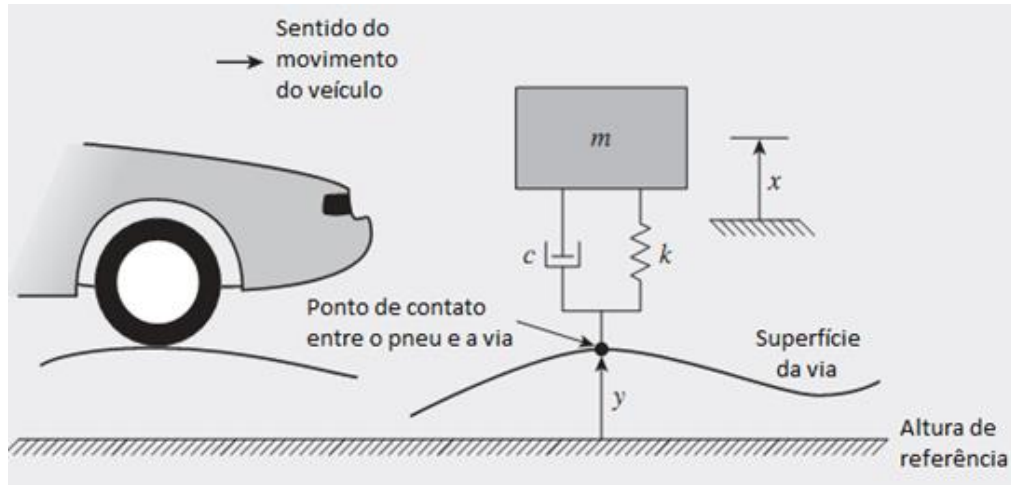
5.3.1 Modelagem do sistema de suspensão de um veículo

Trata-se de um problema de valor inicial de segunda ordem, referente à modelagem da suspensão de um veículo, que pode ser realizada considerando a aplicação de uma força de entrada na forma $ay(t) + by'(t)$ sobre a suspensão em que a e b são constantes (usando um quarto do veículo, por simetria), como mostrado na Figura 38.

No modelo simplificado, as massas da roda, do pneu e do eixo são desprezadas, e a massa m representa um quarto da massa do veículo. A constante da mola k representa a combinação dos efeitos do pneu e da mola da suspensão. A constante c representa a constante de amortecimento do-amortecedor. A posição de equilíbrio da massa m , quando $y = 0$, é $x = 0$. O deslocamento provocado pela superfície da via, $y(t)$, pode ser obtido pelo perfil da superfície da via e da velocidade do veículo. A equação do movimento do veículo pode ser obtida por meio do diagrama de corpo livre, mostrado na Figura 38.

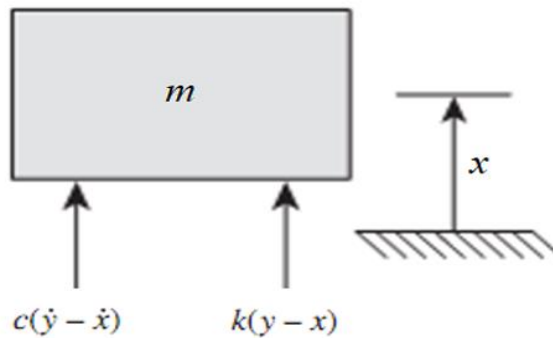
$$m\ddot{x} + c\dot{x} + kx = ky + c\dot{y} \quad (43)$$

Figura 38 - Modelo do sistema de suspensão de um veículo (um quarto de veículo)



Fonte: ÇENGEL; PALM III, 2012.

Figura 39 - Diagrama de corpo livre, assumindo que $dy/dt > dx/dt$ e $y > x$. Somente a força dinâmica é mostrada porque a força estática é cancelada pela força gravitacional mg



Fonte: ÇENGEL; PALM III, 2012.

A resposta do veículo à elevação de uma lombada com $0,5\text{ m}$ de altura e 1 m de comprimento, movendo-se a 18 m/s pode ser simulada resolvendo a equação diferencial do movimento, equação (43). Para isto é necessário o perfil da elevação, nesse caso, dado por:

$$y(z) = 5,4366 z e^{-4z} \quad (44)$$

Aqui z é a distância horizontal percorrida pelo veículo quando ele está sobre a elevação. O deslocamento $y(t)$ sentido pela suspensão está relacionado com $y(z)$, por meio da velocidade, em que $z = v t$. Como $v = 18 \text{ m/s}$, substituindo na equação (44) obtém-se:

$$y(t) = 97,858 t e^{-72t} \quad (45)$$

Os dados representativos de uma suspensão real são:

$$\begin{aligned} \text{Massa:} & \quad m = 240 \text{ kg}; \\ \text{Constante de amortecimento:} & \quad c = 5.000 \text{ [N} \cdot \text{s/m]} \\ \text{Constante elástica da mola:} & \quad k = 16.000 \text{ [N/m]} \end{aligned}$$

Combinando a equação (42) com a equação (45) e substituindo esses valores, obtém-se a equação do movimento:

$$240\ddot{x} + 5.000\dot{x} + 16.000x = (489.290 - 33.663.152 t)e^{-72 t} \quad (46)$$

Simplificando e colocando a equação (46) na sua forma-padrão, tem-se:

$$\ddot{x} + \frac{5.000}{240}\dot{x} + \frac{16.000}{240}x = \left(\frac{489.290 - 33.663.152 t}{240}\right)e^{-72 t} \quad (47)$$

A solução da equação (47) foi realizada usando os métodos Dormand-Prince, Runge-Kutta de quinta ordem e Bulirsch-Stoer transformando-a em um sistema de duas equações de primeira ordem da forma:

$$\begin{cases} \frac{dx_1}{dt} = x_2 \\ \frac{dx_2}{dt} = C_1x_2 + C_2x_1 + f(x) \end{cases} \quad (48)$$

Os resultados são apresentados a seguir, para vários valores do passo de integração para os métodos de Runge-Kutta de ordem 4, Dormand-Prince e Bulirsch-Stoer. A variação do tempo computacional em função do tamanho do passo encontra-se na Figura 39.

Os valores do tempo computacional e as estimativas finais da solução encontram-se na Tabela 3 e representados graficamente nas Figuras 40 a 43.

Tabela 3 - Tempo computacional e estimativas finais da solução do problema do sistema de suspensão de um veículo.

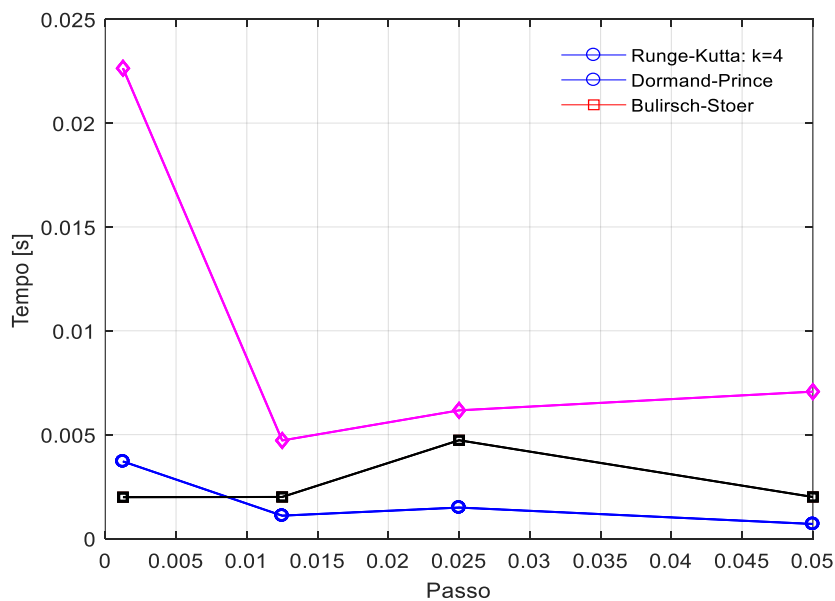
Passo	Tempo Computacional [s]*		
	<i>R – K4</i>	<i>D – P</i>	<i>B – S</i>
$h = 0,05000$	0,00070	0,00200	0,00707
$h = 0,02500$	0,00149	0,00473	0,00617
$h = 0,01250$	0,00110	0,00200	0,00472
$h = 0,00125$	0,00371	0,00199	0,02264

Passo	Estimativa Final		
	<i>R – K4</i>	<i>D – P</i>	<i>B – S</i>
$h = 0,05000$	$1,0355 \times 10^{-1}$	$9,3881 \times 10^{-4}$	$9,3304 \times 10^{-4}$
$h = 0,02500$	$9,7124 \times 10^{-3}$	$9,3881 \times 10^{-4}$	$9,3304 \times 10^{-4}$
$h = 0,01250$	$1,5333 \times 10^{-3}$	$9,3881 \times 10^{-4}$	$9,3304 \times 10^{-4}$
$h = 0,00125$	$9,3310 \times 10^{-4}$	$9,3881 \times 10^{-4}$	$9,3304 \times 10^{-4}$

Fonte: Elaborado pelo autor, 2021.

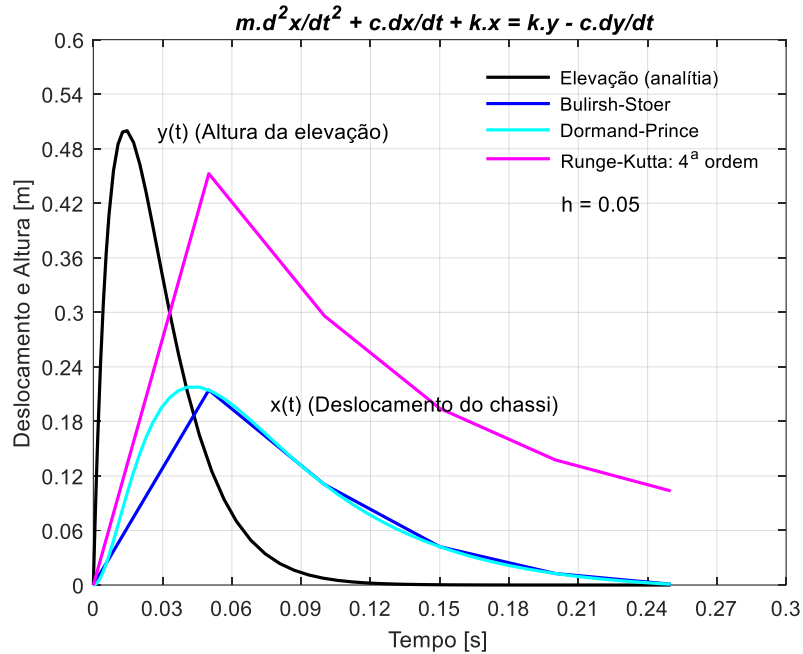
* R-K4: Runge-Kutta, ordem 4; D-P: Dormand-Prince; B-S: Bulirsch-Stoer

Figura 40 - Tempo computacional da solução do problema do sistema de suspensão de um veículo em função do tamanho do passo de integração



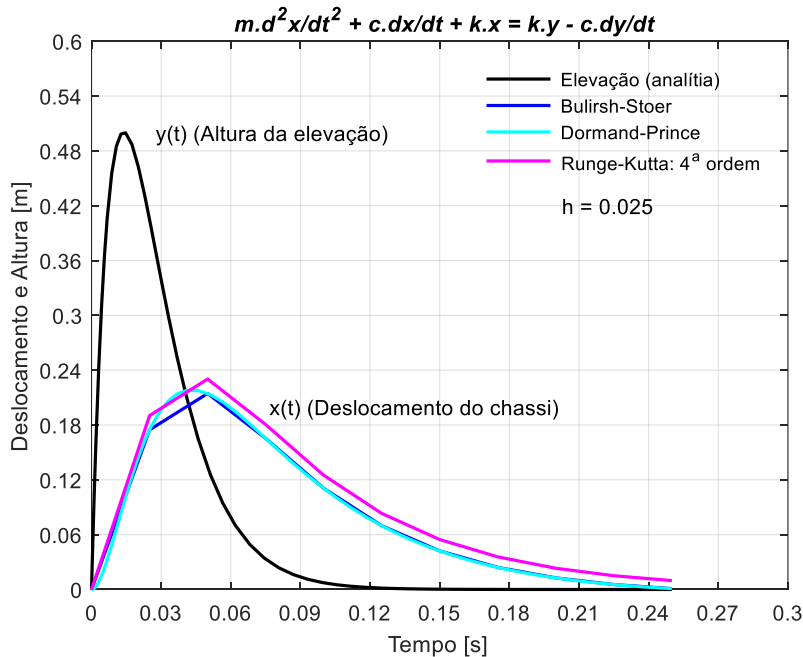
Fonte: Elaborado pelo autor, 2021.

Figura 41 - Soluções numéricas do problema de sistema de suspensão de um veículo, usando o método de Runge-Kutta de quarta ordem ($k = 4$), com passo fixo, e os métodos de passo adaptativo de Dormand-Prince e de Bulirsch-Stoer, com $h = 0,05$



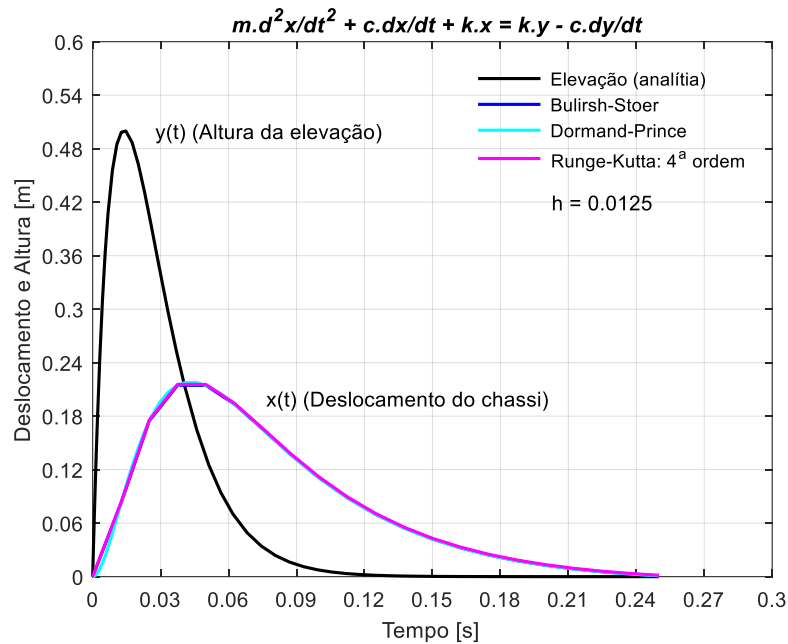
Fonte: Elaborado pelo autor, 2021.

Figura 42 - Soluções numéricas do problema de sistema de suspensão de um veículo, usando o método de Runge-Kutta de quarta ordem ($k = 4$), com passo fixo, e os métodos de passo adaptativo de Dormand-Prince e de Bulirsch-Stoer, com $h = 0,025$



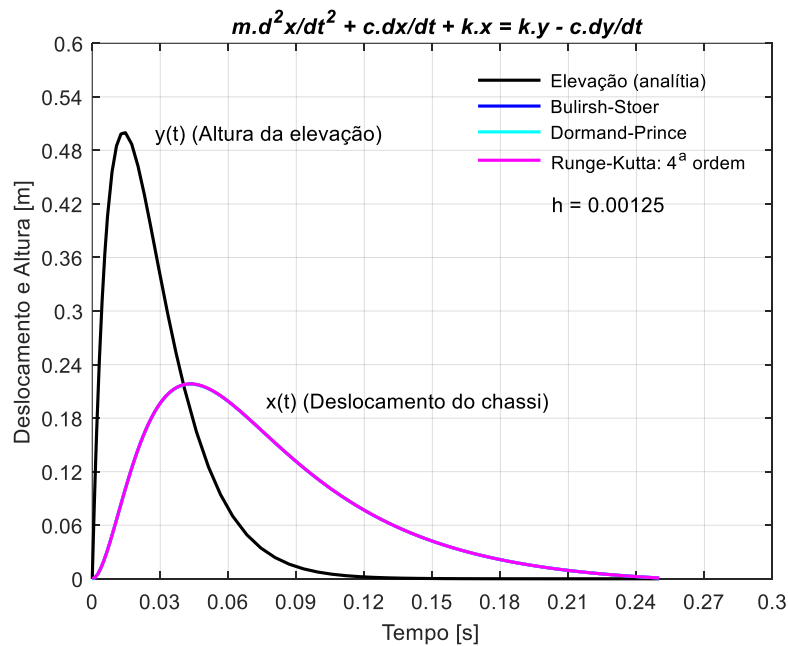
Fonte: Elaborado pelo autor, 2021.

Figura 43 - Soluções numéricas do problema de sistema de suspensão de um veículo, usando o método de Runge-Kutta de quarta ordem ($k = 4$), com passo fixo, e os métodos de passo adaptativo de Dormand-Prince e de Bulirsch-Stoer, com $h = 0,0125$



Fonte: Elaborado pelo autor, 2021.

Figura 44 - Soluções numéricas do problema de sistema de suspensão de um veículo, usando o método de Runge-Kutta de quarta ordem ($k = 4$), com passo fixo, e os métodos de passo adaptativo de Dormand-Prince e de Bulirsch-Stoer, com $h = 0,00125$



Fonte: Elaborado pelo autor, 2021.

Observa-se, na Figura 39, que o comportamento do tempo computacional em relação ao tamanho do passo não apresentou uma tendência funcional bem definida. Não foi possível, neste trabalho, encontrar uma explicação razoável para essa aleatoriedade, embora se cogite que possa ser um efeito do estado transitório do processador do computador, influenciado por algum evento eletromagnético. O que foi constatado, efetivamente, foi a obtenção de valores de tempo de processamento diferente para o mesmo problema, quando se repete o processamento.

Observa-se nas Figuras 40 a 43 que, para todos os métodos, houve uma melhora na precisão das respostas do deslocamento do chassi em função do tempo, com a diminuição do passo de integração. Todavia, observa-se que o método de Dormand-Prince apresentou, praticamente, a mesma resposta, independentemente do tamanho do passo de integração. Portanto, esse método, depende muito pouco do passo de integração inicial, por causa do processo de adaptação do tamanho do passo. Nesse problema, os métodos de Runge-Kutta de quarta ordem e de Bulirsch-Stoer apresentaram soluções viáveis somente para $h < 0,01$, no domínio $0 \leq t \leq 0,25$.

5.3.2 Solução da equação de secagem de Page

Trata-se de um problema de valor inicial de primeira ordem, referente à integração da equação de secagem em camada fina que resulta na equação de Page. A forma integral resultante da solução analítica da equação diferencial é bastante complexa (APÊNDICE G). Devido à sua ampla aplicação na simulação da cinética de secagem de produtos biológicos, esta equação foi selecionada para compor este estudo utilizando três métodos numéricos: (i) Runge-Kutta de quarta ordem, (ii) Dormand-Prince e (iii) Bulirsch-Stoer. Na sua forma diferencial é descrita por:

$$\frac{dM}{dt} + k \cdot b \cdot t^{b-1} \cdot e^{-k \cdot t^b} = 0, \quad M(0) = 1 \quad (49)$$

em que:

- M = Razão do teor de água, $[kg/kg]$;
- t = Tempo, $[min]$;
- k = Constante de secagem, $[kg/kg \cdot min]$
- b = Parâmetro de secagem, $[adimensional]$.

Os parâmetros k e b utilizados neste trabalho são para sementes de acerola secas em camada fina de 1 *cm*, na faixa de temperatura de 40°C a 70°C e fluxo de ar de 1,5 *m/s*. Estes parâmetros são calculados em função da temperatura pelas seguintes expressões:

$$k = 0,1492 \cdot \exp\left(-\frac{1256,223}{T + 273,15}\right) \quad (50)$$

$$b = (-5,2500 \times 10^{-5}) \cdot T^2 + (9,5250 \times 10^{-3}) \cdot T + 0,85925 \quad (51)$$

Para comparação dos resultados numéricos com os dados calculados, foram utilizados valores-limites de tempo diferentes para cada temperatura, já que o tempo de secagem decresce com o aumento de temperatura. O tempo computacional, para cada método, encontra-se na Tabela 4 e as representações gráficas dos resultados estão nas Figuras 44 a 47.

Para todos os métodos, foram utilizados os valores para o tamanho do passo de integração igual a $h = 10$ [min] e $h = 5$ [min]. Os gráficos dos erros dos métodos de Runge-Kutta de quarta ordem e de Dormand-Prince não foram apresentados, devido a suas irrelevâncias.

Tabela 4 - Tempo computacional da solução da equação de secagem de Page.

Método	Tempo [s]			
	<i>h</i> = 10 [min]			
	<i>T</i> = 40°C	<i>T</i> = 50°C	<i>T</i> = 60°C	<i>T</i> = 70°C
Runge-Kutta de ordem 4	0,011700	0,012138	0,015603	0,015703
Dormand-Prince	0,001166	0,001307	0,001076	0,000928
Bulirsch-Stoer	0,001432	0,001134	0,001422	0,000672

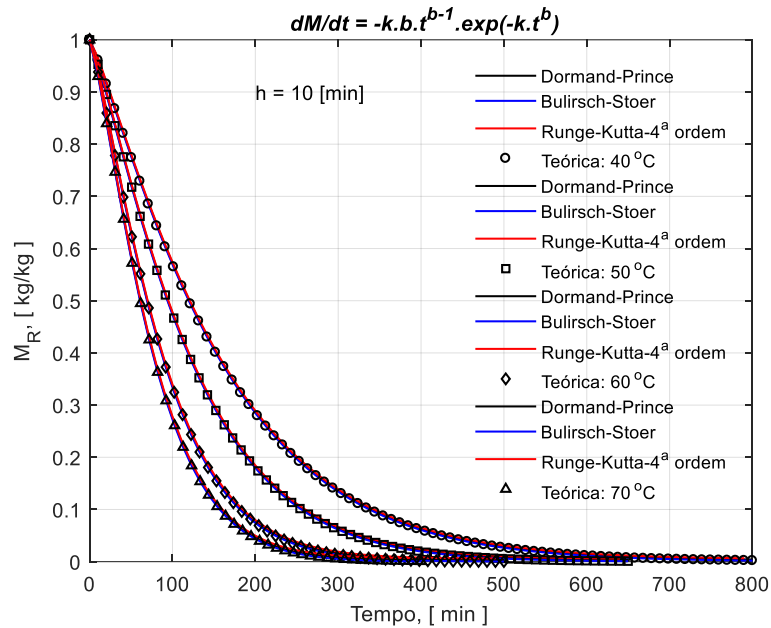
Método	Tempo [s]			
	<i>h</i> = 5 [min]			
	<i>T</i> = 40°C	<i>T</i> = 50°C	<i>T</i> = 60°C	<i>T</i> = 70°C
Runge-Kutta de ordem 4	0,016141	0,013032	0,011756	0,014358
Dormand-Prince	0,001047	0,001052	0,000900	0,001176
Bulirsch-Stoer	0,002124	0,001341	0,001396	0,001569

Método	Erro máximo [%]			
	<i>h</i> = 10 [min]			
	<i>T</i> = 40°C	<i>T</i> = 50°C	<i>T</i> = 60°C	<i>T</i> = 70°C
Runge-Kutta de ordem 4	244,13089	584,21779	2029,48016	1180,16198
Runge-Kutta-Fehlberg	118,44440	281,40535	964,70491	548,78170
Bulirsch-Stoer	$8,15317 \times 10^{-6}$	$1,37024 \times 10^{-5}$	$7,30121 \times 10^{-5}$	$3,20115 \times 10^{-5}$

Método	Erro máximo [%]			
	<i>h</i> = 5 [min]			
	<i>T</i> = 40°C	<i>T</i> = 50°C	<i>T</i> = 60°C	<i>T</i> = 70°C
Runge-Kutta de ordem 4	109,72585	254,93228	855,20943	491,16945
Dormand-Prince	53,21855	122,57705	405,00566	226,94053
Bulirsch-Stoer	$4,57452 \times 10^{-6}$	$1,73249 \times 10^{-5}$	$3,90247 \times 10^{-5}$	$1,68943 \times 10^{-5}$

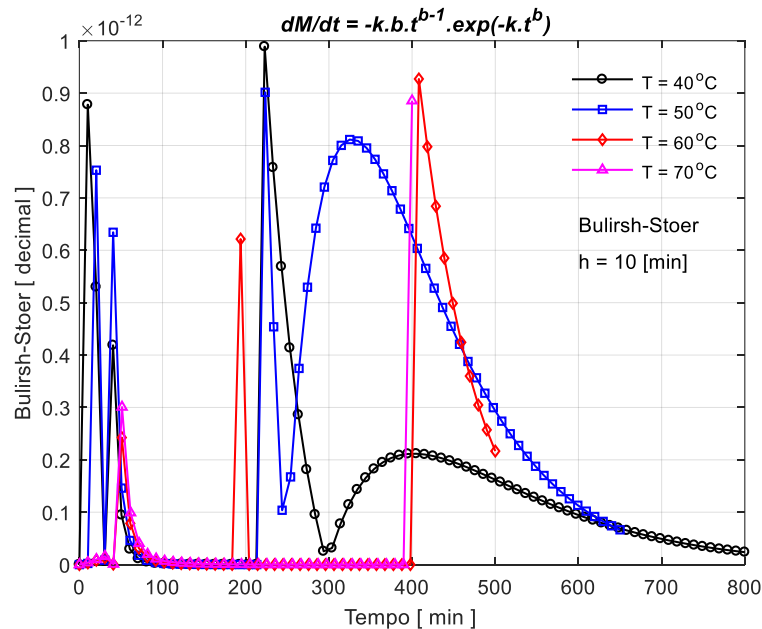
Fonte: Elaborado pelo autor, 2021.

Figura 45 - Soluções numéricas da equação de Page, usando os métodos de Runge-Kutta de quarta ordem ($k = 4$), Dormand-Prince e Bulirsch-Stoer. $h = 10$ [min]



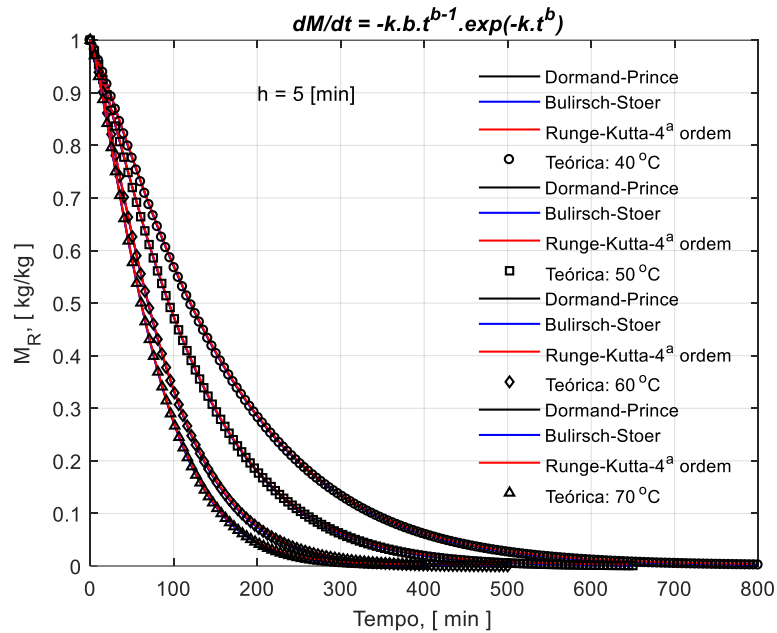
Fonte: Elaborado pelo autor, 2021.

Figura 46 - Variação do erro relativo de aproximação na solução numérica da equação de Page, usando o método de Bulirsch-Stoer. $h = 10$ [min 5]



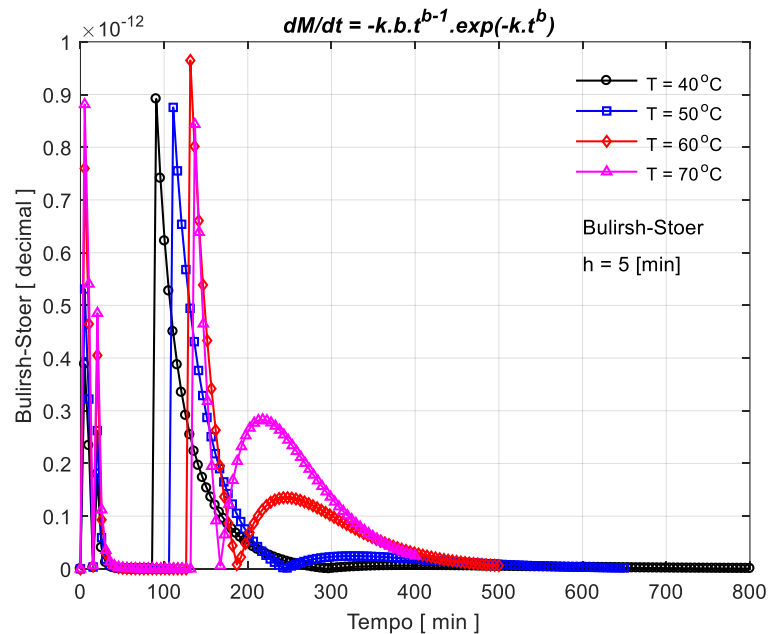
Fonte: Elaborado pelo autor, 2021.

Figura 47 - Soluções numéricas da equação de Page, usando os métodos de Runge-Kutta de quarta ordem ($k = 4$), Dormand-Prince e Bulirsch-Stoer. $h = 5$ [min]



Fonte: Elaborado pelo autor, 2021.

Figura 48 - Variação do erro relativo de aproximação na solução numérica da equação de Page, usando o método de Bulirsch-Stoer. $h = 5$ [min]



Fonte: Elaborado pelo autor, 2021.

Observa-se nas Figuras 44 e 46 que, aparentemente, todos os três métodos forneceram estimativas precisas para o problema, para $h = 10$ min e $h = 5$ min, mas

somente o método de Bulirsch-Stoer forneceu estimativas precisas. Isso pode ser verificado pela análise dos dados da Tabela 4. Além disso, nas Figuras 45 e 47 observa-se o comportamento caótico do erro no método de Bulirsch-Stoer, para manter o valor do erro máximo dentro da de tolerância especificada. Por outro lado, observa-se os valores exorbitantes dos erros nos métodos de Dormand-Prince e Runge-Kutta de quarta ordem. Isso não significa que eles não funcionem, mas que podem exigir um valor de passo de integração infinitesimalmente pequeno, que pode torná-los inviáveis. Portanto, deve-se ter conhecimento da natureza do problema e perspicácia suficientes para escolher o método e o tamanho do passo de integração apropriados.

5.4 Considerações finais

Independentemente do tamanho do passo de integração, a precisão dos resultados depende da característica de cada método utilizado e do tipo de problema que está sendo resolvido. A acurácia dos métodos de Runge-Kutta, por serem de passo fixo, é mais sensível ao tamanho do passo de integração. Observou-se que, principalmente, para os métodos de Runge-Kutta de primeira e de segunda ordem (métodos de Euler) a precisão dos resultados é altamente sensível ao tamanho do passo. Portanto, esses métodos podem ser usados para resolver problemas simples, desde que o tamanho do passo de integração seja adequado.

O método de Runge-Kutta de terceira ordem deve ser utilizado com cautela, ou combinado com outras técnicas, para resolver problemas mais complexos e fornecer resultados satisfatórios, com tamanho do passo adequado. Entretanto, o método de Runge-Kutta clássico de quarta ordem é o mais utilizado para a solução de problemas que não apresentem rigidez na sua solução.

O método de Dormand-Prince possui sete estágios e incorpora estimativas do erro local usando um método de quarta ordem e avança a solução usando uma extrapolação de quinta ordem, com implementação da adaptação do tamanho do passo ao longo do processo de integração, para manter o erro dentro de uma faixa de tolerância especificada. Portanto, é menos sensível ao tamanho do passo inicial, fornecendo excelentes resultados para todo tipo de problema.

O método de Bulirsch-Stoer baseia-se no método do ponto médio modificado e depende, em certo grau, do tamanho do passo, pois faz interpolações para melhorar a precisão dos resultados. É mais eficiente para certos tipos de problema, em que a solução apresenta rigidez no comportamento. Nos problemas dessa dissertação, esse método apresentou os melhores resultados, com valores baixos do erro de convergência.

Nas discussões da seção anterior fica claro que, em geral, há uma tendência de aumento da exatidão dos métodos com a diminuição do tamanho do passo de integração. Entretanto, essa hipótese não pode ser considerada de caráter geral, pois, dependendo da natureza do problema, o melhor método de solução pode ser outro. Isso fica claro na solução do problema de modelagem do sistema de suspensão de um veículo.

6 CONCLUSÕES

É essencial conhecer a natureza do problema a ser resolvido, para escolher adequadamente o método numérico e o tamanho do passo de integração a serem utilizados. Quanto mais alta a ordem do método de integração, maior a possibilidade de usar um tamanho de passo maior com precisão desejada.

Em todos os métodos numéricos, geralmente, a acurácia aumenta com a diminuição do tamanho do passo, principalmente nos métodos de passo fixo. Portanto, os métodos de passo adaptativo como os métodos de Dormand-Prince e Bulirsch-Stoer, em geral, fornecem estimativas mais precisas, desde que usados de forma adequada.

Os métodos com algoritmo de adaptação do passo, como os métodos de Bulirsch-Stoer e Dormand-Prince, são pouco sensíveis ao tamanho inicial do passo porque estes são modificados ao longo do processo para manter a acurácia.

As duas sub-rotinas implementadas nesta dissertação, FixedRK e BulirschStoer, se comportaram muito bem na solução do problema de validação. Para a solução do problema de modelagem do sistema de suspensão de um veículo, todos os três métodos foram adequados, dependendo do tamanho do passo. Somente o método de Dormand-Prince foi conservador, fornecendo excelentes resultados, independentemente do tamanho do passo.

Para a solução da equação de secagem de Page, não-linear e com alta rigidez, apenas o método de Bulirsch-Stoer forneceu resultados excelentes. Os métodos foram inadequados com o tamanho de passo utilizado.

Portanto, conclui-se que o conhecimento da natureza do problema é fundamental para a escolha do método de solução e do tamanho do passo de integração para se obter resultados adequados.

REFERÊNCIAS

- ABRAMOWITZ, M.; STEGUN, I. A. *Handbook of mathematical functions*. Reprinted 1968. Washington: National Bureau of Standards; New York: Dover, 1964.
- ASHINO, R.; NAGASE, M.; VAILLANCOURT, R. Behind and Beyond the Matlab ODE Suite. *CRM-2651*, Jan. 2000.
- BASSANEZI, R. C.; FERREIRA Jr., W. C. *Equações diferenciais com aplicações*. 1. ed. São Paulo: Editora Harbra, 1988.
- BISWAS, B. N.; CHATTERJEE, S.; MUKHERJEE, S. P.; PAL, S. A Discussion on Euler Method: A Review. *Electronic Journal of Mathematical Analysis and Applications*, v. 1, n. 2, p. 294-317, July 2013. ISSN: 2090-792X (online).
- BOYER, C. B. *História da matemática*. Tradução: Elza F. Gomide. 2. ed. São Paulo: Editora Edgard Blücher, 1996.
- BRONSON, R.; COSTA, G. B. *Differential equations*. 4th ed. [S. l.]: McGraw-Hill Education, 2014. 385 p. (Schaum's Outline Series).
- BUTCHER, J. C. Numerical methods for ordinary differential equations in the 20th century. *Journal of Computational and Applied Mathematics*, v. 25, Issues 1-2, 15, p. 1-29, Dec. 2000.
- ÇENGEL, Y. A.; PALM III, W. J. *Differential equations for engineers and scientists*. [S. l.]: McGraw-Hill Education, 2012. 611 p.
- DEUFLHARD, P. Order and stepsize control in extrapolation methods. *Numerische Mathematik*, v. 41, n. 3, p.399-422, 1983. DOI 10.1007/BF01418332.
- DEUFLHARD, P. *SIAM Review*, v. 27, p. 505-535, 1985.
- DUTTA, H; GÜNERHAN, H.; ALI, K. K.; YILMAZER, R. Exact soliton solutions to the cubic-quartic non-linear schrödinger equation with conformable derivative. *Mathematical and Statistical Physics*, Frontiers in Physics, March 2020. <https://doi.org/10.3389/fphy.2020.00062>.
- GEAR, C. W. *Numerical initial value problems in ordinary differential equations* Englewood Cliffs, NJ: Prentice-Hall, 1971.
- GHANBARI, B.; BALEANU, D. New optical solutions of the fractional Gerdjikov-Ivanov equation with conformable derivative. *Mathematical and Statistical Physics*, Frontiers in Physics, May 2020. <https://doi.org/10.3389/fphy.2020.00167>.
- HAIRER, E.; NØRSETT, S. P.; WANNER, G. *Solving ordinary differential equations I: nonstiff problems*. Berlin, New York: Springer-Verlag, 1993. ISBN 978-3-540-56670-0.

HOSSEINI, K.; MIRZAZADEH, M.; OSMAN, M. S.; AL QURASHI, M.; BALEANU, D. Solitons and jacobi elliptic function solutions to the complex Ginzburg–Landau equation. *Mathematical and Statistical Physics*, Frontiers in Physics, June 2020. <https://doi.org/10.3389/fphy.2020.00225>.

HULL, T. E.; W. H., ENRIGHT; FELLEN, B. M.; SEDGWICK, A. E. Comparing numerical methods for ordinary differential equations. *Journal on Numerical Analysis*, SIAM, 2006.

LAPIDUS, L.; SEINFELD, J. *Numerical solution of ordinary differential equations*. New York: Academic Press. 1971.

NAGLE, R. K.; SAFF, E. B.; SNIDER, A. D. *Fundamentals of differential equations and boundary value problems*. 6th ed. Boston, MA: Addison-Wesley, Pearson Education, 2012. 810 p.

OBERLEITHNER, K.; PASCHEREITA, C. O.; SORIAB, J. Stability analysis of time-averaged jet flows: fundamentals and application. In: IUTAM-ABCM SYMPOSIUM ON LAMINAR TURBULENT TRANSITION, *Procedia IUTAM*, v. 1, 2015. p.141–146.

PRESS, W. H.; TEUKOLSKY, S. A.; VETTERLING, W. T.; FLANNERY, B. P. Section 17.3. Richardson Extrapolation and the Bulirsch-Stoer Method. In: *Numerical Recipes: the art of scientific computing*. 3rd ed. New York: Cambridge University Press. 2007. ISBN 978-0-521-88068-8.

QUARTERONI, A.; SACCO, R.; SALERI, F. Numerical mathematics. Texts in Applied Mathematics. [S. l.]: Springer, 2017. 654 p.

STOER, J.; BULIRSCH, R. *Introduction to numerical analysis*, 3rd ed. New York: Springer, 2002. Chapter 7.

STOER, J.; BULIRSCH, R. *Introduction to numerical analysis*. New York: Springer-Verlag, 1980.

VANANI, S. K.; AMINATAEI, A. Numerical solution of differential algebraic equations using a multiquadric approximation scheme. *Mathematical and Computer Modelling*, v. 53, n. 5-6, p. 659-666. 2011. <https://doi.org/10.1016/j.mcm.2010.10.002>.

ZILL, D. G. *Differential equations with boundary value problems*. 9th ed. Boston, MA, USA: CENGAGE Learning, 2016. 630 p.

APÊNDICE A - Sub-Rotina FixedRK

```

function [tvals,yvals] = FixedRK(fname,t0,y0,h,k,n)
%
% Solução da equação diferencial ordinária
% usando os métodos de Runge-Kutta de passo fixo (ordem 1, 2, 3, 4 e
5)
%
%
%                               Implementado por:
%                               J. H. Martins & M. A. A. Ribeiro (2021)
%
%
% Produz a solução aproximada para o problema de valor inicial
%
%                                $y'(t) = f(t,y(t)), y(t_0) = y_0,$ 
%
% usando uma estratégia baseada no método de Runge-Kutta de ordem k,
% com tamanho de passo fixo.
%
% Predefinidos:
% fname = cadeia de caracteres que nomeia a função f.
% t0 = tempo inicial (ou outra variável inicial).
% y0 = vetor das condições iniciais.
% h = tamanho do passo.
% k = ordem do método. (1<=k<=5).
% n = número de passos utilizados.
%
% Posdefinidos:
% tvals(j) = t0 + (j-1)h, j=1:n+1
% yvals(:j) = solução aproximada em t = tvals(j), j=1:n+1
%
tc = t0;
yc = y0;
tvals = tc;
yvals = yc;
fc = feval(fname,tc,yc);
for j=1:n
    [tc,yc,fc] = RKstep(fname,tc,yc,fc,h,k);
    yvals = [yvals yc];
    tvals = [tvals tc];
end

```

Continuação ...

```

function [tnew,ynew,fnew] = RKstep(fname,tc,yc,fc,h,k)
%
% Predefinidos:
% fname - é uma cadeia de caractere que nomeia uma função da forma
f(t,y),
% em que t é um escalar e y é um vetor coluna com as derivadas.
%
% yc - é uma solução aproximada de  $y'(t) = f(t,y(t))$ , em  $t=tc$ .
%
% fc = f(tc,yc).
%
% h - é o passo de integração.
%
% k - é a ordem do método de Runge-Kutta usado:  $1 \leq k \leq 5$ .
%
% Posdefinidos:
% tnew = tc + h, é o valor da variável independente atualizado.
% ynew, é uma solução aproximada em  $t = tnew$ .
% fnew = f(tnew,ynew).
%
if k==1
    k1 = h*fc;
    ynew = yc + k1;
elseif k==2
    k1 = h*fc;
    k2 = h*feval(fname,tc+h,yc+k1);
    ynew = yc + (k1 + k2)/2;
elseif k==3
    k1 = h*fc;
    k2 = h*feval(fname,tc+(h/2),yc+(k1/2));
    k3 = h*feval(fname,tc+h,yc-k1+2*k2);
    ynew = yc + (k1 + 4*k2 + k3)/6;
elseif k==4
    k1 = h*fc;
    k2 = h*feval(fname,tc+(h/2),yc+(k1/2));
    k3 = h*feval(fname,tc+(h/2),yc+(k2/2));
    k4 = h*feval(fname,tc+h,yc+k3);
    ynew = yc + (k1 + 2*k2 + 2*k3 + k4)/6;
elseif k==5
    k1 = h*fc;
    k2 = h*feval(fname,tc+(h/4),yc+(k1/4));
    k3 = h*feval(fname,tc+(3*h/8),yc+(3/32)*k1 ...
        +(9/32)*k2);
    k4 = h*feval(fname,tc+(12/13)*h,yc+(1932/2197)*k1 ...
        -(7200/2197)*k2+(7296/2197)*k3);
    k5 = h*feval(fname,tc+h,yc+(439/216)*k1 ...
        - 8*k2 + (3680/513)*k3 - (845/4104)*k4);
    k6 = h*feval(fname,tc+(1/2)*h,yc-(8/27)*k1 ...
        + 2*k2 - (3544/2565)*k3 + (1859/4104)*k4 - (11/40)*k5);
    ynew = yc + (16/135)*k1 + (6656/12825)*k3 + ...
        (28561/56430)*k4 - (9/50)*k5 + (2/55)*k6;
end
tnew = tc+h;
fnew = feval(fname,tnew,ynew);
% Fim

```

APÊNDICE B - Sub-Rotina BulirschStoer

```

function [z, info] = BulirschStoer(dynFun,t,z0,tol)
% [z, info] = BulirschStoer(dynFun,t,z0,tol)
%
% Solves an initial value problem using the Bulirsch-Stoer method.
This
% method is ideal for high-accuracy solutions to smooth initial value
% problems.
%
% Computes  $z(t)$  such that  $dz/dt = \text{dynFun}(t,z)$ , starting from the
initial
% state  $z_0$ . The solution at the grid-points will be accurate to within
tol.
%
% If the provided grid is insufficient, this function will
automatically
% introduce intermediate grid points to achieve the required accuracy.
%
% INPUTS:
%   dynFun = function handle for the system dynamics
%           dz = dynFun(t,z)
%           t = scalar time
%           z = [nz,1] = state as column vector
%           dz = [nz,1] = derivative of state as column vector
%   t = [1,nt] = time grid-point vector
%   z0 = [nz,1] = initial state vector
%   tol = [nz,1] = error tolerance along each dimension. If tol is a
%         scalar, then all dimensions will satisfy that error tolerance.
%
% OUTPUTS:      (nt = n+1)
%   z = [nz,nt] = solution to the initial value problem
%   info
%       .error = [nx,nt] = error estimate at each grid point
%       .nFunEval = [1,nt] = number of function evaluations for each
point
%
% NOTES:
%   Implementation details:
http://web.mit.edu/ehliu/Public/Spring2006/18.304/implementationbuli
rsch stoer.pdf
%
% TUTORIAL:
%   The big idea here is based on Richardson extrapolation. Suppose
that
%   you solve an initial value problem using the modified mid-point
method
%   (below), where you can pick the number of sub-steps (n). The
solution
%   will become increasingly accurate as the number of sub-steps
becomes
%   large.
%

```

```

% Now, imagine that you solve the same problem several times, each
time
% using more sub-steps. You should be able to observe a trend: the
% solutions asymptotically approach some value. Richardson
extrapolation
% is the math (algorithm) that looks at this sequence of improving
% approximations to the solution, and then extrapolates, to the
limit as
% the step size goes to zero.
%
% The modified mid-point method is not chosen at random. It has a
special
% property: the expression for the error goes up by powers of two,
rather
% than one. This makes the convergence particularly good.
%
% This function (BulirschStoer) is just a wrapper for the sub-function
% BulirschStoerStep, which is where the real work is done.
%
% If a step fails, how many sub-steps should be created in its place?
nStepRefine = 3;

% Simple logistics and memory allocation:
nt = length(t);
nz = size(z0,1);
z = zeros(nz,nt);
z(:,1) = z0;
info.error = zeros(size(z));
info.nFunEval = zeros(1,nt);

% March forward in time, from grid-point to grid-point
for i=2:nt

    tSpan = [t(i-1), t(i)];
    [zF, stepInfo] = BulirschStoerStep(dynFun, tSpan, z(:,i-1), tol);

    if strcmp(stepInfo.exit,'converged') %Successful step!
        z(:,i) = zF;
        info.error(:,i) = stepInfo.error;
        info.nFunEval(i) = stepInfo.nFunEval;

    else %Failed to converge -- try again on a better mesh
        time = linspace(tSpan(1),tSpan(2),nStepRefine+1);
        [zTmp, infoTmp] = BulirschStoer(dynFun,time,z(:,i-1),tol);
        z(:,i) = zTmp(:,end);
        info.error(:,i) = infoTmp.error(:,end);
        info.nFunEval(i) = sum(infoTmp.nFunEval);

    end

end
end
end
%
```

```

function [zF, info] = BulirschStoerStep(dynFun, tSpan, z0, tol)
% [zF, info] = BulirschStoerStep(dynFun, tSpan, z0, tol)
%
% Computes a single step using the Bulirsch-Stoer method
%
% INPUTS:
%   dynFun = function handle for the system dynamics
%           dz = dynFun(t,z)
%           t = scalar time
%           z = [nz,1] = state as column vector
%           dz = [nz,1] = derivative of state as column vector
%   tSpan = [1,2] = [t0, tF] = time span for the step
%   z0 = [nz,1] = initial state vector
%   tol = [nz,1] = error tolerance along each dimension. If tol is a
%         scalar, then all dimensions will satisfy that error tolerance.
%
% OUTPUTS:      (nt = n+1)
%   zF = [nz,1] = final state
%   info = struct with solver information:
%         .exit = exit condition
%         'converge' = successful convergence
%         'maxRefine' = reached max refinement; did not converge
%         .error = [nz,1] = error estimate along each dimension
%         .nFunEval = scalar int = count calls to dynFun
%         .nRefine = how many refinement steps were required?
%
% NOTES:
%   Implementation details:
%
http://web.mit.edu/ehliu/Public/Spring2006/18.304/implementation\_bulirsch\_stoer.pdf
%
% Set an upper limit on the number of mesh refinements in the sequence
nRefineMax = 8;

% Simple logistics and memory allocation
n = 2*(1:nRefineMax);
nz = size(z0,1);
if length(tol)==1
    tol = tol*ones(size(z0));
end
T = zeros(nz,nRefineMax,nRefineMax); %Extrapolation table
E = zeros(nz,nRefineMax); %Error estimate table

info.exit = 'maxRefine'; %Assume that we fail to meet tolerance

```

Continuação ...

```

for j=1:nRefineMax %Loop over the sequence of improving meshes

    % Compute the estimate of the solution on the current mesh
    [~,z] = modifiedMidpointRule(dynFun, tSpan, z0, n(j));
    T(:,j,1) = z(:,end);

    if j>1

        for k=2:j
            num = T(:,j,k-1) - T(:,j-1,k-1);
            den = (n(j)/(n(j-k+1)))^2 - 1;
            T(:,j,k) = T(:,j,k-1) + num/den;
        end

        % Compute the error estimates:
        E(:,j) = abs(T(:,j,j-1) - T(:,j,j));

        % Check convergence:
        if all(E(:,j)<tol)
            info.exit = 'converged';
            break;
        end
    end
end

end

% Other useful things:
info.error = E(:,j); %Error estimate
info.nFunEval = sum(n(1:j)); %number of function evaluations
info.nRefine = j;

% Return the estimate of the solution:
zF = T(:,j,j);

end

%~~~~~%

```

Continuação ...

```

function [t,z] = modifiedMidpointRule(dynFun, tSpan, z0, n)
% [t,z] = modifiedMidpointRule(dynFun, tSpan, z0, n)
%
% Approximates the solution to the initial value problem by numerical
% integration with the modified mid-point rule.
%
% INPUTS:
%   dynFun = function handle for the system dynamics
%           dz = dynFun(t,z)
%           t = scalar time
%           z = [nz,1] = state as column vector
%           dz = [nz,1] = derivative of state as column vector
%   tSpan = [1,2] = [t0,tF] = time span for simulation
%   z0 = [nz,1] = initial state vector
%   n = scalar integer number of steps.   (require: n > 2)
%
% OUTPUTS:      (nt = n+1)
%   t = [1,nt] = time stamps for intermediate points
%   z = [nz,nt] = state estimate at final time
%
% NOTES:
%   Implementation details:
%
http://web.mit.edu/ehliu/Public/Spring2006/18.304/implementation\_bulirsch\_stoer.pdf
%
nt = n+1;
nz = size(z0,1);

t0 = tSpan(1);
tF = tSpan(2);
h = (tF-t0)/n;
t = linspace(tSpan(1), tSpan(2), nt);
z = zeros(nz,n);

% Initialize and then run the modified mid-point method.
z(:,1) = z0;
z(:,2) = z0 + h*dynFun(t0,z0);
for i=3:nt
    z(:,i) = z(:,i-2) + 2*h*dynFun(t(i-1),z(:,i-1));
end

% Refine the final point using the dynamics function at final point.
z(:,nt) = 0.5*(z(:,nt) + z(:,nt-1) + h*dynFun(t(nt),z(:,nt)));
end

```


APÊNDICE C - Amostras de Resultados do Problema 1

Runge-Kutta: Quinta ordem

Ordem: k = 5 h = 0.10000
 Tempo_FixRK [s] = 0.00062

n	t	Exata	RK	ERK
0	0.000	5.0000e+00	5.0000e+00	0.0000e+00
1	0.100	8.2968e+00	8.2968e+00	4.1102e-04
2	0.200	1.0507e+01	1.0507e+01	4.3513e-04
3	0.300	1.1988e+01	1.1988e+01	3.8345e-04
4	0.400	1.2981e+01	1.2981e+01	3.1649e-04
5	0.500	1.3647e+01	1.3647e+01	2.5226e-04
6	0.600	1.4093e+01	1.4093e+01	1.9649e-04
7	0.700	1.4392e+01	1.4392e+01	1.5047e-04
8	0.800	1.4592e+01	1.4592e+01	1.1369e-04
9	0.900	1.4727e+01	1.4727e+01	8.4949e-05
10	1.000	1.4817e+01	1.4817e+01	6.2885e-05

Erro Relativo Máximo [%] 4.3513e-04

Runge-Kutta de quinta ordem; Doramnd-Prince; Bulirsch-Stoer

k = 5 h = 0.10000
 Tempo_FixRK [s] = 0.01420
 Tempo_ode45 [s] = 0.22797
 Tempo_Bulirsch-Stoer [s] = 0.03383

n	t	Exata	RK	DP	BS	ERK	EDP	EBS
0	0.000	5.0000e+00	5.0000e+00	5.0000e+00	5.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00
1	0.100	8.2968e+00	8.2968e+00	8.2968e+00	8.2968e+00	4.1102e-04	2.9939e-04	2.7833e-13
2	0.200	1.0507e+01	1.0507e+01	1.0507e+01	1.0507e+01	4.3513e-04	6.6141e-05	2.1979e-13
3	0.300	1.1988e+01	1.1988e+01	1.1988e+01	1.1988e+01	3.8345e-04	1.5389e-05	1.9263e-13
4	0.400	1.2981e+01	1.2981e+01	1.2981e+01	1.2981e+01	3.1649e-04	4.3108e-05	2.4632e-13
5	0.500	1.3647e+01	1.3647e+01	1.3647e+01	1.3647e+01	2.5226e-04	4.8899e-05	4.0352e-13
6	0.600	1.4093e+01	1.4093e+01	1.4093e+01	1.4093e+01	1.9649e-04	4.5639e-05	4.5377e-13
7	0.700	1.4392e+01	1.4392e+01	1.4392e+01	1.4392e+01	1.5047e-04	3.9080e-05	3.2091e-13
8	0.800	1.4592e+01	1.4592e+01	1.4592e+01	1.4592e+01	1.1369e-04	3.1867e-05	8.5212e-14
9	0.900	1.4727e+01	1.4727e+01	1.4727e+01	1.4727e+01	8.4949e-05	2.5172e-05	1.2062e-14
10	1.000	1.4817e+01	1.4817e+01	1.4817e+01	1.4817e+01	6.2885e-05	2.4425e-05	0.0000e+00

Erro Relativo Máximo [%] 4.3513e-04 2.9939e-04 4.5377e-13

APÊNDICE D - Amostras de Resultados do Problema 2

Bulirsch-Stoer; Dormand-Prince; Runge-Kutta de quarta ordem

h = 0.01250
 Tempo_RK4 [s] = 0.00110
 Tempo_RKF [s] = 0.00200
 Tempo_BS [s] = 0.00472

n	t	Elevação	Bulirsch-Stoer		Dormand-Prince		Runge-Kutta: ordem 4	
0	0.000	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00
1	0.013	4.9784e-01	8.2285e-02	8.9060e+00	8.2288e-02	8.9057e+00	8.3020e-02	8.9070e+00
2	0.025	4.0481e-01	1.7473e-01	5.3233e+00	1.7473e-01	5.3231e+00	1.7568e-01	5.3245e+00
3	0.038	2.4688e-01	2.1471e-01	1.3072e+00	2.1472e-01	1.3065e+00	2.1571e-01	1.3080e+00
4	0.050	1.3383e-01	2.1450e-01	-1.0674e+00	2.1452e-01	-1.0672e+00	2.1552e-01	-1.0672e+00
5	0.063	6.8014e-02	1.9366e-01	-2.0927e+00	1.9367e-01	-2.0926e+00	1.9467e-01	-2.0930e+00
6	0.075	3.3183e-02	1.6541e-01	-2.3382e+00	1.6543e-01	-2.3384e+00	1.6640e-01	-2.3392e+00
7	0.088	1.5740e-02	1.3674e-01	-2.2102e+00	1.3674e-01	-2.2102e+00	1.3772e-01	-2.2117e+00
8	0.100	7.3135e-03	1.1074e-01	-1.9377e+00	1.1074e-01	-1.9377e+00	1.1169e-01	-1.9395e+00
9	0.113	3.3451e-03	8.8419e-02	-1.6338e+00	8.8422e-02	-1.6337e+00	8.9350e-02	-1.6358e+00
10	0.125	1.5111e-03	6.9815e-02	-1.3481e+00	6.9818e-02	-1.3481e+00	7.0718e-02	-1.3504e+00
11	0.138	6.7582e-04	5.4565e-02	-1.0985e+00	5.4568e-02	-1.0985e+00	5.5439e-02	-1.1009e+00
12	0.150	2.9975e-04	4.2188e-02	-8.8823e-01	4.2191e-02	-8.8820e-01	4.3031e-02	-8.9071e-01
13	0.162	1.3202e-04	3.2207e-02	-7.1445e-01	3.2211e-02	-7.1444e-01	3.3019e-02	-7.1698e-01
14	0.175	5.7806e-05	2.4195e-02	-5.7238e-01	2.4198e-02	-5.7236e-01	2.4975e-02	-5.7491e-01
15	0.188	2.5181e-05	1.7787e-02	-4.5692e-01	1.7791e-02	-4.5692e-01	1.8536e-02	-4.5943e-01
16	0.200	1.0920e-05	1.2680e-02	-3.6343e-01	1.2684e-02	-3.6343e-01	1.3398e-02	-3.6591e-01
17	0.212	4.7173e-06	8.6262e-03	-2.8791e-01	8.6303e-03	-2.8791e-01	9.3135e-03	-2.9033e-01
18	0.225	2.0307e-06	5.4218e-03	-2.2700e-01	5.4255e-03	-2.2700e-01	6.0792e-03	-2.2936e-01
19	0.237	8.7151e-07	2.9019e-03	-1.7795e-01	2.9058e-03	-1.7796e-01	3.5303e-03	-1.8024e-01
20	0.250	3.7298e-07	9.3304e-04	-1.3851e-01	9.3681e-04	-1.3852e-01	1.5333e-03	-1.4073e-01

APÊNDICE E - Amostras de Resultados do Problema 3

Runge-Kutta: Quinta ordem

Bulirsch-Stoer; Dormand-Prince; Runge-Kutta de quarta ordem

h = [min] 10.000000
 Tempo_BS [s] = 0.011138
 Tempo_RKF [s] = 0.001491
 Tempo_RK4 [s] = 0.001374

t	Teórica	Bul.-Stoer	Dormand-Prince	R-K4	Erro_BS	Erro_DP	Erro_RK4
0.000	1.00000	1.00000	1.00000	1.00000	0.00000e+00	0.00000	0.00000
10.127	0.96108	0.96108	0.96334	0.96614	1.62877e-08	0.23511	0.52635
20.253	0.91523	0.91523	0.91749	0.92091	1.71199e-08	0.24681	0.62089
30.380	0.86792	0.86792	0.87020	0.87421	1.80531e-08	0.26247	0.72468
40.506	0.82065	0.82065	0.82298	0.82751	1.90940e-08	0.28398	0.83581
50.633	0.77419	0.77419	0.77651	0.78157	2.02400e-08	0.29971	0.95335
60.759	0.72898	0.72898	0.73126	0.73683	2.14952e-08	0.31193	1.07684
70.886	0.68531	0.68531	0.68756	0.69358	2.28650e-08	0.32802	1.20605
81.013	0.64334	0.64334	0.64561	0.65197	2.43565e-08	0.35180	1.34090
91.139	0.60319	0.60319	0.60546	0.61212	2.59781e-08	0.37738	1.48142
101.266	0.56489	0.56489	0.56717	0.57408	2.77394e-08	0.40426	1.62772
111.392	0.52847	0.52847	0.53075	0.53787	2.96511e-08	0.43278	1.77996
121.519	0.49392	0.49392	0.49620	0.50349	3.17252e-08	0.46232	1.93838
131.646	0.46121	0.46121	0.46349	0.47091	3.39748e-08	0.49355	2.10326
141.772	0.43031	0.43031	0.43258	0.44010	3.64144e-08	0.52774	2.27494
151.899	0.40117	0.40117	0.40344	0.41101	3.90600e-08	0.56606	2.45380
162.025	0.37372	0.37372	0.37599	0.38359	4.19289e-08	0.60872	2.64028
718.987	0.00398	0.00398	0.00625	0.00882	3.93936e-06	57.22918	121.70348
729.114	0.00363	0.00363	0.00591	0.00845	4.31141e-06	62.63353	132.55310
739.241	0.00332	0.00332	0.00560	0.00812	4.71952e-06	68.56201	144.44145
749.367	0.00303	0.00303	0.00531	0.00781	5.16728e-06	75.06654	157.47135
759.494	0.00277	0.00277	0.00505	0.00753	5.65862e-06	82.20465	171.75598
769.620	0.00253	0.00253	0.00480	0.00727	6.19786e-06	90.03906	187.41994
779.747	0.00231	0.00231	0.00458	0.00703	6.78978e-06	98.63849	204.60046
789.873	0.00211	0.00211	0.00438	0.00681	7.43962e-06	108.07881	223.44863
800.000	0.00192	0.00192	0.00420	0.00661	8.15317e-06	118.44440	244.13089
Erro Máximo [%]					8.15317e-06	118.44440	244.13089

APÊNDICE F - Solução Analítica do Problema 2

Problema 2: Sistema de suspensão de um veículo

Equação diferencial:

$$\ddot{x} + \frac{5.000}{240}\dot{x} + \frac{16.000}{240}x = \left(\frac{489.290 - 33.663.152 t}{240}\right)e^{-72 t}$$

$$\frac{dx}{dt}(0) = \dot{x}(0) = 0$$

$$x(0) = 0$$

Solução:

$$x(t) = \frac{270824229 \cdot \exp\left(\frac{5 \cdot t \cdot \left(241^{\frac{1}{2}} - 25\right)}{12}\right)}{791296900} - \frac{270824229 \cdot \exp(-72 \cdot t)}{395648450} + (270824229 \cdot \exp(-(241^{\frac{1}{2}} + 25)/12))/791296900 - (10519821 \cdot t \cdot \exp(-72 \cdot t))/281300 -$$

$$\frac{22585566393 \cdot 241^{\frac{1}{2}} \cdot \exp\left(\frac{5 \cdot t \cdot \left(241^{\frac{1}{2}} - 25\right)}{12}\right)}{953512764500} + (22585566393 \cdot 241^{\frac{1}{2}} \cdot \exp(-(5 \cdot t \cdot 241^{\frac{1}{2}} + 25)/12))/953512764500$$

APÊNDICE G - Solução Analítica do Problema 3

Problema 3: Equação de secagem de Page

$$y(t) = C3 - \frac{C1 * t^{C2+1} * \exp\left(-\frac{k * t^b}{2}\right) * whittakerM\left(-\frac{C2 - b + 1}{2 * b}, \frac{C2 + 1}{2 * b}, -k * t^b\right)}{((C2 + 1) * (-k * t^b)^{\frac{C2 + b + 1}{2 * b}})}$$

k, b = Parâmetros empíricos em função da temperatura e do tipo de material.

$$C1 = -k \cdot b; \quad C2 = b - 1$$

$whittakerM$ = Função M de Whittaker

Função M de Whittaker:

$M = whittakerM(a, b, x)$ calcula o valor da função M de Whittaker para os argumentos a, b, x (ABRAMOWITZ; STEGUN, 1964).