



Universidade Federal do Ceará
Centro de Ciências
Departamento de Computação

Métodos Numéricos I (CK0047) 2024.2

Carlos Alberto Sampaio Sales Junior - 485181

Gutemberg dos Santos Andrade - 493938

1.0 Introdução

Os métodos numéricos desempenham um papel crucial em diversas áreas da engenharia, sendo especialmente importantes na indústria aeroespacial, onde testes práticos podem ser extremamente caros e arriscados. Um dos desafios enfrentados é a modelagem do deslocamento da extremidade de um foguete espacial ao entrar na atmosfera terrestre, descrito pela equação $f(d) = a * d - d * \ln(d)$. Essa análise é vital para garantir a segurança e eficiência do projeto, pois deslocamentos superiores a 2 cm podem comprometer a integridade estrutural do material, resultando em falhas catastróficas.

Neste trabalho, são implementados três métodos numéricos — **Bisseção**, **Posição Falsa** e **Newton-Raphson** — para calcular as raízes da equação $f(d)$. Os métodos são comparados em termos de precisão, número de iterações e tempo de execução. O objetivo principal é desenvolver um sistema robusto que permita avaliar se o foguete projetado atende aos critérios de segurança, além de analisar o impacto de diferentes valores do parâmetro a sobre os resultados.

Este relatório apresenta a metodologia aplicada, os resultados obtidos e uma discussão detalhada sobre a eficiência de cada método, destacando suas vantagens e limitações no contexto do problema proposto.

Metodologia

A implementação do sistema para determinar as raízes da equação $f(d) = a * d - d * \ln(d)$ foi realizada em C++ de forma modular, com cada método numérico separado em arquivos específicos. O programa foi estruturado para receber parâmetros de entrada, executar os cálculos utilizando os métodos definidos e consolidar os resultados em tabelas comparativas. A seguir, detalha-se como cada componente do código foi utilizado. Nossa implementação opcionalmente gera arquivos csv com os quadros comparativos gerados via terminal para análise futura.

1. Estrutura do Código

1.1. Arquivo Principal (`main.cpp`)

O arquivo principal gerencia o fluxo do programa, recebendo os parâmetros de entrada (`aaa`, ϵ) e coordenando a execução dos métodos numéricos:

- **Parâmetros padrão:** Parâmetros padrão: $a=1$, intervalo inicial $[2,3]$ e $\epsilon=10^{-5}$
- Chamada sequencial dos métodos:
 - **Bisseção:** função `bisseca()`.
 - **Posição Falsa:** função `posição_falsa()`.
 - **Newton-Raphson:** função `newton_raphson()`.
- Armazenamento dos resultados em objetos `Payload`, que registram o histórico de iterações, valores estimados e erros.

1.2. Estruturas de Dados

- **Payload:** Agrega informações de todas as iterações de cada método, incluindo:
 - Número da iteração.
 - Estimativa atual da raiz (d).

- Valor da função $f(d)$.
- Erro absoluto entre iterações.
- **Epoch**: Representa os dados de uma única iteração, armazenados no **Payload**.

2. Métodos Numéricos Implementados

2.1. Método da Bissecção (**bissecacao.hpp**)

O método da Bissecção foi implementado para reduzir iterativamente o intervalo $[start, end]$, onde a raiz é localizada:

- A cada iteração, calcula-se o ponto médio $mid = (start + end) / 2$
- Dependendo do sinal de $f(mid) < 0$, o intervalo é ajustado:
 - Se $f(start) \cdot f(mid) < 0$, redefine-se $end = mid$.
 - Caso contrário, redefine-se $start = mid$.
- Os valores de mid , $f(mid)$ e o erro relativo são registrados em um **Epoch**.

2.2. Método da Posição Falsa (**posicao_falsa.hpp**)

Este método utiliza uma aproximação linear para localizar a raiz:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}.$$

- O intervalo é ajustado similarmente ao método da Bissecção.
- Em cada iteração, os dados são registrados no **Payload** para posterior análise.

2.3. Método de Newton-Raphson (**newton_raphson.hpp**)

O método de Newton-Raphson baseia-se na fórmula iterativa:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}.$$

- O programa avalia $f'(x)$ a partir de uma função auxiliar definida em **functions.hpp**.
- O ponto inicial é ajustado automaticamente, garantindo convergência rápida.

3. Comparação e Análise

Os resultados de cada método são armazenados no objeto `Payload` e exibidos no terminal como uma tabela comparativa e opcionalmente salvos como arquivo csv. Os dados incluem:

- Raiz (d) encontrada.
- Erro absoluto.
- Número de iterações.
- Tempo de execução.

4.0 Exemplos e Resultados

Realizamos 5 testes, com o objetivo de maximizar o valor de **a** sem que a raiz passasse de 2cm:

Type the desired amount of tests: 5

Type the value of a for test 1: .5

Type the tolerable error for test 1: 1e-5

bissecao()

K	d	f(d)	erro
T01	02.000000	-0.386294	02.000000
T02	01.500000	00.141802	00.500000
T03	01.750000	-0.104328	00.250000
T04	01.625000	00.023550	00.125000
T05	01.687500	-0.039231	00.062500
T06	01.656250	-0.007546	00.031250
T07	01.640625	00.008076	00.015625
T08	01.648438	00.000284	00.007812
T09	01.652344	-0.003626	00.003906
T10	01.650391	-0.001670	00.001953
T11	01.649414	-0.000693	00.000977

T12	01.648926	-0.000205	00.000488
T13	01.648682	00.000040	00.000244
T14	01.648804	-0.000082	00.000122
T15	01.648743	-0.000021	00.000061
T16	01.648712	00.000009	00.000031
T17	01.648727	-0.000006	00.000015

newton_raphson()

K	d	f(d)	erro
T01	01.765174	-0.120472	00.734826
T02	01.652399	-0.003682	00.112775
T03	01.648725	-0.000004	00.003674

posicao_falsa()

K	d	f(d)	erro
T01	01.435571	00.198737	00.435571
T02	01.591449	00.056266	00.155878
T03	01.634240	00.014418	00.042791
T04	01.645117	00.003600	00.010877
T05	01.647828	00.000893	00.002711
T06	01.648500	00.000221	00.000672
T07	01.648667	00.000055	00.000166
T08	01.648708	00.000014	00.000041
T09	01.648718	00.000003	00.000010

Type the value of a for test 2: .6

Type the tolerable error for test 2: 1e-5

bissecao()

K	d	f(d)	erro
T01	02.000000	-0.186294	02.000000
T02	01.500000	00.291802	00.500000
T03	01.750000	00.070672	00.250000
T04	01.875000	-0.053641	00.125000
T05	01.812500	00.009593	00.062500
T06	01.843750	-0.021759	00.031250
T07	01.828125	-0.006016	00.015625
T08	01.820312	00.001805	00.007812
T09	01.824219	-0.002101	00.003906
T10	01.822266	-0.000147	00.001953
T11	01.821289	00.000830	00.000977
T12	01.821777	00.000341	00.000488
T13	01.822021	00.000097	00.000244
T14	01.822144	-0.000025	00.000122
T15	01.822083	00.000036	00.000061
T16	01.822113	00.000006	00.000031
T17	01.822128	-0.000009	00.000015

newton_raphson()

K	d	f(d)	erro
T01	01.899276	-0.078769	00.600724
T02	01.823644	-0.001526	00.075632
T03	01.822119	-0.000001	00.001525

posicao_falsa()

K	d	f(d)	erro
T01	01.572564	00.231627	00.572564
T02	01.763962	00.057219	00.191398

T03	01.809501	00.012574	00.045539
T04	01.819425	00.002692	00.009924
T05	01.821546	00.000573	00.002121
T06	01.821997	00.000122	00.000451
T07	01.822093	00.000026	00.000096
T08	01.822113	00.000006	00.000020

Type the value of a for test 3: .7

Type the tolerable error for test 3: 1e-5

bissecao(): root surpassed the 2cm limit

newton_raphson(): root surpassed the 2cm limit

posicao_falsa(): root surpassed the 2cm limit

bissecao()

K	d	f(d)	erro
T01	02.000000	00.013706	02.000000
T02	02.500000	-0.540727	00.500000
T03	02.250000	-0.249593	00.250000
T04	02.125000	-0.114265	00.125000
T05	02.062500	-0.049333	00.062500
T06	02.031250	-0.017573	00.031250
T07	02.015625	-0.001873	00.015625
T08	02.007812	00.005931	00.007812
T09	02.011719	00.002033	00.003906
T10	02.013672	00.000081	00.001953
T11	02.014648	-0.000896	00.000977
T12	02.014160	-0.000407	00.000488
T13	02.013916	-0.000163	00.000244
T14	02.013794	-0.000041	00.000122
T15	02.013733	00.000020	00.000061
T16	02.013763	-0.000011	00.000031
T17	02.013748	00.000005	00.000015

newton_raphson()

K	d	f(d)	erro
T01	02.055430	-0.042105	00.444570
T02	02.014170	-0.000417	00.041260
T03	02.013753	-0.000000	00.000417

posicao_falsa()

K	d	f(d)	erro
T01	01.738460	00.255554	00.738460
T02	01.960586	00.052458	00.222126
T03	02.004267	00.009464	00.043680
T04	02.012085	00.001667	00.007818
T05	02.013460	00.000292	00.001375
T06	02.013701	00.000051	00.000241
T07	02.013744	00.000009	00.000042

Type the value of a for test 4: .69

Type the tolerable error for test 4: 1e-5

bissecao()

K	d	f(d)	erro
T01	02.000000	-0.006294	02.000000
T02	01.500000	00.426802	00.500000
T03	01.750000	00.228172	00.250000
T04	01.875000	00.115109	00.125000
T05	01.937500	00.055415	00.062500
T06	01.968750	00.024809	00.031250

T07	01.984375	00.009319	00.015625
T08	01.992188	00.001527	00.007812
T09	01.996094	-0.002380	00.003906
T10	01.994141	-0.000425	00.001953
T11	01.993164	00.000551	00.000977
T12	01.993652	00.000063	00.000488
T13	01.993896	-0.000181	00.000244
T14	01.993774	-0.000059	00.000122
T15	01.993713	00.000002	00.000061
T16	01.993744	-0.000028	00.000031
T17	01.993729	-0.000013	00.000015

newton_raphson()

K	d	f(d)	erro
T01	02.038668	-0.045456	00.461332
T02	01.994204	-0.000488	00.044464
T03	01.993716	-0.000000	00.000488

posicao_falsa()

K	d	f(d)	erro
T01	01.720312	00.253736	00.720312
T02	01.939769	00.053210	00.219458
T03	01.983876	00.009815	00.044107
T04	01.991947	00.001767	00.008071
T05	01.993399	00.000317	00.001451
T06	01.993659	00.000057	00.000260
T07	01.993705	00.000010	00.000047
T08	01.993714	00.000002	00.000008

Type the value of a for test 5: .6931

Type the tolerable error for test 5: 1e-5

bissecao()

K	d	f(d)	erro
T01	02.000000	-0.000094	02.000000
T02	01.500000	00.431452	00.500000
T03	01.750000	00.233597	00.250000
T04	01.875000	00.120921	00.125000
T05	01.937500	00.061422	00.062500
T06	01.968750	00.030912	00.031250
T07	01.984375	00.015470	00.015625
T08	01.992188	00.007703	00.007812
T09	01.996094	00.003808	00.003906
T10	01.998047	00.001858	00.001953
T11	01.999023	00.000882	00.000977
T12	01.999512	00.000394	00.000488
T13	01.999756	00.000150	00.000244
T14	01.999878	00.000028	00.000122
T15	01.999939	-0.000033	00.000061
T16	01.999908	-0.000003	00.000031
T17	01.999893	00.000012	00.000015

newton_raphson()

K	d	f(d)	erro
T01	02.043835	-0.044408	00.456165
T02	02.000371	-0.000465	00.043464
T03	01.999906	-0.000000	00.000465

posicao_falsa()

K	d	f(d)	erro

T01	01.725897	00.254316	00.725897
T02	01.946194	00.052983	00.220297
T03	01.990175	00.009707	00.043981
T04	01.998169	00.001736	00.007994
T05	01.999596	00.000309	00.001428
T06	01.999851	00.000055	00.000254
T07	01.999896	00.000010	00.000045

Como queríamos que a raiz fosse < 2 , mudamos o isolamento para $[1,3]$ e foi verificado que para $a = 0.6931$, com $\epsilon = 10^{-5}$, a raiz é o mais próximo possível de 2, mas não ultrapassa, podendo-se aproximar ainda mais com um epsilon menor. Os logs implementados no software nos dão informações cruciais sobre a convergência dos métodos, possíveis problemas e caso a raiz passe do limite estabelecido.

Também nesta seção, apresentamos os resultados obtidos para a solução do problema proposto utilizando os métodos implementados. Foram utilizados os seguintes parâmetros de entrada para os testes:

- Intervalo inicial: $[2,3]$;
- Parâmetro $a = 1$;
- Precisão desejada $\epsilon = 10^{-3}$.

bissecao(): root surpassed the 2cm limit
 newton_raphson(): root surpassed the 2cm limit
 posicao_falsa(): root surpassed the 2cm limit

bissecao()

K	d	f(d)	erro

T01	02.500000	00.209273	02.500000
T02	02.750000	-0.031903	00.250000
T03	02.625000	00.091663	00.125000
T04	02.687500	00.030607	00.062500

T05	02.718750	-0.000468	00.031250
T06	02.703125	00.015114	00.015625
T07	02.710938	00.007334	00.007812
T08	02.714844	00.003436	00.003906
T09	02.716797	00.001485	00.001953
T10	02.717773	00.000508	00.000977
T11	02.718262	00.000020	00.000488
T12	02.718506	-0.000224	00.000244
T13	02.718384	-0.000102	00.000122
T14	02.718323	-0.000041	00.000061
T15	02.718292	-0.000010	00.000031
T16	02.718277	00.000005	00.000015

newton_raphson()

K	d	f(d)	erro
T01	02.728392	-0.010129	00.228392
T02	02.718301	-0.000019	00.010091
T03	02.718282	-0.000000	00.000019

posicao_falsa()

K	d	f(d)	erro
T01	02.674741	00.043190	00.674741
T02	02.716177	00.002104	00.041436
T03	02.718181	00.000100	00.002004
T04	02.718277	00.000005	00.000096

Os métodos analisados foram o **Bisseção**, **Newton-Raphson** e **Posição Falsa**. A seguir, são detalhados os resultados obtidos. A raiz encontrada foi 2.7182 em todos os métodos.

1. Método da Bisseção

O método da Bisseção convergiu para uma solução correta dentro do isolamento mostrado, convergindo em 16 iterações e com a raiz encontrada, o foguete explode.

2. Método de Newton-Raphson

O método de Newton-Raphson também convergiu, porém em bem menos iterações que o método da bisseção, como esperado de um método rápido, em apenas 3 iterações. A raiz encontrada indica que o foguete explode.

3. Método da Posição Falsa

O método da Posição Falsa também convergiu, porém em bem menos iterações que o método da bisseção porém com mais iterações que o de Newton, como esperado de um método rápido, em apenas 4 iterações. A raiz encontrada também indica que o foguete explode.

Análise dos Resultados

- **Bisseção:** Este método demonstrou ser o mais confiável para o intervalo dado, por ser um método robusto de convergência garantida, convergindo de forma estável e atingindo a precisão desejada.
- **Newton-Raphson e Posição Falsa:** Ambos os métodos sucederam em fornecer uma solução válida, além de serem incrivelmente rápidos. Como estes não são robustos, a convergência não é garantida, o que se deve levar em consideração quando implementamos métodos no mundo real.

Os resultados sugerem que, para o problema em questão, o método da Bisseção é mais robusto e menos suscetível a erros relacionados à derivada ou à escolha do ponto inicial.