

RELATÓRIO EP3 - Laboratório de Métodos Numéricos

1. COMPUTANDO TRABALHO

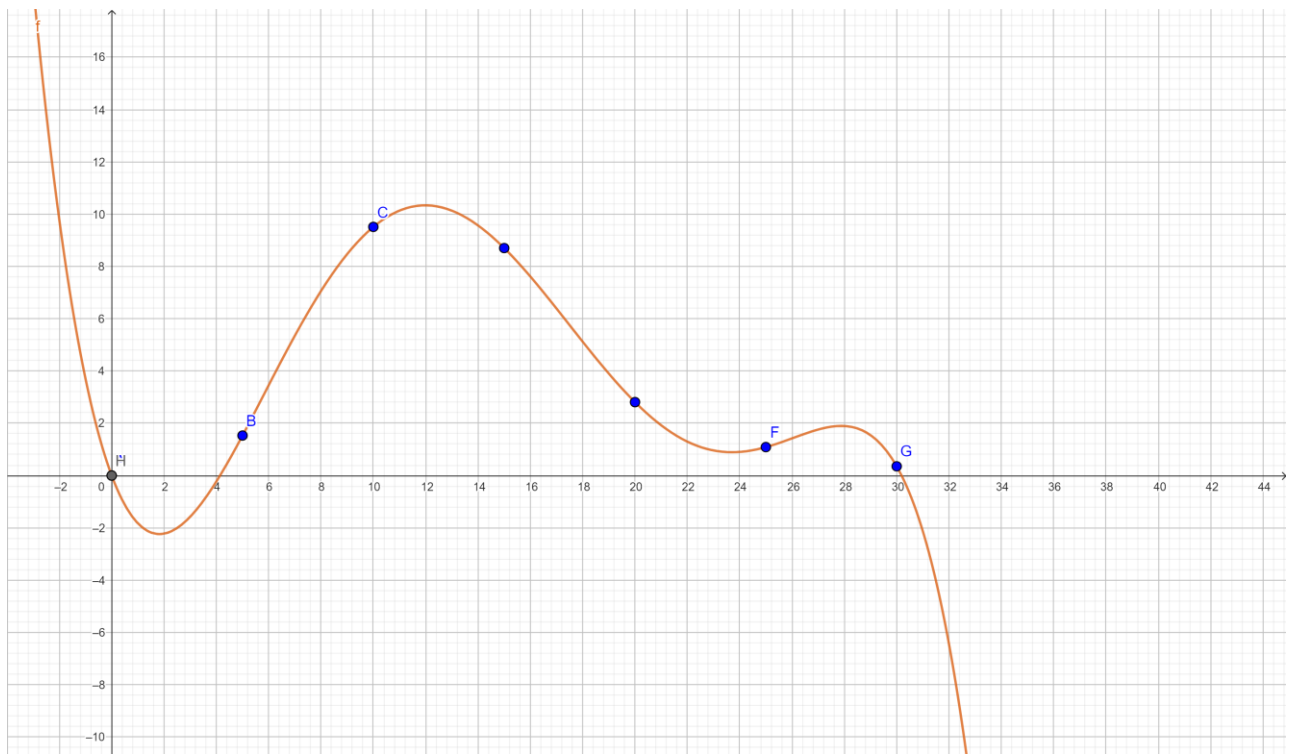
Ele também utiliza a interpolação por Lagrange para obter valores intermediários da função. O programa começa definindo constantes e bibliotecas necessárias. Em seguida, são declaradas as funções para calcular a integral pelos métodos do Trapézio e de Simpson, além da função psi utilizada na interpolação por Lagrange.

A função principal “main()” começa declarando variáveis necessárias, incluindo vetores para armazenar os dados. Os vetores “F”, “h” e “F_tempo” contêm os valores da função, do tamanho do passo e do tempo correspondente, respectivamente. Os vetores “dado_x” e tempo contêm os valores de “x” e “t”, que são utilizados para a interpolação.

O programa então entra em dois loops, um para calcular a integral usando o método do Trapézio e outro para o método de Simpson. Em cada iteração, chama a função “interpolacaoPorLagrange()” para obter os valores interpolados da função em um conjunto de pontos. Em seguida, utiliza o método correspondente para calcular a integral desses pontos. A cada iteração, o programa calcula o erro absoluto e o erro relativo em relação a um resultado analítico conhecido. Em seguida, imprime os resultados na tela.

1.1 INTERPOLAÇÃO

Foi realizada a interpolação por Lagrange nos pontos fornecidos na tabela da função $F(x) \cos(\theta(x))$. A interpolação gerou o polinômio de equação: $f(x) = 0 - 2.6862x + 0.9348x^2 - 0.0780x^3 + 0.0021x^4 - 5.1210 \cdot 10^{-6}x^5 - 4.0823 \cdot 10^{-7}x^6$, e assim, podemos calcular sua integral analiticamente: $\int_0^{30} f(x)dx = 117.13$. Segue a imagem abaixo do polinômio interpolante gerado no geogebra 2D:



1.2 RESULTADOS

Nos resultados abaixo, nota-se que a escolha de h influencia diretamente no erro, além disso, é possível concluir que o método de Simpson tem $Erro = O(h^5)$ e o método do Trapézio possui $Erro = O(h^3)$. Segue a imagem abaixo da saída do código com os resultados:

Integral pelo método do Trapézio:

ErroAbsoluto	ErroRelativo	I	h
3.88252765	3.31471668	121.012528	8.000000
-0.15566292	-0.13289757	116.974337	3.500000
-0.19299072	-0.16476626	116.937009	0.570000
-0.02998388	-0.02559880	117.100016	0.160000
0.00163887	0.00139919	117.131639	0.015000
0.00162337	0.00138595	117.131623	0.005000
0.00151530	0.00129369	117.131515	0.000900

Integral pelo método de Simpson:

ErroAbsoluto	ErroRelativo	I	h
-21.50887832	-18.36325307	95.621122	8.000000
-2.67492126	-2.28372002	114.455079	3.500000
-0.22984097	-0.19622724	116.900159	0.570000
-0.13094376	-0.11179353	116.999056	0.160000
0.00162143	0.00138430	117.131621	0.015000
0.00162143	0.00138430	117.131621	0.005000
0.00119572	0.00102085	117.131196	0.000900

2. INTEGRAÇÃO MONTE CARLO

O código em questão realiza cálculos de integração numérica utilizando o método de Monte Carlo para diferentes funções. O objetivo é estimar o valor da integral e calcular os erros relativos e absolutos em relação a resultados analíticos conhecidos. O programa começa definindo constantes, incluindo o valor de PI. Em seguida, são declaradas as três funções a serem integradas: f1, f2 e f3.

A função "calculandoPI()" é utilizada para estimar o valor de PI usando o método de Monte Carlo. Ela gera pontos aleatórios no plano cartesiano e verifica quantos desses pontos estão dentro do círculo unitário. Com base na proporção desses pontos, é possível aproximar o valor de PI.

A função "integracaoMonteCarlo()" realiza a integração numérica pelo método de Monte Carlo. Ela recebe como parâmetros os limites de integração, o número de pontos a serem gerados e a função a ser integrada. O método consiste em gerar pontos aleatórios dentro do intervalo e calcular a média dos valores da função nesses pontos, multiplicada pelo tamanho do intervalo.

A função “main()” é a função principal do programa. Ela declara variáveis necessárias, incluindo o número de pontos n , a integral calculada, e os erros relativos e absolutos. O programa então entra em uma série de loops para realizar os cálculos de integração para diferentes funções.

Para cada função, o programa executa o método de Monte Carlo com um número crescente de pontos, variando de 10 a $10^{(qtd_N-1)}$. O valor da integral calculada é armazenado na variável `integral`, e os erros relativos e absolutos são calculados em relação aos resultados analíticos conhecidos. Os resultados são então impressos na tela.

2.1 ESTIMANDO AS INTEGRAIS

1) $\int_0^1 \sin(x) dx$

Resultado analítico: $\int_0^1 \sin(x) dx = -\cos(x) \big|_0^1 \approx 0.4596$

Para um $N = 10^8$, temos um $Erro = 2 \cdot 10^{-6}$

I1:			
ErroAbsoluto	ErroRelativo	N	I
0.040410	8.790618	10^1	0.500108
-0.006314	-1.373520	10^2	0.453384
0.002624	0.570900	10^3	0.462322
0.002878	0.625969	10^4	0.462575
0.000850	0.184851	10^5	0.460547
0.000021	0.004461	10^6	0.459718
-0.000059	-0.012810	10^7	0.459639
0.000048	0.010477	10^8	0.459746

2)

$$\int_3^7 x^3 dx$$

Resultado analítico: 580 .

I2:			
ErroAbsoluto	ErroRelativo	N	I
236.294408	40.740415	10 ¹	816.294408
-12.418730	-2.141160	10 ²	567.581270
5.692397	0.981448	10 ³	585.692397
-3.783632	-0.652350	10 ⁴	576.216368
-2.409431	-0.415419	10 ⁵	577.590569
-0.394448	-0.068008	10 ⁶	579.605552
0.178119	0.030710	10 ⁷	580.178119
0.050127	0.008643	10 ⁸	580.050127

3) $\int_3^7 x^3 dx$

Resultado analítico: $\int_0^\infty e^{-x} dx = -e^{-x}|_0^\infty = 1$. Escolhendo um valor K, podemos calcular $\int_0^K e^{-x} dx$ onde $\epsilon = 10^{-8}$ e $E = \int_K^\infty e^{-x} dx < \epsilon$. Logo $\int_0^\infty e^{-x} dx - \int_0^K e^{-x} dx < \epsilon = 10^{-8} \rightarrow \int_K^\infty e^{-x} dx < \epsilon \rightarrow e^{-K} < 10^{-8} \rightarrow M < \log(10^8) \approx 18.4206$

Para um $N = 10^8$, temos $I \approx 1.0005$

I3:			
ErroAbsoluto	ErroRelativo	N	I
-0.989300	-98.930021	10 ¹	0.010700
0.388979	38.897935	10 ²	1.388979
0.316915	31.691517	10 ³	1.316915
0.003599	0.359914	10 ⁴	1.003599
0.001047	0.104705	10 ⁵	1.001047
-0.002066	-0.206601	10 ⁶	0.997934
0.000344	0.034420	10 ⁷	1.000344
0.000978	0.097754	10 ⁸	1.000978

4) Estimativa de π :

Seja $N = 10^8$, Foi obtido uma diferença percentual de $ErroRelativo = 0.004\%$ onde $I\pi \approx 3.141605$ e $\pi = 3.141592$

I4:

ErroAbsoluto	ErroRelativo	N	I
0.058407	1.859164	10^1	3.200000
0.018407	0.585924	10^2	3.160000
-0.061593	-1.960555	10^3	3.080000
-0.004793	-0.152555	10^4	3.136800
-0.005033	-0.160194	10^5	3.136560
-0.000453	-0.014408	10^6	3.141140
-0.000305	-0.009723	10^7	3.141287
0.000025	0.000799	10^8	3.141618

3. COMO COMPILAR

No terminal, digite:

```
gcc calcular_trabalho.c -o calcular_trabalho -lm  
gcc monte_carlo.c -o monte_carlo -lm
```