Métodos Numéricos - Integração Numérica utilizando Open MP

Gustavo Aparecido de Souza Viana

***Abstract*— Métodos númericos são utilizados a fim de aproximar a solução de um determinado problema, como por exemplo resolução de integral, derivada e entre outros. De contra partida, esse método de resolução é mais custoso do que soluções fechadas. Neste trabalho será abordado a implementação da resolução de Integral utilizando técnicas de Integração Numérica. Os resultados mostraram que é possível estimar o valor da integral, mas dependendo do quantidade de divisões aplicadas a integral que é utilizado pode ser que a resposta não apresente a acurácia desejada.**

I. INTRODUÇÃO

Métodos númericos são utilizados em diversas áreas como por exemplo em redes neurais para saber se a rede converge ou não, e para resolver problemas computacionais mas não de forma exata, ou seja, métodos númericos chegam a solução desejada com uma taxa de erro, assim o usuário minimizará o erro para que seja irrelevante.

Métodos númericos pelo fato de serem métodos de otimização, ou seja, necessitam de diversas iterações para se aproximarem da solução desejada. Ou seja, a cada iteração irá se aproximando da solução final, consequentemente são métodos que possuem um alto custo computacional.

Neste trabalho foi abordado a implementação de métodos de Integração Numérica com objetivo de resolver integral, as técnicas utilizadas foram: Ponto Médio, Trapezoidal e Simpson.

II. CONCEITOS FUNDAMENTAIS

Nesta seção, será apresentado os conceitos básicos de Quadratura Numérica, Ponto Médio ou Retângulos, Trapezoidal e Simpson utilizados na integração numérica. E além disso os conceitos de Open MP.

A. Quadratura Numérica

O método consiste em aproximar a resolução da integral, por meio de otimização, ou seja, iterar N vezes até que o método utilizado (Ponto Médio, Trapezoidal ou Simpson) supra o erro máximo desejado. Representação da Quadratura Numérica na Equação 1.

$$\int_a^b f(x)dx \approx \sum_{i=0}^{n} \alpha_i f(x_i) \tag{1}$$

B. Ponto Médio ou Retângulos

Na Figura 1 podemos ver que o valor da integral é aproximada por meio de divisões em formatos de retângulos. A Equação 2 representa a fórmula para integrar e a Equação 3 a fórmula para aproximar o erro.

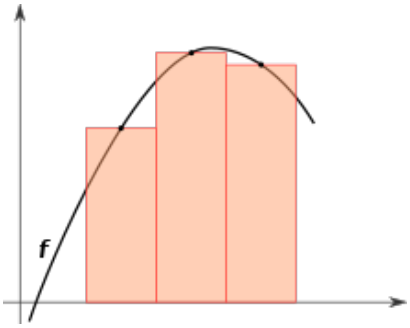

Fig. 1: Representação da Regra Ponto Médio ou Retângulos

$$\int_a^b f(x)dx \approx (b-a)f(\frac{a+b}{2}) \tag{2}$$

$$\frac{(\frac{b-a}{n})^3}{24} f''(\frac{b-a}{2}) \tag{3}$$

C. Trapezoidal

Na Figura 2 podemos ver que o valor da integral é aproximada por meio de divisões em formatos de trapézios. A Equação 4 representa a fórmula para integrar e a Equação 5 a fórmula para aproximar o erro.

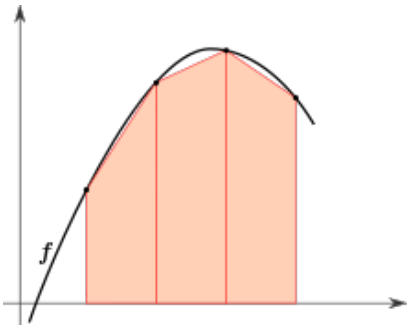

Fig. 2: Representação da Regra Trapezoidal

$$\int_a^b f(x)dx \approx (b-a)\frac{f(a)+f(b)}{2} \tag{4}$$

$$\frac{-(\frac{b-a}{n})^3}{12} f''(\frac{b-a}{2}) \tag{5}$$

D. Simpson

Na Figura 3 podemos ver que o valor da integral é aproximada por meio de divisões em formatos curvos mais precisos. A Equação 6 representa a fórmula para integrar e a Equação 7 a fórmula para aproximar o erro.

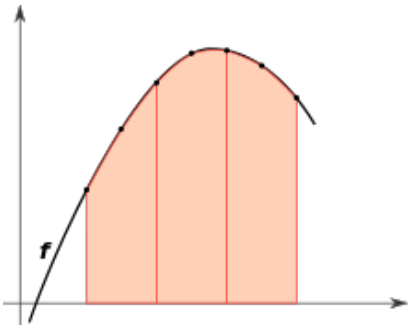

Fig. 3: Representação da Regra de Simpson

$$\int_a^b f(x)dx \approx (b-a)\frac{f(a)+4f(\frac{a+b}{2})+f(b)}{6} \quad (6)$$

$$\frac{-(\frac{b-a}{n})^5}{90}f''''(\frac{b-a}{2}) \quad (7)$$

E. Open MP

Open MP é utilizado para desenvolver aplicações paralelas de memória compartilhada onde possam ser executadas em qualquer plataforma, indo desde computadores comuns até supercomputadores. Ela é construída por diretivas de compilador, ou seja, existem tags utilizadas para demarcar regiões que serão executadas de forma paralela (*pragma*).

Atualmente podemos ter um modelo híbrido utilizando MPI e Open MP para sistemas distribuídos. No trabalho proposto o Open MP foi utilizado com intuíto de paralelisar o cálculo da quadratura numérica, onde cada região de interesse é executada de forma paralela assim mininuindo o tempo de execução. Mas é importante resaltar que nem sempre a utilização de paralelismo pode ter maior eficácia do não uso.

III. METODOLOGIA

Nesta seção será apresentado a metodologia utilizada para implementar regra de Ponto médio, Trapezoidal e Simpson utilizando Quadratura Numérica, implementados em C++. O código fonte pode ser encontrado em https://github.com/apparecidoo/ProgramacaoCientifica.

Grande maioria dos métodos utilizaram função como parâmetros, para isso foi utilizado o método *bind* para passar a função como parâmetro e para receber a classe *function* da biblioteca do C++ chamada *functional.h*.

Para cada técnica, foram implementados dois métodos com a responsabilidade de calcular a integral e o outro calcular a taxa de erro. Ambos métodos seguem as fórmulas citadas na sessão de Conceitos Fundamentais.

O método para calcular a integral, chamado de ”midpoint” ou ”trapezoidal” ou ”simpson”, recebe somente um parâmetro, consiste em ser a função que deverá ser integrada.

O método para calcular o erro, chamado de ”midpoint_error” ou ”trapezoidal_error” ou ”simpson_error”, recebe dois parâmetros, sendo que o primeiro consiste em ser a função que deverá ser integrada e o segundo na quantidade de divisões que a região de integração deverá ser dividida.

O método que implementa a Quadratura Numérica chamado de ”numeric_square_by_divisions_distributed” que recebe quatro parâmetros, o primeiro sendo a função de integração, o segundo a região de integração, e o terceiro o método numérico e por fim, a quantidade de divisões que será feita na região de integração. Como podemos ver no pseudocódigo 1.

Algorithm 1: numeric_square_by_divisions_distributed(Func f, IntegrationRange range, Func method, int divisions)

```
nprocs = get_num_processors()
set_num_threads(nprocs)
// For in parallel
while count < divisions do
    interval_a = (count * (range.b - range.a) / divisions)
     + range.a
    interval_b = ((count + 1) * (range.b - range.a) /
     divisions) + range.a
    result += numeric_method_func(f, new
     IntegrationRange(interval_a, interval_b))
end
return result
```

Outro método que implementa a Quadratura Numérica chamado de ”numeric_square_by_error” usa como referência erro máximo que a técnica utilizada deverá apresentar para achar a solução final. O método recebe cinco parâmetros, o primeiro sendo a função de integração, o segundo a região de integração, o terceiro o método numérico, o quarto a função de cálculo de erro, e o quinto sendo o erro máximo. Como podemos ver no pseudocódigo 2.

Algorithm 2: numeric_square_by_error_rate_distributed(Func f, Func method, IntegrationRange range, Func error_method, double error_rate)

```
divisions = 1
error = numeric_error(f, divisions)
// For in parallel
while error > error_rate do
    error = numeric_error(f, divisions)
    divisions++;
end
result = numeric_square_by_divisions_distributed(f,
 range, numeric_method, divisions)
return result
```

IV. EXPERIMENTOS E RESULTADOS

Nesta seção será apresentado os experimentos e seus respectivos resultados.

A Figura 4 representa os resultados contendo o valor de integração, erro e quantidade de divisões na região de integração, representando a Equação 8, Equação 9 e Equação 10 respectivamente. O experimento feito utilizou uma 1 milhão de divisões na região de integração.

$$\int_0^1 e^x dx \quad (8)$$

$$\int_0^1 \sqrt{1-x^2}dx \quad (9)$$

$$\int_0^1 e^{-x^2} dx \tag{10}$$

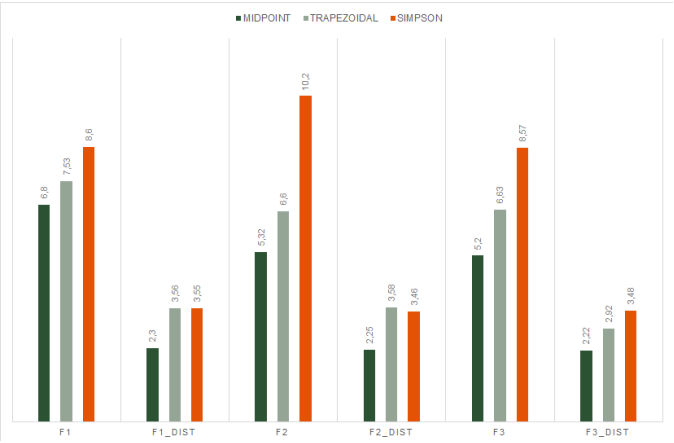

Fig. 4: Representação do tempo de execução em segundos para cada Equação e seus respectivos métodos

V. CONCLUSÃO

Após a análise do conceito e resultados obtidos após a aplicação do métodos comparando os mesmos, podemos concluir que todas as técnicas utilizadas conseguiram atingir o objetivo de integração e que dependendo do número de divisões que é aplicada na região de integração é obtido uma precisão com uma maior acurácia. Além disso, com a utilização do Open MP, é altamente notável que o tempo de execução diminui drasticamente, principalmente com o método de Simpson.

REFERENCES