Implementação e Análise de Métodos Numéricos com OpenMP: Newton-Cotes

Lucas Fontes Buzuti
Departamento de Engenharia Elétrica
Centro Universitário FEI
São Bernardo do Campo-SP, Brasil
lucas.buzuti@outlook.com

***Resumo*—Este artigo tem uma finalidade acadêmica na implementação e análise de métodos numéricos com a utilização de programação paralela usando as fórmulas de Newton-Cotes. As fórmulas usadas para analisar a performance da programação paralela foram: *Regras do Ponto Médio ou das Retângulos*, *Regra Trapezoidal* e *Regra de Simpson*, sendo que também foram adotadas funções para testar a implementação. Na implementação foi utilizada a programação orientada a objeto na linguagem *C++*, tendo em foco a otimização e a velocidade na execução dos métodos numéricos ao utilizar programação paralela.**

***Index Terms*—métodos numéricos, Newton-Cotes, programação paralela, programação orientada a objeto, C++**

I. Introdução

Este artigo tem em seu objetivo a implementação e análise de métodos numéricos com a utilização de programação paralela usando as fórmulas de Newton-Cotes. A implementação tem como alvo a utilização da linguagem *C++* utilizando o *OpenMP* (Open Multi-Processing), no qual é focada na otimização e na velocidade da execução de algoritmos, métodos numéricos, simulações e outros [1] [3].

Em análise numérica há um grande conjunto de algoritmos cujo principal objetivo é a de estimar o valor numérico da integral de uma função definida, com um grau de precisão pré-definido. A integral numérica consiste em encontrar um número que se aproxima da área de uma função. Nem todas as funções admitem uma primitiva de forma explícita, sendo que pode ser muito complicado avaliá-la, ou seja, pode ser mais fácil usar a integração numérica do que fazer a integração simbólica.

As características básicas de programação paralela com *OpenMP* são: um programa é iniciado em um único computador com diversos processadores (núcleos) diferentes, no entanto uma memória única é compartilhada pelos processadores, e o programa é compilado com comandos que verificam quais partes serão paralelizadas. O *OpenMP* torna-se eficiente devido a realização exclusivamente de programas paralelos através de *threads*, uma vez que a execução de um *thread* é uma pequena unidade de processamento que pode ser programado por um operador de sistema.

II. Teoria

Em métodos numéricos, as fórmulas de Newton-Cotes, ou Regras de Quadratura de Newton-Contes, são um conjuntos de fórmulas para integração numérica baseadas na avaliação do integrante em pontos igualmente espaçados.

As fórmulas de Newton-Cotes podem ser úteis se o valor do integrante, em pontos igualmente espaçados, é fornecido. Se for possível ter uma troca de pontos, em que o integrante é avaliado, então outros métodos podem ser mais adequados.

O método básico envolvido na aproximação numérica de integração é denominado de *Quadratura Numérica* e consiste em aproximar a função $f(x)$, definida entre $[a,b]$ e conhecidas, em pontos x_i entre $i = 0, \ldots, n$ igualmente espaçados, em que $x_0 = a$ e $x_n = b$ [2], de tal forma que a aproximação é definida sendo:

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

onde α_i são os coeficientes reais. Pode-se dividir o processo de integração numérica em três fases: decompondo o domínio de integração em sub-intervalos, realizando a integração aproximada da função de cada pedaço e somando os resultados numéricos obtidos.

As fórmulas de Newton-Cotes utilizadas são *Regras do Ponto Médio ou das Retângulos*, *Regra Trapezoidal* e *Regra de Simpson*, e podem ser descritas como [2]:

$$\int_a^b f(x)dx \simeq (b-a) f\left(\frac{a+b}{2}\right), \tag{2}$$

$$\int_a^b f(x)dx \simeq (b-a)\frac{f(a)+f(b)}{2}, \tag{3}$$

$$\int_a^b f(x)dx \simeq (b-a)\frac{f(a)+f\left(\frac{a+b}{2}\right)+f(b)}{6}, \tag{4}$$

respectivamente. O cálculo do erro de aproximação é a diferença entre o valor real da integral, para o valor aproximado. Os erros das fórmulas de Newton-Cotes já é conhecida da literatura [2], e podem ser descritas como:

$$\frac{(b-a)^3}{24} f^{(2)}(\zeta), \tag{5}$$

$$-\frac{(b-a)^3}{12} f^{(2)}(\zeta), \tag{6}$$

$$-\frac{(b-a)^5}{2880} f^{(4)}(\zeta), \tag{7}$$

onde ζ está entre a e b. As Equações 5, 6 e 7 representam as equações de erro das respectivas fórmulas: *Regras do Ponto Médio ou das Retângulos*, *Regra Trapezoidal* e *Regra de Simpson*. Pode-se estimar o erro mudando o tamanho do intervalo, esse método se denomina quadratura adaptativa.

III. Proposta e Implementação

Este artigo propõe utilizar a linguagem *C++* com *OpenMP* para a implementação[1] dos métodos numéricos usando as fórmulas de Newton-Cotes com o método de quadratura adaptativa e computar o tempo de processamento. Esta implementação foi feita em um computador com sistema operacional *Linux*, com o compilador *G++*, a versão da linguagem utilizada foi a *C++11*. Para desenvolver os método numéricos, foi utilizada a programação orientada a objeto (POO), onde visa a construção de classes e métodos.

Foi codificada uma classe denominada *NewtonCotes*. Para avaliar a implementação computou-se três funções: $\int_0^1 e^x \ dx$, $\int_0^1 \sqrt{1-x^2} \ dx$ e $\int_0^1 e^{-x^2} \ dx$.

IV. Experimentos e Resultado

Para analisar as fórmulas de Newton-Cotes usando programação paralela, foi computado o valor de tempo de processamento de cada integral para comparar com os valores de tempo da computação sem a programação paralela. As funções $\int_0^1 e^x \ dx$, $\int_0^1 \sqrt{1-x^2} \ dx$ e $\int_0^1 e^{-x^2} \ dx$ foram utilizadas para obter os resultados das integrais e calcular o tempo de processamento delas. A Tabela IV ilustra os tempos de processamentos sem *OpenMP* e a Tabela II com *OpenMP*.

Tabela I
Tempos de processamentos sem *OpenMP* (sec).

	Retângulos	*Trapezoidal*	*Simpson*
$\int_0^1 e^x \ dx$	$2.86 \ 10^{-7}$	$7.37 \ 10^{-7}$	$3.76 \ 10^{-7}$
$\int_0^1 \sqrt{1-x^2} \ dx$	$4.01 \ 10^{-7}$	$2.51 \ 10^{-7}$	$1.93 \ 10^{-7}$
$\int_0^1 e^{-x^2} \ dx$	$1.66 \ 10^{-7}$	$1.96 \ 10^{-7}$	$1.63 \ 10^{-7}$

Comparando os valores com *OpenMP* e sem *OpenMP* nas Tabelas e , pode-se concluir que: utilizar programação paralela torna o programa mais eficaz e rápido.

[1] https://github.com/buzutilucas/scientific-programming/tree/master/Ex08

Tabela II
Tempos de processamentos com *OpenMP* (sec).

	Retângulos	*Trapezoidal*	*Simpson*
$\int_0^1 e^x \ dx$	$1.95 \ 10^{-7}$	$3.38 \ 10^{-7}$	$1.29 \ 10^{-7}$
$\int_0^1 \sqrt{1-x^2} \ dx$	$1.4 \ 10^{-7}$	$1.48 \ 10^{-7}$	$1.59 \ 10^{-7}$
$\int_0^1 e^{-x^2} \ dx$	$1.59 \ 10^{-7}$	$1.57 \ 10^{-7}$	$1.22 \ 10^{-7}$

V. Conclusão

Neste artigo pode-se implementar e analisar os métodos numéricos em programação paralela usando as fórmulas de Newton-Cotes. Conclui-se que, utilizar programação paralela torna a execução do programa mais eficiente e rápido, assim aumentando a performance de qualquer algoritmo.

Referências

[1] Teukolsky, S. A., Flannery, B. P., Press, W. H., Vetterling, W. T. (1992). Numerical recipes in C. SMR, 693(1), 59-70.

[2] BIANCHI, Reinaldo. Programação Científica 5a Aula. 2019 5 - 18 slides.

[3] Quinn Michael, J. "Parallel Programming in C with MPI and OpenMP"Cap.17 McGraw-Hill Inc. 2004.