Primeiro Trabalho de Implementação (2015/2)
Problema de Quadratura para Integração Numérica

Computação Concorrente (MAB-117)
Prof. Silvana Rossetto

[1]DCC/IM/UFRJ
24 de novembro de 2015

1. Integração numérica

A *integral definida* é uma ferramenta essencial em diversas áreas para definir quantidades como áreas, volumes, pesos, probabilidades, etc.. Entretanto, existem situações nas quais é impossível encontrar o valor exato de uma integral. Uma delas vem do fato de que para calcularmos a integral de uma função $\int_a^b f(x)\,\mathrm{d}x$ usando o Teorema Fundamental do Cálculo precisamos conhecer uma antiderivada de f, o que nem sempre é trivial, por exemplo quando $f(x) = \sqrt{1+x^3}$. Outra situação surge quando a função é determinada por dados coletados de um experimento científico ou outro tipo de observação, e nesse caso pode não existir uma fórmula para a função. Em ambas as situações precisamos encontrar valores aproximados para as integrais (em um dado intervalo) usando algum *método de integração numérica*.

Existem vários métodos de integração numérica. A Figura 1 ilustra o método de aproximação retangular com ponto médio.

Dada uma função contínua e positiva **f(x)** e dois valores limites **l** e **r**, onde $l < r$, o problema consiste em computar a área limitada pela função f(x) entre o eixo **x** e as linhas verticais computadas por **l** e **r** (aproximação da integral de f(x) de l a r). Uma maneira de calcular a área abaixo da curva é dividir o intervalo [l,r] em uma série de subintervalos, e então usar o cálculo aproximado da área desses subintervalos (aproximando-os para retângulos) para encontrar a área total aproximada.

1.1. O problema da quadratura para integração numérica

Sempre que usamos uma técnica de aproximação, uma questão crucial é o quanto a aproximação é precisa. Nesse caso, o resultado computado terá um **erro** definido como a

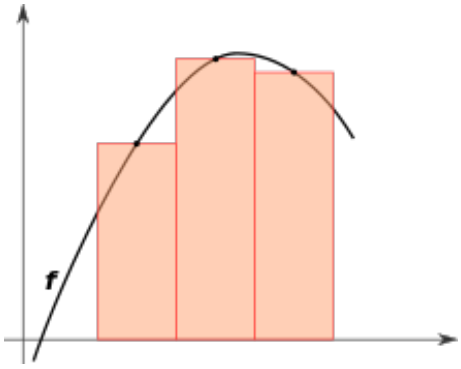

Figura 1. Aproximação por retângulos (ponto médio).

diferença entre a aproximação obtida pela regra e o valor real da integral definida, e dependerá do número de subintervalos N em que o intervalo inicial [a,b] foi dividido. Uma alternativa é não fixar N a priori e construir uma solução dinâmica chamada **quadratura adaptativa**.

Para o caso do **método de integração numérica retangular**, o problema começa com o intervalo inicial [a,b] e computa-se o ponto do meio **m**. Calcula-se a área dos três retângulos (usando o valor da função no ponto médio do subintervalo), compara-se a área do maior retângulo com a soma das áreas dos retângulos menores, se elas forem suficientemente próximas (diferença <= erro máximo), considera-se a área do maior retângulo como uma aproximação aceitável da área de **f**. Caso contrário, o processo é repetido para resolver os dois subproblemas de computar as áreas de [a,m] e [m,b]. Esse **processo é repetido recursivamente** até a solução de cada subproblema convergir.

2. Implementação do problema

Você deverá implementar o **método de integração numérica retangular (com ponto médio)** usando a estratégia de *quadratura adaptativa* para definir os subintervalos de forma dinâmica.

Duas versões deverão ser implementadas: uma **sequencial** e outra **concorrente** A versão concorrente deverá permitir que o número de threads em cada execução da aplicação seja passado como parâmetro, podendo variar de 1 a 8. Na versão concorrente, a solução deverá garantir **balanceamento de carga entre as threads** para obter o máximo de desempenho.

Os tempos de execução de cada solução (sequencial e concorrente) deverão ser medidos e comparados para um conjunto de casos de teste (funções de entrada).

Os programas deverão ser codificados nas linguagens **C**, usando a biblioteca PThreads e os mecanismos de sincronização estudados até aqui (*locks* e variáveis de condição).

2.1. Entrada

Para cada aplicação de teste (uma para cada função de entrada), o usuário deverá informar na linha de comando o **intervalo de integração [a,b]** e o **erro máximo** e. Para a versão concorrente, o usuário deverá informar também o **número de threads**.

As seguintes funções $f(x)$ deverão ser implementadas e pelo menos **três** delas deverão ser incluídas nos casos de teste reportados no relatório final:

(a) $f(x) = 1 + x$
(b) $f(x) = \sqrt{1 - x^2}, -1 < x < 1$
(c) $f(x) = \sqrt{1 + x^4}$
(d) $f(x) = sen(x^2)$
(e) $f(x) = cos(e^{-x}) * (0.005 * x^3 + 1)$

Dica: veja no site `http://www.wolframalpha.com` *o gráfico dessas funções e o valor aproximado da integral para um dado intervalo.*

2.2. Saída

A aplicação deverá retornar como saída o valor da integral da função no intervalo dado e o tempo de execução.

3. Etapas do trabalho

A execução do trabalho deverá ser organizada nas seguintes etapas:

1. Compreender o problema, pensar e esboçar uma solução sequencial e concorrente;
2. Projetar as estruturas de dados e algoritmos sequencial e concorrente;
3. Construir um conjunto de casos de teste para avaliação das soluções e projetar a aplicação de teste;
4. Implementar as soluções projetadas, avaliar a corretude, refinar a implementação e refazer os testes;
5. Avaliar o ganho de desempenho obtido com a versão concorrente, considerando diferentes casos de teste;
6. Redigir os relatórios e documentar os códigos.

4. Artefatos que deverão ser entregues

- **Relatório**: documentação do projeto das soluções (esboço das estruturas de dados e lógica dos algoritmos da aplicação), testes realizados e avaliação de desempenho. O relatório deverá conter informações suficientes para o professor compreender a solução proposta (sem precisar recorrer ao código fonte do programa) e avaliar o ganho de desempenho obtido com a versão concorrente.
- **Código fonte**: versões sequencial e concorrente e aplicação de teste.

5. Data de entrega e avaliação

O trabalho poderá ser feito em **dupla** e deverá ser entregue até o dia **17 de dezembro**.

Os seguintes itens serão avaliados no trabalho com o respectivo peso:

- Relatório (incluindo projeto das soluções, testes realizados e avaliação de desempenho): **2 pontos**
- Modularidade, organização e documentação do código fonte: **2 pontos**
- Compilação e execução correta das duas versões da solução: **4 pontos**
- Ganho de desempenho obtido e avaliação geral do trabalho: **2 pontos**

Os alunos integrantes da equipe poderão ser chamados pela professora para apresentar/explicar o trabalho.