Segundo Exercício-Programa

Norton Trevisan Roman

24 de maio de 2018

1 Integração Numérica

Dada uma determinada curva, definida por uma equação $f(x)$, a área sob essa curva, entre dois pontos bem definidos, pode ser calculada por meio da integral dessa curva dentro desses limites.

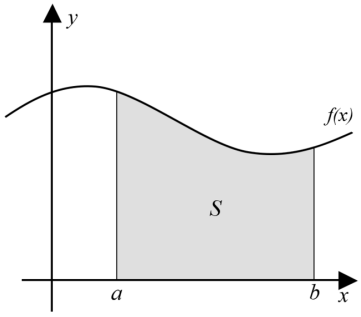

Figura 1: Integral: a área sob uma curva.

Nesse caso, diz-se que

$$S = \int_a^b f(x)dx$$

Embora haja métodos para o cálculo algébrico da integral de uma quantidade considerável de funções, nem sempre é possível calcular esse valor. Por vezes, seu cálculo torna-se algo tão complexo que aproximações são necessárias. Neste trabalho, você irá implementar uma das fórmulas de Newton–Cotes. Mais especificamente, trataremos da Regra dos Trapézios.

A regra dos trapézios trata de aproximar a área sob uma determinada curva $f(x)$ por uma reta, conforme abaixo:

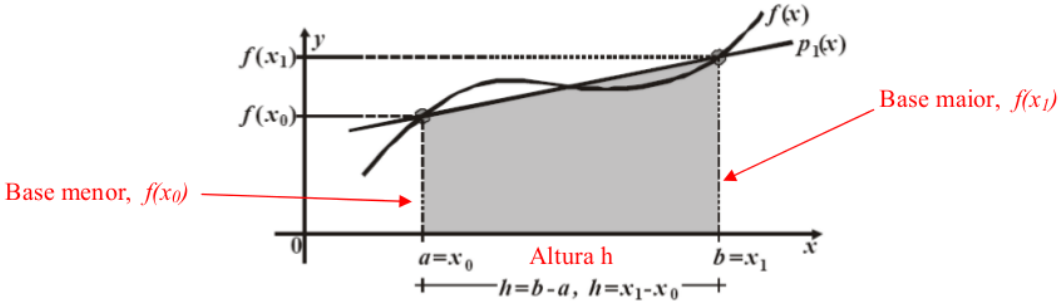

Figura 2: Integral: aproximação com um trapézio.

Assim, a área sob a curva é aproximada pela área do trapézio, ou seja

$$S = \frac{f(x_0) + f(x_1)}{2} \times (x_1 - x_0)$$

Naturalmente, há um erro considerável nisso, que corresponde à área do trapézio que ficou acima ou abaixo de $f(x)$. A solução para esse problema é, então, aumentar o número de trapézios usados:

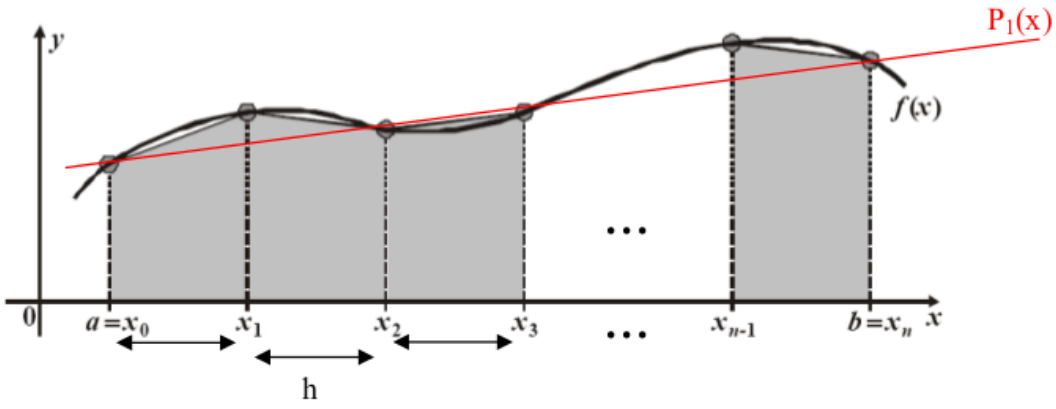

Figura 3: Integral: aproximação com vários trapézios.

A área resultante será então a soma de todas as áreas dos trapézios:

$$S = \frac{f(x_0) + f(x_1)}{2} \times (x_1 - x_0) + \frac{f(x_1) + f(x_2)}{2} \times (x_2 - x_1) + \ldots + \frac{f(x_{n-1}) + f(x_n)}{2} \times (x_n - x_{n-1})$$

ou

$$S = \sum_{i=1}^{n} \frac{f(x_{i-1}) + f(x_i)}{2} \times (x_i - x_{i-1})$$

Note, contudo, que dividimos a área entre $a = x_0$ e $b = x_n$ em n pedaços iguais. Temos então que cada intervalo tem, de fato, o tamanho $h = \frac{b-a}{n}$. Além disso, o valor de x_1 pode ser escrito como $x_1 = x_0 + h$, o de x_2 como $x_2 = x_1 + h = x_0 + h + h = x_0 + 2 \times h$, o de x_3 como $x_3 = x_2 + h = x_0 + 2 \times h + h = x_0 + 3 \times h$, e assim por diante. De um modo geral,

podemos dizer que x_i pode ser escrito como $x_i = x_0 + i \times h$. Como $h = x_i - x_{i-1}$, para um i qualquer, temos que

$$S = \sum_{i=1}^{n} \frac{f(x_{i-1}) + f(x_i)}{2} \times (x_i - x_{i-1}) = \sum_{i=1}^{n} \frac{f(x_{i-1}) + f(x_i)}{2} \times h$$

Com alguma manipulação algébrica, pode-se chegar a

$$\boxed{S = \frac{h}{2} \times \left[f(x_0) + f(x_n) + 2 \times \sum_{i=1}^{n-1} f(x_i) \right]} \quad (1)$$

onde h é a largura de cada trapézio usado, ou seja, $h = \dfrac{b-a}{n}$, n é o número de divisões do intervalo $[a, b]$, $x_0 = a$ e $x_n = b$.

2 Tarefa

Você receberá três classes, devendo implementar o método `resolve(Funcao f, double a, double b, int n)`, da classe `Integral`. As classes recebidas são:

- `Integral`, responsável pelo cálculo da integral de uma determinada função, usando a regra do trapézio. Essa classe possui o método `static double resolve(Funcao f, double a, double b, int n)`, que deve ser implementado por você. Esse método recebe como parâmetros uma função f (objeto da classe `Funcao`), um intervalo $[a, b]$, dentro do qual será calculada a integral da função, além de n, que corresponde ao número de divisões desse intervalo. A partir desses parâmetros, o método deve retornar a área da curva (definida pela função), dentro desse intervalo, conforme a Fórmula 1 acima, ou -1 caso $n \leq 0$ (nesse caso há sim uma ambiguidade, em que não se sabe se -1 é a área ou erro. Isso não será problema para a correção do trabalho). A função a ser usada no cálculo estará implementada na classe `Funcao`, sendo seu valor obtido por meio de seu método `double valor(double ponto)`. **Modifique apenas o corpo desse método** para seus testes, sem mudar sua assinatura. É permitida a inclusão de métodos auxiliares na classe `Integral`, caso precise. **Não crie um main para essa classe**.

- `Funcao`, responsável por fornecer a função sobre a qual o método será calculado. Possui o método `double valor(double ponto)`, que retorna o valor da função implementada na classe no ponto especificado. Você pode modificar essa classe, para testes, mas saiba que **quaisquer modificações a essa classe serão perdidas**, uma vez que ela não deve ser entregue.

- `TestaIntegral`, onde você pode incluir seus testes particulares.

2.1 Entrada

A entrada é composta pelos parâmetros descritos acima (ou seja, `f`, `a`, `b` e `n`).

2.2 Saída

Como saída, o método retorna o valor aproximado de $S = \int_a^b f(x)dx$, calculado pela régra do trapézio, em que são usados n trapézios para a aproximação.

2.3 Material a Ser Entregue

Deverá ser entregue tão somente o arquivo `Integral.java`, com o método `resolve` implementado. A entrega será feita unica e exclusivamente via tidia-ae, até a data marcada. Deverá ser postado nesse sistema um zip com o arquivo .java correspondente ao exercício, tendo seu número USP como nome, ou seja:

`número_usp.zip`

Somente este arquivo zip deve ser postado no tidia. A responsabilidade de postagem nele é exclusiva do aluno. Por isso, problema referentes ao uso do sistema devem ser resolvidos com antecedência.

3 Avaliação

Para avaliação, serão observados os seguintes quesitos:

1. Documentação: se há comentários explicando o que se faz nos passos mais importantes e para que serve o programa (Tanto o método quanto o programa em que está inserido)
2. Apresentação visual: se o código está legível, identado etc
3. Corretude: se o programa funciona

Além disso, algumas observações pertinentes ao trabalho, que influem em sua nota, são:

- Este exercício-programa deve ser elaborado individualmente.
- Não será tolerado plágio, em hipótese alguma.
- Exercícios com erro de sintaxe (ou seja, erros de compilação), receberão nota ZERO

Atenção! Para avaliação, apenas o método `resolve(Funcao f, double a, double b, int n)` será invocado diretamente, então tenha certeza de que o problema é resolvido chamando-se diretamente somente esse método.

O procedimento seguido para correção será:

1. Pegarei a sua classe `Integral.java`
2. Pegarei a minha classe `Funcao.java`
3. Rodarei o método resolve da sua classe `Integral`, passando como parâmetros um objeto da minha classe `Funcao`, além dos valores a, b e n nos parâmetros de `resolve()`

4. Repetirei o mesmo procedimento com a minha Integral.java
5. Verei a diferença entre meu resultado e o resultado matemático (a resposta correta à integral)
6. Verei a diferença entre o seu resultado e o resultado matemático
7. O resultado do item 6 não pode ser maior, em módulo, que 1,5 vezes o resultado do item 5 (ou seja, seu erro pode ser até 50% maior que o meu)

Qualquer coisa que atrapalhe esse caminho irá causar queda (por vezes substancial) na nota.

4 Data de Entrega

25/06/2018.