Integración por el método de los rectángulos

IIP 16-17 (@mrebollo)
Departamento de Sistemas Informáticos y Computación
Universitat Politècnica de València

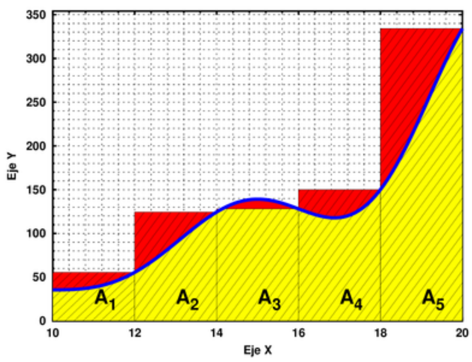
1. Aproximación al valor de la integral

En matemáticas hay determinadas operaciones que no tienen solución analítica o es difícil calcularla. En ese caso, se emplean lo que se conoce como métodos numéricos, que no son más que unos algoritmos que permiten al ordenador aproximarse al valor que estamos buscando.

Un ejemplo de método numérico sencillo es el que nos permite calcular la integral de una función en un intervalo.

$$\int_a^b f(x)\mathrm{d}x$$

Recuerda que el significado de la integral es el área que se encuentra por debajo de la curva que define la función. Una forma de hacerlo es dividir el intervalo $[a, b]$ en varios intevalos más pequeños y construir rectángulos con una altura determinada que corresponderá con el valor de la función en algún punto. Por ejemplo, en la siguiente figura, se construye el rectángulo con una altura igual al valor de $f(x)$ en el valor último del rectángulo



Si sumamos el área de todos los rectángulos, obtendremos un valor que se aproximará al valor real de la integral. Cuanto más pequeños sean los rectángulos, la solución estará más cerca del valor de la integral. Básicamente si dividimos el intervalo $[a, b]$ en N rectángulos

$$\Delta x = \frac{b-a}{N}, \qquad \sum_{i=1}^{N} f(x_i)\Delta x \approx \int_a^b f(x)\mathrm{d}x$$

2. Enunciado del ejercicio

Construye un programa en Java que aproxime el valor de la integral de una función usando el método de los rectángulos. El programa debe constar de los siguientes componentes

Clase `Funcion`

Una clase que modela una función matemática. La clase contendrá un único método

1. `eval` que calcula el valor de la función en un punto x que se le pasa como parámetro

Clase `Intervalo`

Una clase que modela un intervalo entre dos puntos. Esta clase debe contener

1. dos atributos de instancia: los límites inferior y superior del intervalo
2. un constructor general que construye un intervalo a partir de sus límites
3. un constructor por defecto que crea un intervalo $[0, 1]$
4. dos métodos consultores para obtener el límite inferior y superior del intervalo
5. un método `size` que devuelve el tamaño del intervalo

Clase `Rectángulo`

Una clase que modela un rectángulo. Esta clase debe contener

1. dos atributos de instancia: la longitud de cada lado
2. un constructor general que construye un rectángulo a partir de sus lados
3. un método modificador que cambia las dimensiones del rectángulo
4. un método `area` que devuelve el área del rectángulo

Clase `Integral`

Una clase que modela una integral definida en un intervalo. Esta clase debe contener

1. dos atributos de instancia: la función que se desa calcular (de tipo `Funcion`) y el intervalo en el que está definida (de tipo `Intervalo`)
2. un constructor general que dado un intervalo que se le pasa como parámetro, construye la integral
3. un constructor que, dado los límites inferior y superior que se le pasan como parámetro, construye la integral
4. un constructor por defecto que crea una integral en el intervalo $[0, 1]$
5. un método `resolver` calcula el valor de la integral y lo devuelve

Programa principal `CalcIntegral`

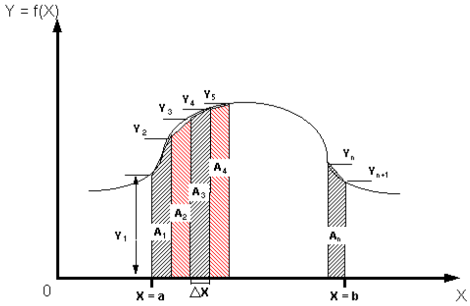
Una clase que contenga el programa principal y que calcule el valor aproximado de una integral usando el método de los rectángulos. El programa contendrá:

1. el método `main` que solicita al usuario los límites inferior y superior del intervalo y el número de intervalos que desea usar, crea una integral con esos parámetros, calcula su valor y lo muestra por pantalla

3. Ampliaciones propuestas

3.1. Aproximación mediante trapecios

La aproximación mediante rectángulos tiene un error alto, pues solo utiliza un punto de la función $f(x)$ en cada intervalo. Una forma de conseguir una mejor solución es emplear trapecios, usando los valores de $f(x_i)$ y $f(x_{i+1})$ para el intervalo i tal y como muestra la siguiente figura



Implementa una clase `Trapecio` que tiene como atributos las dos bases del trapecio y su altura (mira en la figura a qué corresponde cada dimensión) y crea un método `resuelveTrap` que calcula el valor de la integral por el método de los trapecios.

Sugerencia: renombra el método `resuelve` existente a `resuelveRect` para seguir la misma nomenclatura. Puedes definir un método general `resuelve` al que se le pasa como parámetro el número de intervalos y el método de resolución, que invoca al método correspondiente. Se pueden definir dos constantes INT_REC = 0, INT_TRA = 1 para identificar los métodos.

3.2. Precisión del método

Este método de cálculo de la integral es más preciso cuantos más intervalos se generan. Modifica el programa principal para que, en lugar de pedir al usuario el número de intervalos, realice varios experimentos aumentando el número de intervalos. Por ejemplo, con 5, 10, 15,..., 100. Muestra por pantalla el resultado obtenido con cada método.

Puedes compararlo con el valor real a través de la página http://www.wolframalpha.com Por ejemplo, `integrate sin(x) from 0 to 1` nos muestra la integral de $\sin x$ entre 0 y 1.

Para mostrar los resultados en forma de tabla, puedes emplear la función `System.out.printf` en lugar de `System.out.println`. El primer argumento es una cadena en la que se incluye en qué formato aparecen las variables. Por ejemplo

```
System.out.printf("%d + %d = %d\n", a, b, a + b)
```

Muestra en pantalla `2 + 1 = 3` para $a = 2$ y $b = 1$. Lo que hace es sustituir cada %d por la correspondiente variable o expresión en el orden en el que aparecen después. Los más habituales son

`%d` para enteros

`%f` para decimales (double o float)

`%c` para caracteres

`%s` para cadenas (String)

`\t` introduce un tabulador

`\n` introduce un salto de línea