Integración Trapecio
$f(x) = sen(\Pi x)$, $x \in [-2, -1]$

Lára Kristjánsdóttir, Javier Hernández Pérez

Grupo (2 | F)

Técnicas Experimentales. 1^{er} *curso.* 2^{do} *semestre*

Lenguajes y Sistemas Informáticos

Facultad de Matemáticas

Universidad de La Laguna

La Laguna, 17 de mayo de 2013

Índice general

Índice de figuras

Índice de cuadros

Capítulo 1

Motivación y objetivos

1.1. Utilidad el método

No todas las integrales son fáciles de resolver. Algunas son, de hecho, imposibles de calcular analíticamente. Por ello, es muy importante tener métodos que nos permitan aproximar numéricamente las integrales cuando tenemos una integral definida. En este caso nos centraremos en el método de los trapecios.

Además, este método tiene una importancia extra. Si un método, como este, es sencillo.Teniendo en cuenta que con funciones en la vida real, midiendo un terreno por ejemplo,normalmente (salvo casos particulares como por ejemplo una persona con un terreno cuadrado que quiera que cada uno de sus dos hijos hereden la mitad de su terreno dividido por la función $sen(\pi x)$) estas haciendo una aproximación.Usar directamente el método de los trapecios parece una opción lógica.

Por ejemplo hallas el área de un cuadrado al medio del terreno y vas eligiendo puntos de la frontera del terreno que esten unidos aproximadamente por lineas rectas. Ya solo queda hallar el area de los trapecios con base el lado del cuadrado que les quede más cerca y sumar.

1.2. Nuestra integral $f(x) = sen(\pi x)$, $x \in [-2, -1]$ se puede resolver

Nuestra integral es fácil de resolver analíticamente, lo cual es una gran ventaja ya que nos permite comprobar cómo de buenas serán nuestras aproximaciones. Para ello simplemente hemos de restar al valor calculado analíticamente el resultado de la aproximación por el método de los trapecios

Resolvamos ahora $\int_{-2}^{-1} sen(\pi x)dx$. Haciendo el cambio $y = \pi x$ teniendo en cuenta que $\frac{\partial y}{\partial x} = \pi$, $x = -2 \Rightarrow y = -2 \cdot \pi$, $x = -1 \Rightarrow y = -\pi$ obtenemos que $\int_{-2}^{-1} sen(\pi x)dx = \frac{\int_{-2\pi}^{-1\pi} sen(y)dy}{\pi} = \frac{-cos(-\pi)+cos(-2\cdot\pi)}{\pi} = \frac{1+1}{\pi} = \frac{2}{\pi} \simeq 0,6366197724$

Capítulo 2

Fundamentos teóricos

2.1. Método de los trapecios

El método de los trapecios consiste en evaluar la función a integrar en algunos puntos que se unen con líneas rectas para aproximar el resto. Si unimos el eje y los puntos evaluados con rectas perpendiculares al eje. Hemos obtenido los trapecios formados por las rectas perpendiculares a los ejes, el eje y las rectas que unen a los puntos evaluados.La suma del area de esos trapecios es muy parecida a el area de debajo de la función es decir la integral. El área de esos trapecios es facil de hallar$\frac{h}{2}(a+b)$ donde h es la longitud del eje entre la imagen en el eje de los dos puntos por la recta perpendicular al punto, a y b la evaluación de la función en los dos puntos.

2.2. Convergencia del método de los trapecios

Teorema:
Sea f:$[a,b] \to \mathbb{R}$ una función integrable en el intervalo cerrado $[a,b]$. Si $P = \{x_0, x_1, x_2, ..., x_n\}$ es la partición de $[a,b]$ que divide a este intervalo en n partes iguales, entonces el siguiente número aproxima a la integral $\int_a^b f(x)dx$.

$$I_t = \frac{b-a}{n}\left(\frac{f(a)}{2} + f(x_1) + f(x_2) + f(x_3) + ... + f(x_{n-1}) + \frac{f(b)}{2}\right)$$

La aproximación tiene un error de si la función f es C^2 (derivable 2 veces con derivada continua).

$$|I_t - \int_a^b f(x)dx| \leq |\frac{(b-a)^3}{12n^2}f''(\xi)|$$

Para algún $\xi \in [a,b]$, luego si K es una cota superior de la derivada segunda en cualquier punto de $[a,b]$ tenemos:

$$|I_t - \int_a^b f(x)dx| \leq |\frac{(b-a)^3}{12n^2}K|$$

Aunque esto no lo vamos a demostrar (pues es una cota mala y $f(x) = sen(\Pi x)$ es C^∞) si f no es C^2 solo pidiendole monotonía en el intervalo se tiene:
$|I_t - \int_a^b f(x)dx| \leq \frac{b-a}{n}|f(b) - f(a)|$

Demostración:

Primero veamos que: $I_t = \frac{b-a}{n}\left(\frac{f(a)}{2} + f(x_1) + f(x_2) + f(x_3) + ... + f(x_{n-1}) + \frac{f(b)}{2}\right)$ aproxima a la función

Esto es cierto pues $\frac{b-a}{n}$ es la parte del eje x del trapecio entre cada 2 puntos.Si haces todos los trapecios cada punto se repite 2 veces salvo los extremos asi que son solo los extremos los que se dividen entre 2 cuando sumas todas las areas de los trapecios.

Ahora veamos que $|I_t - \int_a^b f(x)dx| \leq \frac{(b-a)^3}{12n^2}K$ es una cota del error

Sea α un punto de la partición P sea $h = \frac{b-a}{n}$ por simplificar notación. La h es la longitud de cada uno de los n intervalos.

Cojamos una función que nos lo hace en intervalos más pequeños que los h para ir hallando el error y que cuando t vale h es un solo trapecio de los n que tenemos menos la integral en ese trapecio es decir el error en un trapecio.

$F_\alpha : [0, h] \to \mathbb{R}$
$t \to F_\alpha(t) = \frac{t}{2}\left(f(\alpha) + f(\alpha + t)\right) - \int_\alpha^{\alpha+t} f dt$

Derivando obtenemos:
$F'_\alpha(t) = \frac{1}{2}\left(f(\alpha) + f(\alpha+t)\right) + \frac{t}{2}0 + \frac{t}{2}f'(\alpha+t) - f(\alpha+t) + 0 = \frac{1}{2}\left(f(\alpha) + f(\alpha+t)\right) + \frac{t}{2}f'(\alpha+t) - f(\alpha+t) = \frac{1}{2}\left(f(\alpha) - f(\alpha+t)\right) + \frac{t}{2}f'(\alpha+t)$

Derivando de nuevo

$F''_\alpha(t) = \frac{1}{2}\left(0 - f'(\alpha+t)\right) + \frac{1}{2}f'(\alpha+t) + \frac{t}{2}f''(\alpha+t) = -\frac{1}{2}f'(\alpha+t) + \frac{1}{2}f'(\alpha+t) + \frac{t}{2}f''(\alpha+t) = \frac{t}{2}f''(\alpha+t)$

Elijamos ξ es decir cojemos el K cota superior de la derivada segunda.
Así que tenemos que $|F''_\alpha(t)| \leq |\frac{t}{2}K|$

Integrando que tenemos la ventaja de que K no depende de t y que $F'_\alpha(0) = \frac{1}{2}\left(f(\alpha) - f(\alpha+0)\right) + \frac{0}{2}f'(\alpha+0) = 0 + 0 = 0$ tenemos:

$|\int_0^t F''_\alpha(t)| \leq |\int_0^t \frac{t}{2}K| \Rightarrow |F'_\alpha(t) - F'_\alpha(0)| \leq |\frac{t^2}{4}K - \frac{0^2}{4}K| \Rightarrow |F'_\alpha(t)| \leq |\frac{t^2}{4}K|$
Ahora integremos entre 0 y h que es donde se mueve la función $F_\alpha(t)$ con la ventaja de que $F_\alpha(0) = \frac{0}{2}\left(f(\alpha) + f(\alpha+0)\right) - \int_\alpha^{\alpha+0} f dt = 0 - 0 = 0$

$|\int_0^h F'_\alpha(t)| \leq |\int_0^h \frac{t^2}{4}K| \Rightarrow |F_\alpha(h) - F_\alpha(0)| \leq |\frac{h^3}{12}K| - \frac{0^3}{12}K| \Rightarrow |F_\alpha(h)| \leq |\frac{h^3}{12}K|$
Hemos obtenido que:

$|F_\alpha(h)| = |\frac{h}{2}\left(f(\alpha) + f(\alpha+h)\right) - \int_\alpha^{\alpha+h} f dh| \leq |\frac{h^3}{12}K|$

Si hacemos n veces esta $F_\alpha(h)$ obtenemos todos los errores en los trapecios del intervalo por tanto la cota que nos interesa es:

$n|\frac{h^3}{12}K|$

Teniendo en cuenta que habiamos defindo h como:

$h = \frac{b-a}{n}$

Obtenemos:

$n|\frac{(b-a)^3}{12(n^3)}K| \Rightarrow |\frac{(b-a)^3}{12(n^2)}K|$

Como queriamos demostrar $\smile$.

2.3. Realización por el método de los trapecios de $f(x) = sen(\Pi x)$, $x \in [-2, -1]$

$f(x) = sen(\Pi x)$, $x \in [-2, -1]$

Capítulo 3

Procedimiento experimental

3.1. Descripción de los experimentos

Vamos a aplicar la regla del trapecio a la función $sin(\pi x)$ en el intervalo $[-2,-1]$ utilizando una cantidad variable de subintervalos, n. Para cada valor de n, mediremos el error absoluto y el tiempo de ejecución del método. Los valores de n que consideraremos irán desde $n = 1$ hasta un n_f tal que el error del método no disminuya casi nada cuando $n > n_f$.

Primero calculamos analíticamente la cota superior del error para comparar los errores que vamos obteniendo con dicha cota. Sabemos que su expresión es

$$error(n) < -\frac{h^2}{12}f''(\xi)(b-a) \quad , \quad \xi \in [a,b]$$

que en nuestro caso será

$$\frac{\pi^2}{12n^2}$$

Para calcular el error absoluto de cada aproximación, restaremos al valor calculado analíticamente para la integral, $\frac{2}{\pi}$, el valor obtenido por el método. De la resta, haremos el valor absoluto.

3.2. Descripción del material

Los experimentos han tenido lugar sobre un procesador Intel Core i3-2350M a 2.30 GHz. Nótese que, aunque la máquina tiene cuatro procesadores, sólo se empleó uno de ellos para realizar los cálculos. El sistema operativo fue Linux Mint 14.1 (Nadia) con la versión 3.5.0-17 del kernel. Los algoritmos fueron implementados en Python y la versión del intérprete de este lenguaje fue la 2.7.3.

3.3. Resultados obtenidos

Resumimos los resultados obtenidos en la siguiente tabla:

n	**error**	**tiempo**
1	0.6366197724	0.0000169277
3	0.0592695032	0.0000441074
5	0.0210830649	0.0000529289
7	0.0107217341	0.0000619888
8	0.0082023359	0.0000700951
12	0.0036402630	0.0000801086
16	0.0020466231	0.0000920296
100	0.0000523607	0.0001320839
200	0.0000130900	0.0002050400

Cuadro 3.1: Resultados obtenidos

donde n es el número de subintervalos, *error* es el error absoluto cometido y *tiempo* es el tiempo de ejecución.

3.4. Análisis de los resultados

Para analizar los resultados, generamos dos gráficas: n contra *error*(3.1) y n contra *tiempo*(3.3). Además, para combrobar la corrección del método, generamos una tercera gráfica en la que superponemos la cota del error al error absoluto obtenido(3.2).

Podemos ver que el error absoluto, calculado con el método del trapecio, converge a cero rápidamente (de manera cuadrática) y que apartir de $n = 20$ ya obtenemos una buena aproximación de la integral que queremos calcular.

Al superponer la cota del error al error obtenido, podemos ver que el segundo siempre está por debajo, como era de esperar. Esto nos indica que el método ha sido implementado correctamente.

Para observar mejor el coste temporal del método, hemos tomado valores de n desde $n = 1$ hasta $n = 500$. Así podemos ver que a partir de $n = 300$ el crecimiento de la gráfica es lineal: es decir, proporcional al valor de n. Esto es lo que esperábamos puesto que, para obtener una aproximación con n subintervalos, necesitamos hacer n operaciones, todas ellas de coste constante.

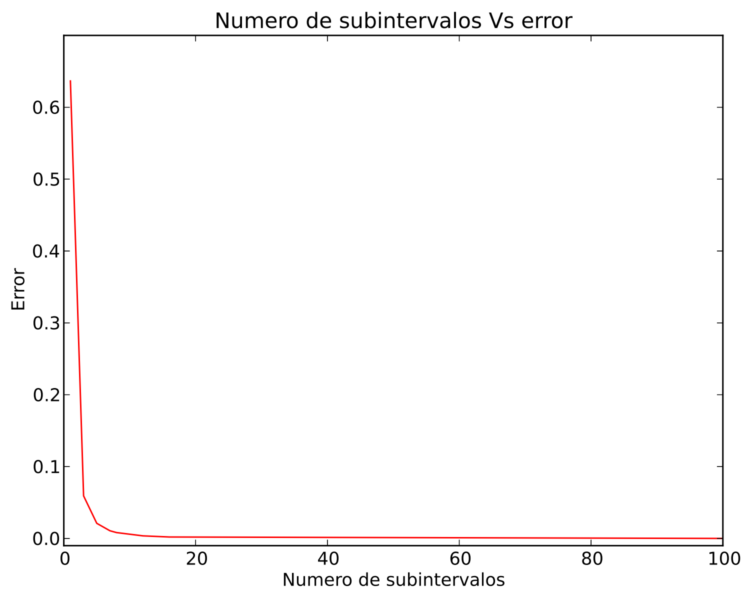

Figura 3.1: nVSerror

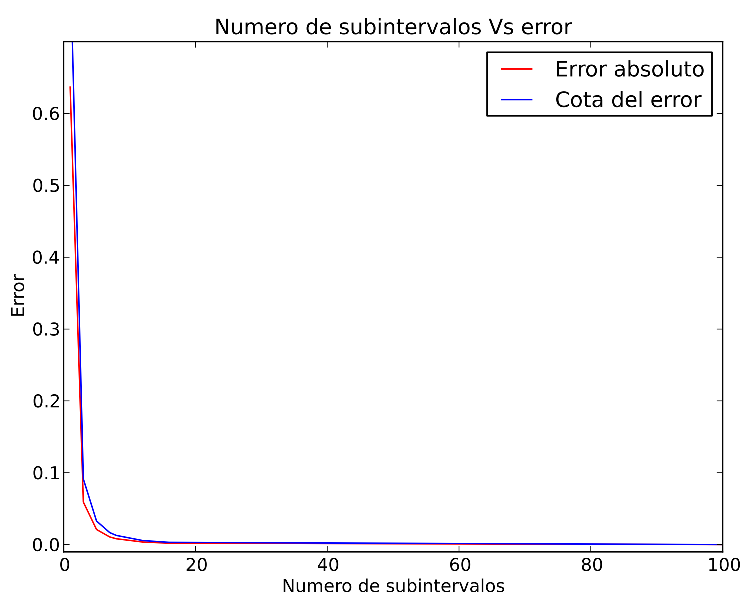

Figura 3.2: nVSerror

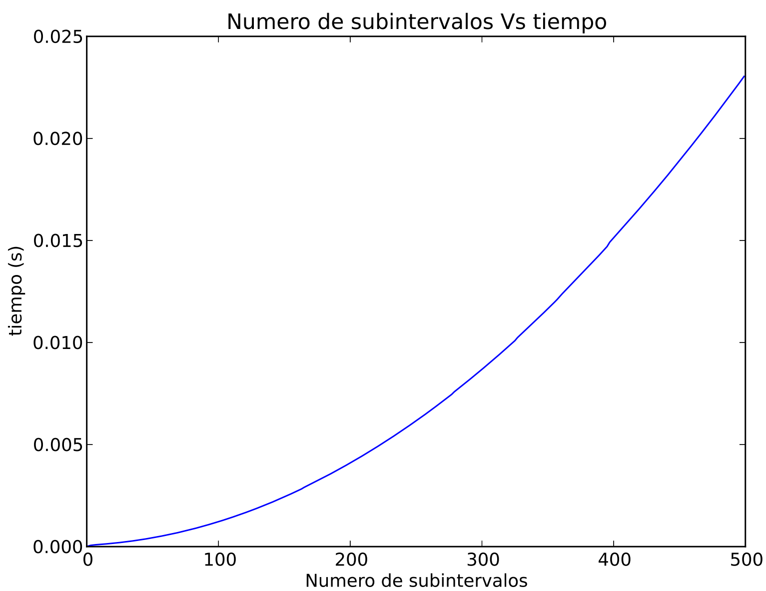

Figura 3.3: nVStiempo

Capítulo 4

Conclusiones

4.1. Conclusiones

El método del trapecio nos da una buena aproximación de la integral que queríamos calcular con un coste temporal razonable. El error obtenido nunca supera la cota calculada y con más subintervalos el error absoluto disminuye y la aproximación es más precisa aunque se tarde más.

Apéndice A

Título del Apéndice 1

A.1. Algoritmo MODULO DE TRAPECIO

```
/###################################################################################
 # modulo_trapecio.py
 ###################################################################################
 #
 # Lara Kristjansdottir y Javier Hernandez Perez
 #
 # 17.mayo 2013
 #
 # Es un modulo que contiene una funcion que usa como parametro el numero de
 # subintervalos, n,  y calcula el valor aproximado de la integral de seno(pi*x)
 # entre -2 y -1 con el metodo del trapcio.
 ##################################################################################
import sys
from math import *

def trapecio_senxpi(n):
  a=-2.0
  b=-1.0
  h=(b-a)/n
  s=(sin(a*pi)+sin(b*pi))/2.0
  for i in range(1,n):
    s+=sin(pi*(a+i*h))
  return h*s

if __name__=='__main__':
  if len(sys.argv)==2:
      n=int(sys.argv[1])

      print trapecio_senxpi(n)
  else:
    print "El modo de uso es:%s <numero de trapecio>" % (sys.argv[0])
```

A.2. Algoritmo MODULO ERROR ABSOLUTO

```
/##################################################################################
 # modulo_error.py
 ##################################################################################
 #
 # Lara Kristjansdottir y Javier Hernandez Perez
 #
 # 17.mayo 2013
 #
 # Es un modulo que contiene una funcion que recibe por parametro un valor y calcula
 # el error absoluto entre este valor y el valor real de la integral de seno(pi*x)
 # entre -2 y -1
 #
 ##################################################################################
import sys
from math import *

def error(valor_trapecio):
  valor_real=2/pi
  return abs(valor_real-valor_trapecio)

if __name__=='__main__':
  if len(sys.argv)==2:
    valor_trapecio=float(sys.argv[1])

    print error(valor_trapecio)
  else:
    print "El modo de uso es:%s <valor con metodo de trapecio>" % (sys.argv[0])
```

A.3. Algoritmo METODO TRAPECIO

```
/##################################################################################
 # uso_modulo_trapecio.py
 ##################################################################################
 #
 # Lara Kristjansdottir y Javier Hernandez Perez
 #
 # 17.mayo 2013
 #
 # Un programa que calcula el valor de la funcion seno(pi*x) con el metodo del
 # del trapecio, el valor del error absoluto y el tiempo de ejecucion y guarda estos
 # resultados en un fichero llamado resultado.txt.
 #
 ##################################################################################
import time
from math import *
from modulo_trapecio import trapecio_senxpi
from modulo_error import error

n=range(1,8,2)+range(8,20,4)+ range(100,300,100)

if __name__=='__main__':
```

```
fichero=open('resultado.txt', 'w')

e0 = time.time()

for i in range(len(n)):
  fichero.write("%i\n%.10f\n%.10f\n"%(n[i],error(trapecio_senxpi(n[i])),time.time() - e0))
  i+=1
fichero.close()
```

Bibliografía

[1] Juan de Burgos Roman. *Calculo infinitesimal de una variable 2 edicion*. McGRAW-HII/INTERAMERICANA DE ESPANA,S.A.U., 2007.

[2] LaTeX. http://www.latex-project.org/.

[3] Python. http://www.python.org/.

[4] Wikipedia.org. http://es.wikipedia.org/wiki/Regla_/del_/trapecio.