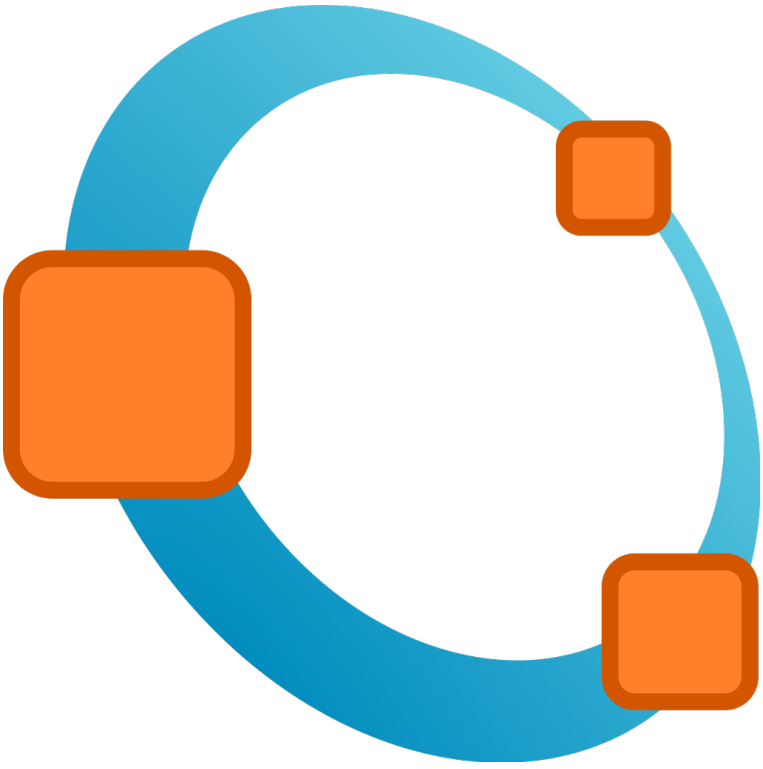
Actividad de Aprenizaje 2
Octave

Análisis Matemático



Autor: Alexander Sebastian Kalis
Profesor: Dr. Juan José Moreno García
Curso: 1o, Ingeniería de Organización Industrial
UDIMA

Índice

1. Introducción

El objetivo de la actividad es la contstrucción de un programa mediante el lenguaje MatLab/Octave que nos permita computar la resolución numérica de integrales por varios métodos, es decir, calcular el área que hay por debajo de una curva.

El método utilizado para la resolución de la actividad será el método de los trapecios. Este método trata de dar los valores que tiene la función f para una serie de valores equidistantes de x. De este modo tenemos trapecios de distintas áreas que podemos sumar para dar el área total. También se utilizará el método de Simpson, que aproxima las curvas a un poliniomio de grado tres.

2. Desarrollo de la actividad

2.1. Apartado A

Calcular numéricamente la siguiente integral definida:

$$I = \int_0^{2\pi} \left| x \operatorname{sen} x^2 \right| dx$$

Usando el sistema de los trapecios y Simpson, para 11, 101, 1001 y 10001 puntos e interpretar ambos casos.

2.1.1. Sistema de trapecios

Para el sistema de trapecios, se ha desarrollado una función que toma 4 argumentos.

- Límite inferior de integración.
- Límite superior de integración.
- La función a integrar.
- El número de trapecios en los que descomponemos el área de la integral.

AlexanderSebastianKalisAA2.tex ● trapz.m × simp13.m ●

```
%La función toma como argumentos el límite inferior de integración,
%el límite superior de integración y el número de trapezoides
%en el que queremos dividir el área
function  I = trapz (arg1, arg2, arg3, arg4)

%Límites de integración
a = arg1;
b = arg2;

%Función a calcular numéricamente
f = arg3;

%Número de trapezoides
n = arg4;

%Altura de cada trapezoide
h = (b - a) / n;

%Primera y última parte de la integral
s = 0.5 * (f(a) + f(b));

%Suma de todos los términos intermedios
for i = 1 : n - 1
    s = s + f(a + i * h);
end

%Total
h * s

end
```

Comparativa de resultados

Esto nos permite comparar los resultados que obtenemos dependiendo de la cantidad de trapecios en los que dividimos el área.

A continuación se verá unos ejemplos de cómo influyen las subdivisiones en la aproximación del resultado.

Resultado por cálculo analítico

Mediante el cálculo analítico podemos concluir que

$$I = \int_0^{2\pi} \left| x \operatorname{sen} x^2 \right| dx = 12{,}604$$

Resultado por cálculo numérico

Utilizando la función programada en Octave, se evalúa el resultado de la integral dividiendo el área en 10, 100, 1000 y 10000 trapecios:

```
PROBLEMS 9  OUTPUT  DEBUG CONSOLE  TERMINAL
octave:30> trapz(0,2*pi,@(x)abs(x*sin(x^2)),10)
ans =  10.235
octave:31> trapz(0,2*pi,@(x)abs(x*sin(x^2)),100)
ans =  12.634
octave:32> trapz(0,2*pi,@(x)abs(x*sin(x^2)),1000)
ans =  12.603
octave:33> trapz(0,2*pi,@(x)abs(x*sin(x^2)),10000)
ans =  12.603
octave:34>
```

Podemos ver claramente cómo el valor se aproxima cada vez más al valor analítico a medida que aumentamos los trapecios. Sim embargo aplicando la fórmula del error absoluto $\epsilon_a = \bar{X} - X_i$ podemos ver que **hay picos de aumento en error absoluto cuando nos acercamos a las decenas:**

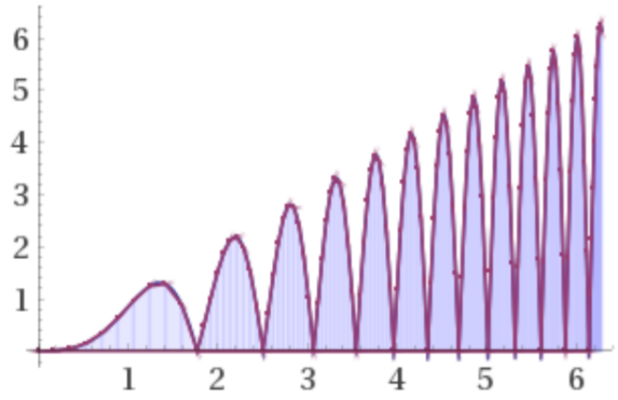


Figura 1: Subdivisiones de f

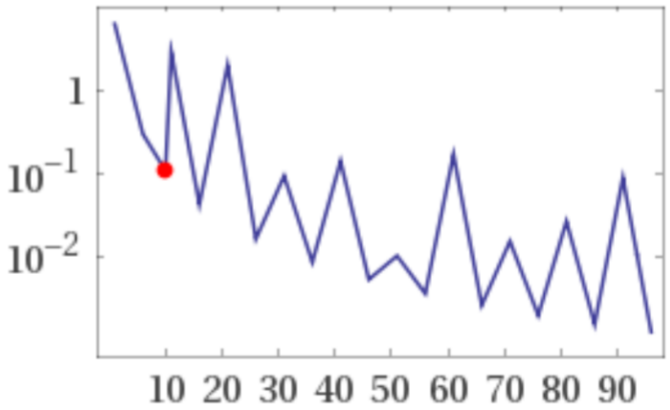


Figura 2: Error absoluto respecto a las subdivisiones

Corregido:
Dr. J.J. Moreno García

2.1.2. Sistema de Simpson

Para el sistema de Simpson, también se ha buscado una forma alternativa de plantear el programa con la finalidad de practicar. En este caso la función solo admite 1 argumento, el número de subdivisiones, las cuales deben ser pares ya que van de 2 en 2.

AlexanderSebastianKalisAA2.tex simp13.m

```
function I = simp13(arg)

    f = @(x)abs(x*sin(x^2));
    a = 0;
    b = 2*pi;
    n = arg;
    h = (b - a) / n;
    s = f(a) + f (b);
    if(rem(arg,2)~=0)
        error('Tiene que usar un numero par de subdivisiones');
    end

    for i = 1 : 2 : n - 1
        s = s + 4*f(a+i*h);
    end

    for i = 2 : 2 : n - 2
        s = s + 2 * f(a + i * h);
    end

    h / 3 * s

endfunction

```

Resultado por cálculo numérico

Se calculan los resultados para 10, 100, 1000 y 10000 subdivisiones:

PROBLEMS 8 OUTPUT DEBUG CONSOLE TERMINAL

```
octave:26> simp13(10)
I =  9.9430
ans =  9.9430
octave:27> simp13(100)
I =  12.609
ans =  12.609
octave:28> simp13(1000)
I =  12.604
ans =  12.604
octave:29> simp13(10000)
I =  12.604
ans =  12.604
octave:30>
```

Volvemos a consultar los gráficos de subdivisión y aproximación:

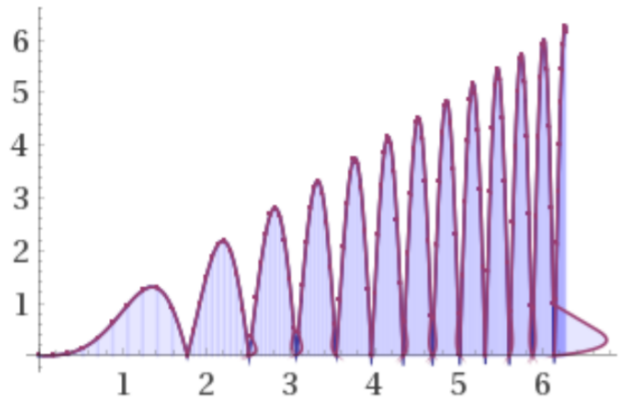


Figura 3: Subdivisiones de f

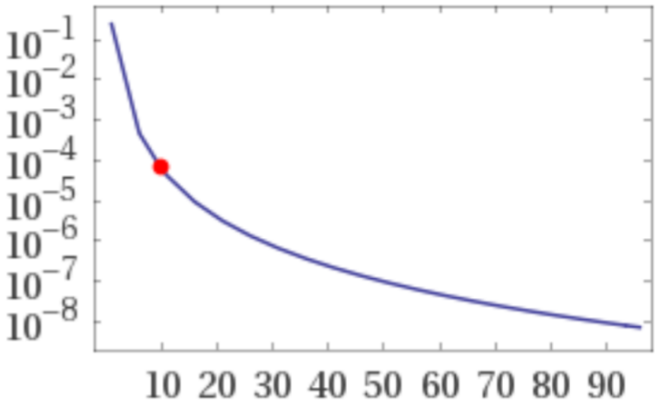


Figura 4: Error absoluto respecto a las subdivisiones

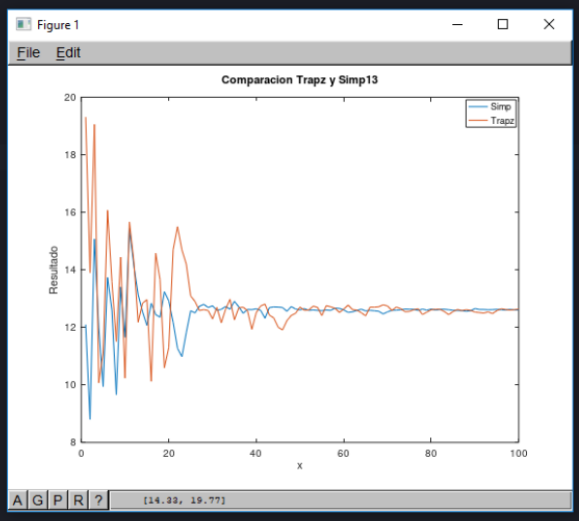
Donde se ve claramente la diferencia con el método de los trapecios, ya que la función del error absoluto baja constantemente a medida que se aumentan las subdivisiones, a diferencia del método de los trapecios, en el cual había altibajos propios de una función sinusoidal que eran provocados por los trapecios saliéndose de la gráfica de la función en las bajadas o bien no llegando hasta el final en las subidas.

Conclusión

Para concluir con ambos métodos conjutamente, utilizamos la función plot() para ver los resultados de ambos métodos juntos en una gráfica:

```
N=100
resultadosTrapz  =   zeros(N,1);
resultadosSimp  =   zeros(N,1);
x = zeros(N/2,1);
for i = 1 : 1 : N
    x(i) = i;
    resultadosSimp(i) = simp13(i*2)
    resultadosTrapz(i) = trapz(0,2*pi,@(x) abs(x*sin(x^2)),i)
endfor

plot(x,resultadosSimp)
title("Comparacion Trapz y Simp13");
  legend('Simp', 'Trapz');
  xlabel("x");
  ylabel("Resultado");
hold on
plot(x,resultadosTrapz)
legend('Simp', 'Trapz');
hold off
```



Corregido:
Dr. J.J. Moreno García

2.2. Apartado B

Calcular analíticamente la integral definida

$$I = \int_0^{\pi} \text{sen}^3\, x\; dx = \int_0^{\pi} \left(1 - \cos^2(x)\right) \sin(x)\, dx \int_1^{-1} -1 + u^2 du = -\int_{-1}^{1} -1 + u^2 du =$$

$$-\left(-\int_{-1}^{1} 1du + \int_{-1}^{1} u^2 du\right) = -\left(-2 + \frac{2}{3}\right) = \frac{4}{3}$$

Entonces, por comparación, calcular la cantidad mínima de puntos que habría que usar con Simpson para tener un resultado con al menos 5 cifras significativas correctas (4 decimales).

Para comparar los resultados, se ha realizado otro script que se ha llamado comparacion.m. Este script consta de un bucle *while* que va ir iterando entre posibles valores de subdivisiones (de 2 en 2) mientras que el resultado no cumpla la condición. Una vez cumplida la condición propuesta, el bucle se romperá e imprimirá por consola el valor que estamos buscando.

AlexanderSebastianKalisAA2.tex | simp13.m | comparacion.m | trapz.m

```
i = 2;

while (1.3333 <= simp13(i) && simp13(i) < 1.3334) ~= true
    i = i + 2;
endwhile

simp13(i)
disp(i)
disp("Es el numero adecuado de subdivisiones para conseguir 4 decimales correctos")
```

Entonces ejecutando este nuevo programa obtenemos el resultado de **22 subdivisiones o 23 puntos**

```
PROBLEMS 14  OUTPUT  DEBUG CONSOLE  TERMINAL

octave:33> comparacion
ans =  1.333304925454986
 22
Es el numero adecuado de subdivisiones para conseguir 4 decimales correctos
octave:34>
```