

INTEGRACIÓN NUMÉRICA

REGLA TRAPEZOIDAL

Yamil Armando Cerquera Rojas
yacerque1@hotmail.com

Objetivos: Resolver el problema de cálculo del área bajo la curva entre dos límites conocidos, dividiendo en N sub áreas para calcular su valor asumiendo cada sub área como un pequeño trapecio.

Temas:

- ✓ Cálculo de áreas.
- ✓ Método de los trapecios.
- ✓ Programación del método de los trapecios.
- ✓ Cálculo del área de múltiples funciones en base a subclases.

Cálculo de áreas

Uno de los problemas matemáticos más frecuentes es el cálculo del área que se forma al graficar una función. Por ejemplo, se necesita calcular el área A que aparece en la siguiente figura por debajo de la función $f(x)$ entre los límites a y b :

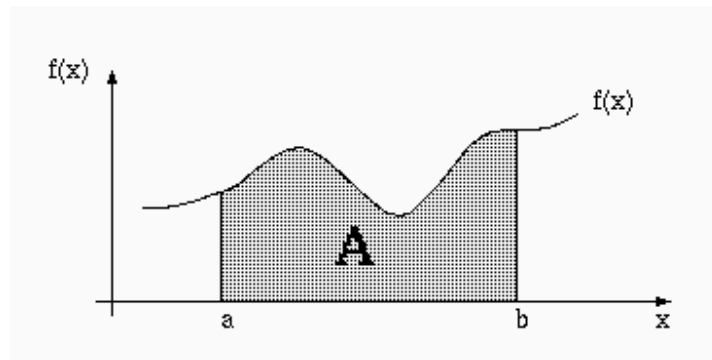


Fig. 1

En donde la función $f(x)$ y los valores a y b son valores conocidos. a se considera como el límite inferior y b se considera como límite superior.

En este tipo de problemas se pueden obtener dos tipos de soluciones:

- ✓ Soluciones algebraicas: se obtiene una fórmula precisa y exacta para el área solicitada.
- ✓ Soluciones numéricas: se calcula numéricamente una estimación del área.

Desde luego, las soluciones algebraicas son mejores que las numéricas, porque son exactas. Pero a veces, la complejidad de las funciones hace imposible (o

difícil) obtener la solución algebraica, por lo que una solución numérica permite ahorrar tiempo.

REGLA TRAPEZOIDAL O REGLA TRAPEZIAL.

La Fig. 2 muestra de color verde como sería el cálculo del área bajo la curva de la función $f(x)$ entre los límites a y b si se dividiera dicha subarea en un solo trapecio. El error que se cometería sería demasiado grande con respecto al área real que se desea obtener. Dependiendo de la forma de la curva el error que se cometería sería por exceso o por defecto. En el caso del ejemplo, el error sería por defecto, es decir el valor que arroje el cálculo de la integral será menor al valor real del área.

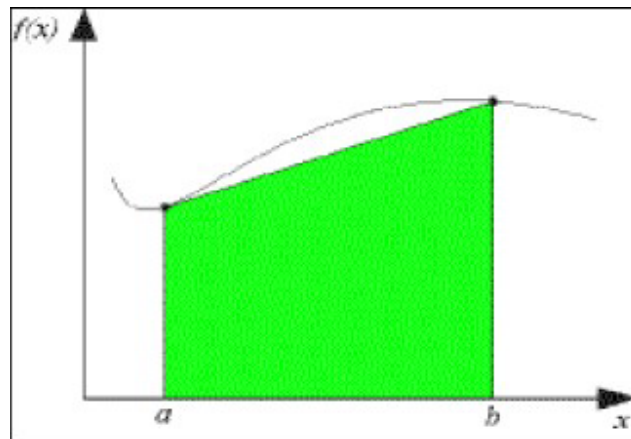


Fig. 2

Si se divide el intervalo (área a calcular) en mas de una sub área, en el caso de la Fig. 3 (dividida en 3 sub áreas), el error en le cálculo de la integral o área total, se disminuye.

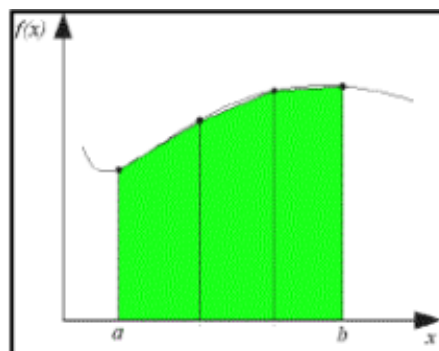


Fig. 3

La estrategia más simple y que evitaría menor error en el cálculo, consiste en subdividir el intervalo pedido para el cálculo del área en n sub intervalos de

pequeño tamaño y aproximar el área como la suma de las áreas de cada uno de los trapecios que se forman:

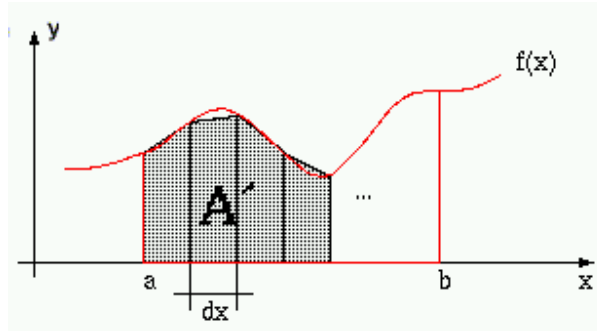


Fig. 4

De la Fig 4 se puede deducir que $dx = (b - a) / n$. Si n es suficientemente grande (delta sería suficientemente pequeño), el área de los trapecios será aproximadamente el área pedida. El área total que correspondería a la suma del área de cada uno de los trapecios se calcula de la siguiente forma:

- ✓ Se determinan los puntos del eje x que delimitarán cada trapecio. Estos puntos son:
 $x_i = a + i \cdot dx$, con $i = 0, 1, 2, \dots, n$
- ✓ Se evalúa la función f en cada uno de los puntos X_i :
 $y_i = f(x_i)$, $i = 0, 1, 2, \dots, n$
- ✓ Se calcula el área de cada trapecio como:
 $a_i = (y_i + y_{(i+1)}) \cdot dx / 2$, $i = 0, 1, 2, \dots, n-1$
- ✓ Se suman las áreas de cada uno de los trapecios.

DEDUCCION DEL MÉTODO DEL TRAPECIO: (Deducción del método desde los Polinomios de Interpolación)

Corresponde al caso donde $n=1$, es decir:

$$\int_a^b f(x) dx \approx \int_a^b f_1(x) dx$$

Donde $f_1(x)$, es un polinomio de interpolación (obviamente de grado 1) para los datos:

x	a	b
y	$f(a)$	$f(b)$

Del capítulo de interpolación y observando la Fig. 5, se sabe que este polinomio de interpolación puede expresarse mediante la expresión:

$$\frac{f(b) - f(a)}{b - a} = \frac{f(x) - f(a)}{x - a}$$

$$\int_a^b f(x)dx \approx (b-a) \left[\frac{f(a) + f(b)}{2} \right]$$

Que es conocida como la *Regla del Trapecio*. Este nombre se debe a la interpretación geométrica que se le puede dar a la fórmula. El polinomio de interpolación para una tabla que contiene dos datos, es una línea recta. La integral, corresponde al área bajo la línea recta en el intervalo $[a,b]$, que es precisamente el área del trapecio que se forma.

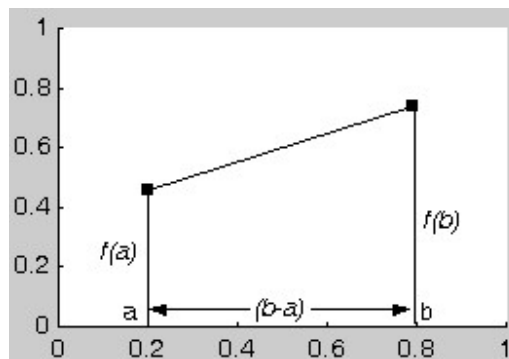


Fig. 6

DESARROLLO DEL MODELO

La *regla del trapecio* o *regla trapezoidal* es una de las fórmulas cerradas de *Newton-Cotes*.

Considérese la función $f(x)$, cuya gráfica entre los límites $x=a$ y $x=b$ se muestra en la Fig. 7. Una aproximación suficiente al área bajo la curva se obtiene dividiéndola en n subáreas de ancho ΔX y aproximando el área de cada una a de las secciones mediante un trapecio, como se indica en la figura.

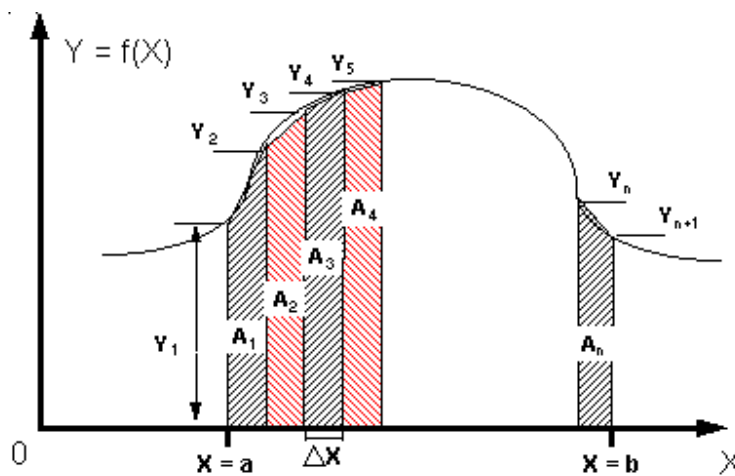


Fig. 7

El método de los trapecios es muy simple y se puede explicar de forma fácil a partir de la figura Fig 8. Suponga que se mira solo la pequeña subarea A_1 de la figura anterior Fig 7.

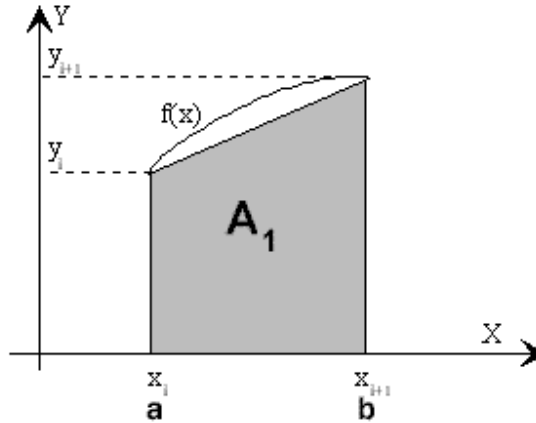


Fig. 8

$\Delta X = (b - a) / n$. Sería el ancho de cada una de las Sub áreas. n Sería el número de pequeñas sub áreas en las que se divide el área total que se desea calcular.

Llamando a las ordenadas Y_i ($i = 0, 1, 2, 3, \dots, n$), las áreas de los trapecios son:

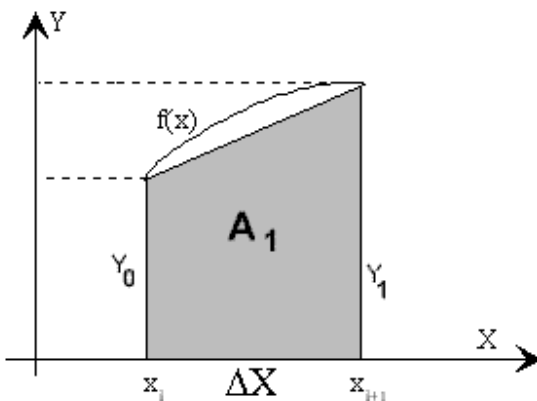


Fig. 9

$$\begin{aligned}
 A_1 &= \Delta X \left(\frac{Y_0 + Y_1}{2} \right) \\
 A_2 &= \Delta X \left(\frac{Y_1 + Y_2}{2} \right) \\
 &\vdots \\
 A_n &= \Delta X \left(\frac{Y_{n-1} + Y_n}{2} \right)
 \end{aligned} \tag{1}$$

El área total comprendida entre $X = a$ y $X = b$ está dada por:

$$A = \int_a^b f(x) dx \cong A_1 + A_2 + A_3 + \dots + A_n \tag{2}$$

Sustituyendo las ecuaciones (1) en la expresión (2) se obtiene:

$$A = \int_a^b f(x)dx \cong \frac{\Delta X}{2}(y_0 + 2y_1 + 2y_2 + \dots + 2y_{n-1} + y_n) \quad (3)$$

La cual recibe el nombre de *Fórmula Trapezoidal*, y se puede expresar como:

$$A = \int_a^b f(x)dx \cong \frac{\Delta X}{2}(y_0 + y_n + 2\sum (y_2 + y_3 + \dots + y_{n-1})) \quad (4)$$

$$A = \int_a^b f(x)dx \cong \frac{\Delta X}{2}\left(y_0 + y_n + 2\sum_{i=1}^{n-1} y_i\right) \quad (5)$$

Ahora se sabe que y_0 y y_n son valores de la evaluación de la función en cada uno de los límites, es decir y_0 es la función evaluada en el límite a y y_n es la función evaluada en el límite b .

$$A = \int_a^b f(x)dx \cong \frac{\Delta X}{2}\left(f(a) + f(b) + 2\sum_{i=1}^{n-1} y_i\right) \quad (6)$$

Ahora, y_i sería la evaluación en cada uno de los puntos sobre el eje x de base común a cada una de las sub áreas.

$$y_1 = f(a + 1dx)$$

$$y_2 = f(a + 2dx)$$

$$y_3 = f(a + 3dx)$$

⋮

⋮

⋮

$$y_i = f(a + idx)$$

Por lo tanto la ecuación general para el cálculo de la integral por el método trapezoidal será:

$$A = \int_a^b f(x)dx \cong \frac{\Delta X}{2}\left(f(a) + f(b) + 2\sum_{i=1}^{n-1} f(a + i * dx)\right) \quad (7)$$

Que también se pudiese representar como

$$A = \int_a^b f(x)dx \cong \Delta X\left(\frac{f(a) + f(b)}{2} + \sum_{i=1}^{n-1} f(a + i * dx)\right) \quad (8)$$

En esencia, la técnica consiste en dividir el intervalo total en intervalos pequeños y aproximar la curva $y = f(x)$ en los diversos intervalos pequeños mediante alguna curva más simple cuya integral puede calcularse utilizando solamente las ordenadas de los puntos extremos de los intervalos.

Si la función $f(x)$ se puede expresar como una función matemática continua que tiene derivadas continuas $f'(x)$ y $f''(x)$, el error que resulta de aproximar el área verdadera en una subárea bajo la curva $f(x)$ comprendida entre X_i y X_{i+1} mediante el área de un trapecio, se demuestra que es igual a:

$$e_i = -\frac{1}{12} f''(\xi)(\Delta x)^3 \text{ con } x_{i-1} < \xi < x_i \quad (9)$$

Este error es la cantidad que se debe agregar al área del trapecio para obtener el área real. Se llama *Error por Truncamiento*, ya que es el error que resulta de utilizar una serie de Taylor truncada, en vez de una serie de Taylor completa, para representar en forma de serie el área de una faja. Generalmente no se puede valorar directamente el término mostrado como error por truncamiento. Sin embargo, se puede obtener una buena aproximación de su valor para cada faja suponiendo que f'' es suficientemente constante en el intervalo de la faja (se supone que las derivadas de orden superior son despreciables) y evaluando f'' para $\xi = X_i$. La estimación del error por truncamiento para la integración total se obtiene sumando las estimaciones para cada faja. Si la estimación obtenida para el error total por truncamiento es mayor de lo que se puede tolerar, se debe utilizar una subárea más angosta o un método más preciso.

Otro error que se introduce al obtener el área aproximada de cada subárea es el *Error por Redondeo*. Este se produce cuando las operaciones aritméticas requeridas se efectúan con valores numéricos que tienen un número limitado de dígitos significativos.

Se puede demostrar que una aproximación al límite del error por redondeo es:

$$\left| e_R \right| \leq \frac{\bar{y}e(b-a)^2}{2} \left(\frac{1}{\Delta x} \right) \quad (10)$$

Se tiene entonces que el límite en el error por redondeo aumenta proporcionalmente a $(1/\Delta X)$, lo cual pronto domina al error por truncamiento que es proporcional a ΔX^2 . En realidad, el error por redondeo en sí no crece proporcionalmente con ΔX^{-1} sino con ΔX^{-p} en donde $0 < p < 1$, pero sin embargo aún supera al error por truncamiento si ΔX decrece lo suficiente.

El error por redondeo se puede minimizar utilizando aritmética de doble precisión o mediante compiladores que pueden manejar un gran número de dígitos significativos.

De la información anterior se puede ver que el *error total* en el intervalo de integración deseado, es la suma de los errores de truncamiento y redondeo. Si el *error total* se debiera únicamente al error por truncamiento, se podría hacer tan pequeño como se deseara reduciendo suficientemente el ancho de la faja. Por ejemplo, bisectando el ancho de la faja se duplicaría el número de errores por truncamiento que hay que sumar, pero la expresión para el error en cada faja indica que cada uno sería aproximadamente un octavo de su valor previo.

Sin embargo, disminuyendo el ancho de la faja se afectaría también el error total al aumentar el error por redondeo, debido al mayor número de operaciones que hay que efectuar al evaluar la ec. (3). Entonces, cuando se disminuye el ancho de la faja para disminuir el error total, existe un punto óptimo en el cual disminuciones adicionales del ancho de la faja harían que el error aumentara en lugar de disminuir, porque el error por redondeo se volvería dominante. El ancho óptimo de la faja para una función especial se puede determinar fácilmente en forma experimental en la computadora (suponiendo que el área real bajo la gráfica de la función se puede evaluar) pero es difícil definirlo analíticamente.

Ejemplo: Utilizar la regla del trapecio para aproximar la integral: $\int_0^1 e^{x^2} dx$.

Tenga en cuenta que el valor real es 1.4626...

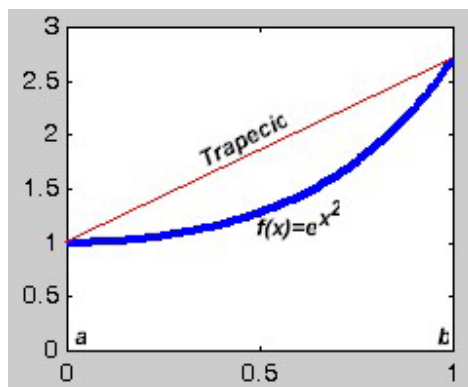


Fig. 10

Solución: Usando la fórmula directamente con los siguientes datos:

$$a = 0$$

$$b = 1$$

$$f(x) = e^{x^2}$$

Si se asume el área como un solo trapecio, se tiene que:

$$\int_0^1 e^{x^2} dx \approx (1-0) \left[\frac{f(0)+f(1)}{2} \right] = \frac{1+e}{2} = 1.85914$$

Observando la Fig. 10 y teniendo en cuenta el resultado obtenido con el método trapezoidal tomando el área como un solo trapecio (1.85914), se puede comprobar que dicho valor es superior al valor real que es de 1.4626. El valor real es el área bajo la curva azul, que corresponde a la función dada y el valor calculado de 1.85914 corresponde al área bajo la curva de color rojo, que correspondería a la forma que toma el área asumiéndola como trapecio.

Desarrollado en MatLab se tendría el siguiente resultado.

```

>>syms x
>>f=exp(x^2);
>>integral=int(f)
integral =
    -1/2*i*pi^(1/2)*erf(i*x)
    
```

ERF Error de la función.

Y = ERF(X) es el error de la función para cada elemento de X. X debe ser real. El error de la función está definido como:

$$\text{erf}(x) = 2/\sqrt{\pi} * \text{integral desde } 0 \text{ a } x \text{ de } \exp(-t^2) dt. \text{ Analice lo anterior.}$$

Ejemplo: Usar la regla del trapecio para aproximar la integral: $\int_2^4 \frac{e^x}{x} dx$

Solución: Igual que en el ejemplo anterior, se sustituyen los datos de manera directa en la fórmula del trapecio. En este caso, se tienen los datos:

$$a=2$$

$$b=4$$

$$f(x) = e^x / x$$

Por lo tanto, se tiene que:

$$\int_2^4 \frac{e^x}{x} dx \approx (4-2) \left[\frac{f(2)+f(4)}{2} \right] = \frac{e^2}{2} + \frac{e^4}{4} = 17.3441$$

La regla del trapecio se puede ampliar si se subdivide el intervalo $[a,b]$ en n subintervalos, todos de la misma longitud $h=(b-a)/n$.

Sea $P=\{x_0, x_1, x_2, \dots, x_n\}$ la partición que se forma al hacer dicha subdivisión. Usando propiedades de la integral se tiene que:

$$\int_a^b f(x)dx = \int_{x_0}^{x_1} f(x)dx + \int_{x_1}^{x_2} f(x)dx + \dots + \int_{x_{n-1}}^{x_n} f(x)dx$$

Aplicando la regla del trapecio en cada una de las integrales, se tiene:

$$\int_a^b f(x)dx \approx (x_1 - x_0) \left[\frac{f(x_0) + f(x_1)}{2} \right] + \dots + (x_n - x_{n-1}) \left[\frac{f(x_{n-1}) + f(x_n)}{2} \right]$$

Ahora bien, ya que todos los subintervalos tienen la misma longitud h , se tiene que:

$$\int_a^b f(x)dx \approx \frac{h}{2} [f(x_0) + 2f(x_1) + 2f(x_2) + \dots + 2f(x_{n-1}) + f(x_n)]$$

Sustituyendo el valor de h y usando la notación sigma, se tiene finalmente:

$$\int_a^b f(x)dx \approx (b - a) \left[\frac{f(x_0) + 2 \sum_{i=1}^{n-1} f(x_i) + f(x_n)}{2n} \right]$$

Esta es la regla del trapecio para n sub intervalos. Obviamente, se espera que entre más sub intervalos use, mejor sea la aproximación a la integral.

Ejemplo: Aplicar la regla del trapecio para aproximar la integral $\int_0^1 e^{x^2} dx$ si se subdivide en 5 intervalos.

Solución: En este caso, se identifica $n=5$, y la partición generada es:

$$P=\{0.0, 0.2, 0.4, 0.6, 0.8, 1.0\}$$

Así, aplicando la fórmula se tiene que:

$$\int_0^1 e^{x^2} dx \approx (1 - 0) \left[\frac{f(0) + 2[f(0.2) + f(0.4) + f(0.6) + f(0.8)] + f(1)}{2 * 5} \right]$$

$$\int_0^1 e^{x^2} dx \approx 1 \left[\frac{1 + 2[e^{0.2^2} + e^{0.4^2} + e^{0.6^2} + e^{0.8^2}] + e}{10} \right]$$

$$\int_0^1 e^{x^2} dx \approx 1.48065$$

Cabe mencionar que el valor verdadero de esta integral es de 1.4626...

Así, se nota que con 5 intervalos, la aproximación no es tan mala. Para hacer cálculos con más subintervalos, es conveniente elaborar un programa que aplique la fórmula con el número de subintervalos que se desee y que permita obtener un valor más cercano al real. Debería realizar su propio programa y chequear con 50, 500, 1000, 10000 y 20000 subintervalos, para observar el comportamiento de la aproximación.

Ejemplo: Evaluar la función $\int_0^{1.2} f(x) dx$, usando la siguiente tabla:

x	0	0.10	0.30	0.50	0.70	0.95	1.20
f(x)	0	6.84	4.00	4.20	5.51	5.77	1.0

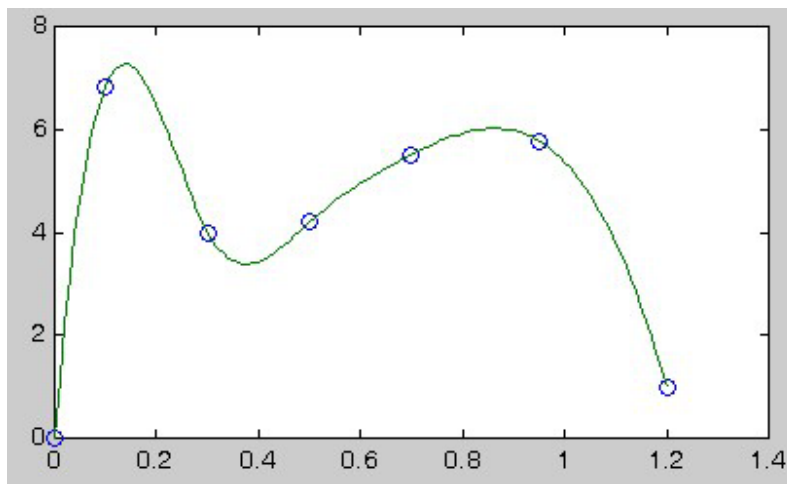


Fig 11

Solución.

Observe en la fig 9 que en el intervalo [0,0.1] se puede aplicar la regla del trapecio, en el intervalo [0.1,0.7] la regla de Simpson de 3/8 y en el intervalo [0.7,1.2] la regla de Simpson de 1/3. Así, se tienen las siguientes integrales:

$$I_1 = \int_0^{0.1} f(x)dx = \frac{0.1-0}{2} [f(0) + f(0.1)] = 0.842$$

$$I_2 = \int_{0.1}^{0.7} f(x)dx = \frac{0.7-0.1}{8} [f(0.1) + 3f(0.3) + 3f(0.5) + f(0.7)] = 2.7712$$

$$I_3 = \int_{0.7}^{1.2} f(x)dx = \frac{1.2-0.7}{6} [f(0.7) + 4f(0.95) + f(1.2)] = 2.4658$$

Finalmente, la integral buscada es la suma de las tres integrales anteriores:

$$\int_0^{1.2} f(x)dx = 0.842 + 2.7712 + 2.4658 = 6.079$$

Ejemplo: Calcula la integral $\int_{-1}^{3.25} f(x)dx$, usando la siguiente tabla de datos:

x	-1	-0.5	0	1	1.75	2.5	3.25
$f(x)$	2	-3	1.5	-1	0.5	0.75	-2

Si se desarrolla la gráfica con Matlab ajustando los datos de la tabla anterior mediante splines cúbicos, se tendría lo siguiente:

- » `x=[-1 -0.5 0 1 1.75 2.5 3.25];`
- » `y=[2 -3 1.5 -1 0.5 0.75 -2];`
- » `xx=-1:0.01:3.5;`
- » `yy=spline(x,y,xx);`
- » `plot(x,y,'o',xx,yy)`

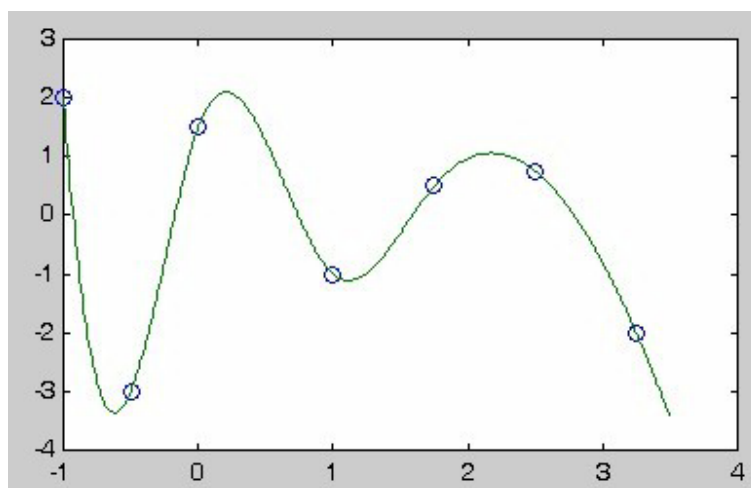


Fig. 12

Solución combinando varios métodos de integración.

Para este caso, se puede aplicar la regla de Simpson de 1/3 en el intervalo [-1,0], la regla del trapecio en el intervalo [0,1] y la regla de Simpson de 3/8 en el intervalo [1,3.25]. Así, se tiene las siguientes integrales:

$$I_1 = \int_{-1}^0 f(x)dx \approx \frac{0 - (-1)}{6} [f(-1) + 4f(-0.5) + f(0)] \approx -1.41667$$

$$I_2 = \int_0^1 f(x)dx \approx \frac{1-0}{2} [f(0) + f(1)] \approx 0.25$$

$$I_3 = \int_1^{3.25} f(x)dx \approx \frac{3.25-1}{8} [f(1) + 3f(1.75) + 3f(2.5) + f(3.25)] \approx 0.210938$$

Por lo tanto, la integral buscada es la suma de las tres integrales anteriores:

$$\int_{-1}^{3.25} f(x)dx \approx -1.41667 + 0.25 + 0.210938 \approx -0.955729$$

Vale la pena comentar que no siempre tiene que suceder que se apliquen exactamente las tres reglas. En realidad, esto depende de cómo se encuentran espaciados los intervalos de la tabla de datos y la forma que pueda tener la curva.

Programación del método de los trapecios bajo lenguaje C (Regla Trapecial o regla Trapezoidal)

Suponga que la función *f* evalúa la función *f(x)*. Entonces la siguiente fórmula permite calcular el área de los trapecios:

$$area = dx [f(a)/2 + f(x_1) + f(x_2) +f(x_{n-1}) + f(b)/2]$$

Que en últimas sería la misma fórmula 8 descrita anteriormente.

Programa en lenguaje C: Se supone que la función *area* se debe llamar con los parámetros *a,b* y *n*, que serían límite inferior, límite superior y número de sub áreas respectivamente.

```
double area(double a, double b, int n)
{
    double dx= (b-a)/n;
    double sum= (f(a)+f(b))/2; // 1/2*(f(a)+f(b))
    double x= a;
    for (int i=1; i<n; i++) // + f(a+i*delta), con i=1 ... n-1
```

```

    { x+= dx;          // x=a+i*dx
      sum+= f(x);
    }
    return dx*sum;
}
double f(double x)
{ double y;
  y = ...           // Se describiría la función a la cual se le
// desea calcular la integral
  return y;
}

```

Esta solución es análoga a la que se programó para buscar los ceros de una función. El problema de esta solución, es que cuando se desea calcular la integral de varias funciones distintas, hay que programar una función para calcular el área de cada función.

Cálculo del área de múltiples funciones en base a subclases

En lenguaje C o Builder de consola, se puede ejecutar el siguiente código para que vea el calculo de la integral $f(x)=x^2$. Si desea encontrar la integral de otra función debe cambiar la función en el procedimiento f sobre la parte final del programa.

```

#include <vcl.h>
#pragma hdrstop
#include <stdio.h>
#include <conio.h>
#include <math.h>
void Lee_Datos(void);
double f(double x);
int n; // Variable para el numero de intervalos
float a,b; // Variables para os limites de la integral
#pragma argsused
int main(int argc, char* argv[])
{ float dx;
  double Area;
  double Suma = 0;
  double Xi;
  int i;
  clrscr();
  printf("\n MÉTODO DE INTEGRACIÓN POR TRAPÉCIOS ");
  printf("\n=====");
  printf("\n Dada una Función  $f(x)=x^2$  ");
  printf("\n");
  Lee_Datos();
}

```

```

dx = (b-a)/n;
Xi = a;
for ( i = 1; i < n; i++)
{ Xi += dx;
  Suma += 2*f(Xi);
}
Area = dx/2 *( f(a) + Suma + f(b));
printf("\n-----");
printf("\n Area bajo La Curva es => %.8lf ",Area);
getche();
return 0;
}
void Lee_Datos(void)
{ printf("\n Numero de Subareas => n : ");
  scanf("%d",&n);
  printf("\n Valor de Límite inferior => a : ");
  scanf("%f",&a);
  printf("\n Valor de Límite Superior => b : ");
  scanf("%f",&b);
}

double f(double x)
{ return x*x; // Ecuación a la cual se le calcula el área bajo la curva
}

```

Si la idea es crear objetos para evaluar múltiples funciones, entonces cada uno de estos objetos poseerá un método **eval** para evaluar la función asignada. La definición del método se hará en una subclase de una clase base común a todas las funciones. Se llamará a esta clase **Función**:

```

class Funcion extends Program { double eval(double x)
{ return 0.0; }
  double area(double a, double b, int n)
{ // el mismo método de antes
  return ...;
}
}

```

Con esta clase ahora es posible calcular en un solo programa el área de $\sin(x)$, de $\exp(x)$ y de x^k :

```

class FSin extends Funcion { double eval(double x)
{ return sin(x); }
  // area se hereda
}
class FExp extends Funcion { double eval(double x)

```

```

    { return exp(x); }
    // area se hereda
}
class FXk extends Funcion { int k; FXk(int k)
{ this.k= k; }
double eval(double x)
{ return pow(x, k); }
// area se hereda
}
...
void run()
{ println("area sin(x)= " + new FSin().area(0.0, PI/2, 100));
  println("area exp(x)= " + new FExp().area(0.0, 1.0, 100));
  println("area x^3= " + new FXk(3).area(0.0, 1.0, 100));
}

```

En la clase Función también se pueden incluir otros métodos para:

- Graficar un intervalo de la función en una pizarra.
- Buscar el mínimo en un intervalo de la función.
- Buscar el máximo en un intervalo de la función.
- Buscar una raíz de la función en un intervalo.

O si mejor desea utilizar MatLab ó Scilab entonces el código sería así, Se debe aclarar que la función debe estar contenida en un archivo nombrado funcion.m en un directorio que este configurado en el Path del MatLab.

```

clc
format long
echo on
% La función a evaluar esta contenida dentro del archivo 'funcion.m' y se
requiere introducir los valores de los límites, como se indica a continuación
pulsa una tecla...
pause
n= input('Digite el número de subareas => n :');
a= input('Digite el Límite inferior => a :');
b= input('Digite el Límite superior => b :');
dx=(b-a)/n;
suma=0;
% Algoritmo para la sumatoria de la regla trapezoidal de segmentos múltiples
% pulsa una tecla...
pause
for l=1:n-1
    xi=a+i*dx
    suma=suma+funcion(xi);
end

```

```
% pulsa una tecla...
pause
% Forma general de la regla trapezoidal de n suba reas.
c=(b-a)*(funcion(a)+2*suma+funcion(b))/2/n
```

Yamil Armando Cerquera Rojas
yacerquel@hotmail.com

Ingeniero Especialista en Sistemas U. Nacional de Colombia
Docente Universidad Surcolombiana
Colombia, Huila - Rivera