Integración numérica

En los cursos de Cálculo Integral aprendemos a calcular una integral definida de una función continúa haciendo uso del Teorema Fundamental del Cálculo que dice que si f(x) es una función continua en un intervalo [a, b] y $F(x)$ es una antiderivada de $f(x)$ entonces:

$$\int_a^b f(x)\,dx = F(b) - F(a)$$

El problema en la práctica se presenta, cuando se nos hace imposible mediante métodos analíticos determinar la antiderivada requerida, aun cuando se trate de integrales aparentemente sencillas que son imposibles de resolver con el Teorema Fundamental del Cálculo.

En estos casos, debemos de recurrir a la Integración Numérica. Dentro del cálculo numérico, la Integración Numérica comprende una amplia familia de algoritmos para el cálculo del valor numérico de una integral definida. En la mayoría de los casos, ese valor numérico es un valor aproximado de la integral definida.

Puede haber varias razones por la cuales se desee o se necesite calcular el valor numérico aproximado de una integral definida:

- La función integrando es desconocida, pero se conocen algunos puntos de la función, por ejemplo, puntos de datos obtenidos experimentalmente.
- La función integrando no tiene función primitiva, por ejemplo: $f(x) = e^{-x^2}$.
- La función primitiva es conocida, pero es más conveniente o más sencillo calcular numéricamente la integral definida.
-

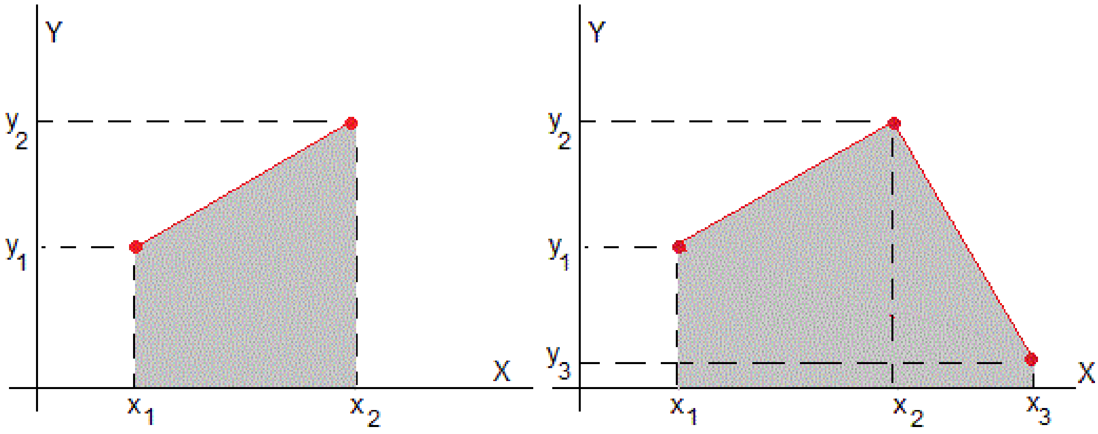
Se denominan Métodos de Newton-Cotes a los que se basan en integrar, en lugar de la función dada $f(x)$, un polinomio de interpolación que aproxime a $f(x)$ en [a, b]. Se trata por tanto de toda una familia general de métodos, según el polinomio de interpolación que se considere (puede elegirse diferente grado, diferentes puntos para interpolar, etc.). Para el caso de las interpolaciones lineal y cuadrática, estos métodos se denominan Método de los Trapecios y Método de Simpson, respectivamente.

Se presentarán los métodos:

- Método del punto medio
- El método del trapecio
- El método de Simpson. El Método de Simpson sustituye a la curva $y = f(x)$ por una serie de arcos contiguos, cada uno de estos arcos es un arco de parábola de eje vertical. Esto nos lleva a aproximar el área bajo la curva mediante la suma de las áreas bajo cada arco de parábola.

Antes de introducir al lector en al cálculo del área comprendida entre una función $y = f(x)$ y el eje X en un intervalo dado [*a*,*b*]. Vamos a ver como se calcula el área de un polígono como suma de las áreas de trapecios.

Áreas



El área del trapecio de la figura de la izquierda es la suma de dos áreas: un rectángulo y un triángulo

$$(x_2 - x_1)\, y_1 + \frac{(x_2 - x_1)(y_2 - y_1)}{2} = \frac{(x_2 - x_1)(y_2 + y_1)}{2}$$

El área de la figura de la derecha formada por los puntos (x_1,y_1), (x_2,y_2) y (x_3,y_3) es

$$\frac{(x_2 - x_1)(y_2 + y_1)}{2} + \frac{(x_3 - x_2)(y_3 + y_2)}{2}$$

$(x_2-x_1)\cdot(y_2+y_1)/2+(x_3-x_2)\cdot(y_3+y_2)/2$

Para *n* puntos, el área es

$$area = \frac{1}{2}\sum_{i=1}^{i=n-1}(x_{i+1} - x_i)(y_{i+1} + y_i)$$

Supongamos que en una experiencia hemos medido la velocidad de un móvil en función del tiempo,

t(s)	1	2	3	4	5	6	7	8	9	10
v(m/s)	5.0	6.0	5.5	7.0	8.3	7.6	6.2	6.1	7.0	5.7

calculamos el desplazamiento aproximado del móvil sumando áreas y utilizando la función ***trapz***.

myode.m | RK_RLC.m | dif_perio.m | dif_periosim.m

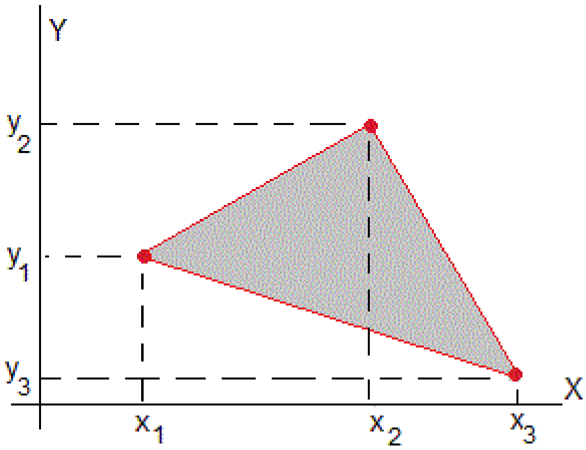
```
x=1:10;
y=[5.0,6.0,5.5,7.0,8.3,7.6,6.2,6.1,7.0,5.7];
area=0;
for i=1:length(x)-1
      area=area+(x(i+1)-x(i))*(y(i)+y(i+1))/2;
end
disp(['Area: ', num2str(area)]);
area=trapz(x,y);
disp(['Area: ', num2str(area)]);
```

```
>> aa
Area: 59.05
Area: 59.05
>>
```

```
>> help trapz
trapz - Trapezoidal numerical integration

    This MATLAB function computes the approximate integral of Y via the trapezoidal
    method with unit spacing.

    Q = trapz(Y)
    Q = trapz(X,Y)
    Q = trapz(___,dim)
```

Área de un polígono

El área del triángulo formado por tres puntos (x_1,y_1), (x_2,y_2) y (x_3,y_3) es



$$\frac{(x_2-x_1)(y_2+y_1)}{2}+\frac{(x_3-x_2)(y_3+y_2)}{2}-\frac{(x_3-x_1)(y_3+y_1)}{2}=$$

$$=\frac{(x_2-x_1)(y_2+y_1)}{2}+\frac{(x_3-x_2)(y_3+y_2)}{2}+\frac{(x_1-x_3)(y_1+y_3)}{2}$$

Para un polígono de *n* lados, formado por los puntos (x_1,y_1), (x_2,y_2) ... (x_n,y_n) el área se calcula mediante la fórmula

$$area=\frac{1}{2}\left[(x_1-x_n)(y_1+y_n)+\sum_{i=1}^{i=n-1}(x_{i+1}-x_i)(y_{i+1}+y_i)\right]$$

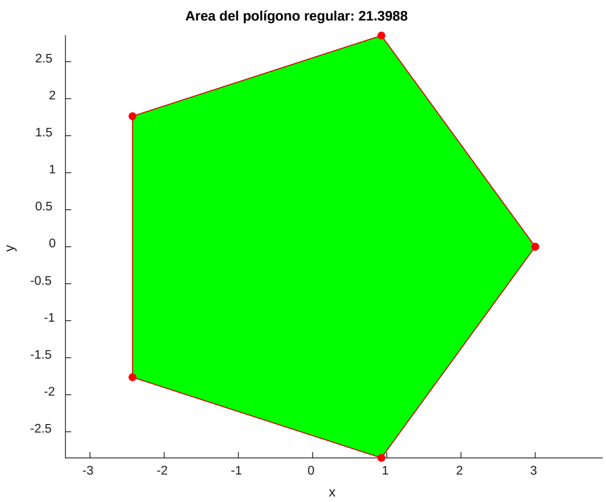
Para cerrar la figura poligonal, añadimos el vértice *n*+1, que coincide con el primer vértice (x_1,y_1), (x_2,y_2) ... (x_n,y_n), (x_1,y_1). El cálculo del área se simplifica

$$area=\frac{1}{2}\left[\sum_{i=1}^{i=n}(x_{i+1}-x_i)(y_{i+1}+y_i)\right]$$

Consideremos primero un polígono regular, un pentágono. Calculamos el área mediante la función MATLAB *polyarea* y también, mediante la fórmula que hemos obtenido anteriormente

```matlab
n=5; %número de lados
radio=3; %radio de la circunferencia
x=radio*cos((0:n)*2*pi/n);
y=radio*sin((0:n)*2*pi/n);
%las coordenadas del último vértice son las del primer vértice
hold on
fill(x,y,'g')
plot(x,y,'-ro', 'markersize',4,'markeredgecolor','r','markerfacecolor','r')
hold off
xlabel('x')
ylabel('y')
axis equal
area=0;
for i=1:n
    area=area+(x(i+1)-x(i))*(y(i+1)+y(i))/2;
end
area=abs(area)
title (['Area del polígono regular: ' num2str(polyarea(x,y))])
```

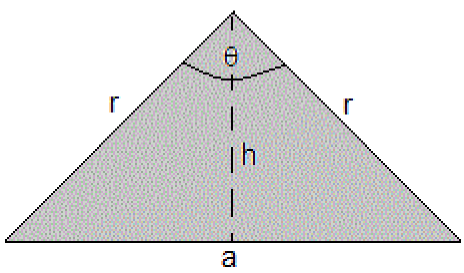
```
>> poli

area =

   21.3988
```



El área de un polígono regular de *n* inscrito una circunferencia de radio *r* es la suma del área de *n* triángulos isósceles de vértice $\vartheta=2\pi/n$.

$$A=\frac{n(ah)}{2}$$

Como vemos en la figura

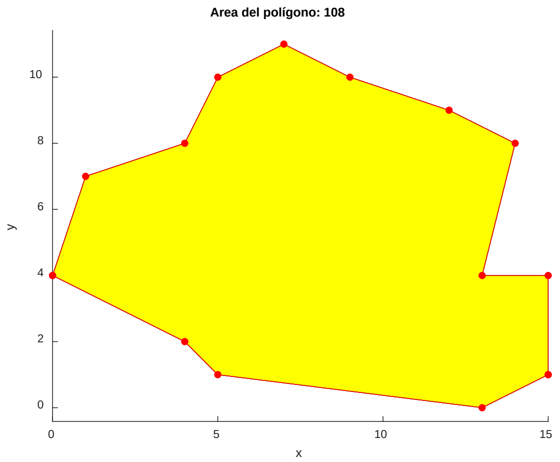


$$a=2r\sin(\theta/2)$$
$$h=r\cdot\cos(\theta/2)$$
$$A=n\,\frac{ah}{2}=\frac{1}{2}n\,r^2\sin\theta=\frac{1}{2}n\,r^2\sin\left(\frac{2\pi}{n}\right)$$

```
>> area=n*radio^2*sin(2*pi/n)/2

area =

   21.3988
```

Consideramos ahora, un polígono cualquiera

```
x = [15,13,5,4,0,0,1,4,5,7,9,12,14,13,15,15];
y = [1,0,1,2,4,4,7,8,10,11,10,9,8,4,4,1];
%las coordenadas del último vértice son las del primer vértice
hold on
fill(x,y,'y')
plot(x,y,'-ro', 'markersize',4,'markeredgecolor','r','markerfacecolor','r')
hold off
xlabel('x')
ylabel('y')
axis equal
area=0;
for i=1:length(x)-1
    area=area+(x(i+1)-x(i))*(y(i+1)+y(i))/2;
end
title (['Area del polígono: ' num2str(area)])
```

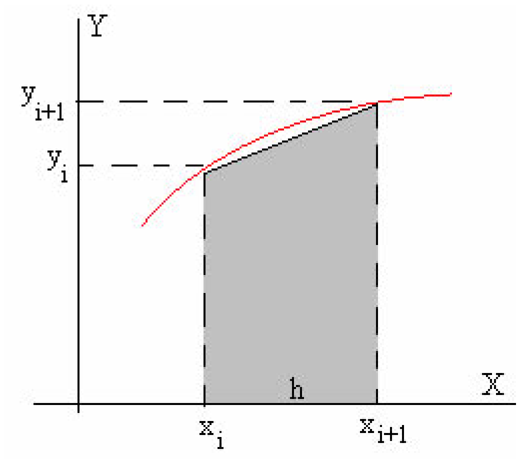


La función MATLAB *polyarea* solamente precisa los *n* vértices del polígono, dos vectores *x* e *y* de dimensión *n*

```
>> polyarea(x,y)

ans =

   108
```

Método del punto medio
Fórmula del trapecio

La regla del trapecio es uno de los métodos más utilizados para calcular aproximaciones numéricas de integrales definidas. Es la primera de las fórmulas cerradas de integración de Newton – Cotes, para el caso cuando el polinomio interpolante es de grado uno



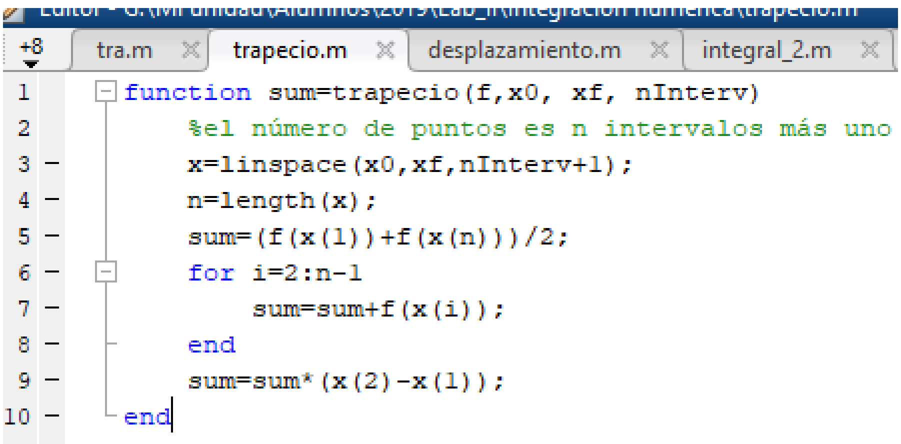
Para calcular la integral definida de la función *f*(*x*) en el intervalo comprendido entre x_0 y x, dividimos este intervalo en pequeños intervalos de longitud $h = x_{i+1} - x_i$. Sustituimos la función por la recta que une los puntos (x_i, y_i) y (x_{i+1}, y_{i+1}). El área sombreada en la figura es la suma del área de un rectángulo más el área de un triángulo, vale

$$hy_i + \frac{1}{2} h(y_{i+1} - y_i) = \frac{1}{2} h(y_{i+1} + y_i)$$

El área total aproximada bajo la curva es

$$\frac{1}{2} h(y_1 + y_2) + \frac{1}{2} h(y_2 + y_3) + \ldots + \frac{1}{2} h(y_{n-1} + y_n) = \frac{1}{2} h(y_1 + 2y_2 + 2y_3 + \ldots + 2y_{n-1} + y_n)$$

Definimos en la función *trapecio* el método. Donde *f* es la función integrando y *x* el vector de *n*+1 datos comprendidos entre la abscisa inicial x_0 y la final x_f ambas incluidas, si el número de intervalos es *n*.




```
function sum=trapecio(f,x0, xf, nInterv)
    %el número de puntos es n intervalos más uno
    x=linspace(x0,xf,nInterv+1);
    n=length(x);
    sum=(f(x(1))+f(x(n)))/2;
    for i=2:n-1
        sum=sum+f(x(i));
    end
    sum=sum*(x(2)-x(1));
end
```

Dada la función integrando *f*(*x*), calculamos la aproximación a la integral definida en el intervalo comprendido entre x_0 y x.

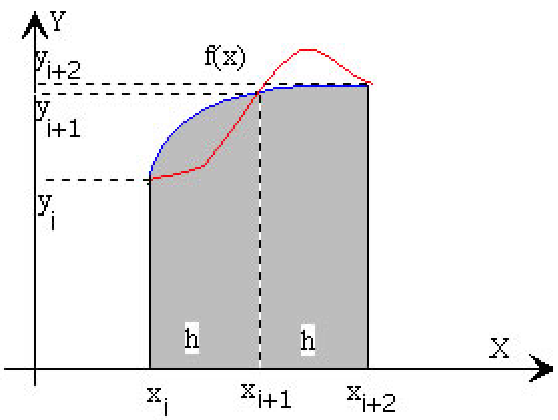
```
x0=input('abscisa inicial, x0: ');
xf=input('abscisa final, xf: ');
n=input('número intervalos, n: ');

deriv=@(x) -x^2+14*x+21;  %definición del integrando
res=trapecio(deriv,x0,xf,n); %calcula la integral
fprintf('El valor aproximado de la integral: %3.2f\n',res)
```

En la ventana de comandos corremos el script

```
>> tra
abscisa inicial, x0: 0
abscisa final, xf: 10
número intervalos, n: 10
El valor aproximado de la integral: 575.00
```

El método de Simpson (1/3)

Además de la regla del trapecio, otra manera de obtener una estimación más exacta de una integral es utilizar polinomios de orden superior para conectar los puntos. El Método de Simpson es un método de Newton-Cotes de segundo orden, es decir basado en integrar un polinomio de interpolación de segundo grado. En este método, se toma el intervalo de anchura $2h$, comprendido entre x_i y x_{i+2}, y se sustituye la función $f(x)$ por la parábola que pasa por tres puntos (x_i, y_i), (x_{i+1}, y_{i+1}), y (x_{i+2}, y_{i+2}).



Calculamos la contribución a la integral del primer intervalo (x_0, x_0+2h) y generalizaremos para el resto de los intervalos.

La ecuación de la parábola $y=ax^2+bx+c$ que pasa por los puntos (x_0, y_0), (x_0+h, y_1), (x_0+2h, y_2) es

$$y_0 = ax_0^2 + bx_0 + c$$
$$y_1 = a\left(x_0 + h\right)^2 + b\left(x_0 + h\right) + c$$
$$y_2 = a\left(x_0 + 2h\right)^2 + b\left(x_0 + 2h\right) + c$$

Este sistema de tres ecuaciones con tres incógnitas, se reduce a

$$y_1 = y_0 + 2ax_0h + ah^2 + bh$$
$$y_2 = y_0 + 4ax_0h + 4ah^2 + 2bh$$

Despejamos el coeficiente *a*, y $2ax_0+b$

$$a = \frac{y_2 - 2y_1 + y_0}{2h^2}$$
$$2ax_0 + b = \frac{4y_1 - 3y_0 - y_2}{2h}$$

Sustituimos la curva por la porción de parábola en el intervalo $\left(x_0,\ x_0 + 2h\right)$.

La integral vale.

$$I = \int_{x_0}^{x_0+2h} \left(ax^2 + bx \ + c\right)dx = \frac{h}{3}\left(y_0 + 4y_1 + y_2\right)$$

En general, el valor del área aproximada, en el intervalo $\left(x_i, x_i + 2h\right)$ sombreada en la figura, es

$$\frac{h}{3}\left(y_i + 4y_{i+1} + y_{i+2}\right)$$

El área aproximada en el intervalo (*a*, *b*) es

$$I = \int_a^b f\left(x\right)dx \approx \frac{h}{3}\left(y_0 + 4y_1 + y_2\right) + \frac{h}{3}\left(y_2 + 4y_3 + y_4\right) + \ldots + \frac{h}{3}\left(y_{n-2} + 4y_{n-1} + y_n\right)$$

o bien, agrupando términos

$$I = \int_a^b f\left(x\right)dx \approx \frac{h}{3}\left[\left(y_0 + y_n + +y_2\right) + 4\left(y_1 + y_3 + \ldots + y_{n-1}\right) + \ldots + 2\left(y_2 + y_4 + \ldots + y_{n-2}\right)\right]$$

El primer paréntesis, contiene la suma de los extremos, el segundo, la suma de los términos de índice impar, y el tercero la suma de los términos de índice par. En el método de Simpson, el número de divisiones *n* debe de ser par.

Definimos en la función *simpson* el método. Donde *f* es la función integrando y *x* el vector de *n*+1 datos comprendidos entre la abscisa inicial x_0 y la final x_f ambas incluidas, si el número de intervalos es *n* que tiene que ser un número PAR.

Editor - G:\Mi unidad\Alumnos\2019\Lab_II\Integracion numerica\simpson.m

+10 | int_sim.m | simpson.m | tra.m | trapecio.m | desplazamiento.m | integ

```
function suma=simpson(f,x0,xf,n)
%n número par de intervalos, n+1 número de puntos en el vector
    x=linspace(x0,xf,n+1);
    h=x(2)-x(1);
    suma=f(x(1))+f(x(n+1));
    for i=2:2:n
            suma=suma+4*f(x(i));
        end
        for i=3:2:n-1
            suma=suma+2*f(x(i));
        end
        suma=suma*h/3;
end
```

Dada la función integrando *f*(*x*), calculamos la aproximación a la integral definida en el intervalo comprendido entre x_0 y x

Editor - G:\Mi unidad\Alumnos\2019\Lab_II\Integracion numerica\int_sim.m

+10 | int_sim.m | simpson.m | tra.m | trapecio.m | desplazamiento.m

```
x0=input('abscisa inicial, x0: ');
xf=input('abscisa final, xf: ');
n=input('número intervalos (par), n: ');
if rem(n,2)==1
  disp('El número intervalos tiene que ser par ');
  return
end

deriv=@(x) -x^2+14*x+21;  %definición del integrando
res=simpson(deriv,x0,xf,n); %calcula la integral
fprintf('El valor aproximado de la integral: %3.2f\n',res)
```

En la ventana de comandos corremos el script

Command Window

```
>> int_sim
abscisa inicial, x0: 0
abscisa final, xf: 10
número intervalos (par), n: 10
El valor aproximado de la integral: 576.67
>>
```

Apreciamos la mejora en el resultado de la integral con el método de Simpson.

Existe una versión generalizada para cada método. Si el intervalo en el que se realiza la integral es grande, el Método de los Trapecios Simple suele ser muy impreciso. Para mejorar la exactitud, es posible subdividir el intervalo en otros más pequeños y aplicar en cada uno de ellos el Método simple. De manera completamente análoga a lo expuesto para el Método de los Trapecios, es posible generalizar (mejorando la precisión) el Método de Simpson por medio de la subdivisión del intervalo dado en otros más reducidos.

Como todo método de aproximación existe un error de truncamiento, que para el caso del método de trapecio es

$$|E| \leq \left| \frac{h^3}{12} M_2 \right|$$

Donde M_2 es el máximo de la derivada segunda de la función en el intervalo. En el caso del Método de Simpson se tiene

$$E \leq \left| \frac{h^5}{90} M_4 \right|$$

Donde M_4 es el máximo de la derivada cuarta de la función en el intervalo.

Para el caso de los métodos compuestos se deben de considerar el error de cada subdivisión del intervalo

$$E \leq \left| \frac{h^5}{90} \left(M_4^1 + M_4^2 + \ldots + M_4^{n/2} \right) \right| \leq \left| \frac{h^5}{90} \frac{n}{2} M_4 \right| = \left| \frac{h^5}{90} \frac{1}{2} \frac{b-a}{h} M_4 \right|$$

$$E \leq \left| \frac{b-a}{180} h^4 M_4 \right|$$