

---

# P9: CASAS Y JIMENEZ

## Table of Contents

1. Exactitud: cálculo de integrales con Simpson compuesta y error. ....	1
2. Cálculo de pi integrando con Simpson compuesta. ....	5
Opcional: integración mediante el sistema de pesos de Vandermonde. ....	5

## 1. Exactitud: cálculo de integrales con Simpson compuesta y error.

```
clear all;
close all;
format long g;

%{
La función que nos devolverá la integral definida utilizando
el método de integración compuesta de simpson es la siguiente:

function I = simpson(a, b, m, fun)
%   Retorna la integral definida utilizant el metode de integracio
    compost.
%   a i b son els intervals d'integracio.
%   fun es la funcio que volem integrar.

H=(b-a)/m ; %distancia entre nodes.
k=[0:2*m];
xk=[a+(H.*k)./2];

% Sumem 1 al primer terme de cada sumatori perquè les arrays comencen
    a 1.
sum1 = 0;
sum2 = 0;
s = 0;
for r=3:2:2*m
    sum1= sum1 + fun(xk(r));
    s = r-1;
    sum2 = sum2 + fun(xk(s));
end
sum2 = sum2 + fun(xk(s+2));

I = (H/6) * (fun(xk(1)) + 2*sum1 + 4*sum2 + fun(xk(end)));
%}

fun1 = @(x)(8*x^7);
fun2 = @(x)(5*x^4);
fun3 = @(x)(4*x^3);

errorf1 = [];
```

```
errorf2 = [];  
errorf3 = [];  
  
a = 0;  
b = 1;  
  
Ms = [1 10 20 40 60 70 80];  
  
%Per calcular l'error cal saber el valor exacte de la integral (fer-la  
a ma):  
result1=1; result2=1; result3=1;  
  
for m = Ms  
    I1 = simpson(0, 1, m, fun1);  
    errorf1 = [errorf1 abs(result1 - I1)];  
  
    I2 = simpson(0, 1, m, fun2);  
    errorf2 = [errorf2 abs(result2 - I2)];  
  
    I3 = simpson(0, 1, m, fun3);  
    errorf3 = [errorf3 abs(result3 - I3)];  
end  
  
figure;  
loglog(Ms, errorf1);  
title('I_7: Error de Simpson');  
xlabel('precisi3n de la malla (m)');  
ylabel('error (E_n)');  
[r1,m1,~] = regression(log10(Ms),log10(errorf1))  
  
figure;  
loglog(Ms, errorf2);  
title('I_4: Error de Simpson');  
xlabel('precisi3n de la malla (m)');  
ylabel('error (E_n)');  
[r2,m2,~] = regression(log10(Ms),log10(errorf2))  
  
figure;  
loglog(Ms, errorf3);  
title('I_3: Error de Simpson');  
xlabel('precisi3n de la malla (m)');  
ylabel('error (E_n)');  
  
%{  
Observamos que, dado que estamos integrando con un Simpson  
de grado 4, el error decrece en ese orden cuando el grado de  
la integral a aproximar es superior a 4 (m1,m2~4) y tiene un  
error cero (o pr3cticamente, seg3n la precisi3n de la m3quina)  
cuando el grado de la integral es inferior a 4.  
%}  
  
r1 =
```

$-0.999916244975825$

$m1 =$

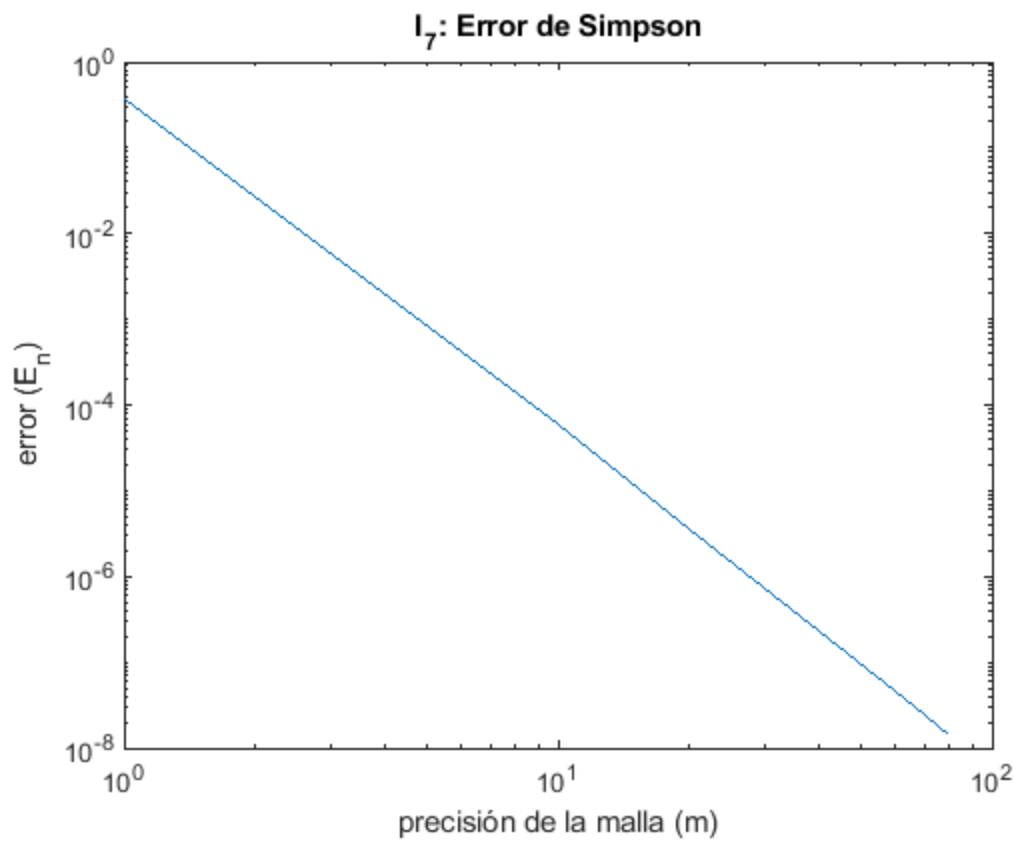
$-3.90566637802679$

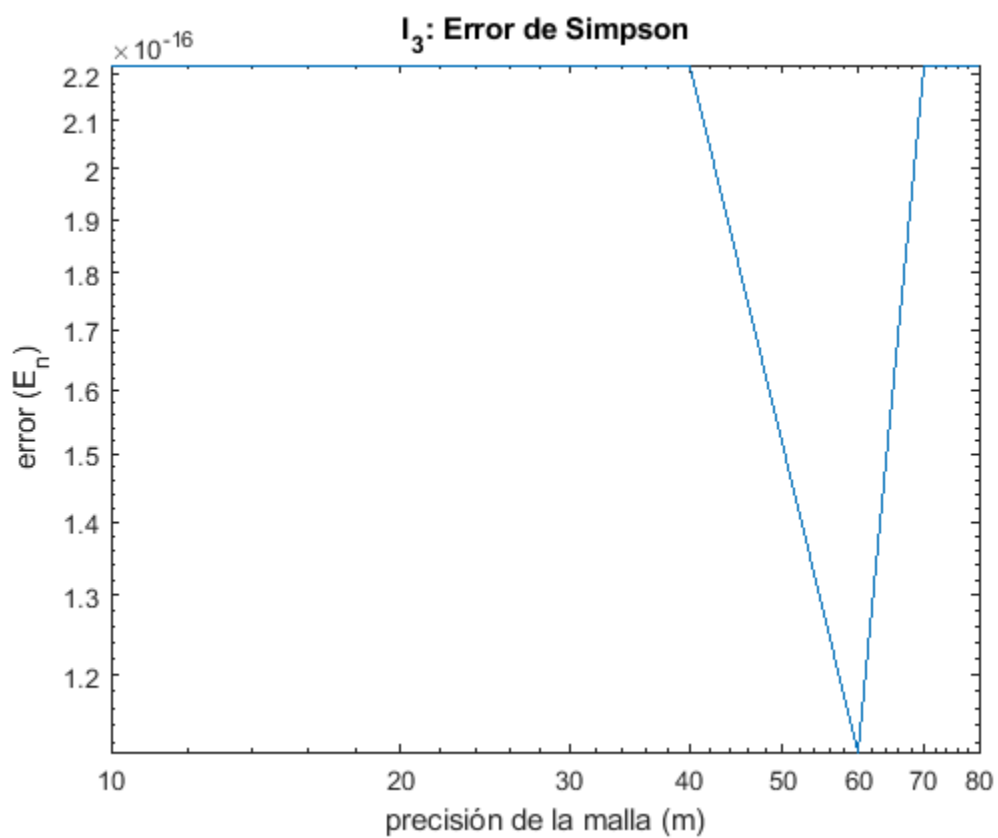
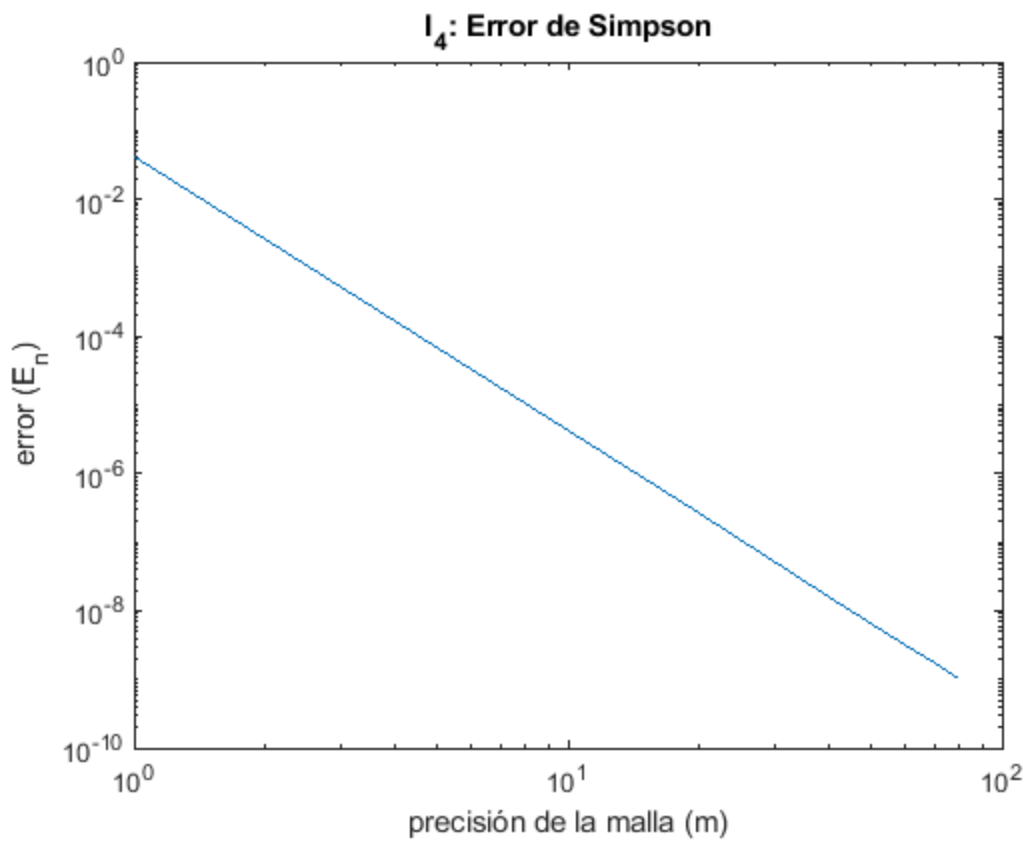
$r2 =$

$-1$

$m2 =$

$-3.99999999552869$





## 2. Cálculo de pi integrando con Simpson compuesta.

```
%{
En este apartado encontraremos pi resolviendo la integral
immediata  $I=4/(1+x^2)$  mediante la regla de Simpson compuesta.

Para determinar pi con catorce cifras significativas, tendremos
que conseguir una precisión (~m) tal que el error del cálculo
numérico respecto al pi original sea menor a  $10^{-14}$ .
%}
clear all;
funcioPi = @(x)(4/(1+x^2));
a=1;
result=pi; %es el resultado exacto de la integral.
errorPi = [];

I=0;%para iniciar el bucle; es evidente que no podemos integrar en 0
nodos.
m = 0;
while abs(I - result) > 10^-14
    m = m + 1;
    I = simpson(0, a, m, funcioPi);
end
% La m mínima para obtener 14 cifras significativas será:
m
% Y el valor de pi que obtendremos es:
I = simpson(0, a, m, funcioPi);
I

m =

    63

I =

    3.14159265358978
```

## Opcional: integración mediante el sistema de pesos de Vandermonde.

```
%{
En este último apartado deberemos resolver la misma integral
del apartado anterior (la cual debe proporcionarnos el número
pi) pero esta vez utilizando los pesos de cuadratura.

Para obtener el vector de pesos, deberemos tener en cuenta que:
```

```
MV * Ws' = Is
```

Y que, de este modo, podremos obtener el vector de Ws resolviendo el sistema donde MV y Is son conocidos.

Una vez construida la matriz transpuesta de Vandermonde MV y generado el vector de valores de la integral (en este caso de una integral polinómica) podremos resolver el sistema, hallar el vector de pesos Ws y aplicarlo al cálculo de la integral  $I=4/(1+x^2)$ .

Por último, consideraremos también el error en la aproximación a pi mediante este método.

```
%}  
clear all;  
  
result = pi; %es el resultat exacte de la integral  
Ns = 1:27;  
errorPi=[];  
funcioPi = @(x)(4./(1+x.^2));  
  
for n=Ns  
    xk=linspace(0,1,n+1); %utilizamos nodos equiespaciados.  
  
    %Construimos la matriz del sistema:  
    MV = zeros(1,n+1);  
    for i = 1:1:n+1  
        MV(i,:) = xk.^(i-1);  
    end  
    %Y también el vector de términos independientes:  
    k = 0:1:n;  
    Is = 1./(k+1);  
    Is = Is';  
  
    %Por lo que podemos resolver el sistema y hayar los pesos Ws:  
    Ws = MV\Is;  
  
    %Aplicamos Ws para aproximar el valor de la integral "funcioPi":  
    fj = funcioPi(xk);  
    nPi_aprox = fj * Ws;  
    errorPi = [errorPi abs(result - nPi_aprox)];  
end  
  
figure;  
loglog(Ns, errorPi);  
title('Error de la aproximación mediante pesos (w_j)')  
xlabel('número de nodos (n)');  
ylabel('error (E_n)');  
[r,m,~] = regression(log10(Ns),log10(errorPi))  
% Decae con potencia p=-10.
```

*Warning: Matrix is close to singular or badly scaled. Results may be inaccurate.*

*RCOND = 1.561138e-16.*

Warning: Matrix is close to singular or badly scaled. Results may be inaccurate.

RCOND = 1.969579e-17.

Warning: Matrix is close to singular or badly scaled. Results may be inaccurate.

RCOND = 2.420401e-18.

Warning: Matrix is close to singular or badly scaled. Results may be inaccurate.

RCOND = 3.075698e-19.

Warning: Matrix is close to singular or badly scaled. Results may be inaccurate.

RCOND = 4.041849e-20.

Warning: Matrix is close to singular or badly scaled. Results may be inaccurate.

RCOND = 2.838625e-21.

Warning: Matrix is close to singular or badly scaled. Results may be inaccurate.

RCOND = 1.597423e-21.

Warning: Matrix is close to singular or badly scaled. Results may be inaccurate.

RCOND = 1.983305e-21.

Warning: Matrix is close to singular or badly scaled. Results may be inaccurate.

RCOND = 1.822603e-22.

Warning: Matrix is close to singular or badly scaled. Results may be inaccurate.

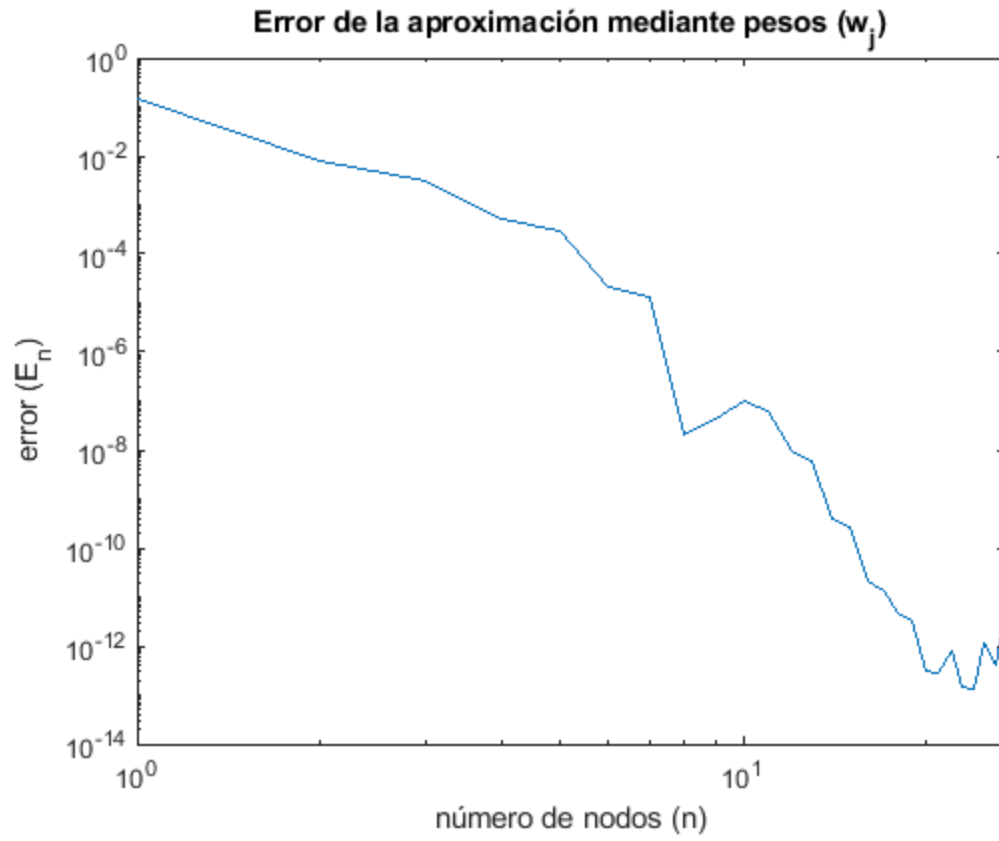
RCOND = 5.674535e-23.

$r =$

-0.958732322966283

$m =$

-9.87553017115229



*Published with MATLAB® R2018b*