

Output:

PUNTO 1

a)

La probabilidad de que esté entre $[-\sigma, \sigma]$, es:

Punto medio: 0.682722

Simpson Compuesto: 0.682690

Trapecio compuesto: 0.682625

Errorpm =

3.226577484161108e-05

Errorsimp =

1.377331604324183e-08

Errortrap =

6.452896829334254e-05

b)

La probabilidad de que esté entre $[-2\sigma, 2\sigma]$, es:

Punto medio: 0.954557

Simpson Compuesto: 0.954500

Trapecio compuesto: 0.954385

|
Errorpm =

5.757959774366217e-05

Errorsimp =

4.948184684216983e-08

Errortrap =

1.151684257548746e-04

c)

La probabilidad de que esté entre $[-3\sigma, 3\sigma]$, es:

Punto medio: 0.997316

Simpson Compuesto: 0.997300

Trapezio compuesto: 0.997268

Errorpm =

1.591446394000329e-05

Errorsimp =

1.834719924875827e-07

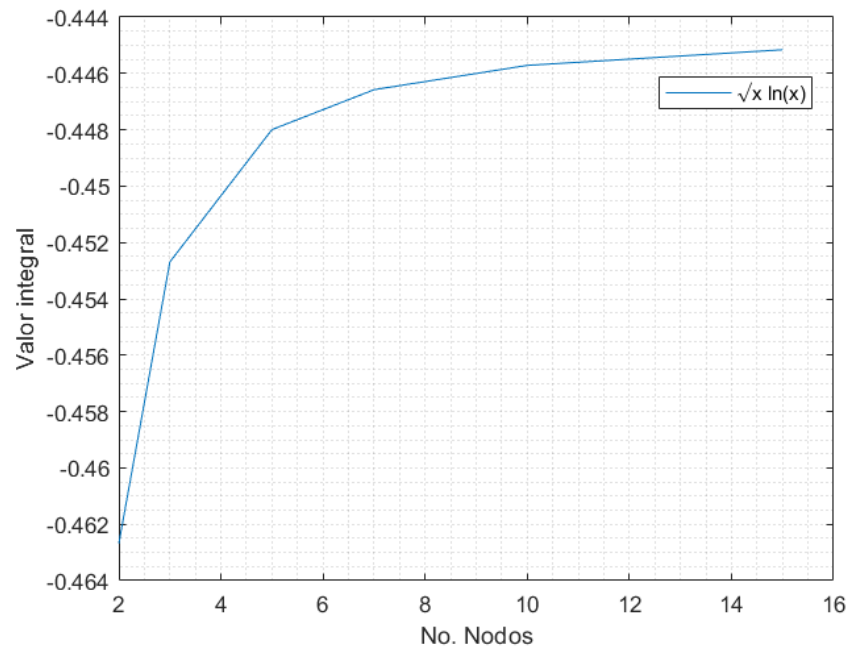
Errortrap =

3.186337504068604e-05

PUNTO 2

Empleando el método de simpson encontramos que la distancia recorrida será: 9858.00 pies

PUNTO 3



CODIGO

```
format long
clear all
clc

disp('PUNTO 1')

f = @(x) 1/sqrt(2*pi)*exp(-1/2*x.^2) ;

disp('a')

a = -1 ; b = 1 ; N = 50 ;
fprintf("\n La probabilidad de que esté entre [-sigma,sigma], es:
\n")
pm = punto_medio_comp(f,a,b,N) ;
simp = simpson_comp(f,a,b,N) ;
trap = trapecio_comp(f,a,b,N) ;
fprintf(' Punto medio: %0.6f \n Simpson Compuesto: %0.6f \n
Trapecio compuesto: %0.6f \n\n',pm,simp,trap)
Errorpm = abs(integral(f,a,b)-pm)
Errorsimp = abs(integral(f,a,b)-simp)
Errortrap = abs(integral(f,a,b)-trap)
disp('b')

a = -2 ; b = 2 ; N = 50 ;
fprintf("\n La probabilidad de que esté entre [-2sigma,2sigma],
es: \n")
pm = punto_medio_comp(f,a,b,N) ;
simp = simpson_comp(f,a,b,N) ;
trap = trapecio_comp(f,a,b,N) ;
fprintf(' Punto medio: %0.6f \n Simpson Compuesto: %0.6f \n
Trapecio compuesto: %0.6f \n\n',pm,simp,trap)
Errorpm = abs(integral(f,a,b)-pm)
Errorsimp = abs(integral(f,a,b)-simp)
Errortrap = abs(integral(f,a,b)-trap)
disp('c')

a = -3 ; b = 3 ; N = 50 ;
fprintf("\n La probabilidad de que esté entre [-3sigma,3sigma],
es: \n")
pm = punto_medio_comp(f,a,b,N) ;
simp = simpson_comp(f,a,b,N) ;
trap = trapecio_comp(f,a,b,N) ;
fprintf(' Punto medio: %0.6f \n Simpson Compuesto: %0.6f \n
Trapecio compuesto: %0.6f \n\n',pm,simp,trap)
Errorpm = abs(integral(f,a,b)-pm)
Errorsimp = abs(integral(f,a,b)-simp)
Errortrap = abs(integral(f,a,b)-trap)

disp('PUNTO 2')
Tiempo = [0,6,12,18,24,30,36,42,48,54,60,66,72,78,84] ;
```

```

Velocidad =
[124,134,148,156,147,133,121,109,99,85,78,89,104,116,123] ;
N = size(Tiempo,2)-1 ;
h = (Tiempo(end)-Tiempo(1))/N ;
Impar = 0 ; Par = 0 ;
    for i=2:N
        if ((-1)^i)==1
            Impar=Impar+Velocidad(i) ;
        else
            Par=Par+Velocidad(i) ;
        end
    end
Distancia = h/3*(Velocidad(1)+4*Impar+2*Par+Velocidad(end)) ;
fprintf('Empleando el método de simpson encontramos que la
\ndistancia recorrida será: %.2f pies \n',Distancia)

disp('PUNTO 3')
f1 = @(x) sqrt(x).*log(x) ;
f2 = @(x) exp(-x).*sin(x)./x ;
f3 = @(x) exp(-x.^4) ;

%Nodos
m = [2 3 5 7 10 15] ;
%Particiones
N = m-1 ;
for i=1:size(m,2)
    Result(i,1) = cuadratura_gaussiana(f1,0,1,N(i)) ;
    Result(i,2) = cuadratura_gaussiana(f2,0,inf,N(i)) ;
    Result(i,3) = cuadratura_gaussiana(f3,-inf,inf,N(i)) ;
end

plot(m,Result)
xlabel('No. Nodos')
ylabel('Valor integral')
grid minor

function g = punto_medio_comp(f,a,b,N)
    h = (b-a)/N ;
    x = linspace(a,b,N+1) ;
    x_prom = (x(2:end)+x(1:end-1))/2 ;
    g = h*sum(f(x_prom)) ;
end

function g = simpson_comp(f,a,b,N)
    h = (b-a)/N ;
    x = linspace(a,b,N+1) ;
    Impar = 0 ; Par = 0 ;
    for i = 2:N
        if ((-1)^i)==1
            Impar=Impar+f(x(i)) ;
        else
            Par=Par+f(x(i)) ;
        end
    end
    g = h*(Impar+4*Par)/3 ;
end

```

```

        end
    end
    g = h/3*(f(a)+4*Impar+2*Par+f(b)) ;
end

function g = trapecio_comp(f,a,b,N)
    h = (b-a)/N ;
    x = linspace(a,b,N+1) ;
    g = h/2*(f(a)+f(b)+2*sum(f(x(2:N)))) ;
end

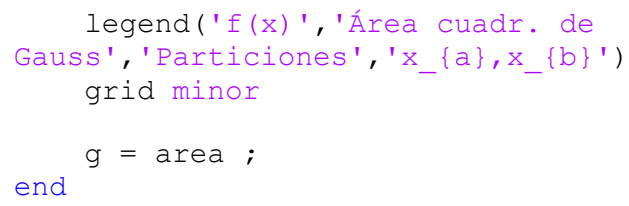
function g = CuadGauss2p(f,a,b)
    x0 = -1/sqrt(3) ;
    x1 = -x0 ;
    xa = (b+a)/2 + (b-a)/2*x0 ;
    xb = (b+a)/2 + (b-a)/2*x1 ;
    g = ((b-a)/2)*(f(xa) + f(xb)) ;
end

function g = cuadratura_gaussiana(f,a,b,N)
    x = linspace(a,b,N+1) ;
    area = 0 ;
    %Para las gráficas
    x0 = -1/sqrt(3) ;
    x1 = -x0 ;
    for i=1:N
        deltaA = CuadGauss2p(f,x(i),x(i+1)) ;
        area = area + deltaA ;
    end

    figure('Color','White')
    xgraf = linspace(a-a/10,b+b/10,100) ;
    recta = zeros(N,2) ;
    hold on
    for i=1:N
        %sub intervalos
        xa(i) = (x(i+1)+x(i))/2 + (x(i+1)-x(i))/2*x0 ;
        xb(i) = (x(i+1)+x(i))/2 + (x(i+1)-x(i))/2*x1 ;
        %rectas entre subintervalos
        m = (f(xb(i))-f(xa(i)))/(xb(i)-xa(i)) ;
        b0 = f(xa(i)) - m*xa(i) ;
        recta(i,:) = b0 + m*[x(i) x(i+1)] ;
        Area = [0 recta(i,:) 0] ;
        xi = [x(i) x(i:i+1)' x(i+1)] ;
        plot(xgraf,f(xgraf),'r','LineWidth',1.5)
        fill(xi,Area,'w') ;
        xline(x(i),'b','LineWidth',1.1) ;
        plot([xa(i) xa(i)],[0 f(xa(i))],'-ok',[xb(i) xb(i)],[0
f(xb(i))],'-ok')
    end
    plot(xgraf,f(xgraf),'r','LineWidth',1.3)

```

```
    legend('f(x)', 'Área cuadr. de  
Gauss', 'Particiones', 'x_{a}, x_{b}')
```



```
    grid minor  
  
    g = area ;  
end
```