

# Introducción a MATLAB

Martín Josemaría Vuelta Rojas

## Problema 1

Utilizando MATLAB, haga un programa (function) que evalúe las funciones singulares: impulso unitario, escalón unitario y función rampa. Debe graficar cada función singular.

## Solución

### Script 1 Función impulso unitario

```
function f = impulso(x,y,z)
    switch (nargin)
        case 1, f = 1.*(x>-eps)-1.*(x>eps);
        case 2, f = 1.*(x>(y-eps))-1.*(x>(y+eps));
        case 3, f = z.*(x>(y-eps))-z.*(x>(y+eps));
        otherwise
            fprintf('Error: Revise los argumentos de entrada')
    end
```

### Script 2 Función escalón unitario

```
function f = escalon(x,y,z)
    switch (nargin)
        case 1, f = 1.*(x>=0);
        case 2, f = 1.*(x>=y);
        case 3, f = z.*(x>=y);
        otherwise
            fprintf('Error: Revise los argumentos de entrada')
    end
```

### Script 3 Función rampa

```
function f = rampa(x,y,z)
    switch (nargin)
        case 1, f = x.*(x>=0);
        case 2, f = (x-y).*(x>=y);
        case 3, f = z*(x-y).*(x>=y);
        otherwise
            fprintf('Error: Revise los argumentos de entrada')
    end
```

---

**Script 4** Programa de lectura de la matriz de cuadrado mágico y determinación del máximo valor y su lugar dentro de la matriz.

---

```
Dim = input('Ingrese la dimension de la matriz: ');

FileName = sprintf('magico_%d.dat',Dim);

fprintf('Leyendo archivo %s\n', FileName);

MagicMatrix = load(FileName);
maxValue = max(max(MagicMatrix));
[r, c] = find(MagicMatrix == maxValue, 1, 'first');

fprintf('Mayor elemento esta en la posicion (%d,%d) y es %d\n', r, c, maxValue);
fprintf('La matriz cargada fue:\n')
disp(MagicMatrix)
```

---

---

**Script 5** Ejemplo de ejecución de los programas mostrados en los *scripts* [1](#) y [4](#)

---

```
>> problema01_a
Ingrese la dimension de la matriz: 6
Escribiendo archivo magico_6.dat
Escritura completa.
La matriz guardada fue:
    35     1     6    26    19    24
     3    32     7    21    23    25
    31     9     2    22    27    20
     8    28    33    17    10    15
    30     5    34    12    14    16
     4    36    29    13    18    11

>> problema01_b
Ingrese la dimension de la matriz: 6
Leyendo archivo magico_6.dat
Mayor elemento esta en la posicion (6,2) y es 36
La matriz cargada fue:
    35     1     6    26    19    24
     3    32     7    21    23    25
    31     9     2    22    27    20
     8    28    33    17    10    15
    30     5    34    12    14    16
     4    36    29    13    18    11
```

---

## Problema 2

Hacer un programa para resolver la ecuación de segundo grado:  $ax^2 + bx + c = 0$ . Los parámetros  $a$ ,  $b$  y  $c$  serán introducidos desde el teclado. Debe tener en cuenta las raíces reales y complejas. Las raíces deben aparecer en la pantalla con 6 decimales. No debe usar la sentencia `roots`.

### Solución

---

#### Script 6 Cálculo de soluciones para la ecuación de segundo grado.

---

```
a = input('Ingrese valor del coeficiente a: ');
b = input('Ingrese valor del coeficiente b: ');
c = input('Ingrese valor del coeficiente c: ');

if(a == 0 && b ~= 0)
    fprintf('El coeficiente del termino cuadratico es nulo. La ecuacion es lineal.\n');
    disp('La raiz de la ecuacion b*x + c = 0 es :');
    fprintf('X[1] = %2.6f.\n', -c/b);

elseif(a == 0 && b == 0)
    fprintf('Los coeficientes de los terminos cuadratico y lineal son nulos.\n');
    fprintf('La ecuacion no se puede resolver\n');

else
    delta = b^2 - 4*a*c;

    disp('Las raices de la ecuacion a*x^2 + b*x + c = 0 son :');

    if(delta == 0)
        fprintf('X[1] = %2.6f\n', -b/(2*a));
        fprintf('X[2] = %2.6f\n', -b/(2*a));

    elseif(delta > 0)
        ra = -b/(2*a);
        ir = sqrt(delta)/(2*a);

        if(ra == 0)
            fprintf('X[1] = + %2.6f\n', ir);
            fprintf('X[2] = - %2.6f\n', ir);
        else
            fprintf('X[1] = %2.6f + %2.6f\n', ra, ir);
            fprintf('X[2] = %2.6f - %2.6f\n', ra, ir);
        end
    end
    re = -b/(2*a);
    im = sqrt(-delta)/(2*a);

    if(ra == 0)
        fprintf('X[1] = + %2.6f\n', im);
        fprintf('X[2] = - %2.6f\n', im);
    else
        fprintf('X[1] = %2.6f + %2.6f\n', re, im);
        fprintf('X[2] = %2.6f - %2.6f\n', re, im);
    end
end
end
end
```

---

---

**Script 7** Ejemplo de ejecución del programa mostrado en el *script 6*

---

```
>> problema02
Ingrese valor del coeficiente a: 1
Ingrese valor del coeficiente b: 5
Ingrese valor del coeficiente c: 3
Las raices de la ecuacion  $a*x^2 + b*x + c = 0$  son :
X[1] = -2.500000 + 1.802776
X[2] = -2.500000 - 1.802776

>> problema02
Ingrese valor del coeficiente a: -8
Ingrese valor del coeficiente b: 3
Ingrese valor del coeficiente c: -20
Las raices de la ecuacion  $a*x^2 + b*x + c = 0$  son :
X[1] = 0.187500 + -1.569982
X[2] = 0.187500 - -1.569982

>> problema02
Ingrese valor del coeficiente a: 0
Ingrese valor del coeficiente b: 1
Ingrese valor del coeficiente c: 1
El coeficiente del termino cuadratico es nulo. La ecuacion es lineal.
La raiz de la ecuacion  $b*x + c = 0$  es :
X[1] = -1.000000.

>> problema02
Ingrese valor del coeficiente a: 0
Ingrese valor del coeficiente b: 0
Ingrese valor del coeficiente c: 5
Los coeficientes de los terminos cuadratico y lineal son nulos.
La ecuacion no se puede resolver
```

---

## Problema 3

Hacer un programa para resolver un sistema de ecuaciones lineales:  $A \cdot X = Y$ , donde  $A$  es una matriz cuadrada y  $X$  e  $Y$  son vectores columna. Los datos serán leídos desde un archivo. Las incógnitas deben aparecer en la pantalla con 4 decimales. Debe grabar las incógnitas en un archivo `solucion.txt`.

### Solución

---

#### Script 8 Programa para la solución de un sistema de ecuaciones lineales

---

```
Dim      = input('1. Numero de ecuaciones del sistema: ');
InFile   = input('2. Nombre del archivo entrada: ', 's');
OutFile  = input('3. Nombre del archivo salida: ', 's');

fprintf('\n')

if isempty(Dim)
    Dim = 5;
    fprintf('* Cargando valor por defecto: %d ecuaciones\n', Dim);
end

if isempty(InFile)
    InFile = 'problema03_data.dat';
    fprintf('* Cargando valor por defecto: Archivo de entrada "%s".\n', InFile);
end

if isempty(OutFile)
    OutFile = 'solucion.txt';
    fprintf('* Cargando valor por defecto: Archivo de salida "%s".\n', OutFile);
end

if(InFile == -1)
    fprintf('\n')
    disp('Archivo no encontrado');
else
    B = load(InFile);

    if(any(size(B) ~= [Dim, Dim+1]))
        fprintf('\n')
        fprintf('Los datos en el archivo "%s" no corresponden a un sistem de %d
↩ ecuaciones.\n', InFile, Dim);
    else
        A = B(:,1:Dim);

        fprintf('\n')
        fprintf('Matiz del sistema: A*X = Y\n')
        fprintf('A = \n')

        for i = 1:Dim
            for j = 1:Dim
                fprintf('\t %10.4f', A(i,j));
            end
            fprintf('\n');
        end

        Y = B(:,Dim+1);
```

---

Continúa en la página siguiente.

```
fprintf('\n')
fprintf('Y = \n')

for i = 1:Dim
    fprintf('\t %10.4f \n', Y(i));
end

if(det(A) == 0)
    fprintf('\n');
    fprintf('La matriz del sistema es singular.\n')
else
    X = A\Y;

    fprintf('\n');
    fprintf('La solución del sistema es:\n')
    fprintf('X = \n')

    SolutionFile = fopen(OutFile, 'w');

    for i = 1:Dim
        fprintf('\t %10.4f \n', X(i));
        fprintf(SolutionFile, '%2.4f\n', X(i));
    end

    if(~fclose(SolutionFile))
        fprintf('\n');
        fprintf('La solución del problema sistema se guardó exitosamente en ↵
↵ "%s"\n', OutFile);
    else
        fprintf('\n');
        fprintf('Error al guardar la solución en "%s"\n', OutFile);
    end
end
end
end
```

---

**Script 9** Ejemplo de ejecución del programa mostrado en el *script* 8

---

```
>> problema03
1. Numero de ecuaciones del sistema: 5
2. Nombre del archivo entrada: problema03_data.dat
3. Nombre del archivo salida: solucion.txt

Matiz del sistema: A*X = Y
A =
      21.5707      6.8034      42.6516      3.7983      20.8634
      45.5324      43.4646      31.1028      11.9958      2.4827
      9.0924      28.9852      17.5476      6.1659      45.1358
      13.1901      27.4930      25.6625      9.1954      47.2394
      7.2769      7.2477      20.0904      11.9976      24.5432

Y =
      24.4626
      16.8860
      45.0027
      18.4623
      5.5601

La solucion del sistema es:
X =
      -40.0155
      24.7443
      24.8792
      2.6207
      -16.8627

La solucion del problema sistema se guardo exitosamente en "solucion.txt"
```

---

## Problema 4

Hacer un programa para calcular la distancia entre dos puntos geográficos de latitud y longitud determinados. Considerar que la Tierra tiene una forma esférica y que la distancia **no** es una línea recta, sino una longitud de arco esférica. ¿Cuál es la distancia entre Lima y New York? Verifique con Google Earth.

**Sugerencia:**  $L = R \times \theta$ , donde  $\theta$  es el ángulo formado por los vectores que van del centro a los puntos geográficos.

## Solución

---

### Script 10 Cálculo de distancias entre dos puntos geográficos sobre la Tierra

---

```
Lat = 1;
Lon = 2;
R   = 3;

Deg = 1;
Min = 2;
Sec = 3;

X   = 1;
Y   = 2;
Z   = 3;

Rt  = 6371.0;
City1 = zeros(3:3);
City2 = zeros(3:3);

fprintf('Ingrese las coordenadas geograficas de las ciudades\n');
fprintf('Las latitudes hacia el norte y longitudes hacia el este se ingresan con ↵
↵  numeros positivos\n');
fprintf('Las latitudes hacia el sur y longitudes hacia el oeste se ingresan con ↵
↵  numeros negativos\n');

fprintf('Ciudad 1:\n')
fprintf('* Latitud\n')
City1(Lat,Deg) = input(' * Grados : ');
City1(Lat,Min) = input(' * Minutos : ');
City1(Lat,Sec) = input(' * Segundos : ');
fprintf('* Longitud\n')
City1(Lon,Deg) = input(' * Grados : ');
City1(Lon,Min) = input(' * Minutos : ');
City1(Lon,Sec) = input(' * Segundos : ');

fprintf('Ciudad 2:\n')
fprintf('* Latitud\n')
City2(Lat,Deg) = input(' * Grados : ');
City2(Lat,Min) = input(' * Minutos : ');
City2(Lat,Sec) = input(' * Segundos : ');
fprintf('* Longitud\n')
City2(Lon,Deg) = input(' * Grados : ');
City2(Lon,Min) = input(' * Minutos : ');
City2(Lon,Sec) = input(' * Segundos : ');
```

---

Continúa en la página siguiente.



Continuación del *script* 10.

---

```
Rlat = (City1(Lat,Deg) + City1(Lat,Min)/60 + City1(Lat,Sec)/3600)*pi/180;
Rlon = (City1(Lon,Deg) + City1(Lon,Min)/60 + City1(Lon,Sec)/3600)*pi/180;

City1(R,X) = Rt*cos(Rlat)*cos(Rlon);
City1(R,Y) = Rt*cos(Rlat)*sin(Rlon);
City1(R,Z) = Rt*sin(Rlat);

Rlat = (City2(Lat,Deg) + City2(Lat,Min)/60 + City2(Lat,Sec)/3600)*pi/180;
Rlon = (City2(Lon,Deg) + City2(Lon,Min)/60 + City2(Lon,Sec)/3600)*pi/180;

City2(R,X) = Rt*cos(Rlat)*cos(Rlon);
City2(R,Y) = Rt*cos(Rlat)*sin(Rlon);
City2(R,Z) = Rt*sin(Rlat);

D = Rt*acos((City1(R,:)*City2(R,:)))/(norm(City1(R,:))*norm(City2(R,:)));
fprintf('La distancia entre ambos puntos es igual a %.4f km\n', D);
```

---

---

### Script 11 Ejemplo de ejecución del programa mostrado en el *script* 10

---

```
>> problema04
Ingrese las coordenadas geograficas de las ciudades
Las latitudes hacia el norte y longitudes hacia el este se ingresan con numeros
↵ positivos
Las latitudes hacia el sur y longitudes hacia el oeste se ingresan con numeros negativos
Ciudad 1:
* Latitud
  * Grados : 40.78139796148296
  * Minutos : 0
  * Segundos : 0
* Longitud
  * Grados : -73.96668954467765
  * Minutos : 0
  * Segundos : 0
Ciudad 2:
* Latitud
  * Grados : -12.067156558830048
  * Minutos : 0
  * Segundos : 0
* Longitud
  * Grados : -77.0886956434249
  * Minutos : 0
  * Segundos : 0
La distancia entre ambos puntos es igual a 5885.2712 km
```

---

---

**Script 12** Script de Javascript que emplea la api de Google Maps para el cálculo de distancias entre dos puntos en el mapa

---

```
var marker1, marker2;
var poly, geodesicPoly;

function initMap() {
    map = new google.maps.Map(document.getElementById('map'), { zoom: 3 });

    map.controls[google.maps.ControlPosition.TOP_CENTER].push(
        document.getElementById('info'));

    marker1 = new google.maps.Marker({
        map: map,
        draggable: true,
        position: {
            lat: 40.78139796148296,
            lng: -73.96668954467765
        }
    });

    marker2 = new google.maps.Marker({
        map: map,
        draggable: true,
        position: {
            lat: -12.067156558830048,
            lng: -77.0886956434249
        }
    });

    var bounds = new google.maps.LatLngBounds(
        marker1.getPosition(), marker2.getPosition());

    map.setCenter({
        lat: (marker1.position.lat() + marker2.position.lat())*0.5,
        lng: (marker1.position.lng() + marker2.position.lng())*0.5
    })

    google.maps.event.addListener(marker1, 'position_changed', update);
    google.maps.event.addListener(marker2, 'position_changed', update);

    poly = new google.maps.Polyline({
        strokeColor: '#FF0000',
        strokeOpacity: 1.0,
        strokeWeight: 3,
        map: map,
    });

    geodesicPoly = new google.maps.Polyline({
        strokeColor: '#CC0099',
        strokeOpacity: 1.0,
        strokeWeight: 3,
        geodesic: true,
        map: map
    });

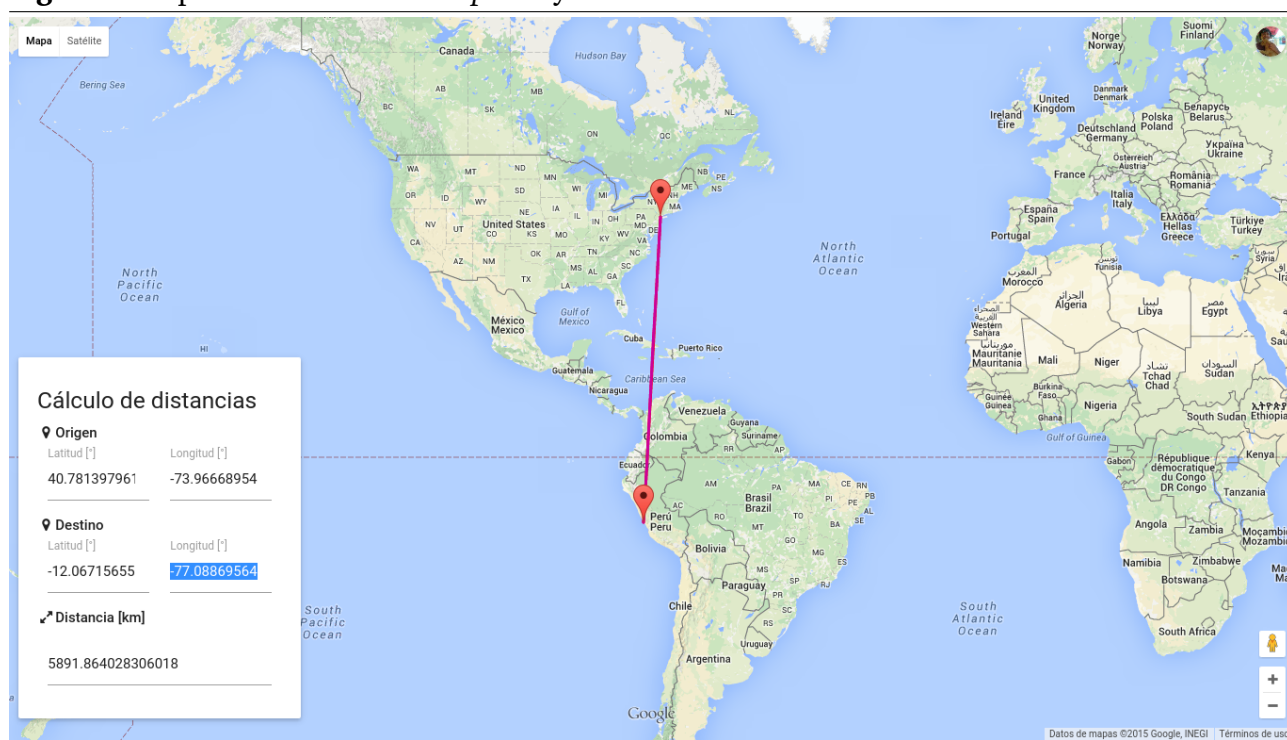
    update();
}
```

---

Continúa en la página siguiente.

```
function update() {
    var path = [marker1.getPosition(), marker2.getPosition()];
    poly.setPath(path);
    geodesicPoly.setPath(path);
    var distance = google.maps.geometry.spherical.computeDistanceBetween(path[0],
    ↵ ↵
    path[1]);
    document.getElementById('distance').value = (distance/1000).toString();
    document.getElementById('latitude1').value = path[0].lat();
    document.getElementById('longitude1').value = path[0].lng();
    document.getElementById('latitude2').value = path[1].lat();
    document.getElementById('longitude2').value = path[1].lng();
}
```

Figura 1 Implementación del *script* 12 y vista de resultados en web.



Para las pruebas de este programa se tomaron las coordenadas de Central Park, Nye York, Estados Unidos y El Parque de las leyendas, Lima, Perú.

## Problema 5

El día juliano es el número de orden que le corresponde a una fecha dada; por ejemplo, el 01 de enero sería el día juliano 1 y el 31 de diciembre sería el día juliano 365. Hacer un programa para convertir de día juliano a fecha. ¿A que fecha corresponde el día juliano 220? Variar el programa para tener en cuenta los años bisiestos: múltiplos de 4, excepto los que terminen en 00, como el año 2000.

### Solución

---

#### Script 13 Cálculo de la fecha juliana.

---

```
i = 1;
J = 0;
D = 0;
M = 0;
B = 0;
Lim = 355;
DpM = [31 28 31 30 31 30 31 31 30 31 30 31];

fprintf('Calculo del dia Juliano y Fecha\n');
fprintf('1. Conversion de Fecha a Dia Juliano\n');
fprintf('2. Conversion de Dia Juliano a Fecha\n');
Opt = input('Elija una opcion: ');

if(Opt ~= 1 && Opt ~= 2)
    fprintf('Error: Opcion no disponible');
else
    fprintf('\n');
    fprintf('Calculo del dia Juliano y Fecha\n');
    Y = input('ingrese ano: ');

    if rem(Y,4)==0 && (rem(Y,100)~=0 || rem(Y,400)==0)
        DpM(2) = 29;
        B = 1;
        Lim = 356;
    end

    if(Opt == 2)
        while (J > Lim || 1 > J)
            J = input('introduzca la fecha juliana: ');
            D = J;
        end

        while D > 0
            D = D - DpM(i);
            i = i + 1;
        end

        fprintf('\n');
        fprintf('La Fecha corresponde al dia juliano %d del ano %d es el
↪ %d/%d/%d\n', J, Y, D + DpM(i-1), i-1, Y);
```

Continúa en la página siguiente.

Continuación del *script* 13.

```
else
    while(M > 12 || 1 > M)
        M = input('ingrese mes: ');
    end

    while(D > DpM(M) || 1 > D)
        D = input('ingrese dia: ');
    end

    while i < M
        J = J + DpM(i);
        i = i + 1;
    end
    J = J + D;

    fprintf('\n');
    fprintf('El dia juliano correspondiente a la fecha %d/%d/%d es %d\n',D,M,Y,J);
end
end
```

---

#### Script 14 Ejemplo de ejecución del programa mostrado en el *script* 13

---

```
>> problema05
Calculo del dia Juliano y Fecha
1. Conversion de Fecha a Dia Juliano
2. Conversion de Dia Juliano a Fecha
Elija una opcion: 1

Calculo del dia Juliano y Fecha
Ingrese ano: 1900
Ingrese mes: 3
Ingrese dia: 1

El dia juliano correspondiente a la fecha 1/3/1900 es 60
>> problema05
Calculo del dia Juliano y Fecha
1. Conversion de Fecha a Dia Juliano
2. Conversion de Dia Juliano a Fecha
Elija una opcion: 2

Calculo del dia Juliano y Fecha
Ingrese ano: 2000
Introduzca la fecha juliana: 60

La Fecha corresponde al dia juliano 60 del ano 2000 es el 29/2/2000
```

La solución se planteó, considerando el día juliano como el número ordinal de la fecha correspondiente dentro de un año dado.

## Problema 6

Se tiene un cuadrado de lado  $L$  y una circunferencia inscrita en él. Supongamos que lanzamos pequeños dardos a gran distancia. Muchos caerán dentro y otros caerán fuera de la circunferencia. Sea  $n$  el número de dardos que caen dentro del círculo y  $N$  el número de dardos que caen dentro del cuadrado. La razón de estas dos cantidades será proporcional a la razón de las áreas del cuadrado y de la circunferencia. Hallar una aproximación de  $\pi$  en función de  $n$  y  $N$ . Hacer un programa para hallar el valor de  $\pi$  para un valor de  $N$  introducido por el usuario.

## Solución

---

### Script 15 Cálculo de aproximado de $\pi$

---

```
fprintf('Metodo de montecarlo para calcular pi\n');
N = input ('Ingrese el numero de dardos : ');

n = 0;
x = 0;
y = 0;
count = 1;

rng(0, 'twister');

while count < N;
    x = rand;
    y = rand;

    if sqrt (x^2 + y^2) <= 1
        n = n + 1;
    end

    count = count + 1;
end

fprintf('Dardos dentro del circulo      : %d\n', n);
fprintf('Valor aproximado de Pi        : %.5f\n', 4*n/N);
```

---

---

### Script 16 Ejemplo de ejecución del programa mostrado en el [script 15](#)

---

```
>> problema06
Metodo de montecarlo para calcular pi
Ingrese el numero de dardos : 1000000
Dardos dentro del circulo      : 784987
Valor aproximado de Pi        : 3.13995
>> problema06
Metodo de montecarlo para calcular pi
Ingrese el numero de dardos : 500000
Dardos dentro del circulo      : 392566
Valor aproximado de Pi        : 3.14053
>> problema06
Metodo de montecarlo para calcular pi
Ingrese el numero de dardos : 9000000
Dardos dentro del cuadrado     : 9000000
Valor aproximado de Pi        : 3.14172
```

---

## Problema 7

Hacer una gráfica en 3 dimensiones de la curva gaussiana:

$$z = A \cdot e^{-(x^2+y^2)}$$

donde  $A = 10$  es la amplitud de la curva. Utilice una grilla para el dominio:  $-10 < x < 10 \wedge -10 < y < 10$

1. Considere que la dimensión de la grilla es unitaria.
2. Considere que la dimensión de la grilla es 0.2.
3. Modifique el programa para visualizar curvas de nivel.

## Solución

---

**Script 17** Programa para graficar la función  $z = A \cdot e^{-(x^2+y^2)}$

---

```
fprintf('Grafica de Funciones\n');
fprintf('Funcion Gausiana z(x,y) = Aexp(-(x^2 + y^2)) en [-10,10]x[-10,10]\n');
A = input('Ingresa la amplitud : ');
N = input('Ingresa el numero de puntos : ');
fprintf(' Amplitud de la malla : %.5f\n', 20/(N-1));

X = linspace(-10,10,N);
Y = linspace(-10,10,N);
[X,Y] = meshgrid(X,Y);
Z = A*exp(-(X.^2 + Y.^2));
str1 = sprintf('Superficie z(x,y) = %.3f*exp(-(x^2 + y^2))',A);

surf(X,Y,Z,'EdgeColor','none');
title (str1);

C = input('Desea graficar curvas de nivel (Y/N)? ', 's');

if(C == 'Y' || C == 'y')
    close all
    str2 = sprintf('%d Curvas de Nivel ', floor(N/3));
    subplot(1,2,1);
    surf(X,Y,Z,'EdgeColor','none');
    title (str1);
    subplot(1,2,2);
    contourf(X,Y,Z,floor(N/3));
    title (str2);
end
```

---

---

**Script 18** Ejemplo de ejecución del programa mostrado en el [script 17](#)

---

```
>>problema07.m
Grafica de Funciones
Funcion Gausiana  $z(x,y) = A\exp(-(x^2 + y^2))$  en  $[-10,10] \times [-10,10]$ 
Ingrese la amplitud      : 5
Ingrese el numero de puntos : 21
Amplitud de la malla      : 1.00000
Desea graficar curvas de nivel (Y/N)? y
>>problema07.m
Grafica de Funciones
Funcion Gausiana  $z(x,y) = A\exp(-(x^2 + y^2))$  en  $[-10,10] \times [-10,10]$ 
Ingrese la amplitud      : 5
Ingrese el numero de puntos : 101
Amplitud de la malla      : 0.20000
Desea graficar curvas de nivel (Y/N)? y
```

---

---

**Figura 2** Resultados de la ejecución del [script 17](#).

---

