Práctica-2-MATLAB-Intro

Martínez Ostoa Néstor Iván
Visualización de la Información - 0605
IIMAS - UNAM

7 de marzo del 2021

Primeros problemas

1. Cree las gráficas de las siguientes funciones, desde $x = 0$ hasta $x = 10$

   a) $y = e^x$
   b) $y = \sin x$
   c) $y = ax^2 + bx + c$; con $a = 5, b = 2, c = 4$
   d) $y = \sqrt{x}$

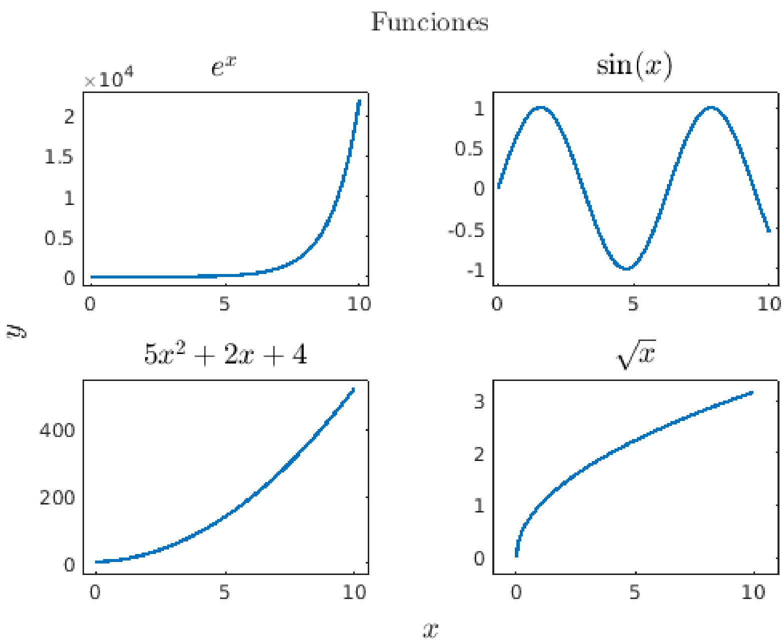


Figura 1: funciones de arriba hacía abajo y de derecha a izquierda: **a)**, **b)**, **c)**, **d)**

2. Grafique las siguientes funciones en la misma gráfica para valores de $x \in [-\pi, \pi]$. [1]

   a) $y = \sin x$
   b) $y = \sin 2x$
   c) $y = \sin 3x$

[1] Se deben ver gráficas suaves

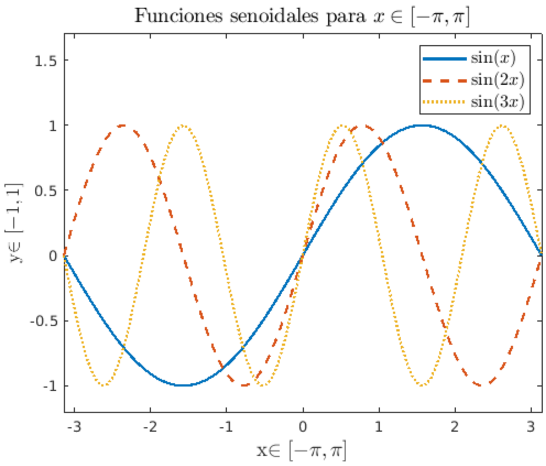


Figura 2: Funciones senoidales

3. La distancia que recorre un proyectil cuando se dispara un ángulo θ es una función del tiempo y se puede dividir en distancia horizontal (h) y vertical (y) de acuerdo a las ecuaciones

$$h(t) = h_0 + tV_0 \cos\theta$$

y

$$y(t) = y_0 + tV_0 \sin\theta - \frac{1}{2}gt^2$$

   a) Suponga que el proyectil descrito se dispara con una velocidad inicial de $100\frac{m}{s}$ y un ángulo de lanzamiento de 60°. Encuentre la distancia recorrida tanto horizontal como verticalmente (en las direcciones h y y) para tiempos desde 0 hasta $20s$.

      1) Grafique distancia horizontal contra tiempo.

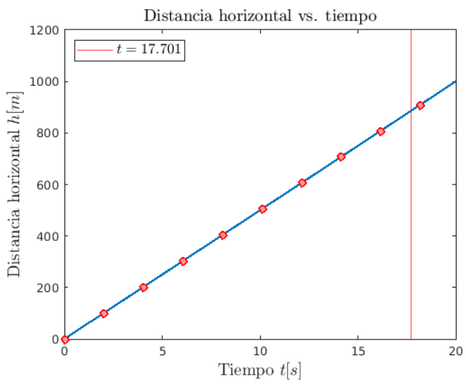


Figura 3: Distancia horizontal vs tiempo

      2) En una nueva ventana de figura, grafique distancia vertical contra tiempo.

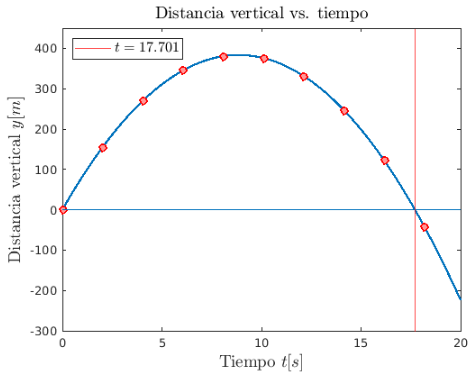


Figura 4: Distancia vertical vs tiempo

b) En una nueva ventana de figura, grafique distancia horizontal sobre el eje x y distancia vertical sobre el eje y.

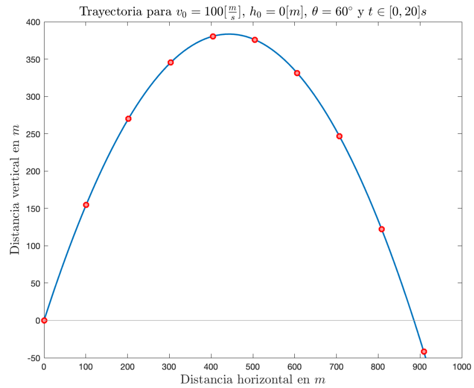


Figura 5: Distancia horizontal vs. Distancia vertical

c) Calcule tres nuevos vectores para cada una de las distancias vertical (y_1, y_2, y_3) y horizontal (h_1, h_2, h_3) recorridas, y suponga ángulos de lanzamiento de 80° , 45° y 30°

1) En una nueva ventana de figura, grafique distancia horizontal en el eje x y distancia vertical en el eje y, para los tres casos. (Tendrá tres líneas)

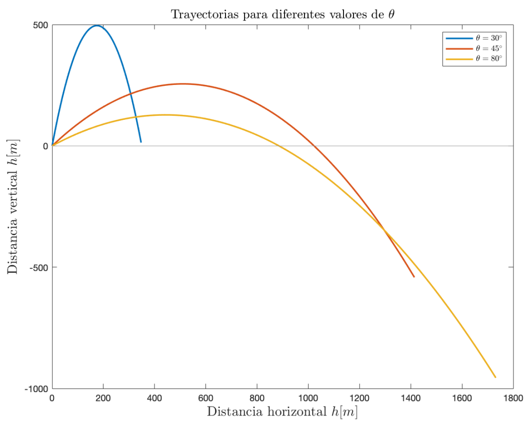


Figura 6: Trayectorias para tres valores del ángulo θ

2) Haga una línea sólida, una rayada y una punteada (aumente el ancho de línea en caso de ser necesario). Agregue una leyenda para identificar cuál línea es cuál.

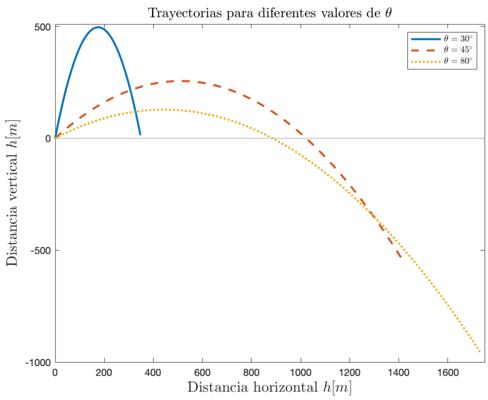


Figura 7: Trayectorias para tres valores del ángulo θ

3) En una nueva figura grafique h_i vs y_i $(i = 1, 2, 3)$. ¿cuál de los tres lanzamientos alcanza una mayor distancia horizontal si se ejecuta sobre un plano?¿si se lanza desde el borde de un abismo de profundidad desconocida?

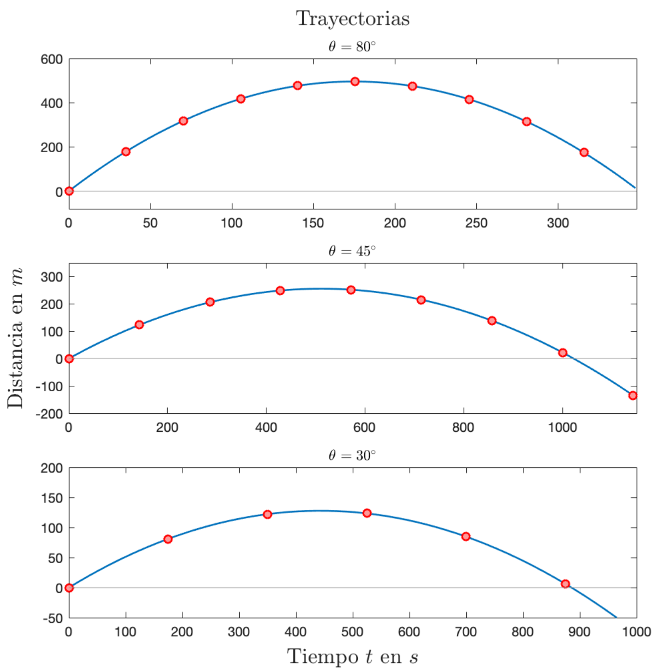


Figura 8: Comparación de distancia horizontal vs. vertical para diversos valores de θ

Lo que podemos observar en la figura 8 es que, de los tres lanzamientos, con un $\theta = 45°$ obtenemos la mayor distancia horizontal (aproximadamente $1000m$). Si lazáramos este experimento desde el borde de un abismo, el ángulo θ que nos da la mayor distancia horizontal sería $\theta = 30°$.

Conjuntos de Mandelbrot y Julia

Ver Anexo 1.4 para el código que genera la siguiente figura 9.

- Genere matrices utilizando `meshgrid` para x y y de tamaño 500 de tal forma que:

$$-1{,}5 \leq x \leq 1$$
$$-1{,}5 \leq y \leq 1$$

- Cree la matriz compleja $C = X + iY$
- Inicialice la matriz compleja Z de tamaño 500×500
- Establezca un número de iteraciones mayor a 50 para ejecutar las siguientes ecuaciones:

$$\begin{aligned} z(0) &= x + yi \\ z(1) &= z(0)^2 + z(0) \\ z(2) &= z(1)^2 + z(0) \\ &\vdots \\ z(n) &= z(n1)^2 + z(0) \end{aligned}$$

- En cada iteración almacene en una matriz (M) aquellos puntos en el plano complejo con $\text{abs}(z) \geq \sqrt{5}$
- Despliegue la matriz M con `image(M)`

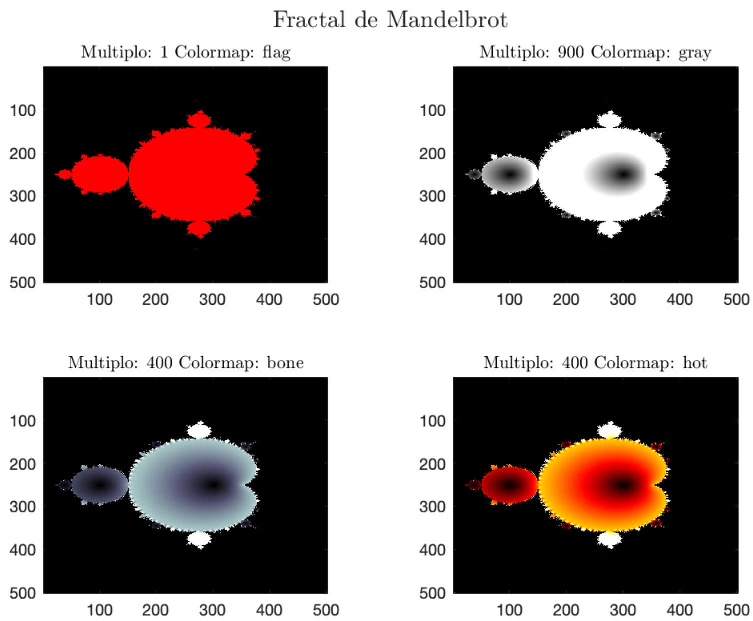


Figura 9: Conjuntos de Mandelbrot

Parámetros de visualización

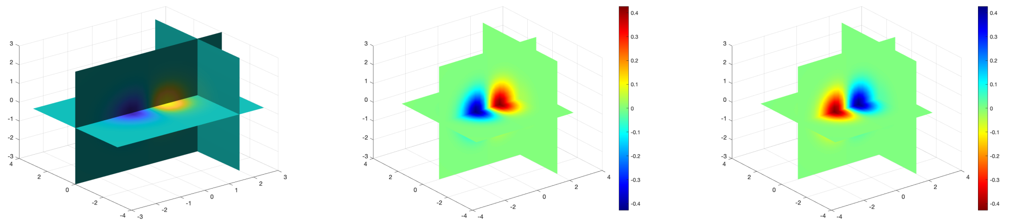
Para la siguiente sección de código modifica los parámetros de visualización para que las figuras se vean bien.

Utilizando la siguiente ecuación:

$$v = x \cdot e^{-x^2 - y^2 - z^2}$$

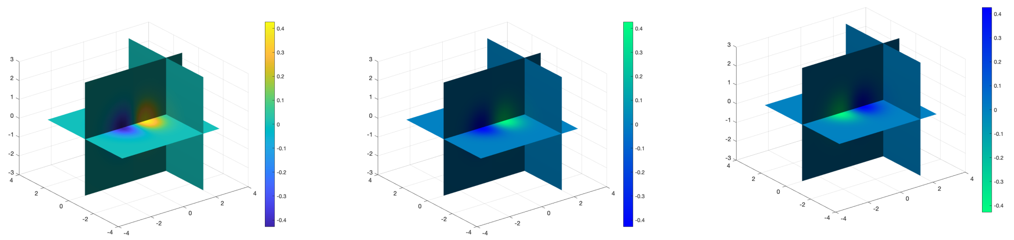
obtenemos las siguientes figuras:

Después de aplicar métodos para controlar la luz y la sombra en las tres figuras y cambiar el mapa de colores, obtenemos las siguientes figuras:



(a) **colormap parula** (b) **colormap jet** (c) **colormap jet invertido**

Figura 10: Figuras originales



(a) **colormap parula** (b) **colormap winter** (c) **colormap winter invertido**

Figura 11: Figuras modificadas con luz y sombra

1. Anexo

En este anexo se muestran los códigos para implementar todas las figuras de este reporte.

1.1. Ejercicio 1: Funciones generales

```
a = 5;
b = 2;
c = 4;
x = (0:0.1:10);
y1 = exp(x);
y2 = sin(x);
y3 = a.*(x.*x) + b.*x + c;
y4 = sqrt(x);

t = tiledlayout(2,2);
t.TileSpacing = 'normal';

title_lab = title(t, 'Funciones');
set(title_lab, 'Interpreter', 'latex');
xlab = xlabel(t, '$x$');
set(xlab, 'Interpreter', 'latex');
set(xlab, 'FontSize', 14);
ylab = ylabel(t, '$y$');
set(ylab, 'Interpreter', 'latex');
set(ylab, 'FontSize', 14);

nexttile;
plot(x,y1, 'linewidth', 1.7); % 'MarkerIndices',1:10:length(y));
```

```
title('$e^x$', 'Interpreter', 'latex', 'FontSize', 14);
axis([-0.3,10.3,-1000,23000]);

nexttile
plot(x,y2, 'linewidth', 1.7);
title('$\sin(x)$', 'Interpreter', 'latex', 'FontSize', 14);
axis([-0.2,10.3,-1.2,1.2]);

nexttile;
plot(x,y3, 'linewidth', 1.7);
title('$5x^2 + 2x + 4$', 'Interpreter', 'latex', 'FontSize', 14);
axis([-0.5,10.5,-30,550]);

nexttile;
plot(x,y4,'linewidth',1.7);
title('$\sqrt{x}$', 'Interpreter', 'latex', 'FontSize', 14);
axis([-1,11,-0.3,3.4]);
```

1.2. Ejercicio 2: Funciones sinusoidales

```
x = linspace(-pi, pi);
y1 = sin(x);
y2 = sin(2.*x);
y3 = sin(3.*x);

figure
plot(x,y1,x,y2,'--',x,y3,':', 'linewidth', 1.7);
tl = title('Funciones senoidales para $x\in[-\pi, \pi]$', 'Interpreter', 'latex');
set(tl, 'FontSize', 14);
xlab = xlabel('x$\in [-\pi, \pi]$', 'Interpreter', 'latex');
set(xlab, 'FontSize', 14);
ylab = ylabel('y$\in [-1,1]$', 'Interpreter', 'latex');
set(ylab, 'FontSize', 13);
leg = legend('$\sin(x)$', '$\sin(2x)$', '$\sin(3x)$');
set(leg, 'Interpreter', 'latex');
set(leg, 'FontSize', 12);
axis([-pi,pi,-1.2,1.7]);
```

1.3. Ejercicio 3: Trayectoria de un proyectil

1.3.1. Ejercicio 3.a

```
v0 = 100;
theta = 60;
time = linspace(0,20);
h0 = 0;
h = h0 + (v0*cos(theta*pi/180)).*time;
y0 = 0;
g = 9.78;
y = y0 + (v0*sin(theta*pi/180)).*time -(0.5*g).*(time.^2);

syms t;
solt = solve(-0.5*g*t^2 + v0*sin(theta*pi/180)*t +y0 == 0, t);
time_zero = double(vpa(solt(2)));

figure;
```

```
p(1) = plot(time, h, '-o', 'linewidth', 1.7);
p(1).MarkerIndices = 1:10:length(time);
p(1).MarkerEdgeColor = 'red';
p(1).MarkerFaceColor = [1 .6 .6];
p(1).MarkerSize = 6;
p(2) = xline(time_zero);
p(2).Color = 'r';
title('Distancia horizontal vs. tiempo', 'Interpreter', 'latex', 'FontSize', 14);
xlabel('Tiempo $t [s]$', 'Interpreter', 'latex', 'FontSize', 14);
ylabel('Distancia horizontal $h [m]$', 'Interpreter', 'latex', 'FontSize', 14);
l = legend(p(2), '$t=17.701$', 'Location', 'northwest');
set(l, 'Interpreter', 'latex');
set(l, 'FontSize', 12);

figure;
p = plot(time, y, '-o', 'linewidth', 1.7);
p.MarkerIndices = 1:10:length(time);
p.MarkerEdgeColor = 'red';
p.MarkerFaceColor = [1 .6 .6];
p.MarkerSize = 6;
title('Distancia vertical vs. tiempo', 'Interpreter', 'latex', 'FontSize', 14);
xlabel('Tiempo $t [s]$', 'Interpreter', 'latex', 'FontSize', 14);
ylabel('Distancia vertical $y [m]$', 'Interpreter', 'latex', 'FontSize', 14);
vline = xline(time_zero);
vline.Color = 'r';
hline = refline(0);
l = legend(vline, '$t=17.701$', 'Location', 'northwest');
set(l, 'Interpreter', 'latex');
set(l, 'FontSize', 12);
axis([0,20,-300,450]);
```

1.3.2. Ejercicio 3.b

```
figure;
p = plot(h, y, '-o', 'linewidth', 1.7);
p.MarkerIndices = 1:10:length(h);
p.MarkerEdgeColor = 'red';
p.MarkerFaceColor = [1 .6 .6];
p.MarkerSize = 6;
hline = refline(0);
hline.Color = [.6 .6 .6];
axis([0 1000, -50, 400]);
xlabel('Distancia horizontal en $m$', 'Interpreter', 'latex', 'FontSize', 15);
ylabel('Distancia vertical en $m$', 'Interpreter', 'latex', 'FontSize', 15);
str = 'Trayectoria para $v_0 = 100 [\frac{m}{s}]$, $h_0 = 0[m]$,
    $\theta = 60^\circ$ y $t\in[0,20]s$';
title(str, 'Interpreter', 'latex', 'FontSize', 15);
```

1.3.3. Ejercicio 3.c

```
v0 = 100;
h0 = 0;
y0 = 0;
g = 9.78;
time = linspace(0,20);
theta = [80*pi/180, 45*pi/180, 30*pi/180];
```

```
h1 = h0 + (v0*cos(theta(1))).*time;
h2 = h0 + (v0*cos(theta(2))).*time;
h3 = h0 + (v0*cos(theta(3))).*time;
y1 = y0 + (v0*sin(theta(1))).*time -(0.5*g).*(time.^2);
y2 = y0 + (v0*sin(theta(2))).*time -(0.5*g).*(time.^2);
y3 = y0 + (v0*sin(theta(3))).*time -(0.5*g).*(time.^2);


% 1)
figure
plot(h1, y1, h2, y2, h3, y3, 'linewidth', 1.7);
hline = refline(0);
hline.Color = [.6 .6 .6];
title('Trayectorias␣para␣diferentes␣valores␣de␣$\theta$', 'Interpreter', 'latex',
    'FontSize', 14);
ylabel('Distancia␣vertical␣$h␣[m]$', 'Interpreter', 'latex', 'FontSize', 15);
xlabel('Distancia␣horizontal␣$h␣[m]$', 'Interpreter', 'latex', 'FontSize', 15);
l = legend('$\theta␣=␣30^\circ$', '$\theta␣=␣45^\circ$', '$\theta␣=␣80^\circ$',
    'Location', 'northeast');
set(l, 'Interpreter', 'latex');
% 2)
figure
plot(h1, y1, h2, y2, '--', h3, y3, ':', 'linewidth', 2);
hline = refline(0);
hline.Color = [.6 .6 .6];
axis([0, 1750, -1000, 510]);
title('Trayectorias␣para␣diferentes␣valores␣de␣$\theta$', 'Interpreter', 'latex',
    'FontSize', 14);
ylabel('Distancia␣vertical␣$h␣[m]$', 'Interpreter', 'latex', 'FontSize', 15);
xlabel('Distancia␣horizontal␣$h␣[m]$', 'Interpreter', 'latex', 'FontSize', 15);
l = legend('$\theta␣=␣30^\circ$', '$\theta␣=␣45^\circ$', '$\theta␣=␣80^\circ$',
    'Location', 'northeast');
set(l, 'Interpreter', 'latex');
% 3)
figure;
t = tiledlayout(3,1);
fontSize = 16;
lineWidth = 1.3;
title(t, 'Trayectorias', 'Interpreter', 'latex', 'FontSize', fontSize);
xlabel(t, 'Tiempo␣$t$␣en␣$s$', 'Interpreter', 'latex', 'FontSize', fontSize);
ylabel(t, 'Distancia␣en␣$m$', 'Interpreter', 'latex', 'FontSize', fontSize);

nexttile;
p = plot(h1, y1, '-o', 'linewidth', lineWidth);
p.MarkerIndices = 1:10:length(h1);
p.MarkerEdgeColor = 'red';
p.MarkerFaceColor = [1 .6 .6];
p.MarkerSize = 6;
axis([0 348 -80 600]);
hline = refline(0);
hline.Color = [.6 .6 .6];
title('$\theta=80^\circ$', 'Interpreter', 'latex');

nexttile;
p = plot(h2, y2, '-o', 'linewidth', lineWidth);
p.MarkerIndices = 1:10:length(h1);
p.MarkerEdgeColor = 'red';
```

```
p.MarkerFaceColor = [1 .6 .6];
p.MarkerSize = 6;
axis([0 1150 -200 350]);
hline = refline(0);
hline.Color = [.6 .6 .6];
title('$\theta=45^\circ$', 'Interpreter', 'latex');

nexttile;
p = plot(h3, y3, '-o', 'linewidth', lineWidth);
p.MarkerIndices = 1:10:length(h1);
p.MarkerEdgeColor = 'red';
p.MarkerFaceColor = [1 .6 .6];
p.MarkerSize = 6;
axis([0 1000 -50 200]);
hline = refline(0);
hline.Color = [.6 .6 .6];
title('$\theta=30^\circ$', 'Interpreter', 'latex');
```

1.4. Ejercicio 4: Conjuntos de Mandelbrot

```
stepsX = 2.5 / 500;
stepsY = 3 / 500;
[X, Y] = meshgrid(-1.5:stepsX:1,-1.5:stepsY:1.5);

Z0 = X + Y*1i;

Z = zeros(501, 501);

nIter = 100;


M = zeros(501, 501);
for k = 1:nIter
    Z = Z.^2 + Z0;
   M(abs(Z) > sqrt(5)) = k;
end

figure;
t = tiledlayout(2,2);
title(t, 'Fractal de Mandelbrot', 'Interpreter', 'latex', 'FontSize', 15);

nexttile;
image(M);
colormap(gca, flag);
title('Multiplo: $1$ Colormap: flag', 'Interpreter', 'latex');

nexttile;
image(abs(Z)*900);
colormap(gca, gray);
title('Multiplo: $900$ Colormap: gray', 'Interpreter', 'latex');

nexttile;
image(abs(Z)*400);
colormap(gca, bone);
title('Multiplo: $400$ Colormap: bone', 'Interpreter', 'latex');
```

```
nexttile;
image(abs(Z)*350);
colormap(gca, hot);
title('Multiplo: $400$ Colormap: hot', 'Interpreter', 'latex');
```

1.5. Ejercicio 5: Parámetros de visualización

```
clear
close all
clc

[x, y, z] = meshgrid(-3:0.25:3);
v = x.*exp(-x.^2 - y.^2 - z.^2);

slice (x, y, z, v, [], 0, []);

figure
clf(figure(1))
xslice = 2;
yslice = 0;
zslice = 0;
slice(x,y,z,v,xslice,yslice,zslice);
view(3);
axis([-4 4 -4 4 -3 3]);
grid on;
colorbar;
light;
lighting gouraud;
camlight('left');
shading interp;

figure
clf(figure(2))
slice(x,y,z,v,xslice,yslice,zslice);
view(3);
axis([-4 4 -4 4 -3 3]);
grid on;
colormap winter;
colorbar;
light;
lighting gouraud;
camlight('left');
shading interp;

figure
clf(figure(3))
slice(x,y,z,v,xslice,yslice,zslice);
view(3);
axis([-4 4 -4 4 -3 3]);
colormap(flipud(winter));
colorbar;
light;
lighting gouraud;
camlight('left');
shading interp;
```