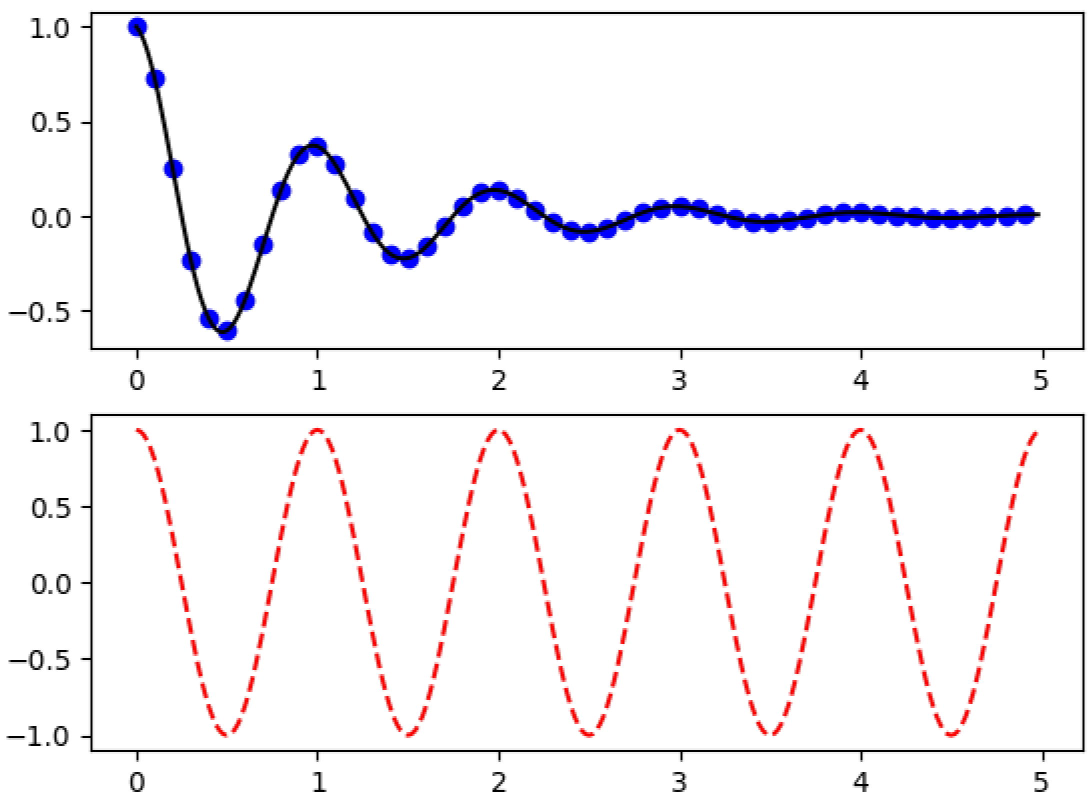
In [ ]:
```
# Judson Carenand
# ENS
# Physique
```

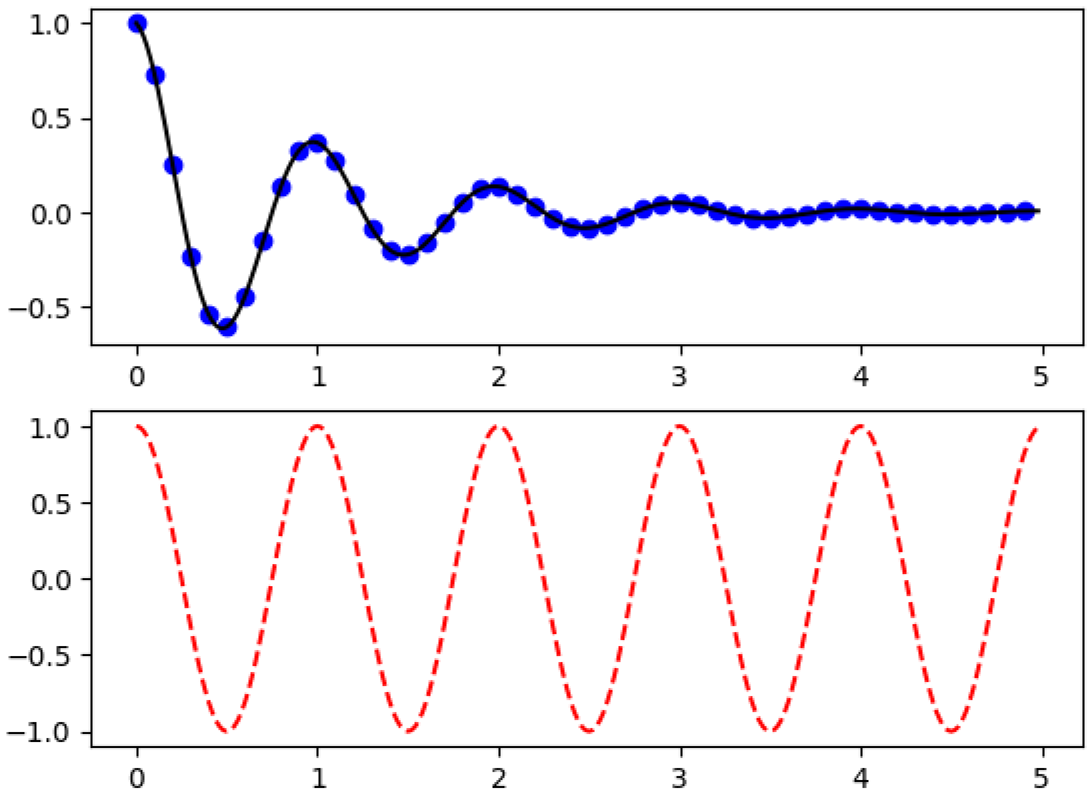
In [ ]:
```
# Judson Carenand

import numpy as np
import matplotlib.pyplot as plt
```

In [1]:
```
import numpy as np
import matplotlib.pyplot as plt

def f(t):
    return np.exp(-t) * np.cos(2 * np.pi * t)

t1 = np.arange(0.0, 5.0, 0.1)
t2 = np.arange(0.0, 5.0, 0.02)

plt.subplot(211)
plt.plot(t1, f(t1), "bo")
plt.plot(t2, f(t2), "k")

plt.subplot(212)
plt.plot(t2, np.cos(2 * np.pi * t2), "r--")

plt.show()
```



In [2]:
```
# Style Orienté d'utilisation subplot
```

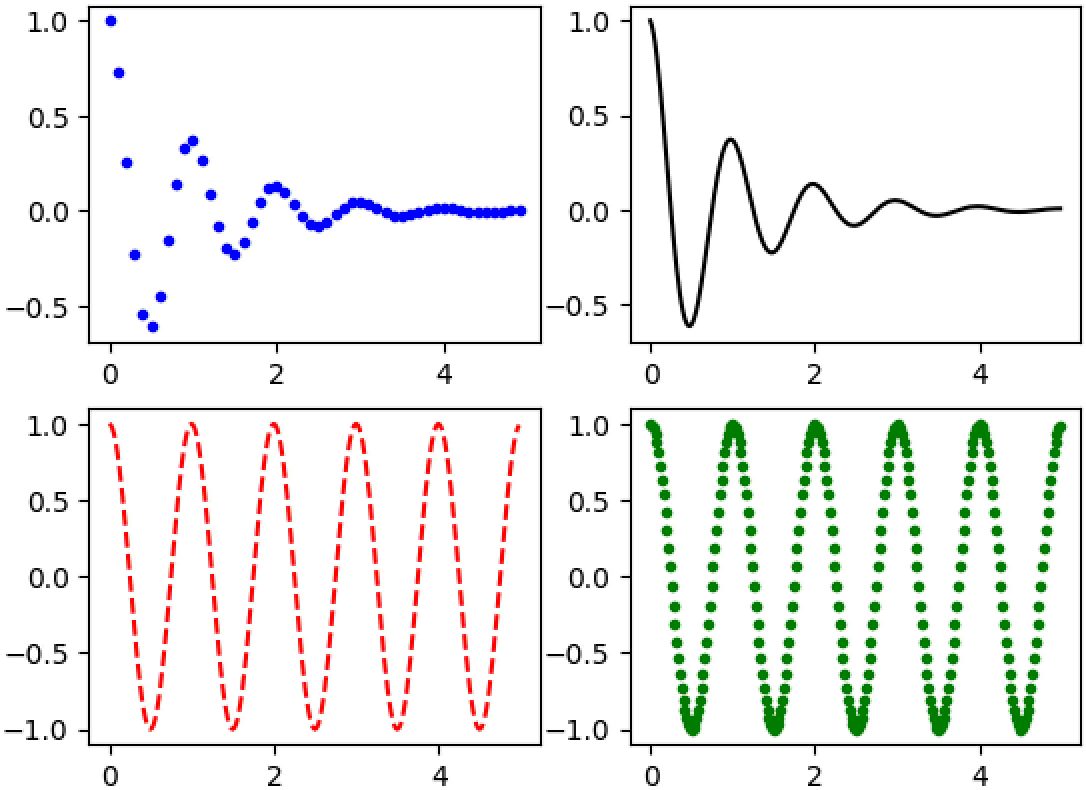
```
import numpy as np
import matplotlib.pyplot as plt

def f(t):
    return np.exp(-t) * np.cos(2 * np.pi * t)

t1 = np.arange(0.0, 5.0, 0.1)
t2 = np.arange(0.0, 5.0, 0.02)

fig, axs = plt.subplots(2, 1)

axs[0].plot(t1, f(t1), "bo")
axs[0].plot(t2, f(t2), "k")

axs[1].plot(t2, np.cos(2 * np.pi * t2), "r--")

plt.show()
```



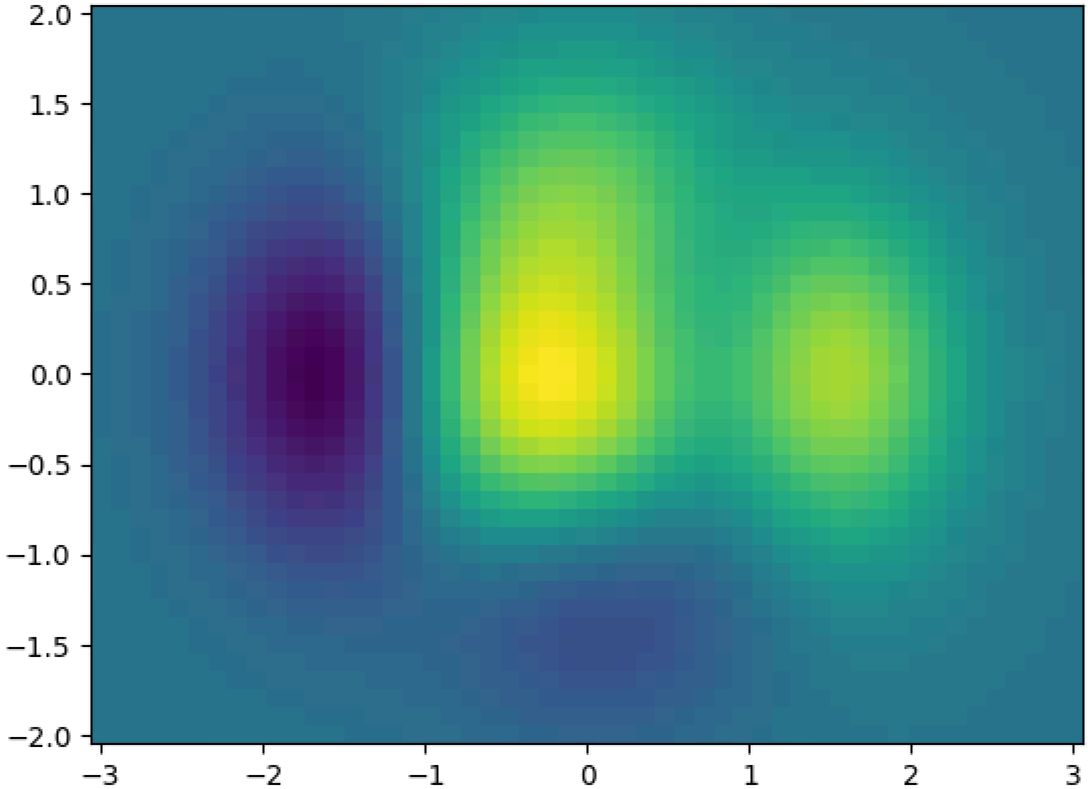
In [3]:

```
import numpy as np
import matplotlib.pyplot as plt

def f(t):
    return np.exp(-t) * np.cos(2 * np.pi * t)

t1 = np.arange(0.0, 5.0, 0.1)
t2 = np.arange(0.0, 5.0, 0.02)

fig, axs = plt.subplots(2, 2)

axs[0, 0].plot(t1, f(t1), "b.")
axs[0, 1].plot(t2, f(t2), "k")
```

```
axs[1, 0].plot(t2, np.cos(2 * np.pi * t2), "r--")
axs[1, 1].plot(t2, np.cos(2 * np.pi * t2), "g.")

plt.show()
```

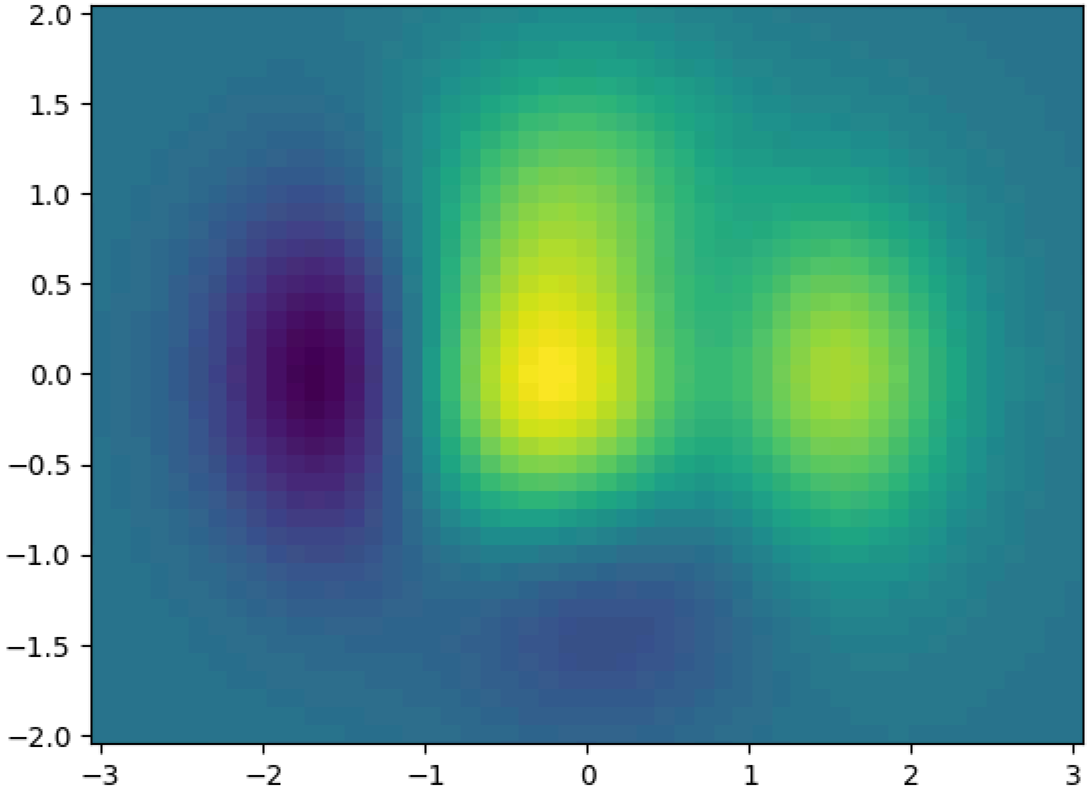


In [4]:
```
import numpy as np
import matplotlib.pyplot as plt

x = np.linspace(-3, 3, 51)
y = np.linspace(-2, 2, 41)

X, Y = np.meshgrid(x, y)

# calcul du tableau des valeurs de Z
Z = (1 - X/2 + X**5 + Y**3) * np.exp(-X**2 - Y**2)

plt.pcolormesh(X, Y, Z)
plt.show()
```

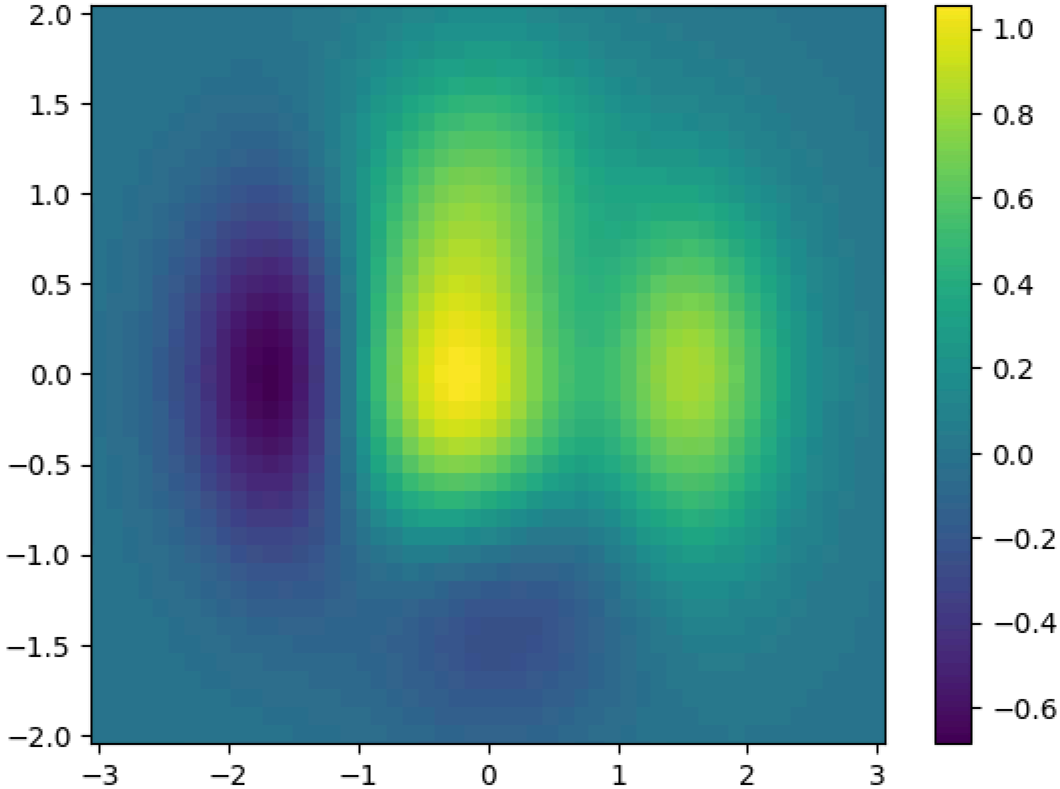


In [5]:

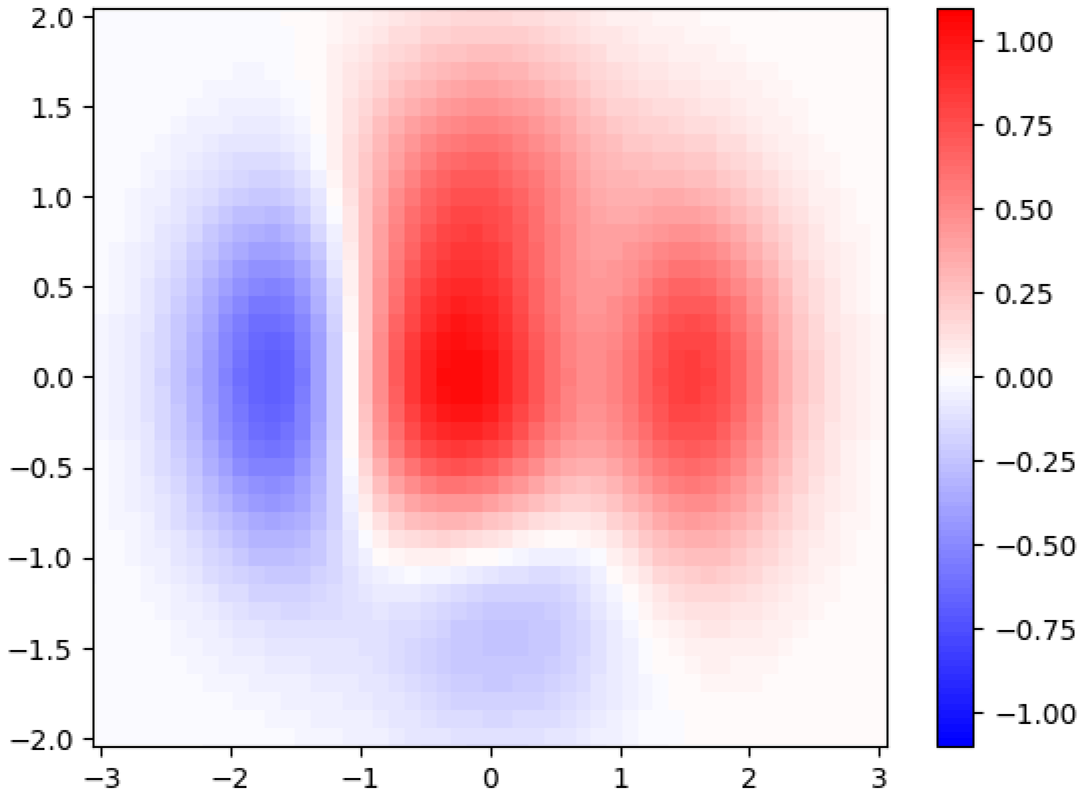
```
import numpy as np
import matplotlib.pyplot as plt

x = np.linspace(-3, 3, 51)
y = np.linspace(-2, 2, 41)

X, Y = np.meshgrid(x, y)

# calcul du tableau des valeurs de Z
Z = (1 - X/2 + X**5 + Y**3) * np.exp(-X**2 - Y**2)

fig, ax = plt.subplots()
ax.pcolormesh(X, Y, Z)

plt.show()
```



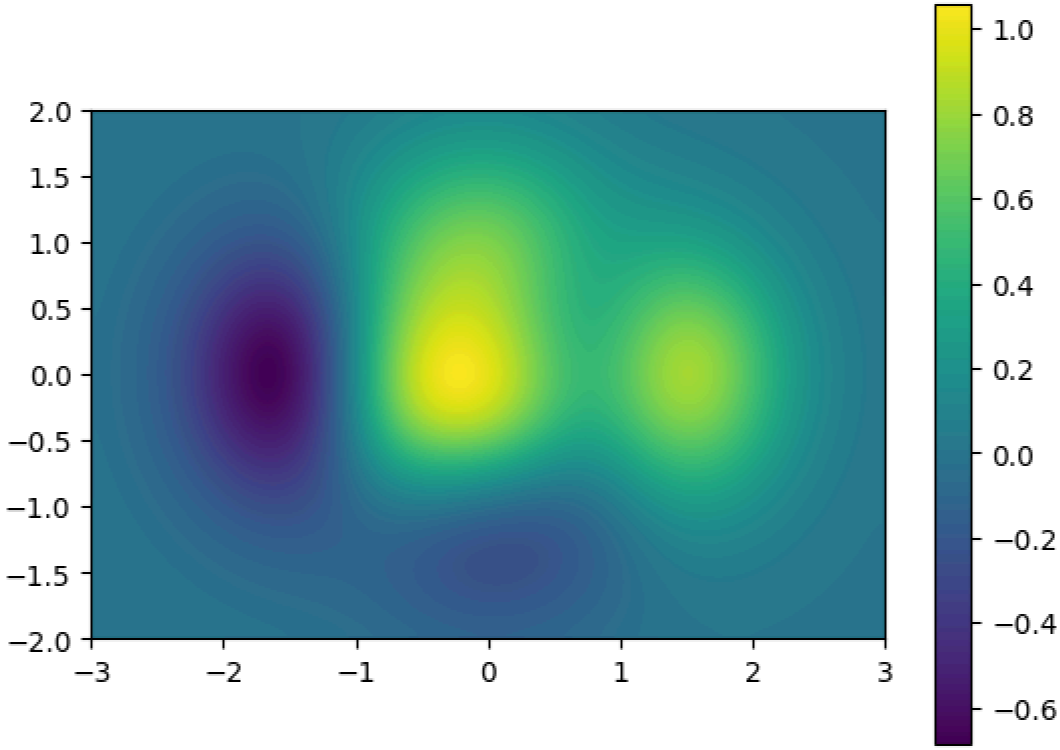
In [6]:
```
import numpy as np
import matplotlib.pyplot as plt

x = np.linspace(-3, 3, 51)
y = np.linspace(-2, 2, 41)

X, Y = np.meshgrid(x, y)

# calcul du tableau des valeurs de Z
Z = (1 - X/2 + X**5 + Y**3) * np.exp(-X**2 - Y**2)

plt.pcolormesh(X, Y, Z)
plt.colorbar()  # Affiche la barre de couleurs
plt.show()
```



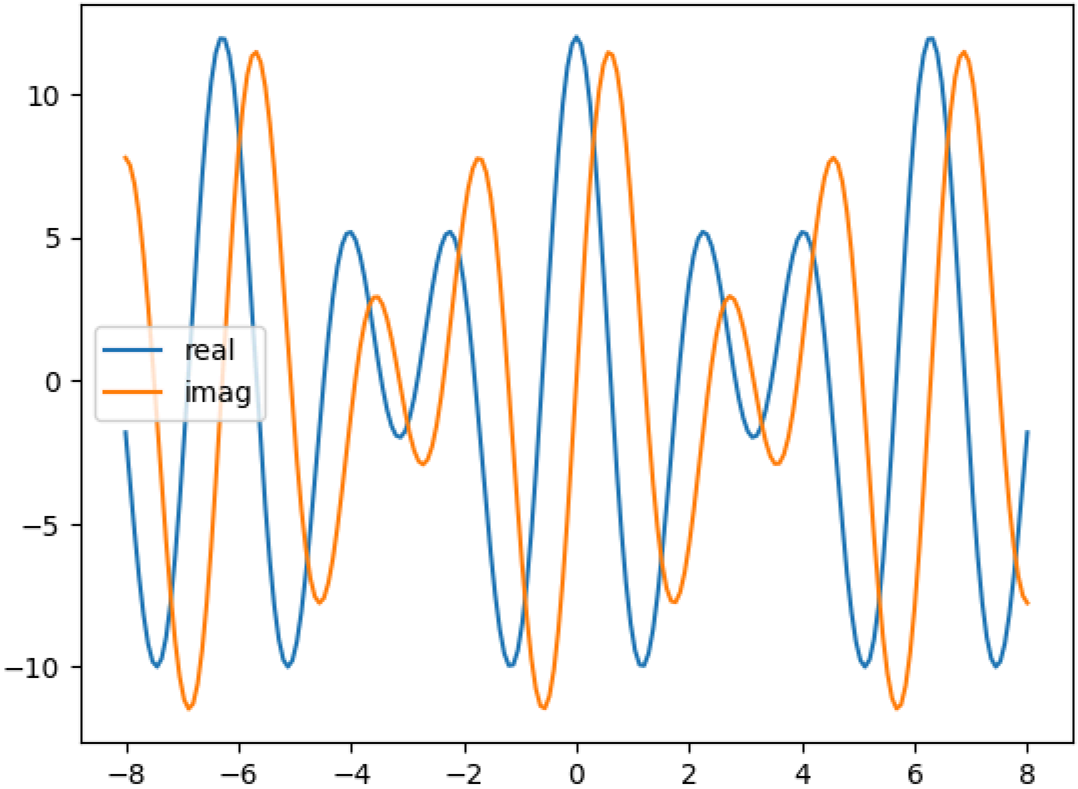
In [7]:
```python
import numpy as np
import matplotlib.pyplot as plt

x = np.linspace(-3, 3, 51)
y = np.linspace(-2, 2, 41)

X, Y = np.meshgrid(x, y)

# calcul du tableau des valeurs de Z
Z = (1 - X/2 + X**5 + Y**3) * np.exp(-X**2 - Y**2)

fig, ax = plt.subplots()

# Création de la carte de couleurs avec limites et palette personnalisée
pc = ax.pcolormesh(X, Y, Z, vmin=-1.1, vmax=1.1, cmap=plt.cm.bwr)

# Ajout de la barre de couleurs
fig.colorbar(pc)

plt.show()
```



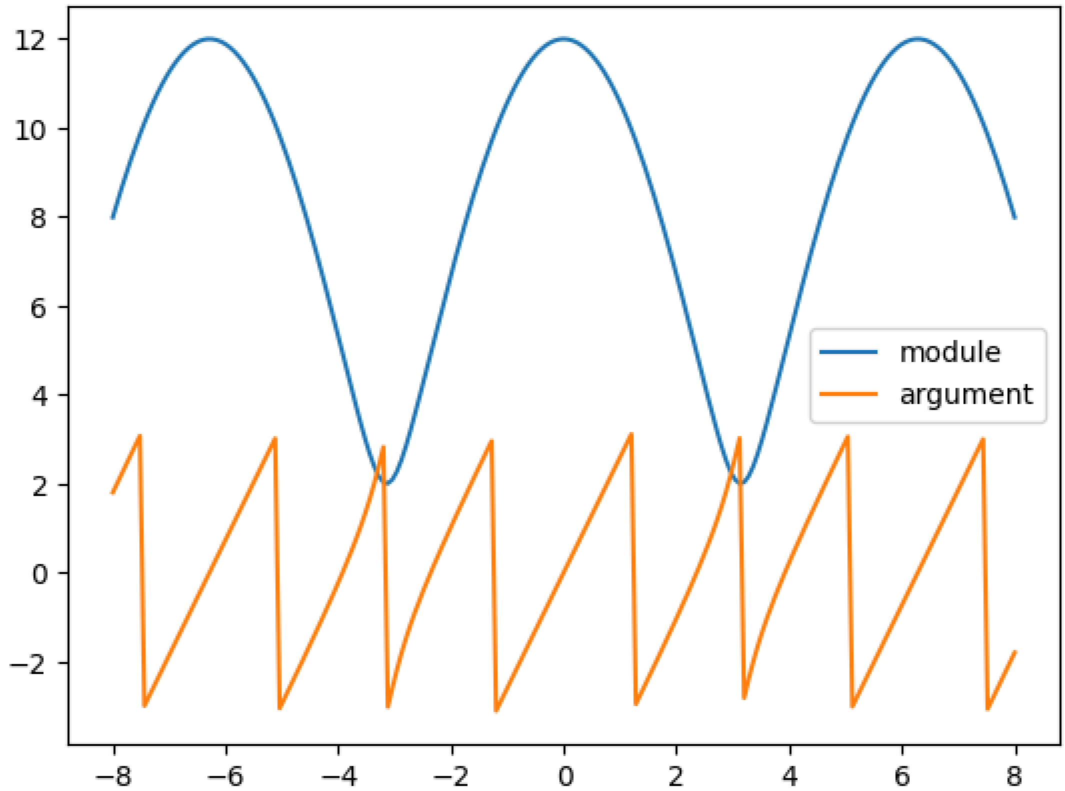
In [8]:
```
import numpy as np
import matplotlib.pyplot as plt

xmin = -3
xmax = 3
nbx = 51
ymin = -2
ymax = 2
nby = 41

x = np.linspace(xmin, xmax, nbx)
y = np.linspace(ymin, ymax, nby)

X, Y = np.meshgrid(x, y)

# calcul du tableau des valeurs de Z
Z = (1 - X/2 + X**5 + Y**3) * np.exp(-X**2 - Y**2)

plt.imshow(Z, interpolation="bicubic", origin="lower",
           extent=[xmin, xmax, ymin, ymax])
plt.colorbar()
plt.show()
```



In [9]:
```
import numpy as np
import matplotlib.pyplot as plt

x = np.linspace(-8, 8, 201)
z = 5 * np.exp(2j * x) + 7 * np.exp(3j * x)

plt.plot(x, np.real(z), label="real")
plt.plot(x, np.imag(z), label="imag")

plt.legend()
plt.show()
```

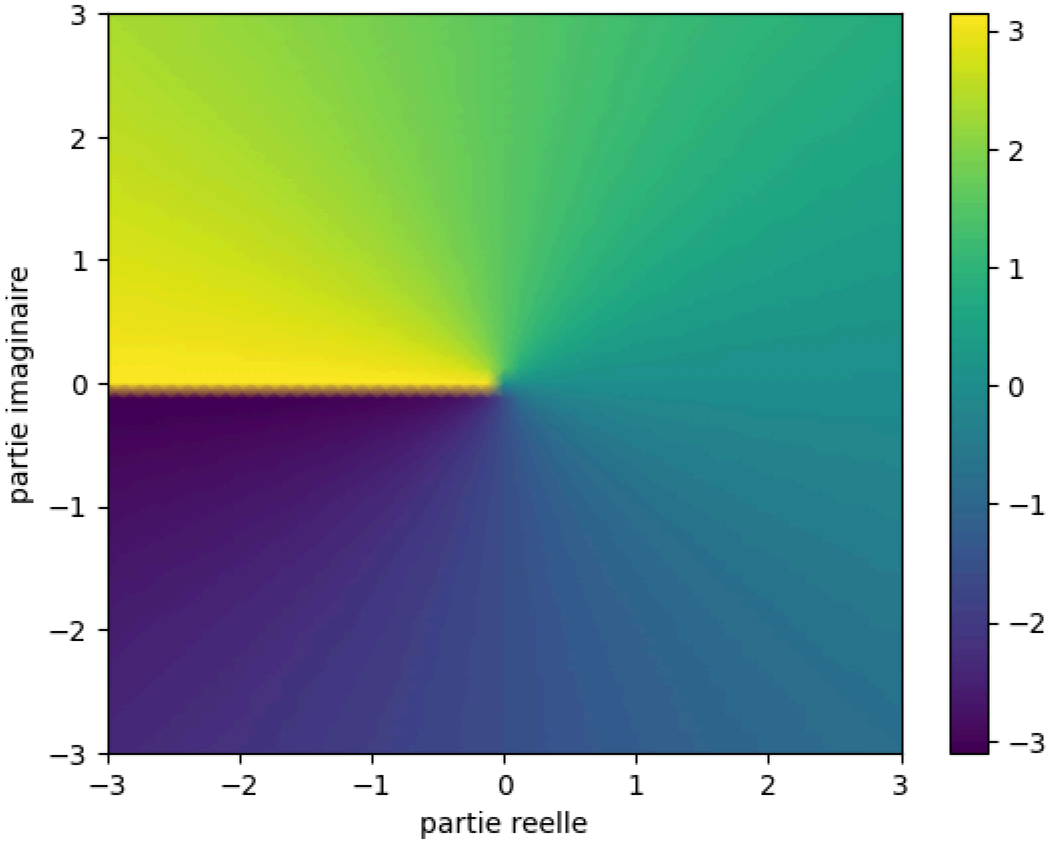


In [11]:
```
import numpy as np
import matplotlib.pyplot as plt

x = np.linspace(-8, 8, 201)
z = 5 * np.exp(2j * x) + 7 * np.exp(3j * x)

plt.plot(x, np.abs(z), label="module")
plt.plot(x, np.angle(z), label="argument")

plt.legend()
plt.show()
```

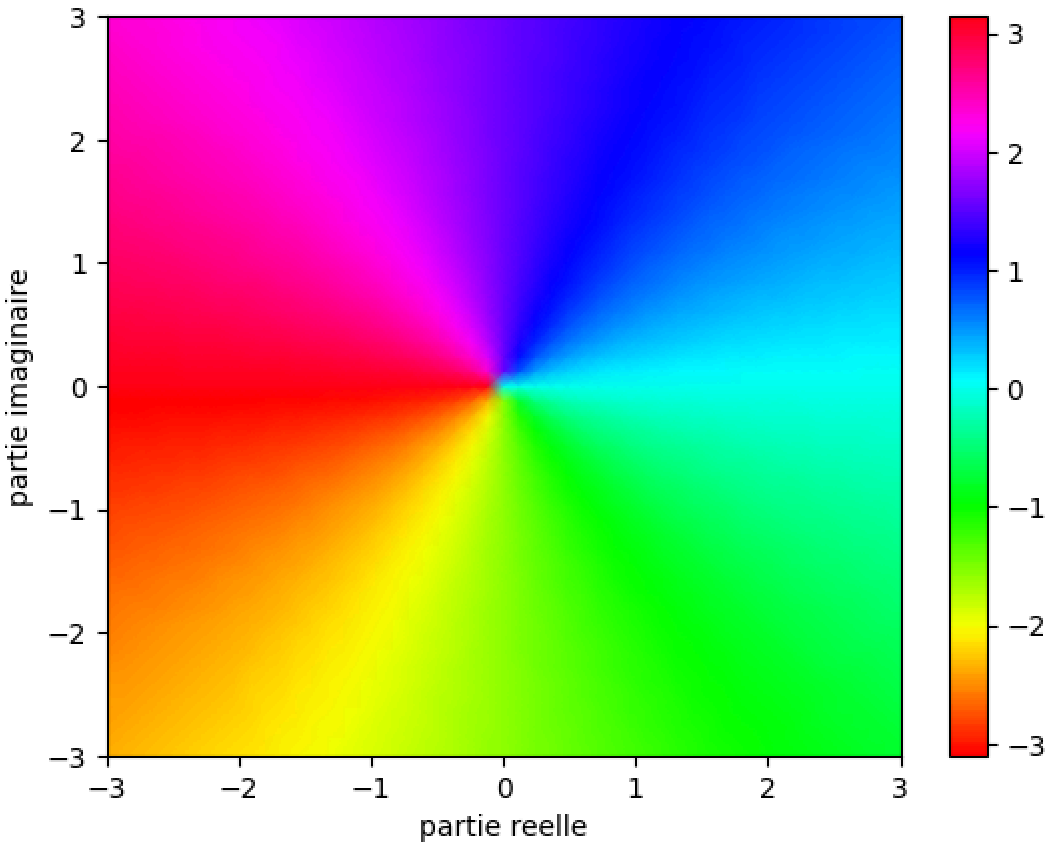


In [12]:

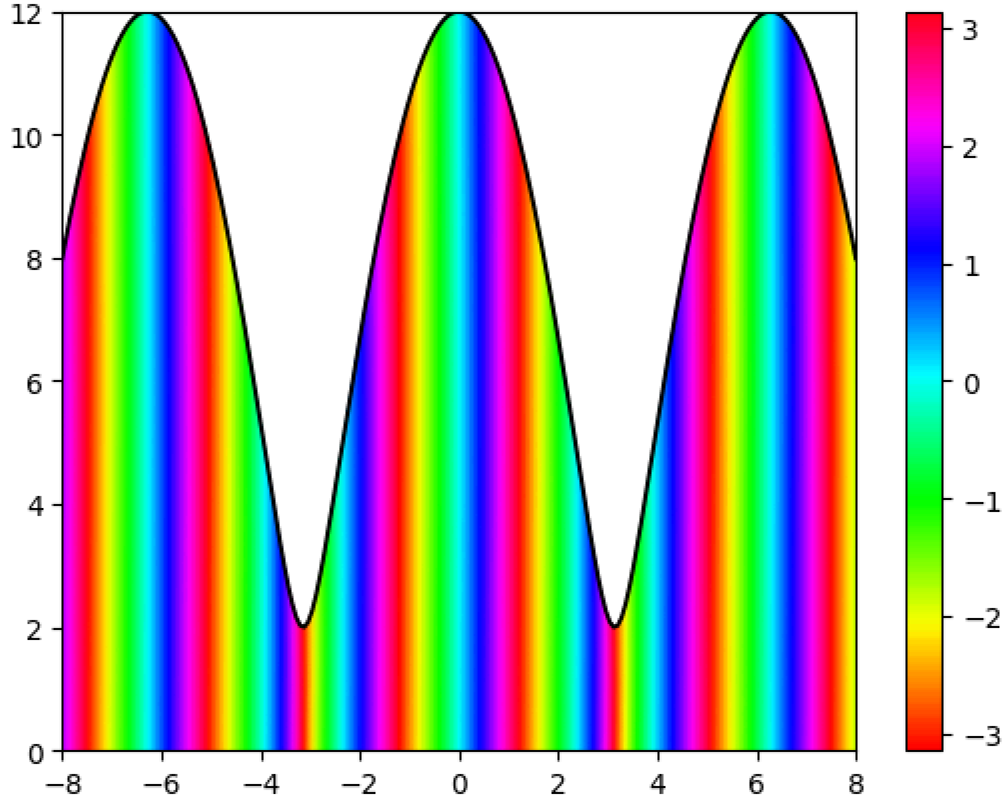
```
import numpy as np
import matplotlib.pyplot as plt

x = np.linspace(-3, 3, 51)
y = np.linspace(-3, 3, 51)

X, Y = np.meshgrid(x, y)

C = np.angle(X + 1j * Y)

plt.pcolormesh(X, Y, C, shading="gouraud")
plt.colorbar()
plt.xlabel("partie reelle")
plt.ylabel("partie imaginaire")
plt.show()
```



In [13]:
```
import numpy as np
import matplotlib.pyplot as plt

x = np.linspace(-3, 3, 51)
y = np.linspace(-3, 3, 51)

X, Y = np.meshgrid(x, y)

C = np.angle(X + 1j * Y)

plt.pcolormesh(X, Y, C, shading="gouraud", cmap=plt.cm.hsv)
plt.colorbar()
plt.xlabel("partie reelle")
plt.ylabel("partie imaginaire")
plt.show()
```



In [14]:
```
import numpy as np
import matplotlib.pyplot as plt

x = np.linspace(-8, 8, 201)
z = 5 * np.exp(2j * x) + 7 * np.exp(3j * x)

X = np.array([x, x])
y0 = np.zeros(len(x))
y = np.abs(z)
Y = np.array([y0, y])

Z = np.array([z, z])
C = np.angle(Z)

plt.plot(x, y, "k")
plt.pcolormesh(X, Y, C, shading="gouraud", cmap=plt.cm.hsv,
               vmin=-np.pi, vmax=np.pi)
plt.colorbar()
plt.show()
```



In [16]:
```
import scipy.integrate as si
import numpy as np

# L'intégrale de la fonction sin entre 0 et pi
I = si.quad(np.sin, 0, np.pi)
print(I)
```

```
(2.0, 2.220446049250313e-14)
```

In [ ]: