In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import math
% matplotlib inline
```

$$P = \begin{pmatrix} 1 & 1 \\ -i & i \end{pmatrix}$$
$$D = \begin{pmatrix} i & 0 \\ 0 & -i \end{pmatrix}$$

とすると、

$$P^{-1}JP = D$$

と対角化できる。

ここで、

$$x = \begin{bmatrix} p \\ q \end{bmatrix}$$

とし、

$$x = Pu$$

と置くと、

$$\begin{aligned} \dot{u} &= P^{-1}\dot{x} \\ &= P^{-1}Jx \\ &= P^{-1}JPP^{-1}x \\ &= Du \end{aligned}$$

つまり、

$$u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

として、

$$\dot{u_1} = iu_1$$

$$\dot{u_2} = -iu_2$$

$$u_1(0) = \frac{1}{2}$$

$$u_2(0) = \frac{1}{2}$$

と書ける。

よって、uについて数値的に考察すれば、

$$x = Pu$$

を用いて

$$x = \begin{bmatrix} p \\ q \end{bmatrix}$$

の挙動を数値的に考えていくことができる。

また、この方程式を解くと、解は

$$x = \begin{bmatrix} cos(t) \\ sin(t) \end{bmatrix}$$

となる。

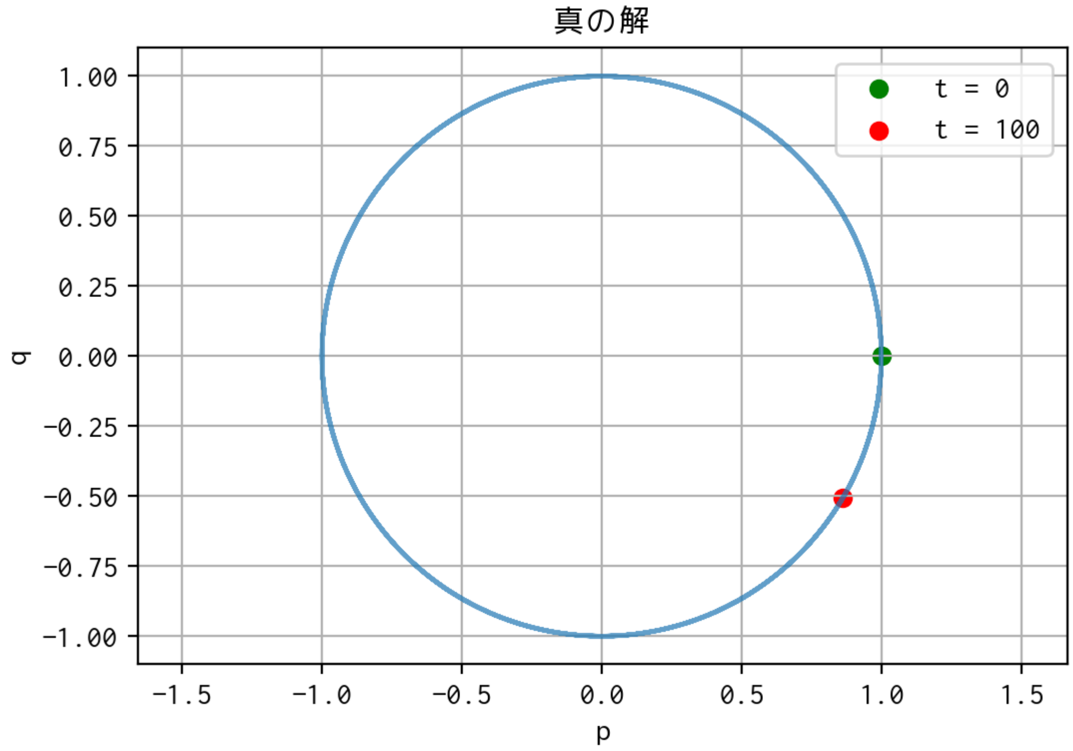
In [2]:

```python
P = np.array([[1, 1], [-1j, 1j]])

def u2x(u_1_array, u_2_array):
    return  np.array([np.dot(P, np.array([[u_1], [u_2]])) for u_1, u_2 in zip(u_1_array, u_2_array)]

def plot_x(x, title):
    plt.scatter(x[0, 0], x[0, 1], color='green', label='t = 0')
    plt.scatter(x[-1, 0], x[-1, 1], color='red', label='t = 100')
    plt.plot(x[:, 0], x[:, 1], alpha = 0.7)

    plt.legend()
    plt.title(title)
    plt.xlabel('p')
    plt.ylabel('q')
    plt.grid()
    plt.axis('equal')
    plt.show()
```

1 真の解

In [3]:

```
T = [i * 0.1 for i in range(1001)]
x = np.array([[[math.cos(t)], [math.sin(t)]] for t in T])

plot_x(x, '真の解')
```



2 陽的Euler法

In [4]:

```
def f_1(x):
    return 1j * x

def f_2(x):
    return -1j * x
```

In [5]:

```
def explicit_euler_method(f, x_0, h = 0.1, T = 100):
    t = 0
    xs = [x_0]
    while t < T:
        xs.append(h * f(xs[-1]) + xs[-1])
        t += h
    return xs
```

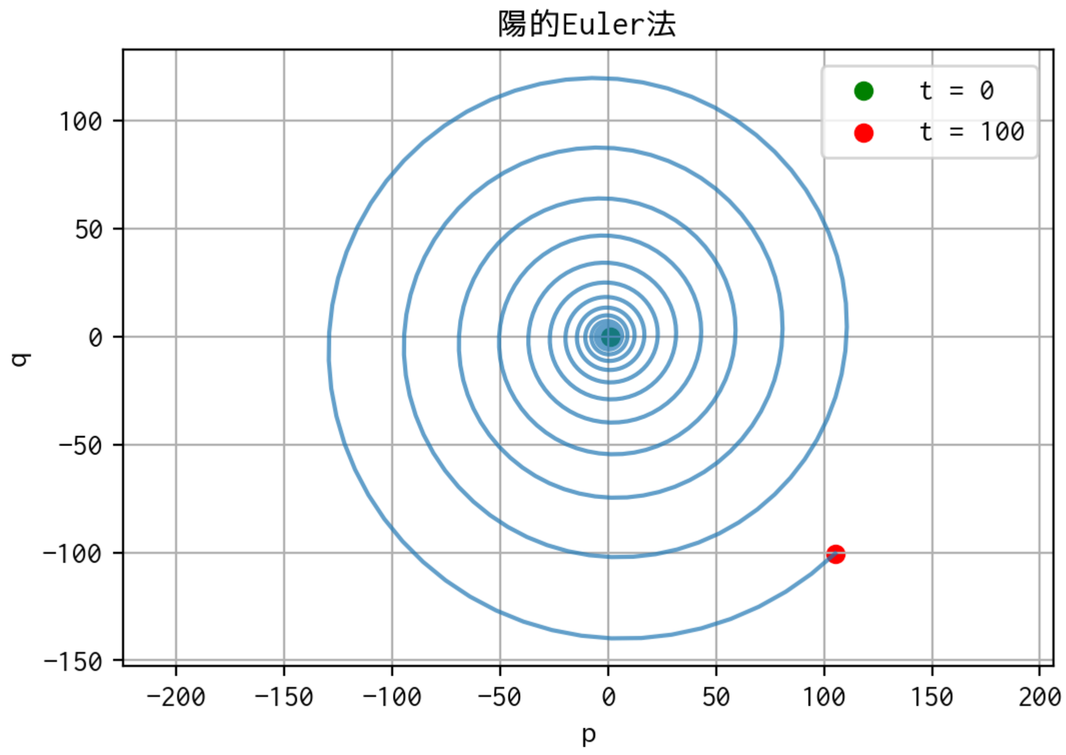
In [6]:

```
u_1 = explicit_euler_method(f_1, 0.5)
u_2 = explicit_euler_method(f_2, 0.5)
```

In [7]:

```
x = u2x(u_1, u_2)

plot_x(x, '陽的Euler法')
```

```
/Users/uedatomohiro/.pyenv/versions/anaconda3-5.1.0/lib/python3.6/site-packages/
numpy/core/numeric.py:544: ComplexWarning: Casting complex values to real discards
the imaginary part
  return array(a, dtype, copy=False, order=order, subok=True)
/Users/uedatomohiro/.pyenv/versions/anaconda3-5.1.0/lib/python3.6/site-packages/
numpy/core/numeric.py:492: ComplexWarning: Casting complex values to real discards
the imaginary part
  return array(a, dtype, copy=False, order=order)
```



3 陰的Euler法

In [8]:

```
def u_1_implicit_euler_method(x_0, h = 0.1, T = 100):
    t = 0
    xs = [x_0]
    while t < T:
        xs.append(xs[-1] / (1 - 1j * h))
        t += h
    return xs
```

In [9]:

```
def u_2_implicit_euler_method(x_0, h = 0.1, T = 100):
    t = 0
    xs = [x_0]
    while t < T:
        xs.append(xs[-1] / (1 + 1j * h))
        t += h
    return xs
```

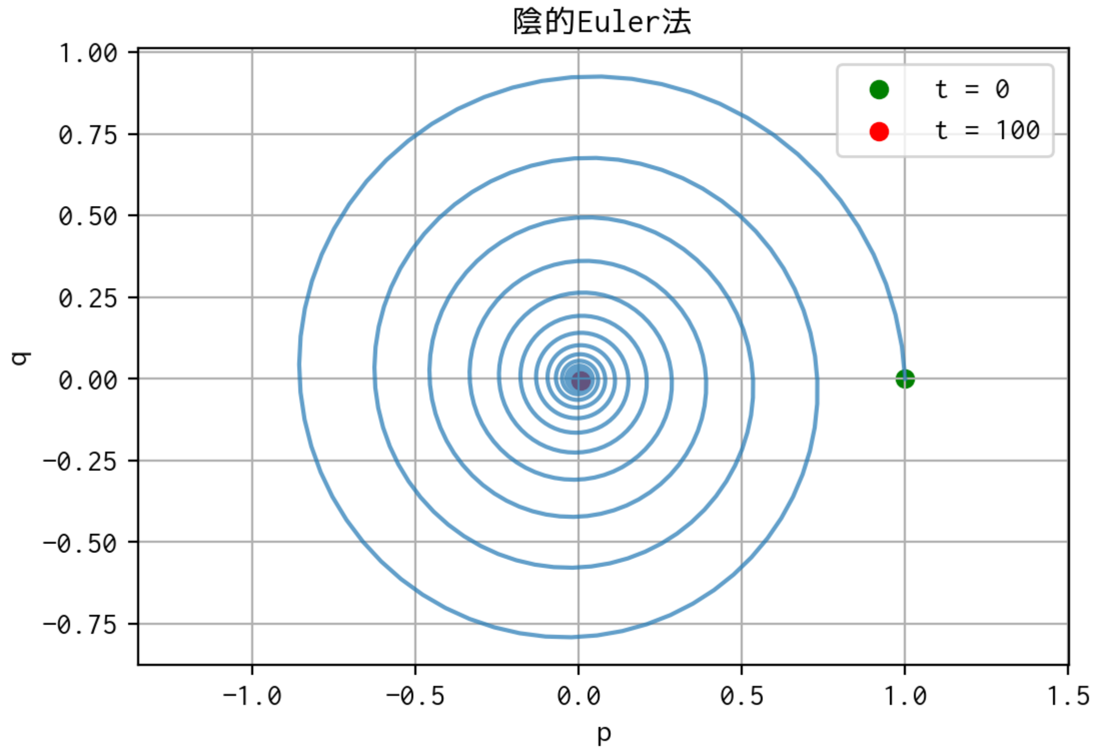
In [10]:

```
u_1 = u_1_implicit_euler_method(0.5)
u_2 = u_2_implicit_euler_method(0.5)
```

In [11]:

```
x = u2x(u_1, u_2)

plot_x(x, '陰的Euler法')
```

```
/Users/uedatomohiro/.pyenv/versions/anaconda3-5.1.0/lib/python3.6/site-packages/
numpy/core/numeric.py:544: ComplexWarning: Casting complex values to real discards
the imaginary part
  return array(a, dtype, copy=False, order=order, subok=True)
/Users/uedatomohiro/.pyenv/versions/anaconda3-5.1.0/lib/python3.6/site-packages/
numpy/core/numeric.py:492: ComplexWarning: Casting complex values to real discards
the imaginary part
  return array(a, dtype, copy=False, order=order)
```



4 台形法

In [12]:

```
def u_1_trapezoidal_method(x_0, h = 0.1, T = 100):
    t = 0
    xs = [x_0]
    while t < T:
        xs.append((2 + h * 1j) / (2 - h * 1j) * xs[-1])
        t += h
    return xs
```

In [13]:

```
def u_2_trapezoidal_method(x_0, h = 0.1, T = 100):
    t = 0
    xs = [x_0]
    while t < T:
        xs.append((2 - h * 1j) / (2 + h * 1j) * xs[-1])
        t += h
    return xs
```

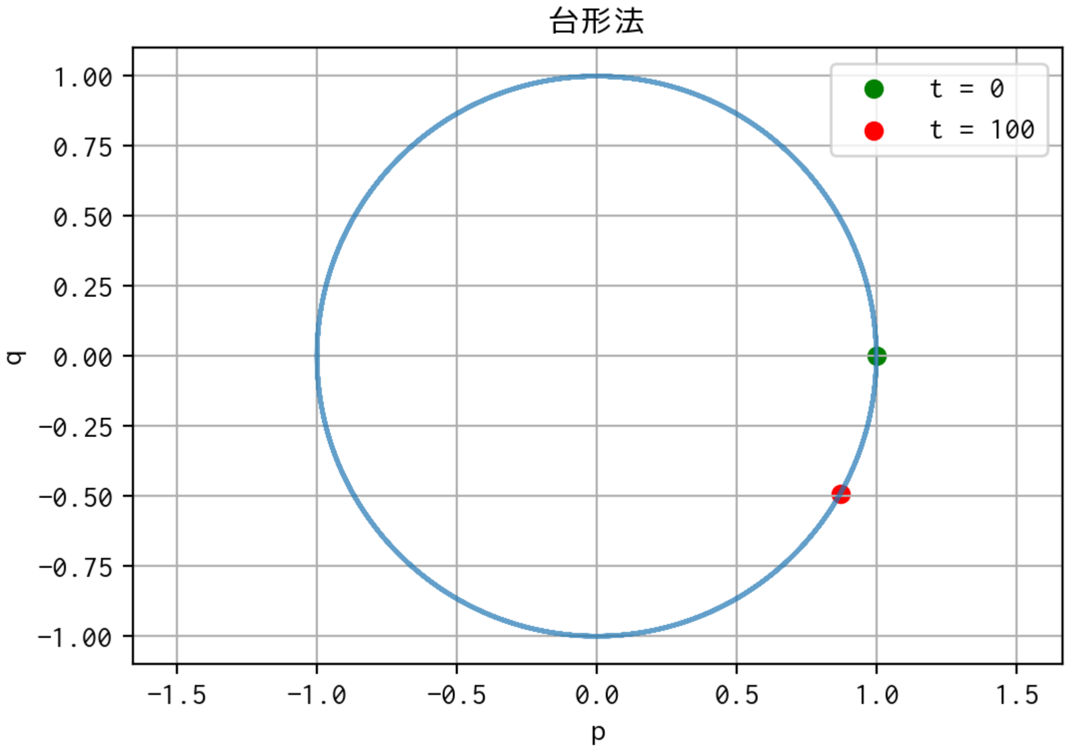
In [14]:

```
u_1 = u_1_trapezoidal_method(0.5)
u_2 = u_2_trapezoidal_method(0.5)
```

In [15]:

```
x = u2x(u_1, u_2)

plot_x(x, '台形法')
```

```
/Users/uedatomohiro/.pyenv/versions/anaconda3-5.1.0/lib/python3.6/site-packages/
numpy/core/numeric.py:544: ComplexWarning: Casting complex values to real discards
the imaginary part
  return array(a, dtype, copy=False, order=order, subok=True)
/Users/uedatomohiro/.pyenv/versions/anaconda3-5.1.0/lib/python3.6/site-packages/
numpy/core/numeric.py:492: ComplexWarning: Casting complex values to real discards
the imaginary part
  return array(a, dtype, copy=False, order=order)
```



5 Runge-Kutta法

In [16]:

```
def runge_kutta_method(f, x_0, h = 0.1, T = 100):
    t = 0
    xs = [x_0]
    while t < T:
        y_n = xs[-1]
        k_1 = f(y_n)
        k_2 = f(y_n + h / 2 * k_1)
        k_3 = f(y_n + h / 2 * k_2)
        k_4 = f(y_n + h * k_3)
        xs.append(xs[-1] + h * (k_1 / 6 + k_2 / 3 + k_3 / 3 + k_4 / 6))
        t += h
    return xs
```

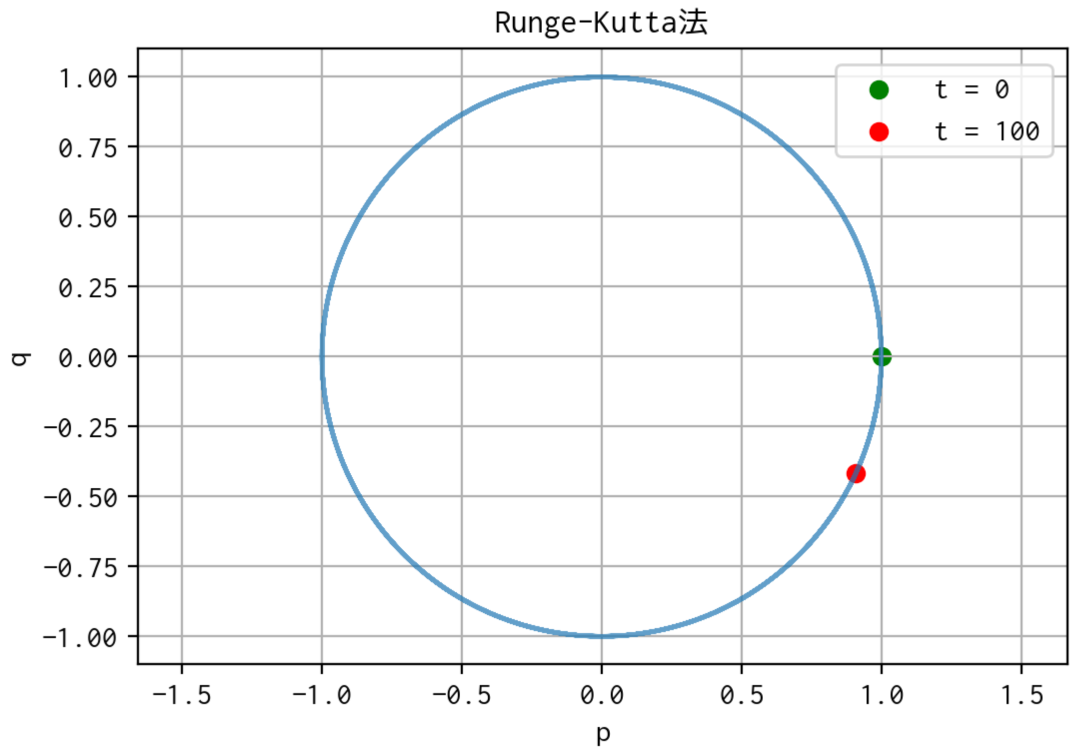
In [17]:

```
u_1 = runge_kutta_method(f_1, 0.5)
u_2 = runge_kutta_method(f_2, 0.5)
```

In [18]:

```
x = u2x(u_1, u_2)

plot_x(x, 'Runge-Kutta法')
```

```
/Users/uedatomohiro/.pyenv/versions/anaconda3-5.1.0/lib/python3.6/site-packages/
numpy/core/numeric.py:544: ComplexWarning: Casting complex values to real discards
the imaginary part
  return array(a, dtype, copy=False, order=order, subok=True)
/Users/uedatomohiro/.pyenv/versions/anaconda3-5.1.0/lib/python3.6/site-packages/
numpy/core/numeric.py:492: ComplexWarning: Casting complex values to real discards
the imaginary part
  return array(a, dtype, copy=False, order=order)
```


Runge-Kutta法
● t = 0
● t = 100
q
p