1.

```
        if np.max(np.abs(u - u_old)) < tol:
            print(f'Converged after {iteration} iterations')
            break

# 顯示結果
print(u)
```

input

```
Converged after 84 iterations
[[ 1.00000000e+00  9.51056516e-01  8.09016994e-01  5.87785252e-01
   3.09016994e-01  6.12323400e-17]
 [ 9.51056516e-01  7.53226426e-01  5.64622994e-01  3.68086061e-01
   1.72807029e-01  0.00000000e+00]
 [ 8.09016994e-01  5.55909927e-01  3.47643255e-01  1.76350631e-01
   5.30880436e-02  0.00000000e+00]
 [ 5.87785252e-01  3.33233604e-01  1.32651078e-01 -4.97139562e-03
  -5.88794667e-02  0.00000000e+00]
 [ 3.09016994e-01  8.58091605e-02 -8.68584755e-02 -1.82342436e-01
  -1.66745322e-01  0.00000000e+00]
 [ 6.12323400e-17 -1.73193865e-01 -3.05627718e-01 -3.53906910e-01
  -2.69906871e-01  0.00000000e+00]
 [-3.09016994e-01 -4.24254494e-01 -5.11145973e-01 -5.11640437e-01
  -3.64159197e-01  0.00000000e+00]
 [-5.87785252e-01 -6.45217714e-01 -6.86173630e-01 -6.42016579e-01
  -4.41309699e-01  0.00000000e+00]
 [-8.09016994e-01 -8.14472849e-01 -8.09944316e-01 -7.24386266e-01
  -4.86319289e-01  0.00000000e+00]
 [-9.51056516e-01 -9.15786353e-01 -8.58892013e-01 -7.25485091e-01
  -4.67873361e-01  0.00000000e+00]
 [-1.00000000e+00 -9.51056516e-01 -8.09016994e-01 -5.87785252e-01
  -3.09016994e-01 -6.12323400e-17]]
```

2.

```
print_solution(T_fd, "Forward-Difference")
print_solution(T_bd, "Backward-Difference")
print_solution(T_cn, "Crank-Nicolson")
```

input

```
=== Forward-Difference ===
t = 0.0 : ['0.000e+00', '2.000e+01', '4.000e+01', '6.000e+01', '8.000e+01', '1.000e+02']
t = 0.5 : ['3.359e+02', '4.367e+02', '3.971e+02', '3.725e+02', '3.578e+02', '1.200e+02']
t = 1.0 : ['-1.266e+04', '-1.646e+04', '1.684e+03', '1.305e+03', '-2.928e+04', '1.400e+02']
t = 1.5 : ['2.212e+06', '2.876e+06', '-2.155e+06', '-4.016e+06', '7.463e+06', '1.600e+02']
t = 2.0 : ['-5.804e+08', '-7.545e+08', '3.327e+08', '1.739e+09', '-2.332e+09', '1.800e+02']
t = 2.5 : ['1.280e+11', '1.664e+11', '6.243e+10', '-7.037e+11', '7.860e+11', '2.000e+02']
t = 3.0 : ['-1.409e+13', '-1.832e+13', '-9.047e+13', '2.869e+14', '-2.788e+14', '2.200e+02']
t = 3.5 : ['-7.157e+15', '-9.304e+15', '5.883e+16', '-1.191e+17', '1.033e+17', '2.400e+02']
t = 4.0 : ['7.279e+18', '9.463e+18', '-3.167e+19', '5.026e+19', '-3.978e+19', '2.600e+02']
t = 4.5 : ['-4.470e+21', '-5.811e+21', '1.572e+22', '-2.151e+22', '1.584e+22', '2.800e+02']
t = 5.0 : ['2.356e+24', '3.063e+24', '-7.469e+24', '9.301e+24', '-6.483e+24', '3.000e+02']
t = 5.5 : ['-1.157e+27', '-1.504e+27', '3.461e+27', '-4.052e+27', '2.712e+27', '3.200e+02']
t = 6.0 : ['5.466e+29', '7.106e+29', '-1.579e+30', '1.775e+30', '-1.154e+30', '3.400e+02']
t = 6.5 : ['-2.524e+32', '-3.281e+32', '7.134e+32', '-7.802e+32', '4.968e+32', '3.600e+02']
t = 7.0 : ['1.149e+35', '1.494e+35', '-3.202e+35', '3.439e+35', '-2.158e+35', '3.800e+02']
t = 7.5 : ['-5.184e+37', '-6.739e+37', '1.431e+38', '-1.518e+38', '9.433e+37', '4.000e+02']
t = 8.0 : ['2.325e+40', '3.022e+40', '-6.379e+40', '6.710e+40', '-4.141e+40', '4.200e+02']
t = 8.5 : ['-1.038e+43', '-1.350e+43', '2.838e+43', '-2.968e+43', '1.823e+43', '4.400e+02']
t = 9.0 : ['4.627e+45', '6.014e+45', '-1.261e+46', '1.314e+46', '-8.044e+45', '4.600e+02']
t = 9.5 : ['-2.058e+48', '-2.675e+48', '5.597e+48', '-5.817e+48', '3.554e+48', '4.800e+02']
t = 10.0 : ['9.140e+50', '1.188e+51', '-2.483e+51', '2.576e+51', '-1.572e+51', '5.000e+02']
```

```
print_solution(T_cn, "Crank-Nicolson")
```

input

```
=== Backward-Difference ===
t = 0.0 : ['0.000e+00', '2.000e+01', '4.000e+01', '6.000e+01', '8.000e+01', '1.000e+02']
t = 0.5 : ['5.467e+01', '7.107e+01', '8.532e+01', '9.801e+01', '1.095e+02', '1.200e+02']
t = 1.0 : ['6.489e+01', '8.436e+01', '1.009e+02', '1.154e+02', '1.283e+02', '1.400e+02']
t = 1.5 : ['7.426e+01', '9.654e+01', '1.155e+02', '1.320e+02', '1.467e+02', '1.600e+02']
t = 2.0 : ['8.360e+01', '1.087e+02', '1.300e+02', '1.486e+02', '1.651e+02', '1.800e+02']
t = 2.5 : ['9.294e+01', '1.208e+02', '1.445e+02', '1.651e+02', '1.835e+02', '2.000e+02']
t = 3.0 : ['1.023e+02', '1.330e+02', '1.590e+02', '1.817e+02', '2.018e+02', '2.200e+02']
t = 3.5 : ['1.116e+02', '1.451e+02', '1.735e+02', '1.983e+02', '2.202e+02', '2.400e+02']
t = 4.0 : ['1.210e+02', '1.572e+02', '1.880e+02', '2.148e+02', '2.386e+02', '2.600e+02']
t = 4.5 : ['1.303e+02', '1.694e+02', '2.025e+02', '2.314e+02', '2.570e+02', '2.800e+02']
t = 5.0 : ['1.396e+02', '1.815e+02', '2.171e+02', '2.480e+02', '2.754e+02', '3.000e+02']
t = 5.5 : ['1.490e+02', '1.937e+02', '2.316e+02', '2.645e+02', '2.937e+02', '3.200e+02']
t = 6.0 : ['1.583e+02', '2.058e+02', '2.461e+02', '2.811e+02', '3.121e+02', '3.400e+02']
t = 6.5 : ['1.676e+02', '2.179e+02', '2.606e+02', '2.977e+02', '3.305e+02', '3.600e+02']
t = 7.0 : ['1.770e+02', '2.301e+02', '2.751e+02', '3.142e+02', '3.489e+02', '3.800e+02']
t = 7.5 : ['1.863e+02', '2.422e+02', '2.896e+02', '3.308e+02', '3.672e+02', '4.000e+02']
t = 8.0 : ['1.957e+02', '2.544e+02', '3.041e+02', '3.473e+02', '3.856e+02', '4.200e+02']
t = 8.5 : ['2.050e+02', '2.665e+02', '3.186e+02', '3.639e+02', '4.040e+02', '4.400e+02']
t = 9.0 : ['2.143e+02', '2.786e+02', '3.331e+02', '3.805e+02', '4.224e+02', '4.600e+02']
t = 9.5 : ['2.237e+02', '2.908e+02', '3.476e+02', '3.970e+02', '4.408e+02', '4.800e+02']
t = 10.0 : ['2.330e+02', '3.029e+02', '3.622e+02', '4.136e+02', '4.591e+02', '5.000e+02']
```

```
print_solution(T_cn, "Crank-Nicolson")
```

input

```
=== Crank-Nicolson ===
t = 0.0 : ['0.000e+00', '2.000e+01', '4.000e+01', '6.000e+01', '8.000e+01', '1.000e+02']
t = 0.5 : ['6.544e+01', '8.507e+01', '9.909e+01', '1.090e+02', '1.158e+02', '1.200e+02']
t = 1.0 : ['5.618e+01', '7.303e+01', '8.980e+01', '1.065e+02', '1.232e+02', '1.400e+02']
t = 1.5 : ['8.283e+01', '1.077e+02', '1.265e+02', '1.408e+02', '1.518e+02', '1.600e+02']
t = 2.0 : ['7.605e+01', '9.886e+01', '1.204e+02', '1.408e+02', '1.606e+02', '1.800e+02']
t = 2.5 : ['1.004e+02', '1.305e+02', '1.540e+02', '1.729e+02', '1.880e+02', '2.000e+02']
t = 3.0 : ['9.576e+01', '1.245e+02', '1.507e+02', '1.750e+02', '1.979e+02', '2.200e+02']
t = 3.5 : ['1.181e+02', '1.535e+02', '1.818e+02', '2.050e+02', '2.242e+02', '2.400e+02']
t = 4.0 : ['1.153e+02', '1.499e+02', '1.809e+02', '2.090e+02', '2.351e+02', '2.600e+02']
t = 4.5 : ['1.359e+02', '1.767e+02', '2.098e+02', '2.373e+02', '2.605e+02', '2.800e+02']
t = 5.0 : ['1.348e+02', '1.752e+02', '2.109e+02', '2.429e+02', '2.723e+02', '3.000e+02']
t = 5.5 : ['1.539e+02', '2.001e+02', '2.379e+02', '2.697e+02', '2.969e+02', '3.200e+02']
t = 6.0 : ['1.542e+02', '2.004e+02', '2.408e+02', '2.768e+02', '3.095e+02', '3.400e+02']
t = 6.5 : ['1.719e+02', '2.235e+02', '2.661e+02', '3.022e+02', '3.333e+02', '3.600e+02']
t = 7.0 : ['1.734e+02', '2.254e+02', '2.706e+02', '3.105e+02', '3.465e+02', '3.800e+02']
t = 7.5 : ['1.901e+02', '2.471e+02', '2.944e+02', '3.348e+02', '3.697e+02', '4.000e+02']
t = 8.0 : ['1.926e+02', '2.504e+02', '3.003e+02', '3.441e+02', '3.836e+02', '4.200e+02']
t = 8.5 : ['2.083e+02', '2.708e+02', '3.228e+02', '3.674e+02', '4.062e+02', '4.400e+02']
t = 9.0 : ['2.117e+02', '2.752e+02', '3.299e+02', '3.777e+02', '4.206e+02', '4.600e+02']
t = 9.5 : ['2.265e+02', '2.945e+02', '3.513e+02', '4.001e+02', '4.428e+02', '4.800e+02']
t = 10.0 : ['2.308e+02', '3.000e+02', '3.594e+02', '4.112e+02', '4.575e+02', '5.000e+02']
```

3.

```
Nr = 10   # r 方向格點數
Ntheta = 10  # theta 方向格點數

r_start = 0.5
r_end = 1.0
theta_start = 0.0
theta_end = np.pi / 3

delta_r = (r_end - r_start) / (Nr - 1)
delta_theta = (theta_end - theta_start) / (Ntheta - 1)

r = np.linspace(r_start, r_end, Nr)
theta = np.linspace(theta_start, theta_end, Ntheta)
```

Input

```
解得溫度場 T[r, θ]:
[[0.000000e+00 5.000000e+01 5.000000e+01 5.000000e+01 5.000000e+01 5.000000e+01 5.000000e+01 5.000000e
+01 5.000000e+01 0.000000e+00]
 [0.000000e+00 2.940727e+01 4.129034e+01 4.683108e+01 4.914106e+01 4.914106e+01 4.683108e+01 4.129034e
+01 2.940727e+01 0.000000e+00]
 [4.856803e-14 2.310331e+01 3.787171e+01 4.623665e+01 4.998890e+01 4.998890e+01 4.623665e+01 3.787171e
+01 2.310331e+01 1.950310e-13]
 [-2.201700e-12 2.221426e+01 3.848938e+01 4.838726e+01 5.295394e+01 5.295394e+01 4.838726e+01 3.848938
e+01 2.221426e+01 -3.331328e-12]
 [4.785931e-13 2.432088e+01 4.219791e+01 5.299101e+01 5.793051e+01 5.793051e+01 5.299101e+01 4.219791e
+01 2.432088e+01 2.305878e-12]
 [3.784257e-12 2.898351e+01 4.861315e+01 5.971736e+01 6.461735e+01 6.461735e+01 5.971736e+01 4.861315e
+01 2.898351e+01 1.111955e-11]
 [9.835502e-13 3.672805e+01 5.768802e+01 6.823729e+01 7.262592e+01 7.262592e+01 6.823729e+01 5.768802e
+01 3.672805e+01 -4.226823e-12]
 [5.284971e-13 4.898232e+01 6.948479e+01 7.816472e+01 8.151204e+01 8.151204e+01 7.816472e+01 6.948479e
+01 4.898232e+01 -3.229012e-15]
 [-2.271996e-13 6.846213e+01 8.387437e+01 8.898247e+01 9.079613e+01 9.079613e+01 8.898247e+01 8.387437
e+01 6.846213e+01 0.000000e+00]
 [0.000000e+00 1.000000e+02 1.000000e+02 1.000000e+02 1.000000e+02 1.000000e+02 1.000000e+02 1.000000e
+02 1.000000e+02 0.000000e+00]]
```

4.

```
    + 0.5 * (dt / dx) ** 2 * (p[0, 2:] - 2 * p[0, 1:-1] + p[0, :-2])
)

# 主時間迴圈：用顯式有限差分法更新解
for n in range(1, nt - 1):
    p[n + 1, 1:-1] = (
        2 * p[n, 1:-1]
        - p[n - 1, 1:-1]
        + (dt / dx) ** 2 * (p[n, 2:] - 2 * p[n, 1:-1] + p[n, :-2])
    )

# 顯示結果：每個時間步對應一行
print("Results (each row = p(x) at a time t):\n")
for n in range(nt):
    row = ", ".join([f"{val:.4f}" for val in p[n]])
    print(f"t = {n*dt:.1f} -> [ {row} ]")
```

Input

```
Results (each row = p(x) at a time t):

t = 0.0 -> [ 1.0000, 0.8090, 0.3090, -0.3090, -0.8090, -1.0000, -0.8090, -0.3090, 0.30
90, 0.8090, 2.0000 ]
t = 0.1 -> [ 1.0000, 1.0238, 0.8476, 0.3476, -0.2852, -0.8090, -1.0238, -0.8476, -0.34
76, 0.7852, 2.0000 ]
t = 0.2 -> [ 1.0000, 1.0385, 1.0624, 0.8714, 0.3476, -0.3090, -0.8476, -1.0624, -0.871
4, 0.8434, 2.0000 ]
t = 0.3 -> [ 1.0000, 1.0385, 1.0624, 1.0624, 0.8476, 0.3090, -0.3476, -0.8714, 0.1286,
 0.8434, 2.0000 ]
t = 0.4 -> [ 1.0000, 1.0238, 1.0385, 1.0385, 1.0238, 0.8090, 0.7852, 0.8434, 0.8434, 1
.2852, 2.0000 ]
t = 0.5 -> [ 1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.5000, 2.0000, 2.0000, 2.0000, 2
.0000, 2.0000 ]
t = 0.6 -> [ 1.0000, 0.9763, 0.9615, 0.9615, 1.4762, 2.1910, 2.7148, 3.1566, 3.1566, 2
.7148, 2.0000 ]
t = 0.7 -> [ 1.0000, 0.9615, 0.9376, 1.4376, 2.1524, 2.6910, 3.3476, 3.8714, 3.8714, 3
.1566, 2.0000 ]
t = 0.8 -> [ 1.0000, 0.9615, 1.4376, 2.1286, 2.6524, 3.3090, 3.8476, 4.0624, 3.8714, 3
.1566, 2.0000 ]
t = 0.9 -> [ 1.0000, 1.4762, 2.1524, 2.6524, 3.2852, 3.8090, 4.0238, 3.8476, 3.3476, 2
.7148, 2.0000 ]
t = 1.0 -> [ 1.0000, 2.1910, 2.6910, 3.3090, 3.8090, 4.0000, 3.8090, 3.3090, 2.6910, 2
.1910, 2.0000 ]
```