```python
# -*- coding: utf-8 -*-
"""
Created on Mon Nov 10 19:38:03 2025

@author: coled_agkeohi
"""

import numpy as np
from scipy.linalg import expm, solve
import matplotlib.pyplot as plt

A = np.array([
    [0,        0,       0,       0,        0,      0,      0,      0,      0],
    [5.0, -42.474,       0,       0,        0,      0,      0,      0,      0],
    [0,        0,   -1.180,      0,        0,      0,      0,      0,      0],
    [0,     42.474,     0,   -0.295,        0,      0,      0,      0,      0],
    [0,        0,    1.180,   0.001, -138.629,     0,      0,      0,      0],
    [0,        0,      0,     0.295,     0,   -0.008,     0,      0,      0],
    [0,        0,      0,       0,   138.629,  0.003,  -0.003,    0,      0],
    [0,        0,      0,       0,      0,      0,    0.003, -0.011,   0],
    [0,        0,      0,       0,      0,      0,      0,     0,   -0.005],
], dtype=float)

x0 = np.zeros(9); x0[0] = 1.0

#exact x(t) by matrix exponential

ts = np.linspace(0.0, 1.0, 200)
X = np.zeros((len(ts), 9))

for i,t in enumerate(ts):
    X[i] = expm(t*A) @ x0

x_exact_t1 = X[-1]

# backward Euler, one step size t=1

t = 1.0
I = np.eye(A.shape[0])
x_BE = solve(I-t*A, x0, assume_a='gen')

# compare error
x_error = x_exact_t1 - x_BE


plt.figure(figsize=(9,5))
for k in range(len(x_BE)):
    plt.plot(ts, X[:, k], label =f'$x_{k+1}(t)$', linewidth=2)

#overlay BE markers
plt.plot(np.full(9, t), x_BE, 'o', label='backward euler at t=1')

# overlay error markers
plt.plot(np.full(9, t), x_error, 'D', label = 'error $x_{exact}(t) - x_{BE}(T)$')\

plt.xlabel('t')
```

```
plt.ylabel('component value')
plt.title('Exact solution via $e^{At}$ with BE marker at $t=1$')
plt.legend(ncol=3, fontsize=8)
plt.show()
```