File - C:\Code\Python\GWProgramming2\1Dtransport-local\1D Finite Element.py

```python
# Groundwater Modeling Coding Assignment #2
# Jim Finnegan
# 1D Transport Equation
# Finite Element Method

# SETUP
import numpy as np
from scipy.sparse import diags
from matplotlib import pyplot as plt

# user inputs
D = float(input('Enter D (m^2/d): '))
R = float(input('Enter R: '))

# other parameters
v, L, dx, t, dt = 0.1, 200, 2, 400, 10
# matrix dimensions
rows = int(t / dt) + 1
n_el = int(L / dx)
cols = int(L / dx) + 1

# initial conditions
C0 = np.zeros(cols)
C = np.zeros((rows, cols))
C[:, 0] = 1  # boundary condition: C/C0 = 1 at x=0

# CONSTRUCT STIFFNESS AND STORAGE MATRICES
# element matrices
alpha = (R * dx) / 6
lam_1 = D / dx
lam_2 = v / 2
Ae = [[lam_1 - lam_2, -lam_1 + lam_2], [-lam_1 - lam_2, lam_1 + lam_2
]]  # element stiffness matrix
Be = [[2 * alpha, alpha], [alpha, 2 * alpha]]  # element storage
matrix
# global matrices
A = np.zeros((cols, cols))
B = np.zeros((cols, cols))
for i in range(cols):
    A[i, i] += Ae[1][1]  # assemble Ae elements
    A[i, i - 1] += Ae[1][0]
    A[i - 1, i] += Ae[0][1]
    A[i - 1, i - 1] += Ae[0][0]
    B[i, i] += Be[1][1]  # assemble Be elements
    B[i, i - 1] += Be[1][0]
    B[i - 1, i] += Be[0][1]
    B[i - 1, i - 1] += Be[0][0]

# finite element equation matrices
A_f = (A/2 + B/dt)

# TIME STEPPING
for k in range(1, rows):
    b_f = np.dot((-A/2 + B/dt), C[k-1, :])  # solve LHS
    C[k, :] = np.linalg.solve(A_f, b_f)     # solve RHS
```

```
54      C[k][0] = 1                                 # boundary condition at
    source
55      C[k][-1] = 0                                # boundary condition at
    end
56
57  # PLOT
58  x = np.linspace(0, L, num=cols)
59  plt.plot(x, C[0, :])
60  plt.plot(x, C[10, :])
61  plt.plot(x, C[20, :])
62  plt.plot(x, C[30, :])
63  plt.plot(x, C[40, :])
64
65  plt.title('Finite Element Solution')
66  plt.xlabel('distance (m)')
67  plt.ylabel('C/C0')
68  plt.legend(['0 days', '100 days', '200 days', '300 days', '400 days'
    ])
69  plt.show()
70
```