10-13_Transient_Finite_Difference_Example

October 13, 2023

[1]:

```
import numpy as np
import matplotlib.pyplot as plt
import scipy.linalg
import scipy.sparse
from matplotlib.animation import FuncAnimation
from IPython import display
```

[26]:

```
N = 30
w = 1
dx = w / N
N2 = N**2
T1 = 50
T2 = 20
Ti = 20
dt = 14
Nt = 250
NN = N2 * Nt
NN
```

[26]:

```
225000
```

[27]:

```
alpha = 18.8e-6
Fo = alpha * dt / (dx**2)
Fo
```

[27]:

```
0.23688
```

[28]:

```
Ab = np.diag([1 - 4 * Fo] * N2)
diag1 = np.array([Fo] * (N2 - 1))
diag1[N-1::N] = 0
Ab += np.diag(diag1, 1)
Ab += np.diag(diag1, -1)
Ab += np.diag([Fo] * (N2 - N), N)
Ab += np.diag([Fo] * (N2 - N), -N)
```

[29]:

```
data = []
i = []
```

```python
j = []

for n in range(NN):
  data.append(-1)
  i.append(n)
  j.append(n)

for ind in np.ndindex(Ab.shape):
  if Ab[ind] != 0:
    for n in range(Nt - 1):
      data.append(Ab[ind])
      i.append(ind[0] + (n + 1) * N2)
      j.append(ind[1] + n * N2)

A = scipy.sparse.coo_array((data, (i, j)), shape=(NN, NN))
```

[30]:

```python
C0 = Ti * np.ones(N2)
C0[0] += Fo * (T2 + T1 - 2 * Ti)
C0[1:N-1] += Fo * (T2 - Ti)
C0[N-1] += 2 * Fo * (T2 - Ti)
C0[N:N2-N:N] += Fo * (T1 - Ti)
C0[2*N-1:N2-N:N] += Fo * (T2 - Ti)
C0[N2-N] += Fo * (T2 + T1 - 2 * Ti)
C0[N2-N+1:N2-1] += Fo * (T2 - Ti)
C0[-1] += 2 * Fo * (T2 - Ti)
C0 *= -1
```

[31]:

```python
Cb = np.zeros(N2)
Cb[0] = Fo * (T1 + T2)
Cb[1:N-1] = Fo * T2
Cb[N-1] = 2 * Fo * T2
Cb[N:N2-N:N] = Fo * T1
Cb[2 * N - 1:N2-N:N] = Fo * T2
Cb[N2-N] = Fo * (T1 + T2)
Cb[N2-N+1:N2-1] = Fo * T2
Cb[-1] = 2 * Fo * T2
Cb *= -1
```

[32]:

```python
data = []
i = []
j = []

for ind in np.ndindex(C0.shape):
  if C0[ind] != 0:
    data.append(C0[ind])
    i.append(ind[0])
    j.append(0)
```

```
for ind in np.ndindex(Cb.shape):
  if Cb[ind] != 0:
    for n in range(1, Nt):
      data.append(Cb[ind])
      i.append(ind[0] + n * N2)
      j.append(0)

C = scipy.sparse.coo_array((data, (i, j)), shape=(NN, 1))
```

[37]:

```
%%time
Tflat = scipy.sparse.linalg.spsolve(A.tocsr(), C)
```

```
CPU times: user 7.53 s, sys: 464 ms, total: 7.99 s
Wall time: 7.99 s
```

[38]:

```
T = Tflat.reshape(Nt, N, N)
```

[39]:

```
fig = plt.figure()
image = plt.imshow(T[0,:,:], vmin=T.min(), vmax=T.max())
plt.colorbar()

def update(frame):
  image.set_data(T[frame,:,:])

anim = FuncAnimation(
 fig,
 update,
 frames=Nt,
 interval=50,
)

video = anim.to_html5_video()
html = display.HTML(video)
display.display(html)
plt.close();
```

```
<IPython.core.display.HTML object>
```

[ ]:

```
```