Climate Modeling

October 24, 2024

PRACTICAL REPORT

TAYLOR'S EXPANSION AND FINITE DIFFERENCING

By:

Duong Thu Phuong

22BI13362

Simulate heat diffusion on a 2D plate. There are 2 heaters at the initial conditions. Boundary conditions are zero at every timestep.

lenX = 50

lenY = 50

k=0.05

dt=0.1

dx=0.5

dy=0.5

At the initial time, t=0:

T[30:50,30:50] = 1

T[60:80,60:80] = 0.8

T=0 elsewhere

[1]:
```
import numpy as np
import matplotlib.pyplot as plt
```

[2]:
```
lenx = 50
leny = 50
dx = 0.5
dy = 0.5
dt = 0.1
k = 0.05
nt = 1000
nx = int(lenx/dx)+1
ny = int(leny/dy)+1
x = np.linspace(0, lenx, nx)
y = np.linspace(0, leny, ny)
```

Heat equation:

$$\frac{\partial T}{\partial t} = k\left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}\right)$$

Laplacian approximation:

$$u_{i,j}^{t+1} = u_{i,j}^{t} + k \cdot \Delta t \cdot \nabla^2 u_{i,j}$$

[3]:
```
def laplacian(heat, coef, st):
    heat_new = np.zeros_like(heat)
    for i in range(st, nx - st):
        for j in range(st, ny - st):
            heat_new[i, j] = heat[i, j] + k * dt * coef(heat, i, j)
    return heat_new
```

5 point square stencil:

$$\nabla^2 u \approx \frac{u_{i+1,j+1} + u_{i-1,j-1} + u_{i-1,j+1} + u_{i+1,j-1} - 4u_{i,j}}{2 \cdot \Delta x^2}$$

5 point diamond stencil:

$$\nabla^2 u \approx \frac{u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - 4u_{i,j}}{\Delta x^2}$$

9 point square stencil:

$$\nabla^2 u \approx \frac{4(u_{i,j+1} + u_{i-1,j} + u_{i,j-1} + u_{i+1,j}) - (u_{i+1,j+1} + u_{i-1,j+1} + u_{i-1,j-1} + u_{i+1,j-1}) - 12u_{i,j}}{2 \cdot \Delta x^2}$$

9 point diamond stencil:

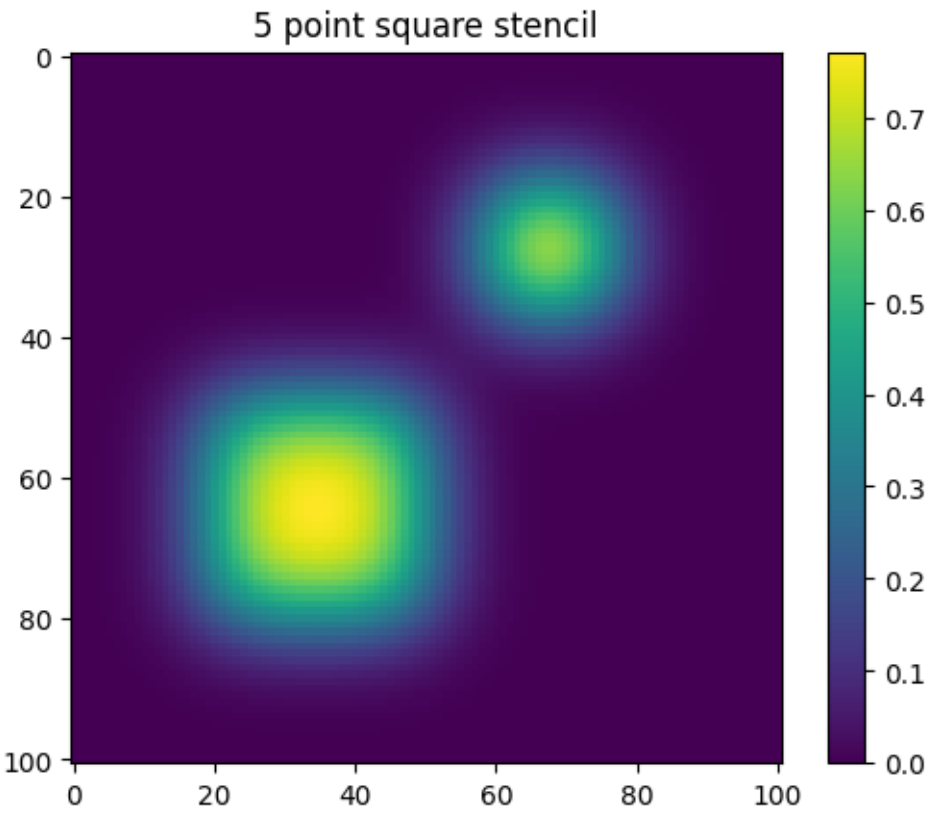
$$\nabla^2 u \approx \frac{16(u_{i-1,j} + u_{i,j-1} + u_{i+1,j} + u_{i,j+1}) - (u_{i,j+2} + u_{i-2,j} + u_{i,j-2} + u_{i+2,j}) - 60u_{i,j}}{12 \cdot \Delta x^2}$$

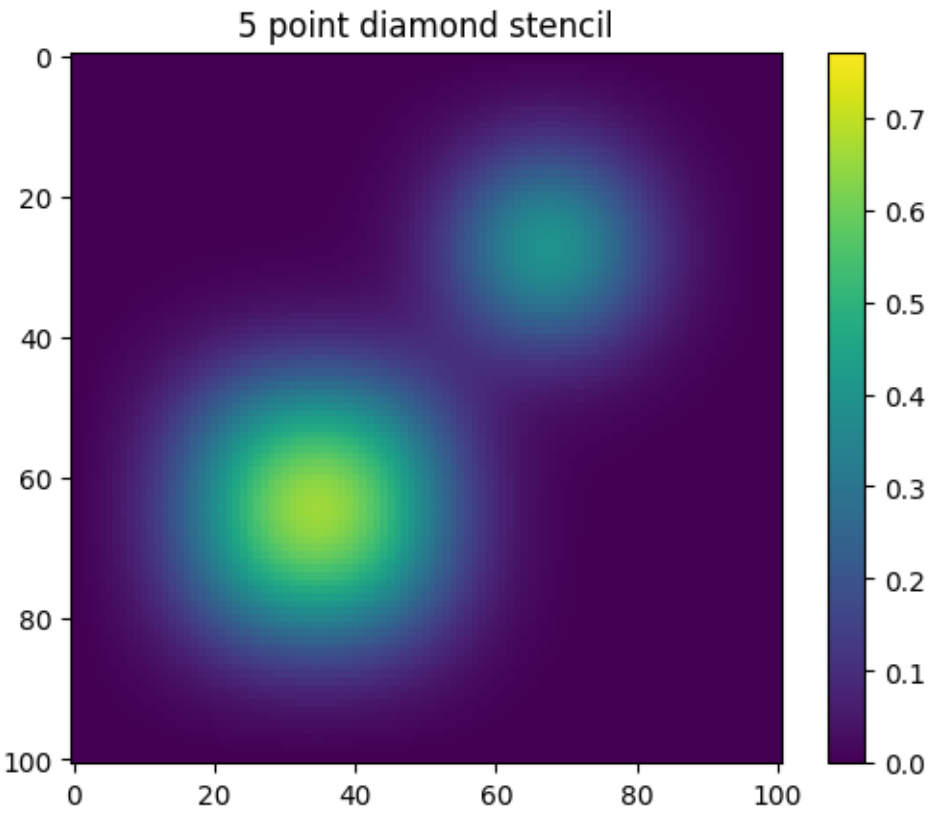
[4]:
```
s5 = lambda heat, i, j: (heat[i + 1, j + 1] + heat[i - 1, j - 1] + heat[i - 1,␣
 ↪j + 1] + heat[i + 1, j - 1] - 4 * heat[i, j]) / (2 * dx**2)
d5 = lambda heat, i, j: (heat[i + 1, j] + heat[i - 1, j] + heat[i, j + 1] +␣
 ↪heat[i, j - 1] - 4 * heat[i, j]) / (dx**2)
s9 = lambda heat, i, j: (4 * (heat[i, j + 1] + heat[i - 1, j] + heat[i, j - 1]␣
 ↪+ heat[i + 1, j]) - (heat[i + 1, j + 1] + heat[i - 1, j + 1] + heat[i - 1, j␣
 ↪- 1] + heat[i + 1, j - 1]) - 12 * heat[i, j]) / (2 * dx**2)
d9 = lambda heat, i, j: (16 * (heat[i - 1, j] + heat[i, j - 1] + heat[i + 1, j]␣
 ↪+ heat[i, j + 1]) - (heat[i, j + 2] + heat[i - 2, j] + heat[i, j - 2] +␣
 ↪heat[i + 2, j]) - 60 * heat[i, j]) / (12 * dx**2)
```

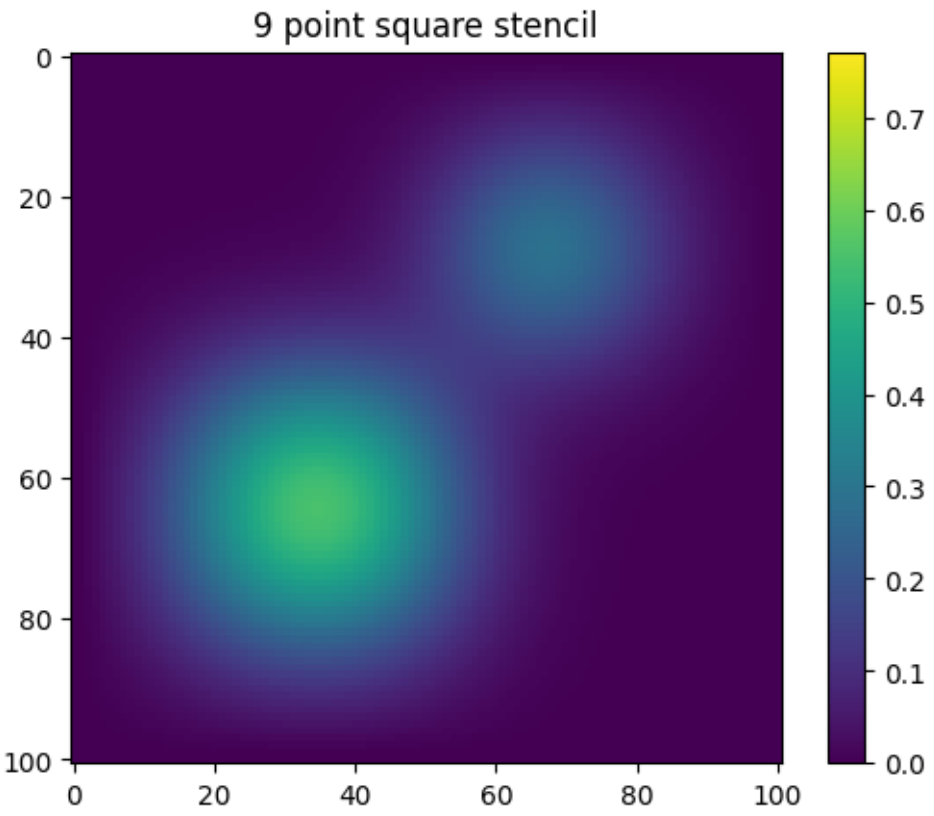
[5]:
```
heat = np.zeros((nx, ny))
heat[20:36, 60:76] = 1
heat[50:80, 20:50] = 0.8

methods = [s5, d5, s9, d9]
m_heats = [[heat.copy()] for _ in range(len(methods))]
```

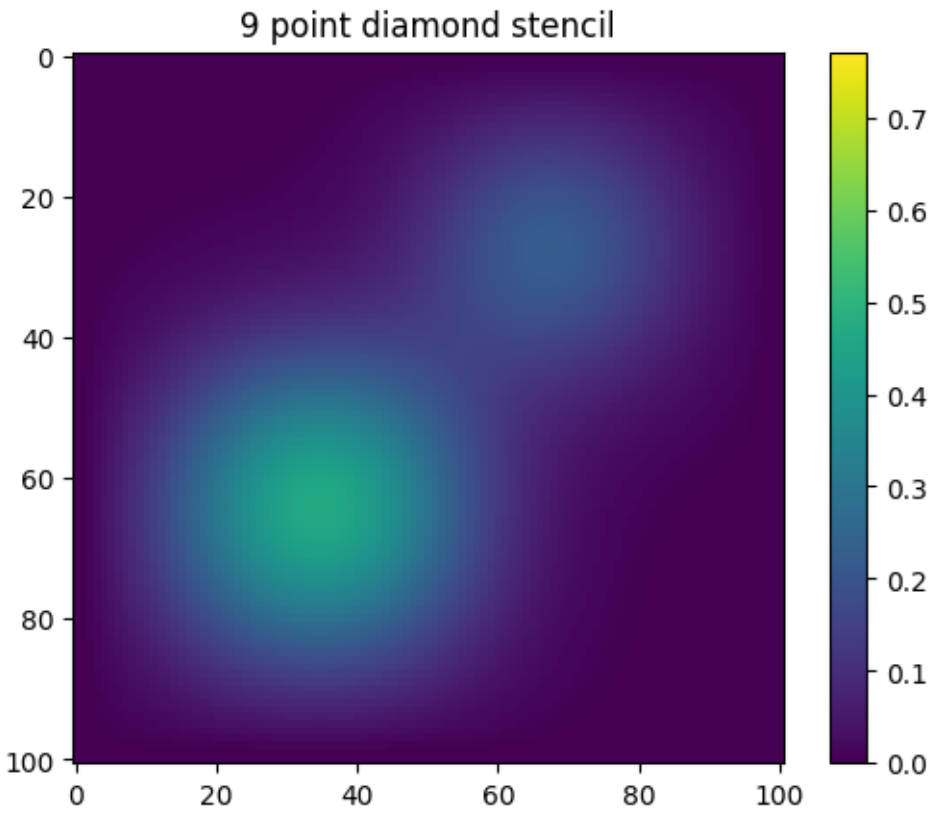
[6]:
```
for idx, method in enumerate(methods):
    for t in range(nt):
        heat = laplacian(heat, method, 1 if idx < 2 else 2)
        m_heats[idx].append(heat.copy())
```

[7]:
```
n = min(nt, min(len(m_heat) for m_heat in m_heats) - 1)
all_data = np.array([m_heat[n] for m_heat in m_heats])
vmin, vmax = all_data.min(), all_data.max()
titles = ["5 point square stencil", "5 point diamond stencil", "9 point square␣
 ↪stencil", "9 point diamond stencil"]

for i, m_heat in enumerate(m_heats):
    plt.figure()
    plt.title(titles[i])
    plt.imshow(m_heat[n], vmin=vmin, vmax=vmax)
    plt.colorbar()
    plt.show()
```


5 point square stencil
0
20
40
60
80
100
0
20
40
60
80
100
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0


5 point diamond stencil
0
20
40
60
80
100
0
20
40
60
80
100
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0


9 point square stencil
0
20
40
60
80
100
0
20
40
60
80
100
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0



- The 5 point square stencil creates a heat pattern with clear and round shapes from the initial heat spots.
- The output of 5 point diamond stencil is similar to the square stencil, but the pattern is a bit more spread out diagonally.
- The 5-point stencils make the heat spread more localized, make the edges sharper.
- The 9 point stencils include more points in th calculations, leading to a smoother and more averaged heat distribution.

[ ]: