Partial Differential Equations (PDEs)
Heat Equation

Alkassab Thaer Q39HXA

Abarca Zuniga Bernold Rodrigo E4NGIC

$$\frac{\partial T}{\partial t} = d.\frac{\partial^2 T}{\partial x^2} + S(x,t)$$

$$T(0,t) = \alpha_0(t)$$

$$T(L,t) = \alpha_L(t)$$

$$T(x,0) = \beta\ (t)$$

HEAT EQUATION FORMULA AND THE INITIAL AND BOUNDARY CONDITIONS

$$N = 100$$
$$w = np.zeros(N)$$
$$L = 10$$
$$h = L\,/\,(N + 1)$$
$$d = 0.1$$
$$steps = 40\_000$$
$$dt = 0.02$$

PARAMETERS FOR THE HEAT EQUATION

$def\ S(x, t)$:

$\quad return\ np.\exp(\,-(x - L/2) ** 2\ /\ L/2)\ +\ 1/(t + 1)$

$def\ alpha0(\ t\ )$:

$\quad return\ 20\ +\ 100\ *\ np.\sin(\ t\ /\ 100\ )$

$def\ alphaL(\ t\ )$:

$\quad return\ 100\ +\ 100\ *\ np.\cos(\ t\ /\ 100\ )$

$def\ beta(\ x\ )$:

$\quad return\ 300$

INITIAL AND BOUNDARY CONDITIONS

$$A = np.diag([-2]*N) + np.diag([1]*(N-1), 1) + np.diag([1]*(N-1), -1)$$

$$A = A*d/h/h$$

```
array([[-20.402,  10.201,   0.   , ...,   0.   ,   0.   ,   0.   ],
       [ 10.201, -20.402,  10.201, ...,   0.   ,   0.   ,   0.   ],
       [  0.   ,  10.201, -20.402, ...,   0.   ,   0.   ,   0.   ],
       ...,
       [  0.   ,   0.   ,   0.   , ..., -20.402,  10.201,   0.   ],
       [  0.   ,   0.   ,   0.   , ...,  10.201, -20.402,  10.201],
       [  0.   ,   0.   ,   0.   , ...,   0.   ,  10.201, -20.402]])
```

CALCULATING A MATRIX

$def\ b(t)$:

$ret\ = [\,S((i+1)*h, t)\ for\ i\ in\ range(N)]$

$ret[0]\ += \ d*alpha0(t)/h/h$

$ret[-1]\ += \ d*alphaL(t)/h/h$

$return\ ret$

$def\ F(w,t)$:

$return\ (np.dot(A,w)) + b(t)$

CALCULATING THE RIGHT SIDE OF W’

$$w0 \ = \ np.array([\, beta(\, (i+1) * h\,)\, for\ i\ in\ range(N)\,])$$

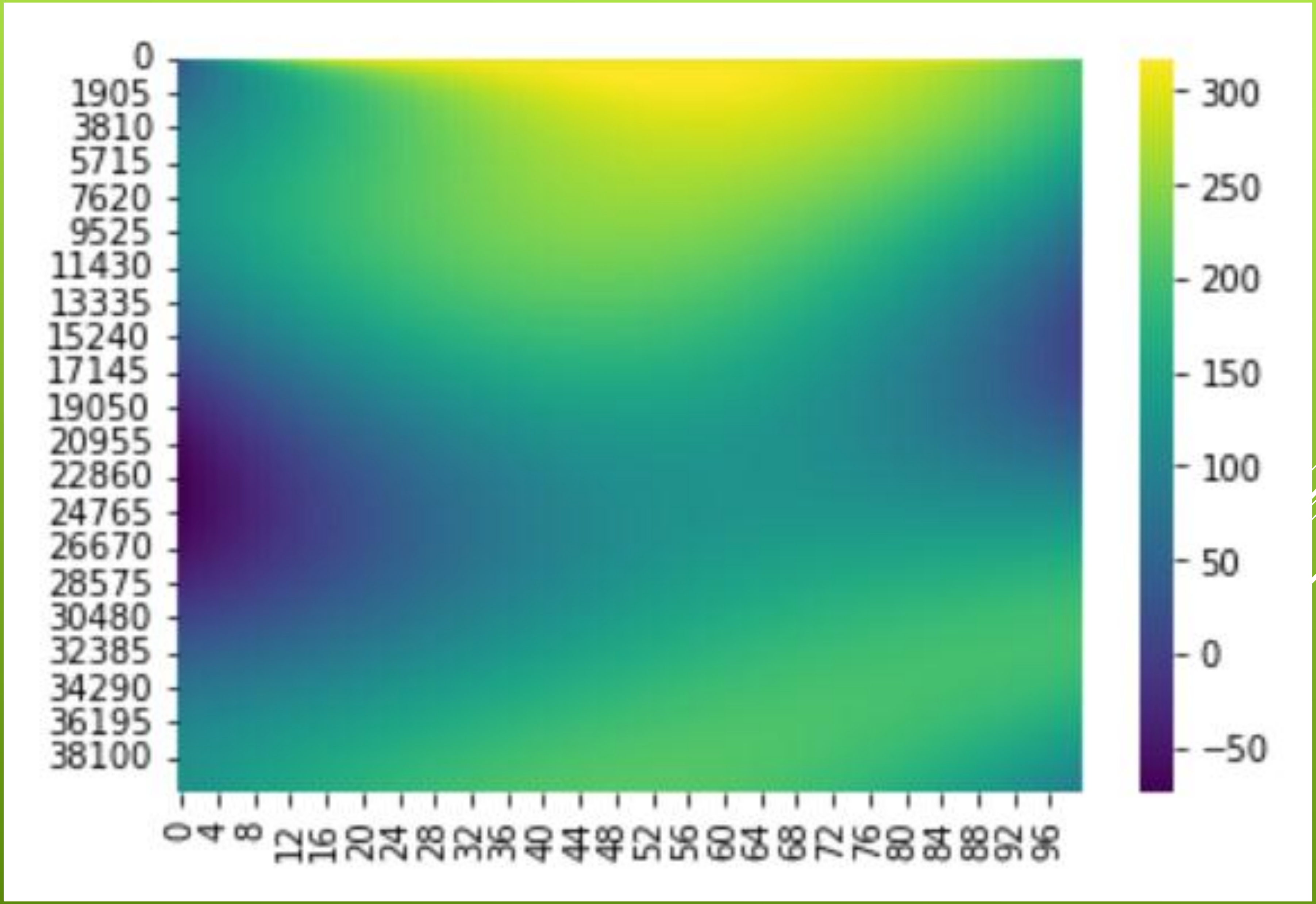
$$X \ = \ np.zeros((steps+1, N))$$

$$X[0] \ = \ w0$$

$$for\ ts\ in\ range(steps):$$

$$\quad X[ts+1] \ = \ dt * F(X[ts], ts * dt) \ + \ X[ts]$$

EXPLICIT EULER


0
1905
3810
5715
7620
9525
11430
13335
15240
17145
19050
20955
22860
24765
26670
28575
30480
32385
34290
36195
38100
0 4 8 12 16 20 24 28 32 36 40 44 48 52 56 60 64 68 72 76 80 84 88 92 96
300
250
200
150
100
50
0
−50

HEATMAP

$$A_r = U_r^T . A . U$$
$$\alpha_0 = U_r^T w_0$$
$$b_r = U_r^T . b$$
$$\alpha' = A_r . \alpha + b_r$$

REDUCE EQUATION

$$Source\_m = np.array([b(ts * dt)\ for\ ts\ in\ range(steps + 1)])$$

```
array([[2.05320898e+02, 1.31570385e+00, 1.33091409e+00, ...,
        1.33091409e+00, 1.31570385e+00, 2.04150090e+03],
       [2.05505310e+02, 1.29609601e+00, 1.31130625e+00, ...,
        1.31130625e+00, 1.29609601e+00, 2.04148127e+03],
       [2.05690476e+02, 1.27724231e+00, 1.29245255e+00, ...,
        1.29245255e+00, 1.27724231e+00, 2.04146235e+03],
       ...,
       [1.21362578e+03, 3.16952351e-01, 3.32162593e-01, ...,
        3.32162593e-01, 3.16952351e-01, 8.72381271e+02],
       [1.21359616e+03, 3.16952320e-01, 3.32162562e-01, ...,
        3.32162562e-01, 3.16952320e-01, 8.72179413e+02],
       [1.21356649e+03, 3.16952289e-01, 3.32162531e-01, ...,
        3.32162531e-01, 3.16952289e-01, 8.71977562e+02]])
```

REDUCING THE SOURCE

$$comp \ = \ 4$$

$$U, s, Vh \ = \ randomized\_svd(X.T, n\_components = comp, random\_state = 0)$$

$$Source\_m\_red \ = \ np.array([U.T \ @ \ source\_ts \ for \ source\_ts \ in \ Source\_m])$$

SINGLE VALUE DECOMPOSITION FOR THE HEAT EQ

$$\# \; Single \; Value \; Decomposition$$

$$U, s, Vh \; = \; randomized\_svd(X.T, n\_components = comp, random\_state = 0)$$

$$\# Compose \; Ar \; matrix$$

$$Ar \; = \; U.T \; @ \; A \; @ \; U$$

```
array([[-0.06445376, -0.10544729, -0.18321865, -0.08567008],
       [-0.10544729, -0.64429409,  0.05298951,  0.60002923],
       [-0.18321865,  0.05298951, -1.01145212, -1.0295829 ],
       [-0.08567008,  0.60002923, -1.0295829 , -1.72597689]])
```

REDUCING THE HEAT EQ

$def\ br(t):$

$return\ \ Source\_m\_red[t]\ \#\ Source_{m_{red}} has the\ heat\ source\ already\ reduced$

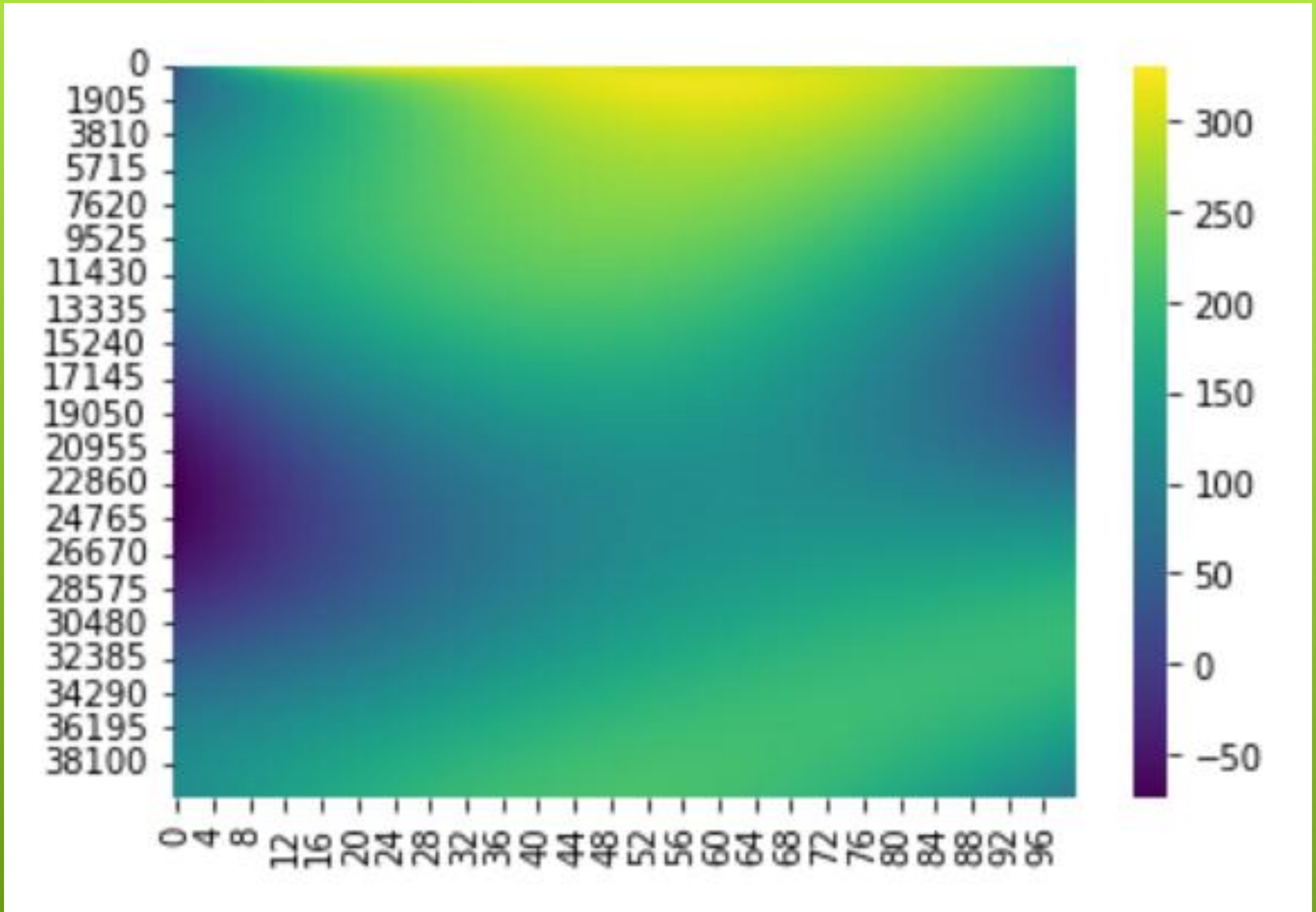
$def\ Fr(w, t):$

$return\ (np.dot(Ar, w)) + br(t)$

CALCULATING THE RIGHT SIDE OF W’

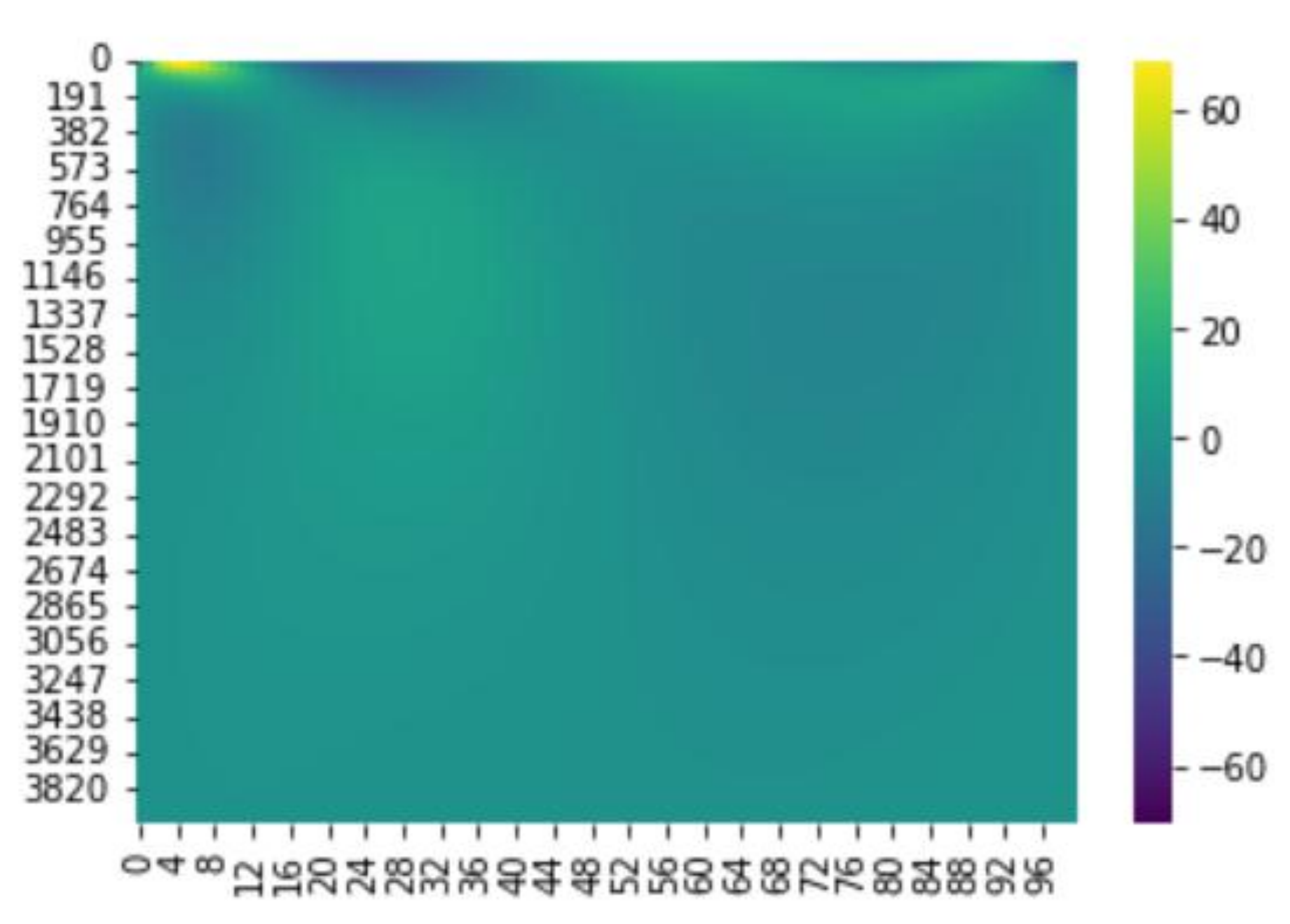
```
wr0 = U.T@w0

Xr = np.zeros((steps+1,4))
Xr[0] = wr0
for ts in range(steps):
    Xr[ts+1] = dt*Fr(Xr[ts], ts) + Xr[ts]
```

EXPLICIT EULER



HEATMAP FOR THE REDACTED EQ

- **Comparing the error**



RESULTS AND CONCLUSIONS

- **Comparing the time**

Explicit Euler

```python
%%time
w0 = np.array([ beta( (i+1)*h ) for i in range(N) ])

X = np.zeros((steps+1,N))
X[0] = w0
for ts in range(steps):
    X[ts+1] = dt*F(X[ts], ts*dt) + X[ts]
```

```
CPU times: user 1min 3s, sys: 6min 37s, total: 7min 41s
Wall time: 18.5 s
```

RESULTS AND CONCLUSIONS

- **Comparing the time**

Explicit Euler

```
%%time
# w0 = np.array([ beta( (i+1)*h ) for i in range(N) ])
wr0 = U.T@w0

Xr = np.zeros((steps+1,4))
Xr[0] = wr0
for ts in range(steps):
    Xr[ts+1] = dt*Fr(Xr[ts], ts) + Xr[ts]
```

```
CPU times: user 628 ms, sys: 45.4 ms, total: 673 ms
Wall time: 635 ms
```

RESULTS AND CONCLUSIONS

THANK YOU FOR YOUR ATTENTION