laplace_equation-relaxation

March 30, 2021

1 Laplace equation using the relaxation method

In this notebook you will see how to solve the **Laplace equation** using the **relaxation method** in `Python` programming language.

1.1 Importing the libraries

[1]:
```
import numpy as np
from pathlib import Path
import matplotlib.pyplot as pl
import matplotlib.animation as animation
```

1.2 The code

[2]:
```
Path("./results").mkdir(parents=True, exist_ok=True)
```

Geometry of the box

[3]:
```
n_x , n_y = 40 , 40
x = np.arange(n_x)
y = np.arange(n_y)
```

Initialize the array

[4]:
```
field = np.zeros((4, n_x,n_y))
```

1.2.1 Set the boundary conditions

Example 0: A line

[5]:
```
def in_cond0(field):
    field[15:25,20] = 1
    return field
```

Example 1: A square

[6]:
```
def in_cond1(field):
    field[15:26,15] = 1
    field[15,15:26] = 1
```

```
    field[15:26,25] = 1
    field[25,15:26] = 1
    return field
```

Example 2: A triangle

[7]:
```
def in_cond2(field):
    field[15:25,15] = 1
    field[15,15:25] = 1
    for ij in range(11):
        field[15+ij,25-ij]=1
    return field
```

Example 3: Open triangle

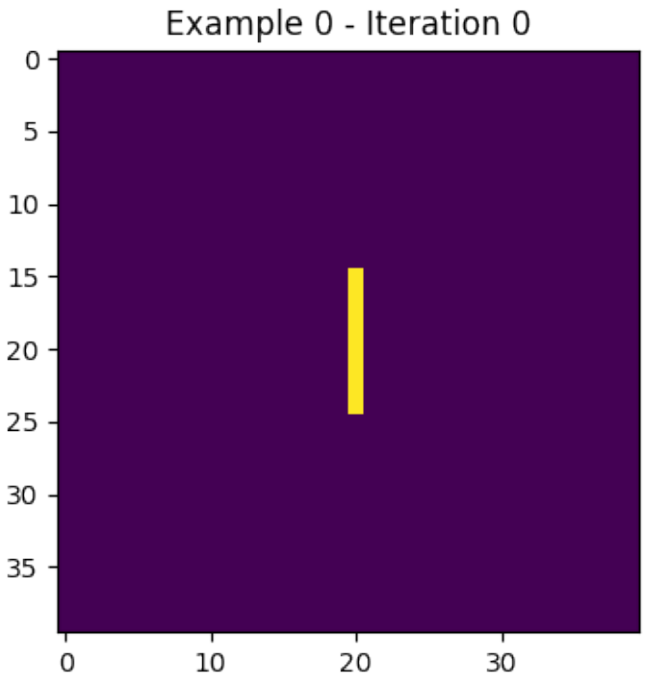
[8]:
```
def in_cond3(field):
    field[15:25,15] = 1
    field[15,15:25] = 1
    for ij in range(8):
        field[15+ij,25-ij]=1
    return field
```

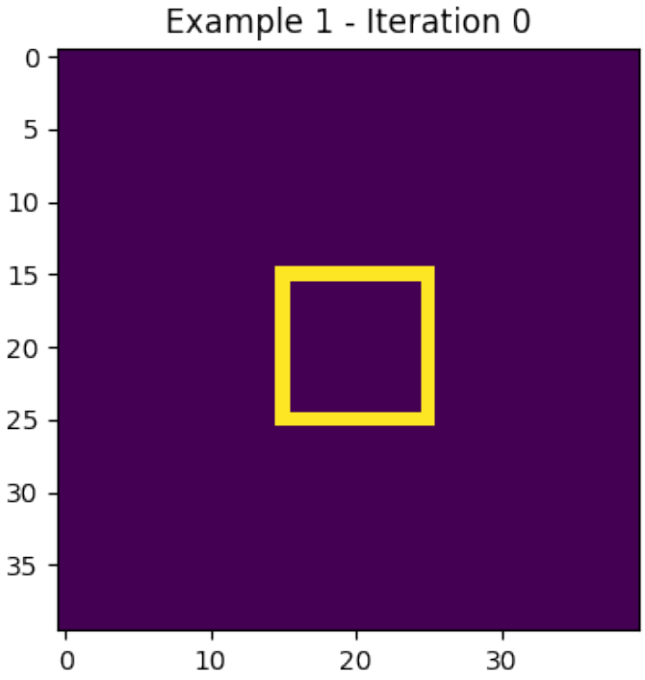
1.2.2 Visualizing the examples

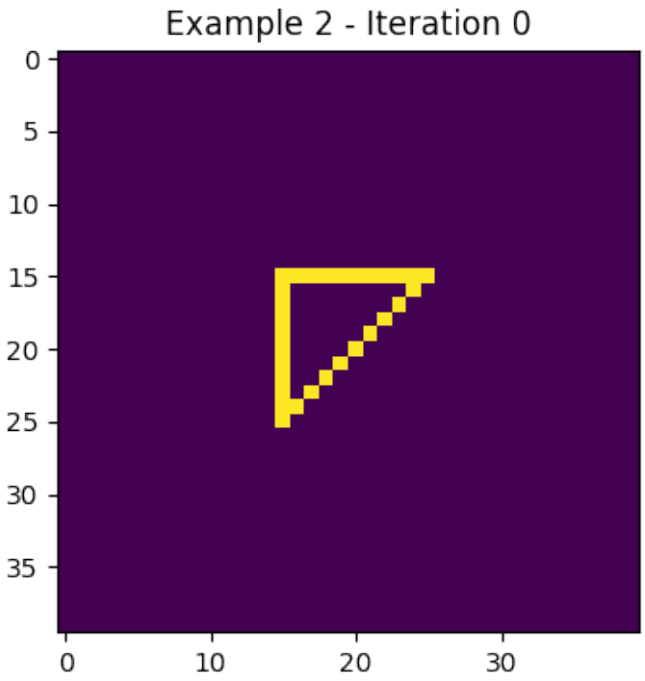
Plot the initial conditions

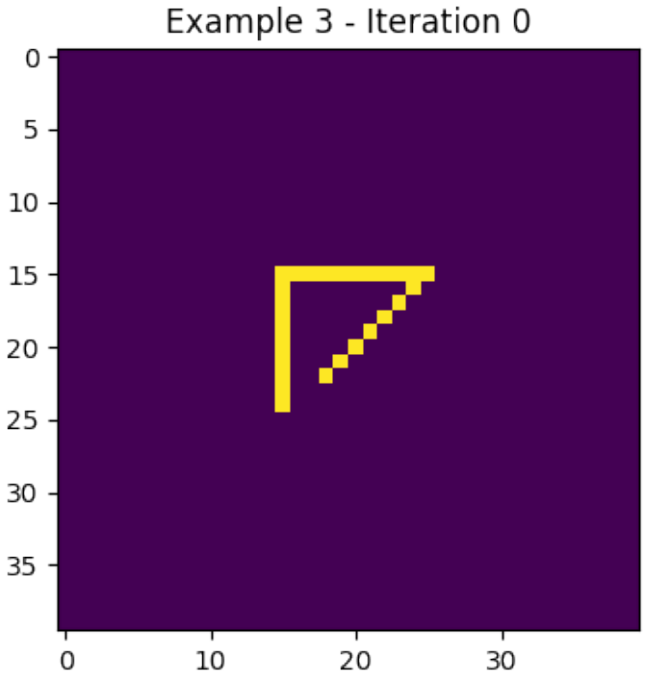
[9]:
```
field = np.zeros((4, n_x,n_y))
field[0] = in_cond0(field[0])
field[1] = in_cond1(field[1])
field[2] = in_cond2(field[2])
field[3] = in_cond3(field[3])
```

[10]:
```
for i in range(4):
    pl.figure(dpi = 100)
    pl.imshow(field[i])
    pl.title(f'Example {i} - Iteration 0')
    pl.savefig(f'results/cond{i}-It_0.pdf')
    pl.show()
```


Example 0 - Iteration 0
0
5
10
15
20
25
30
35
0
10
20
30

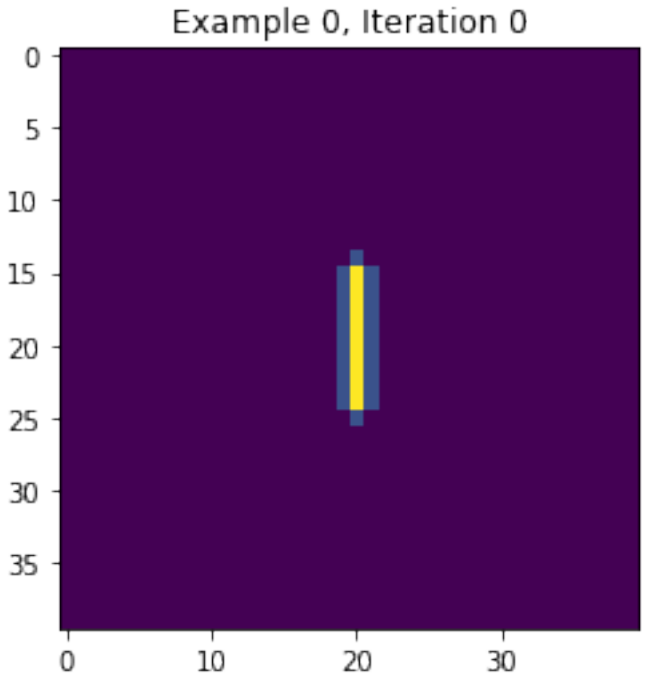

Example 1 - Iteration 0
0
5
10
15
20
25
30
35
0
10
20
30

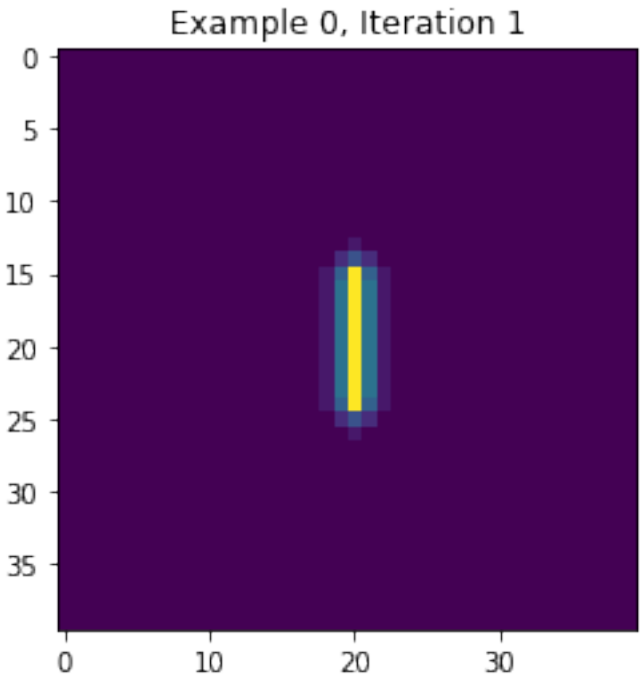


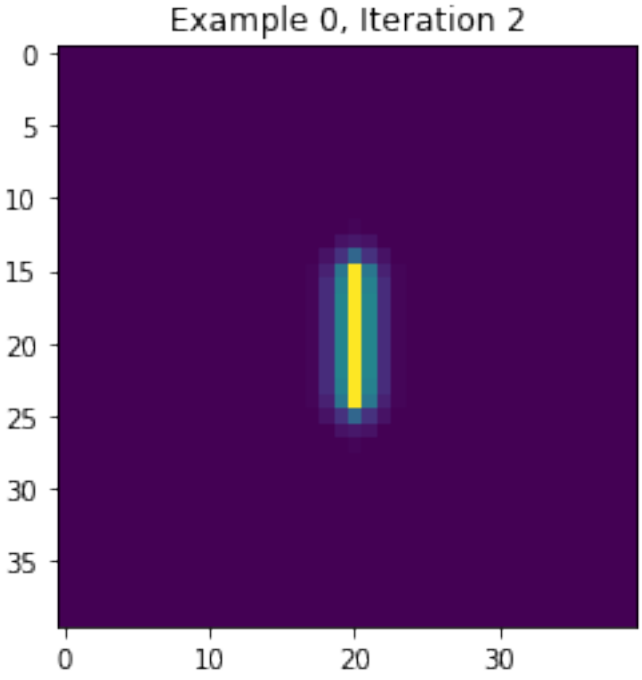


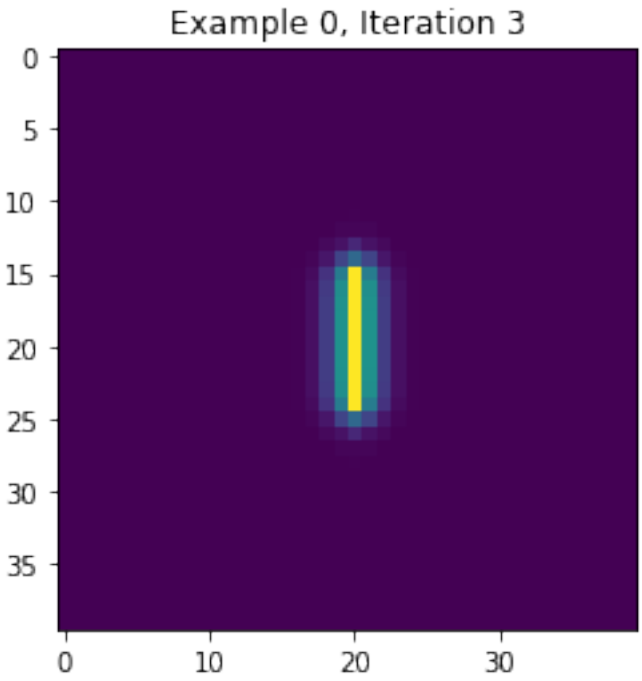
Plots until iteration 10

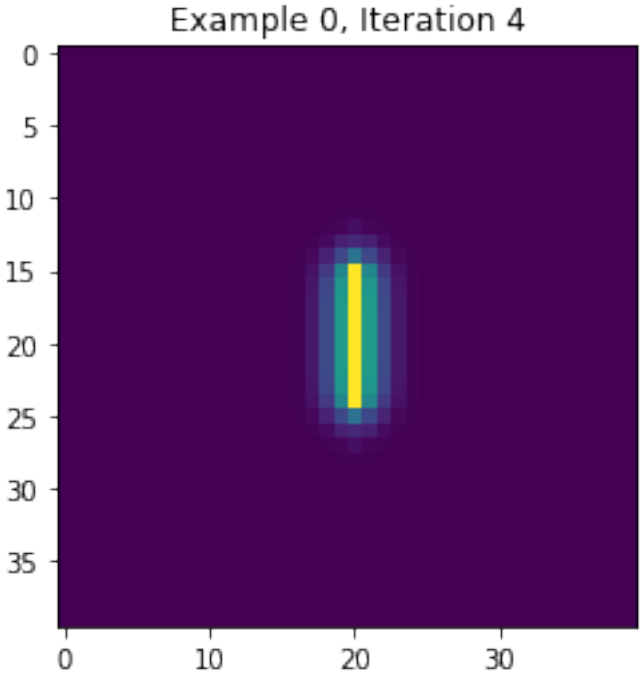
[11]:
```
# Iterate N times, plot each iteration
for i in range(4):
    for nit in range(10):
        field[i,1:-1,1:-1] = 0.25 * (field[i,0:-2,1:-1] + field[i,2:,1:-1] +␣
↪field[i,1:-1,0:-2] + field[i,1:-1,2:])
        # Re-inforce initial conditions
        field[0] = in_cond0(field[0])
        field[1] = in_cond1(field[1])
        field[2] = in_cond2(field[2])
        field[3] = in_cond3(field[3])

        # plot
        pl.imshow(field[i])
        pl.title(f'Example {i}, Iteration {nit}')
        name = f"results/example{i}-It_" + str(nit+1) + ".pdf"
        pl.savefig(name,dpi=100)
        pl.show()
```

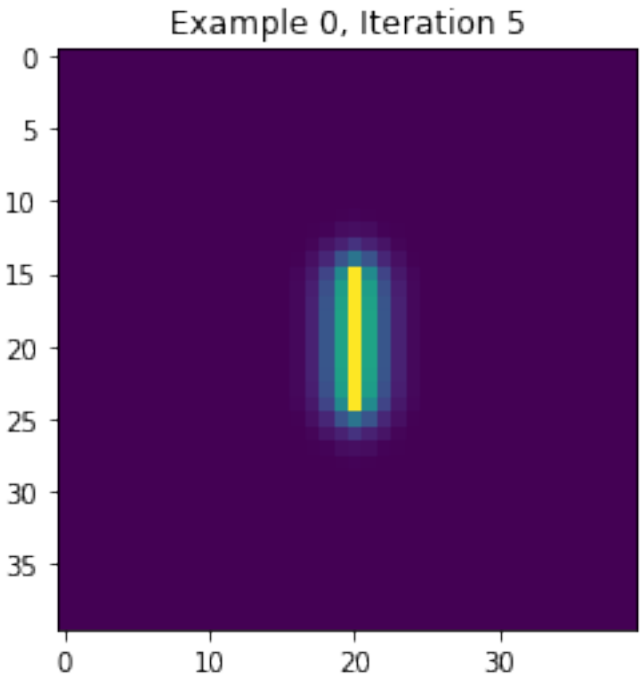


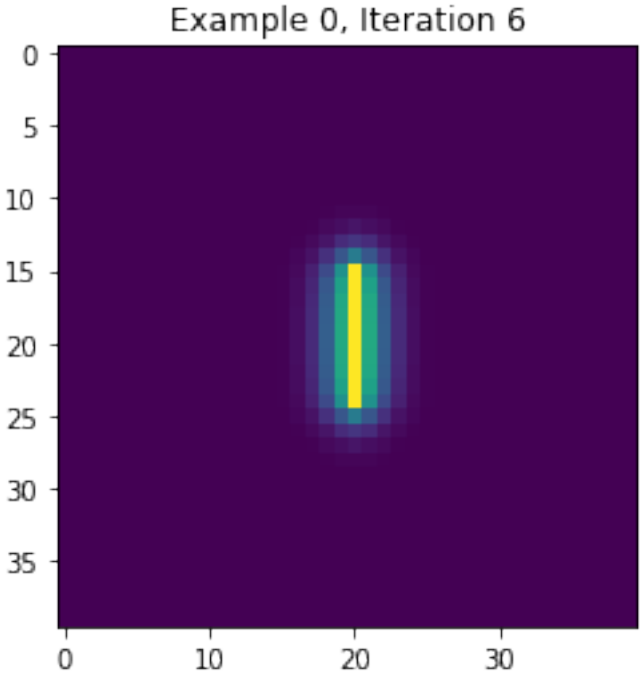

Example 0, Iteration 1
0
5
10
15
20
25
30
35
0
10
20
30

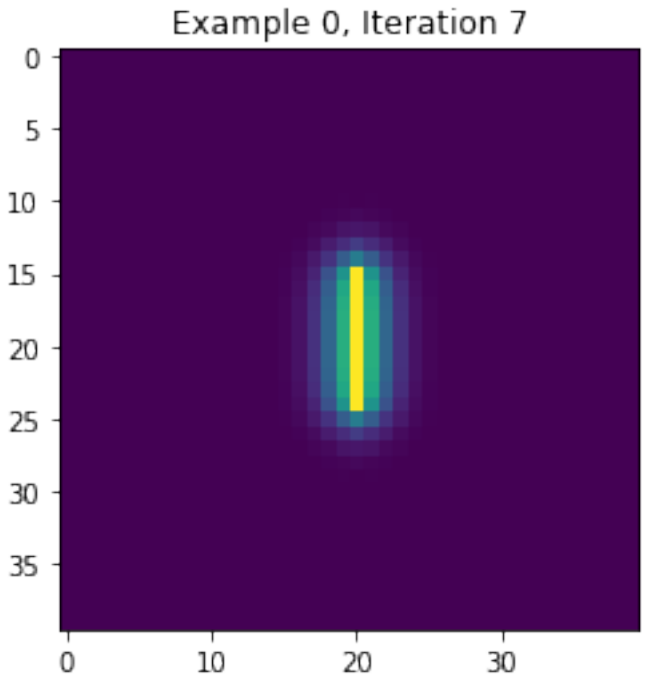

Example 0, Iteration 2
0
5
10
15
20
25
30
35
0
10
20
30

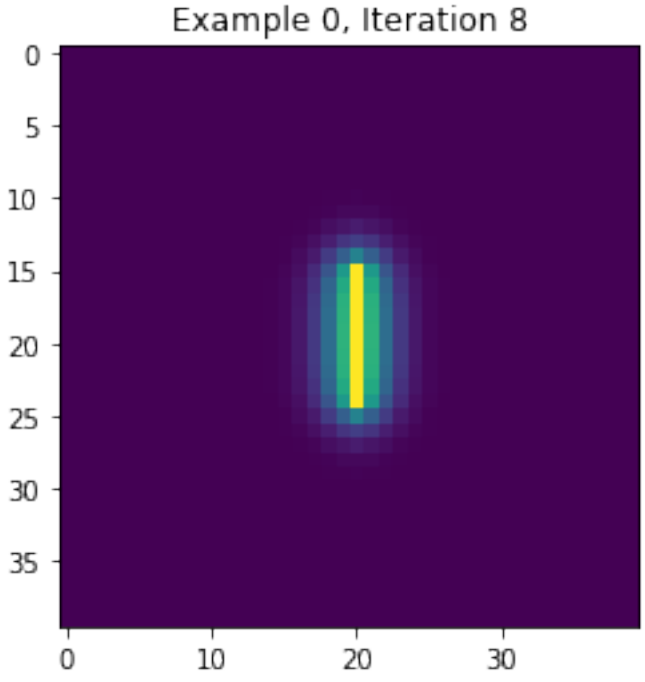

Example 0, Iteration 3
0
5
10
15
20
25
30
35
0
10
20
30

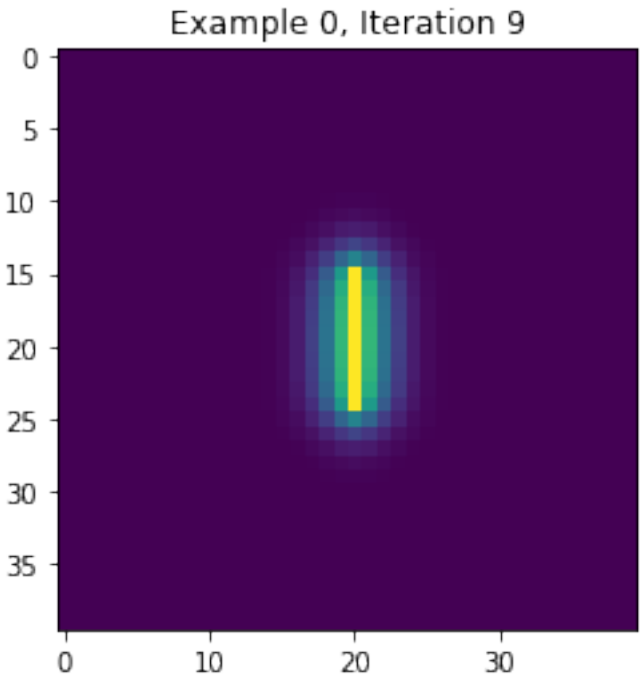

Example 0, Iteration 4
0
5
10
15
20
25
30
35
0
10
20
30

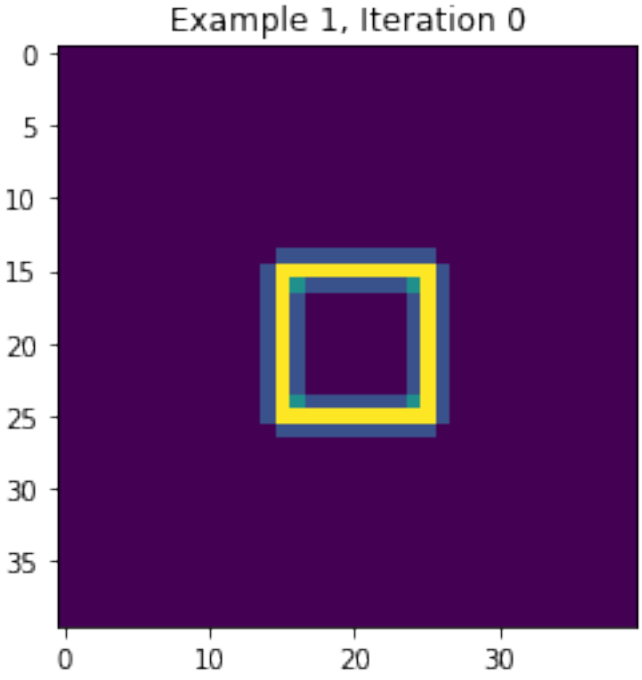

Example 0, Iteration 5
0
5
10
15
20
25
30
35
0
10
20
30

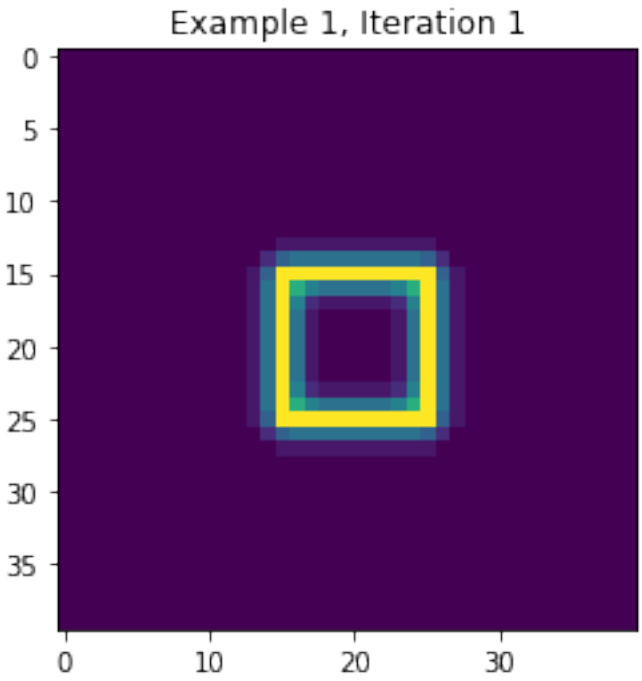

Example 0, Iteration 6
0
5
10
15
20
25
30
35
0
10
20
30

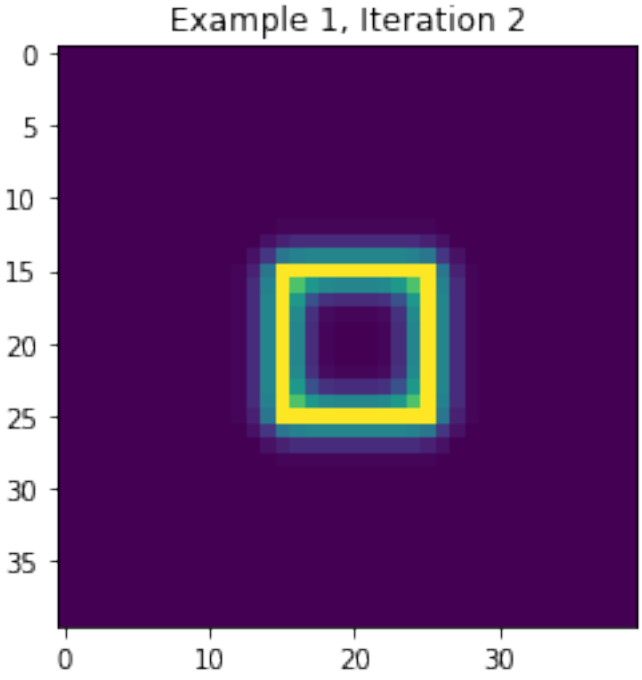

Example 0, Iteration 7
0
5
10
15
20
25
30
35
0
10
20
30

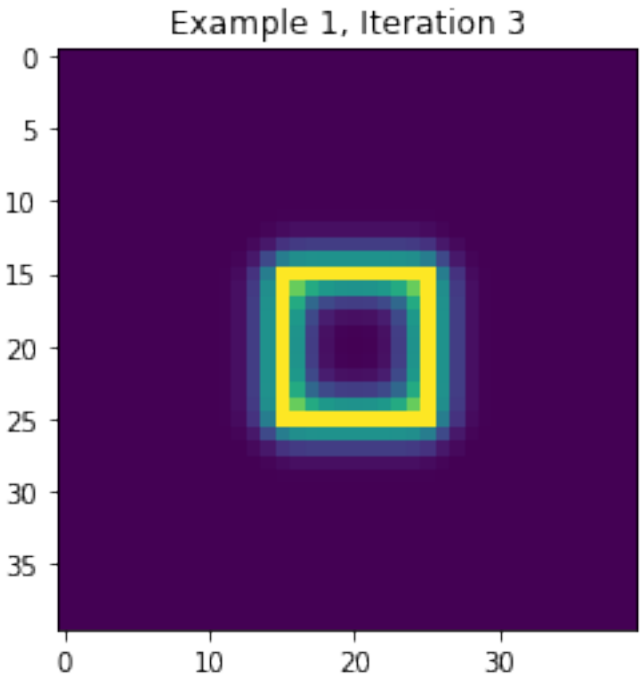

Example 0, Iteration 8
0
5
10
15
20
25
30
35
0
10
20
30

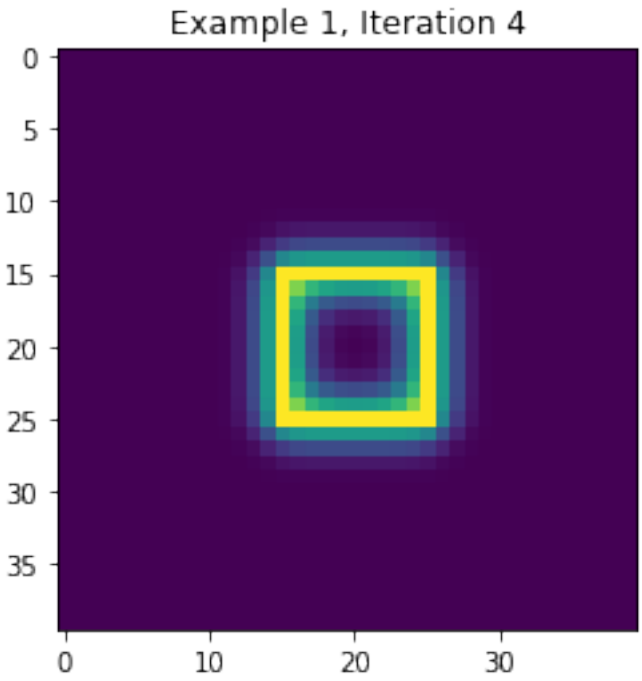

Example 0, Iteration 9
0
5
10
15
20
25
30
35
0
10
20
30

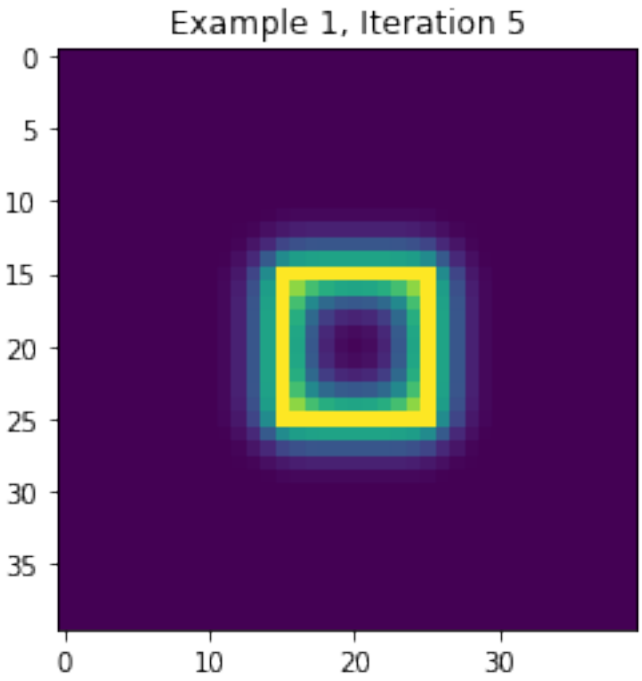

Example 1, Iteration 0
0
5
10
15
20
25
30
35
0
10
20
30

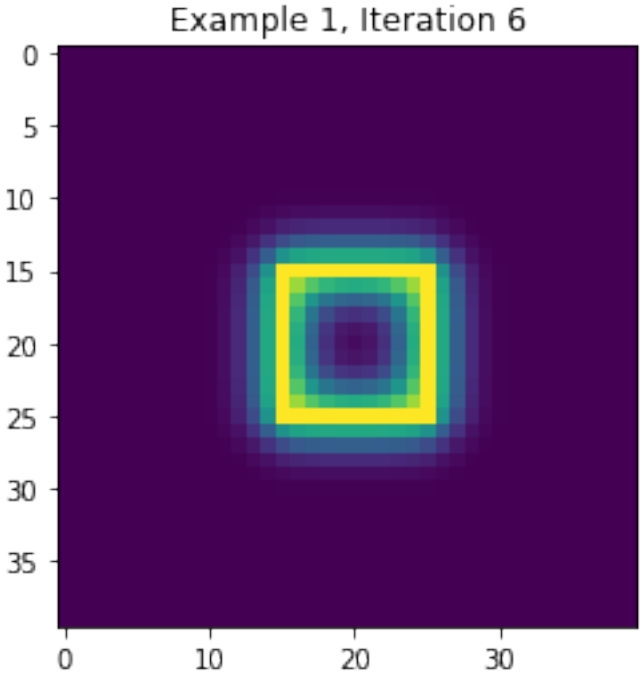

Example 1, Iteration 1
0
5
10
15
20
25
30
35
0
10
20
30

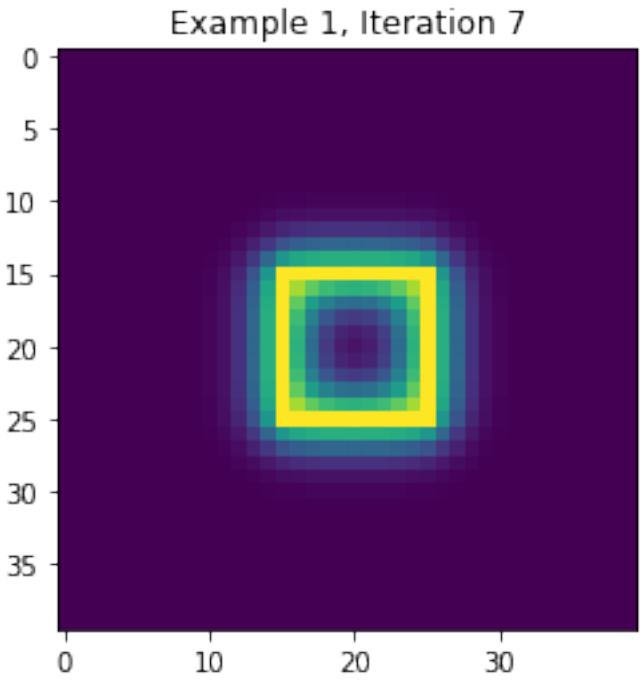

Example 1, Iteration 2
0
5
10
15
20
25
30
35
0
10
20
30

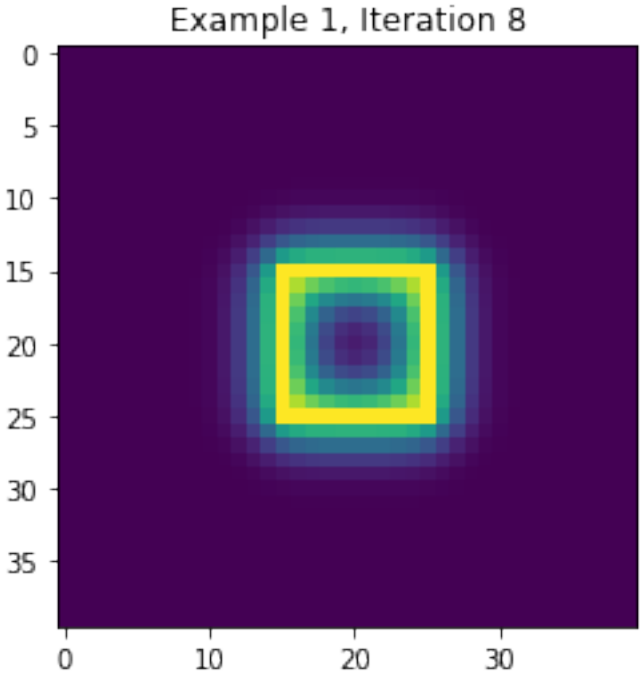

Example 1, Iteration 3
0
5
10
15
20
25
30
35
0
10
20
30

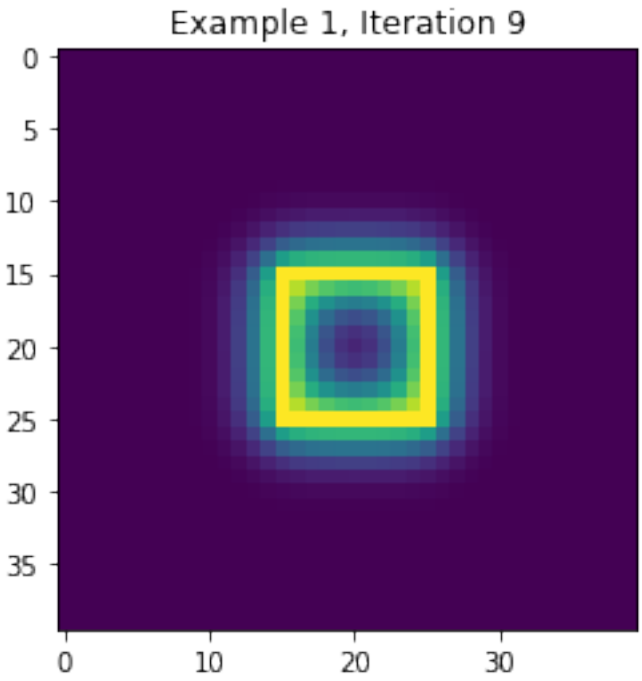

Example 1, Iteration 4
0
5
10
15
20
25
30
35
0
10
20
30

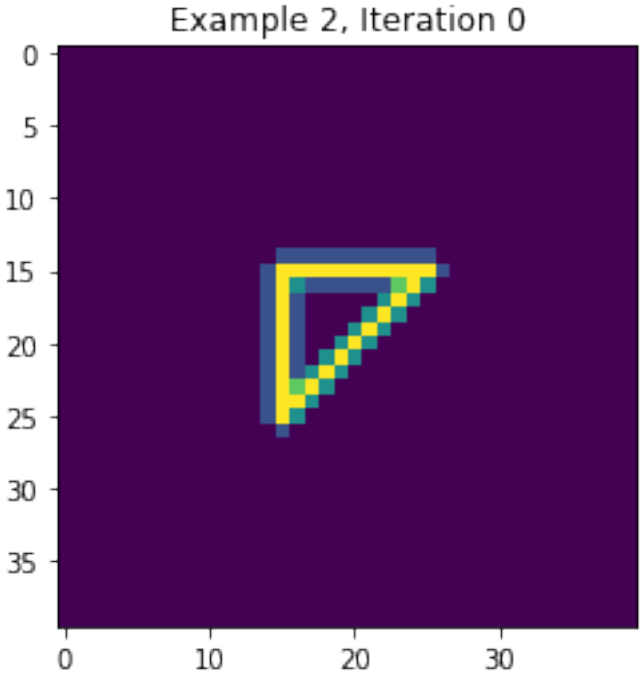

Example 1, Iteration 5
0
5
10
15
20
25
30
35
0
10
20
30

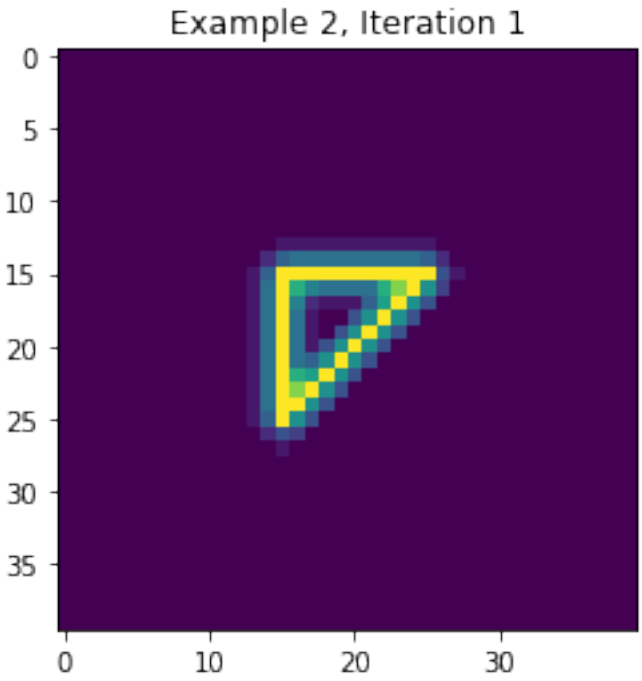

Example 1, Iteration 6
0
5
10
15
20
25
30
35
0
10
20
30

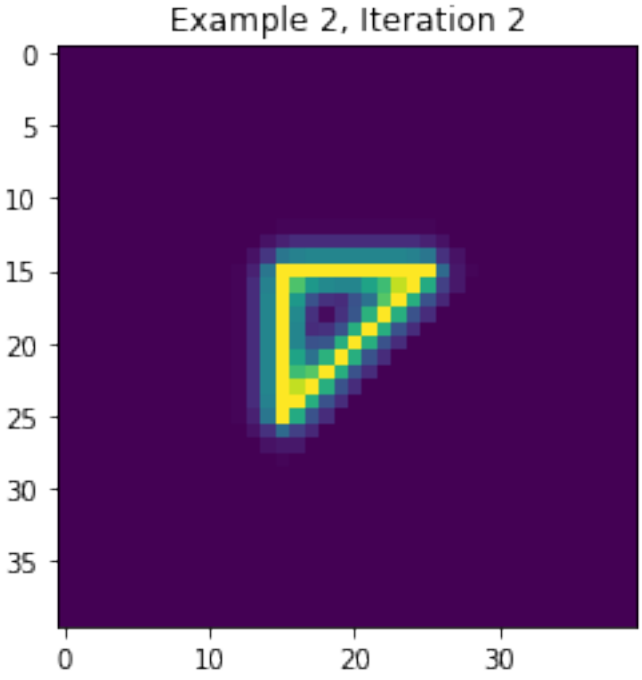

Example 1, Iteration 7
0
5
10
15
20
25
30
35
0
10
20
30

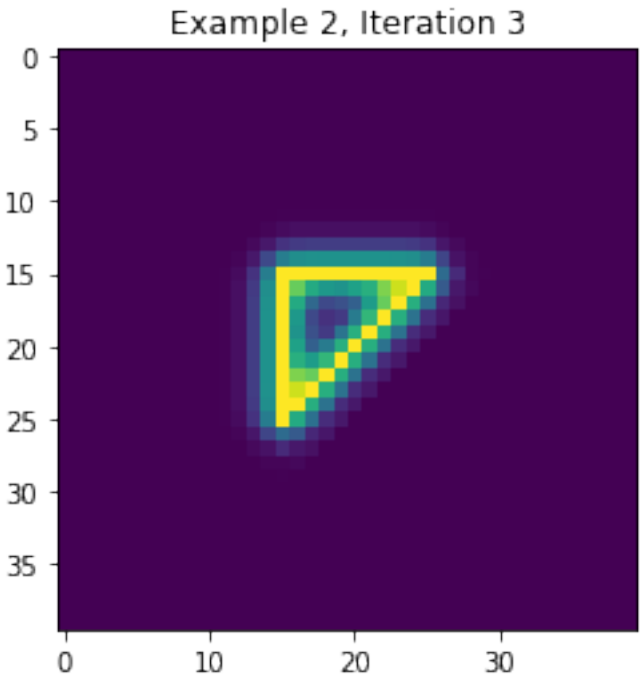

Example 1, Iteration 8
0
5
10
15
20
25
30
35
0
10
20
30

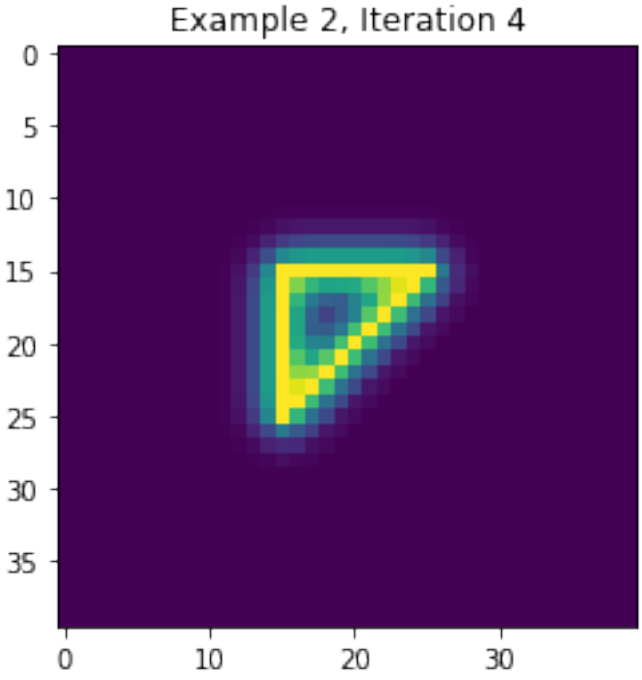

Example 1, Iteration 9
0
5
10
15
20
25
30
35
0
10
20
30

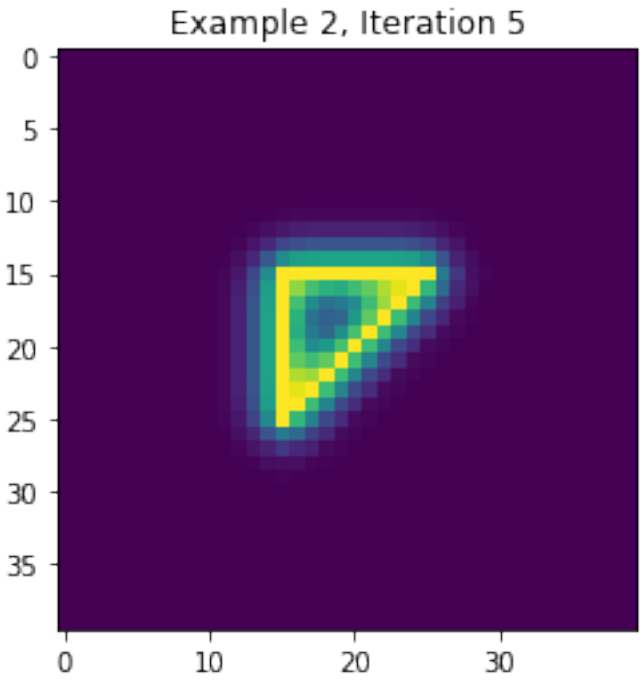

Example 2, Iteration 0
0
5
10
15
20
25
30
35
0
10
20
30

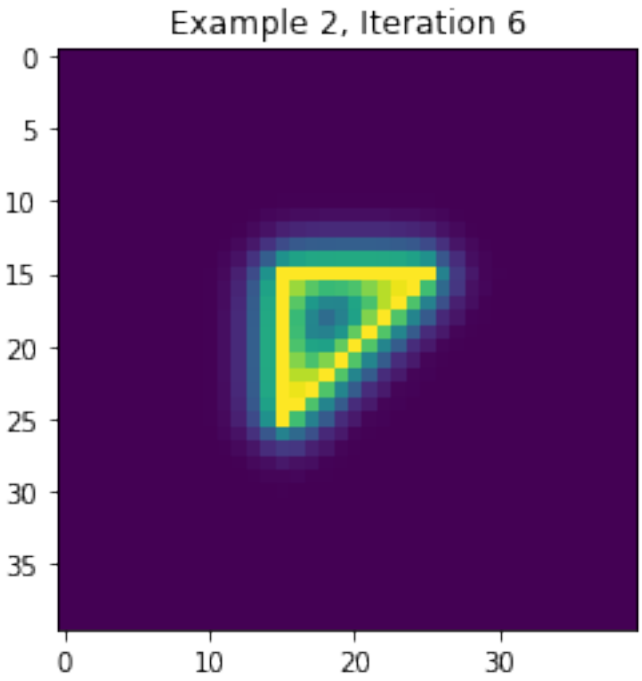

Example 2, Iteration 1
0
5
10
15
20
25
30
35
0
10
20
30

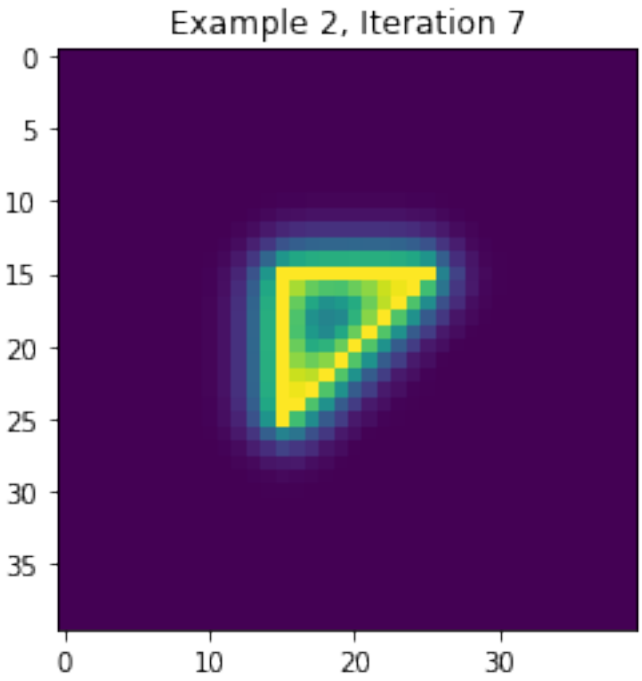

Example 2, Iteration 2
0
5
10
15
20
25
30
35
0
10
20
30

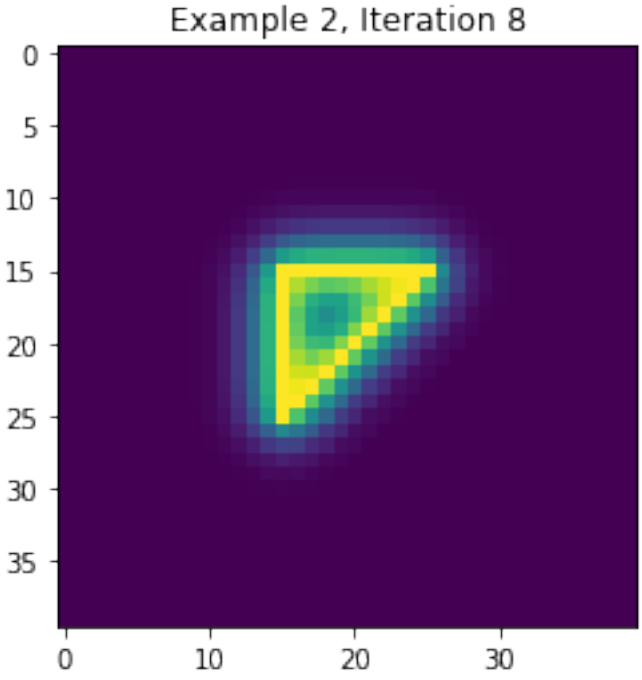

Example 2, Iteration 3
0
5
10
15
20
25
30
35
0
10
20
30

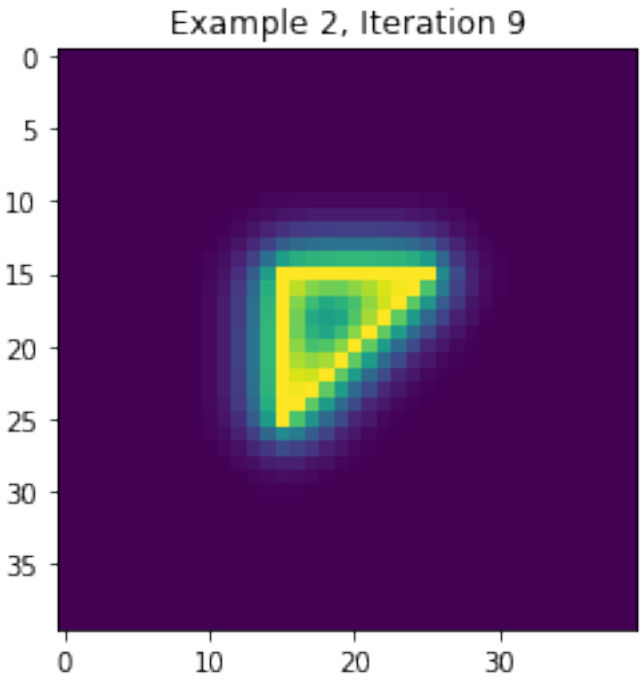

Example 2, Iteration 4
0
5
10
15
20
25
30
35
0
10
20
30

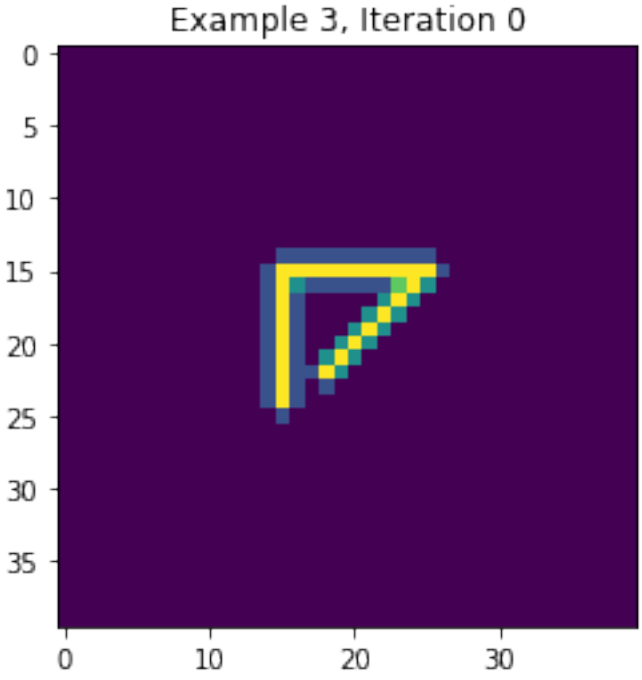

Example 2, Iteration 5
0
5
10
15
20
25
30
35
0
10
20
30

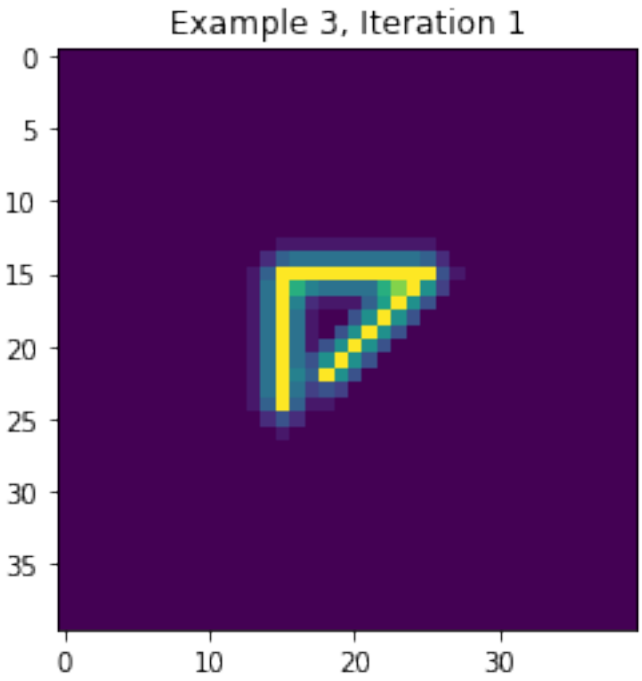

Example 2, Iteration 6
0
5
10
15
20
25
30
35
0
10
20
30

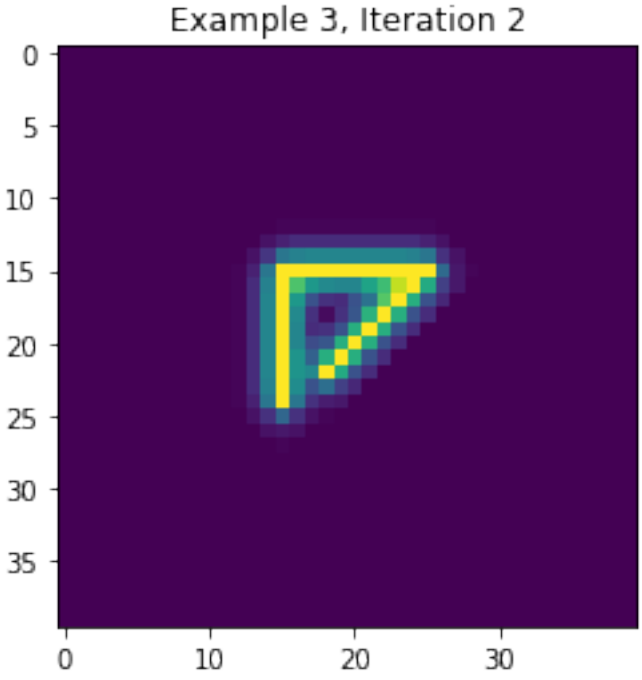

Example 2, Iteration 7
0
5
10
15
20
25
30
35
0
10
20
30

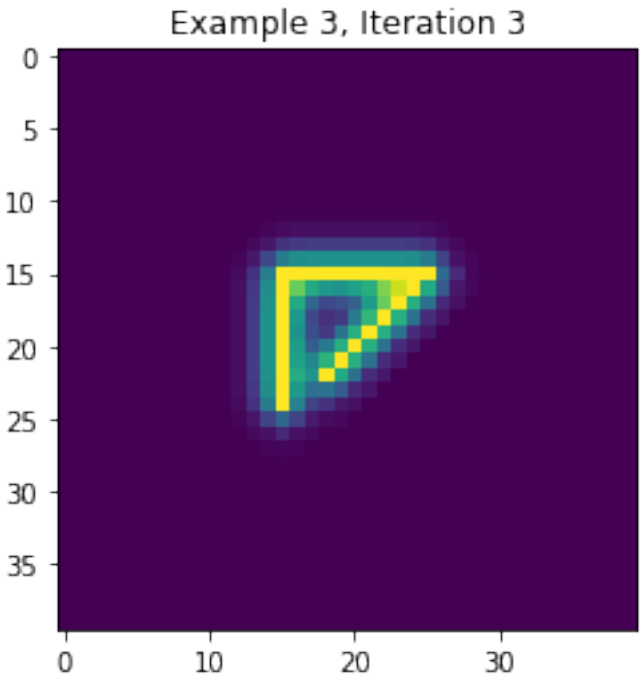

Example 2, Iteration 8
0
5
10
15
20
25
30
35
0
10
20
30

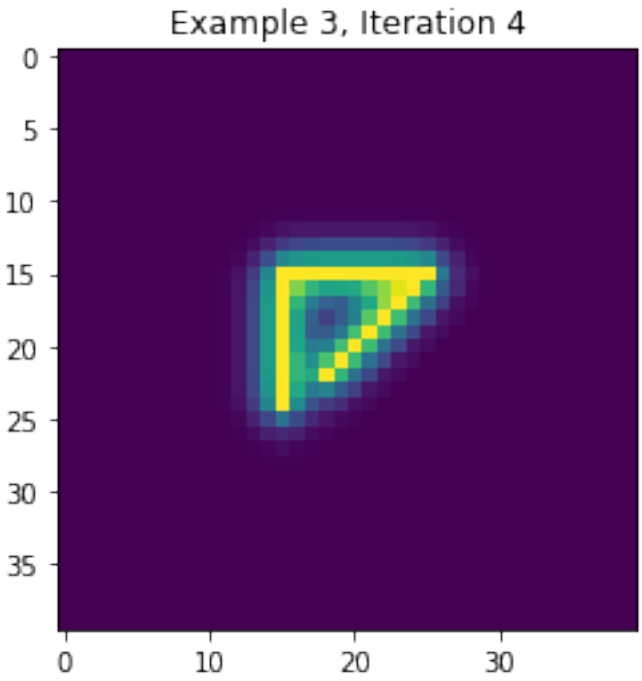

Example 2, Iteration 9
0
5
10
15
20
25
30
35
0
10
20
30

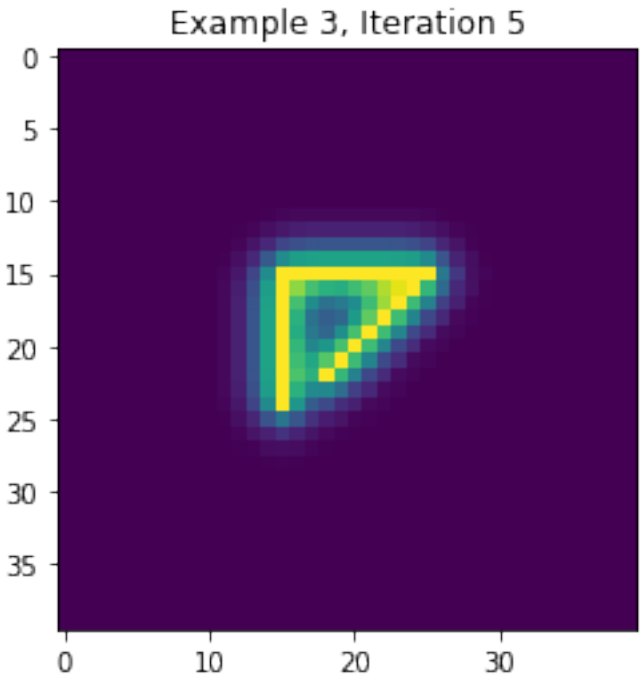

Example 3, Iteration 0
0
5
10
15
20
25
30
35
0
10
20
30

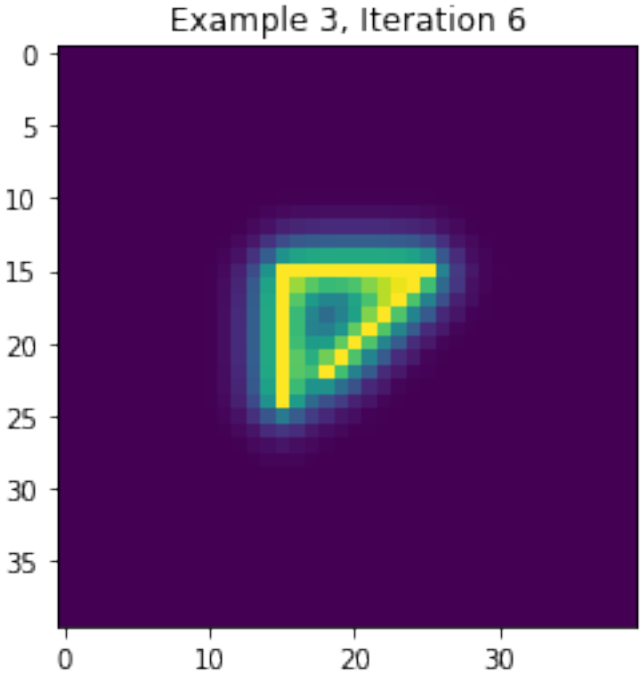

Example 3, Iteration 1
0
5
10
15
20
25
30
35
0
10
20
30

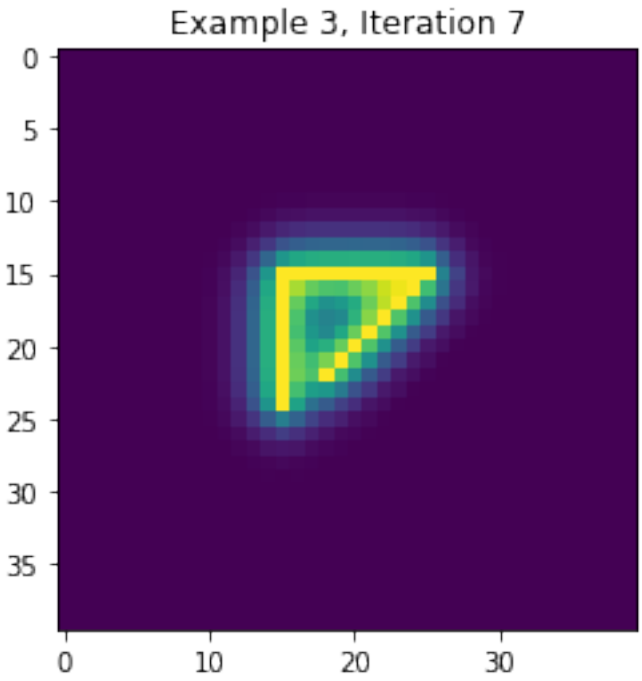

Example 3, Iteration 2
0
5
10
15
20
25
30
35
0
10
20
30

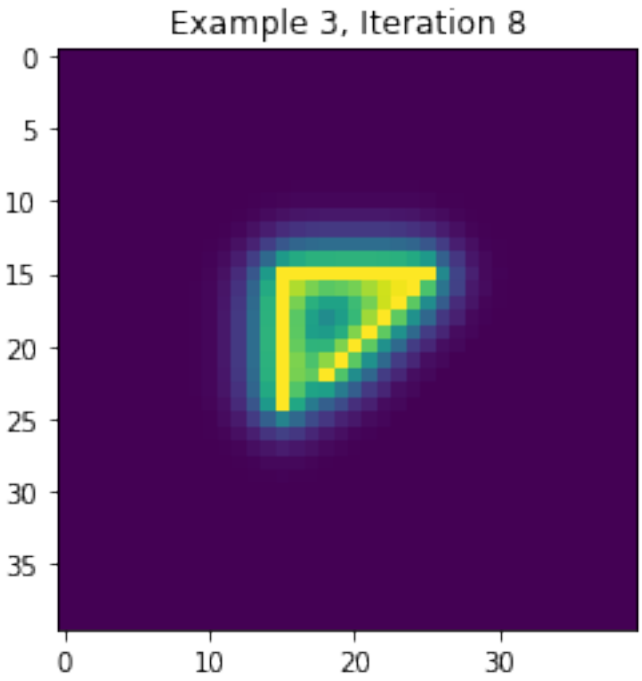

Example 3, Iteration 3
0
5
10
15
20
25
30
35
0
10
20
30

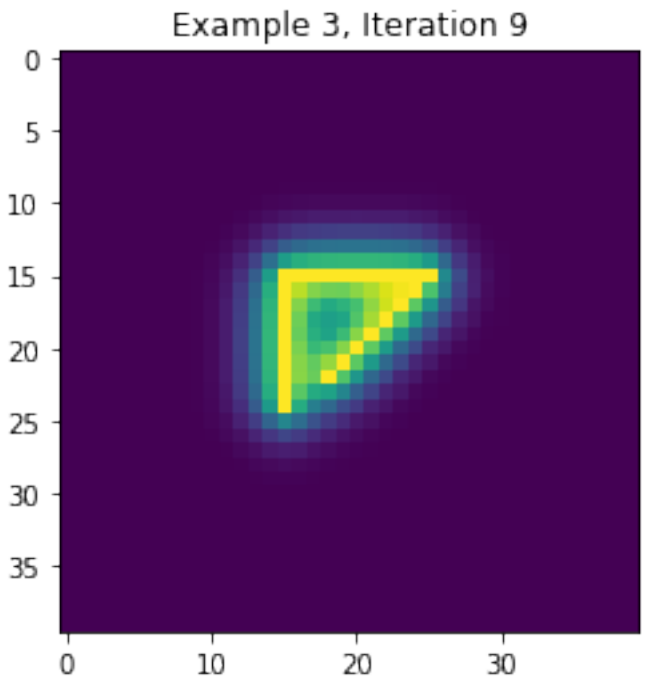

Example 3, Iteration 4
0
5
10
15
20
25
30
35
0
10
20
30


Example 3, Iteration 5
0
5
10
15
20
25
30
35
0
10
20
30


Example 3, Iteration 6
0
5
10
15
20
25
30
35
0
10
20
30

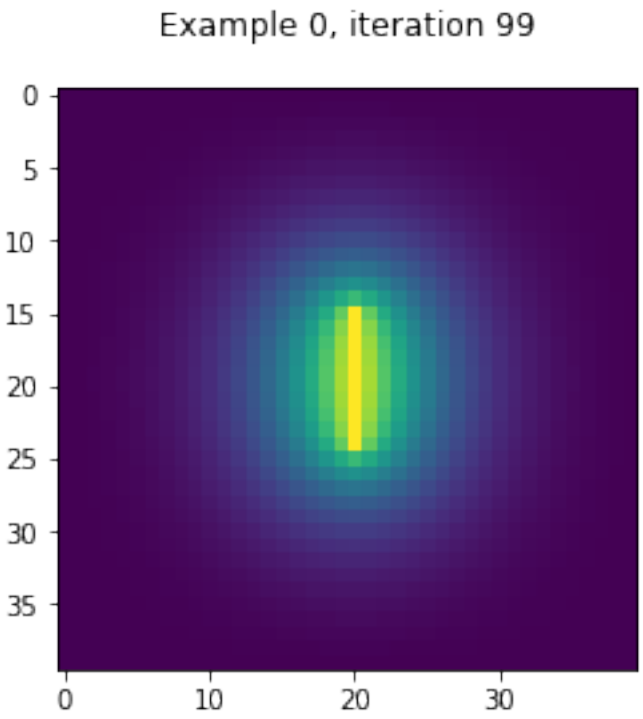

Example 3, Iteration 7
0
5
10
15
20
25
30
35
0
10
20
30

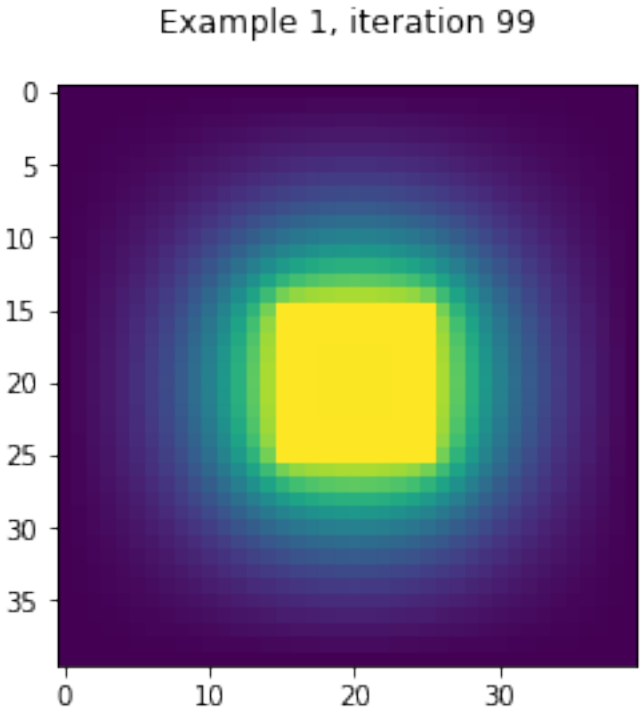

Example 3, Iteration 8
0
5
10
15
20
25
30
35
0
10
20
30

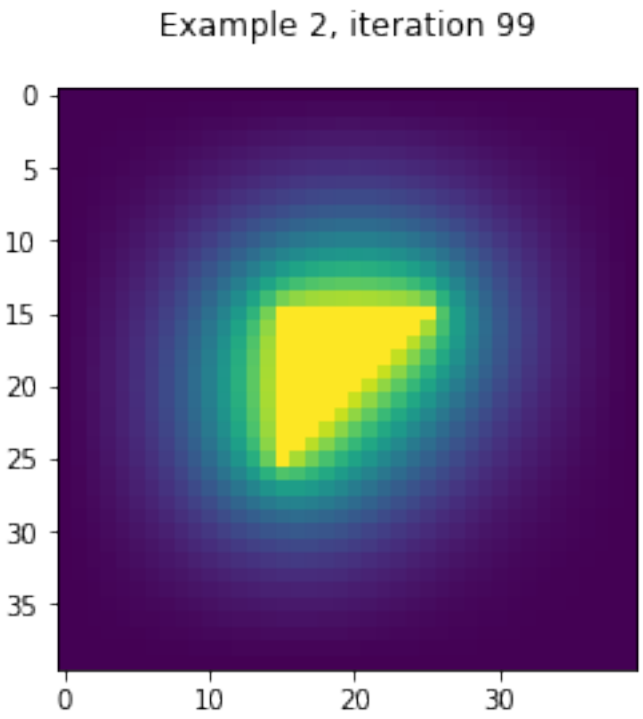


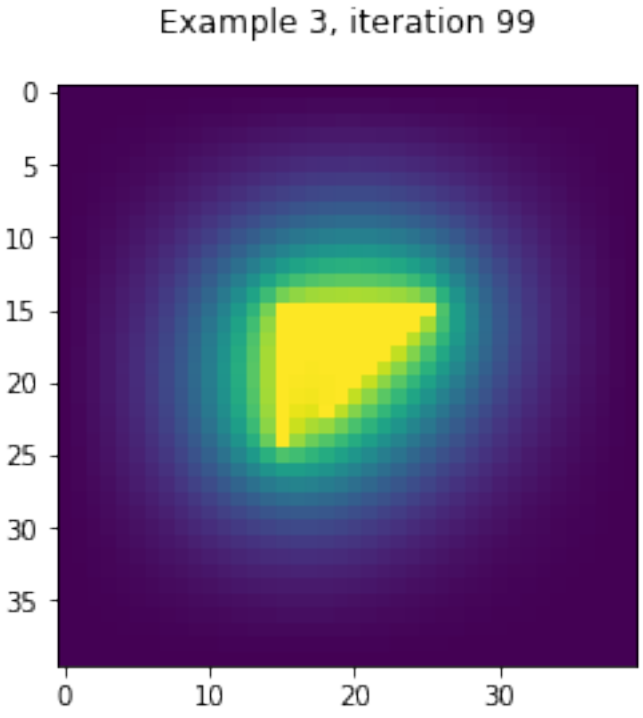
Making a movie (until iteration 100) for each example

[12]:
```
# Iterate N times, plot each iteration
for i in range(4):
    fig, ax = pl.subplots()
    ims = []
    for nit in range(100):
        field[i,1:-1,1:-1] = 0.25 * (field[i,0:-2,1:-1] + field[i,2:,1:-1] +␣
↪field[i,1:-1,0:-2] + field[i,1:-1,2:])
        # Re-inforce initial conditions
        field[0] = in_cond0(field[0])
        field[1] = in_cond1(field[1])
        field[2] = in_cond2(field[2])
        field[3] = in_cond3(field[3])

        # plot
        im = pl.imshow(field[i], animated=True)
        title = ax.text(0.5,1.09,f"Example {i}, iteration {nit}", size=pl.
↪rcParams["axes.titlesize"], ha="center", transform=ax.transAxes, )
        ims.append([im, title])
    ani = animation.ArtistAnimation(fig, ims, interval=100, blit=False)
    ani.save(f"results/example{i}-movie_from_figs.mp4")
```


Example 0, iteration 99
0
5
10
15
20
25
30
35
0
10
20
30


Example 1, iteration 99
0
5
10
15
20
25
30
35
0
10
20
30


Example 2, iteration 99
0
5
10
15
20
25
30
35
0
10
20
30


Example 3, iteration 99
0
5
10
15
20
25
30
35
0
10
20
30