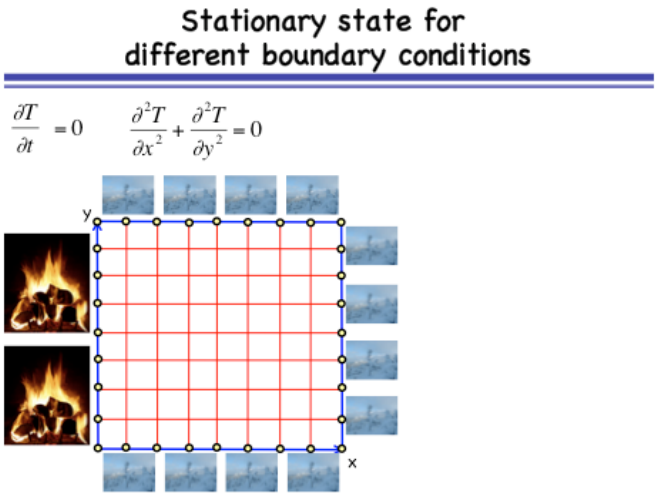
Stationary State of the Heat Equation in 2D

In this section, we want to find the stationary temperature distribution in a 2D square with the following boundary conditions:



Imagine you are renting a room of size L×L that has three cold outside walls and one wall that is heated by your neighbor. The outside temperature is kept fixed at 0°C and inside wall at 100°C. Divide the square into N×N segments. N=51 is a good resolution to start. Solving this problem numerically requires a 2D array $T_{i,j}$ or in Python language an array `T[i,j]`.

Convert the analytic form of the 2D heat equation given in the lecture slide above into the discretized form using $T_{i,j}$. Derive the 2D version of equation (*) given above.

$$T_{i,j}^{new} = \ldots \left[\ldots \ldots \ldots \ldots \right] \qquad (**)$$

(1) Write a code to set the boundary value of the 2D array `T(i,j)` .

(2) Similar to part 1 introduce a main, outer loop over *iterations*.

(3) Inside the main loop, introduce a nested pair of inner loops over i and j that cover all interior points of the square above. At each point, you should set the value of `Tnew(i,j)` according to equation (**).

(4) As in part 1, copy the data back and forth between T and T^{new}.

(5) To plot the temperature distribution in 3D, at the beginning of your code please add

```
from mpl_toolkits.mplot3d import Axes3D
from matplotlib import cm
X = np.linspace(0, L, n) # where n is number of segments
Y = np.linspace(0, L, n)
X, Y = np.meshgrid(X, Y)
fig = plt.figure(dpi=200)
```

Inside the outer loop please add

```
fig.clear()
ax = fig.gca(projection='3d')
ax.plot_surface(X, Y, Z, cmap=cm.coolwarm, antialiased=False)
plt.draw()
plt.pause(0.05)
```

(6) *Optional:* Instead of using a temporary array T^{new}, change the update formula (**) so that you overwrite each `T[i,j]` element immediately:

$$T_{i,j} = \ldots \left[\ldots \ldots \ldots \ldots \right]$$

Answer two questions:

a) Does the code now converge faster or slower to the stationary temperature distribution?

b) Does it make a difference in which order you update the elements? You could imagine interchanging the inside *i* and *j* loops, or running either one backwards.

1D and 2D Animations Captured in mp4 Movie Files