Please hand in a PDF writeup for 1b and 2. Also submit commented code for 1a.

1. Solving the steady-state heat equation

The **steadystate_heat.py** code on the course website solves the steady-state heat equation with given boundary conditions that we defined in class.

a. Please add comments to the program to summarize how it works. In particular:
   - What are the **BOUNDARY_** variables doing?
   - How are they related to the **boundaries** array?
   - What does **scipy.sparse.lil_matrix** do?
   - What are the **neighbor_deltas** for?
   - How the heck does the vectorized building code work?
   - What does the line **A = A.tocsr()** do?

b. For what values of **n** is it advantageous to compute the ‘dense’ solution as opposed to the ‘sparse’ one? Is there any problem size for which the ‘loop’ building method outperforms the ‘vectorized’ one? Why is the sparse solver so much faster for large **n**?

2. Transient heat equation

The **transient_heat.py** code on the course website solves the transient (time-varying) heat equation.

a. Based upon the code, what are the boundary conditions for this system? What temperature are we assuming (either implicitly or explicitly) lies along the border of the plate?

b. Play with the simulation. At which value of **α** does the forward Euler solver blow up with the default value of **n = 64**?

c. Can you make the backward solver blow up?

d. Explain how modifying **n** (by editing the code) affects the behavior of the forward solver. Does the it get more or less stable (in terms of the maximum allowable value of **α**) as **n** increases?