

## 1 Graph Layouts

- a) Which layouts do you think would fit best to which kind of graphs?
- b) Create a random graph containing about 50 nodes. Assign random values  $\in [0, 1]$  to the edges as 'radius' attributes.
- c) Display the graph using the spring layout. Use the edge's 'radius' attributes as weights for the spring layout.
- d) Display the graph using the shell layout. Distribute the nodes across 5 different shells.
- e)\*\* Try to display the graph in a 3-dimensional spring layout (hint: use matplotlib's *mplot3d* toolkit). Scale the color and width of the edges according to their radius.

## 2 Graph Colorings

- a) Load the graph from the file `huge_graph.gpickle`. How large is the graph?

HINT: Matplotlib has difficulties displaying such large quantities of objects (dots and lines). It might be easier to turn off matplotlib's interactive mode (`plt.ioff()`) and save the plots as .png files to look at them (`plt.savefig('plot_name.png', dpi=600)`). Modify the `dpi` argument to increase/decrease resolution.

- b) Plot the graph using its 'x' and 'y' node attributes as coordinates and the edge's 'radius' attribute as edge-widths. What kind of graph is this?
- c) Choose colors, linewidths, marker- and linestyle in a way that they support the information contained in the graph (very subjective :-)).
- d) What is the maximum degree of the nodes in the graph? Color the nodes differently according to their degree. Scale the node sizes according to the graph's node attribute 'conductivity'.
- e)\* Color the thickest 15% of the edges differently than the rest to highlight them.
- f)\*\* Try to find the outermost cycle of the graph and color it differently.