Physics 514 – Basic Python Intro, Part IV Exercises

Emanuel Gull

September 9, 2014

1 2d surface plots

1. Write a function to compute the function

$$f(x, y) = e^{-\frac{x^2 + y^2}{2}} \tag{1}$$

2. compute it on a two-dimensional mesh between -2 and 2.

3. Use Matplotlib's built in `help` to learn about contour and surface plots.

4. Use Matplotlib to produce a contour plot of the function. Save your result as a PDF.

5. Use Matplotlib to produce a surface plot of the function. Save your result as a PDF and e-mail it with the code to `egull@umich.edu`.

2 Matrix Multiplication

1. Create two square matrices as nested lists. Multiply them using nested `for` loops, time your multiplication and show how the time scales as a function matrix size. Make a graph and save your results as a PDF.

2. Create two numpy matrices with random entries. Multiply them using for loops.

3. Consult your python `man` page about optimization in python. Run and time the optimized program.

4. Multiply your numpy matrices with the numpy `dot` function.

5. Generate a scaling plot with `log-log` axes, fit the expected scaling. Plot data and interpolation of all methods on the same graph. Label your axes, put figure captions, save as a pdf, and send the plot and codes to `egull@umich.edu`

3 Percolation

1. Generate a square numpy matrix of size L with entries 1 (with probability p) and 0 (with probability $1-p$).

2. Define clusters of occupied sites (sites with entries 1) using the Hoshen Kopelman cluster labeling algorithm:

   - Loop through all sites of the lattice. Check if the site has entry 1 and if the neighboring sites already have a label.
   - If no neighboring site is occupied or has a cluster label assigned: assign a new label.
   - If one neighboring site is occupied and has a cluster label assigned: assign the same label.
   - If more than one neighboring site is occupied and all have the same label: assign this label to the current site.
   - If more than one neighboring site is occupied: assign one of the labels and mark that the labels are equivalent. We use the following trick to avoid relabeling all wrongly labeled sites. For each label we keep a list of the proper labels, initialized originally to the label itself. To obtain the cluster label of a site we first obtain the label number assigned to the site. Next we check its proper label. If the two agree we are done. Otherwise we replace the label by the stored proper label and repeat this process until the label agrees with its proper label. This way we avoid relabeling all wrongly labeled sites.

3. Compute the probability $M_L(p)$ that a cluster on a lattice with fixed size L is percolating, *i.e.* that it goes from one end of the matrix to the other (e.g. left to right, top to bottom). Average over many clusters.

4. Create a plot with $M_L(p)$ versus p for several L. Roughly estimate the 'critical' percolation density, i.e. the density at which there would be an infinite cluster in the infinite system (try L-values between 8 and 128). Send code, estimate, and graphs to `egull@umich.edu`

4 Convolution

1. Write a program to compute the integral

$$(f * g)(t) = \int f(\tau)g(t - \tau)d\tau \tag{2}$$

   where f and g are discretized on an equidistant grid. Test your function by integrating $f = sin$ and $g = cos$ between 0 and π.

2. Find a way to do the same in $O(NlogN)$ time, using fast convolutions.

3. Plot the result and e-mail it with code to `egull@umich.edu`.