

# Math 116 Numerical PDEs: Homework 7

due Mon 9am Feb 27

*Shorter one to let you do projects. Here you study a toy version of a ‘fast method’ for multiplying a vector by a dense  $N \times N$  interaction (e.g. Nyström) matrix. We work in  $\mathbb{R}^2$  which you’ll find easier to think of as  $\mathbb{C}$ .*

1. Measuring rank of interaction.

- (a) Given  $N$ , set up a list of  $N$  source points chosen uniformly random in  $[-1/2, 1/2]^2$ , and  $N$  target points also chosen uniformly random in  $[3/2, 5/2] \times [-1/2, 1/2]$ . (Use a single complex number to represent each point.)
- (b) Study the numerical rank of the source-target interaction matrix given by the kernel  $(1/2\pi) \log \frac{1}{\|\mathbf{x}-\mathbf{y}\|}$ . As  $N$  increases, what does it do, specifically?
- (c) For  $N = 100$ , compute the numerical rank with tolerance  $10^{-10}$  and compare it to the predicted maximum number of multipole coefficients  $P$  needed for this tolerance as in lecture (you may ignore  $O(1)$  constants).

2. Applying the low-rank property to  $O(N)$  fast evaluation of naive  $O(N^2)$  interactions.

- (a) Read in the list of  $10^6$  charge source locations and charge values from gzip-compressed text file `sourcecharges.dat`. The columns are: coord1 coord2 charge. Each row is one source. [Hint: see `textread` command.] Note that values have been chosen to be multiples of  $10^{-4}$  to keep the file a reasonable size, but that your calc should still be to full accuracy. The target locations are then the sources translated by  $(2, 0)$ . Sources and targets should fall inside the boxes of the previous question.
- (b) Either by evaluating the multipole coefficients, or by filling skinny and tall matrices and applying them, compute the potentials at the targets due to the sources. Use an expansion of order  $P$  from the previous question. [Hint: always vectorize the loop over  $N$ , otherwise too slow. Debug by checking against the direct evaluation for a single target]
- (c) Report the  $l^2$  norm (ie root of sum of squares) of your vector of potentials at all targets. It should be accurate to around 10 digits (I will check it). [Hint: it is nearest to the integer 9721]

BONUS: Estimate how many times faster than naive summation your method is at this  $N$ . [Note it is too big to run the naive sum!]