MPCS 58020 2017 Homework 5

Due date: June 10, 2017

May 25, 2017

1. *Principal Component Analysis*
   Create an $m \times n$ dataset by executing an n-point 1D difussion solve over m time *snapshots*. A matlab 1D diffusion code (diffusion.m) is stored in the hw 5 dropbox folder. If you prefer not to use Matlab translating this to another language will be very quick and simple.

   The diffusion solver simulates the motion of thermal energy on a 1d domain. It starts with a specified initial condition (provided in sample code) and then integrates the 2d linear diffusion equation forward in time, producing a new (slightly changed) spatial distribution of thermal energy at each timestep. You should use the parameter values in the exmaple code – in particular note that you do not want to write the data every timestep (changes are too small).

   The data stored will be the $m \times n$ matrix that is the input dataset to your PCA analysis. You will need to write and submit your own PCA routine for this exercise. It is very simple as you are welcome leverage built-in eiegensolvers (take Numerical Methods if you want to write your own eigensolver).

   (a) create a plot of cumulative explained variance vs. mode number (assuming modes are ordered by most to least significant)

   (b) inspect the output visually compared to the raw data and attempt to provide a plausible explanation for how they might be interpreted.

   (c) inverse transform the PCA data after cutting off at the 90% threshold of explained variance. What features does the compressed data miss/capture?

   (d) compute the periodogram at each timestep and compare data compression efficiency using metrics of your choice.

2. *Branching Brownian Motion*
   Branching Brownian Motion (BBM) exhibits subtle behavior that lends itself to many of the concepts studied this quarter. For this exercise you may either write your own 2D BBM script or directly use/adapt/copy the bbm.m code provided in the course dropbox.

   Carry out the following analyses:

   (a) Using the parameters provided in the course sample code, carry out a simulation for 10,000 timsteps tallying the particle density (fraction) at each timestep. Do this for tally grids of size 10×10, 20×20 and 30×30 using both reflective and periodic boundaries. Provide plots of final tally results for each case.

   (b) Plot the average autocorrelation function (averaged across tally bins) for the simulations above.

   (c) Redo part (a) but using ensembles to calculate mean quantities rather than timesteps. For ensembles you use a burn-in period of several hundred timesteps before recording data.

   (d) For the periodic case and assuming that the correct solution is a uniform distribution, plot the error as a function of time for both the time averaged and ensemble cases. Explain what you see.

   (e) Quantify the amount of particle clustering evident in the time dependent calculations by computing and plotting the following as a function of time: i) mean nearest neighbor distance ii) Ripley's L function iii) mean squared distance

3. *Markov Chain Monte Carlo and Stochastic Integration*
   Write three programs to estimate the value of the 7-dimensional integral of

   $$f(x_1, x_2, ..., x_7) = e^{-x_1^2} e^{-x_2^2} ... e^{-x_7^2}$$

   on the interval $[0, 1]$. The first will use a discrete method – e.g. the trapezoidal rule applied to each dimension. The second will use regular Monte Carlo – ie we pick values of $x1..x7$ randomly, evaluate the function at those points, and average the results. The third will use some implementation of Metropolis Hastings. Provide the codes in each case and performance results for a given accuracy. Specifically, comment on how many samples/grid points are needed in each case to get a "good" estimate of the result.