# Physics 880.05: Problem Set #2

The MATLAB explorations (problems 1–3) are due on Tuesday, Oct. 27 and the other problems (4–5) are due on Friday, Oct. 30. See the online lecture notes for details of any of the individual topics covered in this problem set. All MATLAB scripts referenced below are available from the 880 webpage. Check the 880.05 webpage for suggestions and hints. Please give feedback early and often (and stop by to ask about anything).

1. **MATLAB Sandbox, Part II.** Here are some exercises to follow up on class discussions. You don't need to turn in a printout of your session, just your answers to the questions.

   (a) Seeding the random number generator.

      i. Start a new MATLAB session (quit any current session) and then give the command: `format long; rand(3,1)` to generate three random numbers between 0 and 1. (If you typed `rand(3)`, you would get a $3 \times 3$ matrix of random numbers; `rand(3,1)` gives you a $3 \times 1$ random matrix, i.e., a vector.) Note them down or copy them to a file. Now quit MATLAB, start it again, and give the same command. What do you conclude?

      ii. Start a new session but first *seed* the random number generator with
      `rand('state', sum(100*clock));`.
      (Give the command `clock` by itself and also `help clock`.) Try to generate the three numbers again; are they "random" now? [Note: `rand` and `randn` need to be seeded independently if you use both of them.]

   (b) The random number generator `randn` returns numbers distributed as a gaussian ("normal") with mean zero and standard deviation one. Show how to generate a vector `v` of 100 numbers with mean 10 and standard deviation 5. Check your result by histogramming it: `hist(v)`. Try 1000 and then 10000 numbers with 20 and 50 bins [`hist(v,20)` will use 20 bins)]. Observations? [Note: You might find the "Rolling Dice" Mathematica notebook linked on the 880.05 webpage to be amusing.]

   (c) Exponentiating matrices.

      i. Download and unzip `expm_demos.zip`. There should be three MATLAB "M-files" that calculate the exponential of a matrix three ways. Look at each of the programs and read the comments. Briefly, how do each of them work?

      ii. Create a random $500 \times 500$ matrix `A` (use ";" at the end of the command to keep from printing it out!). Compare the time it takes each method to exponentiate `A` by using `tic;` and `toc;`. For example, `tic; expm_series(A); toc;`. Which is fastest? (Try a few times to see if your answer is robust.) Does it matter whether the matrix is distributed uniformly or gaussian?

iii. The command `disp(x);` will print the value of the variable `x`. Add a line to `expm_series.m` to print out the number of terms used in calculating the series for exponentiating a matrix. What is the typical number of terms needed to exponentiate a random $100 \times 100$ matrix distributed normally versus one distributed uniformly from 0 to 1?

2. **Fun with the One-Particle Stochastic Variational Method in MATLAB.** A zip archive with MATLAB programs to apply the SVM to one-particle problems can be downloaded from the 880.05 webpage as `svm_test.zip`.

   (a) There are five ".m" files with the main program (`svm_test.m`) and some functions. Try to follow the flow of the program (ask questions if confused!). Currently only the Coulomb potential is implemented (the comments tell you the exact eigenvalues in the chosen units). What are the advantages of using a gaussian basis? (You may want to review the SVM discussion in the class notes from 9/28.)

   (b) After starting MATLAB in the directory with the ".m" files (or changing to this directory after launching MATLAB), you run the code by typing `svm_test`. The parameters to play with are at the top of `svm_test.m`. Given a basis size `Ndim`, the program estimates the lowest eigenvalues two ways: by picking `Ndim` gaussian widths at random to determine the basis versus building the basis one gaussian at a time with the SVM procedure. Try running with basis sizes 5, 10, 15, 20, 25. Briefly comment on the results. What goes wrong with the larger basis size (hint: ill-conditioning)?

   (c) For a basis size of 20, which method does better for the first and second excited states of the hydrogen atom? Why for this case does choosing the best basis for the ground state not work so well for excited states? Describe how you would change the program to get good SVM answers for *both* the lowest and first-excited states.

3. **Stochastic calculation of multidimensional integrals.** A zip archive with MATLAB programs to calculate multidimensional integrals stochastically (as described in class) can be downloaded from the 880.05 webpage as `MCIntEval.zip`. Run it with `MCIntEval`.

   (a) What is the integral you are calculating (including the limits)? What roles do the "Action" and "Integrand" play? How is this like calculating a path integral?

   (b) Run the code several times. What are the three outputs? What does the histogram show? Adjust `eps` until the acceptance rate is roughly 50%. Why do want this? Explain the difference in the histogram before and after you change `eps`.

(c) What is "thermalization" supposed to do? Does it have a big effect in this problem? Why or why not?

(d) If you want your answer to be 10 times more precise, what parameter would you change and by how much?

(e) Change the action to add a $\frac{\lambda}{4}x^4$ term. What happens with $\lambda = 0.1$ versus $\lambda = -0.1$? Explain (in physics terms).

4. **Continuing with the Partition Function for One-Particle.** In class we derived the expression

$$\mathcal{Z}[f] = \int \prod_{i=1}^{N_\tau} dx_i \exp\left(-\frac{1}{2}\sum_{i,j=1}^{N_\tau} x_i A_{ij} x_j + \sum_{i=1}^{N_\tau} f_i x_i\right) = \mathcal{Z}_0 \exp\left(-\frac{1}{2}\sum_{i,j=1}^{N_\tau} f_i [A^{-1}]_{ij} f_j\right) \quad (1)$$

where we have assumed that $A$ is a matrix that is real, symmetric and positive definite (i.e., $\sum_{i,j=1}^{N_\tau} x_i A_{ij} x_j > 0$). Here we examine some aspects of this discrete version of $\mathcal{Z}[f]$.

(a) This is a central identity of many-body physics and quantum field theory, so it's important to understand it thoroughly.

i. Rederive the second equality (try to not look at the notes!). What is $\mathcal{Z}_0$?

ii. What happens to the integral if $A$ is not real? What other condition will guarantee that the integral still exists?

iii. What happens if the matrix is real but not symmetric?

(b) Assume, as we saw in class, that (with $m = 1$, $\epsilon = 1$, and $a = 0$)

$$A_{ij} = 2\delta_{ij} - \delta_{i,j+1} - \delta_{i,j-1} \quad (2)$$

with $i, j = 1, \ldots, N_\tau$ and periodic boundary conditions, i.e. if $j \pm 1$ is outside the $[1, N_\tau]$ interval then use these rules: if $b = j+1 = N_\tau + 1$, set $b = 1$; if $b = j - 1 = 0$, set $b = N_\tau$.

i. Construct the matrix $A_{ij}$ explicitly for a small $N_\tau$ (=10 or so) and argue that its elements depend only on the difference $|i-j|$ (e.g., consider $|i-j| = 0$, $|i-j| = 1$, etc.). Why is this property called translational invariance?

(c) Matrices that are translation invariant can be diagonalized using Fourier transforms.

i. Find the (discrete) Fourier transform of $A$ from part (b):

$$\widetilde{A}_{q,q'} = \frac{1}{N_\tau}\sum_{k,j=1}^{N_\tau} \exp\left(i\frac{2\pi q}{N_\tau}k\right) A_{k,j} \exp\left(-i\frac{2\pi q'}{N_\tau}j\right) \quad (3)$$

and check that $\widetilde{A}_{q,q'}$ is diagonal (do this both analytically and using MATLAB).

ii. Generalize the previous result to an arbitrary matrix $B$, i.e. check that it can be diagonalized by a Fourier transform assuming only that it is translation invariant.

(d) Notice that what you just did is a unitary transformation:

$$\widetilde{A} = U^\dagger A U \tag{4}$$

where $U^\dagger U = I$.

i. Identify $U$ and $U^\dagger$; what do they look like in explicit matrix form?

ii. Express $A^{-1}$ in terms of $\widetilde{A}^{-1}$ and $U$, $U^\dagger$. Write $A^{-1}$ explicitly as a sum over $k$, using the result of part (c). Congratulations, you have computed your first lattice propagator!

(e) Use the generating functional $\mathcal{Z}[f]$ (i.e., the partition function with a source) to determine $\langle x_i x_j \rangle$ in terms of $A^{-1}$.

5. **Directly Solving for the Green's function $\mathcal{G}^0$.** When evaluating a continuum path integral in perturbation theory we need the inverse of a differential operator (these are the propagator lines in Feynman diagrams and the continuum version of $A^{-1}$). Your goal is to solve

$$\left(\frac{\partial}{\partial\tau} - \frac{\boldsymbol{\nabla}^2}{2M} - \mu\right)\mathcal{G}^0(\mathbf{x}\tau;\mathbf{x}'\tau') = \delta^3(\mathbf{x}-\mathbf{x}')\delta(\tau-\tau')$$

for the propagator function $\mathcal{G}^0$, subject to the fermion boundary conditions:

$$\mathcal{G}^0(\mathbf{x}\beta;\mathbf{x}'\tau') = -\mathcal{G}^0(\mathbf{x}0;\mathbf{x}'\tau')\ .$$

(a) First go to momentum space. Introduce the Fourier transform:

$$\begin{aligned}\mathcal{G}^0(\mathbf{k};\tau-\tau') &= \int d^3x\, e^{-i\mathbf{k}\cdot(\mathbf{x}-\mathbf{x}')}\,\mathcal{G}^0(\mathbf{x}\tau;\mathbf{x}'\tau') \\ \mathcal{G}^0(\mathbf{x}\tau;\mathbf{x}'\tau') &= \int \frac{d^3k}{(2\pi)^3}\, e^{i\mathbf{k}\cdot(\mathbf{x}-\mathbf{x}')}\,\mathcal{G}^0(\mathbf{k};\tau-\tau')\end{aligned}$$

and find what differential equation $\mathcal{G}^0(\mathbf{k};\tau-\tau')$ satisfies. How do we know that $\mathcal{G}^0$ is a function of $(\mathbf{x}-\mathbf{x}')$ and $(\tau-\tau')$?

(b) Split the time into two regions: $(\tau-\tau') > 0$ and $(\tau-\tau') < 0$. Find general solutions to the differential equation in each region (they will have different coefficients!).

(c) What are the matching conditions for the two solutions? Determine the coefficients by matching and applying the boundary condition, which then gives you the solution to $\mathcal{G}^0(\mathbf{x}\tau;\mathbf{x}'\tau')$. Write it with appropriate $\theta$-functions.