Problem Sheet 1
Lectures in Hadron Spectroscopy
Sinead Ryan

Question 1
Consider the interpolating operator for a pion (flavour non-singlet), $\mathcal{O} = \bar{u}\gamma_5 d$. Verify that the two-point correlation function can be written

$$C(t) = \langle \mathrm{Tr}\left[G_u(x,y)\gamma_5 G_d(y,x)\gamma_5\right]\rangle$$

where $G_f(x,y)$ is the quark propagator for flavour f.

Now consider $\mathcal{O} = \frac{1}{\sqrt{2}}\left(\bar{u}u + \bar{d}d\right)$. Show that the two-point correlation function for this operator leads to the same expression as above with an addition (called *disconnected*) contribution, which can be writte

$$D = 2\langle\left(\mathrm{Tr}\left[\Re(G_u)\right]\right)^2\rangle$$

assuming the u and d quarks are degenerate.

Question 2
At

http://www.maths.tcd.ie/ ryan/INT2012/

you will find a file called

`jpsi_correlator_16x128.dat`

This is two-point correlator data for a J/Ψ meson generated with a sinlge interpolating operator: $\mathcal{O} = \bar{c}\gamma_i c$. The lattice is $16^3 \times 128$ and there are 20 configurations. The data are indexed by timeslice value for each configuration.

- Plot the correlator data to see the exponential (and time-symmetric) behaviour.
- Write a short program to determine the effective mass on each timeslice am_{eff} and plot the result.
- Hopefully you will see a plateau (no time dependence) at large times!
- Calculate the jackknifed errors for each point.

Question 3

Write a short program to implment the Metropolis algorithm (described below) for the one-dimensional harmonic oscillator: $V(x) = x^2/2$ with $m = 1$.

Compare your results with usual quantum mechanics results.

Calculate the energy and wavefunction and calculate

$$G(t) = \frac{1}{N} \sum_j \langle x(t_j + t) x(t_j) \rangle \;\; \forall t = 0, 2a, \ldots (N-1)a$$

Try $N = 20$ sites with spacing $a = 0.5$ and $\epsilon = 1.4$. Use $N_{configs} = 25, 50, 100, 10000$ and see the effect.

Repeat for $V(x) = x^4/2$.

- To calculate an ensemble of fields (paths) weighted by a probability, a simple algorithm is the Metropolis algorithm
  - start with an arbitrary field
  - modify by visiting each site on the lattice updating fields at those sites, one at a time according to the recipe below.
  - continue to apply updates to generate an ensemble of N_{config} fields.
- The updating proceeds by
  - generate a random number $-\epsilon \leq \xi \leq \epsilon$ with uniform probability. [see eg Box-Muller algorithm in Numerical Recipes]
  - change the value of the field $U(x) \rightarrow U(x) + \xi$ and compute ΔS, the change in the action.
  - if $\Delta S < 0$: retain the new value and proceed to next site
  - if $\Delta S > 0$: accept change with probability $e^{-\Delta S}$
    - generate a random number η uniformly distributed in [0,1]
    - keep new value if $e^{-\Delta S} > \eta$, else keep old value and proceed to next site.