# Mass Spectroscopy

Patrick Oare

## 1 Overview

The idea of mass spectrometry is to use the lattice to calculate two point functions, and then to take these two point functions and extract physics. The central equation to this is the following:

$$C(n_t) := \langle \mathcal{O}(n_t)\overline{\mathcal{O}}(0)\rangle = \sum_k \langle 0|\hat{\mathcal{O}}|k\rangle\langle k|\hat{\mathcal{O}}^\dagger|0\rangle \exp\left(-n_t E_k\right) \tag{1}$$

Here $|k\rangle$ represents any intermediate state which can be connected to the operator $\mathcal{O}$ with energy $E_k$, and $n_t$ is the time slice we are calculating the correlation function on. The operators $\hat{\mathcal{O}}$ and $\hat{\mathcal{O}}^\dagger$ are related to the objects $\mathcal{O}$ and $\overline{\mathcal{O}}$ by the following identity:

$$\mathcal{O}(\alpha)\delta(\alpha-\alpha') := \langle \alpha'|\hat{\mathcal{O}}|\alpha\rangle \tag{2}$$

where $|\alpha\rangle$ is a continuous eigenbasis of the space you are considering.

Using the relation between two point functions and matrix elements, we may extract physics about the energies of nucleon states. To do this, we consider evaluating two point functions when $\mathcal{O}$ is an **interpolator** for the state we want to analyze. By an interpolator, we mean that $\mathcal{O}$ creates or destroys the state of interest. For example, a proton interpolator $\chi(n)$ will destroy a proton at site $n$, and its adjoint $\overline{\chi}(n)$ will create a proton at site n.

So, suppose that we want to calculate the mass of a hadron $h$ with corresponding interpolator $\chi$. Then using Equation 1, we have:

$$\langle \chi(n_t)\overline{\chi}(0)\rangle = \sum_k \langle 0|\hat{\chi}|k\rangle\langle k|\hat{\chi}^\dagger|0\rangle \exp(-n_t E_k) \tag{3}$$

Because the operator $\hat{\chi}^\dagger$ creates a hadron state $\hat{\chi}^\dagger|0\rangle$, the sum over intermediate states $|k\rangle$ will only connect states with the quantum numbers of the hadron of interest to the ket $\hat{\chi}^\dagger|0\rangle$. So, the sum $\sum_k$ will only index over intermediate states which contain at least one hadron of species $h$.

Consider taking the large $n_t$ limit of this equation. When $n_t$ gets large, intermediate states with a larger energy $E_k$ become exponentially suppressed, and the dominant term in this series is the state $|k\rangle$ with the lowest energy. Here is the key point: *the lowest energy state is exactly the ground state, and the energy will be given (at zero momentum) by* $E_0 = m_h$. Thus, as we take $n_t$ to be larger and larger, we can extract the hadron mass $m_h$ by calculating the two point function of the interpolating operator:

$$\langle \chi(n_t)\chi^\dagger(0)\rangle = A\exp(-n_t E_0)(1+O(e^{-n_t\Delta E})) \tag{4}$$

where now $A$ is a constant we can determine with a data fit.

### 1.1 Momentum Projection

The state of the hadron we have created with the interpolator has not been completely specified; although we know it lies on a fixed timeslice, we do not know the momentum of the hadron. If we compute the corrlation function $\langle\chi(\mathbf{n}, n_t)\chi^\dagger(0)\rangle$, we will know the location of the particle we have destroyed, but we will not know its momentum. We instead want to make sure that we project onto a state of definite momentum, which we will do with a Fourier transform:

$$\langle\chi(\mathbf{p}, n_t)\chi^\dagger(0)\rangle = \frac{1}{\sqrt{|\Lambda_3|}}\sum_{\mathbf{n}\in\Lambda_3} e^{-ia\mathbf{p}\cdot\mathbf{n}}\langle\chi(\mathbf{n}, n_t)\chi^\dagger(0)\rangle \xrightarrow{n_t\to\infty} A\exp(-n_t E_0(\mathbf{p})) + ... \tag{5}$$

where $\Lambda_3 = \{(\mathbf{n}, n_t) \in \Lambda : \mathbf{n} \in \mathbb{Z}^3\}$ is the set of lattice points on our fixed time slice ($\Lambda$ is the full lattice). The important point here is that we know the dispersion $E(\mathbf{p})$, which is simply:

$$E(\mathbf{p}) = \sqrt{\mathbf{p}^2 + m_h^2} \tag{6}$$

In practice, we will often project onto $\mathbf{0}$ momentum, which is easily achievable through summing on spatial points in the lattice:

$$\langle\chi(\mathbf{p} = \mathbf{0}, n_t)\chi^\dagger(0)\rangle = \frac{1}{\sqrt{|\Lambda_3|}}\sum_{\mathbf{n}\in\Lambda_3} \langle\chi(\mathbf{n}, n_t)\chi^\dagger(0)\rangle \xrightarrow{n_t\to\infty} A\exp(-n_t m_h) + ... \tag{7}$$

### 1.2 Effective Mass

In theory we would like to take the limit as $n_t \to \infty$ to remove contamination from the higher energy states, but on a finite lattice this is not possible. So, we need to define a way to know when $n_t$ is large enough that the energy we extract from this method is sufficiently close to the mass of the hadron. A simple way to do this is to define a mass scale by noticing that as $n_t \to \infty$, $C(n_t) \to Ae^{-n_t m_h}$. In this limit, we can extract the mass by taking a ratio of correlation functions:

$$\frac{C(n_t)}{C(n_t + 1)} \xrightarrow{n_t\to\infty} e^{m_h} \tag{8}$$

This motivates the definition of the **effective mass**:

$$m_{eff}(n_t) := \log\left(\frac{C(n_t)}{C(n_t + 1)}\right) \tag{9}$$

As $n_t$ becomes sufficiently large, $m_{eff}(n_t)$ approaches a constant an plateaus at the value $m_{eff} = m_h$. At smaller time slices, $m_{eff}(n_t)$ will not appear to be constant but instead will also contain contamination from higher energy modes. Thus in essence, it suffices to compute the effective mass and to fit a constant line to the region where it plateaus to determine the actual mass of the hadron.

Because we often use periodic boundary conditions on the lattice, it is often useful to take this into account and define a variation of the effective mass.

### 1.3 Bootstrapping

Assume now that we have calculated values of the interpolation function on $n$ gauge field configurations $\{U_i\}_{i=1}^n$, and that our lattice has size $L^3 \times T$. Let the values of the interpolators be contained in the set:

$$\{C(i, t)\}_{i=1,t=1}^{n,T} \tag{10}$$

where $C(i,t)$ is the value of the interpolator on the $i$th gauge field configuration, momentum projected to 0 on the $t$th time slice. Generally these values $C(i,t)$ will be stored in a matrix $C_{it}$ to make them easier to work with. If we simply want to ensemble average of these interpolators, at this point we could average them over the configuration index to find:

$$\langle C(t)\rangle = \frac{1}{n}\sum_{i=1}^{n} C(i,t) \tag{11}$$

This would give us an average for our correlation function computed at each time slice, which is what we originally sought after.

However, this is not the best way to establish statistics on this problem. Instead, we consider a method called **bootstrapping**, which will allow us to produce a larger amount of ensembles from our initial data $C(i,t)$. With this method, we consider generating $N_{boot}$ new ensembles from our original one. To do this, fix $t$ and pick $n$ entries from $C(i,t)$ randomly *with replacement*. Denote these $n$ samples by $U_1(i,t)$. Repeat this sampling procedure $N_{boot}$ times until you have a collection $U_1(i,t),...,U_{N_{boot}}(i,t)$ of ensembles sampled from your original data. We may use the notation $U(b,i,t)$ for $U_b(i,t)$, where $b = 1,...,N_{boot}$, $j = 1,...,n$, and $t = 1,...,T$ to refer to these ensembles and to emphasize its structure as a 3-dimensional matrix.

The ensemble $\{U_1,...,U_{N_{boot}}\}$ will be used for our data analysis. To compute the correlation functions $\langle C(t)\rangle$, we will first determine the average over each ensemble:

$$\langle C(t)\rangle_b := \frac{1}{n}\sum_{i=1}^{n} U(b,i,t) \tag{12}$$

The idea is that now we have an ensemble of average values for $\langle C(t)\rangle$, so we may compute statistics on the set $\{\langle C(t)\rangle_1,...,\langle C(t)\rangle_{N_{boot}}$, i.e. we may take the average and (sample) variance of it as follows:

$$\langle C(t)\rangle = \frac{1}{N_{boot}}\sum_{b=1}^{N_{boot}} \langle C(t)\rangle_b \tag{13}$$

$$\sigma^2_{C(t)} = \frac{1}{N_{boot}-1}\sum_{b=1}^{N_{boot}} (\langle C(t)\rangle_b - \langle C(t)\rangle_b)^2 \tag{14}$$

Using the bootstrap method, we may calculate statistics on more than just correlation functions. For example, to calculate the effective mass we can create $N_{boot}$ ensembles of effective mass values, then take the statistics on them. Doing this gives us a more accurate estimation of the mean and error on a data set.

## 2 Example: Pion Mass