Three point function analysis

Archana Radhakrishnan

Spring 2019

1 Running the code

Requirements: `itpp, fitting_lib, adat, semble`
Running: The main code is **`ensemble_fit_three_pt_fns`**. The input xml contains:

1. the edb file name
2. the source time slice
3. sink time slice
4. the jack-knife files of the energy levels of the source and sink
5. the xml of kinematic factors
6. the minimum number of time slices to use in the fit

The input contains the jack-knife files of ground state energy at the source and the sink (the exponential of which we divide the three-point function by), if the jack-knife file is not available for all momenta, you can make it for single particle, using the dispersion relation from $p = 0$ by using:

```
/three_pt_analysis/three_pt_fit/build/src/make_elab
```

The xml has the kfac.xml which contains the kinematic factors in future we have to divide the constant in the fit by these factors to get the value of $F(Q^2)$ χ^2 cutoff, the minimum allowed noise levels in the data, and the type of "χ^2" to be used to pick the best fit out of the fits performed. An example of the xml is in:
`/three_pt_analysis/three_pt_fit/build/runs/3pt.ini.xml`

2 Fit Function for three-point fits

A three-point function is of the form,

$$\begin{aligned}&\langle 0|\, O_f(p', \Delta t) j^\mu(p-p', t) O_i^\dagger(p, 0)\, |0\rangle_T \\ &\qquad = \lim_{T\to\infty} \frac{1}{Z_T} \langle 0|\, e^{-(T-\Delta t)\hat{H}} O_f(p', 0) e^{-\Delta t \hat{H}} e^{t\hat{H}} j^\mu(p-p', 0) e^{-t\hat{H}} O_i^\dagger(p, 0)\, |0\rangle\end{aligned}$$

where Z_T is the normalization and we time evolve at the sink. For now we drop Z_T. Since it is not possible to know the exact operator for producing states *apriori*, we use a basis of operators that have the same J^{PC} as the state we wish to produce. We the form 'optimized operators' at the source and the sink. What this means is that we have used the variational method to diagonalize the respective two-point correlation functions and rotate to a new basis which is a linear combination of the old basis and projects optimally to the energy level we require at the source and that sink.

$$\Omega_f^n(p',0) = \sum_{f'} Z_{ff'}^n O_{f'}(p',0) \tag{1}$$

$$\Omega_i^{n\dagger}(p,0) = \sum_{i'} Z_{ii'}^n O_{i'}^\dagger(p,0) \tag{2}$$

where $\Omega_i^{n\dagger}(\Omega_f^n)$ are the optimized operators of the n^{th} state, $Z_{ii'}^n$ $(Z_{ff'}^n)$ are the coefficients or weights and $O_{i'}^\dagger(O_{f'})$ is the old basis at the source(sink). Since the correlation function is diagonal in this basis, the 'n^{th} optimized operator' have a better overlap with the n^{th} state than the old basis provided the original basis that we started from does a good job of spanning the states we wish to produce. We can write it as,

$$\Omega_i^{n\dagger} |0\rangle = |i\rangle_n + \sum_{i'\neq n} \varepsilon_{ii'}^n |i'\rangle$$

$$\Omega_f^{n\dagger} |0\rangle = |f\rangle_n + \sum_{f'\neq n} \varepsilon_{ff'}^n |f'\rangle$$

Here, $|i\rangle_n$ and $|f\rangle_n$ are the n^{th} state of i and f. $\varepsilon_{ii'}^n$ and $\varepsilon_{ff'}^n$ are negligible if there is minimal contamination from other levels.

$$\begin{aligned}&\langle 0| \Omega_f^n(p',\Delta t) j^\mu(p-p',t) \Omega_i^{n\dagger}((p,0) |0\rangle_T \\ &\qquad = \lim_{T\to\infty} \langle 0| e^{-(T-\Delta t)\hat{H}} \Omega_f^n(p',0) e^{-\Delta t\hat{H}} e^{t\hat{H}} j^\mu(p-p',0) e^{-t\hat{H}} \Omega_i^{n\dagger}(p,0) |0\rangle\end{aligned}$$

after taking the limit $T \to \infty$. Adding a complete set of states, $\sum_{f'} |f'\rangle \langle f'|$ and $\sum_{i'} |i'\rangle \langle i'|$,

$$\begin{aligned}&\langle 0| \Omega_f^n(p',\Delta t) j^\mu(p-p',t) \Omega_i^{n\dagger}((p,0) |0\rangle \\ &\qquad = \sum_{f'i'} \langle 0| \Omega_f^n(p',0) e^{-\Delta t\hat{H}} e^{t\hat{H}} |f'\rangle \langle f'| j^\mu(p-p',0) |i'\rangle \langle i'| e^{-t\hat{H}} \Omega_i^{n\dagger}(p,0) |0\rangle \\ &\qquad = \sum_{f'i'} e^{-(\Delta t-t)E_{f'}} e^{-tE_{i'}} \varepsilon_{ff'}^{*n} \langle f'|f'\rangle \langle f'| j^\mu(p-p',0) |i'\rangle \langle i'|i'\rangle \varepsilon_{ii'}^n\end{aligned}$$

where $\varepsilon_{ii}^n = \varepsilon_{ff}^n = 1$ from above. $E_{i'}$ and $E_{f'}$ are the energies of the state $|i'\rangle$ and $|f'\rangle$. Multiplying by the exponential of the energy of the state $|i\rangle_n$ at the source and $|f\rangle_n$ at the sink, $e^{(-\Delta t+t)E_f}$ and e^{tE_i}.

$$\begin{aligned}&\langle 0| \Omega_f^n(p',\Delta t) j^\mu(p-p',t) \Omega_i^{n\dagger}((p,0) |0\rangle \\ &\qquad = \varepsilon_{ff}^n \varepsilon_{ii}^n \langle f| j^\mu(p-p',0) |i\rangle_n \\ &\qquad + \sum_{f'\neq f i'\neq i} \varepsilon_{ff'}^n \varepsilon_{ii'}^n e^{-(\Delta t-t)(E_{f'}-E_f)} e^{-t(E_{i'}-E_i)} \langle f'| j^\mu(p-p',0) |i'\rangle \\ &\qquad = \langle f| j^\mu(p-p',0) |i\rangle_n \\ &\qquad + \sum_{f'\neq f i'\neq i} \varepsilon_{ff'}^n \varepsilon_{ii'}^n e^{-(\Delta t-t)(E_{f'}-E_f)} e^{-t(E_{i'}-E_i)} \langle f'| j^\mu(p-p',0) |i'\rangle\end{aligned} \tag{3}$$

From (Eq: 3), the fit function of the form $y(t) = F$ where F is a constant would work, when $\varepsilon_{ff'} = \varepsilon_{ii'} = 0$ which is a good approximation. But in cases where there is contamination from other states at the source and the sink, we can incorporate all the multi-exponentials at the two-points by a single exponential at the source and the sink, $y(t) = F + F_i e^{-t\Delta E_i} + F_f e^{-(\Delta t-t)\Delta E_f}$ seems to be a good choice. Depending on the contamination, if there is very little contamination at the source(sink) $F_i \sim 0 (F_f \sim 0)$.

So there are four fit functions we are going to use, $F,\ F+F_i e^{-t\Delta E_i},\ F+F_f e^{-(\Delta t-t)\Delta E_f}\ and\ F+F_i e^{-t\Delta E_i}+F_f e^{-(\Delta t-t)\Delta E_f}$, depending on which one describes the data better (where 'better' is defined by the fit quality)[1].

3 Fitting of Correlated Data using Minuit

This code is inferfaced with `fitting_lib` which is a wrapper over Minuit. Consider an ensemble of correlated data like, $n = 1, 2, ..., N$ unbiased data sets for $y_i(n)$ and $i = 1, 2, ..., \Delta t$. The the $y_i(n)$ are correlated in i and independent in n.

The strategy used here is to use the fit functions the user defines based on physics, e.g. in the case of three-point functions the fit function could be just a constant F, on the average of y_i over N, $\bar{y_i}$. Minuit does a correlated fit on the data. We use, SVD for the correlated fit resetting small eigen-values of the correlation matrix. A cut-off value of 10^{-6} is hard-coded.

In general Minuit can minimize any function provided by the user to choose a good fit. We choose to go with the traditional correlated χ^2. Once a minimum is obtained, Minuit returns the goodness of fit and the parameter errors.The parameter errors are calculated by fixing all parameters except the one we are interested in finding the error of and finding the change in the parameter as the function value changes by a fixed amount. If one does not use a χ^2, there will be additional normalizations to the parameter errors. For more details see [2]

For fitting three point functions we use four kinds of functions

- F
- $F + F_i e^{-t\Delta E_i}$
- $F + F_f e^{-(\Delta t-t)\Delta E_f}$
- $F + F_i e^{-t\Delta E_i} + F_f e^{-(\Delta t-t)\Delta E_f}$

We perform average fits for the above functions and fit ranges. The average fit is sorted according to a 'fit quality' defined by the user and is provided to a 'fit selector' that performs an ensemble fit which is a correlated fit by Minuit over single elimination jack-knife samples over N ensembles, using the the parameter values of the average fit as the starting point. The expectation is that the ensemble fit should not be that different from the average-fit unless the data has huge errors which makes it unreliable. In case there are exceptional configurations which tend to drive the fit far from the average fit, we eliminate them by examining the deviations in χ^2 and parameter values. As a rule of thumb, the deviation from the χ^2 for jack-knife samples should be $\sim \sqrt{\frac{2*DOF}{N-1}}$ ($DOF\ =$ degree of freedom). If the ensembble fit χ^2 is more than $8 \times \sqrt{\frac{2*DOF}{N-1}}$ away or the ensemble fit parameter values are more than 8 sigma away from average fit, the fitter eliminates that configuration.

If there are many such configurations, a fraction of 0.05 is hard coded, the ensemble fit is forced to fail.

4 Logic behind the fitting method

The interface to the edb:

1. Each edb contains correlation functions for a particular source-sink separation (Δt)
2. The interface loops over all the edbs the user provides and makes Data objects. The number of such objects is equal to the number of correlation functions in all the edbs. Each individual data object has the size equal to the number of source-sink separations present for that particular correlation function.

3. At this point the three point function data is divided by the exponential of corresponding ground state energies of the operator at the source and the sink. For example for

$$\langle \pi(\Delta t) | \, j^{\mu}(t) \, | \pi\pi \rangle \tag{4}$$

   we multiply by $e^{E_{\pi}(\Delta t - t)} e^{E_{\pi\pi} t}$. We get the energy levels from the "MassJackFiles" of the two-point functions. In case the irreps are not available use the dispersion relation to get the corresponding energies from the zero-momentum jackknife file.

4. The program creates directories in the parent directory for each correlation function and writes the output files in the newly created directory. The correlation function can be deduced by the directory naming.

The steps involved in the fitting procedure:

1. Take an input containing an ensemble of data with multiple Δt.

2. Get rid of data points at $t = 0$ and $t = \Delta t$ (contact terms) and data points which have a huge absolute noise (which is a user input) in y, if the user has a minimum(maximum) t, remove the data points below(above) it.

3. Pick the smallest Δt and do fits on the average. First do a constant fit. The start param is the value at the midpoint. Start from the midpoint, use the minimum time-slice provided by the user for the first fit. Then increase the range one by one on source and sink side. While increasing the ranges slide the fit window over the entire range (as a precaution in case the plateau of the data is not at the midpoint).

4. Pick the 'best' constant fit and use as that start param for the constant in the average fits containing constant + source exponential and constant + source exponential. The start params for the $F_i\ and\ F_f$ is the value of the constant in the 'best' constant fit. Th start param for $\Delta E_i\ and\ \Delta E_f$are 2.0 (Can probably think of a better choice?) Here, the range starts with the 'best' constant fit range and grows on the side of the source(sink) for the source exponential(sink exponential) fits.

5. Pick the 'best' average fit in each category and use it as the start params for constant + source exponential + sink exponential fit. In case there was no constant + source(sink) exponential fit use the start params as above. The range starts from the 'best' fit range among all the previous fits and grows in either direction as we iterate.

6. Choose the 'best' 20 fits for this Δt and go to the next smallest Δt. Repeat the above steps. Get 'best' 20 fits for each. Might be throwing away information, but the combined fit sizes can be enormous if we keep all the average fits. (Can probably have a sensible limit on the size ?).

7. Finally when all Δts are covered, do a combined average fit. If there are n Δt's then do a combined average fit on 20^n fits. Then perform ensemble fits.

   The fit-functions and the fit ranges are the same as that for individual Δts. The only parameter that is common to all Δts is the constant in the fit. The start params for each variable is the value of the average fit of the individual Δt. The start param of the constant is taken to be the average of the all Δts. An average fit is performed on the combined data. The average fits are then sorted based on a fit-quality.

8. Once the average fits for 20^n are done, take the ones that are successful (based on the average fits that cleared the χ^2 cutoff which is a user input and if the fits converged) and do an ensemble fit on the 'best' average fit. If the ensemble fit fails, then go to the next 'best' average fit and so on.

9. The fit selector then selects the first ensemble fit that succeeds as the fit function. The selection of 'best' is based on the fit quality that the user provides.

5 Selection of the 'best' fit

The $\mathcal{X}^2$ of the fit helps to determine the good fits. But when the $\mathcal{X}^2$ good enough for different fit functions and fit ranges we cannot compare them or pick a better fit using $\mathcal{X}^2$. So we have to define better functions that incorporates the constraints that we require.

There are many constraints that arise from the physics of three-point functions. Performing a spectral decomposition as in (Eq: 3), tells us that ideally if the operators are optimized we would just have to fit a constant to the data. This should be given more weightage. If the optimized operators are not ideal, we need to include the exponentials that are a measure of the excited state contaminations. Since the contamination is from the excited states, $\Delta E_i \textit{ and } \Delta E_f \geq 0$. One can use a stricter constraint of $\Delta E_i \geq E_{i_1} - E_{i_0}$ and $\Delta E_f \geq E_{f_1} - E_{f_0}$, since any excited state pollution would be bigger than the mass gap between the level we consider and the level just above that. Also, ΔE_i and ΔE_f are smaller than the value of the highest energy determined in the spectrum. The values of $F_i \textit{ and } F_f$ seem to have no particular constraints because even though the overlaps $\varepsilon^n_{ff'} \textit{ and } \varepsilon^n_{ii'}$ should be small, the matrix elements with the excited states need not be.

However, we need to eliminate fits where $E_i\ (E_f) \rightarrow 0$ and $y(t) = F + F_i\ (F_f) = \textit{ constant}$. Also, the fits where $E_i\ (E_f) \rightarrow \infty$ and $F_i\ (F_f) \rightarrow \infty$ where only a constant fit is required. In these case the data is statistically compatible with constant but the fit produces a constant and exponentials. This can be eliminated by applying the minimum and maximum constrains in the energy exponentials as mentioned above. We also have to eliminate the fits where exponentials are not necessary. This can be done by defining a good 'fit quality' that sorts the fits in an order that is sensible.

To get sensible fits,

- The user can apply the constraints on E_i and E_f discussed above
- The $\mathcal{X}^2$ is a valid estimate of the goodness of fit but maybe not a good function to compare fits by different functions and fit ranges of good enough $\mathcal{X}^2$. Thus, we filter the fits and choose the fits that are above a threshold of $\mathcal{X}^2$ which is defined by the user
- Then the fits which satisfy this condition are sorted by the 'fit quality' that inherently eliminates the "bad" fits defined above. The fit-quality should
  - prefer constant fits when the parameters are highly correlated or adding an exponential does not improove $\mathcal{X}^2$ by much.
  - should prefer to use as much data as possible i.e. prefer a broad range for t.
  - have reasonable values of the exponential i.e. the value that is comparable to the curvature of the data.
  - have a the value of F that is consistent with the region where the data reaches a plateau, and not too far away from this region. If the data does not have a plateau the data is not so reliable, but still the closest datapoint to the constant in the function must not be more than $5 - 6\sigma$ away.

Thus, the problem boils down to defining a good 'fit quality'. The function that I have cooked up to be maximized by the fit selector to get the best fit and that seems to work so far is :

- **`generic:`** The idea is that the fit should prefer broader ranges so we use the inverse of $\mathcal{X}^2$ and multiply it with the ratio of the active time slices to the total time slices available.

  In three-point functions there is hardly any single point with a huge error, the errors are quite evenly distributed. We make sure that the fit function is compatible with the data points by taking an inverse of uncorrelated $\mathcal{X}^2$ of the function which is $|\frac{Fn(t)-y(t)}{\delta Fn(t)-\delta y(t)}|$ (where Fn is the function y's are the data points and t is the timeslice)

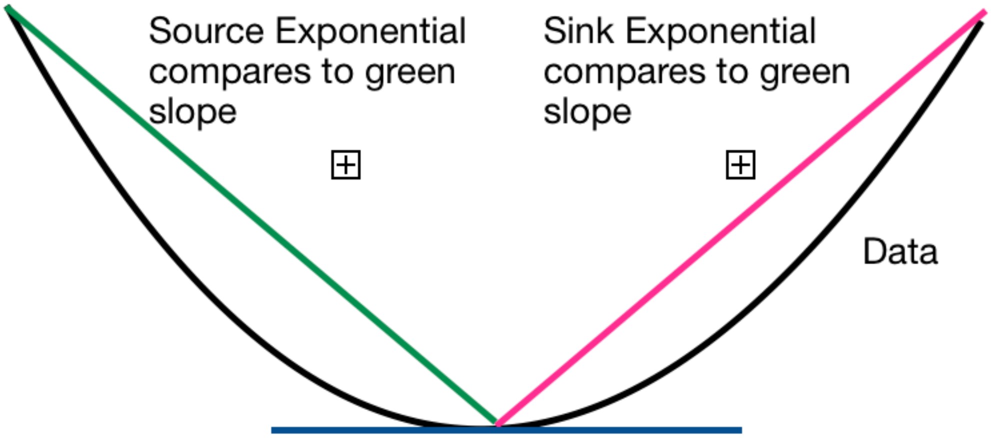
Source Exponential compares to green slope
⊞
Sink Exponential compares to green slope
⊞
Data

Figure 1: Illustration of the fit-quality. The black curve is the data. The green and red lines show the slopes from the midpoint to the ends of the graph

and multiplying the fit-quality. By quadrature, the error in the fit function is,

$$\delta Fn = \delta F + (\delta F_i - \delta(\Delta E_i) F_i t) e^{-\Delta E_i t} + (\delta F_f - \delta(\Delta E_f) F_f (\Delta t - t)) e^{-\Delta E_f (\Delta t - t)}$$

This makes sure that the fit is somewhat compatible to all the data points even if they are outside the fit window. To take care that the fits where the parameters are highly correlated are not present, this quality has a hard cutoff on the parameter correlations -0.55 and 0.65.

The exponentials, if present, should be precise and the value should not be small to avoid $F + F_i e^{-\Delta E_i t}$ or $F + F_f e^{-\Delta E_f (\Delta t - t)} \rightarrow$ constant. Thus, we multiply the function by $|\frac{\Delta E_i}{\delta \Delta E_i}|$ and $|\frac{F_i}{\delta F_i}|$. This factor prefers towards precise values and ΔE_i that is not very small. We do the same at the sink, multiply by $|\frac{\Delta E_f}{\delta \Delta E_f}|$ and $|\frac{F_f}{\delta F_f}|$. But this creates a bias towards the fits with an exponential, so we have to come up with a normalization. If we expand the source and sink exponentials when the value $|\Delta E_i t| < 1$,

$$F_i e^{-\Delta E_i t} = F_i - F_i \Delta E_i F_i t + ...$$

This shows that slope of the data at the source would give a good estimate of the curvature of the graph and thus the value of the exponential. This can be used as the normalization. We can find the slope of the data from the midpoint to the source operator time and from the midpoint to the sink time-slice and this would be a good normalization for the source and the sink (assuming that the plateau is at the midpoint) making sure that the exponential in the fit compares to the curvature in the data as shown in figure **??** the green and the pink slopes are the normalizations for $|\frac{\Delta E_i \times F_i}{\delta \Delta E_i \times \delta F_i}|$ and $|\frac{\Delta E_f \times F_f}{\delta \Delta E_f \times \delta F_f}|$ respectively.

To summarize, we multiply by $|\frac{\Delta E_i \times F_i}{\delta \Delta E_i \times \delta F_i}|$ if there is a source exponential in the fit function if not we multiply by $|\frac{y(0) - y(mid-point)}{x(0) - x(mid-point)}|$. If there is a sink exponential multiply by $|\frac{\Delta E_f \times F_f}{\delta \Delta E_f \times \delta F_f}|$. If not by, $|\frac{y(end) - y(mid-point)}{x(end) - x(mid-point)}|$.

We need to have good estimates and filter out fits that not reasonable, especially the case when the factor multiplying the exponential exactly cancels the constant term and the exponential $\rightarrow 0$. This is taken care of by limiting the value of the exponential. The value of the exponential has an upper and a lower limit. The lower limit is defined by the user and should be greater than 0 (ideally $\sim$ excited level) which is straightforward to see. The exponential is dominated by the first excited state but it is an effective term that sums the contribution of all the energy levels thus the upper limit can be the first threshold (I have used 5× the energy of the first encountered threshold, the user can change it).

6 Application on data

6.1 $\pi\gamma \to \rho$ Data

Here we compare the different fit qualities the first one is using,

- the traditional χ^2/DOF
- the generic function as defined above

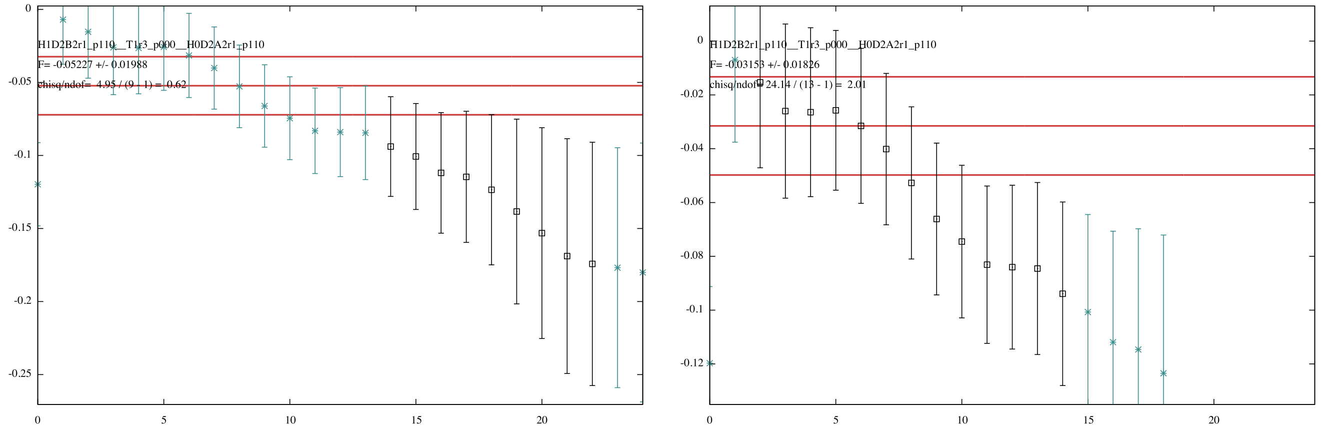

Figure 2: `H1D2B2r1_p110__T1r3_p000__H0D2A2r1_p110` fitted with `chisq_per_dof and generic`

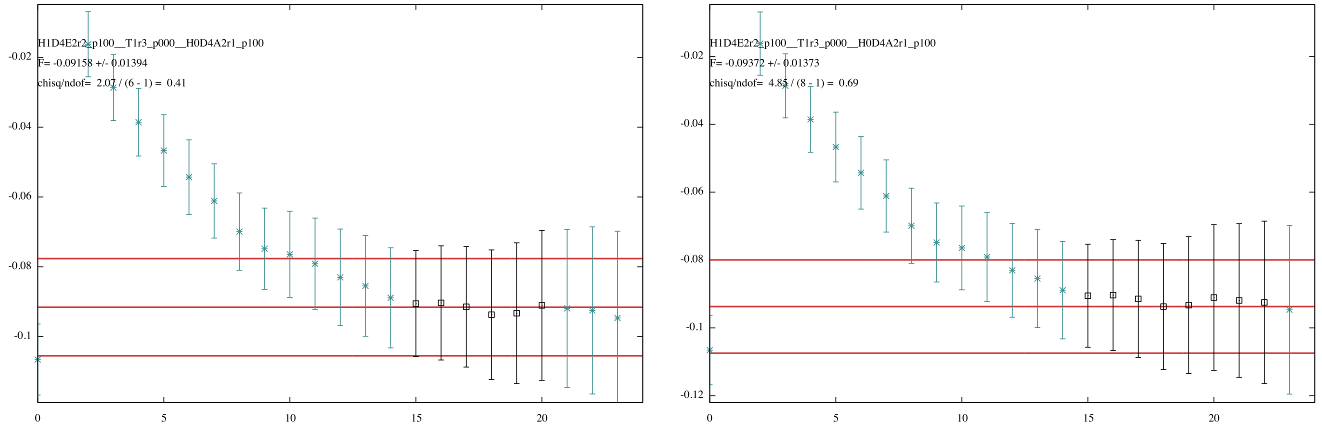

Figure 3: `H1D4E2r2_p100__T1r3_p000__H0D4A2r1_p100` fitted with `chisq_per_dof and generic`

The fits look pretty reasonable. Except the first figure where I think the fit function should be a little higher.

6.2 Fabricated Data

I tested the fit quality on fabricated data with $\Delta t = 24$ and the parameters $F = 1$, $F_i = 0.49$, $F_f = -0.89$, $E_i = 0.44$, $E_f = 0.38$. The parameter correlation is 0.2 for all off-diagonals. Error on the data is $0.1 \times y(x)$.

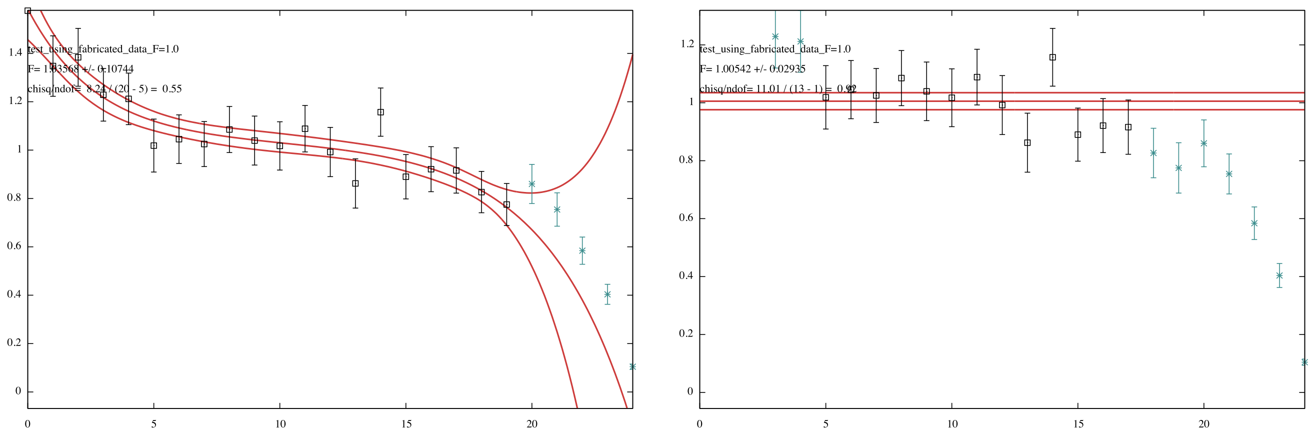

Figure 4: The fabricated data fitted with `chisq_per_dof and generic`

References

[1] *Phys.Rev. D91, 114501 (2015)*, C. J. Shultz, J. J. Dudek, and R. G. Edwards

[2] *http://inspirehep.net/record/1258345/*, Fred James and Matthias Winkler

[3] *10.1103/PhysRevD.51.3745*, C. Michael, A. McKerrell