Poor Man's Advanced Analysis Techniques

New Horizons in Lattice Field Theory

13-27 March, 2013

International Institute of Physics

Natal, Brazil

André Walker-Loud

Advanced Fitting Methods

"The fit of data by a sum of exponentials and in particular by real exponents may be very badly conditioned"

Lanczos 1956
Wiscombe & Evans 1977
Ruhe & Wedin 1980

Advanced Fitting Methods

"The fit of data by a sum of exponentials and in particular by real exponents may be very badly conditioned"

Lanczos 1956
Wiscombe & Evans 1977
Ruhe & Wedin 1980

Everything I will tell you today are things I developed working with Kostas Orginos

Advanced Fitting Methods

"The fit of data by a sum of exponentials and in particular by real exponents may be very badly conditioned"

Lanczos 1956
Wiscombe & Evans 1977
Ruhe & Wedin 1980

Everything I will tell you today are things I developed working with Kostas Orginos

I will not offer proofs of any of the methods but rather introduce them practically for you to implement the proofs are left as an exercise for you

Advanced Fitting Methods

Designed for a-symmetric matrices of correlation functions
(poor man = a-symmetric)

- Variational Projection

- Input-parameter free multi-exponential fitting

- Generalized Pencil of Function (GPOF)

- Matrix Prony

Variational Projection

Golub & Pereyra Inverse Problems 19 (2003)
arXiv:0907.0529

Very simple idea:
if a fit function has mixed linear/non-linear dependence on the fit parameters, one does not need to perform a numerical minimization on all the parameters - one can first perform a linear least squares on the linear parameters, solving as a function of the non-linear ones

Variational Projection

Golub & Pereyra Inverse Problems 19 (2003)
arXiv:0907.0529

Very simple idea: first perform linear-least squares in lattice QFT, correlation functions fit with

$$\chi^2 = \sum_{t,t'} \left[y(t) - f(\lambda, t)\right] C^{-1}_{t,t'} \left[y(t') - f(\lambda, t')\right]$$

$$f(\lambda, t) = \sum_n Z_n \lambda_n^t \qquad \lambda_n = e^{-E_n} \qquad \lambda = \{Z_n, \lambda_n\}$$

$$\frac{\partial \chi^2}{\partial Z_n} = 0$$

(repeated indices summed over)

$$= \left[-\lambda_n^t C^{-1}_{t,t'}(y(t') - Z_m \lambda_m^{t'}) - (y(t) - Z_m \lambda_m^t) C^{-1}_{t,t'} \lambda_n^{t'}\right]$$

Variational Projection

Golub & Pereyra Inverse Problems 19 (2003)
arXiv:0907.0529

Very simple idea: first perform linear-least squares

symmetry in $t \leftrightarrow t'$

$$\frac{\partial \chi^2}{\partial Z_n} = 0$$

$$= \left[-\lambda_n^t C_{t,t'}^{-1} (y(t') - Z_m \lambda_m^{t'}) - (y(t) - Z_m \lambda_m^t) C_{t,t'}^{-1} \lambda_n^{t'} \right]$$

$$\Rightarrow \quad Z_m \lambda_m^t C_{t,t'}^{-1} \lambda_n^{t'} = y(t) C_{t,t'}^{-1} \lambda_n^{t'}$$

Variational Projection

Golub & Pereyra Inverse Problems 19 (2003)
arXiv:0907.0529

Very simple idea: first perform linear-least squares

symmetry in $t \leftrightarrow t'$

$$\frac{\partial \chi^2}{\partial Z_n} = 0$$

$$= \left[-\lambda_n^t C_{t,t'}^{-1} (y(t') - Z_m \lambda_m^{t'}) - (y(t) - Z_m \lambda_m^t) C_{t,t'}^{-1} \lambda_n^{t'} \right]$$

$$\Rightarrow \quad Z_m \lambda_m^t C_{t,t'}^{-1} \lambda_n^{t'} = y(t) C_{t,t'}^{-1} \lambda_n^{t'}$$

$$Z_m \Lambda_{m,n} = y(t) C_{t,t'}^{-1} \lambda_n^{t'} \qquad \Lambda_{m,n} = \lambda_m^t C_{t,t'}^{-1} \lambda_n^{t'}$$

Variational Projection

Golub & Pereyra Inverse Problems 19 (2003)
arXiv:0907.0529

Very simple idea: first perform linear-least squares

$$\frac{\partial \chi^2}{\partial Z_n} = 0 \quad \Rightarrow \quad Z_m = \Lambda^{-1}_{m,n}\, y(t)\, C^{-1}_{t,t'}\, \lambda^{t'}_n$$

we have solved for the overlap factors as functions of the eigenvalues

plug these solutions back into chi^2 and perform numerical minimization on just the eigenvalues

Variational Projection

Golub & Pereyra Inverse Problems 19 (2003)
arXiv:0907.0529

Very simple idea: first perform linear-least squares

$$\chi^2 = \sum_{t,t'} \left[y(t) - f(\lambda, t)\right] C^{-1}_{t,t'} \left[y(t') - f(\lambda, t')\right]$$

$$f(\lambda, t) = \sum_n Z_n \lambda_n^t \qquad \lambda_n = e^{-E_n}$$

$$Z_m = \Lambda^{-1}_{m,n}\, y(t)\, C^{-1}_{t,t'}\, \lambda_n^{t'} \qquad \Lambda_{m,n} = \lambda_m^t C^{-1}_{t,t'} \lambda_n^{t'}$$

Variational Projection

Golub & Pereyra Inverse Problems 19 (2003)
arXiv:0907.0529

Very simple idea: first perform linear-least squares

$$\chi^2 = \sum_{t,t'} \left[y(t) - f(\lambda, t)\right] C^{-1}_{t,t'} \left[y(t') - f(\lambda, t')\right]$$

$$f(\lambda, t) = \sum_n Z_n \lambda_n^t \qquad \lambda_n = e^{-E_n}$$

$$Z_m = \Lambda^{-1}_{m,n}\, y(t)\, C^{-1}_{t,t'}\, \lambda_n^{t'} \qquad \Lambda_{m,n} = \lambda_m^t C^{-1}_{t,t'} \lambda_n^{t'}$$

When counting the degrees of freedom in the fit DON'T forget to count the overlap factors you have also determined!

Variational Projection

Golub & Pereyra Inverse Problems 19 (2003)
arXiv:0907.0529

Very simple idea: first perform linear-least squares

$$\chi^2 = \sum_{t,t'} \left[y(t) - f(\lambda, t)\right] C^{-1}_{t,t'} \left[y(t') - f(\lambda, t')\right]$$

$$f(\lambda, t) = \sum_n Z_n \lambda_n^t \qquad \lambda_n = e^{-E_n}$$

$$Z_m = \Lambda^{-1}_{m,n}\, y(t)\, C^{-1}_{t,t'}\, \lambda_n^{t'} \qquad \Lambda_{m,n} = \lambda_m^t C^{-1}_{t,t'} \lambda_n^{t'}$$

When counting the degrees of freedom in the fit DON'T forget to count the overlap factors you have also determined!

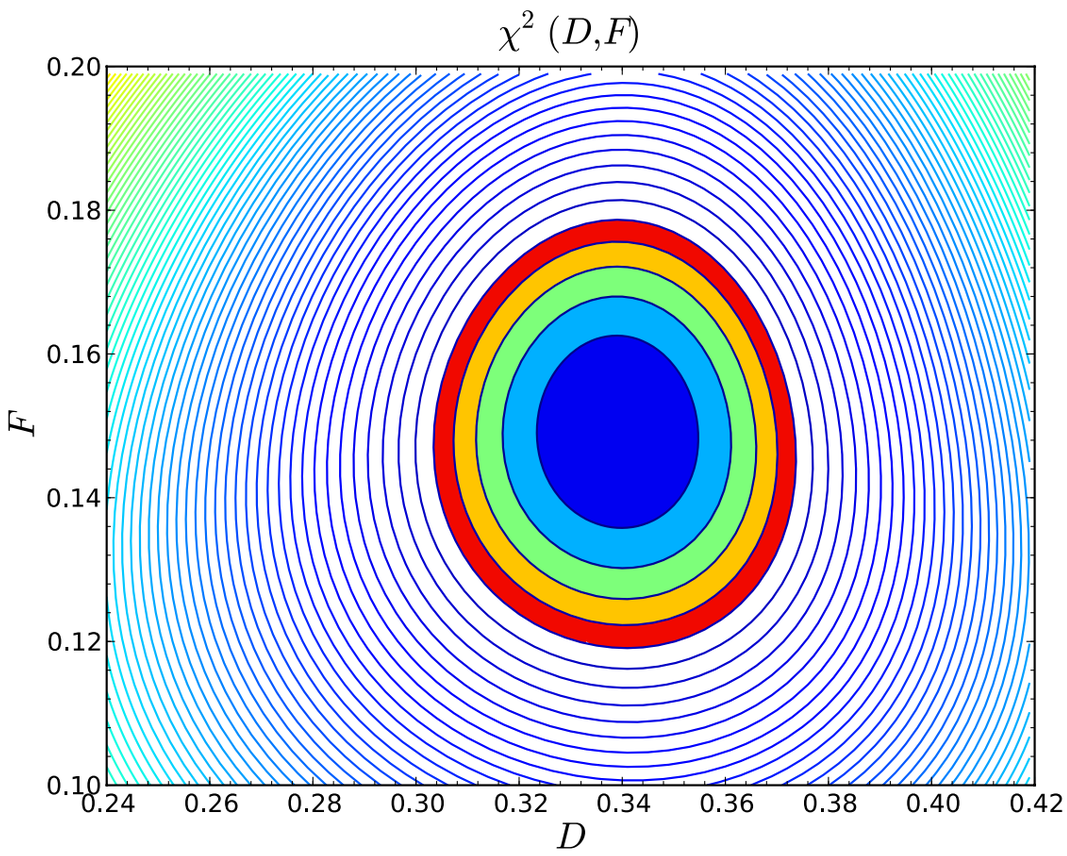
readily extend this to matrix of correlation functions

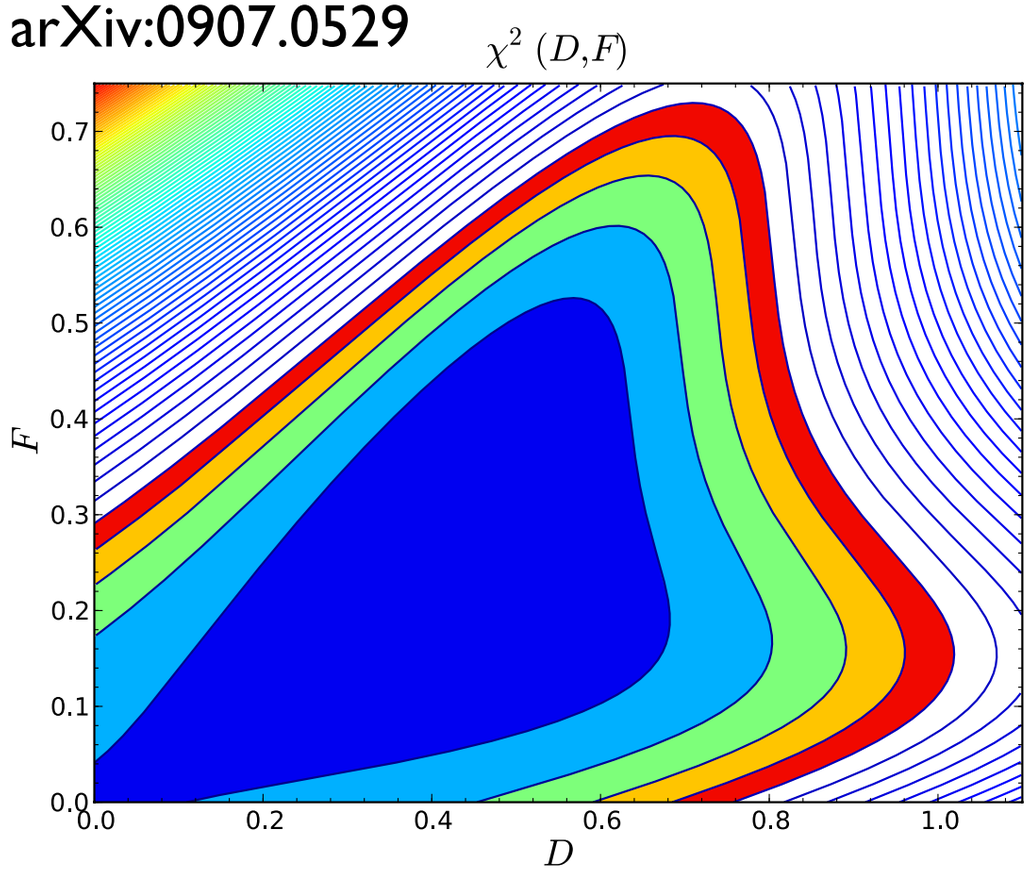
Variational Projection

Golub & Pereyra Inverse Problems 19 (2003)
arXiv:0907.0529

can also apply the same idea to chiral extrapolation formulae which usually have mixed linear/non-linear dependence on LECs

E. Jenkins, A. Manohar, J. Negele and AWL
arXiv:0907.0529





NLO: 11 LECs

NNLO: 30 LECs

Fit to Octet and Decuplet baryon mass results with SU(3) Baryon Chiral Perturbation Theory

Input Parameter Free Multi-Exp Fits

De Groen & De Moor
Comp. Appl. Math. 20 (1987)

"The fit of a sum of exponentials to noisy data"
(Note: typos in paper)

Input Parameter Free Multi-Exp Fits

De Groen & De Moor
Comp. Appl. Math. 20 (1987)

Given the outputs of a non-degenerate n-dimensional linear system

$$y_t = c^T x_t \qquad c_i \neq 0, i = 1, \ldots, n$$

$$\{x_t, c\} \in \mathbb{R}^n$$

$$x_{t+1} = T x_t \qquad x_0^T = (1, \ldots, 1)$$

$$y_t = c^T T^t x_0$$

Input Parameter Free Multi-Exp Fits

De Groen & De Moor
Comp. Appl. Math. 20 (1987)

Given the outputs of a non-degenerate n-dimensional linear system

$$y_t = c^T x_t \qquad c_i \neq 0, i = 1, \ldots, n$$

$$\{x_t, c\} \in \mathbb{R}^n$$

$$x_{t+1} = T x_t \qquad x_0^T = (1, \ldots, 1)$$

$$y_t = c^T T^t x_0$$

construct the p x q Hankel Matrix with p,q > n

$$H_{ij} = y_{i+j-1}$$

$$H = \begin{pmatrix} y_0 & \cdots & y_{q-1} \\ \vdots & \ddots & \vdots \\ y_{p-1} & \cdots & y_{q+p-2} \end{pmatrix}$$

Input Parameter Free Multi-Exp Fits

De Groen & De Moor
Comp. Appl. Math. 20 (1987)

The Hankel Matrix can be factorized

$$H = \begin{pmatrix} y_0 & \cdots & y_{q-1} \\ \vdots & \ddots & \vdots \\ y_{p-1} & \cdots & y_{q+p-2} \end{pmatrix}$$

$$y_t = c^T T^t x_0$$

$$H = FG = \begin{pmatrix} c^T \\ c^T T \\ \vdots \\ c^T T^{p-1} \end{pmatrix} \begin{pmatrix} x_0 & Tx_0 & \cdots & T^{q-1}x_0 \end{pmatrix}$$

Input Parameter Free Multi-Exp Fits

De Groen & De Moor
Comp. Appl. Math. 20 (1987)

The factorization matrices can be inverted $H = FG$

$F : \mathbb{R}^n \rightarrow \mathbb{R}^p \qquad G : \mathbb{R}^q \rightarrow \mathbb{R}^n \qquad (H \text{ is } p \times q \text{ matrix})$

$$\mathbb{1}^n = \tilde{F}^{-1} H \tilde{G}^{-1}$$

$$\{\tilde{F}^{-1}, \tilde{G}^{-1}\} = \text{pseudo-inverse}\{F, G\}$$

Input Parameter Free Multi-Exp Fits

De Groen & De Moor
Comp. Appl. Math. 20 (1987)

The shifted Hankel Matrix can be factorized

$$H_{ij} = y_{i+j-1} \qquad H^{+}_{ij} = y_{i+j} \qquad y_t = c^T T^t x_0$$

$$H = FG \qquad H^{+} = FTG$$

Input Parameter Free Multi-Exp Fits

De Groen & De Moor
Comp. Appl. Math. 20 (1987)

The shifted Hankel Matrix can be factorized

$$H_{ij} = y_{i+j-1} \qquad H^{+}_{ij} = y_{i+j} \qquad y_t = c^T T^t x_0$$

$$H = FG \qquad H^{+} = FTG$$

We can construct a matrix similar to T

$$T_s = \tilde{F}^{-1} H^{+} \tilde{G}^{-1} = \tilde{F}^{-1} FTG\tilde{G}^{-1}$$

Input Parameter Free Multi-Exp Fits

De Groen & De Moor
Comp. Appl. Math. 20 (1987)

The shifted Hankel Matrix can be factorized

$$H_{ij} = y_{i+j-1} \qquad H^{+}_{ij} = y_{i+j} \qquad y_t = c^T T^t x_0$$

$$H = FG \qquad H^{+} = FTG$$

We can construct a matrix similar to T

$$T_s = \tilde{F}^{-1} H^{+} \tilde{G}^{-1} = \tilde{F}^{-1} FTG\tilde{G}^{-1}$$

How do we determine $\{F, G\}$?

Input Parameter Free Multi-Exp Fits

De Groen & De Moor
Comp. Appl. Math. 20 (1987)

The shifted Hankel Matrix can be factorized

$$H_{ij} = y_{i+j-1} \qquad H^{+}_{ij} = y_{i+j} \qquad y_t = c^T T^t x_0$$

$$H = FG \qquad H^{+} = FTG$$

- First - pick a window in time of interest you want to analyze, t_i, t_f
- One can (should) shift with $\Delta t = \tau > 1$

$$H^{\tau}_{ij} = y_{i+j-1+\tau}$$

Input Parameter Free Multi-Exp Fits

De Groen & De Moor
Comp. Appl. Math. 20 (1987)

The shifted Hankel Matrix can be factorized

$$H_{ij} = y_{i+j-1} \qquad H^{+}_{ij} = y_{i+j} \qquad y_t = c^T T^t x_0$$

$$H = FG \qquad H^{+} = FTG$$

How do we determine $\{F, G\}$?

- First - pick a window in time of interest you want to analyze, t_i, t_f
- One can (should) shift with $\Delta t = \tau > 1$

$$H^{\tau}_{ij} = y_{i+j-1+\tau}$$

Input Parameter Free Multi-Exp Fits

De Groen & De Moor
Comp. Appl. Math. 20 (1987)

- First - pick a window in time of interest you want to analyze, t_i, t_f
- One can (should) shift with $\Delta t = \tau > 1$

$$H^{\tau}_{ij} = y_{i+j-1+\tau}$$

$$H = \begin{pmatrix} y_0 & \cdots & y_{q-1} \\ \vdots & \ddots & \vdots \\ y_{p-1} & \cdots & y_{q+p-2} \end{pmatrix}$$

```python
# written in python language
import numpy as np

corr = # data array in form array([ncfg,nt,data[ncfg,nt]])
ncfg = corr.shape[0]
nt = corr.shape[1]
dt = 4 #shift you chose

h_tmp = np.zeros([nt/2 -dt, nt/2 -dt])
h_tmp_shift = np.zeros_like(h_tmp)
y_t = np.mean(corr,axis=0)
for t1 in range(nt/2 - dt):
    for t2 in range(nt/2 - dt):
        h_tmp[t1,t2] = y_t[t1+t2]
        h_tmp_shift[t1,t2] = y_t[t1+t2+dt]

hankel = h_tmp[t0:tf:dt,0:tf-t0:dt]
hankel_shift = h_tmp_shift[t0:tf:dt,0:tf-t0:dt]
```

Input Parameter Free Multi-Exp Fits

De Groen & De Moor
Comp. Appl. Math. 20 (1987)

Perform a singular value decomposition on H

$$H = U\Sigma V^T \qquad \Sigma = \mathrm{diag}(\mathrm{singular\ values})$$

$$T_s = \Sigma^{-1/2}\, U^T\, H^+\, V\, \Sigma^{-1/2} \qquad H^+ = FTG$$

Input Parameter Free Multi-Exp Fits

De Groen & De Moor
Comp. Appl. Math. 20 (1987)

Perform a singular value decomposition on H

$$H = U\Sigma V^T \qquad \Sigma = \mathrm{diag}(\text{singular values})$$

$$T_s = \Sigma^{-1/2}\, U^T\, H^+\, V\, \Sigma^{-1/2} \qquad H^+ = FTG$$

One very important point: need to truncate the singular values to a reasonable number

- If you desire just the ground state - chose range of time suitably (contamination from just a single state) and take two singular values
- For determining 2 states (gs + 1st excited) pick 3 singular values etc.
- study the singular values to see how much information you can get from correlation function

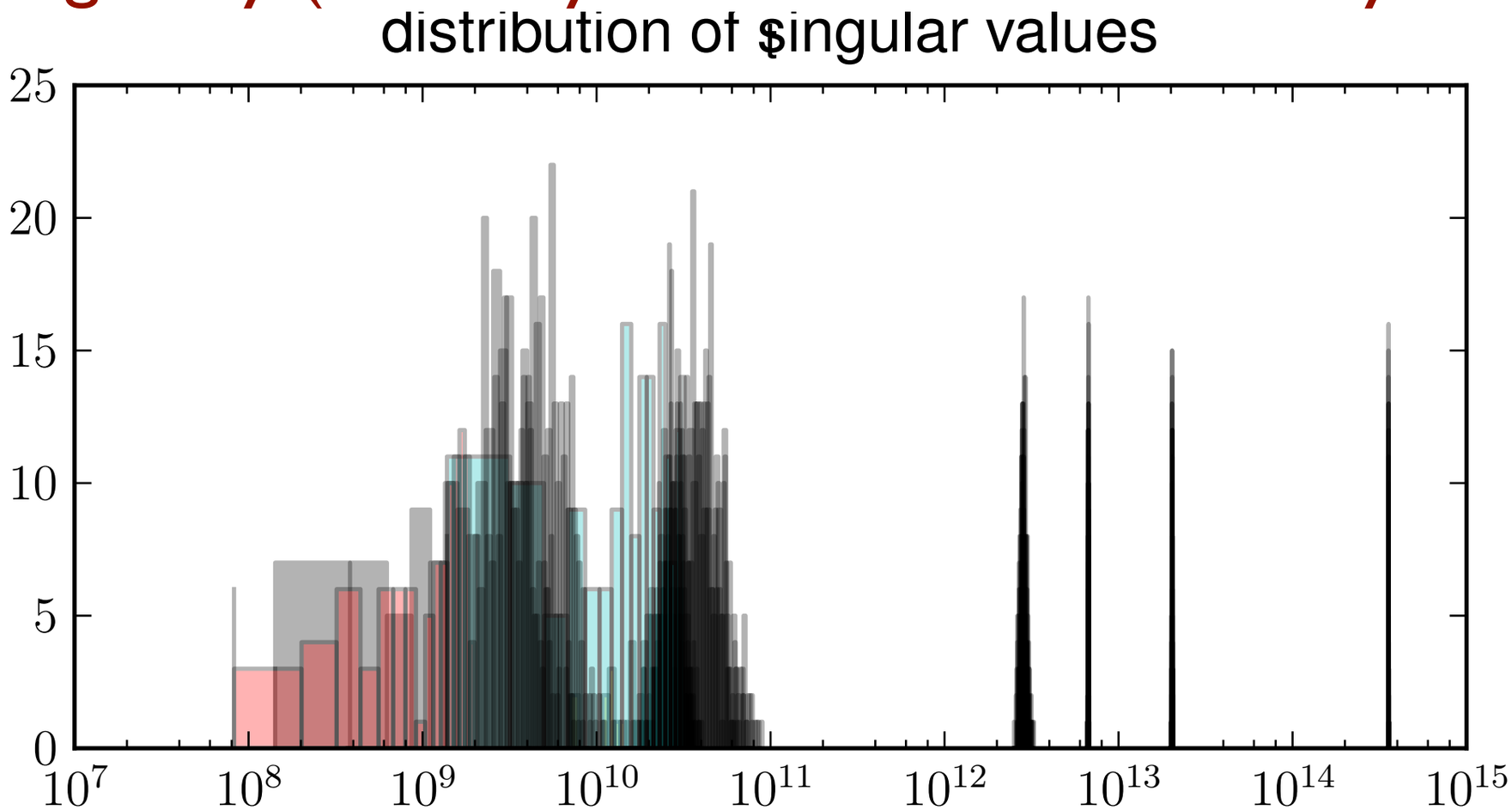
Input Parameter Free Multi-Exp Fits

De Groen & De Moor
Comp. Appl. Math. 20 (1987)

Perform a singular value decomposition on H

$$H = U\Sigma V^T \qquad \Sigma = \mathrm{diag}(\text{singular values})$$

Don't get greedy (don't try and determine too many excited states)



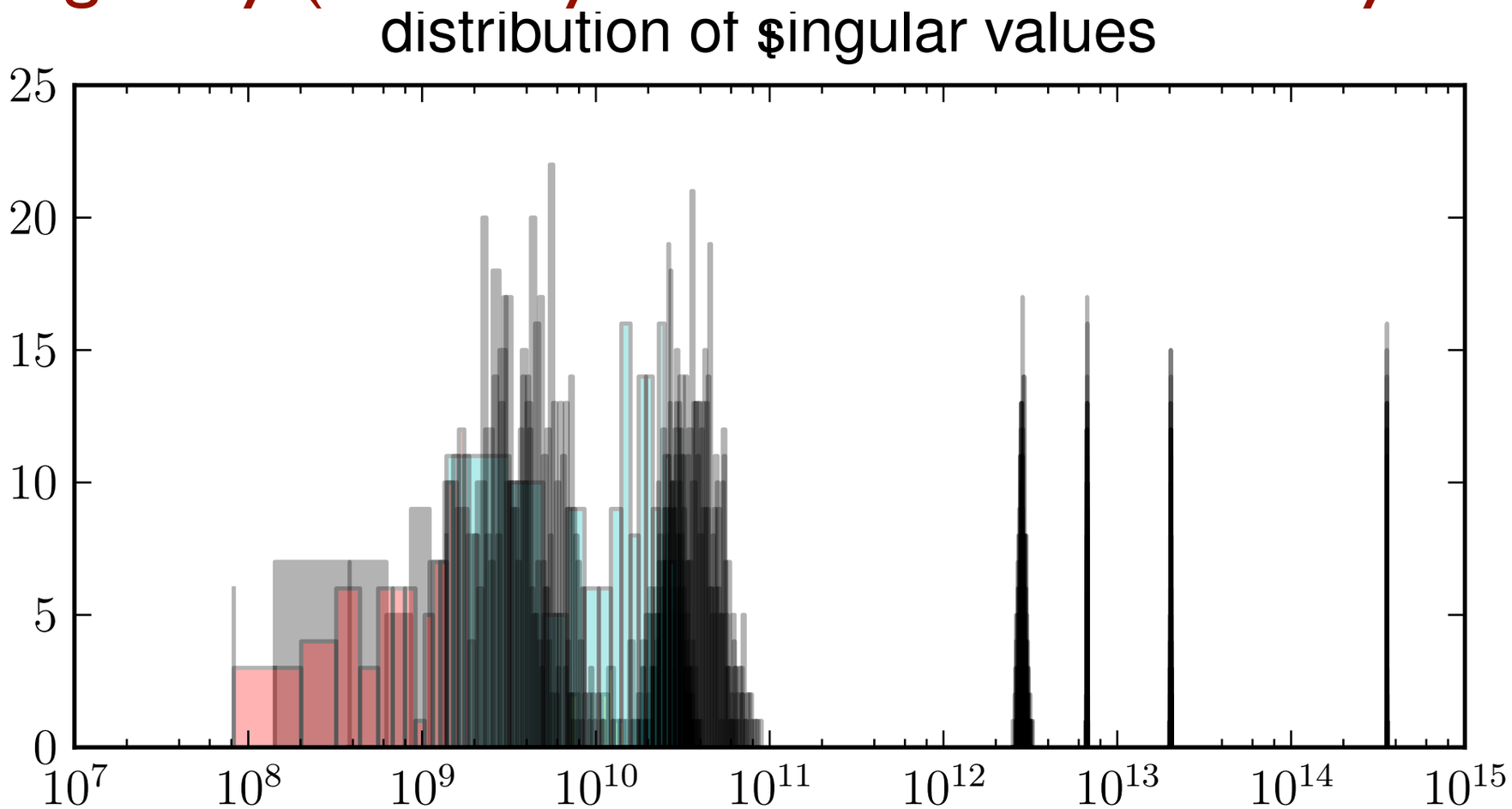
Input Parameter Free Multi-Exp Fits

De Groen & De Moor
Comp. Appl. Math. 20 (1987)

Perform a singular value decomposition on H

$$H = U\Sigma V^T \qquad \Sigma = \mathrm{diag}(\text{singular values})$$

Don't get greedy (don't try and determine too many excited states)



in this case, at most, one can determine 4 energies and only 3 with confidence (the 4th absorbs the "slop")

Input Parameter Free Multi-Exp Fits

De Groen & De Moor
Comp. Appl. Math. 20 (1987)

Perform a singular value decomposition on H

$$H = U\Sigma V^T \qquad \Sigma = \mathrm{diag}(\mathrm{singular\ values})$$

take n_s largest singular values

$$T_s = \Sigma^{-1/2}[0:n_s]\, U^T[0:ns,:]\, H^+\, V[:,0:n_s]\, \Sigma^{-1/2}[0:n_s]$$

$$H^+ = FTG$$

Input Parameter Free Multi-Exp Fits

De Groen & De Moor
Comp. Appl. Math. 20 (1987)

Perform a singular value decomposition on H

$$H = U\Sigma V^T \qquad \Sigma = \mathrm{diag}(\mathrm{singular\ values})$$

take n_s largest singular values

$$T_s = \Sigma^{-1/2}[0:n_s]\, U^T[0:ns,:]\, H^+\, V[:,0:n_s]\, \Sigma^{-1/2}[0:n_s]$$

$$\lambda_n, O_n = \mathrm{eig}(T_s) \qquad H^+ = FTG$$

$$E_n = -\ln(\lambda_n)$$

Input Parameter Free Multi-Exp Fits

De Groen & De Moor
Comp. Appl. Math. 20 (1987)

Perform a singular value decomposition on H

$$H = U\Sigma V^T \qquad \Sigma = \mathrm{diag}(\text{singular values})$$

take n_s largest singular values

$$T_s = \Sigma^{-1/2}[0:n_s]\, U^T[0:ns,:]\, H^+\, V[:,0:n_s]\, \Sigma^{-1/2}[0:n_s]$$

$$\lambda_n, O_n = \mathrm{eig}(T_s) \qquad H^+ = FTG$$

$$E_n = -\ln(\lambda_n)$$

Now we have the eigen-energies - how do we get the overlap factors?

Input Parameter Free Multi-Exp Fits

De Groen & De Moor
Comp. Appl. Math. 20 (1987)

Perform a singular value decomposition on H

$$H = U\Sigma V^T \qquad \Sigma = \mathrm{diag}(\text{singular values})$$

take n_s largest singular values

$$T_s = \Sigma^{-1/2}[0:n_s]\, U^T[0:ns,:]\, H^+\, V[:,0:n_s]\, \Sigma^{-1/2}[0:n_s]$$

$$\lambda_n, O_n = \mathrm{eig}(T_s) \qquad H^+ = FTG$$

$$E_n = -\ln(\lambda_n)$$

Now we have the eigen-energies - how do we get the overlap factors?

See VarPro!

Input Parameter Free Multi-Exp Fits

De Groen & De Moor
Comp. Appl. Math. 20 (1987)

$$H_{ij} = y_{i+j-1} \quad H^+_{ij} = y_{i+j} \qquad H = U\Sigma V^T \qquad n_s$$

$$T_s = \Sigma^{-1/2}[0:n_s]\, U^T[0:ns,:]\, H^+\, V[:,0:n_s]\, \Sigma^{-1/2}[0:n_s]$$

$$\lambda_n, O_n = \mathrm{eig}(T_s) \qquad E_n = -\ln(\lambda_n)$$

$$Z_m = \Lambda^{-1}_{m,n}\, y(t)\, C^{-1}_{t,t'}\, \lambda^{t'}_n \qquad \Lambda_{m,n} = \lambda^t_m C^{-1}_{t,t'} \lambda^{t'}_n$$

Use these values to seed the multi-exponential fit!

GPOF

K. Orginos Latt2010 (unpublished) but on web
C. Aubin & K. Orginos Latt2011 (on PoS)
C. Aubin, K. Orginos & AWL arXiv:1xxx.xxxx

This is a generalization of the previous idea (Hankel Matrix) extended to a vector or matrix of correlators.

For the sake of time - I refer you to the above references

Matrix Prony

arXiv:1301.1114
arXiv:0905.0466

Prony = some guys name

Matrix Prony

arXiv:1301.1114
arXiv:0905.0466

Prony = some guys name

G. R. de Prony Journal de l'cole Polytechnique, volume 1, cahier 22, 24-76 (1795)

Matrix Prony

arXiv:1301.1114
arXiv:0905.0466

Prony = some guys name

G. R. de Prony Journal de l'cole Polytechnique, volume 1, cahier 22, 24-76 (1795)

Matrix = matrix

Matrix Prony

arXiv:1301.1114
arXiv:0905.0466

Start with a vector of correlation functions

perhaps a single src and multiple sinks

$$y(t) = \begin{pmatrix} y_1(t) \\ y_2(t) \\ \vdots \\ y_N(t) \end{pmatrix}$$

Matrix Prony

arXiv:1301.1114
arXiv:0905.0466

Start with a vector of correlation functions
perhaps a single src and multiple sinks

$$y(t) = \begin{pmatrix} y_1(t) \\ y_2(t) \\ \vdots \\ y_N(t) \end{pmatrix}$$

We would like to construct an operator such that

$$y(t+\tau) = \hat{T}(\tau) y(t)$$

Matrix Prony

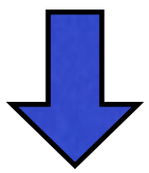
arXiv:1301.1114
arXiv:0905.0466

$$y(t+\tau) = \hat{T}(\tau)y(t)$$

Matrix Prony

arXiv:1301.1114
arXiv:0905.0466

$$y(t+\tau) = \hat{T}(\tau) y(t)$$

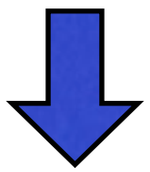


$$y(t+\tau) y^T(t) = \hat{T}(\tau) y(t) y^T(t)$$

Matrix Prony

arXiv:1301.1114
arXiv:0905.0466

$$y(t+\tau) = \hat{T}(\tau) y(t)$$



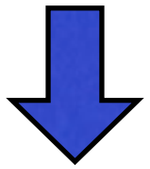
$$y(t+\tau) y^T(t) = \hat{T}(\tau) y(t) y^T(t)$$

Transpose

Matrix Prony

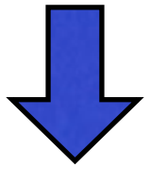
arXiv:1301.1114
arXiv:0905.0466

$$y(t+\tau) = \hat{T}(\tau) y(t)$$



$$y(t+\tau) y^T(t) = \hat{T}(\tau) y(t) y^T(t)$$

Transpose

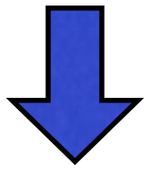


$$\hat{M}(\tau) y(t+\tau) y^T(t) = \hat{V}(\tau) y(t) y^T(t) \qquad \hat{T} = \hat{M}^{-1} \hat{V}$$

Matrix Prony

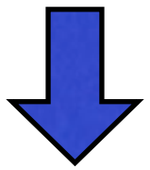
arXiv:1301.1114
arXiv:0905.0466

$$y(t+\tau) = \hat{T}(\tau) y(t)$$



$$y(t+\tau) y^T(t) = \hat{T}(\tau) y(t) y^T(t)$$

Transpose



$$\hat{M}(\tau) y(t+\tau) y^T(t) = \hat{V}(\tau) y(t) y^T(t) \qquad \hat{T} = \hat{M}^{-1} \hat{V}$$

Nothing special about t

Matrix Prony

arXiv:1301.1114
arXiv:0905.0466

$$y(t+\tau) = \hat{T}(\tau) y(t)$$

$$\hat{M}(\tau) \sum_{t_0}^{t_0+\Delta t} y(t+\tau) y^T(t) = \hat{V}(\tau) \sum_{t_0}^{t_0+\Delta t} y(t) y^T(t)$$

We have assumed T (and M and V) are independent of t but this is precisely what we desire from a "transfer matrix"

Matrix Prony

arXiv:1301.1114
arXiv:0905.0466

$$\hat{M}(\tau) \sum_{t_0}^{t_0+\Delta t} y(t+\tau)y^T(t) = \hat{V}(\tau) \sum_{t_0}^{t_0+\Delta t} y(t)y^T(t)$$

a solution

$$\hat{M}(\tau) = \left[\sum_{t_0}^{t_0+\Delta t} y(t+\tau)y^T(t)\right]^{-1}, \; \hat{V}(\tau) = \hat{V} = \left[\sum_{t_0}^{t_0+\Delta t} y(t)y^T(t)\right]^{-1}$$

must sum over sufficient number of time slices to make matrices "full rank" for two-components, must sum over at least two time slices

Matrix Prony

arXiv:1301.1114
arXiv:0905.0466

$$\hat{M}(\tau) \sum_{t_0}^{t_0+\Delta t} y(t+\tau)y^T(t) = \hat{V}(\tau) \sum_{t_0}^{t_0+\Delta t} y(t)y^T(t)$$

a solution

$$\hat{M}(\tau) = \left[\sum_{t_0}^{t_0+\Delta t} y(t+\tau)y^T(t)\right]^{-1}, \; \hat{V}(\tau) = \hat{V} = \left[\sum_{t_0}^{t_0+\Delta t} y(t)y^T(t)\right]^{-1}$$

most robust results come from maximizing Δt

this requires that over a large range of time our ansatz is satisfied - only N states contribute in $t_0 \to t_0 + \Delta t$

Matrix Prony

arXiv:1301.1114
arXiv:0905.0466

$$y(t+\tau) = \hat{M}^{-1}(\tau)\hat{V}y(t) = \hat{T}(\tau)y(t)$$

$$\hat{M}(\tau) = \left[\sum_{t_0}^{t_0+\Delta t} y(t+\tau)y^T(t)\right]^{-1}, \; \hat{V}(\tau) = \hat{V} = \left[\sum_{t_0}^{t_0+\Delta t} y(t)y^T(t)\right]^{-1}$$

given M and V one then solves the eigenvalue equation

$$\hat{T}(\tau)q_n = (\lambda_n)^\tau q_n \qquad \lambda_n = e^{-E_n} \qquad \hat{T}(\tau) = \hat{M}^{-1}(\tau)\hat{V}$$

given λ_n check to see if ansatz is satisfied - over range of $t_0 \to t_0 + \Delta t$ there should be no significant evidence of excited state contamination

Matrix Prony

arXiv:1301.1114
arXiv:0905.0466

in Python

```
''' MATRIX PRONY '''
import numpy as np
from scipy import linalg as spla
def matrix_prony(corr,ti,tau,td=1):
    corr = np.array(corr)
    n_cfg,n_t,n_corr = corr.shape
    if tau < n_corr:
        print 'Matrix Prony Error:'
        print '  must chose at least %d time slices' %n_corr
        sys.exit(-1)
    y_avg = np.mean(corr,axis=0)
    Minvt = np.array(map(np.outer,np.roll(y_avg,-td,0),y_avg))
    Vinvt = np.array(map(np.outer,y_avg,y_avg))
    Minv = np.sum(Minvt[ti:ti+tau],axis=0)
    Vinv = np.sum(Vinvt[ti:ti+tau],axis=0)
    Transfer = np.dot(Minv,spla.inv(Vinv))
    le,Oe = spla.eig(Transfer)
    ind = np.argsort(le); indr = ind[::-1]

    return le, Oe, indr
```

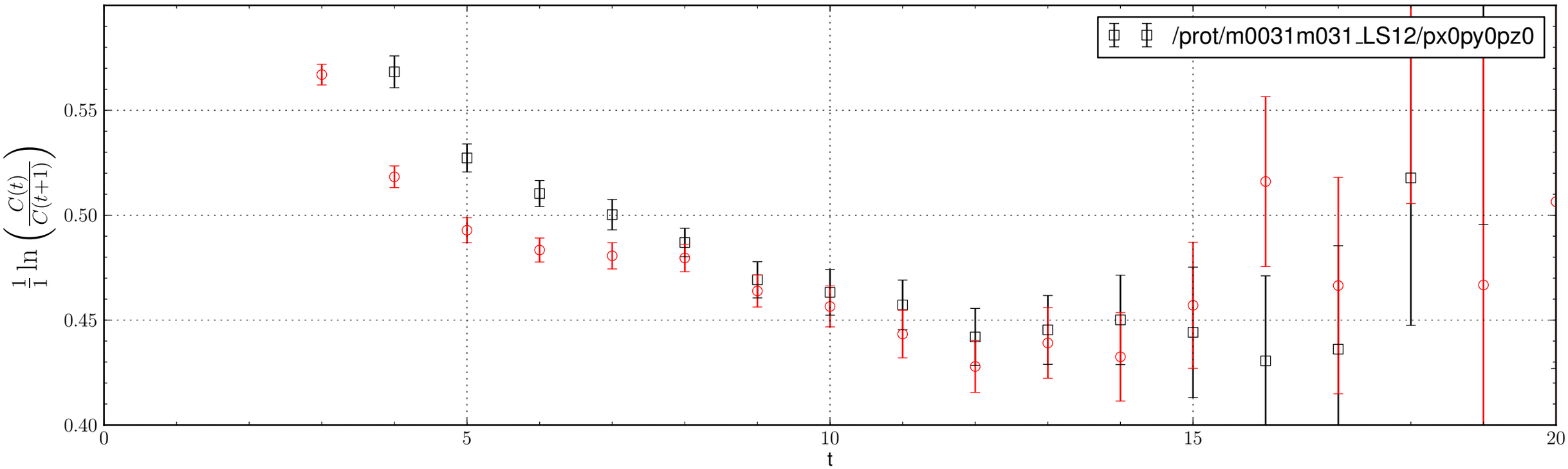
Matrix Prony

arXiv:1301.1114
arXiv:0905.0466

$$y(t+\tau) = \hat{T}(\tau) y(t)$$

take correlation function with one source and two sinks

$$y(t) = \begin{pmatrix} y_{SP}(t) \\ y_{SS}(t) \end{pmatrix}$$



between t=5 and t=15 there appear to be two states

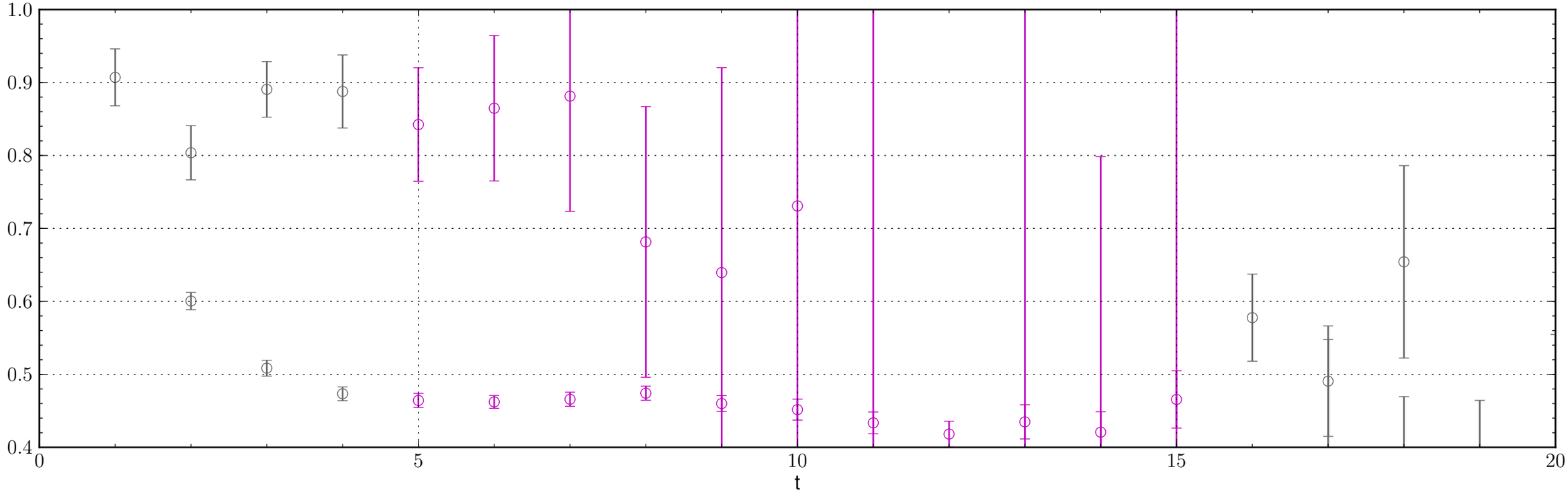
Matrix Prony

arXiv:1301.1114
arXiv:0905.0466

$$y(t+\tau) = \hat{T}(\tau) y(t)$$

take correlation function with one source and two sinks

$$y(t) = \begin{pmatrix} y_{SP}(t) \\ y_{SS}(t) \end{pmatrix}$$



between t=5 and t=15 there appear to be two states

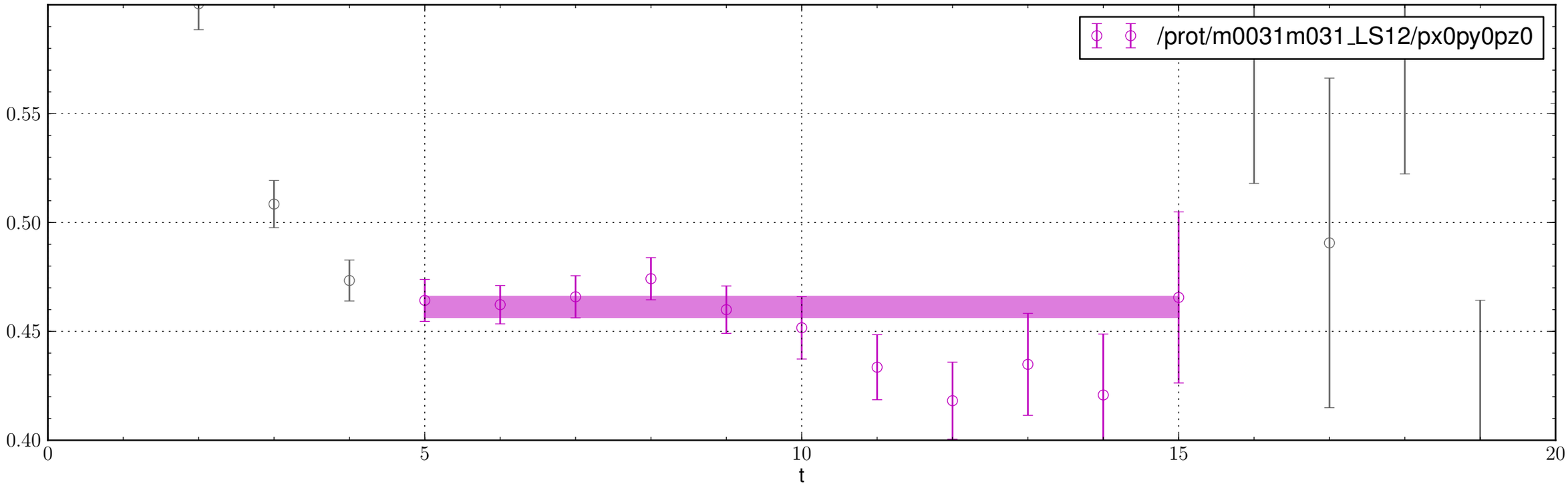
Matrix Prony

arXiv:1301.1114
arXiv:0905.0466

$$y(t+\tau) = \hat{T}(\tau) y(t)$$

take correlation function with one source and two sinks

$$y(t) = \begin{pmatrix} y_{SP}(t) \\ y_{SS}(t) \end{pmatrix}$$



between t=5 and t=15 there appear to be two states

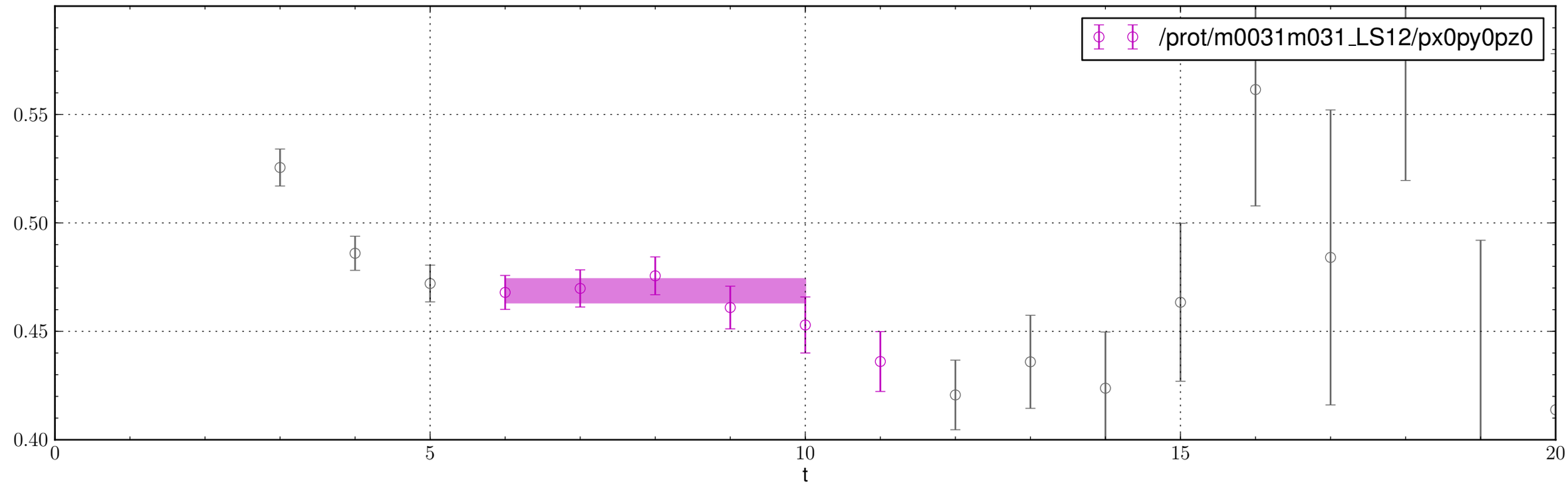
Matrix Prony

arXiv:1301.1114
arXiv:0905.0466

$$y(t+\tau) = \hat{T}(\tau) y(t)$$

take correlation function with one source and two sinks

$$y(t) = \begin{pmatrix} y_{SP}(t) \\ y_{SS}(t) \end{pmatrix}$$



between t=6 and t=11

also good

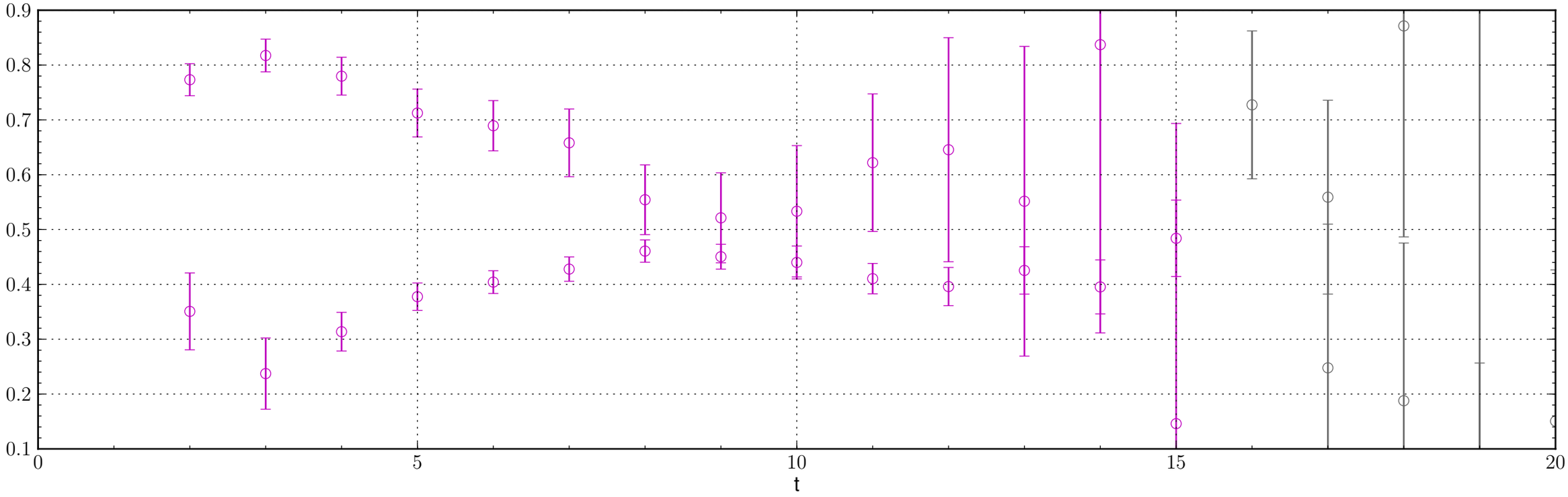
Matrix Prony

arXiv:1301.1114
arXiv:0905.0466

$$y(t+\tau) = \hat{T}(\tau) y(t)$$

take correlation function with one source and two sinks

$$y(t) = \begin{pmatrix} y_{SP}(t) \\ y_{SS}(t) \end{pmatrix}$$



between t=2 and t=15 too aggressive

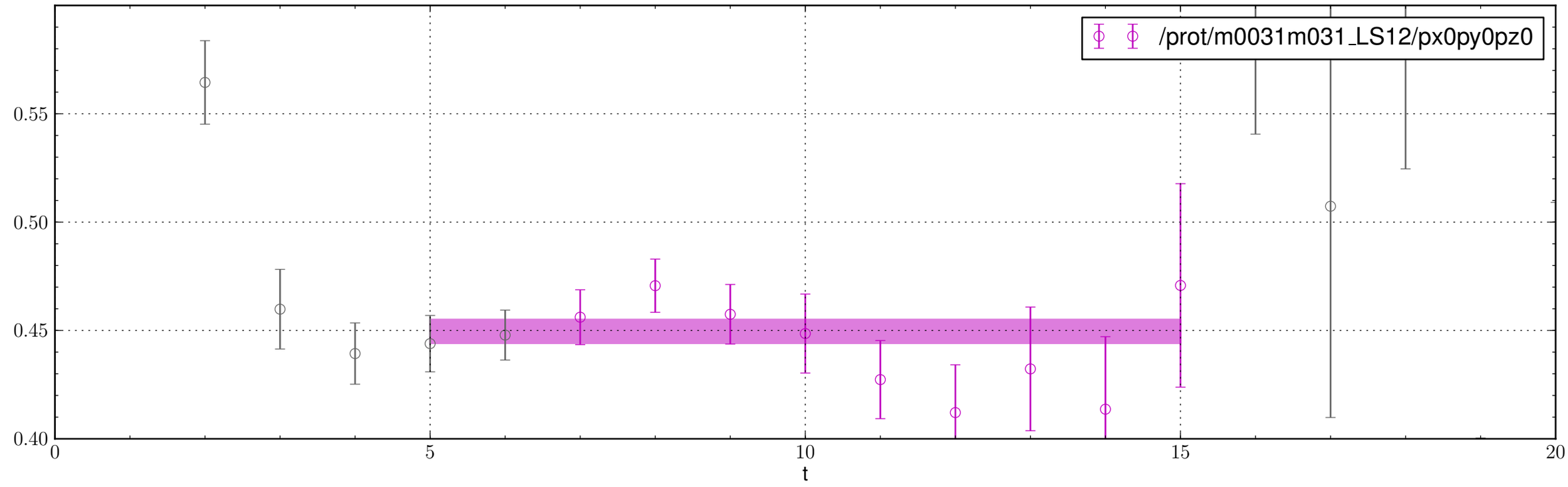
Matrix Prony

arXiv:1301.1114
arXiv:0905.0466

$$y(t+\tau) = \hat{T}(\tau) y(t)$$

take correlation function with one source and two sinks

$$y(t) = \begin{pmatrix} y_{SP}(t) \\ y_{SS}(t) \end{pmatrix}$$



between t=7 and t=15 also good

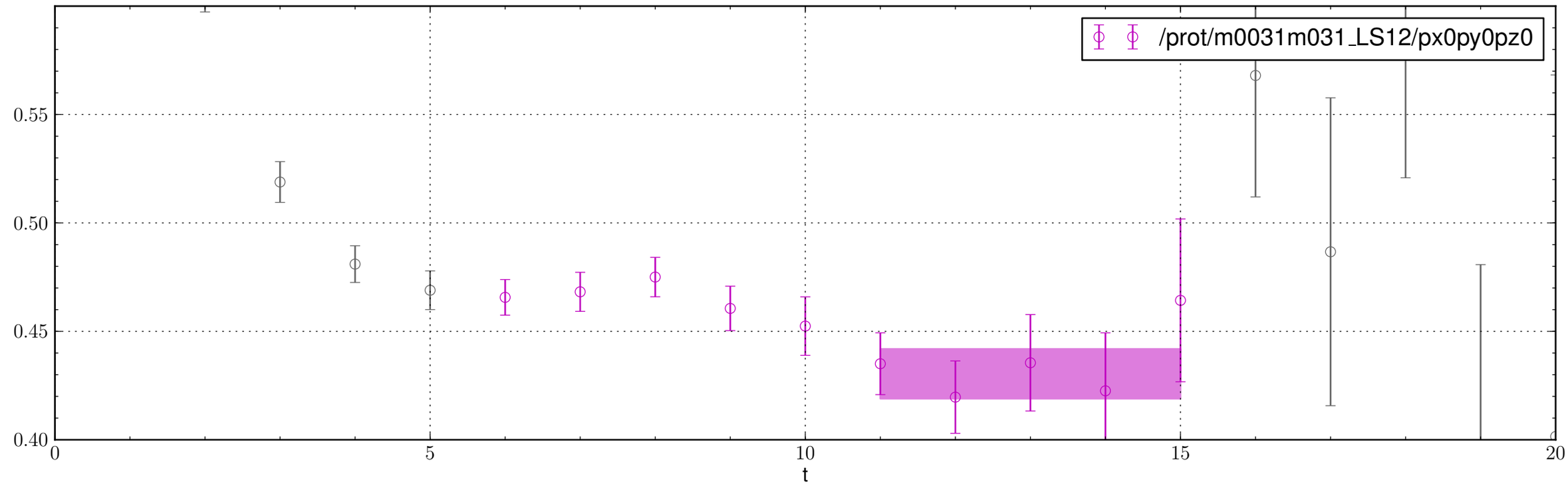
Matrix Prony

arXiv:1301.1114
arXiv:0905.0466

$$y(t+\tau) = \hat{T}(\tau) y(t)$$

take correlation function with one source and two sinks

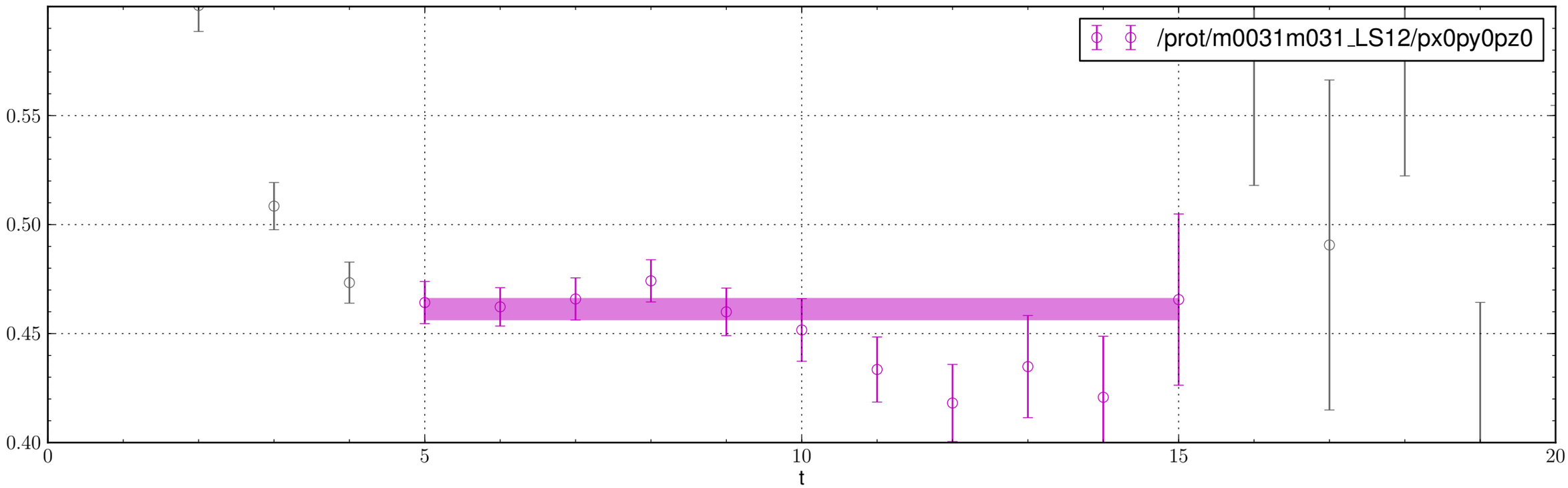
$$y(t) = \begin{pmatrix} y_{SP}(t) \\ y_{SS}(t) \end{pmatrix}$$



between t=7 and t=15 different range of fit ok

Matrix Prony

arXiv:1301.1114
arXiv:0905.0466



exponential decay of signal-to-noise for baryons challenging

- is late-time dip of effective mass true ground state value?
- lack of positive-definite correlation functions allow “false plateaus”
- results much more sensitive to choices of fit-range, MP-range
- want algorithm to weight all possible reasonable choices

Matrix Prony

arXiv:1301.1114
arXiv:0905.0466

The algorithm I currently use
(systematics are not uniquely defined)

- pick minimum Δt_{MP}

$$\hat{M}(\tau) = \left[\sum_{t_0}^{t_0+\Delta t} y(t+\tau) y^T(t) \right]^{-1}$$

- pick minimum Δt_{plat} in exp fit $t_f = t_i + \Delta t_{plat}$

Δt_{plat} choice guided by excited state masses

- pick minimum and maximum **t** you want to consider (correlator dependent)

- loop over independently chosen $t_0, \Delta t_{MP}, t_i, \Delta t_{plat}$

Matrix Prony

arXiv:1301.1114
arXiv:0905.0466

The algorithm I currently use
(systematics are not uniquely defined)

- specific for baryons, for each fit, pick a weight factor

$$w_i = \frac{Q_i}{\sigma_i} \qquad Q = \int_{\chi^2_{min}}^{\infty} d\chi^2 \mathcal{P}(\chi^2, dof) \qquad Q \in [0,1]$$

$\sigma_i =$ statistical uncertainty for a given fit

this choice allows for late time fluctuations being real, but suppresses them by their larger statistical uncertainty

$$\bar{m} = \frac{\sum_i w_i m_i}{\sum_j w_j}$$

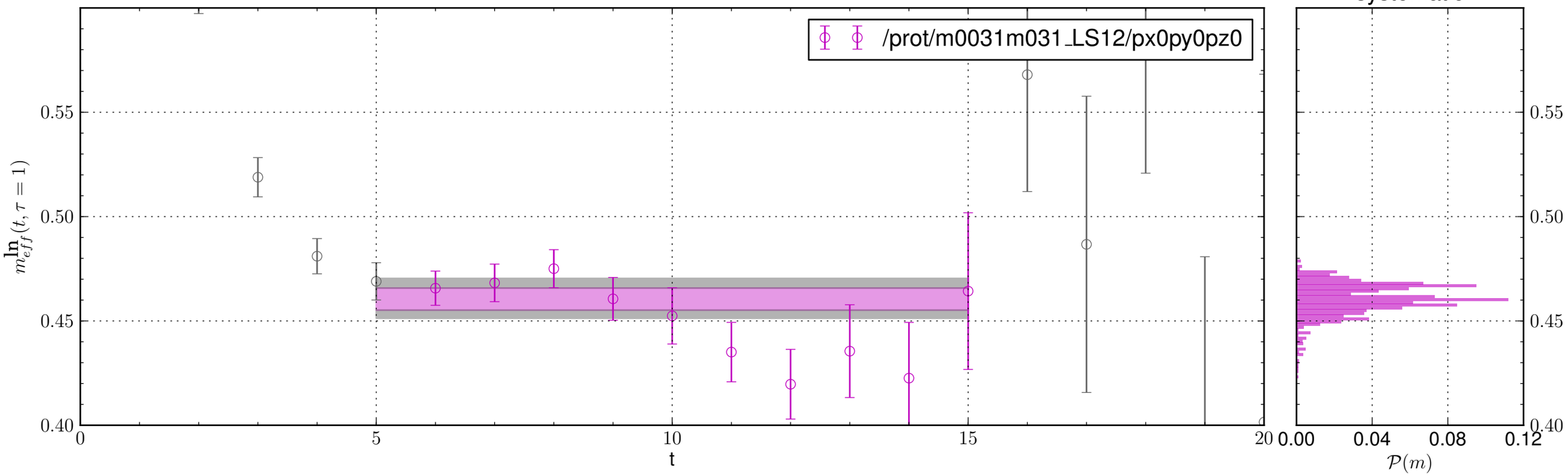
Matrix Prony

arXiv:1301.1114
arXiv:0905.0466

$$y(t+\tau) = \hat{T}(\tau) y(t)$$

take correlation function with one source and two sinks

$$y(t) = \begin{pmatrix} y_{SP}(t) \\ y_{SS}(t) \end{pmatrix}$$



16% and 84% quantiles chosen for systematic uncertainty
inner band statistical outer band stat+sys added in quadrature

$$\mathcal{P}(m) = \text{ mass probability distribution function}$$

$$\mathcal{P}(m_i) = \frac{w_i}{\sum_j w_j}$$

Poor Man's Advanced Fitting Methods

Go forth and try these methods