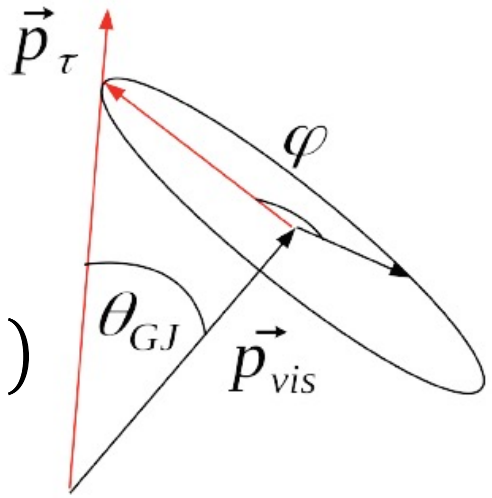
Tau Mass Reconstruction. One More Status Report

Claudio (with lots of help from Hualin)

10 Feb 2022

H → ττ kinematics



There are up to 6 unknowns

$$\vec{a} = (x_1, \varphi_1, m_{\nu\nu 1}, x_2, \varphi_2, m_{\nu\nu 2})$$

$\frac{E_{vis}}{E_\tau}$

νν inv. mass (leptonic decays)

There is "event data"

$$\vec{y} = (E_{vis1}, \vec{p}_{vis1}, E_{vis2}, \vec{p}_{vis2}, \vec{E}_T^{miss})$$

Note: θ_{GJ} can be calculated from the other vars. (not immediately obvious)

SvFit has a likelihood, maximized as a function of m_{test}

$$\mathcal{L}(m_{test}|event\ data) = \int p(m_{\tau\tau}|\vec{y},\vec{a})\delta(m_{\tau\tau} - m_{test})d\vec{a}$$

$m_{\tau\tau}(\vec{y},\vec{a})$

$$p(m_{\tau\tau}|\vec{y},\vec{a}) = ME \cdot TF \cdot REG$$

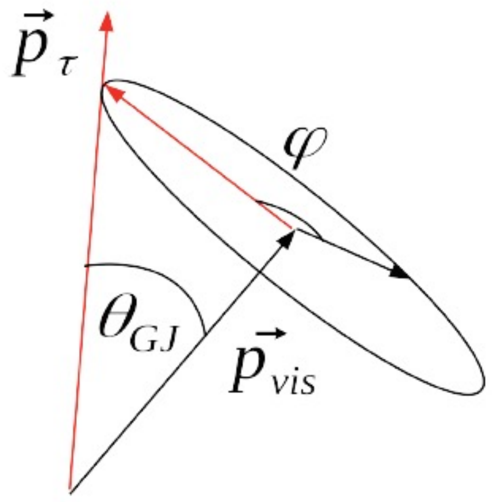
matrix element

transfer factor, ie, detector effect

fudge factor

Integral is complicated, done with MC methods, slow

FastMTT, simplification of SVFit (faster)



- Collinear approximation, $\theta_{GJ} = 0$
  - $\vec{a} = (x_1, \cancel{\varphi_1}, m_{\nu\nu 1}, x_2, \cancel{\varphi_2}, m_{\nu\nu 2})$

- Matrix elements = constant
- Only "Transfer Factor" is for E_T^{miss}

$$\mathcal{L}(m_{test}|event\ data) = MET\_TF(\vec{E}_T^{miss\ reco}, \vec{E}_T^{miss\ hypo}) \cdot \int \delta(m_{\tau\tau} - m_{test}) d\vec{a} \quad \bullet\ REG$$

$$= \frac{1}{m_{test}^3}$$

(controls high tails)

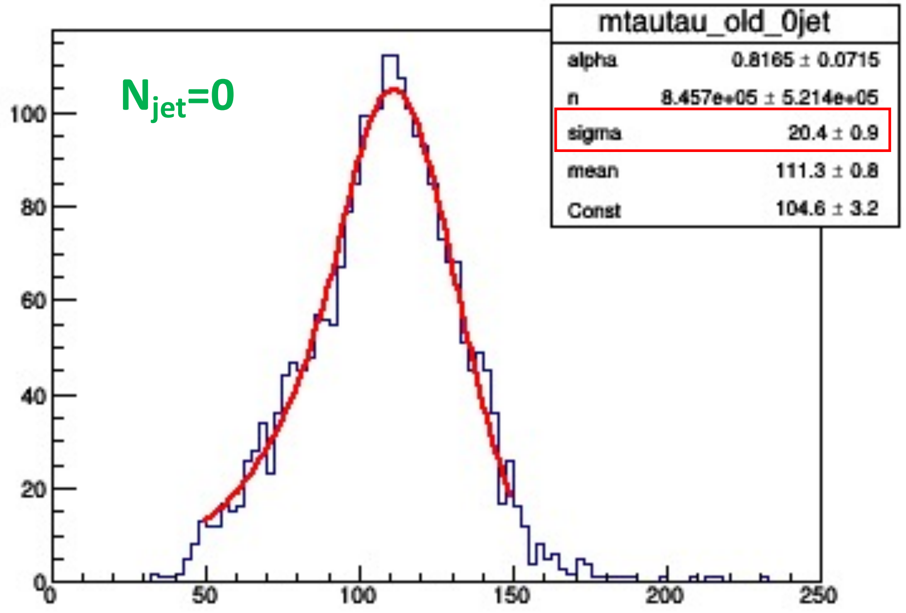
$$MET\_TF = \mathcal{L}(\vec{E}_T^{miss\ reco}|\vec{E}_T^{miss\ hypo}) =$$
$$\frac{1}{2\pi\sqrt{|V|}} \cdot \exp\left[-\frac{1}{2}\left((\vec{E}_T^{miss\ reco} - \vec{E}_T^{miss\ hypo})^T \cdot V^{-1} \cdot (\vec{E}_T^{miss\ reco} - \vec{E}_T^{miss\ hypo})\right)\right]$$

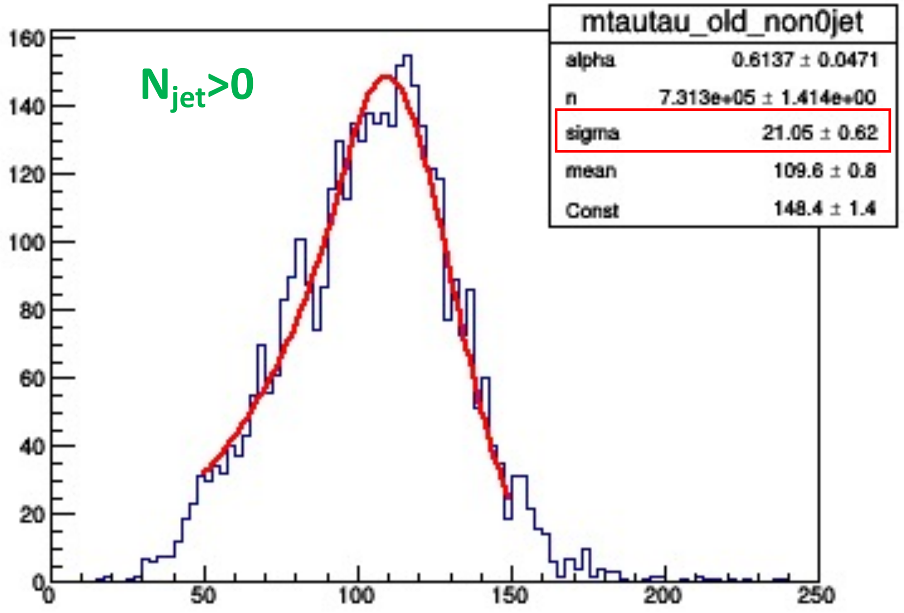
depends on x_1 and x_2

- The integral over the δ-function can be done analytically
  - There is a bit of fudging of the limits of integrations as well

http://cms.cern.ch/iCMS/jsp/openfile.jsp?tp=draft&files=AN2019_032_v3.pdf

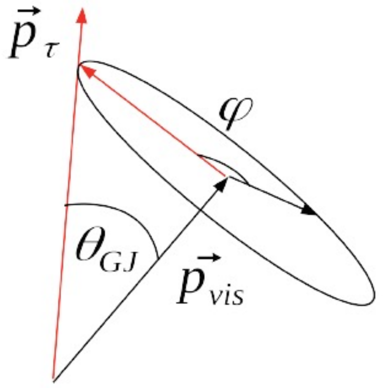
FastMTT out of the box. Crystal Ball Fit



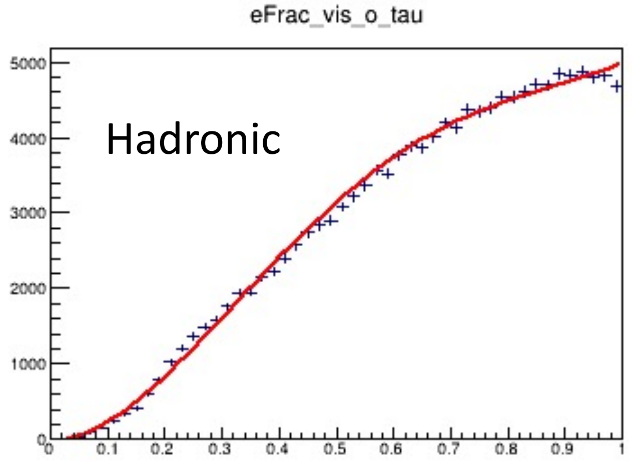


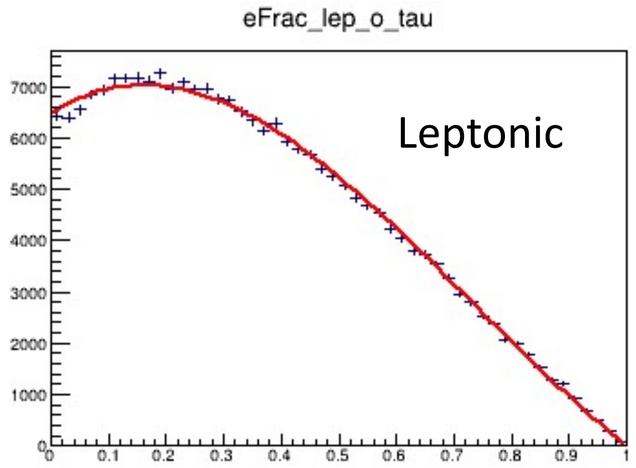
Should probably take the time to look at these for different final states!!!

Improvement of FastMTT (CC version)



- Super-Collinear approximation, $\theta_{GJ} = 0$, <u>νν also collinear</u>
  - $\vec{a} = (x_1, \cancel{\varphi_1}, \cancel{m_{\nu\nu 1}}, x_2, \cancel{\varphi_2}, \cancel{m_{\nu\nu 2}})$
- Include expected x_i pdfs from MC (~matrix element)
  - In practice since the pdfs come from MC the lack of collinearity of the neutrinos in leptonic decays is actually included! (integrated over)





Fits to "pol4" (hadronic)

```
*******************************************
Minimizer is Linear / Migrad
Chi2                      =      201.181
NDf                       =           44
p0                        =     -20.8584   +/-   5.60061
p1                        =      -103.46   +/-   181.624
p2                        =      28912.2   +/-   1039.63
p3                        =       -40754   +/-   1916.33
p4                        =      16937.9   +/-   1087.52
```

"pol3" (leptonic)

```
*******************************************
Minimizer is Linear / Migrad
Chi2                      =      118.649
NDf                       =           46
p0                        =       6462.7   +/-   42.8792
p1                        =      7119.87   +/-   319.185
p2                        =     -24764.4   +/-   642.921
p3                        =      11189.4   +/-   371.598
```

- With polynomial functions the integral with the δ-function can be done analytically (still very fast!)
  - $\int p_1(x_1) p_2(x_2) \delta(m_{\tau\tau} - m_{test}) dx_1 dx_2$

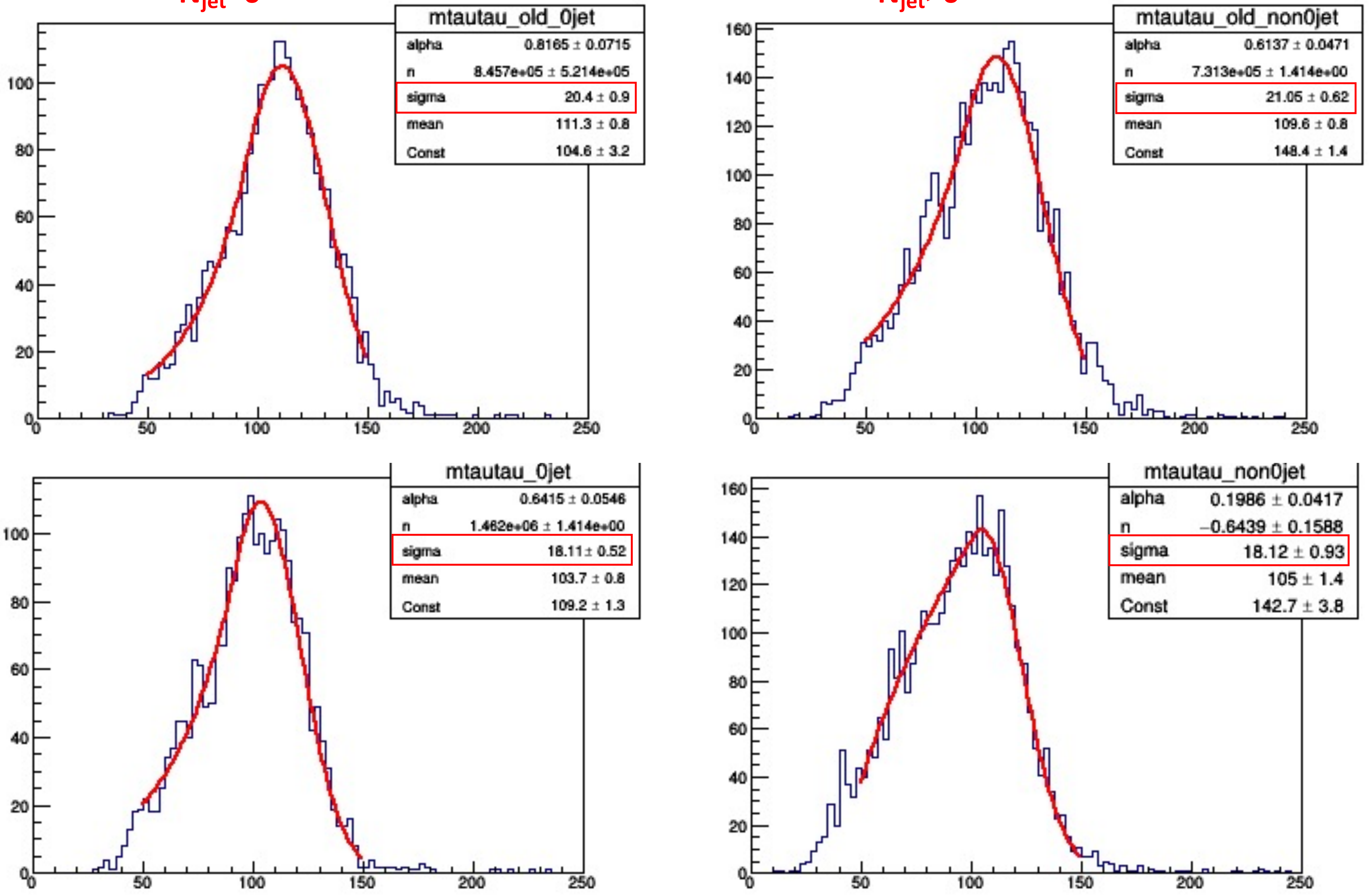
$N_{jet}=0$ $N_{jet}>0$

FastMTT out of the box

FastMTT with matrix element

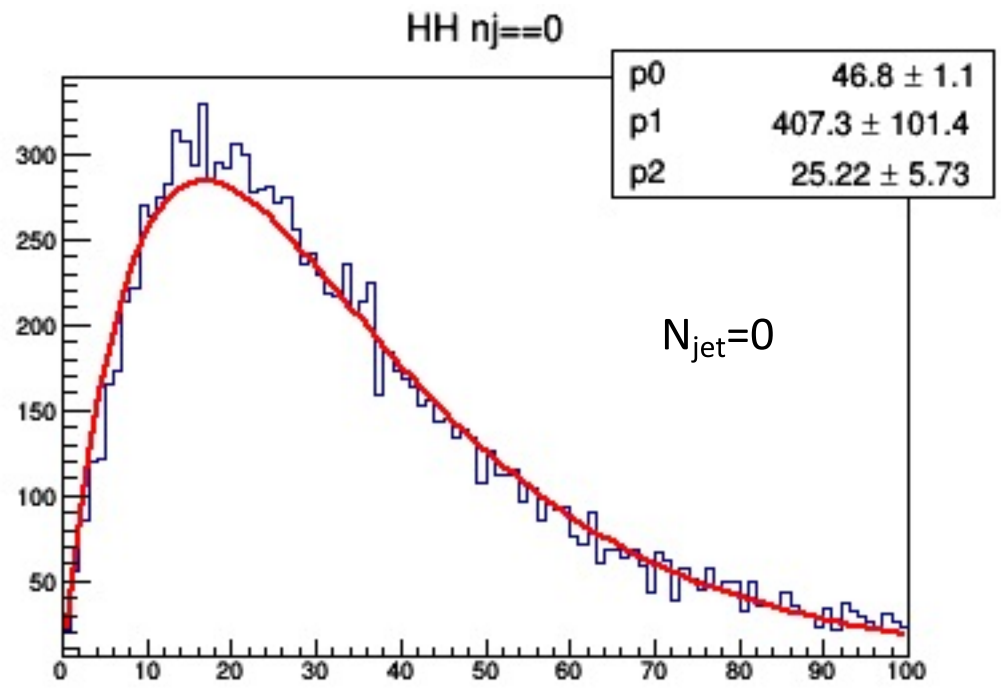
Same fudge factor (not tuned)

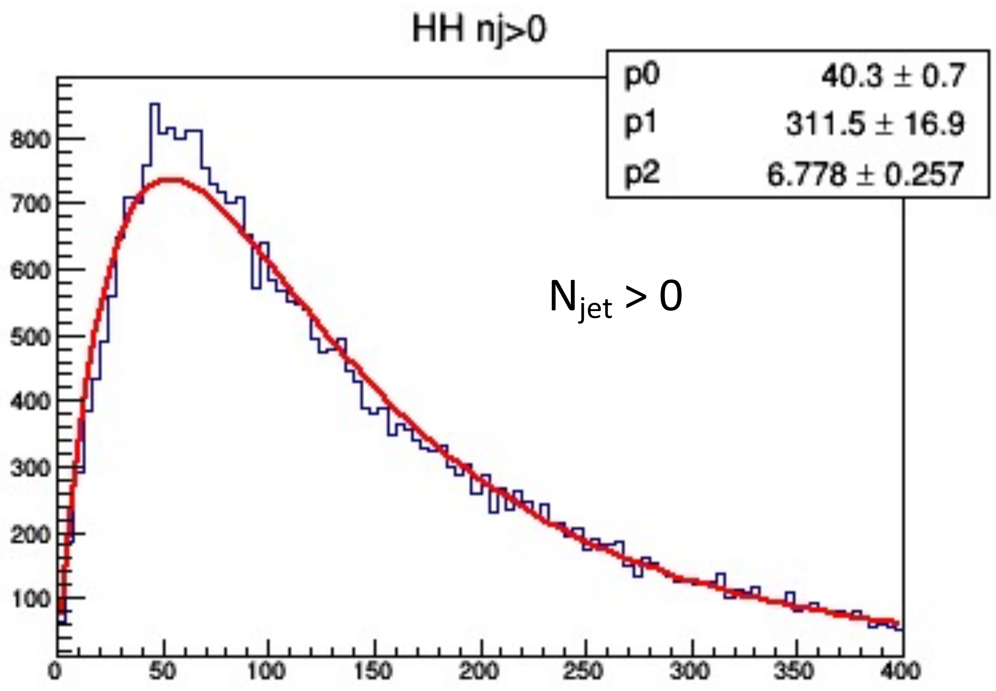
~ 10% narrower



Would like to incorporate P_T(HH) information

GEN expectations:






```
Double_t func1(Double_t *x, Double_t *par){
    Double_t fitval;
    Double_t pt = x[0];
    Double_t p0 = par[0];
    Double_t p1 = par[1];
    Double_t p2 = par[2];
    fitval = pt*p0*pow((1+pt/p1), -p2);
    return fitval;
}
```

Not so easy to do in the SvFit/FastMTT approach

$$\mathcal{L}(m_{test}, p_{T,test}|event\ data) = MET\_TF(\vec{E}_T^{miss\ reco}, \vec{E}_T^{miss\ hypo}) \cdot \int \delta(m_{\tau\tau} - m_{test}) d\vec{a} \cdot \int \delta(p_{Tx,\tau\tau} - p_{Tx,test}) d\vec{a} \cdot \int \delta(p_{Ty,\tau\tau} - p_{Ty,test}) d\vec{a} \quad (8)$$

- The δ functions are nasty unless we remove the matrix element completely
- (There may be a way…need to talk to Hualin as one needs to run the actual FastMTT code, not my kludges)

Instead, try a different approach, similar to Atlas

$$\mathcal{L} = MET\_TF \cdot p(x_1) \cdot p(x_2) \cdot p(P_T) \cdot REG$$

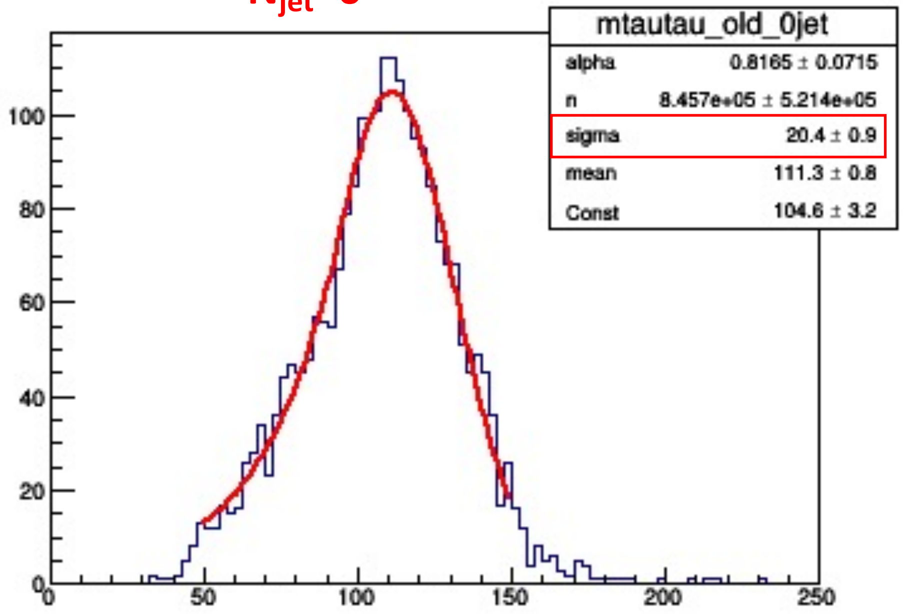
Since $P_T = P_T(x_1,x_2)$ this is probably not entirely kosher. Not sure. But neither is *REG*…

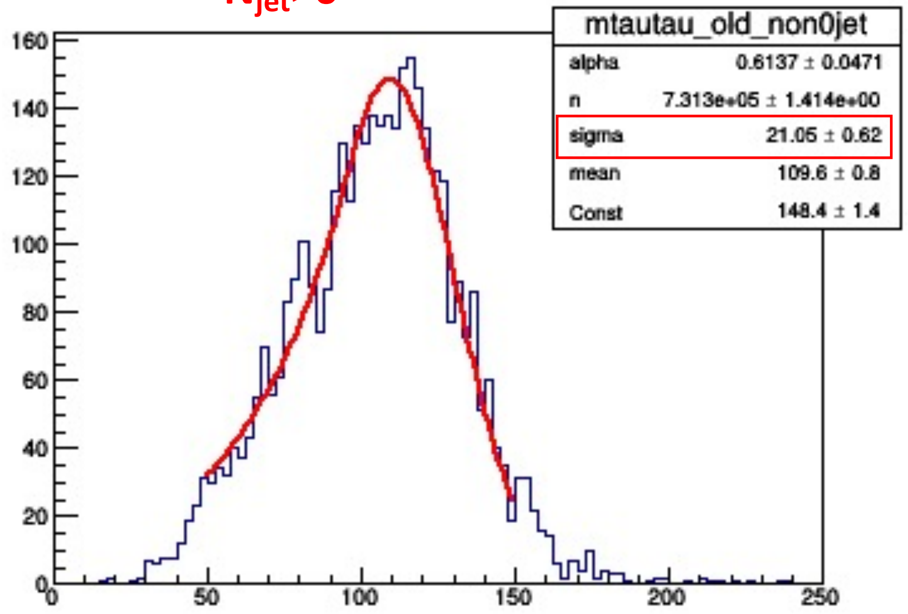
- <u>First, check this out without the P_T information</u>

https://arxiv.org/abs/1012.4686

$N_{jet}=0$ **$N_{jet}>0$**

FastMTT out of the box

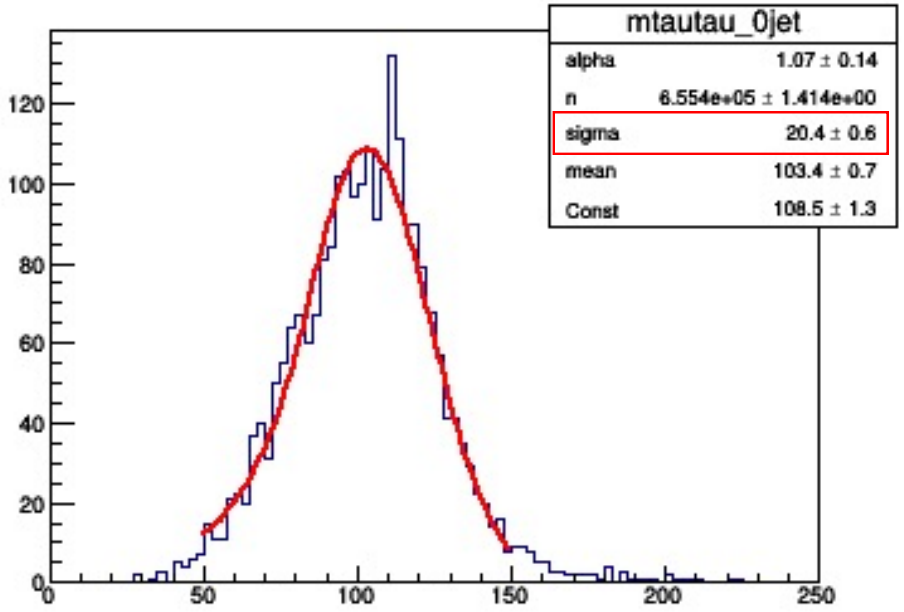


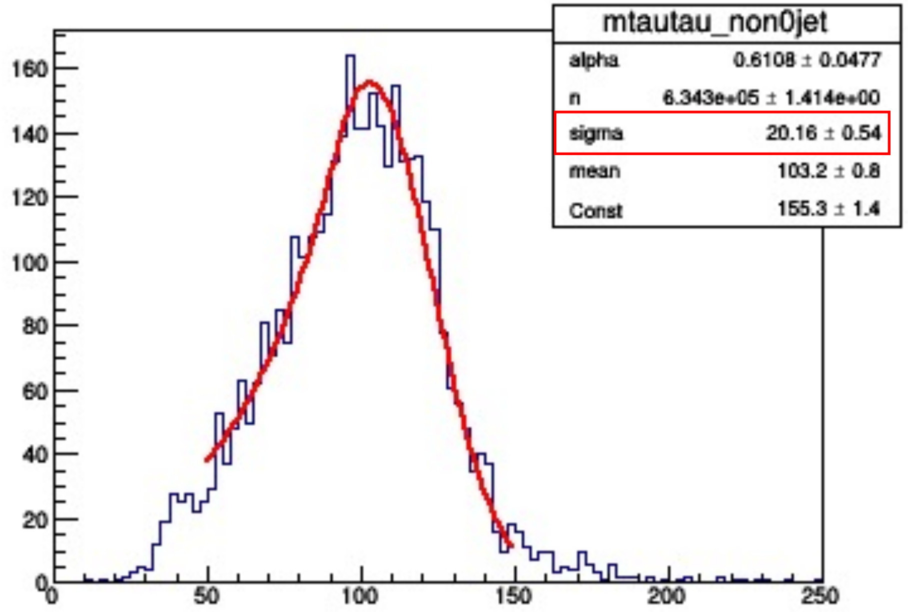


<u>Atlas-like, no P_T</u>

Same fudge factor (not tuned)

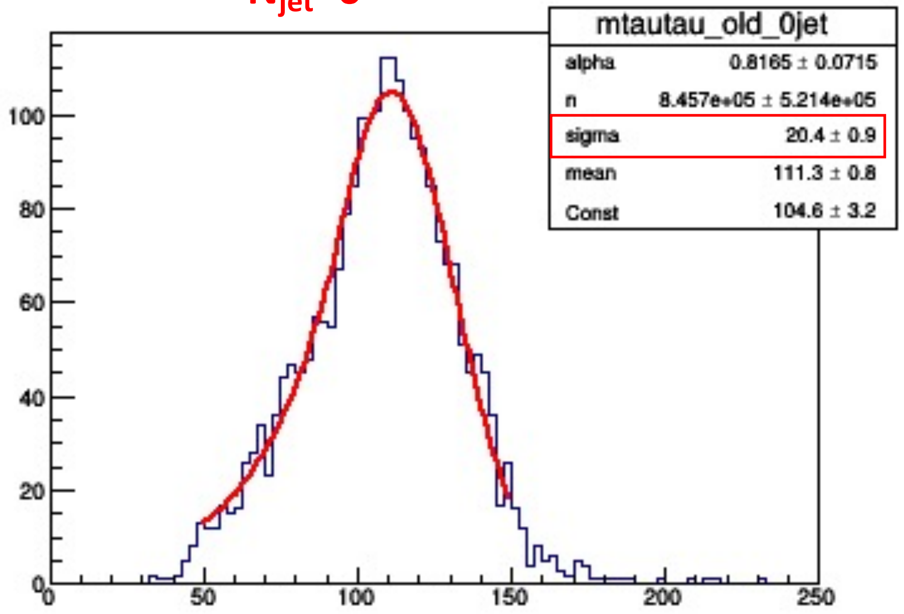
~ same as FastMTT

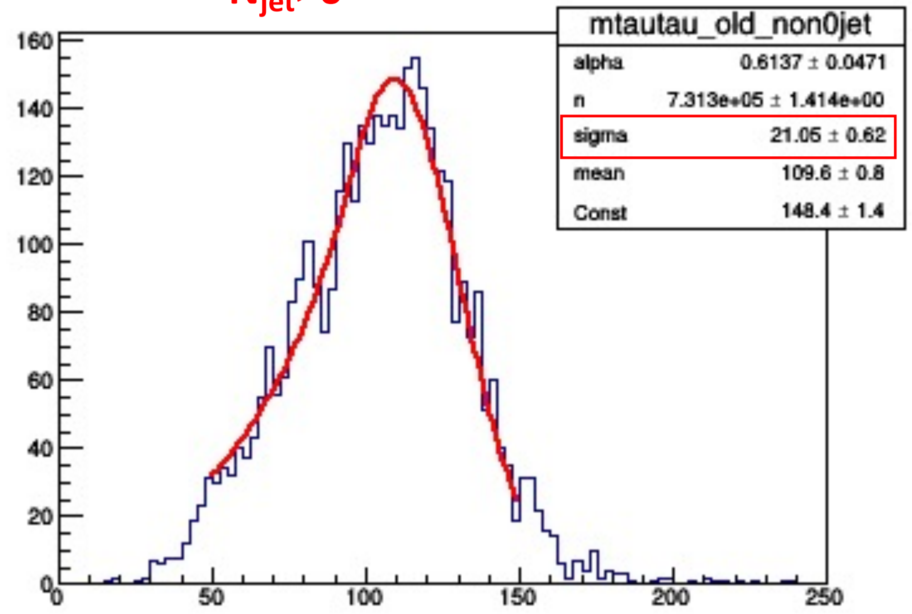




$N_{jet}=0$ $\quad$ $N_{jet}>0$

FastMTT out of the box



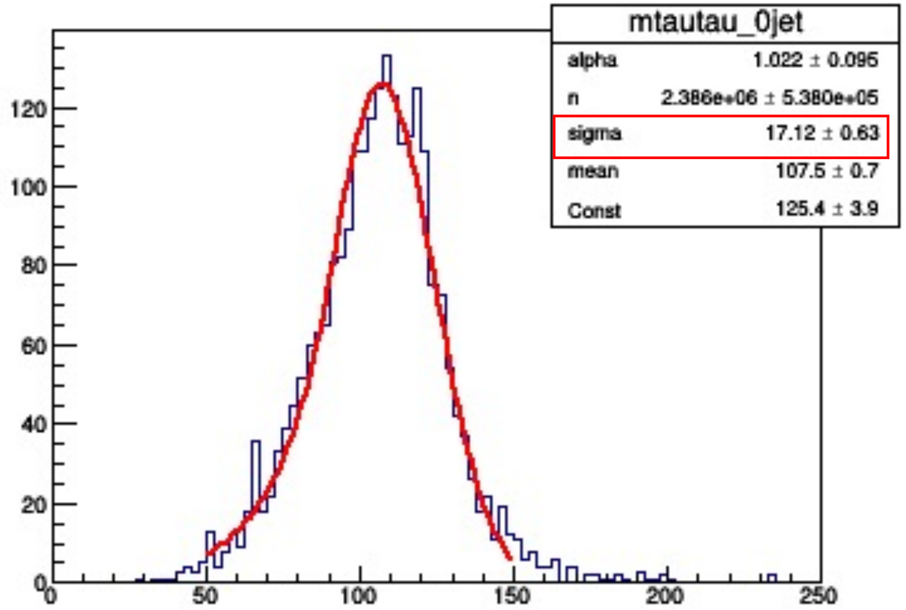


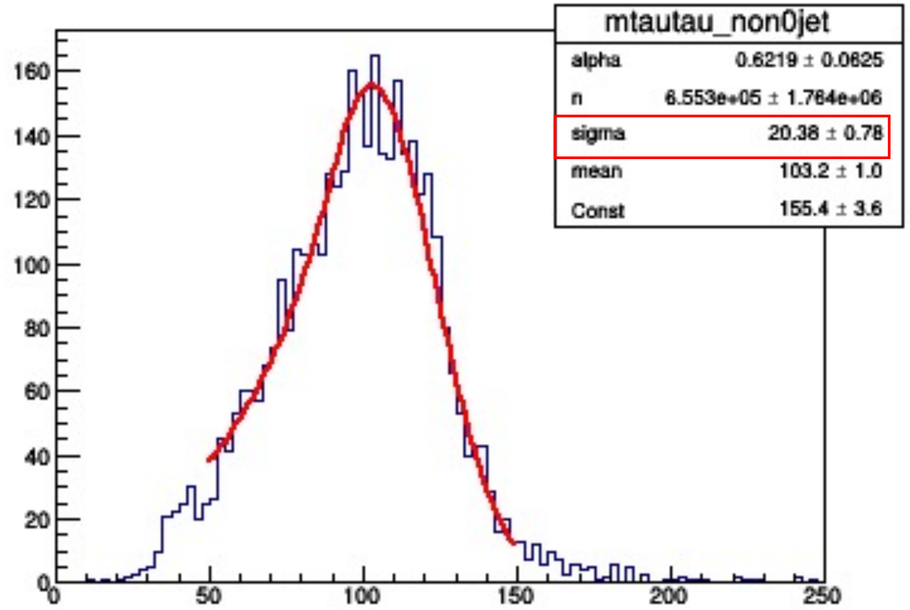
<u>**Atlas-like, with P_T**</u>

Same fudge factor (not tuned)

0 jets:
~ 15% narrower

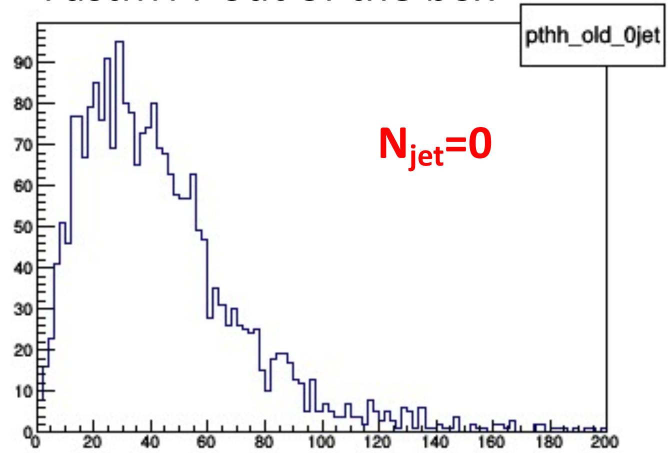
> 0 jet
~ 3% narrower (maybe)

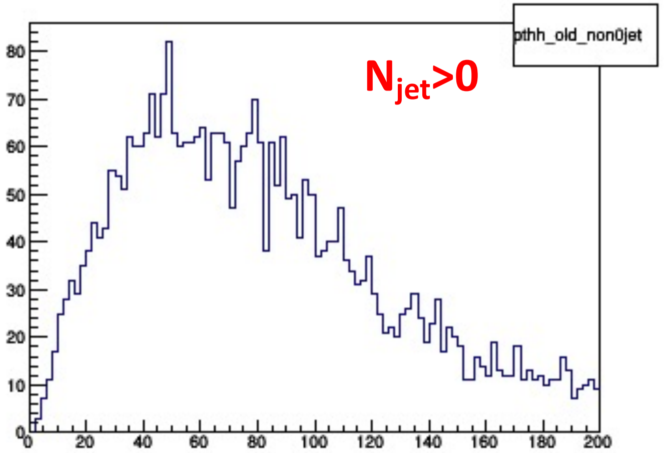




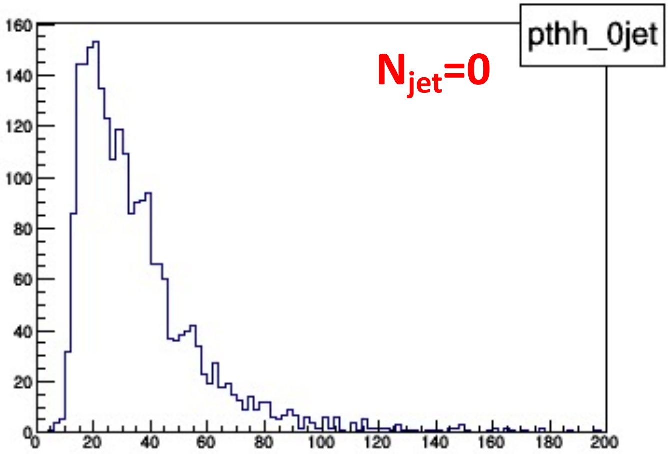
What happened to P_T?

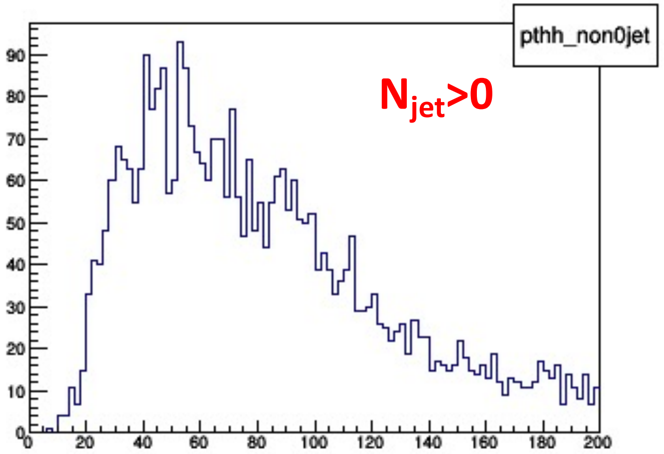
FastMTT out of the box





Atlas-like, with P_T





There is a depletion of events near zero in the Atlas-like (with P_T) algorithm.

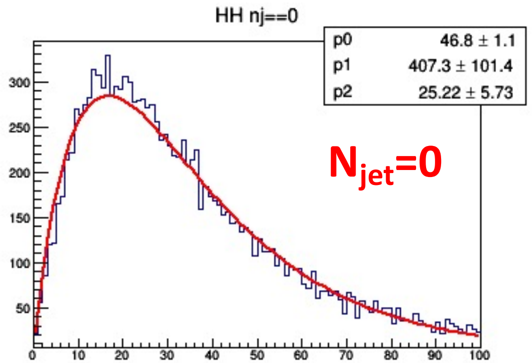
This is not unexpected.

There are large uncertainties.

The P_T pdfs go to zero at $P_T = 0$.

Because of the large uncertainties the penalty to move any given event away from zero is not so large.

Therefore, events are moved away from 0



Summary (1)

- It is possible to improve FastMTT a little bit by adding the matrix element.
- An Atlas-like reconstruction does just as well as FastMTT
- In the Atlas-like reconstruction it is "easy" to include the P_T(HH) information
- The additional information improves resolution in the $N_{jet} = 0$ sample by about 15%, but hardly at all in the $N_{jet}>0$ case.
  - It may be improved by fine-tuning the fudge factor (eg: different fudge factors for different tau decays?)
- The P_T(HH) distribution becomes a little weird, but actually I think it makes sense.
- The inclusion of the P_T(HH) is probably not kosher
- I think there may be a way to put the P_T(HH) information in the FastMTT framework. Not sure
- Franny's study seems to indicate that none of this matters much if at all 😂

Summary (2)

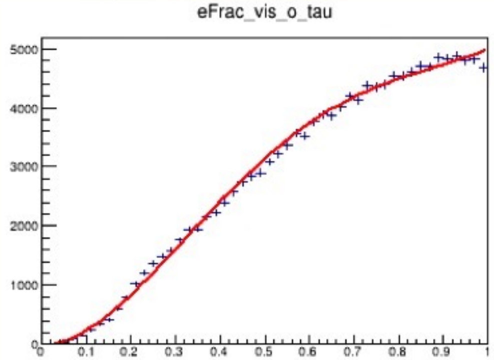
- The “FastMTT+ME” the “Atlas-like” algos are coded by me inside a custom looper and should be repackaged in case we want to use them more widely.
- Think more about the slightly iffy use of the P_T in the Atlas-like algorithm
- Talk to Hualin about one more attempt to include the P_T pdf in the standard FastMTT (ie: without ME) in a sensible way
- I wonder if the average of the the two algorithms leads to better resolution?
- There is some algebra involved in “FastMTT+ME”, needs to be double-checked
  - See backup

Backup

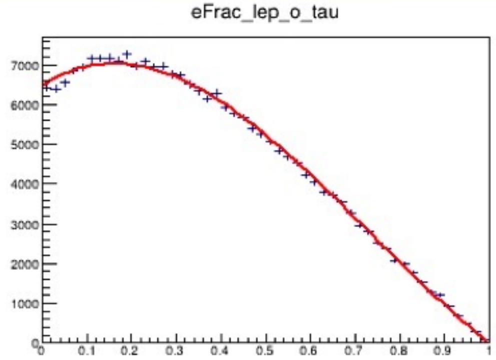
Algebra for FastMTT + ME

TAU DECAY PDFs

HADRONIC TAU



LEPTONIC TAU



Fit to "POL4"

```
****************************************
Minimizer is Linear / Migrad
Chi2                      =      201.181
NDf                       =           44
p0                        =     -20.8584   +/-   5.60061
p1                        =      -103.46   +/-   181.624
p2                        =      28912.2   +/-   1039.63
p3                        =       -40754   +/-   1916.33
p4                        =      16937.9   +/-   1087.52
```

Fit to "POL3"

```
****************************************
Minimizer is Linear / Migrad
Chi2                      =      118.649
NDf                       =           46
p0                        =       6462.7   +/-   42.8792
p1                        =      7119.87   +/-   319.185
p2                        =     -24764.4   +/-   642.921
p3                        =      11189.4   +/-   371.598
```

$$f(x_i) = \sum_{\delta=0}^{4} a_\delta x_i^\delta$$ with $a_4 = 0$ for leptonic

We reuse some of the results from Appendix A of the FastMTT analysis note

$$m_{\tau\tau} \simeq \frac{m_{vis}}{\sqrt{x_1 x_2}}$$

$$\frac{2m_{vis}^2}{m_{test}^3}\frac{1}{x_2}, \quad x_1 x_2 = \left(\frac{m_{vis}}{m_{test}}\right)^2$$

$$x_{min} = \max\left(x_{2,min}, \left(\frac{m_{vis}}{m_{test}}\right)^2\right)$$
$$x_{max} = \min\left(1, \left(\frac{m_{vis}}{m_{test}}\right)^2 \frac{1}{x_{1,min}}\right)$$

$$V_{\tau_h\tau_h}(m_{test}, m_{vis}) = \int_{x_{1,min}}^{1} dx_1 \int_{x_{2,min}}^{1} dx_2 \int_0^{2\pi} d\phi_1 \int_0^{2\pi} d\phi_1 \delta\left(m_{test} - \frac{m_{vis}}{\sqrt{x_1 x_2}}\right) =$$

$$4\pi^2 \int_{x_{1,min}}^{1} dx_1 \int_{x_{2,min}}^{1} dx_2 \delta\left(m_{test} - \frac{m_{vis}}{\sqrt{x_1 x_2}}\right) =$$

$$4\pi^2 \frac{2m_{vis}^2}{m_{test}^3} \int_{x_{min}}^{x_{max}} \frac{1}{x_2} dx_2 = \cancel{4\pi^2 \frac{2m_{vis}^2}{m_{test}^3} \log\left(\frac{x_{max}}{x_{min}}\right)} \leftarrow$$ This is for the case that we ignore $f(x_1) f(x_2)$

$$x_{1,min} = \left(\frac{m_1}{m_\tau}\right)^2, \quad x_{2,min} = \left(\frac{m_2}{m_\tau}\right)^2,$$

m_1 and m_2 are the visible mass of the τ

The formula above works for hadronic taus, which have $M_{\nu\nu} = 0$, which is exactly what we care about

$$\int f_1(x_1) f_2(x_2) \delta\left(M_{test} - \frac{M_{vis}}{\sqrt{x_1 x_2}}\right) dx_1 dx_2$$

$$= \frac{2M_{vis}^2}{M_{test}^3} \int_{x_{min}}^{x_{max}} \frac{1}{x_2} f_1\left(\frac{M_{vis}^2}{M_{test}^2}\frac{1}{x_2}\right) f_2(x_2) dx_2$$

Let $\alpha = \left(\frac{M_{vis}}{M_{test}}\right)^2$

Also $f_1(x) = \sum_{j=0}^{4} a_{1j} x^j \qquad f_2(x) = \sum_{k=0}^{4} a_{2k} x^k$

$$= \frac{2\alpha}{M_{test}} \int_{x_{min}}^{x_{max}} \sum_{j=0}^{4} \sum_{k=0}^{4} \frac{a_{1j}}{x_2} \alpha^j x_2^{-j} a_{2k} x_2^k \, dx_2$$

$$= \frac{2\alpha}{M_{test}} \int_{x_{min}}^{x_{max}} \sum_{j=0}^{4} a_{1j} \alpha^j \sum_{k=0}^{4} a_{2k} x_2^{k-j-1} \, dx_2$$

$$= \boxed{\frac{2\alpha}{M_{test}} \sum_j a_{1j} \alpha^j \sum_k a_{2k} \left[\delta_{kj} \log \frac{x_{max}}{x_{min}} + \frac{1-\delta_{kj}}{k-j} \left(x_{max}^{k-j} - x_{min}^{k-j} \right) \right]}$$

(the factor of 2 is irrelevant for the purpose of maximizing the likelihood)