Modeling Multi-Variate Gaussian Distributions for Higgs Coupling Analysis

Olivia A. Krohn

Introduction: Higgs Coupling

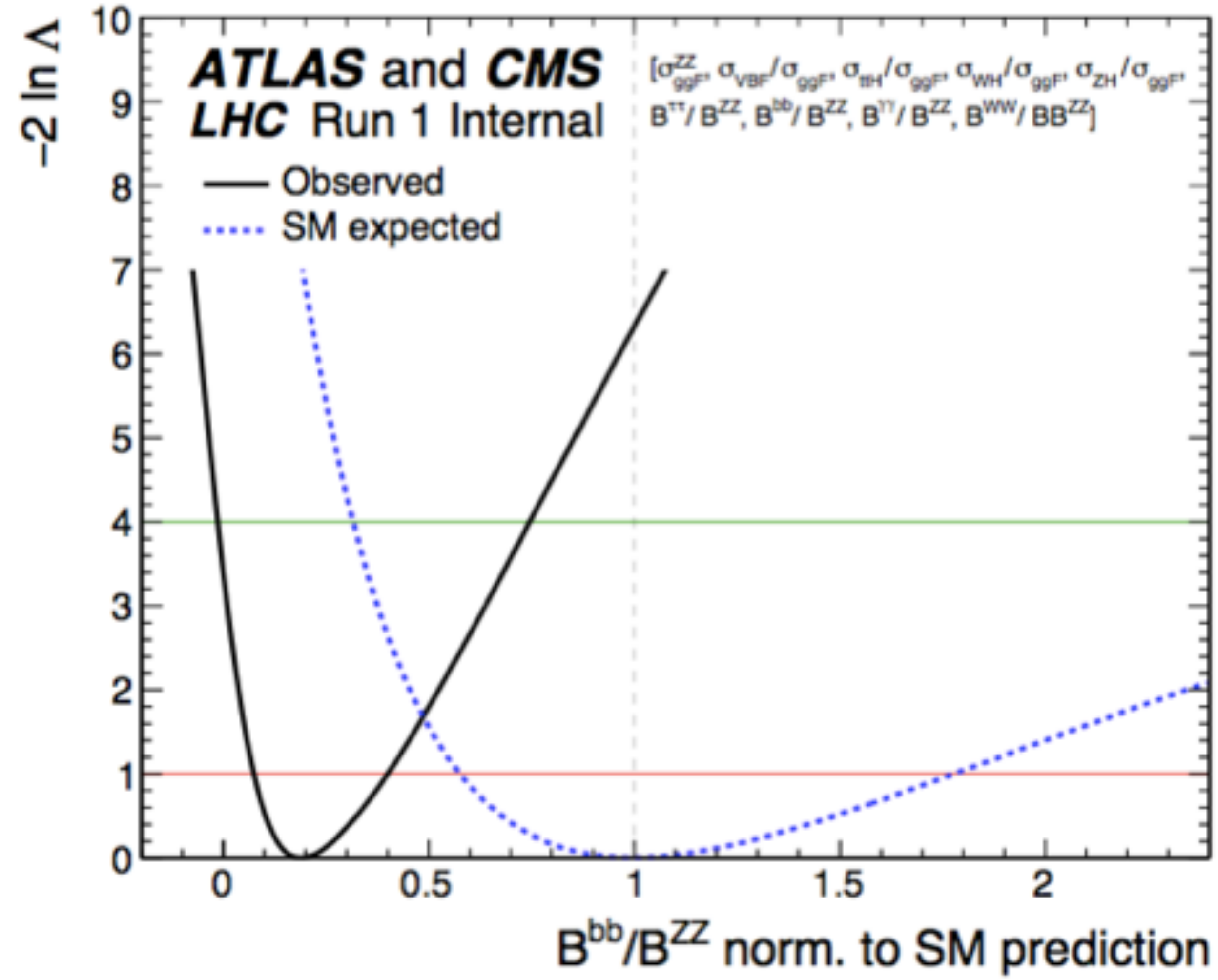
- Interested in joining CMS+ATLAS Run 1 Higgs production and decay modes
- The probability distribution function of the product of the cross-sections and branching fractions ($\sigma_i \cdot B^f$) can be constructed based on experimental observables
- These observables are divided by the SM expectations, and are the parameters of a likelihood function

$$\mu_i = \frac{\sigma_i}{(\sigma_i)_{\text{SM}}} \quad \text{and} \quad \mu^f = \frac{\text{B}^f}{(\text{B}^f)_{\text{SM}}}. \qquad \mu_i^f = \frac{\sigma_i \cdot \text{B}^f}{(\sigma_i)_{\text{SM}} \cdot (\text{B}^f)_{\text{SM}}} = \mu_i \cdot \mu^f.$$

- Note μ_i^f= 1 if data aligns perfectly with standard model, μ_i^f=0 if there is no signal
- Can also parametrize in terms of the ratio of μs

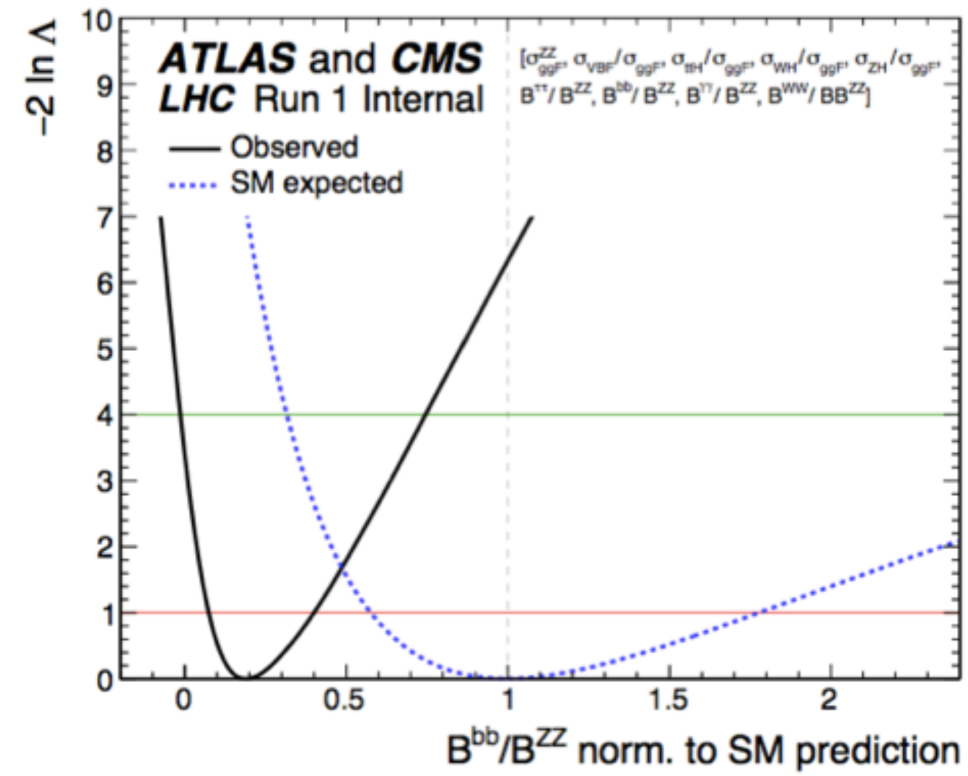
Likelihood Fits & Motivation

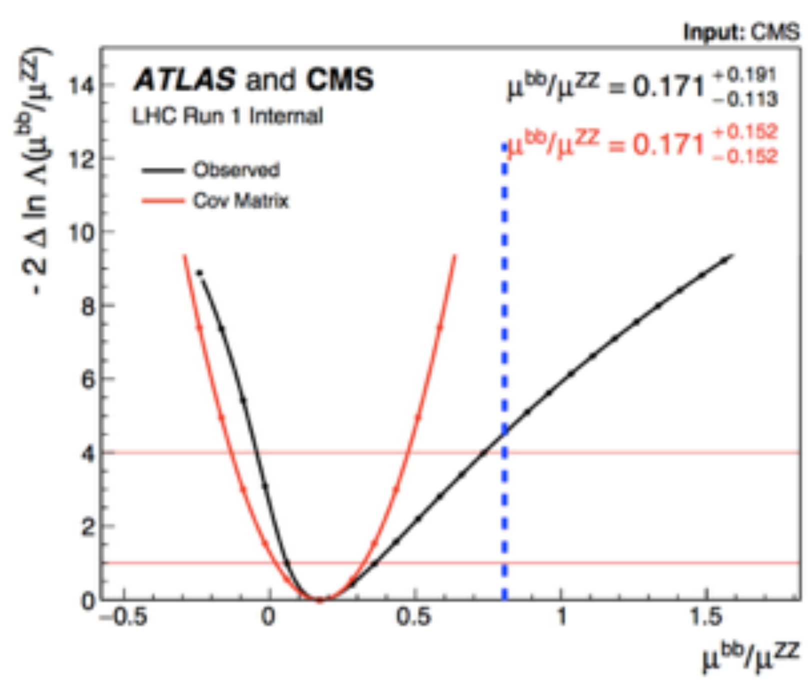
- A likelihood function is directly related to the probability distribution of μ
- An example would be $L(\mu) = P(N \mid \mu^* S + B)$
- You can maximize this L(μ) to find û, the most likely value
- The negative-log-likelihood is often used, $-2 \cdot \ln[L(\mu)/L(\hat{u})] = -2 \cdot \ln \Lambda$
- This is useful because $-2 \cdot \ln \Lambda = 1$ occurs at ±1σ, $-2 \cdot \ln \Lambda = 4$ at ±2σ, etc.

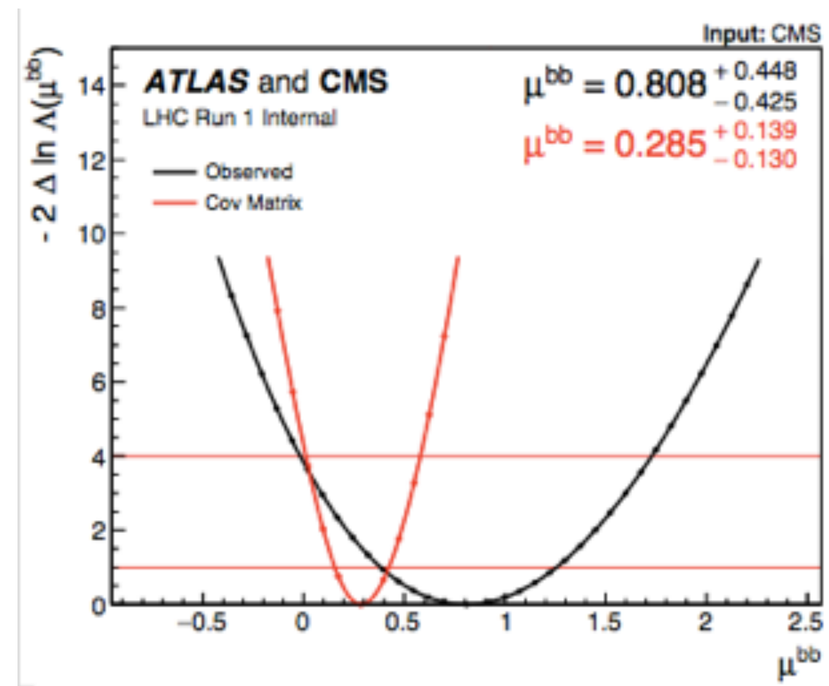


Likelihood Fits & Motivation

- In publishing this Run 1 data, the best fit value and 1-σ values are given
- However, usually the distributions are usually non-Gaussian
- Assuming they are Gaussian with the reported data gives strong bias to results







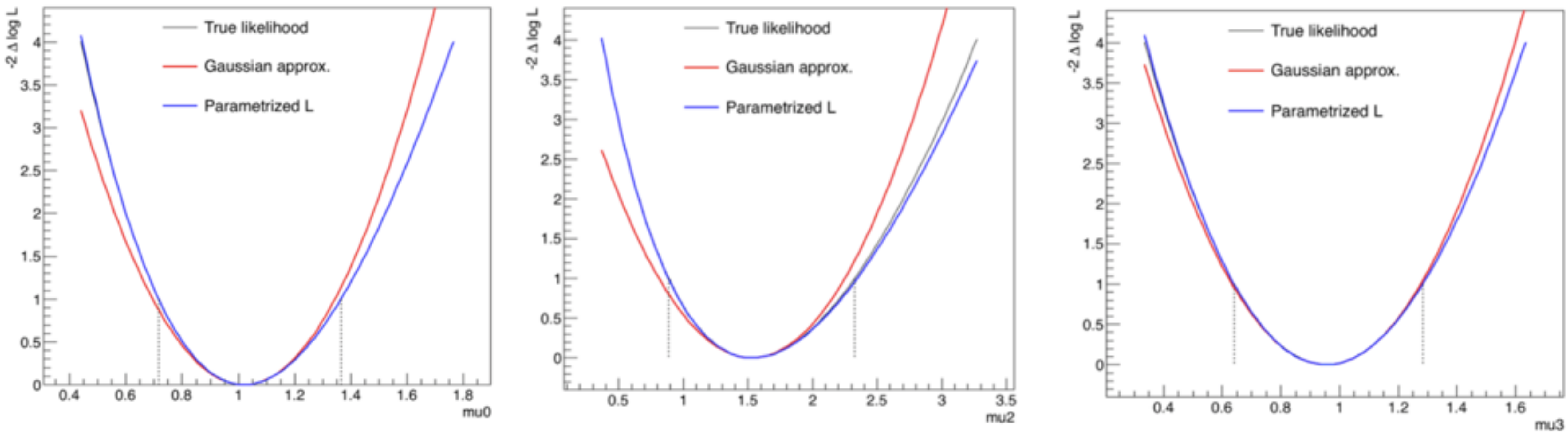
Purpose

- Create the tools necessary that parties interested in this data can build approximations/models of these likelihood curves
- This facilitates understanding and evaluation of how the data fits the standard model
- Create toy signals (Poisson distributions) that are “off-Gaussian” and develop ways to model non-Gaussian effects
- We eventually want to evaluate signal parameters that are correlated to each other, so we need to accurately evaluate correlated signals in multi-variate Gaussians (MVG)

How: μ-Models

- We can already correct for effects in 1-D

$$\mu_i = \frac{\sigma_i}{(\sigma_i)_{\rm SM}} \quad \text{and} \quad \mu^f = \frac{{\rm B}^f}{({\rm B}^f)_{\rm SM}}.$$



- Each plot is a log-likelihood distribution for a single Poisson, P(N | μ*S+B), with different N, S and B

- Black is true likelihood, red fits $-2 \cdot \ln \mathbf{\Lambda} = (\mu - \hat{u})^2/\sigma^2$ with a constant σ

- Blue accounts for $\sigma^2 =$ sigma_B^2 + μ*sigma_S^2+ μ^2*sigma_S_sys^2

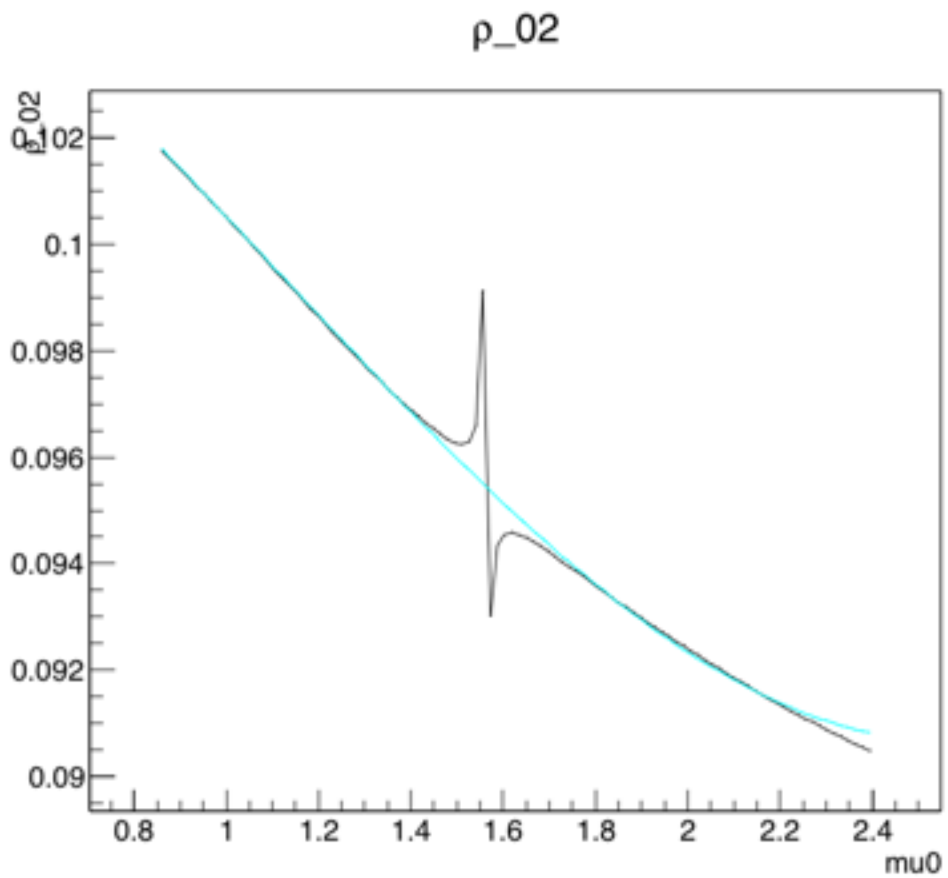
Correlation

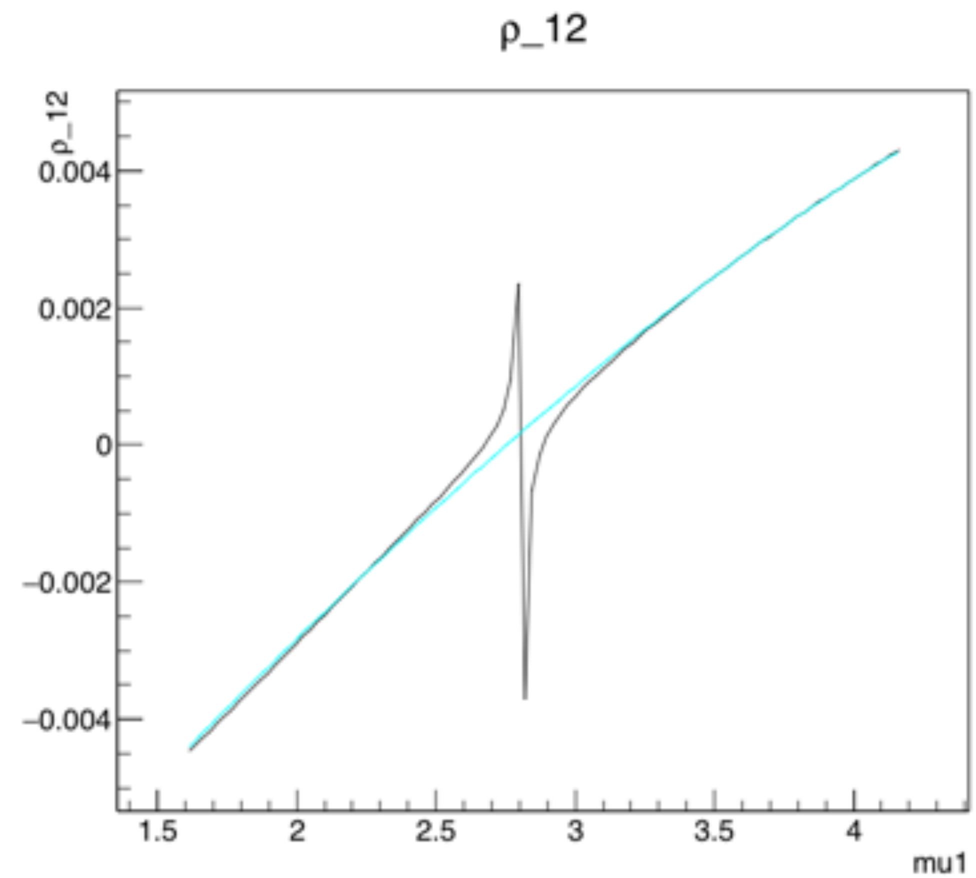
- Previously the correlation constant (ρ) between two signals (μ_i, μ_j) at the minimum was found using a function in RooFit, and used as a constant
- We want to find ρ analytically to see if we can confirm or improve the 2-D MVG approximations

$$\rho_{\mu_i,\mu_j}(\mu_i,\mu_j) = \frac{\hat{\hat{\mu}}_j(\mu_i) - \hat{\mu}_j}{\mu_i - \hat{\mu}_i} \cdot \frac{\sigma_{\mu_i}(\mu_i)}{\sigma_{\mu_j}(\hat{\hat{\mu}}_j(\mu_i))} = \frac{q_{\mu_j}(\hat{\hat{\mu}}_j(\mu_i))}{q_{\mu_i}(\mu_i)}$$

- This function diverges at $\hat{\mu}_i$ so first tried parametrizing this as a parabola (later returned to re-parametrize as third-degree polynomial)
- Note $\hat{\hat{\mu}}_j(\mu_i)$ is the value minimum value of μ_j for a given value of μ_i

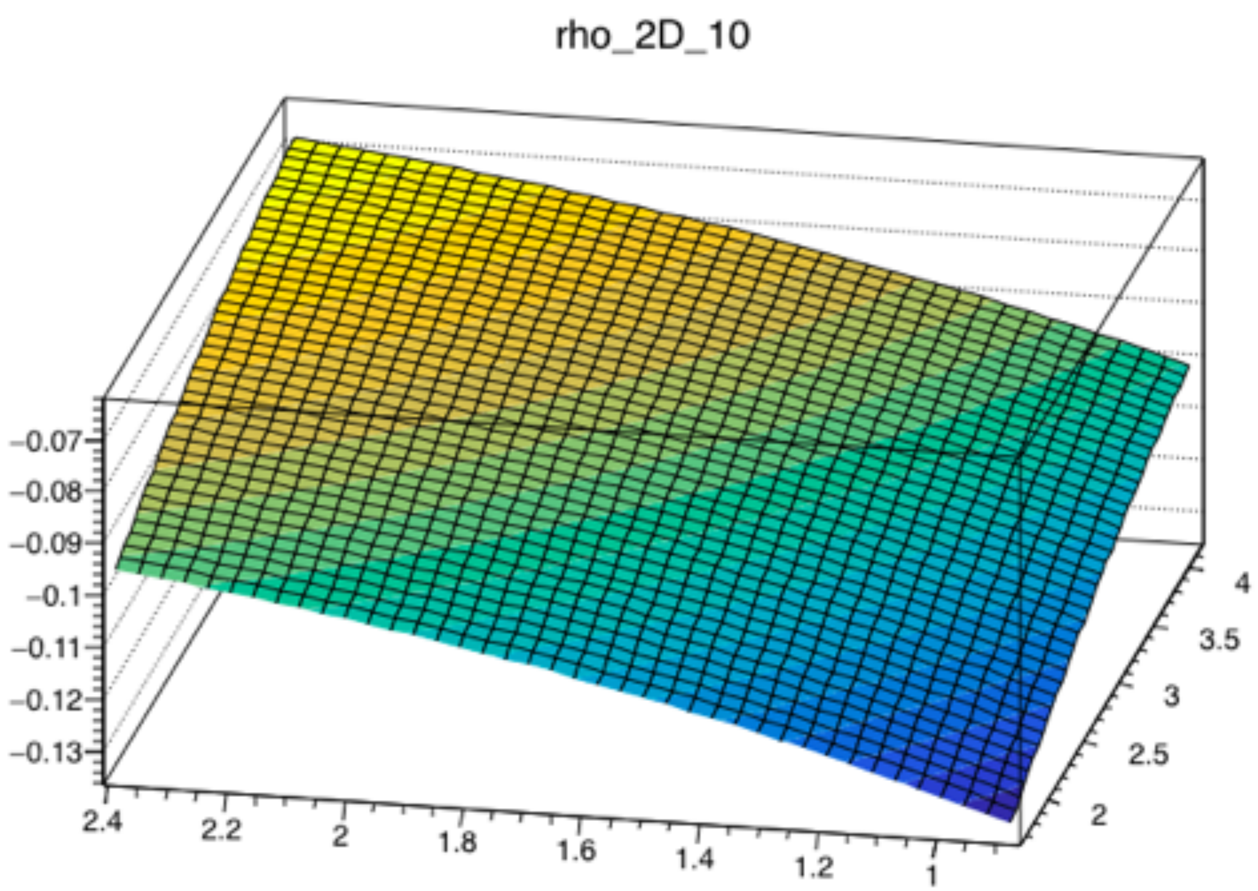
1-D Correlations



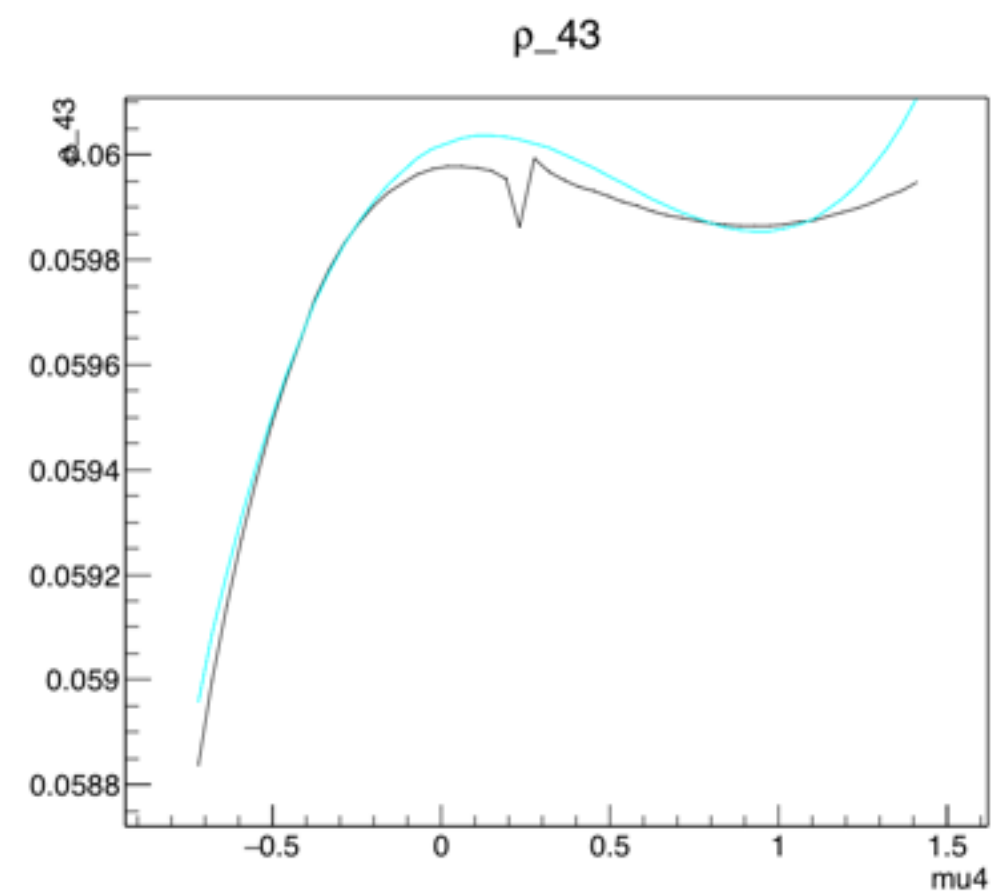


- Now we recognize that these are slices of the 2-D correlation between μ_i and μ_j
- We are interested in this ρ_{ij}, especially to have it only depend on μ_i, and μ_j for the independent variables

2-D Correlations

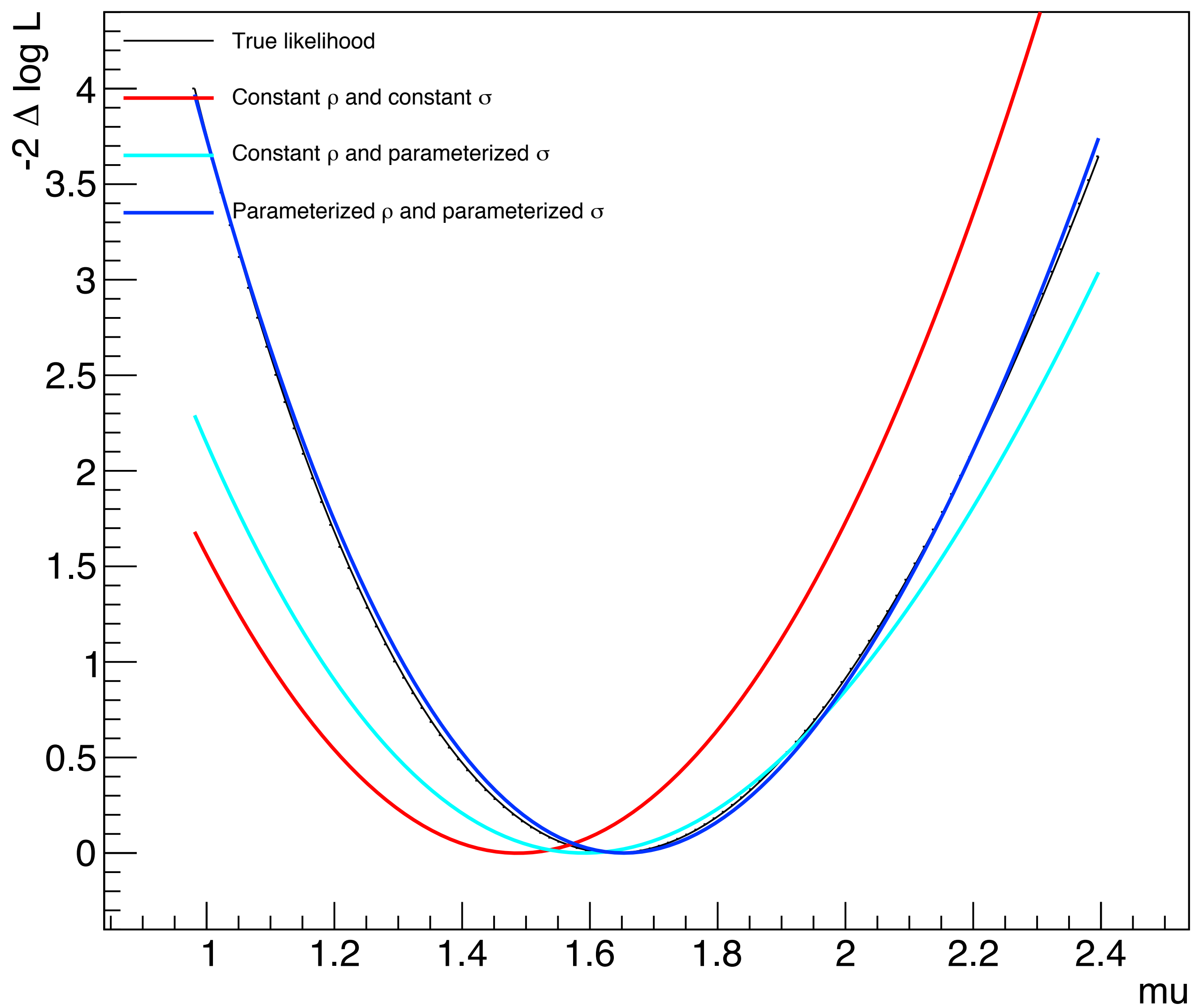


- Adjust the original function for creating the MVG to accept the 2D $\rho_{i,j}$, rather than the constant passed by RooFit



- Also: noted need to re-parametrize the 1-D function for ρ to the third order

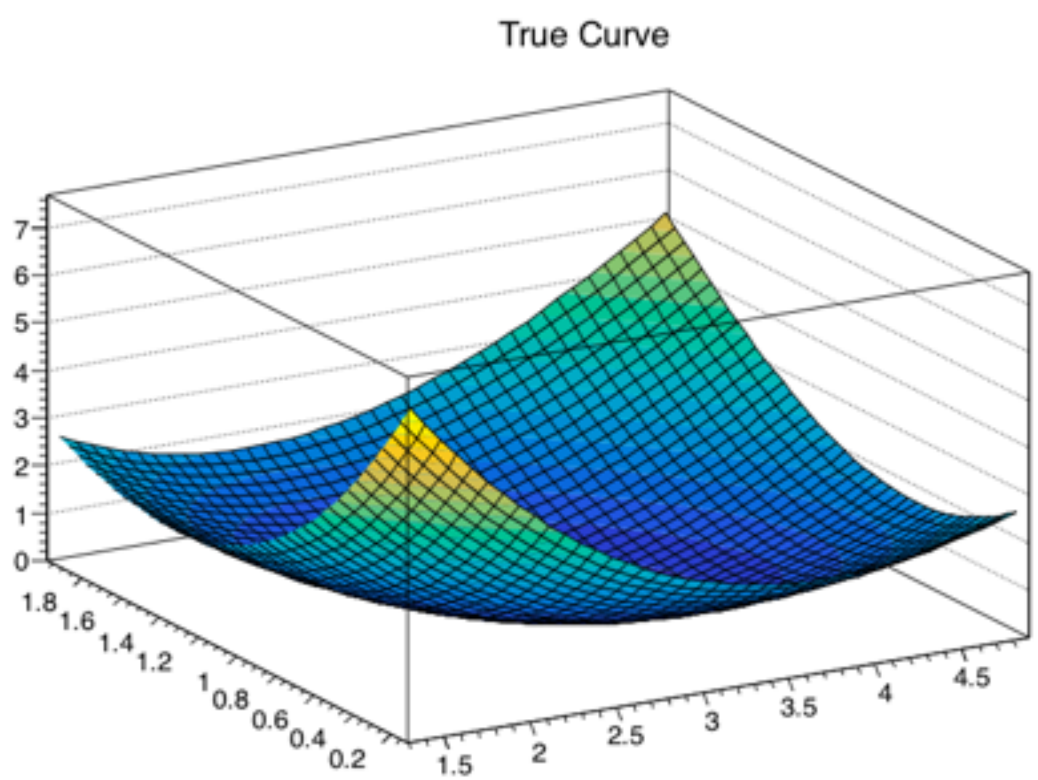
Simplify: 2x2 or “2-μ” Model

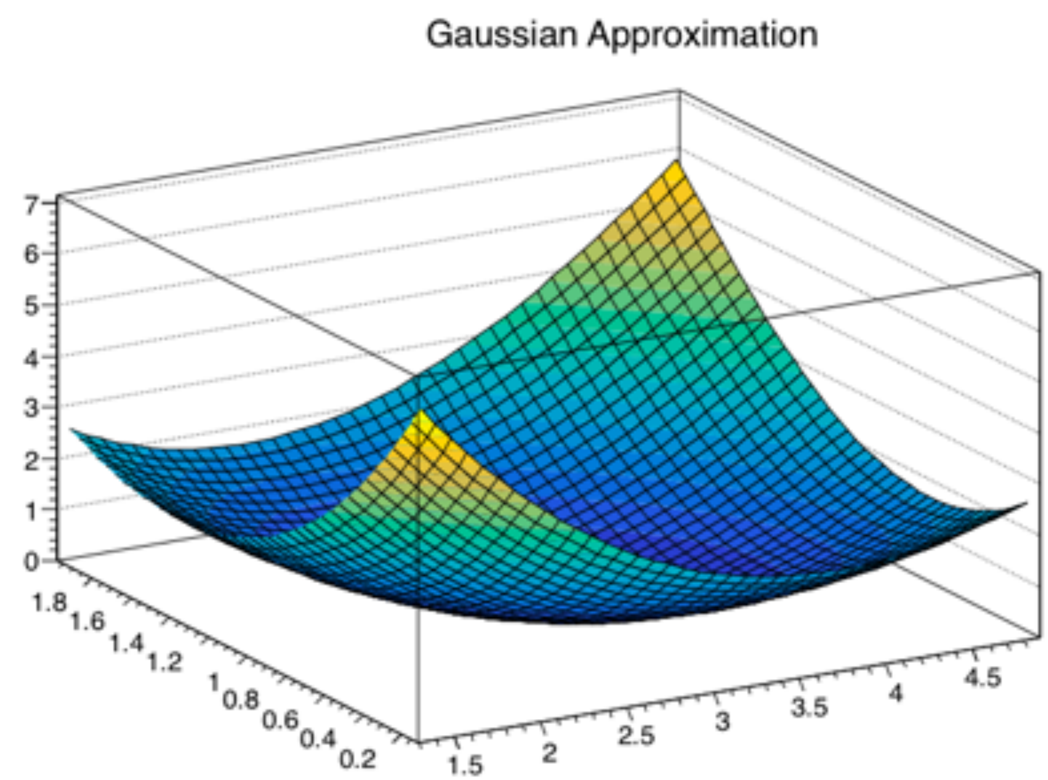


- Backtrack: move from model with five correlated signals to two
- Using only two signals, make an MVG with this new 2-D ρ_{ij}
- Evaluate the fit using $\mu_0 = \mu_1 = \mu$
- Shows excellent improvement

Simplify: 2x2 or “2-μ” Model

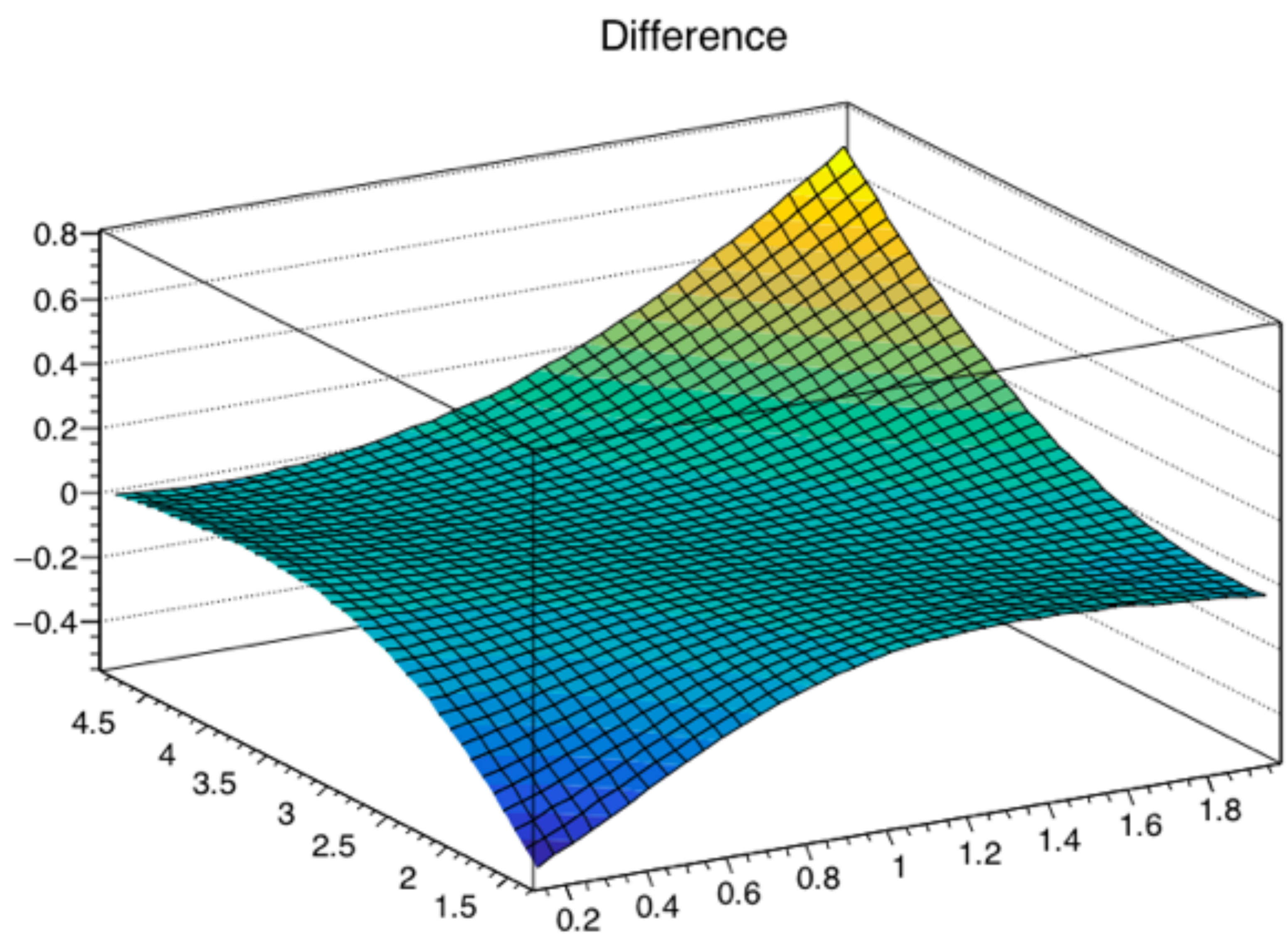
- The real test: build in two dimensions
- Compare our MVG with true 2-D distribution





Simplify: 2x2 or “2 μ” Model

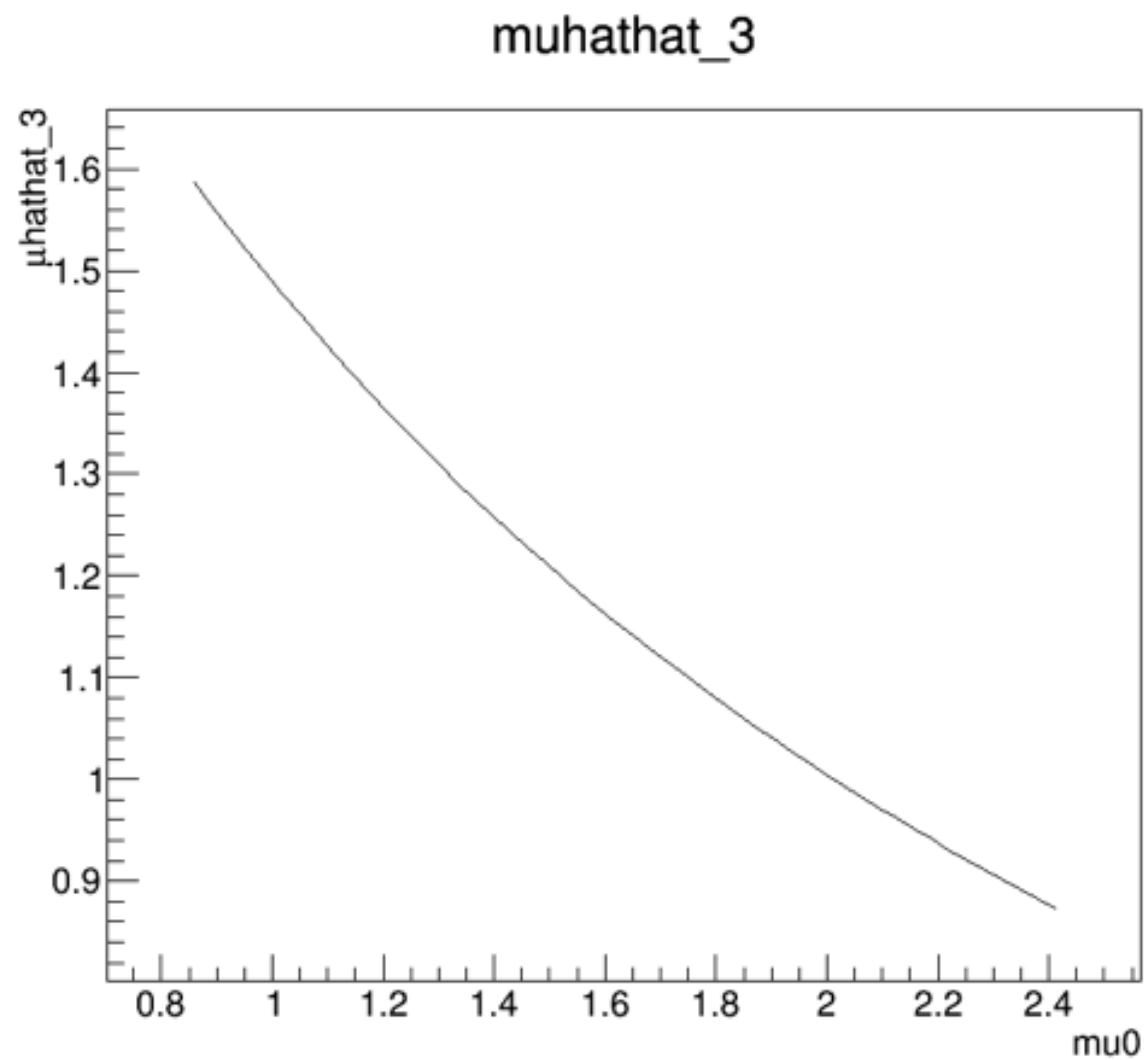
- The real test: build in two dimensions
- Compare our MVG with true 2-D distribution
- Strained at corners: likely due to the equation that transforms 1-D ρs to 2-D ρ, which assumed $\hat{\hat{\mu}}_j(\mu_i)$ was linear ….



Simplify: 2x2 or “2 μ” Model

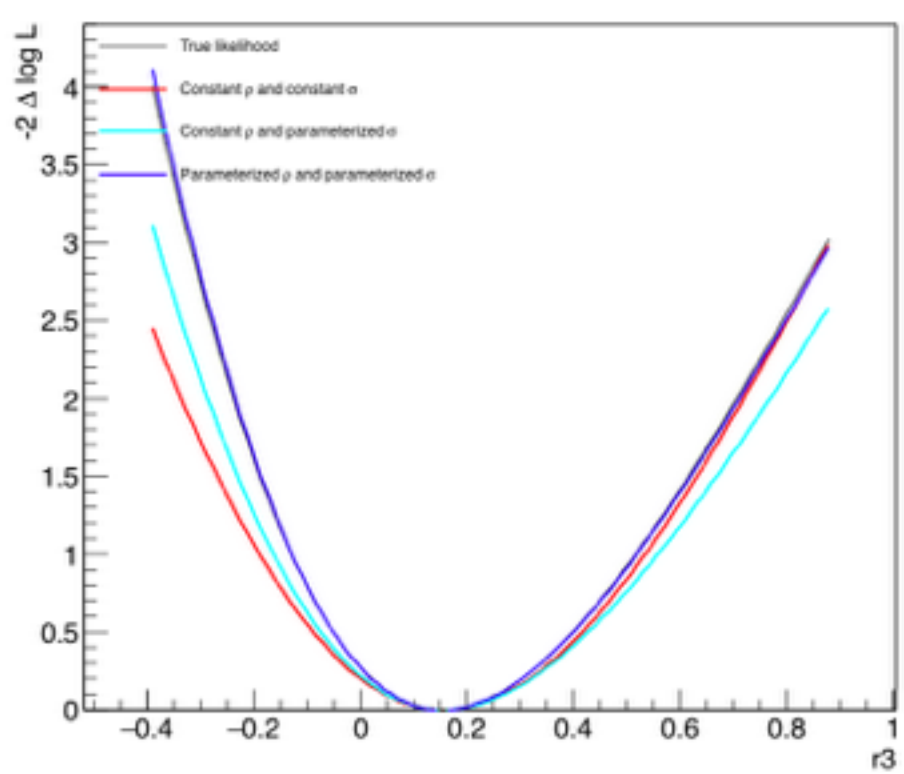
- The real test: build in two dimensions
- Compare our MVG with true 2-D distribution
- Strained at corners: likely due to the equation that transforms 1-D ρs to 2-D ρ, which assumed $\hat{\hat{\mu}}_j(\mu_i)$ was linear ….

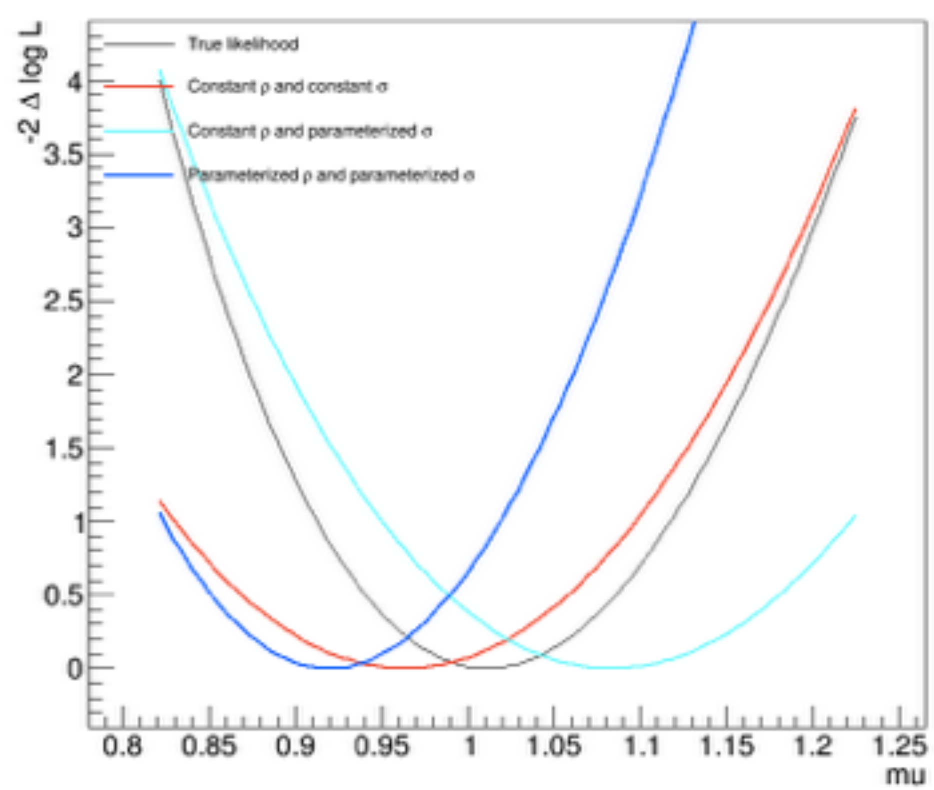
…but it’s not



Un-Simplify: Check 5-μ Model

- 5-μ model passing parametrization/slice tests that deal with ratios quite well, but not the troublesome "overall-μ" (which is the diagonal slice of the 2-D graph)

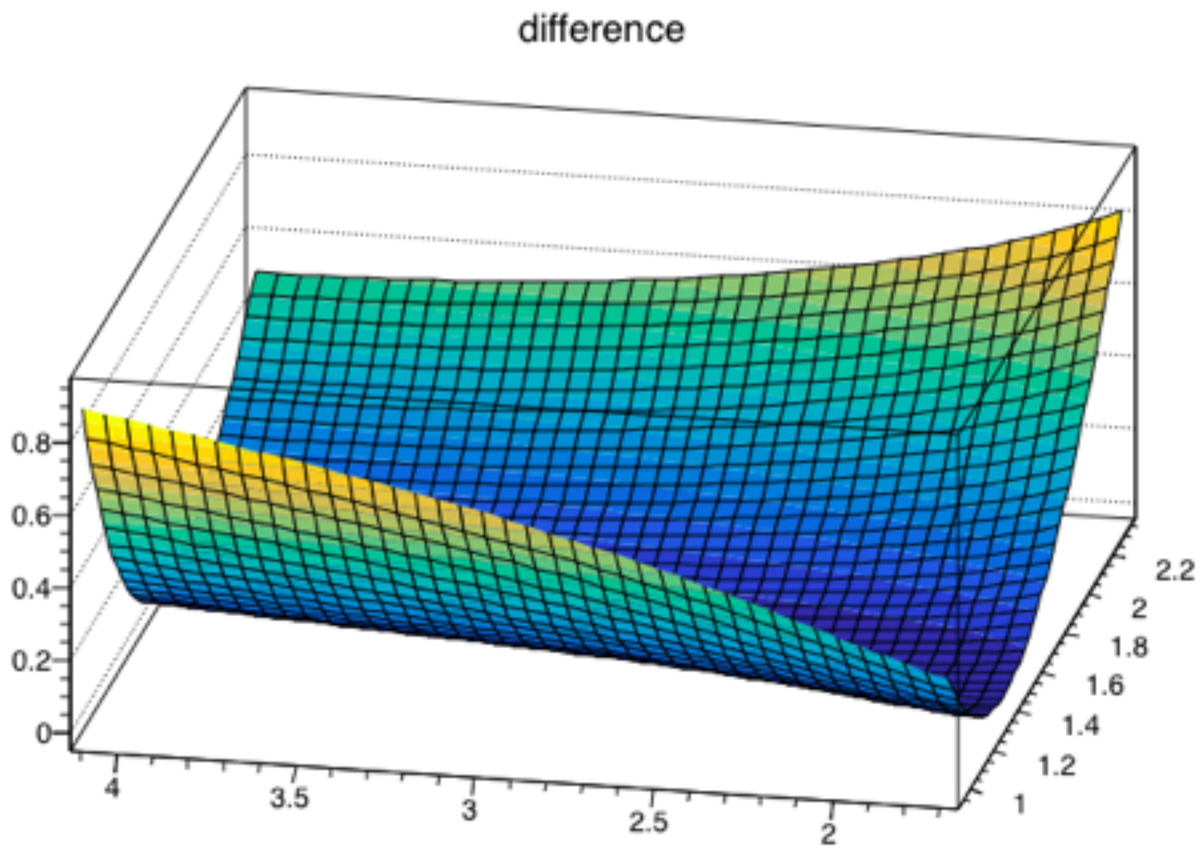


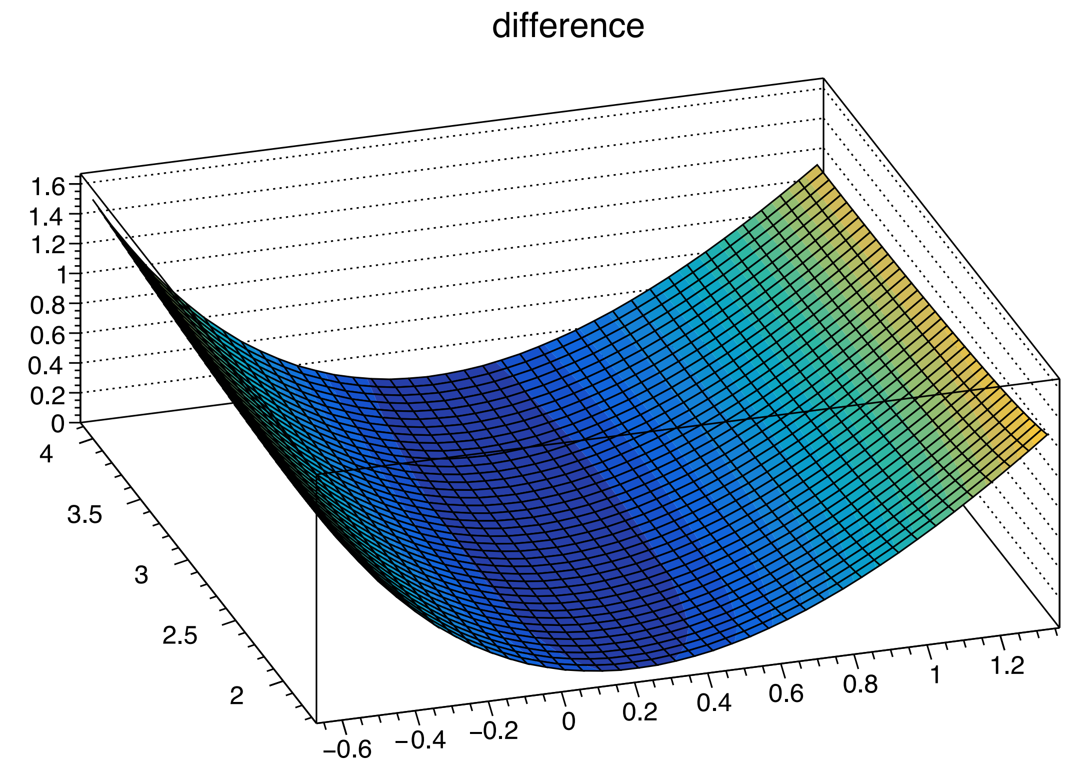


Un-Simplify: Check 5-μ Model

- 5-μ model passing parametrization/slice tests that deal with ratios quite well, but not the troublesome “overall-μ” (which is the diagonal slice of the 2-D graph)

- ….And the 2-D MVG does not do the job

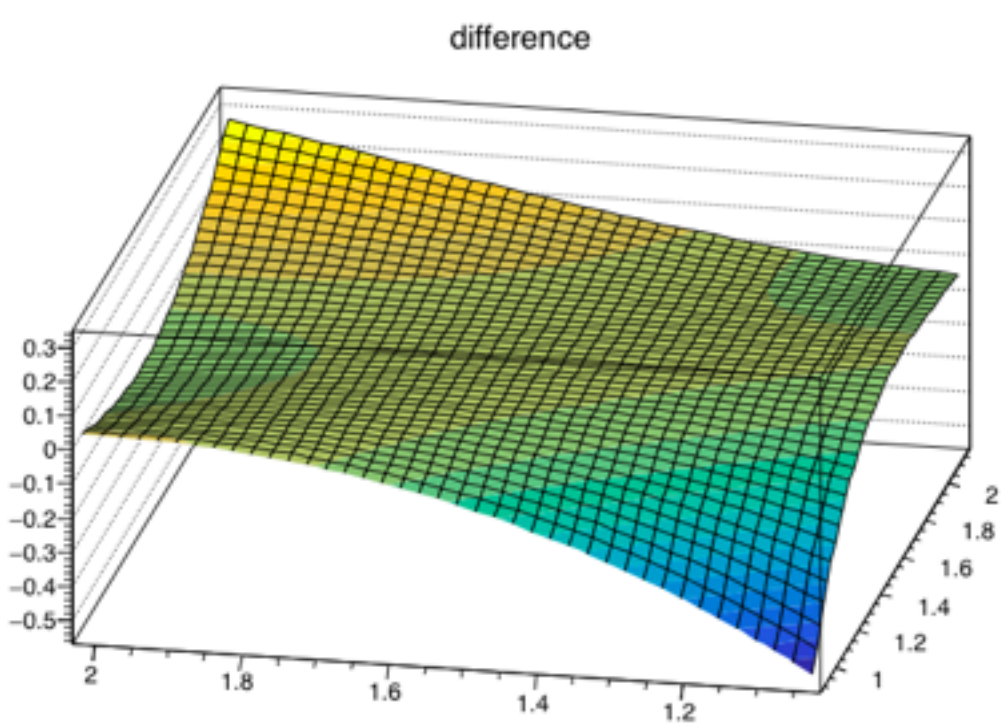


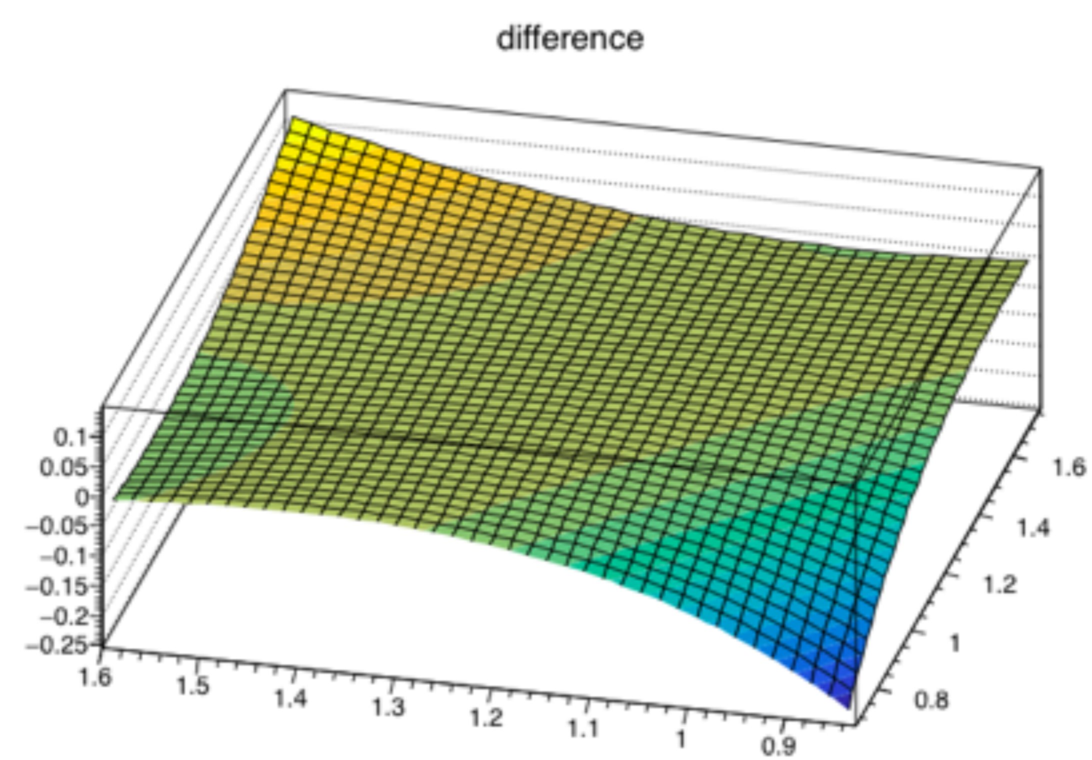


Examine: 4-μ Model

- Could be due to multiple signals correlated with each other, so a 4-μ model with two pairs of signals correlated to within the pair (but not in between) was examined.

	0	1	2	3
0	1	-0.38102	0	0
1	-0.379109	1	0	0
2	0	0	1	-0.45778
3	0	0	-0.45667	1

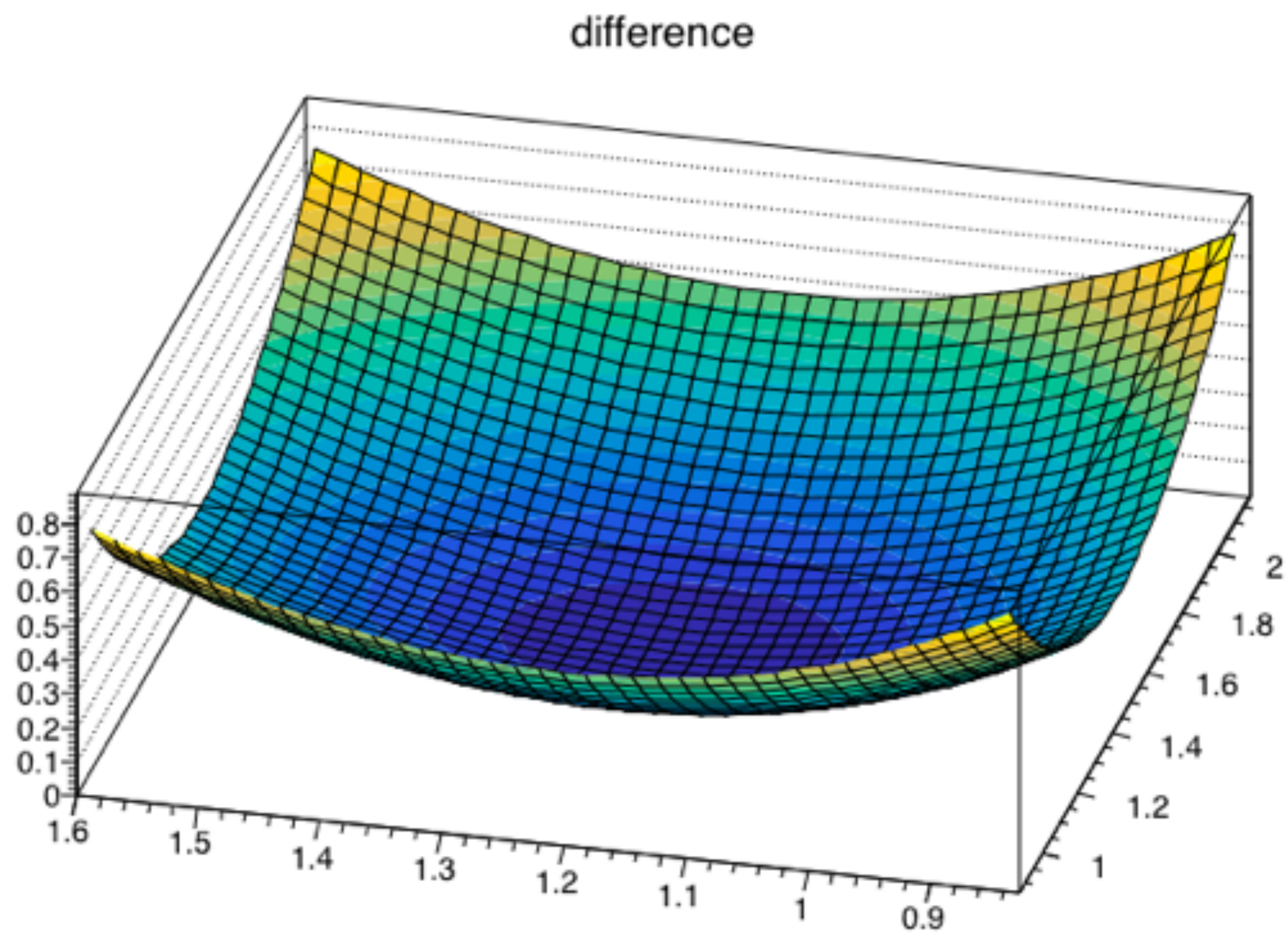




- The pairs correlated to each other had good MVG approximations (except for the “corner problem”)

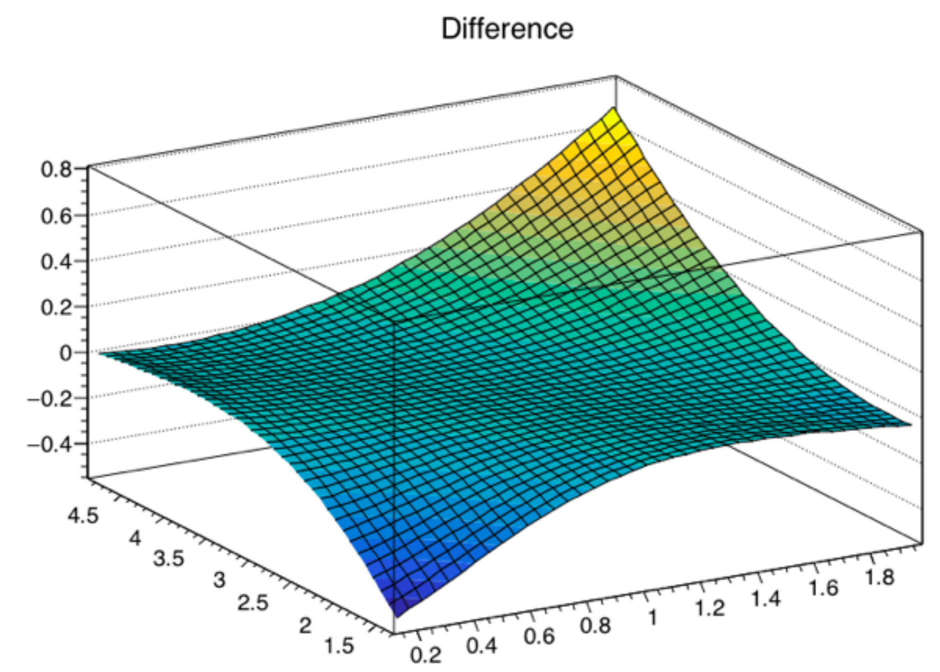
Examine: 4-μ Model

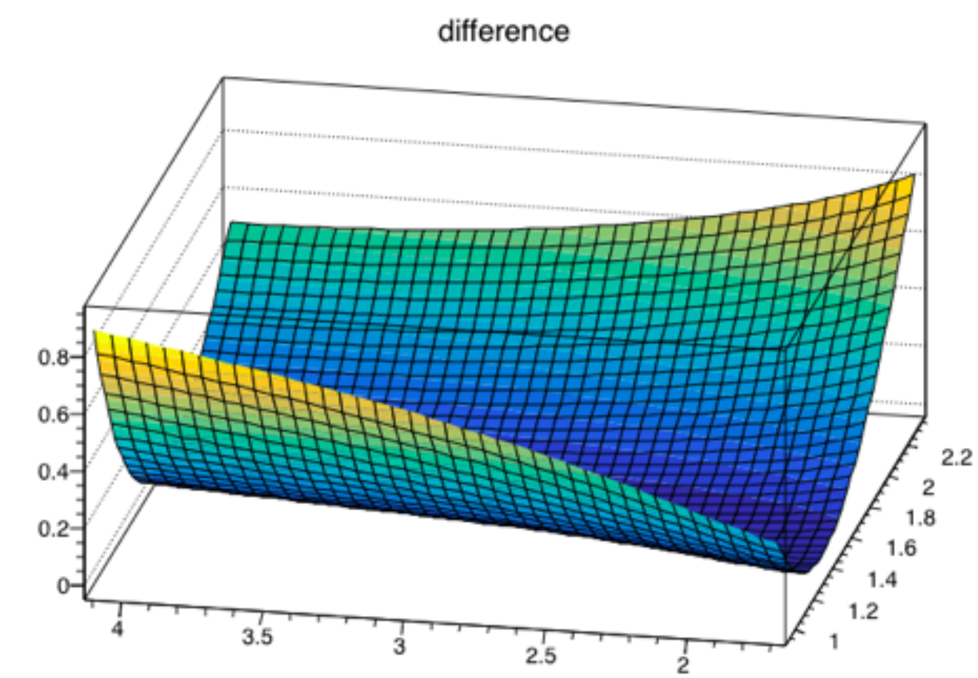
- And the difference between signals from different pairs yielded an MVG with a correlation of 0, according to expectations
- This suggests that multiple signals correlated to each other might contribute to the problem



Next

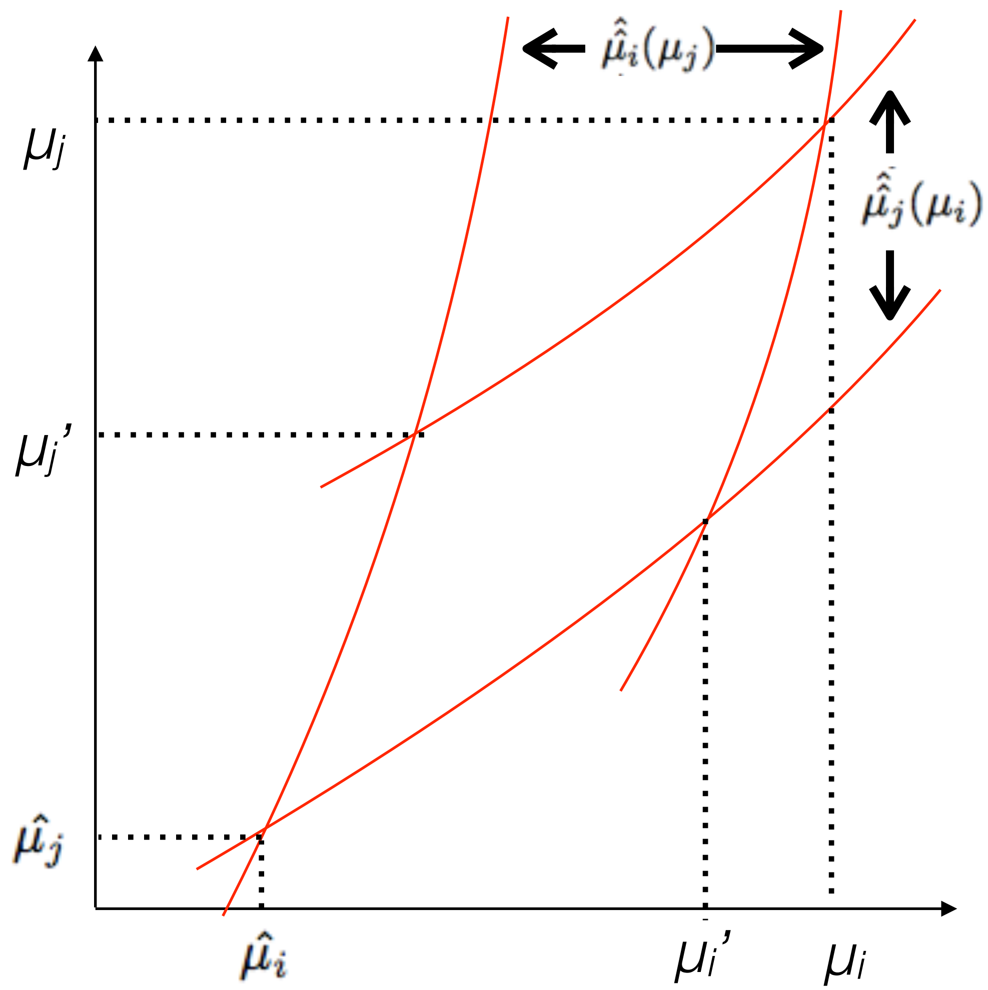
- Find a more accurate way of extending 1-D ρ to 2-D ρ
  - If we assume $\hat{\hat{\mu}}_j(\mu_i)$ is not linear, we must invert this transcendental function to solve for a 2-D rho
  - There is a class that does this that we are looking into using to hopefully solve the "corner problem"
- Then return to 5-μ model to see if further steps are necessary





Backup

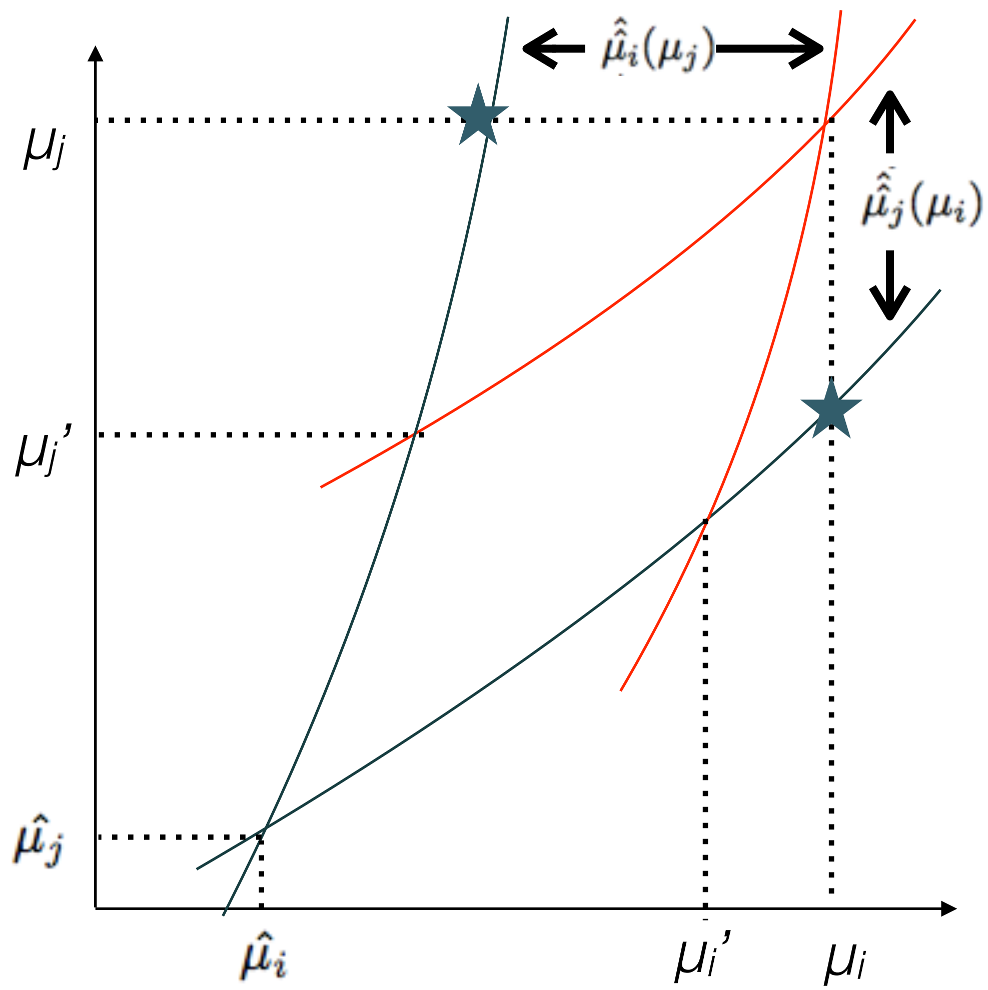
Jumping dimensions



- To build a 2-D ρ, the 1-D ρs and $\hat{\hat{\mu}}_i$ curves are used (these shift uniformly as $\hat{\mu}_i$ is increased
- We have $\hat{\hat{\mu}}_i(\mu_j)$ & $\hat{\hat{\mu}}_j(\mu_i)$ for our arbitrary (μ_i, μ_j)
- We "travel" along these curves to solve for (μ_i', μ_j')
- First find points by by assuming $\hat{\hat{\mu}}_i(\mu_j)$ & $\hat{\hat{\mu}}_j(\mu_i)$ are roughly linear to avoid inverting this complicated function

- Once we have (μ_i', μ_j'), we can approximate $\Delta\rho_{2D} \approx \Delta\rho_{i,j} + \Delta\rho_{j,i}$
- Thus $\rho_{2D} = \rho_{i,j}(\mu_i') + \rho_{j,i}(\mu_i') - \rho(\hat{\mu}_i, \hat{\mu}_j)$

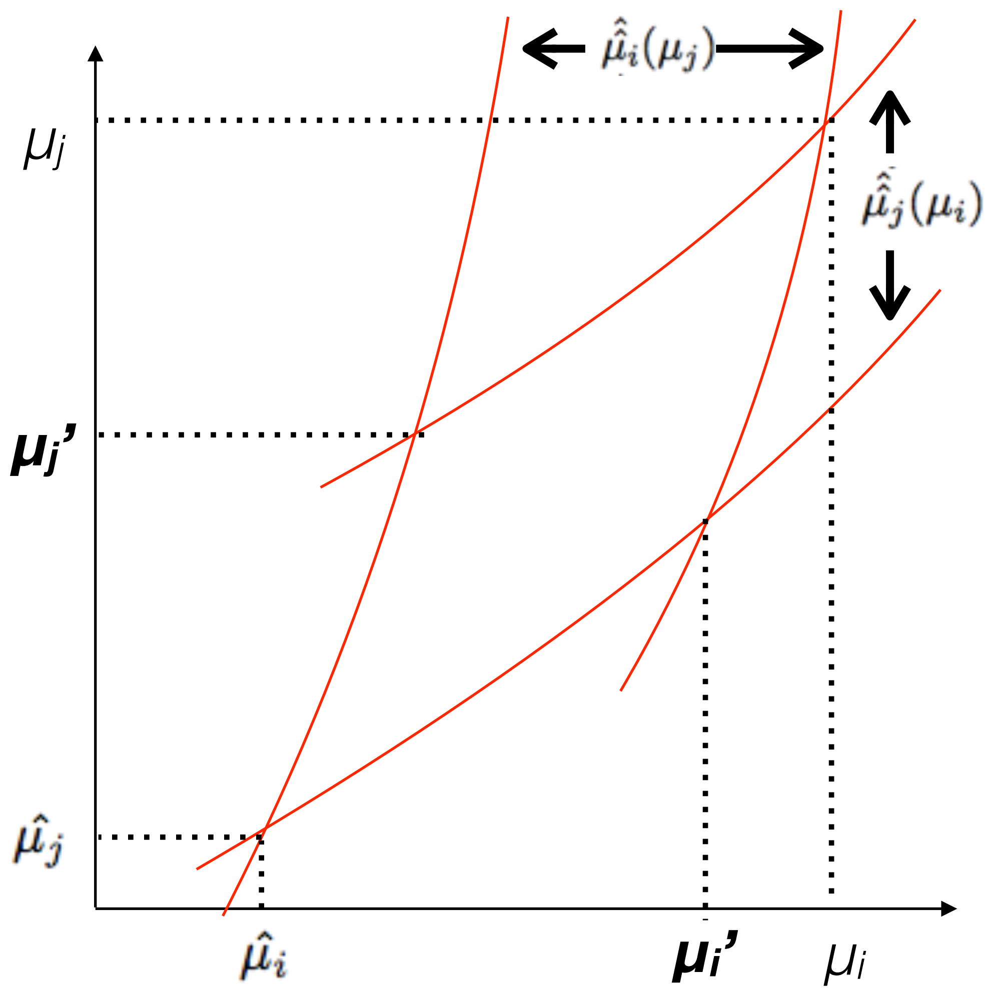
Jumping dimensions



- To build a 2-D ρ, the 1-D ρs and $\hat{\hat{\mu}}_i$ curves are used (these shift uniformly as $\hat{\mu}_i$ is increased
- We have $\hat{\hat{\mu}}_i(\mu_j)$ & $\hat{\hat{\mu}}_j(\mu_i)$ for our arbitrary (μ_i, μ_j)
- We "travel" along these curves to solve for (μ_i', μ_j')
- First find points by by assuming $\hat{\hat{\mu}}_i(\mu_j)$ & $\hat{\hat{\mu}}_j(\mu_i)$ are roughly linear to avoid inverting this complicated function

- Once we have (μ_i', μ_j'), we can approximate $\Delta\rho_{2D} \approx \Delta\rho_{i,j} + \Delta\rho_{j,i}$
- Thus $\rho_{2D} = \rho_{i,j}(\mu_i') + \rho_{j,i}(\mu_i') - \rho(\hat{\mu}_i, \hat{\mu}_j)$

Jumping dimensions



- To build a 2-D ρ, the 1-D ρs and $\hat{\hat{\mu}}_i$ curves are used (these shift uniformly as $\hat{\mu}_i$ is increased
- We have $\hat{\hat{\mu}}_i(\mu_j)$ & $\hat{\hat{\mu}}_j(\mu_i)$ for our arbitrary (μ_i, μ_j)
- We "travel" along these curves to solve for (μ_i', μ_j')
- First find points by by assuming $\hat{\hat{\mu}}_i(\mu_j)$ & $\hat{\hat{\mu}}_j(\mu_i)$ are roughly linear to avoid inverting this complicated function

- Once we have (μ_i', μ_j'), we can approximate $\Delta\rho_{2D} \approx \Delta\rho_{i,j} + \Delta\rho_{j,i}$
- Thus $\rho_{2D} = \rho_{i,j}(\mu_i') + \rho_{j,i}(\mu_i') - \rho(\hat{\mu}_i, \hat{\mu}_j)$