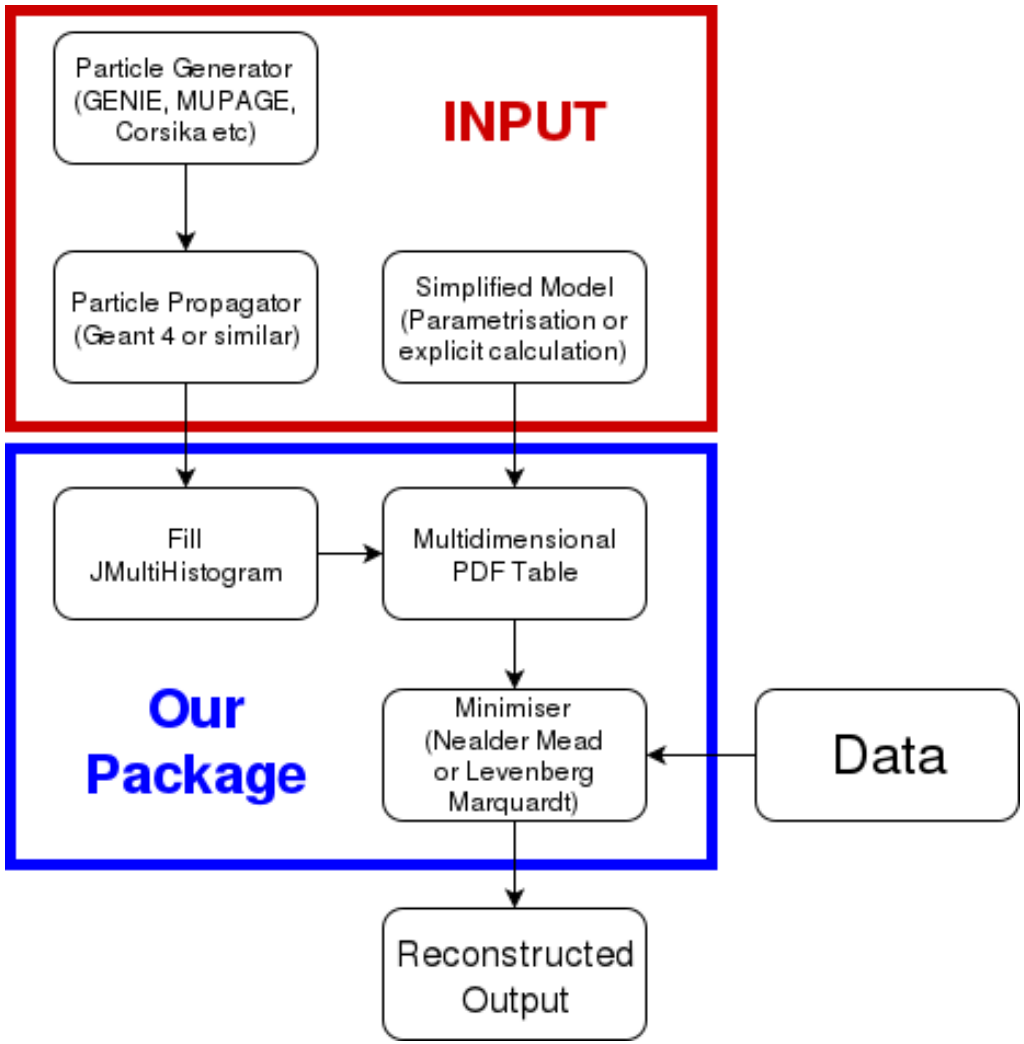
Package for Likelihood-Based Fitting Methods

Liam Quinn

October 15, 2017

Package for Likelihood-Based Fitting Methods
Specifications

- Software package written in C++
- Contains classes and executables for
- Building and interpolating over multidimensional probability density functions
  - PDFs can either be explicitly calculated or filled from simulation
- Multidimensional function minimisation
  - Levenberg Marquardt and Nealder Mead methods supported
  - Object-oriented approach, use of templates for flexibility
  - 3D geometry classes included
- Based on existing KM3NeT software, with a broader scope, by Maarten de Jong.


Particle Generator (GENIE, MUPAGE, Corsika etc)
INPUT
Particle Propagator (Geant 4 or similar)
Simplified Model (Parametrisation or explicit calculation)
Fill JMultiHistogram
Multidimensional PDF Table
Our Package
Minimiser (Nealder Mead or Levenberg Marquardt)
Data
Reconstructed Output

Maximum Likelihood: $\mathcal{L}(\theta; \mathbf{x}) = \prod_i^N \mathcal{P}(x_i|\theta)$

Muons: $x_i = \{t, R, \theta_{PMT}, \phi_{PMT}\}$

Cascades: $x_i = \{t, D, \cos\theta_{\text{emit}}, \theta_{PMT}, \phi_{PMT}\}$

Requirements:

- Parametrise $\mathcal{P}$, many function calls to reach maximum
- Need to evaluate $\mathcal{P}$ quickly and accurately

Solution:

- Multidimensional interpolation tabulated values
- Can also be used in simulations

- Functional operation $(x_0, x_1, x_2, ..., x_N) \rightarrow y$, a set of abscissa values is mapped to an ordinate value
- Underlying structure based on STL containers
  - Can always be iterated through, like an STL vector
- Spline interpolation or polynomial of orders 0-3
- JMultiHistogram class
  - Can be filled like a ROOT histogram and then converted into PDF

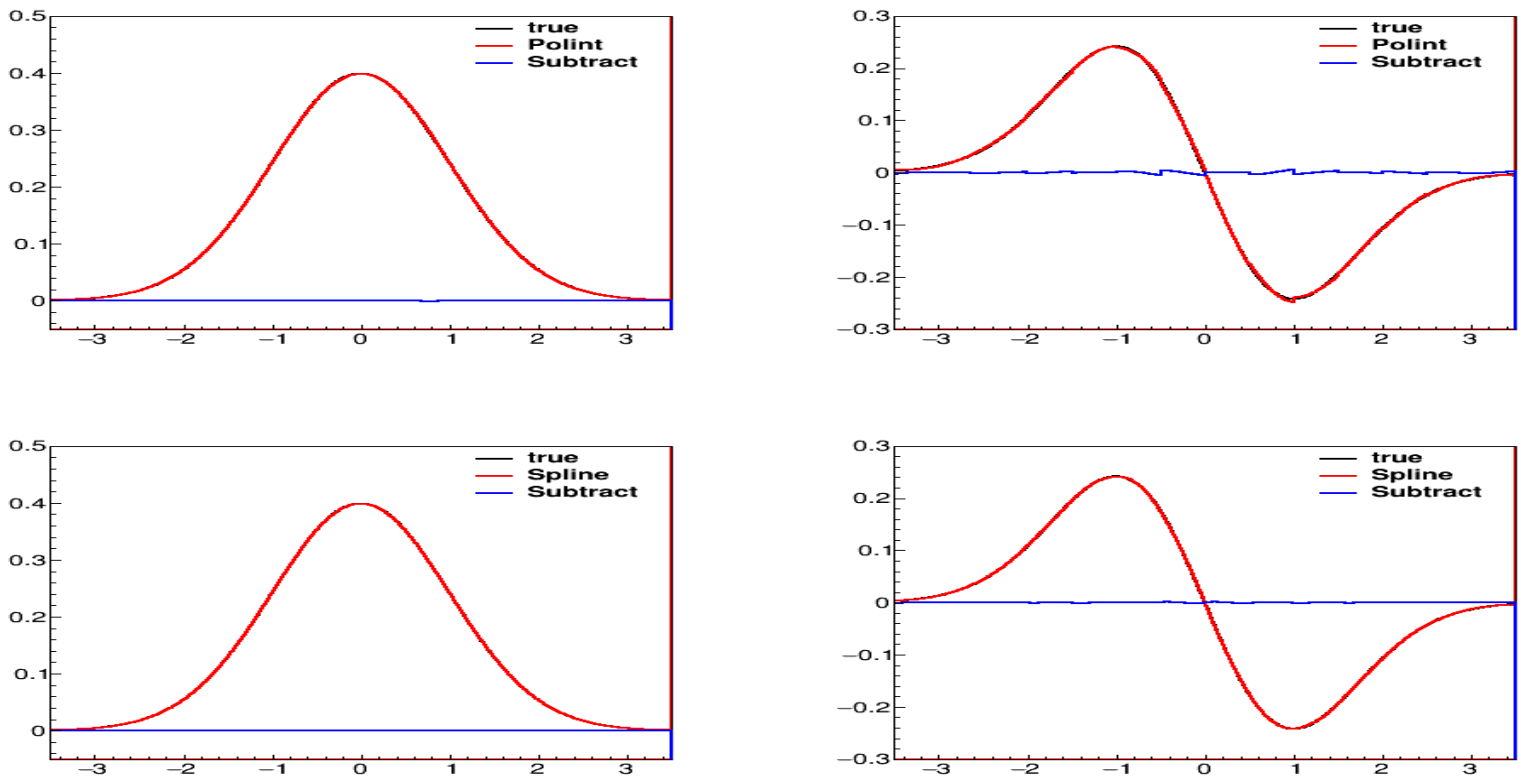


Figure 1 : Example showing spline and 2nd order polynomial interpolation for a Gaussian and its derivative.

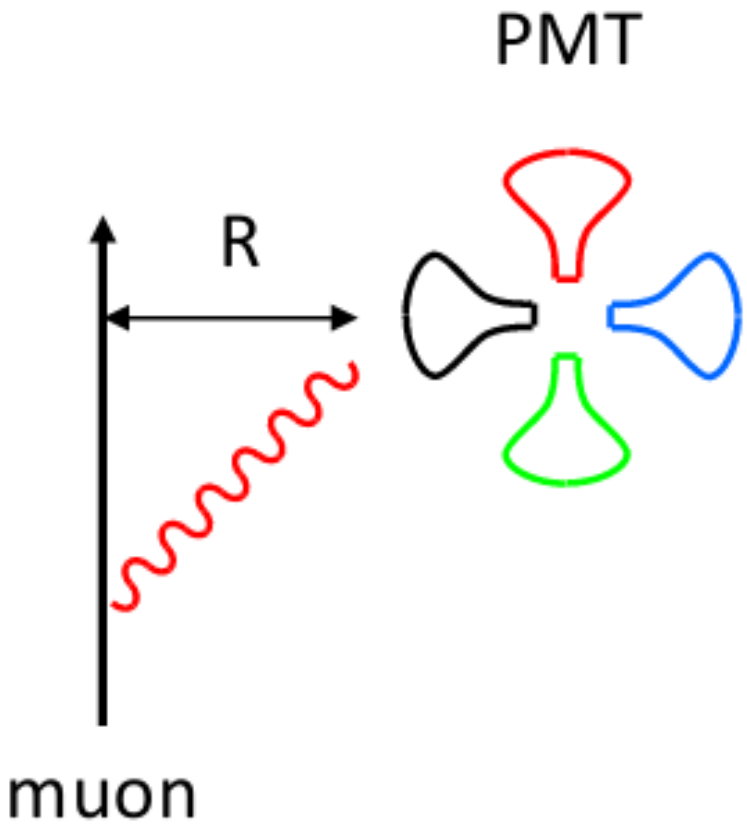
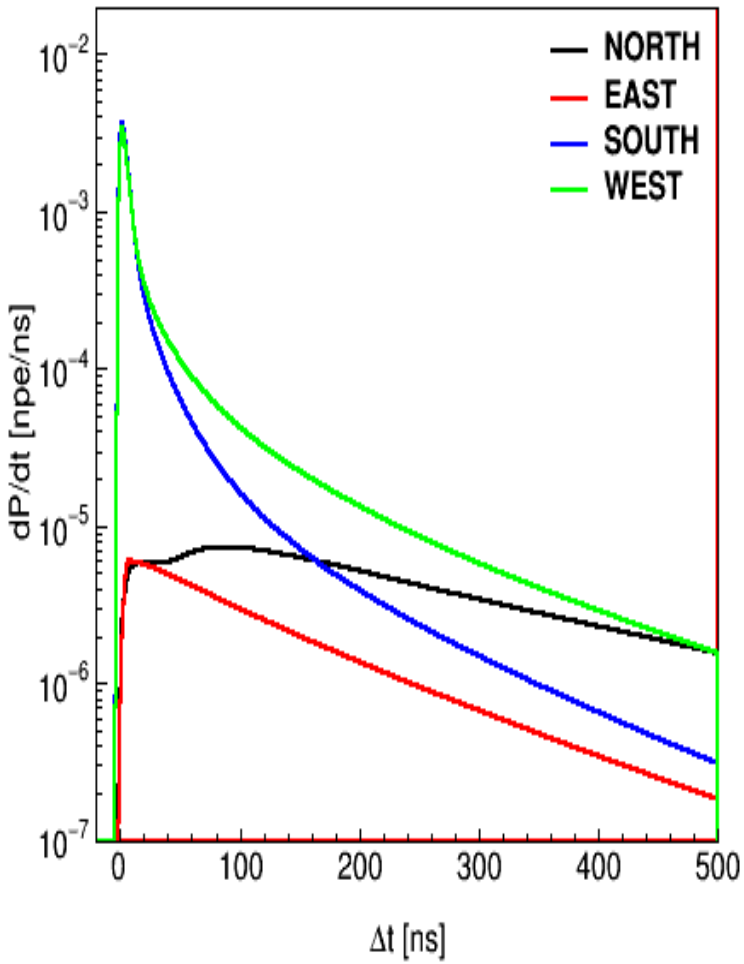
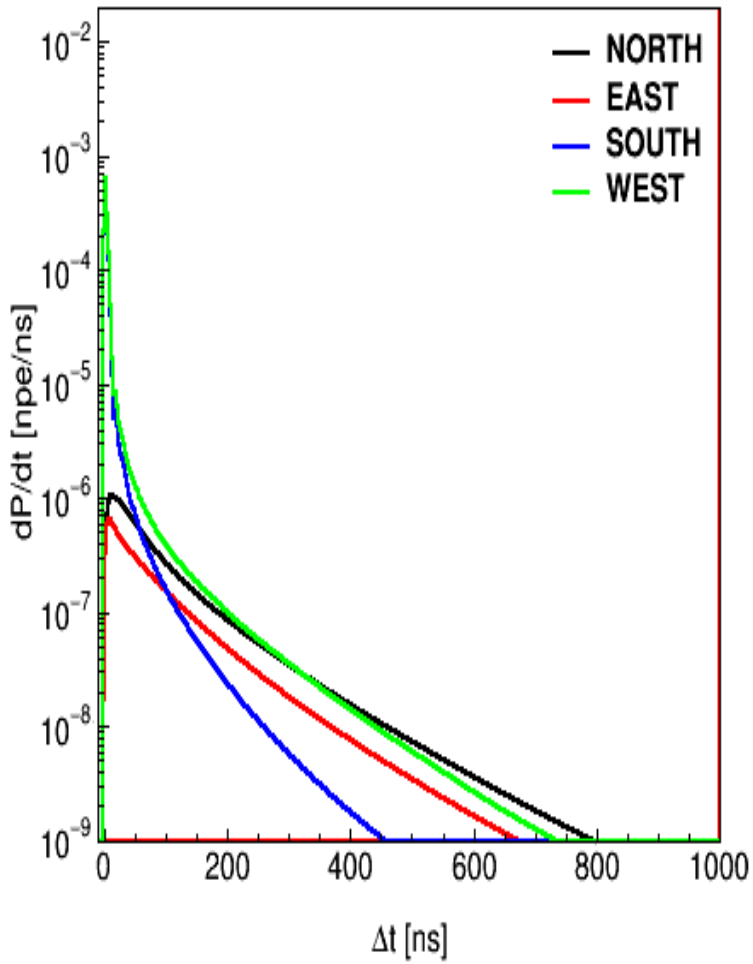


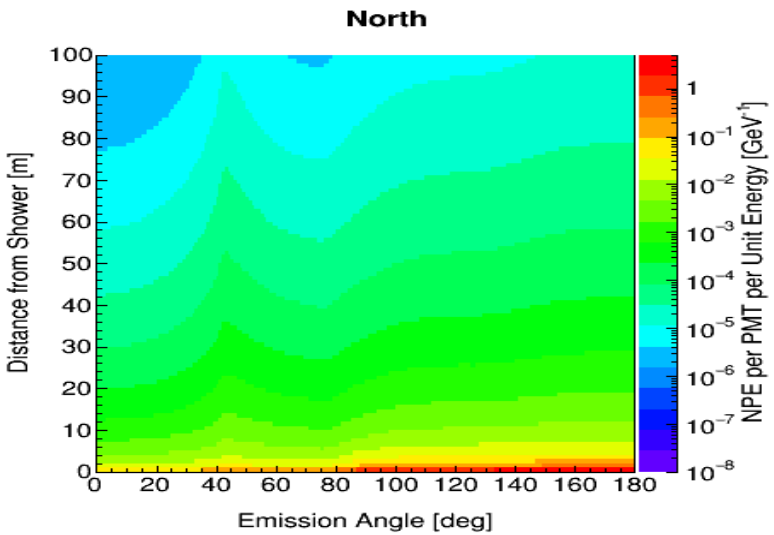
Figure 2 : PMT orientation convention with respect to the incoming lepton

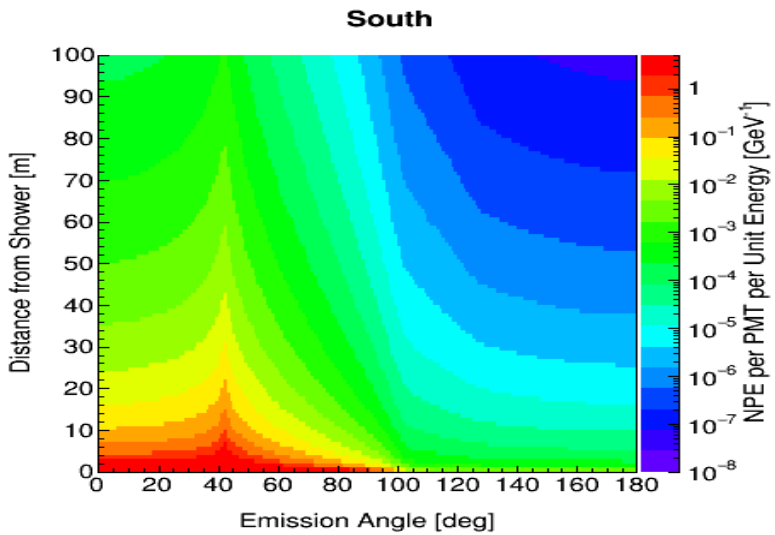
Probability Density Functions
Photon Arrival Times

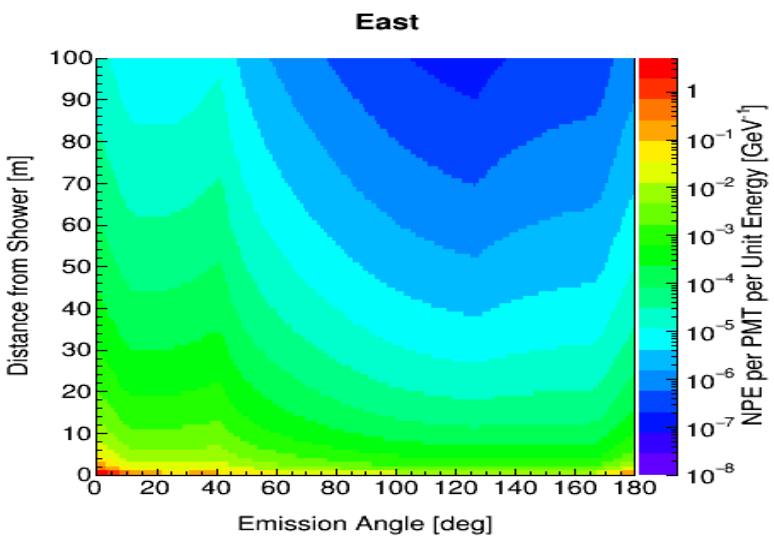


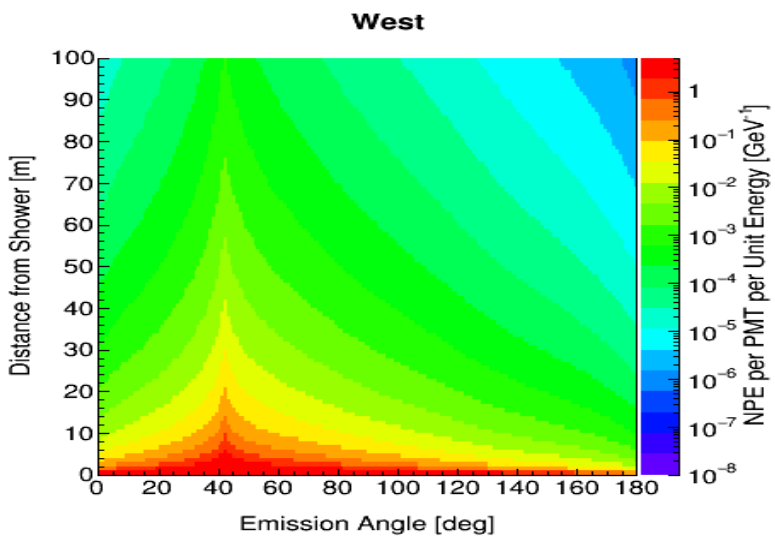


Probability Density Functions
EM Showers in Water









```
template<class JModel_t>
class JMinimiser
{
...
  template<class JFunction_t, class T>
  double operator()(const JFunction_t& fit,
                    T __begin, T __end) {
    ...
    return chi2;
  }
};
```

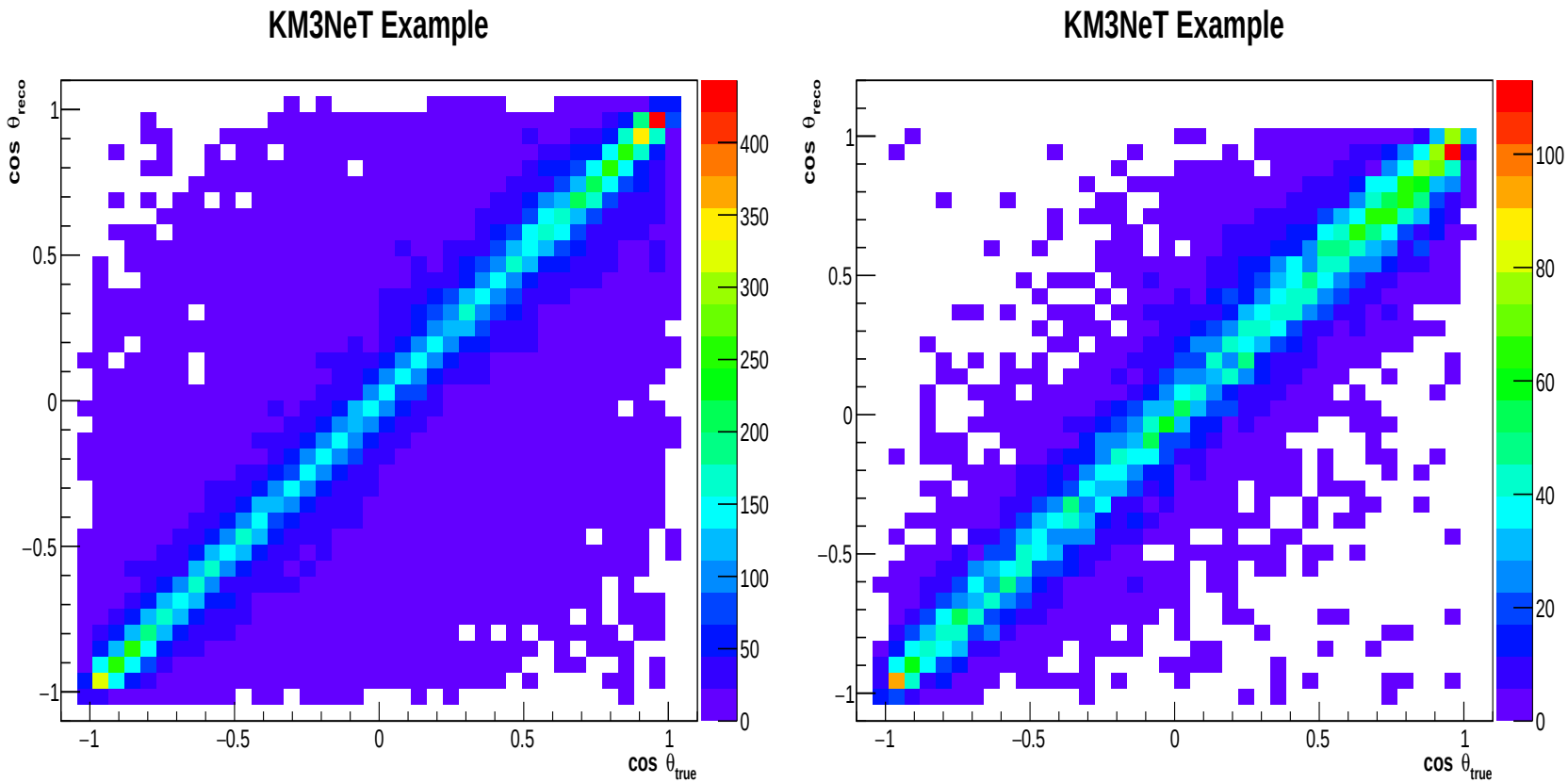
JModel_t	A geometrical description of the event, such as a point, line etc
JFunction_t	A class which contains the fit function as a member
T	An iterator of an stl vector containing the data

The fit function is of the form:

```
template<class JHit_t>
result_type operator()(const JModel_t& track,
                       const JHit_t hit) {
   ...
   return result;
}
```

Nealder Mead: Result type is a double
Levenberg Marquardt: Result type also contains the gradient of the likelihood

- Object oriented approach - avoids use of global variables, such as in Minuit
- Templated structure allows for flexibility



Zenith response for muon neutrinos (left) and electron neutrinos (right), simulated in the 1-100GeV range.