FITQUN AND PYFITQUN PROGRESS FOR WCTE

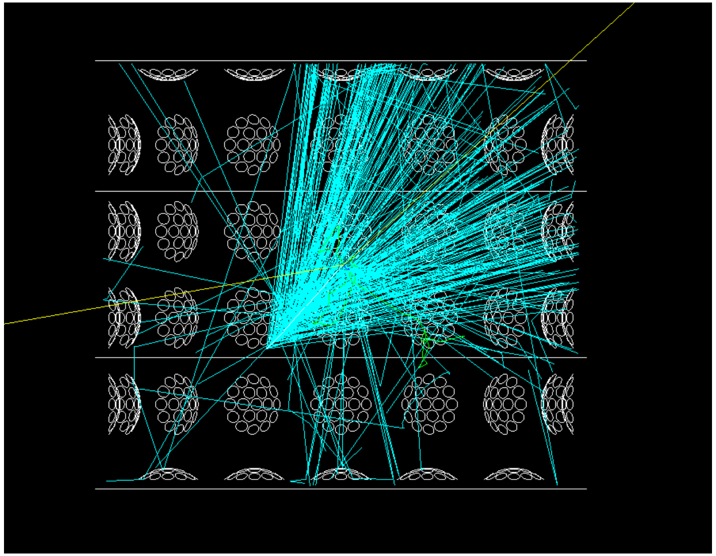
Gonzalo Díaz López

Neutrino group meeting

24 April 2024

I. FITQUN

FITQUN ALGORITHM



Example: simulated μ- for WCTE geometry

Read event hits (time, charge)

Prefit : use time to estimate vertex (t, **x**)

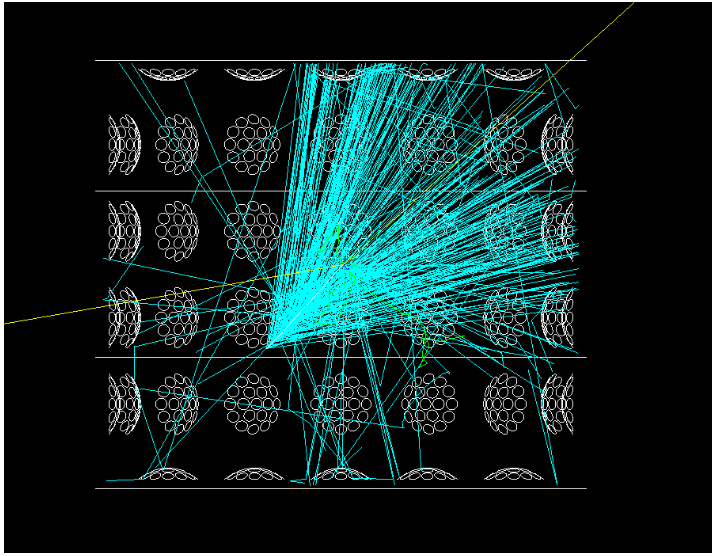
Subeventing

- The peak finder
- Hit clustering

Single ring fits (t, **x**, **p**)

Multi ring fits

FITQUN ALGORITHM



Example: simulated μ- for WCTE geometry

$$\mathbf{X} \equiv (\mathbf{x}, \mathbf{p})_{\text{ring}}$$

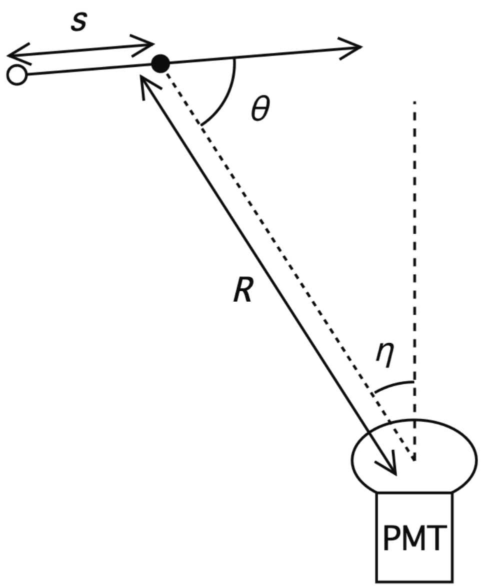
$$\mathcal{L}(\mathbf{X}) = \prod_j^{n_{\text{unhit}}} P_j(\text{unhit}|\mathbf{X}) \prod_i^{n_{\text{hit}}} (1 - P_i(\text{unhit}|\mathbf{X}))\ f_q(q_i|\mathbf{X})\ f_t(t_i|\mathbf{X})$$

$$\mu_i \equiv \mu_i(\mathbf{X})$$ **predicted charge**

$$\mathcal{L}(\mathbf{X}) = \prod_j^{n_{\text{unhit}}} P(\text{unhit}|\mu_j) \prod_i^{n_{\text{hit}}} (1 - P(\text{unhit}|\mu_i))\ f_q(q_i|\mu_i)\ f_t(t_i|\mathbf{X})$$

Each fiTQun iteration requires the computation of the **predicted charge**

FITQUN TUNING



Direct:

$$\mu^{dir}(\mathbf{X}) = \Phi(p) \int ds g(p, s, \cos\theta)\Omega(R)T(R)\epsilon(\eta)$$

Cherenkov Profiles

Angular response

Indirect:

$$\mu^{sct}(\mathbf{X}) = \Phi(p) \int ds \frac{1}{4\pi}\rho(p, s_{\mathbf{X}})J(s_{\mathbf{X}})A(s_{\mathbf{X}})$$

Scattering Table

$$\mathcal{L}(\mathbf{X}) = \prod_j^{n_{\text{unhit}}} P(\text{unhit}|\mu_j) \prod_i^{n_{\text{hit}}} (1 - P(\text{unhit}|\mu_i))\ f_q(q_i|\mu_i)\ f_t(t_i|\mathbf{X})$$

Charge PDF

Time PDF

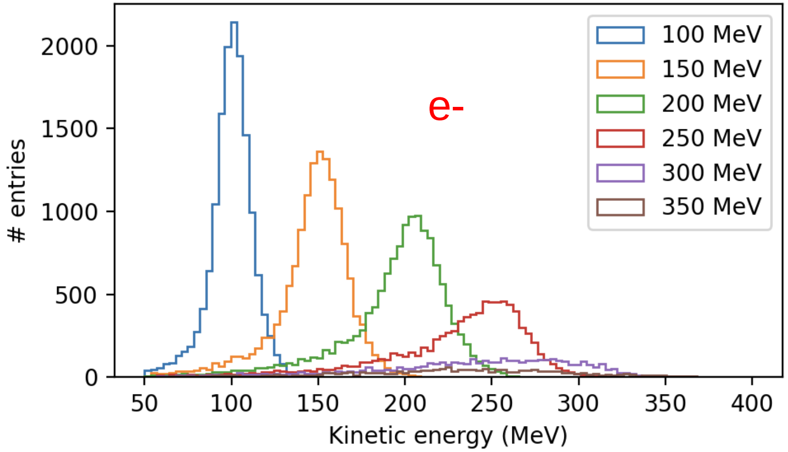
✅ Fully refactored

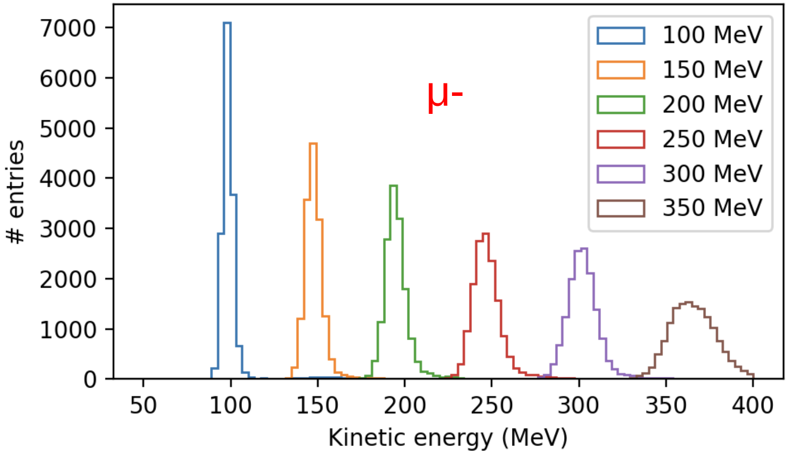
See HK-CM-Oct23

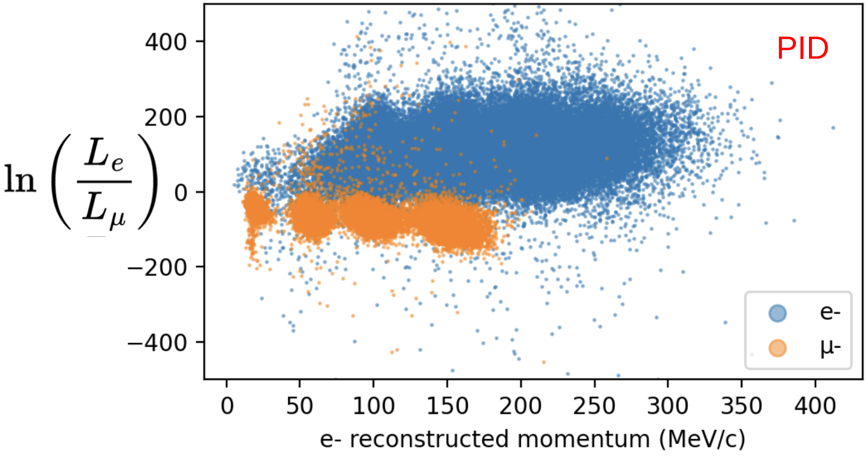
SOME WCTE RESULTS

e- and μ- at the center

Preliminary







Gonzalo D. L. (LPNHE)

II. PYFITQUN

MOTIVATION

PyfiTQun https://github.com/gondiaz/PyfiTQun/

- Simplify **fiTQun** readability using **Python** (early development)
- Objectives:
  - check subalgorithms in **fiTQun** reconstruction (prefiting, subevent search, full fit)
  - debugging and knowing all the requirements for a future performance driven refactoring

```python
# Event Loop
event_loop_iterator = manager.event_loop_iterator()
for next_event in event_loop_iterator:
    if next_event is None: continue
    event, hits = next_event

    # prefit
    logger.info("Performing prefit...")
    success, seed = likelihood_fit_seed(hits, prefit_sigma, cn, max_travel_time, radius, length, False)
    event_data["prefit"] = (success, seed[:4])
    if success: logger.info("Prefit successful")
    else      : logger.warning("Prefit not successful")

    # TODO: Find subevents

    # Likelihood fit
    for particle in manager.particles:

        logger.info(f"Performing likelihood fit for particle {particle}")
```

Implemented :

- Readers for WCSim and tuning files
- Event manager
- Prefit
- Parabolic approximation coefficients
- Predicted charges
- Likelihood function

TODOs:

- Finalize single ring fit
- Subeventing
- Multi rings

THE VERTEX PREFIT

- Event reconstruction in water Cherenkov detectors is based on the pair of values (charge, time)
- Baseline reconstruction algorithm is **fiTQun**: (position, direction, energy, particle type)
- First step, **vertex pre-fit**:
  - **Search for the interaction vertex** (position) using only time information (triangulation)
  - **Seeds multi-ring search**
  - **Seeds full fit**
- **WCTE** is very small ~3x3 m², **how good it is the vertex pre-fit**?

⚠ **No tuning needed except for the σ**

Prefit minimizer:

$$G(t, \mathbf{x}) \equiv \sum_{h}^{n_{hits}} e^{-\frac{1}{2}(T_h/\sigma)^2}$$

$$T_h \equiv t_h - t - \left\| \mathbf{R}_h^{\mathrm{PMT}} - \mathbf{x} \right\| / c_n$$

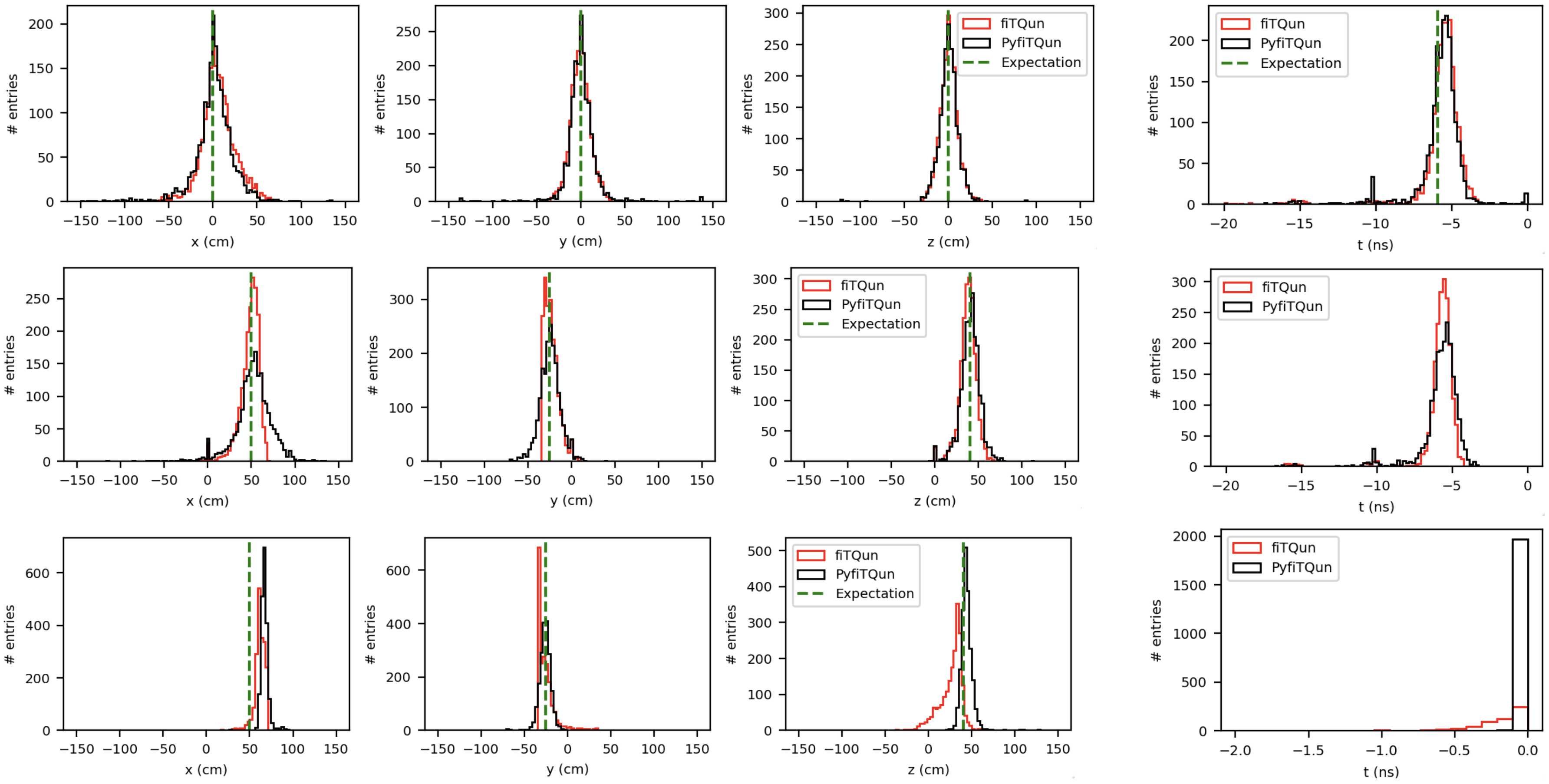
* fiTQun for SuperK: T2K Tech Note 146v2r0

Some WCTE results

e- samples with horizontal direction, prefit σ = 1 ns

Point-like (5 MeV)

Extended (300 MeV)



Gonzalo D. L. (LPNHE)

III. Comments and conclusions

COMMENTS AND CONCLUSIONS

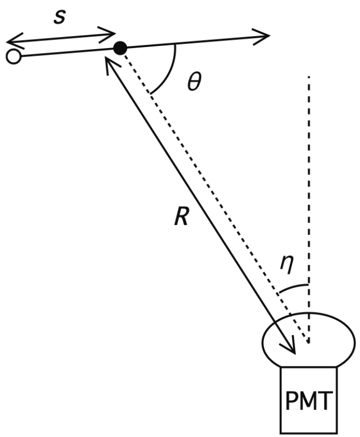
- **Full refactoring** of the tuning procedure
- **Validation** of the tuning is in progress due to differences in the cherenkov profiles
- Preliminary results on the full **fiTQun reconstruction** for WCTE
- **fiTQun prefit** performs well and is stable for WCTE
- Identified some **improvements/fixes in C++ fiTQun**:
  - Likely faster if parabolic coefficients j,k also computed in the tuning procedure
  - Is the parablic approximations actually needed? Couldn't we just tune the predicted charges?
  - Improve performace of Scattering Table computation to increase statistics
  - Indirect light not computed correctly as PMT positions are based on orientations (not valid for mPMTs)

- **PyfiTQun** allows to know the needed requirements of **fiTQun** algorithm
- **PyfiTQun** under development, needed to improve performance, **10 times slower**
- **PyfiTQun** makes easier to include new features such as j, k tuning and analytic derivatives to minimizers

- <u>Short term plan</u>: implement single ring reconstruction, fully validate new tuning tools (Cherenkov Profiles)
- <u>Long term plan</u>: work on performance (simplify tuning and paralelization)

BACKUP

UNEEDED APPROXIMATIONS



$$\mu^{dir}(p, r_0, \cos\theta_0) = \Phi(p)\int_0^{d_m(p)} ds g(p, s, \cos\theta) J(s) \approx \Phi(p)(I_0 j_0 + I_1 j_1 + I_2 j_2)$$

$$J(s) \equiv \Omega(R)T(R)\epsilon(\eta) \approx j_0 + j_1 s + j_2 s^2$$

$$\mu^{sct}(p, r_0, \cos\theta_0) = \Phi(p)\int_0^{d_m(p)} ds \int_\Omega ds \frac{1}{4\pi}\rho(p, s) J(s) A(s) \approx \Phi(p)(K_0 k_0 + K_1 k_1 + K_2 k_2)$$

$$J(s)A(s) \approx k_0 + k_1 s + k_2 s^2$$

Save computing time (they are computed in every iteration) and memory (avoid loading STable)

Cherenkov profiles

$$\mu^{dir}(r_0, \cos\theta_0) = \Phi(p) \int ds g(p, s, \cos\theta)\Omega(R)T(R)\epsilon(\eta)$$

$$J(s) \equiv \Omega(R)T(R)\epsilon(\eta) \approx j_0 + j_1 s + j_2 s^2$$

$$\mu^{dir}(r_0, \cos\theta_0) = \Phi(p) \int ds g(p, s, \cos\theta) J(s)$$
$$\approx \Phi(p)(I_0 j_0 + I_1 j_1 + I_2 j_2)$$

$$I_n(p, r_0, \cos\theta_0) = \int ds g(s) s^n \approx \sum_{i=0}^{n_{par}} J_i(r_0, \cos\theta_0) p^i$$

Scattering table

$$\mu^{sct}(\vec{\mathbf{x}}, \vec{\mathbf{p}}) = \Phi(p) \int ds \frac{1}{4\pi} \rho(p, s_{\vec{\mathbf{x}},\vec{\mathbf{p}}}) J(s_{\vec{\mathbf{x}},\vec{\mathbf{p}}}) A(s_{\vec{\mathbf{x}},\vec{\mathbf{p}}})$$

$$\rho(p, s) \equiv \int g(p, s, \cos\theta) d\Omega$$

$$A(z_s, R_s; z_{\mathrm{PMT}} | R_{\mathrm{PMT}}, \phi; z_d, \theta) = \frac{d\mu^{sct}}{d\mu^{dir,iso}}$$

Charge PDF

$$P(\mathrm{unhit}|\mu) \approx (1 + a_1\mu + a_2\mu^2 + a_3\mu^3)e^{-\mu}$$

$$f_q(q|\mu) \cong \sum_{i=0}^{n_{pars}} a_i(q)\mu^i$$

Time PDF

$$t_h^{res} = t_h - t - s_{mid}/c - |\mathbf{R}_h^{\mathrm{PMT}} - \mathbf{x} - s_{mid}\mathbf{d}|/c_n$$

$$f_t(t_h^{res}) = w f_t^{dir}(t_h^{res}) + (1 - w) f_t^{sct}(t_h^{res})$$

$$w = \frac{1 - e^{-\mu^{dir}}}{1 - e^{-\mu^{dir}} e^{-\mu^{sct}}}$$

Direct:

$$(\mu, \sigma)(p, \mu^{dir}) = \sum_{i=0}^{n_{gauss}} a_i(p)\ \mu^{dir}$$

$$a_i(p) = \sum_{j=0}^{n_{pars}} b_{ij}\ p^j$$