The Parametrized Extrapolator #2

Student: Krištof Špenko
Mentors: Andrii Usachov and Miriam Lucio Martinez, with prof. Pierre Billoir and the TrackFit Group

LHCb Tuesday Meeting, 7.9.2021

Magnetic Field map visualizer: gitlab
Extrapolator sandbox: gitlab

About me

- Live in Kamnik, Slovenia.
- Studying for my Master’s Degree in Mathematical Physics at the University in Ljubljana.
- Love to paint and play music.
- Previous work on RK and NN: github

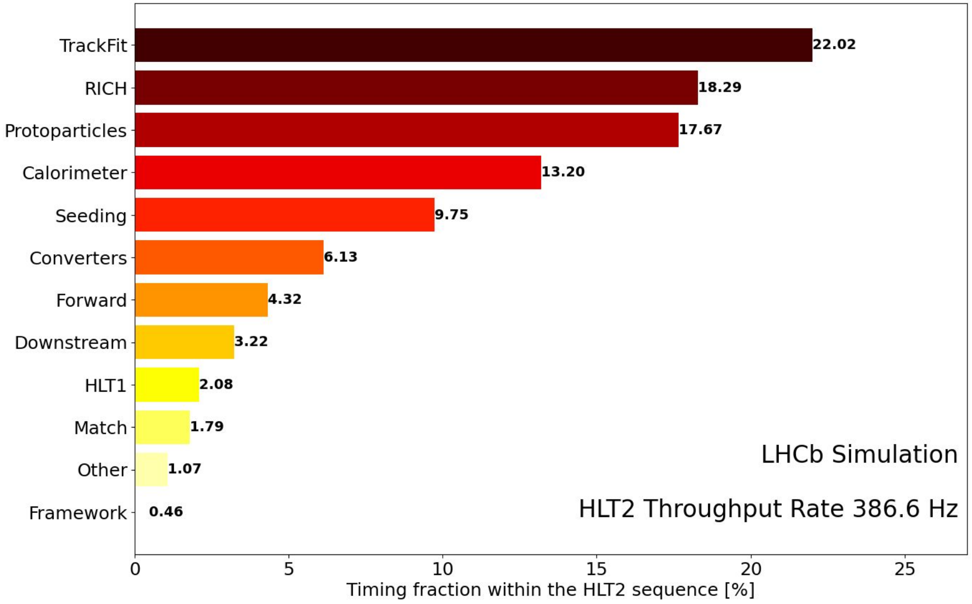
Jožef Stefan Institute

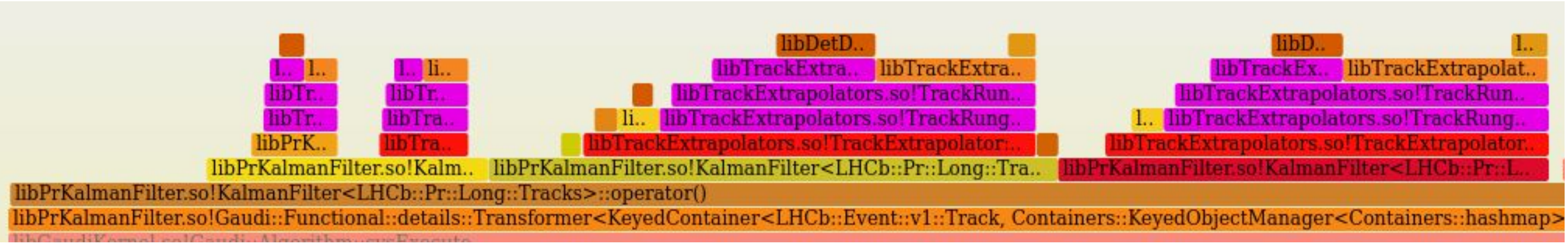
University of Ljubljana
Faculty of Mathematics and Physics

Run 3:

- Higher luminosity ➪ more events ➪ more tracks ➪ Longer CPU Time!!
- Speed-up ➠ Optimizing stack/Reconstruction software.
- TrackFit more than 20% of HLT2 CPU time!
- Baseline RK extrapolator more than 35% of TrackFit CPU time!! **BOTTLENECK**

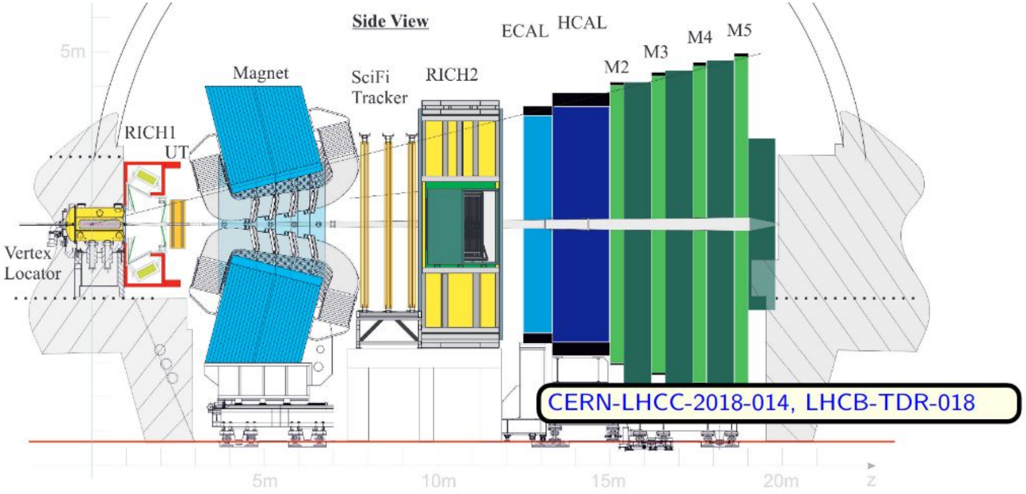
source:



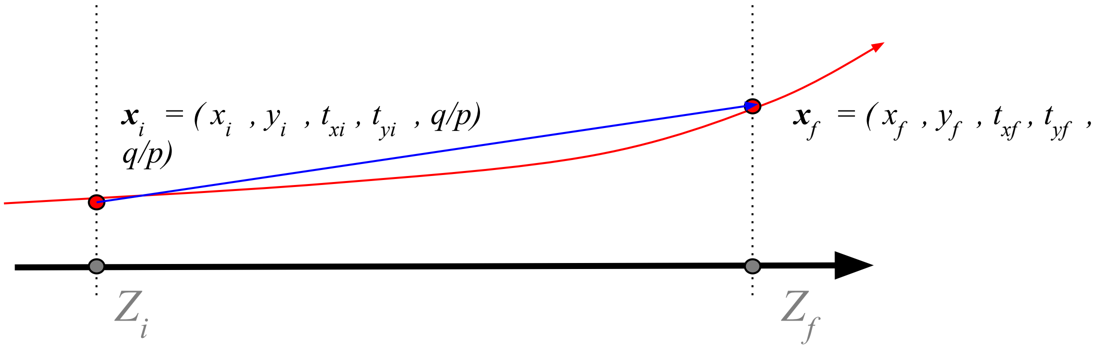


Track reconstruction:

- Propagating particles along tracks.
- Track is composed of state vectors.
- State vector is described by 5 components.
- Extrapolator perform state vector transitions between two consecutive z-values.



$$Z_i \rightarrow Z_f : \vec{x}(Z_i) \rightarrow \vec{x}(Z_f)$$

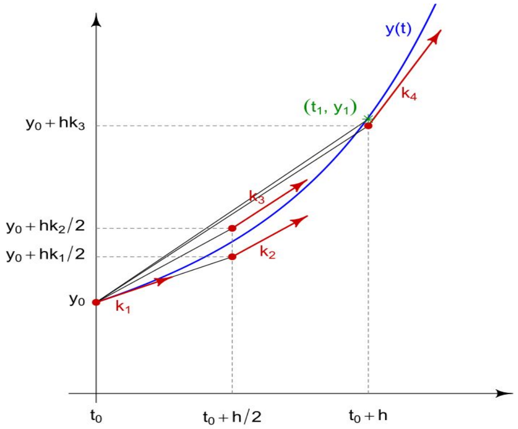


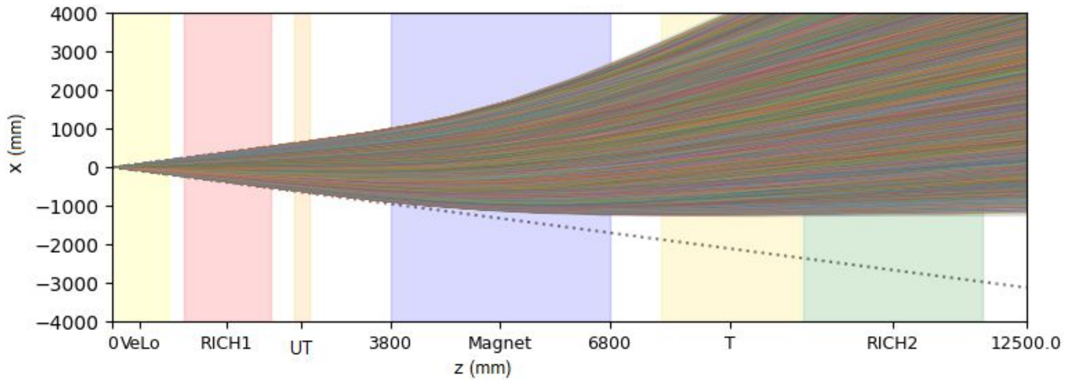
$$\vec{x} = \begin{pmatrix} x \\ y \\ t_x \\ t_y \\ q/p \end{pmatrix} t_x = \frac{\partial x}{\partial z}, t_y = \frac{\partial y}{\partial z}$$

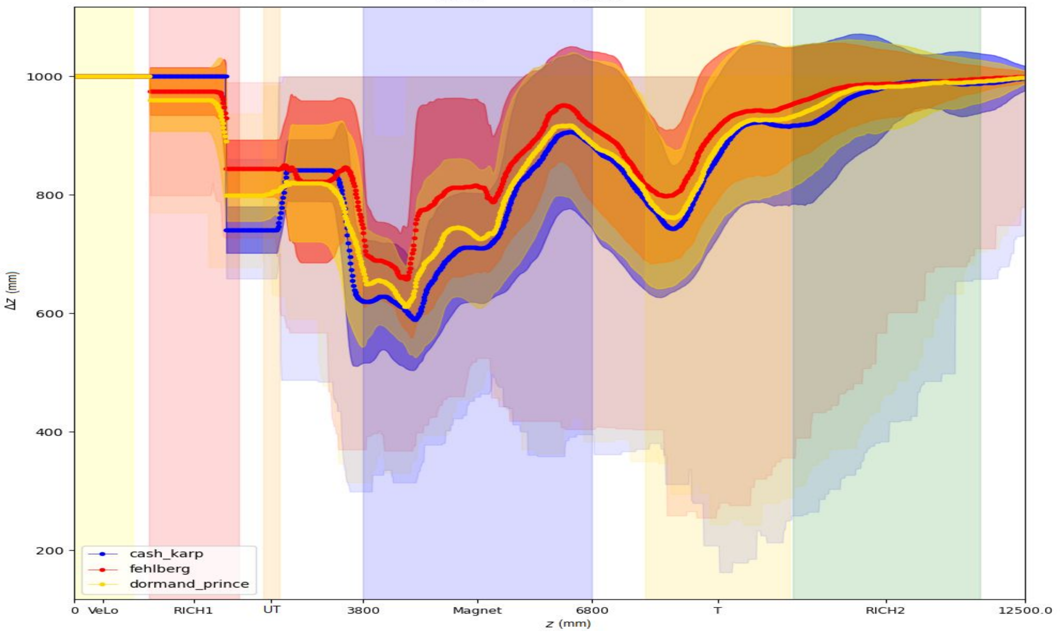
Adaptive Runge-Kutta

- Adaptive step with different schemes (default: “cash-karp”).
- Inability to process multiple tracks in parallel.
- Regions of interest VeLo-UT and UT-T, non-uniform field!

Python LHCb adaptive RK: gitlab



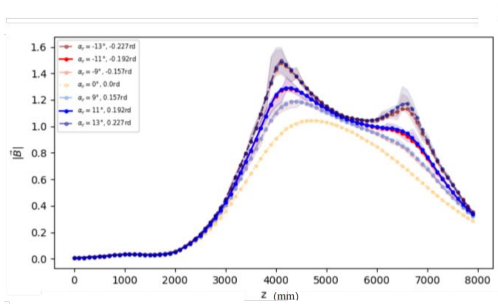


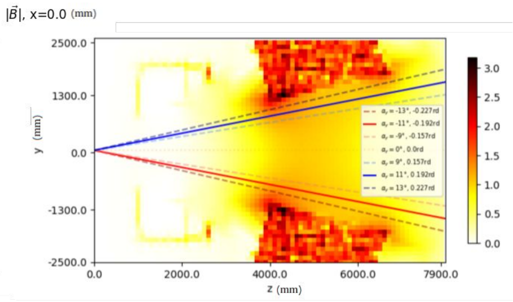


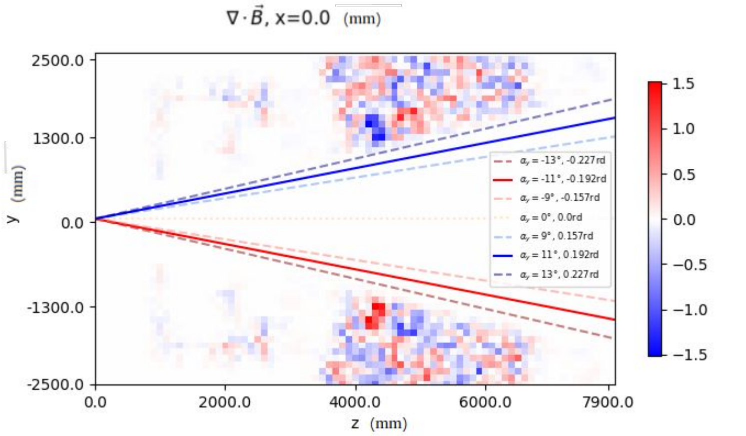
Magnetic Field Map

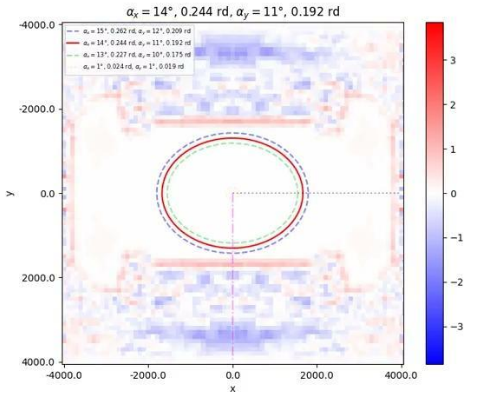
- Studying **B**, and ∇ × **B**, ∇ · **B**.
- Visualizing “smooth” map (by prof. Pierre Billoir).
- *Smooth* regions are promising for parametrizations.
- Defining central zone $|t_y|<0.2$, $|t_x|<0.25$. (coincides with the majority of tracks ~ 96%)

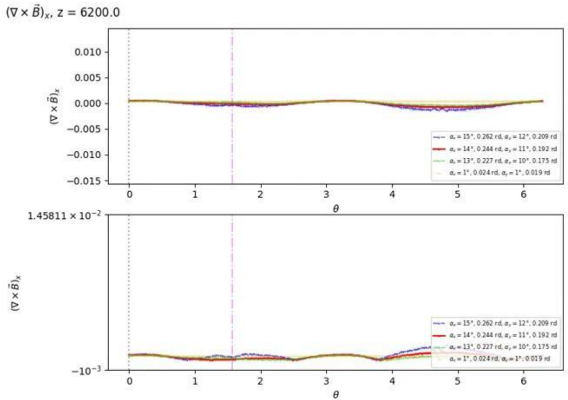
Field Map Visualizer: [gitlab]







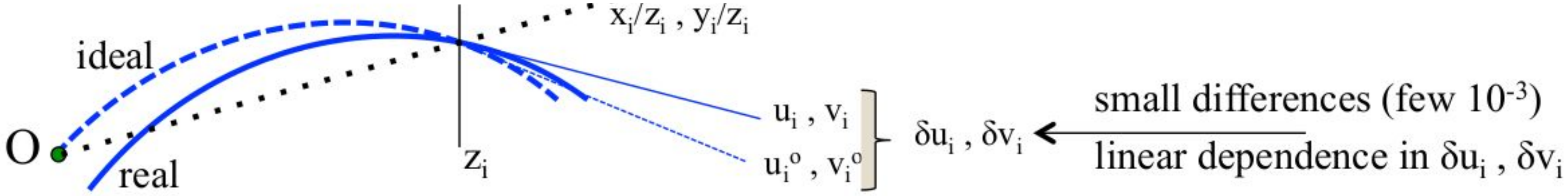




THE IDEA: *Parametrized Mapping* (by prof. Pierre Billoir)

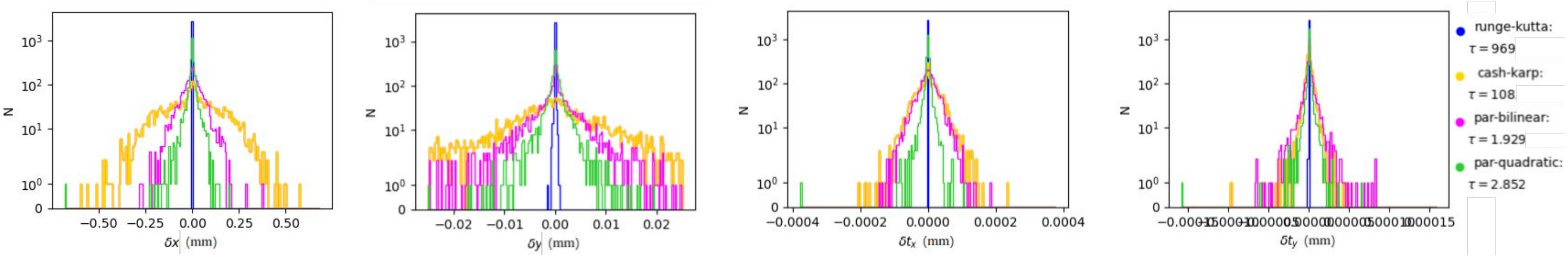
- Extrapolating long tracks coming from origin.
- Fit a grid of coefficients for a “semi-empirical” mapping of state vectors between detectors,
  - Fixed $Z_i \rightarrow Z_f$ extrapolation.
  - Parametrized *A, B, C* in terms of (x_i, y_i) each coefficient fitted on 5x30 tracks.
    (RK Field Map = 28.8MB, 50 thousand coefficients = 2.5MB)
  - Explicit dependence in 9th order polynomial in terms of *q/p* (fitted on range 2.5-100 GeV).
  - Estimate of initial bending factor.
- Similar concept used for Parametrized Kalman Filter. Now trying to use it in Full Kalman Filter.

$$\mathbf{f}(\mathbf{x}_i) = \sum_{k=1}^{K_1} \mathbf{A}_k(x_i, y_i) \left(\frac{q}{p}\right)^k + \sum_{k=1}^{K_2} \left(\mathbf{B}_k(x_i, y_i)\, \delta u + \mathbf{C}_k(x_i, y_i)\, \delta v\right) \left(\frac{q}{p}\right)^k$$



Testing the performance

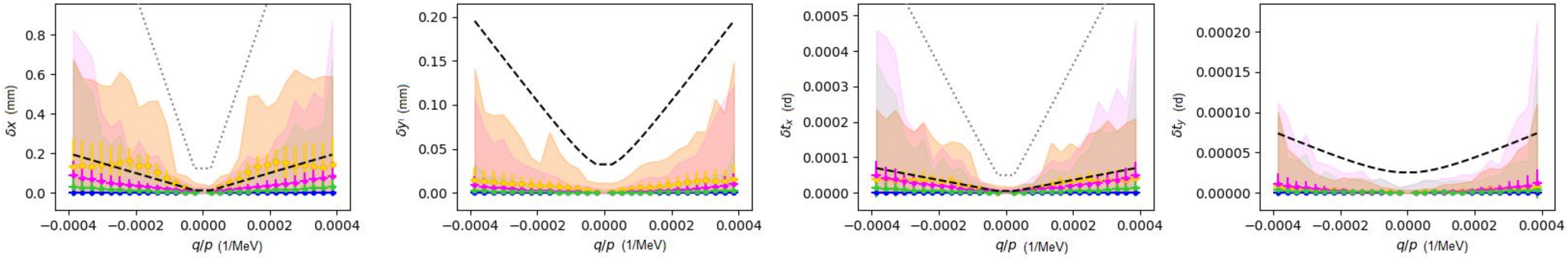
- Not every track dataset is representative!
- *Bouquet* tracks dataset ($z=0$: uniform t_x, t_y, q/p, Gaussian spread in x, y extrapolated to $z=Z_1$)



- Measure of sufficient accuracy → 0.1 × external error estimate (multiple scattering + measurement error).

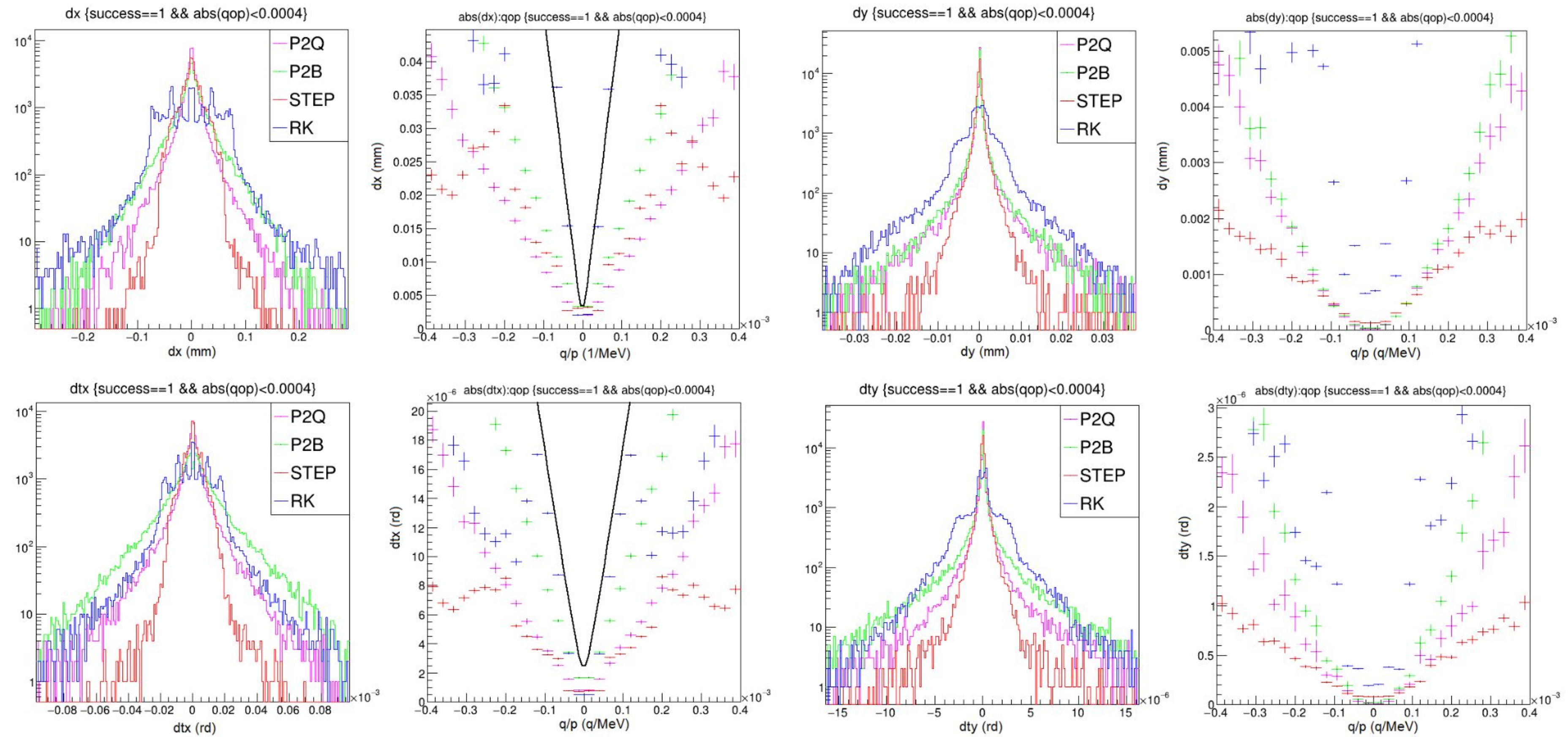
code: gitlab

$$\sigma_{external}^2 = \sigma_{m.s.}^2 + \sigma_{meas}^2$$



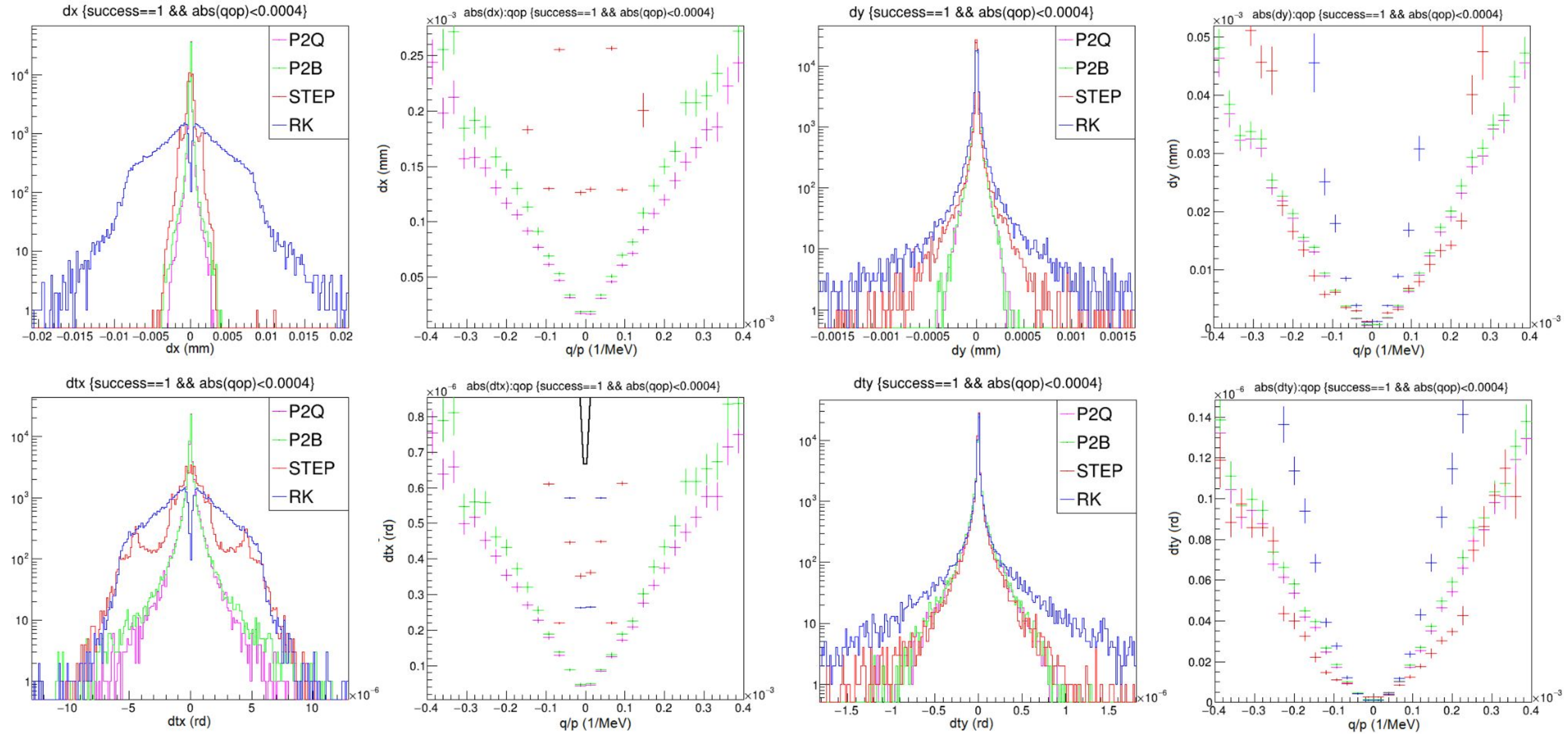
Implementation on the Stack software ✅

- UT-T extrapolations



Implementation on the Stack software ✅

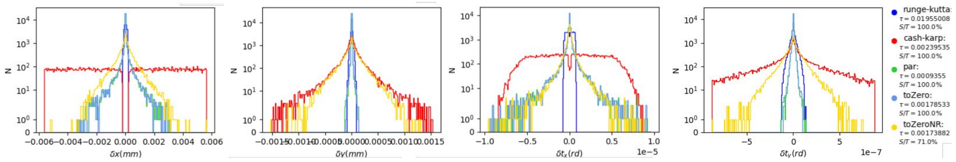
- VeLo-UT extrapolations



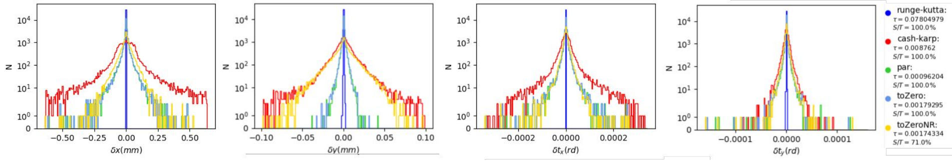
TODO = Different Modes

- Testing different “modes” of propagation:
  - Two step propagation $Z_1 \rightarrow 0 + 0 \rightarrow Z_2$ (avoid initial bending factor).

1. VeLo - UT

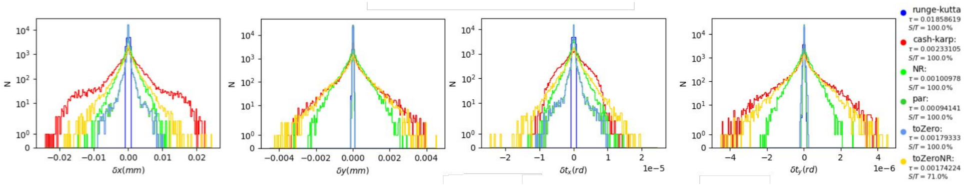


2. UT - T

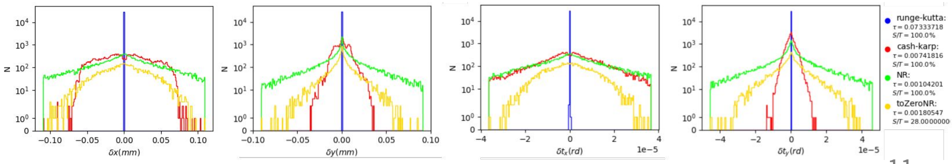


- BACKWARD propagation:
  a. Backward propagation,
  b. Two step propagation to origin,
  c. using “Newton-Raphson” algorithm (one iteration is sufficient):

3. UT-VeLo

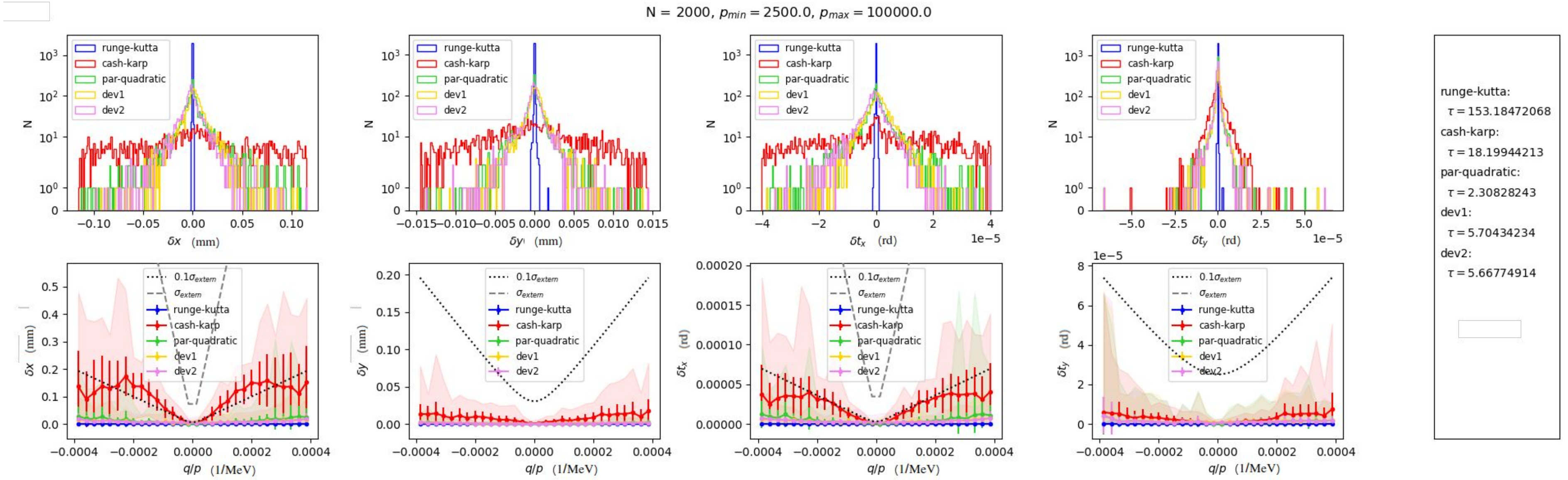


4. T - UT



TODO = Can NN do better?

- Incorporating a NN in the future:
  - Brute-force Machine learning probably not (slower). ❌
  - Capable of adding higher order corrections! ✅
  - *Best of both worlds?* Physics based NN architecture!? (replace fitting coefficients with “training”) ❓
    source



CONCLUSION:

- Magnetic Field Visualizer for studying different field maps. gitlab
- Extrapolator (Python) sandbox for studying and analysing extrapolators. gitlab
- LHCb implementation of parametrized extrapolator works well!
- Code needs to be vectorized/parallelization - potential GPU.
- Jacobian elements future testing.
- Neural network, new field brings interesting but complicated ideas!


SLOW
IT'S DANGEROUS TO THEORIZE ALONE! TAKE THIS.
Parametrized Extrapolator
THANK YOU 3000 !!!

BOUQUET Tracks for $Z_1 \rightarrow Z_2$:

1. Generate random momentum:
   a. $q/p \in$ (-1/2.5GeV, -1/100 GeV) U (1/100 GeV, 1/2.5GeV) UNIFORM,
2. At $z=0$ generate random state vectors:
   a. x GAUSSIAN spread around 0 with $\sigma = 0.03$mm,
   b. y GAUSSIAN spread around 0 with $\sigma = 0.03$mm,
   c. $t_x \in$ (-0.25, 0.25) radians UNIFORM,
   d. $t_y \in$ (-0.20, 0.20) radians UNIFORM,
3. $z=0$ state is propagated to Z_1 with reference extrapolator.
4. Check if state is in central region at Z_1:
   a. $|x| <= Z_1 \times 0.25$,
   b. $|y| <= Z_1 \times 0.20$,
   c. $|t_x| <= 0.25$,
   d. $|t_y| <= 0.20$,
5. If state vector is in central region at Z_1 ACCEPT as initial state vector in dataset. Otherwise go back to STEP 2!

ACCURACY CRITERION: $z_1 = 2500$mm $\longrightarrow$ $z_2 = 7500$mm

the extrapolation error should be « small » compared to the ***external*** error (contribution of multiple scattering along extrapolation and measurement errors in the downstream part)

conservative estimation of external error

- multiple scattering in air (~ 5 m): 0.017 X_{rad}

  $\sigma(t_x) = \sigma(t_y) \approx 1.8\ 10^{-3}$ / p(GeV) $\sigma(x) = \sigma(y) \approx 5$ mm / p(GeV/c)

- measurement error (neglecting mult. scatt. in T1,T2,T3)

  $\sigma(x) \approx 0.03$ mm $\sigma(y) \approx 0.3$ mm $\sigma(t_x) \approx 0.025\ 10^{-3}$ $\sigma(t_y) \approx 0.25\ 10^{-3}$

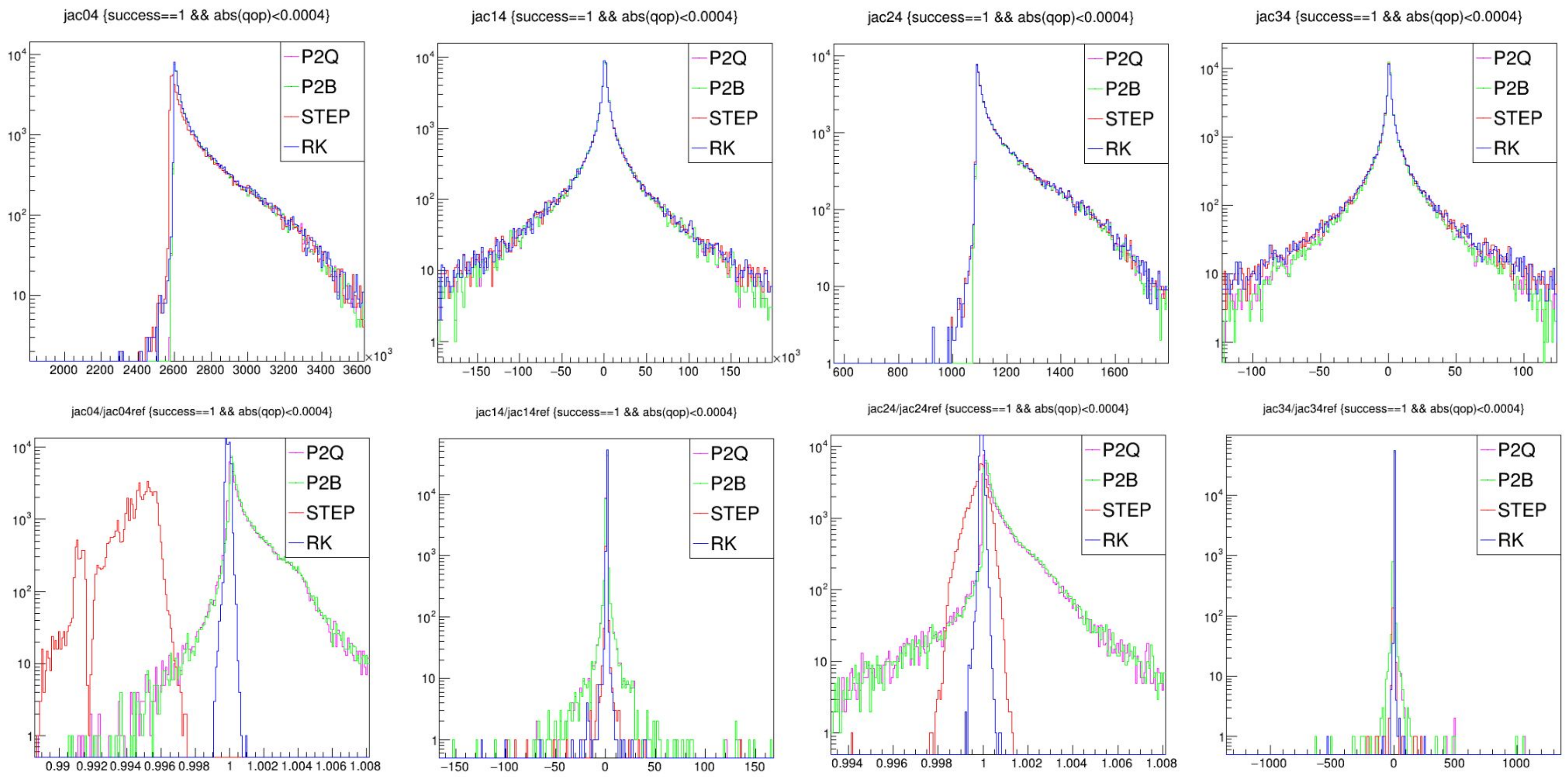
then: $\sigma_{external}^2 = \sigma_{m.s.}^2 + \sigma_{meas}^2$

error dominated by multiple scattering (except for very high momentum)

if extrapolation and external errors are independent:

$\sigma_{extrap} = 0.1\ \sigma_{external}$ ➔ + 0.5 % on combination : acceptable ?

JACOBIAN Elements



$$\boldsymbol{x}_{k|k-1} = \boldsymbol{f}_k(\boldsymbol{x}_{k-1|k-1}),$$

$$\boldsymbol{P}_{k|k-1} = \boldsymbol{F}_k \boldsymbol{P}_{k-1|k-1} \boldsymbol{F}_k^T + \boldsymbol{Q}_k,$$

$$F = J = \begin{pmatrix} \frac{\partial f_x}{\partial x} & \frac{\partial f_x}{\partial y} & \frac{\partial f_x}{\partial t_x} & \frac{\partial f_x}{\partial t_y} & \frac{\partial f_x}{\partial \frac{q}{p}} \\ \frac{\partial f_y}{\partial x} & \frac{\partial f_y}{\partial y} & \frac{\partial f_y}{\partial t_x} & \frac{\partial f_y}{\partial t_y} & \frac{\partial f_y}{\partial \frac{q}{p}} \\ \frac{\partial f_{t_x}}{\partial x} & \frac{\partial f_{t_x}}{\partial y} & \frac{\partial f_{t_x}}{\partial t_x} & \frac{\partial f_{t_x}}{\partial t_y} & \frac{\partial f_{t_x}}{\partial \frac{q}{p}} \\ \frac{\partial f_{t_y}}{\partial x} & \frac{\partial f_{t_y}}{\partial y} & \frac{\partial f_{t_y}}{\partial t_x} & \frac{\partial f_{t_y}}{\partial t_y} & \frac{\partial f_{t_y}}{\partial \frac{q}{p}} \\ \frac{\partial f_{q/p}}{\partial x} & \frac{\partial f_{q/p}}{\partial y} & \frac{\partial f_{q/p}}{\partial t_x} & \frac{\partial f_{q/p}}{\partial t_y} & \frac{\partial f_{q/p}}{\partial \frac{q}{p}} \end{pmatrix}$$