Kinematic Fitting

Julie Munch Torndal
University of Copenhagen
September 2021

This note is a reprint from [1].

Measurements are biased. They reflect the physical world - "das Ding an sich" [2] - but they are biased from the experimental lens that we perceive the world through. In a statistical sense measurements are random variables drawn from an underlying distribution. A single measurement on its own has little to no meaning, but combining it with other measurements and statistical tools, measurements are key to understanding the physical world. They are never fully exact, but they can be used to make estimates of the true world. Estimates themselves from their very nature are also random and therefore include imprecision. They can and will differ from the *true value* because of statistical fluctuations. The aim is to find the "best" value for the estimate that most accurately describes the *true value* while being acutely aware of how reliable it is i.e. the extend of the imprecision. The method of least squares is a simple and powerful tool for the analysis of experimental data and it can be used as a method of estimation. Generally, a model can be expressed in the form of conditions or constraints

$$f(a_1, a_2, \ldots, a_p, y_1, y_2, \ldots, y_n) = 0, \tag{1}$$

where y_i is a set of measurements consisting of n measured values, and a_i is a set of parameters consisting of p values, that the model depends on. Measurements will lie within some confidence region of the true underlying distribution, and they are unlikely to fulfil the conditions a priori, but corrections Δy_i can be added such that $y_i + \Delta y_i$ exactly fulfil the conditions. The principle of least squares requires that the sum of the corrections squared form a minimum:

$$S = \sum_{i=1}^{n} \Delta y_i^2 = \text{min}. \tag{2}$$

As a simple example, the mean of a set of measured values is found from

$$\hat{y} = \frac{1}{n} \sum_{i=1}^{n} y_i \tag{3}$$

which satisfies the principle of least squares. The principle has many applications, where the most common one would be estimating the parameters for a model based on a set of measurements. Another common use is found in testing the hypothesis that the measured data are consistent with the assumed model parametrisation.

With kinematic fitting, the principle of least squares is applied in order to improve the accuracy of the event reconstruction by determining the corrections of the measurements. As an example an important set of constraints will be the conservation law of energy and momentum which defines four conditions. The energy and momentum of the final state must equal the initial state $(\sqrt{s}, 0, 0, 0)$ in order to be conserved. Kinematic fitting works particularly well at an e^+e^- collider because the initial state (E, p_x, p_y, p_z) is already known in all directions unlike at hadron colliders, where the initial state is only known in the plane transverse to the beam axis (z-direction).

Constrained Fit

The derivations in this section are based on [3]. In a constrained fit, the underlying model for a system is expressed in the form of m constraints

$$f_k(\bar{a}, \bar{y}) = 0 \quad , \quad k = 1, \dots, m \tag{4}$$

with n measured parameters for $\bar{y}_i$, $i = 1, \dots, n$ and p unmeasured parameters for a_j, $j = 1, \dots, p$. The model is exactly fulfilled by a set of measurements whose expectation values are the true values $\bar{y}_i$ without bias

$$E(y_i) = \bar{y}_i. \tag{5}$$

For an uncorrelated set of measurements, the covariance matrix is a diagonal matrix

$$\mathbf{V} = \delta_{ij}\sigma_i\sigma_j \tag{6}$$

where σ_i is the resolution of the i-th measurement. In the case of no free parameters, the sum of squares can be rewritten as

$$S(y) = \Delta y^T \mathbf{V}(y)^{-1} \Delta y, \tag{7}$$

where Δy is a vector of the corrections to the measurements such that the sum $S(y)$ obtains a minimum and $\mathbf{V}(y)$ is the covariance matrix. For measurements that follow a Gaussian distribution, this form corresponds to the general form for calculating the χ^2. Lagrangian multipliers can be used to introduce constraints to the function we wish to minimise. The sum of squares can be extended by

$$L(y) = S(y) + 2\sum_{k=1}^{m} \lambda_k f_k(a, y). \tag{8}$$

Under the minimisation, the multipliers must vanish such that $f_k(a, y) = 0$. Only solutions that exactly fulfil the constraints are searched for, hence minimising $L(y)$ is equivalent to minimising $S(y)$ while fulfilling the constraints. The solution simultaneous fulfils

$$\frac{\partial L}{\partial y} = 0 \quad , \quad \frac{\partial L}{\partial a} = 0 \quad , \quad \frac{\partial L}{\partial \lambda} = 0. \tag{9}$$

Generally, least squares problems are solved by iteration. In the special case where the problem is linear, it can be solved analytically. Nonlinear problems can be reduced to a set of linear problems by linearisation techniques and solved numerically. Numerical methods require some attentiveness in order for the solution to (hopefully) converge and initial starting values should be given some thought. The initial values for the measured parameters should be the measurements themselves, y_i. The initial values for the unmeasured parameters depend on the problem.

In the case of semi-leptonic top pair produced events there is only one neutrino that has free parameters. Naively, one would expect that its four-momentum can be reconstructed from the missing four-momentum, but that assumes that the energy-momentum conservation can be satisfied a priori. While neutrinos do carry away some of the missing four-momentum, it also strongly depends on measurement effects for the reconstruction of the remaining event and missing coverage in the detector where some particles can escape detection. Nonetheless, the missing four-momentum is a reasonable initial value for the neutrino as it solves the energy and momentum conditions. In fully-leptonic events there are two undetected neutrinos and hence unmeasured parameters for them both. The directions of the two neutrinos contribute both to the missing four-momentum. Reasonable initial values could be found by solving selected conditions such as invariant mass of particle systems. In fully-hadronic events there are no unmeasured parameters which simplifies the problem of initial values, instead the challenge is figuring out the combinatorics of particle systems from the 6-jet final state.

Non-linear conditions can be linearised by Taylor-expansion

$$f_k(a^n, y^n) + \sum_j \frac{\partial f_k}{\partial a_j^{n+1}} (\Delta a_j^{n+1} - \Delta a_j^n) + \sum_i \frac{\partial f_k}{\partial y_i^{n+1}} (\Delta y_i^{n+1} - \Delta y_i^n) \approx 0 \tag{10}$$

and the function L for the $(n+1)$-th iteration can be rewritten as

$$L = \Delta y^T \mathbf{V}(y)^{-1} \Delta y + 2\lambda^T (A\Delta a + B\Delta y - c) \tag{11}$$

with

$$c = A\Delta a^n + B\Delta y^n - f \tag{12}$$

and

$$A = \begin{pmatrix} \partial f_1/\partial a_1 & \partial f_1/\partial a_2 & \dots & \partial f_1/\partial a_p \\ \partial f_2/\partial a_1 & \partial f_2/\partial a_2 & \dots & \partial f_2/\partial a_p \\ \dots & & & \\ \partial f_m/\partial a_1 & \partial f_m/\partial a_2 & \dots & \partial f_m/\partial a_p \end{pmatrix}, \quad f = \begin{pmatrix} f_1(a^n, y^n) \\ f_2(a^n, y^n) \\ \dots \\ f_m(a^n, y^n) \end{pmatrix},$$
$$B = \begin{pmatrix} \partial f_1/\partial y_1 & \partial f_1/\partial y_2 & \dots & \partial f_1/\partial y_n \\ \partial f_2/\partial y_1 & \partial f_2/\partial y_2 & \dots & \partial f_2/\partial y_n \\ \dots & & & \\ \partial f_m/\partial y_1 & \partial f_m/\partial y_2 & \dots & \partial f_m/\partial y_n \end{pmatrix}. \tag{13}$$

The system of equations to be solved can be written in the form of one matrix equation that is solved by

$$\begin{pmatrix} V^{-1} & 0 & B^T \\ 0 & 0 & A^T \\ B & A & 0 \end{pmatrix} \begin{pmatrix} \Delta y \\ \Delta a \\ \lambda \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ c \end{pmatrix}$$
$$\Rightarrow \begin{pmatrix} \Delta y \\ \Delta a \\ \lambda \end{pmatrix} = \begin{pmatrix} C_{11} & C_{21}^T & C_{31}^T \\ C_{21} & C_{22} & C_{32}^T \\ C_{31} & C_{32} & C_{33} \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ c \end{pmatrix} = \begin{pmatrix} C_{31}^T \\ C_{32}^T \\ C_{33} \end{pmatrix} c \tag{14}$$

where the elements C_{11} through C_{33} are

$$\begin{aligned}
C_{11} &= V - VB^TW_BBV + VB^TW_BAW_A^{-1}A^TW_BBV \\
C_{21} &= -W_A^{-1}A^TW_BBV \\
C_{22} &= W_A^{-1} \\
C_{31} &= W_BBV - W_BAW_A^{-1}A^TW_BBV \\
C_{32} &= W_BAW_A^{-1} \\
C_{33} &= -W_B + W_BAW_A^{-1}A^TW_B
\end{aligned} \tag{15}$$

with $W_B = (BVB^T)^{-1}$ and $W_A^{-1} = (A^TW_BA)^{-1}$. Since the covariance matrix $\mathbf{V}$ for the measurements is symmetric, the elements C_{11}, C_{22}, C_{31}, and C_{33} are also symmetric. The symmetric covariance matrix for the solution is of the form

$$V\begin{pmatrix} \hat{y} \\ \hat{a} \\ \hat{\lambda} \end{pmatrix} = \begin{pmatrix} C_{11} & C_{21}^T & 0 \\ C_{21} & C_{22} & 0 \\ 0 & 0 & -C_{33} \end{pmatrix}. \tag{16}$$

The above covariance matrix shows that the estimates for both measured and unmeasured particles are independent of the estimates of the Lagrange multipliers. They are an aid in treating the minimisation problem of least squares without an explicit parametrisation in terms of constraints, but they do not enter the solution.

Inclusion of Probability Distribution Functions

Above, the energy and momentum conservation were given as examples of constraints on the form of Eq. 4. Other examples of constraints are the mass constraints in the jet systems. On the hadronic side of a semileptonic top pair produced event, the top decays into a W-boson and a b-quark and the W decays further into a quark pair. Therefore there is a system of 3 jets whose mass should equal the top mass and a system of 2 light-flavour jets whose system should equal the W mass. On the leptonic side of the event, there is a jet, a lepton, and a neutrino where the system of all three particles should also equal the top mass and the system of the lepton and neutrino should equal the W mass. The masses of the jet system are not expected to be exact since the particles are unstable. Unstable particles have a total decay rate $\Gamma = 1/\tau$ hence their masses have a Breit-Wigner resonance where the Full Width at Half Maximum (FWHM) corresponds to Γ. Since the masses are not expected to be exactly on resonance, a constraint can be loosened by introducing a Probability Distribution Function (PDF) which further extends the function L by

$$L(y) = S(y) + g(x) + 2\sum_{k=1}^{m} \lambda_k f_k(a, y, x). \tag{17}$$

The additional function $g(x)$ only depends on a scalar variable, and it represents a penalty function. Since the parameters of the distributions are not free in the fit, the PDFs can be handled by extending the B-matrix such that the total set of measured parameters is $\{\bar{y}_i, x\}$

$$B = \frac{\partial f(a, y, x)}{\partial(y, x)} \tag{18}$$

and the extended covariance matrix is

$$\tilde{\mathbf{V}} = \begin{pmatrix} \mathbf{V} & 0 \\ 0 & \left(\frac{1}{2}\frac{\mathrm{d}^2 g}{\mathrm{d}x^2}\big|_{x=x^n}\right)^{-1} \end{pmatrix}. \tag{19}$$

Note that the penalty function has been Taylor expanded to second order. The corrected measured parameters in the $(n+1)$-th iteration are calculated from

$$\begin{aligned} \begin{pmatrix} y^{n+1} \\ x^{n+1} \end{pmatrix} = \begin{pmatrix} y_0 \\ x^n \end{pmatrix} - \tilde{\mathbf{V}} \begin{pmatrix} 0 \\ \frac{1}{2}\frac{\mathrm{d}^2 g}{\mathrm{d}x^2}\big|_{x=x^n} \end{pmatrix} + \tilde{\mathbf{V}} B^T (B\tilde{\mathbf{V}}B^T)^{-1} \\ \times \left[A(a^n - a_0) + B \begin{pmatrix} y^n - y_0 \\ \frac{\mathrm{d}g}{\mathrm{d}x}\big|_{x=x^n} / \frac{\mathrm{d}^2 g}{\mathrm{d}x^2}\big|_{x=x^n} \end{pmatrix} - f(a^n, y^n, x^n) \right] \end{aligned} \tag{20}$$

and the corrected free parameters in the $(n+1)$-th iteration are calculated from

$$\begin{aligned} a^{n+1} &= a_0 + W_A^{-1} A^T W_B \\ &\times \left[A(a^n - a_0) + B \begin{pmatrix} y^n - y_0 \\ \frac{\mathrm{d}g}{\mathrm{d}x}\big|^{-1}_{x=x^n} / \frac{\mathrm{d}^2 g}{\mathrm{d}x^2}\big|_{x=x^n} \end{pmatrix} - f(a^n, y^n, x^n). \right] \end{aligned} \tag{21}$$

ABC-parametrisation

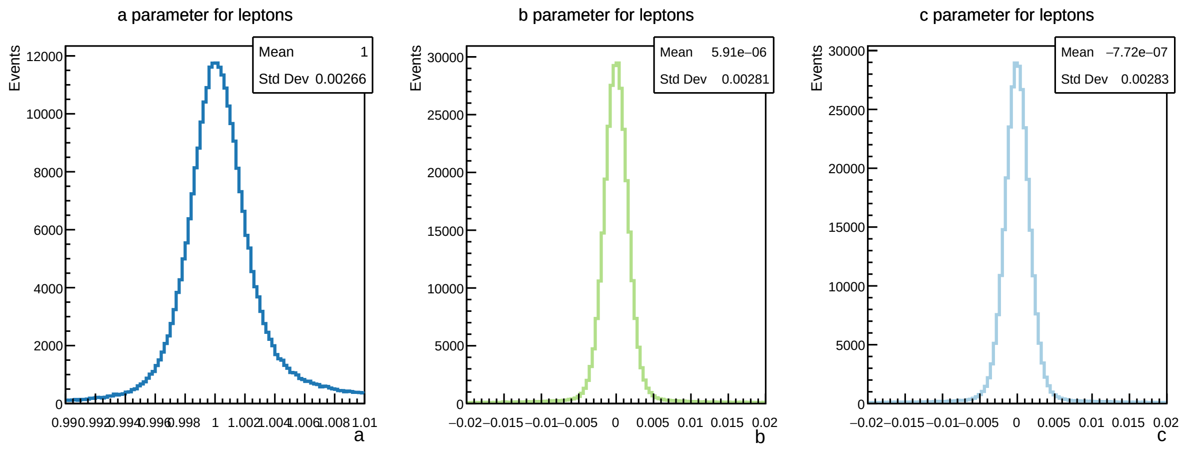
In the constrained fit, the parameters of the underlying quarks and leptons should be Gaussian distributed. Otherwise the interpretation of the χ^2 would have little meaning. Additionally, Gaussian distributed parameters are vital for the convergence of the fit [4]. Using the measured energies and angles directly does not ensure Gaussian distributed parameters. The 3-vector for a reconstructed particle can be written as

$$\vec{p}_j^{\,r} = a_j |\vec{p}_j^{\,m}| \vec{p}_j^{\,a} + b_j \vec{p}_j^{\,b} + c_j \vec{p}_j^{\,c} \tag{22}$$

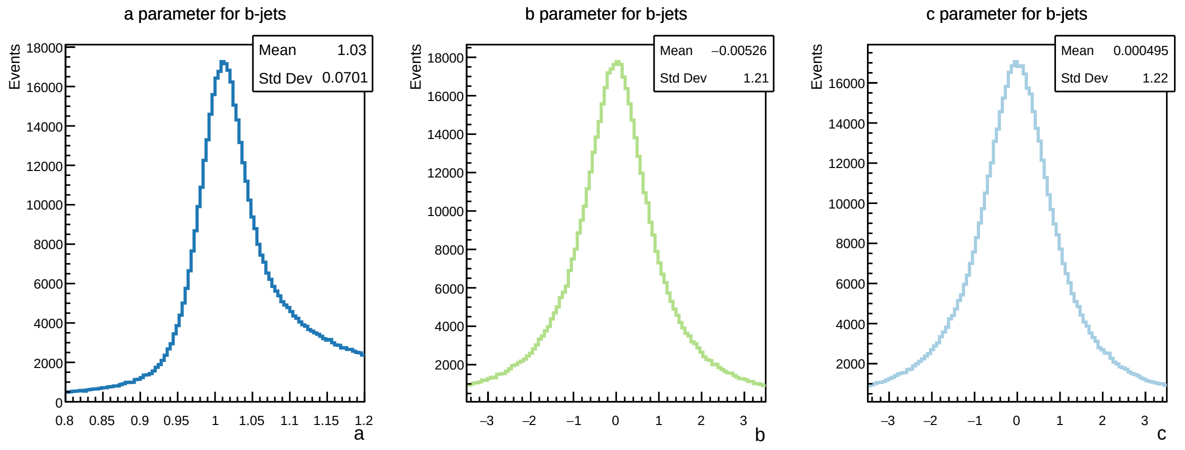
where a_j, b_j and c_j are the parameters used in the fit and $\vec{p}_j^{\,a}$, $\vec{p}_j^{\,b}$ and $\vec{p}_j^{\,c}$ are the unit vectors which form a Cartesian system. They are determined from the measured jet momentum $\vec{p}_j^{\,m}$,

$$\begin{aligned} \vec{p}_j^{\,a} &= \frac{\vec{p}_j^{\,m}}{|\vec{p}_j^{\,m}|}, \\ \vec{p}_j^{\,b} &= \frac{1}{\sqrt{p_{x,m}^2 + p_{y,m}^2}} (p_y^m, -p_x^m, 0), \\ \vec{p}_j^{\,c} &= \frac{1}{\sqrt{|\vec{p}_j^{\,m}|^2 (p_{x,m}^2 + p_{y,m}^2)}} (-p_x^m p_z^m, -p_y^m p_z^m, p_{x,m}^2 + p_{y,m}^2). \end{aligned} \tag{23}$$

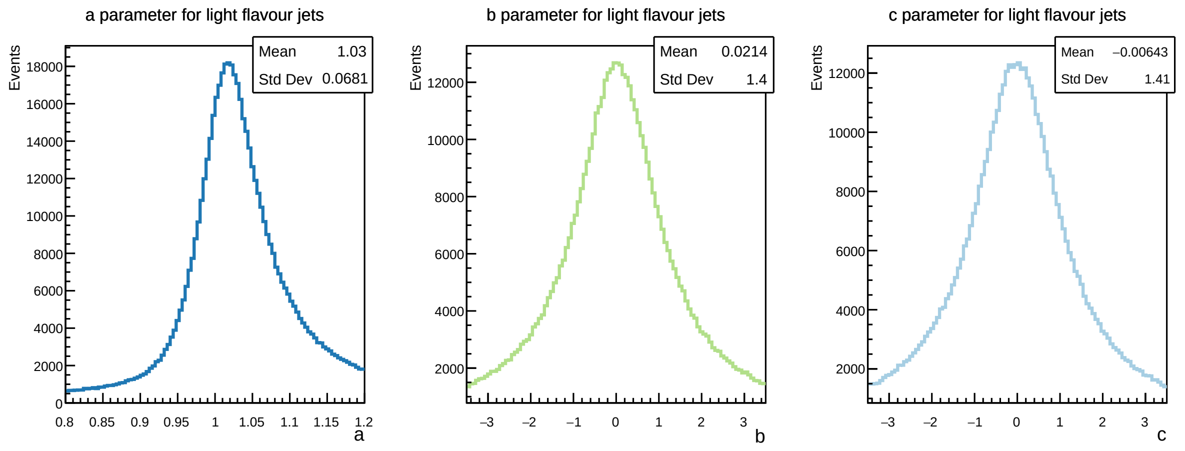
where $\vec{p}_j^{\,a}$ is defined to follow the direction of the measured jet, $\vec{p}_j^{\,b}$ is defined so that the dot product between itself and $\vec{p}_j^{\,a}$ equals zero, and $\vec{p}_j^{\,c}$ is defined as the cross product of $\vec{p}_j^{\,a}$ and $\vec{p}_j^{\,b}$. The initial values of the parameters are set to $\{a_j, b_j, c_j\} = \{1, 0, 0\}$ so that the reconstructed particle vector initially overlaps with the measured particle vector.



(a) a, b, and c parameter distribution for leptons



(b) a, b, and c parameter distribution for b-jets



(c) a, b, and c parameter distribution for light flavour jets

Figure 1: a, b, and c parameter distributions for three types of reconstucted objects; leptons, b-jets, and light flavour jets.

ABCfit++ software package

The ABCfit++ software package was written in connection with this analysis. It is based on ABCfit developed by Oliver Buchmuller and Jørgen Beck Hansen. It contains a set of classes described in the following:

Coordinate Representation The first base class contains the coordinate representation for the particles where `CoordRepr.h` defines the base class. There are four derived classes defined in `PxPyPzM.h`, `PxPyPzE.h`, `PtEtaPhiM.h`, and `ABCD.h`. Each derived class defines how the coordinates transform between itself (this) and the internal representation, the derivative of itself w.r.t the internal representation, the derivative of the internal representation w.r.t. itself, and default expectation parameter values and expectation covariance matrix. The internal representation is PxPyPzM which defines the coordinates in p_x, p_y, p_z, and mass. PxPyPzE defines the coordinates in p_x, p_y, p_z, and energy. As an example, the transformations between PxPyPzM and PxPyPzE are straightforward since only the fourth parameter is transformed by

$$E = \sqrt{m^2 + p_x^2 + p_y^2 + p_z^2} \quad \longleftrightarrow \quad m = \sqrt{E^2 - p_x^2 - p_y^2 - p_z^2} \tag{24}$$

PtEtaPhiM defines the coordinates in terms of transverse momentum p_T, pseudorapidity η, and the polar angle in the transverse plane ϕ. The derivatives are 4×4 matrices and convenient for applying the chain rule when calculating the A and B matrices in Eq. 13. The final coordinates representation is the ABCD representation which defines the A, B, and C coordinates as described in the previous section. The fourth parameter denoted D is set to the mass of the particle. The ABCD representation is currently the only suitable choice in the software package for representing the parameters in Eq. 17 since the parameters will be Gaussian distributed.

In the other three representations, the default expectation values are simply set to coordinates of the particle itself. In ABCD, the default expectation values are $\{1, 0, 0, m\}$ as discussed above. Since the parameters are defined in terms of the measured jet, the unit vectors must be saved. In addition to the four parameters for ABCD, the representation also includes the auxiliary coordinates for the unit vectors bringing the total number of coordinates to twelve.

The base class representation contains an option to overwrite the default expectation values and covariance matrices for individual particles. This is useful since different particles have different expectations as can be seen in Figure 1.

Particle Object The base class for particle objects is defined in `ParticleObject.h`. A particle object contains a set of coordinates and an input coordinate representation in which the coordinates for a particle are defined. Only measured particles contribute to the χ^2, so their objects contain an additional χ^2 coordinate representation for how the parameters of the χ^2 are defined. By default, all parameters of a particle object are subject to change in a fit. The particle object class contains the option to fix certain parameters in the constrained fit.

Constraint Setting up constraints is defined by the base class `constraint.h`. There are five derived classes defined in `SumP{x,y,z}Constraint.h`[1], `SumEConstraint.h`, and `InvMassConstraint.h`. The first four set up the 4-momentum constraints and the last one sets up the constraint for the invariant mass of a particle system. Each constraint takes a list of particles which is a vector of pointers to particle objects. The list defines the particle system. There are two types of constraints.

For the first type, the constraint has to be exactly fulfilled and a constraint value is given as input. The class contains a constraint function which in the case of the SumEConstraint calculates the sum of energies. The constraint is fulfilled when the sum exactly matches the constraint value.

For the other type, the constraint is allowed to vary, and instead of a constraint value a probability distribution function is given as input. These types of constraints contributes to the χ^2 via the penalty function $g(x)$ as seen in equation 17. The more the constraint function value deviates from the reference value, the higher the penalty.

Composite constraint The base class of composite constraints is used to create linear combinations of constraints. The class is defined in `CompositeConstraint.h`. It takes a list of constraints as input where the list is a vector of pairs. Each pair contains a constraint and a constant for how the constraints is added in the linear combination. It also takes either a constraint value or a PDF as input, depending on how tight the composit constraint should be defined. Examples of linear combinations are equal energy constraints or equal mass constraints for two particle systems.

Probability distribution functions PDFs are defined in `ProbDistFunc.h`. Currently there is only one derived class in the software package which sets up a Gaussian PDF in `GaussianPDF.h`. The option to expand the software package and add derived classes for other PDFs such as the Breit-Wigner and Initial State Radiation (ISR) exists. The penalty function is calculated from the PDF by

$$g(x) = -2\ln(\text{pdf}(x)) \tag{25}$$

Since both first and second order derivatives of the penalty function are needed in the fit, the class contains functions for calculating these. The class also contains an expectation value for initially setting up the fit.

Matrix Algebra The base class of `MatrixAlgebra.h` contains helper functions for the tedious matrix calculations. The helper functions include functions for vector and matrix algebra such as addition, multiplication, and calculating inverse and transposed matrices. These functions are called upon many times in the fitting procedure and for optimising performance important features have been efficient allocation of memory and pass-by-pointer.

ABC Fit The fitting procedure is defined in the base class of `ABCFit.h`. It takes a list of composite constraints and a maximum number of iterations. In each iteration, corrections to the parameters are calculated as seen by Eqs. 20 and 21. The fit stops either when it has converged i.e. the difference in the χ^2-values between two iterations is small, or when the maximum number of iterations has been reached.

[1] `SumPXConstraint.h`, `SumPyConstraint.h`, `SumPzConstraint.h`

The fit returns a statuscode indicating whether the fit converged. It also returns the number of iterations it took before the fit converged, a χ^2-value on how well the fitted particles fulfil the constraints, number of degrees of freedom, and a list of the fitted particles. Since the particle objects are passed as pointers into the constraints, a constrained fit can be applied in stages where the fit is first applied for one set of constraints followed by another set of constraint instead of requiring all constraints to be fulfilled at once. This could become relevant in cases of convergence issues.

The full `ABCfit++` software package is published at [5].

References

[1] J. M. Torndal. “A Study of Top Anomalous Couplings at a Future $\boldsymbol{e^+e^-}$ Collider”. Master’s Thesis. University of Copenhagen, Sept. 2021 (cit. on p. 1).

[2] Immanuel Kant. *Critik der reinen Vernunft*. de. Riga: Hartknoch, 1781, p. 856 (cit. on p. 1).

[3] Volker Blobel. “Least Squares Methods”. In: *Formulae and methods in experimental data evaluation, with emphasis on high energy physics v.3 : Articles on statistical and numerical methods*. Vol. 3. CERN, Geneva: The European Physical Society, 1983 (cit. on p. 2).

[4] J. B. Hansen. “Triple Gauge-boson Couplings in W pair Production via e^+e^- Annihilation”. PhD thesis. Copenhagen U., 1996 (cit. on p. 5).

[5] Hansen, Jørgen Beck and Torndal, Julie Munch. *ABCfit++*. `https://github.com/Torndal/ABCfitplusplus/`. 2021 (cit. on p. 9).