



ELSEVIER

Nuclear Instruments and Methods in Physics Research A 490 (2002) 366–378

**NUCLEAR
INSTRUMENTS
& METHODS
IN PHYSICS
RESEARCH**
Section A

www.elsevier.com/locate/nima

Helix fitting by an extended Riemann fit

R. Frühwirth^{a,*}, A. Strandlie^b, W. Waltenberger^a

^a*Institute for High Energy Physics of the Austrian Academy of Sciences, Nikolsdorfer Gasse 18, A-1050 Vienna, Austria*

^b*EP Division, CERN, CH-1211 Genève 23, Switzerland*

Received 4 February 2002; received in revised form 5 April 2002; accepted 5 April 2002

Abstract

We present an extension of the Riemann circle fit to a helix fit in space. The method is studied both in barrel- and disk-type detectors. We show results from two simulation experiments, including a comparison to linear regression and to the Kalman filter. An implementation in C++ is described. © 2002 Elsevier Science B.V. All rights reserved.

PACS: 07.05.Kf; 29.85.+c

Keywords: Circle fit; Helix fit; Riemann fit; Kalman filter; Global least-squares fit

1. Introduction

The fitting of observations to a helical track model will be a task of considerable importance in the data analysis chain of the future LHC experiments. As long as the magnetic field in the inner tracker is sufficiently homogeneous the track model can be assumed to be a perfect helix. This has clear advantages over a more general track model if the tracks are fitted by (linearized) least-squares estimators like global regression [1] or the extended Kalman filter [2]. For simple detector shapes (cylinders or planes), the track model can be computed explicitly, and so can the Jacobians that are needed to build the linear model for the global regression or to propagate the track states

and the associated covariance matrices in the Kalman filter. For this reason, the helical track fit is faster than a fit with a more general track model which requires numerical track-and-error propagation, for instance by a Runge–Kutta-type algorithm [1,3].

There are, however, certain drawbacks to linearized least-squares estimators, the main one being that in many cases an approximate knowledge of the track is required in order to find an expansion point of sufficient quality. In the context of the Kalman filter the expansion point is often called the reference track. If the reference track turns out to be too far from the real track an iteration may be required. In extreme cases, the linear approximation may turn out to be not satisfactory at all, in which case one has to resort to non-linear least-squares methods, which tend to be slow and inconvenient. Recently, an explicit third-order approximation to the helix track model has been put forward in the context of fast alignment with tracks [4]. The method is very fast

*Corresponding author. Tel.: +43-1-544-732833; fax: +43-1-544-732854.

E-mail addresses: fruhwirth@hephy.oeaw.ac.at (R. Frühwirth), are.strandlie@cern.ch (A. Strandlie), walten@hephy.oeaw.ac.at (W. Waltenberger).

but breaks down at low momenta where the approximation error gets too large.

In this note, we present a helix fit that does not require linear expansions and does not require prior knowledge of the track, at least as long as multiple scattering can be neglected. In this case, the method described here is significantly faster than a global regression or a Kalman filter, and gives virtually the same precision. If multiple scattering is significant it can be easily incorporated at the price of running an approximate fit first. The method is an extension of the Riemann circle fit, which has been introduced recently [5,6] and which can deal with the lowest transverse momentum that is of interest in the LHC experiments. For this reason, the helix fit is also called a Riemann (helix) fit.

2. The method

2.1. Brief review of the Riemann circle fit

The central idea of the Riemann circle fit is a mapping of the observations, which are supposed to lie in the x - y -plane, on a second-order surface in space. In the standard Riemann fit, this surface is the Riemann sphere, and the mapping is the stereographic projection (see e.g. [7]). In a recently developed modification [8] the surface is the circular paraboloid $w = x^2 + y^2$, and the mapping is parallel to w , i.e. an observation (x_i/y_i) is mapped on the point $(x_i/y_i/w_i = x_i^2 + y_i^2)$. In both cases observations lying on a circle are mapped on points in space lying on a plane. The problem of fitting a circle to the original observations is thus transformed into the problem of fitting a plane to the mapped observations. The latter problem is solved by finding the eigenvector to the smallest eigenvalue of the sample covariance matrix of the mapped observations. Although the stereographic projection is certainly better known, the projection on the paraboloid is simpler and better suited for our purpose and will henceforth be used. We will also make use of the formalism introduced in Ref. [8] in order to take into account multiple scattering in the fit. This will be described in more detail below.

2.2. Extension to the Riemann helix fit

We now assume that there are n three-dimensional detector hits \mathbf{m}_i given in cylindrical coordinates:

$$\mathbf{m}_i = \begin{pmatrix} R_i \Phi_i \\ R_i \\ z_i \end{pmatrix}.$$

Typically, two out of the three observations are affected by measurement errors, whereas the third one is known exactly, or at least within the precision obtained by the alignment. For instance, in a barrel (cylindrical) detector $R\Phi$ and z are measured, and R is known, whereas in a forward (plane) detector $R\Phi$ and R are measured, and z is known.

The helix fit is based on the observation that after a circle fit in the bending plane the arc length between successive hits can be computed. The linear relation between the arc length and the z -position of the helix can then be used to estimate the polar angle ϑ and the z -position of the track in the barrel case, or the polar angle and the radial position of the track in the forward case.

2.2.1. Circle fit

First the hits are mapped on the paraboloid:

$$x_i = R_i \cos(\Phi_i)$$

$$y_i = R_i \sin(\Phi_i)$$

$$w_i = R_i^2.$$

In order to avoid confusion with the z -axis of the detector, the axis of the paraboloid is denoted by w . For the sake of convenience, the mapped hits are arranged in a matrix \mathbf{X} :

$$\mathbf{X} = \begin{pmatrix} x_1 & y_1 & w_1 \\ x_2 & y_2 & w_2 \\ \vdots & \vdots & \vdots \\ x_n & y_n & w_n \end{pmatrix}.$$

The weight matrix of the circle fit is the inverse \mathbf{W} of the covariance matrix $\mathbf{V}_{R\Phi}$ of the observations $R\Phi$. This matrix is diagonal only if multiple scattering can be neglected. If the transverse momentum is low the track does not

intersect the circles $R = R_i$ at a right angle, and the precision of the fit can be improved by correcting for the deviation from the right angle:

$$\mathbf{V}'_{R\phi} = \mathbf{C}\mathbf{V}_{R\phi}\mathbf{C} + \mathbf{S}\mathbf{V}_R\mathbf{S}$$

where \mathbf{C} and \mathbf{S} are diagonal matrices defined by

$$s_{ii} = R_i\kappa/2, \quad c_{ii} = \sqrt{1 - s_{ii}^2}$$

and κ is the curvature of the projected track. Note that in a barrel detector \mathbf{V}_R vanishes. The correction is valid for small impact parameter of the track.

The correction obviously requires prior knowledge of the curvature, so it can be performed only if a reference track is available. It is most important for tracks of low momentum for which multiple scattering is also non-negligible and for which a reference track has to be computed in any case.

Next, the weighted sample mean $\bar{\mathbf{x}}$ and the weighted sample covariance \mathbf{V}_X matrix are computed:

$$\bar{\mathbf{x}} = \mathbf{1}^T \mathbf{W} \mathbf{X} / \mathbf{1}^T \mathbf{W} \mathbf{1}$$

$$\begin{aligned} \mathbf{V}_X &= (\mathbf{X}^T - \bar{\mathbf{x}}^T \mathbf{1}^T) \mathbf{W} (\mathbf{X} - \mathbf{1} \bar{\mathbf{x}}) \\ &= \mathbf{X}^T \mathbf{W} \mathbf{X} - \bar{\mathbf{x}}^T \bar{\mathbf{x}} \cdot \mathbf{1}^T \mathbf{W} \mathbf{1} \end{aligned}$$

where $\mathbf{1}$ is a column vector of ones. The normal vector \mathbf{n} of the fitted plane is the normalized eigenvector to the smallest eigenvalue of \mathbf{V}_X . Finally, the distance c of the plane from the origin is obtained by

$$c = -\bar{\mathbf{x}} \cdot \mathbf{n}.$$

From c and \mathbf{n} the radius ρ and the origin (x_0/y_0) of the circle can be computed [8]:

$$x_0 = -\frac{n_1}{2n_3}, \quad y_0 = -\frac{n_2}{2n_3}, \quad \rho^2 = \frac{1 - n_3^2 - 4cn_3}{4n_3^2}.$$

The computation of the covariance matrix of the estimated plane parameters c and \mathbf{n} has been worked out in [8], as well as the error propagation to the set of circle parameters used in CMS and ATLAS.

2.2.2. Line fit

Once the origin and the radius of the projected circle are known, we can compute the arc length between successive observations. The fastest way

to find the intersections of the fitted circle with the circles $R = R_i$ is in the paraboloid transform. The circle $R = R_i$ transforms to the plane $w = R_i^2$. The intersection of this plane and the fitted plane $c + n_1x + n_2y + n_3w = 0$ is the straight line

$$\begin{aligned} x &= \lambda n_2 \\ y &= -\frac{c + R_i^2 n_3}{n_2} - \lambda n_1 \\ w &= R_i^2. \end{aligned}$$

Intersection of this line with the paraboloid $w = x^2 + y^2$ gives

$$\lambda = \frac{-n_1(c + R_i^2 n_3) \pm n_2 \sqrt{(n_1^2 + n_2^2)R_i^2 - (c + R_i^2 n_3)^2}}{n_2(n_1^2 + n_2^2)}.$$

Plugging in this value of λ in the equation of the line gives the intersection point:

$$\begin{aligned} x_i &= \frac{-n_1(c + R_i^2 n_3) \pm n_2 \sqrt{(n_1^2 + n_2^2)R_i^2 - (c + R_i^2 n_3)^2}}{n_1^2 + n_2^2} \\ y_i &= \frac{-n_2(c + R_i^2 n_3) \mp n_1 \sqrt{(n_1^2 + n_2^2)R_i^2 - (c + R_i^2 n_3)^2}}{n_1^2 + n_2^2}. \end{aligned}$$

Note that no trigonometric functions and only a single square root per layer need to be computed. The only problem is the choice of the correct sign. This can easily be solved by making sure that the point of intersection is close to the measured hit. The computation of the arc length s between successive observations is now trivial, the radius of the fitted circle being known. The result is a vector \mathbf{s} with $s_1 = 0$.

The line fit in a barrel detector: In a detector of the barrel type, the radius R of a hit can be assumed to be known to very high precision, whereas z has a measurement error. In such a detector we make a regression of z on s , using the full covariance matrix of the observation vector \mathbf{z} if multiple scattering cannot be neglected:

$$\mathbf{z} = \mathbf{A}\mathbf{p} + \boldsymbol{\varepsilon} \quad \text{with} \quad \mathbf{A} = \begin{pmatrix} 1 & s_1 \\ \vdots & \vdots \\ 1 & s_n \end{pmatrix},$$

$$\mathbf{p} = \begin{pmatrix} z \\ \tan \lambda \end{pmatrix}, \quad \text{cov}(\boldsymbol{\varepsilon}) = \mathbf{V}_z$$

where $\lambda = \pi/2 - \vartheta$. From the fitted parameters of the straight line we can immediately determine the z of the track in the innermost layer and the polar angle ϑ of the track.

The line fit in a forward detector: In a detector of the forward type, the z -positions can be assumed to be known, and the radial position of the hits are observed. In this case we make a regression of s on z . The covariance matrix of s is to a very good approximation equal to the covariance matrix of R . Multiple scattering can be included, if required. Thus we get the following regression model:

$$s = \mathbf{A}\mathbf{p} + \boldsymbol{\varepsilon} \quad \text{with } \mathbf{A} = \begin{pmatrix} 1 & z_1 \\ \vdots & \vdots \\ 1 & z_n \end{pmatrix},$$

$$\mathbf{p} = \begin{pmatrix} s \\ \tan \vartheta \end{pmatrix}, \quad \text{cov}(\boldsymbol{\varepsilon}) = V_R.$$

From the fitted parameters $\tilde{\mathbf{p}}$ of the straight line we can immediately determine the polar angle ϑ and the s -values of the track in all layers:

$$\tilde{s} = \mathbf{A}\tilde{\mathbf{p}}.$$

Using \tilde{s} and the fitted radius of the circle we can predict the radial positions of the track in all layers. We recall that (x_0/y_0) is the center of the fitted circle, ρ is its radius, and (x_i/y_i) is the intersection of the fitted circle with the circle $R = R_i$. We define the polar angle Θ_1 of (x_1/y_1) with respect to (x_0/y_0) by

$$\Theta_1 = \arctan \frac{y_1 - y_0}{x_1 - x_0}.$$

The predicted polar angles of all points are then given by

$$\tilde{\Theta}_i = \Theta_1 + \tilde{s}_i/\rho.$$

From these we are finally able to predict the radial positions of all hits

$$\tilde{x}_i = x_0 + \rho \cos(\tilde{\Theta}_i)$$

$$\tilde{y}_i = x_0 + \rho \sin(\tilde{\Theta}_i)$$

$$\tilde{R}_i = \sqrt{\tilde{x}_i^2 + \tilde{y}_i^2}.$$

Since we have used the linear relationship between s and z in deriving these predictions they are on

average closer to the true radial positions than the observed ones. This is particularly important in a detector like the forward silicon tracker of CMS, because the precision of the radial position measurement is rather poor in the single-sided layers, whereas Φ is measured much more accurately. Improving the precision of the radial position therefore improves very much the precision of $R\Phi$ and thus the precision of a subsequent circle fit. For this reason, it is advantageous to redo the circle fit after the line fit.

2.3. The Riemann helix fit in cylindrical detectors

We assume that in a detector of the barrel type the radial position of the hits is known to a precision that is better than the precision in $R\Phi$. In this case, the error on R can be neglected altogether. The helix fit requires the following steps:

1. Reference track

This step needs to be carried out only for low momentum tracks.

- (a) *Circle fit without multiple scattering:* The weight matrix is diagonal; the diagonal terms contain the variances of the $R\Phi$ observations. The intersections of the fitted circle with the circles $R = R_i$ and the arc lengths between successive layers are computed.
- (b) *Line fit without multiple scattering:* The weight matrix is diagonal; the diagonal terms contain the variances of the z observations.
- (c) *Covariance matrix of multiple scattering (optional):* The covariance matrix is evaluated separately for the two fits. We use the simple straight line approximation of [9]. For the circle fit we have:

$$\text{cov}(R\Phi_k, R\Phi_l)$$

$$= \sum_{j=1}^{\min(k-1, l-1)} (R_k - R_j)(R_l - R_j) \frac{\sigma_j^2}{\sin^2 \vartheta}$$

where σ_j^2 is the variance of the projected multiple scattering angle in layer j . In a

similar vein, we have for the line fit:

$$\text{cov}(z_k, z_l) = \sum_{j=1}^{\min(k-1, l-1)} (s_k - s_j)(s_l - s_j) \frac{\sigma_j^2}{\sin^4 \vartheta}.$$

The correlations between $R\Phi$ and z are neglected. We will show below that the effect of this is negligible, except for very low momenta.

2. Circle fit

For low momentum tracks the covariance matrix $V_{R\Phi}$ is corrected for non-orthogonal intersection of the cylinders, using the curvature of the reference track. Then the circle fit proceeds as outlined above, using the inverse of the corrected $V_{R\Phi}$ as its weight matrix. The fitted circle is intersected with the measurement layers, and the arc length between successive layers are computed.

3. Line fit

The line fit is a regression of z on s , using the inverse of V_z as its weight matrix. The fitted regression parameters are z at the innermost layer and $\tan \vartheta$. The combined parameters of the circle and of the line fit fully specify the fitted helix. The joint covariance matrix of the helix parameters is built from the covariance matrices of the two fits, the correlations being neglected.

2.4. The Riemann helix fit in plane detectors

In a detector of the forward type the detectors can be assumed to be planes at fixed z -positions. The observations are now $R\Phi$ and R . As the error on R cannot be neglected the procedure is slightly more complicated than in the case of cylindrical detectors.

1. Reference track

The reference track is always computed.

- (a) *Circle fit without multiple scattering*: The weight matrix is diagonal; the diagonal terms contain the variances of the $R\Phi$ observations. The intersections of the fitted circle with the circles $R = R_i$ and the arc lengths between successive observations are computed.

- (b) *Line fit without multiple scattering*: The weight matrix is diagonal; the diagonal terms contain the variances of the R observations. After the fit the predicted radial positions \tilde{R}_i of the track are computed. The variances of the $R\Phi$ observations are recomputed using the predicted \tilde{R}_i .

- (c) *Covariance matrix of multiple scattering (optional)*: The covariance matrix is evaluated separately for the two fits. For the circle fit the predicted radial positions from the reference track are used. The computation is the same as in cylindrical detectors:

$$\text{cov}(R\Phi_k, R\Phi_l) = \sum_{j=1}^{\min(k-1, l-1)} (R_k - R_j)(R_l - R_j) \frac{\sigma_j^2}{\sin^2 \vartheta}.$$

For the line fit we have

$$\text{cov}(R_k, R_l) = \sum_{j=1}^{\min(k-1, l-1)} (z_k - z_j)(z_l - z_j) \frac{\sigma_j^2}{\cos^4 \vartheta}.$$

The correlations between $R\Phi$ and R are neglected. We will show below that the effect of this is negligible.

2. Preliminary circle fit

For low momentum tracks the covariance matrix $V_{R\Phi}$ is corrected for non-orthogonal intersection of the cylinders, using the curvature of the reference track. Then the circle fit proceeds as outlined above, using the inverse of the corrected $V_{R\Phi}$ as its weight matrix. The fitted circle is intersected with the circles $R = R_i$, and the arc length between successive observations are computed.

3. Preliminary line fit

The line fit is a regression of s on z , using the inverse of V_R as its weight matrix. The predicted radial positions of the helix at all planes $z = z_i$ are computed.

4. Corrections using the preliminary fit

Using the predicted radial positions \tilde{R}_i , the observations $R\Phi$ and their observations

variances are recomputed, and the entire matrix $V_{R\Phi}$ is corrected for non-orthogonal intersection of the circles $R = \tilde{R}_i$, neglecting the errors on \tilde{R}_i .

5. Final fit

Steps 2 and 3 are repeated. The combined parameters of the circle and of the line fit fully specify the fitted helix. The joint covariance matrix of the helix parameters is built from the covariance matrices of the two fits, the correlations being neglected.

3. Results of simulation experiments and comparison with other methods

Two simulation experiments were conducted in order to compare the Riemann fit with other methods, in particular, the Kalman filter and the global fit (linear regression). The main purpose was to compare the precision and the speed of the respective methods. It was therefore assumed that the pattern recognition problem had already been solved. As both the Riemann fit and its competitors are purely least-squares estimators they are affected by outliers or noise in the same way, and thus the assumption of a perfect pattern recognition implies no loss of generality.

3.1. Simulation experiment 1: cylindrical detectors

3.1.1. The model detector

The detector used in the simulation experiment is a simplified model of the barrel part of the CMS Inner Tracker. It has 3 pixel and 10 silicon strip layers. The radial position of each layer, its resolution in $R\Phi$, its resolution in z , and its thickness in radiation lengths are listed in Table 1. With the exception of the thickness, which is approximate, the values have been obtained by analyzing a simulated sample of 1000 tracks generated by the detailed tracker simulation.

3.1.2. Generation of the track samples

We have generated samples of 50 000 tracks in each run of the simulation experiment. A run corresponds to a fixed value of p_T . The following

Table 1

Layout of the barrel-type model detector

Layer	R (mm)	$\sigma_{R\Phi}$ (μm)	σ_z (mm)	X/X_0
1	43	10	0.02	0.04
2	72	10	0.02	0.04
3	110	10	0.02	0.04
4	270	25	0.36	0.02
5	362	25	0.36	0.02
6	441	40	34.0	0.02
7	515	40	34.0	0.02
8	633	40	0.7	0.02
9	719	40	0.7	0.02
10	805	60	54.0	0.02
11	891	60	54.0	0.02
12	988	40	54.0	0.02
13	1103	40	54.0	0.02

values of p_T have been used:

$$p_T = 1, 1.5, 2.5, 5, 10, 20, 50 \text{ GeV}/c.$$

The remaining initial track parameters on the innermost cylinder were generated according to a uniform distribution in the following intervals:

$$0 < \Phi < 2\pi, -5 \text{ cm} < z < 5 \text{ cm}, 0.7 < \vartheta < \pi/2.$$

The initial φ was set equal to the initial Φ . Multiple scattering in the detectors was simulated according to the standard Highland formula [10]. Energy loss was neglected.

3.1.3. Resolution and timing

The Riemann fit (RF) was compared to an extended Kalman filter (KF) and to a linearized regression (global fit, GF). All methods were tested on different levels of precision as far as the treatment of multiple scattering is concerned. A summary is shown in Table 2. The approximate covariance matrix of multiple scattering is the one described in Section 2.4, whereas the exact one is obtained by exact error propagation in the helical track model.

The baseline of the comparison was the Kalman filter on level 3, including multiple scattering, in all cases. Fig. 1 shows the performance on level 0, neglecting multiple scattering altogether. There is perfect agreement between the three methods. The effect of multiple scattering is visible but for the highest momentum.

The execution times of the three methods are shown in Table 3. The figures refer to the C++ implementation described in Section 4. The error (r.m.s.) of the relative timings is about 2%. The baseline of the comparison is again the Kalman filter on level 3. The Riemann fit is faster than the Kalman filter by about 30%. The global fit is somewhat slower than the Kalman filter. The Kalman filter itself is affected very little by switching off multiple scattering, as multiple scattering is included simply by adding a few terms to the local 5×5 covariance matrix, which is non-diagonal in any case.

Table 2
Levels of treatment of multiple scattering

Level	Covariance matrix of multiple scattering	Applies to
0	None	All methods
1	Approximate	GF, RF
2	Exact, but no correlations between projections	GF, RF
3	Exact, including all correlations	GF, KF

The performance on the higher levels is of course much better than on level 0. Fig. 2 shows the relative generalized variance on levels above 0. It turns out that the correlations between the projections (level 3) are necessary for attaining the full precision at low momentum. There are significant differences between levels 1 and 2 only at the very lowest p_T ; in all other cases the approximation used at level 1 (see Section 2.3) is adequate. The behaviour of the global fit and of the Riemann fit is very similar at levels 1 and 2.

The relative timing is summarized in Table 4. The Riemann fit on level 1 is still somewhat faster than the Kalman filter on level 3, and clearly faster than the global fit on the same level. The absolute timing of the baseline is about 0.7 ms per track, on a 1.1 GHz Athlon processor running Linux 2.2.

Table 3
Relative timing of all methods on level 0

Method	Level	t_{rel}
Kalman filter	0	0.98
Global fit	0	1.09
Riemann fit	0	0.72

The baseline is the Kalman filter on level 3.

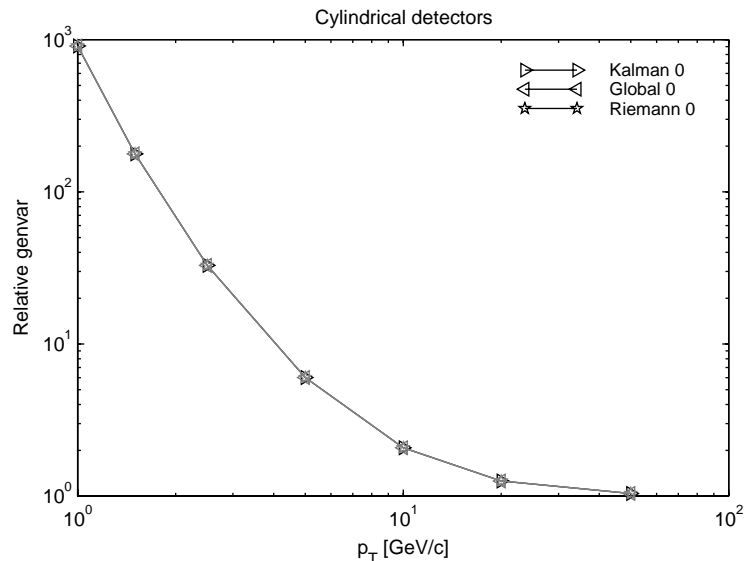


Fig. 1. Relative generalized variance of all methods on level 0, neglecting multiple scattering. The baseline is the Kalman filter on level 3, including multiple scattering.

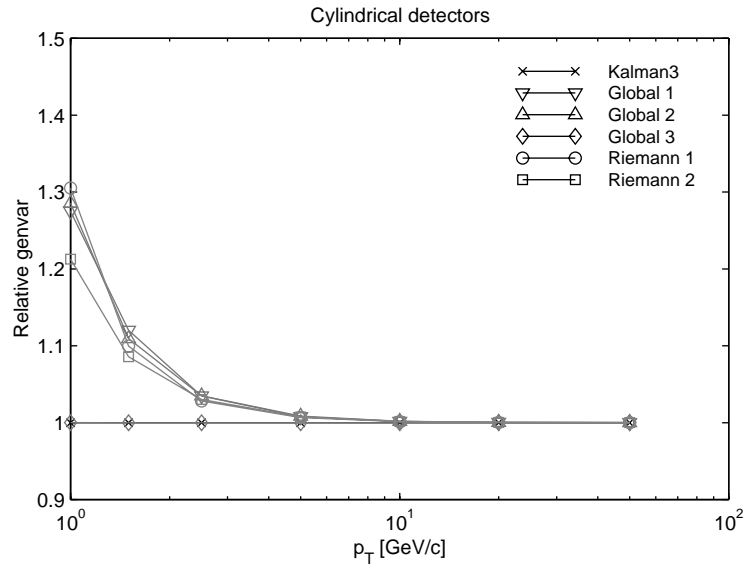


Fig. 2. Relative generalized variance of all methods on levels above 0. The baseline is the Kalman filter on level 3.

Table 4
Relative timing of all methods on levels above 0

Method	Level	t_{rel}
Kalman filter	3	1.00
Global fit	1	1.25
Global fit	2	1.51
Global fit	3	1.54
Riemann fit	1	0.92
Riemann fit	2	1.19

The baseline is the Kalman filter on level 3.

The relative speed of the methods depends to some extent on the programming style and on the implementation. In the prototype version, which was written in an interpreter language (MATLAB [11]), the Kalman filter is always the fastest one, even on level 1, and both the global fit and the Riemann fit are faster on level 2 than on level 1. This is mainly because of the fact that in MATLAB interpreted loops are much slower than the compiled loops in the built-in matrix functions. Table 5 is intended to give the reader some idea on where the computing time is spent in the C++ implementation. For the sake of comparison also the time spent in the generation of the tracks is shown. The time required for building the

covariance matrix refers to the global covariance matrix required by the Riemann fit and the global fit, but not by the Kalman filter, which explains why the corresponding figure is equal to 0.

3.2. Simulation experiment 2: plane detectors

3.2.1. The model detector

The detector used in the simulation experiment is a simplified model of the forward part of the CMS Inner Tracker. It has 2 pixel and 12 silicon strip layers. The z -position of each layers, its resolution in $R\Phi$, its resolution in R , and its thickness in radiation lengths are listed in Table 6.

3.2.2. Generation of the track samples

We have again generated samples of 50 000 tracks in each run of the simulation experiment, using the same set of transverse momenta as before. The remaining initial track parameters on the innermost plane were generated according to a uniform distribution at $z = 0$:

$$-1 \text{ mm} < x < 1 \text{ mm}, -1 \text{ mm} < y < 1 \text{ mm}, \\ 0.2 < \vartheta < 0.6, 0 < \varphi < 2\pi.$$

Table 5
Breakdown of execution time on levels above 0 (in ms per track)

Method	Level	Generation	Ref. track	Cov. matrix	Fit + overhead
Kalman filter	3	0.32	0.27	0.00	0.44
Global fit	1	0.32	0.27	0.13	0.48
Global fit	2	0.32	0.27	0.31	0.48
Global fit	3	0.32	0.27	0.32	0.50
Riemann fit	1	0.32	0.27	0.13	0.26
Riemann fit	2	0.32	0.27	0.32	0.26

Table 6
Layout of the forward type model detector

Layer	Z (mm)	$\sigma_{R\Phi}$ (μm)	σ_R (mm)	X/X_0
1	335	11	0.008	0.04
2	465	11	0.025	0.04
3	766	60	0.55	0.02
4	896	60	0.55	0.02
5	1046	60	0.55	0.02
6	1302	100	25.0	0.02
7	1402	100	25.0	0.02
8	1517	250	32.0	0.02
9	1647	625	32.0	0.02
10	1802	625	32.0	0.02
11	1982	625	32.0	0.02
12	2192	1000	33.0	0.02
13	2412	940	33.0	0.02
14	2652	860	33.0	0.02

The initial φ was set equal to the initial Φ . Multiple scattering in the planes was simulated according to the standard Highland formula [10]. Energy loss was neglected.

3.2.3. Resolution and timing

Fig. 3 shows the performance on level 0, neglecting multiple scattering altogether. There is close agreement between the three methods, the Riemann fit being slightly better. The effect of neglecting multiple scattering is much weaker than in the barrel, because of the higher absolute momenta of the tracks.

The execution times of the three methods are shown in Table 7. The baseline is again the Kalman filter on level 3. The Riemann fit is now slower than the other methods, because it needs to be iterated if full precision is required. If the Riemann fit is intended only as a fast

preliminary fit, e.g. to obtain a good reference track, the iteration can be skipped, and the Riemann fit is again competitive in terms of execution time.

The performance on the higher levels is shown in Fig. 4. The correlations between the projections (level 3) are now less important than before. There is very little difference between levels 1 and 2. At the very lowest p_T the Riemann fit is slightly worse than the other methods. The relative timing is shown in Table 8. The Riemann fit on level 1 is again slower than both the Kalman filter and the global fit, because of the need to iterate. The absolute timing of the baseline is about 0.85 ms per track.

3.3. Discussion

In a detector of the barrel type the Riemann fit is a viable alternative to the Kalman filter and to the global fit, especially if multiple scattering can be neglected. It is faster than the Kalman filter in the case of cylindrical detectors. As no derivative matrices have to be computed the relative speed of the Riemann fit can be expected to increase for more complicated detector shapes. On the other hand, it presupposes a perfect helical track model. Its applicability is therefore confined to regions of the detector where the field is sufficiently homogeneous. It is also unsuitable in cases of strong energy loss. In a detector of the forward type the Riemann fit suffers from the fact that it has to be iterated in order to achieve full precision. As the forward region of a colliding beam experiment also tends to have less homogeneous magnetic field than the central region, the Kalman filter is probably the method of choice.

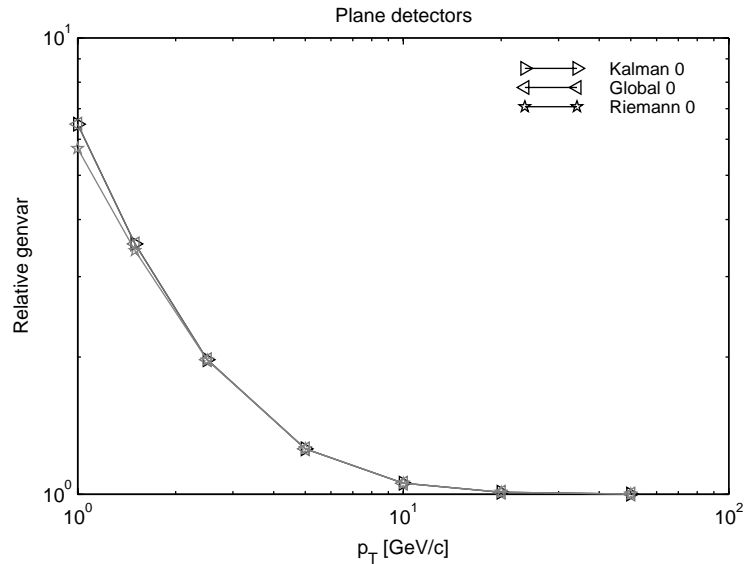


Fig. 3. Relative generalized variance of all methods on level 0, neglecting multiple scattering. The baseline is the Kalman filter on level 3, including multiple scattering.

Table 7
Relative timing of all methods on level 0

Method	Level	t_{rel}
Kalman filter	0	0.98
Global fit	0	1.23
Riemann fit	0	1.61

The baseline is the Kalman filter on level 3.

4. Implementation in C++

The C++ implementation is heavily influenced by the ORCA framework [12], although it can be used as a standalone project. It handles both the generation and the reconstruction of the tracks, assuming only the pattern recognition problem to be solved. The program is built on the CLHEP [13] and the STL [14] libraries, with an optional GUI using HepVis [15] and Motif [16]. The GUI frontend is shown in Fig. 5.

The detector description as well as the ranges of the initial values of the tracks are defined in two separate configuration files, which—along with almost twenty different command line options (see Table 9)—allow for a very flexible usage of the program.

4.1. Class design

Since prototyping had been done in MATLAB, the paradigm shift from procedural MATLAB code (“spaghetti code”) to the object orientedness of C++ demanded a strong emphasis on design issues. The general layout we came up with is depicted in Fig. 6. As can be seen, a clear separation was introduced between what we view as the “physical data”, i.e. everything associated with the detector and its elements, as well as the particle hits, and the “mathematical description”—the plane fitted by the RF, the state vectors and their covariance matrices, and all manipulations acting on these objects.

Only the Track class cannot be fit in any of the categories; it somewhat resembles the “synthesis” of the two. This class has as its member data the hits and the covariance matrix; the fitting algorithms are distinct member functions of this class.

The Plane class is only a “helper class” that has as its member data the coefficients c and \mathbf{n} that define a plane $\mathbf{x} \cdot \mathbf{n} + c = 0$ in Euclidean space. Its only member functions are transformations from and to the plane class.

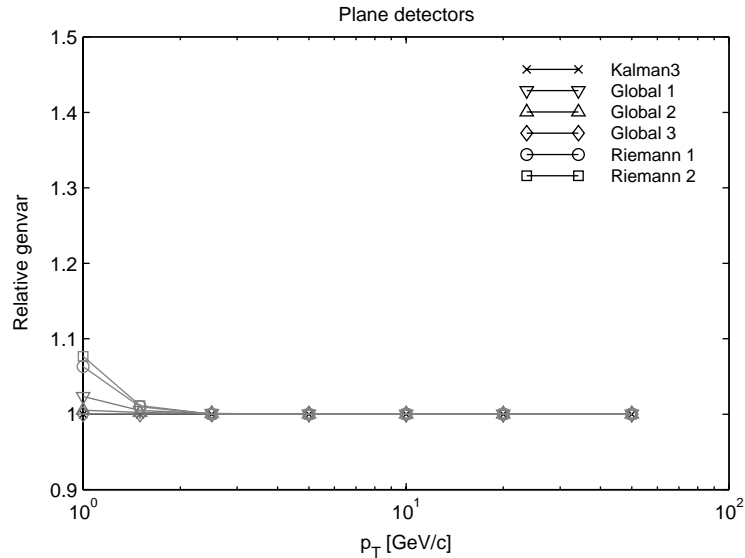


Fig. 4. Relative generalized variance of all methods on levels above 0. The baseline is the Kalman filter on level 3.

Table 8
Relative timing of all methods on levels above 0

Method	Level	t_{rel}
Kalman filter	3	1.00
Global fit	1	1.42
Global fit	2	1.68
Global fit	3	1.70
Riemann fit	1	1.81
Riemann fit	2	2.17

The baseline is the Kalman filter on level 3.

Understanding and changing the C++ implementation should therefore be both easier and less error-prone than in the MATLAB program.

4.2. Builds

We currently support the Linux and Solaris platforms. Since the project is intended to be both ORCA-related and standalone, two build mechanisms are offered. The package is shipped with a BuildFile, such that a compile from within an ORCA install should be straightforward. For the needs of a standalone install we also support a

configure/make mechanism. Note that the configure is not a GNU autoconf script, but a simple shell script that will ask a few questions. We did not implement a GNU style autoconf mechanism, mainly because the development environments in the high-energy physics community are often “non-GNUish”. The CLI binary has been successfully built and tested on (with the build mechanism given in brackets):

- Red Hat 6.1 (make, scram),
- Red Hat 6.2 (make, scram),
- Solaris 7 (make),
- unstable (sid) Debian GNU/Linux 3.0 (make).

The GUI frontend has been successfully built and tested on:

- Red Hat 6.2 (make),
- unstable (sid) Debian GNU/Linux 3.0 (make).

4.3. Availability

The code can be obtained from the authors on request.

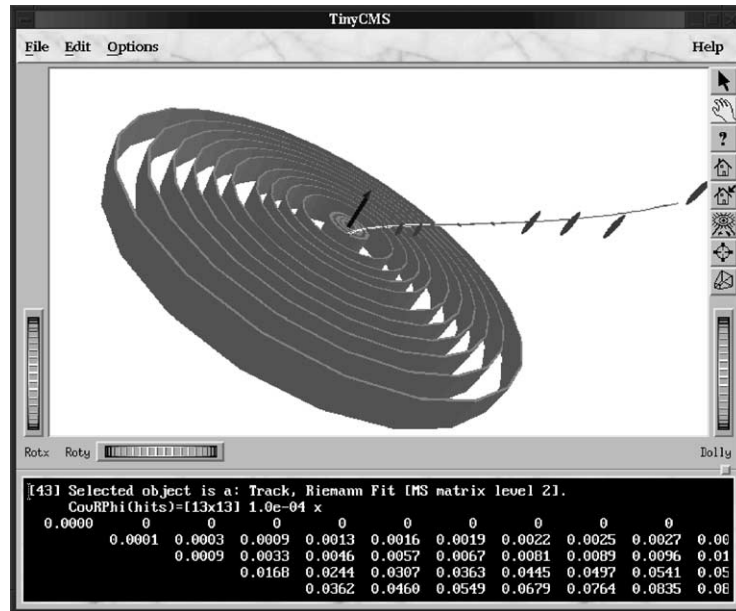


Fig. 5. Snapshot of the GUI frontend.

Table 9

Some command line options of the C++ programs

Operation:

- R, -riemann [levels] Riemann fit
- G, -global [levels] Global fit
- K, -kalman [levels] Kalman filter

Options:

- n, -ntracks [num] Number of tracks to simulate (default: 1)
- f, -detfile [file] Read detector configuration from [file]
- i, -initfile [file] Read initial values from [file]
- s, -seed [num] Set random seed to num
- a, -scatter [0/1/2] Scatter?
- m, -smear [0/1/2] Smear?
- p, -persistent Persistent storage of tracks? Needs a lot of memory per track, but allows for precise timing

Output:

- c, -cms[=[pre]] Write statistics (CMS coords)
- d, -delphi[=[pre]] Write statistics in Delphi coords
- T, -timing Show elapsed CPU times for different code regions

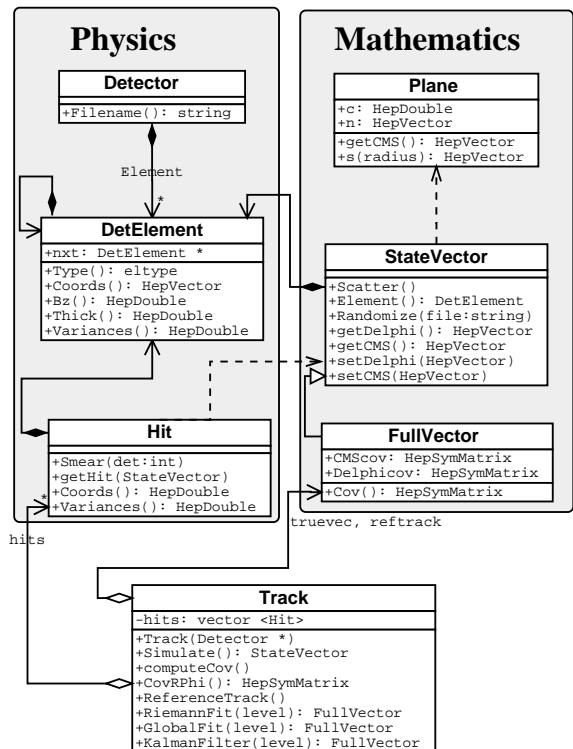


Fig. 6. The C++ class hierarchy, along with a few illustrating methods.

References

- [1] R. Frühwirth, M. Regler, R. Bock, H. Grote, D. Notz, Data Analysis Techniques for High-Energy Physics, Cambridge University Press, Cambridge, 2000.
- [2] R. Frühwirth, Nucl. Instr. and Meth. A 262 (1987) 444.
- [3] J. Myrheim, L. Bugge, Nucl. Instr. and Meth. 160 (1979) 43.
- [4] M. Dima, J. Phys. G 27 (2001) 1481.
- [5] A. Strandlie, J. Wroldsen, R. Frühwirth, B. Lillekjendlie, Comput. Phys. Commun. 131 (2000) 95.
- [6] A. Strandlie, R. Frühwirth, Nucl. Instr. and Meth. A 480 (2002) 34.
- [7] L. Ahlfors, Complex Analysis: An Introduction to the Theory of Analytic Functions of One Complex Variable, International Series in Pure and Applied Mathematics, McGraw-Hill, New York, 1979.
- [8] A. Strandlie, J. Wroldsen, R. Frühwirth, Nucl. Instr. and Meth. A 488 (2002) 332.
- [9] M. Regler, Multiple scattering in least squares fitting, in: R.K. Bock, et al. (Eds.), Formulae and Methods in Experimental Data Evaluation, Vol. 2, European Physical Society, CERN, Geneva, 1984, pp. G1–G11.
- [10] Particle Data Group, D.E. Groom et al., Eur. Phys. J. C 15 (2000) 166.
- [11] The MathWorks, Inc., MATLAB Language Reference Manual, Version 5.1, 1997.
- [12] ORCA—Object-Oriented Reconstruction for CMS Analysis. URL <http://cmsdoc.cern.ch/orca>.
- [13] CLHEP—A class library for high energy physics. URL <http://wwwinfo.cern.ch/asd/lhc++/clhep>.
- [14] A.A. Stepanov, M. Lee, The Standard Template Library, Technical Report X3J16/94-0095, WG21/N0482 (1994). URL citeseer.nj.nec.com/stepanov95standard.html.
- [15] The HEPVis library. URL <http://www-pat.fnal.gov/graphics/HEPVis/www>.
- [16] P. Ferguson, D. Brennan, Motif Reference Manual, The Definitive Guides to the X Window System, Vol. 6B, O'Reilly & Associates, Sebastopol, California, 1993.