

Effective circle fitting for particle trajectories

V. Karimäki

SEFT, University of Helsinki, Helsinki, Finland

Received 5 December 1990

We present a fast method for circular trajectory fitting. Our method is based on an explicit solution of an nonlinear least-squares problem to fit the circle curvature, direction and position parameters. The advantage with respect to previously published methods is that these parameters are Gaussian behaved which implies more reliable error estimation of the fitted parameters. We present formulae for error estimation as well as for propagation of parameters and error matrix to another point of reference.

1. Introduction

Circle fitting methods are widely used in particle data analysis as well as in other fields of scientific data handling. Tracking chambers in large particle detectors are usually placed in an almost uniform magnetic field in order to measure particle momenta. Particle trajectories form circular arcs in the plane normal to the magnetic field. Consequently various pattern recognition and fitting methods rely on algorithms dealing with circles.

Modern collider detectors measure a large number of points along the particle trajectories. In the present day collider experiments there are typically several tens of tracks in a single event. Tracking devices in future large hadron colliders LHC and SSC should be able to measure hundreds of tracks and detect maybe tens of thousands of point per event. Fast and practical fitting algorithms are obviously of great interest.

In this article we introduce a circle fitting algorithm which is a generalization of the method we have developed earlier [1] for track fitting in the CERN SPS collider experiment UA1 and which has been adapted also by the LEP experiments OPAL and L3 [2,3]. Other fast circle fitting methods have been presented in the literature (see refs. [4,5]).

Our method is particularly suitable for particle experiments since it yields directly the circle curvature, direction and distance (from a fixed point) which are all Gaussian and well behaved at the straight track limit. In the following we describe the method in detail. In section 2 we present the basic ideas of the method, in section 3 the detailed formulae to solve the fitted parameters are given and in sections 4 and 5 we discuss the error estimation and transformation to a new reference point.

2. Outline of the method

The least squares circle fitting problem involves the minimisation of the function

$$\chi^2 = \sum_i w_i \epsilon_i^2, \quad (1)$$

where w_i are weights and ϵ_i are measurement residuals normal to the trajectory:

$$\epsilon_i = \pm \left[\sqrt{(x_i - a)^2 + (y_i - b)^2} - R \right]. \quad (2)$$

Here x_i and y_i are the measured coordinates and the parameters a , b and R are the coordinates of the circle centre and radius, respectively. The sign convention in eq. (2) takes into account the two possible directions of propagation along the circle. Our convention is such that positive sign implies clockwise propagation.

In the above exact form the minimizing problem is nonlinear and requires an iterative solution. In almost all practical problems, however, the condition $|\epsilon_i| \ll R$ is valid. Under this condition we have the approximation

$$\epsilon_i \approx \epsilon_i = \pm \frac{1}{2} R^{-1} \left[(x_i - a)^2 + (y_i - b)^2 - R^2 \right], \quad (3)$$

which holds true for a high precision. The relative error on ϵ_i is only of the order of $\frac{1}{2} \epsilon_i R^{-1}$. This expression has been used in refs. [4,5] to solve the circle fitting problem non-iteratively for the parameters a , b and R . The drawback in using these parameters for particle trajectory applications is that they are non-Gaussian. Furthermore, the solution suffers from truncation problems for high momentum tracks (large a , b , R) and is seriously ill behaved at the straight track limit.

The most natural parameters for particle trajectories are the curvature ($\rho = \pm 1/R$), the distance of closest approach (d) to the origin and the direction of propa-

gation (ϕ) at the point of closest approach. In terms of these parameters the expression (3) reads:

$$\varepsilon_i = \frac{1}{2}\rho r_i^2 - (1 + \rho d)r_i \sin(\phi - \varphi_i) + \frac{1}{2}\rho d^2 + d, \quad (4)$$

where r_i and φ_i are the polar coordinates of the point i . This expression is valid for any trajectory length up to a full circle. The eq. (4) represents in fact the equation of circle in polar coordinates if $\varepsilon_i = 0$. This expression is used also in ref. [6] where an iterative three-dimensional track and vertex fitting method is introduced.

The parameters ρ , ϕ , d are Gaussian and they are also well behaved at the straight track limit. Our sign convention in eq. (4) is such that ρ is positive for particles propagating clockwise along the circle and negative otherwise. The sign of the distance parameter d can be verified from $d = x_d \sin \phi - y_d \cos \phi$ where (x_d, y_d) is a point on the circle closest to the origin. The sign is positive in case the vector (x_d, y_d) and the track direction form a right-handed system and negative otherwise.

Our starting point for non-iterative circle fit is the eq. (4) from which it follows that

$$\varepsilon_i = (1 + \rho d)\eta_i,$$

where

$$\eta_i = \kappa r_i^2 - r_i \sin(\phi - \varphi_i) + \delta, \quad (5)$$

with

$$\kappa = \frac{1}{2} \frac{\rho}{1 + \rho d}; \quad \delta = \frac{1 + \frac{1}{2}\rho d}{1 + \rho d} d. \quad (6)$$

In this notation the χ^2 function reads

$$\chi^2 = (1 + \rho d)^2 \hat{\chi}^2, \quad (7)$$

where

$$\hat{\chi}^2 = \sum_i w_i \eta_i^2. \quad (8)$$

The parameters ρ , ϕ , d that minimize the function $\hat{\chi}^2$ are very precisely those minimizing the true χ^2 . We have verified this by a sample of simulated circles with $0 \leq |\rho| < 4 \text{ m}^{-1}$, $|d| < \min(1 \text{ m}, 0.75R)$ and $\langle \varepsilon_i \rangle = 300 \text{ } \mu\text{m}$. The number of points n varied in the range $50 < n < 150$ and the spacing between the points was 1 cm. The parameters minimizing the $\hat{\chi}^2$ yield χ^2 values which are only of the order of 10^{-3} from the true minimum.

We have derived a correction formula so as to find the parameter values at the true minimum. This is discussed in section 4. The corrections appear to be very small, typically less than 5% of the statistical errors of the fitted parameters.

The problem to minimize $\hat{\chi}^2$ is a nonlinear LSQ problem which in principle must be solved iteratively. It turns out, however, that there exists an explicit non-iter-

ative solution to the problem. We discuss the solution in the next section.

3. Non-iterative solution

Following the above outline, the procedure to solve the LSQ circle fitting problem non-iteratively goes as follows:

- Minimize $\hat{\chi}^2 = \sum_i w_i \eta_i^2$ with respect to the parameters κ , ϕ and δ .
- Solve the curvature and distance parameters ρ and d by inverting eqs. (6).

In order to minimize $\hat{\chi}^2$ one has to solve the normal equations:

$$\begin{aligned} \frac{1}{2} S_w^{-1} \partial \hat{\chi}^2 / \partial \kappa &= \kappa \langle r^4 \rangle - \sin \phi \langle x r^2 \rangle + \cos \phi \langle y r^2 \rangle + \delta \langle r^2 \rangle = 0, \\ \frac{1}{2} S_w^{-1} \partial \hat{\chi}^2 / \partial \phi &= -\kappa \cos \phi \langle x r^2 \rangle - \kappa \sin \phi \langle y r^2 \rangle \\ &\quad + \frac{1}{2} \sin 2\phi \langle x^2 - y^2 \rangle - \cos 2\phi \langle xy \rangle \\ &\quad - \delta \cos \phi \langle x \rangle - \delta \sin \phi \langle y \rangle = 0, \end{aligned} \quad (9)$$

$$\frac{1}{2} S_w^{-1} \partial \hat{\chi}^2 / \partial \delta = \kappa \langle r^2 \rangle - \sin \phi \langle x \rangle + \cos \phi \langle y \rangle + \delta = 0,$$

where the notation $\langle \rangle$ means a weighted average and $S_w = \sum_i w_i$ is the sum of the weights. For example $\langle r^4 \rangle = \sum_i w_i r_i^4 / S_w$. We find the following explicit solution to the group of eqs. (9):

$$\begin{aligned} \phi &= \frac{1}{2} \arctan(2q_1/q_2), \\ \kappa &= (\sin \phi C_{xr^2} - \cos \phi C_{yr^2}) / C_{r^2, r^2}, \end{aligned} \quad (10)$$

$$\delta = -\kappa \langle r^2 \rangle + \sin \phi \langle x \rangle - \cos \phi \langle y \rangle,$$

where $q_1 = C_{r^2, r^2} C_{xy} - C_{xr^2} C_{yr^2}$ and $q_2 = C_{r^2, r^2} (C_{xx} - C_{yy}) - C_{xr^2}^2 + C_{yr^2}^2$. The coefficients C_{xx}, \dots are the statistical covariances of the measurements x_i, y_i and $r_i^2 = x_i^2 + y_i^2$ as follows:

$$\begin{aligned} C_{xx} &= \langle x^2 \rangle - \langle x \rangle^2, \\ C_{xy} &= \langle xy \rangle - \langle x \rangle \langle y \rangle, \\ C_{yy} &= \langle y^2 \rangle - \langle y \rangle^2, \\ C_{xr^2} &= \langle x r^2 \rangle - \langle x \rangle \langle r^2 \rangle, \\ C_{yr^2} &= \langle y r^2 \rangle - \langle y \rangle \langle r^2 \rangle, \\ C_{r^2, r^2} &= \langle r^4 \rangle - \langle r^2 \rangle^2. \end{aligned} \quad (11)$$

The curvature and distance parameters ρ , d are then given by inversion of eqs. (6):

$$\rho = \frac{2\kappa}{\sqrt{1 - 4\delta\kappa}}; \quad d = \frac{2\delta}{1 + \sqrt{1 - 4\delta\kappa}}. \quad (12)$$

At the limit $|\rho d| \ll 1$ the solution is similar to one presented in ref. [1]. There the fitting procedure in-

volved translation of the origin to a point near the circle so as to make d small. In the new formalism presented here this limitation is removed.

It is of interest to calculate the value of the χ^2 function. A straightforward method is to compute it by summation using the eq. (7). This is a time consuming method for circles with many measured points. We have derived the following direct formula for fast calculation of the χ^2 as:

$$\chi^2 = S_w(1 + \rho d)^2 (\sin^2 \phi C_{xx} - 2 \sin \phi \cos \phi C_{xy} + \cos^2 \phi C_{yy} - \kappa^2 C_{r^2 r^2}). \quad (13)$$

In this expression the terms added and subtracted are large compared to the resulting χ^2 value. Therefore, in order to avoid truncation problems, eq. (13) should be computed in double precision on 32-bit machines.

4. Error estimation and correction formula

As mentioned above, the circle parameters ρ , ϕ and d are statistically Gaussian distributed. It is therefore natural that the error estimation is performed in terms of these parameters. For reason of formulation let us denote the parameters ρ , ϕ , d as p_1 , p_2 , p_3 , respectively. The error matrix \mathbf{V} of the parameters solved by the LSQ method is given by the inverse of a matrix with elements

$$(V^{-1})_{jk} = \sum_i w_i \frac{\partial \epsilon_i}{\partial p_j} \frac{\partial \epsilon_i}{\partial p_k}, \quad (14)$$

computed at minimum χ^2 . This is a symmetric 3 by 3 matrix. The matrix elements (14) can be calculated in terms of the mean values $\langle x \rangle$, $\langle xy \rangle$, ... or, equivalently, in terms of the sums $\Sigma_i w_i$, $\Sigma_i w_i x_i$, etc. These are exactly the sums needed to solve the circle parameters. Therefore the extra time to calculate the error matrix is independent of the number of measured points.

We have calculated the explicit formulae for the matrix elements (14) and they are given below:

$$\begin{aligned} (V^{-1})_{\rho\rho} &= \frac{1}{4} S_r^4 - d \left\{ S_\delta - d \left[S_{\alpha\alpha} + \frac{1}{2} S_{r^2} - d \left(S_\alpha - \frac{1}{4} d S_w \right) \right] \right\}, \\ (V^{-1})_{\rho\phi} &= -u \left[\frac{1}{2} (\cos \phi S_{xr^2} + \sin \phi S_{yr^2}) - d \left(S_\gamma - \frac{1}{2} d S_\beta \right) \right], \\ (V^{-1})_{\phi\phi} &= u^2 (\cos^2 \phi S_{xx} + \sin 2\phi S_{xy} + \sin^2 \phi S_{yy}), \\ (V^{-1})_{\rho d} &= \rho \left(-\frac{1}{2} S_\delta + d S_{\alpha\alpha} \right) + \frac{1}{2} u S_{r^2} \\ &\quad - \frac{1}{2} d \left[(2u + \rho d) S_\alpha - u d S_w \right], \\ (V^{-1})_{\phi d} &= u (\rho S_\gamma - u S_\beta), \\ (V^{-1})_{dd} &= \rho (\rho S_{\alpha\alpha} - 2u S_\alpha) + u^2 S_w, \end{aligned} \quad (15)$$

where we use the notations $S_x = \Sigma w_i x_i$, $S_y = \Sigma w_i y_i$ and so on together with,

$$\begin{aligned} u &= 1 + \rho d, \\ S_\alpha &= \sin \phi S_x - \cos \phi S_y, \\ S_\beta &= \cos \phi S_x + \sin \phi S_y, \\ S_\gamma &= (\sin^2 \phi - \cos^2 \phi) S_{xy} + \sin \phi \cos \phi (S_{xx} - S_{yy}), \\ S_\delta &= \sin \phi S_{xr^2} - \cos \phi S_{yr^2}, \\ S_{\alpha\alpha} &= \sin^2 \phi S_{xx} - 2 \sin \phi \cos \phi S_{xy} + \cos^2 \phi S_{yy}. \end{aligned} \quad (16)$$

It is too complicated to invert explicitly the matrix \mathbf{V}^{-1} defined in eqs. (15). The inversion to obtain the error matrix \mathbf{V} can be performed by a general inversion routine or by a routine dedicated for symmetric matrix inversion.

Due to the fact that we do not minimize the proper χ^2 function in eq. (7) but rather the function $\hat{\chi}^2$, (8) we make a small error in fitting the parameters ρ , ϕ , d . We have verified that the error is small, indeed, but it is of interest, however, to find a correction formula. It appears that the correction can be computed quickly and the main contribution to the computing time comes from calculating the error matrix \mathbf{V} which is usually needed anyway.

The application of Newton's method appears to be particularly simple and no extra summation is needed. The corrections (to be added) are

$$\begin{aligned} \Delta \rho &= -\frac{1}{2} \left(V_{\rho\rho} \frac{\partial \chi^2}{\partial \rho} + V_{\rho d} \frac{\partial \chi^2}{\partial d} \right), \\ \Delta \phi &= -\frac{1}{2} \left(V_{\phi\rho} \frac{\partial \chi^2}{\partial \rho} + V_{\phi d} \frac{\partial \chi^2}{\partial d} \right), \\ \Delta d &= -\frac{1}{2} \left(V_{d\rho} \frac{\partial \chi^2}{\partial \rho} + V_{dd} \frac{\partial \chi^2}{\partial d} \right), \end{aligned} \quad (17)$$

where

$$\frac{\partial \chi^2}{\partial \rho} = d\sigma; \quad \frac{\partial \chi^2}{\partial d} = \rho\sigma, \quad (18)$$

with $\sigma = -\rho S_\delta + 2u S_{\alpha\alpha} - d(1 + u) S_\alpha$. The derivatives (18) are calculated at the minimum of $\hat{\chi}^2$, see eq. (8). Therefore the derivative of χ^2 with respect to ϕ vanishes since $\partial \chi^2 / \partial \phi = (1 + \rho d)^2 \partial \hat{\chi}^2 / \partial \phi$.

5. Transformation to a new reference point

The circle fitting procedure described above yields the direction and distance parameters ϕ , d as well as the covariance matrix \mathbf{V} with respect to the origin of the coordinates x_i , y_i (the curvature is, of course, independent of the origin). In practical applications one is usually interested in the parameters with respect to a

given point. For example in particle physics a natural reference point is the particle production vertex.

In case the reference point of interest, say (x_0, y_0) , is known prior to the fit one simply makes the transformation

$$x'_i = x_i - x_0,$$

$$y'_i = y_i - y_0,$$

and performs the fit procedure as described above using the transformed coordinates. Then the resulting fit parameters and error matrix are automatically calculated with respect to the point (x_0, y_0) .

A more frequent situation, however, is the case in which the relevant reference point is not known prior to the track fitting. Normally an event reconstruction procedure is such that one fits first the particle trajectories which are then used in a vertex finding algorithm. This means that the particle production vertex will be determined only after the trajectory fitting. Often a vertex finding algorithm is an iterative procedure during which the reference point of interest (i.e. the iterated position of the production vertex) is changing repeatedly. It is therefore of great interest to have an algorithm which quickly calculates the parameters and their error matrix with respect to a new point without a necessity to perform a complete circle fit every time. In the following we describe such a transformation procedure.

Suppose the track parameters ρ , ϕ and d as well as their error matrix \mathbf{V} are calculated with respect to a point (x_0, y_0) . The problem is to find new parameters ρ' , ϕ' , d' and their error matrix \mathbf{V}' with respect to a new reference point (x'_0, y'_0) . The change of parameters is fairly straightforward. The transformation equations read:

$$\begin{cases} \rho' = \rho, \\ \phi' = \arctan \frac{B}{C}, \\ d' = \frac{A}{1+U}, \end{cases} \quad (19)$$

where

$$\begin{aligned} A &= 2\Delta_{\perp} + \rho(\Delta_{\perp}^2 + \Delta_{\parallel}^2), & U &= \sqrt{1 + \rho A}, \\ B &= \rho(x_0 - x'_0) + u \sin \phi, \\ C &= -\rho(y_0 - y'_0) + u \cos \phi, \\ \Delta_{\perp} &= (x_0 - x'_0) \sin \phi - (y_0 - y'_0) \cos \phi + d, \\ \Delta_{\parallel} &= (x_0 - x'_0) \cos \phi + (y_0 - y'_0) \sin \phi, \end{aligned}$$

and u is defined in eqs. (16). The transformation eqs. (19) are formulated so as to avoid singularity at the straight track limit $\rho \rightarrow 0$.

The error matrix \mathbf{V} transforms as:

$$\mathbf{V}' = \mathbf{J}\mathbf{V}\mathbf{J}^T, \quad (20)$$

where \mathbf{J} is the Jacobian derivative matrix of the transformation equations:

$$\mathbf{J} = \begin{pmatrix} \frac{\partial \rho'}{\partial \rho} & \frac{\partial \rho'}{\partial \phi} & \frac{\partial \rho'}{\partial d} \\ \frac{\partial \phi'}{\partial \rho} & \frac{\partial \phi'}{\partial \phi} & \frac{\partial \phi'}{\partial d} \\ \frac{\partial d'}{\partial \rho} & \frac{\partial d'}{\partial \phi} & \frac{\partial d'}{\partial d} \end{pmatrix}. \quad (21)$$

We have calculated the elements of this matrix with the following result:

$$\mathbf{J} = \begin{pmatrix} 1 & 0 & 0 \\ \xi \Delta_{\parallel} & \xi u v & -\xi \rho^2 \Delta_{\parallel} \\ \mu \zeta - \lambda A & 2\mu u \Delta_{\parallel} & 2\mu v \end{pmatrix}, \quad (22)$$

where

$$v = 1 + \rho \Delta_{\perp}, \quad \xi = (B^2 + C^2)^{-1},$$

$$\lambda = \frac{1}{2} A [(1+U)^2 U]^{-1},$$

$$\mu = [U(1+U)]^{-1} + \rho \lambda, \quad \zeta = \Delta_{\perp}^2 + \Delta_{\parallel}^2.$$

The reader can quickly verify that this transformation leaves the variance of ρ invariant as it should since ρ itself is invariant under translation of the coordinate system.

6. Discussion

We have verified the formulae introduced in the previous sections by a Fortran code. In the following we shall discuss our experience on possible difficulties arising in writing such a code. We shall also discuss timing of the algorithm as well as strategies for bad point rejection.

We mentioned that the residual expression (4) is valid for any arc length. It is also valid regardless of the origin of the coordinate system. However, when transforming to the expression (5) one divides by $1 + \rho d$ which implies that $\rho d \neq -1$. This means that the minimization formalism described above has a singularity for circles centred exactly about the origin ($\rho d = -1$). It also implies that circles centred too near the origin will suffer from truncation problems. Fortunately this is a rather rare situation for particle trajectories, because the production vertex is usually near the origin of the coordinate system. A singular or nearly singular situation, in rare cases it might happen, can be avoided by a translation of the coordinate system.

Attention must be paid to truncation problems in general when applying the method. It is recommended to calculate the elements $C_{xx} \cdots C_{r^2, r^2}$ defined in eqs. (11), as well as the relevant sums, in double precision. The truncation effects can be decreased by transforming

to a coordinate system where $\langle x \rangle = \langle y \rangle = \langle xy \rangle = 0$. The transformation involves a translation and a rotation. We have verified that also a single precision version of the fit algorithm gives good results in this case, but there is the drawback that the χ^2 value must be calculated by explicit summation. This is, because the direct formula (13) requires double precision. Therefore the fit procedure including χ^2 calculation is hardly faster in single precision than in double precision.

The residual expression (4) changes sign under the following transformation:

$$\begin{aligned} \rho &\rightarrow -\rho, \\ \phi &\rightarrow \phi + \pi, \\ d &\rightarrow -d. \end{aligned} \quad (23)$$

It follows that ϵ_r^2 and hence χ^2 are invariant under this transformation. Therefore the fit solution has, in principle, twofold ambiguity. It depends on the user which one of the two solutions to accept. For particle trajectories a natural solution is such that the direction ϕ coincides with the direction of propagation of the particle. Due to the ambiguity the fit procedure does not necessarily give the wanted solution. Suppose one knows that the correct direction is from the point 1 towards the point 2. If the test function $\cos \phi(x_2 - x_1) + \sin \phi(y_2 - y_1)$ is negative, one has to perform the transformations (23) as well as to swap the correlation terms $V_{\rho\phi} \rightarrow -V_{\rho\phi}$ and $V_{\phi d} \rightarrow -V_{\phi d}$.

We have tested the code by random generation of circles with different curvature, direction and distance parameters. Measured points were generated along circles in intervals of 1 cm. The points were Gaussian fluctuated assuming 300 μm error which is typical for drift chambers. The Monte Carlo generated circle points were then fitted with the method described above. In order to verify that the code produces correct results we plotted the distribution of the pull values $(p_{\text{fit}} - p_{\text{gen}})/\sigma_p$ where p_{gen} are the true (generated) values of the circle parameters and σ_p the computed error estimation. For all three parameters the pull values followed the normal distribution $N(0, 1)$.

In table 1 we list some typical timing numbers for different computers. There are two numbers for each entry in the table. The first number represents the computing time per point (in μs) without error estimation of the fitted parameters and the second number includes the error estimation.

The IBM3090 was run in scalar mode. For scalar

Table 1
CPU-times [μs] per point for different computers

# points	Cray-XMP	IBM3090	VAX3900	VAX8650
4	21 / 31	8.9/23	100/200	66/144
20	4.4 / 6.4	3.6/ 6.5	41/ 59	26/ 46
100	1.1 / 1.5	2.6/ 3.2	32/ 35	19/ 23
500	0.48/ 0.55	2.4/ 2.5	30/ 31	18/ 21

computing the CPU times *per fitted track* follow approximately the formulae

$$t = t_0(n + 10) \quad \text{without error estimation,}$$

$$t = t_0(n + 30) \quad \text{with error estimation,}$$

where n is the number of points and t_0 is a computer dependent constant.

A non-iterative fitting procedure provides a fast method for removing bad points or adding new ones. For the method introduced here the full fit information is contained in the nine sums $S_w, S_x \cdots S_4$ ($S_{r^2} = S_x^2 + S_y^2$) and the reference point coordinates. If new points are to be added or bad points to be removed one simply modifies the sums by adding or subtracting the contribution of the points in question. In the UA1 experiment, for example, the algorithm to remove bad points automatically during the track fit procedure is: Check the χ^2 value; if it is unacceptable remove the point with largest contribution to the χ^2 ; repeat this until the χ^2 is acceptable.

7. Summary

We present a method for fast non-iterative circle fitting. The method is an upgrade to the algorithm we have developed earlier for track fitting in the UA1 experiment and subsequently adapted also for the OPAL and L3 experiments. A very useful feature in our method is that the fitted parameters are Gaussian behaved.

The algorithm is based on a nonlinear least squares problem for which we have found an explicit solution. We describe the mathematics of the method in detail. We also present a direct and fast formula to compute the value of the χ^2 function at convergence.

We present formulae to compute the covariance matrix of the fitted parameters. We have tested the method by Monte Carlo generated circles and verified the validity of the error estimation by plotting the pull values of the fitted parameters. We give a detailed solution to the problem to transform the circle parameters and covariance matrix to a new point of reference.

Finally we discuss some practical questions such as truncation, sign ambiguity, timing and bad point rejection.

References

- [1] V. Karimäki, CERN UA1/TN-82-24, unpublished.
- [2] M. Hansroul, H. Jeremie and D. Savard, Nucl. Instr. and Meth. A270 (1988) 498.
- [3] J.C. Sens, L3-report L3*E*-810, unpublished.
- [4] J.F. Crawford, Nucl. Instr. and Meth. 211 (1983) 223.
- [5] N.I. Chernov and G.A. Ososkov, Comput. Phys. Commun. 33 (1984) 329.
- [6] D.H. Saxon, Nucl. Instr. and Meth. A234 (1985) 258.