HERA-B 98–113 Revised October 25, 2005
Software 98–018

Optimized Integration of the Equations of Motion of a Particle in the HERA-B Magnet

Alexander Spiridonov
DESY Zeuthen / ITEP Moscow
E-mail: spiridon@ifh.de

July 16, 1998

Abstract

In this note we present a flexible approach to perform the propagation of track parameters and their derivatives in an inhomogeneous magnetic field, keeping the computational effort small. We discuss also a Kalman filter implementation using this optimized computation of the derivatives.

1 Introduction

Dedicated hadronic B factory experiments as HERA-B are designed as forward spectrometers, in adjustment to the huge Lorentz boost under which the B decay particles are produced. The HERA-B tracking concept [1] relies on the propagation of track candidates which have been found in the pattern tracker in the field-free area upstream through the spectrometer magnet. The *concurrent track evolution* strategy [2] which employs the Kalman filter technique is used to cope with the large track densities and still give a high track finding efficiency. Practical implementation of this concept requires a fast and precise procedure to transport both the track parameters and its covariance matrix estimate, in spite of the inhomogeneity of the magnetic field. First solution based on a fifth-order Runge-Kutta method have been shown in [3].

This note presents the mathematical basis and the program implementation of an approach which was developed to achieve a further gain in speed, and at the same time warrant sufficient accuracy for an optimal operation of the track finding process. This was achieved by providing a set of methods which are optimized for different transport ranges, and by testing the validity of each approximation directly within the track finding application [4, 5].

The Kalman filter technique is used very often in high energy physics experiments. We include Sec. 4 in the revised version of the note to discuss the optimized implementation of the Kalman filter for magnet tracking, which reduces strongly the amount of computation. It was not described in detail in [4, 5].

2 Equations of motion and solution

In the following we will use a coordinate system in which the z axis points along the proton beam, the x axis is directed normal to it in the horizontal plane, pointing towards the inside of the HERA ring, and the y axis is oriented upwards such that x, y and z form a right-handed system. This system is identical to the *Arte* coordinate system defined in the appendix of [6].

The following choice of track parameters is suited for fixed target experiments with relatively small transverse momenta. A particle with momentum $\vec{p}$, charge Q_e and coordinates $\vec{x}$ is described by the state vector at a reference z coordinate:

$$\tilde{x}(z) = \begin{pmatrix} x \\ y \\ t_x \\ t_y \\ q \end{pmatrix} , \tag{1}$$

where x and y are the transverse coordinates, $t_x = p_x/p_z$ and $t_y = p_y/p_z$ are the track slopes, and $q = Q_e/\left|\vec{p}\right|$. The parameter Q_e is the particle charge in units of the elementary charge. We use the following units: x, y, z are in centimeters, p is in GeV/c and the magnetic field B in kGauss.

The trajectory of a particle in a static magnetic field $\vec{B}(\vec{x})$, neglecting stochastic perturbations as energy loss and multiple scattering, must satisfy the equations of motion [7] :

$$\begin{aligned} dx/dz = &\quad t_x, \\ dy/dz = &\quad t_y, \\ dt_x/dz = &\quad q \cdot \upsilon \cdot A_x(t_x, t_y, \vec{B}), \\ dt_y/dz = &\quad q \cdot \upsilon \cdot A_y(t_x, t_y, \vec{B}), \\ q = &\quad q_0, \end{aligned} \tag{2}$$

where parameter υ is proportional to the velocity of light and is therefore defined as

$$\upsilon = 0.000299792458 \; (GeV/c) \, kG^{-1} \, cm^{-1}$$

and the functions A_x,A_y are

$$\begin{aligned} A_x &= (1 + t_x^2 + t_y^2)^{\frac{1}{2}} \cdot \left[t_y \cdot (t_x B_x + B_z) - (1 + t_x^2) B_y\right] \quad , \\ A_y &= (1 + t_x^2 + t_y^2)^{\frac{1}{2}} \cdot \left[-t_x \cdot (t_y B_y + B_z) + (1 + t_y^2) B_x\right] \quad . \end{aligned}$$

The initial values at $z = z_0$ are

$$\tilde{x}_0^T = (x_0,\ y_0,\ t_{x0},\ t_{y0},\ q_0). \tag{3}$$

Let us assume we should propagate the particle parameters from plane z_0 to plane z. For solution of the equations (2) three different methods are used. The choice depends on the distance $s = z - z_o$ between these planes.

1. $|s| < 20$ cm: a parabolic expansion of the particle trajectory is used

$$\begin{aligned} x(z) = &\quad x_0 + t_{x0} \cdot s + \tfrac{1}{2} \cdot q_0 \cdot \upsilon \cdot A_x \cdot s^2 \quad , \\ y(z) = &\quad y_0 + t_{y0} \cdot s + \tfrac{1}{2} \cdot q_0 \cdot \upsilon \cdot A_y \cdot s^2 \\ t_x(z) = &\quad t_{x0} + q_0 \cdot \upsilon \cdot A_x \cdot s \quad , \\ t_y(z) = &\quad t_{y0} + q_0 \cdot \upsilon \cdot A_y \cdot s \quad . \end{aligned}$$

2. 20 cm $\leq |s| <$ 60 cm: the classical fourth-order Runge-Kutta method [8] is selected to find solution of the equations (2) .

3. $|s| \geq 60$ cm: a fifth-order Runge-Kutta method with adaptive step size control [8] is used.

Use of the Kalman filter technique [9] for pattern recognition requires that particle parameters and their covariance matrix can be transported to the location of the next measurement. Evaluation of the derivatives of the state vector components with respect to their initial values (3) is needed to transport the covariance matrix. This is achieved by integrating the equations for derivatives together with the 'zero trajectory' (2).

3 Equations for derivatives

Let us assume the magnetic field is smooth enough and field gradients can be neglected. In this case, the equations (2) are invariant with respect to small shifts by x and y. This means that derivatives with respect to initial x_0, y_0 are trivial :

$$\begin{aligned} \partial\tilde{x}^T/\partial x_0 = &\ (1,\ 0,\ 0,\ 0,\ 0) \quad , \\ \partial\tilde{x}^T/\partial y_0 = &\ (0,\ 1,\ 0,\ 0,\ 0) \quad . \end{aligned}$$

To obtain equations for $\partial\tilde{x}/\partial t_{x0}$, let us differentiate equations (2) with respect to t_{x0} and change the order of the derivative operators $\partial/\partial t_{x0}$ and d/dz on the left hand sides :

$$\begin{aligned} d/dz(\partial x/\partial t_{x0}) = &\ \partial t_x/\partial t_{x0}, \\ d/dz(\partial y/\partial t_{x0}) = &\ \partial t_y/\partial t_{x0}, \\ d/dz(\partial t_x/\partial t_{x0}) = &\ q_0 \cdot \upsilon \cdot [(\partial A_x/\partial t_x)(\partial t_x/\partial t_{x0}) + (\partial A_x/\partial t_y)(\partial t_y/\partial t_{x0})] , \\ d/dz(\partial t_y/\partial t_{x0}) = &\ q_0 \cdot \upsilon \cdot [(\partial A_y/\partial t_x)(\partial t_x/\partial t_{x0}) + (\partial A_y/\partial t_y)(\partial t_y/\partial t_{x0})] , \\ \partial q/\partial t_{x0} \ = &\ \qquad 0, \end{aligned} \tag{4}$$

where

$$\begin{aligned} \partial A_x/\partial t_x &= t_x \cdot A_x/(1+t_x^2+t_y^2) + (1+t_x^2+t_y^2)^{\frac{1}{2}} \cdot (t_y \cdot B_x - 2 \cdot t_x \cdot B_y) , \\ \partial A_x/\partial t_y &= t_y \cdot A_x/(1+t_x^2+t_y^2) + (1+t_x^2+t_y^2)^{\frac{1}{2}} \cdot (t_x \cdot B_x + B_z) , \\ \partial A_y/\partial t_x &= t_x \cdot A_y/(1+t_x^2+t_y^2) + (1+t_x^2+t_y^2)^{\frac{1}{2}} \cdot (-t_y \cdot B_y - B_z) , \\ \partial A_y/\partial t_y &= t_y \cdot A_y/(1+t_x^2+t_y^2) + (1+t_x^2+t_y^2)^{\frac{1}{2}} \cdot (-t_x \cdot B_y + 2 \cdot t_y \cdot B_x) . \end{aligned}$$

Initial values for the solution of equations (4) are :

$$\partial\tilde{x}^T/\partial t_{x0} = (0,\ 0,\ 1,\ 0,\ 0). \tag{5}$$

The equations for $\partial\tilde{x}/\partial t_{y0}$ are similar to equations (4) , but the initial values are :

$$\partial\tilde{x}^T/\partial t_{y0} = \ (0,\ 0,\ 0,\ 1,\ 0) \ .$$

To obtain equations for $\partial\tilde{x}/\partial q_0$, let us differentiate the equations (2) with respect to q_0 and change the order of the derivative operators $\partial/\partial q_0$ and d/dz in the left parts :

$$\begin{aligned} d/dz(\partial x/\partial q_0) = &\ \partial t_x/\partial q_0 \ , \\ d/dz(\partial y/\partial q_0) = &\ \partial t_y/\partial q_0 \ , \\ d/dz(\partial t_x/\partial q_0) = &\ \upsilon \cdot A_x + \upsilon \cdot q_0 \cdot [(\partial A_x/\partial t_x)(\partial t_x/\partial q_0) + (\partial A_x/\partial t_y)(\partial t_y/\partial q_0)] \ , \\ d/dz(\partial t_y/\partial q_0) = &\ \upsilon \cdot A_y + \upsilon \cdot q_0 \cdot [(\partial A_y/\partial t_x)(\partial t_x/\partial q_0) + (\partial A_y/\partial t_y)(\partial t_y/\partial q_0)] \ , \\ \partial q/\partial q_0 = &\ 1 \ . \end{aligned} \tag{6}$$

Initial values for the solution of equations (6) are :

$$\partial\tilde{x}^T/\partial q_0 = (0,\ 0,\ 0,\ 0,\ 1). \tag{7}$$

In the case of

$$|B_x/B_y|\,,|B_z/B_y|\,,|t_x|\,,|t_y| \ll\ 1$$

the following relations between A_x, A_y and their derivatives are valid:

$$\begin{array}{cl} |\partial A_{x,y}/\partial t_{x,y}| & \ll 1\ , \\ \left|\partial A_{x(y)}/\partial t_{x,y}\right| & \ll \left|A_{x(y)}\right|\ , \\ |A_y| & \ll |A_x|\ . \end{array}$$

The procedure of evaluating the derivatives is simplified when these relations are taken into account.

Approximation A: The vector $\partial\tilde{x}/\partial t_{x0}$ can be approximated as :

$$\partial\tilde{x}^T/\partial t_{x0} = \ \ (\partial x/\partial t_{x0},\ \partial y/\partial t_{x0},\ 1,\ \partial t_y/\partial t_{x0},\ 0)\ \ ,$$

where the derivatives are solutions of the system of equations

$$\begin{array}{ll} d/dz(\partial x/\partial t_{x0}) = & \partial t_x/\partial t_{x0}\ , \\ d/dz(\partial y/\partial t_{x0}) = & \partial t_y/\partial t_{x0}\ , \\ d/dz(\partial t_x/\partial t_{x0}) = & 0\ , \\ d/dz(\partial t_y/\partial t_{x0}) = & q_0 \cdot \upsilon \cdot [(\partial A_y/\partial t_x)(\partial t_x/\partial t_{x0}) + (\partial A_y/\partial t_y)(\partial t_y/\partial t_{x0})]\ , \\ \partial q/\partial t_{x0}\ \ = & 0 \end{array}$$

with initial values (5). The solution for the derivatives $\partial\tilde{x}/\partial t_{y0}$ in the same approximation is similar :

$$\partial\tilde{x}^T/\partial t_{y0} = \ \ (\partial x/\partial t_{y0},\ \partial y/\partial t_{y0},\ \partial t_x/\partial t_{y0},\ 1,\ 0)\ \ .$$

The derivatives

$$\partial\tilde{x}^T/\partial q_0 = \ \ (\partial x/\partial q_0,\ \partial y/\partial q_0,\ \partial t_x/\partial q_0,\ \partial t_y/\partial q_0,\ 1)$$

are solutions of the equations (6) with the initial values given in (7).

Approximation B: This is the most drastic simplification - the derivatives $\partial A_{x,y}/\partial t_{x,y}$ as well as A_y are neglected in the corresponding equations. The system of equations (4) is simplified to :

$$\begin{array}{ll} d/dz(\partial x/\partial t_{x0}) = & \partial t_x/\partial t_{x0}\ , \\ d/dz(\partial y/\partial t_{x0}) = & \partial t_y/\partial t_{x0}\ , \\ d/dz(\partial t_x/\partial t_{x0}) = & 0\ , \\ d/dz(\partial t_y/\partial t_{x0}) = & 0\ , \\ \partial q/\partial t_{x0}\ \ = & 0\ . \end{array}$$

The solution of this system with initial values (5) is

$$\partial\tilde{x}^T/\partial t_{x0} = \ \ (s,\ 0,\ 1,\ 0,\ 0)\ \ .$$

The solution of similar equations for derivatives $\partial/\partial t_{y0}$ is

$$\partial\tilde{x}^T/\partial t_{y0} = \ (0,\ s,\ 0,\ 1,\ 0)\ .$$

For the vector of derivatives $\partial/\partial q_0$ we obtain :

$$\partial\tilde{x}^T/\partial q_0 = \ (\partial x/\partial q_0,\ 0,\ \partial t_x/\partial q_0,\ 0,\ 1)\ ,$$

where the derivatives are solutions of the system of equations

$$\begin{aligned} d/dz(\partial x/\partial q_0) &= \partial t_x/\partial q_0, \\ d/dz(\partial t_x/\partial q_0) &= \upsilon \cdot A_x, \end{aligned} \tag{8}$$

with initial values from (7).

4 Application of the Kalman filter technique for the magnet tracking

We use the notations from [9] to describe the practical implementation of the Kalman filter technique for track following in the magnet. The system state vector (1) after inclusion of k measurements is denoted by $\tilde{x}_k$, and its covariance matrix by C_k. The coordinate measurement corresponding to the hit k is denoted by m_k. The HERA-B magnet tracking detectors (drift tubes and Micro-Strip Gaseous Chambers) measured only one coordinate and m_k is a scalar and its covariance matrix V_k contains only one element. The relation between the track parameters $\tilde{x}_k$ and the expected measurement m_k is described by the projection matrix H_k. The matrix H_k has the structure:

$$H^{(1)} = (h_1,\ 0,\ 0,\ 0,\ 0)\,, \tag{9}$$

for a detector plane measuring only x coordinates (signal wires of the drift tubes or anode strips of the MSGC are parallel to the y axis), and

$$H^{(2)} = (h_1,\ h_2,\ 0,\ 0,\ 0)\,, \tag{10}$$

for stereo planes rotated around the z axis. The predicted state vector $\tilde{x}_k^{k-1}$ is determined as the solution of the equations (2) with the initial value $\tilde{x}_{k-1}$. The covariance matrix of the vector $\tilde{x}_k^{k-1}$ is obtained by the propagation of the matrix C_{k-1}:

$$C_k^{k-1} = F_k C_{k-1} F_k^T + Q_k, \tag{11}$$

where Q_k denotes the covariance matrix of the process noise and the transport matrix is

$$F_k = \frac{\partial(\tilde{x}_k^{k-1})}{\partial(\tilde{x}_{k-1})}. \tag{12}$$

The estimated residual and variance become

$$\begin{aligned} r_k^{k-1} &= m_k - H_k\,\tilde{x}_k^{k-1}\,, \\ R_k^{k-1} &= V_k + H_k\,C_k^{k-1}\,H_k^T\,. \end{aligned} \tag{13}$$

The updating of the system state vector after inclusion of the measurement k is obtained with the filter equations:

$$\begin{aligned} K_k &= C_k^{k-1}\, H^T\, (R_k^{k-1})^{-1}, \\ \tilde{x}_k &= \tilde{x}_k^{k-1} + K_k\, r_k^{k-1}, \\ C_k &= (1 - K_k\, H_k)\, C_k^{k-1}, \end{aligned} \tag{14}$$

with the filtered residuals

$$\begin{aligned} r_k &= (1 - H_k\, K_k)\, r_k^{k-1}, \\ R_k &= (1 - H_k\, K_k)\, V_k. \end{aligned} \tag{15}$$

The χ^2 contribution of the filtered point is given by:

$$\chi_k^2 = r_k^T\, R_k^{-1}\, r_k.$$

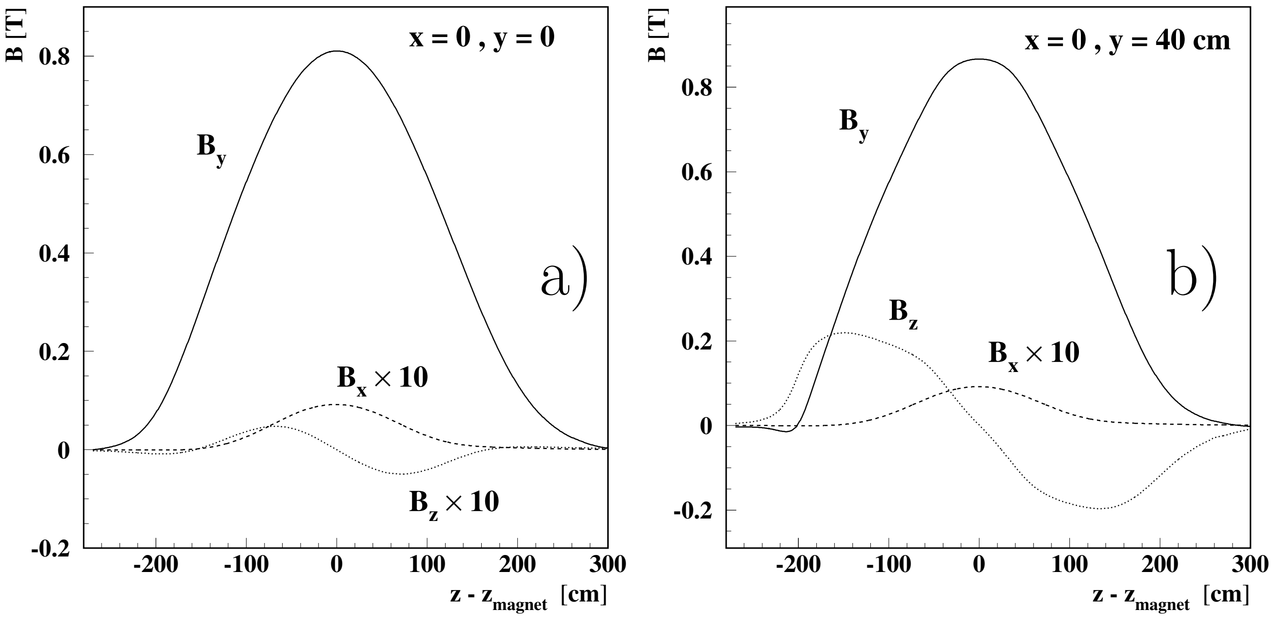

Figure 1: Magnetic field components as a function of the z coordinate in the magnet center (a) and at a vertical displacement of 40 cm (b). The z coordinate is given relative to the magnet center at $z_{magnet} = 450\,\text{cm}$.

Because of sparse projection matrices (9,10), the calculation in (13–15) becomes rather simple. For the matrix $H^{(2)}$ the variance of estimated residuals is

$$\{R_k^{k-1}\}_{11} = \{V_k\}_{11} + h_1\,(h_1\, C_{11} + h_2\, C_{21}) + h_2\,(h_1\, C_{12} + h_2\, C_{22}),$$

i.e. it involves 6 multiplications. Here we use the notation $C_{ij} = \{C_k^{k-1}\}_{ij}$. For the same case, each of the five elements of the gain matrix is calculated as

$$\{K\}_{i1} = k_i = (C_{i1}\, h_1 + C_{i2}\, h_2)\ \{R_k^{k-1}\}_{11}^{-1}.$$

The calculation of the gain matrix includes 15 multiplications and 1 division. The most time consuming operations are the propagation of the covariance matrix in (11) and the calculation of the covariance matrix in the filter equations (14)

A typical behavior of the magnetic field components [10] as a function of z is shown in fig. 1. The main bending component (B_y) has a bell-shaped z–dependence. The components B_x, B_z are sizeable away from the central axis (fig. 1b). There is clearly no region inside the magnet where the field can be regarded as homogeneous.

"Numerical experiments" with the real track finding procedures have shown that the approximation B for derivatives is accurate enough to be used for the propagation of the covariance matrix (11) in the case of the inhomogeneous magnetic field of HERA-B. In approximation B the transport matrix has a rather sparse structure:

$$F_k = \begin{pmatrix} 1 & 0 & s_k & 0 & x'_k \\ 0 & 1 & 0 & s_k & 0 \\ 0 & 0 & 1 & 0 & t'_k \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \tag{16}$$

where $s_k = z_k - z_{k-1}$, and derivatives are denoted $x'_k = \partial x_k^{k-1}/\partial q_{k-1}$ and $t'_k = \partial t_{xk}^{k-1}/\partial q_{k-1}$

For a short distance ($|s| < 20$ cm) the derivatives are obtained in a parabolic expansion as:

$$\partial x_k^{k-1}/\partial q_0 = \frac{1}{2}\,\upsilon\, A_x\, s_k^2 \quad \text{and} \quad \partial t_{xk}^{k-1}/\partial q_0 = \upsilon\, A_x\, s_k\,.$$

For a long distance we find the derivatives as the solution of the system (8) (initial values from (7)), together with the "zero trajectory" (2), by the fourth or fifth-order Runge-Kutta method [8].

The propagation of the covariance matrix $F_k C_{k-1} F_k^T$ in (11) we perform in two steps. First, we define the product of two matrices $U = F_k C_{k-1}$:

$$\begin{aligned}
u_{11} &= c_{11} + c_{13}\, s_k + c_{15}\, x'_k, \\
u_{12} &= c_{12} + c_{23}\, s_k + c_{25}\, x'_k, \\
u_{13} &= c_{13} + c_{33}\, s_k + c_{35}\, x'_k, \\
u_{14} &= c_{14} + c_{34}\, s_k + c_{45}\, x'_k, \\
u_{15} &= c_{15} + c_{35}\, s_k + c_{55}\, x'_k, \\
u_{21} &= c_{12} + c_{14}\, s_k, \\
u_{22} &= c_{22} + c_{24}\, s_k, \\
u_{23} &= c_{23} + c_{34}\, s_k, \\
u_{24} &= c_{24} + c_{44}\, s_k, \\
u_{25} &= c_{25} + c_{45}\, s_k, \\
u_{31} &= c_{13} + c_{15}\, t'_k, \\
u_{32} &= c_{23} + c_{25}\, t'_k, \\
u_{33} &= c_{33} + c_{35}\, t'_k, \\
u_{34} &= c_{34} + c_{45}\, t'_k, \\
u_{35} &= c_{35} + c_{55}\, t'_k, \\
u_{4i} &= c_{4i} \qquad\qquad\qquad (i = 1 \div 5), \\
u_{5i} &= c_{5i} \qquad\qquad\qquad (i = 1 \div 5),
\end{aligned} \tag{17}$$

where $c_{ij} = \{C_{k-1}\}_{ij}$ and $u_{ij} = \{U\}_{ij}$. Then the matrix U is multiplied by F_k^T to obtain the final symmetric matrix $C' = F_k C_{k-1} F_k^T$:

$$
\begin{aligned}
C'_{11} &= u_{11} + u_{13}\, s_k + u_{15}\, x'_k, \\
C'_{21} &= u_{21} + u_{23}\, s_k + u_{25}\, x'_k, \\
C'_{31} &= u_{31} + u_{33}\, s_k + u_{35}\, x'_k, \\
C'_{41} &= u_{41} + u_{43}\, s_k + u_{45}\, x'_k, \\
C'_{51} &= u_{51} + u_{53}\, s_k + u_{55}\, x'_k, \\
C'_{22} &= u_{22} + u_{24}\, s_k, \\
C'_{32} &= u_{32} + u_{34}\, s_k, \\
C'_{42} &= u_{42} + u_{44}\, s_k, \\
C'_{52} &= u_{52} + u_{54}\, s_k, \\
C'_{33} &= u_{33} + u_{35}\, t'_k, \\
C'_{43} &= u_{43} + u_{45}\, t'_k, \\
C'_{53} &= u_{53} + u_{55}\, t'_k, \\
C'_{44} &= u_{44}, \\
C'_{54} &= u_{54}, \\
C'_{55} &= u_{55}.
\end{aligned}
\tag{18}
$$

The evaluation of the elements in (17) and (18) implies 37 multiplications, that is much smaller than the 200 multiplications needed for the case of the completely filled matrix F_k.

The product of the matrices K_k and $H_k^{(1)}$ in (14) is

$$
K_k\, H_k^{(1)} = \begin{pmatrix}
k_1\, h_1 & 0 & 0 & 0 & 0 \\
k_2\, h_1 & 0 & 0 & 0 & 0 \\
k_3\, h_1 & 0 & 0 & 0 & 0 \\
k_4\, h_1 & 0 & 0 & 0 & 0 \\
k_5\, h_1 & 0 & 0 & 0 & 0
\end{pmatrix},
\tag{19}
$$

and for the matrix $H^{(2)}$

$$
K_k\, H_k^{(2)} = \begin{pmatrix}
k_1\, h_1 & k_1\, h_2 & 0 & 0 & 0 \\
k_2\, h_1 & k_2\, h_2 & 0 & 0 & 0 \\
k_3\, h_1 & k_3\, h_2 & 0 & 0 & 0 \\
k_4\, h_1 & k_4\, h_2 & 0 & 0 & 0 \\
k_5\, h_1 & k_5\, h_2 & 0 & 0 & 0
\end{pmatrix}.
\tag{20}
$$

For the case with the matrix $H^{(2)}$ the covariance matrix C_k is given by:

$$\begin{aligned}
\{C_k\}_{11} = &\ C_{11} - k_1\, h_1\, C_{11} - k_1\, h_2\, C_{21}, \\
\{C_k\}_{12} = &\ C_{12} - k_1\, h_1\, C_{12} - k_1\, h_2\, C_{22}, \\
\{C_k\}_{13} = &\ C_{13} - k_1\, h_1\, C_{13} - k_1\, h_2\, C_{23}, \\
\{C_k\}_{14} = &\ C_{14} - k_1\, h_1\, C_{14} - k_1\, h_2\, C_{24}, \\
\{C_k\}_{15} = &\ C_{15} - k_1\, h_1\, C_{15} - k_1\, h_2\, C_{25}, \\
\{C_k\}_{22} = &\ C_{22} - k_2\, h_1\, C_{12} - k_2\, h_2\, C_{22}, \\
\{C_k\}_{23} = &\ C_{23} - k_2\, h_1\, C_{13} - k_2\, h_2\, C_{23}, \\
\{C_k\}_{24} = &\ C_{24} - k_2\, h_1\, C_{14} - k_2\, h_2\, C_{24}, \\
\{C_k\}_{25} = &\ C_{25} - k_2\, h_1\, C_{15} - k_2\, h_2\, C_{25}, \\
\{C_k\}_{33} = &\ C_{33} - k_3\, h_1\, C_{13} - k_3\, h_2\, C_{23}, \\
\{C_k\}_{34} = &\ C_{34} - k_3\, h_1\, C_{14} - k_3\, h_2\, C_{24}, \\
\{C_k\}_{35} = &\ C_{35} - k_3\, h_1\, C_{15} - k_3\, h_2\, C_{25}, \\
\{C_k\}_{44} = &\ C_{44} - k_4\, h_1\, C_{14} - k_4\, h_2\, C_{24}, \\
\{C_k\}_{45} = &\ C_{45} - k_4\, h_1\, C_{15} - k_4\, h_2\, C_{25}, \\
\{C_k\}_{55} = &\ C_{55} - k_5\, h_1\, C_{15} - k_5\, h_2\, C_{25}.
\end{aligned} \tag{21}$$

Only 40 multiplications are sufficient to obtain the matrix C_k in this case and 20 multiplications for the option with the matrix $H^{(1)}$. This has to be compared with 100 multiplications needed for the completely filled matrix H. The optimized matrix evaluation was implemented in functions of the *ranger* package related to the magnet tracking [4].

5 Program implementation

The described approach for optimized integration of the equations 2 and their derivatives was implemented as a set of C++ functions included in the *ranger* package and used for the magnet tracking [4] and the track fit [11, 12]. All functions are of type `void`. In the following , we list the definition of common parameters of these functions in C++:

```
//
//            INPUT PARAMETERS
//
  double z_in ;    // z value for input parameters
  double p_in[5];  // vector of input track parameter (x,y,tx,ty,Q/p)
  double c_in[25]; // covariance matrix of input parameters
  float  error[2]; // desired accuracy in cm
                        // error[0] for Inner Tracker region
                        // error[1] for Outer Tracker region
//
//            OUTPUT PARAMETERS
//
  double z_out;    // z value for output parameters
  double p_out[5]; // vector of output track parameters
  double rkd[25];  // derivatives of output parameters with respect
                   // to input
                   // rkd[0] deriv. of p_in[0] with respect to p_out[0]
```

```
                    // rkd[1]           p_in[0]                 p_out[1]
                    //                    . . .
                    // rkd[5]           p_in[1]                 p_out[0]
                    //                    . . .
                    // rkd[24]          p_in[4]                 p_out[4]

  double c_out[25];// covariance matrix of output parameters
  int    ierror;   // error flag ( = 0 ok, = 1 particle curls)
```

The definition of the corresponding variables in FORTRAN is :

```
c
c           INPUT PARAMETERS
c
      real*8 z_in       ! z value for input parameters
      real*8 p_in(5)    ! vector of input parameters (x,y,tx,ty,Q/p)
      real*8 c_in(5,5)  ! covariance matrix of input parameters
      real  error(2)    ! desired accuracy in cm
c                           error(1) for Inner Tracker region
c                           error(2) for Outer Tracker region
c
c           OUTPUT PARAMETERS
c
      real*8 z_out      ! z value for output parameters
      real*8 p_out(5)   ! vector of output track parameters
      real*8 rkd(5,5)   ! rkd(i,j) derivative of p_in(i) with respect
c                         to p_out(j)
      real*8 c_out(5,5) ! covariance matrix of output parameters
      integer ierror    ! error flag ( = 0 ok , = 1 particle curls)
```

A typical function call in C++

```
  rk4order_(double& z_in, double* p_in,double& z_out, double* p_out,
            double* rkd, int& ierror);
```

invokes function integrating equations for particle parameters and equations for derivatives in the approximation A by a fourth-order Runge-Kutta method .

In FORTRAN this function can be invoked like `subroutine` :

```
      call rk4order(z_in, p_in, z_out, p_out, rkd, ierror)
```

The differences are evident and in following, the call statements in FORTRAN will be mentioned only.

A statement

```
      call rk4fast(z_in, p_in, z_out, p_out, rkd, ierror)
```

effects integration of the equations for the ‘zero trajectory’ and calculation of derivatives in the approximation B by a fourth-order Runge-Kutta method .

The function called as :

```
call rk1fast(z_in, p_in, z_out, p_out, rkd, ierror)
```

calculates parameters and derivatives in approximation B using a parabolic expansion of the particle trajectory.

The function evaluating particle parameters (approximation A for derivatives) by a fifth-order Runge-Kutta method with adaptive step size control is invoked by the statement :

```
call rk5order(z_in, p_in, error, z_out, p_out, rkd, ierror)
```

The corresponding function which evaluates the derivatives in approximation B is executed by :

```
call rk5fast(z_in, p_in, error, z_out, p_out, rkd, ierror)
```

The function invoked as:

```
call rk5numde(z_in, p_in, error, z_out, p_out, rkd, ierror)
```

evaluates the output parameters by a fifth-order Runge-Kutta method with adaptive step size control and calculates derivatives by 'numerical differentiation' of the output parameters as a function of the input parameters. The field gradients are not neglected in this case but the function spends by factor of 5 more computing time than `rk5fast`.

The "fully automatic" function `rktrans` transports particle parameters from plane z_0 to plane z by means of three different methods depending on $h = z - z_0$ as it was described earlier. The derivatives are calculated in approximation B. The call statement for this function is :

```
call rktrans(z_in, p_in, z_out, p_out, rkd, ierror)
```

The function `rktransc` uses a similar approach to transport particle parameters and the corresponding covariance matrix . The structure of the derivative matrix in approximation B is fully exploited for maximum speed. The function can be invoked by a statement

```
call rktransc(z_in, p_in, c_in, z_out, p_out, c_out, ierror)
```

All described functions were designed for the conditions of the complete geometry of the HERA-B detector where typical transport distances are $50 - 100$ cm. The limitations of tracking precision coming from multiple scattering and measurement errors in trackers constrain the tracing accuracy required for such distances.

The function `rk5clip` can be used to propagate particle parameters over larger distances with higher accuracy (approximation A for derivatives). The call statement for this function is

```
call rk5clip(z_in, p_in, z_out, p_out, rkd, ierror)
```

p_0 GeV/c	$rms(x)$ μm	$rms(y)$ μm
5	12	13
10	9	6
30	9	6
60	7	6
90	7	3

Table 1: The computational accuracy of `rk5clip` for different momenta

The following procedure was used for the tuning of the steering parameters. Particle parameters were generated with production slopes $t_{x,y}$ uniformly distributed from -100 mrad to 100 mrad. Each particle was traced by small steps from the target to the area behind the magnet ($z = 700$ cm) and then traced back using `rk5clip`. The difference between initial and final particle coordinates was regarded as a measure of computational accuracy. The accuracy for selected steering parameters is shown in the table. As expected, the function `rk5clip` spends more computing time especially for low momentum (the dependence is roughly $\propto 1/p_0$) . It should be noted that all described functions do not check if the magnetic field is defined in the region where the particle should be traced. The user himself must make sure that a corresponding function is invoked within the magnetic field and should use linear line propagation in the field-free case. Also, an attempt to trace a very slow particle will not be successful because the equations (1) do not describe a particle curling in the magnetic field.

An additional function, called as :

```
real zmin,zmax   !  in cm
   . . .
call rkzfield(zmin,zmax)
```

returns as output the lower and upper bounds `zmin, zmax` (with respect to the center of the magnet) of the region where the field is defined by routines `gufld, utfeld` [10].

This region (roughly ± 5 m from the center of the magnet) includes the area where we have field measurements or at least results from the MAFIA calculation. At the edges of the region the magnetic field can be neglected so that simple linear line searches and fits are sufficient for pattern recognition and track fitting [13]. This can save computing time for event generation and reconstruction. Therefore during the event simulation in HBGEAN [14] the procedure for particle tracing in the magnetic field is invoked only when the particle is within the GEANT volume `MAGN` . The definition of this volume can be found in the *Arte* table `GESL` for the detector component `Magnet`. Note that this volume is not identical with the region of non-zero field as defined by `gufld, utfeld`.

Acknowledgments

I am greatly indebted to Rainer Mankel, the author of the *ranger* package, for sharing with me his experience in the field of track finding and knowledge about *ranger*. I would like to thank him for the careful reading of the manuscript and the useful advices. I am indebted to G. Bohm and H. Kolanoski for the discussion of the revised version. I am very much grateful to DESY Zeuthen for the kind hospitality during my visit.

References

[1] R. Mankel, *Online track reconstruction for HERA-B*, Nucl. Instr. and Meth. A 384 (1996) 201.

[2] R. Mankel, *A Concurrent Track Evolution Algorithm for Pattern Recognition in the HERA-B Main Tracking System*, Nucl. Instr. and Meth. A 395 (1997) 169.

[3] T. Oest, *Particle Tracing through the HERA-B Magnetic Field* , HERA-B note 97-165.

T. Oest, *Fast Particle Tracing through the HERA-B Magnetic Field*, HERA-B note 98-001.

[4] R. Mankel and A. Spiridonov, *ranger – a Pattern Recognition Algorithm for the HERA-B Main Tracking System, Part II: The HERA-B Magnet Tracker*, HERA-B note 98-154 (1998).

[5] R. Mankel and A. Spiridonov, *The Concurrent Track Evolution Algorithm: Extension for Track Finding in the Inhomogeneous Magnetic Field of the HERA-B Spectrometer*, Nucl. Instr. Methods, **A426** (1999) 268; hep-ex/9809021.

[6] R. Mankel, HERA-B note 98-079 (1998)

[7] R.K. Bock, H. Grote, D. Notz and M. Regler, (edited by M. Regler), *Data analysis techniques for high-energy physics experiments,* Cambridge University Press , Cambridge (1990).

[8] W. Press, S. Teukolsky, W. Vetterling and B. Flannery, *Numerical Recipes in C,* Cambridge University Press, Cambridge (1992).

[9] R. Mankel, *Application of the Kalman Filter Technique in the HERA-B Track Reconstruction,* HERA-B note 95-239 (1995).

[10] A. Spiridonov, *A Model of the HERA-B Magnetic Field,* HERA-B note 02-072 (2002).

[11] R. Mankel, *ranger – a Pattern Recognition Algorithm for the HERA-B Main Tracking System, Part IV: The Object–Oriented Track Fit*, HERA-B note 98-079 (1998).

[12] A. Spiridonov, *Momentum and Angular Resolutions in the HERA-B Detector*, HERA-B note 02-069 (2002).

[13] T. Lohse, *Tracking Through Magnetic Field Maps*, HERA-B note 94-028 (1994).

[14] S. Nowak, *HBGEAN and HBRCAN*, HERA-B Internal Note (1995).