



university of
 groningen

faculty of science
and engineering

Bachelor's Thesis Mathematics & Physics

Using Taylor series for fast and precise charged particle tracking in the LHCb magnet

by

Maurice Dekker

Supervisors:

Dr. K. Efstathiou

Dr.ir. C.J.G. Onderwater

July 2018

Abstract

The LHCb detector is an elementary particle detector in which the momentum and charge of particles are determined from small deflections of these particles in a magnetic field. In this thesis, a Taylor series expansion is derived to determine the trajectory of a charged particle in a magnetic field. A rather unusual set of only three expansion variables is used: the angles x'_i , y'_i and a third variable, x_3 , which is the inverse of the bending radius at some reference point. This method has the great advantage that it can be programmed on massively parallel architectures, contrary to the common highly sequential numerical integration methods. The method is worked out and put into a Mathematica code, and some examples are given.

Contents

1	Introduction	1
2	Background theory	4
2.1	Electrodynamics	4
2.1.1	Magnetic vector potential	4
2.1.2	Magnetic rigidity and x_3	5
2.1.3	Shape function of the magnetic field	5
2.2	Equations of motion	5
2.3	Taylor series expansion	6
2.4	Relativistic effects	8
3	The algebraic map	10
3.1	Taylor expansion of the trajectory	10
3.2	Taylor expanding the magnetic field	12
3.3	Putting it all together	13
3.4	Result in second order	14
3.5	Example: Constant magnetic field	15
4	Determination of higher orders	18
4.1	Mathematica code to determine higher orders	18
4.1.1	Taylor expansion of the magnetic field	18
4.1.2	Taylor expansion of the trajectory	19
4.1.3	Comparing coefficients	20
4.2	Example: Constant magnetic field	21
4.3	Another example	21
5	Conclusion	25
A	Euler method	27
B	Mathematica code for determining coefficients	29
	Bibliography	31

Chapter 1

Introduction

The Standard Model of particle physics is a theory that describes the fundamental particles and how they interact under three of the four fundamental forces (gravity is not included). The Standard Model has proven to be a successful framework in providing experimental predictions. For instance, several properties of weak neutral currents and the W and Z bosons have been predicted with high accuracy. Over time and through many experiments, the Standard Model has become a well-tested physics theory.

Despite all the successes the Standard Model has had and the fact that it is believed to be theoretically self-consistent, physicists are still looking for a more complete theory of subatomic particles and their interactions. For example, the Standard Model predicts neutrino's to be massless. But evidence has been found for neutrino oscillations, which would imply that neutrino's do have mass. Also, the Standard Model cannot give a good explanation for the matter-antimatter asymmetry. And there are more examples of such unanswered problems. So the Standard Model needs to be extended, i.e. new physics (physics beyond the Standard Model) has to be found.

The Large Hadron Collider (LHC) near Geneva is one of the places where physicists are looking for new physics. It is the largest and highest-energy particle accelerator in the world. In the LHC, protons are accelerated in both directions of the 26.7 kilometer long (circular) tunnel. When the experiment is running, these protons are boosted up to an energy of almost 7 TeV, which is equivalent to velocities of only a fraction lower than the speed of light. Collisions between these protons result in showers of hundreds of new particles.

Several detectors are placed along the LHC tube. Each of these detectors focuses on different aspects of the collisions. One of these detectors is the LHCb detector, which is used to distinguish the particles that are created in high energy proton-proton collisions. These collisions are studied to measure parameters of CP-violation in interactions between heavy quarks called bottom quarks. The setup of the LHCb detector is shown in figure 1.1. This setup consists of many different types of detectors, but what we are interested in is the dipole magnet. Because when a charged particle travels through a magnetic field, it gets deflected. This principle can be used to reconstruct the trajectories of charged particles and to measure their momenta. There is a detector placed in front of the magnet (TT) and after it (T1-T3). The idea is that we want to connect the particles that are measured in front of the magnet to the particles that are measured after the magnet. This would be an easy task if we had a homogeneous field, because then the trajectories would just be circles. But unfortunately, this is not the case.

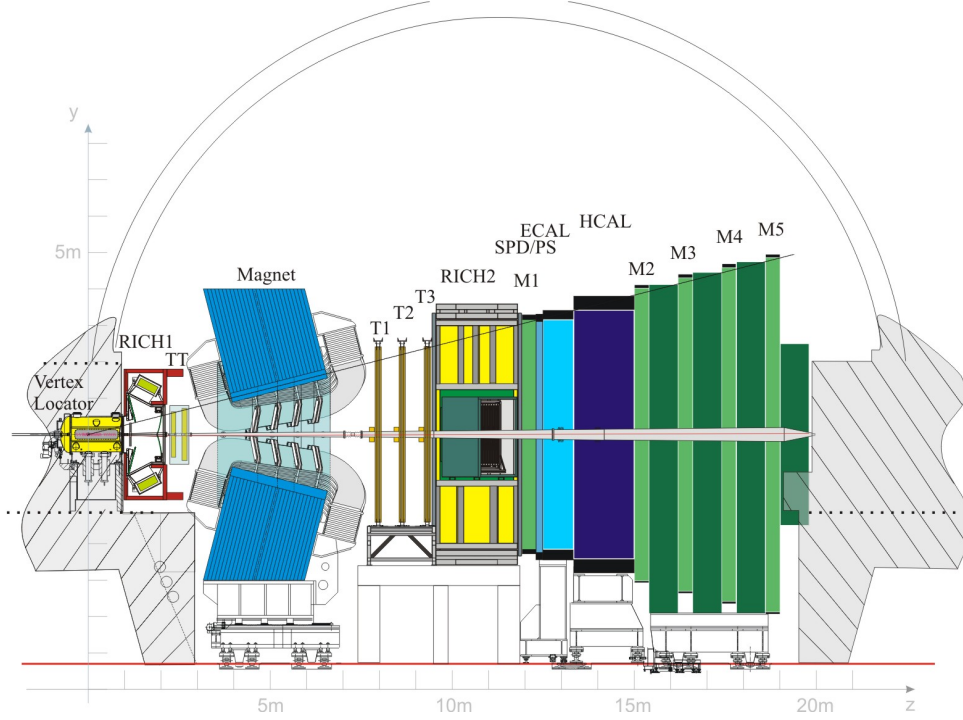


FIGURE 1.1: The LHCb detector. [1]

The LHCb experiment uses a wide aperture magnet, resulting in a highly non-homogeneous field. This field has an integrated bending power of 4 Tm over a track length of 10 meters [2]. The variation of the vertical magnetic field component B_y along the central track is shown in figure 1.2. To get a slight feel of what the magnetic field looks like, all three field components are shown for a track with an angle of 197 mrad in both x and y .

The tracking detectors are placed mainly outside of the magnetic field. Because the particles that are analysed have such high velocities, their deflection is only minor. Hence, high precision is very important when tracking particles in the LHCb magnet. Also, when the experiment is running there are 40 million collisions per second [3], i.e. there is a collision every 25 nanoseconds. In these 25 nanoseconds the complete analysis of the collision needs to be done. The more time is spent on particle tracking, the less time is left for other things. So it is of the essence that the tracking routine is very fast as well.

Different techniques can be used to track charged particles through a magnetic field. In general all these methods try to combine high precision with high speed. Standard techniques for particle tracking through magnets use numerical integration, such as Runge-Kutta, to integrate the equations of motion [4–6]. A major disadvantage of these techniques is their highly sequential behaviour, prohibiting the possibility of computing trajectories in a parallel manner. Since the LHCb experiment needs high precision when tracking particles, using numerical integration requires a very small step size. Note that in one collision hundreds of particles are created. Tracking each particle using numerical integration makes these methods very time-consuming.

Lesser used techniques introduce Taylor series expansions to derive a transfer map \mathcal{M} [7–9] that relates the final coordinates \vec{z}_f to the initial coordinates \vec{z}_i and parameters $\vec{\delta}$,

$$\vec{z}_f = \mathcal{M}(\vec{z}_i, \vec{\delta}).$$

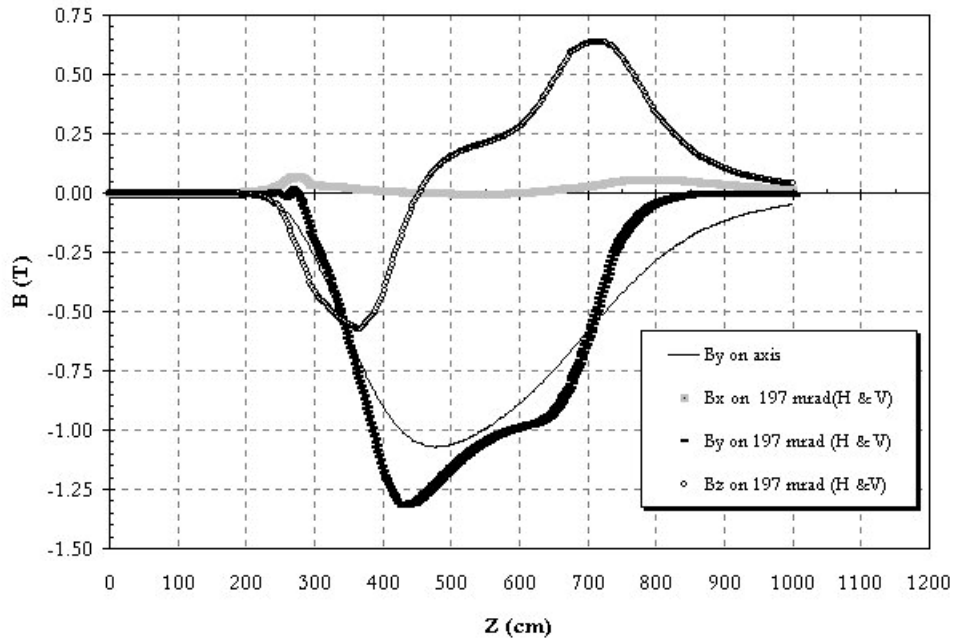


FIGURE 1.2: Magnetic field along selected tracks. [2]

Since in the LHCb experiment the magnetic field is not altered for long periods, the Taylor coefficients can be computed on forehand and be reused for the determination of each individual trajectory. In this way the coefficients can be computed to very high precision without slowing down the tracking routine.

The aim of this bachelor project is to **create a Taylor expanded map for particle motion in arbitrary magnetic fields**. Such a tracking routine can be programmed on massively parallel architectures, which will drastically increase the computation speed. A method presented by J. Bahrdt and G. Wüstefeld is followed [9]. This method uses a rather unusual set of only three expansion parameters: the transverse angle variables (x'_i, y'_i) at the starting point, and the inverse of the bending radius x_3 at some reference point. It has been shown that this results in a fast converging series for large bending radii [9, 10].

The next chapter will give some background theory regarding magnetic fields, equations of motion, Taylor expansions and relativity. In the third chapter we will work out the proposed technique in second order and derive a function that maps the initial coordinates into the final coordinates. We conclude this chapter with a simple example to show that the obtained function is already very accurate at second order. Chapter 4 presents a Mathematica code that can be used to compute the Taylor expanded map up to arbitrary order. Some examples for higher order approximations will be given.

Chapter 2

Background theory

We start this thesis by introducing some background theory that is relevant for the upcoming chapters. The topics of electrodynamics, Taylor series and a little relativity will be discussed briefly.

2.1 Electrodynamics

In this section the basics of magnetism is discussed. A few equations are introduced and it is explained how they can be used. Also, the expansion variable x_3 and shape function of the magnetic field are introduced and an explanation is given of why they are useful.

2.1.1 Magnetic vector potential

Gauss' law for magnetism states that the divergence of any magnetic field is zero, i.e. $\nabla \cdot \vec{B} = 0$. This implies that there exists a vector potential \vec{A} that satisfies

$$\vec{B} = \nabla \times \vec{A}. \quad (2.1)$$

Many methods use this vector potential for calculations instead of the magnetic field. This method is no exception. This is because the vector potential is much simpler to use in computations, as the magnetic field involves taking the cross product of the velocity vector with a position vector. Once the magnetic vector potential has been determined, the magnetic field can be found by simply taking the curl of the vector potential. This is easier than solving directly for the magnetic field.

Although this works well for theory, in practice only the magnetic field is known (the vector potential cannot be measured directly). Equation (2.1) needs to be used to determine the magnetic vector potential. It is important to note that the vector potential has a gauge invariance, meaning that it cannot be determined uniquely from the magnetic field. The gradient of any scalar function ϕ can be added to the vector potential without changing the magnetic field,

$$\nabla \times \vec{A} = \nabla \times (\vec{A} + \nabla\phi),$$

since the curl of a gradient is zero ($\nabla \times \nabla\phi = 0$).

2.1.2 Magnetic rigidity and x_3

For a particle in a uniform magnetic field \vec{B} describing a circular orbit of radius ρ , the Lorentz force is equal to the centrifugal force:

$$qvB = \frac{mv^2}{\rho},$$

with q, m, v the charge, mass and velocity of the particle. Introducing the momentum $p = mv$ and simplifying the equation, we get

$$B\rho = \frac{p}{q}. \quad (2.2)$$

This product of the field and bending radius is called the magnetic rigidity and has the unit T·m. It describes the magnetic bending strength that is required for a given particle to have a bending radius ρ . It can be useful in calculations because only the momentum and charge are needed.

As stated before, the third variable that will be used in the coordinate expansion is x_3 , the inverse of the bending radius at some reference point (x_0, y_0, z_0) :

$$x_3 = \frac{q}{p} \cdot B(x_0, y_0, z_0).$$

Expansion in this variable is rather unusual, and (as we will find out) will lead to some ambiguity. A nice feature of x_3 as an expansion variable is that it unites the charge, momentum and magnetic field into one variable. According to Bahrdt and Wüstefeld this results in fast convergence of the Taylor series [9].

2.1.3 Shape function of the magnetic field

One of the reasons that make it easier to use only three expansion variables, is the introduction of the shape function \vec{R} . This function is independent of the field strength and purely describes the geometric shape of the magnetic vector potential. The shape function is defined as follows:

$$x_3 \cdot \vec{R} = \frac{\vec{A}}{B\rho}. \quad (2.3)$$

When defining x_3 as the inverse of the bending radius ρ , we have $\vec{R} = \vec{A}/B$.

Taking the curl of equation (2.3) and noting that the curl of the vector potential is equal to the field, we get

$$\frac{\vec{B}}{B\rho} = x_3 \cdot (\nabla \times \vec{R}). \quad (2.4)$$

This equation will be used later on to derive an expansion for the magnetic field.

2.2 Equations of motion

In this section the equations of motion of a charged particle in a magnetic field will be introduced. Before deriving the equations of motion we need to specify the coordinate system we are working in. In compliance to figure 1.1, we define the longitudinal axis (the axis along the beam pipe) to be the z -axis. Furthermore, the transverse axes are x (horizontal) and y (vertical).

The equations of motion for a charged particle in a magnetic field follow from the Lorentz force equation:

$$\vec{F} = q\vec{E} + q\vec{v} \times \vec{B}.$$

For a relativistic particle with $v \approx c$ traveling through a magnetic field of 1 Tesla¹, the bending force is $Bv \approx 3 \cdot 10^8$ V/m. In this setting we need an electric field of 300 million volts per meter to create the same bending force as a magnetic field of 1 Tesla. The effects of the electric field are therefore negligible:

$$\vec{F} \approx q\vec{v} \times \vec{B}. \quad (2.5)$$

Setting equation (2.5) equal to Newton's second law ($\vec{F} = m\ddot{\vec{x}}$) and noting that $\vec{v} = (\dot{x}, \dot{y}, \dot{z})$ gives the following equations of motion:

$$\begin{aligned} \ddot{x} &= \frac{q}{m} (\dot{y}B_z - \dot{z}B_y), \\ \ddot{y} &= \frac{q}{m} (\dot{z}B_x - \dot{x}B_z), \\ \ddot{z} &= \frac{q}{m} (\dot{x}B_y - \dot{y}B_x). \end{aligned}$$

Note that the factor q/m is constant. Since the change in the z -direction is always the same, i.e. the length of the magnet, this system can be reduced by one dimension by losing the time dependence and making the equations of motion functions of z , as presented in [11]:

$$\begin{aligned} x'' &= \frac{q}{p} \sqrt{1 + x'^2 + y'^2} (y'B_z - (1 + x'^2)B_y + x'y'B_x), \\ y'' &= -\frac{q}{p} \sqrt{1 + x'^2 + y'^2} (x'B_z - (1 + y'^2)B_x + x'y'B_y). \end{aligned}$$

Here the primes denote derivatives with respect to the longitudinal coordinate z . Note that the factor q/p is constant too, since we have conservation of momentum.

As a final step we introduce the magnetic rigidity $B\rho$ from equation (2.2) to the equations of motion:

$$x'' = \frac{\sqrt{1 + x'^2 + y'^2}}{B\rho} (y'B_z - (1 + x'^2)B_y + x'y'B_x), \quad (2.6)$$

$$y'' = -\frac{\sqrt{1 + x'^2 + y'^2}}{B\rho} (x'B_z - (1 + y'^2)B_x + x'y'B_y). \quad (2.7)$$

Now it should be clear why we introduced the magnetic rigidity in the previous section. In this way properties like mass, charge and momentum are all caught into one variable.

2.3 Taylor series expansion

In this section the general form of a Taylor series expansion in one and multiple variables will be given. Then a description is given on how these expansions will be implemented in the method that is presented in this thesis.

¹Comparable to the LHCb magnet [2].

For an infinitely differentiable function f , the Taylor series expansion in some open interval around $x = a$ is given by

$$\begin{aligned} f(x) &= \sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!} (x-a)^n \\ &= f(a) + f'(a)(x-a) + \frac{f''(a)}{2!} (x-a)^2 + \frac{f'''(a)}{3!} (x-a)^3 + \dots \end{aligned}$$

We can extend this single-variable Taylor series to a multivariable Taylor series expansion around more than one variable. For an infinitely differentiable function f , the 2-D Taylor series expansion around some open neighborhood $(x, y) = (a, b)$ is given by

$$\begin{aligned} f(x, y) &= \sum_{n=0}^{\infty} \left[\frac{1}{n!} \sum_{k=0}^n \binom{n}{k} \frac{\partial^n f}{\partial x^{n-k} \partial y^k} \Big|_{(a,b)} (x-a)^{n-k} (y-b)^k \right] \\ &= f(a, b) + f_x(a, b)(x-a) + f_y(a, b)(y-b) \\ &\quad + \frac{1}{2!} [f_{xx}(a, b)(x-a)^2 + 2f_{xy}(a, b)(x-a)(y-b) + f_{yy}(a, b)(y-b)^2] + \dots \end{aligned} \tag{2.8}$$

where

$$\binom{n}{k} = \frac{n!}{(n-k)!k!}$$

are the binomial coefficients.

A similar formula can be derived for expansions in three or more variables. Note that a multivariable expansion, besides the expansion terms in each variable, also contains cross terms. For example, in a second order expansion in two variables, besides the second derivatives f_{xx} and f_{yy} , there also is a term with a combined derivative f_{xy} . In general, for a n th order expansion in k variables, the sum of derivatives in each term satisfies $0 \leq n_1 + n_2 + \dots + n_k \leq n$.

In this thesis we want to determine a map of the trajectory of a particle in a magnetic field by making a Taylor series expansion of this trajectory. Using the above defined form of a multivariable Taylor series, the Taylor coefficients can be found by computing the derivatives of the trajectory. Since we don't have an expression for the trajectory, the Taylor coefficients cannot be determined straight away. A more general form of a Taylor series expansion in k variables is

$$f(x_1, \dots, x_k) = \sum_{n_1, \dots, n_k} b_{n_1 \dots n_k} (x_1 - a_1)^{n_1} \dots (x_k - a_k)^{n_k}, \tag{2.9}$$

where $b_{n_1 \dots n_k}$ are the Taylor coefficients. These coefficients can now be found by inserting the Taylor expansion into the equations of motion.

The great thing about Taylor series is that they can be used for solving differential equations. This property is what we will be using when determining an approximation for the particle trajectory. To show how Taylor series can be used to solve differential equations, let's do an example. Consider the differential equation

$$y'' + y = 0.$$

If we want to determine a solution to this equation using a series expansion about the point $x_0 = 0$, then this solution will be of the form

$$y(x) = \sum_{n=0}^{\infty} a_n x^n.$$

Determining the second derivative of this expansion and plugging the expansions into the differential equation gives

$$\sum_{n=2}^{\infty} n(n-1)a_n x^{n-2} + \sum_{n=0}^{\infty} a_n x^n = 0.$$

Note that the second derivative of the expansion does not start at $n = 0$. To combine the two series into a single series, they both need to start at the same point. We also want the exponent of x to be the same in both series. This can be easily done by shifting down the first power series by 2. We then find

$$\sum_{n=0}^{\infty} [(n+2)(n+1)a_{n+2} + a_n] x^n = 0.$$

We know that for a power series that is zero for all x , all coefficients have to be zero as well. This gives the following recurrence relation:

$$(n+2)(n+1)a_{n+2} + a_n = 0, \quad n = 0, 1, 2, \dots$$

Solving this relation for the coefficients a_k gives the following two equations:

$$a_{2k} = \frac{(-1)^k a_0}{(2k)!}, \quad k = 0, 1, 2, \dots \quad \text{and} \quad a_{2k+1} = \frac{(-1)^k a_1}{(2k+1)!}, \quad k = 0, 1, 2, \dots$$

Plugging these coefficients back in to the general form of $y(x)$ we started with, we get that the solution of the differential equation is

$$y(x) = a_0 \sum_{k=0}^{\infty} \frac{(-1)^k x^{2k}}{(2k)!} + a_1 \sum_{k=0}^{\infty} \frac{(-1)^k x^{2k+1}}{(2k+1)!},$$

for some constants a_0 and a_1 that can be found by applying the initial conditions.

The actual solution to $y'' + y = 0$ is given to be

$$y(x) = c_1 \cos(x) + c_2 \sin(x),$$

for some constants c_1 and c_2 . Note that the Taylor series of cosine and sine are

$$\cos(x) = \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n}}{(2n)!} \quad \text{and} \quad \sin(x) = \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n+1}}{(2n+1)!}.$$

So, the solution we found using the Taylor series method is exactly the same as the solution from first principles.

2.4 Relativistic effects

In the LHCb experiment particles are accelerated to velocities close to the speed of light. This means that relativistic effects need to be considered in our method as well. In this section a description is given on how to take care of these effects in our tracking routine.

Einstein's theory of relativity states that the laws of physics are the same everywhere and that the speed of light is the same for every observer. To make sure that this theory is satisfied in our method, we need to replace the mass by the relativistic mass.

The relativistic mass is obtained by multiplying the mass by the Lorentz factor

$$\gamma = \frac{1}{\sqrt{1 - \frac{v^2}{c^2}}}.$$

At low speeds $\gamma \approx 1$, but at higher velocities this factor will have a significant effect. That is the reason why we need to implement this effect into the expansion. Notice that we can just replace the mass by the relativistic mass since it is not changed by the magnetic field. The momentum now becomes

$$p = \frac{mv}{\sqrt{1 - \frac{v^2}{c^2}}} = \gamma mv.$$

Then, the expression for x_3 is

$$x_3 = \frac{Bq}{p} = \frac{Bq}{\gamma mv}. \quad (2.10)$$

Chapter 3

The algebraic map

In this chapter the method presented by J. Bahrtdt and G. Wüstefeld [9] is used to derive an algebraic map over a finite interval z . This is done by first deriving a Taylor expanded map of the trajectory in a fixed Cartesian coordinate system. Then, a Taylor series expansion of the shape function \vec{R} is derived to describe the magnetic field. Together with the equations of motion from the previous chapter, we will describe a map that maps the initial coordinates and velocities of a particle to the final coordinates and velocities.

In the second part of this chapter, this procedure is worked out in second order. Finally, as an example, we will use the derived algebraic map to determine the trajectory of a muon with a velocity of $2.997 \cdot 10^8$ m/s in a constant magnetic field of 1.05 T.

3.1 Taylor expansion of the trajectory

In this section the general form of a Taylor expanded map will be given. Preparations will be done regarding the implementation of the maps into the equations of motion (2.6) and (2.7). Integration constants will already be determined for later purposes.

From the previous chapter we know that the general form of a Taylor expanded map along the z -axis with respect to the variables x'_i , y'_i and x_3 will be of the form:

$$\begin{aligned}x(z) &= \sum_{k,l,m} a_{klm}(z) \cdot x_i'^k \cdot y_i'^l \cdot x_3^m \\y(z) &= \sum_{k,l,m} b_{klm}(z) \cdot x_i'^k \cdot y_i'^l \cdot x_3^m\end{aligned}$$

where all three variables are expanded around 0. It should be noted that besides the z -dependence of x and y , these maps are also functions of x'_i , y'_i and x_3 . We will keep this dependence implicit to prevent things from becoming messy. Also, since we are Taylor expanding along the z -axis, the Taylor coefficients a_{klm} and b_{klm} will be functions of z .

To determine the values of the coefficients $a_{klm}(z)$ and $b_{klm}(z)$, we will insert the Taylor expanded maps into the equations of motion (2.6) and (2.7). Hence, we need to differentiate the Taylor series

twice with respect to z :

$$\begin{aligned} x'(z) &= \sum_{k,l,m} a'_{klm}(z) \cdot x_i'^k \cdot y_i'^l \cdot x_3^m \\ x''(z) &= \sum_{k,l,m} a''_{klm}(z) \cdot x_i'^k \cdot y_i'^l \cdot x_3^m \end{aligned}$$

and

$$\begin{aligned} y'(z) &= \sum_{k,l,m} b'_{klm}(z) \cdot x_i'^k \cdot y_i'^l \cdot x_3^m \\ y''(z) &= \sum_{k,l,m} b''_{klm}(z) \cdot x_i'^k \cdot y_i'^l \cdot x_3^m \end{aligned}$$

As we will find out later on, all coefficients $a_{klm}(z)$ and $b_{klm}(z)$ and their first derivatives will be zero at $z = 0$. This means that we can already determine the integration constants that arise when integrating $a''_{klm}(z)$ and $b''_{klm}(z)$. Notice that the z -dependence of $x(z)$ and $y(z)$ completely resides in the coefficients $a_{klm}(z)$ and $b_{klm}(z)$. This implies that the integration constants of $x(z)$ and $y(z)$ are in fact the integration constants of $a_{klm}(z)$ and $b_{klm}(z)$, respectively.

Integration of $x''(z)$ gives

$$\begin{aligned} x'(z) &= \sum_{k,l,m} a'_{klm}(z) \cdot x_i'^k \cdot y_i'^l \cdot x_3^m + c_1 \\ x(z) &= \sum_{k,l,m} a_{klm}(z) \cdot x_i'^k \cdot y_i'^l \cdot x_3^m + z \cdot c_1 + c_2 \end{aligned}$$

Setting $z = 0$, we get

$$\begin{aligned} x_i' &= x'(z=0) = c_1, \\ x_i &= x(z=0) = c_2. \end{aligned}$$

A similar result is found for $y(z)$.

By determining the integration constants right now, we don't have to worry about them at any point later on. This is also an advantage with regards to the implementation of the algorithm in a computer script.

Hence, we can define the general form of a Taylor expanded map along the z -axis with respect to the variables x'_i , y'_i and x_3 to be

$$x(z) = x_i + z \cdot x'_i + \sum_{k,l,m} a_{klm}(z) \cdot x_i'^k \cdot y_i'^l \cdot x_3^m \quad (3.1)$$

$$x'(z) = x'_i + \sum_{k,l,m} a'_{klm}(z) \cdot x_i'^k \cdot y_i'^l \cdot x_3^m \quad (3.2)$$

$$x''(z) = \sum_{k,l,m} a''_{klm}(z) \cdot x_i'^k \cdot y_i'^l \cdot x_3^m \quad (3.3)$$

$$y(z) = y_i + z \cdot y'_i + \sum_{k,l,m} b_{klm}(z) \cdot x_i'^k \cdot y_i'^l \cdot x_3^m \quad (3.4)$$

$$y'(z) = y'_i + \sum_{k,l,m} b'_{klm}(z) \cdot x_i'^k \cdot y_i'^l \cdot x_3^m \quad (3.5)$$

$$y''(z) = \sum_{k,l,m} b''_{klm}(z) \cdot x_i'^k \cdot y_i'^l \cdot x_3^m \quad (3.6)$$

Even though the maps are functions of the longitudinal coordinate z , for the method that we want to derive we only need to know the position and angle variables at the end of the magnet ($z = z_{final}$). An advantage of this way of defining the position and angle variables is that we can plot a trajectory that is determined from the Taylor expanded map and compare this plot with the actual plot.

Recalling the definition of x_3 , equation (2.10), we can conclude that x_3 is zero if either the particle has no charge, the particle has infinite momentum or when there is no magnetic field. So, we will exclude all terms in the expansion for which $x_3 = 0$, because this just gives a straight trajectory. This means that for a n th order approximation the indices k, l, m are restricted to $1 \leq k + l + m \leq n$. Note that $k + l + m$ is at least one since we assume that every term in the expansion has at least one power of x_3 .

3.2 Taylor expanding the magnetic field

As mentioned earlier, the shape function \vec{R} plays an important role. In this section we will derive the Taylor series expansion of \vec{R} and explain how this expansion is used to determine the expansion of the magnetic field \vec{B} .

Using the definition of the 2-D Taylor series, equation (2.8), we can define a Taylor series expansion for the three components of \vec{R} :

$$R^u(x, y, z) = \sum_{n=0}^{\infty} \left[\frac{1}{n!} \sum_{k=0}^n \binom{n}{k} \frac{\partial^n R^u}{\partial x^{n-k} \partial y^k} \Big|_{(x_i, y_i, z)} (x - x_i)^{n-k} (y - y_i)^k \right],$$

where $u = x, y$ or z . It should be noted that R^u is only expanded in x and y to keep the z -dependence in the Taylor coefficients. This is because the biggest variation in the LHCb magnetic field is along the z -axis.

This equation can be simplified by defining the coefficients $R_{n_1 n_2 n_3}^u$ of the shape function as

$$R_{n_1 n_2 n_3}^u = \frac{\partial^{n_1+n_2+n_3} R^u}{\partial x^{n_1} \partial y^{n_2} \partial z^{n_3}}. \quad (3.7)$$

evaluated at the position $(x = x_i, y = y_i, z)$. This works as long as we assume that the angle at which the particles enter the magnetic field and the deflection of the particles is small (which is the case for the LHCb experiment).

Also, note that

$$\frac{1}{n!} \sum_{k=0}^n \binom{n}{k} = \sum_{k=0}^n \frac{1}{(n-k)!k!}.$$

Substituting $n_1 = n - k$ and $n_2 = k$, we get

$$R^u(x, y, z) = \sum_{n_1, n_2} \frac{R_{n_1 n_2 0}^u}{n_1! n_2!} (x - x_i)^{n_1} (y - y_i)^{n_2}. \quad (3.8)$$

Now that we have a Taylor expansion of \vec{R} , the expansion of \vec{B} can be obtained by inserting equation (3.8) into equation (2.4).

3.3 Putting it all together

Now that the coordinate and magnetic field expansions are defined, we can start with putting everything together. This section will explain how the expansions can be combined to obtain the coordinate expansion coefficients.

Before we combine all equations that we have derived thus far to determine the coordinate coefficients $a_{klm}(z)$ and $b_{klm}(z)$, we first need to deal with the $\sqrt{1 + x'^2 + y'^2}$ that appears in the equations of motion. When the expansions are inserted in the equations of motion, the coordinate coefficients can be obtained by comparing the coefficients of products of equal order that are formed by x'_i , y'_i and x_3 . Notice that the square root will contain products of the form $x_i'^a \cdot y_i'^b \cdot x_3^c$ as well. But comparing these products will be quite hard as long as they are inside the square root. Hence, we will approximate $\sqrt{1 + x'^2 + y'^2}$ by taking Taylor series of the square root around $(0, 0)$ using equation (2.8):

$$\begin{aligned} \sqrt{1 + x'^2 + y'^2} &= \sum_{m,n} \frac{1}{m!n!} \cdot \left(\frac{\partial^{m+n}}{\partial x^m \partial y^n} \sqrt{1 + x'^2 + y'^2} \right) \Big|_{(0,0)} \cdot x'^m \cdot y'^n \\ &= 1 + \frac{x'^2}{2} + \frac{y'^2}{2} + \dots \end{aligned} \quad (3.9)$$

We can finally start with putting everything together. Starting with the equations of motion (2.6) and (2.7), we insert the expansions of the square root (3.9), the magnetic field (3.8) and the coordinates expansions (3.2), (3.3), (3.5) and (3.6) into these equations. This results into two huge expressions.

The coefficients of the coordinate expansion can then be obtained by comparing the coefficients of products of equal order. As one might imagine, it's nearly impossible to do this by hand. Even in second order this procedure will be quite tedious and error-prone (as will be shown in the next section). Hence the derivation of the coordinate expansion coefficients should be done with a computer program. In the next chapter an outline will be given on how to implement this method in a Mathematica code.

3.4 Result in second order

Although it's hard to work out the coefficients by hand, in this section two coefficients are worked out in second order to give an idea of how this process should be carried out. The rest of the coefficients will be given to compare for similarities between them.

For a second order expansion the coordinate expansions are given by

$$x(z) = x_i + z \cdot x'_i + a_{001}(z) \cdot x_3 + a_{002}(z) \cdot x_3^2 + a_{011}(z) \cdot y'_i \cdot x_3 + a_{101}(z) \cdot x'_i \cdot x_3 \quad (3.10)$$

$$x'(z) = x'_i + a'_{001}(z) \cdot x_3 + a'_{002}(z) \cdot x_3^2 + a'_{011}(z) \cdot y'_i \cdot x_3 + a'_{101}(z) \cdot x'_i \cdot x_3 \quad (3.11)$$

$$x''(z) = a''_{001}(z) \cdot x_3 + a''_{002}(z) \cdot x_3^2 + a''_{011}(z) \cdot y'_i \cdot x_3 + a''_{101}(z) \cdot x'_i \cdot x_3 \quad (3.12)$$

$$y(z) = y_i + z \cdot y'_i + b_{001}(z) \cdot x_3 + b_{002}(z) \cdot x_3^2 + b_{011}(z) \cdot y'_i \cdot x_3 + b_{101}(z) \cdot x'_i \cdot x_3 \quad (3.13)$$

$$y'(z) = y'_i + b'_{001}(z) \cdot x_3 + b'_{002}(z) \cdot x_3^2 + b'_{011}(z) \cdot y'_i \cdot x_3 + b'_{101}(z) \cdot x'_i \cdot x_3 \quad (3.14)$$

$$y''(z) = b''_{001}(z) \cdot x_3 + b''_{002}(z) \cdot x_3^2 + b''_{011}(z) \cdot y'_i \cdot x_3 + b''_{101}(z) \cdot x'_i \cdot x_3 \quad (3.15)$$

The expansion of the shape function \vec{R} is given by

$$R^u(x, y, z) = R^u_{000}(z) + R^u_{100}(z) \cdot (x - x_i) + R^u_{010}(z) \cdot (y - y_i).$$

Note that for a second order expansion of the trajectory, only the first order expansion of \vec{R} is needed. This is because the coefficients in $x''(z)$ and $y''(z)$ only have products of one or two terms. By definition, one of the x_3 's comes from the magnetic field expansion. So when we compare for coefficients of products of equal order, we only have to look for terms that consist of one factor of x'_i , y'_i or x_3 . And because second order only adds squared terms to the expansion, a first order expansion of \vec{R} is sufficient.

Using a similar reasoning we can conclude that we only need the first term in the expansion of the square root in the equations of motion, i.e. $\sqrt{1 + x'^2 + y'^2} = 1$.

The magnetic field expansions can then be determined from equation (2.4) and are given by

$$\begin{aligned} \frac{B_x}{B\rho} &= x_3 \cdot \left(\frac{\partial}{\partial y} R^z - \frac{\partial}{\partial z} R^y \right) \\ &= x_3 \cdot [R^z_{010} + R^z_{110} \cdot (x - x_i) + R^z_{020} \cdot (y - y_i) - R^y_{001} - R^y_{101} \cdot (x - x_i) - R^y_{011} \cdot (y - y_i)] \end{aligned} \quad (3.16)$$

$$\begin{aligned} \frac{B_y}{B\rho} &= x_3 \cdot \left(\frac{\partial}{\partial z} R^x - \frac{\partial}{\partial x} R^z \right) \\ &= x_3 \cdot [R^x_{001} + R^x_{101} \cdot (x - x_i) + R^x_{011} \cdot (y - y_i) - R^z_{100} - R^z_{200} \cdot (x - x_i) - R^z_{110} \cdot (y - y_i)] \end{aligned} \quad (3.17)$$

$$\begin{aligned} \frac{B_z}{B\rho} &= x_3 \cdot \left(\frac{\partial}{\partial x} R^y - \frac{\partial}{\partial y} R^x \right) \\ &= x_3 \cdot [R^y_{100} + R^y_{200} \cdot (x - x_i) + R^y_{110} \cdot (y - y_i) - R^x_{010} - R^x_{110} \cdot (x - x_i) - R^x_{020} \cdot (y - y_i)] \end{aligned} \quad (3.18)$$

Note that the z -dependence of all coefficients $R^u_{n_1 n_2 n_3}$ is kept implicit.

Let's restate the formulas of the equations of motion from chapter 2 as we will need them a lot:

$$x'' = \frac{1}{B\rho} (y' B_z - (1 + x'^2) B_y + x' y' B_x) \quad (3.19)$$

$$y'' = -\frac{1}{B\rho} (x' B_z - (1 + y'^2) B_x + x' y' B_y) \quad (3.20)$$

As an example to show how the coordinate coefficients should be obtained, let's determine $a''_{001}(z)$. Looking at equation (3.12), we see that the product that belongs to $a''_{001}(z)$ is x_3 . As mentioned before, one factor of x_3 comes from the magnetic field expansions by definition. So we are looking for terms that do not contain any factor of x'_i , y'_i or x_3 . Since both $x'(z)$ and $y'(z)$ only have terms containing these factors, the only term in equation (3.19) that we need to check is $-B_y$. The terms in equation (3.17) without any any factor of x'_i , y'_i or x_3 are R_{010}^x and $-R_{001}^y$. We conclude that

$$a''_{001}(z) = -R_{001}^x(z) + R_{100}^z(z).$$

Let's determine another coefficient to show how complicated this procedure can become. From equation (3.12) we find that the product that belongs to $a''_{002}(z)$ is x_3^2 . Now, besides the term $-B_y$, we also need to consider $y'B_z$. Note that x'^2 and $x'y'$ cannot contain a term with single factor of x_3 . In equation (3.14) there is only one term with a single x_3 , namely $b'_{001}(z)$. There is no term without any of the variables. The next thing we need to do is look at $B_z/(B\rho)$ and again search for terms without any of the three variables. We find that the first part of $a''_{002}(z)$ is $b'_{001}(z) \cdot [R_{100}^y(z) - R_{010}^x(z)]$.

For the second part of $a''_{002}(z)$, we again turn to $-B_y$. Notice that we now need to look for terms that contain a second factor of x_3 . At first sight it might seem that there aren't any, but remember that x and y are expansions as well, i.e. equations (3.10) and (3.13). In the expansion of $x(z)$ and $y(z)$ we find that the terms that contain a single factor of x_3 are $a_{001}(z)$ and $b_{001}(z)$, respectively. Hence, the second part of a''_{002} is $-a_{001}(z) \cdot R_{101}^x(z) + a_{001}(z) \cdot R_{200}^z(z) - b_{001}(z) \cdot R_{011}^x + b_{001}(z) \cdot R_{110}^z$. We finally conclude that

$$\begin{aligned} a''_{002}(z) = & -b'_{001}(z) \cdot R_{010}^x(z) + b'_{001}(z) \cdot R_{100}^y(z) - a_{001}(z) \cdot R_{101}^x(z) + a_{001}(z) \cdot R_{200}^z(z) \\ & - b_{001}(z) \cdot R_{011}^x + b_{001}(z) \cdot R_{110}^z. \end{aligned}$$

The other coefficients can be found in a similar manner. All second derivatives of the coordinate expansion coefficients up to second order are

$$\begin{aligned} a''_{001}(z) &= -R_{001}^x(z) + R_{100}^z(z) \\ b''_{001}(z) &= -R_{001}^y(z) + R_{010}^z(z) \\ a''_{002}(z) &= -b'_{001}(z) \cdot R_{010}^x(z) + b'_{001}(z) \cdot R_{100}^y(z) - a_{001}(z) \cdot R_{101}^x(z) + a_{001}(z) \cdot R_{200}^z(z) \\ &\quad - b_{001}(z) \cdot R_{011}^x + b_{001}(z) \cdot R_{110}^z \\ b''_{002}(z) &= a'_{001}(z) \cdot R_{010}^x(z) - a'_{001}(z) \cdot R_{100}^y(z) - a_{001}(z) \cdot R_{101}^y(z) + a_{001}(z) \cdot R_{110}^z(z) \\ &\quad - b_{001}(z) \cdot R_{011}^y + b_{001}(z) \cdot R_{020}^z(z) \\ a''_{011}(z) &= -z \cdot R_{011}^x(z) + z \cdot R_{110}^z(z) - R_{010}^x(z) + R_{100}^y(z) \\ b''_{011}(z) &= -z \cdot R_{011}^y(z) + z \cdot R_{020}^z(z) \\ a''_{101}(z) &= -z \cdot R_{101}^x(z) + z \cdot R_{200}^z(z) \\ b''_{101}(z) &= -z \cdot R_{101}^y(z) + z \cdot R_{110}^z(z) + R_{010}^x(z) - R_{100}^y(z) \end{aligned}$$

3.5 Example: Constant magnetic field

Now that the second derivatives of the coordinate expansion coefficients are determined, we will end this chapter by using the obtained results to give an easy example. Assume that we have a constant magnetic field of 1.05 T in the y -direction, i.e. $B_y = 1.05$ and $B_x = B_z = 0$. Using equation (2.1) we

can determine a possible form of \vec{A} . It can be verified by the reader that $\vec{A} = (1.05z, 0, 0)$ satisfies $\vec{B} = \nabla \times \vec{A}$. Then

$$\vec{R} = \frac{\vec{A}}{B} = (z, 0, 0).$$

From equation (3.7) we then find that $R_{001}^x = 1$ and all other coefficients are 0. Inserting this into the coefficients from section 3.4, we find that all coefficients of $x''(z)$ and $y''(z)$ are zero except for $a_{001}''(z)$. Integrating twice with respect to z gives

$$a_{001}(z) = -\frac{1}{2}z^2.$$

As explained in section 3.1, the integration constants are already incorporated in the Taylor expanded maps $x(z)$ and $y(z)$. So we can ignore the integration constants of $a_{001}(z)$ here, i.e. set them to 0.

Notice that in this case we could have integrated the coefficients $a_{klm}''(z)$ and $b_{klm}''(z)$ on the forehand, since the coefficients of R^u don't depend on z . In general this is not the case and hence the value of the coefficients of $x(z)$ and $y(z)$ can only be determined after the magnetic field coefficients have been found. Since these coefficients depend on the initial conditions of the particle, we can only determine the second derivatives of the coefficients on forehand. This means that the integration of $a_{klm}''(z)$ and $b_{klm}''(z)$ has to be done for every trajectory individually.

Finally, the Taylor expanded maps are given by

$$\begin{aligned} x(z) &= x_i + z \cdot x'_i - \frac{1}{2}z^2 \cdot x_3 \\ y(z) &= y_i + z \cdot y'_i \end{aligned}$$

Note that $y(z)$ is just the drift transformation. This makes sense as we only have a magnetic field in the y -direction. Also, the maps are very simple since we have a simple magnetic field. For more complex magnetic fields, these maps are expected to be more complex as well.

Since we are dealing with a constant magnetic field, the particle will follow a circular orbit with radius $\rho = p/(Bq)$. To show that the method gives the correct results, we can do a Taylor series expansion of the exact trajectory. For a charged particle entering the magnetic field at $\vec{x} = (0, 0, 0)$ perpendicularly, the trajectory can be described by

$$x(z) = \rho - \sqrt{\rho^2 - z^2}.$$

The second order Taylor expansion of $x(z)$ around $z_0 = 0$ is then given by

$$x(z) = x(0) + x'(0) \cdot (z - 0) + \frac{x''(0)}{2} \cdot (z - 0)^2 = -\frac{z^2}{2\rho}.$$

Note that this is exactly $a_{001}(z)$, since $\rho = 1/x_3$. So, we see that our method produces the same approximation as a Taylor expansion of the exact trajectory would give.

To finish our example, let's consider a muon entering the magnetic field at $x_i = y_i = 0$ perpendicularly ($x'_i = y'_i = 0$) with a velocity of $v = 2.997 \cdot 10^8$ m/s. The value of x_3 can be calculated using equation (2.10): $x_3 \approx -0.0740089$ m⁻¹. The bending radius of this particle is approximately 13.5 meters.

To see how accurate the obtained algebraic map is, we can compare it with the exact trajectory. As can be seen in figure 3.1, the second order solution already is a pretty good approximation for the first few meters. Only for larger distances the difference between the exact trajectory and the approximation becomes significant. The absolute error for the second order approximation is plotted in figure 3.2.

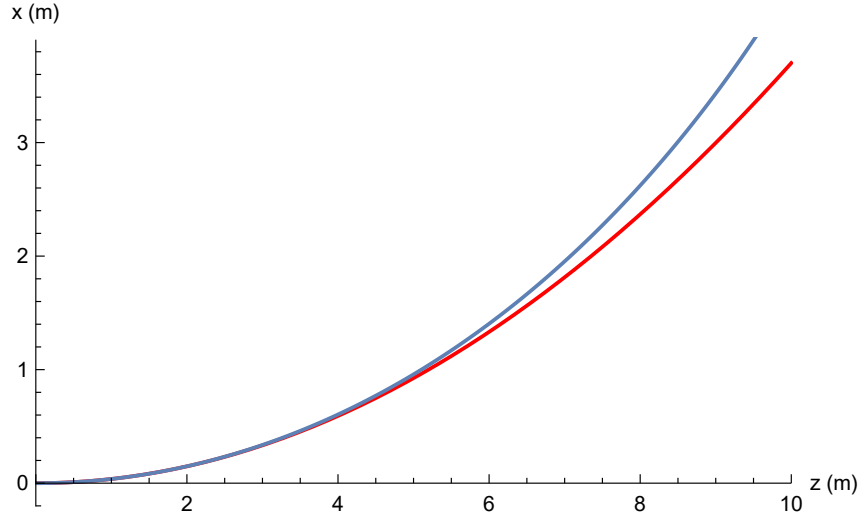


FIGURE 3.1: $x(z)$ -plot of the Taylor expanded map (red) and the exact trajectory (blue) over a distance of 10 meters. The plotted trajectories are for a muon with a velocity of $2.997 \cdot 10^8$ m/s and a field strength of 1.05 T.

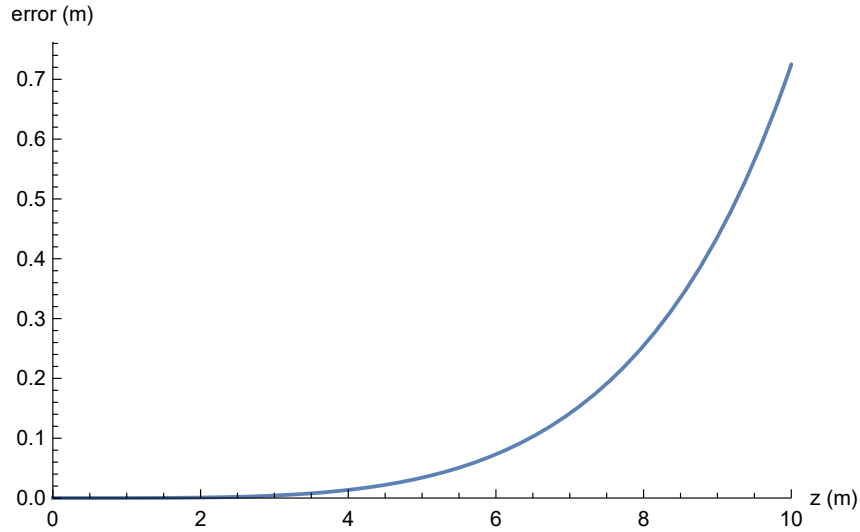


FIGURE 3.2: The error in the approximation over a distance of 10 meters. Notice that especially in the first few meters, the error is very small. At $z = 2$ meters, the approximation is only 0.82 millimeters off.

It should be noted that in this example we have used the (approximately) maximum field strength that can be found in the LHCb magnet (figure 1.2). For lower field strengths the bending radius ρ will be larger and hence x_3 will be smaller. This results in faster convergence. Also, the expanded map in figure 3.1 is only second order. As we will see in the next chapter, the result will be even better for higher order expansions (this will of course increase the computation time).

We can compute that after 2 meters the error in the approximation is still only 0.82 millimeters. This would suggest that it might be a good idea to split up the magnet into smaller segments and apply the method to each segment individually. This could possibly give high precision at low order approximations. This is left for future research.

Chapter 4

Determination of higher orders

In this chapter a description is given of what a Mathematica code that can be used to determine (the second derivatives of) the coordinate coefficients might look like. Then, some results that can be obtained using this Mathematica code are given.

4.1 Mathematica code to determine higher orders

In this section the Mathematica code that can be used to determine the coordinate coefficients is discussed step-by-step. The code is split up into three logical parts. The first part determines the expansion of the magnetic field. The second part determines the expansion of the trajectory. The last part puts everything together and determines the second derivatives of the coordinate coefficients. The complete script can be found in appendix B.

The code starts by defining the order for which the coefficients should be determined. The order is defined in `n`.

4.1.1 Taylor expansion of the magnetic field

The first part of the code starts by creating the Taylor series expansion of the magnetic field. First, we need to determine which coefficients $R_{n_1 n_2 0}$ are needed in n th order approximation, i.e. we need every combination of (n_1, n_2) such that $n_1 + n_2 < n$ (we have discussed in section 3.4 why we can do the expansion of the magnetic field in one order less). These combinations are stored in `magCoeff[i]`.

Since the code uses multiple iteration variables and the value of some of them is needed in later parts, the iteration variables will not be reused for different for-loops and are labeled by `iter#`.

```
iter1 = 0;
For[Subscript[n, 1] = 0, Subscript[n, 1] < n, Subscript[n, 1]++,
  For[Subscript[n, 2] = 0, Subscript[n, 2] < n, Subscript[n, 2]++,
    If[Subscript[n, 1] + Subscript[n, 2] < n, iter1++;
      magCoeff[iter1] = {Subscript[n, 1], Subscript[n, 2], 0}]]]
```

Now that the possible combinations of n_1 and n_2 are defined, we can define the expansion of R^u . For this we use equation (3.8) from section 3.2. Note that x and y are called `xf` and `yf` in the code respectively. As mentioned before, this is not necessarily the case, but it is in principle the only result we are looking for.

As a final remark, note that **Rcoeff** is made dependent on x, y and z . This is because the derivatives of \vec{R} are needed in the next step.

```
R[u_] = Sum[(Rcoef[u, x, y, z, magCoeff[i]] *
  (xf-xi)^magCoeff[i][[1]] * (yf-yi)^magCoeff[i][[2]]) /
  ((magCoeff[i][[1]])! * (magCoeff[i][[2]])!), {i, iter1}];
```

We now define the relation between \vec{B} and \vec{R} . For this we directly use equation (2.4). Notice that the code omits the division by the magnetic rigidity ($B\rho$).

```
B[1] = x3 * (D[R[3], y] - D[R[2], z]);
B[2] = x3 * (D[R[1], z] - D[R[3], x]);
B[3] = x3 * (D[R[2], x] - D[R[1], y]);
```

As a final step, we need to “clean up” the R -coefficients. The **Rcoeff** are not defined as the derivatives of \vec{R} (equation (3.7)) but just as functions of x, y, z . So, taking the derivative of R in the definition of the magnetic field will cause the derivatives of R to be displayed in two places, i.e. **Rcoeff** will be of the form

$$\text{Rcoeff}^{(0,n21,n22,n23,(0,0,0))}[u, x, y, z, (n11, n12, n13)],$$

where $n21, n22$ and $n23$ are the derivatives from the definition of the magnetic field, while $n11, n12$ and $n13$ are the derivatives from the expansion of R . Hence, we will recombine the derivatives $n21, n22$ and $n23$ with $n11, n12$ and $n13$ respectively. Also, we don’t need **Rcoeff** to be dependent on x, y, z anymore. **Rcoeff** will now be of the form

$$\text{Rcoeff}[u, (n1, n2, n3)].$$

```
\!\(\!*\SuperscriptBox[\(Rcoef\),
TagBox[
RowBox[{{"(",
RowBox[{"0", ",", "n21_", ",", "n22_", ",", "n23_", ",",
RowBox[{"(",
RowBox[{"0", ",", "0", ",", "0"}], ")}"}], ")"}],
Derivative],
MultilineFunction->None]\)[u_, x, y, z, {n11_, n12_, n13_}] :=
  R[u, {n11 + n21, n12 + n22, n13 + n23}];

Rcoef[u_, x, y, z, {n1_, n2_, n3_}] := R[u, {n1, n2, n3}];
```

4.1.2 Taylor expansion of the trajectory

Similar to the previous subsection, we start by determining which coefficients a_{klm} and b_{klm} are needed in n th order approximation, i.e. creating every combination of (k, l, m) such that $k+l+m \leq n$. Note that we can ignore every coefficient with $m = 0$, since in this case there is no bending (see section 3.1). The combinations are stored in **posCoeff[i]**.

```
iter2 = 0;
For[k = 0, k <= n, k++,
  For[l = 0, l <= n, l++,
    For[m = 1, m <= n, m++,
      If[k + l + m <= n, iter2++;
        posCoeff[iter2] = {k, l, m}]]]]
```

Now that all possible coefficients of the coordinate expansion are obtained, we can define the Taylor series expansion of x and y . For this we use equations (3.1) and (3.4), respectively.

```

xf = xi + z*xi' +
    Sum[aCoeff[z, posCoeff[i]]*(xi')^posCoeff[i][1]]*
    (yi')^posCoeff[i][2]*x3^posCoeff[i][3], {i, iter2}];

yf = yi + z*yi' +
    Sum[bCoeff[z, posCoeff[i]]*(xi')^posCoeff[i][1]]*
    (yi')^posCoeff[i][2]*x3^posCoeff[i][3], {i, iter2}];

```

We also need to define the expansion of the square root term, equation (3.9). Note that in the code the square root is expanded up to n th order. This is of course way too high, but this at least guarantees that we will not miss any terms from square root.

```

sqrtXY[x_, y_] := Sqrt[1 + x^2 + y^2];
sqrtXY = Sum[1/i! Sum[Binomial[i, k] * D[sqrtXY[a, b], {a, i-k}, {b, k}] *
    D[xf, z]^(i-k) * D[yf, z]^k, {k, 0, i}], {i, 0, n}] /. {a -> 0, b -> 0};

```

Now that all expansions are defined, we can define the equations of motion using equations (2.6) and (2.7). In the definition of the magnetic field we omitted the division by $(B\rho)$, so we need to omit it in the equations of motion as well.

```

EOMx = sqrtXY * (D[yf, z] * B[3] - (1 + (D[xf, z])^2) * B[2] +
    D[xf, z] * D[yf, z] * B[1]);
EOMy = -sqrtXY * (D[xf, z] * B[3] - (1 + (D[yf, z])^2) * B[1] +
    D[xf, z] * D[yf, z] * B[2]);

```

4.1.3 Comparing coefficients

In the script, the variables **EOMx** and **EOMy** are now long equations of (the derivatives of) the coefficients a_{klm} and b_{klm} , the coefficients $R_{n_1 n_2 n_3}^u$, and x'_i , y'_i and x_3 . The coefficients a''_{klm} and b''_{klm} can be obtained by comparing the coefficients of the products $x_i'^k \cdot y_i'^l \cdot x_3^m$ in x_f'' with **EOMx** (section 3.3). We do the same thing in y_f'' and **EOMy**.

First we need to determine all possible products $x_i'^k \cdot y_i'^l \cdot x_3^m$ that appear in x_f'' and y_f'' . We can use the combinations of (k, l, m) that are stored in **posCoeff[i]** to create these products. These products are stored in **product[i]**.

```

For[i = 1, i <= iter2, i++,
    product[i] = x3^posCoeff[i][3] * (xi')^posCoeff[i][1] *
    (yi')^posCoeff[i][2]]

```

To obtain all (second derivatives of the) coefficients of the trajectory, we use the function **Coefficient[]** to search for the coefficients of every product stored in **product[i]**. The results are stored in **EOMxCoeff[i]** and **EOMyCoeff[i]**. Notice that if the product is for example only x_3 , then the function also takes products of x'_i and y'_i as part of the coefficient. These terms don't belong to the coefficient, so we set x'_i and y'_i to zero afterwards to get rid them.

```

For[i = 1, i <= iter2, i++, EOMxCoeff[i] = Coefficient[EOMx, product[i]]
    /. {Derivative[1][xi] -> 0, Derivative[1][yi] -> 0}]

For[i = 1, i <= iter2, i++, EOMyCoeff[i] = Coefficient[EOMy, product[i]]
    /. {Derivative[1][xi] -> 0, Derivative[1][yi] -> 0}]

```

We have now determined the equations of the second derivatives of all coefficients of the coordinate expansions. These equations can be used by a different program to determine the trajectory of a (charged) particle in a magnetic field. This program needs to determine the coefficients $R_{n_1 n_2 n_3}^u$ and insert them into the above derived equations. These equations then have to be integrated twice with

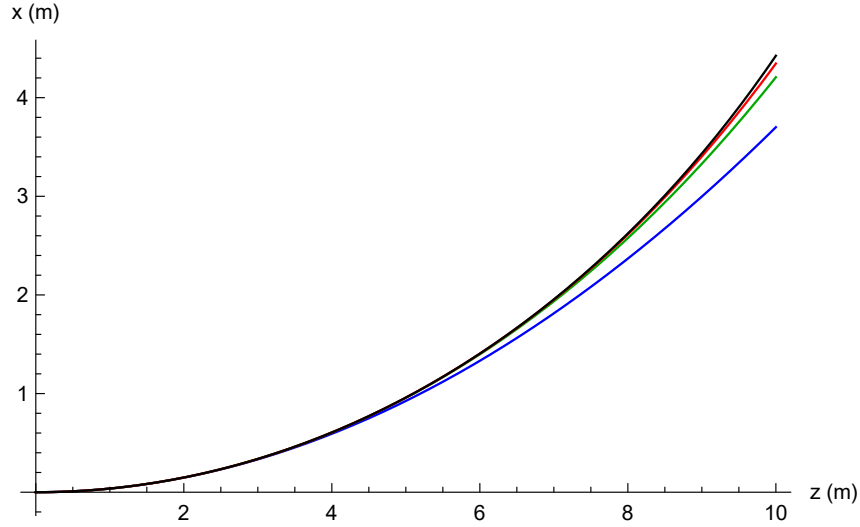


FIGURE 4.1: $x(z)$ -plot of the Taylor expanded map for the orders $n = 2$ (blue), $n = 3$ (green) and $n = 5$ (red) together with the exact trajectory (black) over a distance of 10 meters. The plotted trajectories are for a muon with a velocity of $2.997 \cdot 10^8$ m/s and a field strength of 1.05 T.

respect to z to obtain the coefficients $a_{klm}(z)$ and $b_{klm}(z)$. Finally, inserting these coefficients into the Taylor expansion of the trajectory will give the desired result.

4.2 Example: Constant magnetic field

Continuing with the example from section 3.5, we can use the derived code to determine higher order approximations of the trajectory of a particle in a constant field. Figure 4.1 shows the trajectory for the orders $n = 2, 3$ and 5 together with the exact solution. There is a clear improvement in the precision of the approximations for higher orders. Especially the fifth order approximation looks very promising. The errors in the approximations are shown in figure 4.2. After five meters, the error in the fifth order approximation is still only 0.2 millimeters.

4.3 Another example

The previous example used the most simple type of magnetic field, i.e. a constant field. This meant that we had only one nonzero coefficient $R_{n_1 n_2 n_3}^u$. Also, since we had a constant magnetic field, the value of x_3 was independent of the choice of the reference point at which it was defined. In the next example we will try to use a more complex field to show what the possibilities are and also what problems might arise when using this method. We will conclude that it is currently too difficult to come up with a good example that is relevant to the LHCb magnet. We will finish this section by using the method on a quadrupole magnet.

We want to use a magnetic field that sort of resembles to the actual field in the LHCb magnet. Looking at figure 1.2 would suggest that a Gaussian-shaped field in the y -direction would be a good starting point. So, we want to use a field of the form

$$B_y = a + be^{-\frac{(z-c)^2}{2d^2}},$$

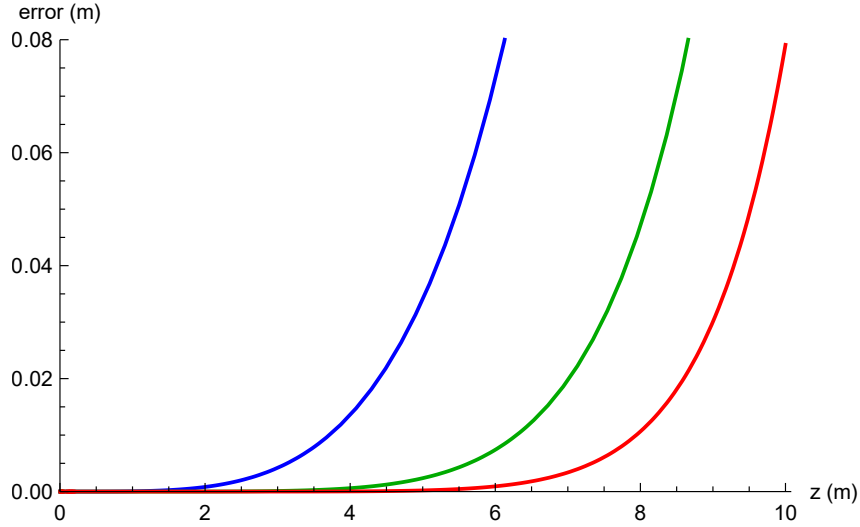


FIGURE 4.2: The error in the approximation over a distance of 10 meters for the orders $n = 2$ (blue), $n = 3$ (green) and $n = 5$ (red). At $z = 5$ meters, the error in the third order approximation is 0.2 centimeters; the error in the fifth order approximation is only 0.2 millimeters. Also, at $z = 10$ meters, the fifth order approximation is still only 7.9 centimeters off.

for some constants a, b, c, d . But this is where things become tricky. The coefficients $R_{n_1 n_2 n_3}^u$ that correspond to such a field will be very hard (if not impossible) to integrate. This example is not going to work.

The magnetic field of the LHCb magnet is not completely known, but rather it is approximated to high accuracy. This approximation could for instance be done using a development in series of basis functions as shown by Hicheur and Conti [12]. Perhaps we should try magnetic field equations that are polynomials, as these are more realistic to be used and also are much easier to integrate. But this will give problems as well. Note that, to find \vec{R} , we divide \vec{A} by the magnitude of the field. This will result in fractions of polynomials. Logarithmic terms will arise that give complex values when integrating the coefficients $R_{n_1 n_2 n_3}^u$. A trajectory with a complex part is not something we want. This is not going to work either.

A possible solution to avoid this last problem is to leave the magnetic field equations for what they are and directly define the components of the shape function. In this way we know for sure that the method will not run into problems that are caused by the manipulation of the magnetic field. It would be a final resort to at least test how well the method functions. Unfortunately, it's not easy to find equations for \vec{R} that result in a decent magnetic field as well. We will leave this as possible future research.

It should be clear that we are interested in a field that changes in the longitudinal direction. As mentioned, this results in too many issues. So, we will not continue with this anymore. What we can do, is test the method for a field that is constant in the z -direction, but does have a x and y -dependence. An example of such a field that is much used in accelerator physics is the quadrupole field. A quadrupole field has the nice property that its magnitude grows rapidly with the radial distance from its longitudinal axis, making it very useful for particle beam focusing. The general form of a quadrupole field is given by

$$\vec{B} = (Ky, Kx, 0),$$

for some field gradient K .

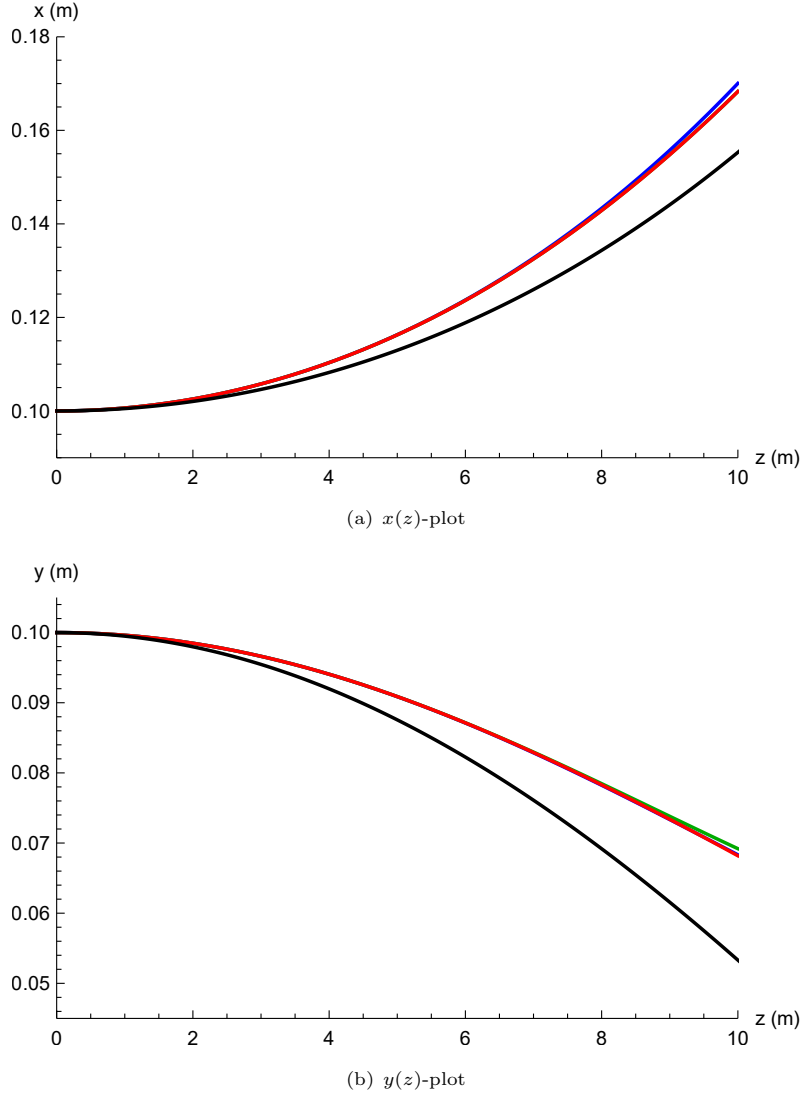


FIGURE 4.3: Plots of the Taylor expanded map for the orders $n = 2$ (blue), $n = 3$ (green) and $n = 5$ (red) together with the exact trajectory (black) over a distance of 10 meters. The plotted trajectories are for a muon with a velocity of $2.997 \cdot 10^8$ m/s. Notice that The precision is not really improved by including higher order terms.

Note that this magnetic field has a focusing effect in the xz -plane, while it is defocusing in the yz -plane. A possible vector potential that corresponds to this field and satisfies Maxwell's equations is

$$\vec{A} = (Kxz, 0, Kx^2/2).$$

We again use a muon entering the field perpendicularly. Only this time the muon enters the field at $x = y = 0.1$ meters. This is because at the center of the magnet, i.e. $x = y = 0$, the field is zero. The approximations for second, third and fifth order are shown in figure 4.3. The exact solution is obtained using the Euler method with very small step size (appendix A). The value of x_3 is defined along the line $(x, y) = (0.1, 0.1)$. Note that the value of x_3 does not depend on z . To show how the precision of the approximations behave over a distance of 10 meters, the errors between the approximations and the exact trajectories is shown in figure 4.4.

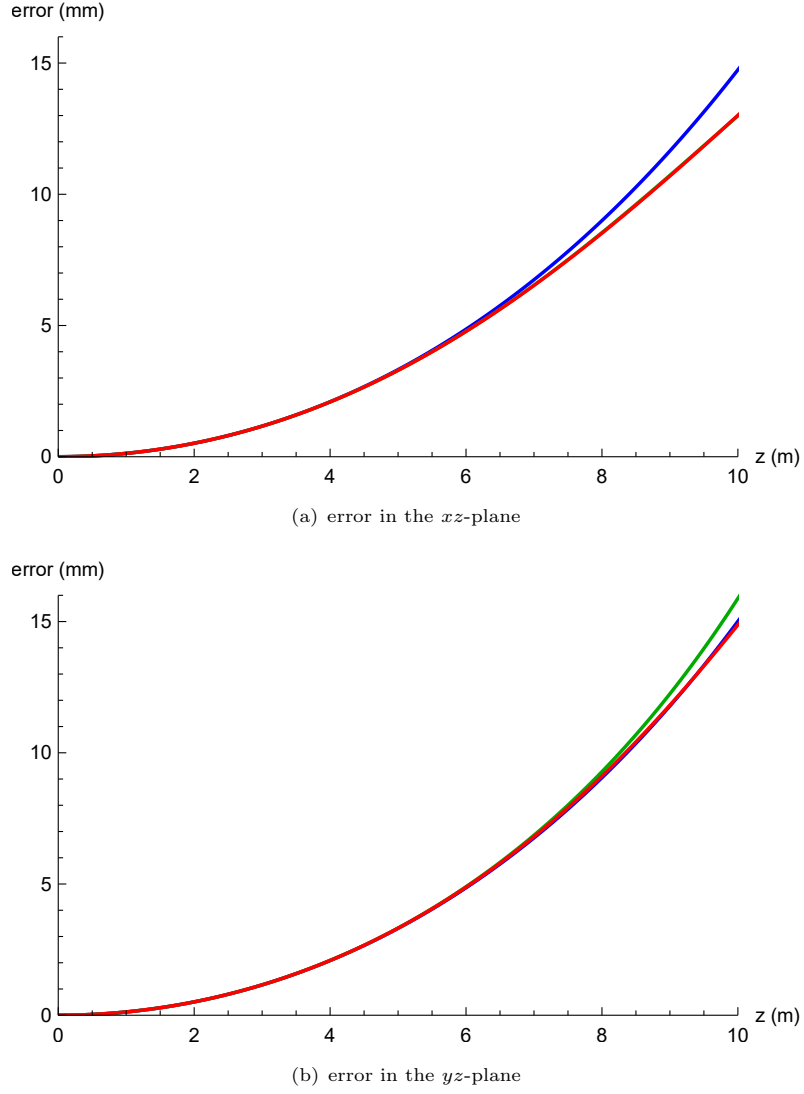


FIGURE 4.4: Errors in the Taylor expanded maps for the orders $n = 2$ (blue), $n = 3$ (green) and $n = 5$ (red) over a distance of 10 meters.

It can be seen in figures 4.3 and 4.4 that the final precision is already reached at low order, which illustrates the fast convergence of this approach. Notice that in the $y(z)$ -plot the third order approximation shows slightly worse results than the second order approximation. It is also interesting to note that in the xz -plane the fifth order approximation almost completely overlaps with the third order, while in the yz -plane the fifth order overlaps with the second order approximation.

Chapter 5

Conclusion

We have shown a possible method that can be used to track charged particles through an arbitrary magnetic field. The final goal was to use this method on the magnetic field of the LHCb detector. We have developed a Taylor expanded function that maps the initial position and transverse angle variables to the final position and angles. A nice feature of this method is that it can be programmed on parallel architectures. This will drastically increase the computation speed. The initial idea was to compute the expansion coefficients $a_{klm}(z)$ and $b_{klm}(z)$ on forehand completely, so that we only would need to fill in the initial conditions of the particle to determine its position and velocity at the end of the magnet. Unfortunately, it turned out that this is not possible with our setup. However, the second derivatives can be determined on forehand, which already saves a lot of computation time. Then, the only task that is left is to insert the magnetic field coefficients $R_{n_1 n_2 n_3}^u$ into the second derivatives $a''_{klm}(z)$ and $b''_{klm}(z)$, and integrate these coefficients twice with respect to z .

An expansion in three variables is assumed to be more advantageous than in four variables. The computation time will be less and the determination of the expansion coefficients $a_{klm}(z)$ and $b_{klm}(z)$ will be less complicated than it would be when expanding in four variables. But as it turned out, this method is very tricky. The result can be very dependent on the choice of the expansion parameter x_3 . There is a lot of room for improvement here, especially in the definition of the reference point at which x_3 is defined. But the introduction of x_3 also brought advantages. Most importantly, parameters like the charge, momentum and the magnetic field are all defined inside this one variable. This means that we don't have to care about these parameters anywhere else in the expansion. This also results in faster convergence.

For a constant field, this method shows decent results. This is because, for such a field the value of x_3 is the same, regardless of the reference point at which we define x_3 . Unfortunately, determining the trajectory of a particle in a constant magnetic field is something we can do without a program: the trajectories are just circles.

Although this method should work for arbitrary fields, we saw that it is hard to come up with an example that is relevant to the LHCb magnet. Perhaps we should put some restrictions on the magnetic field that is put in. Or maybe this issue can be resolved by making some minor adjustments to the method.

One thing to note is that this method works better when the bending radius is large, i.e. when x_3 is small. We also noted that the error in the approximation grows stronger when the distance over which the particle travels becomes larger. A possible way to improve this method to get higher precision is to determine the map for smaller segments. Splitting up the whole magnet into segments

of 1 meter or 10 centimeters and determining the trajectory for each segment individually would probably still be faster than integrating over the whole magnet using numerical integration.

All in all, we can say that this method has potential but is currently not ready to be used to track charged particles through the LHCb magnet.

Appendix A

Euler method

The Euler method is a method for solving ordinary differential equations (ODEs) with a given initial value. Although it is not the most accurate method (it does not converge very fast), it is very easy to program and when using a small enough step size a high precision will be obtained anyway. Since this method is only used for comparison with the expanded map, the long computation time that will arise is not an issue.

To use the Euler method, we need an ODE of the form

$$y'(t) = f(t, y(t)), \quad y(t_0) = y_0,$$

where y_0 is the value at the initial point $t = t_0$. For a step size h , we can define the time $t_n = t_0 + nh$. The general form of one step from t_n to t_{n+1} is then given by

$$y_{n+1} = y_n + hf(t_n, y_n).$$

To determine the exact trajectory of a charged particle in a magnetic field, we can again use the Lorentz force equation (2.5). Note that we are dealing with vectors, as we want to know all three position components. We now have two ODEs that are related to each other:

$$\begin{aligned} \vec{v}'(t) &= \frac{q}{\gamma m} \left[\vec{v}(t) \times \vec{B}(\vec{x}(t)) \right], & \vec{v}(t_0) &= \vec{v}_0, \\ \vec{x}'(t) &= \vec{v}(t), & \vec{x}(t_0) &= \vec{x}_0. \end{aligned}$$

One step in the Euler method will then be

$$\vec{v}_{n+1} = \vec{v}_n + h \cdot \frac{q}{\gamma m} \left[\vec{v}_n \times \vec{B}(\vec{x}_n) \right], \quad (\text{A.1})$$

$$\vec{x}_{n+1} = \vec{x}_n + h \cdot \vec{v}_n. \quad (\text{A.2})$$

Equations (A.1) and (A.2) are implemented in a Mathematica script that can be used to determine the exact trajectory.

```

(*Constants*)
cLight = 299792458;          (* m/s *)
eCharge = 1.60217662*10^-19; (* C *)

(*Parameters*)
mass = 1.88353*10^-28;      (* kg (muon mass) *)
velocity = 2.997*10^8;      (* m/s *)
q = -eCharge;               (* C *)

pos[0] = {0, 0, 0};
vel[0] = {0, 0, velocity};
gamma = 1 / Sqrt[1 - (Norm[vel[0]] / cLight)^2];
Bfield[{x_, y_, z_}] = {0, 1.05, 0}; (* T *)

(*Setting step size*)
time = 10^-7;
steps = 10^4;
h = time/steps;

(*Euler integration*)
Do[{vel[n] = vel[n-1] + h * (q / (gamma * mass)) *
   Cross[vel[n-1], Bfield[pos[n-1]]],
   pos[n] = pos[n-1] + h * vel[n-1]}, {n, 1, steps}]

position = Array[pos, steps];
xPos = position[[All, 1]];
yPos = position[[All, 2]];
zPos = position[[All, 3]];

xyPlot = ListLinePlot[Transpose[{xPos, yPos}], AxesLabel -> {"x (m)", "y (m)"}];
xzPlot = ListLinePlot[Transpose[{zPos, xPos}], AxesLabel -> {"z (m)", "x (m)"},
  PlotRange -> {{0, 10}, Automatic}];
yzPlot = ListLinePlot[Transpose[{zPos, yPos}], AxesLabel -> {"z (m)", "y (m)"},
  PlotRange -> {{0, 10}, Automatic}];

```

Appendix B

Mathematica code for determining coefficients

```
(*Define order*)
n = 2;

(*1: Taylor expansion of the magnetic field*)
iter1 = 0;
For[Subscript[n, 1] = 0, Subscript[n, 1] < n, Subscript[n, 1]++,
  For[Subscript[n, 2] = 0, Subscript[n, 2] < n, Subscript[n, 2]++,
    If[Subscript[n, 1] + Subscript[n, 2] < n, iter1++;
      magCoeff[iter1] = {Subscript[n, 1], Subscript[n, 2], 0}]]]

R[u_] = Sum[(Rcoef[u, x, y, z, magCoeff[i]] *
  (xf-xi)^magCoeff[i][[1]] * (yf-yi)^magCoeff[i][[2]]) /
  ((magCoeff[i][[1]])! * (magCoeff[i][[2]])!), {i, iter1}];

B[1] = x3 * (D[R[3], y] - D[R[2], z]);
B[2] = x3 * (D[R[1], z] - D[R[3], x]);
B[3] = x3 * (D[R[2], x] - D[R[1], y]);

(*Clean up coefficients*)
\\(\(*SuperscriptBox[\(Rcoef\),
TagBox[
RowBox[{"(",
RowBox[{"0", ",", "n21_", ",", "n22_", ",", "n23_", ",",
RowBox[{"(",
RowBox[{"0", ",", "0", ",", "0"}], "}")}],
Derivative],
MultilineFunction->None]\)[u_, x, y, z, {n11_, n12_, n13_}] :=
  R[u, {n11 + n21, n12 + n22, n13 + n23}];

Rcoef[u_, x, y, z, {n1_, n2_, n3_}] := R[u, {n1, n2, n3}];

(*2: Taylor expansion of the trajectory*)
iter2 = 0;
For[k = 0, k <= n, k++,
  For[l = 0, l <= n, l++,
    For[m = 1, m <= n, m++,
      If[k + l + m <= n, iter2++;
        posCoeff[iter2] = {k, l, m}]]]]

xf = xi + z*xi' +
  Sum[aCoeff[z, posCoeff[i]]*(xi')^posCoeff[i][[1]]*
  (yi')^posCoeff[i][[2]]*x3^posCoeff[i][[3]], {i, iter2}];

yf = yi + z*yi' +
  Sum[bCoeff[z, posCoeff[i]]*(xi')^posCoeff[i][[1]]*
  (yi')^posCoeff[i][[2]]*x3^posCoeff[i][[3]], {i, iter2}];

(*Taylor expansion of square root term*)
```



```

sqrtXY[x_, y_] := Sqrt[1 + x^2 + y^2];
sqrtXY = Sum[1/i! Sum[Binomial[i, k] * D[sqrtXY[a, b], {a, i-k}, {b, k}] *
  D[xf, z]^(i-k) * D[yf, z]^k, {k, 0, i}], {i, 0, n}] /. {a -> 0, b -> 0};

(*Equations of motion*)
EOMx = sqrtXY * (D[yf, z] * B[3] - (1 + (D[xf, z])^2) * B[2] +
  D[xf, z] * D[yf, z] * B[1]);
EOMy = -sqrtXY * (D[xf, z] * B[3] - (1 + (D[yf, z])^2) * B[1] +
  D[xf, z] * D[yf, z] * B[2]);

(*3: Compare coefficients*)
(*Create possible products*)
For[i = 1, i <= iter2, i++,
  product[i] = x3^posCoeff[i][[3]] * (xi')^posCoeff[i][[1]] *
    (yi')^posCoeff[i][[2]]

(*Find coefficients of EOMs*)
For[i = 1, i <= iter2, i++, EOMxCoeff[i] = Coefficient[EOMx, product[i]]
  /. {Derivative[1][xi] -> 0, Derivative[1][yi] -> 0}]

For[i = 1, i <= iter2, i++, EOMyCoeff[i] = Coefficient[EOMy, product[i]]
  /. {Derivative[1][xi] -> 0, Derivative[1][yi] -> 0}]

```

Bibliography

- [1] A. Augusto Alves, Jr. et al. The LHCb Detector at the LHC. *JINST*, 3:S08005, 2008.
- [2] LHCb Magnet: Technical Design Report. 2000.
- [3] R. Aaij et al. Precision Luminosity Measurements at LHCb. *JINST*, 9(12):P12005, 2014.
- [4] J. Myrheim and L. Bugge. A Fast Runge-Kutta Method for Fitting Tracks in a Magnetic Field. *Nucl. Instrum. Meth.*, 160:43–48, 1979.
- [5] J.C. Hart and D.H. Saxon. Track and Vertex Fitting in an Inhomogeneous Magnetic Field. *Nucl. Instrum. Meth.*, A220:309–326, 1984.
- [6] E. Valetov, M. Berz, and K. Makino. Direct Calculation of the Transfer Map of Electrostatic Deflectors, and Comparison with the Codes COSY INFINITY and GIOS. 2018.
- [7] M. Venturini and A.J. Dragt. Accurate Computation of Transfer Maps from Magnetic Field Data. *Nucl. Instrum. Meth.*, A427:387–392, 1999.
- [8] M. Berz. Differential Algebraic Description of Beam Dynamics to Very High Orders. *Part. Accel.*, 24:109–124, 1989.
- [9] J. Bahrtdt and G. Wüstefeld. A Taylor-Expanded Generating Function for Particle Motion in Arbitrary Magnetic Fields. In *Proceedings of EPAC 1992, Berlin, Germany*, pages 670–672, 1992.
- [10] J. Bahrtdt and G. Wüstefeld. Canonical Particle Tracking in Undulator Fields. In *Conference Record of the 1991 IEEE Particle Accelerator Conference*, pages 266–268.
- [11] K. Steffen. Basic Course on Accelerator Optics. In *CERN Accelerator School 1984*, 1985.
- [12] A. Hicheur and G. Conti. Parameterization of the LHCb Magnetic Field Map. In *2007 IEEE Nuclear Science Symposium Conference Record*, volume 4, pages 2439–2443, 2007.