Rigorous Field Analysis of Superconducting Magnets and the Influence on Nonlinear Dynamics in Particle Accelerators

By

Michael Lindemann

A THESIS

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

Department of Physics and Astronomy

1998

ABSTRACT

Rigorous Field Analysis of Superconducting Magnets and the Influence on Nonlinear Dynamics in Particle Accelerators

By

Michael Lindemann

The nonlinear dynamics of particles in modern accelerators are governed by the fields of the guiding and focusing superconducting magnets. While for the linear design of the lattice it is sufficient to treat the fields by a rather coarse approximation, a thorough analysis of the magnetic fields is necessary in order to study the nonlinear effects. As an example, the Large Hadron Collider (LHC) is studied, using detailed field data for the High Gradient Quadrupoles in the interaction region. The influence of the resulting nonlinearities on the dynamics is analysed via high-order maps determined with Differential Algebraic (DA) techniques and the code COSY INFINITY. Normal form methods are utilised to determine amplitude dependent tune shifts as well as resonance strengths. It is shown that the end effects change the nonlinear characteristics of the lattice significantly.

This made a rigorous treatment of the fields desirable, which is presented in the second part. Using DA-techniques the Taylor expansion of the field is calculated. Based on this expansion two analytical algorithms to determine the multipole content of the field are developed.

ACKNOWLEDGEMENTS

First of all I would like to thank my wife Wencke and my parents for their continued support during my studies.

I am grateful to Prof. Martin Berz for giving me the opportunity to come to Michigan State University and work in his outstanding research group. This group consisted of Béla Erdélyi, Jens Hoefkens, Khodr Shamseddine, Kyoko Makino, Jens von Bergmann and Lars Diening, who have all in various ways supported me. Especially I would like to express my gratitude to Béla Erdélyi, who was my tutor in the beginning and became my friend and partner in research later. I also appreciate the mighty help in all questions related to computers by my friend Jens Hoefkens.

Outside the group I got the chance to work with several scientists from Fermi National Accelerator Laboratory, namely Dr. Jim Holt, Dr. Carol Johnstone, Dr. Weishi Wan and Dr. Gianlucca Sabbi. I am grateful to them for many stimulating discussions. Specifically I would like to thank Gianlucca Sabbi for his help with the magnetic field calculation.

I appreciated the financial support from the German National Scholarship Foundation.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 The Large Hadron Collider

The Large Hadron Collider (LHC) will be the next accelerator to be built at CERN in Geneva. One of the the main goals of LHC is to study the actual mechanism for symmetry breaking in the electroweak sector of the standard model. This phenomenon is associated with the nature of the Higgs mechanism, the existence of the Higgs particle and the origin of mass.

The LHC poses unprecedented challenges in terms of accelerator physics :

High Luminosity

In the LHC the energy available in the collisions between the constituents of the protons, the quarks and gluons, will reach the 8 TeV range, which is about 10 times that of the Fermilab Tevatron. In order to maintain an equally effective physics program at a higher energy E the luminosity of a collider which is proportional to the number of collisions per second, should increase in proportion to E^2. This is because the cross section of the particle decreases like $1/E^2$. Whereas in past and present colliders the luminosity culminates around $L = 10^{32} cm^{-2} s^{-1}$, in the LHC it is expected to reach $L = 10^{34} cm^{-2} s^{-1}$. The luminosity is given by the formula [27]

$$L = \frac{1}{4\pi}\frac{\gamma}{\beta^*}\left[\frac{N}{\epsilon_n}\right][Nkf]\,F \tag{1.1}$$

where γ is the energy of the protons divided by their rest energy, β^* is the value of the betatron function, corresponding to the width of the beam at the collision point, N is the number of protons in the k bunches, ϵ_n is the invariant transverse emittance, f is the revolution frequency and F is a reduction factor due to the finite crossing angle which is 0.9 for the LHC. In formula (1.1) γ is limited by the bending magnet field and β^* is similarly largely determined by the available technology of high gradient quadrupole lenses. The first bracket is proportional to the beam-beam parameter which is limited by the electromagnetic interaction of colliding bunches. The second bracket is proportional to the beam current, which has to be chosen very large. This will be achieved by filling each of the two rings, in which the particles are rotating in opposite direction, with 2835 bunches of 1011 particles each. The resulting large beam current $I_b = 0.53A$ is particularly challenging in a machine made of delicate superconducting magnets operating at cryogenic temperatures.

Furthermore it is necessary to decrease β^* in order to reach the luminosity goal. This requires the use of High-Gradient Quadrupoles in the interaction region. These quadrupoles combine relatively short length, large aperture, and short focal length with a rather peculiar configuration of the return coils, all of which enhances the relevance of their fringe field effects. This is why a substantial part of this work will deal with the influence of the fringe fields on the nonlinear dynamics of the particles.

Longterm Stability

The nonlinear components of the guiding and focusing magnetic fields of the machine are also very important for the question of long term stability. The beams will be stored at high energy for about 10 hours. During this time the particles make

four hundred million revolutions around the 26.7 km circumference of the machine. Meanwhile the amplitude of their oscillations around the central orbit should not increase significantly, because this would dilute the beams and degrade luminosity.

1.2 Methods and Tools

In this section we present the main methods used in this work. The goal is to provide the reader with a basic intuitive understanding of these tools. We follow closely and summarise [1], [2] and [3]. The methods are discussed in a broader context in [4].

1.2.1 Coordinates and Maps

Usually when studying dynamics, the time t plays the role of the independent variable, and we study the motion of positions $\vec{x}$ and velocities $\vec{v}$ or momenta $\vec{p}$ as coordinates. Using the Lagrange mechanism, it is easy to transfer to new coordinates, in particular the coordinates that describe the relative motion around the reference orbit. Furthermore, instead of using t , we usually use the arc length s along the reference orbit as independent variable.

For the understanding of the motion in relative coordinates, let us assume we have studied and understood the motion of the reference orbit. In the case when there is no field at all, this reference orbit will merely follow a straight line. Furthermore, there are a host of devices used in accelerators that have fields, but along one given straight line, all the fields vanish, and the device is lined up in such a way that the reference particle follows this line. Another important device uses magnetic fields, and along the reference orbit one tries to hold them constant, in which case the reference orbit is circular, at least within the element. In all other cases, it is usually necessary to numerically integrate the reference orbit.

We assume the position and momenta of the reference particle are $\vec{r}_{ref}(s), \vec{p}_{ref}(s)$ are known. As a technical detail, let us also assume that for no point s, we have $\vec{p}_{ref}(s) \parallel \vec{e}_z$, i.e. the motion is never pointing straight up,which for most real accelerators is no limitation whatsoever. Let furthermore ρ_t be smaller than the minimum radius of curvature that the reference orbit experiences in the section of the machine that we want to study. We now consider a "flexible tube" of radius ρ_t centred around the reference orbit, and restrict the particles that we want to describe to only those within the tube. Again, for practical devices this hardly represents a limitation; in the LHC, for example, the "tube" would be more than 4 km wide, much larger than the region required by the beam particles.

For any particle within the tube, there is now a closest point on the reference orbit; because only particles within the tube are allowed, this point is indeed unique. Let s be the arc length at this point, and $\vec{r}_{ref}(s)$ the position of the reference particle on the reference orbit. Then the relative coordinates of the point $\vec{r}$ are obviously $\vec{r} - \vec{r}_{ref}(s)$.

Let now $\vec{e}_s$ be a unit vector in the direction of $\vec{p}_{ref}$. Consider now the plane perpendicular to $\vec{e}_s$. Of all the unit vectors in this plane, let $\vec{e}_y$ be the one with the largest "upward" component; because $\vec{p}_{ref}$ and hence $\vec{e}_s$ are not allowed to go straight up, this vector is well defined. Finally choose a third vector $\vec{e}_x$ as $\vec{e}_x = \vec{e}_y \times \vec{e}_s$. Because $\vec{e}_y$ has maximum "upward" component, $\vec{e}_x$ has vanishing upward component and hence lies in the horizontal plane.

Denote now by $\mathbf{x}$ the component of $\vec{r} - \vec{r}_{ref}(s)$ in the direction of $\vec{e}_x$, and by $\mathbf{y}$ the component of $\vec{r} - \vec{r}_{ref}(s)$ in the direction of $\vec{e}_y$. Similarly, define p_x and p_y to be the momentum components in the directions $\vec{e}_x$ and $\vec{e}_y$.

Furthermore, denote by δ the relative difference between the total (kinetic plus

potential) energy E of the particle under consideration and the reference energy E_0, i.e. $\delta = (E - E_0)/E_0$. Finally, introduce a space-like variable l to be the time of flight t minus the time of flight t_0 of the reference particle, multiplied by a constant k of dimension "velocity", i.e. $l = k\,(t - t_0)$. Then we form the vector $\vec{Z}$ of particle optical coordinates as

$$\vec{Z} = \begin{pmatrix} x \\ y \\ l = k\,(t - t_0) \\ a = p_x/p_0 \\ b = p_y/p_0 \\ \delta = (E - E_0)\,/E_0 \end{pmatrix} \tag{1.2}$$

where p_0 is some previously chosen scaling momentum; a natural choice may be to select the momentum of the reference particle at the beginning.

Note that due to the definition of $\vec{Z}$, the reference particle itself corresponds to $\vec{Z} = 0$, and hence the vector $\vec{Z}$ does indeed describe the relative motion.

The entire action of a beam physics device can now be expressed by how it manipulates the coordinates in the vector $\vec{Z}$. In fact, usually a set of initial conditions $\vec{Z}_0$ at position s_0 uniquely determines the future evolution and hence $\vec{Z}$ at any later position s, so we can define a function relating the initial conditions at s_0 to the conditions at s via

$$\vec{Z}\,(s) = \mathcal{M}\;(s_0, s)\left(\vec{Z}\,(s_0)\right)$$

The function $\mathcal{M}\;(s_0, s)$, which formally summarises the entire action of the system, is of great importance for the description and analysis of beam physics systems. It is often called the transfer function, the transfer map, or simply the map of the system. Note that the transfer functions satisfy the relationship

$$\mathcal{M}\;(s_1, s_2) \circ \mathcal{M}\;(s_0, s_1) = \mathcal{M}\;(s_0, s_2)\;,$$

which merely says that transfer maps of systems can be built up from the transfer maps of the pieces.

Since $\mathcal{M}$ describes the motion in relative coordinates, we always have

$$\mathcal{M}\ (\vec{0}) = \vec{0}$$

Furthermore, since by the very definition of a beam, the coordinates of $\vec{Z}$ are "small", $\mathcal{M}$ is usually only weakly nonlinear; because of this, it is often represented by its Taylor series expansion. In the following section we will present the differential algebraic methods which allows for the computation of the Taylor series expansion very efficiently.

1.2.2 Differential Algebras

In this section we will provide the mathematical background of the theory of differential algebras required for the study of nonlinear particle dynamics via the Taylor series representation of maps.

Historically, the treatment of functions in numerics has been done based on the treatment of numbers; and as a result, virtually all classical numerical algorithms are based on the mere evaluation of functions at specific points. As a consequence, numerical methods for differentiation, which are so relevant for the computation of Taylor representations of the map, are very cumbersome and prone to inaccuracies because of cancellation of digits, and not useful in practice for our purposes.

The method of differential algebra is based on the observation that it is possible to extract more information about a function than its mere values, specifically its Taylor expansion. We define the operation T to be the extraction of the Taylor coefficients of a pre-specified order n of the function. In mathematical terms, T is an

equivalence relation, and the application of T corresponds to the transition from the function to the equivalence class comprising all those functions with identical Taylor expansion to order n. Obviously Taylor coefficients of order n for sums and products of functions as well as scalar products with reals can be computed from those of the summands and factors. This means that the set of equivalence classes of functions can be endowed with well-defined operations, leading to the so-called Truncated Power Series Algebra [12],[13].This led to a method to extract maps to any desired order from a computer algorithm that integrates orbits numerically. Similar to the need for algorithms within floating point arithmetic, the development of algorithms for functions followed, including methods to perform composition of functions, to invert them, to solve nonlinear systems explicitly, and to introduce the treatment of common elementary functions [6], [3].

However, very soon afterwards it became apparent [1] that this only represents a half-way point, and one should proceed beyond mere arithmetic operations on function spaces of addition and multiplication and consider their analytic operations of differentiation and integration. This resulted in the recognition of the underlying differential algebraic structure and its practical exploitation.

In passing we note that in order to avoid loss of order, in practice the derivations have the form $\partial = h \cdot d/dx_i$, where h is a function with $h(0) = 0$. As a first consequence, it allowed to construct integration techniques to any order that for a given accuracy demand are substantially faster than conventional methods [3]. Subsequently, it was realized that the differential algebraic operations are useful for a whole variety of other questions connected to the analytic properties of the transfer map [6]. It was possible to determine arbitrary order generating function representations of maps [7], [3] and normal form methods [8], [9] could be performed to arbitrary order. [3] On the practical end, based on the latter concept, there are also several improvements

regarding methods of computational differentiation [10],[11].

In order to show how this method works in practice , we first address the simplest case of differential algebras, the structure ${}_1D_1$.

The Structure ${}_1D_1$

Consider the vector space R^2 of ordered pairs (a_0, a_1), $a_0, a_1 \in R$ in which an addition and a scalar multiplication are defined in the usual way:

$$(a_0, a_1) + (b_0, b_1) = (a_0 + b_0, a_1 + b_1) \tag{1.3}$$

$$t \cdot (a_0, a_1) = (t \cdot a_0, t \cdot a_1) \tag{1.4}$$

for $a_0, a_1, b_0, b_1 \in R$. Besides the above addition and scalar multiplication a multiplication between vectors is introduced in the following way:

$$(a_0, a_1) \cdot (b_0, b_1) = (a_0 \cdot b_0, a_0 \cdot b_1 + a_1 \cdot b_0) \tag{1.5}$$

for $a_0, a_1, b_0, b_1 \in R$. With this definition of a vector multiplication the set of ordered pairs becomes an algebra, denoted by ${}_1D_1$.

Note that the multiplication is the same one would obtain by multiplying $(a_0 + a_1 \cdot x)$ and $(b_0 + b_1 \cdot x)$ and keeping terms linear in x.

In the same way than in the case of complex numbers, one can identify $(a_0, 0)$ as the real number a_0. Where in the complex numbers, $(0, 1)$ was a root of -1, here it has another interesting property:

$$(0, 1) \cdot (0, 1) = (0, 0) \tag{1.6}$$

which follows directly from equation (1.5). So $(0,1)$ is a root of 0. Such a property suggests thinking of $d = (0,1)$ as something infinitely small, small enough that its square vanishes. Because of this we call $d = (0,1)$ the differential unit. The first component of the pair (a_0, a_1) is called the real part, and the second component is called the differential part.

It is easy to verify that $(1,0)$ is a neutral element of multiplication, because according to equation (1.5)

$$(1,0) \cdot (a_0, a_1) = (a_0, a_1) \cdot (1,0) = (a_0, a_1) \tag{1.7}$$

It turns out that (a_0, a_1) has a multiplicative inverse if and only if a_0 is nonzero; so ${}_1D_1$ is not a field. In case $a_0 \neq 0$ the inverse is

$$(a_0, a_1)^{-1} = (\frac{1}{a_0}, -\frac{a_1}{a_0^2}) \tag{1.8}$$

Using equation (1.5) it is easy to check that in fact $(a_0, a_1)^{-1} \cdot (a_0, a_1) = (1,0)$.

The space ${}_1D_1$ is a subspace of the field R^* introduced in Nonstandard Analysis [8],[9]. Besides the usual real numbers, R^* contains a variety of infinitely small and infinitely large quantities. The outstanding result of the theory of Nonstandard Analysis is that differentiation becomes an algebraic problem: a function f is differentiable if and only if for any arbitrarily small quantity δ, the real part of the quotient

$$\frac{f(x+\delta) - f(x)}{\delta} \tag{1.9}$$

is independent of the choice of the specific δ. Thus, given any differentiable function f, we can compute its derivatives by just evaluating the formula for a special choice of δ. We choose $\delta = d = (0,1)$ and thus obtain

$$
\begin{aligned}
f'(x) &= \mathcal{R}[\frac{f(x+d)-f(x)}{d}] \quad \text{or} \\
f'(x) &= \mathcal{D}[f(x+d)-f(x)] = \mathcal{D}[f(x+d)] \qquad (1.10)
\end{aligned}
$$

where $\mathcal{R}$ denotes the real part and $\mathcal{D}$ denotes the differential part. In the last step use has been made of the fact that $f(x)$ has no differential part. Hence differential algebras are useful to compute derivatives directly, without requiring an analytic formula for the derivative and without the inaccuracies of numerical techniques.

The computation of derivatives shall be illustrated in an example using the following function:

$$f(x) = \frac{1}{x+\frac{1}{x}} \qquad (1.11)$$

The derivative of the function is

$$f'(x) = \frac{\frac{1}{x^2}-1}{(x+\frac{1}{x})^2} \qquad (1.12)$$

Suppose we are interested in the value of the function and its derivative at x=2. We obtain

$$f(2) = \frac{2}{5} \quad f'(2) = -\frac{3}{25} \qquad (1.13)$$

Now take the definition of the function f in equation (1.11) and evaluate it at $2+d=(2,1)$. One obtains:

$$f[(2,1)] = \frac{1}{(2,1)+\frac{1}{(2,1)}}$$

$$
\begin{aligned}
&= \frac{1}{(2,1) + (\frac{1}{2}, -\frac{1}{4})} \\
&= \frac{1}{(\frac{5}{2}, \frac{3}{4})} \\
&= (\frac{2}{5}, -\frac{3}{4} / \frac{25}{4}) \\
&= (\frac{2}{5}, -\frac{3}{25}) \qquad (1.14)
\end{aligned}
$$

As we can see, after the evaluation of the function the real part of the result is just the value of the function at $x = 2$, whereas the differential part is the derivative of the function at $x = 2$.

This is exactly what was to be expected from the theory of Nonstandard Analysis. However, for the sake of not relying on the quite advanced techniques of this relatively new field of mathematics, we also present an elementary but less elegant proof of the result.

By our choice of the starting vector $(2, 1)$, initially the vector contains the value $I(2)$ of the identity function $I : x \to x$ in the first component and the derivative of $I'(2) = 1$ in the second component.

Now assume that in an intermediate step two vectors of value and derivative $(g(2), g'(2))$ and $(h(2), h'(2))$ have to be added. According to (1.3) one obtains $(g(2) + h(2), g'(2) + h'(2))$. But according to the rule for the differentiation of sums, this is just the value and derivative of the sum function $(g + h)$ at $x = 2$.

The same holds for the multiplication: Suppose that two vectors of value and derivatives $(g(2), g'(2))$ and $(h(2), h'(2))$ have to be multiplied. Then according to (1.5) one obtains $(g(2) \cdot h(2), g(2) \cdot h'(2) + g'(2) \cdot h(2))$. But according to the product rule, this is just the value and derivative of the product function $(g \cdot h)$ at $x = 2$.

The evaluation of the function f at $(2, 1)$ can now be viewed as successively

combining two intermediate functions g and h, starting with the identity function and finally arriving at f. At each intermediate step the derivative of the intermediate function is automatically obtained as the differential part according to the above reasoning.

An interesting side aspect is that with the search for a multiplicative inverse in equation (1.8) one has derived a rule to differentiate the function $f(x) = 1/x$ without explicitly using calculus rules.

After discussing the algebra ${}_1D_1$ and its virtues for the computation of derivatives, we now address the most general differential algebra, the structure ${}_nD_v$. It will eventually allow us to arithmetically compute partial derivatives of functions of v variables through order n.

The Structure ${}_nD_v$

We define $N(n, v)$ to be the number of monomials in v variables through order n. We will show that $N(n, v) = \frac{(n+v)!}{n!v!} = C(n + v, v)$ where $C(i, j)$ is the familiar binomial coefficient. First note that the number of monomials with exact order n equals $N(n, v - 1)$. This is true because each monomial of exact order n can be written as a monomial with one variable less times the last variable to such a power that the total power equals n. Thus we have $N(n, v) = N(n - 1, v) + N(n, v - 1)$: the number of monomials in v variables through order n equals the number of monomials of one order less plus the ones of exact order n. This recursive relation is satisfied by $C(n + v, v)$. Since also obviously $C(1 + 1, 1) = 2 = N(1, 1)$, the formula follows by induction.

Now assume that all these N monomials are arranged in a certain manner order by order. For each monomial M we call I_M the position of M according to the ordering. Conversely, with M_I we denote the Ith monomial of the ordering. Finally, for an I

with $M_I = x_1^{i_1} \cdot ... \cdot x_v^{i_v}$ we define $F_I = i_1! \cdot ... \cdot i_v!$.

We now define an addition, a scalar multiplication and a vector multiplication on R^N in the following way:

$$(a_1, ..., a_N) + (b_1, ...b_N) = (a_1 + b_1, ..., a_N + b_N) \tag{1.15}$$

$$t \cdot (a_1, ..., a_N) = (t \cdot a_1, ..., t \cdot a_N) \tag{1.16}$$

$$(a_1, ..., a_N) \cdot (b_1, ...b_N) = (c_1, ..., c_N) \tag{1.17}$$

where the coefficients c_i are defined as follows:

$$c_i = F_i \sum_{\substack{0 \le \nu, \mu \le N \\ M_\nu \cdot M_\mu = M_i}} \frac{a_\nu \cdot b_\mu}{F_\nu \cdot F_\mu} \tag{1.18}$$

To help clarify these definitions, let us look at the case of two variables and second order. In this case, we have $n = 2$ and $v = 2$. There are $N = C(2 + 2, 2) = 6$ monomials in two variables, namely

$$1, \ x, \ y, \ xx, \ xy, \ yy \tag{1.19}$$

As an example, using the ordering in (1.19), we have $I_{xy} = 5$ and $M_3 = y$. Using the ordering in (1.19), we obtain for c_1 through c_6 in equation (1.18):

$$\begin{aligned} c_1 &= a_1 \cdot b_1 \\ c_2 &= a_1 \cdot b_2 + a_2 \cdot b_1 \end{aligned}$$

$$\begin{aligned} c_3 &= a_1 \cdot b_3 + a_3 \cdot b_1 \\ c_4 &= 2 \cdot (a_1 \cdot b_4/2 + a_2 \cdot b_2 + a_4 \cdot b_1/2) \\ c_5 &= a_1 \cdot b_5 + a_2 \cdot b_3 + a_3 \cdot b_2 + a_5 \cdot b_1 \\ c_6 &= 2 \cdot (a_1 \cdot b_6/2 + a_3 \cdot b_3 + a_6 \cdot b_1/2) \end{aligned} \tag{1.20}$$

On ${}_nD_v$ we introduce a third operation ∂_i:

$$\partial_\nu(a_1, ..., a_N) = (c_1, ..., c_N) \tag{1.21}$$

with

$$c_i = \begin{cases} 0 & \text{if } M_i \text{ has order } n \\ a_{I_{(M_i \cdot x_\nu)}} & \text{else} \end{cases} \tag{1.22}$$

So ∂_ν moves the derivatives around in the vector. Suppose a vector contains the derivatives of the function f, then applying ∂_ν to it one obtains the derivatives of $\frac{\partial f}{\partial x_\nu}$ through one order less. With this third operation, ${}_nD_v$ becomes a Differential Algebra as defined in [12].

While in ${}_1D_1$, $d = (0, 1)$ was an infinitely small quantity, here we have a whole variety of infinitely small quantities that have the property that high enough powers of them vanish. We give special names to the ones in components I belonging to first order monomials, denoting them by dM_I. In the example of ${}_2D_2$, we have $dx = (0, 1, 0, 0, 0, 0)$ and $dy = (0, 0, 1, 0, 0, 0)$. It then follows from the theory of Nonstandard Analysis that instead of equation (1.10) we obtain

$$\begin{aligned} &f(x + dx, y + dy) = \\ &(f, \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial^2 f}{\partial x^2}, \frac{\partial^2 f}{\partial x \partial y}, \frac{\partial^2 f}{\partial y^2})(x, y) \end{aligned} \tag{1.23}$$

In the general case of v variables and order n, after evaluating f in the differential algebra one obtains:

$$\frac{\partial^{i_1+i_2+...+i_v} f}{\partial x_1^{i_1} \partial x_2^{i_2} ... \partial x_v^{i_v}} = c_{I_{(x_1^{i_1} \cdot .... x_v^{i_v})}} \tag{1.24}$$

where $I_{(x_1^{i_1} \cdot .... x_v^{i_v})}$ is the index of the monomial $(x_1^{i_1} \cdot ... \cdot x_v^{i_v})$, as defined in the beginning of the section.

Important Functions on Differential Algebras

In this section we will generalise standard functions like exponentials, logarithmic and trigonometric function to differential algebra. As we will see below, virtually all functions existing on a computer can be generalised in a straightforward way.

We start our discussion by noting that for any differential algebra vector of the form $(0, a_1, ..., a_N) \in {}_nD_v$, i.e. with a zero in the component belonging to the zeroth order monomial, we have the following property:

$$(0, a_1, ..., a_N)^i = (0, 0, ..., 0) \quad \text{for } i > n \tag{1.25}$$

which follows directly from the definition of the multiplication in ${}_nD_v$ defined in equation (1.17).

Let us begin our discussion of special functions with the exponential function $\exp(x)$. Assume we have to compute the exponential of a differential algebra vector that has already been created by previous operations. First we note that the functional equation $\exp(x+y) = \exp(x) \cdot \exp(y)$ also holds in Nonstandard Analysis. As we will see, this facilitates the computation of the exponential considerably.

$$\exp[(a_0, a_1, a_2, ..., a_N)] = \exp(a_0) \cdot \exp[(0, a_1, a_2, ..., a_N)]$$

$$= \exp(a_0) \cdot \sum_{i=0}^{\infty} \frac{(0, a_1, a_2, ..., a_N)^i}{i!}$$

$$= \exp(a_0) \cdot \sum_{i=0}^{n} \frac{(0, a_1, a_2, ..., a_N)^i}{i!} \qquad (1.26)$$

In the last step use has been made of equation (1.25) which entails that the sum has to be taken only through order n and thus allows the computation of the root in finitely many steps. Hence the evaluation of the real number exponential $\exp(a_0)$ which internally on a computer requires a power series summation and hence cannot be done accurately, is more subtle then the rest of the operations in differential algebra.

A logarithm of a differential algebra vector exists if and only if $a_0 > 0$. In this case one obtains

$$\begin{aligned} \log[(a_0, a_1, a_2, ..., a_N)] &= \log[a_0 \cdot (1 + (0, \frac{a_1}{a_0}, \frac{a_2}{a_0}, ... \frac{a_N}{a_0})] \\ &= (\log(a_0), 0, ..., 0) + \sum_{i=1}^{\infty} (-1)^{i+1} \frac{1}{i} (0, \frac{a_1}{a_0}, \frac{a_2}{a_0}, ..., \frac{a_N}{a_0})^i \\ &= (\log(a_0), 0, ..., 0) + \sum_{i=1}^{n} (-1)^{i+1} \frac{1}{i} (0, \frac{a_1}{a_0}, \frac{a_2}{a_0}, ..., \frac{a_N}{a_0})^i \end{aligned} \qquad (1.27)$$

Again use has been made of the fundamental property of the logarithm $\log(x \cdot y) = \log(x) + \log(y)$ which transforms directly into Nonstandard Analysis and leads to simplifications by virtue of equation (1.25).

As the last example, we will derive a formula for the root function. Even though there is a direct method to compute roots by solving a set of linear equations for the coefficients of the root, we present here a technique based on power series following

an approach similar to the exponential and logarithm. The root has the following power series expansion:

$$\sqrt{1+x} = \sum_{i=0}^{\infty}(-1)^i \frac{1 \cdot 3 \cdot ... \cdot (2i-3)}{2 \cdot 4 \cdot ... \cdot (2i)} \cdot x^i \tag{1.28}$$

Using this formula and the definitions of addition and multiplication (1.15), (1.17), one directly obtains for the square root of a differential algebra vector:

$$\begin{aligned}
& \sqrt{(a_0, a_1, a_2, ..., a_N)} \\
= \; & \sqrt{a_0} \cdot \sqrt{1 + (0, \frac{a_1}{a_0}, \frac{a_2}{a_0}, ..., \frac{a_N}{a_0})} \\
= \; & \sqrt{a_0} \cdot \sum_{i=0}^{\infty}(-1)^i \frac{1 \cdot 3 \cdot ... \cdot (2i-3)}{2 \cdot 4 \cdot ... \cdot (2i)} \cdot (0, \frac{a_1}{a_0}, \frac{a_2}{a_0}, ..., \frac{a_N}{a_0})^i \\
= \; & \sqrt{a_0} \cdot \sum_{i=0}^{n}(-1)^i \frac{1 \cdot 3 \cdot ... \cdot (2i-3)}{2 \cdot 4 \cdot ... \cdot (2i)} \cdot (0, \frac{a_1}{a_0}, \frac{a_2}{a_0}, ..., \frac{a_N}{a_0})^i
\end{aligned} \tag{1.29}$$

Using the addition theorems for sine and cosine, one obtains formulas with finite sums in a quite similar way; in general, suppose a function f has an addition theorem of the form

$$f(a+b) = g_a(b) \tag{1.30}$$

and $g_a(b)$ can be written in a power series, then by the same reasoning its differential algebraic extension is computable exactly in only finitely many steps. In practice it turns out that this can be done for all commonly supported functions in a FORTRAN computer environment.

1.2.3 Computation of Transfer Maps

Differential Algebras can be used very efficiently to compute the transfer map of particle optical systems in its Taylor series representation.

To illustrate this, let us start the discussion with a very simple example, the midplane motion in a 90^o homogenous bending magnet. Let x_i and $a_i = \sin(\alpha_i)$ denote the initial distance and scaled transverse momentum relative to the reference trajectory. Then we are interested in the values x_f and $a_f = \sin(\alpha_f)$. Since the trajectories in the magnet are circles, we can readily infer:

$$\begin{aligned} A &= R\ \sin(\alpha_i) = R\ a_i \\ B &= R\ (1 - \cos(\alpha_i)) + x_i = R\ (1 - \sqrt{1 - a_i^2}) + x_i \\ a_f &= \sin(\alpha_f) = -\frac{B}{R} \\ x_f &= A - R\ (1 - \cos(\alpha_f)) = A - R\ (1 - \sqrt{1 - a_f^2}) \end{aligned} \tag{1.31}$$

These equations allow the computation of the final coordinates x_f, a_f in terms of the initial coordinates x_i, a_i. However, taking these equations and performing all operations in differential algebra allows us to obtain all derivatives of x_f, a_f with respect to x_i, a_i. These so obtained derivatives, evaluated at $x_i = 0$, $a_i = 0$, are then the expansion coefficients of the map. For the sake of clarity, let us explicitly show how x_f and a_f are computed.

Using the ordering in (1.19) and identifying the variable a with y, we obtain using the arithmetic defined in equations (1.15), (1.16) and (1.17)

$$x_i = (0, 1, 0, 0, 0, 0)$$

$$
\begin{aligned}
a_i &= (0,0,1,0,0,0) \\
A &= (0,0,R,0,0,0) \\
B &= (0,1,0,0,0,R) \\
a_f &= (0,-\frac{1}{R},0,0,0,-1) \\
x_f &= (0,0,R,-\frac{1}{R},0,0)
\end{aligned} \tag{1.32}
$$

As a quick check, the fact that the second component of x_f is zero implies that $\frac{\partial x_f}{\partial x_i} = 0$ and hence (x,x) = 0 which is a well known property of 90^o bends.

In case an additional particle optical element is to follow this bending magnet, one does not have to start all over evaluating this new element at $x_i = (0,1,0,0,0,0)$, $a_i = (0,0,1,0,0,0)$, but one can start already with x_f and a_f of equation (1.32). This way one can save the usually quite involved concatenation process and increase performance significantly.

In the general case in which no closed solution of the problem exists, there is still a way to computationally relate the final to the initial coordinates, by numerical integration of the equations of motion. In this case, the final coordinates are still computed from the initial coordinates using standard arithmetic and functions, however the relations are more complex than in the case of the homogeneous sector.

Now performing all these operations in differential algebra automatically gives all desired derivatives of the transfer function, regardless of the form of the equations of motion. Still there are more elegant DA-based schemes for numerical integration, which are described in [3].

1.2.4 Tune Shifts and Resonance Strengths

We study the Poincare section of an accelerator lattice. In the linear approximation the particles move on ellipses, called invariant ellipses in this section. This is shown for example in [5]. The average angles in the $x - a$ and $y - b$ plane by which the particles rotate during one revolution around the accelerator ring are called the tunes of the system. They usually depend on the distance of the particle from the reference particle in $x - a$ respectively $y - b$ plane, which is called the amplitude, as well as on parameters like the chromaticity. The tunes are of prime importance for the stability analysis of a lattice, as a low order resonance between them entails instability of the particles. The tune shifts with amplitude and parameters are an outcome of the DA Normal Form algorithm described in [3], as well as the resonance strengths, which tell how sensitive the system is to a certain resonance. Given the emittance of the beam we can use the nonlinear tune shifts to calculate the tune foot print. This is the region in the two dimensional tune space which is occupied by the beam. There should be no low order resonances with high resonance strength in the tune foot print.

1.2.5 Fields and Potentials

As a significant part of this thesis will be devoted to the calculation of magnetic fields an introduction to this topic, especially with respect to the terms used later on , is in order. We will restrict ourself to systems with straight reference orbit, because the quadrupoles with which we will deal are falling in this class. Many elements with a straight reference orbit possess a certain rotational symmetry around the axis of the reference orbit, and it is most advantageous to describe the potential in cylindrical coordinates with a z-axis that coincides with the reference orbit. We first begin by expanding the r and ϕ components of the potential in Taylor and Fourier series,

respectively; the dependence on the cylindrical z coordinate, which here coincides with the particle optical coordinate s, is not expanded. So we have

$$V = \sum_{k=0}^{\infty} \sum_{l=0}^{\infty} M_{k,l}(s) \cos(l\phi + \theta_{k,l}) r^k \tag{1.33}$$

In cylindrical coordinates, the Laplacian has the form

$$\Delta V = \frac{1}{r}\frac{\partial}{\partial r}\left(\frac{r\partial V}{\partial r}\right) + \frac{1}{r^2}\frac{\partial^2 V}{\partial \phi^2} + \frac{\partial^2 V}{\partial s^2} = 0; \tag{1.34}$$

By inserting the Fourier-Taylor expansion of the potential, we obtain the following recursion relation

$$M_{l+2n,l}(s) = \frac{M_{l,l}^{(2n)}(s)}{\prod_{v=1}^{n}(l^2 - (l+2v)^2)} \tag{1.35}$$

The $M_{l,l}(s)$ are free parameters as are the $\theta_{k,l}$. The $M_{k,l}$ that cannot be obtained by the above recursion are zero. The number l is called the multipole order, as it describes how many oscillations the field will experience in one 2π sweep of ϕ. The free term $M_{l,l}(s)$ is called the multipole strength, and the term $\theta_{l,l}$ is called the multipole phase. Apparently, frequency l and radial power k are coupled : The lowest order in r that appears is l, and if the multipole strength is s-dependent, the powers $l+2$, $l+4$, ... will also appear. These terms that are induced by the s-dependence of the multipole strength $M_{l,l}(s)$ via equation (1.35) are called pseudo-multipoles. For a multipole of order l, the potential has a total of $2l$ maxima and minima, and is called a 2l-pole. Often Latin names are used for the $2l$ poles, and we have the following table: For a quadrupole the potential is quadratic, so the resulting field $\vec{B}$ is linear. Indeed, the quadrupole is the only s-independent element that leads to linear motion similar to that in glass optics, and thus has great importance. In practice, of course, s-dependence is unavoidable: the field of any particle optical element has to begin and end somewhere, and it usually does this by rising and falling gently with s, entailing s-dependence. This actually entails a crux of particle optics: even the quadrupoles,

l	**Leading Term in V**	**Name**
0	$M_{0,0}(s)\cos(\theta_{0,0})$	
1	$M_{1,1}(s)\cos(\phi+\theta_{1,1})\,r$	Dipole
2	$M_{2,2}(s)\cos(2\phi+\theta_{2,2})\,r^2$	Quadrupole
3	$M_{3,3}(s)\cos(3\phi+\theta_{3,3})\,r^3$	Sextupole
4	$M_{4,4}(s)\cos(4\phi+\theta_{4,4})\,r^4$	Octupole
5	$M_{5,5}(s)\cos(5\phi+\theta_{5,5})\,r^5$	Decapole

the "linear" elements, have nonlinear effects at their edges, requiring higher order correction. The region in which the field falls off from its value in the magnet body to zero is called the fringe field region. Chapter 4 will be devoted to studying the effect of fringe fields on the beam dynamics.

Chapter 2

The AT to COSY Converter

In order to make use of the capabilities of COSY INFINITY for dynamics studies in the LHC we first had to produce the lattice description in the COSY language. The LHC lattice is available from [19] in the optics output format of MAD version 8 [20]. This output is created by MAD using the OPTICS command giving a flat sequence of elements that can be translated to COSY easily. For this purpose we wrote the program given in figure 2. The output created by the OPTICS command is also called AT output, because it explicitly contains the position of each element. We will therefor refer to the conversion tool as the AT to COSY Converter . It is important to note the limitations of the program. First it is only designed to translate particle optical elements. This means that it is not possible to process a MAD-program that performs an analysis task, like the computation of tunes, using AT to COSY and obtain a program that does the same in COSY. Secondly we restricted the program to those elements actually occuring in the LHC lattice version 5.0. These are Drifts, Quadrupoles, Bends, Sextupoles. RF Cavities, Octupoles, Markers and Multipoles are converted to drifts of respective length. This limitation will be remedied by the SXF to COSY Converter described in the next chapter.

```
*       Michael Lindemann
*        11/24/97
*
*       The Program takes the optics output cre
ated by MAD
*       and converts it to COSY Input.
*
*       Translated are drifts, quadrupoles, ben
ds, sextupoles.
*       RfCavities, Octupoles, Markers and Mult
ipoles are
*       converted to drifts of according length
.
*


        Program m2cosy
        character text(7)*24
        double precision zahl(7)
        double precision angle,Pi
        character zeile*200
        integer a,b,n,status,max


          Pi=3.141592653589793
          max=10000

          open (UNIT=10,FILE='double2.dat',
     $     IOSTAT=status,STATUS='OLD',ERR=1000
)
          open (UNIT=11,file='double2_f77.fox',for
m='formatted',
     $    STATUS='UNKNOWN')

          write (11,'(A)') 'PROCEDURE RING;'
 10       READ(10,'(A)') zeile
          if (zeile(1:1).NE.' ') goto 10
          n=0
*         do while (n<max)
 20       n=n+1
          a=1
          do 60 j=1,7,1
*          DO While (zeile(a:a).EQ.' ' .AND. IC
HAR(zeile(b:b)) .NE. 13)
 40         a=a+1
           if (zeile(a:a).EQ.' ' .AND. ICHAR(ze
ile(a:a)) .NE. 13) goto 40
            b=a
*          DO While (zeile(b:b) .NE. ' ' .AND.
ICHAR(zeile(b:b)) .NE. 13)
 50         b=b+1
           if (zeile(b:b) .NE. ' ' .AND. ICHAR(
zeile(b:b)) .NE. 13) goto 50
           text(j)=zeile(a:b)
           a=b
 60        continue
           ! Conversion to float
           do 70 j=3,7,1
            read (text(j),'(e24.16)') zahl(j)
 70        continue
       if (text(1).EQ.'"MARKER"') then
           write (11,'(A2,A10,A3,A4,e24.16,A)') '{',
text(2),'}', 'dl ',0,';'
       else
        if (text(1).EQ.'"QUADRUPOLE"') then
         write (11,'(A2,A10,A3,A4,A8,e24.16,A)') '{'
,text(2),'}', 'mqk '
     $    ,text(4),zahl(6)/zahl(4),' 0.05;'
       else
        if (text(1).EQ.'"SEXTUPOLE"') then
         write (11,'(A2,A10,A3,A4,A8,e24.16,A)') '{'
,text(2),'}', 'msk '
     $    ,text(4),zahl(7)/zahl(4)/2,' 0.05;'
       else
        if (text(1).EQ.'"OCTUPOLE"') then
         write (11,'(A40,A10,A4)') ' { OCTUPOLE set
to drift',text(2),'}'
          write (11,'(A2,A10,A3,A4,e24.16,A)') '{',
text(2),'}', 'dl '
     $    ,zahl(4),';'
       else
        if (text(1).EQ.'"MULTIPOLE"') then
         write (11,'(A40,A10,A4)') ' { MULTIPOLE se
t to drift',text(2),'}'
          write (11,'(A2,A10,A3,A4,e24.16,A)') '{',
text(2),'}', 'dl '
     $    ,zahl(4),';'
       else
        if (text(1).EQ.'"RFCAVITY"') then
         write (11,'(A40,A10,A4)') ' { RFCAVITY set t
o drift',text(2),'}'
          write (11,'(A2,A10,A3,A4,e24.16,A)') '{',
text(2),'}', 'dl '
     $    ,zahl(4),';'
       else
        if (text(1).EQ.'"DRIFT"') then
         write (11,'(A2,A10,A3,A4,e24.16,A)') '{',
text(2),'}', 'dl '
     $    ,zahl(4),';'
       else
        if (text(1).EQ.'"RBEND"') then
         if (zahl(5).GT.0) then
            angle=zahl(5)/Pi*180
         write (11,'(A2,A10,e24.16,A3)') '{',text
(2), zahl(5), '}'
*          write (11,'(e24.16)') Pi
           write (11,'(A4,e24.16,e24.16,A6)') '
dil '
     $    ,zahl(4),angle,'0.05'
           write (11,'(e24.16,A4,e24.16,A5)') a
ngle/2,' 0'
     $    ,angle/2,' 0;'
           else
            angle=-zahl(5)/Pi*180
            write (11,'(A)') '{negative bend !}
'
         write (11,'(A2,A10,e24.16,A3)') '{',text(
2),zahl(5),'}'
*          write (11,'(e24.16)') Pi
           write (11,'(A)') 'cb;'
           write (11,'(A4,e24.16,e24.16,A6)')
'dil '
     $     ,zahl(4),angle,'0.05'
           write (11,'(e24.16,A4,e24.16,A5)')
angle/2
     $    ,' 0',angle/2,' 0;'
           write (11,'(A)') 'cb;'
           endif
         else
          write (11,'(A30,A10,A1)') ' { warning
: unknown element',
     $    text(1),'}'
           print *,' warning : unknown element !',
text(1)
           max=max+1
         end if
         end if
         end if
         end if
         end if
         end if
         end if
        end if
        read (10,'(A)', IOSTAT=status,end=1000
) zeile
        if (n.LT.max) goto 20
 1000    print *
        if (status.EQ.-1) then
         print *, ' complete file converted '
         endif
        print *, max
        write(11,'(A)') ' endprocedure ; '
        close (10)
        close (11)
        end
```

Figure 2.1: The AT to COSY converter

Chapter 3

The SXF to COSY Converter

3.1 Introduction

At the "Berkeley National Laboratory Workshop on the Unified Accelerator Libraries and US-LHC software" a new machine file format was proposed [30]. This format is meant to be a general lattice description language and is intended to facilitate the cooperation between different groups and the comparison of results obtained with different codes. The language was named Standard Exchange Format and abbreviated as SXF. The language specifications were developed by F.Pilat et al.. The SXF to COSY converter can be used online at our web page [18] where the current SXF specifications are also available.

3.2 The Converter

3.2.1 General Features

The SXF to COSY Converter is written in PERL. The source code is given in Appendix A. As the SXF language is still evolving, it is very likely that the converter will have to be adapted to future changes of the language standard. This made it very important to optimise the code for ease of reading and modifying.

3.2.2 Description

In order to facilitate the understanding and changing of the program in the future we give a brief description of how the translation is done. We will especially stress the points where differences between COSY and SXF make the translation less transparent.

The AT attribute

The AT attribute in SXF gives the position of the middle of the element. As COSY does not allow an AT command this is emulated by drifts, keeping in mind that the drift should only extend to the beginning of the element but not to the middle of it.

The bends

There are two bends allowed in SXF, the rectangular bend RBEND and the sector bend SBEND. Because they are in most cases equivalent the use of rectangular bends is deprecated. As described in the MAD-Manual [20] and in the MAD Physics Manual [21] these two elements only differ by the local reference system in which the entrance and exit angles are measured. This results in the addition of half the bending angle to entrance and exit angles for the rectangular bend. By doing this both rectangular and sector bends can be translated to the COSY command DIL. The direction of the bend is determined by the sign of the bending angles. For positive angles the direction is clockwise. The bending direction is changed for negative angles using COSY's CB command.

Another important point concerning the rectangular bends is that there are two ways of specifying their length. One can either use l, the end to end length of the element, or arc, the length along the circular orbit. arc and l are redundant since

they satisfy

$$arc = l\frac{kl[0]/2}{sin(kl[0]/2)} \tag{3.1}$$

where $kl[0]$ is the bending angle. If l is given it is converted to arc using this formula. If l and arc are given, l is ignored.

Multipoles

SXF allows dipoles, quadrupoles, sextupoles and octupoles as well as a general multipole. Each of these elements can contain a list of multipole strengths with not only the strength corresponding to the name of the element nonzero. This means that one might even specify a quadrupole by calling the element dipole in SXF with zero dipole strength and nonzero quadrupole strength. In this sense the element specifier is redundant and the SXF to COSY converter translates all multipole elements to a general multipole in COSY. The information given in the element specifier is preserved as a comment. Following the convention that dipoles are usually described as bends, a dipole component in multipoles is not supported at this point. If a dipole is specified by using a multipole an error message is issued.

Chapter 4

Beam Dynamics Studies

4.1 Introduction

Accelerator lattices are usually described by the position, length and field strength in the main body of their elements. The field of the magnets is considered to change from zero to the value in the main field at the magnet entrance and drop again to zero at the magnet exit. Although this approximation is widely used in beam physics it is very unrealistic. Using COSY INFINITY it is possible to take into account the effect of the exact shape of the magnetic field at the ends of the magnet. To do this the magnet is split into a main section in which the field is independent of the particle optical coordinate s, and an s-dependent element representing the fringe field. The fringe field map, which has finite length, is composed with two negative drifts, to produce a zero-length insertion [14].

4.2 Magnetic Field Data

4.2.1 Introduction

We have seen in section 1.2.5 that in the curl and divergence free region the most general form of the potential in cylindrical coordinates can be written as

$$V = \sum_{k=0}^{\infty}\sum_{l=0}^{\infty} M_{k,l}(s)\cos(l\phi + \theta_{k,l})r^k \tag{4.1}$$

with

$$M_{l+2n,l}(s) = \frac{M_{l,l}^{(2n)}(s)}{\prod_{\upsilon=1}^{n}(l^2-(l+2\upsilon)^2)} \tag{4.2}$$

The $M_{l,l}(s)$ are free parameters as are the $\theta_{l,l}$. In the magnetic case $\theta_{l,l} = -\pi/2$ is chosen by convention. The $M_{k,l}$ that cannot be obtained by the above recursion are zero. For ease of notation we define

$$\begin{aligned} B_l(s) &= M_{l,l}(s)\cos(\theta_{l,l}) \\ A_l(s) &= M_{l,l}(s)\sin(\theta_{l,l}) \end{aligned} \tag{4.3}$$

Using these equations we can rewrite 4.1 as

$$\begin{aligned} V = {} & \sin(2\phi)r^2(B_2(s) + \frac{B_2''(s)}{a}r^2 + \frac{B_2^{(4)}}{b}r^4 + \ldots) \\ & - \cos(2\phi)r^2(A_2(s) + \frac{A_2''(s)}{a}r^2 + \frac{A_2^{(4)}}{b}r^4 + \ldots) \\ & + \sin(6\phi)r^6(B_6(s) + \frac{B_6''(s)}{c}r^2 + \frac{B_6^{(4)}}{d}r^4 + \ldots) \\ & - \cos(6\phi)r^6(A_6(s) + \frac{A_6''(s)}{c}r^2 + \frac{A_6^{(4)}}{d}r^4 + \ldots) \\ & + \sin(10\phi)r^{10}(B_{10}(s) + \frac{B_{10}''(s)}{e}r^2 + \frac{B_{10}^{(4)}}{f}r^4 + \ldots) \\ & - \cos(10\phi)r^{10}(A_{10}(s) + \frac{A_{10}''(s)}{e}r^2 + \frac{A_{10}^{(4)}}{f}r^4 + \ldots) + \ldots \end{aligned}$$

where the coefficients $a, b, \ldots, f$ can be calculated from 4.2. So we can calculate the components of the magnetic field

$$\begin{aligned} B_r &= -\frac{\partial V}{\partial r} \\ &= -2\sin(2\phi)r(B_2(s) + \frac{2B_2''(s)}{a}r^2 + 3\frac{B_2^{(4)}}{b}r^4 + \ldots) \\ &\quad +2\cos(2\phi)r(A_2(s) + \frac{2A_2''(s)}{a}r^2 + 3\frac{A_2^{(4)}}{b}r^4 + \ldots) + \ldots \end{aligned}$$

$$\begin{aligned} B_\phi &= -\frac{1}{r}\frac{\partial V}{\partial \phi} \\ &= 2\cos(2\phi)r(B_2(s) + \frac{B_2''(s)}{a}r^2 + \frac{B_2^{(4)}}{b}r^4 + \ldots) \\ &\quad -2\sin(2\phi)r(A_2(s) + \frac{A_2''(s)}{a}r^2 + \frac{A_2^{(4)}}{b}r^4 + \ldots) + \ldots \end{aligned}$$

G. Sabbi defines the quantities $\bar{b}_n$ in [23]. In the notation introduced above $\bar{b}_2$ and $\bar{b}_6$ are given by

$$\begin{aligned} \bar{b}_2(s) &= (B_2(s) + 2\frac{B_2''}{a}r^2 + 3\frac{B_2^{(4)}(s)}{b}r^4 + \ldots)\frac{1}{B_2(s_0)} \\ \bar{b}_6(s) &= (3B_6(s) + 4\frac{B_6''}{c}r^2 + 5\frac{B_6^{(4)}(s)}{d}r^4 + \ldots)\frac{1}{B_2(s_0)}(\frac{r}{r_0})^4 \end{aligned} \tag{4.4}$$

where s_0 is chosen sufficiently inside the magnet, such that the derivatives of $B_n(s)$ with respect to s vanish. Taking a look at the definition of B_n in (4.3) reveals that $\bar{b}_2$ can be written as

$$\bar{b}_2(s) = (M_{2,2}(s) + 2\frac{M_{2,2}''}{a}r^2 + 3\frac{M_{2,2}^{(4)}(s)}{b}r^4 + \ldots)\frac{1}{M_{2,2}(s_0)}$$

$$\bar{b}_6(s) = (3M_{6,6}(s) + 4\frac{M''_{6,6}}{c}r^2 + 5\frac{M^{(4)}_{6,6}(s)}{d}r^4 + \ldots)\frac{1}{M_{2,2}(s_0)}(\frac{r}{r_0})^4 \qquad (4.5)$$

4.2.2 Fitting

We model $M_{2,2}(s)$ such that equation (4.5) is approximately satisfied for the given radii $r = 5, 10, 20, 30$ mm. In order to calculate $\bar{b}_2$ from $M_{2,2}$ via (4.5) we calculate the out-of-plane expansion of the field by the recursion relation (4.2). This is done in COSY INFINITY using the differential algebraic approach. As the model class we choose Enge functions with 6 parameters namely

$$M_{2,2}(s) = \frac{1}{1 + \exp\left(a_1 + a_2(\frac{s}{d}) + a_3(\frac{s}{d})^2 + \ldots + a_6(\frac{s}{d})^5\right)} \qquad (4.6)$$

where s is the Cartesian distance to the field boundary. The quantity d is the full aperture, respectively twice the radius, of the quadrupole. For the fitting procedure we use the nonlinear optimisers implemented in COSY INFINITY. It has proven useful to obtain the initial values for the Enge coefficients by calculating coefficients such that the resulting Enge function passes exactly through six data points. To do this we calculate the solution of a system of six linear equations.

In the optimisation process we shift $M_{2,2}(s)$ such that the effective field boundary coincides with the origin. This means we have to satisfy the equation

$$\int_{s_i}^{0} 1 - M_{2,2}(s)\, ds = \int_{0}^{s_f} M_{2,2}(s)\, ds \qquad (4.7)$$

where s_i is sufficiently inside the magnet, such that $M_{2,2}(s_i) \approx 1$, and s_f is sufficiently outside the magnet, such that $M_{2,2}(s_f) \approx 0$.

In order to access the quality of the fit we compare in figure 4.1 the B_x component of the field in the return end of the magnet as calculated by COSY with the $\bar{b}_2$ as

calculated by ROXIE for different radii. The same comparison is given for the lead end in figure 4.2. This comparison is valid , because the field generated by the $M_{2,2}(s)$ via equation (4.2) is a pure quadrupole field. In the figures 4.1 and 4.2 the ROXIE values are given by the line while the result of the COSY calculation is given by dots. In each case the origin of the z-axis is chosen as defined in [29]. The z-axis is directed outwards from the magnet body. The iron yoke starts at $z = -15$ cm for the lead end, at $z = -5$ cm for the return end. The vertical axis gives the magnetic field in units of $10^4/\bar{b}_2$, where $\bar{b}_2$ is the quadrupole strength in the main body at the given radius. The plots show clearly that for the region up to $r = 20$ mm the fitting agrees very well with the simulated measurements. This is reassuring because the beam stays within approximately $r = 17$ mm . On the other hand it is obvious from the plot for $r = 30$ mm that this method does not work reliably any more for larger radii, especially because the rather peculiar shape of the fringe field at $r = 30$ can not be modelled accurately by an Enge function.

Nevertheless there is a way of calculating the real multipole content of the field more accurately using the differential algebraic approach which will be described in chapter 5.

4.3 Lattice Description

In our analysis we use the LHC lattice model Version 5.0, which is available from [19]. We used the AT to COSY converter to translate the lattice description given in the @-output format of MAD 8.0 to COSY language.

In the present study the ring is subdivided in three regions, the two inner triplets left and right of the interaction point 5, for which the detailed field data described above is available, and the rest of the ring. The layout of the triplets, which are mirror-

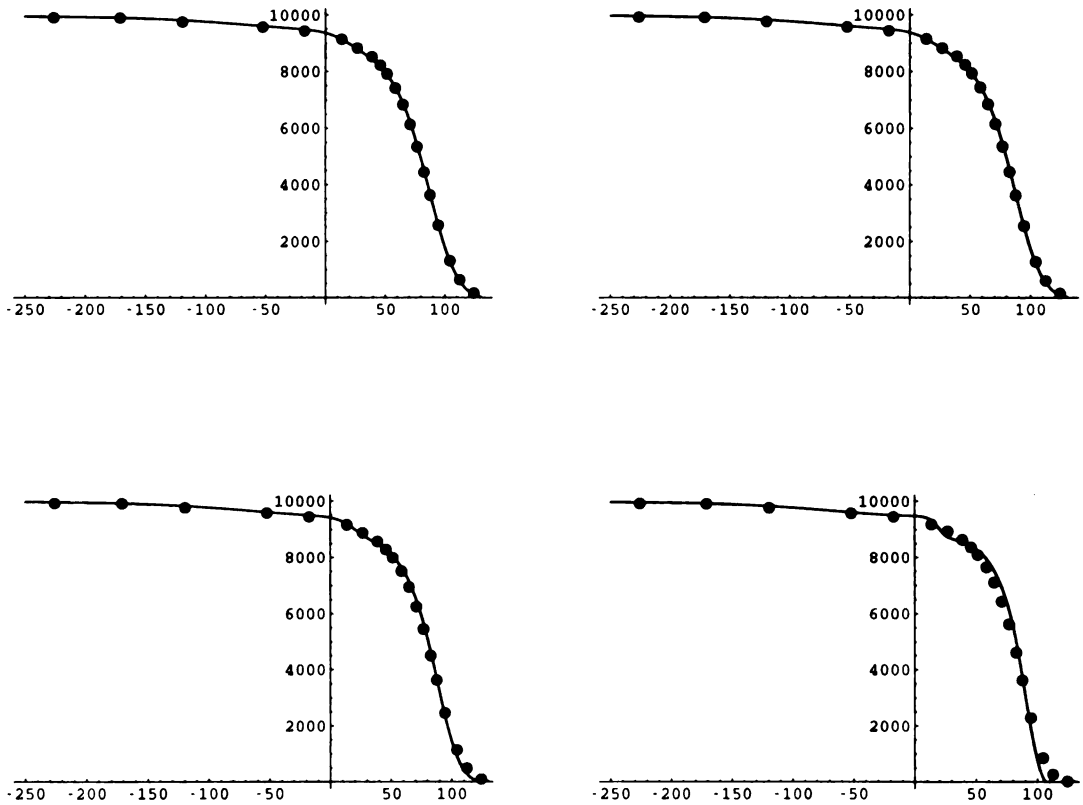

Figure 4.1: Comparison of the quadrupole component of the field in the return end as computed by COSY and ROXIE at different radii. From top left to bottom right: $r = 5$ mm, $r = 10$ mm, $r = 20$ mm and $r = 30$ mm.

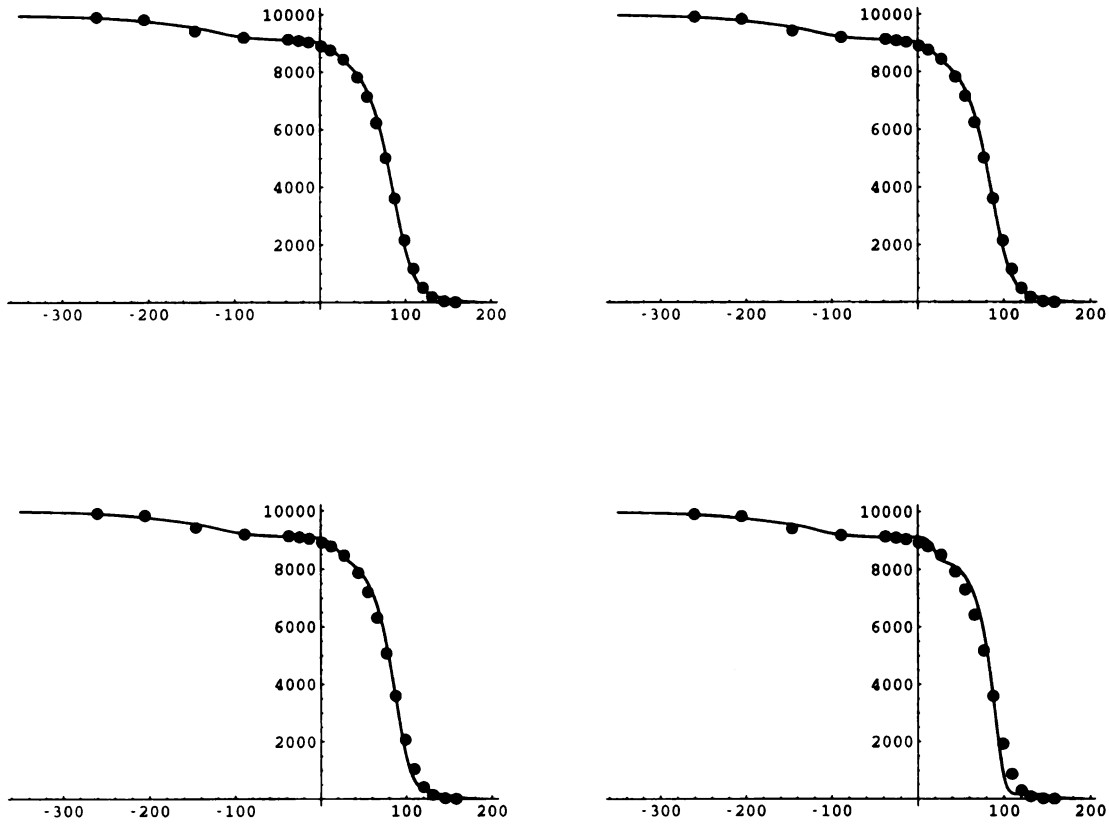

Figure 4.2: Comparison of the quadrupole component of the field in the lead end as computed by COSY and ROXIE at different radii. From top left to bottom right: $r = 5$ mm, $r = 10$ mm, $r = 20$ mm and $r = 30$ mm.

symmetric with respect to the interaction point, is shown in figure 4.3. The High Gradient Quadrupoles are denoted by Q. The elements between them are multipole correctors. For details about the LHC naming convention refer to [22].

We study the seventh order map for the triplets, with and without fringe fields, while the rest of the ring is treated in its linear approximation. Using this approach we can study the nonlinear effects of the fringe fields in the triplets, which is the most critical part of the lattice, as explained earlier. There are no main field errors considered.

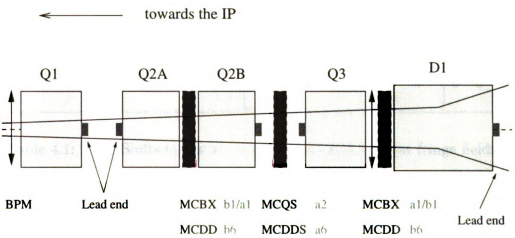

Figure 4.3: Layout of the left low-β triplet.

4.4 Results

4.4.1 Tune Shifts

We computed the amplitude dependent tune shifts with and without fringe fields. The result is given in table 4.1 for the $x-a$ plane and in table 4.2 for the $y-b$ plane.

The tables show two effects. First, the nonlinear tune shifts increase considerably. Secondly, the center tune decreases by $6 \cdot 10^{-3}$. This can be corrected using the trim quadrupoles in the LHC lattice. But even after refitting the center tune to its original value, the nonlinear tune shifts are considerably bigger with fringe fields than without fringe fields. In fact they are still of the same order of magnitude as without refitting the tune.

without fringe fields	with fringe fields		order	exponents
	original	center tune refitted		
.310000	.303921	.310000	0	0 0 0 0
58.5018	592.429	610.796	2	2 0 0 0
20.45303	873.057	902.598	2	0 0 0 2
8352.403	1889188	2191916	4	4 0 0 0
-3360.344	-13279249	-13536156	4	2 0 2 0
9216.0682	11570624	12740611	4	0 0 4 0
13692934	-399181479286	-422995493944	6	6 0 0 0
-51336671	88048290859	96863534946	6	4 0 2 0
124626604	-1035625378625	-1159771282208	6	2 0 4 0
-15542851	1394274127638	1515548018372	6	0 0 6 0

Table 4.1: Tune Shifts in the $x - a$ plane with and without fringe fields.

without fringe fields	with fringe fields		order	exponents
	original	center tune refitted		
.320000	.313929	.320000	0	0 0 0 0
20.4538	873.058	902.600	2	2 0 0 0
58.5018	589.143	610.797	2	0 0 0 2
-1680.1	-7496999	-7667610	4	4 0 0 0
18432.1	24842663	272712816	4	2 0 2 0
11342.9	2181737	2516249	4	0 0 4 0
-17112223	1289593405561	1373994850952	6	6 0 0 0
124626604	-11277831999539	-1259605227516	6	4 0 2 0
-46628554	4712527272105	5749819744288	6	2 0 4 0
13545598	-399572595609	-427111891097	6	0 0 6 0

Table 4.2: Tune shifts in the $y - b$ plane with and without fringe fields.

The tune footprints shown in figure 4.4 clearly demonstrate the significance of the fringe field effects.

Compared to the previous study done by F. Méot et. al. [26], the center tune change shown here is about 20 times larger. Yet we notice the fact that the two studies use different sets of Enge coefficients. Specifically, in the Méot study the detailed shape of the fringe field was not considered but assumed to be similar to that of other standard quadrupoles. In reality, however, the end fields used here look much worse.

As the interaction region 1 has the same design as the interaction region 5, one can apply the fringe fields to the triplets in the interaction region 1 as well. The result is that as expected the center tune shift grows by a factor of two, while the nonlinear tune shifts are still of the same order of magnitude. In the tables 4.3 and 4.4 we compare the tune shifts for the $x - a$ and $y - b$ plane respectively.

without fringe fields	with fringe fields		order	exponents
	in IR 5	in IR5 and IR1		
.310000	.303921	.297997	0	0 0 0 0
58.5018	592.429	1127.460	2	2 0 0 0
20.45303	873.057	1721.358	2	0 0 0 2
8352.403	1889188	6319767	4	4 0 0 0
-3360.344	-13279249	-48466669	4	2 0 2 0
9216.0682	11570624	32720546	4	0 0 4 0
13692934	-399181479286	-624020378914	6	6 0 0 0
-51336671	88048290859	4464097212016	6	4 0 2 0
124626604	-1035625378625	-8236728752799	6	2 0 4 0
-15542851	1394274127638	3983026534205	6	0 0 6 0

Table 4.3: Tune shifts in the $x - a$ plane with fringe fields in IR 5 and in both IR1 and IR5

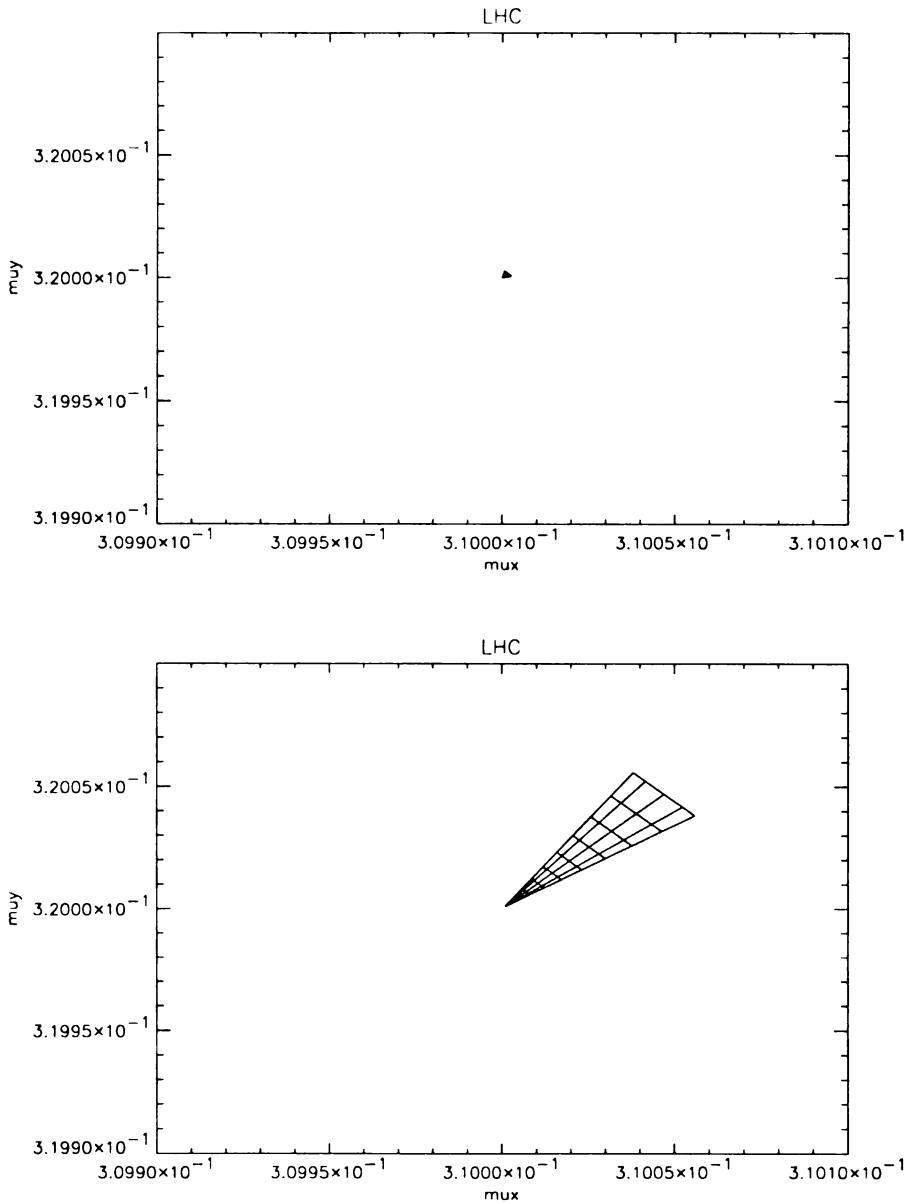

Figure 4.4: Tune footprint without fringe fields (top) and with fringe fields (bottom).

without fringe fields	with fringe fields		order	exponents
	in IR 5	in IR5 and IR1		
.320000	.313929	.307799	0	0 0 0 0
20.4538	873.058	1721.360	2	2 0 0 0
58.5018	589.143	1203.404	2	0 0 0 2
-1680.1	-7496999	-25882333	4	4 0 0 0
18432.1	24842663	68831273	4	2 0 2 0
11342.9	2181737	1386369	4	0 0 4 0
-17112223	1289593405561	3872032968262	6	6 0 0 0
124626604	-11277831999539	-8509355582662	6	4 0 2 0
-46628554	4712527272105	4516085446489	6	2 0 4 0
13545598	-399572595609	-791311504674	6	0 0 6 0

Table 4.4: Tune shifts in the $y - b$ plane with fringe fields in IR 5 and in both IR1 and IR5

4.4.2 Resonance Analysis

In order to further investigate the changes in the dynamical behavior of the system we study the resonance strength. The detailed result is given in the tables 4.5 and 4.6. From table 4.5 one can see the significant increase in the resonance strengths when we take into account the effect of fringe fields. From table 4.6 one can see that refitting the tune to its nominal value does not affect the resonances very much. To compare the resonance strengths with and without fringe field, we calculate the logarithm of the average absolute value of the resonance strengths for every order. The result is given in figure 4.5. It shows clearly that the resonance strength increases by at least one order of magnitude on average, and even more for higher orders.

Resonance Strength								
without fringe fields		with fringe fields		order	exponents			
-3.802e+01	1.268e+02	-3.289e+02	1.484e+03	4	4	0	0	0
9.937e+01	1.508e+02	8.412e+02	1.395e+03	4	3	1	0	0
3.312e+01	-5.028e+01	2.804e+02	-4.650e+02	4	1	3	0	0
-3.135e-09	1.163e+02	1.054e-01	5.760e+03	4	2	0	2	0
1.472e+00	9.386e+01	2.448e+01	3.672e+03	4	1	1	2	0
3.245e+01	1.020e+03	5.842e+02	4.355e+04	4	0	2	2	0
-1.243e+00	7.948e+01	-2.041e+01	3.076e+03	4	2	0	1	1
-4.143e-01	-2.649e+01	-6.785e+00	-1.025e+03	4	0	2	1	1
3.245e+01	-1.020e+03	5.842e+02	-4.355e+04	4	2	0	0	2
4.906e-01	-3.129e+01	8.141e+00	-1.224e+03	4	1	1	0	2
3.709e+01	9.772e+01	3.175e+02	1.133e+03	4	0	0	4	0
-9.893e+01	1.783e+02	-8.327e+02	1.655e+03	4	0	0	3	1
-3.298e+01	-5.942e+01	-2.776e+02	-5.518e+02	4	0	0	1	3
5.926e-07	-1.175e-07	4.011e-05	-1.689e-05	5	5	0	0	0
-1.214e-05	-1.125e-05	-9.245e-04	-8.424e-04	5	4	1	0	0
-7.746e-07	-1.868e-06	-9.106e-05	-1.858e-04	5	3	2	0	0
-1.245e-06	-5.164e-07	-1.239e-04	-6.071e-05	5	2	3	0	0
-2.812e-06	-3.036e-06	-2.106e-04	-2.311e-04	5	1	4	0	0
4.018e-07	2.360e-07	1.534e-04	1.032e-04	5	3	0	2	0
-9.641e-06	-9.068e-06	-4.818e-03	-4.673e-03	5	2	1	2	0
-3.814e-07	-6.630e-07	-2.193e-04	-3.042e-04	5	1	2	2	0
-3.374e-07	-2.458e-07	-9.460e-05	-6.842e-05	5	0	3	2	0
-5.320e-06	-5.630e-06	-2.368e-03	-2.453e-03	5	3	0	1	1
-5.293e-07	-4.194e-07	-5.395e-04	-5.105e-04	5	2	1	1	1
-2.796e-07	-3.529e-07	-3.347e-04	-3.536e-04	5	1	2	1	1
-1.407e-06	-1.330e-06	-5.903e-04	-5.702e-04	5	0	3	1	1
-3.686e-07	-5.061e-07	-1.128e-04	-1.519e-04	5	3	0	0	2
-4.420e-07	-2.543e-07	-2.125e-04	-1.569e-04	5	2	1	0	2
-2.267e-06	-2.410e-06	-1.198e-03	-1.233e-03	5	1	2	0	2
7.545e-08	4.084e-07	4.945e-05	1.482e-04	5	1	0	4	0
-4.701e-07	-3.844e-07	-1.049e-03	-1.000e-03	5	0	1	4	0
-9.907e-06	-1.050e-05	-3.498e-03	-3.690e-03	5	1	0	3	1
-2.446e-07	-1.392e-07	-1.858e-04	-1.567e-04	5	0	1	3	1
-9.698e-07	-4.701e-07	-4.024e-04	-2.629e-04	5	1	0	2	2
-3.134e-07	-6.466e-07	-1.809e-04	-2.745e-04	5	0	1	2	2
-9.280e-08	-1.631e-07	-9.970e-05	-1.183e-04	5	1	0	1	3
-2.626e-06	-2.477e-06	-9.322e-04	-8.844e-04	5	0	1	1	3
-9.609e-08	-1.175e-07	-2.364e-04	-2.479e-04	5	1	0	0	4

Table 4.5: Resonance strengths with and without fringe fields.

Resonance Strength								
without refitting		with refitting		order	exponents			
-3.289e+02	1.484e+03	-3.320e+02	1.304e+03	4	4	0	0	0
8.412e+02	1.395e+03	8.648e+02	1.604e+03	4	3	1	0	0
2.804e+02	-4.650e+02	2.883e+02	-5.347e+02	4	1	3	0	0
1.054e-01	5.760e+03	-1.223e-05	5.080e+03	4	2	0	2	0
2.448e+01	3.672e+03	2.742e+01	4.182e+03	4	1	1	2	0
5.842e+02	4.355e+04	5.947e+02	4.506e+04	4	0	2	2	0
-2.041e+01	3.076e+03	-2.323e+01	3.547e+03	4	2	0	1	1
-6.785e+00	-1.025e+03	-7.722e+00	-1.182e+03	4	0	2	1	1
5.842e+02	-4.355e+04	5.948e+02	-4.506e+04	4	2	0	0	2
8.141e+00	-1.224e+03	9.118e+00	-1.394e+03	4	1	1	0	2
3.175e+02	1.133e+03	3.236e+02	1.000e+03	4	0	0	4	0
-8.327e+02	1.655e+03	-8.608e+02	1.891e+03	4	0	0	3	1
-2.776e+02	-5.518e+02	-2.869e+02	-6.302e+02	4	0	0	1	3
4.011e-05	-1.689e-05	6.353e-05	-5.748e-06	5	5	0	0	0
-9.245e-04	-8.424e-04	-1.488e-03	-1.395e-03	5	4	1	0	0
-9.106e-05	-1.858e-04	-1.179e-04	-2.465e-04	5	3	2	0	0
-1.239e-04	-6.071e-05	-1.643e-04	-7.860e-05	5	2	3	0	0
-2.106e-04	-2.311e-04	-3.488e-04	-3.721e-04	5	1	4	0	0
1.534e-04	1.032e-04	3.123e-04	2.462e-04	5	3	0	2	0
-4.818e-03	-4.673e-03	-9.290e-03	-9.078e-03	5	2	1	2	0
-2.193e-04	-3.042e-04	-3.226e-04	-4.362e-04	5	1	2	2	0
-9.460e-05	-6.842e-05	-1.281e-04	-9.324e-05	5	0	3	2	0
-2.368e-03	-2.453e-03	-4.090e-03	-4.222e-03	5	3	0	1	1
-5.395e-04	-5.105e-04	-7.859e-04	-7.599e-04	5	2	1	1	1
-3.347e-04	-3.536e-04	-5.010e-04	-5.180e-04	5	1	2	1	1
-5.903e-04	-5.702e-04	-1.026e-03	-9.945e-04	5	0	3	1	1
-1.128e-04	-1.519e-04	-1.498e-04	-2.018e-04	5	3	0	0	2
-2.125e-04	-1.569e-04	-3.001e-04	-2.255e-04	5	2	1	0	2
-1.198e-03	-1.233e-03	-2.311e-03	-2.363e-03	5	1	2	0	2
4.945e-05	1.482e-04	1.301e-04	2.621e-04	5	1	0	4	0
-1.049e-03	-1.000e-03	-2.432e-03	-2.339e-03	5	0	1	4	0
-3.498e-03	-3.690e-03	-6.874e-03	-7.122e-03	5	1	0	3	1
-1.858e-04	-1.567e-04	-2.805e-04	-2.271e-04	5	0	1	3	1
-4.024e-04	-2.629e-04	-5.897e-04	-3.869e-04	5	1	0	2	2
-1.809e-04	-2.745e-04	-2.634e-04	-3.992e-04	5	0	1	2	2
-9.970e-05	-1.183e-04	-1.467e-04	-1.816e-04	5	1	0	1	3
-9.322e-04	-8.844e-04	-1.794e-03	-1.732e-03	5	0	1	1	3
-2.364e-04	-2.479e-04	-5.623e-04	-5.844e-04	5	1	0	0	4

Table 4.6: Resonance strengths with and without refitting the linear tune.

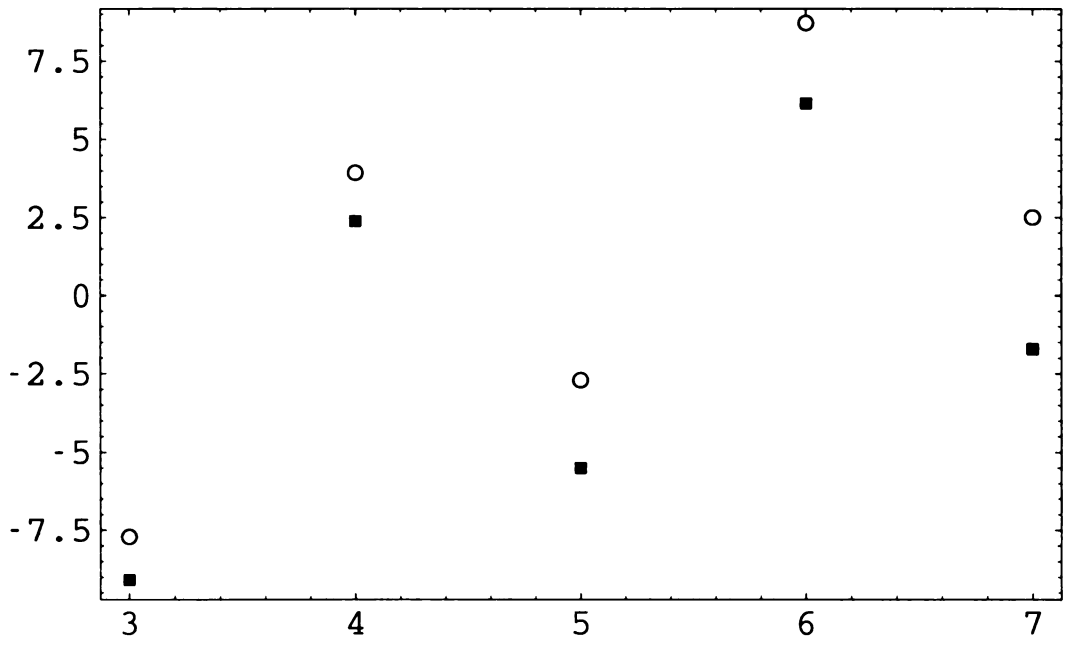

Figure 4.5: Average logarithmic resonance strength for orders 3 to 7. The values are given by squares for the case without fringe fields and circles for the case with fringe fields.

Chapter 5

Differential Algebraic Field Calculation

5.1 Introduction

So far the method of treating the fringe fields has one important shortcoming. The real multipoles, which we have to know for for beam dynamics studies as they permit the calculation of the field at any given point, have to be extracted from the available data, which is in fact a sum of real and pseudo multipoles. This extraction can not be achieved analytically but has to be done by fitting. This problem is remedied by calculating the magnetic field in COSY using the DA method, which allows an analytical scheme for the extraction of multipoles that is exact to machine precision.

5.2 Magnetic Field Calculation by ROXIE

5.2.1 Introduction

The magnetic field calculations for the LHC High Gradient Quadrupoles presented in [23] and [24] are performed by ROXIE [25]. ROXIE uses a set of line currents from the magnet model, specified by their starting and ending point in 3 dimensions and the current they carry. The magnet is subdivided into two sections, the straight

section and the end field. The end of the straight section is given by the end of the iron yoke. This is also the origin of the coordinate system used in ROXIE. The end region extends from the origin to positive z values.

5.2.2 Image Currents

The high gradient quadrupole magnets are surrounded by an iron yoke with a cylindrical inner bore. It serves two purposes: the fringe field outside the coil is greatly reduced and the field on the beam axis is enhanced by 10-20 percent.

The influence of the iron yoke is analysed using the method of image currents assuming that the iron is not saturated and the permeability μ is uniform. Then for a current I inside a hollow iron cylinder of radius R the effect of the iron on the inner field is equivalent to that of an image current I', located at the radius r' :

$$I' = \frac{\mu - 1}{\mu + 1} I \tag{5.1}$$

$$r' = \frac{R^2}{r} \tag{5.2}$$

The current I' is parallel to I. It thus increases the inner field.

It is important to notice that ROXIE calculates image currents only for the straight section.

5.3 Magnetic Field Calculation in COSY

5.3.1 Introduction

G. Sabbi has implemented an output format in ROXIE that writes the positions and strength of all line currents, including the straight, image and end currents to a file. We use this information to calculate the magnetic field in COSY.

5.3.2 Line Currents

From the Biot-Savart law we can calculate the magnetic field contribution of an infinitesimally short line current

$$dB = \mu_0 I \frac{\vec{dl} \times \vec{r}}{4\pi r^3} \tag{5.3}$$

In order to calculate the magnetic field of a line current of finite length we have to integrate along the line, which is for that purpose parametrised by

$$\vec{r(s)} = \vec{r}_s + s(\vec{r}_e - \vec{r}_s) \tag{5.4}$$

where r_s and r_i are giving start and end of the line current in three dimensions. Then the magnetic field of a line current is given by

$$B = kI \int_0^1 \frac{\vec{dl}(s) \times \vec{r}(s)}{|r|^3} ds \tag{5.5}$$

using the definition $k = \mu_0/4\pi = 10^{-7}$. As we are integrating along a straight line, $\vec{\Delta l} = \vec{dl}(s)$ is constant. It is $\vec{\Delta l} = \vec{r}_e - \vec{r}_s$. The remaining integral is easily solved by substituting (5.4) into (5.5).

$$\begin{aligned} B &= kI \int_0^1 \frac{\vec{\Delta l} \times \vec{r}(s)}{|\vec{r}(s)|^3} ds \\ &= kI\vec{\Delta l} \times \int_0^1 \frac{\vec{r}(s)}{|\vec{r}(s)|^3} ds \\ &= kI\vec{\Delta l} \times \int_0^1 \frac{\vec{r}_s + s\vec{\Delta l}}{|\vec{r}_s + s(\vec{r}_e - \vec{r}_s)|^3} ds \\ &= kI\vec{\Delta l} \times \vec{r}_s \int_0^1 \frac{1}{\left|\vec{r}_s + s\vec{\Delta l}\right|^3} ds \end{aligned}$$

For ease of notation we define

$$a = |\vec{r}_s|^2 \,,\; b = 2\left(\vec{r}_s \cdot \vec{\Delta l}\right) \,,\; c = \left|\vec{\Delta l}\right|^2 \tag{5.6}$$

Then the magnetic field is given by

$$B = kI\vec{\Delta l} \times \vec{r}_s \left(\frac{2b}{\sqrt{a}(b^2 - 4ac)} - \frac{2(b+2c)}{\sqrt{a+b+c}(b^2-4ac)} \right) \tag{5.7}$$

The actual implementation of this formula is subtle. We observe for example, that in the case $\vec{r}_s \parallel \vec{\Delta l}$ both the denominator and the numerator in (5.7) vanish. Calculating the limit $\theta \to 0$, where θ denotes the angle between $\vec{r}_s$ and $\vec{\Delta l}$, analytically, shows that in fact for this case the constant part of the field vanishes. Nevertheless this is problematic for high order calculations, as equation (5.7) becomes numerically unstable for small θ. More numerical issues related to the application of the Biot-Savart law are discussed in [31] and [32].

5.3.3 Implementation in COSY

Using the formula derived in the previous section and evaluating it in DA using COSY we are able to calculate the Taylor expansion of the magnetic field produced by a line current distribution. We implemented the formula in a new COSY function LINEFLD which calculates the y-component of the magnetic field in the midplane. LINEFLD has to be called with two parameters, the x and z coordinate of the position in the midplane where the field should be calculated. It reads the line current distribution from the file LCD.DAT in the COSY directory. The format for LCD.DAT is very simple. It contains only numbers, one number per line. The first line specifies the total number of currents in this file. After that 7 lines always specify one line current in the format $x_s, y_s, z_s, \Delta l_x, \Delta l_y, \Delta l_z, I$, where I is the current . It is important to notice that the units meter and ampere have to be used.

The results of the function LINEFLD have been tested against ROXIE and were found to be correct. We checked both the the absolute field value at a given point, including the x- and z-component that are not returned by LINEFLD, and the field gradient. The gradient is a natural result of the computation of the field using differential algebra, namely the term in the expansion of the B_y that is linear in x. During the comparison a typo was discovered in [23]. The correct value for the gradient in the magnet body is 18.22 T/m/kA and not 0.1822 T/m/kA as given in the paper. This has been acknowledged by G. Sabbi.

In future the detailed field information shall be used for beam dynamics simulations. I will outline in the following section how this can be done in COSY.

Creating a new Element

The first way is to create a new partical optical element. An element can be specified by giving an explicit formula. In out case the formula is in fact a subroutine that reads the positions and strengths of the line currents from a file and calculates the resulting field. As the implementation of new elements in COSY is not documented in the manual, I will give a brief prescription here.

All of the following changes are to be made in COSY.FOX. First you should create a new procedure with the name of your new elements. In this case we call it LCD for **L**ine **C**urrent **D**istribution.

```
PROCEDURE LCD L D ; { Line Current Distribution }
NSDP := -6 ; LOFF := 2 ; CE := 'LCD' ;
SDELE 0 L L/500 L/10 L D ; ENDPROCEDURE ;
```

The new element has two parameters, length L and aperture D. The variable $NSDP$ determines which field formula is taken to calculate the matrix of the **S**-

Dependent **ELE**ment *SDELE*. $LOFF = 2$ indicates that the formula for the field in the midplane is given, while $LOFF = 1$ means that we supply the field on the optical axis. These formulas are specified in the procedure POTXZ. We create a new entry for $NSDP$ corresponding to the value we chose in LCD.

```
ELSEIF NSDP=-6 ; { LINE CURRENT DISTRIBUTION }
BY:=LINEFLD(X,Z) ;
```

General Element

The second way of making use of our additional fringe field information is to employ COSY's general element GE. We want to demonstrate this with a simple example.

The program given in figure 5.3.3 compares a quadrupole implemented with GE to the standard quadrupole element in COSY.

```
include 'COSY' ;
procedure run ;
variable s 1000 100 ;
variable h 1000 100 ;
variable v 1000 100 ;
variable w 1000 100 ;
variable i 1 ;


ov 2 2 0 ;
rpp 1000 ;
um ;

loop i 1 6 ;
 s(i):=i-1 ;
 h(i):=0 ;
 v(i):=0 ;
 w(i):=DA(1)*DA(3) ;
endloop ;

GE 6 3 s h v w ;
pm 6 ;

um ;
mq 5 1/100 .01 ;
pm 6 ;

endprocedure ; { run }

run; end ;
```

Figure 5.1: Demonstration of the general element GE in COSY

For a basic introduction to GE refer to [17]. The DA-vectors W are giving the magnetic field at the respective point specified in S. For a quadrupole the potential is proportional to $x\,y$. The coordinates are sorted x, a, y, b, s, δ, so this proportionality is specified by setting $W(i) := DA(1) * DA(3)$, where $DA(1)$ and $DA(3)$ are the first and third DA-base vector respectively.

From the definition of W it follows that the magnetic field at the pole tip of a quadrupole of aperture D is equal to D. So the specified general element and the following standard quadrupole are identical.

It is important to notice that for elements with curved optical axis $H(i)$ contains the DA vector describing the local curvature of the optical axis. This DA vector can be obtained in the following way. Locally the magnetic field can be assumed to be constant. Then we can use the well known equation for the radius of the circular orbit of a particle carrying the charge q in a magnetic field B

$$R = \frac{p}{qB} \tag{5.8}$$

or using the definition $\chi_m = p/q$

$$H = \frac{B}{\chi_m} \tag{5.9}$$

The value of χ_m is available as a global variable in COSY which is initialised as soon as the energy, mass and charge of the particle under study are specified.

5.4 Extraction of the Multipole Content

For beam dynamics studies it is essential to extract the multipole components from the field calculation.

We use the expression for the scalar magnet potential in the curl- and divergence-free region given in equation (1.33). By expanding the cosine and noting that a "normal" quadrupole magnet has $\theta_{2,2} = -\pi/2$, we can define the normal and skew components of the potential by the fact that they reach the maximum, respectively vanish at $-\pi/2$

$$\begin{aligned} b_{k,l}(s) &= -M_{k,l}(s)\sin\theta_{k,l} \\ a_{k,l}(s) &= M_{k,l}(s)\cos\theta_{k,l} \end{aligned}$$

After the expansion in normals and skews the potential becomes

$$V_B = -\sum_{k,l=0}^{\infty} \left(b_{k,l}(s)\sin l\phi + a_{k,l}(s)\cos l\phi\right) r^k \tag{5.10}$$

Using equation (1.35) it is easy to see that the functions $b_{k,l}(s)$ and $a_{k,l}(s)$ are related by recurrence relations

$$b_{l+2n,l}(s) = \frac{b_{l,l}^{(2n)}(s)}{\Pi_{\nu=1}^{n}\left(l^2-(l+2\nu)^2\right)} \tag{5.11}$$

all others being zero, and the same for the skews. We recall that the terms that contain s-derivatives are called pseudo-multipoles while the others are called multipoles. To analyse the multipole content of the field usually B_r is evaluated at a certain radius and different ϕ values and then a numerical Fourier analysis is performed. Obviously this approach does not allow to separate real and pseudo-multipoles as for this task you need both the r and the ϕ dependence of B_r. This information is available in the Taylor expansion of B_r.

5.4.1 Analytical Fourier Transform

The DA-based field calculation yields the Taylor expansion of the Cartesian components of the field depending on x, y and s. We can now use the transformation

formulas from Cartesian to cylindrical coordinates to get B_r and B_ϕ. Obviously it is sufficient to study one of them, so we focus on B_r. We recall that

$$B_r = B_x \cos\phi + B_y \sin\phi \tag{5.12}$$

Inserting in the Taylor expansions for B_x and B_y and using $x = r\cos\phi$, $y = r\sin\phi$ gives us B_r as a function of r and ϕ in the form of

$$B_r = \sum_{i,j} r^{i+j} \sin^i\phi \cos^j\phi \left[c_{i,j}\cos\phi + d_{i,j}\sin\phi\right] \tag{5.13}$$

where $c_{i,j}$ and $d_{i,j}$ are the coefficients of the term proportional to $x^i y^j$ in the Taylor expansion of B_x and B_y respectively. We then use the representation of powers of trigonometric functions in terms of functions of multiples of the argument [33] and the formulas for products of trigonometric functions to perform an analytic Fourier transformation of equation (5.13) with respect to the angle, while at the same time we also preserve the Taylor expansion with respect to the radius that was already achieved before.

5.4.2 Example

We demonstrate this method for the LHC High Gradient Quadrupoles. The field calculations for these magnets presented in [23] and [24] are performed by ROXIE [25]. The calculation is based on a set of line currents which are generated from the magnet model shown in figure 5.2 and 5.3. Only the part of the coil extending beyond the edge of the iron yoke is shown. G. Sabbi has implemented an output format in ROXIE that writes the positions and strengths of all line currents, including the straight, image and end currents to a file. We use this information to calculate the magnetic field in COSY.

In [23] G. Sabbi defines $\overline{b_n}$ and calculates it for the lead and return end of the magnet. For the definition of $\overline{b_n}$ see equation (4.4). Unlike the computation of $M_{l,l}$

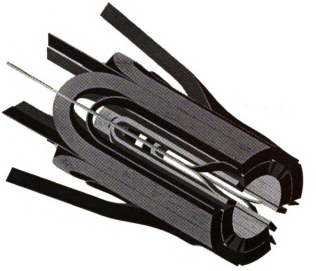

Figure 5.2: Model for the High Gradient Quadrupole's lead end [23].

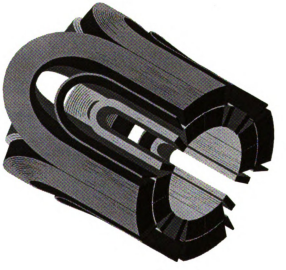

Figure 5.3: Model for the High Gradient Quadrupole's return end [23].

which requires only a DA-computation of order $l-1$ to be exact to machine precision, it requires an infinite series to calculate $\bar{b}_n$. Nevertheless, as you can see from the figures 5.4 and 5.5 the agreement of $\bar{b}_2(s)$ given in [23] and the DA-based computation is already extremely good in order 9.

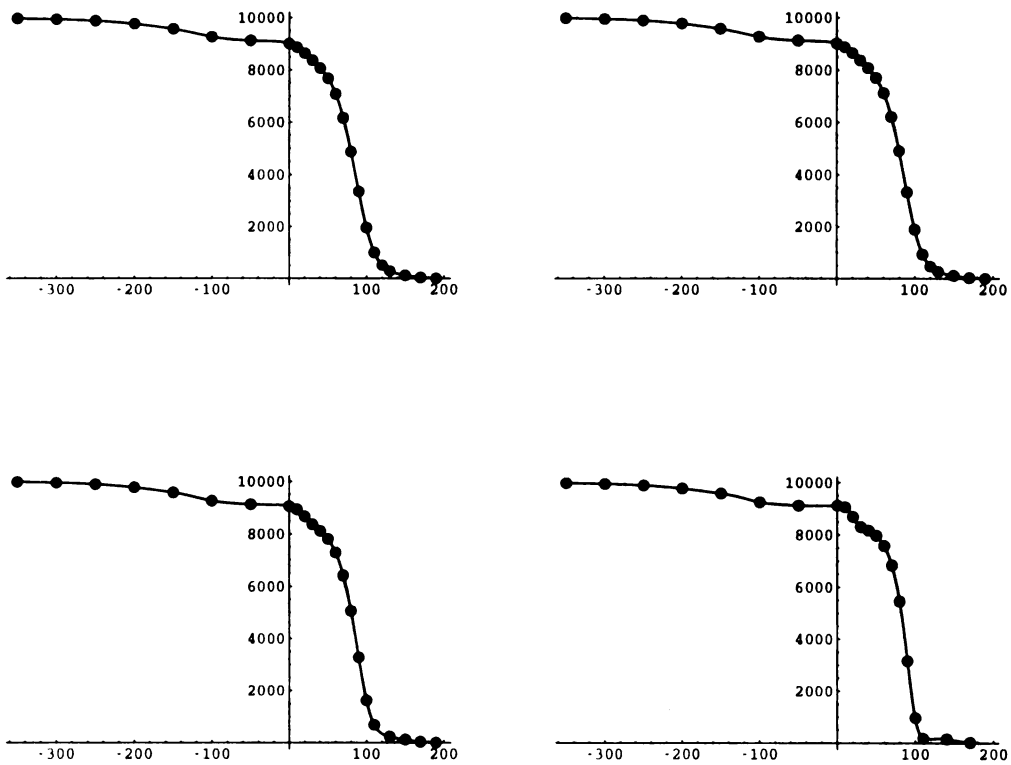

Figure 5.4: Comparison of $\bar{b}_2$ in the lead end as computed by the differential algebraic field calculation in COSY and as given in [23] for different radii. From top left to bottom right : $r = 5$ mm, $r = 10$ mm, $r = 20$ mm and $r = 30$ mm.

5.4.3 Differential Algebraic Multipole Extraction

It is possible to extract the multipole content of magnetic fields by Differential Algebraic methods directly in a very elegant way that is arbitrary in order. We define

$$f_l(r,s) = \sum_{n=0}^{\infty} b_{l+2n,l}(s) r^{2n} = \sum_{n=0}^{\infty} \frac{b_{l,l}^{(2n)}(s)}{\Pi_{\nu=1}^{n} (l^2 - (l+2\nu)^2)} r^{2n}$$

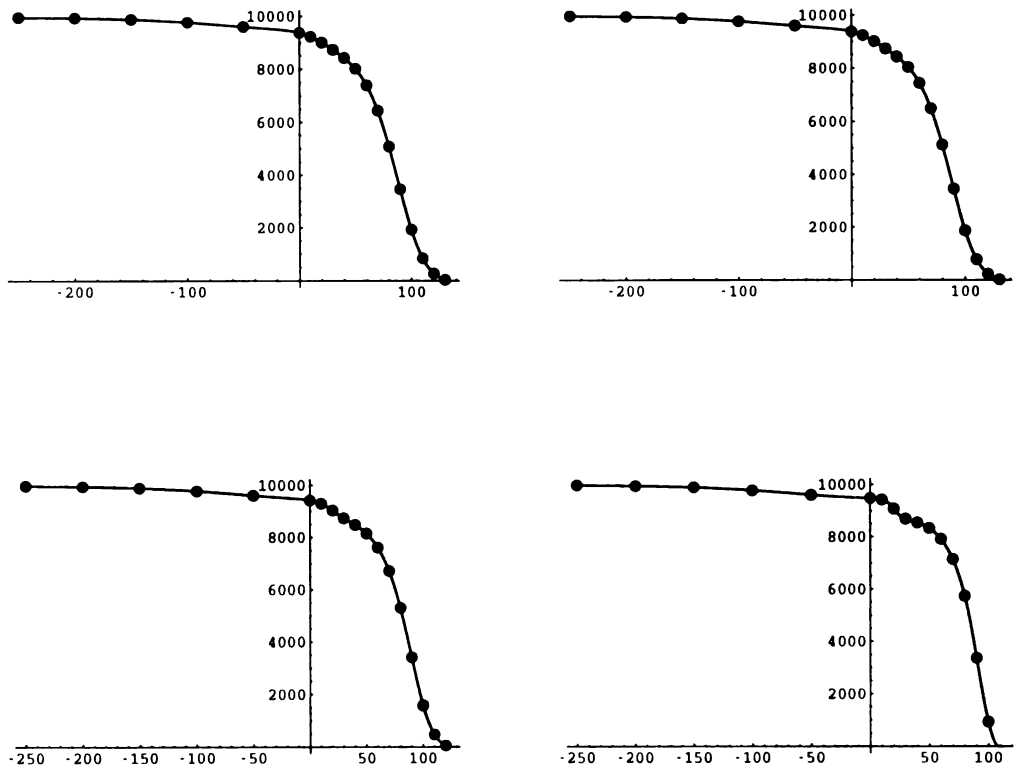

Figure 5.5: Comparison of $\bar{b}_2$ in the return end as computed by the differential algebraic field calculation in COSY and as given in [23] for different radii. From top left to bottom right : $r = 5$ mm, $r = 10$ mm, $r = 20$ mm and $r = 30$ mm.

$$g_l(r,s) = \sum_{n=0}^{\infty} a_{l+2n,l}(s) r^{2n} = \sum_{n=0}^{\infty} \frac{a_{l,l}^{(2n)}(s)}{\prod_{\nu=1}^{n} \left(l^2 - (l+2\nu)^2\right)} r^{2n}$$

Inserting this into equation (5.10) yields

$$V_B = -\sum_{l=0}^{\infty} \left(f_l(r,s)\sin l\phi + g_l(r,s)\cos l\phi\right) r^l \tag{5.14}$$

The magnetic field components in cylindrical coordinates can be calculated using the well known formulas

$$\begin{aligned} B_r &= -\frac{\partial V_B}{\partial r} \\ B_\phi &= -\frac{1}{r}\frac{\partial V_B}{\partial \phi} \\ B_s &= -\frac{\partial V_B}{\partial s} \end{aligned}$$

resulting in the expressions

$$B_r(r,\phi,s) = \tilde{g}_0(r,s) + \sum_{l=1}^{\infty} \left(\tilde{f}_l(r,s)\sin l\phi + \tilde{g}_l(r,s)\cos l\phi\right) r^{l-1}$$

$$B_\phi(r,\phi,s) = \sum_{l=1}^{\infty} \left[l\left(f_l(r,s)\cos l\phi - g_l(r,s)\sin l\phi\right)\right] r^{l-1}$$

$$B_s(r,\phi,s) = \sum_{l=0}^{\infty} \left(f_l'(r,s)\sin l\phi + g_l'(r,s)\cos l\phi\right) r^l$$

where prime denotes derivative with respect to s, and

$$\tilde{f}_l(r,s) = \sum_{n=0}^{\infty} (l+2n) b_{l+2n,l}(s) r^{2n} = \sum_{n=0}^{\infty} \frac{(l+2n) b_{l,l}^{(2n)}(s)}{\prod_{\nu=1}^{n} \left(l^2 - (l+2\nu)^2\right)} r^{2n}$$

$$\tilde{g}_l(r,s) = \sum_{n=0}^{\infty} (l+2n) a_{l+2n,l}(s) r^{2n} = \sum_{n=0}^{\infty} \frac{(l+2n) a_{l,l}^{(2n)}(s)}{\prod_{\nu=1}^{n} \left(l^2 - (l+2\nu)^2\right)} r^{2n}$$

It can be seen that every multipole strength, except for $l = 0$, is multiplied by

r^{l-1}. For the special case $l = 0$, we get

$$\begin{aligned} B_r(r,s) &= \tilde{g}_0(r,s) = -\sum_{k=1}^{\infty}(-1)^{k+1}\frac{k}{2^{2k-1}k!k!}a_{0,0}^{(2k)}(s)r^{2k-1} \\ B_\phi(r,s) &= 0 \\ B_s(r,s) &= -\sum_{k=0}^{\infty}(-1)^{k+1}\frac{1}{2^{2k}k!k!}a_{0,0}^{(2k+1)}(s)r^{2k} \end{aligned} \tag{5.15}$$

In the DA picture, the field calculations are done locally, as a Taylor expansion of the field with respect to Cartesian coordinates x, y, s. Hence, we need the equations relating the cylindrical and Cartesian components of the magnetic field.

$$\begin{aligned} B_x(r,\phi,s) &= B_r(r,\phi,s)\cos\phi - B_\phi(r,\phi,s)\sin\phi \\ B_y(r,\phi,s) &= B_r(r,\phi,s)\sin\phi + B_\phi(r,\phi,s)\cos\phi \end{aligned}$$

and $B_s(r,\phi,s)$ is unchanged. Obviously, if we evaluate the above equations in the midplane ($y = \phi = 0$), then

$$B_r(r,\phi=0,s)\mid_{r\mapsto x} = B_x(x,y=0,s)$$

$$B_\phi(r,\phi=0,s)\mid_{r\mapsto x} = B_y(x,y=0,s)$$

We know that the Cartesian components of the fields from the DA calculation in the midplane

$$\begin{aligned} B_x(x,y=0,s) &= \tilde{g}_0(x,s) + \sum_{l=1}^{\infty}\tilde{g}_l(x,s)\cdot x^{l-1} \\ B_y(x,y=0,s) &= \sum_{l=1}^{\infty} l f_l(x,s)\cdot x^{l-1} \end{aligned}$$

This is all the information we need to extract the multipole strengths up to the order of calculation, because, as previously mentioned, any multipole strength of order l is multiplied by x^{l-1}. Starting at $l = 1$, $a_{1,1}(s)$ is extracted as the x-independent part of B_x, and analogously $b_{1,1}(s)$ from B_y. Evaluating $a_{1,1}(s)$ and $b_{1,1}(s)$ at $s = 0$ yields the skew and normal dipole component. From $a_{1,1}(s)$ and $b_{1,1}(s)$ the functions $f_1(x,s)$,

$\tilde{g}_1(x,s)$ are generated up to the order of calculation and subtracted from $B_x(x, y = 0, s)$, respectively $B_y(x, y = 0, s)$. This cancels the pseudo-multipoles generated by the s-dependence of $a_{1,1}(s)$ and $b_{1,1}(s)$, which otherwise would make the distinction between sextupole terms and pseudo-dipole terms impossible. The procedure can be iterated for the higher order multipoles, up to the order of calculation. After the k-th step, the remainder of the field components should contain just $(k+1)$-th and higher order multipoles.

However, there is an additional problem in the case of solenoidal fields (case $l = 0$). In this case we have an $a_{0,0}(s)$ in the potential, but its contribution vanishes from the field components B_x and B_y, so the function $\tilde{g}_0(x,s)$ cannot be generated from the information available in B_x and B_y. Here comes to the rescue the B_s component, which evaluated at $x = y = 0$ yields $a'_{0,0}(s)$. From this function we can calculate $a_{0,0}(s)$ up to a constant and generate the $l = 0$ contribution to B_x, $\tilde{g}_0(x,s)$. Once this is subtracted from B_x, the method works as previously described, starting with $l = 1$.

Finally, two notes: the method relies on the fact that the magnetic field can be generated by a magnetic scalar potential that satisfies the Laplace equation. Hence, it is important that the field is really curl-free. If the fields are calculated from line currents by the Biot-Savart law, that means that the model should consist only of closed circuits to ensure the curl is vanishes. Secondly, in the region where the magnetic field is not s-dependent, the functions f_l and g_l are the real multipoles and it is $lf_l = \tilde{f}_l$ and $lg_l = \tilde{g}_l$, an assumption that is sometimes made even for the s-dependent region.

The method has been implemented in the code COSY Infinity, and in the following we will present results for two examples : a toy-model consisting of a single square

current-loop, and a realistic model of the LHC interaction region quadrupole's fringe region.

5.4.4 Example

First we demonstrate the algorithm using an easy example. We study the magnetic field of a square current loop of side length 20 cm, orientated parallel to the x, y plane centered around the s-axis at $s = 10$ cm. The current is 10 kA clockwise, looking along the s-direction. We calculate the field expansion around $\vec{r} = 0$. Due to the symmetry of this arrangement, only the $4k$ multipoles are allowed, where $k \in \mathcal{N}$. In table 5.1 we show the skew multipoles and compare the result of the DA-Extraction with the analytical Fourier Transform. The normal multipoles vanish. Table 5.2 gives the derivatives of the solenoidal component $a_{0,0}$ of the field. It should be noticed that according to equation (5.16) from B_r only the even order derivatives of $a_{0,0}$ can be obtained via Fourier transformation.

l	DA-Extraction	Analytical Fourier Transformation
1	0	0
2	0	0
3	0	0
4	0.4677606347601	0.4677606347601
5	0	0
6	0.8052817671947803E-13	0
7	0	0
8	-94.0084410633	-94.0084410633

Table 5.1: Skew multipoles as obtained by the DA-based extraction scheme in comparison with the results of the analytical Fourier transformation at $s = 0$.

For the second example we look at the end fields of the LHC High Gradient Quadrupoles again. In order to apply the DA-Extraction in this case we have to make sure that the line currents used in the magnet model form closed loops as

n	DA-Extraction	Analytical Fourier Transformation
1	0.2309401076758502E-01	
2	-0.307920143566	-0.307920143566
3	3.849001794594	
4	-12.83000598197	- 12.83000598198
5	-2035.694282476	
6	116325.38757	116325.38757
7	-3789128.433348	
8	12972561.60401	- 12972537.8827

Table 5.2: S-Derivatives of the solenoid component of the field as obtained by the DA-based extraction scheme in comparison with the results of the analytical Fourier transformation at $s = 0$. n denotes the order of the derivative.

explained earlier. In table 5.3 we show the resulting normal and skew multipoles in the lead for $z = 0$. The coordinate system is as defined in [23].

multipole strength	DA-Extraction	Analytical Fourier Transformation
b_2	8.270972386	8.270972386
b_6	28371.07332	28371.07796
a_2	0.1043500051	0.1043500051
a_6	-11935.21351	-11935.42465

Table 5.3: Comparison of the allowed multipoles in the lead end at $z = 0$cm in the coordinate system defined in [23] calculated by the DA-Extraction and the analytical Fourier transformation.

Finally, we show in the figures 5.6 and 5.7 a comparison of $\overline{a}_2, \overline{a}_6, \overline{b}_2, \overline{b}_6$ given in [23] with the respective real multipoles as computed by our algorithm for the radii $r = 5$ mm and $r = 20$ mm. As before the values from [23] are given by the line and our values are given by dots. One can see that while there is very good agreement for $r = 5$ mm there are substantial differences especially for the skew components at $r = 20$ mm. We chose $r = 20$ mm because this is close to the new LHC reference radius $r = 17$ mm. Presumably this is the only radius for which magnet measurements will

become available in the future. The plots 5.6 and 5.7 show, that one can in general not assume that the $\bar{a}_n$ and $\bar{b}_n$ measured at the reference radius are equal to the real multipoles.

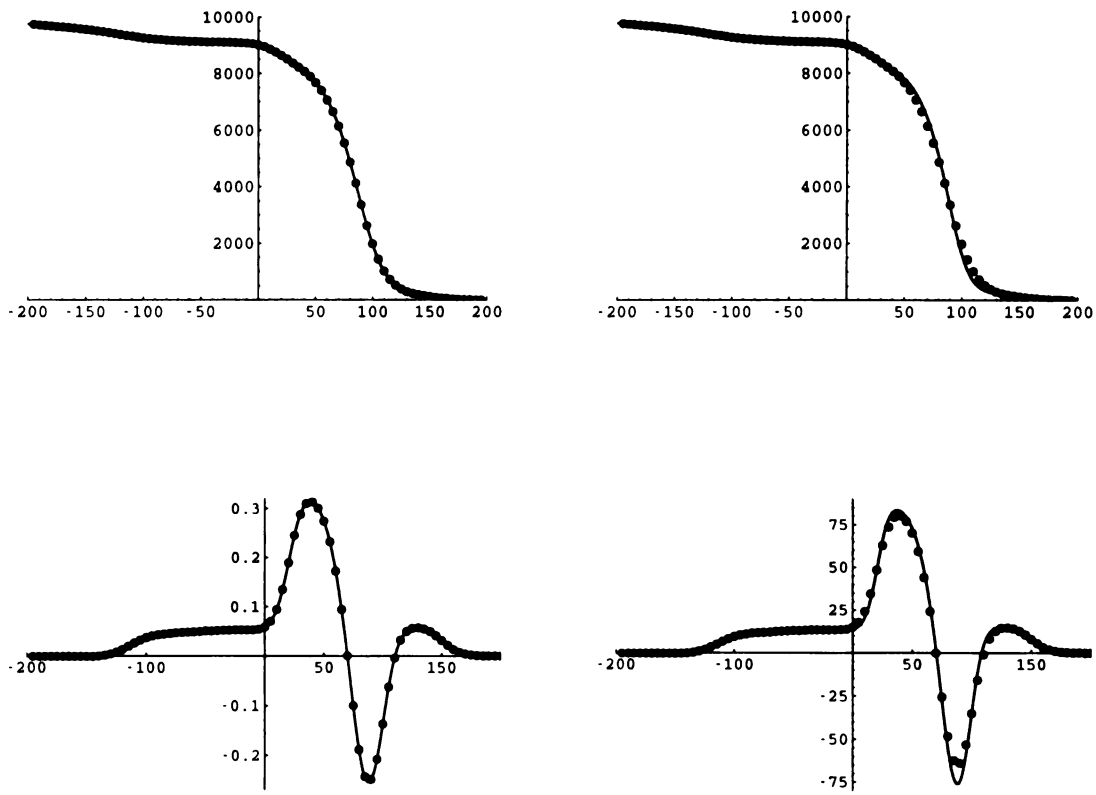

Figure 5.6: Comparison of the normal multipoles $b_{2,2}$ (top) and $b_{6,6}$ (bottom) with $\bar{b}_2$ respectively $\bar{b}_6$ in the lead end for $r = 5$ mm (left) and $r = 20$ mm (right).

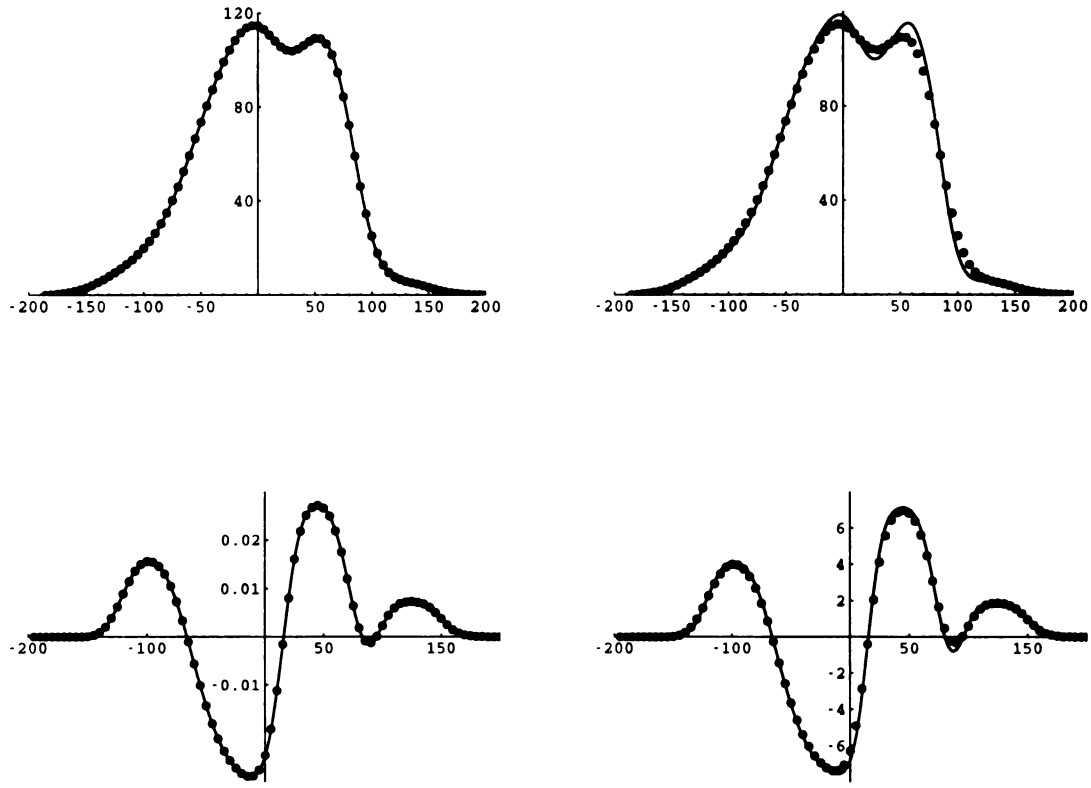

Figure 5.7: Comparison of the skew multipoles $a_{2,2}$ (top) and $a_{6,6}$ (bottom) with $\overline{a}_2$ respectively $\overline{a}_6$ in the lead end for $r = 5$ mm (left) and $r = 20$ mm (right).

Appendix A

SXF to COSY Converter

```
#!/usr/bin/perl
# form free SXF2COSY
# by Michael Lindemann
# July 1998

@elements=("//","drift","rbend","sbend","quadrupole",
           "sextupole","octupole","multipole","solenoid",
           "rfcavity","elseparator","kicker","endsequence",
           "sequence","marker","instrument","monitor",
           "ecollimator","rcollimator","beambeam");

$Pi=3.141592653589793;

sub unsupported
{
    $_=$field[$mk] ;
    $count=tr/A-Z// ;          # keywords are lower case
    $count+=tr/0-9// ;         # contain no numbers
    $count+=tr/\(\)\[\]\=// ;  # and special characters
    if ($count==0)
    { print "{ *** element not supported: $_ }\n"; }
}

sub comment
{
    $mk++ ;
    print "{ " ;
    until ($field[$mk] eq "//")
    { print "".$field[$mk] ; $mk++ }
    print "}\n" ;
}

sub marker { &drift ; }
sub instrument { &drift ; }
sub monitor { &drift ; }

sub sequence
{
    $mk++ ;
}

sub drift
{
    sub drift_l
    {
        $mk+=2 ;
        $l=$field[$mk] ;
    }

sub drift_at
{
    $mk+=2 ;
    $l_at=$field[$mk] ;
    $flag_at=1 ;
}
```

```
@l_elements=("l","at","//") ;

$l=$l_at=$flag_at=0 ;
$out="" ;
$mk+=2 ;
until ($field[$mk] eq "}")
{
    $flag=0 ;
    if ($field[$mk] eq "//") { &comment ; }
    else
    {
        for ($i=0;$i<=@l_elements-1 ; $i++)
        {
            if ($l_elements[$i] eq $field[$mk])
            { $flag=1 ; $l_type=$i ;
            }
        }
        $etype=$l_elements[$l_type] ;
        $etype="drift_".$etype ;
        if ($flag==1) { print " ---- $etype called\n"; &$etype}
        else { &unsupported ; }
    }
    $mk++ ;
}
if ($flag_at==1)  # longitudinal positioning of elements
{
    print "dl".($l_at-$at-$l/2).";\n" ;
    $at=$l_at+$l/2 ;
}
else              # elements are butted up
{
    $at+=$l ;
}
if ($l>0) { print "dl".$l.";\n" ; }
}

sub rbend
{
    sub rbend_l
    {
        $mk+=2 ;
        $flag_l=1 ;
        $l=$field[$mk] ;
    }
sub rbend_arc
{
    $mk+=2 ;
    $flag_arc=1 ;
    $arc=$field[$mk] ;
}
sub rbend_at
{
    $mk+=2 ;
    $l_at=$field[$mk] ;
    $flag_at=1 ;
}
```

```
sub rbend_body
{
    sub rbend_body_kl
    {
        $mk+=3 ;
        $angle=$field[$mk] ;
        $mk++ ;
        $dflag=1 ;
        until ($field[$mk] eq "]")
        { if ($field[$mk]!=0) { $dflag=0 ; }
          $mk++ ; }
        if ($dflag==0)
        {print" { *** warning : higher multipoles ".
               "in dipole not supported } \n" ; }

        # dipoles with higher order multipole errors
        # are not yet supported,
        # so an error message is issued if they occur
    }
sub rbend_body_kls
{
    $dflag=1 ;
    $mk+=3 ;
    until ($field[$mk] eq "]")
    { if ($field[$mk]!=0) { $dflag=0 ; }
      $mk++ ; }
    if ($dflag==0)
    { print " { *** warning : skew multipoles ".
            "in dipole not supported } \n" ; }
}
# rbend_body
@rbend_body_elements=("kl","kls") ;
$mk+=3 ;
until ($field[$mk] eq "}")
{
    $flag=0 ;
    if ($field[$mk] eq "//") { $_=$line ; &comment ; }
    else
    {
        for ($i=0;$i<=@rbend_body_elements-1 ; $i++)
        {
            if ($rbend_body_elements[$i] eq $field[$mk])
            {
                $flag=1 ; $l_type=$i ;
                print " -- in rbend_body sucessfully matched ".
                    $rbend_body_elements[$i]." ".$field[$mk
].")\n" ;
            }
            else { print " -- in rbend_body not sucessfully matched ".
                        $rbend_body_elements[$i]." ".$field[
$mk]."\n" ;
                 }
        }
        $etype=$rbend_body_elements[$l_type];
        $etype="rbend_body_".$etype ;
        if ($flag==1) { &$etype ; print " -- in rbend_body ".
                                    $etype." called\n" ; }
        else { &unsupported ; }
    }
    $mk++ ;
}
}
# rbend
@rbend_elements=("at","l","arc","body","//",) ;
$arc=$l=$nma=$nms=0 ; $out="" ;
$flag_l=$flag_arc=$flag_at=0 ;
$e1=$e2=0 ;
$mk+=2 ;

until ($field[$mk] eq "}")
{
    $flag=0 ;
    if ($field[$mk] eq "//") { $_=$line ; &comment ; }
    else
    {
        for ($i=0;$i<=@rbend_elements-1 ; $i++)
        {
            if ($rbend_elements[$i] eq $field[$mk])
            {
                $flag=1 ; $rbend_type=$i ;
                print " -- in rbend sucessfully matched ".
                    $rbend_elements[$i]." to ".$field[$mk]."\
n" ; }
        }
        $etype="rbend_".$rbend_elements[$rbend_type] ;
        if ($flag==1) { print " -- in rbend ".$etype.
                                " called\n";  &$etype ;  }
        else { &unsupported ;}
    }
    $mk++ ;
}


# the bending direction is determined by the sign of the an
gle in MAD
if ($angle<0) { $flag_cb=1 ; print " cb;\n" ; $angle*=(-1) ;
 }
else { $flag_cb=0 ; }
# the cosy element dil
# calculates the radius from the length of the curved orbit
# and calls di with this radius
# l gives the straight line length so the arc length has to
 be calculated
# from l and angle

# angles are given in radiant in SXF and in degree in COSY
$angle_deg=$angle*180/$Pi ; # angle in degree

# for rbend add half of the bending angle to e1 and e2
# due to different coordiante system
# see MAD Manual page 25,26
```

```
if ($flag_sbend==0) { $el+=$angle_deg/2 ; $e2+=$angle_deg/2
 }
if ($flag_arc==1)
{
# calculation of l from arc
    if ($angle>0) ( $l=$arc/($angle/2/sin($angle/2)) ;)
    else { $l=$arc ; }

    if ($flag_at==1)     # longitudinal positioning of eleme
nts
    {
        print "dl ".($l_at-$at-$l/2)." ;\n" ;
        $at=$l_at+$l/2 ;
    }
    else                              # elements are butted up
    {
        $at+=$l ;
    }
#    print "{ l= ".$l." } \n";
    print "dl ".$arc." ".$angle_deg." D\n   ".
        $el." 0 ".$e2." 0;\n" ;
}
else
{ if ($flag_l== 1)
   {
       if ($angle>0) { $arc=$l*$angle/2/sin($angle/2) ;}
       else { $arc=$l ; }

       if ($flag_at==1)  # longitudinal positioning of eleme
nts
       {
           print "dl ".($l_at-$at-$l/2)." ;\n" ;
           $at=$l_at+$l/2 ;
       }
       else                           # elements are butted up
       {
           $at+=$l ;
       }
#      print "{ l= ".$l." } \n";
       print "dl ".$arc." ".$angle_deg." D\n   ".
           $el." 0 ".$e2." 0;\n" ;
   }
  else { print " { *** warning in rbend : neither l nor arc given } \n" ;
         print " { ***                thin bends are not supported } \n" ;
       }
}
if ($flag_cb==1) { print "cb;\n" ; }
}

sub sbend
{
    $flag_sbend=1 ;
    &rbend ;
    $flag_sbend=0 ;
}
```

```
sub quadrupole { print " { quadrupole } \n" ; &multipole ; }
sub sextupole  { print " { sextupole } \n" ; &multipole ; }
sub octupole   { print " { octupole } \n" ; &multipole ; }

sub multipole
{
    sub multipole_l
    {
        $mk+=2 ;
        $l=$field[$mk] ;
    }
sub multipole_at
{
    $mk+=2 ;
    $l_at=$field[$mk] ;
    $flag_at=1 ;
}
sub multipole_body
{
    sub multipole_body_kl
    {
        $mk+=3 ;
        $nma=0 ;
        until ($field[$mk] eq "]")
        { $ma[1+$nma]=$field[$mk]; $mk++; $nma++; }
    }
sub multipole_body_kls
{
    $mk+=3 ;
    $nms=0 ;
    until ($field[$mk] eq "]")
    { $ms[1+$nms]=$field[$mk]; $mk++; $nms++ }
}
sub multipole_body_lrad
{
    print " { *** warning : lrad ignored } \n" ;
}

# main part of multipole_body
@multipole_body_elements=("kl","kls","lrad") ;
$mk+=3 ;
until ($field[$mk] eq "]")
{
    $flag=0 ;
    if ($field[$mk] eq "//")
    {
        print " ---- comment called in multipole \n";
        &comment ;
    }
    else
    {
        for ($i=0;$i<=@multipole_body_elements-1 ; $i++)
        {
            if ($multipole_body_elements[$i] eq $field[$mk]
)
            { $flag=1 ; $l_type=$i ; }
```

```
        }
        $etype=$multipole_body_elements[$l_type];
        $etype="multipole_body_".$etype ;
        if ($flag==1)
        {  print " —— in multipole_body ".$etype." called\n" ;
           &$etype ;}
        else
        { &unsupported ;}
    }
    $mk++ ;
}
}

sub multipole_body_dev
{
    sub multipole_body_dev_kl
    {
        $mk+=3 ;
        $nma=0 ;
        until ($field[$mk] eq "]")
        { $ma[1+$nma]+=$field[$mk]; $mk++; $nma++ }
    }
sub multipole_body_dev_kls
{
    $mk+=3 ;
    $nms=0 ;
    until ($field[$mk] eq "]")
    { $ms[1+$nms]+=$field[$mk]; $mk++; $nms++ }
}


# main part of multipole_body_dev
@multipole_body_dev_elements=("kl","kls") ;
$mk+=3 ;
until ($field[$mk] eq "}")
{
    $flag=0 ;
    if ($field[$mk] eq "//") { &comment ; }
    else
    {
        for ($i=0;$i<=@multipole_body_dev_elements-1 ; $i++
)
        {
            if ($multipole_body_dev_elements[$i] eq $field[
$mk])
            { $flag=1 ; $l_type=$i ; }
        }
        $etype=$multipole_body_dev_elements[$l_type];
        $etype="multipole_body_dev_".$etype ;
        if ($flag==1)
        {  print " —— in multipole_body_dev ".$etype." called\n" ;
           &$etype ;}
        else
        { &unsupported ; }

        $mk++ ;
```

```
    }
}
}

sub multipole_align_dev
{
    $warn= " { *** warning : alignment deviations ignored } \n" ;
    until ($field[$mk] eq "}") { $mk++; }
}

# main part of multipole
@multipole_elements=("at","l","body","body.dev","align.dev","//",)
 ;
for ($i=1; $i<=20 ; $i++ ) { $ma[$i]=$ms[$i]=0 ;}
$l=$nma=$nms=0 ; $out=$warn="" ;
$l_at=$at ; $flag_at=0 ;
$mk+=2 ;
until ($field[$mk] eq "}")
{
    $flag=0 ;
    if ($field[$mk] eq "//") { $_=$line ; &comment ; }
    else
    {
        for ($i=0;$i<=@multipole_elements-1 ; $i++)
        {
            if ($multipole_elements[$i] eq $field[$mk]) {
                $flag=1 ; $multipole_type=$i ;
                print " —— in multipole sucessfully matched ".
                    $multipole_elements[$i]." to ".$field[$m
k]."\n" ;}
        }
        $etype="multipole_".$multipole_elements[$multipole_ty
pe] ;
        $_=$etype ;
        tr /./_/ ;
        if ($flag==1) { print " —— in multipole ".$_." called \n";
                        &$_ ;  }
        else
        { &unsupported ; }
    }
    $mk++ ;
}
print $warn ;
if ($flag_at==1) # longitudinal positioning of elements
{
    print "dl ".($l_at-$at-$l/2).";\n" ;
    $at=$l_at+$l/2 ;
}
else                        # elements are butted up
{
    $at+=$l ;
}
if ($l==0) { print "dl -1E-10;\n"; $l=1E-10 ; }   # thin mul
tipole
if ($nma>$nms) { $nmax=$nma ; }
else { $nmax=$nms ; } # max of number of multipole coeff
```

```
# setting of multipole coefficients
# the cosy command MMS expects two arrays MA and MS of norm
al
# and skew multipole coefficients
# the first entry MA(1) is the quadrupole component and not
(!)
# the dipole component
# differences between MAD and COSY in the yield a factor of

# CONS(CHIM)*D^(l-1) where l is the multipole order
# This factor can also be found in the COSY element MQK
# which is designed to interface to MAD
if ((($ma[1]!=0)||($ms[1]!=0))
{ print " { *** warning : dipole component ".
        "in multipole not supported \n"; }
if ($nmax<2)
{
    print " { *** warning : no multipole component specified } \n";
    print " MA(1):=0 ;\n" ;
    print " MS(1):=0 ;\n" ;
}

# fixing the sextupole problem
$ma[3]/=2*$l ;
$ms[4]/=2*$l ;
# end of sextupole patch

for ($i=2; $i<=$nmax ; $i++)
{ print " MA(".($i-1)."):= ".$ma[$i]/$l."*CONS(CHIM)*D^".($i
-1)." ;\n" ;}
for ($i=2; $i<=$nmax ; $i++)
{ print " MS(".($i-1)."):= ".$ms[$i]/$l."*CONS(CHIM)*D^".($i-
1)." ;\n" ;}
print " MMS ".$l." MA MS ".($nmax-1)." D ;\n" ; # aperture se
t by virtue
}


sub endsequence
{
    $mk+=3 ;
    print " dl ".($field[$mk]-$at)." ;\n" ;
    print " endprocedure ; \n" ;
    $mk++ ;
}

# unsupported elements
sub solenoid
{
  . sub solenoid_l
    {
        $mk+=2 ;
        $l=$field[$mk] ;
    }
sub solenoid_at
{
    $mk+=2 ;
    $l_at=$field[$mk] ;
    $flag_at=1 ;
}
sub solenoid_body
{
    $mk+=3 ;
    while ($field[$mk] ne "}") { $mk++ ; }
    print " { *** warning : solenoid_body not supported } \n" ;
}

@l_elements=("l","at","body","//") ;
$local="solenoid" ;

$l=$l_at=$flag_at=0 ;
$out="" ;
$mk+=2 ;
until ($field[$mk] eq "}")
{
    $flag=0 ;
    if ($field[$mk] eq "//") { &comment ; }
    else
    {
        for ($i=0;$i<=@l_elements-1 ; $i++)
        {
            if ($l_elements[$i] eq $field[$mk])
            { $flag=1 ; $l_type=$i ;
            }
        }
        $etype=$l_elements[$l_type] ;
        $etype=$local."_".$etype ;
        if ($flag==1) { print " --- $etype called \n"; &$etype}
        else
        { &unsupported ; }
    }
    $mk++ ;
}
if ($flag_at==1)                # longitudinal positioning
of elements
{
    print " dl ".($l_at-$at-$l/2)." ;\n" ;
    $at=$l_at+$l/2 ;
}
else                        # by default elements are butted up
{
    $at+=$l ;
}
print " dl ".$l." ;\n"; # unknown element replaced by drift
}

sub rfcavity
{
    sub rfcavity_l
    {
        $mk+=2 ;
```

```
        $l=$field[$mk] ;
    }
sub rfcavity_at
{
    $mk+=2 ;
    $l_at=$field[$mk] ;
    $flag_at=1 ;
}
sub rfcavity_body
{
    $mk+=3 ;
    while ($field[$mk] ne "}") { $mk++ ; }
    print " { *** warning : rfcavity_body not supported } \n" ;
}

@l_elements=("l","at","body","//") ;
$local="rfcavity" ;

$l=$l_at=$flag_at=0 ;
$out="" ;
$mk+=2 ;
until ($field[$mk] eq "}")
{
    $flag=0 ;
    if ($field[$mk] eq "//") { &comment ; }
    else
    {
        for ($i=0;$i<=@l_elements-1 ; $i++)
        {
            if ($l_elements[$i] eq $field[$mk])
            { $flag=1 ; $l_type=$i ;
          }
        }
        $etype=$l_elements[$l_type] ;
        $etype=$local."_".$etype ;
        if ($flag==1) { print " —— $etype called \n"; &$etype}
        else
        { &unsupported ; }
    }
    $mk++ ;
}
if ($flag_at==1)        # longitudinal positioning of elemen
ts
{
    print " dl ".($l_at-$at-$l/2)." ;\n" ;
    $at=$l_at+$l/2 ;
}
else                    # by default elements are butted up
{
    $at+=$l ;
}
print " dl ".$l." ;\n"; # unknown element replaced by drift
}

sub elseparator
{
```

```
    sub elseparator_l
    {
        $mk+=2 ;
        $l=$field[$mk] ;
    }
sub elseparator_at
{
    $mk+=2 ;
    $l_at=$field[$mk] ;
    $flag_at=1 ;
}
sub elseparator_body
{
    $mk+=3 ;
    while ($field[$mk] ne "}") { $mk++ ; }
    print " { *** warning : elseparator_body not supported } \n" ;
}


@l_elements=("l","at","body","//") ;
$local="elseparator" ;

$l=$l_at=$flag_at=0 ;
$out="" ;
$mk+=2 ;
until ($field[$mk] eq "}")
{
    $flag=0 ;
    if ($field[$mk] eq "//") { &comment ; }
    else
    {
        for ($i=0;$i<=@l_elements-1 ; $i++)
        {
            if ($l_elements[$i] eq $field[$mk])
            { $flag=1 ; $l_type=$i ;
          }
        }
        $etype=$l_elements[$l_type] ;
        $etype=$local."_".$etype ;
        if ($flag==1) { print " —— $etype called \n"; &$etype}
        else
        { &unsupported ;
      }
    }
    $mk++ ;
}
if ($flag_at==1)        # longitudinal positioning of element
s
{
    print " dl ".($l_at-$at-$l/2)." ;\n" ;
    $at=$l_at+$l/2 ;
}
else                    # by default elements are butted up
{
    $at+=$l ;
}
```

```
print " dl ".$l.";\n"; # unknown element replaced by drift
}

sub kicker
{
    sub kicker_l
    {
        $mk+=2 ;
        $l=$field[$mk] ;
    }
sub kicker_at
{
    $mk+=2 ;
    $l_at=$field[$mk] ;
    $flag_at=1 ;
}
sub kicker_body
{
    $mk+=3 ;
    while ($field[$mk] ne "]") { $mk++ ; }
    print " { *** warning : kicker_body not supported } \n" ;
}


@l_elements=("l","at","body","//") ;
$local="kicker" ;

$l=$l_at=$flag_at=0 ;
$out="" ;
$mk+=2 ;
until ($field[$mk] eq "}")
{
    $flag=0 ;
    if ($field[$mk] eq "//") { &comment ; }
    else
    {
        for ($i=0;$i<=@l_elements-1 ; $i++)
        {
            if ($l_elements[$i] eq $field[$mk])
            { $flag=1 ; $l_type=$i ;
            }
        }
        $etype=$l_elements[$l_type] ;
        $etype=$local."_".$etype ;
        if ($flag==1) { print " --- $etype called \n"; &$etype}
        else { &unsupported ; }
    }
    $mk++ ;
}
if ($flag_at==1)                    # longitudinal positioning
of elements
{
    print " dl ".($l_at-$at-$l/2).";\n" ;
    $at=$l_at+$l/2 ;
}
else                                # by default elements are butted up
{
    $at+=$l ;
}
print " dl ".$l.";\n"; # unknown element replaced by drift
$mk++ ;
}

sub ecollimator
{
    sub ecollimator_l
    {
        $mk+=2 ;
        $l=$field[$mk] ;
    }
sub ecollimator_at
{
    $mk+=2 ;
    $l_at=$field[$mk] ;
    $flag_at=1 ;
}
sub ecollimator_body
{
    $mk+=3 ;
    while ($field[$mk] ne "]") { $mk++ ; }
    print " { *** warning : ecollimator_body not supported } \n" ;
}


@l_elements=("l","at","body","//") ;
$local="ecollimator" ;

$l=$l_at=$flag_at=0 ;
$out="" ;
$mk+=2 ;
until ($field[$mk] eq "}")
{
    $flag=0 ;
    if ($field[$mk] eq "//") { &comment ; }
    else
    {
        for ($i=0;$i<=@l_elements-1 ; $i++)
        {
            if ($l_elements[$i] eq $field[$mk])
            { $flag=1 ; $l_type=$i ;
            }
        }
        $etype=$l_elements[$l_type] ;
        $etype=$local."_".$etype ;
        if ($flag==1) { print " --- $etype called \n"; &$etype}
        else { &unsupported ; }
    }
    $mk++ ;
}
if ($flag_at==1)                    # longitudinal positioning
of elements
{
```

```
    print "dl ".($l_at-$at-$l/2).";\n" ;
    $at=$l_at+$l/2 ;
}
else                        # by default elements are butted up
{
    $at+=$l ;
}
print "dl ".$l.";\n"; # unknown element replaced by drift
$mk++ ;
}

sub rcollimator
{
    sub rcollimator_l
    {
        $mk+=2 ;
        $l=$field[$mk] ;
    }
sub rcollimator_at
{
    $mk+=2 ;
    $l_at=$field[$mk] ;
    $flag_at=1 ;
}
sub rcollimator_body
{
    $mk+=3 ;
    while ($field[$mk] ne "}") { $mk++ ; }
    print " { *** warning : rcollimator_body not supported } \n" ;
}


@l_elements=("l","at","body","//") ;
$local="rcollimator" ;

$l=$l_at=$flag_at=0 ;
$out="" ;
$mk+=2 ;
until ($field[$mk] eq "}")
{
    $flag=0 ;
    if ($field[$mk] eq "//") { &comment ; }
    else
    {
        for ($i=0;$i<=@l_elements-1 ; $i++)
        {
            if ($l_elements[$i] eq $field[$mk])
            { $flag=1 ; $l_type=$i ;
          }
        }
        $etype=$l_elements[$l_type] ;
        $etype=$local."_".$etype ;
        if ($flag==1) { print " —— $etype called \n"; &$etype)
        else { &unsupported ; }
    }
    $mk++ ;
}
if ($flag_at==1)        # longitudinal positioning of eleme
nts
{
    print "dl ".($l_at-$at-$l/2).";\n" ;
    $at=$l_at+$l/2 ;
}
else                        # by default elements are butted up
{
    $at+=$l ;
}
print "dl ".$l.";\n"; # unknown element replaced by drift
$mk++ ;
}

sub beambeam
{
    sub beambeam_l
    {
        $mk+=2 ;
        $l=$field[$mk] ;
    }
sub beambeam_at
{
    $mk+=2 ;
    $l_at=$field[$mk] ;
    $flag_at=1 ;
}
sub beambeam_body
{
    $mk+=3 ;
    while ($field[$mk] ne "}") { $mk++ ; }
    print " { *** warning : beambeam_body not supported } \n" ;
}


@l_elements=("l","at","body","//") ;
$local="beambeam" ;

$l=$l_at=$flag_at=0 ;
$out="" ;
$mk+=2 ;
until ($field[$mk] eq "}")
{
    $flag=0 ;
    if ($field[$mk] eq "//") { &comment ; }
    else
    {
        for ($i=0;$i<=@l_elements-1 ; $i++)
        {
            if ($l_elements[$i] eq $field[$mk])
            { $flag=1 ; $l_type=$i ;
          }
        }
        $etype=$l_elements[$l_type] ;
        $etype=$local."_".$etype ;
```

```
        if ($flag==1) { print " --- $etype called \n"; &$etype}
        else { &unsupported ; }
    }
    $mk++ ;
}
if ($flag_at==1)                    # longitudinal positioning
of elements
{
    print " dl ".($l_at-$at-$l/2)." ;\n" ;
    $at=$l_at+$l/2 ;
}
else                        # by default elements are butted up
{
    $at+=$l ;
}
print " dl ".$l." ;\n"; # unknown element replaced by drift
$mk++ ;
}


# main programm
$at=0 ;
$mk=0 ;
$flag_sbend=0 ;      # employed to use the code for rbend fo
r sbends as well

print " procedure lattice ; \n" ;
print " variable MA 1 20 ; \n" ;
print " variable MS 1 20 ; \n" ;
print " variable D 1 ; \n\n" ;
print " D:=0.05 ; \n" ; # aperture set by virtue

# read input file and eliminate linebreaks, tabs amd semiko
lons
# parentheses and equal signs are surrounded by spaces

$line="" ;
while (<>)
{
#    print $_ ;
    s/\cM/ /g ;    # replaces Ctrl-M by blank
    s/\n/ /g ;
    s/\t/ /g ;
    s/\f/ /g ;
    s/=/ = /g ;
    s/\]/ \] /g ;
    s/\[/ \[ /g ;
    s/\{/ \{ /g ;
    s/\}/ \} /g ;
    s/\;/ /g ; # semicolon is ignored because it is redunda
nt
    if (/\/\//) { $_=$_." // " ; } # add // at end of comment
    # comment out the tags
    s/tag\s*=\s*(\S*)/\/\/ tag = $1 \/\//g ;
    $line.=" $_"
    }
```

```
#print $line ;
@field=split(/\s+/,$line) ;

# output for debugging purposes
# print " \n" ;
# for ($i=0 ; $i<@field ; $i++)
#  {
#    print " token number ".$i." : |".$field[$i]."|\n" ;
#  }
# print "\n" ;

# beginning of the translation

$mk=1 ; # skip trailing blank
print " --- number of tokens ".@field." \n" ;
while ($mk<@field)
{
    if (grep /$field[$mk]/,@elements)
    {
        if ($field[$mk] eq "//")
        { print " --- comment called \n"; &comment; }
        else
        {
            if ($field[$mk] ne "endsequence")
            {print " { ".$field[$mk-1]." } \n"; }
            print " --- $field[$mk] called \n" ; &{$field[$mk]} ;
        }
    }
    else
    { &unsupported ; }
    $mk++ ;
}
```

Bibliography

Bibliography

[1] M. Berz, *Differential Algebraic Description of Beam Dynamics to Very High Orders,* Published in "Particle Accelerators", Volume 24, 1989.

[2] Lecture Notes for the Michigan State University Course Physics 861, "Introductory Beam Physics" by Martin Berz , available upon request.

[3] M. Berz, *High-Order Computation and Normal Form Analysis of Repetitive Systems,* in: M. Month (Ed.), Physics of Particle Accelerators, vol. AIP 249, p. 456. American Institute of Physics, 1991.

[4] M. Berz, K. Makino, K.Shamseddine and W.Wan *Applications of Modern Map Methods in Particle Beam Physics,* Academic Press, Orlando, Florida, in print 1998

[5] H.Wiedemann, *Particle Accelerator Physics,* Springer Verlag 1993.

[6] M. Berz, *Truncated Power Series Techniques, Entry in 'Handbook of Accelerator Physics and Engineering', M. Tigner and A. Chao (Eds.)Particle Accelerator Physics,* World Scientific, New York, in preparation, 1997.

[7] M. Berz, *Symplectic Tracking in Circular Accelerators with High Order Maps,* Nonlinear Problems in Future Particle Accelerators World Scientific,1991.

[8] A. J. Dragt and J. M. Finn, *Normal Form for Mirror Machine Hamiltonians,* Journal of Mathematical Physics, 20(12), 1979 .

[9] A. Bazzani, *Normal forms for symplectic maps on* R^{2n}, Celestial Mechanics, 42, 1988 .

[10] K. Shamseddine and M. Berz, *Exception Handling in Derivative Computation with Nonarchimedean Calculus,* in: Computational Differentiation: Techniques, Applications, and Tools, M. Berz, C. Bischof, G. Corliss, A. Griewank (Eds.), SIAM, Philadelphia, 1996 .

[11] K. Shamseddine and M. Berz, *Nonarchimedean Structures as Differentiation Tools,* Proceedings, Beirut International Conference on Numerical Analysis, 1997 .

[12] M. Berz, *The new method of TPSA Algebra for the description of beam dynamics to high orders,* , Los Alamos National Laboratory, AT-6:ATN-86-16, 1986 .

[13] M. Berz, *The method of power series tracking for the mathematical description of beam dynamics,* , Nuclear Instruments and Methods, A258, 1987, 431.

[14] G. H. Hoffstaetter and M. Berz, *Symplectic Scaling of transfer maps including fringe fields,* Phys. Rev. E 54 (1996) 5664.

[15] M. Berz, *Computational aspects of design and simulation: COSY INFINITY,* NIM, A298 (1990) 473.

[16] K. Makino, M. Berz, *COSY INFINITY Version 8,* submitted to NIM

[17] COSY INFINITY Version 8 User's Guide and Reference Manual, MSUCL-1088, 1997.

[18] COSY INFINITY Homepage. [Online] Available http://www.bt.nscl.msu.edu/cosy

[19] FNAL LHC Homepage. [Online] Available http://www-ap.fnal.gov/LHC/

[20] F.Christoph Iselin, The MAD Program - User's Reference Manual, CERN/SL/90-13 (AP)

[21] F.Christoph Iselin, The MAD Program - Physics Methods Manual, CERN/SL/92 (AP)

[22] The Large Hadron Collider - Conceptual Design, CERN/AC/95-05 (LHC), October 1995

[23] G. Sabbi, *Magnetic Field Analysis of HGQ Coil Ends*, Fermilab TD-97-040 , September 1997.

[24] G. Sabbi, *HGQ 9711 Field Quality Table*, Fermilab TD-97-050 , November 1997.

[25] S.Russenschuck, C.Paul (CERN), K.Preis(TU Graz), *ROXIE : A Feature Based Design and Optimisation Program for Superconducting Accelerator Magnets*, CERN LHC Project Report 046 , September 1996.

[26] F. Méot and A. París, *Concerning Effects of Fringe Fields and Longitudinal Distribution of b_{10} in Low-β Regions on Dynamics in LHC*, FERMILAB-TM-2017, 1997

[27] Jacques Gareyte, *Accelerator Physics Issues of the LHC*, CERN LHC Project Report 10, Geneva, 24 June 1996

[28] M.Giovannozzi, W.Scandale, E.Todesco *Inverse Logarithmic Decay of Long Term Dynamic Aperture in Hadron Colliders*, CERN LHC Project Report 118, Jun 1997

[29] J.Brandt, A.Simmons, *Coil End Design for the LHC IR Quadrupole Magnet*, Fermilab TS-96-013 , November 1996.

[30] F.Pilat et al, RHIC/AP/(1998).

[31] S.Caspi, M.Helm, L.J.Laslett and V.O.Brady, *An Approach to 3D Magnetic Field Calculation Using Numerical and Differential Algebraic Methods*, SC-MAG-395, LBL-326240, Berkeley, July 17, 1992.

[32] S.Caspi, M.Helm and L.J.Laslett, *3D Field Harmonics*, SC-MAG-328, LBL-30313, Berkeley, March 30, 1991.

[33] I.S.Gradshteyn, I.M.Ryzhik , *Table of Integrals, Series and Products*, 1979

[34] W.Wan, C.Johnstone, J.Holt, M.Berz, K.Makino, M.Lindemann, B.Erdelyi, *The Influence of Fringe Fields on the Particle Dynamics in the Large Hadron Collider*, conference contribution to CPO5, April 1998, submitted to Nuclear Instruments and Methods