FERMILAB-THESIS-2017-21, FZJ-2018-02062

FIELD MODELING, SYMPLECTIC TRACKING, AND SPIN DECOHERENCE FOR EDM AND MUON *G*-2 LATTICES

By

Eremey Vladimirovich Valetov

A DISSERTATION

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

Physics – Doctor of Philosophy

2017

ABSTRACT

FIELD MODELING, SYMPLECTIC TRACKING, AND SPIN DECOHERENCE
FOR EDM AND MUON *G*-2 LATTICES

By

Eremey Vladimirovich Valetov

While the first particle accelerators were electrostatic machines, and several electrostatic storage rings were subsequently commissioned and operated, electrostatic storage rings pose a number of challenges. Unlike motion in the magnetic field, where particle energy remains constant, particle energy generally changes in electrostatic elements. Conservation of energy in an electrostatic element is, in practice, only approximate, and it requires careful and accurate design, manufacturing, installation, and operational use. Electrostatic deflectors require relatively high electrostatic fields, tend to introduce nonlinear aberrations of all orders, and are more challenging to manufacture than homogeneous magnetic dipoles. Accordingly, magnetic storage rings are overwhelmingly prevalent.

The search for electric dipole moments (EDMs) of fundamental particles is of key importance in the study of C and CP violations and their sources. C and CP violations are part of the Sakharov conditions that explain the matter–antimatter asymmetry in the universe. Determining the source of CP violations would provide valuable empirical insight for beyond-Standard-Model physics. EDMs of fundamental particles have not to this date been experimentally observed. The search for fundamental particle EDMs has narrowed the target search region; however, an EDM signal is yet to be discovered.

In 2008, Brookhaven National Laboratory (BNL) had proposed the frozen spin (FS) concept for the search of a deuteron EDM. The FS concept envisions launching deuterons through a storage ring with combined electrostatic and magnetic fields. The electrostatic and magnetic fields are in a proportion that would, without an EDM, freeze the deuteron's spin along its momentum as the deuteron moves around the lattice. The radial electrostatic field would result in a torque on the spin vector, proportional to a deuteron EDM, rotating the spin vector out of the midplane.

The principle of an anomalous magnetic dipole moment (MDM) measurement using a storage ring, shared by BNL's completed E821 Experiment and the ongoing E989 Experiment operated by Fermi National Accelerator Laboratory (FNAL), requires injecting muons into a magnetic ring at the so-called magic momentum. The magic momentum, as defined in this context, would freeze the muon's spin vector along its momentum if the anomalous MDM was zero. The spin precession in the horizontal plane relative to the momentum is proportional to the anomalous MDM.

Storage rings for measurement of EDM and anomalous MDM present a new frontier in tracking code accuracy requirements. For accurate tracking of storage rings with electrostatic particle optical elements, it is necessary to model the fringe fields of such elements accurately, in particular, because not doing so provides a mechanism for energy conservation violation. However, the previous research on fringe fields tended to focus on magnetic rather than electrostatic particle optical elements. We will study and model the fringe fields of several electrostatic deflectors. Field falloffs of electrostatic deflectors are slower than exponential, and Enge functions are not suitable for accurate modeling of these falloffs. We will propose an alternative function to model field falloffs of electrostatic deflectors. We will use conformal mapping methods to obtain the main field of the Muon g-2 storage ring high voltage quadrupole, and we will calculate its fringe field and effective field boundary (EFB) using Fourier analysis.

Furthermore, we will study tracking of storage rings with electrostatic elements using map methods. We will find that, for simultaneous symplecticity and energy conservation, it is only necessary to enforce symplecticity in *COSY INFINITY*. We will model and track several benchmark lattices – an electrostatic spherical deflector, a homogeneous magnetic dipole, and a proton EDM lattice – in *COSY INFINITY* and *MSURK89*, our in-house eighth order Runge–Kutta–Verner tracking code. Finally, we will investigate spin decoherence and systematic errors in FS and quasi-frozen spin (QFS) lattices. Spin decoherence effects are similar in FS and QFS lattices, and spin decoherence in said lattices often remains in the same range over time, indicating the feasibility of EDM measurement using FS and QFS lattices.

Copyright by
EREMEY VLADIMIROVICH VALETOV
2017

To life itself.

ACKNOWLEDGMENTS

I would like to express my most profound gratitude to and appreciation for the people who guided and inspired me during my years of study and research in the doctoral program.

First and foremost, I am forever indebted to my primary academic adviser, Martin Berz, for guiding me on this journey, for sharing his wisdom and patience with me, and for motivating me to pursue higher standards in scientific rigor and presentation clarity. I thank him for creating a vibrant, engaging, and diverse research experience, as well as for the opportunities to contribute value to meetings, conferences, and collaborations. Moreover, it is now hard to imagine not having experienced the advantages of applying his systematic and iterative approach to problem solving.

Many thanks to my adviser Yurij Senichev, who kindly offered his mentorship at the Institute of Nuclear Physics at Forschungszentrum Jülich, gave me the opportunity to work with him on frozen spin and quasi-frozen spin lattice design, and broadened my perspective of beam physics. I would like to thank my adviser Andreas Lehrach from the RWTH Aachen and the Institute of Nuclear Physics, Forschungszentrum Jülich, for welcoming me into the JEDI collaboration, for discussions about fringe fields, and for sending me to the International Computational Accelerator Physics Conference in Shanghai, China.

I am thankful to Kyoko Makino, who is a great role model in demonstrating scientific and technical attention to detail and who patiently worked with me on a report for the Muon g-2 collaboration and other projects.

My gratitude extends to the other members of my guidance committee, Phillip Duxbury and Filomena Nunes. I thank the graduate, high energy physics (HEP), and institute secretaries for their helpfulness.

I appreciate the collaboration, discussions, and other interactions with the scientists Helmut Soltner, Selcuk Haciomeroglu, Frank Rathmann, Bernd Lorentz, Wanwei Wu, Yannis Semertzidis, Jörg Pretz, Andro Kacharava, Lehel Banjai, and others. Helmut Soltner graciously performed multiple field data calculations for rectangular electrostatic deflectors using *COULOMB*'s boundary

element method (BEM) field solver.

It was a pleasure working with the current and former fellow student members of the MSU Center for Beam Theory and Dynamical Systems: Robert Hipple, David Tarazona, Ravi Jagasia, Ben Loseth, and Alexander Wittig. The same goes for the current and former fellow students at the Institute of Nuclear Physics, including Stanislav Chekmenev, Alexander Skawran, Alexander Aksentev, Marcel Rosenthal, and Vera Schmidt.

I acknowledge the Department of Energy and Forschungszentrum Jülich for providing conference travel funding and other support. I used resources of the National Energy Research Scientific Computing Center to track frozen spin and quasi-frozen spin lattices. Some calculations were performed using MSU's Obelix computing cluster or an Amazon EC2 computing cluster with MIT's *StarCluster* toolkit.

Finally, special thanks to my parents for their peaceful and enlightened parenting and for encouraging me to strive for academic excellence.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

KEY TO ABBREVIATIONS

BEM Boundary element method

BNL Brookhaven National Laboratory

C Charge

CCW Counter-clockwise

CKM Cabibbo–Kobayashi–Maskawa

CL Confidence level

CMBR Cosmic microwave background radiation

CP Charge and parity

CPU Central processing unit

CRDT Cross ratios of the Delaunay triangulation

CW Clockwise

DA Differential algebra

DFT Discrete Fourier transform

EDM Electric dipole moment

EFB Effective field boundary

FDAFD Finite difference approximation of the first derivative

FEM Finite element method

FFT Fast Fourier transform

FNAL Fermi National Accelerator Laboratory

FOM Figure of merit

JEDI Jülich Electric Dipole Moment Investigations

JPL Jet Propulsion Laboratory

LCS Laboratory coordinate system

LSCT Long spin coherence time

MDM Magnetic dipole moment

MIT Massachusetts Institute of Technology

MSU Michigan State University

NSM Non-symmetric model

ODE Ordinary differential equation

QCD Quantum chromodynamics

QED Quantum electrodynamics

RF Radio frequency

RMS Root mean square

RWTH Rheinisch-Westfälische Technische Hochschule

SC Schwarz–Christoffel

SCT Spin coherence time

SI International System of Units

SM Standard Model (of particle physics)

SM Symmetric model

STEP Standard for the exchange of product model data

SUSY Supersymmetry

Thomas–BMT Thomas–Bargmann–Michel–Telegdi

CHAPTER 1

INTRODUCTION

Particle accelerators are machines that use electromagnetic fields to accelerate atomic or subatomic particles to high energies. The applications of particle accelerators include experiments to research fundamental forces and matter, as well as mass spectrometry, isotope production, and medical diagnostics.

Because of the similarity between optics in optical systems and particle beam dynamics in particle accelerators, and for historical reasons, particle beam dynamics are also called particle beam optics.

In high energy physics, after the acceleration of particle bunches, it is often desirable to store them or to observe their dynamics for a period of time in a specific system or regime. This is achieved by storage rings, which are characterized by cyclic motion through a closed system of particle optical elements, often at a constant particle energy.

The first particle accelerators, such as the Van de Graaf accelerator and the Cockroft–Walton generator, were electrostatic machines [101]. There were a few electrostatic storage rings such as DESIREE in Stockholm [39] and, most notably, the 1954 Brookhaven Electrostatic AGS Analog Ring [85, 107].

However, electrostatic storage rings pose multiple challenges. Unlike motion in the magnetic field, where particle energy remains constant, particle energy generally changes in electrostatic elements. Conservation of energy in an electrostatic element is, in practice, only approximate, and it requires careful and accurate design, manufacturing, installation, and operational use. Moreover, electrostatic deflectors require relatively high electrostatic fields, tend to introduce nonlinear aberrations of all orders, and are more challenging to manufacture than homogeneous magnetic dipoles. Accordingly, magnetic storage rings are overwhelmingly prevalent.

1.1 Electric Dipole Moment

An electric dipole is a system characterized by centers of equal and opposite total charges $\pm q$ separated by a very short distance d. The electric dipole moment (EDM) of two point-like charges is defined as $p = qd$. A uniform electrostatic field E acts on an electric dipole of the EDM p with torque $\tau = p \times E$.

For a charge distribution $\rho(r)$ in a volume V, the EDM is

$$p(r) = \int_V \rho(r_0)(r_0 - r)\, d^3 r_0,$$

where r is the point of observation.

If the net charge Q in the volume V is zero, that is, $Q = \int_V \rho(r_0)\, d^3 r_0 = 0$, then the EDM $p(r)$ does not depend on the point of observation r. For charged particles, the net charge Q is not zero, and the EDM definition is specialized by setting the point of observation r as the center of mass of the particle [88].

EDMs of fundamental particles were not experimentally observed so far. However, one of the principal challenges in the Standard Model (SM) is baryon asymmetry, that is, the abundance of matter observed in the universe, in contrast to almost no antimatter. The baryon asymmetry is quantified by the baryon asymmetry parameter $\eta = \left(n_B - n_{\bar{B}}\right)/n_\gamma \simeq 5 \times 10^{-10}$, where n_B, $n_{\bar{B}}$, and n_γ are baryon, antibaryon, and photon number densities in the cosmic microwave background radiation (CMBR), respectively [49, 25].

If the baryon asymmetry existed as an initial condition at the beginning of the Big Bang, it would be eliminated by dilution during the inflationary period [25, 50]. The Sakharov conditions, proposed by Andrei Sakharov in his seminal 1967 article [92], represent a dynamic mechanism for baryon asymmetry generation, called baryogenesis, that does not require baryon asymmetry as an initial condition. The Sakharov conditions are as follows: (1) baryon number B violation, (2) C and CP violation, and (3) departure from thermal equilibrium [103, 92]. A nonzero EDM in a fundamental particle would result in both C and CP violation [88], which would support the Sakharov conditions as the reason for the baryon asymmetry.

The order of a neutron EDM due to C and CP violations is estimated as $|d_n| \simeq 10^{-23}\, e \cdot \text{cm}$ [5, p. 107]. The SM predicts, due to the CP-violating phase δ in the CKM matrix, $|d_n| \simeq 10^{-31}\, e \cdot \text{cm}$ [5, p. 112][74]. This suggests a target search region for a beyond-SM neutron EDM [89] as

$$10^{-31}\, e \cdot \text{cm} \lesssim |d_n| \lesssim 10^{-23}\, e \cdot \text{cm}.$$

This region applies to nucleon EDMs in general because of their order-wise approximate equivalency, as dimensional analysis and SUSY considerations show [3, pp. 7–9].

It is desirable to search for intrinsic EDMs of multiple particles in order to apply complementary methods over a broad search domain and to determine the source of CP violation from EDMs of multiple particles [89] (e.g., the CP-violating phase δ in the CKM matrix, the CP-violating parameter $\bar{\theta}$ of QCD, spontaneous CP violation in the Higgs field interaction [5, pp. 108–116][3, pp. 7–9]).

So far, the search for intrinsic particle EDMs has narrowed the EDM limits, for example, to $|d_e| < 1.85 \times 10^{-27}\, e \cdot \text{cm}$ for electrons (at 95.8% confidence level (CL); indirect measurement using the YbF molecules) [59, p. 29], $|d_n| < 3.6 \times 10^{-26}\, e \cdot \text{cm}$ for neutrons (at 95% CL) [84, p. 21], and $\left|d_p\right| < 7.9 \times 10^{-25}\, e \cdot \text{cm}$ for protons (including 30% theoretical uncertainty; based on measurement of $^{199}_{80}$Hg EDM at 95% CL) [44, p.101601-4], but an EDM signal is yet to be discovered. Experiments continue with increasing measurement precisions, and an expansion of the search domain to include deuterons is envisioned.

The idea of using storage rings for the search of charged hadron EDMs is an alternative with the potential of higher sensitivity than current state-of-the-art methods, including ultracold neutron, paramagnetic atom, and diamagnetic atom EDM experiments [5, 40].

In 2008, BNL's Storage Ring EDM collaboration had proposed the frozen spin (FS) concept [2] for measurement of the deuteron EDM, illustrated in Fig. 1.1. The FS concept envisions launching deuterons through a storage ring with combined radial electrostatic and vertical magnetic fields. The electrostatic (E) and magnetic (B) fields are in a proportion that would, without an EDM, freeze the deuteron's spin[1] along its momentum as the deuteron moves around the lattice. The radial

[1] Spin is the intrinsic angular momentum of a particle, as opposed to orbital angular momentum.

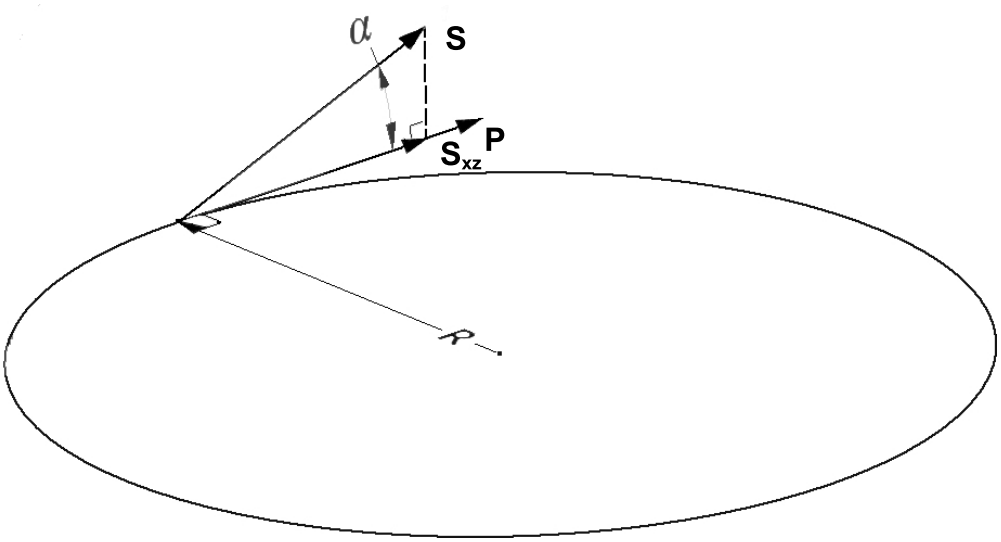


Figure 1.1: In the FS lattice concept, the projection s_{xz} of the deuteron's spin s on the midplane x–z is aligned with the deuteron's momentum p. An deuteron EDM would cause the azimuthal angle α to slowly grow from the initial $\alpha = 0$ to a measurable value.

electrostatic field would result in a torque on the spin vector, proportional to a deuteron EDM, rotating the spin vector out of the midplane.

Measurement of the proton EDM using the FS method had been proposed by the Storage Ring EDM collaboration in 2011 [3]. The Jülich Electric Dipole moment Investigations (JEDI) collaboration, based at the Institute of Nuclear Physics at Forschungszentrum Jülich, is considering the FS method for design of a deuteron EDM storage ring, as well as its modification called the quasi-frozen spin (QFS) method [65, 98]. We will introduce the QFS method in sec. 6.3.2.1.

1.2 Anomalous Magnetic Dipole Moment

The magnetic dipole moment (MDM) μ is defined by the relation $\tau = \mu \times B$, where τ is the torque exerted on an object, such as a magnet, by an external magnetic field B.

The spin MDM of a lepton (an electron e^-, a muon μ^-, or a tau τ^-) is $\mu = g\frac{e}{2m}s$, where the lepton spin is $s = 1/2$, m is the lepton mass, e is the elementary charge, and g is the g-factor (gyromagnetic ratio) of the lepton.

The Dirac equation predicts the g-factor as 2 for leptons, and the quantity $a = (g-2)/2$, arising from quantum effects, is known as anomalous MDM (or MDM anomaly).

The SM-framework prediction for the muon anomalous MDM consists of three parts [52, 14]:

$$a_\mu^{\mathrm{SM}} = a_\mu^{\mathrm{QED}} + a_\mu^{\mathrm{EW}} + a_\mu^{\mathrm{Had}},$$

where QED, EW, and Had denote the QED, electroweak, and hardronic contributions, respectively.

The muon anomalous MDM is predicted by the SM as [14, 23, 48]

$$a_\mu^{\mathrm{SM}} = \begin{cases} 116591802\,(49) \times 10^{-11} & \text{in [23, p. 10]}, \\ 116591828\,(49) \times 10^{-11} & \text{in [48, p. 21]}. \end{cases}$$

It is desirable to experimentally validate the SM prediction of the muon anomalous MDM.

The Muon g-2 Experiment E821 at BNL [53], completed in 2001, measured the muon anomalous MDM as [14, p. 8][43, p. 46]

$$a_\mu^{\mathrm{E821}} = 116592089\,(63) \times 10^{-11},$$

with an experimental precision of 0.54 ppm – a significant improvement over previous muon anomalous MDM measurements (see, e.g., [19, 21, 20, 35]).

The resulting discrepancy between the experimental and theoretical values was [14, 23, 48]

$$\Delta a_\mu = a_\mu^{\mathrm{E821}} - a_\mu^{\mathrm{SM}} = \begin{cases} 287\,(80) \times 10^{-11} & \text{[23, p. 10]}, \\ 261\,(80) \times 10^{-11} & \text{[48, p. 22]}. \end{cases} \tag{1.1}$$

These results represent 3.6σ and 3.3σ disagreements between experiment and theory, respectively, and are not at the discovery threshold of 5σ.

In 2010, the new Muon g-2 Experiment E989 [17, 43], now at FNAL, was proposed to measure the muon anomalous MDM, using the same measurement principle, to 0.14 ppm experimental precision. That should exceed the 5σ discovery threshold, assuming the discrepancy between experiment and theory remains similar to the discrepancy in eq. 1.1.

The principle of anomalous MDM measurement, shared by the E821 and E989 Experiments, requires injecting muons into a magnetic ring at the magic momentum $p = 3.09\,\mathrm{GeV/c}$, corresponding to $\gamma = 29.3$ [53, p. 9][43, p. 73]. The magic momentum, as defined in this context, would

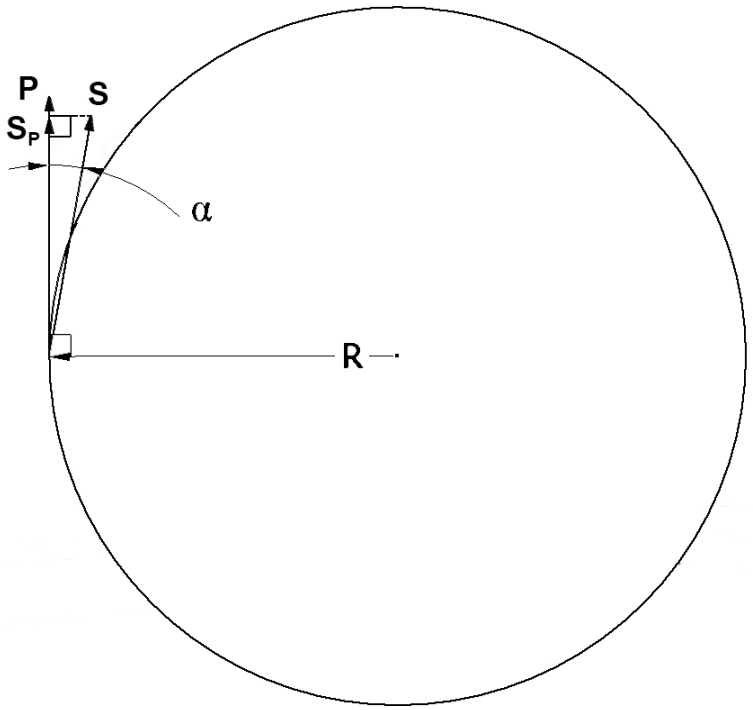


Figure 1.2: In the E821 and E989 lattice concepts, the polar angle α between the muon's spin vector s and momentum p is initially at $\alpha = 0$. A nonzero muon anomalous MDM a_μ causes α to slowly grow to a measurable value.

freeze the muon's spin vector along its momentum if a_μ was zero. Thus, the spin precession in the horizontal plane relative to the momentum is proportional to a_μ. This lattice concept is illustrated in Fig. 1.2.

1.3 Thomas–BMT Equation

The Thomas–BMT equation describes the dynamics of spin vector s in magnetic field B and electrostatic field E, and it is generalized to account for the EDM effects as follows [79, 37]:

$$\frac{ds}{dt} = s \times (\Omega_{\text{MDM}} + \Omega_{\text{EDM}}),$$

where the MDM and EDM angular frequencies Ω_{MDM} and Ω_{EDM} are

$$\Omega_{\text{MDM}} = \frac{q}{m}\left[GB - \left(G - \frac{1}{\gamma^2 - 1}\right)\frac{E \times \beta}{c}\right],$$
$$\Omega_{\text{EDM}} = \frac{q}{m}\frac{\eta}{2}\left[\frac{E}{c} + \beta \times B\right],$$

where m, q, G are the particle mass, electric charge, and anomalous MDM, respectively; β is the ratio of particle velocity to the speed of light; and γ is the Lorentz factor. The EDM factor η is defined by $d = \eta \frac{q}{2mc} s$, where d is the particle EDM and s is the particle spin.

The magic momentum, utilized in the anomalous MDM and the proton EDM storage ring concepts, is a momentum that causes the coefficient of the $\beta \times E$ term in the Thomas–BMT equation to vanish. The magic momentum corresponds to the Lorentz factor $\gamma = \sqrt{(1+G)/G}$.

As the deuteron has negative anomalous MDM, the magic momentum concept is not applicable. Accordingly, the deuteron EDM storage ring concept uses combined $E+B$ fields in such proportion that the entire Ω_{MDM} term vanishes. This is achieved by a radial electrostatic field (compare with [2, p. 10])

$$E = \frac{BcG\beta}{1+\beta^2(1+G)} \approx BcG\beta\gamma^2,$$

where B is the vertical magnetic field and G is the deuteron's anomalous MDM.

1.4 Lattice Tracking Requirements

Lattice simulation and tracking is an essential part of lattice concept feasibility studies, design, and post-commissioning operation.

To validate the feasibility of a lattice for EDM or anomalous MDM, it is necessary to perform spin coherence time (SCT) and systematic errors analysis studies, as well as to simulate measurement of an EDM or an anomalous MDM signal. EDM and anomalous MDM storage rings are high-precision measurement devices and, accordingly, require highly accurate orbital and spin tracking.

A broadly manifesting challenge of EDM and anomalous MDM storage rings is that they require the magic momentum or a γ-dependent proportion of E and B fields; however, generally, electrostatic fields affect particle energy. Symplecticity (see Ch. 4 or [4, 41]) is a fundamental property of Hamiltonian mechanics and results, among other things, in the conservation of phase space volume by Liouville's theorem. For some time, there was a general notion that numerical tracking of nonintegrable systems cannot simultaneously preserve energy and symplecticity [117]. Practically all storage rings are magnetic, and tracking codes tend to have specialized control

mechanisms for symplecticity of motion but not energy conservation [11, pp. 292–295].

Furthermore, for accurate tracking of storage rings with electrostatic particle optical elements, it is necessary to accurately model fringe fields of electrostatic particle optical elements. However, the previous research on fringe fields tended to focus on magnetic (e.g., [82, 83, 112]) rather than electrostatic particle optical elements (e.g., the 2016 article [73]), and previous results on fringe field models of electrostatic particle optical elements are limited.

1.5 Beamline Coordinate System

Particle beam lattices, such as accelerators and storage rings, are typically designed with an ideal particle trajectory called the reference orbit. In storage rings, the beam transport is periodic, and the reference orbit is closed.

Beamline coordinate systems use the distance s along the reference orbit (the arc length) as a natural parametrization of motion.

Let $r(s)$ be the coordinate vector of a particle in a global Cartesian coordinate system as a function of s, and let the index 0 refer to the reference particle.

Each point s on the reference orbit is the origin of a local orthonormal basis. The Frenet–Serret frame is used as the local orthonormal basis. The Frenet–Serret frame comprises tangent e_z, normal e_x, and binormal e_y unit basis vectors [11, pp. 168–190][105, pp. 15–18]

$$
\begin{aligned}
e_z(s) &= \frac{r_0'(s)}{\left|r_0'(s)\right|}, \\
e_x(s) &= \frac{e_z'(s)}{\left|e_z'(s)\right|}, \\
e_y(s) &= e_z(s) \times e_x(s).
\end{aligned}
$$

The Frenet–Serret unit vectors e_x, e_y, and e_z are related to the reference orbit curvature κ and torsion τ by the Frenet–Serret formulas

$$
\begin{aligned}
e_x'(s) &= -\kappa(s)\, e_x(s) + \tau(s)\, e_z(s), \\
e_y'(s) &= -\tau(s)\, e_x(s),
\end{aligned}
$$

$$e_z'(s) = \kappa(s)\, e_y(s)\,.$$

We will primarily perform beam physics computations in *COSY INFINITY* [68]. Accordingly, we will use *COSY INFINITY*'s beamline coordinate system with phase space coordinates [12, p. 9]

$$\begin{aligned} r_1 &= x, & r_2 &= a = p_x/p_0, \\ r_3 &= y, & r_4 &= b = p_y/p_0, \\ r_5 &= l = -(t-t_0)\, v_0 \tfrac{\gamma}{1+\gamma}, & r_6 &= \delta_K = \tfrac{K-K_0}{K}. \end{aligned}$$

Coordinates x and y are the transversal Frenet–Serret position coordinates in meters, p is the momentum, K is the kinetic energy, v is the velocity, t is the time of flight, and γ is the Lorentz factor. The index 0 refers to the reference particle.

The six phase space coordinates form three canonically conjugate pairs (x,a), (y,b), and (l,δ_K).

In some instances, we will use a global Cartesian coordinate system (x,y,z) based on the local Frenet–Serret coordinate system $(x(s), y(s), z(s))$ at a fixed point s. This is mostly useful for individual particle optical elements with a straight reference orbit. In that case, the s defining the coordinate system, and its origin, are typically at the physical exit edge or the exit effective field boundary (EFB) of the element.

1.6 Thesis Outline

First, we will address the issue of accurate description of main and fringe fields of particle optical elements used in spin lattices. In **Chapter 2**, we will calculate the fringe fields of several semi-infinite electrostatic deflectors using conformal mapping methods, compare the results with the fringe fields of several finite rectangular electrostatic deflectors obtained using a boundary element method (BEM) field solver, and propose a function to model field falloffs of electrostatic deflectors. In **Chapter 3**, we will use conformal mapping methods to obtain the main field of the Muon g-2 storage ring high voltage quadrupole, and we will calculate its fringe field and EFB using Fourier analysis.

Next, we will study tracking of storage rings with electrostatic elements using map methods. In **Chapter 4**, we will review symplecticity and symplectification methods, discuss the feasibility

of simultaneous symplectification and energy correction, and conclude that it is only necessary to enforce symplecticity in *COSY INFINITY*, as energy conservation was foundational in the tracking code design. In **Chapter 5**, we will model and track several benchmark lattices – an electrostatic spherical deflector, a homogeneous magnetic dipole, and a Proton EDM lattice – in *COSY INFINITY* and *MSURK89*, our in-house 8th order Runge–Kutta–Verner tracking code. Finally, we will investigate spin decoherence and systematic errors in FS and quasi-frozen spin (QFS) lattices in **Chapter 6**. We will find that the spin decoherence effects are similar in FS and QFS lattices, and spin decoherence often remains in the same range over time, indicating the possible feasibility of EDM measurement using FS and QFS lattices.

CHAPTER 2

FRINGE FIELDS OF ELECTROSTATIC DEFLECTORS

2.1 Introduction

The electrostatic deflector is a core particle optical element type in a lattice for EDM measurement.

Electrostatic deflectors used in beam physics generally comprise two plates of different voltages V_1 and V_2. In the approximation of an electrostatic deflector as a capacitor with parallel flat plates and the reference orbit with zero curvature, nominally, plate voltages have the same magnitude but opposite signs: $V_1 = V$ and $V_2 = -V$.

The distance D between the plates – the full aperture – is usually small compared to plate widths and lengths and the deflector's radius R of curvature. This leads to the idea that fringe fields of realistic electrostatic deflectors can be effectively modeled by fringe fields of semi-infinite capacitors.

We studied the fringe fields of semi-infinite electrostatic capacitors. In many cases, a semi-infinite electrostatic capacitor can be modeled by a generalized polygon with two vertices at the infinity separating two groups of sides characterized by two different constant voltages. The electrostatic field of this capacitor may then be obtained by finding a conformal mapping from a bi-infinite strip to this polygon.

We compared the field falloffs obtained using conformal mapping methods for several semi-infinite electrostatic capacitors to the field falloffs of two finite electrostatic capacitors obtained using a BEM field solver.

Field falloffs of semi-infinite electrostatic capacitors are slower than exponential. We think that this applies to electrostatic deflectors in general, as substantiated by the example of a finite cylindrical electrostatic deflector in Ch. 5. Thus, whereas Enge functions are useful for modeling falloffs of magnetic particle optical elements, they are not generally suitable for electrostatic

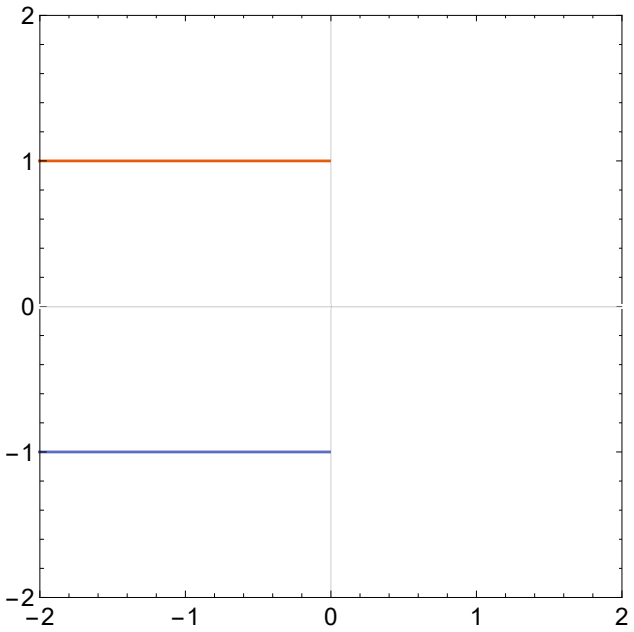


Figure 2.1: The polygonal model of the semi-infinite capacitor with infinitely thin plates.

deflectors. We propose an alternative function to model field falloffs of electrostatic deflectors, and we demonstrate its use in the example of a semi-infinite electrostatic capacitor.

2.2 Semi-Infinite Capacitor with Uniform Charge Distribution

The simplest model of a semi-infinite capacitor comprises infinitely thin semi-planes and assumes uniform charge distribution in the plates.

We consider such a model with plates at horizontal offsets $x = \pm d$ and longitudinal coordinates $z \leq 0$, shown in Fig. 2.1. The z axis is a reference orbit in the approximation of no fringe fields and the radius-of-curvature limit of $R \to +\infty$. The full aperture is $D = 2d$. Without limiting the generality, we set the surface charge densities as $\sigma = 1$ for the $x = d$ plate and $\sigma = -1$ for the $x = -d$ plate.

The electrostatic potential $\varphi(x,y,z)$ between the plates can be obtained as the sum of integrals of the Coulomb potential over the two plates

$$\varphi(x,y,z) = \int_{-\infty}^{+\infty} dt_1 \int_{-\infty}^{0} dt_2 \sum_{j=\pm 1} \frac{|\sigma| j}{\sqrt{(x-jd)^2 + t_1^2 + (z-t_2)^2}}. \tag{2.1}$$

Performing integration over t_1 within eq. 2.1, we have

$$
\begin{aligned}
&\int_{-\infty}^{+\infty} dt_1 \sum_{j=\pm 1} \frac{|\sigma|\, j}{\sqrt{(x-jd)^2+t_1^2+(z-t_2)^2}} = \\
&= \lim_{t_1\to+\infty} \sum_{j=\pm 1} j \ln\left[\frac{t_1+\sqrt{(x-jd)^2+t_1^2+(z-t_2)^2}}{-t_1+\sqrt{(x-jd)^2+t_1^2+(z-t_2)^2}}\right] = \\
&= \lim_{t_1\to+\infty} \ln\left[\frac{\left(\sqrt{\xi^2+t_1^2+4dx}-t_1\right)\left(\sqrt{\xi^2+t_1^2}+t_1\right)}{\left(\sqrt{\xi^2+t_1^2+4dx}+t_1\right)\left(\sqrt{\xi^2+t_1^2}-t_1\right)}\right] = \\
&= \lim_{t_1\to+\infty} \ln\left[1+\frac{2t_1\left(\sqrt{\xi^2+t_1^2+4dx}-\sqrt{\xi^2+t_1^2}\right)}{\left(\sqrt{\xi^2+t_1^2+4dx}+t_1\right)\left(\sqrt{\xi^2+t_1^2}-t_1\right)}\right] = \\
&= \lim_{t_1\to+\infty} \ln\left[1+\frac{2\left(\frac{2dx}{t_1}+O\left(\frac{1}{t_1^3}\right)\right)}{\left(\sqrt{\frac{\xi^2+4dx}{t_1^2}+1}+1\right)\left(\frac{\xi^2}{2t_1}+O\left(\frac{1}{t_1^3}\right)\right)}\right] = \\
&= \ln\left(1+\frac{4dx}{(x-d)^2+(z-t_2)^2}\right),
\end{aligned}
\tag{2.2}
$$

assuming $x \neq d$ or $t_2 \neq z$, and where we denoted $\xi^2 = (x-d)^2+(z-t_2)^2$.

Let

$$
f(x,z) = \left[(t_2-z)\ln\left(1+\frac{4dx}{(x-d)^2+(z-t_2)^2}\right)\right]_{t_2=-\infty}^{0}. \tag{2.3}
$$

Evaluating the right-hand side in eq. 2.3, we obtain

$$
\begin{aligned}
f(x,z) &= -z\ln\left(1+\frac{4dx}{(x-d)^2+(z-t_2)^2}\right) = \\
&= z\ln\left[\frac{(x-d)^2+(z-t_2)^2}{(d+x)^2+(z-t_2)^2}\right] = \\
&= z\ln\left(1-\frac{4dx}{(d+x)^2+(z-t_2)^2}\right).
\end{aligned}
\tag{2.4}
$$

Considering eqns. 2.2, 2.3, and 2.4, we complete the integration in eq. 2.1:

$$\begin{aligned}
\varphi(x,y,z) &= \int_{-\infty}^{+\infty} dt_1 \int_{-\infty}^{0} dt_2 \sum_{j=\pm 1} \frac{|\sigma|\, j}{\sqrt{(x-jd)^2+t_1^2+(z-t_2)^2}} = \\
&= \int_{-\infty}^{0} dt_2 \ln\left(1+\frac{4dx}{(x-d)^2+(z-t_2)^2}\right) = \\
&= f(x,z) + \int_{-\infty}^{0} dt_2 \frac{(t_2-z)^2}{1+\frac{4dx}{(x-d)^2+(z-t_2)^2}} \frac{8dx}{\left[(x-d)^2+(z-t_2)^2\right]^2} = \\
&= f(x,z) + \int_{-\infty}^{0} dt_2 \frac{8dx\,(t_2-z)^2}{\left[(x-d)^2+(z-t_2)^2\right]\left[(x+d)^2+(z-t_2)^2\right]} = \\
&= f(x,z) + \int_{-\infty}^{0} dt_2 \left[-\frac{2(d-x)^2}{(d-x)^2+(t_2-z)^2} + \frac{2(d+x)^2}{(d+x)^2+(t_2-z)^2}\right] = \\
&= f(x,z) + \left[2(x-d)\tan^{-1}\left(\frac{t_2-z}{d-x}\right) + 2(d+x)\tan^{-1}\left(\frac{t_2-z}{d+x}\right)\right]_{t_2=-\infty}^{0} = \\
&= f(x,z) + 2(x-d)\tan^{-1}\left(-\frac{z}{d-x}\right) + 2(d+x)\tan^{-1}\left(-\frac{z}{d+x}\right) + 2\pi x = \\
&= f(x,z) + 2(x-d)\tan^{-1}\left(\frac{d-x}{z}\right) + 2(d+x)\tan^{-1}\left(\frac{d+x}{z}\right) + 4\pi x\,[z<0]
\end{aligned}$$

for $-d < x < d$ and $z \neq 0$, where the Iverson bracket was used, defined as [42, p. 24] $[P] = 1$ if P is true and $[P] = 0$ if P is false.

Thus, we have

$$\begin{aligned}
\varphi(x,y,z) = z\ln\left(1-\frac{4dx}{(d+x)^2+z^2}\right) + 2(x-d)\tan^{-1}\left(\frac{d-x}{z}\right) + \\
+ 2(d+x)\tan^{-1}\left(\frac{d+x}{z}\right) + 4\pi x\,[z<0]
\end{aligned}$$

for $z \neq 0$.

The singularity of the electrostatic potential φ and its derivatives at $z = 0$ is removable and has no physical meaning. We remove the singularity to make φ continuous and infinitely differentiable at $z = 0$.

We obtain the x component $E_x(0,y,z)$ of the electrostatic field in the midplane $x = 0$ from the

electrostatic potential as

$$
\begin{aligned}
E_x(0,y,z) &= -\frac{\partial}{\partial x}\varphi(x,y,z)\,|_{x=0} = \\
&= -\lim_{w\to z}\left[\left(\frac{8dxw(d+x)}{\left((d+x)^2+w^2\right)^2}-\frac{4dw}{(d+x)^2+w^2}\right)\left(1-\frac{4dx}{(d+x)^2+w^2}\right)^{-1}-\right. \\
&-\frac{2w(x-d)}{w^2+(d-x)^2}+2\tan^{-1}\left(\frac{d-x}{w}\right)+\frac{2w(d+x)}{w^2+(d+x)^2}+2\tan^{-1}\left(\frac{d+x}{w}\right)+ \\
&\left.+4\pi\,[w<0]\right]_{x=0} = \\
&= \begin{cases} -4\cot^{-1}(\frac{z}{d})-4\pi & ,z<0, \\ -4\cot^{-1}(\frac{z}{d}) & ,z>0, \\ -2\pi & ,z=0. \end{cases}
\end{aligned}
$$

Normalizing the electrostatic field to $E_x(0,y,-\infty)=1$ (i.e., $\lim_{z\to-\infty}E_x(0,y,z)=1$) and considering the translational symmetry in the y direction, we obtain

$$
E_x(z)=E_x(0,y,z)=\begin{cases} \frac{1}{\pi}\cot^{-1}(\frac{z}{d})+1 & ,z<0, \\ \frac{1}{\pi}\cot^{-1}(\frac{z}{d}) & ,z>0, \\ 1/2 & ,z=0, \end{cases}
$$

Fig. 2.2 shows the falloff of the electrostatic field $E_x(z)$.

The electrostatic field satisfies the condition $E_x(z)+E_x(-z)=1$. This is due to the assumption that the surface charge densities are uniform. As a result of this assumption, the surface charge densities $\pm\sigma$ in the plates are symmetric to the zero surface charge density in their complementary semi-planes at $x=\pm d$ and $z>0$. More generally, the equality $E_x(z)+E_x(-z)=\text{const}$ would hold for any charge density $\rho(x,y,z)$ such that $\rho(x,y,z)+\rho(x,y,-z)=\text{const}$.

2.2.1 Taylor Enge Function Coefficients

To approximate the falloff of the electrostatic field $E_x(z)$ by an n-th order Enge function

$$
F_n(z)=\frac{1}{1+\exp\left(\sum_{j=1}^{n+1}a_j(z/D)^{j-1}\right)},
$$

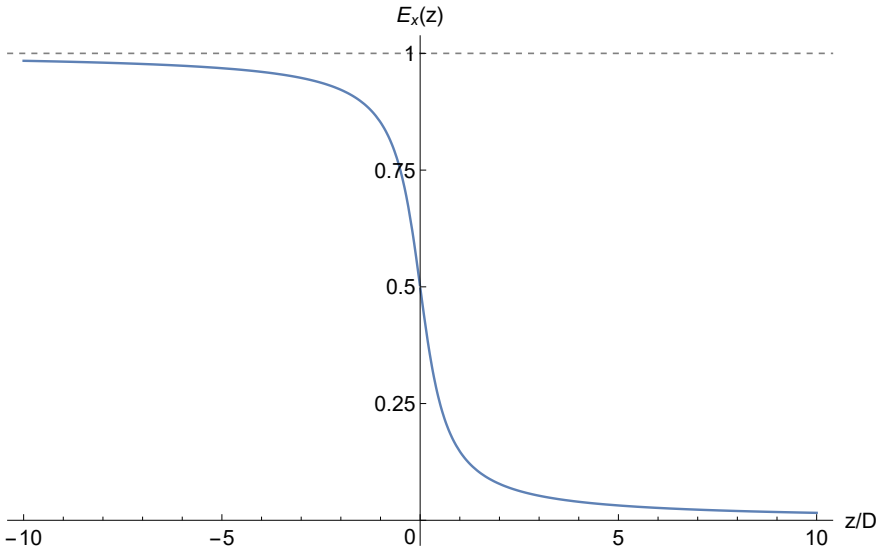


Figure 2.2: Electrostatic field falloff $E_x(z)$ of the semi-infinite capacitor with uniform charge distribution.

one may think of performing an n-th order Taylor expansion of the function $f(z)$ defined by

$$\frac{1}{1+\exp f(z)} = E_x(z).$$

The abscissa of an Enge function is always scaled by the full aperture D of the particle optical element that it represents. This is because electrostatic and magnetic fields scale with geometry.

We had obtained the 5th order Taylor series expansion of

$$f(z) = \ln\left(\frac{1}{E_x(z)} - 1\right)$$

in *Mathematica* as

$$f(z) = \frac{8}{\pi}\left(\frac{z}{D}\right) - \frac{32\left(\pi^2-4\right)}{3\pi^3}\left(\frac{z}{D}\right)^3 + \frac{128\left(48-20\pi^2+3\pi^4\right)}{15\pi^5}\left(\frac{z}{D}\right)^5 + O\left[\left(\frac{z}{D}\right)^6\right],$$

resulting in Taylor Enge function coefficients $a_1 = 0$, $a_2 = 8/\pi$, $a_3 = 0$, $a_4 = 32\left(4-\pi^2\right)/3\pi^3$, $a_5 = 0$, and $a_6 = 128\left(48-20\pi^2+3\pi^4\right)/15\pi^5$.

For an odd function $f(z)$, the Enge function satisfies the symmetry condition $F(z)+F(-z) = 1$. Indeed, assuming that $f(-z) = -f(z)$,

$$F(z)+F(-z) = 1 \Leftrightarrow \frac{1}{1+\exp f(z)} + \frac{1}{1+\exp f(-z)} = 1 \Leftrightarrow$$

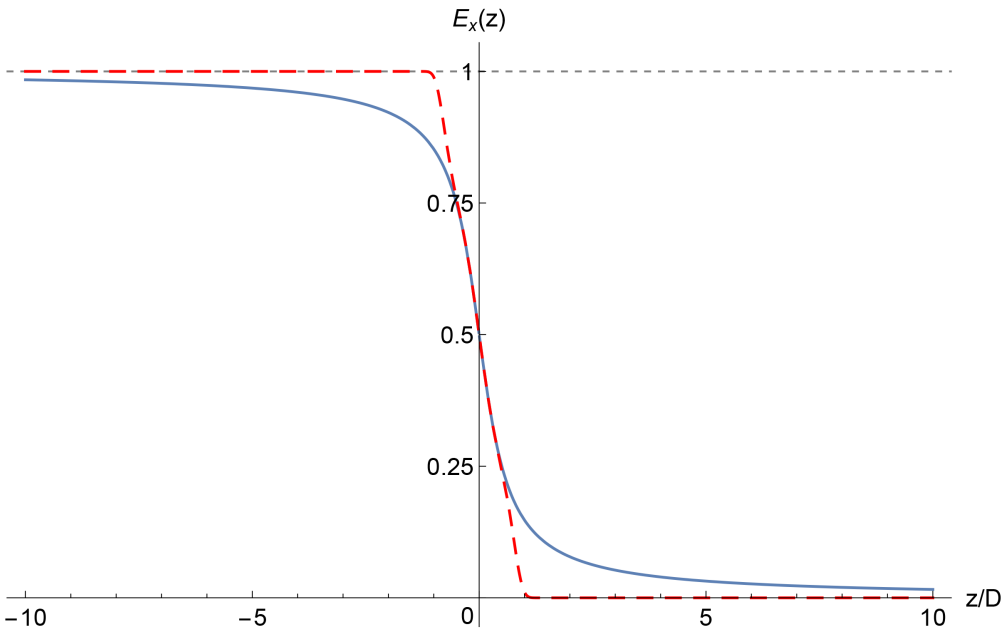


Figure 2.3: 5th order Taylor Enge function $F_5(z)$ (dashed red), based on the electrostatic field falloff $E_x(z)$ (solid blue) of the semi-infinite capacitor with uniform charge distribution.

$$\Leftrightarrow 1 = \exp\left(f(z) + f(-z)\right) \Leftrightarrow$$

$$\Leftrightarrow f(-z) = -f(z).$$

A plot of the Taylor Enge function $F_5(z)$ in Fig. 2.3 demonstrates that, generally, finite-order Taylor expansions do not produce the best polynomial fit of the respective order. However, Taylor function coefficients may be used as the initial guess in a least-squares fit.

One may be tempted to calculate Taylor Enge function coefficients of a higher order to obtain a better approximation. The following are the 16th order Taylor Enge function coefficients of $E_x(z)$, computed as a differential algebra (DA) value of $f(z/D)$ at $z = 0$ in *COSY INFINITY*:

I	COEFFICIENT	ORDER	EXPONENTS
1	2.546479089470326	1	1 0 0 0 0 0
2	−2.019239983477256	3	3 0 0 0 0 0
3	−.8881784197001252E−15	4	4 0 0 0 0 0
4	3.982947197900324	5	5 0 0 0 0 0
5	−.5329070518200751E−14	6	6 0 0 0 0 0
6	−10.10612663161852	7	7 0 0 0 0 0

7	−.3552713678800501E−14	8	8	0	0	0	0
8	28.89989597659871	9	9	0	0	0	0
9	−.1278976924368180E−11	10	10	0	0	0	0
10	−88.63211876076747	11	11	0	0	0	0
11	0.1051603248924948E−11	12	12	0	0	0	0
12	284.5981022831754	13	13	0	0	0	0
13	−.3922195901395753E−11	14	14	0	0	0	0
14	−943.9943887193954	15	15	0	0	0	0
15	−.6893969839438796E−09	16	16	0	0	0	0

For comparison, the following is the 9th order Taylor series expansion of $f(z)$, obtained by symbolic differentiation in *Mathematica*:

$$
\begin{aligned}
f(z) = & \frac{2048\left(80640-60480\pi^2+19152\pi^4-3272\pi^6+315\pi^8\right)}{2835\pi^9}\left(\frac{z}{D}\right)^9- \\
& -\frac{512\left(-2880+1680\pi^2-392\pi^4+45\pi^6\right)}{315\pi^7}\left(\frac{z}{D}\right)^7+ \\
& +\frac{128\left(48-20\pi^2+3\pi^4\right)}{15\pi^5}\left(\frac{z}{D}\right)^5-\frac{32\left(\pi^2-4\right)}{3\pi^3}\left(\frac{z}{D}\right)^3+\frac{8}{\pi}\left(\frac{z}{D}\right)+O\left[\left(\frac{z}{D}\right)^{10}\right]= \\
& =2.546479089470325\left(\frac{z}{D}\right)-2.019239983477256\left(\frac{z}{D}\right)^3+3.982947197900328\left(\frac{z}{D}\right)^5- \\
& -10.10612663161852\left(\frac{z}{D}\right)^7+28.89989597659899\left(\frac{z}{D}\right)^9+O\left[\left(\frac{z}{D}\right)^{10}\right].
\end{aligned}
$$

The CPU times of Taylor Enge function coefficients calculations, up to the 63rd order, were 6.91 s and 0.0125 s in *Mathematica* and *COSY INFINITY*, respectively.

As Fig. 2.4 illustrates, increasing the order of a Taylor Enge function tends to result in faster exponential convergence to the asymptotes, decreasing the Enge function approximation accuracy. Moreover, actual and apparent asymptotes alternate between the intended $\lim_{z\to-\infty}F(z)=1$, $\lim_{z\to+\infty}F(z)=0$ and the unintended $\lim_{z\to-\infty}F(z)=0$, $\lim_{z\to+\infty}F(z)=1$, depending on the leading polynomial terms of the Taylor expansion of $f(z)$.

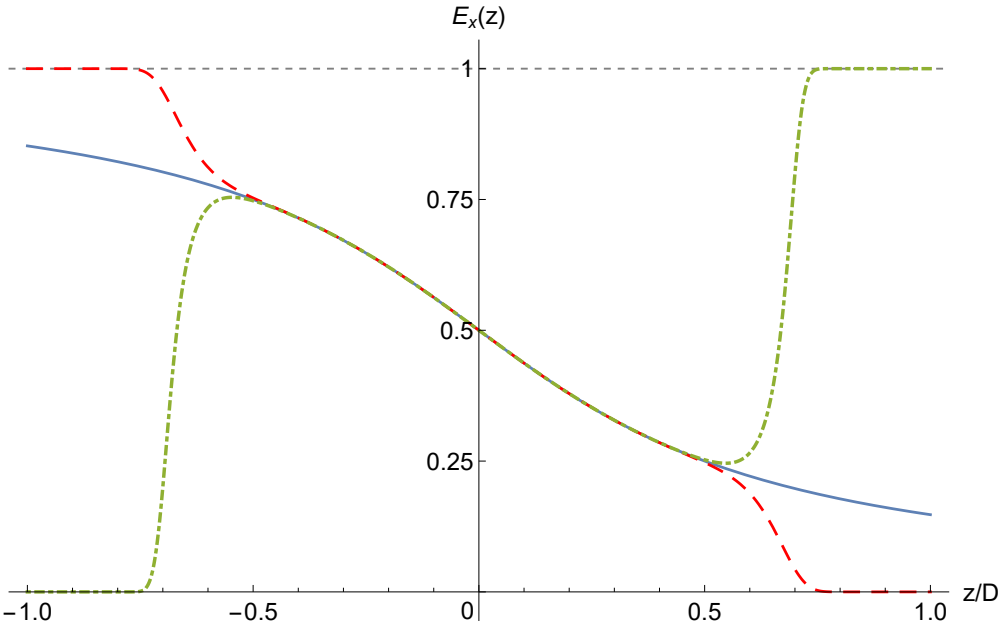


Figure 2.4: 14th (dot-dashed green) and 16th (dashed red) Taylor Enge functions $F_{14}(z)$ and $F_{16}(z)$, based on the electrostatic field falloff $E_x(z)$ (solid blue) of the semi-infinite capacitor with uniform charge distribution.

2.2.2 Least-Squares Enge Function Coefficients

We fitted a 5th order Enge function $F_5(z)$ to the electrostatic field $E_x(z)$ using the Levenberg–Marquardt Gauss–Newton method and the least-squares variance function in *COSY INFINITY* and *Mathematica*. The set $a_1 = 6.2118 \times 10^{-17}$, $a_2 = 1.87847$, $a_3 = 3.35814 \times 10^{-17}$, $a_4 = -0.136631$, $a_5 = 1.09036 \times 10^{-18}$, and $a_6 = 0.00406222$, obtained in *Mathematica*, is a representative example.

The odd Enge function coefficients were practically zero, which agrees with the symmetry condition $F_5(z) + F_5(-z) = 1$.

Although the least-squares fits resulted in improvements over the 5th order Taylor Enge function coefficients, the representation accuracy was still not satisfactory, as Fig. 2.5 illustrates. This is because the electrostatic field falloff is slower than exponential.

We concluded that although Enge functions often accurately represent field falloffs, it is not the case for the semi-infinite capacitor with infinitely thin plates and uniform charge distribution. As we will show later, Enge functions are not good representations of electrostatic deflector field

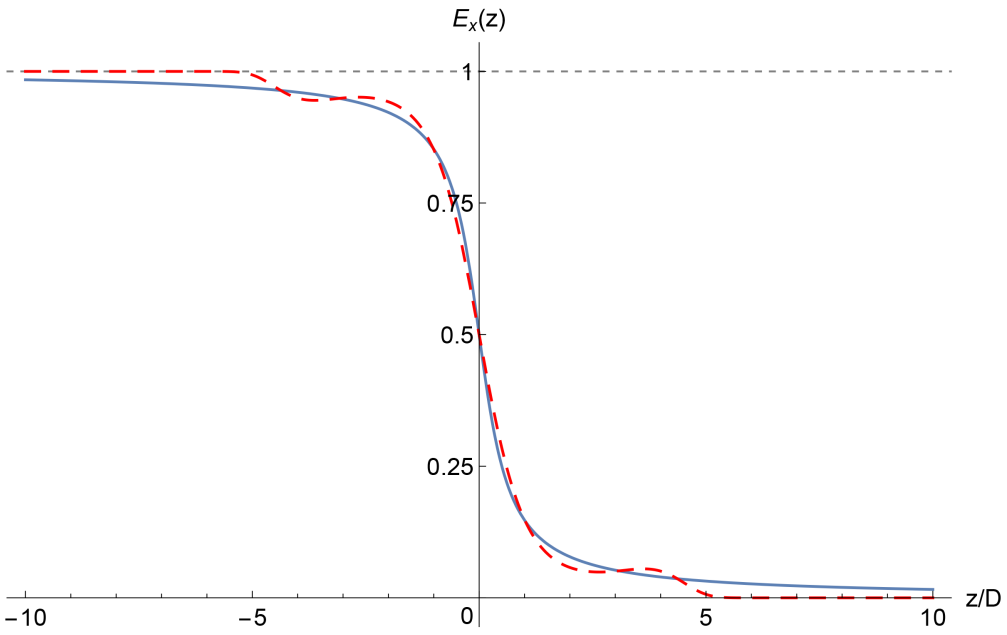


Figure 2.5: Least-squares Enge function $F_5(z)$ (dashed red), fitted to the electrostatic field falloff $E_x(z)$ (solid blue) of the semi-infinite capacitor with uniform charge distribution.

falloffs in general.

In this uniform charge distribution case, the field falloff $E_x(z)$ has a quite simple symbolic form. This is the next best thing in the absence of an accurate Enge function representation.

Moreover, one may attempt to model hypothetical field falloffs of similar visual appearance that satisfy $E_x(z)+E_x(-z)=1$, but do not have a known symbolic form, by a function such as

$$F_n(z)=\begin{cases}\frac{1}{\pi}\cot^{-1}\left(\sum_{j=0}^{n-1}a_j\left(\frac{z}{D}\right)^{2j+1}\right)+1 & ,z<0,\\ \frac{1}{\pi}\cot^{-1}\left(\sum_{j=0}^{n-1}a_j\left(\frac{z}{D}\right)^{2j+1}\right) & ,z>0,\\ 1/2 & ,z=0.\end{cases}$$

2.2.3 Asymptotic Field Falloff Behavior

A Taylor series expansion of the electrostatic field along the z axis $E_x(z)$ at point $z=+\infty$, performed in *Mathematica*, yields

$$E_x(z)=\frac{1}{2\pi(z/D)}-\frac{1}{24\pi(z/D)^3}+\frac{1}{160\pi(z/D)^5}+O\left[\left(\frac{1}{(z/D)}\right)^7\right].$$

However,

1. the electrostatic field of an electrostatic dipole, comprising two point-like charges at $x = \pm d$, $y = 0$, and $z = 0$, falls off at $(x, y, z) = (0, 0, +\infty)$ as $\sim 1/z^3$; and

2. the electrostatic field of a capacitor, comprising two semi-infinite rods at $x = \pm d$, $y = 0$, and $z \leq 0$, falls off at $(x, y, z) = (0, 0, +\infty)$ as $\sim 1/z^2$.

To see that there is no contradiction, it is sufficient to note that the point-like charges electrostatic dipole consists of 0D elements, the semi-infinite rods capacitor consists of 1D elements, and, in our case, the semi-infinite plates capacitor comprises 2D elements.

Consider an observation point $(x, 0, z)$, where $z > 0$. In case of semi-infinite rods with linear charge densities $\lambda = \pm 1$, the electrostatic potential at point $(x, 0, z)$ is

$$\varphi(x, 0, z) = \int_{-\infty}^{0} dt \sum_{\lambda = \pm 1} \frac{\lambda}{\sqrt{(x - \lambda d)^2 + (z - t)^2}},$$

whereas in case of semi-infinite plates the electrostatic potential at point $(x, 0, z)$ is

$$\varphi(x, 0, z) = \int_{-\infty}^{+\infty} dt_1 \int_{-\infty}^{0} dt_2 \sum_{\sigma = \pm 1} \frac{\sigma}{\sqrt{(x - \sigma d)^2 + t_1^2 + (z - t_2)^2}}.$$

The latter is equivalent to the one-dimensional integral

$$\varphi(x, 0, z) = \int_{-\infty}^{0} dt \left(\sum_{\sigma = \pm 1} \frac{\sigma}{\sqrt{(x - \sigma d)^2 + (z - t)^2}} \right) f(z, t),$$

where

$$f(z, t) = 2(z + t) \arccos\left(\frac{z}{t + z} \right)$$

is the length of the arcs of radius $z + t$, sweeping the upper and lower plates from center points $(d, 0, z)$ and $(-d, 0, z)$, respectively.

This further illuminates why the asymptotic order of the electrostatic potential at $(x, y, z) = (0, 0, +\infty)$ is one order higher for the semi-infinite plates capacitor than for the semi-infinite rods capacitor.

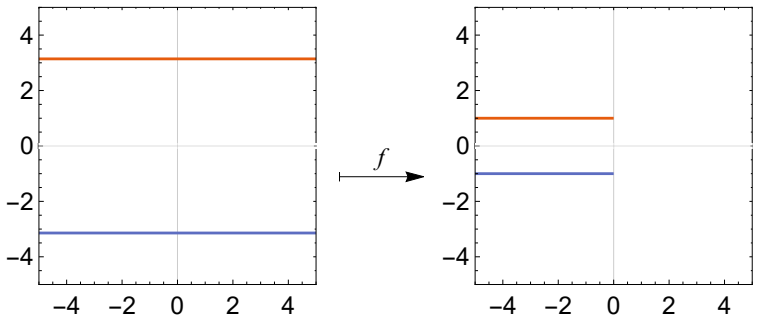


Figure 2.6: The Schwarz–Christoffel map f from the bi-infinite strip $-\pi \leq \Im(v) \leq -\pi$ to the physical domain with semi-infinite infinitely thin plates at $x = \pm d$, $y \leq 0$.

2.3 Semi-Infinite Capacitor with Plates of Uniform Potential

In the realistic case of plates with uniform electrostatic potentials, we expect the surface charge density in the semi-infinite capacitor's plate to be higher near the edge. Indeed, if there were a uniform surface charge density at initial time $z = 0$, integration of the Coulomb electrostatic potentials would have lower total magnitude near the plate edge, which would result in charges rushing to the edge and in relaxation to an equilibrium with higher charge density near the edge.

The relatively higher surface change density near the edges increases the magnitude of the electrostatic field $E_x(z)$ near $z = 0$ compared to the uniform surface charge case, which also means that $E_x(-z)$ increases in magnitude near $z = 0$. As a result, now we have $E_x(z) + E_x(-z) \geq 1 + a$ for some $a > 0$ in a neighborhood $U(0)$ of $z = 0$.

Due to the translational symmetry in the y direction, we can reduce the 3D problem to 2D. Conformal mapping methods, detailed in App. A, are often effective for computing the electrostatic potential for 2D representations of semi-infinite capacitors.

2.3.1 Infinitely Thin Plates

We continue to study a semi-infinite capacitor with infinitely thin plates at $x = \pm d$, $y \leq 0$, with full aperture $D = 2d$, but now the plates are of uniform electrostatic potential instead of uniform charge distribution.

We utilize the Schwarz–Christoffel transformation $f : \mathbb{C} \to \mathbb{C}$ [54, pp. 103–111]

$$z + ix = f(v) = \frac{d}{\pi}(1 + v + \exp(v)),$$

which transforms the real axis onto itself, the line $v = \pi i$ to the plate at $x = d$, $z \leq 0$, and the line $v = -\pi i$ to the plate at $x = -d$, $z \leq 0$. The transformation f is shown in Fig. 2.6.

From the definition of f, we have

$$v + \exp(v) = -1 + \frac{\pi}{d}(z + ix)$$

and

$$\exp(v)\exp(\exp(v)) = \exp\left(-1 + \frac{\pi}{d}(z + ix)\right).$$

Applying the Lambert W function defined as the solution of $W(t)\exp W(t) = t$, we obtain

$$\exp(v) = W\left[\exp\left(-1 + \frac{\pi}{d}(z + ix)\right)\right]$$

and, finally,

$$\begin{aligned} v = f^{-1}(z + ix) = \\ = -1 + \frac{\pi}{d}(z + ix) - W\left[\exp\left(-1 + \frac{\pi}{d}(z + ix)\right)\right]. \end{aligned}$$

If the voltages of the plates at $x = d$ and $x = -d$ are $V = V_0$ and $V = -V_0$, respectively, then the electrostatic potential $\varphi(v)$ at point v is $\varphi(v) = V_0 \Im(v)/\pi$. Without loss of generality, we set $V_0 = 1$.

Thus, we obtain the electrostatic potential as

$$\begin{aligned} \varphi(x,z) = \frac{1}{\pi}\Im\left(f^{-1}(z + ix)\right) = \\ = \frac{x}{d} - \frac{1}{\pi}\Im\left[W\left(\exp\left(-1 + \frac{\pi}{d}(z + ix)\right)\right)\right]. \end{aligned} \tag{2.5}$$

The branch $\kappa(z)$ of the Lambert W function that produces the required range $[-1, 1]$ of the electrostatic potential $\varphi(x,z)$ is

$$\kappa(z) = \left\lceil \frac{1}{2\pi}\Im\left(\frac{\pi}{d}z\right) - \frac{1}{2} \right\rceil.$$

The x component $E_x(x,z)$ of the electrostatic field is then

$$E_x(x,z) = -\frac{\partial}{\partial x}\varphi(x,z) = \\ = \frac{1}{d} - \frac{1}{d}\Re\left[\frac{W\left(\exp\left(-1+\pi\left(z+ix\right)/d\right)\right)}{1+W\left(\exp\left(-1+\pi\left(z+ix\right)/d\right)\right)}\right].$$

We normalize the electrostatic field to $E_x(0,-\infty)=1$. Then, along the z axis, the electrostatic field is

$$E_x(z) = 1 - \Re\left(\frac{W\left(\exp\left(-1+\pi z/d\right)\right)}{1+W\left(\exp\left(-1+\pi z/d\right)\right)}\right).$$

Unlike in the case of plates with uniform charge distribution, the electrostatic field falloff $E_x(z)$ is asymmetric relative to point $(z,x)=(0,1/2)$. Accordingly, we offset the electrostatic field by the non-zero effective field boundary (EFB) $z_{\mathrm{EFB}}/D = 0.920945$. The EFB is computed as

$$z_{\mathrm{EFB}} = z_{\mathrm{int}} + \frac{1}{E_x(z_{\mathrm{int}})}\int_{z_{\mathrm{int}}}^{z_{\mathrm{ext}}} E_x(z)\,dz, \tag{2.6}$$

where z_{int} and z_{ext} are points well inside and well outside the particle optical element, respectively. As a result, the area of the region defined by $E_x(z) \le E \le 1$, $z \le z_{\mathrm{EFB}}$ is equal to the area of the region defined by $0 \le E \le E_x(z)$, $z \ge z_{\mathrm{EFB}}$, with accuracy depending on z_{int} and z_{ext}. We use $z_{\mathrm{ext}}/D = -5$ and $z_{\mathrm{ext}}/D = 20$, and we denote the EFB computed this way as EFB_{20}.

Fig. 2.7 shows a plot of $E_x(z)$ and its 5th order Enge function $F_5(z)$ fitted using the Levenberg–Marquardt Gauss–Newton method and the least-squares variance function. We recognize that the Enge function is an accurate representation only up to about $z/D = 4.5$ due to the electrostatic field falloff being slower than exponential, and we will address this issue in sec. 2.6.

2.3.1.1 Charge Distribution in the Plates

Consider the 2D representation of the upper infinitely thin plate as

$$\mathscr{A} = \{(d,t)\,|t \in (-\infty,0]\}$$

at voltage $V = 1$.

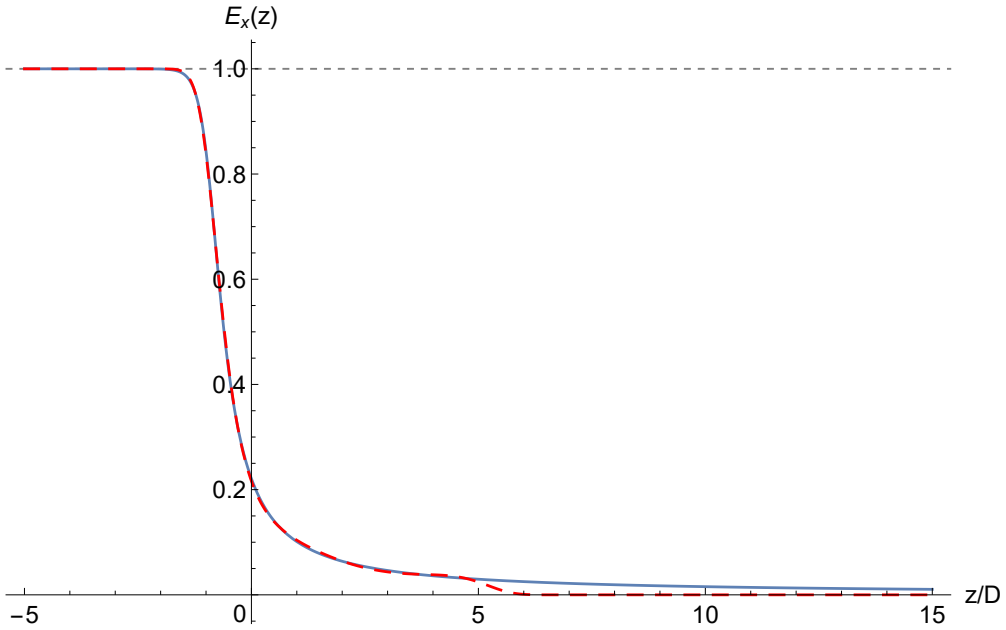


Figure 2.7: Up to about $z/D = 4.5$, the least-squares Enge function $F_5(z)$ (dashed red) is a good approximation of the electrostatic field falloff $E_x(z)$ (solid blue) of a semi-infinite capacitor with infinitely thin plates. The curves are offset horizontally by the EFB_{20} $z_{\mathrm{EFB}}/D = 0.920945$ so that the EFB_{20} is at $z = 0$.

The surface charge density $\sigma(z)$ of the plate is proportional to the jump in the normal component E_x of the electrostatic field; specifically,

$$\sigma(z) = \begin{cases} \varepsilon_0 \left(E_x(d+0,z) - E_x(d-0,z)\right) & \text{in SI units,} \\ \frac{1}{4\pi} \left(E_x(d+0,z) - E_x(d-0,z)\right) & \text{in Gaussian units,} \end{cases}$$

where ε_0 is the vacuum permittivity.

Alternatively to obtaining $\sigma(z)$ from the electrostatic field, we can obtain $\sigma(z)$ by solving the Fredholm equation of the first type

$$f(z) = \int_{-\infty}^{+\infty} \sigma(z) K(z,t)\, dt, \tag{2.7}$$

where

1. $\sigma(z)$ is the surface charge distribution in the plate, with $\sigma(z) = 0$ for $z > 0$ (beyond the plate edge);

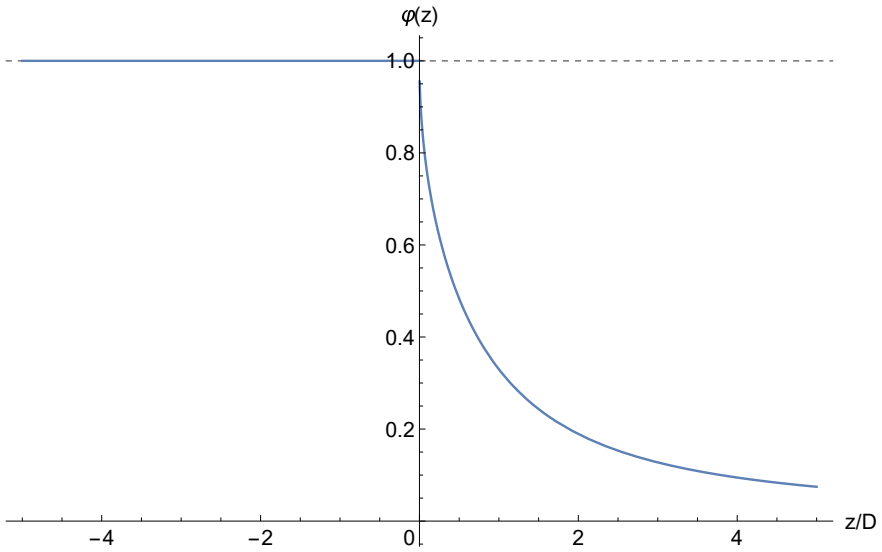


Figure 2.8: Electrostatic potential $f(z) = \varphi(d,z)$ in plane of the upper plate.

2. $f(z)$ is the electrostatic potential $f(z) = \varphi(d,z)$ in the plane of the plate, with $f(z) = 1$ for $z \leq 0$, shown in Fig. 2.8;

3. $K(z,t)$ is the difference kernel $K(z,t) = K(z-t)$ obtained from the Coulomb potential as

$$K(z,t) = \int_{-\infty}^{+\infty} \left(\frac{1}{\sqrt{(t-z)^2 + y^2}} - \frac{1}{\sqrt{(2d)^2 + (t-z)^2 + y^2}} \right) dy,$$

from which we obtain, similarly to eq. 2.2,

$$K(z,t) = \ln \left[\frac{4d^2}{(t-z)^2} + 1 \right].$$

From eq. 2.5, we have

$$f(z) = \varphi(d,z) = 1 - \frac{1}{\pi} \Im \left(W \left(-e^{\pi z/d - 1} \right) \right).$$

To check that eq. 2.7 indeed produces the solution for $\sigma(z)$, it is sufficient to insert $K(z,t)$ into the integral, which results in the Coulomb potential integral

$$f(z) = \int_{-\infty}^{0} \int_{-\infty}^{+\infty} \sigma(z) \left(\frac{1}{\sqrt{(t-z)^2 + y^2}} - \frac{1}{\sqrt{(2d)^2 + (t-z)^2 + y^2}} \right) dy\, dt$$

with integration over the two plates.

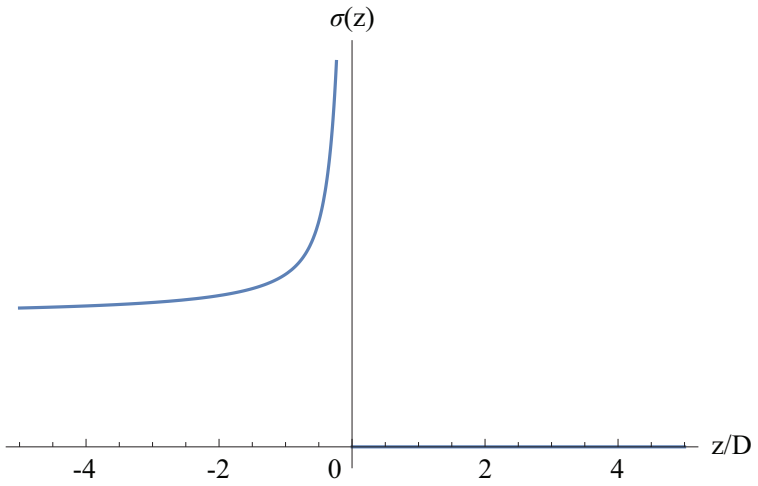


Figure 2.9: Surface charge distribution $\sigma(z)$ in the upper plate, located at $x = d$, $z \leq 0$.

To solve this Fredholm equation of the first kind [86, sec. 10.3-1], we apply the Fourier transform to both sides of the equation, obtaining

$$\mathscr{F}_z(\sigma(z)) = \frac{\mathscr{F}_z(f(z))}{\mathscr{F}_z(K(z))}.$$

Next, we take the inverse Fourier transform of the expression and obtain

$$\begin{aligned}\sigma(z) &= \mathscr{F}_\omega^{-1}(\mathscr{F}_z(\sigma(z)))(z) = \\ &= \mathscr{F}_\omega^{-1}\left(\frac{\mathscr{F}_z(f(z))}{\mathscr{F}_z(K(z))}\right)(z).\end{aligned}$$

The resulting form of surface charge distribution $\sigma(z)$ is shown in Fig. 2.9.

2.3.2 Solid Metal Plates

We proceed to study electrostatic deflectors with solid metal plates. Using the *Schwarz–Christoffel Toolbox* (*SC Toolbox*) for *MATLAB* [28] (see App. A for details), we obtain derivatives of conformal maps for calculation of the electrostatic potential. We consider the cases of one capacitor and two adjacent capacitors.

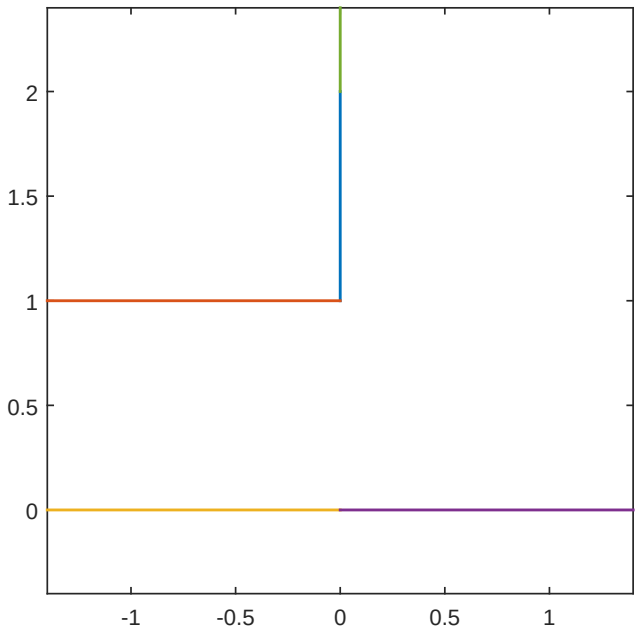


Figure 2.10: The polygonal model of the semi-infinite capacitor with infinitely thick plates.

2.3.2.1 One Capacitor

2.3.2.1.1 Infinitely Thick Plates

Consider a solid metal semi-infinite capacitor with full aperture $D = 2$ and infinitely thick plates

$$A = \{(t_0, t_1, t_2) \,|t_0 \in [d, +\infty) \wedge t_1 \in (-\infty, +\infty) \wedge t_2 \in (-\infty, 0]\}$$

and

$$B = \{(t_0, t_1, t_2) \,|t_0 \in (-\infty, -d] \wedge t_1 \in (-\infty, +\infty) \wedge t_2 \in (-\infty, 0]\},$$

at voltages $V = 1$ and $V = -1$, respectively.

The geometry and the electrostatic potential are, up to the change of sign, reflection-symmetric relative the $x = 0$ plane. By this symmetry, the electrostatic potential at $x = 0$ is zero. This enables us to reduce the problem to its $x \geq 0$ part.

We represent the $x \geq 0$ half of the electrostatic capacitor by a complex region bounded by (1) the real axis at zero potential and (2) the 2D representation of the upper plate

$$\mathscr{A} = \{t + id | t \in (-\infty, 0]\} \cup \{it | t \in [d, +\infty)\}$$

at voltage $V = 1$, where $d = 1$. This polygonal model is shown in Fig. 2.10.

We encode this polygonal physical domain in the *SC Toolbox* by its vertices $w = z + ix = (i, \infty, 0, \infty, 2i)$ and interior angles $\alpha\pi = (3/2, 0, 1, -1/2, 1)\,\pi$ as

```
p1 = polygon([i, Inf, 0, Inf, 2*i], [3/2, 0, 1, -1/2, 1])
```

where vertices 0 and $2i$ were added as guiding points for the *SC Toolbox*'s algorithm.

We obtain a conformal mapping f from the bi-infinite strip $0 \leq \Im(v) \leq 1$ using the command

```
f1 = stripmap(p1, [2, 4], options)
```

with the second and fourth prevertices $+\infty$ and $-\infty$ mapping to the infinite vertices of the polygon.

The resulting output lists prevertices $v_j = f^{-1}(w_j)$ of the conformal mapping, corresponding to vertices w_j and interior angles $\pi\alpha_j$ of the polygonal model.

```
  stripmap object:

     vertex                 alpha          prevertex
 ------------------------------------------------------------

     Inf + 0.00000 i      -0.50000                  -Inf
 0.00000 + 2.00000 i       1.00000   0.000000000000e+00
 0.00000 + 1.00000 i       1.50000   6.933630404535e-01
     Inf + 0.00000 i       0.00000                   Inf
 0.00000 + 0.00000 i       1.00000   9.552476375563e-01 + i

 c = -1.7238602 - 1.7238602 i

 Apparent accuracy is 2.48e-16
```

The formula for a conformal mapping from the bi-infinite strip $0 \leq \Im(v) \leq 1$ is [30, p. 46]

$$f(v) = f(v_0) + c \int_{v_0}^{v} \exp\left[\frac{\pi}{2}(\alpha_- - \alpha_+)\,\zeta\right] \prod_{j=1}^{n} \left[\sinh\frac{\pi}{2}(\zeta - v_j)\right]^{\alpha_j - 1} d\zeta,$$

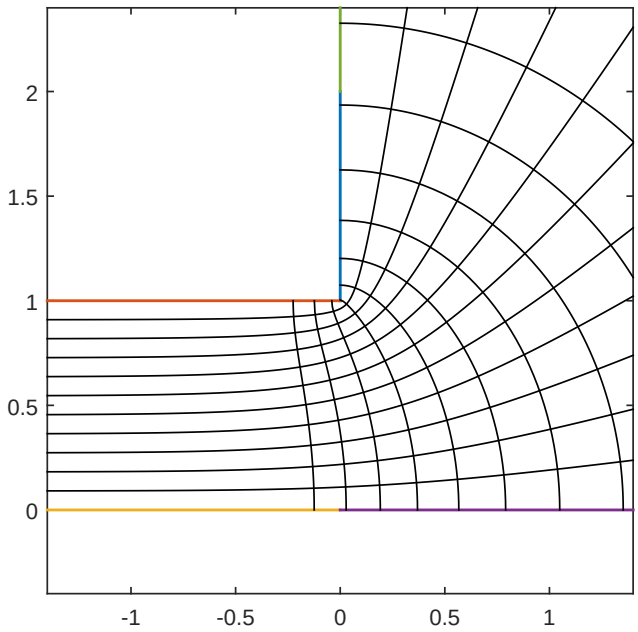


Figure 2.11: The electrostatic field and equipotential lines of the semi-infinite capacitor with infinitely thick plates.

where $n = 3$ is the number of finite prevertices of the polygon in the canonical domain,

$$v = (0, 0.6933630404535, 0.9552476375563 + i)$$

are the finite prevertices; $\alpha\pi = (1, 1.5, 1)\,\pi$ are the respective interior angles of the polygon; and $\alpha_-\pi = -0.5\pi$, $\alpha_+\pi = 0$ are the divergence angles at the ends of the bi-infinite strip. A prevertex v_j can be used for the reference point v_0; then $f(v_0)$ is the respective vertex w_j. The scaling constant $c = -2.4379064258967187$ in this formula is different from scaling constant c in the *SC Toolbox*'s output and can be obtained by matching prevertice images $f(v_j)$ to respective vertices w_j.

Fig. 2.11 shows the resulting electrostatic field and equipotential lines of the semi-infinite capacitor with infinitely thick plates. The x component $E_x(z)$ of the electrostatic field along the z axis is shown in Fig. 2.12. The falloff $E_x(z)$ is very slow, and the EFB as a function of an increasing z_{ext} does not appear to converge. The EFB_{20} is $z_{\text{EFB}}/D = 1.41566$.

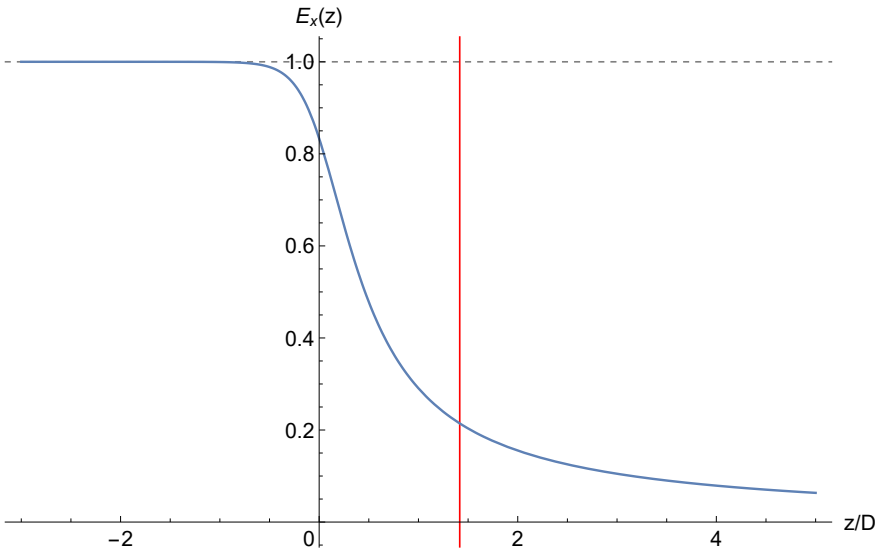


Figure 2.12: Electrostatic field falloff $E_x(z)$ of the semi-infinite capacitor with infinitely thick plates. The red vertical gridline denotes the EFB_{20} $z_{\mathrm{EFB}}/D = 1.41566$.

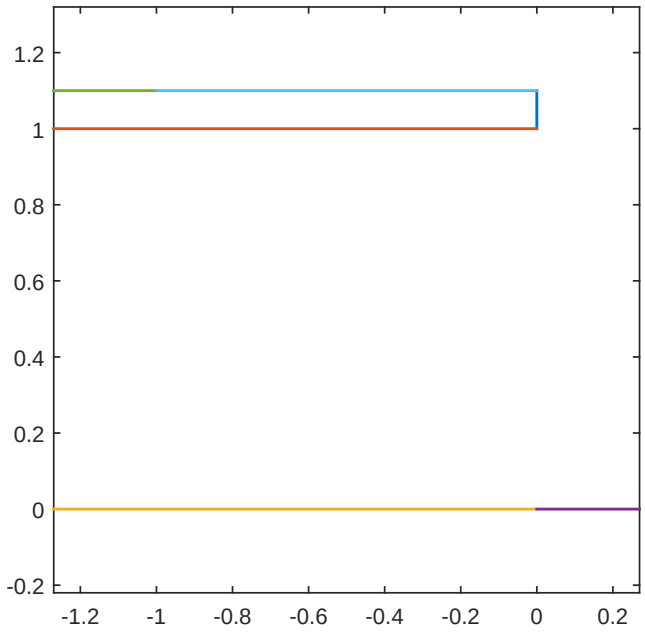


Figure 2.13: The polygonal model of the semi-infinite capacitor with finitely thick plates.

2.3.2.1.2 Plates of Finite Thickness

Consider a semi-infinite capacitor with full aperture $D = 2$ and plates of $D/20$ thickness $2r = 0.1$. We model the $x \geq 0$ half of this capacitor by a complex region bounded by the real axis at zero potential and the plate

$$\mathscr{A} = \{t + id | t \in (-\infty, 0]\} \cup \{it | t \in [d, d + 2r]\} \cup \{t + (d + 2r)\, i | t \in (-\infty, 0]\}$$

at voltage $V = 1$, where $d = 1$. This polygonal model is shown in Fig. 2.13.

We encode this polygonal physical domain in the *SC Toolbox* by its vertices w_j and interior angles $\alpha_j \pi$ as

```
p2 = polygon([1.1i, i, Inf, 0, Inf, -1+1.1i], [3/2, 3/2, 0, 1, -1, 1])
```

where vertices 0 and $-1 + 1.1i$ were added as guiding points for the *SC Toolbox*'s algorithm.

Similarly to the previous case, we obtain the following conformal mapping from a bi-infinite strip to the physical domain using the *SC Toolbox*:

$$f(v) = f(v_0) + c \int_{v_0}^{v} \exp\left[\frac{\pi}{2}(\alpha_- - \alpha_+)\,\zeta\right] \prod_{j=1}^{n} \left[\sinh \frac{\pi}{2}(\zeta - v_j)\right]^{\alpha_j - 1} d\zeta,$$

where $n = 4$ is the number of finite prevertices of the polygon in the canonical domain,

$$v = (0, 0.4475104912323, 0.7298948123684, 1.051921215239 + i)$$

are the finite prevertices; $\alpha\pi = (1, 1.5, 1.5, 1)\,\pi$ are the respective interior angles of the polygon; and $\alpha_-\pi = -\pi$, $\alpha_+\pi = 0$ are the divergence angles at the ends of the bi-infinite strip. The scaling constant is $c = -5.042384793987374$.

Fig. 2.14 shows the resulting electrostatic field and equipotential lines of the semi-infinite capacitor with plates of $D/20$ thickness. The falloff of the x component $E_x(z)$ of the electrostatic field along the z axis has the EFB_{20} $z_{\mathrm{EFB}}/D = 0.952043$ and is shown in Fig. 2.15.

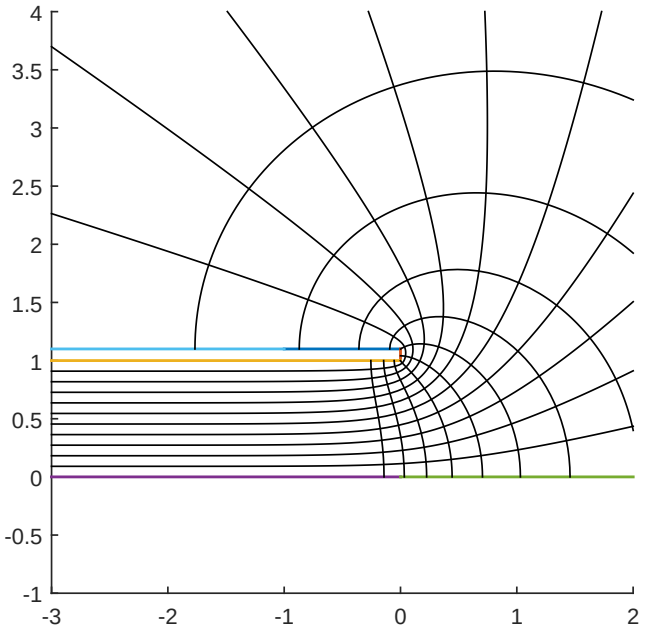


Figure 2.14: The electrostatic field and equipotential lines of the semi-infinite capacitor with plates of $D/20$ thickness.

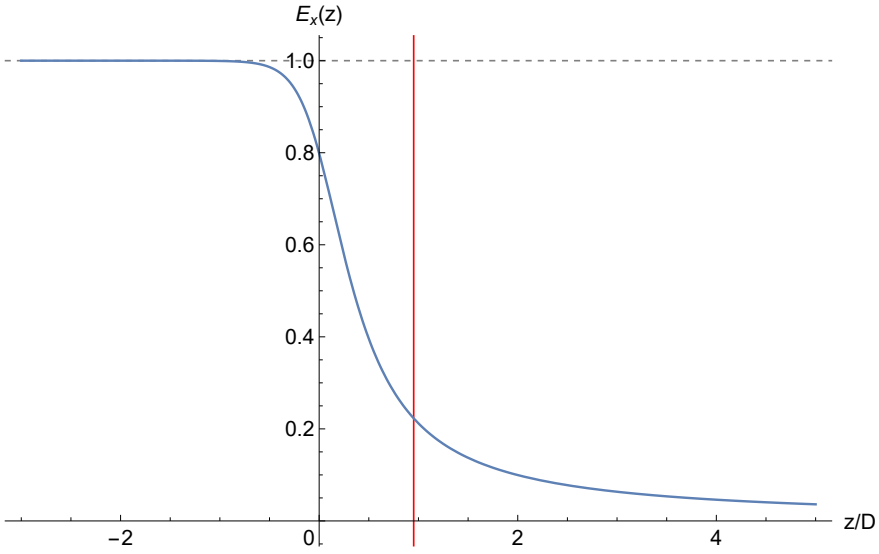


Figure 2.15: Electrostatic field falloff $E_x(z)$ of the semi-infinite capacitor with plates of $D/20$ thickness. The red vertical gridline denotes the EFB_{20} $z_{\text{EFB}}/D = 0.952043$.

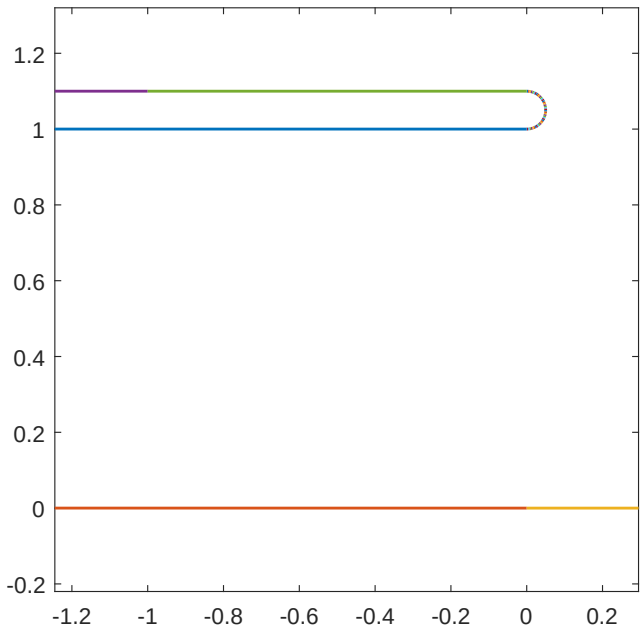


Figure 2.16: The polygonal model of the semi-infinite capacitor with plates of $D/20$ thickness and rounded edges.

2.3.2.1.3 Plates of Finite Thickness with Rounded Edges

Consider a semi-infinite capacitor with full aperture $D = 2$ and plates of $D/20$ thickness $2r = 0.1$ with rounded edges. We model the $x \geq 0$ half of this capacitor by a complex region bounded by the real axis at zero electrostatic potential and the plate

$$\mathscr{A} = \{t + di | t \in (-\infty, 0]\} \cup \left\{(d+r)\, i + r \exp(it) \,|t \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]\right\} \cup \{t + (d+2r)\, i | t \in (-\infty, 0]\}$$

at voltage $V = 1$, where $d = 1$. We approximate the arc by a piecewise linear curve with 42 line segments of equal lengths. This polygonal model is shown in Fig. 2.16.

We encode this polygonal physical domain in the *SC Toolbox* as listed in App. B.3.

Proceeding as in the previous cases, we obtain the following conformal mapping from a bi-infinite strip to the physical domain using the *SC Toolbox*:

$$f(v) = f(v_0) + c \int_{v_0}^{v} \exp\left[\frac{\pi}{2}(\alpha_- - \alpha_+)\,\zeta\right] \prod_{j=1}^{n} \left[\sinh \frac{\pi}{2}(\zeta - v_j)\right]^{\alpha_j - 1} d\zeta,$$

where $n = 45$ is the number of finite prevertices of the polygon in the canonical domain,

$$v = (0, 0.3805344321060, 0.3866459670307, 0.3933092616665,$$

$$0.4004238033105, 0.4079574599867, 0.4158853193485, 0.4241790327221,$$
$$0.4328391532257, 0.4418431833483, 0.4511799906304, 0.4608417087954,$$
$$0.4708063751390, 0.4810908733017, 0.4916343816466, 0.5024841810347,$$
$$0.5136015853200, 0.5249435318814, 0.5365474909385, 0.5483716273552,$$
$$0.5604324317077, 0.5726549790831, 0.5850879474069, 0.5976858775498,$$
$$0.6104007470804, 0.6232810803701, 0.6362432805046, 0.6492989469765,$$
$$0.6623928970331, 0.6755595399189, 0.6887381824925, 0.7018677939905,$$
$$0.7149929505145, 0.7280204245606, 0.7409539842791, 0.7537450473342,$$
$$0.7663616643523, 0.7787656435760, 0.7909001604658, 0.8027400857953,$$
$$0.8142156994585, 0.8252593198779, 0.8357891776466, 0.8456098356243,$$
$$1.088069920623 + i)$$

are the finite prevertices;

$$\alpha\pi = \left(1, \frac{85}{84}, \underbrace{\frac{43}{42}, \ldots, \frac{43}{42}}_{41 \text{ times}}, \frac{85}{84}, 1\right)\pi$$

are the respective interior angles of the polygon; and $\alpha_-\pi = -\pi$, $\alpha_+\pi = 0$ are the divergence angles at the ends of the bi-infinite strip. The scaling constant is $c = -5.096904973741541$.

Fig. 2.17 shows the resulting electrostatic field and equipotential lines of the semi-infinite capacitor with plates of $D/20$ thickness and rounded edges. The falloff of the x component $E_x(z)$ of the electrostatic field along the z axis has the EFB_{20} $z_{\text{EFB}}/D = 0.962347$ and is shown in Fig. 2.18.

Additionally, we performed calculations for the same case of a semi-infinite capacitor with finitely thick plates and rounded edges, but with the plate thickness increased to $D/4$. The resulting falloff of the electrostatic field $E_x(z)$ has the EFB_{20} $z_{\text{EFB}}/D = 1.10338$ and is shown in Fig. 2.19.

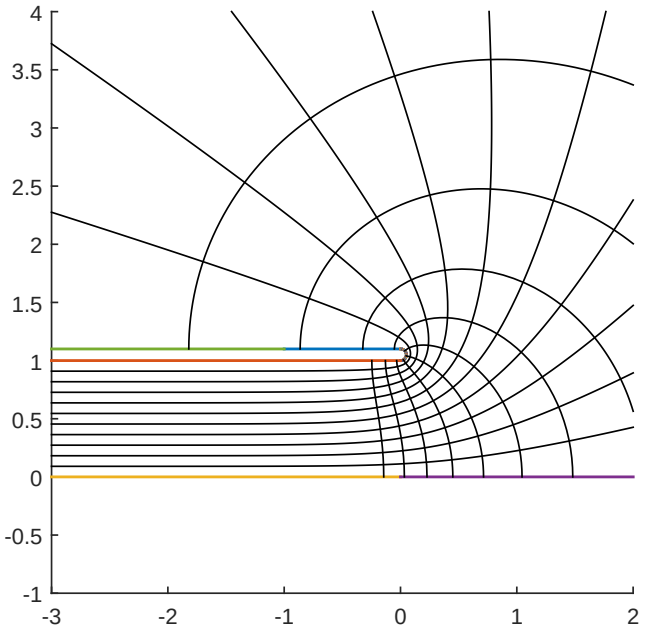


Figure 2.17: The electrostatic field and equipotential lines of the semi-infinite capacitor with plates of $D/20$ thickness and rounded edges.

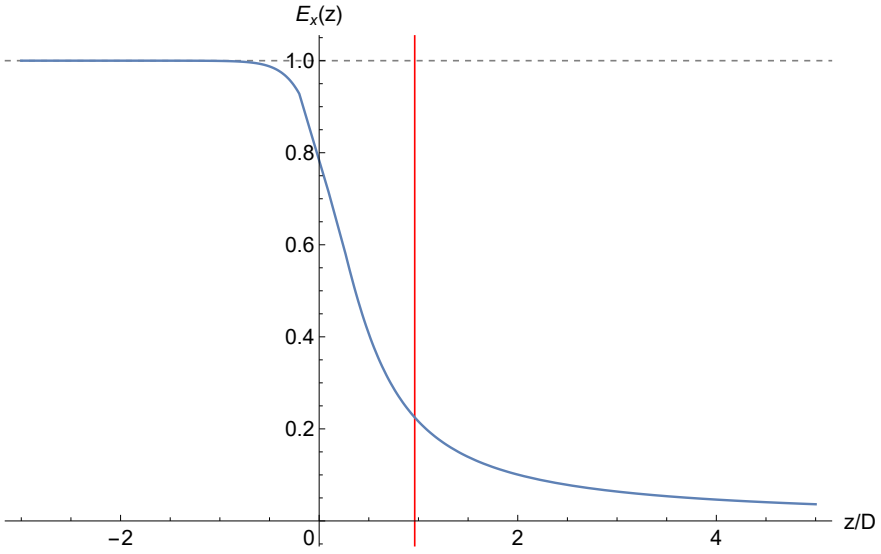


Figure 2.18: Electrostatic field falloff $E_x(z)$ of the semi-infinite capacitor with plates of $D/20$ thickness and rounded edges. The red vertical gridline denotes the EFB_{20} $z_{\mathrm{EFB}}/D = 0.962347$.

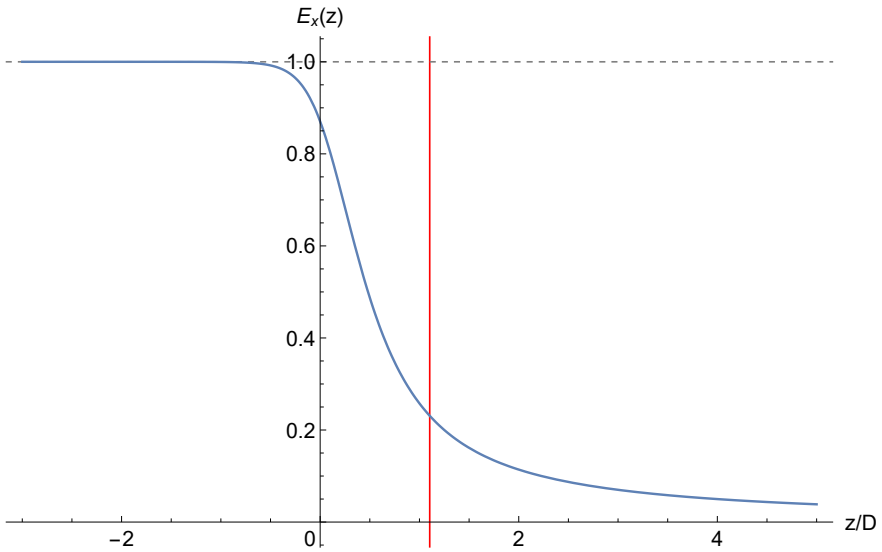


Figure 2.19: Electrostatic field falloff $E_x(z)$ of the semi-infinite capacitor with plates of $D/4$ thickness and rounded edges. The red vertical gridline denotes the EFB_{20} $z_{\mathrm{EFB}}/D = 1.10338$.

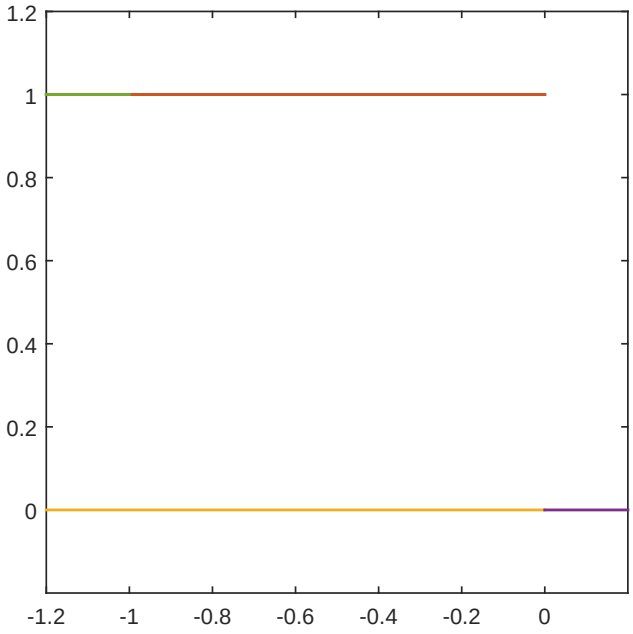


Figure 2.20: The polygonal model of the semi-infinite capacitor with infinitely thin plates.

2.3.2.1.4 Infinitely Thin Plates

For accuracy comparison, we return to the case of a semi-infinite capacitor with full aperture $D = 2$ and infinitely thin plates. We model the $x \geq 0$ half of this capacitor by a complex region bounded by the real axis at zero electrostatic potential and the plate

$$\mathscr{A} = \{t + di | t \in (-\infty, 0]\}$$

at voltage $V = 1$, where $d = 1$. This capacitor model is shown in Fig. 2.20.

We encode this polygonal physical domain in the *SC Toolbox* by its vertices w_j and interior angles $\alpha_j \pi$ as

```
p0 = polygon([i, Inf, 0, Inf, -1+i], [2, 0, 1, -1, 1])
```

where vertices 0 and $-1+i$ were added as guiding points for the *SC Toolbox*'s algorithm.

We obtain the following conformal mapping from a bi-infinite strip to the physical domain using the *SC Toolbox*:

$$f(v) = f(v_0) + c \int_{v_0}^{v} \exp\left[\frac{\pi}{2}(\alpha_- - \alpha_+)\zeta\right] \prod_{j=1}^{n} \left[\sinh \frac{\pi}{2}(\zeta - v_j)\right]^{\alpha_j - 1} d\zeta,$$

where $n = 3$ is the number of finite prevertices of the polygon in the canonical domain,

$$v = (0, 0.5660574857390, 0.9730053888353 + i)$$

are the finite prevertices; $\alpha\pi = (1,2,1)\pi$ are the respective interior angles of the polygon; and $\alpha_-\pi = -\pi$, $\alpha_+\pi = 0$ are the divergence angles at the ends of the bi-infinite strip. The scaling constant is $c = -4.866174962862232$.

Fig. 2.21 shows the resulting electrostatic field and equipotential lines of the semi-infinite capacitor with infinitely thin plates. The falloff of the x component $E_x(z)$ of the electrostatic field along the z axis is shown in Fig. 2.22 and agrees well with the previous result from Fig. 2.7. The EFB_{20} is $z_{\text{EFB}}/D = 0.920945$ – same as obtained in sec. 2.3.1.

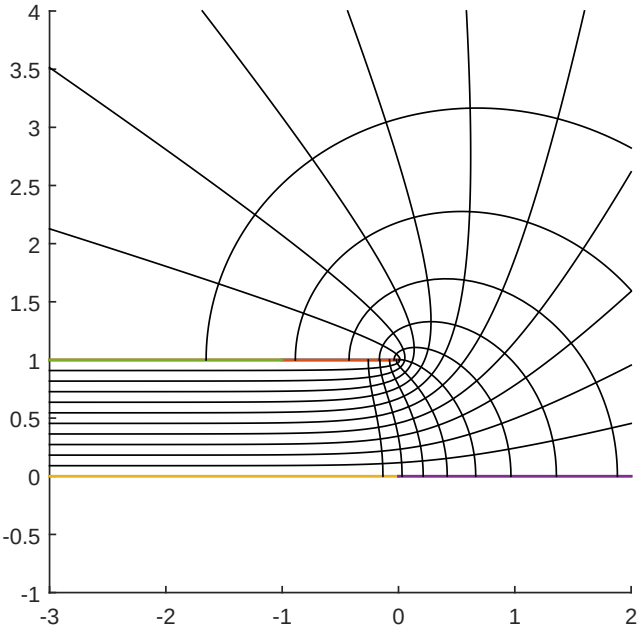


Figure 2.21: The electrostatic field and equipotential lines of the semi-infinite capacitor with infinitely thin plates.

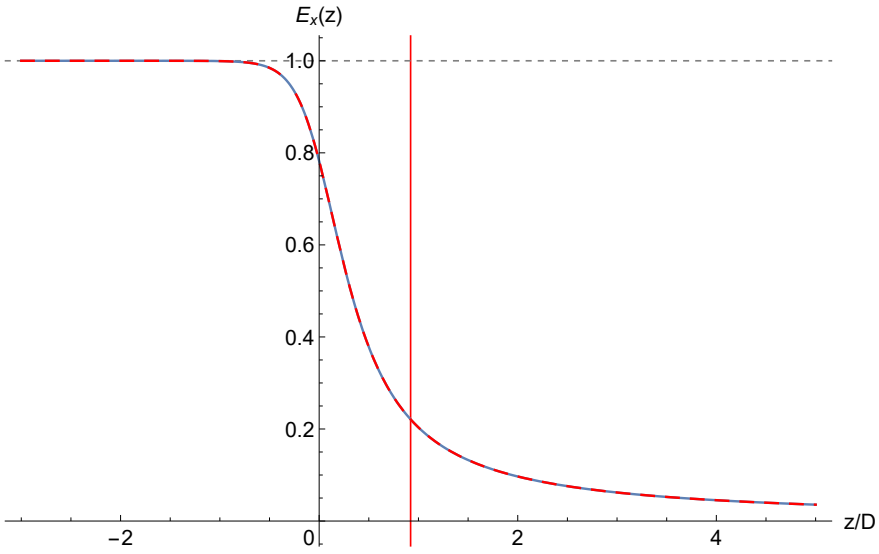


Figure 2.22: Electrostatic field falloff $E_x(z)$ of the semi-infinite capacitor with infinitely thin plates obtained using the *SC Toolbox* (dashed red) is visually coincident with the previous result from Fig. 2.7 (dashed blue). The red vertical gridline denotes the EFB_{20} $z_{\text{EFB}}/D = 0.920945$.

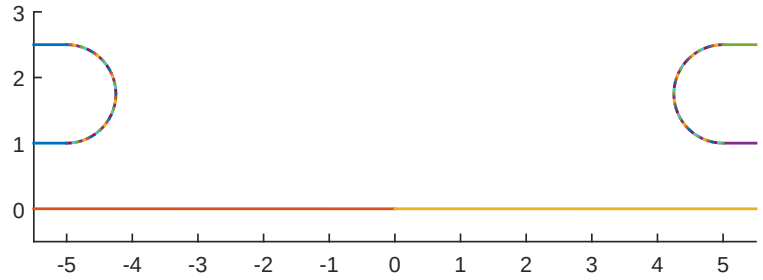


Figure 2.23: The polygonal model of the two adjacent semi-infinite capacitors with plates of $3D/4$ thickness, symmetric voltages, and rounded edges.

2.3.2.2 Two Adjacent Capacitors with Symmetric Voltages

We now consider two adjacent semi-infinite capacitors with full aperture $D = 2$, plates of $3D/4$ thickness $2r = 2.5$, and rounded edges. Let the voltages of these capacitors have the same magnitude and symmetric alignment along the x axis, i.e., the upper ($x > 0$) plates of these capacitors have voltages of the same sign. This model straightforwardly follows from the approximation of an electrostatic deflector as a parallel plates capacitor with (1) the deflection radius as infinity and (2) plate voltages of the same magnitude and the opposite sign. We specify and model the capacitors by a complex region bounded by the real axis at zero electrostatic potential, the first capacitor's upper plate

$$\mathscr{A} = \{-z_{\text{off}} + t + di | t \in (-\infty, 0]\} \cup \left\{-z_{\text{off}} + (d+r)\, i + r \exp(it) \,|t \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]\right\} \cup$$
$$\cup \{-z_{\text{off}} + t + (d+2r)\, i | t \in (-\infty, 0]\}$$

at voltage $V = 1$, and the adjacent capacitor's upper plate

$$\mathscr{B} = \{z_{\text{off}} + t + di | t \in [0, +\infty)\} \cup \left\{z_{\text{off}} + (d+r)\, i - r \exp(it) \,|t \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]\right\} \cup$$
$$\cup \{z_{\text{off}} + t + (d+2r)\, i | t \in [0, +\infty)\}$$

at the same, symmetric voltage $V = 1$, where $d = 1$. Let the distance between the capacitors be $2z_{\text{off}} = 10$, not including the rounded parts of their edges. We approximate each arc by a piecewise linear curve with 42 line segments of equal lengths. This polygonal model is shown in Fig. 2.23.

We encode this polygonal physical domain in the *SC Toolbox* as listed in App. B.5.

We obtain a conformal mapping f from the bi-infinite strip $0 \leq \Im(v) \leq 1$ using the command

```
f3b = stripmap(p, [44, 46], options)
```

with the 44th and 46th prevertices $+\infty$ and $-\infty$ mapping to the infinite vertices of the polygon.

The resulting conformal mapping f is as follows:

$$f(v) = f(v_0) + c \int_{v_0}^{v} \exp\left[\frac{\pi}{2}(\alpha_- - \alpha_+)\,\zeta\right] \prod_{j=1}^{n} \left[\sinh\frac{\pi}{2}(\zeta - v_j)\right]^{\alpha_j - 1} d\zeta,$$

where $n = 90$ is the number of finite prevertices of the polygon in the canonical domain,

$$\begin{aligned} v = (&0, 0.06898503648935, 0.1407073105108, 0.2134982040465,\\ &0.2864841921843, 0.3590056630890, 0.4305569321602, 0.5007505699052,\\ &0.5692474769696, 0.6358038856252, 0.7002087675776, 0.7623352142584,\\ &0.8220702444247, 0.8793651508882, 0.9341719308222, 0.9864907936049,\\ &1.036323118213, 1.083711410132, 1.128699072557, 1.171331486694,\\ &1.211673088789, 1.249799897438, 1.285781148129, 1.319690530978,\\ &1.351613922478, 1.381632958122, 1.409823501502, 1.436268003408,\\ &1.461053624759, 1.484253948760, 1.505950018365, 1.526211988764,\\ &1.545118039791, 1.562734119036, 1.579130994654, 1.594366892925,\\ &1.608502284766, 1.621587383544, 1.633672991990, 1.644798981369,\\ &1.654992351819, 1.664262998209, 1.672540497375, 1.752685224415,\\ &1.973073503798, 2.193484524393, 2.273641964824, 2.281920852719,\\ &2.291193070108, 2.301388186731, 2.312516104216, 2.324603832939,\\ &2.337691257349, 2.351829196176, 2.367067879712, 2.383467798869,\\ &2.401087200915, 2.419996878987, 2.440262806666, 2.461963194376,\\ &2.485168228891, 2.509958989885, 2.536409099130, 2.564605762845,\\ &2.594631480586, 2.626562168172, 2.660479520094, 2.696469479019, \end{aligned}$$

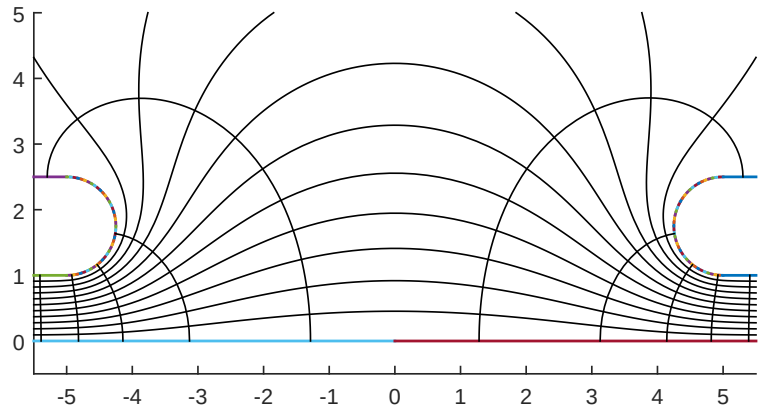


Figure 2.24: The electrostatic field and equipotential lines of the two adjacent semi-infinite capacitors with plates of $3D/4$ thickness, symmetric voltages, and rounded edges.

$$2.734605805023, 2.774957809473, 2.817601596109, 2.862601691144,$$
$$2.910003568099, 2.959850729000, 3.012185784615, 3.067010211839,$$
$$3.124324320648, 3.184080193251, 3.246229204345, 3.310658424767,$$
$$3.377240981341, 3.445765832052, 3.515989160286, 3.587571751257,$$
$$3.660126008227, 3.733145992744, 3.805971730819, 3.877729193575,$$
$$3.946748819345, 1.973050207634 + i)$$

are the finite prevertices;

$$\alpha\pi = \left(\frac{85}{84}, \underbrace{\frac{43}{42}, \ldots, \frac{43}{42}}_{41 \text{ times}}, \frac{85}{84}, 1, -1, 1, \frac{85}{84}, \underbrace{\frac{43}{42}, \ldots, \frac{43}{42}}_{41 \text{ times}}, \frac{85}{84}, 1 \right) \pi$$

are the respective interior angles of the polygon; and $\alpha_{\pm}\pi = 0$ are the divergence angles at the ends of the bi-infinite strip. The scaling constant is $c = 4.9197519279647315 + i\,1.0048276665744966$.

Fig. 2.24 shows the resulting electrostatic field and equipotential lines of the two adjacent semi-infinite capacitors with plates of $5D/4$ thickness and rounded edges.

We denote the x component of the electrostatic field along the z axis of the two adjacent capacitors as $E_{A\&B}(z)$, of the left capacitor $\mathscr{A}$ in free space as $E_A(z)$, and of the right capacitor $\mathscr{B}$ in free space as $E_B(z)$.

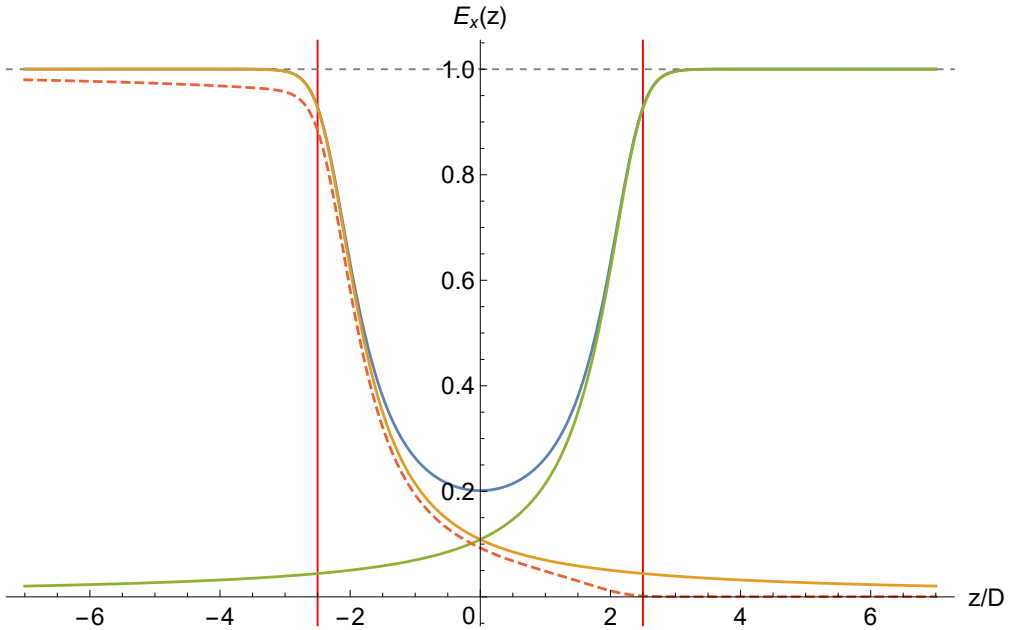
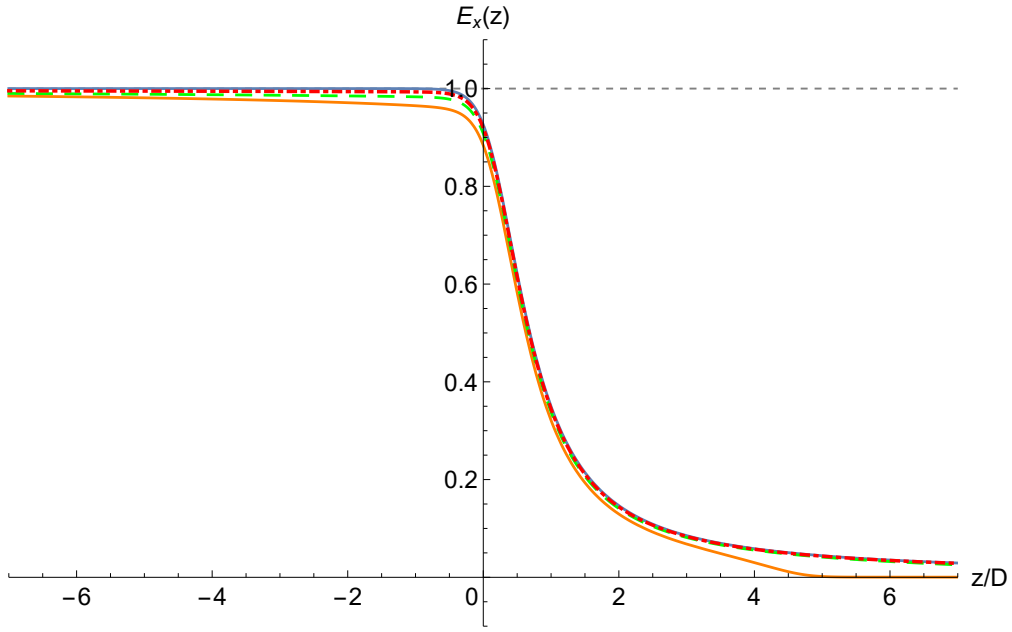


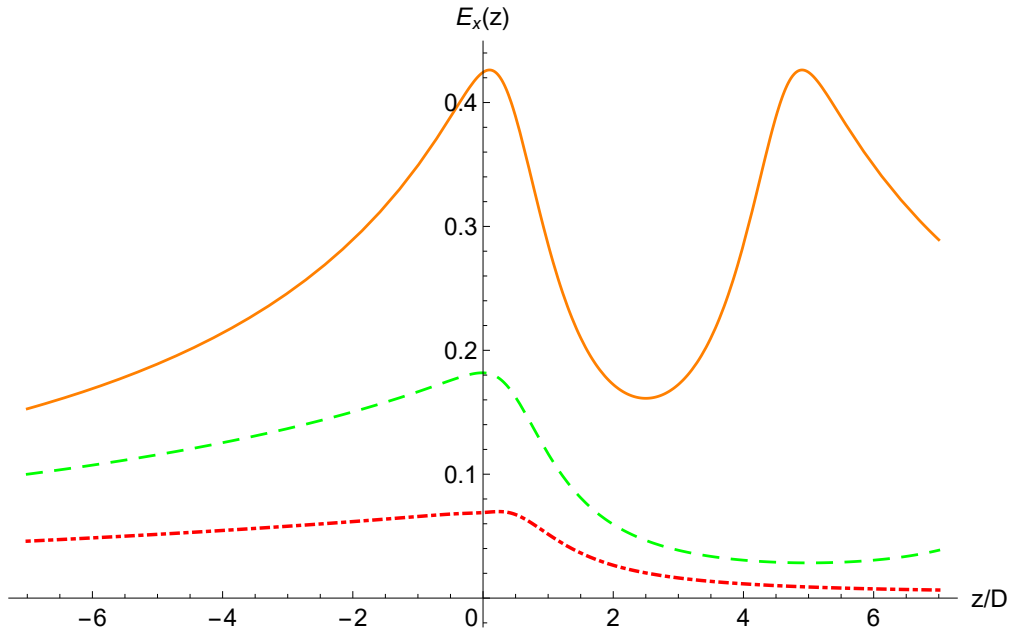
Figure 2.25: $E_{A\&B}(z)$ (solid blue) is the electrostatic field of two adjacent semi-infinite capacitors with plates of $3D/4$ thickness, symmetric voltages, and rounded edges. $E_A(z)$ (solid orange) and $E_B(z)$ (solid green) would be the individual electrostatic fields of the left and the right capacitor in free space, respectively. The difference $E_{A\&B}(z) - E_B(z)$ (dashed red) demonstrates that $E_{A\&B}(z)$ is not a superposition of $E_A(z)$ and $E_B(z)$ due to electrostatic induction.

Fig. 2.25 shows a plot of electrostatic fields $E_{A\&B}(z)$, $E_A(z)$, $E_B(z)$, and the difference $E_{A\&B}(z) - E_B(z)$. If the electrostatic field $E_{A\&B}(z)$ of the two adjacent capacitors was a superposition of their individual free-space electrostatic fields $E_A(z)$ and $E_B(z)$, $E_{A\&B}(z) - E_B(z)$ would be equal to $E_A(z)$, but neither is the case. This is because of electrostatic induction effected by the capacitors onto each other. We note that $E_{A\&B}(z) - E_B(z)$ quickly falls off to zero, and we think it can be effectively represented by an Enge function.

A comparison of the electrostatic field differences $E_{A\&B}(z) - E_B(z)$ for cases $z_{\text{off}} \in \{5, 10, 20\}$ is shown in Fig. 2.26, and it demonstrates that the effect of electrostatic induction on the field $E_{A\&B}(z)$ decreases as the distance $2z_{\text{off}}$ between the electrostatic capacitors increases. The absolute value of $E_A(z) + E_B(z) - E_{A\&B}(z)$ is the highest near the edges of the capacitors.



(a) As z_{off} increases, the $E_{A\&B}(z) - E_B(z)$ curves approach their $z_{\text{off}} \to +\infty$ limit $E_A(z)$ (solid blue). Note the $z_{\text{off}} = 5$ (solid orange) curve quickly falling off to zero near $z/D = 5$ and forming a peculiar angle. This is because the edge of the capacitor $\mathscr{B}$ in the $z_{\text{off}} = 5$ case is at $z/D = 5$. In cases $z_{\text{off}} = 10$ and $z_{\text{off}} = 15$, the edge of the capacitor $\mathscr{B}$ is outside the range of the plot.



(b) Plots of $E_A(z) + E_B(z) - E_{A\&B}(z)$ are scaled by 0.1. As z_{off} increases, the $E_A(z) + E_B(z) - E_{A\&B}(z)$ curves decrease toward zero. In case of $z_{\text{off}} = 5$, the curve makes a second peak near the edge of the capacitor $\mathscr{B}$ at $z/D = 5$. In cases $z_{\text{off}} = 10$ and $z_{\text{off}} = 15$, the edge of the capacitor $\mathscr{B}$ is outside the range of the plot.

Figure 2.26: Electrostatic field differences $E_{A\&B}(z) - E_B(z)$ for cases $z_{\text{off}} = 5$ (solid orange), $z_{\text{off}} = 10$ (dashed green), and $z_{\text{off}} = 15$ (dot-dashed red), of the two adjacent semi-infinite capacitors with plates of $3D/4$ thickness, symmetric voltages, and rounded edges. The origin of each curve is at the edge of the left capacitor $\mathscr{A}$ in the respective case.

2.3.2.3 Two Adjacent Capacitors with Antisymmetric Voltages

Next, we consider two adjacent semi-infinite capacitors with full aperture $D = 2$, plates of $3D/4$ thickness $2r = 2.5$, and rounded edges. Let the voltages of these capacitors have the same magnitude and antisymmetric alignment along the x axis, i.e., the upper ($x > 0$) plates of these capacitors have voltages of the opposite sign. We specify and model the capacitors by a complex region bounded by the real axis at zero potential, the first capacitor's upper plate

$$\mathscr{A} = \{-z_{\text{off}} + t + di | t \in (-\infty, 0]\} \cup \left\{-z_{\text{off}} + (d+r)\, i + r \exp(it) \,|t \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]\right\} \cup$$
$$\cup \{-z_{\text{off}} + t + (d+2r)\, i | t \in (-\infty, 0]\}$$

at voltage $V = 1$, and the adjacent capacitor's upper plate

$$\mathscr{B} = \{z_{\text{off}} + t + di | t \in [0, +\infty)\} \cup \left\{z_{\text{off}} + (d+r)\, i - r \exp(it) \,|t \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]\right\} \cup$$
$$\cup \{z_{\text{off}} + t + (d+2r)\, i | t \in [0, +\infty)\}$$

at the opposite, antisymmetric voltage $V = -1$, where $d = 1$.

Due to reflection symmetry, the potential at the vertical axis $i\mathbb{R}$ is zero. Utilizing this symmetry, we reduce the problem to solving the Laplace equation in the complex region bounded by the capacitor's upper plate $\mathscr{A}$ at voltage $V = 1$ and the curve

$$\mathscr{C} = \{t | t \in (-\infty, 0]\} \cup \{it | t \in [0, +\infty)\}$$

at zero electrostatic potential.

Let the distance between the capacitors be $2z_{\text{off}} = 10$, not including the rounded parts of their edges. We approximate the arc by a piecewise linear curve with 42 line segments of equal lengths. This polygonal model is shown in Fig. 2.27.

We encode this polygonal physical domain in the *SC Toolbox* as listed in App. B.6.

We obtain a conformal mapping f from the bi-infinite strip $0 \le \Im(v) \le 1$ using the command

```
f6as1 = stripmap(p6as1, [46, 48], options)
```

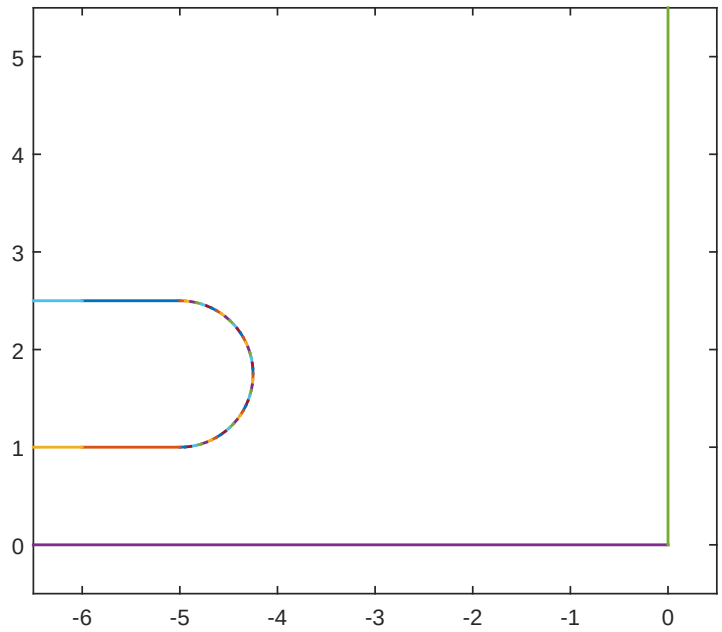


Figure 2.27: The polygonal model representing the two adjacent semi-infinite capacitors with plates of $3D/4$ thickness, antisymmetric voltages, and rounded edges.

with the 46th and 48th prevertices $+\infty$ and $-\infty$ mapping to the infinite vertices of the polygon.

The resulting conformal mapping f is as follows:

$$f(v) = f(v_0) + c\int_{v_0}^{v} \exp\left[\frac{\pi}{2}(\alpha_- - \alpha_+)\,\zeta\right] \prod_{j=1}^{n} \left[\sinh\frac{\pi}{2}(\zeta - v_j)\right]^{\alpha_j - 1} d\zeta,$$

where $n = 46$ is the number of finite prevertices of the polygon in the canonical domain,

$$\begin{aligned} v = (&0, 0.2082710702589, 0.2268694858472, 0.2471979527171,\\ &0.2689758999462, 0.2921056174591, 0.3165255488086, 0.3421986885406,\\ &0.3691065983606, 0.3972266859049, 0.4265528768239, 0.4570728278723,\\ &0.4887944128923, 0.5217160650131, 0.5558559351444, 0.5912199103837,\\ &0.6278324936766, 0.6657097679051, 0.7048910918775, 0.7454131347512,\\ &0.7873094700208, 0.8306281371978, 0.8754293623704, 0.9217656617397,\\ &0.9696928619268, 1.019283409164, 1.070604438361, 1.123710114681,\\ &1.178661078068, 1.235524415901, 1.294329412306, 1.355110639100,\\ &1.417856827234, 1.482560341451, 1.549149509315, 1.617540278889, \end{aligned}$$

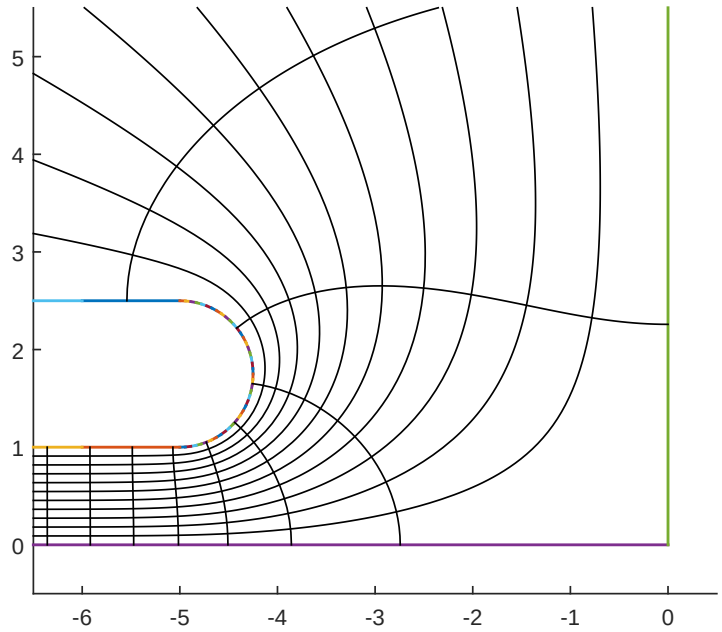


Figure 2.28: The electrostatic field and equipotential lines of the two adjacent semi-infinite capacitors with plates of $3D/4$ thickness, antisymmetric voltages, and rounded edges in the $z \leq 0$, $x \geq 0$ quadrant.

$$1.687562902355, 1.759014151587, 1.831592691508, 1.904946457872,$$

$$1.978601404745, 2.051925721199, 2.124069791109, 2.193382958646,$$

$$3.233898704303, 0.6626244510635 + i)$$

are the finite prevertices;

$$\alpha\pi = \left(1, \frac{85}{84}, \underbrace{\frac{43}{42}, \ldots, \frac{43}{42}}_{41 \text{ times}}, \frac{85}{84}, 1, \frac{1}{2}\right)\pi$$

are the respective interior angles of the polygon; and $\alpha_-\pi = -0.5\pi$, $\alpha_+\pi = 0$ are the divergence angles at the ends of the bi-infinite strip. The scaling constant is $c = -2.807110433411235 + i2.8071104400617424$.

Fig. 2.28 shows the resulting electrostatic field and equipotential lines of the two adjacent semi-infinite capacitors with plates of $3D/4$ thickness and rounded edges in the $z \leq 0$, $x \geq 0$ quadrant.

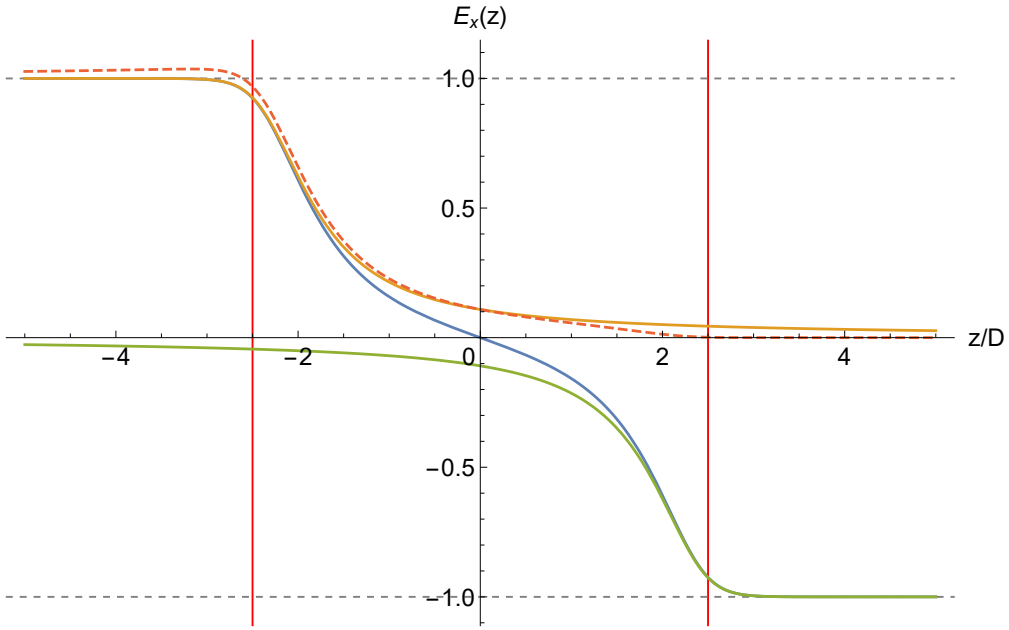
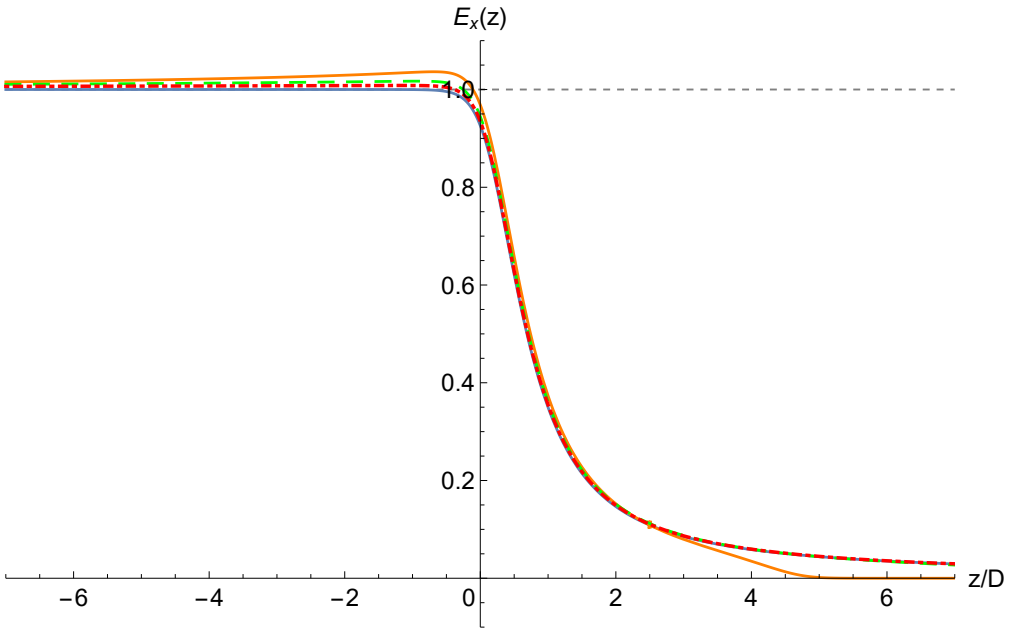


Figure 2.29: $E_{A\&B}(z)$ (solid blue) is the electrostatic field of two adjacent semi-infinite capacitors with plates of $3D/4$ thickness, asymmetric voltages, and rounded edges. $E_A(z)$ (solid orange) and $E_B(z)$ (solid green) would be the individual electrostatic fields of the left and the right capacitor in free space, respectively. The difference $E_{A\&B}(z) - E_B(z)$ (dashed red) demonstrates that $E_{A\&B}(z)$ is not a superposition of $E_A(z)$ and $E_B(z)$ due to electrostatic induction.

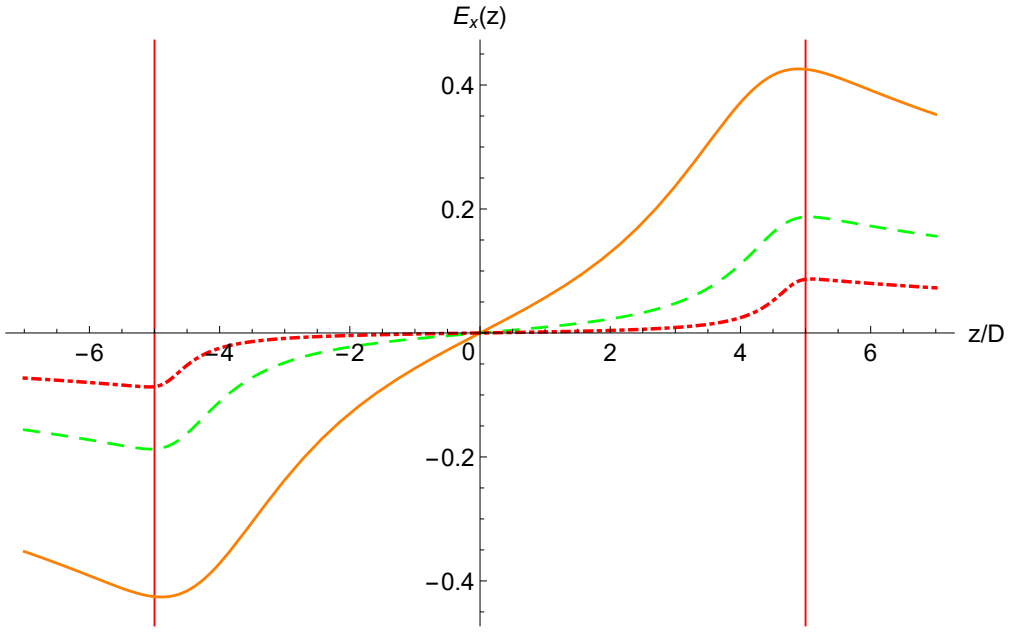
We denote the x component of the electrostatic field along the z axis of the two adjacent capacitors as $E_{A\&B}(z)$, of the left capacitor $\mathscr{A}$ in free space as $E_A(z)$, and of the right capacitor $\mathscr{B}$ in free space as $E_B(z)$.

Fig. 2.29 shows a plot of electrostatic fields $E_{A\&B}(z)$, $E_A(z)$, $E_B(z)$, and the difference $E_{A\&B}(z) - E_B(z)$. The plot demonstrates that the electrostatic field of the two adjacent capacitors $E_{A\&B}(z)$ is not a superposition of their individual electrostatic fields $E_A(z)$ and $E_B(z)$.

A comparison of the electrostatic field differences $E_{A\&B}(z) - E_B(z)$ for cases $z_{\text{off}} \in \{5, 10, 20\}$ is shown in Fig. 2.30, and it demonstrates that the effect of electrostatic induction on the field $E_{A\&B}(z)$ decreases as the distance $2z_{\text{off}}$ between the electrostatic capacitors increases. The absolute value of $E_A(z) + E_B(z) - E_{A\&B}(z)$ is the highest near the edges of the capacitors.



(a) As z_{off} increases, the $E_{A\&B}(z) - E_B(z)$ curves approach their $z_{\text{off}} \to +\infty$ limit $E_A(z)$ (solid blue). Note the $z_{\text{off}} = 5$ (solid orange) curve quickly falling off to zero near z/D =5 and forming a peculiar angle. This is because the edge of the capacitor $\mathscr{B}$ in the $z_{\text{off}} = 5$ case is at $z/D = 5$. In cases of $z_{\text{off}} = 10$ and $z_{\text{off}} = 15$, the edge of the capacitor $\mathscr{B}$ is outside the range of the plot. The origin of each curve is at the edge of the left capacitor $\mathscr{A}$ in the respective case.



(b) Plots of $E_A(z) + E_B(z) - E_{A\&B}(z)$ are scaled vertically by 0.1 and horizontally by $z_{\text{off}}/5$. The capacitor edges are at $z = \pm 5$ and denoted by red vertical gridlines. As z_{off} increases, the $E_A(z) + E_B(z) - E_{A\&B}(z)$ curves decrease toward zero.

Figure 2.30: Electrostatic field differences $E_{A\&B}(z) - E_B(z)$ for cases $z_{\text{off}} = 5$ (solid orange), $z_{\text{off}} = 10$ (dashed green), and $z_{\text{off}} = 15$ (dot-dashed red), of the two adjacent semi-infinite capacitors with plates of $3D/4$ thickness, asymmetric voltages, and rounded edges.

2.3.2.4 Two Adjacent Capacitors with Different Voltages

In the case of two adjacent semi-infinite capacitors with different voltages, the total number of different plate voltages is four, counting voltages of the same magnitude but opposite signs. Therefore, the method of solving the Laplace equation by finding a conformal mapping from a bi-infinite strip no longer applies.

However, the problem can be solved as follows [30, pp. 77–83]. Let V_1 and V_2 be the upper plate voltages of the left and the right semi-infinite capacitors, respectively. First, we obtain a conformal mapping f from the upper half-plane $H^+ \subset \mathbb{C}$ to the polygonal model that represents the upper half $x \geq 0$ of the two capacitors. As a result, the preimages of the capacitor's plates are intervals on the real axis, delimited by $\pm\infty$ and a set of points

$$-\infty < v_1 < \cdots < v_{n-1} < +\infty$$

for some $n \geq 3$, to account for the three constant Dirichlet boundary condition values 0 (on the real axis), V_1, and V_2.

For n Dirichlet boundary conditions $\phi_1, \ldots \phi_n$ and $n-1$ points $v_1, \ldots v_{n-1}$ on the real axis, the conformal mapping $\varphi : H^+ \to H^+$,

$$\begin{aligned}\phi(v) = \Re\Bigg[-\frac{i}{\pi}\Bigg(&\phi_1 \ln(v - v_1) + \phi_1 \ln\left(\frac{v - v_2}{v - v_1}\right) + \\ &+ \ldots + \phi_{n-1} \ln\left(\frac{v - v_{n-1}}{v - v_{n-2}}\right) + \phi_n \ln\left(\frac{1}{v_{n-1} - v}\right)\Bigg)\Bigg]\end{aligned}$$

satisfies [30, pp. 77–83]

$$\phi(x) = \begin{cases} \phi_1 & -\infty < v < v_1, \\ \phi_2 & v_1 < v < v_2, \\ \vdots & \vdots \\ \phi_{n-1} & v_{n-2} < v < v_{n-1}, \\ \phi_n & v_{n-1} < v < +\infty. \end{cases}$$

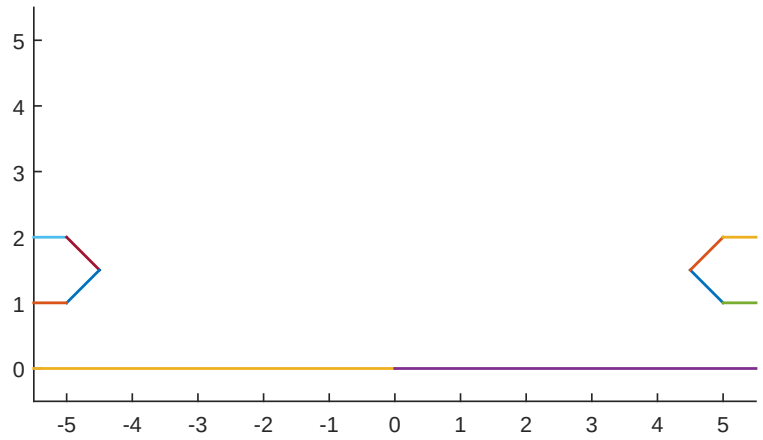


Figure 2.31: The polygonal model of the two adjacent semi-infinite capacitors with plates of $D/2$ thickness at different voltages.

It follows that the solution of the Laplace equation for the two adjacent semi-infinite capacitors with different voltages is

$$\varphi(w) = \Re\left(\phi\left(f^{-1}(w)\right)\right). \tag{2.8}$$

We can find a conformal mapping f using the *SC Toolbox*'s command `hplmap`. To find the conformal mapping ϕ, we can use the *SC Toolbox*'s command `lapsolve`.

The command `lapsolve` has no provisions for the crowding effect. Experimentation has shown that, in this case, modeling rounded plate edges using a high-fidelity piecewise linear curve results in no convergence or very low computational accuracy. Computational accuracy was somewhat acceptable when the rounded edges were modeled by two diagonal lines intersecting at 90° ("triangular edges").

Consider two adjacent semi-infinite capacitors with full aperture $D = 2$, plates of $D/2$ thickness $2r = 1$, and such triangular edges. Let the voltages of these capacitors have different magnitudes V_1, V_2 and symmetric alignment along the x axis, i.e., the upper ($x > 0$) plates of these capacitors have voltages of the same sign. We specify and model the capacitors by a complex region bounded by the real axis at zero potential, the first capacitor's upper plate

$$\mathscr{A} = \{-z_{\text{off}} + t + di | t \in (-\infty, 0]\} \cup \{-z_{\text{off}} + di + (r + ri)t | t \in [0,1]\} \cup$$
$$\cup \{-z_{\text{off}} + (d + 2r)i + (r - ri)t | t \in [0,1]\} \cup \{-z_{\text{off}} + t + (d + 2r)i | t \in (-\infty, 0]\}$$

at voltage V_1, and the adjacent capacitor's upper plate

$$\mathscr{B} = \{z_{\text{off}} + t + di | t \in [0, +\infty)\} \cup \{z_{\text{off}} + di + (ri - r)t | t \in [0,1]\} \cup$$
$$\cup \{z_{\text{off}} + (d + 2r)i - (ri - r)t | t \in [0,1]\} \cup \{z_{\text{off}} + t + (d + 2r)i | t \in [0, +\infty)\}$$

at voltage V_2, where $d = 1$. For definiteness, let $V_1 = 1$ and $V_2 = 3$. Let the distance between the capacitors be $2z_{\text{off}} = 10$, not including their edges. This polygonal model is shown in Fig. 2.31.

We encode this polygonal physical domain in the *SC Toolbox* by its vertices w_j and interior angles $\alpha_j \pi$ as

```
p3bm = polygon([0, inf, 5.+1.i, 4.5+1.5i, 5.+2.i, 6.+2.i, inf,...
    -6.+2.i, -5.+2.i, -4.5+1.5i, -5.+1.i, inf],...
    [1, 0, 5/4, 3/2, 5/4, 1, -1, 1, 5/4, 3/2, 5/4, 0]);
```

where vertices 0, $-6+2i$, and $6+2i$ were added as guiding points for the *SC Toolbox*'s algorithm.

We obtain a conformal mapping f from the upper half-plane H^+ using the command `hplmap`.

```
f3bm = hplmap(p3bm, options);
```

Next, we compute the composite conformal mapping as in eq. 2.8 using the command `lapsolve`

```
bmod = [3, 3, 3, 3, 1, 1, 1, 1, 1, 0, 0, 3];
phi = lapsolve(f3bm, bmod);
```

where its Dirichlet boundary conditions are supplied as the vector `bmod`.

Fig. 2.32 shows a plot of electrostatic fields $E_{A\&B}(z)$, $E_A(z)$, $E_B(z)$, and the difference $E_{A\&B}(z) - E_B(z)$. The plot demonstrates that the electrostatic field of the two adjacent capacitors $E_{A\&B}(z)$ is not a superposition of their individual electrostatic fields $E_A(z)$ and $E_B(z)$.

A comparison of the electrostatic field differences $E_{A\&B}(z) - E_B(z)$ for cases $z_{\text{off}} \in \{5, 7.5, 10\}$ is shown in Fig. 2.33, and it demonstrates that the effect of electrostatic induction on the field $E_{A\&B}(z)$ decreases as the distance $2z_{\text{off}}$ between the electrostatic capacitors increases. The absolute value of $E_A(z) + E_B(z) - E_{A\&B}(z)$ is the highest near the edges of the capacitors.

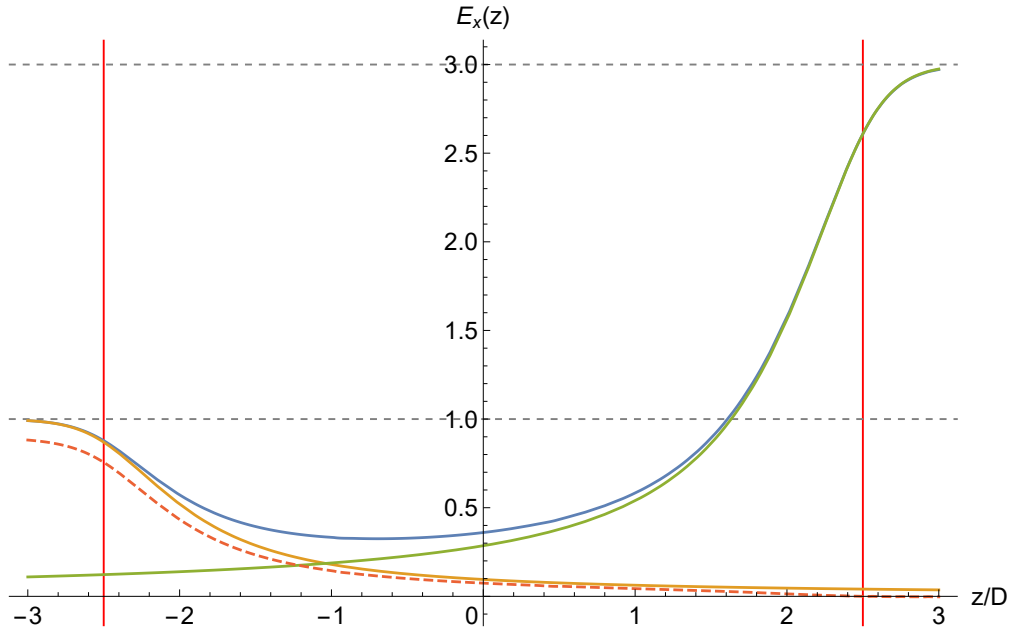


Figure 2.32: $E_{A\&B}(z)$ (solid blue) is the electrostatic field of the two adjacent semi-infinite capacitors with plates of $D/2$ thickness and different voltages. $E_A(z)$ (solid orange) and $E_B(z)$ (solid green) would be the individual electrostatic fields of the left and the right capacitor in free space, respectively. The difference $E_{A\&B}(z) - E_B(z)$ (dashed red) demonstrates that $E_{A\&B}(z)$ is not a superposition of $E_A(z)$ and $E_B(z)$ due to electrostatic induction. The capacitor edges in this $z_{\text{off}} = 5$ case are at $z/D = \pm z_{\text{off}}/D = \pm 2.5$, denoted by red vertical gridlines.

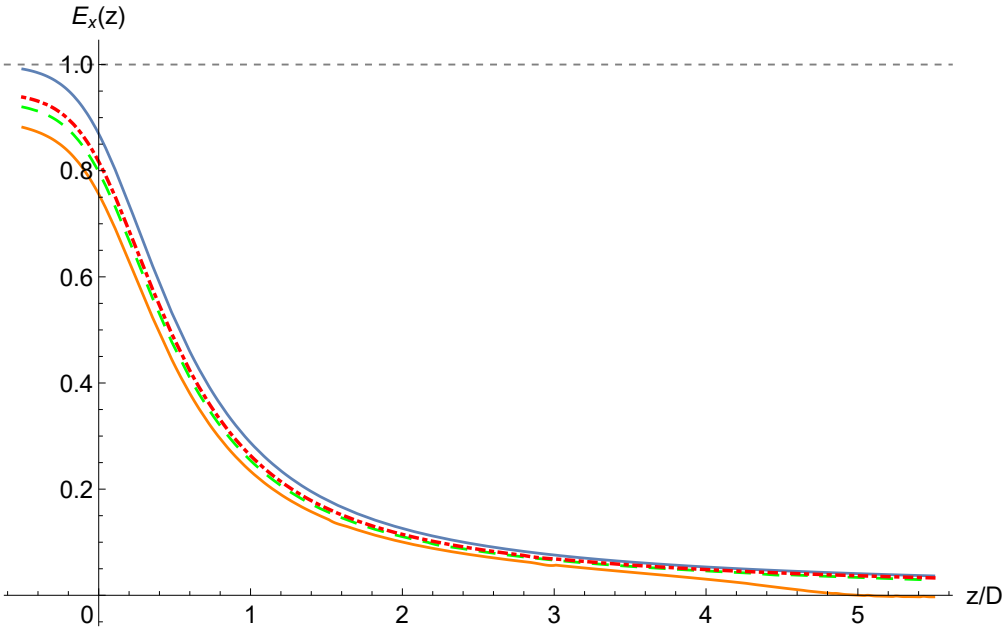
2.4 Finite Rectangular Electrostatic Capacitors

Helmut Soltner (Forschungszentrum Jülich, Germany) had calculated the electrostatic field for several finite rectangular electrostatic capacitors with finitely thick plates. The calculations were performed in *COULOMB*.

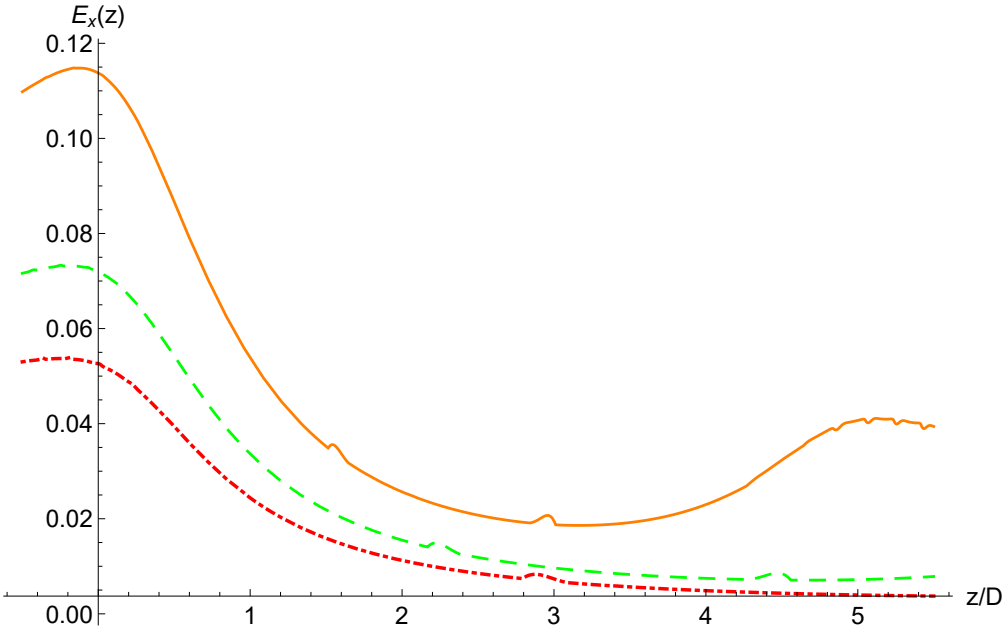
2.4.1 *COULOMB*

COULOMB is a 3D electric computer-aided design (CAD) and analysis software program [32]. *COULOMB*'s boundary element method (BEM) field solver is useful for calculations of field falloffs of electrostatic particle optical elements.

The 3D model of a particle optical element is specified as a solid part in a STEP-formatted [55] file. Surfaces of the 3D model are assigned voltages and are discretized by meshes. A grid-point set of coordinates, listed in an input text file, specifies where the electrostatic potential or the



(a) As z_{off} increases, the $E_{A\&B}(z) - E_B(z)$ curves approach their $z_{\text{off}} \to +\infty$ limit $E_A(z)$ (solid blue). Note the $z_{\text{off}} = 5$ (solid orange) curve quickly falling off to zero near $z/D = 5$ and forming a peculiar angle. This is because the edge of the capacitor $\mathscr{B}$ in the $z_{\text{off}} = 5$ case is at $z/D = 5$. In cases $z_{\text{off}} = 7.5$ and $z_{\text{off}} = 10$, the edge of the capacitor $\mathscr{B}$ is outside the range of the plot.



(b) Plots of $E_A(z) + E_B(z) - E_{A\&B}(z)$ are scaled vertically by 0.1. As z_{off} increases, the $E_A(z) + E_B(z) - E_{A\&B}(z)$ curves decrease toward zero. The $z_{\text{off}} = 5$ curve makes a second peak near the edge of the capacitor $\mathscr{B}$ at $z/D = 5$. In cases $z_{\text{off}} = 7.5$ and $z_{\text{off}} = 10$, the edge of the capacitor $\mathscr{B}$ is outside the range of the plot. Some artifacts due to computation errors are visible.

Figure 2.33: Electrostatic field differences $E_{A\&B}(z) - E_B(z)$ for cases $z_{\text{off}} = 5$ (solid orange), $z_{\text{off}} = 7.5$ (dashed green), and $z_{\text{off}} = 10$ (dot-dashed red), of the two adjacent semi-infinite capacitors with plates of $D/2$ thickness and different voltages. The origin of each curve is at the edge of the left capacitor $\mathscr{A}$ in the respective case.

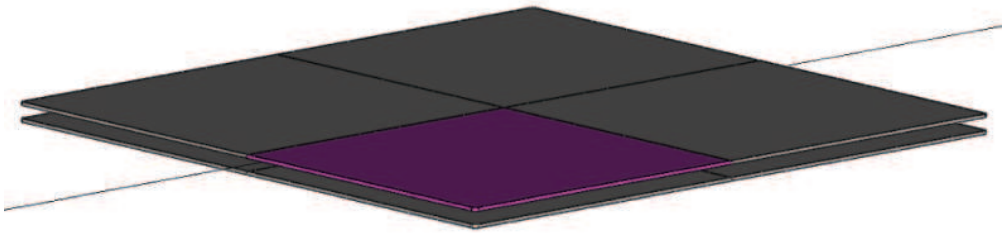


Figure 2.34: Rendering of the large rectangular capacitor with plates of $D/4$ thickness and rounded edges. The infinite line represents the z axis. One-eighth of the capacitor geometry, denoted by purple, is stored in the STEP file. The full geometry is recovered in *COULOMB* by specifying the three reflection symmetries.

electrostatic field is to be calculated.

Next, the BEM solver performs calculations, and the results are written to a text file. Our calculations typically took several hours to complete.

2.4.2 Large Rectangular Capacitor

Consider a rectangular electrostatic capacitor comprising solid plates of $D/4$ thickness

$$\mathscr{A} = f\left(\{(x,y,z)\,|x \in [1,1.5]\text{ cm}; y,z \in [-1,1]\text{ m};\}\right)$$

at voltage $V = 1\,\text{V}$ and

$$\mathscr{B} = f\left(\{(x,y,z)\,|x \in -[1,1.5]\text{ cm}; y,z \in [-1,1]\text{ m};\}\right)$$

at voltage $V = -1\,\text{V}$, respectively. The function f represents adding material to the plates to form rounded edges. This rectangular capacitor is shown in Fig. 2.34.

The *COULOMB* calculations were performed with about $85,000$ boundary elements and produced electrostatic field data at points

$$H = \left\{\left(k_x h_x, k_y h_y, 40\,\text{cm} + k_z h_z\right)|k_x, k_y = 1,2,\ldots,10; k_z = 1,2,\ldots,100\right\},$$

where $\left(h_x, h_y, h_z\right) = (0.1,5,0.2)$ cm.

The falloff of the x component $E_x(z)$ of the electrostatic field along the z axis is shown in Fig. 2.35.

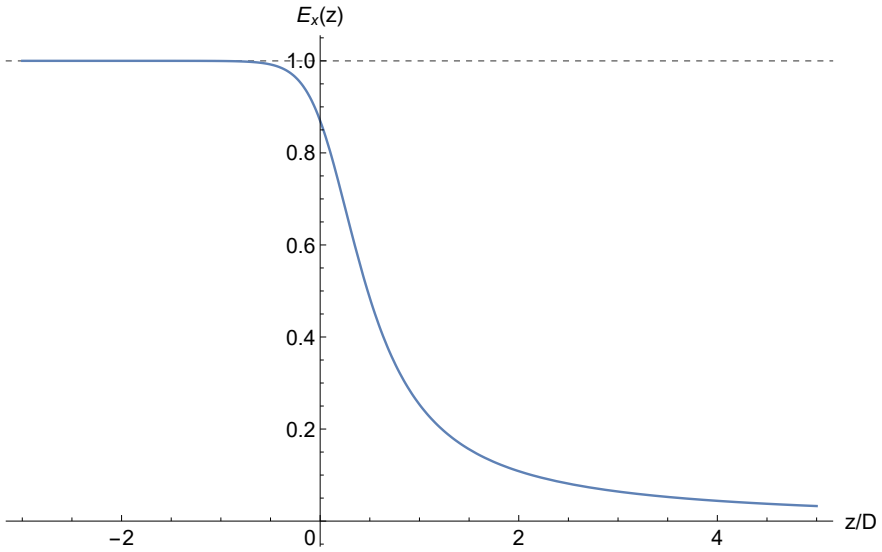


Figure 2.35: Electrostatic field falloff $E_x(z)$ of the large rectangular capacitor with plates of $D/4$ thickness and rounded edges.

2.4.3 Small Rectangular Capacitor

Consider a smaller rectangular electrostatic capacitor comprising solid plates of $D/4$ thickness

$$\mathscr{A} = f\left(\{(x,y,z)\,|x \in [1,1.5]\text{ cm}; y \in [-5.5,5.5]\text{ cm}; z \in [-1,1]\text{ m}\}\right)$$

at voltage $V = 1\,\text{V}$ and

$$\mathscr{B} = f\left(\{(x,y,z)\,|x \in -[1,1.5]\text{ cm}; y \in [-5.5,5.5]\text{ cm}; z \in [-1,1]\text{ m}\}\right)$$

at voltage $V = -1\,\text{V}$, respectively. The function f represents adding material to the plates to form rounded edges.

The *COULOMB* calculations were performed with $70,000$ boundary elements and produced electrostatic field data at points

$$H = \left\{\left(k_x h_x, k_y h_y, 40\,\text{cm} + k_z h_z\right)|k_x, k_y = 1,2,\ldots,10; k_z = 1,2,\ldots,100\right\},$$

where $\left(h_x, h_y, h_z\right) = (0.1, 0.05, 0.2)$ cm.

The falloff of the x component $E_x(z)$ of the electrostatic field along the z axis is shown in Fig. 2.36.

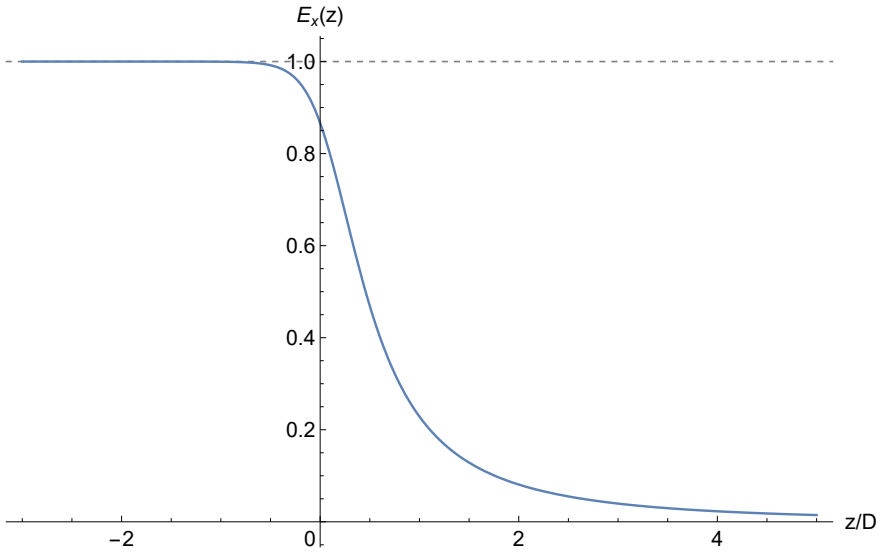


Figure 2.36: Electrostatic field falloff $E_x(z)$ of the small rectangular capacitor with plates of $D/4$ thickness and rounded edges.

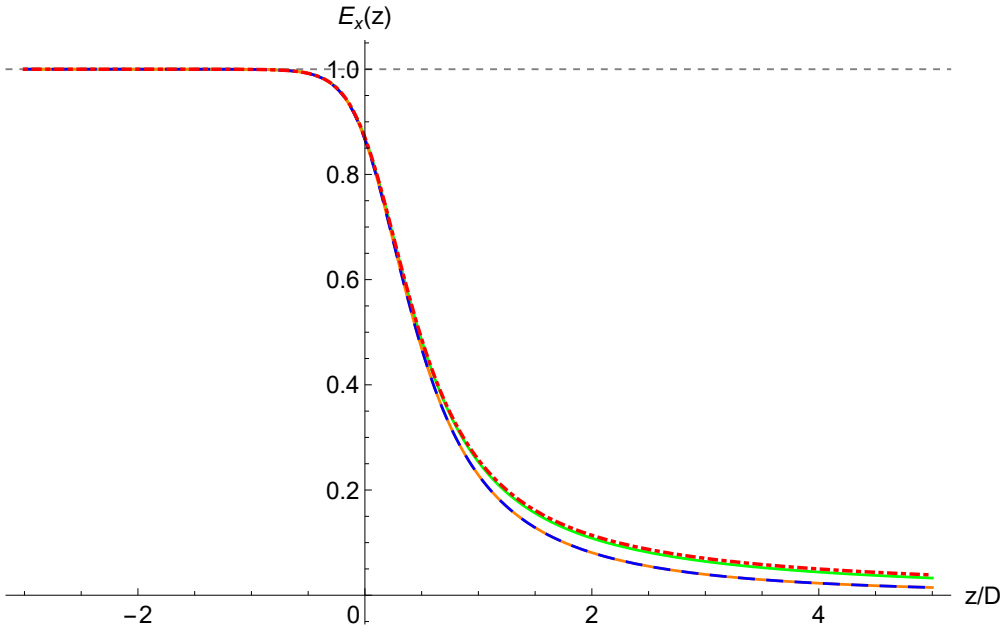


Figure 2.37: Electrostatic field falloffs $E_x(z)$ of the large (solid green) and small (solid orange) rectangular capacitors modeled by $70,000$ boundary elements, the small rectangular capacitor modeled by $150,000$ boundary elements (dashed blue), and the semi-infinite capacitor with plates of $D/4$ thickness and rounded edges (dot-dashed red). The $E_x(z)$ curves of the small rectangular capacitor modeled by $70,000$ and $150,000$ boundary elements visually coincide. As expected, the $E_x(z)$ curve of the large rectangular capacitor is only slightly lower than of the semi-infinite capacitor with finitely thick plates and rounded edges.

To assess the accuracy of the *COULOMB* calculations, the calculations were repeated with $150,000$ boundary elements instead of the initial $70,000$. Fig. 2.37 shows, *inter alia*, a comparison of the electrostatic field falloffs in these two cases. We are satisfied with the initial accuracy of the calculations, as the electrostatic field falloffs are visually indistinguishable.

2.5 Comparison of Single-Capacitor Results

In the above, we obtained electrostatic field falloffs for the following cases of one electrostatic capacitor:

1. Semi-infinite capacitor with infinitely thin plates, manually obtained conformal mapping;
2. Semi-infinite capacitor with infinitely thin plates, *SC Toolbox* calculations;
3. Semi-infinite capacitor with infinitely thick plates, *SC Toolbox* calculations;
4. Semi-infinite capacitor with plates of $D/20$ thickness, *SC Toolbox* calculations;
5. Semi-infinite capacitor with plates of $D/20$ thickness and rounded edges, *SC Toolbox* calculations;
6. Semi-infinite capacitor with plates of $D/4$ thickness and rounded edges, *SC Toolbox* calculations;
7. Large rectangular capacitor with plates of $D/4$ thickness, *COULOMB* calculations; and
8. Small rectangular capacitor with plates of $D/4$ thickness, *COULOMB* calculations.

We compare the falloffs of the electrostatic fields $E_x(z)$ from cases 1, 3, 4, and 5 in Fig. 2.38. The electrostatic fields $E_x(z)$ from cases 2, 3, 5, 6, 7, and 8 are compared in Fig. 2.39. In Fig. 2.22 above, we compared the electrostatic fields $E_x(z)$ for cases 1 and 2. We observed a general pattern that the electrostatic field falloffs are in or near the region bounded by the electrostatic field falloffs of semi-infinite capacitors with infinitely thin and infinitely thick plates. Increasing the thickness of the plates – *ceteris paribus* – visibly increased $E_x(z)$ for $z/D \gtrsim 1$.

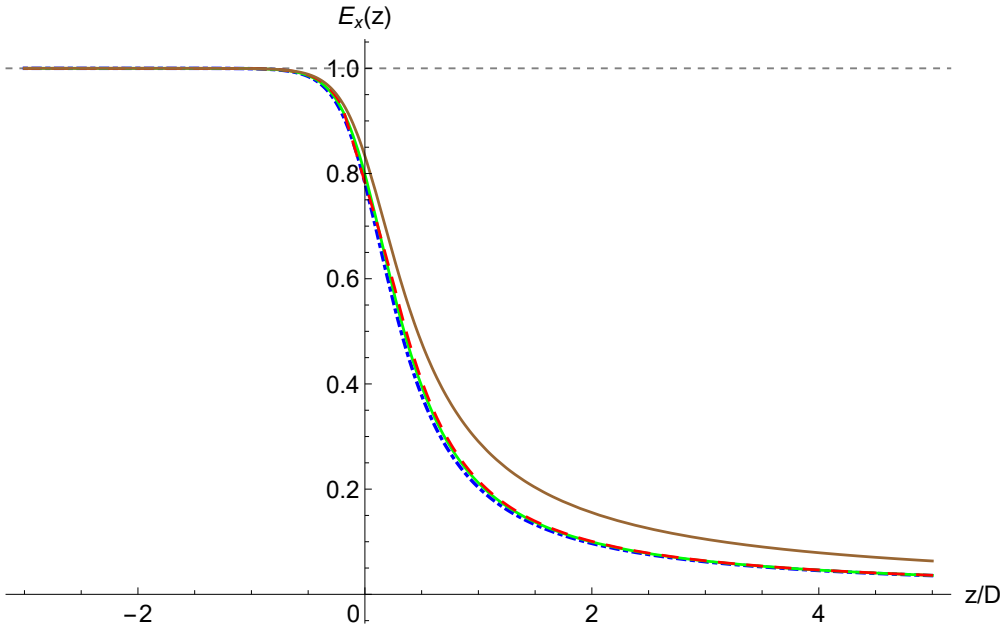


Figure 2.38: The electrostatic field falloffs $E_x(z)$ in cases 1 (dot-dashed blue), 3 (solid brown), 4 (solid green), and 5 (dashed red).

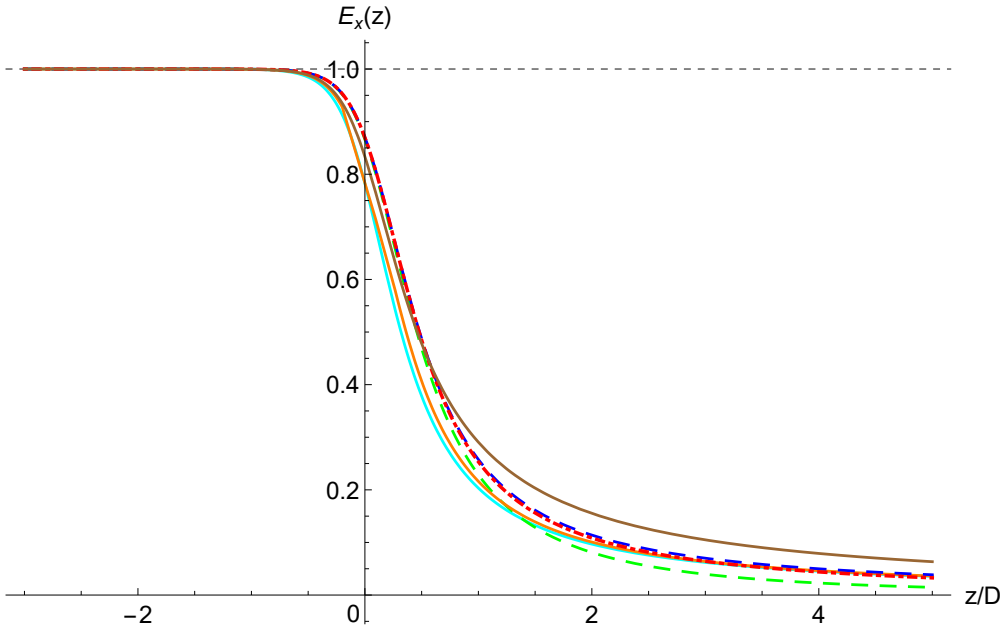


Figure 2.39: The electrostatic field falloffs $E_x(z)$ in cases 2 (solid cyan), 3 (solid brown), 5 (solid orange), 6 (dashed blue), 7 (dashed green), and 8 (dot-dashed red).

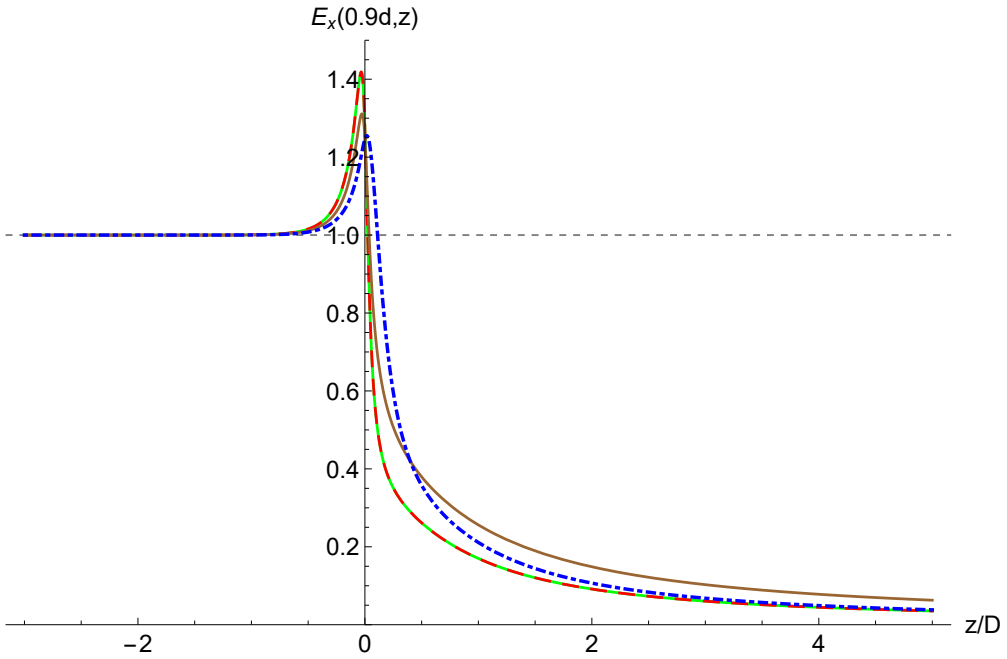


Figure 2.40: The x component $E_x(0.9d, z)$ of the electrostatic field along the $x = 0.9d$ line in cases 1 (solid green), 2 (dashed red), 3 (solid brown), and 6 (dot-dashed blue).

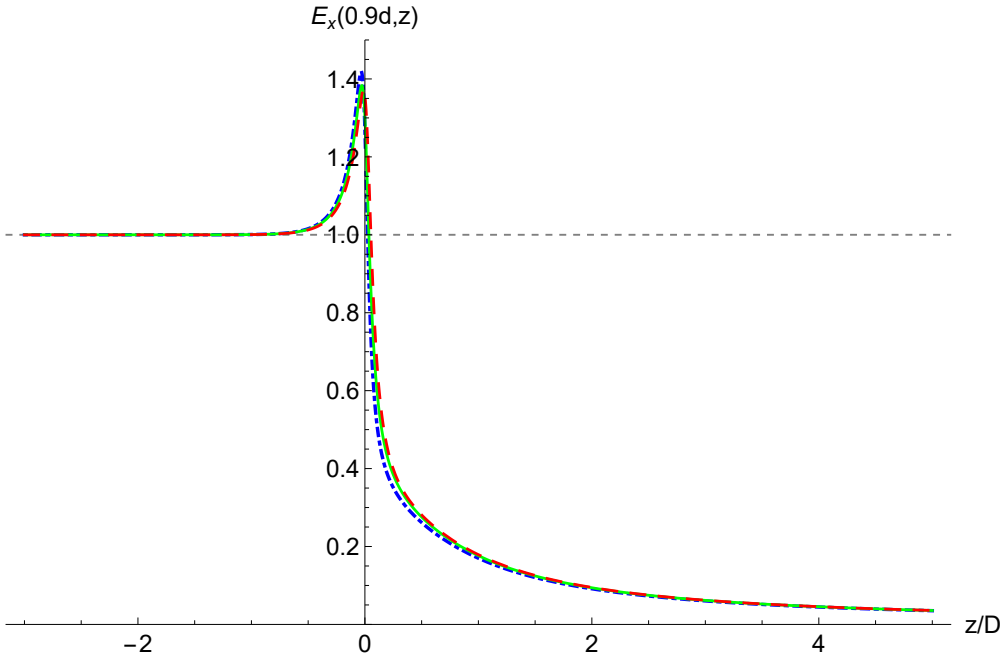


Figure 2.41: The x component $E_x(0.9d, z)$ of the electrostatic field along the $x = 0.9d$ line in cases 1 (dot-dashed blue), 4 (solid green), and 5 (dashed red).

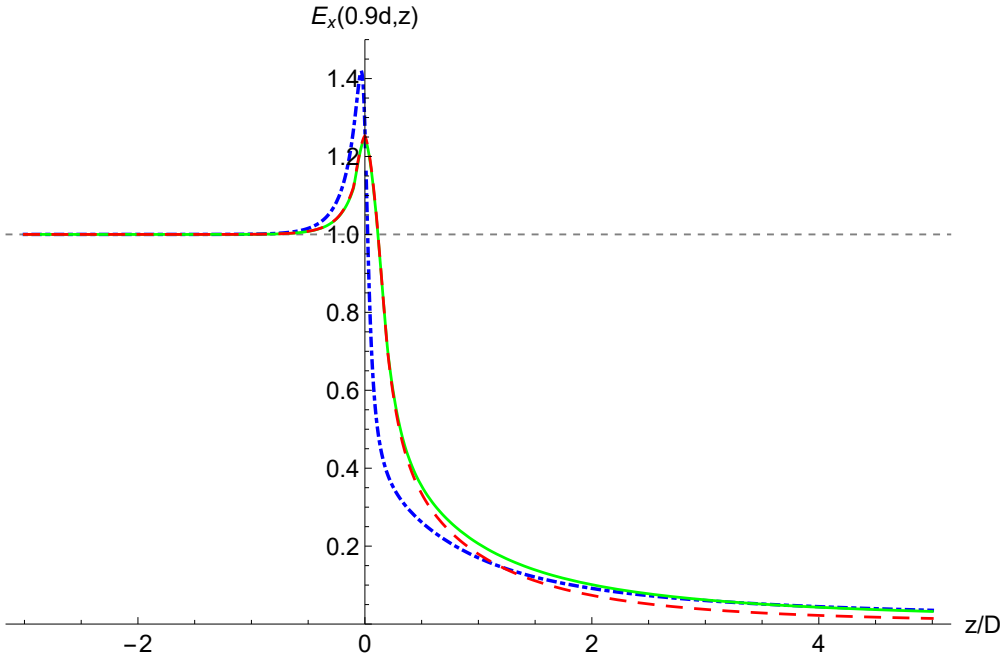


Figure 2.42: The x component $E_x(0.9d,z)$ of the electrostatic field along the $x = 0.9d$ line in cases 1 (dot-dashed blue), 7 (solid green), and 8 (dashed red).

As illustrated previously in Fig. 2.9, the plate surface charge density $\sigma(z)$ spikes near the plate edge. Accordingly, the electrostatic field $E_x(x,z)$ along a line $x = a$ near the plate at $x = d$ spikes near the plate edge at $z = 0$ (same applies to the lower plate at $x = -d$). As an example, we consider the line $x = 0.9d$. Falloffs of the x component $E_x(0.9,z)$ of the electrostatic field along the line $x = 0.9d$ in cases 1, 2, 3, 6 are shown in Fig. 2.40, in cases 1, 4, 5 in Fig. 2.41, and in cases 1, 7, 8 in Fig. 2.42. The results are consistent with the standard electrostatics result [56, pp. 75–79] that the surface charge density tends to be higher near sharp edges and not as high near blunt or rounded edges.

2.6 Accurate Fringe Fields Representation

Enge functions

$$F_N(z) = \frac{1}{1+\exp\left(\sum_{j=1}^{N} a_j \left(\frac{z}{D}\right)^{j-1}\right)},$$

are effective at representing fringe fields of magnetic dipoles. As Fig. 2.43 illustrates, fringe fields of electrostatic deflectors fall off slower than fringe fields of magnetic dipoles.

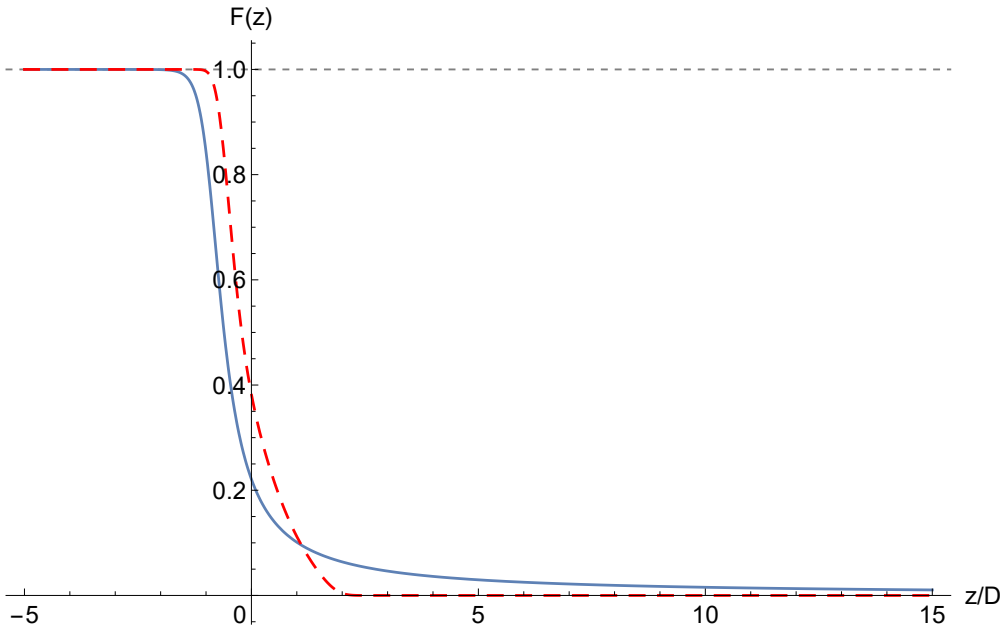


Figure 2.43: The electrostatic field falloff $E_x(z)$ of a semi-infinite electrostatic capacitor with infinitely thin plates (solid blue), compared to the magnetic field falloff of a homogeneous magnetic dipole (dashed red). Each curve is horizontally offset so that origin $z = 0$ is at its EFB (the EFB_{20}, computed with $z_{\text{ext}}/D = 20$, in the electrostatic case).

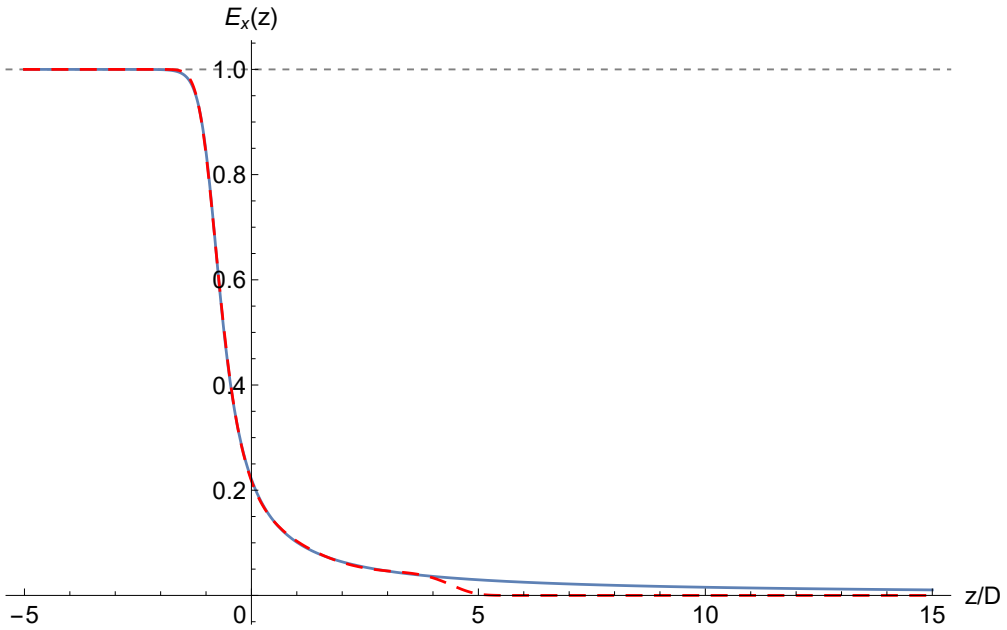


Figure 2.44: An Enge function $F_5(z)$ (dashed red), fitted to the electrostatic field falloff $E_x(z)$ (solid blue) of a semi-infinite capacitor with infinitely thin plates. The electrostatic field falloff $E_x(z)$ is scaled to 1 well inside the capacitor. The origin $z = 0$ is at the EFB_{20}.

Coefficient	Value
a_1	1.2769683902492515
a_2	1.4049887582360876
a_3	−0.882202957166947
a_4	0.48659244279111313
a_5	−0.1378864476786552
a_6	0.014397164677247012

Table 2.1: Coefficients of the Enge function $F_5(z)$, fitted to the electrostatic field falloff $E_x(z)$ of the semi-infinite capacitor with infinitely thin plates. The Enge function is accurate for $z/D \leq 4$.

Coefficient	Value
b_1	4.057204346021459
b_2	5.656314468345938
b_3	0.08405676267706107
b_4	−0.0074011699896916186
b_5	0.0003532805303953159
b_6	$-6.893356779357708 \times 10^{-6}$

Table 2.2: Fitted coefficients of the function $H_1(z)$, approximating the electrostatic field falloff $E_x(z)$ at $z/D \geq 3$.

Consider the x component $E_x(z)$ of the electrostatic field along the z axis of a semi-infinite electrostatic capacitor with infinitely thin plates. In this section, $E_x(z)$ is offset by the EFB_{20} $z_{\text{EFB}}/D = 0.920945$ so that the EFB_{20} is at the origin $z = 0$. Fig. 2.44 shows a 5th order Enge function $F_5(z)$ as a representative attempt to fit an Enge function to the electrostatic field $E_x(z)$. The resulting Enge function coefficients are listed in Table 2.1. Unlike $E_x(z)$, the Enge function $F_5(z)$ quickly falls off to zero.

Increasing the number of Enge function order N resulted in some erratic-looking Enge function behavior and did not resolve this issue. The same issue was observed for fringe field falloffs of other electrostatic deflectors. We concluded that this is because the field falloff of an electrostatic deflector is slower than exponential.

Since the electrostatic field of a point charge falls off as $\sim 1/r^2$ and the electrostatic field of a semi-infinite capacitor is an integral of charge density over its plates, we supposed that the fringe field falloff of a semi-infinite capacitor is similar in form to $1/r$.

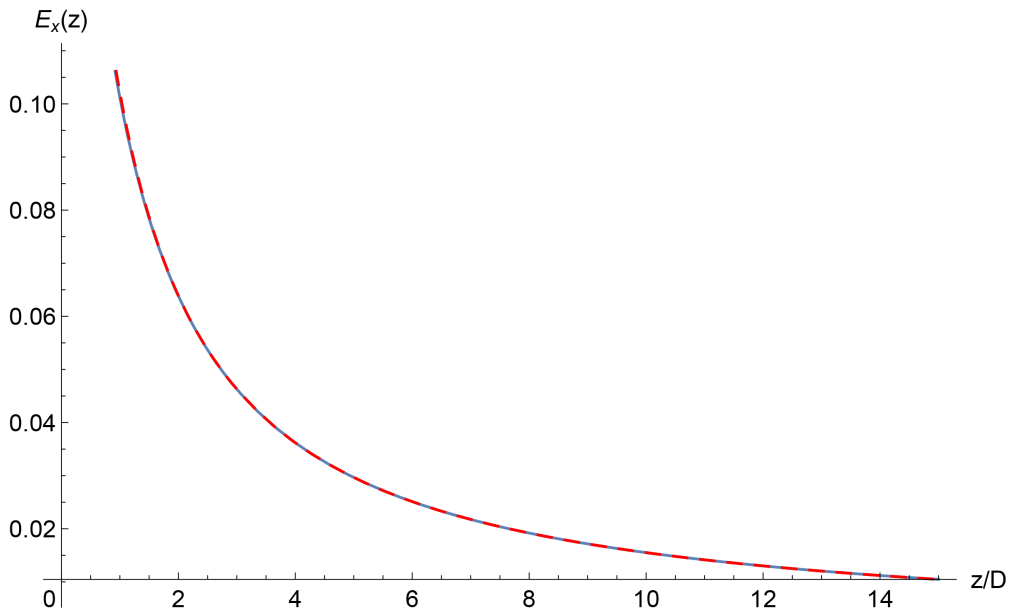


Figure 2.45: Function $H_1(z)$ (dashed red), fitted to the electrostatic field falloff $E(z)$ (solid blue) over the interval $3 \leq z/D \leq 15$. The origin $z = 0$ is at the EFB_{20}.

The Enge function $F_5(z)$ is a good approximation of $E_x(z)$ up to about $z/D = 4$, as Fig 2.44 shows. We note that this Enge function is already a significant improvement over using magnetic dipole Enge function coefficients to model the fringe field of the electrostatic deflector.

Complementarily, we fitted the function

$$H_1(z) = \frac{1}{\sum_{j=1}^{N} b_j \left(\frac{z}{D}\right)^{j-1}}$$

to $E_x(z)$ in the interval $3 \leq z/D \leq 15$. The results of the fit are shown in Fig. 2.45. The fitted values of coefficients b_j are listed in Table 2.2.

Next, we smoothly glued the Enge function $F(z)$ and the function $H_1(z)$ using the formula

$$H_2(z) = F_5(z) \frac{1}{1+\exp\left[\left(\frac{z}{D}-3.5\right)^2\right]} + H_1(z) \frac{1}{1+\exp\left[-\left(\frac{z}{D}-3.5\right)^2\right]}.$$

We propose the general form

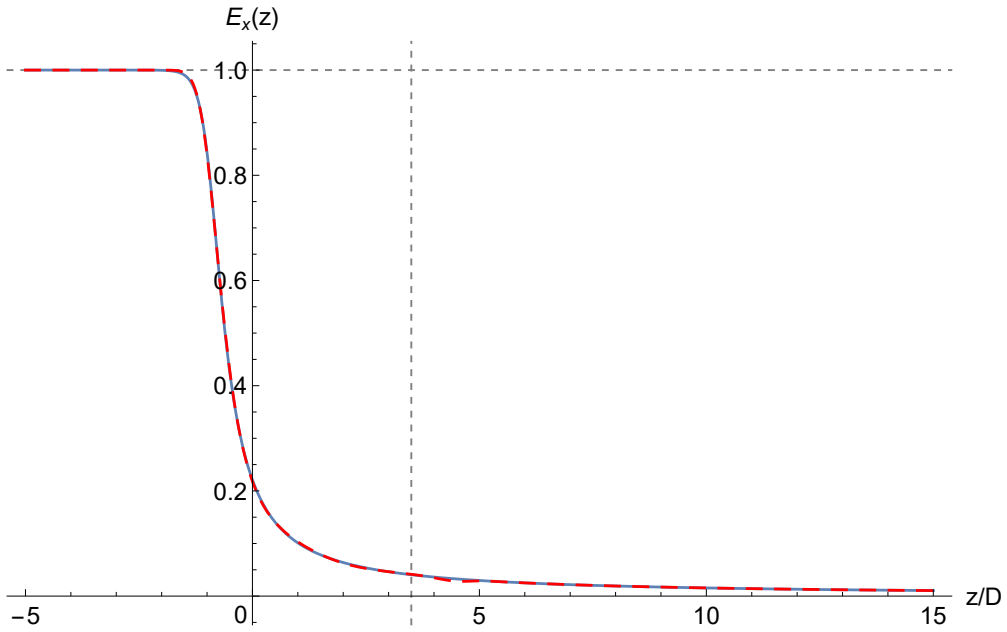
$$H(z) = \frac{1}{1+\exp\left[\sum_{j=1}^{N_1} a_j \left(\frac{z}{D}\right)^{j-1}\right]} \frac{1}{1+\exp\left[\left(\frac{z}{D}-c\right)^2\right]} + \\ + \frac{1}{\sum_{j=1}^{N_2} b_j \left(\frac{z}{D}\right)^{j-1}} \frac{1}{1+\exp\left[-\left(\frac{z}{D}-c\right)^2\right]}.$$

for such an approximation of the field falloff of an electrostatic deflector, where a_j, b_j, and c are parameters.

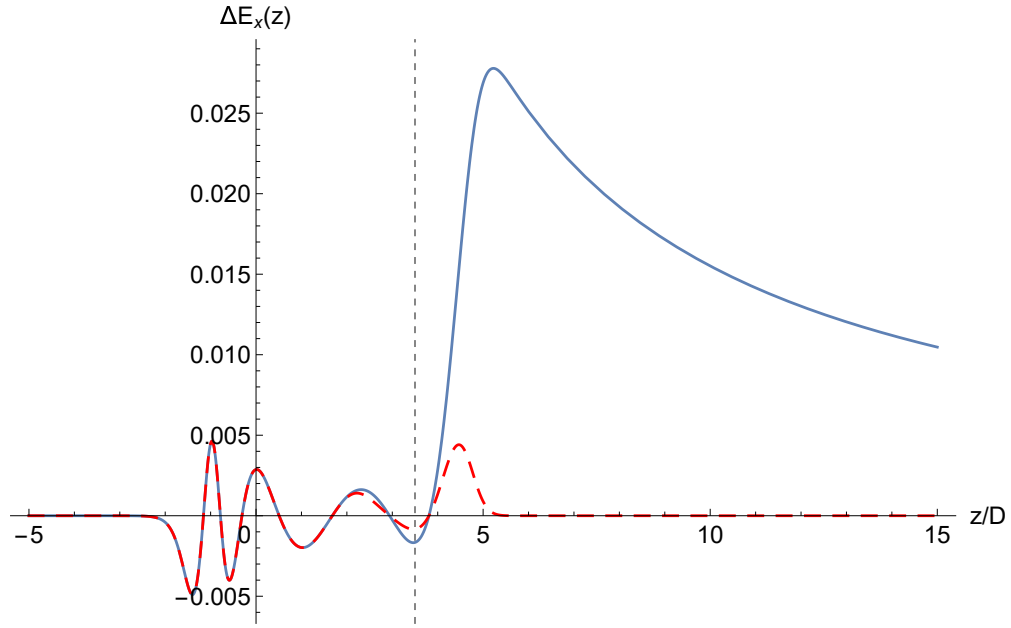
The function $H_2(z)$ is plotted in Fig. 2.46 against $E_x(z)$ and shows a good fit over at least $-5 \leq z/D \leq 15$, with $\sup_{-5\leq z/D\leq 15} |H_2(z) - E_x(z)|$ of about 0.005, and with the difference $|H_2(z) - E_x(z)|$ the most significant in the interval $-2 \leq z/D \leq 5$.

Not completely satisfied with the accuracy of $H_2(z)$, we calculated a numerical approximation $H_3(z)$ to the Fourier exponential series of $E_x(z) - H_2(z)$ of order 20, with Fourier parameters $a = 1$ and $b = \pi/5$, in the interval $1.5 - \pi/|b| \leq z/D \leq 1.5 + \pi/|b|$ (i.e., $-3.5 \leq z/D \leq 6.5$). The resulting Fourier expansion $H_3(z)$ is

$$
\begin{aligned}
H_3(z) = {} & (0.000034592190696502585 + 0.000015478461970141042i)\exp(-4i\pi(z/D-1.5)) - \\
& - (0.000018077924429864712 - 0.000045932575389375 67i)\exp(-19i\pi(z/D-1.5)/5) - \\
& - (0.000063530822760611 36 + 0.000023106674840721097i)\exp(-18i\pi(z/D-1.5)/5) + \\
& + (0.000028478297945671638 - 0.000080177795306942 98i)\exp(-17i\pi(z/D-1.5)/5) + \\
& + (0.000110302963576555 44 + 0.000031545142879819836i)\exp(-16i\pi(z/D-1.5)/5) - \\
& - (0.000043749144828268773 - 0.000136871734271458 14i)\exp(-3i\pi(z/D-1.5)) - \\
& - (0.000173898094678503 26 + 0.000037412570641180 89i)\exp(-14i\pi(z/D-1.5)/5) + \\
& + (0.000060835817049885 6 - 0.000223951085115175 42i)\exp(-13i\pi(z/D-1.5)/5) + \\
& + (0.000246439871275104 84 + 0.000039439492222297 56i)\exp(-12i\pi(z/D-1.5)/5) - \\
& - (0.000051390417457536415 - 0.000330892554475562 5i)\exp(-11i\pi(z/D-1.5)/5) - \\
& - (0.000299184317057232 6 + 0.000050383147167314 68i)\exp(-2i\pi(z/D-1.5)) - \\
& - (0.000033648323544147 2 + 0.000412262038052496 85i)\exp(-9i\pi(z/D-1.5)/5) + \\
& + (0.000272884644367153 97 + 0.000059492609979352665i)\exp(-8i\pi(z/D-1.5)/5) + \\
& + (0.000188991361000 99786987 + 0.000311413330689541 47i)\exp(-7i\pi(z/D-1.5)/5) - \\
& - (0.000117324374869032 5 + 0.000082376636409463 21i)\exp(-6i\pi(z/D-1.5)/5) - \\
& - (0.000268591328898847 1 - 0.000227766412409432i)\exp(i\pi(z/D-1.5)) -
\end{aligned}
$$



(a) $H_2(z)$ (dashed red) and $E_x(z)$ (solid blue).



(b) $H_2(z) - E_x(z)$ (dashed red) and $F_5(z) - E_x(z)$ (solid blue). $\sup_{-5 \le z/D \le 15} |H_2(z) - E_x(z)|$ is about 0.005.

Figure 2.46: The piecewise function $H_2(z)$ is an improved approximation of electrostatic field falloff $E_x(z)$ compared to the Enge function $F_5(z)$. The function $H_2(z)$ is formed by smoothly gluing $F_5(z)$ and $H_1(z)$ at $z/D = 3.5$, indicated by the vertical gridline. The origin $z = 0$ is at the EFB_{20}.

$$
\begin{aligned}
&- (0.00012021044338404635 - 0.0006224497940703973i) \exp(-4i\pi (z/D - 1.5)/5) - \\
&- (0.0003677923876745 23 + 0.00012020905645207353i) \exp(-3i\pi (z/D - 1.5)/5) - \\
&- (0.0001448084077170699 + 0.00026198489039169006i) \exp(-2i\pi (z/D - 1.5)/5) - \\
&- (0.00025995183468993 67 - 0.00046040606549954 73i) \exp(-i\pi (z/D - 1.5)/5) + \\
&+ 0.00014791662685726133 - \\
&- (0.00025995183468993 78 + 0.00046040606549954 72) \exp(1i\pi (z/D - 1.5)/5) - \\
&- (0.0001448084077170699 - 0.00026198489039169) \exp(2i\pi (z/D - 1.5)/5) - \\
&- (0.00036779238767452 28 - 0.00012020905645207357) \exp(3i\pi (z/D - 1.5)/5) - \\
&- (0.00012021044338404654 + 0.0006224497940703973) \exp(4i\pi (z/D - 1.5)/5) - \\
&- (0.00026859132889884717 + 0.00022776641240943203) \exp(i\pi (z/D - 1.5)) - \\
&- (0.00011732437486903 25 - 0.00008237663640946321) \exp(6i\pi (z/D - 1.5)/5) + \\
&+ (0.00018899136099786 98 - 0.00031141333068954126) \exp(7i\pi (z/D - 1.5)/5) + \\
&+ (0.00027288464436715 41 - 0.00005949260997935 2705) \exp(8i\pi (z/D - 1.5)/5) - \\
&- (0.00033648323544147 26 - 0.00041226203805249685i) \exp(9i\pi (z/D - 1.5)/5) - \\
&- (0.00029918431705723 25 - 0.00005038314716731472i) \exp(2i\pi (z/D - 1.5)) - \\
&- (0.00005139041745753 6605 + 0.00033089255447556 27i) \exp(11i\pi (z/D - 1.5)/5) + \\
&+ (0.00024643987127510484 - 0.00003943949222229763) \exp(12i\pi (z/D - 1.5)/5) + \\
&+ (0.00006083581704988 56 + 0.00022395108511517544) \exp(13i\pi (z/D - 1.5)/5) - \\
&- (0.00017389809467850 33 - 0.00003741257064118092) \exp(14i\pi (z/D - 1.5)/5) - \\
&- (0.00004374914482826 8773 + 0.00013687173427145811) \exp(3i\pi (z/D - 1.5)) + \\
&+ (0.00011030296357655545 - 0.00003154514287981986i) \exp(16i\pi (z/D - 1.5)/5) + \\
&+ (0.00002847829794567166 + 0.00008017779530694296) \exp(17i\pi (z/D - 1.5)/5) + \\
&+ (0.00006353082276061137 - 0.00002310667484072 11) \exp(18i\pi (z/D - 1.5)/5) + \\
&+ (0.00001807792442986 4712 + 0.00004593257538937565) \exp(19i\pi (z/D - 1.5)/5) +
\end{aligned}
$$

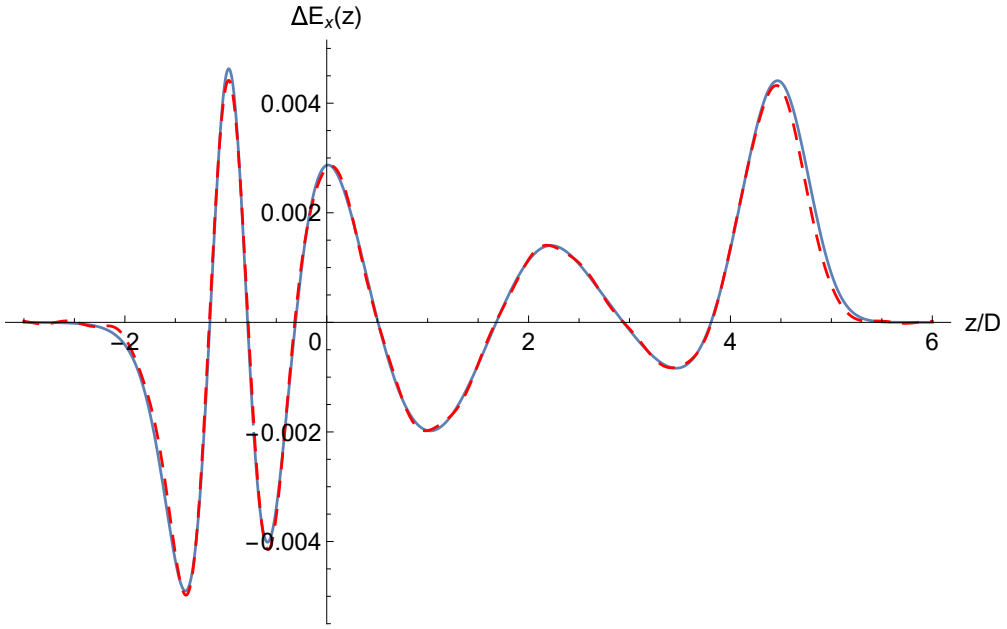


Figure 2.47: Function $H_3(z)$ (dashed red) is a Fourier expansion of $E_x(z) - H_2(z)$ (solid blue) over the interval $-3.5 \leq z/D \leq 6.5$. The origin $z = 0$ is at the EFB_{20}.

$$+ (0.000034592190696502 58 - 0.000015478461970141045i) \exp(4i\pi(z/D - 1.5)).$$

Fig. 2.47 illustrates that the function $H_3(z)$ approximates $E_x(z) - H_2(z)$ well.

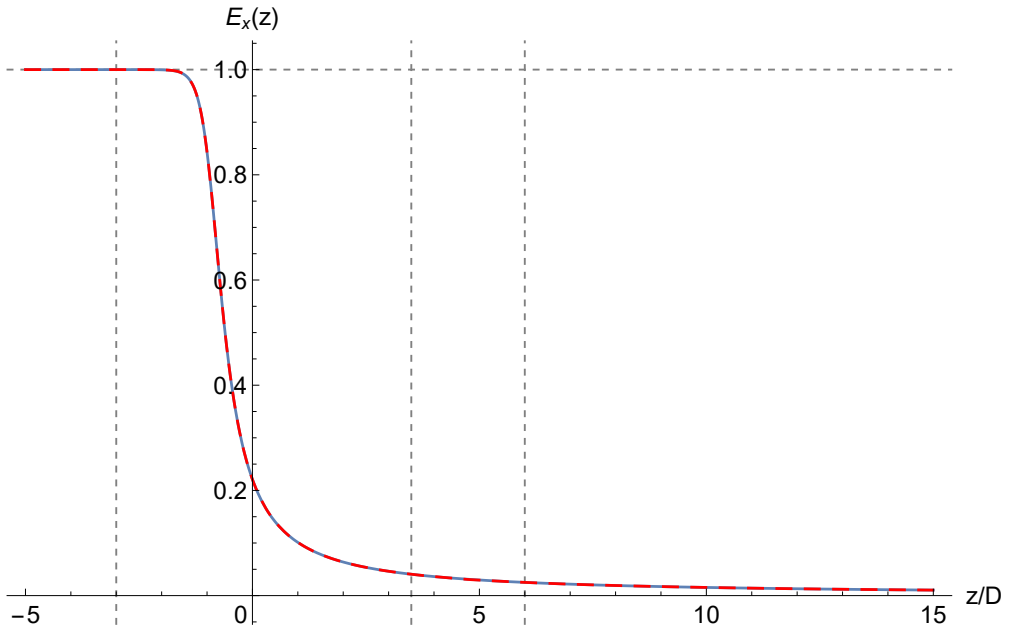
Finally, we obtained a function $G(z)$, a sum of $H_2(z)$ and $H_3(z)$ with smooth boundaries of $H_3(z)$ at $z/D = -3$ and $z/D = 6$, as

$$G(z) = H_2(z) + F_3(z) \frac{1}{1 + \exp\left[-\left(\frac{z}{D} + 3\right)^2\right]} \frac{1}{1 + \exp\left[\left(\frac{z}{D} - 3\right)^2\right]}.$$

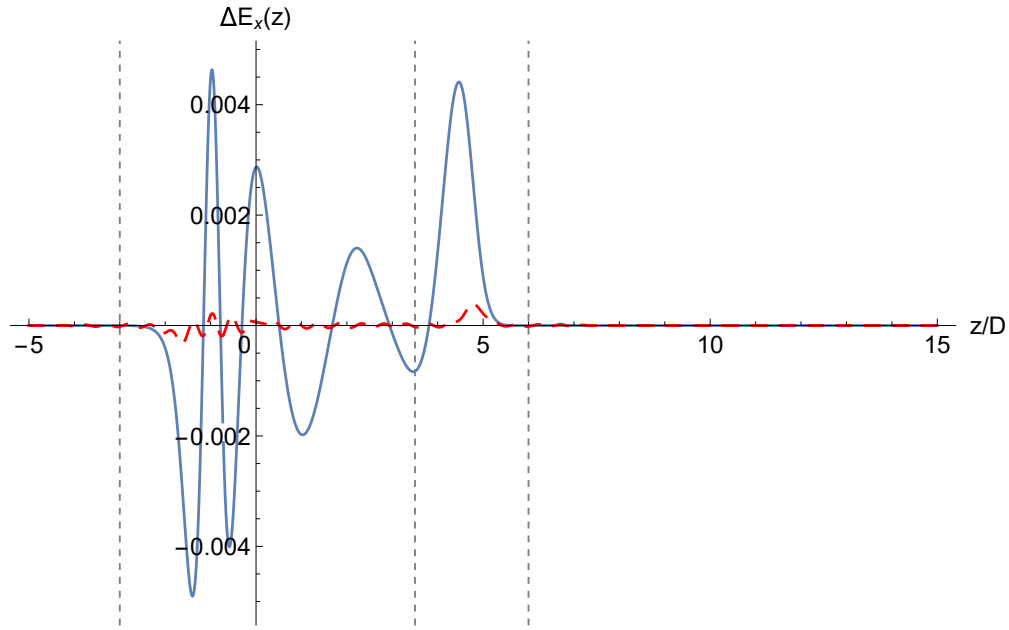
As Fig. 2.48 shows, $G(z)$ is a good approximation of $E_x(z)$ over at least the interval $-5 \leq z/D \leq 15$, with $\sup_{-5 \leq z/D \leq 15} |G(z) - E_x(z)|$ of about 4×10^{-4}.

2.7 Conclusion

We modeled fringe fields of semi-infinite electrostatic capacitors with infinitely thin, infinitely thick, and finitely thick plates, including plates with rounded edges. Additionally, we modeled fringe fields of two adjacent semi-infinite capacitors with finitely thick plates and symmetric, antisymmetric, and different voltages. We used conformal mapping methods to perform this modeling.



(a) $G(z)$ (dashed red) and $E_x(z)$ (solid blue).



(b) $G(z) - E_x(z)$ (dashed red) and $H_2(z) - E_x(z)$ (solid blue). $\sup_{-5 \le z/D \le 15} |G(z) - E_x(z)|$ is about 4×10^{-4}.

Figure 2.48: The piecewise function $G(z)$, an improved approximation of electrostatic field falloff $E_x(z)$ over function $H_2(z)$. Vertical gridlines at $z/D = -3$, 3.5, and 6 are points of smooth gluing or smooth boundaries used in construction of $G(z)$, as described in the text. The origin $z = 0$ is at the EFB_{20}.

Comparison of the results for fringe fields of semi-infinite capacitors with fringe fields of several finite rectangular electrostatic capacitors obtained using a BEM field solver shows good agreement.

We found that the field falloff of an electrostatic deflector is slower than exponential. Enge functions are not suitable for accurate modeling of such falloffs. We proposed an alternative function to model field falloffs of electrostatic deflectors..

CHAPTER 3

MAIN AND FRINGE FIELDS OF THE MUON g-2 STORAGE RING HIGH VOLTAGE QUADRUPOLE

3.1 Introduction

We studied the main and fringe fields of the Muon g-2 storage ring high voltage quadrupole (or the Muon g-2 collaboration quadrupole) described in [93]. The Muon g-2 storage ring is the lattice that was used for the E821 MDM measurement experiment and will be used, with modifications, for the E989 MDM measurement experiment.

In this chapter, we present a method to calculate the 2D main field of the quadrupole, accounting for geometric asymmetries and mispowered plates (e.g., for RF scraping).

Conformal mapping methods provide an analytic formula for a fully Maxwellian map from a canonical domain such as a rectangle to a polygon.

The cross section of the quadrupole is represented by a polygon. We obtained an accurate and fully Maxwellian approximation of the electrostatic potential for this 2D geometry using the following general method (or its adaptation):

1. Calculate the electrostatic potential using conformal mapping methods for one plate at 1 V and the other plates, the rectangular enclosure, and the trolley rails at 0 V.

2. Apply appropriate rotations to the resulting solutions for the potential, scale those solutions (e.g., by $\pm 2.4 \times 10^4$ or with mispowered values), and use their superposition.

This method allows rapid recalculations with adjustments to the geometry and the individual plate voltages.

The multipole terms of the electrostatic potential may be calculated (1) using Fourier analysis or (2) by taking the DA value of the potential. We calculated the multipole terms of the Muon g-2 collaboration quadrupole using these methods and performed an error analysis.

Moreover, we developed a code to automatically generate a 3D model in STEP[1] file format from a cross-sectional geometry represented by a polygon. We prepared a 3D model of the Muon g-2 collaboration quadrupole using this code, and the electrostatic potential data ("Soltner–Valetov data") for a respective grid-point set was calculated using *COULOMB*'s BEM field solver by Helmut Soltner (Forschungszentrum Jülich).

We obtained an approximation of the multipole strengths by calculating Fourier modes of the electrostatic potential at a set of radii in the transversal plane. From this data, we obtained the falloff of the quadrupole strength and the EFB.

We applied the same method for calculating multipole strengths to the electrostatic field data for the Muon g-2 collaboration quadrupole obtained using *Opera-3d*'s finite element method (FEM) field solver [1] by Wanwei Wu (FNAL and University of Mississippi) [116]. The quadrupole strength falloffs and the EFBs obtained from Soltner–Valetov and Wu field data agree well, and so do the tunes calculated based on them.

3.2 Main Field

3.2.1 Conformal Mapping Methods

In this section, we use concepts and notation and refer to software introduced in App. A Conformal Mapping Methods.

3.2.2 Conformal Mappings for Main Field Calculation

To calculate the electrostatic potential for the cross section of the quadrupole using the superposition method we outlined above, we consider each plate with the constant Dirichlet boundary condition 1 V and the other plates, the rectangular enclosure, and the trolley rails with the constant Dirichlet boundary condition 0 V. These two constant Dirichlet boundary conditions are interposed by two von Neumann boundary conditions, forming a logical quadrilateral. A conformal mapping from a

[1] STEP is an abbreviation for STandard for the Exchange of Product model data – ISO 10303-242 [55].

rectangular part of a bi-infinite strip is suitable for a physical domain as this logical quadrilateral, as App. A notes.

The derivative of the conformal mapping $f : \mathbb{C} \to \mathbb{C}$, $v \mapsto w$ from a rectangle is

$$f'(v) = c \operatorname{cn}(v|m) \operatorname{dn}(v|m) \prod_{j=1}^{n} \left(\operatorname{sn}(v|m) - \operatorname{sn}\left(x_j + i y_j | m\right) \right)^{\alpha_j - 1}, \tag{3.1}$$

where n is the number of polygon vertices, $x_j + i y_j = v_j$ are the prevertices $f^{-1}\left(w_j\right)$ of the polygon vertices w_j, $\pi\alpha_j$ are the interior polygon angles, m is the elliptic parameter, and c is a constant [30, p. 49].

For computational efficiency and to implicitly account for the singularity at $iK' = i\,\mathrm{K}(1-m)$, a conformal mapping f from a rectangle and its derivative f' are often calculated using a composite map $f = f_2 \circ f_1$ where $f_1(v) = \log(\operatorname{sn}(v|m))/\pi$ is the conformal mapping from a rectangle to the strip $0 \leq \Im(z) \leq 1$ and

$$f_2(z) = A + C \int^{z} \exp\left[\frac{\pi}{2}(\alpha_- - \alpha_+)\zeta\right] \prod_{k=1}^{n} \left[\sinh \frac{\pi}{2}(\zeta - z_k)^{\alpha_k - 1}\right] d\zeta$$

is the conformal mapping from the strip $0 \leq \Im(z) \leq 1$ to the interior of a polygon. Points z_k are the prevertices of the polygon other than the complex infinity at the two sides of the infinite strip, angles $\pi\alpha_k$ are the respective interior angles of the polygon, and $\pi\alpha_\pm$ are the divergence angles of the sides of the strip at the ends of the strip [30, p. 46].

We note that

$$f_1'(v) = \frac{\operatorname{cn}(v|m) \operatorname{dn}(v|m)}{\pi \operatorname{sn}(v|m)}.$$

The following is an outline of a calculation process of the prevertices v_j in a rectangular canonical domain, as performed in the *SC Toolbox* [28, 30].

1. The side-length ratios and the prevertices z_j on the sides of the infinite strip $0 \leq \Im(z) \leq 1$ are obtained using a quasi-Newton method, or a modular nonlinear equation solver such as Algorithm D6.1.3 `NEDRIVER` in [24, p. 285], from the $n-3$ real side-length conditions that arise from the integrals of f' along the polygon side preimages between consecutive

prevertices:

$$\int_{z_j}^{z_{j+1}} f'(\zeta)\, d\zeta = w_{j+1} - w_j, \quad j = 1, 3, \ldots, n-2,$$

where the initial guess for the prevertices z_j is made by placing points equidistantly on the sides of the infinite strip $0 \leq \Im(z) \leq 1$ matching the Dirichlet boundary conditions.

2. The corners of the rectangle are determined in the form

$$\left(K, K + iK', -K + iK', -K\right)$$

from the side-length ratios.

3. The prevertices v_j are obtained as preimages of z_j under the conformal mapping from the rectangle to the infinite strip

$$f_1(v) = \frac{1}{\pi} \log\left(\operatorname{sn}(v|m)\right),$$

where the initial guess for the prevertices v_j is made by placing points equidistantly on the sides of the rectangle matching the Dirichlet boundary conditions.

3.2.3 Multipole Terms

Consider an element with the electrostatic potential $\varphi(s, r, \theta)$, where s is the longitudinal and (r, θ) are the polar transversal particle optical coordinates. We perform a Taylor series expansion of φ in r, followed by a Fourier series expansion of the result in θ [11, pp. 120–125]. The resulting Taylor–Fourier series expansion

$$\varphi(s, r, \theta) = \sum_{k=0}^{+\infty} \sum_{l=0}^{+\infty} M_{k,l}(s) \cos\left(l\theta + \theta_{k,l}\right) r^k$$

is the **multipole expansion** of the potential[2]. The term $M_{l,l}(s)$ is called the **multipole strength** of order l, and $\theta_{l,l}$ is called the **phase**.

[2]Whenever r is not scaled in a formula explicitly, e.g., to a radius R relative to a reference radius R_{ref}, it is assumed to be implicitly scaled.

As [11, pp. 120–125] shows, the terms $M_{k,l}(s)$ with (k,l) other than $k = l, l+2, l+4, \ldots$ vanish. For $n \in \mathbb{N}$,

$$M_{l+2n,l}(s) = \frac{M_{l,l}^{(2n)}(s)}{\prod_{\nu=1}^{n}\left(l^2 - (l+2\nu)^2\right)}.$$

If $M_{l,l}(s)$ is independent of s, the terms $M_{k,l}(s)$ vanish for $k > l$. For an s-independent electrostatic potential φ, the multipole expansion has the form [11, pp. 120–125]

$$\varphi(r,\theta) = \sum_{l=0}^{+\infty} M_{l,l} \cos\left(l\theta + \theta_{l,l}\right) r^l.$$

3.2.3.1 Fourier Analysis

To obtain the multipole terms for an s-independent main field, it is sufficient to obtain the Fourier coefficients of the electrostatic potential at one radius R_{ref}, which is called the reference radius. The reference radius R_{ref} is normally chosen as large as possible for the numerical computation to most accurately distinguish and determine the multipole terms, but less than the aperture. The numerical solution for the electrostatic potential often has inaccuracies excessively close to the aperture, e.g., because of the nearby singularities in the ODE or the approach to the convergence radius of the Taylor series expansion.

If the Fourier series expansion at the reference radius R_{ref} is

$$\varphi(r,\theta) = \frac{a_{0R_{\text{ref}}}}{2} + \sum_{j=1}^{+\infty} r^j \left(a_{jR_{\text{ref}}} \cos(j\theta) + b_{jR_{\text{ref}}} \sin(j\theta)\right),$$

the multipole expansion is then

$$\varphi(r,\theta) = \frac{a_{0R_{\text{ref}}}}{2} + \sum_{j=1}^{+\infty} \left(r\frac{R}{R_{\text{ref}}}\right)^j \left(a_{jR_{\text{ref}}} \cos(j\theta) + b_{jR_{\text{ref}}} \sin(j\theta)\right), \tag{3.2}$$

where R is the scaling radius.

Denoting

$$A_j = a_{jR_{\text{ref}}} \left(\frac{R}{R_{\text{ref}}}\right)^j, \quad B_j = b_{jR_{\text{ref}}} \left(\frac{R}{R_{\text{ref}}}\right)^j$$

in eq. 3.2, we obtain

$$\varphi(r,\theta) = \frac{A_0}{2} + \sum_{j=1}^{+\infty} r^j \left(A_j \cos(j\theta) + B_j \sin(j\theta)\right). \tag{3.3}$$

In the cross section of an element with an s-independent main field, the aperture radius[3] a is also a dimensional scale (δr) of the radius. The scaling radius R is often chosen at or near the aperture[4].

3.2.3.2 Differential Algebra Methods

One of the advantages of DA computations is that Taylor series expansions can be produced automatically for a highly complex analytic function by substituting its comprising functions with their DA values. Another advantage is the DA fixed-point algorithms, e.g., for the inversion of a Taylor series expansion, that complete in finitely many steps [11, pp. 96–102].

Knowing the analytic expression for f' and the constant part (the scalar value)

$$g_0 = \mathrm{cons}\,(g\,(0)) \in \mathbb{C}$$

of the DA value of $g = f^{-1}$ at the origin $w = 0$, we can obtain the DA value[5] of $f'\,(v)$ at $v = g_0$. Then the DA value of $f\,(g_0)$ is

$$f\,(g_0) = 0 + \partial^{-1} f'\,(g_0)\,,$$

where ∂^{-1} is the antiderivation operator, which is the DA equivalent of taking the integral $\int_0^z p\,(t)\,dt$ of a Taylor expansion $p\,(z)$.

The DA value of $g\,(0)$ is then obtained by taking the DA inverse of $\partial^{-1} f'\,(g_0)$ and adding the constant part:

$$g\,(0) = g_0 + \left(\partial^{-1} f'\,(g_0)\right)^{-1}.$$

Now we obtain the DA value of the electrostatic potential φ at the origin as

$$\varphi\,(0) = \psi \circ g\,(0)\,,$$

[3]The aperture is the radius of the largest circle with center at the origin of the local Frenet–Serret frame that can be inscribed in the cross section of the element. Thus, the aperture is also the radius of convergence of the Taylor expansion of the electrostatic potential at the origin of the complex plane.

[4]The reasons for scaling are twofold: (1) scaling gives practical advantages in some calculations, such as reduction of floating-point errors; and (2) scaling applies the notions of magnitude and scale, which are universal in physics and are useful, e.g., for phenomenological and asymptotic analysis.

[5]The appropriate DA algebra here is ${}_nD_v$, where n is the desired order of computation and v is the number of variables $v \geq 1$.

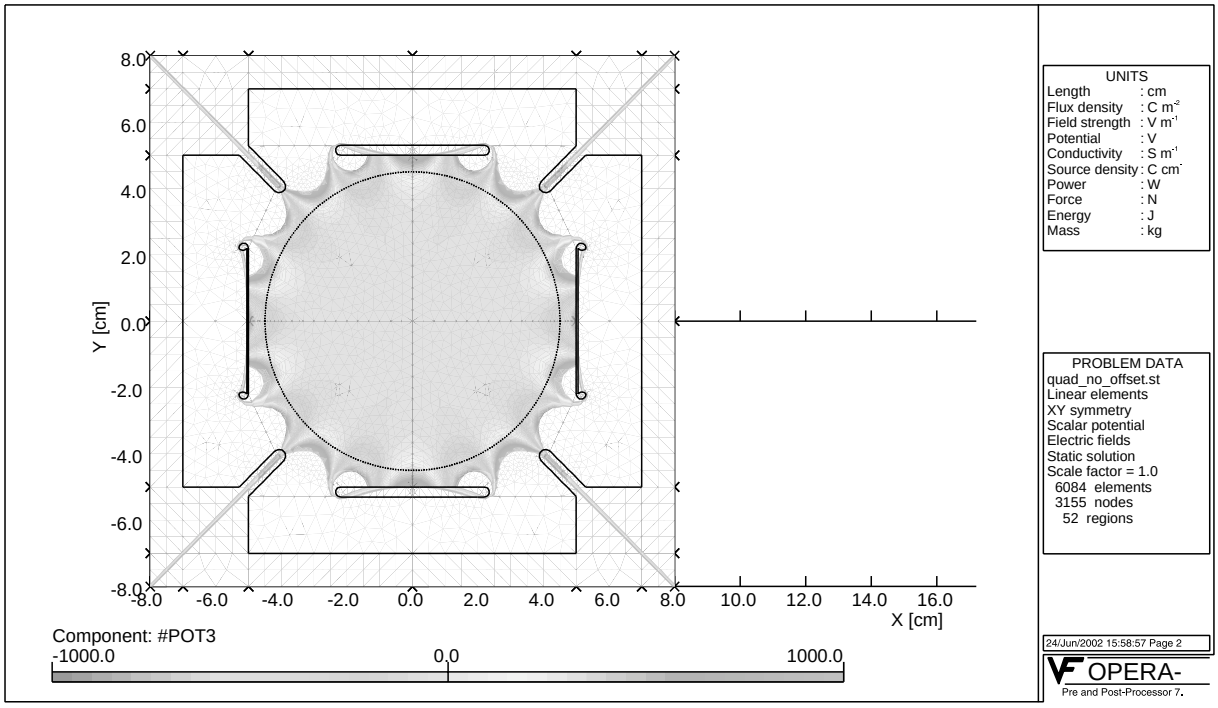


Figure 3.1: Cross section of the Muon g-2 collaboration quadrupole. (Reprinted from *Nucl. Instr. Meth. Phys. Res. A*, Vol. 503, Yannis K. Semertzidis *et al.*, The Brookhaven Muon $(g\text{-}2)$ Storage Ring High Voltage Quadrupoles, Page No. 476, Copyright 2003, with permission from Elsevier.)

where ψ is the solution of the Laplace equation for the electrostatic field in the canonical domain.

We obtain the Taylor series expansion of φ from the DA value of $\varphi(0)$ as

$$\varphi(w) = \sum_{j=0}^{n} c_k w^k + o(w^n).$$

To obtain the multipole coefficients in the trigonometric form, we take $a_0 = 2c_0$, $a_j = \Re\left(c_j\right)$, and $b_j = \Re\left(ic_j\right)$, where $j = 1,2,\ldots,N$.

We may now apply a scaling radius R relative to a reference radius R_{ref}. If scaling was not performed prior to these calculations, the reference radius is $R_{\text{ref}} = 1$ in the units used for the calculations.

3.2.4 Calculations and Results

Fig. 3.1 shows the cross section of the Muon g-2 collaboration quadrupole.

First, we considered the case with non-mispowered plates and no geometric asymmetries. We solved the problem for one-eighth of the cross section and extended the solution to the entire cross section, utilizing the four-fold rotational symmetry and the four mirror symmetries.

Next, we considered the general case of mispowered plates and geometric asymmetries. In that case, we approximated the cross-sectional geometry by a simply-connected region using connecting rods between the plates and the rectangular enclosure. Having calculated the electrostatic potential using conformal mapping methods with one plate at 1 V and the other plates and the rectangular enclosure at 0 V, we applied geometrical perturbations, rotations, and scaling to four copies of the potential and superimposed them to obtain the solution for the potential.

All multipole expansions presented here are up to the order 24. The unit of length in the physical domain is centimeters.

Unless otherwise specified, we used *MATLAB* R2016b, the *SC Toolbox* 2.3, *COSY INFINITY* 9.1 [68], and *Mathematica* 11 for calculations.

We enumerated the quadrupole plates, as well as their voltage coefficients V_j and errors D_j in the distance from the origin, where $j = 1,2,3,4$, in the counter-clockwise direction starting from the left plate on the cross section drawing.

3.2.4.1 Nominal Symmetric Case ("SM")

The four-fold rotational symmetry and the four mirror symmetries (with accounting for the alternating sign of the voltages) in the case of non-mispowered plates and symmetrical geometry allow us to reduce the problem of finding the electrostatic potential in the entire cross section of the quadrupole to obtaining it only in one-eighth of the geometry. This one-eighth part of the cross-sectional geometry is shown in Fig. 3.2. We denote this nominal symmetric case as "SM".

The *MATLAB* function listed in App. C.1 returns the conformal mapping object `f` for a conformal mapping from a rectangle to the interior of the polygon object `p` representing the 2D geometry.

Entering $[\mathtt{p},\mathtt{f}] = \mathtt{initquad4}$ in the *MATLAB* command line yields the prevertices and the corners of the rectangular canonical domain:

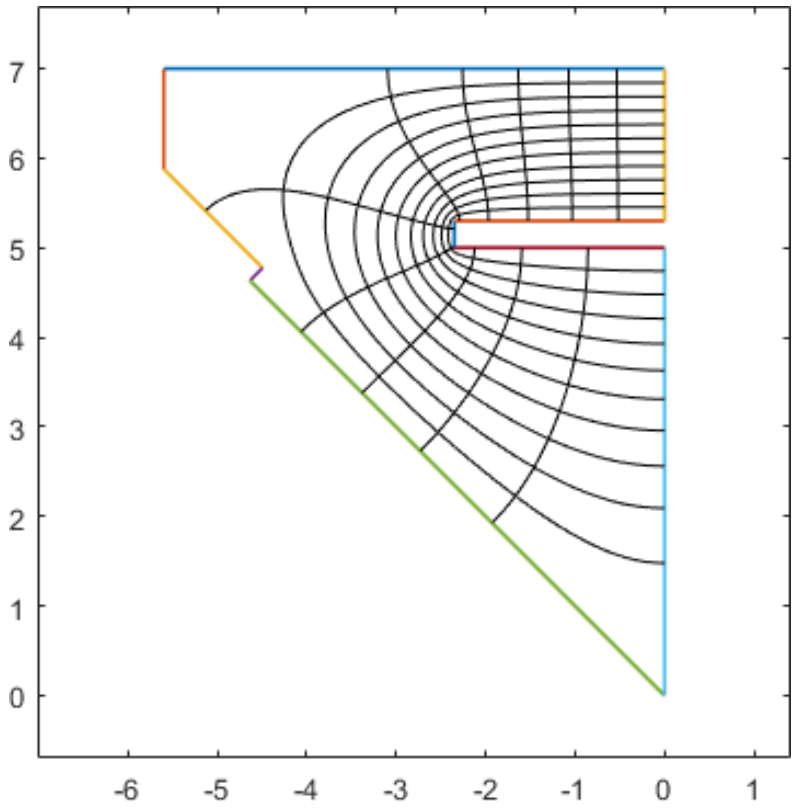


Figure 3.2: Model of one-eighth of the cross section of the Muon g-2 collaboration quadrupole for the case of non-mispowered plates and no geometric asymmetries ("SM").

rectmap object:

cnr	vertex	alpha	prevertex
1	0.00000 + 7.00000i	0.50000	1.57079633e+00
	−5.59792 + 7.00000i	0.50000	1.57079633e+00 + 5.68557125e+00i
	−5.59792 + 5.87990i	0.75000	1.57079633e+00 + 5.74755826e+00i
	−4.49396 + 4.77594i	1.50000	1.57079633e+00 + 6.26976031e+00i
	−4.63495 + 4.63495i	0.50000	1.57079633e+00 + 6.35334334e+00i
2	0.00000 + 0.00000i	0.25000	1.57079633e+00 + 1.07013887e+01i
3	0.00000 + 5.00000i	0.50000	−1.57079633e+00 + 1.07013887e+01i
	−2.35000 + 5.00000i	1.50000	−1.57079633e+00 + 6.91809897e+00i
	−2.35000 + 5.30000i	1.50000	−1.57079633e+00 + 5.27565570e+00i
4	0.00000 + 5.30000i	0.50000	−1.57079633e+00

```
c = -3.9440549e-15 - 64.411305i
Conformal modulus = 3.4063578

Apparent accuracy is 6.40e-14
```

A heatmap plot of the electrostatic potential as the inverse of the conformal mapping object `f` is shown in Fig. 3.3.

Now we apply the list of prevertices, the list of α's, the scaling constant c, and the parameter m to the analytic formula for the derivative f' of the conformal mapping in eq. 3.1. Note that the scaling constant c in eq. 3.1 is different from the constant `c` in the *MATLAB* output. The constant c and the parameter m are obtained as described in App. A.

The resulting derivative $f' : \mathbb{C} \to \mathbb{C}$, $v \mapsto w$ of the conformal mapping from the rectangular canonical domain to the physical domain is

$$f'(v) = c\,\mathrm{cn}(v|m)\,\mathrm{dn}(v|m)\prod_{j=1}^{n}\left(\mathrm{sn}(v|m) - \mathrm{sn}\left(x_j + i y_j|m\right)\right)^{\alpha_j - 1},$$

where n is the number of polygon vertices. The scaling constant is $c = -i\,12429.568123663117$, and the parameter m is $m = 8.1098319813220654 \times 10^{-9}$. The other parameters are as follows:

$$\alpha = (0.5, 0.5, 0.75, 1.5, 0.5, 0.25, 0.5, 1.5, 1.5, 0.5),$$

$$x = \left(\underbrace{\mathrm{K}(m), \ldots, \mathrm{K}(m)}_{6\text{ times}}, \underbrace{-\mathrm{K}(m), \ldots, -\mathrm{K}(m)}_{4\text{ times}}\right),$$

and

$$\begin{aligned} y = (&0, 5.6855712537881429, 5.7475582573144708,\\ &6.2697603145320198, 6.3533433441730081, 10.701388724040664,\\ &10.701388724040664, 6.9180989665683397, 5.2756556972892490, 0). \end{aligned}$$

The preimage v_0 of the origin $w = 0$ is

$$v_0 = g(0) = 1.570796329979620 + i\,10.70138872404065.$$

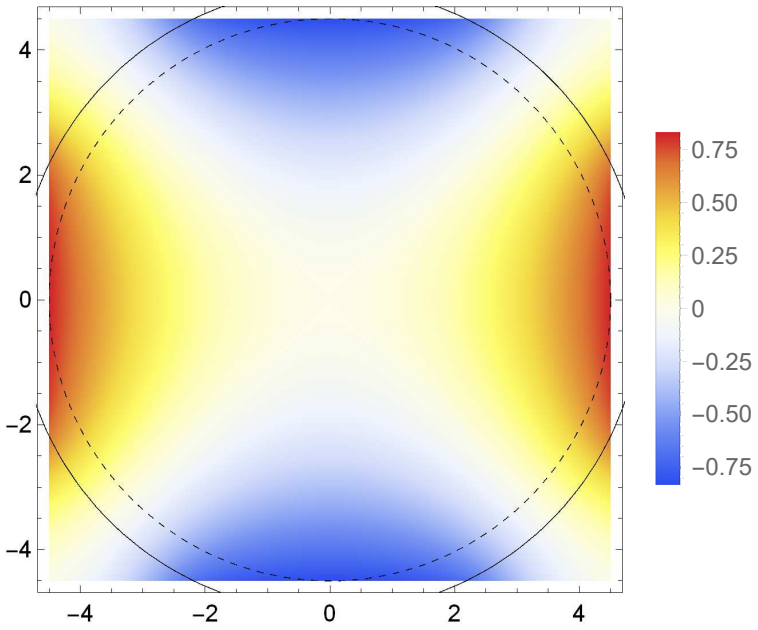


Figure 3.3: "SM *MATLAB*": heatmap plot of the electrostatic potential as the inverse of the conformal mapping object `f`.

The derivative f' has a branch point at $v = v_0$, which presents certain difficulties in the analysis. For example, it is not possible to obtain the electrostatic potential multipole terms by obtaining f via a Taylor series expansion of f' and then calculating the inverse series. The same applies to the calculation of DA values of f at point $v = v_0$.

3.2.4.1.1 Differential Equation Solution in *Mathematica* ("*Mathematica* PDE")

First, we obtained the electrostatic potential φ on the vertical edge of the polygon, shown in Fig. 3.4, in *Mathematica* by solving the restriction of the ODE

$$\frac{dg(w)}{dw} = \frac{1}{f'(g(w))}, \quad g(w_0) = z_0$$

(see App. A) to the respective one-dimensional case, where a point w_0 was chosen on the vertical edge.

Next, we solved the Cauchy–Riemann PDE in *Mathematica*, with the boundary conditions as found in the previous step on the vertical edge and as zero on the diagonal edge of the polygon. The result is shown in Figs. 3.5 and 3.6.

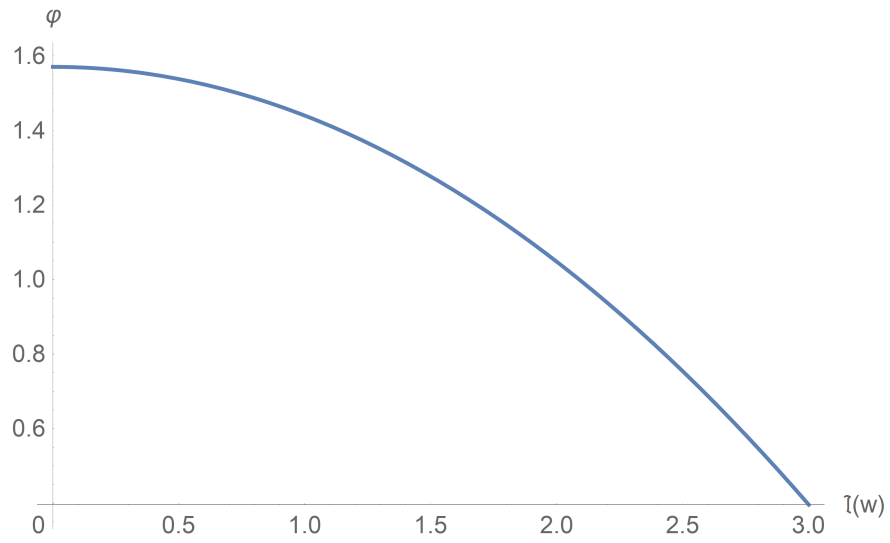


Figure 3.4: "SM *Mathematica* PDE": vertical boundary condition of the Laplace equation for the electrostatic potential φ.

Finally, we calculated the Fourier modes of φ at the reference radii

$$R_{\text{ref}} = 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5\,\text{cm}$$

by taking numerical integrals. For each radius, we formulated the multipole terms in accordance to eq. 3.3.

3.2.4.1.2 Fourier Analysis in *MATLAB* ("*MATLAB* Fourier")

For the conformal mapping object `f` and a reference radius `R`, the *MATLAB* function listed in App. D.1 calculates the discrete Fourier transform (DFT) d_n and the Fourier modes a_0, a_n, b_n of the electrostatic potential φ using a fast Fourier transform (FFT).

To that end, the *MATLAB* function performs an equidistant discretization of the circle of radius `R`, centered at the origin $r = 0$, with $N = 1001$ arc intervals of length $\triangle t = 2\pi\mathtt{R}/N$. It computes the inverse values of the conformal mapping object `f` at that grid-point set:

$$u = \left(\mathtt{f}^{-1} \left(\mathtt{R}\cos\left(j\triangle t\right), \mathtt{f}^{-1}\left(\mathtt{R}\sin\left(j\triangle t\right)\right)\right)_{j=0}^{N-1} \right).$$

The DFT of the electrostatic potential φ around the circle of reference radius `R` is obtained as

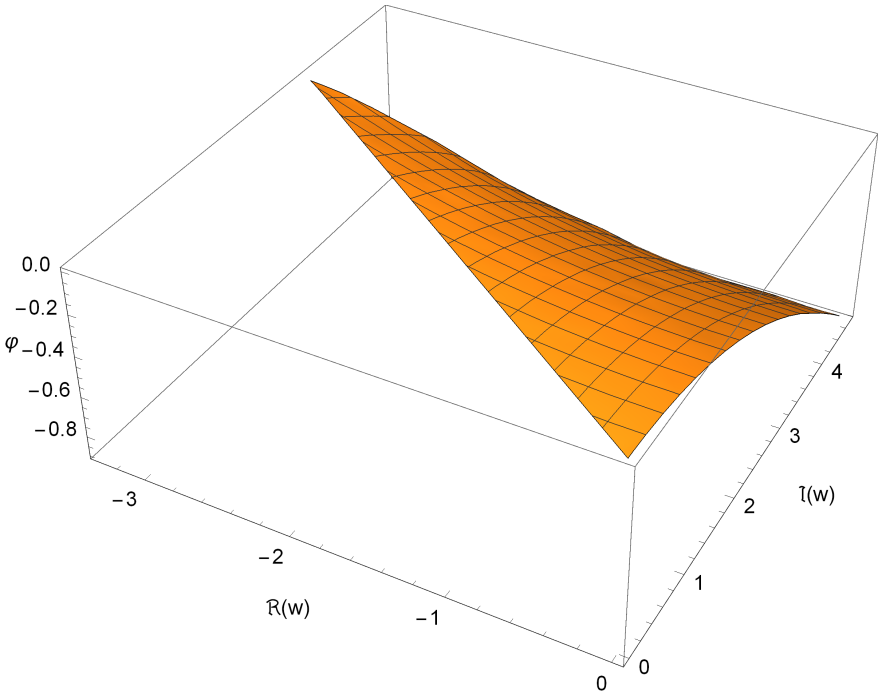


Figure 3.5: "SM *Mathematica* PDE": 3D plot of the solution of the Cauchy–Riemann PDE for the electrostatic potential φ.

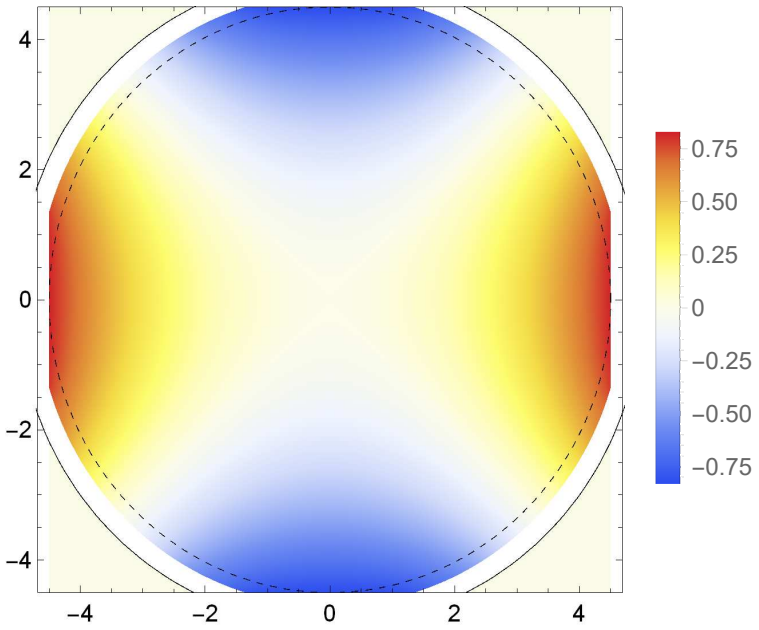


Figure 3.6: "SM *Mathematica* PDE": heatmap plot of the multipole expansion of the electrostatic potential, up to order 24.

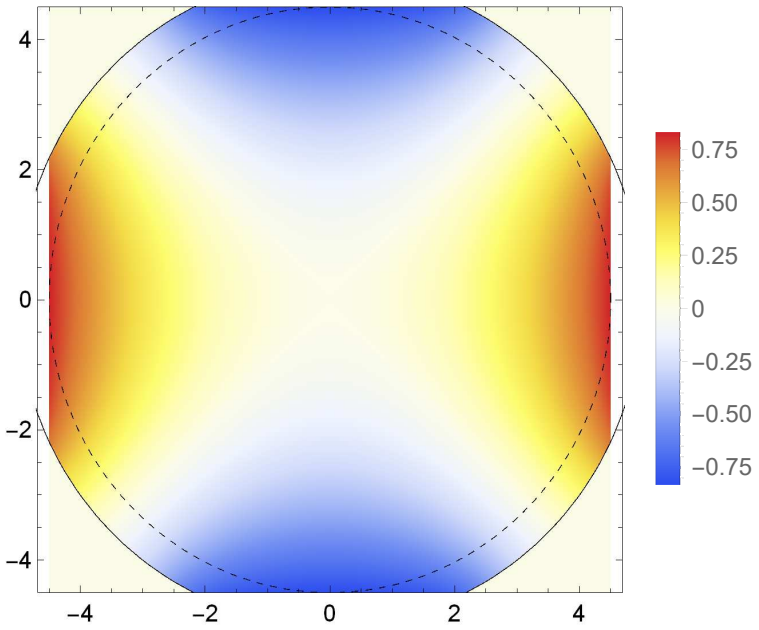


Figure 3.7: "SM *MATLAB* Fourier": heatmap plot of the multipole expansion of the electrostatic potential, up to order 24.

the FFT transform of u, with Fourier parameters $(a,b) = (1,-1)$, divided by N:

$$d_n = \frac{1}{N}\sum_{j=0}^{N-1} u_j \exp\left(-\frac{2\pi i}{N} jn\right).$$

The Fourier modes in trigonometric form are obtained from the DFT, utilizing the Hermitian symmetry [16, pp. 76–77], as

$$a_0 = 2d_0,$$

$$a_n = d_n + d_{N-n},$$

$$b_n = d_n - d_{N-n}.$$

From the Fourier modes calculated for each of the reference radii

$$R_{\text{ref}} = 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5\,\text{cm},$$

we formulated the multipole terms in accordance to eq. 3.3.

Figs. 3.7 and 3.8 show plots of the multipole expansion with $R_{\text{ref}} = 4.5\,\text{cm}$ starting from the zeroth and the third order, respectively.

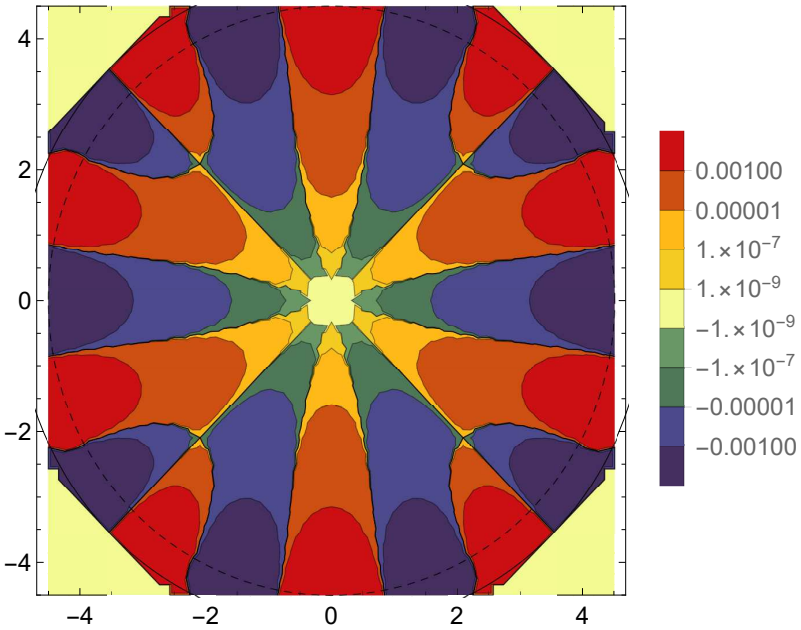


Figure 3.8: "SM *MATLAB* Fourier": contour plot of the multipole expansion of the electrostatic potential, orders 3 to 24.

3.2.4.2 Non-Symmetric Case ("NSM")

Regarding the general case of mispowered plates and geometric asymmetries, we cannot simplify the problem by reducing the physical domain, using the four-fold symmetry and the four mirror symmetries, as we did in the SM case. We denote this general non-symmetric case as "NSM".

The cross section of the quadrupole is represented by a multiply-connected region consisting of five simple polygons: the rectangular enclosure and the trolley rails as one simple polygon and each of the four plates as one simple polygon.

The conformal mapping theory for physical domains as n-connected regions for $n > 2$ is quite challenging (see, e.g., [22]), and, in practice, various approximations and assumptions are used. Doubly connected regions require specialized methods and present certain restrictions and issues [30, pp. 64–70].

We approximated the cross-sectional geometry by a simply-connected region using connecting rods between the rectangular enclosure and the four plates, as Fig. 3.9 shows. We placed the rods in the middle of the back side of each plate to minimize their impact on the multipole terms.

We consider geometric asymmetries as errors in the distances between the plates and the origin

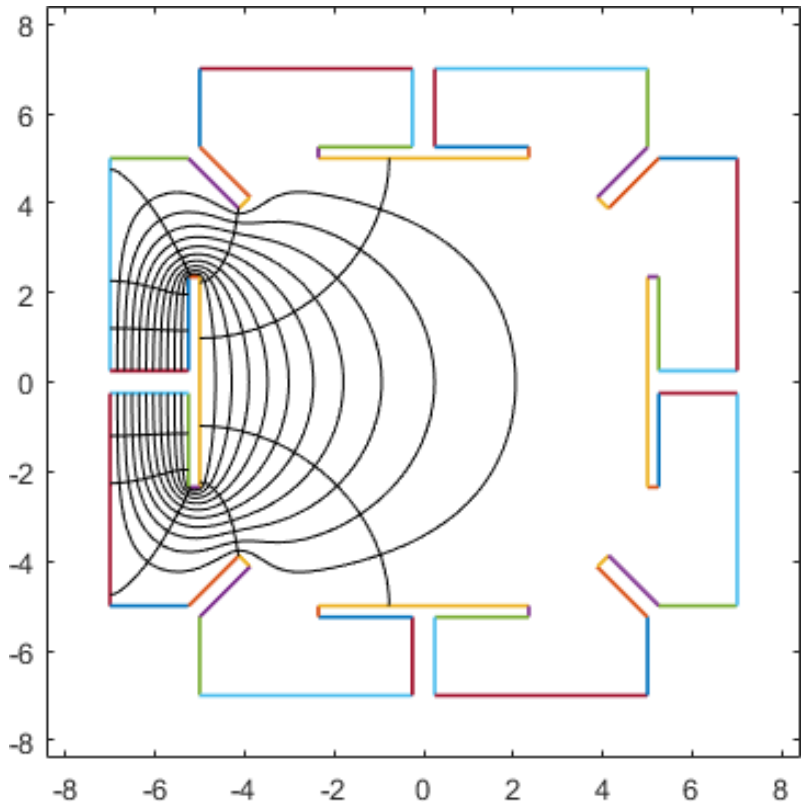


Figure 3.9: Model of the cross section of the Muon g-2 collaboration quadrupole for the case of mispowered plates and geometric asymmetries ("NSM").

in the cross section.

In principle, we could supply arbitrary plate distance errors as parameters in the polygonal model and compute the respective conformal mapping using the *SC Toolbox*. However, due to the crowding phenomenon [30, 6, 31, 7] illustrated by the distances between successive prevertices in Fig. 3.10 reaching $\sim 10^{-4}$, the optimization algorithm that obtains conformal mapping prevertices in the *SC Toolbox* is affected by numerical stability issues and often does not converge. Hence, we treat plate distance errors as perturbations and, for each plate distance error, use a two-point linear interpolation (or extrapolation, as applicable, which has the same formula in this case) based on the solution for the nominal geometry and one other solution where optimization had successfully converged.

Having prepared the interpolation data, we obtain the electrostatic potential in the NSM case using the following adaptation of the general method:

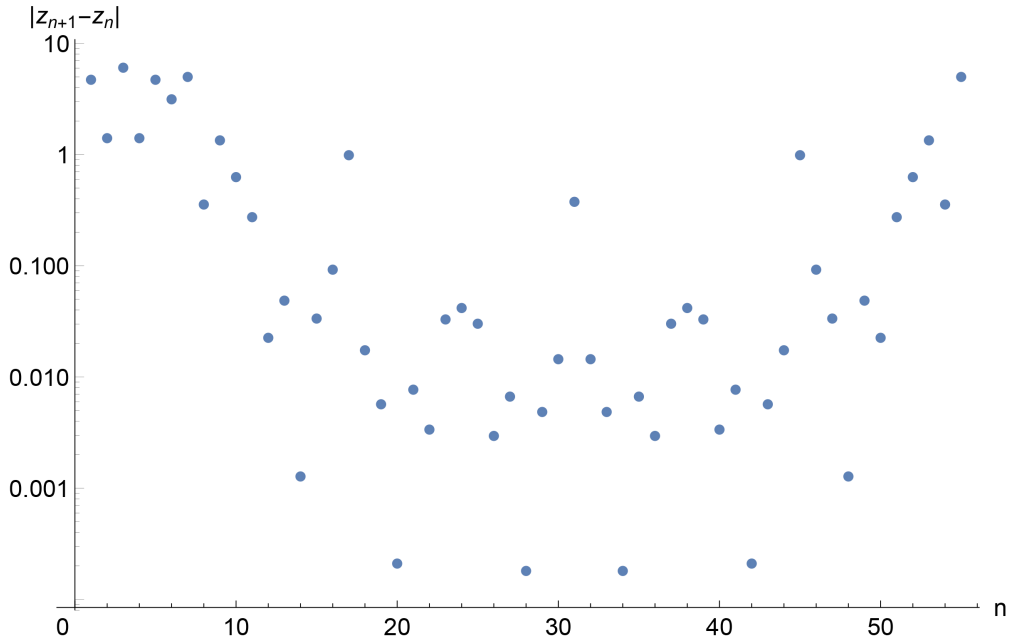


Figure 3.10: Distances between successive prevertices z_n of a conformal mapping from a rectangle to the interior of the polygon representing the cross-sectional geometry in the NSM case.

1. Calculate the electrostatic potential using conformal mapping methods with one plate – the left plate on the cross section drawing (as in Fig. 3.1) – at 1 V and the other Dirichlet boundary conditions (the remaining plates, the rectangular enclosure, and the trolley rails) of 0 V.

2. Apply plate distance errors as perturbations to four copies of the potential, each copy corresponding to one plate at 1 V and the other Dirichlet boundary conditions of 0 V.

3. Apply appropriate rotations to these four copies of the potential, scale the copies (e.g., by $\pm 2.4 \times 10^4$ or with mispowered values), and use their superposition.

To test NSM results against SM results, we used the identical configuration of non-mispowered plate voltages ($\pm 2.4 \times 10^4$ V) and no geometric asymmetries.

The *MATLAB* function listed in App. C.2 returns the conformal mapping object `f` for a conformal mapping from a rectangle to the interior of the polygon object `p` representing the 2D geometry.

We note that, of *MATLAB* releases R2009a, R2014b, R2016b, and R2017a, this calculation successfully completes only in releases R2016b and R2017a.

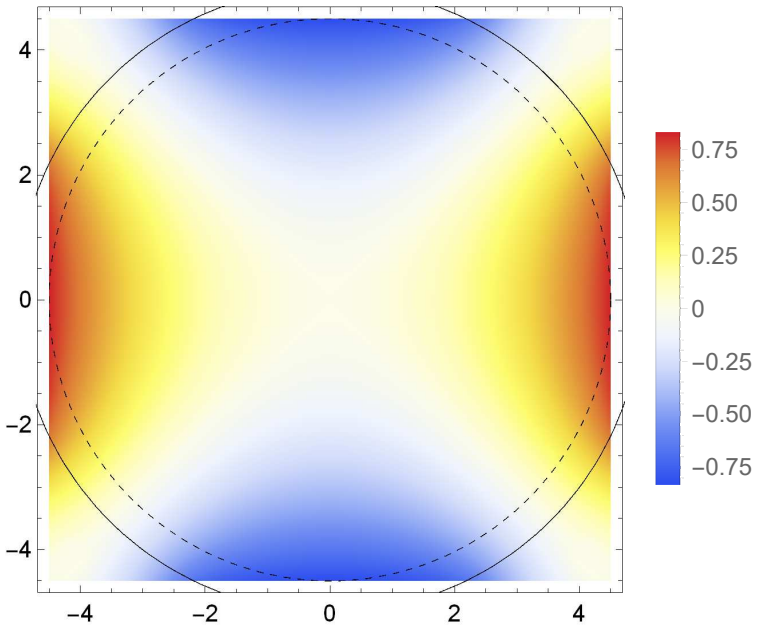


Figure 3.11: "NSM *MATLAB*": heatmap plot of the electrostatic potential as the inverse of the conformal mapping object `f`.

A heatmap plot of the electrostatic potential as the inverse of the conformal mapping object `f` is shown in Fig. 3.11.

Next, we apply the list of prevertices, the list of α's, and the parameter m to the analytic formula for the derivative f' of the conformal mapping in eq. 3.1. The scaling constant c in eq. 3.1 differs from the constant `c` in the *MATLAB* output. The constant c and the parameter m are obtained as described in App. A.

The resulting derivative $f' : \mathbb{C} \to \mathbb{C}$, $v \mapsto w$ of the conformal mapping from the rectangular canonical domain to the physical domain is

$$f'(v) = c\,\mathrm{cn}(v|m)\,\mathrm{dn}(v|m)\prod_{j=1}^{n}\left(\mathrm{sn}(v|m) - \mathrm{sn}\left(x_j + iy_j|m\right)\right)^{\alpha_j - 1},$$

where n is the number of polygon vertices. The scaling constant is $c = 1.1819869586349204 \times 10^{7}$, and the parameter m is $m = 2.1776166063953954 \times 10^{-15}$. The other parameters are as follows:

$$\alpha = (0.5, 1.5, 1.5, 1.5, 1.5, 0.5, 0.5, 0.5, 0.75, 1.5, 1.5, 0.75, 0.5, 0.5,$$
$$0.5, 1.5, 1.5, 1.5, 1.5, 0.5, 0.5, 0.5, 0.75, 1.5, 1.5, 0.75, 0.5, 0.5,$$
$$0.5, 1.5, 1.5, 1.5, 1.5, 0.5, 0.5, 0.5, 0.75, 1.5, 1.5, 0.75, 0.5, 0.5,$$

$$0.5, 1.5, 1.5, 1.5, 1.5, 0.5, 0.5, 0.5, 0.75, 1.5, 1.5, 0.75, 0.5, 0.5),$$

$$x = \left(\underbrace{\mathrm{K}(m), \ldots, \mathrm{K}(m)}_{6 \text{ times}}, \underbrace{-\mathrm{K}(m), \ldots, -\mathrm{K}(m)}_{50 \text{ times}} \right),$$

and

$$\begin{aligned} y = (&0, 4.707844817895257, 6.1123226415433312 \\ &12.154244427584317, 13.558722251232082, 18.266567069128161, \\ &18.266567069128161, 13.274266140422423, 12.918629029969397, \\ &11.576281153221727, 10.948991934692465, 10.675215453051434, \\ &10.65269458697405, 10.604072782442005, 10.6027963210765, \\ &10.569241422270052, 10.476916632115067, 9.4896639886962486, \\ &9.4722605135357565, 9.4665721145687325, 9.4663618193493928, \\ &9.4586650413212396, 9.4552969316026854, 9.4223471739893334, \\ &9.3806007759118621, 9.3504541968427848, 9.3474996905761945, \\ &9.3408322140301419, 9.3406516641648345, 9.3358001108809781, \\ &9.3213334522713787, 8.9452336168578217, 8.9307669582482223, \\ &8.9259154049643676, 8.9257348550990603, 8.9190673785530059, \\ &8.9161128722864156, 8.8859662932173382, 8.8442198951398652, \\ &8.8112701375265097, 8.8079020278079554, 8.8002052497797987, \\ &8.799994954560459, 8.794306555593435, 8.7769030804329375, \\ &7.7896504370136723, 7.6973256468586344, 7.6637707480521646, \\ &7.6624942866866617, 7.6138724821545702, 7.5913516160771657, \\ &7.3175751344358959, 6.6902859159059922, 5.347938039156598, \\ &4.9923009287031883, 0). \end{aligned}$$

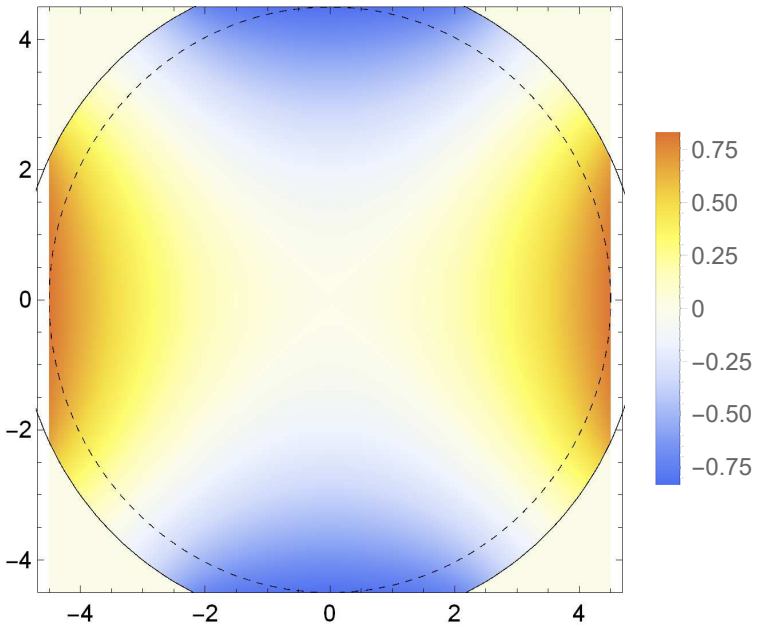


Figure 3.12: "NSM *MATLAB* Fourier": heatmap plot of the multipole expansion of the electrostatic potential, up to order 24.

The preimage v_0 of the origin $w = 0$ is

$$v_0 = g(0) = -0.9486801376613026 + i\,9.133283534564583.$$

3.2.4.2.1 Fourier Analysis in *MATLAB* ("*MATLAB* Fourier")

For the conformal mapping object `f` and a reference radius `R`, the *MATLAB* function listed in App. D.2 calculates the DFT d_n and the Fourier modes a_0, a_n, b_n of the electrostatic potential φ using an FFT.

From the Fourier modes calculated for each of the reference radii

$$R_{\text{ref}} = 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5\,\text{cm},$$

we formulated the multipole terms in accordance to eq. 3.2.

Figs. 3.12 and 3.13 show plots of the multipole expansion with $R_{\text{ref}} = 4.5\,\text{cm}$ starting from the zeroth and the third order, respectively.

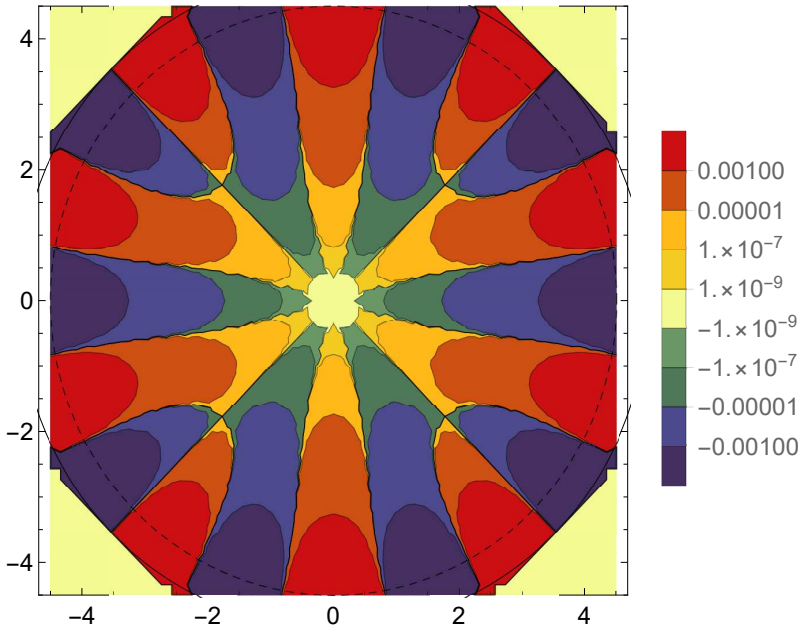


Figure 3.13: "NSM *MATLAB* Fourier": contour plot of the multipole expansion of the electrostatic potential, orders 3 to 24.

3.2.4.2.2 DA Calculation in *Mathematica* ("*Mathematica* DA")

Mathematica has a DA implementation, where DA values are encoded in `SeriesData` objects. Mathematical operations are defined for `SeriesData` objects and produce results as `SeriesData` objects.

We developed a *Mathematica* notebook that, for the conformal mapping derivative f in the analytic form, calculates the multipole terms a_0, a_n, b_n (trigonometric form), and c_n (exponential form) of the electrostatic potential φ using the DA method described in sec. 3.2.3.2. This *Mathematica* notebook is listed in App. E.

The *Mathematica* notebook outputs the unscaled Fourier modes a_0, a_n, and b_n (trigonometric form) as the scalar value `a0` and the lists `an` and `bn`, respectively:

```
a0=0.39605146671241603
an={−0.06885669876025119,0.010385498335510007,
−0.0012979130328780464,0.00011476439020259089,
−8.616661281676434∗10^−6,−8.001918237312118∗10^−8,
2.0824622994066985∗10^−7,−5.650565366333305∗10^−8,
```

```
9.49311257232955*10^-9,-1.1783959917396393*10^-9,
8.674992182657303*10^-11,4.7439258953997255*10^-12,
-2.0785575798462163*10^-12,4.148094011328047*10^-13,
-6.33695369929115*10^-14,8.492752468745464*10^-15,
-1.873561549973473*10^-15,3.4306594232475737*10^-16,
-4.803243173690091*10^-17,3.624992112376469*10^-18,
6.630918080108423*10^-19,-3.3213814949302035*10^-19,
8.966581646343429*10^-20,-1.9952929970350762*10^-20,
3.793742595796802*10^-21,-2.8012134244396143*10^-22}
bn={8.336931069526638*10^-10,-3.794138538379455*10^-10,
9.09371800225003*10^-11,-1.3418145691681178*10^-11,
1.0494507650372826*10^-12,1.0332235646552139*10^-13,
-6.485090438390214*10^-14,1.6930587560640178*10^-14,
-3.120390662980916*10^-15,4.06780222146135*10^-16,
-1.5013659690567033*10^-17,-1.929422691566964*10^-17,
1.4144128584705*10^-17,-7.650940703168617*10^-18,
3.0481313019656075*10^-18,-5.667559487084578*10^-19,
-3.51094538704568*10^-19,4.6557961617641415*10^-19,
-3.166471479008234*10^-19,1.574207595519098*10^-19,
-5.787690633861256*10^-20,1.1856606658833134*10^-20,
3.380342869683127*10^-21,-5.488606040416573*10^-21,
3.8095070621154405*10^-21,-1.9698604453416265*10^-21}
```

Figs. 3.14, 3.15 show plots of the multipole expansion starting from the zeroth and the third order, respectively.

3.2.4.2.3 DA Calculation in *COSY INFINITY* ("*COSY INFINITY* DA")

We implemented the calculation of the multipole terms using the DA method described in sec. 3.2.3.2 as a *COSY INFINITY* program called `mterms`.

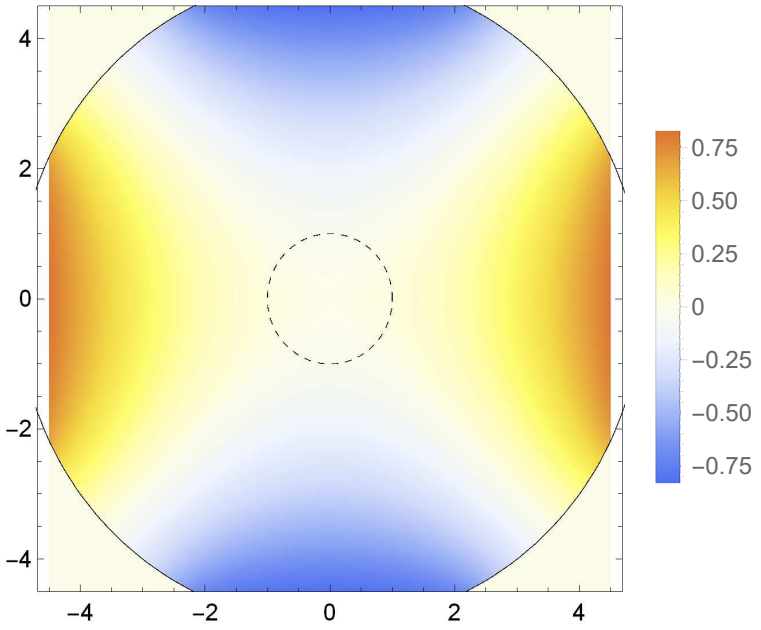


Figure 3.14: "NSM *Mathematica* DA": heatmap plot of the multipole expansion of the electrostatic potential, up to order 24.

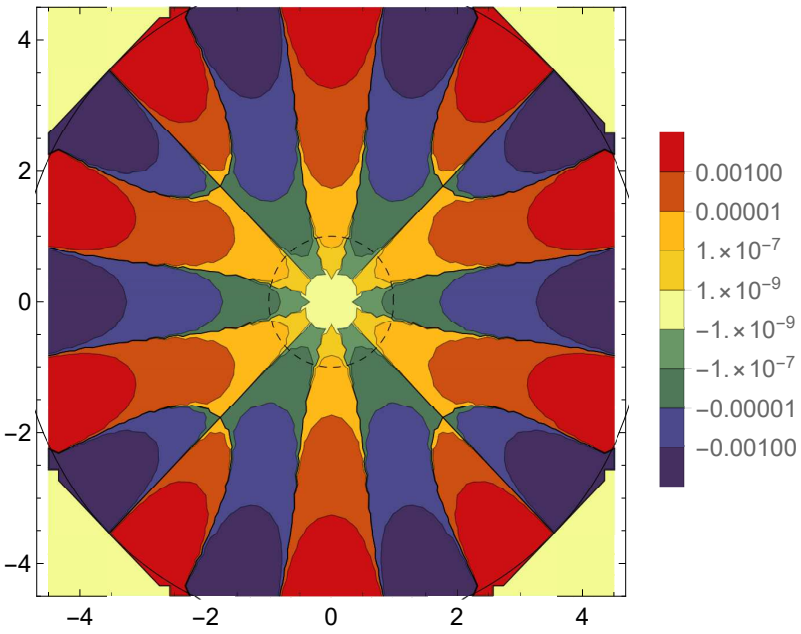


Figure 3.15: "NSM *Mathematica* DA": contour plot of the multipole expansion of the electrostatic potential, orders 3 to 24.

The program queries the user for optional changes to nominal plate voltage magnitude (default value $24,000\,\mathrm{V}$), plate voltage coefficients (default values $V_1 = 1$, $V_2 = -1$, $V_3 = 1$, and $V_4 = -1$), errors in distances from the origin to the plates (default values $D_1 = 0\,\mathrm{cm}$, $D_2 = 0\,\mathrm{cm}$, $D_3 = 0\,\mathrm{cm}$, $D_4 = 0\,\mathrm{cm}$), and scaling radius of the multipole coefficients (default value $R = 4.5\,\mathrm{cm}$).

Nonzero values of D_j were implemented using linear interpolation. For details, see the description in the error analysis section below.

The 2D geometry data is stored in `mterms` in the *SC Toolbox* format. The program generates a *MATLAB* M-file that uses the *SC Toolbox* to calculate the prevertices z_j, the constant c, the strip length L, and the preimage $g(0)$ of the origin $w = 0$. Next, multipole terms are calculated using DA. Alternatively, the user can select the option to use pre-calculated multipole terms stored in the program.

The following is the program output with the default parameters.

```
MULTIPOLE TERMS
SCALED TO R =     4.500000000000000 cm
NOMINAL PLATE VOLTAGE PHI =     24000.00000000000 V
PLATE VOLTAGE COEFFICIENTS:
PLATE  1: V1 =     1.000000000000000
PLATE  2: V2 =    -1.000000000000000
PLATE  3: V3 =     1.000000000000000
PLATE  4: V4 =    -1.000000000000000
PLATE DISTANCE ERRORS:
PLATE  1: D1 =    0.0000000000000000E+00 cm
PLATE  2: D2 =    0.0000000000000000E+00 cm
PLATE  3: D3 =    0.0000000000000000E+00 cm
PLATE  4: D4 =    0.0000000000000000E+00 cm


A0 =    0.0000000000000000E+00
```

N	AN	BN
1	−0.1270392504150563E−11	−0.1498801083243961E−11
2	20189.41051825105	0.1766027536678210E−02
3	0.1018547399907894E−11	−0.9484600604903193E−12
4	0.0000000000000000E+00	0.5532731267819562E−12
5	−0.9367417464417151E−12	−0.2250740182608863E−12
6	−63.78872501452805	−0.1248379149706613E−02
7	0.1753671159615466E−12	0.1661679275441700E−12
8	0.0000000000000000E+00	−0.2234113065318811E−12
9	0.4862302476884544E−12	0.1802625285412201E−12
10	−385.2021979951332	0.2398685362109508E−03
11	0.7086316906170801E−13	0.1568495243459249E−12
12	0.0000000000000000E+00	0.1153720159933965E−13
13	−0.7883469452910978E−13	0.3007767867428958E−13
14	55.60328952513591	0.2232092717536040E−03
15	−0.1587524669086704E−13	−0.5139053379739946E−13
16	0.0000000000000000E+00	0.1128975791355853E−13
17	−0.3264494074407542E−13	0.8431259451135848E−14
18	18.83394598327191	−0.4786556212617289E−13
19	0.1739271370955237E−13	−0.1753062093467197E−13
20	0.0000000000000000E+00	0.9765821268656394E−14
21	−0.7524327143791555E−15	−0.7234953952318767E−15
22	−6.616981865806735	0.2512483933577411E−13
23	0.1422827463617597E−14	0.1129331093494757E−13
24	0.0000000000000000E+00	0.0000000000000000E+00
25	0.0000000000000000E+00	0.0000000000000000E+00

Figs. 3.16, 3.17 show plots of the multipole expansion starting from the zeroth and the third order, respectively.

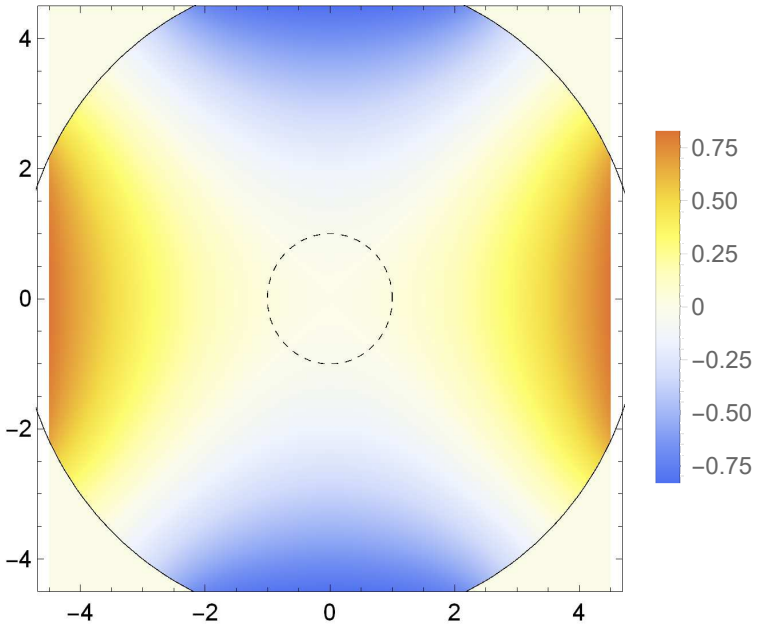


Figure 3.16: "NSM *COSY INFINITY* DA": heatmap plot of the multipole expansion of the electrostatic potential, up to order 24.

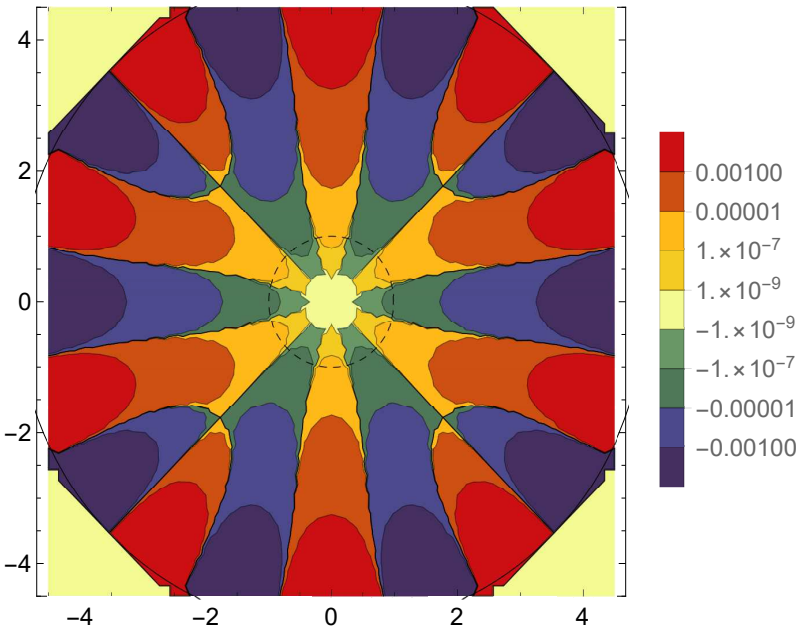


Figure 3.17: "NSM *COSY INFINITY* DA": contour plot of the multipole expansion of the electrostatic potential, orders 3 to 24.

The calculation of multipole terms using *"Mathematica* DA" and *"COSY INFINITY* DA" completed in 14.5 min and 0.07 s on the same computer.

3.2.4.2.4 *Python* Version of `mterms`

We implemented a *Python* version of the *COSY INFINITY* program `mterms` (see App. F for details and source code.). As *Python* does not have DA functionality, the *Python* version of `mterms` only use pre-calculated sets of multipole terms, with the same method and results as in the *COSY INFINITY* version introduced in case "*COSY INFINITY* DA". The *Python* version does not include *de novo* computation of multipole terms using the *SC Toolbox* and DA.

3.2.4.3 Multipole Terms

Table 3.1 lists multipole terms for cases "SM *Mathematica* PDE", "SM *MATLAB* Fourier", "NSM *MATLAB* Fourier", and "NSM *Mathematica* DA". Table 3.2 lists multipole terms for case "NSM *COSY INFINITY* DA".

For each plate voltage coefficient $V_j + \triangle V_j$ and distance from the origin D_j (8 cases in total), we computed multipole terms in "NSM *Mathematica* DA" with a perturbation $\triangle V_j$ or D_j, respectively, the rest of these parameters remaining unperturbed. The unperturbed values are $D_j = 0\,\text{cm}$ and $V_j + \triangle V_j$ with $\triangle V_j = 0$, $V_1 = 1$, $V_2 = -1$, $V_3 = 1$, and $V_4 = -1$.

Multipole terms with perturbations $\triangle V_j$ and D_j of plate voltage coefficients and plate distances are listed in Tables 3.3, 3.4 and 3.5, 3.6, respectively.

We note that in "NSM *COSY INFINITY* DA", multipole terms for order $n = 24$ are numerically calculated as zero even if the computation order in `mterms` is increased from the default value of 25 to 30. This is due to floating-point underflow.

3.2.4.4 Error Analysis

In this *a posteriori* error analysis, we use methods and measures that include the following:

Order n	SM *Mathematica* PDE	SM *MATLAB* Fourier	NSM *MATLAB* Fourier	NSM *Mathematica* DA
1	$1.9745943647419334 \times 10^{-12}$	$9.079982791749051 \times 10^{-14}$	0.	0.
2	20214.97653712142	20232.98426780849	20189.40925816141	20189.40876423145
3	$2.5046868116525192 \times 10^{-12}$	$-1.5984993366164723 \times 10^{-11}$	0.	0.
4	$-1.999071489496519 \times 10^{-6}$	$-1.6264478976965794 \times 10^{-11}$	0.	0.
5	$8.845917707820406 \times 10^{-13}$	$9.14806631798289 \times 10^{-12}$	0.	0.
6	-99.48139575133254	-118.26614359059352	-63.788529979036376	-63.788211449330845
7	$1.5853077240419715 \times 10^{-12}$	$8.78863940492948 \times 10^{-12}$	0.	0.
8	$-8.017648259522616 \times 10^{-13}$	$-2.7459971391853632 \times 10^{-11}$	0.	0.
9	$1.298726494930936 \times 10^{-12}$	$-2.6192553454071565 \times 10^{-11}$	0.	0.
10	-259.77056186360966	-360.8978554756162	-385.20109622396365	-385.20119645612806
11	$7.355032700884383 \times 10^{-13}$	$8.287110710130606 \times 10^{-12}$	0.	0.
12	$-1.1562641498237162 \times 10^{-12}$	$3.043766956845234 \times 10^{-11}$	0.	0.
13	$1.762557385977699 \times 10^{-12}$	$2.3045039516510903 \times 10^{-11}$	0.	0.
14	27.990065388237984	49.6181080101752	55.602386906876	55.602589324698265
15	$3.3925916602277514 \times 10^{-12}$	$4.107208903423128 \times 10^{-12}$	0.	0.
16	$-1.828818941841522 \times 10^{-12}$	$1.0954794995043528 \times 10^{-14}$	0.	0.
17	$1.3782403619675239 \times 10^{-12}$	$8.153607909284305 \times 10^{-12}$	0.	0.
18	-84.09455183738272	19.954184119037713	18.8341954294158	18.85706451271575
19	0.	$5.86335215814601 \times 10^{-13}$	0.	0.
20	-0.000035820067317830976	$-3.6922316850239525 \times 10^{-11}$	0.	0.
21	$1.5107634736951705 \times 10^{-12}$	$-4.0231745053258074 \times 10^{-11}$	0.	0.
22	0.3874045390909366	-7.1158335523071745	-6.616941027862705	-7.486266749401171
23	$1.013801804716496 \times 10^{-12}$	$6.954205882858485 \times 10^{-12}$	0.	0.
24	$-2.87575152448993 \times 10^{-12}$	$2.4426166197145148 \times 10^{-11}$	0.	0.

Table 3.1: Multipole terms $M_{n,n}$ in cases "SM *Mathematica* PDE", "SM *MATLAB* Fourier", "NSM *MATLAB* Fourier", and "NSM *Mathematica* DA".

Order n	Cosine terms A_0, A_n	Sine terms B_n
0	0.	N/A
1	$-1.270392504150563 \times 10^{-12}$	$-1.498801083243961 \times 10^{-12}$
2	20189.41051825105	$1.766027536678210 \times 10^{-3}$
3	$1.018547399907894 \times 10^{-12}$	$-9.484600604903193 \times 10^{-13}$
4	0.	$5.532731267819562 \times 10^{-13}$
5	$-9.367417464417151 \times 10^{-13}$	$-2.250740182608863 \times 10^{-13}$
6	-63.78872501452805	$-1.248379149706613 \times 10^{-3}$
7	$1.753671159615466 \times 10^{-13}$	$1.661679275441700 \times 10^{-13}$
8	0.	$-2.234113065318811 \times 10^{-13}$
9	$4.862302476884544 \times 10^{-13}$	$1.802625285412201 \times 10^{-13}$
10	-385.2021979951332	$2.398685362109508 \times 10^{-4}$
11	$7.086316906170801 \times 10^{-14}$	$1.568495243459249 \times 10^{-13}$
12	0.	$1.153720159933965 \times 10^{-14}$
13	$-7.883469452910978 \times 10^{-14}$	$3.007767867428958 \times 10^{-14}$
14	55.60328952513591	$2.232092717536040 \times 10^{-4}$
15	$-1.587524669086704 \times 10^{-14}$	$-5.139053379739946 \times 10^{-14}$
16	0.	$1.128975791355853 \times 10^{-14}$
17	$-3.264494074407542 \times 10^{-14}$	$8.431259451135848 \times 10^{-15}$
18	18.83394598327191	$-4.786556212617289 \times 10^{-14}$
19	$1.739271370955237 \times 10^{-14}$	$-1.753062093467197 \times 10^{-14}$
20	0.	$9.765821268656394 \times 10^{-15}$
21	$-7.524327143791555 \times 10^{-16}$	$-7.234953952318767 \times 10^{-16}$
22	-6.616981865806735	$2.512483933577411 \times 10^{-14}$
23	$1.422827463617597 \times 10^{-15}$	$1.129331093494757 \times 10^{-14}$
24	0.	0.

Table 3.2: Multipole terms A_0, A_n, and B_n in case "NSM *COSY INFINITY* DA". The sine multipole terms are included for comparison with "NSM *COSY INFINITY* DA" multipole terms calculated with perturbed geometry and voltages in Tables 3.3, 3.4, 3.5, and 3.6.

Order n	$\triangle V_1 = 10^{-5}$	$\triangle V_2 = -10^{-5}$	$\triangle V_3 = 10^{-5}$	$\triangle V_4 = -10^{-5}$
0	$9.505235201160260 \times 10^{-2}$	$-9.505235201160260 \times 10^{-2}$	$9.505235201160260 \times 10^{-2}$	$-9.505235201160260 \times 10^{-2}$
1	$-7.436523597891738 \times 10^{-2}$	$8.266246815751127 \times 10^{-11}$	$7.436523597637661 \times 10^{-2}$	$-8.515344572191423 \times 10^{-11}$
2	20189.46099177735	20189.46099177735	20189.46099177735	20189.46099177735
3	$-2.838536489744372 \times 10^{-2}$	$5.994811899668481 \times 10^{-9}$	$2.838536489948082 \times 10^{-2}$	$-5.992913734586033 \times 10^{-9}$
4	$1.129454381431691 \times 10^{-2}$	$-1.129454381431691 \times 10^{-2}$	$1.129454381431691 \times 10^{-2}$	$-1.129454381431691 \times 10^{-2}$
5	$-3.816042156893218 \times 10^{-3}$	$5.361648904967867 \times 10^{-10}$	$3.816042155019734 \times 10^{-3}$	$-5.380342277459531 \times 10^{-10}$
6	-63.78888448634059	-63.78888448634059	-63.78888448634059	-63.78888448634059
7	$1.867570482658803 \times 10^{-3}$	$-3.900047013374706 \times 10^{-9}$	$-1.867570482308069 \times 10^{-3}$	$3.900383589753429 \times 10^{-9}$
8	$-2.280364607148135 \times 10^{-3}$	$2.280364607148135 \times 10^{-3}$	$-2.280364607148135 \times 10^{-3}$	$2.280364607148135 \times 10^{-3}$
9	$1.723985667991497 \times 10^{-3}$	$1.310874637723331 \times 10^{-9}$	$-1.723985667019037 \times 10^{-3}$	$-1.309883051616690 \times 10^{-9}$
10	-385.2031610006283	-385.2031610006283	-385.2031610006283	-385.2031610006283
11	$3.190201723315910 \times 10^{-4}$	$1.851384811087571 \times 10^{-9}$	$-3.190201721898647 \times 10^{-4}$	$-1.851244694012594 \times 10^{-9}$
12	$7.850716940114915 \times 10^{-5}$	$-7.850716940114915 \times 10^{-5}$	$7.850716940114915 \times 10^{-5}$	$-7.850716940114915 \times 10^{-5}$
13	$-1.547902248761007 \times 10^{-4}$	$-1.535845107803755 \times 10^{-9}$	$1.547902247184313 \times 10^{-4}$	$1.535685413417173 \times 10^{-9}$
14	55.60342853335972	55.60342853335972	55.60342853335972	55.60342853335972
15	$-9.555798106664925 \times 10^{-5}$	$-2.717804380859608 \times 10^{-10}$	$9.555798103489875 \times 10^{-5}$	$2.717485332842149 \times 10^{-10}$
16	$5.761744754729378 \times 10^{-5}$	$-5.761744754729378 \times 10^{-5}$	$5.761744754729378 \times 10^{-5}$	$-5.761744754729378 \times 10^{-5}$
17	$-5.717782937401201 \times 10^{-5}$	$7.582411050566096 \times 10^{-10}$	$5.717782930872212 \times 10^{-5}$	$-7.583065672290997 \times 10^{-10}$
18	18.83399306813687	18.83399306813687	18.83399306813687	18.83399306813687
19	$-2.965993100024038 \times 10^{-5}$	$1.739271370955237 \times 10^{-14}$	$2.965993103502581 \times 10^{-5}$	$1.739297528979414 \times 10^{-14}$
20	$1.022093822923510 \times 10^{-5}$	$-1.022093822923510 \times 10^{-5}$	$1.022093822923510 \times 10^{-5}$	$-1.022093822923510 \times 10^{-5}$
21	$7.572076577709902 \times 10^{-6}$	$-7.524327143791555 \times 10^{-16}$	$-7.572076579214769 \times 10^{-6}$	$-7.524438561832755 \times 10^{-16}$
22	-6.616998408261400	-6.616998408261400	-6.616998408261400	-6.616998408261400
23	$1.787269032222077 \times 10^{-5}$	$1.422827463617597 \times 10^{-15}$	$-1.787269031937511 \times 10^{-5}$	$1.422783699214458 \times 10^{-15}$
24	0.	0.	0.	0.

Table 3.3: Cosine multipole terms A_0, A_n in case "NSM *COSY INFINITY* DA" with specified perturbations $\triangle V_j$ of plate voltage coefficients.

Order n	$\triangle V_1 = 10^{-5}$	$\triangle V_2 = -10^{-5}$	$\triangle V_3 = 10^{-5}$	$\triangle V_4 = -10^{-5}$
1	$8.243405957841787 \times 10^{-11}$	$7.436523597614819 \times 10^{-2}$	$-8.543166174490580 \times 10^{-11}$	$-7.436523597914579 \times 10^{-2}$
2	$1.766031951747060 \times 10^{-3}$	$1.766031951747054 \times 10^{-3}$	$1.766031951747048 \times 10^{-3}$	$1.766031951747042 \times 10^{-3}$
3	$-5.994741812329063 \times 10^{-9}$	$-2.838536489941073 \times 10^{-2}$	$5.992844892208082 \times 10^{-9}$	$2.838536489751381 \times 10^{-2}$
4	$3.161414553920099 \times 10^{-9}$	$-3.160308004931769 \times 10^{-9}$	$3.161414548386968 \times 10^{-9}$	$-3.160307999398638 \times 10^{-9}$
5	$5.368765582249675 \times 10^{-10}$	$3.816042155731402 \times 10^{-3}$	$-5.373267062614893 \times 10^{-10}$	$-3.816042156181550 \times 10^{-3}$
6	$-1.248382270654487 \times 10^{-3}$	$-1.248382270654487 \times 10^{-3}$	$-1.248382270654487 \times 10^{-3}$	$-1.248382270654487 \times 10^{-3}$
7	$3.900364810142848 \times 10^{-9}$	$1.867570482649604 \times 10^{-3}$	$-3.900032474287760 \times 10^{-9}$	$-1.867570482317268 \times 10^{-3}$
8	$-3.117515154626748 \times 10^{-9}$	$3.117068330893074 \times 10^{-9}$	$-3.117515152436465 \times 10^{-9}$	$3.117068328702791 \times 10^{-9}$
9	$1.310568670004184 \times 10^{-9}$	$-1.723985667325004 \times 10^{-3}$	$-1.310208144947101 \times 10^{-9}$	$1.723985667685530 \times 10^{-3}$
10	$2.398691358822895 \times 10^{-4}$	$2.398691358822901 \times 10^{-4}$	$2.398691358822907 \times 10^{-4}$	$2.398691358822948 \times 10^{-4}$
11	$-1.851161851410052 \times 10^{-9}$	$3.190201724175774 \times 10^{-4}$	$1.851475550458744 \times 10^{-9}$	$-3.190201721038783 \times 10^{-4}$
12	$2.145763565102821 \times 10^{-9}$	$-2.145740490658827 \times 10^{-9}$	$2.145763565021230 \times 10^{-9}$	$-2.145740490536441 \times 10^{-9}$
13	$-1.535736195430552 \times 10^{-9}$	$1.547902248273437 \times 10^{-4}$	$1.535796350787900 \times 10^{-9}$	$-1.547902247671883 \times 10^{-4}$
14	$2.232098297767835 \times 10^{-4}$	$2.232098297767833 \times 10^{-4}$	$2.232098297767832 \times 10^{-4}$	$2.232098297767836 \times 10^{-4}$
15	$2.717131723054726 \times 10^{-10}$	$-9.555798110216454 \times 10^{-5}$	$-2.718159533730674 \times 10^{-10}$	$9.555798099938346 \times 10^{-5}$
16	$-7.035211691164543 \times 10^{-10}$	$7.035437486894595 \times 10^{-10}$	$-7.035211692308105 \times 10^{-10}$	$7.035437487956475 \times 10^{-10}$
17	$7.582821812568048 \times 10^{-10}$	$5.717782934979833 \times 10^{-5}$	$-7.582653187379024 \times 10^{-10}$	$-5.717782933293581 \times 10^{-5}$
18	$-4.786556212617289 \times 10^{-14}$	$-4.786561402226913 \times 10^{-14}$	$-4.786566591836537 \times 10^{-14}$	$-4.786588509350543 \times 10^{-14}$
19	$-1.753062093467197 \times 10^{-14}$	$-2.965993103516372 \times 10^{-5}$	$-1.753062093467197 \times 10^{-14}$	$2.965993100010247 \times 10^{-5}$
20	$9.765821268656394 \times 10^{-15}$	$9.765833785695678 \times 10^{-15}$	$9.765796234577818 \times 10^{-15}$	$9.765931443908367 \times 10^{-15}$
21	$-7.234953952318767 \times 10^{-16}$	$-7.572076579185831 \times 10^{-6}$	$-7.234953952318767 \times 10^{-16}$	$7.572076577738841 \times 10^{-6}$
22	$2.512483933577411 \times 10^{-14}$	$2.512492039090965 \times 10^{-14}$	$2.512500144604519 \times 10^{-14}$	$2.512484741876086 \times 10^{-14}$
23	$1.129331093494757 \times 10^{-14}$	$1.787269033209125 \times 10^{-5}$	$1.129331093494757 \times 10^{-14}$	$-1.787269030950463 \times 10^{-5}$
24	0.	0.	0.	0.

Table 3.4: Sine multipole terms B_n in case "NSM *COSY INFINITY* DA" with specified perturbations $\triangle V_j$ of plate voltage coefficients.

Order n	$D_1 = 10^{-5}$ cm	$D_2 = 10^{-5}$ cm	$D_3 = 10^{-5}$ cm	$D_4 = 10^{-5}$ cm
0	$-4.424991842810755 \times 10^{-2}$	$4.424991842810755 \times 10^{-2}$	$-4.424991842810755 \times 10^{-2}$	$4.424991842810755 \times 10^{-2}$
1	$1.57876858921876 \times 10^{-2}$	$5.40683282654414 \times 10^{-6}$	$-1.578768589310954 \times 10^{-2}$	$-5.406834852347936 \times 10^{-6}$
2	20189.38795547691	20189.38795547691	20189.38795547691	20189.38795547691
3	$4.104796527084991 \times 10^{-3}$	$2.413725184267611 \times 10^{-4}$	$-4.104796525037468 \times 10^{-3}$	$-2.413725160359214 \times 10^{-4}$
4	$-2.947473221000326 \times 10^{-3}$	$2.947473221000326 \times 10^{-3}$	$-2.947473221000326 \times 10^{-3}$	$2.947473221000326 \times 10^{-3}$
5	$-1.817416554008738 \times 10^{-3}$	$2.062335646140603 \times 10^{-5}$	$1.81741655220565 \times 10^{-3}$	$-2.062335826776784 \times 10^{-5}$
6	-63.78592414544445	-63.78592414544445	-63.78592414544445	-63.78592414544445
7	$-2.467557291457782 \times 10^{-3}$	$-1.555384668326032 \times 10^{-4}$	$2.467557291799718 \times 10^{-3}$	$1.555384671863416 \times 10^{-4}$
8	$3.07300944321362 \times 10^{-3}$	$-3.07300944321362 \times 10^{-3}$	$3.07300944321362 \times 10^{-3}$	$-3.07300944321362 \times 10^{-3}$
9	$-1.556422642040572 \times 10^{-3}$	$5.247298500533006 \times 10^{-5}$	$1.556422643039743 \times 10^{-3}$	$-5.247298399725565 \times 10^{-5}$
10	-385.2009674038089	-385.2009674038089	-385.2009674038089	-385.200967403809
11	$-2.600024552646485 \times 10^{-4}$	$7.378270763189497 \times 10^{-5}$	$2.600024554055848 \times 10^{-4}$	$-7.378270748964422 \times 10^{-5}$
12	$-1.818382215511822 \times 10^{-4}$	$1.818382215511822 \times 10^{-4}$	$-1.818382215511822 \times 10^{-4}$	$1.818382215511822 \times 10^{-4}$
13	$4.506844218163286 \times 10^{-4}$	$-6.115724123437846 \times 10^{-5}$	$-4.506844219761083 \times 10^{-4}$	$6.115724107481854 \times 10^{-5}$
14	55.60278639311861	55.60278639311861	55.60278639311861	55.60278639311861
15	$3.710213603257118 \times 10^{-4}$	$-1.10878930476378 \times 10^{-5}$	$-3.710213603579178 \times 10^{-4}$	$1.108789301549602 \times 10^{-5}$
16	$-3.223080948135712 \times 10^{-4}$	$3.223080948135712 \times 10^{-4}$	$-3.223080948135712 \times 10^{-4}$	$3.223080948135712 \times 10^{-4}$
17	$1.182560483985171 \times 10^{-4}$	$3.004623224282604 \times 10^{-5}$	$-1.182560484631099 \times 10^{-4}$	$-3.004623230765092 \times 10^{-5}$
18	18.83386356141041	18.83386356141041	18.83386356141041	18.83386356141041
19	$-1.370788357518164 \times 10^{-5}$	$1.38000473541436 \times 10^{-7}$	$1.370788360996695 \times 10^{-5}$	$-1.380004387559051 \times 10^{-7}$
20	$6.371905486653597 \times 10^{-5}$	$-6.371905486653597 \times 10^{-5}$	$6.371905486653597 \times 10^{-5}$	$-6.371905486653597 \times 10^{-5}$
21	$-8.12996954732784 \times 10^{-5}$	$6.224477582157721 \times 10^{-8}$	$8.12996954717736 \times 10^{-5}$	$-6.224477732632099 \times 10^{-8}$
22	-6.616896403654835	-6.616896403654835	-6.616896403654835	-6.616896403654835
23	$-5.935334052843666 \times 10^{-5}$	$1.729810664202958 \times 10^{-10}$	$5.935334053128231 \times 10^{-5}$	$-1.729782206155512 \times 10^{-10}$
24	0.	0.	0.	0.

Table 3.5: Cosine multipole terms A_0, A_n in case "NSM *COSY INFINITY* DA" with specified perturbations D_j of plate distances from the origin.

Order n	$D_1 = 10^{-5}$ cm	$D_2 = 10^{-5}$ cm	$D_3 = 10^{-5}$ cm	$D_4 = 10^{-5}$ cm
1	$5.406833480936513 \times 10^{-6}$	$-1.578768589410862 \times 10^{-2}$	$-5.406834979737596 \times 10^{-6}$	$1.578768589111101 \times 10^{-2}$
2	$2.117816741216531 \times 10^{-3}$	$2.117816741216532 \times 10^{-3}$	$2.117816741216535 \times 10^{-3}$	$2.117816741216537 \times 10^{-3}$
3	$-2.413725186012523 \times 10^{-4}$	$4.104796524838491 \times 10^{-3}$	$2.413725162301022 \times 10^{-4}$	$-4.104796527209642 \times 10^{-3}$
4	$-6.241207228382077 \times 10^{-7}$	$6.241218293842069 \times 10^{-7}$	$-6.241207228369357 \times 10^{-7}$	$6.241218293831894 \times 10^{-7}$
5	$2.062335718565881 \times 10^{-5}$	$1.817416552817827 \times 10^{-3}$	$-2.062335763580684 \times 10^{-5}$	$-1.817416553342999 \times 10^{-3}$
6	$-1.496641666694401 \times 10^{-3}$	$-1.496641666694403 \times 10^{-3}$	$-1.496641666694403 \times 10^{-3}$	$-1.496641666694404 \times 10^{-3}$
7	$1.55538467170032 \times 10^{-4}$	$-2.467557291475427 \times 10^{-3}$	$-1.555384668614345 \times 10^{-4}$	$2.467557291807763 \times 10^{-3}$
8	$5.479028625297525 \times 10^{-7}$	$-5.479033093512959 \times 10^{-7}$	$5.479028625271038 \times 10^{-7}$	$-5.479033093486471 \times 10^{-7}$
9	$5.247298470503409 \times 10^{-5}$	$1.556422642722333 \times 10^{-3}$	$-5.247298431446527 \times 10^{-5}$	$-1.556422642361809 \times 10^{-3}$
10	$2.875084409755162 \times 10^{-4}$	$2.875084409755162 \times 10^{-4}$	$2.875084409755149 \times 10^{-4}$	$2.875084409755111 \times 10^{-4}$
11	$-7.378270740737561 \times 10^{-5}$	$-2.600024551827504 \times 10^{-4}$	$7.378270772107465 \times 10^{-5}$	$2.600024554916964 \times 10^{-4}$
12	$1.982877226920842 \times 10^{-7}$	$-1.982876996177626 \times 10^{-7}$	$1.982877226923698 \times 10^{-7}$	$-1.982876996180889 \times 10^{-7}$
13	$-6.115724112566527 \times 10^{-5}$	$-4.506844218671139 \times 10^{-4}$	$6.115724118582062 \times 10^{-5}$	$4.506844219272692 \times 10^{-4}$
14	$2.676013289590522 \times 10^{-4}$	$2.676013289590525 \times 10^{-4}$	$2.676013289590529 \times 10^{-4}$	$2.676013289590516 \times 10^{-4}$
15	$1.108789298159123 \times 10^{-5}$	$3.710213602890859 \times 10^{-4}$	$-1.108789308437229 \times 10^{-5}$	$-3.710213603918668 \times 10^{-4}$
16	$-1.716345429854064 \times 10^{-7}$	$1.716345655646772 \times 10^{-7}$	$-1.716345429848183 \times 10^{-7}$	$1.716345655641217 \times 10^{-7}$
17	$3.004623228456449 \times 10^{-5}$	$-1.18256048423242 \times 10^{-4}$	$-3.004623226770197 \times 10^{-5}$	$1.182560484401045 \times 10^{-4}$
18	$-4.786550271350386 \times 10^{-14}$	$-4.786548313818143 \times 10^{-14}$	$-4.786534788640852 \times 10^{-14}$	$-4.786507688191113 \times 10^{-14}$
19	$-1.380004736793519 \times 10^{-7}$	$-1.370788361010491 \times 10^{-5}$	$1.380004386181101 \times 10^{-7}$	$1.370788357504367 \times 10^{-5}$
20	$9.76581027479493 \times 10^{-15}$	$9.76589863347751 \times 10^{-15}$	$9.76565929472537 \times 10^{-15}$	$9.76649188717796 \times 10^{-15}$
21	$6.224477585051353 \times 10^{-8}$	$8.12996954718025 \times 10^{-5}$	$-6.224477729750432 \times 10^{-8}$	$-8.12996954732495 \times 10^{-5}$
22	$2.512482697969831 \times 10^{-14}$	$2.512441608739869 \times 10^{-14}$	$2.512405049788011 \times 10^{-14}$	$2.512476577073213 \times 10^{-14}$
23	$-1.72968350279657 \times 10^{-10}$	$-5.935334051856618 \times 10^{-5}$	$1.729909369015269 \times 10^{-10}$	$5.93533405411528 \times 10^{-5}$
24	0.	0.	0.	0.

Table 3.6: Sine multipole terms B_n in case "NSM *COSY INFINITY* DA" with specified perturbations D_j of plate distances from the origin.

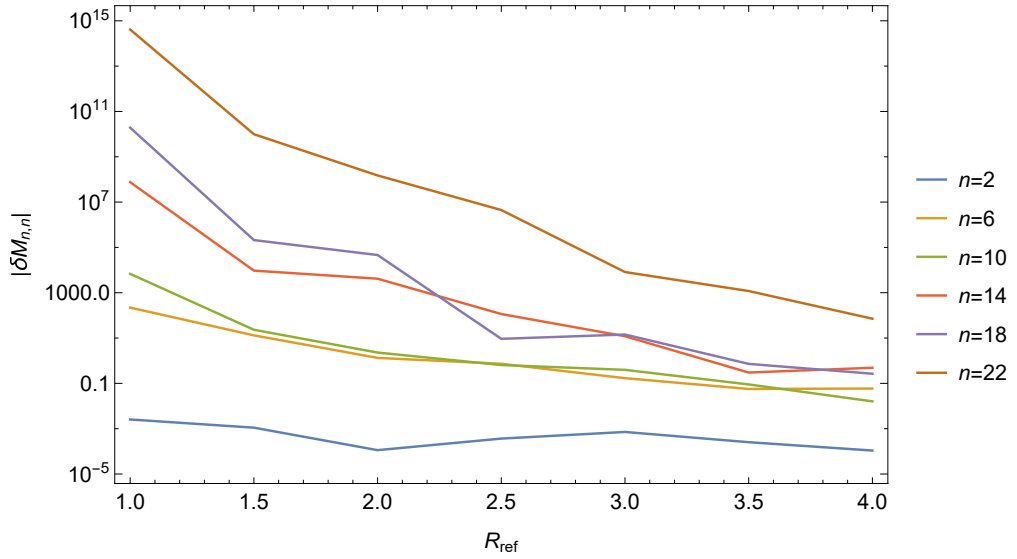


Figure 3.18: "SM *Mathematica* PDE": multipole terms $M_{n,n}$ versus reference radius R_{ref}.

1. The apparent accuracy computed by the *SC Toolbox*. It is defined as the maximum of the differences of (1) the differences $w_{j+1} - w_j$ between successive finite polygon vertices and (2) the integrals $\int_{v_j}^{v_{j+1}} f'(\zeta)\, d\zeta$ between the respective prevertices.

2. Adaptations of C. Runge's heuristic rule of *a posteriori* error estimation: "if the difference between two approximate solutions computed on a coarse mesh $\mathscr{T}_h$ with mesh size h and refined mesh $\mathscr{T}_{h_{\text{ref}}}$ with mesh size h_{ref} (e.g., $h_{\text{ref}} = h/2$)" becomes small, then both approximated solutions $u_{h_{\text{ref}}}$ and u_h are probably close to the exact solution" [72, p. 31] and "also, the quantity $u_h - u_{h_{\text{ref}}}$ can be viewed (in terms of modern terminology) as a certain *a posteriori* error indicator" [91].

3. Relative differences between the approximations of multipole terms.

4. Difference plots between the approximated multipole expansions and the approximation of the electrostatic potential in the *SC Toolbox* from the inverse of the conformal mapping object.

3.2.4.4.1 "SM *Mathematica* PDE"

Fig. 3.18 shows the multipole terms $M_{n,n}$, scaled to $R = 4.5\,\text{cm}$, versus the reference radius R_{ref}. The relative difference between multipole terms $M_{n,n}$ at $R_{\text{ref}} = 4.0\,\text{cm}$ and $R_{\text{ref}} = 4.5\,\text{cm}$ ranges

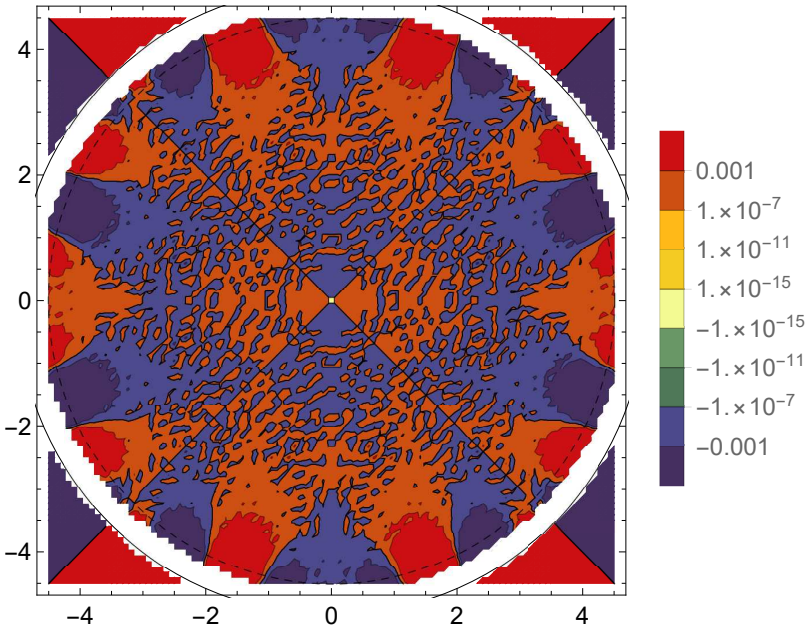


Figure 3.19: Contour plot of the difference between the electrostatic potential φ obtained in (1) "SM *Mathematica* PDE" as a multipole expansion up to order 24 and in (2) SM from the inverse of the conformal mapping object `f`.

from $\sim 10^{-4}\%$ for $M_{2,2}$ to $\sim 10^{2}\%$ for $M_{22,22}$.

Fig. 3.19 shows the difference between the potential φ obtained in (1) "SM *Mathematica* PDE" and in (2) SM from the inverse of the conformal mapping object `f`.

3.2.4.4.2 "SM *MATLAB* Fourier"

Fig. 3.20 shows the convergence of multipole terms $M_{n,n}$, scaled to $R = 4.5\,\text{cm}$, as a function of error tolerance ε specified by a line of the form

```
options=scmapopt('Tolerance',epsilon);
```

in the function `initquad4`. The relative difference between multipole terms $M_{n,n}$ at $\varepsilon = 10^{-15}$ and $\varepsilon = 10^{-16}$ ranges from $\sim 10^{-15}\%$ for $M_{2,2}$ to $\sim 10^{-12}\%$ for $M_{22,22}$.

Fig. 3.21 shows the multipole terms $M_{n,n}$, scaled to $R = 4.5\,\text{cm}$, versus the reference radius R_{ref}. The relative difference between multipole terms $M_{n,n}$ at $R_{\text{ref}} = 4.0\,\text{cm}$ and $R_{\text{ref}} = 4.5\,\text{cm}$ ranges from $\sim 10^{-15}\%$ for $M_{2,2}$ to $\sim 10^{-10}\%$ for $M_{22,22}$.

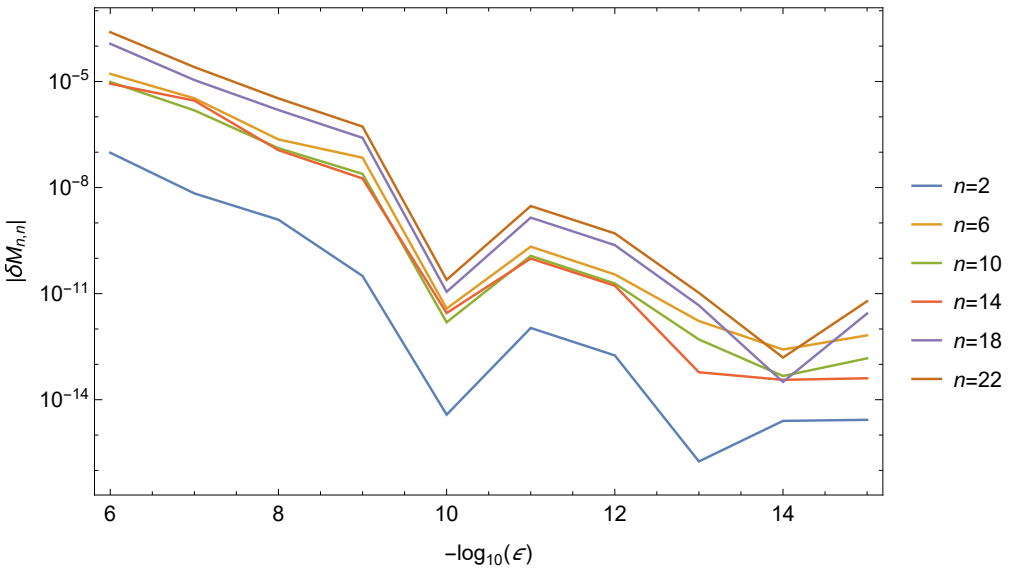


Figure 3.20: "SM *MATLAB* Fourier": convergence of multipole terms $M_{n,n}$ versus error tolerance ϵ.

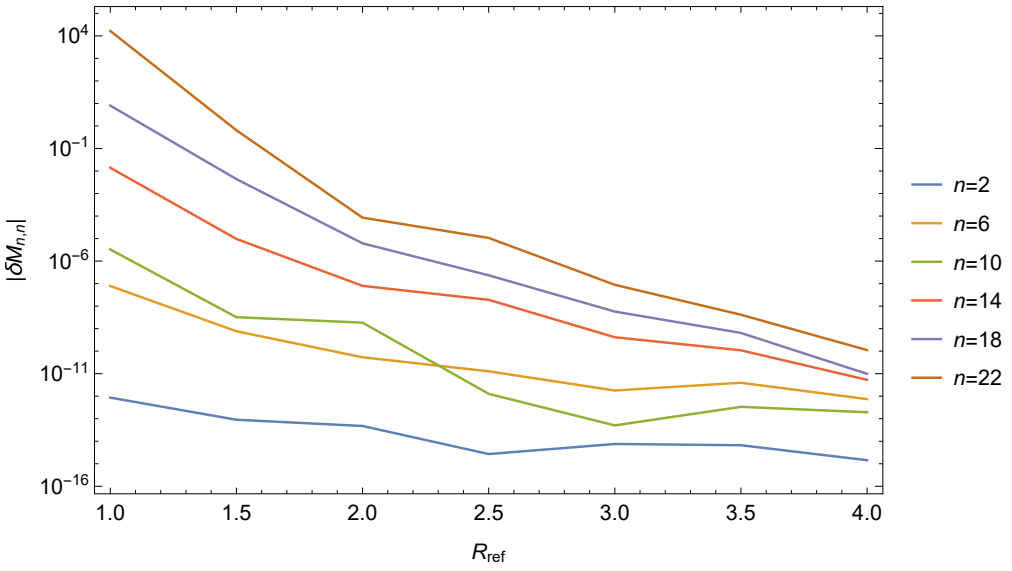


Figure 3.21: "SM *MATLAB* Fourier": multipole terms $M_{n,n}$ versus reference radius R_{ref}.

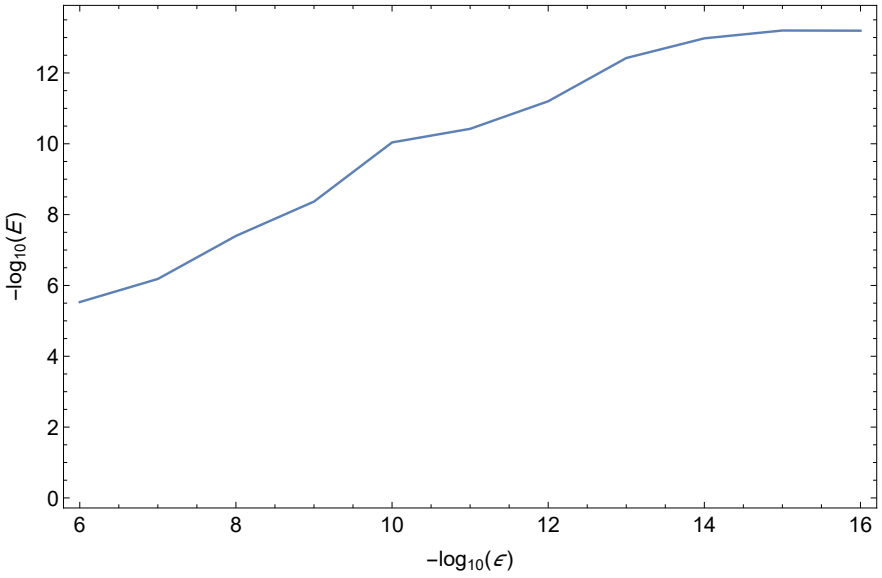


Figure 3.22: "SM *MATLAB* Fourier": apparent accuracy E versus error tolerance ε.

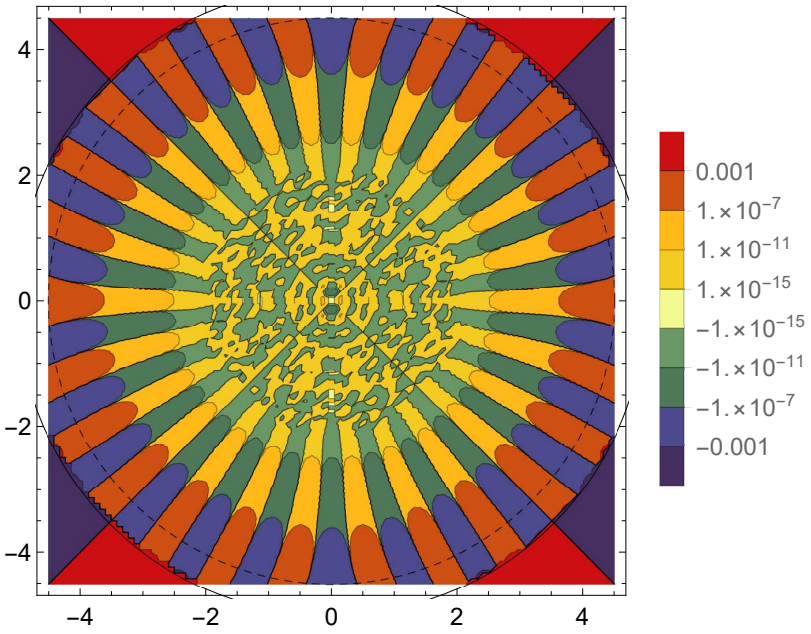


Figure 3.23: Contour plot of the difference between the electrostatic potential φ obtained in (1) "SM *MATLAB* Fourier" as a multipole expansion up to order 24 and (2) SM from the inverse of the conformal mapping object `f`.

The apparent accuracy in the case of $\varepsilon = 10^{-16}$ is 6.40×10^{-14}. Fig. 3.22 shows the resulting apparent accuracy E versus the specified error tolerance ε.

Fig. 3.23 shows the difference between the electrostatic potential φ obtained in (1) "SM *MATLAB* Fourier" and (2) SM from the inverse of the conformal mapping object `f`.

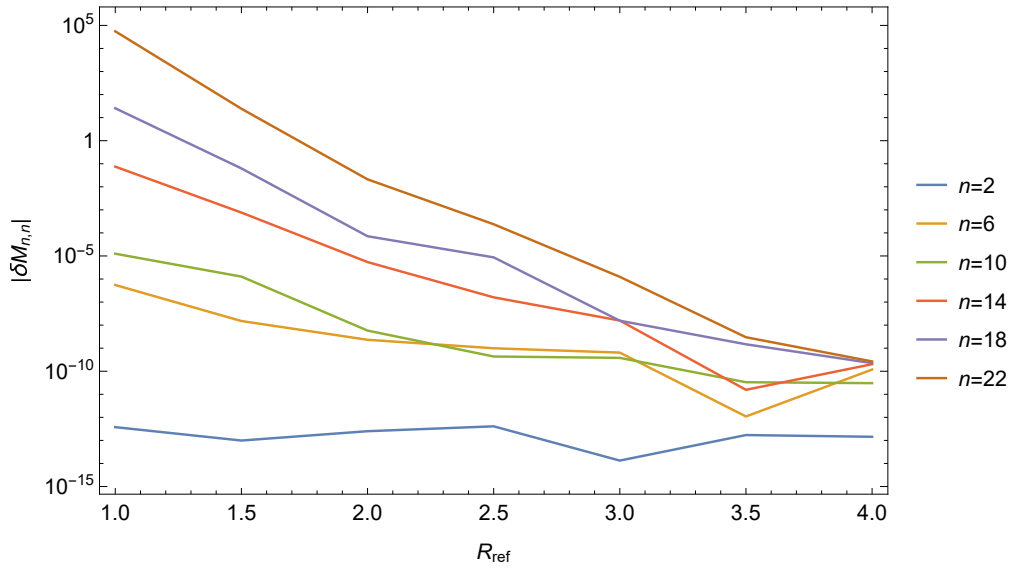


Figure 3.24: "NSM *MATLAB* Fourier": multipole terms $M_{n,n}$ versus reference radius R_{ref}.

3.2.4.4.3 "NSM *MATLAB* Fourier"

With the error tolerance of $\varepsilon = 10^{-16}$, out of MATLAB releases R2009a, R2014b, R2016b, and R2017a, the computations successfully completed only in releases R2016b and R2017a. Changing the error tolerance ε in the function `initquad7` resulted in aborted computation or very low apparent accuracy.

Fig. 3.24 shows the multipole terms $M_{n,n}$, scaled to $R = 4.5\,\text{cm}$, versus the reference radius R_{ref}. The relative difference between multipole terms $M_{n,n}$ at $R_{\text{ref}} = 4.0\,\text{cm}$ and $R_{\text{ref}} = 4.5\,\text{cm}$ ranges from $\sim 10^{-13}\%$ for $M_{2,2}$ to $\sim 10^{-10}\%$ for $M_{22,22}$.

The apparent accuracy in the case of $\varepsilon = 10^{-16}$ is 8.07×10^{-11}.

Fig. 3.25 shows the difference between the electrostatic potential φ obtained in (1) "NSM *MATLAB* Fourier" and (2) NSM from the inverse of the conformal mapping object `f`.

3.2.4.4.4 "NSM *Mathematica* DA"

Fig. 3.26 shows the difference between the electrostatic potential φ obtained in (1) "NSM *Mathematica* DA" and (2) NSM from the inverse of the conformal mapping object `f`.

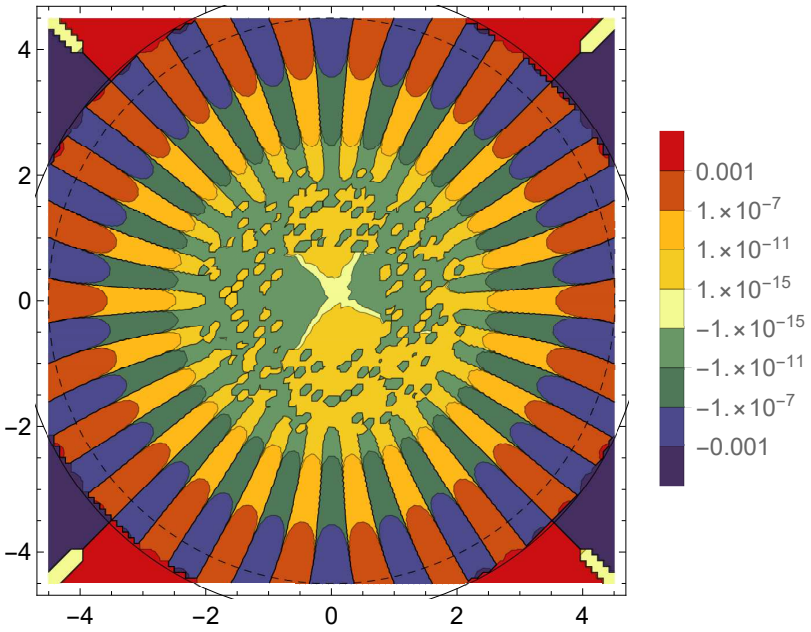


Figure 3.25: Contour plot of the difference between the electrostatic potential φ obtained in (1) "NSM *MATLAB* Fourier" as a multipole expansion up to order 24 and (2) NSM from the inverse of the conformal mapping object `f`.

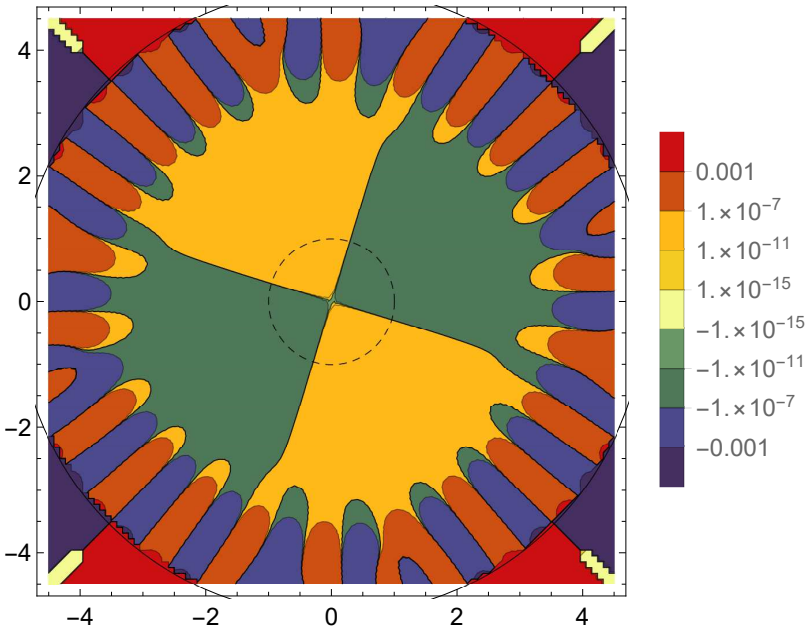


Figure 3.26: Contour plot of the difference between the electrostatic potential φ obtained in (1) "NSM *Mathematica* DA" as a multipole expansion up to order 24 and in (2) NSM from the inverse of the conformal mapping object `f`.

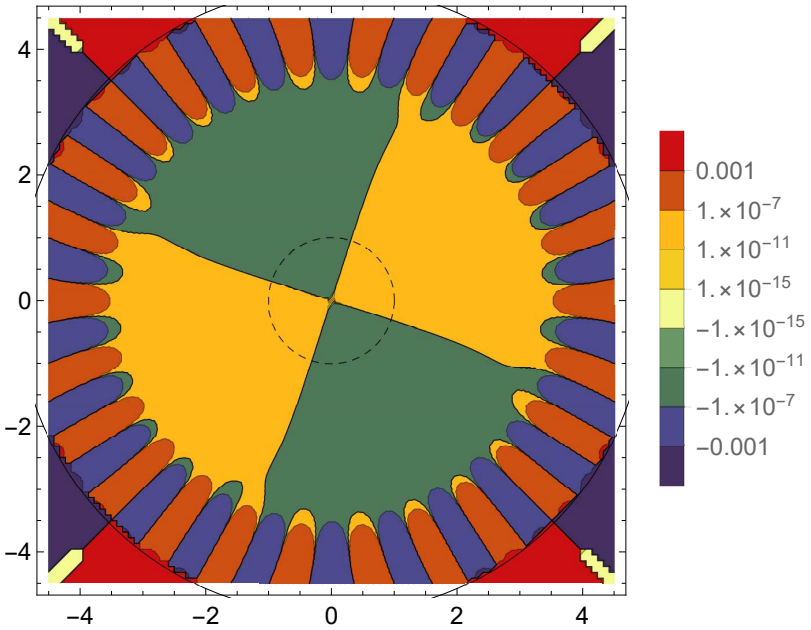


Figure 3.27: Contour plot of the difference between the electrostatic potential φ obtained in (1) "NSM *COSY INFINITY* DA" as a multipole expansion up to order 24 and in (2) NSM from the inverse of the conformal mapping object `f`.

3.2.4.4.5 "NSM *COSY INFINITY* DA"

Fig. 3.27 shows the difference between the electrostatic potential φ obtained in (1) "NSM *COSY INFINITY* DA" and (2) NSM from the inverse of the conformal mapping object `f`.

3.2.4.4.6 Relative Differences

Table 3.7 shows relative differences between (1) significant multipole terms $M_{n,n}$ in the three cases "SM *Mathematica* PDE", "SM *MATLAB* Fourier", "NSM *Mathematica* DA", and "NSM *MATLAB* Fourier" and (2) significant multipole terms $M_{n,n}$ in the "NSM *COSY INFINITY* DA" case.

The significant difference between SM and NSM results may be explained by the modeling error due to the approximation of the 2D geometry using the connecting rods in the NSM case.

We note the particularly small relative differences between the "NSM *COSY INFINITY* DA" and "NSM *MATLAB* Fourier" cases. Heuristically, we suppose that the orders of the relative errors of these methods do not exceed the orders of these relative differences.

Order n	NSM *MATLAB* Fourier	NSM *Mathematica* DA	SM *MATLAB* Fourier	SM *Mathematica* PDE
2	$-6.24134 \times 10^{-6}\%$	$-8.68782 \times 10^{-6}\%$	0.215825%	0.126631%
6	−0.000305752%	−0.000805103%	85.4029%	55.9545%
10	−0.000286024%	−0.000260003%	−6.3095%	−32.5625%
14	−0.00162332%	−0.00125928%	−10.7641%	−49.6611%
18	0.00132445%	0.122749%	5.94797%	−546.505%
22	−0.000617169%	13.1372%	7.53896%	−105.855%

Table 3.7: Relative differences between (1) significant multipole terms $M_{n,n}$ in cases "SM *Mathematica* PDE", "SM *MATLAB* Fourier", "NSM *Mathematica* DA", and "NSM *MATLAB* Fourier" and (2) significant multipole terms $M_{n,n}$ in the "NSM *COSY INFINITY* DA" case.

We also note that in "SM *Mathematica* PDE", the Laplace equation was solved in *Mathematica* with boundary conditions obtained from the conformal mapping object `f`.

3.2.4.4.7 Plate Distance Perturbations

In the NSM case, we used linear interpolation for perturbations of plate distances from the origin, based on successful calculations in the *SC Toolbox* with small perturbations of plate distances.

In the process of obtaining the linear interpolation, we considered the effect on multipole terms of various nonzero perturbations D_j of distances of the plates from the origin.

Plate distance perturbations are applied to four copies of the electrostatic potential, where plate voltage coefficients are $V_1 = 1$, $V_2 = 0$, $V_3 = 0$, and $V_4 = 0$ (i.e., $V = (1,0,0,0)$). At this stage of the calculations, nominal plate voltage magnitude is not yet applied and is effectively 1 V.

For each plate number j, we attempted calculations in the *SC Toolbox* with about 20–40 different values of perturbation D_j, about 6–8 of which completed successfully despite the crowding phenomenon.

For the linear interpolation, we selected four sets of perturbed multipole terms data, namely, calculated with perturbations $D_1 = -5.04 \times 10^{-5}$ cm, $D_2 = -5 \times 10^{-5}$ cm, $D_3 = -5.1 \times 10^{-5}$ cm, and $D_4 = -4.8 \times 10^{-5}$ cm. These perturbation values were based on proximity to -5×10^{-5} cm – one value of a suitable order of magnitude – and availability of data from successfully completed calculations.

We considered finite difference approximations $\triangle A_0/D_j$, $\triangle A_n/D_j$, and $\triangle B_n/D_j$ of the first derivatives (FDAFDs) of multipole terms by and as functions of perturbations D_j. Their behavior may be placed in two groups:

1. FDAFDs have relatively small variances, typically within one order of magnitude. Linear interpolation is mostly justified. See Fig. 3.28 for an example with $\triangle A_n/D_2$. This group comprises mostly the cosine sets and one sine set: $\triangle A_0/D_1$, $\triangle A_n/D_1$, $\triangle A_0/D_2$, $\triangle A_0/D_2$, $\triangle A_n/D_2$, $\triangle A_0/D_3$, $\triangle A_n/D_3$, $\triangle A_0/D_4$, $\triangle A_n/D_4$, and $\triangle B_n/D_2$.

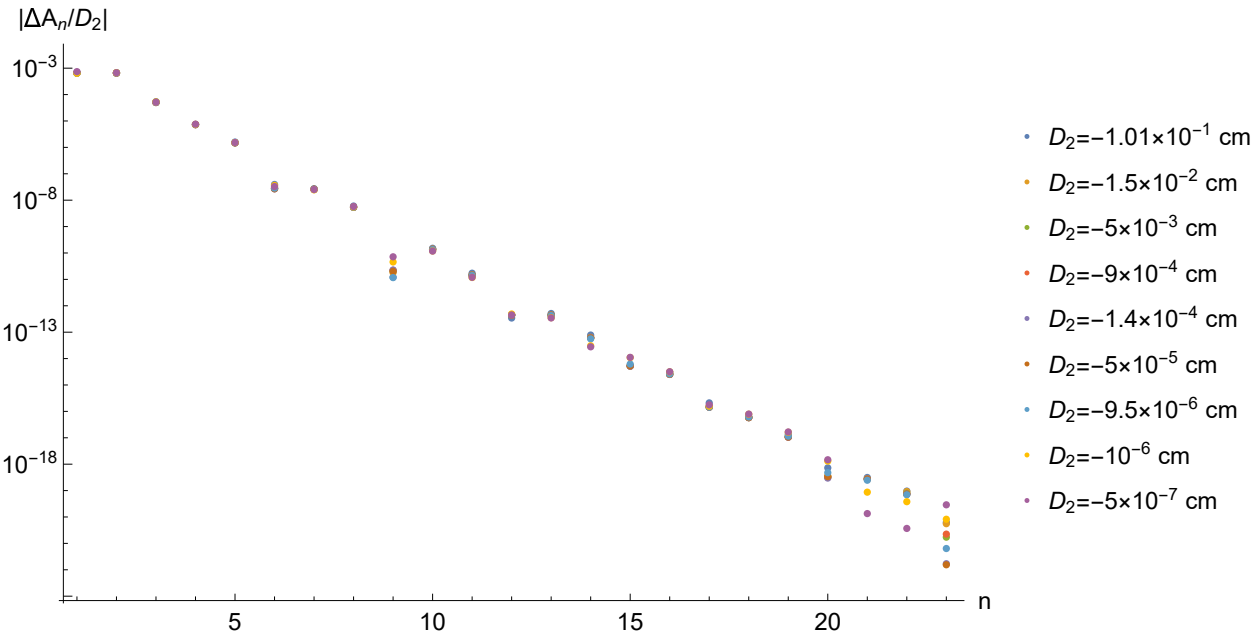


Figure 3.28: Absolute values of finite difference approximations $\triangle A_n/D_2$ of the first derivatives of cosine multipole terms A_n with respect to perturbations D_2 in case "NSM *COSY INFINITY* DA". In many cases, plot markers denoting $\triangle A_n/D_2$ for lower magnitudes of D_2 cover or are in close proximity to plot markers higher magnitudes of D_2. This indicates the validity of the linear interpolation, with limitations shown by some disagreement in $\triangle A_n/D_2$ at orders $n = 9$ and $n \geq 20$.

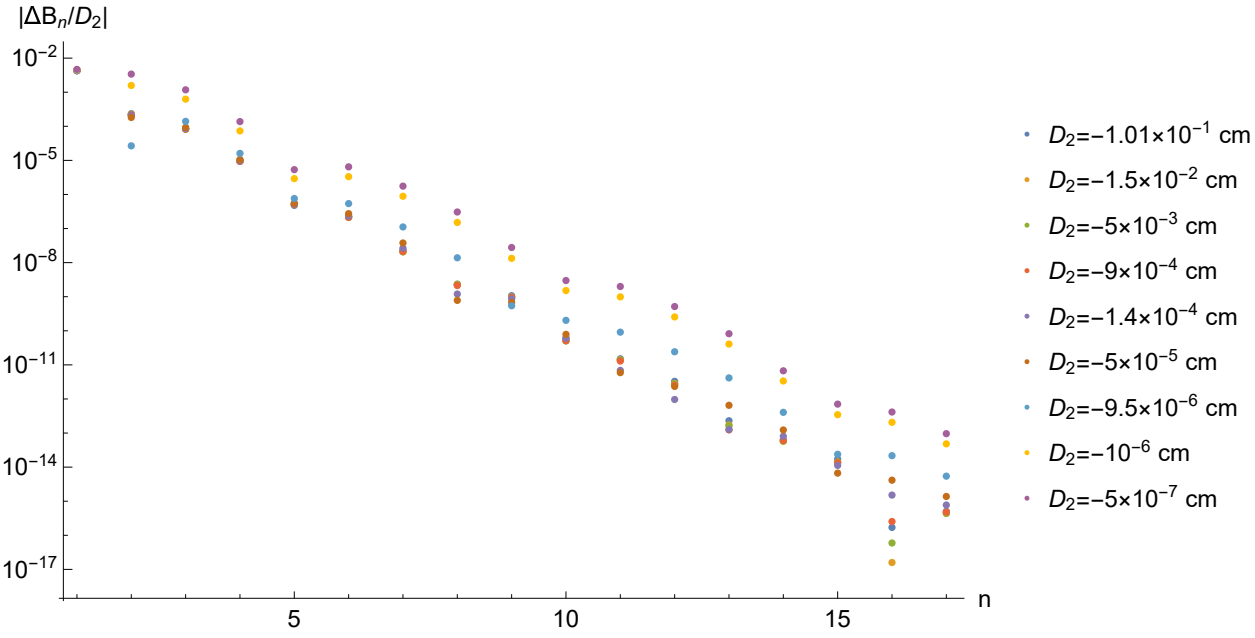


Figure 3.29: Absolute values of finite difference approximations $\triangle B_n/D_2$ of the first derivatives of sine multipole terms B_n with respect to perturbations D_2 in case "NSM *COSY INFINITY* DA". Some plot markers denoting $\triangle B_n/D_2$ for lower magnitudes of D_2 cover or are in close proximity to plot markers higher magnitudes of D_2. Overall, the dependence of multipole terms B_n on D_2 does not appear to be linear.

2. FDAFDs have relatively high variances. The dependence of multipole terms on the D_j does not appear to be linear. See Fig. 3.29 for an example with $\triangle B_n/D_2$. This group comprises three sine sets: $\triangle B_n/D_1$, $\triangle B_n/D_2$, and $\triangle B_n/D_3$.

Linear interpolation results are relatively accurate for cosine multipole terms but should be considered with care for sine multipole terms.

We note that the theoretical values for sine multipole terms are zero if the cross-sectional geometry has a reflection symmetry relative to the horizontal axis, that is, if $D_2 = D_4$. However, in application of plate distance errors to the four copies of the electrostatic potential, the plate distance errors value set is rotated as $\{D_1, D_2, D_3, D_3\}$, $\{D_2, D_3, D_4, D_1\}$, $\{D_3, D_4, D_1, D_2\}$, and $\{D_4, D_1, D_2, D_3\}$ to account for the subsequent rotation of the four copies of the electrostatic potential. Thus, considering the rotations of the electrostatic potential, the sine multipole terms are generally useful even if $D_2 = D_4$ in the unrotated cross-sectional geometry.

3.2.5 Main Field Discussion

We presented a method to calculate the main field of the Muon g-2 collaboration quadrupole. In this context, we provided the theoretical framework; the method with several submethods; *MATLAB, Python*, and *Mathematica* source code; calculations and their results; and an *a posteriori* error analysis.

The use of conformal mapping methods for the calculation of the main field has the advantage of an analytic, fully Maxwellian formula and allows rapid recalculations with adjustments to the geometry and mispowered plates.

The submethods represent different approaches and their implementations to obtain the inverse conformal mapping in order to calculate the multipole expansion of the electrostatic potential: (1) taking the DA inverse of the conformal mapping or (2) solving an ODE (explicitly or implicitly) to obtain the inverse conformal mapping from an analytic expression for the first derivative of the conformal mapping. For the "*COSY INFINITY* DA" submethod, we developed a *COSY INFINITY* program that uses the *SC Toolbox* to calculate the parameters of the conformal mapping from a

rectangle to a polygonal representation of the cross-sectional geometry and then calculates the DA inverse of the conformal mapping to obtain the multipole terms.

Apart from the geometrical approximation of the polygonal model, the error analysis indicates that *"COSY INFINITY* DA" and *"MATLAB* Fourier" submethods are the most accurate compared to the other considered submethods.

In the general case of mispowered plates or geometric asymmetries, we applied the common practice of approximation of the geometry to obtain a simply-connected polygon. To minimize the impact on the multipole terms, we performed the approximation by connecting the plates with the rectangular enclosure in the middle of each plate's back side.

It may be possible to obtain a method similar to the method presented with an analytic formula for a conformal mapping from a canonical domain to a multiply-connected region. In that case, it would no longer be necessary to approximate the cross section by a simply-connected polygon for mispowered-plates calculations. See, e.g., [22] for the current state of the theory of conformal maps to multiply-connected regions.

To improve the accuracy of the polygonal model, a script or a software program may be developed to automatically generate the polygonal model data from a CAD drawing of the cross section of the quadrupole.

Calculations in case of geometric asymmetries often fail in the *SC Toolbox* due to the crowding phenomenon [6]. To expand the functionality to geometric asymmetries, we considered an adaptation of the cross ratios of the Delaunay triangulation (CRDT) algorithm, as described in [30, 31, 7].

We implemented the geometric asymmetries functionality by treating plate distance errors as perturbations and using linear interpolation based on geometric asymmetries calculations that were successful in the *SC Toolbox* despite the crowding phenomenon.

This section presented a method for calculation of 2D main field of the Muon g-2 collaboration quadrupole, including rapid recalculation capability for geometric asymmetries and mispowered plates.

3.3 Fringe Field

3.3.1 Coordinate System

To study the fringe field of the Muon g-2 collaboration quadrupole, we used a Cartesian laboratory coordinate system based on a specific local Frenet–Serret frame with a horizontal x axis and a vertical y axis in the transversal section and a z axis tangent to the curvilinear reference trajectory coordinate line s. We approximated the internal geometry of the quadrupole by a z-invariant model, considering the relatively large radius of curvature $R_0 = 7112\,\mathrm{mm}$ of the quadrupole. The reference orbit is straight in this model, and, therefore, coordinates s and z are here practically equivalent. We will use s and z interchangeably, depending on the context.

The origin of coordinate lines x and y is the intersection of the cross-sectional plane with the reference trajectory. The origin of the coordinate line z is aligned with the exit edge of the quadrupole element. Coordinates $z > 0$ are outside the quadrupole, and coordinates in the range $z < 0$ are inside the quadrupole.

In the x–y plane, we will alternatively use the polar coordinate system with radial coordinate r and angular coordinate θ. The following relations between (x, y, z) and (s, r, θ) apply: $x = r\cos\theta$, $y = r\sin\theta$, and $z = s$.

3.3.2 Multipole Terms

We begin the discussion with the Taylor–Fourier expansion of the potential φ [13, p. 50]

$$\varphi(r,\theta,s) = \sum_{k=0}^{+\infty}\sum_{l=0}^{+\infty} M_{k,l}(s)\cos\left(l\theta + \theta_{k,l}\right) r^k. \tag{3.4}$$

Furthermore, we compare it with the Fourier expansion

$$\varphi(r,\theta,s) = \frac{a_0(r,s)}{2} + \sum_{l=0}^{+\infty} a_l(r,s)\cos\left(l\theta + \theta_l\right).$$

We can assume the constant parts in these expansions to be zero, considering the gauge invariance of the electrostatic potential.

Multipole terms $M_{k,l}(s)$ vanish for $k < l$ and $k = l+1, l+3, \ldots$ in the general case. If the field is s-independent, multipole terms $M_{k,l}(s)$ vanish for all $k \neq l$ [13, p. 52].

The fringe field is s-dependent, and, respectively, multipole terms $M_{k,l}(s)$ for $k > l$ do not entirely vanish. To obtain multiple terms $M_{k,l}(s)$ using Fourier analysis, an equivalent of considering the Fourier modes at all radii $r > 0$ would be necessary. In practice, this approach presents challenges, and a reasonable approximation is to perform the Fourier expansion at a finite set of radii.

For simplicity of notation, going forward, we omit the coordinate s, writing $M_{k,l}$ and $a_l(r)$ instead of $M_{k,l}(s)$ and $a_l(r,s)$, respectively.

Thus, for a set of radii r_j for $j = 1, 2, \ldots, N$,

$$a_l\left(r_j\right) = \sum_{m=0}^{+\infty} M_{l+2m,l} r_j^{l+2m}. \tag{3.5}$$

An approximation of the terms $M_{k,l}$, where $k = l, l+2, \cdots, l+2(N-1)$, [57] is the solution of the matrix equation

$$\begin{pmatrix} a_l(r_1) \\ \vdots \\ a_l(r_N) \end{pmatrix} = \begin{pmatrix} r_1^l & \cdots & r_1^{l+2(N-1)} \\ \vdots & \ddots & \vdots \\ r_N^l & \cdots & r_N^{l+2(N-1)} \end{pmatrix} \begin{pmatrix} M_{l,l} \\ \vdots \\ M_{l+2(N-1),l} \end{pmatrix}.$$

The falloffs of the multipole strengths $M_{l,l}(s)$ can be computed by solving this matrix equation at a grid-point set of coordinates s. Along the coordinate line s, the gradient of the electrostatic potential is the highest near the edge of the element. Accordingly, for computational efficiency, the grid-point set should have a higher density near the edge of the element than elsewhere in the set.

This method, as described here, uses electrostatic potential data. However, the method can be adapted to use electrostatic field data by taking the gradient or a partial derivative of the electrostatic potential, e.g., $\frac{\partial \varphi}{\partial r}$, in the multipole and Fourier expansions [57].

3.3.3 STEP Files

In this method, we generate 3D model data from a polygonal model of the cross-sectional geometry in the STEP file format (ISO 10303-242 [55]), and we cross-checked our resulting STEP format

3D model data files by importing to *CATIA*, *Autodesk Inventor*, *IDA-STEP* (developed by LKSoft, `http://www.ida-step.net/`), and *STEP File Analyzer* (developed by Robert Lipman, `http://go.usa.gov/yccx`).

We developed a software program called *STEP File Generator* (or `stepfg`). This program produces 3D STEP files from polygonal models specified by vertices. Compared to performing this process manually in CAD software, our software has workflow efficiency advantages. The resulting STEP file can be used in many general 3D, including BEM (or FEM) solver, programs. In our case, we feed the output file to *COULOMB*.

This software was implemented as a *Mathematica* notebook and as a *Python* script. The *Python* script is listed in App. G.

3.3.4 STEP Model of the Quadrupole

For the purposes of calculating the multipole strength falloffs and the EFB, the length L of the model representing the quadrupole is chosen to be sufficient to cover the fringe field falloff area but as short as possible for computational efficiency. The *STEP File Generator* effectively extruded a polygonal model of a 90° section of the full cross section, specified in the x–y plane, to span the $z > -L/2$ half of the full model along the z axis, i.e., the range $-50 \leq z \leq 0\,\text{cm}$.

The following *STEP File Generator* input produces the STEP-formatted 3D part as shown in Fig. 3.30.

```
[[[[0,5.3],[2.35,5.3],[2.35,5],[0,5]],[[5,0],[5,2.35],[5.3,2.35],
[5.3,0]],[[0,7],[0,8],[8,8],[8,0],[7,0],[7,5.59792],
[5.879900042201571253,5.59792],
[4.775937526375983,4.493957484174412],
[4.493957484174412,4.775937526375983],
[5.59792,5.879900042201571253],[5.59792,7,0]]],[0,50],10]
```

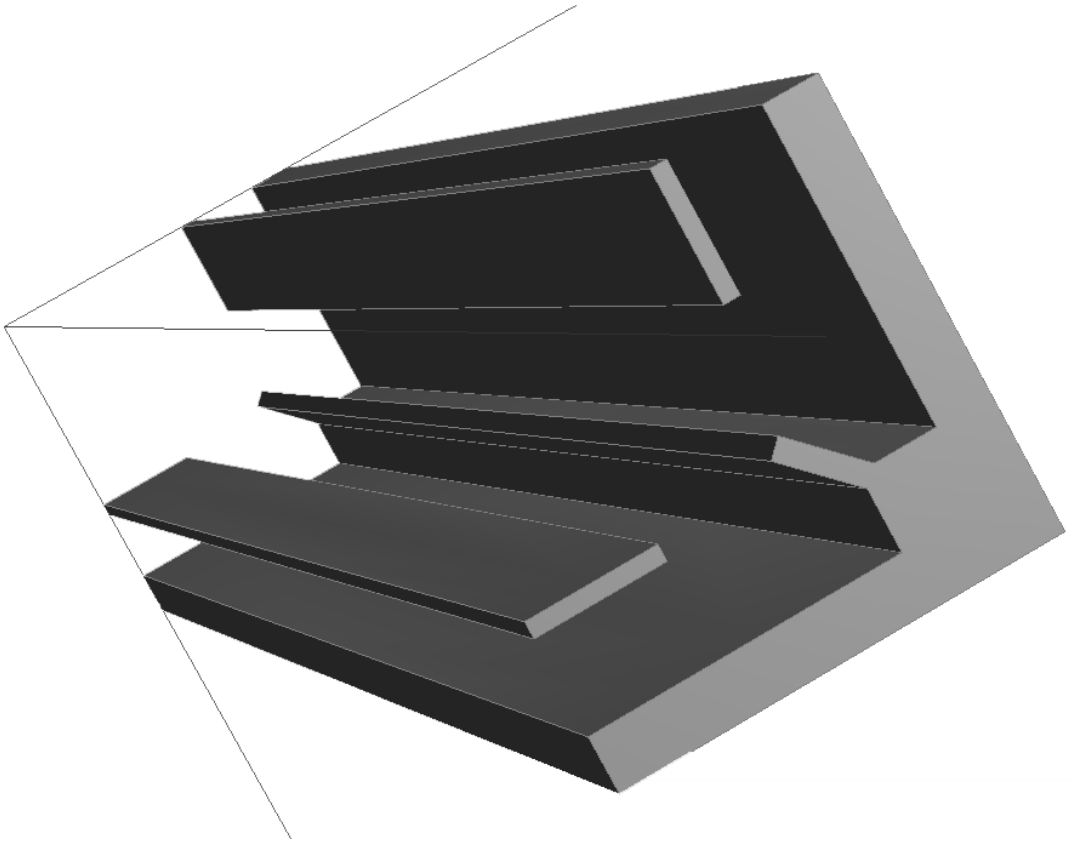


Figure 3.30: Input for the Muon g-2 collaboration quadrupole model: rendering of the resulting STEP-formatted 3D part.

3.3.5 Grid-Point Set for Approximated Potential

We discretized concentrically in the radial direction with every 0.3 cm in the range $1.8 \leq r \leq 4.5$ cm, considering that n-th Fourier modes of the potential are asymptotically equivalent to r^n as $r \to 0$.

On the z axis, we discretized the range $-50 \leq z \leq 150$ cm into a set of 605 points. As justified above and as Fig. 3.31 shows, this is relatively dense near the edge of the quadrupole.

The 3D grid was obtained as combinations of the (x, y) and z coordinates from the respective discretizations. Considering the cross-sectional symmetries, it sufficed to only include points on the circles within an arc of central angle 45°.

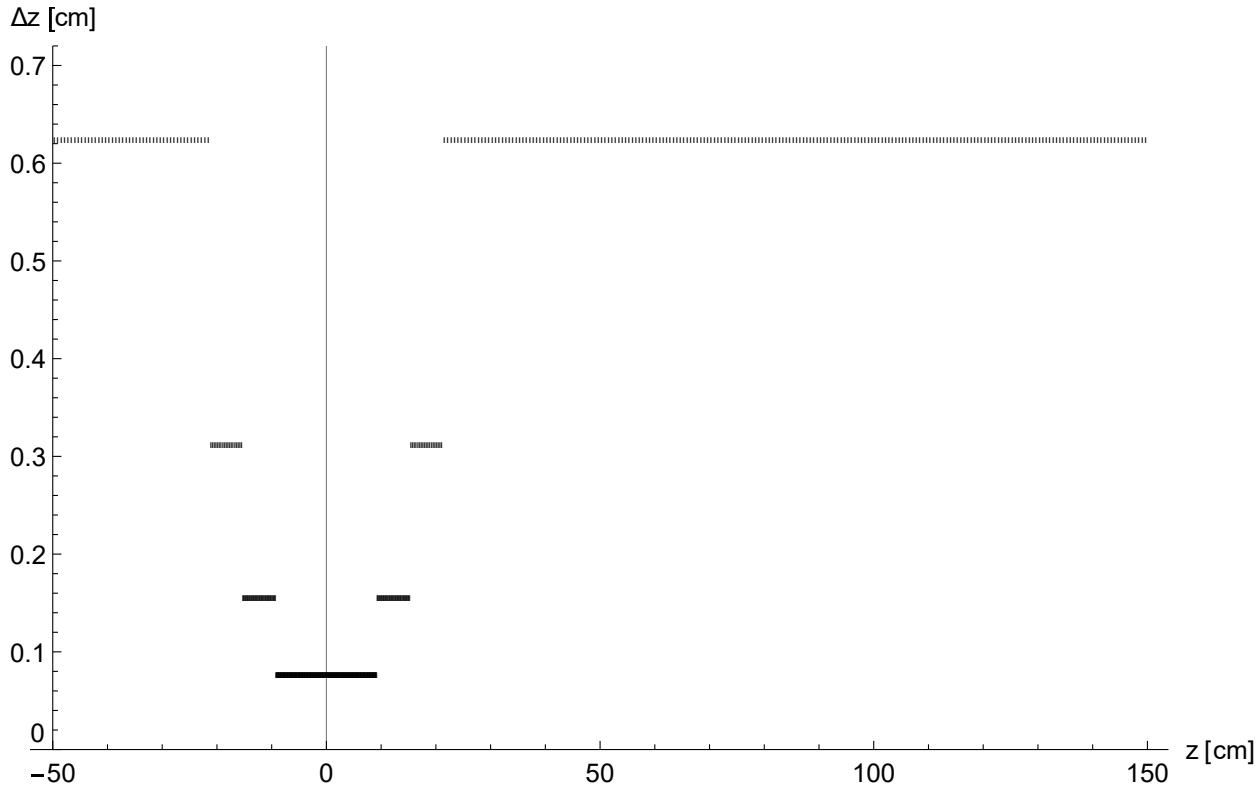


Figure 3.31: Coordinate grid for calculation of Fourier modes of the electrostatic potential, with 605 grid points: z discretization size $\triangle z$. The discretization size $\triangle z$, as a function of z, was set in an iterative discretization process by the mesh refinement function $f(z) = 1/\left(1+9\exp\left(-0.006z^2\right)\right)$. Initially, $\triangle z$ was set to 10 cm uniformly in the range $-50 \leq z \leq 150$ cm. The grid simplexes were then bisected until the condition $\triangle z(z) \leq f(z)$ was met.

3.3.6 Calculations in *COULOMB*

The full geometry of the model was reconstructed in *COULOMB* (see sec. 2.4.1) by specifying the four-fold rotational symmetry in the cross section and the longitudinal reflection symmetry. Such geometrical symmetries are also utilized by *COULOMB*'s BEM solver.

Calculations were performed as detailed above.

3.3.7 The Enge Function and the EFB

Applying a fast Fourier (FFT) transform to the electrostatic potential at grid points along circles at $z = -50$ cm, we obtained the multipole terms of the internal electrostatic potential of the quadrupole. The multipole terms are listed in Table 3.8 (compare to multipole terms of the quadrupole calculated

Order n	Multipole Terms
1	0
2	20214.96881114685
3	0
4	$-4.894473818198626 \times 10^{-13}$
5	0
6	-116.75916539917267
7	0
8	$1.3760505820154435 \times 10^{-12}$
9	0
10	-360.85446229120816
11	0
12	$3.767315759511883 \times 10^{-13}$
13	0
14	49.42528231242991
15	0
16	$-3.5694866918021975 \times 10^{-12}$
17	0
18	20.0146160932168
19	0
20	$1.8421872795880613 \times 10^{-11}$
21	0
22	-7.123965949192845
23	0
24	$-4.643530388208997 \times 10^{-12}$

Table 3.8: Multipole terms $M_{n,n}$ obtained from the 3D electrostatic potential data calculated in *COULOMB*, using an FFT transform along the circle of the radius $R_{\text{ref}} = 3.6\,\text{cm}$ at $z = -50\,\text{cm}$, scaled to $R = 4.5\,\text{cm}$.

from the cross section using conformal maps, listed in sec. 3.2).

We obtained an approximation of the multipole strengths $M_{n,n}(z)$ in the discretized range $-50 \leq z \leq 150\,\text{cm}$ using the multipole terms method as above. Because our approximation uses the first several orders of the multipole expansions, it is necessary to minimize the effect of higher orders. Therefore, we omitted the $r > 3\,\text{cm}$ part.

Fig. 3.32 shows the falloff of the multipole strength $M_{2,2}$. The EFB is $z_{\text{EFB}} = 1.2195\,\text{cm}$.

Table 3.9 lists the Enge function coefficients fitted, using the Levenberg–Marquardt Gauss–Newton method and the least-squares variance function, to the falloff of the multipole

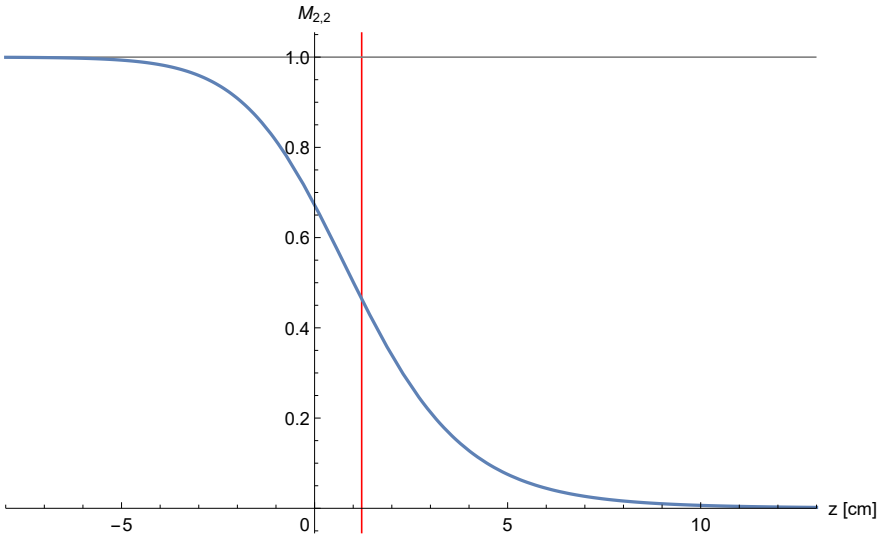


Figure 3.32: Falloff of the multipole strength $M_{2,2}$, normalized to $M_{2,2} = 1$ well inside the element. The origin of the z axis is aligned with the quadrupole edge. The vertical gridline denotes the EFB $z_{\mathrm{EFB}} = 1.2195\,\mathrm{cm}$.

strength $M_{2,2}$.

We produced plots of the Enge function and its 2nd and 4th derivatives, shown in Fig. 3.33, using the FP command in the following *COSY INFINITY* [68] code. A set of such plots is useful for visual analysis of an Enge function, including the Enge function's asymmetry relative to the intersection of the Enge function and its EFB gridline.

```
INCLUDE 'COSY' ;
PROCEDURE RUN ;

VARIABLE A1 1 ; VARIABLE A2 1 ; VARIABLE A3 1 ;
VARIABLE A4 1 ; VARIABLE A5 1 ; VARIABLE A6 1 ;

OV 7 2 0 ;

A1 := 0.14389528689809122 ; A2 := 6.85939850586528;
A3 := -1.8709693628345963 ; A4 := 0.8015805270725156;
A5 := -0.4070432571162661 ; A6 := 0.06588881367640281;
```

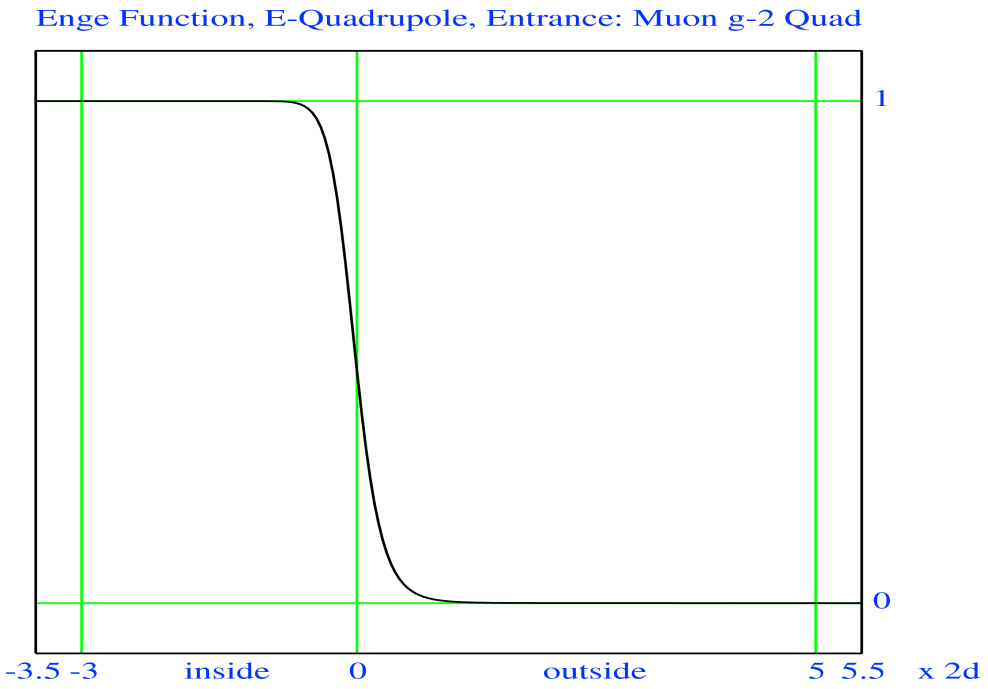
Coefficient	Value
a_1	0.14389528689809122
a_2	6.85939850586528
a_3	−1.8709693628345963
a_4	0.8015805270725156
a_5	−0.4070432571162661
a_6	0.06588881367640281

Table 3.9: Enge function coefficients for the falloff of the multipole strength $M_{2,2}$.

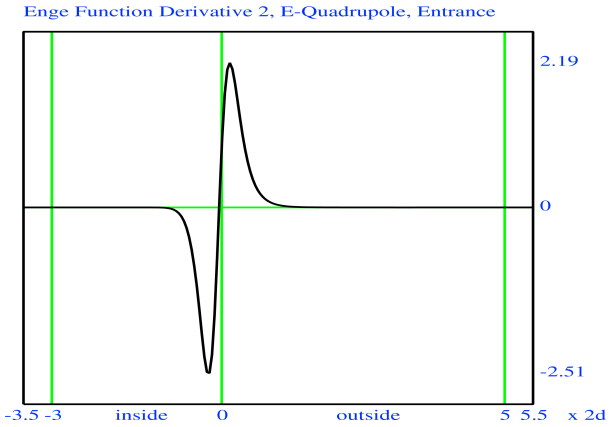
```
FC 2 1 2 A1 A2 A3 A4 A5 A6 ;

FP 2 1 2 'Muon g-2 Quad' 0 -10 ;
FP 2 1 2 'Muon g-2 Quad' 2 -10 ;
FP 2 1 2 'Muon g-2 Quad' 4 -10 ;

ENDPROCEDURE ;
RUN ;
END ;
```

3.3.8 Comparison with Wu Data

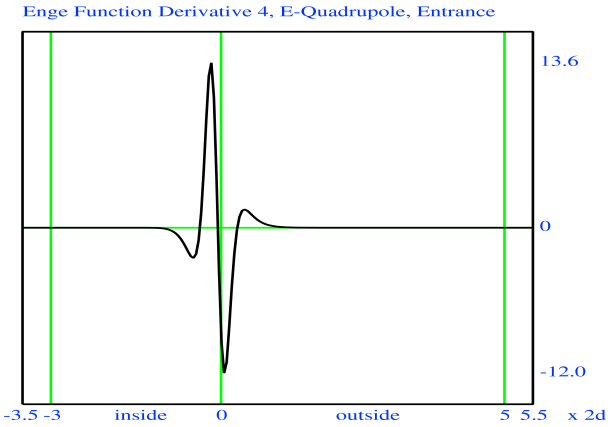
We performed the same calculations as in sec. 3.3.7 for electrostatic field data obtained using *Opera-3d*'s FEM field solver [1] by Wanwei Wu (FNAL and University of Mississippi) [116]. The model, shown in Fig. 3.34, approximates the full cage, chamber, and plates geometry of the Muon g-2 collaboration quadrupole. The plates span the polar angles 0° through 13°, measured from the center of curvature of the quadrupole, whereas the cage and the chamber have a longer length and span polar angles -5° through 20°. The field data comprises three parts: 3D data at the downstream (-1° to 1°) and upstream ends (12° to 14°), and, considering the approximate longitudinal translational symmetry of the main field, cross-sectional 2D data for the middle (1° to



(a) Enge function.



(b) Second derivative of the Enge function



(c) Fourth derivative of the Enge function.

Figure 3.33: Enge function and its 2nd and 4th derivatives, for the falloff of the multipole strength $M_{2,2}$, plotted using the `FP` command in *COSY INFINITY*.

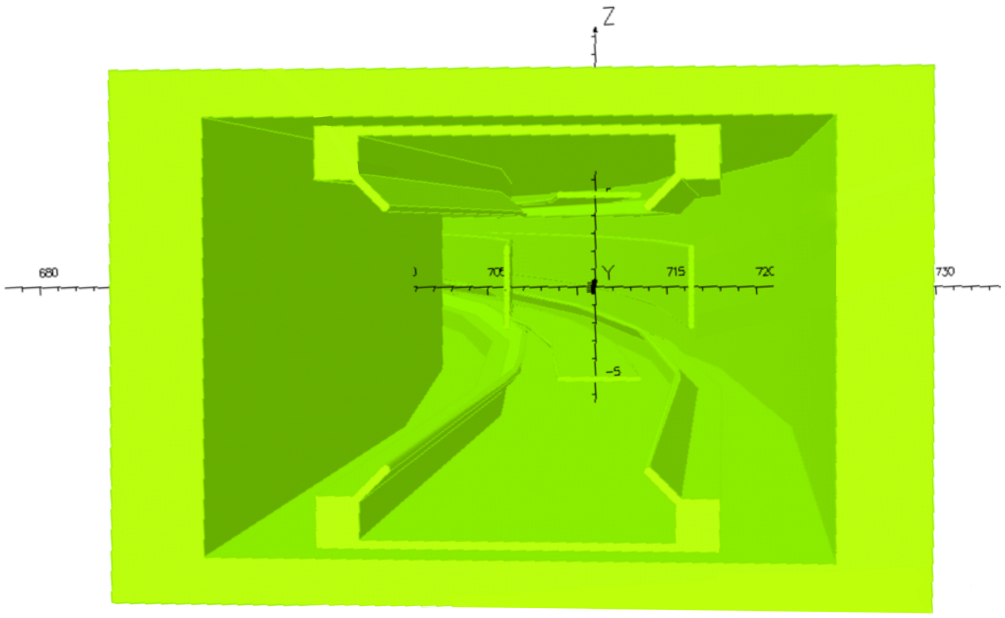


Figure 3.34: Muon g-2 collaboration high voltage quadrupole geometry used by Wu for field data calculations using *Opera-3d.* (Adapted by permission from Wanwei Wu, 3D Electric Field Map From Opera 3D, G-2 Experiment Document GM2-doc-4490, Muon g-2 Collaboration, Fermi National Accelerator Laboratory, Batavia, IL, Oct. 2016.)

12°). For the end parts, electrostatic field and potential data is provided for a grid-point set with 2 mm steps from 701.4 cm to 721.0 cm in radius r, 1 mm steps from -6.4 cm to 6.4 cm in vertical coordinate z, and 0.01° steps in polar angle θ.

For each cross section in the downstream field data set, we interpolated the field data and applied the multipole terms method as in secs. 3.3.2 and 3.3.7. Additionally, we fitted a nonlinear model defined as the multipole expansion to the raw field data in each cross section and obtained similar results.

Fig. 3.35 shows falloffs of 2nd order Fourier modes $a_2\left(r_j\right)$ calculated at radii

$$r = 1.8, 2.1, 2.4, 2.7, 3.0\,\text{cm}$$

from Wu's field data. We note that the 2nd order Fourier modes $a_2\left(r\right)$ alone fall off more quickly than the true quadrupole strength $M_{2,2}$, as Fig. 3.36 shows. This is because the second derivative of $M_{2,2}\left(s\right)$ is negative in the beginning of the fringe field and positive on the outside, impacting the additional terms based on the second derivative of $M_{2,2}$ in eq. 3.5.

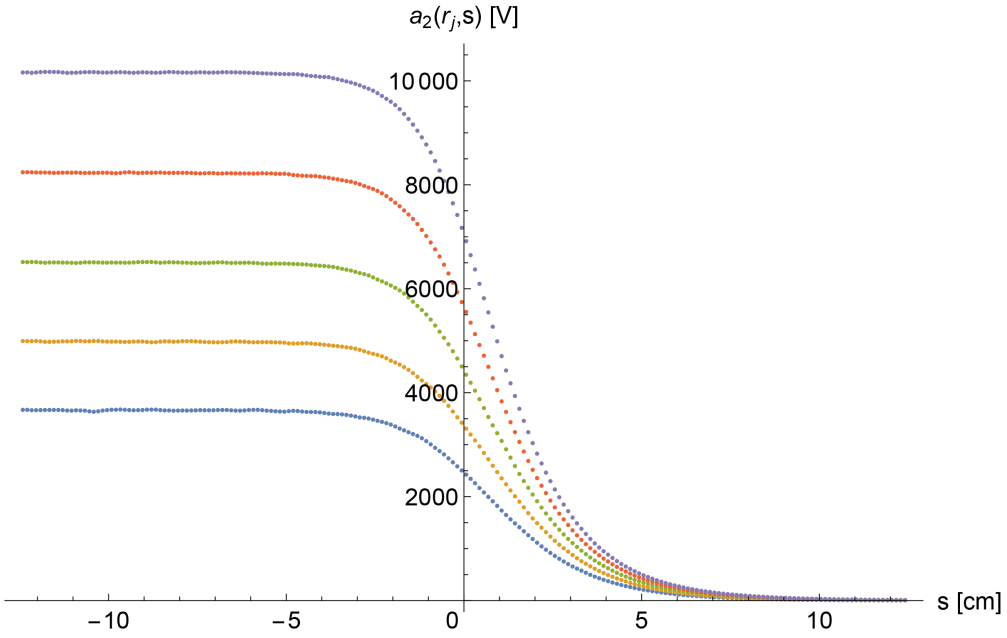


Figure 3.35: Falloffs of 2nd order Fourier modes $a_2\left(r_j\right)$ calculated at radii $r = 1.8$, 2.1, 2.4, 2.7, 3.0 cm from Wu's field data. Curves with larger magnitudes correspond to larger radii.

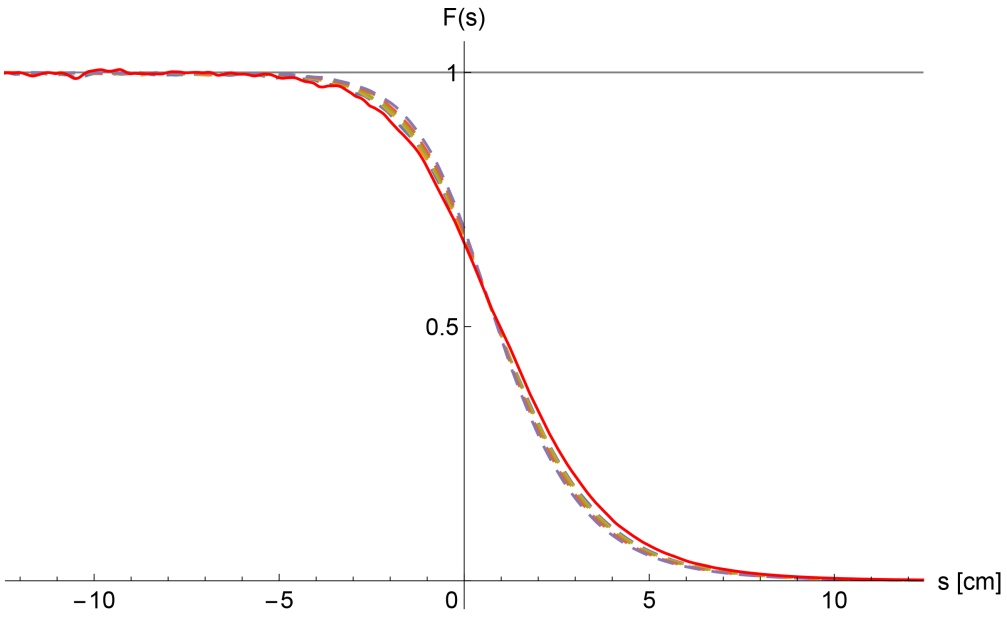


Figure 3.36: Fourier modes $a_2\left(r_j\right)$ (dashed plot style) versus the true quadrupole strength $M_{2,2}$ (solid red), all scaled to 1 well inside the quadrupole.

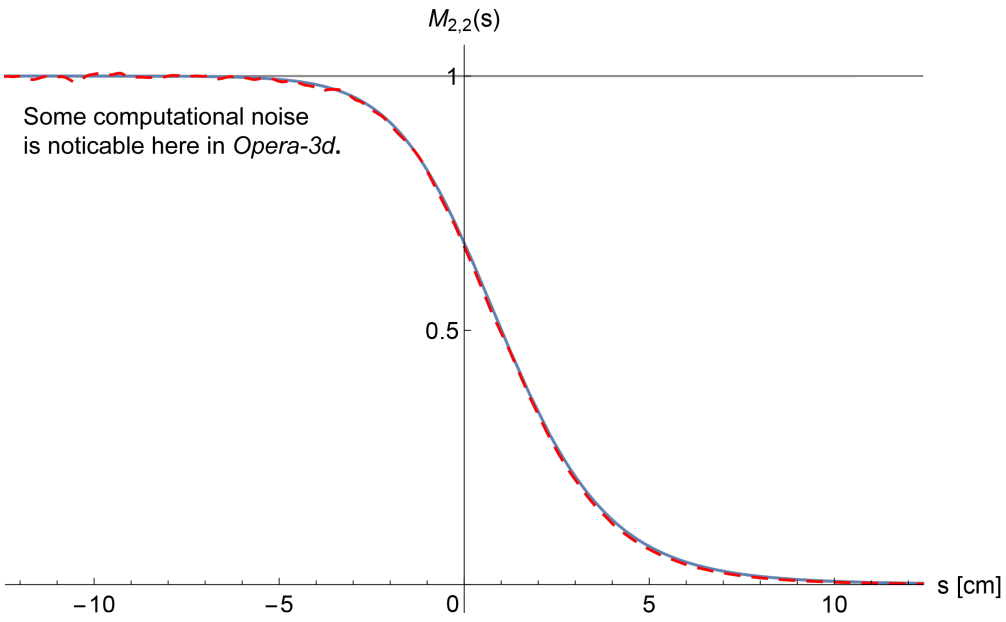


Figure 3.37: The falloff of the multipole term $M_{2,2}$ agrees well between calculations based on Soltner–Valetov field data ($z_{\mathrm{EFB}} = 1.2195\,\mathrm{cm}$; solid blue) and field data by Wu ($z_{\mathrm{EFB}} = 1.1233\,\mathrm{cm}$; dashed red).

Although the geometric models used for Soltner–Valetov and Wu field data calculations, in *COULOMB* and *Opera-3d*, respectively, differ a little in that in Soltner–Valetov data, the exterior cage is not considered and the element is assumed to be straight, there is a remarkable agreement in the field falloffs obtained from these data sets, as Fig. 3.37 shows.

The EFBs $z_{\mathrm{EFB}} = 1.2195\,\mathrm{cm}$ and $z_{\mathrm{EFB}} = 1.1233\,\mathrm{cm}$ obtained from Soltner–Valetov and Wu field data, respectively, agree quite well, and so do the tunes calculated based on them. The Muon g-2 ring linear tunes calculated based on Soltner–Valetov field data and the respective results of this section are listed in [71].

3.3.9 Fringe Field Conclusion

We obtained the quadrupole strength falloff and the EFB for the Muon g-2 collaboration quadrupole by calculating Fourier modes of its electrostatic potential at a set of radii in the transversal plane. The respective electrostatic potential data was obtained using *COULOMB*'s BEM field solver by Helmut Soltner from a 3D model of the quadrupole.

In the context of this study, we developed a code called `stepfg` to automatically generate a 3D model in STEP file format from cross-sectional geometry represented by a polygon. We used this code to produce a 3D model of the Muon g-2 collaboration quadrupole from its cross-sectional geometry.

For a confirmatory comparison, we applied the same method of calculating multipole strengths to the electrostatic field data obtained for the Muon g-2 collaboration quadrupole using *Opera-3d*'s FEM field solver by Wanwei Wu. The field falloffs and the EFBs obtained from Soltner–Valetov and Wu field data are in good agreement, validating the Muon g-2 ring tunes calculated based on the quadrupole EFB calculated from Soltner–Valetov field data.

CHAPTER 4

SYMPLECTIFICATION METHODS, AND SIMULTANEOUS SYMPLECTICITY AND ENERGY CONSERVATION

4.1 Symplecticity and Symplectification of Transfer Maps

In classical physical systems with coordinates $\vec{q}$ as the position and $\vec{p}$ as the conjugate momentum, if the Hamiltonian $H(\vec{q},\vec{p})$ exists, the equations of motion are [11, p. 22]

$$\frac{d}{dt}\begin{pmatrix}\vec{q}\\ \vec{p}\end{pmatrix} = \hat{J}\begin{pmatrix}\partial H/\partial\vec{q}\\ \partial H/\partial\vec{p}\end{pmatrix}.$$

The matrix $\hat{J}$ is defined as [11, p. 21]

$$\hat{J} = \begin{pmatrix}0 & I_n\\ -I_n & 0\end{pmatrix}, \tag{4.1}$$

where I_n is an $n\times n$ identity matrix and n is the phase space dimension.

Such systems are subject to the condition of symplecticity, which we introduce here. Deviation from symplecticity arises in calculations due to approximation and machine precision errors.

In *COSY INFINITY*, the phase space coordinates are grouped pair-wise as

$$z = \begin{pmatrix} x\\ a=\frac{p_x}{p_0}\\ y\\ b=\frac{p_y}{p_0}\\ l=-\frac{(t-t_0)v_o\gamma}{1+\gamma}\\ \delta_K=\frac{K-K_0}{K_0}\end{pmatrix},$$

where each canonical position $q_j = x$, b, or l is immediately followed by the respective conjugate momentum $p_j = a$, b, or δ_K. That is, the coordinates are ordered as

$$z = (q_1, p_1 \ldots, q_n, p_n),$$

where q_j is the j-th canonical position and p_j is the j-th conjugate momentum. We call coordinates arranged in this order **type I coordinates**.

However, for the purpose of studying symplecticity, it is useful to reorder the coordinates as

$$\tilde{z} = (q_1, \ldots, q_n, p_1, \ldots, p_n),$$

which, for *COSY INFINITY*'s beamline coordinate system, yields

$$\tilde{z} = (x, y, l, a, b, \delta_K).$$

We call coordinates arranged in this order **type II coordinates**.

A linear transfer matrix M is the Jacobian of the respective linear transfer map $\mathscr{M}$ [11, p. 37]:

$$M = \mathrm{Jac}(\mathscr{M}) = \begin{pmatrix} \partial\vec{Q}/\partial\vec{q} & \partial\vec{Q}/\partial\vec{p} \\ \partial\vec{P}/\partial\vec{q} & \partial\vec{P}/\partial\vec{p} \end{pmatrix},$$

where $\vec{q} = (q_1, \ldots, q_n)$ and $\vec{Q} = (Q_1, \ldots, Q_n)$ are the initial and final canonical position coordinates and $\vec{p} = (p_1, \ldots, p_n)$ and $\vec{P} = (P_1, \ldots, P_n)$ are the initial and final conjugate momentum coordinates, respectively.

Consider a linear transfer matrix M, which transforms the initial phase space coordinates vector x_i into final coordinates $z_\mathrm{f} = Mz_\mathrm{i}$.

The linear transfer matrix M is called **symplectic** if it satisfies the condition [11, p. 37]

$$M \cdot \hat{J} \cdot M^\mathrm{T} = \hat{J}. \tag{4.2}$$

There are two possibilities: either (1) the transfer matrix M is expressed in type **II** coordinates, and the matrix $\hat{J}$ is as defined in eq. 4.1; or (2) the transfer matrix M is expressed in type **I** coordinates; and the matrix $\hat{J}$ is transformed or defined accordingly.

In the general case of a nonlinear transfer map, symplecticity is defined by the same eq. 4.2, where M is the Jacobian $\mathrm{Jac}(\mathscr{M})$ of the transfer map $\mathscr{M}$. A transfer map $\mathscr{M}$ with a symplectic Jacobian $\mathrm{Jac}(\mathscr{M})$ is called symplectic.

4.1.1 Symplecticity Norm

We define the symplecticity function Ψ as

$$\Psi(M) = M \cdot \hat{J} \cdot M^t - \hat{J},$$

where M is the transfer map Jacobian.

Further, we consider a symplecticity norm for a transfer map Jacobian M as

$$\|\Psi(M)\| = \sum_{k,l=1}^{2n} \mathtt{ABS}\left[(\Psi(M))_{kl}\right],$$

where the summation reflects the standard matrix norm; and ABS is the max norm of the DA coefficients, as in *COSY INFINITY*.

4.1.1.1 Weighted Symplecticity Norm

In practical applications, we are interested in performing symplectification for a relevant region $\mathscr{E}$ in the phase space, somewhat larger than the beam envelope. Let this region be

$$\mathscr{E} = \left([-a_1, a_1], [-a_2, a_2], \ldots, [-a_{2n}, a_{2n}]\right).$$

In that case, we can use the weighted symplecticity norm

$$\|\Psi(M)\| = \sum_{k,l=1}^{2n} \mathtt{ABS}\left[(\Psi(M) \circ W)_{kl}\right],$$

where W is the linear weight function that transforms each independent variable x_j as

$$W\left(x_j\right) = a_j^{-1} x_j.$$

4.1.2 Symplectification Methods

Symplectification methods render the transfer map, or the motion they represent, symplectic, while attempting to make the changes as small as possible. Thus, whereas the exact transfer map is generally not known, the result is a symplectic transfer map that is close to the exact transfer map.

Symplecticity preserves phase space volume and is useful for approximate long-term tracking, as it tends to preserve radial information and to produce realistic-looking tracking pictures [11, p. 295].

Commonly used symplectification methods include symplectification using generating functions (including symplectification using Lie algebra methods), Gram–Schmidt symplectification, Miguel Furman's iterative symplectification, and Healy's symplectification [36, pp. 5331–5352][11, 66]. Additionally, we considered symplectification by applying the method of gradient descent to the weighted symplecticity norm introduced above.

4.1.2.1 Block-Wise Symplectification

If the transfer map Jacobian M is a block-diagonal matrix

$$M = \begin{pmatrix} a_1 & b_1 & & & \\ c_1 & d_1 & & 0 & \\ & & \ddots & & \\ & & & \ddots & \\ & 0 & & a_n & b_n \\ & & & c_n & d_n \end{pmatrix}$$

in type **I** coordinates, the block-diagonal form should be preserved by symplectification. Furthermore, symplectification can be performed for the blocks independently, which is essential if the nonsymplecticity of the transfer map is mostly in one phase space dimension, while being mostly symplectic in other phase space dimensions. Thus, if high precision in some phase space dimensions is already achieved, that precision is preserved.

If a transfer map represents stable motion, that is, if all tunes are real and mutually distinct, there exists a basis where the linear part of its Jacobian has a block-diagonal form [11, Ch. 7]. Block diagonalization may be performed by writing the Jacobian in its eigenvector basis, or by performing step-wise block diagonalization.

The symplecticity condition for a block-wise matrix is essentially that each block has a determinant equal to one.

4.1.2.2 The Gram–Schmidt Method

The Gram–Schmidt symplectification method is useful for symplectification of the linear part of a nonlinear transfer map Jacobian. Let $\vec{m}^j$ be the j-th column of the transfer map Jacobian, that is [36, p. 5350],

$$M = \left(\vec{m}^1, \ldots, \vec{m}^{2n}\right),$$

where n is the phase space dimension.

The symplecticity condition can be expressed through the columns $\vec{m}^j$ as

$$\left\langle \vec{m}^i, \vec{m}^j \right\rangle = J_{ij},$$

where $\langle , \rangle$ is the antisymmetric scalar

$$\left\langle \vec{m}^i, \vec{m}^j \right\rangle = \vec{m}^{i\mathrm{T}} J \vec{m}^j.$$

The symplectified transfer map Jacobian is given by the formula [36, pp. 5350–5351]

$$M_{\mathrm{S}} = \left(\frac{\vec{m}^1}{\left\langle \vec{m}^1, \vec{m}^2 \right\rangle^{\frac{1}{2}}}, \frac{\vec{m}^2}{\left\langle \vec{m}^1, \vec{m}^2 \right\rangle^{\frac{1}{2}}}, \cdots, \frac{\vec{m}^{2n-1}}{\left\langle \vec{m}^{2n-1}, \vec{m}^{2n} \right\rangle^{\frac{1}{2}}}, \frac{\vec{m}^{2n}}{\left\langle \vec{m}^{2n-1}, \vec{m}^{2n} \right\rangle^{\frac{1}{2}}} \right) \tag{4.3}$$

in type **I** coordinates.

In case the transfer map Jacobian can be brought to the block-diagonal form, the Gram–Schmidt symplectification method performs symplectification independently for each block.

Assuming that the original transfer map Jacobian is almost symplectic, the distance between the symplectified and the original transfer map Jacobians is almost zero.

The symplectified transfer map Jacobian preserves the block-diagonal structure, if one is present, as well as any other symmetries, as the formula in eq. 4.3 only divides each element of the matrix by the respective antisymmetric scalar. For a matrix, we define a symmetry as an element or elements of the matrix being equal to zero.

4.1.2.3 Miguel Furman's Iterative Method

Miguel Furman's iterative symplectification method [36, p. 5351][38] obtains an improved transfer map Jacobian

$$M_{j+1} = \left(1 - \frac{1}{2}E_j\right) M_j$$

in each iteration, where

$$E_j = -M_j \hat{J} M_j^{\mathrm{T}} \hat{J} - 1$$

and M_0 is the original transfer map Jacobian.

In case of a block-diagonal transfer map Jacobian, each element is only multiplied by a respective expression from E_j, thus preserving the block-diagonal structure. Similarly, the method preserves any other transfer map Jacobian symmetries.

4.1.2.4 The Gradient Descent Method

Let M_0 be an approximate transfer map Jacobian. We can use the method of gradient descent to converge to a local minimum M_{lm} of the weighted symplecticity norm $F(M) = \|\Psi(M)\|$. Since the transfer map Jacobian M_0 approximates an exact, symplectic transfer map Jacobian M_{exact}, there is at least one transfer map Jacobian in a small neighborhood of M_0 that is symplectic and solves the equation $F(M) = 0$.

Applying the gradient descent method to M_0, each successive approximation M_{j+1} is obtained from the previous approximation M_j as

$$M_{j+1} = M_j - \varepsilon_j \left.\frac{\partial F(M)}{\partial M}\right|_{M=M_j}, \tag{4.4}$$

where ε_j may be set using the line search algorithm or its modification.

Following the approach in [67, pp. 107–109], the matrices in eq. 4.4 can be viewed as vectors, including the interpretation

$$\frac{\partial F(M)}{\partial M} = \left(\frac{\partial F(M)}{\partial M}\right)_k = \\ = \frac{\partial F(\mathrm{vec}\, M)}{\partial (\mathrm{vec}\, M)_k},$$

where vec M is the vector representation of matrix M. Reverting back to matrix notation, we have

$$\frac{\partial F(M)}{\partial M} = \left(\frac{\partial F(M)}{\partial M}\right)_{kl} = \\ = \frac{\partial F(M)}{\partial M_{kl}}.$$

The derivative $\partial F(M)/\partial M$ can be found using DA differentiation or, alternatively, using numerical differentiation.

4.1.2.5 Symplectification by Generating Functions

Transfer maps are often symplectified using generating functions of canonical transformations. At least one such generating function exists for a transfer map Jacobian. This method requires the transfer map Jacobian to be invertible; however, symplectic transfer maps are always invertible, and this method is generally applicable if the linear part of the transfer map Jacobian is approximately symplectic.

The four generating functions $F_1(\vec{q}_\mathrm{i}, \vec{q}_\mathrm{f})$, $F_2(\vec{q}_\mathrm{i}, \vec{p}_\mathrm{f})$, $F_3(\vec{p}_\mathrm{i}, \vec{q}_\mathrm{f})$, and $F_4(\vec{p}_\mathrm{i}, \vec{p}_\mathrm{f})$ defined by [11, p. 164][10, p. 290]

$$\begin{aligned}
(\vec{p}_\mathrm{i}, \vec{p}_\mathrm{f}) &= \left(\vec{\nabla}_{\vec{q}_\mathrm{i}} F_1, -\vec{\nabla}_{\vec{q}_\mathrm{f}} F_1\right), \\
(\vec{p}_\mathrm{i}, \vec{q}_\mathrm{f}) &= \left(\vec{\nabla}_{\vec{q}_\mathrm{i}} F_2, \vec{\nabla}_{\vec{p}_\mathrm{f}} F_2\right), \\
(\vec{q}_\mathrm{i}, \vec{p}_\mathrm{f}) &= \left(-\vec{\nabla}_{\vec{p}_\mathrm{i}} F_3, -\vec{\nabla}_{\vec{q}_\mathrm{f}} F_3\right), \\
(\vec{q}_\mathrm{i}, \vec{q}_\mathrm{f}) &= \left(-\vec{\nabla}_{\vec{p}_\mathrm{i}} F_4, \vec{\nabla}_{\vec{p}_\mathrm{f}} F_4\right),
\end{aligned}$$

where $(\vec{q}_\mathrm{i}, \vec{p}_\mathrm{i})$ are the initial and $(\vec{q}_\mathrm{f}, \vec{p}_\mathrm{f})$ are the final canonical coordinates, are traditionally introduced in the theory of generating functions, and are commonly used to symplectify transfer maps. Additional generating functions can be constructed, *inter alia*, as combinations of the generating functions F_1–F_4.

To illustrate transfer map symplectification using generating functions, consider a transfer map $\mathscr{M}$, for which the generating function F_2 exists.

Let $\mathscr{Q}$ and $\mathscr{P}$ be the position and momentum parts of the symplectified version of the transfer map $\mathscr{M}$, respectively, and $\mathscr{I}_q$ and $\mathscr{I}_p$ the position and momentum parts of the identity map.

Then [11, p. 165][10, p. 292]

$$\begin{pmatrix} \vec{p}_\mathrm{i} \\ \vec{q}_\mathrm{f} \end{pmatrix} = \begin{pmatrix} \mathscr{I}_p \\ \mathscr{Q} \end{pmatrix} \circ \begin{pmatrix} \mathscr{P} \\ \mathscr{I}_q \end{pmatrix}^{-1} \begin{pmatrix} \vec{p}_\mathrm{f} \\ \vec{q}_\mathrm{i} \end{pmatrix}.$$

At the same time,

$$\begin{pmatrix} \vec{p}_\mathrm{i} \\ \vec{q}_\mathrm{f} \end{pmatrix} = \begin{pmatrix} \vec{\nabla}_{\vec{q}_\mathrm{i}} F_2 \left(\vec{q}_\mathrm{i}, \vec{p}_\mathrm{f} \right) \\ \vec{\nabla}_{\vec{p}_\mathrm{f}} F_2 \left(\vec{q}_\mathrm{i}, \vec{p}_\mathrm{f} \right) \end{pmatrix},$$

hence

$$\begin{pmatrix} \mathscr{I}_p \\ \mathscr{Q} \end{pmatrix} \circ \begin{pmatrix} \mathscr{P} \\ \mathscr{I}_q \end{pmatrix}^{-1} \begin{pmatrix} \vec{p}_\mathrm{f} \\ \vec{q}_\mathrm{i} \end{pmatrix} = \begin{pmatrix} \vec{\nabla}_{\vec{q}_\mathrm{i}} F_2 \left(\vec{q}_\mathrm{i}, \vec{p}_\mathrm{f} \right) \\ \vec{\nabla}_{\vec{p}_\mathrm{f}} F_2 \left(\vec{q}_\mathrm{i}, \vec{p}_\mathrm{f} \right) \end{pmatrix}. \tag{4.5}$$

The symplectification of the transfer map $\mathscr{M}$ is obtained by solving eq. 4.5 for

$$\begin{pmatrix} \mathscr{I}_p \\ \mathscr{Q} \end{pmatrix} \circ \begin{pmatrix} \mathscr{P} \\ \mathscr{I}_q \end{pmatrix}^{-1},$$

and further for $\mathscr{Q}$ and $\mathscr{P}$ [11, p. 165].

Alternatively to symplectifying the transfer map, a fixed-point iteration or a Newton iteration [11, p. 293] may be used to find the final canonical coordinates $(\vec{q}_\mathrm{f}, \vec{p}_\mathrm{f})$ that correspond to the initial canonical coordinates$(\vec{q}_\mathrm{i}, \vec{p}_\mathrm{i})$ and one of the generating functions, as in *COSY INFINITY*'s tracking procedure `TR` [12, pp. 41–42]. The initial coordinates $(\vec{q}_\mathrm{i}, \vec{p}_\mathrm{i})$ pushed by the non-symplectified transfer map should be used as the initial guess in the iterative method.

As shown in [33], the EXPO (Extended Poincaré) generating function provides an optimal symplectification by minimizing Hofer's metric, which is an "essentially unique intrinsic bi-invariant Finsler metric for compactly supported Hamiltonian symplectic maps" [33, p. 114302-1]. Hofer's metric is defined as the distance

$$\rho \left(\mathscr{M}, \mathscr{N} \right) = \inf_{\phi_0 = \mathscr{M}, \phi_1 = \mathscr{N}} \int_0^1 \| H_t \| \, dt$$

between the approximate transfer map $\mathscr{M}$ and its symplectification $\mathscr{N}$, where the H_t are the Hamiltonians generating the smooth paths $\{\phi_t\}, t \in [0,1]$ in $\mathrm{Ham}^{\mathrm{c}}\left(\mathbb{R}^{2n}\right)$ from $\phi_0 = \mathscr{M}$ to $\phi_1 = \mathscr{N}$; and $\|H_t\| = \sup H_t(z) - \inf H_t(z)$, $z \in \mathbb{R}^{2n}$ is the oscillation norm [33, p. 114302-2]. EXPO symplectification is usually the preferred symplectification method in *COSY INFINITY*.

4.2 Simultaneous Symplecticity and Energy Conservation

4.2.1 Feasibility of Simultaneous Symplecticity and Energy Conservation

For some time, it was an established opinion that it is not possible for tracking codes to simultaneously enforce symplecticity and energy conservation. This may be traced to [117], which stated that approximate symplectic algorithms cannot conserve energy for nonintegrable systems:

> Let H be a Hamiltonian which has no other conserved quantities (in a given class $\mathscr{K}$, for example analytic functions) other than functions of H. That is, if $\{F,H\} = 0$, then $F(z) = F_0(H(z))$ for a function F_0. Let $\Phi_{\triangle t}$ be an algorithm which is defined for small $\triangle t$ and is smooth. If this algorithm is symplectic and conserves H exactly, then it is the time advance map for the exact Hamiltonian system up to a reparametrization of time. In other words, approximate symplectic algorithms cannot preserve energy for nonintegrable systems.

Subsequently, it was shown [81, pp. 142–143][75, pp. 177–179][58] that approximate symplectic algorithms with variable time steps can preserve energy.

As an alternative that can be implemented with minor changes to an existing transfer map–based tracking code, we propose that simultaneous symplectification and energy correction can be applied to an already computed one-turn transfer map. This can be done using an iterative algorithm with alternating symplectification and energy correction steps. An algorithm acting on a one-turn transfer map is defined for a specific, relatively large time $\Delta t = T$, where T is the revolution period, and is not defined for times $\Delta t < T$. Thus, the statement from [117] does

not apply to such an algorithm up to the precision to which the iterative correction schemes are executed.

4.2.2 Concept of Iterative Symplectification and Energy Correction

Due to finite computation order and machine precision errors, the transfer map $\mathscr{M}$ computed in *COSY INFINITY*, while order-by-order symplectic up to machine precision, slightly differs from the respective exact transfer map $\mathscr{M}_{\text{exact}}$ and is generally not exactly symplectic.

Suppose, for the sake of argument, that the computed transfer map $\mathscr{M}$ slightly deviates from both symplecticity and energy conservation.

DA variables can be viewed as representations of Taylor expansions. Lagrange's form of the remainder of the multivariable Taylor series gives the truncation error a DA variable of computation order n as

$$\sum_{|\beta|=n+1} R_\beta(x)(x-y)^\beta,$$

with the uniform estimate

$$\left|R_\beta(x)\right| \leq \frac{1}{\beta!} \max_{|\beta|=n+1} \sup_{y\in B} \left|D^\beta f(y)\right| \forall x \in B,$$

where B is a closed ball, the function f approximated by the DA variable is of differentiability class C^{n+1} in B, and β is the multi-index defined by conventions $x^\alpha = \prod_{j=1}^{v} x_j^{\alpha_j}$, $\alpha! = \prod_{j=1}^{v} \alpha_j!$, and $|\alpha| = \sum_{j=1}^{v} \alpha_j$.

Thus, the transfer map $\mathscr{M}$ is approximately symplectic and energy conserving, and it is located in a small neighborhood of the exact symplectic and energy-conserving transfer map $\mathscr{M}_{\text{exact}}$. As the exact transfer map $\mathscr{M}_{\text{exact}}$ is generally unknown, a good symplectification or energy correction method will minimize the distance $d(A,\mathscr{M})$ or $d(B,\mathscr{M})$ between the approximate transfer map $\mathscr{M}$ and the set A of all symplectic maps or the set B of all energy-conserving maps, respectively. Same applies to symplectification or energy correction that acts on the final coordinates $z_{\text{f}} = \mathscr{M}(z_{\text{i}})$ instead of the transfer map $\mathscr{M}$ itself.

If an iterative method $z_{j+1} = T\left(z_j\right)$ in a metric space is contracting in the sense that, for all $j \in \mathbb{N}$,

$$d\left(T^2\left(z_{j+1}\right), T\left(z_j\right)\right) \leq kd\left(T\left(z_{j+1}\right), z_j\right), \tag{4.6}$$

where $k < 1$, then, utilizing parts of the classic proof [80, pp. 68–69] of the Banach fixed-point theorem, z_j a Cauchy sequence and converges to a fixed point z_{fp}.

An iterative symplectification method tends to converge to a fixed point z_{s} near the initial point z_0 in steps with decreasing sizes. With some care, we can assume the iterative symplectification method to be contracting in the sense of eq. 4.6. The same applies to an iterative energy correction method.

Consider iterative symplectification and energy correction methods $z_{j+1} = T_{\text{s}}\left(z_j\right)$ and $z_{j+1} = T_{\text{ec}}\left(z_j\right)$, respectively. Because symplectification and energy correction methods should achieve their objectives while minimizing other effects, we assume the symplectification and energy conservation methods to be effective in the sense that

$$d\left(\left(T_{\text{ec}} \circ T_{\text{s}}\right)^2\left(z_{j+1}\right), \left(T_{\text{ec}} \circ T_{\text{s}}\right)\left(z_j\right)\right) \leq \tilde{k}d\left(\left(T_{\text{ec}} \circ T_{\text{s}}\right)\left(z_{j+1}\right), z_j\right)$$

for some $\tilde{k} < 1$.

In that case, the combined iterative method $z_{j+1} = \left(T_{\text{ec}} \circ T_{\text{s}}\right)\left(z_j\right)$ with alternating application of T_{s} and T_{ec} converges to a fixed point z_{ecs}. Unless it happens that $T_{\text{s}}\left(z_{\text{ecs}}\right) = T_{\text{ec}}^{-1}\left(z_{\text{ecs}}\right)$ but $T_{\text{s}}\left(z_{\text{ecs}}\right) \neq z_{\text{ecs}}$, which is unlikely, we have $T_{\text{s}}\left(z_{\text{ecs}}\right) = z_{\text{ecs}}$, and the combined iterative method performs simultaneous symplectification and energy correction.

We note that, having made a number of assumptions, we only demonstrated that the simultaneous symplectification and energy correction method is likely to work and not that it would definitely work in all cases. However, as we will show below, we can achieve simultaneous symplecticity and energy conservation in *COSY INFINITY* without relying on these assumptions.

4.2.3 Energy Conservation in *COSY INFINITY*

The Hamiltonian of a relativistic charged particle is

$$\mathscr{H} = c\sqrt{m^2c^2 + (P - qA)^2} + q\varphi =$$
$$= E + q\varphi,$$

where m is the mass, P is the canonical momentum, q is the charge, and E is the relativistic energy of the particle. A is the magnetic vector potential, and φ is the electric scalar potential.

In *COSY INFINITY*'s beamline coordinate system, the relativistic energy of a charged particle is

$$E = mc^2 + mc^2\eta_0(1 + \delta_K) - q\varphi,$$

where $\eta_0 = K_0/mc^2$, δ_K is the particle's relative kinetic energy offset coordinate $\delta_K = (K - K_0)/K_0$ (in zero electrostatic potential), and K_0 is the reference kinetic energy.

Hence, the Hamiltonian is expressed simply as

$$\mathscr{H} = mc^2 + mc^2\eta_0(1 + \delta_K),$$

and

$$\Phi(\delta_K) = \left|\delta_K - \delta_K^0\right|$$

may be used as the energy conservation measure.

To enforce energy conservation in *COSY INFINITY*, as well as in our in-house eighth order Runge–Kutta–Verner tracking code *MSURK89* (see sec. 5.6), which uses *COSY INFINITY*'s beamline coordinate system, it is sufficient to perform calculations only in the x–a and y–b planes, without calculations in the l–δ_K plane. This conserves energy during both transfer map integration and tracking, with $\Phi(\delta_K) = 0$. Tracking in the l–δ_K plane is not necessary if one wishes to enforce energy conservation when tracking an energy-conserving lattice.

4.3 Conclusion

We introduced the concept of symplecticity, described several symplectification methods, and identified the role of symplecticity in long-term tracking.

Whereas there was an established opinion that approximate symplectic algorithms cannot conserve energy, we cited the possibility of that with variable time steps and suggested that enforcing symplecticity and energy conservation on a one-turn map is an alternative. In particular, an iterative algorithm with alternating symplectification and energy correction steps can be used.

In *COSY INFINITY* and *MSURK89*, energy conservation is equivalent to the δ_K coordinate being constant. Thus, if necessary, energy conservation may be achieved by performing calculations in the x–a and y–b planes but not in the l–δ_K plane. This conserves energy both at transfer map integration time and during the transfer map tracking. For simultaneous energy conservation and symplecticity, it is then sufficient only to perform symplectification.

In case an RF cavity is used for energy averaging, calculations in the l–δ_K plane are necessary to simulate the off-energy effects; however, energy errors are also subject to the RF energy averaging.

CHAPTER 5

ORBITAL TRACKING USING *COSY INFINITY* AND A CONVENTIONAL RUNGE–KUTTA INTEGRATOR

5.1 Introduction

Storage rings for sub-ppm measurement of EDM and anomalous MDM present a new frontier in tracking code accuracy requirements [77, 51, 65].

In [77], considering that many tracking codes ignore or inaccurately account for nonlinear effects, Metodiev *et al*. proposed to benchmark tracking code accuracy by tracking a homogeneous magnetic dipole and an electrostatic cylindrical deflector.

Particle orbits in electrostatic spherical deflectors and homogeneous magnetic dipoles are known in analytic form. In these cases, except for relativistic motion through electrostatic spherical deflectors, the orbital transfer map is identity, that is, particles return to the same position after each turn.

Tracking particles through electrostatic spherical deflectors and homogeneous magnetic dipoles is a simple benchmark for tracking code integrators and particle optical element representations. Despite the simplicity, this benchmark is effective in assessing tracking code limitations, which already become apparent at this level of complexity.

Here, we will perform this benchmark for *COSY INFINITY* and *MSURK89*. *MSURK89* is an in-house eighth order Runge–Kutta–Verner tracking code we developed in the context of this study.

Furthermore, we will describe and perform the direct computation of the DA transfer map of an electrostatic spherical deflector in a laboratory coordinate system (LCS) using simple ODEs and Kepler's theory. We will compare the results with transfer maps of the electrostatic spherical deflector obtained using the built-in elements of *COSY INFINITY* and *GIOS* [114].

Finally, we will use *MSURK89* and *COSY INFINITY* to analyze, model, and track a variant [78] of the Proton EDM Long Spin Coherence Time (LSCT) lattice [3].

5.2 Bending Elements

5.2.1 Homogeneous Magnetic Dipole

A homogeneous magnetic dipole is defined as a particle optical element with homogeneous magnetic field B_y in the vertical y direction.

The reference particle trajectory is in the horizontal x–s plane, and the reference particle moves along an arc of radius

$$R_0 = \frac{p}{ZeB_y},$$

where p is the particle momentum and Z is the particle charge in terms of the elementary charge e. Because the magnetic rigidity $\chi_{\mathrm{m}} = \frac{p}{Ze}$ and the curvature $h = 1/R_0$ of the reference orbit often have a precedence as known or specified quantities in lattice design, including in *COSY INFINITY*, we express the magnetic field as

$$\left(B_x, B_y, B_z\right) = \left(0, h\chi_{\mathrm{m}}, 0\right).$$

5.2.2 Electrostatic Spherical Deflector

We consider an electrostatic spherical deflector with inner and outer shells of radii R_1 and R_2, respectively, such that the reference particle orbit is a circular orbit of radius R_0. For the relevance of the analysis, we assume that the motion, including the reference trajectory, is inside the deflector (at $R_1 < r < R_2$).

A particle with beamline coordinates (x, y) is at radius

$$r = \sqrt{(R_0 + x)^2 + y^2}.$$

By Gauss's law, considering the spherical symmetry, the electrostatic field of the electrostatic deflector has the magnitude

$$E(r) = \frac{Q}{4\pi\varepsilon_0 r^2},$$

where Q is the charge of the deflector's inner shell, and is aligned with the radius vector. For the direction of particle beam deflection to agree with the electrostatic spherical deflector's geometry,

charges Q and Ze must be of the opposite sign, where Ze is the charge of a particle in the particle beam.

Hence, we obtain the electrostatic potential inside the spherical deflector as

$$\varphi(r) = \frac{Q}{4\pi\varepsilon_0 r} + \text{const.}$$

The voltage of the electrostatic spherical deflector is

$$\begin{aligned} |\Delta V| &= |\varphi(R_2) - \varphi(R_1)| = \\ &= \frac{|Q|}{4\pi\varepsilon_0}\frac{R_2 - R_1}{R_1 R_2}. \end{aligned}$$

The electrostatic field and potential can be expressed in terms of the voltage using

$$\frac{|Q|}{4\pi\varepsilon_0} = |\Delta V|\frac{R_1 R_2}{R_2 - R_1}.$$

Let E_0 be the radial component $E_r(R_0)$ of the electrostatic field at radius $r = R_0$, which is, by Newton's second law, $E_0 = -h\chi_e$, where χ_e is the electric rigidity $\chi_e = \frac{pv}{Ze}$.

The radial component of the electrostatic field at any radius $r < R_S$ can then be expressed as

$$E_r(r) = E_0\frac{R_0^2}{r^2}. \tag{5.1}$$

In vector form, the electrostatic field is

$$(E_x, E_y, E_s) = E_0\frac{R_0^2}{R^2}\left(\frac{R_0 + x}{R}, \frac{y}{R}, 0\right).$$

Following the convention of electrostatic potential as zero at the reference orbit, the electrostatic potential is

$$\begin{aligned} \varphi(r) &= \frac{Q}{4\pi\varepsilon_0}\left(\frac{1}{r} - \frac{1}{R_0}\right) = \\ &= E_0 R_0^2\left(\frac{1}{r} - \frac{1}{R_0}\right). \end{aligned}$$

5.2.2.1 Representations in *COSY INFINITY*

There is a specialized particle optical element `ESP` in *COSY INFINITY* to represent an electrostatic spherical deflector. For small beam envelopes, the inhomogeneous combined function electrostatic deflector element `ES` can be alternatively used as an approximation.

The inhomogeneity indices in ES are specified by a 5-vector n, defined by the following expansion of electrostatic field $E(x)$ in the x–s plane [12, p. 19]:

$$E(x) = E_0 \left(1 - \sum_{k=1}^{5} n_i \left(\frac{x}{R_0} \right)^k \right) + O\left(x^6\right), \tag{5.2}$$

where E_0 is electrostatic field at the reference orbit and R_0 is the curvature radius. From eq. 5.1, we obtain for the electrostatic spherical deflector the expansion

$$E(x) = E_0 \left(1 - \sum_{k=1}^{5} (-1)^{k+1} (k+1) \left(\frac{x}{R_0} \right)^k \right) + O\left(x^6\right),$$

resulting in the inhomogeneity coefficients

$$n = (2, -3, 4, -5, 6).$$

For comparison, the inhomogeneity coefficients of an electric cylindrical deflector can be similarly obtained as $n = (1, -1, 1, -1, 1)$ [12, p. 19].

Additionally, *COSY INFINITY* has the inhomogeneous combined function electrostatic element EC. The EC element differs from ES by taking the inhomogeneity indices as a vector of any dimension and using the resulting expansion of electrostatic field $E(x)$ to the respective order.

5.3 Off-Midplane Terms of the Potential

In many cases, the electrostatic potential φ is known in the midplane, e.g., as an expansion $\varphi(x)|_{y=0}$ arising from eq. 5.2. Then, $\varphi|_{y=0}$ may be used as a boundary condition for the Laplace equation to obtain off-midplane terms of the electrostatic potential.

To that end, we assert the general Laplace equation

$$\triangle\varphi = a_1 \frac{\partial}{\partial x}\left(a_2 \frac{\partial \varphi}{\partial x} \right) + b_1 \frac{\partial}{\partial x}\left(b_2 \frac{\partial \varphi}{\partial x} \right) + c_1 \frac{\partial}{\partial x}\left(c_2 \frac{\partial \varphi}{\partial x} \right) = 0$$

in the fixed-point form [70] as

$$\begin{aligned} \varphi = F(\varphi) = \varphi|_{y=0} + \int_0^y \frac{1}{b_2} \left[\left(b_2 \frac{\partial \varphi}{\partial y} \right) \Big|_{y=0} \right] dy - \\ - \int_0^y \frac{1}{b_2} \int_0^y \left(\frac{a_1}{b_1} \frac{\partial}{\partial x}\left(a_2 \frac{\partial \varphi}{\partial x} \right) + \frac{c_1}{b_1} \frac{\partial}{\partial z}\left(c_2 \frac{\partial \varphi}{\partial z} \right) \right) dydy, \end{aligned} \tag{5.3}$$

where a_i, b_i, and c_i are functions of x, y, and z determined by the coordinate system.

The boundary conditions $\varphi|_{y=0}$ and $\left(b_2 \frac{\partial \varphi}{\partial y}\right)\Big|_{y=0}$ are set as DA values.

Eq. 5.3 satisfies the DA fixed-point theorem, and the sequence $\varphi_{k+1} = F(\varphi_k)$, for $k = 0, 1, \ldots$, converges to a fixed point φ – the electrostatic potential with off-midplane terms – in a finite number of steps [70][11, pp. 96–100].

5.4 Equations of Motion

Unless otherwise stated, we use equations of motion in *COSY INFINITY*'s beamline coordinate system. Here, we will list the relativistic and non-relativistic equations of motion in this beamline coordinate system.

The standard distribution of *COSY INFINITY* includes `COSY.FOX`, its advanced particle beam dynamics simulation package, which uses relativistic equations of motion. In the context of this study, we developed a non-relativistic version `COSYNR.FOX` of `COSY.FOX`. In *MSURK89*, we implemented both relativistic and non-relativistic equations of motion, and the user can switch between them by toggling the variable `RELATIVISTIC`.

As and where applicable, we use appropriate relativistic or non-relativistic expressions for physical quantities such as kinetic energy and momentum.

Quantity name	Relativistic	Non-relativistic
Kinetic energy	$K = (\gamma - 1) mc^2$	$K = \frac{p^2}{2m}$
Momentum	$p = \gamma m v$	$p = mv$

Let h be the curvature of the reference orbit, $\eta = [K_0 (1 + \delta_K) - ZeV] / mc^2$ a relativistic measure, p the particle momentum, and $\zeta = p_s / p_0$. We denote electric and magnetic rigidities by $\chi_\mathrm{e} = pv/Ze$ and $\chi_\mathrm{m} = p/Ze$, respectively. The index 0 refers to the reference particle. We use the convention of the electrostatic potential V as zero for the reference particle. Hence, $\eta_0 = K_0 / mc^2$.

The relativistic equations of motion are as follows [69, p. 1].

$$
\begin{aligned}
x' &= a\,(1+hx)\,\tfrac{1}{\zeta}, \quad & a' &= (1+hx)\left(\tfrac{1+\eta}{1+\eta_0}\,\tfrac{1}{\zeta}\,\tfrac{E_x}{\chi_{e0}} - \tfrac{B_y}{\chi_{m0}} + b\,\tfrac{1}{\zeta}\,\tfrac{B_s}{\chi_{m0}}\right) + h\zeta, \\
y' &= b\,(1+hx)\,\tfrac{1}{\zeta}, \quad & b' &= (1+hx)\left(\tfrac{1+\eta}{1+\eta_0}\,\tfrac{1}{\zeta}\,\tfrac{E_y}{\chi_{e0}} + \tfrac{B_x}{\chi_{m0}} - a\,\tfrac{1}{\zeta}\,\tfrac{B_s}{\chi_{m0}}\right), \\
\delta_K' &= 0, \quad & l' &= -\tfrac{\gamma}{1+\gamma}\left[(1+hx)\,\tfrac{1+\eta}{1+\eta_0}\,\tfrac{1}{\zeta} - 1\right].
\end{aligned}
$$

Expressing the magnitude p of momentum through the vector components and dividing by p_0, we obtain

$$
\zeta = \frac{p_s}{p_0} = \sqrt{\left(\frac{p}{p_0}\right)^2 - \left(\frac{p_x}{p_0}\right)^2 - \left(\frac{p_y}{p_0}\right)^2} = \sqrt{\left(\frac{p}{p_0}\right)^2 - a^2 - b^2}.
$$

Since $E = (1+\eta)\,mc^2$, we have $E^2 = \left(1+2\eta+\eta^2\right)m^2c^4$, and, thus, $p = mc\sqrt{\eta\,(\eta+2)}$. Therefore,

$$
\zeta = \frac{p_s}{p_0} = \sqrt{\frac{\eta\,(\eta+2)}{\eta_0\,(\eta_0+2)} - a^2 - b^2}. \tag{5.4}
$$

In the non-relativistic limit, we have $\eta \to 0$, and

$$
\lim_{\gamma \to 1} \frac{\eta + \alpha}{\eta_0 + \alpha} = 1
$$

for any constant $\alpha \neq 0$.

Thus, the non-relativistic limit of 5.4 is

$$
\zeta = \frac{p_s}{p_0} = \sqrt{\frac{\eta}{\eta_0} - a^2 - b^2}. \tag{5.5}
$$

Similarly, the non-relativistic version of the above relativistic equations of motion is as follows, with non-relativistic ζ from eq. 5.5.

$$
\begin{aligned}
x' &= a\,(1+hx)\,\tfrac{1}{\zeta}, \quad & a' &= (1+hx)\left(\tfrac{1}{\zeta}\,\tfrac{E_x}{\chi_{e0}} - \tfrac{B_y}{\chi_{m0}} + b\,\tfrac{1}{\zeta}\,\tfrac{B_s}{\chi_{m0}}\right) + h\zeta, \\
y' &= b\,(1+hx)\,\tfrac{1}{\zeta}, \quad & b' &= (1+hx)\left(\tfrac{1}{\zeta}\,\tfrac{E_y}{\chi_{e0}} + \tfrac{B_x}{\chi_{m0}} - a\,\tfrac{1}{\zeta}\,\tfrac{B_s}{\chi_{m0}}\right), \\
\delta_K' &= 0, \quad & l' &= -\tfrac{1}{2}\left[(1+hx)\,\tfrac{1}{\zeta} - 1\right].
\end{aligned}
$$

5.5 Tracking in *COSY INFINITY*

In *COSY INFINITY*, the equations of motion are integrated, once for each particle optical element comprising the lattice, in phase space represented as a vector space $({}_nD_v)^v$ over a differential

algebra ${}_nD_v$ [11, pp. 86–100], where n is the computation order and v is the number of phase space coordinates. The result is a transfer map $\mathscr{M}$ that expresses the final coordinates z_f of any ray as $z_\mathrm{f} = \mathscr{M}(z_\mathrm{i})$, where z_i are the initial coordinates [11, Chs. 4–5].

The phase space dimension is set by the command `OV`, where the phase space dimension argument `N` specifies that motions $(x\text{–}a, y\text{–}b, l\text{–}\delta_K)_j$ for $1 \leq j \leq \mathtt{N}$ should be computed [12, p. 9].

The equation $z_\mathrm{f} = \mathscr{M}(z_\mathrm{i})$ can be viewed as a multi-variable Taylor expansion of vector z_f to the order n in terms of vector z_i. Respectively, the accuracy of computation depends on the computation order n.

The procedure `TR` performs tracking, in essence, by repetitively applying the transfer map to a set of particle rays. Every specified number `NR` of turns up to the specified total number `N` of turns, `TR` draws projections of ray positions on the specified phase plane [12, p. 41]. There are several symplectification methods in *COSY INFINITY*, with a choice of the four conventional generating functions F_1, F_2, F_3, F_4 [11, pp. 37–59] and the EXPO generating function [102]. Here, we will use the EXPO generating function, which results in the most accurate symplectification.

5.6 *MSURK89* Tracking Code

We developed an in-house tracking code, *MSURK89*, which integrates the equations of motion using an 8th order Runge–Kutta–Verner method ("RK89") [111] in the phase space represented as a vector space $\mathbb{R}^v$ over $\mathbb{R}$, where v is the number of phase space coordinates.

As in *COSY INFINITY*, the phase space dimension in *MSURK89* comprises three dimensions: namely, the x–a, y–b, and l–δ_K planes.

In *MSURK89*, particle trajectories are individually integrated (as a vector, for computational efficiency) in every tracking turn. This has accuracy advantages, especially for rays far from the reference orbit, but has the disadvantage of a substantially slower tracking speed.

Tracking is performed using the `MITR` procedure, which is analogous to *COSY INFINITY*'s tracking procedure `TR`, by repetitively applying one-turn integration of the equations of motion to a set of particle rays. Procedure `MITR` produces phase space plots, with particle ray coordinates drawn

every specified number of turns.

Symplectification functionality was not implemented in *MSURK89*.

Particle rays are set by calling the procedure `SETRAY`, an analogue of *COSY INFINITY*'s ray-setting procedure `SR`.

5.6.1 Particle Optical Elements

We implemented several particle optical elements in *MSURK89*, including a homogeneous magnetic dipole `MIDI`, an electrostatic spherical deflector `MIESP`, and a drift `MIDL`, analogous to *COSY INFINITY*'s elements `DI`, `ESP`, and `DL`, respectively. We also implemented the command `MITA` – a tilt of momentum, analogous to *COSY INFINITY*'s tilt of the optic axis `TA`.

5.6.1.1 Homogeneous Magnetic Dipole `MIDI`

A homogeneous magnetic dipole in *MSURK89* is computed using the command "`MIDI` R_0 α", where radius R_0 is in meters and angle α is in degrees.

The path length of the reference particle through the element is $\triangle s = \left(\frac{2\pi}{360^\circ}\right) R_0 \alpha$.

The interior electrostatic field and potential are $\left(E_x, E_y, E_s\right) = (0,0,0)$ and $\varphi = 0$, respectively. The interior magnetic field is

$$\left(B_x, B_y, B_s\right) = (0, h\chi_m, 0)\,.$$

5.6.1.2 Electrostatic Spherical Deflector `MIESP`

An electrostatic deflector in *MSURK89* is computed using the command "`MIESP` R_0 α", where radius R_0 is in meters and angle α is in degrees.

The path length of the reference particle through the element is $\triangle s = \left(\frac{2\pi}{360^\circ}\right) R_0 \alpha$.

The interior electrostatic field at point (x, y, s) is

$$\left(E_x, E_y, E_s\right) = \left(\frac{R_0 + x}{R_0} E, \frac{y}{R_0} E, 0\right),$$

where $E = E_0 R_0^2 / r^2$ is the magnitude of the electrostatic field and

$$r = \sqrt{(R_0 + x)^2 + y^2}$$

is the distance from the center of the electrostatic spherical deflector. The interior electrostatic potential is

$$\varphi = E_0 R_0^2 \left(\frac{1}{r} - \frac{1}{R_0} \right),$$

and the interior magnetic field is $\left(B_x, B_y, B_s\right) = (0,0,0)$.

For any angle α (corresponding to a sector of the full geometry of an electrostatic spherical deflector), the electrostatic field within a `MIESP` element is the same as it would be if the angle were 360° (full electrostatic spherical deflector geometry). Of course, an actual standalone sector of an electrostatic sphere deflector, e.g., in free space, would have a different proper electrostatic field. In that sense, if $\alpha/360^\circ$ is not an integer, `MIESP` is an artificial element, unless (1) a system of `MIESP` elements of the same radius and the total angle 360° or (2) some method of particle ray injection and extraction is assumed or implemented.

5.6.1.3 Drift `MIDL`

A drift in *MSURK89* is computed using the command "`MIDL L`", where length L is in meters.

The path length of the reference particle through the element is $\triangle s = L$.

The interior electrostatic field, electrostatic potential, and magnetic field of the element are all zero: $E = 0$, $\varphi = 0$, and $B = 0$, respectively.

5.6.1.4 Momentum Tilt `MITA`

In *MSURK89*, the momentum can be tilted using the command "`MITA` α_x α_y", by angle $-\alpha_x$ in the x direction and angle $-\alpha_y$ in the y direction, specified in degrees.

A positive angle results in momentum tilt in the negative direction of the respective transversal axis in `MITA`. This is consistent with a positive angle resulting in optic axis tilt in the positive direction of the respective transversal axis by *COSY INFINITY*'s axis tilt procedure `TA`.

The coordinate transformation equations are

$$\begin{aligned} a' &= (a\cos(\alpha_x) - \zeta\,\sin(\alpha_x))\cos(\alpha_y) + b\sin(\alpha_y), \\ b' &= b\cos(\alpha_y) - \sqrt{\zeta^2 + a^2}\,\sin(\alpha_y), \end{aligned}$$

where $\zeta = p_s/p_0$, p_s is the s component of momentum p, and p_0 is the momentum of the reference particle.

5.6.2 Integration Step Size

5.6.2.1 Automatic Step-Size Adjustment

The integrator RK89 in *MSURK89* has an automatic step-size adjustment functionality. In case the integration interval $[s_1, s_2]$ is known *ab initio*, RK89 trivially sets the length of the last integration sub-interval to end the integration at exactly s_2.

However, when s_2 is determined in the course of integration by reaching a boundary, e.g., the particle trajectory reaching an edge of the particle optical element, an iterative process is used to set the length of the last integration sub-interval to ensure the integration ends at that boundary.

In a lattice consisting of 16 particle optical elements as 22.5° sectors of the Proton EDM LSCT lattice from [78] (see below), the reference orbit was *ab initio* unknown, and, therefore, the length of the integration interval was *ab initio* unknown. For that case, we implemented a "G-stop"[1] function `TF` that returns the distance to the nearest edge of the element. When the `TF` value becomes smaller than the current integration step size, the step size is optimized to stop the integration precisely at the element boundary.

5.6.2.2 Initial Step Size

Decreasing the integration step size improves numerical integration accuracy until this effect is overtaken by the loss of significance in floating-point operations when step size becomes excessively

[1] The term "G-stop" for such functions traces back to JPL's numerical integrator program `DVDQ` [61], written in *FORTRAN* **IV** and **V**.

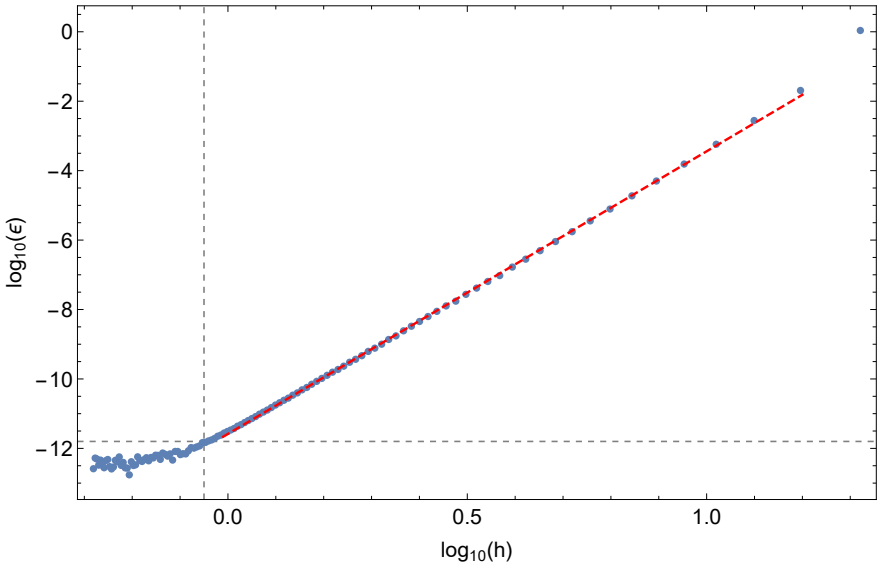


Figure 5.1: RK89 numerical integration error ε versus initial step size h, in a 100-turn integration of a radius $R = 10.1\,\mathrm{m}$ circular orbit inside an electrostatic spherical deflector. Non-relativistic equations of motion were used. The dashed red line of slope 8.14 is a least-squares fit to the linear pattern. The gridlines separate the linear regime from the part dominated by the floating-point loss of significance, corresponding to initial step size $h \simeq 1\,\mathrm{m}$.

small.

Whether or not there is an automatic step-size adjustment, it is desirable to study the effect of fixed or initial step size on calculation accuracy, as applicable. It is good practice to optimize step size using an integrand that is reasonably typical for the category of problems to be solved.

Non-relativistic Kepler orbits in a gravitational or electrostatic field, as well as circular orbits in a magnetic field, are closed and analytically known trajectories and are useful for testing integrators and calibrating integrator step sizes.

To calibrate the initial step size in RK89, we integrated a circular Kepler orbit of radius $R = 10.1\,\mathrm{m}$ (as $\sim 10\,\mathrm{m}$ is a quite realistic radius of curvature for a lattice; the initial particle position was $(x,a,y,b,l,\delta_K)_\mathrm{i} = (0.1\,\mathrm{m},0,0,0,0,0)$ with the reference orbit at $R_0 = 10\,\mathrm{m}$), through a spherical electrostatic deflector for 100 full turns. The distance

$$\varepsilon = \|(x,a,y,b,l,\delta_K)_\mathrm{f} - (x,a,y,b,l,\delta_K)_\mathrm{i}\|_2 \tag{5.6}$$

between the initial and final positions in the six phase space coordinates was used as the error function. Calculation results for a range of initial step sizes are shown in Fig. 5.1 and indicate a

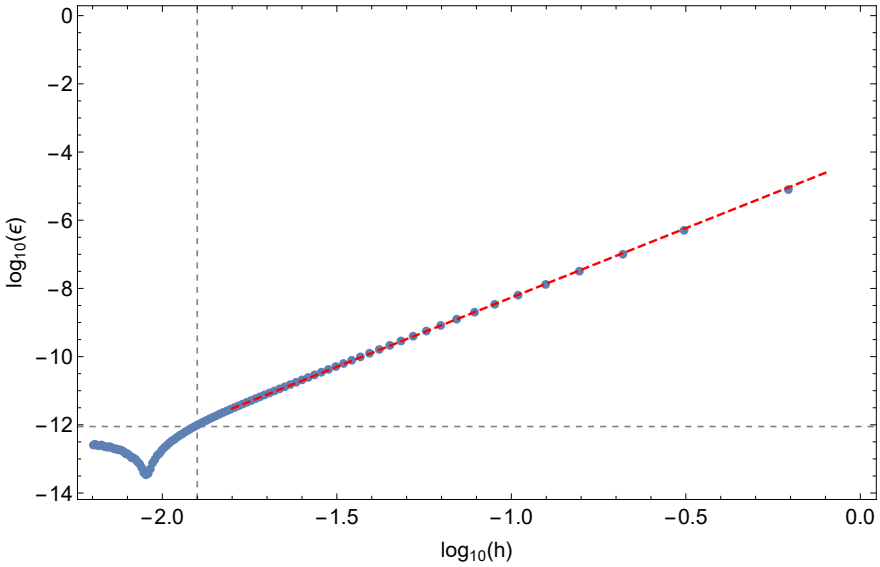


Figure 5.2: RK4 numerical integration error ε versus initial step size h, in a 100-turn integration of a radius $R = 10.1\,\mathrm{m}$ circular orbit inside an electrostatic spherical deflector. Non-relativistic equations of motion were used. The dashed red line of slope 4.07 is a least-squares fit to the linear pattern. The gridlines separate the linear regime from the part dominated by the floating-point loss of significance, corresponding to step size $h \simeq 10^{-2}\,\mathrm{m}$.

reasonable initial step size of $h \simeq 1\,\mathrm{m}$. For comparison, see the results in Fig. 5.2 for the same procedure applied to a 4th order Runge–Kutta integrator ("RK4") with fixed step size, with a resulting reasonable step size of $h \simeq 10^{-2}\,\mathrm{m}$.

The particle tracked for the initial step size calibration had the kinetic energy 1 MeV, mass 1 amu, and charge $1\,e$ (in *COSY INFINITY*, this may be set by the command `RP 1 1 1`). However, changing the kinetic energy to, e.g., 500 MeV, did not substantially affect the calibration results. This makes sense, as (1) the element `MIESP` is specified by the reference orbit geometry and (2) the equations of motion are expressed in terms of arc length s as the independent variable. Thus, changing the kinetic energy did not affect the non-relativistic Kepler orbit in the beamline coordinate system.

Replacing the electrostatic deflector by a magnetic dipole in the step size calibration procedure yielded practically identical results. This illustrates that the initial step size of $h \simeq 1\,\mathrm{m}$ is suitable for integration of both electrostatic and magnetic elements.

At the initial step size $h = 1\,\mathrm{m}$ in RK89 and the step size $h = 0.01\,\mathrm{m}$ in RK4, respectively, the

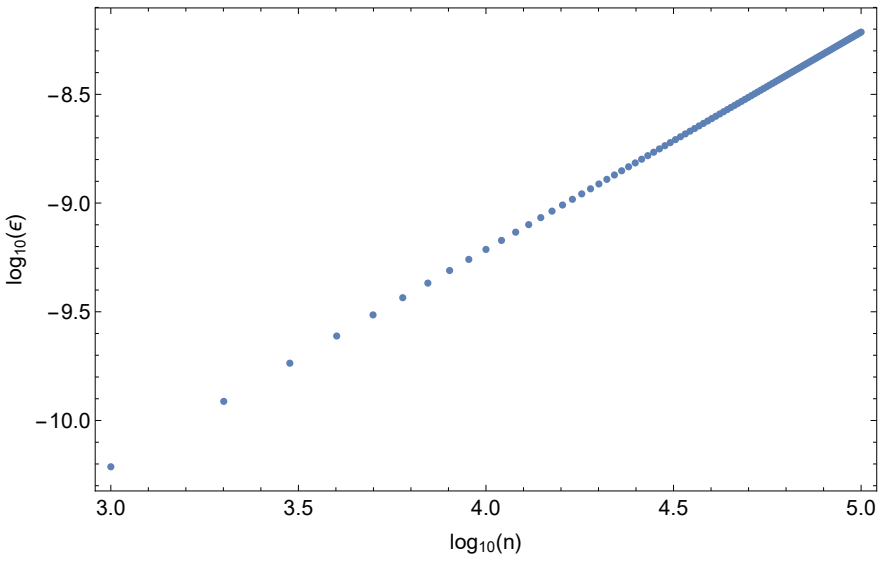


Figure 5.3: RK89 numerical integration error measure ε versus the number of turns n, in the integration of a radius $R = 10.1$ m circular orbit inside an electrostatic spherical deflector, using non-relativistic equations of motion, with initial step size $h = 1.1$ m, up to 10^5 turns.

error function values were both at $\sim 10^{-12}$. However, the calculation CPU times were 0.651 s and 14.7 s, respectively, which illustrates the computational speed advantage of a higher-order integrator.

The numerical integration error ε is related to step size h and integrator order n as $\varepsilon = ah^n$, where a is a constant, except for the nonlinear effects at extreme values of h, including the floating-point loss of significance. This is illustrated by Fig. 5.1, where the line slope of the linear pattern is 8.14.

5.6.2.3 Growth of Integrator Errors

Numerical integration errors tend to grow as functions of the number of turns through a lattice. To illustrate this, we tracked for 10^5 turns the same `RP 1 1 1` particle as in the initial step size calibration of RK89, using non-relativistic equation of motion, through the same spherical electrostatic deflector, with a circular Kepler orbit of radius $R = 10.1$ m and initial step size $h = 1.1$ m.

The growth of the error measure ε defined in eq. 5.6 as a function of the number of turns n is shown in Fig. 5.3. As expected, the growth of ε is linear.

The energy error ΔE was zero in the tracking run, which illustrates that the equations of motion and the beamline coordinate system used in *COSY INFINITY* and *MSURK89* are fundamentally

energy conserving for energy conserving lattices by design.

5.7 Magnetic Dipole and Spherical Deflector Tracking Test Cases

5.7.1 Homogeneous Magnetic Dipole

We analyzed particle motion inside a homogeneous magnetic dipole, and we performed particle tracking through the element in *COSY INFINITY* and *MSURK89*.

5.7.1.1 Analytic Representation of Motion

We consider a particle of mass m in a magnetic field B that is perpendicular to the particle's initial velocity. The Lorenz force F acting on the particle, transversal to the particle velocity v, is

$$F = ZevB,$$

resulting in circular motion. We denote the radius of the particle's circular trajectory as R.

Newton's second law yields, for the centrifugal force counterbalancing the Lorenz force,

$$\frac{\gamma m v^2}{R} = ZevB.$$

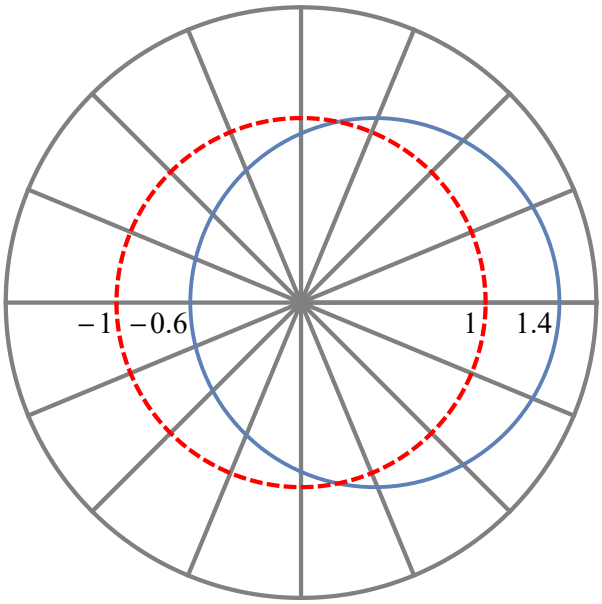
Simplifying the equation, we obtain

$$R = \frac{p}{ZeB}.$$

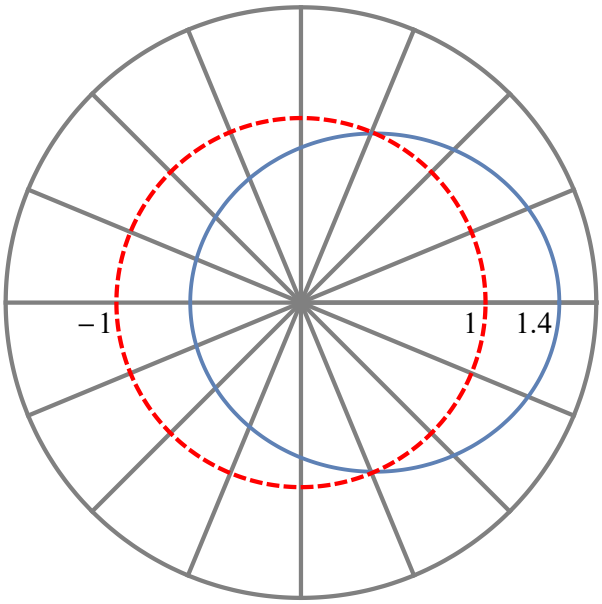
This relation is valid as both relativistic and non-relativistic.

If the homogeneous magnetic dipole is specified by the curvature radius R of the reference orbit, the magnetic field is set as $B = p/ZeR$. In that case, the trajectory is identical for relativistic equations of motion and their non-relativistic limit.

A particle launched in a homogeneous magnetic dipole with a horizontal transversal offset x relative to the reference particle (that is, with an initial beamline coordinate x) has, by translational symmetry, a circular orbit of the same radius as the reference particle, as Fig. 5.4(a) illustrates.



(a) Homogeneous magnetic deflector, comprising sixteen 22.5° sectors.



(b) Electrostatic spherical deflector, comprising sixteen 22.5° sectors.

Figure 5.4: The reference orbit (dashed red) and an orbit with initial horizontal offset $x_\mathrm{i} = 0.4\,\mathrm{m}$ (solid blue), in a system of sixteen 22.5° sectors. (Images source: Kyoko Makino, Eremey Valetov, & Martin Berz, Computation of Nonlinear Fields and Orbit and Spin Transfer Maps of Electrostatic Elements Using Differential Algebras, In *Proc. of International Computational Accelerator Physics Conference (ICAP'15), Shanghai, China, 12–16 October 2015*, JACoW, Geneva, Switzerland, 2016; use permitted under the Creative Commons Attribution License CC BY 3.0.)

5.7.1.2 Tracking results

We tracked a set of particles with kinetic energy 1 MeV, mass 1 amu, and charge $1\,e$ (in *COSY INFINITY* script, set by `RP 1 1 1`) through a homogeneous magnetic dipole specified by radius $R_0 = 1\,\mathrm{m}$ and angle $\alpha = 22.5°$, in *COSY INFINITY* and *MSURK89*, for 160,000 turns.

The phase space position projection to the x–a plane was drawn every 111 turns. The turn step of 111 was chosen to be ~ 100 and a coprime of 16, the number of 22.5° sectors in the full physical lattice, thereby moderating the number of data points but still drawing the system state at each of the 16 sectors.

The initial particle ray coordinates were

$$(x, a, b, y, l, \delta_K) = (k \cdot 0.07\,\mathrm{m}, 0, 0, 0, 0, 0)$$

for $k = 1, 2, \ldots, 10$.

In *COSY INFINITY*, the computation order was 19, the highest possible for this system without exhaustion of *COSY INFINITY*'s internal variables.

Fig. 5.5 shows the tracking pictures produced in *COSY INFINITY*, with and without symplectification. Fig. 5.6 shows a tracking picture produced in *MSURK89*.

As expected, there was no visual difference in tracking pictures between relativistic and non-relativistic equations of motion.

The tracking pictures are in accord with the particles having circular orbits in the midplane. For example, the tracking pictures demonstrate that, after passing through eight 22.5° sectors (a total angle of 180°) from the initial position, the x particle ray coordinates change sign but are of the same magnitude as the initial x coordinates.

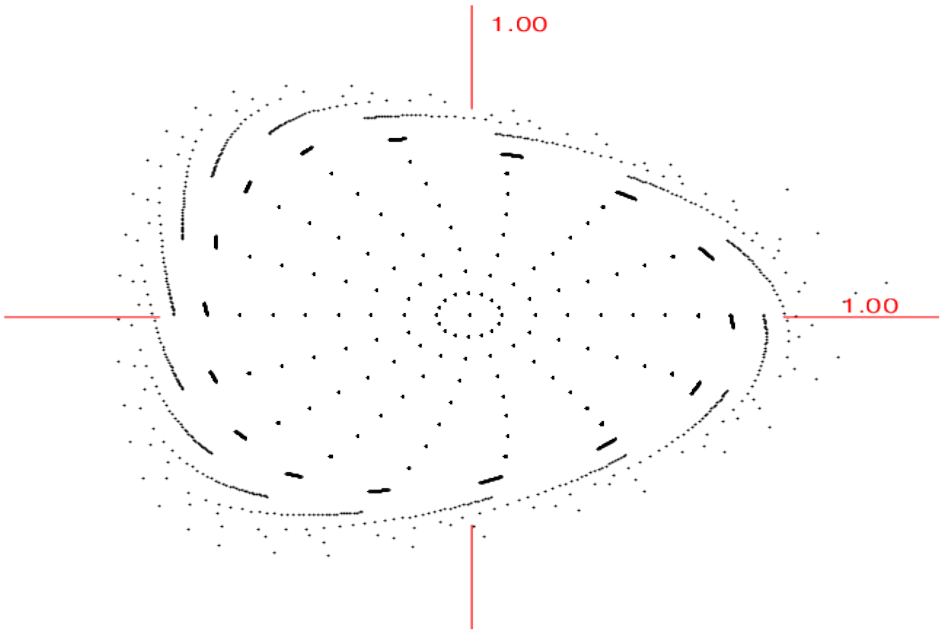
There is no visible distortion in the *MSURK89* tracking pictures, except for slight smearing of the outer orbits.

The smearing was significant in the outer orbit in *COSY INFINITY* without symplectification, with particles flying off to infinity. There was some visible smearing in the second outermost orbit. Turning on symplectification in *COSY INFINITY* decreased the smearing effect.

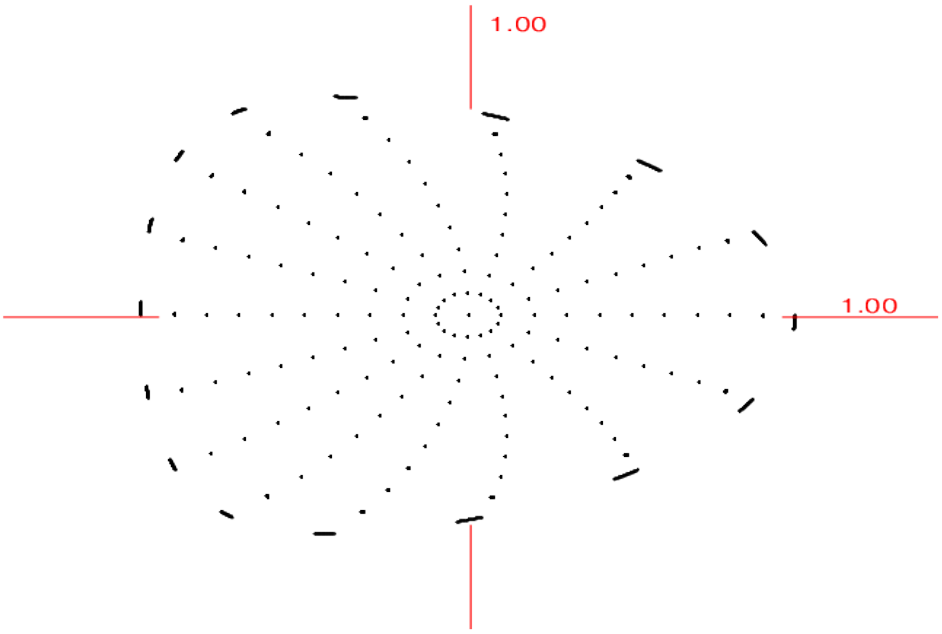
The CPU times are specified in the respective figure captions. As expected, calculations take significantly longer in *MSURK89*, as the integration of ODEs in *MSURK89* is performed in each tracking turn, whereas DA integration of ODEs in *COSY INFINITY* is performed for one turn to obtain the transfer map prior to tracking. Symplectification in *COSY INFINITY* increased the CPU time compared to no symplectification by an order of magnitude, but the CPU time was still an order of magnitude lower than in *MSURK89*.

5.7.2 Electrostatic Spherical Deflector

We analyzed particle motion inside an electrostatic spherical deflector, and we performed particle tracking through the element in *COSY INFINITY* and *MSURK89*.



(a) Tracking without symplectification. Calculation CPU time: 2.7 s.



(b) Tracking with EXPO symplectification. Calculation CPU time: 36.5 s.

Figure 5.5: Homogeneous magnetic dipole: the x–a projections of particles tracked in *COSY INFINITY* for $160,000$ turns. Particles were launched horizontally, with initial x offsets up to 0.7 m, through a homogeneous magnetic dipole of angle 22.5° and radius $R_0 = 1$ m.

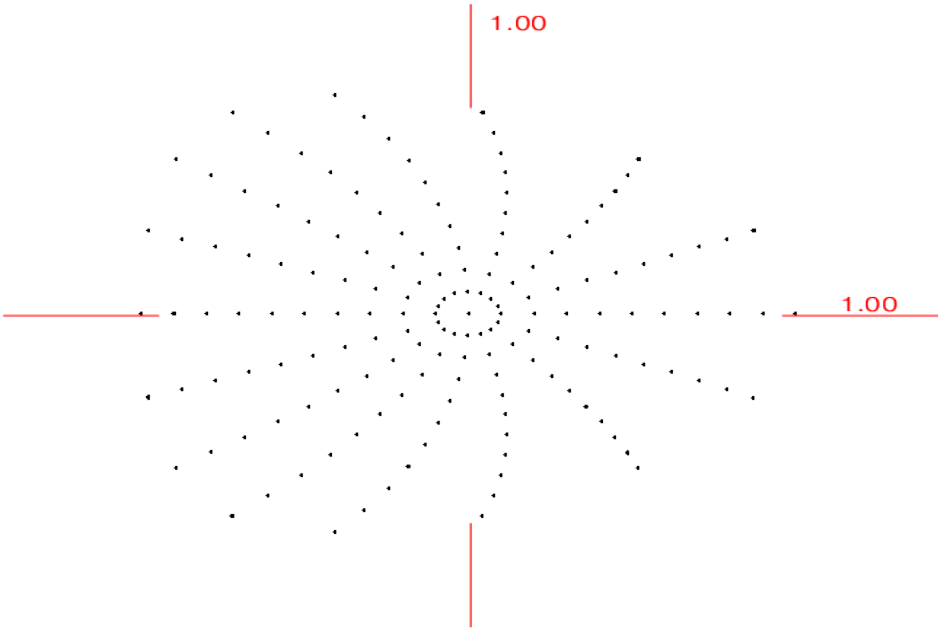


Figure 5.6: Homogeneous magnetic dipole: the x–a projections of particles tracked in *MSURK89* for 160,000 turns. Particles were launched horizontally, with initial x offsets up to 0.7 m, through a homogeneous magnetic dipole of angle 22.5° and radius $R_0 = 1$ m. Calculation CPU time: 314.5 s.

5.7.2.1 Analytic Representation of Non-Relativistic Motion

Following [62, pp. 30–39], we will show that the non-relativistic motion of a particle inside an electrostatic spherical deflector forms an ellipse.

In a central field, the Lagrangian of a particle of mass m is

$$L = \frac{1}{2} m \left(\dot{r}^2 + r^2 \ddot{\theta}^2 \right) - U(r), \tag{5.7}$$

where (r, θ) are the polar coordinates of the particle in the plane of its motion and $U(r)$ is the particle's potential energy in the central field.

Because the Lagrangian L in eq. 5.7 does not depend on θ (i.e., $\frac{\partial L}{\partial \theta} = 0$), by Euler–Lagrange's equation for coordinate θ, the angular momentum is conserved:

$$M = m r^2 \dot{\theta} = \text{const.} \tag{5.8}$$

Substituting eq. 5.8 into the expression for full energy $\mathscr{E}$, we obtain

$$\mathscr{E} = \frac{1}{2} m \dot{r}^2 + U_{\text{eff}}(r),$$

where U_{eff} is the effective potential energy

$$U_{\text{eff}}(r) = U(r) + \frac{M^2}{2mr^2}.$$

Consider the electrostatic potential $U(r)$ inside an electrostatic sphere:

$$U(r) = -\frac{\alpha}{r}, \alpha > 0.$$

Now, therefore,

$$\mathscr{E} = \frac{1}{2}m\dot{r}^2 - \frac{\alpha}{r} + \frac{M^2}{2mr^2}.$$

Expressing $\dot{r}$ through r, we obtain

$$\frac{dr}{dt} = \sqrt{\frac{2}{m}\left(\mathscr{E} + \frac{\alpha}{r}\right) - \frac{M^2}{mr^2}}.$$

However, from conservation of the angular momentum,

$$\frac{d\theta}{dt} = \frac{M}{mr^2},$$

so, applying the chain rule, we obtain θ as a function of r:

$$\theta = \int \frac{Mdr}{r^2\sqrt{2m\left(\mathscr{E} + \frac{\alpha}{r}\right) - \frac{M^2}{r^2}}}.$$

Performing analytic integration yields

$$\theta = \arccos\left(\frac{\frac{M}{r} - \frac{m\alpha}{M}}{\sqrt{2m\mathscr{E} + \frac{m^2\alpha^2}{M^2}}}\right) + \text{const}.$$

We rotate the coordinate system so that the integration constant is zero. The result is the equation of an ellipse in polar form, with a focus as the origin [62, p. 36]:

$$\frac{p}{r} = 1 + e\cos\theta,$$

where p is the focal parameter

$$p = \frac{M^2}{m\alpha}$$

and e is the eccentricity

$$e = \sqrt{1 + \frac{2\mathscr{E}M^2}{m\alpha^2}}.$$

We will now demonstrate that the origin is indeed a focus of the ellipse. In Cartesian coordinates, a parametric form of the ellipse equation is [62, pp. 36–38]

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} a(\cos\theta - e) \\ a\sqrt{1-e^2}\sin\theta \end{pmatrix},$$

where $a = p/\left(e^2 - 1\right)$ is the major semi-axis of the ellipse.

The leftmost point of the ellipse is $(x,y) = (-a - ae, 0)$, and the rightmost point is $(a - ae, 0)$. If the origin is a focus of the ellipse, then the other focus is, by symmetry, at point $(-2ae, 0)$.

The sum of distances from each supposed focus point to a point on the ellipse is

$$\begin{aligned}
\mathscr{L} &= l_1 + l_2 = \\
&= a\sqrt{1 - 2e\cos\theta + e^2 - e^2\sin^2\theta} + a\sqrt{1 + 2e\cos\theta + e^2 - e^2\sin^2\theta} = \\
&= a(1 - e\cos\theta) + a(1 + e\cos\theta) = \\
&= 2a = \text{const},
\end{aligned}$$

which satisfies the two-center bipolar coordinate equation $l_1 + l_2 = 2a$ and the definition of the foci of an ellipse.

5.7.2.2 Analytic Representation of Relativistic Motion

In the relativistic case, the angular momentum of a particle is

$$M = pr = \gamma mrv,$$

where p and v are the particle's momentum and velocity components perpendicular to the radius-vector, respectively.

The angular momentum M can be expressed in polar coordinates (r, θ) as

$$M = \frac{mr^2\dot{\theta}}{\sqrt{1 - \frac{\dot{r}^2 + r^2\dot{\theta}^2}{c^2}}}.$$

The total energy $\mathscr{E}$ of the particle in Coulomb field $U(r) = -\alpha/r$ is

$$\mathscr{E} = \sqrt{p^2c^2 + m^2c^4} - \frac{\alpha}{r}.$$

As shown in [63, pp. 100–102][15], there are three cases:

1. $Mc > |\alpha|$. The particle trajectory equation is

$$\left(c^2M^2 - \alpha^2\right)\frac{1}{r} = c\sqrt{(M\mathscr{E})^2 - m^2c^2\left(M^2c^2 - \alpha^2\right)}\cos\left(\theta\sqrt{1 - \frac{\alpha^2}{M^2c^2}}\right) + \mathscr{E}\alpha$$

   and has the visual appearance of "rosettes". Whereas in the non-relativistic case the trajectory is a closed elliptical orbit, the trajectory in the relativistic case with $Mc > |\alpha|$ is an elliptical trajectory with relativistic precession, which is no longer closed.

2. $Mc < |\alpha|$. The particle trajectory equation is

$$\left(\alpha^2 - c^2M^2\right)\frac{1}{r} = \pm c\sqrt{(M\mathscr{E})^2 - m^2c^2\left(M^2c^2 - \alpha^2\right)}\cosh\left(\theta\sqrt{\frac{\alpha^2}{M^2c^2} - 1}\right) - \mathscr{E}\alpha,$$

   where the sign at the beginning of the right-hand side is the same as the sign of α. Trajectories in this case spiral into the center, out of the center, or both.

3. $Mc = |\alpha|$. The particle trajectory equation is [63, pp. 100–102]

$$\frac{2\mathscr{E}\alpha}{r} = \mathscr{E}^2 - m^2c^4 - \theta^2\left(\frac{\mathscr{E}\alpha}{Mc}\right)^2.$$

Only the $Mc > |\alpha|$ case has a qualitatively similar correspondence in the non-relativistic limit. The other cases $Mc \leq |\alpha|$ degenerate into a straight-line orbit [15].

For particles such as the proton and the deuteron, $Mc > |\alpha|$, which, *inter alia*, makes their use in electrostatic storage rings possible.

5.7.2.3 Reference and Non-Reference Trajectories

In an electrostatic spherical deflector, we choose the reference trajectory as a circular orbit. We obtain the voltages of the inner and outer spherical shells of the deflector corresponding to the

particle kinetic energy and the orbit radius by solving Newton's second law equation for zero radial acceleration. Alternatively, this can be done by solving for zero orbital eccentricity.

A particle launched in an electrostatic spherical deflector has an elliptic orbit, as Fig. 5.4(b) illustrates. Particle orbits are subject to the physical aperture limitation.

5.7.2.4 Tracking Results

We tracked a set of particles with kinetic energy 1 MeV, mass 1 amu, and charge $1\,e$ (in *COSY INFINITY* script, set by `RP 1 1 1`) through an electrostatic spherical deflector sector specified by radius $R_0 = 1\,\mathrm{m}$ and angle $\alpha = 22.5^\circ$, in *COSY INFINITY* and *MSURK89*, for $160,000$ turns.

The phase space position projection to the x–a plane was drawn every 111 turns. The turn step of 111 was chosen to be ~ 100 and a coprime of 16, the number of 22.5° sectors in the full physical lattice, thereby moderating the number of data points but still drawing the system state at each of the 16 sectors.

The initial particle ray coordinates were

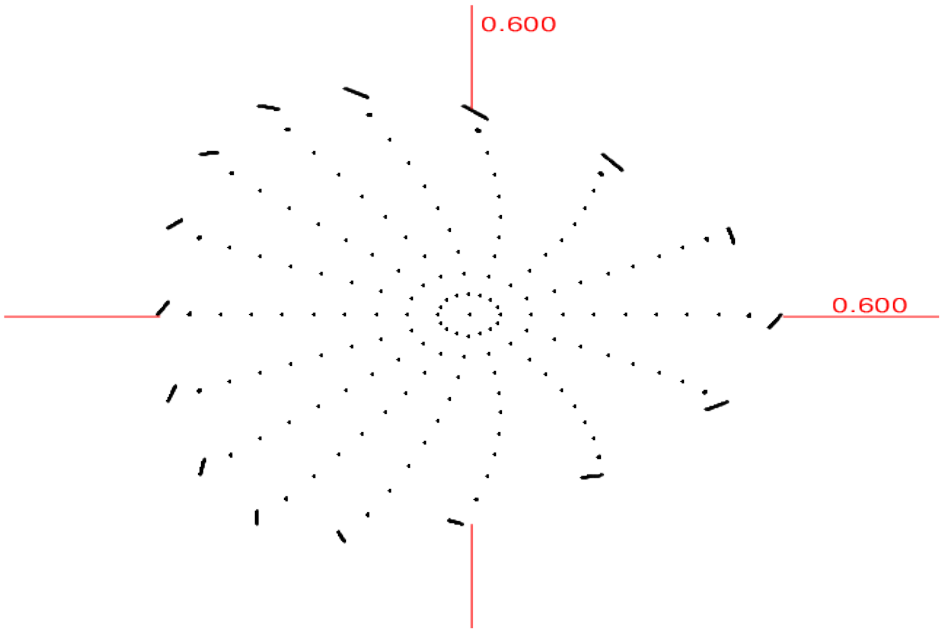
$$(x, a, b, y, l, \delta_K) = (k \cdot 0.04\,\mathrm{m}, 0, 0, 0, 0, 0)$$

for $k = 1, 2, \ldots, 10$.

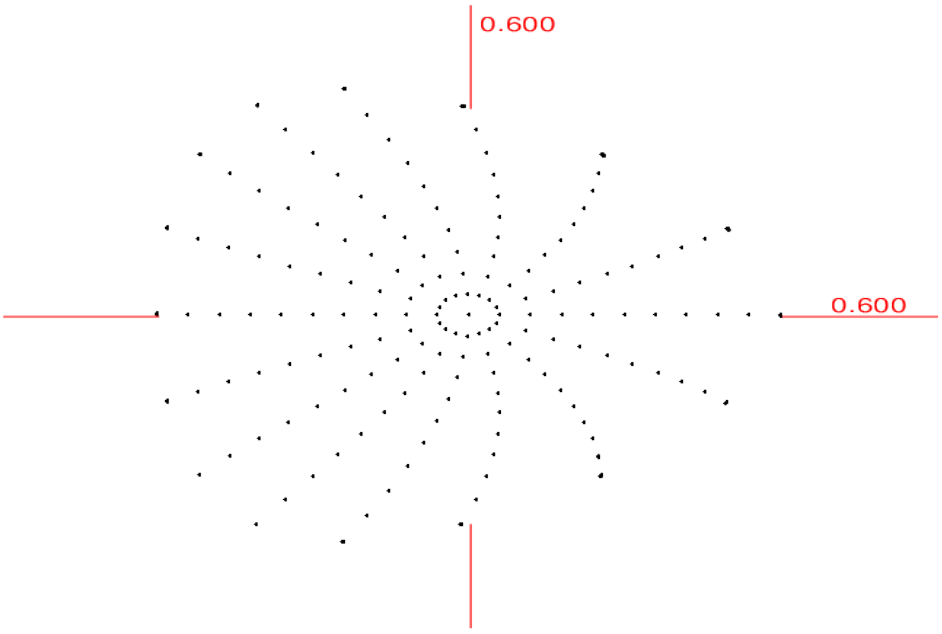
In *COSY INFINITY*, the computation order was 19, the highest possible for this system without exhaustion of *COSY INFINITY*'s internal variables.

Tracking pictures produced in *COSY INFINITY*, using non-relativistic and relativistic equations of motion, are shown in Figs. 5.7 and 5.9, respectively. Each set of *COSY INFINITY* tracking pictures comprises versions with and without symplectification. Tracking pictures produced in *MSURK89*, using non-relativistic and relativistic equations of motion, are shown in Figs. 5.8 and 5.10, respectively.

The tracking pictures illustrate that Kepler orbits are (1) closed with one-turn periodicity under non-relativistic equations of motion and (2) are not closed and perform relativistic orbital precession under relativistic equations of motion.



(a) Tracking without symplectification. Calculation CPU time: 3.4 s.



(b) Tracking with EXPO symplectification. Calculation CPU time: 39.2 s.

Figure 5.7: Electrostatic spherical deflector: the x–a projections of particles tracked in *COSY INFINITY* for 160,000 turns, with non-relativistic equations of motion. Particles were launched horizontally, with initial x offsets up to 0.4 m, through an electrostatic spherical deflector sector of angle 22.5° and radius $R_0 = 1$ m.

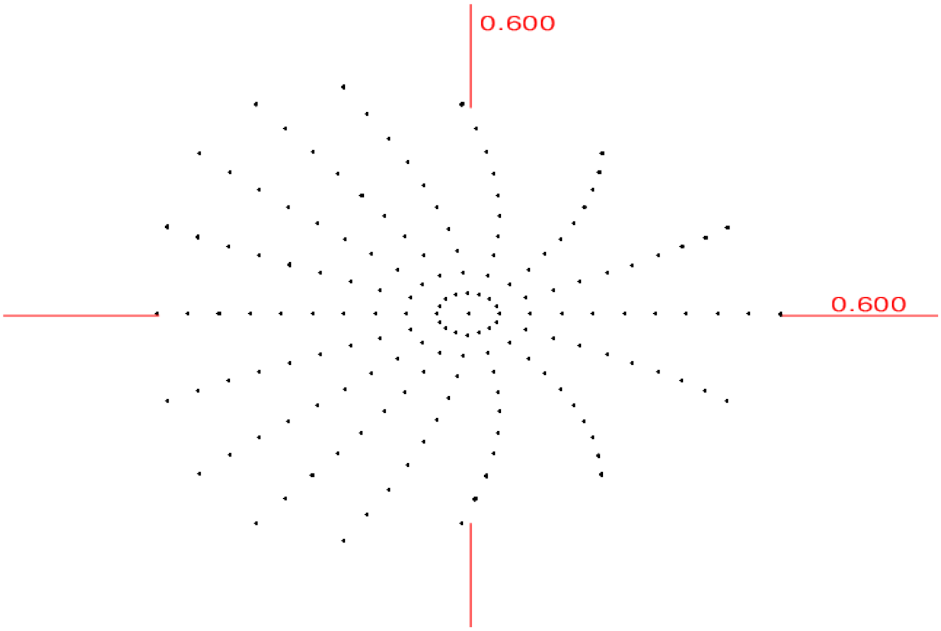


Figure 5.8: Electrostatic spherical deflector: the x–a projections of particles tracked in *MSURK89* for 160,000 turns, with non-relativistic equations of motion. Particles were launched horizontally, with initial x offsets up to 0.4 m, through an electrostatic spherical deflector sector of angle 22.5°. Calculation CPU time: 1579.9 s.

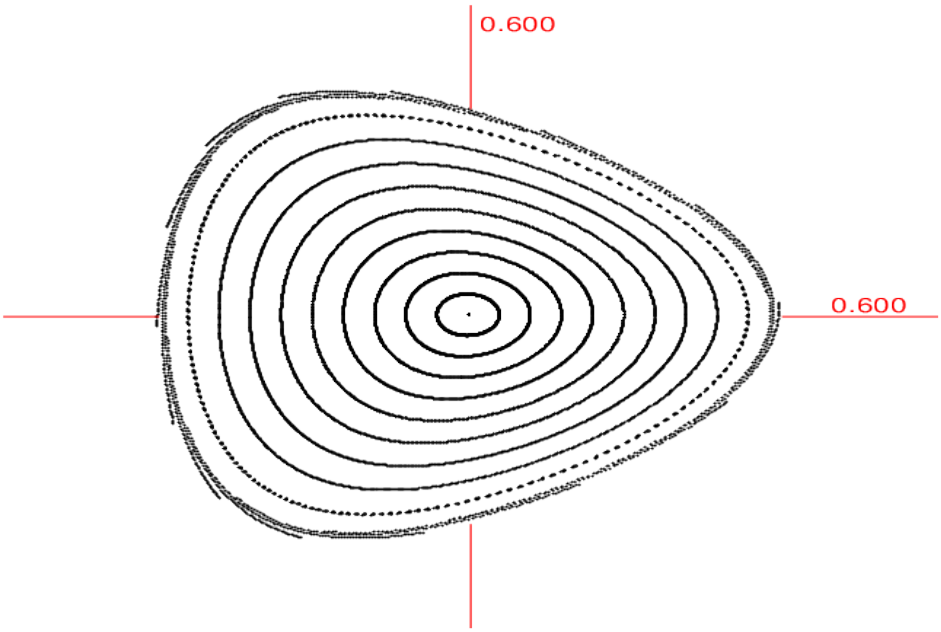
There was no visible distortion in *MSURK89* or *COSY INFINITY* with symplectification on, except for slight smearing in the outer orbits in Fig. 5.7(b).

Distortion was somewhat significant in the outer orbit in *COSY INFINITY* with non-relativistic equations of motion and no symplectification. Turning on symplectification in *COSY INFINITY* decreased the distortion.

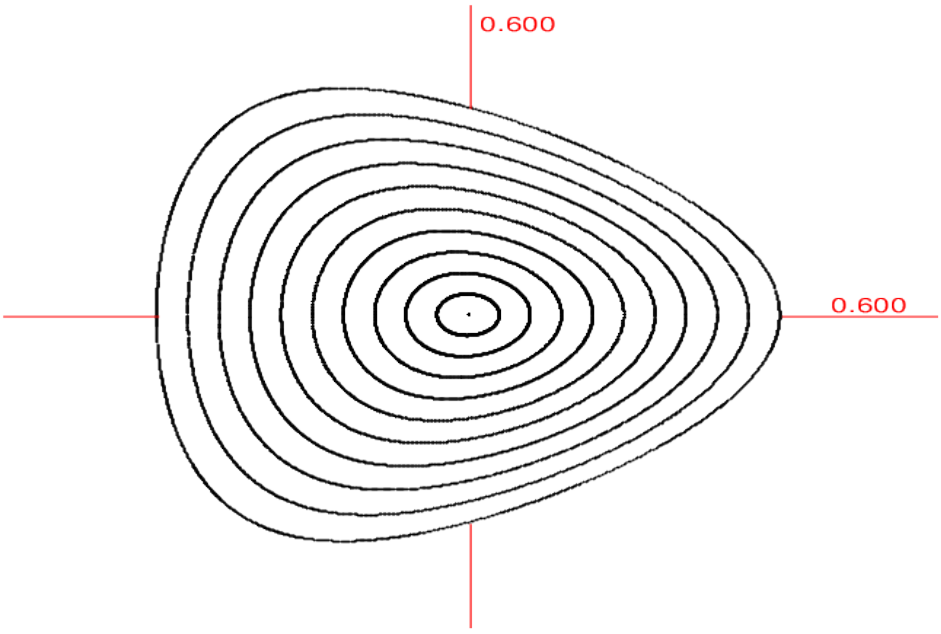
Calculation CPU times were similar for the electrostatic spherical deflector and the homogeneous magnetic dipole.

5.7.2.4.1 Effect of the Computation Order on Accuracy

Figs. 5.11 and 5.12 illustrate the effect of the computation order on accuracy with and without symplectification, respectively. The motion is linear and consists of ellipses in the x–a plane for computation order 1. Comparison with higher orders up to 19 illustrates how accuracy increases with computation order. Whereas accuracy varies strongly between orders without symplectification,



(a) Tracking without symplectification. Calculation CPU time: 3.6 s.



(b) Tracking with EXPO symplectification. Calculation CPU time: 39.4 s.

Figure 5.9: Electrostatic spherical deflector: the x–a projections of particles tracked in *COSY INFINITY* for 160,000 turns, with relativistic equations of motion. Particles were launched horizontally, with initial x offsets up to 0.4 m, through an electrostatic spherical deflector sector of angle 22.5° and radius $R_0 = 1$ m.

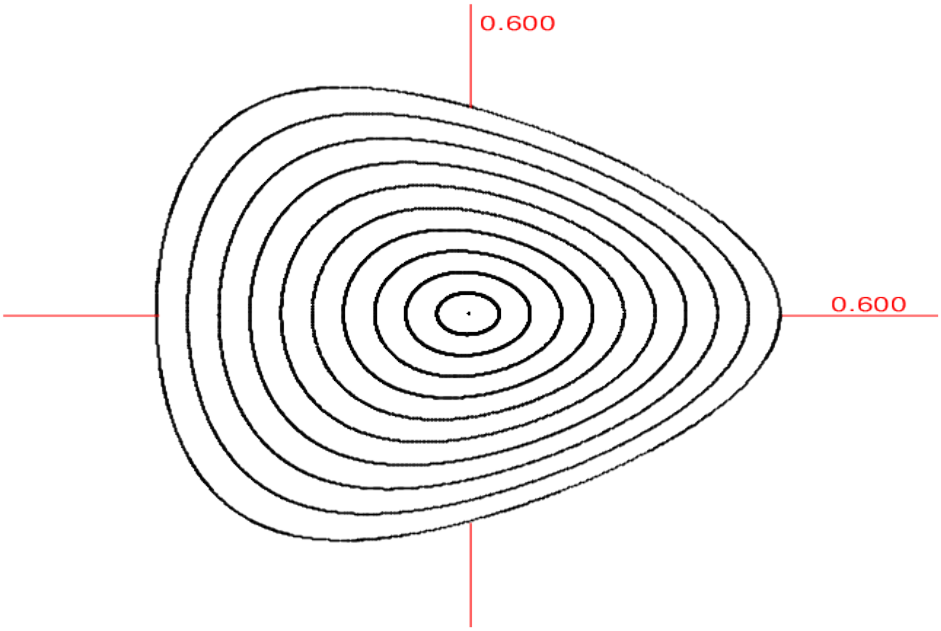


Figure 5.10: Electrostatic spherical deflector: the x–a projections of particles tracked in *MSURK89* for 160,000 turns, with relativistic equations of motion. Particles were launched horizontally, with initial x offsets up to 0.4 m, through an electrostatic spherical deflector sector of angle 22.5° and radius $R_0 = 1$ m. Calculation CPU time: 416.0 s.

with symplectification the visual difference between orders is generally insignificant, demonstrating how effective symplectification can be for the generation of realistic tracking pictures.

Computation CPU times, listed in the figure captions, illustrate their monotonous increase with computation order due to the increasing complexity of the underlying basic DA operations.

5.7.2.4.2 An Additional Benchmark Example

We performed particle tracking through the same electrostatic spherical deflector configuration as detailed above and illustrated in Figs. 5.7–5.10, but with particles launched with initial y offsets up to 0.4 m instead of initial x offsets up to 0.4 m. Again, computation order 19 was used.

Figs. 5.13 and 5.14 show tracking results in y–b and x–a planes, respectively. In both figures, tracking pictures produced in *COSY INFINITY* and *MSURK89* are visually indistinguishable.

In *COSY INFINITY*, computations in the 2D phase space (x–a and y–b canonical coordinate pairs) are more computation-intensive than in the 1D phase space (x–a only). Accordingly, without

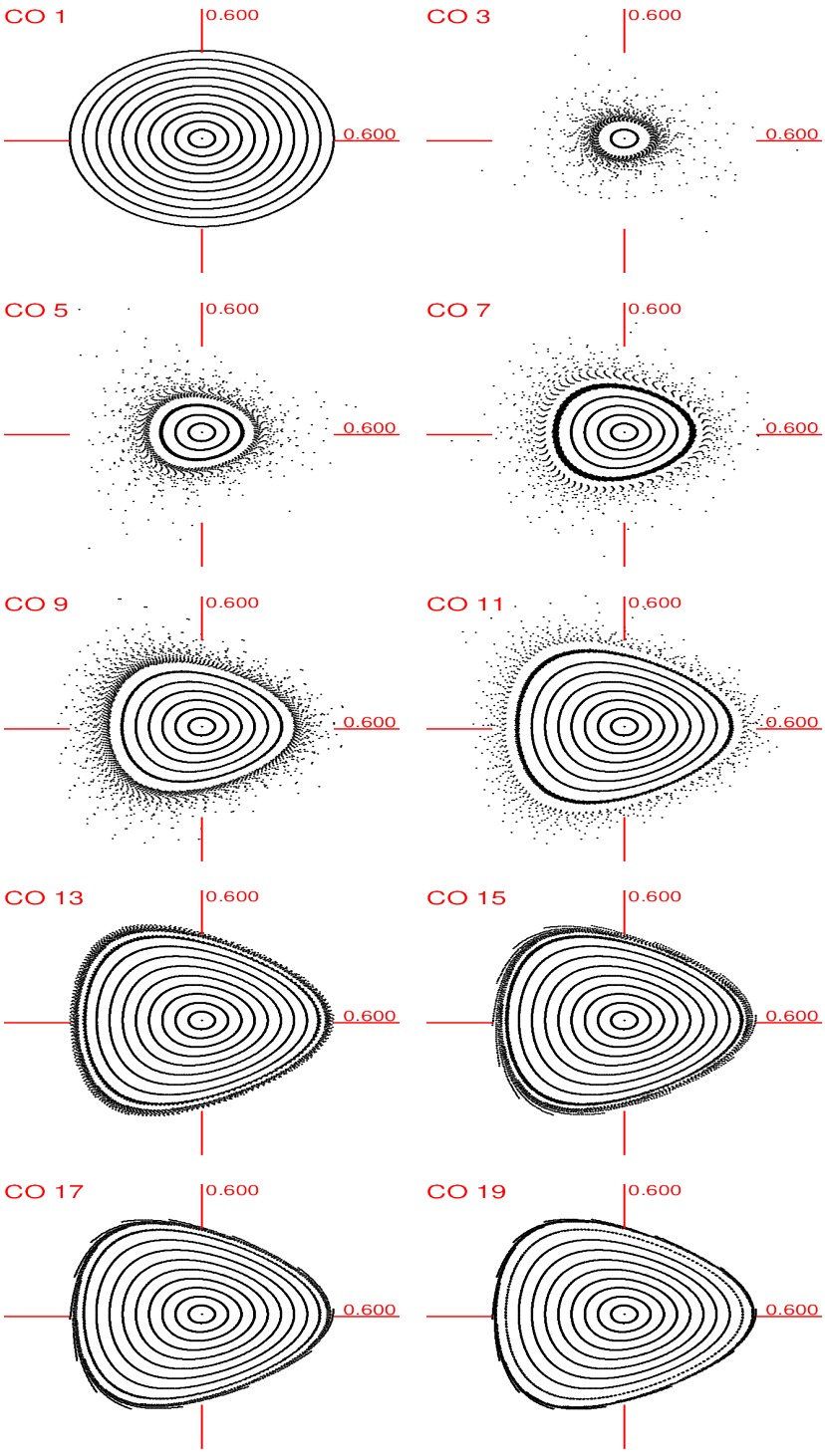


Figure 5.11: For odd computation orders 1 to 19 (computation order specified in the pictures), the x–a projections of particles tracked in *COSY INFINITY,* for 160,000 turns, without symplectification, with relativistic equations of motion. Particles were launched horizontally, with initial x offsets up to 0.4 m, through an electrostatic spherical deflector sector of angle 22.5° and radius $R_0 = 1$ m. Calculation CPU times: 0.4 s, 0.5 s, 0.6 s, 0.9 s, 1.2 s, 1.5 s, 1.9 s, 2.4 s, 2.9 s, and 3.6 s, respectively.

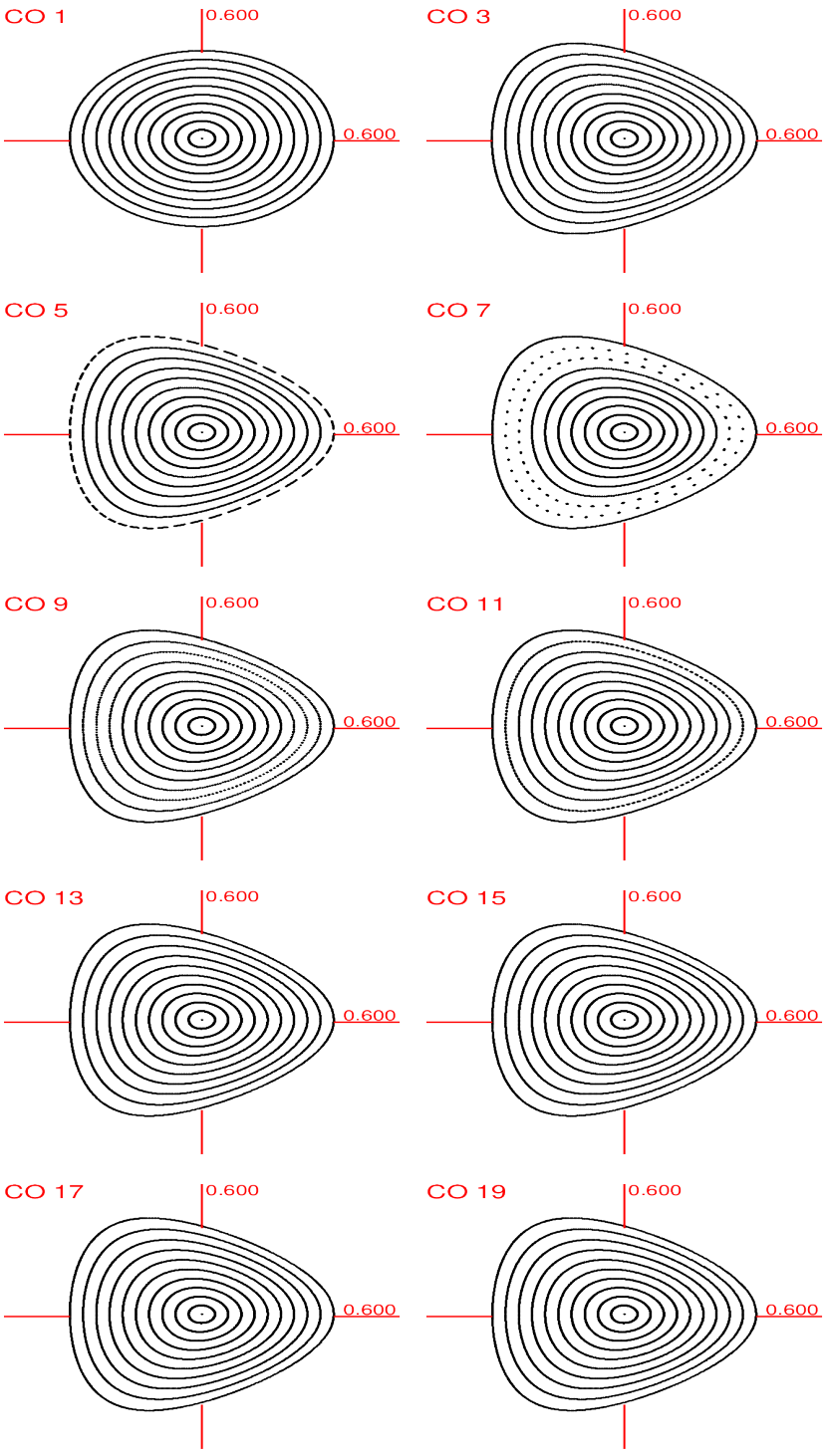


Figure 5.12: For odd computation orders 1 to 19 (computation order specified in the pictures), the x–a projections of particles tracked in *COSY INFINITY,* for 160,000 turns, with EXPO symplectification, with relativistic equations of motion. Particles were launched horizontally with initial x offset up to 0.4 m through an electrostatic spherical deflector sector of angle 22.5° and radius $R_0 = 1$ m. Calculation CPU times: 1.0 s, 7.6 s, 9.8 s, 12.9 s, 16.2 s, 19.9 s, 25.1 s, 30.5 s, 34.9 s, 39.4 s, respectively.

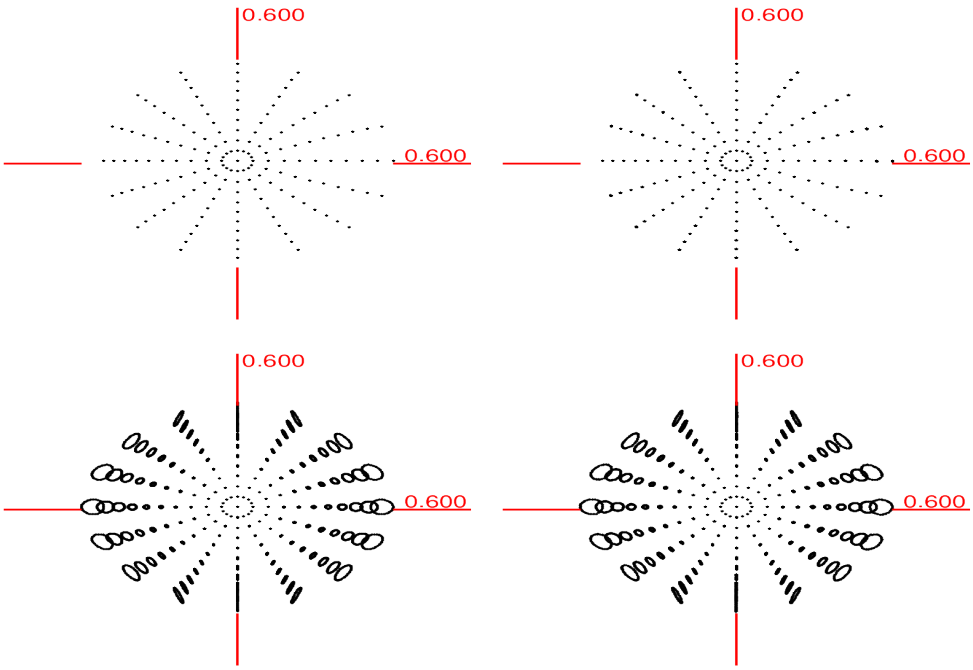


Figure 5.13: Electrostatic spherical deflector: the y–b projections of particles tracked in *COSY INFINITY* (left) and *MSURK89* (right), for 160,000 turns, with non-relativistic (top) and relativistic (bottom) equations of motion. Particles were launched horizontally, with initial y offsets up to 0.4 m, through an electrostatic spherical deflector sector of angle 22.5° and radius $R_0 = 1$ m.

symplectification, tracking with y offset particles completed in ~ 140 s, whereas tracking with x offset particles completed in ~ 7 s. With symplectification, tracking with y offset particles completed in ~ 950 s, whereas tracking with x offset particles completed in ~ 40 s.

5.8 Spherical Deflector Transfer Map Test Cases

In this section, we will describe and perform computation of the non-relativistic DA transfer map of an electrostatic spherical deflector in a laboratory coordinate system using two conventional methods: (1) by integrating the ODEs of motion using an 8th order Runge–Kutta–Verner integrator and (2) by computing analytically and in closed form the properties of the respective elliptical orbits from Kepler theory. We will compare the resulting transfer maps with (3) the DA transfer map of the non-relativistic version of *COSY INFINITY*'s built-in electrostatic spherical deflector element `ESP` and (4) the transfer map of the electrostatic spherical deflector computed using the program *GIOS* [114].

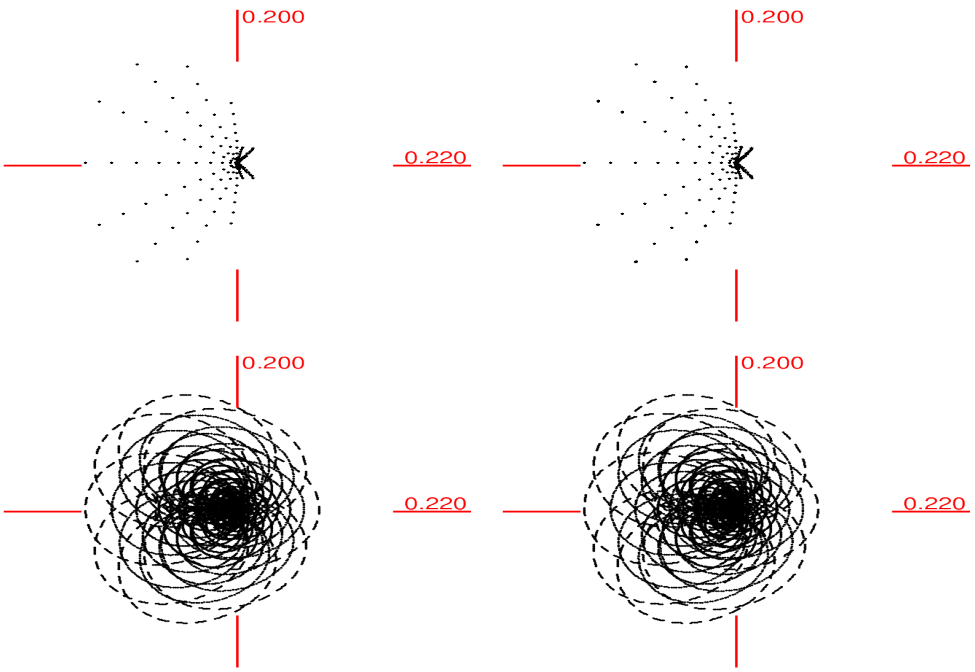


Figure 5.14: Electrostatic spherical deflector: the x–a projections of particles tracked in *COSY INFINITY* (left) and *MSURK89* (right), for 160,000 turns, with non-relativistic (top) and relativistic (bottom) equations of motion. Particles were launched horizontally, with initial y offsets up to 0.4 m, through an electrostatic spherical deflector sector of angle 22.5° and radius $R_0 = 1$ m.

5.8.1 Integration of the ODEs in Polar Laboratory Coordinates

Consider a bunch of non-relativistic charged particles launched with kinetic energy $K_0 = mv_0^2/2$ and zero potential energy, where m is the particle mass and v_0 is the reference velocity. Suppose that a circular reference orbit of radius r_0 is defined for the particle bunch in an electrostatic spherical deflector. Now, for concreteness, consider one particle in this bunch. Following the convention of the potential energy as zero at the reference orbit, we calibrate the potential energy of the charged particle to

$$U(r) = -\frac{\alpha}{r} + \frac{\alpha}{r_0}.$$

By energy conservation, an off-reference particle at initial radius r_i would upon entering the deflector have an initial velocity magnitude v_i such that

$$K_0 = \frac{mv_\mathrm{i}^2}{2} + U(r) = \frac{mv_\mathrm{i}^2}{2} - \frac{\alpha}{r_\mathrm{i}} + \frac{\alpha}{r_0},$$

from where we obtain

$$v_{\mathrm{i}} = \sqrt{v_0^2 - \frac{2\alpha}{m}\left(\frac{1}{r_0} - \frac{1}{r_{\mathrm{i}}}\right)}, \tag{5.9}$$

or, expressing α in terms of the particle charge $q = Ze$, the reference orbit radius r_0, and the electric rigidity $\chi_{\mathrm{e}} = \frac{pv}{Ze}$ as $\alpha = Ze\chi_{\mathrm{e}} r_0$,

$$v_{\mathrm{i}} = \sqrt{v_0^2 - \frac{2}{m} Ze\chi_{\mathrm{e}}\left(1 - \frac{r_0}{r_{\mathrm{i}}}\right)}. \tag{5.10}$$

COSY INFINITY's horizontal transversal coordinate x is defined relative to the circular reference orbit as $x = r - r_0$, where r is the length of the projection of a particle's radius vector on the plane of the reference particle's orbital motion. *COSY INFINITY*'s horizontal momentum component $a = p_x/p_0$ can be, in the non-relativistic case, expressed as $a = v_x/v_0$, where v_x is the x component of a particle's velocity and v_0 is the reference velocity.

Now, let $(x_{\mathrm{i}}, a_{\mathrm{i}})$ be the initial beamline coordinates of the particle in the electrostatic spherical deflector. In reality, the deflector would have a fringe field; in this model, we approximate the fringe field by an instant jump in the electrostatic potential from zero to the electrostatic potential of the electrostatic spherical deflector at radius $r_{\mathrm{i}} = r_0 + x_{\mathrm{i}}$. Thus, all particles in a bunch experience a "step down" from kinetic energy K_0 to a kinetic energy $K = K_0 - U(r)$ at time $t = 0$.

Similarly, at the end of the element, the particle will experience a "step up" due to the change of the potential energy back to zero. Note that these changes of energy are essential even in the absence of a true fringe field treatment to preserve the actual beam energy.

For calculations, we will use the polar laboratory coordinate system (r, θ), with particles launched at polar angle $\theta = 0$. Additionally, we will use the Cartesian laboratory coordinate system

$$(\tilde{x}, \tilde{y}) = r(\cos\theta, \sin\theta).$$

In this Cartesian laboratory coordinate system, the particle has the initial position

$$\vec{r}_{\mathrm{i}} = (r_0 + x_{\mathrm{i}}, 0) \tag{5.11}$$

and the initial velocity

$$\vec{v}_{\mathrm{i}} = \left(a_{\mathrm{i}} v_0, \sqrt{v_{\mathrm{i}}^2 - (a_{\mathrm{i}} v_0)^2}\right), \tag{5.12}$$

where v_i is the initial particle velocity magnitude at the initial radius r_i obtained using eq. 5.10.

From Newton's second law, the radial acceleration of the particle in the electrostatic spherical deflector is

$$\frac{d^2}{dt^2}r = -\frac{\mu}{r^2} + \omega^2 r,$$

where $\mu = \alpha/m$ and ω is the particle's angular velocity.

Considering eqns. 5.11 and 5.12, the conservation of the angular momentum can be expressed in terms of massless angular momentum

$$h = \omega r^2 = |\vec{r} \times \vec{v}|$$

as

$$\begin{aligned} h &= |\vec{r}_\mathrm{i} \times \vec{v}_\mathrm{i}| = |(\vec{r}_\mathrm{i})_1 (\vec{v}_\mathrm{i})_2 - (\vec{r}_\mathrm{i})_2 (\vec{v}_\mathrm{i})_1| = \\ &= \left| (r_0 + x_\mathrm{i}) \sqrt{v_\mathrm{i}^2 - (a_\mathrm{i} v_0)^2} - 0 \cdot a_\mathrm{i} v_0 \right| = \\ &= (r_0 + x_\mathrm{i}) \sqrt{v_\mathrm{i}^2 - (a_\mathrm{i} v_0)^2}. \end{aligned} \tag{5.13}$$

Taking the time derivative of $\omega = h/r^2$, we obtain

$$\frac{d^2}{dt^2}\theta = -2\frac{h}{r^3} v_r.$$

The final position $\vec{r}_\mathrm{f}$ and velocity $\vec{v}_\mathrm{f}$ can be obtained by integrating the ODEs of motion in the polar coordinates

$$\frac{d}{dt}\begin{pmatrix} r \\ v_r \\ \theta \\ \omega \end{pmatrix} = \begin{pmatrix} v_r \\ -\frac{\mu}{r^2} + \omega^2 r \\ \omega \\ -2\frac{h}{r^3} v_r \end{pmatrix}, \tag{5.14}$$

with the initial condition

$$\begin{pmatrix} r \\ v_r \\ \theta \\ \omega \end{pmatrix}_0 = \begin{pmatrix} (\vec{r}_\mathrm{i})_1 \\ (\vec{v}_\mathrm{i})_1 \\ 0 \\ (\vec{v}_\mathrm{i})_2 / (\vec{r}_\mathrm{i})_1 \end{pmatrix}.$$

Considering that $d\theta/dt = \omega$, applying the chain rule, this system of ODEs can be expressed in terms of polar angle θ as the independent variable as

$$\frac{d}{d\theta}\begin{pmatrix} r \\ v_r \\ \theta \\ \omega \end{pmatrix} = \begin{pmatrix} \frac{v_r}{\omega} \\ -\frac{\mu}{r^2}\frac{1}{\omega} + \omega r \\ 1 \\ -2\frac{h}{r^3}\frac{v_r}{\omega} \end{pmatrix} \tag{5.15}$$

with the same initial condition.

Having obtained the final position

$$\vec{r}_\mathrm{f} = r_\mathrm{f}(\cos\theta_\mathrm{f}, \sin\theta_\mathrm{f})$$

and velocity $\vec{v}_\mathrm{f}$, the final (x,a) coordinates are

$$(x_\mathrm{f}, a_\mathrm{f}) = \left(r_\mathrm{f} - r_0, \frac{\vec{v}_\mathrm{f}\cdot\vec{r}_\mathrm{f}}{v_0 r_\mathrm{f}}\right).$$

We calculated the DA transfer map of the electrostatic spherical deflector, in (x,a) beamline coordinates, with reference orbit radius $r_0 = 1\,\mathrm{m}$, by integration of the ODEs of motion in polar laboratory coordinates using an 8th order Runge–Kutta–Verner integrator. This transfer map is listed in sec. 5.8.3.

To obtain a DA-valued transfer map, we set the initial phase space coordinates as

$$(x_\mathrm{i}, a_\mathrm{i}) = (d_1, d_2) = (\mathtt{DA}(1), \mathtt{DA}(2)),$$

the respective DA generators [11, pp. 86–96]. We performed the calculations with the computation order 3.

A version of the *COSY INFINITY* program that uses a 4th order Runge–Kutta integrator to obtain the transfer map by solving the ODEs of motion in polar laboratory coordinates is listed in App. H.1. Additionally, App. H.2 lists a *Mathematica* program that integrates the ODEs in polar laboratory coordinates for an individual orbit, plots one turn of the orbit, and outputs the final (x,a) coordinates.

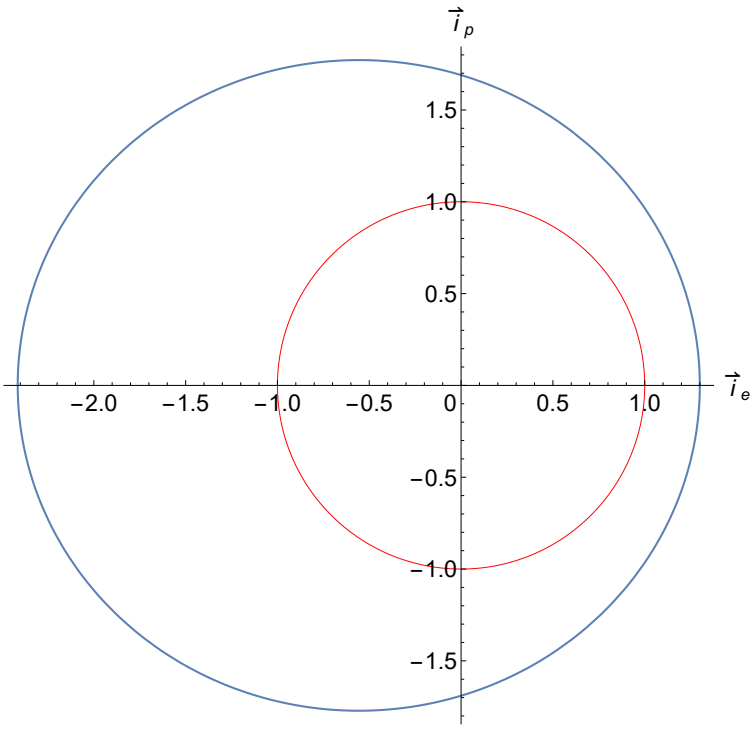


Figure 5.15: Orbit of a particle launched counter-clockwise at polar angle $f_\mathrm{i} = 0$ with initial beamline coordinates $(x_\mathrm{i}, a_\mathrm{i}) = (0.3, 0)$ through an electrostatic spherical deflector (blue). The reference orbit (red) has the radius $r_0 = 1\,\mathrm{m}$. The plot illustrates the basis vectors $\left(\vec{i}_e, \vec{i}_p\right)$ of the heliocentric coordinate system in relation to the orbit geometry. The plot was generated by the *Mathematica* notebook for integration of the ODEs for individual orbits in the polar laboratory coordinate system listed in App. H.2.

5.8.2 Kepler Theory Transition Matrix and Lagrange Coefficients

Motion in a central field with a potential energy of the form $U(r) = -\alpha/r + \text{const}$ is described by conventional Kepler theory; in particular, by the equation of orbit

$$r = \frac{p}{1 + e\cos f}, \tag{5.16}$$

where p is the focal parameter, e is the orbit eccentricity, and polar angle f is called the true anomaly [9, p. 117]. True anomaly $f = 0$ corresponds to the direction of the perihelion, that is, the point of the orbit nearest to the orbital focus at the origin of the polar coordinate system (r, f).

The particle position $\vec{r}$ is expressed as

$$\vec{r} = \vec{i}_e r\cos f + \vec{i}_p r\sin f, \tag{5.17}$$

in terms of polar coordinates (r, f) and the basis vectors $\left(\vec{i}_e, \vec{i}_p\right)$ of the heliocentric coordinate system, where $\vec{i}_e$ is a unit vector in the direction of perihelion and unit vector $\vec{i}_p$ is chosen so that $\vec{i}_e \times \vec{i}_p$ is codirectional with the vector of the angular velocity of the particle, as Fig. 5.15 illustrates.

According to Newton's second law, zero radial acceleration at the reference orbit requires

$$\frac{mv_0^2}{r_0} = \frac{\alpha}{r_0^2}.$$

Thus, the parameter $\mu = \alpha/m$ can be expressed as

$$\mu = \frac{\alpha}{m} = v_0^2 r_0 \tag{5.18}$$

in terms of the reference velocity v_0 and the reference orbit radius r_0.

The focal parameter p can be expressed in terms of μ and the massless angular momentum $h = |\vec{r} \times \vec{v}|$ as

$$p = \frac{h^2}{\mu}, \tag{5.19}$$

which is obtained in course of a standard derivation of the equation of orbit, eq. 5.16, as in [9, pp. 114–116].

We note that taking the time derivative of both sides of eq. 5.16 gives

$$\dot{r} = \dot{f} r \frac{e \sin f}{1 + e \cos f}. \tag{5.20}$$

Considering eqns. 5.20 and 5.19 and conservation of massless angular momentum $h = \dot{f} r^2$, taking the time derivative of the position vector $\vec{r}$ in eq. 5.17 yields for the particle velocity

$$\begin{aligned}
\vec{v} &= \vec{i}_e \left(\dot{r} \cos f - r \dot{f} \sin f\right) + \vec{i}_p \left(\dot{r} \sin f + r \dot{f} \cos f\right) = \\
&= \vec{i}_e \frac{h}{r} \left(\frac{e \sin f}{1 + e \cos f} \cos f - \sin f\right) + \vec{i}_p \frac{h}{r} \left(\frac{e \sin f}{1 + e \cos f} \sin f + \cos f\right) = \\
&= \vec{i}_e \frac{h}{r} \left(-\frac{\sin f}{1 + e \cos f}\right) + \vec{i}_p \frac{h}{r} \left(\frac{e + \cos f}{1 + e \cos f}\right) = \\
&= -\vec{i}_e \frac{\mu}{h} \sin f + \vec{i}_p \frac{\mu}{h} \left(e + \cos f\right).
\end{aligned} \tag{5.21}$$

Rewriting eqns. 5.17 and 5.21 in matrix form, we have

$$\begin{pmatrix} \vec{r} \\ \vec{v} \end{pmatrix} = A \begin{pmatrix} \vec{i}_e \\ \vec{i}_p \end{pmatrix}, \quad \text{where } A = \begin{pmatrix} r \cos f & r \sin f \\ -\frac{\mu}{h} \sin f & \frac{\mu}{h} \left(e + \cos f\right) \end{pmatrix}. \tag{5.22}$$

We note that the determinant of the matrix A is, considering eq. 5.19,

$$\begin{aligned} \det A &= r\frac{\mu}{h}(e+\cos f)\cos f + r\frac{\mu}{h}\sin^2 f = \\ &= r\frac{\mu}{h}(1+e\cos f) = r\frac{\mu}{h}\frac{p}{r} = h. \end{aligned} \tag{5.23}$$

Let f_i be the initial true anomaly of a particle in the spherical electrostatic deflector. Then, from the equation of orbit in eq. 5.16,

$$e\cos f_\mathrm{i} = \frac{p}{r_\mathrm{i}} - 1, \tag{5.24}$$

where r_i is the initial radius.

Taking the scalar product of initial position $\vec{r}_\mathrm{i}$ and initial velocity $\vec{v}_\mathrm{i}$, we obtain from eq. 5.22 that

$$\begin{aligned} \vec{r}_\mathrm{i}\cdot\vec{v}_\mathrm{i} &= -r_\mathrm{i}\frac{\mu}{h}\sin f_\mathrm{i}\cos f_\mathrm{i} + r_\mathrm{i}\frac{\mu}{h}\sin f_\mathrm{i}(e+\cos f_\mathrm{i}) = \\ &= r_\mathrm{i}\frac{\mu}{h}e\sin f_\mathrm{i}. \end{aligned}$$

Hence,

$$e\sin f_\mathrm{i} = \frac{\sqrt{p}\sigma_0}{r_\mathrm{i}}, \tag{5.25}$$

where σ_0 is defined as [9, p. 130]

$$\sigma_0 = \frac{\vec{r}_\mathrm{i}\cdot\vec{v}_\mathrm{i}}{\sqrt{\mu}}.$$

Let $\theta = f_\mathrm{f} - f_\mathrm{i}$ be the true anomaly difference between the final and initial positions. Applying eqns. 5.24 and 5.25 to the equation of orbit in eq. 5.16 yields the final radius

$$\begin{aligned} r_\mathrm{f} &= \frac{p}{1+e\cos f_\mathrm{f}} = \frac{p}{1+e\cos(f_\mathrm{i}+\theta)} = \\ &= \frac{p}{1+\cos\theta\cos f_\mathrm{i} - \sin\theta\sin f_\mathrm{i}} = \\ &= \frac{p}{1+\left(\frac{p}{r_\mathrm{i}}-1\right)\cos\theta - \frac{\sqrt{p}\sigma_0}{r_\mathrm{i}}\sin\theta} = \\ &= r_\mathrm{i}\frac{p}{r_\mathrm{i}+(p-r_\mathrm{i})\cos\theta - \sqrt{p}\sigma_0\sin\theta}. \end{aligned} \tag{5.26}$$

Solving eq. 5.22 for the basis vectors $\left(\vec{i}_e, \vec{i}_p\right)$ of the heliocentric coordinate system in terms of the initial position $\vec{r}_\mathrm{i}$, velocity $\vec{v}_\mathrm{i}$, and true anomaly f_i, considering that $\det A = h$ according to eq.

5.23, we have

$$\begin{aligned}\begin{pmatrix} \vec{i}_e \\ \vec{i}_p \end{pmatrix} &= A_{\mathrm{i}}^{-1} \begin{pmatrix} \vec{r}_{\mathrm{i}} \\ \vec{v}_{\mathrm{i}} \end{pmatrix} = \frac{1}{\det A} \begin{pmatrix} \frac{\mu}{h}(e+\cos f_{\mathrm{i}}) & -r_{\mathrm{i}} \sin f_{\mathrm{i}} \\ \frac{\mu}{h} \sin f_{\mathrm{i}} & r_{\mathrm{i}} \cos f_{\mathrm{i}} \end{pmatrix} \begin{pmatrix} \vec{r}_{\mathrm{i}} \\ \vec{v}_{\mathrm{i}} \end{pmatrix} = \\ &= \begin{pmatrix} \frac{\mu}{h^2}(e+\cos f_{\mathrm{i}}) & -\frac{r_{\mathrm{i}}}{h} \sin f_{\mathrm{i}} \\ \frac{\mu}{h^2} \sin f_{\mathrm{i}} & \frac{r_{\mathrm{i}}}{h} \cos f_{\mathrm{i}} \end{pmatrix} \begin{pmatrix} \vec{r}_{\mathrm{i}} \\ \vec{v}_{\mathrm{i}} \end{pmatrix}. \end{aligned} \tag{5.27}$$

An analytic expression of the final position $\vec{r}_{\mathrm{f}}$ and velocity $\vec{v}_{\mathrm{f}}$ in terms of the initial position $\vec{r}_{\mathrm{i}}$ and velocity $\vec{v}_{\mathrm{i}}$ is obtained by inserting eq. 5.27 in eq. 5.22, which gives

$$\begin{aligned}\begin{pmatrix} \vec{r}_{\mathrm{f}} \\ \vec{v}_{\mathrm{f}} \end{pmatrix} &= A_{\mathrm{f}} A_{\mathrm{i}}^{-1} \begin{pmatrix} \vec{r}_{\mathrm{i}} \\ \vec{v}_{\mathrm{i}} \end{pmatrix} = \begin{pmatrix} r_{\mathrm{f}} \cos f_{\mathrm{f}} & r_{\mathrm{f}} \sin f_{\mathrm{f}} \\ -\frac{\mu}{h} \sin f_{\mathrm{f}} & \frac{\mu}{h}(e+\cos f_{\mathrm{f}}) \end{pmatrix} \cdot \\ &\cdot \begin{pmatrix} \frac{\mu}{h^2}(e+\cos f_{\mathrm{i}}) & -\frac{r_{\mathrm{i}}}{h} \sin f_{\mathrm{i}} \\ \frac{\mu}{h^2} \sin f_{\mathrm{i}} & \frac{r_{\mathrm{i}}}{h} \cos f_{\mathrm{i}} \end{pmatrix} \begin{pmatrix} \vec{r}_{\mathrm{i}} \\ \vec{v}_{\mathrm{i}} \end{pmatrix} = \\ &= \begin{pmatrix} F & G \\ F_t & G_t \end{pmatrix} \begin{pmatrix} \vec{r}_{\mathrm{i}} \\ \vec{v}_{\mathrm{i}} \end{pmatrix}. \end{aligned} \tag{5.28}$$

The matrix on the left is usually referred to as the transition matrix. Note that this is not the same as the so-called transfer matrix of beam physics, because the transition matrix contains all nonlinear effects by virtue of its elements on orbit parameters. Specifically, we obtain for the transition matrix elements: the transition matrix element F is

$$\begin{aligned} F &= \frac{\mu}{h^2} r_{\mathrm{f}} [\cos f_{\mathrm{f}} (e+\cos f_{\mathrm{i}}) + \sin f_{\mathrm{f}} \sin f_{\mathrm{i}}] = \\ &= \frac{\mu}{h^2} r_{\mathrm{f}} (e \cos f_{\mathrm{f}} + \cos\theta) = \\ &= \frac{\mu}{h^2} r_{\mathrm{f}} \left[\left(\frac{p}{r_{\mathrm{f}}} - 1 \right) + \cos\theta \right] = \\ &= 1 - \frac{r_{\mathrm{f}}}{p} (1 - \cos\theta), \end{aligned}$$

the transition matrix element F_t is

$$F_t = \frac{\mu^2}{h^3} [-\sin f_{\mathrm{f}} (e+\cos f_{\mathrm{i}}) + (e+\cos f_{\mathrm{f}}) \sin f_{\mathrm{i}}] =$$

$$
\begin{aligned}
&= \frac{\mu^2}{h^3}\left[e\left(\sin f_\mathrm{i} - \sin f_\mathrm{f}\right) - \sin\theta\right] = \\
&= \frac{\mu^2}{h^3}\left[e\left(\sin f_\mathrm{i} - \sin f_\mathrm{i}\cos\theta - \sin\theta\cos f_\mathrm{i}\right) - \sin\theta\right] = \\
&= \frac{\mu^2}{h^3}\left[e\sin f_\mathrm{i}\left(1-\cos\theta\right) - \sin\theta\left(1+e\cos f_\mathrm{i}\right)\right] = \\
&= \frac{\mu^2}{h^3}\left[\frac{\sqrt{p}\sigma_0}{r_\mathrm{i}}\left(1-\cos\theta\right) - \frac{p}{r_\mathrm{i}}\sin\theta\right] = \\
&= \frac{\sqrt{\mu}}{r_\mathrm{i}p}\left[\sigma_0\left(1-\cos\theta\right) - \sqrt{p}\sin\theta\right],
\end{aligned}
$$

the transition matrix element G is

$$
\begin{aligned}
G &= \frac{1}{h} r_\mathrm{f} r_\mathrm{i}\left(-\cos f_\mathrm{f}\sin f_\mathrm{i} + \sin f_\mathrm{f}\cos f_\mathrm{i}\right) = \\
&= \frac{r_\mathrm{f} r_\mathrm{i}}{h}\sin\theta = \frac{r_\mathrm{f} r_\mathrm{i}}{\sqrt{\mu p}}\sin\theta,
\end{aligned}
$$

and the transition matrix element G_t is

$$
\begin{aligned}
G_t &= \frac{\mu}{h^2} r_\mathrm{i}\left[\sin f_\mathrm{f}\sin f_\mathrm{i} + \left(e+\cos f_\mathrm{f}\right)\cos f_\mathrm{i}\right] = \\
&= \frac{\mu}{h^2} r_\mathrm{i}\left(e\cos f_\mathrm{i} + \cos\theta\right) = \\
&= \frac{\mu}{h^2} r_\mathrm{i}\left[\left(\frac{p}{r_\mathrm{i}} - 1\right) + \cos\theta\right] = \\
&= 1 - \frac{r_\mathrm{i}}{p}\left(1-\cos\theta\right).
\end{aligned}
$$

Thus, we obtained a transition matrix

$$
\Phi = \begin{pmatrix} F & G \\ F_t & G_t \end{pmatrix}
$$

that expresses the final coordinates $(\vec{r}_\mathrm{f}, \vec{v}_\mathrm{f})$ as a function of the initial coordinates $(\vec{r}_\mathrm{i}, \vec{v}_\mathrm{i})$ as

$$
\begin{pmatrix} \vec{r}_\mathrm{f} \\ \vec{v}_\mathrm{f} \end{pmatrix} = \Phi \begin{pmatrix} \vec{r}_\mathrm{i} \\ \vec{v}_\mathrm{i} \end{pmatrix} \tag{5.29}
$$

and comprises elements [9, pp. 128–131]

$$
F\left(\theta\right) = 1 - \frac{r_\mathrm{f}}{p}\left(1-\cos\theta\right),
$$

$$F_t(\theta) = \frac{\sqrt{\mu}}{r_\mathrm{i} p}\left[\sigma_0(1-\cos\theta) - \sqrt{p}\sin\theta\right],$$
$$G(\theta) = \frac{r_\mathrm{f} r_\mathrm{i}}{\sqrt{\mu p}}\sin\theta,$$
$$G_t(\theta) = 1 - \frac{r_\mathrm{i}}{p}(1-\cos\theta).$$

The elements F, F_t, G, and G_t of the transition matrix Φ are called Lagrange coefficients. The Lagrange coefficients F_t and G_t are simply time derivatives of F and G, respectively.

Applying the transfer matrix from eq. 5.29 to the initial position $\vec{v}_\mathrm{i}$ and velocity $\vec{p}_\mathrm{i}$ from eqns. 5.11 and 5.12, we obtain the final position $\vec{r}_\mathrm{f}$ and velocity $\vec{v}_\mathrm{f}$.

The final (x,a) coordinates are obtained from the final position

$$\vec{r}_\mathrm{f} = r_\mathrm{f}(\cos\theta_\mathrm{f}, \sin\theta_\mathrm{f})$$

and velocity $\vec{v}_\mathrm{f}$ as

$$(x_\mathrm{f}, a_\mathrm{f}) = \left(r_\mathrm{f} - r_0, \frac{\vec{v}_\mathrm{f}\cdot\vec{r}_\mathrm{f}}{v_0 r_\mathrm{f}}\right).$$

We calculated the DA transfer map of the electrostatic spherical deflector, in (x,a) beamline coordinates, with reference orbit radius $r_0 = 1\,\mathrm{m}$, using the transition matrix with elements as the Lagrange coefficients in terms of the true anomaly difference. This transfer map is listed in sec. 5.8.3.

The transfer map only depends on reference orbit radius r_0 and the central angle of the tracked sector of the electrostatic spherical deflector. As long as the reference orbit radius r_0 is kept the same by adjusting the voltages of the inner and outer shells of the deflector, the transfer map does not depend on the charged particle's kinetic energy, mass, or charge. Indeed, considering eqns. 5.19, 5.13, 5.11, and 5.18, the focal parameter p is

$$\begin{aligned} p &= \frac{h^2}{\mu} = \frac{|\vec{r}_\mathrm{i}\times\vec{v}_\mathrm{i}|^2}{v_0^2 r_0} = \\ &= \frac{(r_0+x_\mathrm{i})^2\left[v_\mathrm{i}^2 - (a_\mathrm{i}v_0)^2\right]}{v_0^2 r_0} = \\ &= \frac{(r_0+x_\mathrm{i})^2\left[v_0^2 - \frac{2\alpha}{m}\left(\frac{1}{r_0} - \frac{1}{r_0+x_\mathrm{i}}\right) - (a_\mathrm{i}v_0)^2\right]}{v_0^2 r_0} = \end{aligned}$$

$$= \frac{(r_0 + x_\mathrm{i})^2 \left[v_0^2 - 2v_0^2 r_0 \left(\frac{1}{r_0} - \frac{1}{r_0 + x_\mathrm{i}} \right) - (a_\mathrm{i} v_0)^2 \right]}{v_0^2 r_0} =$$
$$= \frac{(r_0 + x_\mathrm{i}) \left[2r_0 - \left(1 + a_\mathrm{i}^2 \right) (r_0 + x_\mathrm{i}) \right]}{r_0},$$

and the eccentricity e is [9, p. 116] such that, considering eqns. 5.11, 5.9, and 5.18,

$$1 - e^2 = p \left(\frac{2}{r} - \frac{v^2}{\mu} \right) = p \left(\frac{2}{r_0 + x_\mathrm{i}} - \frac{v_\mathrm{i}^2}{\mu} \right) =$$
$$= p \left[\frac{2}{r_0 + x_\mathrm{i}} - \frac{v_0^2 - 2v_0^2 r_0 \left(\frac{1}{r_0} - \frac{1}{r_0 + x_\mathrm{i}} \right)}{v_0^2 r_0} \right] =$$
$$= p \left[\frac{2}{r_0 + x_\mathrm{i}} - \frac{1 - 2r_0 \left(\frac{1}{r_0} - \frac{1}{r_0 + x_\mathrm{i}} \right)}{r_0} \right] =$$
$$= p \left[\frac{2}{r_0 + x_\mathrm{i}} - \frac{2r_0 - (r_0 + x_\mathrm{i})}{r_0 (r_0 + x_\mathrm{i})} \right] = \frac{p}{r_0}.$$

Thus, the focal parameter p and the eccentricity e depend only on reference orbit radius r_0 and initial beamline coordinates $(x_\mathrm{i}, a_\mathrm{i})$.

To obtain a DA-valued transfer map, we set the initial phase space coordinates as

$$(x_\mathrm{i}, a_\mathrm{i}) = (d_1, d_2) = (\mathtt{DA}(1), \mathtt{DA}(2)),$$

the respective DA generators [11, pp. 86–96]. We performed the calculations with the computation order 3.

5.8.3 Transfer Maps of the Spherical Deflector and Comparison

Here, we list and compare the DA transfer maps for particles passing through a 45° sector of the electrostatic spherical deflector, calculated

1. by integration of the ODEs of motion in polar laboratory coordinates using an 8th order Runge–Kutta–Verner integrator;

2. using the Kepler theory transition matrix with Lagrange-coefficients elements in terms of the true anomaly difference;

3. for *COSY INFINITY*'s built-in electrostatic spherical deflector element `ESP`; and

4. using the code sequence `ES` in the program *GIOS*.

In all cases, the reference orbit radius is $r_0 = 1\,\mathrm{m}$, and non-relativistic equations of motion were used. For definiteness, the particles in the bunch were set to kinetic energy 1 MeV, mass 1 amu, and charge $1\,e$; however, as noted above, this setting has no impact on the orbit geometry as long as the reference orbit radius is kept the same by adjusting the voltages of the inner and outer spherical shells of the electrostatic spherical deflector. For visual transfer map comparison purposes, the computation order 3 was used in all cases, except for *GIOS*, where the computation order 2 was used.

For each transfer map calculation case, we computed deviations from the conditions of symplecticity [11, pp. 155–159][115] for the first and second order aberration coefficients and motion in the x–a plane. These conditions of symplecticity are as follows:

$$\begin{aligned} g_1 &= (x|x)\,(a|a) - (a|x)\,(x|a) - 1 = 0, \\ g_2 &= (x|x)\,(a|xa) - (a|x)\,(x|xa) + (x|xx)\,(a|a) - (a|xx)\,(x|a) = 0, \\ g_3 &= (x|x)\,(a|aa) - (a|x)\,(x|aa) + (x|xa)\,(a|a) - (a|xa)\,(x|a) = 0, \end{aligned} \tag{5.30}$$

where an aberration coefficient $\left(z_i|z_{j_1}\cdots z_{j_n}\right)$ is the partial derivative

$$\left(z_i|z_{j_1}\cdots z_{j_n}\right) = \left(\frac{\partial^n \left(\mathscr{M}\left(z\right)\right)_i}{\partial z_{j_1}\cdots\partial z_{j_n}}\right)_{z=0}$$

of the i-th component of the respective transfer map $\mathscr{M}$ applied to a coordinate vector $z = (z_1\cdots z_{2m})$, and $2m$ is the number of phase space coordinates.

5.8.3.1 Integration of the ODEs in Polar Laboratory Coordinates

The transfer map from the integration of the ODEs of motion in polar laboratory coordinates using an 8th order Runge–Kutta–Verner integrator is as follows.

TRANSFER MAP OBTAINED IN LAB COORDINATES

BY INTEGRATION OF THE ODEs OF MOTION

X_f

I	COEFFICIENT	ORDER	EXPONENTS
1	0.7071067811878048	1	1 0 0
2	0.7071067811853606	1	0 1 0
3	−.4999999999982248	2	2 0 0
4	0.9999999999999907	2	1 1 0
5	0.2071067811860403	2	0 2 0
6	−.3535533905925685	3	3 0 0
7	0.6066017178148258E−01	3	1 2 0
8	0.2928932188134486	3	0 3 0

A_f

I	COEFFICIENT	ORDER	EXPONENTS
1	−.7071067811852640	1	1 0 0
2	0.7071067811878422	1	0 1 0
3	−.7071067811852740	2	0 2 0
4	−.3535533905924618	3	3 0 0
5	−1.060660171777971	3	1 2 0

DA coefficients with absolute values less than 10^{-11} are omitted.

The deviations g_1, g_2, and g_3 from the conditions of symplecticity listed in eq. 5.30 in this case were as follows:

$$g_1 = 0.8881784197001252 \times 10^{-15},$$

$$g_2 = -0.2120476675655668 \times 10^{-14},$$

$$g_3 = 0.1217396563490169 \times 10^{-14}.$$

A 4th order Runge–Kutta integrator version of the *COSY INFINITY* code for calculation of this transfer map by integration of the ODEs in the polar laboratory coordinates is listed in App. H.1.

5.8.3.2 Kepler Theory Transition Matrix

The transfer map obtained in laboratory coordinates using the Kepler theory transition matrix with Lagrange-coefficients elements is as follows.

TRANSFER MAP OBTAINED IN LAB COORDINATES

USING LAGRANGE COEFFICIENTS

X_f

I	COEFFICIENT	ORDER	EXPONENTS
1	0.7071067811865475	1	1 0 0
2	0.7071067811865475	1	0 1 0
3	−.5000000000000000	2	2 0 0
4	1.000000000000000	2	1 1 0
5	0.2071067811865475	2	0 2 0
6	−.3535533905932737	3	3 0 0
7	0.606601717798212 2E−01	3	1 2 0
8	0.2928932188134523	3	0 3 0

A_f

I	COEFFICIENT	ORDER	EXPONENTS
1	−.7071067811865475	1	1 0 0
2	0.7071067811865476	1	0 1 0
3	−.7071067811865475	2	0 2 0
4	−.3535533905932737	3	3 0 0
5	−1.060660171779821	3	1 2 0

DA coefficients with absolute values less than 10^{-11} are omitted.

The deviations g_1, g_2, and g_3 from the conditions of symplecticity listed in eq. 5.30 in this case

were as follows:

$$g_1 = -0.1110223024625157 \times 10^{-15},$$
$$g_2 = 0.3119771259853192 \times 10^{-16},$$
$$g_3 = 0.1110223024625157 \times 10^{-15}.$$

5.8.3.3 *COSY INFINITY*'s Spherical Deflector Element

The transfer map of *COSY INFINITY*'s built-in electrostatic spherical deflector element `ESP`, obtained using non-relativistic equations of motion, is as follows.

```
TRANSFER MAP OF COSY INFINITY'S ESP ELEMENT
X_f
      I   COEFFICIENT                ORDER  EXPONENTS
      1   0.7071067811865475           1    1  0   0
      2   0.7071067811865475           1    0  1   0
      3   -.4999999999999999           2    2  0   0
      4    1.000000000000000           2    1  1   0
      5   0.2071067811865475           2    0  2   0
      6   -.3535533905932738           3    3  0   0
      7   0.6066017177982123E-01       3    1  2   0
      8   0.2928932188134525           3    0  3   0
      ------------------------------------------------
A_f
      I   COEFFICIENT                ORDER  EXPONENTS
      1   -.7071067811865475           1    1  0   0
      2   0.7071067811865475           1    0  1   0
      3   -.7071067811865475           2    0  2   0
      4   -.3535533905932737           3    3  0   0
      5   -1.060660171779821           3    1  2   0
```

The deviations g_1, g_2, and g_3 from the conditions of symplecticity listed in eq. 5.30 were as follows:

$$g_1 = -0.2220446049250313 \times 10^{-15},$$
$$g_2 = 0.2220446049250313 \times 10^{-15},$$
$$g_3 = 0.3330669073875470 \times 10^{-15}.$$

The *COSY INFINITY* code for calculation of this transfer map of the `ESP` element is listed in App. H.4.

5.8.3.4 *GIOS*'s Spherical Deflector Element

The transfer map of the electrostatic spherical deflector computed using the code sequence `ES` in the program *GIOS* is as follows.

TRANSFER MAP COMPUTED USING THE PROGRAM GIOS

X_f

I	COEFFICIENT	ORDER	EXPONENTS
1	0.7071067812	1	1 0 0
2	0.7071067812	1	0 1 0
3	−.5000000000	2	2 0 0
4	0.2928932188	2	1 1 0
5	0.2071067812	2	0 2 0

A_f

I	COEFFICIENT	ORDER	EXPONENTS
1	−.7071067812	1	1 0 0
2	0.7071067812	1	0 1 0
3	−.5000000000	2	2 0 0
4	−.7071067812	2	1 1 0
5	−.2071067812	2	0 2 0

The deviations g_1, g_2, and g_3 from the conditions of symplecticity listed in eq. 5.30 were as follows:

$$g_1 = 0.3804934145534844 \times 10^{-10},$$
$$g_2 = -0.2928932188380493,$$
$$g_3 = 0.7071067812000000.$$

We note that the deviations g_2 and g_3 are significant in magnitude and indicate error(s) in the program *GIOS*.

We also note that these differences are not due to the fact that *GIOS* uses momentum-like coordinates that differ from those of *COSY INFINITY*. The respective effects manifest themselves only in order three in x and a terms, which we are not comparing here.

5.8.4 Calculation Results, Comparison, and Conclusion

The electrostatic spherical deflector transfer maps calculated in laboratory coordinates by integration of the ODEs and using the Kepler theory transition matrix are in good agreement with the transfer map of *COSY INFINITY*'s built-in electrostatic spherical deflector element `ESP`. The deviations from the conditions of symplecticity g_1, g_2, and g_3 were $\sim 10^{-17}$ to $\sim 10^{-16}$ in case of calculation using the Kepler theory transition matrix, $\sim 10^{-16}$ in case of *COSY INFINITY*'s built-in electrostatic spherical deflector element `ESP`, and $\sim 10^{-15}$ in case of integration of the ODEs in polar laboratory coordinates.

The transfer map of the electrostatic spherical deflector computed using the program *GIOS* significantly disagrees with the other three transfer maps. However, the deviations g_2 and g_3 in the *GIOS* case were also significant in magnitude and indicate that the disagreement is due to error(s) in the program *GIOS*.

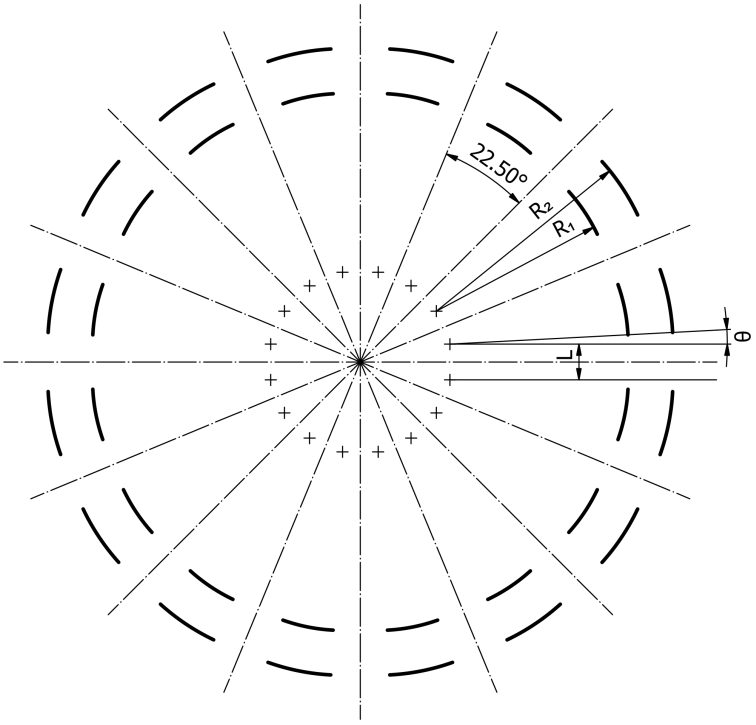


Figure 5.16: The Metodiev variant of the Proton EDM LSCT lattice. The lattice comprises 16 sections. Each section contains an electrostatic cylindrical deflector, denoted by the continuous wide lines. Geometrical parameters: $R_1 = 39.985\,\mathrm{m}$, $R_2 = 40.015\,\mathrm{m}$, $\alpha = \pi/8 - 2\theta$, and $L = 2\,\mathrm{m}$.

5.9 Proton EDM Lattice

5.9.1 Lattice Geometry

A Proton EDM Long Spin Coherence Time (LSCT) was proposed by the Storage Ring EDM collaboration in [3, incl. App. RingLat and LattParams]. Here, we consider a variant of this Proton EDM LSCT lattice that was defined and studied by Eric Metodiev *et al.* [78].

The lattice uses protons at the magic momentum $p_0 = 0.7\,\mathrm{GeV/c}$, at which, without an EDM signal, the proton's spin vector would be aligned with its momentum, and an EDM would rotate the spin vector out of the midplane. The lattice, shown in Fig. 5.16, consists of 16 cylindrical electrostatic deflectors with radial electrostatic field $10.5\,\mathrm{MV/m}$, curvature radius $R_0 = 40\,\mathrm{m}$, and arc angle $\alpha = \pi/8 - 2\theta$, separated by drifts. Each drift is defined as two arcs of radius $R_0 = 40\,\mathrm{m}$ and small angle θ, separated by a standard straight drift of length $L = 2\,\mathrm{m}$. The gap D between the plates of the electrostatic cylindrical deflector, that is, the full aperture, is $D = 3\,\mathrm{cm}$, with the plates

at radii $R_1 = 39.985\,\mathrm{m}$ and $R_2 = 40.015\,\mathrm{m}$.

The arc-shaped drifts of small parametric angle θ were optimized to $\theta = 1\,\mathrm{mrad}$ in [78] to account for the curvature of the reference orbit due to the fringe fields of the cylindrical electrostatic deflectors. This is an alternative approach to the standard practice of delimiting particle optical elements by their EFBs and not by their physical edges. *COSY INFINITY* conforms to this standard practice and encapsulates the treatment of each element's fringe field within that element.

5.9.2 Electrostatic Cylindrical Deflector

5.9.2.1 Semi-Circular Geometry

First, we consider an electrostatic cylindrical deflector with inner radius $\rho = R_1$ and outer radius $\rho = R_2$, spanning the arc $-\pi \leq \psi \leq 0$ [78] in a polar coordinate system (ρ, ψ). Let $-V_0$ and V_0 be the voltages of the inner and outer plates of this deflector, respectively.

To solve the Laplace equation and to obtain the electrostatic potential of the electrostatic cylindrical deflector, we use the conformal mapping $(\rho, \psi) = f(u, v)$, [78, p. 2]

$$\begin{aligned} \rho &= \sqrt{R_1 R_2} \exp\left[\frac{1}{2\pi} \ln\frac{R_2}{R_1} (e^u \sin v + v)\right], \\ \psi &= \frac{1}{2\pi} \ln\frac{R_2}{R_1} (1 + e^u \cos v + u) \end{aligned} \tag{5.31}$$

from the canonical domain as the semi-infinite strip $-\infty < u < \infty$, $-\pi \leq v \leq \pi$ to the physical domain with the deflector geometry. This can be equivalently rewritten as

$$w = \frac{1}{2\pi} \ln\left(\frac{R_2}{R_1}\right) (1 + z + e^z), \tag{5.32}$$

where $w = \psi + i \ln(\rho/\sqrt{R_1 R_2})$ and $z = u + iv$.

Mapping the boundary conditions from the physical domain to the canonical domain, the voltage at $v = -\pi$ is $-V_0$, and the voltage at $v = \pi$ is V_0. The solution of the Laplace equation in the canonical domain is trivial: $\varphi_0(v) = V_0 v/\pi$.

The inverse conformal mapping $z = f^{-1}(w)$ is expressed, using the Lambert W function (defined by $z = W(z)\exp(W(z))$), as [78, p. 2]

$$z = g(w) = -1 + \frac{2\pi w}{\ln\left(\frac{R_2}{R_1}\right)} - W_{\kappa(w)}\left[\exp\left(-1 + \frac{2\pi w}{\ln\left(\frac{R_2}{R_1}\right)}\right)\right], \tag{5.33}$$

where $\kappa(w)$ is the branch of the Lambert W function

$$\kappa(w) = \left\lceil \Im\left(\frac{w}{\ln\left(\frac{R_2}{R_1}\right)} - \frac{1}{2}\right)\right\rceil.$$

We obtain the electrostatic potential in the physical domain as

$$\varphi_0(w) = V_0 \Im(g(w))/\pi,$$

with g from eq. 5.33 and with $w = \psi + i\ln(\rho/\sqrt{R_1 R_2})$.

5.9.2.2 Generalized Geometry

We now consider an electrostatic cylindrical deflector with inner radius $\rho = R_1$ and outer radius $\rho = R_2$ ($|R_1 - R_2| \ll R_1$), spanning the arc $-\alpha \leq \psi \leq 0$.

Well inside the deflector, the potential is practically s-independent, and the fringe field near the edge of the deflector is mostly formed, due to the $\sim 1/r$ shape of the potential, by the nearby parts of the deflector. Hence, the electrostatic potential $\varphi(\rho, \psi)$ of the electrostatic cylindrical deflector, near the reference orbit and the deflector, can be approximated by taking out a middle section of the semi-circular electrostatic cylindrical deflector and gluing together the remaining parts as

$$\varphi(\rho, \psi) = \begin{cases} \varphi_0(\rho, \psi) & , -\alpha/2 \leq \psi \leq \delta, \\ \varphi_0(\rho, \psi + \alpha - \pi) & , -\alpha - \delta \leq \psi \leq -\alpha/2, \end{cases}$$

where φ_0 is the electrostatic potential of a semi-circular electrostatic cylindrical deflector and $\delta \sim |R_1 - R_2|$.

However, for $\delta \sim |R_1 - R_2|$, we would then have a significant, unphysical discontinuity at $\psi = \delta$ and $\psi = -\alpha - \delta$ of the potential φ or the electrostatic field $E = -\nabla\varphi$, depending on the adjacent

elements. Because φ quickly falls off as the distance from the electrostatic cylindrical deflector increases and becomes relatively insignificant well inside the adjacent deflector, we cut the potential at $\delta = \pi - \alpha/2$.

5.9.2.3 Computation of the Scalar Value

In [78, p. 5], to avoid floating-point overflow or underflow, the value of $W \circ \exp$ in eq. 5.33 was obtained by

1. computing the Lambert W function using Haley's method for $\psi < 0$ and $R_1 \leq \rho \leq R_2$; and
2. computing the Wright function $\omega = W \circ \exp$ using Newton's method otherwise.

We had attempted this approach; however, we encountered same the type of floating-point issues it is intended to address.

As a solution, we reverted from eq. 5.33 to solving eq. 5.32 directly using *COSY INFINITY*'s optimization commands `FIT`, `ENDFIT`. We used the objective function

$$\texttt{OBJ}(w,z) = \left| w - \frac{1}{2\pi} \ln\left(\frac{R_2}{R_1}\right)(1+z+e^z) \right| \tag{5.34}$$

in the optimization.

The calculation of the electrostatic potential is thus performed for plate voltages $\pm 1\,\mathrm{V}$, and then scaled to satisfy the radial electrostatic field of $10.5\,\mathrm{MV/m}$ well inside the electrostatic deflector.

We tested the computation accuracy of the scalar (non-DA) value of φ, for $\alpha = \pi/8$, using this method in *COSY INFINITY* against evaluation of φ using *Mathematica*'s built-in implementation `LambertW` of the Lambert W function. In this accuracy test, the largest apparent error was of the order of $10^{-13}\%$.

5.9.2.4 Computation of the DA Value

For DA integration of the equations of motion or computation of the electrostatic field, it is useful to compute the DA value of the electrostatic potential.

To that end, we note that, from eq. 5.32,

$$1+z+e^z = \frac{2\pi}{\ln\left(\frac{R_2}{R_1}\right)} w,$$

thus

$$\nabla\left(1+z+e^z\right) = \left(1+e^z\right)\nabla z = \nabla\left(\frac{2\pi}{\ln\left(\frac{R_2}{R_1}\right)} w\right).$$

Further, we express z in fixed-point form as follows:

$$z = G(z) = \mathrm{cons}(z) + \nabla^{-1}\left[\frac{1}{1+\exp(z)}\nabla\left(\frac{2\pi}{\ln\left(\frac{R_2}{R_1}\right)} w\right)\right], \tag{5.35}$$

where z and w are DA values; $\mathrm{cons}(z)$ is the scalar value of z calculated using `FIT`, `ENDFIT` as detailed above; and ∇^{-1} is the DA inverse gradient, which computes a potential of its argument.

Eq. 5.35 satisfies the DA fixed-point theorem, and the sequence $z_{k+1} = G(z_k)$, for $k = 0, 1, \ldots$, converges to a fixed point z – the DA value of z from eq. 5.33 – in a finite number of steps [11, pp. 96–100]. With computation order n, this calculation completes in $n+1$ steps.

We implemented this method of computing the DA value of the electrostatic potential of the cylindrical electrostatic deflector in *MSURK89* as the following *COSY INFINITY*–script function:

```
FUNCTION YLVDA PSI RHO F0 ;
    VARIABLE LHS NM ; VARIABLE DVAR 1 ;
    VARIABLE J 1 ;
    VARIABLE F NM ; VARIABLE FD NM 2 ;
    VARIABLE LHS1 NM ; VARIABLE LHS2 NM ;
    F := F0 ;
    DVAR := LOG(R2/R1) ;
    LHS := 2*PI*(PSI+I*LOG(RHO/SQRT(R2*R1)))/DVAR ;
    LHS1 := DER(1,LHS) ;
    LHS2 := DER(2,LHS) ;
    LOOP J 1 ORDER+1 ;
        CO J ;
```

```
        FD(1) := LHS1/(1+EXP1(F)) ;
        FD(2) := LHS2/(1+EXP1(F)) ;
        POTI FD 2 F ;
        CO ORDER ;
        F := F0 + F ;
    ENDLOOP ;
ENDFUNCTION ;
```

5.9.3 Lattice Potential Model

Considering the $\sim 1/r$ falloff shape of the electrostatic potential, we modeled the Proton EDM LSCT lattice by superposition of the electrostatic potentials of two nearest electrostatic cylindrical deflectors at every point of the lattice.

For that purpose, we utilized a subsystem of two adjacent electrostatic cylindrical deflectors, and implemented its representation in *Mathematica* and, as a *COSY INFINITY* script, in *MSURK89*. The plates of one deflector ("primary deflector") are placed at $\rho \in \{R_1, R_2\}$, and $-\pi/8 \leq \psi \leq 0$. The other deflector ("adjacent deflector") and its potential are obtained using appropriate coordinate transformations, namely, a rotation, a translation, and a reflection. The subsystem comprising the primary and the adjacent deflector is shown in Fig. 5.17, and a plot of this subsystem's electrostatic potential is shown in Fig. 5.18.

5.9.4 Tracking in *MSURK89*

We implemented tracking of the Proton EDM LSCT lattice in *MSURK89*. Fig. 5.19 shows a 16-turn tracking picture for one sector of the lattice. Tracking begins and ends at the central cross section of an electrostatic cylindrical deflector (points A and F in Fig. 5.17). The largest value of the objective function $\varepsilon = \texttt{OBJ}(w,z)$ from eq. 5.34 encountered in this tracking was $\sim 10^{-13}$, as Fig. 5.20 shows.

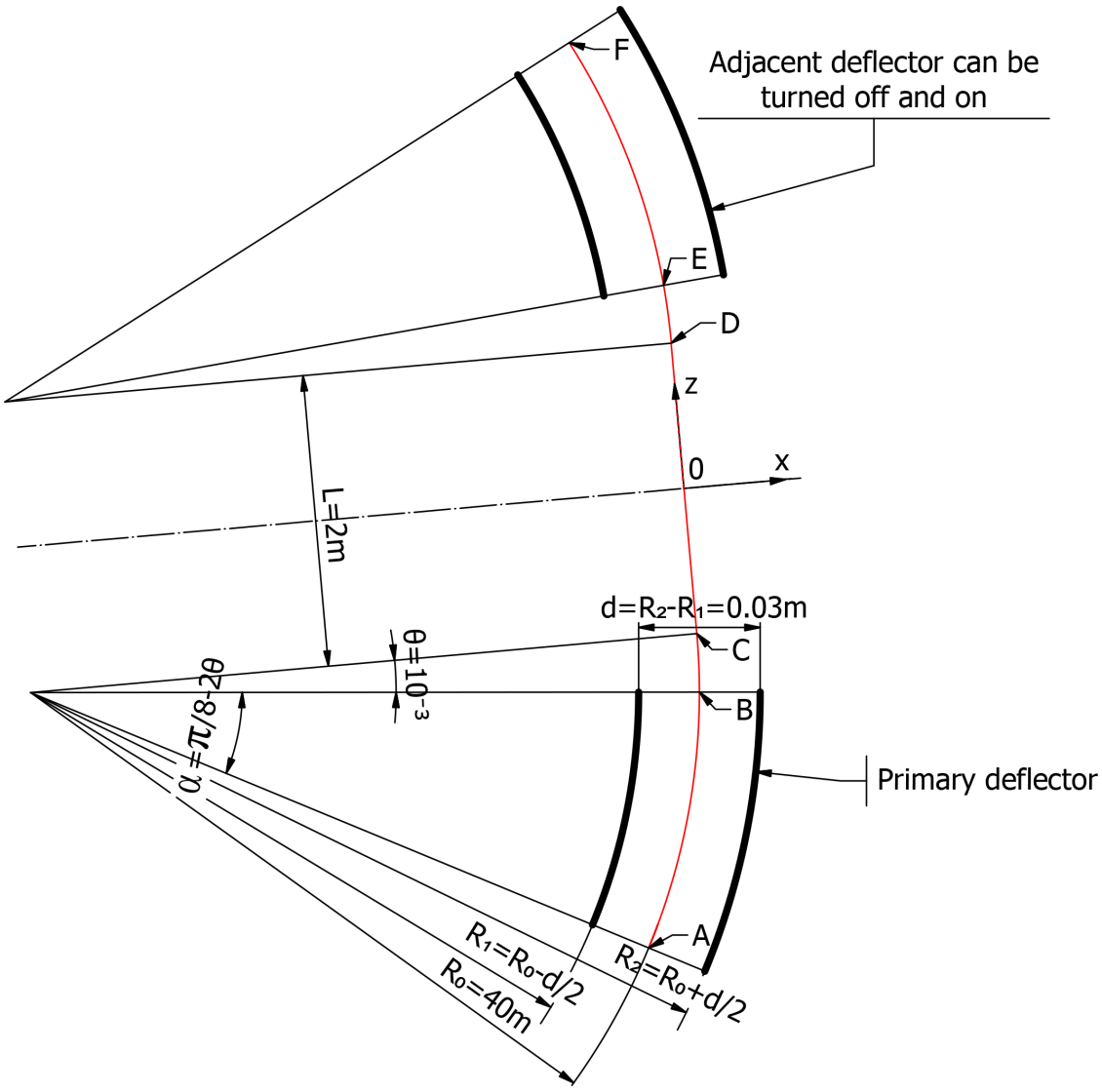


Figure 5.17: The subsystem of two adjacent electrostatic cylindrical deflectors in the Proton EDM LSCT lattice. The subsystem is bounded by the central cross sections of the deflectors. Geometrical parameters: $R_1 = 39.985\,\mathrm{m}$, $R_2 = 40.015\,\mathrm{m}$, $\alpha = \pi/8 - 2\theta$, and $L = 2\,\mathrm{m}$.

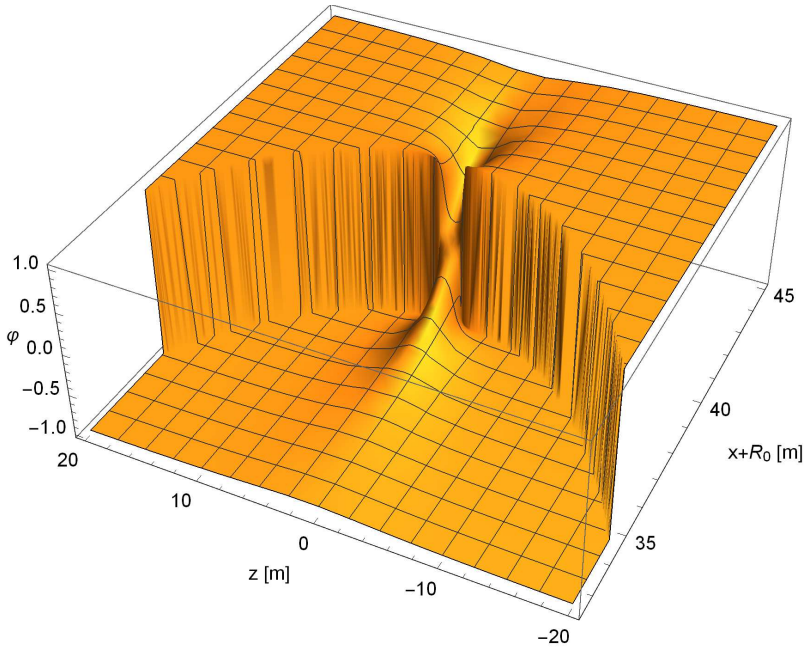


Figure 5.18: Electrostatic potential φ of the two-deflector subsystem, shown in Fig. 5.17. The electrostatic potential is scaled to ± 1 at the plates. The x–z coordinate system is the same as in Fig. 5.17.

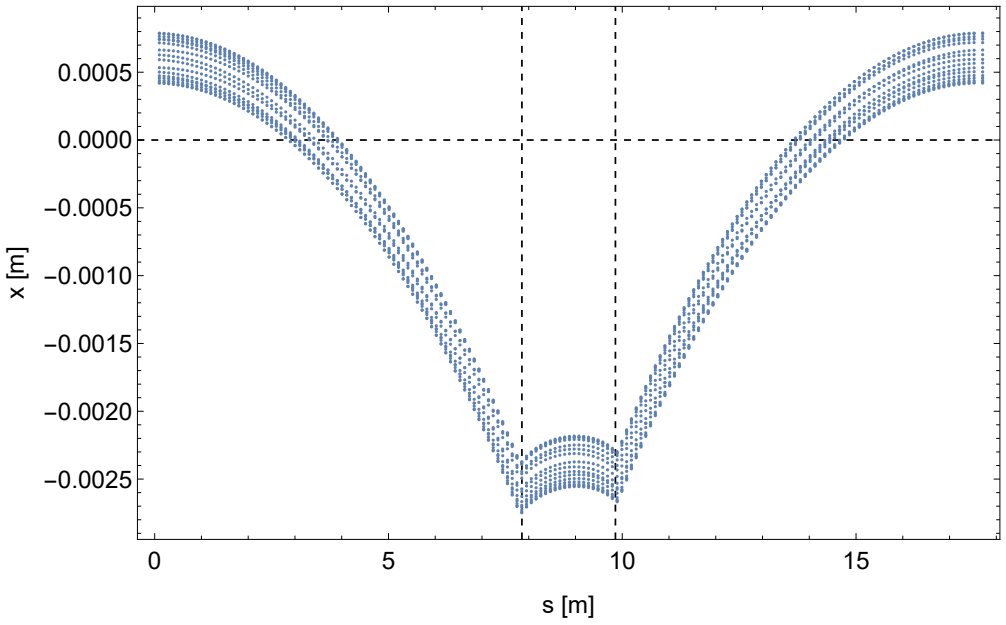


Figure 5.19: A particle launched with initial offset $x_{\mathrm{i}} = 7.43682 \times 10^{-4}$ m and momentum direction angle $\theta_x = 5.67880 \times 10^{-4}$ rad through 16 sectors (16 one-sector turns, or one full turn) of the Proton EDM LSCT lattice. The s coordinate is the distance along the path ABCDEF, and x is the horizontal local transversal coordinate. The two vertical gridlines indicate element edges between the arc-shaped drifts and the straight drift (approximately, the electrostatic cylindrical deflector edges). The asymmetry of the x minima may be explained by the formal use of arc-shaped drifts in the lattice model, whereas particles in drifts actually have straight-line trajectories.

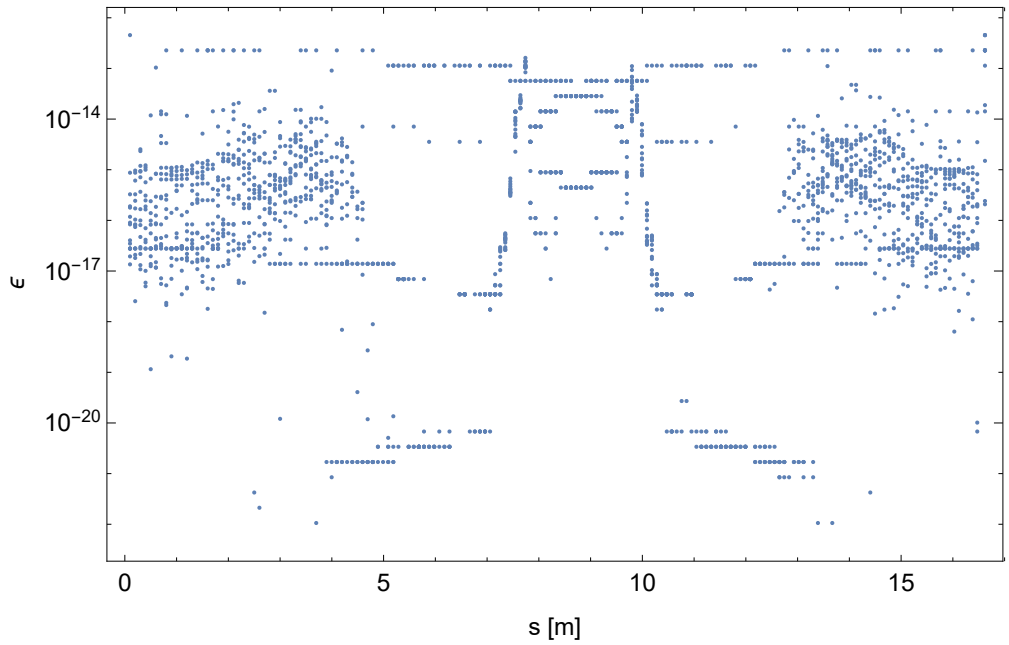


Figure 5.20: Values of the objective function $\varepsilon = \texttt{OBJ}(w,z)$ from eq. 5.34 encountered in the particle tracking shown in Fig. 5.19. The s coordinate is the distance along the path ABCDEF. The objective function ε values are indicative of the electrostatic potential calculation accuracy.

The objective function values are indicative of the electrostatic potential calculation accuracy and are ideally zero.

5.9.5 Lattice Implementation in *COSY INFINITY*

To implement the Proton EDM LSCT lattice in *COSY INFINITY*, we could consider specifying its particle optical elements, that is, the cylindrical electrostatic deflectors and the drifts, in one-to-one correspondence to the lattice specification from [78]. However, the issue with that approach is that there are no arc-shaped drift elements in *COSY INFINITY*. Furthermore, particle optical elements in *COSY INFINITY* are specified in Frenet–Serret beamline coordinates relative to the reference orbit, and, in this representation, an arc-shaped drift would be unphysical. A particle in a drift has a straight-line trajectory, and formally setting the reference trajectory in a drift as a circular arc would not satisfy the definition of reference trajectory that is foundational in *COSY INFINITY*, thus invalidating *COSY INFINITY* calculations for that element.

However, we can construct a custom particle optical element in *COSY INFINITY* by its specifying

its fields and using a valid reference orbit. We would need to specify the entry point of the reference trajectory for this element, which could be calculated in *COSY INFINITY* or otherwise. Following this approach, we could make a particle optical element spanning the space between the central cross sections of two adjacent electrostatic cylindrical deflectors. This would implicitly include the curved drifts while abstracting from their problematic details.

Continuing to analyze our problem statement, we recall that the arc-shaped drifts in [78] are used to account for fringe fields of the electrostatic cylindrical deflectors. However, *COSY INFINITY* already has rigorous and fully Maxwellian fringe field–modeling functionality (fringe field calculations mode `FR3`) that takes into account the impact of fringe fields on the reference orbit. Element lengths in *COSY INFINITY* are from EFB to EFB, not from physical edge to physical edge. It is a good idea to replace, not augment the arc-shaped drifts by *COSY INFINITY's* built-in fringe fields treatment.

Thus, we removed the arc-shaped elements from the model of the Proton EDM LSCT lattice, leaving 16 electrostatic cylindrical deflectors of (EFB-to-EFB) angle $\alpha = \pi/8$ separated by standard straight-line drifts of length $L = 2\,\mathrm{m}$. This is in good agreement with the Storage Ring EDM collaboration's Proton EDM LSCT lattice description [3, app. RingLat and LattParams].

5.9.5.1 Fringe Field of Electrostatic Cylindrical Deflector

Whereas the path ABCDEF in Fig. 5.17 is not a true reference orbit, it is within $\sim 10^{-3}\,\mathrm{m}$ of the reference orbit, as Fig. 5.19 indicates, and provides a reasonable approximation of the fringe field along the reference orbit. Fig. 5.21 shows the E_x component of the electrostatic field along the BCDE part of the path ABCDEF ("path BCDE").

As we intended to use separate particle optical elements for each of the electrostatic cylindrical deflectors and for the drift, we modeled the fringe field of one electrostatic cylindrical deflector, shown in Fig. 5.22. The EFB_{20} was $s_{\mathrm{EFB}}/D = 0.920939$ (see sec. 2.3.1 for definition of EFB denoted as EFB_{20}).

We note that this fringe field falloff is practically identical to the fringe field falloff of the

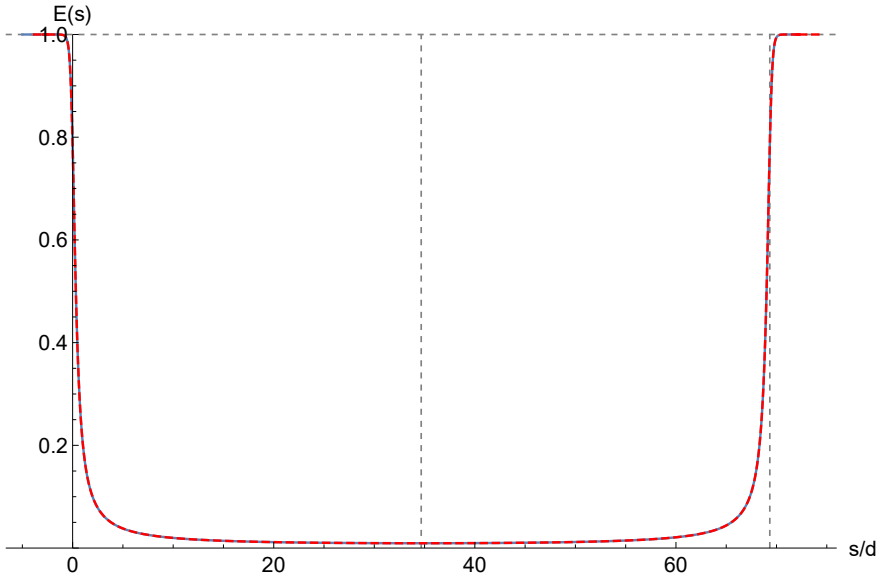


Figure 5.21: Blue color denotes the horizontal transversal component E_x of the electrostatic field of two adjacent electrostatic cylindrical deflectors along the path BCDE. The abscissa s/D is the distance s along the path ABCDEF, scaled by the distance D between the plates, and with the origin at point B (a deflector edge). The dashed-red curve is a reflection of the solid-blue curve relative to the midpoint between the deflectors at $s/D \approx 34.7$, demonstrating that the field satisfies the reflection symmetry.

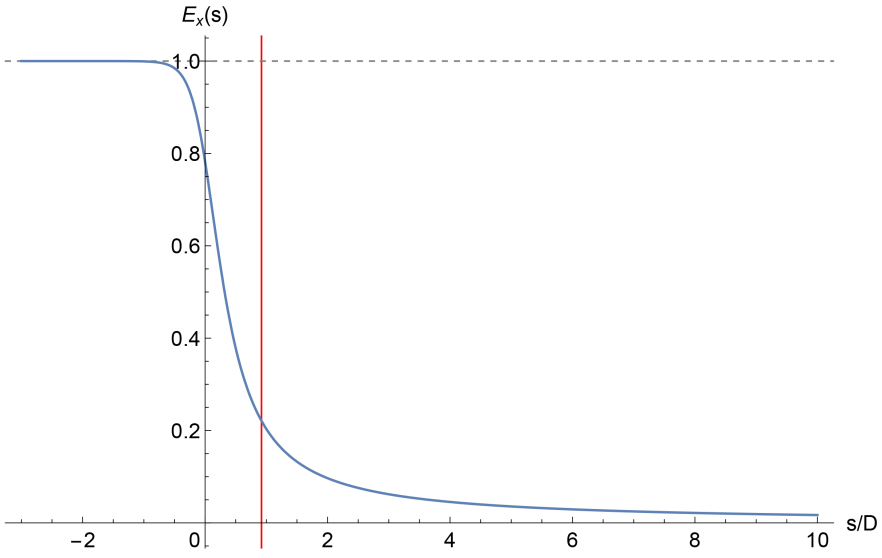
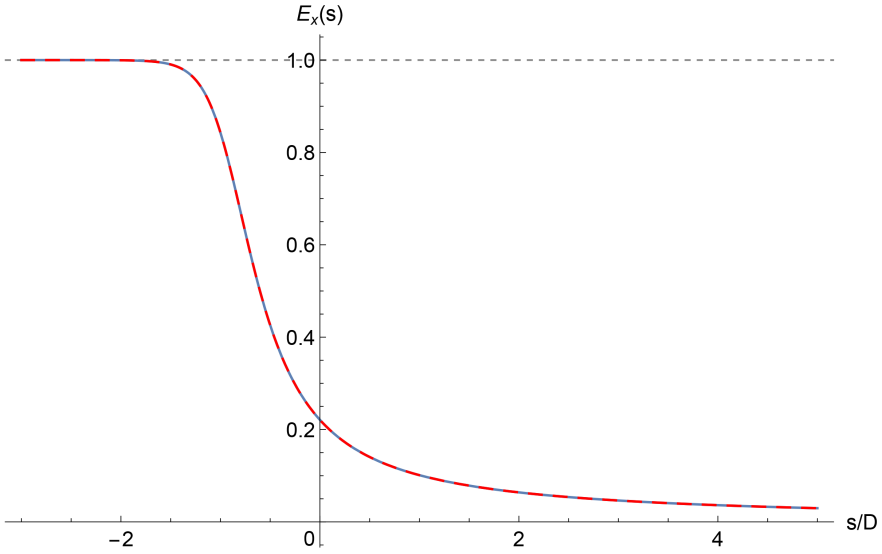
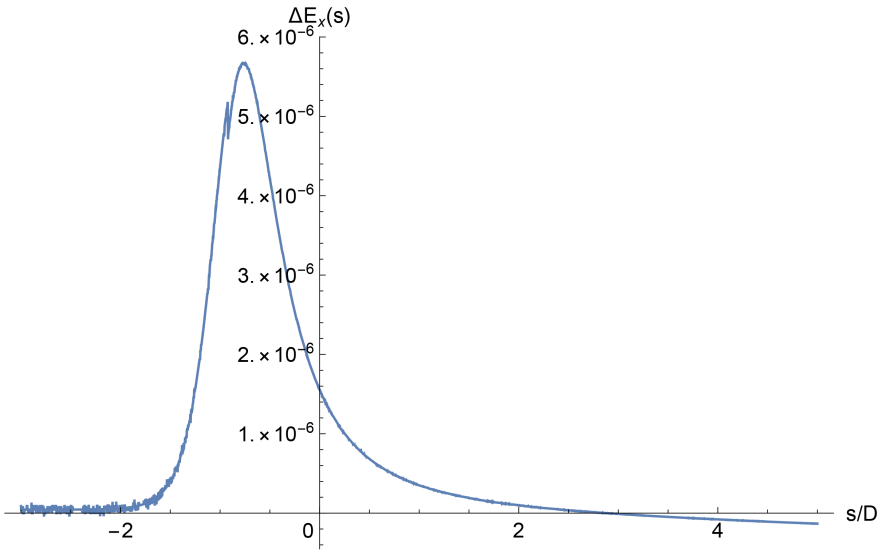


Figure 5.22: Fringe field of one electrostatic cylindrical deflector. The red vertical gridline at $s_{\mathrm{EFB}}/D = 0.920939$ denotes the EFB_{20}. The electrostatic field E_x is scaled to 1 well inside the deflector. The abscissa s/D is the distance s along the path ABCDEF, scaled by the full aperture d, and with the origin at point B (the deflector edge).



(a) Fringe field falloffs of the electrostatic cylindrical deflector (solid blue) and a semi-infinite capacitor with infinitely thin plates (dashed red).



(b) Difference between fringe field falloffs of (1) the electrostatic cylindrical deflector and (2) a semi-infinite capacitor with infinitely thin plates.

Figure 5.23: Fringe field falloffs of (1) the electrostatic cylindrical deflector and (2) a semi-infinite capacitor with infinitely thin plates. The electrostatic fields $E_x(s)$ are scaled to 1 well inside the deflectors. The falloffs are horizontally offset by the EFB_{20}.

Coefficient	Value
a_1	1.0614024399605924
a_2	1.6135741290714967
a_3	−0.9401447081042862
a_4	0.4781500036872176
a_5	−0.14379986967718494
a_6	0.017831089071215347

Table 5.1: Enge function coefficients of a semi-infinite capacitor with rounded edges of $D/20$ plate thickness. Here, we used these Enge function coefficients to model the fringe field of the electrostatic cylindrical deflector in the Proton EDM LSCT lattice.

semi-infinite capacitor with parallel, infinitely thin plates, as Fig. 5.23 shows. Therefore, we can use the results for fringe fields of semi-infinite capacitors to model the fringe field of the electrostatic cylindrical deflector.

In [78], the electrostatic cylindrical deflectors were modeled by infinitely thin plates at radii $\rho = R_1$ and $\rho = R_2$ in order to apply the conformal mapping specified in eq. 5.31. However, we had already obtained, using conformal mapping methods, fringe field falloffs of semi-infinite capacitors with plates of arbitrary thickness. The plate thickness of $D/20$ is quite realistic.

We used Enge function coefficients of a semi-infinite capacitor with rounded edges of $D/20$ plate thickness, listed in Table 5.1.

5.9.6 Tracking in *COSY INFINITY*

We encoded the Proton EDM LSCT lattice in *COSY INFINITY* as follows.

```
R := 40 ; PHI := 22.5 ; A := 0.015 ;


N1 := 1 ; N2 := -1 ; N3 := 1 ;
N4 := -1 ; N5 := 1 ;


A1 := 1.0614024399605924 ; A2 := 1.6135741290714967 ;
A3 := -0.9401447081042862 ; A4 := 0.4781500036872176 ;
```

```
A5 := -0.14379986967718494 ; A6 := 0.017831089071215347 ;
FC 1 1 2 A1 A2 A3 A4 A5 A6 ;
FC 1 2 2 A1 A2 A3 A4 A5 A6 ;

UM ;
LOOP J 1 16 ;
    DL L/2 ;
    ES R PHI A N1 N2 N3 N4 N5 ;
    DL L/2 ;
ENDLOOP ;
```

Variables `N1–N5` and `A1–A6` are the electrostatic cylindrical deflector's (1) inhomogeneity coefficients and (2) Enge function coefficients listed in Table 5.1 and derived in sec. 2.6, respectively. *COSY INFINITY*'s most accurate fringe field computations mode `FR3` was used. The lattice begins and ends at the midpoint of one of the drift elements.

We tracked protons at the magic momentum $p_0 = 0.7\,\mathrm{GeV/c}$ through this lattice for $16,000$ turns, with computation order 7. Phase space position projections to the *x–a* plane were drawn every 11 turns (co-prime with the 16-fold rotational symmetry). Relativistic equations of motion were used.

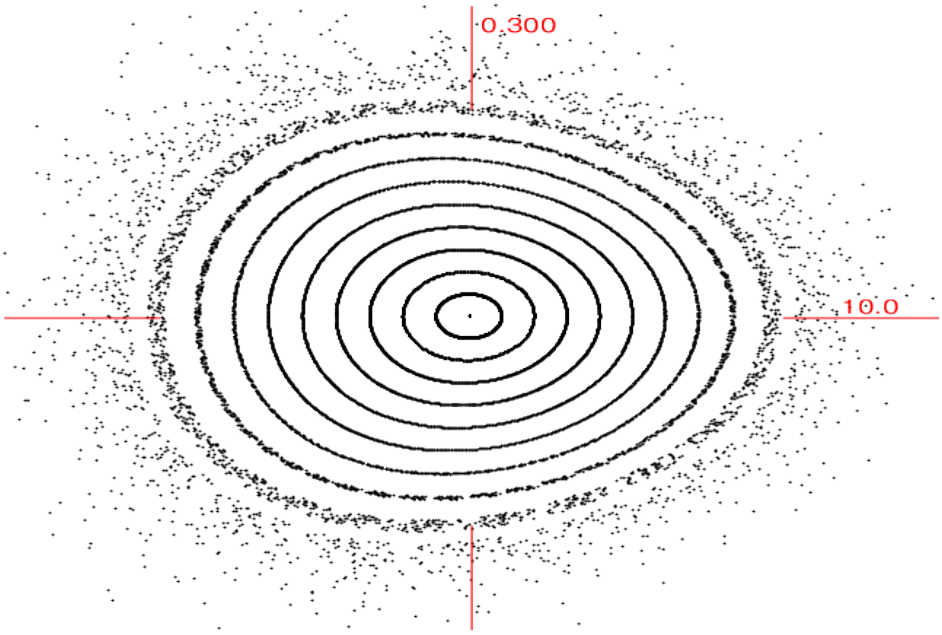
First, we tracked particle rays with initial x offsets within the physical aperture. The motion was highly linear, and there was practically no visible difference between tracking picture without symplectification and with EXPO symplectification.

To observe the nonlinear motion and the formal dynamic aperture, we eventually set the initial particle ray coordinates as

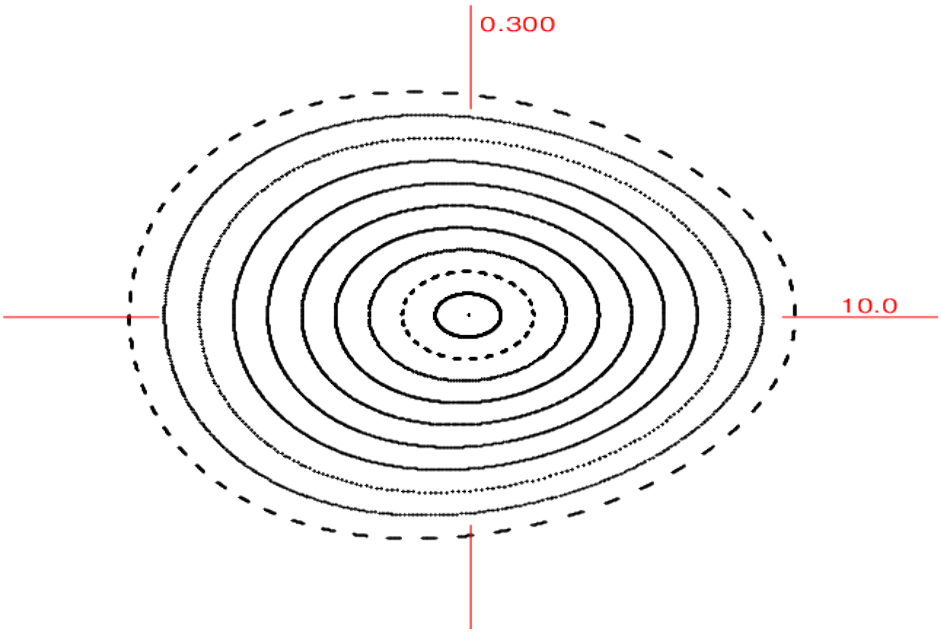
$$(x, a, b, y, l, \delta_K) = (k \cdot 0.7\,\mathrm{m}, 0, 0, 0, 0, 0)\,,$$

for $k = 1, 2, \ldots, 10$.

The resulting tracking pictures are shown in Fig. 5.24. In sub-figure (a), without symplectification, the outer particles gently fly off to infinity. In sub-figure (b), EXPO symplectification results



(a) Tracking without symplectification. Calculation CPU time: 14.7 s.



(b) Tracking with EXPO symplectification. Calculation CPU time: 13.2 s.

Figure 5.24: Proton EDM LSCT lattice: the x–a projections of particles tracked in *COSY INFINITY*, for 16,000 turns, with relativistic equations of motion. Particles were launched horizontally, with initial x offsets up to 7 m.

in the outer particles having quite a linear motion and apparently closed orbits in the x–a projection.

5.10 Conclusion

We benchmarked *COSY INFINITY* and *MSURK89* by tracking particles through an electrostatic spherical deflector and a homogeneous magnetic dipole for $160,000$ turns. *MSURK89* is our in-house eighth order Runge–Kutta–Verner tracking code that integrates individual orbits in each tracking turn, and we developed *MSURK89* in the context of this study. We studied integrator errors in *MSURK89* and optimized the initial integrator step size. Tracking was performed with relativistic and non-relativistic equations of motion, as well as with EXPO symplectification and without symplectification. The test-cases results are in good agreement with analytical solutions, especially for *COSY INFINITY* with EXPO symplectification and for *MSURK89*.

We described and performed the direct computation of the DA transfer map of an electrostatic spherical deflector LCS using simple ODEs and Kepler's theory. The results agree well with transfer maps of the electrostatic spherical deflector obtained using the built-in element of *COSY INFINITY*. The deviations from the conditions of symplecticity g_1, g_2, and g_3 were $\sim 10^{-17}$ to $\sim 10^{-16}$ in case of calculation using the Kepler theory transition matrix, $\sim 10^{-16}$ in case of *COSY INFINITY*'s built-in electrostatic spherical deflector element `ESP`, and $\sim 10^{-15}$ in case of integration of the ODEs in polar laboratory coordinates. However, the transfer map of the electrostatic spherical deflector computed using the program *GIOS* significantly disagrees with the other three transfer maps. The deviations g_2 and g_3 in *GIOS* were also significant in magnitude and indicated that the disagreement is due to error(s) in the program *GIOS*.

Finally, we modeled and tracked a variant [78] of the Proton EDM LSCT lattice [3]. In *MSURK89*, we implemented a particle optical element representing a 22.5° sector of this lattice that calculates the DA-valued electrostatic potential inside the element, and we tracked the lattice. The lattice, as defined in [78], uses short arc-shaped drifts to account for fringe fields of its electrostatic cylindrical deflectors. We replaced this construct by *COSY INFINITY*'s rigorous and fully Maxwellian fringe field treatment, and we tracked the lattice for $16,000$ turns.

CHAPTER 6

INVESTIGATION OF SPIN DECOHERENCE AND SYSTEMATIC ERRORS IN FROZEN SPIN AND QUASI-FROZEN SPIN LATTICES

6.1 Introduction

The quasi-frozen spin (QFS) method for measurement of the deuteron EDM was proposed by Yurij Senichev [98] as a modification of the frozen spin (FS) method. Whereas the FS method requires spin to be constantly aligned with momentum, the QFS method requires spin to be aligned with momentum only on average. Another advantage of the QFS method is that existing storage rings, such as the Cooler Synchrotron COSY at Forschungszentrum Jülich, can be adapted to satisfy the QFS method conditions.

We considered one FS and two QFS lattices in *COSY INFINITY*. For the FS and QFS methods to be feasible, the spin coherence time in the respective lattices must exceed the required EDM signal measurement time of 10^3 s. Our spin decoherence study shows that, with an RF cavity and optimized sextupole family strengths, the spin decoherence in these lattices often remains in the same range for at least 4.2×10^5 turns. This range is of the same order in the FS and QFS lattices.

Systematic errors are a fundamental limiting factor for EDM signal detection and measurement. Rotational magnet misalignments are especially relevant to EDM measurement, as B_x and B_z magnetic field components that arise from them rotate the spin vector out of the midplane, creating a fake EDM signal.

To address systematic errors due to rotational magnet misalignments, the following modification can be added to the FS and QFS methods: (1) particle bunches are tracked through the lattice in forward and reverse directions, using the horizontal plane to calibrate the magnetic field when switching the tracking direction; and (2) the EDM signal is extracted by taking the half-sum of the spin precessions in the vertical plane. This approach requires suppression of the B_z error field component using additional trim coils. Additionally, it is necessary that the vertical spin

decoherence due to systematic errors and the transversal and longitudinal beam emittance remain sufficiently low for EDM signal extraction from vertical spin precession.

To simulate rotational magnet misalignments in the FS and QFS lattices, we applied normally distributed spin kicks to the magnetic dipoles or the combined *E*+*B* field elements, as appropriate. The spin kick values were generated from uniformly distributed random numbers[1]. Vertical spin decoherence in the resulting lattices often remains in the same range (or grows within the range of a related spin decoherence curve) for at least 4.2×10^5 turns, and it is comparable in the FS and QFS structures.

6.2 Spin Coherence in Lattices for EDM

In beam physics, spin coherence refers to a measure or quality of preservation of polarization in an initially fully polarized beam. Methods for EDM or MDM signal measurement using storage rings, such as those introduced in Ch. 1, are based on the alignment of a particle's spin vector with its momentum if the measured quantity was zero, and they focus on spin dynamics relative to that condition. Ideally, all particles in a bunch would have the same orientation; however, in reality, there is always a spin decoherence effect. Consequently, it is necessary to minimize spin decoherence and to ensure that it is small enough for a meaningful measurement to be made. For definiteness, we generally consider spin polarization to be measurable if the spin decoherence is below 1 rad. The time during which spin coherence meets the requirements is called the spin coherence time (SCT).

6.2.1 Spin Coherence Requirements

Consider the reference particle in a storage ring. Its spin has an s-dependent invariant spin axis $\bar{n}(s)$ [90, p. 15][11, pp. 274–279], around which the spin precesses, and which is the stable direction in terms of polarization. Storage ring experiments with polarized beams typically align the spin vectors with $\bar{n}(s)$. In these cases, although the spin vectors relatively quickly decohere in terms of

[1] A random variable is called a random variate if it (1) satisfies a specific probability distribution and (2) is generated from uniformly distributed random numbers [8, Ch. 5].

phase in precessive motion around $\bar{n}(s)$, this does not affect the polarization along the $\bar{n}(s)$ vector, which has a larger polarization lifetime [90, p. 15].

However, EDM and MDM storage ring methods require spin polarization along the momentum vector, which is horizontal, whereas $\bar{n}(s)$ is vertical. Furthermore, not only spin coherence in terms of the phase of polar coordinate θ in the horizontal plane is crucial for the experiment, but alignment with the momentum vector needs to be maintained. This presents new challenges regarding spin coherence and SCT [90, p. 16][45].

In an FS lattice for EDM, the spin vectors would be rotated out of the horizontal plane due to EDM and a horizontal component E_x of the electrostatic field. According to the Thomas–BMT equation (introduced in sec. 1.3), the horizontal component E_x of the electrostatic field would result in a horizontal component $\Omega_{\mathrm{EDM},x}$ of the spin vector's EDM angular frequency,

$$\Omega_{\mathrm{EDM},x} = \eta \frac{qE_x}{2mc}, \tag{6.1}$$

where η is the EDM factor defined by

$$d = \eta \frac{q}{2mc} s,$$

which proportionally relates EDM d and spin s.

Considering that the target search region for neutron EDM has been narrowed down to $10^{-31}\, e{\cdot}\mathrm{cm} \lesssim |d_n| \lesssim 10^{-26}\, e{\cdot}\mathrm{cm}$ [89][84, p. 21] and the order-wise approximate equivalency of the deuteron EDM d_D and neutron EDM d_n according to dimensional analysis and SUSY considerations [3, pp. 7–9], suppose that $d_D \simeq 10^{-29}\, e{\cdot}\mathrm{cm}$. We remark that, likewise, the so-called nominal EDM is defined as $10^{-29}\, e{\cdot}\mathrm{cm}$ in [106, Ch. 8, p. 9] for protons and electrons.

This yields

$$\eta_D = \frac{2m_D c d_D}{\hbar s_D q_D} \simeq 10^{-15},$$

where $m_D = 2.01\,\mathrm{u}$ is the deuteron mass, $s_D = 1$ is the deuteron spin, and $q_D = 1\, e$ is the deuteron electric charge.

According to Newton's second law, for zero radial acceleration at the reference orbit, a cylindrical electrostatic deflector with a realistic bending radius $R_0 = 42.17\,\mathrm{m}$ must have an electrostatic field

$$E_x = \frac{\gamma m_D v_0^2}{q_D R_0} = 1.2 \times 10^7\,\frac{\mathrm{V}}{\mathrm{m}},$$

where $v_0 = 1.46 \times 10^8\,\mathrm{m/s}$ is the velocity of a deuteron at kinetic energy of 270 MeV. The kinetic energy of 270 MeV results in the maximal figure of merit (FOM; in this case, the product of the cross section by the analyzing power) for deuteron scattering on a carbon target [94][104, Ch. 5].

The resulting x component $\Omega_{\mathrm{EDM},x}$ of the spin vector's angular velocity due to EDM is, applying eq. 6.1,

$$\Omega_{\mathrm{EDM},x} = \eta_D \frac{q_D E_x}{2 m_D c} \simeq 10^{-9}\,\mathrm{rad/s}.$$

Therefore, the required SCT for deuteron spin to reach a measurable azimuthal angle $\varphi = 1\,\mu\mathrm{rad}$ would be $T \simeq 1000\,\mathrm{s}$. In contrast, the observation time in muon MDM experiments is limited by the muon lifetime $\tau_\mu = 2.2 \times 10^{-6}\,\mathrm{s}$ [43, p. 36], which is relativistically dilated to $\tau_\mu \gamma = 64.4 \times 10^{-6}\,\mathrm{s}$ in the E989 Experiment [43, p. 515].

6.2.2 General Principles of Spin Decoherence Optimization

Integration of the orbital equations of motion through one turn around the lattice in phase space as a vector space $({}_nD_v)^v$ over differential algebra ${}_nD_v$ [11, pp. 84–117] results in an orbital transfer map $\mathscr{M} \in ({}_nD_v)^v$. The orbital transfer map $\mathscr{M}$ expresses the orbital motion as

$$z_n = \mathscr{M}\left(z_{n-1}\right),$$

where z_{n-1} are the orbital coordinates at the beginning of the n-th turn around the lattice and z_n are the orbital coordinates at the end of said turn.

Similarly, integration of the Thomas–BMT equation results in a spin transfer map $\mathscr{S} \in \mathrm{SO}_3\left({}_nD_v\right)$, a 3×3 special orthogonal matrix with ${}_nD_v$-valued elements, which expresses the spin motion as

$$s_n = \mathscr{S}\left(z_{n-1}\right) s_{n-1},$$

where s_{n-1} is the spin vector at the beginning of the n-th turn around the lattice and s_n is the spin vector at the end of said turn.

Spin decoherence can be diminished by minimizing the injection-time beam size Δz_0 and spin decoherence Δs_0, as well as by optimizing the transfer maps $\mathscr{M}$ and $\mathscr{S}$ via the lattice composition and particle optical element parameters. The latter is achieved by first optimizing the first-order DA (or Taylor series expansion) effects, then optimizing the second-order effects, and sequentially proceeding to higher orders as necessary.

The sequential order of optimizing spin decoherence by effects of different orders is essential, as electrostatic and magnetic multipoles of an order k typically have significant higher-order components of orders $k+1$, $k+2$, and so on. On the other hand, lower-order components are insignificant and ideally zero, except under the presence of alignment errors, in which case their magnitude is typically still small in practice. Therefore, optimization of the k-th order of the orbital transfer map generally affects the higher orders while not disrupting the already completed optimization of the lower orders.

6.2.3 Suppression of Longitudinal Effects Using an RF Cavity

Radio-frequency (RF) cavities are particle optical elements, in which the electromagnetic field oscillates as a function of time. Most RF cavities are variations of the cylindrical pillbox cavity, with TM_{mnp} (transverse magnetic, conventionally enumerated by m, n, and p) modes specifying the electric field E and the transverse magnetic field B [13, pp. 241–242]. For example, in the fundamental mode TM_{010} [13, p. 242],

$$\begin{aligned}(E_r, E_\theta, E_z) &= \left(0, 0, E_0 J_0\left(\frac{x_{01} r}{R_\mathrm{c}}\right)\cos(\omega t)\right), \\ (B_r, B_\theta, B_z) &= \left(0, -\frac{E_0}{c} J_1\left(\frac{x_{01} r}{R_\mathrm{c}}\right)\sin(\omega t), 0\right)\end{aligned} \tag{6.2}$$

in the cylindrical coordinates (r, θ, z), where z is the longitudinal coordinate, the Frenet–Serret coordinates are $(x, y) = r(\cos\theta, \sin\theta)$, J_m is the Bessel function of the first kind of order m, R_c is the RF cavity radius, $x_{01} = 2.405$, and ω is the angular frequency of the RF cavity.

Conventionally, RF cavities are used for beam acceleration or deceleration. However, RF cavities are also instrumental in the suppression and averaging of longitudinal effects such as momentum deviation. This is accomplished by synchronizing the particle beam and the RF cavity so that the reference particle passes through the RF cavity when its electric field is zero, particles with a positive longitudinal offset are decelerated by the electric field, and particles with a negative longitudinal offset are accelerated by the electric field. Hence, the RF cavity's angular frequency is $\omega = h\omega_0$, where h is a natural number called the harmonic number and ω_0 is the beam angular frequency.

For example, consider a reference particle that passes through an RF cavity in the fundamental mode TM_{010}, with E and B fields as in eq. 6.2, at times $t = \pi(2k - 1/2)/\omega$, where $k \in \mathbb{Z}$. An off-reference particle of charge q and longitudinal offset $l = -\gamma v_0 \Delta t/(1+\gamma)$, where v_0 is the reference velocity and $\Delta t = t - t_0$ is the time-of-flight offset, is acted upon by longitudinal electric field

$$E_z(l) = -E_0 J_0\left(\frac{x_{01} r}{R_\mathrm{c}}\right) \sin\left(\frac{(1+\gamma)\,\omega l}{\gamma v_0}\right).$$

Provided that $qE_0 > 0$, the particle is accelerated (decelerated) in case of a small negative (positive, respectively) longitudinal offset l.

This results in elliptic dynamics of particles in the longitudinal l–δ_K plane in the time-of-flight range $-\pi < \omega\Delta t < \pi$, where l is the longitudinal offset and $\delta_K = (K - K_0)/K_0$ is the relative kinetic energy offset, for particles sufficiently close to the reference particle. By translational symmetry of period $T = 2\pi/\omega$, elliptic dynamics also occur in the time-of-flight ranges $-\pi + 2\pi k < \omega\Delta t < \pi + 2\pi k$ for $k \in \mathbb{Z}$ other than $k = 0$. Another, qualitatively distinct regime is that if the relative kinetic energy offset δ_K of a particle is too large, the particle has an unstable motion and eventually slips away to $|l| = \infty$.

The boundary separating stable, elliptic dynamics of particles in the l–δ_K plane from other modes of behavior is called the separatrix. An example of a separatrix is shown in Fig. 6.1 (see also Figs. 9.5–7 in [113, pp. 269–271]).

The length of an RF cavity can be estimated [94] by the quarter wavelength $L = c/4\nu_\mathrm{RF}$. This

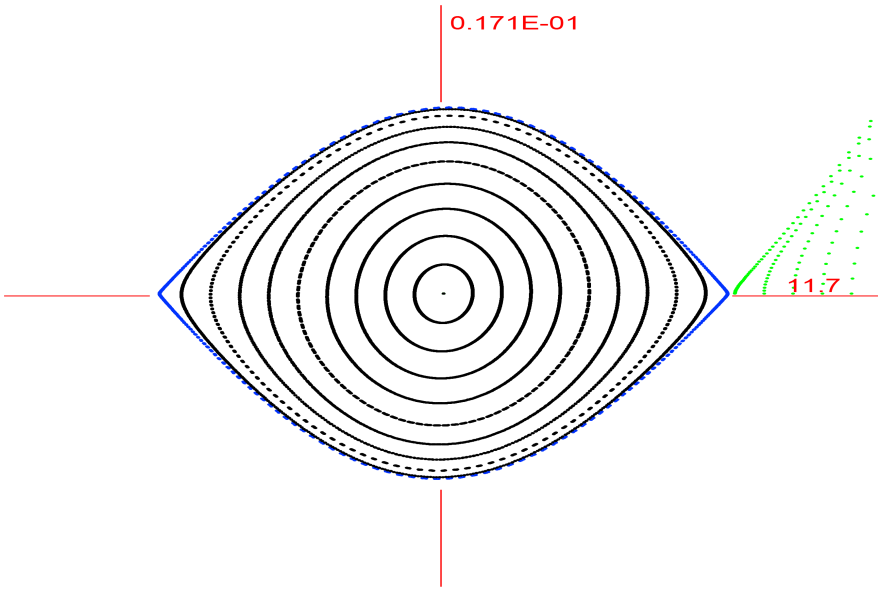


Figure 6.1: The separatrix (blue) of the *Senichev BNL* FS lattice (introduced in sec. 6.3) with the RF cavity voltage $V = 100\,\mathrm{kV}$, the RF cavity harmonic number $h = 5$, and fringe field calculations off. Particles are launched with initial l offsets up to 12.45 m and tracked in the l–δ_K plane. Particles within the separatrix (black) have ellipsoidal orbits centered at $(l, \delta_K) = (0, 0)$, whereas the five particles outside the separatrix (green), by translational symmetry in l, should have elliptic dynamics centered at approximately $(l, \delta_K) = (17\,\mathrm{m}, 0)$. However, these five particles appear to have unstable motion because of the decreasing accuracy of a DA representation excessively far from the reference orbit. The blue separatrix graph was obtained by tracking a particle very near the separatrix.

gives practical constraints on the harmonic number h determining the RF cavity revolution frequency ν_{RF}, due to engineering and lattice configuration considerations.

6.2.4 Spin Decoherence Minimization Using an RF Cavity and Sextupoles

In this subsection, we will describe the theory of first- and second-order spin decoherence minimization using an RF cavity and sextupoles.

6.2.4.1 Synchronous Acceleration Principle

A lattice using an RF cavity to suppress the longitudinal effects, as outlined above, satisfies the synchronous acceleration principle (which is also called the phase stability principle [113, pp. 266–270]) independently proposed in 1945 by Veksler [110] and McMillan [76]. The longitudinal

dynamics in such a system are described by the equations [100, p. 2579]

$$\frac{d\varphi}{dt} = -\omega_{\mathrm{RF}}\eta\,\delta,$$
$$\frac{d\delta}{dt} = \frac{qV_{\mathrm{RF}}\omega_{\mathrm{RF}}}{2\pi h\beta^2 E}\sin\varphi,$$

where φ is the phase offset relative to the synchronous (reference) phase $\varphi = 0$; $\delta = \Delta p/p_0$ is the relative momentum deviation from the reference momentum; η is the slip factor; q is the particle charge; V_{RF} is the voltage gap of the RF cavity; ω_{RF} is the angular frequency of the RF cavity; and h is the harmonic number.

6.2.4.2 Orbit Lengthening Due to the Betatron Motion and Momentum Deviation

Let T_{rev} be the revolution period of a particle in a system that utilizes the synchronous acceleration principle to suppress the longitudinal effects. Senichev *et al.* demonstrated in [100, p. 2579] that

$$\frac{\Delta T_{\mathrm{rev}}}{T_{\mathrm{rev}}} = \frac{\Delta(L/v)}{L/v} = \\ = \left(\frac{\Delta L}{L}\right)_\beta + \left(\alpha_0 - \frac{1}{\gamma^2}\right)\delta + \left(\alpha_1 - \frac{\alpha_0}{\gamma^2} + \frac{1}{\gamma^4}\right)\delta^2 + O\left(\delta^3\right),$$

where L is the particle orbit length, v is the particle velocity, $(\Delta L/L)_\beta$ is orbit lengthening due to the betatron motion, $\delta = \Delta p/p_0$ is the relative momentum deviation, and $\alpha = \alpha_0 + \alpha_1\delta + O\left(\delta^2\right)$ is the first-order Taylor expansion of the momentum compaction factor.

Thus, every particle in the system has its own equilibrium-level momentum shift [100, p. 2580]

$$\Delta\delta_{\mathrm{eq}} = \frac{\gamma_0^2}{\gamma_0^2\alpha_0 - 1}\left[\frac{\delta_{\mathrm{m}}^2}{2}\left(\alpha_1 - \frac{\alpha_0}{\gamma_0^2} + \frac{1}{\gamma_0^4}\right) + \left(\frac{\Delta L}{L}\right)_\beta\right],$$

where δ_{m} is the particle's amplitude of relative momentum deviation.

In a particle bunch, the maximal orbit lengthening due to the betatron motion can be expressed as [100, p. 2580]

$$\left(\frac{\Delta L}{L}\right)_\beta = \frac{\pi}{2L}\left[\varepsilon_x v_x + \varepsilon_y v_y\right],$$

where ε_x and ε_y are horizontal and vertical beam emittances, respectively; and v_x and v_y are horizontal and vertical tunes, respectively.

Therefore, the beam has the following equilibrium momentum spread due to momentum deviation and the betatron motion [100, p. 2581]:

$$\Delta\tilde{\delta}_{\text{eq}} = \frac{\gamma_0^2}{\gamma_0^2\alpha_0 - 1}\left[\frac{\tilde{\delta}_{\text{m}}^2}{2}\left(\alpha_1 - \frac{\alpha_0}{\gamma_0^2} + \frac{1}{\gamma_0^4}\right) + \frac{\pi}{2L}\left(\varepsilon_x \nu_x + \varepsilon_y \nu_y\right)\right],$$

where $\tilde{\delta}_{\text{m}}$ is the amplitude of relative momentum deviation of the entire particle beam.

6.2.4.3 Orbit Lengthening Correction Using Sextupoles

The equilibrium momentum spread $\Delta\delta_{\text{eq}}$ results in an equilibrium energy spread expressed as $\Delta\gamma_{\text{eq}}$. Since the spin tune is $\nu = \gamma G$, this results in a spin tune spread [100, p. 2581]

$$2\pi\left\langle\Delta\nu\right\rangle N = 2\pi G\left\langle\Delta\gamma_{\text{eq}}\right\rangle N$$

after N turns, which signifies spin decoherence growth.

To minimize the spin decoherence growth due to momentum deviation and the betatron motion, sextupoles or octupoles may be used.

A sextupole of length l_{sext} and strength

$$S_{\text{sext}} = \frac{q}{p}\frac{\partial^2 B_y}{\partial x^2},$$

where B is the magnetic field and χ_{m} is the magnetic rigidity, affects the second-order momentum compaction factor as [100, p. 2581]

$$\Delta\alpha_{1,\text{sext}} = -\frac{S_{\text{sext}} l_{\text{sext}} D_0^3}{L},$$

where D_0 is the momentum deviation–independent part of dispersion $D(s,\delta) = D_0(s) + D_1(s)\,\delta + O\left(\delta^2\right)$. At the same time, it directly affects the orbit lengthening as [100, p. 2581][97, p. 3019]

$$\left(\frac{\Delta L}{L}\right)_{\text{sext}} = \mp\frac{S_{\text{sext}} l_{\text{sext}} D_0 \beta_{x,y} \varepsilon_{x,y}}{L},$$

where β_x and β_y are the horizontal and vertical beta functions, respectively; and minus and plus refer to orbit lengthening due to horizontal and vertical plane effects, respectively.

To minimize the spin decoherence by compensating the orbit lengthening, we can use three or more families of sextupoles to solve the following system of equations [97, p. 3019]:

$$
\begin{aligned}
-\frac{\varepsilon_x}{2L}\sum_i S_i l_{si} D_{xi}\beta_{xi} &= \frac{\varepsilon_x}{2L}\pi\nu_x,\\
\frac{\varepsilon_y}{2L}\sum_i S_i l_{si} D_{xi}\beta_{yi} &= \frac{\varepsilon_y}{2L}\pi\nu_y,\\
-\delta^2\frac{1}{L}\sum_i S_i l_{si} D_{xi}^3 &= \delta^2\alpha_1,
\end{aligned}
$$

where l_{si} is the length and D_{xi} is the horizontal dispersion function of the i-th sextupole, respectively.

The sextupole strengths that solve this system of equations should be the smallest possible to minimize other effects of the sextupoles, which in turn may have to be corrected if they become too significant. To this end, sextupoles may be placed in the lattice at minima/maxima of D_x, β_x, and β_y [100, p. 2581].

6.2.4.4 Chromaticity Correction Using Sextupoles

To minimize spin decoherence, in addition to correcting the orbit lengthening, it is necessary to perform chromaticity correction. Chromaticity can be compensated by sextupole family strengths that solve the system of equations [46][97, p. 3019][64, pp. 160–170]

$$
\begin{aligned}
\sum_i S_i l_{si} D_{xi}\beta_{xi} &= \xi_x,\\
-\sum_i S_i l_{si} D_{xi}\beta_{yi} &= \xi_y,
\end{aligned}
$$

where horizontal and vertical chromaticities are defined as the derivatives

$$
\xi_{x,y} = \left.\frac{\partial \nu_{x,y}}{\partial\delta}\right|_{\delta=0}
$$

of the betatron tunes ν_x and ν_y by the relative momentum deviation δ, respectively.

We remark that chromaticity is also often defined as derivatives of the betatron tunes ν_x and ν_y by the relative offset of the particle's kinetic energy $\delta_K = (K - K_0)/K_0$. The conversion between chromaticities defined through momentum and through kinetic energy can be obtained using the

chain rule and the relativistic formula for the kinetic energy

$$K = \sqrt{p^2c^2 + m^2c^4} - mc^2$$

or the non-relativistic formula $K = p^2/2m$, as suitable. The two chromaticity definitions are essentially equivalent (modulo this conversion), including for the purposes of chromaticity compensation.

6.2.4.5 Calculation of Chromaticity

Working with beamline coordinate systems that use the relative offset of the particle's kinetic energy δ_K, as in *COSY INFINITY*, it is convenient to define the horizontal and vertical chromaticities as the derivatives of the betatron tunes ν_x and ν_y by δ_K, respectively.

To derive expressions for the chromaticities

$$\xi_{x,y} = \left.\frac{\partial \nu_{x,y}}{\partial \delta_K}\right|_{\delta_K=0},$$

consider a transfer map $\mathscr{M}$ that acts on coordinate vector z:

$$\mathscr{M}(z) = \sum_{i\geq 1} \sum_{i_1+\cdots+i_6=i} M^j_{i_1\cdots i_6} z_1^{i_1} \cdots z_6^{i_6},$$

where $M^j_{i_1\cdots i_6}$ are the transfer map's DA coefficients.

A particle with a sufficiently small energy offset $\delta_K = \delta$ and zero longitudinal offset $l = 0$ will have a closed orbit with a fixed point

$$\mathring{z} = (\mathring{z}_1, \mathring{z}_2, \mathring{z}_3, \mathring{z}_4, 0, \delta).$$

Let $\tilde{\mathscr{M}}$ be the re-expansion of the original transfer map $\mathscr{M}$ around the fixed point $\mathring{z}$ in coordinates $\tilde{z} = z - \mathring{z}$, and let $\tilde{M}^j_{i_1\cdots i_6}$ be its DA coefficients:

$$\begin{aligned}\tilde{\mathscr{M}}(\tilde{z}) &= \sum_{i\geq 1} \sum_{i_1+\cdots+i_6=i} \tilde{M}^j_{i_1\cdots i_6} \tilde{z}_1^{i_1} \cdots \tilde{z}_6^{i_6} = \\ &= \sum_{i\geq 1} \sum_{i_1+\cdots+i_6=i} M^j_{i_1\cdots i_6} (\tilde{z}_1 + \mathring{z}_1)^{i_1} \cdots (\tilde{z}_4 + \mathring{z}_4)^{i_4} \tilde{z}_5^{i_5} (\tilde{z}_6 + \delta)^{i_6} - \mathring{z}.\end{aligned}$$

For the reference particle, the horizontal and vertical betatron tunes are

$$\nu_{x,y} = \frac{1}{2\pi} \arccos\left[\frac{\mathrm{tr}\left(M_{x,y}\right)}{2}\right],$$

where M_x and M_y are the 2×2 x and y submatrices of the linear part M of the transfer map $\mathscr{M}$, respectively.

Let $\nu_{x,y}(\delta)$ be the betatron turns corresponding to the closed orbit for the energy offset δ. We have the Taylor expansions

$$\nu_{x,y}(\delta) = \nu_{x,y}(0) + \frac{\partial \nu_{x,y}}{\partial \delta}(0)\,\delta + O\left(\delta^2\right),$$

where

$$\xi_{x,y} = \frac{\partial \nu_{x,y}}{\partial \delta}(0)$$

are the chromaticities.

Expressing $\nu_{x,y}(\delta)$ in terms of the re-expanded transfer map,

$$\nu_{x,y} = \frac{1}{2\pi} \arccos\left[\frac{\mathrm{tr}\left(\tilde{M}_{x,y}\right)}{2}\right], \tag{6.3}$$

where the 2×2 x and y submatrices $\tilde{M}_{x,y}$ of the linear part $\tilde{M}$ of the transfer map $\tilde{\mathscr{M}}$ can be obtained from the equation $\mathscr{M}(z) - \mathring{z} = \tilde{\mathscr{M}}(\tilde{z})$.

Because $\mathring{z} = 0$ in case $\delta = 0$, we have for $\mathring{z}$ that

$$\mathring{z} = \frac{\partial \mathring{z}}{\partial \delta}\delta + O\left(\delta^2\right).$$

In a lattice satisfying the midplane symmetry, it is sufficient to consider the coordinates

$$(z_1, z_2, z_6) = (x, a, \delta_K),$$

as the coefficients $(y|\delta_K)$ and $(b|\delta_K)$ vanish.

To obtain $\partial \mathring{z}/\partial\delta$, we solve the equation

$$\begin{pmatrix} \lambda_1\delta \\ \lambda_2\delta \\ \delta \end{pmatrix} = M \begin{pmatrix} \lambda_1\delta \\ \lambda_2\delta \\ \delta \end{pmatrix},$$

where

$$M = \begin{pmatrix} M_1^1 & M_2^1 & M_6^1 \\ M_1^2 & M_2^2 & M_6^2 \\ 0 & 0 & 1 \end{pmatrix}$$

is the linear part of the transfer map $\mathscr{M}$ in (x, a, δ_K) coordinates.

This can be rewritten as

$$I_H \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ 1 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} M_1^1 & M_2^1 & M_6^1 \\ M_1^2 & M_2^2 & M_6^2 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ 1 \end{pmatrix},$$

where

$$I_H = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

This yields

$$\frac{\partial \mathring{z}}{\partial \delta} = (\lambda_1, \lambda_2, 1), \tag{6.4}$$

where

$$\begin{pmatrix} \lambda_1 \\ \lambda_2 \\ 1 \end{pmatrix} = (M - I_H)^{-1} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}.$$

We note that

$$\begin{aligned} \frac{\partial \left(\mathscr{M}(z) \right)}{\partial \delta} &= \frac{\partial \left(\tilde{\mathscr{M}}(\tilde{z}) \right)}{\partial \delta} + \frac{\partial \mathring{z}}{\partial \delta} = \\ &= \frac{\partial \tilde{\mathscr{M}}}{\partial \delta}(\delta, \tilde{z}) + \frac{\partial \tilde{\mathscr{M}}}{\partial \tilde{z}}(\delta, \tilde{z}) \frac{\partial \tilde{z}}{\partial \delta} + \frac{\partial \mathring{z}}{\partial \delta} = 0; \end{aligned}$$

thus, considering eq. 6.4, we have

$$\begin{aligned} \frac{\partial \tilde{\mathscr{M}}_X}{\partial \delta}(\tilde{z}) &= -\frac{\partial \tilde{\mathscr{M}}_X}{\partial \tilde{z}}(\tilde{z}) \frac{\partial \tilde{z}}{\partial \delta} - \frac{\partial \mathring{z}}{\partial \delta} = \\ &= \frac{\partial \tilde{\mathscr{M}}_X}{\partial \tilde{z}}(\tilde{z}) \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ 1 \end{pmatrix} - \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ 1 \end{pmatrix}, \end{aligned} \tag{6.5}$$

where

$$\frac{\partial \tilde{\mathscr{M}}_X}{\partial \tilde{z}}(\tilde{z}) = \left(\frac{\partial \left(\tilde{\mathscr{M}}_X\right)_i}{\partial \tilde{z}_j}(\tilde{z}) \right)_{1 \le i,j \le 3}$$

and $\tilde{\mathscr{M}}_X$ is the (x, a, δ_K) part of the transfer map $\tilde{\mathscr{M}}$.

Further, writing out the transfer map coefficients for the 2×2 x submatrices $\tilde{M}_x$ and M_x of $\tilde{M}$ and M, respectively, we obtain

$$\begin{aligned} \tilde{M}_x = M_x + & \begin{pmatrix} 2M^1_{11}\lambda_1 & 2M^1_{22}\lambda_2 \\ 2M^2_{11}\lambda_1 & 2M^2_{22}\lambda_2 \end{pmatrix} \delta + \\ + & \begin{pmatrix} M^1_{12}\lambda_2 & M^1_{21}\lambda_1 \\ M^2_{12}\lambda_2 & M^2_{21}\lambda_1 \end{pmatrix} \delta + \begin{pmatrix} M^1_{16} & M^1_{26} \\ M^2_{16} & M^2_{26} \end{pmatrix} \delta + O\left(\delta^2\right) \end{aligned} \tag{6.6}$$

in terms of the first and second order DA coefficients of the transfer map $\mathscr{M}$.

From eq. 6.3 the formula for the horizontal chromaticity is

$$\xi_x = \frac{1}{2\pi} \frac{\partial}{\partial \delta} \arccos\left[\frac{1}{2} \mathrm{tr}\left(\tilde{M}_x(\delta)\right) \right] \Bigg|_{\delta=0}.$$

Using eq. 6.6 to write out the transfer map coefficients, we have for the horizontal chromaticity

$$\begin{aligned} \xi_x = & \left[2\pi \left(M^1_2 M^2_1 - M^1_1 \left(M^2_2 - 1 \right) + M^2_2 - 1 \right) \sqrt{4 - \left(M^1_1 + M^2_2 \right)^2} \right]^{-1} \cdot \\ & \cdot \Bigg[M^1_6 \left(-M^2_1 \left(M^1_{12} + 2M^2_{22} \right) + 2 \left(M^2_2 - 1 \right) M^1_{11} + \left(M^2_2 - 1 \right) M^2_{12} \right) + \\ & + M^1_2 \left(M^2_1 \left(M^1_{16} + M^2_{26} \right) - M^2_6 \left(2M^1_{11} + M^2_{12} \right) \right) + \\ & + \left(M^1_1 - 1 \right) \left(M^2_6 \left(M^1_{12} + 2M^2_{22} \right) - \left(M^2_2 - 1 \right) \left(M^1_{16} + M^2_{26} \right) \right) \Bigg]. \end{aligned}$$

Using `K*PARA(1)` to define the reference particle of kinetic energy `K` in *COSY INFINITY*, essentially, solves the same problem, and gives the correct result for chromaticity as the first derivative of the betatron tune by the parameter `PARA(1)`. Calculating the chromaticity manually using formulas above, we obtained the same numerical results as in *COSY INFINITY*.

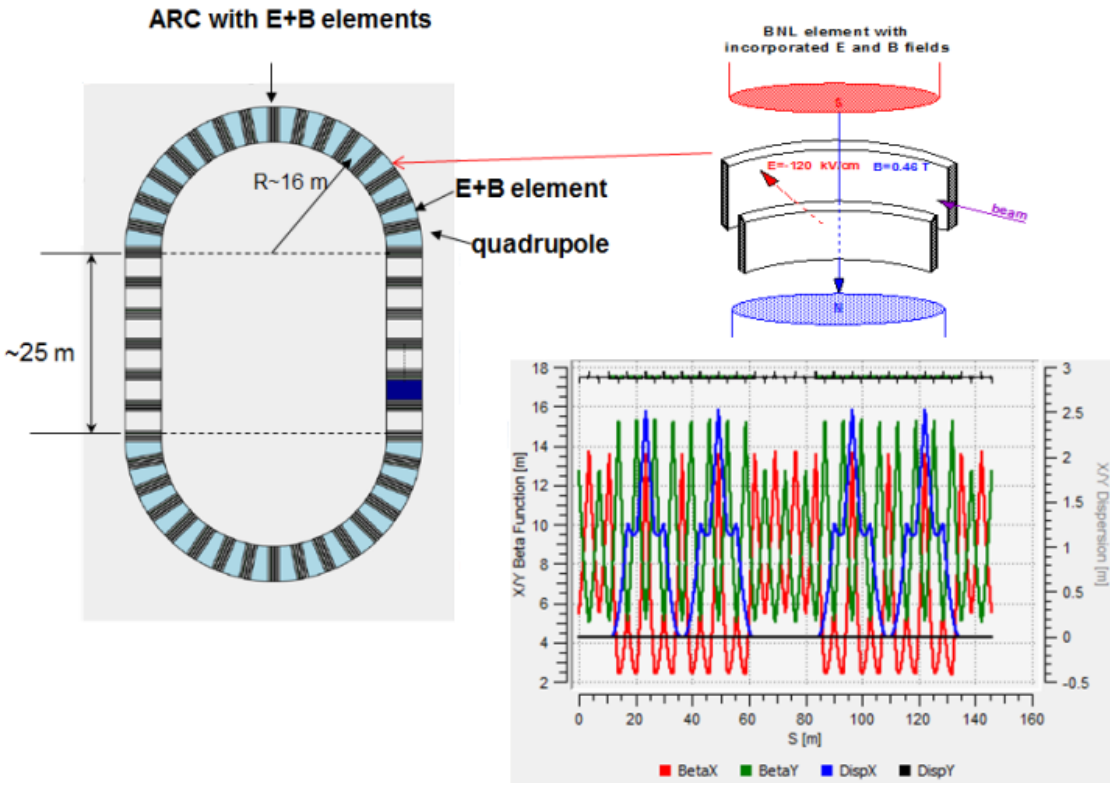


Figure 6.2: The *Senichev BNL* FS lattice layout, a plot of its beta and dispersion functions, and a diagram of the BNL *E+B* bending element used in its bending sections. (Image source: Yurij Senichev, Serge Andrianov, Andrei Ivanov, Stanislav Chekmenev, Martin Berz, & Eremey Valetov, Investigation of Lattice for Deuteron EDM Ring, In *Proc. of International Computational Accelerator Physics Conference (ICAP'15), Shanghai, China, 12–16 October 2015*, pages 17–19, JACoW, Geneva, Switzerland, 2016; use permitted under the Creative Commons Attribution License CC BY 3.0.)

6.3 Frozen Spin and Quasi-Frozen Spin Lattices

6.3.1 *Senichev BNL* Frozen Spin Lattice

The frozen spin (FS) lattice concept was introduced in Ch. 1. We implemented and studied an FS lattice for deuteron EDM with combined *E+B* bending elements, where the entire MDM angular velocity Ω_{MDM} term in the Thomas–BMT equation vanishes, which is achieved by a radial electrostatic field [2, p. 10]

$$E = \frac{BcG\beta\gamma^2}{1 - G\beta^2\gamma^2} \approx BcG\beta\gamma^2,$$

where B is the vertical magnetic field and G is the deuteron's anomalous MDM. We call such E+B bending elements "BNL elements", as they were first proposed at BNL, and we codenamed this lattice in our work as the *Senichev BNL* lattice.

The lattice, proposed by Senichev [96], is 145.85 m in length and uses deuterons at the kinetic energy of 270 MeV. It comprises two E+B bending sections interposed by two straight sections. The lattice is shown in Fig. 6.2.

An RF cavity is used to suppress first-order and, partially, second-order spin decoherence by mixing the particles relatively to their equilibrium energies, averaging out the $\triangle\gamma G$ for each particle. The RF cavity is at voltage $V = 100\,\mathrm{kV}$, in the realistic range of $\sim 0.1\,\mathrm{MV}$ to $\sim 0.5\,\mathrm{MV}$ [94]. For good energy averaging, the RF cavity frequency must be one to two orders higher than $\Delta\gamma G$ [94]. The RF frequency is $\nu_{\mathrm{RF}} = 5\nu_0$, where $\nu_0 = 1.00\,\mathrm{MHz}$ is the beam revolution frequency, providing synchronicity and good energy averaging. The remaining second-order component of spin decoherence, due to the average of $\triangle\gamma G$ being different for each particle, is suppressed by two families of sextupoles with optimized strengths.

6.3.2 Quasi-Frozen Spin Lattices

6.3.2.1 The QFS Concept

The quasi-frozen spin (QFS) lattice concept is based on the FS concept, but the requirement that spin needs to be aligned with momentum is relaxed: in QFS, spin is aligned with momentum *on average* during each turn. The QFS condition to maintain an average alignment of spin with momentum is $\theta_B + \theta_E = 0$ [98, pp. 213–214], where θ_B and θ_E are the polar rotation angles of spin relative to momentum in the magnetic field and electrostatic field, respectively.

According to the Thomas–BMT equation, the MDM angular frequency is

$$\Omega_{\mathrm{MDM}} = \frac{q}{m}\left[GB - \left(G - \frac{1}{\gamma^2 - 1}\right)\frac{E \times \beta}{c}\right]. \tag{6.7}$$

At the same time, the momentum precession of a particle in magnetic field B is

$$\omega_B = \frac{q}{m}\frac{B}{\gamma}, \tag{6.8}$$

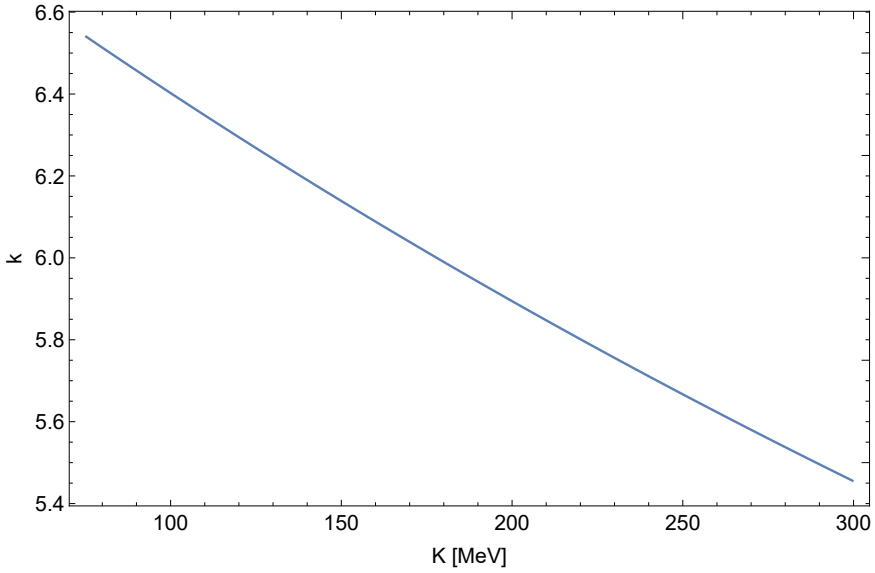


Figure 6.3: The ratio k of spin tune ν_s^E in the electrostatic field to spin tune ν_s^B in the magnetic field as a function of kinetic energy K, from 75 MeV to 300 MeV, for the deuteron.

and the momentum precession of a particle in electrostatic field E is

$$\omega_E = \frac{q}{m}\frac{E \times \beta}{c\beta^2\gamma}. \tag{6.9}$$

Scaling spin precessions due to magnetic and electric fields in eq. 6.7 by momentum precessions in eqns. 6.8 and 6.9, respectively, we obtain

$$\theta_B = \gamma G \Phi_B \tag{6.10}$$

and

$$\theta_E = \beta^2\gamma\left(\frac{1}{\gamma^2-1} - G\right)\Phi_E, \tag{6.11}$$

where Φ_B and Φ_E are the total amplitudes of momentum rotation due to magnetic and electric fields, respectively.

This yields the QFS condition in terms of momentum rotation as [98, pp. 213–214]

$$\gamma G \Phi_B + \beta^2\gamma\left(\frac{1}{\gamma^2-1} - G\right)\Phi_E = 0. \tag{6.12}$$

Moreover, we obtain from eqns. 6.11 and 6.10 that the ratio k of spin tune $\nu_s^E = \theta_E/\Phi_E$ in the electrostatic field to spin tune $\nu_s^B = \theta_B/\Phi_B$ in the magnetic field is

$$k = \frac{\nu_s^E}{\nu_s^B} = \\ = \frac{\beta^2}{G}\left(G - \frac{1}{\gamma^2 - 1}\right),$$

which decreases from $k = 6.5$ at kinetic energy of $K = 75\,\text{MeV}$ to $k = 5.5$ at kinetic energy of $K = 300\,\text{MeV}$ for the deuteron [98, p. 214], as Fig. 6.3 shows.

Considering that the total momentum rotation in one turn around the lattice is $\Phi_B + \Phi_E = 360°$, eq. 6.12 uniquely defines the total amplitudes of momentum rotation Φ_B and Φ_E at a kinetic energy K.

Because $\theta_E = -\theta_B$ and spin is aligned with momentum on average, the maximum deviation of spin relative to momentum is, at most, by angle $\theta/2$ in the horizontal plane, where $\theta = |\theta_E| = |\theta_B|$ is the total amplitude of spin rotation relative to momentum in the magnetic or electrostatic field. With radial electrostatic field, the torque due to EDM rotating spin out of the horizontal plane is proportional to the projection of spin on the momentum vector. Thus, the EDM signal growth in the QFS concept is, at most, smaller than in the FS concept proportionally to the factor [98, pp. 213–215][96, p. 18]

$$J_0(\theta) \simeq 1 - \frac{\theta^2}{4},$$

where spin is aligned with momentum. Considering that the deuteron's anomalous MDM is $G = -0.143$, we expect the EDM signal to be reduced by, at most, only two percent [98, p. 215] in a QFS lattice compared to an FS lattice.

6.3.2.2 *Senichev 6.3* QFS Lattice

First, we considered a QFS lattice proposed by Senichev [98] and codenamed *Senichev 6.3* QFS lattice. The lattice uses separate electrostatic and magnetic bending elements with curvatures in opposite directions to satisfy the QFS condition, and it has a characteristic "hourglass" shape. A

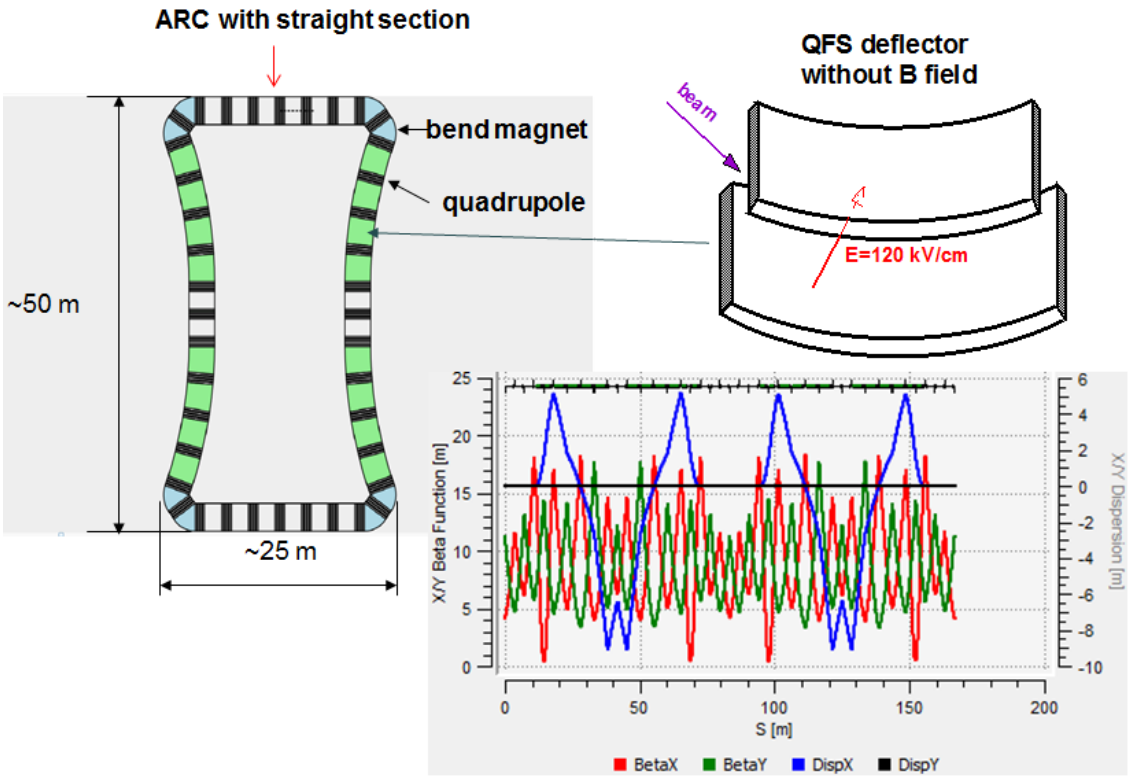


Figure 6.4: The *Senichev 6.3* QFS lattice layout, a plot of its beta and dispersion functions, and a diagram of the curved electrostatic deflector used in this lattice. (Image source: Yurij Senichev, Serge Andrianov, Andrei Ivanov, Stanislav Chekmenev, Martin Berz, & Eremey Valetov, Investigation of Lattice for Deuteron EDM Ring, In *Proc. of International Computational Accelerator Physics Conference (ICAP'15), Shanghai, China, 12–16 October 2015*, pages 17–19, JACoW, Geneva, Switzerland, 2016; use permitted under the Creative Commons Attribution License CC BY 3.0.)

variation of this lattice can be implemented with relatively minor changes to a number of existing lattices, including the Cooler Synchrotron COSY at Forschungszentrum Jülich.

The lattice is 166.67 m in length and uses deuterons at the kinetic energy of 270 MeV. It comprises four magnetic bending sections, four electrostatic bending sections, and four straight sections. The lattice is shown in Fig. 6.4.

An RF cavity is used to suppress first-order and, partially, second-order spin decoherence by mixing the particles relatively to their equilibrium energies, averaging out the $\triangle\gamma G$ for each particle. The RF cavity is at voltage $V = 100\,\text{kV}$ or $V = 200\,\text{kV}$, which are two cases in the realistic range of $\sim 0.1\,\text{MV}$ to $\sim 0.5\,\text{MV}$ [94]. Good energy averaging by an RF cavity is achieved if the resulting longitudinal tune is one to two orders higher than the spin tune spread $\Delta\gamma G$ [94]. The RF frequency

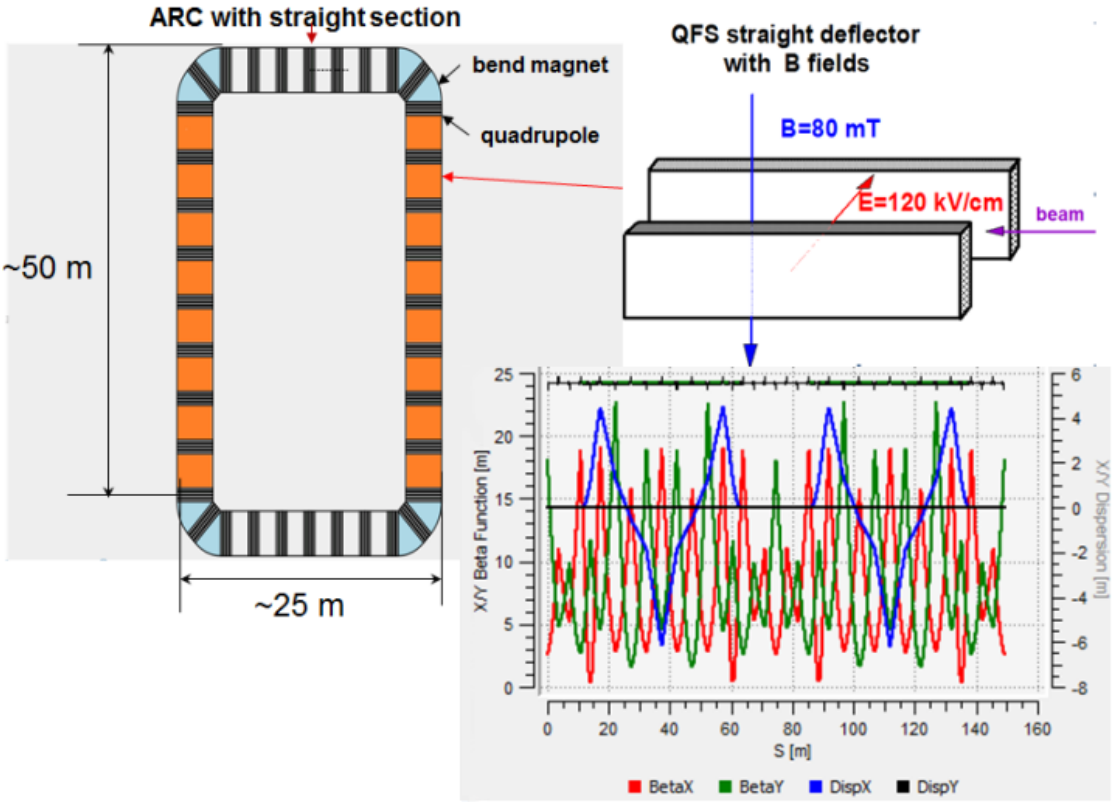


Figure 6.5: The *Senichev E+B* QFS lattice layout, a plot of its beta and dispersion functions, and a diagram of the straight *E+B* Wien filter element used in this lattice. (Image source: Yurij Senichev, Serge Andrianov, Andrei Ivanov, Stanislav Chekmenev, Martin Berz, & Eremey Valetov, Investigation of Lattice for Deuteron EDM Ring, In *Proc. of International Computational Accelerator Physics Conference (ICAP'15), Shanghai, China, 12–16 October 2015*, pages 17–19, JACoW, Geneva, Switzerland, 2016; use permitted under the Creative Commons Attribution License CC BY 3.0.)

is, for synchronicity and good energy averaging, $\nu_{\mathrm{RF}} = 5\nu_0$ or $\nu_{RF} = 6\nu_0$, where $\nu_0 = 0.87\,\mathrm{MHz}$ is the beam revolution frequency. The remaining second-order component of spin decoherence, due to the average of $\triangle\gamma G$ being different for each particle, is suppressed by six families of sextupoles with optimized strengths.

6.3.2.3 *Senichev E+B* QFS Lattice

Next, we considered a QFS lattice by Senichev [96] codenamed *Senichev E+B* QFS lattice. This lattice uses straight *E+B* static Wien filter elements instead of curved electrostatic deflectors to (1) remove the nonlinear electrostatic field components due to curvature and (2) to simplify lattice from

the engineering perspective.

The lattice is 149.21 m in length and uses deuterons at the kinetic energy of 270 MeV. It comprises four magnetic bending sections, two *E*+*B* combined field sections, and two straight sections. The lattice is shown in Fig. 6.5.

An RF cavity is used to suppress first-order and, partially, second-order spin decoherence by mixing the particles relatively to their equilibrium energies, averaging out the $\triangle\gamma G$ for each particle. The RF cavity is at voltage $V = 100\,\mathrm{kV}$ or $V = 200\,\mathrm{kV}$, two cases in the realistic range of $\sim 0.1\,\mathrm{MV}$ to $\sim 0.5\,\mathrm{MV}$ [94]. The RF frequency is, for synchronicity and good energy averaging, $\nu_{\mathrm{RF}} = 5\nu_0$ or $\nu_{RF} = 6\nu_0$, where $\nu_0 = 0.98\,\mathrm{MHz}$ is the beam revolution frequency. The remaining second-order component of spin decoherence, due to the average of $\triangle\gamma G$ being different for each particle, is suppressed by four families of sextupoles with optimized strengths.

6.3.3 Sextupole Family Nomenclature

In the Senichev FS and QFS lattices, the sextupole families are named SFP*n*, SDP*n*, SFN*n*, and SDN*n*. The first letter S means sextupole; the second letter is either F or P, denoting focusing or defocusing sextupoles, respectively; and the third letter is P or N, denoting positive or negative dispersion, respectively. The optional number *n* distinguishes between different sextupole families of the same type.

6.4 Spin Decoherence Study

We studied spin decoherence in the three FS and QFS lattices presented above, with optimization of the spin decoherence by sextupole strengths and as a function of the number of tracking turns *N*. In this study, we were concerned with the decoherence of polar angles of the spin vectors, which is key to the feasibility of EDM signal measurement using these lattices and the FS and QFS methods.

6.4.1 Computational Software

In the context of this chapter, we used *COSY INFINITY* [68] for various spin tracking calculations, including:

1. the manual and automatic spin decoherence optimization by sextupole family strengths;
2. the investigation of spin decoherence growth as a function of the number of turns; and
3. the study of the effects of systematic errors on spin decoherence.

We used *Wolfram Mathematica* notebooks for:

1. the automated preparation of *COSY INFINITY* input files from templates using the specified parameters, patterns, and markers in the template files; and
2. the storage, processing, quality assurance, and report generation using data from the *COSY INFINITY* output files. All optimization and tracking plots in the remainder of this chapter were generated using the report-generating *Mathematica* notebook.

6.4.2 Fringe Fields of Electrostatic Deflectors

For the fringe fields of the electrostatic deflectors (electrostatic dipoles) in the *Senichev* FS and QFS lattices, we used the Enge function coefficients that we calculated in sec. 2.3 for a semi-infinite electrostatic deflector with rounded edges and a realistic $D/20$ plate thickness, where D is the full aperture of the electrostatic deflector. These Enge function coefficients are listed in Table 6.1.

6.4.3 Manual Optimization of Sextupole Strengths

As the first step in the optimization of the sextupole family strengths, we performed their manual optimization. We tracked the three lattices for 2.5×10^4 turns with equidistant grid-point sets of initial particle offsets in x–a, y–b, and l–δ_K planes, with the RF cavity off an on, and with the various RF cavity frequencies and voltages as listed for each of the lattices.

Coefficient	Value
h_0	1.0614024399605924
h_1	1.6135741290714967
h_2	−0.9401447081042862
h_3	0.4781500036872176
h_4	−0.14379986967718494
h_5	0.017831089071215347

Table 6.1: Enge function coefficients of a semi-infinite capacitor with rounded edges of $D/20$ plate thickness. Here, we used these Enge function coefficients to model the fringe fields of the electrostatic deflectors in the *Senichev* FS and QFS lattices.

As the spin decoherence objective function, we used the maximum spin vector's polar angle θ deviation from that of the reference particle:

$$\texttt{OBJ} = \max_i |\theta_i - \theta_0| ,$$

where θ_i is the spin vector's polar angle of the i-th particle in the particle bunch and index $i = 0$ denotes the reference particle.

In each manual-optimization tracking run, we obtained a set of objective function values `OBJ` for a grid-point set of sextupole strengths assigned to a sextupole family. We performed manual optimization with two iterations per sextupole family: the first iteration used a grid-point set of the sextupole family strength with a large range to observe the general pattern of spin decoherence as a function of the sextupole family strength, and the second iteration decreased the range of the grid-point set to a neighborhood of the apparent spin decoherence minimum.

For the example of the *Senichev 6.3* QFS lattice, with the RF cavity on and initial particle offsets in the x–a plane, Fig. 6.6 shows a plot of objective function values `OBJ` (ordinate) as a function of sextupole strengths (abscissa). Each plot marker color corresponds to a sextupole family – SFP1, SFP2, SDP1, SDP2, SFN1, SDN1, or SDN2 – and the wide-range or a narrow-range set of sextupole strengths, as denoted in the plot legend.

Compared with the x–a plane, in case of initial particle offsets in the y–b plane, the objective function curves are more parabolic and lower in magnitude.

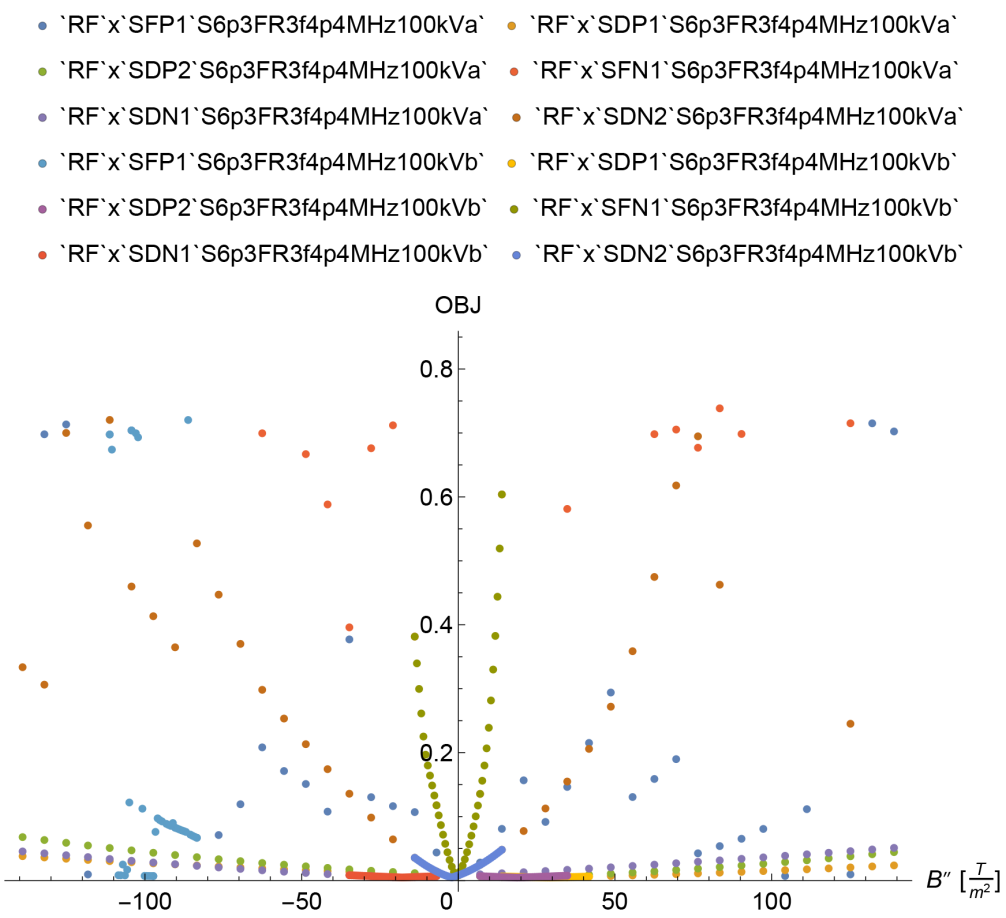


Figure 6.6: Objective function `OBJ` as a function of sextupole family strengths in the *Senichev 6.3* QFS lattice. Each plot marker color corresponds to a sextupole family – SFP1, SFP2, SDP1, SDP2, SFN1, SDN1, or SDN2 – and the wide-range ('a') or a narrow-range ('b') set of sextupole strengths, as specified within the plot legend codes. Tracking was performed for 2.5×10^4 turns with the RF cavity on, particles launched with initial x offsets up to $x_\mathrm{i} = \pm 5 \times 10^{-3}$ m, fringe field mode `FR3`, RF cavity frequency $\nu = 4.4\,\mathrm{MHz}$, and RF cavity voltage $V = 100\,\mathrm{kV}$.

6.4.4 Automatic Optimization of Sextupole Strengths

Having obtained the approximate global minima for the spin decoherence objective function `OBJ` as a function of each of the sextupole family strengths, we further optimized the spin decoherence using *COSY INFINITY*'s built-in Levenberg–Marquardt optimizer.

For the example of the *Senichev 6.3* QFS lattice, with the RF cavity on and initial particle offsets in the x–a plane, Fig. 6.7 shows the spin vector's polar angle θ as a function of the initial x offset of a tracked particle after 2.5×10^4 turns. Each plot marker color corresponds to a sextupole family – SFP1, SFP2, SDP1, SDP2, SFN1, SDN1, or SDN2 – as denoted in the plot legend. We note that,

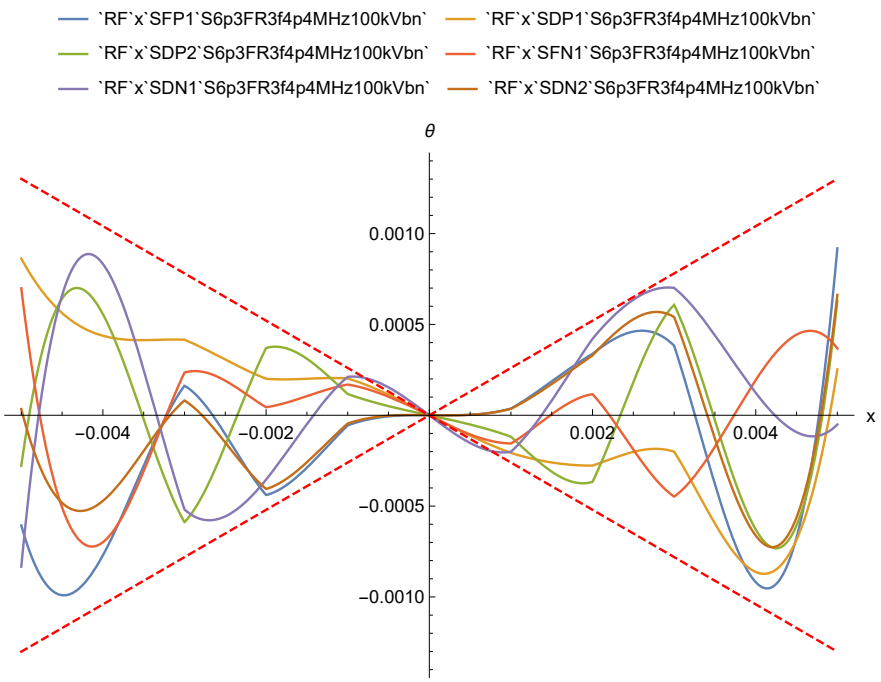


Figure 6.7: Spin vector's polar angle θ after 2.5×10^4 turns as a function of the initial x offset of the tracked particle, after manual and automatic spin decoherence optimization by a sextupole family strength in the *Senichev 6.3* QFS lattice. Each solid curve color corresponds to a sextupole family – SFP1, SDP1, SDP2, SFN1, SDN1, or SDN2 – as specified within the plot legend codes. Optimization and tracking were performed with the RF cavity on, particles launched with initial x offsets up to $x_\mathrm{i} = 5 \times 10^{-3}$ m, fringe field mode `FR3`, RF cavity frequency $\nu = 4.4$ MHz, and RF cavity voltage $V = 100$ kV. The curves appear to be bounded by two symmetric lines (dashed red).

with the RF cavity on, the polar angle θ vs. initial particle offset curves often appear to be bounded by two symmetric slanted lines.

For implementation feasibility, it is necessary for the optimized sextupole strengths to have sufficiently large neighborhoods, within which the objective function `OBJ` does not increase excessively. Fig. 6.8 shows, on a log–log scale, the objective function `OBJ` versus deviation from the optimal sextupole strengths. Considering that physical sextupoles strengths can be realistically set with the accuracy of $\sim 0.1\%$, the thickness of the sextupole strength optimums is quite acceptable.

6.4.5 Spin Decoherence Analysis

At 2.5×10^4 turns, the spin decoherence in the FS and QFS lattices was sufficiently large for $x = 5 \times 10^{-3}$ m or $\delta_K = 5 \times 10^{-3}$ beam apertures to require the study of its longer-term

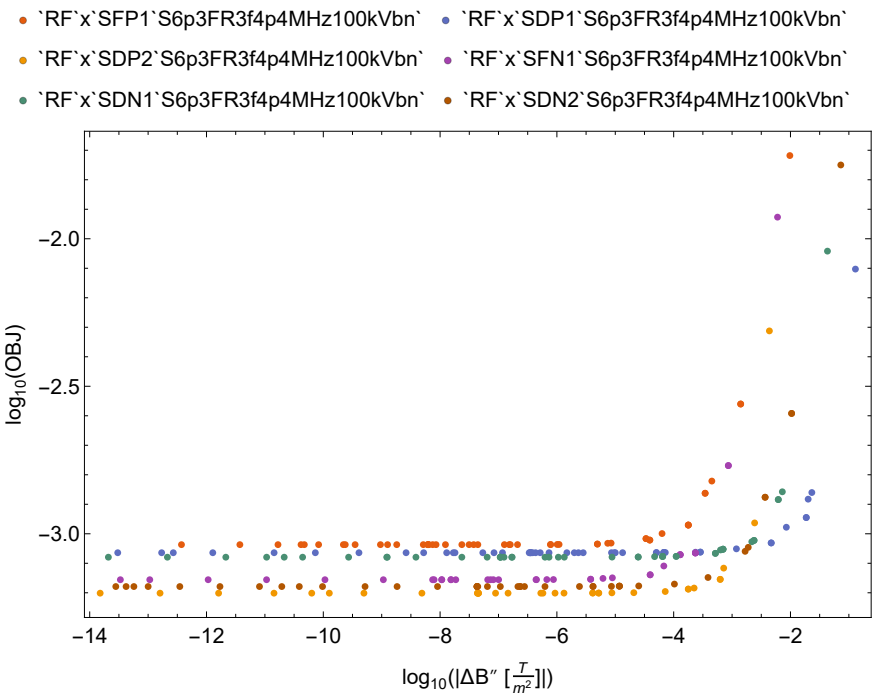


Figure 6.8: Objective function `OBJ` as a function of sextupole family strengths in the *Senichev 6.3* QFS lattice, in the neighborhoods of the respective optimal sextupole family strengths obtained in the automatic optimization. The absolute value $\left|\Delta B''\right| = 0$ of sextupole family strength offset corresponds to its optimal value. Each plot marker color corresponds to a sextupole family – SFP1, SDP1, SDP2, SFN1, SDN1, or SDN2 – as specified within the plot legend codes. Optimization and tracking were performed with the RF cavity on, particles launched with initial x offsets up to $x_\mathrm{i} = \pm 5 \times 10^{-3}$ m, fringe field mode `FR3`, RF cavity frequency $\nu = 4.4$ MHz, and RF cavity voltage $V = 100$ kV.

behavior [109].

With optimized sextupole strengths and the RF cavity on, we tracked the FS and QFS lattices for 4.2×10^5 and, in some cases, for 10^6 turns. Normally, we tracked the lattices using *COSY INFINITY*'s most accurate fringe field calculation mode `FR3`. For comparison, we additionally performed tracking with the RF cavity off, as well as without fringe field calculations (fringe field mode `FR0`). Spin decoherence objective function `OBJ` as a function of the number of tracked turns N is shown in Figs. 6.9, 6.10, and 6.11 for the *Senichev BNL*, *6.3*, and *E+B* lattices, respectively.

We found that spin decoherence suppression using an RF cavity and sextupole strengths often results in the spin decoherence remaining in approximately the same range for at least 4.2×10^5 turns. Spin decoherence effects were similar in the three FS and QFS lattices. The calculation results

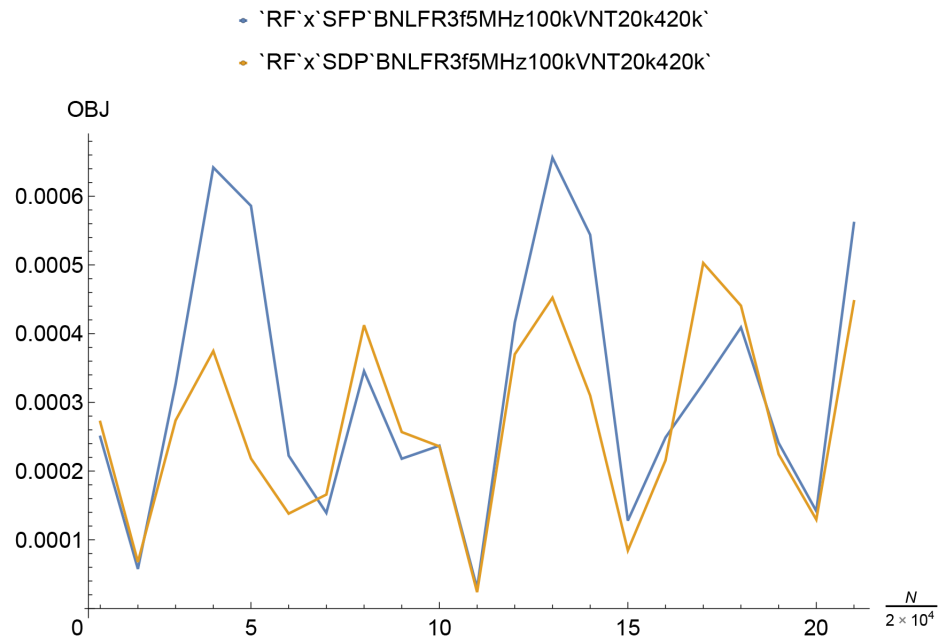


Figure 6.9: Horizontal spin decoherence objective function `OBJ` as a function of the number N of tracked turns in the *Senichev BNL* FS lattice, up to 4.2×10^5 turns. The spin decoherence remains approximately in the same range below 7×10^{-4} during the tracking. Each curve color corresponds to spin decoherence optimization by a sextupole family – SFP or SDP – as denoted in the plot legend. Optimization and tracking were performed with the RF cavity on, particles launched with initial x offsets up to $x_\mathrm{i} = \pm 5 \times 10^{-3}$ m, fringe field mode `FR3`, RF cavity frequency $\nu = 3.5$ MHz, and RF cavity voltage $V = 100$ kV.

indicate that spin decoherence suppression using an RF cavity and sextupoles may be sufficient to optimize spin decoherence to less than 1 rad in one billion turns, achieving the spin coherence time (SCT) of 10^3 s required for the EDM signal measurement [109].

6.5 Systematic Errors Study

Systematic errors due to imperfections in the physical lattice, including particle optical element misalignments [18, pp. 448–451] and geometric errors [60, ch. 13, pp. 73–76], are causative to a fake EDM signal. Considering the Thomas–BMT equation, rotational magnet misalignments are particularly problematic in this respect. We note that (1) the electrostatic deflector field strengths are interdependent with the strengths of magnetic dipoles via the lattice structure and the QFS or FS condition, as applicable, and that (2) the magnetic field direction needs to be accurately reversed for reverse lattice traversal, whereas the electrostatic field remains unchanged.

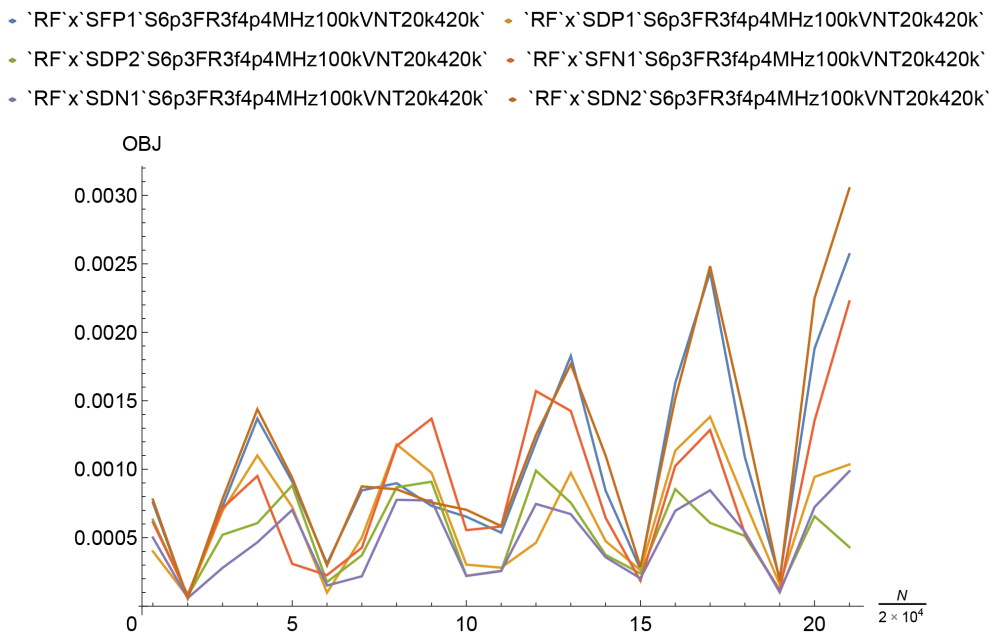


Figure 6.10: Horizontal decoherence objective function `OBJ` as a function of the number N of tracked turns in the *Senichev 6.3* QFS lattice, up to 4.2×10^5 turns. For three of the sextupole families, the spin decoherence remains approximately in the same range below 2×10^{-3} during the tracking. Each curve color corresponds to spin decoherence optimization by a sextupole family – SFP1, SDP1, SDP2, SFN1, SDN1, or SDN2 – as denoted in the plot legend. Optimization and tracking were performed with the RF cavity on, particles launched with initial x offsets up to $x_\mathrm{i} = \pm 5 \times 10^{-3}$ m, fringe field mode `FR3`, RF cavity frequency $\nu = 4.4$ MHz, and RF cavity voltage $V = 100$ kV.

To extract the EDM signal in the presence of rotational magnet misalignments, Senichev proposed a modification of the QFS and FS methods, where polarized particle bunches are tracked in both clockwise (CW) and counterclockwise (CCW) directions [99].

Horizontal magnetic error field components B_x and B_z rotate spin in the vertical plane, causing a fake EDM signal. On the other hand, a vertical magnetic error field component B_y results in rotation in the horizontal plane, which has an impact on the satisfaction of the QFS or FS condition and on spin decoherence but does not result in a fake EDM signal.

We studied spin dynamics in the QFS and FS lattices in the presence of rotational magnet misalignments causing horizontal magnetic error field components B_x and B_z. In particular, we studied spin decoherence and precession frequencies in the vertical plane.

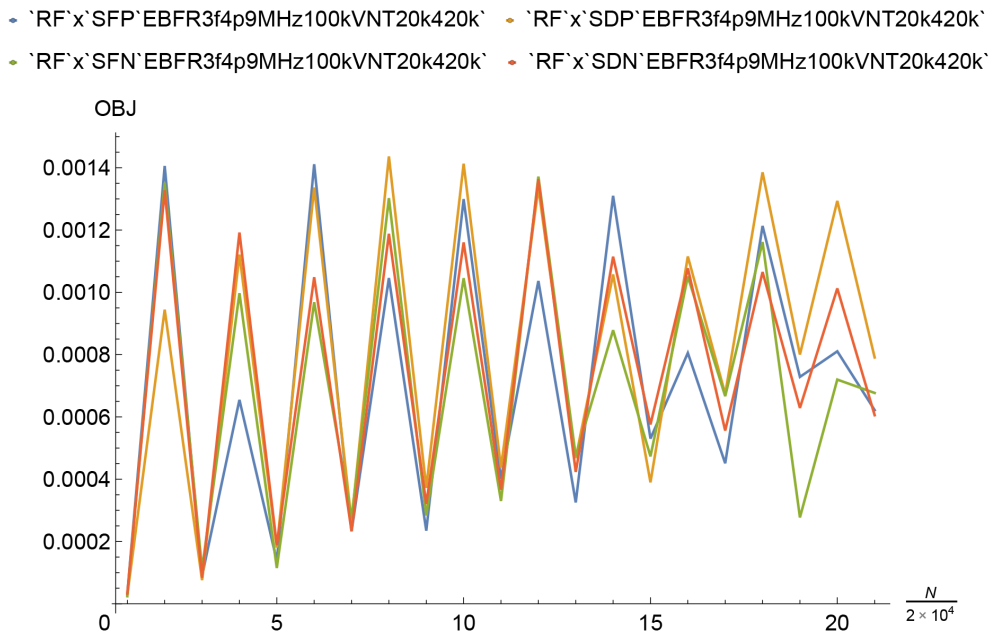


Figure 6.11: Horizontal decoherence objective function `OBJ` as a function of the number N of tracked turns in the *Senichev E+B* QFS lattice, up to 4.2×10^5 turns. The spin decoherence remains approximately in the same range below 1.5×10^{-3} during the tracking. Each curve color corresponds to spin decoherence optimization by a sextupole family – SFP, SDP, SFN, or SDN – as denoted in the plot legend. Optimization and tracking were performed with the RF cavity on, particles launched with initial x offsets up to $x_\mathrm{i} = \pm 5 \times 10^{-3}$ m, fringe field mode `FR3`, RF cavity frequency $\nu = 4.9$ MHz, and RF cavity voltage $V = 100$ kV.

6.5.1 EDM Signal Extraction by Tracking in Opposite Directions

Considering the Lorentz force, the reversal of an orbital transfer map reverses the direction of the magnetic field but does not affect the electrostatic field. At the same time, traversal of a lattice in the reverse direction reverses the direction of the reference particle's momentum. As a result, the MDM angular velocity of the reference particle,

$$\Omega_\mathrm{MDM} = \frac{q}{m}\left[GB - \left(G - \frac{1}{\gamma^2 - 1}\right)\frac{E \times \beta}{c}\right],$$

in the Thomas–BMT equation reverses direction if the lattice is traversed in the reverse direction. On the other hand, the EDM angular velocity of the reference particle,

$$\Omega_\mathrm{EDM} = \frac{q}{m}\frac{\eta}{2}\left[\frac{E}{c} + \beta \times B\right],$$

remains the same due to reverse traversal of the lattice. This can be seen by substituting $-E$ for E and $-\beta$ for β in the Ω_{MDM} and Ω_{EDM} equations.

Therefore, with a sufficiently small transversal and longitudinal beam emittance, it is possible to extract the EDM signal by tracking the lattice in CW and CCW directions and canceling out the fake EDM signal due to magnet misalignments.

6.5.2 Reverse Spin Transfer Map

To perform spin tracking in a lattice traversed in the CCW direction, it is necessary to compute the reverse orbital and spin transfer maps. *COSY INFINITY*'s built-in procedure `MR` computes the reverse orbital transfer map for a given orbital transfer map. Additionally, if each particle optical element in a lattice is symmetric relative to its central cross section, the reverse transfer map can be obtained by merely reversing the order of the particle optical elements in the lattice model.

To complement the procedure `MR`, we developed a procedure called `SMR` that calculates the reverse spin transfer map. The procedure `SMR` is as follows.

```
PROCEDURE SMR NAP LAP ; {REVERSES SPIN MAP NAP TO LAP}
    VARIABLE COD 1 NV ; VARIABLE NUM 1 ;
    VARIABLE I 1 ; VARIABLE J 1 ;
    VARIABLE MM NM1 3 3 ;
    VARIABLE T1 NM1 1 ; VARIABLE T2 NM1 1 ;
    VARIABLE FLG 1 ;
    NUM := MIN(TWOND,4) ;
    MATINV NAP MM ;
    MM(1,3) := -MM(1,3) ; MM(2,3) := -MM(2,3) ;
    MM(3,1) := -MM(3,1) ; MM(3,2) := -MM(3,2) ;
    LOOP I 2 NUM 2 ; COD(I-1) := DD(I-1) ;
        COD(I) := -DD(I) ; ENDLOOP ;
    LOOP I NUM+1 NV ; COD(I) := DD(I) ; ENDLOOP ;
    IF ND>2 ; COD(5) := -DD(5) ; ENDIF ;
```

```
    LOOP I 1 3 ; LOOP J 1 3 ;
        T1(1) := MM(I,J) ;
        POLVAL 1 T1 1 COD NV T2 1 ;
        LAP(I,J) := T2(1) ;
    ENDLOOP ; ENDLOOP ;
ENDPROCEDURE ;
```

Consider a spin transfer map $\mathscr{S} : (X_i, Z_i) \to X_{i+1}$, where X_i is the spin vector space and Z_i is the orbital phase space after the i-th turn around the lattice in the forward direction. For each turn i, X_i is the 3D sphere S^3 and $Z_i = \mathbb{R}^{2v}$, where v is the phase space dimension. To take into account the nonlinear terms, the elements of the 3×3 matrix $\mathscr{S}$ can be specified as DA values, which we assert as $\mathscr{S} \in \mathrm{SO}_3({}_nD_v)$. In contrast, in the linear case or approximation, $\mathscr{S}$ is an element of $\mathrm{SO}_3(\mathbb{R})$. The inverse spin transfer map $\mathscr{S}^{-1} : \left(X_{i+1}, \mathbb{R}^{2v}\right) \to X_i$ is the inverse matrix of the matrix $\mathscr{S}$.

Time reversal changes signs of momentum coordinates a and b, the longitudinal offset l [11, p. 147], and the longitudinal component s_z of the spin vector.

The reverse spin transfer map $\mathscr{S}^{\mathrm{R}}$ is obtained by applying the respective transformation to the inverse spin transfer map:

$$\mathscr{S}^{\mathrm{R}} = \hat{R}_{\mathrm{S}} \cdot (\mathscr{S}^{-1} \circ \hat{R}) \cdot \hat{R}_{\mathrm{S}}, \tag{6.13}$$

where

$$\hat{R} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \end{pmatrix}$$

acts on the phase space coordinates $(x, y, \delta_K, a, b, l)$ and

$$\hat{R}_{\mathrm{S}} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix}$$

acts on the spin vector coordinates $\left(s_x, s_y, s_z\right)$. To obtain eq. 6.13, we made use of the property $\hat{R}_{\mathrm{S}}^{-1} = \hat{R}_{\mathrm{S}}$.

6.5.3 Error Field Implementation

Considering the Thomas–BMT equation, a small perturbation of the magnetic field acts with a first-order perturbation correction as a small proportional rotation on the spin vector. In the systematic errors study, we implemented rotational magnet misalignments as augmentation of the magnetic dipoles or combined *E+B* elements, as appropriate, with small, normally distributed rotational spin kicks.

For calculation of the spin kicks, we noted that magnetic error field components B_x and B_z in a homogeneous magnetic dipole result in the following change to the MDM angular velocity:

$$\triangle\Omega_{\mathrm{MDM}} = \frac{qG}{m}\left(B_x, 0, B_z\right).$$

The total spin kick for a homogeneous magnetic dipole of angle α and radius R is then

$$\begin{aligned} t_0 \triangle\Omega_{\mathrm{MDM}} &= \frac{\alpha R}{v_0} \triangle\Omega_{\mathrm{MDM}} = \\ &= \frac{\alpha G q R}{m v_0}\left(B_x, 0, B_z\right), \end{aligned}$$

where t_0 is the time of flight of the reference particle with velocity v_0 through the magnetic dipole.

We implemented the spin kicks in *COSY INFINITY* as elements `RSX`, `RSY`, and `RSZ`. The spin kick elements rotate the spin vector by a specified angle `ANG` around the x, y, and z axes, respectively. As a representative example, the `RSX` spin kick element is as follows.

```
PROCEDURE RSX ANG ;
    VARIABLE I 1 ; VARIABLE J 1 ;
    {Set rotation matrix}
    UMS ;
    LOOP I 1 3 ; LOOP J 1 3 ; SSCR(I,J) := 0*DD(1) ;
    ENDLOOP ; ENDLOOP ;
    SSCR(1,1) := 1 + 0*DD(1) ;
    SSCR(2,2) := COS(ANG) + 0*DD(1) ;
    SSCR(3,2) := -SIN(ANG) + 0*DD(1) ;
    SSCR(2,3) := SIN(ANG) + 0*DD(1) ;
    SSCR(3,3) := COS(ANG) + 0*DD(1) ;
    LOCSET 0 0 0 0 0 0 ; CE := 'RSX' ; DR := 0 ;
    UPDATE 1 1 1 ; {Apply element}
ENDPROCEDURE ;
```

Our *Mathematica* notebook for automated preparation of *COSY INFINITY* input files from templates based on specified settings (1) calculated the spin kick parameters and (2) inserted the spin kick elements into the generated *COSY INFINITY* code.

6.5.4 B_x Magnetic Field Component Due to Systematic Errors

Consider the spin dynamics of the reference particle in an FS or a QFS lattice. Let the effective EDM angular velocity be Ω_{EDM}, aligned with the x axis and rotating the spin vector out of the midplane. Traversing the lattice in the CW direction, a nonzero B_x magnetic field component arising from systematic errors would add an effective MDM angular velocity Ω_{B_x}, for the total effective angular velocity in the vertical plane [99, p. 3395–3396]

$$\Omega_x^{\mathrm{CW}} = \Omega_{B_x}{}^{\mathrm{CW}} + \Omega_{\mathrm{EDM}}. \tag{6.14}$$

Traversal of the lattice in the CCW direction would change the sign of the MDM angular

velocity, leaving the angular velocity Ω_{EDM} due to EDM unchanged:

$$\begin{aligned}\Omega_x^{\mathrm{CCW}} &= {\Omega_{B_x}}^{\mathrm{CCW}} + \Omega_{\mathrm{EDM}} \approx \\ &\approx -{\Omega_{B_x}}^{\mathrm{CW}} + \Omega_{\mathrm{EDM}}.\end{aligned} \tag{6.15}$$

In the horizontal plane, for the particle bunch, the average spin angular velocity can be expressed as $\Omega_y = 0 + \langle \delta\Omega_{\mathrm{decoh}} \rangle$, where $\langle \delta\Omega_{\mathrm{decoh}} \rangle$ accounts for spin decoherence in the horizontal plane.

Considering eqns. 6.14 and 6.15, with B_x magnetic field components due to systematic errors, angular velocity due to EDM can be extracted as [99, p. 3395–3396]

$$\Omega_{\mathrm{EDM}} = \frac{\Omega_x^{\mathrm{CW}} + \Omega_x^{\mathrm{CCW}}}{2} - \frac{{\Omega_{B_x}}^{\mathrm{CCW}} + {\Omega_{B_x}}^{\mathrm{CW}}}{2}. \tag{6.16}$$

Accordingly, to apply this EDM measurement method, it is necessary (1) to minimize the MDM angular velocity decoherence $\sigma\left(\Omega_{B_x}\right)$ in the vertical plane using the same method as for spin decoherence minimization in the horizontal plane, that is, using the RF cavity and by optimizing the sextupole family strengths; (2) to minimize $\left|\Omega_{B_x}^{\mathrm{CW}} + \Omega_{B_x}^{\mathrm{CCW}}\right|$ by calibrating the magnetic fields, which can be achieved by matching the spin precession frequencies in the horizontal plane [99, p. 3395–3396]; and (3) to minimize the transversal and longitudinal emittance.

6.5.5 B_z Magnetic Field Component Due to Systematic Errors

Now consider the B_z magnetic field component in a homogeneous magnetic dipole due to a rotational misalignment. We denote the z component of the MDM angular velocity due to the B_z magnetic error field component as $\Omega_z = \Omega_{B_z}$, whereas the x and y components of the MDM angular velocity are $\Omega_x = 0$ and $\Omega_y = 0 + \delta\Omega_{\mathrm{decoh}}$, where $\delta\Omega_{\mathrm{decoh}}$ accounts for spin decoherence in the horizontal plane [99, p. 3396]. Hence, the MDM angular velocity is

$$\Omega_{\mathrm{MDM}} = \left(0, \delta\Omega_{\mathrm{decoh}}, \Omega_{B_z}\right).$$

Let the initial condition for the spin vector be $s_0 = (0,0,1)$. In this case, Ω_{MDM} is tilted at an angle $\alpha = \arctan\left(\Omega_{B_z}/\delta\Omega_{\mathrm{decoh}}\right)$ from the vertical axis, and spin precesses in a circle of the unit sphere centered at $\left(0, \sin\alpha\cos\alpha, \sin^2\alpha\right)$. The radius of the circle is $\cos\alpha$.

We note that

$$\sin\alpha = \frac{\tan\alpha}{\sqrt{1+\tan^2\alpha}} = \\ = \frac{\Omega_{B_z}}{\sqrt{\Omega_{B_z}^2 + \delta\Omega_{\text{decoh}}^2}}$$

and

$$\cos\alpha = \frac{1}{\sqrt{1+\tan^2\alpha}} = \\ = \frac{\delta\Omega_{\text{decoh}}}{\sqrt{\Omega_{B_z}^2 + \delta\Omega_{\text{decoh}}^2}}.$$

The solution of the Thomas–BMT equation in this case is

$$\begin{aligned} s(t) = &\left(0, \sin\alpha\cos\alpha, \sin^2\alpha\right) + \\ &+ \cos\alpha\left(\sin\left(|\Omega_{\text{MDM}}|t\right), -\sin\alpha\cos\left(|\Omega_{\text{MDM}}|t\right), \cos\alpha\cos\left(|\Omega_{\text{MDM}}|t\right)\right) = \\ &= \frac{1}{|\Omega_{\text{MDM}}|^2}\Big(\delta\Omega_{\text{decoh}}|\Omega_{\text{MDM}}|\sin\left(|\Omega_{\text{MDM}}|t\right), \\ &\Omega_{B_z}\delta\Omega_{\text{decoh}}\left(1-\cos\left(|\Omega_{\text{MDM}}|t\right)\right), \Omega_{B_z}^2 + \delta\Omega_{\text{decoh}}^2\cos\left(|\Omega_{\text{MDM}}|t\right)\Big), \end{aligned} \tag{6.17}$$

where $|\Omega_{\text{MDM}}| = \sqrt{\Omega_{B_z}^2 + \delta\Omega_{\text{decoh}}^2}$ (see also [99, p. 3396]).

Changing the lattice traversal direction from CW to CCW reverses the direction of spin vector's MDM angular velocity; however, it also reverses the direction of the z axis, which is why the same eq. 6.17 applies to both CW and CCW traversal modulo the difference between the inverse and the reverse spin transfer maps. Thus, the method of B_x magnetic error field component mitigation described above is not applicable to the B_z case, and the solution is to minimize Ω_{B_z} using additional trim coils [99, p. 3396].

6.5.6 EDM Measurement Accuracy with Rotational Magnet Misalignments

D. Eversmann *et al.* [34, 95, p. 3] presented a new method of spin tune measurement, which enables measurement of spin angular frequency Ω with standard deviation

$$\sigma_\Omega = 2\delta\varepsilon_A\sqrt{\frac{6}{N}}\frac{1}{T},$$

where N is the total number of recorded events, $\delta\varepsilon_A$ is the relative error of polarization asymmetry measurement, and T is the measurement duration.

The FS and QFS methods require the measurement duration to be at least $\sim 1000\,\mathrm{s}$. We assume realistic values of measurement duration $T_1 \simeq 1000\,\mathrm{s}$ and the number of particles per fill $N_0 = 10^{11}$. Detector count events are aggregated into groups of 2000, resulting in $\delta\varepsilon_A \approx 0.03$. With beam exhaustion at measurement of 75% and polarimeter efficiency of $R = 1\%$, the x components Ω_x^{CW} and Ω_x^{CCW} of spin angular frequencies may be measured with standard deviation $\sigma_\Omega^1 \simeq 2.4 \times 10^{-7}\,\mathrm{rad/s}$ [95, p. 2277].

With a realistic annual accelerator beamtime of $T_{\mathrm{ann}} \simeq 6000\,\mathrm{h}$, the standard deviation of

$$\sigma_\Omega^{\mathrm{ann}} = \frac{\sigma_\Omega^1}{\sqrt{T_{\mathrm{ann}}/T_1}} \simeq 1.6 \times 10^{-9}\,\mathrm{rad/s}$$

may be achieved. Considering that an EDM of $\sim 10^{-30}\,e\cdot\mathrm{cm}$ would, by eq. 6.1, result in an EDM angular velocity of $\Omega_{\mathrm{EDM},x} \simeq 10^{-8}\,\mathrm{rad/s}$, the standard deviation $\sigma_\Omega^{\mathrm{ann}} \simeq 1.6 \times 10^{-9}\,\mathrm{rad/s}$ would enable EDM signal measurement using eq. 6.16 with an accuracy of $10^{-30}\,e\cdot\mathrm{cm}$ [95, p. 2277].

To maintain the EDM signal measurement accuracy of $10^{-30}\,e\cdot\mathrm{cm}$, it is necessary, considering eq. 6.17, to minimize the z component Ω_{B_z} of the MDM angular frequency due to rotational magnet misalignment using trim coils to a substantially lower level than $\sigma_\Omega^{\mathrm{ann}} \simeq 10^{-9}\,\mathrm{rad/s}$, such as $\Omega_{B_z} \simeq 10^{-10}\,\mathrm{rad/s}$ [99, p. 3396].

Similarly, when changing the direction of the lattice traversal, the magnetic field must be reversed with a sufficient accuracy for $\left|\Omega_{B_x}^{\mathrm{CW}} + \Omega_{B_x}^{\mathrm{CCW}}\right|$ to be at a substantially lower level than $\sigma_\Omega^{\mathrm{ann}} \simeq 10^{-9}\,\mathrm{rad/s}$, such as $\left|\Omega_{B_x}^{\mathrm{CW}} + \Omega_{B_x}^{\mathrm{CCW}}\right| \simeq 10^{-10}\,\mathrm{rad/s}$.

6.5.7 Spin-Tracking Datasets

To assess the feasibility of EDM signal measurement by tracking the lattice in CW and CCW directions and applying eq. 6.16, it is necessary to study spin dynamics and decoherence in the vertical plane in the presence of rotational magnet misalignments.

In the context of our systematic errors study, we produced more than 46 spin-tracking datasets. These datasets include combinations of the following: (1) *Senichev 6.3*, *E+B*, and *BNL* lattices; (2) optimization by the SFP, SDP, SFP1, and SDP2 sextupole families, as appropriate, as well as no optimization; (3) $-0.5x$, $0x$, $0.05x$, $0.5x$, $1x$, and $2x$ corrective horizontal spin kicks, where $1x$ is exactly satisfies the QFS/FS condition and $0x$ is the lack of a spin kick; and (4) normally distributed random-variate rotational magnet misalignments with standard deviation $\sigma(\alpha) = 10^{-4}$ rad and, to ensure that vertical spin decoherence is proportional to rotational magnet misalignments, $\sigma(\alpha) = 5 \times 10^{-5}$ rad. As expected, the vertical spin decoherence was proportional to the rotational magnet misalignments.

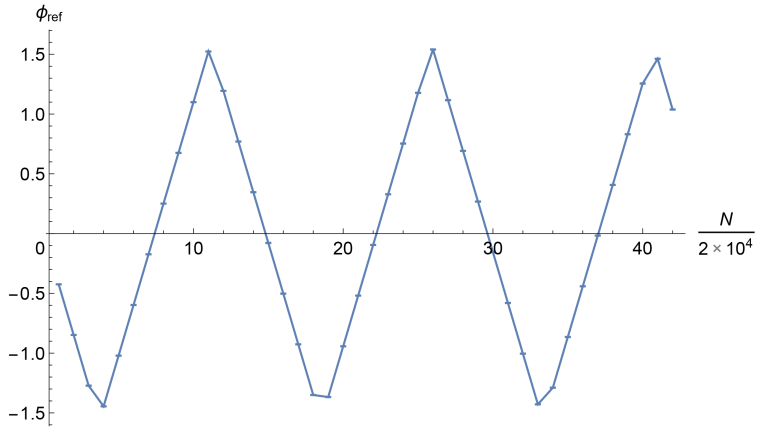
Each spin-tracking dataset comprises 24 *COSY INFINITY* output files, one for each combination of the following: (1) B_x and B_z error fields and no error field; (2) CW and CCW lattice traversal directions; (3) spin tracking in the horizontal and the vertical plane; and (4) particles launched with equidistant initial x and δ_K offsets.

6.5.8 Spikes in the Horizontal Spin Decoherence Measure at the Poles

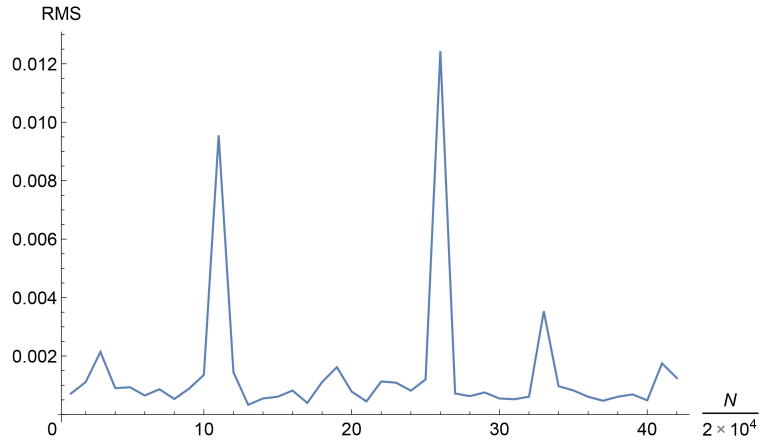
We often observed apparent horizontal spin decoherence spikes when the spin vectors of the particle bunch passed through the poles of the spherical coordinate system at azimuthal angles $\phi = \pm\pi/2$, as illustrated in Fig. 6.12. These spikes are due to the use of spin decoherence measures σ_θ and σ_ϕ, where θ is the polar angle, and do not have a physical meaning.

To confirm that this interpretation is correct, we considered a minimal test case of three particles with initial spin spherical coordinates $(r, \theta, \phi) = (1, -\varepsilon, 0)$, $(1, 0, 0)$, and $(1, \varepsilon, 0)$. We rotated the three spins around the x axis in *Mathematica*, and observed the same pattern of spikes in the horizontal spin decoherence σ_θ at azimuthal angles $\phi = \pm\pi/2$.

The spikes were not a hindrance to our systematic errors study; however, we note that this spikes effect can be avoided by using a different horizontal spin decoherence measure, such as $\sigma_{s_{xy}}$, where $\left(s_x, s_y, s_z\right)$ are the spin vector components in Cartesian coordinates (x, y, z) and $s_{xy} = \sqrt{s_x^2 + s_y^2}$.



(a) Azimuthal angle ϕ_{ref} of the reference particle's spin as a function of the number N of tracked turns.



(b) RMS of the spin polar angles Θ of the particle bunch as a function of the number N of tracked turns.

Figure 6.12: In this example, the RMS of the spin polar angles Θ of the particle bunch (sub-figure b) spikes when the azimuthal angle ϕ_{ref} of the reference particle's spin (sub-figure a) passes through the poles $\phi_{\text{ref}} = \pm\pi/2$. Particles were launched through the *Senichev E+B* lattice with initial x offsets up to $x_{\text{i}} = \pm 5 \times 10^{-3}$ m. $1x$ corrective spin kicks were used. Tracking was performed in the CCW direction, with a B_x magnetic error field component, in fringe field mode `FR3`, with the RF cavity on. The RF cavity frequency was $\nu = 4.9\,\text{MHz}$, and the RF cavity voltage was $V = 100\,\text{kV}$. The horizontal spin decoherence objective function `OBJ` was pre-optimized by the SDP sextupole family strength.

6.5.9 Vertical Spin Decoherence Analysis

The plots in Fig. 6.13 show typical vertical spin decoherence as a function of the number of turns in the *Senichev E+B* and *Senichev BNL* lattices in case of no corrective spin kicks. The plots in Fig. 6.14 show typical vertical spin decoherence as a function of the number of turns in the *Senichev E+B* and *Senichev BNL* lattices in case of $1x$ corrective spin kicks.

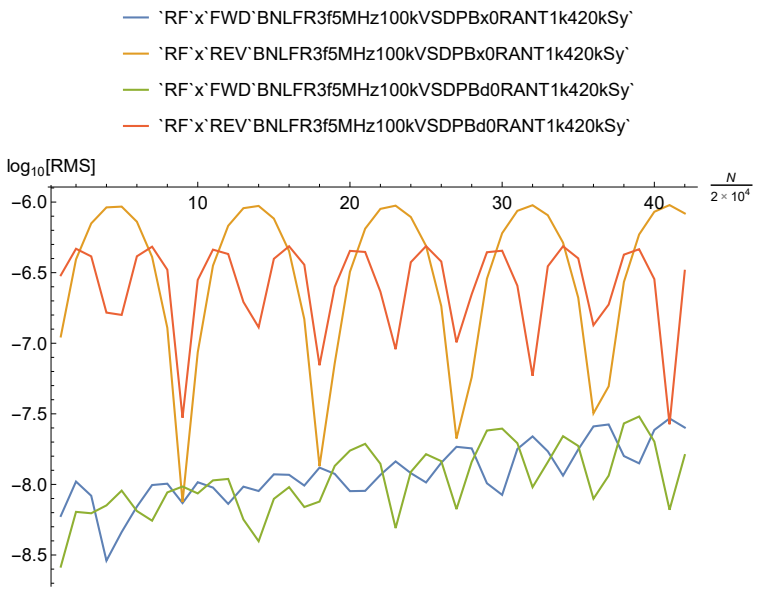
Vertical spin decoherence often remains in the same range, or grows in case of CW lattice traversal within the range bounding vertical spin decoherence for the CCW lattice traversal, for at least 4.2×10^5 turns. This indicates that suppression of spin decoherence using an RF cavity and optimized sextupole family strengths may be sufficient for a 10^3 s SCT in the vertical plane for the EDM signal measurement via vertical spin precession frequencies and tracking particles through the FS or QFS lattice in the opposite directions to be feasible.

6.6 Conclusion

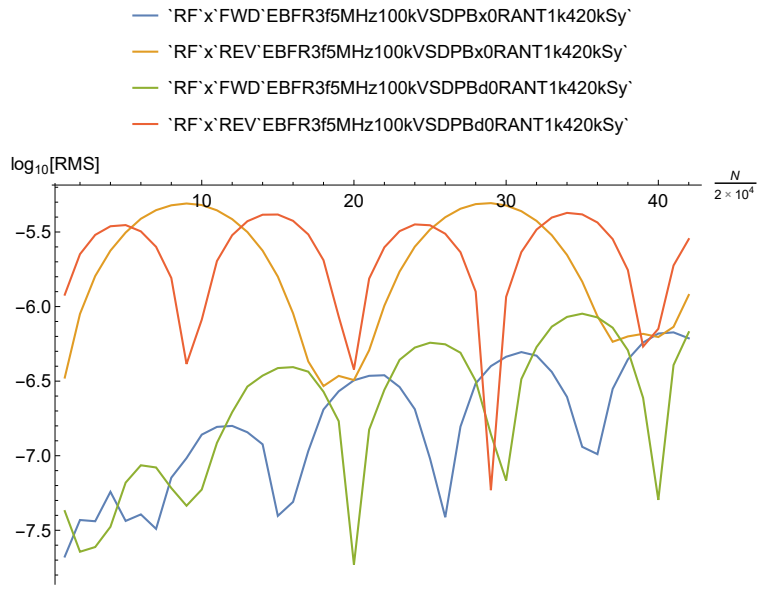
Our spin decoherence study of FS and QFS lattices shows that, with an optimized sextupole family strength, the spin decoherence often remained in the same range for at least 4.2×10^5 turns. This is promising with respect to the requirement of maintaining a spin coherence time of 10^3 seconds to possibly build a measurable EDM signal. The QFS structure decoherence is similar to that of the FS structure.

Furthermore, the systematic errors study shows that, for at least 4.2×10^5 turns, the vertical spin decoherence due to rotational magnet misalignments often remained in the same range (or grew within the range of the spin decoherence curve for tracking in the opposite direction) in both FS (*Senichev BNL*) and QFS (*Senichev E+B*) lattices. The vertical spin decoherences in the FS and QFS lattices were qualitatively very similar and quantitatively within about 1–2 orders between each other. As the sextupole family strengths were optimized for the CW lattice traversal, vertical spin decoherence is somewhat higher when tracking in the CCW direction than when tracking in the CW direction.

We estimate for the QFS and FS lattices that one year of measurement statistics will result in an

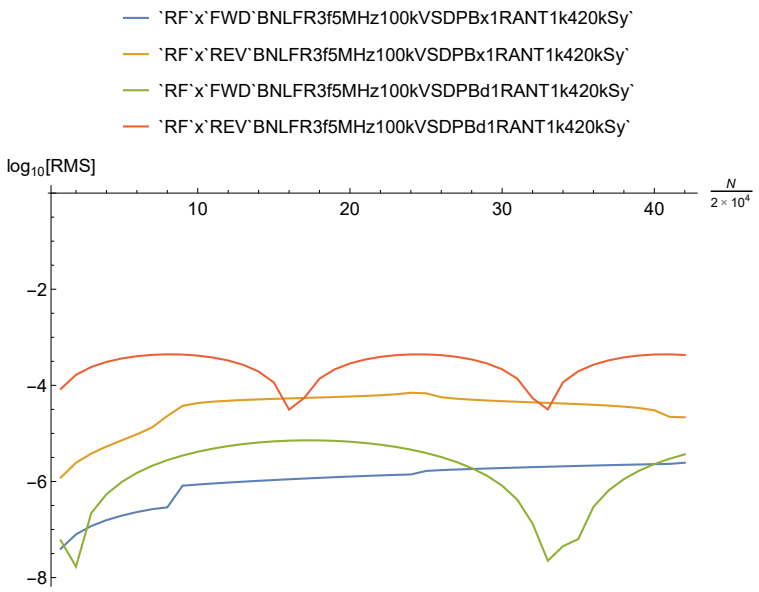


(a) *Senichev BNL* FS lattice.

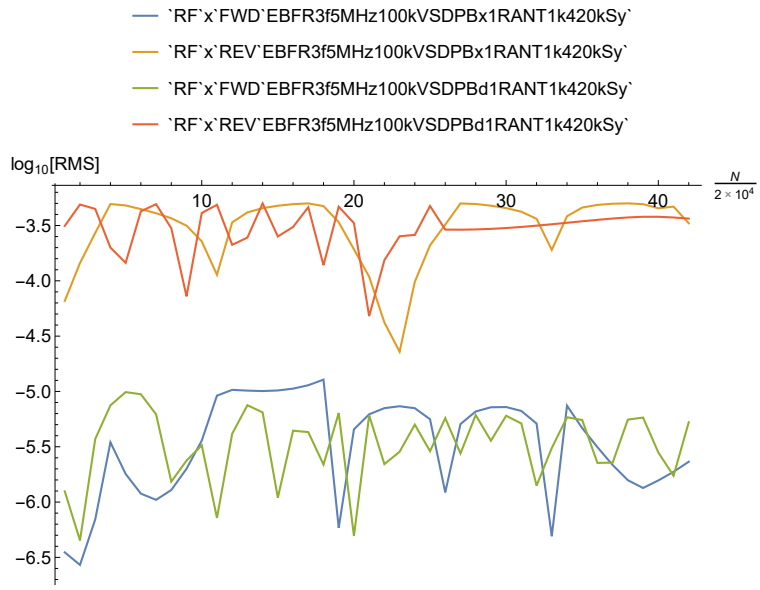


(b) *Senichev E+B* QFS lattice.

Figure 6.13: RMS of the vertical spin decoherence as a function of the number N of tracked turns in the *Senichev BNL* and *E+B* lattices, up to 4.2×10^5 turns. No corrective spin kicks were used. The vertical spin decoherence remains in approximately in the same range during the tracking for the CCW lattice traversal and grows below the respective upper bound for the CW lattices. For the CW direction, tracking with a B_x and B_z magnetic error field component is denoted by blue and green curve colors, respectively. For the CCW direction, tracking with a B_x and B_z magnetic error field component is denoted by orange and red curve colors, respectively. Particles were launched with initial x offsets up to $x_\mathrm{i} = \pm 5 \times 10^{-3}$ m. Tracking was performed with the RF cavity on, with the spin decoherence function `OBJ` pre-optimized by the SDP sextupole family strength, and in fringe field mode `FR3`. The RF cavity harmonic number was $h = 5$, and the RF cavity voltage was $V = 100\,\mathrm{kV}$.



(a) *Senichev BNL* FS lattice.



(b) *Senichev E+B* QFS lattice.

Figure 6.14: RMS of the vertical spin decoherence as a function of the number N of tracked turns in the *Senichev BNL* and *E+B* lattices, up to 4.2×10^5 turns. $1x$ corrective spin kicks were used. The vertical spin decoherence remains in approximately the same range. For the CW direction, tracking with a B_x and B_z magnetic error field component is denoted by blue and green curve colors, respectively. For the CCW direction, tracking with a B_x and B_z magnetic error field component is denoted by orange and red curve colors, respectively. Particles were launched with initial x offsets up to $x_i = \pm 5 \times 10^{-3}$ m. Tracking was performed with the RF cavity on, with the spin decoherence function `OBJ` pre-optimized by the SDP sextupole family strength, and in fringe field mode `FR3`. The RF cavity harmonic number was $h = 5$, and the RF cavity voltage was $V = 100\,\text{kV}$.

EDM signal measurement accuracy of $\sim 10^{-30}\, e \cdot \mathrm{cm}$.

Possible further research directions include (1) optimizing lattice parameters to more precisely satisfy the FS and QFS conditions; (2) optimizing spin decoherence for both CW and CCW lattice traversals simultaneously; (3) performing spin tracking of the lattices for larger numbers of turns; (4) using DA normal form methods to minimize spin decoherence in the FS and QFS lattices simultaneously by all sextupole families, which is otherwise nontrivial to perform automatically due to convergence to local minima; (5) performing spin tracking with 3D distribution of the launched particles in phase space for spin decoherence and systematic errors analysis.

APPENDICES

APPENDIX A

CONFORMAL MAPPING METHODS

A.1 Elliptic Functions and Integrals

In Legendre's notation, the **incomplete integral of the first kind** (or **Legendre elliptic integral of the first kind**) is defined as

$$u = \mathrm{F}(\varphi, k) = \int_0^{\varphi} \frac{dt}{\sqrt{1 - k^2 \sin^2 t}},$$

where the parameter φ is called the **Jacobi amplitude**[1].

Carlson's elliptic integral of the first kind $\mathrm{R_F}(x, y, z)$ is defined as

$$\mathrm{R_F}(x, y, z) = \frac{1}{2} \int_0^{+\infty} \frac{dt}{\sqrt{(t + x)(t + y)(t + z)}},$$

where $x, y, z \geq 0$.

The function F is equivalently expressed in terms of $\mathrm{R_F}(x, y, z)$ as

$$\mathrm{F}(\varphi, k) = \sin(\varphi)\, \mathrm{R_F}\left(\cos^2(\varphi), 1 - k^2 \sin^2(\varphi), 1\right).$$

The DA value of $\mathrm{R_F}$ is calculated as the sum of the scalar value (i.e., the value in $\mathbb{R}$) of $\mathrm{R_F}$ and the antiderivative of the DA value of the first derivative of $\mathrm{R_F}$. A numerical method to approximate the scalar value of $\mathrm{R_F}$ is described in [87, pp. 309–316].

The **Jacobi elliptic functions** sn, cn, and dn[2] are defined as

$$\begin{aligned} \mathrm{sn}(u|k) &= \sin\left(\mathrm{F}^{-1}(u, k)\right), \\ \mathrm{cn}(u|k) &= \cos\left(\mathrm{F}^{-1}(u, k)\right), \\ \mathrm{dn}(u|k) &= \sqrt{1 - k^2 \sin^2\left(\mathrm{F}^{-1}(u, k)\right)}, \end{aligned} \tag{A.1}$$

[1] We use φ to denote the Jacobi amplitude only in the context of definitions of elliptic functions and integrals. Otherwise, φ usually denotes an electrostatic potential (or voltage).

[2] Additional information on the Jacobi elliptic functions can be found at `http://mathworld.wolfram.com/JacobiEllipticFunctions.html`.

where $0 < k^2 < 1$ and $k = \mathrm{mod}\, u$ is the **elliptic modulus**.

A numerical method to approximate scalar values of sn, cn, dn, and φ as functions of u and k is described in [26, sec. 22.20].

The DA values of sn, cn, and dn can be obtained by applying eqns. A.1 to the DA value of φ. The DA value of φ can be calculated as follows:

1. Calculate the DA value of $\mathscr{M} = \mathrm{F}(\varphi(\mathrm{cons}(u), k), k)$, where cons takes the constant part of a DA value[3].

2. Calculate the inverse DA value of $\mathscr{N} = \mathscr{M} - \mathrm{cons}(\mathscr{M})$.

3. The DA value of the Jacobi amplitude is the sum of the DA inverse of $\mathscr{N}$ and the constant part $\mathrm{cons}(\varphi(\mathrm{cons}(u), k))$ of the Jacobi amplitude.

The **complete elliptic integral of the first kind** $\mathrm{K}(k)$ is defined as[4]

$$\mathrm{K}(k) = \mathrm{F}\left(\frac{1}{2}\pi, k\right),$$

where $\mathrm{F}(\varphi, k)$ is the incomplete elliptic integral of the first kind. Note that the incomplete elliptic integral F can be also written as

$$\mathrm{K}(k) = \frac{\pi}{2} \sum_{n=0}^{+\infty} \left[\frac{(2\pi - 1)!!}{(2n)!!}\right]^2 k^{2n}.$$

The parameter $m = k^2$ is often used in lieu of the elliptic modulus k as an argument of the Jacobi elliptic functions and the elliptic integrals; for example, $\mathrm{sn}(u|m)$ is equivalent to $\mathrm{sn}(u|k)$. In effect, there are two versions of each elliptic function and integral: one takes the parameter m, the other takes the elliptic modulus k as an argument, and whether m or k is supplied as that argument determines the version to be used.

[3] In terms of [11, ch. 2], $\mathrm{cons}(a)$ has the same meaning as $[a]_0$ for any DA value a.

[4] Additional information on the elliptic integral of the first kind can be found at `http://mathworld.wolfram.com/CompleteEllipticIntegraloftheFirstKind.html`.

A.2 Laplace Equation and Conformal Mappings

The electrostatic potential inside a beam physics element can be obtained by solving the Laplace equation $\triangle\varphi = 0$ for electrostatic potential φ with Dirichlet boundary conditions as the voltages at the boundaries, e.g., the plates and the rectangular enclosure of an electrostatic quadrupole. In many cases, for the main field of a particle optical element, the problem may be accurately modeled by 2D geometry. This also holds for the fringe field of a semi-infinite electrostatic capacitor. The Laplace equation then has the 2D form

$$\frac{\partial^2\varphi}{\partial x^2} + \frac{\partial^2\varphi}{\partial y^2} = 0.$$

The isomorphism $(x,y) \leftrightarrow z = x + iy$ gives an equivalent representation of the 2D geometry in $\mathbb{R}^2$ on the complex plane $\mathbb{C}$.

If a complex function $f(x+iy) = u(x,y) + iv(x,y)$ is complex-differentiable at every point of a region $R \subseteq \mathbb{C}$, it is called **holomorphic** (or **analytic**) in the region R. In turn, a complex function f is **complex-differentiable** at point $z_0 = x_0 + iy_0$ if there exists a neighborhood $U(z_0)$ of z_0 such that (1) $f \in C^1(U(z_0))$, i.e., f has continuous first partial derivatives in $U(z_0)$; and (2) f satisfies the Cauchy–Riemann equations

$$\frac{\partial u}{\partial x} = \frac{\partial v}{\partial y}, \quad \frac{\partial v}{\partial x} = -\frac{\partial u}{\partial y}$$

in $U(z_0)$.

Holomorphic functions satisfy the 2D Laplace equation, i.e.,

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0, \quad \frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} = 0.$$

A **conformal mapping** (or **conformal map**) is a transformation $w = f(z)$ that is locally angle preserving. A holomorphic function is conformal at point z if $f'(z) \neq 0$. Conformal maps are also called **biholomoprhic**, which reflects that both a conformal mapping and its inverse are holomorphic.

The domain of a conformal mapping is called the **canonical domain**, and the image of a conformal mapping is called the **physical domain**.

A **Schwarz–Christoffel mapping** is a conformal mapping from the upper half-plane as the canonical domain to the interior of a polygon as the physical domain. The polygon is generalized in the sense that it may have vertices at the complex infinity ∞ [29, 30].

Consider an n-sided polygon as the physical domain. The preimages $z_j = f^{-1}\left(w_j\right)$ of the polygon's vertices w_j, where $j = 1, 2, \ldots, n$, are called **prevertices**. We use the notation $\pi\alpha_j$, $j = 1, 2, \ldots, n$, for the interior angles of the polygon. For finite vertices w_j, the trivial condition $0 < \alpha_j \leq 2$ applies. For vertices at the complex infinity, i.e., $w_j = \infty$, the respective α's are in the interval $-2 \leq \alpha_j \leq 0$ [30, pp. 1–3].

Variants of the Schwarz–Christoffel mapping include using a unit disk, a bi-infinite strip, or a rectangle as the canonical domain. We will use the general term "conformal mapping" for variants of the Schwarz–Christoffel mapping.

For canonical domains as the upper half-plane, a unit disk, a bi-infinite strip, or a rectangle, the analytic expressions for the respective conformal maps f or their derivatives f' can be found in [30]. Those formulas contain prevertices z_j as parameters, which can be obtained using optimization methods.

The electrostatic potential for a cross section or a longitudinal section modeled by a generalized polygon may be found by obtaining a conformal mapping from a suitable canonical domain to the polygon. A bi-infinite strip is a suitable canonical domain if the polygon comprises two groups of consecutive sides characterized by the same constant Dirichlet boundary condition, with two constant values in total. A rectangular part of a bi-infinite strip is a suitable canonical domain when the physical domain is a logical (or generalized) quadrilateral. In this context, a generalized quadrilateral is a polygon comprising four groups of sides, where the groups are defined as follows:

1. the groups are sets of consecutive sides;

2. two consecutive sides are in different groups if and only if one has a Dirichlet boundary condition and the other one has a von Neumann boundary condition; and

3. each Dirichlet boundary condition group is characterized by one constant value as the Dirichlet

boundary condition.

If the solution of the Laplace equation in the canonical domain is ϕ, the solution of the Laplace equation in the physical domain is $\varphi = \phi \circ f^{-1}$. In practice, the electrostatic potential is usually the appropriately selected, shifted, and scaled real or imaginary part of f^{-1}.

The solution for the electrostatic potential obtained this way is fully Maxwellian in the sense that the analytic formula for f' results in the solution for the potential satisfying the Laplace equation.

As described in [30, 108], inverse conformal mapping $g = f^{-1}$ may be obtained by

1. solving the equation

$$f(g(w)) - w = 0 \tag{A.2}$$

   using the Newton–Raphson or another numerical method; or

2. solving the ODE

$$\frac{dg(w)}{dw} = \frac{1}{f'(g(w))}, \quad g(w_0) = z_0. \tag{A.3}$$

A.3 *Schwarz–Christoffel Toolbox*

There is a number of specialized software packages for Schwarz–Christoffel and conformal mapping computations, including the *FORTRAN* packages *SCPACK*, *DSCPACK*, and *CONFPACK*, as well as the *MATLAB* package *Schwarz–Christoffel Toolbox* (*SC Toolbox*) [28]. The *SC Toolbox* may be downloaded from `http://www.math.udel.edu/~driscoll/SC/`. See its User's Guide [29] for usage details.

As described on the *SC Toolbox*'s website [27], "the *SC Toolbox* is a problem-solving environment for computation and interaction with conformal maps to regions bounded by polygons, including unbounded regions, logical quadrilaterals, and channels". The interactive features of the *SC Toolbox* include the optional interactive selection of polygon points and Dirichlet boundary conditions (e.g., voltages).

We use the *SC Toolbox* to obtain prevertices, strip length of the canonical domain, and other information, as applicable, for conformal maps specified by vertices and interior angles of polygonal models.

The function `polygon(W,ALPHA)` constructs a polygon object, which is the *SC Toolbox* representation of a polygon in the physical domain. The parameter `W` is a complex vector

$$\mathtt{W} = [\mathtt{w_1}, \cdots, \mathtt{w_n}]$$

that specifies the polygon vertices w_j, $j = 1, 2, \ldots, n$, in consecutive order; and the optional parameter `ALPHA` is a real vector

$$\mathtt{ALPHA} = [\mathtt{alpha_1}, \cdots, \mathtt{alpha_n}]$$

that specifies the respective values α_j, $j = 1, 2, \ldots, n$, defined above, i.e., the interior angles at the vertices divided by π. If the polygon has vertices at the complex infinity ∞, (1) the parameter `ALPHA` is required, and (2) polygon vertices must be specified in the clockwise order[5] [29].

Once a polygon object is created, e.g., by

```
p=polygon([0,1,2+i/2,1+i,i])
```

a conformal mapping can be computed from a canonical domain to the interior of that polygon. For example,

```
f=rectmap(p,[1,2,4,5])
```

returns a conformal mapping `f` from the interior of a rectangle to the interior of the polygon `p`. Here, the second parameter of the function `rectmap` specifies the indices j of the prevertices z_j that are the corners of the rectangle [29].

When called without a parameter from the *MATLAB* command line, the object `f` returns a list of vertices, prevertices, α values, and other information.

[5] Sometimes, it may be necessary to rotate (shift) the order of the vertices of the polygon in the `W` vector or insert a "guiding" degenerate vertex between a vertex at the infinity and another vertex. A degenerate vertex has the internal angle of π and does not change the function graph of the polygon.

By default, the parameter m is not in this list. Internally, the *SC Toolbox* calculates and uses the strip length L instead of the parameter m. This serves to address the floating-point error in $1-m$ for small values of the parameter m. *Mathematica* and similar software use highly accurate real number representations similar to Unum [47], where this is not an issue. The parameter m is related to L as

$$m = \exp(-2\pi L).$$

The strip length L can be displayed for the object `f`, e.g., by the commands

```
z = parameters(f);
disp(num2str(z.stripL,'%22.16e'))
```

The constant c may be obtained as the quotient of (1) the values of `evaldiff(f,z)`, which returns $f'(z)$ in the *SC Toolbox*; and (2) the respective values of the analytic expression of $f'(z)$ without the constant c.

To evaluate the inverse of a conformal mapping `f`, by default, the *SC Toolbox* solves the ODE from eq. A.3 and then uses Newton iterations to improve the solution. Optionally, the *SC Toolbox* can either only use the ODE solution or only solve eq. A.2 using Newton iterations.

APPENDIX B

SC TOOLBOX POLYGONAL MODELS OF SEMI-INFINITE CAPACITORS

B.1 One Capacitor with Infinitely Thick Plates

The following is a polygonal model code representation of the semi-infinite capacitor with infinitely thick plates in the *SC Toolbox* in terms of vertices w_j and interior angles $\alpha_j\pi$. Vertices 0 and $2i$ were added as guiding points for the *SC Toolbox*'s algorithm.

```
p1 = polygon([i, Inf, 0, Inf, 2*i], [3/2, 0, 1, -1/2, 1])
```

B.2 One Capacitor with Plates of $D/20$ Thickness

The following is a polygonal model code representation of the semi-infinite capacitor with plates of $D/20$ thickness in the *SC Toolbox* in terms of vertices w_j and interior angles $\alpha_j\pi$. Vertices 0 and $-1+1.1i$ were added as guiding points for the *SC Toolbox*'s algorithm.

```
p2 = polygon([1.1i, i, Inf, 0, Inf, -1+1.1i], [3/2, 3/2, 0, 1, -1, 1])
```

B.3 One Capacitor with Plates of $D/20$ Thickness and Rounded Edges

The following is a polygonal model code representation of the semi-infinite capacitor with plates of $D/20$ thickness and rounded edges in the *SC Toolbox* in terms of vertices w_j and interior angles $\alpha_j\pi$. Vertices 0 and $-1+1.1i$ were added as guiding points for the *SC Toolbox*'s algorithm.

```
p2a = polygon([1.1i, 0.0037365+1.09986i, 0.00745211+1.09944i,...
    0.011126+1.09875i, 0.0147378+1.09778i, 0.0182671+1.09654i,...
    0.0216942+ 1.09505i, 0.025+1.0933i, 0.028166+1.09131i,...
    0.0311745+ 1.08909i, 0.0340086+1.08665i, 0.0366526+1.08401i,...
    0.0390916+ 1.08117i, 0.0413119+1.07817i, 0.0433013+1.075i,...
    0.0450484+ 1.07169i, 0.0465437+1.06827i, 0.0477786+1.06474i,...
```

```
0.0487464+ 1.06113i, 0.0494415+1.05745i, 0.0498602+1.05374i,...
0.05+ 1.05i, 0.0498602+1.04626i, 0.0494415+1.04255i,...
0.0487464+ 1.03887i, 0.0477786+1.03526i, 0.0465437+1.03173i,...
0.0450484+ 1.02831i, 0.0433013+1.025i, 0.0413119+1.02183i,...
0.0390916+ 1.01883i, 0.0366526+1.01599i, 0.0340086+1.01335i,...
0.0311745+ 1.01091i, 0.028166+1.00869i, 0.025+1.0067i,...
0.0216942+ 1.00495i, 0.0182671+1.00346i, 0.0147378+1.00222i,...
0.011126+ 1.00125i, 0.00745211+1.00056i, 0.0037365+1.00014i,...
0.+1.i, inf, 0, inf, -1+1.1i],...
[85/84, 43/42, 43/42, 43/42, 43/42, 43/42, 43/42, 43/42, 43/42,...
43/42, 43/42, 43/42, 43/42, 43/42, 43/42, 43/42, 43/42, 43/42,...
43/42, 43/42, 43/42, 43/42, 43/42, 43/42, 43/42, 43/42, 43/42,...
43/42, 43/42, 43/42, 43/42, 43/42, 43/42, 43/42, 43/42, 43/42,...
43/42, 43/42, 43/42, 43/42, 43/42, 43/42, 85/84, 0, 1, -1, 1]);
```

B.4 One Capacitor with Infinitely Thin Plates

The following is a polygonal model code representation of the semi-infinite capacitor with infinitely thin plates in the *SC Toolbox* in terms of vertices w_j and interior angles $\alpha_j\pi$. Vertices 0 and $-1+i$ were added as guiding points for the *SC Toolbox*'s algorithm.

```
p0 = polygon([i, Inf, 0, Inf, -1+i], [2, 0, 1, -1, 1])
```

B.5 Adjacent Capacitors with Plates of $3D/4$ Thickness, Symmetric Voltages, and Rounded Edges

The following is a polygonal model code representation of the two adjacent semi-infinite capacitors with plates of $3D/4$ thickness, symmetric voltages, and rounded edges in the *SC Toolbox* in terms of vertices w_j and interior angles $\alpha_j\pi$. Vertices 0, $6+2.5i$, and $-6+2.5i$ were added as guiding points for the *SC Toolbox*'s algorithm.

```
p3b = polygon([-5.+2.5i, -4.94395+2.4979i, -4.88822+2.49162i,...
    -4.83311+2.4812i, -4.77893+2.46668i, -4.72599+2.44816i,...
    -4.67459+2.42573i, -4.625+2.39952i, -4.57751+2.36968i,...
    -4.53238+2.33637i, -4.48987+2.29979i, -4.45021+2.26013i,...
    -4.41363+2.21762i, -4.38032+2.17249i, -4.35048+2.125i,...
    -4.32427+2.07541i, -4.30184+2.02401i, -4.28332+1.97107i,...
    -4.2688+1.91689i, -4.25838+1.86178i, -4.2521+1.80605i,...
    -4.25+1.75i, -4.2521+1.69395i, -4.25838+1.63822i,...
    -4.2688+1.58311i, -4.28332+1.52893i, -4.30184+1.47599i,...
    -4.32427+1.42459i, -4.35048+1.375i, -4.38032+1.32751i,...
    -4.41363+1.28238i, -4.45021+1.23987i, -4.48987+1.20021i,...
    -4.53238+1.16363i, -4.57751+1.13032i, -4.625+1.10048i,...
    -4.67459+1.07427i, -4.72599+1.05184i, -4.77893+1.03332i,...
    -4.83311+1.0188i, -4.88822+1.00838i, -4.94395+1.0021i, -5.+1.i,...
    inf, 0, inf, 5.+1.i, 4.94395+1.0021i, 4.88822+1.00838i,...
    4.83311+1.0188i, 4.77893+1.03332i, 4.72599+1.05184i,...
    4.67459+1.07427i, 4.625+1.10048i, 4.57751+1.13032i,...
    4.53238+1.16363i, 4.48987+1.20021i, 4.45021+1.23987i,...
    4.41363+1.28238i, 4.38032+1.32751i, 4.35048+1.375i,...
    4.32427+1.42459i, 4.30184+1.47599i, 4.28332+1.52893i,...
    4.2688+1.58311i, 4.25838+1.63822i, 4.2521+1.69395i, 4.25+1.75i,...
    4.2521+1.80605i, 4.25838+1.86178i, 4.2688+1.91689i,...
    4.28332+1.97107i, 4.30184+2.02401i, 4.32427+2.07541i,...
    4.35048+2.125i, 4.38032+2.17249i, 4.41363+2.21762i,...
    4.45021+2.26013i, 4.48987+2.29979i, 4.53238+2.33637i,...
    4.57751+2.36968i, 4.625+2.39952i, 4.67459+2.42573i,...
    4.72599+2.44816i, 4.77893+2.46668i, 4.83311+2.4812i,...
    4.88822+2.49162i, 4.94395+2.4979i, 5.+2.5i, 6.+2.5i, inf,...
```

```
    -6.+2.5i ],...
    [85/84, 43/42, 43/42, 43/42, 43/42, 43/42, 43/42, 43/42, 43/42,...
    43/42, 43/42, 43/42, 43/42, 43/42, 43/42, 43/42, 43/42, 43/42,...
    43/42, 43/42, 43/42, 43/42, 43/42, 43/42, 43/42, 43/42, 43/42,...
    43/42, 43/42, 43/42, 43/42, 43/42, 43/42, 43/42, 43/42, 43/42,...
    43/42, 43/42, 43/42, 43/42, 43/42, 43/42, 85/84, 0, 1, 0, 85/84,...
    43/42, 43/42, 43/42, 43/42, 43/42, 43/42, 43/42, 43/42, 43/42,...
    43/42, 43/42, 43/42, 43/42, 43/42, 43/42, 43/42, 43/42, 43/42,...
    43/42, 43/42, 43/42, 43/42, 43/42, 43/42, 43/42, 43/42, 43/42,...
    43/42, 43/42, 43/42, 43/42, 43/42, 43/42, 43/42, 43/42, 43/42,...
    43/42, 43/42, 43/42, 43/42, 43/42, 85/84, 1, -1, 1]);
```

B.6 Adjacent Capacitors with Plates of $3D/4$ Thickness, Antisymmetric Voltages, and Rounded Edges

The following is a polygonal model code representation of the two adjacent semi-infinite capacitors with plates of $3D/4$ thickness, antisymmetric voltages, and rounded edges in the *SC Toolbox* in terms of vertices w_j and interior angles $\alpha_j\pi$. Vertices 0 and $-6+i$ were added as guiding points for the *SC Toolbox*'s algorithm.

```
p6as1 = polygon([-6.+2.5i, -5.+2.5i, -4.94395+2.4979i,...
-4.88822+2.49162i, -4.83311+2.4812i, -4.77893+2.46668i,...
-4.72599+2.44816i, -4.67459+2.42573i, -4.625+2.39952i,...
-4.57751+2.36968i, -4.53238+2.33637i, -4.48987+2.29979i,...
-4.45021+2.26013i, -4.41363+2.21762i, -4.38032+2.17249i,...
-4.35048+2.125i, -4.32427+2.07541i, -4.30184+2.02401i,...
-4.28332+1.97107i, -4.2688+1.91689i, -4.25838+1.86178i,...
-4.2521+1.80605i, -4.25+1.75i, -4.2521+1.69395i,...
-4.25838+1.63822i, -4.2688+1.58311i, -4.28332+1.52893i,...
-4.30184+1.47599i, -4.32427+1.42459i, -4.35048+1.375i,...
```

```
-4.38032+1.32751i, -4.41363+1.28238i, -4.45021+1.23987i,...
-4.48987+1.20021i, -4.53238+1.16363i, -4.57751+1.13032i,...
-4.625+1.10048i, -4.67459+1.07427i, -4.72599+1.05184i,...
-4.77893+1.03332i, -4.83311+1.0188i, -4.88822+1.00838i,...
-4.94395+1.0021i, -5.+1.i, -6+i, inf, 0, inf],...
    [1, 85/84, 43/42, 43/42, 43/42, 43/42, 43/42, 43/42, 43/42,...
 43/42, 43/42, 43/42, 43/42, 43/42, 43/42, 43/42, 43/42, 43/42,...
 43/42, 43/42, 43/42, 43/42, 43/42, 43/42, 43/42, 43/42, 43/42,...
 43/42, 43/42, 43/42, 43/42, 43/42, 43/42, 43/42, 43/42, 43/42,...
 43/42, 43/42, 43/42, 43/42, 43/42, 43/42, 43/42, 85/84, 1, 0,...
 1/2, -1/2]);
```

B.7 Adjacent Capacitors with Plates of $D/2$ Thickness at Different Voltages

The following is a polygonal model code representation of the two adjacent semi-infinite capacitors with plates of $D/2$ thickness at different voltages in the *SC Toolbox* in terms of vertices w_j and interior angles $\alpha_j\pi$. Vertices 0, $-6+2i$, and $6+2i$ were added as guiding points for the *SC Toolbox*'s algorithm.

```
p3bm = polygon([0, inf, 5.+1.i, 4.5+1.5i, 5.+2.i, 6.+2.i, inf,...
    -6.+2.i, -5.+2.i, -4.5+1.5i, -5.+1.i, inf],...
    [1, 0, 5/4, 3/2, 5/4, 1, -1, 1, 5/4, 3/2, 5/4, 0]);
```

APPENDIX C

SC TOOLBOX POLYGONAL MODELS OF THE MUON g-2 QUADRUPOLE

C.1 Nominal Symmetric Case ("SM")

This material pertains to the calculation of the multipole terms the Muon g-2 quadrupole in the SM case (sec. 3.2.4.1). The following *MATLAB* function returns the conformal mapping object `f` for a conformal mapping from a rectangle to the interior of the polygon object `p` representing the 2D geometry.

```
function [p, f] = initquad4
options = scmapopt('Tolerance', 1e-16);
p = polygon([0+0i, 0+5i, -2.35+5i, -2.35+5.3i, 0+5.3i, 0+7i,...
    -5.59792+7i, -5.59792+5.879900042201571253i,...
    -4.493957484174412+4.775937526375983i,...
    -4.634947505275196722+4.634947505275196722i]);
f = rectmap(p, [6, 1, 2, 5], options);
```

C.2 Non-Symmetric Case ("NSM")

This material pertains to the calculation of the multipole terms the Muon g-2 quadrupole in the NSM case (sec. 3.2.4.2). The following *MATLAB* function returns the conformal mapping object `f` for a conformal mapping from a rectangle to the interior of the polygon object `p` representing the 2D geometry.

```
function [p, f] = initquad7
options = scmapopt('Tolerance', 1e-16);
p = polygon([-7-5i, -5.25-5i, -4.125-3.875i, -3.875-4.125i, -5-5.25i,...
 -5-7i, -0.25-7i, -0.25-5.25i, -2.35-5.25i, -2.35-5i, 2.35-5i,...
 2.35-5.25i, 0.25-5.25i, 0.25-7i, 5-7i, 5-5.25i, 3.875-4.125i,...
 4.125-3.875i, 5.25-5i, 7-5i, 7-0.25i, 5.25-0.25i, 5.25-2.35i,...
 5-2.35i, 5+2.35i, 5.25+2.35i, 5.25+0.25i, 7+0.25i, 7+5i, 5.25+5i,...
 4.125+3.875i, 3.875+4.125i, 5+5.25i, 5+7i, 0.25+7i, 0.25+5.25i,...
 2.35+5.25i, 2.35+5i, -2.35+5i, -2.35+5.25i, -0.25+5.25i, -0.25+7i,...
 -5+7i, -5+5.25i, -3.875+4.125i, -4.125+3.875i, -5.25+5i, -7+5i,...
 -7+0.25i, -5.25+0.25i, -5.25+2.35i, -5+2.35i, -5-2.35i, -5.25-2.35i,...
 -5.25-0.25i, -7-0.25i],...
[0.5, 0.75, 1.5, 1.5, 0.75, 0.5, 0.5, 0.5, 1.5, 1.5, 1.5, 1.5,...
 0.5, 0.5, 0.5, 0.75, 1.5, 1.5, 0.75, 0.5, 0.5, 0.5, 1.5, 1.5,...
 1.5, 1.5, 0.5, 0.5, 0.5, 0.75, 1.5, 1.5, 0.75, 0.5, 0.5, 0.5,...
 1.5, 1.5, 1.5, 1.5, 0.5, 0.5, 0.5, 0.75, 1.5, 1.5, 0.75, 0.5,...
 0.5, 0.5, 1.5, 1.5, 1.5, 1.5, 0.5, 0.5]);
f = rectmap(p, [50, 55, 56, 49], options);
```

APPENDIX D

MATLAB CODES FOR MUON g-2 QUADRUPOLE FOURIER MODES CALCULATION

D.1 Nominal Symmetric Case ("SM")

This material pertains to the calculation of the multipole terms the Muon g-2 quadrupole in the SM case (sec. 3.2.4.1). For the conformal mapping object `f` and a reference radius `R`, the following *MATLAB* function calculates the discrete Fourier transform (DFT) d_n and the Fourier modes a_0, a_n, b_n of the electrostatic potential φ using a fast Fourier transform (FFT). The function returns the DFT as the array `dn`, and it returns the Fourier modes as the scalar `a0` and the arrays `an`, `bn`, respectively. The function also returns the set of equidistant grid points discretizing the interval $[0,2\pi]$ as the array `t`, the values of the electrostatic potential φ at these grid points as the array `inseries`, and, for quality assurance purposes, the inverse FFT transform of the forward FFT transform of `inseries` as the array `checkseries`.

```
function [a0,an,bn,cn,t,inseries,checkseries] = fourieran_v4(f,R)
Fs=1001;
T=2*pi/Fs;
if nargin<2
% By default, the reference radius is 4.5 cm
R=4.5;
end
t=0:T:2*pi-T;
z=parameters(f);
function f20=func2(xin,yin)
    % Apply symmetries
    f20=-(1-real(evalinv(f,-abs(xin)+abs(yin)*1i)+...
        z.prevertex(1))/2/z.prevertex(1)).*(abs(yin)>=abs(xin))...
        +(1-real(evalinv(f,-abs(yin)+abs(xin)*1i)+...
```

```
        z.prevertex(1))/2/z.prevertex(1)).*(abs(yin)<abs(xin));
end
inseries=func2(R*cos(t),R*sin(t));
% plot(inseries)
outseries=fft(inseries)/length(t);
a0=2*outseries(1);
an=(outseries(2:1:(Fs/2))+outseries(Fs:-1:(Fs/2+2)));
bn=1i*(outseries(2:1:(Fs/2))-outseries(Fs:-1:(Fs/2+2)));
cn=outseries;
checkseries=transpose(a0/2+cos(transpose(t).*(1:length(an)))*...
    transpose(an)+sin(transpose(t).*(1:length(1:length(bn))))*...
    transpose(bn));
% plot(checkseries)
end
```

D.2 Non-Symmetric Case ("NSM")

This material pertains to the calculation of the multipole terms the Muon g-2 quadrupole in the NSM case (sec. 3.2.4.2). For the conformal mapping object `f` and a reference radius `R`, the following *MATLAB* function calculates the discrete Fourier transform (DFT) d_n and the Fourier modes a_0, a_n, b_n of the electrostatic potential φ using a fast Fourier transform (FFT). The function returns the DFT as the array `dn`, and it returns the Fourier modes as the scalar `a0` and the arrays `an`, `bn`, respectively. The function also returns the set of equidistant grid points discretizing the interval $[0, 2\pi]$ as the array `t`, the values of the electrostatic potential φ at these grid points as the array `inseries`, and, for quality assurance purposes, the inverse FFT transform of the forward FFT transform of `inseries` as the array `checkseries`.

```
function [a0,an,bn,cn,t,inseries,checkseries] = fourieran_v5(f,R)
Fs=1001;
T=2*pi/Fs;
if nargin<2
R=4.5;
end
t=0:T:2*pi-T;
inseries=evalinv(f,R*cos(t)+R*sin(t)*1i);
z=parameters(f);
inseries=real((inseries+z.prevertex(1))/2/z.prevertex(1));
% plot(inseries)
outseries=fft(inseries)/length(t);
a0=2*outseries(1);
an=(outseries(2:1:(Fs/2))+outseries(Fs:-1:(Fs/2+2)));
bn=1i*(outseries(2:1:(Fs/2))-outseries(Fs:-1:(Fs/2+2)));
cn=outseries;
checkseries=transpose(a0/2+cos(transpose(t).*(1:length(an)))*...
    transpose(an)+sin(transpose(t).*(1:length(1:length(bn))))*...
    transpose(bn));
```

APPENDIX E

MATHEMATICA NOTEBOOK FOR MUON g-2 QUADRUPOLE MULTIPOLE TERMS CALCULATION

This material pertains to the calculation of the multipole terms the Muon g-2 quadrupole in the NSM case (sec. 3.2.4.2). For the conformal mapping derivative f in the analytic form, the following *Mathematica* notebook calculates the multipole terms a_0, a_n, b_n (trigonometric form), and c_n (exponential form) of the electrostatic potential φ using the DA method described in sec. 3.2.3.2. The effective reference radius is $R_{\text{ref}} = 1\,\text{cm}$.

```
SetOptions[EvaluationNotebook[], PrintPrecision -> 32];
V = 24 10^3;
\[Alpha] = {0.5‘, 1.5‘, 1.5‘, 1.5‘, 1.5‘, 0.5‘, 0.5‘, 0.5‘, 0.75‘,
    1.5‘, 1.5‘, 0.75‘, 0.5‘, 0.5‘, 0.5‘, 1.5‘, 1.5‘, 1.5‘, 1.5‘,
    0.5‘, 0.5‘, 0.5‘, 0.75‘, 1.5‘, 1.5‘, 0.75‘, 0.5‘, 0.5‘, 0.5‘,
    1.5‘, 1.5‘, 1.5‘, 1.5‘, 0.5‘, 0.5‘, 0.5‘, 0.75‘, 1.5‘, 1.5‘,
    0.75‘, 0.5‘, 0.5‘, 0.5‘, 1.5‘, 1.5‘, 1.5‘, 1.5‘, 0.5‘, 0.5‘,
    0.5‘, 0.75‘, 1.5‘, 1.5‘, 0.75‘, 0.5‘, 0.5‘};
zxm = 1.5707963267948974;
zx = Join[Array[zxm &, 6], Array[-zxm &, 50]];
zy = {0, 4.707844817895257, 6.1123226415433312,
    12.154244427584317, 13.558722251232082, 18.266567069128161,
    18.266567069128161, 13.274266140422423, 12.918629029969397,
    11.576281153221727, 10.948991934692465, 10.675215453051434,
    10.65269458697405,  10.604072782442005, 10.6027963210765,
    10.569241422270052, 10.476916632115067, 9.4896639886962486,
    9.4722605135357565, 9.4665721145687325, 9.4663618193493928,
    9.4586650413212396, 9.4552969316026854, 9.4223471739893334,
    9.3806007759118621, 9.3504541968427848, 9.3474996905761945,
```

```
    9.3408322140301419, 9.3406516641648345, 9.3358001108809781,
    9.3213334522713787, 8.9452336168578217, 8.9307669582482223,
    8.9259154049643676, 8.9257348550990603, 8.9190673785530059,
    8.9161128722864156, 8.8859662932173382, 8.8442198951398652,
    8.8112701375265097, 8.8079020278079554, 8.8002052497797987,
    8.799994954560459,  8.794306555593435,  8.7769030804329375,
    7.7896504370136723, 7.6973256468586344, 7.6637707480521646,
    7.6624942866866617, 7.6138724821545702, 7.5913516160771657,
    7.3175751344358959, 6.6902859159059922, 5.347938039156598,
    4.9923009287031883, 0};
c = -1.1819869586349204`*^7 ;
n = 56;
i0 = 1;
m = 2.1776166063953954 10^-15;
zpoint0 = -9.486801376613026*10^-1 + 9.133283534564583 I;
JacobiSNSer[z_, order_?NumberQ, mpar_: m] :=
  Series[JacobiSN[z + t, mpar], {t, 0, order}];
JacobiCNSer[z_, order_?NumberQ, mpar_: m] :=
  Series[JacobiCN[z + t, mpar], {t, 0, order}];
JacobiDNSer[z_, order_?NumberQ, mpar_: m] :=
  Series[JacobiDN[z + t, mpar], {t, 0, order}];
fprime[z_, order_?NumberQ, mpar_: m] :=
  c JacobiCNSer[z, order, mpar] JacobiDNSer[z, order,
    mpar] Product[(JacobiSNSer[z, order, mpar] -
      JacobiSN[zx[[j]] + I zy[[j]], mpar])^(\[Alpha][[j]] - 1),
        {j, 1, n}];
fprimeexpansion = fprime[zpoint0,25];
fexpansion = Integrate[fprimeexpansion, t];
gexpansion =
```

```
  InverseSeries[fexpansion] +
    (-9.486801376613026 10^-1 + 9.133283534564583 I);
gexpansion = (gexpansion + zxm)/2/zxm;
coefflist = CoefficientList[Normal[gexpansion], t];
a0 = 2 Re[coefflist][[1]];
an = Re[coefflist][[2 ;;]];
bn = Re[coefflist I][[2 ;;]];
Print["a0=", a0];
Print["an=", an];
Print["bn=", bn];
```

APPENDIX F

PYTHON VERSION OF MTERMS

The program `mterms` computes multipole terms of the Muon g-2 collaboration quadrupole with optional geometrical asymmetries and/or mispowered plates specified by the user as plate distance errors and plate voltages. `mterms` was implemented in *COSY INFINITY* and *Python* versions. As *Python* does not have DA functionality, the *Python* version only uses a set of pre-calculated multipole terms. The *COSY INFINITY* version, in addition to pre-calculated multipole terms, can use the *SC Toolbox* and a DA method to calculate multiple terms *de novo*. Below, we describe the *Python* version of the program and its package.

F.1 Package Contents

The program package includes `mterms` as a *Python* 3 script and as *Linux* and *Windows* standalone executables.

1. `readme.txt`: the readme file.
2. `mterms.fox`: *Python* 3 source code.
3. `mterms`: *Linux* version (prepared and tested in *Ubuntu* 16).
4. `mterms.exe`: *Windows* version (prepared and tested in *Windows* 7).
5. `quad.pdf`: Plot of the quadrupole's cross-sectional geometry with default distances.

F.2 Interactive Mode

Run the program and follow the prompts to enter calculation parameters. Resulting multipole terms will be displayed on-screen and written to a file, which is named "`mterms.txt`" by default.

F.3 Command-Line Arguments

`mterms`'s optional command-line arguments are

```
mterms [-i filenameIn] [-o filenameOut] [-h] [/h]
```

where

1. `filenameIn` is an input file with calculation parameters, detailed below;
2. `filenameOut` is an output file with calculation results (default: "`mterms.txt`"); and
3. `-h` or `/h` displays the help screen.

F.4 Input File Format

Optionally, the program accepts an input file specifying the calculation parameters. The file format is ten lines of numbers as follows.

1. Scaling radius R for multipole terms [cm].
2. Nominal plate voltage magnitude φ [V].
3. 1st plate voltage coefficient V_1.
4. 2nd plate voltage coefficient V_2.
5. 3rd plate voltage coefficient V_3.
6. 4th plate voltage coefficient V_4.
7. 1st plate distance error D_1 in centimeters.
8. 2nd plate distance error D_2 in centimeters.
9. 3rd plate distance error D_3 in centimeters.
10. 4th plate distance error D_4 in centimeters.

Plates are numbered counter-clockwise, starting from the left plate. Plate voltages are products of the nominal plate voltage magnitude and the respective plate voltage coefficients. Plate distance errors are errors in distances from the origin to the plates.

F.5 Program Listing

```
#!/usr/bin/python
helpstr = '''
This Python program computes multipole terms of the Muon (g-2) collaboration
quadrupole with optional geometrical asymmetries and/or mispowered plates
specified by the user as plate distance errors and plate voltages.

INTERACTIVE MODE

Run the program and follow the prompts to enter calculation parameters.
Resulting multipole terms will be displayed on-screen and written to a
file, which is "mterms.txt" by default.

OPTIONAL COMMAND-LINE ARGUMENTS

mterms [-i filenameIn] [-o filenameOut] [/h] [-h]
    filenameIn      Input file with calculation parameters, detailed below
    filenameOut     Output file with calculation results
                    (default: "mterms.txt")
    -h or /h        Help information

INPUT FILE FORMAT

Optionally, the program accepts an input file specifying the calculation
parameters. The file format is 10 lines of numbers as follows:

    (1) Scaling radius R for multipole terms [cm]
    (2) Nominal plate voltage magnitude PHI [V]
    (3) 1st plate voltage coefficient V1
    (4) 2nd plate voltage coefficient V2
    (5) 3rd plate voltage coefficient V3
    (6) 4th plate voltage coefficient V4
    (7) 1st plate distance error D1 [cm]
```

```
    (8) 2nd plate distance error D2 [cm]
    (9) 3rd plate distance error D3 [cm]
    (10) 4th plate distance error D4 [cm]

Plates are numbered counter-clockwise, starting from the left plate. Plate
voltages are products of the nominal plate voltage magnitude and the
respective plate voltage coefficients. Plate distance errors are errors in
distances from the origin to the plates.
'''


__author__ = "E. Valetov and M. Berz"
__version__ = "1.0.2"
__maintainer__ = "E. Valetov"
__email__ = "valetove@msu.edu"
__status__ = "Production"

import sys
from math import pi, cos, sin, pow
from pathlib import Path


def isFloat(value):
    try:
        float(value)
        return True
    except:
        return False


def nominalcoeff():
    AN = []
    BN = []
    A0 = 0.39605146671241620026648 89
    AN.append(-0.68856699978729993261161 72E-01)
    AN.append(0.10385499237783464218654 04E-01)
    AN.append(-0.12979133469664615695149 39E-02)
    AN.append(0.11476445475014541855798 83E-03)
    AN.append(-0.86166678655278317793073 80E-05)
    AN.append(-0.80019826615470076255524 43E-07)
```

```
AN.append(0.20824671676850084408847151E-06)
AN.append(-0.56505788542806673088196 36E-07)
AN.append(0.94931378052990516412794 43E-08)
AN.append(-0.11783990556178414789659 39E-08)
AN.append(0.86749953757143685332646 41E-10)
AN.append(0.47440353009637148521696 90E-11)
AN.append(-0.20785937167835847449318 25E-11)
AN.append(0.41481462480543824493008 98E-12)
AN.append(-0.63367710627405382310623 28E-13)
AN.append(0.84906815576397714242400 70E-14)
AN.append(-0.18724217972303592913289 62E-14)
AN.append(0.34264534769384516906330 81E-15)
AN.append(-0.47964633811455258023940 53E-16)
AN.append(0.36730703496684511976840 33E-17)
AN.append(0.60470138938143403239677 24E-18)
AN.append(-0.29357117315031832019707 29E-18)
AN.append(0.70484053455134636745643 02E-19)
AN.append(0.00000000000000000000000000E+00)
AN.append(0.00000000000000000000000000E+00)
BN.append(0.77669481071210756873035 87E-10)
BN.append(0.90845038085748151860098 11E-09)
BN.append(-0.27407097840147603856757 95E-09)
BN.append(0.32117678004156419584109 02E-10)
BN.append(0.12127723883788978348202 23E-11)
BN.append(-0.15660304088215073972820 51E-11)
BN.append(0.43489954703136236488969 82E-12)
BN.append(-0.77244238716242726212351 77E-13)
BN.append(0.72155566366947736445212 80E-14)
BN.append(0.73379865741811126551620 78E-15)
BN.append(-0.50342126590151524464437 16E-15)
BN.append(0.12966361474049731314912 58E-15)
BN.append(-0.20622942579955773500699 50E-16)
BN.append(0.16651977089032312524826 12E-17)
BN.append(0.18021611803975753941105 35E-18)
BN.append(-0.10367467369788289705439 80E-18)
BN.append(0.24831453662985460874714 22E-19)
BN.append(0.00000000000000000000000000E+00)
BN.append(0.00000000000000000000000000E+00)
BN.append(0.00000000000000000000000000E+00)
```

```
    BN.append(0.000000000000000000000000E+00)
    BN.append(0.000000000000000000000000E+00)
    BN.append(0.000000000000000000000000E+00)
    BN.append(0.000000000000000000000000E+00)
    BN.append(0.000000000000000000000000E+00)
    return [A0, AN, BN]


def perturbcoeff1(D1, A0, AN, BN):
    DAN = []
    DBN = []
    DA0 = -0.1444884897365176
    DAN.append(0.024848403415070462)
    DAN.append(-0.0035964314565587524)
    DAN.append(0.0003975167425589697)
    DAN.append(-0.00001902783744042575 7)
    DAN.append(-2.6389842208771386e-6)
    DAN.append(1.3200429446831776e-6)
    DAN.append(-3.336076829265742e-7)
    DAN.append(6.623339431143421e-8)
    DAN.append(-1.041062666661125e-8)
    DAN.append(1.2617566878151998e-9)
    DAN.append(-8.210887578907022e-11)
    DAN.append(-1.1765160099419854e-11)
    DAN.append(5.596411607742279e-12)
    DAN.append(-1.3920004680873368e-12)
    DAN.append(2.6920021350544594e-13)
    DAN.append(-4.29932669278841e-14)
    DAN.append(5.745154992880337e-15)
    DAN.append(-4.983850750279471e-16)
    DAN.append(-2.0577218923151605e-17)
    DAN.append(2.2283046912641997e-17)
    DAN.append(-6.547768415309331e-18)
    DAN.append(1.3884321371806001e-18)
    DAN.append(-2.376780998696832e-19)
    DAN.append(0.)
    DAN.append(0.)
    DBN.append(3.5484213725721443e-6)
    DBN.append(0.00003601436018293621 5)
```

```
    DBN.append(-0.000010881035411630925)
    DBN.append(1.2753318430030797e-6)
    DBN.append(4.819410386204099e-8)
    DBN.append(-6.220211484910526e-8)
    DBN.append(1.727264556597449e-8)
    DBN.append(-3.0676944205138244e-9)
    DBN.append(2.8646167242545424e-10)
    DBN.append(2.9186368392854654e-11)
    DBN.append(-2.000518511207103e-11)
    DBN.append(5.152243236370143e-12)
    DBN.append(-8.194149177053863e-13)
    DBN.append(6.614926637530062e-14)
    DBN.append(7.16612976746983e-15)
    DBN.append(-4.121119580780476e-15)
    DBN.append(9.87224257084664e-16)
    DBN.append(0.)
    DBN.append(0.)
    DBN.append(0.)
    DBN.append(0.)
    DBN.append(0.)
    DBN.append(0.)
    DBN.append(0.)
    DBN.append(0.)
    A0 += D1 * DA0
    AN = list(map(lambda x1, x2: x1 + D1 * x2, AN, DAN))
    BN = list(map(lambda x1, x2: x1 + D1 * x2, BN, DBN))
    return [A0, AN, BN]


def perturbcoeff2(D2, A0, AN, BN):
    DAN = []
    DBN = []
    DA0 = 0.02463158081389416
    DAN.append(-0.0006531514532004401)
    DAN.append(-0.0006519016577943848)
    DAN.append(0.000050219855279518)
    DAN.append(7.308752762074169e-6)
    DAN.append(-1.4847818318198809e-6)
    DAN.append(2.7373194983682978e-8)
```

```
DAN.append(2.575921941229351e-8)
DAN.append(-5.371460655667032e-9)
DAN.append(-2.0187850804427197e-11)
DAN.append(1.3699349709711163e-10)
DAN.append(-1.4367069594248677e-11)
DAN.append(-4.480616938928591e-13)
DAN.append(4.389622545399623e-13)
DAN.append(-6.026734883546398e-14)
DAN.append(-5.3789105734122435e-15)
DAN.append(2.5287025520619607e-15)
DAN.append(-1.440333425607398e-16)
DAN.append(-5.852426774904934e-17)
DAN.append(1.0732657190214119e-17)
DAN.append(-3.371932262158921e-19)
DAN.append(-2.696899131755618e-19)
DAN.append(7.290960456984063e-20)
DAN.append(1.5465718628549282e-22)
DAN.append(0.)
DAN.append(0.)
DBN.append(0.004291141557968972)
DBN.append(-0.00018121709368299234)
DBN.append(-0.00009307793575405624)
DBN.append(0.00001057597023085074 2)
DBN.append(5.56435100630618e-7)
DBN.append(-2.7373406889299316e-7)
DBN.append(3.7729278212684966e-8)
DBN.append(-7.810016388882003e-10)
DBN.append(-6.805548960513614e-10)
DBN.append(7.860964185613506e-11)
DBN.append(-5.91594881675488e-12)
DBN.append(2.3337007215824654e-12)
DBN.append(-6.514714236739883e-13)
DBN.append(1.2302360401338842e-13)
DBN.append(-6.670448858970904e-15)
DBN.append(-4.141239834729562e-15)
DBN.append(1.3793259742916664e-15)
DBN.append(0.)
DBN.append(0.)
DBN.append(0.)
```

```
    DBN.append(0.)
    DBN.append(0.)
    DBN.append(0.)
    DBN.append(0.)
    DBN.append(0.)
    A0 += D2 * DA0
    AN = list(map(lambda x1, x2: x1 + D2 * x2, AN, DAN))
    BN = list(map(lambda x1, x2: x1 + D2 * x2, BN, DBN))
    return [A0, AN, BN]


def perturbcoeff3(D3, A0, AN, BN):
    DAN = []
    DBN = []
    DA0 = 0.009376974330997085
    DAN.append(0.0016466903355588974)
    DAN.append(0.00025673876508554755)
    DAN.append(0.00003463306817851880 5)
    DAN.append(3.699119998049537e-6)
    DAN.append(4.0035108069670176e-7)
    DAN.append(3.1135075378037185e-8)
    DAN.append(-3.4013761560371777e-10)
    DAN.append(-8.329064640595302e-10)
    DAN.append(-1.9072194257427367e-10)
    DAN.append(-3.0553601098349e-11)
    DAN.append(-3.256616872039333e-12)
    DAN.append(-1.1364648285565948e-13)
    DAN.append(2.8168736314034743e-14)
    DAN.append(1.2258346799565113e-14)
    DAN.append(2.6975760287450926e-15)
    DAN.append(5.295885854313568e-16)
    DAN.append(9.992654208203225e-17)
    DAN.append(1.483293712681986e-17)
    DAN.append(1.590520323241747e-18)
    DAN.append(-2.018229056940692e-20)
    DAN.append(-5.522461977746537e-20)
    DAN.append(-1.7833523207981785e-20)
    DAN.append(-3.607909453318381e-21)
    DAN.append(0.)
```

```
    DAN.append(0.)
    DBN.append(2.6012561396628937e-6)
    DBN.append(-3.264427228831575e-7)
    DBN.append(1.6219024479853696e-8)
    DBN.append(-9.301485458960886e-11)
    DBN.append(4.875609189517354e-11)
    DBN.append(-3.020687618033502e-11)
    DBN.append(-2.008207049699983e-12)
    DBN.append(8.618373852593456e-13)
    DBN.append(-5.025368408773574e-13)
    DBN.append(1.7929831598457703e-13)
    DBN.append(-4.6486252361772815e-14)
    DBN.append(9.821321480592008e-15)
    DBN.append(-1.6317932526659849e-15)
    DBN.append(8.621596651502531e-17)
    DBN.append(3.3157898412483947e-17)
    DBN.append(-1.2810764392090319e-17)
    DBN.append(3.821237355640557e-18)
    DBN.append(0.)
    DBN.append(0.)
    DBN.append(0.)
    DBN.append(0.)
    DBN.append(0.)
    DBN.append(0.)
    DBN.append(0.)
    DBN.append(0.)
    A0 += D3 * DA0
    AN = list(map(lambda x1, x2: x1 + D3 * x2, AN, DAN))
    BN = list(map(lambda x1, x2: x1 + D3 * x2, BN, DBN))
    return [A0, AN, BN]


def perturbcoeff4(D4, A0, AN, BN):
    DAN = []
    DBN = []
    DA0 = 0.02463156389964638
    DAN.append(-0.0006572106163680683)
    DAN.append(-0.0006509517704236568)
    DAN.append(0.00050080414813327 58)
```

```
DAN.append(7.311960481097025e-6)
DAN.append(-1.4832042664218915e-6)
DAN.append(2.686993667956254e-8)
DAN.append(2.582815475252795e-8)
DAN.append(-5.375006788041004e-9)
DAN.append(-2.216645711303911e-11)
DAN.append(1.3763870154246403e-10)
DAN.append(-1.447182447679387e-11)
DAN.append(-4.4261541328000695e-13)
DAN.append(4.424264496060864e-13)
DAN.append(-6.138744911938371e-14)
DAN.append(-5.159127712706641e-15)
DAN.append(2.5039160264365163e-15)
DAN.append(-1.4456449599913898e-16)
DAN.append(-5.772285926076793e-17)
DAN.append(1.0509489396965041e-17)
DAN.append(-2.984829601057637e-19)
DAN.append(-2.7466074284149757e-19)
DAN.append(7.314845072502751e-20)
DAN.append(1.5397501179808952e-22)
DAN.append(0.)
DAN.append(0.)
DBN.append(-0.0042923438435735)
DBN.append(0.00021791378621024795)
DBN.append(0.00008211499671272027)
DBN.append(-9.294389673533813e-6)
DBN.append(-5.079774526104112e-7)
DBN.append(2.1139313062627526e-7)
DBN.append(-2.0388970595219676e-8)
DBN.append(-2.29940758712421e-9)
DBN.append(9.6889886221242e-10)
DBN.append(-4.9679879227888535e-11)
DBN.append(-1.406671943084948e-11)
DBN.append(2.8163816961828453e-12)
DBN.append(-1.677183835725091e-13)
DBN.append(-5.678868964822446e-14)
DBN.append(1.3795447987588035e-14)
DBN.append(3.2602077248549906e-17)
DBN.append(-3.9333149423065704e-16)
```

```
    DBN.append(0.)
    DBN.append(0.)
    DBN.append(0.)
    DBN.append(0.)
    DBN.append(0.)
    DBN.append(0.)
    DBN.append(0.)
    DBN.append(0.)
    A0 += D4 * DA0
    AN = list(map(lambda x1, x2: x1 + D4 * x2, AN, DAN))
    BN = list(map(lambda x1, x2: x1 + D4 * x2, BN, DBN))
    return [A0, AN, BN]


def calcSet(D1, D2, D3, D4):
    [A0, AN, BN] = nominalcoeff()
    [A0, AN, BN] = perturbcoeff1(D1, A0, AN, BN)
    [A0, AN, BN] = perturbcoeff2(D2, A0, AN, BN)
    [A0, AN, BN] = perturbcoeff3(D3, A0, AN, BN)
    [A0, AN, BN] = perturbcoeff4(D4, A0, AN, BN)
    return [A0, AN, BN]


def rotateFourierAN(AN, BN, PHI, J):
    return AN * cos(J * PHI) - BN * sin(J * PHI)


def rotateFourierBN(AN, BN, PHI, J):
    return AN * sin(J * PHI) + BN * cos(J * PHI)


def superImposeForierA0(A01, A02, A03, A04, V1, V2, V3, V4):
    return V1 * A01 + V2 * A02 + V3 * A03 + V4 * A04


def superImposeFourierAN(AN1, BN1, AN2, BN2, AN3, BN3, AN4, BN4, V1, V2, V3,
                         V4, J):
    return V1 * AN1 + V2 * rotateFourierAN(AN2, BN2, pi / 2, J) + \
           V3 * rotateFourierAN(AN3, BN3, pi, J) + \
```

```
            V4 * rotateFourierAN(AN4, BN4, 3 * pi / 2, J)


def superImposeFourierBN(AN1, BN1, AN2, BN2, AN3, BN3, AN4, BN4, V1, V2, V3,
                         V4, J):
    return V1 * BN1 + V2 * rotateFourierBN(AN2, BN2, pi / 2, J) + \
           V3 * rotateFourierBN(AN3, BN3, pi, J) + \
           V4 * rotateFourierBN(AN4, BN4, 3 * pi / 2, J)


def multipoleTerms(R, Rref, fileOutName):
    outputArray = []
    outputString = '\nCALCULATION RESULTS\n'
    print(outputString)
    outputArray.append(outputString)
    outputString = 'Multipole terms, scaled to R = ' + str(R) + ' cm'
    print(outputString)
    outputArray.append(outputString)
    outputString = 'Nominal plate voltage PHI = ' + str(V) + ' V'
    print(outputString)
    outputArray.append(outputString)
    outputString = 'Plate voltage coefficients:'
    print(outputString)
    outputArray.append(outputString)
    outputString = 'Plate 1: V1 = ' + str(V1)
    print(outputString)
    outputArray.append(outputString)
    outputString = 'Plate 2: V2 = ' + str(V2)
    print(outputString)
    outputArray.append(outputString)
    outputString = 'Plate 3: V3 = ' + str(V3)
    print(outputString)
    outputArray.append(outputString)
    outputString = 'Plate 4: V4 = ' + str(V4)
    print(outputString)
    outputArray.append(outputString)
    outputString = 'Plate distance errors:'
    print(outputString)
    outputArray.append(outputString)
```

```
outputString = 'Plate 1: D1 = ' + str(D10) + ' cm'
print(outputString)
outputArray.append(outputString)
outputString = 'Plate 2: D2 = ' + str(D20) + ' cm'
print(outputString)
outputArray.append(outputString)
outputString = 'Plate 3: D3 = ' + str(D30) + ' cm'
print(outputString)
outputArray.append(outputString)
outputString = 'Plate 4: D4 = ' + str(D40) + ' cm'
print(outputString)
outputArray.append(outputString)
outputString = ''
print(outputString)
outputArray.append(outputString)
outputString = 'A0 = ' + str(
    V * superImposeForierA0(A01, A02, A03, A04, V1, V2, V3, V4))
print(outputString)
outputArray.append(outputString)
outputString = '\nAN = '
print(outputString)
outputArray.append(outputString)
for J, item in enumerate(AN1):
    outputString = str(
        V * pow((R / Rref), (J + 1)) * superImposeFourierAN(AN1[J], BN1[J],
                                                            AN2[J], BN2[J],
                                                            AN3[J], BN3[J],
                                                            AN4[J], BN4[J],
                                                            V1, V2, V3, V4,
                                                            J + 1))
    print(outputString)
    outputArray.append(outputString)
outputString = '\nBN = '
print(outputString)
outputArray.append(outputString)
for J, item in enumerate(AN1):
    outputString = str(
        V * pow((R / Rref), (J + 1)) * superImposeFourierBN(AN1[J], BN1[J],
                                                            AN2[J], BN2[J],
```

```
                                                        AN3[J], BN3[J],
                                                        AN4[J], BN4[J],
                                                        V1, V2, V3, V4,
                                                        J + 1))
            print(outputString)
            outputArray.append(outputString)
        outputArray = [i + "\n" for i in outputArray]
        with open(fileOutName, 'w+') as file_out:
            for item in outputArray:
                file_out.write("%s" % item)


V = 24000
R = 4.5
Rref = 1.0
(V1, V2, V3, V4) = 1, -1, 1, -1
(D10, D20, D30, D40) = 0, 0, 0, 0
fileOutName = 'mterms.txt'
useInputFile = False
inputArray = []

print('-------------------------------------------------------------')
print('    Muon (g-2) Collaboration High Voltage Quadrupole       ')
print('         Multipole Terms Calculation ("mterms")            ')
print('                  E. Valetov & M. Berz                     ')
print('                Muon (g-2) Collaboration                   ')
print('                   Created 16-Apr-2017                     ')
print('                 Email: valetove@msu.edu                   ')
print('-------------------------------------------------------------\n')

if len(sys.argv) > 1:
    sysArgv = sys.argv[1:]
    while len(sysArgv) > 0:
        sysArgvNext = sysArgv.pop(0).lower()
        if (sysArgvNext == '-h') or (sysArgvNext == '/h'):
            print(helpstr)
            sys.exit()
        elif sysArgvNext == '-o':
            if len(sysArgv) == 0:
```

```
            print("[ERROR] Output file name expected, not found.")
            sys.exit()
        fileOutName = sysArgv.pop(0)
    elif sysArgvNext == '-i':
        if len(sysArgv) == 0:
            print("[ERROR] Input file name expected, not found.")
            sys.exit()
        fileInName = sysArgv.pop(0)
        if not Path(fileInName).is_file():
            print("[ERROR] Input file '" + fileInName + "' doesn't exist.")
            sys.exit()
        useInputFile = True
        with open(fileInName) as fileIn:
            for line in fileIn:
                if not (isFloat(line)):
                    print("[ERROR] NaN in input file.")
                    sys.exit()
                inputArray.append(float(line))
        if len(inputArray) != 10:
            print(
                "[ERROR] Input file length is not 10 lines of numbers.")
            print("\n[INFO] Help screen.")
            print(helpstr)
            sys.exit()
        R = inputArray.pop(0)
        V = inputArray.pop(0)
        V1 = inputArray.pop(0)
        V2 = inputArray.pop(0)
        V3 = inputArray.pop(0)
        V4 = inputArray.pop(0)
        D10 = inputArray.pop(0)
        D20 = inputArray.pop(0)
        D30 = inputArray.pop(0)
        D40 = inputArray.pop(0)
    else:
        print(
            "[ERROR] Invalid command-line parameter '" +
            sysArgvNext + "'.")
        print("\n[INFO] Help screen.")
```

```
            print(helpstr)
            sys.exit()

if not (useInputFile):
    print(
        'Default errors in distances from the origin to the plates' +
        ' are 0, 0, 0, 0 [cm].')
    print(
        'Plates are numbered counter-clockwise, starting from the left plate.')
    print("Enter 'Y' to change the values of the plate distance errors.")
    print("Enter 'N' or <CR> to keep the default value.")
    userInput = '-'
    while not (userInput[:1].lower() in ['y', 'n', '']):
        userInput = input('[y/N] > ')
    if userInput[:1].lower() == 'y':
        userInput = ''
        print('Enter the 1st plate distance error D1.')
        while not isFloat(userInput):
            userInput = input('> ')
        D10 = float(userInput)
        userInput = ''
        print('Enter the 2nd plate distance error D2.')
        while not isFloat(userInput):
            userInput = input('> ')
        D20 = float(userInput)
        userInput = ''
        print('Enter the 3rd plate distance error D3.')
        while not isFloat(userInput):
            userInput = input('> ')
        D30 = float(userInput)
        userInput = ''
        print('Enter the 4th plate distance error D4.')
        while not isFloat(userInput):
            userInput = input('> ')
        D40 = float(userInput)
    userInput = '-'
    print('Default nominal plate voltage magnitude is PHI = 24000 V.')
    print('Enter a new value of PHI.')
    print("To keep the default value of PHI, enter 'D' or <CR>.")
```

```
while not ((userInput[:1].lower() in ['d', '']) or (isFloat(userInput))):
    userInput = input('[D/(Real number)] > ')
if isFloat(userInput):
    V = float(userInput)
userInput = '-'
print('Default plate voltage coefficients are 1, -1, 1, -1.')
print(
    'Plate voltages are products of the nominal plate voltage magnitude')
print('and respective plate voltage coefficients.')
print(
    'Plates are numbered counter-clockwise, starting from the left plate.')
print("Enter 'Y' to change the values of the plate voltage coefficients.")
print("Enter 'N' or <CR> to keep the default value.")
while not (userInput[:1].lower() in ['y', 'n', '']):
    userInput = input('[y/N] > ')
if userInput[:1].lower() == 'y':
    userInput = ''
    print('Enter the 1st plate voltage coefficient V1.')
    while not isFloat(userInput):
        userInput = input('> ')
    V1 = float(userInput)
    userInput = ''
    print('Enter the 2nd plate voltage coefficient V2.')
    while not isFloat(userInput):
        userInput = input('> ')
    V2 = float(userInput)
    userInput = ''
    print('Enter the 3rd plate voltage coefficient V3.')
    while not isFloat(userInput):
        userInput = input('> ')
    V3 = float(userInput)
    userInput = ''
    print('Enter the 4th plate voltage coefficient V4.')
    while not isFloat(userInput):
        userInput = input('> ')
    V4 = float(userInput)
userInput = '-'
print('Default scaling of multipole coefficients is to R = 4.5 cm.')
print('Enter a new value of R.')
```

```
    print("To keep the default value of R, enter or 'D' or <CR>.")
    while not ((userInput[:1].lower() in ['d', '']) or (isFloat(userInput))):
        userInput = input('[D/(Real number)] > ')
    if isFloat(userInput):
        R = float(userInput)

[A01, AN1, BN1] = calcSet(D10, D20, D30, D40)
[A02, AN2, BN2] = calcSet(D20, D30, D40, D10)
[A03, AN3, BN3] = calcSet(D30, D40, D10, D20)
[A04, AN4, BN4] = calcSet(D40, D10, D20, D30)

multipoleTerms(R, Rref, fileOutName)
```

APPENDIX G

STEP FILE GENERATOR

STEP File Generator (`stepfg`) converts a list of polygons in the x–y plane specified by vertices into a STEP file[1] containing a 3D part obtained by extrusion of the interior regions of these polygons along the z axis.

G.1 Package Contents

The program package includes *STEP File Generator* as a *Python* script and as *Linux* and *Windows* standalone executables.

1. `readme.txt`: the readme file.
2. `stepfg.py`: *Python* source code.
3. `stepfg`: *Linux* version (prepared and tested in *Ubuntu* 16).
4. `stepfg.exe`: *Windows* version (prepared and tested in *Windows* 7).
5. `part_geometry.txt`: input file containing the representation of the Muon g-2 collaboration quadrupole detailed in sec. 3.3.4.

G.2 Command-Line Arguments

STEP File Generator's command-line arguments are

```
stepfg [filename_in [filename_out]] [-h] [/h]
```

where

1. `filename_in` is the input file containing 2D geometry data (default: "`part_geometry.txt`");

[1] For syntax and rules for STEP files, please refer to ISO 10303-242 [55].

2. `filename_out` is the output STEP file with resulting 3D part (default: "`part_out.stp`"); and

3. `−h` or `/h` displays the help screen.

G.3 Input File Format

The input file format is three parameters as follows:

```
[First_argument,
Second_argument,
Third_argument]
```

`First_argument`: List of polygon specifications $[p_1, p_2, \ldots, p_n]$. Each polygon specification is a sequential list $[v_1, v_2, \ldots, v_m]$ of the polygon's vertices in the x–y plane. Each vertex is specified as a list $[x, y]$ or $[x, y, 0]$.

`Second_argument`: z coordinate interval $[z_1, z_2]$ that the resulting 3D part should span.

`Third_argument`: Geometric proportionality coefficient. The output unit of length in the STEP file is mm, so use 10 if the 2D geometry is specified in cm.

G.4 Program Listing

The following is a program listing of *STEP File Generator* as a *Python* 3 script.

```
#!/usr/bin/python
helpstr = '''
This program converts a list of polygons in the x-y plane specified by
vertices into a STEP file containing a 3D part obtained by extrusion of the
interior regions of these polygons along the z axis.

stepfg [filename_in [filename_out]] [-h] [/h]
    filename_in     Input file containing 2D geometry data
                    (default: "part_geometry.txt")
    filename_out    Output STEP file with resulting 3D part
                    (default: "part_out.stp")
    -h or /h        This information
```

```
The input file format is three parameters as follows:

[First_argument,
Second_argument,
Third_argument]

First_argument: List of polygon specifications [pol1,pol2,...,poln]. Each
    polygon specification is a sequential list [vert1,vert2,...,vertm] of the
    polygon's vertices in the x-y plane. Each vertex is specified as a list
    [x,y] or [x,y,0].
Second_argument: z coordinate interval [z1, z2] that the resulting 3D part
    should span.
Third_argument: Geometric proportionality coefficient. The output unit of
    length in the STEP file is mm, so use 10 if the 2D geometry is specified
    in cm.

A sample input file, "part_geometry.txt" containing a representation of the
Muon g-2 collaboration quadrupole, is supplied with this program.
'''

__author__ = "E. Valetov and M. Berz"
__version__ = "1.0.1"
__maintainer__ = "E. Valetov"
__email__ = "valetove@msu.edu"
__status__ = "Production"

import re
import math
import operator
import ast
import sys
import datetime
from numbers import Number
from pathlib import Path


def line_index(line):
    search_result = re.search('#(.+?)=', line)
```

```
    return 0 if search_result is None else int(search_result.group(1))


def rotate(list_in, x):
    return list_in[-x:] + list_in[:-x]


def item_exists_q(string_in):
    if not work_array:
        return False
    search_result = [i for i, item in enumerate(work_array) if
                     item.endswith(string_in)]
    return False if not search_result else True


def existing_item_ln(string_in):
    if not work_array:
        return False
    if not item_exists_q(string_in):
        return False
    search_result = [item for item in work_array if item.endswith(string_in)]
    return line_index(search_result[0])


def new_item(string_in):
    global current_index
    global work_array
    if item_exists_q(string_in):
        return_index = existing_item_ln(string_in)
    else:
        work_array.append('#' + str(current_index) + '=' + string_in)
        return_index = current_index
        current_index += 1
    return return_index


def to_coord(clist):
    if len(clist) != 3:
        print('to_coord: Error. Coordinates not 3D.')
```

```
        sys.exit()
    return str(clist[0]) + ',' + str(clist[1]) + ',' + str(clist[2])


def to_step_list(slist):
    if not isinstance(slist, list):
        slist1 = '#' + str(slist)
    else:
        slist1 = ''
        for j in range(0, len(slist) - 1):
            slist1 += '#' + str(slist[j]) + ','
        slist1 += '#' + str(slist[-1])
    return slist1


def fort_bool(bool_in):
    return '.T.' if (bool_in == True) or (bool_in == '.T.') else '.F.'


def normalize(vector_in):
    if len(vector_in) != 3:
        print('normalize: Error. Coordinates not 3D.')
        sys.exit()
    magnitude = math.sqrt(sum([i ** 2 for i in vector_in]))
    return [x / magnitude for x in vector_in]


def cross_product(x, y):
    return [-x[2] * y[1] + x[1] * y[2], x[2] * y[0] - x[0] * y[2],
            -x[1] * y[0] + x[0] * y[1]]


def point(coord_in):
    return new_item("CARTESIAN_POINT('',(" + to_coord(coord_in) + ")) ;\n")


def line(origin, direction):
    coord_ln = new_item(
        "CARTESIAN_POINT('Origin Line',(" + to_coord(origin) + ")) ;\n")
```

```
    dir_ln = new_item("DIRECTION('Vector Direction',(" + to_coord(
        normalize(direction)) + ")) ;\n")
    vec_ln = new_item("VECTOR('Line Direction',#" + str(dir_ln) + ",1.) ;\n")
    return new_item(
        "LINE('Line',#" + str(coord_ln) + ",#" + str(vec_ln) + ") ;\n")


def vertex(coord_in):
    coord_ln = new_item(
        "CARTESIAN_POINT('Vertex',(" + to_coord(coord_in) + ")) ;\n")
    return new_item("VERTEX_POINT('',#" + str(coord_ln) + ") ;\n")


def edge_curve(vertex1_ln, vertex2_ln, line_coord_ln, same_sense=True):
    return new_item(
        "EDGE_CURVE('',#" + str(vertex1_ln) + ",#" + str(
            vertex2_ln) + ",#" + str(line_coord_ln) + "," + fort_bool(
            same_sense) + ") ;\n")


def edge_curve_0(vertex1, vertex2, same_sense=True):
    return edge_curve(vertex(vertex1), vertex(vertex2), line(
        [x / 2 for x in list(map(operator.add, vertex1, vertex2))],
        list(map(operator.sub, vertex2, vertex1))),
                      fort_bool(same_sense))


def oriented_edge(edge_curve_ln, same_sense=True):
    return new_item(
        "ORIENTED_EDGE('',*,*,#" + str(edge_curve_ln) + "," + fort_bool(
            same_sense) + ") ;\n")


def edge_loop(lines):
    return new_item("EDGE_LOOP('',(" + to_step_list(lines) + ")) ;\n")


def edge_loop_0(vertices):
    return edge_loop(list(
```

```
        map(lambda x1, x2: oriented_edge(edge_curve_0(x1, x2)), vertices,
            rotate(vertices, -1))))


def face_outer_bound(edge_loop_ln, same_sense: True):
    return new_item(
        "FACE_OUTER_BOUND('',#" + str(edge_loop_ln) + "," + fort_bool(
            same_sense) + ") ;\n")


def edge_loop_1(vertices, same_sense: True):
    return face_outer_bound(edge_loop_0(vertices), same_sense)


def axis2_placement_3d(origin_coord, direction1, direction2):
    origin_ln = new_item("CARTESIAN_POINT('Axis2P3D Location',(" + to_coord(
        origin_coord) + ")) ;\n")
    direction1_ln = new_item(
        "DIRECTION('Axis2P3D ZDirection',(" + to_coord(direction1) + ")) ;\n")
    direction2_ln = new_item(
        "DIRECTION('Axis2P3D XDirection',(" + to_coord(direction2) + ")) ;\n")
    return new_item(
        "AXIS2_PLACEMENT_3D('Plane Axis2P3D',#" + str(origin_ln) + ",#" + str(
            direction1_ln) + ",#" + str(
            direction2_ln) + ") ;\n")


def plane(axis2_placement_3d_ln):
    return new_item("PLANE('',#" + str(axis2_placement_3d_ln) + ") ;\n")


def plane_0(origin_coord, direction1, direction2):
    return plane(axis2_placement_3d(origin_coord, direction1, direction2))


def advanced_face(face_outer_bound_ln, plane_ln, same_sense_plane=True):
    return new_item(
        "ADVANCED_FACE('PartBody',(" + to_step_list(
            face_outer_bound_ln) + "),#" + str(plane_ln) + "," + fort_bool(
```

```
            same_sense_plane) + ") ;\n")


def advanced_face_0(vertices, zaxis, same_sense_1=True, same_sense_2=True):
    if list(map(operator.add, normalize(zaxis), normalize(
            cross_product(list(map(operator.sub, vertices[2], vertices[1])),
                          list(map(operator.sub, vertices[2],
                                   vertices[0]))))) == [0, 0, 0]:
        af_ln = advanced_face(edge_loop_1(vertices, same_sense_1), plane(
            axis2_placement_3d(vertices[0], normalize(zaxis),
                               normalize(list(map(operator.sub, vertices[1],
                                                  vertices[0])))),
                               same_sense_2)
    else:
        af_ln = advanced_face(
            edge_loop_1(list(reversed(vertices)), same_sense_1), plane(
                axis2_placement_3d(vertices[0], normalize(zaxis), normalize(
                    list(map(operator.sub, list(reversed(vertices))[1],
                             list(reversed(vertices))[0])))), same_sense_2)
    return af_ln


def advanced_face_1(vertices, same_sense_1=True, same_sense_2=True):
    return advanced_face(edge_loop_1(vertices, same_sense_1),
                         plane(axis2_placement_3d(vertices[0], normalize(
                             cross_product(list(
                                 map(operator.sub, vertices[2], vertices[1])),
                                 list(map(operator.sub, vertices[1],
                                          vertices[0]))), normalize(
                             list(map(operator.sub, vertices[1],
                                      vertices[0])))), same_sense_1)


def closed_shell(advanced_face_ln_list):
    return new_item("CLOSED_SHELL('Closed Shell',(" + to_step_list(
        advanced_face_ln_list) + ")) ;\n")


def manifold_solid_brep(closed_shell_ln):
```

```
    global part_body_index
    msb = new_item(
        "MANIFOLD_SOLID_BREP('PartBody." + str(part_body_index) + "',#" + str(
            closed_shell_ln) + ") ;\n")
    part_body_index += 1
    return msb


def advanced_brep_shape_representation(manifold_solid_brep_list, init_ln=45):
    return new_item(
        "ADVANCED_BREP_SHAPE_REPRESENTATION('NONE',(" + to_step_list(
            manifold_solid_brep_list) + "),#" + str(
            init_ln) + ") ;\n")


def shape_representation_relationship(advanced_brep_shape_representation_ln,
                                      shape_representation_ln=48):
    return new_item("SHAPE_REPRESENTATION_RELATIONSHIP(' ',' ',#" + str(
        shape_representation_ln) + ",#" + str(
        advanced_brep_shape_representation_ln) + ") ;\n")


def zface(vertex1, vertex2, geom_depth_list):
    z_neg = geom_depth_list[0]
    z_pos = geom_depth_list[1]
    return advanced_face_0(
        [list(map(operator.add, vertex1, [0, 0, z_neg])),
         list(map(operator.add, vertex2, [0, 0, z_neg])),
         list(map(operator.add, vertex2, [0, 0, z_pos])),
         list(map(operator.add, vertex1, [0, 0, z_pos]))],
        normalize(cross_product(list(map(operator.sub, vertex2, vertex1)),
                                [0, 0, -(z_pos - z_neg)])))


def xyface(vertex_list, depth, zdir):
    return advanced_face_0(list(
        map(lambda x: list(map(operator.add, x, [0, 0, depth])), vertex_list)),
        zdir)
```

```
def af2d3d(vertex_list, geom_depth_list):
    taflist = []
    z_neg = geom_depth_list[0]
    z_pos = geom_depth_list[1]
    taflist.append(xyface(vertex_list, z_pos, [0, 0, 1]))
    taflist.append(xyface(vertex_list, z_neg, [0, 0, -1]))
    taflist += list(
        map(lambda x1, x2: zface(x1, x2, geom_depth_list), vertex_list,
            rotate(vertex_list, -1)))
    return taflist


def af_list_2_assembly(af_list):
    return shape_representation_relationship(
        manifold_solid_brep(closed_shell(af_list)))


def af_list_2_part(af_list):
    return manifold_solid_brep(closed_shell(af_list))


def part_2_assembly(part_list):
    return shape_representation_relationship(
        advanced_brep_shape_representation(part_list))


def generate_part(vert_list, geom_depth, clockwise_p=True):
    return af_list_2_part(
        af2d3d(vert_list, geom_depth)) if clockwise_p else af_list_2_part(
        af2d3d(reversed(vert_list), geom_depth))


def convert_3d(element_in):
    if (isinstance(element_in, list)) and (len(element_in)) == 2 and (
            isinstance(element_in[0], Number)) and (
            isinstance(element_in[1], Number)):
        return [element_in[0], element_in[1], 0]
    else:
```

```
        return element_in


def convert_to_clockwise(part_list):
    pol_sum = sum(list(
        map(lambda x1, x2: (x2[0] - x1[0]) * (x2[1] + x1[1]), part_list,
            rotate(part_list, 1))))
    if pol_sum == 0:
        print(
            "[FAILED]\nconvert_to_clockwise: Error. Polygon is" +
            " neither clockwise nor counter-clockwise.")
        sys.exit()
    elif pol_sum > 0:
        return reversed(part_list)
    else:
        return part_list


def generate_assembly(list_vert_list, geom_depth_list, p_coeff=1):
    print("Generating assembly... ", end="")
    if not isinstance(p_coeff, Number):
        print(
            "[FAILED]\ngenerate_assembly: Error. NaN supplied for" +
            " proportionality coefficient.")
        sys.exit()
    if p_coeff == 0:
        print(
            "[FAILED]\ngenerate_assembly: Error. Zero supplied as" +
            " the proportionality coefficient.")
        sys.exit()
    if not isinstance(geom_depth_list, list):
        print(
            "[FAILED]\ngenerate_assembly: Error. z-coordinate interval" +
            " [z1, z2] expected, scalar supplied.")
        sys.exit()
    if not geom_depth_list:
        print(
            "[FAILED]\ngenerate_assembly: Error. z-coordinate interval" +
            " [z1, z2] expected, empty list supplied.")
```

```
    sys.exit()
for depth_element in geom_depth_list:
    if not isinstance(depth_element, Number):
        print(
            "[FAILED]\ngenerate_assembly: Error. NaN found in the" +
            " z-coodinate interval.")
        sys.exit()
if geom_depth_list[0] == geom_depth_list[1]:
    print(
        "[FAILED]\ngenerate_assembly: Error. z2 must be different" +
        " from z1 in the z-coordinate interval [z1, z2].")
    sys.exit()
if geom_depth_list[0] > geom_depth_list[1]:
    tvar = geom_depth_list[0]
    geom_depth_list[0] = geom_depth_list[1]
    geom_depth_list[1] = tvar
if not isinstance(list_vert_list, list):
    print(
        "[FAILED]\ngenerate_assembly: Error. List of vertices lists" +
        " expected, scalar supplied.")
    sys.exit()
if not list_vert_list:
    print(
        "[FAILED]\ngenerate_assembly: Error. Empty list of vertices" +
        " lists.")
    sys.exit()
for part_element in list_vert_list:
    if not part_element:
        print(
            "[FAILED]\ngenerate_assembly: Error. Empty list of" +
            " vertices.")
        sys.exit()
    if not isinstance(part_element, list):
        print(
            "[FAILED]\ngenerate_assembly: Error. List of vertices" +
            " expected, scalar supplied")
        sys.exit()
    for vertex_element in part_element:
        if not vertex_element:
```

```
            print(
                "[FAILED]\ngenerate_assembly: Error. Empty list of" +
                "vertex coordinates.")
            sys.exit()
        if not isinstance(part_element, list):
            print(
                "[FAILED]\ngenerate_assembly: Error. List of vertex" +
                " coordinates expected, scalar supplied")
            sys.exit()
        if (len(vertex_element) < 2) or (len(vertex_element) > 3):
            print(
                "[FAILED]\ngenerate_assembly: Error. Number of vertex" +
                " coordinates should be 2 or 3, " + str(
                    len(vertex_element)) + " coordinates supplied.")
            sys.exit()
        for coordinate_element in vertex_element:
            if not isinstance(coordinate_element, Number):
                print(
                    "[FAILED]\ngenerate_assembly: Error. NaN is supplied" +
                    " for a vertex coordinate.")
                sys.exit()
list_vert_list = [
    [convert_3d(vertex_element) for vertex_element in part_element] for
    part_element in
    list_vert_list]
list_vert_list = [convert_to_clockwise(x) for x in list_vert_list]
list_vert_list = [
    [[p_coeff * 1.0 * coordinate_element for coordinate_element in
      vertex_element] for vertex_element in
     part_element] for part_element in list_vert_list]
geom_depth_list = [p_coeff * 1.0 * i for i in geom_depth_list]
part_list = [generate_part(x, geom_depth_list) for x in list_vert_list]
part_2_assembly(part_list)
resulting_array = file_array[:index1] + work_array + file_array[
                                                     index1 + 1:]
print("[DONE]")
print("Writing STEP file... ", end="")
with open(file_out_name, 'w+') as file_out:
    for item in resulting_array:
```

```
            file_out.write("%s" % item)
    print("[DONE]")


file_in2_name = 'part_geometry.txt'
file_out_name = 'part_out.stp'

print("-------------------------------------------------------------")
print("           STEP File Generator ('stepfg')                    ")
print("              E. Valetov and M. Berz                         ")
print("             Michigan State University                       ")
print("                Created 03-Feb-2017                          ")
print("              Email: valetove@msu.edu                        ")
print("-------------------------------------------------------------")

if (len(sys.argv) > 1):
    if ((str(sys.argv[1]) == '-h') or (str(sys.argv[1]) == '/h')):
        print(helpstr)
        sys.exit()
    file_in2_name = str(sys.argv[1])
    if (len(sys.argv) > 2):
        if ((str(sys.argv[2]) == '-h') or (str(sys.argv[2]) == '/h')):
            print(helpstr)
            sys.exit()
        file_out_name = str(sys.argv[2])

print("Use command-line option -h or /h for help.\n")

print("Reading 2D geometry file " + file_in2_name + "... ", end="")
if not Path(file_in2_name).is_file():
    print(
        "[FAILED]\nError. 2D geometry file " + file_in2_name +
        " doesn't exist.")
    sys.exit()
with open(file_in2_name, 'r') as file_in:
    data = ast.literal_eval(file_in.read())
if not isinstance(data, list):
    print("[FAILED]\nError. Input data is not a list.")
    sys.exit()
```

```
if len(data) != 3:
    print(
        "[FAILED]\nError. The top-level length of the input data list is not" +
        " 3.")
    sys.exit()
in_array = data[0]
in_depth = data[1]
in_coeff = data[2]
print("[DONE]")

print("Initializing STEP file data... ", end="")
d = datetime.datetime.now()
file_array = ["ISO-10303-21;",
              "HEADER;",
              "FILE_DESCRIPTION(('none'),'2;1');",
              "",
              "FILE_NAME('" + file_out_name + "','none',('none'),('none')," +
              "'none','none','none');",
              "",
              "FILE_SCHEMA(('CONFIG_CONTROL_DESIGN'));",
              "",
              "ENDSEC;",
              "DATA;",
              "#1=APPLICATION_CONTEXT('configuration controlled 3D design of" +
              " mechanical parts and assemblies') ;",
              "#2=MECHANICAL_CONTEXT(' ',#1,'mechanical') ;",
              "#3=DESIGN_CONTEXT(' ',#1,'design') ;",
              "#4=APPLICATION_PROTOCOL_DEFINITION('international standard'," +
              "'config_control_design',1994,#1) ;",
              "#5=PRODUCT('Part1','','',(#2)) ;",
              "#6=PRODUCT_DEFINITION_FORMATION_WITH_SPECIFIED_SOURCE('',' '" +
              ",#5,.NOT_KNOWN.) ;",
              "#7=PRODUCT_CATEGORY('part',$) ;",
              "#8=PRODUCT_RELATED_PRODUCT_CATEGORY('detail',$,(#5)) ;",
              "#9=PRODUCT_CATEGORY_RELATIONSHIP(' ',' ',#7,#8) ;",
              "#10=COORDINATED_UNIVERSAL_TIME_OFFSET(0,0,.AHEAD.) ;",
              "#11=CALENDAR_DATE(" + str(getattr(d, 'year')) + "," + str(
                  getattr(d, 'month')) + "," + str(
                  getattr(d, 'day')) + ") ;",
```

```
"#12=LOCAL_TIME(" + str(getattr(d, 'hour')) + "," + str(
    getattr(d, 'minute')) + "," + str(
    getattr(d, 'second')) + ".,#10) ;",
"#13=DATE_AND_TIME(#11,#12) ;",
"#14=PRODUCT_DEFINITION('','',#6,#3) ;",
"#15=SECURITY_CLASSIFICATION_LEVEL('unclassified') ;",
"#16=SECURITY_CLASSIFICATION('','',#15) ;",
"#17=DATE_TIME_ROLE('classification_date') ;",
"#18=CC_DESIGN_DATE_AND_TIME_ASSIGNMENT(#13,#17,(#16)) ;",
"#19=APPROVAL_ROLE('APPROVER') ;",
"#20=APPROVAL_STATUS('not_yet_approved') ;",
"#21=APPROVAL(#20,'') ;",
"#22=PERSON('','','',$,$,$) ;",
"#23=ORGANIZATION('','','') ;",
"#24=PERSONAL_ADDRESS('','','','','','','','','','','" +
"','','','',(#22),'') ;",
"#25=PERSON_AND_ORGANIZATION(#22,#23) ;",
"#26=PERSON_AND_ORGANIZATION_ROLE('classification_officer') ;",
"#27=CC_DESIGN_PERSON_AND_ORGANIZATION_ASSIGNMENT(#25,#26," +
"(#16)) ;",
"#28=DATE_TIME_ROLE('creation_date') ;",
"#29=CC_DESIGN_DATE_AND_TIME_ASSIGNMENT(#13,#28,(#14)) ;",
"#30=CC_DESIGN_APPROVAL(#21,(#16,#6,#14)) ;",
"#31=APPROVAL_PERSON_ORGANIZATION(#25,#21,#19) ;",
"#32=APPROVAL_DATE_TIME(#13,#21) ;",
"#33=CC_DESIGN_PERSON_AND_ORGANIZATION_ASSIGNMENT(#25,#34," +
"(#6)) ;",
"#34=PERSON_AND_ORGANIZATION_ROLE('design_supplier') ;",
"#35=CC_DESIGN_PERSON_AND_ORGANIZATION_ASSIGNMENT(#25,#36," +
"(#6,#14)) ;",
"#36=PERSON_AND_ORGANIZATION_ROLE('creator') ;",
"#37=CC_DESIGN_PERSON_AND_ORGANIZATION_ASSIGNMENT(#25,#38," +
"(#5)) ;",
"#38=PERSON_AND_ORGANIZATION_ROLE('design_owner') ;",
"#39=CC_DESIGN_SECURITY_CLASSIFICATION(#16,(#6)) ;",
"",
"#40=PRODUCT_DEFINITION_SHAPE('','',#14) ;",
"#41=(LENGTH_UNIT()NAMED_UNIT(*)SI_UNIT(.MILLI.,.METRE.)) ;",
"#42=(NAMED_UNIT(*)PLANE_ANGLE_UNIT()SI_UNIT($,.RADIAN.)) ;",
```

```
                "#43=(NAMED_UNIT(*)SI_UNIT($,.STERADIAN.)SOLID_ANGLE_UNIT()) ;",
                "#44=UNCERTAINTY_MEASURE_WITH_UNIT(LENGTH_MEASURE(0.005),#41," +
                "'distance_accuracy_value','CONFUSED CURVE" +
                " UNCERTAINTY') ;",
                "#45=(GEOMETRIC_REPRESENTATION_CONTEXT(3)" +
                "GLOBAL_UNCERTAINTY_ASSIGNED_CONTEXT((#44))" +
                "GLOBAL_UNIT_ASSIGNED_CONTEXT((#41,#42,#43))" +
                "REPRESENTATION_CONTEXT(' ',' ')) ;",
                "",
                "#46=CARTESIAN_POINT(' ',(0.,0.,0.)) ;",
                "#47=AXIS2_PLACEMENT_3D(' ',#46,$,$) ;",
                "#48=SHAPE_REPRESENTATION(' ',(#47),#45) ;",
                "#49=SHAPE_DEFINITION_REPRESENTATION(#40,#48) ;",
                "",
                "/* Part Specification */",
                "",
                "ENDSEC;",
                "END-ISO-10303-21;"]
file_array = [i + "\n" for i in file_array]

index1 = file_array.index("/* Part Specification */\n")
initial_work_array = list(filter(lambda k: k.startswith('#'), file_array))

highest_index = line_index(sorted(initial_work_array, key=line_index)[-1])
current_index = highest_index + 1
part_body_index = 1
work_array = []
print("[DONE]")

generate_assembly(in_array, in_depth, in_coeff)
```

APPENDIX H

CODES FOR SPHERICAL DEFLECTOR TRANSFER MAP AND ORBIT CALCULATIONS

H.1 *COSY INFINITY* Code for Transfer Map Calculation Using ODEs in Laboratory Coordinates

The following is a *COSY INFINITY* code that calculates the transfer map of a 45° sector of the electrostatic spherical deflector in (x,a) beamline coordinates by integrating the ODEs of motion in polar laboratory coordinates. This version of the code uses a 4th order Runge–Kutta integrator with fixed step size. The voltages of the inner and outer spherical shells of the electrostatic spherical deflector are set to result in a circular reference orbit of radius $r_0 = 1$ m. The DA computation order 3 is used.

```
INCLUDE 'COSY' ;

PROCEDURE RUN ;

VARIABLE Y1 4000 8 ; VARIABLE YT 4000 8 ;
VARIABLE XX0 4000 ; {Initial DA-valued x coordinate}
VARIABLE XX1 4000 ; {Final DA-valued x coordinate}
VARIABLE AA0 4000 ; {Initial DA-valued a coordinate}
VARIABLE AA1 4000 ; {Final DA-valued a coordinate}
VARIABLE NM 1 ; {DA variable size}
VARIABLE MU 1 ; {mu}
VARIABLE HPAR 4000 ; {h}

PROCEDURE FNC F X T ;
    {FUNCTION F(X,T) DEFINING THE ODEs OF MOTION
        X  COORDINATES VECTOR
        T  TIME (OR ARC LENGTH) }
```

```
    F(1) := X(2)/X(4) ; {r'}
    F(2) := -MU/X(1)/X(1)/X(4) + X(1)*X(4) ; {v_r'}
    F(3) := 1 ; {theta'}
    F(4) := -2*HPAR/X(1)/X(1)/X(1) * X(2)/X(4) ; {omega'}
ENDPROCEDURE ;

PROCEDURE RK4A N X0 X1 Y0 NS Y1 ;
    {FOURTH ORDER RUNGE KUTTA INTEGRATOR}
    VARIABLE I 1 ; VARIABLE J 1 ; VARIABLE T 1 ;
    VARIABLE Z1 NM 8 ; VARIABLE Z2 NM 8 ;
    VARIABLE Z3 NM 8 ; VARIABLE Z4 NM 8 ;
    VARIABLE Z5 NM 8 ; VARIABLE F NM 8 ;
    VARIABLE HS1 1 ;
    T := X0 ;
    HS1 := (X1-X0)/NS ;
    LOOP J 1 N ; Y1(J) := Y0(J) ; ENDLOOP ;
    LOOP I 1 NS ;
        FNC F Y1 T ;
        LOOP J 1 N ; Z1(J) := HS1*F(J) ; ENDLOOP ;
        LOOP J 1 N ; Z5(J) := Y1(J) + Z1(J)/2 ; ENDLOOP ;
        FNC F Z5 T+HS1/2 ;
        LOOP J 1 N ; Z2(J) := HS1*F(J) ; ENDLOOP ;
        LOOP J 1 N ; Z5(J) := Y1(J) + Z2(J)/2 ; ENDLOOP ;
        FNC F Z5 T+HS1/2 ;
        LOOP J 1 N ; Z3(J) := HS1*F(J) ; ENDLOOP ;
        LOOP J 1 N ; Z5(J) := Y1(J) + Z3(J) ; ENDLOOP ;
        FNC F Z5 T+HS1 ;
        LOOP J 1 N ; Z4(J) := HS1*F(J) ; ENDLOOP ;
        LOOP J 1 N ;
            Y1(J) := Y1(J) + (1/6)*(Z1(J)+2*Z2(J)+2*Z3(J)+Z4(J)) ;
        ENDLOOP ;
        T := T + HS1 ;
    ENDLOOP ;
```

```
ENDPROCEDURE ;

PROCEDURE KEPLERODEPOLAR R00 PHI X0 A0 X1 A1 ;
    {This procedure calculates the transfer map of the electrostatic
    spherical deflector in (x,a) coordinates by integrating the
    ODEs of motion in laboratory coordinates using a Runge-Kutta
    integrator.
    Input parameters:
        R00     Radius of the circular reference orbit
        PHI     Central angle of tracked sector of the deflector
        X0      Initial x coordinate
        A0      Initial a coordinate
        X1      Final x coordinate
        A1      Final a coordinate }
    VARIABLE R0 NM 2 ; {Initial radius vector}
    VARIABLE W0 NM 2 ; {Initial velocity vector}
    VARIABLE R1 NM 2 ; {Final radius vector}
    VARIABLE W1 NM 2 ; {Final velocity vector}
    VARIABLE DOT NM ; {A scalar product}
    VARIABLE R0S NM ; {Initial radius magnitude}
    VARIABLE RS NM ; {Final radius magnitude}
    VARIABLE SIGMA0 NM ; {sigma_0}
    VARIABLE P NM ; {Focal parameter}
    VARIABLE F NM ; VARIABLE G NM ; {Lagrange coefficients F, G}
    VARIABLE FT NM ; VARIABLE GT NM ; {Lagrange coefficients FT, GT}
    VARIABLE V0 NM ; {Reference velocity}
    VARIABLE V1 NM ; {Initial velocity magnitude}
    VARIABLE V2 NM ; {Final velocity magnitude}
    VARIABLE CHIE 1 ; {Electric rigidity}
    PHI := PHI*DEGRAD ;
    V0 := CONS(SQRT(2*ETA)*CLIGHT) ;
    R0(1) := X0 + R00 ; R0(2) := 0 ;
    CHIE := AMU*RE(M0)*V0*V0 / Z0/EZERO ; {R00*CHIE}
```

```
     V1 := SQRT(V0^2-2*Z0*EZERO*CHIE*(1-R00/(R0(1)))/AMU/RE(M0)) ;
     W0(1) := A0*V0 ; W0(2) := SQRT(V1*V1-W0(1)*W0(1)) ;
     MU := R00*CONS(V1)^2 ;
     HPAR := R0(1)*W0(2) ;
     YT(1) := R0(1) ;
     YT(2) := W0(1) ;
     YT(3) := 0 ;
     YT(4) := W0(2)/YT(1) ;
     {RK8A 4 0 PHI YT HS Y1 1e-5 RESCODE ;}
     RK4A 4 0 PHI YT 1e6 Y1;
     X1 := Y1(1) - R00 ;
     V2 := SQRT(Y1(2)*Y1(2)+Y1(1)*Y1(1)*Y1(4)*Y1(4)) ;
     A1 := Y1(2) / CONS(V2) ;
ENDPROCEDURE ;

OV 3 1 0 ;
RP 1 1 1 ;

NM := 4000 ;
XX0 := DA(1) ; AA0 := DA(2) ;

KEPLERODEPOLAR 1 45 XX0 AA0 XX1 AA1 ;
WRITE 6 'TRANSFER MAP OBTAINED IN LAB COORDINATES' ;
WRITE 6 'BY INTEGRATION OF THE ODEs' ;
WRITE 6 'X_f' XX1 'A_f' AA1 ;

ENDPROCEDURE ; RUN ; END ;
```

H.2 *Mathematica* Code for Integration of the ODEs in Laboratory Coordinates

The following is a *Mathematica* notebook that integrates the ODEs of motion of a particle with specified initial beamline coordinates (x,a) through the electrostatic spherical deflector in polar laboratory coordinates, plots one turn of the orbit, and outputs the final beamline coordinates at true anomaly difference $\theta = 45^\circ$. The voltages of the inner and outer spherical shells of the electrostatic spherical deflector are set to result in a circular reference orbit of radius $r_0 = 1\,\text{m}$.

```
(* Integration of the ODEs of motion for a charged \
particle of kinetic energy 1 MeV, mass 1 amu, and \
charge 1 e in an electrostatic spherical deflector \
with the inner sphere charge such that the particle \
would have a circular reference orbit of radius 1 m. *)
v00 = 13891388.79714028; (* Reference velocity *)
r00 = 1; (* Reference orbit radius *)
x = -0.5; (* Initial x beamline coordinate *)
a = 0.5; (* Initial a beamline coordinate *)
r0 = {x + r00, 0, 0}; (* Initial radius vector *)
v0 = {a v00, v00 Sqrt[1 - a^2], 0}; (* Initial velocity vector *)
mu = r00  v00^2; (* mu *)
vr0 = v0[[1]];
rr0 = r0[[1]];
omega0 = v0[[2]]/rr0; (* omega_0 *)
theta0 = 0; (* Initial polar angle *)
system = { (* ODEs and the initial condition *)
   vr'[t] == -mu/(omega[t] (rr[t])^2) + rr[t] omega[t],
   rr'[t] == vr[t]/omega[t],
   omega'[t] == -2 rr0 v0[[2]] vr[t]/(omega[t] (rr[t])^3),
   theta'[t] == 1,
   vr[0] == vr0,
   rr[0] == rr0,
   omega[0] == omega0,
```

```
    theta[0] == theta0
    };
sol = NDSolve[system, {vr, rr, omega, theta}, {t, 0, 2 \[Pi]}] ;
rsol[t_] = First[Evaluate[rr[t] /. sol]];
 (* Radius magnitude as a function of polar angle theta *)
vsol[t_] = First[Evaluate[vr[t] /. sol]];
 (* Velocity magnitude as a function of polar angle theta *)
Print[PolarPlot[
    rsol[t], {t, 0, 2 \[Pi]}]]; (* Plot one turn of the orbit *)
x1[t_] = rsol[t] - r00; (* x coordinate at polar angle theta *)
a1[t_] = vsol[t]/v00; (* a coordinate at polar angle theta *)
Print["x_f=", x1[Pi/4]];
Print["a_f=", a1[Pi/4]];
```

H.3 *COSY INFINITY* Code for Transfer Map Calculation Using the Kepler Theory Transition Matrix

The following is a *COSY INFINITY* code that calculates the transfer map of a 45° sector of the electrostatic spherical deflector in (x,a) beamline coordinates using the Kepler theory transition matrix with Lagrange-coefficients elements. This version of the code uses a 4th order Runge–Kutta integrator with fixed step size. The voltages of the inner and outer spherical shells of the electrostatic spherical deflector are set to result in a circular reference orbit of radius $r_0 = 1\,\mathrm{m}$. The DA computation order 3 is used.

```
INCLUDE 'COSY' ;

PROCEDURE RUN ;

VARIABLE XX0 4000 ; {Initial DA-valued x coordinate}
VARIABLE XX1 4000 ; {Final DA-valued x coordinate}
VARIABLE AA0 4000 ; {Initial DA-valued a coordinate}
VARIABLE AA1 4000 ; {Final DA-valued a coordinate}
```

```
VARIABLE NM 1 ; {DA variable size}

PROCEDURE KEPLERANALYTICPOLAR R00 PHI X0 A0 X1 A1 ;
    VARIABLE R0 NM 2 ; {Initial radius vector}
    VARIABLE W0 NM 2 ; {Initial velocity vector}
    VARIABLE R1 NM 2 ; {Final radius vector}
    VARIABLE W1 NM 2 ; {Final velocity vector}
    VARIABLE MU NM ; {mu}
    VARIABLE DOT NM ; {A scalar product}
    VARIABLE R0S NM ; {Initial radius}
    VARIABLE RS NM ; {Final radius}
    VARIABLE SIGMA0 NM ; {sigma_0}
    VARIABLE P NM ; {Focal parameter p}
    VARIABLE F NM ; VARIABLE G NM ; {Lagrange coefficients F and G}
    VARIABLE FT NM ; VARIABLE GT NM ; {Lagrange coefficients F_t and G_t}
    VARIABLE V0 NM ; {Initial velocity at zero potential}
    VARIABLE V1 NM ; {Initial velocity}
    VARIABLE V2 NM ; {Final velocity}
    VARIABLE CHIE 1 ; {Electric rigidity}
    PHI := PHI*DEGRAD ;
    V0 := CONS(SQRT(2*ETA)*CLIGHT) ;
    R0(1) := X0 + R00 ; R0(2) := 0 ;
    CHIE := AMU*RE(M0)*V0*V0 / Z0/EZERO ; {R00*CHIE}
    V1 := SQRT(V0^2 - 2*Z0*EZERO*CHIE*(1-R00/(R0(1)))/AMU/RE(M0)) ;
    W0(1) := A0*V0 ;
    W0(2) := SQRT(V1*V1-W0(1)*W0(1)) ;
    MU := R00*CONS(V1)^2 ;
    DOT := R0(1)*W0(1) + R0(2)*W0(2) ;
    SIGMA0 := DOT/SQRT(MU) ;
    P := ((R0(1)*R0(1)+R0(2)*R0(2))*(W0(1)*W0(1)+W0(2)*W0(2))-DOT*DOT) / MU ;
    R0S := R0(1) ;
    RS := P*R0S / (R0S+(P-R0S)*COS(PHI)-SQRT(P)*SIGMA0*SIN(PHI)) ;
    F := 1 - (RS/P)*(1-COS(PHI)) ;
```

```
    G := (RS*R0S/SQRT(MU*P)) * SIN(PHI) ;
    FT := (SQRT(MU)/R0S/P) * (SIGMA0*(1-COS(PHI))-SQRT(P)*SIN(PHI)) ;
    GT := 1 - (R0S/P)*(1-COS(PHI)) ;
    R1(1) := F*R0(1) + G*W0(1) ; R1(2) := F*R0(2) + G*W0(2) ;
    W1(1) := FT*R0(1) + GT*W0(1) ; W1(2) := FT*R0(2) + GT*W0(2) ;
    X1 := (COS(PHI)*R1(1)+SIN(PHI)*R1(2)) - R00 ;
    V2 := SQRT(W1(1)*W1(1)+W1(2)*W1(2)) ;
    A1 := (COS(PHI)*W1(1)+SIN(PHI)*W1(2)) / CONS(V2) ;
ENDPROCEDURE ;

OV 3 1 0 ;
RP 1 1 1 ;

NM := 4000 ;
XX0 := DA(1) ; AA0 := DA(2) ;

KEPLERANALYTICPOLAR 1 45 XX0 AA0 XX1 AA1 ;
WRITE 6 'TRANSFER MAP OBTAINED IN LAB COORDINATES' ;
WRITE 6 'USING LAGRANGE COEFFICIENTS' ;
WRITE 6 'X_f' XX1 'A_f' AA1 ;

ENDPROCEDURE ; RUN ; END ;
```

H.4 *COSY INFINITY* Code for Transfer Map Calculation of the Built-in Spherical Deflector Element

The following is a *COSY INFINITY* code that uses our non-relativistic version `COSYNR.FOX` of its beam physics macro package `COSY.FOX` to compute the transfer map of a 45° sector of the electrostatic spherical deflector. The voltages of the inner and outer spherical shells of the electrostatic spherical deflector are set to result in a circular reference orbit of radius $r_0 = 1\,\mathrm{m}$. The DA computation order 3 is used.

```
INCLUDE 'COSYNR' ; {Non-relativistic version of COSY.FOX}

PROCEDURE RUN ;

OV 3 1 0 ;
RP 1 1 1 ;

UM ;
ESP 1 45 10 ; {Electrostatic spheric deflector}

WRITE 6 'TRANSFER MAP OF COSY INFINITY''S ESP ELEMENT' ;
WRITE 6 'X_f' MAP(1) 'A_f' MAP(2) ;

ENDPROCEDURE ; RUN ; END ;
```

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] *OPERA-3d User Guide*. Vector Fields Limited, Oxford, England, 2004. 3.1, 3.3.8

[2] D. Anastassopoulos et al. AGS Proposal: Search for a Permanent Electric Dipole Moment of the Deuteron Nucleus at the $10^{-29}\, e \cdot \mathrm{cm}$ Level. BNL Report, Brookhaven National Laboratory, Upton, NY, April 2008. goo.gl/1K1TNq. 1.1, 1.3, 6.3.1

[3] V. Anastassopoulos et al. A Storage Ring Experiment to Detect a Proton Electric Dipole Moment. *Rev. Sci. Instrum.*, 87(11):115116, 2016. 1.1, 1.1, 5.1, 5.9.1, 5.9.5, 5.10, 6.2.1

[4] V. I. Arnold and S. P. Novikov, editors. *Dynamical Systems IV: Symplectic Geometry and its Applications*, volume 4 of *Encyclopaedia of Mathematical Sciences*. Springer-Verlag Berlin Heidelberg, Berlin, Heidelberg, 2nd edition, 2001. 1.4

[5] Roger Balian, Wolf Beiglböck, and Harald Grosse. *CP Violation Without Strangeness: Electric Dipole Moments of Particles, Atoms, and Molecules*. Springer, Berlin/Heidelberg, 1997. 1.1

[6] L. Banjai. Private communication. 3.2.4.2, 3.2.5

[7] L. Banjai. Revisiting the Crowding Phenomenon in Schwarz–Christoffel Mapping. *SIAM J. Sci. Comput.*, 30(2):618–636, 2008. 3.2.4.2, 3.2.5

[8] Jerry Banks, editor. *Handbook of Simulation: Principles, Methodology, Advances, Applications, and Practice*. John Wiley & Sons, Inc., New York, NY, 1998. 1

[9] Richard H. Battin. *An Introduction to the Mathematics and Methods of Astrodynamics, Revised Edition*. American Institute of Aeronautics and Astronautics, Reston, VA, rev. edition, 1999. 5.8.2, 5.8.2, 5.8.2, 5.8.2

[10] Martin Berz. *Symplectic Tracking Through Circular Accelerators with High Order Maps*, pages pp. 288–296. World Scientific Publishing, Singapore, 1991. 4.1.2.5

[11] Martin Berz. *Modern Map Methods in Particle Beam Physics*. Advances in Imaging and Electron Physics. Academic Press, San Diego, CA, 1999. 1.4, 1.5, 3.2.3, 3.2.3.2, 4.1, 4.1, 4.1.2, 4.1.2.1, 4.1.2.5, 4.1.2.5, 5.3, 5.5, 5.8.1, 5.8.2, 5.8.3, 5.9.2.4, 6.2.1, 6.2.2, 6.5.2, 3

[12] Martin Berz and Kyoko Makino. COSY INFINITY 10.0 Beam Physics Manual. MSU Report MSUHEP 151103-rev, Department of Physics and Astronomy, Michigan State University, East Lansing, MI 48824, 2017. See also http://cosyinfinity.org. 1.5, 4.1.2.5, 5.2.2.1, 5.2.2.1, 5.5

[13] Martin Berz, Kyoko Makino, and Weishi Wan. *An Introduction to Beam Physics*. CRC Press, London, 2014. 3.3.2, 3.3.2, 6.2.3

[14] Thomas Blum et al. The Muon (g-2) Theory Value: Present and Future, 2013. arXiv:1311.2198. 1.2

[15] Timothy H. Boyer. Unfamiliar Trajectories for a Relativistic Particle in a Kepler or Coulomb Potential. *Am. J. Phys.*, 72(8):992–997, 2004. 5.7.2.2, 5.7.2.2

[16] William L. Briggs and Van Emden Henson. *The DFT: An Owner's Manual for the Discrete Fourier Transform*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 1995. 3.2.4.1.2

[17] R. M. Carey et al. The New (g-2) Experiment: A Proposal to Measure the Muon Anomalous Magnetic Moment to $\pm$0.14 ppm Precision. Technical report, New g-2 Collaboration, Fermi National Accelerator Laboratory, Batavia, IL, Feb. 2009. 1.2

[18] Alexander W. Chao, Karl H. Mess, and Maury Tigner. *Handbook of Accelerator Physics and Engineering (2nd Edition)*. World Scientific Publishing, Singapore, 2nd edition, 2013. 6.5

[19] G. Charpak et al. The Anomalous Magnetic Moment of the Muon. *Nuovo Cim.*, 37(4):1241–1363, 1965. 1.2

[20] F. Combley, F. J. M. Farley, and E. Picasso. The CERN Muon (g-2) Experiments. *Phys. Rep.*, 68(2):93–119, 1981. 1.2

[21] F. Combley and E. Picasso. The Muon (g-2) Precession Experiments: Past, Present and Future. *Phys. Rep.*, 14(1):1–58, 1974. 1.2

[22] Darren Crowdy. The Schwarz–Christoffel Mapping to Bounded Multiply Connected Polygonal Domains. *Proc. R. Soc. A*, 461(2061):2653–2678, 2005. 3.2.4.2, 3.2.5

[23] Michel Davier, Andreas Hoecker, Bogdan Malaescu, and Zhiqing Zhang. Reevaluation of the Hadronic Contributions to the Muon g-2 and to $\alpha(m_Z^2)$. *Eur. Phys. J. C*, 71(1):1–13, 2011. 1.2, 1.1

[24] J. E. Dennis and Robert B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Prentice Hall, Englewood Cliffs, NJ, 1983. 1

[25] Michael Dine and Alexander Kusenko. Origin of the Matter–Antimatter Asymmetry. *Rev. Mod. Phys.*, 76(1):1–30, 2003. 1.1

[26] *NIST Digital Library of Mathematical Functions*. http://dlmf.nist.gov/, Release 1.0.14 of 2016-12-21. F. W. J. Olver, A. B. Olde Daalhuis, D. W. Lozier, B. I. Schneider, R. F. Boisvert, C. W. Clark, B. R. Miller and B. V. Saunders, eds. A.1

[27] Tobin A. Driscoll. *The Schwarz–Christoffel Toolbox for MATLAB*. http://www.math.udel.edu/ driscoll/SC/. A.3

[28] Tobin A. Driscoll. Algorithm 756: A MATLAB Toolbox for Schwarz–Christoffel Mapping. *ACM Trans. Math. Softw.*, 22(2):168–186, June 1996. 2.3.2, 3.2.2, A.3

[29] Tobin A. Driscoll. *Schwarz–Christoffel Toolbox User's Guide*. Version 2.3, October 2002. http://www.math.udel.edu/ driscoll/SC/guide.pdf. A.2, A.3, A.3

[30] Tobin A. Driscoll and Lloyd N. Trefethen. *Schwarz–Christoffel Mapping*. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, Cambridge, UK, 2002. 2.3.2.1.1, 2.3.2.4, 3.2.2, 3.2.4.2, 3.2.4.2, 3.2.5, A.2, A.2

[31] Tobin A. Driscoll and Stephen A. Vavasis. Numerical Conformal Mapping Using Cross-Ratios and Delaunay Triangulation. *SIAM J. Sci. Comput.*, 19(6):1783–1803, 1998. 3.2.4.2, 3.2.5

[32] INTEGRATED Engineering Software. *COULOMB*. https://www.integratedsoft.com/product/coulomb/. 2.4.1

[33] B. Erdélyi and M. Berz. Optimal Symplectic Approximation of Hamiltonian Flows. *Phys. Rev. Lett.*, 87(11):114302, 2001. 4.1.2.5

[34] D. Eversmann et al. New Method for a Continuous Determination of the Spin Tune in Storage Rings and Implications for Precision Experiments. *Phys. Rev. Lett.*, 115(9):094801, 2015. 6.5.6

[35] F. J. M. Farley. The 46 Years of Muon g-2. *Nucl. Phys. B (Proc. Suppl.)*, 131:235–243, 2004. 1.2

[36] Étienne Forest. Geometric Integration for Particle Accelerators. *J. Phys. A: Math. Gen.*, 39(19):5321–5377, 2006. 4.1.2, 4.1.2.2, 4.1.2.3

[37] Takeshi Fukuyama and Alexander J. Silenko. Derivation of Generalized Thomas-Bargmann-Michel-Telegdi Equation for a Particle with Electric Dipole Moment. *Int. J. Mod. Phys. A*, 28(29):1350147, 2013. 1.3

[38] Miguel Furman. Simple Method to Symplectify Matrices. Technical Report SSC-006, SSC Central Design Group/URA, Lawrence Berkeley Lab., Berkeley, CA, 1985. 4.1.2.3

[39] M. Gatchell et al. Commissioning of the DESIREE Storage Rings – a New Facility for Cold Ion-Ion Collisions. *J. Phys. Conf. Ser.*, 488(1):012040, 2014. 1

[40] R. Golub, D. Richardson, and S. K. Lamoreaux. *Ultra-Cold Neutrons*. CRC Press, New York, NY, 1991. 1.1

[41] Mark J. Gotay and James A. Isenberg. The Symplectization of Science: Symplectic Geometry Lies at the Very Foundations of Physics and Mathematics. *Gazette des mathématiciens*, 54:59–79, 1992. 1.4

[42] Ronald L. Graham, Donald E. Knuth, and Oren Patashnik. *Concrete Mathematics: A Foundation for Computer Science*. Addison-Wesley, Reading, MA, 1989. 2.2

[43] J. Grange et al. Muon (g-2) Technical Design Report. Design Report FERMILAB-FN-0992-E, Muon g-2 Collaboration, Fermi National Accelerator Laboratory, Batavia, IL, January 2015. 1.2, 1.2, 1.2, 6.2.1

[44] W. C. Griffith, M. D. Swallows, T. H. Loftus, M. V. Romalis, B. R. Heckel, and E. N. Fortson. Improved Limit on the Permanent Electric Dipole Moment of ^{199}Hg. *Phys. Rev. Lett.*, 116(16):161601, 2016. 1.1

[45] Greta Guidoboni. Spin Coherence Time Studies for the Storage Ring EDM Search. Ph.D. thesis, Università degli Studi di Ferrara, Ferrara, Italy, 2012. 6.2.1

[46] S. Guiducci. Chromaticity. In *CERN Accelerator School: Course on General Accelerator Physics Jyvaskyla, Finland, September 7-18, 1992*, pages 191–206, 1992. http://inspirehep.net/record/347563/files/p191.pdf. 6.2.4.4

[47] John L. Gustafson. *The End of Error: Unum Computing*, volume 24 of *Chapman & Hall/CRC Computational Science*. Chapman and Hall/CRC, Boca Raton, FL, USA, 2015. A.3

[48] Kaoru Hagiwara, Ruofan Liao, Alan D. Martin, Daisuke Nomura, and Thomas Teubner. $(g\text{-}2)_\mu$ and $\alpha(m_z^2)$ Re-evaluated Using New Precise Data. *J. Phys. G: Nucl. Part. Phys.*, 38:085003, 2011. 1.2, 1.1

[49] Thomas Hambye. CP Violation and the Matter–Antimatter Asymmetry of the Universe. *C. R. Phys.*, 13(2):193–203, 2012. 1.1

[50] Mark P. Hertzberg and Johanna Karouby. Generating the Observed Baryon Asymmetry From the Inflaton Field. *Phys. Rev. D*, 89(6):063523, 2014. 1.1

[51] Robert Hipple and Martin Berz. Implementation of Benchmarks for Precision Tracking in Storage Rings. In *Proc. of International Computational Accelerator Physics Conference (ICAP'15), Shanghai, China, 12–16 October 2015*, pages 13–16, Geneva, Switzerland, 2016. JACoW. https://doi.org/10.18429/JACoW-ICAP2015-MODBC3. 5.1

[52] A. Hoecker and W. J. Marciano. The Muon Anomalous Magnetic Moment. *Chin. Phys. C*, (10):635–638, 2016. 1.2

[53] V. W. Hughes et al. BNL AGS E821: A New Precision Measurement of the Muon (g-2) Value at the Level of 0.35 ppm. Design Report, Brookhaven National Laboratory, Upton, NY, 2nd (revised) printing, Sep. 1995. 1.2, 1.2

[54] N. Ida. *Numerical Modeling for Electromagnetic Non-Destructive Evaluation*. Chapman & Hall, London, 1995. 2.3.1

[55] ISO/IEC. *IS 10303-242:2014 Industrial Automation Systems and Integration – Product Data Representation and Exchange – Part 242: Application Protocol: Managed Model-Based 3D Engineering*. International Organization for Standardization, Geneva, Switzerland, 2014. 2.4.1, 1, 3.3.3, 1

[56] John D. Jackson. *Classical Electrodynamics*. John Wiley & Sons, New York, 3rd edition, 1999. 2.5

[57] Ravi Jagasia. Private communication. 3.3.2

[58] C. Kane, J. E. Marsden, and M. Ortiz. Symplectic-Energy-Momentum Preserving Variational Integrators. *J. Math. Phys.*, 40(7):3353–3371, 1999. 4.2.1

[59] D. M. Kara, I. J. Smallman, J. J. Hudson, B. E. Sauer, M. R. Tarbutt, and E. A. Hinds. Measurement of the Electron's Electric Dipole Moment Using YbF Molecules: Methods and Data Analysis. *New J. Phys.*, 14(10):103051, 2012. 1.1

[60] Frank Kreith and D. Yogi Goswami, editors. *The CRC Handbook of Mechanical Engineering, Second Edition*. CRC Press, Boca Raton, FL, 2nd edition, 2004. 6.5

[61] F. T. Krogh. Variable Order Integrators for the Numerical Solution of Ordinary Differential Equations. NASA Tech Brief 71-10248, Jet Propulsion Lab., Pasadena, CA, 1971. 1

[62] L. D. Landau and E. M. Lifshitz. *Mechanics*. Elsevier Butterworth-Heinemann, Oxford, 3rd English edition, 2005. 5.7.2.1, 5.7.2.1

[63] L. D. Landau and E. M. Lifshitz. *The Classical Theory of Fields*. Elsevier Butterworth-Heinemann, Oxford, 4th English edition, 2010. 5.7.2.2, 3

[64] S. Y. Lee. *Accelerator Physics*. World Scientific Publishing, Singapore, 3 edition, 2012. 6.2.4.4

[65] Andreas Lehrach. Searching Electric Dipole Moments in Storage Rings – Challenges, Status and Computational Needs. In *Proc. of International Computational Accelerator Physics Conference (ICAP'15), Shanghai, China, 12–16 October 2015*, pages 128–131, Geneva, Switzerland, 2016. JACoW. https://doi.org/10.18429/JACoW-ICAP2015-THAJI1. 1.1, 5.1

[66] W. W. Mackay. Comment on the Healy's Symplectification Algorithm. Technical Report BNL-75461-2006-CP, Collider-Accelerator Department, Brookhaven National Laboratory, Upton, NY, June 2006. Presented at the 10th Biennial European Particle Accelerator Conference (EPAC), Edinburgh, UK, June 26–30, 2006. 4.1.2

[67] Jan R. Magnus and Heinz Neudecker. *Matrix Differential Calculus with Applications in Statistics and Econometrics*. John Wiley & Sons, New York, 2nd edition, 1999. 4.1.2.4

[68] Kyoko Makino and Martin Berz. COSY INFINITY Version 9. *Nucl. Instr. Meth. Phys. Res. A*, 558(1):346–350, 2006. 1.5, 3.2.4, 3.3.7, 6.4.1

[69] Kyoko Makino and Martin Berz. Dynamics in Electrostatic Repetitive Systems via High-Order Transfer Maps. *Microsc. Microanal.*, 21(S4):36–43, 2015. 5.4

[70] Kyoko Makino, Eremey Valetov, and Martin Berz. Computation of Nonlinear Fields and Orbit and Spin Transfer Maps of Electrostatic Elements Using Differential Algebras. In *Proc. of International Computational Accelerator Physics Conference (ICAP'15), Shanghai, China, 12-16 October 2015*, number 12 in International Computational Accelerator Physics Conference, pages 168–171, Geneva, Switzerland, Feb. 2016. JACoW. doi:10.18429/JACoW-ICAP2015-THDBC1. 5.3, 5.3

[71] Kyoko Makino, Eremey Valetov, and Martin Berz. E989 Note 104: Detailed Linear Optics and Tunes of the g-2 Ring. G-2 Experiment Document GM2-doc-5715, Muon g-2 Collaboration, Fermi National Accelerator Laboratory, Batavia, IL, Mar. 2017. 3.3.8

[72] Olli Mali, Pekka Neittaanmaeki, and Sergey Repin. *Accuracy Verification Methods: Theory and Algorithms*, volume 32 of *Computational Methods in Applied Sciences*. Springer, Dordrecht, Netherlands, 1st edition, 2014. 2

[73] J. A. Maloney, R. Baartman, T. Planche, and S. Saminathan. Electrostatic Potential Map Modelling with COSY INFINITY. *Nucl. Instr. Meth. Phys. Res. B*, 376:171–174, 2016. 1.4

[74] Thomas Mannel and Nikolai Uraltsev. Loop-Less Electric Dipole Moment of the Nucleon in the Standard Model. *Phys. Rev. D*, 85(9):096002, 2012. 1.1

[75] Jerrold E. Marsden. *Lectures on Mechanics*. Cambridge University Press, Cambridge, 2009. 4.2.1

[76] Edwin M. McMillan. The Synchrotron–A Proposed High Energy Particle Accelerator. *Phys. Rev.*, 68(5-6):143–144, 1945. 6.2.4.1

[77] E. M. Metodiev et al. Analytical Benchmarks for Precision Particle Tracking in Electric and Magnetic Rings. *Nucl. Instrum. Methods Phys. Res. A*, 797:311–318, 2015. 5.1

[78] E. M. Metodiev, K. L. Huang, Y. K. Semertzidis, and W. M. Morse. Fringe Electric Fields of Flat and Cylindrical Deflectors in Electrostatic Charged Particle Storage Rings. *Phys. Rev. ST Accel. Beams*, 17(7):074002, 2014. 5.1, 5.6.2.1, 5.9.1, 5.9.2.1, 5.9.2.1, 5.9.2.3, 5.9.5, 5.9.5.1, 5.10

[79] Eric M. Metodiev. Thomas–BMT Equation Generalized to Electric Dipole Moments and Field Gradients, 2015. arXiv:1507.04440. 1.3

[80] Piotr Mikusiínski and Michael D. Taylor. *Introduction to Multivariable Analysis from Vector to Manifold*. Birkhäuser Basel, Basel, Switzerland, 1 edition, 2002. 4.2.2

[81] Jorge Cortés Monforte. *Geometric, Control and Numerical Aspects of Nonholonomic Systems*. Springer-Verlag Berlin Heidelberg, Berlin, Heidelberg, 2002. 4.2.1

[82] B. D. Muratori, J. K. Jones, and A. Wolski. Analytical Expressions for Fringe Fields in Multipole Magnets. *Phys. Rev. ST Accel. Beams*, 18(6):064001, 2015. 1.4

[83] Marcin Patecki and Rogelio Tomás. Effects of Quadrupole Fringe Fields in Final Focus Systems for Linear Colliders. *Phys. Rev. ST Accel. Beams*, 17(10):101002, 2014. 1.4

[84] J. M. Pendlebury et al. A Revised Experimental Upper Limit on the Electric Dipole Moment of the Neutron. *Phys. Rev. D*, 82:092003, 2015. 1.1, 6.2.1

[85] Martin Plotkin. The Brookhaven Electron Analogue, 1953–1957. Internal Report BNL-45058, AGS Department, Brookhaven National Laboratory, Upton, NY, Dec. 1991. 1

[86] A. D. Polyaanin and A. V. Manzhirov. *Handbook of Integral Equations*. CRC Press, Boca Raton, Florida, 1998. 2.3.1.1

[87] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, New York, NY, 3rd edition, 2007. A.1

[88] N. F. Ramsey. Electric-Dipole Moments of Elementary Particles. *Rep. Prog. Phys.*, 45(1):95–113, 1982. 1.1

[89] Frank Rathmann. Private communication. 1.1, 6.2.1

[90] Frank Rathmann. Spin Physics and Deuteron EDMs in Storage Rings. Presentation, KITP Conference: Symmetry Tests in Nuclei and Atoms, Kavli Institute for Theoretical Physics, University of California, Santa Barbara, CA, Sep. 2016. 6.2.1

[91] Sergey Repin. Lectures on a Posteriori Error Analysis. Lecture notes, University of Saarbrüucken, Saarbrüucken, Germany, 2008. https://www.math.uni-sb.de/ag/fuchs/Menupkte/SB01.pdf. 2

[92] A. D. Sakharov. Violation of CP Invariance, C Asymmetry, and Baryon Asymmetry of the Universe. *Pisma Zh. Eksp. Teor. Fiz.*, 5:32–35, 1967. [Usp. Fiz. Nauk 161, 61–64 (1991)]. 1.1

[93] Yannis K. Semertzidis et al. The Brookhaven Muon (g-2) Storage Ring High Voltage Quadrupoles. *Nucl. Instr. Meth. Phys. Res. A*, 503(3):458–484, 2003. 3.1

[94] Yurij Senichev. Private communication. 6.2.1, 6.2.3, 6.3.1, 6.3.2.2, 6.3.2.3

[95] Yurij Senichev, Alexander Aksentyev, Stanislav Chekmenev, Jörg Pretz, Serge Andrianov, Andrei Ivanov, Martin Berz, and Eremey Valetov. Quasi-Frozen Spin Concept of Deuteron Storage Ring as an Instrument to Search for the Electric Dipole Moment. In *Proc. of International Particle Accelerator Conference (IPAC'17), Copenhagen, Denmark, 14–19 May, 2017*, number 8 in International Particle Accelerator Conference, pages 2275–2277, Geneva, Switzerland, May 2017. JACoW. https://doi.org/10.18429/JACoW-IPAC2017-TUPVA084. 6.5.6

[96] Yurij Senichev, Serge Andrianov, Andrei Ivanov, Stanislav Chekmenev, Martin Berz, and Eremey Valetov. Investigation of Lattice for Deuteron EDM Ring. In *Proc. of International Computational Accelerator Physics Conference (ICAP'15), Shanghai, China, 12–16 October 2015*, pages 17–19, Geneva, Switzerland, 2016. JACoW. https://doi.org/10.18429/JACoW-ICAP2015-MODBC4. 6.3.1, 6.3.2.1, 6.3.2.3

[97] Yurij Senichev, Andrei Ivanov, Andreas Lehrach, Rudolf Maier, Denis Zyuzin, and Serge Andrianov. Spin Tune Decoherence in Multipole Fields. In *Proceedings, 5th International Particle Accelerator Conference (IPAC'2014): Dresden, Germany, June 15–20, 2014*, pages 3017–3019, Geneva, Switzerland, 2014. JACoW. https://doi.org/10.18429/JACoW-IPAC2014-THPRO062. 6.2.4.3, 6.2.4.4

[98] Yurij Senichev, Andreas Lehrach, Bernd Lorentz, Rudolf Maier, Serge Andrianov, Andrei Ivanov, Stanislav Chekmenev, Martin Berz, and Eremey Valetov. Quasi-Frozen Spin Method for EDM Deuteron Search. In *Proceedings, 6th International Particle Accelerator Conference (IPAC'2015): Richmond, Virginia, USA, May 3–8, 2015*, pages 213–215, Geneva, Switzerland, 2015. JACoW. https://doi.org/10.18429/JACOW-IPAC2015-MOPWA044. 1.1, 6.1, 6.3.2.1, 6.3.2.1, 6.3.2.1, 6.3.2.2

[99] Yurij Senichev, Bernd Lorentz, Serge Andrianov, Andrei Ivanov, Stanislav Chekmenev, Jörg Pretz, Martin Berz, and Eremey Valetov. Systematic Errors Investigation in Frozen and Quasi-Frozen Spin Lattices of Deuteron EDM Ring. In *Proceedings, 7th International Particle Accelerator Conference (IPAC'2016): Busan, Korea, May 8–13, 2016*, pages 3394–3396, Geneva, Switzerland, 2016. JACoW. https://doi.org/10.18429/JACoW-IPAC2016-THPMR005. 6.5, 6.5.4, 6.5.4, 6.5.4, 6.5.5, 6.5.5, 6.5.6

[100] Yurij Senichev, Rudolf Maier, Denis Zyuzin, and Natalia Kulabukhova. Spin Tune Decoherence Effects in Electro- and Magnetostatic Structures. In *Proceedings, 4th International Particle Accelerator Conference (IPAC'2013): Shanghai, China, May 12–17, 2013*, pages 2579–2581, Geneva, Switzerland, 2013. JACoW. Program code WEPEA036. 6.2.4.1, 6.2.4.2, 6.2.4.3

[101] Andrei Seryi. *Unifying Physics of Accelerators, Lasers and Plasma*. CRC Press, London, 2015. 1

[102] M. L. Shashikant, Martin Berz, and Bela Erdélyi. COSY INFINITY's EXPO Symplectic Tracking for LHC. *Inst. Phys. Conf. Ser.*, 175:299–306, 2005. 5.5

[103] Jing Shu, Tim M. P. Tait, and Carlos E. M. Wagner. Baryogenesis From an Earlier Phase Transition. *Phys. Rev. D*, 75(6):063510, 2007. 1.1

[104] Mariléne da Silva e Silva. Sensitive Polarimetry in a Search for the Deuteron Electric Dipole Moment. Ph.D. thesis, University of Groningen, Groningen, the Netherlands, 2010. 6.2.1

[105] Orestes N. Stavroudis. *The Mathematics of Geometrical and Physical Optics: The k-Function and its Ramifications*. Wiley-VCH, Weinheim, Germany, 1st edition, 2006. 1.5

[106] Richard Talman. *The Electric Dipole Moment Challenge*. Morgan & Claypool Publishers, San Rafael, CA, 2017. 6.2.1

[107] Richard M. Talman and John D. Talman. EDM Planning Using ETEAPOT with a Resurrected AGS Electron Analogue Ring, 2015. arXiv:1503.08494. 1

[108] Lloyd N. Trefethen. Numerical Computation of the Schwarz–Christoffel Transformation. *SIAM J. Sci. and Stat. Comput.*, 1(1):82, 1980. A.2

[109] Eremey Valetov, Martin Berz, and Yurij Senichev. Search for the Optimal Spin Decoherence Effect in a QFS Lattice. In *Proc. of International Computational Accelerator Physics Conference (ICAP'15), Shanghai, China, 12–16 October 2015*, pages 172–174, Geneva,

Switzerland, 2016. JACoW. https://doi.org/10.18429/JACoW-ICAP2015-THDBC2. 6.4.5, 6.4.5

[110] V. Veksler. A New Method of Acceleration of Relativistic Particles. *J. Phys.*, 9(3):153–158, 1945. USSR. 6.2.4.1

[111] J. H. Verner. Explicit Runge–Kutta Methods with Estimates of the Local Truncation Error. *SIAM J. Numer. Anal.*, 15(4):772–790, 1978. 5.6

[112] Weishi Wan, Carol Johnstone, Jim Holt, et al. The Influence of Fringe Fields on Particle Dynamics in the Large Hadron Collider. *Nucl. Instr. Meth. Phys. Res. A*, 427(1):74–78, 1999. 1.4

[113] Helmut Wiedemann. *Particle Accelerator Physics*. Springer International Publishing, Cham, Switzerland, 2015. 6.2.3, 6.2.4.1

[114] H. Wollnik, J. Brezina, and M. Berz. GIOS-BEAMTRACE – A Program Package to Determine Optical Properties of Intense Ion Beams. *Nucl. Instr. Meth. Phys. Res. A*, 258(3):408–411, 1987. 5.1, 5.8

[115] Hermann Wollnik and Martin Berz. Relations Between Elements of Transfer Matrices Due to the Condition of Symplecticity. *Nucl. Instr. Meth. Phys. Res. A*, 238(1):127–140, 1985. 5.8.3

[116] Wanwei Wu. 3D Electric Field Map From Opera 3D. G-2 Experiment Document GM2-doc-4490, Muon g-2 Collaboration, Fermi National Accelerator Laboratory, Batavia, IL, Oct. 2016. VC/BD/Ring Meeting held on 27 Oct 2016. 3.1, 3.3.8

[117] Ge Zhong and Jerrold E. Marsden. Lie–Poisson Hamilton–Jacobi Theory and Lie–Poisson Integrators. *Phys. Lett. A*, 133(3):134–139, 1988. 1.4, 4.2.1