

Derivation of Analytic Formulas for Electrostatic Deflector Aberrations, and Comparison with the Code *COSY INFINITY*

MSU Report MSUHEP 180212

Eremey Valetov and Martin Berz

February 12, 2018

Abstract

Analytic formulas for aberrations of an electrostatic deflector can be obtained using an iterative order-by-order perturbation method. We will derive the first and second order analytic formulas in the horizontal transverse plane x - a for an electrostatic deflector specified by the reference orbit radius, the central angle spanning the deflector, and its inhomogeneity coefficients. The derivation will be performed in a Frenet–Serret beamline coordinate system.

For the test cases of an electrostatic spherical deflector and an electrostatic cylindrical deflector, we will perform a comparison between aberrations obtained using (1) the derived analytic formulas and (2) differential-algebraic (DA) numerical integration of the equations of motion in *COSY INFINITY*. A comparison will be also made with the results of our earlier report [1], where we computed the DA transfer maps of electrostatic spherical and cylindrical deflectors in polar laboratory coordinates. The appendices include a C program titled `edabrt` for calculation of the first and second order aberrations of an electrostatic deflector, as well as the *COSY INFINITY* codes used for the test cases comparison.

Michigan State University, East Lansing, MI 48824, USA

Contents

1	Introduction	2
2	General Considerations	3
3	Order-by-Order Derivation of Aberration Formulas	3
4	Comparison of Aberration Calculations with <i>COSY INFINITY</i>	10
4.1	Electrostatic Spherical Deflector	11
4.1.1	Calculation using <code>edabrt</code>	11
4.1.2	Calculation using <i>COSY INFINITY</i>	11
4.2	Electrostatic Cylindrical Deflector	12
4.2.1	Calculation using <code>edabrt</code>	12
4.2.2	Calculation using <i>COSY INFINITY</i>	12
5	Conclusion	13
	References	13

A	Electrostatic Deflector Aberrations Calculated in Polar Laboratory Coordinates	15
A.1	Electrostatic Spherical Deflector Aberrations Obtained by Integrating the ODEs in Laboratory Coordinates	15
A.2	Electrostatic Spherical Deflector Aberrations Obtained Using Lagrange Coefficients in Laboratory Coordinates	16
A.3	Electrostatic Cylindrical Deflector Aberrations Obtained by Integrating the ODEs in Laboratory Coordinates	17
B	Inhomogeneity Coefficients of Electrostatic Spherical and Cylindrical Deflectors	18
C	<i>COSY INFINITY</i> Code for Calculation of Aberrations of an Electrostatic Spherical Deflector	19
D	<i>COSY INFINITY</i> Code for Calculation of Aberrations of an Electrostatic Cylindrical Deflector	20
E	C Program edabrt for Calculation of Aberrations of an Electrostatic Deflector	21

1 Introduction

We will derive the first and second order aberration formulas in the horizontal transverse plane x - a for an electrostatic deflector of reference orbit radius R_0 , central angle α , midplane symmetry, and the electrostatic field $E(x)$ in the midplane given by first and second order inhomogeneity indices n_1 and n_2 . The derivation will be performed using an iterative order-by-order perturbation method, resulting in exact, analytic aberration formulas. Subsequently, we will calculate the first and second order aberrations for an electrostatic spherical deflector and an electrostatic cylindrical deflector using these analytic formulas, and we will compare the results with *COSY INFINITY*'s respective elements ESP and ECL. Additionally, we will compare the results of this report to the results of our earlier report [1], where we computed the DA transfer maps of electrostatic spherical and cylindrical deflectors in polar laboratory coordinates (1) by integrating the ODEs of motion using a Runge–Kutta integrator and, in case of the electrostatic spherical deflector, (2) by performing analytic computations for the respective elliptical orbits from Kepler theory.

The beam physics tracking code *COSY INFINITY* [9] uses a beamline coordinate system with the phase space coordinates [3, p. 9]

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

Coordinates x and y are the transversal Frenet-Serret position coordinates, 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 arc length coordinate in *COSY INFINITY* is denoted by s .

COSY INFINITY's particle optical elements ESP and ECL compute the transfer map of electrostatic spherical and cylindrical deflectors, respectively. Additionally, *COSY INFINITY*'s inhomogeneous combined-function electrostatic elements ES and EC calculate the transfer map for electrostatic deflectors with the expansion of the electrostatic field $E(x)$ in the midplane x - s given by inhomogeneity indices n_i in the expansion

$$E(x) = E_0 \left[1 - \sum_{j=1}^N n_j \left(\frac{x}{R_0} \right)^j \right] + O(x^{N+1}), \tag{1}$$

where R_0 is the reference orbit radius. The expansion order N is $N = 5$ in case of ES and an arbitrary natural number in case of EC. The electrostatic deflectors are assumed to have the midplane symmetry.

2 General Considerations

We consider a bunch of non-relativistic particles of mass m and charge Ze , where e is the elementary charge. The particles are launched with kinetic energy $K_0 = mv_0^2/2$, where v_0 is the reference velocity. We denote the electrostatic potential by φ .

Let h be the curvature of the reference orbit, $\eta = [K_0(1 + \delta_K) - Ze\varphi]/mc^2$, $p = mv$ the particle momentum, and¹

$$\zeta = \frac{p_s}{p_0} = \sqrt{\frac{\eta}{\eta_0} - a^2 - b^2}. \quad (2)$$

We denote electric and magnetic rigidities by $\chi_e = pv/Ze$ and $\chi_m = p/Ze$, respectively. We use the convention of the electrostatic potential φ as zero for the reference particle. Hence, $\eta_0 = K_0/mc^2$.

The particle beam is subject to the following [11, pp. 146–147] non-relativistic ODEs of motion.

$$\begin{aligned} x' &= a(1 + hx) \frac{1}{\zeta}, & a' &= (1 + hx) \left(\frac{1}{\zeta} \frac{E_x}{\chi_{e0}} - \frac{B_y}{\chi_{m0}} + b \frac{1}{\zeta} \frac{B_s}{\chi_{m0}} \right) + h\zeta, \\ y' &= b(1 + hx) \frac{1}{\zeta}, & b' &= (1 + hx) \left(\frac{1}{\zeta} \frac{E_y}{\chi_{e0}} + \frac{B_x}{\chi_{m0}} - a \frac{1}{\zeta} \frac{B_s}{\chi_{m0}} \right), \\ \delta'_K &= 0, & l' &= -\frac{1}{2} \left[(1 + hx) \frac{1}{\zeta} - 1 \right]. \end{aligned}$$

In [1], we used these non-relativistic ODEs of motion for calculation of transfer maps of *COSY INFINITY*'s electrostatic deflector elements ESP and ECL, and we ascertained that they produce the same results as trivial non-relativistic ODEs of motion in a laboratory coordinate system and, in case of an electrostatic spherical deflector, analytic Kepler theory solutions.

For convenience of notation, we will denote² the n -th order multivariable Taylor expansion of a function $f(\vec{z})$ in beamline coordinates \vec{z} as $[f(\vec{z})]_n$.

3 Order-by-Order Derivation of Aberration Formulas

For an electrostatic deflector with reference orbit curvature $h = R_0^{-1}$ and inhomogeneity indices n_1, n_2 as per eq. 1, the 2nd order Taylor expansions of the ODE of x - a motion in the electrostatic deflector are

$$\left[\frac{d}{ds} \begin{pmatrix} x \\ a \end{pmatrix} \right]_2 = \left[\frac{1}{\zeta} \begin{pmatrix} a(1 + hx) \\ (1 + hx) E_x / \chi_{e0} + h\zeta^2 \end{pmatrix} \right]_2, \quad (3)$$

where χ_{e0} is the reference electric rigidity $\chi_{e0} = mv_0^2/Ze$, the E_x component of the electrostatic field is

$$[E_x]_2 = [E_x(x)]_2 = E_0 \left(1 - \sum_{j=1}^2 n_j h^j x^j \right)$$

in the midplane,

$$\zeta = \sqrt{\frac{\eta}{\eta_0} - a^2}$$

expresses ζ from eq. 2 in the midplane, and

$$\eta = \frac{1}{mc^2} (K_0(1 + \delta_K) - Ze\varphi).$$

We note that the inhomogeneity indices (n_1, n_2) are $(2, -3)$ for an electrostatic spherical deflector and $(1, -1)$ for an electrostatic cylindrical deflector (see app. B or [3, p. 19]).

¹The relativistic version of eq. 2 is $\zeta = \frac{p_s}{p_0} = \sqrt{\frac{\eta(\eta+2)}{\eta_0(\eta_0+2)} - a^2 - b^2}$.

²This is similar, but not the same, as the notation used for differential-algebraic equivalence classes introduced in [2, p. 91].

By Newton's second law, zero radial acceleration at the reference orbit requires

$$ZeE_0 + \frac{mv_0^2}{R_0} = 0,$$

which we rewrite as

$$E_0 = -h\chi_{e0}. \quad (4)$$

To obtain the electrostatic potential in the midplane, we integrate the electrostatic field:

$$\begin{aligned} [\varphi(x)]_3 &= [\varphi(x, y)|_{y=0}]_3 = -\int_0^x [E_x(x)]_2 dx = \\ &= -E_0x + \frac{1}{2}E_0hn_1x^2 + \frac{1}{3}E_0h^2n_2x^3. \end{aligned} \quad (5)$$

We have for ζ , considering eqns. 5 and 4,

$$\begin{aligned} [\zeta]_2 &= \left[\sqrt{\frac{\eta}{\eta_0} - a^2} \right]_2 = \\ &= \left[\sqrt{1 + \delta_K - \frac{2Ze\varphi}{mv_0^2} - a^2} \right]_2 = \\ &= \left[\left(1 + \delta_K - a^2 - \frac{2}{\chi_{e0}} \left(-E_0x + \frac{1}{2}E_0hn_1x^2 + \frac{1}{3}E_0h^2n_2x^3 \right) \right)^{\frac{1}{2}} \right]_2 = \\ &= 1 - \frac{a^2}{2} + \frac{\delta_K}{2} - \frac{\delta_K^2}{8} + h \left(\frac{1}{2}\delta_K - 1 \right) x - \frac{1}{2}h^2(n_1 - 1)(x^2 - y^2), \end{aligned}$$

from where we also obtain

$$\begin{aligned} \left[\frac{1}{\zeta} \right]_2 &= 1 + \frac{a^2}{2} - \frac{\delta_K^2}{2} + \frac{3}{8}\delta_K + h \left(1 - \frac{3}{2}\delta_K \right) x + \\ &+ \frac{1}{2}h^2(3 - n_1)x^2 + \frac{1}{2}h^2(n_1 - 1)y^2. \end{aligned}$$

Applying this to eq. 3, we have

$$\left[\frac{d}{ds} \begin{pmatrix} x \\ a \end{pmatrix} \right]_2 = \begin{pmatrix} a + 2ahx \\ h^2(n_1 - 3)x + h^3(3n_1 + n_2 - 3)x^2 - ha^2 \end{pmatrix}, \quad (6)$$

where we kept only the x - a terms.

Following [4, pp. 107–113], performing a Taylor expansion of the right-hand side, we rewrite the ODE of motion

$$\frac{d}{ds}\vec{r} = \vec{f}(\vec{r}, s)$$

as

$$\frac{d}{ds}\vec{r} = \hat{M}(s)\vec{r} + \sum_{j=2}^{+\infty} \vec{N}_j(s, \vec{r}), \quad (7)$$

where $\hat{M}(s)$ is a square matrix and $\vec{N}_j(s, \vec{r})$ are polynomials of exact order j in the beamline coordinates, coefficients of which generally may depend on s .

In this one-dimensional derivation,

$$\vec{r} = \begin{pmatrix} x \\ a \end{pmatrix}.$$

Comparing eqns. 6 and 7, we find that

$$\hat{M}(s) = \begin{pmatrix} 0 & 1 \\ -3h^2 + h^2n_1 & 0 \end{pmatrix}$$

and

$$\vec{N}_2(s, \vec{r}) = \begin{pmatrix} 2ahx \\ h^3(3n_1 + n_2 - 3)x^2 - ha^2 \end{pmatrix}.$$

The main field of the electrostatic deflector is s -independent, which is why $\hat{M}(s)$ and $\vec{N}_2(s, \vec{r})$ do not depend on s , and we will omit their argument s from here on forward.

First, we solve the equation

$$\frac{d}{ds} \vec{r} = \hat{M} \cdot \vec{r}, \quad (8)$$

which represents a linearization of the ODE of motion.

Because \hat{M} in eq. 8 is s -independent, its solution can be found as [10, pp. 57–58][7, pp. 42–44]

$$\vec{r}_f = \vec{r}(s) = \hat{L}(s) \vec{r}_i = \exp(\hat{M}s) \vec{r}_i, \quad (9)$$

where \vec{r}_i are the initial coordinates $\vec{r}_i = \vec{r}(0)$, \vec{r}_f are the final coordinates $\vec{r}_f = \vec{r}(s)$, and $\exp(\hat{M}s)$ is the matrix exponential

$$\exp(\hat{M}s) = \sum_{j=0}^{\infty} \frac{1}{j!} \hat{M}^j s^j. \quad (10)$$

Indeed, taking the Laplace transform \mathcal{L} of eq. 8 yields [5, p. 193]

$$\mathcal{L} \left(\frac{d}{ds} \vec{r}(s) - \hat{M} \cdot \vec{r}(s) \right) = -\vec{r}(0) + (\tilde{s}\hat{I} - \hat{M}) \vec{R}(\tilde{s}) = 0,$$

from where we have

$$\vec{R}(\tilde{s}) = (\tilde{s}\hat{I} - \hat{M})^{-1} \vec{r}(0),$$

where \tilde{s} is the complex frequency parameter, \hat{I} is the unit matrix, and $\vec{R}(\tilde{s}) = \mathcal{L}(\vec{r}(s))$. The operator

$$\Phi(\hat{M}) = (\tilde{s}\hat{I} - \hat{M})^{-1}$$

is called the resolvent of \hat{M} [8, p. 385]. We note [5, p. 193] that

$$\begin{aligned} \mathcal{L}^{-1} \left[(\tilde{s}\hat{I} - \hat{M})^{-1} \right] &= \mathcal{L}^{-1} \left[\sum_{j=0}^{+\infty} \hat{M}^j \tilde{s}^{-j-1} \right] = \\ &= \sum_{j=0}^{+\infty} \frac{1}{j!} \hat{M}^j s^j = \exp(\hat{M}s). \end{aligned}$$

Conversely, the matrix exponential in eq. 10 can be computed using the inverse Laplace transform \mathcal{L}^{-1} as [5, p. 213]

$$\exp(\hat{M}s) = \mathcal{L}^{-1} \left[(\tilde{s}\hat{I} - \hat{M})^{-1} \right]. \quad (11)$$

We calculate the $\hat{L}(s)$ in eq. 9 using eq. 11:

$$\begin{aligned}
\hat{L}(s) &= \exp(\hat{M}s) \\
&= \mathcal{L}^{-1} \left[\left(\tilde{s}\hat{I} - \hat{M} \right)^{-1} \right] = \\
&= \mathcal{L}^{-1} \left[\left(\tilde{s}\hat{I} - \begin{pmatrix} 0 & 1 \\ -3h^2 + h^2n_1 & 0 \end{pmatrix} \right)^{-1} \right] = \\
&= \mathcal{L}^{-1} \left[\frac{1}{3h^2 - h^2n_1 + \tilde{s}^2} \begin{pmatrix} \tilde{s} & -1 \\ 3h^2 - h^2n_1 & \tilde{s} \end{pmatrix} \right] = \\
&= \begin{pmatrix} \frac{\cos(h\sqrt{3-n_1}s)}{-h\sqrt{3-n_1} \sin(h\sqrt{3-n_1}s)} & \frac{\sin(h\sqrt{3-n_1}s) / (h\sqrt{3-n_1})}{\cos(h\sqrt{3-n_1}s)} \end{pmatrix}.
\end{aligned} \tag{12}$$

The Taylor expansion of $\vec{r}(s)$, considering the solution $\vec{r}(s) = \hat{L}(s) \vec{r}_i$ of the linearized problem, is

$$\vec{r}(s) = \hat{L}(s) \vec{r}_i + \sum_{j=2}^{+\infty} \vec{R}_j(s, \vec{r}_i), \tag{13}$$

where \vec{R}_j is a polynomial of exact order j in the initial conditions, with coefficients as functions of s .

Inserting the Taylor expansion of $\vec{r}(s)$ from eq. 13 into the ODE as written in eq. 7, we obtain

$$\begin{aligned}
\frac{d}{ds} \vec{r}(s) &= \left(\frac{d}{ds} \hat{L}(s) \right) \vec{r}_i + \sum_{j=2}^{+\infty} \frac{d}{ds} \vec{R}_j(s, \vec{r}_i) = \\
&= \hat{M} \hat{L}(s) \vec{r}_i + \hat{M} \sum_{j=2}^{+\infty} \vec{R}_j(s, \vec{r}_i) + \sum_{j=2}^{+\infty} \vec{Q}_j(s, \hat{L}, \vec{R}_j),
\end{aligned} \tag{14}$$

where

$$\vec{Q}_j(s, \hat{L}, \vec{R}_j) = \vec{N}_j(\vec{r}(s, \hat{L}, \vec{R}_j))$$

is a polynomial of exact order j in \vec{r} .

The linear part of eq. 14 gives

$$\frac{d}{ds} \hat{L}(s) = \hat{M} \hat{L}(s), \tag{15}$$

and its parts of orders $j \geq 2$ give

$$\frac{d}{ds} \vec{R}_j(s, \vec{r}_i) = \hat{M} \vec{R}_j(s, \vec{r}_i) + \vec{Q}_j(s, \hat{L}, \vec{R}_j). \tag{16}$$

At $s = 0$, $\vec{r}(s)$ is the initial condition \vec{r}_i . By eq. 13, this yields $\hat{L}(0) = \hat{I}$ and $\vec{R}_j(0, \vec{r}_i) = 0$ for all $j \geq 2$.

The polynomial \vec{Q}_2 will be useful in the calculations below. To find \vec{Q}_2 , we insert the linear solution specified by eqns. 9 and 12 into the ODE expansion from eq. 6, obtaining

$$\begin{aligned}
\frac{d}{ds} x_f &= a_f + 2a_f h x_f = \\
&= (\cos(\xi s) a_i - \xi \sin(\xi s) x_i) [1 + 2h (\cos(\xi s) x_i + \xi^{-1} \sin(\xi s) a_i)]
\end{aligned} \tag{17}$$

and

$$\begin{aligned}
\frac{d}{ds} a_f &= h^2 (n_1 - 3) x_f + h^3 (3n_1 + n_2 - 3) x_f^2 - h a_f^2 = \\
&= h^2 (\cos(\xi s) x_i + \xi^{-1} \sin(\xi s) a_i) [n_1 - 3 + = \\
&\quad + h (3n_1 + n_2 - 3) (\cos(\xi s) x_i + \xi^{-1} \sin(\xi s) a_i)] - \\
&\quad - h (\cos(\xi s) a_i - \xi \sin(\xi s) x_i)^2,
\end{aligned} \tag{18}$$

where $\xi = h\sqrt{3 - n_1}$.

Next, we extract the 2nd order part \vec{Q}_2 from the right-hand sides of eqns. 17 and 18:

$$\begin{aligned}\vec{Q}_2 &= \begin{pmatrix} Q_{2x} \\ Q_{2a} \end{pmatrix} = \\ &= \begin{pmatrix} -h\xi \sin(2\xi s) & \frac{1}{2}h^3(4n_1 + n_2 - 6 + (2n_1 + n_2)\cos(2\xi s)) \\ 2h \cos(2\xi s) & h^3\xi^{-1}(2n_1 + n_2)\sin(2\xi s) \\ h\xi^{-1}\sin(2\xi s) & \frac{1}{2}h^3\xi^{-2}(4n_1 + n_2 - 6 - (2n_1 + n_2)\cos(2\xi s)) \end{pmatrix}^T \begin{pmatrix} x_1^2 \\ x_1 a_1 \\ a_1^2 \end{pmatrix}.\end{aligned}$$

The solution $\vec{r}(s) = \hat{L}(s)\vec{r}_1$ of the linearized problem is the homogeneous solution of eq. 14. To obtain its 2nd order inhomogeneous solution, we perform the ansatz

$$\vec{R}_2(s) = \hat{L}(s)\vec{T}(s). \quad (19)$$

Inserting eq. 19 into eq. 16 with $j = 2$, considering eq. 15, we have

$$\begin{aligned}\frac{d}{ds}\vec{R}_2(s) &= \left(\frac{d}{ds}\hat{L}(s)\right)\vec{T}(s) + \hat{L}(s)\left(\frac{d}{ds}\vec{T}(s)\right) = \\ &= \hat{M}\hat{L}(s)\vec{T}(s) + \hat{L}(s)\left(\frac{d}{ds}\vec{T}(s)\right) = \\ &= \hat{M}\hat{L}(s)\vec{T}(s) + \vec{Q}_2(s),\end{aligned}$$

which yields

$$\hat{L}(s) \cdot \frac{d}{ds}\vec{T}(s) = \vec{Q}_2(s). \quad (20)$$

Solving eq. 20 for $\vec{T}(s)$, we obtain

$$\vec{T}(s) = \begin{pmatrix} T_x \\ T_a \end{pmatrix} = \int_0^s \hat{L}^{-1}\vec{Q}_2 ds, \quad (21)$$

where, considering that $\det(\hat{L}(s)) = 1$,

$$\hat{L}^{-1}(s) = \begin{pmatrix} \cos(\xi s) & -\xi^{-1}\sin(\xi s) \\ \xi \sin(\xi s) & \cos(\xi s) \end{pmatrix}.$$

The integral in eq. 21 is called the aberration integral, and its integrand is called the driving term [4, p. 110].

We obtain for elements T_x and T_a of vector $\vec{T}(s)$:

$$\begin{aligned}T_x &= \int_0^s \left(\hat{L}_{11}^{-1}Q_{2x} + \hat{L}_{12}^{-1}Q_{2a} \right) ds = \\ &= \int_0^s \left[\cos(\xi s) \left(-h\xi \sin(2\xi s) x_1^2 + 2h \cos(2\xi s) x_1 a_1 + h\xi^{-1} \sin(2\xi s) a_1^2 \right) - \right. \\ &\quad \left. -\xi^{-1} \sin(\xi s) \left(h^3(2n_1 + n_2) \left(\frac{1}{2} \cos(2\xi s) (x_1^2 - \xi^{-2} a_1^2) + \xi^{-1} \sin(2\xi s) x_1 a_1 \right) + \right. \right. \\ &\quad \left. \left. + \frac{1}{2} h^3(4n_1 + n_2 - 6) (x_1^2 + \xi^{-2} a_1^2) \right) \right] ds\end{aligned}$$

and

$$\begin{aligned}
T_a &= \int_0^s \left(\hat{L}_{21}^{-1} Q_{2x} + \hat{L}_{22}^{-1} Q_{2a} \right) ds = \\
&= \int_0^s \left[\xi \sin(\xi s) \left(-h\xi \sin(2\xi s) x_1^2 + 2h \cos(2\xi s) x_1 a_1 + h\xi^{-1} \sin(2\xi s) a_1^2 \right) + \right. \\
&\quad + \cos(\xi s) \left(h^3 (2n_1 + n_2) \left(\frac{1}{2} \cos(2\xi s) (x_1^2 - \xi^{-2} a_1^2) + \xi^{-1} \sin(2\xi s) x_1 a_1 \right) + \right. \\
&\quad \left. \left. + \frac{1}{2} h^3 (4n_1 + n_2 - 6) (x_1^2 + \xi^{-2} a_1^2) \right) \right] ds.
\end{aligned}$$

We collect the x_1^2 , $x_1 a_1$, and a_1^2 terms, and we carry out the integration:

$$\begin{aligned}
T_x &= \int_0^s \begin{pmatrix} -\frac{1}{2} h^3 \xi^{-1} (2n_1 + n_2 + (6 + n_2) \cos(2\xi s)) \sin(\xi s) \\ -h^3 \xi^{-2} (2n_1 + n_2 - (6 + n_2) \cos(2\xi s)) \cos(\xi s) \\ -\frac{1}{2} h^3 \xi^{-3} (6n_1 + n_2 - 12 - (6 + n_2) \cos(2\xi s)) \sin(\xi s) \end{pmatrix}^T ds \cdot \begin{pmatrix} x_1^2 \\ x_1 a_1 \\ a_1^2 \end{pmatrix} = \\
&= \begin{pmatrix} -\frac{1}{3} h^3 \xi^{-2} (3(2n_1 + n_2) + (6 + n_2) (2 \cos(\xi s) + \cos(2\xi s))) \sin^2(\frac{1}{2} \xi s) \\ -\frac{1}{3} h^3 \xi^{-3} (6n_1 + n_2 - 12 - (6 + n_2) \cos(2\xi s)) \sin(\xi s) \\ \frac{1}{3} h^3 \xi^{-4} (3(12 - 6n_1 - n_2) + (6 + n_2) (2 \cos(\xi s) + \cos(2\xi s))) \sin^2(\frac{1}{2} \xi s) \end{pmatrix}^T \begin{pmatrix} x_1^2 \\ x_1 a_1 \\ a_1^2 \end{pmatrix} \quad (22)
\end{aligned}$$

and

$$\begin{aligned}
T_a &= \int_0^s \begin{pmatrix} \frac{1}{2} h^3 (6n_1 + n_2 - 12 + (6 + n_2) \cos(2\xi s)) \cos(\xi s) \\ h^3 \xi^{-1} (2n_1 + n_2 + (6 + n_2) \cos(2\xi s)) \sin(\xi s) \\ \frac{1}{2} h^3 \xi^{-2} (2n_1 + n_2 - (6 + n_2) \cos(2\xi s)) \cos(\xi s) \end{pmatrix}^T ds \cdot \begin{pmatrix} x_1^2 \\ x_1 a_1 \\ a_1^2 \end{pmatrix} = \\
&= \begin{pmatrix} \frac{1}{6} h^3 \xi^{-1} (18n_1 + 5n_2 - 24 + (6 + n_2) \cos(2\xi s)) \sin(\xi s) \\ \frac{2}{3} h^3 \xi^{-2} (3(2n_1 + n_2) + (6 + n_2) (2 \cos(\xi s) + \cos(2\xi s))) \sin^2(\frac{1}{2} \xi s) \\ -\frac{1}{6} h^3 \xi^{-3} (12 - 6n_1 - n_2 + (6 + n_2) \cos(2\xi s)) \sin(\xi s) \end{pmatrix}^T \begin{pmatrix} x_1^2 \\ x_1 a_1 \\ a_1^2 \end{pmatrix}. \quad (23)
\end{aligned}$$

An aberration coefficient $(z_i | z_{j_1} \cdots z_{j_n})$ is the partial derivative

$$(z_i | z_{j_1} \cdots z_{j_n}) = \left(\frac{\partial^n (\mathcal{M}(z))_i}{\partial z_{j_1} \cdots \partial z_{j_n}} \right)_{z=0} \quad (24)$$

of the i -th component of the respective transfer map $\mathcal{M}(z)$, where z is a coordinate vector $z = (z_1, \dots, z_{2m})$ and $2m$ is the number of phase space coordinates.

The first order aberrations are the elements of the linear part $\hat{L}(s)$ of the transfer map. Noting that $\vec{R}_2(s)$ is the second order part of the transfer map, the second order aberrations are

$$(z_i | z_{j_1} z_{j_2}) = (1 + \delta_{j_1 j_2}) [(z_{j_1})_i (z_{j_2})_i] \left(\vec{R}_2(s) \right)_i, \quad (25)$$

where $[z]f$ denotes [6, pp. xi, 197] the coefficient of z in f , and where $\delta_{j_1 j_2}$ is the Kronecker delta defined as $\delta_{ij} = 1$ if $i = j$ and $\delta_{ij} = 0$ otherwise.

From eq. 12, we extract the first order aberrations:

$$(x|x) = \hat{L}_{11} = \cos(\xi s), \quad (26a)$$

$$(x|a) = \hat{L}_{12} = \xi^{-1} \sin(\xi s), \quad (26b)$$

$$(a|x) = \hat{L}_{21} = -\xi \sin(\xi s), \quad (26c)$$

$$(a|a) = \hat{L}_{22} = \cos(\xi s). \quad (26d)$$

We obtain [4, pp. 107–113] the second order aberrations using eqns. 12, 22, 23, and 25:

$$\begin{aligned}
(x|xx) &= 2 [x_i^2] \left(\hat{L}_{11}T_x + \hat{L}_{12}T_a \right) = \\
&= -\frac{2}{3}h^3\xi^{-2} (3(2n_1 + n_2) + (6 + n_2)(2\cos(\xi s) + \cos(2\xi s))) \sin^2\left(\frac{1}{2}\xi s\right) \cos(\xi s) + \\
&+ \frac{1}{3}h^3\xi^{-2} (18n_1 + 5n_2 - 24 + (6 + n_2)\cos(2\xi s)) \sin^2(\xi s) = \\
&= \frac{4}{3}h^3\xi^{-2} (9n_1 + 2n_2 - 15 + (6n_1 + n_2 - 12)\cos(\xi s)) \sin^2\left(\frac{1}{2}\xi s\right),
\end{aligned} \tag{27a}$$

$$\begin{aligned}
(x|xa) &= [x_i a_i] \left(\hat{L}_{11}T_x + \hat{L}_{12}T_a \right) = \\
&= -\frac{1}{6}h^3\xi^{-3} (6n_1 + n_2 - 12 - (6 + n_2)\cos(2\xi s)) \sin(2\xi s) + \\
&+ \frac{2}{3}h^3\xi^{-3} (3(2n_1 + n_2) + (6 + n_2)(2\cos(\xi s) + \cos(2\xi s))) \sin^2\left(\frac{1}{2}\xi s\right) \sin(\xi s) = \\
&= \frac{2}{3}h^3\xi^{-3} (3n_1 + n_2 - 3 - (6n_1 + n_2 - 12)\cos(\xi s)) \sin(\xi s),
\end{aligned} \tag{27b}$$

$$\begin{aligned}
(x|aa) &= 2 [a_i^2] \left(\hat{L}_{11}T_x + \hat{L}_{12}T_a \right) = \\
&= \frac{2}{3}h^3\xi^{-4} (3(12 - 6n_1 - n_2) + (6 + n_2)(2\cos(\xi s) + \cos(2\xi s))) \sin^2\left(\frac{1}{2}\xi s\right) \cos(\xi s) - \\
&- \frac{1}{3}h^3\xi^{-4} (12 - 6n_1 - n_2 + (6 + n_2)\cos(2\xi s)) \sin^2(\xi s) = \\
&= -\frac{4}{3}h^3\xi^{-4} (3 - 3n_1 - n_2 + (6n_1 + n_2 - 12)\cos(\xi s)) \sin^2\left(\frac{1}{2}\xi s\right),
\end{aligned} \tag{27c}$$

$$\begin{aligned}
(a|xx) &= 2 [x_i^2] \left(\hat{L}_{21}T_x + \hat{L}_{22}T_a \right) = \\
&= \frac{2}{3}h^3\xi^{-1} (3(2n_1 + n_2) + (6 + n_2)(2\cos(\xi s) + \cos(2\xi s))) \sin^2\left(\frac{1}{2}\xi s\right) \sin(\xi s) + \\
&+ \frac{1}{6}h^3\xi^{-1} (18n_1 + 5n_2 - 24 + (6 + n_2)\cos(2\xi s)) \sin(2\xi s) = \\
&= \frac{2}{3}h^3\xi^{-1} (3n_1 + n_2 - 3) (\sin(\xi s) + \sin(2\xi s)),
\end{aligned} \tag{27d}$$

$$\begin{aligned}
(a|xa) &= [x_i a_i] \left(\hat{L}_{21}T_x + \hat{L}_{22}T_a \right) = \\
&= \frac{1}{3}h^3\xi^{-2} (6n_1 + n_2 - 12 - (6 + n_2)\cos(2\xi s)) \sin^2(\xi s) + \\
&+ \frac{2}{3}h^3\xi^{-2} (3(2n_1 + n_2) + (6 + n_2)(2\cos(\xi s) + \cos(2\xi s))) \sin^2\left(\frac{1}{2}\xi s\right) \cos(\xi s) = \\
&= -\frac{2}{3}h^3\xi^{-2} (3n_1 + n_2 - 3) (\cos(2\xi s) - \cos(\xi s)),
\end{aligned} \tag{27e}$$

$$\begin{aligned}
(a|aa) &= 2 [a_i^2] \left(\hat{L}_{21}T_x + \hat{L}_{22}T_a \right) = \\
&= -\frac{2}{3}h^3\xi^{-3} (3(12 - 6n_1 - n_2) + (6 + n_2)(2\cos(\xi s) + \cos(2\xi s))) \sin^2\left(\frac{1}{2}\xi s\right) \sin(\xi s) - \\
&- \frac{1}{6}h^3\xi^{-3} (12 - 6n_1 - n_2 + (6 + n_2)\cos(2\xi s)) \sin(2\xi s) = \\
&= -\frac{2}{3}h^3\xi^{-3} (15 - 9n_1 - 2n_2 + 2(3n_1 + n_2 - 3)\cos(\xi s)) \sin(\xi s),
\end{aligned} \tag{27f}$$

where $\xi = h\sqrt{3 - n_1}$ and $[z]f$ denotes the coefficient of z in f . The arc length is $s = \alpha h^{-1} = \alpha R_0$, where α is the central angle and R_0 is the reference orbit radius specifying the deflector geometry.

The aberration formulas in eqns. 26 and 27 remain valid for $n_1 > 3$, entailing trivial computations with complex numbers or an equivalent restatement in terms of hyperbolic trigonometric functions \sinh and \cosh . The same derivation method as used above is applicable to the special case $n_1 = 3$, with the difference that $n_1 = 3$ is assumed at the beginning of the derivation. Without proof, we will state the analytic aberration formulas for the special case $n_1 = 3$: $(x|x) = 1$, $(x|a) = s$, $(x|xx) = h^3(6 + n_2)s^2$, $(x|xa) = 2hs + h^3(6 + n_2)s^3/3$, $(x|aa) = hs^2(6 + h^2(6 + n_2)s^2)/6$, $(a|x) = 0$, $(a|a) = 1$, $(a|xx) = 2h^3(6 + n_2)s$, $(a|xa) = h^3(6 + n_2)s^2$, and $(a|aa) = 2hs(h^2(6 + n_2)s^2 - 3)/3$.

4 Comparison of Aberration Calculations with *COSY INFINITY*

We composed a C program called `edabrt` that calculates first and second order aberrations of an electrostatic deflector in the x - a plane using the formulas derived in sec. 3. This program is listed in app. E.

In this section, we will compare the aberrations computed using `edabrt` and *COSY INFINITY* for the cases of an electrostatic spherical deflector and an electrostatic cylindrical deflector. In both cases, the deflectors are specified by the reference orbit radius $R_0 = 1$ m and central angle $\alpha = 45^\circ$. We remark that the applicability of the derived aberration formulas and the program `edabrt` is not limited to electrostatic spherical and cylindrical deflectors; any inhomogeneity indices n_1 and n_2 can be used.

In *COSY INFINITY*, the particle beam was defined by kinetic energy 1 MeV, mass 1 amu, and charge $1e$; however, as eqns. 26–27 show, these settings have no effect on the aberrations as long as the reference orbit radius is kept the same by adjusting the voltages of the inner and outer shells of the electrostatic deflector. Non-relativistic equations of motion were used in *COSY INFINITY* for commensurability with the non-relativistic aberration formulas in `edabrt`. In *COSY INFINITY*, aberrations are obtained from the elements of the DA transfer map, which in turn is calculated by performing numerical DA integration of the equations of motion.

The Jacobian $M = \text{Jac}(\mathcal{M})$ of the transfer map \mathcal{M} of any Hamiltonian system satisfies the symplecticity condition $M \cdot \hat{J} \cdot M^T = \hat{J}$ [2, pp. 155–159], where

$$\hat{J} = \begin{pmatrix} 0 & I_m \\ -I_m & 0 \end{pmatrix},$$

I_m is an $m \times m$ identity matrix, m is the phase space dimension, and the canonical position (q_1, \dots, q_m) and canonical momentum (p_1, \dots, p_m) coordinates are ordered as $z = (q_1, \dots, q_m, p_1, \dots, p_m)$.

In each aberrations calculation case, we computed deviations

$$\begin{aligned} g_1 &= (x|x)(a|a) - (a|x)(x|a) - 1 = 0, \\ g_2 &= (x|x)(a|xa) - (a|x)(x|xa) + (x|xx)(a|a) - (a|xx)(x|a) = 0, \\ g_3 &= (x|x)(a|aa) - (a|x)(x|aa) + (x|xa)(a|a) - (a|xa)(x|a) = 0. \end{aligned} \tag{28}$$

from the conditions of symplecticity [2, pp. 155–159][12].

For comparison with the report [1], app. A lists first and second order aberrations computed in polar laboratory coordinates (1) by integration of the ODEs of motion using a 4th order Runge–Kutta integrator for the electrostatic spherical and cylindrical deflectors and (2) by computing the Kepler theory transition matrix with elements as Lagrange coefficients for the electrostatic spherical deflector.

4.1 Electrostatic Spherical Deflector

4.1.1 Calculation using edabrt

The aberrations of the electrostatic spherical deflector calculated by the C program `edabrt` using the derived analytic formulas are as follows.

(x ...)				
I	COEFFICIENT	ORDER	EXPONENTS	
1	7.071067811865476e-01	1	1	0
2	7.071067811865475e-01	1	0	1
3	-1.0000000000000000e+00	2	2	0
4	1.0000000000000000e+00	2	1	1
5	4.142135623730951e-01	2	0	2

(a ...)				
I	COEFFICIENT	ORDER	EXPONENTS	
1	-7.071067811865475e-01	1	1	0
2	7.071067811865476e-01	1	0	1
3	-1.414213562373095e+00	2	0	2

The deviations g_1 , g_2 , and g_3 from the conditions of symplecticity listed in eq. 28 in this case were

$$\begin{aligned}
 g_1 &= 0, \\
 g_2 &= -2.22044604925031 \times 10^{-16}, \\
 g_3 &= 1.11022302462516 \times 10^{-16}.
 \end{aligned}$$

The program `edabrt` is listed in app. E.

4.1.2 Calculation using COSY INFINITY

The aberrations of *COSY INFINITY*'s built-in electrostatic spherical deflector element ESP, obtained using non-relativistic equations of motion, are as follows.

(x ...)				
I	COEFFICIENT	ORDER	EXPONENTS	
1	0.7071067811865475	1	1	0
2	0.7071067811865475	1	0	1
3	-.9999999999999999	2	2	0
4	1.0000000000000000	2	1	1
5	0.4142135623730951	2	0	2

(a ...)				
I	COEFFICIENT	ORDER	EXPONENTS	
1	-.7071067811865475	1	1	0
2	0.7071067811865475	1	0	1
3	-1.414213562373095	2	0	2

The deviations g_1 , g_2 , and g_3 from the conditions of symplecticity listed in eq. 28 in this case were

$$\begin{aligned}
 g_1 &= -0.2220446049250313 \times 10^{-15}, \\
 g_2 &= 0.2220446049250313 \times 10^{-15}, \\
 g_3 &= 0.3330669073875470 \times 10^{-15}.
 \end{aligned}$$

The *COSY INFINITY* code used to calculate these aberrations is listed in app. C.

4.2 Electrostatic Cylindrical Deflector

4.2.1 Calculation using edabrt

The aberrations of the electrostatic cylindrical deflector calculated by the program `edabrt` using the derived analytic formulas are as follows.

(x ...)				
I	COEFFICIENT	ORDER	EXPONENTS	
1	4.440158403262133e-01	1	1	0
2	6.335810656653995e-01	1	0	1
3	-2.058644564816544e+00	2	2	0
4	4.452197131126671e-01	2	1	1
5	1.953460428975922e-01	2	0	2

(a ...)				
I	COEFFICIENT	ORDER	EXPONENTS	
1	-1.267162131330799e+00	1	1	0
2	4.440158403262133e-01	1	0	1
3	-7.974807494918663e-01	2	2	0
4	-3.499052358016756e-01	2	1	1
5	-1.502002822250266e+00	2	0	2

The deviations g_1 , g_2 , and g_3 from the conditions of symplecticity listed in eq. 28 in this case were

$$\begin{aligned}
 g_1 &= 0, \\
 g_2 &= 0, \\
 g_3 &= -5.55111512312578 \times 10^{-17}.
 \end{aligned}$$

The program `edabrt` is listed in app. E.

4.2.2 Calculation using COSY INFINITY

The aberrations of *COSY INFINITY*'s built-in electrostatic cylindrical deflector element ECL, obtained using non-relativistic equations of motion, are as follows.

(x ...)				
I	COEFFICIENT	ORDER	EXPONENTS	
1	0.4440158403262133	1	1	0
2	0.6335810656653997	1	0	1
3	-2.058644564816545	2	2	0
4	0.4452197131126671	2	1	1
5	0.1953460428975922	2	0	2

(a ...)				
I	COEFFICIENT	ORDER	EXPONENTS	
1	-1.267162131330799	1	1	0
2	0.4440158403262133	1	0	1
3	-.7974807494918666	2	2	0
4	-.3499052358016756	2	1	1
5	-1.502002822250265	2	0	2

The deviations g_1 , g_2 , and g_3 from the conditions of symplecticity listed in eq. 28 in this case were

$$\begin{aligned}g_1 &= 0.4440892098500626 \times 10^{-15}, \\g_2 &= 0.2220446049250313 \times 10^{-15}, \\g_3 &= 0.2498001805406602 \times 10^{-15}.\end{aligned}$$

The *COSY INFINITY* code used to calculate these aberrations is listed in app. D.

5 Conclusion

We derived analytic formulas for first and second order aberrations of an electrostatic deflector in the x - a plane using an order-by-order perturbation method. We performed the derivations in *COSY INFINITY*'s Frenet–Serret beamline coordinate system.

We implemented a C program called `edabrt` that uses these analytic formulas to calculate the first and second order aberrations on an electrostatic deflector. We calculated the aberrations for the cases of an electrostatic spherical deflector and an electrostatic cylindrical deflector using `edabrt`, and we compared the results with *COSY INFINITY*'s elements ESP and ECL, respectively. The comparison shows good agreement between `edabrt` and *COSY INFINITY*.

The deviations from the conditions of symplecticity g_1 , g_2 , and g_3 in *COSY INFINITY* calculations were all of the order $\sim 3 \times 10^{-16}$; for aberrations computed using `edabrt`, they ranged from 0 to $\sim 10^{-16}$.

The comparison cross-validates the analytic aberration formulas derived in this report, the C program `edabrt`, and *COSY INFINITY*. In particular, it illustrates the high accuracy of numerical DA integration performed in *COSY INFINITY* for calculation of transfer maps of particle optical elements.

The electrostatic deflector aberrations presented in this report are in a good agreement with the results of our earlier report [1], as comparing the calculation results in sec. 4 with those in app. A shows.

References

- [1] Eremey Valetov, Martin Berz, and Kyoko Makino. 2017. Direct Calculation of the Transfer Map of Electrostatic Deflectors, and Comparison with the Codes COSY INFINITY and GIOS. MSU report MSUHEP 171023. Department of Physics and Astronomy, Michigan State University, East Lansing, MI 48824.
- [2] Martin Berz. 1999. *Modern Map Methods in Particle Beam Physics. Advances in Imaging and Electron Physics*. Academic Press, San Diego, CA, 1–318. ISBN: 0-12-014750-5.
- [3] Martin Berz and Kyoko Makino. 2017. COSY INFINITY 10.0 Beam Physics Manual. MSU report MSUHEP 151103-rev. Department of Physics and Astronomy, Michigan State University, East Lansing, MI 48824.
- [4] Martin Berz, Kyoko Makino, and Weishi Wan. 2014. *An Introduction to Beam Physics*. English. CRC Press, London. ISBN: 0-75-030263-1.
- [5] Richard C. Dorf and Robert H. Bishop. 2017. *Modern Control Systems*. (13th edition). Pearson, Harlow, England. ISBN: 978-0-13-440762-3.
- [6] Ronald L. Graham, Donald E. Knuth, and Oren Patashnik. 1994. *Concrete Mathematics: A Foundation for Computer Science*. (2nd edition). Addison–Wesley, Reading, MA. ISBN: 0-20-155802-5.
- [7] Roger Grimshaw. 1990. *Nonlinear Ordinary Differential Equations*. Blackwell Scientific Publications, Oxford. ISBN: 0-84-938607-1.
- [8] John K. Hunter and Bruno Nachtergaele. 2001. *Applied Analysis*. World Scientific, River Edge, NJ. ISBN: 9-81-0-24191-7.
- [9] Kyoko Makino and Martin Berz. 2006. COSY INFINITY Version 9. *Nucl. Instr. Meth. Phys. Res. A*, 558, 1, 346–350. ISSN: 0168-9002.

- [10] Gerald Teschl. 2012. *Ordinary Differential Equations and Dynamical Systems*. English. *Graduate Studies in Mathematics*. Volume 140. American Mathematical Society, Providence, RI. ISBN: 0-82-188328-3.
- [11] Eremey Valetov. 2017. Field Modeling, Symplectic Tracking, and Spin Decoherence for EDM and Muon $g-2$ Lattices. Ph.D. thesis. Fermilab report FERMILAB-THESIS-2017-21. Michigan State University, East Lansing, MI.
- [12] Hermann Wollnik and Martin Berz. 1985. Relations Between Elements of Transfer Matrices Due to the Condition of Symplecticity. *Nucl. Instr. Meth. Phys. Res. A*, 238, 1, 127–140. ISSN: 0168-9002.

A Electrostatic Deflector Aberrations Calculated in Polar Laboratory Coordinates

In our earlier report [1], among other things, we computed the transfer maps of 45° sectors of electrostatic spherical and cylindrical deflectors of $R_0 = 1$ m reference radius in polar laboratory coordinates by (1) integrating the ODEs of motion using a 4th order Runge–Kutta integrator and by (2) computing analytically the Kepler theory transition matrix with elements as Lagrange coefficients. Here, we list the first and second order aberrations obtained from the respective transfer map elements using eq. 24.

A.1 Electrostatic Spherical Deflector Aberrations Obtained by Integrating the ODEs in Laboratory Coordinates

The aberrations of the electrostatic spherical deflector obtained from its transfer map computed in [1] by integration of the ODEs of motion in polar laboratory coordinates using a 4th order Runge–Kutta integrator are as follows.

(x ...)					
I	COEFFICIENT	ORDER	EXPONENTS		
1	0.7071067811865617	1	1	0	0
2	0.7071067811865485	1	0	1	0
3	-.9999999999996280	2	2	0	0
4	0.999999999999932	2	1	1	0
5	0.4142135623730224	2	0	2	0

(a ...)					
I	COEFFICIENT	ORDER	EXPONENTS		
1	-.7071067811864190	1	1	0	0
2	0.7071067811865486	1	0	1	0
3	-1.414213562373043	2	0	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. 28 in this case were

$$\begin{aligned}
 g_1 &= -0.7949196856316121 \times 10^{-13}, \\
 g_2 &= -0.2514743315828557 \times 10^{-12}, \\
 g_3 &= -0.1188442801551391 \times 10^{-12}.
 \end{aligned}$$

The 4th order Runge–Kutta integrator code used to calculate the transfer map corresponding to these aberrations is listed in [1, app. A].

A.2 Electrostatic Spherical Deflector Aberrations Obtained Using Lagrange Coefficients in Laboratory Coordinates

The aberrations of the electrostatic spherical deflector obtained from its transfer map computed in [1] using analytic Lagrange coefficients of Kepler theory are as follows.

(x ...)					
I	COEFFICIENT	ORDER	EXPONENTS		
1	0.7071067811865475	1	1	0	0
2	0.7071067811865475	1	0	1	0
3	-1.0000000000000000	2	2	0	0
4	1.0000000000000000	2	1	1	0
5	0.4142135623730949	2	0	2	0
(a ...)					
I	COEFFICIENT	ORDER	EXPONENTS		
1	-.7071067811865475	1	1	0	0
2	0.7071067811865476	1	0	1	0
3	-1.414213562373095	2	0	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. 28 in this case were

$$\begin{aligned}
 g_1 &= -0.1110223024625157 \times 10^{-15}, \\
 g_2 &= 0.3119771259853192 \times 10^{-16}, \\
 g_3 &= 0.1110223024625157 \times 10^{-15}.
 \end{aligned}$$

The *COSY INFINITY* code used to calculate the transfer map corresponding to these aberrations is listed in [1, app. C].

A.3 Electrostatic Cylindrical Deflector Aberrations Obtained by Integrating the ODEs in Laboratory Coordinates

The aberrations of the electrostatic cylindrical deflector obtained from its transfer map computed in [1] by integration of the ODEs of motion in polar laboratory coordinates using a 4th order Runge–Kutta integrator are as follows.

(x ...)						
I	COEFFICIENT	ORDER	EXPONENTS			
1	0.4440158403262955	1	1	0	0	
2	0.6335810656653742	1	0	1	0	
3	-2.058644564816371	2	2	0	0	
4	0.4452197131126942	2	1	1	0	
5	0.1953460428975713	2	0	2	0	
(a ...)						
I	COEFFICIENT	ORDER	EXPONENTS			
1	-1.267162131330700	1	1	0	0	
2	0.4440158403261798	1	0	1	0	
3	-.7974807494918986	2	2	0	0	
4	-.3499052358015541	2	1	1	0	
5	-1.502002822250153	2	0	2	0	

The deviations g_1 , g_2 , and g_3 from the conditions of symplecticity listed in eq. 28 in this case were

$$\begin{aligned}
 g_1 &= -0.7316369732279782 \times 10^{-13}, \\
 g_2 &= 0.1615374500829603 \times 10^{-12}, \\
 g_3 &= -0.2078615057854449 \times 10^{-12}.
 \end{aligned}$$

Refer to [1, app. A] for the 4th order Runge–Kutta integrator code used to calculate the transfer map corresponding to these aberrations.

B Inhomogeneity Coefficients of Electrostatic Spherical and Cylindrical Deflectors

According to Gauss' law, the radial component E_r of the electrostatic field of an electrostatic spherical deflector is

$$E_r(r) = \frac{Q}{4\pi\epsilon_0 r^2},$$

where Q is the charge of the deflector's inner shell and r is the radius from the deflector's center of curvature. The tangential components of the electrostatic field are zero by spherical symmetry.

Similarly, by Gauss' law, the electrostatic field of a cylindrical deflector is subject to cylindrical symmetry, and its radial component E_r is

$$E_r(r) = \frac{\lambda}{2\pi\epsilon_0 r},$$

where λ is the linear charge density of the deflector's inner shell.

The beamline coordinate x in both cases is $x = r - R_0$, where R_0 is the reference radius. Performing the Taylor series expansion of $E_x(x)$, we have

$$E_x(x) = \frac{Q}{4\pi\epsilon_0 R_0^2} \left(1 - 2\frac{x}{R_0} + 3\left(\frac{x}{R_0}\right)^2 \right) + O(x^3)$$

for the electrostatic spherical deflector and

$$E_x(x) = \frac{\lambda}{2\pi\epsilon_0 R_0} \left(1 - \frac{x}{R_0} + \left(\frac{x}{R_0}\right)^2 \right) + O(x^3)$$

for the electrostatic cylindrical deflector.

Comparing these results to the expansion

$$E(x) = E_0 \left[1 - \sum_{j=1}^N n_j \left(\frac{x}{R_0}\right)^j \right] + O(x^{N+1})$$

from eq. 1, we obtain the inhomogeneity indices (n_1, n_2) as $(2, -3)$ for an electrostatic spherical deflector and $(1, -1)$ for an electrostatic cylindrical deflector.

C *COSY INFINITY* Code for Calculation of Aberrations of an Electrostatic Spherical Deflector

The following is a *COSY INFINITY* code that calculates the first and second order aberrations in the x - a plane of a 45° sector of an electrostatic spherical deflector. The voltages on the plates of the electrostatic deflector are implicitly set to result in a circular reference orbit of radius $R_0 = 1$ m. The built-in particle optical element **ESP** is used to model the electrostatic spherical deflector. Non-relativistic equations of motion are used.

```
INCLUDE 'COSYNR' ; {Non-relativistic version of COSY.FOX}

PROCEDURE RUN ;

VARIABLE M 4000 5 ;
VARIABLE J 1 ;

OV 2 1 0 ;
RP 1 1 1 ;

UM ;
ESP 1 45 10 ; {Electrostatic spherical deflector}

LOOP J 1 2 ;
    M(J) := MAP(J) + ME(J,11)*DA(1)*DA(1) + ME(J,22)*DA(2)*DA(2) ;
ENDLOOP ;

WRITE 6 'ABERRATIONS OF COSY INFINITY' 'S ESP ELEMENT' ;
WRITE 6 '(x|...)' M(1) '(a|...)' M(2) ;

ENDPROCEDURE ; RUN ; END ;
```

D *COSY INFINITY* Code for Calculation of Aberrations of an Electrostatic Cylindrical Deflector

The following is a *COSY INFINITY* code that calculates the first and second order aberrations in the x - a plane of a 45° sector of an electrostatic cylindrical deflector. The voltages on the plates of the electrostatic deflector are implicitly set to result in a circular reference orbit of radius $R_0 = 1$ m. The built-in particle optical element ECL is used to model the electrostatic cylindrical deflector. Non-relativistic equations of motion are used.

```
INCLUDE 'COSYNR' ; {Non-relativistic version of COSY.FOX}

PROCEDURE RUN ;

VARIABLE M 4000 5 ;
VARIABLE J 1 ;

OV 2 1 0 ;
RP 1 1 1 ;

UM ;
ECL 1 45 10 ; {Electrostatic cylindrical deflector}

LOOP J 1 2 ;
    M(J) := MAP(J) + ME(J,11)*DA(1)*DA(1) + ME(J,22)*DA(2)*DA(2) ;
ENDLOOP ;

WRITE 6 'ABERRATIONS OF COSY INFINITY' 'S ECL ELEMENT' ;
WRITE 6 '(x|...)' M(1) '(a|...)' M(2) ;

ENDPROCEDURE ; RUN ; END ;
```

E C Program edabrt for Calculation of Aberrations of an Electrostatic Deflector

The following is a C program titled `edabrt` that calculates the first and second order aberrations in the x - a plane of an electrostatic deflector. The deflector is specified by the reference orbit radius R_0 , central angle α , and inhomogeneity indices n_1 and n_2 . Calculations are performed using the analytic aberration formulas derived in this report based on non-relativistic equations of motion.

```

/*****
*                               edabrt: Electrostatic Deflector Aberrations                               *
*                               E. Valetov & M. Berz                                           *
*                               Created 25-Jan-2018                                           *
*                               Email: valetove@msu.edu                                         *
*                                                                                               *
* This program computes the first and second order aberrations of an                       *
* electrostatic deflector in the x-a plane using exact analytic formulas.                   *
* Please refer to the following report for a derivation of these formulas:                 *
* E. Valetov and M. Berz                                                                    *
* Derivation of Analytic Formulas for Electrostatic Deflector Aberrations,                *
* and Comparison with the Code COSY INFINITY                                               *
* MSUHEP-180212, Michigan State University (2018)                                         *
*****/

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>

double calculate_aberration(double radius, double ang, double n1, double n2,
                           int index1, int index2) {
    double s = radius * (M_PI / 180) * ang; // arc length
    double h = 1 / radius; // curvature radius
    if (n1 < 3) {
        double s3n1 = h * sqrt(3 - n1);
        switch (index1) {
            case 1:
                switch (index2) {
                    case 10:
                        return cos(s3n1 * s);
                    case 01:
                        return sin(s3n1 * s) / s3n1;
                    case 20:
                        return -4 * h * (9 * n1 + 2 * n2 - 15 +
                                         (6 * n1 + n2 - 12) * cos(s3n1 * s)) *
                               pow(sin(s3n1 * s / 2), 2) / (3 * (n1 - 3));
                    case 11:
                        return -2 * pow(h, 3) * (3 - 3 * n1 - n2 +
                                                  (6 * n1 + n2 - 12) *
                                                  cos(s3n1 * s)) *
                               sin(s3n1 * s) /
                               (3 * pow(h * h * (3 - n1), 1.5));
                    case 02:

```

```

        return -4 * (3 - 3 * n1 - n2 +
                    (6 * n1 + n2 - 12) * cos(s3n1 * s)) *
                    pow(sin(s3n1 * s / 2), 2) /
                    (3 * h * pow((n1 - 3), 2));
    default:
        return 0;
    }
case 2:
    switch (index2) {
    case 10:
        return -s3n1 * sin(s3n1 * s);
    case 01:
        return cos(s3n1 * s);
    case 20:
        return 2 * pow(h, 3) * (3 * n1 + n2 - 3) *
                (sin(s3n1 * s) + sin(2 * s3n1 * s)) /
                (3 * s3n1);
    case 11:
        return -4 * h * (3 * n1 + n2 - 3) *
                (1 + 2 * cos(s3n1 * s)) *
                pow(sin(s3n1 * s / 2), 2) /
                (3 * (n1 - 3));
    case 02:
        return -2 * pow(h, 3) * (15 - 9 * n1 - 2 * n2 +
                                2 * (3 * n1 + n2 - 3) *
                                cos(s3n1 * s)) *
                sin(s3n1 * s) /
                (3 * pow(h * h * (3 - n1), 1.5));
    default:
        return 0;
    }
default:
    return 0;
}
} else if (n1 > 3) {
    double s3n1 = h * sqrt(n1 - 3);
    switch (index1) {
    case 1:
        switch (index2) {
        case 10:
            return cosh(s3n1 * s);
        case 01:
            return -sinh(s3n1 * s) / s3n1;
        case 20:
            return h * (4 * (3 * n1 + n2 - 3) * cosh(s3n1 * s) -
                        3 * (4 * n1 + n2 - 6) *
                        (2 * cosh(2 * s3n1 * s) - 1) -
                        (6 + n2) * cosh(4 * s3n1 * s)) /
                    (6 * (n1 - 3));
        case 11:
            return (4 * (3 * n1 + n2 - 3) * sinh(s3n1 * s) -

```

```

        (6 + n2) * sinh(4 * s3n1 * s)) /
        (6 * pow(n1 - 3, 1.5));
    case 02:
        return -(4 * (9 * n1 + 2 * n2 - 15) * cosh(s3n1 * s) -
            3 * (4 * n1 + n2 - 6) *
            (2 * cosh(2 * s3n1 * s) + 1) +
            (6 + n2) * cosh(4 * s3n1 * s)) /
            (6 * h * pow(n1 - 3, 2));
    default:
        return 0;
}
case 2:
    switch (index2) {
        case 10:
            return s3n1 * sinh(s3n1 * s);
        case 01:
            return cosh(s3n1 * s);
        case 20:
            return 2 * h * h * h * (3 * n1 + n2 - 3) *
                (sinh(s3n1 * s) + sinh(2 * s3n1 * s)) /
                (3 * s3n1);
        case 11:
            return 2 * h * (3 * n1 + n2 - 3) *
                (cosh(2 * s3n1 * s) - cosh(s3n1 * s)) /
                (3 * (n1 - 3));
        case 02:
            return 2 * (15 - 9 * n1 - 2 * n2 +
                2 * (3 * n1 + n2 - 3) *
                cosh(s3n1 * s)) *
                sinh(s3n1 * s) /
                (3 * pow(n1 - 3, 1.5));
        default:
            return 0;
    }
default:
    return 0;
}
} else {
    switch (index1) {
        case 1:
            switch (index2) {
                case 10:
                    return 1;
                case 01:
                    return s;
                case 20:
                    return h * h * h * (6 + n2) * s * s;
                case 11:
                    return 2 * h * s +
                        h * h * h * (6 + n2) * s * s * s / 3;
                case 02:

```

```

        return h * s * s * (6 + h * h * (6 + n2) * s * s) / 6;
    default:
        return 0;
    }
    case 2:
        switch (index2) {
            case 10:
                return 0;
            case 01:
                return 1;
            case 20:
                return 2 * h * h * h * (6 + n2) * s;
            case 11:
                return h * h * h * (6 + n2) * s * s;
            case 02:
                return 2 * h * s * (h * h * (6 + n2) * s * s - 3) / 3;
            default:
                return 0;
        }
    default:
        return 0;
    }
}

int print_aberrations(double radius, double ang, double n1, double n2,
                    int index1) {
    double aberration;
    int max_order = 2;
    int number_of_variables = 2;
    int first_non_zero = 0;
    char index_string[20];
    int counter = 1;
    for (int order = 1; order <= max_order; order++) {
        int index_array[number_of_variables + 1];
        for (int j = 0; j <= number_of_variables; j++)
            index_array[j] = 0;
        index_array[0] = order;
        while (index_array[number_of_variables] == 0) {
            int index2 = 0;
            for (int j = 0; j <= number_of_variables - 1; j++)
                index2 = index2 * 10 + index_array[j];
            aberration = calculate_aberration(radius, ang, n1, n2, index1,
                                           index2);
            sprintf(index_string, "%i", index_array[0]);
            if (number_of_variables > 1)
                for (int j = 1; j <= number_of_variables - 1; j++) {
                    snprintf(index_string, sizeof index_string, "%s %i",
                             index_string, index_array[j]);
                }
            if (aberration != 0) {

```



```

        if (counter == 1)
            printf("      I COEFFICIENT          ORDER EXPONENTS\n");
        printf("      %i % 16.15le  %i  %s\n", counter, aberration,
            order, index_string);
        counter++;
    }
    for (int j = 0; j <= number_of_variables - 1; j++) {
        if (index_array[j] != 0) {
            first_non_zero = j;
            break;
        }
    }
    if (index_array[first_non_zero] == order) {
        index_array[0] = order - 1;
        index_array[first_non_zero + 1] = 1;
    } else {
        index_array[first_non_zero]--;
        index_array[first_non_zero + 1]++;
    }
}
}
if (counter == 1)
    printf("      ALL COMPONENTS ZERO\n");
printf("      _____\n");
}

int invalid_option_exit(char *option) {
    printf("edabrt: invalid option — %s\n", option);
    printf("Try 'edabrt —help' for more information.\n");
    exit(-1);
}

int main(int argc, char **argv) {
    double r; // reference radius
    double ang; // central angle
    double n1, n2; // first and second order inhomogeneity coefficients
    double dummy;
    int read_code;
    printf("_____ \n");
    printf("      edabrt: Electrostatic Deflector Aberrations      \n");
    printf("      E. Valetov & M. Berz                               \n");
    printf("      Created 25-Jan-2018                                 \n");
    printf("      Email: valetove@msu.edu                             \n");
    printf("_____ \n");
    if (argc >= 2)
        if (strcmp(argv[1], "-h") == 0 || strcmp(argv[1], "/h") == 0 ||
            strcmp(argv[1], "--help") == 0) {
            printf("\nThis program computers the first and second order ");
            printf("aberrations of an electrostatic deflector\n");
            printf("in the horizontal x-a plane using exact analytic ");
            printf("formulas.\n");
        }
}

```

```

printf("\nINTERACTIVE MODE\n");
printf("Run the program and follow the prompts to specify the ");
printf("electrostatic deflector parameters.\n");
printf("\nCOMMAND-LINE ARGUMENTS\n");
printf("Electrostatic deflector parameters may be optionally ");
printf("supplied using the command line:\n");
printf("edabrt [r ang n1 n2] [--help]\n");
printf("    r          Reference orbit radius in meters\n");
printf("    ang        Central angle spanning the deflector ");
printf("in degrees\n");
printf("    n1         First order electrostatic field ");
printf("inhomogeneity coefficient\n");
printf("    n2         Second order electrostatic field ");
printf("inhomogeneity coefficient\n");
printf("    --help    This information\n");
exit(0);
}
switch (argc) {
case 5:
    if (sscanf(argv[1], "%lf", &r) != 1) invalid_option_exit(argv[1]);
    if (r <= 0) {
        printf("\nedabrt: supplied radius r is not positive\n");
        exit(-1);
    }
    if (sscanf(argv[2], "%lf", &ang) != 1)
        invalid_option_exit(argv[2]);
    if (sscanf(argv[3], "%lf", &n1) != 1)
        invalid_option_exit(argv[3]);
    if (sscanf(argv[4], "%lf", &n2) != 1)
        invalid_option_exit(argv[4]);
    printf("\nReference radius r = % 16.15le m\n", r);
    printf("Central angle ang = % 16.15le°\n", ang);
    printf("1st order inhomogeneity coefficient n1 = % 16.15le\n",
        n1);
    printf("2nd order inhomogeneity coefficient n2 = % 16.15le\n",
        n2);
    break;
case 1:
    printf("\n");
    do {
        fseek(stdin, 0, SEEK_END);
        printf("Please enter the reference orbit radius r ");
        printf("in [m].\n> ");
        stdin = freopen(NULL, "r", stdin);
        read_code = scanf("%lf", &r);
        if (read_code != 1)
            printf("Not a numerical value.\n");
        else if (r <= 0) {
            printf("The radius must be positive.\n");
            read_code = 0;
        }
    }
}

```

```

    } while (read_code != 1);
do {
    printf("Please enter the central angle ang spanning the ");
    printf("deflector in [°].\n> ");
    stdin = freopen(NULL, "r", stdin);
    read_code = scanf("%lf", &ang);
    if (read_code != 1)
        printf("Not a numerical value.\n");
} while (read_code != 1);
do {
    printf("Please enter the first order inhomogeneity ");
    printf("coefficient n1.\n> ");
    stdin = freopen(NULL, "r", stdin);
    read_code = scanf("%lf", &n1);
    if (read_code != 1)
        printf("Not a numerical value.\n");
} while (read_code != 1);
do {
    printf("Please enter the first order inhomogeneity ");
    printf("coefficient n2.\n> ");
    stdin = freopen(NULL, "r", stdin);
    read_code = scanf("%lf", &n2);
    if (read_code != 1)
        printf("Not a numerical value.\n");
} while (read_code != 1);
break;
default:
    printf("\n");
    for (int j = 1; j < argc; j++) {
        if (sscanf(argv[j], "%lf", &dummy) != 1) {
            printf("edabrt: invalid option -- %s\n", argv[j]);
            printf("Try 'edabrt --help' for more information.\n");
            exit(-1);
        }
    }
    printf("edabrt: 4 numerical arguments expected, %i supplied\n",
           argc - 1);
    printf("Try 'edabrt --help' for more information.\n");
    exit(-1);
}
printf("\nFirst and second order aberrations in the x-a plane:\n\n");
printf("(x|...)\n");
print_aberrations(r, ang, n1, n2, 1);
printf("(a|...)\n");
print_aberrations(r, ang, n1, n2, 2);
}

```