SLAC/AP-100
January 1995
(AP)

# Notes on Lie algebraic analysis of achromats*

Chunxi Wang, Alex Chao

Stanford Linear Accelerator Center
Stanford University, Stanford, CA 94305

## 1 Introduction

This note is the summary of an exercise of using Lie algebra method to analyze third order achromats. Analytical results that reveal the dependency of the achromatic condition on the linear properties (beta function etc.) and nonlinear parameters(sextupole strength etc.) are pursued. Only some basic knowledge of Lie algebra technique are assumed. Reviews starting from first principles are included as needed.

The concept of an achromat can be roughly described as a transport system that maps a beam from one location to another without any change (at least in the transverse dimensions) of the beam phase space distribution. We study the achromats that consist of $N$ identical cells. It is well known that, due to the symmetry of such a repetitive structure, many geometrical and chromatical aberrations of a beam transport system can be eliminated effectively[1, 8].

In the Lie map language, a beam transport is represented by a symplectic map $\mathcal{M}$ that maps the dynamical variables $X = (x, P_x, y, P_y, z, \delta)$ from location 0 to $s$, i.e.

$$X(s) \equiv \mathcal{M}X\big|_{X=X(0)} = e^{:h(X):}X\big|_{X=X(0)} \equiv (1+ :h: + \frac{1}{2!}:h:^2 + \frac{1}{3!}:h:^3 + \cdots)X\big|_{X=X(0)}, \quad (1)$$

where $:h:$ represents the classical Poisson bracket operation

$$:h: f(X) \equiv [h, f] \quad (2)$$

According to the factorization theorem[4, 5] of Lie maps, a Lie map $\mathcal{M}$ can be factorized as

$$\mathcal{M} = e^{:f_2:}e^{:f_3:}e^{:f_4:}e^{:f_5:}\cdots, \quad (3)$$

where $f_n$ is an n-th order homogeneous polynominal of $X$. In this note, we consider up to third order achromats, so we assume each cell is represented by a map

$$\mathcal{M}_{cell} = e^{:g_2:}e^{:g_3:}e^{:g_4:} \quad (4)$$

and the total map of an $N$-cell achromat is

$$\begin{aligned} \mathcal{M}_N &= (e^{:g_2:}e^{:g_3:}e^{:g_4:})^N \\ &= e^{:h_2:}e^{:h_3:}e^{:h_4:}e^{:O(X^5):}. \end{aligned} \quad (5)$$

Using this representation, the goal of an achromat is to obtain a map with $h_n$ depending only on $\delta$. For example, a second order achromat has the map $\mathcal{M} = e^{:c\delta^2:}e^{:d\delta^3:}e^{:O(X^4):}$, where $c$,$d$ are some

*Work supported by the Department of Energy contract DE-AC03-76SF00515.

constants determined by the achromat design. This means that the map is an identity map except for the $z$ coordinate, which is easy to see from Eqs.(1)–(2).

Normal form technique is a powerful method to analyze the achromat problem. Assume the one cell map $\mathcal{M}_{cell} = \mathcal{A}\mathcal{R}e^{:h_3:}e^{:h_4:}\mathcal{A}^{-1}$, where $h_3$,$h_4$ are the normal forms of the generators of the unit cell map, and $\mathcal{A}$ is the nonlinear transformation that brings $\mathcal{M}_{cell}$ into its normal form; then the map of the whole system is $\mathcal{M}_N = \mathcal{M}_{cell}^N = \mathcal{A}\mathcal{R}^N\mathcal{A}^{-1} = \mathcal{I}$, provided that we can set $e^{:h_3:}$, $e^{:h_4:}$, and $\mathcal{R}^N$ to the identity (or only $\delta$ dependent) maps. Therefore, the conditions to form an achromat are $h_3$ and $h_4$ equal to zero (or $\delta$ dependent only) and the total linear map is identity. In this report, we will apply these conditions to a FODO array (a simple model system) to make it an achromat. We will start from Hamiltonians and work all the way up to obtain the analytical expressions of the required sextupole and octupole strengths.

# 2 Hamiltonians for basic magnets

In order to obtain the map for a magnet, we start from the Hamiltonian. For a given Hamiltonian, the system evolves according to

$$\frac{dX}{ds} = - : H : X. \tag{6}$$

If H is independent of $s$ as in a magnet element, we have a formal solution

$$X(s) = e^{-s:H:}X\big|_{X=X(0)}. \tag{7}$$

So for a magnet element of length $L$, its Lie map is $e^{-L:H:}$. In the following, we work out the Hamiltonians for single multipole magnet, pointing out various approximations as we proceed.

Use the standard coordinate system with planar reference orbit and local radius $\rho$. According to Courant & Synider[3] or R. Ruth [13], the exact Hamiltonian (in MKS units) reads

$$H = -eA_s - (1+\frac{x}{\rho})P[1 - (\frac{P_x - eA_x}{P})^2 - (\frac{P_y - eA_y}{P})^2]^{\frac{1}{2}}, \tag{8}$$

where $A_s = \vec{A}\cdot\hat{s}(1+\frac{x}{\rho})$ with $\vec{A}$ the vector potential of the magnetic field and $\hat{s}$ the unit vector along the tangential direction of the reference orbit. The canonical coordinates associated with this Hamiltonian are $(x, P_x, y, P_y, t, -E)$, where $P_x$,$P_y$ are the kinetic momentum, $E = \sqrt{m^2c^4 + P^2c^2}$ is the particle energy.

We can further transform the $(t, -E)$ coordinate into $(z, P_0\delta)$ via the generating function

$$F_3(-E, z, s) = (s-z)\sqrt{(\frac{E}{c})^2 - m^2c^2} + P_0 z \tag{9}$$

where $P_0$ is the reference momentum. From this generating function we have

$$\begin{aligned} t &= -\frac{\partial F_3}{\partial(-E)} = \frac{s-z}{c}\frac{E}{Pc}, \\ P_0\delta &= -\frac{\partial F_3}{\partial z} = \sqrt{(\frac{E}{c})^2 - m^2c^2} - P_0 = P - P_0. \end{aligned} \tag{10}$$

The physical meaning of the new coordinates is clear now,

$$z = s - \beta ct, \qquad \delta = \frac{P - P_0}{P_0}. \tag{11}$$

The new Hamiltonian reads

$$\begin{aligned} H(x, P_x; y, P_y; z, P_0\delta) &= H(x, P_x; y, P_y; t, -E) + \frac{\partial F_3}{\partial s} \\ &= -eA_s - \{(1+\frac{x}{\rho})[1 - (\frac{P_x - eA_x}{P_0(1+\delta)})^2 - (\frac{P_y - eA_y}{P_0(1+\delta)})^2]^{\frac{1}{2}} - 1\}P_0(1+\delta). \end{aligned} \tag{12}$$

Scale both the Hamiltonian and momentum by a constant $P_0$, the nominal momentum of the particle, we get

$$H_{p_0}(x,P_x;y,P_y;z,\delta) = -\frac{eA_s}{P_0} - \{(1+\frac{x}{\rho})[1-(\frac{P_x - eA_x/P_0}{1+\delta})^2 - (\frac{P_y - eA_y/P_0}{1+\delta})^2]^{\frac{1}{2}} - 1\}(1+\delta). \quad (13)$$

Where $P_x$,$P_y$ are the normalized momenta with respect to $P_0$. Up till now, no approximation has been made.

For ideal multipoles, $\rho = \infty$, $A_s = A_z$, $A_x = A_y = 0$. The magnetic field potential $A_s$ can be parameterized as:

$$A_s = A_z = -\frac{P_0}{e}\frac{1}{n+1}Re[(\lambda_n + i\bar{\lambda}_n)(x+iy)^{(n+1)}], \quad (14)$$

where $n = 1, 2, 3, \cdots$ for quadrupole, sextupole, octupole, etc. The quantities $\lambda$ and $\bar{\lambda}$ describe the normal and the skew components of the multipole fields. They are related to the multipole field strengths via

$$\lambda_n = \frac{e}{P_0}\frac{1}{n!}\frac{\partial^n B_y}{\partial x^n}(0,0) \quad and \quad \bar{\lambda}_n = \frac{e}{P_0}\frac{1}{n!}\frac{\partial^n B_x}{\partial x^n}(0,0), \quad (15)$$

which are easy to show using $\vec{B} = \nabla \times \vec{A} = \frac{\partial A_z}{\partial y}\hat{e}_x - \frac{\partial A_z}{\partial x}\hat{e}_y$. Another instructive relation is $\lambda_n = \frac{e}{P_0}\frac{B}{r^n}$, where $B$ is the pole tip field strength and $r$ is the radius of the pole tips.

For dipole (sector bending magnet), $\rho \neq 0$, $\vec{A} = -\frac{1}{2}\vec{x} \times \vec{B} = -\frac{B_0}{2}(\rho + x)\hat{s}$. So

$$A_s = \vec{A}\cdot\hat{s}(1+\frac{x}{\rho}) = -\frac{B_0\rho}{2}(1+\frac{x}{\rho})^2 = -\frac{P_0}{2e}(1+\frac{x}{\rho})^2. \quad (16)$$

Insert the magnetic field potentials into $H_{p_0}$ we get specific Hamiltonians for sector magnet,

$$\begin{aligned} H_{p_0} &= \tfrac{1}{2}(1+\tfrac{x}{\rho})^2 - \{(1+\tfrac{x}{\rho})[1-(\tfrac{P_x}{1+\delta})^2 - (\tfrac{P_y}{1+\delta})^2]^{\frac{1}{2}} - 1\}(1+\delta) \\ &\simeq -\tfrac{x\delta}{\rho} + \tfrac{x^2}{2\rho^2} + \tfrac{1}{2(1+\delta)}(P_x^2 + P_y^2) + \tfrac{x}{2\rho(1+\delta)}(P_x^2+P_y^2) + O(X^4) \end{aligned} \quad (17)$$

and multipole magnets,

$$\begin{aligned} H_{p_0} &= \tfrac{1}{n+1}Re[(\lambda_n + i\bar{\lambda}_n)(x+iy)^{(n+1)}] - \{[1-(\tfrac{P_x}{1+\delta})^2 - (\tfrac{P_y}{1+\delta})^2]^{\frac{1}{2}} - 1\}(1+\delta) \\ &\simeq \tfrac{1}{n+1}Re[(\lambda_n + i\bar{\lambda}_n)(x+iy)^{(n+1)}] + \tfrac{1}{2(1+\delta)}(P_x^2+P_y^2) + O(X^4). \end{aligned} \quad (18)$$

We have dropped some constants from the Hamiltonians. The approximation in the above expressions is due to dropping the kinematic nonlinear effect terms, which appear in the fourth or higher order.

The Hamiltonians of Eqs.(13), (17)–(18) are rigorous in 6-D phase space. However, $P_x = (1+\delta)x' \neq x'$, $P_y = (1+\delta)y' \neq y'$ in this case, which is often an inconvenience. To deal with this, we can normalize the Hamiltonian $H$ and momentum with respect to the particle momentum $P$ instead of the nominal momentum $P_0$, and get new Hamiltonians as

$$H_p(x,P_x;y,P_y;z,\delta) \equiv \frac{H}{p} = -\frac{eA_s}{P_0(1+\delta)} - (1+\frac{x}{\rho})[1-(P_x - \frac{eA_x}{P_0(1+\delta)})^2 - (P_y - \frac{eA_y}{P_0(1+\delta)})^2]^{\frac{1}{2}}. \quad (19)$$

Where $P_x$,$P_y$ are the $P$ normalized momentum. Similar to Eqs.(17)–(18), we have the Hamiltonians for sector magnet,

$$\begin{aligned} H_p &= \tfrac{1}{2(1+\delta)}(1+\tfrac{x}{\rho})^2 - (1+\tfrac{x}{\rho})[1-P_x^2-P_y^2]^{\frac{1}{2}} \\ &\simeq -\tfrac{x\delta}{\rho(1+\delta)} + \tfrac{x^2}{2\rho^2(1+\delta)} + \tfrac{1}{2}(P_x^2+P_y^2) + \tfrac{x}{2\rho}(P_x^2+P_y^2) + O(X^4) \end{aligned} \quad (20)$$

and multipole magnets,

$$\begin{aligned} H_p &= \frac{1}{n+1} Re[(\lambda_n + i\bar{\lambda_n})(x+iy)^{(n+1)}]\frac{1}{1+\delta} - [1 - P_x^2 - P_y^2]^{\frac{1}{2}} \\ &\simeq \frac{1}{n+1} Re[(\lambda_n + i\bar{\lambda_n})(x+iy)^{(n+1)}]\frac{1}{1+\delta} + \frac{1}{2}(P_x^2 + P_y^2) + O(X^4). \end{aligned} \tag{21}$$

Now the new canonical momentum have the perfered property that $P_x = \frac{\partial H}{\partial P_x} = x'$ and $P_y = \frac{\partial H}{\partial P_y} = y'$. However, the change from $H_{p_0}$ to $H_p$ is not canonical, i.e. the two Hamiltonians do not describe the same system. They do not give the same equation of motion for the z coordinate, even though they give the same results for the other 5 dimensions $(x, P_x, y, P_y, \delta)$. Therefore, the maps based on $H_p$ should be good for these 5 dimensions but not for z, which is sufficient for many applications.

Further reduction of Hamiltonians via the introduction of $\hat{\delta} = \frac{\delta}{1+\delta}$ is often very useful[11]. This significantly simplifies the $\delta$ dependency.

## 3 Linear map from a general quadratic form Hamiltonian

This section describes a method to obtain the linear map from a Hamiltonian with general quadratic form. Although the linear maps for all standard magnets are well known, the method presented here may be useful for calculating the linear maps when various linear couplings exist among the phase space coordinates.

Any quadratic form can be written as $f_2 = -\frac{1}{2}\tilde{X}FX$ with $F$ a symmetric matrix, which is positive definite when the motion is stable, $X$ is the vector consisting of the phase space coordinates, and $\tilde{}$ means transpose. Due to A. Dragt[4], it is well known that the second order homogeneous polynominal $f_2$ as a generator gives a linear map $e^{:f_2:}$. How can we calculate the map given a general quadratic Hamiltonian? First thought is to calculate it via the power series definition

$$e^{:f_2:} = 1 + :f_2: + \frac{1}{2!} :f_2:^2 + \frac{1}{3!} :f_2:^3 + \cdots \tag{22}$$

Apply this series to the dynamical variable $X = (x, P_x, y, P_y, z, \delta)$ and using,

$$:f_2: X = SFX \tag{23}$$

where $S$ is the fundamental symplectic matrix with $\begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$ as diagonal blocks. This way one can calculate up to any order terms in principle and may sum them up. This is the most straight forward method that is applicable to a general (linear as well as nonlinear) map. The problem with this approach is that the series, Eq.(22), does not terminate (for a second order generator, all terms are even in the same order of $X$). So a summation of infinite series has to be performed in order to obtain a closed form expression, which is not always trivial although practically done for all standard magnets. Another way is via the Hamilton- Cayley theorem for matrices[2]. From Eqs.(22)–(23), we basically need to calculate the exponential matrix $e^{SF}$ of a Hamiltonian matrix $SF$. See the reference[2] for details.

The method described here relies on a basic theorem of matrix theory. Any square matrix $A$ is similar to a Jordan form $J$, the simplest form in the family of matrixes that are similar to $A$. That is, there exists a matrix $B$ that

$$A = BJB^{-1}, \qquad J = diag[J_1, J_2, \cdots] \quad and \quad J_i = \begin{pmatrix} \lambda & 1 & & \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ & & & \lambda \end{pmatrix} \tag{24}$$

The diagonal blocks are irreducible and $\lambda$ is an eigenvalue of $A$ (for non-degenerate eigenvalue, $J_i$ is just the eigenvalue). The rank of such irreducible block is determined by the structure of eigenvector space associated with the degenerate eigenvalue $\lambda$. In non-degenerate cases, $J$ is just a diagonal matrix.

Here we are interested in a special type of matrix which can be factorized as $SF$. The Jordan form as well as the eigen-spectrum of such Hamiltonian matrixes has special features:

1. If $\lambda$ is an eigenvalue, so are $-\lambda$, $\lambda^*$, and $-\lambda^*$.

2. For non-zero eigenvalues, their Jordan blocks are diagonal even if there are degeneracy.

The proof for the first statement is simple, noticing that

$$\begin{aligned} P(\lambda) &= \det[SF-\lambda I] = \det[S^{-1}(SF-\lambda I)S] = \det[FS-\lambda I] \\ &= \det[(SF-\lambda I)\tilde{\ }] = (-1)^{2N}\det[FS+\lambda I] \\ &= P(-\lambda) \end{aligned} \tag{25}$$

Therefore $\lambda$ and $-\lambda$ must appear in pair. The appearance of $\lambda^*$ is due to the fundmental algebra theorem. In fact, $P(\lambda)$ must be formed by factors like

$$\lambda^4 - 2(a^2-b^2)\lambda^2 + (a^2+b^2)^2, \qquad \lambda^2 - a^2, \qquad or \qquad \lambda^2 + a^2. \tag{26}$$

To prove the second statement, first notice that,

$$if \quad M(SF)M^{-1} = J, \qquad then \quad (MS\tilde{M})^{-1}\tilde{J}(MS\tilde{M}) = -\tilde{J}. \tag{27}$$

Thus, $-\tilde{J}$ is similar to $J$. Here the minus sign is special, although it is generally true that $\tilde{J}$ and $J$ are similar to each other. Based on this observation, one can prove the second statement, for example, by solving $BJ = -JB$ straightforwardly.

With the knowledge of the Jordan form of a Hamiltonian matrix, the linear map can be calculated according to

$$e^{SF} = e^{M^{-1}JM} = M^{-1}e^{J}M, \tag{28}$$

while $e^J$ is easy to calculate due to the simple Jordan form. For diagonal cases, it is simply $e^J = diag[e^{\lambda_1}, e^{\lambda_2}, \cdots]$. For the non-diagonal n by n blocks, notice that its n-th power is the null matrix. Thus, simple Taylor expansion will terminate in a few terms and each term is easy to calculate. Therefore, we can calculate any linear map matrix $e^{SF}$ in 6-D phase space as long as the Jordan decomposition is done. Jordan decomposition of a matrix is a well studied problem. Standard procedures are available, especially symbolic calculation routines exist in, for example, Mathematica software[17]. The method described here is closely related to the method using Hamilton-Cayley theorem in the sense that both methods take advantage of the eigenstructure of the matrix $SF$. It is also useful in solving the reverse problem, from a matrix to find its generator.

As examples, here are two non-conventional maps which can be calculated easily using the described algorithm. One is the angular-momentum operator

$$e^{-\theta:xP_y - yP_x:} \longrightarrow \begin{bmatrix} \cos\theta & 0 & -\sin\theta & 0 & 0 & 0 \\ 0 & \cos\theta & 0 & -\sin\theta & 0 & 0 \\ \sin\theta & 0 & \cos\theta & 0 & 0 & 0 \\ 0 & \sin\theta & 0 & \cos\theta & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \tag{29}$$

the other is a linear coupling map[14]

$$e^{-:axy+bxP_y+cP_xy+dP_xP_y:} \longrightarrow \begin{bmatrix} \cosh k & 0 & \frac{c\sinh k}{k} & \frac{d\sinh k}{k} & 0 & 0 \\ 0 & \cosh k & -\frac{a\sinh k}{k} & -\frac{b\sinh k}{k} & 0 & 0 \\ \frac{b\sinh k}{k} & \frac{d\sinh k}{k} & \cosh k & 0 & 0 & 0 \\ -\frac{a\sinh k}{k} & -\frac{c\sinh k}{k} & 0 & \cosh k & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \tag{30}$$

where $k = \sqrt{bc - ad}$.

# 4 Nonlinear map and factorization

Having described the calculation of the linear map for a given quadratic Hamiltonian, we review the calculation of the nonlinear map of a polynominal Hamiltonian as well as map factorization and truncation following A. Dragt and E. Forest[6]. Although the exponential map $e^{-s:H:}$ contains all information of a dynamical system in the most concise form and guarantees the symplecticity of the map, we often do need the Taylor map of the system for tracking or understanding some physics, even though a Taylor map usually has to be truncated up to a certain order and may not be symplectic after the truncation. A general approach to obtain a Taylor map from a Lie map is to factorize the Lie map into a series of Lie maps as Eq.(3). Then calculate the linear map as described above and every nonlinear map using power series expansion as Eq.(22). Finally concatenrate these maps into one map. The following reviews details of this procedure.

The factorization theorem guarantees that a Lie map can be factorized into a series of Lie maps as Eq.(3). There are some important features of the factorized map. Unlike the truncation of a Taylor map, we can truncate the Lie map series at any order while maintaining the symplecticity of the truncated map. The Lie map truncated to $f_{n+1}$ is accurate up to n-th order in Taylor map. The leading factor $e^{:f_2:}$ generates the linear map and higher order factors $e^{:f_n:}$ generate the nonlinear map. The power series of $e^{:f_2:}$ never terminates, while always terminates for higher order maps up to the order of map of interest. This is the result of the fact that : $f_n : X^m = O(X^{n+m-2})$.

For a given Hamiltonian $H$, the factorized series $\{f_n\}$ can be obtained via BCH formula[2, 4]. First one orders the Hamiltonian into $-sH = h_1 + h_2 + h_3 + \cdots$, where $h_n$ is an n-th order polynominal. Since $h_1$ generates a coordinate shift, we assume it is zero by appropriate choice of the coordinates. So our goal here is to find $\{f_n\}$ that satisfy

$$e^{-s:H:} = e^{:h_2+h_3+h_4+\cdots:} = e^{:f_2:}e^{:f_3:}e^{:f_4:}e^{:O(X^5):}. \tag{31}$$

This can be done via a variation of BCH formula

$$e^{:f+h:} = e^{:f:}\exp\{:\int_0^1 due^{-u:f:}h - \frac{1}{2}\int_0^1 du\int_0^u dv[e^{-v:f:}h, e^{-u:f:}h]:\}e^{:O(h^3):}. \tag{32}$$

So we have

$$\begin{aligned} & e^{:h_2+h_3+h_4+\cdots:} \\ = \;& e^{:h_2:}\exp\{:\textstyle\int_0^1 due^{-u:h_2:}h_3 + \int_0^1 due^{-u:h_2:}h_4 - \frac{1}{2}\int_0^1 du\int_0^u dv[e^{-v:h_2:}h_3, e^{-u:h_2:}h_3]:\}e^{:O(X^5):} \\ = \;& e^{:h_2:}\exp\{:\textstyle\int_0^1 due^{-u:h_2:}h_3:\} \\ & \exp\{:\textstyle\int_0^1 due^{-u:h_2:}h_4 - \frac{1}{2}\int_0^1 du\int_0^u dv[e^{-v:h_2:}h_3, e^{-u:h_2:}h_3]:\}e^{:O(X^5):}. \end{aligned} \tag{33}$$

Therefore the factorized generators are

$$\begin{array}{ll} f_2 & = h_2 \\ f_3 & = \int_0^1 du e^{-u:h_2:} h_3 \\ f_4 & = \int_0^1 du e^{-u:h_2:} h_4 - \frac{1}{2}\int_0^1 du \int_0^u dv [e^{-v:h_2:} h_3, e^{-u:h_2:} h_3]. \end{array} \tag{34}$$

These equations suffice for this note. A. Dragt and E. Forest paper[6] provides a general method to compute higher order generators.

To obtain the Taylor map from the factorized Lie map, we follow two different approaches as mentioned above. Calculate the matrix of the linear map $e^{:f_2:}$ according to section 3; usually we can obtain a closed form. Calculate the Taylor series of the nonlinear map $e^{:f_{>2}:}$ and truncate at a desired order $max$ as indicated.

$$e^{:f_n:} = \sum_{k=0}^{\frac{(max-1)}{(n-2)}} \frac{1}{k!} :f_n:^k + O(X^{max+1}) \tag{35}$$

From a series of maps to get the total map is the process of map concatenation, which is the subject of the next section.

# 5 Effective Hamiltonian and concatenation of maps

As mentioned in the last section, in general, we often need to concatenate a series of maps to obtain a single map. In fact, there are two issues. One is to calculate the total truncated Taylor map, the other is to obtain an effective Hamiltonian for the total Lie map. Since

$$e^{:f:} e^{:g:} X = e^{:f:} G(X) = G(e^{:f:} X) = G(F(X)) = G(X)\big|_{X=F(X)}, \tag{36}$$

the first issue is no more than a series of coordinate transformations, each corresponding to one of the factor maps. The number of terms and their orders increase rapidly in this process, truncation is necessary to keep it under control, nonetheless it is straightforward.

To obtain the effective Hamiltonian is harder. It can be done via similarity transformations and BCH formula . It is basically the reverse of factorization. Here the goal is to find a series $\{h_n\}$ for a series factorized generators $\{f_n^i\}$ that satisfy

$$\begin{array}{ll} e^{:h_2:} e^{:h_3:} e^{:h_4:} \dots & = \prod_{i=1} (e^{:f_2^i:} e^{:f_3^i:} e^{:f_4^i:} \dots) \\ & \equiv (e^{:f_2^1:} e^{:f_3^1:} e^{:f_4^1:} \dots)(e^{:f_2^2:} e^{:f_3^2:} e^{:f_4^2:} \dots) \dots \end{array} \tag{37}$$

The solution is

$$\begin{array}{ll} \mathcal{R} & = e^{:h_2:} = \prod_{i=1}^N e^{:f_2^i:} \equiv e^{:f_2^1:} e^{:f_2^2:} e^{:f_2^3:} \dots e^{:f_2^N:} \\ h_3 & = \sum_{i=1}^N \tilde{f}_3^i, \\ h_4 & = \sum_{i=1}^N \tilde{f}_4^i + \frac{1}{2} \sum_{j>i=1}^N [\tilde{f}_3^i, \tilde{f}_3^j] \\ \dots & \end{array} \tag{38}$$

where $\tilde{f}^i$ means $\tilde{f}^i(X) = f^i(R_{i\to N}^{-1} X)$ with $R_{i\to N}^{-1} = R_{N\to i} \leftrightarrow e^{-:f_2^N:} e^{-:f_2^{N-1}:} \dots e^{-:f_2^{i+1}:}$, the linear transformation matrix from the last element $N$ to the $i$-th element. The proof of Eq.(38) is easy to see from a simple example:

$$
\begin{aligned}
& e^{:f_2^1:}e^{:f_3^1:}e^{:f_4^1:}e^{:f_2^2:}e^{:f_3^2:}e^{:f_4^2:}e^{:f_2^3:}e^{:f_3^3:}e^{:f_4^3:} \\
= \;& e^{:f_2^1:}e^{:f_2^2:}e^{:f_2^3:}e^{-:f_2^3:}e^{-:f_2^2:}e^{:f_3^1:}e^{:f_4^1:}e^{:f_2^2:}e^{:f_2^3:}e^{-:f_2^3:}e^{:f_3^2:}e^{:f_4^2:}e^{:f_2^3:}e^{:f_3^3:}e^{:f_4^3:} \\
= \;& e^{:f_2^1:}e^{:f_2^2:}e^{:f_2^3:}\exp\{:e^{-:f_2^3:}e^{-:f_2^2:}f_3^1:\}\exp\{:e^{-:f_2^3:}e^{-:f_2^2:}f_4^1:\}\exp\{:e^{-:f_2^3:}f_3^2:\} \\
& \exp\{:e^{-:f_2^3:}f_4^2:\}e^{:f_3^3:}e^{:f_4^3:} \\
= \;& e^{:f_2^1:}e^{:f_2^2:}e^{:f_2^3:}\exp\{:e^{-:f_2^3:}e^{-:f_2^2:}f_3^1:\}\exp\{:e^{-:f_2^3:}f_3^2:\}e^{:f_3^3:}\exp\{:e^{-:f_2^3:}e^{-:f_2^2:}f_4^1:\} \\
& \exp\{:e^{-:f_2^3:}f_4^2:\}e^{:f_4^3:}e^{:O(X^4):} \\
= \;& e^{:f_2^1:}e^{:f_2^2:}e^{:f_2^3:}\exp\{:e^{-:f_2^3:}e^{-:f_2^2:}f_3^1+e^{-:f_2^3:}f_3^2+f_3^3+\tfrac{1}{2}([e^{-:f_2^3:}e^{-:f_2^2:}f_3^1,e^{-:f_2^3:}f_3^2]+ \\
& [e^{-:f_2^3:}e^{-:f_2^2:}f_3^1,f_3^3]+[e^{-:f_2^3:}f_3^2,f_3^3]):\}\exp\{:e^{-:f_2^3:}e^{-:f_2^2:}f_4^1+e^{-:f_2^3:}f_4^2+f_4^3:\}e^{:O(X^4):} \\
= \;& \mathcal{R}\exp\{:\tilde{f}_3^1+\tilde{f}_3^2+\tilde{f}_3^3:\}\exp\{:\tilde{f}_4^1+\tilde{f}_4^2+\tilde{f}_4^3+\tfrac{1}{2}([\tilde{f}_3^1,\tilde{f}_3^2]+[\tilde{f}_3^1,\tilde{f}_3^3]+[\tilde{f}_3^2,\tilde{f}_3^3]):\}e^{:O(X^4):}
\end{aligned}
\tag{39}
$$

Using Eqs.(37)–(38), we can concatenate a series maps with known generators and obtain the generators of a single map. The linear part of the map $\mathcal{R}$ can be calculated with traditional matrix operations. The factorized higher order operators provide concise information of the nonlinear behavior of the system. Various tools such as normal form analysis can be used to analyze these nonlinear operators in order to understand and control the nonlinear behavior. When necessary, a total effective Hamiltonian can be constructed from the factorized generators of the final map.

# 6 Resonance basis

Up till now, we discussed how to calculate the Lie map as well as the Taylor map of a transport line up to 4-th Hamiltonian order in the dynamical variables starting from the basic vector potentials of the magnetic fields of element magnets. Before working on the achromat, we will discuss one more tool that will be used to analyze the Lie map and obtain the conditions to form achromats. Although the method is quite general, here our description is somewhat specific to our case: an achromat consisting of repetitive unit cells with map $\mathcal{M}_N = (\mathcal{R}e^{:h_3:}e^{:h_4:})^N e^{:O(X^5):}$.

As we have shown the power of Jordan form in matrix theory, essentially the same idea leads to the resonance basis and normal form analysis of a Lie map. For linear maps, they are actually the same thing. The key idea is to reduce the map to a form which is as simple as possible using similarity transformations. Note that a similarity transformation corresponds to coordinate transformation which is not essential to the dynamics we are interested in, while a reduced simpler map is much easier to handle and understand.

In this note we consider a system with mid-plane symmetry and no linear x-y coupling. The linear transformation matrix from the end of a unit cell to a location $s$ inside the unit cell can be parameterized with Twiss parameters as[2]

$$
R(s) = \begin{bmatrix}
R_{11} & R_{12} & 0 & 0 & 0 & \eta(s) - R_{11}\eta(0) - R_{12}\eta'(0) \\
R_{21} & R_{22} & 0 & 0 & 0 & \eta'(s) - R_{21}\eta(0) - R_{22}\eta'(0) \\
0 & 0 & R_{33} & R_{34} & 0 & 0 \\
0 & 0 & R_{43} & R_{44} & 0 & 0 \\
A & B & 0 & 0 & 1 & C - A\eta(0) - B\eta'(0) \\
0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}
\tag{40}
$$

where

$$
\begin{aligned}
R_{11} &= \sqrt{\tfrac{\beta_x(s)}{\beta_x(0)}}\,(\cos\psi_x(s) + \alpha_x(0)\sin\psi_x(s))\\
R_{12} &= \sqrt{\beta_x(0)\,\beta_x(s)}\,\sin\psi_x(s)\\
R_{21} &= \tfrac{1}{\sqrt{\beta_x(0)\,\beta_x(s)}}\,((\alpha_x(0) - \alpha_x(s))\,\cos\psi_x(s) - (1 + \alpha_x(0)\,\alpha_x(s))\,\sin\psi_x(s))\\
R_{22} &= \sqrt{\tfrac{\beta_x(0)}{\beta_x(s)}}\,(\cos\psi_x(s) - \alpha_x(s)\sin\psi_x(s))\\
R_{33} &= \sqrt{\tfrac{\beta_y(s)}{\beta_y(0)}}\,(\cos\psi_y(s) + \alpha_y(0)\sin\psi_y(s))\\
R_{34} &= \sqrt{\beta_y(0)\,\beta_y(s)}\,\sin\psi_y(s)\\
R_{43} &= \tfrac{1}{\sqrt{\beta_y(0)\,\beta_y(s)}}\,((\alpha_y(0) - \alpha_y(s))\,\cos\psi_y(s) - (1 + \alpha_y(0)\,\alpha_y(s))\,\sin\psi_y(s))\\
R_{44} &= \sqrt{\tfrac{\beta_y(0)}{\beta_y(s)}}\,(\cos\psi_y(s) - \alpha_y(s)\sin\psi_y(s))\\
A &= R_{21}\eta(s) - R_{11}\eta'(0) + \eta'(0)\\
B &= -R_{12}\eta'(s) + R_{22}\eta(s) - \eta(0)\\
C &= -\textstyle\int_0^s ds\tfrac{\eta(s)}{\rho(s)}
\end{aligned}
$$

and $\psi_{x,y}(s)$ are the betatron phase advances from location 0 to $s$. Since we are interested in a repetitive structure, the Twiss parameters at both ends of the unit cell should be equal and the linear map $\mathcal{R}$ of the cell reads

$$
R = \begin{bmatrix}
\cos\mu_x + \alpha_x\sin\mu_x & \beta_x\sin\mu_x & 0 & 0 & 0 & \eta - R_{11}\eta - R_{12}\eta' \\
-\gamma_x\sin\mu_x & \cos\mu_x - \alpha_x\sin\mu_x & 0 & 0 & 0 & \eta' - R_{21}\eta - R_{22}\eta' \\
0 & 0 & R_{33} & R_{34} & 0 & 0 \\
0 & 0 & R_{43} & R_{44} & 0 & 0 \\
\eta' - \eta' R_{11} + \eta R_{21} & -\eta + \eta R_{22} - \eta' R_{12} & 0 & 0 & 1 & -C\alpha_c + A_\eta\sin\mu_x \\
0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix} \tag{41}
$$

where $A_\eta = \gamma_x\eta^2 + 2\alpha_x\eta\eta' + \beta_x\eta'^2$; $R_{33}, R_{34}, R_{43}, R_{44}$ are the same as the $x$ block except changing $x$ to $y$; $\mu$ is the phase advance of one cell. Transform the physical coordinates to the normalized coordinates using a symplectic matrix $A$ as

$$
\begin{bmatrix} \bar{x} \\ \bar{P}_x \\ \bar{y} \\ \bar{P}_y \\ \bar{z} \\ \bar{\delta} \end{bmatrix} = A^{-1} \begin{bmatrix} x \\ P_x \\ y \\ P_y \\ z \\ \delta \end{bmatrix} = \begin{bmatrix} \frac{x - \eta\delta}{\sqrt{\beta_x}} \\ \frac{\alpha_x(x - \eta\delta) + \beta_x(P_x - \eta'\delta)}{\sqrt{\beta_x}} \\ \frac{y}{\sqrt{\beta_y}} \\ \frac{\alpha_y y + \beta_y P_y}{\sqrt{\beta_y}} \\ z + \eta' x - \eta P_x \\ \delta \end{bmatrix} \tag{42}
$$

After this transformation, the linear map is reduced to a much simpler form

$$
N = A^{-1} R A = \begin{bmatrix}
\operatorname{ccs}\mu_x & sin\mu_x & 0 & 0 & 0 & 0 \\
-\sin\mu_x & \cos\mu_x & 0 & 0 & 0 & 0 \\
0 & 0 & \cos\mu_y & sin\mu_y & 0 & 0 \\
0 & 0 & -\sin\mu_y & \cos\mu_y & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & -C\alpha_c \\
0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix} \tag{43}
$$

Notice that, this form is not a Jordan form. Although we can further reduce Eq.(43) into a simpler Jordan form via a similarity transformation, we choose not to do so to avoid the use of complex quantities. The results obtained will be the same.

Notice that the $x$ and $y$ blocks of $N$ is a simple rotation, so we can further reduce the Hamiltonian via the action angle transformation as

$$\bar{x} = \sqrt{2J_x}\sin\phi_x, \qquad \bar{p}_x = \sqrt{2J_x}\cos\phi_x, \qquad \bar{y} = \sqrt{2J_y}\sin\phi_y, \qquad \bar{p}_y = \sqrt{2J_y}\cos\phi_y \tag{44}$$

Using the new dynamical variables $\phi_{x,y}$ and $J_{x,y}$, the linear map generator $h_2$ is reduced to a very simple form

$$h_2 = -\mu_x J_x - \mu_y J_y - \frac{1}{2}\bar{\alpha}_c\delta^2 \tag{45}$$

One can define the eigenmodes of $: h_2 :$ as

$$|abcd, e>= J_x^{(a+b)/2} J_y^{(c+d)/2} e^{i(a-b)\phi_x} e^{i(c-d)\phi_x} \delta^e \tag{46}$$

which satisfy

$$: h_2 : |abcd, e>= i[(a-b)\mu_x + (c-d)\mu_y]\ |abcd, e> \tag{47}$$

or

$$e^{:h_2:}|abcd, e>= e^{i[(a-b)\mu_x+(c-d)\mu_y]}\ |abcd, e> .$$

Since $(a-b)\mu_x + (c-d)\mu_y = 2\pi\times$integer constitutes the resonance conditions of the linear map $e^{:h_2:}$, $|abcd, e>$ are often called resonance basis. The eigenmodes of Eq.(45) are then used to expand the nonlinear generators $h_n$, as

$$h_n = \sum_{\substack{a,b,c,d,e\geq 0 \\ a+b+c+d+e=n}} C^n_{abcd,e}\ |abcd, e>, \tag{48}$$

where $C^n_{abcd,e}$ are constants depending on the lattice parameters. We will see their role later.

Combine the two transformations of Eqs. (42) and (44) into

$$\begin{aligned} x &= \sqrt{2J_x\beta_x}\sin\phi_x + \eta\delta \\ P_x &= \sqrt{\tfrac{2J_x}{\beta_x}}(\cos\phi_x - \alpha_x\sin\phi_x) + \eta'\delta \\ y &= \sqrt{2J_y\beta_y}\sin\phi_y \\ P_y &= \sqrt{\tfrac{2J_y}{\beta_y}}(\cos\phi_y - \alpha_y\sin\phi_y). \end{aligned} \tag{49}$$

This transformation is for the phase space at a given location. Assuming it is at the end of the unit cell, then $\beta_x = \beta_x(0)$, $\alpha_x = \alpha_x(0)$, etc. The quantities in Eq.(38) require phase space variables at various locations, i.e. we need to transform the dynamical variables from the end to a location $s$ via Eq.(40). This can be combined with the above normal form transformation of Eq.(49) and get

$$\begin{aligned} \tilde{x}(s) &= \sqrt{2J_x\beta_x(s)}\sin(\phi_x + \psi_x(s)) + \eta(s)\delta \\ \tilde{P}_x(s) &= \sqrt{\tfrac{2J_x}{\beta_x(s)}}(\cos(\phi_x + \psi_x(s)) - \alpha_x(s)\sin(\phi_x + \psi_x(s))) + \eta'(s)\delta \\ \tilde{y}(s) &= \sqrt{2J_y\beta_y(s)}\sin(\phi_y + \psi_y(s)) \\ \tilde{P}_y(s) &= \sqrt{\tfrac{2J_y}{\beta_y(s)}}(\cos(\phi_y + \psi_y(s)) - \alpha_y(s)\sin(\phi_y + \psi_y(s))) \end{aligned} \tag{50}$$

which is the expected result. $\tilde{X}(s) \equiv R_{N\to s}X$ is the quantity to be used in calculating the effective Hamiltonian.

Since we need to transform a large number of various homogeneous polynominals into the resonance basis, the following identities can significantly reduce the amount of calculations. Their proofs are straightforward:

$$
\begin{array}{l}
ax^2+bxP_x+cP_x^2=(a\beta_x-b\alpha_x+c\gamma_x)J_x \\
-\frac{1}{2\beta_x}[a\beta_x^2-b\beta_x(\alpha_x-i)+c(\alpha_x-i)^2]J_xe^{i2(\phi_x+\psi_x)}+c.c. \\
-\frac{i}{\sqrt{2\beta_x}}\{[2a\beta_x-b(\alpha_x-i)]\eta_x+[b\beta_x-2c(\alpha_x-i)]\eta_x'\}\sqrt{J_x}e^{i(\phi_x+\psi_x)}\delta+c.c. \\
+(a\eta_x^2+b\eta_x\eta_x'+c\eta_x'^2)\delta^2
\end{array}
\tag{51}
$$

where *c.c.* means complex conjugate of the previous term. Of course they apply to $y$ dimension also.

$$
\begin{array}{l}
ax^3+bx^2P_x+cxP_x^2+dP_x^3=[(3a\beta_x-2b\alpha_x+c\gamma_x)\eta_x+(b\beta_x-2c\alpha_x+3d\gamma_x)\eta_x']J_x\delta \\
+i(2\beta_x)^{-\frac{3}{2}}[-3a\beta_x^3+b\beta_x^2(3\alpha_x-i)-c\beta_x(3\alpha_x+i)(\alpha_x-i)+3d\beta_x\gamma_x(\alpha_x-i)]J_x^{\frac{3}{2}}e^{i(\phi_x+\psi_x)}+c.c. \\
+i(2\beta_x)^{-\frac{3}{2}}[a\beta_x^3-b\beta_x^2(\alpha_x-i)+c\beta_x(\alpha_x-i)^2-d(\alpha_x-i)^3]J_x^{\frac{3}{2}}e^{i3(\phi_x+\psi_x)}+c.c. \\
\quad+(2\beta_x)^{-1}\{\ [-3a\beta_x^2+2b\beta_x(\alpha_x-i)-c(\alpha_x-i)^2]\eta_x+ \\
\qquad\qquad [-b\beta_x^2+2c\beta_x(\alpha_x-i)-3d(\alpha_x-i)^2]\eta_x'\ \}J_xe^{i2(\phi_x+\psi_x)}\delta+c.c. \\
\quad+i(2\beta_x)^{-\frac{1}{2}}\{\ [-3a\beta_x+b(\alpha_x-i)]\eta_x^2+2[-b\beta_x+c(\alpha_x-i)]\eta_x\eta_x'+ \\
\qquad\qquad [-c\beta_x+3d(\alpha_x-i)]\eta_x'^2\ \}\sqrt{J_x}e^{i(\phi_x+\psi_x)}\delta^2+c.c. \\
+(a\eta_x^3+b\eta_x^2\eta_x'+c\eta_x\eta_x'^2+d\eta_x'^3)\delta^3
\end{array}
\tag{52}
$$

In the 4-th order, we write down only those terms which are independent of the phases $\phi_{x,y}$, since we do not use the other terms while they become messy:

$$
\begin{array}{l}
ax^4+bx^3P_x+cx^2P_x^2+dxP_x^3+eP_x^4= \\
\frac{3}{2}[a\beta_x^2-b\beta_x\alpha_x+c(\alpha_x^2+\frac{1}{3})-d\alpha_x\gamma_x+e\gamma^2]J_x^2 \\
+[(6a\beta_x-3b\alpha_x+c\gamma_x)\eta_x^2+(3b\beta_x-4c\alpha_x+3d\gamma_x)\eta_x\eta_x'+(c\beta_x-3d\alpha_x+6e\gamma_x)\eta_x'^2]J_x\delta^2 \\
+(a\eta_x^4+b\eta_x^3\eta_x'+c\eta_x^2\eta_x'^2+d\eta_x\eta_x'^3+e\eta_x'^4)\delta^4 \\
+function(\phi_x)
\end{array}
\tag{53}
$$

Using Eqs.(49)–(53) we are able to decompose the generators into resonance basis efficiently. In fact, identities similar to Eqs.(51)–(53) but in 3-D instead of 1-D are used in most of our calculations.

A very useful feature of the resonance basis is that the Poisson bracket of two resonance basis can be expressed as

$$
\begin{array}{l}
[|a_1b_1c_1d_1,e_1>,|a_2b_2c_2d_2,e_2>]= \\
i(a_1b_2-a_2b_1)|a_1+a_2-1,b_1+b_2-1,c_1+c_2,d_1+d_2,e_1+e_2>+ \\
i(c_1d_2-c_2d_1)|a_1+a_2,b_1+b_2,c_1+c_2-1,d_1+d_2-1,e_1+e_2>
\end{array}
\tag{54}
$$

which allows Poisson bracket calculations without differentiation. Notice that the Poisson bracket operation resembles the creation and annihilation operators in quantum mechanics.

Another important feature of the resonance basis is its Poisson bracket with the dynamical variables $X$.

$$
\begin{array}{l}
[|abcd,e>,x]=-\sqrt{\frac{\beta_x}{2}}\{a|a-1,b,c,d,e>+b|a,b-1,c,d,e>\} \\
[|abcd,e>,P_x]=\sqrt{\frac{1}{2\beta_x}}\{a(\alpha_x+i)|a-1,b,c,d,e>+b(\alpha_x-i)|a,b-1,c,d,e>\} \\
[|abcd,e>,y]=-\sqrt{\frac{\beta_y}{2}}\{c|a,b,c-1,d,e>+d|a,b,c,d-1,e>\} \\
[|abcd,e>,P_y]=\sqrt{\frac{1}{2\beta_y}}\{c(\alpha_y+i)|a,b,c-1,d,e>+d(\alpha_y-i)|a,b,c,d-1,e>\}
\end{array}
\tag{55}
$$

These identities play a key role in the recent development of the "nPB" tracking technique, which is faster and more accurate than the usual Taylor series tracking[12]. The point is that after calculation of the resonance basis for the initial condition, only arithmetic computation of some coefficients are involved in tracking.

The importance of the resonance basis is (at least in our case) due to the appearance of the operator $\frac{1}{1-e^{:h_2:}}$ in many basic calculations. This operator is singular under resonance conditions. The resonance basis expansion can effectively separate the resonant and non-resonant part of a function, so that we can deal with them separately.

# 7 Normal form analysis

In the effort to reduce a nonlinear map to an as-simple-as possible form, it turns out that (in cases there are no resonances nearby) all the non-resonance terms can be transformed away via a symplectic similarity transformation. At each order the only terms left are

$$h_n = \sum_{2a+2c+e=n} C^n_{aacc,e}\,|aacc, e> \tag{56}$$

Such kind of irreducible form is called normal form, which depends on $J_x$,$J_y$,$\delta$ only. In this note, we will use $h_3$ and $h_4$, so their calculations is summarized in this section. For details see Ref.[2]

Given a third order map $\mathcal{M} = e^{:f_2(X):}e^{:f_3(X):}e^{:f_4(X):}$, a similarity transformation transforms it into a normal form as

$$\mathcal{N} = \mathcal{A}\mathcal{M}\mathcal{A}^{-1} = e^{:h_2(J_x,J_y,\delta):}e^{:h_3(J_x,J_y,\delta):}e^{:h_4(J_x,J_y,\delta):} + O(X^4) \tag{57}$$

where

$$\mathcal{A} = e^{:F_4:}e^{:F_3:}\mathcal{A}_2$$

$\mathcal{A}_2$ is the transformation for the linear map as discussed in the previous section. $F_n$ is given by

$$F_n = \sum^{n}_{\left\{\begin{array}{l} a,b,c,d,e = 0 \\ a+b+c+d+e=n \\ \text{unless} \quad a=b \quad \text{and} \quad c=d \end{array}\right.} C^n_{abcd,e}\frac{1}{1-e^{-i(a-b)\mu_x - i(c-d)\mu_y}}|abcd, e> \tag{58}$$

where $|abcd, e>$ is the resonance basis and $C^n_{abcd,e}$ are the coefficients of the nonlinear generators expanded in the resonance basis, i.e.

$$\begin{array}{l} \sum^3_{a,b,c,d,e} C^3_{abcd,e}|abcd, e> = f_3 \\ \sum^4_{a,b,c,d,e} C^4_{abcd,e}|abcd, e> = f_4 - \frac{1}{2} : f_3 : F_3 + : (f_3 - F_3) : h_3 \end{array} \tag{59}$$

Notice the extra term on the 4-th order. Again these are for cases away from any resonance.

Normally, there are only three possible terms in the 3-rd order. All of them are chromatic terms, $h_3$ reads

$$h_3 = C^3_{1100,1}J_x\delta + C^3_{0011,1}J_y\delta + C^3_{0000,3}\delta^3 \tag{60}$$

In case there are resonances (i.e.$(a-b)\nu_x + (c-d)\nu_y = integer$), other terms may appear. For example, if $\nu_x = \nu_y$, terms $C^3_{1001,1}\sqrt{J_xJ_y}e^{i(\phi_x-\phi_y)}\delta$ and $C^3_{0110,1}\sqrt{J_xJ_y}e^{-i(\phi_x-\phi_y)}\delta$ will appear. However, these two terms vanish by mid-plane symmetry (recall that $y \propto \sqrt{J_y}$). There are a total of 6 possible terms in the 4-th order away from resonances, which reads

$$h_4 = C^4_{2200,0}J_x^2 + C^4_{0022,0}J_y^2 + C^4_{1111,0}J_xJ_y + C^4_{1100,2}J_x\delta^2 + C^4_{0011,2}J_y\delta^2 + C^4_{0000,4}\delta^4 \tag{61}$$

Also a few more resonance terms may appear for special cases. See reference[8] for details.

## 8 Second order achromats

Up till now, we described the methods to be used to work out an achromat. In the following, we consider a basic system consisting of repetitive thin lens FODO cells with thick sector bending magnets filling the drift spaces and thin sextupoles and octupoles are inserted to correct the chromatic and geometric aberrations. The unit cell layout looks like

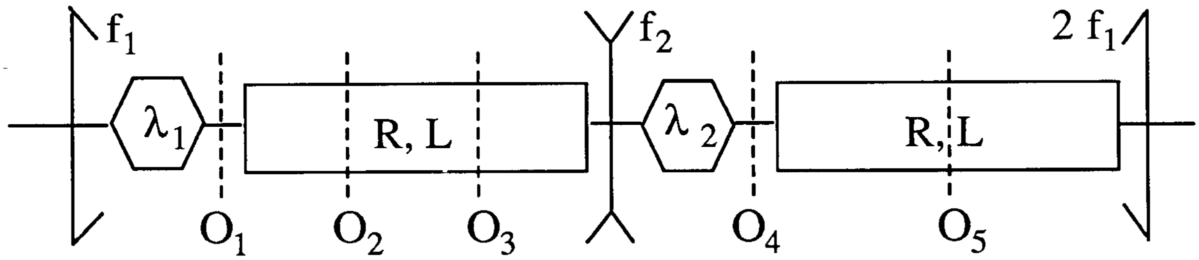


Figure 1: Sketch of a unit cell

According to the Eqs.(20)–(21) of section 2, the following Hamiltonians (the generators in fact) are used in our calculation.

- thick dipole

$$H = -L\left(\frac{P_x^2+P_y^2}{2}+\frac{x^2}{2R^2}-\frac{x\delta}{R}+\frac{x(P_x^2+P_y^2)}{2R}-\frac{x^2\delta}{2R^2}+\frac{x\delta^2}{R}+\frac{(P_x^2+P_y^2)^2}{8}-\frac{x\delta^3}{R}+\frac{x^2\delta^2}{2R^2}\right)$$

- thin quadrupole $H = -\frac{1}{2f_s}(x^2-y^2)(1-\delta+\delta^2)$
- thin sextupole $H = -\frac{\lambda_s}{3}(x^3-3xy^2)(1-\delta)$
- thin octupole $H = -\frac{O_s}{4}(x^4-6x^2y^2+y^4)$

Where $R$ is the bending radius, $L$ the total dipole length of each half cell; $f_s$ is the focal length of a quadrupole; $\lambda_s$ and $O_s$ are the $s$-th integrated sextupole and octupole strengths.

The linear lattice of such a system is well known, we assume the Twiss parameters as well as dispersion functions are known. Using Eq.(34) of section 4, the nonlinear generators of each element reads

- thick dipole

$$\begin{aligned}
f_3 = \; & -\tfrac{1}{6R^2}\sin^3\tfrac{L}{R}\,x^3-\tfrac{1}{4R}\sin\tfrac{L}{R}\sin\tfrac{2L}{R}\,x^2P_x-\tfrac{1}{4}\cos\tfrac{L}{R}\sin\tfrac{2L}{R}\,xP_x^2+\tfrac{R}{6}(1-\cos^3\tfrac{L}{R})P_x^3\\
& -\tfrac{1}{2}\sin\tfrac{L}{R}\,xP_y^2+R\sin^2\tfrac{L}{2R}\,P_xP_y^2+\tfrac{1}{2R}\sin\tfrac{L}{R}(\cos\tfrac{L}{R}+\sin^2\tfrac{L}{R})x^2\delta-2\sin^2\tfrac{L}{2R}\sin^2\tfrac{L}{R}\,xP_x\delta\\
& -\tfrac{R}{2}\sin^2\tfrac{L}{2R}\sin\tfrac{2L}{R}\,P_x^2\delta-\tfrac{1}{2}(L-R\sin\tfrac{L}{R})P_y^2\delta-\tfrac{1}{2}(\sin^3\tfrac{L}{R}+\sin\tfrac{2L}{R})x\delta^2\\
& +\tfrac{R}{2}(2-\cos\tfrac{L}{R})\sin^2\tfrac{L}{R}\,P_x\delta^2+\tfrac{1}{12}(-6L+2R\sin^3\tfrac{L}{R}+3R\sin\tfrac{2L}{R})\delta^3\\
f_4 = \; & -\tfrac{1}{8R}\sin^3\tfrac{L}{R}\,x^2(P_x^2+P_y^2)-\tfrac{1}{8}\sin\tfrac{L}{R}\sin\tfrac{2L}{R}\,xP_x(P_x^2+P_y^2)-\tfrac{R}{8}\cos^2\tfrac{L}{R}\sin\tfrac{L}{R}\,P_x^2(P_x^2+P_y^2)\\
& -\tfrac{R}{8}\sin\tfrac{L}{R}\,P_y^2(P_x^2+P_y^2)+\tfrac{1}{12R^2}\sin^3\tfrac{L}{R}\,x^3\delta+(\tfrac{1}{2}+\cos\tfrac{L}{R})\sin^2\tfrac{L}{2R}\sin\tfrac{L}{R}\,xP_x^2\delta\\
& +\tfrac{R}{12}(3+4\cos\tfrac{L}{R}+5\cos\tfrac{2L}{R})\sin^2\tfrac{L}{2R}\,P_x^3\delta+\tfrac{1}{4}\sin^2\tfrac{L}{2R}\sin\tfrac{2L}{R}\,xP_y^2\delta\\
& +\tfrac{R}{4}(3+\cos\tfrac{2L}{R})\sin^2\tfrac{L}{2R}\,P_xP_y^2\delta-\tfrac{1}{4R}(\sin^3\tfrac{L}{R}+\sin\tfrac{2L}{R})\,x^2\delta^2+\tfrac{1}{2}\sin^2\tfrac{L}{R}\,xP_x\delta^2\\
& -\tfrac{R}{4}(1+3\cos\tfrac{L}{R})\sin^2\tfrac{L}{2R}\sin\tfrac{L}{R}\,P_x^2\delta^2+\tfrac{R}{2}\sin^4\tfrac{L}{2R}\sin\tfrac{L}{R}\,P_y^2\delta^2+(\cos\tfrac{L}{R}+\tfrac{1}{4}\sin^2\tfrac{L}{R})\sin\tfrac{L}{R}\,x\delta^3\\
& -\tfrac{R}{2}\sin^2\tfrac{L}{R}\,P_x\delta^3+\tfrac{1}{12}(6L-R\sin^3\tfrac{L}{R}-3R\sin\tfrac{2L}{R})\delta^4
\end{aligned}$$

- thin quadrupole $f_3 = \frac{1}{2f_s}(x^2 - y^2)\delta$, $f_4 = -\frac{1}{2f_s}(x^2 - y^2)\delta^2$
- thin sextupole $f_3 = -\frac{\lambda_s}{3}(x^3 - 3xy^2)$, $f_4 = \frac{\lambda_s}{3}(x^3 - 3xy^2)\delta$
- thin octupole $f_3 = 0$, $f_4 = -\frac{O_s}{4}(x^4 - 6x^2y^2 + y^4)$

where thick element integrations for the dipole are performed.

It is not hard to solve the second order case. According to Eqs.(38),(51), and (60), the second order normal form of the unit cell is

$$h_3 = C_x J_x \delta + C_y J_y \delta + C_\delta \delta^3 \tag{62}$$

where

$$C_x = \sum_{s=1,2}^{Q} [\frac{1}{2f_s} - \lambda_s \eta_x(s)]\, \beta_x(s) + w_x \qquad C_y = -\sum_{s=1,2}^{Q} [\frac{1}{2f_s} - \lambda_s \eta_x(s)]\, \beta_y(s) + w_y$$

and

$$\begin{aligned} w_x = \textstyle\sum_{s=1,2}^{D} \frac{1}{2}\sin^2(\frac{L}{R})\Big\{ \; & \frac{\beta_x(s)}{R}\left[\sin\frac{L}{R} + \cot\frac{L}{R} - \sin\frac{L}{R}\frac{\eta_x(s)}{R} - \cos\frac{L}{R}\eta_x'(s)\right] + \\ & 2\alpha_x(s)\left[1 - \cos\frac{L}{R} + \cos\frac{L}{R}\frac{\eta_x(s)}{R} + \cos\frac{L}{R}\cot\frac{L}{R}\eta'(s)\right] + \\ & \gamma_x(s)R\left[-\cos\frac{L}{R}\tan\frac{L}{2R} - \cos\frac{L}{R}\cot\frac{L}{R}\frac{\eta_x(s)}{R} + (\cos\frac{L}{R} + \frac{1}{2}\sec^2\frac{L}{2R})\eta'(s)\right] \Big\} \\ w_y = \textstyle\sum_{s=1,2}^{D} \frac{1}{2}\gamma_y(s)R & \left[\sin\frac{L}{R} - \frac{L}{R} - \sin\frac{L}{R}\eta_x(s) + (1 - \cos\frac{L}{R})\eta_x'(s)\right]. \end{aligned}$$

The quantities in the summations are evaluated at the two quadrupoles in $C_{x,y}$ and at the ends of the two dipoles in $w_{x,y}$ respectively. In the limit of weak bending, we can expand $w_{x,y}$ in terms of $\epsilon_1 = \frac{L}{R}$ and get the leading term as

$$\begin{aligned} w_x &\simeq \epsilon_1 \textstyle\sum_s^D \alpha_x(s)\eta_x'(s) + \frac{1}{4}\gamma_x(s)(3L\eta_x'(s) - 2\eta_x(s)) \\ w_y &\simeq \epsilon_1 \textstyle\sum_s^D \frac{1}{4}\gamma_y(s)(L\eta_x'(s) - 2\eta_x(s)) \end{aligned} \tag{63}$$

To form a second order achromat, set $C_x = 0$ and $C_y = 0$, we get a set of linear equations for $\lambda_1$ and $\lambda_2$, which has solution

$$\begin{aligned} \lambda_1 &= \frac{1}{2\eta_x(1)f_1} + \frac{\beta_y(2)w_x + \beta_x(2)w_y}{\eta_x(1)w_\beta} \\ \lambda_2 &= \frac{1}{2\eta_x(2)f_2} - \frac{\beta_y(1)w_x + \beta_x(1)w_y}{\eta_x(2)w_\beta} \end{aligned} \tag{64}$$

where $w_\beta = \beta_x(1)\beta_y(2) - \beta_x(2)\beta_y(1)$. (equals $\beta_{max}^2 - \beta_{min}^2$ for a FODO cell without bending). The first terms in the solution are the donimant terms. $w_x$ and $w_y$ are due to the finite thickness of the bending magnets. Normally they are small. Note that the leading terms are proportional to the inverse of the dispersion function and the focal length of the quardrupoles. These shows clearly the well known fact that sectupoles work better in area with large dispersion(more exactly, $\eta_x\beta_x$ instead of $\eta_x$ only). Moreover, a strong focusing system needs strong sextupole to correct its chromaticities.

We can also express the sextupole strengths explicitly in terms of the lattice parameters as

$$\begin{aligned} \lambda_1 &\simeq \frac{12f^2(1+2f)^2 + 24f^2(1-4f^2)\varepsilon_2 - f(2+13f+31f^2+35f^3+24f^4+12f^5)\varepsilon_1^2 + 48(1+2f+2f^2-4f^3)\varepsilon_2^2}{6L^2\varepsilon_1 f^4(1+2f)^3} \\ \lambda_2 &\simeq \frac{12f^3(1-2f) + 24f^2(2-5f+2f^2)\varepsilon_2 - f^2(1+11f-13f^2+9f^3-6f^4)\varepsilon_1^2 + 48(1+2f-8f^2+4f^3)\varepsilon_2^2}{6L^2\varepsilon_1 f^5(-1+2f)^2} \end{aligned} \tag{65}$$

where the dimensionless parameters are given by

$$f = \frac{2f_1}{L}, \qquad \varepsilon_1 = \frac{L}{R}, \qquad \varepsilon_2 = \frac{f_1 + f_2}{L}. \tag{66}$$

Usually $\varepsilon_1$ and $\varepsilon_2$ are small. Eq.(65) has taken advantage of this fact, and is accurate up to the second order in $\varepsilon_{1,2}$

# 9 Third order achromats

The third order calculation is in principle the same as the second order except that it is much harder. We have to use Mathematica to do most of the algebras. In this section, we report the third order result without calculation details. As for the second order, we assume no resonance nearby. Therefore we can write the normal form of the third order generator for a unit cell as:

$$h_3 = C_{xx}J_x^2 + C_{xy}J_xJ_y + C_{yy}J_y^2 + C_{xd}J_x\delta^2 + C_{yd}J_y\delta^2 + C_{dd}\delta^4 \tag{67}$$

where

$$\begin{aligned} C_{xx} &= -\tfrac{3}{8}\textstyle\sum_{s=1}^{5} \beta_x(s)^2 O_s + w_{xx} \\ C_{xy} &= \ \ \tfrac{3}{2}\textstyle\sum_{s=1}^{5} \beta_x(s)\beta_y(s) O_s + w_{xy} \\ C_{yy} &= -\tfrac{3}{8}\textstyle\sum_{s=1}^{5} \beta_y(s)^2 O_s + w_{yy} \\ C_{xd} &= -\tfrac{3}{2}\textstyle\sum_{s=1}^{5} \beta_x(s)\eta_x(s)^2 O_s + w_{xd} \\ C_{yd} &= \ \ \tfrac{3}{2}\textstyle\sum_{s=1}^{5} \beta_y(s)\eta_x(s)^2 O_s + w_{yd} \end{aligned} \tag{68}$$

and

$$\begin{aligned} w_{xx} \simeq\ & -\tfrac{3L}{16}\textstyle\sum_s^D \gamma_x(s)^2 \ +\ \tfrac{1}{8}\csc\tfrac{3\mu_x}{2}(3\cos\tfrac{\mu_x}{2} + 2\cos\tfrac{3\mu_x}{2})\textstyle\sum_s^S \lambda_s^2\beta_x(s)^3 \ +\ \tfrac{1}{4}\csc\tfrac{3\mu_x}{2}(2+3\cos\mu_x)\textstyle\prod_s^S \lambda_s\beta_x(s)^{\frac{3}{2}} \\ w_{xy} \simeq\ & -\tfrac{L}{4}\textstyle\sum_s^D \gamma_x(s)\gamma_y(s) \ -\ \tfrac{1}{4}\csc(\tfrac{\mu_x}{2}+\mu_y)\csc(\tfrac{\mu_x}{2}-\mu_y)\sin 2\mu_y \textstyle\sum_s^S \lambda_s^2\beta_x(s)\beta_y(s)^2 - \\ & \tfrac{1}{2}\cot\tfrac{\mu_x}{2}\textstyle\sum_s^S \lambda_s^2\beta_x(s)^2\beta_y(s) \ +\ \tfrac{1}{2}\left[\csc(\tfrac{\mu_x}{2}+\mu_y) - \csc(\tfrac{\mu_x}{2}-\mu_y) - \csc\tfrac{\mu_x}{2}\textstyle\sum_s^S \tfrac{\beta_x(s)}{\beta_y(s)}\right]\textstyle\prod_s^S \lambda_s\sqrt{\beta_x(s)}\beta_y(s) \\ w_{yy} \simeq\ & -\tfrac{3L}{16}\textstyle\sum_s^D \gamma_y^2(s) \ +\ \tfrac{1}{16}\left[4\cot\tfrac{\mu_x}{2} + \sin\mu_x\csc(\tfrac{\mu_x}{2}+\mu_y)\csc(\tfrac{\mu_x}{2}-\mu_y)\right]\textstyle\sum_s^S \lambda_s^2\beta_x(s)\beta_y(s)^2 + \\ & -\tfrac{1}{8}\left[4\csc\tfrac{\mu_x}{2} + \csc(\tfrac{\mu_x}{2}+\mu_y) + \csc(\tfrac{\mu_x}{2}-\mu_y)\right]\textstyle\prod_s^S \lambda_s\sqrt{\beta_x(s)}\beta_y(s) \\ w_{xd} \simeq\ & -\tfrac{3L}{4}\textstyle\sum_s^D \gamma_x(s)\eta_x'(s)^2 \ -\ \textstyle\sum_s^S \beta_x(s)(\tfrac{1}{2f_s} - \lambda_s\beta_x(s)) \ +\ \tfrac{1}{2}\cot\mu_x\textstyle\sum_s^S \left[\beta_x(s)(\tfrac{1}{2f_s} - \lambda_s\beta_x(s))\right]^2 + \\ & \csc\mu_x\textstyle\prod_s^S \beta_x(s)(\tfrac{1}{2f_s} - \lambda_s\beta_x(s)) \ +\ \tfrac{1}{2}\csc\tfrac{\mu_x}{2}\textstyle\sum_{1,2}^S \lambda_2\beta_x(2)\eta_x(1)(\lambda_1\eta_x(1) - \tfrac{1}{f_1})\sqrt{\beta_x(1)\beta_x(2)} + \\ & \tfrac{1}{2}\cot\tfrac{\mu_x}{2}\textstyle\sum_s^S \lambda_s\eta_x(s)(\lambda_s\eta_x(s) - \tfrac{1}{f_s})\beta_x(s)^2 \\ w_{yd} \simeq\ & -\tfrac{L}{4}\textstyle\sum_s^D \gamma_y(s)\eta_x'(s)^2 \ +\ \textstyle\sum_s^S \beta_y(s)(\tfrac{1}{2f_s} - \lambda_s\beta_x(s)) \ +\ \tfrac{1}{2}\cot\mu_y\textstyle\sum_s^S \left[\beta_y(s)(\tfrac{1}{2f_s} - \lambda_s\beta_x(s))\right]^2 + \\ & \csc\mu_y\textstyle\prod_s^S \beta_y(s)(\tfrac{1}{2f_s} - \lambda_s\beta_x(s)) \ -\ \tfrac{1}{2}\csc\tfrac{\mu_x}{2}\textstyle\sum_{1,2}^S \lambda_2\beta_y(2)\eta_x(1)(\lambda_1\eta_x(1) - \tfrac{1}{f_1})\sqrt{\beta_x(1)\beta_x(2)} - \\ & \tfrac{1}{2}\cot\tfrac{\mu_x}{2}\textstyle\sum_s^S \lambda_s\eta_x(s)(\lambda_s\eta_x(s) - \tfrac{1}{f_s})\beta_x(s)\beta_y(s) \end{aligned} \tag{69}$$

These expressions are approximate and similar to Eq.(63) instead of Eq.(62). We do not present the exact expressions of $w_{**}$ because they contain a couple of hundred terms and are difficult to read (the results are available). Next order terms (unformatted) are included in the appendix. It is necessary to take the next order into account for $w_{xd}$ and $w_{yd}$. After including the next order terms, these approximate forms have very good accuracy. The analytical form reveals the complicated dependency on linear lattice as well as nonlinear parameters, which are instructive information for the design of achromats. Although the exact results are too complicated to be presented here, they are good for practical calculations. Computer files containing these coefficients are available from the authors.

To obtain the required octupole strengths, simply set the five coefficients $C_{**}$ in Eq.(68) equal to zero and solve the five linear equations. The existence of solution depends on a nonzero determinant of the coefficient matrix of octupole strengths $O_s$, which are determined by the beta and dispersion functions at the locations of those octupoles. A general form of solution will not be useful due to its complexity. In the following, we present a specific solution with the octupole layout as shown in Fig. 1, i.e. two octupoles are following the two sextupoles and the other three at $\frac{1}{3}$, $\frac{2}{3}$, and $\frac{1}{2}$ of the two bending magnets. In this case, the octupole strengths reads

$$O_1 \simeq \frac{a+b}{6f^3D}, \quad O_2 \simeq \frac{81(c+d)}{2fD}, \quad O_3 \simeq \frac{81(c-d)}{2fD}, \quad O_4 \simeq \frac{a-b}{6f^3D}, \quad O_5 \simeq \frac{128e}{3(2f^2-1)D} \tag{70}$$

where

$$\begin{aligned}
a &= 2f(1360 - 22846f^2 - 74476f^4 + 695809f^6 - 1438146f^8 + 1200096f^{10} - 326592f^{12}) \\
b &= -352 - 3360f^2 + 233290f^4 - 1070910f^6 + 1917603f^8 - 1364850f^{10} + 361584f^{12} \\
c &= 6f(-42 + 1076f^2 - 7409f^4 + 16306f^6 - 14368f^8 + 4032f^{10}) \\
d &= 8 - 394f^2 + 5322f^4 - 16907f^6 + 14866f^8 - 4464f^{10} \\
e &= -368 + 10536f^2 - 92342f^4 + 307222f^6 - 470547f^8 + 330642f^{10} - 81648f^{12} \\
D &= (4f^2-1)^2(3f^2-4)(10 - 173f^2 - 261f^4 + 324f^6)L^3\epsilon_1^2
\end{aligned}$$

Eq.(70) contains leading terms only. The next order in $\epsilon_1$ and $\epsilon_2$ may be necessary and is available from the authors.

# 10 Conclusion remarks

Our analytical results on the second and third order achromats show explicitly the sextupole and octupole strengths required to form an achromat for a given repetitive FODO structure working away from resonances. The dependency on the linear lattice and nonlinear parameters that revealed in this report should be useful for achromat designs. Numerical examples calculated using our results are checked with MARYLIE, and they agree with each other very well (see appendix).

No approximation (up to forth Hamiltonian order) is used in our calculation following the single element Hamiltonians. However, the Hamiltonians include thin lens approximation for the multipoles and neglect any fringe fields. These approximations may affect the accuracy of our results, especially at the third order. Non-perfect repetitive symmetry due to field errors will affect the result also[8]. However, we expect they provide reasonably good estimate for a wide range of achromat layouts. Further improvement to accommodate more practical considerations, especially important cases that involve resonance conditions (e.g. unit cell with 90° phase advances), is possible and may be considered in the future. Another interesting direction is to consider symmetries other than the repetition, such as mirror symmetry[15, 16].

# Acknowledgement

We would like to thank Y. T. Yan of SLAC for his help on using his Zlib package as well as H. Ye of the University of Maryland for his help on setting up MARYLIE code. The author C. X. Wang is grateful to J. Irwin for his support. This work is partly supported by the Applied Physics Department of Stanford University.

# References

[1] K.L. Brown, First- and second-order charged particle optics, SLAC-PUB-3381

[2] A. W. Chao, Notes on nonlinear dynamics, (unpublished)

[3] E.D. Courant and H.S. Snyder, Theory of the alternating- gradient synchrotron, Ann. of Phys. 3,1(1958)

[4] A. J. Dragt, Lectures on nonlinear orbit dynamics, Physics of high energy particle accelerators, AIP conf. Proc. No.87(1982)

[5] A. J. Dragt and J. M. Finn, Lie series and invariant functions for analytic symplectic maps, J. Math. Phys. 17, 2215(1976)

[6] A. J. Dragt and E. Forest, Computation of nonlinear behavior of Hamiltonian systems using Lie algebraic methods, J.Math. Phys.24(12)1983

[7] A. J. Dragt, Elementary and advanced Lie algebraic methods with applications to accelerator design, electron microscopes, and light optics. NIM A258(1987)339-354

[8] E. Forest, D. Douglas and B. Leemann, Study of the aberrations of periodic arc using the Lie algebraic techniques, NIM A258(1987)355-363

[9] E. Forest, Normal form algorithm on non-linear symplectic maps, SSC-29

[10] J. Irwin, The application of Lie algebra techniques to beam transport design, NIM A298(1990)460-472

[11] J. Irwin, The application of Lie algebraic methods to the of linear and circular colliders, (to be published)

[12] J. Irwin, nPB tracking

[13] R. Ruth, Single particle dynamics in circular accelerators, SLAC-PUB-4103

[14] N. J. Walker, J. Irwin, M. Woodley, Global tuning knobs for the SLC final focus, IEEE 1993 Vol.1 P.95

[15] W. Wan, E. Goldmann and M. Berz, the design of a four cell third order achromat, Proceedings 1993 Particle Accelerator Conference

[16] W. Wan and M. Berz, Four cell third order achromats and their application to multi-pass time-of flight spectrometers, Proceedings 1993 Particle Accelerator Conference

[17] S. Wolfram, Mathematica, 2nd ed. Addison-Wesley 1991

# Appendix A

The following is the approximate expressions for $w_{xx}$, $w_{xy}$, $w_{yy}$, $w_{xd}$, and $w_{yd}$. The leading term is presented in the previous section.

```
wxx = HoldForm[wxx0] + t*HoldForm[wxx1]

wxx0 = (S1*S2*betax[0]^(3/2)*betax[3]^(3/2)*(2 + 3*Cos[mux])*Csc[(3*mux)/2])/
   4 + (S1^2*betax[0]^3*(3*Cos[mux/2] + 2*Cos[(3*mux)/2])*Csc[(3*mux)/2])/
   8 + (S2^2*betax[3]^3*(3*Cos[mux/2] + 2*Cos[(3*mux)/2])*Csc[(3*mux)/2])/
   8 - (3*L*gammax[3]^2)/16 - (3*L*gammax[6]^2)/16

wxx1 = (-3*L*(S1 + S2))/8 + (3*S1*alphax[6]*betax[0])/4 +
   (3*S2*alphax[3]*betax[3])/4 - (3*L*S2*alphax[3]*Cot[(3*mux)/2])/8 -
   (3*L*S1*alphax[6]*Cot[(3*mux)/2])/8 -
   (3*L*S1*alphax[3]*betax[0]^(3/2)*Csc[(3*mux)/2])/(8*betax[3]^(3/2)) -
   (3*L*S2*alphax[6]*betax[3]^(3/2)*Csc[(3*mux)/2])/(8*betax[0]^(3/2)) -
   (3*S1*betax[0]^(3/2)*Cos[mux/2]^2*Csc[(3*mux)/2])/(2*betax[3]^(1/2)) -
   (3*S2*betax[3]^(3/2)*Cos[mux/2]^2*Csc[(3*mux)/2])/(2*betax[0]^(1/2)) -
   (3*S1*betax[0]*Cos[mux/2]*Cos[mux]*Csc[(3*mux)/2])/2 -
   (3*S2*betax[3]*Cos[mux/2]*Cos[mux]*Csc[(3*mux)/2])/2 +
   (9*L*S2*betax[3]*gammax[3])/16 +
   (3*L*S1*alphax[3]*betax[0]^(3/2)*(2 + 3*Cos[mux])*Csc[(3*mux)/2]*
      gammax[3])/(16*betax[3]^(1/2)) +
   (3*S1*betax[0]^(3/2)*betax[3]^(1/2)*(2 + 3*Cos[mux])*Csc[(3*mux)/2]*
      gammax[3])/8 + (3*L*S2*alphax[3]*betax[3]*
      (3*Cos[mux/2] + 2*Cos[(3*mux)/2])*Csc[(3*mux)/2]*gammax[3])/16 +
   (3*S2*betax[3]^2*(3*Cos[mux/2] + 2*Cos[(3*mux)/2])*Csc[(3*mux)/2]*
      gammax[3])/8 + (9*L*S1*betax[0]*gammax[6])/16 +
   (3*L*S2*alphax[6]*betax[3]^(3/2)*(2 + 3*Cos[mux])*Csc[(3*mux)/2]*
      gammax[6])/(16*betax[0]^(1/2)) +
   (3*S2*betax[0]^(1/2)*betax[3]^(3/2)*(2 + 3*Cos[mux])*Csc[(3*mux)/2]*
      gammax[6])/8 + (3*L*S1*alphax[6]*betax[0]*
      (3*Cos[mux/2] + 2*Cos[(3*mux)/2])*Csc[(3*mux)/2]*gammax[6])/16 +
   (3*S1*betax[0]^2*(3*Cos[mux/2] + 2*Cos[(3*mux)/2])*Csc[(3*mux)/2]*
      gammax[6])/8

wxy = HoldForm[wxy0] + t*HoldForm[wxy1]

wxy0 = -(S1^2*betax[0]^2*betay[0]*Cot[mux/2])/2 -
   (S2^2*betax[3]^2*betay[3]*Cot[mux/2])/2 -
   (S1*S2*betax[0]^(1/2)*betax[3]^(3/2)*betay[0]*Csc[mux/2])/2 -
   (S1*S2*betax[0]^(3/2)*betax[3]^(1/2)*betay[3]*Csc[mux/2])/2 +
   betax[0]^(1/2)*betax[3]^(1/2)*betay[0]*betay[3]*
    (-(S1*S2*Csc[mux/2 - muy])/2 + (S1*S2*Csc[mux/2 + muy])/2) -
   (L*gammax[3]*gammay[1])/4 - (L*gammax[6]*gammay[4])/4 -
   (S1^2*betax[0]*betay[0]^2*Csc[mux/2 - muy]*Csc[mux/2 + muy]*Sin[2*muy])/
    4 - (S2^2*betax[3]*betay[3]^2*Csc[mux/2 - muy]*Csc[mux/2 + muy]*
      Sin[2*muy])/4

wxy1 = -(L*S2*alphax[3]*alphay[3])/2 - (L*S1*alphax[6]*alphay[6])/2 -
   S1*alphay[6]*betax[0] - S2*alphay[3]*betax[3] -
   (S1*alphax[6]*betay[0])/2 - (S2*alphax[3]*betay[3])/2 +
   (S1*betay[0]*Cot[mux/2])/2 + (S2*betay[3]*Cot[mux/2])/2 +
   (S1*betax[0]^(1/2)*betay[0]*Csc[mux/2])/(2*betax[3]^(1/2)) +
   (S2*betax[3]^(1/2)*betay[3]*Csc[mux/2])/(2*betax[0]^(1/2)) -
   (3*L*S2*betay[3]*gammax[3])/8 -
   (3*L*S2*alphax[3]*betay[3]*Cot[mux/2]*gammax[3])/8 -
   (3*S2*betax[3]*betay[3]*Cot[mux/2]*gammax[3])/4 -
   (3*L*S1*alphax[3]*betax[0]^(1/2)*betay[0]*Csc[mux/2]*gammax[3])/
    (8*betax[3]^(1/2)) - (3*S1*betax[0]^(1/2)*betax[3]^(1/2)*betay[0]*
      Csc[mux/2]*gammax[3])/4 - (3*L*S1*betay[0]*gammax[6])/8 -
   (3*L*S1*alphax[6]*betay[0]*Cot[mux/2]*gammax[6])/8 -
```

```
    (3*S1*betax[0]*betay[0]*Cot[mux/2]*gammax[6])/4 -
    (3*L*S2*alphax[6]*betax[3]^(1/2)*betay[3]*Csc[mux/2]*gammax[6])/
     (8*betax[0]^(1/2)) - (3*S2*betax[0]^(1/2)*betax[3]^(1/2)*betay[3]*
       Csc[mux/2]*gammax[6])/4 + (L*S2*betax[3]*gammay[1])/8 +
    (L*S2*alphax[3]*betax[3]*Cot[mux/2]*gammay[1])/8 +
    (S2*betax[3]^2*Cot[mux/2]*gammay[1])/4 +
    (L*S1*alphax[3]*betax[0]^(3/2)*Csc[mux/2]*gammay[1])/(8*betax[3]^(1/2)) +
    (S1*betax[0]^(3/2)*betax[3]^(1/2)*Csc[mux/2]*gammay[1])/4 +
    (L*S1*betax[0]*gammay[4])/8 +
    (L*S1*alphax[6]*betax[0]*Cot[mux/2]*gammay[4])/8 +
    (S1*betax[0]^2*Cot[mux/2]*gammay[4])/4 +
    (L*S2*alphax[6]*betax[3]^(3/2)*Csc[mux/2]*gammay[4])/(8*betax[0]^(1/2)) +
    (S2*betax[0]^(1/2)*betax[3]^(3/2)*Csc[mux/2]*gammay[4])/4 +
    (L*S2*alphay[3]*Csc[mux/2 - muy]*Csc[mux/2 + muy]*Sin[mux])/4 +
    (L*S1*alphay[6]*Csc[mux/2 - muy]*Csc[mux/2 + muy]*Sin[mux])/4 -
    (L*S2*alphay[3]*Csc[mux/2 - muy]*Csc[mux/2 + muy]*Sin[2*muy])/4 -
    (L*S1*alphay[6]*Csc[mux/2 - muy]*Csc[mux/2 + muy]*Sin[2*muy])/4 -
    (S1*betax[0]*Csc[mux/2 - muy]*Csc[mux/2 + muy]*Sin[2*muy])/2 -
    (S2*betax[3]*Csc[mux/2 - muy]*Csc[mux/2 + muy]*Sin[2*muy])/2 +
    (L*S2*alphax[3]*betay[3]*Csc[mux/2 - muy]*Csc[mux/2 + muy]*gammay[1]*
       Sin[2*muy])/8 + (S2*betax[3]*betay[3]*Csc[mux/2 - muy]*
       Csc[mux/2 + muy]*gammay[1]*Sin[2*muy])/4 +
    (L*S1*alphax[6]*betay[0]*Csc[mux/2 - muy]*Csc[mux/2 + muy]*gammay[4]*
       Sin[2*muy])/8 + (S1*betax[0]*betay[0]*Csc[mux/2 - muy]*Csc[mux/2 + muy]*
       gammay[4]*Sin[2*muy])/4

wyy = HoldForm[wyy0] + t*HoldForm[wyy1]

wyy0 = (S1^2*betax[0]*betay[0]^2*(1 - 3*Cos[mux] + 2*Cos[2*muy])*Cot[mux/2]*
       Csc[mux/2 - muy]*Csc[mux/2 + muy])/16 +
    (S2^2*betax[3]*betay[3]^2*(1 - 3*Cos[mux] + 2*Cos[2*muy])*Cot[mux/2]*
       Csc[mux/2 - muy]*Csc[mux/2 + muy])/16 +
    (S1*S2*betax[0]^(1/2)*betax[3]^(1/2)*betay[0]*betay[3]*
       (-4*Cos[mux] - Cos[mux - muy] + 2*Cos[muy] + 4*Cos[2*muy] -
         Cos[mux + muy])*Csc[mux/2]*Csc[mux/2 - muy]*Csc[mux/2 + muy])/16 -
    (3*L*gammay[1]^2)/16 - (3*L*gammay[4]^2)/16

wyy1 = (L*(S1 + S2))/8 + (L*S1*alphax[3]*betax[0]^(1/2)*betay[0]*
       Csc[mux/2 - muy])/(16*betax[3]^(1/2)*betay[3]) -
    (L*S1*alphay[3]*betax[0]^(1/2)*betay[0]*Csc[mux/2 - muy])/
     (16*betax[3]^(1/2)*betay[3]) +
    (S1*betax[0]^(1/2)*betax[3]^(1/2)*betay[0]*Csc[mux/2 - muy])/
     (8*betay[3]) + (L*S2*alphax[6]*betax[3]^(1/2)*betay[3]*Csc[mux/2 - muy])/
     (16*betax[0]^(1/2)*betay[0]) -
    (L*S2*alphay[6]*betax[3]^(1/2)*betay[3]*Csc[mux/2 - muy])/
     (16*betax[0]^(1/2)*betay[0]) +
    (S2*betax[0]^(1/2)*betax[3]^(1/2)*betay[3]*Csc[mux/2 - muy])/
     (8*betay[0]) + (L*S1*alphax[3]*betax[0]^(1/2)*betay[0]*Csc[mux/2 + muy])/
     (16*betax[3]^(1/2)*betay[3]) +
    (L*S1*alphay[3]*betax[0]^(1/2)*betay[0]*Csc[mux/2 + muy])/
     (16*betax[3]^(1/2)*betay[3]) +
    (S1*betax[0]^(1/2)*betax[3]^(1/2)*betay[0]*Csc[mux/2 + muy])/
     (8*betay[3]) + (L*S2*alphax[6]*betax[3]^(1/2)*betay[3]*Csc[mux/2 + muy])/
     (16*betax[0]^(1/2)*betay[0]) +
    (L*S2*alphay[6]*betax[3]^(1/2)*betay[3]*Csc[mux/2 + muy])/
     (16*betax[0]^(1/2)*betay[0]) +
    (S2*betax[0]^(1/2)*betax[3]^(1/2)*betay[3]*Csc[mux/2 + muy])/
     (8*betay[0]) - (3*L*S2*betay[3]*gammay[1])/16 +
    (L*S2*alphax[3]*betay[3]*(-1 + 3*Cos[mux] - 2*Cos[2*muy])*Cot[mux/2]*
       Csc[mux/2 - muy]*Csc[mux/2 + muy]*gammay[1])/32 +
    (S2*betax[3]*betay[3]*(-1 + 3*Cos[mux] - 2*Cos[2*muy])*Cot[mux/2]*
       Csc[mux/2 - muy]*Csc[mux/2 + muy]*gammay[1])/16 +
    (L*S1*alphax[3]*betax[0]^(1/2)*betay[0]*
       (4*Cos[mux] + Cos[mux - muy] - 2*Cos[muy] - 4*Cos[2*muy] +
         Cos[mux + muy])*Csc[mux/2]*Csc[mux/2 - muy]*Csc[mux/2 + muy]*
       gammay[1])/(64*betax[3]^(1/2)) +
```

```
    (S1*betax[0]^(1/2)*betax[3]^(1/2)*betay[0]*
       (4*Cos[mux] + Cos[mux - muy] - 2*Cos[muy] - 4*Cos[2*muy] +
         Cos[mux + muy])*Csc[mux/2]*Csc[mux/2 - muy]*Csc[mux/2 + muy]*
      gammay[1])/32 - (3*L*S1*betay[0]*gammay[4])/16 +
    (L*S1*alphax[6]*betay[0]*(-1 + 3*Cos[mux] - 2*Cos[2*muy])*Cot[mux/2]*
      Csc[mux/2 - muy]*Csc[mux/2 + muy]*gammay[4])/32 +
    (S1*betax[0]*betay[0]*(-1 + 3*Cos[mux] - 2*Cos[2*muy])*Cot[mux/2]*
      Csc[mux/2 - muy]*Csc[mux/2 + muy]*gammay[4])/16 +
    (L*S2*alphax[6]*betax[3]^(1/2)*betay[3]*
       (4*Cos[mux] + Cos[mux - muy] - 2*Cos[muy] - 4*Cos[2*muy] +
         Cos[mux + muy])*Csc[mux/2]*Csc[mux/2 - muy]*Csc[mux/2 + muy]*
      gammay[4])/(64*betax[0]^(1/2)) +
    (S2*betax[0]^(1/2)*betax[3]^(1/2)*betay[3]*
       (4*Cos[mux] + Cos[mux - muy] - 2*Cos[muy] - 4*Cos[2*muy] +
         Cos[mux + muy])*Csc[mux/2]*Csc[mux/2 - muy]*Csc[mux/2 + muy]*
      gammay[4])/32 + (L*S2*alphax[3]*Csc[mux/2 - muy]*Csc[mux/2 + muy]*
      Sin[mux])/16 + (L*S1*alphax[6]*Csc[mux/2 - muy]*Csc[mux/2 + muy]*
      Sin[mux])/16 + (S1*betax[0]*Csc[mux/2 - muy]*Csc[mux/2 + muy]*
      Sin[mux])/8 + (S2*betax[3]*Csc[mux/2 - muy]*Csc[mux/2 + muy]*Sin[mux])/
     8 - (L*S2*alphay[3]*Csc[mux/2 - muy]*Csc[mux/2 + muy]*Sin[2*muy])/16 -
    (L*S1*alphay[6]*Csc[mux/2 - muy]*Csc[mux/2 + muy]*Sin[2*muy])/16

wxd = HoldForm[wxd0] + t*HoldForm[wxd1]

wxd0 = -betax[0]/(2*f1) - betax[3]/(2*f2) + (betax[0]^2*Cot[mux])/(8*f1^2) +
    (betax[3]^2*Cot[mux])/(8*f2^2) + (betax[0]*betax[3]*Csc[mux])/(4*f1*f2) +
    S1*betax[0]*etax[0] - (S2*betax[0]^(1/2)*betax[3]^(3/2)*Csc[mux/2]*
       etax[0])/(2*f1) - (S1*betax[0]*betax[3]*Csc[mux]*etax[0])/(2*f2) -
    (S1*betax[0]^2*(1 + 2*Cos[mux])*Csc[mux]*etax[0])/(2*f1) +
    (S1*S2*betax[0]^(1/2)*betax[3]^(3/2)*Csc[mux/2]*etax[0]^2)/2 +
    (S1^2*betax[0]^2*(1 + 2*Cos[mux])*Csc[mux]*etax[0]^2)/2 +
    S2*betax[3]*etax[3] - (S1*betax[0]^(3/2)*betax[3]^(1/2)*Csc[mux/2]*
       etax[3])/(2*f2) - (S2*betax[0]*betax[3]*Csc[mux]*etax[3])/(2*f1) -
    (S2*betax[3]^2*(1 + 2*Cos[mux])*Csc[mux]*etax[3])/(2*f2) +
    S1*S2*betax[0]*betax[3]*Csc[mux]*etax[0]*etax[3] +
    (S1*S2*betax[0]^(3/2)*betax[3]^(1/2)*Csc[mux/2]*etax[3]^2)/2 +
    (S2^2*betax[3]^2*(1 + 2*Cos[mux])*Csc[mux]*etax[3]^2)/2 -
    (3*L*etaxp[3]^2*gammax[3])/4 - (3*L*etaxp[6]^2*gammax[6])/4

wxd1 = (L*S1*betax[0])/4 + (L*S2*betax[3])/4 +
    (L*S1*alphax[6]*betax[0]*Cot[mux/2])/4 + (S1*betax[0]^2*Cot[mux/2])/2 +
    (L*S2*alphax[3]*betax[3]*Cot[mux/2])/4 + (S2*betax[3]^2*Cot[mux/2])/2 +
    (L*S1*alphax[3]*betax[0]^(3/2)*Csc[mux/2])/(4*betax[3]^(1/2)) +
    (S1*betax[0]^(3/2)*betax[3]^(1/2)*Csc[mux/2])/2 +
    (L*S2*alphax[6]*betax[3]^(3/2)*Csc[mux/2])/(4*betax[0]^(1/2)) +
    (S2*betax[0]^(1/2)*betax[3]^(3/2)*Csc[mux/2])/2 - (alphax[6]*etax[0])/f1 +
    (betax[0]^(1/2)*Csc[mux/2]*etax[0])/(2*f1*betax[3]^(1/2)) +
    (betax[3]*Csc[mux]*etax[0])/(2*f2*betax[0]) +
    ((1 + 2*Cos[mux])*Csc[mux]*etax[0])/(2*f1) +
    (3*S1*alphax[6]*etax[0]^2)/2 -
    (S1*betax[0]^(1/2)*Csc[mux/2]*etax[0]^2)/(2*betax[3]^(1/2)) -
    (S1*(1 + 3*Cos[mux])*Csc[mux]*etax[0]^2)/2 - (alphax[3]*etax[3])/f2 +
    (betax[3]^(1/2)*Csc[mux/2]*etax[3])/(2*f2*betax[0]^(1/2)) +
    (betax[0]*Csc[mux]*etax[3])/(2*f1*betax[3]) +
    ((1 + 2*Cos[mux])*Csc[mux]*etax[3])/(2*f2) -
    (S1*betax[0]*Csc[mux]*etax[0]*etax[3])/betax[3] -
    (S2*betax[3]*Csc[mux]*etax[0]*etax[3])/betax[0] +
    (3*S2*alphax[3]*etax[3]^2)/2 -
    (S2*betax[3]^(1/2)*Csc[mux/2]*etax[3]^2)/(2*betax[0]^(1/2)) -
    (S2*(1 + 3*Cos[mux])*Csc[mux]*etax[3]^2)/2 +
    (3*L*alphax[3]*etaxp[3])/(4*f2) + (betax[3]*etaxp[3])/(2*f2) -
    (3*L*Cot[mux]*etaxp[3])/(4*f2) +
    (alphax[3]*betax[0]*Csc[mux]*etaxp[3])/(2*f1) -
    (3*L*betax[0]*Csc[mux]*etaxp[3])/(4*f1*betax[3]) -
    S1*alphax[3]*betax[0]*Csc[mux]*etax[0]*etaxp[3] +
    (3*L*S1*betax[0]*Csc[mux]*etax[0]*etaxp[3])/(2*betax[3]) -
```

```
(3*L*S2*alphax[3]*etax[3]*etaxp[3])/2 -
(3*S2*betax[3]*etax[3]*etaxp[3])/2 +
(3*L*S2*Cot[mux]*etax[3]*etaxp[3])/2 -
(S1*alphax[3]*betax[0]^(3/2)*Csc[mux/2]*etax[3]*etaxp[3])/
 (2*betax[3]^(1/2)) - (S2*alphax[3]*betax[3]*(1 + 3*Cos[mux])*Csc[mux]*
   etax[3]*etaxp[3])/2 + (3*L*S2*betax[3]*etaxp[3]^2)/8 +
(3*L*S2*alphax[3]*betax[3]*Cot[mux/2]*etaxp[3]^2)/8 +
(S2*betax[3]^2*Cot[mux/2]*etaxp[3]^2)/4 +
(3*L*S1*alphax[3]*betax[0]^(3/2)*Csc[mux/2]*etaxp[3]^2)/
 (8*betax[3]^(1/2)) + (S1*betax[0]^(3/2)*betax[3]^(1/2)*Csc[mux/2]*
   etaxp[3]^2)/4 + (3*L*alphax[6]*etaxp[6])/(4*f1) +
(betax[0]*etaxp[6])/(2*f1) - (3*L*Cot[mux]*etaxp[6])/(4*f1) +
(alphax[6]*betax[3]*Csc[mux]*etaxp[6])/(2*f2) -
(3*L*betax[3]*Csc[mux]*etaxp[6])/(4*f2*betax[0]) -
(3*L*S1*alphax[6]*etax[0]*etaxp[6])/2 -
(3*S1*betax[0]*etax[0]*etaxp[6])/2 +
(3*L*S1*Cot[mux]*etax[0]*etaxp[6])/2 -
(S2*alphax[6]*betax[3]^(3/2)*Csc[mux/2]*etax[0]*etaxp[6])/
 (2*betax[0]^(1/2)) - (S1*alphax[6]*betax[0]*(1 + 3*Cos[mux])*Csc[mux]*
   etax[0]*etaxp[6])/2 - S2*alphax[6]*betax[3]*Csc[mux]*etax[3]*etaxp[6] +
(3*L*S2*betax[3]*Csc[mux]*etax[3]*etaxp[6])/(2*betax[0]) +
(3*L*S1*betax[0]*etaxp[6]^2)/8 +
(3*L*S1*alphax[6]*betax[0]*Cot[mux/2]*etaxp[6]^2)/8 +
(S1*betax[0]^2*Cot[mux/2]*etaxp[6]^2)/4 +
(3*L*S2*alphax[6]*betax[3]^(3/2)*Csc[mux/2]*etaxp[6]^2)/
 (8*betax[0]^(1/2)) + (S2*betax[0]^(1/2)*betax[3]^(3/2)*Csc[mux/2]*
   etaxp[6]^2)/4 - (3*L*alphax[3]*betax[0]^(1/2)*Csc[mux/2]*etax[0]*
   gammax[3])/(8*f1*betax[3]^(1/2)) -
(3*betax[0]^(1/2)*betax[3]^(1/2)*Csc[mux/2]*etax[0]*gammax[3])/(4*f1) +
(3*L*S1*alphax[3]*betax[0]^(1/2)*Csc[mux/2]*etax[0]^2*gammax[3])/
 (8*betax[3]^(1/2)) + (3*S1*betax[0]^(1/2)*betax[3]^(1/2)*Csc[mux/2]*
   etax[0]^2*gammax[3])/4 - (3*L*etax[3]*gammax[3])/(8*f2) -
(3*L*alphax[3]*Cot[mux/2]*etax[3]*gammax[3])/(8*f2) -
(betax[0]*Csc[mux]*etax[3]*gammax[3])/(4*f1) -
(betax[3]*(3 + 4*Cos[mux])*Csc[mux]*etax[3]*gammax[3])/(4*f2) +
(S1*betax[0]*Csc[mux]*etax[0]*etax[3]*gammax[3])/2 +
(3*L*S2*etax[3]^2*gammax[3])/8 +
(3*L*S2*alphax[3]*Cot[mux/2]*etax[3]^2*gammax[3])/8 +
(S2*betax[3]*(3 + 5*Cos[mux])*Csc[mux]*etax[3]^2*gammax[3])/4 +
(3*L*etaxp[3]*gammax[3])/4 +
(3*L*betax[3]*Cot[mux]*etaxp[3]*gammax[3])/(8*f2) +
(3*L*betax[0]*Csc[mux]*etaxp[3]*gammax[3])/(8*f1) -
(3*L*S1*betax[0]*Csc[mux]*etax[0]*etaxp[3]*gammax[3])/4 -
(3*L*S2*betax[3]*Cot[mux]*etax[3]*etaxp[3]*gammax[3])/4 -
(3*L*etax[0]*gammax[6])/(8*f1) -
(3*L*alphax[6]*Cot[mux/2]*etax[0]*gammax[6])/(8*f1) -
(betax[3]*Csc[mux]*etax[0]*gammax[6])/(4*f2) -
(betax[0]*(3 + 4*Cos[mux])*Csc[mux]*etax[0]*gammax[6])/(4*f1) +
(3*L*S1*etax[0]^2*gammax[6])/8 +
(3*L*S1*alphax[6]*Cot[mux/2]*etax[0]^2*gammax[6])/8 +
(S1*betax[0]*(3 + 5*Cos[mux])*Csc[mux]*etax[0]^2*gammax[6])/4 -
(3*L*alphax[6]*betax[3]^(1/2)*Csc[mux/2]*etax[3]*gammax[6])/
 (8*f2*betax[0]^(1/2)) - (3*betax[0]^(1/2)*betax[3]^(1/2)*Csc[mux/2]*
   etax[3]*gammax[6])/(4*f2) +
(S2*betax[3]*Csc[mux]*etax[0]*etax[3]*gammax[6])/2 +
(3*L*S2*alphax[6]*betax[3]^(1/2)*Csc[mux/2]*etax[3]^2*gammax[6])/
 (8*betax[0]^(1/2)) + (3*S2*betax[0]^(1/2)*betax[3]^(1/2)*Csc[mux/2]*
   etax[3]^2*gammax[6])/4 + (3*L*etaxp[6]*gammax[6])/4 +
(3*L*betax[0]*Cot[mux]*etaxp[6]*gammax[6])/(8*f1) +
(3*L*betax[3]*Csc[mux]*etaxp[6]*gammax[6])/(8*f2) -
(3*L*S1*betax[0]*Cot[mux]*etax[0]*etaxp[6]*gammax[6])/4 -
(3*L*S2*betax[3]*Csc[mux]*etax[3]*etaxp[6]*gammax[6])/4 +
(alphax[3]*betax[3]*Cos[mux]*Csc[mux/2]*etaxp[3]*Sec[mux/2])/(4*f2) +
(alphax[6]*betax[0]*Cos[mux]*Csc[mux/2]*etaxp[6]*Sec[mux/2])/(4*f1)

wyd = HoldForm[wyd0] + t*HoldForm[wyd1]
```

```
wyd0 = betay[0]/(2*f1) + betay[3]/(2*f2) + (betay[0]^2*Cot[muy])/(8*f1^2) +
   (betay[3]^2*Cot[muy])/(8*f2^2) + (betay[0]*betay[3]*Csc[muy])/(4*f1*f2) -
   S1*betay[0]*etax[0] + (S1*betax[0]*betay[0]*Cot[mux/2]*etax[0])/(2*f1) -
   (S1*betay[0]^2*Cot[muy]*etax[0])/(2*f1) +
   (S2*betax[0]^(1/2)*betax[3]^(1/2)*betay[3]*Csc[mux/2]*etax[0])/(2*f1) -
   (S1*betay[0]*betay[3]*Csc[muy]*etax[0])/(2*f2) -
   (S1^2*betax[0]*betay[0]*Cot[mux/2]*etax[0]^2)/2 +
   (S1^2*betay[0]^2*Cot[muy]*etax[0]^2)/2 -
   (S1*S2*betax[0]^(1/2)*betax[3]^(1/2)*betay[3]*Csc[mux/2]*etax[0]^2)/2 -
   S2*betay[3]*etax[3] + (S2*betax[3]*betay[3]*Cot[mux/2]*etax[3])/(2*f2) -
   (S2*betay[3]^2*Cot[muy]*etax[3])/(2*f2) +
   (S1*betax[0]^(1/2)*betax[3]^(1/2)*betay[0]*Csc[mux/2]*etax[3])/(2*f2) -
   (S2*betay[0]*betay[3]*Csc[muy]*etax[3])/(2*f1) +
   S1*S2*betay[0]*betay[3]*Csc[muy]*etax[0]*etax[3] -
   (S2^2*betax[3]*betay[3]*Cot[mux/2]*etax[3]^2)/2 +
   (S2^2*betay[3]^2*Cot[muy]*etax[3]^2)/2 -
   (S1*S2*betax[0]^(1/2)*betax[3]^(1/2)*betay[0]*Csc[mux/2]*etax[3]^2)/2 -
   (L*etaxp[3]^2*gammay[1])/4 - (L*etaxp[6]^2*gammay[4])/4

wyd1 = -(L*S1*betay[0])/4 - (L*S2*betay[3])/4 -
   (L*S1*alphax[6]*betay[0]*Cot[mux/2])/4 -
   (S1*betax[0]*betay[0]*Cot[mux/2])/2 -
   (L*S2*alphax[3]*betay[3]*Cot[mux/2])/4 -
   (S2*betax[3]*betay[3]*Cot[mux/2])/2 -
   (L*S1*alphax[3]*betax[0]^(1/2)*betay[0]*Csc[mux/2])/(4*betax[3]^(1/2)) -
   (S1*betax[0]^(1/2)*betax[3]^(1/2)*betay[0]*Csc[mux/2])/2 -
   (L*S2*alphax[6]*betax[3]^(1/2)*betay[3]*Csc[mux/2])/(4*betax[0]^(1/2)) -
   (S2*betax[0]^(1/2)*betax[3]^(1/2)*betay[3]*Csc[mux/2])/2 +
   (alphay[6]*etax[0])/(2*f1) - (Cot[muy]*etax[0])/(2*f1) -
   (betay[3]*Csc[muy]*etax[0])/(2*f2*betay[0]) - S1*alphay[6]*etax[0]^2 +
   S1*Cot[muy]*etax[0]^2 + (alphay[3]*etax[3])/(2*f2) -
   (Cot[muy]*etax[3])/(2*f2) - (betay[0]*Csc[muy]*etax[3])/(2*f1*betay[3]) +
   (S1*betay[0]*Csc[muy]*etax[0]*etax[3])/betay[3] +
   (S2*betay[3]*Csc[muy]*etax[0]*etax[3])/betay[0] - S2*alphay[3]*etax[3]^2 +
   S2*Cot[muy]*etax[3]^2 - (L*alphay[3]*etaxp[3])/(4*f2) +
   (L*Cot[muy]*etaxp[3])/(4*f2) +
   (L*betay[0]*Csc[muy]*etaxp[3])/(4*f1*betay[3]) -
   (L*S1*betay[0]*Csc[muy]*etax[0]*etaxp[3])/(2*betay[3]) +
   (L*S2*alphay[3]*etax[3]*etaxp[3])/2 + (S2*betay[3]*etax[3]*etaxp[3])/2 +
   (S2*alphax[3]*betay[3]*Cot[mux/2]*etax[3]*etaxp[3])/2 -
   (L*S2*Cot[muy]*etax[3]*etaxp[3])/2 +
   (S1*alphax[3]*betax[0]^(1/2)*betay[0]*Csc[mux/2]*etax[3]*etaxp[3])/
    (2*betax[3]^(1/2)) - (3*L*S2*betay[3]*etaxp[3]^2)/8 -
   (3*L*S2*alphax[3]*betay[3]*Cot[mux/2]*etaxp[3]^2)/8 -
   (S2*betax[3]*betay[3]*Cot[mux/2]*etaxp[3]^2)/4 -
   (3*L*S1*alphax[3]*betax[0]^(1/2)*betay[0]*Csc[mux/2]*etaxp[3]^2)/
    (8*betax[3]^(1/2)) - (S1*betax[0]^(1/2)*betax[3]^(1/2)*betay[0]*
      Csc[mux/2]*etaxp[3]^2)/4 - (L*alphay[6]*etaxp[6])/(4*f1) +
   (L*Cot[muy]*etaxp[6])/(4*f1) +
   (L*betay[3]*Csc[muy]*etaxp[6])/(4*f2*betay[0]) +
   (L*S1*alphay[6]*etax[0]*etaxp[6])/2 + (S1*betay[0]*etax[0]*etaxp[6])/2 +
   (S1*alphax[6]*betay[0]*Cot[mux/2]*etax[0]*etaxp[6])/2 -
   (L*S1*Cot[muy]*etax[0]*etaxp[6])/2 +
   (S2*alphax[6]*betax[3]^(1/2)*betay[3]*Csc[mux/2]*etax[0]*etaxp[6])/
    (2*betax[0]^(1/2)) - (L*S2*betay[3]*Csc[muy]*etax[3]*etaxp[6])/
    (2*betay[0]) - (3*L*S1*betay[0]*etaxp[6]^2)/8 -
   (3*L*S1*alphax[6]*betay[0]*Cot[mux/2]*etaxp[6]^2)/8 -
   (S1*betax[0]*betay[0]*Cot[mux/2]*etaxp[6]^2)/4 -
   (3*L*S2*alphax[6]*betax[3]^(1/2)*betay[3]*Csc[mux/2]*etaxp[6]^2)/
    (8*betax[0]^(1/2)) - (S2*betax[0]^(1/2)*betax[3]^(1/2)*betay[3]*
      Csc[mux/2]*etaxp[6]^2)/4 -
   (L*alphax[3]*betax[0]^(1/2)*Csc[mux/2]*etax[0]*gammay[1])/
    (8*f1*betax[3]^(1/2)) - (betax[0]^(1/2)*betax[3]^(1/2)*Csc[mux/2]*
      etax[0]*gammay[1])/(4*f1) +
   (L*S1*alphax[3]*betax[0]^(1/2)*Csc[mux/2]*etax[0]^2*gammay[1])/
```

```
 (8*betax[3]^(1/2)) + (S1*betax[0]^(1/2)*betax[3]^(1/2)*Csc[mux/2]*
   etax[0]^2*gammay[1])/4 - (L*etax[3]*gammay[1])/(8*f2) -
(L*alphax[3]*Cot[mux/2]*etax[3]*gammay[1])/(8*f2) -
(betax[3]*Cot[mux/2]*etax[3]*gammay[1])/(4*f2) +
(betay[3]*Cot[muy]*etax[3]*gammay[1])/(4*f2) +
(betay[0]*Csc[muy]*etax[3]*gammay[1])/(4*f1) -
(S1*betay[0]*Csc[muy]*etax[0]*etax[3]*gammay[1])/2 +
(L*S2*etax[3]^2*gammay[1])/8 +
(L*S2*alphax[3]*Cot[mux/2]*etax[3]^2*gammay[1])/8 +
(S2*betax[3]*Cot[mux/2]*etax[3]^2*gammay[1])/4 -
(S2*betay[3]*Cot[muy]*etax[3]^2*gammay[1])/2 + (L*etaxp[3]*gammay[1])/4 -
(L*betay[3]*Cot[muy]*etaxp[3]*gammay[1])/(8*f2) -
(L*betay[0]*Csc[muy]*etaxp[3]*gammay[1])/(8*f1) +
(L*S1*betay[0]*Csc[muy]*etax[0]*etaxp[3]*gammay[1])/4 +
(L*S2*betay[3]*Cot[muy]*etax[3]*etaxp[3]*gammay[1])/4 -
(L*etax[0]*gammay[4])/(8*f1) -
(L*alphax[6]*Cot[mux/2]*etax[0]*gammay[4])/(8*f1) -
(betax[0]*Cot[mux/2]*etax[0]*gammay[4])/(4*f1) +
(betay[0]*Cot[muy]*etax[0]*gammay[4])/(4*f1) +
(betay[3]*Csc[muy]*etax[0]*gammay[4])/(4*f2) +
(L*S1*etax[0]^2*gammay[4])/8 +
(L*S1*alphax[6]*Cot[mux/2]*etax[0]^2*gammay[4])/8 +
(S1*betax[0]*Cot[mux/2]*etax[0]^2*gammay[4])/4 -
(S1*betay[0]*Cot[muy]*etax[0]^2*gammay[4])/2 -
(L*alphax[6]*betax[3]^(1/2)*Csc[mux/2]*etax[3]*gammay[4])/
 (8*f2*betax[0]^(1/2)) - (betax[0]^(1/2)*betax[3]^(1/2)*Csc[mux/2]*
   etax[3]*gammay[4])/(4*f2) -
(S2*betay[3]*Csc[muy]*etax[0]*etax[3]*gammay[4])/2 +
(L*S2*alphax[6]*betax[3]^(1/2)*Csc[mux/2]*etax[3]^2*gammay[4])/
 (8*betax[0]^(1/2)) + (S2*betax[0]^(1/2)*betax[3]^(1/2)*Csc[mux/2]*
   etax[3]^2*gammay[4])/4 + (L*etaxp[6]*gammay[4])/4 -
(L*betay[0]*Cot[muy]*etaxp[6]*gammay[4])/(8*f1) -
(L*betay[3]*Csc[muy]*etaxp[6]*gammay[4])/(8*f2) +
(L*S1*betay[0]*Cot[muy]*etax[0]*etaxp[6]*gammay[4])/4 +
(L*S2*betay[3]*Csc[muy]*etax[3]*etaxp[6]*gammay[4])/4
```

# Appendix B

The following is just an example calculated using our result and the MARYLIE output for it. The parameters are

$R = 20$, $L = 1$, $\mu_x = \frac{2\pi}{5}$, $\mu_y = \frac{\pi}{3}$, $f_1 = -0.958158888525893$, $f_2 = 0.879517775753043$,
$\lambda_1 = -5.002280837708291$, $\lambda_2 = 3.086685263039085$,
$O_1 = -793.2975968091157$, $O_2 = 5843.196053728994$, $O_3 = 4185.979331401783$,
$O_4 = -303.7894419701254$, $O_5 = -8919.69223689852$

```
#beam
  5.657387873499175
  1.065780392113396
  1.00000000000000
  1.00000000000000
#menu
 bend/3    nbnd
  0.954929658551372      0.000000000000000E+00  0.000000000000000E+00
  0.282869393674959      0.000000000000000E+00  0.000000000000000E+00
 bend/2    nbnd
   1.43239448782706      0.000000000000000E+00  0.000000000000000E+00
  0.282869393674959      0.000000000000000E+00  0.000000000000000E+00
 f1/2      thlm
  -2.95221802002116      0.000000000000000E+00  0.000000000000000E+00
  0.000000000000000E+00  0.000000000000000E+00  0.000000000000000E+00
 f2        thlm
   6.43237468242790      0.000000000000000E+00  0.000000000000000E+00
  0.000000000000000E+00  0.000000000000000E+00  0.000000000000000E+00
 s1        thlm
  0.000000000000000E+00  0.000000000000000E+00  -28.2998429510882
  0.000000000000000E+00  0.000000000000000E+00  0.000000000000000E+00
 s2        thlm
  0.000000000000000E+00  0.000000000000000E+00   17.4625757764259
  0.000000000000000E+00  0.000000000000000E+00  0.000000000000000E+00
 oct1      thlm
  0.000000000000000E+00  0.000000000000000E+00  0.000000000000000E+00
  0.000000000000000E+00  -4487.99220426393      0.000000000000000E+00
 oct2      thlm
  0.000000000000000E+00  0.000000000000000E+00  0.000000000000000E+00
  0.000000000000000E+00   33057.2264968447      0.000000000000000E+00
 oct3      thlm
  0.000000000000000E+00  0.000000000000000E+00  0.000000000000000E+00
  0.000000000000000E+00   23681.7087081906      0.000000000000000E+00
 oct4      thlm
  0.000000000000000E+00  0.000000000000000E+00  0.000000000000000E+00
  0.000000000000000E+00  -1718.65470509887      0.000000000000000E+00
 oct5      thlm
  0.000000000000000E+00  0.000000000000000E+00  0.000000000000000E+00
  0.000000000000000E+00  -50462.1586963745      0.000000000000000E+00
 fileout   pmif
   1.00000000000000       12.0000000000000       3.00000000000000
 tunes     tasm
   1.00000000000000      0.000000000000000E+00   1.00000000000000
  0.000000000000000E+00   3.00000000000000      0.000000000000000E+00
 mapout    ptm
   3.00000000000000       3.00000000000000       3.00000000000000
   3.00000000000000       1.00000000000000
 clear     iden
 fin       end
#lines
 bendpkg1
     1*oct1        1*bend/3      1*oct2        1*bend/3      1*oct3      &
     1*bend/3
 bendpkg2
     1*oct4        1*bend/2      1*oct5        1*bend/2
 cell
```

```
     1*f1/2         1*s1           1*bendpkg1     1*f2           1*s2           &
     1*bendpkg2     1*f1/2
 achromat
    30*cell
#lumps
#loops
#labor
    1*fileout
    1*cell
    1*tunes
    1*mapout
    1*clear
    1*achromat
    1*mapout
    1*fin

twiss analysis of static map

tunes and chromaticities for delta defined in terms of P sub tau:

horizontal tune =  0.1999999999999998
first order horizontal chromaticity =  2.3787898555112552E-15
second order horizontal chromaticity = -2.1047638660052968E-11
horizontal tune when delta =  0.0000000000000000E+00
 0.1999999999999998

vertical tune =  0.1666666666666663
first order vertical chromaticity =  1.7299787876779857E-15
second order vertical chromaticity =  2.5815360806559701E-11
vertical tune when delta =  0.0000000000000000E+00
 0.1666666666666663

tune separation when delta=  0.0000000000000000E+00
 3.3333333333333510E-02

normalized anharmonicities
 hhn=  5.3829121008946985E-12
 vvn= -6.8394647870191462E-12
 hvn= -4.0168285257096573E-12

matrix for map is :

 3.09017E-01  8.60629E-01  0.00000E+00  0.00000E+00  0.00000E+00 -8.17232E-02
-1.05099E+00  3.09017E-01  0.00000E+00  0.00000E+00  0.00000E+00 -1.24301E-01
 0.00000E+00  0.00000E+00  5.00000E-01  3.13699E+00  0.00000E+00  0.00000E+00
 0.00000E+00  0.00000E+00 -2.39083E-01  5.00000E-01  0.00000E+00  0.00000E+00
 1.24301E-01  8.17232E-02  0.00000E+00  0.00000E+00  1.00000E+00  6.08675E-01
 0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  1.00000E+00

nonzero elements in generating polynomial are :

 f( 28)=f( 30 00 00 )=-0.36087733494838D+01
 f( 29)=f( 21 00 00 )= 0.67988077172876D+01
 f( 33)=f( 20 00 01 )=-0.82811556873374D+00
 f( 34)=f( 12 00 00 )=-0.35967814322323D+01
 f( 38)=f( 11 00 01 )= 0.73641977083402D+00
 f( 39)=f( 10 20 00 )= 0.67795642688862D+00
 f( 40)=f( 10 11 00 )= 0.33040527594929D+00
 f( 43)=f( 10 02 00 )=-0.10654698053081D+02
 f( 48)=f( 10 00 02 )=-0.66708622334423D-01
 f( 49)=f( 03 00 00 )=-0.22364720013566D-01
 f( 53)=f( 02 00 01 )=-0.79579186885371D+00
 f( 54)=f( 01 20 00 )= 0.37423728186523D+00
 f( 55)=f( 01 11 00 )=-0.10543525395979D+02
 f( 58)=f( 01 02 00 )= 0.39398512852412D+02
 f( 63)=f( 01 00 02 )=-0.39371337857938D-01
```

```
 f( 67)=f( 00 20 01 )=-0.73767698643843D-01
 f( 70)=f( 00 11 01 )=-0.17409983445913D+01
 f( 76)=f( 00 02 01 )= 0.75982680476656D+00
 f( 83)=f( 00 00 03 )=-0.35257444190990D+00
 f( 84)=f( 40 00 00 )= 0.40302615555435D+04
 f( 85)=f( 31 00 00 )=-0.25249116719569D+04
 f( 89)=f( 30 00 01 )= 0.18854858999716D+03
 f( 90)=f( 22 00 00 )=-0.43656277598876D+04
 f( 94)=f( 21 00 01 )=-0.82298416963088D+03
 f( 95)=f( 20 20 00 )=-0.89540738027763D+04
 f( 96)=f( 20 11 00 )=-0.66193602059579D+04
 f( 99)=f( 20 02 00 )= 0.41805766977363D+05
 f(104)=f( 20 00 02 )=-0.26146876979076D+02
 f(105)=f( 13 00 00 )= 0.51635942464328D+04
 f(109)=f( 12 00 01 )= 0.95045249484747D+03
 f(110)=f( 11 20 00 )= 0.37874005739449D+04
 f(111)=f( 11 11 00 )= 0.33531939439938D+05
 f(114)=f( 11 02 00 )=-0.94122503839720D+05
 f(119)=f( 11 00 02 )= 0.54354932742993D+02
 f(123)=f( 10 20 01 )=-0.14946543985032D+03
 f(126)=f( 10 11 01 )= 0.22326221268730D+04
 f(132)=f( 10 02 01 )=-0.51740933019890D+04
 f(139)=f( 10 00 03 )= 0.84818133692899D+00
 f(140)=f( 04 00 00 )=-0.15365762959048D+04
 f(144)=f( 03 00 01 )=-0.33743811716499D+03
 f(145)=f( 02 20 00 )= 0.94230160900497D+03
 f(146)=f( 02 11 00 )=-0.21031891203500D+05
 f(149)=f( 02 02 00 )= 0.49179167805833D+05
 f(154)=f( 02 00 02 )=-0.27567976480448D+02
 f(158)=f( 01 20 01 )= 0.23253251952814D+03
 f(161)=f( 01 11 01 )=-0.23548138902015D+04
 f(167)=f( 01 02 01 )= 0.51522990812897D+04
 f(174)=f( 01 00 03 )=-0.89178158407600D+00
 f(175)=f( 00 40 00 )= 0.53707234500912D+03
 f(176)=f( 00 31 00 )= 0.37719560119157D+03
 f(179)=f( 00 22 00 )=-0.12032053633894D+05
 f(184)=f( 00 20 02 )= 0.66187335188831D+01
 f(185)=f( 00 13 00 )= 0.38124752583720D+05
 f(190)=f( 00 11 02 )=-0.62005819310487D+02
 f(195)=f( 00 04 00 )=-0.40141159909343D+05
 f(200)=f( 00 02 02 )= 0.12429656509976D+03
 f(209)=f( 00 00 04 )=-0.43573554224229D+00

nonzero elements in second order matrix are :

 t1(  7)=t1( 20 00 00 )=-0.29113888753308D+01
 t1(  8)=t1( 11 00 00 )=-0.94795665694566D+01
 t1( 12)=t1( 10 00 01 )=-0.11978340275467D+01
 t1( 13)=t1( 02 00 00 )=-0.31162270003652D+01
 t1( 17)=t1( 01 00 01 )=-0.14195765647213D+00
 t1( 18)=t1( 00 20 00 )=-0.36059925408102D+01
 t1( 19)=t1( 00 11 00 )= 0.29737722247719D+01
 t1( 22)=t1( 00 02 00 )= 0.30050697971915D+01
 t1( 27)=t1( 00 00 02 )=-0.69577775503190D-01
 t2(  7)=t2( 20 00 00 )=-0.94228514068444D+01
 t2(  8)=t2( 11 00 00 )=-0.14422927716320D+02
 t2( 12)=t2( 10 00 01 )=-0.10331164265804D+01
 t2( 13)=t2( 02 00 00 )=-0.47460508077301D+01
 t2( 17)=t2( 01 00 01 )=-0.11966437306390D+01
 t2( 18)=t2( 00 20 00 )=-0.47903751649514D+00
 t2( 19)=t2( 00 11 00 )= 0.45088941836360D+01
 t2( 22)=t2( 00 02 00 )= 0.45261199149792D+01
 t2( 27)=t2( 00 00 02 )=-0.12644012852834D-01
 t3(  9)=t3( 10 10 00 )=-0.26965493556271D+02
 t3( 10)=t3( 10 01 00 )= 0.96182211018688D+01
 t3( 14)=t3( 01 10 00 )= 0.29238079577202D+01
```

```
u3( 51)=u3( 02 01 00 )=-0.16923083123868D+05
u3( 57)=u3( 01 10 01 )= 0.30088779842340D+03
u3( 60)=u3( 01 01 01 )= 0.22736506876264D+04
u3( 64)=u3( 00 30 00 )=-0.69449221304839D+04
u3( 65)=u3( 00 21 00 )=-0.89660657696518D+04
u3( 68)=u3( 00 12 00 )= 0.18299426464375D+05
u3( 73)=u3( 00 10 02 )=-0.15294468526749D+02
u3( 74)=u3( 00 03 00 )= 0.37132679641762D+05
u3( 79)=u3( 00 01 02 )=-0.55017952817406D+02
u4( 30)=u4( 20 10 00 )= 0.72392062383916D+04
u4( 31)=u4( 20 01 00 )=-0.23897933072994D+05
u4( 35)=u4( 11 10 00 )=-0.44083237411048D+04
u4( 36)=u4( 11 01 00 )=-0.61423224143364D+05
u4( 42)=u4( 10 10 01 )= 0.68769172176665D+03
u4( 45)=u4( 10 01 01 )= 0.34730598868319D+04
u4( 50)=u4( 02 10 00 )=-0.60224778867911D+04
u4( 51)=u4( 02 01 00 )=-0.33722263967008D+05
u4( 57)=u4( 01 10 01 )= 0.79188119969148D+03
u4( 60)=u4( 01 01 01 )= 0.36081068855791D+04
u4( 64)=u4( 00 30 00 )=-0.10096510584836D+04
u4( 65)=u4( 00 21 00 )= 0.61777995255206D+04
u4( 68)=u4( 00 12 00 )= 0.39034780561603D+05
u4( 73)=u4( 00 10 02 )=-0.21339464281211D+02
u4( 74)=u4( 00 03 00 )= 0.57779121822527D+05
u4( 79)=u4( 00 01 02 )=-0.96454014456750D+02
u5( 28)=u5( 30 00 00 )=-0.79722407116222D+03
u5( 29)=u5( 21 00 00 )=-0.84294332523951D+03
u5( 33)=u5( 20 00 01 )= 0.99595762907565D+01
u5( 34)=u5( 12 00 00 )=-0.23531655907834D+03
u5( 38)=u5( 11 00 01 )=-0.52627361220503D+00
u5( 39)=u5( 10 20 00 )= 0.85287725859185D+03
u5( 40)=u5( 10 11 00 )= 0.79105547159717D+03
u5( 43)=u5( 10 02 00 )=-0.33654785162717D+03
u5( 48)=u5( 10 00 02 )= 0.48984832937943D+00
u5( 49)=u5( 03 00 00 )=-0.15063676852571D+01
u5( 53)=u5( 02 00 01 )=-0.39770430660179D+01
u5( 54)=u5( 01 20 00 )= 0.31867104805823D+03
u5( 55)=u5( 01 11 00 )= 0.98458028533960D+02
u5( 58)=u5( 01 02 00 )=-0.58598957778585D+03
u5( 63)=u5( 01 00 02 )= 0.47294680213133D+00
u5( 67)=u5( 00 20 01 )=-0.25143481691889D+01
u5( 70)=u5( 00 11 01 )= 0.86878333150077D+01
u5( 76)=u5( 00 02 01 )= 0.47811396750042D+02
u5( 83)=u5( 00 00 03 )= 0.16841905694309D+01
```

matrix for map is :

| | | | | | |
|---|---|---|---|---|---|
| 1.00000E+00 | -3.06491E-14 | 0.00000E+00 | 0.00000E+00 | 0.00000E+00 | 9.21572E-17 |
| 3.74561E-14 | 1.00000E+00 | 0.00000E+00 | 0.00000E+00 | 0.00000E+00 | 4.47294E-15 |
| 0.00000E+00 | 0.00000E+00 | 1.00000E+00 | -2.36887E-13 | 0.00000E+00 | 0.00000E+00 |
| 0.00000E+00 | 0.00000E+00 | 1.81938E-14 | 1.00000E+00 | 0.00000E+00 | 0.00000E+00 |
| -4.39579E-15 | -3.30682E-17 | 0.00000E+00 | 0.00000E+00 | 1.00000E+00 | 1.78192E+01 |
| 0.00000E+00 | 0.00000E+00 | 0.00000E+00 | 0.00000E+00 | 0.00000E+00 | 1.00000E+00 |

nonzero elements in generating polynomial are :

```
f( 28)=f( 30 00 00 )= 0.21984751419411D-12
f( 29)=f( 21 00 00 )= 0.62698211313604D-13
f( 33)=f( 20 00 01 )=-0.23708377019442D-12
f( 34)=f( 12 00 00 )=-0.14155644533019D-12
f( 38)=f( 11 00 01 )=-0.37725385885148D-14
f( 39)=f( 10 20 00 )= 0.70415558764384D-13
f( 40)=f( 10 11 00 )=-0.11926168931577D-11
f( 43)=f( 10 02 00 )=-0.11034106168155D-13
f( 48)=f( 10 00 02 )=-0.64544479417829D-13
f( 49)=f( 03 00 00 )= 0.47092321361908D-15
```

```
t3( 15)=t3( 01 01 00 )= 0.63236140752887D+01
t3( 21)=t3( 00 10 01 )=-0.13333157491110D+01
t3( 24)=t3( 00 01 01 )= 0.47016618203245D+01
t4(  9)=t4( 10 10 00 )=-0.28575288861685D+01
t4( 10)=t4( 10 01 00 )= 0.44383282607149D+01
t4( 14)=t4( 01 10 00 )= 0.14100927276588D+01
t4( 15)=t4( 01 01 00 )= 0.28993397722101D+01
t4( 21)=t4( 00 10 01 )=-0.66326994337330D-01
t4( 24)=t4( 00 01 01 )=-0.13309923108453D+01
t5(  7)=t5( 20 00 00 )= 0.11972552292409D+01
t5(  8)=t5( 11 00 00 )= 0.51934652354946D+00
t5( 12)=t5( 10 00 01 )= 0.89602490200475D-01
t5( 13)=t5( 02 00 00 )= 0.49351157948672D+00
t5( 17)=t5( 01 00 01 )= 0.58910224896218D-01
t5( 18)=t5( 00 20 00 )=-0.23311177270443D+00
t5( 19)=t5( 00 11 00 )=-0.45742925734938D+00
t5( 22)=t5( 00 02 00 )= 0.32667123308212D+01
t5( 27)=t5( 00 00 02 )= 0.10473777878925D+01

nonzero elements in third order matrix are :

u1( 28)=u1( 30 00 00 )=-0.13217108499851D+05
u1( 29)=u1( 21 00 00 )=-0.92984289345182D+04
u1( 33)=u1( 20 00 01 )=-0.26709520650885D+03
u1( 34)=u1( 12 00 00 )= 0.27373762011917D+04
u1( 38)=u1( 11 00 01 )=-0.88564615349971D+03
u1( 39)=u1( 10 20 00 )= 0.14322717418497D+05
u1( 40)=u1( 10 11 00 )=-0.18450546550516D+04
u1( 43)=u1( 10 02 00 )=-0.44517430713237D+05
u1( 48)=u1( 10 00 02 )= 0.24674773774807D+02
u1( 49)=u1( 03 00 00 )= 0.25162230724583D+04
u1( 53)=u1( 02 00 01 )=-0.51959613528269D+03
u1( 54)=u1( 01 20 00 )= 0.38567376870702D+04
u1( 55)=u1( 01 11 00 )=-0.16196439375810D+05
u1( 58)=u1( 01 02 00 )=-0.50031165531545D+05
u1( 63)=u1( 01 00 02 )= 0.26840621207159D+02
u1( 67)=u1( 00 20 01 )= 0.60251062927382D+02
u1( 70)=u1( 00 11 01 )= 0.11677532141795D+04
u1( 76)=u1( 00 02 01 )= 0.27115585706305D+04
u1( 83)=u1( 00 00 03 )=-0.55102974874563D+00
u2( 28)=u2( 30 00 00 )=-0.77635673596330D+04
u2( 29)=u2( 21 00 00 )= 0.67163504976368D+04
u2( 33)=u2( 20 00 01 )=-0.10885023412963D+04
u2( 34)=u2( 12 00 00 )= 0.18794457272262D+05
u2( 38)=u2( 11 00 01 )=-0.25548529560628D+04
u2( 39)=u2( 10 20 00 )= 0.94739830284905D+04
u2( 40)=u2( 10 11 00 )=-0.40043995622099D+05
u2( 43)=u2( 10 02 00 )=-0.12413665656236D+06
u2( 48)=u2( 10 00 02 )= 0.68283881242166D+02
u2( 49)=u2( 03 00 00 )= 0.79774711048435D+04
u2( 53)=u2( 02 00 01 )=-0.13707200754260D+04
u2( 54)=u2( 01 20 00 )=-0.95815774840939D+03
u2( 55)=u2( 01 11 00 )=-0.54401902892894D+05
u2( 58)=u2( 01 02 00 )=-0.13105016317590D+06
u2( 63)=u2( 01 00 02 )= 0.70048974816771D+02
u2( 67)=u2( 00 20 01 )= 0.29741215412090D+03
u2( 70)=u2( 00 11 01 )= 0.31397221109509D+04
u2( 76)=u2( 00 02 01 )= 0.68982901882357D+04
u2( 83)=u2( 00 00 03 )=-0.12429509966526D+01
u3( 30)=u3( 20 10 00 )= 0.59794397025521D+05
u3( 31)=u3( 20 01 00 )= 0.20299478350741D+05
u3( 35)=u3( 11 10 00 )= 0.40128896515394D+05
u3( 36)=u3( 11 01 00 )=-0.13054009954643D+05
u3( 42)=u3( 10 10 01 )=-0.17447374345732D+03
u3( 45)=u3( 10 01 01 )= 0.19405000143050D+04
u3( 50)=u3( 02 10 00 )= 0.40303185679133D+04
```

```
f( 53)=f( 02 00 01 )=-0.24442947665904D-12
f( 54)=f( 01 20 00 )=-0.76119312393632D-13
f( 55)=f( 01 11 00 )= 0.85117053866994D-12
f( 58)=f( 01 02 00 )=-0.15566432691461D-13
f( 63)=f( 01 00 02 )=-0.23960868011930D-15
f( 67)=f( 00 20 01 )=-0.52327447184013D-13
f( 70)=f( 00 11 01 )=-0.22227480449214D-14
f( 76)=f( 00 02 01 )=-0.53258786270050D-12
f( 83)=f( 00 00 03 )=-0.10508945328211D+02
f( 84)=f( 40 00 00 )=-0.32129601448636D-09
f( 85)=f( 31 00 00 )=-0.83158593157642D-09
f( 89)=f( 30 00 01 )=-0.16520264023710D-09
f( 90)=f( 22 00 00 )=-0.50064492396829D-09
f( 94)=f( 21 00 01 )=-0.10202731189904D-09
f( 95)=f( 20 20 00 )= 0.11467560567127D-09
f( 96)=f( 20 11 00 )= 0.46716508352656D-08
f( 99)=f( 20 02 00 )= 0.12634983150736D-08
f(104)=f( 20 00 02 )= 0.10626680934698D-08
f(105)=f( 13 00 00 )= 0.28474929160768D-09
f(109)=f( 12 00 01 )=-0.13985201666485D-09
f(110)=f( 11 20 00 )= 0.12099219539472D-08
f(111)=f( 11 11 00 )=-0.54813453048725D-10
f(114)=f( 11 02 00 )=-0.10677493932566D-07
f(119)=f( 11 00 02 )=-0.68072789969995D-11
f(123)=f( 10 20 01 )= 0.26312196818661D-10
f(126)=f( 10 11 01 )=-0.28442647430246D-09
f(132)=f( 10 02 01 )= 0.39828022140099D-09
f(139)=f( 10 00 03 )= 0.25570716669590D-09
f(140)=f( 04 00 00 )=-0.25447686456595D-09
f(144)=f( 03 00 01 )=-0.46796707722724D-10
f(145)=f( 02 20 00 )= 0.11100258028345D-09
f(146)=f( 02 11 00 )= 0.32520055274940D-09
f(149)=f( 02 02 00 )= 0.12597988069149D-08
f(154)=f( 02 00 02 )= 0.87919613103323D-09
f(158)=f( 01 20 01 )= 0.10513837141607D-09
f(161)=f( 01 11 01 )= 0.14198925800423D-11
f(167)=f( 01 02 01 )=-0.13284442380090D-08
f(174)=f( 01 00 03 )=-0.13059185082036D-11
f(175)=f( 00 40 00 )= 0.16860026331519D-10
f(176)=f( 00 31 00 )=-0.62878728935754D-09
f(179)=f( 00 22 00 )= 0.45414820069460D-09
f(184)=f( 00 20 02 )=-0.33434725367681D-09
f(185)=f( 00 13 00 )= 0.16324993686761D-08
f(190)=f( 00 11 02 )=-0.35867768978799D-10
f(195)=f( 00 04 00 )= 0.23536822337791D-08
f(200)=f( 00 02 02 )=-0.43795822451026D-08
f(209)=f( 00 00 04 )=-0.12738495535630D+02

nonzero elements in second order matrix are :

t1(  7)=t1( 20 00 00 )=-0.62698211313593D-13
t1(  8)=t1( 11 00 00 )= 0.28311289066038D-12
t1( 12)=t1( 10 00 01 )= 0.37725385885344D-14
t1( 13)=t1( 02 00 00 )=-0.14127696408659D-14
t1( 17)=t1( 01 00 01 )= 0.48885895331807D-12
t1( 18)=t1( 00 20 00 )= 0.76119312393617D-13
t1( 19)=t1( 00 11 00 )=-0.85117053866998D-12
t1( 22)=t1( 00 02 00 )= 0.15566432691662D-13
t1( 27)=t1( 00 00 02 )= 0.23960868012148D-15
t2(  7)=t2( 20 00 00 )= 0.65954254258234D-12
t2(  8)=t2( 11 00 00 )= 0.12539642262716D-12
t2( 12)=t2( 10 00 01 )=-0.47416754038884D-12
t2( 13)=t2( 02 00 00 )=-0.14155644533019D-12
t2( 17)=t2( 01 00 01 )=-0.37725385885016D-14
t2( 18)=t2( 00 20 00 )= 0.70415558764362D-13
t2( 19)=t2( 00 11 00 )=-0.11926168931578D-11
```

```
 t2( 22)=t2( 00 02 00 )=-0.11034106167873D-13
 t2( 27)=t2( 00 00 02 )=-0.64544479417829D-13
 t3(  9)=t3( 10 10 00 )= 0.11926168931577D-11
 t3( 10)=t3( 10 01 00 )= 0.22068212336029D-13
 t3( 14)=t3( 01 10 00 )=-0.85117053866998D-12
 t3( 15)=t3( 01 01 00 )= 0.31132865383122D-13
 t3( 21)=t3( 00 10 01 )= 0.22227480449371D-14
 t3( 24)=t3( 00 01 01 )= 0.10651757254010D-11
 t4(  9)=t4( 10 10 00 )= 0.14083111752874D-12
 t4( 10)=t4( 10 01 00 )=-0.11926168931577D-11
 t4( 14)=t4( 01 10 00 )=-0.15223862478725D-12
 t4( 15)=t4( 01 01 00 )= 0.85117053867002D-12
 t4( 21)=t4( 00 10 01 )=-0.10465489436803D-12
 t4( 24)=t4( 00 01 01 )=-0.22227480448929D-14
 t5(  7)=t5( 20 00 00 )= 0.23778377019442D-12
 t5(  8)=t5( 11 00 00 )= 0.37725385885186D-14
 t5( 12)=t5( 10 00 01 )= 0.12908895883566D-12
 t5( 13)=t5( 02 00 00 )= 0.24442947665904D-12
 t5( 17)=t5( 01 00 01 )= 0.47921736023682D-15
 t5( 18)=t5( 00 20 00 )= 0.52327447184013D-13
 t5( 19)=t5( 00 11 00 )= 0.22227480449160D-14
 t5( 22)=t5( 00 02 00 )= 0.53258786270050D-12
 t5( 27)=t5( 00 00 02 )= 0.31526835984633D+02

nonzero elements in third order matrix are :

 u1( 28)=u1( 30 00 00 )= 0.83158593157646D-09
 u1( 29)=u1( 21 00 00 )= 0.10012898479364D-08
 u1( 33)=u1( 20 00 01 )= 0.10202731189906D-09
 u1( 34)=u1( 12 00 00 )=-0.85424787482298D-09
 u1( 38)=u1( 11 00 01 )= 0.27970403332969D-09
 u1( 39)=u1( 10 20 00 )=-0.12099219539472D-08
 u1( 40)=u1( 10 11 00 )= 0.54813453049662D-10
 u1( 43)=u1( 10 02 00 )= 0.10677493932566D-07
 u1( 48)=u1( 10 00 02 )= 0.68072789969348D-11
 u1( 49)=u1( 03 00 00 )= 0.10179074582638D-08
 u1( 53)=u1( 02 00 01 )= 0.14039012316818D-09
 u1( 54)=u1( 01 20 00 )=-0.22200516056688D-09
 u1( 55)=u1( 01 11 00 )=-0.65040110549879D-09
 u1( 58)=u1( 01 02 00 )=-0.25195976138299D-08
 u1( 63)=u1( 01 00 02 )=-0.17583922620665D-08
 u1( 67)=u1( 00 20 01 )=-0.10513837141607D-09
 u1( 70)=u1( 00 11 01 )=-0.14198925799473D-11
 u1( 76)=u1( 00 02 01 )= 0.13284442380090D-08
 u1( 83)=u1( 00 00 03 )= 0.13059185081958D-11
 u2( 28)=u2( 30 00 00 )=-0.12851840579455D-08
 u2( 29)=u2( 21 00 00 )=-0.24947577947292D-08
 u2( 33)=u2( 20 00 01 )=-0.49560792071133D-09
 u2( 34)=u2( 12 00 00 )=-0.10012898479364D-08
 u2( 38)=u2( 11 00 01 )=-0.20405462379807D-09
 u2( 39)=u2( 10 20 00 )= 0.22935121134275D-09
 u2( 40)=u2( 10 11 00 )= 0.93433016705311D-08
 u2( 43)=u2( 10 02 00 )= 0.25269966301446D-08
 u2( 48)=u2( 10 00 02 )= 0.21253361869396D-08
 u2( 49)=u2( 03 00 00 )= 0.28474929160771D-09
 u2( 53)=u2( 02 00 01 )=-0.13985201666484D-09
 u2( 54)=u2( 01 20 00 )= 0.12099219539472D-08
 u2( 55)=u2( 01 11 00 )=-0.54813453049973D-10
 u2( 58)=u2( 01 02 00 )=-0.10677493932566D-07
 u2( 63)=u2( 01 00 02 )=-0.68072789970660D-11
 u2( 67)=u2( 00 20 01 )= 0.26312196818662D-10
 u2( 70)=u2( 00 11 01 )=-0.28442647430246D-09
 u2( 76)=u2( 00 02 01 )= 0.39828022140101D-09
 u2( 83)=u2( 00 00 03 )= 0.25570716669590D-09
 u3( 30)=u3( 20 10 00 )=-0.46716508352656D-08
 u3( 31)=u3( 20 01 00 )=-0.25269966301453D-08
```

```
u3( 35)=u3( 11 10 00 )= 0.54813453049375D-10
u3( 36)=u3( 11 01 00 )= 0.21354987865131D-07
u3( 42)=u3( 10 10 01 )= 0.28442647430245D-09
u3( 45)=u3( 10 01 01 )=-0.79656044280185D-09
u3( 50)=u3( 02 10 00 )=-0.32520055274945D-09
u3( 51)=u3( 02 01 00 )=-0.25195976138303D-08
u3( 57)=u3( 01 10 01 )=-0.14198925800054D-11
u3( 60)=u3( 01 01 01 )= 0.26568884760179D-08
u3( 64)=u3( 00 30 00 )= 0.62878728935753D-09
u3( 65)=u3( 00 21 00 )=-0.90829640138982D-09
u3( 68)=u3( 00 12 00 )=-0.48974981060285D-08
u3( 73)=u3( 00 10 02 )= 0.35867768978958D-10
u3( 74)=u3( 00 03 00 )=-0.94147289351151D-08
u3( 79)=u3( 00 01 02 )= 0.87591644902052D-08
u4( 30)=u4( 20 10 00 )= 0.22935121134271D-09
u4( 31)=u4( 20 01 00 )= 0.46716508352655D-08
u4( 35)=u4( 11 10 00 )= 0.24198439078943D-08
u4( 36)=u4( 11 01 00 )=-0.54813453049561D-10
u4( 42)=u4( 10 10 01 )= 0.52624393637336D-10
u4( 45)=u4( 10 01 01 )=-0.28442647430247D-09
u4( 50)=u4( 02 10 00 )= 0.22200516056684D-09
u4( 51)=u4( 02 01 00 )= 0.32520055274935D-09
u4( 57)=u4( 01 10 01 )= 0.21027674283214D-09
u4( 60)=u4( 01 01 01 )= 0.14198925800037D-11
u4( 64)=u4( 00 30 00 )= 0.67440105326041D-10
u4( 65)=u4( 00 21 00 )=-0.18863618680727D-08
u4( 68)=u4( 00 12 00 )= 0.90829640139018D-09
u4( 73)=u4( 00 10 02 )=-0.66869450735362D-09
u4( 74)=u4( 00 03 00 )= 0.16324993686759D-08
u4( 79)=u4( 00 01 02 )=-0.35867768978641D-10
u5( 28)=u5( 30 00 00 )= 0.16520264023711D-09
u5( 29)=u5( 21 00 00 )= 0.10202731189904D-09
u5( 33)=u5( 20 00 01 )=-0.21253361869396D-08
u5( 34)=u5( 12 00 00 )= 0.13985201666484D-09
u5( 38)=u5( 11 00 01 )= 0.13614557993999D-10
u5( 39)=u5( 10 20 00 )=-0.26312196818660D-10
u5( 40)=u5( 10 11 00 )= 0.28442647430246D-09
u5( 43)=u5( 10 02 00 )=-0.39828022140101D-09
u5( 48)=u5( 10 00 02 )=-0.76712150008771D-09
u5( 49)=u5( 03 00 00 )= 0.46796707722719D-10
u5( 53)=u5( 02 00 01 )=-0.17583922620665D-08
u5( 54)=u5( 01 20 00 )=-0.10513837141607D-09
u5( 55)=u5( 01 11 00 )=-0.14198925799532D-11
u5( 58)=u5( 01 02 00 )= 0.13284442380090D-08
u5( 63)=u5( 01 00 02 )= 0.39177555246187D-11
u5( 67)=u5( 00 20 01 )= 0.66869450735362D-09
u5( 70)=u5( 00 11 01 )= 0.71735537957600D-10
u5( 76)=u5( 00 02 01 )= 0.87591644902052D-08
u5( 83)=u5( 00 00 03 )= 0.50953982142521D+02
```