MANUAL FOR THE PROGRAM *TEAPOT*
NONINTERACTIVE *FORTRAN* VERSION

(THIN ELEMENT ACCELERATOR PROGRAM
FOR OPTICS AND TRACKING)

Lindsay Schachinger, Lawrence Berkeley Laboratory, Berkeley, CA, 94720
and Richard Talman, Laboratory of Nuclear Studies,
Cornell University, Ithaca, NY 14853

The recommended procedure for acquiring the code is to communicate with either of the above authors. The code and this manual can be obtained using "anonymous ftp", from hazel.tn.cornell.edu, 128.84.251.20. Assuming that the ftpot release is to be untarred from a directory $HOME/tpot, proceed as follows:

```
% cd $HOME
% mkdir tpot
% cd tpot
% ftp hazel.tn.cornell.edu
Connected to hazel.
220 hazel FTP server (SunOS 4.1) ready.
Name (hazel.tn.cornell.edu:talman):  TYPE anonymous <cr>
331 Password required for anononymous.
Password:  TYPE e-mail address <CR>
230 Guest login ok, access restrictions apply.
ftp> TYPE cd pub <CR>
250 CWD command successful.
ftp> TYPE ls <CR>
200 PORT command successful.
150 ASCII data connection for /bin/ls (128.84.251.20,1182) (0 bytes).
ftpot.tar.Z
226 ASCII Transfer complete.
13 bytes received in 0.65 seconds (0.02 Kbytes/s)
ftp> TYPE get ftpot.tar.Z <CR>
200 PORT command successful.
150 ASCII data connection for ftpot.tar.Z (128.84.251.20,1183) (3081895
bytes).
226 ASCII Transfer complete.
local:  ftpot.tar.Z remote:  ftpot.tar.Z
3095826 bytes received in 20 seconds (1.5e+02 Kbytes/s)
ftp> TYPE quit <CR>
221 Goodbye.
% uncompress ftpot.tar.Z
```

```
% tar xvf ftpot.tar
```

- This will have generated a directory $HOME/tpot/ftpot as well as self-consistent contents and subdirectories. For more instructions refer to README files in $HOME/tpot/ftpot and $HOME/tpot/ftpot/test. A postscript version of this file, tpotfullFORTRAN.ps, can be also obtained from the same directory using anonymous ftp. Otherwise, there is a TeX file tpotfullFORTRAN.tex or a dvi file can be obtained, but the only way of getting a copy of the manual that includes figures is to get tpotfullFORTRAN.ps and print it on a postscript printer. This includes encapsulated postscript produced using "*TEXSIS*" starting from tpotfullFORTRAN.tex. If a version of TeX other than *TEXSIS* is used for converting file.tex to file.dvi it is necessary to emulate this encapsulation some other way, or to remove the figures.
- Sample lattice description files are in subdirectory $HOME/tpot/ftpot/test.
- It is recommended that a directory structure like $HOME/tpot/ftpot/test's be established. Instructions for doing this, command files for routing output files into these directories are included, and instructions for using them are contained in the README files mentioned above. A standard directory organization is useful for writing, then later reading, an intermediate, "flat", machine description file (`fort.7`) that encapsulates preliminary lattice tuning already performed and can be used for tracking or further tuning.
- Since accelerators are "flaky", and since *TEAPOT* models accelerators faithfully, it follows that *TEAPOT* may occasionally appear "flaky" also. As with real accelerators, it is normal to have to iterate correction procedures. Lattice manipulations, especially tune changes, sometimes cause the lattice to go unstable. Usually, as with real storage rings, taking smaller steps overcomes the problem. Most other difficulties have to do with misinterpretation of this (obscure) manual (outright errors in the manual have

not been unknown). R. Talman will be pleased to attempt to repair any such misinterpretations as well as any bugs that surface. Based on tracking down hundreds of problems on tens of accelerators, the thin element approach is never the ultimate source of disagreement with either expectation or other codes. In rings with only a few dipoles they should be broken up so that no deflection is greater than a few degrees, and "mini-beta" quadrupoles should be broken up, but otherwise compulsive splitting up of elements is rarely justified or necessary.

GENERAL DESCRIPTION AND SURVEY OF CAPABILITIES

TEAPOT is an accelerator modeling code that treats all elements (aside from drifts) as thin elements. This gives it the feature, unique among numerical codes, of being symplectic to all orders (not counting computer precision limitation.) The command language for *TEAPOT* is a dialect of that used by *MAD*. This section gives a brief overview of many of the capabilities. Detailed syntax and parameter definition is contained in later sections. All variables are measured in SI units (except energies are measured in GeV.)

TEAPOT reads a lattice in Standard Input Format and converts all thick elements to thin ones. The user can acheve arbitrarily close agreement with thick element representations by breaking elements into shorter lengths. Because this process is most important for quadrupoles there is special provision for breaking them up—if a quadrupole is of type IR or IRn (where n = 2, 4, 8 or 16), it is split into four or 4n thin quadrupoles. This breaking up is handled internally in a way that is transparent to the user, including the assignment of errors. The sublengths chosen are not quite uniform—rather they follow an algorithm for enhanced tracking accuracy. It is legitimate, and even sensible if computing time is not an issue and comparison with other lattice codes is, to make all quadrupoles have type IR or IRn. A Twiss analysis can be performed and the tunes can be adjusted using a thin lens matrix representation of the machine. Magnetic errors and misalignments, either systematic or random, can be added to elements, after

which the closed orbit can be found and particles can be tracked exactly through the resulting lattice. By means of this tracking numerous accelerator operations can be modeled.

A preliminary Twiss analysis is available which assumes an ideal, uncoupled, lattice and works with 2×2 matrices. This is useful for comparison with other two dimensional codes. However most other operations of *TEAPOT* use a general 4×4 formalism which explicitly evaluates eigenmotions in the presence of arbitrary errors. These analyses use tracked particle trajectories to compute the first and second order transfer matrices for the entire machine or through arbitrary sectors of the lattice delimited by concatenation flags. Subsequent tracking is performed exactly or using these maps depending on the flag settings. A 'BEAMTRAK beam' of up to 1024 particles can be randomly generated and the particles tracked individually with the beam centroid coordinates being recorded at beam position monitor locations. The machine can be decoupled using skew quadrupoles, the tunes can be adjusted, and the chromaticity can be fit in the presence of errors. Nonlinear fitting of the values of user-declared parameters can be performed to cause the large amplitude lattice transfer map agree as nearly as possible with the small amplitude transfer map.

Longitudinal path length and velocity deviations needed for synchrotron oscillations are calculated. RF cavities are supported. It is assumed that output of longitudinal displacements during tracking is desired if and only if there is an RF cavity in the lattice. Since the nominal revolution period is calculated internal to the program, the RF frequency is input as the harmonic number (an integer) multiple of the revolution frequency.

The two main purposes of *TEAPOT* are the analysis of the expected performance of already designed lattices and the simulation of lattice tuning and correction operations. *TEAPOT*'s design capability is restricted to making the best use of the elements present and hence distinguishing among different possible designs. It is not intended to be used for basic linear lattice design and

matching operations such as are performed by *MAD* or *SYNCH*.

TEAPOT is intended as the central computational part of an interactive operations simulation using a unix workstation but this manual is restricted to the setting up and noninteractive use of the program. Graphics and interactive descriptions are contained elsewhere. Most capabilities described here have been ported to the VAX and have been or can easily be ported to the CRAY or presumably to any other mainframe computer. Also the code has compiled and run on most unix workstations; including `SUN`, `HP`, `DEC`, `IBM` and `Silicon Graphics`.

The first portion of this document is a description of the elements, commands and options recognized by *TEAPOT*. Appendices A through E describe the format of output or input files. Appendix F defines the various special type codes recognized by *TEAPOT* and explains how they can be used to control calculations. The material to this point is intended to give succinct instructions on how to set up an input file. The remaining two appendices give an extremely verbose description of many of the formulas used in the program. They are intended to augment the original publication in Particle Accelerators, 1987, which is also SSC-52.

There are versions of the code in two computer languages, Fortran and C. Initially the code had been written in Fortran; the conversion to C was performed in 1988 by Vern Paxson. Subsequent coding was sometimes in C, subsequently ported to Fortran, and sometimes vice versa. The two versions are almost completely interchangeable, both in input and output. At least they were until roughly summer of 1990. Since then numerous changes, having to do with beam steering, tracking of longitudinal displacements, allowance for RF cavities, support for dynamic modeling such as resonant extraction, and ability to model aperture limits and maximum element strengths have been coded in the Fortran version, and not yet ported to the C version. This version of the manual reflects those changes, as well as some features supported by both versions, but not yet described in the official manual, which is supposed to be valid for both languages.

There is a small difference between the "thin element circumference" and the "thick element circumference"; it is due to the use of straight line segments. It is explicitly indicated in the output printout, and can either be ignored, reduced by breaking elements into shorter lengths, or otherwise allowed for.

Effective October 1993, appreciable (but backward compatible) extensions will have been incorporated going beyond the "standard lattice description language" of Snowmass 1984 which, justified or not, we also call *MAD 4.0*. (Miscellaneous minor extensions had crept in previously.) New features include aperture limitations at (almost) all elements, and maximum bend strengths for most active elements. These changes are compatible with, and encouraged by, the simultaneous inclusion of support for "sds", self-describing data files. While this disciplined data format has not yet achieved sufficient acceptance to be adopted as "standard", its use would relax the constraints that presently tend to freeze file formats and impede the introduction of new features. The following two tables list the elements recognized by *TEAPOT*: the first lists "standard" parameters, the second lists extensions. Though *TEAPOT* does not support bend elements other than sector bends (SBEND's), it is straightforward to model rectangular bends (RBEND's) and bend elements with other pole angles. An "awk" script "mad2tpot", that performs this operation mechanically is available. It can also be tailored to perform other minor variations. There is also a "wrapper" script called "teacozy" that permits going beyond the standard to use element names longer than eight characters or containing special characters.

ELEMENTS THAT ARE RECOGNIZED BY *TEAPOT*
(WITH "STANDARD" ALLOWED PARAMETERS)

DRIFT	L			drift space
SBEND	L	ANGLE	K1	K2
				sector bending magnet
(RBEND)	L	ANGLE	K1	K2
				rectangular bending magnet
QUADRUPOLE	L	K1	TILT	quadrupole
SEXTUPOLE	L	K2	TILT	sextupole
OCTUPOLE	L	K3	TILT	octupole
MULTIPOLE	KnL	Tn (n=1,9)		general thin multipole
SOLENOID	L	KS		solenoid
HKICK	L	KICK		horizontal closed orbit corrector
VKICK	L	KICK		vertical closed orbit corrector
HMON	L			horizontal monitor
VMON	L			vertical monitor
MON	L			monitor in both planes
MARKER				marker
RFCAVITY	L			accelerating cavity
ECOLLIMA	L	X(Y)SIZE		elliptic aperture
RCOLLIMA	L	X(Y)SIZE		rectangular aperture
XSEPTUM				horizontal septum

All elements also accept special TYPE declarations. See Appendix F. Since ECOLLIMA and RCOLLIMA elements are replaced internally by a thin mask at each end, they should probably not be given a TYPE declaration. For example making a collimator TYPE=X2 will give repeated output to fort.28.

Effective January, 1994, RBEND elements are allowed, primarily to support the modeling of wigglers. They are risky in the sense that the code using them has been less exercised and debugged than that supporting SBEND's.

ELEMENTS THAT ARE RECOGNIZED BY *TEAPOT*

(WITH "EXTENDED" ALLOWED PARAMETERS)

DRIFT				
SBEND	ANGLEMAX	TYPEAPER	X(Y)APSIZE	X(Y)OFFSET
WIGGLER	KZ	KX	AX	B0
	TYPEAPER	X(Y)APSIZE	X(Y)OFFSET	
QUADRUPOLE	K1MAX	TYPEAPER	X(Y)APSIZE	X(Y)OFFSET
SEXTUPOLE	K2MAX	TYPEAPER	X(Y)APSIZE	X(Y)OFFSET
OCTUPOLE	K3MAX	TYPEAPER	X(Y)APSIZE	X(Y)OFFSET
MULTIPOLE				
SOLENOID	KSMAX			
HKICK	KICKMAX	TYPEAPER	X(Y)APSIZE	X(Y)OFFSET
VKICK	KICKMAX	TYPEAPER	X(Y)APSIZE	X(Y)OFFSET
HMON		TYPEAPER	X(Y)APSIZE	X(Y)OFFSET
VMON		TYPEAPER	X(Y)APSIZE	X(Y)OFFSET
MON		TYPEAPER	X(Y)APSIZE	X(Y)OFFSET
MARKER				
RFCAVITY	VOLT	VOLTMAX	LAG	FREQ(=harm. num.)
	RAMPFREQ	RELMORAT	ENRGFRAC	ENRGLOSS
ECOLLIMA				
RCOLLIMA				
XSEPTUM	XIN(±)	XTHCKNSS(+)	KICK	KICKMAX
	SUP(±)	SDWN(>SUP)	LAMBRTSN	
		TYPEAPER	X(Y)APSIZE	X(Y)OFFSET
QUADEND	INPLOUTM	X(Y)POLANG	INVFOCLN	QUADLENG
BEAMBEAM	BMBMCHRG	X(Y,Z)BMOFSET	X(Y,Z)BMRMS	
STOCHAST			SIG(X,Y,CT)	DMPDEC(X,Y,CT)

Elements ending with “MAX” are all called “MXSTRENG” internally.

SPECIAL *TEAPOT* ELEMENTS

Except *TEAPOT* specials, elements and parameters are described in detail in the *MAD* manual (version 4).

Stochastic emittance alteration can be modeled with the STOCHAST command. This command was designed primarily with electrons in mind (but, with reinterpretation, the input parameters can be altered to allow modeling stochastic emittance growth in proton accelerators.) In electron accelerators, equilibrium between stochastic growth (due to quantum fluctuations) and damping (due to making up the lost energy in the rf cavities) is established within milliseconds. The inputs SIGX, SIGY, and SIGCT are the desired equilibrium rms values of bunch width, height, and length. The inputs DMPDECX, DMPDECY, and DMPDECCT are the fractional reduction per turn in phase space radius (with momentum axis normalized for circular phase space orbits). The actual stochastic exitation applied each turn can be inferred from *SUBROUTINE stochini*, which has detailed comments. Since noise and damping are applied only to the coordinates x, y, and ct (and not to the momenta px, py, pz) the damping decrement and noise strength are accordingly modified internally. The x damping is reckoned as fraction DMPDECX of the deviation from the off-momentum horizontal closed orbit appropriate to each particle. Vertical, y, damping has a similar reference orbit subtraction, but there is no longitudinal, ct, reference orbit subtraction.

Since the equilibration time will be many thousands of turns there is a large advantage in establishing starting distributions that are close to equilibrium. This can be done using the GAUSSIAN beam feature of TRACK or TRACKCLO.

Hardware maxima of elements physical elements that have a single natural strength parameter are assigned by ANGLEMAX, K1MAX, K2MAX, K3MAX, KSMAX, VOLTMAX, or KICKMAX. MXSTRENG is synonomous with each of these, depending on the context. By default all these hardware limits are taken to be infinite. At this time the only *TEAPOT* operations that respond usefully to finite values for these limits are HSTEER and VSTEER, which will refuse to

exceed the KICKMAX values of the HKICK or VKICK elements they use. (This last sentence was added to this manual at an instant in time at which the code to support bend maxima had been written and was "certain" to be installed "in the next couple of days". Unfortunately this has not yet happened.)

Most elements accept aperture definitions. During particle tracking particles outside the aperture are lost. The aperture check is made at the end of every drift, which is to say at the beginning of every active element. If an aperture check is required in the interior, the element has to be split at that point. This has the effect of inserting a (zero length) drift. Apertures can be elliptical, TYPEAPER=1, rectangular, TYPEAPER=2, or diamond, TYPEAPER=3. The default when no aperture parameters are given is a circle of radius one meter. It is not directly legitimate to use, say, TYPEAPER=ELLIPSE, but the same can be accomplished by first defining ELLIPSE=1 in the lattice file.

Aperture shapes and dimensions are illustrated in the figure. XAPSIZE and YAPSIZE are half-aperture dimensions. If only one is given the other one defaults to the same value. Cartesian coordinates of the aperture center are given by (XOFFSET,YOFFSET)—default is (0,0).

Aperture limits are not supported for some elements. For backward compatibility this includes collimator types ECOLLIMA and RCOLLIMA, which become less flexible than ordinary elements for defining apertures as a consequence. Also excluded are elements DRIFT, MARKER, and MULTIPOLE; also SOLENOID, RFCAVITY and XSEPTUM. (Technical aside: the latter three are segregated because they exercise "squatter's rights" in high order multipole storage locations of `fort.7`, the flat machine description file. This accomplishes nothing, but may simplify slightly the task of supporting variable maximum multipole index sometime in the future. The amount of extra storage space required to record aperture parameters and magnet strenth limits for every element is far from trivial. It is anticipated that some users will be forced to expand their computer memory allocation on this account. In principle this extra memory could be captured from

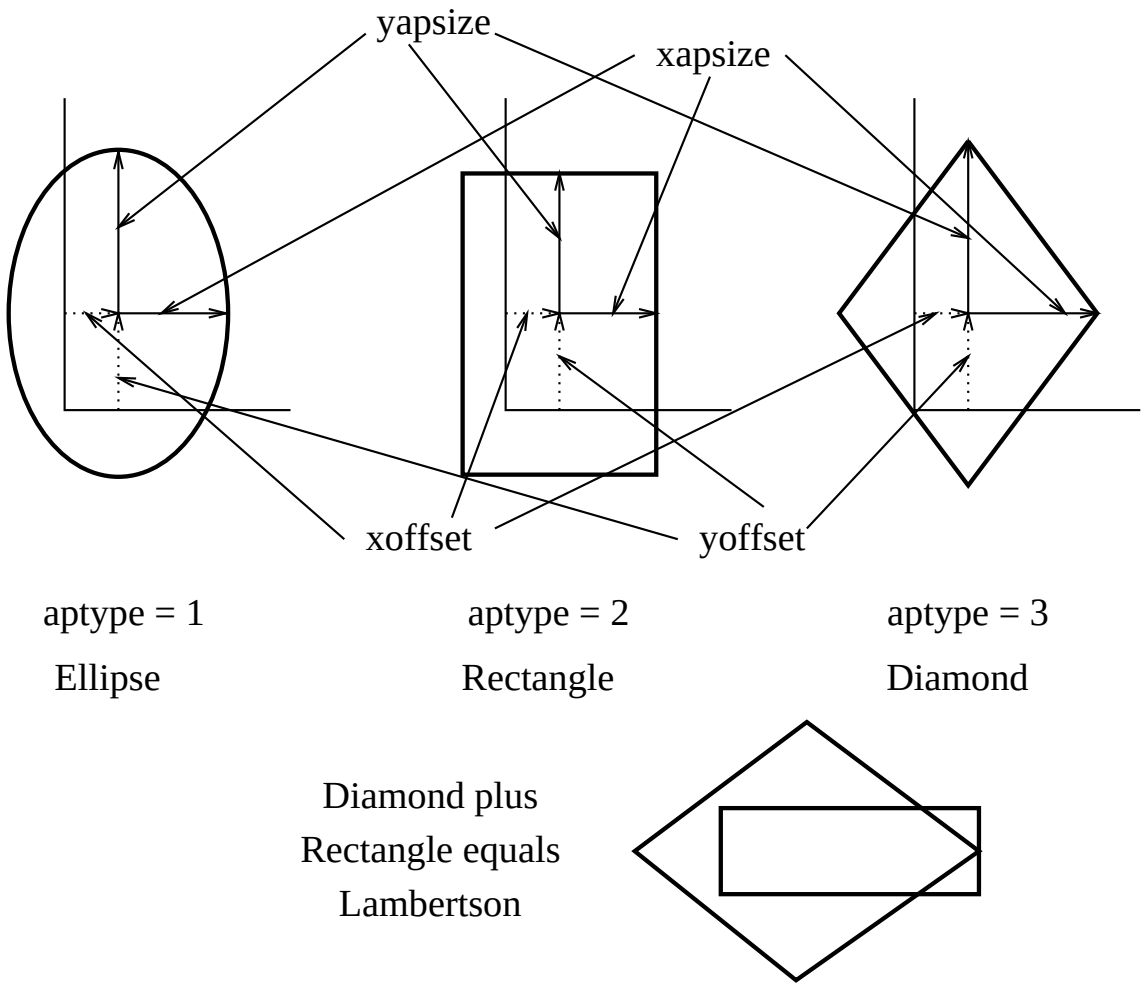

Figure 1: Aperture definition parameters.

the (rarely used) memory allotted to high order multipole coefficients.)

For modeling slow particle extraction the XSEPTUM element can be used. Presence of any XSEPTUM element forces EXTRACT to be true. With EXTRACT true, a BEAMTRAK mode is enabled that allows tune adjustment elements (or any other elements) to be adjusted in increments, periodically during particle tracking. Initially the TRACK command works as usual until every particles has been tracked for the requested number of turns. After that, and after

an ERRORS (or MODIFY) command has entered the desired element changes, a command CONTINUE, turns=<integer> causes tracking of the same set of particles to continue for the newly requested number of turns. This can be repeated as often as desired. EXTRACT also has the effect of suppressing subtraction of the closed orbit from coordinates printed out and also of suppressing certain beam moment calculations and their printout. When finished tracking a set of particles, if further operations are to be performed, it is first necessary to issue a DUMPBEAM command. This BEAMTRAK condition only occurs (at time of writing) because BEAMTRAK or GAUSSIAN tracking has been explicitly requested, or because an XSEPTUM element is present anywhere in the lattice.

The XSEPTUM element: administers horizontal deflection KICK to any particle outside its outer surface; has no effect on a particle inside its inner surface; and completely absorbs any particle hitting the septum electrode. To allow for septum thickness and particle slope the (necessarily positive) thickness XTHCKNSS and the longitudinal coordinates of the septum ends relative to its nominal position in the lattice, SUP and SDWN (>SUP), must be given. But the XSEPTUM element is otherwise equivalent to a marker—it must not be alloted any arc length in the lattice, even if SUP and SDWN have non-zero values. Effectively, this permits the element to intrude on its neighbors allotted space. As a result, the only influence of the values of SUP and SDWN is in the determination of whether a particle passes inside, passes outside, or hits the septum. Otherwise they have no effect.

Letting x_{out} be the x coordinate of the outer surface of the septum electrode, its outer surface coordinate is

$$x_{out} = XIN(1 + \text{XTHCKNSS}/|XIN|).$$

When a particle passes the element the code determines which side of its trajectory each of the four corners of the septum electrode inhabits and determines its fate accordingly: if all four corners are outside the trajectory, the septum has

no effect; if all are inside the particle is kicked; otherwise the particle is "absorbed". If the XSEPTUM attribute list includes LAMBRTSN=1, the particle is also absorbed in the case where it would otherwise be kicked.

Standard practice is to make the assignment, TYPE=X1, for the regular septum. This causes printout (`fort.18`) of every particle on every turn at the septum. Similarly, making the assignment, TYPE=X2, for the Lambertson causes turn-by-turn printout there (`fort.28`). Similary output at any two other locations can be obtained by placing an element of TYPE=X3 (`fort.38`), or TYPE=X4 (`fort.48`). TRACK, not TRACKCLO, is probably appropriate, since the septum location is presumably fixed in absolute coordinates, while TRACKCLO calculates displacements from the closed orbit.

For collimators, XSIZE and YSIZE denote the half-axes or half-widths. A non-zero length collimator is split into two elements with a drift region between them. In tracking the aperture is checked both at the beginning and end of this region. A conical shaped collimator can be modeled by two zero length collimators separated by a drift.

Treatment of solenoids is discussed in Appendix G. The solenoid strength parameter has the *MAD* definition, KS = $cB_0/(pc/e)$. The strength parameter used in TRANSPORT and in Appendix G is $K = cB_0/(2pc/e)$ = (KS)/2. The bend angle of a particle with transverse momentum $p_\perp$ in a solenoid of length L is $2\theta = 2KL$ = (KS)L.

For RFCAVITY the entries used by *TEAPOT* differ from *MAD*. Treatment of longitudinal quantities is described in the report teapot_longit.ps. FREQ is the harmonic number (which can also be thought of as the frequency in units of the revolution frequency.) This permits *TEAPOT* to use its own calculated value for the revolution frequency. For the normal choice of units, the units of VOLT are GV, gigavolts.

For storage ring operation leave RELMORAT undefined or set it to 0.0, in which case RAMPFREQ is ignored. ENRGLOSS is the energy loss per turn

(presumeably from synchrotron radiation) to be made up by the cavity; normal units are GeV. For more than one cavity the values of ENRGLOSS must sum to the correct theoretical value. In tracking individual particles the energy loss (equal to ENRGLOSS) is taken all at once at the cavity, not uniformly around the ring. As a result the program cannot be used, without special coding, to study spiralling in between cavities, or to obtain damping partition numbers. (It would be inconsistent to do that in any case without introducing stochastic anti-damping.) Adiabatic damping that accompanies energy ramping is correctly described. With ENRGLOSS non-zero in a cavity, the RF phase of that cavity is adjusted to make up that energy in that cavity. Hence effects of nonlinear rf shape, such as particle loss if there is insufficient RF voltage, should be modeled correctly. Each cavity phase can be shifted by LAG, but if the ENRGLOSS value is non-zero it must be done with care as it is likely to upset the global energy recovery; see below for more about LAG.

For simulation of acceleration, RAMPFREQ is the frequency in Hertz of the accelerator magnetic field variation, assumed to be a biased sinuisoid, with the starting momentum(times c) being p00c and the magnetic field varying as

RELMORAT = relative momentum ratio = [(B(max)-B(min))/B(min)].

p0c = p00c*(1.0 + 0.5*RELMORAT*(1 - cos(2*pi*telapsed*RAMPFREQ))).

The rf phase is calculated internally to correspond to the phase needed by the reference particle (the RF voltage VOLT being assumed to be constant) to match the given reference-momentum dependence. The progam chooses the phase that will cause stable oscillation. (This includes switching the phase on passage through transition.) If there is more than one RF cavity they should be given different names, so their parameters can be assigned independently. If ENRGFRAC is defined (not 0.0) then the cavity phase is adjusted so the cavity makes up ENRGFRAC*(energy gain per turn). If ENRGFRAC is defined for one cavity it should be defined for all, and the values must add to 1. If both energy loss and energy ramping are in effect it is just asking for trouble to not make ENRGFRAC

proportional to ENRGLOSS. If ENRGFRAC is not defined then the cavity makes up energy proportional to the arc length since the most recent previous cavity. For uniform azimuthal spacing that is what would normally be desired. This can be altered using the variable LAG which shifts the RF phase of the individual cavity, relative to what the program would set. This needs to be tailored to the actual geometry. As an example, consider two cavities side-by-side, with no other cavities in the machine. With LAG=0 for both cavities, and ENRGFRAC not defined, the first cavity will automatically be phased to make up the entire energy gain required for the ramp. Assuming one really wants equal energy gain from the two cavities one could, in this case, set

$$\text{LAG}=\mp 1/2 \arcsin \Delta E/VOLT$$

where ΔE is the energy gain per full turn. This is not really recommended as it would not allow for variable energy gain per turn and frequency modulation. For further tailoring of the RF, or for changing the energy ramp, it is necessary to change the source code (within the do 190 loop of subroutine traconc) appropriately, and to check by stepping through that section.

In order to be restorable in the writefile-readfile sequence, all RF parameters are assigned names in the "flat" machine description arrays "atw(m, ielem)" and "btw(m, ielem)", which can be written to the "`fort.7`" external file. The parameters fit in slots that contain multipole elements for magnetic elements. (This special use of the structure renders it non-flat, but never mind.) If the element number of the rf cavity is "i", the slots used are btw(7,i)=relmorat, atw(7,i)=rampfreq, btw(8,i)=vrf, atw(8,i)=phirf, btw(9,i)=freqhnum, atw(6,i)=enrgfrac, atw(9,i)=enrgloss. Naturally the code must take care not to interpret these entries as magnetic multipole elements. The parameters of XSEPTUM, QUADEND, and BEAMBEAM elements are also "packed" into high atw() and btw() slots. The rationale for "breaking" the flat structure is that only a few of these elements are typically present.

No vertical bends are allowed for calculation of the Reference Orbit (the

MAKETHIN command), however vertical bends may be introduced as errors, misalignments, or in orbit correction—this is done with ERRORS or MODIFY. (Note added 17 March 1992 : A version of the code called ftpotv supports vertical bends. That code has had little testing and fails to support certain features spelled out in this manual.) All elements are converted to single thin elements except quadrupoles or dipoles of TYPE = 'IR','IR2','IR4','IR8', 'IR16', which are split into four, eight, sixteen, thirty-two, sixty-four thin quadrupoles, respectively. Disagreement with other modeling programs on the precise tune values, or on whether a lattice is even stable, can often be rectified by splitting up some or all quadrupoles or dipoles in this way. Splitting dipole elements is useful mainly for combined function lattices or for small ring for which the dipoles provide appreciable focusing. Quadrupoles in regions of rapid β-function variation (often near low β intersection regions, which is the source of the "ir" notation) often require subdivision also.

The effects of nonzero entry angle into bend elements should be modeled with multipole elements. This includes RBEND's (i.e. rectangular bends) in the standard input format. One exception to this is permitted for zero length bends which are useful for comparison with simple analytic formulas. Rectangular bends of zero length are modeled by SBEND elements of very short, but not zero, length, of TYPE = 'NOFO'. See Appendix F. An *awk* script *mad2tpot.awk* exists that can be used to mechanically insert end effect multipoles into the input data file, or which can be used as an example to perform to do it yourself "by hand".

Effective January 1994, RBEND's are allowed, especially for modeling wigglers. This incorporates the vertical focusing and absence of horizontal focusing that accompany RBEND's. Modeling of wigglers is discussed further in Appendix I.

A QUADEND element placed at the end of a quadrupole will account for the inevitable solenoidal field (causing a deflection of cubic order, like an octupole) as well as a sextupole, if the entrance or exit angle deviates from nor-

mal. A QUADEND element must be introduced explicitly at each end, with "INPLOUTM=+1" indicating entrance and "INPLOUTM=-1" indicating exit. The horizontal pole-angle is entered as XPOLANG, with the sign depending on whether it is entrance or exit. In both cases a positive sign indicates that a particle with $x > 0$ "sees" a longer quadrupole because of the pole angle. Similarly, the sign of the vertical pole angle YPOLANG is positive if a particle with $y > 0$ "sees" a longer quad. The quad length is QUADLENG and its inverse focal length is INVFOCLN, both in meters. (Actually, since only the ratio enters, except for needing to be the same, the units do not matter. The focusing strength per unit length K1=INVFOCLN/QUADLENG of the quadrupole, for which this element forms the end, must be the same as for the QUADRUPOLE itself.) If a non-zero value of QUADLENG is not given the program will issue an error message.

Passage through an oncoming beam is represented by a BEAMBEAM element. This is necessarily a "weak-strong" representation with the single particle being tracked seeing the constant fields of the other beam like those of any other fixed beam-line element. Since passage through the bunch is treated as occurring impulsively at a single thin longitudinal element, it is only the longitudinally integrated transverse kick that matters. Hence, though rms bunch length ZBMRMS is accepted for possible future use, it is not used at present. The transverse profiles are entered by XBMRMS aqnd YBMRMS. It is the responsibility of the user to calculate these for example by knowing the β-functions and the emittances. All profiles are assumed to be Gaussian ellipsoidal. The total total bunch charge is BMBMCHRG which is equal in magnitude to the number of particles in the bunch. The sign of BMBMCHRG determines whether the linear beam-beam effect is focusing or defocusing. To allow correctly for particle stiffness, which is calculated internally, the momentum (or energy) must have been correctly entered by an ANALYSIS command. It is important to remember that the units of energy are GeV (the only non-MKS unit in *TEAPOT*) and that the particle type (electron or proton) must be correctly specified in all analysis

and tracking commands. The reason this is emphasized here is that, unlike most lattice elements for which the particle stiffness is "scaled out", (for example a quadrupole is characterized by its focal length), that is not true of BEAMBEAM. Horizontal, vertical, and longitudinal offsets from the ideal orbit are entered by X(Y,Z)BMOFSET. This can be used, for example, to represent parasitic beam-beam crossings. Presence of a BEAMBEAM element will be reflected in small amplitude tune shifts (of QAA and QDD) calculated by ANALYSIS but not (of QX and QY) calculated by TWISS. If these agree in the absence of BEAMBEAM element (as they should at least for ideal lattices) the difference can be used as a check of code behavior. More recommended for the same purpose is to FFT multiturn tracking data. The code currently assumes that both beams are highly relativistic, but a comment in `SUBROUTINE trkbmbm` indicates how that can be rectified if necessary.

There is a version of the code called *TEASPOON* which tracks particle spins. It is due to Sateesh Mane and is available via anonymous ftp.

Effective 16 August, 1994, type WIGGLER is supported. See the report teapot_wiggler.ps, written by Weiru Wang and R. Talman for details. One wiggler section consists of a half-pole, full-pole, half-pole sequence. In the code such a section is modeled by a R-matrix and T-matrix only (making it non-symplectic.) A full wiggler is made up of, say, 12 such sections. The inputs are: $B0$, the maximum magnetic field; KZ, the longitudinal wave number with the z-dependence taken as sinuisoidal; KX, the horizontal, possibly imaginary "wave number"; the corresponding quantity KY is calculated internally using $KZ^2 = KX^2 + KY^2$; AX is the maximum transverse orbit displacement of the actual orbit, measured from a straight line coinciding with the orbit before the input and after the output. It satisfies $AX = 2 * c * B0/(pc/e)/KZ^2$ but it must be included as input; p is obtained internally by inverting this equation.

ANALYSIS

ANALYSIS, {ENERGY, PC} = <value>, [PARTICLE = {PROTON, ELECTRON},] XTYP = <value>, PXTYP = <value>, YTYP = <value>, PYTYP = <value>, DPTYP = <value>, DELTA = <value>, [TAPE, [UNIT = <value>],], [PRINT]

The ANALYSIS command finds the closed orbit and performs a full Twiss analysis of the machine by tracking particles with the given ("typical") initial amplitudes. ENERGY is given in GeV and DELTA allows an analysis to be performed at any momentum offset DELTA. DPTYP is then the change in momentum used to calculate η and the chromaticity. PARTICLE is optional, the default is PROTON. The UNIT number for the output file can be selected, and several analyses can be run on a machine (for example, before and after fitting). The default UNIT is 4 (see Appendix D). Note that if one wants to use tpot plot package (XPOT) "TAPE" is essential in the ANALYSIS command. The designation "TAPE", handed down via the code *MAD*, is somewhat archaic and misleading; as well as not in fact resulting in any magnetic tape being written, it causes certain reference quantities to be updated. Though the usage is not yet consistent, invoking TAPE sometimes has the effect of updating what constitutes the "central" machine parameters. Tape output is obtained for every ANALYSIS command and the last one is retained. Inclusion of the PRINT directive causes a table of Twiss parameters to be written to standard output along with a reasonably abbreviated progress-describing commentary.

BPMERROR, KDERR is similar These are commands like the ERRORS command but specialized to SSC requirements. They can be regarded as templates from which special error requirements can be patterned,

BPMERRORS, {<element1>, <element2>,..., <element16>} , SIGXWRTQ = <value>, SIGYWRTQ = <value>, SIGXSOLO = <value>}, SIGYSOLO = <value>, SIGXWRTS = <value>, SIGYWRTS = <value>, [CUT = <value>, { SEED = <value>, SYSTEMATIC = <value>}]

where SIGXSOLO and SIGYSOLO request displacement of the BPM with respect to the reference orbit, SIGXWRTQ and SIGYWRTQ request displacement of the BPM with respect to the closest quadrupole, SIGXWRTS and SIGYWRTS request displacement of the BPM with respect to the closest sextupole.

CHFITOLD, see DECOUPLE

CHROMFIT, see DECOUPLE

CLOSEDORBIT

CLOSEDORBIT, X = <value>, PX = <value>, Y = <value>, PY = <value>

This command sets the closed orbit to the given values. This is useful if the machine has strong non-linearities and the program has difficulty finding the closed orbit. In that case, this command would be followed by an ANALYSIS command, and the values input would serve as a starting point for the closed orbit search.

CPLTRK

CPLTRK, KICKX = <value>, KICKY = <value>, THRESHOLD = <value>

This command computes interpolated coordinates at BPM's which measure beam position in only one plane for use in decoupling. To accomplish this two particles are tracked, one with initial kick px=`kickx` and one with py=`kicky` (in mrad) at the origin. At each detector of type `chcd` or `cvcd`, the coordinates of the tracked particles and the ideal twiss parameters are recorded. At each detector, the missing coordinate (in the plane of the tracked particle for greatest accuracy) is obtained by interpolation, using the ideal twiss parameters and the coordinates of the tracked particle at the adjacent detectors on either side. Output files of the ideal-interpolated and error-present actual coordinates at each `chcd` and `cvcd` are written (see the description of the fort.50, fort.51, fort.60 and fort.61 in the Appendix). These can be used to study the accuracy of the interpolation in detail. The rms of the difference between the interpolated and actual coordinates

is computed as a figure-of-merit for the interpolation in each plane. A warning message is printed whenever the fractional deviation of the interpolated from the actual coordinate is greater than the user-specified `threshold` at a detector.

DECOUPLE (DECOUPLO is different), TUNETHIN, CHROMFIT, CHFITOLD, TUNTHOLD

All these commands must be preceded by an ANALYSIS command. The <parameter name>s cannot be global parameters, and must be specified by the multipole coefficient, for example, qf[b1] (NOT k1). The output results of the fits are values for the multipole coefficients; they are related to the k_n used by *MAD* by

$$\tilde{b}_n = k_n L/n!$$

where $\tilde{b}_n = b_n(LB_0/\text{“}B\rho\text{”})$. All elements of the “family” given in the command are assumed to have the *same* value and are set to have the same value by the fitting routines. For DECOUPLE there are three variants:

DECOUPLE, A11 = <parameter name>, A12 = <parameter name>, A13 = <parameter name>, A14 = <parameter name>

DECOUPLE, A11 = <parameter name>, A12 = <parameter name>, A13 = <parameter name>, A14 = <parameter name> , A11M = <parameter name>, A12M = <parameter name>, A13M = <parameter name>, A14M = <parameter name>

DECOUPLE, A11 = <parameter name>, A12 = <parameter name>, B1F = <parameter name>, B2F = <parameter name> , TUNEX = <value>, TUNEY = <value>

The first variant zeroes the four matrix elements Ryx, Ryx′, Ry′x, and Ry′x′ using the specified skew quadrupoles. The second variant sets antisymmetric skew quad pairs (in lattices having mirror symmetry.) For this it is necesary to include the parameters A11M, A12M, A13M and A14M. The element containing A11M must be mirror symmetric with the element containing A11 and

the fit will insure A11M = -A11 and so on. The third variant (courtesy of Dan Trbojevic and Steve Tepikian) zeros two elements, E12 and E22, of the matrix $E = B + \overline{C}$, (see Appendix G), as well as adjusting the tunes to the values requested for TUNEX and TUNEY. This adjustment sets determinant $|E|$ to zero, which is a necessary condition for making the tunes coincide. An example skew quadrupole declaration is "a11=sk2[a1]". An example erect quadrupole declaration is "b1f=quadhf[b1]".

TUNETHIN, MUX = <value>, MUY = <value>, B1F = <parameter name>, B1D = <parameter name>, [NUMTRIES = <value>, TOLERANCE = <value>, STEPSIZE = <value>]

The TUNETHIN command fits the tunes of the thin lens machine using the specified quadrupoles. NUMTRIES is the maximum number of iterations for the fitting, TOLERANCE is the maximum absolute value of the difference between the requested values and the fitted values at convergence, and STEPSIZE is the size of the first step in the fit. The defaults are 100 for NUMTRIES and $10^{\perp 6}$ for TOLERANCE.

Effective 2 March 1994, both TUNETHIN and CHROMFIT were changed in ways that were almost, but not exactly, backward compatible. To obtain precise backward compatibility one must use TUNTHOLD or CHFITOLD, with exactly the syntax just described. One can also use the new versions of TUNETHIN or CHROMFIT with the same syntax and obtain essentially identical results. The only reason for difference is that the algorithm alters correction elements multiplicatively (old $\rightarrow$ new=old*(1+Δ)) rather than additively (old $\rightarrow$ new = old+Δ'). In cases where "old" was zero this will not work and it will be necessary either to use TUNTHOLD or CHFITOLD or to assign non-zero (but otherwise non-critical) starting values. The reason for this change was to support the ganging of non-identical elements into a family sharing the same power bus. In that case the thin lens length-strength products will be proportional to the element length for each element. The reason this is almost backward compatible

is that previously it was assumed that all members of each family were identical, and hence had identical strengths.

To make this feature more easily applicable, a new syntax for TUNETHIN is allowed. Each element of the family (i.e. on the same bus) is declared to be "type = xtun" or "type = ytun" as appropriate. (It is assumed there are precisely two families.) Then in the TUNETHIN line one includes the entries "famf = xtun[b1]" , "famd = ytun[b1]," (note that there are no variables here, it must be exactly as shown) rather than giving element names to specify the elements to be tuned. (If you think this syntax is obscure you should look at the code supporting it.) Example input file lines fom the file "$HOME/tpot/ftpot/test/dat/fam_tune" follow:

```
quadhf :  quadrupo, l = lq , k1 = kq1, type = xtun
quadvf :  quadrupo, l = lq , k1 = kq2, type = ytun
sext1 :  sext, l = ls, k2 =  ks1, type = xchr
sext2 :  sext, l = ls, k2 =  ks2, type = ychr
tunethin, mux = 2.38, muy = 2.42, famf=xtun[b1] , famd=ytun[b1]
chromfit, famf=xchr[b2], famd=ychr[b2], chromx = 0.0, chromy = 0.0
```

Note from this example that CHROMFIT supports the corresponding syntax, but with "xchr" and "ychr". The keywords "famf" and "famd" are mnemonic for "horizontal family" and "vertical family". The keywords "xtun", "ytun", "xchr", and "ychr", cannot be altered (except by simple hacking of the code.)

CHROMFIT, CHROMX = <value>, CHROMY = <value>, B2F= <parameter name>, B2D = <parameter name>, [NUMTRIES = <value>, TOLERANCE = <value>, STEPSIZE = <value>]

CHROMFIT fits the chromaticity of the thin lens machine with the specified sextupoles. NUMTRIES is the maximum number of iterations for the fitting, TOLERANCE is the maximum absolute value of the difference between the requested values and the fitted values at convergence, and STEPSIZE is the size of the first step in the fit. The defaults are 10 for NUMTRIES and $10^{\perp 4}$ for TOLERANCE.

DECOUPLO

DECOUPLO, [NUMTRIES = <value>, BADNESS = <value>]

DECOUPLO is similar to HSTEER and VSTEER—read the HSTEER entry for a more complete description. The prescription used by DECOUPLO to decouple the lattice (i.e. make the approximately-horizontal eigenplane be as nearly horizontal as possible in a least-squares sense) is described in Section 7 of Appendix G. At least one skew adjusting element (i.e. a multipole with a t1 entry and with k1l set to a definite value, normally zero) with TYPE='cpla' and at least as many TYPE='cpld' detector elements must be present. DECOUPLO includes the capability of iterating until a goal badness (BADNESS) is reached, or of iterating a specified number of times (NUMTRIES). If NUMTRIES and BADNESS are not explicity set, NUMTRIES is set to 1, whereas if BADNESS is specified but NUMTRIES is not, NUMTRIES defaults to 10.

DUMPBEAM, see also TRACK

By default (effective 10 July 1993) particles that have been tracked by TRACK(CLO) are retained for possible subsequent tracking. This makes it possible to adjust lattice parameters and continue tracking the same particles, for example to simulate resonant extraction. In the event that any other instruction is to be processed after the RUN instruction has completed, it is necessary to follow the RUN line with a line containing only DUMPBEAM if the BEAMTRAK flag has been set (either by the presence of an XSEPTUM element or by GAUSSIAN tracking.) This retains the code's backward compatiblity, for lattices without septum, in that control continues to be returned automatically to subroutine *cntrol* when the requested number of particles and turns has been completed. Subroutine *cntrol*) fields all subsequent teapot instructions (such as STOP, TRACK, ERRORS, etc.) (A DUMPBEAM line present when it is not necessary will cause the command parser to issue an unexplained warning message '*** error *** "," expected, enter scanning mode' but will not otherwise cause malfunction.) (i.e. make the approximately-horizontal eigenplane be as nearly horizontal as possible

in a least-squares sense. **ERRORS, see also MODIFY and PARMFIT**

ERRORS, {<element1>, <element2>,..., <element16>} , {SIG{A, B}{0-9} = <value>, SIGX = <value>, SIGY = <value>, SIGTHETA = <value>}, [BZEROL = <value>, CUT = <value>, FILE = <value>, { SEED = <value>, SYSTEMATIC = <value>}, ORBCHEAT = 0,1]

This command sets multipole and position errors for all named elements. The SIGB1, SIGA1, etc., are the sigmas for the errors of the multipole components b_n, a_n specified by

$$B_y + iB_x = B_0 \sum_{n=0}^{n_{\max}} (b_n + i\, a_n)[(x - \Delta x) + i(y - \Delta y)]^n.$$

For error-free dipoles $b_0 = 1$, $a_0 = 0$. As an example, suppose the (fully relativistic) nominal accelerator energy is $pc/e = E(\text{TeV}) = 20$. Conventionally a quantity "$B\rho$" is defined by

$$\text{“}B\rho\text{”} = \frac{pc/e}{c} \simeq \frac{2 \times 10^{13}\,\text{Volts}}{3 \times 10^8\,\text{m/sec}} = 0.667 \times 10^5 \ \text{Tesla-m},$$

When the ERRORS command is applied to a dipole element of length L, a factor BZEROL(dipole)= $B_0 L/$"$B\rho$", which is the bend angle in radians in the small angle limit, is calculated internally—it should not be specified in the ERRORS command. The magnetic field then is used internally in the form

$$\frac{LB_y + iLB_x}{\text{“}B\rho\text{”}} = \text{BZEROL}[1 + \sum_{n=1}^{n_{\max}} (b_n + i\, a_n)((x - \Delta x) + i(y - \Delta y))^n]$$

From this form it can be seen that the b_n and a_n are field errors expressed as fractions of the nominal dipole field. The actual numerical values stored internal to *TEAPOT* are

$$\widetilde{b_0} = \text{BZEROL}, \quad \widetilde{a_n} = \text{BZEROL} \cdot a_n, \quad \widetilde{b_n} = \text{BZEROL} \cdot b_n.$$

These quantities can be output in ascii form to file "`fort.77`" by issuing a "write-file, slow" command. They can be read back (modified if desired) from the same

file, renamed "`fort.7`" by issuing a "readfile, slow" command. (If editing access to these quantities is not required, the "fast" option is faster and more accurate.) Provisions that allow the code to determine values for the a_n and b_n internally either as systematic or as random variables are explained below.

For elements other than dipoles the field is expressed as a similar series, but the b_0 element vanishes and a different scale factor needs to be supplied, either internally or externally. For historical, and not easily defensible reasons, this is done by changing the meaning of the dimensionless parameter BZEROL, depending on the element type. Because of the variety of conventional units and definitions, and the variety of element types, it is difficult to spell out in clear and simple terms how this common factor is to be interpreted, calculated, and supplied in all cases. There is a script described below that can help to assure that correct values are being used. Here we provide an illustrative calculation for the next most important case after dipoles, namely quadrupoles.

Suppose the following: the nominal, maximum accelerator energy is $p_m c/e = E_m(\mathrm{TeV}) = 20$; (being in the fully relativistic limit) all magnetic fields are proportional to E; and the maximum nominal quadrupole gradient of the main quadrupoles is $g_m(\mathrm{Tesla/m}) = 215$. In one conventional description, the magnetic field variation in the horizontal plane ($y = 0$), of an imperfect quadrupole, is written

$$B_y(E;x) = \frac{E}{20}\, g\, [x + 10^{\perp 4} \sum_{n=2}^{n_{\max}} (b_n^Q + i a_n^Q) \frac{(x + 0\, i)^n}{R_r^{n \perp 1}}],$$

where R_r is a reference radius, say 1 cm. The value of R_r and the factor $10^{\perp 4}$ are normally chosen such that the numerical values of a_n^Q and b_n^Q are of order 1 for "bad", low order, multipoles and much less than 1 for high order multipoles. The strength coefficient of the quadrupole (defined the same way in *MAD* and *TEAPOT*) is

$$k = \frac{dB_m/dx}{\text{“}B\rho\text{”}_m} = \frac{dB/dx}{\text{“}B\rho\text{”}} = \frac{g}{\text{“}B\rho\text{”}} = \frac{215}{0.667 \times 10^5}\, \mathrm{m}^{\perp 2},$$

independent of beam energy. Dividing the multipole series by "$B\rho$" (in order to cancel dependence on E), multiplying by quadrupole length L, and separating into an ideal term and an error term yields

$$\left.\frac{B_y(E;x)L}{\text{“}B\rho\text{”}}\right|_{E=20} = \frac{gL}{\text{“}B\rho\text{”}}x + \frac{gL}{\text{“}B\rho\text{”}}10^{\perp 4}\sum_{n=2}^{n_{\max}}(b_n^Q + ia_n^Q)\frac{x^n}{R_r^{n\perp 1}}$$
$$= \text{BZEROL}\cdot x + \text{BZEROL}\cdot\sum_{n=2}^{n_{\max}}(b_n^Q + ia_n^Q)x^n$$

The factors k and L, required to calculate the ideal term, are supplied with the QUADRUPOLE declaration statement, and using these,

$$\text{BZEROL(quadrupole)} = \frac{gL}{\text{“}B\rho\text{”}}$$

is calculated internally. This same value is used as the common factor for the error series. Matching coefficients we have

$$a_n = 10^{\perp 4}\frac{a_n^Q}{R_r^{n\perp 1}}, \quad b_n = 10^{\perp 4}\frac{b_n^Q}{R_r^{n\perp 1}}.$$

Coefficients of different orders have different dimensionality. Without exception, *TEAPOT* always uses meters as the length dimension. Since 1 cm is a typical value of R_r, the a_n and b_n tend to increase by a factor of order 100 when the order is increased by one. As explained above for dipoles, these quantities are multiplied by BZEROL and stored as $\widetilde{a_n}$ and $\widetilde{b_n}$.

For elements of type MULTIPOLE, the same error series is used, but the "ideal" part of the multipole series vanishes and the value of BZEROL has to be supplied in the form BZEROL=`value` in the ERRORS command.

SIGX and SIGY are the sigmas for x and y displacements, and SIGTHETA is the sigma for an angular rotation about the s axis. CUT is the cutoff in sigma for the generation of random errors. The default CUT is 3 sigma. SEED is the seed

for the random number generator, one per run. If no seed is specified, the program supplies one. *TEAPOT* accepts only the first seed input; subsequent seed entries in the same run are ignored. SYSTEMATIC is a request for systematic errors, and the value is the factor the sigmas are multiplied by to get the systematic errors. When used in this way, inputs beginning with SIG, such as {SIG{A, B}{0-9}, stand for systematic error values, **not** sigmas. Refer to the actual code of subroutine "errors" for more information.

The ERRORS command must be preceded by a MAKETHIN command. It has been found preferable to place ERRORS commands for quadrupoles containing SIGTHETA as late as possible because the presence of coupling seriously degrades the performance of the closed orbit finding algorithm.

The error distribution can be read from a file, using FILE=`value`, where `value` is the unit number read. In this file the first two lines are primarily for the user's convenience; *TEAPOT* keeps track of them but they do not influence the run. The first is a user-provided seed, presumably keyed to the subsequent entries if they were Monte Carlo produced. The second is an eight character string saved as a label. Subsequent lines are matched sequentially with occurrences of the named element in the lattice and the entries (one per line) after multiplication by a systematic factor, are applied to the corresponding element. The multipole element affected and a common multiplicative factor are indicated by the particular choices in {{SIG{A, B}{0-9}, SIGX, SIGY, SIGTHETA}=`value`. This `value` multiplies the quantities read from the file and, by the BZEROL `value` (supplied either internally or externally) and the results are applied sequentially to the named elements. The number of entries in the file must match the number of occurrences of the named element. Other error distributions in the same format can be concatenated to the same file and read by subsequent error commands. The SYSTEMATIC directive should not be used in conjunction with FILE.

An ERRORS command can include "ORBCHEAT=1" or "ORBCHEAT=0"

where "0/1" stand for "false/true" and no other values are allowed. The ERRORS command must be preceeded by an ANALYSIS with TAPE turned on. The ORBCHEAT directive only makes sense if there are reasonable complements of horizontal (type=KHKA) and vertical (type=KVKA) elements. When ORBCHEAT is true, the closest preceding and closest following steering elements are automatically adjusted to compensate for any steering that occurs when errors are added to elements being modified by the particular ERRORS command. The compensation is such that the previous corrector, present element, and following corrector form a closed three-bump. Compared to most other features in *TEAPOT* this is "cheating" because it uses information that the computer "knows" but which would not not be known operationally. As well as checking self-consistency, there are two intended uses for ORBCHEAT. One is to check that the particular distribution of steering elements is capable, given perfect information, of achieving a satisfactory closed orbit in the presence of the particular errors being added. The other is to expedite investigations of other multipole effects when one does not wish to take time to fix the orbit operationally. The state of ORBCHEAT in one ERRORS command has no effect on any other ERRORS command. ORBCHEAT cannot be used with KDERRS or any of the sector steering algorithms HSTEER1, HSTEER2, etc. Switching to "ORBCHEAT=0" is equivalent to removing ORBCHEAT altogether.

There is a (perl) script available, called "Fort7Moments.pl" which extracts the actual errors that have been entered, from the "`fort.7`" file generated by "writefile, slow", and then works out their means and sigmas. Since there are various conventions and ambiguities in multipole definitions, it is good practice to confirm what errors are actually being assigned.

The ERRORS command can also be used ahead of a PARMFIT instruction in order to flag elements whose strengths will be used as the parameters to be adjusted by PARMFIT to improve the large amplitude lattice behavior in the presence of nonlinearity. Any of the magnet element names, {SIG{A, B}{0-9} can be prepended by the letter F, to yield {FSIG{A, B}{0-9} to flag that element

for fitting. Effectively this establishes a "family" of correctors of that particular multipolarity, superimposed on all elements at all locations in the lattice with the element name contained in that ERRORS command.

The instruction MODIFY is entirely equivalent to ERRORS.

HSTEER, VSTEER, HSTEER[1-6], VSTEER[1-6]

Any one of these three commands must be preceded by an ANALYSIS command. HSTEER and VSTEER are normally used together, and iteratively. For example: ANALYSIS, HSTEER, ANALYSIS, VSTEER, ANALYSIS, HSTEER, ANALYSIS, VSTEER.

HSTEER utilizes the calculations performed in the previous ANALYSIS assuming that there is at least one TYPE='khka' steering adjustor element (i.e. a hkick type element with kick set to a definite value, normally zero) and at least as many TYPE='khkd' detector elements in the input lattice description, to flatten the orbit (i.e. improve the central orbit steering) horizontally. Elements can be placed arbitrarily, except that placing two adjustors with no separation makes the solution indeterminate. Also the possibility of two adjustors "fighting each other" needs to be kept in mind and made a part of whatever studies the code is being used for. It is all right for detectors to be arbitrarily close to each other—in fact, this is a legitimate way to assign increased weight to the beam detection in regions of special importance such as intersection regions. The least-squares procedure used is described in Section 6 of Appendix H.

VSTEER utilizes the calculations performed in the previous ANALYSIS assuming that there is at least one TYPE='kvka' steering adjustor element (i.e. a vkick type element with kick set to a definite value, normally zero) and at least as many TYPE='kvkd' detector elements in the input lattice description, to flatten the orbit (i.e. improve the central orbit steering) vertically. Elements can be placed arbitrarily. The least-squares procedure used is described in Section 6 of Appendix H.

HSTEER[1-6], VSTEER[1-6] perform steering that emphasizes the closed-orbit quality in a particular sector, or at any set of locations defined to make up a family. As many as 6 matched families of adjustors and detectors of both horizontal and vertical detectors are allowed. As an example, HSTEER1 minimizes a sum of squares of closed-orbit deviations at all family 1 detectors as well as at all detectors that are not members of any family; it uses only family 1 adjustors. (The inclusion of detectors not belonging to the family is to avoid large excursions elsewhere in the lattice.) Family membership is specified by the element name—the family number is a number from 1 to 6 which is the third character of the element name. This naming distinction is ignored by HSTEER and VSTEER.

KDERRS, see BPMERRORS

MAKETHIN

MAKETHIN, [PRINT = {BEAMLINE, ELEMENTS}]

The MAKETHIN command creates the data structures which represent the thin lens machine in a form appropriate for tracking. The reference orbit is found, and the thick element representation is converted to a thin element one. This command should follow the TUNE and USE commands, and must precede all the following commands. Inclusion of the PRINT directive causes geometric information at each element to be written to standard output.

KDERRS, see BPMERRORS

MODIFY, synonym for ERRORS

The ERRORS command was initially intended only to introduce random or systematic errors, but it has often been used to make intentional lattice modifications or compensations. Use of the synonym MODIFY should make the input file more intuititive when the command is used to introduce intentional lattice madifications. All attributes of ERRORS and MODIFY are identical.

PARMFIT, see also ERRORS

PARMFIT, {ENERGY, PC} = <value>, XTYP = <value>, PX-
TYP = <value>, YTYP = <value>, PYTYP = <value>, DPTYP
= <value>, DELTA = <value>

The PARMFIT instruction invokes a nonlinear fitting routine that adjusts all elements flagged in the ERRORS command (*q.v.*) to optimize the large amplitude behavior of the lattice. Here "optimize" means "make the large amplitude interpolated transfer map be as closely equal to the small amplitude transfer map as possible." The "large" amplitudes at which the interpolated map is to be evaluated are enterred in the same format as the TYP values in the ANALYSIS command. This command is implemented only in the C version, ctpot.

READFILE

READFILE, { FAST, SLOW, COMPACT, SDSFILE }

READFILE reads the thin lens machine information from unit 7. Except for a TITLE line, READFILE is normally the first line of the input lattice file unless a special version is required. It should be followed immediately by an ANALYSIS line, in order for teapot to regenerate various quantities that are not preserved in the "`fort.7`" file. (Some quantities are recalculated only if the TAPE directive is included.) These steps restore teapot to the same state it is in after processing an input lattice in standard format and MAKETHIN. The purpose of this command is to make it possible to input lattices tuned by *TEAPOT* or other programs, for instance orbit correction programs. The option FAST(default), SLOW or COMPACT refers to reading unformatted (FAST), formatted (SLOW) or formatted short (without the atw, btw etc) files (COMPACT) respectively. As explained under WRITEFILE, the SLOW or COMPACT option should be avoided if possible. Use of SDSFILE is described under WRITEFILE.

RUN, see TRACK and DUMPBEAM

Effective 10 July 1993, a RUN instruction with an XSEPTUM element present, or one that uses GAUSSIAN or BEAMTRAK tracking, should be followed by

a DUMPBEAM instructruction. This change was necessitated by a change in default to keep particles, rather than dumping them, so that tracking the same particles can continue lattice parameters have been changed. This is required for modeling resonant extraction and other dynamic processes.

TRACK, TRACKCLO

TRACK[CLO],[{SINGLE,GAUSSIAN,BEAMTRAK}], {ENERGY, PC}=<value>,
[PARTICLE = {PROTON,ELECTRON}, NUMPART = <value>,
SEED = <value>, SIGMACUT = <value>, APERTURE = <value>,
EIGENAMP, PRINT1ST, USER1 = <value>, USER2 = <value>,
USER3 = <value>, NSPRSSMX = <value>,]

START,[X = <value>, PX = <value>, Y = <value>, PY = <value>,
DP = <value>] , [EPSX = <value>, EPSY = <value>, SIGDELTA
= <value>, PH = <value>], DL = <value>

RUN, TURNS = <value>

The track command tracks up to 1024 particles. The default particle type is proton, and the energy is given in GeV. START starts a particle with the given initial conditions. TRACK interprets START command variables as being "absolute" displacements from the ideal orbit, and it prints out corresponding absolute values. TRACKCLO interprets START command variables as being "relative" displacements from the actual closed orbit (which was calculated by the most recent ANALYSIS call, or input with the CLOSEDORBIT command), and it prints out corresponding relative values.

The default TYPE of tracking is for SINGLE particles. In the case of GAUSSIAN tracking, the number of particles NUMPART, random number generator SEED and the cutoff (SIGMACUT) for the gaussian distribution should be specified. The default for NUMPART is 100, for SEED is 1 and for SIGMACUT is 4. For particle distributions other than Gaussian, to obtain the same beam-type processing (beam centroid etc.) and the same tracking sequence as applies for GAUSSIAN tracking, specify BEAMTRAK and enter starting coordinates

one-by-one using one START instruction per particle. An instance where this might be found useful is modeling particle extraction, with only special particles, perhaps having large amplitudes, being launched. To do this, element type XSEPTUM can be used. Attributes of XSEPTUM are defined under SPECIAL *TEAPOT* ELEMENTS. If there is any XSEPTUM present in the lattice it is assumed that BEAMTRAK=.TRUE. tracking is desired and EXTRACT=.TRUE. turn-by-turn output files written. As explained with the XSEPTUM description under Special Teapot Elements, the command CONTINUE can be used alternately to contunue tracking the same particles as machine elements are varied in small steps.

The coordinate system is defined in the *MAD* manual. The tracking output contains pure horizontal and vertical displacements as default. For coupled machines, if transformed-to-eigen-coordinates are required (so that linear coupling does not contribute to "smear" for example) you should include an "EIGENAMP" directive in "TRACK" or "TRACKCLO". PX and PY are the horizontal and vertical momenta divided by the reference momentum. DP is the momentum deviation from the reference momentum divided by the reference momentum. DL is the (negative) path length difference.

One START command is issued for each particle to be tracked. If a particle leaves a circle of radius equal to APERTURE, or receives a kick which produces a transverse momentum greater than the total momentum during tracking, it is not tracked further. The parameter APERTURE describes the radius $[m]$ of the physical beam pipe and/or windows. The default value of APERTURE is $1m$. The coordinates of the particles at each turn are written to unit 8 (and unit 18, 28, 38, and 48 in some cases) (see Appendix B). Note that in the GAUSSIAN and BEAMTRAK tracking cases, these files contain the average, or centroid, of all the particles coordinates instead the coordinates of all the particles in the distribution. For GAUSSIAN tracking two START commands are necessary, the first specifying the conditions at the mean of the distribution and the second specifying the rms beam dimensions in the form of x (EPSX) and y (EPSY)

emittances (ϵ), and the parameter P_h (PH). The parameters are related by the following equation:

$$\epsilon_i \equiv \epsilon_i(\text{invariant})/\gamma = -2\pi \ln(1 - P_h)\sigma_i^2/\beta_i$$

These can be made to represent σ_i^2/β_i by choosing PH=0.14714. The energy spread (SIGDELTA) is defined as σ_ϵ/E_0.

To get first turn track output information at every element you should include a "PRINT1ST" directive in "TRACK" or "TRACKCLO"; the output will be called "firstturn.trk". Note that this is disabled for gaussian tracking.

If there is any RFCAVITY present in the lattice it is assumed that the longitudinal position and momentum are to be tracked and output, turn-by-turn. Otherwise only the four transverse components are tracked and output.

As an option, parameters USER1, USER2, USER3 can be used to introduce definable parameters for use by special user-generated tracking code. NSPRSSMX can be used to suppress printout of intermediate turns in long tracking runs. Turns are printed out from the initial turn to *endofbeg = (nturns - nsprssmx)/2* and from *begofend = endofbeg + nsprssmx* to the end. By default no printout is suppressed.

TRACKCLO, see TRACK

TUNE

TUNE, MUX = <value>, MUY = <value>, K1F = <parameter name>, K1D = <parameter name>

The TUNE command uses the matrix representation of the thin lens machine, in which all elements are replaced by one thin element except quadrupoles of TYPE=IR, which are replaced by four thin quadrupoles. The tunes are fit by varying the parameters K1X and K1Y. This fitting compensates for the change in tunes in going from the thick lens to the thin lens machine. The <parameter name> can be a global parameter or an element parameter, for example, qf[k1].

TUNTHOLD, see DECOUPLE

TUNETHIN, see DECOUPLE

TWISS

TWISS, [PRINT = {BEAMLINE, ELEMENTS}], [TAPE]

Twiss analysis using the matrix representation. Since all elements, including dipoles are treated as thin, TWISS will agree with other matrix codes only if the elements are adequately sub-divided. This typically requires special effort only for small machines with tunes significantly less than 10. In any case, no subsequent *TEAPOT* operations rely on TWISS and, its further use beyond confirming that the lattice file is more or less as expected is discouraged. Nevertheless, a TWISS command should normally be included before MAKETHIN to avoid certain (probably harmless) apparent discrepancies in the output printout. If PRINT is omitted, the results of the analysis are only printed at the end of the machine. If PRINT = BEAMLINE is chosen, the Twiss parameters are printed at the beginning of all sublines. PRINT = ELEMENTS causes printing of Twiss parameters at each element. If TAPE is specified, the Twiss parameters are written to unit 3 in a format similar to the *MAD* tape3 format (see Appendix C).

USE

USE <machine>

Selects machine for subsequent operations.

WRITEFILE

WRITEFILE, { FAST, SLOW, COMPACT, SDSFILE }

WRITEFILE writes a file to unit 7 which describes the thin lens machine (see Appendix A). The options are:

- FAST, the default, is unformatted.

- SLOW, ascii, formatted, This file can be edited, but there is some loss of precision due to truncation of the output format. The main value of SLOW (or COMPACT) files is to acquire diagnostic information for checking the calculations, or to change parameter values before subsequent processing.
- COMPACT, formatted, without the atw, btw etc. being printed out for nonmagnetic elements; otherwise like SLOW.
- SDSFILE, a "Self-Descriptive Standard" file consistent with ISTK (Integrated Scientific ToolKit.) This can be browsed and edited using the ISTK tool called "sid". At this time only sparc executables are included in the software release, but they can be acquired for numerous other computer architectures. If "sid" is used, it should be customized using the ".sidopt" file.

If the purpose of the WRITEFILE is to generate a file that will later be used, unedited, for input using READFILE, then FAST should be used. If both diagnostic printout and a file for later input are required, they should be generated in two runs. For example the first run ends WRITEFILE[,FAST], its output is saved, then the second consists only of

READFILE[,FAST]
ANALYSIS
WRITEFILE, SLOW or COMPACT

Probably the only instance in which the sequence WRITEFILE, SLOW then READFILE, SLOW is really needed is to permit hand editing of a "`fort.7`" file. Since this file is "flat", sequentially numbered (rather than named) elements can be located and modified. This could also be accomplished by breaking out and naming the element in question in the standard input language file, but that has disadvantages: it complicates the hierarchy of the input file and it may necessitate repeating lengthy calculations. If the computer architecture being used is such that an sdsfile can be edited using "sid", Sds Interactive Dispaly, then WRITEFILE, SDSFILE, followed by "sid", followed by READFILE, SDSFILE

is the procedure of choice.

VERSION

This command lets the user select a particuler version and must be entered in the first line of the input lattice file.

The command usage : " VERSION, 2.05 " selects tpot version 2.05. If the version command is omitted the default (2.1, at least in early 1993) is used. Users with collimators in their input lattice files, for example, have to invoke the version command and select 4.0, since the standard version does not allow for collimators.

VSTEER, see HSTEER

ZWRITE

This command causes a 'flat' lattice description file called 'zfile' to be written. This file can be used to submit the lattice, including any errors that have been included, and any compensation that has been accomplished, to be transferred to a parallel processing computer, such as a Cray or a hypercube for subsequent processing. The output zfile includes sufficient headings to be self explanatory. Further detail on content and formatting can be inferred from the file zwrite.f or zwrite.c.

APPENDIX A

FORMAT OF UNIT 7, THE THIN LENS MACHINE DESCRIPTION

Formatted Version: fileform = 2.1

line 1	file format—format: f8.2
line 2	version, date, time, jobname, seed, 'random' or 'ordered'
	ntot+1, nelem
	format: 4a8,i8,a8,2i8
line 3	title—format: a80
line 4	'initial '—format: a8
lines 5,6	x0, y0, z0, $\sum$l, theta0, phi0, psi0
	format: 1p4e16.9, 1p3e16.9
lines 7,8	betaxim0, betayim0, alphaxim0, alphayim0, xnuideal0, ynuideal0
	format: 1p4e16.9, 1p2e16.9
line 9	keyword, elname, eltype, nmax (=max nonzero pole order)
	format: a8, a8, a4, i4
lines 10-14	(bn, an, n=0,9), b01, a01, thklen
	format: 4(5e16.9), 3e16.9,
line 15	Δx, Δy—format: 1p2e16.9
lines 16,17	x, y, z, $\sum$l, theta, phi, psi, thklen
	format: 1p4e16.9, 1p4e16.9
lines 18,19	betaxim, betayim, alphaxim, alphayim, xnuideal, ynuideal,
	format: 1p4e16.9, 1p2e16.9
	(above 4 lines repeated for all thin elements. drifts are implicit.)
line n	'endmach '—format: a8
lines n+1,n+2	xf, yf, zf, $\sum$l, thetaf, phif, psif
	format:1p4e16.9, 1p3e16.9
line n+3	orbit—format:1p4e16.9

FORMAT OF UNIT 7, Unformatted Version, Fileform > 2.0

item 1	file format
item 2,3,4,5,6,7,8	version, date, time, jobname, seed, 'random' or 'ordered', nelem
item 9,10,11	totlen, totang, ntot
item 12	title
item 13,14,15	elkeyw, elname, eltype
item 16	nmax (=max nonzero pole order)
item 17,18	bn[0-9], an[0-9]
item 19,20	b01, a01
item 21,22	Δx, Δy
item 23,24	X, Y, Z
item 25	suml
item 26,27,28	theta, phi, psi
item 29,30	ideal betax, ideal betay
item 31,32	ideal alphax, ideal alphay
item 32,33	mux/2π, muy/2π
item 34	orbit(1-4), (only at beginning of lattice), fileform > 2.01
item 35	thklen, fileform > 2.02

APPENDIX B

FORMAT OF THE TRACKING OUTPUT FILES, UNIT 8, 18, 28, 38, 48

1. Single Particle Tracking: (output on unit 8 only)

line 1 version, 'tracking', date, time, seed
format: 4(a8,2x), i8
line 2 title
format: a80
The rest of the file is free format.
line 3 nparts, nturns, betax, betay, alphax, alphay, Qx, Qy

Repeat for each particle: {

line 1 0 xi pxi yi pyi delta
(xi pxi yi pyi with respect to the last calculated closed orbit)
lines 2+ nturn xi pxi yi pyi (again wrt the closed orbit)
for as many turns as the particle survives
line n -1 0.0 0.0 0.0 0.0
after last turn if particle does not survive nturns turns
}

2. Gaussian Distribution Tracking:

line 1	version, 'tracking', date, time, seed
	format: 4(a8,2x), i8
line 2	title
	format: a80
	The rest of the file is free format.
line 3	nparts (=1) , nturns, betax, betay, alphax, alphay, Qx, Qy
[	for unit 8:
line 4	0 x1 x2 y1 y2 delta
	(x1 x2 y1 y2 are the coordinates at monitors x1 and x2
	with respect to the last calculated closed orbit)
lines 5+	nturn x1 x2 y1 y2 (again wrt the closed orbit)]
[	for unit 18:
line 4	0 x px y py nsurv delta
	(x px y py with respect to the last calculated closed orbit)
lines 5+	nturn x px y py nsurv (again wrt the closed orbit)
	nsurv number of surviving particles at nturn]

Tracking output is written to unit 8 on each passage of the lattice starting point. For BEAMTRAK tracking, only centroid values are printed. The same printouts are available at up to four lattice locations, marked by TYPE=X1,X2,X3, or X4; output goes to units 18,28,38 or 48 respectively. Eigencoordinates are not available under BEAMTRAK tracking irrespective of the EIGENAMP directive. Longitudinal coordinates are output if and only if there is an RFCAVITY in the lattice. For BEAMTRAK tracking, for historical reasons only, the longitudinal

output to unit 8 is suppressed, but output goes to units 18,28,38, or 48 as requested. To obtain output at a collimator one must introduce a marker to carry the TYPE=X1,X2,X3, or X4. Otherwise doubled output results because the collimator is replaced by two elements internally. A nuisance resulting from the tracking output feature is that you get fort.18,28,38,48 files generated whether or not you ask for them.

APPENDIX C

FORMAT OF UNIT 3, MACHINE PARAMETERS FILE
(SEE THE *MAD* MANUAL)
RESULTS OF THE TWISS COMMAND (no errors)

line 1	file format
	format: f8.2
line 2	version, 'twiss ', date, time, jobname, seed, nonsense, npos
	(one more than the number of elements)
	format: 5a8, i8, l8, i8
line 3	title
	format: a80
line 4	8x, 'initial ', 4x, 0.0, 0.0, 0.0, 0.0
	format: a8, a8, a4, f12.6, 3e16.9
line 5	zeros
	format: 5e16.9
lines 6-8	alphax, betax, mux/2π, nonsense, nonsense
	alphay, betay, muy/2π, nonsense, nonsense
	nonsense, nonsense, nonsense, nonsense, $\sum$l
	format: 5e16.9, 5e16.9, 5e16.9

the elements follow in order, multipoles only (no drifts) element data according to *MAD* manual except length of all elements given as 0.5 meters due to a quirk of the graphics program used to plot the Twiss functions.

Repeat for all elements: {

line 9	keyword, elname, eltype, length=0.5, eldata
line 10	more eldata
	format: 2a8, a4, f12.6, 3e16.9/5e16.9
lines 11-13	alphax, betax, mux/2π, nonsense, nonsense
	alphay, betay, muy/2π, nonsense, nonsense
	nonsense, nonsense, nonsense, nonsense, $\sum$l
	format: 5e16.9, 5e16.9, 5e16.9

}

	final record is
line n	nonsense, nonsense, nonsense
line n+1	cos(mux), Qx, nonsense, βx max, nonsense
line n+2	cos(muy), Qy, nonsense, βy max, nonsense
	format: 3e16.9/5e16.9/5e16.9

APPENDIX D

FORMAT OF UNIT N, MACHINE PARAMETERS FILE
(SEE THE *MAD* MANUAL)
RESULTS OF THE ANALYSIS COMMAND (errors included)

line 1	file format
line 2	version, 'twiss ', date, time, jobname, seed, nonsense, npos
	(one more than the number of elements)
	format: 5a8, i8, l8, i8
line 3	title
	format: a80
line 4	8x, 'initial ', 4x, 0.0, 0.0, 0.0, 0.0
	format: a8, a8, a4, f12.6, 3e16.9
line 5	zeros
	format: 5e16.9
lines 6-8	alphax, betax, mux/2π, etax, etax$'$
	alphay, betay, muy/2π, etay, etay$'$
	(closed orbit):x0, px0, y0, py0, $\sum$l
	format: 5e16.9, 5e16.9, 5e16.9

the elements follow in order, multipoles only (no drifts) element data according to *MAD* manual except length of all elements given as 0.5 meters due to a quirk of the graphics program used to plot the Twiss functions.

Repeat for all elements: {

line 9 keyword, elname, eltype, 0.0, alfaa, betaa, thetaa
line 10 badlocaa, alfdd, betdd, thetdd, badlocdd
line 11 alphaxim, betaxim, xnuideal, etam(1-2,ielem) [etax, etaxp]
line 12 alphayim, betayim, ynuideal, etam(3-4,ielem) [etay, etayp]
line 13 orbitm(1-4,ielem) [cloorbx, cloorbxp, cloorby, cloorbyp], suml
}

final record is
line n nonsense, nonsense, nonsense
line n+1 cos(mux), Qx, nonsense, βx max, etaxmax
line n+2 cos(muy), Qy, nonsense, βy max, etaymax
format: 3e16.9/5e16.9/5e16.9

APPENDIX E

FORMAT OF UNITS 50, 51, 60 and 61, CPLTRK OUTPUT FILES

RESULTS OF THE CPLTRK COMMAND

These files can be displayed using xgraph

UNIT 50

line 1	Comment ('TitleText: X at Y BPM interpolated from adjacent X BPMs')
line 2	Comment ('Markers: 1')
line 3	blank
line 4	Comment ('"interp"')
lines 5 - N	detector number, interpolated X at Y BPM for tracked particle with initial XKI
line N+1	blank
line N+2	Comment ('"actual"')
lines N+3 - 2N+3	detector number, actual X at Y BPM

UNIT 51

line 1	Comment ('TitleText: Y at X BPM interpolated from adjacent Y BPMs')
line 2	Comment ('Markers: 1')
line 3	blank
line 4	Comment ('"interp"')
lines 5 - N	detector number, interpolated Y at X BPM for tracked particle with initial XKI
line N+1	blank
line N+2	Comment ('"actual"')
lines N+3 - 2N+3	detector number, actual Y at X BPM

UNIT 60

line 1	Comment ('TitleText: X at Y BPM interpolated from adjacent X BPMs')
line 2	Comment ('Markers: 1')
line 3	blank
line 4	Comment ('"interp"')
lines 5 - N	detector number, interpolated X at Y BPM for tracked particle with initial YKI
line N+1	blank
line N+2	Comment ('"actual"')
lines N+3 - 2N+3	detector number, actual X at Y BPM

UNIT 61

line 1	Comment ('TitleText: Y at X BPM interpolated from adjacent Y BPMs')
line 2	Comment ('Markers: 1')
line 3	blank
line 4	Comment ('"interp"')
lines 5 - N	detector number, interpolated Y at X BPM for tracked particle with initial YKI
line N+1	blank
line N+2	Comment ('"actual"')
lines N+3 - 2N+3	detector number, actual Y at X BPM

APPENDIX F

DEFINITION AND USE OF TYPE CODES

Any lattice element can be assigned a type code consisting of up to four characters, by including TYPE='xxxx' in the element definition. Upper and lower case letters are not distinguished. The type codes recognized are:

'ir ': causes a quadrupole to be split into four thin quadrupoles.

'slnd': causes a sign reversal of the y component of the deflection in a solenoidal type element. See Appendix G.

' x1' and ' x2': mark the positions of beam position monitors at which particle coordinates are recorded during tracking.

'cncb' or 'cnce': mark the ends of sectors for which transfer matrices are to be calculated. The presence of even one such element in the ring causes the precalculation during every ANALYSIS operation of the transfer matrices between all successive pairs of x2, cncn, and cnce elements and the origin; (if there is an x1 element it is assumed to be at the origin.) During tracking, concatenated tracking begins at every cncb element and regular, exact, element-by-element tracking begins at every cnce element. A cncb element is implicitly assumed to be present at the origin if there is any cnc* element explicitly present. A flow chart typed as a comment in the code can be referred to for further information. The purpose of such concatenation is to increase the tracking speed (by big factors of like 10 or more depending on the lattice) through sectors known to include no 'dominant' nonlinear elements.

'cpla' and 'cpld': mark the locations of coupling adjustors and couplingdetectors respectively. Their presence cause precalculations to be performed during every ANALYSIS command so the command DECOUPRT can set the adjustors, based on information obtained at the detectors. See Appendix G.

'khka' and 'khkd': mark the locations of horizontal steering adjustors and detectors for orbit flattening just like the above described for decoupling. See Appendix H.

'kvka' and 'kvkd': mark the locations of vertical steering adjustors and detectors for orbit flattening just like the above described for decoupling. See Appendix H.

'nofo': causes the zeroing of the b_{01} multipole of an SBEND element (see SSC-52, p13) thereby permitting the representation of a rectangular bend element of very short length. It is not to be used to represent RBEND elements of finite length; the slanted ends should instead be represented by standard MULTIPOLE elements or, effective Oct.'94, by RBEND elements. (Note that the limit as a SBEND element is reduced to zero with the bend angle held fixed is not graceful, since in that limit the dipole focusing effect becomes large, as it would with an actual, infinite magnetic field, dipole.) A solenoid of type=nofo is a pure rotater, with no focusing action.

APPENDIX G

COUPLED BETATRON MOTION. FORMALISM

A general description of coupled betatron motion is given, first in a four component and then in a two component formalism. The state of coupling around the ring is represented by generalized Twiss parameters as well as the parameters of the two eigenplanes. A prescription for adjusting correction elements to achieve decoupled motion is given. These formulas have been incorporated in the accelerator modeling program **teapot** for which this, the first part of a two part report, is an appendix. It includes the prescription used for adjusting coupling correction elements to decouple the motion, based on diagnostic information obtained from beam position monitors.

1. Introduction.

Various elements present in an accelerator such as skew quadrupoles, misaligned quadrupoles and solenoids cause coupling between horizontal and vertical betatron motions of the particles. Such motion in the presence of arbitrary coupling will be analysed. The 4×4 linear matrix description of coupled betatron motion is, in principle, straightforward, but the absence of a simple pseudoharmonic description like that available for motion in one plane is a serious impediment. To recover this simplicity of description it is highly desireable that the lattice be approximately decoupled. For some of the more explicit and more practical formulas in this report, mainly appearing in the later sections, it is assumed that the coupling elements are weak enough that, perhaps after preliminary decoupling with two skew quadrupoles, horizontal and vertical motion can be treated separately, with the other motion having only a perturbative effect. This means that a particle launched with a purely horizontal deflection will remain within a few degrees of horizontal for a full turn. The purpose of this paper, as well as giving a general formulation of coupled motion, is to give prescriptions for achieving this decoupling. These prescriptions are used in the accelerator simulation program **teapot**.[1]

Formulas are given which are valid even with strong coupling; they can be used for analysing the initial situation and for an initial global decoupling, say with two skew quadrupoles, which is assumed to be sufficiently good to validate some of the approximations made later. Traditionally a 4×4 formalism has been used for the description of coupled motion and that path will be followed initially. But even the fully general results will be re-expressed in 2×2 form. For pedagogical purposes this paper will be self-contained so that derivations of well-known formulas make up, loosely speaking, the first third of the paper,[2,3,4] reformulation of results known in a different form the second third, and new results the last third.

The paper consists of the following sections.

1. Introduction.
2. General coupled motion.
3. Transformation to an eigenbasis.
4. Propagation of the generalized Twiss parameters around the ring.
5. Behaviour near the coupling resonance.
6. Solenoids.
7. Compensation of coupling.

Some notation to be used is

$$Q = \frac{\mu}{2\pi} = \text{tune} \equiv \text{frequency in } 1/\text{turns}$$

$$C = \cos\mu; \quad S = \sin\mu$$

There will be various subscripts on these: x and y for horizontal and vertical, A and D for eigenmotions close to horizontal and vertical respectively, and E for externally imposed. e.g. Q_E is the "tune" of an external shaker.

2. General Coupled Motion.

Initially we follow Courant and Snyder[2] closely, so as to have available the main general results.

Letting x and y describe horizontal and vertical displacements from the closed orbit, with p_x and p_y being the corresponding momenta, the transverse phase space displacement can be represented by a vector(transposed) $X^T = (x, p_x, y, p_y) \equiv (x_1, x_2, x_3, x_4)$. Using distance s along the closed orbit as the independent variable, the equations of motion can be written in Hamiltonian

form, with derivatives with respect to s being symbolized by primes, as

$$x' = \frac{\partial \mathcal{H}}{\partial p_x}; \qquad p_x' = -\frac{\partial \mathcal{H}}{\partial x} \tag{2.1}$$

$$y' = \frac{\partial \mathcal{H}}{\partial p_y}; \qquad p_y' = -\frac{\partial \mathcal{H}}{\partial y} \tag{2.2}$$

and the Hamiltonian $\mathcal{H}$, in linearized approximation and using matrix notation and the summation convention, is given by

$$\mathcal{H} = \frac{1}{2} X^T H X = \frac{1}{2} x_i H_{ij} x_j \tag{2.3}$$

where H_{ij} is a symmetric matrix.

Aside. A common source of confusion results when the term "phase space" is applied both to the x, x' space and the x, p_x space. The former is common during practical operations but the latter, which we will stick to, is better for preserving relativistic and Hamiltonian features in theoretical analysis. When the absolute value of the particle's momentum is preserved (e.g. because there is no r.f. acceleration) as will be assumed in this paper, then the ratio p_x/p is (at least for small angles) approximately equal to x' and for most purposes that identification can be assumed. This can be regarded as a choice of momentum units for purposes of interpreting the formulas in this paper.

Introducing a matrix S given in one and two dimensions by

$$S = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}; \qquad S = \begin{pmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{2.4}$$

Hamilton's equations take the form

$$X' = SHX \tag{2.5}$$

Observe that

$$S^T = -S \qquad \text{and} \qquad S^2 = -I \tag{2.6}$$

From any two solutions X_1 and X_2 of (2.5) an expression

$$X_2^T S X_1 = -x_2 p_{x1} + x_1 p_{x2} - y_2 p_{y1} + y_1 p_{y2} \tag{2.7}$$

can be constructed whose invariance follows from (2.5) and (2.6). Evolution of a vector X from s_0 to s is described by a transfer matrix M,

$$X(s) = M(s, s_0) X(s_0) \tag{2.8}$$

The invariance of $X_2^T S X_1$, when X_1 and X_2 evolve according to (2.8), yields a relation which the transfer matrix must satisfy

$$M^T S M = S \tag{2.9}$$

which is called the symplectic condition.

For analysing stability the eigenvalues of M are of paramount importance. That they come in reciprocal pairs can be seen from the following equations. Assuming that λ is an eigenvalue of M and hence also of M^T then

$$\det |M^T - \lambda I| = 0. \tag{2.10}$$

Multiplying by SM and using (2.9) yields

$$\det |S - \lambda S M| = 0.$$

Multiplying by S, using (2.6) and dividing by λ yields

$$\det |M - \frac{1}{\lambda} I| = 0 \tag{2.11}$$

which completes the proof.

The 4×4 matrix M can be partitioned in terms of 2×2 matrices

$$M = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \tag{2.12}$$

A useful matrix operation is "symplectic conjugation" defined by

$$\bar{A} = -SA^T S \tag{2.13}$$

For a 2×2 matrix

$$\bar{A} = \overline{\begin{pmatrix} a & b \\ c & d \end{pmatrix}} = \begin{pmatrix} d & -b \\ -c & a \end{pmatrix} = A^{\perp 1} \det |A| \tag{2.14}$$

The last expression is meaningful only if the determinant is non-zero. When applied to the 4×4 matrix M the result is

$$\bar{M} = \begin{pmatrix} \bar{A} & \bar{C} \\ \bar{B} & \bar{D} \end{pmatrix} \tag{2.15}$$

But, because M is symplectic, one gets using (2.9) and (2.6) that $SM^T SM = SS = -I$ and, as a result $\bar{M}M = I$ or

$$\bar{M} = M^{\perp 1} \tag{2.16}$$

When written out explicitly this gives relations among A,B,C, and D which follow from the symplectic condition:

$$\begin{aligned} A\bar{C} = -B\bar{D} \qquad &\equiv \qquad C\bar{A} = -D\bar{B} \\ \bar{A}B = -\bar{C}D \qquad &\equiv \qquad \bar{B}A = -\bar{D}C \\ A\bar{A} + B\bar{B} = 1 \qquad &\text{and} \qquad C\bar{C} + D\bar{D} = 1 \end{aligned} \tag{2.17}$$

These are not all independent. One useful result follows when one of the off-diagonal sub-matrices, say C, vanishes, since then the other, B, must also vanish.

Digression on determinants. Just in this section M will stand for an arbitrary, not necessarily symplectic, matrix. There is no simple expression for a determinant such as $\det|M|$ in terms of the sub-matrices A,B,C, and D unless they have some special property which, for our intended application, will be the case. In particular suppose that $A = aI$; i.e. A is proportional to the identity. One then finds that

$$\det\begin{vmatrix} A & B \\ C & D \end{vmatrix} = \det|AD - CB| \qquad \text{[for A proportional to I]} \tag{2.18}$$

To prove this one shows first that

$$\det\begin{vmatrix} A & 0 \\ C & D \end{vmatrix} = \det|A|\det|D|$$

and then

$$\det\begin{vmatrix} A & B \\ C & D \end{vmatrix} = \det|A|\det|D - CA^{\perp 1}B|$$

(both true for arbitrary A.) The latter follows by multiplying M on the left by a matrix whose determinant is one

$$\begin{pmatrix} 1 & 0 \\ -CA^{\perp 1} & 1 \end{pmatrix}$$

and the desired result (2.18) follows.

We can exploit the result of this digression and the earlier results by working on the matrix

$$M + \bar{M} = M + M^{\perp 1} = \begin{pmatrix} A + \bar{A} & B + \bar{C} \\ C + \bar{B} & D + \bar{D} \end{pmatrix} \tag{2.19}$$

This matrix exploits the fact that the eigenvalues of M come in reciprocal pairs so that the eigenvalues of $M + M^{\perp 1}$ are two doubly-degenerate values, each of

the form $\Lambda = \lambda + \lambda^{\perp 1}$. In the physically important case these sums will be real even though the individual eigenvalues are complex and this will permit us to complete the analysis without ever working with complex numbers.

The determining equation for the eigenvalues Λ is

$$\det \begin{vmatrix} (\mathrm{tr}A - \Lambda)I & B + \bar{C} \\ C + \bar{B} & (\mathrm{tr}D - \Lambda)I \end{vmatrix} = 0 \tag{2.20}$$

Result (2.18) is applicable and we get

$$\begin{aligned} 0 &= |M + M^{\perp 1} - \Lambda I| \\ &= |(\mathrm{tr}A - \Lambda)(\mathrm{tr}D - \Lambda)I - (C + \bar{B})(B + \bar{C})| \end{aligned} \tag{2.21}$$

This simplifies further since both terms are proportional to I. Letting

$$C + \bar{B} = \begin{pmatrix} c_{11} + b_{22} & c_{12} - b_{12} \\ c_{21} - b_{21} & c_{22} + b_{11} \end{pmatrix} \equiv \begin{pmatrix} e & f \\ g & h \end{pmatrix} \tag{2.22}$$

we have that

$$B + \bar{C} = \begin{pmatrix} h & -f \\ -g & e \end{pmatrix} \tag{2.23}$$

and

$$(C + \bar{B})(B + \bar{C}) = (B + \bar{C})(C + \bar{B}) = (eh - fg)I = \det|C + \bar{B}|I \tag{2.24}$$

as a result (2.21) yields

$$(\mathrm{tr}A - \Lambda)(\mathrm{tr}D - \Lambda) - \det|C + \bar{B}| = 0 \tag{2.25}$$

whose solutions are

$$\Lambda_{A,D} = (\mathrm{tr}A + \mathrm{tr}D)/2 \pm \sqrt{(\mathrm{tr}A - \mathrm{tr}D)^2/4 + \det|C + \bar{B}|} \tag{2.26}$$

where $A(D)$ goes with the $+(-)$ sign if $\mathrm{tr}A - \mathrm{tr}D$ is positive and vice versa. This choice assures, for weak coupling, that A will correspond to x and D will

correspond to y. In the physically important case the eigenvalues of M have modulus 1 so that there are real angles μ_A and μ_D satisfying

$$\Lambda_{A,D} = \lambda_{A,D} + 1/\lambda_{A,D} = \exp(i\mu_{A,D}) + \exp(-i\mu_{A,D}) = 2\cos\mu_{A,D} \tag{2.27}$$

In the special uncoupled case for which B and C vanish these angles degenerate into the horizontal and vertical phase advances μ_x and μ_y which satisfy

$$\Lambda_{A,D} = \mathrm{tr}A, D = 2\cos\mu_{x,y} = 2\cos\mu_{A,D} \tag{2.28}$$

From (2.26) and (2.27) follow the useful relation

$$(\cos\mu_A - \cos\mu_D)^2 = (\mathrm{tr}A - \mathrm{tr}D)^2/4 + \det|C + \bar{B}| \tag{2.29}$$

From these formulas it can be seen that the sign of the determinant $\det|C + \bar{B}|$ has a special importance; if it is negative and the traces of A and D are equal then μ_A and/or μ_D will be complex, which implies instability. This can potentially occur on "difference resonances" for which

$$Q_x - Q_y = \text{integer} \tag{2.30}$$

or on "sum resonances" for which

$$Q_x + Q_y = \text{integer} \tag{2.31}$$

It will be demonstrated in section 5 that the difference resonances are inherently stable and the sum resonances inherently unstable. Commonly accelerators (especially proton accelerators) are run close to a difference resonance (since areas bounded by nonlinear resonances are largest there.) This is only possible because the coupling resonance does not lead to instability though it can strongly influence the particle distributions. In what follows it will be assumed that operation

is not close to a sum resonance (2.31) and some extra attention will be paid to operation close to the difference resonance (2.30), especially in section 5 where $\det|C+\bar{B}|$ will be evaluated explicitly.

To determine the eigenvectors of $M+M^{\perp 1}$ it is useful to represent a displacement within the x phase space by $\chi^T=(x,p_x)$ and similarly $\xi^T=(y,p_y)$. For eigenvalue Λ it is easy to check that the vectors

$$X=\begin{pmatrix}\chi\\ \frac{C+\bar{B}}{\Lambda\perp \mathrm{tr}D}\chi\end{pmatrix};\qquad Y=\begin{pmatrix}\frac{B+\bar{C}}{\Lambda\perp \mathrm{tr}A}\xi\\ \xi\end{pmatrix} \tag{2.32}$$

satisfy the equations

$$(M+M^{\perp 1})X=\Lambda X;\qquad (M+M^{\perp 1})Y=\Lambda Y \tag{2.33}$$

for arbitrary χ or ξ. These are however not independent, as can be seen using (2.24)and (2.25), and the same vector can be represented either as X or Y. On the other hand, as mentioned above, in the case of weak coupling, the motion labelled A is close to x and D is close to y. It is natural then to pick Λ_A in defining X and Λ_D in defining Y. Toward this end we define 2×2 matrices R_A and R_D by

$$R_A=\frac{C+\bar{B}}{\Lambda_A-\mathrm{tr}D};\qquad R_D=\frac{B+\bar{C}}{\Lambda_D-\mathrm{tr}A} \tag{2.34}$$

in terms of which independent basis vectors can be written as

$$X=\begin{pmatrix}\chi\\ R_A\chi\end{pmatrix};\qquad Y=\begin{pmatrix}R_D\xi\\ \xi\end{pmatrix} \tag{2.35}$$

From (2.26) one can see that

$$\Lambda_A-\mathrm{tr}D=-(\Lambda_D-\mathrm{tr}A) \tag{2.36}$$

from which it follows that

$$\bar{R}_D=-R_A \tag{2.37}$$

The two components of χ can be chosen independently to give two independent

eigenvectors each appproximately horizontal and expressed in the form X and similarly for Y (near vertical). In the next section such a specific choice will be made.

If R_A were proportional to the identity then the vector X would be inclined relative to the horizontal by a small angle $\arctan[\sqrt{\det|C+\bar{B}|}/(\Lambda_A - \mathrm{tr}D)]$. But that is not normally the case and the eigenmotion is not restricted to a single plane. Rather the x and y motions resemble the electric field vectors in elliptically polarized light; this analogy will be developed further below, as will the geometry of the motion.

It is conventional[3,4] to define an angle ψ, which for weak coupling is loosely similar to that proposed in the previous paragraph, by

$$\tan 2\psi = \frac{2\sqrt{\det|C+\bar{B}|}}{\mathrm{tr}A - \mathrm{tr}D} \tag{2.38}$$

This definition is motivated by formula (2.29) which can be regarded as a kind of Pythagorean relation as shown in the figure.

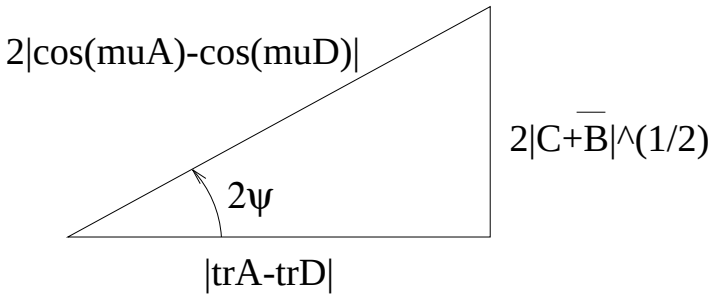

Figure 2: "Pythagorean triangle" of parameters for nearly equal tunes.

The angle ψ does not have a simple orientational interpretation. Shortly a different angle which specifies the orientation of an eigenplane will be introduced.

If the uncoupled tunes are brought close together (by adjusting the normal quads in the ring) the angle 2ψ increases and approaches $\pi/2$. By (2.29) or by the figure, the eigenfrequencies cannot become equal. Their minimum separation is given by

$$|Q_D - Q_A|_{min} = \frac{\sqrt{\det|C + \bar{B}|}}{\pi(\sin\mu_A + \sin\mu_D)} \tag{2.39}$$

A routine accelerator decoupling operation which depends only on having a position detector and spectrum analyser capable of measuring Q_A and Q_D, consists of empirical adjustment of regular and skew quads to minimize the tune separation. It is then assumed that the small change of regular quads to bring the tunes to their desired (normally not quite equal) values, re-introduces negligible coupling. This maneuver by no means assures that the eigenmotions are horizontal and vertical but, as we will see below, it does suppress resonant sloshing between horizontal and vertical motion over many turns.

3. Transformation To An Eigenbasis.

In order to define Twiss parameters in a coupled lattice it is necessary to perform a linear transformation from the x,y basis to an eigenvector basis. Though the eigenvalues are complex this transformation will be performed in this section without use of complex numbers.

In a two component space basis vectors can be expressed as

$$\hat{\chi}_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}; \qquad \hat{\chi}_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{3.1}$$

These can be used to define an x,y basis in the four component space.

$$\hat{\imath}^{(1)} = \begin{pmatrix} \hat{\chi}_1 \\ 0 \end{pmatrix}; \quad \hat{\imath}^{(2)} = \begin{pmatrix} \hat{\chi}_2 \\ 0 \end{pmatrix}; \quad \hat{\imath}^{(3)} = \begin{pmatrix} 0 \\ \hat{\chi}_1 \end{pmatrix}; \quad \hat{\imath}^{(4)} = \begin{pmatrix} 0 \\ \hat{\chi}_2 \end{pmatrix} \tag{3.2}$$

Similarly, from (2.35), a basis of eigenvectors is

$$\hat{I}^{(1)} = g\begin{pmatrix} \hat{\chi}1 \\ R_A\hat{\chi}1 \end{pmatrix}; \hat{I}^{(2)} = g\begin{pmatrix} \hat{\chi}2 \\ R_A\hat{\chi}2 \end{pmatrix}; \hat{I}^{(3)} = g\begin{pmatrix} R_D\hat{\chi}1 \\ \hat{\chi}1 \end{pmatrix}; \hat{I}^{(4)} = g\begin{pmatrix} R_D\hat{\chi}2 \\ \hat{\chi}2 \end{pmatrix} \tag{3.3}$$

where g is a numerical factor yet to be determined. These bases are related by a linear transformation

$$\hat{I}^{(k)} = G_{ki}\hat{\imath}^{(i)} \tag{3.4}$$

where summation here and in the sequel is assumed. A general vector can be expressed in terms of either basis, yielding the equality

$$\begin{aligned} x_i\hat{\imath}^{(i)} &= X_k\hat{I}^{(k)} \\ &= X_kG_{ki}\hat{\imath}^{(i)} \end{aligned} \tag{3.5}$$

and from this the coordinates are related, in component and in matrix notation by

$$x_i = X_kG_{ki}; \qquad x = G^TX \tag{3.6}$$

By substituting from (3.3)into (3.4)one obtains

$$G^T = g\begin{pmatrix} I & R_D \\ R_A & I \end{pmatrix} \tag{3.7}$$

Furthermore, using (2.24) one can check that the inverse of G^T is

$$(G^T)^{\perp 1} = \frac{\Lambda_D - \mathrm{tr}A}{g(\Lambda_D - \Lambda_A)}\begin{pmatrix} I & -R_D \\ -R_A & I \end{pmatrix} \tag{3.8}$$

From (3.7) and (3.8) it is clear that the choice

$$g = \sqrt{\frac{|\Lambda_D - \mathrm{tr}A|}{|\Lambda_D - \Lambda_A|}} \tag{3.9}$$

yields the relations

$$|G^T| = |(G^T)^{\perp 1}| = 1 \tag{3.10}$$

as well as

$$\bar{G}^T = (G^T)^{\perp 1} \tag{3.11}$$

which shows that G is symplectic, a result which will be essential in the next section.

In the x, y basis the one turn map is given by (2.8)

$$x^+ = Mx \tag{3.12}$$

where x^+ is the displacement after one turn. Substituting from (3.6) one gets

$$G^T X^+ = M G^T X \tag{3.13}$$

which means that the transfer matrix in the transformed basis is

$$\begin{aligned} \underline{M} = (G^T)^{\perp 1} M G^T &= g^2 \begin{pmatrix} I & -R_D \\ -R_A & I \end{pmatrix} \begin{pmatrix} A & B \\ C & D \end{pmatrix} \begin{pmatrix} I & R_D \\ R_A & I \end{pmatrix} \\ &= \begin{pmatrix} \underline{A} & 0 \\ 0 & \underline{D} \end{pmatrix} \end{aligned} \tag{3.14}$$

where

$$\begin{aligned} \underline{A} &= g^2(A + BR_A - R_D C - R_D D R_A) \\ \underline{D} &= g^2(-R_A A R_D - R_A B + C R_D + D) \end{aligned} \tag{3.15}$$

Exercise. Demonstrate, using relations (2.17) ,that the off-diagonal elements of (3.14) do, in fact, vanish. (I have only succeeded in showing it indirectly.)

From $\underline{A}$ and $\underline{D}$ the Twiss parameters in the eigenbasis can be extracted. The determinants $\det|\underline{A}|$ and $\det|\underline{D}|$ must both be unity since they are equal to the

product of eigenvalues, which is one. As a result $\underline{A}$, for example, can be written in "Twiss form"

$$\begin{pmatrix} \underline{A}_{11} & \underline{A}_{12} \\ \underline{A}_{21} & \underline{A}_{22} \end{pmatrix} = \begin{pmatrix} \cos\mu_A + \alpha_A \sin\mu_A & \beta_A \sin\mu_A \\ -\gamma_A \sin\mu_A & \cos\mu_A - \alpha_A \sin\mu_A \end{pmatrix} \tag{3.16}$$

where

$$\mu_A = \arccos(\mathrm{tr}\underline{A}/2) \tag{3.17}$$

It is assumed here that any ambiguity has already been resolved in (2.26). The Twiss parameters are obtained from element-by-element comparison in (3.16) and they are

$$\beta_A = \underline{A}_{12}/\sin\mu_A; \quad \gamma_A = -\underline{A}_{21}/\sin\mu_A; \quad \alpha_A = (\underline{A}_{11} - \underline{A}_{22})/(2\sin\mu_A) \tag{3.18}$$

and similarly for $\underline{D}$.

A given vector x will, in general, have non-vanishing components in both of the eigenbases. The corresponding invariants can be evaluated using the inverse of (3.6) to obtain X, followed by substitution into the relations which define invariants in the eigenbases,

$$\begin{aligned} \epsilon_A &= \gamma_A X_1^2 + 2\alpha_A X_1 X_2 + \beta_A X_2^2 \\ \epsilon_D &= \gamma_D X_3^2 + 2\alpha_D X_3 X_4 + \beta_D X_4^2 \end{aligned} \tag{3.19}$$

Finally we wish to characterize each of the eigenbases by a single orientation. It has already been observed that such motion is not restricted to a single plane, but rather the phase point moves on an ellipse in x, y space. It is reasonable to characterize the orientation of the $\underline{A}$-eigenbasis, by the angular deviation, θ_A, of the major principle axis of the ellipse, away from the x-axis, and similarly for $\underline{D}$.

Exercise. With the eigenmotion given in the "pseudoharmonic" form

$$\begin{aligned} X_{A1} &= \cos\psi_A \\ X_{A2} &= (\sin\psi_A - \alpha_A \cos\psi_A)/\beta_A \end{aligned} \tag{3.20}$$

and with ψ_A regarded as a free parameter, substitute into (3.6) to express the motion in the form

$$\begin{aligned} x &= g\cos\psi_A \\ y &= ge_A \cos(\psi_A + \epsilon_A) \end{aligned} \tag{3.21}$$

where

$$\begin{aligned} e_A^2 &= [R_{A11} - (\alpha_A/\beta_A)R_{A12}]^2 + (R_{A12}/\beta_A)^2 \\ \epsilon_A &= -\arctan\frac{R_{A12}/\beta_A}{R_{A11} - (\alpha_A/\beta_A)R_{A12}} \end{aligned} \tag{3.22}$$

In section 7 equations like (3.21) will be interpreted as relationships between measureable quantities; they will serve a diagnostic purpose. Next find the equation of the ellipse in question and show that its angle of orientation is given by

$$\tan 2\theta_A = -\frac{2[R_{A11} - (\alpha_A/\beta_A)R_{A12}]}{1 - [R_{A11} - (\alpha_A/\beta_A)R_{A12}]^2 - (R_{A12}/\beta_A)^2} \tag{3.23}$$

The orientation of the other eigenaxis can be found similarly.

In general, the two axes are not orthogonal. Normally, since ideal behaviour would have the eigenaxes exactly horizontal and vertical, the deviations of these angles from zero can be regarded as a measure of the seriousness of the coupling. On the other hand, tilt of the eigenplanes may be considered inoccuous and a better measure might be the area of the eigenellipse which is equal to $\pi g^2|R_{A12}|/\beta_A$. Notice, using (2.14) and (2.37), that $R_{D12} = R_{A12}$ which means that the areas of the two eigenellipses are equal except for a coupling-independent (for weak coupling) factor.

4. Propagation of the Generalized Twiss Parameters Around the Ring.

In an uncoupled machine the "pseudoharmonic" description is a representation in which the evolution of either of the transverse coordinates is represented by the increase of a single angle, "the betatron phase." We now obtain a similar representation, valid even in the presence of coupling.

Consider two points in the ring labelled (0) and (1) and located at longitudinal positions $s^{(0)}$ and $s^{(1)}$. Propagation through the region is represented by a transfer matrix $M^{(01)}$ in the x,y basis.

$$x^{(1)} = M^{(01)} x^{(0)} \tag{4.1}$$

where, for the present discussion $M^{(01)}$, is assumed to be known. The transformation (4.1) in general couples the two transverse coordinates.

If it is assumed that $M^{(0)}$, the once-around transfer matrix at (0), is also known, then the once-around transfer matrix at (1) is given by

$$M^{(1)} = M^{(01)} M^{(0)} \overline{M^{(01)}} \tag{4.2}$$

In writing this we have exploited the fact that $M(01)$ is symplectic so that formula (2.16) can be used to obtain $(M^{(01)})^{\perp 1}$ as being equal to $\overline{M^{(01)}}$ which can in turn be obtained using (2.13). This circumvents the need for numerical evaluation of the matrix inverse. With $M^{(1)}$ known the Twiss parameters at (1) can be determined as in the previous section.

What remains is to find the generalized betatron phase advances in going from (0) to (1). The coordinates $X^{(0)}$ and $X^{(1)}$ in the eigenbases at (0) and (1) can be obtained from the inverse of (3.6) . It follows that the transfer matrix

from (0) to (1) in the eigenbasis is given by

$$\underline{M}^{(01)} = (G^{(1)T})^{\perp 1} M^{(01)} G^{(0)T} \tag{4.3}$$

Since no propagation transformation such as this could mix the components corresponding to two different eigenvalues, this transformation is block-diagonal.

$$\underline{M}^{(01)} = \begin{pmatrix} \underline{A}^{(01)} & 0 \\ 0 & \underline{D}^{(01)} \end{pmatrix} \tag{4.4}$$

Furthermore it was shown in (3.11) that the factors in (4.3) are individually symplectic so that (4.4) consists of two 2×2 transformations of the form (3.16). Unfortunately the phase angle μ in such a representation is not the exact analog of the betatron phase advance, except in the special case that the Twiss parameters at (0) and (1) are the same.

The true betatron phase advance should behave additively as successive sections are concatenated. By analogy with the 2×2 uncoupled formalism[2] the transfer sub-matrix can be parameterized as

$$\begin{pmatrix} \sqrt{\beta_A^{(1)}/\beta_A^{(0)}}(\cos\psi_A^{(01)} + \alpha_A^{(0)} \sin\psi_A^{(01)}) & \sqrt{\beta_A^{(0)}\beta_A^{(1)}} \sin\psi_A^{(01)} \\ & \sqrt{\beta_A^{(1)}/\beta_A^{(0)}}(\cos\psi_A^{(01)} - \alpha_A^{(1)} \sin\psi_A^{(01)}) \end{pmatrix} \tag{4.5}$$

where the missing element can be filled in to make the determinant equal to one. Direct comparison yields

$$\psi_A^{(01)} = \arcsin \frac{\underline{A}_{12}^{(01)}}{\sqrt{\beta_A^{(0)}\beta_A^{(1)}}} \tag{4.6}$$

which completes the determination of the generalized Twiss parameters for the near horizontal eigenmotion. The $\underline{D}$ parameters can be extracted similarly.

5. Behaviour Near The Coupling Resonance.

In section 2 it has been shown that the expression $\det|C+\bar{B}|$ is especially important near the coupling resonance and here we will evaluate it in terms of the strengths of an arbitrary number of thin skew quadrupoles in an otherwise decoupled lattice. To simplify this calculation we use a well-known transformation of uncoupled motion in which propagation from point to point is represented by pure rotation in phase space with the transfer matrix taking the form

$$\begin{pmatrix} \cos\Delta\psi & \sin\Delta\psi \\ -\sin\Delta\psi & \cos\Delta\psi \end{pmatrix} \tag{5.1}$$

where $\Delta\psi$ is the appropriate x or y betatron phase advance in going from the first to the second point. To achieve this one performs the following transformation from $X^T \equiv (x, p_x, y, p_y)$ to $\tilde{X}^T \equiv (\tilde{x}, \tilde{p}_x, \tilde{y}, \tilde{p}_y)$

$$\tilde{X} = \mathcal{B}X \tag{5.2}$$

where

$$\mathcal{B} = \begin{pmatrix} \mathcal{B}_x & 0 \\ 0 & \mathcal{B}_y \end{pmatrix} \tag{5.3}$$

$$\mathcal{B}_x = \begin{pmatrix} \beta_x^{\perp 1/2} & 0 \\ \alpha_x\beta_x^{\perp 1/2} & \beta_x^{1/2} \end{pmatrix} \tag{5.4}$$

$$\mathcal{B}_x^{\perp 1} = \begin{pmatrix} \beta_x^{1/2} & 0 \\ -\alpha_x\beta_x^{\perp 1/2} & \beta_x^{\perp 1/2} \end{pmatrix} \tag{5.5}$$

and similarly for y. In the new variables the x invariant emittance is given by $\epsilon_x = \tilde{x}^2 + \tilde{p}_x^2$ and similarly for y. Defining $\tilde{M}$, the once-around transfer matrix in

this representation, by

$$\tilde{M} = \begin{pmatrix} \tilde{A} & \tilde{B} \\ \tilde{C} & \tilde{D} \end{pmatrix} \tag{5.6}$$

one obtains

$$\begin{aligned} A =& \mathcal{B}_x^{\perp 1} \tilde{A} \mathcal{B}_x \\ B =& \mathcal{B}_x^{\perp 1} \tilde{B} \mathcal{B}_y \\ C =& \mathcal{B}_y^{\perp 1} \tilde{C} \mathcal{B}_x \\ D =& \mathcal{B}_y^{\perp 1} \tilde{D} \mathcal{B}_y \end{aligned} \tag{5.7}$$

This will be called the circular representation. In this representation it is natural to work with dimensionless skew quadrupole strengths given by

$$q = \sqrt{\beta_x \beta_y}/f \tag{5.8}$$

where f is the focal length of the rotated (by 45 degrees) quadrupole since the transfer matrix is given by

$$\begin{pmatrix} \mathcal{B}_x & 0 \\ 0 & \mathcal{B}_y \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1/f & 0 \\ 0 & 0 & 1 & 0 \\ 1/f & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mathcal{B}_x^{\perp 1} & 0 \\ 0 & \mathcal{B}_y^{\perp 1} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & q & 0 \\ 0 & 0 & 1 & 0 \\ q & 0 & 0 & 1 \end{pmatrix} \tag{5.9}$$

The calculation of the transfer function once around the lattice proceeds by evaluating expressions such as

$$M = \begin{pmatrix} C_x^{\bar{\imath}} & S_x^{\bar{\imath}} & 0 & 0 \\ -S_x^{\bar{\imath}} & C_x^{\bar{\imath}} & 0 & 0 \\ 0 & 0 & C_y^{\bar{\imath}} & S_y^{\bar{\imath}} \\ 0 & 0 & -S_y^{\bar{\imath}} & C_y^{\bar{\imath}} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & q_i & 0 \\ 0 & 0 & 1 & 0 \\ q_i & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} C_x^i & S_x^i & 0 & 0 \\ -S_x^i & C_x^i & 0 & 0 \\ 0 & 0 & C_y^i & S_y^i \\ 0 & 0 & -S_y^i & C_y^i \end{pmatrix} \tag{5.10}$$

where the notation

$$\begin{aligned} \mu_x^{\bar{\imath}} &= \mu_x - \mu_x^i \\ S_x^i &= \sin \mu_x^i \\ S_x^{\bar{\imath}} &= \sin \mu_x^{\bar{\imath}} \end{aligned} \tag{5.11}$$

and similarly for y and for cosines has been used. In this notation i refers to the trip from the origin to the i'th skew quadrupole and $\bar{\imath}$ refers to the trip through the rest of the lattice back to the origin. The complete expression for M has a product like (5.10) with a matrix for each skew element sandwiched between appropriate rotation matrices. We will, however, keep only terms quadratic in the q_i factors; that is we make a weak coupling assumption, to be quantified later. As an exercise the reader can assure his or herself that the expansion of $\det|C+\bar{B}|$ contains no terms linear in the q_i and that the only quadratic terms are included in (5.10). More accurately, one should say that all the skew quad matrices should be set to the identity except one and in that one the diagonal elements should be set to zero. Then one must sum over all skew quad locations.

With the notation of (2.12) and a certain amount of algebra one obtains

$$\tilde{B} = \begin{pmatrix} \sum q_i S_x^{\bar{\imath}} C_y^i & \sum q_i S_x^{\bar{\imath}} S_y^i \\ \sum q_i C_x^{\bar{\imath}} C_y^i & \sum q_i C_x^{\bar{\imath}} S_y^i \end{pmatrix} \tag{5.12}$$

$$\tilde{C} = \begin{pmatrix} \sum q_i S_y^{\bar{\imath}} C_x^i & \sum q_i S_y^{\bar{\imath}} S_x^i \\ \sum q_i C_y^{\bar{\imath}} C_x^i & \sum q_i C_y^{\bar{\imath}} S_x^i \end{pmatrix} \tag{5.13}$$

For the time being we can drop the tilde's since $\det|C+\bar{B}|$ is equal to $\det|\tilde{C}+\overline{\tilde{B}}|$. We get

$$\begin{aligned} \det|C+\bar{B}| = \sum_{i,j} q_i q_j [&(S_y^{\bar{\imath}} C_x^i + C_x^{\bar{\imath}} S_y^i)(C_y^{\bar{\jmath}} S_x^j + S_x^{\bar{\jmath}} C_y^j) \\ &- (S_y^{\bar{\imath}} S_x^i - S_x^{\bar{\imath}} S_y^i)(C_y^{\bar{\jmath}} C_x^j - C_x^{\bar{\jmath}} C_y^j)] \end{aligned} \tag{5.14}$$

This can be manipulated further to yield the result

$$\begin{aligned}\det|C+\bar{B}| &= \sum_{i,j} q_i q_j [\sin(\mu_y^j-\mu_y^i)\sin(\mu_x^j-\mu_x^i) \\ &\qquad + \sin(\mu_y+\mu_y^j-\mu_y^i)\sin(\mu_x+\mu_x^j-\mu_x^i)] \\ &= \sum_{i,j} q_i q_j [(1-C_yC_x)\sin(\mu_y^j-\mu_y^i)\sin(\mu_x^j-\mu_x^i) \\ &\qquad + S_xS_y\cos(\mu_y^j-\mu_y^i)\cos(\mu_x^j-\mu_x^i)]\end{aligned} \tag{5.15}$$

where $S_{x,y}$ and $C_{x,y}$ refer to propagation around the entire lattice. This expression can be separated into squared terms, q_i^2(all having the same sign), and crossed terms q_iq_j (of either sign).

$$\begin{aligned}&\det|C+\bar{B}| = S_xS_y\sum_i q_i^2 + 2\sum_{j>i} q_iq_j \\ &\times[(1-C_xC_y)\sin(\mu_y^j-\mu_y^i)\sin(\mu_x^j-\mu_x^i) + S_xS_y\cos(\mu_y^j-\mu_y^i)\cos(\mu_x^j-\mu_x^i)]\end{aligned} \tag{5.16}$$

This formula makes it plausible that the sign of $\det|C+\bar{B}|$ is the same as the sign of S_xS_y. That will now be proved.

A result such as that could only be true for sufficiently small values of the q_i and is only of interest near a resonance. Hence we set $C_x = C_y = C$ and $S_y = S_x$ and obtain

$$\det|C+\bar{B}| = S_xS_y[\sum_i q_i^2 + 2\sum_{j>i} q_iq_j\cos(\Delta_j-\Delta_i)] \tag{5.17}$$

where $\Delta_i = \mu_y^i - \mu_x^i$. (At a sum resonance one would use $S_y = -S_x$ and obtain a somewhat different expression.) We can define a skew quad "phasor" strength $\tilde{q}_i = q_i\exp(i\Delta_i)$ and then obtain

$$\det|C+\bar{B}| = S_xS_y(\sum_i \tilde{q}_i)\overline{(\sum_j \tilde{q}_j)} \tag{5.18}$$

The factor multiplying S_xS_y is inherently positive which completes the proof. We can now complete the discussion of sum and difference resonances begun in

section 2. On sum resonances the factor $S_x S_y$ is negative and hence so also is $\det|C + \bar{B}|$. As we saw then, this causes instability on sum resonances. On difference resonances $S_x S_y$ is positive and the motion is stable.

It can now be seen from (2.26) that, for the success of the experimental procedure of adjusting skew quads until the eigenfrequencies coincide, it is necessary and sufficient that $\sum_i \tilde{q}_i$ vanish. This can be accomplished by the empirical adjustment of any two skew quads in the ring, unless by chance they have the same "phase", which could perhaps occur because of the symmetry of their placement.

The formula analogous to (5.16) but valid near the sum resonance can be applied to another important issue which is to estimate the strength of the sum resonance caused by N random skew quads in the lattice. The crossed terms can be expected to average to zero, unless there is a "structure" effect causing the phases and strength's to be correlated. Hence we get

$$\det|C + \bar{B}| \simeq S_x S_y \sqrt{N} < q^2 > \tag{5.19}$$

which can be used to obtain the "stop-band" width using (2.26).

6. Solenoids

Ordinarily skew quadrupole are present in an accelerator only unintentionally unless they have been included to compensate for solenoids present for detectors of particle interactions. In this section we analyse such solenoids.

Most magnetic elements in accelerators have only field components B_x and B_y normal to the central trajectory, but a solenoid (length L) has mainly a longitudinal field B_z given by

$$\begin{aligned} B_z &= 0 \qquad\quad z < -L/2 \\ &= B_0 \quad -L/2 < z < L/2 \\ &= 0 \qquad\quad L/2 < z \end{aligned} \tag{6.1}$$

There will be an important end effect but initially we will calculate only the effect

of this longitudinal field.

Consider a particle incident on this magnet with momentum and velocity vectors given by

$$\vec{p} = \vec{p}_\perp + \vec{p}_\parallel \tag{6.2}$$

$$\vec{v} = \vec{v}_\perp + \vec{v}_\parallel = \vec{p}c^2/E \tag{6.3}$$

In a paraxial approximation $p \simeq p_\parallel$. It is conventional to express the solenoid strength by a factor K equal to $(2R_0)^{\perp 1}$ where R_0 is the cyclotron radius if the full momentum were transverse ($p_\perp = p$.)

$$K = \frac{1}{2R_0} = \frac{cB_0}{2pc/e} \tag{6.4}$$

In **transport** notation let the incident vector of the above particle be given by $(0, x', 0, 0)^T$ so that its transverse momentum is given by $p_\perp = x'p$ and it follows a spiral whose radius is $R_\perp = x'/(2K)$ with a transverse speed given by $x'pc^2/E$. The time spent in the solenoid is $L/v_\parallel \simeq LE/(c^2p)$. Labeling the angle through which the spiral turns by 2θ as shown in the figure, and combining the above formulas one obtains the result that

$$\theta = KL \tag{6.5}$$

The output coordinates are given by

$$\begin{aligned} x_{out} &= x'CS/K \\ y_{out} &= -x'S^2/K \end{aligned} \tag{6.6}$$

where $C = \cos KL$ and $S = \sin KL$. Similar calculations for the dependence on y' as well as calculation of the output values of x' and y' show that linearized

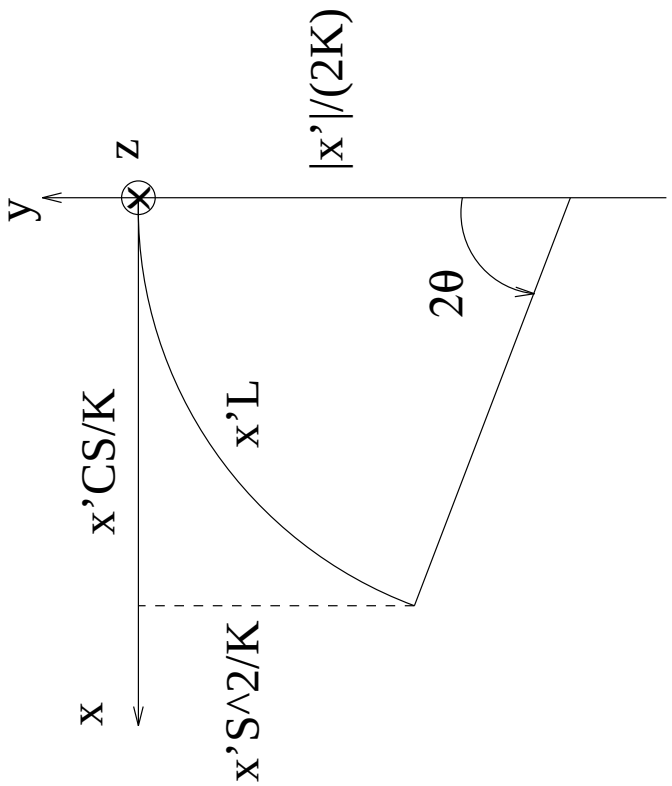

Figure 3: Solenoid trajectory viewed from downstream.

propagation through the solenoid can be represented by the mapping

$$\begin{pmatrix} x \\ x' \\ y \\ y' \end{pmatrix}_{out} = \begin{pmatrix} 1 & CS/K & 0 & S^2/K \\ 0 & C^2-S^2 & 0 & 2CS \\ 0 & -S^2/K & 1 & CS/K \\ 0 & -2CS & 0 & C^2-S^2 \end{pmatrix} \begin{pmatrix} x \\ x' \\ y \\ y' \end{pmatrix}_{in} \tag{6.7}$$

As mentioned above, however, we cannot, to the accuracy we are working, ignore end effects. At the ends there will be x and y components of magnetic field which can be approximated as

$$B_x = a(z)x; \qquad B_y = a(z)y \tag{6.8}$$

where $a(z)$ is a factor which can be obtained from the actual field component B_z which presumeably falls off more slowly than is given by (6.1). These field components are related by

$$\frac{\partial B_x}{\partial x} + \frac{\partial B_y}{\partial y} + \frac{\partial B_z}{\partial z} = 0 \tag{6.9}$$

When integrated from well outside to well inside the solenoid this yields

$$\int_{(\perp L/2)\perp}^{(\perp L/2)+} a(z)dz = \frac{-B_0}{2} \tag{6.10}$$

Neglecting the length of the interval over which the fields are changing, the end can be represented by a transfer matrix resembling that of a skew quadrupole as in (5.10) but with one sign reversed. For example a particle with input vector $(x,0,0,0)^T$ acquires a vertical deflection given by

$$\Delta y' = -\frac{cB_0/2}{pc/e}x = -Kx \tag{6.11}$$

and similarly for $(0,0,y,0)^T$. When this is represented as a transfer matrix and the output is also then, when combined with (6.7), the full transfer map for the solenoid is

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & -K & 0 \\ 0 & 0 & 1 & 0 \\ K & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & CS/K & 0 & S^2/K \\ 0 & C^2-S^2 & 0 & 2CS \\ 0 & -S^2/K & 1 & CS/K \\ 0 & -2CS & 0 & C^2-S^2 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & K & 0 \\ 0 & 0 & 1 & 0 \\ -K & 0 & 0 & 1 \end{pmatrix}$$

Performing the multiplication yields the solenoid transfer matrix

$$R_S = \begin{pmatrix} C & S \\ -S & C \end{pmatrix} \begin{pmatrix} E & 0 \\ 0 & E \end{pmatrix} \tag{6.12}$$

where

$$E = \begin{pmatrix} C & S/K \\ -SK & C \end{pmatrix}; \tag{6.13}$$

For a "thin" solenoid (which in high energy accelerators is what they almost

always are) there is a simple way in which it can be represented by a thin "multipole", call it R_t sandwiched between two drifts of length $L/2$, call their transfer matrices $R_{L/2} = \ell^{\perp 1}$. R_t is given by

$$R_t = \ell R_S \ell = \begin{pmatrix} C\mathcal{M} & S\mathcal{M} \\ -S\mathcal{M} & C\mathcal{M} \end{pmatrix} = \begin{pmatrix} C & S \\ -S & C \end{pmatrix} \begin{pmatrix} \mathcal{M} & 0 \\ 0 & \mathcal{M} \end{pmatrix}, \tag{6.14}$$

where the matrix $\mathcal{M}$ is given by

$$\mathcal{M} = \ell E \ell = \begin{pmatrix} 1 & -L/2 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} C & S/K \\ -SK & C \end{pmatrix} \begin{pmatrix} 1 & -L/2 \\ 0 & 1 \end{pmatrix}. \tag{6.15}$$

To consider the possibility of breaking the solenoid into shorter lengths one can let L become small while holding K fixed; in that limit

$$\mathcal{M} = \begin{pmatrix} C + \frac{KL}{2}S & \frac{C}{K}(S - KL) + \frac{KL^2}{4}S \\ -KS & C + \frac{KL}{2}S \end{pmatrix} \to \begin{pmatrix} 1 & 0 \\ -K^2L & 1 \end{pmatrix}. \tag{6.16}$$

This has the satisfactory property that x and y are both continuous as a particle passes through it which makes it a "kick" and hence symplectic. Of course the drifts are also symplectic. This means that a solenoid can be made into multiple "thin" elements by breaking it into shorter lengths. In practice, this is probably academic because realistic solenoids really are thin in the sense that KL is much smaller than 1, and in any case breaking the solenoids into lengths shorter than the distance over which the end fields fall off would be illusory.

In this way we see that solenoids can be replaced by thin elements in the same spirit as other elements are in the program *teapot*, with symplecticity preserved, tracking being exact in the thin element lattice, and more faithful representation of thick elements resulting from breaking them into thinner elements. The

approximate thin element transfer matrix is

$$R_t = \begin{pmatrix} C & S \\ -S & C \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ -K^2L & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -K^2L & 1 \end{pmatrix} \tag{6.17}$$

This is a lens which is focusing in both planes and has focal length $(K^2L)^{\perp 1}$ followed by a coupling element which is a rotation by the small angle KL around the longitudinal axis. The *teapot* code uses Eq. (6.17) which though approximate, being the product of a pure rotation and an obviously symplectic transformation, is exactly symplectic. As with other elements one can investigate the accuracy of the approximations by splitting a solenoid into shorter solenoids. Normally the dominant effect of a solenoid is the rotation.

7. Compensation of Coupling

Returning to equations (5.12) and (5.13) and incorporating the results of the previous section on solenoids we can express the requirement that the lattice be decoupled at one point by the requirement that the four elements of $C + \bar{B}$ each vanish. This will be much stronger than the condition that $\det|C + \bar{B}|$ vanish, which, in section 5, was shown to be necessary for the tunes to be brought into coincidence. At the location of a detector (label it d) one attempts to reduce the "off-diagonal matrix"

$$\frac{C + \bar{B}}{\Lambda_A - \mathrm{tr}D} = \begin{pmatrix} R_{A11}(d) & R_{A12}(d) \\ R_{A21}(d) & R_{A22}(d) \end{pmatrix} = \mathcal{B}_y^{\perp 1}(d) \begin{pmatrix} \Sigma q_i \tilde{T}_i(d) & \Sigma q_i \tilde{U}_i(d) \\ \Sigma q_i \tilde{V}_i(d) & \Sigma q_i \tilde{W}_i(d) \end{pmatrix} \mathcal{B}_x(d) \tag{7.1}$$

where

$$\begin{aligned} \tilde{T}_i(d) =& \frac{1}{2(C_x - C_y)} \left[\pm S_y^{\bar{i}}(d) C_x^i(d) + C_x^{\bar{i}}(d) S_y^i(d) \right] \\ \tilde{U}_i(d) =& \frac{1}{2(C_x - C_y)} \left[\pm S_y^{\bar{i}}(d) S_x^i(d) - S_x^{\bar{i}}(d) S_y^i(d) \right] \end{aligned} \tag{7.2}$$

and there are two similar equations for $\tilde{V}_i(d)$ and $\tilde{W}_i(d)$ which will not be needed

in what follows. Performing the matrix multiplication in (7.1)and using an *ad hoc* abbreviated notation we obtain

$$\begin{pmatrix} R_{A11} & R_{A12} \\ . & . \end{pmatrix} = \begin{pmatrix} \beta_y^{1/2} & 0 \\ -\alpha_y\beta_y^{\perp 1/2} & \beta_y^{\perp 1/2} \end{pmatrix} \begin{pmatrix} \tilde{T} & \tilde{U} \\ \tilde{V} & \tilde{W} \end{pmatrix} \begin{pmatrix} \beta_x^{\perp 1/2} & 0 \\ \alpha_x\beta_x^{\perp 1/2} & \beta_x^{1/2} \end{pmatrix}$$
$$= \begin{pmatrix} \beta_x^{\perp 1/2}\beta_y^{1/2}(\tilde{T} + \alpha_x\tilde{U}) & \beta_x^{1/2}\beta_y^{1/2}\tilde{U} \\ . & . \end{pmatrix} \tag{7.3}$$

This performs the transformation back from the circular representation to correlate with formulas in sections 1-5.

Various features of these equations have to be explained. The "off-diagonal" matrix R_A has been reintroduced from (2.34) and evaluation of its elements in terms of coupling strengths q_i is from (5.12) and (5.13). The $\pm$ option distinguishes between skew quads and solenoids as was described in the last section with the + sign being appropriate for skew quads. The index d has been introduced since these conditions may be applied at various locations d in the ring where the state of coupling can be measured. New symbols for the trigonometric functions have natural meanings such as $S_y^{\bar{i}}(d) = \sin(\mu_y - \mu_y^i + \mu_y^d)$, if $\mu_y^i > \mu_y^d$. For y betatron motion μ_y is the phase advance around the whole ring and $\mu_y^i - \mu_y^d$ is the phase advance from the detector location d to the skew element location i. This amounts to setting the origin at d in the formulas derived up to this point. Care must however be taken in evaluating (7.1) since the rotation angles in (5.10) were implicitly assumed to be positive. To make $\mu_y^i - \mu_y^d$ always positive (and similarly for x), the route from the detector to element i should always be in the same, say clockwise, direction.

For these formulas it is assumed that the coupling is weak enough that terms beyond linear in the q_i's can be neglected. Also, in evaluating the various factors, the unperturbed phase advances and Twiss parameters are assumed to be available and reliable. (As mentioned before they include the focusing effect of solenoids.) In a real accelerator with errors these requirements would not necessarily be met.

We now consider the application of these relations to the decoupling of an accelerator. In the sums appearing in (7.1) some of the q_i's are presumeably unknown while others (call them q_a with a standing for "adjustor") are adjustable skew quads at locations labeled a. Separating these off the elements of R_A can be written

$$\begin{aligned} R_{A11}(d) &= R^{(0)}_{A11}(d) + \sum_{a=1}^{N_a} q_a T_a(d) \\ R_{A12}(d) &= R^{(0)}_{A12}(d) + \sum_{a=1}^{N_a} q_a U_a(d) \end{aligned} \tag{7.4}$$

where

$$\begin{aligned} T_a(d) =& \beta_x^{\perp 1/2}(d)\beta_y^{1/2}(d)[\tilde{T}_a(d) + \alpha_x(d)\tilde{U}_a(d)] \\ U_a(d) =& \beta_x^{1/2}(d)\beta_y^{1/2}(d)\tilde{U}_a(d) \end{aligned} \tag{7.5}$$

where there are N_a adjustors in all and again only two of the four matrix elements have been written. Unknown couplers have been lumped in the terms $R^{(0)}_{A11}(d)$ and $R^{(0)}_{A12}(d)$.

As usual with correctors there are various ways of proceeding. One idea (which is only rarely a good one) is to write as many conditions as there are adjustors and solve the resultant equations (7.1) for the adjustor strengths to make R_A vanish. We will proceed instead with a least squares prescription defining and then minimizing a "badness function"

$$B(q_1, ..., q_{N_a}) = \sum_{d=1}^{N_d} e_A^2(d)\beta_x(d)/\beta_y(d) \tag{7.6}$$

where $e_A^2(d)$ was defined in (3.22). Recall that e_A is the directly measurable ratio of "wrong-plane amplitude" to "in-plane amplitude" for an eigenmotion. The β ratio converts this to an emittance ratio which results in all detector measurements having comparable weight in the subsequent fitting procedure. The badness is also approximately the appropriately normalized sum of the squares

of the elements in the upper row of R_A summed over all of the N_d detector locations. The reason that only these elements are used can be understood by reviewing the exercise at the end of section 3. In that exercise it was shown that those two elements can be extracted from measurements with the monitor at location d. The lower-row-elements are related to slopes at d which are not directly measureable. It will be seen below that the explicit solution depends on having measured R_{A11} and R_{A12} individually, even though it is only a sum of their squares which enters B.

Values of $q_1, ..., q_{N_a}$ will be sought which minimize the badness B and that leads to the conditions

$$\frac{\partial B}{\partial q_a} = 0; \quad a = 1, ..., N_a \tag{7.7}$$

It is assumed that there are at least as many measurements as there are adjustors. (That is $N_a \leq 2N_d$.) By working with two detectors which are close to each other and are known to have no coupling elements between them, reduction of the other two elements of R_A could also be enforced. Alternatively, the sum of squares in (7.6) could be extended to include them, but then the prescription would probably not be practical in a real accelerator.

What follows amounts to solving the equations (7.7), with enough abbreviations being introduced to make the equations look tractible. Using (3.22), (7.4), and (7.5), B can be expressed as a quadratic function of the unknown skew quad strengths, which can be be represented as a vector $\mathcal{Q} = (q_1, ..., q_{N_a})^T$. Also define a $N_a \times N_a$ coefficient matrix

$$\mathcal{M} = (\mathcal{M}_{ab}) = \sum_{d=1}^{N_d} \big(\big[T_a(d) - U_a(d)\alpha_x(d)/\beta_x(d)\big] \big[T_b(d) - U_b(d)\alpha_x(d)/\beta_x(d)\big] + \\ + U_a(d)U_b(d)/\beta_x^2(d)\big) \beta_x(d)/\beta_y(d) \tag{7.8}$$

and define a vector containing the inhomogeneous terms in (7.7)

$$\mathcal{V} = (\mathcal{V}_a) = -\sum_{d=1}^{N_d} \big(\big[R^{(0)}_{A11}(d) - R^{(0)}_{A12}(d)\alpha_x(d)/\beta_x(d) \big] \big[T_a(d) - U_a(d)\alpha_x(d)/\beta_x(d) \big] + \\ + R^{(0)}_{A12}(d) U_a(d)/\beta_x^2(d) \big] \big) \beta_x(d)/\beta_y(d) \tag{7.9}$$

Then (7.7) becomes $\mathcal{M}\mathcal{Q} = \mathcal{V}$ with solution

$$\mathcal{Q} = \mathcal{M}^{\perp 1}\mathcal{V} \tag{7.10}$$

APPENDIX H

LATTICE IMPERFECTIONS AND THEIR COMPENSATION

1. Introduction.

This appendix has been copied from another source. The early sections have been left in for coherence even though the material is not that germane to the operation of **teapot**. The relevant material begins in Section 4.

In these notes some of the more elementary possible lattice defects will be analysed and methods will be described which can be used to compensate for them. In order to perform such compensation it is first necessary to have instruments present to measure the deviation from design values. We will adhere to a notation in which the letter d (for detector) is used as the index identifying such detectors, there being N_d detectors altogether. The other requirement is to have N_a elements (a for adjustor) which are to be set based on the detector readings to give a "best" compensation. Though it is not the only possibility we will describe only methods for which "best" means a least-squares minimum solution and there will be at least as many detectors as there are adjustors. ($N_d \geq N_a$)

Emphasis will be placed on defects of periodic lattices. Any circular accelerator satisfies this requirement, at least once a circulating beam has been achieved. Another important problem is the adjustment of finite nonperiodic lattice sections. The terminology "open sector" will be used to refer to such a section of beam-line.

Since the most important parameters of an accelerator are its tunes, it is appropriate to analyse first those errors which affect them. It is, however, economical before that to derive a general formula which can be used to analyse any perturbation of an otherwise ideal lattice; that will be done in Section 2, and the formula used to calculate the local closed orbit shift caused by a dipole perturbation. In Section 3 "The Golden Rule", which gives the tune shift caused

by a change in quadrupole strength, is derived. This is used to perform those tune adjustments whose importance was just emphasized. During accelerator operations it is customary to fix the tunes first thing and to reset them every time any subsequent adjustment causes them to shift. This places a very reliable constraint on the lattice functions, which are used in calculating subsequent adjustments, thereby enhancing confidence in the validity of the procedures. The next most important lattice adjustment is the adjustment of the beam orbit onto the centerline of the magnetic elements. Bend errors in the dipole elements and survey errors in quadrupole placement are normally the dominant sources of closed orbit errors. To simplify the analysis an algebraic transformation to a "circular" representation of betatron motion is described in Section 4; it is used to propagate the closed orbit around the ring in Section 5. In Section 6 closed orbit adjustment is described and in Section 7 the adjustment of the central orbit through open sectors is described.

2. A Difference Equation Which Describes The Effect of a Single Bend Error.

Let us suppose that there is a zero-length perturbing element in the ring at a point which, for now, we take to be the origin. On the t'th turn it causes a deflection given by Δp_{yt}. Here, for definiteness, we are working explicitly with y motion but the formulas apply equally to x. Propagation once around the ring is described by the transfer map in "Twiss form",

$$\begin{pmatrix} y \\ p_y - \Delta p_y/2 \end{pmatrix}_{t+1} = \begin{pmatrix} C_y + \alpha_y S_y & \beta_y S_y \\ -\gamma_y S_y & C_y - \alpha_y S_y \end{pmatrix} \begin{pmatrix} y \\ p_y + \Delta p_y/2 \end{pmatrix}_t \tag{2.1}$$

and a similar equation can be written for backwards propagation from t to $t-1$. Note that p_y is evaluated at the middle of the perturbing element; it is necessary to be specific about this since p_y varies discontinuously in passing through the element. With μ standing for the tune Q times 2π, we are using the notation

$C_y \equiv \cos\mu_y$ and $S_y \equiv \sin\mu_y$ and are intentionally using the subscript t as a turn index to be suggestive of the time measured in units of the revolution period. It will however always be an integer. For these two maps the top equations are

$$\begin{aligned} y_{t+1} &= (C_y + \alpha_y S_y) y_t + \beta_y S_y (p_y + \Delta p_y/2)_t \\ y_{t\perp 1} &= (C_y - \alpha_y S_y) y_t - \beta_y S_y (p_y - \Delta p_y/2)_t \end{aligned} \tag{2.2}$$

By adding the equations (2.2) one eliminates p_y and obtains

$$y_{t+1} - 2C_y y_t + y_{t\perp 1} = \beta_y S_y \Delta p_{yt} \tag{2.3}$$

After solving this for y_t it will be possible to obtain p_{yt} from the equation

$$p_{yt} = \frac{y_{t+1} - y_{t\perp 1} - 2\alpha_y S_y y_t}{2\beta_y S_y} \tag{2.4}$$

which is obtained by subtracting the equations (2.2).

Initially we will analyse the effect of a constant bend error so the deflection ΔP_{yt} will not, in fact, depend on t, and hence will be symbolized by ΔP_y. The term on the rhs of (2.3) can be called an inhomogeneous term while all terms on the lhs are homogeneous. As with differential equations the solution will be the sum of a definite solution of the inhomogeneous equation plus the superposition of any solution of the homogeneous equation. We know that the latter solution corresponds to free betatron oscillation which is not presently of interest, and we set it to zero. Solution of the inhomogeneous equation is trivial, with the result

$$y = \beta_y \frac{S_y/2}{1 - C_y} \Delta p_y \tag{2.5}$$

and using (2.4) the slope at the center of the perturbing element can be obtained

$$p_y = -\alpha_y \frac{S_y/2}{1 - C_y} \Delta p_y \tag{2.6}$$

The displacement y is continuous across the thin bend element but there is a kink in the slope as shown in the figure.

An important feature of closed orbit deformation by bend errors can be inferred immediately from (2.5) and that is that the deformation becomes arbitrarily large when the cosine of the tune C_y approaches 1. This occurs when the tune approaches an integer and is a manifestation of the so-called "integer-resonance". When the tune is an exact integer both the particle coordinate and slope repeat exactly after a full turn so that the deflection Δp_y accumulates every turn; a divergent process. Resonances are the natural enemies of accelerators; they are always due to the accumulation of undesireable behavior over many turns. Since a particle in an accelerator circulates without damping for so many turns it is highly susceptible to this. This integer resonance is the most elementary and the most lethal of such resonances. Even when the cosine is not exactly 1 the presence of the factor $1 - C_y$ in the denominator of (2.5)leads to a strong sensitivity of the closed orbit to bend errors for tune values close to an integer, and for that reason, such tune values are normally avoided.

3. The Golden Rule Relating Quadrupole Perturbations and Tunes.

Starting from the difference equation (2.3) which relates the displacements on three successive turns the tune shift due to a quadrupole perturbation can be obtained directly. The deflections suffered by a particle as it passes through an erect thin quadrupole of focal length f are given by

$$\begin{aligned} \Delta p_{xt} =&(q_x/\beta_x)x_t \\ \Delta p_{yt} =&(q_y/\beta_y)y_t \end{aligned} \tag{3.1}$$

where "normalized" quad strengths q_x and q_y have been defined by

$$\begin{aligned} q_x =&\beta_x/f \\ q_y =&-\beta_y/f \end{aligned} \tag{3.2}$$

These are dimensionless. If f were the focal length of a regular arc quad then q_x and q_y would be of order 1 (worth remembering for the mental evaluation of some of the following formulas), but the magnitude of a typical perturbation where these formulas will be employed will be less by perhaps a factor of 100 or more. The reader may be annoyed to see the intrusion of beta-functions into such basic formulas but it can be further justified as follows. It is only for historic reasons that a thin lens is characterized by its focal length. The inverse focal length is more natural, being proportional to the lens "strength", and the symbol q is often used for that. As long as one is paying the price of introducing a new symbol it seems sensible to obtain some further benefit. Working with dimensionless quadrupole strengths symplifies many future formulas and incorporating the minus sign at this point will save us from writing separate formulas for x and y motion. As defined, for either plane, positive q corresponds to a defocusing quad. When (3.1) is substituted into (2.3), the result is

$$y_{t+1} - 2Cy_t + y_{t\perp 1} = Sqy_t \tag{3.3}$$

where y can refer either to horizontal or vertical motion and it has accordingly

been suppressed as a sub-script. Naturally C and S are to be evaluated for the corresponding tune.

In (3.3) the effect of the perturbing quadrupole is incorporated on the rhs of the equation while the rest of the lattice is described by the lhs. But clearly the rhs can be grouped with the second term on the lhs since they are both proportional to y_t. Since the coefficient of this combined term can be nothing other than $\cos 2\pi(Q + \Delta Q)$, the cosine of the perturbed phase advance per turn, we get

$$\cos 2\pi(Q + \Delta Q) = \cos 2\pi Q + qS/2 \tag{3.4}$$

where ΔQ is, naturally enough, called the tune shift caused by the quadrupole perturbation. This is an exact relationship, and it is simple enough, but an approximate form obtained by Taylor expansion valid for small ΔQ is what is normally used. That result is

$$\Delta Q = -\frac{q}{4\pi} \tag{3.5}$$

This will be referred to as "The Golden Rule" as it is so simple and so important. Notice that the result is independent of the location in the ring where the element is placed, though a lattice dependent factor has been factored out in the definition (3.2). Also note that a focusing quad in fact focuses, which shortens the betatron wavelength and increases the tune.

If there are many small quadrupole perturbations q_i then, to terms linear in the q_i's, the tune shift is given by

$$\Delta Q = -\frac{1}{4\pi} \sum q_i \tag{3.6}$$

Commonly in a sum like this some of the terms, being due to errors, are unknown, while others correspond to compensating elements which the accelerator operator can adjust. Grouping the former terms and calling their sum $\Delta Q^{(0)}$, the tune

shift is given by

$$\Delta Q = \Delta Q^{(0)} - \frac{1}{4\pi}\sum_{a=1}^{N_a} q_a \tag{3.7}$$

where, as mentioned earlier, the subscript a is used for adjustors. This is the first encountered, and the simplest, of the equations of this type which are used to determine the settings of adjustors. If there is just one adjustor and it is desired that ΔQ vanish, we get

$$q = 4\pi \Delta Q^{(0)} \tag{3.8}$$

To apply this formula $\Delta Q^{(0)}$ would be measured using a beam position monitor and spectrum analyser and q would be set accordingly.

4. The Circular Representation of Betatron Motion.

We shall be analysing betatron motion in a lattice which has small deviations away from the design elements and for that it is convenient to perform a transformation of uncoupled motion in which propagation from point to point is represented by pure rotation in phase space with the transfer matrix taking the form

$$\begin{pmatrix} \cos\Delta\psi & \sin\Delta\psi \\ -\sin\Delta\psi & \cos\Delta\psi \end{pmatrix} \tag{4.1}$$

where $\Delta\psi$ is the appropriate x or y betatron phase advance in going from the first to the second point. To achieve this one performs the following transformation from $X^T \equiv (x, p_x, y, p_y)$ to $\tilde{X}^T \equiv (\tilde{x}, \tilde{p}_x, \tilde{y}, \tilde{p}_y)$

$$\tilde{X} = \mathcal{B} X \tag{4.2}$$

where

$$\mathcal{B} = \begin{pmatrix} \mathcal{B}_x & 0 \\ 0 & \mathcal{B}_y \end{pmatrix} \tag{4.3}$$

$$\mathcal{B}_x = \begin{pmatrix} \beta_x^{\perp 1/2} & 0 \\ \alpha_x \beta_x^{\perp 1/2} & \beta_x^{1/2} \end{pmatrix} \tag{4.4}$$

$$\mathcal{B}_x^{\perp 1} = \begin{pmatrix} \beta_x^{1/2} & 0 \\ -\alpha_x \beta_x^{\perp 1/2} & \beta_x^{\perp 1/2} \end{pmatrix} \tag{4.5}$$

and similarly for y. In the new variables (which will be called the circular representation because the phase-space orbit is a circle) the x invariant emittance is given by $\epsilon_x = \tilde{x}^2 + \tilde{p}_x^2$ and similarly for y. Defining the once-around transfer matrix by

$$M = \begin{pmatrix} A & 0 \\ 0 & D \end{pmatrix} \tag{4.6}$$

and the once-around transfer matrix in the new representation by

$$\tilde{M} = \begin{pmatrix} \tilde{A} & 0 \\ 0 & \tilde{D} \end{pmatrix} \tag{4.7}$$

one obtains

$$\begin{aligned} A =& \mathcal{B}_x^{\perp 1} \tilde{A} \mathcal{B}_x \\ D =& \mathcal{B}_y^{\perp 1} \tilde{D} \mathcal{B}_y \end{aligned} \tag{4.8}$$

In terms of the dimensionless quadrupole strengths defined in (3.2), where f is the focal length of the quadrupole (which is assumed to be erect), the transfer matrix is given by

$$\begin{pmatrix} \mathcal{B}_x & 0 \\ 0 & \mathcal{B}_y \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1/f & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -1/f & 1 \end{pmatrix} \begin{pmatrix} \mathcal{B}_x^{\perp 1} & 0 \\ 0 & \mathcal{B}_y^{\perp 1} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ q_x & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & q_y & 1 \end{pmatrix} \tag{4.9}$$

It is difficult to maintain a sign convention which is universally regarded as natural and, for that reason, the reader has to take responsibility for getting the signs right when these formulae are applied to an actual problem. In a later section in

which skew quadrupoles and other elements which couple the x and y motions are analysed this full 4×4 formalism will be re-introduced, but for now we can analyse either the x or the y motion separately using a 2×2 formalism.

5. Propagation of the Closed Orbit Around the Ring.

Still assuming a single bend error the results of the previous sections can be combined to obtain the closed orbit anywhere in the ring, since propagation around the unperturbed lattice is indistinguishable from a free betatron oscillation. At this point we will simplify the notation a bit by suppressing the subscript y but adding a new subscript i which identifies the particular bend error in preparation for handling many such errors. Adapting (2.5) and (2.6) accordingly yields the closed orbit coordinates at the center of and just after an element causing deflection Δp_i as

$$\begin{aligned} y_i &= \beta_i \frac{S/2}{1-C} \Delta p_i \\ p_i &= -\alpha_i \frac{S/2}{1-C} \Delta p_i \\ p_i+ &= (-\alpha_i \frac{S/2}{1-C} + \frac{1}{2}) \Delta p_i \end{aligned} \tag{5.1}$$

We wish to propagate this closed orbit through the lattice to the location of a position detector d. This is accomplished by transformation to the circular representation followed by propagation around the ring using matrix multiplication. Explicitly $(y_d, p_d)^T / \Delta p_i$ is given by

$$\begin{pmatrix} \beta_d^{1/2} & 0 \\ -\alpha_d \beta_d^{\perp 1/2} & \beta_d^{\perp 1/2} \end{pmatrix} \begin{pmatrix} C(d,i) & S(d,i) \\ -S(d,i) & C(d,i) \end{pmatrix} \begin{pmatrix} \beta_i^{\perp 1/2} & 0 \\ \alpha_i \beta_i^{\perp 1/2} & \beta_i^{1/2} \end{pmatrix} \begin{pmatrix} \beta_i \frac{S/2}{1 \perp C} \\ -\alpha_i \frac{S/2}{1 \perp C} + \frac{1}{2} \end{pmatrix} \tag{5.2}$$

where the notation is that $C(d,i)$ stands for the cosine of the phase advance ϕ_{di} from i to d and similarly for $S(d,i)$. Some of the following formulas will only make sense if ϕ_{di} is nonnegative. Completing the matrix multiplications in

(5.2)yields the result

$$\begin{aligned}\frac{y_d}{\sqrt{\beta_d}} =&[\frac{S}{1-C}C(d,i)+S(d,i)]\sqrt{\beta_i}\Delta p_i/2\\ =&\frac{\cos(\mu/2-\phi_{di})}{2\sin\frac{\mu}{2}}\sqrt{\beta_i}\Delta p_i\end{aligned} \tag{5.3}$$

and a similar relation for p_d which will not be as useful to us since the detector at d measures y_d not p_d.

As was done in the discussion of quadrupole perturbations we now superimpose the terms from all bend errors after first segregating those terms due to unknown elements from those due to adjustors, with the former being lumped into a term $y_d^{(0)}/\sqrt{\beta_d}$.

$$\frac{y_d}{\sqrt{\beta_d}} = \frac{y_d^{(0)}}{\sqrt{\beta_d}} + \sum_{a=1}^{N_a}\frac{\cos(\mu/2-\phi_{da})}{2\sin\mu/2}\sqrt{\beta_a}\Delta p_a \tag{5.4}$$

6. Improvement of the Closed Orbit Using Steering Correctors.

In an accelerator there are invariably steering elements present for the purpose of improving the closed orbit. In the design of early (and hence small) accelerators bend and survey tolerances were held tight enough to to assure that the closed orbit stayed within the vacuum chamber, but as the machines became larger this became progressively more difficult. Fortunately it was also found to be operationally easy to adjust steering elements based on the orbit measurement by beam position monitors (bpm's). That will now be described. Usually with correctors there are various ways of proceeding. One idea (which is only rarely a good one) is to write as many conditions as there are adjustors and then solve the resultant equations (5.4) for the adjustor strengths to make y_d vanish at each of the detectors. We will proceed instead with a least-squares prescription, defining

and then minimizing a "badness function"

$$B(\Delta p_1, ..., \Delta p_{N_a}) = \sum_{d=1}^{N_d} y_d^2(d)/\beta(d) \tag{6.1}$$

Values of $\Delta p_1, ..., \Delta p_{N_a}$ will be sought which minimize the badness B and that leads to the conditions

$$\frac{\partial B}{\partial \Delta p_a} = 0; \quad a = 1, ..., N_a \tag{6.2}$$

It is assumed that there are at least as many measurements as there are adjustors. (That is $N_a \leq 2N_d$.) By working with two detectors which are close to each other and are known to have no bend errors between them, reduction of the the slope p_d could also be enforced.

What follows amounts to solving the equations (6.2), with abbreviations being introduced to make the equations compact. Using (5.4) B can be expressed as a quadratic function of the unknown bend strengths, which can be be represented as a vector $\mathcal{Q} = (\Delta p_1, ..., \Delta p_{N_a})^T$. Also define a $N_a \times N_a$ coefficient matrix

$$\mathcal{M} = (\mathcal{M}_{ab}) = \sum_{d=1}^{N_d} \big(T_a(d) T_b(d)\big) \tag{6.3}$$

where

$$T_a(d) = \frac{\cos(\mu/2 - \phi_{da})}{2 \sin \mu/2} \sqrt{\beta_a} \tag{6.4}$$

and define a vector containing the inhomogeneous terms in (6.2)

$$\mathcal{V} = (\mathcal{V}_a) = -\sum_{d=1}^{N_d} \frac{y_d^{(0)}}{\sqrt{\beta_{(d)}}} T_a(d) \tag{6.5}$$

Then (6.2) becomes $\mathcal{M}\mathcal{Q} = \mathcal{V}$ with solution

$$\mathcal{Q} = \mathcal{M}^{\perp 1} \mathcal{V} \tag{6.6}$$

APPENDIX I

MODELING THE EFFECTS OF WIGGLERS

Weiru Wang and Richard Talman

An ideally designed wiggler would cause neither an orbit displacement nor an orbit deflection, but that is not the usual situation. An even number of identical but alternating poles cause no net deflection, but cause an orbit translation. An odd number of poles could cause no orbit translation but cause a deflection. Only by having half-poles at each end would one achieve both advantages. If (as we assume) the wiggler designer was not far-sighted enough to provide this feature, it is necessary for the lattice designer to provide extra steering as shown in the figure.

An easy compensation is possible for a wiggler with an odd number of identical alternating poles, if there are steering elements just upstream and just downstream as shown.

The lower figure shows that one certainly does not want to insist on normal entry to the first bend element. This, plus the fact that negative bend sector magnets, are at best confusing, suggests that wigglers are best modeled by rectangular bending magnets, or "RBEND's" as they are called in the standard lattice description. For this reason *TEAPOT* was modified to accept RBEND's in January, 1994. RBEND elements cause vertical focusing but no horizontal focusing. Upon deep reflection you are supposed to be able to convince yourself that both + and − poles cause focusing (as contrasted to both defocusing, or focusing/defocusing).

Tuning up *TEAPOT* to model wigglers must proceed by stages, and depends on understanding the idiosyncrasies of the code. The most important of these is that magnet elements such as SBEND, RBEND, QUADRUPOLE, appearing in the lattice description, **define** the design orbit. No matter how mispowered such

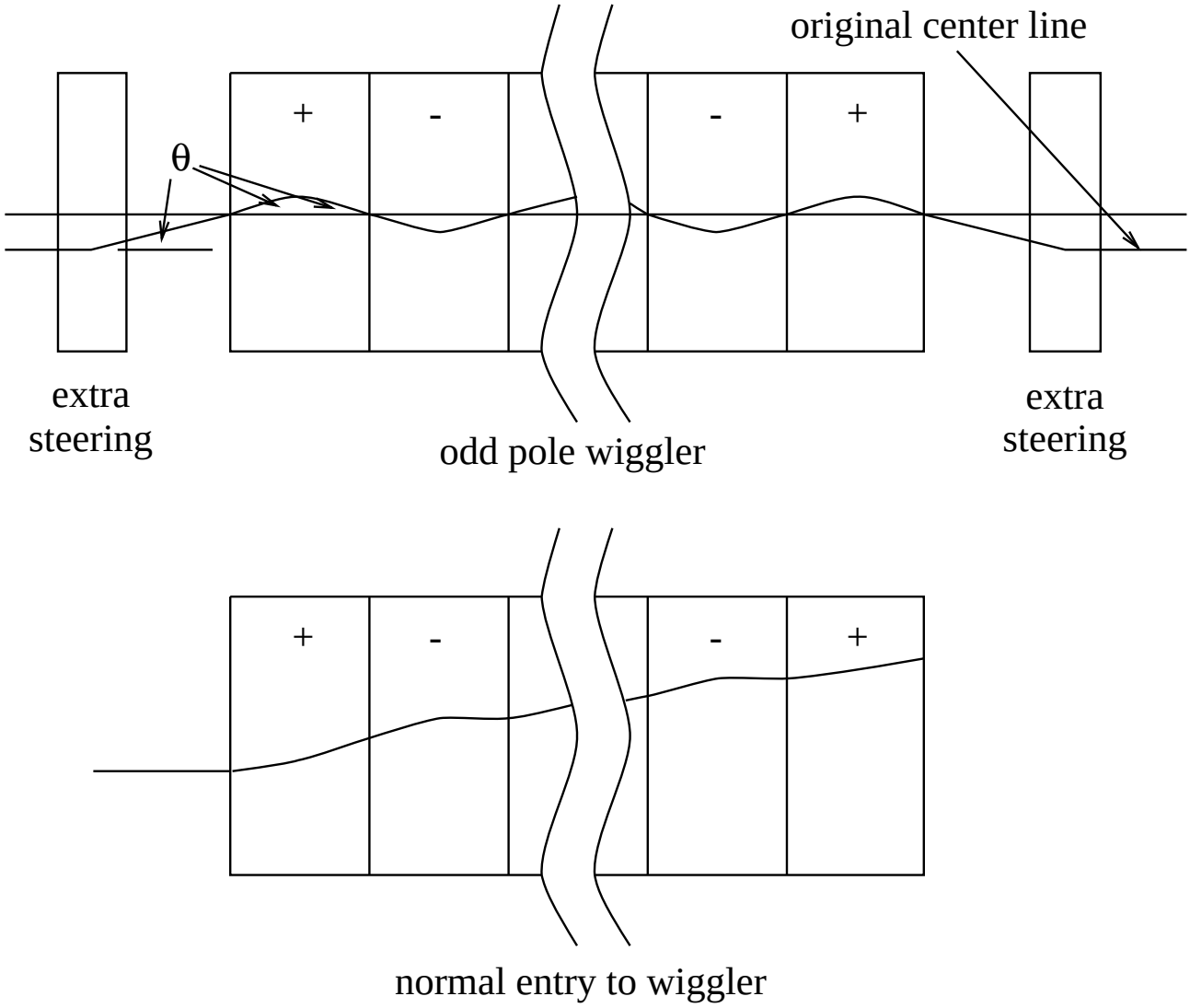

Figure 4: Steering effects of wiggler.

elements are, the code declares there to be no closed orbit deviations. (There is, however, control on the global closed orbit in that deviations from once-around closure, both in angle and position, are printed out.) When RBEND's are included in the lattice, they physically change the design orbit relative to what it was before but *TEAPOT* (correctly) asserts that there are no closed

orbit deviations (relative to what the input file has declared to be the desired lattice.) If the geometry is simple, as in the upper figure, the external steering elements can be dead-reckoned so that insertion of the wiggler plus steering has no effect on the orbit (except for a small increase in circumference and the small deviations internal to the wiggler.)

In the more realistic situation where there are not conveniently located external steering elements it is necessary to use more remote steering to compensate for the (inevitable) wiggler steering and/or displacement. If one wishes to use *TEAPOT* to adjust this steering one must go through a first stage in which the wiggler is modeled with HKICK elements. *TEAPOT* calculates deviations from (what it assumes is) the design orbit, and if asked using HSTEER or ORBCHEAT sets nearby steering elements (type khka) to minimize the closed orbit deviations at detectors (type khkd). At this phase the vertical focusing effect of the wiggler will not yet have been correctly included because HKICK does not include focusing. Once the HKICK settings have been determined, the HKICK elements modeling both the wiggler and the external steering must be replaced by RBEND elements. As mentioned before (except for the closure requirement) TEAPOT will (misleadingly) declare that the compensation has been perfect, and will indicate zero closed orbit deviations everywhere. This simply reflects (as in true machine operation) that one has redefined the design orbit to be the wiggler-on closed orbit. Sample files illustrating some of these procedures are in $TPOT/ftpot/test/dat/wigg and $TPOT/ftpot/test/dat/wigg . The following report by Weiru Wang gives the formulas used.

EFFECTIVE R AND T MATRICES FOR 3-D WIGGLER FIELDS

These notes describe formulas used for modeling wigglers in *TEAPOT*. Since longitudinal magnetic field components play an essential role, the traditional magnetic field "multipole expansion" cannot be used, but a truncated Taylor series can be. At this time it assumed in the code that the motion is fully relativistic. For application to proton accelerators it will be necessary to review

the formulas and correct them as appropriate. The $\mathcal{R}$ and $\mathcal{T}$ matrices calculated here (and confirmed by Runge-Kutta comparison) have been incorporated in *TEAPOT*. It should be appreciated that inclusion of this truncated approximation is necessarily non-symplectic, which may invalidate long term tracking results.

In Halbach's approximation, the magnetic field components in the wiggler are

$$\vec{B} = \begin{cases} B_x = B_0 \dfrac{k_x}{k_y} \sinh(k_x x) \sinh(k_y y) \cos(k_z z) \\ B_y = B_0 \cosh(k_x x) \cosh(k_y y) \cos(k_z z) \\ B_z = -B_0 \dfrac{k_z}{k_y} \cosh(k_x x) \sinh(k_y y) \sin(k_z z) \end{cases} \tag{8}$$

The wave numbers are related by

$$k_z^2 = k_x^2 + k_y^2.$$

A complete wiggler is made up of numerous wiggler sections. In this report one such period will be analysed and in a lattice description file there will be one entry for each period. For example, for a CESR wiggler, there are twelve periods. Consider a period centered on a symmetry point z=0 where $B_y = B_0$. as in Fig. 1 .

Each section contains a half pole on each end. When placed one after the other, only the end sections are left with half poles. This may or may not correspond to the actual hardware, but the present discussion assumes it to be the case. (For the present CESR configuration it appears to be a reasonable approximation.)

As a zeroth approximation, assume that the effect of the wiggler on the motion of electrons is sine-like in the x-direction and negligible in the y-direction.

$$x(z) = x_{in} + x'_{in} z + A_x \cos^2(\frac{k_z z}{2})$$

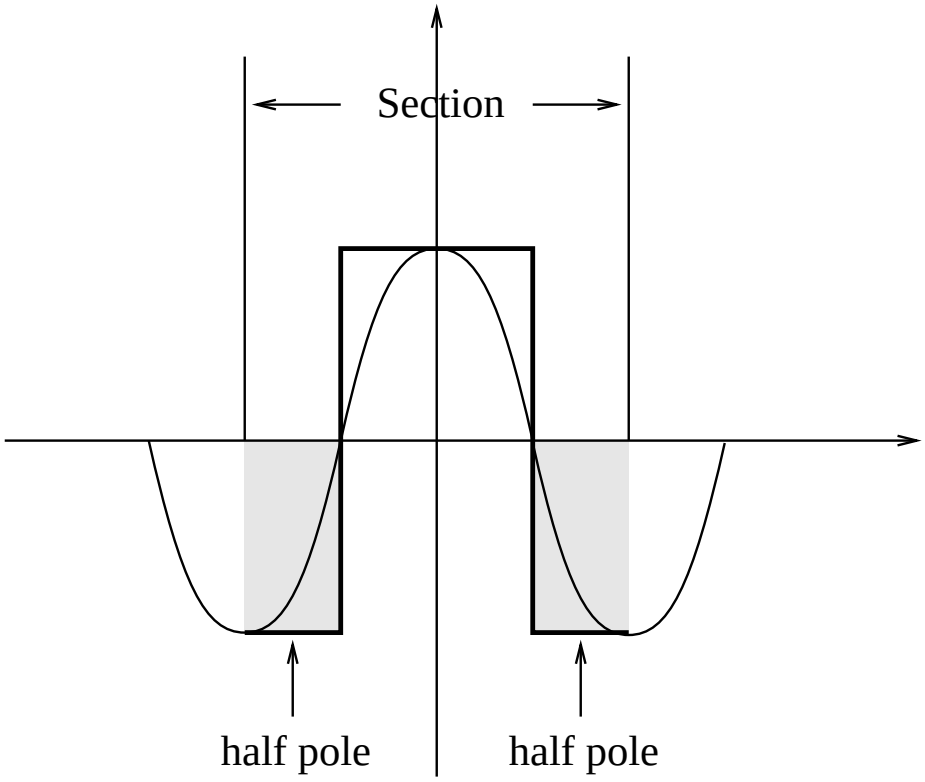

Figure 5: Section

$$\begin{cases} x'(z) = x'_{in} - \dfrac{k_z A_x}{2} \sin(k_z z) \\ y(z) = y_{in} + y'_{in} z \\ y'(z) = y'_{in} \\ z = ct \end{cases} \tag{9}$$

$x_{in},\ x'_{in},\ y_{in},\ y'_{in}$ are the initial values. Since the amplitude A_x is to be regarded as an input parameter for the wiggler it must be precalculated—the integral over B_y is easy for the equilibrium orbit if transverse velocity is neglected. The

coefficient A_x of the "ansatz" form used as zeroth approximation is

$$A_x = \frac{2cB_0}{(pc/e)k_z^2} \tag{10}$$

Corresponding to the displacements Eq. (9), the velocity components are

$$\vec{v} = \begin{cases} \frac{dx}{dt} = x'_{in}c - \frac{k_z A_x}{2}\sin(k_z z)c \\ \frac{dy}{dt} = y'_{in}c \\ \frac{dz}{dt} = c \end{cases} \tag{11}$$

Using the Lorentz force equation

$$\frac{dp}{dt} = e\vec{v} \times \vec{B} \tag{12}$$

the trajectory equation is

$$x''\hat{x} + y''\hat{y} = \frac{cB_0}{(p_0 c/e)k_y} \begin{pmatrix} \hat{x} & \hat{y} & \hat{z} \\ \frac{dx}{dt} & \frac{dy}{dt} & 1 \\ B_x & B_y & B_z \end{pmatrix} \tag{13}$$

and the angular deflections are

$$\Delta x' = \frac{cB_0}{(p_0 c/e)k_y}(y'_{in}k_z I_1 + k_y I_2)$$
$$\Delta y' = \frac{cB_0}{(p_0 c/e)k_y}(k_x I_3 + k_z x'_{in} I_1 - \frac{k_z^2 A_x}{2} I_4)$$

where

$$I_1 = \int_{\perp l}^{l} \cosh(k_x x)\sinh(k_y y)\sin(k_z z)dz$$

$$I_2 = \int_{\perp l}^{l} \cosh(k_x x)\cosh(k_y y)\cos(k_z z)dz$$
$$I_3 = \int_{\perp l}^{l} \sinh(k_x x)\sinh(k_y y)\cos(k_z z)dz \qquad (14)$$
$$I_4 = \int_{\perp l}^{l} \cosh(k_x x)\sinh(k_y y)\sin^2(k_z z)dz$$

To evaluate the integrals $I_1 \sim I_4$, we keep only leading terms for $x \to 0$

$$\cosh(x) \to 1 + \frac{x^2}{2} \quad \text{and} \quad \sinh(x) \to x$$

Assume $x_{in},\ x'_{in},\ y_{in},\ y'_{in}$ are small. Putting Eqs. (9) into Eqs. (14), and keeping only first order terms, we obtain the quantities needed to determine the $\mathcal{R}$-matrix elements

$$I_1 \approx y'_{in}(\frac{2\pi k_y}{k_z^2})$$
$$I_2 \approx \frac{\pi A_x^2 k_x^2}{4k_z} + x_{in}(\frac{\pi k_x^2 A_x}{2k_z})$$
$$I_3 \approx y_{in}(\frac{A_x \pi k_x k_y}{2k_z})$$
$$I_4 \approx y_{in}(\frac{\pi k_y}{k_z})$$

Note that there is a (small) constant contribution to I_2. It presumably causes a closed orbit shift that must, at least in principle, be compensated when the wiggler ie turned on.

To obtain both $\mathcal{R}$ and $\mathcal{T}$ matrices we use Mathematica to perform the expansions and integrations. Truncating to quadratic order, the results are

$$\frac{\Delta x'}{\frac{\pi c B_0}{p_0 c/e}} = \frac{A_x^2 k_x^2}{4k_z} + \frac{A_x k_x^2}{2k_z}x - \frac{2k_x^2}{k_z^3}x'^2 + \frac{A_x^2 k_x^2 k_y^2}{8k_z}y^2 - \frac{2k_y^2}{k_z^3}y'^2 - \frac{55A_x^2 k_x^2 k_y^2}{144k_z^3}y'^2$$

$$+\frac{\pi^2 A_x^2 k_x^2 k_y^2}{24 k_z^3} y'^2 + \frac{5 A_x^2 k_x^2}{24 k_z} y'^2 \tag{15}$$

$$\frac{\Delta y'}{\frac{\pi c B_0}{p_0 c/e}} = \frac{A_x k_x^2}{2 k_z} y - \frac{A_x k_z}{2} y - \frac{5 A_x^3 k_x^2 k_z}{64} y + \frac{A_x^2 k_x^2 k_z}{4} x y - \frac{2}{k_z} x' y' - \frac{5 A_x^2 k_x^2}{9 k_z} x' y'$$
$$+\frac{4 k_x^2}{k_z^3} x' y' + \frac{\pi^2 A_x^2 k_x^2}{12 k_z} x' y' \tag{16}$$

At the zero'th order approximation, the transfer matrix is

$$\mathcal{R}(z) \approx \begin{pmatrix} 1 & z & 0 & 0 \\ \frac{cB_0}{p_0 c/e}\frac{\pi k_x^2 A_x}{k_z} & 1 & 0 & 0 \\ 0 & 0 & 1 & z \\ 0 & 0 & \frac{cB_0}{p_0 c/e}\left(\frac{\pi A_x k_x^2}{2 k_z} - \frac{\pi k_z A_x}{2}\right) & 1 \end{pmatrix}$$

For this report, the numerical parameter values used are

$$B_0 = 1.2\,\mathrm{T},\ p_0 c/e = 5{\times}10^9\,\mathrm{V},\ k_x = 6.4\,\mathrm{m}^{\perp 1},\ k_z = \pi/0.098\,\mathrm{m}^{\perp 1},\ A_x = 1.4{\times}10^{\perp 4}\,\mathrm{m}.$$

These are the parameters used to describe the wiggler in the lattice description file (with the exception that, since they are redundant according to Eq. (10), the value $p_0 c/e$ is neither required nor allowed in the input.)

The "analytic" linear transfer matrix is

$$\mathcal{R} \approx \begin{pmatrix} 1 & 0.196 & 0 & 0 \\ 2.02 \times 10^{\perp 5} & 1 & 0 & 0 \\ 0 & 0 & 1 & 0.196 \\ 0 & 0 & -4.87 \times 10^{\perp 3} & 1 \end{pmatrix} \tag{17}$$

From Eq. (15) and Eq. (16) we obtain the second order transfer matrix

$$\mathcal{T}_{2ij} = \begin{pmatrix} 0.0000 & 0000 & 0.0000 & 0.0000 \\ 0.0000 & -0.5621 \times 10^{\perp 3} & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.6987 \times 10^{\perp 6} & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & -0.1355 \times 10^{\perp 1} \end{pmatrix} \tag{18}$$

$$\mathcal{T}_{4ij} = \begin{pmatrix} 0.0000 & 0.0000 & 0.7277 \times 10^{\perp 6} & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & -0.6494 \times 10^{\perp 2} \\ 0.7277 \times 10^{\perp 6} & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & -0.6494 \times 10^{\perp 2} & 0.0000 & 0.0000 \end{pmatrix} \tag{19}$$

Alternatively, the transfer matrix can be obtain by solving the equation of motion numerically.

$$\begin{aligned} \vec{X}'' &= \frac{cB_0}{p_0 c/e} \vec{X}' \times \vec{B} \\ &= \frac{cB_0}{p_0 c/e} \begin{pmatrix} \hat{x} & \hat{y} & \hat{z} \\ x' & y' & 1 \\ B_x & B_y & B_z \end{pmatrix} \end{aligned}$$

$$\begin{aligned} x'' &= \frac{cB_0}{p_0 c/e}(-\frac{k_x}{k_y} C_x S_y c_z y' - C_x C_y c_z) \\ y'' &= \frac{cB_0}{p_0 c/e}(\frac{k_x}{k_y} S_x S_y c_z + \frac{k_z}{k_y} C_x S_y s_z x') \end{aligned}$$

This is an initial value problem that can be solved numerically by using the Runge-Kutta method. The value $A_x = 1.4 \times 10^{\perp 4}\,\mathrm{m}$. Starting with different initial conditions, we obtained the transfer matrix

$$\begin{pmatrix} dx \\ dx' \\ dy \\ dy' \end{pmatrix} = \mathcal{R} \begin{pmatrix} dx_0 \\ dx'_0 \\ dy_0 \\ dy'_0 \end{pmatrix}$$

$$\mathcal{R} = \begin{pmatrix} 1.0000 & 0.1960 & 0.0000 & 0.0000 \\ 2.0249 \times 10^{\perp 5} & 1.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.99994 & 0.1960 \\ 0.0000 & 0.0000 & -4.8778 \times 10^{\perp 3} & 0.99995 \end{pmatrix}$$

The vertical and horizontal focusing agrees very well with Eq. (17). It remains to approximate the leading nonlinear deflection. The $\mathcal{T}$ matrix obtained by using

Runge-kutta method is

$$\mathcal{T}_{1ij} = \begin{pmatrix} 0.8530 \times 10^{\perp 8} & 0.5621 \times 10^{\perp 3} & 0.0000 & 0.0000 \\ 0.5621 \times 10^{\perp 3} & 0.5508 \times 10^{\perp 4} & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & -0.5788 \times 10^{\perp 5} & 0.1354 \times 10^{\perp 1} \\ 0.0000 & 0.0000 & 0.1354 \times 10^{\perp 1} & 0.2207 \times 10^{\perp 2} \end{pmatrix}$$

$$\mathcal{T}_{2ij} = \begin{pmatrix} 0.1907 \times 10^{\perp 6} & 0.1748 \times 10^{\perp 7} & 0.0000 & 0.0000 \\ 0.1748 \times 10^{\perp 7} & -0.5621 \times 10^{\perp 3} & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.3405 \times 10^{\perp 4} & 0.9086 \times 10^{\perp 5} \\ 0.0000 & 0.0000 & 0.9086 \times 10^{\perp 5} & -0.1354 \times 10^{\perp 1} \end{pmatrix}$$

$$\mathcal{T}_{3ij} = \begin{pmatrix} 0.0000 & 0.0000 & 0.1454 \times 10^{\perp 6} & -0.5621 \times 10^{\perp 3} \\ 0.0000 & 0.0000 & 0.6489 \times 10^{\perp 2} & -0.5508 \times 10^{\perp 4} \\ 0.1454 \times 10^{\perp 6} & 0.6489 \times 10^{\perp 2} & 0.0000 & 0.0000 \\ -0.5621 \times 10^{\perp 3} & -0.5508 \times 10^{\perp 4} & 0.0000 & 0.0000 \end{pmatrix}$$

$$\mathcal{T}_{4ij} = \begin{pmatrix} 0.0000 & 0.0000 & 0.7686 \times 10^{\perp 6} & 0.2790 \times 10^{\perp 6} \\ 0.0000 & 0.0000 & 0.7922 \times 10^{\perp 7} & -0.6489 \times 10^{\perp 2} \\ 0.7686 \times 10^{\perp 6} & 0.7922 \times 10^{\perp 7} & 0.0000 & 0.0000 \\ 0.2790 \times 10^{\perp 6} & -0.6489 \times 10^{\perp 2} & 0.0000 & 0.0000 \end{pmatrix}$$

$\mathcal{T}_{2ij}$ and $\mathcal{T}_{4ij}$ are essentially in agreement with Eq. (18) and Eq. (19).

In *TEAPOT*, since the wiggler section is treated as infinitely thin and at the center of the range, only the deflection terms, $\mathcal{R}_{2i}$, $\mathcal{R}_{4i}$, $\mathcal{T}_{2ij}$, $\mathcal{T}_{4ij}$ have been incorporated. (It can be seen mentally, from Eq. (17), that the off-diagonal terms $\mathcal{R}_{12}$ and $\mathcal{R}_{34}$ account for the drift sections preceeding and following the thin element.) The neglected $\mathcal{T}_{1ij}$, $\mathcal{T}_{3ij}$ terms (the largest being $\mathcal{T}_{134}$) are numerically small for typical lattice locations, but they could conceivably be important if the wiggler were situated at a point of extremely small β.

APPENDIX J

COMPARING AND DEFINING MAGNETIC MULTIPOLES

RHIC/AP/99

June 1996

I. MAD and Teapot and
II. RHIC Measurements and Teapot
G. Trahern, F. Pilat

Introduction The definitions of the magnetic field in MAD and in the standard multipole expansion (SME) used internally by Teapot are different. This note, perhaps for the umpTeenth time, defines the relationship between the two field definitions, assuming that they are in fact referring to the same physical quantity. After this discussion we define the relationship between RHIC's measured multipole coefficients and those of Teapot.

B Field in MAD

The magnetic field in MAD is defined by a Taylor series expansion along the x axis as[1]

$$B_y^{MAD}(x,0) = \sum_{n=0}^{N_{max}} \frac{B_n^{MAD} x^n}{n!} \tag{21}$$

The strength of a multipole, K_n, is defined to be

$$K_n = \frac{B_n^{MAD}}{p_o/e}, \tag{22}$$

and thus B_n^{MAD} can be computed as

$$(\frac{\partial^n B_y^{MAD}}{\partial x^n})_{x=y=0}.$$

B field according to the Standard Multipole Expansion

The true magnetic field of a physcial magnet can be described by a field strength $B_x^{True}(x,y,z), B_y^{True}(x,y,z)$. The thin element model then expresses this true field strength in terms of nominal field strengths $B_x(x,y), B_y(x,y)$ as

$$\int [B_y^{True}(x,y,z) + iB_x^{True}(x,y,z)]dz = L[B_y(x,y) + iB_x(x,y)] \tag{23}$$

where the integral is taken over the length of the magnet. The standard multipole expansion for B_y and B_x is then given by[2]

$$(LB_y) + i(LB_x) = (LB_o) \sum_{n=0}^{N_{max}} (b_n + ia_n)(x+iy)^n. \tag{24}$$

where we specialize to the case of a dipole. So B_o is the dipole field strength at the origin. N_{max} is the highest order of multipole in the series expansion. Defining $R_n + iI_n = (x+iy)^n$ we can re-write the field components in the SM expansion as

$$\tilde{B}_y = \frac{LB_y}{p_o/e} = \sum_{n=0}^{N_{max}} \tilde{b}_n R_n - \tilde{a}_n I_n \tag{25}$$

$$\tilde{B}_x = \frac{LB_x}{p_o/e} = \sum_{n=0}^{N_{max}} \tilde{b}_n I_n + \tilde{a}_n R_n \tag{26}$$

where

$$\tilde{a}_n = \frac{LB_o}{p_o/e} a_n, \quad \tilde{b}_n = \frac{LB_o}{p_o/e} b_n. \tag{27}$$

The scaling factors above are conventional, but the magnetic fields $(\tilde{B}_x, \tilde{B}_y)$ so defined are just the deflections that particles experience passing through the field, and the coefficients $(\tilde{a}_n, \tilde{b}_n)$ are the ones used directly by Teapot. The factor of p_o/e is often referred to as "$B\rho$". The following table[2] gives some

explicit examples of the multipole expansion. Also note the traditional jargon used there, e.g., $\Delta\theta$ is the "bend angle", f is the "focal length" and S is the "sextupole strength", etc.

Relation between MAD K_n and SME b_n

Assuming that the field strength B_y is the same physical quantity in either representation, we want to find how K_n is related to b_n. Using the definition in terms of partial derivatives and noticing that

$$(\frac{\partial^n I_n}{\partial x^n})_{x=y=0} = 0$$

$$(\frac{\partial^n R_n}{\partial x^n})_{x=y=0} = n!$$

we see that

$$K_n \equiv (\frac{B_n^{MAD}}{p_o/e}) = L^{\perp 1} n! \tilde{b}_n = \frac{B_o}{p_o/e} n! b_n . \tag{28}$$

The integrated strength is

$$K_n \cdot L = \frac{B_o L}{p_o/e} n! b_n = \frac{B_o L}{B\rho} n! b_n \tag{29}$$

and in terms of $\tilde{b}_n$,

$$K_n \cdot L = n! \tilde{b}_n . \tag{30}$$

The above K_n apply to the case of standard elements of length L. If one is instead talking about MAD's multipole element which is defined to have zero length, replace $K_n \cdot L$ in Eq. (30) by Kl_n to get the corresponding MAD multipole notation.

Skew components in MAD and SME

The definition of the magnetic field from the MAD documentation in Eq. (21) explicitly excludes skew multipole moments, so it is not possible to derive a

	n	R_n	I_n	$\tilde{b}_n$	$\tilde{a}_n$	$\Delta x' = -\tilde{B}_y$	$\Delta y' = \tilde{B}_x$
Horizontal bend	0	1	0	$\Delta\theta_x$	0	$-\Delta\theta_x$	0
Vertical bend				0	$\Delta\theta_y$	0	$\Delta\theta_y$
Erect quadrupole	1	x	y	$q = 1/f$	0	$-qx$	qy
Skew quadrupole				0	$q_s = 1/f_s$	$q_s y$	$q_s x$
Erect sextupole	2	$x^2 - y^2$	$2xy$	$S/2$	0	$-\frac{S}{2}(x^2 - y^2)$	$\frac{S}{2}2xy$
Skew sextupole				0	$S_s/2$	$\frac{S_s}{2}2xy$	$\frac{S_s}{2}(x^2 - y^2)$
Erect octupole	3	$x^3 - 3xy^2$	$3x^2y - y^3$	$O/6$	0	$-\frac{O}{6}(x^3 - 3xy^2)$	$\frac{O}{6}(3x^2y - y^3)$
Skew octupole				0	$O_s/6$	$\frac{O_s}{6}(3x^2y - y^3)$	$\frac{O_s}{6}(x^3 - 3xy^2)$
Erect decapole	4	$x^4 - 6x^2y^2$	$4xy(x^2 - y^2)$	$D/24$	0	$-\frac{D}{24}(x^4 - 6x^2y^2 + y^4)$	$\frac{D}{24}4xy(x^2 - y^2)$
Skew decapole		$+y^4$		0	$D_s/24$	$\frac{D_s}{24}4xy(x^2 - y^2)$	$\frac{D_s}{24}(x^4 - 6x^2y^2 + y^4)$

Table 1: Deflections, $\Delta x', \Delta y'$, caused by standard magnets and notation for their strengths

relation between MAD's way of defining skew multipole elements and SME's a_n directly unless one looks into the Teapot code itself. We have done this (with the help of R. Talman). However, one can perhaps see the result on physical grounds if one considers only a single multipole. One can 'convert' an erect multipole of order n into its corresponding skew element by a rotation of the erect element around the longitudinal direction by its natural symmetry angle of $\pi/(2n+2)$. In this way a pure erect multipole of order n becomes a pure skew multipole of the same order. So one can conclude that a the strength of a skew element which is defined in MAD by specifying a multipole of strength $Kl_n = value$ and T_n (without a value) is related to Teapot's a_n as

$$(Kl_n)^{skew} = \frac{B_o L}{p_o/e} n! a_n = \frac{B_o L}{B\rho} n! a_n, \tag{31}$$

and

$$(Kl_n)^{skew} = n! \tilde{a}_n. \tag{32}$$

(If one is instead interested in skew magnets of finite length, then replace Kl_n above by $K_n \cdot L$ as discussed earlier for the non-skew case.)

0.2. Aside: How *does* Teapot do it?

For those who might want to know precisely how Teapot transforms the MAD input specification of multipole strength into the SME form, here is an example of the (old) Fortran code that was used to do this transformation.(The C++ version of Teapot uses an equivalent formulation.) The example shown below is for MAD's octupole and general multipole. The relevant lines to focus on have been indicated with arrow marks. The explanation follows the listing.

```
ELSEIF (itype .EQ. 7) THEN
C           ---- "octupole"
nmax(ikelem) = 3
-->       el = pdata(idp)
```

```
thklen(ikelem) = el
-->       val = pdata(idp + 1)*el/6.
*-->*      ang = pdata(idp + 2)*4.
-->       btw(3, ikelem) = val*cos(ang)
-->       atw(3, ikelem) = val*sin(ang)
typeaper(ikelem) = pdata(idp + 3)
xapsize(ikelem) = pdata(idp + 4)
yapsize(ikelem) = pdata(idp + 5)
xoffset(ikelem) = pdata(idp + 6)
yoffset(ikelem) = pdata(idp + 7)
mxstreng(ikelem) = pdata(idp + 8)
ELSEIF (itype .EQ. 8) THEN
C           ---- "multipol"
DO i = 1, 9
-->         val = pdata(idp)/fact(i)
*-->*        ang = pdata(idp + 1)*(i + 1)
idp = idp + 2
-->         btw(i, ikelem) = val*cos(ang)
-->         atw(i, ikelem) = val*sin(ang)
IF (iptyp(idp - 2) .NE. -1) nmax(ikelem) = i
ENDDO
typeaper(ikelem) = 0
```

Note: The variables $atw(n,-), btw(n,-)$ in the above code have exactly the same meaning as $\tilde{a}_n, \tilde{b}_n$ in Eq. (27).

First we discuss the octupole case. In the above code *pdata* is the array containing all the information gleaned by MAD's parser from the original standard input file. *el* is the length L of the octupole. *val* is a local variable which equals $K_n * L/3!$. *ang* is another local variable which in the case where one just specifies $TILT$ without an argument is the default roll angle of $\pi/8$ times 4. (This

is the mysterious point. We only show how Teapot does this transformation, we don't explain the basis for it). One can see that $4 * \pi/8 = \pi/2$, and thus $btw(3,-) = 0, atw(3,-) = val = K_n * L/3!$.

Now that we've done the octupole case, consider the thin multipole case that follows it. The meaning of the code variables is the same as in the octupole case. Since thin multipole have no length, the code is simpler in some respects. The variable ang specifies how the roll angle determines the strength of the skew element. For example, in the case where one takes the default roll angle of $\pi/2i+2$ for a skew multipole of strength Kl_i, one states T_i without an argument in the input file. Then the value of $ang = pdata(idp+1)*(i+1) = (\pi/2i+2)*(i+1) = \pi/2$. This leads to values of $btw(i,-), atw(i,-)$ of 0 and $pdata(idp)/fact(i) = Kl_i/i!$, respectively, in agreement with Eq. (32).

Relating RHIC's Measured Multipole Coefficients to those of Teapot

The field expansion in Eq. (24) actually applies in general only to dipoles since the central field value B_o vanishes (or at least ought to) for other types of magnets such as quadrupoles, sextupoles, etc. So one must adopt a different but analogous convention for other magnet types. In chapter two of Ref. 2 there is a clear discussion of one way to do this for the case of quadrupoles, and we can compare that with the way RHIC describes a general magnet. The multipole expansion for a quadrupole magnet from Ref. 2 is

$$(LB_y^Q) + i(LB_x^Q) = (L\frac{\partial B_y^Q}{\partial x})[x + iy + 10^{\perp 4}\sum_{n=2}^{N_{max}}(b_n^Q + ia_n^Q)\frac{(x+iy)^n}{R_r^{n\perp 1}}] \qquad (33)$$

where R_r is the reference radius where the measurement is made, and along with R_r, the factor of $10^{\perp 4}$ is chosen so that a_n^Q, b_n^Q are of order 1 for "bad", low order multipoles. The prefactor, in this case the field gradient, $(\partial B_y^Q/\partial x)_{x=y=0}$, serves the same purpose as B_o in Eq. (24).

In general for every type of magnet, there is a formula of this type. The prefactor like $B_o(\partial B_y^Q/\partial x)$ in the case of dipoles(quadrupoles) sets the scale

so that the coefficients $a_n, b_n (a_n^Q, b_n^Q)$ represent fractional deviations from the measured field strength. A similar analysis can be done for the other types of magnets. To summarize, the normalization of multipole coefficients via Eq. (33) requires knowing the behaviour of the field at the origin.

In contrast to Eq. (33) RHIC has used a slightly different form to represent the multipole expansion for a general magnet. According to our sources Refs. 3,4,5,and 6, there is uniform strategy for every type of magnet that is representative of the way the magnets are actually measured. In the following we will assume that the local coordinate system of Teapot and the magnetic measurement system are the same. If this is not true, for example, if the magnet is oriented differently in the lattice compared to the way it was measured, appropriate modifications to the sign of the coefficients will need to be made Ref. 6. With this caveat, the RHIC convention for a magnet's multipole expansion is Ref. 5.

$$(LB_y) + i(LB_x) = L\,B(R_r)[10^{\perp 4} \sum_{n=0}^{N_{max}} (b_n^M + i a_n^M) \frac{(x+iy)^n}{R_r^n}] \qquad (34)$$

where the superscript M in a_n^M, b_n^M denotes the fact that these are *measured* multipole coefficients. $B(R_r)$ is a normalization factor. This normalization is chosen so that the magnitude of the term of order k in the expansion, $|b_k^M + i a_k^M| = 10^4$ for a magnet with multipolarity $2(k+1)$. Consequently the multipole coefficient, b_k^M, for a "normal" or "upright" magnet of order k is 10^4. I.e., b_0^M for dipoles is 10^4, b_1^M for quadrupoles is 10^4, and similarly for skew magnets so that for a skew quadrupole a_1^M would be 10^4.

Since RHIC normalizes its multipole coefficients in this way, comparison with an expression like Eq. (33) for a specfic kind of magnet can be obtained by evaluating Eq. (34) along the x axis near the origin. We will do this exercise in the appendix, but it is not actually necessary. Teapot only requires that the magnetic field be brought to a form like Eq. (24). Eq. (34) is already in this

form, so making the correspondence with Teapot is straightforward up to possible reversals in sign that are discussed in RHIC/AP/95, Ref/ 6 and summarized in the next section.

The factor $L\,B(R_r)$ on the right hand side of Eq. (34) is measured at a fixed current by the magnetic measurement group of RHIC and quoted as the Integral Transfer Function or ITF, i.e., $ITF\cdot I = L\,B(R_r)$, where I is the current in kA at which the measurement was made. The reference radius, R_r, is also given for each measurement.

The $\tilde{a}_n, \tilde{b}_n$ of Teapot are recovered from the above *measured* expansion coefficients in analogy to Eq. (27) by

$$\tilde{b}_n = (\frac{ITF\cdot I}{p_o/e})\frac{10^{\perp 4}}{R_r^n} b_n^M \tag{35}$$

$$\tilde{a}_n = (\frac{ITF\cdot I}{p_o/e})\frac{10^{\perp 4}}{R_r^n} a_n^M \tag{36}$$

where the b_n^M, a_n^M are the measured multipole coefficients, p_o/e is $B\rho$, and I is the current at which the measurement was made in kA.

In some cases, particularly for dipoles, the RHIC magnetic measurements group does more detailed measurements of the magnetic multipoles. They measure them at the body center as well as the return and lead ends of the magnet. If this group of measurements is available, a different form of the Teapot coefficients is needed since the physical dimension of the measured multipole coefficients are different for the body and end data.

If Body measurements exist, we specify body $\tilde{b}_n^{Body}, \tilde{a}_n^{Body}$ for Teapot as

$$\tilde{b}_n^{Body} = (\frac{BTF\cdot I}{p_o/e})(\frac{ITF}{BTF})\frac{10^{\perp 4}}{R_r^n} b_n^{M\perp Body} = (\frac{ITF\cdot I}{p_o/e})\frac{10^{\perp 4}}{R_r^n} b_n^{M\perp Body} \tag{37}$$

$$\tilde{a}_n^{Body} = (\frac{BTF\cdot I}{p_o/e})(\frac{ITF}{BTF})\frac{10^{\perp 4}}{R_r^n} a_n^{M\perp Body} = (\frac{ITF\cdot I}{p_o/e})\frac{10^{\perp 4}}{R_r^n} a_n^{M\perp Body} \tag{38}$$

where BTF is the body transfer function with dimension $Tesla/kA$, and the

superscript $M-Body$ refers to "measured Body". The factor of ITF/BTF has dimension of length in $meters$ and is needed to scale BTF so that it has the dimensions of an integral transfer function since $b_n^{M\perp Body}, a_n^{M\perp Body}$ are dimensionless.

If End measurements exist, then the lead and return end coefficients for Teapot are given by:

$$\tilde{b}_n^{End} = (\frac{BTF \cdot I}{p_o/e})\frac{10^{\perp 4}}{R_r^n} b_n^{M\perp End} \tag{39}$$

$$\tilde{a}_n^{End} = (\frac{BTF \cdot I}{p_o/e})\frac{10^{\perp 4}}{R_r^n} a_n^{M\perp End} \tag{40}$$

where BTF is again the body transfer function referred to above, and note that in this case since the dimension of $b_n^{M\perp End}, a_n^{M\perp End}$ is in $meters$, only BTF rather than ITF is needed.

Afterward on Sign Conventions for Multipole Coefficients RHIC magnets are measured in a standard way, i.e., the lead end of each magnet is oriented with respect to a local magnet coordinate system in the same way during the measurement process. Thus the measured multipole coefficients are directly tied to the local measurement coordinate system's orientation.

During installation in the tunnel a magnet may need to be rotated by π radians around the Y axis relative to the coordinate system in which it was measured either for physics or mechanical/installation reasons. In these cases the sign of some multipole coefficients used in Teapot will need to change (relative to their signs in the measurement database) to properly model the dynamics in the global coordinate system used by Teapot. The nature of these sign changes has been explained in Ref. 6, and we will not reproduce their detailed analysis here. However, for purposes of keeping the definitions of Teapot multipole coefficients in terms of RHIC's measured values all in one place, we include the necessary rules here. We thank Fritz Dell for the following formulation of these rules.

The rules require an understanding of a magnet's "orientation". A magnet's orientation is defined to be positive if a positive displacement relative to the horizontal closed orbit corresponds to a positive horizontal displacement with respect to the *magnet local coordinate system* discussed above. Otherwise the orientation is negative. See Ref. 6 for a clear statement of the definition of the local magnet measurement coordinate system and its relation to the lead and non-lead ends of the magnet.

1. For Normal magnets whose main multipole is even (dipoles, sextupoles, etc.), or for Skew magnets whose main multipole is odd (quadrupoles, octupoles, etc.)
   - Positive orientation: use b_n^M, a_n^M as is.
   - Negative orientation: change sign of b_n^M with *odd* n, and change sign of a_n^M with *even* n.

2. For Normal magnets whose main multipole is odd (quadrupoles, octupoles, etc.), or for Skew magnets whose main multipole is even (dipoles, sextupoles, etc.)
   - Positive orientation: use b_n^M, a_n^M as is.
   - Negative orientation: change sign of b_n^M with *even* n, and change sign of a_n^M with *odd* n.

Acknowledgements

The authors would like to thank Richard Talman for many helpful and definitive conversations. And we would also thank Animesh Jain for both carefully reading and correcting sections six and seven.

Appendix Relating $B(R_r)$ to the Field at the Origin

The normalizing factor $B(R_r)$ can be related theoretically to the value of the field at the origin in the following way. From Eq. (34) the value of the integrated field at $y = 0$ is

$$(LB_y + iLB_x)|_{y=0} = L\,B(R_r)[10^{\perp 4}\sum_{n=0}^{N_{max}}(b_n^M + ia_n^M)\frac{x^n}{R_r^n}] \tag{41}$$

For a "normal" magnet of order $k, b_k^M = 10^4, a_k^M = 0$. Taking partial derivatives k times, we have

$$\frac{\partial^k(LB_y)}{\partial x^k}|_{x=y=0} = k!(LB(R_r))10^{\perp 4}\frac{b_k^M}{R_r^k}. \tag{42}$$

Noting that $b_k^M = 10^4$ in the case of a normal magnet of order k, we find

$$B(R_r) = (\frac{\partial^k B_y}{\partial x^k})|_{x=y=0}\frac{R_r^k}{k!}. \tag{43}$$

For the case of a skew magnet of order k a similar analysis yields

$$B(R_r) = (\frac{\partial^k B_x}{\partial x^k})|_{x=y=0}\frac{R_r^k}{k!}. \tag{44}$$

References

1. C. Iselin, "The MAD Program, version 8.1 documentation."
2. R. Talman, "Accelerator Mathematics", Laboratory for Nuclear Studies, Cornell University., Ithaca, N.Y. 1994.
3. J. Wei, F. Dell, private communications.
4. A. Jain, private communication.
5. R. Gupta, "Estimating and Adjusting Field Quality in Superconducting Accelerator Magnets", RHIC/AP/87, 1996.
6. P. Wanderer, A. Jain, S. Peggs, F. Dell, J. Wei, D. Trbojevic, "RHIC Magnetic Measurements Definition", RHIC/AP/95, 1996.