TEXT FOR UAL ACCELERATOR SIMULATION COURSE

N. Malitsky and R. Talman

Abstract. ual (Unified Accelerator Libraries)[**4**] is an accelerator simulation environment whose purpose is to *homogenize* diverse simulation codes. This document is available at `http://www.ual.bnl.gov`, where it can be expected to be updated occasionally. Its initial purpose has been to serve as instructions for the 2005 USPAS (U.S. Particle Accelerator School) course held in Ithaca. As such, some of the material, such as filenames, filename extensions, tutorials, and xml tools, are specialized unnecessarily to what happened to be in use for this course. The document is intended to serve also as a *UAL Physics User's Guide* for the ual environment. Some of the other documents and user guides referred to are available at the same web site. The mad8 manual is especially important since, to the extent possible, geometry, terminology and definitions of ual are adopted from that source. Some correction algorithms, such as orbit smoothing and local decoupling are documented in the teapot manual.

This text complements the *UAL User Guide,*[**5**] which, though now largely outdated, describes much of the motivation, organization and evolution of ual. The main ways in which the User Guide is outdated are that the user interface has been migrated from perl to c++ and a graphical user interface is now supported. Updating of the User Guide, now in progress, will consist primarily of the conversion of line-by-line perl script explanations to line-by-line explanations of essentially equivalent c++ code. The physics underlying the code will be largely unchanged.

Contents

CHAPTER 1

Introduction

1.1. Code Installation From the CDROM's

This text is meant to resemble a physics text more nearly than it resembles a cookbook. But this section is an exception; it gives rudimentary instructions on initializing the UAL computer environment. The instructions are written as if for a student in the USPAS school held at Cornell University, in Ithaca, New York, in 2005, but they should be applicable to anyone else looking for an intensive introduction to UAL.

This documentation assumes that the username is `ualusr`, with home directory `/home/ualusr`. In these instructions this directory is referred to as ∼. LINUX is the only operating system used in the course. (For the USPAS school, RedHat Enterprise linux is used.) Everything needed over and above code included in the (complete) RedHat release is contained on two CDROM's, which are labeled `USPAS` and `TOY-LATTICES`. The code they contain installs into subdirectories `~/USPAS` and `~/TOY-LATTICES`. Installation instructions are given on the CDROM labels. They can be installed in either order. The `USPAS` code uses the `TOY-LATTICES` code only as a source of example lattice files. Though the `ADXF` lattice format was introduced initially in UAL, in principle it exists in a more-general-than-UAL world. The `TOY-LATTICES` codes process lattice files in that world. But these instructions assume that both CDROM's are being installed at the same time.

The instructions include copying a file called `cshrc-tentative` to `~/.cshrc`. But this must be done cautiously. The purpose for this file is to establish required environment variables and search paths. To avoid unexpected conflicts it is recommended that the `ualusr` account be dedicated entirely to UAL work. The user wishing to personalize the `.cshrc` file is responsible for protecting and merging its contents appropriately before blindly following the CDROM label's instruction to over-write this file.

A partial directory structure (with directory labels giving hints concerning content) follows:

- `USPAS`
  - `cshrc-tentative`, to be copied to `~/.cshrc`
  - `rootrc`, to be copied to `~/.rootrc`
  - `ual1`, the main UAL code.
    - `gui`
    - `codes`
    - `env`
    - `ext`
    - `tools`
    - `examples`

    - `doc/adxf`, location of local copu of `adxf.xsd` schema
  - `examples`, simulations, loosely coupled with chapters
    - `transverse`, for transverse simulations
    - `longitudinal`
    - `nonlinear`
    - `decoherence`
  - `setup-linux`, initialization file
  - `tools`,
    - `Instructions.ps`, extracted from `http://www.ual.bnl.gov`
    - `qt`, Troll Tech's `Qt` graphical user interface
    - `root`, CERN, object-oriented, data-processing environment.
    - `SoQt`, library integrating `Coin` and `Qt`
    - `simage`, support for loading and saving images, `coin3D` development
    - `Coin`, 3D graphics rendering library
- `TOY-LATTICES`
  - `oxygen5`, inexpensive commercial, XML-aware editor, licensed for US-PAS school
  - `grace`, for `xmgrace` graphical post-processing
  - `xslt`
    - `MAD8`, for lattice function comparisons
      - `OUTPUT`, lattice function graphs from MAD8 runs
    - `SCRIPTS`, XSLT scripts for translating among various file formats.
    - `ADXF`
      - `INPUT-adxf`, *primary* location of lattice files
      - `OUTPUT-adxf-num`, purely numerical, `.adxf` files
      - `OUTPUT-specialized`, specially tailored files, see `README`
      - `OUTPUT-xsl`, files prepared for transformation by XSLT
      - `OUTPUT-sxf`, purely numerical, `.sxf` files
      - `OUTPUT-mad`
      - `OUTPUT-tpot`
      - `INPUT-sif`, archive of original MAD lattice files
      - `INPUT-xsl-archive`, archive of original `.xsl` files
    - `ADXF2.10`, schema, and documentation thereof
    - `tools`
      - `java`
      - `Xerces`
      - `xml-xalan`

Apart from containing the toy lattice files to be used in the school, the code in `~/TOY-LATTICES` consists mainly of non-UAL-specific utility codes useful in translating lattice files among various file formats, and for validating the results, as regards both XML and physics. Much of this code is transitory, intended especially for the USPAS course, or provided for backward compatibility—for example by providing `.sxf` files that can be used in some old simulations, or `.mad` or `.tpot` files.

The primary repository for files to be used as input to UAL simulations is

```
~/TOY-LATTICES/xslt/ADXF/INPUT-adxf
```

which contains `.adxf` files for all toy-lattices, with variables and expressions unevaluated. However, some files, such as `ags`, support parameter evaluation only via the `ags.xsl` route. *All* toy lattice files are therefore available with all variables evaluated to numbers in

`~/TOY-LATTICES/xslt/ADXF/OUTPUT-adxf-num`

UAL simulations can be started from `.adxf` files in either of these directories, with or without modification. Though these files have `.adxf` extensions, they are to be processed exactly as if their extensions were `.xml`. They are required to be XML-valid against the schema,

`http://www.ual.bnl.gov/adxf/adxf.xsd`

A local copy of the schema, needed for when the web is inaccessible, is

`/USPAS/ual1/doc/adxf/adxf.xsd`

The `.adxf` files can be modified using any editor, but the XML-aware editor provides many valuable utilities. Especially useful are context-aware pull-down menus, that present all schema-legal options. Files can be tested for XML-*well-formedness* and *validity.* Files can also be *pretty-printed* and stepped through for line-by-line feedback, for debugging or for browsing purposes. Obviously it is sensible to save copies of the originals. They can also be regenerated using the scripts described next. This route must be taken if comparison output from MAD8 for a modified lattice is desired.

For backward compatability all toy lattice files are also available in `.sxf` form in the directory

`~/TOY-LATTICES/xslt/ADXF/OUTPUT-sxf`

These files can be used as input to various (old and now unsupported) PERL-controlled batch-mode scripts. Since this interface is in the process of being replaced there is little support available in case of malfunctioning of these scripts.

The scripts in the `SCRIPTS` directory perform various lattice description transformations, such as populating all the lattice file directories mentioned so far. Some of the scripts combine several separate manipulations. If only one of these manipulations is required the appropriate line (a shell script statement) should be extracted and run by itself. In all cases the transformations can be stepped through using the `oxygen` XML-aware editor. A few of the scripts (which call some of the others) are:

- `process-qfile`: taking a single argument, such as `racetrack`, this script processes the toy lattice `racetrack.adxf`, and produces equivalent files in diverse formats, such as `racetrack.mad`, and `racetrack.sxf` (now deprecated). These are in forms that can be (and are) immediately processed. Also produced (for convenience in performing XSLT transformations) are `racetrack.xsl` and (in a different directory) `racetrack.adxf` which is a purely numerical version with all variables and expressions evaluated to pure numbers.
- `process-all-files`: runs `process-qfile` for all available `.adxf` files.
- `check-results`: for a particular lattice, such as `racetrack`, checks for the presence of files generated by `process-qfile` and validates the `.adxf` file against its XML-Schema.

All these tools are available for any lattice file written in `.adxf` form. The OXYGEN, XML-aware editor makes the generation of such a file straightforward, by presenting, at every step, only legal options for the developer to choose among. The only major task in importing a new accelerator into this environment is the initial generation of its `.adxf` form. The XML-aware editor is helpful, but the task is about as tedious as generating a lattice description file for any accelerator analysis software.

1.2. Organization of the Text and Course

This text has been written primarily to provide reference material in support of a course on computer simulation of accelerators. Since the choice of subject matter is rather conventional for the field, the material would be more or less appropriate for any accelerator modeling software. Many of the problems are this general, but we think of this text as being a kind of UAL *Physics User's Guide*. Almost all of the simulations assume the UAL environment is being used. Furthermore, many of the figures accompanying the theoretical material in the text is produced by the UAL GUI.

The UAL environment is intended to be useful for both off-line models and for the models used in online control system applications. Only the off-line applications are documented in this text.

Nearly everthing of importance in accelerator physics can be subjected to study by simulation. From the time that a tentative lattice for a new accelerator has been written down, then refined and the accelerator built and commissioned, until the accelerator is eventually de-commissioned, simulation can contribute to the understanding and improvement of the accelerator's performance. A long, yet still incomplete, list of important ingredients of a simulation follows:

- initial lattice description
- Monte Carlo particle bunch and field error assignment
- calculation of lattice functions, both ideal and real
- correction capabilities, orbit smoothing, decoupling, *etc.*
- simultaneous presence of multiple effects
- determination of bunch evolution, emittance dilution, particle loss, halo generation, injection and extraction efficiency.
- collimation design
- space charge, beam-wall, and beam-beam effects
- design and performance modeling of feedback and control

Some of the essential requirement of the physics underlying a simulation are:

- correct basic physics (e.g. symplecticity and Maxwell equations)
- sensible inclusion of (only) essential physics of sufficient generality
- freedom from bugs, blunders, and conceptual errors

Though almost too obvious to write down, the third of these requirements is hard to achieve, and achieving it depends strongly on the first two, between which there is a kind of complementarity. The striving for unreasonably faithful description tends to increase complexity which makes the code more error-prone. This justifies the expenditure of much effort in choosing what idealizations are to be adopted.

To be most effective (like all theory) simulation is best used in conjunction with experimental observations on a real accelerator. If the model underlying a simulation is *too* idealized, that fact should become rapidly apparent during actual machine studies.

For avoiding bugs the quality of the code architecture and the careful application of tests are probably the most important determinants of the code reliability. In many quarters it is thought that object-oriented code, with its highly disciplined interfaces, is favorable in this respect. UAL is based (almost) entirely on C++ which *is* object-oriented. Amongst other features, this facilitates extensibility and maintainabily. These issues are discussed in detail in other UAL documents. For example, further discussion of the merits of C++, and why it is being superceded in UAL is discussed in the User Guide.

Another likely source of error is the incorrect interpretation of correct code. This may be caused by incorrect interpretation of physics or by computational issues. In this text much emphasis will be placed on the physics, but, in spite of its importance, the contribution of software architecture to code reliability is little discussed. On the other hand the benchmarking of simulation results against independently derived and trusted results might almost be called the *theme* of the present text.

It is not enough for a simulation just to have correct physics and sensible models. It is also necessary for it to be sufficiently user friendly that it actually gets used. The user interface has to display its results in an accessible fashion, to provide the rapid feedback needed to support rapid changing of conditions. An all too common experience with many (all?) existing accelerators is that there is one or more detailed but hard to use offline models and a crude, but easy to use, online model. From experience one knows which of these two models will actually be used in the control room. This scenario be mainly due to the difficulty of providing good user interfaces. Of course there are also other reequirements for a simulation, such as performing peripheral calculations and providing post-processing tools.

So what is the relation of this text to simulations? It might seem appropriate to provide derivations for all formulas used in the code. Though considerable technical material of this sort is included, only a relatively small fraction of the formulas in the UAL code are derived here. Far more important is the task, mentioned already, of result corroboration. It is important for material supporting such tests to be available. Much of this text is therefore devoted to developing idealized models and to deriving analytic formulas describing them.

An all too prevalent practice in accelerator investigations is to accept uncritically the results of this or that computer program. Based on the likely valid assumption that the program's author knows more than the user, the results of the program are accepted as being reliable. Apart from the possibility of bugs, this approach is likely to mask the presence of built-in assumptions that are valid in some circumstances but not in others. It is very rare for a user to have line-by-line familiarity with the code he or she is relying on, but to reduce the likelihood of error it is essential to have a fundamental understanding of what has gone into the code.

Summarizing what has been said, to the extent that this text is an "Accelerator Physics Users Guide", it is intended to provide:

- an overview of the subject of accelerator physics, in support of the enlightened usage of simulation code, especially UAL
- exploration of simplifying idealizations
- emphasis on keying simulation to actual machine studies

- detailed technical information emphasizing methods appropriate for simulation
- test cases that can be used to test simulation results against theoretical results and therefore provide confidence in results obtained in situations too complicated for analytic treatment
- pedagogical material for a course on the subject
- practice assignments to help a user gain experience with the methods and the code

1.3. Assignments: Tutorials, Problems, Simulations

There are assignments sprinkled more or less uniformly through the text. For easy visability all these assignments are printed in italic type and separated from the main text by horizontal lines. The assignments are of three different types, referred to as *tutorials*, *problems*, and *simulations*.

Tutorials. Tutorials are intended more to guide the student through practicalities of the UAL code than to explore accelerator physics. They are not intended to be difficult. If a tutorial assignment *is* difficult it may be because object-oriented software architecture is unfamiliar or because of the obscurity or absence of documentation. In any case the difficulty probably reflects more on the software than on the student. Tutorials occur mainly in the early chapters. They can only be performed while sitting in front of a computer screen.

Problems. As mentioned in the previous section, the attempt is made in this text for each topic to be addressed by two parallel methods—one analytic, one by simulation. The *problems* relate to the analytic members of these pairs. Much like the problems in any book discussing theoretical methods these problems are intended to exercise the student's understanding of the material. In many cases the problems extend the treatment in the text. Whether a problem is hard or easy depends (obviously) on the student's level of familiarity with the particular topic. When the purpose of a problem is to derive a result for comparison with simulation the answer is usually given in the statement of the problem. This may allow the result to be used even without the problem having been worked. In other words, it is not necessary to slavishly work through all the problems—they are, after all, intended primarily as being complementary to the simulations. That said, it bears repeating, that one of the most important initial uses of any simulation is to corroborate its results against known results, usually obtained analytically. Mainly the problems are to be worked out with pencil and paper.

Simulations. The ultimate purpose for UAL is to simulate the behavior of real accelerators and storage rings. As in all science, in controlling and observing an accelerator, there is tension between what is expected and what is observed. By increasingly realistic simulation of what is expected, and reconciliation with what is observed, the performance of an accelerator can be improved. The UAL *simulations* attempt to model accelerator performance in conditions that are as nearly realistic as possible, consistent with avoiding undue complexity. The various simulations apply to an accelerator lattice supplied by the user. When the lattice is one of the so-called "toy" lattices the results are intended to be relatively easy to interpret and to compare with theory. When the lattice file describes a real accelerator the results will be more realistic but less easily checkable.

Documentation of the dynamic simulations is scattered through this text (and in other online material). Here it will only be stated that they are launched by instructions like

```
$ cd ~/USPAS/examples/longitudinal/linux
$ ./run
```

Actually this does not start a run; rather it echoes a usage message such as,

```
$ usage: run ringName latticeFile apdfFile
```

Many lattice files describe only one beamline, but the "ringName" argument is present in case the file "latticeFile" describes more than one beamline. The final argument refers to a "propagator description file" called, for example, `tibetan.apdf`.[1] Reminded by this hint, one types

```
$ cd ~/USPAS/examples/longitudinal/linux
$ ./run ring ~/TOY-LATTICES/xslt/ADXF/INPUT-adxf/eq_tune_fodo.adxf ../data/tibetan.apdf
```

(This assumes the directory structure is such that the files are where they are stated to be. The directories shown are the preferred directories for the USPAS school. The files can be referenced by absolute or relative addresses.) This command brings up a GUI that permits beam conditions and other parameters to be changed from default values. Other features of the simulation can be tailored in the GUI and ouput in the form of plots or files can be requested. First the GUI-available parameters are tailored as desired. Then clicking on `setup` followed by clicking on `run` starts the run. Beam plots are updated at regular (adjustable) intervals. If desired, the run can be paused. While viewing a plot, hard copy can be produced and, optionally, its data saved for post-processing. Parameters not accessible via the GUI are also changeable, but only by re-coding the main control file `/USPAS/examples/longitudinal/src/run.cc`. After editing, this file has to be recompiled using

```
$ cd ~/USPAS/examples/longitudinal/src
$ make
```

before restarting the simulator as above. Similarly, to start the `transverse` simulator,

```
$ cd ~/USPAS/examples/transverse/linux
$ ./run ringName latticeFile apdfFile
```

Other UAL simulations, going by the names `nonlinear` and `decoherence` have similar organization and usage. A systematic listing of simulator properties is contained in Table 1.1.

By their very nature all simulations are somewhat open-ended and the instructions may not be very specific. Generally the simulation will involve changing one or more lattice or beam parameters. In some cases the GUI accepts data entry to change parameter values. For more extensive changes it is necessary to edit the input file as described above. The student is invited, no required, to go beyond the explicit instructions, especially by formulating questions that the software application should be expected to be able to address. Especially encouraged, though very ambitious for a short course, would be to generate an input file corresponding to some existing or planned accelerator, and to simulate its behavior. (Manual translation of an existing MAD lattice description to ADXF is straightforward but

[1] In the case of the `longitudinal` simulation, the RF cavity is under the control of the GUI, so treatment of the RF cavity is not described in the `.apdf` file.

TABLE 1.1. Some Properties of the Simulations.

simulation name	chapter number	variables emphasized	algorithms	functions. displayed
`transverse`	2,3	x, y	matrix, kick	β, D
				phase space
	4		FFT	Q
			SVD	spatial, temporal eigenvectores
`longitudinal`	6	s		separatrix phase space
`decoherence`	7	x, y, s	matrix, kick map	$< x >, < y >, < s >$
`nonlinear`	8	x, y, s	matrix, kick map	Q map dynamic aperture

tedious.) In short, the simulations are the main content of the UAL/USPAS course. Obviously they require a computer, but in most cases they also require a lively understanding of the material, such as may be obtained by reading the text and doing the problems.

1.4. The UAL Element-Algorithm-Probe Simulation Framework

UAL (which stands for "Unified Accelerator Libraries") is an accelerator simulation environment. It differs from some other environments by its rigorous separation of physical *elements* (magnets, cavities, etc.) from the formulas or *algorithms* describing beam evolution through the elements. The quantities being evolved (particle positions, bunches, maps, Twiss functions, etc.) are referred to as *probes*. These are the cornerstones of the so-called *element-algorithm-probe* framework of UAL.

The parameters of the physical *elements* making up an accelerator lattice are contained in a so-called ADXF file (which stands for "Accelerator Description eXchange Format".) This format is capable of describing lattices that range from the simplest possible design lattice to the most complicated, fully-instantiated, operational lattice. It is important for all tools to function consistently and effectively over this full range of complication.

For an actual simulation, after the *probe* quantities to be evolved have been specified, it is necessary to associate a specific evolution *algorithm* with each lattice *element*. These linkages are described by an APDF file (which stands for "Accelerator Propagator Description Format"). This file is usually quite brief, since there are only a few algorithms and default algorithms are usually appropriate for most elements.

The early tutorials concentrate on gaining familiarity with ADXF. Use of APDF is considerably more technical and more specialized. The general idea can be inferred from the following sample, called `tracker.apdf`:

```
<apdf>
  <propagator id="teapot" accelerator="blue">
    <create>
```

```
        <link algorithm="TEAPOT::DriftTracker" types="Default"/>
        <link algorithm="TEAPOT::DriftTracker" types="Marker|Drift"/>
        <link algorithm="TEAPOT::DipoleTracker" types="Sbend" />
        <link algorithm="TEAPOT::MltTracker"
           types="Quadrupole|Sextupole|Multipole|[VH]kicker|Kicker"/>
        <link algorithm="TIBETAN::RfCavityTracker" types="RfCavity"/>
        <link algorithm="AIM::Monitor" types="Monitor|[VH]monitor"/>
     </create>
  </propagator>
</apdf>
```

With the linkages shown, magnets and drifts are handled by `TEAPOT`, r.f. cavities by `TIBETAN`, and monitors are treated as `AIM:Monitor`'s at which particle positions are recorded each turn (for later post-processing). (`AIM` stands for "accelerator instrumentation module".) Users familiar with other accelerator simulation codes may consider it a nuisance that beam positions are available only at monitors. But UAL attempts to be realistic in the sense of making available, and making use of, only data that would be realistically avaiable in a real accelerator.

The purpose of separating elements from algorithms in this way is to support the "mixing and matching" of physical methods (matrix, map, Runge-Kutta, etc.) with physical elements (bends, quads, RF cavities, etc.) As well as facilitating the bench-marking and comparison of methods, this structure permits a simulation to link the most appropriate evolution method with each element. More detailed examples of `.apdf` files, along with more detailed explanations, are given in Chapter 7.

1.5. ADXF

Most users are presumably familiar with MAD lattice descriptions. This form of description is often referred to as SIF which stands for "Standard Input Format". The ADXF format, while incorporating all SIF features, supercedes SIF in three main ways. The most essential of these ways is that ADXF *extends* SIF and is, itself, *extensible.* These extensions include the ability to fully-instantiate the lattice by giving every element its own identity and its own deviations, parameters, etc. The second essential innovation ADXF brings is that it is based on XML. To make the format, as well as any extensions, *self-descriptive*, the XML-Schema discipline is employed. Furthermore, the importation, into the accelerator world, of standard, up-to-date, computer world formalism, makes available tools developed in the vastly-better-developed external world.

Following an innovation in E. Forrest's PTC code, a third, more specialized, feature of ADXF is the distinction between between "uninstalled" ("on the bench" in Forest's terminology) and "installed" elements. Naturally an accelerator contains only installed elements, each potentially having its own positioning and field deviations. As well as being faithfull to reality, this abstraction permits the simulataneous description of more than one ring, including the ability to describe elements that are shared by two or more rings, or that are multiply-traversed (possibly with changed conditions) within a single ring. There are workarounds to provide this feature within SIF. For example, to represent shared elements, a single element can be treated artificially as two distinct elements. But this complicates the inclusion of field or positioning deviations. This can be done consistently, but is error-prone.

Like `.mad` files, `.adxf` files support parameters and expressions and can be pretty much internally self-explanatory. This is especially true for the toy lattices to be used initially.

A considerably more technical overview of ADXF is available at the UAL website `http:www.ual.bnl.gov` under ADXF 2.0. Definitions, examples, schema, and relationships of ADXF to other formats are given there.

In order to serve for both design of idealized lattices and representation of fully-instantiated operational lattices ADXF supports specification of both design parameter values and deviations. This issue is too technical for discussion at this point, but it can be roughly understood by considering ordinary elements such as `sbend`, `quadrupole`, and `marker`. The ADXF fragment

```
<elements>
  <marker name="mk1"/>
  <sbend name="bend" l="lq" angle="deltheta"/>
  <quadrupole name="quadhv" l="lq" k1="kq1"/>
  ...
</elements>
```

essentially just re-expresses MAD input element descriptions as XML. To describe *deviations* (from a design element) `ADXF` uses syntax

```
<elements>
  <sbend name="d0mp08" l="3.58896" angle="-0.0151186"/>
  <element name="bi8-dh0" design="d0mp08"/>
   <mfield b="0 0 0.005476 0.033503"
           a="0.0 -0.010166 0.024366"/>
  </element>
</elements>
```

The <`mfield`> tag also allows a `method` attribute with the default value being `method="set"`, in which case the `a` and `b` entries are absolute values of the parameters. Other possibilities are `method="add"` and `method="multiply"`. In the case of `multiply` an entry `b="1 1.01"` would result in $b_0 \to b_0$ and $b_1 \to 1.01\, b_1$.

1.6. "Toy" Lattices

It is regrettably true that accelerator lattices are complicated. In fact the need for a sophisticated simulation environment like UAL is at least partly due to this complexity. Even professionals, with decades of experience, can be confused as to which components are causing which behavior. A student in a one week course can scarcely, therefore, be expected to generate the lattice descriptions that UAL needs to work with. For this reason, to get started, the "toy lattices" shown in Table 1.2 and FIG 1.1 are to be used as starting points. These lattices are sufficiently detailed to exhibit most of the behaviors important in accelerators. Furthermore, though referred to as "toys", the lattices are parameterized in such a way that they can be generalized to describe accelerators, storage rings and colliding beams of arbitrary circumference, energy, particle type, tunes, and so on.

As they stand, not including `ags.adxf`, which describes the BNL alternating gradient synchrotron, the toy lattices are *thin element* lattices, meaning that the quadrupole and sextupole lengths are negligible compared to the cell length. (To enable comparisons with programs, such as MAD, that do not smoothly incorporate

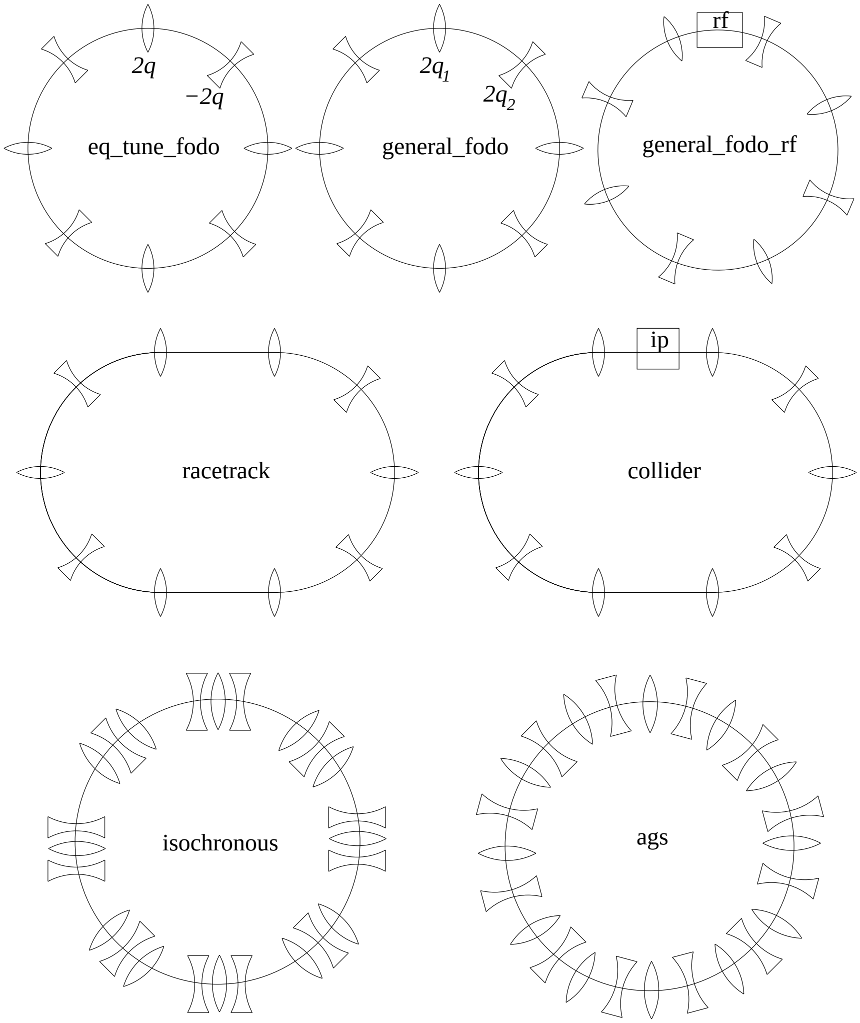


FIGURE 1.1. "Toy lattices" to be used as starting points for accelerator simulations. Note: "ags" is *not* a "toy".

the zero length limit for thin elements, the element lengths in these files are chosen to be not quite zero, but small enough to have negligible effect on the optics) Since the element lengths are expressed as parameters, later on the elements can be turned into thick elements, and the ring retuned.

The early chapters of these notes largely contain introductory exercises intended to provide a gentle introduction to the UAL accelerator simulation environment. The first two tutorials relate to two simple "toy" lattices called `eq_tune_fodo.adxf` and `general_fodo.adxf`. These lattices are designed for getting started on the accelerator modeling course.

These files could have been generated from scratch but some were in fact, derived from `.xsl` files which is a "pre-`adxf`" form. XSL is an XML processing tool. XML stands for "eXtensible Markup Language", which is ideal for modern lattice

TABLE 1.2. "Toy lattices" to be used as starting points of simulations. "P/N/FI" stand for "parameterized/numerical/fully-instantiated".

		INPUT		OUTPUT		
filename	feature	.mad	.xsl	.adxf	.mad	.sxf
`eq_tune_fodo`	equal tunes	P	P	N	N	N
`general_fodo`	unequal tunes	P	P	N	N	N
`general_fodo_rf`	acceleration	P	P	N	N	N
`racetrack`	long straights	P	P	N	N	N
`collider`	low beta IP	P	P	N	N	N
`isochronous`	mom. ind. period	P	P	N	N	N
`ags`	fully realistic	FI-P	FI-P	FI-N	FI-N	FI-N

description. The term "pre-`adxf`" implies that named parameters are allowed and they can be expressed in terms of mathematical expressions and functions. These expressions are "parsed" into numerical expressions using an XML tool known as XSLT. (At this time the UAL parser is also able to parse algebraic expressions so the need for the XSLT expression parser has already been largely eliminated. However the XSLT transformation tools provide powerful help in translating from `.adxf` format to the formats of other accelerator programs.)

Based on XML, the ADXF protocol (or something like it) is beginning to, but has not yet, superceeded SIF (which stands for "Standard Input Format"). Over time SIF has evolved into the MAD (Methodical Accelerator Description) format.

For the UAL simulation course it is the `.adxf` files that are the starting points for the various dynamic simulations. The `.sxf` format (motivated a few years ago by the US-LHC collaboration) was an early prescription for exchanging fully-instantiated lattice descriptions among diverse simulation environments. This format has been superceded by the `.adxf` format, which is extensible, with extensions disciplined by XML *schema*. This makes the format "self-describing".

The `eq_tune_fodo.adxf` file is especially introductory in character and is intended to be superceded by the slightly more general `general_fodo.adxf` after preliminary study. These lattices are parameterized in such a way that they can describe rings of arbitrary radius and arbitrary tunes. The parameters of the input `.xsl` files are intended to be adjusted in performing the tutorials. Later, while performing dynamical simulations using the graphical user interface, the few most important, but not all, parameters will be interactively changeable without recompilation.

When these or other lattices are processed by UAL, various output displays and files are generated. Example output corresponding to the `isochronous` toy lattice is shown in FIG 1.2. When the same lattice is processed using MAD the results are shown in FIG 1.3 and FIG 1.4 are obtained.

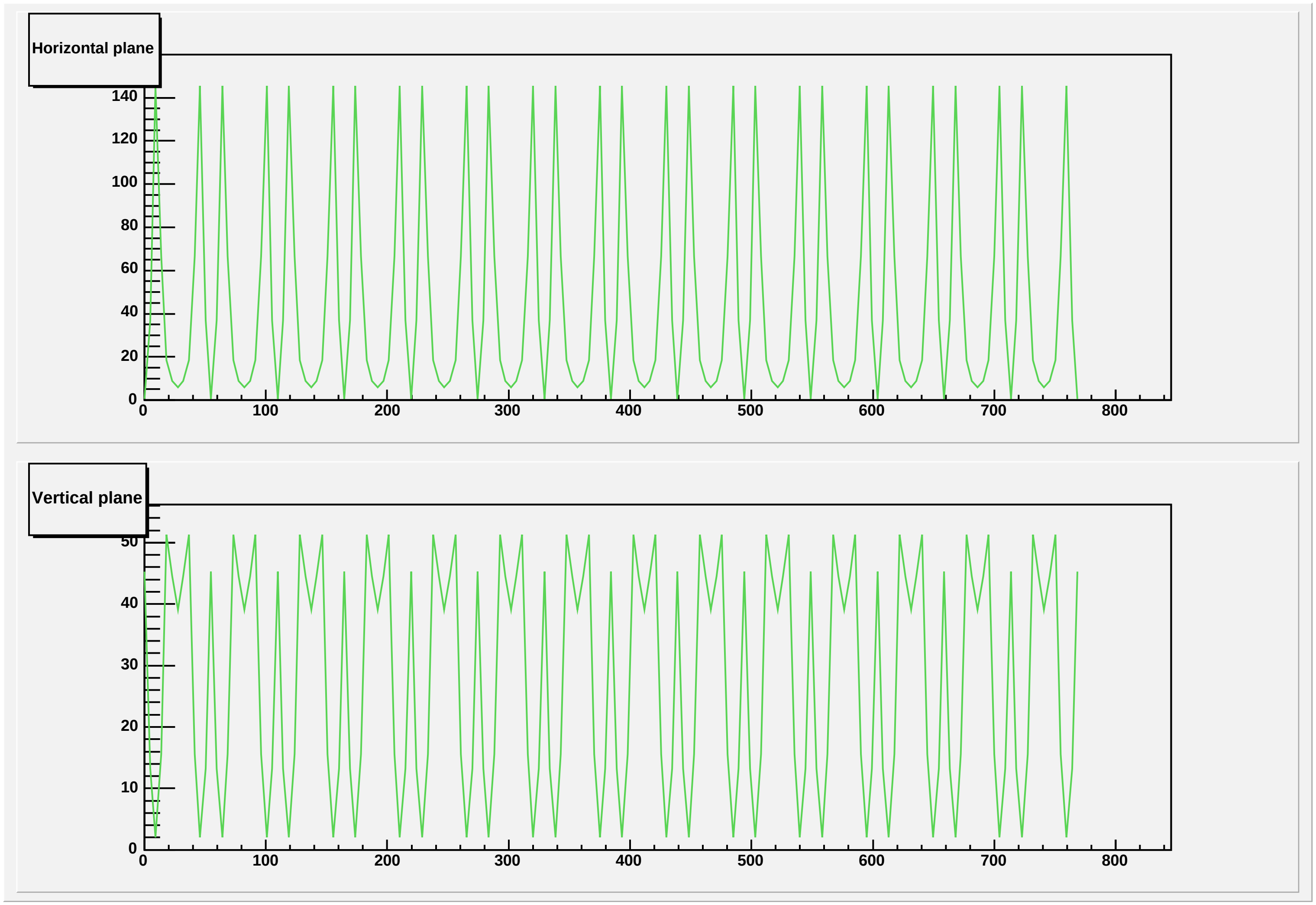


FIGURE 1.2. UAL-generated Twiss functions for the `isochronous` lattice.

The toy lattice files are useful for gaining familiarity with the environment. Introductory tasks mainly amount to checking and correlating a few results in this output by hand calculation (using any computational tool you wish, such as hand or online calculator) using formulas given in these notes.

One thing that makes accelerator physics hard is that the presence of small effects of one sort or another almost always causes minor disagreements among quantities calculated in different ways. Examples will be given shortly. Only with experience does one obtain good judgment about what to insist upon, and what to let pass. One purpose of this course is to learn how to use the UAL environment to provide some of this experience. It is to be a "laboratory course" where laboratory is being used in the sense of computational laboratory. After understanding the meanings of quantities, they will be estimated and the estimations compared to accurately computed values. Even when quantities cannot be calculated with absolute accuracy, analytic formulas can often be used to calculate the *changes* of these quantities when a lattice element strength is changed by a small amount.

Commonly in an accelerator control room the installed lattice only agrees "more or less" with the design. (As accelerators have become larger and larger this becomes more and more invevitable. Not only are there more elements, for which the parameters are only controlled approximately, but the same fractional error is more significant in a big ring than in a small ring.) After confirming that the installed lattice resembles the design, it is necessary to perform various fine tuning operations to correct for minor unknown errors. Furthermore the algorithms, based as they are on an idealized model, are typically not "orthogonal" (meaning that errors of one sort, such as coupling, degrade algorithms intended to correct other effects, such as closed-orbit deviation.) This forces methods to be iterated, either successfully, in the case of convergence, if the errors are small enough and the methods powerful enough, or unsuccessfully.

One application of the UAL code is to simulate these lattice tuning and correction procedures and then to determine accurate lattice properties. Some of these are just recalculations of quantities previously calculated. Most of the calculations are too complicated to be checked by hand. The associated assignments in the tutorials are to spot-check the results against the input file specifications, referring to the UAL manual to "get the drift" of what can be calculated, what can be modeled, what can be adjusted, and how to do it.

Output files for both of the initial practice lattices are available (along with other data) from the CDROM. Students are to work through the instructions associated with these lattices. There are various other, more realistic, more detailed, but still "toy lattices", that are available for similar study; for example a colliding beam lattice, and a proton accelerator (AGS) are available in `.xsl` form. Some longitudinal studies are based on RHIC, the Brookhaven Relativistic Ion Collider. That lattice is available in `.sxf` form.

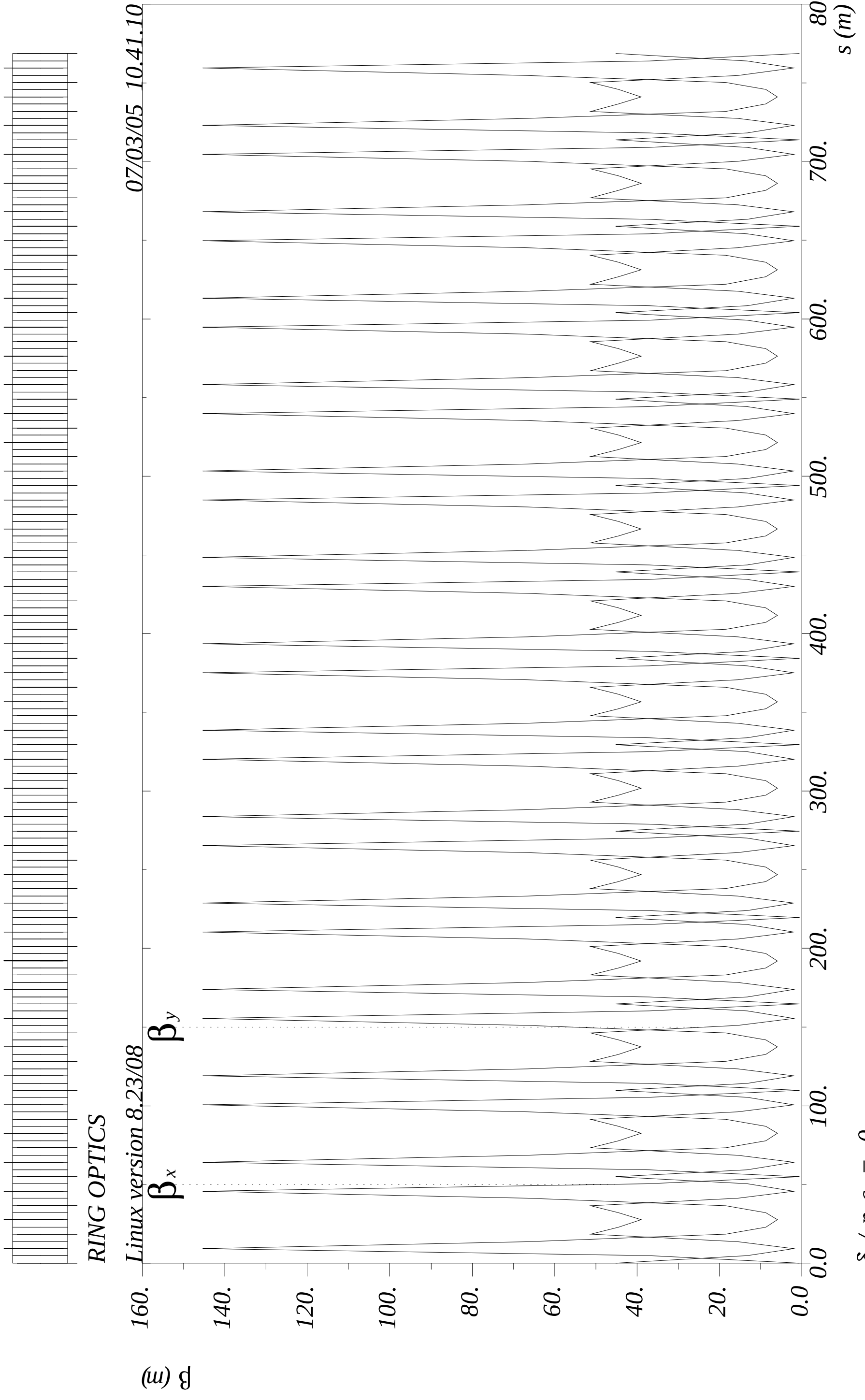


FIGURE 1.3. MAD-generated Twiss functions for the `isochronous` lattice.

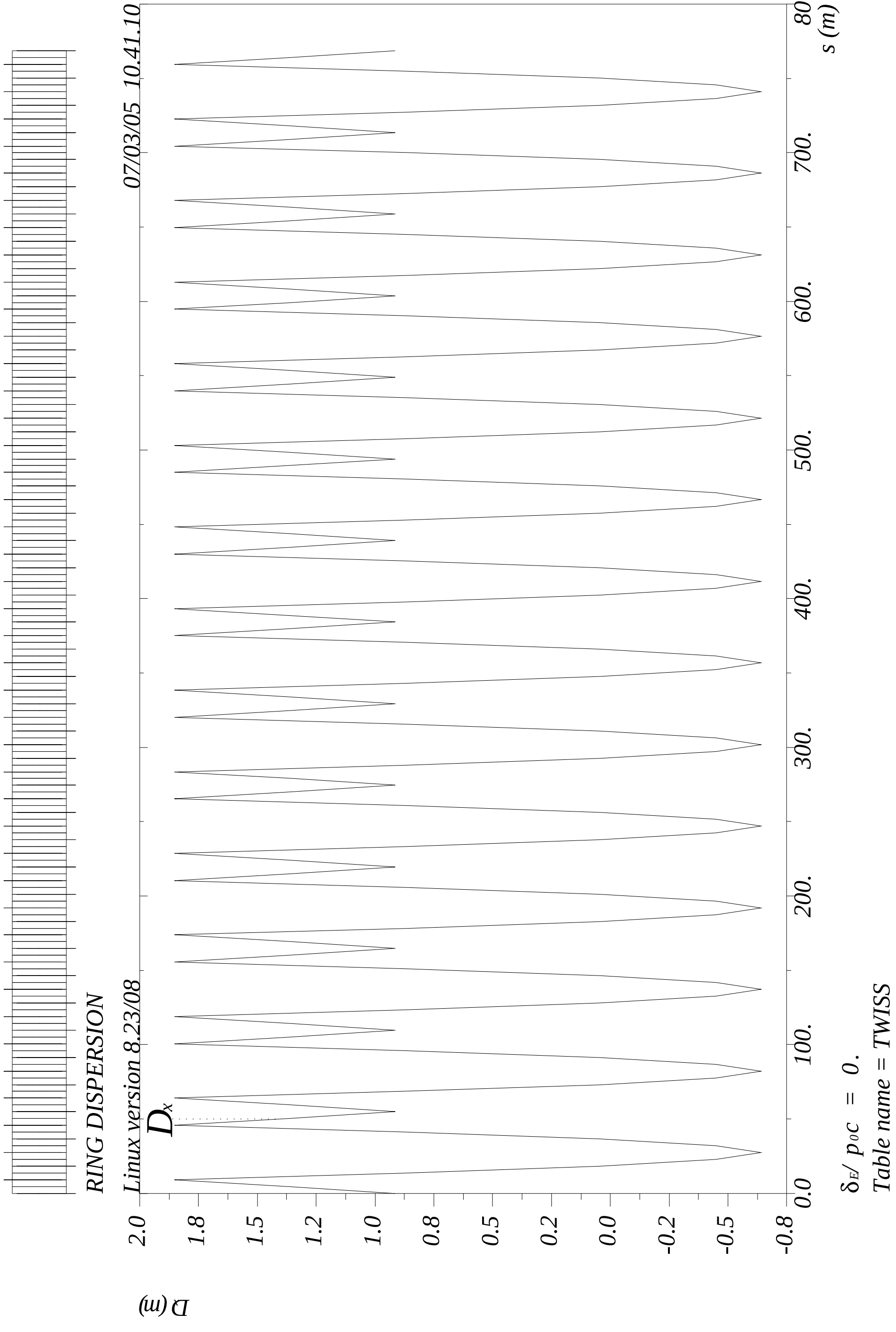


FIGURE 1.4. MAD-generated dispersion functions for the `isochronous` lattice.

TUTORIAL 1.1. *Use* UAL *to obtain the dispersion curve for the* `isochronous` *lattice, and compare the result with FIG 1.4.*

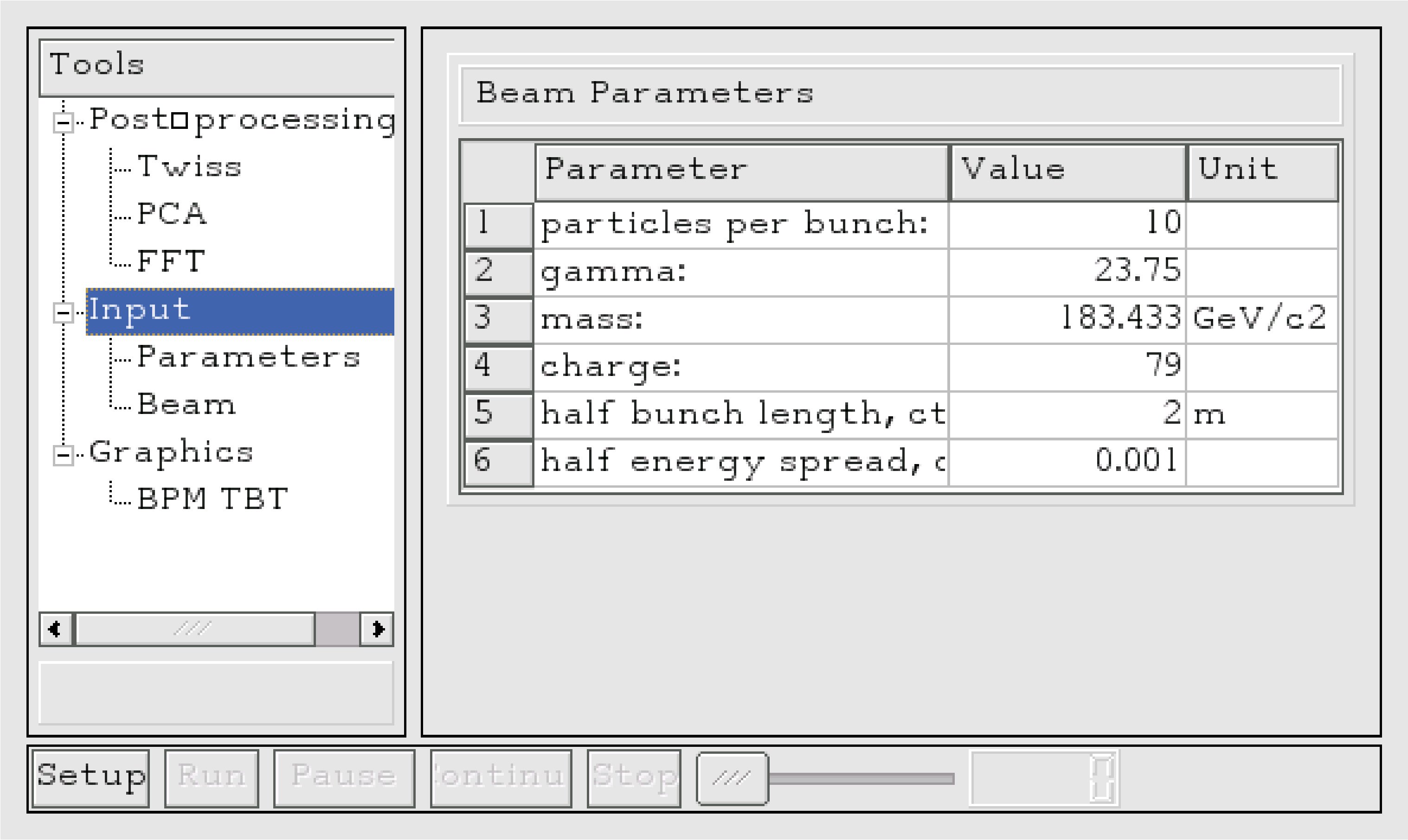
TUTORIAL 1.2. *Practice, using the mouse and caret, to zoom one of the GUI graphs. Note that there is no "box zoom". Rather, each axis is to be zoomed individually. Also learn how to read accurate numericaal values for one of the plotted lattice functions.*

After completing these initial assignments students are encouraged to prepare an input file description for a particular accelerator or type of accelerator of interest. Starting with this file it will be possible to complete the later stages of the course in which various physical effects are investigated.

1.7. Graphical User Interface

The main graphical user interface which controls the UAL simulator is shown in FIG 1.5. More detailed windows are shown in FIG 4.7, FIG 4.8, FIG 7.7 and FIG 7.8. This interface is based on QT[**1**] and ROOT[**2**][**3**].

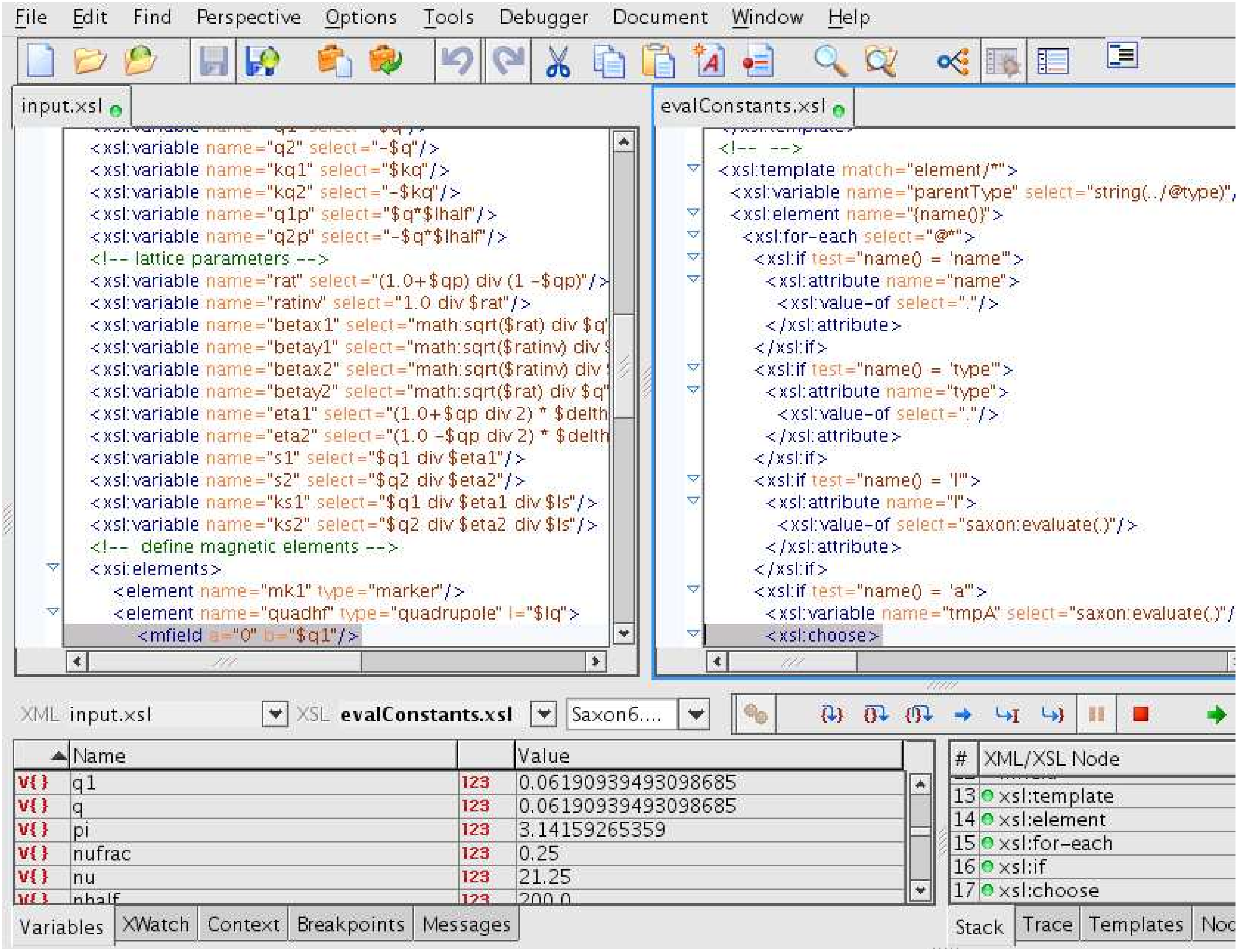
The graphical interface for the debugger for the XSLT-transformer contained in the OXYGEN, XML-aware editor is shown split between FIG 1.6 and FIG 1.7.



Beam Parameters

	Parameter	Value	Unit
1	particles per bunch:	10	
2	gamma:	23.75	
3	mass:	183.433	GeV/c2
4	charge:	79	
5	half bunch length, ct	2	m
6	half energy spread, c	0.001	

FIGURE 1.5. UAL "PlayerUI" GUI window.



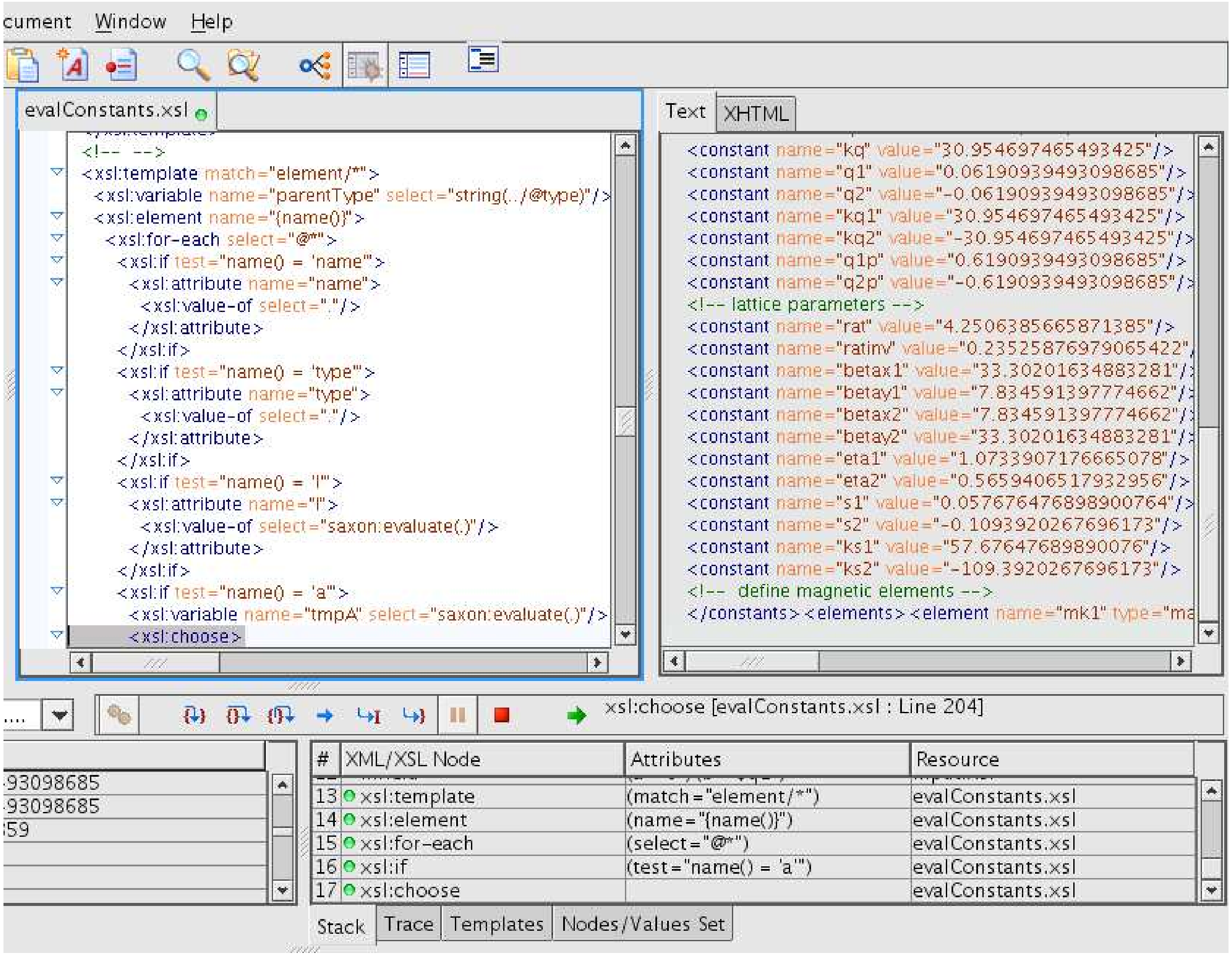
```
<xsl:variable name="q2" select="-$q"/>
<xsl:variable name="kq1" select="$kq"/>
<xsl:variable name="kq2" select="-$kq"/>
<xsl:variable name="q1p" select="$q*$lhalf"/>
<xsl:variable name="q2p" select="-$q*$lhalf"/>
<!-- lattice parameters -->
<xsl:variable name="rat" select="(1.0+$qp) div (1 -$qp)"/>
<xsl:variable name="ratinv" select="1.0 div $rat"/>
<xsl:variable name="betax1" select="math:sqrt($rat) div $q
<xsl:variable name="betay1" select="math:sqrt($ratinv) div
<xsl:variable name="betax2" select="math:sqrt($ratinv) div
<xsl:variable name="betay2" select="math:sqrt($rat) div $q"
<xsl:variable name="eta1" select="(1.0+$qp div 2) * $delth
<xsl:variable name="eta2" select="(1.0 -$qp div 2) * $delth
<xsl:variable name="s1" select="$q1 div $eta1"/>
<xsl:variable name="s2" select="$q2 div $eta2"/>
<xsl:variable name="ks1" select="$q1 div $eta1 div $ls"/>
<xsl:variable name="ks2" select="$q2 div $eta2 div $ls"/>
<!--  define magnetic elements -->
<xsl:elements>
  <element name="mk1" type="marker"/>
  <element name="quadhf" type="quadrupole" l="$lq">
    <mfield a="0" b="$q1"/>
```

```
<!--  -->
<xsl:template match="element/*">
 <xsl:variable name="parentType" select="string(../@type)",
 <xsl:element name="{name()}">
  <xsl:for-each select="@*">
   <xsl:if test="name() = 'name'">
    <xsl:attribute name="name">
     <xsl:value-of select="."/>
    </xsl:attribute>
   </xsl:if>
   <xsl:if test="name() = 'type'">
    <xsl:attribute name="type">
     <xsl:value-of select="."/>
    </xsl:attribute>
   </xsl:if>
   <xsl:if test="name() = 'l'">
    <xsl:attribute name="l">
     <xsl:value-of select="saxon:evaluate(.)"/>
    </xsl:attribute>
   </xsl:if>
   <xsl:if test="name() = 'a'">
    <xsl:variable name="tmpA" select="saxon:evaluate(.)"/
    <xsl:choose>
```

	Name		Value
V{}	q1	123	0.06190939493098685
V{}	q	123	0.06190939493098685
V{}	pi	123	3.14159265359
V{}	nufrac	123	0.25
V{}	nu	123	21.25
V{}	nhalf	123	200.0

#	XML/XSL Node
13	● xsl:template
14	● xsl:element
15	● xsl:for-each
16	● xsl:if
17	● xsl:choose

FIGURE 1.6. Left half of OXYGEN Debugger window.



```
<!-- -->
<xsl:template match="element/*">
 <xsl:variable name="parentType" select="string(../@type)"/>
 <xsl:element name="{name()}">
  <xsl:for-each select="@*">
   <xsl:if test="name() = 'name'">
    <xsl:attribute name="name">
     <xsl:value-of select="."/>
    </xsl:attribute>
   </xsl:if>
   <xsl:if test="name() = 'type'">
    <xsl:attribute name="type">
     <xsl:value-of select="."/>
    </xsl:attribute>
   </xsl:if>
   <xsl:if test="name() = 'l'">
    <xsl:attribute name="l">
     <xsl:value-of select="saxon:evaluate(.)"/>
    </xsl:attribute>
   </xsl:if>
   <xsl:if test="name() = 'a'">
    <xsl:variable name="tmpA" select="saxon:evaluate(.)"/>
    <xsl:choose>
```

```
<constant name="kq" value="30.954697465493425"/>
<constant name="q1" value="0.06190939493098685"/>
<constant name="q2" value="-0.06190939493098685"/>
<constant name="kq1" value="30.954697465493425"/>
<constant name="kq2" value="-30.954697465493425"/>
<constant name="q1p" value="0.6190939493098685"/>
<constant name="q2p" value="-0.6190939493098685"/>
<!-- lattice parameters -->
<constant name="rat" value="4.2506385665871385"/>
<constant name="ratinv" value="0.23525876979065422"
<constant name="betax1" value="33.30201634883281"/
<constant name="betay1" value="7.834591397774662"/
<constant name="betax2" value="7.834591397774662"/
<constant name="betay2" value="33.30201634883281"/
<constant name="eta1" value="1.0733907176665078"/>
<constant name="eta2" value="0.5659406517932956"/>
<constant name="s1" value="0.057676476898900764"/>
<constant name="s2" value="-0.1093920267696173"/>
<constant name="ks1" value="57.67647689890076"/>
<constant name="ks2" value="-109.3920267696173"/>
<!-- define magnetic elements -->
</constants><elements><element name="mk1" type="ma
```

xsl:choose [evalConstants.xsl : Line 204]

```
93098685
93098685
59
```

#	XML/XSL Node	Attributes	Resource
13	○ xsl:template	(match="element/*")	evalConstants.xsl
14	○ xsl:element	(name="{name()}")	evalConstants.xsl
15	○ xsl:for-each	(select="@*")	evalConstants.xsl
16	○ xsl:if	(test="name() = 'a'")	evalConstants.xsl
17	○ xsl:choose		evalConstants.xsl

FIGURE 1.7. Right half of OXYGEN Debugger window.

CHAPTER 2

Linearized Transverse Motion

2.1. Equations of Small Amplitude Transverse Motion

Consider a beam of particles being guided along a possibly-curved path, with longitudinal position specified by arc length s. To prevent the eventual loss of particles no matter how slightly divergent, it is necessary to have focusing elements. For high energy charged particles this means quadrupoles. The differential equation describing such focusing, for example in the vertical plane, is

$$\frac{d^2y}{ds^2} = K(s)y, \tag{2.1}$$

where $K(s)$ is the "vertical focusing strength". Its dependence on s permits the description of systems in which the focusing strength varies along the orbit. In particular $K(s) = 0$ describes "drift spaces" in which case Eq. (2.1) is trivially solved, and yields the obvious result that particles in free space travel in straight lines.

It is conventional to designate dy/ds by y'. There are (at least) three candidates for describing particle slopes; angle θ_y, slope y', or momentum p_y (which is scaled to the total momentum P_0). All of these are exhibited in FIG 2.1, and one sees that

$$y' \equiv \frac{dy}{ds} = \tan\theta_y = \frac{p_y}{\cos\theta_y}. \tag{2.2}$$

This multiple ambiguity in what constitutes the coordinate conjugate to y is something of a nuisance at large amplitudes but, fortunately, all three definitions approach equality in the small-angle limit that characterizes Gaussian or "paraxial" optics. One knows from Hamiltonian mechanics that p_y is the safest choice but, while limiting ourselves to Gaussian optics, we will refer loosely to y' as "vertical momentum" so that we can refer to the (y, y')-plane as "vertical phase space".

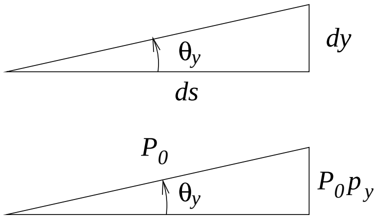


FIGURE 2.1. Spatial displacements and momentum vectors showing relations among transverse angle, momentum, and slope.

Starting from any point s_0 along the beamline, one defines two special orbits, a "cosine-like" orbit $C(s, s_0)$ with unit initial amplitude and zero slope, and a "sine-like" orbit $S(s, s_0)$ with zero initial amplitude and unit slope.

$$\begin{aligned} C(s_0, s_0) &= 1, \quad C'(s_0, s_0) = 0, \\ S(s_0, s_0) &= 0, \quad S'(s_0, s_0) = 1. \end{aligned} \tag{2.3}$$

Since unity slope is manifestly *not* a small angle, these definitions only make sense after the exact equations of motion have been linearized as in Eq. (2.1). Because Eq. (2.1) is linear and second order, *any* solution $y(s)$ and its first derivative $y'(s)$ can be expressed as that linear superposition of these two solutions that matches initial conditions $y(s_0)$ and $y'(s_0)$:

$$\begin{aligned} y(s) &= C(s, s_0) y(s_0) + S(s, s_0) y'(s_0), \\ y'(s) &= C'(s, s_0) y(s_0) + S'(s, s_0) y'(s_0). \end{aligned} \tag{2.4}$$

This can be expressed in matrix form, with $\mathbf{y} = (y, y')^{\mathrm{T}}$ being a "vector in phase space":

$$\mathbf{y}(s) \equiv \begin{pmatrix} y(s) \\ y'(s) \end{pmatrix} = \begin{pmatrix} C(s, s_0) & S(s, s_0) \\ C'(s, s_0) & S'(s, s_0) \end{pmatrix} \mathbf{y}(s_0) = \mathbf{M}(s_0, s)\mathbf{y}(s_0). \tag{2.5}$$

This serves to define $\mathbf{M}(s_0, s)$, the "vertical transfer matrix from s_0 to s". Since *any* solution of Eq. (2.1) can be expressed in this way, an entire beamline can be characterized by $\mathbf{M}(s_0, s)$. This matrix can be "composed" by multiplying (or "concatenating") the matrices for the successive beamline elements making up the line.

2.2. Pseudoharmonic Trajectory Description

An "ansatz" for solving Eq. (2.1), based on the known, "harmonic", dependence proportional to $\cos(\psi - \psi_0)$ when $K(s)$ is constant, is

$$y(s) = a\sqrt{\beta(s)}\, \cos(\psi - \psi_0). \tag{2.6}$$

Here ψ (an intermediate "independent" variable) and $\beta(s)$ depend on s but a is a constant amplitude. Differentiation of Eq. (2.6) yields

$$y'(s) = -a\sqrt{\beta(s)}\, \psi' \, \sin(\psi - \psi_0) + \frac{a\beta'}{2\sqrt{\beta}} \cos(\psi - \psi_0). \tag{2.7}$$

Substituting into Eq. (2.1), we can demand that the coefficients of sin and cos terms vanish independently, since that is the only way of maintaining equality for all values of ψ_0. This leads to the equations

$$\beta\, \psi'' + \beta' \psi' = 0, \quad 2\beta\, \beta'' - \beta'^2 - 4\beta^2 \psi'^2 + 4\beta^2 K(s) = 0. \tag{2.8}$$

From the first equation it follows that $\beta\, \psi'$ is constant. To obtain the conventional description we pick this constant to be 1 and obtain

$$\psi' = \frac{1}{\beta}, \quad \text{or} \quad \psi(s) = \psi(s_0) + \int_{s_0}^{s} \frac{ds'}{\beta(s')}. \tag{2.9}$$

Since ψ is the argument of a sinusoidal function, and the argument of a harmonic wave is $2\pi s$/wavelength, this permits us to interpret $2\pi\beta(s)$ as a "local wavelength"

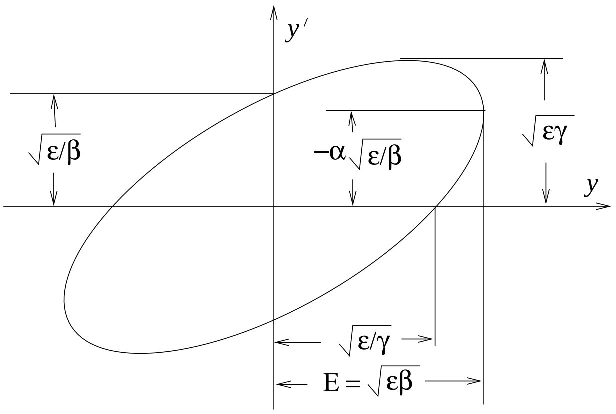


FIGURE 2.2. An elliptical beam in vertical phase space, showing the geometric configuration of a beam ellipse matched to the local Twiss parameters α, β and γ. Except for a factor of order 1, depending on the detailed beam distribution, the area of the ellipse is the emittance ε. The skew orientation depends primarily on Twiss parameter α. E is the "beam envelope".

or, equivalently, $1/\beta(s)$ is the "local wave number". Substituting into the second of Eqs. (2.8), we obtain

$$\beta'' = 2\beta\, K(s) + 2\frac{1+{\beta'}^2/4}{\beta}. \tag{2.10}$$

This second order, nonlinear differential equation is usually considered to be the fundamental defining relationship for the evolution of the lattice β-function. Because $K(s)$ depends on s, solving the equation may be quite difficult in general. In a problem below a "first integral" of Eq. (2.10) is obtained;

$$\frac{y^2}{\beta} + \beta(y' - \frac{\beta'}{2\beta}y)^2 = a^2. \tag{2.11}$$

Even without $\beta(s_0)$ yet having been determined, this equation makes it natural, at fixed position s_0, where $(\alpha(s_0), \beta(s_0)) = (\alpha_0, \beta_0)$, to plot the ellipse shown in FIG 2.2, for $(\alpha, \beta) = (\alpha_0, \beta_0)$. As s increases away from s_0, individual points evolve independently, but points sharing this ellipse at s_0 will share the same $(\alpha, \beta) = (\alpha_1, \beta_1)$ ellipse at $s = s_1$.

2.3. Relations Among the "Twiss" Lattice Functions

Since properties of the so-called "Twiss lattice functions", $\alpha(s)$, $\beta(s)$ and $\gamma(s)$, are spelled out in all accelerator books, only the briefest of outlines will be given here. With $\beta(s)$ already defined, the defining relation for $\alpha(s)$ and $\gamma(s)$ are

$$\alpha = -\frac{1}{2}\frac{d\beta}{ds} \equiv -\frac{\beta'}{2}, \quad \gamma = \frac{1+\alpha^2}{\beta}. \tag{2.12}$$

In this text, while analysing toy lattices, since nothing but thin elements and drifts are used, it is sufficient to work out the s dependence of the Twiss functions for just drifts and thin lenses, which is to say, quadrupoles. Other than requiring input

TABLE 2.1. Lattice function evolution formulas. Sign of q is governed by horizontal ($q > 0 \rightarrow$ horizontal defocus) and is the same for all entries. Both $\Delta\theta$ and ρ are assumed to be positive.

	Vertical β_y	Horizontal β_x	Dispersion $\tilde{D} \equiv \tilde{D}_x$
DRIFT length s	$\beta = \beta_0 - 2\alpha_{0+}s + \gamma_{0+}s^2$ $\alpha(s) = \alpha_0 - \gamma_0 s$ $\gamma(s) = \gamma_0$	$\beta = \beta_0 - 2\alpha_{0+}s + \gamma_{0+}s^2$ $\alpha(s) = \alpha_0 - \gamma_0 s$ $\gamma(s) = \gamma_0$	$\tilde{D} = \tilde{D}_0 + \tilde{D}'_{0+}s$ $\tilde{D}'(s) = \tilde{D}'_0$
thin QUAD q	$\beta_+ = \beta_-$ $\alpha_+ = \alpha_- + \beta_0 q$ $\gamma_+ = \gamma_- + 2\alpha_- q + \beta_0 q^2$	$\beta_+ = \beta_-$ $\alpha_+ = \alpha_- - \beta_0 q$ $\gamma_+ = \gamma_- - 2\alpha_- q + \beta_0 q^2$	$\tilde{D}_+ = \tilde{D}_-$ $\tilde{D}'_+ = \tilde{D}'_- + \tilde{D}_0 q$
thin BEND $\Delta\theta$	no change	$\beta_+ = \beta_-$ $\alpha_+ = \alpha_- + \beta_0\,\Delta\theta/\rho$ $\gamma_+ = \gamma_- + 2\alpha_-\Delta\theta/\rho + \beta_0\,\Delta\theta^2/\rho^2$	$\tilde{D}_+ = \tilde{D}_-$ $\tilde{D}'_+ = \tilde{D}'_- + \Delta\theta$

and output positions to be the same, the defining equation of a thin lens of focal length $f \equiv -1/q$ is

$$\Delta y' = qy. \tag{2.13}$$

The lens strength q and focusing function $K(s)$ are related by

$$q = \int K(s)ds, \tag{2.14}$$

where the range of integration spans the lens location.

The Twiss function dependencies for drifts and thin lenses are given by formulas in the second and third columns of Table 2.1. All entries are to be worked out in problems below. For drifts, propagation is from 0 to s. When applied to a drift, for a potentially discontinuous quantity such as α, the value α_{0+} stands for $\alpha(0+)$, the value just after the thin element at $s = 0$. This defines the start of the drift. In general α_- and α_+ are the values just before and just after a thin element. For quadrupoles, the vertical and horizontal columns are related by the the well-known result (also proved in Eq (8.6)) that the focal lengths of quadrupoles are equal in magnitude but opposite in sign for horizontal and vertical planes.

The convention used in the table is that positive q corresponds to the quad being focusing in the horizontal plane. Since there is no universally accepted convention, it is necessary to be checking the signs of quadrupole strengths carefully when different formalisms are compared.

For convenience the fourth column of the table also gives the variation of dispersion[1] $\tilde{D}(s)$. Those entries, which depend of bending magnets, will be discussed later.

What makes drifts simple is that, since $K = 0$, the first term on the right hand side of Eq. (2.10) vanishes. What makes thin lenses simple is that, since $K = \infty$ (at the lens location) the second term on the right hand side of Eq. (2.10) can be

[1] Tildes on $\tilde{D}(s)$ and $\tilde{\delta}$, here and later, will be explained below. They are introduced for notational consistency with the treatment of longitudinal dynamics in later chapters. Except for a factor β_0, δ and $\tilde{\delta}$ are identical. At the level of faithfulness justified for the toy lattices under study one should assume fully relativistic motion for which $\beta_0 = 1$. This justifies simply ignoring the tildes.

neglected there. (Individual particle trajectories have to be continuous, even in passing through thin elements. As a result $\beta(s)$ has to be a continuous function of s, even at a lens location. This means that β' has to be finite there.) At a lens location, because $\beta''=\infty$, it follows from Eq. (2.12) that $\alpha(s)$ is discontinuous at the lens position. In other words, the function $\beta(s)$ has a kink there.

This section has now included all the hints necessary to derive all entries in columns two and three of Table 2.1. Problems to this effect are given next.

PROBLEM 2.1. *With a view toward eliminating the argument $\psi-\psi_o$ from Eqs. (2.6) and (2.7), solve the second of these equations for $a\sqrt{\beta(s)}\,\sin(\psi-\psi_0)$, expressed in terms of y and y'. Then square and sum the two equations. In this way prove the constancy of the "first integral" introduced in Eq. (2.11).*

PROBLEM 2.2. *In a drift region $K=0$, which simplifies Eq. (2.10) markedly. Solve this differential equation to show that the variation of $\beta(s)$ has to be quadratic in s. This has derived the top row entry in each of the first two columns of Table 2.1.*

PROBLEM 2.3. *Continuing from the previous problem, use the relations contained in Eqs. (2.12) to derive the dependencies of $\alpha(s)$ and $\gamma(s)$ in drift regions. In other words derive the second and third rows of Table 2.1 for the variation of Twiss functions in drift sections.*

PROBLEM 2.4. *For thin elements it was argued above that the last term of Eq. (2.10) can be dropped. Use the resulting equation, along with Eqs. (2.12) and (2.14), to derive the $\alpha_+-\alpha_-$ discontinuity relations given in Table 2.1.*

PROBLEM 2.5. *Continuing from the previous problem, use the relations contained in Eqs. (2.12) to derive the discontinuity equations for $\beta(s)$ and $\gamma(s)$ at thin lens locations. In other words derive the first and third rows of Table 2.1 for the variation of Twiss functions at thin lens locations..*

In drifts and quads the graph of $\tilde{D}(s)$ is the same as that of a horizontal particle trajectory. When passing through a thin dipole that causes inward deflection through angle $\Delta\theta$, $\tilde{D}$ suffers an outward kink $\Delta\theta$. This means that (except in reverse bends, which are rarely used) a dipole causes $\tilde{D}(s)$ to be "repelled" from the horizontal axis. For this reason $\tilde{D}(s)$ is positive everywhere in ordinary lattices. Both β_x and β_y are necessarily everywhere positive because they are "repelled" from the axis in drift spaces (with strength inversely dependent on β). (A counter example, in which negative dispersion is intentionally present is the `isochronous.adxf` lattice.)

2.4. Establishing Absolute Values of the Twiss Functions

Equation 2.10 fixes only *variation* of $\beta(s)$. As with any second order, ordinary differential equation, it is necessary to use initial conditions or boundary conditions to fix the two undetermined parameters. Which of these conditions is to be used depends on the way the Twiss functions are to be interpreted. There are two main lines of development, depending on whether an open "transfer line" or a closed "circular ring" is under discussion.

- The ellipse shown in FIG 2.2 can be regarded as the aspect ratio of a beam of particles in one dimensional phase space. (There would be a similar plot for the other transverse plane.) In this case the parameters

$(\alpha_0, \beta_0, \gamma_0)$ are properties of the beam. They can be established or varied with no reference to any lattice; (for example, by changing voltages on the electrodes of the source or "gun" from which the particles are generated.) This triplet of values serves as initial conditions establishing absolute values of the Twiss functions for the transfer line into which this beam is injected. In this case the Twiss functions can be regarded as properties of the beam.

- For a (more or less) circular ring, it is natural to establish absolute values for the Twiss functions by using boundary conditions. Assuming a closed orbit is known, and that the coordinates being used are measured relative to that orbit, the focusing function $K(s)$ is necessarily periodic, with period $\mathcal{C}_0$, which is the orbit circumference. Requiring the boundary conditions $\alpha(\mathcal{C}_0) = \alpha_0$ and $\beta(\mathcal{C}_0) = \beta_0$ fixes the absolute values of the Twiss functions. In this case the Twiss functions can be regarded as properties of the lattice.

When a beam is injected into a circular ring there is a clash between these two sets of Twiss functions. Ideally the two sets would be identical, in which case the beam is said to be "matched". In this case it is unnecessary to distinguish between the two definitions, and the Twiss functions are defined unambiguously for one full turn around the machine and, for that matter, for all subsequent turns.

Naturally the beam are never be *exactly* matched to the lattice. If the lattice were truly linear then the bunch characteristics would survive indefinitely. But the lattice is never truly linear and, after a sufficiently long time, a process called filamentation, will cause the beam to adapt itself to the lattice. This process, which also goes by various names such as "emittance dilution" and "decoherence" is the subject of Chapter 7.

2.5. Transfer Matrices for Simple Elements

2.5.1. Drift space. The most important transfer matrix is $\mathbf{M}_l$, which describes propagation through a drift space of length ℓ. Since the orbits are given by $y(s) = y_0 + y_0's, y'(s) = y_0'$, we have

$$\mathbf{M}_l = \begin{pmatrix} 1 & \ell \\ 0 & 1 \end{pmatrix}. \tag{2.15}$$

2.5.2. Thin lens. The next most important transfer matrix describes a "thin lens" where the definition of "thin" is that the thickness Δs is sufficiently small that coordinate $y(s - \Delta s/2)$, just before the lens, and $y(s + \Delta s/2)$, just after, can be taken to be equal. A typical focusing profile is shown in FIG 2.3. The lens causes a "kink" $\Delta y' = y'(s + \Delta s/2) - y'(s - \Delta s/2)$ in the orbit which, as shown in the figure, is taken as occurring at the center of the lens. The kink can be obtained by integrating Eq. (2.1) from O_-, just before the lens to O_+, just after it:

$$\Delta y' = \int_{O_-}^{O_+} \frac{d}{ds}\Big(\frac{dy}{ds}\Big) ds = y \int_{O_-}^{O_+} K(s) ds \equiv y\, K \Delta s. \tag{2.16}$$

This relation defines the product $K\Delta s$. Now, a focusing strength that changes discontinuously from 0 to K is not actually realistic. But the product $K\Delta s$, known as a "field integral", can be regarded as an abbreviation for $\int_{O_-}^{O_+} K(s)ds$ where O_- and O_+ are well outside the field region. If K is taken to be equal to K_O (the value

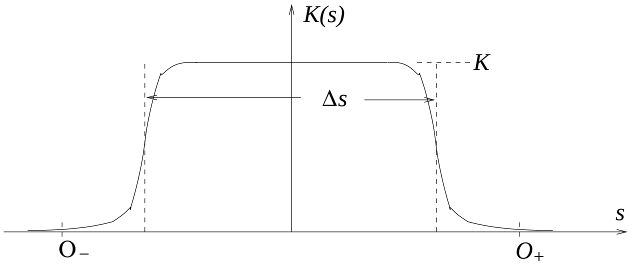


FIGURE 2.3. Realistic thin lens focusing profiles are more or less constant with value K over a central region and become negligible outside points O_- and O_+. Effective length Δs is determined by matching $K\Delta s$ to the "field integral".

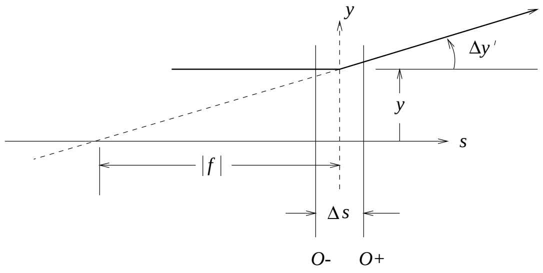


FIGURE 2.4. Focusing action of a thin lens for which the focusing strength-length product is $K\Delta s$ and fields outside the range $O_- < s < O_+$ can be neglected. The "focal length" is $f = -(K\Delta s)^{-1} \equiv -q^{-1}$.

at the center of the element) then Δs is typically equal to (or, because of fringe fields, slightly greater than) the physical length of the element. The "focal length" f of the lens, defined in FIG 2.4, and the "lens strength" $q = -1/f$, are then given by

$$q = -\frac{1}{f} = \frac{\Delta y'}{y} = K\Delta s. \tag{2.17}$$

Building in the approximation that y is contant through the lens, the transfer matrix is then given by

$$\mathbf{M}_q = \begin{pmatrix} 1 & 0 \\ q & 1 \end{pmatrix}. \tag{2.18}$$

As drawn, K and q are positive, f is negative, and the lens is "defocusing".

2.5.3. Thick lens. The condition for the thin lens formula just given to be valid ($\Delta s << |f|$) is usually well satisfied for accelerator beamlines. Even if non-vanishing, if $K(s)$ is constant (as it usually is, by design anyway) it is easy to integrate Eq. (2.1). This yields matrix elements of $\mathbf{M}$ that are no worse than sines and cosines (or hyperbolic sines and cosines, depending on the sign of K). Formulas for ideal thick quadrupoles are given in most accelerator books and, other than in a problem below, won't be discussed here. When the linear lattice assumption is regarded as acceptably accurate, these matrices can be used in UAL, for example in the `FastTeapot`, to minimize computation time through linear sections.

For low energy, few element, accelerators, explicit thick element matrices used to be considered "canonical". But, for high energy accelerators, the thin lens approximation is usually adequate. In any case, making use of now readily available computer power, one can always split elements longitudinally to better validate the assumption that the elements are "thin". Quite apart from improving accuracy, it is handy to split elements in two in order to enable lattice function evaluations at lens centers. (The lattice functions typically go through maxima or minima near these points and aperture limits are usually established at those points.) Even in the most extreme cases of intersection region quads, splitting by another factor of, say, four, is more than adequate, especially since the residual inaccuracy is typically less than the errors due to the neglect of other factors like fringe fields. For these reasons thick element formulas are de-emphasized here.

The emphasis on thin elements in UAL resembles the restriction to finite elements in the *finite element method* of mechanical engineering. In both cases the explicit need for numerical treatment reduces the usefulness of idealized, thick element, analytical approximations. In the case of particle dynamics the requirement of symplecticity makes this discretization obligatory, since there is no known symplectic treatment of distributed nonlinearity. Historically this led to the use of so-called "kick codes" for incorporating nonlinearity. The first general-purpose code taking this route was TEAPOT[**6**]. Over time this has evolved into the vastly more general and more inclusive UAL code. As well as incorporating TEAPOT, the UAL environment has come to incorporate the homogeneous inclusion of heterogeneous codes while preserving the symplectic capability. By using truncated Taylor series (TPS) maps, even thick elements can be represented analytically, though symplecticity can only be assured up to a given polynomial order.

2.5.4. Bending Magnet. Bending magnets (also known as uniform field magnets or, based on a "multipole expansion" to be described in Chapter 8, as "dipoles") are obviously needed to bend the particles into closed curves. The main parameters defining a bending magnet are the bend angle $\Delta\theta$ and the arc length l_0 of the reference orbit as it passes through the magnet. Since there are significant end effects it is necessary to also specify entrance and exit angles. The simplest special case has both entrance and exit normal to the pole face, in which case the magnet is referred to as an `sbend`, which stands for "sector bend". In this text bending magnets without further detailing will be assumed to be `sbend`s. Another special bending magnet, haing parallel exit and entrance pole faces, is referred to as an `rbend`, which stands for "rectangular bend". (In the case of an `rbend`, instead of design orbit arc length, the "magnet length" is usually taken to be the design orbit chord length, which is the same as distance from pole to pole.) For a general bending magnet the entrance and exit pole face angles `E1` and `E2` also need to be given. A rather careful treatment of orbits in an `sbend` is given in section 8.6.4; the present section contains only a simplified discussion.

Particles travelled in perfect circles in the original particle accelerator, the cyclotron. Starting from one arbitrary point P, the central, reference particle travels in one such circle, of radius ρ. A particle starting from point P with zero momentum offset, but with small angular deviation x'_0 from the reference momentum, travels in a circle of the same radius and, as a result, returns to the same point P after one revolution. (This is the basis for the 360° spectrometer.) With x being the radial coordinate, one sees that there is a "geometric focusing effect" tending to "restore" x toward its equilibrium value $x = 0$. The radial motion is described approximately by the equations

$$x = x_0'\rho \sin\frac{s}{\rho}, \quad \frac{dx}{ds} = x_0' \cos\frac{s}{\rho}, \quad \frac{d^2x}{ds^2} = -\frac{x_0'}{\rho}\sin\frac{s}{\rho} = -\frac{1}{\rho^2}\,x. \tag{2.19}$$

This means that, for radial motion, Eq. (2.1) needs to be replaced by

$$\frac{d^2x}{ds^2} = \big(K_q(s) + K_b(s)\big)\,x, \quad \text{where} \quad K_b = -\frac{1}{\rho^2}. \tag{2.20}$$

Here K_q replaces K and represents the focusing of a so-called combined function magnet, and K_b represents the geometric focusing. In a uniform field magnet the $K_q(s)$ term vanishes. Note that the $K_b\,x$ term appears only in the x equation; unlike quad focusing, there is geometric focusing only in one of the two transverse planes. With the newly-included K_b term, the calculation of transfer matrices and Twiss evolution in a bending magnet is just like the corresponding calculation in a quadrupole. For inclusion in Table 2.1, the relation $l_0/\rho = \Delta\theta$ has been used. As with thick quadrupoles, for greater accuracy, the sector bend can be sliced into slimmer sectors.

PROBLEM 2.6. *For an ideal, horizontally-focusing, thick quadupole, the focusing strength is $K_x(s) = -K$, with K positive, and trajectory equations (2.1) become*

$$\frac{d^2x}{ds^2} = -Kx, \quad \textit{and} \quad \frac{d^2y}{ds^2} = Ky. \tag{2.21}$$

Show that the transfer matrices through such a quadrupole of length L are given by

$$\mathbf{M}_x = \begin{pmatrix} \cos\sqrt{K}L & \frac{1}{\sqrt{K}}\sin\sqrt{K}L \\ -\sqrt{K}\sin\sqrt{K}L & \cos\sqrt{K}L \end{pmatrix}, \tag{2.22}$$

$$\mathbf{M}_y = \begin{pmatrix} \cosh\sqrt{K}L & \frac{1}{\sqrt{K}}\sinh\sqrt{K}L \\ \sqrt{K}\sinh\sqrt{K}L & \cosh\sqrt{K}L \end{pmatrix}. \tag{2.23}$$

PROBLEM 2.7. *In Eq. (2.5), the transfer matrix $\mathbf{M}(s_0, s)$ was defined, with matrix elements C, S, C', and S'. Find these elements for a drift section and show that the Twiss function evolution through the drift can be expressed as*

$$\begin{pmatrix} \beta \\ \alpha \\ \gamma \end{pmatrix} = \begin{pmatrix} C^2 & -2CS & S^2 \\ -CC' & CS' + SC' & -SS' \\ C'^2 & -2C'S' & S'^2 \end{pmatrix} \begin{pmatrix} \beta_0 \\ \alpha_0 \\ \gamma_0 \end{pmatrix}. \tag{2.24}$$

PROBLEM 2.8. *Show that Eq. (2.24) is also valid for propagation through a thin lens.*

PROBLEM 2.9. *Show that Eq. (2.24) remains valid when applied to a drift followed by a thin lens or, for that matter, to any sequence of thin elements and drifts. Even a thick lens transfer matrix can be composed by concatenating drifts and thin lenses. It follows that Eq. (2.24) can be applied to an arbitrary linear transfer line. [This problem is noticeably more difficult than the others.]*

2.6. Off-Momentum Behavior

2.6.1. Fractional Momentum and "Fractional" Energy. There is a mild inconsistency in these notes, and in the accelerator field at large, concerning the definition of the fractional longitudinal momentum/energy variable δ. As with many other quantities, there are both electron/hadron conventions and American/European conventions. Until quite recently the conventional meaning for δ, at least in the electron world, was "fractional momentum deviation". This definition is especially appropriate for transverse dynamics because magnetic deflections are inversely proportional to p. To "geometricize" lattice theory it is conventional to "factor out" dependence on the total momentum p_0 from the transverse transverse momentum components. We therefore define

$$\tilde{\delta} = \frac{\Delta p}{p_0}, \tag{2.25}$$

where $\Delta p = p - p_0$. The purpose of the overhead tilde is to distinguish this definition from the following alternate definition. For longitudinal dynamics, the fundamental effect of the cavity is an energy change $\Delta E = E - E_0$ (rather than a momentum change). The fractional effect can be specified by

$$\delta = \frac{\Delta E}{p_0 c}. \tag{2.26}$$

For reasons of symplecticity the normalizing factor here has to be the same as the normalizing factor for transverse momenta. This, and the factor c inserted for units convenience, account for the choice of denominator p_0c in Eq. (2.26). So it is almost, but *not quite*, valid to describe δ as "fractional change of energy".

These two definitions are related by

$$\delta = \frac{\Delta E}{p_0 c} = \frac{\Delta p}{p_0}\,\frac{dE}{d(pc)} = \beta_0 \tilde{\delta}. \tag{2.27}$$

Since these definitions differ only by the factor β_0, which approaches 1 in the relativistic limit, the distinction is unimportant for fully relativistic accelerators. For electrons this includes essentially all accelerators, but for practical hadron accelerators δ and $\tilde{\delta}$ may differ appreciably.

For an introductory discussion of transverse lattice optics (such as the analysis of toy lattices in the early chapters of these notes) use of δ defined by Eq. (2.26) introduces seemingly *ad hoc* factors of β_0 into all magnetic deflection formulas. There are two ways to overcome this inconvenience. One way is to declare that all formulas apply only to fully relativistic motion, where $\beta_0 = 1$. Another way is to use the variable $\tilde{\delta}$ defined by Eq. (2.25). In the early chapters of these notes *both* of these approaches will be taken. Not only will fractional momentum be expressed as $\tilde{\delta}$ (to make the formulas technically correct) but also the formulas will be assumed to apply to fully-relativistic motion for which the relation $\delta = \tilde{\delta}$ is valid in any case.

When advancing to accurate description of longitudinal motion in hadron accelerators it will be important and necessary to consider these issues more carefully.

2.6.2. Dispersion. During acceleration the radial coordinate x and the off-momentum coordinate $\tilde{\delta}$ are "coupled" by the dynamics. But at fixed energy the momentum $p = p_0(1 + \tilde{\delta})$, for any particular particle, and therefore also $\tilde{\delta}$, can be regarded as a constant parameter of that particle. It is traditional, therefore, for given $\tilde{\delta}$, to find the the closed orbit $x_{\tilde{\delta}}(\tilde{\delta})$, and, from it, to define "dispersion function"[2] $\tilde{D}(s)$ by

$$x_{\tilde{\delta}}(\tilde{\delta}) = \tilde{D}(s)\,\tilde{\delta}, \tag{2.28}$$

This equation is exact and does not assume that $\tilde{\delta}$ is small, even though the right hand side looks like the first term in a Taylor expansion.

The dispersion function is used to separate a general horizontal displacement x into two parts:

$$x = x_{\tilde{\delta}} + x_{\beta} \equiv \tilde{D}(s)\,\tilde{\delta} + x_{\beta}. \tag{2.29}$$

Since $\tilde{\delta}$ is a constant parameter, the entire Courant-Snyder formalism applies to the x_β evolution, provided the Twiss functions are worked out for the off-momentum orbit. They should therefore be symbolized as $\tilde{\alpha}, \tilde{\beta}, \tilde{\gamma}$. Nothing in this formalism has required $\tilde{\delta}$ to be small. But most lattices, in fact, have limited momentum aperture which restricts $\tilde{\delta}$ to quite small values.

As it has been defined, $\tilde{D}(s)\tilde{\delta}$ is simply a particle trajectory and so also is $\tilde{D}(s)$. In the linearized formalism, a constant factor, such as the factor $\tilde{\Delta}(s)$, affects only the amplitude, not the shape of the trajectory. *All* orbits in field free regions are simply straight lines. This accounts for the dispersion entries for `drift` elements in Table 2.1;

All that remains is to evaluate the kinks occuring in the dispersion function at the locations of thin elements. A particle of momentum $p = p_0(1+\tilde{\delta})$, when in a uniform magnetic field, travels in a circle of radius $\rho_0(1+\tilde{\delta})$. In traveling a distance Δs such a particle suffers an outward angular deflection (relative to reference)

$$\Delta x' = \Delta s\Big(\frac{1}{\rho_0} - \frac{1}{\rho}\Big) \approx \Delta s\,\frac{1}{\rho_0}\,\tilde{\delta}. \tag{2.30}$$

The equation of the off-momentum orbit is obtained by adding this contribution to Eq. (2.20);

$$\frac{d^2\tilde{D}(s)}{ds^2} + \Big(-K_q(s) + \frac{1}{\rho_0^2}\Big)\,\tilde{D}(s) = \frac{1}{\rho_0}, \tag{2.31}$$

where a common factor $\tilde{\delta}$ has been divided out.

This equation can be used to derive all entries in the "Dispersion" column of Table 2.1. The interpretation of the $\tilde{D}$ kink occurring at a bend magnet is not quite as clean as the kink at the center of a thin lens. There is no really consistent way to let the bending magnet length go to zero while preserving its bend angle.[3] As in FIG 2.4 one can approximate $\tilde{D}$ by straight lines, with a kink at the center of the bend. But letting the magnet length approach zero while holding the bend

[2] The off-momentum closed orbit deviation is traditionally expressed as $D(s)\delta$, where $D(s)$ is known as the "dispersion function". To be consistent, when using $\tilde{\delta}$, we have introduced a modified dispersion function $\tilde{D}$ such that the off-momentum orbit deviation is $\tilde{D}(s)\tilde{\delta} \equiv D(s)\delta$. This is the source of the tildes appearing on $\tilde{D}$ in Table 2.1. As mentioned already, at the level of accuracy justified for the toy lattices under study, the tildes on $\tilde{\delta}$ and $\tilde{D}$ should probably simply be ignored. The mnemonic variable names in the toy lattice descriptions suppress the tildes. (Also the electron-world notation `eta`, rather than `D`, is used for dispersion in the toy lattice descriptions.)

[3] The so-called *drift/kick split* symplectic integration algorithm (discussed in Chapter 8) involves the longitudinal splitting of bends into arbitrarily short intervals. But in that case the bend per interval also approaches zero. Nevertheless, the *bend/kick split*, in which the orbit is formed from circular arcs, avoids this sensitivity and is therefore a safer approach.

angle fixed would entail also $\rho_0 \to 0$. Clearly the result in Table 2.1 for the $\tilde{D}$ kink at a bend assumes $\tilde{D} << \rho_0$. This is amply valid in high energy accelerators. But small accelerators require greater care. That is, $\tilde{D}$ has to be obtained by actually solving Eq. (2.31).

PROBLEM 2.10. *Eq. (2.31) is sufficiently general to describe both quadrupoles and bending magnets and, for that matter, also combined function magnets. Assuming $\tilde{D} << \rho_0$ in the case of bends, use Eq. (2.31) to derive all entries in the "Dispersion" column of Table 2.1.*

For off-momentum, horizontal particle propagation from an arbitrary starting point s_0 to another point s it is useful to define a two-argument dispersion function $\tilde{D}(s, s_0)$ With this function Eq. (2.5) generalizes to

$$\begin{pmatrix} x(s) \\ x'(s) \\ \tilde{\delta} \end{pmatrix} = \begin{pmatrix} C(s,s_0) & S(s,s_0) & \tilde{D}(s,s_0) \\ C'(s,s_0) & S'(s,s_0) & \tilde{D}'(s,s_0) \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x(s_0) \\ x'(s_0) \\ \tilde{\delta} \end{pmatrix} . \tag{2.32}$$

CHAPTER 3

Thin Element “Toy” Lattices

3.1. Introduction

In preparation for investigating toy lattices with UAL, this chapter begins by deriving analytic formulas for the simplest possible accelerator lattice, one made up entirely of equal tune FODO sections. The analytic formulas are used to give starting values for a first-cut design that will later be made more realistic and tuned up by UAL. As already mentioned, a characteristic feature of accelerator lattices is that it is fairly difficult to obtain an absolute design but that it is fairly easy to make small changes around an existing design. Also, once a coarse but stable design has been achieved, it is straightforward to adjust many of the parameters to good accuracy. In the control room this is only practical if instrumentation is available for measuring the quantity in question to good accuracy. In a computer simulation the quantity can be calculated to arbitrary accuracy.

Once the parameters of a lattice to be studied have been established, more detailed, more visual studies are to be performed using a graphical user interface (GUI). This interface enables the user to adjust those parameters that are especially important for the physical study being performed. Before getting to that stage, the gross accelerator outline has to be established.

3.2. An Equal Tune FODO Lattice

We start with the `eq_tune_fodo.adxf` input file. When UAL is run, starting from this file (or possibly from the file `eq_tune_fodo.sxf` derived from it) the needed parameters are calculated using formulas given in this section and the results are echoed for comparison. These formulas are coded into the `eq_tune_fodo.adxf` input file using variable names similar to the names used here. A listing of the file is in Table 3.1. Depending on the status of the ADXF parser, a different syntax for `element`'s and/or `sector`'s may be required. Either these changes can be made or the `eq_tune_fodo.sxf` variant used instead.

Our immediate purpose then, for a simple accelerator lattice, is to give prescriptions by which the parameters of a first-cut design can be obtained. Later the properties can be compared with the more accurate values that UAL calculates. This is intended to serve the pedagogical purpose of showing the determining relationships.

There are various reasons analytic formulas can be inaccurate. Some of these are: thick element effects, presence of errors, chromatic effects, and dipole focusing. For various reasons then, one should not be surprised by disagreements in “absolute” quantities at, say, the few percent level. The accuracy of “relative” quantities, for example the change in some lattice parameter when some element strength is changed, can be almost arbitrarily accurate.

TABLE 3.1. The eq_tune_fodo.adxf lattice file.

```
<?xml version="1.0" encoding="UTF-8"?><adxf xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="file:/home/ualusr/USPAS/ual1/doc/adxf/adxf.xsd">
    <constants>
        <!--  nhalf*celltuni must be integer; number of "fullcell"s in "lattice" must be nhalf/2 -->
        <constant name="pi" value="3.14159265359"/>
        <constant name="twopi" value="2*pi"/>
        <constant name="c" value="299792458.0"/>
        <constant name="nhalf" value="20"/>
        <constant name="scale" value="1/20"/>
        <!-- tamper with scale at your own risk. with scale=1.0/20, the half cell length with
             10 cells is 1 meter, circumference 20 m.To scale up the number of cells, change nhalf,
             leaving scale fixed. This assumes
                 - momentum~(nhalf)^2
                 - constant phase advance per cell and constant magnetic field
                 - bend per dipole = 2*pi/nhalf ~ ld/momentum ~ lhalf/momentum
                 - therefore, lhalf ~ nhalf                                     -->
        <constant name="dipfrac" value="0.9994"/>
        <constant name="quadfrac" value="0.0002"/>
        <constant name="sextfrac" value="0.0001"/>
        <constant name="nufrac" value="0.25"/>
        <constant name="celltuni" value="0.20"/>
        <constant name="lhalf" value="scale*nhalf"/>
        <constant name="ld" value="dipfrac*lhalf"/>
        <constant name="lq" value="quadfrac*lhalf"/>
        <constant name="ls" value="sextfrac*lhalf"/>
        <!--  !  derived parameters -->
        <constant name="deltheta" value="twopi/nhalf"/>
        <constant name="nu" value="0.5*celltuni*nhalf + nufrac"/>
        <constant name="cellmu" value="twopi*nu*2/nhalf"/>
        <constant name="sby2" value="sin(0.5*cellmu)"/>
        <constant name="qp" value="sby2"/>
        <constant name="q" value="sby2/lhalf"/>
        <constant name="kq" value="q/lq"/>
        <constant name="q1" value="q"/>
        <constant name="q2" value="-q"/>
        <constant name="kq1" value="kq"/>
        <constant name="kq2" value="-kq"/>
        <constant name="q1p" value="q*lhalf"/>
        <constant name="q2p" value="-q*lhalf"/>
        <!-- lattice parameters -->
        <constant name="rat" value="(1.0+qp)/(1 -qp)"/>
        <constant name="ratinv" value="1.0/rat"/>
        <constant name="betax1" value="sqrt(rat)/q"/>
        <constant name="betay1" value="sqrt(ratinv)/q"/>
        <constant name="betax2" value="sqrt(ratinv)/q"/>
        <constant name="betay2" value="sqrt(rat)/q"/>
        <constant name="eta1" value="(1.0+qp/2) * deltheta/q/q/lhalf"/>
        <constant name="eta2" value="(1.0 -qp/2) * deltheta/q/q/lhalf "/>
        <constant name="s1" value="q1/eta1"/>
        <constant name="s2" value="q2/eta2"/>
        <constant name="ks1" value="q1/eta1/ls"/>
        <constant name="ks2" value="q2/eta2/ls"/>
    </constants>
    <!--  define magnetic elements -->
    <elements>
        <marker name="mk1"/><marker name="mbegin"/><marker name="mend"/>
        <sbend name="bend" l="lq" angle="deltheta"/>
        <quadrupole name="quadhf" l="lq" k1="kq1"/>
        <quadrupole name="quadvf" l="lq" k1="kq2"/>
        <sextupole name="sext1" l="ls" k2="ks1"/>
        <sextupole name="sext2" l="ls" k2="ks2"/>
    </elements>
    <sectors>
        <sector name="fullcell"
          line="mk1  quadhf sext1 bend sext2 quadvf quadvf sext2 bend sext1 quadhf mk1"/>
        <sector name="lattice"
          line="mbegin fullcell fullcell fullcell fullcell fullcell
                fullcell fullcell fullcell fullcell fullcell mdnd"/>
    </sectors>
</adxf>}
```

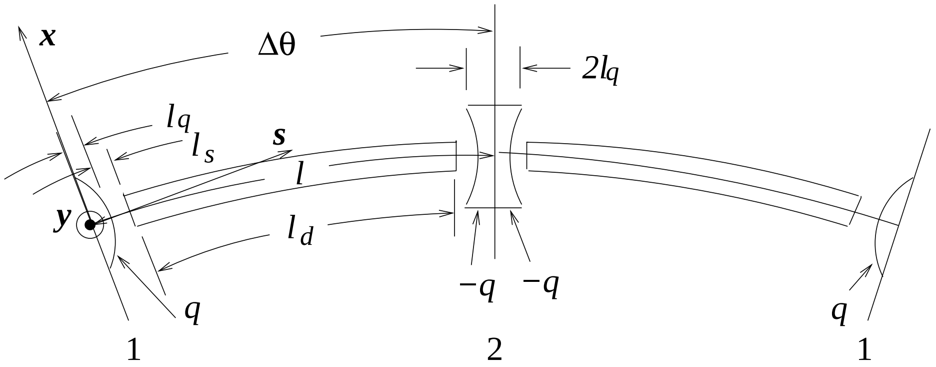


FIGURE 3.1. An idealized, thin lens, FODO lattice, showing dimensioning and element strength parameters.

TABLE 3.2. Parameters of a pure, equal tune, FODO lattice. `nhalf` is an even integer and `nhalf*celltuni` also has to be an integer. The final entry doesn't really belong in this table; it is listed only to call attention to the inconsistency between its symbols `eta` and D.

Quantity	variable name	symbol	expression
scale	`scale`		
number of halfcells	`nhalf`	n	
halfcell (arc) length	`lhalf`	l	
dipole length	`ld`	l_d	`lhalf*dipfrac`
quadrupole length	`lq`	l_q	`lhalf*quadfrac`
sextupole length	`ls`	l_s	`lhalf*sextfrac`
dipole bend angle	`deltheta`	$\Delta\theta$	$2\pi/n$
cell tune (int. tune part only)	`celltuni`		$2\nu_{\rm int}/n$
fractional ring tune	`nufrac`	$\nu_{\rm frac}$	
integer ring tune		$\nu_{\rm int}$	`celltuni*nhalf/2`
ring tune	`nu`	$\nu = \nu_{\rm int} + \nu_{\rm frac}$	
cell phase advance	`cellmu`		$4\pi\nu/n$
halfquad strength	`q`	q	
quad gradient	`kq`	q/l_q	
dispersion	`eta`	D	

A partial glossary of translations from symbols in these notes to symbols in the `.adxf` computer files is given in Table 3.2.

The fractions of `lhalf` taken up by dipole, quadrupole, and sextupole, as shown in the table. Note that the entire arc is filled with magnetic elements—there are no drifts. Of course this is impractical, but this is just a practice lattice. To make realistically long quadrupoles and to introduce the inevitable drifts needed for vacuum pumps, flanges, bellows, etc. length can be taken from the dipoles (which are optically almost "inert") provided that their bend fields are increased to hold the bend angle $\Delta\theta$ constant. When some input parameters are changed (preferably using an XML-aware editor) all other parameters are calculated self-consistently.

But the scaling of some parameters, especially in "highly-tuned" lattices such as the colliding beam lattice, large parameter changes are likely to make the lattice "go unstable". As well as being defined here and in FIG 3.1, many symbols are further defined in comments in the input file. Lengths are scaled by the factor `scale` (which does not otherwise occur explicitly in the analytic formulas given in these notes.) In particular `lhalf` is evaluated as `scale*nhalf`. This "builds in" the relation between circumference and tunes in "typical" accelerators, having "typical" phase advance per cell. Since this is almost certainly confusing, don't worry about it and don't tamper with `scale`, at least initially. With `scale=1.0/20`, the halfcell length with 10 cells `nhalf=20` is 1 meter, and the circumference is 20 m

To a surprising extent accelerator lattices can be described purely in geometric terms, without reference to the particle type or momentum of the particles being accelerated. Still, for some of the modeling instructions, "dynamical" information is required. At that time particle type and particle momentum/energy will have to be assigned.

To a "zero'th" approximation, the generally circular nature of an accelerator can be ignored in calculating the optics. In this approximation the quadrupoles can be regarded as stretched out in a straight line and, for pedagogical simplicity we start with this approach. This neglects a small focusing effect of dipoles. For large rings with many bending magnets this is a small effect, but for small rings the dipole focusing effect is appreciable. As it happens a ring with just ten cells and circumference of 20 m is *small* in the sense that dipole focusing cannot be neglected. To give a lattice with simpler properties, while still just analysing a single cell, `nhalf` can be increased, for example by a factor of ten. For this reason the variable `nhalf` has been set to 200 in the default version of lattice `eq_tune_fodo.adxf`,

For the formulas given here to be as accurate as possible, the quadrupoles and sextupoles should be very short (compared to the half-cell length) since these elements are treated as *thin*; ie. `quadfrac` $<< 1$, `sextfrac` $<< 1$. This restriction applies to initial investigation only. Later the elements can be made realistically thick and, if desired, be segmented for greater accuracy. The UAL environment makes available various evolution "engines". When using the original TEAPOT thin element code, to preserve symplecticity, elements that are, in fact, thick are modeled by segmenting them artificially into thin elements. Originally even ideal bending magnets had to be treated this way for good accuracy. But by now UAL analytic, thick element formulas are available in UAL for treating ideal bending magnets directly as thick elements. If field nonuniformity needs to be modeled, in combined function magnets for example, the old thin element segmentation is employed.

Notationally subscript 1 identifies the start of the first half-cell (or equally the end of the second half-cell, which is the start of the next cell) and 2 identifies the mid-cell location (at the quad center.) The cell layout, dimensions and element strength parameters are indicated in FIG 3.1. The difference in effect of a sextupole between horizontal and vertical is more complicated than just switching the sign, but we defer consideration of this.

In a later section a more general FODO description, permitting unequal quad strengths $|q_1| \neq |q_2|$ and unequal tunes will be described, and many more symbols will be introduced. The UAL filenames for that discussion are "general_fodo.adxf" and "general_fodo.adxf". The reader will be expected to figure out the translations of symbols in that file without benefit of a glossary either because they are mnemonic or because they are the same (or almost the same) as symbols defined here.

3.2.1. Longitudinal Variation of the Lattice Functions.

In the drift regions between quadrupoles the β-functions are quadratic functions of the longitudinal coordinate s, with the origin of s taken to be at location 1. At the quad center the slope $\beta' = d\beta/ds \equiv -2\alpha$ vanishes (by symmetry when the lattice is made of repeated identical cells) but there are slope discontinuities related to the quad strengths by

$$\Delta\beta'^{(x)} = -2q\beta_1^{(x)}, \quad \Delta\beta'^{(y)} = 2q\beta_1^{(y)}, \tag{3.1}$$

so the other Twiss parameters at the quadrupole exit are given by

$$\alpha_{1+}^{(x)} = q\beta_1^{(x)}; \qquad \alpha_{1+}^{(y)} = -q\beta_1^{(y)}; \tag{3.2}$$

$$\gamma_{1+}^{(x)} = \frac{1+q^2(\beta_1^{(x)})^2}{\beta_1^{(x)}}; \quad \gamma_{1+}^{(y)} = \frac{1+q^2(\beta_1^{(y)})^2}{\beta_1^{(y)}}; \tag{3.3}$$

In the region from 1 to 2 the β-functions are given by

$$\beta^{(x)}(s) = \beta_1^{(x)} - 2\alpha_{1+}^{(x)}s + \gamma_{1+}^{(x)}s^2, \quad \beta^{(y)}(s) = \beta_1^{(y)} - 2\alpha_{1+}^{(y)}s + \gamma_{1+}^{(y)}s^2. \tag{3.4}$$

By substituting $s = l$, it can be checked that $\beta^{(x)}(l)$ agrees with $\beta_2^{(x)}$ as determined by Eq. (3.16). The β-functions in the region from 2 to 1 can be obtained by symmetry.

The horizontal dispersion function $\tilde{D}_x \equiv \tilde{D}$ through the cell can be determined similarly. (With no vertical deflections there is no vertical dispersion.) The slope of the dispersion function vanishes at the quadrupole center, but there is a slope discontinuity at 1, due to the quadrupole, so that

$$\tilde{D}'_{1+} = -q\tilde{D}_1. \tag{3.5}$$

There is a slope discontinuity $\Delta\theta$ at $l/2$ due to the bend (which is being treated as if concentrated at the center of the half cell). Using the fact that in drifts and bends the dispersion function propagates like a particle displacement, $\tilde{D}_2$ can be obtained from $\tilde{D}_1$ and then $\tilde{D}_1$ can be obtained from $\tilde{D}_2$;

$$\tilde{D}_2 = \tilde{D}_1 - q\tilde{D}_1 l + \Delta\theta\frac{l}{2}, \quad \tilde{D}_1 = \tilde{D}_2 + q\tilde{D}_2 l + \Delta\theta\frac{l}{2}. \tag{3.6}$$

Solving yields

$$\tilde{D}_1 = \frac{(1+ql/2)\Delta\theta}{q^2 l}, \quad \tilde{D}_2 = \frac{(1-ql/2)\Delta\theta}{q^2 l}, \tag{3.7}$$

with the useful consequence that an average, or "typical" value of the dispersion function is

$$\tilde{D}_{typ} = \frac{\tilde{D}_1 + \tilde{D}_2}{2} = \frac{\Delta\theta}{q^2 l}. \tag{3.8}$$

3.2.2. Establishing Quadrupole Strengths. Transfer matrices for a quadrupole of strength q are

$$M^{(x)} = \begin{pmatrix} 1 & 0 \\ -q & 1 \end{pmatrix}, \quad M^{(y)} = \begin{pmatrix} 1 & 0 \\ q & 1 \end{pmatrix}. \tag{3.9}$$

Note that an explicit negative sign appears with q where it enters the horizontal transfer matrix; this means that positive q corresponds to *focusing* in the x-plane (horizontal). The cell layout was shown previously. With quad locations labeled 1 and 2, the quadrupole strengths have been set equal and opposite, $q_1 = q$ and $q_2 = -q < 0$. With the choice $q > 0$, location 1 is a horizontally focusing point. A bend through angle $\Delta\theta$ is assumed to occur at the center of each half cell, and sextupoles of strength S_1 and S_2 are located immediately adjacent to the quads. In the y plane there are no dipole deflections and the quadrupole sign reversals are indicated. The x transfer matrix $2 \leftarrow 1$ is

$$M_{21}^{(x)} = \begin{pmatrix} 1 & 0 \\ q & 1 \end{pmatrix} \begin{pmatrix} 1 & l \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ -q & 1 \end{pmatrix} = \begin{pmatrix} 1-ql & l \\ -q^2 l & 1+ql \end{pmatrix}. \tag{3.10}$$

As usual the matrix furthest to the right corresponds to the element furthest to the left; the notation $2 \leftarrow 1$ and the order of subscripts on M_{21} are intended to represent this. There is a similar matrix for $1 \leftarrow 2$, obtained by reversing the sign of q. For the full cell $1 \leftarrow 1$,

$$M_{11}^{(x)} = \begin{pmatrix} 1+ql & l \\ -q^2 l & 1-ql \end{pmatrix} \begin{pmatrix} 1-ql & l \\ -q^2 l & 1+ql \end{pmatrix} = \begin{pmatrix} 1-2q^2l^2 & 2l(1+ql) \\ -2q^2l(1-ql) & 1-2q^2l^2 \end{pmatrix}. \tag{3.11}$$

For a periodic lattice made by repeating these cells, this matrix can be written in "Twiss" form, with α vanishing by symmetry, which is consistent with the 11 and 22 elements being equal;

$$M_{11}^{(x)} = \begin{pmatrix} \cos\mu_1^{(x)} & \beta^{(x)} \sin\mu_1^{(x)} \\ -\sin\mu_1^{(x)}/\beta^{(x)} & \cos\mu_1^{(x)} \end{pmatrix}, \tag{3.12}$$

The subscript 1 μ_1 indicates that it applies to one cell. Equating coefficients and including y motion by switching the sign of q leads to

$$C_1 = \cos\mu_1^{(x)} = \cos\mu_1^{(y)} = 1 - 2q^2l^2. \tag{3.13}$$

This leads to a simple relation among q, l, and $\mu_1^{(x)} = \mu_1^{(y)} = \mu_1$;

$$\sin\frac{\mu_1}{2} = ql. \tag{3.14}$$

The "tune advance per cell" is $\nu_1 = \frac{\mu_1}{2\pi}$. If an entire ring is formed from n half-cells, the tune of the ring is

$$\nu = \nu_{int} + \nu_{frac} = \frac{n}{2}\nu_1 \tag{3.15}$$

The β-functions are obtained by equating off-diagonal elements in Eqs. (3.11) and (3.12).

$$\beta_1^{(x)} = \sqrt{\frac{1+ql}{1-ql}}\,\frac{1}{q}, \quad \beta_2^{(x)} = \sqrt{\frac{1-ql}{1+ql}}\,\frac{1}{q}, \tag{3.16}$$

$$\beta_1^{(y)} = \sqrt{\frac{1-ql}{1+ql}}\,\frac{1}{q}, \quad \beta_2^{(y)} = \sqrt{\frac{1+ql}{1-ql}}\,\frac{1}{q}.$$

Note the identities

$$\frac{\beta_1^{(x)}}{\beta_2^{(x)}} = \frac{1+ql}{1-ql}; \quad \frac{\beta_1^{(y)}}{\beta_2^{(y)}} = \frac{1-ql}{1+ql}. \tag{3.17}$$

Also a geometric mean or "typical" β-function value is

$$\beta_{\mathrm{typ}} = \sqrt{\beta_1^{(x)}\beta_2^{(x)}} = \sqrt{\beta_1^{(y)}\beta_2^{(y)}} = \frac{1}{q}. \tag{3.18}$$

The β-functions calculated by UAL are plotted in FIG 3.2, for `nhalf=20`. To simplify the calculations (by reducing the importance of dipole focusing) a larger value `nhalf=200` is suggested for the following exercises.

TUTORIAL 3.1. *To practice correlating computer variable names with mathematical symbols, fill in the remaining entries in column 2 of the Tutorial Worksheet.*

TUTORIAL 3.2. *After running UAL with the* `eq_tune_fodo` *input file, fill in the blanks of columns 4 and 5 of the Worksheet. Certain entries, such as lengths, need not be entered, as they are obviously the same in all columns. Lattice functions, such as β_x, are to be obtained both using the numerical readout at the mouse position in the GUI application (to be entered in column 4) and from the file the GUI can be instructed to generate (column 5). Except for accuracy the entries in these columns are supposed to be redundant.*

TUTORIAL 3.3. *The purpose for this exercise is to practice editing and processing input files but the result will also be useful in making the next simulation agree more accurately with the theoretical formulas (by reducing the importance of dipole focusing,) Using the* `oxygen` *editor, edit the input* `eq_tune_fodo.adxf` *file to change* `nhalf` *to 200. It is necessary to include the corresponding number of* `fullcell` *elements in the* `lattice` *line. This change will have reduced the bend per dipole magnet by a factor of 10. Process the file using the instruction*

```
~/TOY-LATTICES/xslt/SCRIPTS/process-qfile eq_tune_fodo
```

SIMULATION 3.1. *Change* `nufrac` *by an amount small enough that the lattice remains stable but large enough that the tabular entries in the "modified values" column differ from entries in the "sample value" column by a numerically significant amount. Save these data for a problem in the next section (or plan to regenerate them later.)*

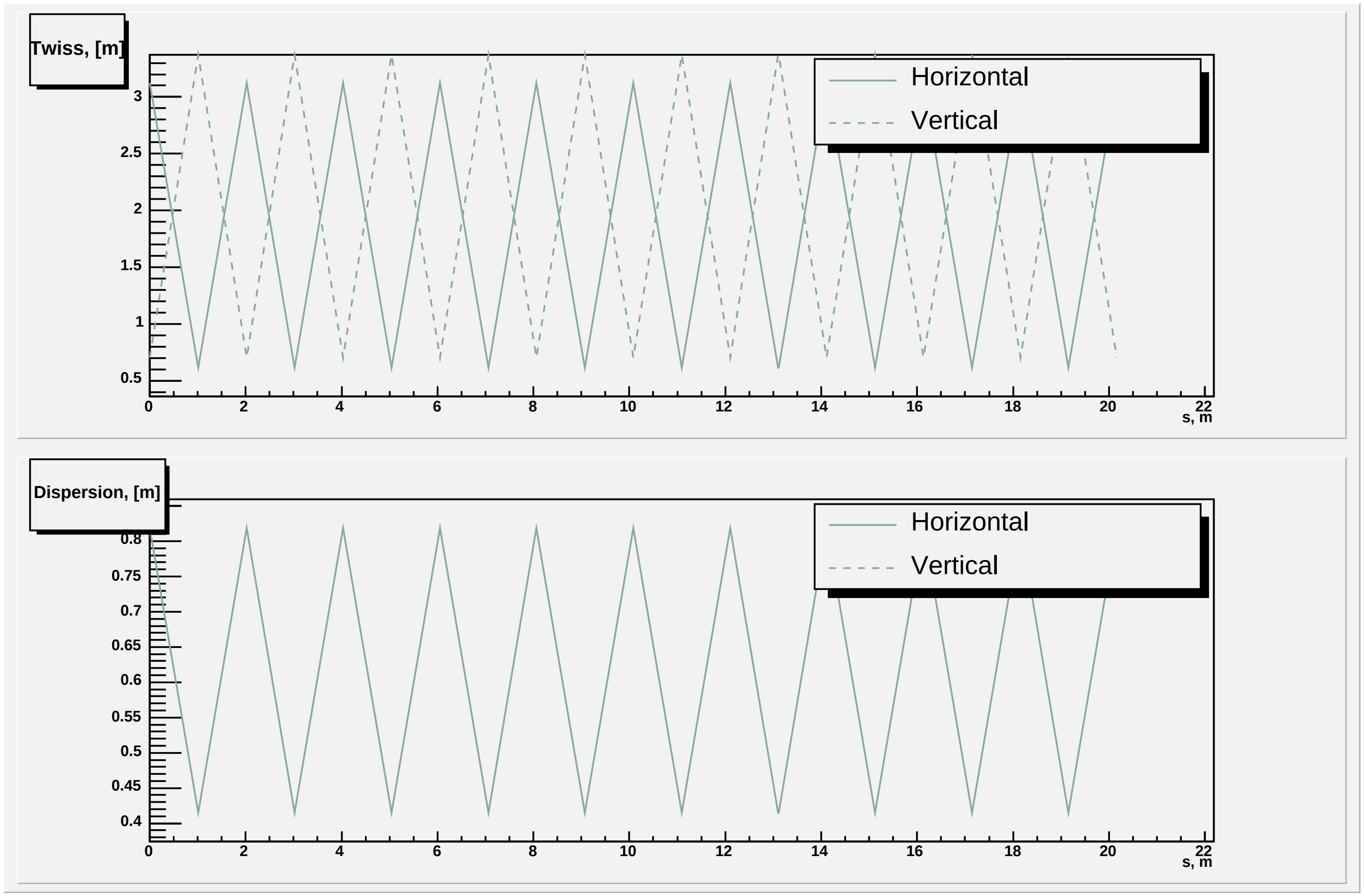


FIGURE 3.2. Lattice functions for the `eq_tune_fodo` lattice. `nhalf=20`.

Table 3.3. TUTORIAL WORKSHEET I. Copy as needed.

variable name	symbol	sample value	UAL plot	UAL file	modified val	UAL value
scale		0.05				
nhalf		20				
ls		0.0001				
lq		0.0002				
lhalf		1.0				
ld		0.9994				
celltuni		0.2				
nufrac		0.25				
deltheta		0.314159				
q2	q_2	-0.70710				
q1		0.70710				
s2	S_2	-1.7409				
s1	S_1	0.83144				
nu						
cellmu		1.41371				
betax1		3.3400				
betax2		0.70984				
betay1		0.70984				
betay2		0.70984				
eta1		0.9867				
eta2		0.5030				

3.2.3. Chromaticity Compensation. Nothing has been said so far about the sextupoles present in the lattice. Their purpose is to correct chromaticity, which quantifies the dependence of tune on momentum. Chromaticities for the two planes are defined by

$$Q'_x = \frac{dQ_x}{d\tilde{\delta}}, \quad Q'_y = \frac{dQ_y}{d\tilde{\delta}}. \tag{3.19}$$

The symbols χ_x and χ_y are also often used for chromaticities. The fundamental cause of chromaticity is the inverse dependence of quad strength on momentum. An off-momentum particle passing through a quad of strength q can treated, to a lowest approximation, as being on-momentum, but with a focusing perturbation of strength $-\tilde{\delta}q$. But there is also a shift of the off-momentum orbit which, if there is a sextupole superimposed on the quad, also provides a focusing perturbation. The field dependence of a sextupole magnet is derived in Chapter 8. For now all that is required is that a sextupole os strength S causes horizontal deflection equal to $Sx^2/2$.

In order to contribute no chromaticity, the combination of a sextupole of strength S_1 superimposed on a quadrupole of strength q must be arranged to suppress the term proportional to $\tilde{\delta}x$ in the deflection

$$q(1-\tilde{\delta})x + \frac{1}{2}S_1(x + \tilde{D}_1\tilde{\delta})^2, \tag{3.20}$$

and a similar relation can be written at location 2. Assuming that "nominal" sextupole strengths should correspond to zero chromaticity, this leads to the values

$$S_1 = \frac{q}{\tilde{D}_1}; \quad S_2 = \frac{-q}{\tilde{D}_2}. \tag{3.21}$$

Note that the sextupole strengths are unequal even though the linear optics is the same in both planes. The sextupole located at a vertical focusing quad has to be stronger because the horizontal dispersion is less there. The chromatic compensation power is proportional to the quadrupole field at a displacement value $\tilde{D}\tilde{\delta}$, a factor which is smaller at vertically focusing quads.

3.3. A Universal, Unequal Phase Advance FODO Lattice

3.3.1. The Twiss Parameters in Terms of the Quadrupole Strengths. Formulas in this chapter relate to the file `general_fodo.xsl`. Variation from what has gone before includes allowing the horizontal and vertical tunes to be different, compensating for dipole focusing, and defining parameters needed to analyse longitudinal motion.

In practice the capability to have greatly different horizontal and vertical tunes is often not needed. Though the fractional tunes are almost always separated in functioning accelerators, the integer tunes are often close. In this case the separation in fractional tunes can usually be achieved as a perturbation away from the symmetric tune situation. (This operation can be performed using the `tunethin` instruction of UAL.) Nevertheless, for greater flexibility, it is convenient to have a closed form, unequal tune, basis lattice. Transfer matrices for a quadrupole of strength q were given in Eq. (3.9). We now introduce unequal quad strengths q_1 and q_2, labeled 1 and 2, without yet specifying which is focusing in which plane. Recall that positive q corresponds to *focusing* in the x-plane (horizontal). One of q_1 and q_2 will be positive, the other negative. A bend through angle $\Delta\theta$ is assumed to occur at the center of each half cell, and sextupoles of strength S_1 and S_2 are located beside the quads.

The x transfer matrix $2 \leftarrow 1$ is

$$M_{21}^{(x)} = \begin{pmatrix} 1 & 0 \\ -q_2 & 1 \end{pmatrix} \begin{pmatrix} 1 & l \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ -q_1 & 1 \end{pmatrix} = \begin{pmatrix} 1-q_1 l & l \\ -q_1 - q_2 + q_1 q_2 l & 1 - q_2 l \end{pmatrix}, \tag{3.22}$$

and a similar matrix for $1 \leftarrow 2$ is obtained by switching q_1 and q_2. The full cell, $1 \leftarrow 1$, x-transfer matrix is

$$\begin{aligned} M_{11}^{(x)} & \begin{pmatrix} 1-q_2 l & l \\ -q_1 - q_2 + q_1 q_2 l & 1 - q_1 l \end{pmatrix} \begin{pmatrix} 1-q_1 l & l \\ -q_1 - q_2 + q_1 q_2 l & 1 - q_2 l \end{pmatrix} \\ &= \begin{pmatrix} 1 - 2q_1 l - 2q_2 l + 2q_1 q_2 l^2 & 2l(1-q_2 l) \\ 2(-q_1 - q_2 + q_1 q_2 l)(1 - q_1 l) & 1 - 2q_1 l - 2q_2 l + 2q_1 q_2 l^2 \end{pmatrix}. \end{aligned} \tag{3.23}$$

For a periodic lattice made by repeating these cells, this matrix can be written in "Twiss" form, with α again vanishing by symmetry;

$$M_{11}^{(x)} = \begin{pmatrix} \cos\mu_1^{(x)} & \beta^{(x)} \sin\mu_1^{(x)} \\ -\sin\mu_1^{(x)}/\beta^{(x)} & \cos\mu_1^{(x)} \end{pmatrix}. \tag{3.24}$$

Equating coefficients and generalizing to include y motion by switching the signs of q_1 and q_2 leads to

$$\begin{gathered} C^{(x)} = \cos\mu_1^{(x)} = 1 - 2q_1 l - 2q_2 l + 2q_1 q_2 l^2, \quad \sin^2\frac{\mu_1^{(x)}}{2} = q_1 l + q_2 l - q_1 q_2 l^2, \\ C^{(y)} = \cos\mu_1^{(y)} = 1 + 2q_1 l + 2q_2 l + 2q_1 q_2 l^2, \quad \sin^2\frac{\mu_1^{(y)}}{2} = -q_1 l - q_2 l - q_1 q_2 l^2 \end{gathered} \tag{3.25}$$

The β-functions are obtained similarly;

$$\begin{aligned} \beta_1^{(x)} &= l\sqrt{\frac{1-q_2 l}{1-q_1 l}}\sqrt{\frac{1}{q_1 l + q_2 l - q_1 q_2 l^2}} = l\sqrt{\frac{1-q_2 l}{1-q_1 l}}\sqrt{\frac{2}{1-C^{(x)}}} = \sqrt{\frac{1-q_2 l}{1-q_1 l}}\frac{l}{\sin\frac{\mu_1^{(x)}}{2}}, \\ \beta_1^{(y)} &= l\sqrt{\frac{1+q_2 l}{1+q_1 l}}\sqrt{\frac{1}{-q_1 l - q_2 l - q_1 q_2 l^2}} = l\sqrt{\frac{1+q_2 l}{1+q_1 l}}\sqrt{\frac{2}{1-C^{(y)}}} = \sqrt{\frac{1+q_2 l}{1+q_1 l}}\frac{l}{\sin\frac{\mu_1^{(y)}}{2}}, \\ \beta_2^{(x)} &= l\sqrt{\frac{1-q_1 l}{1-q_2 l}}\sqrt{\frac{1}{q_1 l + q_2 l - q_1 q_2 l^2}} = l\sqrt{\frac{1-q_1 l}{1-q_2 l}}\sqrt{\frac{2}{1-C^{(x)}}} = \sqrt{\frac{1-q_1 l}{1-q_2 l}}\frac{l}{\sin\frac{\mu_1^{(x)}}{2}}, \\ \beta_2^{(y)} &= l\sqrt{\frac{1+q_1 l}{1+q_2 l}}\sqrt{\frac{1}{-q_1 l - q_2 l - q_1 q_2 l^2}} = l\sqrt{\frac{1+q_1 l}{1+q_2 l}}\sqrt{\frac{2}{1-C^{(y)}}} = \sqrt{\frac{1+q_1 l}{1+q_2 l}}\frac{l}{\sin\frac{\mu_1^{(y)}}{2}}. \end{aligned} \tag{3.26}$$

Note the simple identities,

$$\sqrt{\beta_1^{(x)}\beta_2^{(x)}} = \frac{l}{\sin\frac{\mu_1^{(x)}}{2}}, \quad \sqrt{\beta_1^{(y)}\beta_2^{(y)}} = \frac{l}{\sin\frac{\mu_1^{(y)}}{2}}. \tag{3.27}$$

and

$$\frac{\beta_1^{(x)}}{\beta_2^{(x)}} = \frac{1-q_2 l}{1-q_1 l}, \quad \frac{\beta_1^{(y)}}{\beta_2^{(y)}} = \frac{1+q_2 l}{1+q_1 l}. \tag{3.28}$$

Often $\mu^{(x)}$ and $\mu^{(y)}$ are approximately equal. If they are exactly equal, the formulas simplify considerably. Taking point 1 to be a horizontally focusing quadrupole location we define

$$q_1 = -q_2 = |q|, \tag{3.29}$$

we obtain

$$\cos\mu_1 = 1 - 2|q|^2 l^2, \quad \sin\frac{\mu_1}{2} = |q|l, \tag{3.30}$$

as well as the relations,

$$\begin{aligned} \beta_1^{(x)} &= \sqrt{\frac{1+|q|l}{1-|q|l}}\frac{1}{|q|}, \quad \beta_1^{(y)} = \sqrt{\frac{1-|q|l}{1+|q|l}}\frac{1}{|q|}, \\ \beta_2^{(x)} &= \sqrt{\frac{1-|q|l}{1+|q|l}}\frac{1}{|q|} = \beta_1^{(y)}, \quad \beta_2^{(y)} = \sqrt{\frac{1+|q|l}{1-|q|l}}\frac{1}{|q|} = \beta_1^{(x)} \end{aligned} \tag{3.31}$$

Note that these formulas agree with Eqs. (3.16), which led to identities (3.17) and (3.18).

3.3.2. Longitudinal Variation of the Lattice Functions. Referring again to Table 2.1, in the drift regions between quadrupoles the β-functions vary quadratically with s. At the quad center the slope $\beta' = d\beta/ds \equiv -2\alpha$ vanishes, but there are slope discontinuities related to the quad strengths by

$$\Delta\beta'^{(x)} = -2q_1\beta_1^{(x)}, \quad \Delta\beta'^{(y)} = 2q_1\beta_1^{(y)}, \tag{3.32}$$

so the Twiss parameters at the quadrupole exit are given by

$$\begin{aligned} \alpha_{1+}^{(x)} &= q_1\beta_1^{(x)}, \quad \alpha_{1+}^{(y)} = -q_1\beta_1^{(y)}, \\ \gamma_{1+}^{(x)} &= \frac{1+q_1^2(\beta_1^{(x)})^2}{\beta_1^{(x)}}, \quad \gamma_{1+}^{(y)} = \frac{1+q_1^2(\beta_1^{(y)})^2}{\beta_1^{(y)}}. \end{aligned} \tag{3.33}$$

In the region from 1 to 2 the β-functions vary as

$$\beta^{(x)}(s) = \beta_1^{(x)} - 2\alpha_{1+}^{(x)}s + \gamma_{1+}^{(x)}s^2, \quad \beta^{(y)}(s) = \beta_1^{(y)} - 2\alpha_{1+}^{(y)}s + \gamma_{1+}^{(y)}s^2. \tag{3.34}$$

The slope of the horizontal dispersion function $\tilde{D}(s)$ vanishes at the quadrupole center, but there is a slope discontinuity at 1, due to the quadrupole, such that

$$\tilde{D}'_{1+} = -q_1\tilde{D}_1, \tag{3.35}$$

and a slope discontinuity $\Delta\theta$ at $l/2$ due to the bend (which is being treated as if concentrated at the center of the half cell). As a result, the value of $\tilde{D}_2$ is

$$\tilde{D}_2 = \tilde{D}_1 - q_1\tilde{D}_1 l + \Delta\theta\frac{l}{2}, \quad \tilde{D}_1 = \tilde{D}_2 - q_2\tilde{D}_2 l + \Delta\theta\frac{l}{2}, \tag{3.36}$$

where the same argument has given the second equation also. Solving Eq. (3.36) yields

$$\tilde{D}_1 = \frac{(1-q_2 l/2)l\Delta\theta}{\sin^2\frac{\mu^{(x)}}{2}}, \quad \tilde{D}_2 = \frac{(1-q_1 l/2)l\Delta\theta}{\sin^2\frac{\mu^{(x)}}{2}}. \tag{3.37}$$

For the case of equal tunes as in Eq. (3.29) these become

$$\tilde{D}_1 = \frac{(1+|q|l/2)l\Delta\theta}{|q|l^2}, \quad \tilde{D}_2 = \frac{(1-|q|l/2)l\Delta\theta}{|q|l^2}, \tag{3.38}$$

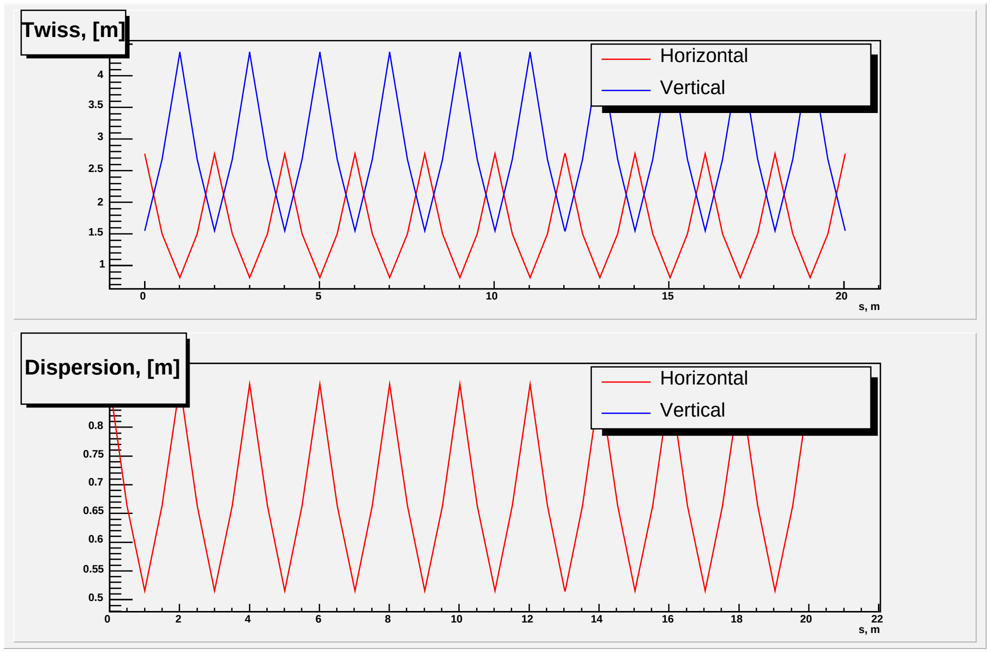


FIGURE 3.3. Lattice functions for the `general_fodo` lattice, with `nhalf=20`, `celltnxi=0.2`, `celltnyi=0.1`.

with the useful consequence that

$$\frac{\tilde{D}_1 + \tilde{D}_2}{2} = \frac{\Delta\theta}{lq^2}. \tag{3.39}$$

Twiss function variation for the `general_fodo` lattice are shown in FIG 3.3.

3.3.3. Setting the Tunes. One can adjust the strengths q_1 and q_2 to achieve desired values for the phase advances $\mu^{(x)}$ and $\mu^{(y)}$. Defining the "average" quantity

$$\overline{\mathcal{S}^2} = \frac{1}{2}\big(\sin^2\frac{\mu^{(x)}}{2} + \sin^2\frac{\mu^{(y)}}{2}\big), \tag{3.40}$$

and the "difference" quantity,

$$\Delta(\mathcal{S}^2) = \sin^2\frac{\mu^{(y)}}{2} - \sin^2\frac{\mu^{(x)}}{2}, \tag{3.41}$$

Eqs. (3.25) become

$$q_1 l + q_2 l = -\Delta(\mathcal{S}^2)/2, \quad q_1 l q_2 l = -\overline{\mathcal{S}^2}. \tag{3.42}$$

These lead to the quadratic equation

$$(q_1 l)^2 + \frac{1}{2}\Delta(\mathcal{S}^2) q_1 l - \overline{\mathcal{S}^2} = 0, \tag{3.43}$$

with the roots being

$$q_1 l = \pm\sqrt{\overline{\mathcal{S}^2} + (\Delta(\mathcal{S}^2))^2/16} - \Delta(\mathcal{S}^2)/4. \tag{3.44}$$

The sign choice depends upon which of the two quads is horizontally focusing—for FODDOF $q_1 > 0, q_2 < 0$, for DOFFOD $q_1 < 0, q_2 > 0$.

3.3.4. Chromaticity Compensation. In order to contribute no chromaticity, the combination of a sextupole of strength S_1 superimposed on a quadrupole of strength q_1 must be arranged to suppress the term proportional to $\tilde{\delta}x$ in the deflection

$$q_1(1-\tilde{\delta})x + \frac{1}{2}S_1(x+\tilde{D}_1\tilde{\delta})^2. \tag{3.45}$$

Assuming that "nominal" sextupole strengths should correspond to zero chromaticity, this leads to the values

$$S_1 = \frac{q_1}{\tilde{D}_1}; \quad S_2 = \frac{q_2}{\tilde{D}_2}. \tag{3.46}$$

For equal tunes the same formulas have been derived earlier.

3.3.5. Compensation For Dipole Focusing. The tune shift caused by a small quadrupole perturbation. A result that is so important in accelerator physics that it deserves to be called "the golden rule" is that a quadrupole perturbation of strength Δq, at a point in the lattice where the beta-functions are β_x and $\beta^{(y)}$, causes tune shifts given by

$$\Delta\nu_x = \frac{1}{4\pi}\beta_x\Delta q, \quad \Delta\nu_y = -\frac{1}{4\pi}\beta_y\Delta q. \tag{3.47}$$

For positive q the horizontal tune is shifted to higher value. The same quad shifts the vertical tune to lower value.

Use of the golden rule to compensate for dipole focusing. There is a focusing effect due a dipole, say a sector bend, that shifts the horizontal tune. Especially in small rings, compensating for this shift improves agreement between desired and achieved tunes. Assume that the magnet lengths satisfy

$$l_d + l_q + l_s = l. \tag{3.48}$$

The effective focusing strength of the dipole (it acts only in the horizontal plane) is

$$q_d = \frac{(\Delta\theta)^2}{l_d}. \tag{3.49}$$

This quadrupole perturbation shifts the tune by an amount

$$\Delta\nu^{(x)} = \frac{n}{4\pi}q_d\overline{\beta^{(x)}} = \frac{n}{2\pi}(\Delta\theta)^2\frac{l/l_d}{\sin\frac{\mu^{(x)}}{2}}. \tag{3.50}$$

where $\overline{\beta^{(x)}}$ has been approximated using Eq. (3.27) and q_d taken from Eq. (3.49). This tune shift is necessarily positive. To compensate for this perturbation, which to this point has been neglected, we apply changes Δq_1 and Δq_2 to q_1 and q_2, applying the condition that both total tune shifts vanish yields

$$\begin{aligned} 4\pi\Delta\nu_1^{(x)} &= 0 = \Delta q_1\beta_1^{(x)} + \Delta q_2\beta_2^{(x)} + q_d\overline{\beta^{(x)}}, \\ 4\pi\Delta\nu_1^{(y)} &= 0 = -\Delta q_1\beta_1^{(y)} - \Delta q_2\beta_2^{(y)}. \end{aligned} \tag{3.51}$$

Solving these equations yields

$$\Delta q_1 = -q_d\frac{\beta_2^{(y)}}{\beta_1^{(x)}\beta_2^{(y)} - \beta_1^{(y)}\beta_2^{(x)}}\frac{l}{\sin\frac{\mu^{(x)}}{2}}, \quad \Delta q_2 = -\Delta q_1\frac{\beta_1^{(y)}}{\beta_2^{(y)}}. \tag{3.52}$$

3.3.6. Orbit Length and Transition Gamma. The `general_fodo.adxf` file also includes calculations that are primarily of importance for longitudinal dynamics. Description of these calculations is deferred until section 6.3 in chapter 6, which deals with longitudinal dynamics. Since that material does not depend on anything not covered so far, the reader wishing to fully understand the `general_fodo.adxf` file could turn to it next.

SIMULATION 3.2. *This simulation continues to use the* `eq_tune_fodo` *lattice. It is deferred to this location because Eq. 3.47 is needed for the analysis. Continuing an earlier simulation, change* `nufrac` *by an amount small enough that the lattice remains stable but large enough that the tabular entries in the "modified values" column differ from entries in the "sample value" column by a numerically significant amount.*

SIMULATION 3.3. *For the input quantity q and the output quantity ν, by performing the subtractions alluded to in the previous problem, evaluate $d\nu/dq$ both for entries from the analytic column and from the UAL column. If the values differ significantly it may because you have changed parameters by too great an amount for "linearized" formulas to be valid.*

CHAPTER 4

Instrumental Analysis of 1D Particle and Bunch Motion

NOTE: This chapter introduces the distinction between idealized single particle motion and the motion of the centroid of a bunch of particles (which is the only thing that is measurable in practice). The methods developed are intended primarily to be applied to bunch centroid motion. In spite of this, only the `transverse` *simulator, which tracks one, or a few, particles individually, will be used for simulations described in this chapter. This can be regarded as testing the data-processing algorithms in simple cases before applying them later in more realistic, more complicated, cases. In Chapter 7, the* `decoherence` *simulator will be applied to the dynamics of bunch centroid motion.*

4.1. Introduction

The lattice that will mainly be used for simulations in this chapter is called `collider_mon` (in either `.adxf` or `.sxf`) form. The investigations have nothing whatsoever to do with colliding beams. The only reason for using this lattice is that it has regions of very small and very large β-function values. This makes it more challenging to extract the β-functions using beam-based methods. The `_mon` in the name indicates that multiple BPM's have been distributed around the ring. These are the only locations at which particle positions are considered (by the simulation) to be known. But, to make the lattice file handier for tutorial purposes, BPM's have also been placed at points which would be physically inaccessible, for example at the intersection points, where elementary particle detectors would occupy the space.

The Twiss functions can be *calculated* for the `collider_mon` lattice, using methods described in Chapter 3.3. The results are shown in FIG 4.1. One of the tasks of this chapter will be to use BPM's to *measure* these functions. This will be an example of the so-called "model independent analysis", in which properties of the lattice are obtained without relying on the design model of the lattice. Of course the beam based *measurements* will only be *simulated* here.

This chapter discusses the instrumentation needed to measure beam properties in an accelerator, and the methods employed to process this data. It might be thought that there would be no need for such experimental apparatus in a simulation context that is entirely theoretical. But the essence of simulation is to replicate actual conditions and to acquire information about the beam using methods that are practical in a control room. Instrumentation issues determine what is available and what is useful.

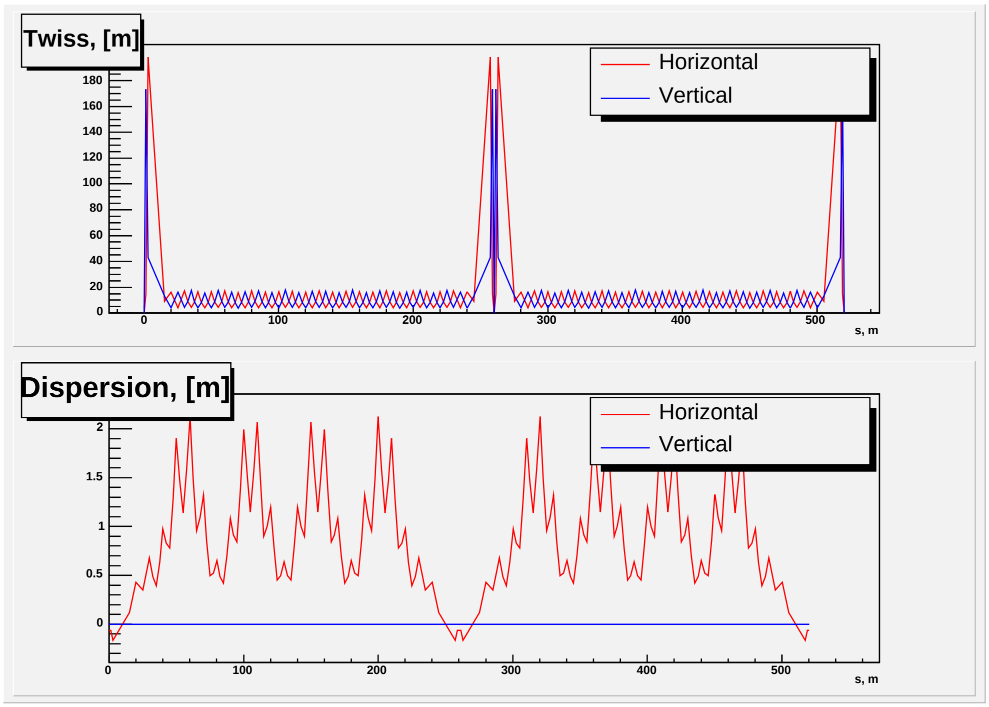


FIGURE 4.1. Lattice functions for the `collider_mon` lattice, obtained using the matrix method of Chapter 2.

Ideal accelerators have only linear, error free elements and highly collimated, monoenergetic, low current beams, that can be measured with noise-free instrumentation. The behavior of such ideal machines can be investigated by studying single particles. None of these idealizations is fully applicable to real accelerators and most of the deviant features are quite difficult to handle by purely analytic calculations. The existence of these non-ideal features is perhaps the greatest justification for investigation by simulation.

The presence of electronic noise limits the accuracy of beam detector measurements. The main tool available for de-sensitization from noise is the use of Fourier-like methods. These permit the coherent superposition of the effects of multiple measurements for which the effects of noise tend to average toward zero. Analysis of such methods is the subject of this chapter.

Since the sources of electronic noise are rarely well understood, the noise level in a simulation has to be represented by one or more empirical coefficients. Even so, for single particle motion, one can, in principle, make the noise negligible by extending the measurement over long enough times.

Unfortunately, many beam detectors respond only to centroid motion of the beam bunches being studied. For low emittance beams this centroid motion is quite accurately the same as single particle motion. In particular the Courant-Snyder invariant of the bunch centroid is, in fact, nearly invariant. But bunches of finite size in nonlinear fields suffer from effects variously known as filamentation, decoherence, and Landau damping, that cause the centroid to exhibit damping-like

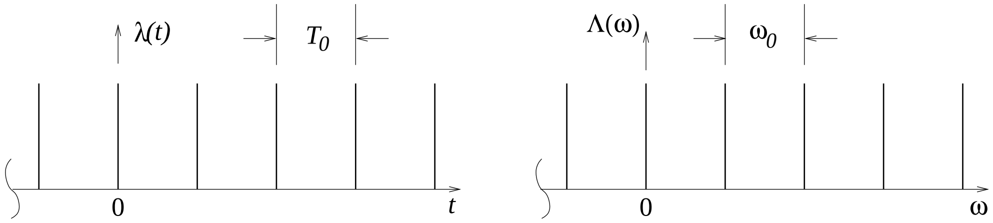


FIGURE 4.2. Time domain and frequency domain signals from a point reference particle.

behavior. The fundamental physics underlying these effects is relatively well known and hence is easily and reliably incorporated into simulation. Filamentation and decoherence is analysed in Chapter 7.

The presence of centroid "damping" means that the single-particle-like motion of a bunch survives for only a limited number of turns. Averaging over times large compared to this is counter-productive as it increases noise without enhancing the signal. This makes it important to attempt to maximize the information extraction from a limited number of terms. A method known as "Principle Component Analysis" (PCA), while effectively subsuming Fourier methods, takes this approach. This method, which UAL uses to analyse one-dimensional motion, is described in later sections of this chapter.

Because the PCA method makes no use of Hamiltonian features, i.e. symplecticity, it is somewhat immune to the presence of centroid damping. But to make progress in the analytic treatment of coupled motion, i.e. two or three dimensional, symplectic features seem to be required. Analysis of coupled motion within UAL is the subject of a later chapter.

4.2. Spectral Analysis of BPM Signals

4.2.1. Spectrum of Reference Particle. Let s stand for the arc length coordinate in a circular accelerator of circumference $\mathcal{C}_0$. A particle of charge e, traveling at speed v_0, on the central orbit passes a fixed point (call it $s = 0$) at regular intervals of time of length $T_0 = \mathcal{C}_0/v_0$. The line charge density, per unit length, corresponding to a single passage of the particle at $t = 0$, is

$$\lambda(t) = e\delta(s) = \frac{e}{|ds/dt|}\delta(t) = \frac{e}{v_0}\delta(t). \tag{4.1}$$

Here λ is that quantity which, when multiplied by a spatial interval ds, yields the charge contained in range ds. Adding all the passages yields

$$\lambda(t) = \frac{e}{v_0}\sum_{l=-\infty}^{\infty}\delta(t - lT_0). \tag{4.2}$$

This is a "comb" of equally-spaced, equal-strength lines in the time domain. A fast, digitized, beam current monitor (BCM) would record the pulse heights of the sequence of pulses shown on the left part of FIG 4.2. This is referred to as longitudinal "turn-by-turn" data.

The same information can be represented analytically, as a sum of terms having sinusoidal time variation, using the easily derived Fourier series relationship

$$\sum_{l=-\infty}^{\infty} \delta(t - lT_0) = \frac{1}{T_0} \sum_{n=-\infty}^{\infty} \cos\frac{2\pi nt}{T_0}. \tag{4.3}$$

Defining a "fundamental" oscillation $\cos\omega_0 t$ where $\omega_0 = 2\pi/T_0$, the current signal can be regarded as the superposition of "harmonics" of the fundamental,

$$\lambda(t) = \frac{e}{v_0 T_0} \sum_{n=-\infty}^{\infty} \cos n\omega_0 t. \tag{4.4}$$

As a check, calculation of the charge in the complete circumference (using the n=0, DC term) correctly yields $\lambda\mathcal{C}_0 = e$.

The Fourier series Eq. (4.4) can be replaced by an integral over a frequency variable ω, that is, as a Fourier integral, by representing the coefficients by δ-functions;

$$\lambda(t) = \frac{e}{\mathcal{C}_0}\int_{-\infty}^{\infty} d\omega \sum_{n=-\infty}^{\infty} \delta(\omega - n\omega_0)\cos\omega t = \int_{-\infty}^{\infty} d\omega\, \Lambda(\omega)\cos\omega t, \tag{4.5}$$

where the frequency domain spectral function $\Lambda(\omega)$ is given by

$$\Lambda(\omega) = \frac{e}{\mathcal{C}_0} \sum_{n=-\infty}^{\infty} \delta(\omega - n\omega_0). \tag{4.6}$$

This shows that the signal is also a "comb" of equally spaced equal strength in the frequency domain. Pictorially the situation is shown in FIG 4.2. Processing the BCM signal with a spectrum analyser would exhibit this spectrum. Typically a bandwidth less than ω_0 would be exhibited and only one line would be visible.

4.2.2. Spectrum of Gaussian Bunch. The line density of a bunch containing unit charge, having Gaussian profile with r.m.s. length σ_s is

$$\lambda = \frac{1}{\sqrt{2\pi}\sigma_s}\exp(-\frac{v_0^2 t^2}{2\sigma_s^2}) = \int_{-\infty}^{\infty} d\omega\, \Lambda(\omega)\cos\omega t, \quad \text{where} \quad \Lambda(\omega) = \exp(-\frac{\sigma_s^2\omega^2}{2v_0^2}). \tag{4.7}$$

Accounting for all beam revolutions, the time domain formula for the line charge of a bunch containing N charges e is

$$\lambda = \frac{Ne}{\sqrt{2\pi}\sigma_s}\sum_{l=-\infty}^{\infty} \exp(-\frac{v_0^2(t - lT_0)^2}{2\sigma_s^2}). \tag{4.8}$$

This can be regarded as the convolution of distributions (4.2) and (4.7). According to a theorem of Fourier analysis, convolution in the time domain corresponds to multiplication, in the frequency domain, of the two transforms. As a result

$$\Lambda(\omega) = \frac{Ne}{\mathcal{C}_0 v_0}\exp(-\frac{\sigma_s^2\omega^2}{2v_0^2})\sum_{n=-\infty}^{\infty} \delta(\omega - n\omega_0). \tag{4.9}$$

Pictorially the situation is shown in FIG 4.3.

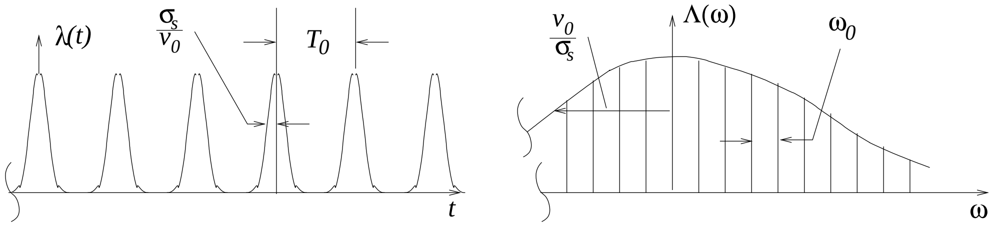


FIGURE 4.3. Time domain and frequency domain signals from an on-axis Gaussian-distributed, bunch of length σ_s.

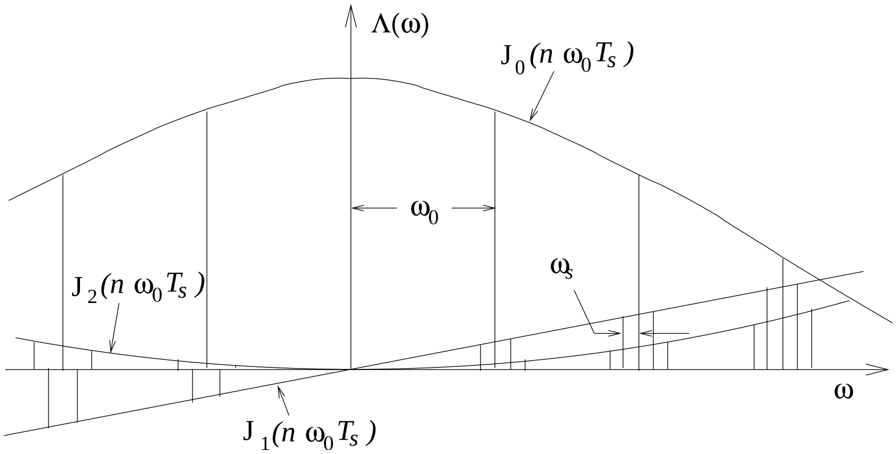


FIGURE 4.4. Spectrum of BCM signal due to a single particle executing longitudinal oscillations.

4.2.3. Spectrum of a Longitudinally Oscillating Particle. Consider a particle that is oscillating longitudinally (relative to the reference particle). The particle's arrival time at the BPM is modulated away from its nominal value by a sinuisoidal factor oscillating at the synchrotron oscillation frequency ω_s, and with longitudinal amplitude $v_0 T_s$. Adjusting the time origin so that the initial oscillation phase is zero, substitution into Eq. (4.2) yields

$$\lambda = \frac{e}{v_0} \sum_{l=-\infty}^{\infty} \delta(t - lT_0 - T_s \cos \omega_s t). \tag{4.10}$$

This "phase modulated" expression can be expressed as a sum of harmonics of the fundamental, along with "synchrotron sidebands" that are displaced away by small integer multiples of the synchrotron frequency. The coefficients in this expansion are proportional to Bessel functions $J_m(n\omega_0 T_s)$, where $m = 0, \pm 1, \pm 2...$ are labels for the sidebands, and $n = 0, \pm 1, \pm 2...$ are labels for the harmonics of the fundamental. Typically the "modulation depth" T_s/T_0 is a very small number, so the arguments of the Bessel functions are very small compared to 1, at least for small n (i.e. low harmonics.) In this case, the leading term, with coefficient $J_0(n\omega_0 T_s)$ is dominant, which makes the sidebands insignificant. At large values of n the sidebands become relatively more important. The situation is illustrated in Fig. 3.

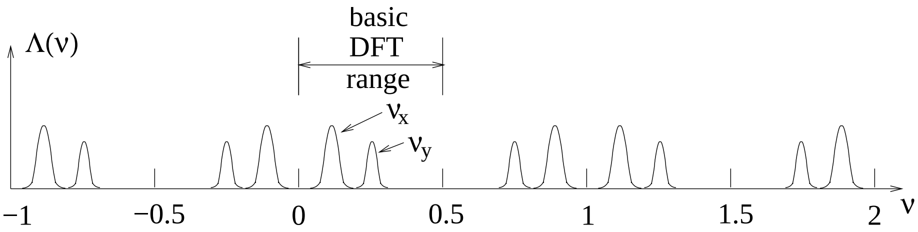


FIGURE 4.5. Cartoon showing spectrum extracted from single plane BPM. Due to coupling both horizontal and vertical tune lines are visible. There is aliasing to all integer tune ranges as well as reflection above and below integers.

4.2.4. Spectrum of a Transversely Oscillating Particle. For a particle undergoing pure horizontal transverse oscillations, the transverse amplitude $x_\beta = \sqrt{\beta_x\epsilon_x}\cos(\nu_x\omega_0 t)$ register as x-dependent currents $\lambda_\pm$;

$$\lambda_\pm = \frac{e}{v_0}(1 \pm A\frac{\sqrt{\epsilon_x\beta_x}\cos(\nu_x\omega_0 t)}{2R}\sum_{l=-\infty}^{\infty}\delta(t-lT_0), \tag{4.11}$$

in current monitors symmetically left and right of the beamline. A beam position monitor (BPM) consists of such a pair of pickups, which produces a signal proportional to x as the difference of the two signals in Eq. (4.11). In Eq. (4.11), with R being the beam pipe radius, the BPM sensitivity A is a dimensionless number of order 1. Performing the subtraction, and substituting from Eq. (4.11), yields a "transverse signal";

$$\begin{aligned}\lambda_+ - \lambda_- &= \frac{e}{v_0}\,\frac{A}{T_0R}\,\sqrt{\epsilon_x\beta_x}\cos(\nu_x\omega_0 t)\sum_{n=-\infty}^{\infty}\cos(n\omega_0 t)\\ &= \frac{e}{v_0}\,\frac{A}{2T_0R}\,\sqrt{\epsilon_x\beta_x}\sum_{n=-\infty}^{\infty}\Big(\cos\big((n+\nu_x)\omega_0 t\big)+\cos((n-\nu_x)\omega_0 t)\Big). \end{aligned}\tag{4.12}$$

One sees that the betatron spectrum consists of identical sidebands, symmetrically above and below all revolution harmonics. In the presence of coupling both horizontal and vertical lines are visible in the same plot, for example as in FIG 4.5. In the presence of nonlinearity even more spectral lines are observed.

The horizontal and vertical tune specta for the `collider_mon` lattice are shown in FIG 4.6. Since this lattice is ideal, and therefore has no x, y coupling, no "wrong plane" lines are visible. There are however hints of lines of unknown origin, for example at $\nu_x = 0.34$. Lines like this will be discussed in Chapter 8. Note, however, with the vertical scale being logarithmic, the amplitudes of thes lines are extremely small.

4.3. Discrete Time, DFT Analysis

The discrete Fourier transform (DFT)[1] is a numerical tool that can be used to determine the coefficients in continuous time expansions like Eq.(4.12). This

[1] A common terminology refers to all discrete Fourier methods as FFT (fast Fourier transform) methods. This is not quite appropriate as the FFT is just one efficient algorithm for evaluating DFT's

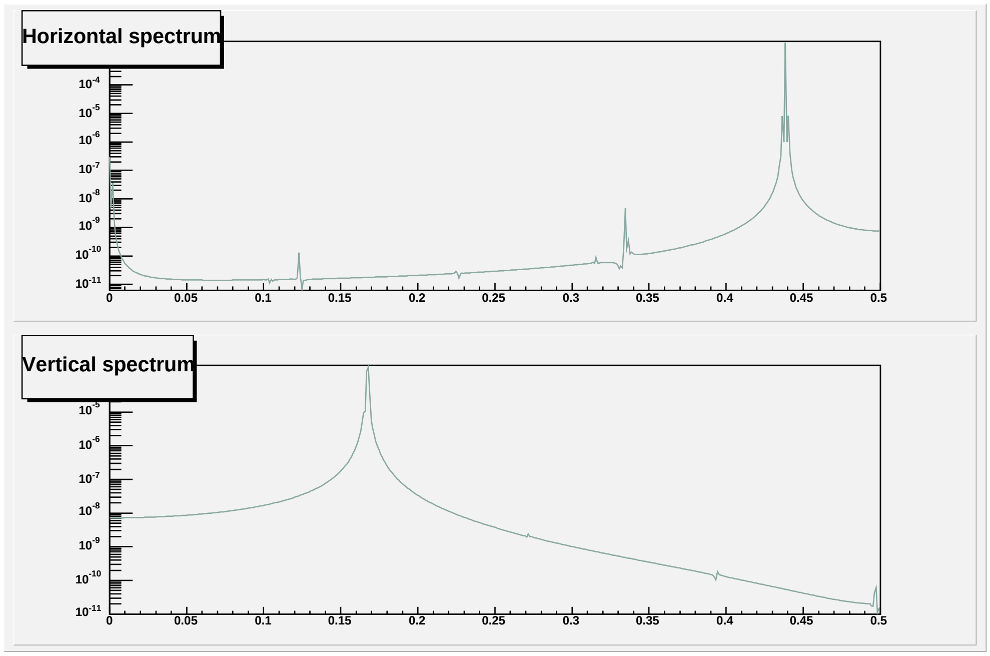


FIGURE 4.6. Horizontal and vertical tune spectra for the `collider_mon` lattice.

procedure entails sampling and digitizing the signal to be analysed at discrete time intervals. For beam detectors in circular rings it is natural to choose the revolution time T_0 as the digitization time. Especially for transverse oscillations, since there are many oscillation periods per revolution period, this represents gross *under-sampling*. This causes "aliasing" in which oscillation at one frequency is detected at another frequency. For the relatively simple spectra under discussion this aliasing does not destroy the usefulness of the spectra and the aliasing is little worse than a nuisance. This will not be further discussed here.

The "natural" domain of the DFT is complex numbers. Wanting to analyse real functions, it is economical to transform two real functions at the same time. Normally one has more (usually many more) than one BPM to analyse, and it is quite satisfactory to process them in pairs. The fundamental DFT formula starts from two sets, each containing an even number N of real samples, $x(n)$ and $y(n)$, of the two functions to be transformed, formed into a single complex sequence $z(n) = x(n) + iy(n)$, $n = 0, 1, \ldots, N-1$. It does no harm to think of this as a complexified read out of horizontal and vertical positions at the same location on turn number n, but the DFT process keeps the x and y sequences separate, so the sequences could just as well have come from different BPM's. The "transform" consists of N complex numbers $Z(n)$, $n = 0, 1, , \ldots, N-1$ given by

$$Z(k) = \frac{1}{N}\sum_{n=0}^{N-1} z(n)\exp\Big(-\frac{i2\pi kn}{N}\Big). \tag{4.13}$$

The output range can be regarded as a fractional tune range from 0 to 1, binned into tune intervals of width $1/N$. (The integer part of the tune is suppressed by the aliasing mentioned earlier.) The N values of $Z(k)$, $k = 1, 2, \dots, N$, can be interpreted as the binned spectral content of the function represented by samples $z(n)$. The inverse transform, the IDFT, is given by

$$z(n) = \sum_{k=0}^{N-1} Z(k) \exp\Big(\frac{i2\pi kn}{N}\Big). \tag{4.14}$$

Essentially the same program can be used for either DFT or IDFT because, with $*$ indicating complex conjugation,

$$x = N(\mathrm{DFT}\{X^*\})^*. \tag{4.15}$$

In this form the Fourier transforms of x and y are "mixed together". To separate them: suppose that $X(k)$ and $Y(k)$ (both complex) are the transforms of x and y, (both real). That is

$$z(n) \leftrightarrow Z(k), \quad x(n) \leftrightarrow X(k), \quad y(n) \leftrightarrow Y(k). \tag{4.16}$$

These show that

$$X(k) = \frac{1}{2}\big(Z^*(N-k) + Z(n)\big), \quad Y(k) = \frac{i}{2}\big(Z^*(N-k) - Z(n)\big). \tag{4.17}$$

There is redundancy in these relations. It can be shown that the $Z(n)$ values are symmetric about $n = (N-1)/2$, which is to say about fractional tune 0.5. This further restriction of the output tune range, already visible in FIG 4.5, is another aliasing effect. Because of this symmetry there is no point in exhibiting spectrum outside the tune range $0 < \nu < 0.5$ as there is no extra information outside this range.

PROBLEM 4.1. *For this and the following problem you are to use any computer language you have access to that is capable of handling matrices; for example* MATLAB, MAPLE, *or a spread sheet. For pedagogical purposes a programmable hand calculator is ideal, even though it may be too slow at processing large matrices in production usage. In the statement of the problem (and similar problems elsewhere in the text), example code valid on an HP calculator will be used to spell out the task. Even for someone unfamiliar with this calculator should find the code simple enough to serve as pseudocode or to be convinced that coding from scratch is not a formidable task.*

Consider a 4×4 *once-around transfer matrix* M *given by*

```
0.231876 -0.029239  -0.031317  0.000000
32.36138  0.231876  -0.822722  0.000000
0.000000  0.000000   1.000000  0.000000
0.822722  0.031317  -0.012908  1.000000
```

Key it into the calculator, or whatever program you are using, as a variable `M`.

For a starting displacement X_0 *such as* $x_0 = 1, x'_0 = 0, y_0 = 1, y'_0 = 0$, *i.e. in the calculator* `[1 0 1 0]`, *iterate the matrix multiplication* $X_{i+1} = MX_i$ `NTR=16` *times to generate simulated data at a single "BPM" as sample data to by subjected to FFT analysis analysis in the next problem. (*`NTR` *can be any integer power of 2.) For example, on an HP calculator, the following code defines a program named* `$XY` *(following the convention that program names start with* `$` *signs). Starting with the initial condition vector on the stack, this code generates* $x + iy$ *for* `N` *turns and stores the sequence of complex numbers in variable* `TRK`*. The actual code appears between the* `<<` *and the* `>>` *signs.*

```
$XY: << DUP DUP 1 GET SWAP 3 GET R->C
        1 N 1 - START SWAP M SWAP *
        DUP DUP 1 GET SWAP 3 GET R->C
        NEXT
        SWAP DROP N ->ARRY 'TRK' STO >>
```

For comparison with a "canned" FFT routine, it is convenient to have, say, a pure $x(n)$ sequence, which can be obtained from similar code;

```
$X: << DUP 1 GET 1 N 1 -
       START SWAP M SWAP * DUP 1 GET NEXT
       SWAP DROP N ->ARRY 'TRK' STO >>
```

Use the `FFT` *provided by the software you are using to calculate the* `DFT` *of the sequence* `TRK`*. Then, as a check, confirm that* `IFFT` *restores the original sequence.*

PROBLEM 4.2. *Using Eq. (4.13), find the* `DFT` *of the turn-by-turn data generated in the previous problem. Then calculate the* `IDFT` *and confirm that the result agrees with the original data set. On the HP calculator the matrix of coefficients needed in Eq. (4.13) can be calculated and stored in matrix* `EJK` *using*

```
$EJK: << N N << -> j k
         << (0.0,1.0) NEG 2 * PI * j * k * N / EXP N / >>
      >>
      LCXM 'EJK' STO >>
```

and a program to produce the FFT of the array on the stack is

```
$DFT: << EJK SWAP * >>
```

Finally, for the `IDFT`

```
$IDFT: << CONJ EJK SWAP * CONJ N * >>
```

For a value such as `N=16` *check that the transform evaluated using* `$DFT` *agrees with the* `FFT` *calculated in the previous problem. Compare the time taken by the* `FFT` *program provided by the software you are using.*

4.4. PCA, Principle Component Analysis

4.4.1. Introduction and Motivation. As stated by Jolliffe[**9**], "The central idea of principal component analysis (PCA) is to reduce the dimensionality of a large number of interrelated variables, while retaining as much as possible of the variation present in the data set". The method can have the heavily statistical emphasis of drawing maximal inference from minimal data sets, or a more purely descriptive, algebraic emphasis on the economical representation of multiple observations by a minimal number of parameters. The diagnosis of an accelerator based on BPM signals is made difficult both by the inherent complexity of the data and by the noise it contains. With the former being judged the more fundamental complication, UAL stresses the algebraic aspect of PCA rather than the statistical. Though the field errors that degrade accelerator performance were random at the time the accelerator was being constructed, they do not contribute stochastically to the sorts of data sets to be investigated, as they are mainly assumed to have been "frozen in" when the accelerator was built.

One way of viewing the spectral analysis of a multiturn BPM signal described in previous sections is that a large number of measurements $z(n)$ (one for each of N turns) has been distilled into a small number of spectral amplitudes. If the tune happens to be an integral multiple of $1/N$, the entire data set is characterized by two numbers, the amplitudes of sine-like and cosine-like oscillations. All other spectral amplitudes are zero, or are at least very small relative to the dominant lines. Even for arbitrary tunes only a few bins have appreciable amplitudes. Taking Eq. (4.13) as example, this distillation has been accomplished by generating numbers that are linear superpositions of data values $z(n)$ with (theoretically derived) coefficients $\exp(-i2\pi kn/N)$. Since there were grounds for expecting different BPM's to exhibit the same tune lines their data sets could be subjected to identical analyses.

If the individual BPM signals were not expected to show essentially sinusoidal variation the Fourier transform procedure would not be expected to work so well. Some extraneous effects merely exhibit the power of Fourier transforms. For example noise tends to average to zero. BPM misalignment, which causes even the reference particle to give transverse signals, gives spectral response at zero frequency, which is easily distinguished from the true transverse lines.

But other effects may not be so benign. One can inquire whether a linear superposition using coefficients other than the sinusoidal Fourier coefficients might be found that achieves comparable distillation even when the signals are theoretically less predictable. The answer is "yes", the method is called "Principle Component Analysis" (PCA) and the distilled parameters are called "Principle Components" (PC). (By this definition, the Fourier transform values could be —but typically are not—legitimately referred to as principle components.)

Before spelling out this procedure it is appropriate to identify essential features of the data. The turn-by-turn data from a single BPM, say the j'th, can be expressed by a vector

$$\mathbf{x}^j = \begin{pmatrix} x_1{}^j \\ x_2{}^j \\ \ldots \\ x_N{}^j \end{pmatrix}. \tag{4.18}$$

A matrix can be formed from the measurements from M BPM's, each arrayed as in Eq. (4.18);

$$\mathbf{X} = \begin{pmatrix} \mathbf{x}^1 & \mathbf{x}^2 & \ldots & \mathbf{x}^M \end{pmatrix} = \begin{pmatrix} x_1{}^1 & x_1{}^2 & \ldots & x_1{}^M \\ x_2{}^1 & x_2{}^2 & \ldots & x_2{}^M \\ x_3{}^1 & x_3{}^2 & \ldots & x_3{}^M \\ x_4{}^1 & x_4{}^2 & \ldots & x_4{}^M \\ x_5{}^1 & x_5{}^2 & \ldots & x_5{}^M \\ & \ldots & & \\ x_N{}^1 & x_N{}^2 & \ldots & x_N{}^M \end{pmatrix} \tag{4.19}$$

(The upper indices have been shifted slightly to the right to make it natural, in matrix operations, to regard the upper index as the second index. Also the matrix has been been given the cosmetic appearance of being higher than it is wide, since that is a requirement of the SVD method to be introduced in the next section; i.e. $N \geq M$, more turns than BPM's.) All entries in this matrix are *commensurate*, meaning they have the same units—in this case length. This facilitates matrix manipulations since the coefficients of transformation matrices can be dimensionless. Though all elements are commensurate, it is essential to realize that the characteristics of columns and the characteristics of rows are different. In performing matrix operations on the matrix $\mathbf{X}$ it is important to use only methods that respect this difference between rows and columns.

Within the matrix $\mathbf{X}$ the row index can be referred to as "temporal", since time increases from top to bottom of the matrix. The elements of any one column map out a "temporal pattern". In purely conservative, or Hamiltonian, motion there will be no systematic tendency for the elements to become small with increasing row index. But, for lossy or decohering systems, the elements will tend to get smaller toward the bottom.

The column index of $\mathbf{X}$, which distinguishes different locations in the ring, can be referred to as a "spatial" index. In principle, in PCA, the different columns could refer to quite dissimilar properties of the system under study. In our case the dynamics at different BPM locations is expected to have quite similar characteristics. But the transverse scale can change discontinuously and erratically in progressing from one BPM to the next. Such variation can be displayed as a spatial "pattern" or "shape". One may have some theoretical expectation of this variation, but in a so-called "model independent analysis" (MIA) this spatial pattern is regarded as *a priori* unknowable, and a major purpose of the analysis is to extract the transverse scaling factors (i.e. $\sqrt{\beta}$-functions) from the data.

There are too many optional features in the properties of data matrices like $\mathbf{X}$ for all options to be discussed here. For example the various columns could be multiple measurements from the same BPM. This is the case to be assumed for the time being. Or each column could be derived from a different BPM, a case that will be adopted later, for purposes of β-function determination. There could be more turns than BPM's, $N > M$ (the only case to be considered here) or *vice versa.*

An important algebraic parameter that helps to distinguish these cases mathematically is the *rank* R, which characterizes the extent to which the columns of $\mathbf{X}$ are algebraically independent. If all columns of $\mathbf{X}$ have come from the same detector, all with proportional initial conditions, all under ideal conditions, then all columns would be proportional and R would be 1. Slightly more general would be the same situation but with random starting conditions, in which case there would be sine-like and cosine-like columns (actually superpositions thereof) and $R = 2$.

In any case R cannot exceed the smaller of M and N. Since we require $N > M$ it follows that $R \leq M$. Beyond this point, determining the value of R becomes a bit squishy. The condition for R to be less than M is that one or more determinants formed from elements of $\mathbf{X}$ vanish. Unfortunately, no experimentally measured quantity can vanish exactly (because there is *always* noise at some level). But the whole point of PCA is to identify linear dependencies among the columns of $\mathbf{X}$—such dependencies can be used to reduce the number of parameters needed to describe the data. If exact, each such (independent) relationship corresponds to a reduction of R.

In spite of this lack of rigor, the following approximate approach often succeeds. For $R = M$ the number of PC's is M. Each PC quantifies the contribution to the data of one *mode.* If the PC is big then the corresponding mode is important. In practice, with "good" data, only a few of the PC's are big and the rest are "small" (without noise or other peripheral effects the small PC's would presumably vanish). Setting the small PC's exactly to zero causes the rank R to be exactly equal to the number of retained PC's. Only the non-zero PC's that survive deserve to be called "principal components". Such a phenomenological procedure is of course only approximate and its validity has to be investigated further on a case-by-case basis. In the simplest nontrivial, but still ideal, case there will be two PC's corresponding to sine-like and cosine-like motions, and $R = 2$.

4.4.2. Rotation to Principle Axes.

The matrix $\mathbf{X}$ contains a large number MN of data points, probably all that will be needed to characterize the lattice, at least in the case where each column represents a different BPM. Even so, one can regard this data as just one sampling of MN random variables, or of M samplings of an N-component random vector. In principle the SVD procedure works even if these variables are mutually independent, but any *useful* inference to be drawn reflects their internal relationships.

Consider multiple samplings of a particular column $\mathbf{x}^j$, and, for each sample, interpret the entries as coordinates of a point to be plotted in an N-dimensional dot plot. To simplify discussion take $N = 3$ and assume that all M columns correspond to the same BPM. The points will be scattered in a roughly ellipsoidal region. A general treatment would permit the vectors $\mathbf{x}^j$ to get contributions from noise, from momentum dependence, from misalignment, etc. Because of effects like this the ellipsoid would not necessarily be centered on the origin but, deferring this possibility for simplicity, let us assume that the theoretical averages of all entries are zero. The different data sets reflect evolution of the same system but with random start times. As such the M data points in any one data set under study will be imbedded indistinguishably within this plot.

It is natural to identify principle axes of this ellipsoid and to perform a *rotation* to coordinates in which the ellipsoid is erect relative to all axes. Being erect implies the absence of correlation between any pair of components. Restricting this transformation to be a *rotation* exploits the commensurate nature of the elements and prevents the distortion that would result when different multiplicative factors are applied to components along different axes.

It could happen that the ellipsoid just discussed is very long and skinny. It would be natural then to choose the axis along which the ellipsoid is long as the first coordinate axis. The component along this axis would indeed be a *principle* component since the motion would be essentially one dimensional, with this component describing the state of motion. Normally the data will be more than one dimensional. The next simplest case would have the data approximately restricted to an elliptical region lying in some skew plane. The first principle component would then be chosen along the major axis, the second along the minor axis. All this discussion generalizes to the N-dimensional region needed to describe all N components of $\mathbf{x}^j$. In general there will be N principle components, ordered from largest to smallest. If a few of these components are "large" and the rest "small", one will have reduced the dimensionality of the data set (in some approximate sense) from N to the number of "large" PC's.

4.4.3. The SVD Method of Determining Principle Components.

The numerical, algebraic "singular value decomposition" (SVD) method is described by Press *et al.*[**10**] Though ancient (Beltrami, 1873, Jordan, 1874) the SVD method was recently introduced into accelerator physics by Irwin, Wang et al[**8**]. Unlike DFT, let us require all components and all coefficients to be real.

SVD is a robust algebraic algorithm, that permits an arbitrary, not necessarily square, $N \times M$ matrix $\mathbf{X}$ to be expressed in the form

$$\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^T = \sigma_1 \mathbf{u_1}\mathbf{v}_1^T + \sigma_2 \mathbf{u_2}\mathbf{v}_2^T + \dots . \tag{4.20}$$

Here $\mathbf{S}$ is an $N \times M$ matrix which, though not square, has non-vanishing elements only along the main diagonal, their values being $\sigma_1 > \sigma_2 > \dots > \sigma_M$.. Matrix $\mathbf{U}$,

$$\mathbf{U} = \begin{pmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \dots & \mathbf{u}_N \end{pmatrix}, \quad \mathbf{u}_i^T \mathbf{u}_j = \delta_{ij}, \tag{4.21}$$

is an $N \times N$ orthonormal matrix whose columns are vectors $\mathbf{u}_i$ that satisfy the orthonormality relations shown. Similarly,

$$\mathbf{V}^T = \begin{pmatrix} \mathbf{v}_1^T \\ \mathbf{v}_2^T \\ \dots \\ \mathbf{v}_M^T \end{pmatrix}, \quad \mathbf{v}_i^T \mathbf{v}_j = \delta_{ij}, \tag{4.22}$$

is an $M \times M$ orthonormal matrix.

Because the vectors $\mathbf{u}_i$ and $\mathbf{v}_j$ are normalized, the magnitudes of the coefficients σ_i in Eq. (4.20) accurately reflect the importance of the individual terms. Being ordered with largest σ_i first, the terms in Eq. (4.20) are ordered from most important to least important.

By spelling out Eq. (4.20) in more detail;

$$\mathbf{X} = \sigma_1 \begin{pmatrix} v_{1,1} \begin{pmatrix} u_{1,1} \\ u_{1,2} \\ \dots \end{pmatrix} & v_{1,2} \begin{pmatrix} u_{1,1} \\ u_{1,2} \\ \dots \end{pmatrix} & \dots \end{pmatrix} + \sigma_2 \begin{pmatrix} v_{2,1} \begin{pmatrix} u_{2,1} \\ u_{2,2} \\ \dots \end{pmatrix} & v_{2,2} \begin{pmatrix} u_{2,1} \\ u_{2,2} \\ \dots \end{pmatrix} & \dots \end{pmatrix} + \dots, \tag{4.23}$$

one sees that $\mathbf{u}_i$ describes the temporal pattern (defined earlier) of the i'th mode and $\mathbf{v}_i$ describes the spatial pattern of the same mode. Since the SVD factorization is unique, and can be performed mechanically using readily available software tools, one sees that considerable information can be extracted automatically from data matrix $\mathbf{X}$.

To obtain the spatial patterns $\mathbf{v}_i$ one can introduce a matrix, referred to as the "sample covariance matrix", and defined by

$$\mathbf{C}\big|_{ij} = \frac{1}{N}\mathbf{X}^T\mathbf{X}\big|_{ij} = \frac{1}{N}\sum_k x_k^i x_k^j . \tag{4.24}$$

The summation is over turn number k and division by N converts the elements to average values;

$$C_{ij} = \begin{pmatrix} < x^1 x^1 > & < x^1 x^2 > & \dots & < x^1 x^M > \\ < x^2 x^1 > & < x^2 x^2 > & \dots & < x^2 x^M > \\ & \dots & & \\ < x^M x^1 > & < x^M x^2 > & \dots & < x^M x^M > \end{pmatrix}. \tag{4.25}$$

Using Eq. (4.20) and the orthonormality of $\mathbf{U}$ and $\mathbf{V}$, one derives

$$\mathbf{C}\mathbf{V} = \frac{1}{N}\mathbf{X}^T\mathbf{X}\mathbf{V} = \frac{1}{N}\mathbf{V}\mathbf{S}\mathbf{U}^T\,\mathbf{U}\mathbf{S}\mathbf{V}^T\,\mathbf{V} = \frac{1}{N}\mathbf{V}\mathbf{S}^2. \tag{4.26}$$

This equation shows that the columns of $\mathbf{V}$ (which have previously been called spatial patterns) are eigenvectors of $\mathbf{C}$. Because it is symmetric, the eigenvalues of $\mathbf{C}$ are known to be real, and its eigenvectors are known to be orthogonal. Also from Eq. (4.26) one infers that the eigenequations for individual eigenvectors are

$$\mathbf{C}\mathbf{v}_i = \lambda_i \mathbf{v}_i, = \mathbf{v}_i \sigma_i^2/N. \tag{4.27}$$

From each eigenvalue λ of $\mathbf{C}$ this determines a diagonal element $\sigma = \sqrt{\lambda N}$ of $\mathbf{S}$.

The matrix $\mathbf{C}$, to the extent it is statistically well determined by the data, enables geometric determination of the principle components. Since the vectors $\mathbf{v}_i$ form an orthonormal set they define a characteristic reference frame. For an arbitrary data vector $\mathbf{x}$, its principle components z_i can be obtained as the components of $\mathbf{x}$ along these axes. In vector algebra these components would be determined as "dot products" $\mathbf{v}_i \cdot \mathbf{x}$. In matrix notation,

$$z_i = \mathbf{v}_i^T \mathbf{x}, \quad \text{or} \quad \mathbf{z} = \mathbf{V}^T \mathbf{x}, \tag{4.28}$$

where, in the second form, the vectors have been arrayed within matrices. Finally, consider the family of N-dimensional ellipsoids defined by the equation

$$\mathbf{x}^T \mathbf{C}^{-1} \mathbf{x} = \text{ constant}. \tag{4.29}$$

Using Eq. (4.28), and the orthonormality of $\mathbf{V}$, this equation can be expressed as

$$\text{constant } = \mathbf{z}^T \mathbf{V}^T \mathbf{C}^{-1} \mathbf{V} \mathbf{z}. \tag{4.30}$$

By manipulating Eq. (4.26), this equation can be transformed to

$$\text{constant } = \mathbf{z}^T \frac{\mathbf{S}^{-2}}{N} \mathbf{z} = \frac{1}{N} \sum_{i=1}^{N} \frac{z_i^2}{\sigma_i^2}, \tag{4.31}$$

which is the equation of the characteristic ellipsoid in the coordinates in which it is erect. This derivation has been formal. The detailed geometric interpretation of the equation depends on the treatment of the previously-mentioned erratic behavior of small σ_i values. Dropping such indeterminate terms from Eq. (4.31), the formula is useful primarily in the case that it reduces to a relatively short sum of squared terms. In this frame the distribution of points can be represented, at least crudely, as a one dimensional distribution of points binned into ellipsoidal shells, with the shell size parameterized by allowing the *constant* appearing on the left hand side of the equation to vary.

4.4.4. Extracting Betatron Phases and Beta Functions. Much of this section follows Wang *et al.*[**7**]. From here on it will be assumed that the M columns in data matrix $\mathbf{X}$ consist of turn-by-turn data from separate BPM's. The main new physics the PCA approach is intended to address is the *non*-invariance of the Courant-Snyder invariant evaluated from a bunch centroid when it is treated as the position of a single particle. Such variation invalidates equations like (4.12) in which the CS invariant ϵ was assumed to be constant. In practice the decoherence effect that causes ϵ to vary is usually fairly weak and the fractional variation of ϵ during any one turn is tiny.

Based on the slowness of this variation, for extracting betatron parameters from the turn-by-turn data matrix $\mathbf{X}$, Wang *et al.*[**7**] suggest the following sensible approximation. During any one turn ϵ is taken to be exactly constant. This means the spatial pattern introduced earlier will be exactly the same as for single particle motion. But, to support decoherence or other non-Hamiltonian behavior, the emittance is permitted to have arbitrary (though slowly varying) temporal dependence. Notationally, the bunch CS-invariant is simply given a temporal index i so that ϵ_i, even though its value varies with i, is constant during any single turn. The centroid motion can therefore be expressed as

$$x_i{}^m = \sqrt{\epsilon_i \beta_m} \cos(\phi_i + \psi_m), \tag{4.32}$$

where β_m is the beta-function and ψ_m the phase at the m-th BPM and ϕ_i is the time-evolving phase at some reference location in the ring (to be referred to as "origin").

As stated already, the measured components of $\mathbf{X}$ will have other extraneous contributions (due, for example, to noise and dispersion) but we will trust the PCA approach to suppress their effects. By substituting from Eq. (4.32) into previously derived formulas we can derive theoretical expressions for the various eigenvalues, eigenvectors, and patterns that have been introduced, in terms of the newly-introduced parameters ϵ_i, β_m, ϕ_m, and ψ_i.

For a given data set, taken at arbitrary time, the starting phase at the origin is arbitrary. The starting phase can be inferred from the elements in the top two rows of $\mathbf{X}$ along with Eq. (4.32). Wang *et al.*[**7**] give an explicit formula (double-valued and not reproduced here) for this phase, as well as the following formulas for the two non-vanishing eigenvalues;

$$\begin{aligned} \lambda_\pm &= \frac{<\epsilon>}{4} \sum_{m=1}^{M} \beta_m (1 + \cos 2\psi_m) \\ &= \frac{<\epsilon>}{4} \Big(\sum_{m=1}^{M} \beta_m \pm \sqrt{(\sum_m \beta_m \cos 2\psi_m)^2 + (\sum_m \beta_m \sin 2\psi_m)^2} \Big). \end{aligned} \tag{4.33}$$

where and initial phase ϕ_0 has been set to zero, by judicious choice of starting turn index.[2] Wang *et al.* also give theoretical formulas for the spatial shapes,

$$\begin{aligned} v_+ &= \frac{<\epsilon>\beta_m}{2\sqrt{\lambda_+}} \cos\psi_m, \quad m = 1, \dots, M, \\ v_- &= \frac{<\epsilon>\beta_m}{2\sqrt{\lambda_-}} \sin\psi_m, \quad m = 1, \dots, M, \end{aligned} \tag{4.34}$$

as well as for the temporal shapes,

$$\begin{aligned} u_+ &= \frac{2\epsilon_i}{N<\epsilon>} \cos\phi_i, \quad i = 1, \dots, N, \\ u_- &= -\frac{2\epsilon_i}{N<\epsilon>} \sin\phi_i. \quad i = 1, \dots, N. \end{aligned} \tag{4.35}$$

Finally these formulas can be used at each BPM to extract, from the SVD expansion, the betatron phase,

$$\psi = \tan^{-1} \Big(\frac{\sqrt{\lambda_-}\, v_-}{\sqrt{\lambda_+}\, v_+} \Big) \tag{4.36}$$

and the β-function,

$$\beta = \frac{2}{<\epsilon>} (\lambda_+ v_+^2 + \lambda_- v_-^2). \tag{4.37}$$

[2] The quantity $<\epsilon>$ in Eq. (4.33) is multiturn average of the possibly-time-varying emittance. If the centroid motion is damped, and the measurement extended to times long compared to the damping time, then this average varies inversely with the number of samples. Obviously there is no point in extending data collection into this region.

4.4.5. The PCA Feature of the `transverse` bSimulator. The following series of figures were obtained using the SVD feature of the `transverse` simulator. FIG 4.7 shows the SVD opening window. Only the leading PC's are shown. All those not shown are tiny. As expected, there are only two large PC's. The third, not quite tiny PC is the subject of one of the assignments.

The spatial and temporal eigenvectors are shown in FIG 4.8. Their interpretations will be discussed further in Chapter 7. The spatial eigenvector is closely related to the β-function variation. In this case the interpretation is complicated by the fact that the β-function is necessarily positive while the eigenvector components can have either sign. The signs are resolved using Eqs. (4.36) and (4.37). The β-functions and phase advances derived from this data are shown in FIG 4.9. The β-function extraction uses Eq. (4.37). Treating the emittance $<\epsilon>$ as unknown, this leaves an overall scale factor undetermined, but the variation of β_x around the ring is determined. This "model-independent" determination can be seen to agree will with the model-based determination. The phase advance determinations using Eq. (4.36) also agree well with matrix-based determinations.

Simulation 4.1. *Vary the noise level in the* `.apdf` *file and investigate the degradation in accuracy of the PCA-determined Twiss functions. The noise level (relative to signal level) which causes a typical degradation of, say,* 10% *in accuracy should be determined.*

Problem 4.3. *Suggest a possible source for the third, not quite negligible PC indicated in FIG 4.7. Derive a quantitative fomula for the magnitude of this PC.*

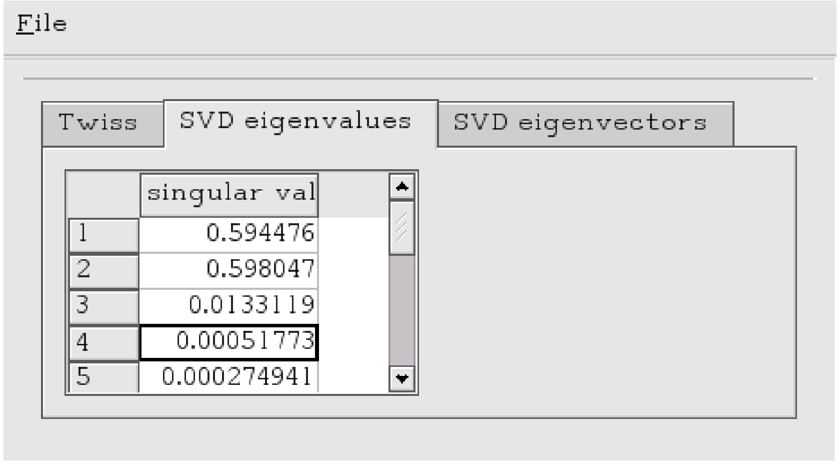
Simulation 4.2. *Compare the PC's obtained when the energy offset δ of the particle is varied. Pay especial attention to the third component and plot its value against δ.*

Simulation 4.3. *Augment the SVD simulation feature so that it also extracts the dispersion at the locations of* `monitor` *elements in the lattice.*

Simulation 4.4. *Radiation damping in an electron accelator causes the x, y, s Courant-Snyder invariants of each particle to be multiplied each turn by factors a_x, a_y, a_s, each less than 1 by an amount that is typically of order one part in 10,000. $ct \rightarrow a_s\, ct$.*

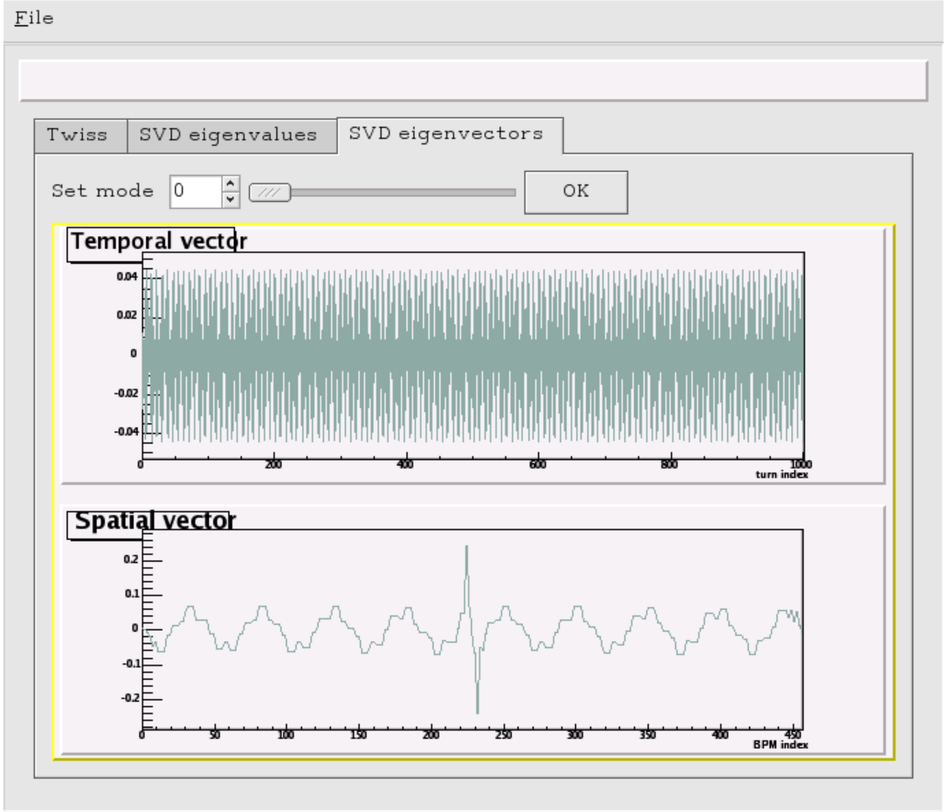
The presence of small deviations like these can be modeled in the simulator in much the same way that noise is simulated. The main differences are that the damping decrements are necessary negative and they are not stochastic. Rather the decrements are proportional to the particle amplitude. For example $x \rightarrow x(1 - a_x)$.

Augment the `transverse` *simulator to model the presence of damping. Measure the damping rate and relate it* accurately *to the damping coefficient a_s you have buile into the code.*



	singular val
1	0.594476
2	0.598047
3	0.0133119
4	0.00051773
5	0.000274941

FIGURE 4.7. Opening window for the UAL PCA simulator. The leading principal components (PC)'s are shown.



Set mode 0

FIGURE 4.8. Spatial and temporal SVD eigenvectors derived from the PCA simulation of the `collider_mon` lattice are exhibited.

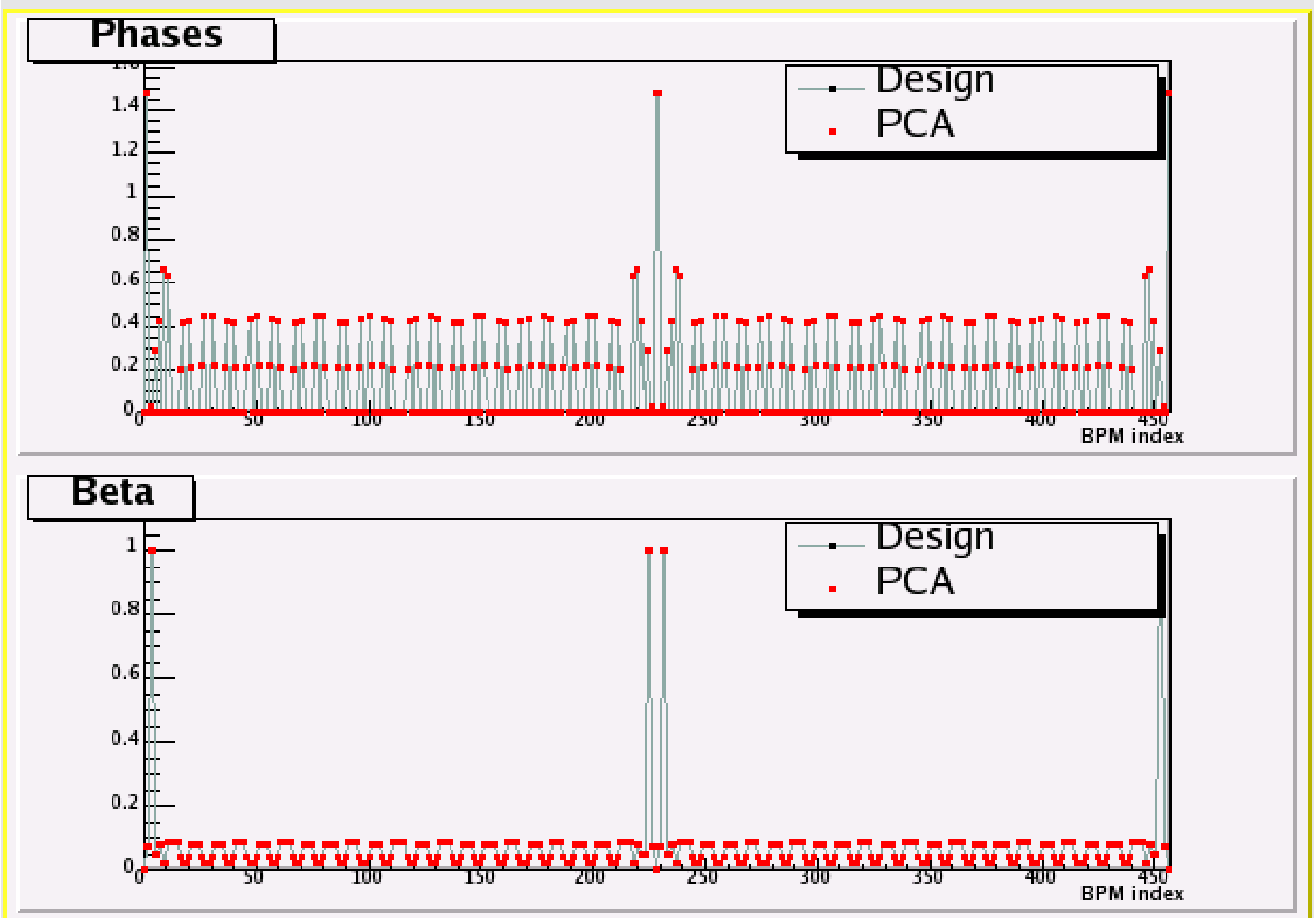


FIGURE 4.9. Twiss function and phase advances derived for the `collider_mon` lattice using the PCA simulator. Bold dots superimposed on curves indicate agreement between model-independent and model-based determinations.

PROBLEM 4.4. *In problem 4.1 a programmable calculator program* `$X` *operated on the 4 components of an initial condition vector to produce a multiturn sequence* `TRK` *by repeated multiplication by a matrix (here called* `MM` *so that* `M` *can stand for the number of BPM's). This code can be re-used here. The routine*

```
$RI: << RAND RAND RAND RAND 4 ->ARRY $X TRK >>
```

repeats this calculation for a random starting position. `$RI` *is used within the program*

```
$DATA: << 1 M START $RI NEXT M ROW-> TRN 'DATA' STO>>
```

After transforming rows and columns with `TRN`*, the result is an* $N \times M$ *(for example* 16×8*) data matrix, like that defined in Eq. (4.20).*

The elements in each column are consistent with being the N *measurements on a single BPM. The random starting coordinates simulate random phase and amplitude starting conditions at the same BPM. The columns are saved as a matrix* `DATA`*. Using code like this, produce such a matrix and subject it to SVD analysis. Again a programmable calculator is slow but adequate. You should find two large elements on the diagonal of matrix* `S` *and the rest very small. Interpret this result. Confirm that* $\mathbf{U}$ *and* $\mathbf{V}$ *are orthogonal.*

CHAPTER 5

Analysis of Coupled Motion

5.1. Analysis of a 4×4 Symplectic Matrix

General 3D linearized propagation around a ring is represented by a 6×6 transfer matrix. Much simpler, and often approximately valid, is the representation of one dimensional motion by a 2×2 transfer matrix. In this section an intermediate approach is taken in which the coupling between two planes, usually x and y, but sometimes x and s, cannot be neglected. This calls for a 4×4 matrix treatment. In the last section a 6×6 formula is also given.

No matter how messed up an accelerator lattice is, as long as it elements are constant in time, there is a small amplitude domain in which transverse motion is accurately represented by a transfer matrix, which we are taking to be a 4×4 matrix $\mathbf{M}$. In a computer simulation all 16 elements of this matrix would be known but, operationally, in the control room, none of the elements are known *a priori.* The task of "model independent analysis" (MIA) is to use measurements from available instruments (in our case only beam position monitors (BPM) are assumed to be available) to infer valid accelerator properties even though $\mathbf{M}$ is completely unknown. Even though unknown, $\mathbf{M}$ is guaranteed to be a symplectic matrix.

This chapter describes several MIA methods, using BPM data and FFT analysis, that can be used to infer tunes and closed orbit deviations. The first step is to exploit symplecticity to derive a difference equation satisfied by $\mathbf{M}$.

The transverse particle position is specified by a 4-element column vector $\mathbf{x} = (x, x', y, y')^T$. This vector represents small transverse deviations from the equilibrium orbit. Note that the equilibrium orbit is *not* the same thing as the design orbit. About the only feature guaranteed to be true about the equilibrium orbit is that it stays in the vacuum chamber and repeats exactly turn after turn. Any bunch, no matter how poorly injected, by decoherence and filamentation, eventually centers itself on an equilibrium orbit. The vector $\mathbf{x}$, by definition, measures deviations from that orbit. Linearized evolution of $\mathbf{x}$ from longitudinal coordinate s_0 to s is described by the matrix equation

$$\mathbf{x}(s) = \mathbf{M}(s, s_0)\mathbf{x}(s_0). \tag{5.1}$$

The fact that $\mathbf{M}$ is symplectic, critical to the derivation of the difference equation being sought, can be expressed using the matrix

$$\mathbf{S} = \begin{pmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{5.2}$$

For $\mathbf{M}$ to be symplectic, its inverse must be equal to its "symplectic conjugate" $\overline{\mathbf{M}}$,

$$\mathbf{M}^{-1} = \overline{\mathbf{M}} = -\mathbf{S}\mathbf{M}^T\mathbf{S}. \tag{5.3}$$

Partitioning the 4×4 matrix $\mathbf{M}$ into 2×2 elements, it and its symplectic conjugate are

$$\mathbf{M} = \begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{pmatrix}, \quad \overline{\mathbf{M}} = \begin{pmatrix} \overline{\mathbf{A}} & \overline{\mathbf{C}} \\ \overline{\mathbf{B}} & \overline{\mathbf{D}} \end{pmatrix}. \tag{5.4}$$

A 2×2 matrix $\mathbf{A}$ and its symplectic conjugate are related by

$$\overline{\mathbf{A}} = \overline{\begin{pmatrix} a & b \\ c & d \end{pmatrix}} = \begin{pmatrix} d & -b \\ -c & a \end{pmatrix} = \mathbf{A}^{-1} \det|\mathbf{A}|, \tag{5.5}$$

provided the determinant $\det|\mathbf{A}|$ is non-vanishing.

Especially important for analysing the state of (x, y) coupling is a particular off-diagonal combination from Eq. (5.4), $\mathbf{E} = \mathbf{C} + \overline{\mathbf{B}}$ and its determinant $\mathcal{E} = \det|\mathbf{E}|$.

$$\mathbf{E} \equiv \mathbf{C} + \overline{\mathbf{B}} \equiv \begin{pmatrix} e & f \\ g & h \end{pmatrix} = \begin{pmatrix} c_{11} + b_{22} & c_{12} - b_{12} \\ c_{21} - b_{21} & c_{22} + b_{11} \end{pmatrix}, \quad \det|\mathbf{E}| = eh - fg \equiv \mathcal{E}. \tag{5.6}$$

For a stable lattice, eigenvalues λ_A and λ_D, of $\mathbf{M}$ (with their complex conjugate twins) satisfy the relations

$$\begin{aligned} \Lambda_A &\equiv \lambda_A + 1/\lambda_A = \exp(i\mu_A) + \exp(-i\mu_A) = 2\cos\mu_A \\ \Lambda_D &\equiv \lambda_D + 1/\lambda_D = \exp(i\mu_D) + \exp(-i\mu_D) = 2\cos\mu_D, \end{aligned} \tag{5.7}$$

where $\mu_A \equiv 2\pi\nu_A$ and $\mu_D \equiv 2\pi\nu_D$ are real angles. The quantities Λ_A and Λ_D, eigenvalues of $\mathbf{M} + \overline{\mathbf{M}}$, satisfy

$$\begin{aligned} \Lambda_A + \Lambda_D &= \mathrm{tr}\mathbf{A} + \mathrm{tr}\mathbf{D}, \\ \Lambda_A\Lambda_D &= \mathrm{tr}\mathbf{A}\,\mathrm{tr}\mathbf{D} - \mathcal{E}. \end{aligned} \tag{5.8}$$

For motion at small amplitude the linearized transfer matrix description gives a thoroughly satisfactory description of the motion. In the presence of coupling the tunes ν_A and ν_D are only approximately equal to the ideal, (or nominal, or design) tunes ν_x and ν_y. But ν_A and ν_D are readily measurable, no matter how badly coupled the lattice is. For this reason, they can be regarded as known, or at least operationally measurable, quantities. In fact the most common application of Fourier analysis of particle motion (as measured with beam position monitors) is for the operational measurement of these tunes. The formulation of this section can be used to exploit this procedure.

As partially seen already, the combination

$$\mathbf{M} + \overline{\mathbf{M}} = \mathbf{M} + \mathbf{M}^{-1} = \begin{pmatrix} \mathrm{tr}\mathbf{A} & 0 \\ 0 & \mathrm{tr}\mathbf{D} \end{pmatrix} + \begin{pmatrix} 0 & \overline{\mathbf{E}} \\ \mathbf{E} & 0 \end{pmatrix} \tag{5.9}$$

has simpler properties than $\mathbf{M}$. Using the fact that $\mathbf{M}^{-1}$ can be used to propagate backwards in time, this relation can be used to obtain four third-order, coupled difference equations that relate the coordinates on three successive turns (labeled $-$, 0, $+$):

$$\begin{aligned} x_+ - \mathrm{tr}\mathbf{A}\,x_0 + x_- &= hy_0 - fy'_0 \\ x'_+ - \mathrm{tr}\mathbf{A}\,x'_0 + x'_- &= -gy_0 + ey'_0 \\ y_+ - \mathrm{tr}\mathbf{D}\,y_0 + y_- &= ex_0 + fx'_0 \\ y'_+ - \mathrm{tr}\mathbf{D}\,y'_0 + y_- &= gx_0 + hx'_0. \end{aligned} \tag{5.10}$$

It is possible to uncouple these equations. Start by squaring Eq. (5.9), subtracting 2I, and using Eqs. (5.8);

$$\mathbf{M}^2+\mathbf{M}^{-2} = \begin{pmatrix} \mathrm{tr}^2\mathbf{A} + (\mathcal{E}-2)\mathbf{I} & 0 \\ 0 & \mathrm{tr}^2\mathbf{D} + (\mathcal{E}-2)\mathbf{I} \end{pmatrix} + (\mathrm{tr}\mathbf{A}+\mathrm{tr}\mathbf{D}) \begin{pmatrix} 0 & \overline{\mathbf{E}} \\ \mathbf{E} & 0 \end{pmatrix}. \quad (5.11)$$

From Eqs. (5.9) and (5.11), form the combination that eliminates the off-diagonal blocks,

$$\mathbf{M}^2 + \mathbf{M}^{-2} - (\Lambda_A + \Lambda_D)(\mathbf{M} + \mathbf{M}^{-1}) + (2 + \Lambda_A\Lambda_D)\mathbf{I} = 0. \quad (5.12)$$

Using this equation to obtain a difference equation for the phase space coordinates on successive turns yields

$$\mathbf{x}_{++} + \mathbf{x}_{--} - (\Lambda_A + \Lambda_D)(\mathbf{x}_+ + \mathbf{x}_-) + (2 + \Lambda_A\Lambda_D)\mathbf{x}_0 = 0. \quad (5.13)$$

This is the equation we have been seeking. Before applying it to practical problems such as closed-orbit finding and feedback control, we note the simpler equations that hold in case there is no cross-plane coupling. In that case, Eqs. (5.7) and (5.8) reduce to

$$\Lambda_A = \mathrm{tr}\mathbf{A} = 2\cos\mu_x, \qquad \Lambda_D = \mathrm{tr}\mathbf{D} = 2\cos\mu_y; \quad (5.14)$$

the right hand sides of Eqs. (5.10) vanish; and the first equation, for example, becomes

$$x_+ - 2\cos\mu_x\, x_0 + x_- = 0. \quad (5.15)$$

It is left as an exercise to show that this equation and the corresponding y-equation are consistent with Eq. (5.13) when there is no coupling.

Problem 5.1. *In the case that there is no coupling between x and y motion, show that Eq. (5.13) reduces to Eq. (5.15) and the corresponding y-equation.*

Problem 5.2. *Algorithm for converting an almost symplectic matrix into a symplectic matrix. Define*

$$\mathbf{S} = \begin{pmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \quad (5.16)$$

The symplectic conjugate of a matrix $\mathbf{M}$ *is defined by*

$$\overline{\mathbf{M}} = -\mathbf{S}\mathbf{M}^T\mathbf{S}, \quad (5.17)$$

where $\mathbf{M}^T$ *is the transpose of* $\mathbf{M}$. *One can write a calculator routine to perform this operation*

```
$BAR: << TRN S * S SWAP * NEG >>
```

A matrix M_s is symplectic if and only if

$$\overline{\mathbf{M}}_s = \mathbf{M}_s^{-1}. \quad (5.18)$$

Suppose that $\mathbf{M}$ *is "almost" symplectic. Define a new matrix, close to* $\mathbf{M}$ *by*

$$\mathbf{M}_s = \mathbf{M} + \frac{\mathbf{M} - \mathbf{M}\overline{\mathbf{M}}\mathbf{M}}{2}. \quad (5.19)$$

Neglecting terms quadratic in $\mathbf{M} - \mathbf{M}_s$, *show that M_s is approximately symplectic.*

Using 4×4 *unit matrix* `I44`, *One can write a calculator routine,*

```
$SYM: << DUP DUP $BAR * 2 / NEG I44 1.5 * + SWAP * >>
```

that "symplectifies" a transfer matrix.

In a simulation program an artificial manipulation like this is risky. Just because a matrix is symplectic does not mean it is correct. *But sometimes one is quite confident that a matrix is essentially correct, and wishes to use it for iteration, say millions of times. Even the tiniest of failures of symplecticity will cause this operation to give artificial emittance growth over such long times. In this case artificial symplectification may be justified.*

PROBLEM 5.3. *The matrix* **M**

```
0.231876 -0.029239  -0.031317  0.000000
32.36138  0.231876  -0.822722  0.000000
0.000000  0.000000   1.000000  0.000000
0.822722  0.031317  -0.012908  1.000000
```

appeared in an earlier problem. This matrix is very nearly symplectic but, to make it look less gross in print, its elements are given to only 5 or 6 decimal points. It can therefore not be exactly symplectic. Perform the "symplectification" defined in the previous problem one or more times so that the matrix is symplectic to machine precision, typically 12 places or so.

PROBLEM 5.4. *With the symplectified matrix* **M** *obtained in the previous problem, confirm that Eq (5.12) is satisfied. It is necessary to first find* Λ_A *and* Λ_D *by solving a characteristic equation.*

5.2. Finding the Tunes and Closed Orbit, Uncoupled Case

Koutchouk[**11**] has described a closed-orbit finding procedure, based on Eq. (5.15), which he ascribes to Verdier and Risselada.[**12**] That method, which assumes purely uncoupled motion, will now be described and then generalized. Much the same description applies whether one is discussing operational procedures applied in the control room of an actual accelerator or simulation in a computer. In either case finding the closed orbit is usually performed by starting with a guess and iteratively improving it.

In the derivations of the preceeding section it was implicitly assumed that transverse coordinates were measured relative to an unknown closed orbit. Let us assume that the BPM's are positioned perfectly on the design orbit. This is rarely precisely true, but to get our feet on the ground we have to start somewhere. Then, in principle, the BPM is capable of measuring the closed orbit deviation at that point; call it $\mathbf{x}_{\rm co}$. Making the replacement $\mathbf{x} \to \mathbf{x} - \mathbf{x}_{\rm co}$ in Eq. (5.13), after simplification, yields

$$\mathbf{x}_{++} + \mathbf{x}_{--} - (\Lambda_A + \Lambda_D)(\mathbf{x}_{+} + \mathbf{x}_{-}) + (2 + \Lambda_A\Lambda_D)\mathbf{x}_0 = (2 - \Lambda_A)(2 - \Lambda_D)\mathbf{x}_{\rm co}. \tag{5.20}$$

Making the same replacement $x \to x - x_{\rm co}$ in Eq. (5.15) yields

$$x_{+} - 2\cos\mu_x\, x_0 + x_{-} = 2(1 - \cos\mu_x)x_{\rm co}. \tag{5.21}$$

The parameters Λ_A and Λ_D in Eq. (5.20) are simple functions of operationally measurable tunes, as is $\cos\mu_x$ in Eq. (5.21). In the control room of an actual accelerator, if circulating beam can be obtained, Λ_A and Λ_D can be measured by spectral analysis of beam position monitor signals. Similarly, in a computer simulation, if multiple turns survive, the tunes can be obtained by FFT analysis. Unfortunately the "if's" in the two previous sentences are sometimes not satisfied.

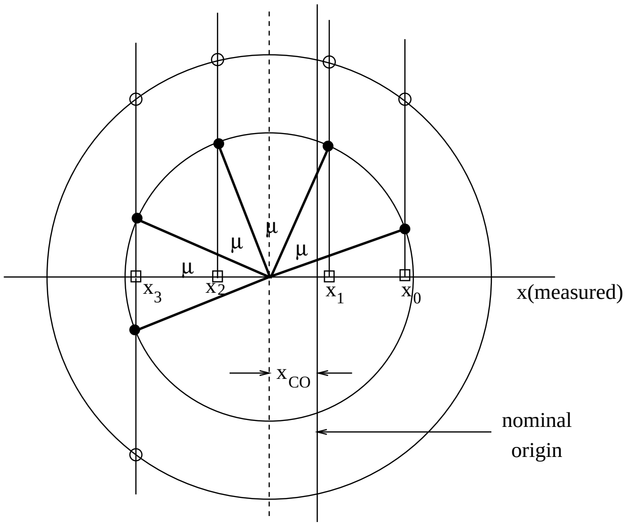


FIGURE 5.1. Geometric construction indicating how tune and closed orbit can be found from measuring the transverse displacement for several successive turns. Open square symbols are measured. Solid circle symbols lie on and define the correct phase space circle. Open circles lie on a candidate, but wrong, phase space circle.

For this reason it is desirable to have a robust procedure for extracting tunes that makes minimal operational demands. Following Verdier and Risselada we obtain another equation like Eq. (5.21) by incrementing the indices by one, (which leaves the right hand side unchanged), and then eliminate $x_{\rm co}$ from the two equations,

$$x_3 - 2\cos\mu_x\, x_2 + x_1 = x_2 - 2\cos\mu_x\, x_1 + x_0. \tag{5.22}$$

Solving for μ_x yields

$$\cos\mu_x = \frac{x_3 - x_2 + x_1 - x_0}{2(x_2 - x_1)}. \tag{5.23}$$

From this equation, starting with x_0, if the particle (or beam) can survive three full turns, and the displacement measured on each passage through the origin, the tune can be obtained. Once the tune is known, the closed orbit is obtained from Eq. (5.21),

$$x_{\rm co} = \frac{x_2 - 2\cos\mu_x\, x_1 + x_0}{2(1 - \cos\mu_x)}. \tag{5.24}$$

This prescription can be foiled by measurement errors, by the presence of coupling, or by the presence in the lattice of nonlinear elements that violate the conditions used in deriving the difference equation. There is nothing we can do about measurement errors except complain about the instrumentation. Before proceeding to discuss what can be done about coupling we consider nonlinearity.

Because of nonlinearity, Eqs. (5.23) and (5.24) will be not quite satisfied and the closed orbit not quite found. This performance is typical of almost all operational accelerator procedures. The universal attempted fix is to proceed by iteration. In this case, having found a tentative value for $x_{\rm co}$ we launch another particle from that point. Assuming a sensible prescription for picking the initial slope is available, the ability of nonlinearity to foil this approach will rapidly decrease with each succeeding iteration, as the orbit will stay in progressively reduced, and hence more linear, regions. It can certainly happen however, that the first iteration fails due to nonlinearity. Either the particle is lost completely, (a possibility the derivation excluded) or the errors make the "improved" closed orbit worse than the tentative starting value. In either case an alternate approach must be found. In practice the alternate approach is usually trial-and-error or "knob-twiddling", which normally succeeds eventually. From that point rapid convergence employing Eqs. (5.23) and (5.24) is typical.

A procedure for improving the starting slope would improve the convergence. The earlier derivation showed that the slope variables satisfy the same difference equations as the displacements. As a result we obtain

$$x'_{\rm co} = \frac{x'_2 - 2\cos\mu_x\, x'_1 + x'_0}{2(1-\cos\mu_x)}. \tag{5.25}$$

Assuming slope values are available (which is certainly true in a computer simulation, but would only be true by using and adjacent BPM in the laboratory) this equation can be used to improve the tentative closed orbit initial conditions.

5.3. Example of MIA in a Coupled Lattice

We now wish to generalize this prescription in order to make its convergence robust even in the presence of coupling. In practice, the presence of coupling seriously compromises the effectiveness of closed orbit determinations. Because coupling is a "linear effect", its fractional importance does not reduce with succeeding stages of iteration. For too great coupling the iterative approach described above simply does not converge. For this reason we contemplate using the more general Eq. (5.20) to obtain simultaneous convergence in both planes. As in the uncoupled case, there are two stages, the first to find the tune(s), the second to find the closed orbit.

Several alternative approaches to finding the tunes suggest themselves. The first two are applicable only if the coupling is weak (which is often the case since its presence is unintentional) and hence its effect on the tunes is likely to be negligible (because tune shifts depend quadratically on skew quadrupole strengths.) In this case the "design" tunes could be used in a computer simulation, but this is not really practical operationally. Instead the few turn determination of μ_x using Eq. (5.23) and a corresponding determination of μ_y might be adequate.

A more robust approach is to obtain equations for Λ_A and Λ_D in a manner analogous to the derivation of Eq. (5.23). We write the vector equation

$$\begin{aligned}\mathbf{x}_4 + \mathbf{x}_0 - (\Lambda_A + \Lambda_D)(\mathbf{x}_3 + \mathbf{x}_1) + (2 + \Lambda_A\Lambda_D)\mathbf{x}_2 \\ = \mathbf{x}_5 + \mathbf{x}_1 - (\Lambda_A + \Lambda_D)(\mathbf{x}_4 + \mathbf{x}_2) + (2 + \Lambda_A\Lambda_D)\mathbf{x}_3.\end{aligned} \tag{5.26}$$

Collecting terms yields

$$\begin{pmatrix} x_4 - x_3 + x_2 - x_1 & -x_3 + x_2 \\ y_4 - y_3 + y_2 - y_1 & -y_3 + y_2 \end{pmatrix} \begin{pmatrix} \Lambda_A + \Lambda_D \\ \Lambda_A\Lambda_D \end{pmatrix} = \begin{pmatrix} x_5 - x_4 + 2x_3 - 2x_2 + x_1 - x_0 \\ y_5 - y_4 + 2y_3 - 2y_2 + y_1 - y_0 \end{pmatrix}. \tag{5.27}$$

These equations can be solved for Λ_A and Λ_D if data from five consecutive full turns is available.

By far the most accurate determination of Λ_A and Λ_D is applicable when multiple turns can be obtained. Define the expectation value $< f >$ of N samples

f_i by $\sum_i^N f_i/N$. Multiplying the x and y components of (5.13) by x_0 and y_0 respectively, taking expectation values, and rearranging to express as equations for Λ_A and Λ_D yields

$$\begin{pmatrix} <(x_+ + x_-)x_0> & -<x_0^2> \\ <(y_+ + y_-)y_0> & -<y_0^2> \end{pmatrix} \begin{pmatrix} \Lambda_A + \Lambda_D \\ \Lambda_A\,\Lambda_D \end{pmatrix} = \begin{pmatrix} <(x_{++} + x_{--})x_0> +2<x_0^2> \\ <(y_{++} + y_{--})y_0> +2<y_0^2> \end{pmatrix}. \tag{5.28}$$

When this equation was applied for 512 turn data at LEP and solved for Λ_A and Λ_D, accuracies of approximately ± 0.003 were obtained for the tunes ν_x and ν_y.[**13**] Once the tunes are known, the coefficients in Eq. (5.20) can be evaluated, and improved values for all four closed orbit coordinates can be obtained from the equation

$$\mathbf{x}_{\rm co} = \mathbf{x}_{++} + \mathbf{x}_{--} - (\Lambda_A + \Lambda_D)(\mathbf{x}_+ + \mathbf{x}_-) + (2 + \Lambda_A\Lambda_D)\frac{\mathbf{x}_0}{(2-\Lambda_A)(2-\Lambda_D)}. \tag{5.29}$$

In an accelerator control room the quantities Λ_A and Λ_D are usually available to high accuracy from one of the ubiquitous spectrum analyser displays. Since these are *global* quantities, they can be measured anywhere in the lattice. In this case Eq. (5.29) can be applied to find the closed orbit position at every dual-plane BPM with no further ado.

If the BPM's are not dual-plane, they are usually arranged alternately around the lattice, with vertical measured at vertically-focusing quads, horizontal at horizontal quads. In this case, to use Eq. (5.29), say at the location of a horizontally focusing quad, it is necessary to “interpolate” a vertical measurement from the adjacent vertical BPM's. This operation cannot be said to be “model-independent” since it relies on the lattice design in the region of the three quads. But, barring serious field imperfections over such a small region, the interpolation can be said to be “somewhat model-insensitive”.

5.4. Eigenanalysis of 3D Maps

Possibly the most difficult step in the MIA analysis of 2D coupled motion was the derivation of explicit formulas for the eigenvalues of a symplectic 4 × 4 equation. (The quadratic equation formed from Eqs. (5.8).) It is possible to perform the analogous calculation even in the most general case. For fully-general 3D motion the linear transfer matrix $\mathbf{M}$, and its symplectic conjugate $\overline{\mathbf{M}}$ can be written in partitioned form as

$$\mathbf{M} = \begin{pmatrix} A & B & E \\ C & D & F \\ G & H & J \end{pmatrix}, \quad \overline{\mathbf{M}} = \begin{pmatrix} \overline{A} & \overline{C} & \overline{G} \\ \overline{B} & \overline{D} & \overline{H} \\ \overline{E} & \overline{F} & \overline{J} \end{pmatrix}. \tag{5.30}$$

Let us assume that interplane coupling is sufficiently weak that the matrices A, D, and J, are “not too far from” the uncoupled 2×2 “design” transfer matrices corresponding to pure x, y, and z motion respectively. However, the purpose of this assumption is not to justify a perturbative expansion, since the formulas will be exact. Rather it is to resolve ambiguities in identifying the roots of the equations by considerations of continuity. Because $\mathbf{M}$ is necessarily symplectic, its symplectic conjugate, defined using block-diagonal matrix S, each of whose diagonal blocks is $\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$,

$$\overline{\mathbf{M}} = -S\mathbf{M}^T S, \tag{5.31}$$

is also its inverse

$$\overline{\mathbf{M}} = \mathbf{M}^{-1}. \tag{5.32}$$

We define an auxiliary matrix,

$$\Xi = \mathbf{M} + \overline{\mathbf{M}} = \mathbf{M} + \mathbf{M}^{-1}, \tag{5.33}$$

having much simpler properties than X. In particular, if (as it does) $\mathbf{M}$ has eigenvalue $\lambda = e^{i\mu}$ with μ real, then Ξ has real eigenvalue $\Lambda = \lambda + \lambda^{-1} = 2\cos\mu$. This implies that Ξ has three, real, double eigenvalues, Λ_x, Λ_y, and Λ_z, for a stable lattice.

Explicitly Ξ is given by

$$\Xi = \begin{pmatrix} \mathrm{tr}A\; I & T & U \\ \overline{T} & \mathrm{tr}D\; I & V \\ \overline{U} & \overline{V} & \mathrm{tr}J\; I \end{pmatrix}, \tag{5.34}$$

where

$$B + \overline{C} = T = \begin{pmatrix} h & -f \\ -g & e \end{pmatrix}, \qquad \overline{T} = \begin{pmatrix} e & f \\ g & h \end{pmatrix}, \tag{5.35}$$

$$E + \overline{G} = U = \begin{pmatrix} n & -l \\ -m & k \end{pmatrix}, \qquad \overline{U} = \begin{pmatrix} k & l \\ m & n \end{pmatrix}, \tag{5.36}$$

$$F + \overline{H} = V = \begin{pmatrix} s & -q \\ -r & p \end{pmatrix}, \qquad \overline{V} = \begin{pmatrix} p & q \\ r & s \end{pmatrix}. \tag{5.37}$$

Note especially that the 2×2 diagonal blocks of Ξ are proportional to identity matrix I. For simplifying formulas which follow, two relations, valid for 2×2 matrices, are useful:

$$A\bar{A} = \det A \equiv |A|, \quad A + \bar{A} = \mathrm{tr}A\, I. \tag{5.38}$$

The characteristic equation is

$$\Delta(\Lambda) = \det \begin{vmatrix} (\mathrm{tr}A - \Lambda)I & T & U \\ \bar{T} & (\mathrm{tr}D - \Lambda)I & V \\ \bar{U} & \bar{V} & (\mathrm{tr}J - \Lambda)I \end{vmatrix} = 0. \tag{5.39}$$

This determinant can be worked out by following Gantmacher.[**14**] To simplify the algebra it is useful to introduce a symbols

$$\alpha = (\mathrm{tr}A - \Lambda)I. \tag{5.40}$$

Though this is a 2×2 matrix it commutes with everything and can be treated just like a scalar factor. We obtain

$$\Delta(\Lambda) = \Lambda^3 - p_1\Lambda^2 - p_2\Lambda - p_3 \tag{5.41}$$

where

$$\begin{aligned} p_1 &= \mathrm{tr}A + \mathrm{tr}D + \mathrm{tr}J = \Lambda_A + \Lambda_D + \Lambda_J && (5.42) \\ p_2 &= -\,\mathrm{tr}A\mathrm{tr}D - \mathrm{tr}A\mathrm{tr}J - \mathrm{tr}D\mathrm{tr}J - |U| - |T| - |V| \\ &= -\,(\Lambda_A\Lambda_D + \Lambda_A\Lambda_J + \Lambda_D\Lambda_J) && (5.43) \\ p_3 &= -\,\mathrm{tr}D|U| - \mathrm{tr}J|T| - \mathrm{tr}A|V| + \mathrm{tr}(\bar{V}\bar{T}U) = \Lambda_A\Lambda_D\Lambda_J. && (5.44) \end{aligned}$$

The expression for p_3 has a suspicious-looking lack of symmetry, but it is invariant to reordering of the (x, y, z) coordinates; so also is its last term. For a stable lattice the three roots of Eq. (5.41) are all real, and an explicit formula can be written for them. Following Press et al.[**15**], and defining

$$Q = p_1^2 + \frac{3p_2}{9} \tag{5.45}$$

$$R = -2p_1^3 - 9p_1p_2 - \frac{27p_3}{54} \tag{5.46}$$

$$\theta = \arccos(R/\sqrt{Q^3}), \tag{5.47}$$

the roots are given by

$$\Lambda_1 = -2\sqrt{Q}\cos(\frac{\theta}{3}) + p_1/3 \tag{5.48}$$

$$\Lambda_2 = -2\sqrt{Q}\cos(\frac{\theta+2\pi}{3}) + p_1/3 \tag{5.49}$$

$$\Lambda_3 = -2\sqrt{Q}\cos(\frac{\theta+4\pi}{3}) + p_1/3. \tag{5.50}$$

The eigenvalue triplet $(\Lambda_1, \Lambda_2, \Lambda_3)$ is some permutation of the triplets $(\Lambda_x, \Lambda_y, \Lambda_z)$. These can also be labeled $(\Lambda_A, \Lambda_D, \Lambda_J)$ assuming the perturbations away from design, uncoupled optics, leaves the tunes close to their design values. We will assume relabeling has been performed so that $(1, x, A)$ go together, as do $(2, y, D)$ and $(3, z, J)$.

CHAPTER 6

Longitudinal Dynamics

6.1. Synchrotron Oscillations

6.1.1. Equations of Motion. When passing through an RF cavity at phase angle $\phi_0(t)$ the reference particle acquires energy ΔE_{ref} given by

$$\Delta E_{\mathrm{ref}} = Q\hat{V} \sin\phi_0(t) - u_{\mathrm{ref}}(t), \tag{6.1}$$

with Q and $\hat{V}$ both assumed positive. Here $Q\hat{V}$ is the maximum possible energy gain in the RF cavity. For electrons or protons $Q = |e|$, for ions $Q = Z|e|$. Possible energy loss represented by $u_{\mathrm{ref}}(t)$ is due, for example, to synchrotron radiation or beam wall interaction. This loss is distributed more or less continuously around the ring, but we assume that it can be adequately represented by a single loss occurring at the RF cavity. It is obviously important for correct relativistic formulas be used to calculate energy E gains and the corresponding changes in total momentum p_0. We will return to these detailed calculations after describing the essential features of synchrotron oscillations.

Formula (6.1) may represent storage ring operation for which $< \Delta E >= 0$, in which case, neglecting possible small shift due to nonlinearity, ϕ_0 adjusts itself such that

$$Q\hat{V} \sin\phi_0 = u_{\mathrm{ref}}. \tag{6.2}$$

Formula (6.1) also applies to acceleration, in which case the maximum possible energy gain per turn is given by

$$\Delta E_{\mathrm{ref}} < \Delta E_{\max} = Q\hat{V} - u_{\mathrm{ref}}(t). \tag{6.3}$$

Since the stable bucket area vanishes in this limit, the acceleration rate has to be made substantially smaller. To support an acceleration interval followed by a storage interval, or any other acceleration scenario, the time dependence of $\phi_0(t)$ has to be programmed accordingly. For truly adiabatic acceleration the acceleration has to be *much* less than the limit given by Eq. (6.3). Much of the following analysis assumes that a steady energy beam is being described, but the formulas will apply also even during acceleration, at least in the truly adiabatic case.

Since we are primarily interested in motion of a general beam particle *relative to* the reference particle we wish, to the extent possible, to suppress the variation of $E_{\mathrm{ref}}(t)$ from the formulas, or rather to "hide" the dependence by making it implicit. The energy gain ΔE of a general particle, with arrival "time" ct, relative to the reference particle, is

$$\Delta E = Q\hat{V} \sin\Big(\frac{\omega_{\mathrm{rf}}}{c} ct + \phi_0\Big) - Q\hat{V} \sin\phi_0 - \Delta u, \tag{6.4}$$

where Δu represents energy loss over and above that suffered by the reference particle. In electron accelerators the Δu term, because of its dependendence on position in phase space, influences bunch distributions. In this chapter Δu will be set to zero.

It is necessary for the RF frequency $\omega_{\rm rf}$ to be synchronized to the revolution frequency $\omega_{\rm rev}$, but the former can differ by a (typically large) integer factor known as the "harmonic number" h, defined by

$$h = \frac{\omega_{\rm rf}}{\omega_{\rm rev}} \tag{6.5}$$

With ϕ_0 near zero, the sign of the first term of Eq. (6.4) has been chosen such that a particle with $ct > 0$ (which arrives *late*) has its energy *increased* by the cavity. Below transition, where revolution time is affected more by velocity than by circumference, this is the sign of energy increment needed for stability.

The particle's revolution period $T_{\rm rev} \equiv T(0)$ is related to the revolution frequency by $T_{\rm rev} = 2\pi/\omega_{\rm rev}$. The single most important lattice parameter influencing longitudinal motion is $\eta_{\rm rf}$, the so-called "slip factor"[1] [2] defined by

$$\eta_{\rm rf}\tilde{\delta} \equiv \eta_{\rm rf}\frac{\delta}{\beta_0} = \frac{\Delta t_{\rm circ}(\tilde{\delta}) + \Delta t_{\rm vel}(\tilde{\delta})}{T_{\rm rev}}. \tag{6.6}$$

The tildes present in this equation require explanation. The need for introducing $\tilde{\delta} = \delta/\beta_0$ to describe less-than-fully-relativistic motion was explained in section 2.6.1. Eq (6.6), following tradition, defines the slip factor in terms of fractional momentum, for which our symbol is $\tilde{\delta}$.

The energy gain given by Eq. (6.4) is negative below transition, where the second term, which is negative for positive δ, dominates the first. The (linearized) change $\Delta(ct)$ in arrival time of a particle at the RF cavity is governed by the slip factor;

$$\Delta(ct) = cT_{\rm rev}(\eta_{\rm rf}/\beta_0)\delta. \tag{6.7}$$

Eqs. (6.4) and (6.7) are equations of motion relating the dependent variables by giving the change in $\Delta\delta = \Delta E/(p_0c)$ on a given turn due to ct and *vice versa.*

To analyse longitudinal motion it is practical to use either a continuous independent variable t_a (with subscript a used to differentiate absolute, wall clock, time from relative-to-reference-particle arrival time t) or to use turn index i. Though the latter choice is usually adopted for analysing transverse motion, the use of t_a is common for studying longitudinal motion. Because the longitudinal oscillation period is always long compared to the revolution period, the longitudinal phase advance per turn is small compared to 2π and, except for a scale factor, turn index i is a kind of (not very fine grain) "discretized" version of t_a. Quantitatively the transition to continuous time is based on the relation between continuous and discrete rates, which is

$$\frac{d(ct)}{dt_a} \approx \frac{\Delta(ct)}{T_{\rm rev}}, \tag{6.8}$$

[1] As explained in the text, the phase space variable $\delta = \Delta E/(p_0c)$, as defined in Eq.(2.26), is now being used. Until now it has been suggested that the distinction between δ and $\tilde{\delta}$ was largely cosmetic. But the very phenomenon of *transition crossing* reflects not quite fully relativistic motion, where the distinction *is* important.

[2] In the electron world the dispersion function $D(s)$ is often denoted by $\eta(s)$ which must not be confused with $\eta_{\rm rf}$. In the "toy" lattice files, this accounts for the use of `eta` as the variable name for dispersion. The purpose for the subscript "rf" on $\eta_{\rm rf}$ is to avoid the same ambiguity.

where $\Delta(ct)$ is a ct deviation ocurring in one turn. The so-called "smooth approximation" assumes that the fractional variation of this ratio during one term is negligible.

6.1.2. Small Amplitude Motion. As usual with oscillations, it is easiest to start with small amplitudes. For arrival time ct the (linearized at $ct = 0$) change $\Delta\delta$ is

$$\Delta\delta = \frac{\Delta E}{p_0 c} = \frac{Q\hat{V}\omega_{\rm rf}\cos\phi_0}{p_0 c^2}\, ct, \tag{6.9}$$

For stability this impulse needs to be "restoring". However, what constitutes restoration switches sign at transition, because $\eta_{\rm rf}$ changes sign there. There are two choices for the angle ϕ_0 that lead to the same acceleration per turn of the reference particle; their typical separation is somewhat less than π. As the beam accelerates through transition it is necessary to switch from one of these choices to the other to preserve stability. To do this the RF phase ϕ_0 is altered discontinuously.

Neglecting the Δu term in Eq. (6.4), and assuming small amplitudes, the change in δ at the RF can be expressed by a difference formula, using i as turn index;

$$\delta_{i+} - \delta_{i-} = \frac{Q\hat{V}}{p_0 c}\left(\sin\left(\frac{\omega_{\rm rf}}{c}ct + \phi_0\right) - \sin\phi_0\right) \approx \frac{Q\hat{V}\omega_{\rm rf}\cos\phi_0}{p_0 c^2}\, ct_i. \tag{6.10}$$

Since the energy is constant everywhere except at the cavity, the end points of this difference can be taken anywhere in the post and pre-passage turns. The longitudinal evolution for two consecutive turns is given by

$$ct_{i+1} - ct_i = cT_{\rm rev}(\eta_{\rm rf}/\beta_0)\delta_{i+}, \tag{6.11}$$

$$ct_i - ct_{i-1} = cT_{\rm rev}(\eta_{\rm rf}/\beta_0)\delta_{i-}. \tag{6.12}$$

Subtracting these two equations, and applying Eq. (6.10) yields[3]

$$ct_{i+1} - 2ct_i + ct_{i-1} = (T_{\rm rev}\omega_{\rm rf})((\eta_{\rm rf}/\beta_0)\cos\phi_0)\,\frac{Q\hat{V}}{p_0 c}\, ct_i \tag{6.13}$$

Depending on the relative values of the parameters, the solution of such a second order difference condition can be oscillatory.

PROBLEM 6.1. *Using standard trigonometric identities show that either of the functions (sequences, if you prefer)*

$$x_j = A\,\frac{\sin(\mu_0\, j)}{\cos(\mu_0\, j)}, \quad j = 0, 1, 2, \ldots, \tag{6.14}$$

satisfies the second order difference equation

$$x_{j+1} - 2\cos\mu_0\, x_j + x_{j-1} = 0. \tag{6.15}$$

If the numerical value of the coefficient of the second term exceeds 2 in absolute value it is clear that the equation represents unstable motion as there is no real angle μ_0 consistent with the equation.

[3] There are two rationales behind maintaining factors grouped in the combination $((\eta_{\rm rf}/\beta_0)\cos\phi_0)$. If the phase is "jumped" appropriately, then this combination is continuous in passing through transition, and the grouping $(\eta_{\rm rf}/\beta_0)$ was justified earlier.

Using the result of the previous problem, in the oscillatory case, the μ_s can be obtained by inspection of the equation because the coefficient of the ct_i term is $-2\cos\mu_s$;

$$\cos\mu_s = 1 + (T_{\rm rev}\omega_{\rm rf})((\eta_{\rm rf}/\beta_0)\cos\phi_0)\frac{Q\hat{V}}{2p_0c} \tag{6.16}$$

In most practical cases μ_s is sufficiently small to allow the small angle approximation, so that

$$\mu_s^2 = -(T_{\rm rev}\omega_{\rm rf})((\eta_{\rm rf}/\beta_0)\cos\phi_0)\frac{Q\hat{V}}{p_0c} \tag{6.17}$$

For stable motion it is required that μ_s be real, which implies that the choice of angle ϕ_0 must be such that $\cos\phi_0$ and $(\eta_{\rm rf}/\beta_0)$ have opposite signs. To complete the analogy with transverse formalism, the synchrotron "tune" ν_s is introduced as

$$\nu_s = \frac{\mu_s}{2\pi}. \tag{6.18}$$

The description has been in terms of difference equations, rather than differential equations. With the RF concentrated at one point this constitutes a correct description. But, because μ_s is usually small, the angular steps in phase space each turn are small, and the usual approximation is to introduce a "smoothed" description in which the longitudinal variables execute simple harmonic motion. In this spirit, the "synchrotron (radian) frequency" Ω_s can be obtained from the synchrotron tune;[4]

$$\Omega_s = \frac{\mu_s}{T_{\rm rev}}. \tag{6.19}$$

We can employ transfer matrix notation for longitudinal motion but using longitudinal quantities (ct, δ) rather than (x, x') as phase space variables. In analysing motion in longitudinal phase space the following points should be noted:

- Unlike x', δ is *not* $d(ct)/ds$. The correct relation will be given shortly.
- Because of the externally imposed time dependent RF, it is natural to use "absolute" time t_a rather than s as independent variable.
- If the longitudinal focusing were uniform around the ring it would lead to pure simple harmonic oscillation but this is not the case. The RF cavity acts like a "thin lens" for longitudinal motion, retarding front runners and advancing tardy particles once per turn. In principle, with more than one cavity, a longitudinal "β-function" formalism would be required. But we assume that the focusing is weak in the sense that the effective "focal length" of the RF cavity is long compared to the ring radius. This is equivalent to assuming that the sychrotron oscillation phase advance per turn μ_s satisfies $\mu_s << 1$.

Longitudinal, once-around evolution can be described in Twiss-like form by[5]

$$\begin{pmatrix} ct \\ \delta \end{pmatrix}_{n+1} = \begin{pmatrix} \cos\mu_s & \tilde{\beta}_s\sin\mu_s \\ -\sin\mu_s/\tilde{\beta}_s & \cos\mu_s \end{pmatrix} \begin{pmatrix} ct \\ \delta \end{pmatrix}_n \tag{6.20}$$

[4] There is an unfortunate redundancy of symbols. The quantities μ_s, ν_s, and Ω_s, differ only by constant factors and are therefore essentially equivalent parameters. The main virtue of Ω_s is that $\Omega_s/(2\pi)$ is the frequency observed on a control room spectrum analyser that is sensing synchrotron oscillations.

[5] The tilde that appears over $\tilde{\beta}_s$ (which is a beta function, *not* a relativistic factor) in this section, is completely unrelated to the distinction between δ and $\tilde{\delta}$. *This* tilde represents the fact that $\tilde{\beta}_s$ is almost, but not quite, the analog of a transverse beta function.

The analogy between $\tilde{\beta}_s$ and an ordinary beta-function has to be established. Consistent with $\tilde{\beta}_s$ being treated as constant, the analog of the Twiss alpha function has been taken to be zero. To convert to circular phase space motion the scale along the energy axis can be changed;

$$\begin{pmatrix} ct \\ \tilde{\beta}_s\delta \end{pmatrix}_{n+1} = \begin{pmatrix} \cos\mu_s & \sin\mu_s \\ -\sin\mu_s & \cos\mu_s \end{pmatrix} \begin{pmatrix} ct \\ \tilde{\beta}_s\delta \end{pmatrix}_n . \tag{6.21}$$

The factor $\tilde{\beta}_s$ provides the phase space aspect ratio or, what is actually observable, the ratio of (r.m.s) bunch length σ_{ct} to the (r.m.s.) "fractional"[6] energy spread σ_δ;

$$\sigma_{ct} = \tilde{\beta}_s\,\sigma_\delta. \tag{6.22}$$

In an electron accelerator the energy spread is calculable from synchrotron radiation formulas, so this equation establishes the bunch length. In a proton accelerator the energy spread is inherited at injection, and evolves adiabatically, causing the aspect ratio in phase space to be governed by $\omega_{\rm rf}$, RF voltage $\hat{V}$, and beam energy E_0 (or equivalently γ_0). The usual way to change this aspect ratio appreciably is by "rebucketing". This maneuver requires the turning off of cavity excitation at one frequency and turning on a cavity at another frequency.

It is again convenient to change the scale of the second component so that the phase space motion is circular. For this purpose, as always in simple harmonic motion, the phase space coordinates need to be in the ratio $(ct, \frac{1}{\Omega_s}\frac{d(ct)}{dt_a})$. Combining this result with Eq. (6.22) and Eq. (6.8), components $(ct, \tilde{\beta}_s\delta)$ map out a circle in phase space if

$$\tilde{\beta}_s = \frac{\sigma_{ct}}{\sigma_\delta} = \frac{|(\eta_{\rm rf}/\beta_0)|}{\mu_{\rm s}}\,\mathcal{C}(0). \tag{6.23}$$

This is the result needed to obtain bunch length from energy spread. In particular, to lowest approximation, the bunch length goes to zero at transition, where $(\eta_{\rm rf}/\beta_0) = 0$. Passage through transition will be analysed shortly. A UAL simulation is shown in FIG 6.6.

For using beta-function terminology, the circular phase space coordinates need to be $(ct, \beta_s\frac{d(ct)}{ds})$. This requires

$$\beta_s = \frac{1}{\mu_s}\,\mathcal{C}(0). \tag{6.24}$$

This can be regarded as the "longitudinal beta-function". The phase advance per turn μ_s is rarely as great as 0.1. As a result the "longitudinal beta function" β_s normally greatly exceeds the ring radius R. This justifies some of the assumptions that have been made, such as neglecting α_s and treating β_s as constant. It can also be noted, since the transverse beta function satisfies $<\beta_x> \approx \mathcal{C}(0)/\mu_x$, that

$$\frac{\beta_s}{<\beta_x>} \approx \frac{\mu_x}{\mu_s}, \tag{6.25}$$

the ratio typically being in the range of hundreds or thousands.

6.1.3. Large Amplitude Motion. The analysis to this point has been linearized, but description of large amplitude motion is also important, especially in hadron accelerators for which the region of stability can be nearly filled. In general

[6] Recall that calling $\delta = \Delta E/p_0c$ "fractional energy spread" is only really valid for fully relativistic motion.

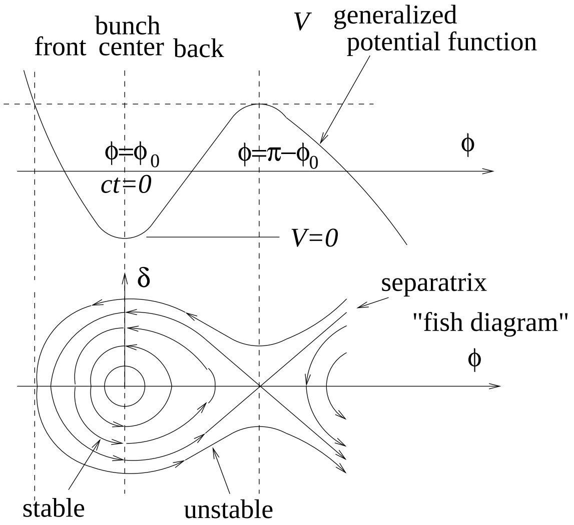


FIGURE 6.1. "Fish diagram" of longitudinal phase space motion. On passing through transition the pattern is right-left reflected. i.e. the fish points in the opposite direction.

this requires numerical treatment, but analytic formulas for features of the separatrix, such as the maximum excursions of the variables and the approximate "bucket area", can be obtained. The large amplitude motion is easiest to understand in connection with the "fish diagram" shown in FIG 6.1.

Because RF cavities cause essentially discontinuous motion it would be appropriate, and not difficult, to continue to use difference equations for this discussion. But most accelerator physicists are more comfortable using the differential equation that becomes a good approximation for $\mu_s << 1$. This is known as the "smooth approximation" because the acceleration is regarded as spread out uniformly around the ring. This approximation is excellent for essentially all hadron accelerators and most electron accelerators. In this approximation the result of the following problem can be used to derive the appropriate differential equation.

PROBLEM 6.2. *If the quantity z_i, $i = 0, 1, 2, ...$ varies slowly enough (i.e. $|z_{i+1} - z_i| << |z_i|$) the index i can be considered to be continuous rather than discrete. Then the continuous time variable t is given by $t = T_s i$, where T_s is the revolution period. Derivatives $dz(t)/dt$ and $d^2(t)/dt^2$ can then be approximated by difference formulas based on successive values, z_{i-1}, z_i, and z_{i+1}. Derive these formulas.*

For large amplitude motion Eq. (6.13) can be repeated, but without linearization, by substituting directly from Eq. (6.4);

$$\frac{ct_{i+1} - 2ct_i + ct_{i-1}}{T_{\rm rev}^2} = (\eta_{\rm rf}/\beta_0)\,\frac{c}{T_{\rm rev}}\,\frac{Q\hat{V}\Big(\sin\big(\frac{\omega_{\rm rf}}{c}ct + \phi_0\big) - \sin\phi_0\Big)}{p_0 c}. \tag{6.26}$$

Interpreting the left hand side as a difference approximation to the second derivative $d^2(ct)/dt_a^2$, one obtains the "Newton's law" equation satisfied by ct;

$$\frac{d^2(ct)}{dt_a^2} = (\eta_{\rm rf}/\beta_0)\,\frac{c}{T_{\rm rev}}\,\frac{Q\hat{V}\Big(\sin\big(\frac{\omega_{\rm rf}}{c}ct+\phi_0\big)-\sin\phi_0\Big)}{p_0c}. \tag{6.27}$$

Especially in situations where ϕ_0 is not changing, this equation is simplified considerably by introducing a new variable ϕ, defined by

$$\phi = \frac{\omega_{\rm rf}}{c}ct+\phi_0, \quad ct = \frac{c}{\omega_{\rm rf}}\,(\phi-\phi_0). \tag{6.28}$$

In terms of ϕ, Eqs. (6.8) and (6.27) become

$$\frac{d(\phi-\phi_0)}{dt_a} = \omega_{\rm rf}\eta_{\rm fr}\delta, \quad \frac{d^2(\phi-\phi_0)}{dt_a^2} = (\eta_{\rm rf}/\beta_0)\,\frac{\omega_{\rm rf}}{T_{\rm rev}}\,\frac{Q\hat{V}}{p_0c}(\sin\phi-\sin\phi_0). \tag{6.29}$$

"Fixed points" are points where the "force" on the right hand side of the second equation vanishes;

$$\sin\phi = \sin\phi_0 \quad \rightarrow \quad \phi=\phi_0, \quad \text{or} \quad \phi=\pi-\phi_0. \tag{6.30}$$

Referring to FIG 6.1, the angle ϕ_0 has already been defined so that the stable fixed point, at the center of the pattern, is at $ct=0$, which is to say at $\phi=\phi_0$. The unstable fixed point, the point of maximum excursion of ct, is therefore given by

$$\phi_{\rm max} = \pi-\phi_0. \tag{6.31}$$

The separatrix separating stable and unstable motion passes through the ct axis at this point. The rest of the separatrix is determined by analogy with "conservation of energy", which will be explained next. Quotation marks indicate that the quantities being discussed do not actually have the dimensions of energy. To exploit the fact that the right hand side of Eq. (6.26) depends only on the dependent variable ϕ, the equation can be written

$$\frac{d^2(\phi-\phi_0)}{dt_a^2} = -\frac{\partial\mathcal{V}}{\partial\phi}, \tag{6.32}$$

where $\mathcal{V}$ is a "potential energy" function given by

$$\mathcal{V} = (\eta_{\rm rf}/\beta_0)\,\frac{\omega_{\rm rf}}{T_{\rm rev}}\,\frac{Q\hat{V}}{p_0c}(\cos\phi-\cos\phi_0+(\phi-\phi_0)\sin\phi_0). \tag{6.33}$$

Note that, without spoiling its use, a constant term $\cos\phi_0+\phi_0\sin\phi_0$ has been subtracted so that $\mathcal{V}$ vanishes at the stable fixed point. The "total energy" is the numerical value of a function $\mathcal{H}$, known as the "Hamiltonian";

$$\begin{aligned}\mathcal{H} &= \frac{1}{2}\Big(\frac{d(\phi-\phi_0)}{dt_a}\Big)^2+\mathcal{V}\\ &= \frac{1}{2}\,\omega_{\rm rf}^2\eta_{\rm rf}^2\,\delta^2+(\eta_{\rm rf}/\beta_0)\,\frac{\omega_{\rm rf}}{T_{\rm rev}}\,\frac{Q\hat{V}}{p_0c}(\cos\phi-\cos\phi_0+(\phi-\phi_0)\sin\phi_0). \end{aligned}\tag{6.34}$$

To recover small oscillation theory one must extract the leading term in the potential energy term;

$$\mathcal{H}(\phi,\delta) = \frac{1}{2}\,\omega_{\rm rf}^2\eta_{\rm rf}^2\,\delta^2-\frac{1}{2}\,((\eta_{\rm rf}/\beta_0)\cos\phi_0)\,\frac{\omega_{\rm rf}}{T_{\rm rev}}\,\frac{Q\hat{V}}{p_0c}\,(\phi-\phi_0)^2. \tag{6.35}$$

As explained previously, the parenthesized factor has to be kept negative for stability; that is, for $\mathcal{H}$ to be positive definite.

As always in oscillatory motion, energy sloshes between kinetic and potential, and phase space trajectories are curves of constant $\mathcal{H}$. In this language, the first term of $\mathcal{H}$ is to be thought of as "kinetic energy". The area enclosed by such a trajectory is known to be an adiabatic invariant ϵ_s. (The true adiabatic invariant is the area in true momentum phase space. Since we work with δ, our phase space area actually shrinks proportional to p_0 which, in the relativistic regime, is proportional to γ_0. This is the well-known "adiabatic damping" as it influences longitudinal motion.) For any individual particle this area is the analog of the Courant-Snyder invariant of transverse motion. For the bunch as a whole it is known as the longitudinal emittance. All particles in the bunch, if they are not to be lost, must lie inside a separatrix enclosing the stable bucket.

Since $\delta = 0$ and $\phi = \pi - \phi_0$ at the maximum excursion point of the motion, the maximum value of $\mathcal{H}$ is given by

$$\mathcal{H}_{\rm max} = -((\eta_{\rm rf}/\beta_0)\cos\phi_0)\,\frac{\omega_{\rm rf}}{T_{\rm rev}}\,\frac{Q\hat{V}}{p_0 c}\big(2 - (\pi - 2\phi_0)\tan\phi_0\big). \tag{6.36}$$

The maximum value of δ is then obtained from the stuation when the "energy is all kinetic" by

$$\delta_{\rm max} = \frac{\sqrt{2\mathcal{H}_{\rm max}}}{(\eta_{\rm rf}/\beta_0)\omega_{\rm rf}}. \tag{6.37}$$

The particular curve for which $\mathcal{H} = \mathcal{H}_{\rm max}$, because it separates stable and unstable motion, is referred to as the "separatrix". It encloses the so-called "stable bucket". The approximate bucket area is $\pi\delta_{\rm max}(ct)_{\rm max}$ when the phase space axes are δ and ct. The bucket area, re-expressed in units of electron-volt-seconds, is given approximately by

$$\mathcal{A} \approx \pi\delta_{\rm max}(ct)_{\rm max} p_0 c/c. \tag{6.38}$$

One of the UAL/USPAS simulations will investigate longitudinal bunch evolution within the stable bucket, emittance growth due to filamentation and passage through transition. "Rebucketing" in which RF of one frequency is gradually replaced by RF of another frequency is also a candidate for simulation. The purpose of rebucketing is to permit the use long bunches during passage through transition and shorter bunches at collision (in a colliding beam facility.)

6.2. Some Formulas for Relatistic Kinematics

For the study of longitudinal dynamics it is necessary to calculate deviation of the revolution period, and for this both the circumference $\mathcal{C}(\tilde{\delta})$ and velocity $v(\tilde{\delta})$ must be calculated as functions of the fractional momentum offset. For electrons the velocity is normally close enough to the velocity of light that the difference from c can be neglected, but for protons the following calculations are necessary. The dynamical variable governing deflection in a magnetic field is the momentum p, which deviates from the central momentum p_0 according to

$$p = p_0(1 + \tilde{\delta}). \tag{6.39}$$

This defining equation for $\tilde{\delta}$ was introduced earlier in Eq. (2.25). Note that Eq. (6.39) is an exact equation and the use of the incremental quantity $\tilde{\delta}$ by no means implies that linearized equations are necessarily being used.

A few relativistic results will now be quoted without proof.

$$\beta = \frac{v}{c}, \quad \gamma = \frac{1}{\sqrt{1-\beta^2}}, \tag{6.40}$$

$$E^2 = p^2c^2 + m^2c^4 = \text{ total energy squared}, \tag{6.41}$$

$$p = mv\gamma, \quad E = mc^2\gamma \tag{6.42}$$

$$v = \frac{pc^2}{E}, \quad \frac{dE}{dp} = \frac{pc^2}{E} = v, \quad \frac{dv}{d\tilde{\delta}} = \frac{p_0}{p}\frac{v}{\gamma^2}. \tag{6.43}$$

These may be familiar or can be looked up or derived. The relation we will actually use, evaluated on the central orbit, is

$$\frac{d}{d\tilde{\delta}}\Big(\frac{1}{v}\Big)_{\tilde{\delta}=0} = -\frac{1}{\gamma_0^2}\frac{1}{v_0}, \tag{6.44}$$

and

$$\frac{1}{v} = \frac{1}{v_0} - \frac{1}{\gamma_0^2}\frac{1}{v_0}(\tilde{\delta} + \frac{1}{2}\frac{v_0^2}{c^2}\tilde{\delta}^2) + \cdots . \tag{6.45}$$

Near transition even the $\tilde{\delta}^2$ term has observable effect.

For comparison purposes a few formulas will next be copied from Jie Wei's thesis[**16**].

$$\frac{1}{1+\Delta t_{\rm vel}(\tilde{\delta})/T(0)} \equiv \frac{v-v_0}{v_0} = \frac{1}{\gamma_0^2}\tilde{\delta} - \frac{3\beta_0^2}{2\gamma_0^2}\tilde{\delta}^2 - \frac{(1-5\beta_0^2)\beta_0^2}{2\gamma_0^2}\tilde{\delta}^3 + \cdots . \tag{6.46}$$

The first two terms agree with Eq. (6.45). The corresponding dependence of circumferential length is defined by

$$\frac{\mathcal{C}(\tilde{\delta}) - \mathcal{C}(0)}{\mathcal{C}(0)} = \frac{\tilde{\delta}}{\gamma_t^2}(1 + \alpha_1\tilde{\delta} + \alpha_2\tilde{\delta}^2 + \cdots). \tag{6.47}$$

For a closed orbit made up of straight line segments, such as `general_fodo.adxf` the coefficients in this expression can be obtained by comparison with Eq. (6.58).

Using the expansion coefficients just defined, a more detailed expansion of the slip factor can be defined;

$$\eta_{\rm rf} = \eta_0 + \eta_1\tilde{\delta} + \eta_2\tilde{\delta}^2 + \cdots . \tag{6.48}$$

The coefficients are given by

$$\eta_0 = \frac{1}{\gamma_t^2} - \frac{1}{\gamma_0^2}, \quad \eta_1 = \frac{\alpha_1}{\gamma_t^2} - \frac{1}{\gamma_0^2}\Big(\frac{1}{\gamma_t^2} - \frac{1}{\gamma_0^2}\Big) + \frac{3\beta_0^2}{2\gamma_0^2}, \ldots . \tag{6.49}$$

This expansion is useful for a purely analytical treatment of passage through transition. But for numerical simulation there is little point in using such term-by-term expansions. Instead, the direct evaluation of the time delay $\Delta t(\delta)$, as given, for example, in Eq. (6.59) below, gives the needed phase slip per turn at the RF cavity for arbitrary δ.

6.3. The Off-Momentum Orbit Length

For longitudinal dynamics the arrival time of the particle at the RF cavity is of critical importance. As well as depending on particle velocity, the arrival time also depends on the off-momentum path length. Because of the elongated geometry in a large ring, the geometry of this calculation is difficult. The so-called "momentum compaction factor" α_0, (also known as $1/\gamma_t^2$) the fractional momentum proportional increase in circumference $\mathcal{C}$, is calculable in terms of local radius $\rho(s)$ and off-momentum closed orbit function $\tilde{D}(s)$;

$$\alpha_0 \equiv \frac{1}{\gamma_t^2} = \frac{d\mathcal{C}/\mathcal{C}(0)}{d\tilde{\delta}} = \frac{1}{\mathcal{C}(0)} \int_0^{\mathcal{C}(0)} \tilde{D}(s)\, \frac{ds}{\rho(s)}. \tag{6.50}$$

For reasons already discussed γ_t is known as "transition gamma" A rule-of-thumb formula gives $\tilde{D}_{\text{typ.}}$ in terms of the horizontal tune Q_x;

$$\tilde{D}_{\text{typ.}} \approx \frac{\mathcal{C}(0)}{2\pi Q_x^2} \tag{6.51}$$

From this follows the semi-quantitative heuristic relation,

$$\gamma_t \approx Q_x. \tag{6.52}$$

This section now proceeds to calculate the excess arc length for our thin element, general_fodo lattice. For this calculation, to simplify the geometry, we assume the dipole fills the entire half-cell length ($l_d = l$) but still apply the entire deflection as a kink at the midpoint. This is not very accurate but the (relatively) simple formulas convey the essence and the difficulties of the calculation.

As shown in FIG 6.2, the on-momentum, design path length through one halfcell is $\mathcal{C}_{1/2} = \frac{2l}{\Delta\theta} \tan\frac{\Delta\theta}{2}$, Referring to FIG 6.3, and using coordinate axes s, x centered at q_1 with s being normal to the multipole plane, the equation of the off-momentum closed orbit is

$$x = (\tilde{D}_1 - q_1 \tilde{D}_1 s)\tilde{\delta}. \tag{6.53}$$

The equation of the transverse plane through the dipole center is

$$s = (x + \rho_0) \tan\frac{\Delta\theta}{2}, \tag{6.54}$$

where $\rho_0 = l/\Delta\theta$ is the bending radius of the reference orbit. At the center of a focusing quad, using its local (s, x) coordinates, the coordinates of the off-momentum closed orbit are $(0, \tilde{D}_1\tilde{\delta})$, and the coordinates of the intersection with the centerline of bending magnet are

$$s_{d1}(\tilde{\delta}) = \frac{(\rho_0 + \tilde{D}_1\tilde{\delta}) \tan\frac{\Delta\theta}{2}}{1 + q_1\tilde{D}_1\tilde{\delta}\tan\frac{\Delta\theta}{2}}, \tag{6.55}$$

$$x_{d1}(\tilde{\delta}) = \frac{\tilde{D}_1\tilde{\delta} - q_1\tilde{D}_1\tilde{\delta}\rho_0 \tan\frac{\Delta\theta}{2}}{1 + q_1\tilde{D}_1\tilde{\delta}\tan\frac{\Delta\theta}{2}}.$$

Starting from q_2 the corresponding quantities $s_{d2}(\tilde{\delta})$ and $x_{d2}(\tilde{\delta})$ are obtained by replacing 1 by 2. The contribution to the circumference of the off-momentum closed orbit from one halfcell is

$$\mathcal{C}_{1/2}(\tilde{\delta}) = \sqrt{(x_{d1}(\tilde{\delta}) - \tilde{\delta}\tilde{D}_1)^2 + s_{d1}^2(\tilde{\delta})} + \sqrt{(x_{d2}(\tilde{\delta}) - \tilde{\delta}\tilde{D}_2)^2 + s_{d2}^2(\tilde{\delta})}. \tag{6.56}$$

There is a useful numerical trick that avoids the need for evaluating these square root expressions. (In a "kick code" the trajectory consists entirely of straight line segments like this.) Defining $\mathcal{C}(\tilde{\delta})$ as the off-momentum path length along a straight line segment, and $\mathcal{C}(0)$ as the corresponding on-momentum path length, the quantity needed is

$$\mathcal{C}(\tilde{\delta}) - \mathcal{C}(0) = \frac{\mathcal{C}^2(\tilde{\delta}) - \mathcal{C}^2(0)}{\mathcal{C}(\tilde{\delta}) + \mathcal{C}(0)} \approx \frac{\mathcal{C}^2(\tilde{\delta}) - \mathcal{C}^2(0)}{2\mathcal{C}(0)}. \tag{6.57}$$

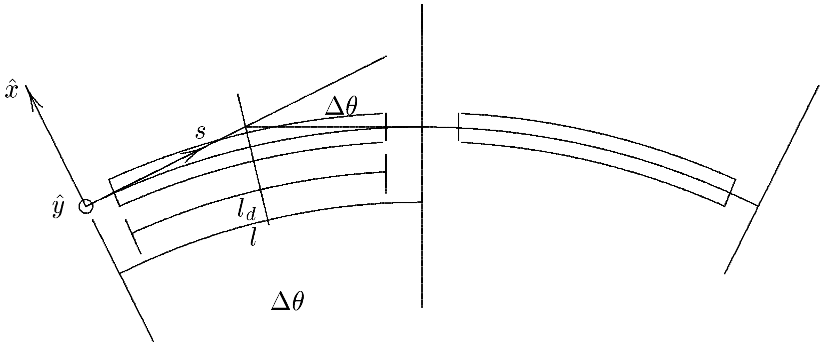


FIGURE 6.2. Geometry for transition gamma calculation.

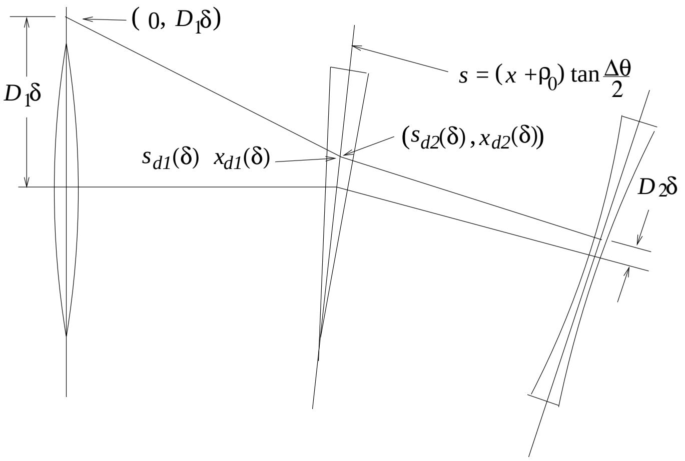

$(0, D_1\delta)$
$D_1\delta$
$s = (x + \rho_0) \tan\frac{\Delta\theta}{2}$
$(s_{d2}(\delta), x_{d2}(\delta))$
$s_{d1}(\delta)\ \ x_{d1}(\delta)$
$D_2\delta$

FIGURE 6.3. Geometry for calculation of off-momentum, polygonal path length. The on-momentum and off-momentum orbits are shown. The bending magnet is represented by a prism symbol.

For each straight line segment the numerator expression is one of the factors under the square root signs in Eq. (6.56). The final approximation here is adequate for most purposes. Applying this formula and Eq. (6.56) to the `general_fodo.adxf` lattice, the fractional momentum-dependent increase in circumference is

$$\frac{\mathcal{C}(\tilde{\delta}) - \mathcal{C}_{1/2}}{\mathcal{C}_{1/2}} \approx \frac{(x_{d1}(\tilde{\delta}) - \tilde{\delta}\tilde{D}_1)^2 + s_{d1}^2(\tilde{\delta}) + (x_{d2}(\tilde{\delta}) - \tilde{\delta}\tilde{D}_2)^2 + s_{d2}^2(\tilde{\delta}) - 0.5\mathcal{C}_{1/2}^2}{\mathcal{C}_{1/2}^2}. \tag{6.58}$$

(For improved numerical accuracy, the TEAPOT code expands the numerator expressions to take advantage of the explicit cancellation of the dominant length terms.)

With there being $n/2$ cells in all, the off-momentum arrival time delay after one revolution, relative to the reference trajectory, is

$$\begin{aligned}\Delta t(\tilde{\delta}) &= \frac{n}{2}\frac{\mathcal{C}_1(\tilde{\delta})}{v} - \frac{n}{2}\frac{\mathcal{C}_1(0)}{v_0} = \frac{n}{2}\frac{\mathcal{C}_1(\tilde{\delta}) - \mathcal{C}_1(0)}{v} + \frac{n}{2}\mathcal{C}_1(0)\Big(\frac{1}{v} - \frac{1}{v_0}\Big) \qquad (6.59)\\ &= \Delta t_{circ}(\tilde{\delta}) + \Delta t_{vel}(\tilde{\delta}).\end{aligned}$$

The two terms in Eq. (6.59) can be interpreted separately. The first is the delay in arrival time due to altered circumference

$$\Delta t_{circ}(\tilde{\delta}) = \frac{n}{2}\frac{\mathcal{C}_1(\tilde{\delta}) - \mathcal{C}_1(0)}{v} \simeq \frac{1}{v_0}\frac{d\mathcal{C}(\tilde{\delta})}{d\tilde{\delta}}\tilde{\delta} = \text{ normally positive for } \tilde{\delta} > 0. \qquad (6.60)$$

$$\Delta t_{circ}(\tilde{\delta}) = \frac{\mathcal{C}(0)}{v_0\gamma_t^2}\tilde{\delta} \qquad (6.61)$$

The second term in Eq. (6.59) is the delay in arrival time due to altered velocity;

$$\Delta t_{vel}(\tilde{\delta}) = \mathcal{C}(0)\Big(\frac{1}{v} - \frac{1}{v_0}\Big) \simeq -\frac{\mathcal{C}(0)}{v_0}\frac{1}{\gamma_0^2}\tilde{\delta}, \qquad (6.62)$$

which is always negative for $\tilde{\delta} > 0$. Combining Eqs. (6.61) and (6.62) yields

$$\Delta t = T_{\rm rev}\Big(\frac{1}{\gamma_t^2} - \frac{1}{\gamma_0^2}\Big)\tilde{\delta}, \qquad (6.63)$$

where $T_{\rm rev}$ is the period of one on-momentum revolution. Near transition, where this expression vanishes, more accurate determination of Δt may be called for.

The term Δt_{vel} does not depend on lattice design but Δt_{circ} can be altered over a large range by altering the machine optics. Notice the possibility that $\Delta t(\tilde{\delta}) = 0$, which is known as the condition for "transition". In this condition there is no change in revolution period accompanying an increase in momentum because the increase in circumference cancels the effect of increased velocity. In electron accelerators γ^2 is normally so large that transition would be crossed only at energies far below the injection energy. In proton accelerators transition crossing normally occurs for energies of several or several tens of GeV. If transition is regarded as too undesirable, $1/\gamma_t^2$ can be reduced to zero, or even made negative, by causing the average dispersion to be artificially small (by making the dispersion negative over large sectors of the ring.) In this case γ_t becomes imaginary, which violates no physical principle. The toy lattice `isochronous.adxf` has been designed to give $\gamma_t = \infty$.

6.4. Numerical Approach Using TIBETAN

Within UAL, instead of element-by-element tracking, it is (optionally) possible to treat transverse motion in linearized fashion while representing longitudinal dynamics with faithful nonlinear formulas. This formalism (due to Jie Wei) is called TIBETAN. It is a semi-empirical formalism in which propagation around the ring is represented by linear transfer matrices and the representation of RF cavity (or cavities) effects include realistic nonlinear dependencies. For the `general_fodo.adxf` lattice introduced earlier, most of the needed parameters have already been obtained.

The `APDF` (Accelerator Propagator Description Format) file for this simulation is especially simple. In its entirety it is

```
<apdf>
  <propagator id="tibetan" accelerator="blue">
     <create>
        <link algorithm="TIBETAN::OneTurnTracker" sector="Default" />
     </create>
  </propagator>
</apdf>
```

There is no RF entry because the RF parameters are under control of the GUI. This permits these parameters to be adjusted without requiring any code recompilation.

Simulation 6.1. *Investigate the longitudinal behavior of the* `racetrack` *lattice. Though there is an element named* `rf` *in this lattice, it is designated as a* `drift` *element. This is* not *inconsistent with the design of the* `longitudinal` *simulator. The reason for this is that the* `longitudinal` *simulator "takes control" of the longitudinal dynamics, applying a longitudinal kick to each particle on each turn. This puts the relevant longitudinal parameters under the control of the GUI.*

Since the `racetrack` *lattice was originally generated with electron rings in mind, it might not seem to be appropriate as an accelerator for gold ions. Nevertheless, to begin, leave most parameters the same as for the simulation of acceleration of fully ionized gold ions in* RHIC, *which stands for "Relativistic Heavy Ion Collider". But the beam energy should be adjusted to be below transition. Before being able to make this setting you have to find γ_t. Allowing the simulation to run you should see filamentation like that exhibited in FIGs 6.4 and 6.5. Find the emittance growth the beam has suffered due to filamentation, from the time it is injected until it has equilibrated. Since there are no quantum fluctuations, nor radiation damping, this simulation is not at all applicable to an electron storage ring. Then find the fraction of beam captured by the RF. Finally, change the initial bunch length and momentum spread to be "matched", and as large as possible so that all or most of the particles are captured. Calculate the longitudinal emittance of this maximal beam bunch. Compare your values with formulas given in the text.*

Simulation 6.2. *Reproduce FIG 6.4 and FIG 6.5.*

Problem 6.3. *This problem should only be attempted by individuals with experience in object-oriented programming, preferably C++. The previous tutorial identified the location of the code to be modified to introduce* ad hoc *damping into the longitudinal motion. The problem here is to study the code, starting with* `~/USPAS/examples/longitudinal/src/run.cc` *to figure out how the program calls the code that was modified in the tutorial.*

6.5. Typical Parameter Values for RHIC

A consistent set of parameter values can be taken from a paper by Montag and Kewisch[**17**], which describes longitudinal bunch manipulation in RHIC. Some of these values are given in Table 6.1. They apply to the acceleration of fully-ionized gold ions in RHIC. Harmonic number h is defined by $h = \omega_{\rm rf}/\omega_{\rm rev}$ and nonlinear momentum compaction factor α_1 was defined in Eq. (6.47).

TABLE 6.1. RHIC parameters for fully stripped gold ions[**17**].

Property	Symbol	Value	Unit
Transition gamma	γ_t	22.8	
Acceleration rate	$d\gamma/dt \equiv \gamma'$	0.5	1/s
Maximum off-energy parameter	$\delta_{\rm max}$	0.0043	
Circumference	$\mathcal{C}(0)$	3833.845	m
Atomic number	Z	79	
Atomic weight	A	197	
Transition energy per nucleon	E_t	21.4	GeV
Peak RF voltage	$\hat{V}$	300	kV
Harmonic number	h	360	
Nonlinear momentum compaction parameter	α_1	-0.54	
Characteristic nonadiabatic time	T_c	0.053	s
Chromatic nonlinear time	$T_{\rm nl}$	0.188	s
Transition gamma jump	$\Delta\gamma_t$	1.0	
Transition gamma jump time	$\Delta t_{\rm jump}$	0.030	s

6.6. Simulation of Longitudinal Motion

The following figures show various examples of longitudinal bunch evolution, as exhibited by the UAL `longitudinal` simulator. The conditions are indicated in the captions, and in the inset parameter boxes. (The contents of these baxes are not readable in the figures shown in this text, but, using the GUI, the boxes can be resized for readability.) In each case 10,000 gold ions are tracked for for some thousands of turns, and the distributions are updated every 100 turns. The simulation uses a realistic representation of the RHIC accelerator. In FIG's 6.4 and FIG 6.5 the figures are grouped into quartets defined in the figure caption. In longitudinal phase space plots the abscissa is longitudinal phase ϕ and the ordinate is $\Delta E/(p_0c)$. "Mountain range" plots (which are commonly available in the control room using a longitudinal profile monitor) consist of a series of one dimensional plots with beam charge density represented by color code, or by shading, at regular intervals. By shifting the plots up the plot as time progresses, evolution of the longitudinal beam profile is displayed. The three dimensional spatial bunch distribution is exhibited in the upper left corner of each quartet.

6.7. Longitudinal Dynamics Near Transition

The condition that must be satisfied for the adiabatic analysis described so far to be valid is

$$\frac{1}{\Omega_s^2}\,\frac{d\Omega_s}{dt_a} << 1. \tag{6.64}$$

Stating this in words, during unit advance of longitudinal phase, the fractional change in Ω_s is small compared to 1. This condition cannot be met at transition since $(\eta_{\rm rf}/\beta_0)$, and hence Ω_s, vanish at that point. Redefining the time variable t_a to be zero at the instant of crossing transition, the beam energy evolves according to

$$\gamma = \gamma_t + \gamma_t' t. \tag{6.65}$$

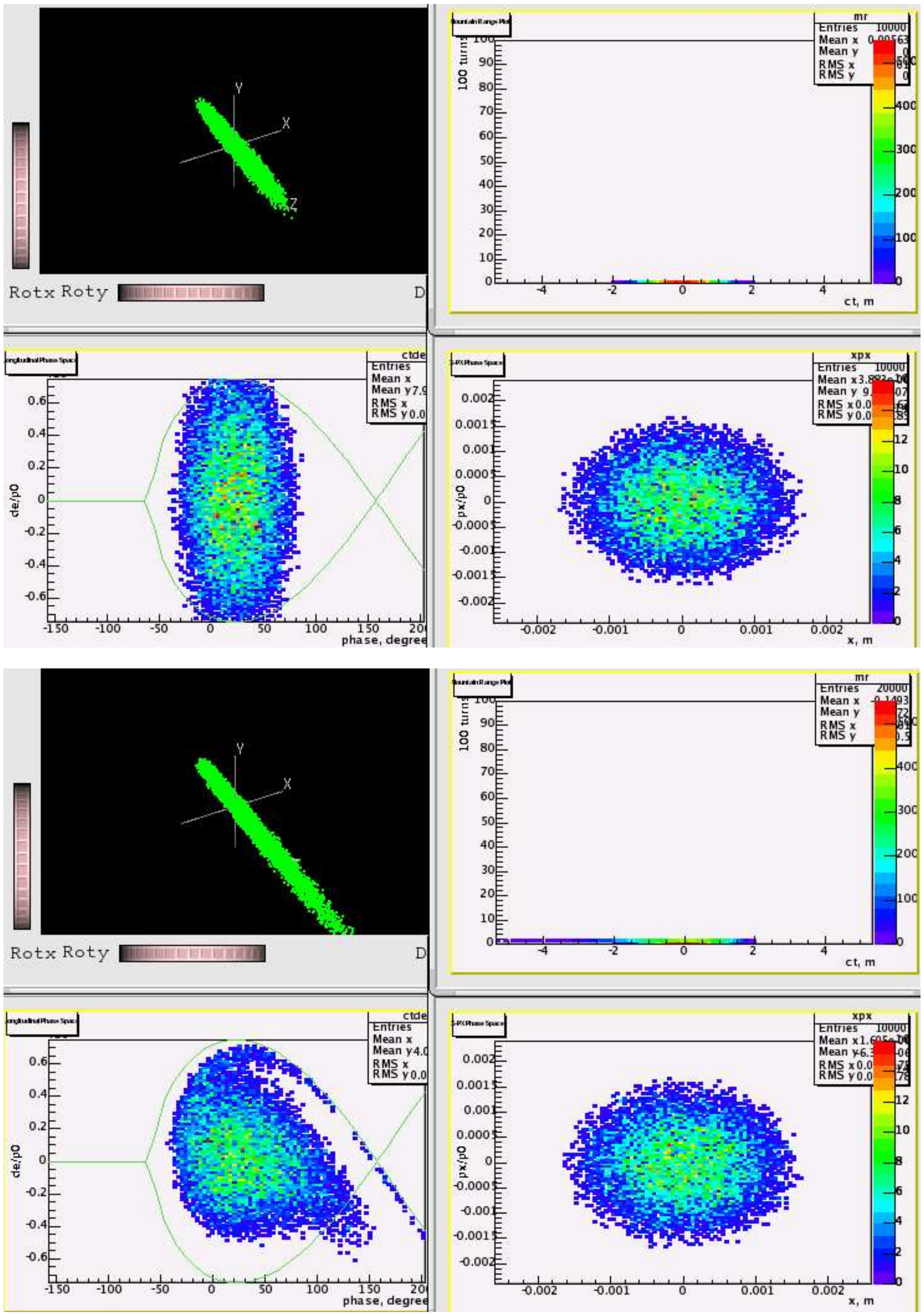


FIGURE 6.4. Top: Particle distribution just after injection of slightly over-sized 10000 particle beam. Bottom: Particle distributions 100 turns after injection of the same beam. Counterclockwise, starting at upper left, the figures are 3D space, longitudinal phase space, transverse phase space, and turn number vs. longitudinal.

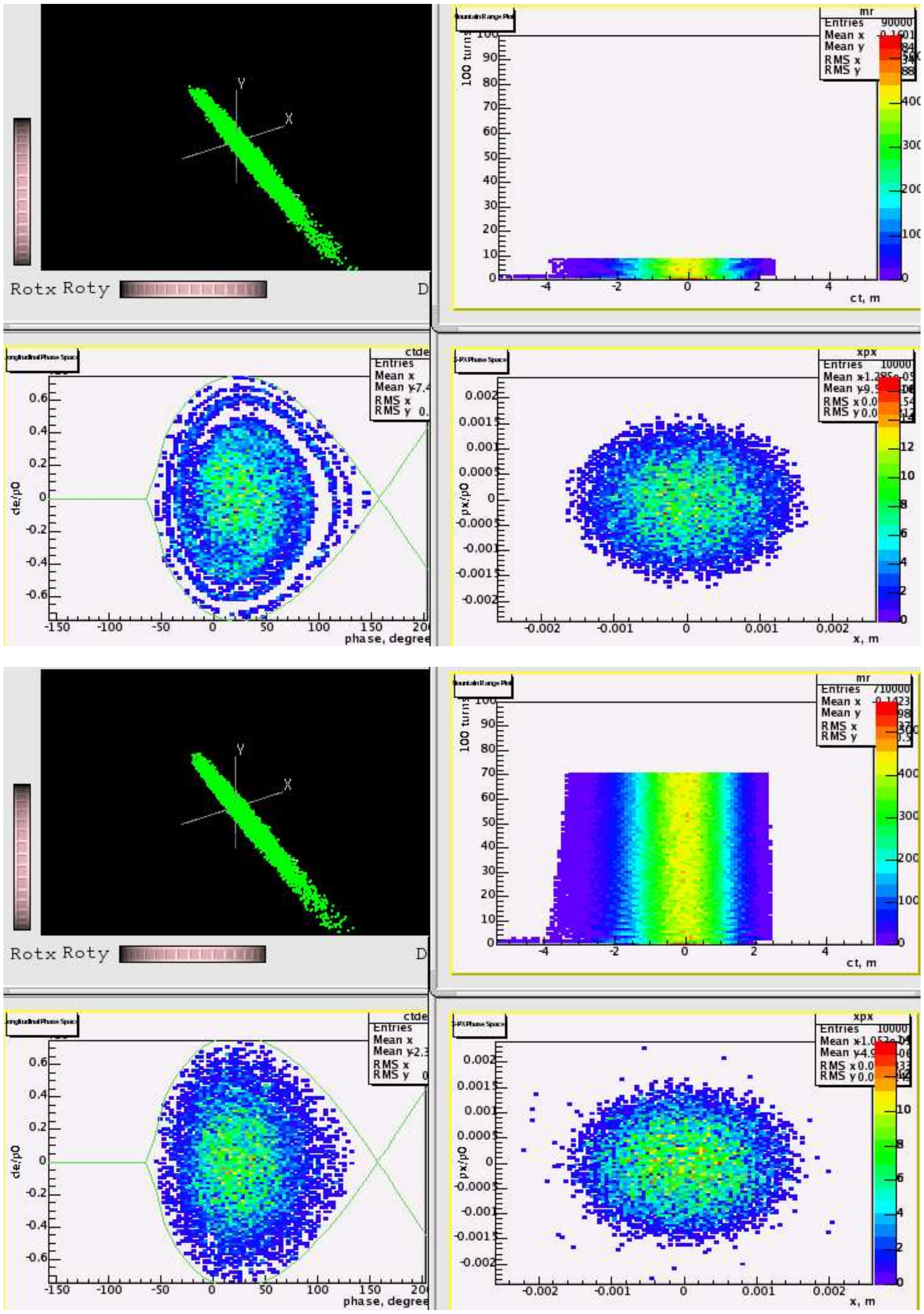


FIGURE 6.5. Top: Particle distribution 800 turns after injection of the same beam. Bottom: Particle distributions 7000 turns after injection of the same beam.

Some form of alternative analysis (such as numerical simulation) has to be performed during a time interval containing the transition time. The length T_c of an adequately long interval can be obtained by working problem 6.4.

Another complication of transition crossing is that, because the particles have different synchrotron oscillation amplitudes, and hence different values of Ω_s, they do not all cross transition at the same time. This means that, in principle, the time origin has to be defined for each particle individually. For beams of small enough longitudinal emittance this effect is negligible since Ω_s is essentially the same for all particles. But for bunches that nearly fill the stable bucket the effect becomes important. In fact, in that limit, some loss of particles out of the stable bucket is inevitable. Again this effect is best studied numerically.

To quantify the nonlinear effect one defines a "chromatic nonlinear time" $T_{\rm nl}$, such that, for a particle of maximal energy offset $\pm\delta_{\rm max}$, the transition-crossing time is shifted by $\pm T_{\rm nl}$. As a following problem, $T_{\rm nl}$ can be derived using formulas given previously. A typical numerical value is given in section 6.5.

If nothing else were to be done, the beam would become unstable after transition crossing, and it would blow up inexorably and be lost. One therefore switches the RF phase as has been discussed previously. The time taken for this phase switch to occur can be assumed negligible relative to the other times that have been discussed.

If beam degradation during transition crossing is too great some other longitudinal gymnastics are called for. Because of the large inductance of superconducting (or any other) magnets, it is typically not practical to increase γ_t' by increasing the magnetic ramp rate. But recall from Eq. (6.52) that γ_t depends on lattice properties and hence γ_t' can potentially be increased by shifting Q_x toward lower values as the beam energy increases through transition. This is done by impulsively altering the currents in some lattice quadrupoles. Of course this cannot be done instantaneously. For the example given in the next section, for a jump $\Delta\gamma_t = 1$, the time taken is $\Delta t_{\rm jump} = 30\,{\rm ms}$. This increase the effective value of γ_t' by a factor of 60 which, according to Eq. (6.67), reduces T_c by a factor of about 4.

PROBLEM 6.4. *For times t close to the time of transition crossing, show that the synchrotron frequency is given approximately by*

$$\Omega_s^2 \approx \frac{|t|}{T_c^3}, \tag{6.66}$$

where

$$T_c = \Big(\frac{\pi E_t \beta_t^2 \gamma_t^3}{Q\hat{V}|cos\phi_t|\gamma_t' h\omega_{\rm rev}^2}\Big)^{1/3}. \tag{6.67}$$

Time T_c is referred to as the "characteristic nonadiabatic time". Typical values for T_c and other parameters are given in section 6.5.

PROBLEM 6.5. *Estimate the chromatic nonlinear time $T_{\rm nl}$ to be*

$$T_{\rm nl} = \frac{|\alpha_1 + 1.5\beta_0^2\delta_{\rm max}\gamma_t|}{\gamma_t'}. \tag{6.68}$$

SIMULATION 6.3. *Reconstitute the simulation illustrated in FIG 6.6. In that figure, to illustrate phase space evolution, an unrealistically small emittace bunch (especially as regards energy spread) was used. Alter the phase space before transition to be better matched and to almost fill the stable bucket. Then calculate the growth of all three emittances ϵ_x, ϵ_y, and ϵ_s as gold ions are accelerated through transition in RHIC.*

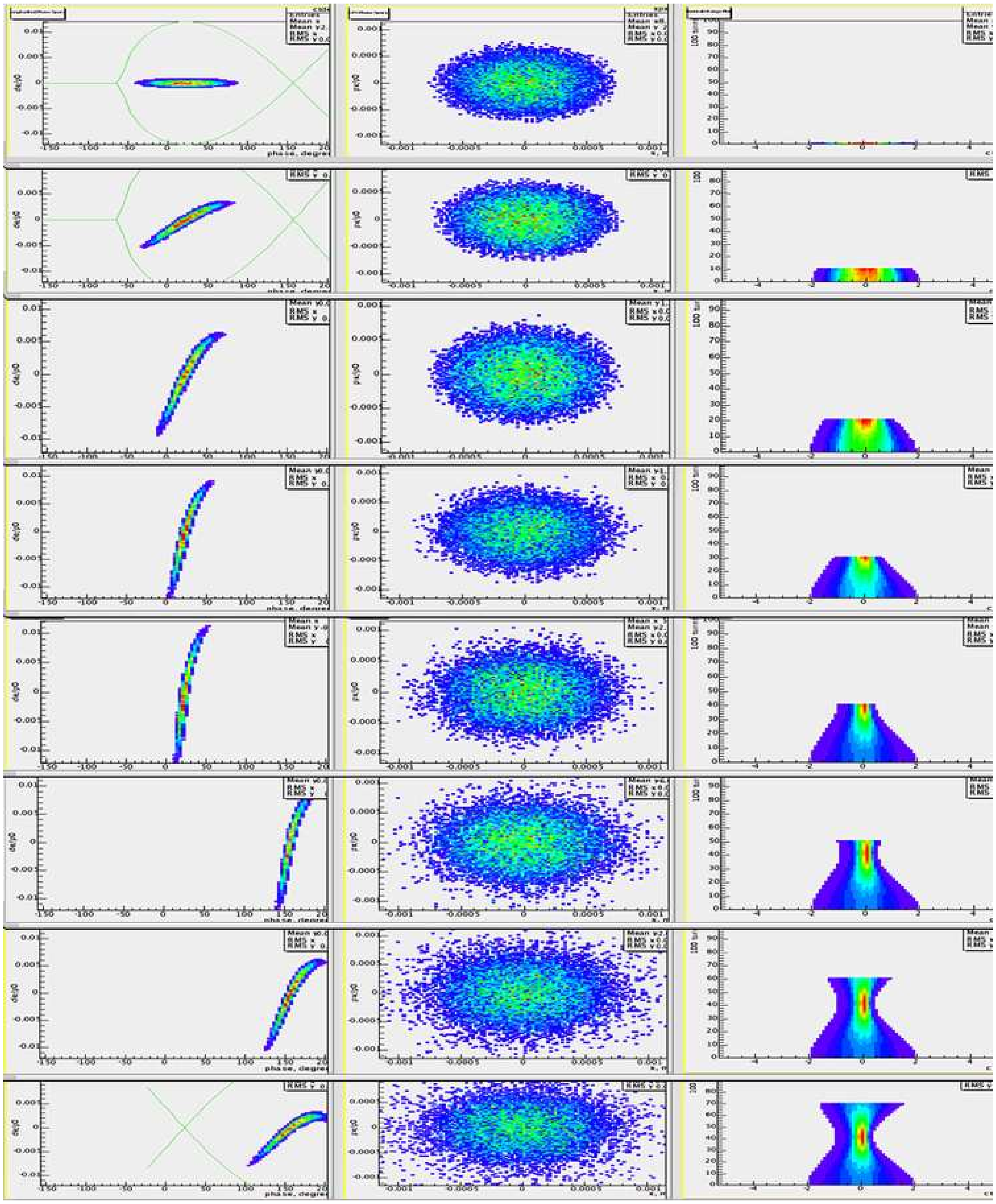


Figure 6.6. Series of distribution snapshots (every 1000 turns for a beam of 10000 particles) starting just before, and ending just after transition. The RF phase is jumped at the instant of transition crossing.

CHAPTER 7

Decoherence and Filamentation

7.1. Introduction

Even in electron rings, where some damping due to synchrotron radiation is to be expected, the damping is so weak that the Courant-Snyder invariant of any one particle is, in fact, almost constant for thousands of turns. The sensitivity of the beam position monitors (BPM) used to record such motion are usually "in the noise" unless a bunch of some large number N of co-moving particles, equal to millions or more, are sensed. If all of these particles were exactly superimposed then the observed signal would simply be N times greater than the signal from a single particle. Of course this is not the case. Commonly there is some physical phenomenon limiting the density of particle in phase space, with the result that the more particles there are present in a bunch, the larger the phase space volume containing all particles tends to be. Even so, if all deflections in the ring were linear (as that term has been defined in earlier chapters) then the signals induced in a BPM would be very nearly the same as that of a single, magnitude N "macroparticle" situated at the centroid of the moving bunch. Again, this is unrealistic, since the linearity requirement is never met exactly.

Many effects differentiate bunch motion and single particle motion. Some of these, such as space charge forces, including beam-beam forces in colliding beam facilities, beam wall forces, and coherent synchrotron radiation, are further complicated by the need to treat the charges both as sources (from which force fields have to be calculated) and dynamical objects whose trajectories need to be determined. Other multiparticle topics, such as feedback and stochastic cooling, are complicated by the fact that external pickups and kickers are implicated in the bunch dynamics.

To avoid these complications, in this chapter, particles are assumed to not interact with each other, or with other "external" apparatus (not including the magnets and RF cavities making up the basic lattice.) Even so, there are important multiparticle effects, that go by names such as filamentation, decoherence, Landau damping, and beam echos. Even though these phenomena are all based on essentially the same basic physics, the different terms are used to distinguish among seemingly different observed phenomena. These are the topics of this chapter.

7.2. Experimental Observation

An example of filamentation from CESR[**18**] is exhibited in FIG 7.1. This data was obtained only by simulation, but signals observed in real life were in semi-quantitative agreeement. For this plot a bunch of electrons was injected off-axis and tracked for 1000 turns. The centroid displacement is calculated each turn and recorded as a point on the plot. Since this is an electron beam, some damping, due to synchrotron radiation, is expected. But the observed damping time (seen from

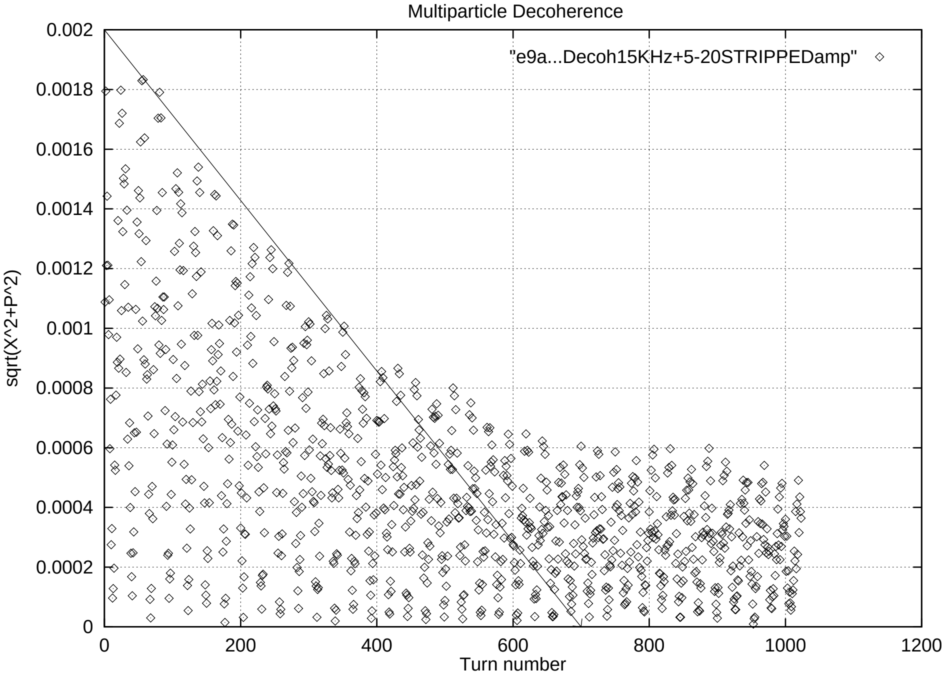


FIGURE 7.1. Multiparticle decoherence. Using the modeling program TEAPOT, an appropriately distributed "beam" consisting of 200 macroparticles, $\sigma_E/E = 0.0006$, is tracked for 1024 turns and the horizontal centroid coordinate is plotted each turn.

the plot to be about 700 turns) is some 10 times shorter than can be accounted for by radiation damping. In this case the nominal tune setting has $Q_x = Q_y$ and the chromaticities were large and highly unbalanced, $Q'_x = +5$, $Q'_y = -20$, (in connection with a study of an effect called "chromaticity sharing"). The data of FIG 7.1 look very much like what would be observed in single particle dissipative motion with a damping time of 700 turns. The term "Landau damping", applicable in this case, includes the term *damping* even though there is no dissipation anywhere present in the system. The effect is also known as *decoherence.* In a bunch of particles injected off-axis, if the bunch dimensions are smaller than the offset, all particles have sufficiently the same amplitude that they each contribute the same amplitude to the centroid signal. But there is inevitably a *tune spread*; call it ΔQ, typically a part in a thousand or so. With $\Delta Q = 0.001$, two particles with identical starting conditions, but with tunes differing this much would, after 500 turns, have phase space phase advances differing by π. In this condition their contributions to the centroid amplitude would cancel. This would give the centroid motion the appearance of being damped, even though there is no actual dissipation present. This phenomenon is known as "decoherence", conveying the notion that particles initially in phase, gradually drop out of phase over multiple turns.

The fundamental tune spread causing the centroid damping visible in FIG 7.1 is due to momentum spread. In the presence of large chromaticities this causes the large tune spread that causes the observed decoherence. This form of damping is present even if the lattice is perfectly *linear*. The result is the approximately exponential damping observed in the figure. It will be seen in the next section that the apparent damping due to decoherence is not necessarily exponential, whether it is observed in the laboratory or in a simulation.

Furthermore, because of the relatively small number of particles used in this (or any) simulation, the damping cannot be expected to be faithfully represented over times much longer than are shown in the plot. For such a small number of particles as 200, once the apparent centroid motion has dropped to some “floor” value, it can be expected to exhibit erratic motion depending on the accidental constructive and destructive combinations of the particles present.

The phenomenon of *filamentation* is closely related to the decoherence just described, in that the effect is due to tune spread. The different terminology is used to convey the idea that the tune spread in the case of filamentation is due to *nonlinear* forces (of the RF waveform in this case). An example of longitudinal *filamentation* was shown in the lower left of the sequence of plots in FIG 6.4 and FIG 6.5. These are phase space plot that corresponds to the “mountain range plots” shown in the same figures. Nonlinearity causes the longitudinal tune to depend on longitudinal amplitude. As a result the rate of revolution in phase space changes with increasing phase space radius. This causes the beam extremities to “shear” gradually as time advances. According to Liouville’s theorem, this process, even though it depends on nonlinearity, should still preserve the local density of particles. This is *not* contradicted by the figure. The final figure in the sequence o plots ending with FIG 6.5 shows the same phase space distribution a long time later, after the extremities have completely wrapped around many times. The wispy tails have become so narrow that they deserve to be called filaments. This picture is still consistent with conservation of local phase space density. But this has become academic because the filaments are so skinny and so hopelessly entwined with areas initially devoid of particles. One could say that vacuum has been stirred into the fluid. For all practical purposes this has reduced the density of particles in phase space. This still does not contradict Liouville’s theorem. But, for all practical uses of the beam, its density has been diluted. The corresponding *emittance increase* due to filamentation can be inferred from the data inserts shown in the figures.

7.3. Analytic Treatment of Decoherence

The decoherence and filamentation phenomena mentioned so far can be analysed theoretically. As well as knowing the distribution of amplitudes it is necessary also to know the dependence of the tunes on the transverse and longitudinal amplitudes. Dependence of tunes on amplitude for the `racetrack.sxf` file are shown in FIG 7.2,

Phase space evolution can be studied with plots like FIG 7.3, which shows (x, p) betatron phase space. Such plots, showing a particle’s position each turn, are also known as “Poincaré plots”. Actually, in order to suppress the dominant, small amplitude evolution, this is a Poincaré plot in a frame rotating at a rate such that a small amplitude particle appears not to move at all. Vectors in the figure show, therefore, changes in position over and above what their linear motion would cause.

The scales have been adjusted so that (linearized) motion in phase space is along circles centered on the origin, with phase advance per turn $\Delta\mu_0$. Scaling p

tune vs momentum curves for racetrack lattice needed her

FIGURE 7.2. Average transverse tunes exhibited by a single particle executing longitudinal oscillations in the `racetrack` lattice, for various chromaticities Q'_x and Q'_y

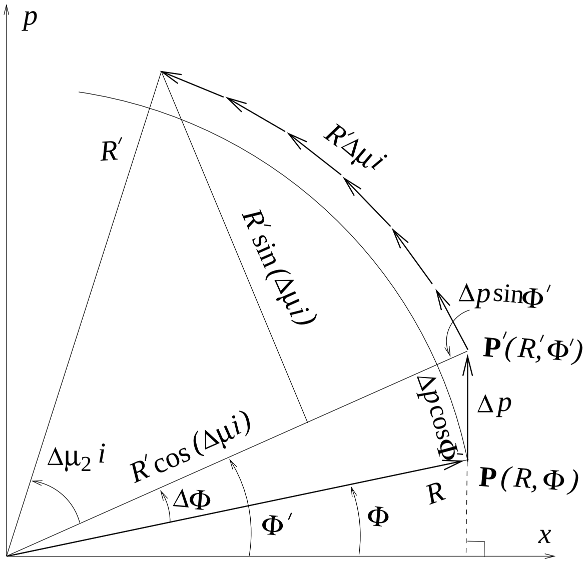


FIGURE 7.3. Evolution with turn number i of a point $\mathbf{P}$ in betatron (horizontal) phase space, as viewed from a frame rotating at nominal phase advance per turn μ_0. The trigonometry of this figure only makes sense for small perturbations ($\Delta p << R$) which is assumed.

to be a length, and assuming the beam is Gaussian and isotropic in phase space, the beam distribution can be expressed either as $P_R(R)$ or as $P_x(x)P_p(p)$, depending on whether polar or cartesian coordinates are employed. The particles are also distributed with distribution $P_{\hat{\delta}}(\hat{\delta})$ in $\hat{\delta}$ which is the maximum value (as the particle oscillates longitudinally) of its fractional momentum deviation $\delta p/p$. The distributions are given by

$$\begin{aligned} P_x(x) &= \frac{1}{\sqrt{2\pi}\sigma}\exp\Big(-\frac{x^2}{2\sigma^2}\Big), \\ P_p(p) &= \frac{1}{\sqrt{2\pi}\sigma}\exp\Big(-\frac{p^2}{2\sigma^2}\Big), \\ P_{x,p}(x,p) &= P_x(x)P_p(p) = \frac{P_R(R)}{2\pi R}, \quad \text{or} \quad P_R(R) = \frac{R}{\sigma_x^2}\exp\Big(-\frac{R^2}{2\sigma^2}\Big). \end{aligned} \tag{7.1}$$

The last distribution can also be expressed as a joint probability distribution $P_{R,\Phi}(R,\Phi) = P_R(R)/(2\pi)$. In every case the probability of a particle lying in a differential interval of the subscripted variable(s) is obtained as the P-function multiplied by a differential (or product of differentials) of the indicated variable(s).

To start the decoherence investigation a "kick" Δp is administered to every particle in the beam at $i = 0$ and hence also to the beam centroid. Motion of a particle initially at point $\mathbf{P}$ is shown in FIG 7.3. If every particle advances at the same angular rate, the centroid does the same and the centroid radius remains constant. But, in general, since $\mu(R,\Phi,\delta)$ depends on the location of $\mathbf{P}$, as well as on δ, the particle motions "decoherere" causing the centroid amplitude to "damp".

Digression. There may or may not be a subsequent recoherence. The predominant decoherence/recoherence occurs through each cycle of synchrotron oscillation. As the energy of a particle oscillates due to synchrotron oscillation, the particle tune is too small when the particle energy is positive (relative to the reference particle). As a result its betatron phase accumulates negatively. On the other hand, when the relative energy is negative the betatron phase accumulates positively. During one complete cycle of synchrotron oscillation the net betatron phase accumulation is zero to excellent accuracy. At the instant the transverse kick is applied the longitudinal phases are distributed uniformly. Thereafter, depending on their starting longitudinal phase, some particles initially gain betatron phase and others lose betatron phase. This causes decoherence and (transverse) centroid damping. At a later time, exactly one synchrotron period after the initial kick, every particle, whatever its initial longitudinal phase had been, will have completed exactly one cycle of synchrotron oscillation. As just shown, the excess betatron phase accumulation during this time will have averaged exactly to zero. As a result the bunch "unscrambles" itself, and reconstitutes the original just-kicked distribution. This phenomenon is commonly observed in most accelerator control rooms. The reconstituted pulse is often referred to as an "echo". The effect is simulated in FIG 7.6. There are also situations in which even more complicated, even with nonlinearity-caused decoherence, that exhibit echos. **End of digression**.

To simplify the present discussion, let us neglect synchrotron oscillation (either because the beam energy spread is negligible or because we will be concerned with times long compared to the synchrotron period). Averaging over the longitudinal motion, any surviving (small amplitude, transverse) tune dependence is expressable

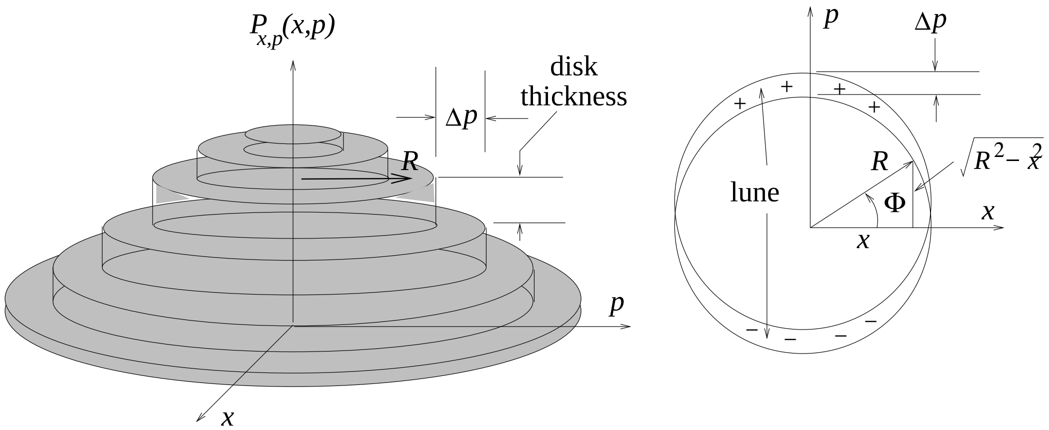


FIGURE 7.4. On the left the betatron phase space distribution is visualized as a sum of distributions, uniform over disks of radii successively changing in steps of Δp. This permits deviations from the unkicked distribution to be represented by positive and negative distributions uniform over the "lunes" shown on the right.

as dependency of phase advance $\mu(\hat{\delta})$, where $\hat{\delta}$ is the longitudinal Courant-Snyder invariant expressed as the maximum energy offset.

We assume the transverse decoherence is due entirely to the "shearing" motion along circles of different radius in phase space for different values of $\hat{\delta}$. This neglects the small effect that, because of nonlinearity at large R, the transverse phase space curves, even while remaining regular, become distorted (though not chaotic).

For points close to the origin in transverse phase space, and having small $\hat{\delta}$, the shear is negligible and the distribution rotates undistorted, as if rigid. To take advantage of this, FIG 7.3 is a snapshot (of the i'th turn) from a frame of reference rotating at rate μ_0. The effect of kick Δp is to change the initial phase space location of point $\mathbf{P}$ to (approximately)

$$R' = R + \Delta R = R + \Delta p \sin\Phi', \quad \Phi' = \Phi + \Delta\Phi = \Phi + \frac{\Delta p \cos\Phi'}{R}. \tag{7.2}$$

After the kick, the particle tune is $\mu_0 + \Delta\mu(R', \hat{\delta})$, and its positions on subsequent turns are indicated by short arrows in FIG 7.3. After i turns its coordinates are

$$\begin{pmatrix} x_i(\Delta p, R, \Phi, \hat{\delta}) \\ p_i(\Delta p, R, \Phi, \hat{\delta}) \end{pmatrix} = R' \cos\Delta\mu\, i \begin{pmatrix} \cos\Phi' \\ \sin\Phi' \end{pmatrix} + R' \sin\Delta\mu\, i \begin{pmatrix} -\sin\Phi' \\ \cos\Phi' \end{pmatrix}. \tag{7.3}$$

The centroid coordinates are given then by

$$\begin{pmatrix} \overline{x_i}(\Delta p) \\ \overline{p_i}(\Delta p) \end{pmatrix} = \int_0^\infty dR \int_0^{2\pi} d\Phi \int_0^\infty d\hat{\delta} \begin{pmatrix} x_i(\Delta p, R, \Phi, \hat{\delta}) \\ p_i(\Delta p, R, \Phi, \hat{\delta}) \end{pmatrix} P_{R,\Phi}(R,\Phi) P_{\hat{\delta}}(\hat{\delta}). \tag{7.4}$$

These formulas are impractical for calculation because of the complicated dependence of Φ' on position $\mathbf{P}$. Since the trigonometry of FIG 7.3 breaks down near the origin, we assume

$$\Delta p << R. \tag{7.5}$$

Even with assumption (7.5), it is not legitimate to approximate Φ' by Φ. If this approximation *is* made, Eqs. (7.3) and (7.4) give a seriously incorrect answer even for $\Delta\mu = 0$ and $i = 0$. This failure is at least partly due to the extravagance of not taking advantage of the strong tendency for cancellation in pairs of particles symmetric about the origin.

To take advantage of this cancellation, we reformulate the calculation by following instead the evolution of *deviations* from the unperturbed distributions as suggested by FIG 7.4. (For the time being we suppress indications of δ dependency from the formulas, since they will be easily restored later.) Since volumes in the plot on the left correspond to probabilities, the units along the vertical axis are length^{-2} and the total "volume" is 1. Planning eventually to apply kick Δp to the beam, the unkicked beam distribution can be re-expressed in terms of the particular deflection Δp that will be applied.

The volume shown in FIG 7.4 can, on the one hand, be visualized as nested "collars" of inner radius $R - \Delta p/2$, wall thickness Δp. The height of a collar is constant and can be evaluated along the x-axis to be $P_x(0)P_p(R)$. On the other hand, the volume can be visualized as the pile of stacked disks shown in the figure, with radius $R + \Delta p/2$ and

$$\text{disk thickness} = P_x(0)\Big(-\frac{dP_p(p)}{dp}\Big)_{p=R}\Delta p = \frac{R}{2\pi\sigma^4}e^{-\frac{R^2}{2\sigma^2}}\Delta p = \frac{P_R(R)\Delta p}{2\pi\sigma^2}. \quad (7.6)$$

When the beam is displaced by Δp along the p axis most of the probability in any particular one of the stacked disks, for example the one with radius R, can be regarded as unchanged; the entire change can be ascribed to an increase in probability density in the positive-p "lune" shown on the right in FIG 7.4 and a corresponding reduction in the negative-p lune. (Though the latter probability density is negative the total probability density in the region remains positive.) Since the entire deviation in this region comes from this particular disk and is accounted for by these lunes, and the subsequent shearing motion respects ring boundaries, it is sufficient to work out the subsequent evolution on a ring-by-ring basis. From these distributions the ring centroids will then be found and finally the overall centroid location.

Toward this end the lune (two dimensional) density can be squashed into an angular (one dimensional) distribution. Furthermore the negative lune can be dropped, compensating by doubling the positive-lune probability. With the area of one lune being $2R\Delta p$, the volume it represents is equal to the lune area times the disk thickness. Including both lunes, this volume is equal to $4R(\Delta p)^2P_R(R)/(2\pi\sigma^2)$. Since volumes represent probabilities, this quantity will be referred to as "deviation probability". Even though not normalized, this volume can be used for centroid calculations as if it is a normalized probability distribution, since the extra volume needed to yield unit total probability contributes no centroid shift.

Letting $P_R^{\rm dev}dR$ stand for the deviation probability in range dR we have

$$P_R^{\rm dev}(R) = \frac{2\Delta p}{\pi\sigma^2}RP_R(R). \quad (7.7)$$

Because the shearing motion preserves the radius in phase space, $P_R^{\rm dev}$ is independent of turn index i. When distributed in x, the just-kicked deviation probability $P_R^{\rm dev}(R)dR$ is uniform. Therefore, when distributed in Φ, which is related to x by $x = R\cos\Phi$, the distribution is proportional to $dx/d\Phi = R\sin\Phi$. We therefore define a (normalized) angular probability distribution,

$$P_{\Phi,0}(\Phi) = \begin{cases} 0 & \text{for } \Phi < 0 \\ (1/2)\sin\Phi & \text{for } 0 < \Phi < \pi \ . \\ 0 & \text{for } \pi < \Phi \end{cases} \tag{7.8}$$

This is a universal initial angular distribution, independent of R.

Then the joint probability distribution $P^{\rm dev}_{R,\Phi}$, defined so that $P^{\rm dev}_{R,\Phi} dRd\Phi$ stands for the deviation probability in range $dRd\Phi$, can be factorized

$$P^{\rm dev}_{R,\Phi}(R,\Phi,i) = P^{\rm dev}_{R}(R) P_\Phi(\Phi,R,i). \tag{7.9}$$

Initially it is given by

$$P^{\rm dev}_{R,\Phi}(R,\Phi,i=0) = P^{\rm dev}_{R}(R) P_\Phi(\Phi,R,i=0) = \frac{\Delta p}{\pi\sigma^2} R P_R(R)\sin\Phi. \tag{7.10}$$

Except for the eventual integration over R, all that is required is to evaluate angular distribution $P_\Phi(\Phi,R,i)$ as it evolves away from $P_{\Phi,0}(\Phi)$—a one dimensional calculation. Furthermore the R dependence allowed for notationally by the second argument of $P_\Phi(\Phi,R,i)$, will be present only to the extent the betatron motion is nonlinear.

The centroid coordinates are obtained as the averages of $x = R\cos\Phi$ and $p = R\sin\Phi$ weighted by $P^{\rm dev}_{R,\Phi}(R,\Phi,i)$;

$$\begin{pmatrix} \overline{x_i}(\Delta p) \\ \overline{p_i}(\Delta p) \end{pmatrix} = \int_0^\infty R\, P^{\rm dev}_{R}(R) dR \int d\Phi\ P_\Phi(\Phi,R,i) \begin{pmatrix} \cos\Phi \\ \sin\Phi \end{pmatrix}. \tag{7.11}$$

Here the limits of the Φ integration are not indicated. They can safely be set large since, for finite i, the integrand vanishes exactly outside a finite range. At $i = 0$ the non-vanishing range is from 0 to π and for other values of i the range needs to be extended only by $\Delta\mu_{\rm max} i$ where $\Delta\mu_{\rm max}$ is the maximum possible tune deviation from nominal.

To check for consistency, let us calculate the $i = 0$ centroid location;

$$\overline{p_0}(\Delta p) = \int_0^\infty RdR \int_0^\pi d\Phi\ P^{\rm dev}_{R,\Phi}(R,\Phi,i=0)\sin\Phi = \Delta p \frac{4}{\pi}\int_0^\pi \frac{\sin^2\Phi}{2} d\Phi = \Delta p, \tag{7.12}$$

as expected.

The only dependence on i in Eq. (7.11) is introduced via

$$\Phi_i = \Phi_0 + \Delta\mu(R,\hat\delta)\, i, \tag{7.13}$$

which, for a particle with initial phase Φ_0, gives its phase after i turns. Any contribution to $\Delta\mu(R,\hat\delta)$ that is independent of R and $\hat\delta$ causes no shearing and has been subsumed in Φ_0. The leading dependence of $\Delta\mu$ is then given by

$$\Delta\mu(R,\hat\delta) = r_1 R + r_2 R^2 + \cdots + d_1\hat\delta + d_2\hat\delta^2 + \cdots\ . \tag{7.14}$$

After i turns the distribution originally given by $P_{\Phi,0}$, having precessed through angle $\Delta\mu(R,\hat\delta)\, i$, will have become $P_{\Phi,i}(\Phi,R,\hat\delta) = (1/2)\sin(\Phi - \Delta\mu(R,\hat\delta)\, i)$, (and zero outside the central lobe.) This, along with Eq. (7.11), is exact in the small kick limit where approximation (7.5) is valid, and the dependence is simple enough for easy and accurate numerical evaluation. But, because of various other uncertainties, great precision is rarely justified. This makes it sensible to approximate the angular distribution in a way that will simplify subsequent calculations. Also we take the opportunity to introduce a more convenient azimuthal angle Θ in terms of which the starting distribution is symmetric about $\Theta = 0$;

$$\Theta = \Phi - \frac{\pi}{2}. \tag{7.15}$$

The approximate form to be used is

$$P_{\Theta,i}(\Theta, R, \hat{\delta}) \approx \frac{1}{\sqrt{2\pi}\sigma_{\rm fit}} \exp(-\frac{(\Theta - \Delta\mu(R,\hat{\delta})\, i)^2}{2\sigma_{\rm fit}^2}). \tag{7.16}$$

This form eliminates the need for the multiple cases of Eq. (7.8) and permits an infinite Φ integration range. The quantity $\sigma_{\rm fit}$ is simply a dimensionless number (an angle in radians) chosen to make the approximation in Eq. (7.16) as accurate as possible. The value $\sigma_{\rm fit} = (2\pi)^{-1/6} = 0.736$ would match the quadratic variation at $\Phi = \pi/2$, but we choose instead

$$\exp\Big(- \frac{\sigma_{fit}}{2}\Big) = \frac{\pi}{4}, \quad \text{or} \quad \sigma_{\rm fit} = 0.695, \tag{7.17}$$

which causes Eq. (7.12) to be satisfied, thereby avoiding a (small but inelegant) error in the just-kicked centroid location. Substituting Eq. (7.16) into Eq. (7.11) yields

$$\begin{pmatrix} \overline{x_i}(\Delta p, \hat{\delta}) \\ \overline{p_i}(\Delta p, \hat{\delta}) \end{pmatrix} = \Delta p \int_0^\infty dR \frac{2}{\pi\sigma^4} R^3 e^{-\frac{R^2}{2\sigma^2}} \int_{-\infty}^{\infty} d\Theta\ \frac{1}{\sqrt{2\pi}\sigma_{\rm fit}} e^{-\frac{(\Theta - \Delta\mu(R,\hat{\delta})\, i)^2}{2\sigma_{\rm fit}^2}} \begin{pmatrix} -\sin\Theta \\ \cos\Theta \end{pmatrix} \tag{7.18}$$

$$= \ \Delta p \,\frac{2}{\pi\sigma^4}\, \frac{1}{\sqrt{2\pi}\sigma_{\rm fit}} \int_0^\infty dR R^3 e^{-\frac{R^2}{2\sigma^2}} \begin{pmatrix} -\sin\Delta\mu\, i \\ \cos\Delta\mu\, i \end{pmatrix} \int_{-\infty}^{\infty} d\Theta\ \cos\Theta\, e^{-\frac{\Theta^2}{2\sigma_{\rm fit}^2}} \tag{7.19}$$

$$= \ \Delta p \,\frac{1}{2\sigma^4} \int_0^\infty dR R^3 e^{-\frac{R^2}{2\sigma^2}} \begin{pmatrix} -\sin\Delta\mu(R,\hat{\delta})\, i \\ \cos\Delta\mu(R,\hat{\delta})\, i \end{pmatrix} \tag{7.20}$$

where the dependence on $\hat{\delta}$ has been restored to the notation. This formula, with $\Delta\mu(R,\hat{\delta})$ expressed, for example, as in Eq. (7.14), is the main formula describing the effect of decoherence due to R-dependent tune caused by nonlinear betatron motion. For small i, evaluating the integral numerically is easy. For large i, the method of stationary phase may be applicable.[**19**]

Since there has been no averaging over $\hat{\delta}$ as yet, Eq. (7.20) should also be valid with $\hat{\delta}$ replaced by δ. The major effect of this would be evident in FIG 7.3 where the phasor amplitudes would vary sinusoidally because of chromaticity and synchrotron oscillation. Whatever shearing this causes is exactly undone over a complete longitudinal cycle, causing periodic decoherence/recoherence each period of synchrotron oscillation. By performing these calculations it would be possible to compare to a formula due to Meller *et al.*[**20**] but this has not been done. The feature distinguishing the present calculation from Meller's is that he assumed no systematic dependence of tune on $\hat{\delta}$. It is not easy to compare formulas here with his paper since the order of integation is different and he does not make the approximation Eq. (7.16). (This should cause only small numerical differences.)

We continue, but now keeping just the term $\Delta\mu(R,\hat{\delta}) = d_1\hat{\delta}$ (which permits the R integration to be performed) and assume that $\hat{\delta}$ is distributed according to

$$P_{\hat{\delta}}(\hat{\delta}) = \frac{\hat{\delta}}{\sigma_{\hat{\delta}}^2} \exp\Big(- \frac{\hat{\delta}^2}{2\sigma_{\hat{\delta}}^2}\Big), \tag{7.21}$$

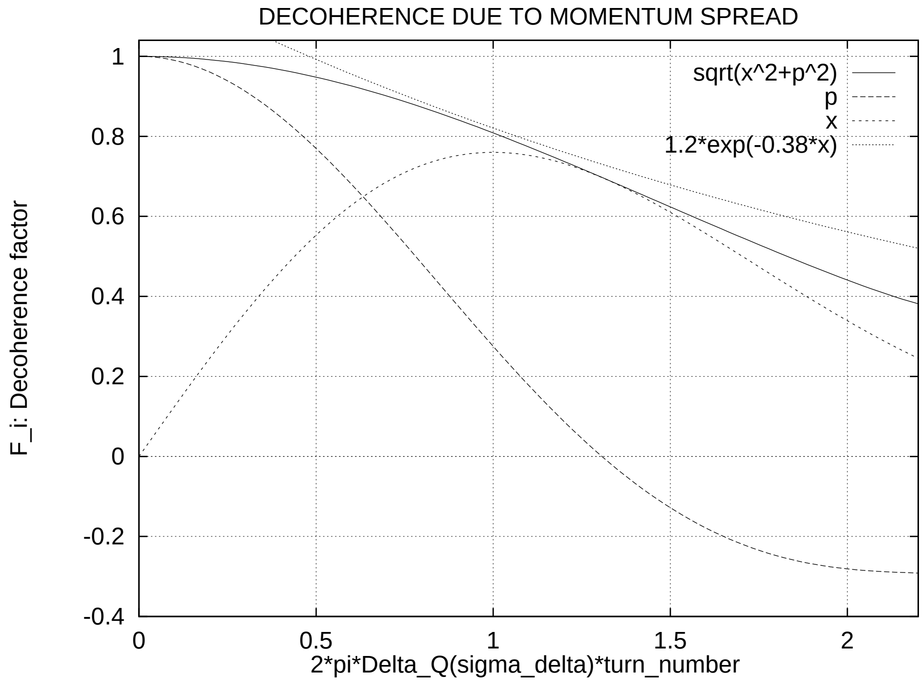


FIGURE 7.5. Time evolution of (fractional) centroid position $x/\Delta p$, slope $p/\Delta p$, and $\sqrt{x^2+p^2}/\Delta p$ after initial deflection Δp, viewed in a frame of reference rotating at the small amplitude tune, as given by Eqs. (7.23-7.25). An exponentially decaying function $1.2\exp(-0.38 d_1\sigma_\delta i)$ is also shown for comparison.

and average over $\hat\delta$ to obtain[1]

$$\frac{\overline{\overline{p_i}}(\Delta p)}{\Delta p} = \frac{1}{\sigma_\delta^2}\int_0^\infty d\hat\delta\ \hat\delta\ \exp\Big(-\frac{\hat\delta^2}{2\sigma_\delta^2}\Big)\cos(d_1\hat\delta\, i) \tag{7.22}$$

$$= 1-(d_1\sigma_\delta i)^2+\frac{1}{3}(d_1\sigma_\delta i)^4-\frac{1}{7.5.3}(d_1\sigma_\delta i)^6+\frac{1}{9.7.5.3}(d_1\sigma_\delta i)^8+\cdots, \tag{7.23}$$

and

$$\frac{\overline{\overline{x_i}}(\Delta p)}{\Delta p} = -\frac{1}{\sigma_\delta^2}\int_0^\infty d\hat\delta\,\hat\delta\ \exp\Big(-\frac{\hat\delta^2}{2\sigma_\delta^2}\Big)\sin(d_1\hat\delta)\,i = -\sqrt{\frac{\pi}{2}(d_1\sigma_\delta i)}\ \exp\Big(-\frac{(d_1\sigma_\delta i)^2}{2}\Big). \tag{7.24}$$

The most directly observable quantity is the "decoherence factor",

$$F_i(d_1\sigma_\delta i) = \sqrt{\Big(\frac{\overline{\overline{x_i}}(\Delta p)}{\Delta p}\Big)^2+\Big(\frac{\overline{\overline{x_i}}(\Delta p)}{\Delta p}\Big)^2}. \tag{7.25}$$

These functions are plotted in FIG 7.5. As explained previously, the quantities $\overline{\overline{x_i}}$ and $\overline{\overline{p_i}}$ tend to vary slowly because they refer to a frame of reference rotating

[1] Formula (7.23 is poorly convergent and can only be used for values of the argument less than 2 or so.

at rate μ_0. The corresponding invariant amplitude $\sqrt{\overline{\overline{x_i}}^2 + \overline{\overline{p_i}}^2}$ varies even more slowly. But when it is viewed as a vector in a stationary frame it rotates rapidly and is interpreted as the betatron oscillation of the centroid. The magnitude of the invariant amplitude is the same in stationary and rotating frames. This was the basis for the statement made above that the "decoherence factor" F_i is the theoretical quantity that can most easily be correlated with experimental observations.[2] From FIG 7.5 it can be seen that the time evolution of $\overline{\overline{x_i}}$ and $\overline{\overline{p_i}}$ are very different, the latter falls off in more or less Gaussian fashion while the former rises initially, then falls. Neither of these behaviors seems deserving of the name "damping", but the function F_i falls off more nearly as the decaying exponential that is normally associated with damping. To illustrate this point a pure exponential decay curve that crudely matches F_i is also shown in FIG 7.5.

When damping rates are measured experimentally in the control room, the observed response is usually not a pure exponential decay. Rather, an initial transient (that is hard to interpret and may be instrumental in nature) is followed by a curve well fit by a pure exponential. An empirical recipe extracting damping rates has been to select the range over which the log plot is most nearly linear as the signal falls by $1/e$—typically this is from about 0.8 to about 0.3 of the just-kicked signal. This is not very different from the range over which the exponential described in the previous paragraph gives a tolerable fit to the theoretical response curve. Considering the only-semi-quantitave "absolute" accuracy of the measurements and the lack of accuracy with which the various parameters influencing the phenomenon are known, we therefore judge the exponential fit shown in the graph to be a reasonable representation of the theory, for comparison with the experimental data.

We have obtained a simple prescription for predicting the "damping rate" $\alpha_{\sigma,\mathrm{dec}}$ with which the centroid will be observed to damp after the beam has been pinged. By particle tracking in the lattice under consideration (CESR in our case with parameters corresponding to FIG 7.1), for a particle with invariant longitudinal invariant equal to the r.m.s. value $\hat{\delta} = \sigma_\delta = 0.6 \times 10^{-3}$, the tune shift is found to be $\Delta Q(\sigma_\delta) \approx 1.5 \times 10^{-3}$. Assuming the dependence linear, this fixes the d_1 coefficient in Eq. (7.14);

$$d_1 = \frac{2\pi \Delta Q(\sigma_\delta)}{\sigma_\delta}. \tag{7.26}$$

Then the centroid amplitude varies as $\exp(-\alpha_{\sigma,\mathrm{dec}} t) = \exp(-0.38 \times 2\pi \times |\Delta Q(\sigma_\delta)| i)$, where time t and turn number i are related by $t = i/f_0$, with $f_0 = 0.39 \times 10^6\,\mathrm{Hz}$. Then we obtain

$$\alpha_{\sigma,\mathrm{dec}} = 2.39 |\Delta Q(\sigma_\delta)| f_0 \tag{7.27}$$

When parameters appropriate to FIG 7.1 are used, the predicted damping rate is $1160\,\mathrm{s}^{-1}$. This exceeds the rate inferred from the multiparticle simulation by a factor 1.7. Considering the various uncertainties, this is probably as good agreement as can be expected.

[2] Though more detailed information about the beam is measurable in principle, we are mainly concerned with signals from the beam position monitors which contain only information about the centroid.

7.4. Simulation of Decoherence/Recoherence Echos

More than one of the decoherence phenomena mentioned above may be operative at the same time. This is illustrated in FIG 7.6, which was obtained using the `decoherence` simulator on the `racetrack` lattice. Initially there is decoherence in both horizontal and vertical motion due to the spread of tunes. That is, the centroid motion "damps". As discussed earlier, the centroid amplitude is expected to recohere to give an echo after one complete synchrotron period and to repeat this cycle at the synchrotron frequency. This is visible in both x and y motion. From the graphs it is clear that the situation is a bit more complicated than this expectation. Especially the $< x >$ centroid does not return to its full starting amplitude. This presumably indicates the presence of (betatron)-amplitude dependence of the transverse tunes. These phenomena are to be investigated in assignments given below.

Two views of the GUI used for this simulation are exhibited in FIG 7.7 and FIG 7.8. As shown in FIG 7.7 the requested chromaticities are $Q'_x = 20$ and $Q'_y = -20$ in the basic lattice description. Modified entries can be typed into those two slots. From the `racetrack.sxf` file one can infer that the names of the chromaticity sextupoles are `sext1` and `sext2`. These names are to be typed into the slots labeled `b2f` and `b2d`. Clicking on the setup button adjusts the strengths of sextupoles with those names to achieve the desired chromaticities. Successful completion of this process is printed.

For simulating decoherence a bunch of, say, 1000 particles is initialized, then kicked transversely by the `kicker` element, and then tracked for, say, 1000 turns. All three centroids $< x >$, $< y >$, and $< s >$ are plotted every 10 turns. These values are plotted in FIG 7.6.

The detailed simulation is controlled by the APDF file. Two such files are shown in Table 7.1. How the kick is administered is governed by the line

```
<link algorithm="UAL::USPAS::OneTurnKicker" elements="kicker" />
```

which is common to both APDF files. Propagation around the ring can be done in element-by-element, kick code, fashion. This calculation is governed by the `teapot.apdf` file. Alternatively the tracking can proceed by truncated power series (TPS) tracking. This calculation is governed by the `mapping.apdf` file. For this propagation method the maps from every `bpm` to its adjacent `bpm` are first calculated, and then the map is used to evolve individual particles. Not shown in the APDF file (for now) is the truncation order, which was 3, "octupole order", for the simulation shown in FIG 7.6.

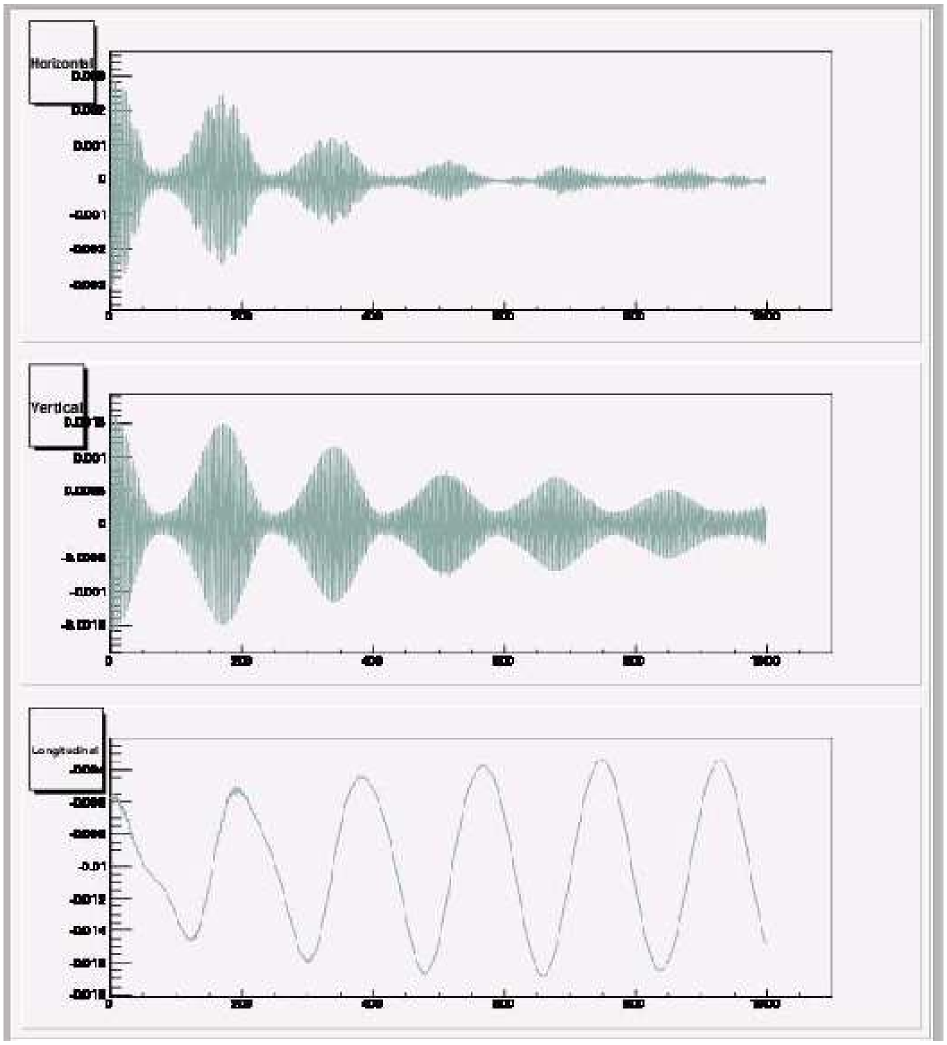
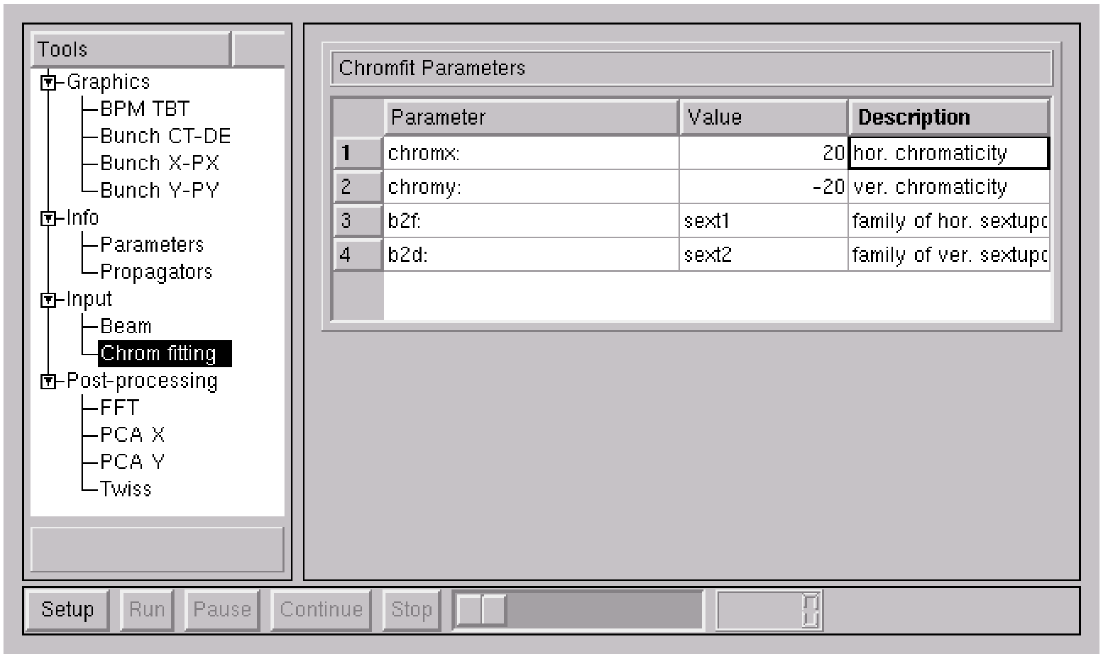


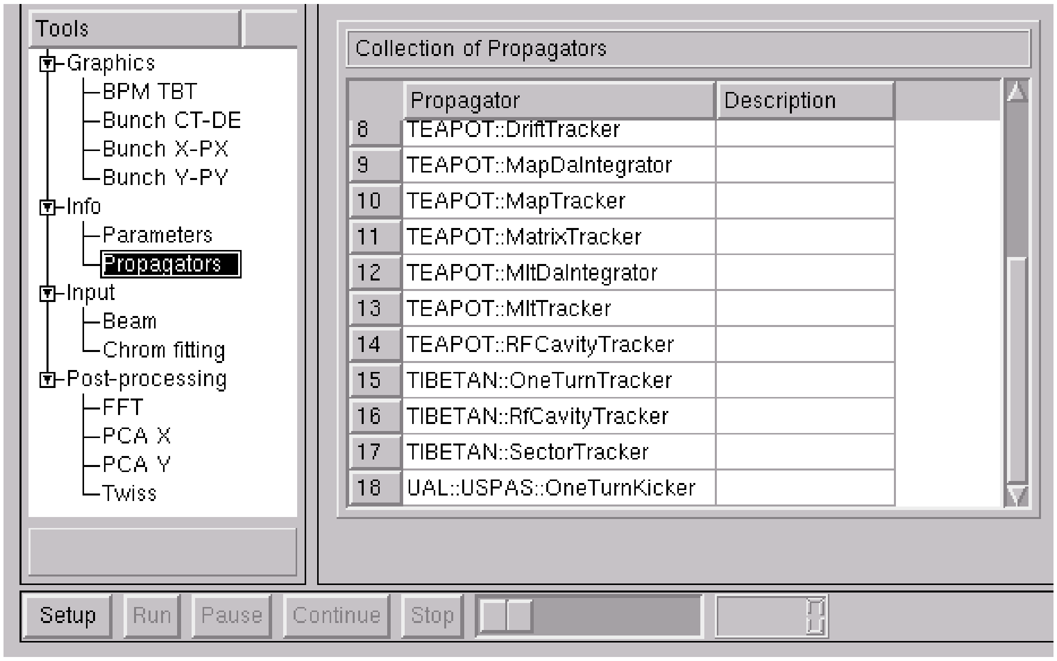
FIGURE 7.6. Beam echos observed one synchroton period (and multiples thereof) after a beam is kicked both horizontally and vertically in the `racetrack` lattice. $Q'_x = 20$, $Q'_y = -20$.



Chromfit Parameters

	Parameter	Value	Description
1	chromx:	20	hor. chromaticity
2	chromy:	-20	ver. chromaticity
3	b2f:	sext1	family of hor. sextupc
4	b2d:	sext2	family of ver. sextupc

FIGURE 7.7. Simulator window used for adjusting chromaticities.



Collection of Propagators

	Propagator	Description
8	TEAPOT::DriftTracker	
9	TEAPOT::MapDaIntegrator	
10	TEAPOT::MapTracker	
11	TEAPOT::MatrixTracker	
12	TEAPOT::MltDaIntegrator	
13	TEAPOT::MltTracker	
14	TEAPOT::RFCavityTracker	
15	TIBETAN::OneTurnTracker	
16	TIBETAN::RfCavityTracker	
17	TIBETAN::SectorTracker	
18	UAL::USPAS::OneTurnKicker	

FIGURE 7.8. Simulator window listing available propagators.

Table 7.1. Two APDF files for simulating the same decoherence/recoherence phenomenon by two different methods—element-by-element, and map tracking.

```
<apdf>
  <propagator id="teapot" accelerator="ring">
     <create>
        <link algorithm="TEAPOT::DriftTracker"     types="Default" />
        <link algorithm="TEAPOT::DriftTracker"     types="Marker|Drift" />
        <link algorithm="TEAPOT::DipoleTracker"    types="Sbend" />
        <link algorithm="TEAPOT::MltTracker"       types="Quadrupole|Sextupole|Multipole|[VH]kicker"/>
        <link algorithm="TIBETAN::RfCavityTracker" types="RfCavity"/>
        <link algorithm="AIM::Monitor"             types="Monitor|[VH]monitor" />
        <link algorithm="UAL::USPAS::OneTurnKicker" types="Kicker" />
     </create>
  </propagator>
</apdf>

<apdf>
  <propagator id="mapping" accelerator="ring">
     <create>
        <link algorithm="TEAPOT::MapTracker"     sector="Default" />
        <link algorithm="AIM::Monitor"           elements="bpm" />
        <link algorithm="UAL::USPAS::OneTurnKicker" elements="kicker" />
     </create>
  </propagator>
</apdf>
```

SIMULATION 7.1. *Use the* `decoherence` *simulator to correlate decoherence time with momentum-dependent tune spread. One can adjust the tune spreads by adjusting the chromaticities. One can plot the tunes versus δ by preparing initial particle conditions for various values of δ.*

SIMULATION 7.2. *By vastly reducing the number of* `bpm`*'s in the* `racetrack` *lattice, speed up the decoherence/recoherence simulation that uses map tracking.*

SIMULATION 7.3. *Investigate the dependence of decoherence time on initial kick amplitude. Since the kick amplitude is not available from the GUI it is necessary to "hard code" the kick amplitude and recomplile that part of the code.*

CHAPTER 8

General Transverse Motion

A linearized treatment of transverse particle dynamics was given in Chapter 2. Here, starting from a more general formulation of the equation of motion, we include the possibility of nonlinear deflections. The next few sections spell out the bending effects of the simpler magnetic elements. Then the generalization from transfer matrices to transfer maps, in order to include nonlinear effects, is discussed. The following sections discuss symplecticity-preserving evolution algorithms. Finally the discussion of FFT beam diagnosis, started in Chapter 4, is extended to nonlinear effects.

8.1. Magnetic Deflections

The magnetic field of an ideal, erect, accelerator magnet can be expressed as

$$\mathbf{B} = B_y \begin{pmatrix} 0 \\ 1 \end{pmatrix} + \frac{\partial B_y}{\partial x} \begin{pmatrix} y \\ x \end{pmatrix} + \frac{1}{2}\frac{\partial^2 B_y}{\partial x^2} \begin{pmatrix} 2xy \\ x^2 - y^2 \end{pmatrix} + \cdots . \tag{8.1}$$

Though not indicated by the notation, the (constant) coefficients, B_y and its derivatives, are evaluated on the magnet centerline, in the interior of the magnet. Only x and y components are shown, since the fields are assumed to be transverse. Also the fields are assumed to be independent of z (except for vanishing outside the magnet). One can confirm immediately that $\nabla \cdot \mathbf{B} = 0$ and that

$$\nabla \times \mathbf{B} = \begin{pmatrix} \hat{\mathbf{x}} & \hat{\mathbf{y}} & \hat{\mathbf{s}} \\ \partial/\partial x & \partial/\partial y & 0 \\ B_x & B_y & 0 \end{pmatrix} = 0. \tag{8.2}$$

In fact the form (8.1) amounts to starting with a field for which only B_y is nonvanishing for $x = 0$, but is otherwise arbitrary, expanding it in powers of x, and then extrapolating off the $x = 0$ plane using the Maxwell equations to obtain the y dependendence and B_x.

Using the Lorentz force law, the change in velocity $d\mathbf{v}$, as charge Q passes longitudinal distance ds, is given by

$$\frac{m\gamma}{Q}\frac{d\mathbf{v}}{ds/v} = \mathbf{v} \times \mathbf{B} =$$

$$B_y \begin{pmatrix} \hat{\mathbf{x}} & \hat{\mathbf{y}} & \hat{\mathbf{s}} \\ v_x & v_y & v_s \\ 0 & 1 & 0 \end{pmatrix} + \frac{\partial B_y}{\partial x} \begin{pmatrix} \hat{\mathbf{x}} & \hat{\mathbf{y}} & \hat{\mathbf{s}} \\ v_x & v_y & v_s \\ y & x & 0 \end{pmatrix} + \frac{1}{2}\frac{\partial^2 B_y}{\partial x^2} \begin{pmatrix} \hat{\mathbf{x}} & \hat{\mathbf{y}} & \hat{\mathbf{s}} \\ v_x & v_y & v_s \\ 2xy & x^2 - y^2 & 0 \end{pmatrix} + \cdots . \tag{8.3}$$

To treat such a magnet as a single "short" element, while allowing for fringe fields, it is necessary to integrate this formula from well before the magnet to well after it. The effect of passage of a particle through such a magnet are "kinks", i.e.

discontinuities $\Delta x'$ and $\Delta y'$ in the slope coordinates $x' = dx/ds$ and $y' = dy/ds$. For a dipole (the first term) the bend angle $\Delta\theta$ (assumed small) is determined by

$$\Delta\theta \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} \Delta x' \\ \Delta y' \end{pmatrix} = \begin{pmatrix} \Delta(v_x/v_s) \\ \Delta(v_y/v_s) \end{pmatrix} = \int ds \, \frac{cB_y}{pc/Q} \begin{pmatrix} -x \\ y \end{pmatrix}, \tag{8.4}$$

with the result that

$$\Delta\theta = \int ds \, \frac{cB_y}{pc/Q}. \tag{8.5}$$

This has neglected any variation of v_s, x, or y as the particle passes the magnet. From the derivation this result might seem to provide only a rough approximation to the deflection. But, in fact, a *thick* magnet can be segmented into arbitrarily many *thin* magnets. So this formulation can be the basis for accurate numerical integration of particle orbits through (ideal) magnets. Because B_y and pc vary proportionally, $\Delta\theta$ has the desired property of being momentum-independent. Furthermore $\Delta\theta$ is dimensionless as an angle must be.

For a quadrupole (the second term of Eq. (8.3)), the strength (i.e. inverse focal length) q is determined by

$$q \begin{pmatrix} -x \\ y \end{pmatrix} = \begin{pmatrix} \Delta x' \\ \Delta y' \end{pmatrix} = \begin{pmatrix} \Delta(v_x/v_s) \\ \Delta(v_y/v_s) \end{pmatrix} = \int ds \, \frac{c\partial B_y/\partial x}{pc/Q} \begin{pmatrix} -x \\ y \end{pmatrix}, \tag{8.6}$$

with the result that

$$q = \int ds \, \frac{c\partial B_y/\partial x}{pc/Q}. \tag{8.7}$$

Like $\Delta\theta$, q is a purely geometric quantity, an inverse length. It is independent of particle momentum and charge which cancel against B_y. The inevitability of opposite-sign focal properties in the two transverse planes has also been exhibited.

For an erect sextupole (the third term of Eq. (8.3) the strength S is determined by

$$\frac{1}{2} S \begin{pmatrix} x^2 - y^2 \\ 2xy \end{pmatrix} = \begin{pmatrix} \Delta x' \\ \Delta y' \end{pmatrix} = \frac{1}{2} \int ds \, \frac{c\partial^2 B_y/\partial x^2}{pc/Q} \begin{pmatrix} x^2 - y^2 \\ 2xy \end{pmatrix}, \tag{8.8}$$

with the result that

$$S = \int ds \, \frac{c\partial^2 B_y/\partial x^2}{pc/Q}. \tag{8.9}$$

Expressions like $\Delta\theta$, q, and S are known as “field integrals” because integration over a complete element is implied. Higher pure multipole magnets are defined similarly. A potential source of error to be aware of comes from the factor $1/n!$ that enters the relation between multipole strength and $\partial B^n/\partial x^n$. This factor comes from the Taylor series expansion formula. For the sextupole the factor is $1/2$.

In spite of the fact that such magnet fields violate Maxwell’s equations, it is customary to treat the fields of ideal magnets as uniform within an effective or “magnetic” length l_m and dropping discontinuously to zero outside that length.[1] In this convention the field B_y and its derivatives are evaluated at the longitudinal center of the magnet and l_m is adjusted to match the field integral over the magnet. For this reason l_m tends to be close to, but not exactly equal to the geometric length l. Repeating the discussion of thin elements made earlier, though these magnetic strength definitions have the appearance of being crude approximations, they can be made increasingly accurate by reducing l_m. (This accuracy is vitiated by the

[1] Within ADXF the notation for magnetic length l_m is `ml` which is distinct from the “geometric length” l, notation `l`. But, by default, `ml` and `l` are taken to be equal.

inevitable presence of longitudinal field components, present when transverse field components depend on s. But in most cases the deflections caused by B_s are extremely small.)

In this approximation the above formulas become,

$$\Delta\theta = \frac{cB_y}{pc/Q}\,l_m, \quad q = \frac{c\partial B_y/\partial x}{pc/Q}\,l_m, \quad S = \frac{c\partial^2 B_y/\partial x^2}{pc/Q}\,l_m, \quad \cdots . \tag{8.10}$$

8.2. ADXF and SIF Element Strengths and Deviations

Within "standard input format" SIF, which has evolved into MAD input format, the strengths of magnetic elements can be expressed using the formulas just derived. In particular, for quadrupoles and sextupoles, the leading factors in Eqs. (8.10) are expressed as

$$\mathtt{K1} = \frac{1}{B\rho}\,\frac{\partial B_y}{\partial x} \equiv \frac{c\partial B_y/\partial x}{pc/Q}, \quad \mathtt{K2} = \frac{1}{B\rho}\,\frac{\partial^2 B_y}{\partial x^2} \equiv \frac{c\partial^2 B_y/\partial x^2}{pc/Q}, \tag{8.11}$$

where the conventional "B-rho" quantity is defined by either of these equations or, better motivated, by the first of formulas (8.10):

$$(B\rho) \equiv B_y \frac{l_m}{\Delta\theta} = \frac{pc/Q}{c}, \tag{8.12}$$

where, as usual, rather than canceling factors, it is convenient to retain the combination pc/Q which has S.I. units, namely volts.

As already implied, the merit in grouping factors into $B\rho$ is to reduce strength parameters to purely geometric terms. The dimensions of `K1` and `K2` are m^{-2} and m^{-3} respectively. These dimensions are appropriate for longitudinally variable "thick element description" for which integration over the magnetic element remains to be performed. Generally speaking, element strengths is SIF are expressed as local strength functions, like `K1` and `K2`. The convention in ADXF's generic element <element> is different; strengths are represented by integrated (over length) values of local strength functions. Examples q and S were given earlier. As has been mentioned before, this distinction is purely cosmetic, and is unrelated to actual precision of representation. Actually, for backward compatibility, ADXF also supports a MAD-like syntax which, other than being expressed in XML is equivalent to MAD8.

The main way in which ADXF extends MAD8 is that pre-existing elements can be assigned new attributes such as magnetic field deviations. A simple example file showing the syntax for including deviations is **`eq_tune_fodo.adxf`**. In ADXF a distinction is drawn between *uninstalled elements* (in Etienne Forest's PTC terminology they are called "on-the-bench elements") and *installed elements.* The latter are, of necessity, fully-instantiated, meaning they have their own individual names, positioning, and strength deviations. The strengths shown in the **`eq_tune_fodo.adxf`**, say for **`quadhf`** elements, are *uninstalled* properties, shared by all occurences of **`quadhf`** elements in the lattice. Within ADXF there is an "inheritance" mechanism in which an individual *installed* element, with its own individual name, can be referred to a *design* element, such as **`quadhf`**, from which it inherits parameters. Such an *installed* element can be assigned parameters that override or augment the *uninstalled* parameters.

This can be illustrated by the following code fragment

```
<elements>
  <marker name="mk1"/>
  <sbend name="bend" l="lq" angle="deltheta"/>
  <quadrupole name="quadhv" l="lq" k1="kq1"/>
  ...
</elements>
```

which amounts to re-expressing MAD input language as XML, with an example exhibiting inheritance:

```
<elements>
  <sbend name="d0mp08" l="3.58896" angle="-0.0151186"/>
  <element name="bi8-dh0" design="d0mp08"/>
    <mfield b="0 0 0.005476 0.033503"
            a="0.0 -0.010166 0.024366"/>
  </element>
</elements>
```

where the "bi8-dh0" element is based on the "d0mp08" design element, but with field deviations described within the <mfield> tag which is associated with magnetic field attributes.

In ADXF, as in most lattice descriptions, magnetic field deviations are expressed as *multipole series*. For a bending magnet the magnetic field, as well as the effect of transverse positioning deviations $(\Delta x, \Delta y)$, are expressed as a (complex) series:

$$(B_y + iB_x)l_m = B_0 l_m \sum_{n=0}^{M} (b_n + ia_n)\big((x - \Delta x) + i(y - \Delta y)\big)^n. \tag{8.13}$$

The maximum multipole index M is usually fixed at a moderately large value, such as 10, but with dynamic memory allocation M can be made larger, if desired. When truncated power series (TPS) are used a highest retained power is defined. Multipole terms with index higher than this are simply ignored in TPS calculations. Field expansions for B_x and B_y individually are obtained by separating Eq. (8.13) into real and imaginary parts. The factor l_m has been intentionally left as a common factor on both sides of this equation. Note that this representation includes the possibility of arbitrary roll angles around the longitudinal axis, even including purely vertical bends. The actual factors expressing dipole field deviations in the ADXF file are

$$\tilde{b}_n = B_0 l_m b_n, \quad \text{and} \quad \tilde{a}_n = B_0 l_m a_n. \tag{8.14}$$

In particular, for an ideal thin sector bend, $\tilde{b}_0 = -\Delta\theta$. For an arbitrary sector bend magnet this formula becomes

$$\Delta\theta = -2\sin^{-1}\frac{\tilde{b}_0}{2}. \tag{8.15}$$

Note that, like similar quantities introduced previously, $\tilde{b}_0$ is a "field integral" quantity. Bend fields that depend on longitudinal coordinate s can be represented numerically and accurately by longitudinal segmentation.

The coefficients in multipole series (8.13) can be related to other conventional magnet strength parameters as shown in Table 8.1. Real and imaginary coefficients R_n and I_n are defined by

$$(x + iy)^n = R_n + iI_n. \tag{8.16}$$

TABLE 8.1. Deflections caused by standard magnets and notations for their strengths

	n	R_n	I_n	$\tilde{b}_n$	$\tilde{a}_n$	$\Delta x' = -\tilde{B}_y$	$\Delta y' = \tilde{B}_x$
Horz. bend	0	1	0	$\Delta\theta_x$	0	$-\Delta\theta_x$	0
Vert. bend				0	$\Delta\theta_y$	0	$\Delta\theta_y$
Erect quad	1	x	y	$q = 1/f$	0	$-qx$	qy
Skew quad				0	$q_s = 1/f_s$	$q_s y$	$q_s x$
Erect sext	2	$x^2 - y^2$	$2xy$	$S/2$	0	$-\frac{S}{2}(x^2 - y^2)$	$\frac{S}{2}2xy$
Skew sext				0	$S_s/2$	$\frac{S_s}{2}2xy$	$\frac{S_s}{2}(x^2 - y^2)$
Erect oct	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 oct				0	$O_s/6$	$\frac{O_s}{6}(3x^2y - y^3)$	$\frac{O_s}{6}(x^3 - 3xy^2)$
Erect deca	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 deca		$+y^4$		0	$D_s/24$	$\frac{D_s}{24}4xy(x^2 - y^2)$	$\frac{D_s}{24}(x^4 - 6x^2y^2 + y^4)$

The factors 1!, 2!, 3! entering the definitions of quad strength q, sextupole strength S, octupole strength O, *etc.* are conventional. Notice, for example, the relation to SIF syntax;

$$\tilde{b}_2 = \frac{S}{2} = \frac{\mathtt{K2*L}}{2}. \tag{8.17}$$

Formulas giving transverse kinks are

$$\begin{aligned} \Delta x' = &\ -\tilde{B}_y|_n = -\tilde{b}_n R_n + \tilde{a}_n I_n, \\ \Delta y' = &\ \tilde{B}_x|_n = \tilde{b}_n I_n + \tilde{a}_n R_n. \end{aligned} \tag{8.18}$$

Multipole expansions to express field nonuniformity are reasonably standard for dipoles.[2] But magnet types other than dipoles have vanishing fields on axis, which makes it necessary to replace the factor B_0 if an expansion like Eq. (8.13) is still to be used. Otherwise the coefficients, as measures of fractional deviation, can be placed in one-to-one correspondance with dipole field multipoles, but with the indices shifted by one. A common choice for an (approximately) erect quadrupole is to write

$$(l_m B_y^Q) + i(l_m B_x^Q) = (l_m \frac{\partial B_y^Q}{\partial x})\big(x + iy + 10^{-4} \sum_{n=2}^{M} (b_n^Q + i a_n^Q) \frac{(x+iy)^n}{R_r^{n-1}}\big), \tag{8.19}$$

where R_r is a reference radius, such as 1 cm. The value of R_r and the factor 10^{-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 in well-designed magnets.

8.3. Nonlinear Magnetic Field Example

The outline of a typical horizontal steering magnet, viewed from downstream, is shown in FIG 8.1, which also shows the same magnet rotated to steer vertically. The measured, midplane, vertical magnetic field B_y is plotted in FIG 8.2; it has the form

$$B_y(x, y = 0) = B_{y0}(1 + b_2 x^2 + b_4 x^4), \tag{8.20}$$

[2] Actually expansion (8.13) is standard only in America. In Europe, coefficients are defined initially in an expansion of a vector potential which is then differentiated to obtain a series similar to Eq. (8.13) but with factors of $n!$ and indices shifted by 1.

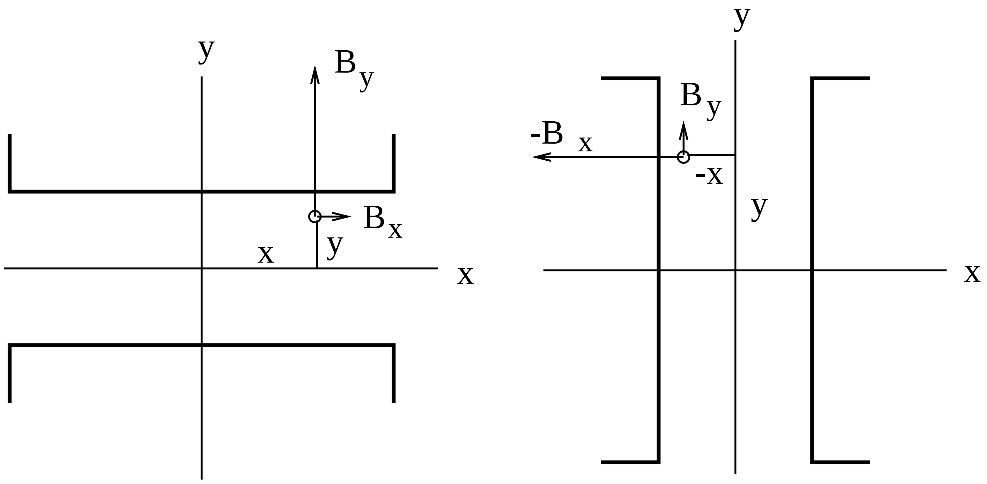


FIGURE 8.1. A horizontal steering dipole and the same magnet rotated so that it steers vertically. The median-plane field and its multipole approximation is shown in FIG 8.2. The analytic description of the rotated fields is given in the text.

where B_{y0} is the nominal value of B_y. For this magnet the dominant multipole imperfection coefficient happens to be decapole b_4. Referring to Table 8.1 it can be seen that this form can be continued off the median plane by

$$B_y(x,y) = B_{y0}(1 + b_2(x^2 - y^2) + b_4(x^4 - 6x^2y^2 + y^4)). \tag{8.21}$$

This also yields the other field component;

$$B_x(x,y) = B_{y0}(b_2 2xy + b_4 4xy(x^2 - y^2)). \tag{8.22}$$

Note that the field uniformity of the actual magnet is somewhat better at small amplitudes than either multipole would give by itself. This illustrates the not uncommon possibility that truncation of the multipole series can, by defeating desirable cancellation, yield overly pessimistic field values. Also plotted in FIG 8.2 is the "wrong field component" $B_x(x, y = 10)$ plotted as a function of x, along a line displaced to positive $y = 10\,\mathrm{mm}$.

If the steering magnet is rotated by 90 degrees (anti-clockwise to an observer looking from downstream, so that positive horizontal deflection becomes positive vertical deflection) the new multipole expansion can be obtained from series (8.21) and (8.22) by transformations suggested by the labels on FIG 8.1,

$$\begin{aligned} -B_x(x,y) &= (-B_{x0})(1 + b_2(y^2 - x^2) + b_4(y^4 - 6y^2x^2 + x^4)), \\ B_y(x,y) &= (-B_{x0})(b_2 2y(-x) + b_4 4y(-x)(y^2 - x^2)), \end{aligned} \tag{8.23}$$

where the nominal field is now B_{x0} with sign opposite to B_{y0}. Hence we have

$$\begin{aligned} B_y(x,y) &= B_{x0}(b_2 2xy - b_4 4xy(x^2 - y^2)), \\ B_x(x,y) &= B_{x0}(1 - b_2(x^2 - y^2) + b_4(x^4 - 6x^2y^2 + y^4)). \end{aligned} \tag{8.24}$$

Renaming the original multipole coefficients $b_{2,\mathrm{old}}$ and $b_{4,\mathrm{old}}$, to make this form match expansion Eq. (8.13) it is necessary to introduce skew coefficients a_2 and a_4

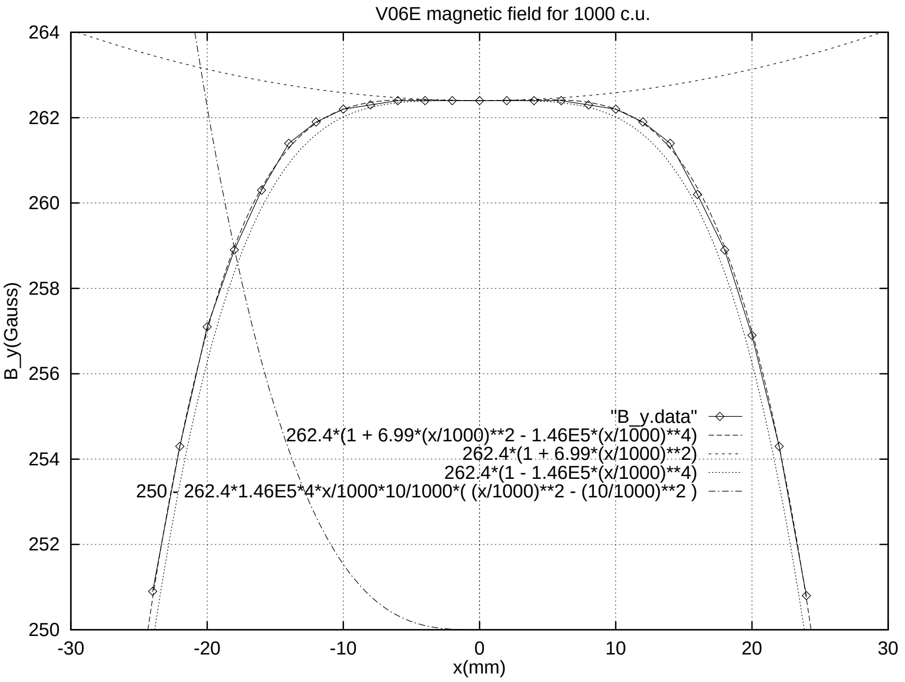


FIGURE 8.2. Midplane magnetic field B_y in a transversely-limited erect dipole magnet. Curves with sextupole or decapole "turned off" are also plotted, as well as a curve showing the decapole contribution to the off-median-plane, "wrong field component" $B_x(x, y = 10\,\text{mm})$ (displaced upwards by 250 Gauss for plotting purposes.). The multipole coefficients are $b_2 = 6.99\,\text{m}^{-2}$ and $b_4 = -1.46 \times 10^5\,\text{m}^{-4}$. When the same magnet is used for vertical steering the non-vanishing coefficients are $a_2 = -6.99\,\text{m}^{-2}$ and $a_4 = -1.46 \times 10^5\,\text{m}^{-4}$. The field calculation is due to Sasha Temnyck.

into Eq. (8.18);

$$\begin{aligned} B_y(x,y) &= B_{x0}(-a_2 2xy - a_4 4xy(x^2 - y^2)), \qquad (8.25)\\ B_x(x,y) &= B_{x0}(1 + a_2(x^2 - y^2) + a_4(x^4 - 6x^2y^2 + y^4)). \end{aligned}$$

If the field of the erect magnet is described by the series in Eq. 8.13 with the parameter set $(B_0 = B_{y0}, b_2, a_2 = 0, b_4, a_4 = 0)$ then the field of the same magnet, anti-clockwise rotated by 90 degrees, will be described by the same series (8.13) with the parameters set $(B_0 = B_{x0}, b_2 = 0, a_2 = -b_{2,\text{old}}, b_4 = 0, a_4 = b_{4,\text{old}})$. In general, the multipole coefficients a_n and b_n for a rotated magnet are linearly related to the old a_n and b_n, which coefficients depending on the angle of rotation.

8.4. TRANSPORT Matrix Elements

8.4.1. "Canonical" Coordinate Definitions. A notation for describing low order transfer map elements was originated by Karl Brown, at Stanford, in his program called TRANSPORT. This program was initially devoted to spectrometer design but the notation has been adopted for accelerator lattices, and the elements referred to as "TRANSPORT elements" even though, by now, their definitions have been slightly changed. The details though purely conventional, have to be understood for any inveestigation to higher than linear order. In TRANSPORT notation the general spatial motion of a point particle in a lattice is described by it 6 displacements from an ideal, or design, or reference particle;

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{pmatrix} \equiv \begin{pmatrix} x \\ x' \\ y \\ y' \\ \ell \\ \delta \end{pmatrix}, \tag{8.26}$$

where ℓ is longitudinal deviation from bunch center and the other components have been defined earlier. Small amplitude propagation from point (0) to general downstream point in a lattice can be approximated by the teading terms in a Taylor series;

$$x_i = \sum_{j=1}^{6} R_{ij} x_j(0) + \sum_{j=1^6} \sum_{k=j}^{6} T_{ijk} x_j(0) x_k(0). \tag{8.27}$$

One detail, now obsolete, was that by TRANSPORT convention, to reduce storage requirements and evaluation time, Brown chose to keep only "above diagonal" elements of T_{ijk}. i.e. the k summation starts at $k = j$. This exploits the fact that $x_j(0)x_k(0) = x_k(0)x_j(0)$. So, when comparing matrix elements, one has to check whether the corresponding off-diagonal elements are symmetric, or have had above-diagonal elements doubled and below-diagonal elements dropped.

A more significant distinction between conventional matrix element definitions concerns the choice of components for $\mathbf{x}$. In order for Hamiltonian dynamics to be applied most directly to particle propagation, true "canonical" coordinates should be used. But, to "geometricize" lattice theory, one divides all momentum components by the total momentum p_0, yielding a form of "scaled" phase space. (This, basically, is the source of the γ factor in the definition of *invariant* emittance.) In UAL the longitudinal spatial coordinate is usually referred to as $-ct$ which has units of length, with the sign adjusted so coordinate increases toward the front of the bunch. Following MAD, which follows MARYLIE, the longitudinal "momentum" is taken to be $\Delta\mathcal{E}/(p_0 c)$. Only by choosing these coordinates to be *canonical* can the requirements of symplecticity be exploited economically. All these definitions reduce to the original TRANSPORT definitions in the small amplitude, fully relativistic limit.

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{pmatrix} \equiv \begin{pmatrix} x \\ p_x/p_0 \\ y \\ p_y/p_0 \\ -ct \\ \Delta\mathcal{E}/(p_0 c) \end{pmatrix}, \tag{8.28}$$

From Hamiltonian point of view the fifth component is *time*, but for convenience this component is expressed as a distance by multiplication by c. For fully-relativistic motion this is, in fact, longitudinal displacement from reference position. The actual numerical values of the matrix elements depend on what conventions have entered the coordinate definitions. Fortunately the R_{ij} elements are largely independent of these choices and conversion of the T_{ijk} elements is fairly straightforward.

8.4.2. Linear Matrix Elements in a "Toy" Lattice. To illustrate the evaluation of matrix elements, and to check the UAL code, matrix elements for the `eq_tune_fodo` "toy" lattice illustrated in FIG 3.1 will be worked out and compared with output from the code. According to Eqs. (3.11) and (3.12) the transfer matrices through one complete full cell are given by

$$\begin{aligned} M_{11}^{(x)} &= \begin{pmatrix} 1-2q^2l^2 & 2l(1+ql) \\ -2q^2l(1-ql) & 1-2q^2l^2 \end{pmatrix} = \begin{pmatrix} \cos\mu_1 & \beta_1^{(x)}\sin\mu_1 \\ -\sin\mu_1/\beta_1^{(x)} & \cos\mu_1 \end{pmatrix}, \qquad (8.29) \\ M_{11}^{(y)} &= \begin{pmatrix} 1-2q^2l^2 & 2l(1-ql) \\ -2q^2l(1+ql) & 1-2q^2l^2 \end{pmatrix} = \begin{pmatrix} \cos\mu_1 & \beta_1^{(y)}\sin\mu_1 \\ -\sin\mu_1/\beta_1^{(y)} & \cos\mu_1 \end{pmatrix}, \end{aligned}$$

where expressions for $\beta_1^{(x)}$ and $\beta_1^{(y)}$ were given in Eqs. (3.16).

The diagonal **R**-matrix elements describing propagation through the fullcell can be read off by inspection;

$$R_{11} = R_{22} = R_{33} = R_{44} = \cos\mu_1, \qquad (8.30)$$

and, because δ is conserved

$$R_{55} = R_{66} = 1. \qquad (8.31)$$

Off-diagonal elements are

$$\begin{aligned} R_{12} &= \sin\mu_1\beta_1^{(x)}, \quad R_{21} = -\sin\mu_1/\beta_1^{(x)}, \qquad (8.32) \\ R_{34} &= \sin\mu_1\beta_1^{(y)}, \quad R_{43} = -\sin\mu_1/\beta_1^{(y)}. \end{aligned}$$

Since the deflection in sextupoles is quadratic in x and y the sextupoles do not contribute to **R**. The deflections in quads depend on δ but they are also linear in x and y, so quads do not contribute to R_{55} and R_{66}. But the deflection in the bending element (here treated as if at a single point) is (inversely) proportional to momentum. It is left as an exercise to show that

$$R_{16} = 2l(1+\frac{1}{2}lq)\Delta\theta, \quad R_{26} = (2-lq-l^2q^2)\Delta\theta. \qquad (8.33)$$

Working out these formulas also makes a good start toward working out the T_{ijk} elements.

One can check the **R** matrix elements with the values shown in Table 8.2. (Note that, as displayed (for programming convenience) this printout actually reads as the transpose $\mathbf{R}^T$.) For this data the relevant numerical values were

```
nhalf := 200
lhalf := 10
q := 0.061909
deltheta := 0.031415
cellmu := 1.33517
s1 := 0.05767
s2 := -0.10939
betax1 := 33.302
betay1 := 7.8345
```

TABLE 8.2. R_{ij} elements for a weakly-coupled system

0.231876	-0.029239	0.000000	0.000000	-0.031317	0.000000
32.36138	0.231876	0.000000	0.000000	-0.822722	0.000000
0.000000	0.000000	0.233472	-0.124089	0.000000	0.000000
0.000000	0.000000	7.619439	0.233472	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	1.000000	0.000000
0.822722	0.031317	0.000000	0.000000	-0.012908	1.000000

PROBLEM 8.1. *Adjust the input parameters to* `eq_tune_fodo.adxf` *to replicate the* **R** *matrix listed as Table (8.2). The value* `nhalf=200` *has been chosen to be large enough to make this a* big *accelerator, to de-emphasize the dipole focusing effect. Give two matrix elements that are very nearly equal, but would be exactly equal, if there were no bending elements in the FODO cell. The surviving x, z coupling figures prominently in the following problems.*

PROBLEM 8.2. *Use formulas in this section to derive all* **R** *matrix elements except* $\texttt{R}_{5i}$, `i=1..6`. *Note that* $\texttt{R}_{66}$`=1` *and* $\texttt{R}_{6i}$`=0` *for* `i`$\neq$ `6` *since magnetic elements cannot change the particle energy or, therefore, δ.*

PROBLEM 8.3. *Using any matrix processing device you have available, such as programmable hand calculator or spreadsheet program, to calculate the determinant* $\det\mathbf{M}$. *Is this correct? If, as a last resort, you choose to work out the determinant by brute force calculation with a matrix-deprived calculator, be sure to observe that there are no off-block-diagonal elements coupling either x or z to y. This means the y elements can be treated as a 2×2 matrix and the x and z components as 4×4.*

PROBLEM 8.4. *Refer back to formulas in the section 5.1, "Analysis of a 4×4 Symplectic Matrix". Those formulas emphasized coupling between x and y but the same formulas apply to the x and z coupling in the matrix* **M** *under study. Simply suppress the third and fourth rows and columns. Since any elements dropped having indices corresponding to x or z vanish, the remaining 4×4 matrix can be analysed as in that section. Evaluate the elements of the off-diagonal submatrices* **B** *and* **C** *and show that the coupling terms of our (now 4×4) matrix* **M** *do not shift either horizontal or longitudinal tunes.*

PROBLEM 8.5. *Use Eq. (5.3) to derive* $\mathbf{M}^{-1}$ *and confirm the result to be correct by evaluating* $\mathbf{M}^{-1}\mathbf{M}$. *This proves (or not) that* **M** *is symplectic. Unit determinant is a necessary, but not sufficient condition for symplecticity.*

PROBLEM 8.6. *Previously all but the elements* $\texttt{R}_{5i}$, `i=1..6` *have been checked analytically. Use symplecity to determine the* $\texttt{R}_{51}$, $\texttt{R}_{52}$, *and* $\texttt{R}_{56}$ *elements.*

8.4.3. Second Order Matrix Elements For Individual Elements. In working out the T_{ijk} elements for the full FODO cell one must first have second order expressions for the individual elements. For a thin quad the deflections are

$$\Delta x' = -\frac{q}{1+\delta}\,x \approx -qx + \delta\, qx, \quad \Delta y' = \frac{q}{1+\delta}\,y \approx qy - \delta\, qy. \tag{8.34}$$

These formulas yield second order thin quad elements

$$T_{216} = q, \quad T_{436} = -q. \tag{8.35}$$

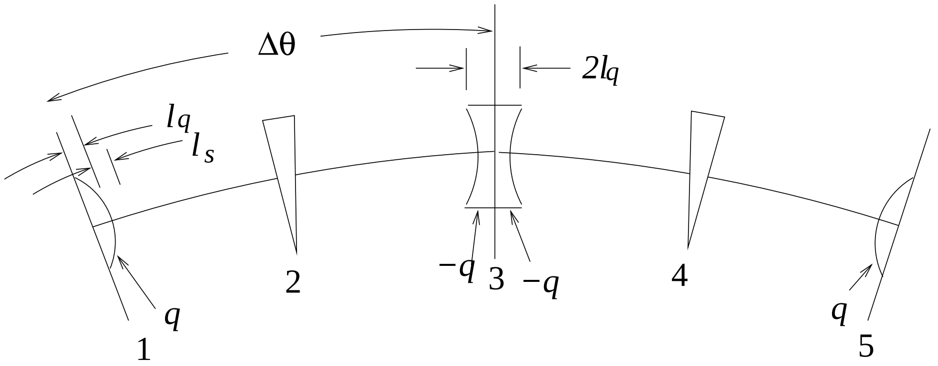


Figure 8.3. Labeling of points in a full cell of the `eq_tune_fodo` lattice.

Referring to Table 8.1, for a thin sextupole the deflections are given by

$$\Delta x' = -\frac{S}{2}\,(x^2 - y^2), \quad \Delta y' = Sxy, \tag{8.36}$$

which produce

$$T_{211} = -\frac{S}{2}, \quad T_{233} = \frac{S}{2}, \quad T_{413} = S. \tag{8.37}$$

To quadratic order in δ the deflections in a thin dipole are given by

$$\Delta x' = -\frac{\Delta\theta}{1+\delta} = -\Delta\theta + \Delta\theta\,\delta - \Delta\theta\,\delta^2, \quad \Delta y' = 0, \tag{8.38}$$

which give

$$T_{266} = -\Delta\theta, \tag{8.39}$$

as the only non-vanishing second order matrix element.

8.4.4. Concatenation of Matrix Elements. Even for the simplest possible nontrivial lattice (namely the `eq_tune_fodo` full cell we are working with) it is fairly laborious to work out even second order elements. (This is known as "concatenating" the maps sequentially.) Still it is worth working out a few to get a feel for what is involved. Within UAL this concatenation is performed using differential algebra (DA) on truncated power series (TPS) but here a more elementary approach is to be used for calculating some of the elements.

Labeling of intermediate points is given in FIG. 8.3. Let us work on only horizontal motion. To obtain the subsequent effect of a deflection at one point it is useful to define an "influence function" or "sine-like function" $G_x(j,i)$ that gives the effect at element position j of unit deflection at position i. As this function propagates through subsequent elements it neglects all but linearized deflections. Extra deflections at downstream points will each launch their own sine-like contributions.

For our lattice it is straightforward to work out all needed $G_x(j,i)$ entries.

$$\begin{aligned}
&G_x(5,1) = 2l(1+ql), \quad &&G'_x(5,1) = 1 - 2q^2l^2, \\
&G_x(5,2) = \frac{3}{2}\,l + ql^2, \quad &&G'_x(5,2) = 1 - \frac{1}{2}ql - q^2l^2, \\
&G_x(5,3) = l, &&G'_x(5,3) = 1 - ql, \\
&G_x(5,4) = \frac{1}{2}\,l, &&G'_x(5,4) = 1 - \frac{1}{2}\,ql, \\
&G_x(5,5) = 0, &&G'_x(5,5) = 1.
\end{aligned} \tag{8.40}$$

$$\tag{8.41}$$

Defining $\Delta x'(i)$ to be the *exact* deflection at location i, propagation from 1 to 5 is given by the *exact* equations

$$\begin{aligned}
x(5) = {}& \cos\mu\, x(1) + \beta_1^{(x)} \sin\mu_1\, x'(1) + \sum_{i=1}^{5} G_x(5,i)\Delta x'(i), \\
x'(5) = {}& -\frac{1}{\beta_1^{(x)}} \sin\mu_1\, x(1) + \cos\mu_1\, x'(1) + \sum_{i=1}^{5} G'_x(5,i)\Delta x'(i).
\end{aligned} \tag{8.42}$$

Here a sine-like contribution is added for each perturbation (i.e. at sextupoles) and (if $\delta \neq 0$) at bends. These equations are deceptively simple. The *exact* deflections $\Delta x'(i)$ can only be obtained by exactly accounting for all preceeding deflections. But, since we are satisfied to calculate only quadratic terms, it is easy to drop terms that don't contribute. Eqs. (8.42) are also somewhat ambiguous as to the treatment of momentum dependence. Certainly the deflections depend on momentum, but it is not clear whether δ is to be treated as a parameter in a $2D$ approach or as a component of $\mathbf{x}$. We will follow the latter, $3D$ approach.

We plan (here) to keep only linear terms, x, x', and δ, and quadratic terms x^2, xx', x'^2, δ^2, $x\delta$, and $x'\delta$. Of course there is also the possibility of a *constant* term appearing. Even though it is not mathematically consistent, let us refer to such a constant term as also being "linear". The only possible justification for doing this is that the presence of such a term really implies a shift of the equilibrium orbit which could be used to suppress this term.

There is an important physical distinction between "geometric" terms x^2, xx', and x'^2 and "chromatic" terms δ^2, $x\delta$, and $x'\delta$. The former terms are *nonlinear* functions of the dependent variable $\mathbf{x}$. They have the property of becoming arbitrarily large as x or x' becomes large. Such terms inevitably cause motion to be unstable at sufficiently large amplitude. (The only exception to this behavior is when the nonlinear term is, by itself, unphysical. This can occur when a force that is well-behaved at large amplitude, such as a beam-beam force, is approximated by a term, such as a cubic, which diverges at large amplitude.) This leads to the concept of "dynamic aperture" that will be pursued later on. The chromatic terms actually leave the theory linear. All such terms could be incorporated exactly by treating δ as a parameter rather than as a coordinate. This could be regarded to be a more elegant, though less straightforward, approach. The leading effect of chromatic terms is to cause the equilibrium orbit to be displaced more or less proportional to δ and to cause the tunes to depend on momentum. In practice, the optical degradation due to δ is often at least as serious as the degradation due to nonlinear terms.

Suppressing the (0)'s from the initial displacements, Eqs. (8.42) can be expressed in terms of **R**-matrix elements

$$\begin{aligned} x(5) &= R_{11}\,x + R_{12}\,x' + R_{16}\,\delta + \sum_{i=1}^{5} G_x(5,i)\Delta x'(i), \\ x'(5) &= R_{21}\,x + R_{22}\,x' + R_{26}\,\delta + \sum_{i=1}^{5} G'_x(5,i)\Delta x'(i). \end{aligned} \tag{8.43}$$

Apart from having introduced the term linear in δ the only new feature of these equations is that we need retain only quadratic terms in the final summations. To calculate to this accuracy it is adequate to calculate them using the following unperturbed (i.e. first order accuracy) trajectory.

$$\begin{aligned} x(1) &= x, \\ x(3) &= (1-ql)x + lx' + \Delta\theta\frac{l}{2}\,\delta, \\ x(5) &= (1-2q^2l^2)x + 2l(1+ql)x' + 2l(1+\frac{1}{2}ql)\Delta\theta\,\delta. \end{aligned} \tag{8.44}$$

The second order deflections at the various points in the cell are

$$\begin{aligned} \Delta x'(1) &= q\,x(1)\,\delta - \frac{S_1}{2}\,x^2(1), \\ \Delta x'(2) &= \;-\Delta\theta\,\delta^2, \\ \Delta x'(3) &= \;-2q\,x(3)\delta - S_2\,x^2(3), \\ \Delta x'(4) &= \;-\Delta\theta\,\delta^2, \\ \Delta x'(5) &= q\,x(5)\,\delta - \frac{S_1}{2}\,x^2(5). \end{aligned} \tag{8.45}$$

To get an element T_{1jk} or T_{2jk} for the full cell we substitute into Eq. (8.45) from Eq. (8.44) and then substitute the result, along with Eq. (8.40), into Eq. (8.43). Then the T_{ijk} elements are obtained by matching terms. For example the coefficient of x^2 in the expansion of $x(5)$ is

$$T_{111} = -S_1\,l(1+ql) - S_2\,l(1-ql). \tag{8.46}$$

All second order elements for x and x' are:

$$\begin{aligned}
T_{111} &= -\,S_1\,l(1+ql) - S_2\,l(1-ql)^2,\\
T_{122} &= -\,S_2 l^3,\\
T_{166} &= -\,2\Delta\theta\,l(1+ql) - \frac{1}{4}S_2\Delta\theta^2\,l^3,\\
T_{112} &= -\,2S_2\,l^2(1-ql),\\
T_{116} &= 4q^2l^2 - S_2\Delta\theta\,l(1-ql),\\
T_{126} &= -\,2ql^2 - S_2\Delta\theta\,l^3,\\
T_{211} &= -\,S_2(1-3q^2l^2+2q^4l^4),\\
T_{222} &= -\,S_2\,l^2(1-ql) - 2S_1\,l^2(1+ql)^2,\\
T_{266} &= -\,\Delta\theta(2-2ql-3q^2l^2) - \frac{1}{4}S_2\,l^2\Delta\theta^2(1-ql) - 2S_1\,l^2\Delta\theta^2(1+\frac{1}{2}ql)^2,\\
T_{212} &= -\,2S_2\,l(1-ql)^2 - 2S_1\,l(1-2q^2l^2)(1+ql),\\
T_{216} &= 4q^2l(1-\frac{3}{2}ql) - S_2\Delta\theta\,l(1-ql)^2 - 2S_1\,l\Delta\theta(1+\frac{1}{2}ql)(1-2q^2l^2),\\
T_{226} &= 4q^2l^2 - S_2\,l^2\Delta\theta(1-ql) - 4S_1\,l^2\Delta\theta(1+ql)(1+\frac{1}{2}ql).
\end{aligned} \tag{8.47}$$

As mentioned earlier, some of these elements depend on the detailed definition of the components of $\mathbf{x}$. Unless adjusted appropriately these elements will therefore not agree well with values calculated by UAL.

PROBLEM 8.7. *Using formulas (8.47), check several entries in the* $\mathtt{T}_{ijk}$ *table resulting from the processing the* `eq_tune_fodo` *lattice. Be sure that all parameters are identical to the parameters used in the earlier problem set in which* $\mathtt{R}_{ij}$ *elements were derived. It may be necessary to modify the code to cause the* $\mathtt{T}_{ijk}$ *elements to be evaluated and printed.*

8.5. Truncated Power Series and Lie Maps

8.5.1. Function evolution. Truncated power series play an important role in UAL. They are used to approximate the "maps" that express "output" particle coordinates (at a later place in the ring) in terms of "input" particle coordinates (at an initial place in the ring). When truncated to linear order these power series reduce to the elements of the traditional, Courant-Snyder, transfer matrix description of the accelerator lattice. Historically, most of accelerator physics has been (very successfully) based on analysis performed in this linear limit. But effects appearing already at a "next order of approximation" such as chromaticity and amplitude-dependent detuning, have ways of intruding, even in elementary contexts, and nonlinearity becomes increasingly important as amplitudes are increased to achieve higher beam current. As soon as any nonlinearity whatsoever is allowed to enter the description the issue of symplecticity, or rather lack thereof, rears its head. Especially for hadron accelerators, for which there is essentially no true damping, any anti-damping artificially and erroneously introduced through non-symplecticity can ruin an accelerator simulation program's ability to predict the long term future.

Symplectic maps (typically nonlinear) are also known as Lie maps. One therefore seeks to describe particle trajectories in an accelerator by a Lie map. As with all physics, such a description can only be approximate. For one thing the idealized model of the accelerator, on which the "idealized map" is based, is undoubtedly inaccurate and incomplete. Accepting this as inevitable, possible further inaccuracy results from the computer program's representation of the map. It is the latter source of inaccuracy that is the subject of this section. Maps based on truncated power series can only approximate idealized maps. For reasons explained in the previous paragraph, failure of symplecticity is expected to be more serious than other inaccuracy. An important goal of UAL is to preserve symplecticity, or rather to keep the inevitable failure of symplecticity controllably small.

There is no shortage of excellent reference material concerning Lie maps; for example Dragt[**24**] and Forest[**25**]. Because the subject is abstract, and is sometimes considered impenetrable, this section tries to give a self-contained, elementary discussion of the general ideas. To reduce complexity the discussion will be restricted to two dimensional (x, p) phase space, with p used instead of p_x. All results generalize naturally to higher dimensions.

If (x_0, p_0) represents input particle coordinates, the sort of map $\mathcal{M'}_{10}$ under discussion expresses output coordinates (x_1, p_1) as functions of input coordinates (x_0, p_0). (The prime on $\mathcal{M'}_{10}$ will be explained shortly.) For linear maps this map reduces to a 2×2 matrix, the traditional transfer matrix of standard accelerator theory. If nonlinearity is present it is natural to introduce a "generalized transfer matrix" $\mathcal{M'}_{10}$ in which the four matrix elements are nonlinear functions of x_0 and p_0. Usually these nonlinear functions are expressed as truncated Taylor seris. Like it or not, this is the representation one is forced to use in a computer representation of the map.

Consider an arbitrary function $f(x, p)$—one may think of f as expressing the dependence on position in phase space of some physical quantity. A particle trajectory defines an evolution of the particle coordinates and it is natural to inquire about the corresponding evolution of f. One has to be aware of the ambiguity accompanying the distinction between function *form* and function *value*. For example, suppose transformation $\mathcal{M}'_{10}$ yields forward formula $x_1 = x_1(x_0, p_0) = ap_0 + bp_0$ and backward formula $x_0 = x_0(x_1, p_1) = cx_1 + dp_1$, and that the value of function f is defined to be "the first component squared"; at input this is x_0^2, at output it is x_1^2. An assignment one might have received in calculus class was to figure out the value of x_0^2 from knowledge only of x_1 and p_1. Expressed in terms of output coordinates the input value of f is $(x_0(x_1, p_1))^2 = (cx_1 + dp_1)^2$. From a physicist's point of view, this is tortured usage. By the "evolved value of f" one presumably means x_1^2, the square of the first component, evaluated at the evolved location. This is the way functions of coordinates are to be interpreted; for example

$$x_1^2 = f(x_1, p_1) = f(\mathcal{M}'_{10}(x_0), \mathcal{M}'_{10}(p_0)) = (ax_0 + bp_0)^2. \tag{8.48}$$

Since the *form* of the function does not change, to evaluate this evolution, as Eq. (8.48) shows, it is adequate to have formulas for the evolution of individual components. This is the functionality provided by the vectors of truncated power series provided, for example, by UAL. But, for theoretical purposes, a slightly more abstract generalization of transfer matrices is preferable. Let us define transfer map $\mathcal{M}_{10}$ as operating on *functions* (of location phase space) rather than acting individually on the components. That is

$$f_1 = \mathcal{M}_{10} f_0, \tag{8.49}$$

which is defined to mean the same thing as Eq. (8.48). Forest calls $\mathcal{M}$ a "compositional map". It is a one-component map acting in an infinite dimensional space (of functions defined on phase space.) Note that it is the *value* of the function that evolves; the *form* of the function does not change. Since x_0 and p_0 can, individually, be thought of as functions of the (x_0, p_0) pair, the specialization back to the representation by a vector-organized set of nonlinear functions is immediate. So there is no "physics" in Eq. (8.49) to distinguish it from Eq. (8.48).

Assuming, as we are, that the physical elements in the lattice are known perfectly, the equations of motion can, in principle, be used to determine $x(s), p(s)$, the dependence on longitudinal coordinate s of a particle trajectory. Commonly the equations of motion are written in Hamiltonian form and knowing the equation of motion is sometimes expressed as "knowing the Hamiltonian". Because of the complexity of accelerator lattices it is almost never practical to solve the equations of motion analytically and it is rarely practical to solve them numerically. Rather the map through a sector of the lattice is formed by concatenating the maps of the individual elements in the sector. This usually involves truncation of power series.

8.5.2. Taylor series in more than one dimension and Lie maps. The Taylor series representation of one dimensional functions is second nature to most scientists (perhaps because learned about in high school as the binomial theorem?) The function of Lie maps is to generalize this description to more than one dimension.

The theory of function evolution, as invented by Lie, has been applied a century later, in the context of celestial mechanics, by Hori[**26**] and, in the context of accelerator mechanics, by Dragt.[**27**] The discussion here more nearly follows Hori than Dragt.

Let (x, p) be coordinates in 2D phase space, and $f(x, p)$ be a function that is arbitrary (except for possible requirements such as smoothness and absence of vanishing derivatives.) We wish to express the value of f at some phase space point in terms of the values of its derivatives at some other point.

We know how to do this in 1D—use a Taylor series. We therefore try to reduce the 2D problem to 1D. Toward this end we draw a family of smooth curves in phase space (to be referred to as a "congruence" of curves) that have properties: (a) there is a curve through every point, (b) no curve crosses any other in the region under discussion, and (c) there is a function $S(x,p)$, not necessarily unique, such that $x(\tau),p(\tau)$ (the coordinates of the curve as functions of a running parameter τ) are solutions of the equations

$$\frac{dx}{d\tau} = \frac{\partial S}{\partial p}, \quad \frac{dp}{d\tau} = -\frac{\partial S}{\partial x}. \tag{8.50}$$

The function $S(x,p)$ is such that its derivatives on the right hand side of this equation define, at every point (x,p), the direction of the tangent to the curve passing through that point. Note that S is *a priori* an arbitrary function, unrelated to the dynamics under study.

Along any one of the curves of the congruence, the value of arbitrary function f can be expressed, as a function of τ, by $f(x(\tau),p(\tau))$. One can define an along-the-curve derivative operator

$$\{\cdot,S\} \equiv \left.\frac{d}{d\tau}\right|_S = \frac{dx}{d\tau}\frac{\partial}{\partial x} + \frac{dp}{d\tau}\frac{\partial}{\partial p} = \frac{\partial S}{\partial p}\frac{\partial}{\partial x} - \frac{\partial S}{\partial x}\frac{\partial}{\partial p}. \tag{8.51}$$

In this notation the $\cdot$ is a "place holder" indicating the operator $\{\cdot,S\}$ is "waiting for" a function, such as f, for its argument. (Except for change in sign/order-of-arguments, $\{\cdot,S\}$ is the same as the function for which Dragt introduced the notation $:S:$.) When acting on function f, the result is $\{f,S\} \equiv\, :S:(f)$, which can be recognized as the "Poisson bracket" of f and S.

Now we can exploit our congruence of curves for its advertised purpose of relating values of f at separated points This is especially easy if the points happen lie on the same curve because, on that curve, the function depends only on the single variable τ. In this case, let the parameters of the points that are to be related be τ and $\tau+\epsilon$. It may be helpful conceptually to regard ϵ as being "small", and this may be appropriate when discussing the convergence of the series, but no such formal requirement is assumed. Expressing the Taylor series in new, unconventional form, we have

$$f(\tau+\epsilon) = \left.(1+\epsilon\,\{\cdot,S\} + \frac{1}{2!}\epsilon^2\,\{\{\cdot,S\},S\} + \frac{1}{3!}\epsilon^3\,\{\{\{\cdot,S\},S\},S\} + \cdots)f(\tau+\epsilon)\right|_{\epsilon=0}, \tag{8.52}$$

As usual the derivatives on the right hand side must be evaluated for general ϵ but then ϵ is set to zero. This is known as the Lie map corresponding to function S. Recognizing the terms in this series as corresponding to an exponential function, this series is traditionally abbreviated to

$$f(\tau+\epsilon) = e^{\epsilon\,\{\cdot,S\}}\,f(\tau); \tag{8.53}$$

but, to evaluate the series numerically, expansion Eq. (8.52) is what is required. Furthermore the evaluation has to be truncated at some point. Any differential algebra package, such as COSY[**28**] or the ZLIB module of UAL, can calculate derivatives of functions, and can therefore evaluate the Poisson bracket expressions appearing in Eq. (8.52).

This section has been about calculus, no more, no less. There has been no mechanics, Hamiltonian or otherwise. If the signs in Eq. (8.50) had been chosen differently, say both positive, the analysis would have gone through unchanged, except for the switching the sign in the bracket expression, which would therefore no longer deserve be called a "Poisson bracket".

8.5.3. Symplecticity of Lie map. Hori[**26**] gave a different interpretation to Eq. (8.53), regarding it as a change of variable rather than as an evolution equation. To encourage this interpretation let us replace (x_0, p_0) by (ξ, η) and (x_1, p_1) by (x, p) and interpret the equation as a change of variables from (ξ, η) coordinates to (x, p) coordinates. The coordinates (ξ, η) are assumed to be "canonical"—this means that their Poisson brackets reckoned using some known-to-be canonical starting coordinates, call them (x', p'), have the appropriate, 0 or 1 values. Copying from Eq. (8.52) and restoring the 2D arguments of f;

$$f(x,p) = (1+\epsilon\,\{\cdot,S\}+\frac{1}{2!}\epsilon^2\,\{\{\cdot,S\},S\}+\frac{1}{3!}\epsilon^2\,\{\{\{\cdot,S\},S\},S\}+\cdots)f(\xi,\eta)\Big|_0 \,. \qquad (8.54)$$

Here S is, as before, an arbitrary function, and evaluation of the derivatives on the right hand side depends upon the congruence of curves determined by Eqs. (8.50). (The cryptic subscript 0 is supposed to convey this.)

It was mentioned earlier that either one of the coordinates, say ξ, is a satisfactory version of the function f. Plugging this into Eq. (8.54) yields

$$x = (1+\epsilon\,\{\cdot,S\}+\frac{1}{2!}\epsilon^2\,\{\{\cdot,S\},S\}+\frac{1}{3!}\epsilon^2\,\{\{\{\cdot,S\},S\},S\}+\cdots)\xi\Big|_0 \,, \qquad (8.55)$$

and a similar formula relates p to η. By restoring the single variable, along-curve parameterization (and for brevity, arraying formulas as components of a vector) these equations can be written in a more useful form;

$$\begin{pmatrix}\xi(\tau+\epsilon)\\ \eta(\tau+\epsilon)\end{pmatrix} = \begin{pmatrix}x\\ p\end{pmatrix} = (1+\epsilon\,\{\cdot,S\}+\frac{1}{2!}\epsilon^2\,\{\{\cdot,S\},S\}+\frac{1}{3!}\epsilon^2\,\{\{\{\cdot,S\},S\},S\}+\cdots)\begin{pmatrix}\xi(\tau+\epsilon)\\ \eta(\tau+\epsilon)\end{pmatrix}\Bigg|_{\epsilon=0} \qquad (8.56)$$

This shows that the pair (x, p) are, except for "translation" along a curve of the congruence, the same as the pair (ξ, η).

This has still been "just calculus", but let us now use the assumption that (ξ, η) are canonical variables of a Hamiltonian system. Then Eq. (8.56) provides a change of variables to new variables (x, p). Now the amazing part; since the (ξ, η) variables are, by hypothesis, canonical through the region under discussion and (x, p) are just "translations" of (ξ, η), transformation (8.56) is necessarily canonical.

Hori[**26**] goes on to develop a perturbation theory based on this formulism. He regards the function S as a *kind of* "generating function" (though it must not be confused with a "Goldstein" generating function) and goes on to develop an iterative procedure to determine S and new coordinates in ascending powers of a "small parameter" of the perturbation. None of this is relevant for UAL. What *is* relevant is that transformations generated by Lie maps are symplectic. By controlling the number of terms retained in the power series evaluation one can control (or even make negligible) the degree of nonsymplecticity.

8.5.4. Hamiltonian maps. Returning to the trajectory evolution interpretation of our equations, the Taylor series derived so far might seem to be useless for the following reason: it relates only phase space points lying on the same curve and no prescription has been given for choosing the function $S(x,p)$ such that two arbitrarily chosen points lie on the same curve. But, as it happens, we do not have to insist that the points be arbitrarily chosen. We are interested in points lying on a single particle trajectory. One visualizes this trajectory as a three dimensional curve in the (x,p,t) space, where t is time, or if one prefers, a longitudinal coordinate. Projected onto the (x,p) plane the curve passing through input point (x_0,p_0) necessarily passes through output point (x_1,p_1). The orbit is determined by solving Hamilton's equations;

$$\frac{dx}{dt} = \frac{\partial H}{\partial p}, \quad \frac{dp}{dt} = -\frac{\partial H}{\partial x}. \tag{8.57}$$

where $H(x,p)$ is the Hamiltonian function. Notice that these equations are identical to Eqs. (8.50) if the function S in those equations is replaced by H (and τ by t.) This magically eliminates both limitations of the formalism of the previous section. The map has become

$$f(t_0+t) = e^{t\,\{\cdot,H\}}\, f(t_0). \tag{8.58}$$

(As explained above, when written in this form, the notation is too compressed for the required operations to be exhibited explicitly, as they are in Eq. (8.52).) Replacing f by the individual coordinates, as before, yields

$$\begin{pmatrix} x(t_0+t) \\ p(t_0+t) \end{pmatrix} = e^{t\,\{\cdot,H\}} \begin{pmatrix} x(t_0) \\ p(t_0) \end{pmatrix}. \tag{8.59}$$

Generalized to six dimensions and truncated to arbitrary order, Eq. (8.59) is a form in which the evolution of a particle trajectory can be simulated in a computer. If Hamiltonian H is only approximate the evolution it produces can be only approximate, but any failure of symplecticity can be reduced by keeping more terms in the expansion.

8.5.5. Discrete maps. Eq. (8.59) represents a continuous mapping—the explicit appearance of t invites taking the limit $t \to 0$. Similarly the occurence of factor ϵ in Eqs. (8.56) invites the limit $\epsilon \to 0$ and a continuous interpretation. But, if the ϵ factor is subsumed into the S function, Eqs. (8.53) represents a discrete map, potentially propagating the particle coordinates through a sector of arbitrary length.

For example consider the function

$$S = S_0^3 x^3 + S_1^3 x^2 p + S_2^3 x p^2 + S_3^3 p^3. \tag{8.60}$$

Substitution into Eq. (8.56) yields propagation $(x,p) \to (x',p')$

$$\begin{aligned} x' &= x + \{x,S\} + \cdots = x + S_1^3 x^2 + 2S_2^3 xp + 3S_3^3 p^2 + \dots, \\ p' &= p + \{p,S\} + \cdots = p - 3S_0^3 x^2 - 2S_1^3 xp - S_2^3 p^2 + \dots. \end{aligned} \tag{8.61}$$

This map is special in that it is an identity map to linear order. It could therefore not represent arbitrary propagation through a general sector. But, after "factoring out" the linear part of a general map the remaining part can be reduced to Eq. (8.60) by truncation to quadratic order.

Perhaps the procedure just mentioned can be reversed? Suppose that propagation formulas (8.61) have been determined by applying some integrator to an arbitrary lattice sector. If the sector has more than a few nonlinear elements such a determination would have required truncation, for example to quadratic order, as in Eq. (8.61). The integrator will therefore have determined the coefficients in expansions

$$\begin{aligned} x' &= x + X_0^2 x^2 + X_1^2 xp + X_2^2 p^2 + \dots, \\ p' &= p + P_0^2 x^2 + P_1^2 xp + P_2^2 p^2 + \dots. \end{aligned} \tag{8.62}$$

For these equations to be consistent with Eqs. (8.61) the six equations obtained by equating coefficients must be satisfied. Regarding the four S_i^3 coordinates as the unknowns, they can be determined from just four of the equations. The remaining two equations will not, in general, be satisfied. But, if the integrator determining series (8.62) were symplectic (to the order of terms retained), then these equations would be redundant and the redundant equations would necessarily be satisfied. These equations can therefore be applied as a check on the symplecticity of the integrator.

Assuming the integrator is symplectic, so that the redundant equations (to quadratic order) are satisfied, the function S will have been determined to cubic order. A function S determined in this way can be called a "pseudo-Hamiltonian". By using this function in Eq. (8.56), and retaining more terms in the series, propagation formulas for the coordinates can be obtained to higher than quadratic order. Such formulas would be useless for studying large amplitude features such as resonant islands, onset of chaos, or dynamic aperture. But for "intermediate" amplitude trajectories the formulas can represent propagation that is both "correct to quadratic order" (for example modeling chromaticity) while being symplectic to higher than quadratic order.

This procedure can be illustrated by explicit example. Consider a map

$$\mathbf{x}_2 = \mathbf{M}\,\mathbf{x}_1 \approx \mathbf{M}^{(1)}\,\mathbf{x}_1, \tag{8.63}$$

where $\mathbf{M}^{(1)}$ is the necessarily symplectic, linearized matrix approximation of the map. (Since $\mathbf{x}$ represents the components as a vector, we may as well take it to represent the coordinates in 6D phase space.) Define $\widetilde{\mathbf{M}}$ such that

$$\mathbf{x}_2 = \widetilde{\mathbf{M}}\,\mathbf{M}^{(1)}\,\mathbf{x}_1, \quad \text{or} \quad \widetilde{\mathbf{M}} = \mathbf{M}\,{\mathbf{M}^{(1)}}^{-1}. \tag{8.64}$$

Suppose that $\mathbf{M}$ has been obtained to some order of accuracy, say $\mathbf{M}^{(2)}$. Then $\widetilde{\mathbf{M}}$ is known to corresponding order. Let S be determined such that

$$\widetilde{\mathbf{M}}^{(2)} = \mathbf{M}^{(2)}\,{\mathbf{M}^{(1)}}^{-1} = 1 + \{\cdot, S\}. \tag{8.65}$$

Defining

$$\widetilde{\mathbf{M}}^{(3)} = 1 + \{\cdot, S\} + \frac{1}{2}\,\{\{\cdot, S\}, S,\}, \tag{8.66}$$

then

$$\mathbf{M} \approx \widetilde{\mathbf{M}}^{(3)}\,\mathbf{M}^{(1)}, \tag{8.67}$$

is symplectic to higher order than was $\widetilde{\mathbf{M}}^{(2)}$.

Quadrupole end field correction is an example of this procedure. Since the longitudinal interval for this correction was taken to have zero length, terms beyond the first vanish because they are proportional to higher powers of ϵ.

8.5.6. Bunch Evolution Using Maps. In Section 7.4, which discussed a decoherence/recoherence simulation, it was shown that particle evolution could be described either by element-by-element tracking, or by map tracking. It would be just as logical for that example to appear here as there.

SIMULATION 8.1. *Run the decoherence/recoherence example using the map tracking option for various order of truncated power series (TPS). For the results shown in FIG 7.6 the order was 3, which is also known as "octupole order".*

8.6. Thin and Thick Elements, and Symplecticity

8.6.1. Pure Kicks. Elements that are thin enough that they can be represented by discontinuous slope changes (or kinks) at fixed position s are known as "kicks" and computer codes that employ only kicks are known as "kick code". The rationale for the existence of kick codes is that, consistently employed, they preserve symplecticity. The transfer matrix for such an element is

$$\mathbf{M}_{\text{kick}} = \begin{pmatrix} 1 & 0 \\ f(\mathbf{x}) & 1 \end{pmatrix}, \tag{8.68}$$

where $f(\mathbf{x})$ is an arbitrary function of transverse position $\mathbf{x}$. The Jacobean for this matrix is 1, independent of $\mathbf{x}$. Unity determinant (for all $\mathbf{x}$ is the necessary and sufficient condition for a 2×2 matrix to be symplectic. Furthermore a 4×4 kick can be decomposed into a product of two 2×2 kicks. It follows that kicks always preserve symplecticity.

Another way of assuring that a transformation is symplectic is to derive it exactly, rigorously respecting Maxwell's equations and Newton's law (appropriately generalized to relativistic mechanics).

8.6.2. Pure Drifts. Since kick elements are unphysical idealizations, one would greatly prefer to represent accelerator elements by symplectic, thick element matrices. Unfortunately the number of element types for which such matrices are known is quite small. Even the transfer matrix for a drift

$$\mathbf{M}_{\text{drift}} = \begin{pmatrix} 1 & \ell \\ 0 & 1 \end{pmatrix}, \tag{8.69}$$

is symplectic only for appropriate transverse coordinates. This matrix corresponds to the exact transformation equations for a drift, which are

$$\frac{x_1 - x_0}{\ell} = \frac{dx}{ds}\Big|_0, \quad \frac{dx}{ds}\Big|_1 = \frac{dx}{ds}\Big|_0, \quad p_{x1} = p_{x0}. \tag{8.70}$$

Assuming (unrealistically) that there is no vertical momentum, and recalling that p_x is the actual transverse momentum divided by the total momentum p_0, one sees from a momentum vector diagram that

$$\frac{dx}{ds} = \frac{p_x}{\sqrt{1 - p_x^2}}. \tag{8.71}$$

But, knowing that x and p_x are canonically conjugate variables, one sees that x and dx/ds cannot be. So, without taking care to include the square root factor, even a drift can be non-symplectic. Because $p_x << 1$, one is sometimes justified in neglecting the denominator factor. But in a hadron accelerator, where particles rotate for billions of turns, even the tiniest of failures of symplecticity can give totally wrong results.

8.6.3. Drift-Kick Split. We have seen that pure kicks are naturally symplectic and that, with care, drifts can be treated symplectically. For a purely numerical calculation of a particle trajectory in an arbitrary magnetic field it is natural to introduce "symplectic integration algorithms" in which the deflections are concentrated in (infinitely) thin elements and the thick dimension is filled up with drifts. The simplest such algorithm is known as *drift-kick split.* A thick element is segmented into two or more drifts with a kick sandwiched at each interface. This approach was introduced in TEAPOT[**6**] and it is documented further in the TEAPOT manual. This code has been ported to C++ as one of the propagation algorithms available in UAL.

8.6.4. Symplectic Propagation Through Sector Bends . Other than for drifts, the only static magnetic elements for which exact, thick element equations are available are uniform field magnets. The trajectories are perfect circles or, in three dimensiona, helices. Even in this case, finding the exact exit coordinates requires the use of quite complicated geometry. In the following calculation we have the temerity to work out in detail a result due to E. Forest, which he states (without derivation) in his *Beam Dynamics* book.

To meet the "exactness" requirement for a thick element one needs analytic expressions for output coordinates as functions of input coordinates. To meet the requirement of polymorphic description with coordinates given either as numbers or as truncated power series (TPS), these formulas need to be explicit and free from branched evaluation routes. For a finite sector bend of angle $-\Phi$ these conditions are met by the following formulas, which relate to the geometry exhibited in FIG 8.4. These formulas are given as Eqs. (12.18) of Forest's book.

To reduce complexity a bit the figure illustrates pure radial (typically horizontal) motion. (i.e. in the plane of symmetry of the magnet.) This can also be regarded as the projection onto the magnet midplane of a completely general orbit of momentum $p_0(1+\delta)$. By conventional definition of transverse "momentum" coordinate p_x, if the nominal momentum is p_0, then the actual transverse momentum is $p_x p_0$. With vertical momentum (scaled the same way) given by p_y, the total in-plane momentum is

$$p_{\|} = \sqrt{(1+\delta)^2 - p_y^2} \tag{8.72}$$

which is independent of s. Using this, (radial) entrance angle θ_x and the exit angle $\theta_x(s)$ are given in terms of momentum components by

$$\sin\theta_x = \frac{p_x}{p_{\|}}, \quad \sin\theta_x(s) = \frac{p_x(s)}{p_{\|}}. \tag{8.73}$$

The coordinate s, arc length along the design orbit, satisfies

$$\Phi = -\frac{s}{\rho_c}, \tag{8.74}$$

where ρ_c is the design bend radius. As shown in the figure, the exit momentum components, expressed in the appropriate local exit frame, are

$$p_x(s), \quad p_y(s) = p_y, \quad p_s(s) = \sqrt{(1+\delta)^2 - p_y^2 - p_x^2(s)}. \tag{8.75}$$

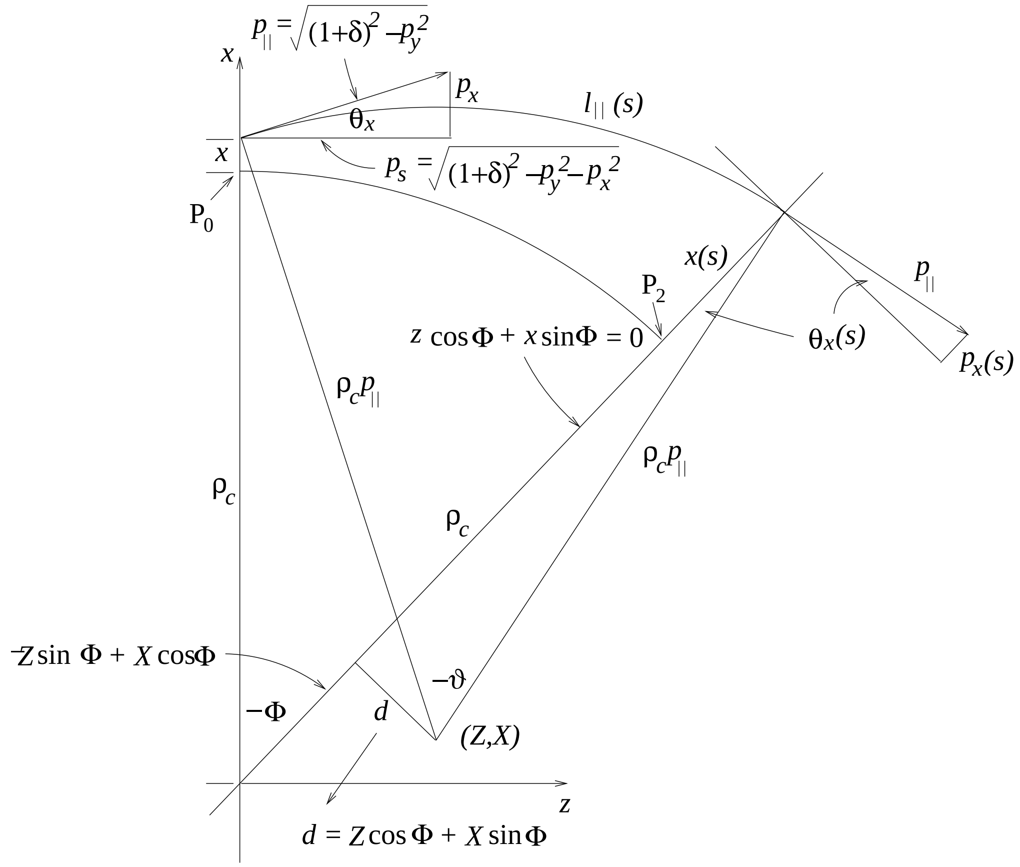


Figure 8.4. Shown along with the central trajectory is the projection of a general trajectory onto the midplane of a sector bend of nominal bend angle $-\Phi$. With the total particle momentum being $(1+\delta)p_0$, where p_0 is the central momentum, the scaled momentum is $1+\delta$. (A vector of magnitude p_0 would have unit length in the figure.) Since vertical and horizontal motion is uncoupled it is valid to regard the projected curve as the trajectory of a particle actually lying in the midplane of the magnet but having momentum $p_{\|} = \sqrt{(1+\delta)^2 - p_y^2}$. This quantity is conserved along the orbit. The radial coordinates at the magnet faces are displacement x and (in units of p_0) radial momentum p_x.

Expressed in the entrance frame coordinates, with origin at the apex of the sector bend, the coordinates (Z, X) of the center of curvature of the displaced trajectory are

$$(Z,\ X) = (\rho_c p_x,\ x + \rho_c(1 - p_s)), \tag{8.76}$$

and the equation of the exit face of the sector bend is

$$z\cos\Phi + x\sin\Phi = 0. \tag{8.77}$$

The perpendicular (directed) distance from the center of curvature to this line is

$$d = Z\cos\Phi + X\sin\Phi. \tag{8.78}$$

From the right triangle with vertex at (Z, X) we get output angle $\theta_x(s)$ to be given by

$$\sin\theta_x(s) = \frac{d}{\rho_c p_{\|}}. \tag{8.79}$$

In view of the second of Eqs. (8.73), after simplification, this gives the locally-radial output momentum component to be

$$p_x(s) = p_x\cos\Phi + \Big(\frac{x}{\rho_c} + 1 - p_s\Big)\sin\Phi. \tag{8.80}$$

Notice that this has automatically re-expressed the radial momentum in the local Frenet frame of reference appropriate at the exit face of the magnet. This is Forest's Eq. (12.18.b). It meets the exactness requirement as well as the TPS description requirement by giving one output coordinate as an explicit analytic function of the input coordinates. Subsequent formulas can be regarded as explicit even if they depend on $p_x(s)$. In particular, $p_s(s)$ is given by the third of Eqs. (8.75). As has been implicit in the discussion so far, the fractional momentum maps according to $\delta(s) = \delta$.

The actual arc length $l(s)$ and the visible-in-figure arc length $l_{\|}(s)$ are related by $l_{\|}(s)/l(s) = p_{\|}/(1+\delta)$. The total bend angle ϑ can be determined either in terms of $l_{\|}(s)$ and radius of curvature or in terms of the angles visible in the figure;

$$\frac{l(s)}{\rho_c(1+\delta)} = \frac{l_{\|}(s)}{\rho_c p_{\|}} = -\vartheta = -\Phi - \theta_x(s) + \theta_x. \tag{8.81}$$

Solving for $l(s)$ yields

$$l(s) = \rho_c(1+\delta)\Big(-\Phi - \sin^{-1}\frac{p_x(s)}{p_{\|}} + \sin^{-1}\frac{p_x}{p_{\|}}\Big). \tag{8.82}$$

Vertical evolution is given similarly by

$$y(s) = y + p_y\rho_c\Big(-\Phi - \sin^{-1}\frac{p_x(s)}{p_{\|}} + \sin^{-1}\frac{p_x}{p_{\|}}\Big). \tag{8.83}$$

All that remains is to determine $x(s)$ using the same output right triangle;

$$\rho_c + x(s) = \rho_c p_{\|}\,\cos\theta_x(s) + Z\sin(-\Phi) + X\cos\Phi. \tag{8.84}$$

Re-expressed, this becomes

$$x(s) = \rho_c\Big(-1 + p_s(s) - p_x\sin\Phi + \big(\frac{x}{\rho_c} + 1 - p_s\big)\cos\Phi\Big). \tag{8.85}$$

All of the needed output coordinates have now been obtained.

8.7. Identifying Sources of Nonlinearity by Spectral Analysis

This section can be regarded as an extension, to nonlinear motion, of chapter 4, in which analysis of diagnostic instrumentation was discussed. The present emphasis is on identifying sources of nonlinearity. The idea is to correlate peaks observed in spectra, derived from multiturn BPM output, with the sources which could produce them. These spectra can be obtained using a hardware spectrum analyser or by Fourier transformation of digitized BPM turn-by-turn data.

TABLE 8.3. Fourier Expansions and Labels for Nonlinear Motion

$$
\begin{aligned}
\Big|^{m_x}_{m_y} &\equiv \cos((m_x\nu_x + m_y\nu_y)2\pi t), \quad \Big|^{m_x}_{\pm m_y} \equiv \Big|^{m_x}_{m_y} + \Big|^{m_x}_{-m_y}, \quad 1 = \Big|^{0}_{0}\\
x &= a_x\cos(2\pi\nu_x t) = a_x\Big|^{1}_{0}\\
y &= a_y\cos(2\pi\nu_y t) = a_y\Big|^{0}_{1}\\
x^2 - y^2 &= \frac{a_x^2}{2}\Big|^{2}_{0} - \frac{a_y^2}{2}\Big|^{0}_{2} + \frac{a_x^2 - a_y^2}{2}\Big|^{0}_{0}\\
2xy &= a_x a_y\Big|^{1}_{\pm 1}\\
x^3 - 3xy^2 &= \frac{a_x^3}{4}\Big|^{3}_{0} - \frac{3a_x a_y^2}{4}\Big|^{1}_{\pm 2} + \frac{3a_x^3 - 6a_x a_y^2}{4}\Big|^{1}_{0}\\
3x^2y - y^3 &= -\frac{a_y^3}{4}\Big|^{0}_{3} + \frac{3a_x^2 a_y}{4}\Big|^{2}_{\pm 1} - \frac{3a_y^3 - 6a_x^2 a_y}{4}\Big|^{0}_{1}\\
x^4 - 6x^2y^2 + y^4 &= a_x^4\Big|^{4}_{0} - \frac{6a_x^2 a_y^2}{8}\Big|^{2}_{\pm 2} + \frac{a_y^4}{4}\Big|^{0}_{4}\\
&\quad + \frac{4a_x^4 - 12a_x^2 a_y^2}{8}\Big|^{2}_{0} + \frac{4a_y^4 - 12a_x^2 a_y^2}{8}\Big|^{0}_{2} + \frac{3a_x^4 - 12a_x^2 a_y^2 + 3a_y^4}{8}\Big|^{0}_{0}\\
4x^3y - 4xy^3 &= \frac{4a_x^3 a_y}{8}\Big|^{3} \pm 1 - \frac{4a_x a_y^3}{8}\Big|^{1}_{\pm 3} + \frac{12a_x^3 a_y - 12a_x a_y^3}{8}\Big|^{1}_{\pm 1}
\end{aligned}
\tag{8.86}
$$

For sufficiently small amplitudes all nonlinear terms become negligible and the motion is described approximately by the pure (uncoupled by preference) betatron motions $x_t = a_x\cos(2\pi\nu_x t)$ and $y_t = a_y\cos(2\pi\nu_y t)$ for $t = 0, 1, \dots$. These "fundamental oscillations" can be regarded as the "zero'th" approximation to the motion. These time dependencies, and the time dependencies they produce when nonlinear elements are present, are shown in Table 8.3. Like quantum numbers in spectroscopy, integers m_x and m_y can be used to label observed lines. The notation $\Big|^{m_x} m_y$ in the table, having x-harmonic number m_x on top and y-harmonic number m_y on the bottom, is intended to help in assigning labels to spectral lines. When longitudinal motion is included another, similar index, m_s, is introduced.

A nonlinear element can be treated perturbatively, with the deflection it causes proportional to a quadratic or higher power of the fundamental oscillations amplitudes. Since the small amplitude motion is harmonic these extra contributions are periodic with the same periods, and they can be expessed as "nonlinear harmonics", also known as "higher harmonic frequencies" of the fundamental oscillations. Exploiting this feature, a sextupole deflection proportional to x^2 appears as a double-frequency motion—though with aliasing this line may appear at an initially-unexpected location. Octupole deflections proportional to x^3 cause tripled (and other) lines. Other nonlinear deflections, for example proportional to xy, can also cause "mixing" (sum and difference frequencies).

The power of Fourier expansion is to "linearize" this weakly nonlinear motion. In lowest order of perturbation theory, a nonlinear element contributes as if linear, but driven at sum and difference frequencies of the fundamental frequencies. Higher order perturbation theory can be described by iterating this procedure, but the possible harmonics proliferate badly, and the spectra rapidly become uniterpretable as the basic amplitudes increase. Eventually the motion becomes chaotic.

There is one case in which proceeding to second order perturbation theory is justified. It is the (quite common) case in which a nonlinear element has no effect in lowest order. An example of this is the absence of tune shift due to sextupoles in lowest order. To calculate amplitude-dependent tune shifts it is necessary to include octupoles in first order and sextupoles in second order.

Observing which peaks are present, and with what strength, and then correlating with Fourier expansions of particular multipole fields can give clues as to which fields are causing the motion to be nonlinear. Conversely the importance of nonlinearities known (from magnetic measurements) can be assessed. For interpreting spectra in this way it is necessary to write Fourier expansions of the multipole expressions for deflections $\Delta x'$ and $\Delta y'$ appearing in Table 8.1. The following formulas are needed for those expansions:

Since the factors a_x and a_y are presumeably, in some sense, "small", the dominant lines tend to be those having minimal powers of these factors.

The convergence (*i.e.* the extent to which succeeding terms become less important) of the multipole series as a formula for magnetic field at displacement a_x can be assessed by the numerical value of ratios $(b^{n+1}a_x^{n+1})/(b^n a_x^n) = (b^{n+1}/b^n)a_x$. But to estimate the absolute influence on accelerator performance of a particular multipole an estimator like that calculated in Problem 8.10 is needed. In practice, as mentioned before, there is a strong tendency for lower powers of a_n and b_n to dominate.

Example spectra extracted for the toy lattice `general_fodo_rf` are shown in FIG 8.5. These plots were obtained by using the graphics program `xmgrace` to post-process turn-by-turn data produced by UAL. (This uses the PERL interface that is being in the process of being phased out.) The code is run by

```
$ cd /home/ualusr/USPAS/ual1/examples/UI_Xmgr
$ perl shell_USPAS.pl
```

The lower three graphs show the horizontal, vertical and longitudinal turn-by-turn data and the upper graphs are the corresponding tune spectra. Low order tune lines are shown, labeled with triplets (m_x, m_y, m_z). The fundamental lines are shown by solid vertical lines. They are located by first finding the maxima in the three spectra. From these tunes, tune combinations are formed as $m_x\nu_x + m_y\nu_y + m_s\nu_s$, where m_x, m_y, and m_s are small integers. Furthermore the tune lines have to be "unaliased" into the range from 0 to 0.5. (This amounts to finding the absolute deviation from the nearest integer.)

Some features that can be observed in these spectra are:

- The fundamental tunes are $Q_x = 0.26$, $Q_y = 0.28$, $Q_s = 0.08$.
- There is a strong $1, 0, 0$ signal visible in the longitudinal spectum. This is evidence of coupling between horizontal and longitudinal.
- A reciprocal $0, 0, 1$ line is visible in the horizontal spectrum.
- The vertical signal is almost pure harmonic. Absence of $1, 0, 0$ signal implies the absense of x, y coupling.
- There is a weak $2, 0, 0$ line in the horizontal spectrum. This is due to the presence of chromaticity sextupoles.
- The line $1, -1, 0$ in the vertical spectrum comes, presumably, from the same source.

Similar plots can be obtained using the UAL player.

Table 8.4. Spectral lines in X-spectrum (horizontal) caused by particular multipoles. a_x and a_y are "fundamental" amplitudes. There are also numerical factors, of order one, not shown.

m_x	0	1	0	2	1	0	3	2	1	0	4	3	2	1	0
m_y	0	0	1	0	±1	2	0	±1	±2	3	0	±1	±2	±3	4
b_0	1														
a_0															
b_1		a_x													
a_1			a_y												
b_2	a_x^2, a_y^2			a_x^2		a_y^2									
a_2					$a_x a_y$										
b_3		$a_x^3, a_x a_y^2$					a_x^3		$a_x a_y^2$						
a_3			$a_x^2 a_y, a_y^3$					$a_x^2 a_y$		a_y^3					
b_4	$a_x^4, a_x^2 a_y^2, a_y^4$			$a_x^4, a_x^2 a_y^2$		$a_x^2 a_y^2, a_y^4$					a_x^4		a_x^2, a_y^2		a_y^4
a_4					$a_x^3 a_y, a_x a_y^3$							$a_x^3 a_y$		$a_x a_y^3$	

Table 8.5. Spectral lines in Y-spectrum (vertical) caused by particular multipoles. a_x and a_y are "fundamental" amplitudes. There are also numerical factors, of order one, not shown.

m_x	0	1	0	2	1	0	3	2	1	0	4	3	2	1	0
m_y	0	0	1	0	±1	2	0	±1	±2	3	0	±1	±2	±3	4
b_0															
a_0	1														
b_1			a_y												
a_1		a_x													
b_2					$a_x a_y$										
a_2	a_x^2, a_y^2			a_x^2		a_y^2									
b_3			$a_x^2 a_y, a_y^3$					$a_x^2 a_y$		a_y^3					
a_3		$a_x^3, a_x a_y^2$					a_x^3		$a_x a_y^2$						
b_4					$a_x^3 a_y, a_x a_y^3$							$a_x^3 a_y$		$a_x a_y^3$	
a_4	$a_x^4, a_x^2 a_y^2, a_y^4$			$a_x^4, a_x^2 a_y^2$		$a_x^2 a_y^2, a_y^4$					a_x^4		a_x^2, a_y^2		a_y^4

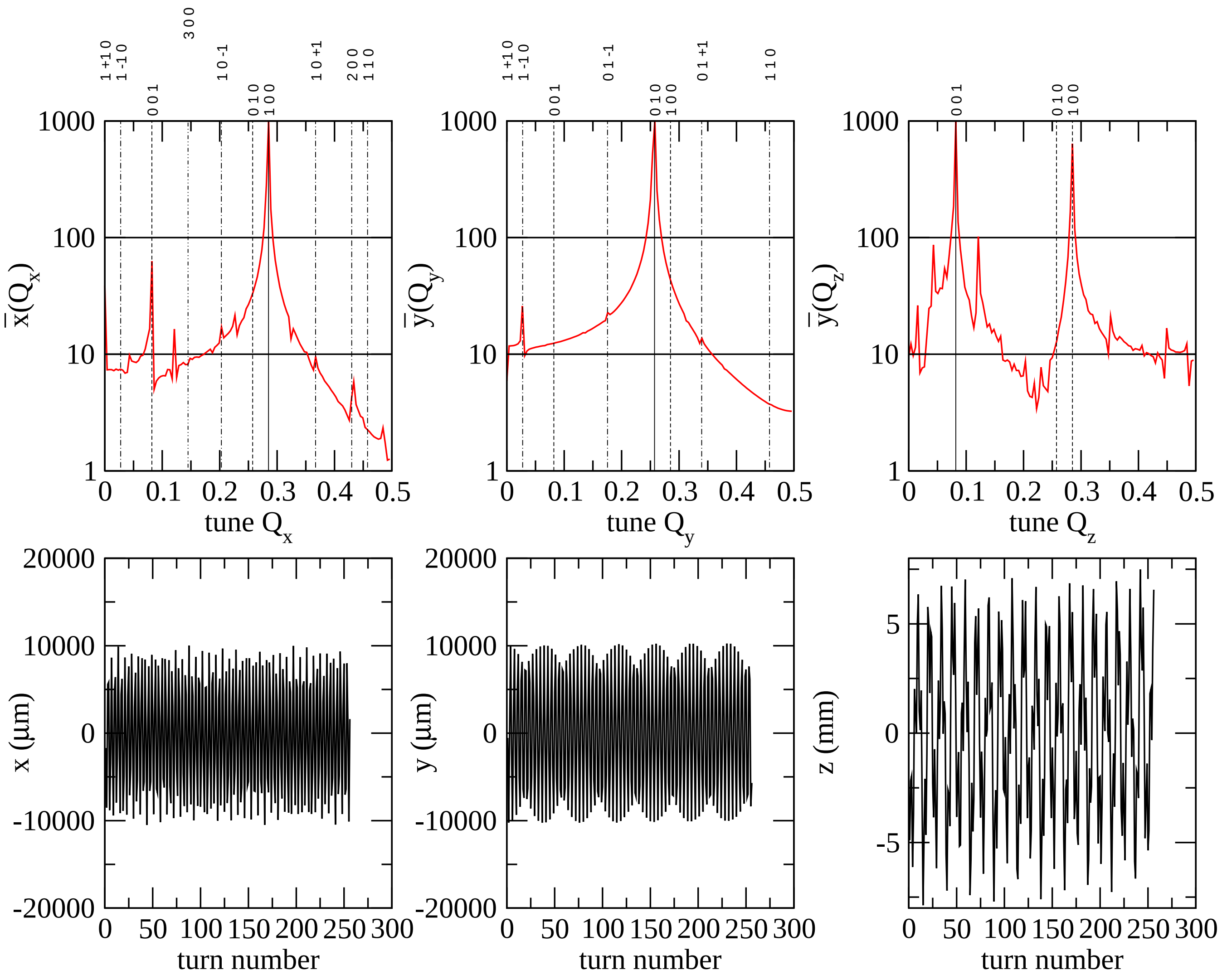


FIGURE 8.5. Turn-by-turn data, spectrum analysed, for collider_rf lattice. Fractional tunes are $Q_x = 0.26$, $Q_y = 0.28$, $Q_s = 0.08$. Low order tune lines are labeled with m_x, m_y, and m_s indicies.

PROBLEM 8.8. *Show that the effect of closed orbit displacements $\underline{\Delta x}$ and $\underline{\Delta y}$ are to produce "feed-down" such that the presence of multipole coefficients b_n and a_n leads to multipoles*

$$b_{n-1}^{(\underline{\Delta x})} = -nb_n\underline{\Delta x}, \quad a_{n-1}^{(\underline{\Delta x})} = -na_n\underline{\Delta x}, \quad b_{n-1}^{(\underline{\Delta y})} = na_n\underline{\Delta y}, \quad a_{n-1}^{(\underline{\Delta x})} = -nb_n\underline{\Delta y}. \tag{8.87}$$

PROBLEM 8.9. *The Courant-Snyder invariant ϵ of a particle executing one dimensional betatron oscillations is given by $\gamma_x x^2 + 2\alpha_x xx' + \beta_x {x'}^2$. With proper axis-scaling, if the motion is linear, the point in phase space with coordinates (x, x') lies on a circle and rotates at uniform rate. The effect of a deflection $x' \to x' + \Delta x'$ will sometimes be to increase ϵ and sometimes to decrease it. Show however that on the average there is a net increase given by*

$$< \Delta\epsilon >= \beta_x(\Delta x')^2. \tag{8.88}$$

PROBLEM 8.10. *Consider the one dimensional motion of a particle with amplitude a_x through a bending element that bends the central trajectory through angle $\Delta\theta$. The field nonuniformity of the magnet is described by a multipole coefficient b_n. Show that he deflection suffered is $b_n a_x^n \Delta\theta$ times an oscillatory factor in the range from -1 and 1. Continuing to drop a numerical factor of order 1, show that the average fractional increase in the Courant-Snyder invariant in passing through the magnet is*

$$\frac{< \Delta\epsilon >}{\epsilon} \sim (b_n a_x^{n-1}\beta_x\Delta\theta)^2 \sim (b_n\Delta\theta)^2 \epsilon_x^{\frac{n-1}{2}} \beta_x^{\frac{n+1}{2}} N_x^{n-1}. \tag{8.89}$$

where a_x is quoted as $N\sigma_x$ where ϵ_x is the horizontal beam emittance and $\sigma_x = \sqrt{\epsilon_x\beta_x}$ is the r.m.s. horizontal beam size. This formula is not valid for $n = 1$. Why not?

PROBLEM 8.11. *Check all (or at least many of) entries in Table 8.4.*

PROBLEM 8.12. *An erect quadrupole is misaligned from its design orientation by a small roll angle $\Delta\phi << 1$ around the longitudinal axis. In its natural (x', y') coordinates, the quadrupoles multipole expansion is*

$$B_{y'} + iB_{x'} = b_1'(x' + iy'), \tag{8.90}$$

and in the design lattice coordinates it is

$$B_y + iB_x = (b_1 + ia_1)(x + iy). \tag{8.91}$$

Show, to lowest order in $\Delta\phi$, that

$$b_1 = b_1', \quad a_1 = -2b_1'\Delta\phi. \tag{8.92}$$

In modeling the effect of a misaligned quad the factor of 2 in Eq. (8.92) must not be overlooked. A mnemonic for remembering this factor is that a quadrupole need only be rotated through angle $\pi/4$ (not $\pi/2$) for pure b_1 to become pure a_1.

PROBLEM 8.13. *Perform the numerical calculations (accounting for aliasing if necessary) to account for the locations of spectral lines labeled 200, 001, 10-1, and 10+1, in the Q_x plot of FIG 8.5.*

CHAPTER 9

Colliding Beams

9.1. The collider.adxf lattice

The collider.adxf lattice is derived from the general_fodo lattice, via the racetrack lattice. The long straight sections of racetrack are replaced by "low beta optics" appropriate for achieving maximum luminosity from given beam currents. The elements making up the irtoarc transition section are shown in FIG 9.1, and their parameter values are listed next; the initial numerical values are such that lfac=1.0 m.

```
<constant name="lfac" value="scale*20*nhalf/100"/>
<constant name="l01" value="1.397753023209*lfac"/>
<constant name="l12" value="1.497222048429*lfac"/>
<constant name="l23" value="12.177703590031*lfac"/>
<constant name="l34" value="5.0*lfac"/>
<constant name="l45" value="5.0*lfac"/>
<constant name="l56" value="5.0*lfac"/>
<constant name="lqir" value="0.001"/>
<constant name="qir1" value="-1.05/lfac"/>
<constant name="qir2" value="0.5494/lfac"/>
<constant name="qir3" value="-0.33/lfac"/>
<constant name="qir4" value="0.1837/lfac"/>
<constant name="qir5" value="-0.2389/lfac"/>
<constant name="qir6" value="0.1194/lfac"/>
<sector name="irtoarc">
    <frame ref="mkst"/>
    <frame ref="dr01"/>
    <frame ref="quadir1"/>
    <frame ref="dr12"/>
    <frame ref="quadir2"/>
    <frame ref="dr23"/>
    <frame ref="quadir3"/>
    <frame ref="dr34"/>
```

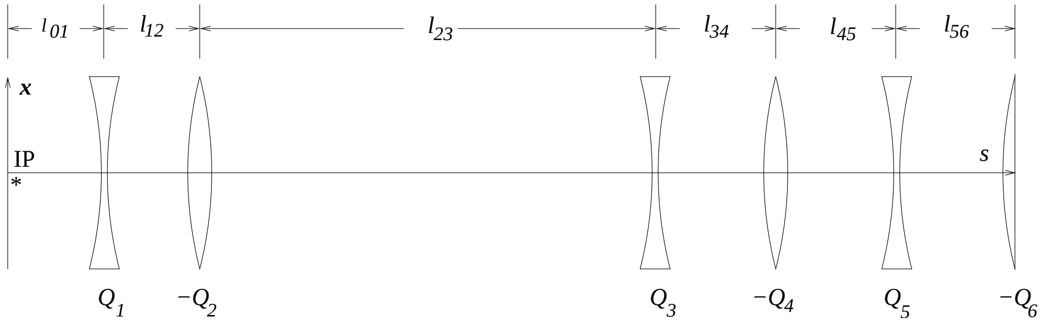


FIGURE 9.1. Lengths and strengths of intersection region optical elements. Lens symbols and quad strengths appropriate for horizontal (convex is focusing) are shown and all Q_i are positive.

```
        <frame ref="quadir4"/>
        <frame ref="dr45"/>
        <frame ref="quadir5"/>
        <frame ref="dr56"/>
        <frame ref="quadir6"/>
    </sector>
```

Not shown in this listing are the regular arc quadrupole half-strengths; $q1 = 0.1194$ and $q2 = -0.1199$. The lattice functions for the lattice with these values are shown in FIGs 9.2, 9.3, and 9.4. One sees from FIG 9.2 that the IR region is well matched to the arcs (which are the same as in the `racetrack` lattice). But the presence of small beta functions at the intersection point (IP) cause high beta values nearby. This follows inexorably, as can be seen, for example, from Table 2 in Chapter 2. Since the lattice is mirror symmetric about the IP's, $\alpha_*^x = \alpha_*^y = 0$ at the IP. Here the use of asterisk to specify IP is a traditional notation. It then follows from Eq. (3.4) that

$$\beta_1^x = \beta_*^x + \frac{l_{01}^2}{\beta_*^x}, \quad \beta_1^y = \beta_*^y + \frac{l_{01}^2}{\beta_*^y}. \tag{9.1}$$

The optics has been adjusted so that $\beta_*^y << \beta_*^x$. Eqs. (9.1) then shows that β_1^y increases much more rapidly than β_1^x, as s increases from zero. This requires the first quad, Q_1, to be vertically focusing, as shown. (The concave/convex symbols refers to horizontal focusing character in this figure, and all Q_i are taken to be positive.) FIG 9.4 confirms these features.

Quadrupole lens optics is too complicated to be taken lightly, as you will discover if you start changing parameters recklessly, but certain features of a beamline like this are subject to quite simple treatment. One good plan is to work with "doublets" which, together, behave something like glass lenses by having the same character (either focusing or defocusing) in both x and y planes.

The dominant visual feature of these IR optics is that l_{23} is much greater than either l_{01} or l_{12}. This, plus the match to the FODO arc optics, is what leads to small beta functions at the IP. There is a "beam waist" at the origin. This is as close to a point focus as ever occurs in lattice optics. Basically the Q_1, Q_2 doublet focuses rays emerging from the origin to an image somewhere in the vicinity of Q_3 or beyond. With l_{23} assumed large, we may as well assume that the Q_1, Q_2 doublet focuses "at infinity". In other words we want this doublet to produce *point to parallel* focusing.

PROBLEM 9.1. *Let* $\mathbf{M^x}$ *and* $\mathbf{M^y}$ *stand for the* 2×2 *matrices governing propagation from the origin to point 2. Show that, for point to parallel focusing, we need* $M_{22}^x = M_{22}^y = 0$. *Work out these matrix elements and then, assuming* l_{01} *and* l_{12} *are known, show that the doublet quadrupole strengths are*

$$Q_1 = \frac{1}{l_{01}}\sqrt{\frac{l_{01}+l_{12}}{l_{12}}}, \quad Q_2 = \sqrt{\frac{1}{(l_{01}+l_{12})l_{12}}}. \tag{9.2}$$

Finally, show that the entries in the `collider` *lattice are in rough agreement with your values; (actually somewhat greater in magnitude because we don't really want the focus as far as infinity.)*

In practice l_{01} is chosen to be as short as possible, consistent with fitting the required particle detector into that drift, and allowing for Q_1 itself to use some of the space. Typically Q_2 has design similar to Q_1 and is butted right up against Q_1, effectively fixing $l_{12} \approx l_{01}$.

Commonly it is the large value of β_1^y that fixes the minimum achievable value of β_*^y. Viewing FIG 9.4 let us assume that the optics has been adjusted to satisfy the constraint $\beta_1^y < \beta^{\max} = 200\,\mathrm{m}$. From Eq. (9.1), treating the leading term as negligible, we therefore have

$$\beta_*^y > \frac{l_{01}^2}{\beta^{\max}}. \tag{9.3}$$

Though quad Q_1 restrains β^y, it causes β^x to increase even more rapidly in the region from Q_1 to Q_2. This increase is stopped by Q_2 but, typically, not before β^x has increased until it is equal, or almost equal, to $\beta^{\max}$. For simplicity, we take this to be the same as the maximum tolerable vertical value. (i.e. $\beta_2^x \approx 200\,\mathrm{m}$.)

PROBLEM 9.2. *With Q_1 and Q_2 given by Eqs. (9.2), find the value of β_*^x such that $\beta_2^x = 200$ m, under the (reasonable) requirement that the maximum of β^x occurs at Q_2. FIG 9.4 shows this assumption is quite accurately correct, but you cannot expect comparably good agreement for the value of β_*^x since other approximations have been made.*

FIG 9.4 shows that the optics is roughly matched to the arc optics already at Q_4. In fact, to make the matching clearer, Q_4 could be treated as two thin lenses Q_{41} and Q_{42}, butted together, with $Q_{42} = q_1$. With this observation, and noting that $Q_5 = 2q_2$ and $q_6 = q_1$, one sees that the optics are quite accurately matched already at a point "part way through" Q_4.

What this has meant is that, with all lengths fixed, as well as strengths Q_1 and Q_2, the only adjustable parameters left are Q_3 and Q_{41}. All Twiss functions are constrained (by the need to match the arc optics) at the boundary between Q_{41} and Q_{42}. The other constraints that must be met are $\alpha_*^x = \alpha_*^y = 0$. These two equations fix Q_3 and Q_{41}.

If all lengths are held fixed, by the arguments given so far, the entire optics is fixed, at least approximately. There is little point, therefore, in trying to adjust the IP beta functions by just altering quadrupole strengths for the simple IR `collider` configuration. It is typically difficult (but not impossible, using rails,) to change drift lengths in a storage ring. From the arguments given, one sees that this makes it quite difficult to change the beta-star values using the `collider` intersection region optics.

Of course one can vary all lengths in the entire ring. By equations like Eq. (3.17), scaling all lengths up and all quadrupole strengths down in the same ratio, will leave the optics, including the IR optics, matched. The lengths β_*^x and β_*^y would therefore be reduced in the same ratio. But one never has the luxury of scaling the circumference of a storage ring, so this alteration is impractical.

PROBLEM 9.3. *In spite of its impracticality, scale all lengths in the* `collider` *lattice, and scale all quad strengths by the inverse factor. Use the simulation code to determine the new optics and confirm the scaling behavior described in the previous paragraph.*

If one *must* change beta-star values, the most promising procedure is to alter l_{01} and l_{12} by the same (preferably fairly close to 1) factor and preserving ring geometry by altering l_{23}. For simplicity, let all other lengths remain unchanged.

By Eqs. (6) Q_1 and Q_2 scale inversely with l_{01} and l_{12}. Then it should be possible to restore the match by adjusting Q_3 and Q_4 using $\alpha^x_* = \alpha^y_* = 0$.

By varying the ratio l_{01}/l_{12} it would be possible to vary the ratio β^y_*/β^x_*. But, as explained earlier the use of "doublet optics" forces this ratio to be large. Switching the signs of Q_1 and Q_2 would produce $\beta^x_* << \beta^y_*$.

PROBLEM 9.4. *Using the UAL simulation code, determine accurate values for β^x_* and β^y_* and compare your values with values calculated by formulas in this section.*

The optics that has been described is typical of electron accelerators, where the vertical emittance is already much less than the horizontal emittance rendering the beams ribbon-shaped. In hadron accelerators the transverse beam emittances are normally approximately equal and there is a luminosity advantage in having the beams approximately round at the IP. This can only be achieved by using "triplet optics", more complicated than can be achieved with the `collider` toy lattice.

Just replacing the doublet Q_1, Q_2 by a triplet, most of the preceeding arguments will remain roughly true, and more-or-less equal betas at the IP can be achieved. Though $\beta^y{*} \approx 1\,\mathrm{cm}$, is practical with doublet optics in an electron accelerator, $\beta^x{*} \approx \beta^y{*} \approx 1\,\mathrm{m}$ is more typical of hadron colliders. This is partly due to the inherently longer focal lengths achievable using triplet optics and partly due to the much higher particle momenta (and hence longer quadrupoles) in hadron accelerators.

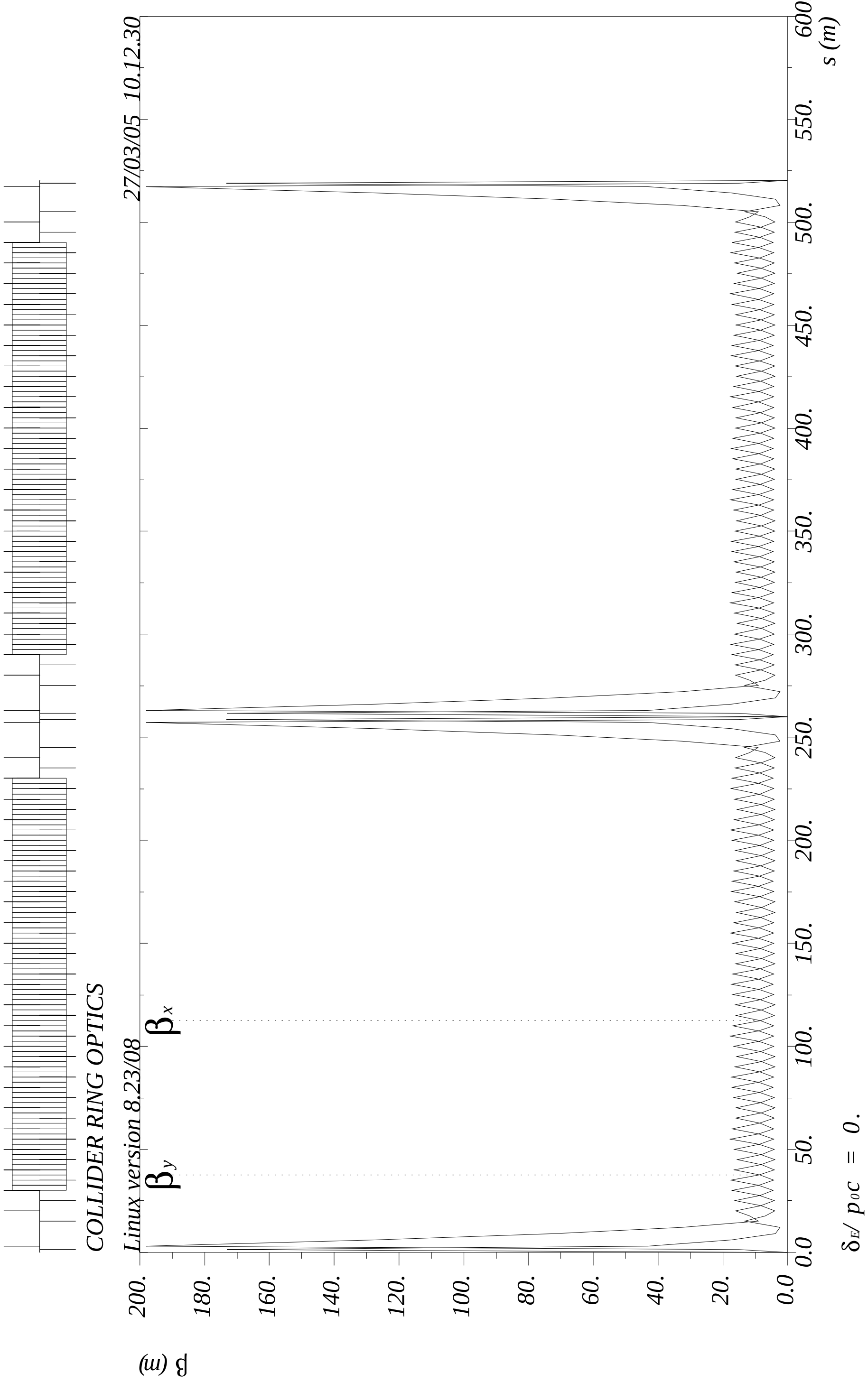


FIGURE 9.2. Twiss functions for the `collider` lattice.

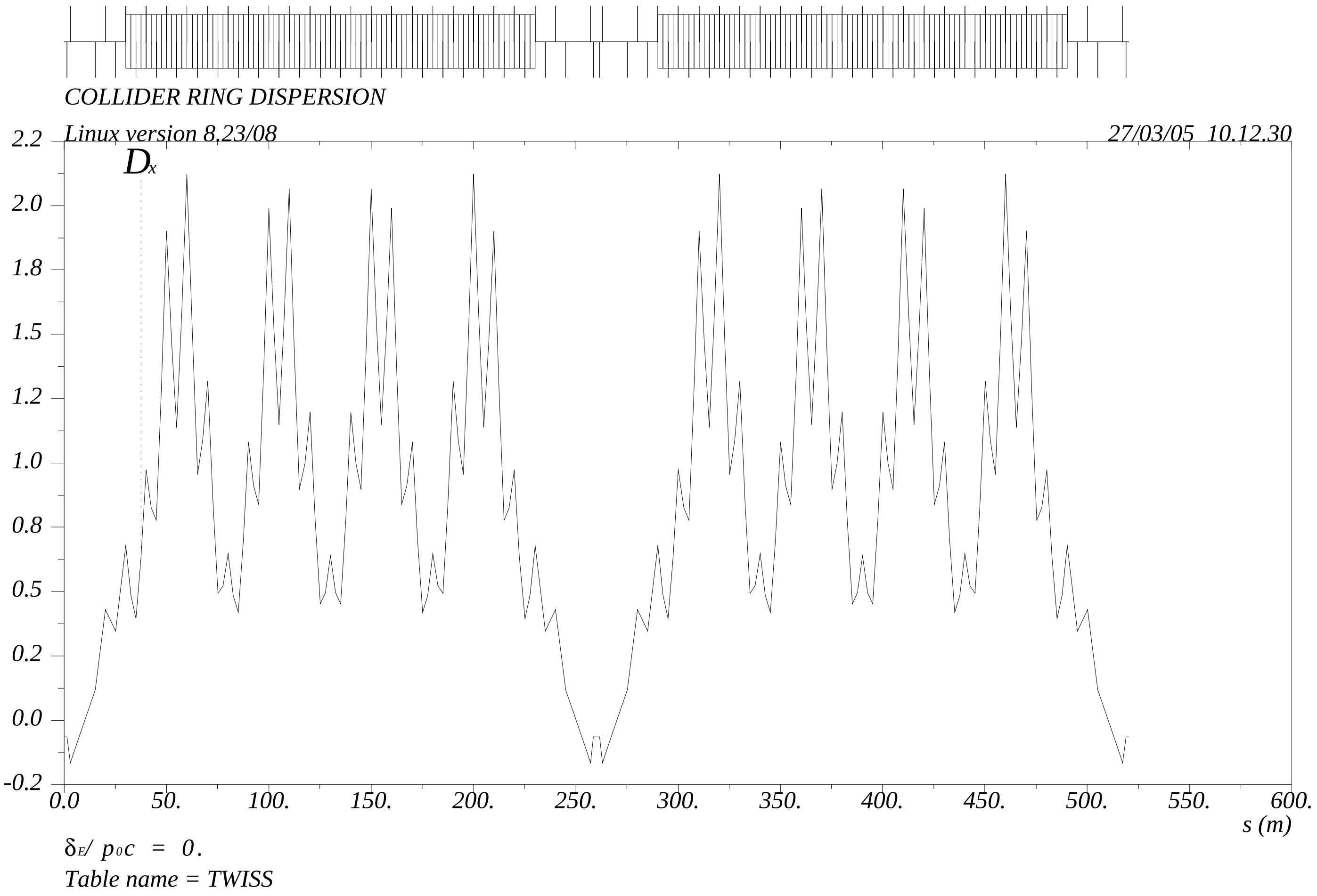


FIGURE 9.3. Dispersion function for the `collider` lattice.

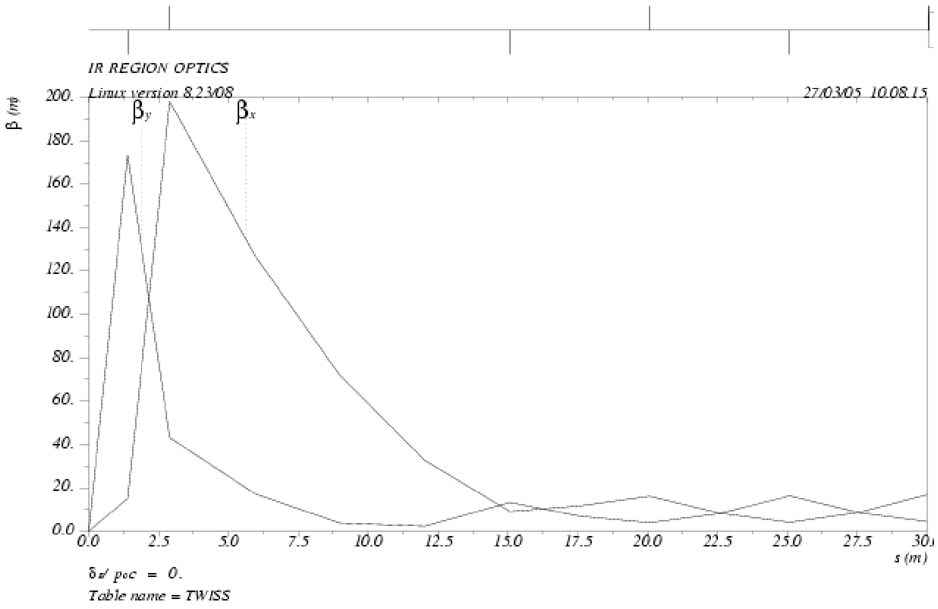


FIGURE 9.4. Twiss functions for the IR region of the `collider` lattice.

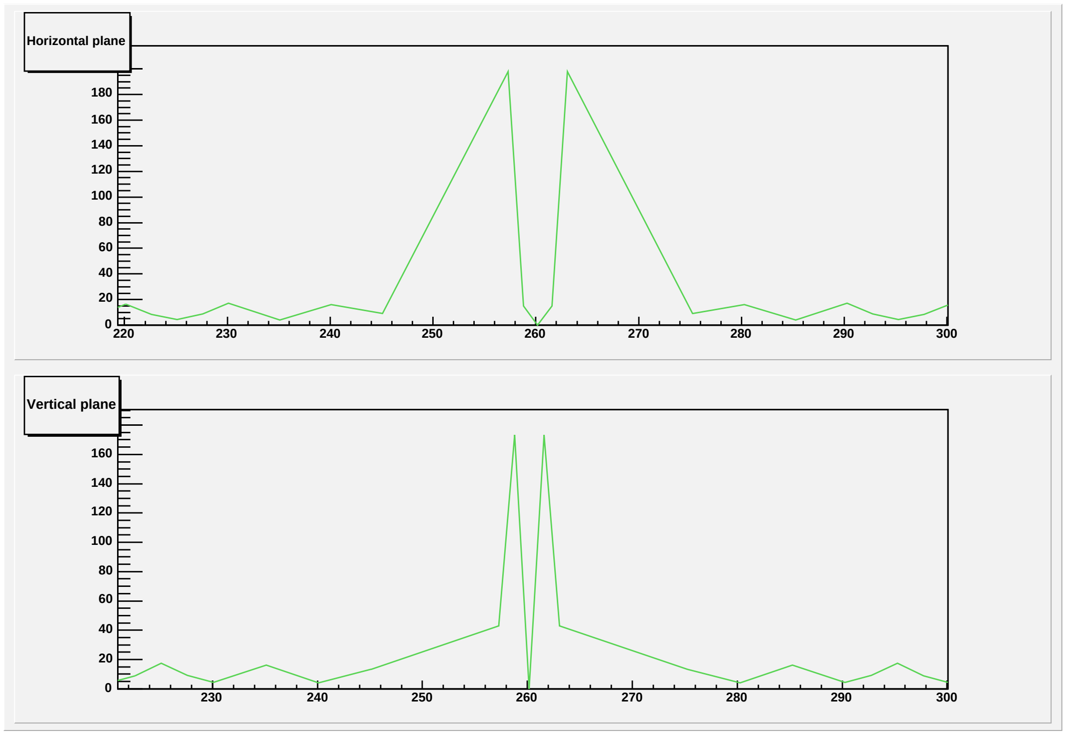


FIGURE 9.5. UAL-generated Twiss functions for the IR region of the `collider` lattice.

This remainder of this chapter is rather more technical than is appropriate for the USPAS accelerator simulation course. It is more appropriate as a section of the Physics User's Manual. *A beam-beam test program* `bmbm.cpp` *is appended as a convenient way to convey various numerical coefficients. A tar file with consistent README, data and makefile is available for actual use of the code.*

As of March 2005, the elements `headonbb` *and* `remotebb` *have not yet, in fact, been implemented in* UAL. *As a result, parasitic beam-beam interactions cannot be concatenated into nonlinear* TPS *lattice maps.*

9.2. Particle Deflection Caused by Oncoming Bunch

The deflection of a charged particle caused by an approaching charged bunch can be modeled as deflection by an artificial `beambeam` element that is located where the bunches meet and acts much like any physical lattice element. The oncoming beam is assumed to be Gaussian in all six phase space coordinates.

In practice the bunches are long (z coordinate) compared to their transverse (x,y) components. Invariably both bunches are fully relativistic. It is well known in this circumstance that the electromagnetic fields of the bunch are compressed longitudinally to become almost exactly transverse. This makes it appropriate to treat the fields as purely transverse and derivable from a charge distribution with shape dependent only on transverse coordinates and magnitude proportional to the longitudinal bunch distribution.

To a lowest approximation the beam-beam interaction can be treated as a single transverse impulse at the crossing point. More accurate, and especially for non-zero crossing angle of long bunches, it is appropriate to "slice" the bunch longitudinally and to represent each bunch crossing by several, londitudinally-displaced deflections. Even in this case, it is a good approximation to assume transverse fields that are independent of z except for being modulated by the longitudinal bunch profile. Beam-beam interactions are modeled to both of these levels of approximation in UAL. For even greater accuracy, especially in situations where the beam-beam interaction excites synchrobetatron oscillations, it would be appropriate to include longitudinal forces, but this level of approximation is not supported at this time.

In the purely-transverse, fully-relativistic, field approximation, the electric and magnetic fields are related exactly as are the fields in a plane electromagnetic wave. (Weisszacker-Williams approximation.) For a 2D Gaussian transverse charge distribution the fields can be expressed as an analytic function of the (complexified) transverse position $z = x+iy$; the function is variant $W(z)$ of the so-called complex error function. In UAL (as in most other simulations) this function is evaluated by various nonpolynomial numerical algorithms for evaluation of the complex error function.[**21**].

Particle dynamics can be handled *polymorphically* in UAL. What this means is that (optionally) the particle coordinates being evolved can be treated as truncated power series (TPS). It might be thought automatic to treat an analytic function like the complex error function by a TPS. As it happens this is is diametrically incorrect. The inappropriateness manifests itself mathematically, physically, and computationally. Mathematically, though analytic in some regions, $W(z)$ is *not* analytic at the $z = x + iy = 0$ origin, which is precisely where a TPS *has to be* a differentiable function of x and y individually. Physically, the beam-beam fields fall off (proportional to $1/r$) at large radial radius, while a TPS, truncated to any finite order, blows up at large r. Computationally the complex plane has to be segmented into more than one patch, with a separate evaluation algorithm applying in each patch. The computer program evaluating $w(z)$ starts by selecting the correct patch and proceeds by applying the algorithm appropriate for this patch. This is very satisfactory for purely numerical evaluation, but limits the validity of concatenation to such a restricted patch as to be useless for multiturn simulation.

Though beam-beam deflections can be represented differentiably (as required by TPS) in local patches, for example near the origin, the nonanalytic nature of the electric field excludes the possibility of a patch of size even as great as the smaller of the strong beam's transverse r.m.s. sizes.

Because of this mismatch, for head-on beam-beam collisions, the beam-beam deflection cannot be handled polymorphically. This means that the beam-beam interaction map cannot be concatenated with a lattice TPS map. To exploit a (possibly nonlinear) map representing the lattice sector preceeding an IP, it is necessary to derive each particle's coordinates at the IP from the map, then to calculate and apply the beam-beam deflection, and finally to treat the result as input coordinates for the map representing the next sector.

For sufficiently non-head-on, parasitic, beam-beam interactions, the limitation just described is not operative. Here "sufficiently" means something like "several sigma". For such collisions the parasitic collisions can be concatenated into a lattice map much the way any other lattice element would be.

Because the nonanalytic problem is not fatal for sufficiently remote collision, it makes sense to introduce two distinct `beambeam` types, called `headonbb` and `remotebb`. The `remotebb` type can be concatenated polymorphically just like any other beam line element. The `headonbb` type can be subjected to near-linear map analysis, to obtain small amplitude tune shifts and even amplitude-dependent tune shifts, but the evaluation patch valid at the origin is already very inaccurate for transverse displacement amplitudes comparable with the opposing bunch size. Certainly the description of long term evolution by iterating a single map representing both lattice and beam-beam interaction is unphysical with a `headonbb` element in the ring. But there is nothing to prevent concatenation of any number of `remotebb` elements into a single nonlinear map for the sector from one `headonbb` element to the next.

9.3. Electric Field Due to Gaussian Charge Distribution

Bassetti and Erskine[**22**] give formulas for the electric field components at x, y of a Gaussian charge distribution, total charge per unit length λ, in terms of complex error function $w(z)$, where $z = x + iy$, The function $w(z)$ is related to the so-called "error function complement" erfc(z) by

$$w(z) = e^{-z^2}\mathrm{erfc}(-iz). \tag{9.4}$$

(Introduction of complex numbers into the discussion leads to an unfortunate clash of meanings for the word "real". Whether "real" means the real part of a complex number or a single decimal number, as in computer programming language, has to be inferred from the context.)

The electric field is given by

$$\begin{pmatrix} E_x \\ E_y \end{pmatrix} = \frac{\lambda}{2\epsilon_0\sqrt{\pi}}\,\frac{1}{s}\begin{pmatrix} \Im \\ \Re \end{pmatrix}\Big(w(\frac{x_w}{s} + i\,\frac{y_w}{s}) - e^{-(\frac{x_w^2}{2s_x^2} + \frac{y_w^2}{2s_y^2})}\,w(\frac{x_w}{s}\,\frac{s_y}{s_x} + i\,\frac{y_w}{s}\,\frac{s_x}{s_y})\Big), \tag{9.5}$$

where x_w and y_w are horizontal and vertical deviations of a particular particle in a "weak beam" from the center of an approaching "strong beam". Here we are using the conventional "strong/weak" terminology of colliding beams, where the distribution of the strong beam, i.e. the other beam, or the unperturbed beam, is treated as constant, at least until it is updated after multiple passes have been evaluated.

The horizontal and vertical strong beam "sigmas" are s_x and s_y and, to simplify the expression a bit, a modified transverse beam size s given by

$$s = \sqrt{2(s_x^2 - s_y^2)}, \tag{9.6}$$

has been introduced. So that s is real, the beam is assumed to be wider than it is tall, $s_x > s_y$, and s_x/s_y exceeds some nominal value, slightly greater than 1, such as 1.1. Since the "strong beam" aspect ratio depends on the local β-functions, the strong beam will normally be at least this much out of round even if the emittances in the two planes are equal.

At crossing points where the strong beam is nearly round a different approach is required. For an exactly round beam, using S.I. units, the electric field is

$$\begin{pmatrix} E_x \\ E_y \end{pmatrix} = \frac{\lambda(z)}{2\pi\epsilon_0} \frac{1 - e^{-r^2/(2s^2)}}{r^2} \begin{pmatrix} x \\ y \end{pmatrix}, \tag{9.7}$$

where $\lambda(z)$ is the longitudinal charge density. For a Gaussian-distributed, strong beam bunch with N_s charges Q_s,

$$\lambda(z) = \frac{N_s Q_s}{\sqrt{2\pi}\,\sigma_z} \exp\Big(-\frac{z^2}{2\sigma_z^2}\Big). \tag{9.8}$$

For long bunches, especially if they are relativistic, the longitudinal electric field can normally be neglected. Also the magnetic deflection is, except for factor v^2/c^2, the same as the electric deflection. The total deflection is therefore obtained by multiplying the electric deflection by $1 \pm v^2/c^2$ where the minus sign would be appropriate for the space charge force on a particle co-moving with the bunch. The positive sign is appropriate for the counter-traveling bunches assumed for `beambeam` elements. The total deflection suffered in passing the other bunch is proportional to its bunch length, and hence to its total charge. (While including the factor of nearly two to add electric and magnetic force one must not forget to include an almost-exactly-canceling factor due to the relative velocity of the counter-moving bunches being almost $2c$.)

The deflection, say horizontal, $\Delta\theta_x$, suffered by a weak beam particle of charge Q_w as it passes a strong beam bunch is given by

$$\Delta\theta_x = \frac{\Delta p_x}{p_0 c} = \frac{1}{p_0 c}\int dt\,\frac{dp_x}{dt} = \frac{1}{p_0 c}\int 2Q_w E_x(z)\,\frac{dt}{dz}\,dz. \tag{9.9}$$

The factor 2 in the numerator includes the magnetic force, and the factor $dt/dz = 1/(2c)$ accounts for the relative velocity of the bunches.

9.4. The Beam-Beam Tune Shift Parameters

To a lowest (linearized) approximation the force on a particle in the weak beam due to the strong beam is lens-like, but unlike a quadrupole in that the focus/defocus character is the same in both planes. Repeating "golden rule" formulas from Eqs. (3.47), the leading effects of such focusing are tune shifts given by

$$\xi_x = \frac{1}{4\pi}\beta_x \Delta q_x, \quad \xi_y = \frac{1}{4\pi}\beta_y \Delta q_y, \tag{9.10}$$

where Δq_x and Δq_y are inverse focal lengths depending on the bunch charge and profile of the strong beam. Unlike a normal quadrupole, Δq_x and Δq_y have the same sign. Especially in electron colliding beams Δq_x and Δq_y can be vastly different but, for maximum luminosity, ξ_x and ξ_y tend to be roughly equal.

The values of ξ due to a single interaction point rarely exceed 0.05 in electron colliders or 0.01 in hadron colliders. But colliding beam facilities usually have more than one interaction point. The tune shifts they cause are strictly additive, and can add up to a total tune shift as great as 0.2 or more, even though this might seem to have required crossing destructive resonances.

PROBLEM 9.5. *For the round beam distribution of Eq. (9.7), show that the horizontal, beam-beam tune shift parameter is given by*

$$\xi_x = -\frac{\beta_x}{4\pi}\,\frac{1}{p_0 c/Q_w}\,\frac{N_s Q_s}{4\pi\epsilon_0 s^2}. \tag{9.11}$$

PROBLEM 9.6. *By keeping the next term in the expansion of the round beam deflection, calculate the octupole-order horizontal deflection.*

9.5. Impracticality of Taylor Map Beam-Beam Representation

The "altitude chart", for $w(z)$ in Abramowitz and Stegun's[**23**] FIG. 7.3, shows regular behavior in the upper right quadrant, but there are poles in the lower right quadrant. From the figure and the relation

$$w(\overline{z}) = \overline{w(-z)}, \tag{9.12}$$

one concludes that expression Eq.(9.5) has the correct symmetry for left-right reflection through the vertical axis, but that it does not give the physically demanded up-down symmetry for reflection through the horizontal axis. It follows that Eq. (9.5) is valid in the upper half-plane and invalid in the lower half-plane. Any analytic approximation to this formula will have similar properties. For purely numerical tracking this is not a problem since lower half-plane deflections can be inferred by using the reflection symmetry. But application of symmetry for purpose of evaluation is inconsistent with TPS representation.

Part of experimental, storage ring, parasitic-bunch-crossing lore is that a particle in one bunch will be lost if its amplitude encroaches on the opposing bunch's "space", meaning the region of non-negligible charge density. This is not qualitatively inconsistent with the mathematical observation that non-vanishing charge density is incompatible with analytic two dimensional field (as follows from Ampére's law.) A formalism such as TPS, that relies on differentiability, is therefore, in principle, invalid for even remote collisions of Gaussian-shaped beam bunches. But Gaussian charge distributions fall off dramatically at large amplitudes and physically realistic bunches truly vanish outside, say, 5 or 6 sigma. This makes it valid to contemplate incorporating remote, or "parasitic" beam-beam collisions into a TPS simulation formalism. As mentioned previously, to distinguish such collisions from head on collisions (for which the element type name `headonbb` is employed) parasitic collisions are modeled by the `remotebb` type.

The map simulation of remotebb parasitic crossings will inevitably break down for crossings that are too close. If particles can actually be lost due to parasitic crossings one cannot expect the simulation to model the loss evolution accurately. (When contemplating train wrecks one concentrates on avoiding them, not on accurate description of the wreckage.) On the other hand, since failure of analyticity manifests itself by unphysical forces proportional to a possibly-high power of amplitude, one can be optimistic that numerical simulation can predict with quite good accuracy the amplitude beyond which particles are lost. The next section describes formulas to be used, within UAL, to model the remotebb type polymorphically.

For a headonbb type, one can contemplate using its TPS map to simulate its effect on weak beam particles of small amplitude, such as “one tenth sigma”. The so-called “beam-beam tune shift” parameterization of the collision amounts to just such a pure linearization of the deflecting force. Even some amplitude dependent effects can perhaps be estimated while restricting the amplitudes to sufficiently small values. But the code warns against concatenation of headonbb elements in any simulation that pretends to be a self-consistent model using a distribution of weak beam particle amplitudes matching the strong beam distribution. For faithful modeling in this case the code uses pure numerical evaluation of each headonbb interaction. This is not a serious limitation in practice, but it causes the application of some nonlinear tools, such as normal form analysis, to be invalid.

There are approximation algorithms available for calculating the fields of Gaussian charge distributions accurately for all possible bunch separations. But, to avoid complication, it is sensible to accept reduced accuracy in the interest of reducing the number of evaluation patches. The next section discusses evaluation for parasitic collisions and the section after that discusses headon evaluation. Since the code takes care of choosing a good algorithm in each case, for benchmarking and checking the code reliability, it should be adequate to spot check only a relatively small number of cases.

9.6. Padé Approximation For remotebb Type Collisions

The truncated Taylor series formalism is introduced into particle trajectory description to model nonlinear deflecting elements. The deflections caused by such elements can be expressed as power series in the transverse coordinates (x and y). Unfortunately the beam beam deflection cannot be described by a single, everywhere-convergent, power series. For example, the field at large distances falls off as $1/r$ and such a term diverges at the origin. This problem is somewhat ameliorated by the use of Padé approximation. As explained in reference[**21**], the needed function $w(z)$ can be approximated in the vicinity of a point z_0, in terms of (complex) deviation $z - z_0$, in a Padé form

$$w(z) \approx \frac{c_0 + c_1(z-z_0) + c_2(z-z_0)^2 + c_3(z-z_0)^3}{1 + d_1(z-z_0) + d_2(z-z_0)^2 + d_3(z-z_0)^3 + d_4(z-z_0)^4}. \tag{9.13}$$

This expression is relatively quick to calculate, and retains at least as many terms as are likely to be needed for any practical Taylor series to be generated subsequently. Because the denominator terminates in a power N higher by one than the numerator M, the behavior at large radius is appropriate. If needed, expressions with larger values of M and N are easily obtainable.

The coefficients c_i and d_i can be calculated by a program such as the following MAPLE program:

```
restart: with(numapprox): Digits:=30:M:=6: N:=M+1:
z0:=0.0+I*0.0:
open("PCoeffs.dat",WRITE):
fprintf("PCoeffs.dat", "%d\\n", M):
fprintf("PCoeffs.dat", "%15.8e\\n", Re(z0)):
fprintf("PCoeffs.dat", "%15.8e\\n", Im(z0)):
evalf(pade( exp(-z^2)*erfc(-I*z), z=z0, [M,N])):
w := subs(z-z0=d, %):
wnum:=numer(w): wden:=denom(w):
sd:=coeff(wden,d,0):
for j from 0 by 1 to M do
   fprintf("PCoeffs.dat", "%15.8e\\n", Re(coeff(wnum,d,j))/sd):
   fprintf("PCoeffs.dat", "%15.8e\\n", Im(coeff(wnum,d,j))/sd):
end do:
for j from 0 by 1 to N do
   fprintf("PCoeffs.dat", "%15.8e\\n", Re(coeff(wden,d,j))/sd):
   fprintf("PCoeffs.dat", "%15.8e\\n", Im(coeff(wden,d,j))/sd):
end do:
fclose("PCoeffs.dat");
```

This program outputs M, $z0$, and the coefficients to file "PCoeffs.dat" in a format (one number per line) convenient for reading into the C++ program that calculates electric fields. (For example "bmbm.cpp", listed at the end of this report.) For example, with $(M,N) = (6,7)$ and $z_0 = 0$,

```
c0 = 1.00000000e+00  0.00000000e-01
c1 = 0.00000000e-01 -1.25647718e+00
c2 =-8.25059157e-01  0.00000000e-01
c3 = 0.00000000e-01  3.19300157e-01
c4 = 7.63191604e-02  0.00000000e-01
c5 = 0.00000000e-01 -1.04697937e-02
c6 =-6.44878650e-04  0.00000000e-01

d0 = 1.00000000e+00  0.00000000e-01
d1 = 0.00000000e-01 -2.38485634e+00
d2 =-2.51608137e+00  0.00000000e-01
d3 = 0.00000000e-01  1.52579039e+00
d4 = 5.75922692e-01  0.00000000e-01
d5 = 0.00000000e-01 -1.35740709e-01
d6 =-1.85678083e-02  0.00000000e-01
d7 = 0.00000000e-01  1.14243694e-03
```

An example with $(M,N) = (3,4)$ and $z_0 = 3$ is,

```
c0 = 1.23409804e-04  2.01157317e-01
c1 = 2.33554192e-01  1.60868941e-01
c2 = 1.25324805e-01 -4.04528997e-02
c3 = 8.84183536e-03 -1.80649734e-02

d0 = 1.00000000e+00  0.00000000e-01
d1 = 1.19099484e+00 -1.16400275e+00
d2 = 8.87402357e-02 -1.07153615e+00
d3 =-1.68262727e-01 -2.69022561e-01
d4 =-3.19855404e-02 -1.57189728e-02
```

This expansion is appropriate for calculating $w(z)$ with argument near $z = 3$. The program is easily extended to calculate the Padé coefficients with center points located on a regular grid.

Even with Padé approximation, problems remain. An expansion like that of Eq. (9.13) is valid only in a restricted patch. An expansion centered on the origin gives a respectable approximation to w in the entire upper half-plane. CBN 80-13 gives algorithms appropriate for enough patches to cover the entire x, y space with high accuracy. But this multi-patch treatment can be incorporated into the TPS formalism only by applying the map, particle-by-particle, to obtain the numerical coordinate values of every particle at every `headonbb` location.

When used to calculate the electric field, because of its $x \to -x$ symmetry, the Padé expansion centered on the origin fits the electric field (which immediately provides the magnetic field as well) reasonably well in the upper half-plane. Preliminary investigations have shown that some remote crossing cases can be modeled adequately with just the Padé expansion centered on the strong beam. This is a sensible approach to take for preliminary investigations. As shown in CBN 80-13 the origin-centered Padé expansion with $M = 6, N = 7$ (called `PADE1` in CBN 80-13) gives a respectable approximation over a the entire upper half plane and a narrow band just below the real axis. This can be referred to as a "default" representation. It is implemented with hardwired coefficients in subroutine `wpade3` (listed below) and with externally supplied coefficients in `wpade4`.

For higher accuracy each parasitic collision could have its own Padé approximation. Since the two occurrences of w in Eq.(9.5) have different arguments, it would be necessary to use two Padé formulas for a single bunch crossing. The MAPLE program listed above gives Padé coefficients at such a point. Coefficients are needed for the two argument values corresponding to the separation of the bunch centers. The same procedure can handle the case of bunch separation more vertical than horizontal and the case of beams higher than they are wide ($s_x > s_y$.)

To avoid the need for programming the coefficient determination into the C++ code, UAL requires them to be calculated off-line, or at least in a separate module. The coefficients are then be passed to the simulation as parameters of the `remotebb` element. Other required parameters are (s_x, s_y, s_z), $(\Delta x_{01}, \Delta x_{02}$, and the strong beam strength (equivalent to λ).

Consider a parasitic bunch crossing in which the particle being tracked lies in a "weak beam" that is displaced approximately horizontally by (positive) distance Δx from the center of the "strong beam". This separation might also have non-zero vertical deviation. Also, since the horizontal deviation depends, weakly to be sure, on the parasitic interaction, it is, in principle, necessary to determine Δx iteratively.

Two error function determinations are required. Always taking the strong beam center as origin, one requires Padé expansions centered on the two points

$$x_{01} = \frac{\Delta x}{s}, \quad x_{02} = \frac{\Delta x}{s}\,\frac{s_y}{s_x}. \tag{9.14}$$

If the beams have vertical separation it is necessary to alter the Padé center points accordingly. The case $s_y > s_s$ also requires special treatment not exhibited here.

Re-expressing Eq.(9.5) in terms of coordinates x and y relative to the weak beam center yields

$$\begin{pmatrix} E_x \\ E_y \end{pmatrix} = \frac{\lambda}{2\epsilon_0\sqrt{\pi}}\,\frac{1}{s}\begin{pmatrix} \Im \\ \Re \end{pmatrix}\,(w(x_{01}+\frac{x}{s}+i\,\frac{y}{s}) - e^{-(\frac{(\Delta x+x)^2}{2s_x^2}+\frac{y^2}{2s_y^2})}\,w(x_{02}+\frac{x}{s}\,\frac{s_y}{s_x}+i\,\frac{y}{s}\,\frac{s_x}{s_y})). \tag{9.15}$$

For typical, many sigma, bunch separations at parasitic crossing, and for weak beam betatron amplitudes out to all but unphysically large amplitudes, the two w terms will be close enough to the Padé expansion points that the Padé approximations give accurate deflections. This should not be surprising since the strong beam is simply a multipole-rich electromagnet from the point of view of the weak beam.

9.7. Padé Approximation For headonbb Collisions

As explained earlier, the procedure of the previous section can be applied only for small particle amplitudes. In this limit both expansions in Eq. (9.15) are centered on the origin. As mentioned above PADE1 (which has been ported to wpade3 and wpade4 below) gives a decent approximation in the entire upper half region. But for negative vertical displacements y more negative, say, than $-s_y$ (which is a very probable amplitude) the calculated electric field rapidly becomes incorrect. Even for amplitudes of smaller amplitude than this, any apparent amplitude-dependent tune shifts would be suspect and would require careful verification.

For trajectory following in the presence of headonbb elements it is still possible to use maps, but every trajectory has to be converted to numbers at every crossing. The most compact map representation possible would concatenate all elements, including parasitic crossings, in every arc into entire-arc maps. At every head-on crossing the next-arc map would be applied to each particle's (numerical) post-interaction coordinates to obtain its (numerical) pre-interaction coordinates at the next crossing point.

Bibliography

[1] QT website, `http://www.trolltech.com`

[2] R. Brun et al., *The ROOT Framework,* AIHEPNH-96, Lausanne, Switzerland, August 1996

[3] V. Fine, *Cross-Platform Qt-Based Implementation of the Low Level GUI Layer of ROOT,* ACAT'2002 Workshop, NIM **502**, Issues 2-3, April 2003

[4] N. Malitsky and R. Talman, *Unified Accelerator Libraries,* Proceedings of Conference on Accelerator Software, Williamsburg, VA, 1996.

[5] N. Malitsky and R. Talman, *UAL User Guide,* BNL-71010-2003, available at `http://www.ual.bnl.gov`, 2003

[6] L. Schachinger and R. Talman, *Teapot: A Thin-Element Accelerator Program for Optics and Tracking,* Part. Accel. **22**, 35, 1987

[7] C. Wang, V. Sajaev, and C. Yao, *Phase advance and β function measurements using model-independent analysis,* PRST-AB **6**, 104001 (2003)

[8] J. Irwin, C. Wang et al., *Model-Independent Beam Dynamics Analysis,* Phys. Rev. Letters, **82**, 1684 (1999)

[9] I. Jolliffe, *Principle Component Analysis*, Second Edition, Springer, 2002

[10] F. Press et al., *Numerical Recipes*, ISBN 0-521-43108-5, Cambridge University Press, 1992

[11] Koutchouk, J-P, *Trajectory and Closed Orbit Correction*, in Frontiers of Particle Beams; Observation, Diagnosis and Correction, Joint US-CERN School on Particle Accelerators, M. Month and S. Turner, (Eds.), Anacapri, Italy, 1988

[12] A. Verdier and T. Risselada, CERN/ISR-BOM-OP/82-19(1982)

[13] G. Bourianoff, S. Hunt, D. Mathieson, F. Pilat, R. Talman, G. Morpurgo, *Determination of Coupled-Lattice Properties Using Turn-By-Turn Data*, SSCL-Preprint-181, 1992

[14] F.R. Gantmacher, *The Theory of Matrices*, Chelsea, New York, 1977, Section II§5

[15] W. Press, B. Flannery, S. Teukolsky, and W. Vetterling, *Numerical Recipes in C,* Cambridge University Press, Cambridge, 1988, p156

[16] Jie Wie, *Longitudinal Dynamics of the Non-Adiabatic Regime on Alternating Gradient Synchrotrons,* SUNY Stony Brook PhD thesis, 1990, revised 1994

[17] C. Montag and J. Kewisch, *Commissioning of a first-order matched transition jump at the Brookhaven Relativistic Heavy Ion Collider*, PRST-AB, **7**, 011001 (2004)

[18] R. Talman, *Theory of Head-Tail Chromaticity Sharing,* Cornell Report, CBN 97-16 (1997)

[19] L. Mandel and E. Wolf, *Optical Coherence and Quantum Optics*, Cambridge, p. 128 (1995)

[20] R. Meller, A. Chao, J. Peterson, S. Peggs, and M. Furman, *Decoherence of Kicked Beams*, SSC-N-360 (1987)

[21] Y. Okamoto and R. Talman, *Rational Approximation of the Complex Error Function and the Electric field of a Two-Dimensional Gaussian Charge Distribution,* Cornell Report, CBN 80-13 (1980)

[22] M. Bassetti and G. Erskine, *Closed expression for the electrical field of a two-dimensional Gaussian charge,* CERN-ISR-TH/80-06

[23] M. Abramowitz and I.A. Stegun, *Handbook of Mathematical Functions,* National Bureau of Standards, 1966

[24] A. Dragt, *Lectures on Nonlinear Orbit Dynamics,* American Institute of Physics, New York, 1981

[25] E. Forest, *Beam Dynamics*

[26] G. Hori, *Theory of General Perturbations with Unspecified Canonical Variables,* Publ. Astro. Soc. Japan, Vol. 18, No. 4, 1966

[27] A. Dragt, in *Proceedings of the 1984 Study on the Design and Utilization of the SSC*, ed. by R. Donaldson and J. Morphin, Am. Phys. Soc., Snowmass, 1985

[28] M. Berz, *Differential Algebraic Description of Beam Dynamics to Very High Orders,* Particle Accelerators, **24**, 109, 1989