



OPA User Guide version 4.047

Andreas Streun, PSI, June 22, 2022

1 Introduction

OPA's main purpose is to support the development of electron (positron) storage rings. Emphasis is on visualization and interactivity.

OPA is in particular useful for designing high brightness light source lattices, but may be used for transfer lines and other types of lattices as well.

OPA uses simplified models and approximations in order to support straightforward design work. Results should be checked with other codes using more complete beam dynamics models.

For the beam dynamics implemented in OPA see the accompanying document “inside OPA”. To get started try “OPA tutorial”. In this document we try to give a complete overview on data and function of OPA. While reading you may have the two other documents within reach and OPA running.

1.1 History and acknowledgements

OPA is based on the code OPTIK from Klaus Wille, who started in the 80's already to work on a design tool for electron rings. In 1993 he kindly passed it on to the author, who developed it further and used (and uses) it for the design of the Swiss Light Source, SLS (and the upgrade, SLS-2). Algorithms for sextupole optimization and signal processing were kindly contributed by Johan Bengtsson. Simon Leemann did a lot of tests and suggested many extensions and changes during the design of MAX-IV. Michael Borland and Chun-xi Wang tested the module for non-linear optimization for consistency with the ELEGANT code and helped to find several bugs. Volker Ziemann helped with implementation of coupling and calculation of radiation integrals.

1.2 Capabilities and limitations

OPA includes the following features:

- Human readable lattice input/output and robust lattice reader.
- Text editor and modular editor for variables, elements and segments.
- Interactive graphical design of linear optics with many convenient functions as knobs, zoom, matching, tune diagram etc.

- Implementation of the Edward-Teng formalism for coupled betatron motion (not all modules).
- Interactive optimization of sextupole and octupole Hamiltonian in 1st and 2nd order.
- Tracking with FFT: Poincaré plots, resonance guesses and amplitude dependent tunes.
- Tracking of dynamic apertures.
- Tracking of Touschek lifetime.
- Correlated misalignments
- Display of lattice geometry.
- Orbit distortion and correction.
- Simulation of the injection process.
- Calculation of magnet currents and export of EPICS snap files.
- Export of TRACY-2, TRACY-3, MAD-X, ELEGANT and BMAD lattice files.

OPA has the following limitations:

- Only correct for rings with large circumference and high energy:
 - High relativistic approximation: $\gamma \gg 1$
 - Paraxial approximation: $x' = dx/ds \ll 1$
 - Large curvature approximation: $x \ll \rho$ (bending magnet radius)
- Exactly correct only on energy (i.e. $\Delta p/p = 0$), since it uses internally x' instead of the canonical coordinate p_x .
- Nonlinear elements are treated by a 2nd order symplectic integrator only.
- Only planer insertion devices allowed. Simple model as dipole array.
- Only 4D tracking (no synchrotron oscillations).

2 Installation

OPA started in Borland Turbo Pascal, was converted to Borland Delphi, and then to Lazarus Free Pascal, which is like an open source version of Delphi. Lazarus is platform independent. Presently OPA runs on Windows-10 and on xUbuntu with GTK2. Both versions are built from the same source files.

In W10 Lazarus creates a single executable **opa.exe**, which can be downloaded and should run on any W10-system. In Linux an executable **opa** is created, but it only runs on the system where it was created. A debian file to install the program and set the dependencies has not yet been established. Instead one has to use Lazarus to build the executable from the source.

\implies *to do: create a debian file to install on Linux.*

W10-executable, source files and documentation is available at the OPA page ados.web.psi.ch/opa.

For using OPA it is recommended to create a folder for lattice data, probably with subfolders for different projects. The executable should be in a separate program folder. If the executable is built from Lazarus, it is usually a subfolder named `/lib/x86_64-linux` or `-win64` under the folder where the source files are located. Note, that OPA needs write permission for the folder where the executable is located in order to write two settings files, named `opa4_glob.ini` and `opa4_path.ini`

⇒ *This should be improved to make a clear separation between the program files and the user files changed at run-time.*

When starting OPA, it will write a settings files `opa4_set.ini` in each data folder selected in order to remember settings for the next session. These settings can be different for each data folder. The file `opa4_path.ini`, which exists only once in the executable's folder, remembers the path names of the the last used files.

OPA lattice files are named `*.opa`. Unlike many other lattice codes, these files are not read-only but can be composed in OPA and saved. OPA can start from scratch without reading a lattice files. Many more files, to be described below, are exported by different modules on request during the sessions.

3 Lattice Data

Lattice **elements** are of different type, like quadrupole, drift space, dipole etc. A lattice is a line-up of elements, called a **segment**. Segments may contain a mixture of other segments and elements, thus the lattice is made from nested segments to be unpacked recursively in order to get an array of elements to work with.

An element has a **name**, a **type** and a set of **parameters** depending on the type. Parameters may be numbers (of different type like integer, real, boolean etc.) or strings representing an algebraic expression. Expressions may contain **variables**, which are either numbers or expressions themselves. – Consider this simple example:

<code>V1 = 0.5;</code>	A variable given as number
<code>V2 = -2*V1;</code>	A variable given as expression
<code>D: Drift, L=1.0;</code>	A drift space of length 1m
<code>Q1: Quadrupole, L=0.2, K=V1;</code>	Quadrupole length 0.2 m, strength 0.5 m^{-2}
<code>Q2: Quadrupole, L=0.2, K=V2+0.1;</code>	Quadrupole, strength -0.9 m^{-2}
<code>B: Bending, L=0.5, T=10, T1=5, T2=0;</code>	Dipole of angle 10° , entry/exit edge angles $5^\circ/0^\circ$
<code>S1: B, D, Q1;</code>	a segment connecting dipole drift, quadrupole
<code>S2: D, -S1, 2*D, Q2;</code>	segment with inverted segment and repeated elements.
⇒ <code>D, -Q1, -D, -B, D, D, Q2;</code>	this would be the expanded lattice.

Note that both quadrupole strengths depend on the variable `V1` which will be a “knob” in OPA and thus they are varied in a correlated way. Elements and segments can be repeated

or inverted. For most elements, inversion does not change their properties, but in case of a bending magnet, edge angles are interchanged.

⇒ Expressions are not allowed for all element parameters, presently only for length, quad strength, bending and edge angles) however this would be straightforward to extend to all.

Lattices are read from and written to *.opa lattice files. The syntax, as indicated above, is rather simple and similar to lattice files for other codes like TRACY or MAD (check out an example file or the tutorial).

First come a few global parameters, →3.1, then comes the list of variables, →3.2, then the list of elements, →??, and finally the list of segments, →??, which will be unpacked recursively to generate the lattice.

All inputs are optional, since OPA may start without lattice as well. For parameters of an element not given in the input file, OPA assumes reasonable defaults. If a parameter is mistyped, OPA will ignore it and set it to the default value.

Text in curly brackets { ... } is treated as comment, i.e. ignored by OPA, except text bracketed by {com ... com}, which is the "official" comment text like lattice title and some notes. Further comments visible in the *.opa-file are generated by OPA for the user's convenience. Comments written manually into the *.opa file will be lost.

All input lines (defining an element, a segment), which may extend over several physical lines in the file, have to be terminated by semicolon (;).

3.1 Global Parameters

The lattice header includes the official comment {com ... com}, the beam **Energy** in GeV (default 1 GeV) and a flag **Rotinv=0/1**; telling if rotations are inverted too if an element is inverted (depending on the application the one or the other option is preferable.)

Further the initial optical parameters (**BetaX=...** etc.) at the beginning of the last used segment are written to the file header. *⇒ This has historical reasons and should be removed and replaced by an OpticsMarker (see below).*

To use the option of exporting magnet currents and channel names (→4.14), a link to the corresponding allocation and calibration files is required and should be given at the beginning of the *.opa file:

allocation = (allocation-file).dat; calibration = (calibration-file).dat; See **ados.web.psi.ch** for SLS allocation (lookup) and calibration files.

3.2 Variables

Variables can be defined to be referenced by each other or by some of the elements. Variables can be numbers or expressions using basic operators (+-*/) and arbitrary number of bracket pairs. Variable name has to start with a character, then numbers and characters may follow. Syntax check is performed. Example: `A = 0.5; XB3S = (1.0-A)/2;`

3.3 Elements

Tables 2 and 1 list the parameters of the elements: column “in file” tells how to address them in the lattice file (case insensitive), column “in record” refers to OPA’s internal `ElementType` record (in unit `OPAGlobal.pas`). An asterisk indicates that an algebraic expression is allowed too. In this case the record contains a string of same name followed by `_exp`. “unit” is the physical unit for the number in the lattice file. “note” refers to a comment below. The element parameters can be accessed interactively from the OPAEditor menu as well.

Comments to the tables:

- a) a rectangular aperture $(\pm a_x) \times (\pm a_y)$ is assumed for aperture checks in tracking, however an elliptic aperture (half axes a_x, a_y) is assumed for calculation of linear acceptance and for Coulomb lifetime.
- b) positive rotation is counter-clockwise around the beam direction, i.e. 90° rotation of a dipole will deflect the beam upwards.
- c) if keyword is present, the drift is shown as grey solid, not invisible; useful to indicate protected space, e.g. coils of a magnet.
- d) Positive deflection is towards the ring center, i.e. in $-x$ -direction and thus will create positive dispersion.
- e) K_1 is the fringe field parameter defined by Brown (see “inside OPA”/Elements/on-axis). If only `K1` is given in the lattice, the same parameters is applied to both edges.
- f) “Bending” and “Combined” are almost identical, except that “Combined” allows a sextupole component. This has historical reasons. \implies *to be merged in one element*.
- g) Multipole definition includes the factorial, i.e. $b_n = \frac{p}{e} \frac{1}{(n-1)!} \frac{dB_y^{(n-1)}}{dx^{n-1}}$
- h) Internally the thick, nonlinear magnet is approximated by N slices of length L/N with a thin multipole of integrated strength $b_n L/N$ at the center.
- i) If sextupole length is given as zero, the strength is integrated strength, unit m^{-2} .
- j) Multipole is always considered thin. If a length `LMAG` (not `L`!) is given, it just reserves spaces of `LMAG/2` on both sides of the single, thin kick.
- k) Helical undulators are not (yet) implemented, only planar undulators are supported.

Table 1: Element description, part I

Property	in file	in record	unit	note
All elements				
Name		nam		
Length	L	l*	m	
Apertures	Ax, Ay	ax, ay	mm	a)
Rotation	Rot	rot*	°	b)
Drift	Drift	cdrif		
(protect)	Block	block	-/-	c)
Quadrupole	Quadrupole	cquad		
Strength	K	kq*	m ⁻²	
Bending magnet	Bending	cbend		
Deflection	T	phi*	°	d)
Edge angles	T1, T2	tin*, tex*	°	
Focusing strength	K	kb*	m ⁻²	
Full gap	Gap	gap	mm	
Edge K1	K1, K1in, K1ex	k1in, k1ex	1	e)
Combined function	Combined	ccomb		f)
Deflection	T	cphi*	°	d)
Edge angles	T1, T2	cerin*, cerex*	°	
Focusing strength	K	ckq*	m ⁻²	
Full gap	Gap	cgap	mm	
Edge K1	K1, K1in, K1ex	ck1in, ck1ex	1	e)
Sextupole strength	M	cms	m ⁻³	g)
(slices)	N	cslice	1	h)
Sextupole	Sextupole	csext		
Sextupole strength	K	ms	m ⁻³	g)i)
(slices)	N	sslice	1	h)
Thin multipole	Multipole	cmpol		
Multipole order	N	nord	1	
Integrated strength	K	bnl	m ¹⁻ⁿ	g)
length	Lmag	lmpol	m	j)
Undulator	Undulator	cundu		k)
Period length	Lamb	lam	mm	
Peak field	Bmax	bmax	T	
Full gap	Gap	ugap	mm	
Fill factors	F1,2,3	fill1,2,3	1	l)
(half)	Half	halfu	-/-	m)

Table 2: Element description, part II

Property	in file	in record	unit	note
Kicker		ckick		
Multipole order	N	mpol	1	
Peak field offset	X	xoff	mm	n)
Int. strength \leftrightarrow peak kick	Amp	$m^{1-n} \leftrightarrow$ mrad		
Period	T	tau	ns	o)
Delay	Delay	delay	ns	o)
(slices)	NK	kslice	1	
Septum		csept		
(not yet implemented)				
Solenoid		csole		
Strength	K	ks	m^{-1}	
Hor. corrector	H-corrector	ccorh		p)
Kick	Dxp	dxp	mrad	
Vert. corrector	V-corrector	ccorv		p)
Kick	Dyp	dyp	mrad	
BPM	Monitor	cmoni		p)
– no parameters to set –				
Girder		cgird		
Type	typ	gtyp	1	q)
Shift	(not implemented)			
Optics marker		comrk		r)
Orbit	OrbitX/Y, OrbitXp/Yp	om[^].orb[4]	mm, mrad	
Normal mode betas	BetaX/Y, AlphaX/Y	om[^].bet[4]	m, 1	
Dispersion	EtaX/Y, EtaXp/Yp	om[^].eta[4]	m, 1	
Coupling matrix	C11, C12, C21, C22	om[^].cma[4]	m, 1, m^{-1}	
(screen)	Screen	screen	-/-	
Photon beam	PhotonBeam	cxmrk		s)
Length	XL	xl	m	
Style	Style	nstyle	1	

- l) Undulator fill factors used for fast calculation of radiation integral (see “inside OPA“ sec.3), depending on the field variation over a period. Default is sinusoidal field.
- m) Internally, an undulator is modelled as a series of small dipoles, where the dipoles at entry and exit are attenuated to 1/4 and $-3/4$ field of the inner array dipoles. The keyword **half** suppresses the attenuation at the exit, because sometimes one likes to split the undulator (U) to insert a marker (M). In this case the full undulator is modeled from two halves as (U,M,-U).
- n) For $X=0$ the kicker is considered a pure multipole kicker and K gives its integrated strength. Non-zero X identifies a non-linear kicker with $\sin^{n-1}(2\pi x/X)$ shaped field, and K is the maximum kick in mrad at X. For a dipole kicker (N=1) the field shape is the same for both cases, and K is the kick in mrad.
- o) The pulse is assumed to be of half-sine shape with length (= half period) T, starting/ending at $\text{Delay} \pm T/2$. Beam time of flight is counted from the start of the lattice, and the delay has to be adjusted appropriately to kick the beam (the injection module provides a synchronization button).
- p) Correctors with reserved names CH, CV, and monitors with name MON are considered components of the orbit correction system, will be expanded into a series of individual elements and set by the orbit correction procedure. In the lattice a series of elements named CH001, CH002 ... will appear, but these are not saved to the file, since they do have design values but will be set by orbit correction. The same is true for skew quadrupoles with reserved name CS.
 \Rightarrow However, OPA does not yet contain a coupling correction scheme but values are temporarily imported from a TRACY run (Extra \rightarrow test).
- q) The girder element defines the begin or end of a girder, thus they should appear always in pairs. Type 0 is a loose end, type 1 is a “train link”, i.e. a [virtual] joint to the next girder, type 2,3 are secondary girders sitting on top of primary girders, where the roll of the primary girder is transmitted for type 3, but not for type2 For example the SLS dipoles bridge the girders, with 2 contact points (type 3) on one girder, and 1 contact point (type 2) on the other girder.
- r) OpticsMarkers are useful for building a lattice and matching. They are markers in the lattice which on request save all 16 transverse optical parameters defining the beam (orbit, beta functions, dispersions and the coupling matrix), and which can be used as starting point for optics calculations or target points for matching. The keyword **screen** indicates an optical screen to be indicated in the geometry layout.
- s) Presently photon beam is only used for plotting x-ray beams in the geometry layout. There are 3 different styles (0,1,2) which can be used for different source types.

3.4 Segments

A segment is a line-up of elements and segments. Many segments may be defined, the segment to be used to built the lattice is selected interactively later. A minus sign reverses the segment. A factor repeats it.

The order of segments does not matter, so a segment used by another segment may appear later in the file. A test is performed for circular references, which are not allowed of course.

The string `nper=n` appearing somewhere in a segment line-up defines the periodicity, i.e. the factor `n` is applied for convenience to tunes, sextupole terms etc of the lattice built from this segment, e.g. `ARC : HALF, XXX, -HALF, nper=6;`

Note: if one likes to split a bend, for example to put a marker in the centre, and a non-zero gap is given, then one has to set `K1EX=0.0` otherwise the internal edge where the dipole is split, will have an effect on the vertical focusing!

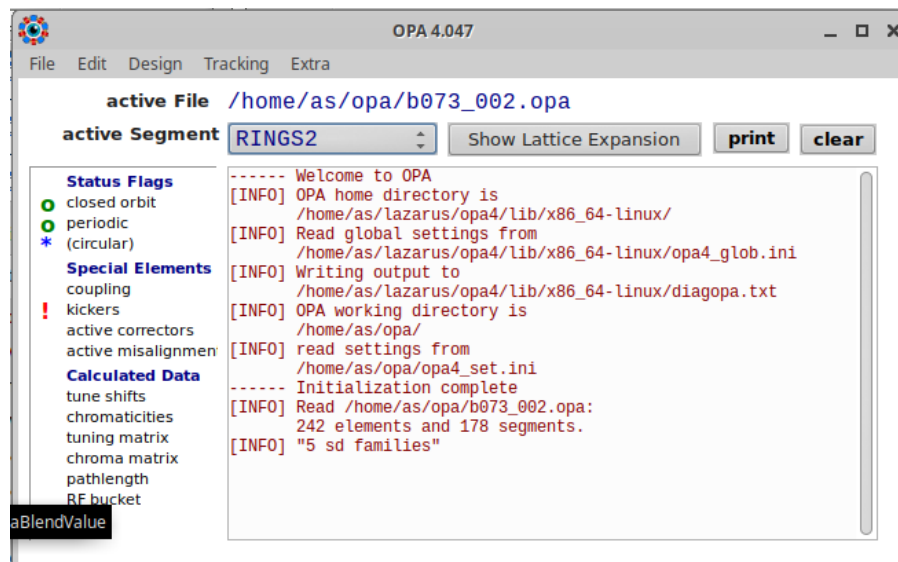


Figure 1: OPA main menu on xubuntu

4 OPA functions

For getting started, one may try the tutorial.

The OPA menubar (see Fig.3.4) gives access to all functions. If functions are not available, because a prerequisite is missing (e.g. periodic solution for tracking) the items are disabled and shown in gray. The table gives an overview of the functions behind the menubar (arrow → leads to a sub-menu, items in (brackets) are only temporary or tests):

File	Editor	Design	Tracking	Extra
New	Text Editor	Linear Optics	Phase Space	Magnet parameters
Open	OPA Editor	Off Momentum	Dynamic Aperture	Magnet currents
Last Used →		Non-linear	Touschek Lifetime	(test)
Save		Orbit Correction		(BD pattern)
Save as		Injection Bumps		(spectra)
Export to →		RF Bucket View		Output →
Exit		LGB optimizer		
		Geometry Layout		

“Active file” shows the selected lattice file (*.opa). The “active segment” is selected from the segment list on the GUI. From this segment, the **lattice** is built by recursive expansion of the segments to get the elements line-up, which is shown in the message window when pressing “show lattice expansion”.

The small window (left) indicates lattice status, if special elements are present and which data have been calculated for internal transfer between modules. The large logging window is for general status information and error messages. It may be cleared or saved to a file named `LogPrint.txt`.

4.1 File

Here are the usual file operations.

New erases all saved lattice information to start composing a new lattice from scratch.

Open reads a lattice file. Usually this is an *.opa file. Also lattice files for other codes can be read, including files for TRACY (*.lat), MAD (*.mad, *.seq), ELEGANT (*.lte). However this works partially only but nevertheless may be useful to reduce manual conversion work.

Last Used opens a list of up to 10 files used recently. Selecting one will switch to the corresponding folder and red the local `opa4_set.ini` file to restore the settings for this folder.

Save, **Save as** write a *.opa lattice file.

Export to writes lattice files for TRACY-2, TRACY-3, MAD-X, ELEGANT, BMAD. Only a basic file is exported and some inspection/editing may be required before running the other code. If OPA Elements or element properties cannot be exported, a corresponding comment is inserted in the exported file. Note that TRACY-2, MAD-X and BMAD do not correctly invert segments containing bending magnets with unequal edge parameters: if an inverted segments contains such a bending magnet, the edge properties have to be interchanged at each (of perhaps many nested) segment inversions. Therefore, for these codes, an inverted segment is generated together with the original one to avoid this problem. Other peculiarities include using rad or degrees for angles, defining multipole strengths without or with the factorial etc.

The sixth option in the submenu **OPA (novar)** writes an opa-file replacing all variables by values. This is sometimes useful to make properties directly accessible, which are connected by variables.

Exit closes OPA after writing the local settings to the local `opa4_set.ini` file and some global settings and the list of last used files to `opa4_glob.ini` and `opa4_path.ini` in the folder of the executable.

4.2 Edit

There is the OPA editor and the text editor. New users may prefer the OPA editor, advanced users the text editor.

TextEditor shows the file exactly as it will be saved to disk, The “test” button performs a syntax check.

OPAEditor shows two lists, one for variables and elements, and one for segments. Further, there are some fields and buttons to

- enter the beam energy in GeV,
- enter the global comment to be saved with the lattice file
- set the element apertures. If the box “set all” is checked, all elements will be set, otherwise only elements where the apertures are larger than the new values in the two input fields,
- enter a number for the magnet pole radius, which is used to show pole profiles and calculate fields and gradients,
- invert all dipole polarities, if needed, and
- expand undulator elements in explicit series of drifts and dipoles.

Clicking into the left table “Elements and Variables” opens a menu for the selected element or variable. If the elements is a linear magnet (quad, bending, combined), the field profile is calculated and plotted using the magnet pole radius number mentioned above. Variables and elements can be edited giving number values or algebraic expressions.

Clicking into the right table “Segments” opens a grid to edit the line-up of elements and segments in the selected segment. Insert and delete keys may be used. Undefined elements are shown in red, before saving the segment they need either to be deleted or to be defined using the element editor. The periodicity field stores a number of repetitions of the segment, for example used to calculate tunes etc.

If the name of an element is changed, its entry in all segments will be changed too.

4.3 Design → Linear Optics

The lattice structure appears on the optics panel main window, and a start menu pops up:

“periodic” will try to find a solution with same parameters at both ends. If the box “coupling” is checked, calculations are performed for normal mode beta functions using the 4D Edward-Teng formalism, otherwise the simple 2D formalism is used, which, of course is wrong if there is a coupling element. Often, for nonzero coupling, there are two solutions for the periodic normal mode betas with interchanged tunes. Clicking the “flip” box shows the other solution.

“symmetric” will try to find solution with $\alpha_x = \alpha_y = \eta' = 0$ at both ends, so appending the mirror image makes it periodic. \Rightarrow *this option is not fully implemented and should be avoided.*

“single pass” calculates the optics starting at its left (“initial”) or right (“final”) end or from one of the “optics markers” if there are any in the lattice. The start value for the optics calculations may be edited with the tabs at right to manually enter orbit, [normal mode] betas, dispersions [and the coupling matrix].

If periodic or symmetric solution fails, the single pass solution from initial (left) values is calculated. If a periodic solution was found, the tune diagram pops up, see 4.3.1.

The main window will show the optical functions: By default, [normal mode] betas and the dispersion will be shown (β_x in blue, β_y in red, η_x in green and η_y in fuchsia). The table at right gives in its upper part total values of the lattice (some of them only valid, of course, if the lattice would be repeated periodically). In the lower part, local optics values are displayed. The location can be selected by right-clicking on the plot.

The element line-up is shown at the bottom of the figure. If variables have been defined, they will appear as yellow boxes under the plot. If a variable depends on other variables, it is shown in pale yellow.

Moving the mouse over one of the elements displayed at the bottom of the figure shows its name. Left-click on an element or on a variable and releasing the mouse button after moving to one of the “knob” fields, connects the variable or the most important parameter of the element (e.g. length of drift, strength of quadrupole etc.) to the knob. If the variable or the element parameter value is no number but an expression linking it to one or more variables, the knob only displays the value but cannot change it. So only variables and element parameters, which are pure numbers, can be varied using then knobs.

Variation of values by moving the knob slider or entering numbers changes the optics. The range of variation is given by the fields at the bottom of the knob, which are adjusted automatically, but can be changed using the “<>” and “><” buttons.

Left double-click on a variable or element opens the same panel as **OPAEditor**, which allows all parameters to be changed. However, element parameter expressions cannot be changed here but only in the editors.

Zoom and shift buttons at the top of the plot window magnify a region of interest: calculation through elements proceeds in slices of maximum length corresponding to 1 pixel on screen. The plot range is adjusted automatically or by the Betafunction/Dispersion scaling buttons, depending how the “auto/fixed” is set. The “save” button saves curves as references and dis-

plays them in darker color. The “clear” buttons clears the reference.

The “PlotMode” button opens a panel to select different plots:

“Beta functions and dispersion” is the default. The panel can be used to fix the plot range and to select what to show in the lower and upper part of the window: by default the normal mode betas are shown. Alternatively or in addition, the projections of the normal modes to the horizontal and vertical (dashed line to x, y and dotted line to y, x) can be shown. (Without coupling, the normal mode is identical to x, y and its y, x projections are zero.) A third option shows the normal mode betas after each element calculated from the periodic solution of the one-turn matrix taken at this location. However, this requires to first press the button “pp” in the start menu – in a large lattice, this calculation may take quite long! The results are shown in darker color and with not interpolation. Further one can select to show the dispersion (default), or the determinant of the coupling matrix or the orbit.

“Envelopes” shows orbit and 1-sigma beam sizes along the lattice. It may be calculated using the equilibrium emittance from the radiation integrals or manually input values. In a coupled lattice, usually both emittances are non-zero, in the ideal case without coupling, a value may be given for emittance coupling to get a vertical envelope too. The element apertures are also shown to see where the beam may scratch. Dispersion and coupling are included in the total beam sizes, the dispersive contributions are shown in green and fuchsia, the coupling contributions are shown as dashed and dotted lines like for the beta functions.

“Magnetic field” shows the ole tip field of the magnets using again the reference radius (which was already entered in **OPAEditor**). Dipole, quadrupole and sextupole components are shown in blue, red, green.

Now back to the start menu: usually the calculation is on-momentum as indicated by the label/button “dp/p[%]=0”. Pressing the button enables a momentum offset to be entered. Further checking the [-0+] box will perform three calculations for $0, \pm \Delta p/p$ and show them superimposed. Use the scrollbar at the bottom of the table at right to see the results for the three calculations. However, envelope mode shows only the on-momentum calculation.

In strongly coupled lattice an effect called “mode-flip” may occur: locally one of the two solutions may not exist (one has to exist always, otherwise we would not have a periodic solution at all), so the optics has to “jump” to the other solution. The lattice needs an even number of flips to get correct frational tunes and radiation integrals at the end, therefore, one may add a zero rotation at the end of the lattice to force a back-flip (flips occur only at rotations). Mode flips are indicated by pink vertical lines in the beta function plot.

⇒ The integer part of the tunes becomes wrong in a lattice with flips, not yet understood why and how to fix it.

The buttons at the lower right of the main window have following functions:

- “Start” opens again the start menu (in case it was closed)
- “PlotMode” opens the plot menu
- “Tune Matrix” gathers quadrupoles in two groups depending on the sign of the k -value and establishes a sensitivity matrix for a relative change of strength. This allows the global tune of the lattice to be varied rather smoothly within a small range.
- “Matching” opens the panel for iteratively adjusting beam parameters, see sec.4.3.2.
- “Write OMK” writes the current optics into the optics marker elements, which can be selected on a panel that will pop up.
- “linear/nonlinear” enables/disables non-linear elements, which affect the off-axis optics.
- “Kicker OFF/ON” enables the kickers which affect the orbit and with it the optics, if linear elements are enabled too.
- “→txt” writes several text files, beta functions to `(nam)_beta.txt`, lattice parameter to `(nam)_data.txt/.html` in plain text and html, magnet parameters to `(nam)_mag.txt` and radiation data to `(nam)_rad.txt`.
- “→GP” writes text files `(nam)_beta.out` with the beta functions and `(nam)_mcod.out` with element data to be read by a GNU-plot command file `(nam)_beta.gp`, which is also created. Running GNU-plot on this file (outside OPA) then will create an EPS file `(nam)_beta.eps`. \implies *Incomplete, to be continued and extended*.
- “Exit” terminates the optics design. Note that the question to save the data refers only to an internal save to further proceed inside OPA, but does not save the data to the file!

If optics design ends with a periodic solution, the **Design** option **Non-Linear** and the **Tracking** group are enabled.

4.3.1 Tune diagram

The tune diagram pops up when a periodic solution is found, it shows the working point at the centre and the resonance lines in the neighbourhood.

The lines $aQ_x + bQ_y = c$ to be shown are selected by the order buttons and the checkboxes:

- Order buttons select $|a| + |b| \leq \text{order}$.
- “nsys” not checked selects only systematic lines, where $c \bmod P = 0$, with P the periodicity of the lattice.
- “skew” not checked selects only regular lines, where $b \bmod 2 = 0$.

Buttons at bottom modify the plot range.

⇒ *The “Export” button has no function.*

Double-clicking the image saves the bitmap to the clipboard – this works for all plots in OPA!

4.3.2 Matching

⇒ *The matching module is one of the eldest OPA parts and rather outdated, therefore it is described only in brief. It works only for uncoupled lattice, but extension to coupling would be rather straightforward, however would require a reorganization of the GUI. More powerful and robust algorithms should be implemented which also take into account limits for the knobs.*

Matching can be done from any location to another in the lattice, i.e. begin, end or optics markers. Constraints at an intermediate marker may be added.

On the left panel, select parameters to adjust and enter target values. On the right panel, select variables or elements to be used for matching. Transverse gradients of bends are excluded by default but can be included (this is because otherwise the slices of a longitudinal gradient bend fill up the panel). If the number of variables/elements is equal or larger than the number of constraints, the “go” button is enabled and matching may start.

The algorithm is a Newton search using the inverse square sensitivity matrix of the most effective knobs as tangent for extrapolation. This method converges quadratically but is rather fragile. Reducing the “fraction” of iteration to be applied stabilizes. Other parameters are number of iterations and required precision for termination.

If the iteration was successful, one may watch the solution (“show”), “accept” or “reset” it to the initial. “retry” returns to the first screen to change the settings and try again.

The “scan” button allows one parameter to be selected and repeat the matching over some range of values, afterwards results for other parameters and the knob values may be plotted as a function of the parameter varied, and data may be exported.

“Cancel” buttons close the matching panel.

4.4 Design → Off-momentum

This module calculates the linear optics as a function of momentum offset. It works for periodic and also for single pass systems: the checkbox at top right makes the selection and is checked at start if a periodic solution was previously calculated.

The range of momentum variation and the number of steps may be set, and “Go”. The tune diagram will pop up in case of periodic solution and show the chromatic tune footprint.

Afterwards, several plots may be selected by the radio buttons at left. Each curve is fitted by a polynomial, the order to be selected by the field at right where also the coefficients will be

shown in a table. The “Units” button switches between plot units and SI units for displaying the coefficients.

The button. “→TXT” writes a text file of the data named (name) `momentum.N.txt`, where N is the number of the plot (radio button). Double-clicking the image copies it to the clipboard.

If pathlength “Dpath” was selected, the fitted data will be saved internally, and in the main menu (left panel), the flag “pathlength” will be marked. Then the data are available for calculating the RF-bucket, see 4.9.

In the same way, a previous run of the non-linear module (→4.5) may have saved the theoretical chromaticity, then the corresponding flag “chromaticity” will be marked in the main menu and when selecting tune polts (the first four radio buttons), also these theoretical values are shown by solid lines.

(\Rightarrow bug: chromaticity flag is NOT marked.)

The module also contains a rather simple, improvised optimizer to fit some of the momentum dependant functions to target values:

Up to four functions may be selected for automatic minimization by checking the box after the name of the function. It then appears as a button label in the minimizer panel at bottom. Clicking this button shows the function and the polynomial fit (in green) and opens a second column beside the fit coefficients to enter target values for the coefficients, which are then shown as pink crosses.

The minimizer is of Powell type and works on the penalty function

$$\sum_f \sum_{\delta} w_f (f(\delta) - f_T(\delta))^2,$$

with δ the array of momentum values covering the selected range, f the momentum dependent function, f_T the target for this function, and w_f a weighting factor. The latter is entered left of the button. Only nonlinear elements are available for minimization. The minimizer knobs are the strengths normalized to the initial values.

“Optimize” starts the minimization, “Break” interrupts it, “Reset” restores the initial values. “Plot absolute/relative” toggles between showing function and target, or the difference between both.

\Rightarrow This minimization was once implemented “quick and dirty” for a study on a non-linear bunch compressor and is not useful for storage rings, since Tayloring chromaticities in this rude way usually destroys dynamic aperture. The non-linear optimizer should be used instead (next section).

4.5 Design \rightarrow Non-linear optics

Sextupoles, octupoles and decapoles may be optimized in order to suppress resonance driving terms and compress the tune footprint while providing the desired chromaticity correction.

Starting the non-linear optics module requires to have a periodic solution. As much as possible is calculated only once when the module starts. This may cause some delay depending on the total number of sextupole kicks in the lattice (which depends on the number of slices per sextupole etc.) – a message in the main logging window will tell about. But afterwards the iterations will be fast.

Important note: the non-linear module works only **without coupling!** Lattices with weak (parasitic) coupling may work well anyway, but for strongly coupled lattice (e.g. Moebius insertion) one has to carefully check if the module can be used at least for some sections of the lattice.

Source of the resonance driving terms is the non-linear Hamiltonian. It is calculated in first and second order of sextupole strength, and in first order of octupole strength (for more details see “inside OPA” and the documents referenced therein). The tune footprint is controlled by amplitude dependent tune shifts (ADTS) and by chromaticity. Linear ADTS (which is a second order sextupole/first order octupole effect) and chromaticities up to third order are calculated. ADTS and second order chromaticity are second order sextupole and first order octupole effects. Third order chromaticity is a first order decapole effect.

The panel shows on the left side the quantities to be treated: the phase independent terms (ADTS and chromaticities) are real numbers with “Target” fields in order to adjust them to some desired value. The resonant terms are complex, showing only the absolute value, and should ideally be suppressed to zero. To each resonant mode exists a complex conjugate, which is not shown.

There are 10 terms in first order (2 linear chromaticities and 8 resonances), 13 in second order (2 quadratic chromaticities, 3 ADTS and 8 octupolar resonances), and the 2 cubic chromaticities. The first order modes of the octupole Hamiltonian are added as complex vectors to the second order sextupole modes. Chromaticity and path length calculation is done by numeric differentiation of the dispersive orbit, whereas all other quantities are calculated analytically.

As an additional term the sum of all sextupole strength is provided too. This turned out useful to get an overall reduction of higher orders.

The terms are calculated for one lattice period and multiplied with a complex (resonant) resp. scalar (phase independent) factor to be entered in the “periods” field at left bottom. A value of 0 means ∞ and is applied only to the resonant terms, of course.

All Hamiltonian modes are given in SI-units, i.e. normalized to betatron amplitudes of $2J = 1$ m and a relative energy deviation of $\Delta p/p = 1$. ADTS values are displayed as $\partial Q/\partial(2J)$. The relevant amplitudes are much smaller and may be entered in the three fields below the left panel in practical units of mm·mrad and %.

A tune diagram will have popped up when starting the module. In addition to the tune diagram

opening with linear optics, it shows the expected tune spread of the beam: the magenta/cyan parabola shows the tune variation due to chromaticity and the green straight lines due to ADTS using the relevant betatron amplitudes and momentum range.

The bars in the black fields of the left panel are products of the Hamiltonian modes with the relevant amplitudes and empirical weighting factors. Another (empirical) factor in the “Res x 10[^]” field magnifies the resonant terms to make them comparable, otherwise they would be invisible compared to the phase independent terms.

The weighting factors are given by $(2^w - 1)$ with w stepped up/down by the [+] [-] buttons. The bars provide a visual comparison of the different modes and set up the scalar penalty function for the minimizer.

The non-linear minimizer for the sextupoles is of Powell-type. The check boxes in the “inc” column of the left panel include the terms in the penalty function. The octupoles form a linear system which is solved by SVD for the terms selected by the checkboxes in the column marked “O”.

The right panel contains the available knobs. These are sextupoles and octupoles, and if selected (checkbox “incl.Comb”) also the combined function bends which may contain a sextupole component.

The values display the integrated strengthes of the sextupoles and combined function bends. Maximum strength and step size for pressing the [>], [<] buttons are given underneath. [>>], [<<] apply 20 steps. [off] sets to zero, [res] restores the initial value. The “lock” check box excludes the family from the minimizer.

For automatic adjustment of linear chromaticities to the target values, two sextupole families have to be selected by clicking the check boxes in the row labeled “ ξ ”. If these sextupoles have no dispersion, chromaticity correction will be impossible and an error message is shown. There may be a small deviation from the target value of chromaticity. This is due to the fact, that linear, quadratic and cubic chromaticities are obtained from numeric differentiation, whereas the chromatic sextupole values are calculated using the simple 2×2 matrix containing the well known sums over beta functions and dispersion.

If there are octupoles in the lattice, they will be shown underneath the sextupoles, and a column of check boxes labeled “O” appears in the left panel. Checking these boxes activates a singular value decomposition (SVD) routine which is controlled by the buttons appearing underneath the octupoles:

The “Condition” and “Nweight” labels shows the ratio of smallest non-zero to largest value of the weight vector and the number of non-zero values. The small [+] [-] buttons may be used to filter these values and improve the condition. The [SVD] button performs a calculation which can be canceled using the [undo] button. Checking the “auto” box enables automatic SVD after each change of a sextupole. Checking the “lock” box of an octupole excludes it from SVD.

If there are decapoles in the lattice, they will be shown underneath the octupoles. They only affect the cubic chromaticities and may be modified manually only.

At the bottom of the right panel is shown a cubical fit for the momentum dependant orbit length (longitudinal chromaticity). The numbers given are the first three orders of the momentum compaction factor multiplied by the lattice length.

The **[select]** button launches a separate window for visualization of the first order resonances: it displays the sextupole kick vectors in the complex plane. Pressing **[select]** again toggles steps through the 8 resonance modes which then will be highlighted in the left panel. The sum vector is indicated by a white circle.. A 9th plot shows the sum vectors of the 8 resonance modes for comparison.

The field “minimizer ini. step” sets the initial step for the minimizer (but actually has little effect, since the Powell algorithm adapts the step size anyway).

The field “dp/p num. diff [%]” sets the momentum range for chromaticity calculation from numeric differentiation. The cubic chromaticity may be critical to this value due to round-off errors, so one should try to find a stable value.

The **[Start]** button starts the Powell minimizer. If two sextupole families were selected to correct the chromaticity, they are not touched by the minimizer but set to maintain chromaticity after each step. The minimizer will inform on its progress by listing the current value of the penalty function (normalized to its start value) and also by writing messages to the message window on the OPA main panel. After successful termination, the penalty function is shown in blue, after interrupt in magenta.

The minimizer does not use the octupoles, however if the “auto” option for SVD is activated, the octupoles are set after each minimizer step.

“Exit” terminates the sextupole programs and asks if the new sextupole values should be saved. Note that the question to save the data refers only to an internal save to further proceed inside OPA, but does not save the data to the file.

4.6 Design → LGB-optimizer

The LGB (longitudinal gradient bend) editor is for optimization of the longitudinal field variation $B_y(s)$ of a dipole with regard to minimum emittance:

One may start with an existing LGB or create a new one. In the latter case, one has to select if half of a symmetric bending magnet is considered, or a dispersion suppressor magnet, which has zero dispersion at its entry. Then deflection angle, length and number of slices for representation by a stack of homogeneous, rectangular dipoles, are selected. Optionally, the minimum beta function in the LGB and its maximum field can be constrained by activating the corresponding fields and entering number.

⇒ *The field “half pole width” was introduced for a study on LGBs with transverse gradients, but not maintained further.*

Optimization will run Powell’s minimizer on the field of f the slices. Minimizations may be repeated until one likes the result.

The data at the lower right panel give results in red, and the values for a homogenous dipole in blue for comparison: “I5” is the fifth radiation integral, which can be reduced by and LGB. From this follows the “emittance”, also taking into account the increase of “radiated loss”, however damping partition is not included at this stage. “Beta” is the minimum beta function. For a symmetric LGB the minimum dispersion is given (“Disp”), for a dispersion suppressor LGB instead the “Focus position” where the minimum beta occurs. Finally the “energy spread” is displayed too, which is always increased in an LGB. Data can be written to a text file for further analysis.

The [Create new] button creates elements for the single slices of the LGB and a segment for their line-up for further use. Also existing LGBs found in the lattice may be selected, viewed or modified.

4.7 Design → Orbit Correction

If there are monitors and correctors in the lattice, this option becomes available. If monitors, horizontal and vertical correctors have the special names MON, CH and CV, the family is expanded in single elements names MONnnn etc., where nnn is a number. Only these elements will be used for orbit correction. Correctors or monitors with other names remain families and can only be used manually.

On startup the main Orbit panel appears, showing showing monitors and correctors as green, blue and red objects. The same start panel like in linear optics design pops up too, however it is reduced allowing only the orbit to be set as initial conditions.

Correctors can be dragged to knobs in the orbit panel to set them manually. A monitor can be dragged to the BPM-field. It will show its reading there (positions – and also angles, which is not possible in reality). By switching from “actual” to “Reference” a target value can be entered for positions x and y and is written to the BPM after pressing “set”.

The “status + statistics” panel at lower left contains a [Start] button to open the start menu again if it was close, and a checkbox to include nonlinear elements. Two labels tell if there is a periodic (or single-pass) solution or if no orbit was found. This may happen if non-linear elements are switched on (In this case switch off, perform correction, and switch on again.)

The statistics panel shows mean, rms and max values for the orbit relative to the references (“BPM”), for the absolute orbit at all elements (“all”) or for the corrector kicks (“Corr”) – press to toggle.

The “plot” panel at lower right allows to choose between plots of orbit and BPM, correctors and misalignments. If “keep max.” is checked, no autoscaling of the plot is done (this is nice to see the effect of orbit correction!) The [print] button should write all data to a file (\Rightarrow *not yet implemented*).

The “misalign and correct” panel sets misalignments (transverse displacements and roll errors) and performs orbit corrections: Misalignments are entered in micron resp. micro-rad and will be applied to all magnets (but the correctors) as Gaussian distributed random numbers with the given cut-off in sigma (field “ σ ”). Errors can be applied to elements, to girders and for girder joints: elements are located on girders, so a girder misalignment will cause a correlated elements misalignments. Girders are assumed to be connected to adjacent girders by [virtual] joints, which may have some play (see comment q) in sec.3.3 above).

The field “seed” determines the series of random numbers. The [set] button applies the misalignments, z[zero] sets them to zero, and [re-set] applies a previous setting again (to save retyping the input fields).

[correct] calculates the response matrix, sets-up the SVD and performs the orbit correction. The results as shown underneath the bootton are “COD”: no correction done,
“zeroed”: success: orbit agrees exactly with the reference everywhere,
“minimized”: success: orbit does not agree with reference, but does not converge any further. This is always the case if there are less correctors than monitors.
“no convergence”: failure: too many iterations, the orbit loop did not converge.
“failed”: complete failure, the iterations diverged.

The buttons [Corr=0] and [BPM=0] set all correctors, resp. all BPM references to zero.

The figure displays the SVD weight factors, where the excluded (zero) values are shown in darker color. The small button toggles [X] and [Y], and the slider allows to reduce the weighting factors to realize a “soft” correction which is incomplete but lowers the corrector strength.

When the [Loop] button is pressed, the “Seed” field label switches to “Nseed” and the number entered will be the numbers of orbit corrections to be automatically performed for a couple of seeds. The main logging window will inform about the progress: the number displayed tells about the result, which is the sum of two numbers:

00: nonlinear orbit found.

10: nonlinear orbit found but correction failed, correct linear orbit first, switch on non-linear and correct again.

20: nonlinear orbit NOT found, linear found and corrected, switch on non-linear and correct again.

30: nonlinear orbit NOT found, linear found and corrected, but lost again after switching on non-linear, correction failed.

40: even linear orbit NOT found, lattice unstable.

+0: successfull correction, zero at BPMs.

+1: almost successful, not zero in all BPMs (minimized).

+2: correction didn't converge, too many iterations.

+3: orbit lost in correction.

Afterwards the “Nseed” field gives the number of successful orbit corrections (\implies *not yet implemented?*), and the statistics button in the lower left panel now give at “mean” the mean rms orbit from all seeds, at “rms” the maximum rms orbit from all seeds, and at “max” the absolute maximum excursion from all seeds.

When closing the orbit module misalignment and corrector settings are kept in the lattice, and the corresponding flags are highlighted in the left panel of the main menu. Subsequent linear optics or tracking calculations will be based on the distorted (and perhaps corrected) orbit. In order to “clean” the lattice, correctors and misalignments have to be set to zero in the orbit module before leaving.

4.8 Design \longrightarrow Injection Bumps

The injection module is very similar to the orbit module, but here the start menu only allows the “single pass” option, and instead of the correctors, the kickers appear now on the panel as orange symbols and may be dragged to knobs.

\implies *Actually, the panel IS the orbit panel with some modifications, because it had to be realized in short time. Some corrector entries are still visible, however have no meaning. This should be cleaned up.*

Initial orbit coordinates should be zero to see the impact on the stored beam and non-zero for the injected beam.

The [ON/OFF] button in the lower middle panel activates the kicker. The [Sync] button adjusts the delay (see comment o) in 3.3) to the time the beam passes by, which is just given by the longitudinal position (“Spos[m]” in the plot) divided by speed of light (OPA is high-relativistic).

If the lattice is a closed ring (flag “circular” in main menu status panel), the [+] [-] buttons to show more turns (max.3) are enabled.

When leaving the injection module, the kicker settings are saved and will be used in linear optics design (if kickers are active) and in phase space tracking (\rightarrow 4.11).

4.9 Design \longrightarrow RF-bucket View

This module plots the bucket for voltage up to fifth harmonic and momentum compaction factors up to fifth order. The bucket is shown by the separatrix (thick dark green line) with its hyperbolic and elliptic fixpoints as red and blue symbols.

If a fit was done to pathlength in a previous run of the **Off-momentum** module (\rightarrow 4.4) the data are saved internally (flag “pathlength” in main menu status panel) and appear now in the “MCP” column of the upper panel.

The lower panel contains lattice parameters affecting the bucket, like energy, circumference, radiation loss and the harmonic number (= circumference/wavelength). The [Lattice] bucket allows other parameters to be tested or to switch back to the lattice parameters. The remaining three parameters mainly control the appearance of the plot.

The program tries to guess momentum acceptance and bunchlength from the plot and writes it to the bottom of the panel. The acceptance is shown as bright green bars in the plot, bunch length is not shown because it is too small usually.

If acceptance is not found, probably parts of the separatrix are outside the plot window, so the “dp/p max” value should be increased.

Bunch length is defined by linear synchrotron oscillation and thus can only be determined if the central fix point is elliptic. If it still is not found it may help to increase the resolution of the plot by adjusting the “N.points” field, because bunch length is estimated from an (invisible) elliptic equipotential very close to the origin.

When closing the module, the RF momentum acceptance values are saved (flag “RF bucket” in main menu status panel) to be used in subsequent Touschek lifetime calculations (→4.13).

4.10 Design → Geometry Layout

A plot of the lattice layout: press and hold the left mouse button to mark a rectangle to zoom in. The curved arrow buttons on top will follow the lattice back or forth, the third button zooms out again.

If there are marker elements named “center” in the lattice, the plot will be centered at the center of gravity of these markers.

Unchecking the “fix aspect ratio” box stretches the plot horizontally and vertically to the canvas, otherwise (default) same scales are used horizontally and vertically.

Parameters in the table control width (and length of some) elements.

“wmf” writes a file (file)_geometry.wmf, “data” writes three files, named markers.txt, devices.txt and orbit.txt containing data of points where a new straight or curved section starts, the data of the polygons relative to the markers and the data of the orbit. Markers and orbit can be visualized by checking the corresponding boxes.

4.11 Tracking → Phase space

This panel shows Poincaré plots of particle motion in horizontal and vertical phase spaces (x, x') and (y, y') (*it is x', y' although the axis are labeled p_x, p_y*). The circles or ellipses show the linear acceptance given by the physical aperture. When starting a particle, the effective apertures are also shown, which are reduced due to coupling assuming elliptical beam pipes (see sec.??).

By left-clicking in the diagrams or by entering numbers in the centre panel, the starting con-

ditions are selected. A momentum deviation may be entered in the "dp/p" field. "run" tracks the particle a number of turns as given in the top panel, "more" adds more turns, "clear" clears the panel, "exit" terminates the program.

A Fourier spectrum of particle motion is calculated after tracking and shown in the lower diagrams. The algorithm used is FFT with sine window and $\sin x/x$ peak interpolation to obtain frequency and amplitude of the peak in order to identify (guess) the underlying resonance [?]. Results are shown in the lower panel. The tune considered as the fundamental is also marked in the tune diagram. The "param" button opens a panel to set parameters for resonance guessing: "fundamental tune range" sets the maximum deviation of the guessed tune from the linear tune. "peak identification tolerance" sets a parameter p defining $\Delta Q = p/c$ as the maximum tune deviation, with c the order of the resonance. The two filter levels are for accepting peaks and for the range of the plot.

By default, tracking uses the apertures of the elements to test on particle loss. By unchecking "element apertures", the apertures in the "Ax", "Ay" fields are used instead. Aperture check assumes an elliptical beam pipe and tests for $(x/a_x)^2 + (y/a_y)^2 \leq 1$. Note, that the aperture check may underestimate losses, because internally, sequences of linear elements are concatenated into matrices at start. Aperture checks are only done at locations of nonlinear kicks and at the track point. Therefore, particle tracking may also paint outside the linear acceptance ellipse.

The trackpoint by default is at the begin/end of the lattice, but the field "trackpoint" allows to select any location. The ellipses indicating the physical acceptance will change accordingly.

The "amplitude dependent tune shifts" panel performs three series of trackings stepping up the initial angles x', y' in steps as given in the "steps" field until the maximum betatron amplitudes given by physical acceptance, are reached. The first (third) series goes in horizontal (vertical) direction up to the maximum betatron amplitude with a tiny vertical (horizontal) amplitude in order to excite coupling. The second series sets the initial coordinates for a constant coupling. Coupling and maximum amplitudes are defined in sec.??.

The identified tunes are marked during execution in the tune diagram, and afterwards they are plotted vs. betatron amplitude, or vs. "ping", i.e. the initial kick angle. The three series correspond to the lines blue-X, lilac-X (both showing $\Delta\nu_x$ vs. $2J_x$) and purple-Y (showing $\Delta\nu_x$ vs. $2J_y$) in the left diagram, and to the lines lilac-X (showing $\Delta\nu_y$ vs. $2J_x$), purple-Y and red-Y (both showing $\Delta\nu_y$ vs. $2J_y$) in the right diagram.

Since the amplitude is derived from the height of the fundamental peak in the Fourier spectrum, it may happen in case of very strong non-linearity, that the curve "returns" towards the origin, if a harmonic peak grows on expense of the fundamental peak. It may also happen, that the tune gets stuck on a resonance (i.e. the particle is trapped in an island in phase space) and the tune shift vs. amplitude shows a flat line.

If theoretical amplitude dependent tune shifts have been calculated previously in the sextupole module, they are shown as dotted line for comparison.

4.12 Tracking → Dynamic Aperture

There are three modes: x vs. y , x vs. $\Delta p/p$ and y vs. $\Delta p/p$. When selecting the first, a $\Delta p/p$ offset may be given, for the others it is a $\Delta p/p$ range. The plot will show the geometric acceptance based on linear beam dynamics and the grid for probing stability by starting particles at these locations. Grid parameters may be stepped up and down by the "cells" buttons. The geometric acceptance is modified by setting the apertures as described in sec.???. Other parameters are the number of turns (*and the location of the trackpoint*).

"Start" starts probing the grid: surviving particles are shown in green, lost ones in red. "Export" writes graphic files (name)_dynap_xy/_xp/_yp.wmf. Double-clicking the image copies it to the clipboard.

In x vs. y mode, the geometric acceptance is shown as a blue polygon. It is given by the common area of the projections of all the elliptical apertures to the track point (see appendix ??). This geometric acceptance includes non-linear and chromatic contributions to orbit and beta functions but else assumes linear transformations. In tracking, it may happen, that particles outside the geometric acceptance survive for two reasons: 1) aperture checks are only performed at non-linear elements (because to speed up tracking series of linear elements are concatenated and stored as matrices), and 2) the non-linear eigenfigure of betatron motion may be different from an ellipse.

4.13 Tracking → Touschek lifetime

This module for Touschek lifetime calculation proceeds in two steps:

1. Work sheet to calculate lifetime related parameters in a linear model: one may enter parameters in the left panel and see the results in the right panel and in the plots. Available plots include

- the beta functions,
- the lattice invariant \mathcal{H} ,
- the rms beam envelopes,
- the apertures,
- the bunch volume,
- the momentum acceptances from RF (green) and from apertures (brown),
- the ζ -parameter for the Touschek function (see section ??)
- the local loss rate in linear scale, and
- the local loss rate in logarithmic scale.

2. Tracking for local dynamic momentum acceptance: after pressing "track", the program will step along the lattice in steps of Δs and start particles on-axis but with momentum deviation

$\Delta p/p$ to simulate Touschek scattering. A binary search on $\Delta p/p$ determines the minimum and maximum values of $\Delta p/p$ accepted at the particular location. Only locations where \mathcal{H} (and its nonlinear equivalent) has changed need to be tested, therefore the loop jumps over non-dipole elements and just copies the previous momentum acceptance data.

During execution, the plot will switch to momentum acceptance to show the progress of the calculation. "Break" allows to interrupt. When done, all calculations for ζ , loss rates and lifetime like in the linear case are done and the results are added to the plots. In the ζ plot, 2 additional lines for positive (red) and negative (blue) dynamic momentum acceptance will appear. In the loss rate plot it is only one line for the total losses (pos. and neg.).

Input parameters:

- Energy may be changed for testing, but the original value will be restored on exit.
- Coupling is the emittance ratio.
- Total beam current in the machine, and
- number of bunches give the charge per bunch.
- The cavity voltage defines bunch length and RF momentum acceptance. If it is lower than the energy loss per turn, the bunch length is set to the ring circumference and the momentum acceptance is zero.
- The harmonic number contains RF wavelength. The number of buckets can not be higher than the harmonic number, this is checked.
- The longitudinal resolution defines Δs for tracking along the lattice.
- Momentum resolution defines the precision of the binary search.
- Numbers of turns to track to decide on loss or survival.
- Residual gas is defined by atomic number, atoms per molecule and partial pressure.

The gray fields show quantities derived from the input values, which can not be modified.

"Track" starts the tracking, which is a 4-D tracking, i.e. keeping the momentum constant.

\implies "Track sync" is an improvised 5-D tracking, varying the momentum deviation $\delta = \Delta p/p$ in turn k simply as $\delta_k = \delta_o \cos(2\pi\nu_s k)$, with ν_s the synchrotron tune. The implementation is very inefficient, and therefore this tracking is rather slow.

"Export plot" writes a file `(name)_touschek_N.wmf`, where $N = 0..8$ is the number of the plot ($N=0$ for beta function plot etc.).

"Export data" writes a file `(name)_touschek.txt` with all plotted data.

"ZAPLAT" writes a file `(name)_zaplat.dat` to be used as input file `zaplat.dat` by the ZAP code [?] for calculation of intra beam scattering. Note, that aperture data as needed for Touschek lifetime calculations by ZAP are not included.

4.14 Extra → Currents

If allocation and calibration files were given in the lattice file (see sec.??), the magnet currents are calculated and shown. "Power supply" is the name of a magnet family, "N" is the number of magnets in this family, and "Current" is the current, which is the average of the current calculated for the single family members (differences may be due to different magnet types connected in a family). Clicking a field shows at right the family member data.

"Snap export" writes `(file).snap` for uploading to EPICS control system

⇒ *This is for internal use at SLS and probably would have to be modified for other machines.*

.