TraceWin

D. Uriot, N. Pichoff

Université Paris-Saclay, CEA, Département des Accélérateurs, de la Cryogénie et du Magnétisme, 91191, Gif-sur-Yvette, France.

Updated, December 03, 2021

Abstract

The TraceWin code calculates the **beam dynamics in particle accelerator**. The beam is modeled both by its **second order momentum** (fast calculation, in linearized force) or/and by a **macro-particle distribution** (longer calculation, in non linear forces). Their simultaneous use allows easy study of the impact of non linear effects.
The **different elements of a linac** can be modeled eiher using **analytic expression** or **field map**s. The code is able to run **automatic procedures** of accelerator and beam tuning including statistics errors on elements and the diagnostics.
TraceWin used a very **powerfull GUI** able to deliver a large variety of plots. The user can change any parameter and observe the effect very easily with the very **powerful graphics display** which allows to visualizing most of the useful parameters of the simulation (envelopes, beam ellipses, emittances, phase advances…). All these outputs can be easily stored on disk, save in **several image formats** and inserted into reports (using **copy and paste tools**).
A huge number of cases can be simulated remotely via home made client/server architecture. A heterogeneous array of machines can be used (**window**, **linux**, **MacOS**). It has been mainly written in C++ and Qt5.4 for Windows, Linux and MacOS operating system. Started in 1998 (URIOT Didier and PICHOFF Nicolas), it is distributed since 2009 under **CEA license**.

Technical support & maintenance services

Updates of the software

The CEA agrees to notify the Licensee of any Updates and Corrected Versions of the Software. The content of the Updates shall be decided by the CEA in its entire discretion, and shall comprise fixes for anomalies, improvements and upgrades to make the Software correspond to the most recent version of the Software. However, Updates may not include: versions of the Software that are compatibles with new operating systems, improvements, updates or new versions that are sold separately.
Updates and Corrected Versions shall be made available to the Licensee, in executable form, in the manner indicated by the CEA by email.

Corrective maintenance

The CEA agrees to correct reproducible errors in software programming. The Licensee shall notify the CEA of any unidentified error and provide it with all information necessary for error reproduction. The resources implemented by the CEA to correct an error shall depend on its seriousness.
If the error stems from a modification that was not made by the CEA or expressly authorized by it, the Licensee agrees to compensate the CEA for the time spent on correcting the error and restoring the Software to good operating condition, at the CEA rates then applicable.

Technical support

The CEA shall provide technical support to the Licensee during its normal office hours, Mondays to Fridays (excluding bank holidays and annual weeks of closure). The CEA shall use its best efforts to provide a detailed response within a maximum period of thirty (30) business days depending on its human resources available at the time of the request.
Requests should be sent to the CEA by email to contact address defined below. The code can be downloaded here: http://irfu.cea.fr/Sacm/logiciels/ .

Evolutive maintenance

The CEA shall add new functions to the Software where required by a change in the legal or regulatory requirements or in the software or hardware environment. New Versions shall be made available to the Licensee, in executable form, in the manner indicated by the CEA by email

Contact

This version of **TraceWin** is supported by **Didier URIOT**. We would appreciate hearing from you if you found a bug. Your questions or remarks can be sent at the following Emails address:

didier.uriot@cea.fr

Very important: How report a bug: Please use the button "*Send project*" in main page and include the specified files. By this way we are able to know which system you used and which version produced the bug and finally with the include files we can easily reproduce it, thus fix it

New: since December 2020, our communication policy regarding hotline, query, bugs, help or more generally any question related to this software has evolved. We now provide exchanges via a forum specially set up for this purpose.
https://dacm-codes.fr/forum/

Installation

Minimum configuration required

1 gigahertz (GHz) or faster 32-bit (x86) or 64-bit (x64) processor
1 gigabyte (GB) RAM (32-bit or 64-bits)
30 megabytes (MB) available hard disk space (32-bit or 64-bit)

Operating system**:**

Window (32-bit or-64-bit), version equal or bigger then **WinXp**
Linux (32-bit or-64-bit) with GLIBC library version equal or bigger **2.6** (“*ldd --version*”)
MacOS with system version equal or bigger **10.6**

Remark about 64 bits Linux OS:

You're on a 64-bit system, you probably don’t have 32-bit library support installed. It’s mandatory for the defaut WebBrowser, QtWeb, internally used to get direct access to manual and also if want to use a 32bits TraceWin version.

To install (baseline) support for 32-bit executables

(if you don't use sudo in your setup read note below)
Most desktop Linux systems in the Fedora/Red Hat family:

```
pkcon install glibc.i686
```

Possibly some desktop Debian/Ubuntu systems:

```
pkcon install ia32-libs
```

Fedora or newer Red Hat, CentOS:

```
sudo dnf install glibc.i686
```

Older RHEL, CentOS:

```
sudo yum install glibc.i686
```

Even older RHEL, CentOS:

```
sudo yum install glibc.i386
```

Debian or Ubuntu:

```
sudo apt-get install ia32-libs
```

should grab you the (first, main) library you need.

Once you have that, you'll probably need support libs

Anyone needing to install glibc.i686 or glibc.i386 will probably run into other library dependencies, as well. To identify a package providing an arbitrary library, you can use

```
ldd /usr/bin/YOURAPPHERE
```

if you're not sure it's in /usr/bin you can also fall back on

```
ldd $(which YOURAPPNAME)
```

The output will look like this:

```
linux-gate.so.1 =>  (0xf7760000)
libpthread.so.0 => /lib/libpthread.so.0 (0xf773e000)
libSM.so.6 => not found
```

Check for missing libraries (e.g. libSM.so.6 in the above output), and for each one you need to find the package that provides it.

Commands to find the package per distribution family

Fedora/Red Hat Enterprise/CentOS:

```
dnf provides /usr/lib/libSM.so.6
```

or, on older RHEL/CentOS:

```
yum provides /usr/lib/libSM.so.6
```

or, on Debian/Ubuntu:
first, install and download the database for apt-file

```
sudo apt-get install apt-file && apt-file update
```

then search with

```
apt-file find libSM.so.6
```

Note the prefix path /usr/lib in the (usual) case; rarely, some libraries still live under /lib for historical reasons … On typical 64-bit systems, 32-bit libraries live in /usr/lib and 64-bit libraries live in /usr/lib64. (Debian/Ubuntu organise multi-architecture libraries differently.)

Installing packages for missing libraries

The above should give you a package name, e.g.:

```
libSM-1.2.0-2.fc15.i686 : X.Org X11 SM runtime library
Repo        : fedora
Matched from:
Filename    : /usr/lib/libSM.so.6
```

In this example the name of the package is libSM and **the name of the 32bit version of the package is libSM.i686**.
You can then install the package to grab the requisite library using pkcon in a GUI, or sudo dnf/yum/apt-get as appropriate…. E.g pkcon install libSM.i686. If necessary you can specify the version fully. E.g sudo dnf install ibSM-1.2.0-2.fc15.i686.
Some libraries will have an “epoch” designator before their name; this can be omitted.

Warning

Incidentially, the issue you are facing either implies that your RPM (resp. DPkg/DSelect) database is corrupted, or that the application you're trying to run wasn't installed through the package manager. If you're new to Linux, you probably want to avoid using software from sources other than your package manager, whenever possible...

If you don't use "sudo" in your set-up

Type

```
su -c
```

every time you see sudo, eg,

```
su -c dnf install glibc.i686
```

Installation and start:

No installation is necessary; all the extra files used by TraceWin are directly extracted from main code and installed if a process requires it.
The downloadable version is not a full capability version. To activate its full capabilities you have to include in the TraceWin executable directory a key file: “*tracewin.key*”.

Since TraceWin version 2.4.1.0 :
The activation key is automatically generated when the code is launched, if it is connected on the WEB and if the user has been registered in the database of users. In order to use the code on a computer with no network access, you have to bring this key file with the executable to keep the user rigths.

Run the code directly by double clicking on the executable file or on the bundle for MacOS. For run from a batch command file see chapter “*Use TraceWin batch command file*”.

- TraceWin is based in part on the work on ***Qt*** project (http://qt.nokia.com) with Qt commercial license agreement verion: 5.4
- TraceWin is based in part on the work of the ***Qwt*** project (http://qwt.sf.net)
- Documentaion browser is made by QtWeb developed as a project of LogicWare & LSoft Technologies (http://qtweb.net/)

Main features

- A wide range elements including RFQ with Toutatis module,
- 2D and 3D space-charge,
- 1D, 2D and 3D static and RF magnetic or/and electric field maps (with superposition capability),
- envelope simulation,
- macro-particle tracking simulations (number of particle depending of your free memory),
- each particle has an can be detailed analysis of its trajectory is available,
- start-to-end simulations from source to target,
- transport of two beams in the same structure,
- gas stripping and scattering analysis,
- automatic transverse and longitudinal beam tuning in envelope or/and tracking mode,
- beam tuning in period structure based on smoothing phase advances,
- correction procedure based on diagnostics,
- static and dynamic error simulations for all elements,
- simulations with large number of particles for large scale computations (Monte Carlo) based on a client/server architecture,
- statistical analysis including beam loss location,
- GUI and various help tools,
- Windows/Linux/MACOS versions,
- reference code for IFMIF, LINAC4, SPIRAL2, EUROTRANS, EURISOL and SPL projects.

Way of using

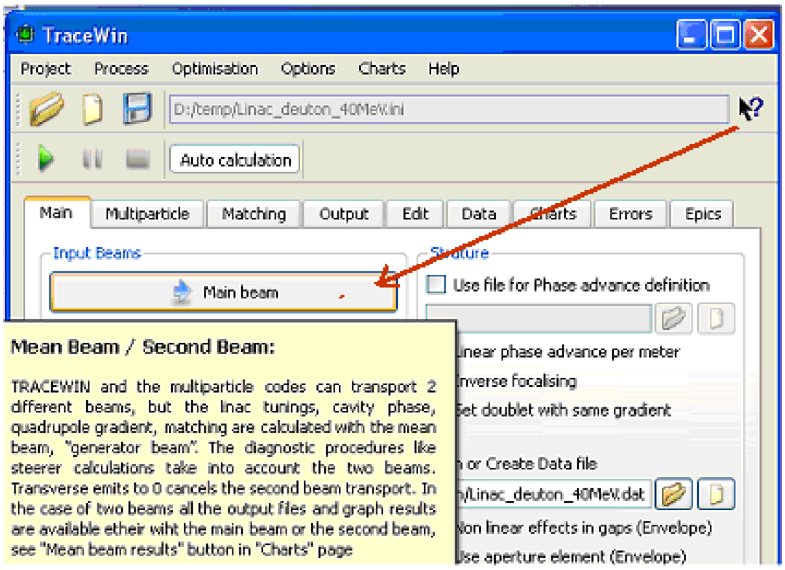
General description and help

TraceWin's program is organized in 8 pages and 3 toolbars. You can find more details about these pages below.

Menu : Shows the 20 last current opened projects.
First ToolBar : Open save or create a project file (configuration file, *.ini), show the current open project.
Second ToolBar : To launch the process, to have a break or stop it and set "***auto_calculation***".
Third ToolBar : Visible only during matching, to stop or visualize criteria variation.
Main page : To set input beam parameters, structure options and calculation options.
Multiparticle page : To configure multiparticle code options.

Matching page : To configure beam matching options
Output page : To visualize the calculation stages
Edit page : To modify or visualize the main input and output files.
Data page : To visualize the elements and commands list from the data file
Charts page : To visualize the results with plots.
Errors page : To parameterize the error study and visualize results.
Epics page : For EPICS virtual machine configuration

Each input or widget item of TraceWin GUI owns explanatory text. The default way for users to view the help is to move the focus to the relevant widget and press Shift+F1. The help text appears immediately. A second ways is to use the help button, see following picture.



Mean Beam / Second Beam:

TRACEWIN and the multiparticle codes can transport 2 different beams, but the linac tunings, cavity phase, quadrupole gradient, matching are calculated with the mean beam, "generator beam". The diagnostic procedures like steerer calculations take into account the two beams. Transverse emits to 0 cancels the second beam transport. In the case of two beams all the output files and graph results are available etheir wiht the main beam or the second beam, see "Mean beam results" button in "Charts" page

Code stages

TraceWin process is organized in several stages. The stages can automatically run one behind the other or not (“***Auto calculation***” button of ToolBar). Some of them can be disabled according to some options and commands.

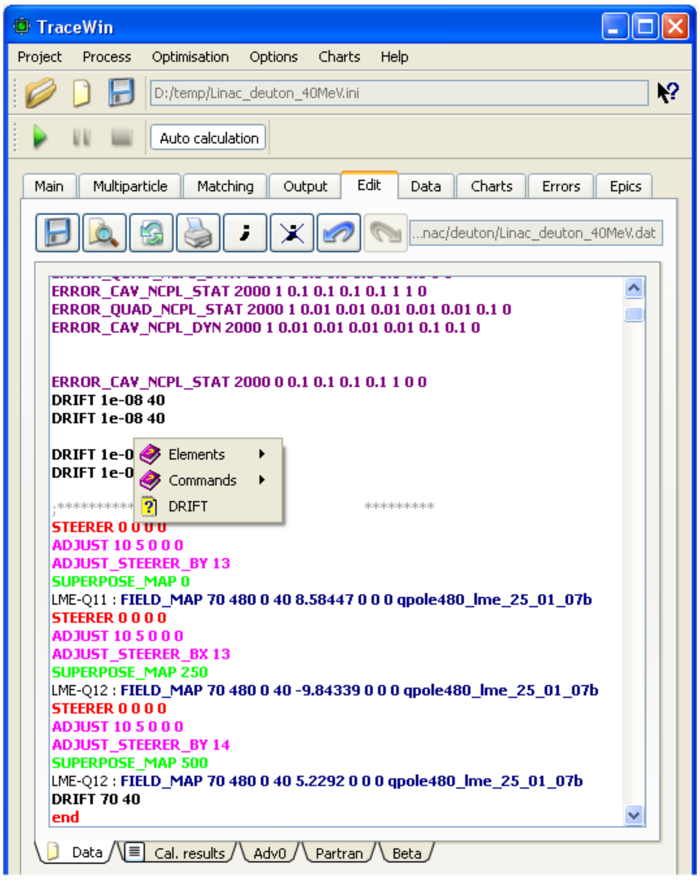
			Results saved	**Affected or used beam**	
	Different stages	**Needed Condition(s)**		**Main**	**Second**
1	Read input data file and set tab in “***Data***” page	Data file			
2	Transport of the reference particle			X	
3	Set Phase advance law (set Quadrupole, Solenoid, Field map strengths)	SET_ADV commands in data file or “***Use file for phase adv definition***” checked LATTICE commands in data file		X	
4	Read particle files	Particle file defined in “***Main***” page		X	X

5	Calculate the input matched beam	"Calculate match beam" checked in "Matching" page		X	X
6	Set quads or cavity strengths to match the beam through the different linac sections or set Twiss parameters (In order of theirs positions)	"Matching with family" checked in page "Matching" MATCH_FAM commands and (LATTICE or SET_TWISS command) in data file	X	X	
7	Diagnostics (Example: Steerers) calculations (In the order of theirs numbers)	"Match with diagnostics" checked in page "Matching" Diagnostic elements and Adjust commands in data file	X	X	X
8	Input beam distribution (*.dst) is adjusted in order to fit the input beam defined in "***Main***" page or to fit the input matched beam calculated in (5)	"Calculate match beam" -> "With partran" checked in "Matching" page "Use particle file" in "Input distribution type" in "Multipart" page Particle file defined in "***Main***" page			
9	Repetition of the preceding stages (5)(6)(7) using mutliparticle code	On or several options "***With partran***" checked in "***Match***" page	X	X	X
10	Random errors generator initialised	"Reinitialize random generator" checked in "Main" page.			
11	Apply Static errors	"Include error defined in…" checked in "Main" page "***..Data No***" in "***Main***" page set to errors defined in "***Errors setup***" of page "***Error***" "***ERROR_xxx_STAT_xxxx***" commands in data file		X	X
12	Diagnostics (Example: Steerers) calculations	"Match wiht diagnostics" in "Match" page Diagnostic elements and ADJUST commands in data file		X	X
13	Repetition of the preceding stages (10 to 13) using mutliparticle code	"Launch Partran" checked in "Multiparticle" page		X	X
14	Apply dynamic errors	"Include error defined in…" checked in "Main" page "***..Data No***" in "***Main***" page set to errors defined in "***Errors setup***" of page "***Error***" "***ERROR_xxx_DYN_xxxx***" commands in data file		X	X
15	Calculates the transport line envelope	Always		X	X
	Losses and beam parameters variations estimated in envelope transport	"***Nbr of particles***" in "***Main***" page greater then 10 "***Use aperture element***" checked in "Main" page		X	X

16	Write new data file in "*calculation directory*"	Always			
17	Make the Error studies (envelope). N linacs, Loop with stage (10,11,12,14,15)	*"**Study Envelope**"* checked in "***Errors***" page and error selection done		X	X
18	Make the Error studies (Particles). N linacs, Loop with stage (10,11,12,13,14,15,19)	*"**Study Multipartilce**"* checked in "***Errors***" page and error selection done		X	X
19	Write input files of multiparticle codes PARTRAN and TOUTATIS and launch them	"Launch Partran or Toutatis" checked in "Multiparticle" page		X	X

Editor

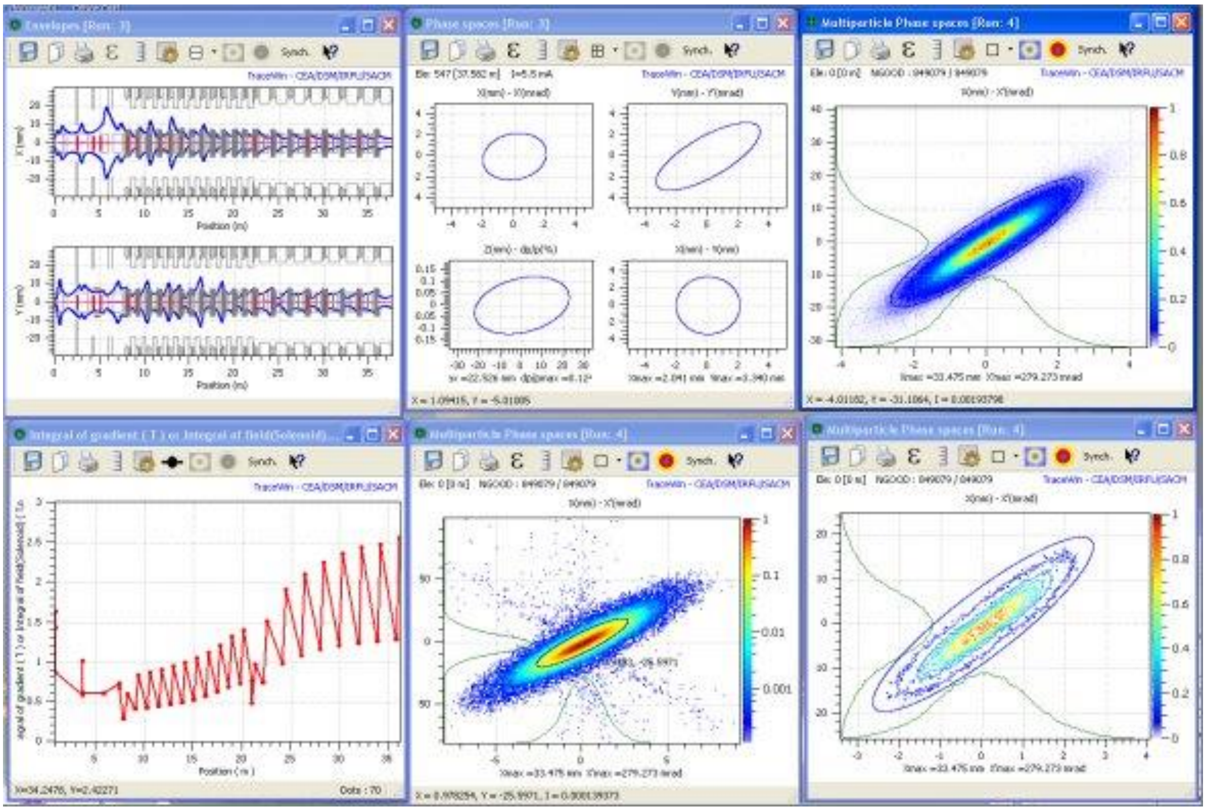
Some help about elements or commands in the data file editor can be obtained by using the right mouse button. If you don’t release it; the element number is shown before.



```
ERROR_CAV_NCPL_STAT 2000 1 0.1 0.1 0.1 0.1 1 1 0
ERROR_QUAD_NCPL_STAT 2000 1 0.01 0.01 0.01 0.01 0.01 0.1 0
ERROR_CAV_NCPL_DYN 2000 1 0.01 0.01 0.01 0.01 0.1 0.1 0


ERROR_CAV_NCPL_STAT 2000 0 0.1 0.1 0.1 0.1 1 0 0
DRIFT 1e-08 40
DRIFT 1e-08 40

DRIFT 1e-0
DRIFT 1e-0

;**********                          *********
STEERER 0 0 0 0
ADJUST 10 5 0 0 0
ADJUST_STEERER_BY 13
SUPERPOSE_MAP 0
LME-Q11 : FIELD_MAP 70 480 0 40 8.58447 0 0 0 qpole480_lme_25_01_07b
STEERER 0 0 0 0
ADJUST 10 5 0 0 0
ADJUST_STEERER_BX 13
SUPERPOSE_MAP 250
LME-Q12 : FIELD_MAP 70 480 0 40 -9.84339 0 0 0 qpole480_lme_25_01_07b
STEERER 0 0 0 0
ADJUST 10 5 0 0 0
ADJUST_STEERER_BY 14
SUPERPOSE_MAP 500
LME-Q12 : FIELD_MAP 70 480 0 40 5.2292 0 0 0 qpole480_lme_25_01_07b
DRIFT 70 40
end
```

Different king of plots, **<u>explore and test</u>** all the top buttons to well understand all available options.

Some output details


Envelopes [Run: 3]
Phase spaces [Run: 3]
Multiparticle Phase spaces [Run: 4]
Integral of gradient (T) or Integral of field(Solenoid)...
Multiparticle Phase spaces [Run: 4]
Multiparticle Phase spaces [Run: 4]
Synch.
TraceWin - CEA/DSM/IRFU/SACM
Position (m)
X(mm) - X'(mrad)
Y(mm) - Y'(mrad)
Z(mm) - dp/p(%)
X(mm) - Y(mm)
Position (m)
Dots : 70

Zoom in by dragging a rectangle with mouse left button.
Zoom out with left double click.
Move the plot area (only after zooming), with mouse right button.
Save by clicking on dedicated icon in several picture formats, pdf, ps, data ASCII file.
Copy by clicking on dedicated icon in png format.
Plot options, color, size, font, dots type… are configurable from icons □ · ●.
All charts can be **synchronized** (refreshed after each calculation) by clicking on the dedicated icon.
The plot window can contains a maximum 6 plots using this icon ⊟ ·.

On **envelope plots**, contextual information can be obtained by right click on each element; the envelope types (*X, Y, X', Y', Z, Z', Phase, Energy, Z, dp/p…*) by right click on the plot.

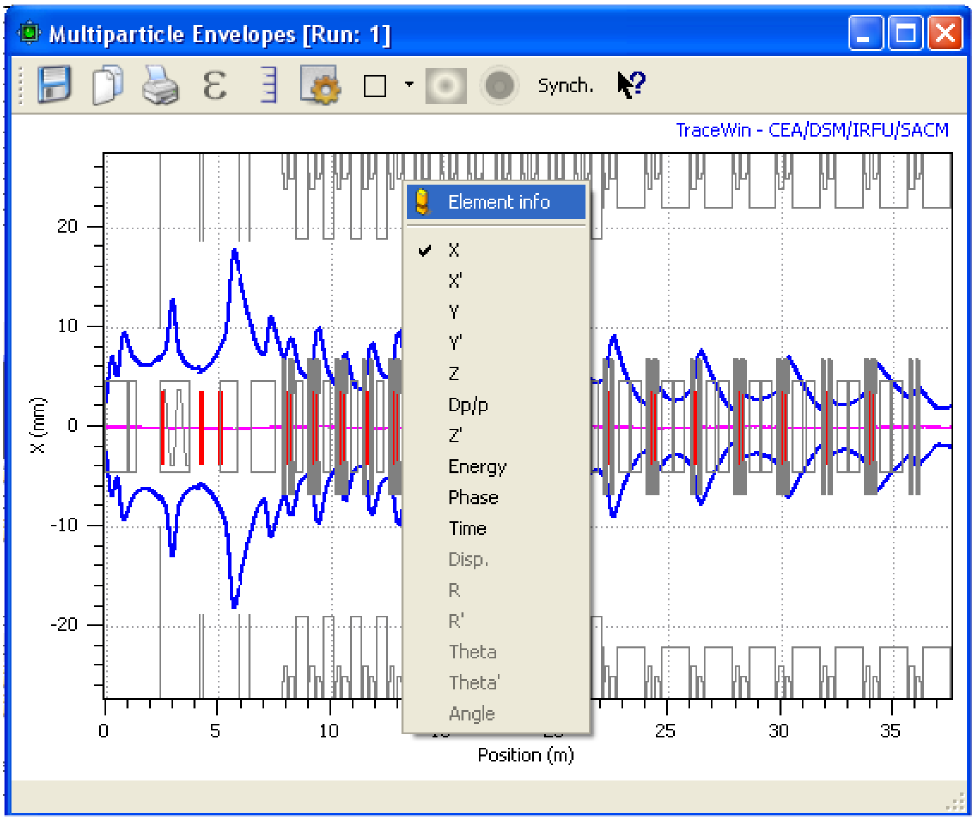
In **phase-space plots**, the selection of the **coordinates** is obtained by right click on the chart. In this case, one can either select one of the 8 proposed 2D phase-spaces or chose the variable for each axis. One can "**Select"** or "**Unselect"** the visible particles on the chart area. The sizes, colors of the selected or non selected particles are configurable by using the option button. This is very convenient to localize some particles in 6D phase-space and study their behavior in a line.

The emittance icon ε calculates and shows statistic information on plotted distribution. The code runs the same calculations for all the phase-space [*x*, *y*] plotted, whatever they are:

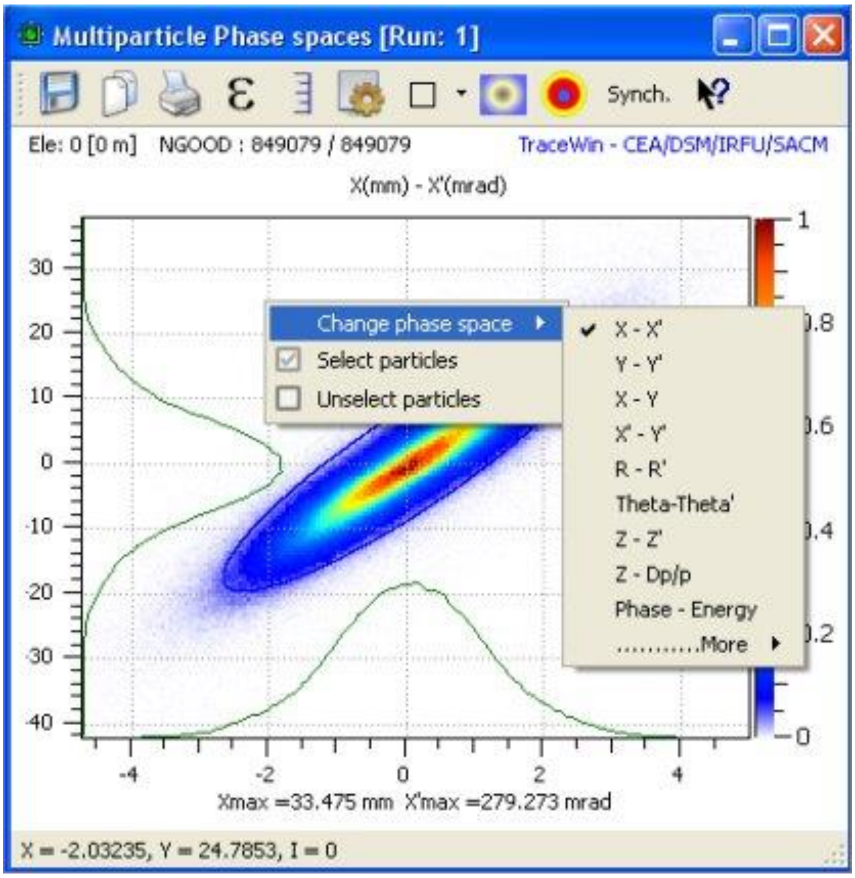
- $\varepsilon_{xy\_rms} = \sqrt{\left\langle \left(x-\langle x\rangle\right)^2 \right\rangle \cdot \left\langle \left(y-\langle y\rangle\right)^2 \right\rangle - \left\langle \left(x-\langle x\rangle\right)\cdot\left(y-\langle y\rangle\right) \right\rangle^2}$,

- $\beta_{xy} = \dfrac{\left\langle \left(x - \langle x \rangle \right)^2 \right\rangle}{\varepsilon_{xy\_rms}}$
- $\alpha_{xy} = \dfrac{-\left\langle \left(x - \langle x \rangle \right) \cdot \left(y - \langle y \rangle \right) \right\rangle}{\varepsilon_{xy\_rms}}$

Emit [xx%] gives the ellipsis surface, divided by π, homothetic to the rms ellipsis, containing xx% of the beam particles. The xx is either the calculated faction of particles inside a homothetic ellipses whose area is N times the rms ellipses, or a given fraction of the beam. The plotted ellipses are these last ellipses. Position of the c.o.g. and size of the beam can be also calculated. Finally, one can plot 2 graphs, the first showing the evolution of the number of particles outside a given emittance (scaled to the rms emittance calculated above), the second showing the evolution of the number of particles outside a given size (scaled to the rms size calculated above). The last button (σ beam) shows the beam 6x6 matrix.
The emittances are calculated according to "***Energy and Phase limit***" defined in "***Multipartcle***" page.



Envelope plot example



Multiparticle phase space plot

Using TraceWin in batch command

There are 3 ways to run TraceWIN in batch command corresponding to 3 types of executable file:

X11 or GUI full version

On an X console, hiding the full TraceWin GUI / X11 with "*hide*" command.
In this case, multi-threading and full feature of code are available, such as, for example, statistical error studies.
<u>Examples of syntax:</u>

- For linux or mac:
  ./TraceWin /full_path/project_name.ini hide current1=80 freq1=352
- For windows use bat file and send standart output to a file :
  TraceWin d:\full_path\project_name.ini hide freq1=352 path_cal=c:\temp > output.txt

NoX11 or noGUI full version

On a command console:

- Linux executables: "*./TraceWin64_noX11 ../projet_path/project.ini"*
- Windows executables: "*./TraceWin64_noGUI ../projet_path/project.ini"*
- MacOS version could be proposed, if some users need them.

This executable is fully similar to GUI version except no X11 / GUI is required. Multithreading and full error study on locate and remote computers are also performed. The project configuration and file localizations must be exactly the same than for the GUI version. In other terms, use GUI version to configure your project or visualize results during an error study for example, but use the noX11 version to launch the code if you need to quit GUI during a process.

noX11 or noGUI minimal version

These versions are internally used by the main code for remote calculations performed during error studies. They are not available in the web site but can be directly extracted using the top menu "*Exe*".
On a command console:

- Windows executables: "*tracew32.exe" & "tracew64.exe"*
- Linux executables: "*tracelx" & "tracelx64"*
- MacOS executables: "*tracemac" & "tracemac64"*

In this case, multi-threading and statistical error studies are not possible. All project files must be located in the same directory with the executable file and thus no path has to be defined in the command line. Here, "*hide*" argument means no standard output and "*hide_esc*" argument means standard output without esc characters.

<u>Examples of syntax:</u>

- For linux or mac:
  ./tracelx project_name.ini hide current1=80 freq1=352
- For windows use bat file and send standart output to a file
  Tracew32 project_name.ini freq1=352 current1=80

For all the cases, the project file **.ini* is automatically loaded. Input variables listed as argument are changed. The calculation is started and at the end TraceWin is closed. The name of project must be the first parameter and the input variables available are shown in the following tab. You don't need to respect case for input variables. The syntax is always "*variable=value*" without space, except for the "*hide*" variable. If user needs another one, please, contact developer.

The "*tracewin.key*" and eventually the "*toutatis.key*" must always be copied with the executable file.

In addition to these generic variables defined in the following table, the parameters of the element can be changed.

Examples of syntax changing element parameters:

> *tracew32 project_name.ini freq1=352 current1=80 Ele[15][2]=10*
> In this example, the second parameter of the 15th element will be set to 10. Respects the parameter units defined in the documentation for each element.
>
> *./tracelx64 project_name.ini freq1=352 current1=80 ele[15][2]=10 ele[25][3]=0.255*

Ele[n][v]	Change the vth parameter of for the nth element
hide	Hide the GUI, or cancel console output (no parameter)
tab_file	Save to file the data sheet at the end of calcul (only GUI version)
Synoptic_file	Save the geometric layout at (entance (=1), middle (=2), exit (=3) of elements. (See "Synoptic" tools for file name) .
nbr_thread	Set the max. number of core/thread used
path_cal	Calculation directory
dat_file	Full name of structure file
dst_file1	Full name Input dst of main beam (*)
dst_file2	Full name Input dst of second beam (*)
current1	Input beam current (mA) of main beam
current2	Input beam current (mA) of second beam
nbr_part1	Number of particle of main beam
nbr_part2	Number of particle of second beam
energy1	Input kinetic energy (MeV) of main beam
energy2	Input kinetic energy (MeV) of second beam
etnx1	Input XX' emittance (mm.mrad) of main beam
etnx2	Input XX' emittance (mm.mrad) of second beam
etny1	Input YY' emittance (mm.mrad) of main beam
etny2	Input YY' emittance (mm.mrad) of second beam
eln1	Input ZZ' emittance (mm.mrad) of main beam
eln2	Input ZZ' emittance (mm.mrad) of second beam
freq1	Input beam frequency (MHz) of main beam
freq2	Input beam frequency (MHz) of second beam
duty1	Duty cycle of main beam
duty2	Duty cycle of second beam
mass1	Input beam mass (eV) of main beam
mass2	Input beam mass (eV) of second beam
charge1	Input particle charge state of main beam
charge2	Input particle charge state of second beam

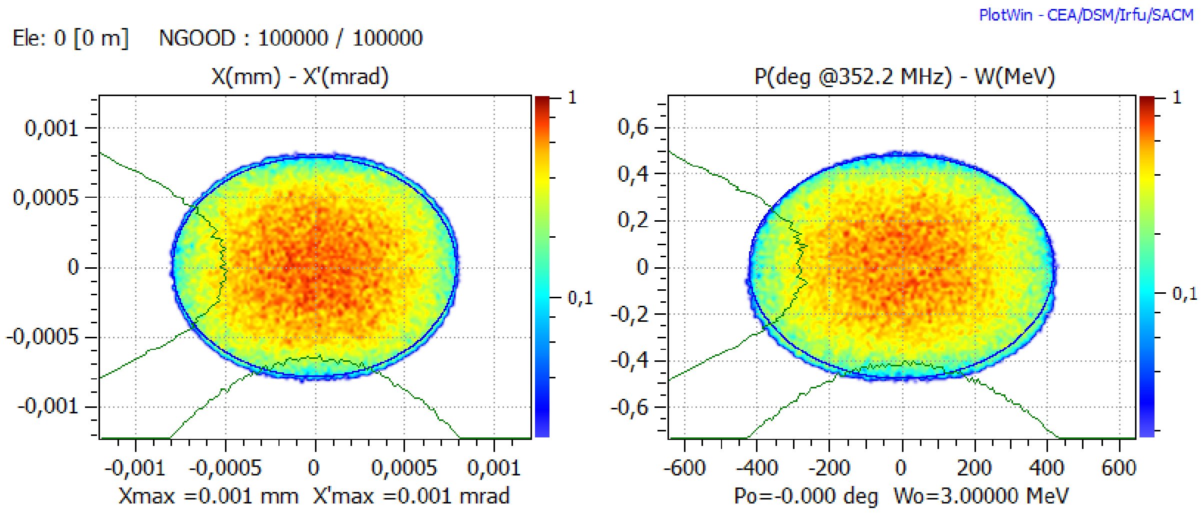
alpx1	Input twiss parameter alpXX' of main beam
alpx2	Input twiss parameter alpXX' of second beam
alpy1	Input twiss parameter alpYY' of main beam
alpy2	Input twiss parameter alpYY' of second beam
alpz1	Input twiss parameter alpZZ' of main beam
alpz2	Input twiss parameter alpZZ' of second beam
betx1	Input twiss parameter betXX' of main beam
betx2	Input twiss parameter betXX' of second beam
bety1	Input twiss parameter betYY' of main beam
bety2	Input twiss parameter betYY' of second beam
betz1	Input twiss parameter betZZ' of main beam
betz2	Input twiss parameter betZZ' of second beam
x1	Input X position of main beam
x2	Input X position of second beam
y1	Input Y position of main beam
y2	Input Y position of second beam
z1	Input Z position of main beam
z2	Input Z position of second beam
xp1	Input X angle of main beam
xp2	Input X angle of second beam
yp1	Input Y angle of main beam
yp2	Input Y angle of second beam
zp1	Input Z angle of main beam
zp2	Input Z angle of second beam
dw1	Input Dw of main beam
dw2	Input Dw of second beam
spreadw1	Input spread energy for CW beam of main beam
spreadw2	Input spread energy for CW beam of second beam
part_step	Partran calculation step per meter (per beta.lambda if < 0)
vfac	Change RFQ Ucav (ex : "vfac 0.5", half reduce of Ucav)
random_seed	Set the random seed
partran	Force or avoid tracking simulation (1 / 0)
toutatis	Force or avoid Toutatis simulation (1 / 0)

(*) : if *dst_filex* input is specified, the input beam parameters (emittances and twiss parameters) are automatically extracted from the specified file and used for computation and cannot be changed by corresponding input commands. Other parameters like beam centroid, current, number of particles are still modifiable.

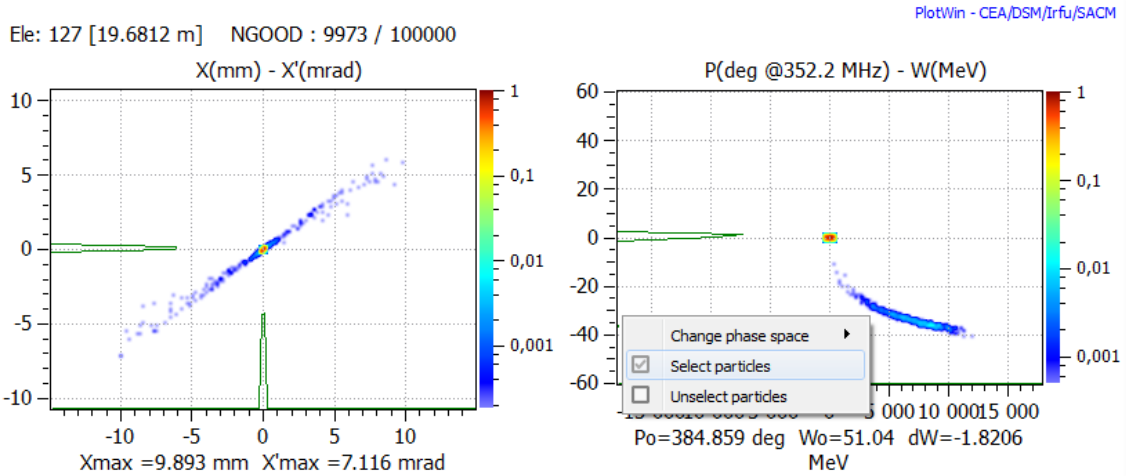
Acceptance calculation using TraceWin and PlotWin

Step allowing to performing the on-axis, zero current, longitudinal acceptance calculation.

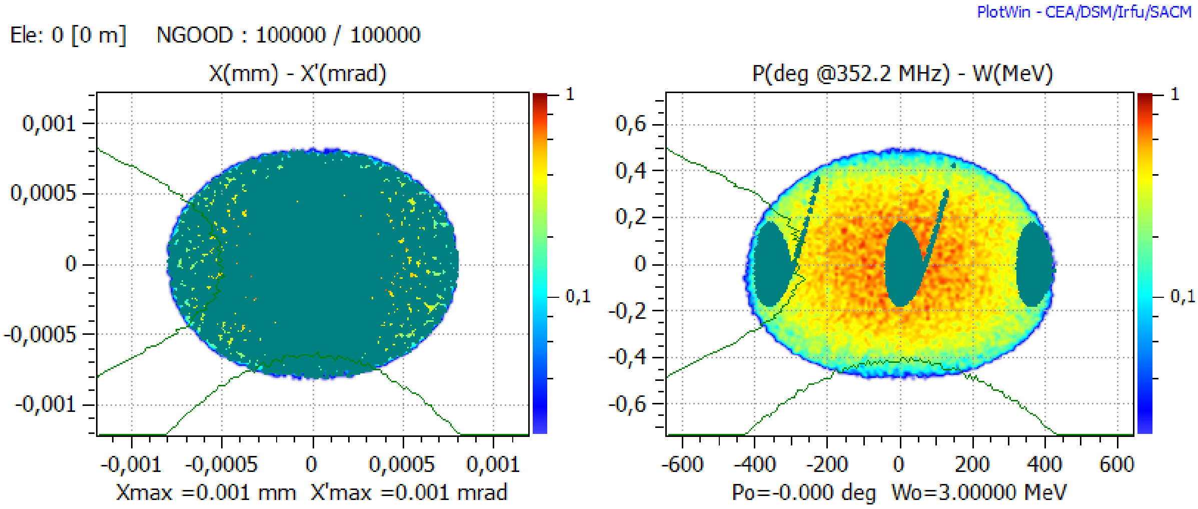
- Set transverse emittances to a very small value (not zero).
- Set the longitudinal emittance and Twiss parameters in order to get a phase and energy spreads bigger than the expected acceptance of your structure.
- Set current to zero.
- Select uniform distribution.
- Set the number of particle to 10.000 for example.
- Set in "Multiparticle" page → "Distribution option file, PLT" option → "Last element" in order to avoid to generating a huge plt file size.
- Remove plt compress level.
- Remove in "Multiparticle" page → "Phase en energy limits" all condition (set everything to zero).
- Make a run in multipartcle mode (no matching have to be done here).
- In "Chart" page, start "PlotWin" tool (if don't get it, upload it on the CEA site. This code must have been started once until TraceWin is able to recognize it). Starting PlotWin from TraceWin allows to PlotWin to directly use the good PLT file, but you can also open the good one directly from PlotWin.
- In PlotWin, plot phase-space distribution of the last element of your structure. Here are only the surviving particles. If the number of particle is to small compared to your input beam, you probably have to reduce the phase or/and the energy spread of your input beam.
- In the output distribution chart, select all particles (left click and "Select particles", like in the preceding picture).
- Now, in PlotWin, change the element number form the last one to 0, and plot again the phase-space distribution. The selected surviving particles have to appear in another color. In the chart option (button at the top of the chart, you can chose the size, color of the unselected and selected particles, you can also hide unselected particle, in order to have only the input surviving particles).
- You visualize your longitudinal acceptance.
- In "Save" menu, you have the possibility to save it (select "Particle checked") in a dst file.
- You can, now, restart the full step increasing the number of input particle in order to precisely define the acceptance (be careful, close PlotWin before to start another process).



Input beam distribution at element 0, small in transverse and big in longitudinal


PlotWin - CEA/DSM/Irfu/SACM
Ele: 127 [19.6812 m] NGOOD : 9973 / 100000
X(mm) - X'(mrad)
10
5
0
-5
-10
-10 -5 0 5 10
Xmax =9.893 mm X'max =7.116 mrad
1
0,1
0,01
0,001
P(deg @352.2 MHz) - W(MeV)
60
40
20
0
-20
-40
-60
5 000 10 000 15 000
Po=384.859 deg Wo=51.04 dW=-1.8206 MeV
☑ Select particles
Change phase space
□ Unselect particles
1
0,1
0,01
0,001

Surviving output distribution at last element (use menu to select all of them)


PlotWin - CEA/DSM/Irfu/SACM
Ele: 0 [0 m] NGOOD : 100000 / 100000
X(mm) - X'(mrad)
0,001
0,0005
0
-0,0005
-0,001
-0,001 -0,0005 0 0,0005 0,001
Xmax =0.001 mm X'max =0.001 mrad
1
0,1
P(deg @352.2 MHz) - W(MeV)
0,6
0,4
0,2
0
-0,2
-0,4
-0,6
-600 -400 -200 0 200 400 600
Po=-0.000 deg Wo=3.00000 MeV
1
0,1

Input distribution at element 0, where acceptance is visible, here the space spread could be reduced.

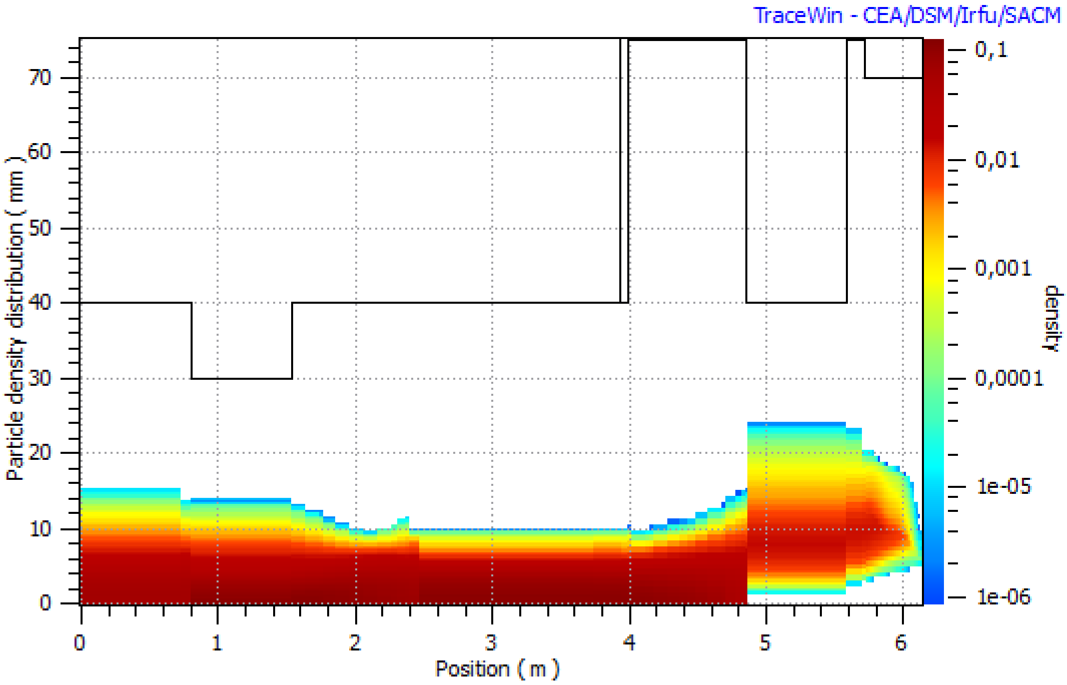
PlotWin code

PlotWin is a post-processing tool allowing to projecting and plotting a 6D beam distribution in 2D sub-phase-spaces and associated 1D beam density profiles. As many as 6 phase-spaces can be plotted on the same chart. The number of phase-spaces and the plot distribution can be chosen. The beam is represented by a set of particles with the same weight. This tool allows to observing each particle transport individually.

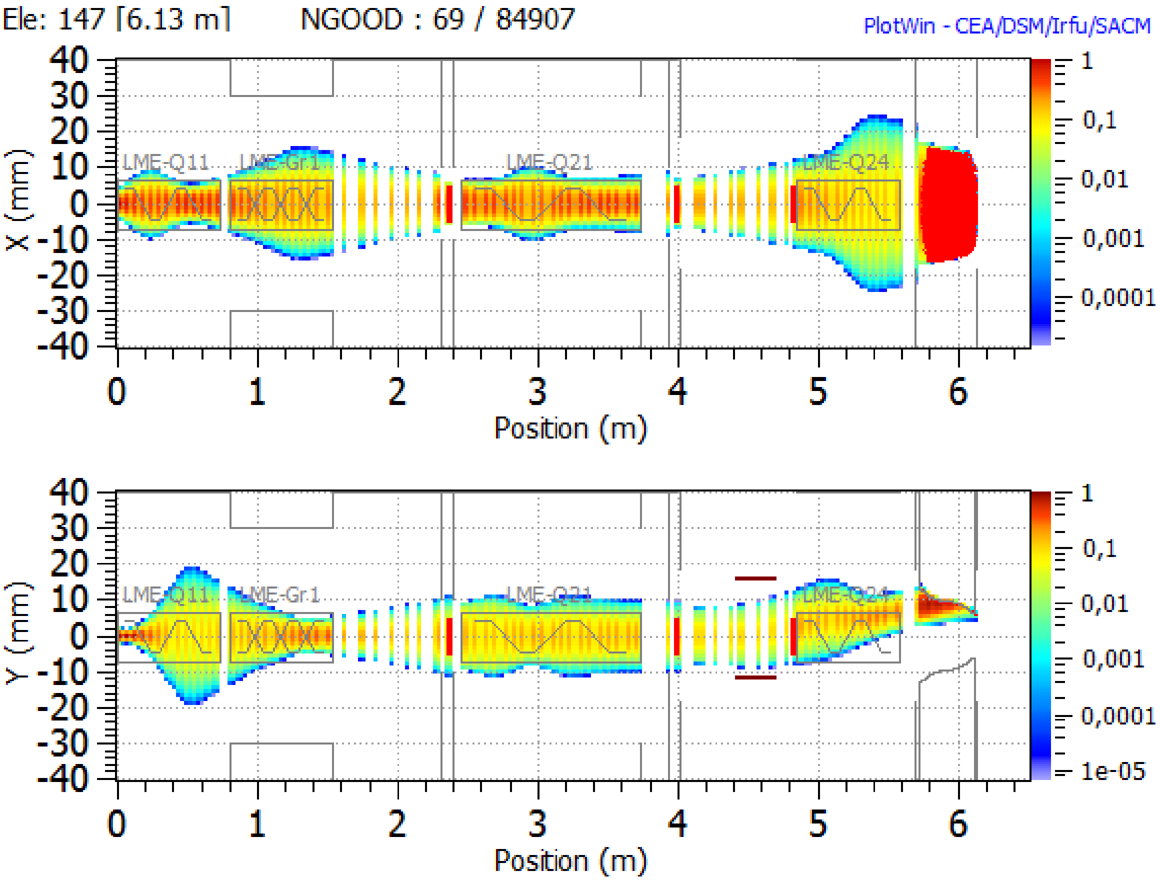
You can find the code at this location: http://irfu.cea.fr/Sacm/logiciels/

The beam density profile performed by TraceWin with "*Density*" button is just a roughly view of the beam density in *R* space. The aperture of elements is divided in 100 rings, where the number of particles is counted in order to generate the density plot.

PlotWin provides much better quality density plots, as illustrated in the following page.



Distribution plot from **TraceWin**



The same distribution plot from **PlotWin**

Back tracking feature

This option allows to transport a beam upside down through the machine from the end to the beginning. When the option “***Back Simulation***” is selected in the “*Main*” tab-sheet, the input beam has to be defined as the output beam of the machine. At the beginning of the run the structure of the machine is reversed (including the FIELD_MAP elements even superposed and other elements, accelerating or not). The space charge is also taken into account.
Two extra matrix elements are added at the begging and the end of structure in order to apply the following changes on the beam.

$$\begin{cases} x' = -x' \\ y' = -y' \\ z = L - z \end{cases}$$

L is the length of the elements.

For the FIELD_MAP elements specific case, two transformations are made:

$$\begin{cases} B_z = -B_z \\ E_z = -E_z \end{cases}$$

And the name is change from “*FIEL_MAP”* to “*MAP_FIELD”* to make a clear distinction

A new structure file (*.dat) is created corresponding to the reversed machine

Example of reversed transport

The following example includes quadrupole and cavity field map:

```
SUPERPOSE_MAP 0
LME-Q11 : FIELD_MAP 70 480 0 40 9.174 0 0 0 qpole480_lme_25_01_07b
SUPERPOSE_MAP 250
LME-Q12 : FIELD_MAP 70 480 0 30 -7.4796 0 0 0 qpole480_lme_25_01_07b
DRIFT 70 40 0
SUPERPOSE_MAP 0
SET_SYNC_PHASE
LME-Gr1 : FIELD_MAP 7700 300 -20 30 0.33 0.33 0 0 carte_3gap_2b
SUPERPOSE_MAP 250
LME-Q13 : FIELD_MAP 70 480 0 40 2.784 0 0 0 qpole480_lme_25_01_07b
DRIFT 100 40 0
END
```

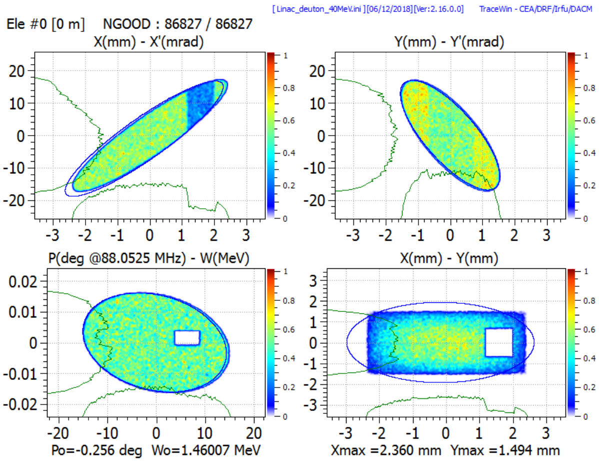
Once the run done, we can find in the “*Calculation*” directory the new structure file corresponding to the back simulation done (bellow). Two THIN_MATRIX elements are there to reverse beam and reverse back at the end, FIEL_MAP has been renamed to MAP_FIELD.

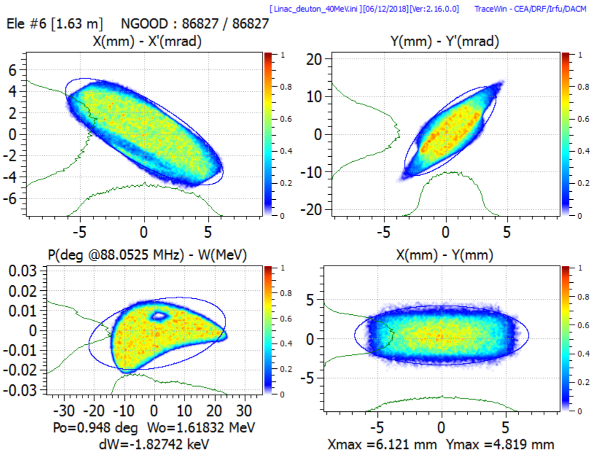
```
;-----------------------------------
```

```
; ELEMENTS OF BACK TRACKING FILE
; ELEMENTS OF BACK TRACKING FILE

; 2 THIN_MATIRIX has been added at the beginning
; and at the end of the structure to reverse the
; beam for back transport
;-----------------------------------

THIN_MATRIX 0 1 0 0 0 0 0 0 -1 0 0 0 0 0 0 1 0 0 0 0 0 0 -1 0 0 0 0 0 0 -1 0 0 0 0 0 0 1
DRIFT 100 40 0 0 0
SUPERPOSE_MAP 0 0 0 0 0 0
LME-Q13 : MAP_FIELD 70 480 0 40 2.784 0 0 0 qpole480_lme_25_01_07b
SET_SYNC_PHASE
SUPERPOSE_MAP 430 0 0 0 0 0
LME-Gr1 : MAP_FIELD 7700 300 -160 30 0.33 0.33 0 0 carte_3gap_2b
DRIFT 70 40 0 0 0
SUPERPOSE_MAP 0 0 0 0 0 0
LME-Q12 : MAP_FIELD 70 480 0 30 -7.4796 0 0 0 qpole480_lme_25_01_07b
SUPERPOSE_MAP 250 0 0 0 0 0
LME-Q11 : MAP_FIELD 70 480 0 40 9.174 0 0 0 qpole480_lme_25_01_07b
THIN_MATRIX 0 1 0 0 0 0 0 0 -1 0 0 0 0 0 0 1 0 0 0 0 0 0 -1 0 0 0 0 0 0 -1 0 0 0 0 0 0 1
END
```

Normal transport results

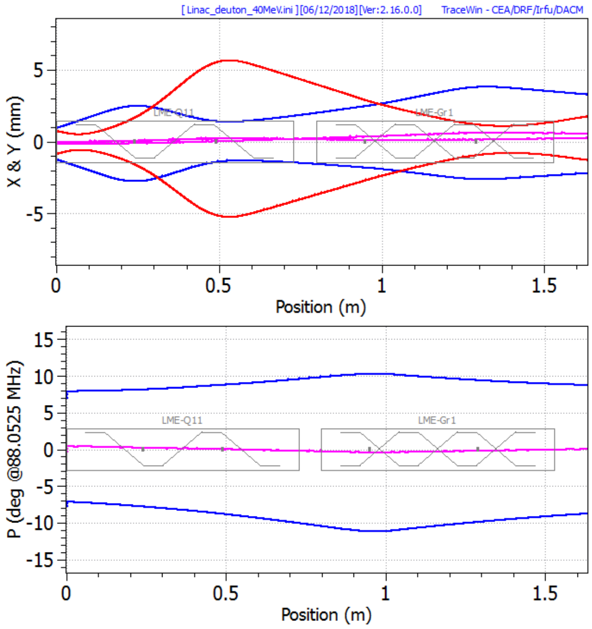
Back simulation option is unchecked; this is the normal and usual beam transport.



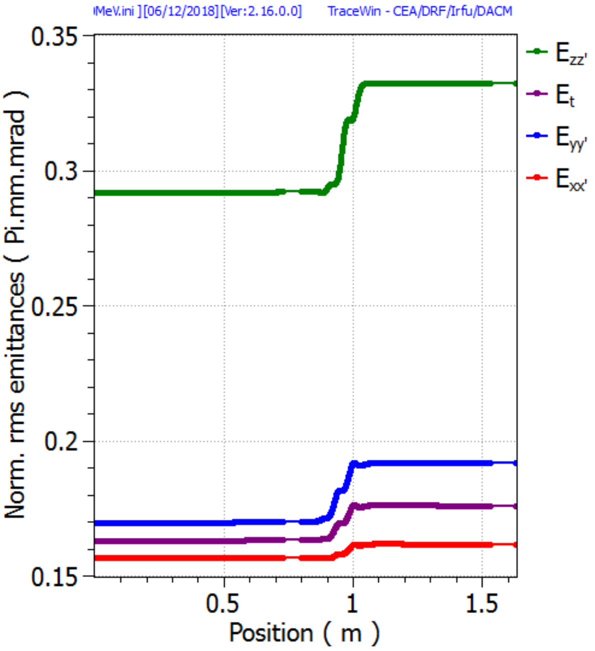


Input beam distribution. Holes are intentionally introduced into the transverse and longitudinal phase space.

Output beam distribution

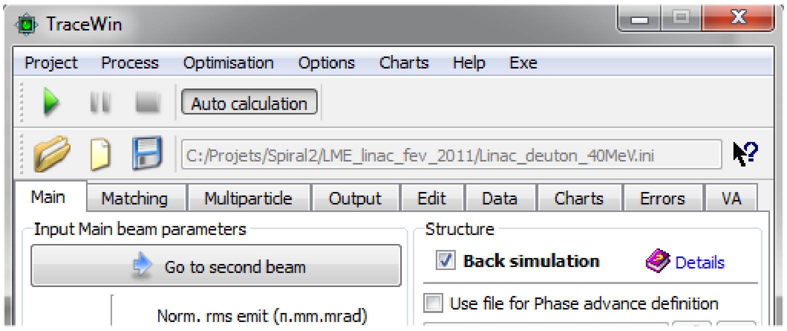


Beam envelops.



Emittances.

Back transport results



C:/Projets/Spiral2/LME_linac_fev_2011/Linac_deuton_40MeV.ini

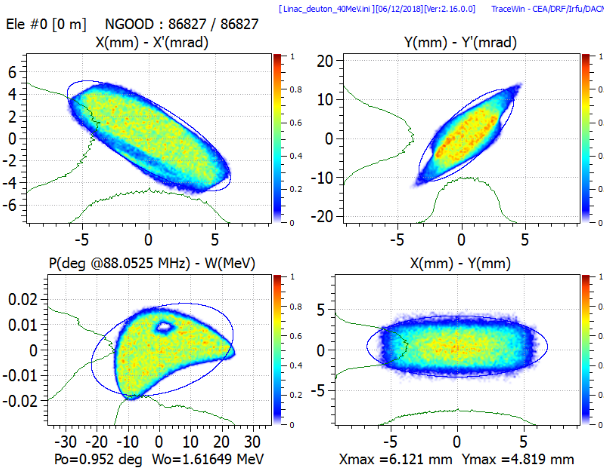
☑ Back simulation

□ Use file for Phase advance definition

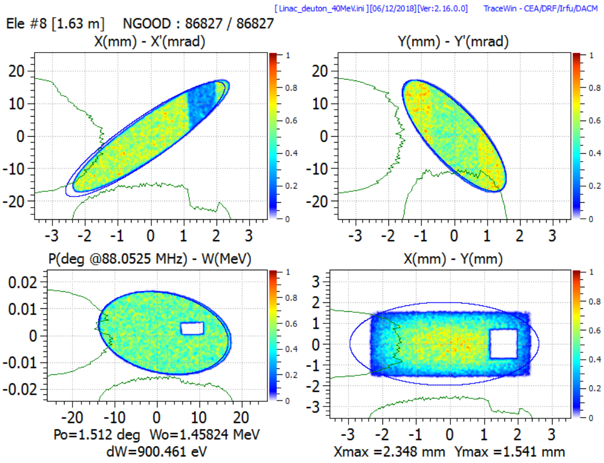
Back simulation option is checked and output beam distribution of preceding simulation is used as input beam

Another way consists to uncheck "*Back simulation*" option and to use in normal simulation mode, the reversed structure file.

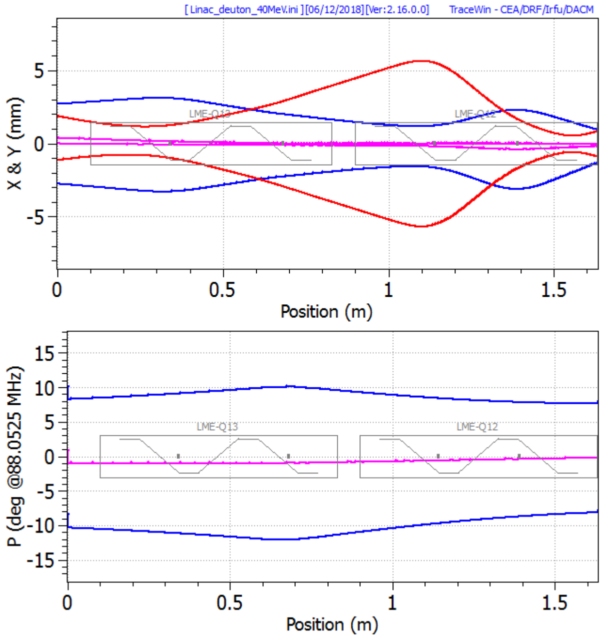
Both methods give the same results.



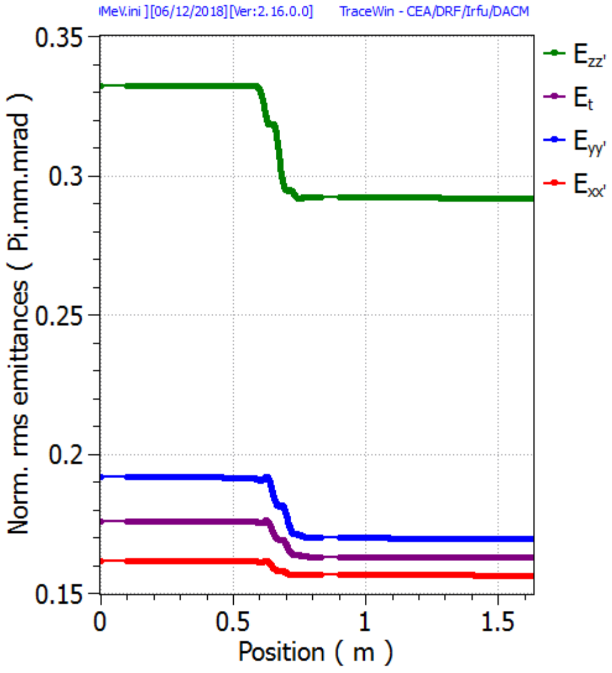
Input beam distribution



Output beam distribution



Beam envelops.



Emittances.

Checking reversed structure and code

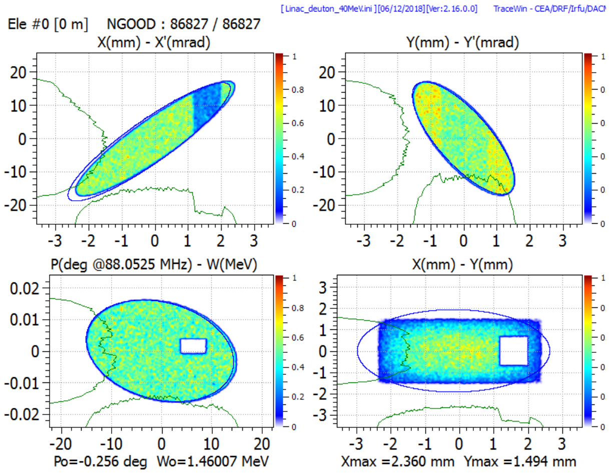
Because reverse a structure could be a complex work and because it's a nice way to verify this point and to check if the transport model (tracking or envelop) is consistent, it could be interesting to transport beam in the structure and in the reverse structure together. If every is good, you have to get output distribution identical to output one and also the transfer matrix of both structures has to be equal to the identity matrix. So considering the following structure file (Back simulation option unchecked)

```
SUPERPOSE_MAP 0
```

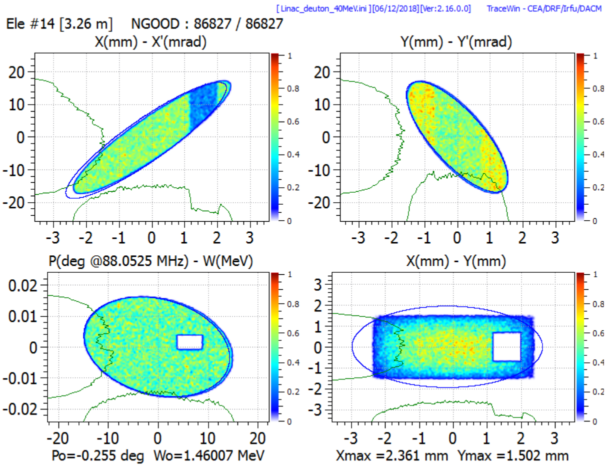
```
LME-Q11 : FIELD_MAP 70 480 0 40 9.174 0 0 0 qpole480_lme_25_01_07b
SUPERPOSE_MAP 250
LME-Q12 : FIELD_MAP 70 480 0 30 -7.4796 0 0 0 qpole480_lme_25_01_07b
DRIFT 70 40 0
SUPERPOSE_MAP 0
SET_SYNC_PHASE
LME-Gr1 : FIELD_MAP 7700 300 -20 30 0.33 0.33 0 0 carte_3gap_2b
SUPERPOSE_MAP 250
LME-Q13 : FIELD_MAP 70 480 0 40 2.784 0 0 0 qpole480_lme_25_01_07b
DRIFT 100 40 0
;-----------------------------------
; ELEMENTS OF BACK TRACKING FILE
; ELEMENTS OF BACK TRACKING FILE
;-----------------------------------
THIN_MATRIX 0 1 0 0 0 0 0 0 -1 0 0 0 0 0 0 1 0 0 0 0 0 0 -1 0 0 0 0 0 0 -1 0 0 0 0 0 0 1
DRIFT 100 40 0 0 0
SUPERPOSE_MAP 0 0 0 0 0 0
LME-Q13 : MAP_FIELD 70 480 0 40 2.784 0 0 0 qpole480_lme_25_01_07b
SET_SYNC_PHASE
SUPERPOSE_MAP 430 0 0 0 0 0
LME-Gr1 : MAP_FIELD 7700 300 -160 30 0.33 0.33 0 0 carte_3gap_2b
DRIFT 70 40 0 0 0
SUPERPOSE_MAP 0 0 0 0 0 0
LME-Q12 : MAP_FIELD 70 480 0 30 -7.4796 0 0 0 qpole480_lme_25_01_07b
SUPERPOSE_MAP 250 0 0 0 0 0
LME-Q11 : MAP_FIELD 70 480 0 40 9.174 0 0 0 qpole480_lme_25_01_07b
THIN_MATRIX 0 1 0 0 0 0 0 0 -1 0 0 0 0 0 0 1 0 0 0 0 0 0 -1 0 0 0 0 0 0 -1 0 0 0 0 0 0 1
END
```

Results

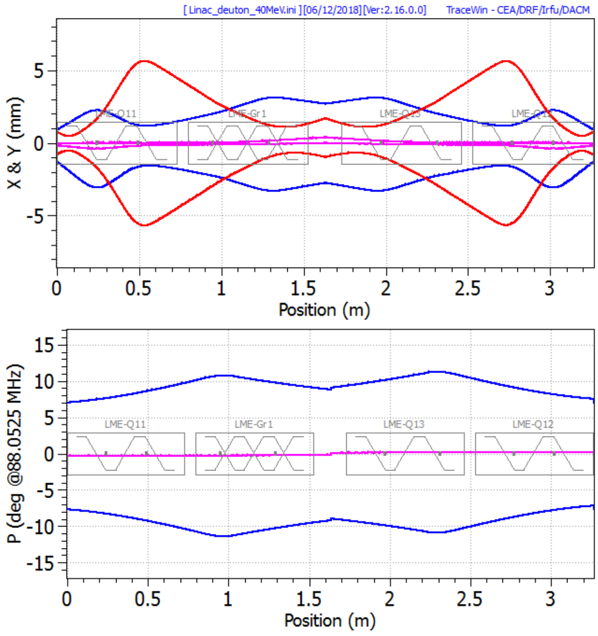
There is a perfect symmetry of the different beam parameters like output/input distributions, envelops and emittances.



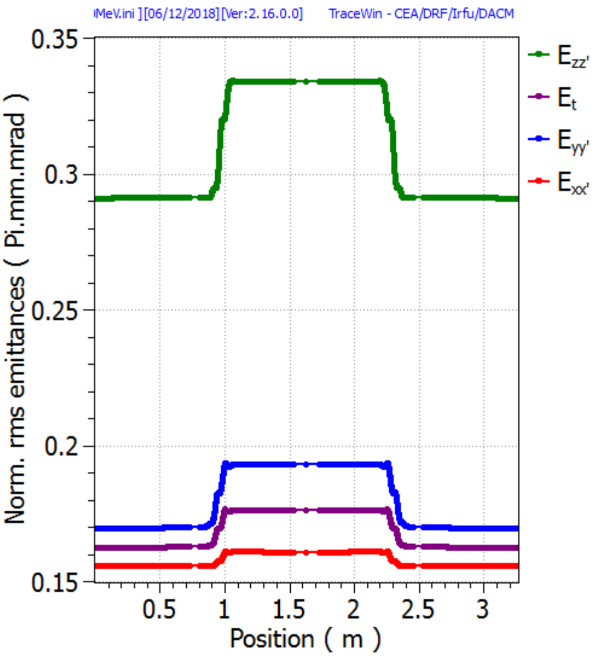
Input beam distribution



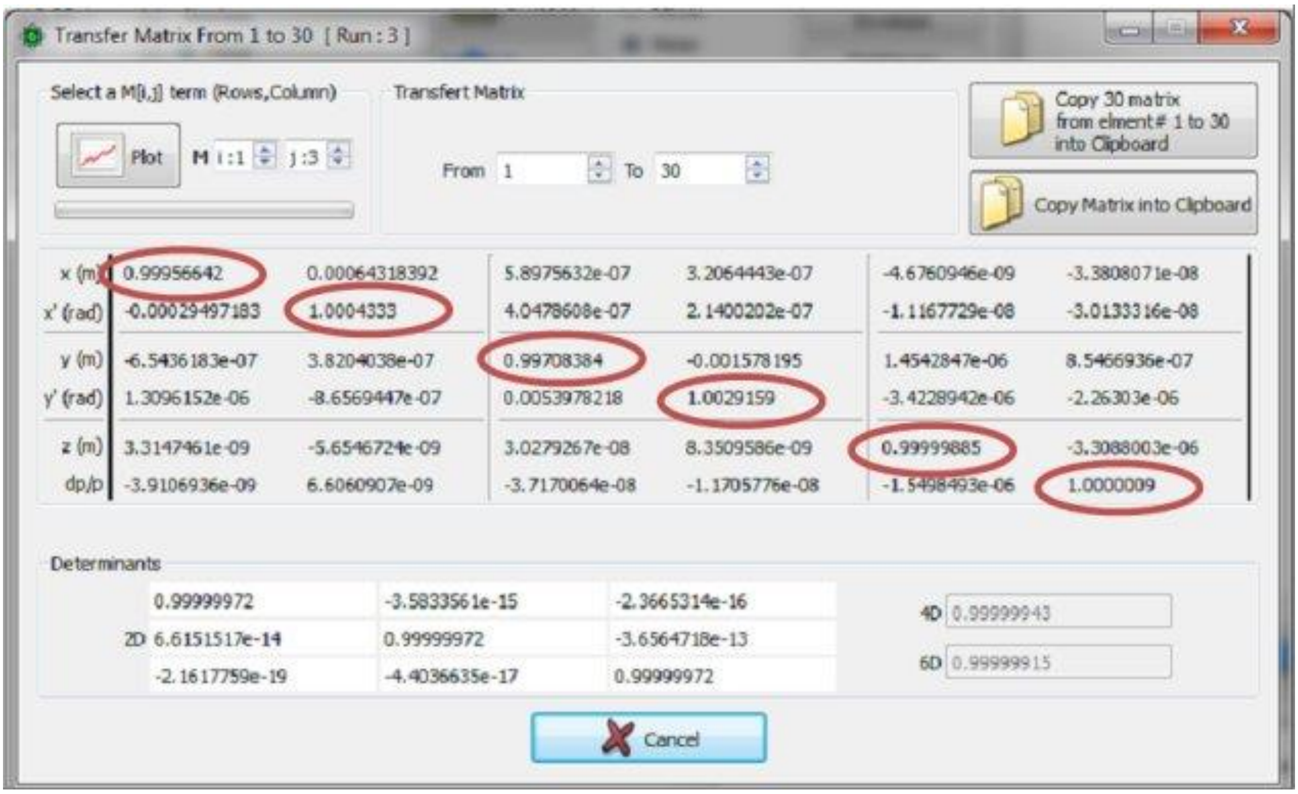
Output beam distribution



Beam envelops.



Emittances.


Plot
Copy 30 matrix from element# 1 to 30 into Clipboard
Copy Matrix into Clipboard
Cancel

Transfer Matrix From 1 to 30 [Run : 3]

Select a M[i,j] term (Rows,Column)

M i:1 j:3

Transfert Matrix

From 1 To 30

x (m)	0.99956642	0.00064318392	5.8975632e-07	3.2064443e-07	-4.6760946e-09	-3.3808071e-08
x' (rad)	-0.00029497183	1.0004333	4.0478608e-07	2.1400202e-07	-1.1167729e-08	-3.0133316e-08
y (m)	-6.5436183e-07	3.8204038e-07	0.99708384	-0.001578195	1.4542847e-06	8.5466936e-07
y' (rad)	1.3096152e-06	-8.6569447e-07	0.0053978218	1.0029159	-3.4228942e-06	-2.26303e-06
z (m)	3.3147461e-09	-5.6546724e-09	3.0279267e-08	8.3509586e-09	0.99999885	-3.3088003e-06
dp/p	-3.9106936e-09	6.6060907e-09	-3.7170064e-08	-1.1705776e-08	-1.5498493e-06	1.0000009

Determinants

	0.99999972	-3.5833561e-15	-2.3665314e-16
2D	6.6151517e-14	0.99999972	-3.6564718e-13
	-2.1617759e-19	-4.4036635e-17	0.99999972

4D 0.99999943

6D 0.99999915

Transfer matrix closed to be identical matrix, higher is the step of calculation, closer is this equality

Files

Data file **(*.dat)**
Init project file **(*.ini)**
Results file **(*.cal)**
Adjusted value file **(*.txt)**
Steerer strength file **(*.txt)**
Cavity setting point file **(*.txt)**
Sigma0 file **(*.sig0)**
Input file for multiparticle program **(*.par, *.dat)**
Density file **(*.dat)**
Particle distribution **(*.dat)**
Lost particles **(*.dat)**
Error file results **(*.txt)**
Error set of data **(*.txt)**
Input & Output particle distribution **(*.dst, *.plt)**
Partran or Toutatis output **(*.out)**
Electric or magnetic field map
Current or space charge compensation map **(*.scc)**
Aperture map **(*.ouv)**
Magnetic stripping file **(*.los)**
Gas stripping file **(*.los)**
Random seed **(*.log)**
Transfer matrix **(*.dat)**

Data file

The data file *("*.dat"*) contains the list of elements and commands. It must be ended by the command "END". The elements and the commands syntax are described in the rubric "Element definitions" and "Command definitions". Comments line begins by the character ';'.
A name for each element can be specified with 50 characters max, (See example below).
Result files are automatically created at the first use of the data file. At the end of a run, TraceWin creates another data file with the same name but located in the calculation directory, which contains the new element list. With, for example, the quadrupole values calculated to have the wanted phase advance law. If calculation directory is the same than the data file directory, the name of the new data file start with "*new_...*"
Warning:
Each command concerns the following element, par example "SET_TWISS" will impose some Twiss parameters at the output of the following element.
Two identical commands cannot be consecutive.

Example 1:

```
; **********************************************
DRIFT 1e-08 100
SPACE_CHARGE_COMP 0.7
DRIFT 350 100
DRIFT 60 100
DRIFT 192 100
MATCH_FAM_GRAD 1 1
ADJUST 1 2 1 0 0
SOLENOID 410 0.25 100
DRIFT 100 100
MATCH_FAM_GRAD 1 2
STEERER 0 0 100 0
ADJUST_STEERER 2
ADJUST 1 2 2 0 0
QUAD 200 0.18 100 0
DRIFT 150 100
END
;**********************************************
```

Example 2:

```
; **********************************************
DR1    : DRIFT 1e-08 100
SP1    :  SPACE_CHARGE_COMP 0.7
DR2    : DRIFT 350 100
DR3    : DRIFT 60 100
SOL 1 : SOLENOID 410 0.25 100
DR5    : DRIFT 100 100
QPF1   : QUAD 200 0.18 100 0
DR6    : DRIFT 150 100
;***********************************************
```

Init project file

The init file *"project_name.ini"* contains all the TraceWin project parameters. It can be loaded, saved, copied by using the TraceWin menu.

Results file

Created by TraceWin, his name is "*Data_file_name.cal*" and it is located in the data file directory and contains the results of the already done matching calculations, to avoid redundant calculations. See the following example.

```
Twiss_parameters_of_matched_beam
0.3167415265 0.1850852302 0.5246751875
-0.0938830920 0.0822867115 -0.0875778140

Matching_Between_Section_1_to_2
-8.11004 8.16711  -8.19803 8.21871
-3.8207 -2.8207 -4.1476 -1.1476
0.00834559 0.000887792
BEAM_FAM_69_0.DST
```

The three first lines are written after a matching beam calculation. The second line contain the Twiss parameters $\beta_{xx'}$, $\beta_{yy'}$, $\beta_{zz'}$, and the last $\alpha_{xx'}$, $\alpha_{yy'}$, $\alpha_{zz'}$.
The five following lines are written after a matching calculation, it contains the result of a matching between two sections. The first line contains the quadrupole gradients witch have been adjusted ("MATCH_FAM_GRAD" command), the second line is either the phase shift or the field factor correction or both, of the accelerator elements witch have been adjusted ("MATCH_FAM_PHASE", "MATCH_FAM_FIELD" or "MATCH_FAM_LFOC" command). The third line correspond to the element length witch have been adjusted ("MATCH_FAM_LENGTH" command) and the last is the name of a beam distribution file (located in the file data path), which is save when the matching family calculated is done with the option "*With beam from Partran*". All of these lines are optional and depend from the "MATCH_FAM…"command in your data file.
For more details, see the matching commands and its examples. You can also force the optimization process of calculation with starting values by the following commands

```
Init_Matching_Between_Section_1_to_2
-8.11004 8.16711 -8.19803 8.21871
-2.8207 -2.8207 -3.1476 -3.1476
```

To put in comment a result, simply add the char **";"** as a first character.
This file contains also all the diagnostic results like the following example. For more details, see the adjust commands and its examples. You can also force the optimization process of diagnostic calculation with starting values by using "***Init_***" syntax

```
Diagnostic_10
10.7992 -10.5893 5.81701

Init_Diagnostic_10
10.7992 -10.5893 5.81701
```

For all these result, an extension "*_PAR*" is added when the result comes from a multiparticle optimization '***With Partran***" is checked

Sigma0 file

Created by TraceWin or No, his name is "**.sig*". It's located in the data file directory and contains the transverse phase advances law with no current, one value per lattice. See following example, where red values correspond to optional vertical phase advance, by default sigy=sigx.

60 40
60 41
61 41
62 42
…
..
.

Magnetic or electric Field map

Input field for “FIELD_MAP” element, see also FIELD_MAP details.
In “Chart” page a tool allows to visualize (1D or 2D) the field maps from elements defined in data file. This tool also allows to convert the field ASCII format to Binary format. That allows code to be faster when the field maps size are too big.

The field map file syntax is the following in the ASCII format:

Fz are in MV/m for electric field or in T for magnetic field.
For specific 3D aperture field map file, Fz = 0 or 1, 1 corresponding to material.

The dimensions are in meter.

- Dimension 1 :

Be careful, to the number of data requested, *N=(nz+1)*

```
nz zmax
Norm
for k=0 to nz
    Fz(k.zmax/nz)
    Return
```

- Dimension 2 :

Be careful, to the number of data requested, *N=(nz+1)*(nr+1)*

```
nz zmax
nr rmax
Norm
for k=0 to nz
    for i=0 to nr
        $Fz(k \cdot z_{max}/nz, i \cdot r_{max}/nr)$
        Return
```

or

Be careful, to the number of data requested, *N=(nx+1)*ny+1)*

```
nx xmin xmax
ny ymin ymax
Norm
for k=0 to ny
    for i=0 to nx
        $Fz(k \cdot x_{max}/nx, i \cdot y_{max}/ny)$
        Return
```

- Dimension 3 :

```
nz zmax
nx xmin xmax
ny ymin  ymax
Norm
```

```
for k=0 to nz
    for j=0 to ny
        for i=0 to nx
            Fz($k \cdot z_{max}/nz$, $y_{min}+j \cdot (y_{max}-y_{min})/ny$, $x_{min}+i \cdot (x_{max}-x_{min})/nx$)
            Return
```

The field map file syntax is the following in the BINARY format:

- Dimension 1 :

```
nz (integer 4 bytes) zmax (double 8 bytes)
Norm (double 8 bytes)
for k=0 to nz
    Fz(k.zmax/nz) (float 4 bytes)
```

- Dimension 2 :

```
nz  (integer 4 bytes) zmax (double 8 bytes)
nr (integer 4 bytes) rmax (double 8 bytes)
Norm (double 8 bytes)
for k=0 to nz
    for i=0 to nr
        Fz($k \cdot z_{max}/nz$, $i \cdot r_{max}/nr$) (float 4 bytes)
```

- Dimension 3 :

(Be careful to the dimention order)

```
nz (integer 4 bytes) zmax (double 8 bytes)
nx (integer 4 bytes )xmin (double 8 bytes) xmax (double 8 bytes)
ny (integer 4 bytes) ymin (double 8 bytes) ymax (double 8 bytes)
Norm (double 8 bytes)
for k=0 to nz
    for j=0 to ny
        for i=0 to nx
            Fz($k \cdot z_{max}/nz$, $y_{min}+j \cdot (y_{max}-y_{min})/ny$, $x_{min}+i \cdot (x_{max}-x_{min})/nx$) (float 4 bytes)
```

Warning: The lattice has to be regular.
The normalization factor is equal to k_e/Norm or k_b/Norm.

Fz are in MV/m for electric field or in T for magnetic field.
For specific 3D aperture field map file, Fz = 0 or 1, 1 corresponding to material.

The dimensions are in meter.

Current or space charge compensation map

"FileMapName.scc"
A flag in "FIELD_MAP" element syntax allow to include it.
The space charge compensation or current file syntax is like following:

- Space charge compensation according to *Z* format:

```
0 N
for i=0 to N-1
    $Z_i$ $Scc_i$
```

- Current evolution according to *Z* file format:

```
1 N
for i=0 to N-1
```

```
            Zi Ii
```

- *Zi* is the position (m)
- *Scci* is the space charge compensation at the *Zi* position, (1 for 100%)
- *Ii* is the current (mA) at the *Zi* position

Partran and TraceWin codes make an interpolation in between this figure.

Aperture map

“FileMapName.ouv”
A flag in FIELD_MAP element syntax allow to include it.
For the field map elements, sometime we need to define a beam pipe radius geometry according to z axis. The file syntax is the following:
Warning in case of superposed field map these aperture map have to be defined in the first FIELD_MAP element and have to get a length equivalent to all field_map.

- Aperture according to *Z* format:

```
N
for i=0 to N-1
            Zi Ouvi
```

- *Zi* is the position (m)
- Ouv*i* is the aperture radius(m) at *Zi*.

The first location Zi has to be 0.

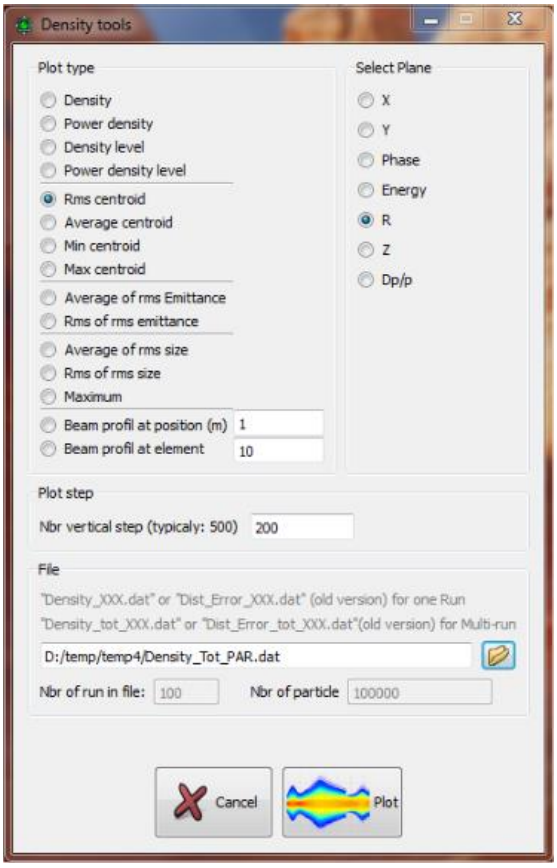
Input files for multiparticle programs

At the end of a calculation TraceWin creates the input files for multiparticle library, PARTRAN (*Data_file_name.par*), and TOUTATIS (*toutatis.dat*).

New Particle density distribution

Since TraceWin version ***2.4.2.0***, the files “***Desnity_PAR.dat***” (for multiparticle) and “***Density_Env.dat***” (for envelope) replace respectively “*Dist_Error_PAR.dat*” and “*Dist_Error_Env.dat*” files. These new files contain much more data in order to visualize the transverse and longitudinal beam densities according to Z and some other beam parameters (all these data can come from the sum of statistic studies). All available charts are accessible via the Toolbox "Density". This tool is still able to read old density file version.
The following C++ example shows how density files are written.

These files replace also the obsolete file (particle loss distribution, “*Dist_Error_Tot_PAR.loss*”)



Density tools

Plot type

○ Density
○ Power density
○ Density level
○ Power density level
● Rms centroid
○ Average centroid
○ Min centroid
○ Max centroid
○ Average of rms Emittance
○ Rms of rms emittance
○ Average of rms size
○ Rms of rms size
○ Maximum
○ Beam profil at position (m) 1
○ Beam profil at element 10

Select Plane

○ X
○ Y
○ Phase
○ Energy
● R
○ Z
○ Dp/p

Plot step

Nbr vertical step (typicaly: 500) 200

File

"Density_XXX.dat" or "Dist_Error_XXX.dat" (old version) for one Run
"Density_tot_XXX.dat" or "Dist_Error_tot_XXX.dat"(old version) for Multi-run

D:/temp/temp4/Density_Tot_PAR.dat

Nbr of run in file: 100 Nbr of particle 100000

In case of statistical error study 2 new files are created name **Density_tot_ENV.dat** and **Density_tot_PAR.dat** witch contain the sum of all the simulation.
If the "*Nbr of Step*" parameter of the tab-sheet "*Error*" is bigger than 1 the name of the 2 files become for example for 5 steps

"Density_Tot_Env_0.2000.dat" for 20%
"Density_Tot_Env_0.4000.dat" for 40%,
…
"Density_Tot_Env_1.0000.dat" for 100%,

Example to read the Density_XXX file.

```
#define den_year 2011
#define den_version 11
#include <stdio.h>

void Density_file_reading()
{
  short int ver,year,vlong;
  int nelp,Nrun,n=7,step;
  float moy[7],moy2[7],maxb[7],minb[7], maxR[7],minR[7],rms_size[7],rms_size2[7];
  float rms_emit[3],rms_emit2[3],min_pos_moy[7],max_pos_moy[7];
  float Zg,Xouv,Youv,dXouv,dYouv,ib,Eouv,PhPouv,PhMouv,Mipowlost,Mapowlost;
  long long int Np=0,lost2,Milost,Malost,longfichier=0;
  double powlost2;

  // All the following tables have to be correclty initialized
  long long int *lost=NULL;
  unsigned long long int **tab=NULL;
  unsigned int **stab=NULL;
  float **tabp=NULL,*powlost=NULL;
  FILE *f=fopen("Density_PAR.dat","rb");

  if (f==NULL) {
    printf("Erreur: Impossible to open Density file\n");
    exit(1);
  }

  if (fseek(f,0,SEEK_END)==0) {
    longfichier=ftell(f);
    fseek(f,0,SEEK_SET);
  }

  do {
    /* The following sequence is repeated for each position (Zg) in the machine */
    fread(&ver,sizeof(short int),1,f);           /* Density file version : 3 to 10*/
    fread(&year,sizeof(short int),1,f);          /* year of development : 2011 */
    fread(&vlong,sizeof(short int),1,f);        /* vlong=1 if the number of particle is greater
than 2e9 */
    fread(&Nrun,sizeof(int),1,f);               /* Number of run (1 for envelop or multiparicle
simulation) more for statistical error studies */
    fread(&nelp,sizeof(int),1,f);                /* element # */
    fread(&ib,sizeof(float),1,f);                /* Beam courant (A) */
    fread(&Zg,sizeof(float),1,f);           /* Position (m), end of element or at step position
calculation in field_map or in envelope mode */
    fread(&Xouv,sizeof(float),1,f);              /* Horizontal aperture */
    fread(&Youv,sizeof(float),1,f);              /* Vertical aperture */
    if (ver>=9) {
      fread(&dXouv,sizeof(float),1,f);           /* Horizontal aperture shift */
      fread(&dYouv,sizeof(float),1,f);           /* Vertical aperture shift*/
    }
    fread(&step,sizeof(int),1,f);               /* The beam is slice en step from max. to min.
beam size */
    /* n=7 (0:X(m)) (1:Y(m)) (2:Phase(°)) (3:Energy(MeV)) (4:R(m)) (5:Z(m)) (6:dp/p) */
    fread(moy,sizeof(float),n,f);                /* Beam average for each plane */
    fread(moy2,sizeof(float),n,f);              /* Squared beam average for each plane (needed
when Nrun>1) */
    fread(maxb,sizeof(float),n,f);                  /* Maximum beam size or particle excursion
for each plane */
    fread(minb,sizeof(float),n,f);                  /* Minimum beam size or particle excursion
for each plane */
    if (ver>=11) {
      fread(&phaseF,sizeof(float),1,f);                   /* Absolute beam phase */
      fread(&phaseG,sizeof(float),1,f);            /* Absolute referenve beam phase */
    }
    if (ver>=10) {
      fread(maxR,sizeof(float),n,f);                /* Minimum of maximum beam size or particle
excursion for each plane */
      fread(minR,sizeof(float),n,f);                /* Maximum of minimum beam size or particle
excursion for each plane */
    }
    if (ver>=5) {
      fread(rms_size,sizeof(float),n,f);         /* rms beam size */
      fread(rms_size2,sizeof(float),n,f);        /* Squared beam rms size */
    }
    if (ver>=6) {
      fread(min_pos_moy,sizeof(float),n,f);      /* Min. if the beam average */
      fread(max_pos_moy,sizeof(float),n,f);      /* Maximum if the beam average */
    }
    if (ver>=7) {
      fread(rms_emit,sizeof(float),3,f);         /* rms emittances, xx’, yy’, zdp (m.rad)*/
      fread(rms_emit2,sizeof(float),3,f);           /* Squared rms emittances, xx’, yy’, zdp
(m.rad)2 */
    }
    if (ver>=8) {
      fread(&Eouv,sizeof(float),1,f);            /* Energy Acceptance (eV) */
      fread(&PhPouv,sizeof(float),1,f);          /* Positive Phase acceptance (deg)*/
      fread(&PhMouv,sizeof(float),1,f);           /* Negative Phase acceptance (deg)*/
    }
    fread(&Np,sizeof(long long int),1,f);

    if (Np>0) { //several linac simulation
      /* particle lost and beam power lost for each linac */
      if (lost!=NULL && powlost!=NULL) {
        for (int i=0;i<Nrun;i++) {
          fread(&lost[i],sizeof(long long int),1,f);  /* Number of particle lost at position Zg
*/
          fread(&powlost[i],sizeof(float),1,f);       /* Beam power lost(W) at position Zg  */
        }
      }
      else fseek(f,Nrun*(sizeof(long long int)+sizeof(float)),SEEK_CUR);

      fread(&lost2,sizeof(long long int),1,f);      /* Squared particle number lost at position
Zg  */
      fread(&Milost,sizeof(long long int),1,f);     /* Minimum particle lost at position Zg when
Nrun>1 */
      fread(&Malost,sizeof(long long int),1,f);     /* Maximum particle lost at position Zg when
Nrun>1 */
      fread(&powlost2,sizeof(double),1,f);           /* Squared beam power lost(W) at position
Zg  */
      fread(&Mipowlost,sizeof(float),1,f);           /* Minimum beam power lost at position Zg
when Nrun>1 */
      fread(&Mapowlost,sizeof(float),1,f);           /* Maximum beam power lost at position Zg
when Nrun>1 */

      /*tab or stab contains beam distribution from max. to min. size for each plane (7) slice
in step*/
      if (tab!=NULL && stab!=NULL) {
        for (int j=0;j<n;j++) {
          if (vlong==1) fread(tab[j],sizeof(unsigned long long int),step,f);
          else fread(stab[j],sizeof(unsigned int),step,f);
        }
      }
      else {
        if (vlong==1) fseek(f,n*step*sizeof(unsigned long long int),SEEK_CUR);
        else fseek(f,n*step*sizeof(unsigned int),SEEK_CUR);
      }
      if (ib>0) {
        /* tabp contains beam power distribution from max. to min. size for X, Y and R planes
*/
        if (tab!=NULL && stab!=NULL && tabp!=NULL) {
          for (int j=0;j<3;j++) {
            fread(tabp[j],sizeof(float),step,f);
          }
        }
        else fseek(f,3*step*sizeof(float),SEEK_CUR);
      }
    }
    /* next step  */
    /* break when Zg>=Linac length */
    if (ftell(f)+16>=longfichier) break;
  } while (!feof(f));
  fclose(f);
}
```

Particle density distribution

Dist_Error_Env.dat **(OBSOLETE see *Density_Env.dat* file)**

Contain the beam distribution at the end of each element after an envelope calculation. This file is created if “***nbr of particles”*** is greater than 10 and “***Use aperture element”*** of ***“Main”*** page is selected. During an error study the condition “***nbr of particles”*** is sufficient. You can visualize these results in the “***Error***” page by setting the “***Distribution file***” and using the right buttons

Dist_Error_PAR.dat **(OBSOLETE see *Density_PAR.dat* file)**

Contain the beam distribution at the end of each element after a multiparticle calculation. This file is created either by PARTRAN or TOUTATIS. You can visualize these results in the “***Error***” page by setting the “***Distribution file***” and using the right buttons

- N: Number of linac = 1

- Element number
- Element aperture (cm)
- Element aperture (cm)
- $\sum_{1}^{N}\sqrt{x^2+y^2}$ (cm)
- $\sum_{1}^{N}x^2+y^2$ (cm^2)
- $\sum_{1}^{N}x$ (cm)
- $\sum_{1}^{N}x^2$ (cm^2)
- $\sum_{1}^{N}y$ (cm)
- $\sum_{1}^{N}y^2$ (cm^2)

- 100 integers corresponding to the particle distribution along the aperture divided in 100 steps.
- 100 Doubles corresponding to the power distribution along the aperture divided in 100 steps.
- $\sum_{1}^{N}particle\_lost$
- $\sum_{1}^{N}particle\_lost^2$

- Max particle lost
- Min particle lost
- $\sum_{1}^{N}power\_lost$ (w)
- $\sum_{1}^{N}power\_lost^2$ (w)

- Max power lost (w)
- Min power lost (w)

In case of statistical error study 2 new files are created name **Dist_Error_tot_ENV.dat** and **Dist_Error_tot_PAR.dat** witch contain the sum of the 2 preceding files **(N>1).**
If the “*Nbr of Step*” parameter of the tab-sheet “*Error*” is bigger than 1 the name of the 2 files become for example for 5 steps

“Dist_Error_Tot_Env_0.2000.dat“ for 20%
“Dist_Error_Tot_Env_0.4000.dat“ for 40%,
…
“Dist_Error_Tot_Env_1.0000.dat“ for 100%,

Steerer strength file

The file "*Steerer_Values.txt*" is created after diagnostic position calculation. Its syntax is number of diagnostic follows by all the steerer strengths associated in T.m (Plane X and Y).

In case of statistical error study a new file named "*Steerer_Values_Tot_X_XX.txt"* is written including all the steerer strength of the whole linac simulated. A tool available in the "Errors" page-sheet allows to extract all useful statistical results from this file.

Cavity setting point file

The file "Cav_set_point_res.*txt*" is created after envelope calculation. It contains the synchronous phase and Voltage for each RF cavity of the structure. Reference values and final tuned values are included.

In case of statistical error study a new file named "Cav_set_point_res_*Tot_X_XX.txt"* is written including all the cavity setting points of the whole linac simulated. A tool available in the "Errors" page-sheet allows to extract all useful statistical results from this file.

Adjusted values file

The file "*Adjusted_Values.txt*" is created after each diagnostic optimization adjusting some elements parameters.

In case of statistical error study a new file named "*Adjusted_value_Tot_X_XX.txt"* is written including all the element parameters adjusted of the whole linac simulated.

Magnetic stripping file

The file "*MAGSTRIP1.LOS*" is created only in multiparticle mode if option "*Magnetic stripping*" is selected in "*Option*" of multiparticle codes. You can directly exploit these results using "*Stripping*" button in tab-sheet "*Graphs*". It contains the probability losses due to Lorentz magnetic stripping. The syntax is the following. (One line per element):

Number of element, Position (m), Energy (MeV), Losses probability

In case of statistical error study a new file named "*MAGSTRIP1_TOT.LOS"* is written including the probability sum of the whole linacs simulated. You can directly exploit these results using "***Stripping losses probability results"*** button in "***Errors***" page. The syntax is the following (One line per element):

Number of element, Number of linac simulated, Sum of losses probability

The probability corresponding to Sum of Magnetic stripping probability divided by Number of linac simulated.

Gas stripping file

The file “*GASSTRIP1.LOS*” is created only in multiparticle mode if option “Gas *stripping*” is selected in “*Option*” of multiparticle codes and if command Gas pressure is included in the data file. It contains the probability losses due to Gas stripping. The syntax is the following. (One line per element):

Number of element, Position (m), Energy (MeV), Losses probability

In case of statistical error study a new file named “GAS*STRIP1_TOT.LOS”* is written including the probability sum of the whole linacs simulated. You can directly exploit these results using “***Stripping losses probability results”*** button in “***Errors***” page. The syntax is the following (One line per element):

Number of element, Number of linac simulated, Sum of losses probability

The probability corresponding to Sum of Gas stripping probability divided by Number of linac simulated.

Lost particles file

Losses_PAR.dat
Losses_PAR_TOT.dat (in statistical error study)

These ASCII files are created only while multiparticle simulation, if the option “*Losses file*“ is activated in the “*Multparticle*” page. These files contain all coordonates of each lost particles. The syntax of the file is described at the first line.

Input & Output particle distribution

These following files are created while multiparticle simulation
part_dtl1.dst: Binary file containing the output beam distribution at the end of the linac.
part_rfq1.dst: Binary file containing the beam distribution at the entrance of the linac.

A .dst file use a binary format. It contains information of a beam at a given longitudinal position: number of particles, beam current, repetition frequency and rest mass as well as the 6D particles coordinates. The format is the following:

2xCHAR+INT(Np)+DOUBLE(Ib(mA))+DOUBLE(freq(MHz))+CHAR+
Np×[6×DOUBLE(x(cm),x'(rad),y(cm),y'(rad),phi(rad),Energie(MeV))]+
DOUBLE(mc2(MeV))

Comments:

- CHAR is 1 byte long ,
- INT is 4 bytes long,
- DOUBLE is 8 bytes long.
- Np is the number of particles,
- Ib is the beam current,
- freq is the bunch frequency,
- mc2 is the particle rest mass.

dtl1.plt: Binary file containing the beam distribution at the end of each element.

A .plt file use a binary format. It contains information of a beam at many longitudinal positions: longitudinal position, number of particles, beam current, repetition frequency and rest mass as well as the 6D particles coordinates. The format is the following:

2xCHAR+INT(Ne)+INT(Np)+DOUBLE(Ib(A))+DOUBLE(freq(MHz))+
DOUBLE(mc2(MeV))+
Ne×[CHAR+INT(Nelp)+DOUBLE(Zgen)+DOUBLE(phase0(deg))+DOUBLE(wgen(MeV))+
Np×[7×FLOAT(x(cm),x'(rad),y(cm),y'(rad),phi(rad),Energie(MeV),Loss)]]

Comments:

- CHAR is 1 byte long,
- INT is 4 bytes long,
- FLOAT is a Real 4 bytes long.
- DOUBLE is a Real 8 bytes long.
- Ne is the number of different positions,
- Np is the number of particles,
- Ib is the beam current,
- freq is the bunch frequency,
- mc2 is the particle rest mass,
- Nelp is the longitudinal element position,
- Zgen is the longitudinal position in cm,
- Phase0 & wgen are the phase and energy references of the beam,

Compressions format

Error set of data

At the end of a simulation, the list of the final errors applied on each element can be found on the calculation directory in the file “Error_Datas.txt”. A file using the same syntax can be use associated with the boith commands, ERROR_STAT_FILE & ERROR_DYN_FILE.
During a statistical error study, for each linac a file, “Error_Datas_XX.txt” is saved.

```
QUAD_ERROR dx(mm),dy(mm),drx(°),dry(°),drz(°),dG(%),L(mm),dG3(%),dG4(%),dG5(%),dG6(%)
CAV_ERROR dx(mm),dy(mm),drx(°),dry(°),drz(°),dE(%),dPhase(°),L(mm)
BEND ERROR dx(mm),dy(mm),drx(°),dry(°),drz(°),dg(%),dz(mm)
BEAM_ERROR dx(mm), dy(mm), dϕ(°), dxp(mrad), dyp(mrad), de(MeV), dEx(%), dEy(%), dEz(%), mx(%), my(%), mz(%), dib(mA ),αxx'_min,
αxx'_max, βxx'_min(mm/mrad), βxx'_max(mm/mrad), αyy'_min, αyy'_max, βyy'_min(mm/mrad), βyy'_max(mm/mrad), αzdp_min,
αzdp_max, βzdp_min(mm/mrad), βzdp_max(mm/mrad)

ERROR_BEAM 0.0180812 0.029287 0 0.0815901 -0.0979198 -0 0 0 0 0 -0 0 -0 0 0 0 0 0 0 0 0 0 0 0 0 0
QUAD_ERROR [1] -0.0358043 -0.106052 -0.0152857 -0.0392236 0.0497174 -0.790136 0.73 0 -0 -0 0
QUAD_ERROR [2] -0.0629623 -0.053688 -0.0216135 -0.07538 0.0389759 0.0504835 0 0 -0 -0 0
CAV_ERROR [4] 0.00226118 0.00857076 0.00804547 0.000738873 0 0.785965 0.0847529 0.73
QUAD_ERROR [5] 0.0661981 -0.0781597 -0.0605019 0.0523095 0.0493564 0.0607798 0 0 0 0 0
CAV_ERROR [160] 0.00875094 0.00181656 -0.0293326 -0.0246627 0 0.618743 -0.408912 0.3
QUAD_ERROR [167] -0.0792799 -0.0387784 0.010346 0.00170561 0.193866 0.958619 0.13 0 0 -0 -0
QUAD_ERROR [168] -0.0792799 -0.0387784 0.010346 0.00170561 0.193866 0.958619 0.13 0 0 -0 -0
QUAD_ERROR [169] -0.0792799 -0.0387784 0.010346 0.00170561 0.193866 0.958619 0.13 0 0 -0 -0
QUAD_ERROR [170] -0.0792799 -0.0387784 0.010346 0.00170561 0.193866 0.958619 0.13 0 0 -0 -0
QUAD_ERROR [176] -0.0648002 0.0634561 -0.00654776 0.00490306 -0.162641 -0.214648 0.13 -0 0 0 0
CAV_ERROR [183] -0.000958443 -0.00685156 0.0171741 0.0266001 0 -0.0191761 0.132693 0.3
QUAD_ERROR [190] -0.0676107 0.0954914 -0.0285646 0.0129899 -0.0218936 0.436949 0.13 0 0 0 0
QUAD_ERROR [191] -0.0676107 0.0954914 -0.0285646 0.0129899 -0.0218936 0.436949 0.13 0 0 0 0
QUAD_ERROR [192] -0.0676107 0.0954914 -0.0285646 0.0129899 -0.0218936 0.436949 0.13 0 0 0 0
QUAD_ERROR [193] -0.0676107 0.0954914 -0.0285646 0.0129899 -0.0218936 0.436949 0.13 0 0 0 0
QUAD_ERROR [199] -0.0884702 0.103422 0.0626978 0.0526471 0.164511 -0.561579 0.13 0 0 0 0
CAV_ERROR [206] 0.00760207 0.00346884 0.00935432 -0.00380327 0 -0.965653 0.852592 0.3
```

[X] is the element number, *L(mm)* parameters can be use to find the particle coordinates at the input of the considered element.

Horizontal example:

$$x(mm) = x - dx + drx \cdot L \cdot \frac{\pi}{180}\,, \qquad x'(rad) = x' - drx \cdot \frac{\pi}{180}$$

Error file result

The final results can be found on the calculation directory. Named "*Error_study_Name_TRA.txt*" when the result comes from an envelope calculation and "*Error_study_Name_PAR.txt*" when it is a PARTRAN study. The format of these files is the same for each kind of error. For each step of calculation one line of 32 parameters is written with the following format.

- Step of error (0->1)
- 1-(Nbr of particles)/(Nbr of particles, reference case)
- (Emittance rms xx',yy',zz')/ (Reference rms emittance xx',yy',zz')-1
- X beam center (m)
- Y beam center (m)
- X' beam center (rad)
- Y' beam center (rad)
- Energy beam center (keV)
- Phase beam center (deg)
- RMS Beam X size (m)
- RMS Beam Y size (m)
- RMS Beam X' size (m)
- RMS Beam Y' size (m)
- RMS Beam Energy size (keV)
- RMS Beam Phase size (deg)
- Halo parameter xx'
- Halo parameter yy'
- Halo parameter zz'
- Horizontal dispersion (m)
- Vertical dispersion (m)
- Horizontal dispersion / dz
- Vertical dispersion /dz
- Apha_XX'
- Beta_XX' (m/rad)
- Apha_YY'
- Beta_YY' (m/rad)
- Apha_PE
- Beta_PE (deg/MeV)
- Emittance 4D/Reference emittance 4D-1
- Emittance 6D/Reference emittance 6D-1

Halo definition.

All these values are relative to the output beam without errors.

In case of statistical study, where each step of calculation contains several runs, the format becomes:

- Step of error (0->1)
- AVERAGE(1-(Nbr of particles)/(Nbr of particles, reference case))
- (AVERAGE(Emittance rms xx',yy',zz'))/ (Reference rms emittance xx',yy',zz')-1
- RMS(X beam center (m))
- RMS(Y beam center (m))
- RMS(X' beam center (rad))
- RMS(Y' beam center (rad))
- RMS(Energy beam center (keV))
- RMS(Phase beam center (deg))
- AVERAGE(RMS Beam X size (m))
- AVERAGE(RMS Beam Y size (m))
- AVERAGE(RMS Beam X' size (m))
- AVERAGE(RMS Beam Y' size (m))
- AVERAGE(RMS Beam Energy size (keV))
- AVERAGE(RMS Beam Phase size (deg))
- AVERAGE(Halo parameter xx')
- AVERAGE(Halo parameter yy')
- AVERAGE(Halo parameter zz')
- AVERAGE(Apha_XX')
- AVERAGE(Beta_XX' (m/rad))
- AVERAGE(Apha_YY')
- AVERAGE(Beta_YY' (m/rad))
- AVERAGE(Apha_PE)
- AVERAGE(Beta_PE (deg/MeV))
- AVERAGE(X beam center (m))
- AVERAGE(Y beam center (m))
- AVERAGE(X' beam center (rad))
- AVERAGE(Y' beam center (rad))
- AVERAGE(Energy beam center (keV))
- AVERAGE(Phase beam center (deg))

And a file call "*Error_study_Name_TRA_tot.txt*" or "*Error_study_Name_PAR_tot.txt*" is written containing all run results.

Partran and Toutatis output

The final multiparticle results contain one line by element output, the first line being the input beam parameters. The format is like following.

- Element number
- Element position (m)
- Relativistic parameters: (γ-1)
- Centroid position: x(mm), y(mm), Phase(°), x'(mrad), y'(mrad), W(MeV)

- RMS_SIZE(x(mm), y(mm), Phase(deg))
- RMS (xx'(mm.mrad), yy'(mm), Phase.Energy(deg.MeV))
- Normalized rms emit: xx'(mm.mrad), yy'(mm.mrad), PW (Deg.MeV).
- Halo parameters: ($H_{xx'}$, $H_{yy'}$, $H_{z.dp/p}$)
- Number of particles
- Phase advance with space charge (deg/mm): σ_x, σ_y, σ_z.
- Emittance at 99%: $\varepsilon_{xx'}$, $\varepsilon_{yy'}$, $\varepsilon_{z.dp/p}$
- $\Delta\varphi_s$ (°), average beam phase - reference beam phase
- ΔW_s (MeV), average beam energy - reference beam energy
- Beam current (mA) used for space charge calculation
- Aperture (mm)
- Normalized 4D transverse emittance $E_{xx'yy'}$ (mm.mrad)^2
- Normalized rms emit (mm.mrad): $\varepsilon_{rr'}$.
- Phase advance with space charge (deg/mm): σ_r
- Lost power (w)
- Maximum excursion particle : Xmax(mm), Ymax(mm),
- Normalized long. rms emit: εz.dp/p (mm.mrad) [replace PW(Deg.Mev)]
- Dispersion: Dh (mm), Dv(mm)
- Derivative dispersion: Dh' (mrad), Dv'(mrad)
- σ_{xy}
- $\sigma_{x'y'}$

Since TraceWin version 2.1.0.0 longitudinal rms emittance PW is set to zero and has been replaced by εz.dp/p

Random seed file

User has the possibility to set the random seed, in "main" page, in order to regenerate an error case. For remote computing the seed values are saved in a file name "*Random_seed.log*".

Transfer matrix file

"*Transfer_matrix1.dat*" file contains the 6x6 tranfer matrix of the structure from the first to the last element. The syntax is similar to the outputs visible in the tool "*Structure tranfer Matrix*" from the "*Charts*" tab-sheet.

In "*Charts*" tab-sheet the option "*Include structure errors in transfer matrix*" defined if the transfer matrix shown in the corresponding tools or saved into the file take or not into account the errors applied to the structure.

During error study, if option "*Keep all result files*" is selected in tab_shhet "*Errors*" a list of file ""*Transfer_matrix1.dat_XXX*" is created.

Elements

Drift

Mnemonic	Parameter	Definition
DRIFT	L R R_y R_x_*shift* R_y_*shift*	Length (mm) Aperture (mm) Aperture (mm) Orizontal aperture shift (mm) Vertical aperture shift (mm)

If R_y equals 0, aperture is circular with R radius.
If R_y is not equal 0, aperture is rectangular with R the half dimension in x plane and Ry in y plane.
$R_{x/y}$_*shift* allows to introduce a non-cental aperture.

Link to Drift matrix

Quadrupole

Mnemonic	Parameter	Definition
QUAD	L G R Θ G_3 / u_3 G_4 / u_4 G_5 / u_5 G_6 / u_6 *GFR*	Length (mm) Magnetic field gradient (T/m) Aperture (mm) Skew Angle (°) Sextupole gradient (T/m^2) or relative sex. component Octupole gradient (T/m^3) or relative oct. component Decapole gradient (T/m^4) or relative deca. component Dodecapole gradient (T/m^5) or relative dode. component Good field radius (mm)

Attention to the TraceWin gradient definition.

Red values are optional.
If L equals 0, the quadrupole is simulated as thin lens and all gradients have to be replaced by their integral (over longitudinal direction) values.
Multipole kicks are applied at the middle of the quadrupole.

If *GFR* is non null, u_n are relative multipole components will defined such as:

$$u_n = \frac{10000 \times G_n \times GFR^{n-2}(m)}{G}$$

Link to Quadrupole matrix

Beam Rotation

Mnemonic	Parameter	Definition
BEAM_ROT	*θxy* *θxz* *θyz* *dx* *dy* *dxp* *dyp* *dz* *centroid_flag*	Angle (°) in the XY space around Z Angle (°) in the XZ space around Y Angle (°) in the YZ space around X X shift (mm) Y shift (mm) Xp shift (mrad) Yp shift (mrad) Z (mm) 0 : turn around gravity center

Rotations are first performed in the following order: around Z, Y and X, and finally the shifts. Centroid_flag defines if the rotation is done around beam center of gravity (=0) or if the rotation is done around the synchronous particle (also applied on beam centroid, except energy). BEAM_ROT is considered as an element

Link to Beam rotation matrix

Thin Lens

Mnemonic	Parameter	Definition
THIN_LENS	*fx* *fy* *R*	Focal Length (m) Focal Length (m) Aperture (mm)

Link to Thin lens matrix

Thin Matrix

Mnemonic	Parameter	Definition
THIN_MATRIX	*lg* a_{00} to a_{55}	Length (mm) Matrix terms, row per row

The 36 terms of a transfer matrix have to be set row by row from a_{00} to a_{55}. Length, *lg*, is just used in graphic view.
Link to **matrix format *R***
The associated phase-space are respectively: *x* (m), *x'* (rad), *y* (m), *y'* (rad), *z* (m), *dp*/*p* (rad)

Quadrupole Example: THIN_MATRIX 10.0 1 0 0 0 0 0 -2.5 1 0 0 0 0 0 0 1. 0 0 0 0 0 2.5 1 0 0 0 0 0 0 1 0 0 0 0 0 0 1

Electrostatic Quadrupole

Mnemonic	Parameter	Definition
QUAD_ELE	*L* *Vo* *R* *Θ* V_3 / u_3 V_4/ u_4 V_5/ u_5 V_6/ u_6 *GVR*	Length (mm) Voltage between electrodes (V) Aperture (mm) Skew Angle (°) Sextupole voltage gradient component (V/m) or relative Octupole voltage gradient component (V/m^2) or relative Decapole voltage gradient component (V/m^3) or relative Dodecapole voltage gradient component (V/m^4) or relative Good voltage radius (mm)

Attention to the TraceWin gradient definition.

Vo is the voltage difference between neighboring electrodes (+*Vo*/2 on one and –*Vo*/2 on its neighbors). The distance between opposite electrodes is 2*R*.
In order to keep coherence with old TraceWin version, all electrostatic components are in voltage: (V) and not in field: E (V/m)
If *L* equals 0, electrostatic quadrupoles are simulated as thin lens and all voltage components have to be replaced by their integral (over longitudinal direction) values.
Multipole kicks are applied at the middle of the electrostatic quadrupole.

If *VFR* is non null, u_n are relative multipole components will defined such as:

$$u_n = \frac{10000 \times V_n \times GFR^{n-2}}{V_à}$$

Link to Quadrupole matrix

Bunched cavity or thin gap

Mnemonic	Parameter	Definition
GAP	E_0TL	Effective gap voltage (V)
	θ_s	RF phase (deg) (absolute or relative)
	R	Aperture (mm)
	P	0: θ_s is relative, 1: θ_s is absolute phase, 2: θ_s is relative + beam phase error set to 0 3: θ_s if set to 0 all phases are absolute including phase error resulting in preceding structures. If not set to 0, θ_s is the relative phase including phase error resulting in preceding structure.
	β_s	Particle reduced velocity
	T_s	Transit time factor
	kT'_s	(*)
	$k^2T''_s$	(*)

Red parameters are optional

(*) See Transit time factor definition

Bunched cavity or thin gap matrix

Sinus cavity or CCL

Mnemonic	Parameter	Definition
CAVSIN	L	Length (mm)
	N	Cell number
	E_oT	Average accelerating field (V/m)
	θ_s	Phase of the synchronous particle at the entrance(deg)(*)
	R	Aperture (mm)
	P	1: θ_s is absolute phase, 0: θ_s is relative

(*) Use SET_SYNC_PHASE command in order to change this phase as the synchronous phase

Link to Sinus cavity or CCL matrix

Bending magnet

Mnemonic	Parameter	Definition
BEND	α $\lvert\rho\rvert$ N R HV	Bend angle in the rotation plane (deg) Curvature radius of central trajectory (mm) Field gradient index Aperture (mm) 0 : horizontal, 1 : vertical

Warning: A bend always have to be surrounded by an edge element, even if edge is set to zero.
By definition: a positive bend (denoted α>0) bends the trajectory to the right in the horizontal plane, whatever the sign of the particle charge state.
The field gradient index is treated in the first order and applies only for horizontal bending.

Link to Bending magnet matrix

Edge angle on bending magnet

Mnemonic	Parameter	Definition
EDGE	β $\lvert\rho\rvert$ G K_1 K_2 R HV	Pole face rotation angle (deg) Curvature radius of bend (mm) Total gap of magnet (mm) Fringe-field factor (default = 0.45) Fringe-field factor (default = 2.80) Aperture (mm) 0 : horizontal, 1 : vertical

By definition: an edge focalizes if $\beta < 0$, whatever the curvature radius sign, the bending angle sign and the particle charge state.
Set K_1 and K_2 to a very small values to disable fringe field estimation.
G is used for particle loss estimation in bend.

Link to Edge angle on bending magnet matrix

Electrostatic bend

Mnemonic	Parameter	Definition
BEND_ELE	α $\lvert\rho\rvert$ ne R HV	Bend angle in the rotation plane (deg) Curvature radius of central trajectory (mm) 1:Cylindrical, 2:Spherical, 3:Toroidal Aperture (mm) 0 : horizontal, 1 : vertical

By definition: a positive bend (denoted α>0) bends the trajectory to the right in the horizontal plane, whatever the sign of the particle charge state.
Only cylindrical bends are available in multiparticle mode.

Circular or rectangular aperture

Mnemonic	Parameter	Definition
APERTURE	*dx* *dy* *n*	X half width (mm) or radius hole (pepperpot) Y half width (mm) or distance between hole (pepperpot) Type : 0 : Rectangular aperture 1 : Circular aperture 2 : Pepperpot mode 3 : Rectangular aperture with *dx* & *dy* corresponding to a beam fraction intercepted by the aperture (adjusted with 0.1 mm step) if value <1, otherwise *dx* or *dy* are used as type=0 4 : Horizontal finger with *dx*=finger center position, *dy*=total finger width. 5 : vertical finger with *dx*=finger center position, *dy*=total finger width. 6 : Ring aperture if particle radius, *r*>*dy* or *r*<*dx* with (*dx*<*dy*) particle is lost.

Space charge compensation

Mnemonic	Parameter	Definition
SPACE_CHARGE_COMP	*k*	Beam current is compensated by a factor *k*

Set a beam new current: $Ib = (1\text{-}k)\times Ib$.

Beam current

Mnemonic	Parameter	Definition
CURRENT	*Ib*	Beam current (mA)

Set a beam new current: *Ib* (mA).

Solenoid

Mnemonic	Parameter	Definition
SOLENOID	*L* *B* *R*	Length (mm) Magnetic field (T) Aperture (mm)

Solenoid treated as thick length.

Link to Solenoid matrix

Thin steering

Mnemonic	Parameter	Definition
THIN_STEERING	*BLx or ELx* *Bly or ELy* *r* *Elec*	x-component (T.m or V) y-component (T.m or V) Aperture (mm) 0: magnetic deviation (default) 1: electric deviation

Transverse kick.

Link to Thin steering matrix

DTL cell

Mnemonic	Parameter	Definition
DTL_CEL	L	Cell length (mm)
	Lq_1	First ½ quadruploe length (mm)
	Lq_2	Second ½ quadruploe length (mm)
	g_c *(*)*	Gap center shift (mm) (*)
	B_1 *'*	First magnetic field gradient (T/m)
	B_2 *'*	Second magnetic field gradient (T/m)
	E_oTL	Effective gap voltage (V)
	θs	RF phase (deg)
	R	Aperture (mm)
	P *(**)*	0: θ_s is relative (**), 1: θ_s is absolute phase, 2: θ_s is relative + beam phase error set to 0 3: θ_s if set to 0 all phases are absolute including phase error resulting in preceding structures. If not set to 0, θ_s is the relative phase including phase error resulting in preceding structure.
	β_s	Particle reduced velocity
	T_s	Transit time factor
	kT'_s	(***)
	$k^2T''_s$	(***)

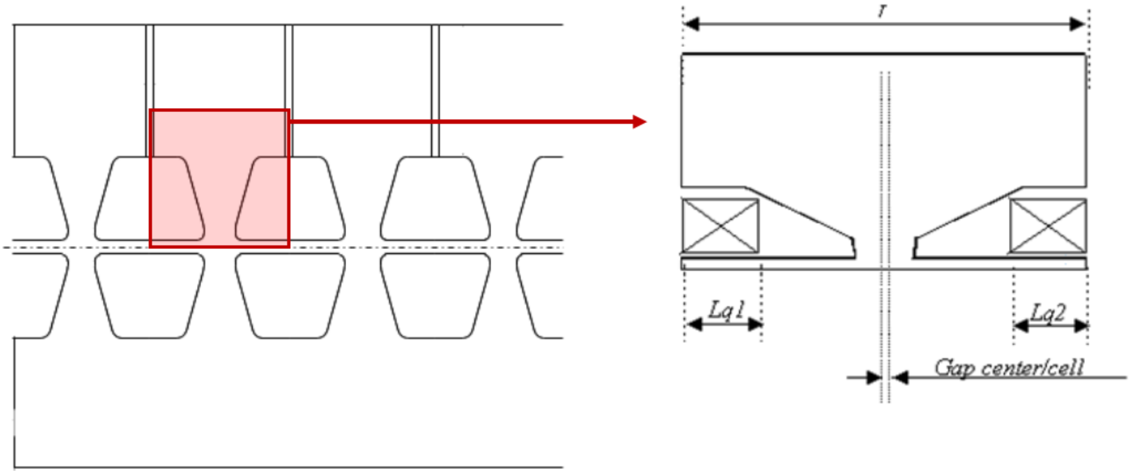
Define a cell of a classical Drift Tube Linac (DTL).

(*) The g_c*(mm)* dimension is defined as: $gap\ position = \frac{L}{2} - g_c$

(**) This following example is the most realistic RF sequence: All synchronous phase are absolute phase and calculated according to the cell lengths, excepted for the first one which is set to the relative value. The phase error resulting in preceding structures is taken into account. You can canceled the preceding structure phase error by replacing *type 3* by *type 2* for the first cell

```
LATTICE 4 1
DTL_CEL 68.729 22.5 22.5 0.0125565 52.5 -52.5 176568 -30 10 3 0.0807374 0.778491 -0.377246 -0.1456
DTL_CEL 70.4468 22.5 22.5 0.00912044 -52.5 -52.5 182799 0 10 3 0.0827489 0.786305 -0.366349 -0.150419
DTL_CEL 72.1883 22.5 22.5 0.00491998 -52.5 52.5 188980 0 10 3 0.0847886 0.79329 -0.356535 -0.154494
DTL_CEL 73.9529 22.5 22.5 0.000837313 52.5 49.5 195297 0 10 3 0.0868553 0.800245 -0.346592 -0.157865
DTL_CEL 75.74 22.5 22.5 -0.0035088 49.5 -49.5 201675 0 10 3 0.0889485 0.806881 -0.336968 -0.16056
```

(**) See Transit time factor definition
Red parameters are optional.
If β_s does not equal *0*, T_s is mandatory.



Link to DTL cell matrix

Cavity multi-gap

Mnemonic	Parameter	Definition
NCELLS	*Mode*	(0)2π, (1)π, (2)π & 2π
	N_c	Number of cell
	β_g	Geometric β > 0: all cell lengths are the same according to β_g = 0: cell lengths are set according to beam velocity < 0: cell lengths are set according to an initial beam velocity $-\beta_g$, which is increasing from cell to cell according to θ_s and E_oT parameters.
	E_oT	Effective gap voltage (V/m)
	θ_s	RF phase at the first gap position (deg) (**)
	R	Aperture (mm)
	P	0: θ_s is relative , 1: θ_s is absolute phase, 2: θ_s is relative + beam phase set to 0
	kE_oT_i	Input field correction, $E_o = E_o*(1+k)$
	kE_oT_o	Output field correction, $E_o = E_o*(1+k)$
	dz_i	First gap displacement (mm)
	dz_o	Last gap displacement (mm)
	β_s	Particle reduced velocity
	T_s	Transit time factor of the middle gaps
	kT'_s	(*)
	$k^2T''_s$	(*)
	T_i	Transit time factor of the input gaps
	kT'_i	(*)
	$k^2T''_i$	(*)
	T_o	Transit time factor of the output gaps
	kT'_o	(*)
	$k^2T''_o$	(*)

Modelisation of a multicell cavity by a set of gaps, with one gap at the middle of each cell.
Red parameters are optional
If β_s is not equal to *0* T_s become needed

(*) See Transit time factor definition
(**) Use SET_SYNC_PHASE command in order to change this phase as the synchronous phase

Link to Cavity multi-gap matrix

RFQ cell

Mnemonic	Parameter	Definition
RFQ_CELL	*V*	Effective gap voltage (V)
	R₀	Vane average radius (mm) (*)
	A10	Acceleration parameter
	m	Vane modulation
	L	Cell length (mm)
	θs	RF phase (deg)
	Type	
	Tc	Transverse curvature (mm)
	dP	Phase shift (allow to recenter beam phase)
	a	Minimum radius (**)

(*) Vane geometry plotted in envelope chart is made for illustrate the RFQ_CELL element. It's a rought drawing, expecialy for front-end cells.

(**) Optional parameter, but necessary defined for TWOTERM RFQ.

Link to RFQ cell matrix

Diagnostic elements

Diagnostics **measure** beam properties at given position.
These measurements are used for the **automatic tuning** of transport elements to set the measured beam properties to **wanted** ones. The diagnostics and the associated adjustable transport elements are then linked with a **common number**. The association is made by preceding the transport elements with the adjust commands followed by the number of the associated diagnostics. The tuning is made in the order of the diagnostic numbers.
In these conditions, a uniform random measurement error is used according to the diagnostic **accuracy**. If accuracy > 0, the errors are uniformly distributed between ± accuracy value. If accuracy < 0, the errors are Gaussian distributed; accuracy value is then the rms value of the distribution.

Mesured property	Mnemonic	Parameter	Definition
Current	**DIAG_CURRENT**	*N* *Ib*	Diagnostic number Wanted beam current (mA)
Delta Current (2 are needed)	**DIAG_DCURRENT**	*N* Ib_n-Ib_{n-1} *dIb*	Diagnostic number Wanted delta beam current (mA) Accuracy (mA)
Positions	**DIAG_POSITION**	*N* *X* *Y* *dm*	Diagnostic number Wanted X beam position (mm), (If Y'<1e50) Wanted Y beam position (mm), (If Y'<1e50) Diagnostic Accuracy (mm)
Divergences	**DIAG_DIVERGENCE**	*N* *X'* *Y'* *dm*	Diagnostic number Wanted X divergence (mrad), (If Y'<1e50) Wanted Y divergence (mrad), (If Y'<1e50) Diagnostic Accuracy (mrad)
Sizes	**DIAG_SIZE**	*N* *Sx* *Sy* *ΔP* *Dm* *dΔP* *fo*	Diagnostic number Wanted X rms size (mm) (If ≠ 0) Wanted Y rms size (mm) (If ≠ 0) Wanted rms Phase spread (°) (If ≠ 0) Size Accuracy (mm) Phase spread Accuracy (°) *ΔP* Low-pass filter frequency (MHz)
Divergences if S<0 Twiss parameter alpha will set <0	**DIAG_SIZEP**	*N* *Sx'* *Sy'* *ΔW* *Dm* *dW*	Diagnostic number Wanted X'rms divergence (mrad) (If ≠ 0) Wanted Y'rms divergence (mrad) (If ≠ 0) Wanted rms Energy spread (MeV) (If ≠ 0) Divergence Accuracy (mrad) Energy spread Accuracy (%)
Delta size Xrms-Yrms	**DIAG_DSIZE**	*N* x_n-y_n *dm*	Diagnostic number Wanted x-y rms beam delta size (mm) dx-y size Accuracy (mm)
Delta size	**DIAG_DSIZE2**	*N*	Diagnostic number

between 2 positions (At least 2 are needed)		x_n-x_{n-1} y_n-y_{n-1} *dm*	Wanted x rms beam delta size (mm) Wanted y rms beam delta size (mm) dx & dy size accuracy (mm)
Delta phase spread Measurement 2 (At least 2 are Needed)	**DIAG_DSIZE3**	*N* ΔP_n-ΔP_{n-1} $d\Delta P$ *fo*	Diagnostic number Wanted rms delta phase spread (°) Phase spread accuracy (°) $d\Delta P$ Low-pass filter frequency (MHz)
Delta divergence between 2 positions (At least 2 are needed)	**DIAG_DPSIZE2**	*N* x'_n-x'_{n-1} y'_n-y'_{n-1} *dpm*	Diagnostic number Wanted x’ rms beam delta div. (mrad) Wanted y’ rms beam delta div. (mrad) dx' & dy’ div. accuracy (mrad)
Phase measurement	**DIAG_PHASE**	*N* Θ	Diagnostic number Wanted centroïd Phase (°)
Energy measurement	**DIAG_ENERGY**	*N* *W* *dw*	Diagnostic number Wanted Energy (MeV) Accuracy (%)
Beam energy - Perfect linac energy measurement	**DIAG_DENERGY**	*N* *W* *dw*	Diagnostic number Wanted Delta Energy (MeV) Accuracy (%)
Beam phase - Perfect linac phase measurement	**DIAG_DPHASE**	*N* Θ	Diagnostic number Wanted delta centroïd Phase (°)
Luminosity	**DIAG_LUMINOSITY**	*N* *Lu* *Dlu*	Diagnostic number Wanted luminosity (mm^{-2}) Luminosity accuracy (mm^{-2})
Waist setting	**DIAG_WAIST**	*N* *fx* *fy* *dxy*	Diagnostic number XX’ waist asked (for *fx* not equal to *0*) YY’ waist asked (for *fy* not equal to *0*) Transverse waist Accuracy
Achromat setting	**DIAG_ACHROMAT**	*N* *Ele#* $^{(**)}$ *f1* *f2* *plan*	Diagnostic number First element number $^{(**)}$ If =1 set achromatic position If =1 set achromatic angle 0 : X or Y or both planes acccoding to bend direction are concerned 1 : only X plan is concerned 2 : only Y plan is concerned 3 : both planes are concerned
Emittance setting (rms values)	**DIAG_EMIT**	*N* *Exx’* *Eyy’* *Epw*	Diagnostic number Wanted emittance if greater than 0 Wanted emittance if greater than 0 Wanted emittance if greater than 0
Emittance setting	**DIAG_EMIT_99 (*)**	*N*	Diagnostic number

(99% value or others)		*Exx'* *Eyy'* *Epw*	Wanted emittance if greater than 0 Wanted emittance if greater than 0 Wanted emittance if greater than 0
Halo setting	**DIAG_HALO**	*N* *Hx* *Hy* *Hz*	Diagnostic number Wanted haloX if greater than 0 Wanted haloY if greater than 0 Wanted haloZ if greater than 0
Transfer matrix setting	**DIAG_SET_MATRIX**	*N* *Ele#* (**) *Row(i)* *Column(j)* *k* *Mij*	Diagnostic number Transfer matrix form element number, *Ele#*, to diag. position (**) Row transfer matrix term Column transfer matrix term Corrector coefficient Wanted transfer matrix term value
Beam Twiss parameters setting Parameter is used if it is not equal to 0, if you want *α=0* set very small α	**DIAG_TWISS**	*N* $\alpha_{xx'}$ $\beta_{xx'}$ $\alpha_{yy'}$ $\beta_{yy'}$ $\alpha_{zdp/p}$ $\beta_{zdp/p}$	Diagnostic number Wanted alpXX' Wanted betXX' (mm/mrad) Wanted alpYY' Wanted betYY' (mm/mrad) Wanted alpZdp/p Wanted betZdp/p (mm/mrad) = 10 x time betZdp/p (mm/%)
Beam separation setting	**DIAG_SEPARATION**	*N* *k* *S* *0/1* *ks* *kc*	Diagnostic number Correction factor (set 1) Wanted separation (ks.Size/Position+kc.Position) 0 for X plane, 1 for Y Weigth on separation Weigth on position

(*) 99% fraction must be set in “Multiparticle” tabsheet, then go to “99.XX% emittance and halo”,then select the fraction you want.

(**) if (Ele#>0) then Ele# is absolute
if (Ele#<0) then Ele# is relative, e.g. -15 means relative to location 15 elements upstream.

TraceWin can be used to verify if a beam line contains enough diagnostics to control the beam and correct the errors coming from the input beam or from the different element errors. In order to put one diagnostics in the line use the elements “*DIAG_...*” followed by the diagnostic number and the wanted parameters which have to be imposed at the diagnostic location (current, position, size or emittance). Precede the associated adjustable transport elements by the adjust commands followed by the number of the associated diagnostic.
There are two ways of using these diagnostics. The first one is for example to see if your design is able to control the beam position at a given location when you input beam is not at the center or if a few

elements, like steerers induce misalignment. The second way is to use diagnostics in an error study in order to see if you scheme of misalignment correction for example is efficient enough.
The diagnostics adjustments are independent process, which occurs after the matching process. If any diagnostic element is present any adjustment is started. Finally, at the end of a line design process, only diagnostic elements should be used to tune it.
Warning: They are treated in the order of the diagnostic number.

Link to Adjust and diagnostics examples

Note: many transport elements can be coupled to many diagnostics. The tuning is made by minimizing a criterium being the quadratic sum of the differences between effective beam properties and wanted ones divided by the wanted one. The relative weight on the criterion of each difference can be changed by putting a weight *factor* between brackets like following:

DIAG_EMIT(1e-4) 1 0.1 0.1 0.2 , here this diagnostic emittance criteria is reduced by a factor 10000.

MY_DIAG(my_diag_func)(1e-4) 1 0.1 0.1 0.2 , for diagnostic developed by user.

By default, without bracket, the factor is set to 1.

NEW (2.13.0.5): Diagnostics transverse positions can be linked to following element position affected by errors and SHFIT commands. The syntax consists to add @ to the diagnostic name, ex: “DIAG_POSITION@ “.

NEW (2.15.0.0): Diagnostics can overlop Field map elements, see SHIFT_IN_FIELD_MAP

Funneling gap

A funneling gap models a RF cavity filed with a **RF transverse electric field** allowing to merging two beam lines into one.

	Mnemonic	Parameter	Definition
Funneling Gap	**FUNNEL_GAP**	E_oTL Θ_s R p	Effective gap voltage (V) RF phase (deg) (absolute or relative) Aperture (mm) 1: θ_s is absolute phase, 0: θ_s is relative
Frame change	**CHFRAME**	X_o X'_o	(mm) (deg)

Link to Funneling gap matrix

Alpha magnet

An alpha magnet is a magnetic element elongating the trajectory of particles with higher energy. It is used to compress bunched beam.

Mnemonic	Parameter	Definition
ALPHA_MAGNET	*Θ* *K* *R* *plan*	Entrance angle (°) (T/m) Aperture (mm) 0 (x)/1 (y)

Link to Alpha magnet matrix

Electrostatic Acceleration

The element simulates acceleration in electrostatic accelerators.

Mnemonic	Parameter	Definition
ELECTROSTA_ACC	*Vo* *L* *K* *R*	Voltage (V) Length (mm) Transverse defocal (eV/ mm²) Aperture (mm)

Link to Electrostatic Acceleration matrix

Field Map

This element introduces static or RF electromagnetic elements whose field maps (1D, 2D or 3D) are given in files. These maps can be superimposed. Dedicated to linac, it was initially dedicated to fields not deflecting the main trajectory (quadrupoles, solenoids, cavities). Since version xxx, it can also deal with fields deflecting the beam main trajectory.

Mnemonic	Parameter	Definition
FIELD_MAP	*geom* *L* θ_i *R* k_b *ke* *Ki* *Ka* *FileName* *P*	Field map type Field map length (mm) RF input field phase (°) Aperture (mm) Magnetic field intensity factor Electric field intensity factor Space charge compensation factor Aperture flag File name without extension (abs. or relative path) 0: θ_i is relative phase 1: θ_i is absolute phase

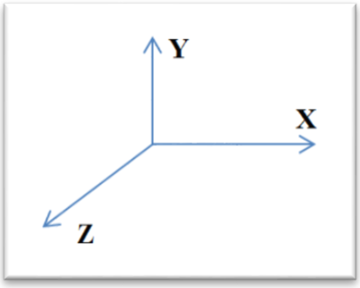
geom parameter is an integer, made of 5 figures, defining the field map type :

- unit figure (10^0) : static electric field,
- tens figure (10^1) : static magnetic field,
- hundreds figure (10^2) : RF electric field,
- thousands figure (10^3) : RF magnetic field,
- ten thousands figure (10^4) : 3D aperture map (since 2.4.0.1 version),

where each figure describes the field geometry :

0 - no field,
1 - 1D : Fz(z),
2 - not available,
3 - not available,
4 - 2D cylindrical static or RF electric field : Ez(r,z), Er(r,z) and Bθ(r,z) for RF,
5 - 2D cylindrical static or RF magnetic field : Bz(r,z), Br(r,z) and Eθ(r,z) for RF,
6 - 2D Cartesian field: Fx(x,y), Fy(x,y),
7 - 3D Cartesian field: Fx(x,y,z), Fy(x,y,z), Fz(x,y,z),
8 - 3D cylindrical field : Fr(r, θ,z), Fθ (r,θ,z), Fz(r,θ,z), not implemented yet,
9 - 1D : G(z), only use for magnetic quadrupole, thus *geom* parameter has to be set to *0090* (see 3D field development for more details).

By convention, one uses: Ez(x,y,z,t) = Ez0(x,y,z).cos(ωt+φ) and Bz(x,y,z,t) = Bz0(x,y,z).sin(ωt+φ).
Be careful, the relative sign of the electric and magnetic fields given in the files should be coherent with this convention.
For example, the given magnetic field should be the one obtained a quarter of RF period after the electric one. In another words, if the electric field amplitude is given at a time when it is fully real, one should give the opposite of the imaginary part of the magnetic field at the same time.
For cylindrical field the convention is following:
Bx= - Bθ(r,z) y/r
By= + Bθ(r,z) x/r
x, y, z is in a direct frame.



L is the field map length along z direction.

θi is the RF phase when the generatrix particle enters the cavity.
Use SET_SYNC_PHASE command in order to change this phase as the synchronous phase.

R is the pipe radius along synchronous trajectory.

kb and ***ke*** are multiplicative factors between the field (respectively magnetic and electric) stored in the file anf the field used in the simulation. *Kb* can be defined as power supply current, if an excitation curbe is defined, see EXCITATION_CURVE command

If ***Ki*** is greater than 0, a space charge compensation map or current map is readed in “*FileName*.scc” and *Ki* is the normlization factor (see Space charge compensation file syntax). In the case overlapping field map, the length of the space charge compensation map must cover the whole set of field maps and must start at position 0.

If ***Ka*** equals 0, a particle is considered as lost when it comes out of the aperture ***R*** or field_map frame size defined in field_map files. If ***Ka*** equals 1, a beam pipe radius map is read in "*FileName*.ouv" factor (see Aperture map file syntax). If ***Ka*** equals 2, *R* is not used and the particle is not lost when it comes out field map frame. That allows superposing small size field maps inside bigger one. In the case overlapping field map, the length of the pipe map must cover the whole set of field maps and must start at position 0.

FileName is the root name (without extension) of the files where are localized the field maps (extension: .bsz, .bsr, .edx …), the space charge compensation map (.scc) and the beam pipe radius evolution with z (.ouv). See the file formats description

The file extensions are the following:

- the first character indicates the field nature : either electric (→ 'e') or magnetic (→ 'b'),
- the second character indicates the field type : either static (→ 's') or RF (→ 'd'),
- the third character indicates the field component : 'x', 'y', 'z', 'r' (radial) or 'q' (azimuthal),

Field in 1D: the field can be described in 1D, field according to z position

- 1 file contains the static electric field : *.esz
- 1 file contains the static magnetic field : *.bsz
- 1 file contains the RF electric field : *.edz
- 1 file contains the RF magnetic field : *.bdz
- 1 file contains the G(z) : *.bsz

From v.2.3.1.8, if the ***geom*** parameter value is negative, the off-axis development is performed at second order (default: first order).

Field in 2D: the fields can be described in 2D Cartesian (invariant through translation on z axis) or in cylindrical (invariant through rotation around the z axis) coordinates.

- In Cartesian coordinates (x, y) :
  - 2 files contain the static electric field : *.esx, *.esy
  - 2 files contain the static magnetic field : *.bsx, *.bsy
  - 2 files contain the RF electric field : *.edx, *.edy
  - 2 files contain the RF magnetic field : *.bdx, *.bdy

- In cylindrical coordinates (r, z, θ) :
  - 2 files contain the static electric field : *.esr, *.esz
  - 2 files contain the static magnetic field : *.bsr, *.bsz
  - 3 files contain the RF fields (TM) : *.edr, *.edz, *.bdq
  - 3 files contain the RF fields (TE) : *.bdr, *.bdz, *.edq

Field in 3D: the fields can be described in 3D either Cartesian or cylindrical frame.

- In Cartesian coordinates (x, y, z)
  - 3 files contain the static electric field : *.esx, *.esy, *.esz
  - 3 files contain the static magnetic field : *.bsx, *.bsy, *.bsz
  - 3 files contain the RF electric field : *.edx, *.edy, *.edz
  - 3 files contain the RF magnetic field : *.bdx, *.bdy, *.bdz

- In cylindrical coordinates (r, θ, z)
  - 3 files contain the static electric field : *.esr, *.esq, *.esz

- 3 files contain the static magnetic field : *.bsr, *.bsq, *.bsz
- 3 files contain the RF electric field : *.edr, *.edq, *.edz
- 3 files contain the RF magnetic field : *.bdr, *.bdq, *.bdz

3D aperture: the aperture file has to be described in 3D Cartesian coordinates. ‘1’ for matter and ‘0’ for air.

- 1 file contains 3D aperture data : *.ouv

File name syntax:

- without extension and without path if files are in the data file (*.dat) directory,
- including path without extension if not.
- including path without extension between quote marks, if there are some space characters in the path.
- for simplicity, use “FIELD_MAP_PATH *path of my_field_map_files*” command at the beginning of the data file (.dat), replacing all the defined field map path. No quote is needed. Path can be defined as relative of absolute to ini project file path.

Link to Particle motion in electromagnetic field

Field map with curved reference trajectory

(No space-charge in multiparticle mode)

Each individual element, described by a field map, is associated to a FIELD_MAP keyword:
FIELD_MAP *xxxx L θ R kb ke Ke Ka Filename*

The file *Filename* contains the field value in a regular mesh (*x*, *y*, *z*) in the element frame $\left(\vec{e}_x,\vec{e}_y,\vec{e}_z\right)$.

If field maps of different elements are superposed, their respective positions are given in the laboratory frame (*X*, *Y*, *Z*) where:
- (*X*=0, *Y*=0, *Z*=0) is the position of the reference particle when entering the field map(s) superposition,
- $(\vec{e}_X,\vec{e}_Y,\vec{e}_Z)$ are along the associated directions in TraceWin when entering the field map(s) superposition.

In this frame, each field map is located using the SUPERPOSE_MAP keyword whose syntax is:
SUPERPOSE_MAP Z_0 X_0 Y_0 θ_{Z0} θ_{X0} θ_{Y0}

Where:
(X_0, Y_0, Z_0) gives to the position in [mm], in (*X*, *Y*, *Z*) frame, of the (*x*=0, *y*=0, *z*=0) point of the field map. To start at the *Z(mm)* position in a field map, set Z_0 = -*Z*.
(θ_{X0}, θ_{Y0}, θ_{Z0}) gives the rotation angles in [°] between $\left(\vec{e}_x,\vec{e}_y,\vec{e}_z\right)$ and $(\vec{e}_X,\vec{e}_Y,\vec{e}_Z)$,

θ_{X0} corresponds to the rotation angle around X axis (in YZ plane).

θ_{y0} corresponds to the rotation angle around Y axis (in XZ plane).
θ_{z0} corresponds to the rotation angle around Z axis (in XY plane).
The first applied rotation is around Z, then Y, and finally X.

Be careful to the order of parameters!

Before version 2.8.0.0, the preceding keywords were already valid and could be used only when the field map were not deviating the beam for which the reference trajectory was a straight line (describing, for example, solenoids, quadrupole or RF cavities).
From version xxx, they can also be used to indicate that the reference trajectory is curved by the elements. The SUPERPOSE_MAP_OUT keyword informs it to TraceWin. It has to be placed before the description of the field maps. Its syntax is:

SUPERPOSE_MAP_OUT Z_0 X_0 Y_0 θ_{Z0} θ_{X0} θ_{Y0}

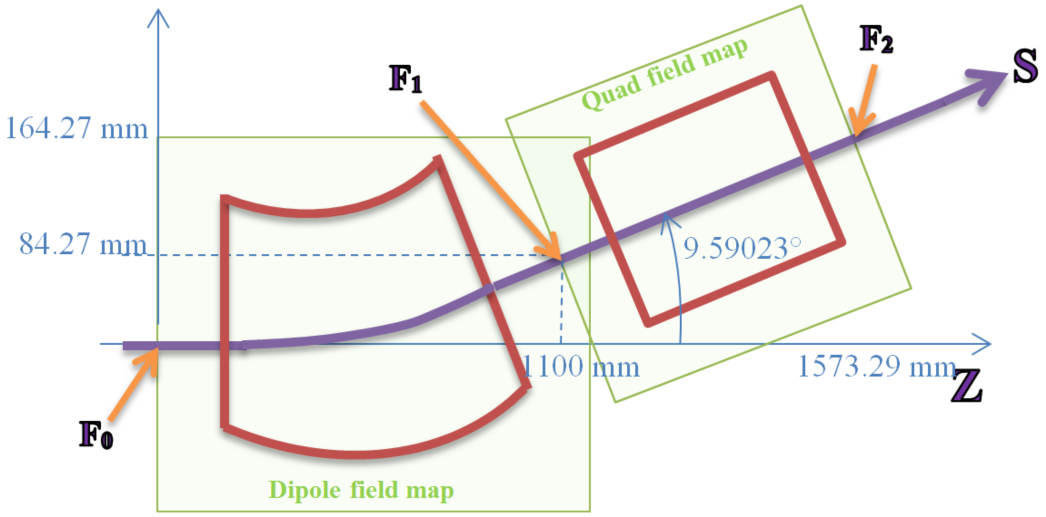
(X_0, Y_0, Z_0) gives to the position in [mm], in (X, Y, Z) frame, of the exit point of the simulation in the field map,
(θ_{X0}, θ_{Y0}, θ_{Z0}) gives the rotation angles in [°] between the reference trajectory directions at the exit point of the simulation in the field map $\left(\vec{e}_x, \vec{e}_y, \vec{e}_z\right)$ and $(\vec{e}_X, \vec{e}_Y, \vec{e}_Z)$,

θ_{X0} corresponds to the rotation angle around X axis (in YZ plane).
θ_{y0} corresponds to the rotation angle around Y axis (in XZ plane).
θ_{z0} corresponds to the rotation angle around Z axis (in XY plane).
The first applied rotation is around Z, then Y, and finally X.

Be careful to the order of parameters!

TraceWin calculates the deviated reference trajectory until it crosses the exit plan. The beam dynamics is then calculated around this reference trajectory. At the exit of calculation, TraceWin displaces and rotates the beam according to the positions and angles given with SUPERPOSE_MAP_OUT.

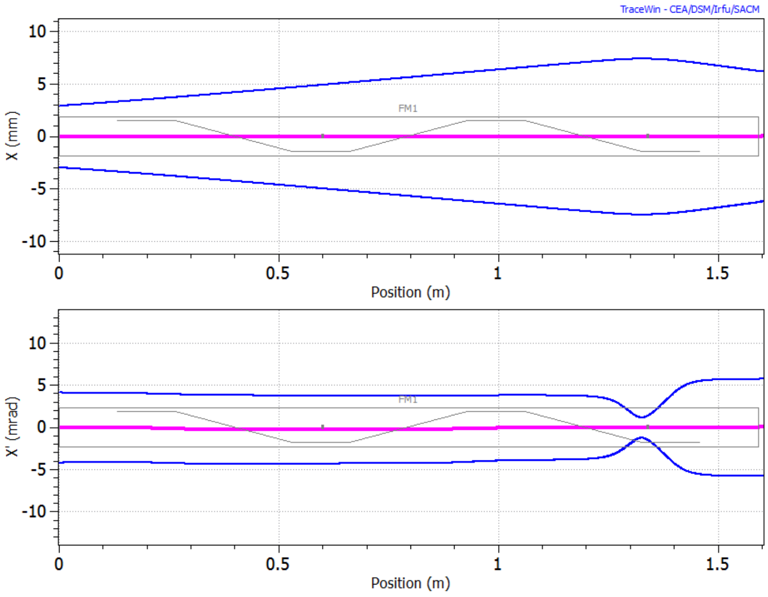
In the following example, available in the menu “example” of the help menu (“*field_map_dipole.ini*” project), a magnetic dipole field map is partially superposed with a quadrupole magnetic field map.



This configuration is described with the following TraceWin file:

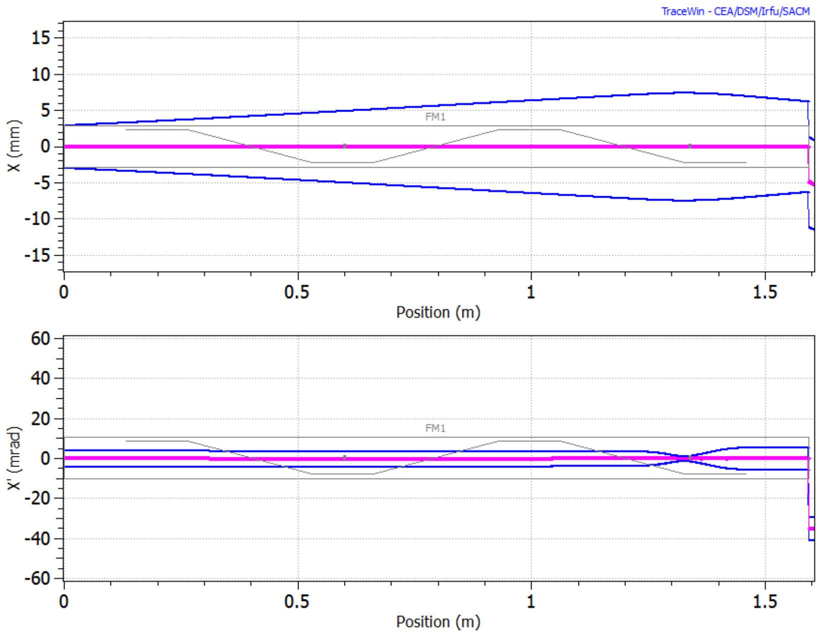
```
; F2
SUPERPOSE_MAP_OUT 1573.29 164.27 0 0 0 -9.59023
; F0
SUPERPOSE_MAP 0 0 0 0 0 0
FIELD_MAP 0070 1200 0 1000 1.0 0 0 0 dipole
; F1
SUPERPOSE_MAP 1100 84.27 0 0 0 9.59023
FIELD_MAP 70 480 0 200 10.0 10.0 0 0 qpole
END
```

The associated beam envelopes are given below.



If your ouptut frame is not correctly defined in SUPERPOSE_MAP_OUT, positions and angles shifts are applied at the end of the calculation in the superposed field_map.

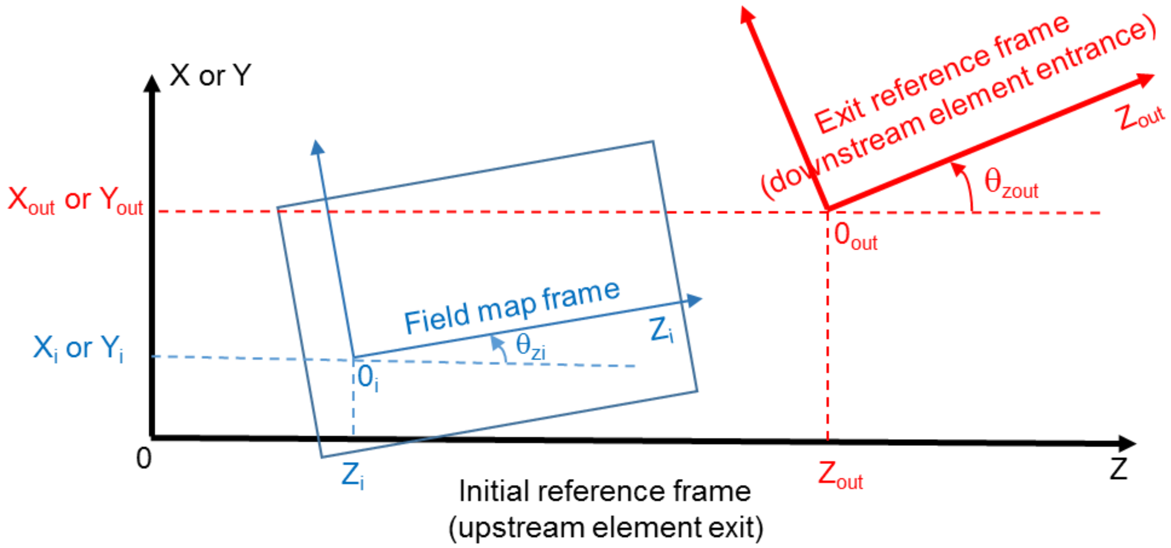
Using: **SUPERPOSE_MAP_OUT 1573.29 170.0 0 0 0 -11.0**
Instead of: **SUPERPOSE_MAP_OUT 1573.29 164.27 0 0 0 -9.59023**



Positioning of field maps

Each individual element, described by a field map, is associated to a FIELD_MAP keyword:
FIELD_MAP ***xxxx L θ R kb ke Ke Ka Filename***

It is possible to place the field map with any shifted position or/and orientation relatively to the reference frame. The principle is to give the coordinates of the field map origin along with the angle of its frame in the reference frame.



Let's (0, X, Y, Z) be the origin and the axes of the initial reference frame, at the exit of the upstream element, and (0_i, X_i, Y_i, Z_i) those of the frame of the field map i. In order to place the field map i as indicated in the figure, use the command SUPERPOSE_MAP before the FIELD_MAP command as following:

SUPERPOSE_MAP Z_i X_i Y_i θ_{Zi} θ_{Xi} θ_{Yi}
Pay attention to the order of parameters!

The lengths are in meter and the angles in degree.
Several field maps can be superimposed in the same space, in any order.
In fact, when the SUPERPOSE_MAP commands are used alone, only the parameter Z_i is effective. For the other parameters to be taken into account, those commands must preceded by the SUPERPOSED_MAP_OUT command. The latter can also be used for defining an exit reference frame that is different from the initial one. This is particularly useful in case one of the field maps is a dipole.
The principle is the same as for the field map frame. In order to place the exit reference frame as indicated in the figure, use the command SUPERPOSE_MAP_OUT as following:
SUPERPOSE_MAP_OUT Z_{out} X_{out} Y_{out} θ_{Zout} θ_{Xout} θ_{Yout}
Notice that this last command, SUPERPOSE_MAP_OUT, must be used ahead all the SUPERPOSE_MAP commands.
To summarize, the positioning of field maps can be done with the typical commands:

```
SUPPERPOSE_MAP_OUT ……….
.
.
.
SUPPERPOSE_MAP …………
FIELD_MAP ………….

SUPPERPOSE_MAP …………..
FIELD_MAP ……….
.
.
.
DRIFT ………….
```

Note that when the SUPERPOSE_MAP_OUT command is used, the reference trajectory is the one linking properly, with the proper deviation, the initial reference frame to the exit one. In graphic representations such as for beam envelope, this reference trajectory is a straight line placed at axis origin. This is what is classically done in accelerator physics.
In order to check if the exit reference frame is correctly positioned as regard to the field map(s) or not, add a drift of enough length at the end of the structure. If it is correct, then the reference trajectory would not present any discontinuity neither in position nor in angle when passing from the field map to the drift.
In order to see the reference trajectory in the field map(s) with its real deviations and positions (and not a straight line), don't use the SUPERPOSE_MAP_OUT command, while keeping the SUPERPOSE_MAP commands. In this case, only the Z_i are taken into account and all the other parameters are ignored. So this is only useful when all the maps are aligned with same X, Y, θ parameters. Then the beam can be made entering the field map(s) with right X, Y, θ parameters by using the Beam Center shifts available at the "Twiss parameters" button of the "Main" tab.
Notice that in all cases, the "Field Map Viewer" will take into account only the Z_i parameters and not the other parameters.

Multipole Field Map

Mnemonic	Parameter	Definition
MULTIPOLE	*Order* *L* *Nstep* *B or E* *R* *Lsol* *Zstep* *Elec*	Multipole order Field map length (mm) Number of step along x & y direction Magnetic field on pole (T) or E (MV/m) Aperture (mm) Physical length (mm) of solenoid (*Order=0*) Number of step for solenoid case (*Order=0*) 0: Magnetic field map, 1:Electrique field map

Attention to the TraceWin gradient definition.

The mutipole element generates a **2D (*x*, *y*) static magnetic field map file** whose steps sizes are: d*x* = d*y* = 2.*R*/*Nstep*. The step along z direction is defined by TraceWin calculation step. The simulation is made in this field map. Be aware that 2.*R*/*Nstep* must be much lower than the beam size.

Order parameter sets the order of the multipole field:

- *Order* = 0: (Special mode) for solenoid field map (Br(r,z) & Bz(r,z)),
- *Order* = 1: for dipole,
- *Order* = 2: for quadrupole,
- *Order* = 3: for sextupole,
- *Order* = 4: for octupole,
- …

L is the field map length along z direction.

Nstep defines the number of steps in the generated field map.

B is the magnet

Commands

Change element parameters:

RFQ:

Lattice commands:

Matching commands:

Change structure frequency

FREQ *f*(Mhz)

FREQ command changes the R.F. frequency of the following structure, the beam frequency is not affected. By default, the RF frequency is the beam frequency defined in the TraceWIN GUI.

Add or change variable (including or not mathematical expressions)

VARIABLE *name value*

Varaible can added or changed at any position in the structure as shown in the example below. Name is case sensitive. Attention in the final structure file (*.dat) written at the end of a run, the variables in the elements will all have been replaced by their values.

```
VARIABLE Gr1 100.0
VARIABLE Gr2 150.0
VARIABLE Apert 35.0

DRIFT 10 10
QUAD 56 Gr1 25
DRIFT 10 10
QUAD 56 Gr1 Apert
DRIFT 10 10
QUAD 56 Gr2 25
DRIFT 10 10
QUAD 56 Gr2 25
DRIFT 10 10
VARIABLE Gr1 120.0
QUAD 56 Gr1 25
END
```

Mathematical expressions containing the usual mathematical operators (+,-,/,*,^,%) and standard mathematical functions can be implemented when at least one variable is used, as in the following examples:
(The following math functions are supported: *abs, acos, asin, atan, atan2, ceil, cos, cosh, exp, floor, ln, log, log10, pow, sin, sinh, sqrt, tan, tanh, fac*)

```
VARIABLE VA1 0.11
VARIABLE VA2 0.22
```

DRIFT VA1 10	**OK**
DRIFT 2*VA1 10	**OK**
DRIFT cos(VA1) 10	**OK**
DRIFT cos(VA1+VA2+0.1) 10	**OK**
DRIFT cos(VA1 +VA2+0.1) 10	**NOT_OK** (*because space chars in the equation string are not allowed*)
DRIFT sqrt(0.2*VA1+0.1) 10	**OK**
DRIFT sqrt(0.2+0.1) 10	**NOT_OK** (*because there is no variable*)
DRIFT sqrt(0.2*VA1+0.1) 10*VAR2	**OK**

An ADJUST command placed on a variable will not change the variable but the result of the expression, so this variable will not be changed on the other locations where it is located (see following example).

VARIABLE VA1 0.11	
ADJUST 1	
DRIFT VA1 10	**drift length=0.11 value will be adjsuted**
DRIFT 2*VA1 10	**drift length will be set to 2*0.11, even after adjustment.**

Define doublet section

DOUBLET_START

DOUBLET_END

When '*Set doublet with the same gradient'* option from "*Main*' page is selected, the quadrupoles of doublet located in the same machine period are set to same gradient. Both command '*DOUBLET_START*' and '*DOUBLET_END*' allows to control the zones where are applied this options.

Begin of lattice

LATTICE *n1 n2*

LATTICE command defines the periodic focusing lattices, ***n1*** is the number of element per **basic lattice**, ***n2*** is the number of lattice per **macro-lattice** (usually 1). The number of element per macro-lattice is then *n1·n2*. The basic lattice is used for set the phase advances according to SET_ADV commands. The macro-lattice is used for the phase advance calculation and corresponding matching. All following elements are considered as part of the next lattices until reaching the commands LATTICE_END or END.

The following elements are not included in lattice element counting:

- DIAG_XXX,
- APERTURE,
- THIN_STEERING.

End of lattice

LATTICE_END

LATTICE_END command ends the periodic focusing lattices.

Structure file end

END

END command ends a structure file (*.dat) description (compulsory).

Set phase advance

SET_ADV kx_{0t} ky_{0t}

The SET_ADV command sets the zero-current horizontal phase advance to kx_{ot} and the zero-current vertical phase advance to ky_{ot}.
By default, without ky_{ot} value, $ky_{ot} = kx_{ot}$.

the phase advance definition.

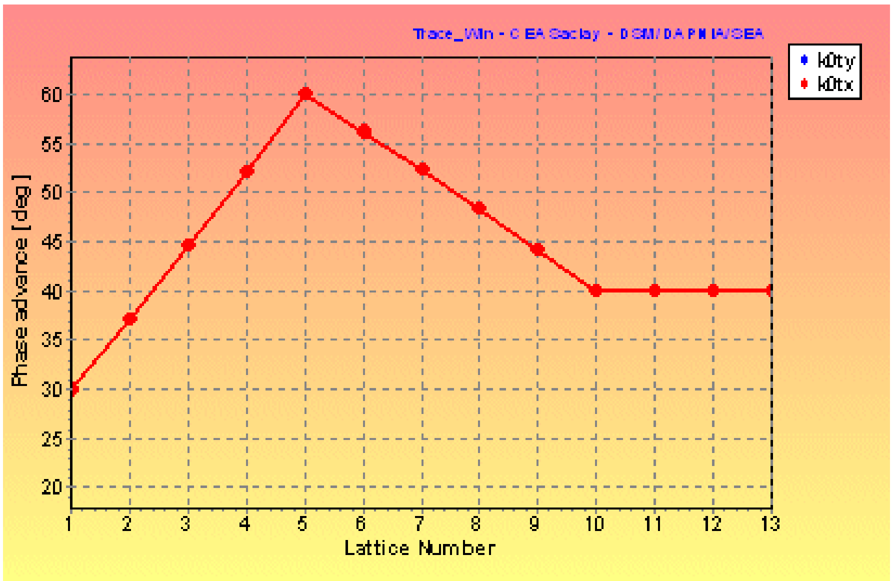
To set a transverse phase advance law

The periodic focusing lattices structure has to be defined by using the command LATTICE and LATTICE_END, in order to indicate the number of lattice and the number of element per lattice.
The zero-current transverse phase advance law can be imposed in two ways:
When the option "***Use phase advance definition***" of "***Main***" page is checked, TraceWin imposes the zero-current transverse phase advance law describe in the Sigma0 file. This file is editable in the "***Main***" page
If the option "***Use phase advance definition***" is unchecked that means you have to use the commands SET_ADV in you data file in order to describe your phase advance law.

Phase advance example

```
;FODO lattices in a DTL tank
;Tank 1
LATTICE 2 1
SET_ADV 30
DTL_CEL 87.5277 28 28 0.00548673 71.3977 -71.6524 74699.4 -50 10 0 0.102751 0.723252 -0.475485 -0.194532
DTL_CEL 87.9518 28 28 0.0080072 -71.6524 71.9632 76721.8 -49.5 10 0 0.103248 0.726628 -0.470582 -0.195991
DTL_CEL 88.3919 28 28 0.0112292 71.9632 -72.2903 79282.7 -49 10 0 0.103764 0.734598 -0.458632 -0.197949
DTL_CEL 88.8487 28 28 0.0144758 -72.2903 72.5931 81864.3 -48.5 10 0 0.104299 0.742159 -0.447189 -0.199371
DTL_CEL 89.3222 28 28 0.0177645 72.5931 -72.9101 84484.9 -48 10 0 0.104853 0.74948 -0.436027 -0.200384
DTL_CEL 89.8133 28 28 0.0212224 -72.9101 73.1977 87251.7 -47.5 10 0 0.105428 0.757488 -0.423767 -0.201319
DTL_CEL 90.3217 28 28 0.0245936 73.1977 -73.4992 89953.5 -47 10 0 0.106024 0.764331 -0.413161 -0.201532
DTL_CEL 90.8475 28 28 0.0280032 -73.4992 73.7703 92694.2 -46.5 10 0 0.10664 0.770931 -0.402823 -0.201301
DTL_CEL 91.3909 28 28 0.0314415 73.7703 -74.0554 95477.4 -46 10 0 0.107276 0.777317 -0.392802 -0.200986
SET_ADV 60
DTL_CEL 91.952 28 28 0.0349097 -74.0554 74.3058 98302.2 -45.5 10 0 0.107934 0.783481 -0.38306 -0.200411
DTL_CEL 92.5298 28 28 0.0379774 74.3058 -74.5691 100888 -45 10 0 0.108611 0.787243 -0.377176 -0.2003
DTL_CEL 93.124 28 28 0.0410704 -74.5691 74.8056 103520 -44.5 10 0 0.109308 0.790922 -0.371402 -0.200107
DTL_CEL 93.7349 28 28 0.0442596 74.8056 -75.056 106233 -44 10 0 0.110024 0.794763 -0.365289 -0.199539
DTL_CEL 94.3626 28 28 0.0473999 -75.056 75.2835 108961 -43.5 10 0 0.11076 0.798277 -0.35973 -0.199184
DTL_CEL 95.0078 28 28 0.0507833 75.2835 -75.5255 111880 -43 10 0 0.111516 0.802715 -0.352626 -0.198381
DTL_CEL 95.6703 28 28 0.0539705 -75.5255 75.7413 114710 -42.5 10 0 0.112293 0.80606 -0.347306 -0.197902
DTL_CEL 96.3499 28 28 0.0571744 75.7413 -75.9704 117592 -42 10 0 0.11309 0.809328 -0.342081 -0.197341
DTL_CEL 97.0469 28 28 0.0603973 -75.9704 76.1784 120527 -41.5 10 0 0.113907 0.812524 -0.336963 -0.196748
DTL_CEL 97.7612 28 28 0.0636044 76.1784 -76.3982 123492 -41 10 0 0.114744 0.815493 -0.332338 -0.196611
SET_ADV 40
DTL_CEL 98.4937 28 28 0.0670821 -76.3982 76.5998 126672 -40.5 10 0 0.115603 0.819426 -0.325941 -0.195469
DTL_CEL 99.244 28 28 0.0703571 76.5998 -76.8124 129777 -40 10 0 0.116483 0.822424 -0.321067 -0.19464
DTL_CEL 100.013 28 28 0.0738459 -76.8124 77.0048 133061 -39.5 10 0 0.117384 0.826103 -0.315053 -0.193443
DTL_CEL 100.8 28 28 0.0771374 77.0048 -77.209 136276 -39 10 0 0.118307 0.828909 -0.31049 -0.192635
DTL_CEL 101.604 28 28 0.0804404 -77.209 77.395 139551 -38.5 10 0 0.11925 0.831652 -0.306016 -0.191796
DTL_CEL 102.428 28 28 0.0839794 77.395 -77.5934 143023 -38 10 0 0.120216 0.835125 -0.30028 -0.190411
DTL_CEL 103.271 28 28 0.0875319 -77.5934 77.7738 146561 -37.5 10 0 0.121204 0.838521 -0.29462 -0.188871
DTL_CEL 104.132 28 28 0.0908581 77.7738 -77.9644 150025 -37 10 0 0.122213 0.841052 -0.290457 -0.187938
LATTICE_END
END
```

"*LATTICE 2 1*" defines the periodic focusing lattices, 2 is the number of element per lattice (2 DTL cell), 1 is the number of lattice per macro-lattice (generally 1). And TraceWin imposes a phase advance linear continuity between each "SET_ADV" command. This linear continuity can be per lattice or per meter according to the option "***Linear phase advance per meter***" of the "***Main***" page. This option is very useful, if you have lattice length discontinuity in order to keep continuity in the phase advance law per meter. TraceWin calculates the quadrupole gradients to obtain the asked zero-current phase advance law, using the lattice transfer matrix.
Warning: it is not always possible to find the required quadrupole gradients.



Matching commands

MATCH_FAM_GRAD *f_n, n, 1/0*
MATCH_FAM_FIELD *f_n, n*
MATCH_FAM_PHASE *f_n, n*
MATCH_FAM_LFOC *f_n, n*
MATCH_FAM_LENGTH *f_n, n*

f_n is the family or section number and *n* is the matching element number. When one of these commands precedes an element, it is used (modified) to match the section *f_n*. For example, if two independent cavities are needed to match the beam at the entrance of a section two commands "*MATCH_FAM_FIELD*" or "*MATCH_FAM_PHASE*" or "*MATCH_FAM_LFOC*", with different matching element numbers *n*, have to be placed before these cavity entries. But if these two cavities must have the same field, they must have the same matching element number *n*. If you need for the matching five different quadrupoles and 2 different cavities, then the last command "*MATCH_FAM…*" must have *n* = 7. All combinations are possible. You can use the number of quadrupole, the number of cavity and the number of quadrupole or cavity coupled, you want. The coupled elements are not necessary consecutive. Two examples are presented below. These commands can also be used to match the beam at the entrance of the linac. The only constraint about the family numbers *f_n* is to have a different number per matching family, you can use for example 5,4,8,1...If you set *n=0* in each comment of a matching, TraceWin sets automatically a different number for each command, but no coupling are possible between elements.
New (10/09/2012): The third parameter, *(0/1)*, of the command "*MATCH_FAM_GRAD*" is used for "*DTL_CEL*" elements to make difference between the first half quadrupole (0 by default) and second one (1).

MATCH_FAM_GRAD: the quadrupole gradient is adjusted.
MATCH_FAM_FIELD: the field is adjusted.
MATCH_FAM_PHASE: the synchronous phase is adjusted.
MATCH_FAM_LFOC: the longitudinal focalization is adjusted by moving the field and the synchronous phase and the energy gain is kept.
MATCH_FAM_LENGTH: the element length is adjusted.

Matching Way

TraceWin is able to match the beam at the entrance of the linac or between the different sections. In these two cases the criterion for a good matching is either keep the longitudinal and transverse phase advances as smooth as possible or have at the input and the output lattice the same Twiss parameters (taking into account the acceleration parameters.
TraceWin is also able to impose Twiss parameters at a position in the linac. In this case you have to insert a SET_TWISS command behind the matching commands.
These three processes are named "*Optimization*", and can be stop by using the menu "*Stop*".

Optimization

To make the optimization of the beam at the entrance of the linac or between the different sections, TraceWin calculates the phase advance on *N* lattices, using the option "***Nbr of phase advance period to smooth***" of "***Matching***" page ". The choice of the criterion (Twiss parameter or smooth phase advances) depends of this option. If the number of lattice is too small (below 6) or if the number of lattice to optimize is set to 0, the Twiss criterion is used. You can watch the optimization process by plotting the phase advance chart, "***Beam***" and starting the "***Synch.***" option in the chart. The results of the optimizations can be found in the Results file. The optimization automatically stops when the criterion reaches "***Max. Number of iteration***" defined in the "***Matching***" page, but it can be stopped before in the menu "***Stop/Optimize***". During a matching some other criterions can be included in order to control the beam size, the beam separation, the beam transverse position or the beam emittance growth...(See matching commands) In this case all the different criterions are added.

Phase advance criterion: $vcr = \frac{1}{NM}\sum_{i=2}^{N-2}\sum_{j=1}^{M}(d^2\sigma_{i,j})$ With $d^2\sigma_{i,j} = \frac{\sigma_{i+1,j}+\sigma_{i-1,j}-2\sigma_{i,j}}{\sigma_{i,j}}$

With *M=2* for DC beam (*x*,*y*) or *M=3* for bunched beam (*x*,*y*,*z*), and N is the number of lattice to optimize. $\sigma_{t,\varphi}$ is in °/m.

See also the phase advance definition and setting

Matching section or family

If the option "***Matching with family & Twiss commands***" of page "***Matching***" is checked, TraceWin changes some quads and cavities strength or element lengths pointed by "*MATCH_FAM…*" commands, in order to match the beam between two sections. The syntax is not always very easy to use because it has been defined in order to be able to represent most of the cases. Some examples are shown below to help you and if you meet some difficulties yet, send an Email with your data file attached to the authors.

Matching example 1

Matching line calculation between a RFQ and a 352 MHz DTL. Two 704MHz buncher cavities and 4 quadrupoles are used in order to match the beam to the structure. A command SET_SIZE_MAX has been include in order to try to reduce the beam size. To help the matching optimization two drift lengths are adjusted by using the command "*MATCH_FAM_LENGTH*".

```
; *****   RFQ-DTL Matching line    ******
DRIFT 1e-05 10
FREQ 704
SET_SIZE_MAX 0.025 25 4.5 4.5 25 1
DRIFT 20 10
```

```
MATCH_FAM_GRAD 1 0
QUAD 56 55.9938 10
MATCH_FAM_LENGTH 1 0
DRIFT 58.85 10
MATCH_FAM_FIELD 1 0
GAP 251000 -90 10 0 0 0 0 0 0 0
DRIFT 58.85 10
MATCH_FAM_GRAD 1 0
QUAD 56 -56.1374 10
MATCH_FAM_LENGTH 1 0
DRIFT 50.0 10
MATCH_FAM_GRAD 1 0
QUAD 56 62.7152 10
DRIFT 58.85 10
MATCH_FAM_FIELD 1 0
GAP 282000 -90 10 0 0 0 0 0 0 0
DRIFT 58.85 10
MATCH_FAM_GRAD 1 0
QUAD 56 -56.5712 10
DRIFT 70 10
FREQ 352
QUAD 28 71.3977 10

; *****************   DTL   *****************
LATTICE 2 1
SET_ADV 46
DTL_CEL 87.5277 28 28 0.00548673 71.3977 -71.6524 74699.4 -50 10 0 0.102751 0.723252 -0.475485 -0.194532
DTL_CEL 87.9518 28 28 0.0080072 -71.6524 71.9632 76721.8 -49.5 10 0 0.103248 0.726628 -0.470582 -0.195991
DTL_CEL 88.3919 28 28 0.0112292 71.9632 -72.2903 79282.7 -49 10 0 0.103764 0.734598 -0.458632 -0.197949
DTL_CEL 88.8487 28 28 0.0144758 -72.2903 72.5931 81864.3 -48.5 10 0 0.104299 0.742159 -0.447189 -0.99371
DTL_CEL 89.3222 28 28 0.0177645 72.5931 -72.9101 84484.9 -48 10 0 0.104853 0.74948 -0.436027 -0.200384
DTL_CEL 89.8133 28 28 0.0212224 -72.9101 73.1977 87251.7 -47.5 10 0 0.105428 0.757488 -0.423767 -0.201319
DTL_CEL 90.3217 28 28 0.0245936 73.1977 -73.4992 89953.5 -47 10 0 0.106024 0.764331 -0.413161 -0.201532
…
..
.
```

The calculation result is the following:
The first line result contains the 4 quadrupole gradients in T/m, the second is the cavity field corrections and the last the drift lengths in mm.

```
Matching_Between_Section_0_to_1
56.3399 -56.1173 62.8806 -56.4843
0.934963 0.991466
0.00834559 0.000887792 0.0294733 0.0874924
```

Matching example 2

Matching line calculation between a two super conducting cavity families by adjusting the synchronous phases of the last cavities of the first family and the first cavities of the second family. Four quadrupoles are also adjusted in order to match the beam.

```
.....
DRIFT 325 100
QUAD 400 10 100
DRIFT 400 100
QUAD 400 -10 100
DRIFT 950 100
NCELLS 1 5 0.6579000 1.11349e+07 -12.331 100 0 0.1801715 0.2934260 -10.0848222 14.7245834 …
DRIFT 475 100
NCELLS 1 5 0.6579000 1.11616e+07 -10.584 100 0 0.1801715 0.2934260 -10.1057623 14.7737516 …
DRIFT 475 100
NCELLS 1 5 0.6579000 1.11873e+07 -8.890 100 0 0.1801715 0.2934260 -10.1259198 14.8211363 …
```

```
DRIFT 625 100
SET_ADV 70
DRIFT 325 100
MATCH_FAM_GRAD 8 1
QUAD 400 10 100
DRIFT 400 100
MATCH_FAM_GRAD 8 2
QUAD 400 -10 100
DRIFT 950 100
MATCH_FAM_PHASE 8 3
NCELLS 1 5 0.6579000 1.12121e+07 -7.248 100 0 0.1801715 0.2934260 -10.1453288 14.8668118 …
DRIFT 475 100
MATCH_FAM_PHASE 8 3
NCELLS 1 5 0.6579000 1.12361e+07 -5.657 100 0 0.1801715 0.2934260 -10.1640222 14.9108503 …
DRIFT 475 100
MATCH_FAM_PHASE 8 3
NCELLS 1 5 0.6579000 1.12593e+07 -4.114 100 0 0.1801715 0.2934260 -10.1820316 14.9533216 …
DRIFT 625 100
LATTICE_END
DRIFT 1187.35 100

; Second superconducting family
LATTICE 13 1
SET_ADV 85
DRIFT 325 100
MATCH_FAM_GRAD 8 4
QUAD 500 10 100
DRIFT 400 100
MATCH_FAM_GRAD 8 5
QUAD 500 -10 100
DRIFT 975 100
MATCH_FAM_PHASE 8 6
NCELLS 1 5 0.8458000 1.14453e+07 -94.435 100 0 0.1560511 0.2270898 -10.0611253 13.2916014 …
DRIFT 525 100
MATCH_FAM_PHASE 8 6
NCELLS 1 5 0.8458000 1.14816e+07 -92.773 100 0 0.1560511 0.2270898 -10.0883587 13.3415267 …
DRIFT 525 100
MATCH_FAM_PHASE 8 6
NCELLS 1 5 0.8458000 1.15177e+07 -91.116 100 0 0.1560511 0.2270898 -10.1153599 13.3910179 …
DRIFT 525 100
MATCH_FAM_PHASE 8 6
NCELLS 1 5 0.8458000 1.15536e+07 -89.464 100 0 0.1560511 0.2270898 -10.1421056 13.4400346 …
DRIFT 650 100
DRIFT 325 100
QUAD 500 10 100
DRIFT 400 100
QUAD 500 -10 100
DRIFT 975 100
…
```

Matching example 3

TraceWin sets Twiss parameter to the entrance of a RFQ. To impose Twiss parameter in your structure by adjusting elements you have to use the same "*MATCH_FAM…*" commands, but a SET_TWISS command has been put in front of the element where you want to impose Twiss parameters at its output. The first parameter of SET_TWISS and "*MATCH_FAM…*" commands is the family or the section number.

```
..
…
DRIFT 50.1 100
MATCH_FAM_FIELD 1 0
GAP 154335 -90 100 0 0
DRIFT 50.1 100
MATCH_FAM_LENGTH 1 0
```

```
DRIFT 60 100
MATCH_FAM_GRAD 1 0
QUAD 60 16.9089 100
MATCH_FAM_LENGTH 1 0
DRIFT 40 100
DRIFT 20 100
MATCH_FAM_GRAD 1 0
QUAD 60 -16.9089 100
DRIFT 50.1 100
MATCH_FAM_FIELD 1 0
GAP 154335 -90 100 0 0
DRIFT 50.1 100
DRIFT 20 100
MATCH_FAM_GRAD 1 0
QUAD 60 16.9089 100
MATCH_FAM_LENGTH 1 0
DRIFT 40 100
DRIFT 20 100
APERTURE 2.7 2.9 1
FREQ 352.21
DRIFT 0.00001 10
RFQ_CELL 120000 4.96362 0 1 109.954 -60 3
RFQ_CELL 120000 4.96362 0.114946 1.13305 27.4884 -60 4
SET_TWISS  1 -1.5440 0.2462 0.7924 0.1898 -0.0754 0.9523 0 0 0 0 0 0
RFQ_CELL 120000 4.96362 0.114946 1.13305 27.4884 -60 2
RFQ_CELL 120000 4.96362 0.115356 1.13354 27.5261 -60 -2
RFQ_CELL 120000 4.96362 0.115799 1.13406 27.567 -59.9802 2
RFQ_CELL 120000 4.96362 0.116304 1.13465 27.611 -59.9208 -2
RFQ_CELL 120000 4.96362 0.116874 1.13532 27.6552 -59.822 2
RFQ_CELL 120000 4.96362 0.117509 1.13609 27.6995 -59.6843 -2
RFQ_CELL 120000 4.96362 0.118213 1.13692 27.7441 -59.5081 2
RFQ_CELL 120000 4.96362 0.118987 1.13786 27.789 -59.2943 -2
RFQ_CELL 120000 4.96362 0.119833 1.13888 27.8343 -59.0435 2
RFQ_CELL 120000 4.96362 0.120755 1.13999 27.8799 -58.7569 –2
…
..
.
```

Matching example 4

4 quadrupoles and 2 cavities are used to match the beam between section 1 and 2. The longitudinal matching is done by a cavity field modification. This is the simplest case.

```
LATTICE 7 1                          «section 1»
DRIFT 100 100
MATCH_FAM_GRAD 1 1
QUAD 150 10 100                      «first quadrupole used for matching »
DRIFT 178.56 100
MATCH_FAM_GRAD 1 2
QUAD 150 -10 100                     «second quadrupole used for matching»
DRIFT 488.64 100
MATCH_FAM_FIELD 1 3
GAP 2.94577e+06 -30.5122 100         «first cavity used for matching, the field is adjusted»
DRIFT 722.08 100

LATTICE 9 1                          «section 2»
DRIFT 150 100
MATCH_FAM_GRAD 1 4
QUAD 350 10 100                      «third quadrupole used for matching»
DRIFT 100 100
MATCH_FAM_GRAD 1 5
QUAD 350 -10 100                     «fourth quadrupole used for matching»
DRIFT 935.126 100
```

```
MATCH_FAM_FIELD 1 6
GAP 3.04084e+06 -30.5122 100             «second cavity used for matching, the field is  adjusted»
DRIFT 1385.81 100
GAP 3.07847e+06 -30.5122 100
DRIFT 729.063 100
```

Matching example 5

4 quadrupoles and 4 cavities are used for the matching between section 1 and 2. The longitudinal matching is done by a cavity phase adjustment. Here, the cavities are coupled 2 by 2.

```
LATTICE 8 1                              «section 1»
…
DRIFT 100 100
MATCH_FAM_GRAD 1 1
QUAD 150 10 100                          «first quadrupole used for matching »
DRIFT 178.56 100
MATCH_FAM_GRAD 1 2
QUAD 150 -10 100                         «second quadrupole used for matching»
DRIFT 488.64 100
MATCH_FAM_PHASE 1 3
GAP 2.94577e+06 -30.5122 100             «first cavity used for matching, the phase is adjusted. n=3 ».
MATCH_FAM_PHASE 1 3
GAP 2.94577e+06 -30.5122 100             «second cavity used for matching. It will conserve the same phase as the
first cavity. n=3 »
DRIFT 722.08 100
LATTICE 10 1                             «section 2»
DRIFT 150 100
MATCH_FAM_GRAD 1 4
QUAD 350 10 100                          «third quadrupole used for matching»
DRIFT 100 100
MATCH_FAM_GRAD 1 5
QUAD 350 -10 100                         «fourth quadrupole used for matching»
DRIFT 935.126 100
MATCH_FAM_PHASE 1 6
GAP 3.04084e+06 -30.5122 100             «third cavity used for matching, the phase is adjusted. n=6 ».
DRIFT 1385.81 100
MATCH_FAM_PHASE 1 6
GAP 3.04084e+06 -30.5122 100             «fourth cavity used for matching, It will conserve the same phase as the
third cavity. n=6 »
DRIFT 729.063 100
```

Set Twiss parameters

SET_TWISS f_n, α_x, β_x*(mm/mrad)*, α_y, β_y*(mm/mrad)*, α_z, β_z(mm/mrad),k_{ax},k_{bx},k_{ay},k_{by},k_{az},k_{bz}

The forces of elements pointed with the matching commands "*MATCH_FAM_XXX*" are adjusted to impose the Twiss parameters given by the "*SET_TWISS*" command at the output element following the command. More than one command "*SET_TWISS*" can be used in the same optimization. In this case, use the same f_n parameter.

Twiss criterion:

$$vcr = \frac{1}{2M}\sum_{j=1}^{M}\left(\left(\sqrt{\frac{R_j+\sqrt{R_j^2-4}}{2}}\right)-1\right)^2$$

With $R_j = \beta_j \gamma_{j0} + \beta_{j0} \gamma_j - 2\alpha_j \alpha_{j0}$

fn is the section or family number.
βj, γj, αj being the beam Twiss parameters of the space phase *xx', yy', zz'*.
The 6 following parameters "*k*" are optional, set one to "*1*" allows to not taking account of the corresponding Twiss parameters
Write "SET_TWISS *fn"* corresponds to write "SET_TWISS *fn 0 0 0 0 0 0 1 1 1 1 1 1".* That disables the SET_TWISS command and allows different other optimisations by inserting another commands like for example "SET_SIZE".

Set beam centroid position

SET_POSITION *k*(*m*$^{-1}$), *x(mm), x'(mrad), y(mm), y'(mrad)*

x, x', y, y' are the centroid beam positions imposed at the point where this command appears, *k* is used in the criterion calculation.

Set position criterion: $$vcr = \frac{k}{2} \cdot \left[(x - x_0)^2 + (y - y_0)^2 + (x' - x'_0)^2 + (y' - y'_0)^2 \right]$$

With *x(mm), x'(mrad), y(mm), y'(mrad)* being the beam centroid transverse positions at the place where the command "*SET_POSITION*" appears and *xo (mm), xo'(mrad), yo (mm), yo'(mrad)* being the beam imposed positions.

Set achromatic line

SET_ACHROMAT *k f1 f2 plane*

Located after a deviation for instance this command allows to make achromatic the preceding line. *k* is used to balance the criterion calculation
(*f1=1* means set achromatic position).
(*f2=1* means set achromatic angle).

If (*plane=0*) : X or Y or both planes acccoding to bend direction are concerned (default)
If (*plane=1*) : only X plan is concerned
If (*plane=2*) : only Y plan is concerned
If (*plane=3*) : both planes are concerned

Example:

```
DRIFT 300 100
START_ACHROMAT
EDGE 22.5 500 100 0.45 2.8 50 0
```

```
BEND 45 500 0 50 0
EDGE 22.5 500 100 0.45 2.8 50 0
DRIFT 200 100
MATCH_FAM_GRAD 4
QUAD 200 2
DRIFT 200 100
EDGE 22.5 500 100 0.45 2.8 50 0
BEND 45 500 0 50 0
EDGE 22.5 500 100 0.45 2.8 50 0
DRIFT 107 100
SET_ACHROMAT 1.0 1 1
DRIFT 100 100
SET_TWISS 4
DRIFT 100 100
END
```

The command SET_TWISS is always obligatory to allow the matching of the gradrupole. And the option "***Matching with family & Twiss commands***" of page "***Matching***" has to be checked

Set maximum beam max

SET_SIZE_MAX *k, N, x(mm), y(mm), φ(°)/z(mm), k2 (0/1)*

x, y, φ are the imposed beam size max in the N elements following this command, *N*, *k* and *k2* is used in the criterion calculation.

- Set size max criterion: $$vcr = \frac{k}{M}\cdot\left(\exp\left(-\left(\frac{x_0}{x}\right)^4\right)+\exp\left(-\left(\frac{y_0}{y}\right)^4\right)+\exp\left(-\left(\frac{\varphi_0}{\varphi}\right)^4\right)\right)$$

With *x(mm), y(mm), φ(°)* being the beam maximum sizes in the *N* elements following the command "SET_SIZE_MAX", and *x₀(mm), y₀(mm), φ₀(°)* being the beam imposed sizes. If *φ* is lower than 0 the longitudinal size used is *z(mm).* If one of these last parameters is set to 0, no optimization is done on this size and *M* is reduced by one. If *k2=0,* the transverses sizes are is calculated without taking account the beam centroid position. If one of the parameters *x, y* or *φ* is equal to 0, no optimization is done on this size and *M* is reduced by one.
The sizes are the effective beam sizes (rms*sqrt(5) for bunched beam or rms*sqrt(4) for CW beam)

Set beam size

SET_SIZE *k, x(mm), y(mm), φ(°)/z(mm), k2 (0/1)*

x, y, φ are the imposed beam size in the output element following this command, *N*, *k* is used in the criterion calculation. If *k2=0*, the transverses sizes are calculated without taking account the beam centroid position.

Set size criterion: $$vcr = \frac{k}{M} \cdot \left[\left(\frac{x - x_0}{x_0} \right)^2 + \left(\frac{y - y_0}{y_0} \right)^2 + \left(\frac{\varphi - \varphi_0}{\varphi_0} \right)^2 \right]$$

With *x(mm), y(mm), φ(°)* being the beam sizes in the output elements following the command "SET_SIZE", and *x₀(mm), y₀(mm),), φ₀(°)* being the beam imposed sizes. If *φ* is lower than 0 the longitudinal size used is *z(mm).* If one of these last parameters is equal to 0, no optimization is done on this size and *M* is reduced by one.
The sizes are the effective beam sizes (rms*sqrt(5) for bunched beam or rms*sqrt(4) for CW beam)

Set beam separation

SET_SEPARATION *k, Sx, Sy*

Sx₀, Sy₀ are the centroid beam positions divided by the imposed beam size max at the point where this command appears, *k* is used in the criterion calculation.

Set separation criterion: $vcr3 = k \cdot \left[(Sx_0 - Sx)^2 + (Sy_0 - Sy)^2 \right]$

Minimize emittance growth

MIN_EMIT_GROW *k, N, ex, ey, ez, f*

k and *N* are used in the criterion calculation.

ex, ey, ez and *f* are optional: if *ex* equal to 1 the criterion doesn't take into account ε_x beam emittance. That is the same way for *ey* and *ez*. f parameter define if the criterion calculation is relative or not (see formula below).

The emittance growth criterion:

- If (f=1) $vcr = k \cdot \sum_N \left[\left(\frac{\varepsilon_x - \varepsilon_{x0}}{\varepsilon_{x0}} \right)^2 + \left(\frac{\varepsilon_y - \varepsilon_{y0}}{\varepsilon_{y0}} \right)^2 + \left(\frac{\varepsilon_z - \varepsilon_{z0}}{\varepsilon_{z0}} \right)^2 \right]$

- If (f=0) $vcr = k \cdot \left(|\varepsilon_x| + |\varepsilon_y| + |\varepsilon_z| \right)$ at the command position

With ε_{xo}, ε_{yo}, ε_{zo} being the beam emittances where this command appears, and ε_x, ε_y, ε_z being the beam emittances after *N* elements.

Steerer

STEERER *Bx(T) By (T) Bmax 0 coef$_1$(m) coef$_2$(m)*

STEERER *Ex(V/m) Ey (V/m) Emax(V/m) 1*

Its a command not a element (Corresponding element: *THIN_STERRING*). The magnetic steerer is inserted in the element (magnetic or electric quadrupole, solenoid and field map) placed just after the instruction “Steerer” (keeping the same length). parameters *coef$_1$* and *coef$_2$* allow to introduce non-linear forces.

$$x' = x' - \frac{q \cdot \Delta s \cdot B_y c}{mc^2 \beta\gamma} \quad \text{And} \quad y' = y' + \frac{q \cdot \Delta s \cdot B_x c}{mc^2 \beta\gamma}$$

$$x' = x' + \frac{\Delta s \cdot E_x}{E\rho} \quad \text{And} \quad y' = y' + \frac{\cdot \Delta s \cdot E_y}{E\rho}$$

Where *x'* and *y'* being respectively the horizontal and vertical beam centroid slope.
Used in diagnostic optimization (with *ADJUST_STERRER*), *Bmax* and *Emax* is the maximum limit for *Bx*, *By* and *Ex*, *Ey*, if it’s greater than zero.

FIELD_MAP case: in field map, kick angle is applied on the full field map length.

The both parameters *coef$_1$* and *coef$_2$* allow to introduce non-linear forces in the steerer deviation.
coef$_1$ and *coef$_2$* units are meter.

- If *coef$_1$* is not equal to 0 (**SPIRAL2** type steerer):

$$B_x(x,y) = B_{x0} \cdot (1 + \frac{y^2}{coef_1{}^2}) + 2 \cdot B_{y0} \frac{xy}{coef_1{}^2}$$
$$B_y(x,y) = B_{y0} \cdot (1 + \frac{x^2}{coef_1{}^2}) + 2 \cdot B_{x0} \frac{xy}{coef_1{}^2}$$

- If *coef$_2$* is not equal to 0 (**ESS** type steerer):

$$B_x(x,y) = B_{x0} \cdot (1 + coef_2 \cdot (x^2 - y^2)) + B_{y0} \cdot 2 \cdot coef_2 \cdot xy$$
$$B_y(x,y) = B_{y0} \cdot (1 + coef_2 \cdot (x^2 - y^2)) - B_{x0} \cdot 2 \cdot coef_2 \cdot xy$$

With: $Coef_2(m^{-2}) = \frac{G(sextu)}{B_{x0}} or \frac{G(sextu)}{B_{y0}}$ see TraceWin gradient definition.

Chopper

CHOPPER *N, U(V), D(mm), C(mm), p(0/1)*

The chopper is inserted in the *N* elements placed just after the instruction "Chopper" (keeping the same length). *U* is the voltage between axis and plates and *C* is the chopper transverse position. ±D is the distance between axis and plates

If (p=0) $x' = x' + \frac{q \cdot \Delta s \cdot U}{mc^2 \beta^2 \gamma \cdot D}$ and if (p=1) $y' = y' + \frac{q \cdot \Delta s \cdot U}{mc^2 \beta^2 \gamma \cdot D}$

Where x' and y' are respectively the horizontal and vertical beam centroid slope.

Input beam errors

ERROR_BEAM_STAT *r(0/1/2), dx(mm), dy(mm), dφ(°), dxp(mrad), dyp(mrad), de(MeV),dEx(%), dEy(%), dEz(%), mx(%), my(%), mz(%), dib(mA), αxx'_min, αxx'_max, βxx'_min(mm/mrad), βxx'_max(mm/mrad), αyy'_min, αyy'_max, βyy'_min(mm/mrad), βyy'_max(mm/mrad), αzdp_min, αzdp_max, βzdp_min(mm/mrad), βzdp_max(mm/mrad)*

ERROR_BEAM_DYN

Five kind of error can be set:
Beam displacement: (*dx*, *dy*, *dρ*, *dxp*, *dyp*, *de*) The beam input position is not centered.
Emittance growth: (*dEx*, *dEy*, *dEz*) The input beam emittance is increased by a percentage. For CW beam *dEz* error is applied on the energy spread.
Beam mismatch: (*mx*, *my*, *mz*) The input beam is mismatched by a percentage. A 20 % mismatch in x plane means α_x and β_x are multiplied by 1.2^2.
Beam Current error: (*dib*) Allows to study the effect of the input beam current variation (*Ibeam=Ibeam0+dib*).
Twiss parameter range: (*α_min, α_max, β_min, β_max*) The input beam is randomly select in the twiss parameter ranges. The 4 parameters, *α_min, α_max, β_min, β_max*, must be different of 0 and mismatch parameter *m* must be equal to 0.
If (*r = 0*), *α_beam = α_min+(α_max- α_min)*k*
(*k* varing from 0 to 1 according to the "*Nbr o f step*" parameter. If "*Nbr_step=4*", k={0, 0.25 ,0.50 ,0.75 ,1)

This command concerns only the input beam. The error distribution depends on the *r* parameter:

- *r = 0*, the errors are constant and equal to each value of the command line.
- *r = 1*, the errors are uniformly distributed (+/-); each value of the command line is the maximum range error.
- *r = 2*, the errors are Gaussian distribute; each value of the command line is rms value of the distribution.

NCP CPL STAT DYN meaning

See also ERROR_STAT_FILE & ERROR_DYN_FILE.

Quadrupole errors

ERROR_QUAD_NCPL_STAT *N r(0/1/2/4/5), dx(mm), dy(mm), φ_x(°), φ_y(°), φ_z(°), dG(%), dz(mm) , dG3(%), dG4(%), dG5(%), dG6(%), Nb*

ERROR_QUAD_NCPL_DYN

ERROR_QUAD_CPL_STAT

ERROR_QUAD_CPL_DYN

ERROR_QUAD_NCPL_STAT_FILE *file*

See also ERROR_STAT_FILE & ERROR_DYN_FILE.

New since version 2.16.1.2:
If ERROR_CAV_CPL_XXX and ERROT_QUAD_CPL_XXX commands are located at the same position with the same N parameters then error displacements and rotations applied on cavities and magnet elements will be the same. In this case, error amplitudes are by default defined by ERROR_CAV_CPL_XX.

New since version 2.16.1.0:
Nb parameter allows to limit the number of elements where error are applied, selecting them randomly. For example if you want to move only one quadrupole randomly chosen, this command will do it: *ERROR_QUAD_NCPL_STAT 20000 0 0.1 0 0 0 0 0 0 0 0 0 0 0 1*
Nb is possible only for _NCPL_ error types.

dx and dy being respectively the horizontal and vertical magnetic element displacement. *dz* being the longitudinal shift. φ_x ,φ_y, φ_z being respectively the quadrupole rotation around *x*, *y*, *z*-axis and *dG* being the gradient amplitude error. *dG3, dG4, dG5, dG6* correspond respectively to the sextupole, octupole, decapole, dodecapole gradient errors and concern only QUAD and QUAD_ELE elements. These errors are applied in the *N* elements following this command, excepted, if a new error command appears. The error distribution depends on the *r* parameter:

- *r = 0*, the errors are constant and equal to each value of the command line.
- *r = 1*, the errors are uniformly distributed (+/-); each value of the command line is the maximum range error.
- *r = 2*, the errors are Gaussian distribute; each value of the command line is rms value of the distribution.
- *r = 4*, the errors randomly distributed between 2 values, +*M*axErr *or* -*M*axErr. *MaxErr* is the error amplitude set in the command line.
- *r = 5*, the errors randomly distributed between 2 values, 0 *or M*axErr. *MaxErr* is the error amplitude set in the command line.

Remark about *dG3, dG4, dG5, dG6 errors:* These errors are only applied on QUAD and QUAD_ELE elements. When the multipole gradients in the QUAD element are set to zero, then the respective gradient errors are introduced in terms of quadrupole main field *B2*, e.g. $G3 = B2\left(\frac{0.01 \cdot dG3}{R^2}\right)$, $G4 = B2\left(\frac{0.01 \cdot dG4}{R^3}\right)$, etc.. With *B2* as magnetic field on pole.

The fifth command uses a file to set errors. This file is located by default in structure file (*.*dat*) path. It contains lines like the following syntax (*be careful to Z unit*):

Z$_0$(m) *r(0/1/2) dx(mm) dy(mm) φ_x(°) φ_y(°) φ_z(°) dG(%) dz(mm), dG3(%) dG4(%) dG5(%) dG6(%)*
Z$_1$(m) *r(0/1/2) dx(mm) dy(mm) φ_x(°) φ_y(°) φ_z(°) dG(%) dz(mm), dG3(%) dG4(%) dG5(%) dG6(%)*
…
Z$_n$(m) *r(0/1/2) dx(mm) dy(mm) φ_x(°) φ_y(°) φ_z(°) dG(%) dz(mm) dG3(%) dG4(%) dG5(%) dG6(%)*

An interpolation is performed to define errors applied to an element at a given position. It's not the case for '*r*' parameter which is defined without interpolation.

A new command "*ERROR_QUAD_NCPL_STAT_FILE*" without file, put in the *.dat file, stop the action of the preceding similar command.

Be careful: Errors defined by *ERROR_QUAD_NCPL_STAT_FILE* are added to errors coming from *ERROR_QUAD_NCPL_STAT* commands.

This error set affects:
- Quadrupole
- Solenoid
- Quadrupole of DTL
- Field map if defined as static magnetic field

NCP CPL STAT DYN meaning

Set phase error to synchronous phase

PHASE_ERROR_TO_SYNC_PHASE *flag*

Set to 1 this command allows to apply phase errors directlly to synchronous phase of the cavity instraed to RF phase which is the default case. Set to 0 cancels this feature for all cavities localized after this command.
This command makes errors are canceled and a new synchronous phase setting of the linac are generated.

Phase error amplitudes are defined by ERROR_CAV_NCPL_STAT command, but in this case, errors are transformed in new synchronous phase setting of the linac.

Cavity errors

ERROR_CAV_NCPL_STAT *N r(0 /±1/2/±3/4/5), dx(mm), dy(mm), φ_x(°), φ_y(°), E(%), φ(°), dz(mm), Nb*

ERROR_CAV_NCPL_DYN

ERROR_CAV_CPL_STAT

ERROR_CAV_CPL_DYN

ERROR_CAV_NCPL_STAT_FILE *file*

See also ERROR_STAT_FILE & ERROR_DYN_FILE.

New since version 2.16.1.2:
If ERROR_CAV_CPL_XXX and ERROT_QUAD_CPL_XXX commands are located at the same position with the same N parameters then error displacements and rotations applied on cavities and magnet elements will be the same. In this case, error amplitudes are by default defined by ERROR_CAV_CPL_XX.

New since version 2.16.1.0:
Nb parameter allows to limit the number of elements where error are applied, selecting them randomly. For example if you want to cut off RF field of a cavity randomly chosen, this command will do it: *ERROR_CAV_NCPL_STAT 20000 0 0 0 0 0 -100 0 0* ***1***
Nb is possible only for _NCPL_ error types.

dx and *dy* being respectively the horizontal and vertical electric element displacement. *dz* being the longitudinal shift, φ_x, φ_y being respectively the cavity rotation around x, y-axis. *E* being the field amplitude error. φ being the field phase error. These errors are applied in the *N* elements following this command, excepted, if a new error command appears. The error distribution depends on the *r* parameter:

- *r = 0*, the errors are constant and equal to each value of the command line.
- *r = 1*, the errors are uniformly distributed (±); each value of the command line is the maximum range error.
- *r = -1*, the errors are Gaussian distribute; each value of the command line is rms value of the distribution.
- *r = 4*, the errors randomly distributed between 2 values, +*M*axErr *or* -*M*axErr. *MaxErr* is error amplitude set in the command line.
- *r = 5*, the errors randomly distributed between 2 values, 0 *or M*axErr. *MaxErr* is the error amplitude set in the command line.

Specail feature for *r=2* or *r=±3*
According to *r* parameter:

- *r = 2*, the errors are constant and equal to each value of the command line.
- *r = 3*, the errors are uniformly distributed (+/-); each value of the command line is the maximum range error.
- *r = -3*, the errors are Gaussian distribute; each value of the command line is rms value of the distribution.

And the command syntax becomes:
ERROR_CAV_XXX_XXX *N r(2/±3), dx(mm), dy(mm),* φ_x*(°),* φ_y*(°), E(%), k*φ*(deg/%), dz(mm)*
The phase error applied on the cavities is $\boldsymbol{\varphi_{err} = k\varphi * E_{rr}}$
In case of coupled error (_CPL_) the field phase and the field amplitude error sign **are reversed** each new cavity.

The fifth command uses a file to set errors. This file is located by default in structure file (**.dat*) path. It contains lines like the following syntax (*be careful to Z unit*):

Be careful: Errors defined by *ERROR_CAV_NCPL_STAT_FILE* are added to errors coming from *ERROR_CAV_NCPL_STAT* commands.

Z₀(m) *r(0/±1/2/±3) dx(mm) dy(mm)* φ_x*(°)* φ_y*(°) E(%)* φ*(°) dz(mm)*

***Z_1*(m)** *r(0/±1/2/±3) dx(mm) dy(mm) φ_x(°) φ_y(°) E(%) φ(°) dz(mm)*
…

***Z_n*(m)** *r(0/±1/2/±3) dx(mm) dy(mm) φ_x(°) φ_y(°) E(%) φ(°) dz(mm)*

An interpolation is performed to define errors applied to an element at a given position. It's not the case for '*r*' parameter which is defined without interpolation.

A new command "*ERROR_CAV_NCPL_STAT_FILE*" without file, put in the *.dat file, stop the action of the preceding similar command.

By default phase errors are applied to RF phase of concerned cavities. The specific command PHASE_ERROR_TO_SYNC_PHASE allows to applied error directly to synchronous phase, but in this case errors are transformed in new synchronous phase setting of the linac.

This error set affects:
- Bunched cavity
- Cavity multi-gap
- DTL (Warning the DTL are only concerns by the field errors)
- Sinus cavity
- Field map if not defined as static magnetic field

NCP CPL STAT DYN meaning

Bend errors

ERROR_BEND_NCPL_STAT *N, r(0/1/2/4/5), dx(mm), dy(mm), φ_x(°), φ_y(°), φ_z(°), dg(%), dz(mm)*

ERROR_BEND_NCPL_DYN

ERROR_BEND_CPL_STAT

ERROR_BEND_CPL_DYN

See also ERROR_STAT_FILE & ERROR_DYN_FILE.

dx and dy being respectively the horizontal and vertical magnetic element displacement. *dz* being the longitudinal shift. φ_x ,φ_y, φ_z being respectively the bend rotation around *x*, *y*, *z*-axis and *dg* being the magnetic field amplitude error. These errors are applied in the *N* elements following this command, excepted, if a new error command appears. The error distribution depends on the *r* parameter:

- *r = 0*, the errors are constant and equal to each value of the command line.
- *r = 1*, the errors are uniformly distributed (±); each value of the command line is the maximum range error.
- *r = 2*, the errors are Gaussian distribute; each value of the command line is rms value of the distribution.
- *r = 4*, the errors randomly distributed between 2 possible values, +*Max*Err *or* -*Max*Err. *MaxErr* is error amplitude set in the command line.

- *r = 5*, the errors randomly distributed between 2 values, 0 *or M*axErr. *MaxErr* is the error amplitude set in the command line.

This error set affects only magnet bend and edge
Valid only if the BEND element is not cut in several parts.

Warning, the "ERROR_BEND_CPL_X" commands concern only the magnetic field amplitude errors, all other errors (shift, rotation,…) are not couple.

NCP CPL STAT DYN meaning

See detail about bend error treatment.

Adjust commands

ADJUST *N, v, n, min, max, first_step*

Red parameters are optional.
This command has to be associated with a Diagnostic elements number *N*. *v*, an integer, is the v-th variable to be adjust in the next element list of variable. As an exemple if v=2 and the next element is a quad then the gradient will be adjusted. n (if it's different from zero) allows to link ADJUST commands each other, therefore, two ADJUST commands with the same *n* will give the same value to the *v* variable point. *v* lower than zero allows to replace both command "ADJUST_DPHASE" and "ADJUST_DFIELD" when *v* points to phase or field variable.

New in version 2.2.0.16 : *first_step* parameter has been added in order to choose the first step of optimization.

New in version 2.2.1.9: The *n* parameter allow to link two variables: Now if you set *n* parameter to opposite sign you'll get opposite variation for the linked variables. For example two drifts coupled with (*n*=1 & *n*=-1) will keep the same total sum length.

5 consecutives ADJUST command can be set.
(See Adjust and diagnostic examples)

ADJUST_STEERER *N, max, first_step*

ADJUST_STEERER_BX *N, max, first_step*

ADJUST_STEERER_BY *N, max, first_step*

These commands have to be associated with Diagnostic elements number *N* and should be placed before a STEERER command. It allows to adjust the horizontal and/or vertical magnetic field steering.

Particular case of steerer adjustment: When the number of steerers corresponds to the number of BMP, no optimization is performed and the resolution of the system is directly made by a matrix inversion

New in version 2.2.0.20: Input beam parameters can be adjusted in order to fit with a set of diagnostic values at different positions of a line
(Example adjusted beam emittances and Twiss parameters at the input of a simulated line in order to obtain the beam sizes measured at different positions with real beam)

ADJUST_BEAM_TWISS *N, AlpX_flag, betX_flag, AlpY_flag, BetY_flag, AlpZ_flag, betZ_flag*
If flag is set to 1 the selected Twiss parameter will be adjusted, if you want to have alpX=AlpY set alpX_flag=1 and alpY_flag=2, same way for Bet_flag)

ADJUST_BEAM_EMIT *N, Ex_flag, Ey_flag, Ez_flag*
If flag is set to 1 the selected emittance will be adjusted, if you want to have Ex=Ey set Ex_flag=1 and Ey_flag=2)

ADJUST_BEAM_CENTROID *N, X_flag, Xp_flag, Y_flag, Yp_flag, Z_flag, Zp_flag*
If flag is set to 1 the selected beam centroid parameter will be adjusted, if you want to have X=Y set X_flag=1 and Y_flag=2, same way for Xp_flag)

ADJUST_BEAM_CURRENT *N, I_flag*

All “*ADJUST_BEAM_XXX*” commands must be located in front of the first element. Some elements can be also adjusted in the same matching process.

Adjust and diagnostic examples1

Beam alignment from a RFQ to a DTL. Two ADJUST commands are associated with two STEERER and with two Diagnostic elements.

```
DRIFT 0.00001 6.5
DRIFT 56.0 6.5
ADJUST_STEERER 1                    «Adjust the steerer inside the following quadrupole »
STEERER -.01 0.01
ERROR_QUAD_NCPL_STAT 1 0 0.1 0.1 0 0 0 0 «Include 0.1mm misalignment and 0.3° rotation errors in the quad »
QUAD 56.0 55.0 6.5
DRIFT 2.15 6.5
DRIFT 58.85 6.5
GAP 180000 -90 6.5
DRIFT 58.85 6.5
ADJUST_STEERER 1                    «Adjust the steerer inside the following quadrupole »
STEERER .01 0.01
QUAD 56.0 -55.0 6.5
DRIFT 82.9 6.5
DIAG_POSITION 1 0 0                 « Position monitor with x and y position imposed to 0 »
DRIFT 20 6.5
QUAD 56.0 55.0 6.5
DRIFT 58.85 6.5
GAP 210000 -90 6.5
DRIFT 38.85 6.5
QUAD 56.0 -55.0 6.5
DRIFT 54.3 6.5
DIAG_POSITION 1 0 0                 « Position monitor with x and y position imposed to 0 »
DRIFT 20.0 6.5
…
```

You can see the result in the result file. This result is not used in a statistical error study where an adjustment is calculated for each random error distribution.

```
Diagnostic_1
-0.0661744 -0.0663664 0.0554  0.0458
```

Adjust and diagnostic examples2

Triplet adjusting in order to match a beam size

```
ADJUST 1 2 1                « Quadrupole 1 and 3 are linked, the point variable ‘2’ is the gradient »
QUAD 152 2.14 52
DRIFT 119 52
ADJUST 1 1 0 200 300        « Here, the quadrupole length is adjusted, but the length will keep between (200 & 300) »
QUAD 281 -2.13 52
DRIFT 119 52
ADJUST 1 2 1
QUAD 152 2.0 52
DRIFT 3190 100
APERTURE 2 2 0
DIAG_SIZE 1 2 2
DRIFT 3590 100
DIAG_SIZE 1 4 4
DRIFT 100 100
…
```

Adjust and diagnostic examples3

Set achromatic line part from element 3 to achromatic diagnostic position

```
DRIFT 30 100
DRIFT 30 100
DRIFT 30 100
DRIFT 30 100
EDGE 22.5 500 100 0.45 2.8 50 0
BEND 45 500 0 50 0
EDGE 22.5 500 100 0.45 2.8 50 0
DRIFT 200 100
ADJUST 11 2 1
QUAD 100 -200 20
DRIFT 200 100
ADJUST 11 2 1
QUAD 100 200 20
DRIFT 200 100
ADJUST 11 2 1
QUAD 100 -200 20
DRIFT 200 100
EDGE 22.5 500 100 0.45 2.8 50 0
BEND 45 500 0 50 0
EDGE 22.5 500 100 0.45 2.8 50 0
DRIFT 107 100
DIAG_ACHROMAT 11 3 1 1
DRIFT 100 100
DRIFT 100 100
End
```

Adjust and diagnostic examples4

Find the input beam Twiss parameters and emittances given the rms sizes measured at the end of the line.

```
; Proton @20 MeV, 10 mA
ADJUST_BEAM_CURRENT 99 1
ADJUST_BEAM_EMIT 99 1 2 0 ; Ex=Ey
ADJUST_BEAM_TWISS 99 1 1 1 1 0 0
DRIFT 0 100
DRIFT 100 100
QUAD 100 -15.18 20
```

```
DRIFT 200 100
QUAD 100 15.18 20
DRIFT 200 100
QUAD 100 -15.18 20
DRIFT 200 100
DIAG_SIZE 99 2.2 1.6
DRIFT 200 100
DIAG_SIZE 99 2.2 1.6
DRIFT 0 100
END
```

RFQ errors

ERROR_RFQ_CEL_NCPL_STAT *N, r(0/1/2), dR(mm), d (mm), E(%),φ(°), TEpe(mm), TEpa(mm), DEpe(mm), DEpa(mm), DELong(mm) TSVerti(mm), TSHori(mm), DSVerti(mm), DSHori(mm), DSLong(m)*

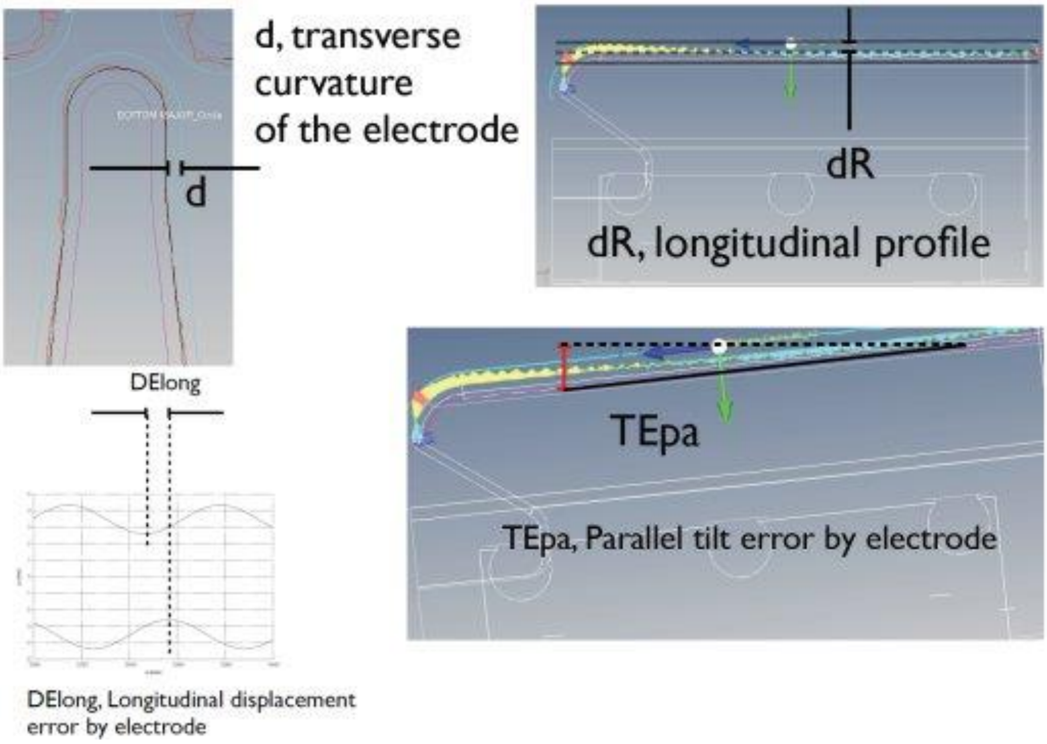
ERROR_RFQ_CEL _NCPL_DYN *(not implemented yet)*

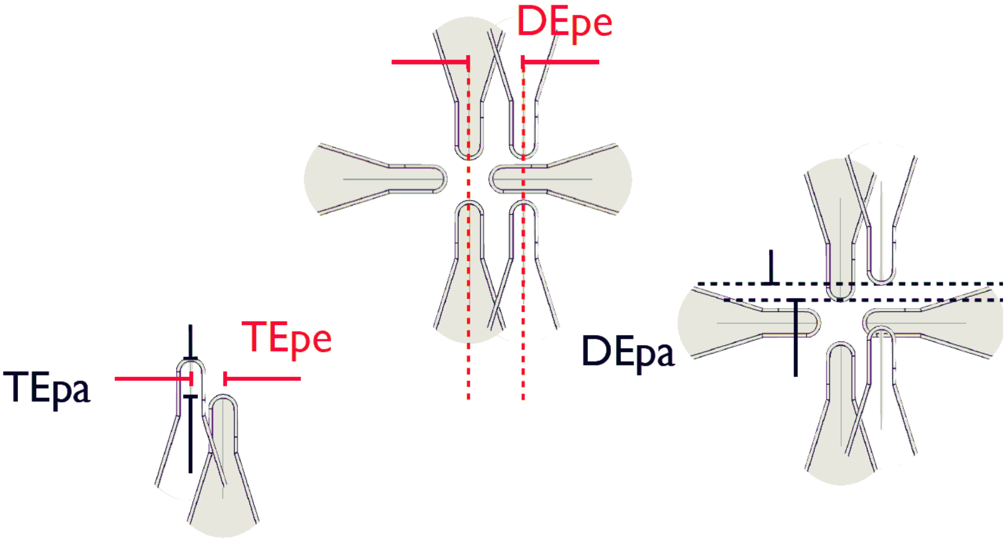
dR is the error for the longitudinal profile, d is the error for the transverse curvature of the electrode, E is the voltage amplitude error. *φ* being the field phase error (uneffective).
For the transverse plane, TEpe is the perpendicular tilt error by electrode, TEpa is the parallel tilt error by electrode, DEpe is the perpendicular displacement error by electrode, DEpa is the parallel displacement error by electrode, DELong is the a longitudinal displacement error by electrode.
For each segment (block of four electrodes defined with the RFQ_GAP command), TSVerti is the vertical tilt error, TSHori is the horizontal tilt error, DSVerti is the vertical displacement error, DSHori is the horizontal displacement error and DSLong is the longitudinal displacement error.
These errors are applied in the N elements following this command, excepted, if a new error command appears. The error distribution depends on the *r* parameter:

- *r = 0*, the errors are constant and equal to each value of the command line (only envelope mode).
- *r = 1*, the errors are uniformly distributed (±); each value of the command line is the maximum range error.
- *r = 2*, the errors are Gaussian distribute; each value of the command line is rms value of the distribution.

RFQ is a special element in TraceWin and all these errors are no effect in an envelope calculation except the phase and filed errors.

NCP CPL STAT DYN meaning


d, transverse curvature of the electrode
d
dR
dR, longitudinal profile
DElong
DElong, Longitudinal displacement error by electrode
TEpa
TEpa, Parallel tilt error by electrode



TEpa, Parallel tilt error by electrode

TEpe, Perpendicular tilt error by electrode

DEpa, Parallel displacement error by electrode

DEpe, Perpendicular displacement error by electrode

(*) Images provided by Ibon Bustinduy, ESS Bilbao.

Plot distribution

PLOT_DST *file_out*

PLOT_DST_LOST *file_out*

If *file_out* parameter is empty, indicate to the multiparticle code to store beam distribution (in the *.plt file) at the end of the next element, otherwise save the beam distribution in *file_out* as a dst file (.*dst* extension has to be omitted).

"*PLOT_DST_LOST*" command allows to include in the dst file, the lost particles.

Gas pressure

GAS *C*, *N*, *P*

Only used in multiparticle simulation, this command set the gas pressure parameter until a new command *GAS*
C is the cross section at 1 MeV (m²), *N* is the atomic number and *P* is the pressure (hPa).
You have to select '*gas stripping*' or '*gas scattering*" in the '*Multiparticle*' page.

For gas stripping, the actual cross section used in the simulation is: $\sigma(m^2) = C \cdot E^{\frac{-3}{4}}$, E being the beam energy in MeV (*N* is useless).

For gas scattering calculations, only *N* and *P* are used (*C* is useless).

Several "gas" commands can be set simultaneously (Max. 9).

Minimize enveloppe variation

MIN_ENV_VARIATION *k*, *N*

Minimize the variation of the maximum and minimum transverse beam envelope in magnetic element. For bunched beam, the phase-spread variation is also minimizing in the accelerating elements. The parameter *k* is a weighting factor for the contribution of this constraint to the overall penalty function and *N* is the number of element positions over which the constraint applies. The first affected position is the one immediately following the command. Only magnetic and accelerating elements falling within the specified range are affected.

Minimize phase variation

MIN_PHASE_VARIATION *k*, *N*, $\Delta\theta_{max}$

During a matching procedure it is often useful to limit the maximum cavity (synchronous) phase amplitude variation. The parameter *k* is a weighting factor for the contribution of this constraint to the overall penalty function, θ is the max allowed angle variation in degrees and *N* is the number of element positions over which the constraint applies. The first affected position is the one immediately following the command. Only cavities falling within the specified range are affected.

$$vcr = k \cdot \exp\left[\left(\frac{\Delta\phi_{\max}}{\Delta\phi_{sync}}\right)^4\right]$$

NCPL CPL STAT DYN meaning

NCPL: means: No Coupled. The errors are individually applied on each element
CPL: means: Coupled. The errors are coupled on the *N* elements. In other words, a rotation error corresponds to an *N* elements block rotation.
STAT: means: Static. The effect of these errors can be detected and corrected with appropriate diagnostic and correctors. For example, beam position measurement coupled with steerers can compensate the quadrupole or cavities misalignments. Correction strategy should be known to be able to estimate their impact on beam dynamics
DYN: means: Dynamic. The effect of these errors cannot be measured and then corrected. Fortunately, they have usually lower amplitude than static errors. They are, for example, the vibrations of the elements or the RF field control errors (in phase or amplitude). The knowledge of the correction scheme is not needed to study their statistic impact. They are responsible of orbit oscillations around the corrected orbit (this notion of orbit is also extended in the longitudinal motion).

Set RFQ vane geometry

RFQ_GEOM *type N dz*

Put this command just before the first RFQ cell.
According to *type* parameter:

If *type* = 0: Toutatis generates vane geometry file with 50steps/cell. It’s the default case.

If *type* = 1: Toutatis reads vane geometry file
Example: RFQ_GEOM 1 c:\my_project\rfq\My_rfq.vane

If *type* = 2: and *N* = 0: Toutatis generates vane geometry file with a cell step = *dz*
Example: **RFQ_GEOM 2 0 0.001** for a 1 mm step

If *type* = 2: and *N* ≠ 0: Toutatis generates vane geometry file with *N step*/cell
Example: RFQ_GEOM 2 20

If *type* = 3: For statistical error study case: (for X cases)
Example: RFQ_GEOM **3** c:\my_project\rfq\My_rfq.vane
Toutatis, for each case, will kook for a RFQ geometry vane file called:
c:\my_project\rfq\My_rfq000001.vane for first run
c:\my_project\rfq\My_rfq000002.vane for second one
…
..
.
c:\my_project\rfq\My_rfq00000X.vane for last one

Set RFQ electrode type

FOUR_RODS

This option generates electrode profile with varying transverse curvature. By default the transverse curvature is proportional to *Ro* (four vane type). Put this command just before the first RFQ cell element.

TWOTERMS

This command allows to generate a longitudinal profile (way to modulate) which is governed by the classical 2 terms potential. In case this option is used, only *m* and *a* are taken into account, *Ro* is recomputed. The sinus modulation is the default in case nothing is specified. Put this command just before the first RFQ cell.

Set RFQ coupling gap

RFQ_GAP *Lp(m), Lg(m), Sl(m), St(m)*

This command defines in a RFQ structure a resonant coupling gap. The position of the command in the data file doesn't matter, if you respect 2 rules: place it before a RFQ cell element, and if you need more than one coupling gap you must avoid to put the new command before the same RFQ cell element. *Lp* is the longitudinal position of the center of the gap. *Lg* is gap width. *Sl* is the half-ellipse size in the beam direction and *St* is the half-ellipse height in the perpendicular beam direction.

Set RFQ front-end cell gaps

RFQ_GAP_RMS_FFS *GapRMS (mm), GapFFS (mm)*

This command defined in a RFQ structure, before the first RFQ_CELL element, allows to redefine the front-end gap of the first (GapRMS) and the last (GapFFS) RFQ_CELL (type 3). In Toutatis, these gaps are by defaut defined as ¼ of the front-end cell length. Set GapRMS or GapFFS parameter to *0* to use the Toutatis default values.

Set of field map

SET_OF_MAP

Put this command just before a FIELD_MAP element. This command provides a way to do an error study using a list of field maps.

Used only during statistical error study case: (for *X* linac)

For each linac, the reference field map will be replaced by a new one named like in the following example:

SET_OF_MAP
LME-Q11 : FIELD_MAP 70 480 0 40 9.174 0 0 0 qpole480_lme

qpole480_lme000001 *for first run*
qpole480_lme000002 *for second one*
…
..
.
qpole480_lme00000X *for last run*

Set of error from file

ERROR_STAT_FILE ***static_error_file.txt***

ERROR_DYN_FILE ***dynamic_error_file.txt***

Put this command wherever in the structure file (*.dat), whatever the position, allow to replace static or/and dynamic errors defined by usual error commands (ERROR_QUAD_.XX, ERROR_CAV_XX…), by the values set in the file. Both definition systems cannot coexist, defined errors in files will always dominate. The syntax of the file is identical to the ouput file, "*Error_Datas.txt*", containing the applied errors, defined here.

Example:
ERROR_STAT_FILE D:/temp/temp5/Error_Datas_static.txt
ERROR_DYN_FILE D:/temp/temp5/Error_Datas_dynamic.txt

For the special case of a statistical error study: (for *X* linac)
For each linac, the error file will be replaced by a new one named like in the following example:

D:/temp/temp5/Error_Datas_static000001.txt *for first run*
D:/temp/temp5/Error_Datas_dynamic000001.txt *for first run*
D:/temp/temp5/Error_Datas_static000002.txt *for second run*
D:/temp/temp5/Error_Datas_dynamic000002.txt *for second run*
…
..
D:/temp/temp5/Error_Datas_static00000X.txt *for last run*
D:/temp/temp5/Error_Datas_dynamic00000X.txt *forlast run*

Set error ratios

ERROR_SET_RATIO

‘*Nstep*’ parameter in “*Error*” page allows to apply progressively the errors from 0% to 100% of the error amplitudes defined by “*ERROR_XX*” commands. For example “*Nstep=4*” will perform 4 error studies applying 0% 25% 50% 75% 100% of the maximal error amplitudes.

The command “*ERROR_SET_RATIO*” allows to user to choose the ratio values.
Example: “*ERROR_SET_RATIO 0 0.25 0.75 1.5*” will apply 0%, 25% 75% and 150% of the error amplitudes.

To read a particle file

The particle file from the page “***Main***” allows to define the input particle file, but if you want to force the envelope or multiparticle calculation to load a new particle file characteristics (Twiss parameters, current, emittances, centroid) at a given position in the linac, you have to include the command below followed by the full name of the particle file.

READ_DST *full_particle_file_name*

Warning: This command is read at its position and not like other commands at the end of the following element.

To read a multiparticle output file

If you choose to run multiparticle code (Partran or Toutatis), you can avoid to run a linac part or even all the linac which has been already computed. Then, you have to indicate to TraceWin, which part doesn’t have to be run, by insert at the beginning of this part the command “*READ_OUT”* followed by the full name of the Partran or Toutatis output file

READ_OUT *full_output_multiparticle_file_name*

This command is necessary associated with a READ_DST command (See examples below).

Warning: This command is read at its position and not like other commands at the end of the following element.

The following examples are allowed:
(Where calculation directory is: “*D:\temp\temp3\*”)

LBET + RFQ + MEBT: here to avoid to perform the LEBT already calculated, the LEBT is first of all simulated alone and the 3 following file are copied:

- “partran1.out” is copied to “lebt.out”
- “Density_PAR.dat” is copied to “lebt.dat”
- “part_dtl1.dst” is copied to “lebt.dst”

No element before READ_OUT command
No element except, RFQ_CEL, afer READ_DST command

```
; LEBT
READ_OUT d:\temp\temp3\lebt.out
DRIFT 0 40
DRIFT 10 40
SOLENOID 100 1 40
DRIFT 10 40
SOLENOID 100 1 40
DRIFT 10 40
DRIFT 0 40
READ_DST d:\temp\temp3\lebt.dst

;RFQ
RFQ_CELL 70000 9.94396 0 1 66.9319 -90 3 7.5 1 0 0 0 0
RFQ_CELL 70000 9.94396 6.69068e-05 1.00028 11.1553 -90 4 7.5 1 0 0 0 0
LATTICE 2 1
RFQ_CELL 70000 9.94244 0.000125255 1.00052 11.1553 -90 2 7.5 1 0 0 0 0
RFQ_CELL 70000 9.93848 0.000206589 1.00086 11.1553 -90 -2 7.5 1 0 0 0 0
RFQ_CELL 70000 9.93291 0.000269428 1.00112 11.1553 -90 2 7.5 1 0 0 0 0
RFQ_CELL 70000 9.92632 0.000314328 1.0013 11.1553 -90 -2 7.5 1 0 0 0 0
…
…
RFQ_CELL 70000 8.90751 0.0188169 1.06427 11.1553 -88.3866 -2 7.5 1 0 0 0 0
RFQ_CELL 70000 8.89561 0.0193 1.0658 11.1553 -90.8234 2 7.5 1 0 0 0 0
RFQ_CELL 70000 8.88381 0.0198278 1.06748 11.1553 -89.3208 -2 7.5 1 0 0 0 0
RFQ_CELL 70000 8.87215 0.0203424 1.06911 11.1553 -90.7149 2 7.5 1 0 0 0 0

; MEBT
DRIFT 1 40
QUAD 100 1 40
DRIFT 100 40
QUAD 100 -1 40
DRIFT 100 40
DRIFT 0 40
END
```

LBET + RFQ + MEBT: here to avoid to perform the RFQ already calculated, the RFQ is first of all simulated alone and the 3 following file are copied:

- “partran1.out” is copied to “rfq.out”
- “Density_PAR.dat” is copied to “rfq.dat”
- “part_dtl1.dst” is copied to “rfq.dst”

No element after READ_OUT command, except RFQ_CEL
No element before READ_DST command, except, RFQ_CEL

If you have some matching commands in line2, it’s strongly recommended to add at least 5 null drifts just after READ_DST command.

```
; LEBT
DRIFT 0 40
DRIFT 10 40
SOLENOID 100 1 40
DRIFT 10 40
SOLENOID 100 1 40
DRIFT 10 40
DRIFT 0 40

;RFQ
READ_OUT d:\temp\temp3\lebt.out
RFQ_CELL 70000 9.94396 0 1 66.9319 -90 3 7.5 1 0 0 0 0
RFQ_CELL 70000 9.94396 6.69068e-05 1.00028 11.1553 -90 4 7.5 1 0 0 0 0
LATTICE 2 1
RFQ_CELL 70000 9.94244 0.000125255 1.00052 11.1553 -90 2 7.5 1 0 0 0 0
RFQ_CELL 70000 9.93848 0.000206589 1.00086 11.1553 -90 -2 7.5 1 0 0 0 0
RFQ_CELL 70000 9.93291 0.000269428 1.00112 11.1553 -90 2 7.5 1 0 0 0 0
RFQ_CELL 70000 9.92632 0.000314328 1.0013 11.1553 -90 -2 7.5 1 0 0 0 0
…
…
RFQ_CELL 70000 8.90751 0.0188169 1.06427 11.1553 -88.3866 -2 7.5 1 0 0 0 0
RFQ_CELL 70000 8.89561 0.0193 1.0658 11.1553 -90.8234 2 7.5 1 0 0 0 0
RFQ_CELL 70000 8.88381 0.0198278 1.06748 11.1553 -89.3208 -2 7.5 1 0 0 0 0
RFQ_CELL 70000 8.87215 0.0203424 1.06911 11.1553 -90.7149 2 7.5 1 0 0 0 0
READ_DST d:\temp\temp3\lebt.dst

; MEBT
DRIFT 1 40
QUAD 100 1 40
DRIFT 100 40
QUAD 100 -1 40
DRIFT 100 40
DRIFT 0 40
END

--------------------------------------------------------------------
```

Line1 + Line2 : here to avoid to perform the Line1 already calculated, the Line1 is first of all simulated alone and the 3 following file are copied:

- “partran1.out” is copied to “line1.out”
- “Density_PAR.dat” is copied to “line1.dat”
- “part_dtl1.dst” is copied to “line1.dst”

No element before READ_OUT command.

If you have some matching commands in line2, it’s strongly recommended to add at least 5 null drifts just after READ_DST command.

```
; Line1
READ_OUT d:\temp\temp3\line1.out
DRIFT 0 40
DRIFT 10 40
SOLENOID 100 1 40
DRIFT 10 40
SOLENOID 100 1 40
DRIFT 10 40
DRIFT 0 40
READ_DST d:\temp\temp3\line1.dst

; Line2
DRIFT 1 40
QUAD 100 1 40
DRIFT 100 40
QUAD 100 -1 40
DRIFT 100 40
DRIFT 0 40
END
```

Set marker

MARKER *marker_name*

Change Energy and Phase limit

W_P_LIMIT *energy_limit(MeV) phase_limit(°)*

Works only in tracking mode
If (*energy_limit* > 0 *or phase_limit* > 0) then
  If | *W –Synchonous energy* | > *energy_limit* particle is set as lost
  If | *P –Synchronous phase* | > *phase_limit* particle is set as lost

If (*energy_limit* < 0 *or phase_limit* < 0) then
  If | *W –beam energy centroid* | > | *energy_limit* | particle is set as lost
  If | *P –beam phase centroid* | > | *phase_limit* | particle is set as lost

Change beam parameters

CHANGE_BEAM *q* ΔW_{gE} ΔW_{fE} ΔW_{gM} ΔW_{fM} Δz *auto_flag(0/1)*

q : Change the particle charge state.
ΔW_{gE} *(eV): shift the linac reference energy* ($W_g=W_g+\Delta W_g$)*in envelope mode*
ΔW_{fE} *(eV): shift the beam energy* ($W_f=W_f+\Delta W_f$) *in envelope mode*

ΔW_{gM} *(eV): shift the linac reference energy* ($W_g=W_g+\Delta W_g$) *in tracking mode*
ΔW_{fM} *(eV): shift the beam energy* ($W_f=W_f+\Delta W_f$) *in tracking mode*

Δz*(mm): allows to shift the changing point inside a FIELD_MAP element*

If *auto_flag* is set to ‘*1*’ linac referecene energy is automatically set to beam energy

(This command could be extended according to futur needs)

TOUTATIS using in TraceWin

The “*CHANGE_BEAM*” command is mainly used to make transition betwen TraceWin and Toutatis because both codes don’t use exactly the same way to calculate the output reference and beam energies. Mainly because Toutatis is a ‘*t*’ code and TraceWin is a ‘*z*’ code. The command “CHANGE_BEAM” has been implemented to cancel these differences and make more compatible both codes. You just have to follow the procedure below to define the right command parameters.

In input “inp” file of Toutatis code:

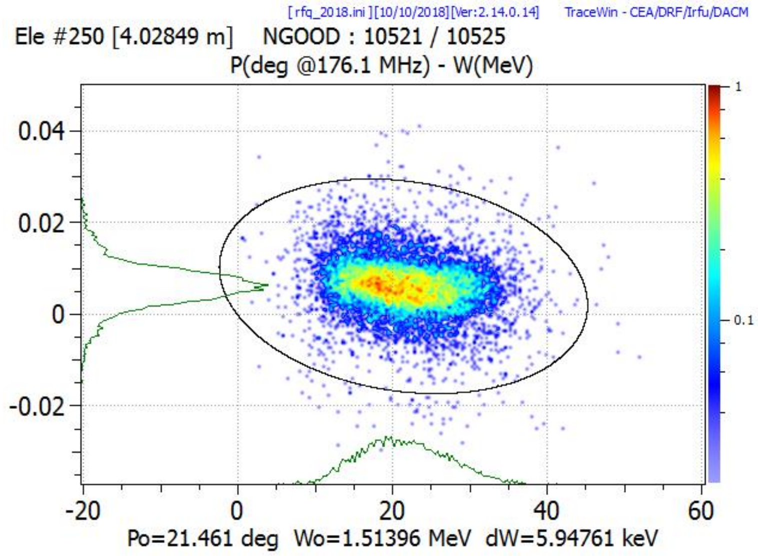
- Remove « DatFlag » in the input « inp » file of Toutatis in order to generate input data file “rfq.dat” for TraceWin code.
- Set “theAcceleratedFlag 1”

- Set “theAccuracyFlag 1”
- Run Toutatis
-

In TraceWin project:

- Use “*rfq.dat*” file generated by Toutatis as structure file in TraceWin project
- Check TraceWin toutatis options in TraceWin (“*Multiparticle*” tab-sheet and “*Toutatis options*”):
  - UnCheck “*Coase mesh*”
  - Check “*Exclude not accelerated particles*”
  - Check “*Number of step per period*” is equal to *NStep* parameter of “*inp*” file

- Put the same particle number, beam current and input energy parameters used in “*inp*” file in TraceWin project.
- In tab-sheet “*Matching*” select “*Calculate input matched beam*”.
- In “*Multiparticle*” tab-sheet select “*Launch Tracking*” for “*Partran*” and “*Toutatis*”.
- Run project
- In “*Charts*” tab-sheet have a look on the output phase space distribution for Tracking (Partran). You should have something like following:



- Put at the end of RFQ structure the command line

CHANGE_BEAM q 0 0 **<u>5947.61</u>** 0 0

5947.61 eV corresponds to the difference between reference and output beam energies from Toutatis.

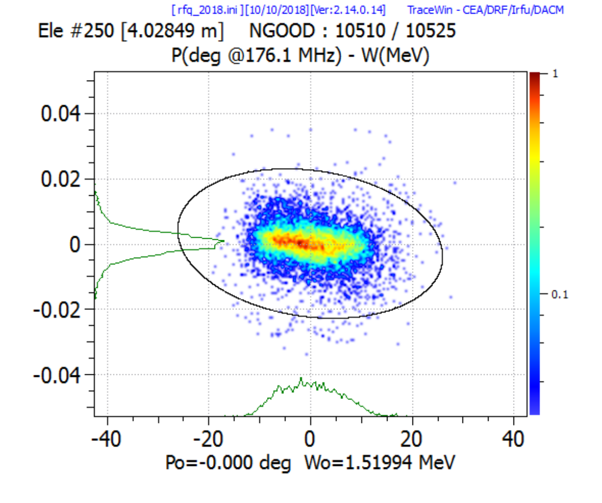
21.461° corresponds to the difference between reference and output beam phase from Toutatis. This phase shift has to be set in the two last RFQ_CEL *Dp* parameters (see bellow)

Set “q” to your particle charge state, 1 for proton.

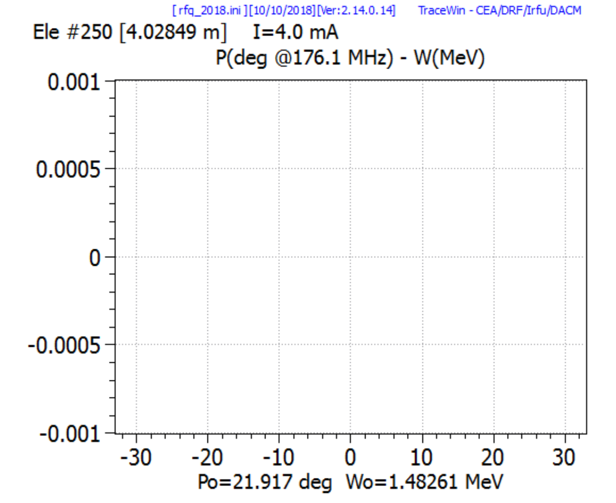
```
..
...
RFQ_CELL 79495.8 8.19952 0.582636  1.99734 64.9099 -34.2808 2 7.5 -0.236878 5.47119
RFQ_CELL 79495.8 8.19952 0.582986  1.99734 65.621 -34.2356 -2 7.5 -0.0365966 5.47119
RFQ_CELL 79495.8 8.19952 0.583322  1.99734 66.325 -34.1928 2 7.5 0.149151 5.47119
```

```
LATTICE_END
RFQ_CELL 79495.8 8.19952 -0 1.99734  67.0223 -34.1521 -4 7.5 -21.461 5.47119
RFQ_CELL 79495.8 8.19952 0 1  40 -90 -3 7.5 -21.461 5.47119
CHANGE_BEAM 1 0 0 5947.61 0 0 0 0
DRIFT 0.00001 6
DRIFT 0.00001 6
DRIFT 0.00001 6
END
```

- Run another time the project and have a look on the output phase space distribution for Tracking and envelope. You should have something like following:



Output beam distribution in Tracking (Partran)



Output phase space in envelope

Output shift energy between envelope and tracking results is:
1.51994 MeV – 1.48261 MeV = 37330 eV

- Change the command to: **CHANGE_BEAM q 37330 37330 5947.61 0 0**

Command **CHANGE_BEAM** is correctly set to compensate output energy differences between Toutatis and TraceWin codes.

Shift

SHIFT *dx(mm), dy (mm)*

Move the following element.

Change transverse beam centroid

SHIFT_BEAM *dx(mm), dy (mm), dx'(mrad), dy' (mrad)*

Change transverse beam centroid values before the following element.

PARTRAN step calculation

PARTRAN_STEP *step1*, *step2*

Step1 is the new step of calculation per metre; *step2* is the new step of space-charge calculation per meter until reaching a new "*PARTRAN_STEP*" command. The default *step1* and *step2* value is put in the "***Multiparticle***" page, see "***Partran step of calcul***".
These two step concern only DRIFT and FIELD_MAP elements, all other elements are treated in 2 steps.

Magnetic or electric static field

FIELD *B_x(T), B_y (T), B_z (T) E_x (V/m) E_y (V/m) E_z (V/m)*

Add to the following elements a magnetic or electric force until a new command "*FIELD*" canceling the preceding. In PARTRAN simulation, only the field command of the first element is considering.

$$x' = x' - \frac{q \cdot \Delta s \cdot B_y c}{mc^2 \beta\gamma} \qquad y' = y' + \frac{q \cdot \Delta s \cdot B_x c}{mc^2 \beta\gamma}$$

$$x' = x' + \frac{q \cdot \Delta s \cdot E_x}{mc^2 \beta^2 \gamma} \qquad .\, y' = y' + \frac{q \cdot \Delta s \cdot E_y}{mc^2 \beta^2 \gamma}. \qquad \gamma \;= \gamma + \frac{qE_z}{mc^2}$$

Where *x'* , *y'* and g being respectively the horizontal and vertical beam centroid slope and the the reduced energy

Set beam phase advance

SET_BEAM_PHASE_ADV *k, N, φ_x(°), φ_y(°), φ_z(°)*

φ_x, φ_y, φ_z, are the imposed beam phase advance in space charge during *N* elements from command. *k* is used in the criterion calculation. The behaviour of this command is unusual because it applies to the entrance of the element where it is placed. So for example if it is localized at element#1 and N is 7, the phase advance is calculated from the input of element#1 to the output of element#7.

Set beam adv criterion: $vcr = \frac{k}{M} \cdot \left[\left(\frac{\varphi_x - \varphi_{x0}}{\varphi_{x0}} \right)^2 + \left(\frac{\varphi_y - \varphi_{y0}}{\varphi_{y0}} \right)^2 + \left(\frac{\varphi_z - \varphi_{z0}}{\varphi_{z0}} \right)^2 \right]$

With $\varphi_x(°)$, $\varphi_y(°)$, $\varphi_z(°)$ being the beam phase advances in the output elements following the command “SET_BEAM_PHASE_ADV”, and $\varphi_{x0}(°)$, $\varphi_{y0}(°)$, $\varphi_{z0}(°)$, being the imposed beam phase advances. If one of these last parameters is equal to 0, no optimization is done on this phase and *M* is reduced by one.

Set beam energy and phase

SET_BEAM_E0_P0 *k, ΔE, Δφ, ke, kp*

ΔE and *Δφ,* are the imposed delta beam energy and phase.

Set beam energy and phase criterion: $vcr = k \cdot \sqrt{k_e\left(\Delta\varphi - \Delta\varphi_0\right)^2 + k_p\left(\Delta E - \Delta E_0\right)^2}$

With *ΔE0* being the difference between the beam energy and the linac energy (close to 0 in the ideal case without error). And *Δφ0* being the difference between the beam phase and the linac phase (close to 0 in the ideal case without error)
ke, *kp* allow to take into account or no one of both.

Set beam energy

SET_BEAM_ENERGY *k*, E_i

E_i is the imposed beam energy (MeV).

Set beam energy criterion: $vcr = k \cdot (E - E_i)^2$

With *E* being the the beam energy.

Set synchronous phase

SET_SYNC_PHASE

The phase given in an RF element preceded by the SET_SYNC_PHASE command is the synchronous phase of the generatrix particle, rather than the absolute phase (by default). It applies to following elements:

FIELD_MAP
RFQ_CEL

CAVSIN
NCELLS

Minimize field variation

MIN_FIELD_VARIATION *k*, *N*, ΔE_{max}

During a matching procedure it is often useful to limit the maximum field amplitude variation. The parameter *k* is a weighting factor for the contribution of this constraint to the overall penalty function, θ is the max allowed angle variation in degrees and *N* is the number of element positions over which the constraint applies. The first affected position is the one immediately following the command. Only cavities falling within the specified range are affected.

$$vcr = k \cdot \exp\left[\left(\frac{\Delta E_{\max}}{\Delta E}\right)^4\right]$$

Duplicate elements

REPEAT_ELE *k*, *n*

Allows to duplicate the following *n* elements *k* times. Caution: The associated commands are also repeated except: '*LATTICE*', '*LATTICE_END*' and '*SET_ADV*' one. A '*REPEAT_ELE'* command cannot include another one.

```
;Example: 100 times a period.
DRIFT 5 30
DRIFT 5 30
REPEAT_ELE 100 4
LATTICE 4 1
SET_ADV 20
QUAD 100 1530 0
DRIFT 100 30
QUAD 100 -15 30 0
DRIFT 100 30
LATTICE_END
DRIFT 5 30
DRIFT 5 30
END
```

```
;Example : Repeat 10 times a quadrupole
DRIFT 5 30
REPEAT_ELE 10 1
QUAD 10 15 30 0
DRIFT 5 30
END
```

```
;Example : Divide in 10 steps a dipole with Edge and fringe-field.
EDGE 20 600 40 0 0 25 0
BEND -90 600 0 20 0
EDGE 20 600 40 0 0 25 0
```

Become

```
EDGE 20 600 40 0 0 25 0
BEND -9 600 0 20 0
```

```
EDGE 0 600 40 1e-12 1e-12 25 0
REPEAT_ELE 8 3
EDGE 0 600 40 1e-12 1e-12 25 0
BEND -9 600 0 20 0
EDGE 0 600 40 1e-12 1e-12 25 0
EDGE 0 600 40 1e-12 1e-12 25 0
BEND -9 600 0 20 0
EDGE 20 600 40 0 0 25 0
```

Move diagnostics in field map element

SHIFT_IN_FIELD_MAP *dz*

Using this command in front of diagnostics elements allows overlapping with the following FIELD_MAP. Dz must be greater than 0 and several diagnostics can be used. See following example where 2 diagnostics are localised into 2 supperposed field maps.

```
DRIFT 100 100
SHIFT_IN_FIELD_MAP 200
DIA1 : DIAG_SIZE 2
SHIFT_IN_FIELD_MAP 360
DIA2 : DIAG_SIZE 5
SUPERPOSE_MAP 0
LME-Q11 : FIELD_MAP 70 480 0 40 9.174 0 0 0 qpole480_lme_25_01_07b
SUPERPOSE_MAP 350
LME-Q11 : FIELD_MAP 70 480 0 40 -9.174 0 0 0 qpole480_lme_25_01_07b
DRIFT 100 100
end
```

Superpose field maps

SUPERPOSE_MAP Z_0 X_0 Y_0 θ_{Z0} θ_{X0} θ_{Y0}

A FIELD_MAP element already allows to superpose 4 field map types (electrostatic, electrodynamic, magnetic or electric field map). With the "*SUPERPOSE_MAP*" command, different FIELD_MAP elements can be superposed at different postions Z_0 of the reference trajectory.

The optionnals parameter, X_0 Y_0 θ_{Z0} θ_{X0} θ_{Y0} are required when the following field map curves the reference trajectory (See Field map with curved reference trajectory). Another command is also needed to inform code to output frame.

If you want to use field map aperture or current in case of superposed field map, you have to include a empty field map element a the position 0 including aperture and/or current field map.

```
;Example: quadrupole inside a solenoid.
DRIFT 5 30
SUPERPOSE_MAP 400
FIELD_MAP  70 100 0 42 -0.3 0 0 0 qpole_field_map_file
SUPERPOSE_MAP 0
FIELD_MAP  70 1000 0 100 -1.3 0 0 0 solenoid_fiedl_map_file
DRIFT 5 30
End
```

Set field map files path

FIELD_MAP_PATH *path*

Set at the top of the structure file (*.dat), it allows to indicate the directory where is store thefield map file. This path can be absolute of relatice to the project file path.

Set output field map frame

SUPERPOSE_MAP_OUT Z_0 X_0 Y_0 θ_{Z0} θ_{X0} θ_{Y0}

(X_0, Y_0, Z_0) gives to the position in [m], in (X, Y, Z) frame, of the exit point of the simulation in the field map,

(θ_{X0}, θ_{Y0}, θ_{Z0}) gives the rotation angles in [°] between the reference trajectory directions at the exit point of the simulation in the field map $\left(\vec{e}_x, \vec{e}_y, \vec{e}_z\right)$ **and** $(\vec{e}_X, \vec{e}_Y, \vec{e}_Z)$

See Field map with curved reference trajectory chapter or FIELD_MAP element for get more details.

Set beam phase error

SET_BEAM_PHASE_ERROR *Dp(°) RamdomFlag(0/1)*

If *Dp* is equal to zero, this command allows to cancel, phase error coming from preceding elements. The beam phase is not affected by this command, the strategy is to shift all the following cavity RF phases, by the difference between reference design and beam phase observe at the command position. By this way, you have the possibility to uncouple the RF phase of some part of the machine and start a linac part with a new RF phase.
This command cannot correct to phase errors coming from dynamic errors, only static or input beam error can be corrected.
If *Dp* not equal to zero and *RamdomFlag* is equel to zero, the RF shift will be increased of *Dp* value.
If *Dp* not equal to zero and *RamdomFlag* is not equal to zero, the RF shift will be increased of a ramdom value between +/-*Dp*.

Develop its own element or diagnostics

This feature allows to each user to develop its own element or diagnostics. A detailed example following explains how to perform it. Use the following '*main.cpp*' file and compile it as a dynamic library. This library has be located either in the structure (*.dat) directory or in the executable directory.

```
/***************************************************************************
main.cpp
Windows -> Dynamic library (dll)
Linux or MacOS -> Dynamic shared object (so, dylib)
-------------------
begin : Wed Dec 1 2010
copyright : (C) 2010 by URIOT Didier
email : duriot@cea.fr
***************************************************************************/

#ifdef _WIN32
  #include <windows.h>
  #define DLL_EXPORT __declspec(dllexport)
#else
  #define DLL_EXPORT
#endif
#include <cmath>
```

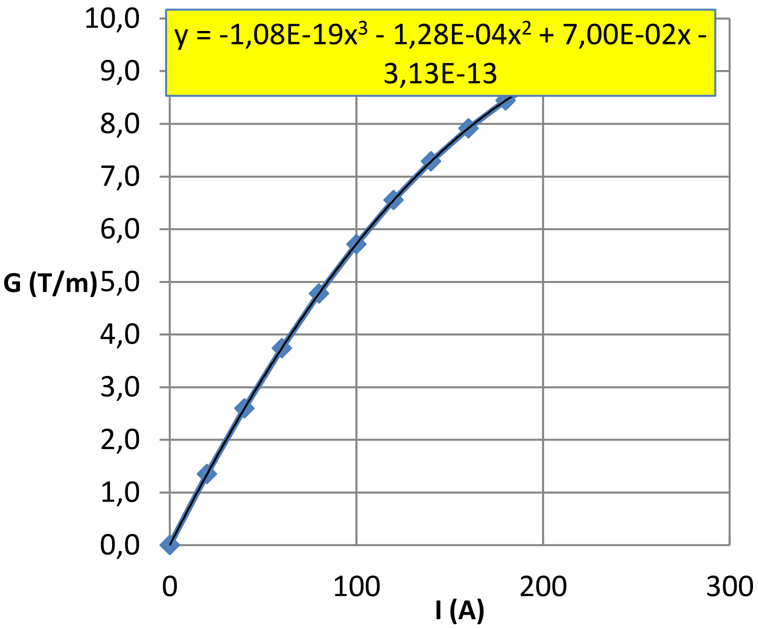
```
#include <cstdio>
#include <cstdlib>
#include <cstring>


#ifdef __cplusplus
extern "C" {
#endif


//---------------------------------------------------------------------------
//-- MY_ELE - MY_ELE - MY_ELE - MY_ELE - MY_ELE - MY_ELE - MY_ELE -----------
//-- MY_ELE - MY_ELE - MY_ELE - MY_ELE - MY_ELE - MY_ELE - MY_ELE -----------
//---------------------------------------------------------------------------
//
// Syntax in struture.dat file
//
// MY_ELE(my_ele_name) L(mm),R(mm),Nstep,arg3,arg4...arg20
// following functions have to be defined
//  - my_ele_name_partran   : for the beam tracking treatment
//  - my_ele_name_envelope  : for the beam envelope treatment
//
// If "my_ele_name" is not defined (MY_ELE target arg2,arg3...arg10)
// following functions have to be defined
//  - my_ele_partran   : for the beam tracking treatment
//  - my_ele_envelope  : for the beam envelope treatment
//
// Settings (L, R, Nstep) must be defined
//  - Element length (mm)
//  - Aperture R (mm): This parameter are only used for plotting, you have to
manage yourself the particle losses.
//  - Nstep : the element will cut in Nstep part with a space-charge kick each 2
steps
//            If Nstep = 0 the number of space-charge kick will be defined by
TraceWin according to "step of space-charge" parameter of "Multiparticle" page.
//            If Nstep = 1 the element will not be cut in and no space-charge kick
will be applied
//  - arg3 to arg20 : Free parameters available for user
//

// Below you can find an example (Drift element)
//
// Syntax: MY_ELE(my_drift) L(mm),R(mm),Nstep,arg3,arg4...arg20
// 2 functions habe to be defined
//  - my_drift_partran   : for the beam tracking treatment
//  - my_drift_envelope  : for the beam envelope treatment

// You have to compile this example as an dll
// - my_elements.dll   (for windows)
// - my_elements.so    (for linux)
// - my_elements.dylib (for MacOS)
//
// Commands to compil and link GNU gcc compiler on Windows:
// g++.exe -m32 -Wall -c main.cpp -o main.o
// g++.exe -m32 -shared -Wl main.o -o my_elements.dll
// If you use 64bits TraceWin version replace both -m32 by -m64

// Commands to compil and link GNU gcc compiler on Linux:
// g++ -m32 -Wall -c main.cpp -o main.o
// g++ -m32 -fPIC -shared -Wl main.o -o my_elements.so
// If you use 64bits TraceWin version replace both -m32 by -m64

// Commands to compil and link GNU gcc compiler on MAC OS:
// g++ -m32 -Wall -pedantic -c  main.cpp -o main.o
// g++ -m32 -Wall -shared -dynamiclib  cone.o -o my_elements.dylib
// If you use 64bits TraceWin version replace both -m32 by -m64

// MY_ELE Multiparticle (Drift example) [ MY_ELE(my_drift) L(mm), R(mm), Nstep ]
// MY_ELE Multiparticle (Drift example) [ MY_ELE(my_drift) L(mm), R(mm), Nstep ]
int DLL_EXPORT my_drift_partran(double Zs,double *param,int Nele,int npart,double
*cord,double *loss,double freq,double mass0,int q,double *ws,double *Ibeam,double
*extra_param,char *error_mess)
{
  double x,y,xp,yp,z,w,dpsp,r,zzs,bgs,gamma,gams,betas,ds,Aperture;
  //
  // Zs           :  Current position in the element (from 0 to Length-Length/Nstep)
  // param[1]     : Length/Nstep (m)
  // param[2]     : Aperture (m)
  // param[3]     : Nstep
  // cord         : See example
  // loss         : see example
  // freq         : beam frequency (Hz)
  // mass         : Particle mass (eV)
  // q            : particle charge state
  // *ws          : reference kinetic energy (eV) (can be modified)
  // *Ibeam       : Beam current (A) (can be modified)
  // error_mess   : TraceWin stop and show this error message if this function
return 0
  // if error_mess!="" and function return 1, this message is print to the standard
console without stop TraceWin

  ds=param[1];
  Aperture=param[2];
  strcpy(error_mess,"");
  zzs=(*ws)/mass0;
  bgs=sqrt(zzs*(2.+zzs));
  gams=1.0+zzs;
  betas=bgs/gams;

  for (int i=0;i<npart;i++) {
    if ((int)loss[i]==0) {
      x=cord[i*6];      // m
      xp=cord[i*6+1];   // rad
      y=cord[i*6+2];    // m
      yp=cord[i*6+3];   // rad
      z=cord[i*6+4];    // m
      dpsp=cord[i*6+5]; // dp/p
      r=sqrt(x*x+y*y);
      if (r>Aperture) loss[i]=Nele; // A particle lost has to be set to Element
number
      else {
        x=x+ds*xp;
        y=y+ds*yp;
        w=dpsp*betas*betas*gams*mass0+(*ws);
        gamma = 1+w/mass0;
        z=z+dpsp*ds/(gamma*gamma);
      }
      cord[i*6]=x;
      cord[i*6+2]=y;
      cord[i*6+4]=z;
    }
  }
  return(1);
}


//------------------------------------------------------------------------------
--------------------------------------------------------------------------------

// MY_ELE Envelope (Drift example)
// MY_ELE Envelope (Drift example)
int DLL_EXPORT my_drift_envelope(double Zs,double *param,int Nele,double
Tmat[][6],double *Bcent,double freq,double mass0,int q,double *ws,double
*ibeam,double *extra_param,char *error_mess)
{
  //
  // Zs          :  Current position in the element (from 0 to Length-Length/Istep)
  //                my_ele_envelope is first time called wiht parma[1]=Length (in
this case Zs=-10)
  //                the following calls is performed Istep times for Zs from 0 to
Length-Length/Istep (Istep is defined by TraceWin according to "step of calcul"
parameter of "Main" page.
  // param[1]    : Length(m) or Length/Istep (m)
  // param[2]    : Aperture (m)
  // param[3]    : Nstep
  // cord        : See example
  // loss        : see example
  // freq        : beam frequency (Hz)
  // mass        : Particle mass (eV)
  // q           : particle charge
  // *ws         : reference kinetic energy (eV) (can be modified)
  // *Ibeam      : Beam current (A) (can be modified)
  // error_mess  : TraceWin stop and show this error message if this function
return 0
  // if error_mess!="" and function return 1, this message is print to the standart
console without stop TraceWin

  double gams;
  double ds=param[1];
  strcpy(error_mess,"");


  gams = 1+(*ws)/(mass0);
  Tmat[0][0]=Tmat[1][1]=1;
  Tmat[0][1]=ds;
  Tmat[2][2]=Tmat[3][3]=1;
  Tmat[2][3]=ds;
  Tmat[4][4]=Tmat[5][5]=1;
  Tmat[4][5]=ds/(gams*gams);

  return(1);
}



//--------------------------------------------------------------------------------
//-- MY_DIAG - MY_DIAG - MY_DIAG - MY_DIAG - MY_DIAG - MY_DIAG - MY_DIAG ---------
//-- MY_DIAG - MY_DIAG - MY_DIAG - MY_DIAG - MY_DIAG - MY_DIAG - MY_DIAG ---------
//--------------------------------------------------------------------------------
//
// Syntax in struture.dat file
//
// MY_DIAG(my_diag_name)(Weight) Diag_number target arg2,arg3...arg10
// Diag_number is the number to associated to ADJUST commands
// “(Weight)” is optional
//
// following functions have to be defined
//  - my_diag_name_partran   : for the beam tracking treatment
//  - my_diag_name_envelope  : for the beam envelope treatment
//
// If "my_diag_name" is not defined (MY_DIAG target arg2,arg3...arg10)
// 2 functions are defined
//  - my_diag_partran   : for the beam tracking treatment
//  - my_diag_envelope  : for the beam envelope treatment
//
// TraceWin Criteria = pow(target_value - diag_value),2)
//
//
// below you can find an example of a beam position or beam size measurment :
// Syntax: MY_DIAG(my_pos_and_size) target arg2,arg3...arg10
// following functions have to be defined
//  - my_pos_and_size_partran   : for the beam tracking treatment
//  - my_pos_and_size_envelope  : for the beam envelope treatment
// You have to compile this example as an dll
// - my_elements.dll   (for windows)
// - my_elements.so    (for linux)
// - my_elements.dylib (for MacOS)

// Commands to compil and link GNU gcc compiler on Windows:
// g++.exe -m32 -Wall -c main.cpp -o main.o
// g++.exe -m32 -shared -Wl main.o -o my_elements.dll
// If you use 64bits TraceWin version replace both -m32 by -m64

// Commands to compil and link GNU gcc compiler on Linux:
// g++ -m32 -Wall -c main.cpp -o main.o
// g++ -m32 -fPIC -shared -Wl main.o -o my_elements.so
// If you use 64bits TraceWin version replace both -m32 by -m64

// Commands to compil and link GNU gcc compiler on MAC OS:
// g++ -m32 -Wall -pedantic -c  main.cpp -o main.o
// g++ -m32 -Wall -shared -dynamiclib  cone.o -o my_elements.dylib
// If you use 64bits TraceWin version replace both -m32 by -m64


// MY_DIAG Multiparticle
// MY_DIAG Multiparticle
int DLL_EXPORT my_pos_and_size_partran(double *diag_value,double *param,int
Nele,int npart,double *cord,double *loss,double freq,double mass0,int q,double
ws,double Ibeam,double *extra_param,char *error_mess)
{
  double x2,x,y,xp,yp,z,dpsp,xmoy;
  int ng;
  //
  // param[1]     : target_value
  // param[2]     : you can use this second parameter to define to diagnostic types
  // param[3->8]  : free
  // diag_value   : value of your diagnostic
  // cord         : See example
  // loss         : see example
  // freq         : beam frequency (Hz)
  // mass         : Particle mass (eV)
  // q            : particle charge state
  // ws           : reference kinetic energy (eV) (can be modified)
  // Ibeam        : Beam current (A) (can be modified)
  // error_mess   : TraceWin stop and show this error message if this function
return 0
  // if error_mess!="" and function return 1, this message is print to the standart
console without stop TraceWin

  strcpy(error_mess,"");

  xmoy=x2=0;
  ng=0;
  for (int i=0;i<npart;i++) {
    if ((int)loss[i]==0) {
      x=cord[i*6];       // m
      xp=cord[i*6+1];    // rad
      y=cord[i*6+2];     // m
      yp=cord[i*6+3];    // rad
      z=cord[i*6+4];     // m
      dpsp=cord[i*6+5]; // dp/p

      x2+=x*x;
      xmoy+=x;
      ng++;
    }
  }
  if (param[2]==0) {
    xmoy/=ng;
    *diag_value=xmoy;  // X position (m)
  }
  if (param[2]==1) {
    x2=sqrt(x2/ng);
    *diag_value=x2;  // X size (m)
  }

//  sprintf(error_mess,"%lg %d %lg %lg",param[2],ng,x2,*diag_value);

  return(1);
}

//----------------------------------------------------------------------------------
------------------------------------------------------------------------------------

// MY_DIAG Envelope
// MY_DIAG Envelope
int DLL_EXPORT my_pos_and_size_envelope(double *diag_value,double *param,int
Nele,double Bmat[][6],double *Bcent,double freq,double mass0,int q,double ws,double
ibeam,double *extra_param,char *error_mess)
{
  //
  // param[1]    : target_value
  // param[2]    : you can use this second parameter to define to diagnostic types
  // param[3->8] : free
  // diag_value  : value of your diagnostic
  // Nele        : Element number
  // Bmat        : Beam matrix 6x6 (x(m), xp(rad, y(m), yp(rad), z(m), dp/p)
  // Bcent       : beam centroid vector (x(m), xp(rad, y(m), yp(rad), z(m), dp/p)
  // freq        : beam frequency (Hz)
  // mass        : Particle mass (eV)
  // q           : particle charge state
  // ws          : reference kinetic energy (eV) (can be modified)
  // Ibeam       : Beam current (A) (can be modified)
  // error_mess  : TraceWin stop and show this error message if this function
return 0
  // if error_mess!="" and function return 1, this message is print to the standart
console without stop TraceWin

  printf("POS ENV\n");

  if (param[2]==0) {
    *diag_value=Bcent[1];   // X beam position (m)
  }
  if (param[2]==1) {
    *diag_value=sqrt(Bmat[0][0]);  // beam size (X); (m)
  }
  return(1);
}

#ifdef __cplusplus
} // "C"
#endif
```

Set magnetic excitation curve

EXCITATION_CURVE a_0 a_1 a_2 a_3

To set the strength of magnetic element as quadrupole, solenoid or dipole, the field map parameter *kb* has to be set to amplitude depending of the field defined in the map. It is sometime more convenient to use directly the power supply current when we get the excitation curbe B(I) or G(I). The gradient normalization is calculated at
5% of the bore radius. This command has to be defined at the first occurance of a field map defined by this file name. All following field map using the same file are automatiquely concerned. See following example using quadrupole.

```
DRIFT 10 100
LME-Q11 : FIELD_MAP 70 480 0 40 5.20 0 0 0 qpole_file ; kb=5.20 -> gradient is not knew directly
DRIFT 10 10
LME-Q12 : FIELD_MAP 70 480 0 40 10.4 0 0 0 qpole_file; kb=10.4 -> gradient is not knew directly
DRIFT 10 100
END
; Here the gradient values depend of the kb parameters and the field amplitudes defined in the qpole_file.
```



The excitation curbe G(I) can be aproximativement defined as polynome and and its coefficients are used in the commande.

```
DRIFT 10 100
EXCITATION_CURVE -3.12e-13 7.00e-2 -1.28e-4 -1.08e-19
LME-Q11 : FIELD_MAP 70 480 0 40 40 0 0 0 qpole_file  ; kb=40A -> gradient is set 2.59 T/m
DRIFT 10 10
LME-Q12 : FIELD_MAP 70 480 0 40 80 0 0 0 qpole_file ;  kb=80A -> gradient is set 4.77 T/m
DRIFT 10 100
END
```

EXCITATION_CURVE2

Thi second command allows to use all table G(I) data. In this case a file named as the field map name ("qpole_file.gi" in the following example) has to be located with the magnetic field map files. The syntax of the file is show below, the first number indicates the nomber of lines, then first collum is the power supply currant (A) and the second one is the gradient (T/m). A splin interpolator will be used to estimate G(I).

```
DRIFT 10 100
EXCITATION_CURVE2
LME-Q11 : FIELD_MAP 70 480 0 40 40 0 0 0 qpole_file  ; kb=40A -> gradient is set 2.59 T/m
DRIFT 10 10
LME-Q12 : FIELD_MAP 70 480 0 40 80 0 0 0 qpole_file ;  kb=80A -> gradient is set 4.77 T/m
DRIFT 10 100
END
```

11	
0	0,0000
25	1,6813
50	3,3407
75	5,0350
100	6,6783
125	8,3435
150	9,9867
175	11,5855
200	12,9745
225	14,0765
265	15,4146

Cavity tuning

TUNE_CAVITY *Diag# Δp Nm RPos#1 type#1 Err#1 Noise#1 RPos#2 type#2 Err#2 Noise#2*

This command allows simulating the RF linac cavity tune process.
Valid for all RF accelerating elements (FIELD_MAP, GAP, NCELLS…).

It is made by:

- performing a scan phase of the *perfect model*,
- performing a scan phase of the *real model*,
- adjusting the RF field phase and amplitude in the real model to minimize difference between above scans.

The *perfect model* is given by the transport in the cavity with nominal RF field of the reference beam associated to a measurement without error.
The *real model* is given by the transport in the cavity with wrong RF field of the real beam associated to a measurement with error.

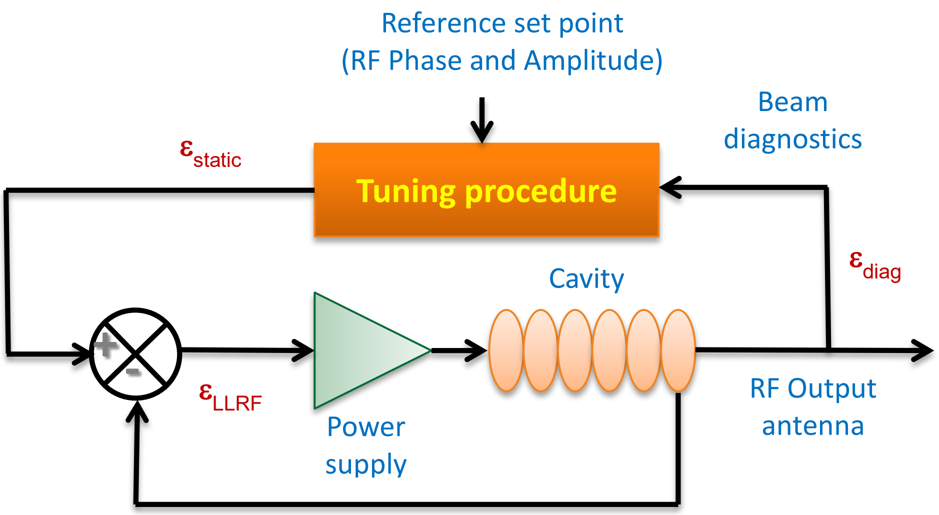
The results, which can be found like usual ADJUST results in the result file (*.cal), are the relative RF amplitude and absolute phase correction (with respect to theoretical value) to be applied to the considering cavity. For "perfect" case without error, results are respectively 1 for amplitude (multiply by 1 the amplitude) and 0° for phase (add 0° to the phase).

Simulating the RF tuning process in TraceWin allows making a clear distinction between static and dynamic longitudinal RF errors:

- *Static error* is then the error on the field set point obtained by this diagnostics-based tuning,
- *Dynamic error* is the error of the cavity field control around this set point (LLRF, thermal shifts…).

When using this method, the parameters set by "*ERROR_CAV_NCPL_STAT*" don't give anymore the static errors on the RF field, but only the field starting point for the tuning.

This new command makes more coherent simulations, much closer to realistic behavior of a machine. The main objective is to be able to define the measurement accuracy required for diagnostics involved in the cavity tuning process and check the robustness of the RF tuning process. By this way, the RF static error, usually set arbitrary (to 1°, 1% for example), which make longitudinal transport diverges very quickly, should be compensated by the RF tuning algorithm itself.



Parameters:
(Red parameters are optional)

Diag#: Order of the tuning processes in the diagnostics tuning procedure list.
Δp: Range of RF phase scan [°], ±*Δp* around theoretical set value.
Nm: Number of scan steps (*Nm*+1 measurement).

Rpos#1: Relative positon (number of element, excluding DIAG_XX) according to the cavity position.
Type#1: flag (0/1), 0: tuning using TOF, 1: beam phase measurement.
Err#1: Diagnostics systematic error amplitude in % for TOF and in mm for phase corresponding to the BPM longitudinal position error.
Noise#1: Diagnostics random error amplitude in % for TOF and in degree for phase.

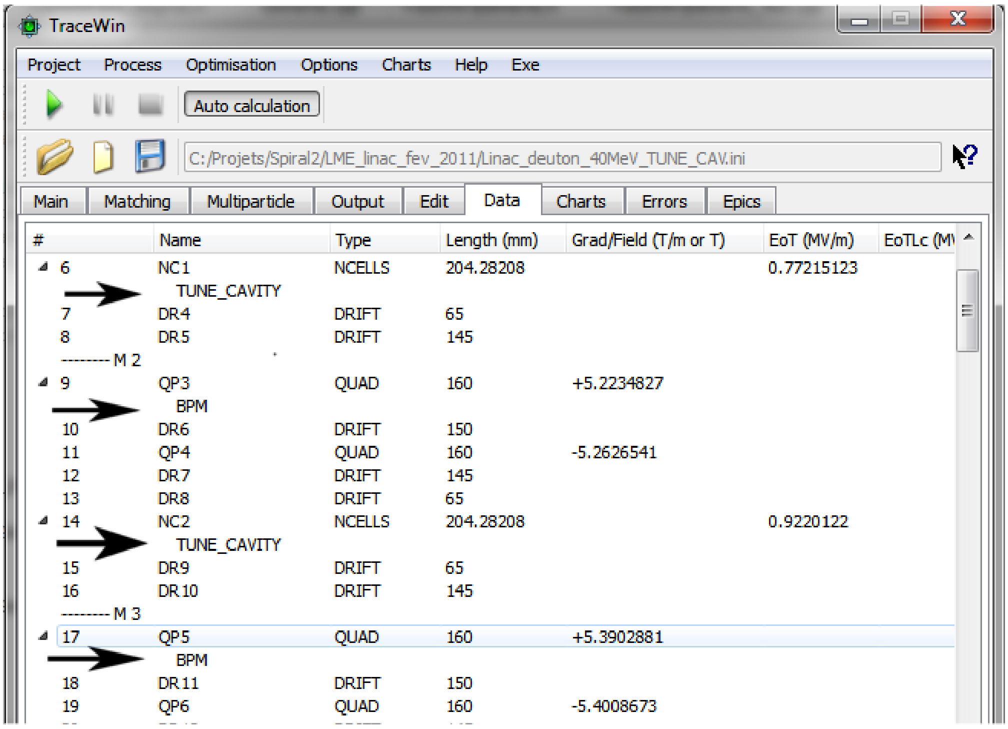
Rpos#2: Relative positon (number of element, excluding DIAG_XX) according to the cavity position.
Type#2: flag (0/1), 0: tuning using TOF, 1: beam phase measurement.

Err#2: Diagnostics systematic error amplitude in % for TOF and in mm for phase corresponding to the BPM longitudinal position error.
Noise#2: Diagnostics random error amplitude in % for TOF and in degree for phase.

Rpos# set to '*0*' means no diagnostic.
BPM: Beam phase measurement.
TOF: Beam energy measurement.

For this specific diagnostics command, the user doesn't have to specify diagnostics elements in the structure file. They are directly defined after the 3rd of this command.
For energy measurement, if both TOFs are positioned around the cavity, the cavity energy gain is used instead of absolute output energy.
For phase measurement, if only one diagnostics is defined then the relative beam phase is used at the diagnostics position.
Otherwise, phase difference between both diagnostics positions is used. Considering in this case that a systematic phase offset, depending on the cable length and hardware electronics, are unknown, we applied the following method:
The quantity $(\varphi_{bpm2}\text{-}\ \varphi_{bpm2.0}) - (\varphi_{bpm1}\text{-}\ \varphi_{bpm.1.0})$ is matched to $(\varphi_{bpm2simul}\text{-}\ \varphi_{bpm2.0simul}) - (\varphi_{bpm1simul}\ \text{-}\ \varphi_{bpm1.0simult})$, where '*.0*' in the phase subscript refers to the phase values with the RF off. This approach makes cavity tuning less sensitive to BPM position errors.

During tuning process, cavities downstream the tuned cavity, are considered detuned (field set to 0).
As for all diagnostic elements, errors are considered only if the option "*Take into account diagnostic accuracy*" option is checked in the "*Matching*" tab-sheet. In case of 2 different measurement positions, errors are applied independently on each of them. If a BPM or TOF positon defined in 2 diffierent TUNE_CAVITY commands are the same, the applied error will be identical. To see position of BPM or TOF use the page-sheet "Data" to visualize the item "BPM" or "TOF", (see picture below)



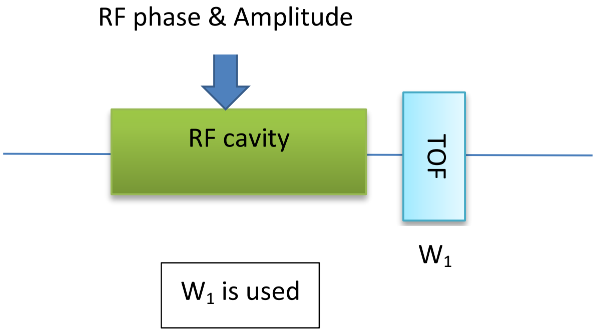
#	Name	Type	Length (mm)	Grad/Field (T/m or T)	EoT (MV/m)	EoTLc (M
6	NC1	NCELLS	204.28208		0.77215123	
	TUNE_CAVITY					
7	DR4	DRIFT	65			
8	DR5	DRIFT	145			
-------- M 2						
9	QP3	QUAD	160	+5.2234827		
	BPM					
10	DR6	DRIFT	150			
11	QP4	QUAD	160	-5.2626541		
12	DR7	DRIFT	145			
13	DR8	DRIFT	65			
14	NC2	NCELLS	204.28208		0.9220122	
	TUNE_CAVITY					
15	DR9	DRIFT	65			
16	DR10	DRIFT	145			
-------- M 3						
17	QP5	QUAD	160	+5.3902881		
	BPM					
18	DR11	DRIFT	150			
19	QP6	QUAD	160	-5.4008673		

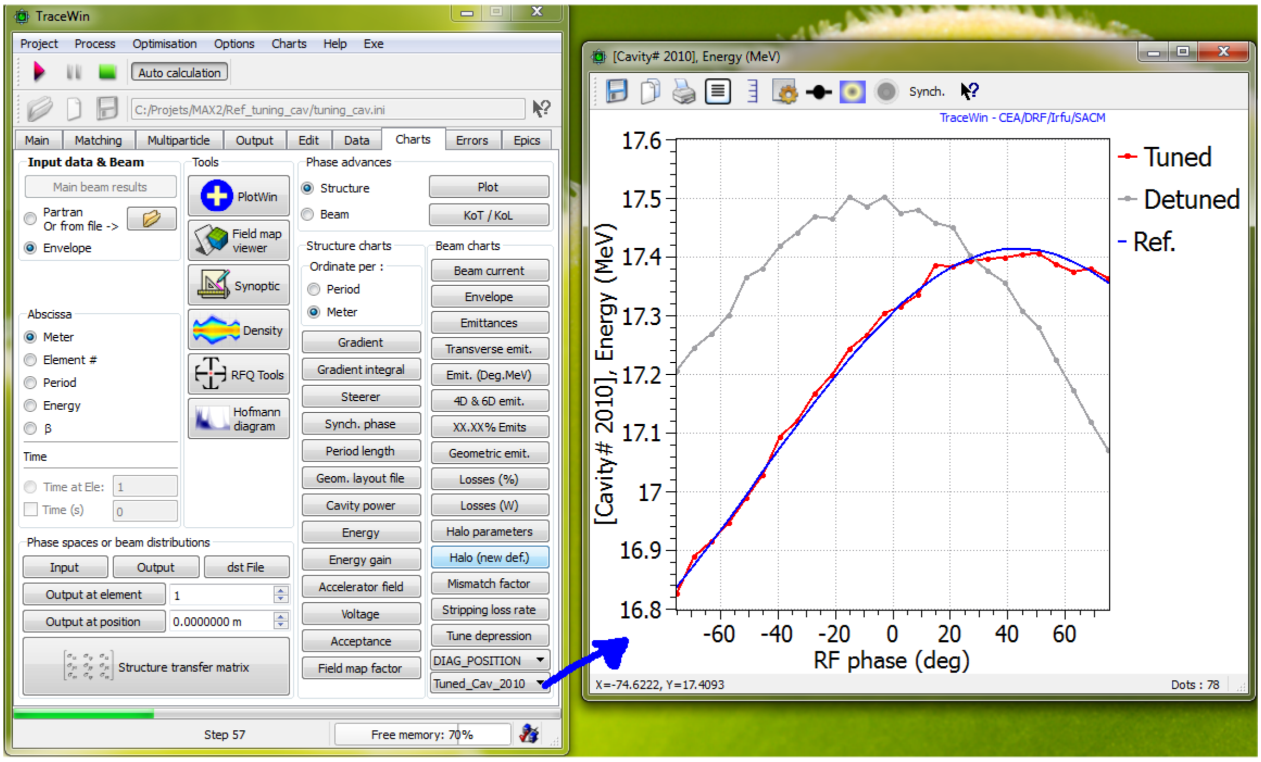
The following examples illustrate different possible configurations where the cavity is simulated using his 3D field map.

<u>Example 1 (Absolute energy):</u>

The cavity is tuned using a downstream TOF diagnostic (1 element later).

```
ERROR_CAV_NCPL_STAT 20 1 0. 0. 0. 0. 50 15 0
; Static RF errors are defined to 20% for amplitude and 50° for the phase
DRIFT 100 28 0
DRIFT 100 28 0
SET_SYNC_PHASE
TUNE_CAVITY 2010 75 25 1 0 0.1 0.1
FIELD_MAP 100 415.16 -45 28 1.83091 1.83091 0 0 spoke
DRIFT 100 28 0
; Position of the TOF1
DRIFT 100 28 0
```





C:/Projets/MAX2/Ref_tuning_cav/tuning_cav.ini

Time at Ele: 1

Time (s) 0

Output at element 1

Output at position 0.0000000 m

DIAG_POSITION

Tuned_Cav_2010

X=-74.6222, Y=17.4093

Dots : 78

Step 57

Free memory: 70%

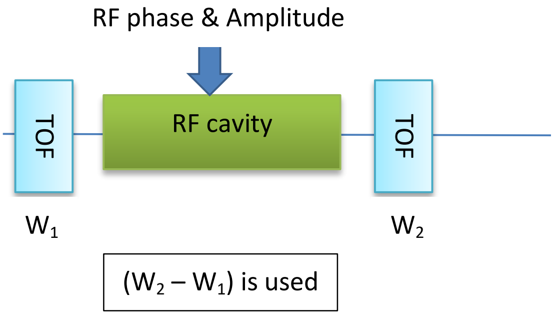
Using TraceWin output chart page, we can observe the cavity tuning procedure results. The gray curve (*Detuned*) show the initial RF tuning of the cavity which takes into account RF static error defined in this example (20% & 50°). The blue curve is the RF tuning objectives based on perfect model and the red one is the result of the cavity after the tuning procedure. Here, the procedure is based on a RF phase scan of ± 75° with 25 steps of measurement.

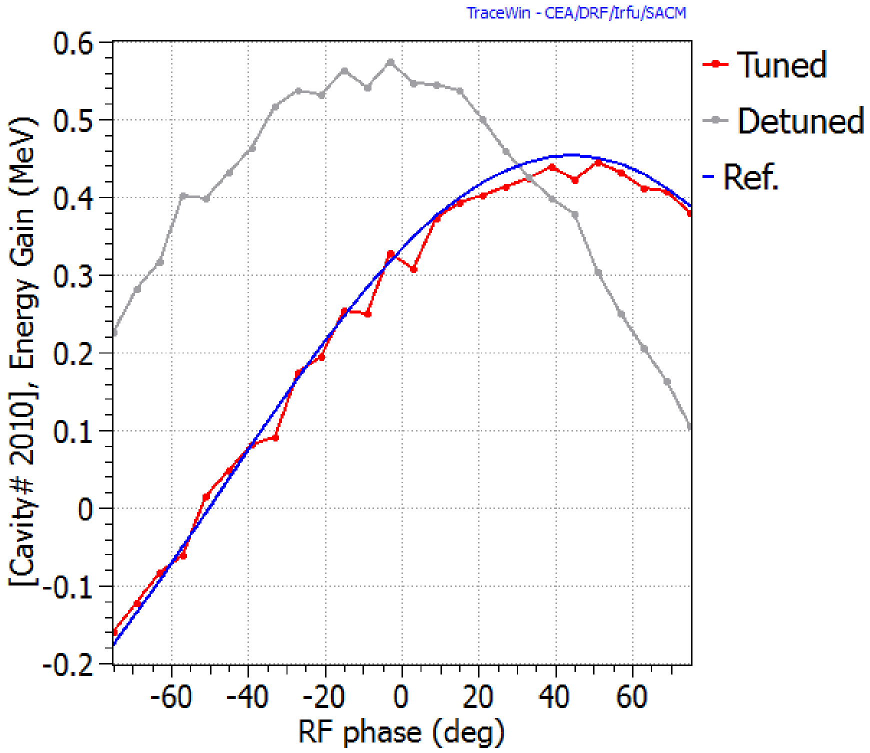
So, starting from the gray curve, the RF amplitude and phase are adjusted to minimize the difference between the results (red curve) with objectives (blue curve). This tuning is not perfect because diagnostics measurement errors are also included. In this example, a ±1% systematic error and a ±0.01% random noise are applied to the 26 beam energy measurements.

<u>Example 2 (relative energy):</u>

The cavity is tuned using a 2 x TOF diagnostics located upstream and downstream the tuned cavity, which corresponds to the measurement of the beam energy gain in the cavity.

```
ERROR_CAV_NCPL_STAT 20 1 0. 0. 0. 0. 50 15 0
; Static RF errors are defined to 20% for amplitude and 50° for the phase
DRIFT 100 28 0
; Position of the TOF1, 1 element backward
DRIFT 100 28 0
SET_SYNC_PHASE
TUNE_CAVITY 2010 75 25 -1 0 0.1 0.1 1 0 0.1 0.1
FIELD_MAP 100 415.16 -45 28 1.83091 1.83091 0 0 spoke
DRIFT 100 28 0
; Position of the TOF2, 1 element forward
DRIFT 100 28 0
```

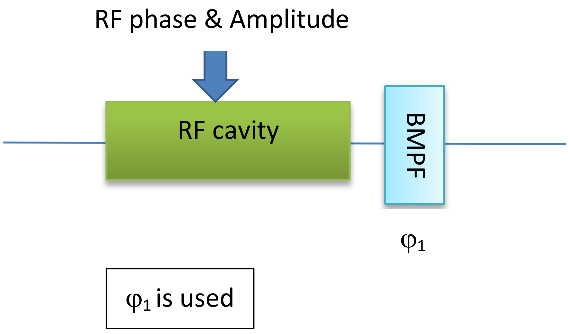


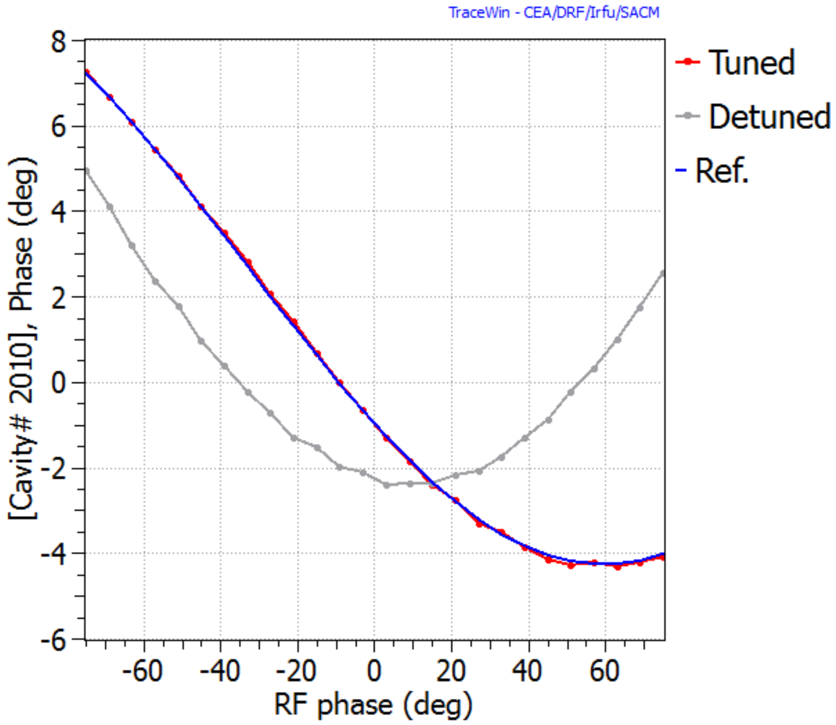


Example 3 (relative phase):

The cavity is tuned using the relative beam phase measurement. In this configuration, the distance between the cavity being adjusted and the BPM is an important parameter. We can see on the results curve, the effect of the 0.3 mm or BPM position and the 0.25° of measurement noise.

```
ERROR_CAV_NCPL_STAT 20 1 0. 0. 0. 0. 50 15 0
; Static RF errors are defined to 20% for amplitude and 50° for the phase
DRIFT 100 28 0
DRIFT 100 28 0
SET_SYNC_PHASE
TUNE_CAVITY 2010 75 25 1 1 0.1 0.1
FIELD_MAP 100 415.16 -45 28 1.83091 1.83091 0 0 spoke
DRIFT 100 28 0
; Position of the BPM1 , 1 element forward
DRIFT 100 28 0
```

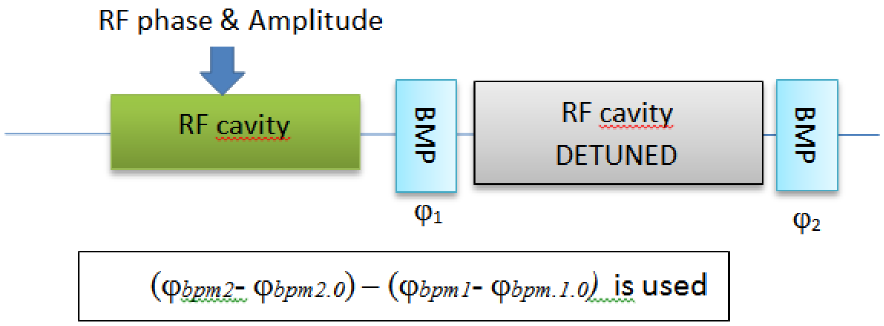


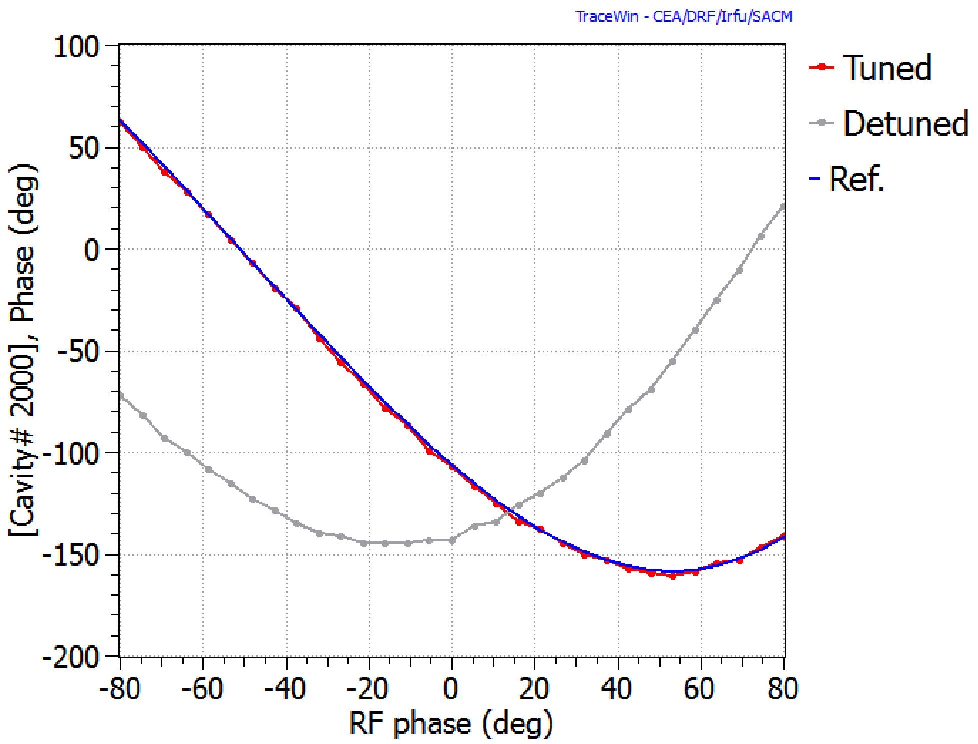


Example 4 (absolute phase):

The cavity is tuned using the absolute phase measurements of the beam according to master RF phase. In this configuration, a first measurement is done with RF cavity off.

```
DRIFT 100 28 0
DRIFT 100 28 0
SET_SYNC_PHASE
TUNE_CAVITY 2010 75 25 1 1 2.0 1.0 4 1 2.0 1.0
FIELD_MAP 100 415.16 -45 28 1.83091 1.83091 0 0 spoke
DRIFT 100 28 0
; Position of the BPM1, 1 element forward
DRIFT 100 28 0
; This cavity is detuned during scan process
SET_SYNC_PHASE
FIELD_MAP 100 415.16 -44 28 1.83091 1.83091 0 0 spoke
DRIFT 100 28 0
; Position of the BPM2, 1 element forward
DRIFT 100 28 0
```

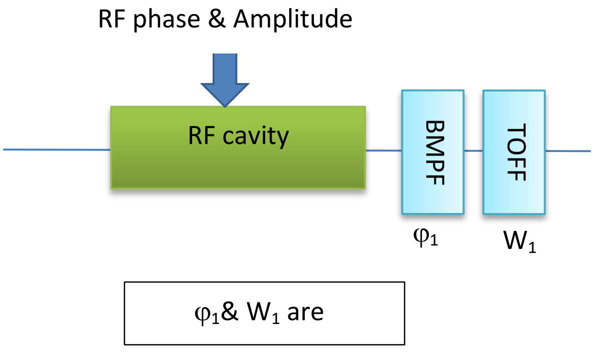


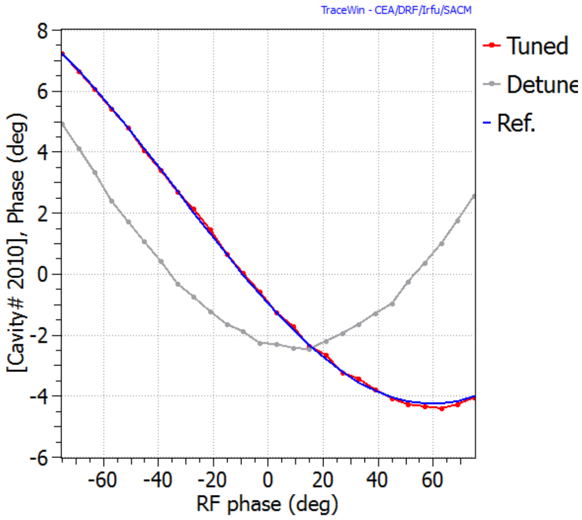


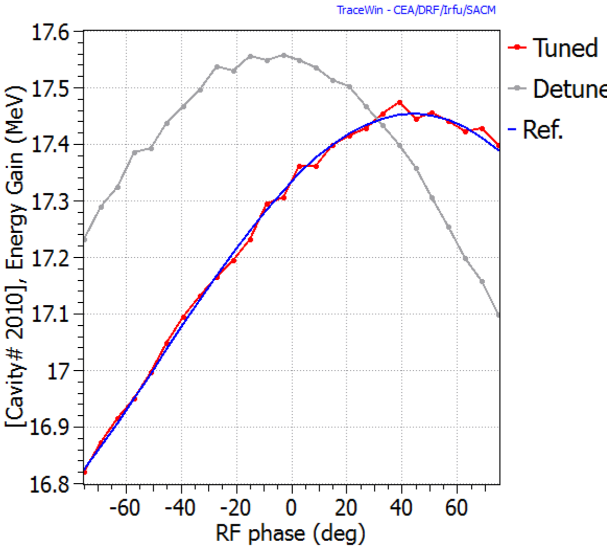
Example 5:

The cavity is tuned using the absolute phase and energy measurement of the beam.

```
DRIFT 100 28 0
DRIFT 100 28 0
SET_SYNC_PHASE
TUNE_CAVITY 2010 75 25 1 1 0.1 0.1 1 0 0.1 0.1
FIELD_MAP 100 415.16 -45 28 1.83091 1.83091 0 0 spoke
DRIFT 100 28 0
; Position of the BPM1, 1 element backward
; Position of the TOF1, 1 element forward
DRIFT 100 28 0
```





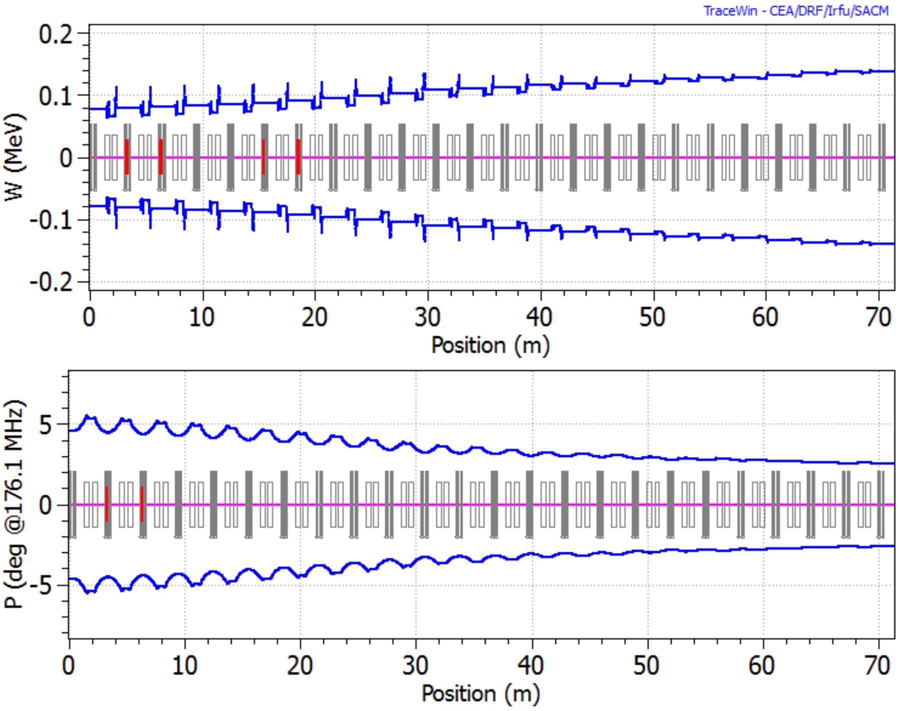


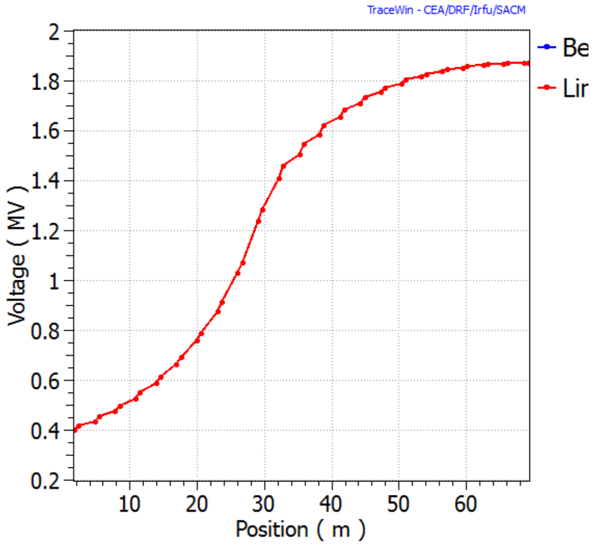
Example 6:

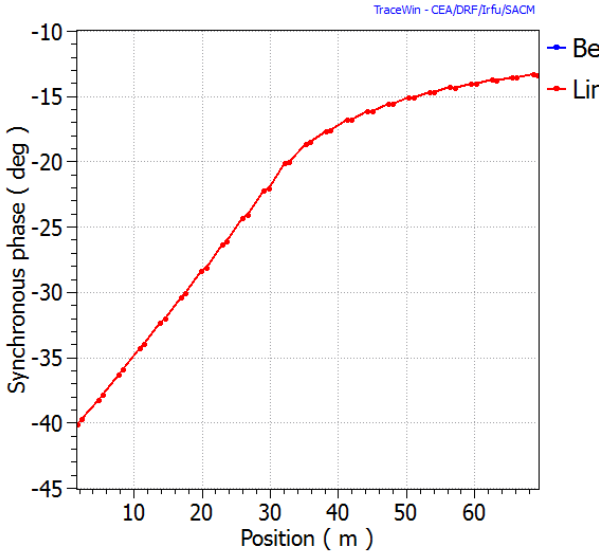
See "*tuning_cav.ini*" project in example list.

This example is the tuning results of a full linac composed of 45 cavities accelerating the beam from 17 MeV to 73 MeV. All cavities are tuned using only one beam phase monitor located about 1 meter downstream.

Without errors:



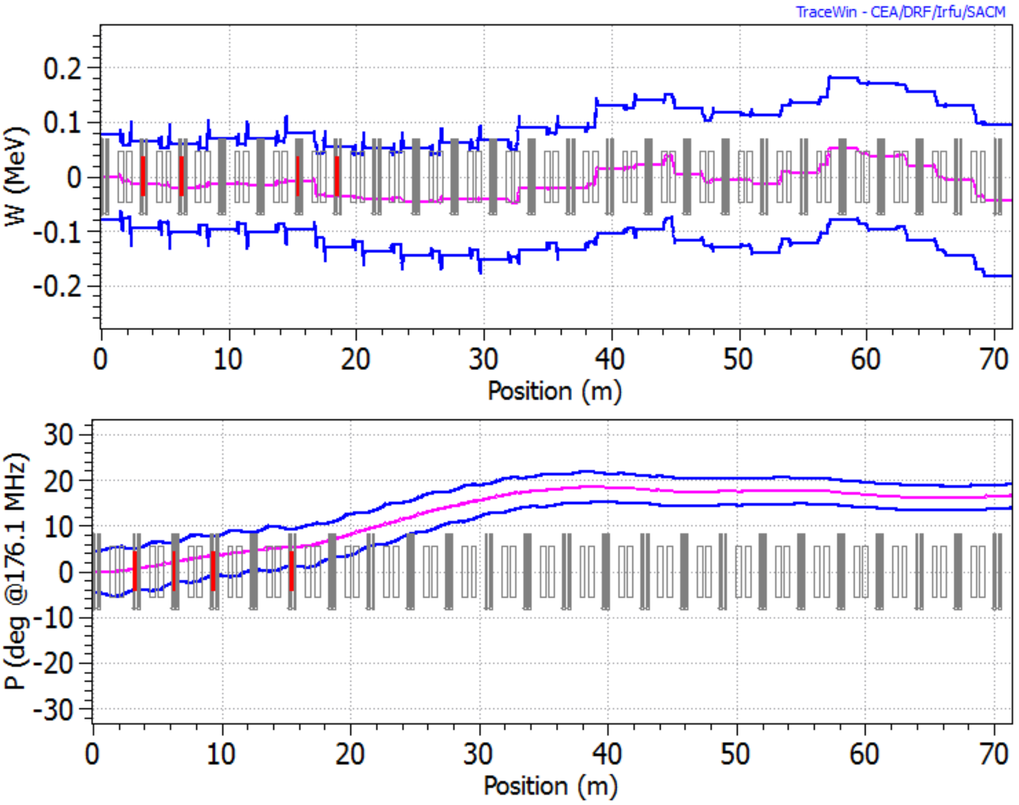


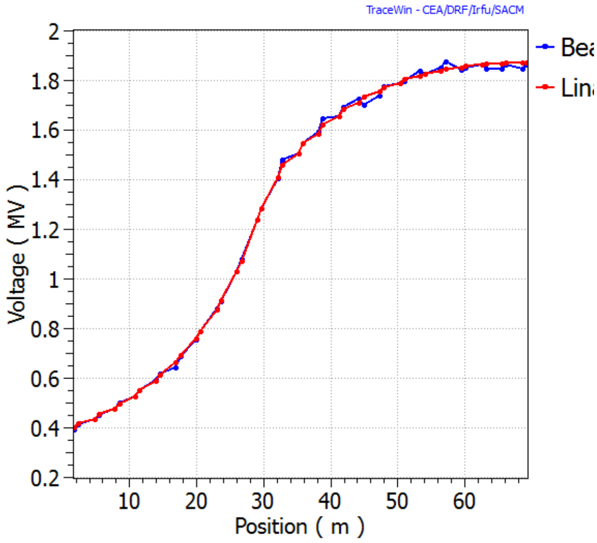


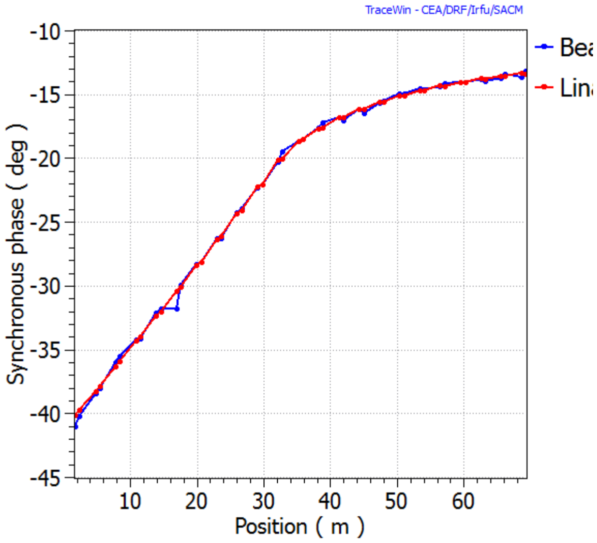
With errors:
The diagnostics position error is within ±0.3 mm and the phase measurement noise is within ±0.25°. To tune the cavity, the RF phase scan is within ±75° with 25 steps.

The initial RF errors are respectively set to 20% and 50° to simulate an initial tuning when operator doesn't know at all the starting set point of the RF.

This result shown below corresponds only to one set of errors and statistics should be performed to be able to define accurately the acceptable error levels for this machine.

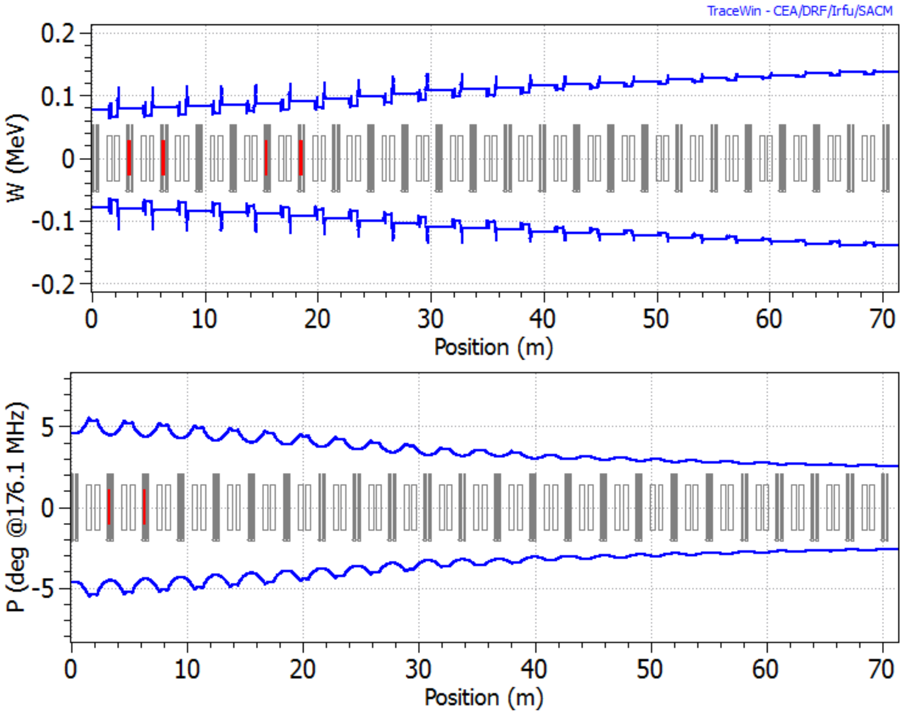






Despite huge initial errors, the final tuned cavity voltages and synchronous phases are close to the perfect machine. The output beam energy and phase differ from the reference ones, respectively by +0.04 MeV and -17 °. However the longitudinal acceleration and focalization give a very good transport (see below, envelope plots without beam centroid).

The consequence of using tuning procedure is to generate different longitudinal reference linacs. In other terms, **there is an infinity of tuned machines.** Another set of errors will give another linac tuning. Only statistical study is able to calculate the statistical behavior of the machine including cavities tuning considering errors applied to structure and diagnostics.



Remark: using 2 BPMs makes machine tuning much less sensitive to BPM position errors.

<u>Example 7:</u>

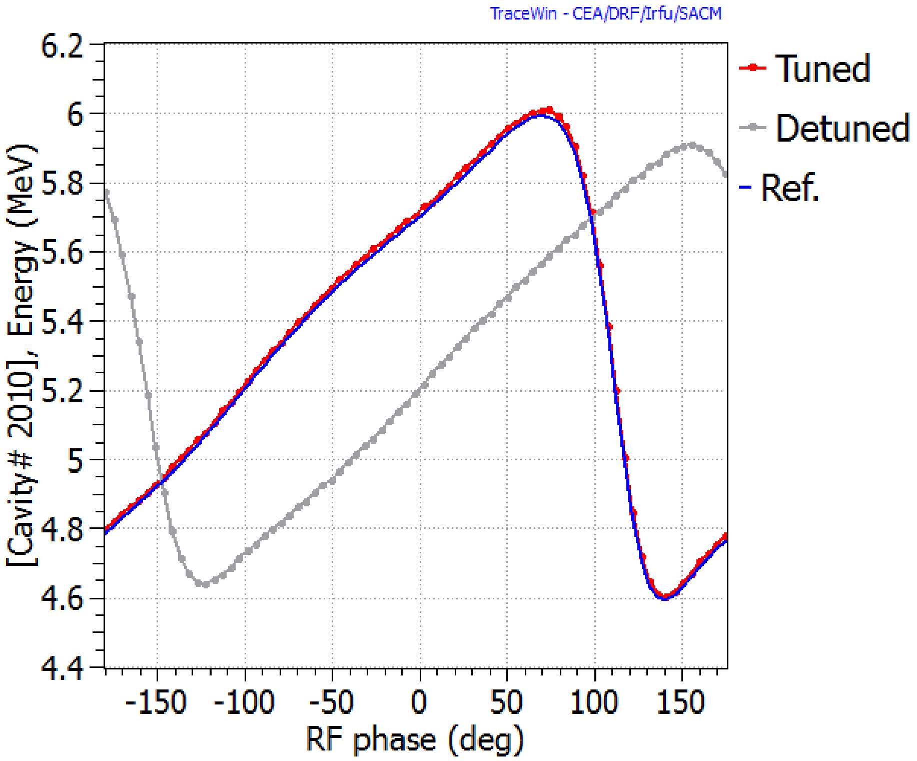
See “*tuning_cav2.ini*” project in example list.

This example is the tuning results of a full linac composed of 24 cavities accelerating the beam from 0.75 MeV to 40 MeV. All cavities are tuned using two beam phase monitors located in the following quadrupoles. Considering 1 mm of position errors, this scheme is very efficient.

Example 8 (DTL cavity):

Here, TOF is used to adjust RF field of a short DTL cavity. For long structure as DTL, the range of RF phase scan has to be large enough. To get realistical tuning simulation, DTL phases are set unsing absolute phase option (*P* parameter of DTL_CEL = 3). The starting RF point errors are set using the coupling version of ERROR_CAV_XXX command.

```
; Proton @ 5.028 MeV
; Frequency = 352.2 MHz

ERROR_CAV_CPL_STAT 2000 1 0. 0. 0. 0. 20 120 0 0
SET_ADV 50
LATTICE 2 0
TUNE_CAVITY 2010 180 75 14 0 0.3 0.1
; 12 cells of DTL structure tank
DTL_CEL 88.0116 28 28 0.00502696 49 -49 66380.6 -50 10 3 0.103342 0.717258 -0.485156 -0.196227
DTL_CEL 88.3889 28 28 0.00833366 -50 50 69119.9 0 10 3 0.103784 0.723731 -0.475542 -0.198168
DTL_CEL 88.7837 28 28 0.0118928 49 -49 72056.6 0 10 3 0.104247 0.731513 -0.463853 -0.199942
DTL_CEL 89.1965 28 28 0.015516 -50 50 75048 0 10 3 0.104732 0.739058 -0.452425 -0.20129
DTL_CEL 89.627 28 28 0.0190039 49 -49 77950.8 0 10 3 0.105237 0.745 -0.443401 -0.202249
DTL_CEL 90.0759 28 28 0.0227398 -50 50 81051.7 0 10 3 0.105763 0.752113 -0.432503 -0.203008
DTL_CEL 90.5436 28 28 0.0265536 49 -49 84218.3 0 10 3 0.106312 0.759079 -0.421723 -0.203311
DTL_CEL 91.0306 28 28 0.0304112 -50 50 87435.5 0 10 3 0.106883 0.765757 -0.411327 -0.20332
DTL_CEL 91.5368 28 28 0.0343218 49 -49 90710.9 0 10 3 0.107477 0.772215 -0.401212 -0.20308
DTL_CEL 92.0622 28 28 0.0380767 -50 50 93904.6 0 10 3 0.108093 0.777295 -0.393249 -0.20285
DTL_CEL 92.6072 28 28 0.042087 49 -49 97300 0 10 3 0.108732 0.783367 -0.383636 -0.20219
DTL_CEL 93.1711 28 28 0.0457241 -50 50 100477 0 10 3 0.109394 0.787047 -0.377858 -0.201992
DRIFT 100 28 0
; Position of the TOF1
DRIFT 0 28 0
END
```


TraceWin - CEA/DRF/Irfu/SACM
[Cavity# 2010], Energy (MeV)
6.2
6
5.8
5.6
5.4
5.2
5
4.8
4.6
4.4
-150
-100
-50
0
50
100
150
RF phase (deg)
Tuned
Detuned
Ref.

Transfer matrices

Funneling gap (*EoTL*, *φs*)

EoTL (V) is the maximum energy gain, *φs* (°) is the synchronous phase.

$$\beta_z = \sqrt{\frac{1-\frac{1}{\gamma_i^2}}{1+x'^2+y'^2}}$$

$$K = |q| E_0 TL \cdot \cos\varphi$$

$$x' = x' + \frac{K}{\beta_z^2 \sqrt{(\gamma_i \cdot mc^2)^2 + \frac{K^2}{\beta_z^2}}}$$

Where *x'* being the horizontal beam centroid slope.

$$R_{xx} = R_{yy} = R_{zz} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad R_{xz} = \begin{bmatrix} 0 & 0 \\ \frac{2\pi |q| E_0 TL \cdot \sin\phi_s}{\gamma_i \beta_z^3 \cdot \lambda \cdot mc^2} & 0 \end{bmatrix}$$

Drift (*Δs*)

Δs (mm) is the drift length.
The non null 2×2 transfer sub-matrixes are:

$$R_{xx} = R_{yy} = \begin{bmatrix} 1 & \Delta s \\ 0 & 1 \end{bmatrix}, \text{ and } \qquad R_{zz} = \begin{bmatrix} 1 & \frac{\Delta s}{\gamma^2} \\ 0 & 1 \end{bmatrix}.$$

Thin lens (*f_x*, *f_y*)

f_x, f_y are focal length in meter
The non null 2×2 transfer sub-matrixes are:

$$R_{xx} = \begin{bmatrix} 1 & 0 \\ \frac{-1}{f_x} & 1 \end{bmatrix}, R_{yy} = \begin{bmatrix} 1 & 0 \\ \frac{-1}{f_y} & 1 \end{bmatrix}, \quad \text{and} \quad R_{zz} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Beam rotation (*θxy, θxz, θzy, dx, dy, dxp, dyp*)

θxy are the beam rotation XY angle in degre
The ellipsoid can be brought upright by rotations -*θxy* accomplished by applying the transfer matrixes:

$$R_{xy} = \left[\begin{array}{cc:cc:cc} \cos(\theta_{xy}) & 0 & \sin(\theta_{xy}) & 0 & 0 & 0 \\ 0 & \cos(\theta_{xy}) & 0 & \sin(\theta_{xy}) & 0 & 0 \\ \hdashline -\sin(\theta_{xy}) & 0 & \cos(\theta_{xy}) & 0 & 0 & 0 \\ 0 & -\sin(\theta_{xy}) & 0 & \cos(\theta_{xy}) & 0 & 0 \\ \hdashline 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right],$$

The rotation is then applied:

$$[\sigma] = R_{xy} \cdot [\sigma] \cdot R_{xy}{}^{T},$$

Quadrupole (*Δs, G*)

Δs (mm) is the quadrupole length, G (T/m) its gradient.

For electrostatic quadrupole $G = \frac{V_0}{\beta_c R^2}$, where *Vo* is the voltage between electrode, *βc* beam speed and *R* is the half distance between electrode.

Lets use $k = \sqrt{\left|\frac{G}{B\rho}\right|}$, with $B\rho = \frac{m_0 c\beta\gamma}{q}$, the magnetic rigidity of the particle.

If q·G is positive, the quadrupole is focusing in the horizontal direction, else it’s defocusing.
The non null 2×2 transfer sub-matrixes are:

In the longitudinal direction, we have: $R_{zz} = \begin{bmatrix} 1 & \frac{\Delta s}{\gamma^2} \\ 0 & 1 \end{bmatrix}$.

In the transverse direction, two possibilities:

Focusing quadrupole (in horizontal direction)

$$R_{xx} = \begin{bmatrix} \cos(k\Delta s) & \dfrac{\sin(k\Delta s)}{k} \\ -k\sin(k\Delta s) & \cos(k\Delta s) \end{bmatrix}, \quad R_{yy} = \begin{bmatrix} \cosh(k\Delta s) & \dfrac{\sinh(k\Delta s)}{k} \\ k\sinh(k\Delta s) & \cosh(k\Delta s) \end{bmatrix}.$$

Defocusing quadrupole (in horizontal direction)

$$R_{xx} = \begin{bmatrix} \cosh(k\Delta s) & \dfrac{\sinh(k\Delta s)}{k} \\ k\sinh(k\Delta s) & \cosh(k\Delta s) \end{bmatrix}, \quad R_{yy} = \begin{bmatrix} \cos(k\Delta s) & \dfrac{\sin(k\Delta s)}{k} \\ -k\sin(k\Delta s) & \cos(k\Delta s) \end{bmatrix}.$$

Bunched cavity or thin gap (*EoTL, φs, p, β*s, *T*s, *kT's,k²T''s,kS', k²S''*)

E₀TL (eV) is the maximum energy gain, *φs* (°) is the synchronous phase.
The reduced energy change in the gap is:

$$\gamma_o = \gamma_i + \frac{|q|E_0TL_{scaled}\cos(\varphi_s)}{mc^2}, \quad \beta = \sqrt{1-\frac{1}{\gamma^2}}$$

The phase shift in the gap is:

If $\beta s = 0$: $\Delta\varphi = 0$ and $L_{scaled}=L$

else $$\Delta\phi = \frac{qE_0TL_{scaled}\cdot\sin(\varphi_s)}{mc^2\cdot\overline{\gamma}^3\cdot\overline{\beta}^2}\left(\frac{kT'}{T}\right) \qquad L_{scaled} = E_0TL\frac{T}{T_s}$$

The changes in the normalized momentum caused by the gap are given by:

$$k_{xy} = \frac{-q\pi E_0TL_{scaled}\sin(\varphi_s)}{mc^2\overline{\beta}^2\cdot\overline{\gamma}^2\ \lambda}, \text{ With } \overline{\gamma} = \frac{\gamma_o+\gamma_i}{2} \text{ and } \overline{\beta} = \frac{\beta_o+\beta_i}{2}.$$

$$k_z = \frac{2q\pi E_0TL_{scaled}\sin(\varphi_s)}{mc^2\overline{\beta}^2\ \lambda}.$$

The non null 2×2 transfer sub-matrixes are:

$$, R_{xx} = R_{yy} = \begin{bmatrix} k_1 C & 0 \\ \frac{k_{xy}}{(\beta\gamma)_o} & k_2 \cdot C \end{bmatrix}, \qquad R_{zz} = \begin{bmatrix} 1 & 0 \\ \frac{k_z}{(\beta\gamma)_o} & \frac{(\beta\gamma)_i}{(\beta\gamma)_o} \end{bmatrix}.$$

- If $\beta_s = 0$: $k_1 = k_2 = 1 - \frac{qE_0 T L_{scaled} \cos(\varphi_s)}{2mc^2 \overline{\beta}^2 \cdot \overline{\gamma}}$

- If $\beta_s \neq 0$:

$$k_1 = 1 - \frac{qE_0 T\, L_{scaled} \cos(\varphi_s)}{2mc^2 \cdot \overline{\gamma}^3 \cdot \overline{\beta}^2} \left(\overline{\gamma}^2 + \frac{kT'}{T} \right), \qquad k_2 = 1 - \frac{qE_0 T\, L_{scaled} \cos(\varphi_s)}{2mc^2 \cdot \overline{\gamma}^3 \cdot \overline{\beta}^2} \left(\overline{\gamma}^2 - \frac{kT'}{T} \right)$$

(*) See Transit time factor definition according to β_s

C: is a coefficient allowing to keep the matrix determinant equal to: $\frac{(\beta\gamma)_i}{(\beta\gamma)_o}$

$$C = \sqrt{\frac{\frac{(\beta\gamma)_i}{(\beta\gamma)_o}}{k_1 k_2}}$$

Solenoid (*Δs, B*)

Δs (mm) is the solenoid length, *B (T)* its axis magnetic field.

Let's use $k = \frac{B}{2B\rho}$, with $B\rho = \frac{m_0 c\beta\gamma}{q}$, the magnetic rigidity of the particle.

The non null 2×2 transfer sub-matrixes are:

$$R_{xx} = R_{yy} = \begin{bmatrix} \cos^2(k\Delta s) & \frac{\sin(k\Delta s)\cos(k\Delta s)}{k} \\ -k\sin(k\Delta s)\cos(k\Delta s) & \cos^2(k\Delta s) \end{bmatrix},$$

$$R_{xy} = -R_{yx} = \begin{bmatrix} \sin(k\Delta s)\cos(k\Delta s) & \dfrac{\sin^2(k\Delta s)}{k} \\ -k\sin^2(k\Delta s) & \sin(k\Delta s)\cos(k\Delta s) \end{bmatrix},$$

$$R_{zz} = \begin{bmatrix} 1 & \dfrac{\Delta s}{\gamma^2} \\ 0 & 1 \end{bmatrix}.$$

Bending magnet (*Δα, |ρ|, n, ouv, HV*)

Δα (°) is the rotation angle, *ρ (mm)* is the curvature radius, *n* is the field index, *ouv(mm)* the aperture and *HV* means: horizontal (=0) or vertical bend (=1).
A positive bend (*α>0*) bends the particles to the right in the horizontal plane, regardless of the sign of the particle charge state. A negative α bends particles to the left.

$$h = \frac{1}{|\rho|}\frac{\Delta\alpha}{|\Delta\alpha|}, \quad k_x = \sqrt{(1-n)\cdot h^2}, \quad k_y = \sqrt{n\cdot h^2}, \ \Delta s = \rho|\Delta\alpha|.$$

The non null 2×2 transfer sub-matrixes are:

$$R_{xx} = \begin{bmatrix} \cos(k_x\Delta s) & \dfrac{\sin(k_x\Delta s)}{k_x} \\ -k_x\sin(k_x\Delta s) & \cos(k_x\Delta s) \end{bmatrix}, R_{yy} = \begin{bmatrix} \cos(k_y\Delta s) & \dfrac{\sin(k_y\Delta s)}{k_y} \\ -k_y\sin(k_y\Delta s) & \cos(k_y\Delta s) \end{bmatrix},$$

$$R_{zz} = \begin{bmatrix} 1 & \dfrac{-h^2(k_x\Delta s\beta^2 - \sin(k_x\Delta s))}{k_x^3} + \dfrac{\Delta s}{\gamma^2}(1-\dfrac{h^2}{k_x^2}) \\ 0 & 1 \end{bmatrix},$$

$$R_{zx} = \begin{bmatrix} \dfrac{-h\sin(k_x\Delta s)}{k_x} & \dfrac{-h(1-\cos k_x\Delta s))}{k_x^2} \\ 0 & 0 \end{bmatrix} \cdot R_{xz} = \begin{bmatrix} 0 & \dfrac{h(1-\cos k_x\Delta s))}{k_x^2} \\ 0 & \dfrac{h\sin(k_x\Delta s)}{k_x} \end{bmatrix}$$

Edge angle on bending magnet (*β* , *|ρ|, g, K1, K2, ouv, HV*)

β (°) is the edge angle, *ρ (mm)* is the curvature radius in the bending magnet, *g (mm)* is the gap between the poles of the bending magnet, *K1* and *K2* are used in a development for the fringe-field correction. If they are equal to zero, *K1* = 0.45 and *K2* = 2.8. Set small values to cancel fringe-field correction.
ouv (mm) is the aperture and *HV* means: horizontal (=0) or vertical bend (=1).
The edge angle is treated as a thin lens. *Ψ* is the fringe-field correction.

$$\Psi = K_1 \frac{g}{|\rho|}\left(\frac{1+\sin^2(\beta)}{\cos(\beta)}\right)\cdot\left(1-K_1 K_2 \frac{g}{|\rho|}\tan(\beta)\right).$$

The non null 2×2 transfer sub-matrixes are:

$$R_{xx} = \begin{bmatrix} 1 & 0 \\ \dfrac{\tan(\beta)}{|\rho|} & 1 \end{bmatrix}, \quad R_{yy} = \begin{bmatrix} 1 & 0 \\ \dfrac{-\tan(\beta-\Psi)}{|\rho|} & 1 \end{bmatrix} \text{ And } \quad R_{zz} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Thin steering (*BLx or ELx, Bly or ELy, r, Elec_flag*)

$$x' = x' - \frac{qBL_y \cdot c}{mc^2\beta\gamma} \quad \text{And} \quad y' = y' + \frac{qBL_x \cdot c}{mc^2\beta\gamma}$$

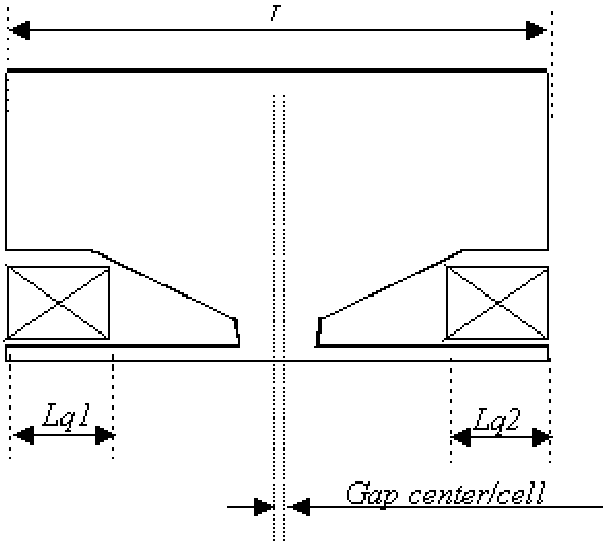
If *Elec_flag* equal 1

$$x' = x' + \frac{EL_x}{E\rho} \quad \text{and} \quad y' = y' + \frac{\cdot EL_y}{E\rho}$$

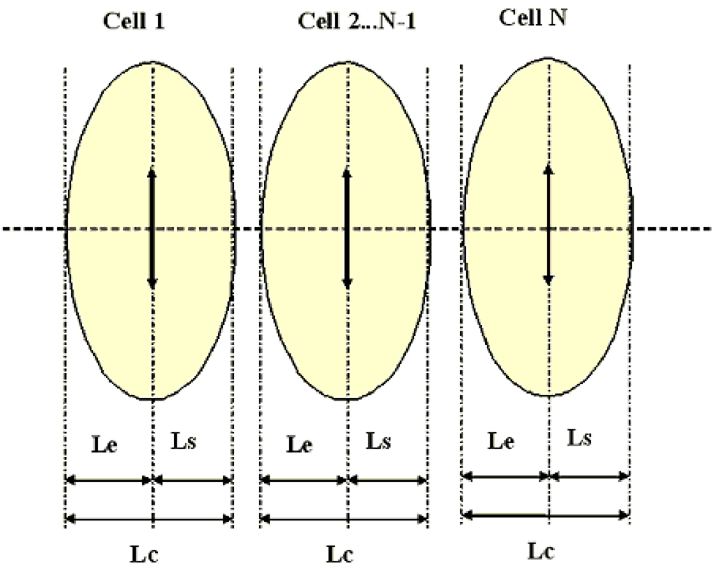
Where *x'* and *y'* being respectively the horizontal and vertical beam centroid slope.

DTL cell (*L, Lq1, Lq2, g$_c$, B1', B2', EoTL, θs, r, p, βs, Ts, kT's,k^{2}T''s*)

The dimension *g$_c$(mm)* is defined as: *gap position* $= \frac{L}{2} - g_c$.


l
Lq1
Lq2
Gap center/cell

Cavity multi-gap (*m, N, βg, EoT, θs, r, p, kEoTi, kEoTo, Dzi, Dzo, βs, Ts, kT's,k^{2}T''i, Ts, kT'i,k^{2}T''i, To, kT'o,k^{2}T''o*)



If m=0:
For cell1 : Le = ½ βλ +dzi, Ls = ½ βλ-dzi, Lc = βλ, EoT(1) = EoT (1+kEoTi).(Ti/Ts)
For cell2 to N-1 : Le = Ls = ½ βλ, Lc = βλ, EoT(2..N-1) = EoT
For cellN : Le = ½ βλ +dzo, Ls = ½ βλ - dzo, Lc = βλ, EoT(N) = EoT.(1+kEoTo).(To/Ts)

If m=1:
For cell1 : Le=1/4 βλ +dzi, Ls=1/4 βλ-dzi, Lc= ½ βλ, EoT(1)= EoT.(1+kEoTi).(Ti/Ts)
For cell2 to N-1: Le=Ls=1/4 βλ, Lc = ½ βλ, EoT(2..N-1)= EoT.(1+kEoTi).(Ti/Ts)
For cellN : Le=1/4 βλ+dzo,Ls=1/4 βλ - dzo, Lc= ½ βλ, EoT(N)= EoT.(1+kEoTo).(To/Ts)

If m=2:
For cell1 : Le=1/4 βλ+dzi + dzi, Ls=½ βλ-dzi, Lc= ¾ βλ, EoT(1)= EoT.(1+kEoTi).(Ti/Ts)
For cell2 to N-1: Le=Ls= ½ βλ, Lc = βλ, EoT(2..N-1)= EoT.(1+kEoTi).(Ti/Ts)
For cellN : Le=½ βλ +dz0, Ls=1/4 βλ - dzo, Lc= ¾ βλ, EoT(N)= EoT.(1+kEoTo).(To/Ts)

For all cases : EoTL= EoT(x).(Ls + Le)

Electrostatic Acceleration (*Vo, Δs, K*)

Vo (V) is the voltage, *Δs* (mm) is the step length, *L* (mm) is the element length, and *K* (eV/mm²) is the transverse defocalisation contribution.
The reduced energy change in the gap is:

$$\gamma_s = \gamma_e + \frac{|q|\frac{V_0}{L}\Delta_s}{mc^2}.$$

Let's use:

$$K* = \frac{qK\Delta_s}{m_0 c^2 \overline{\beta}^2 \overline{\gamma}} \text{ and } \delta = \sqrt{\frac{(\beta\gamma)_e}{(\beta\gamma)_s}} \text{ with } \overline{\gamma} = \frac{\gamma_e + \gamma_s}{2} \text{ and } \overline{\beta} = \frac{\beta_e + \beta_s}{2}.$$

The non null 2×2 transfer sub-matrixes are:

$$R_{xx} = R_{yy} = \begin{bmatrix} 1 - K^* \frac{\Delta_s}{2} & \frac{\Delta_s}{2}\left(1 - K^* \frac{\Delta_s}{2} + \delta\right) \\ -K^* \delta & -K^* \delta \frac{\Delta_s}{2} + \delta^2 \end{bmatrix}$$

$$R_{zz} = \begin{bmatrix} 1 & \frac{\Delta_s}{\gamma_s^2} \\ 0 & \delta^2 \end{bmatrix}$$

Correction: $R_{zz} = \begin{bmatrix} \frac{1}{\delta^2} & \frac{\Delta_s}{\gamma_s^2} \\ 0 & \delta^4 \end{bmatrix}$ have to be validated

Sinus cavity or CCL (*L*, *N*, *EoT*, *θs*)

L (mm) is the cavity length, *N* is the number of cells, *E₀T (eV/m)* is the mean electric field of the cavity, *θs (°)* is the phase of the synchronous particle at the entrance of the cavity (relative to the R.F. phase).

Fields

Let's assume the electric field on the axis:

$$E_z(z,t) = E_0 \sin(\omega t + \phi_0) \sin\left(\frac{K}{\beta_c} z\right), \qquad \text{With } \beta_c = \frac{2L}{N\lambda}.$$

The transverse electric field component can be deduced from Maxwell equations with a first order expansion:

$$E_r(z,t,r) = -\frac{r}{2}\frac{dE_z}{dz} = -\frac{KE_0}{2\beta_c} \sin(\omega t + \phi_0) \cos\left(\frac{K}{\beta_c} z\right) \cdot r,$$

$$\Rightarrow \quad \begin{cases} E_x(z,t,x) = -\frac{KE_0}{2\beta_c} \sin(\omega t + \phi_0) \cos\left(\frac{K}{\beta_c} z\right) \cdot x \\ E_y(z,t,y) = -\frac{KE_0}{2\beta_c} \sin(\omega t + \phi_0) \cos\left(\frac{K}{\beta_c} z\right) \cdot y \end{cases}$$

The same way, the transverse magnetic field component can be deduced:

$$B_\theta(z,t,r) = \frac{r}{2c^2}\frac{dE_z}{dt} = \frac{KE_0}{2c}\cos(\omega t + \phi_0)\sin\left(\frac{K}{\beta_c}z\right)\cdot r\,,$$

$$\Rightarrow \quad \begin{cases} B_x(z,t,y) = -\dfrac{KE_0}{2c}\cos(\omega t + \phi_0)\sin\left(\dfrac{K}{\beta_c}z\right)\cdot y \\ B_y(z,t,x) = \dfrac{KE_0}{2c}\cos(\omega t + \phi_0)\sin\left(\dfrac{K}{\beta_c}z\right)\cdot x \end{cases}$$

With $K = \frac{2\pi}{\lambda} = \frac{\omega}{c}$, $\omega = 2\pi f$, f the R.F frequency and c the speed of light in vacuum.

Longitudinal motion

Let's p_s be the synchronous particle momentum at a given z, its evolution is given by:

$$\dot{p}_s = qE_z(z,t_s) = qE_{zs}\,.$$

The particle momentum is: $p = \sqrt{p_x^2 + p_y^2 + p_z^2} = p_z\sqrt{1 + x'^2 + y'^2}$,

$$\text{With} \quad \begin{cases} \dot{p}_x = q(E_x - v_z B_y + v_y B_z) \\ \dot{p}_y = q(E_y + v_z B_x - v_x B_z) \\ \dot{p}_z = q(E_z + v_x B_y - v_y B_x) \end{cases}$$

The evolution with time of the particle momentum p is given by:

$$\dot{p} = \frac{p_z}{p}\left(\dot{p}_z + x'\dot{p}_x + y'\dot{p}_y\right).$$

Let's δ be the momentum of the particle relative to that of the synchronous particle:

$$\delta = \frac{p - p_s}{p_s}\,.$$

We have: $p = (1+\delta)p_s = p_z\sqrt{1 + x'^2 + y'^2}$.

The evolution of δ is given by the equation:

$$\dot{\delta} = \frac{1}{p_s^2}\left(p_s\dot{p} - \dot{p}_s p\right),$$

$$\dot{\delta} = \frac{1}{p_s^2}\left(\frac{p_s p_z}{p}\left(\dot{p}_z + x'\dot{p}_x + y'\dot{p}_y\right) - \dot{p}_s p\right),$$

$$\dot{\delta} = \frac{q}{p_s}\left[\left(1 + x'^2 + y'^2\right)^{1/2}\cdot\left(E_z + x'E_x + y'E_y\right) - (1+\delta)E_{zs}\right].$$

With $E_z = E_{zs} + \left.\frac{dE_z}{d\varphi}\right]_{\varphi=\varphi_s} \cdot\phi$, $\phi = \varphi - \varphi_s$ and $\frac{d}{dz} = \frac{1}{\beta_s c}\cdot\frac{d}{dt}$,

We finally find at first order:

$$\boxed{\frac{d\delta}{dz} = \delta' = \frac{q\,E_0}{\gamma_s\,\beta_s^2\,mc^2}\sin\left(\frac{K\,z}{\beta_c}\right)\left[\cos\left(\omega t_s+\phi_0\right)\cdot\varphi - \sin\left(\omega t_s+\phi_0\right)\cdot\delta\right]}.$$

Both focusing and damping effects can be observed.

Thin lens approximation

Using: $\phi = -\frac{K}{\beta_s}\cdot\delta z$ the matrix transport can then be written over a small step *dz*:

$$\begin{pmatrix}\delta z\\ \delta\end{pmatrix}_{z+dz/2} = \begin{pmatrix}1 & dz/2\gamma_I^{*2}\\ 0 & 1\end{pmatrix}\begin{pmatrix}1 & 0\\ K_1 & K_2\end{pmatrix}\begin{pmatrix}1 & dz/2\gamma_O^{*2}\\ 0 & 1\end{pmatrix}\cdot\begin{pmatrix}\delta z\\ \delta\end{pmatrix}_{z-dz/2} = M_z\cdot\begin{pmatrix}\delta z\\ \delta\end{pmatrix}_{z-dz/2},$$

With: γs and γe being the synchronous normalized energy before and after the "gap",

$$\boxed{K_1 = -\frac{qE_0\,K}{\gamma_s\,\beta_s^3\,mc^2}\sin\left(\frac{K\,z}{\beta_c}\right)\cos\left(\omega t_s+\varphi_0\right)}$$ And

$$\boxed{K_2 = 1 - \frac{qE_0}{\gamma_s\,\beta_s^2\,mc^2}\sin\left(\frac{K\,z}{\beta_c}\right)\sin\left(\omega t_s+\varphi_0\right)}.$$

Transverse motion

The evolution of x' with time is driven by the equation:

$$\dot{x}' = \frac{d}{dt}\left(\frac{p_x}{p_z}\right) = \frac{1}{p_z^2}\left(\dot{p}_x\,p_z - \dot{p}_z\,p_x\right)$$

$$\dot{x}' = \frac{q}{p_s}\cdot\frac{\left(1+x'^2+y'^2\right)^{1/2}}{1+\delta}\cdot\left(E_x - v_z B_y - x'\left(E_z + v_x B_y - v_y B_x\right)\right).$$

At first order, we finally have:

$$\boxed{x'' = -\frac{q\,E_0}{\gamma_s\,\beta_s^2\,mc^2}\left[\begin{array}{l}\frac{K}{2}\left(\frac{1}{\beta_c}\sin\left(\omega t_s+\phi_0\right)\cos\left(\frac{K\,z}{\beta_c}\right)+\beta_s\cos\left(\omega t_s+\phi_0\right)\sin\left(\frac{K\,z}{\beta_c}\right)\right)\cdot x\\ +\sin\left(\omega t_s+\phi_0\right)\sin\left(\frac{K\,z}{\beta_c}\right)\cdot x'\end{array}\right]}.$$

Both focusing and damping effects can be observed.

Thin lens approximation

The matrix transport over a small step *dz* can then be written:

$$\begin{pmatrix} x \\ x' \end{pmatrix}_{z+dz/2} = \begin{pmatrix} 1 & dz/2 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} 1 & 0 \\ k_1 & k_2 \end{pmatrix}\begin{pmatrix} 1 & dz/2 \\ 0 & 1 \end{pmatrix}\cdot\begin{pmatrix} x \\ x' \end{pmatrix}_{z-dz/2} = M_x\cdot\begin{pmatrix} x \\ x' \end{pmatrix}_{z-dz/2},$$

With: $\boxed{k_1 = -\frac{qE_0K}{2\gamma_s\beta_s^2 mc^2}\left[\frac{1}{\beta_c}\sin(\omega t_s+\varphi_0)\cos\left(\frac{Kz}{\beta_c}\right)+\beta_s\cos(\omega t_s+\varphi_0)\sin\left(\frac{Kz}{\beta_c}\right)\right]}$,

And $\boxed{k_2 = K_2}$.

Thick lens approximation

The differential equation can be written:

$$x'' + 2\cdot\alpha\cdot x' + \beta\cdot x = 0,$$

With: $\beta = \frac{qE_0K}{2\gamma_s\beta_s^2 mc^2}\left[\frac{1}{\beta_c}\sin(\omega t_s+\varphi_0)\cos\left(\frac{Kz}{\beta_c}\right)+\beta_s\cos(\omega t_s+\varphi_0)\sin\left(\frac{Kz}{\beta_c}\right)\right]$

And: $\alpha = \frac{qE_0}{2\gamma_s\beta_s^2 mc^2}\cdot\sin(\omega t_s+\varphi_0)\cdot\sin\left(\frac{Kz}{\beta_c}\right)$,

And: $\omega = \sqrt{\beta-\alpha^2}$,

The solution of this differential equation gives:

$$\boxed{M_x = e^{-\alpha\cdot dz}\cdot\begin{pmatrix} \cos(\omega dz)+\frac{\alpha}{\omega}\cdot\sin(\omega dz) & \frac{\sin(\omega dz)}{\omega} \\ -\frac{\beta\sin(\omega dz)}{\omega} & \cos(\omega dz)-\frac{\alpha}{\omega}\cdot\sin(\omega dz) \end{pmatrix}}$$

, If ω is real,

And

$$\boxed{M_x = e^{-\alpha\cdot dz}\cdot\begin{pmatrix} ch(|\omega| dz)+\frac{\alpha}{|\omega|}\cdot sh(|\omega| dz) & \frac{sh(|\omega| dz)}{|\omega|} \\ \frac{\beta\, sh(|\omega| dz)}{|\omega|} & ch(|\omega| dz)-\frac{\alpha}{|\omega|}\cdot sh(|\omega| dz) \end{pmatrix}}$$

, If $j\ \omega$ is real.

Transport through a sin-like cavity

The *Nc*-cells cavity is divided in n·*Nc* steps of length: $dz = \frac{\beta_c \lambda}{2n}$.

As input, we have:

γo = γin; $t_s = \frac{dz}{2\beta_{in} c}$; $z_s = \frac{dz}{2}$; $\begin{pmatrix} x \\ x' \end{pmatrix}_0$, $\begin{pmatrix} \delta z \\ \delta \end{pmatrix}_0$: Particle co-ordinates at cavity input.

Then, we loop until reaching the end of the cavity:
For i from 0 to n·Nc-1 do

$$\gamma_I^* = \gamma_O^*,\ \gamma_O^* = \gamma_{i+1} = \gamma_i + \frac{qE_0}{mc^2}\sin(\omega t_s + \varphi_0)\sin\left(\frac{K}{\beta_c}\cdot z_s\right)\cdot dz,\ \ \beta_{i+1} = \sqrt{1-\gamma_{i+1}^{-2}},$$

$$\gamma_s = \frac{\gamma_{i+1} + \gamma_i}{2},\ \ \beta_s = \sqrt{1-\gamma_s^{-2}},$$

$$\begin{pmatrix} x \\ x' \end{pmatrix}_{i+1} = M_x \cdot \begin{pmatrix} x \\ x' \end{pmatrix}_i,\ \ \begin{pmatrix} \delta z \\ \delta \end{pmatrix}_{i+1} = M_z \cdot \begin{pmatrix} \delta z \\ \delta \end{pmatrix}_i,$$

$$t_s = t_s + \frac{dz}{\beta_{i+1} c},\ \ z_s = z_s + dz.$$

RFQ cell (*V, ro, A10, m, L, θs, Type, Tc, dP*)

L (mm) is the RFQ cell length, *V(V)* is the mean voltage of the cell, *θs(°)* is the phase of the synchronous particle, *ro* is the vane radius, *m* is the modulation, *Tc* is the transverse curvature and dP is a phase shift defined only *dP* which is a phase shift allowing to reset the output phases of Toutatis who does not own phase reference, *Type* parameter is defined bellow.

Cell *type*:

±2: Accelerating cell.
±3: Front-end cell.
±4: Transcell.

The sign of type being…..

$W_{i+1} = W_i + |q| dzE_z$ And $\overline{W} = \frac{1}{2}(W_i + W_{i+1})$

$$\Phi_{i+1} = \Phi_i + dz\frac{2\pi}{\overline{\beta}\lambda}$$

$$E_z = \frac{\pi A_{10} V}{2L} \sin\left(\frac{\pi}{L} z\right) \sin(\omega t_s + \varphi_0)$$

Thin lens approximation (Longitudinal)

$\omega = 2\pi f$, f the R.F frequency and *c* the speed of light in vacuum.
The matrix transport can then be written over a small step *dz*:

$$\begin{pmatrix} \delta z \\ \delta \end{pmatrix}_{z+dz/2} = \begin{pmatrix} 1 & dz/2\gamma_o^{*2} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ K_1 & K_2 \end{pmatrix} \begin{pmatrix} 1 & dz/2\gamma_i^{*2} \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \delta z \\ \delta \end{pmatrix}_{z-dz/2} = M_z \cdot \begin{pmatrix} \delta z \\ \delta \end{pmatrix}_{z-dz/2},$$

With: γs and γe being the synchronous normalized energy before and after the "gap",

$$\boxed{K_1 = -\frac{|q| d_z A_{10} V}{\gamma_s \beta_s^2 2mc^2} \left(\frac{\pi}{L}\right)^2 \cdot C_3 \cos(\omega t_s + \varphi_0)}$$

$$\boxed{K_2 = 1 - \frac{|q| d_z A_{10} V}{\gamma_s \beta_s^2 2mc^2} \left(\frac{\pi}{L}\right) \cdot C_3 \cos(\omega t_s + \varphi_0)}$$

*C*₃ depend of the cell *type*.

±2 or ±3: $C_3 = \sin\left(\frac{\pi}{L} z\right)$

±4: $C_3 = \frac{1}{2} \sin\left(\frac{\pi}{L} z\right)$

Thin lens approximation (Transverse)

The matrix transport over a small step *dz* can then be written:

$$\begin{pmatrix} x \\ x' \end{pmatrix}_{z+dz/2} = \begin{pmatrix} 1 & dz/2 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ k_{x1} & k_{x2} \end{pmatrix} \begin{pmatrix} 1 & dz/2 \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ x' \end{pmatrix}_{z-dz/2} = M_x \cdot \begin{pmatrix} x \\ x' \end{pmatrix}_{z-dz/2},$$

$$\begin{pmatrix} y \\ y \end{pmatrix}_{z+dz/2} = \begin{pmatrix} 1 & dz/2 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ k_{y1} & k_{y2} \end{pmatrix} \begin{pmatrix} 1 & dz/2 \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} y \\ yx \end{pmatrix}_{z-dz/2} = M_y \cdot \begin{pmatrix} y \\ y' \end{pmatrix}_{z-dz/2}$$

With: $\boxed{k_{x1} = -\frac{|q| d_z}{\gamma_s \beta_s^2 2mc^2} \cos(\omega t_s + \varphi_0) \left(S \frac{V}{r_0^2} A_{01} C_1 - \left(\frac{\pi}{L}\right)^2 \frac{A_{10} V}{4} C_2 \right)}$,

$$\boxed{k_{y1} = -\frac{|q| d_z}{\gamma_s \beta_s^2 2mc^2} \cos(\omega t_s + \varphi_0) \left(-S \frac{V}{r_0^2} A_{01} C_1 - \left(\frac{\pi}{L}\right)^2 \frac{A_{10} V}{4} C_2 \right)}$$

And $\boxed{k_{x2} = k_{y2} = K_2}$.

$$\begin{pmatrix} x \\ x' \end{pmatrix}_{i+1} = M_x \cdot \begin{pmatrix} x \\ x' \end{pmatrix}_i \text{ And } \begin{pmatrix} y \\ y' \end{pmatrix}_{i+1} = M_y \cdot \begin{pmatrix} y \\ y' \end{pmatrix}_i$$

C_1 , C_2 and S depend of the cell *type*.

±2: $C_1 = 1, C_2 = \sin\left(\frac{\pi}{L} z\right)$, $S = -sign(type)$

+3: $C_1 = \frac{1}{4}\left(3\cos(\frac{1}{2}\frac{\pi}{L}z - \frac{\pi}{2}) + \cos\left(3 \cdot \left(\frac{1}{2}\frac{\pi}{L}z - \frac{\pi}{2}\right)\right)\right), C_2 = 0,$

$S = -sign(type[n+1])$

-3: $C_1 = \frac{3}{4}\left(\cos(\frac{1}{2}\frac{\pi}{L}z) - \frac{1}{3}\cos(\frac{3}{2}\frac{\pi}{L}z)\right)$, $C_2 = 0$, $S = -sign(type[n-1])$

+4: $C_1 = 1, C_2 = \frac{1}{2}\left(\cos\left(\frac{\pi}{L}z\right) + 1\right)$, $S = -sign(type[n+1])$

-4: $C_1 = 1, C_2 = -\frac{1}{2}\left(\cos\left(\frac{\pi}{L}z\right) - 1\right)$, $S = -sign(type[n-1])$

With *type[n+1]* being the type from the next cell and *type[n-1]* the type from preceding cell.

Transport through a RFQ cell

The rfq cell is divided in ·N steps of length: $dz = \frac{L}{N}$.

As input, we have:

$\gamma_0 = \gamma_{in}$; $t_s = \frac{dz}{2\beta_{in}c}$; $z_s = \frac{dz}{2}$; $\begin{pmatrix} x \\ x' \end{pmatrix}_0$, $\begin{pmatrix} \delta z \\ \delta \end{pmatrix}_0$: Particle co-ordinates at cavity input.

Then, we loop until reaching the end of the cavity:
For i from 0 to N-1 do

$$\gamma_I^* = \gamma_O^*,\ \gamma_O^* = \gamma_{i+1} = \gamma_i + \frac{|q|\pi AV}{2Lmc^2}\sin(\omega t_s + \varphi_0)\sin\left(\frac{\pi}{L} \cdot z_s\right) \cdot dz,\ \beta_{i+1} = \sqrt{1 - \gamma_{i+1}^{-2}},$$

$$\gamma_s = \frac{\gamma_{i+1} + \gamma_i}{2},\ \beta_s = \sqrt{1 - \gamma_s^{-2}},$$

$$\begin{pmatrix} x \\ x' \end{pmatrix}_{i+1} = M_x \cdot \begin{pmatrix} x \\ x' \end{pmatrix}_i,\ \begin{pmatrix} \delta z \\ \delta \end{pmatrix}_{i+1} = M_z \cdot \begin{pmatrix} \delta z \\ \delta \end{pmatrix}_i,$$

$$t_s = t_s + \frac{dz}{\beta_{i+1} c}, \quad z_s = z_s + dz\,.$$

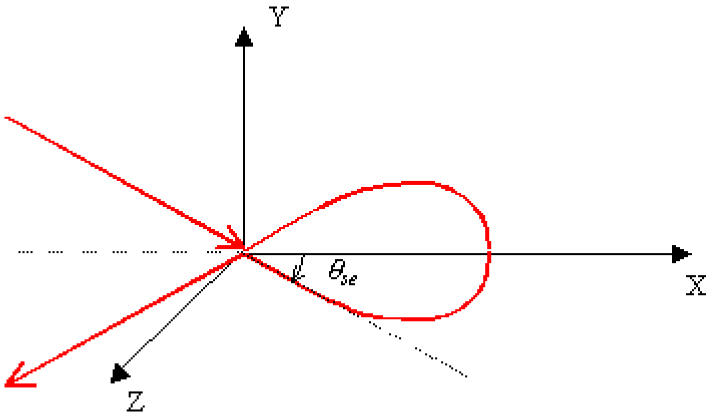
Transfer matrix of alpha magnet (*Θ, K, R, plan*)

The following paper come from a CEA report named:
“Transfer matrix of a constant gradient alpha magnet for ELSA extension”
Ref: DSM/DAPNIA/SACM/2002/13
N. Pichoff

In this paper, the motion equation of a particle in such a device is linearised to get the transfer matrix. The matrix has been validated with a step-by-step integration of the motion of a particle in the magnetic field.

Magnetic field and fixed coordinate system

The trajectory of the synchronous particle (ideal trajectory) is, by definition, in the plan (X, Y). The frame origin is the point where the particle enters the magnet. The X direction is the direction of the magnetic field gradient. At any position, the main trajectory makes an angle θ_s with the X axis. The entrance angle for negative particle θ_{se} is: -40.71°. With this particular angle, the synchronous particle exits the magnet at the same position as the entering particle (and with the opposite angle).



In this frame, the magnetic field is:

$$\vec{B} = \begin{pmatrix} B_X & = k \cdot Z \\ B_Y & = 0 \\ B_Z & = k \cdot X \end{pmatrix}$$

Trajectory of the synchronous particle

The synchronous particle with particle charge *q* state moves in the (X, Y) plan.
Its motion equations with time *t* are:

$$\begin{cases} \dfrac{dp_X}{dt} = q \cdot v_Y \cdot B_Z \\ \dfrac{dp_Y}{dt} = -q \cdot v_X \cdot B_Z \end{cases}$$

p is the particle momentum, *v* its velocity,

giving:

$$\begin{cases} \dfrac{dp_X}{ds} = q \cdot \dfrac{p_Y}{p_0} \cdot k \cdot X \\ \dfrac{dp_Y}{ds} = -q \cdot \dfrac{p_X}{p_0} \cdot k \cdot X \\ \dfrac{dX}{ds} = \dfrac{p_X}{p_0} \\ \theta_s = \arcsin\left(\dfrac{p_Y}{p_0}\right) = \arccos\left(\dfrac{p_X}{p_0}\right) \end{cases}$$

$p_0 = \beta_0\gamma_0 m_0 c$ is the particle momentum modulus,

s is the curvilign abscissa, with $ds = v_0 \cdot dt$.

These equations have to be solved using as initial conditions:

$$\begin{cases} p_X = p_0 \cos\theta_{se} \\ p_Y = p_0 \sin\theta_{se} \\ X = 0 \\ \theta_s = \theta_{se} \end{cases}$$

The maximum *penetration of the particle X_M* in the magnet can be calculated the following way:

$$\frac{dp_Y}{dX} = -q \cdot k \cdot X \qquad \Rightarrow \qquad p_Y = p_{Y0} - \frac{1}{2} \cdot q \cdot k \cdot X^2.$$

At maximum penetration, one has: $p_Y = p_0$, giving:

$$X_M = \sqrt{\frac{2 \cdot (p_{Y0} - p_0)}{q \cdot k}}.$$

The length *L* of the trajectory is obtained from the integration of:

$$dt = \frac{\gamma_0 m_0 \cdot dX}{p_X} .$$

One obtains:

$$L = 2 \cdot p_0 \cdot \int_0^{X_M} \left(p_{X0}^2 + q \cdot k \cdot p_{Y0} \cdot X^2 - 0.25 \cdot q^2 k^2 \cdot X^4\right)^{-\frac{1}{2}} \cdot dX \cdot$$

This integral cannot be solved analytically but can be easily calculated numerically.

The moving coordinate system

In the transfer matrix formalism, a beam particle is referenced, at a given curvilign abscissa ***s***, to the synchronous particle in a 6D phase-space, with a 6-coordinates vector:

$$\vec{r} = \begin{pmatrix} x \\ x' \\ y \\ y' \\ \varphi \\ \delta \end{pmatrix},$$

with:

- ***x*** is the particle transverse position in the deviation plan,

- ***x’*** is the particle transverse slope in the x direction. $x' = \frac{p_x}{p_s} = \frac{dx}{ds}$,

p

$\boldsymbol{p_x}$ the x-component of the particle momentum, and $\boldsymbol{p_s}$ the s-component of the particle momentum,

- **y** is the particle transverse position orthogonal to the x direction,

- ***y’*** is the particle transverse slope in the y direction. $y' = \frac{p_y}{p_s} = \frac{dy}{ds}$,

$\boldsymbol{p_y}$ the y-component of the particle momentum,

- *φ* is the time difference between the particle arriving in *s* (***t***) and the synchronous particle arriving in s ($\boldsymbol{t^{(S)}}$). This time is normalized with the RF frequency $\boldsymbol{f_{RF}}$.

$$\varphi = 2\pi \cdot f_{RF} \cdot \left(t - t^{(S)}\right),$$

- *δ* is the particle momentum ***p*** relative to the synchronous particle momentum $\boldsymbol{p^{(S)}}$.

$$\delta = \frac{p - p^{(S)}}{p^{(S)}} .$$

Matrix calculation

In the matrix formalism, the particle vector change from a point (1) to a point (2) is given by:

$$\left(\vec{r}\right)_{(2)} = \left[T\right]\cdot\left(\vec{r}\right)_{(1)},$$

where [*T*] is the transfer matrix from (1) to (2). The coefficients of [*T*] are:

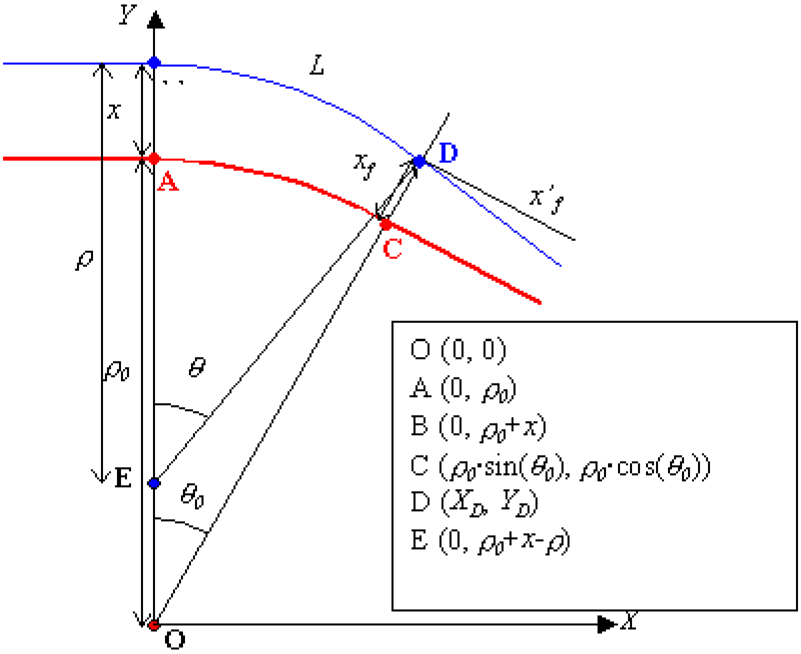
$$T_{i,j} = \frac{\partial r_i\big)_{(2)}}{\partial r_j\big)_{(1)}}.$$

i is the line index, *j* is the column index (between 1 and 6).

The matrix coefficients can be calculated by varying the input particle coordinate along each direction independently and looking at the output coordinates.

Matrix first column
Matrix second column
Matrix sixth column
Matrix fifth column
Matrix third and fourth columns

Matrix first column: variation with x



Calculus of $T_{1,1}$

D is at the intersection of (O, C) and the circle with centre E and radius ρ. Its coordinates satisfy:

$$\begin{cases} X_D = \tan(\theta_0)\cdot Y_D \\ X_D^2 + \left(Y_D - (\rho_0 + x - \rho)\right)^2 = \rho^2 \end{cases}.$$

Giving: $\left(1+\tan^2\theta_0\right)\cdot Y_D^2 - 2\cdot(x - d\rho)\cdot Y_D + (\rho_0 + x)\cdot(x - 2\cdot d\rho - \rho_0) = 0$,

with: $\rho = \rho_0 + d\rho = \rho_0 \cdot \left(1 + \dfrac{d\rho}{\rho_0}\right)$.

As: $\rho = \dfrac{p}{q\cdot B}$, one has: $\dfrac{d\rho}{\rho_0} = -\dfrac{dB}{B_0} = \sin\theta_s \cdot \dfrac{x}{X_s}$.

The solution of (X) is obtained giving a reduced discriminator:

$$\Delta' = (x - d\rho)^2 - \frac{(\rho_0 + x)\cdot(x - d\rho - \rho_0)}{\cos^2\theta_0},$$

giving:

$$\Delta' = \frac{\rho_0^2}{\cos^2\theta_0}\cdot\left[\left(\frac{x}{\rho_0} - \frac{d\rho}{\rho_0}\right)^2\cdot\cos^2\theta_0 - \left(1 + \frac{x}{\rho_0}\right)\cdot\left(\frac{x}{\rho_0} - 2\cdot\frac{d\rho}{\rho_0} - 1\right)\right].$$

A first order development in $\dfrac{x}{\rho_0} << 1$ and $\dfrac{d\rho}{\rho_0} << 1$ gives:

$$\Delta' = \frac{\rho_0^2}{\cos^2\theta_0}\cdot\left[1 + 2\cdot\frac{d\rho}{\rho_0}\right].$$

This gives the coordinates if D:

$$\begin{cases} X_D = \rho_0\cdot\sin\theta_0\cdot\left(1 + \cos\theta_0\cdot\dfrac{x}{\rho_0} + (1 - \cos\theta_0)\cdot\dfrac{d\rho}{\rho_0}\right) \\ Y_D = \rho_0\cdot\cos\theta_0\cdot\left(1 + \cos\theta_0\cdot\dfrac{x}{\rho_0} + (1 - \cos\theta_0)\cdot\dfrac{d\rho}{\rho_0}\right) \end{cases}$$

The final position of the particle in the moving frame is then:

$$\boxed{x_f = \frac{Y_D - Y_C}{\cos\theta_0} = \left(\cos\theta_0 + (1 - \cos\theta_0)\cdot\frac{\rho_0\cdot\sin\theta_s}{X_s}\right)\cdot x = T_{1,1}\cdot x}$$

Calculus of $T_{2,1}$

$$x'_f = \tan(\theta_0 - \theta) = \frac{\tan\theta_0 - \tan\theta}{1+\tan\theta_0 \cdot \tan\theta},$$

with: $\tan\theta = \dfrac{X_D - X_E}{Y_D - Y_E}$.

At first order, one has:

$$\tan\theta = \tan\theta_0 \cdot \left(1 + \frac{x - d\rho}{\rho_0 \cos\theta_0}\right),$$

giving :

$$\boxed{x'_f = -\sin\theta_0 \cdot \left(1 - \frac{\rho_0 \cdot \sin\theta_s}{X_s}\right) \cdot \frac{x}{\rho_0} = T_{2,1} \cdot x}$$

Calculus of $T_{5,1}$

$$\varphi = \frac{2\pi \cdot f_{RF}}{\beta_0 c}(L - \rho_0\theta_0) = \frac{2\pi \cdot f_{RF}}{\beta_0 c}\rho_0\theta_0 \cdot \left(\frac{d\rho}{\rho_0} + \frac{d\theta}{\theta_0}\right).$$

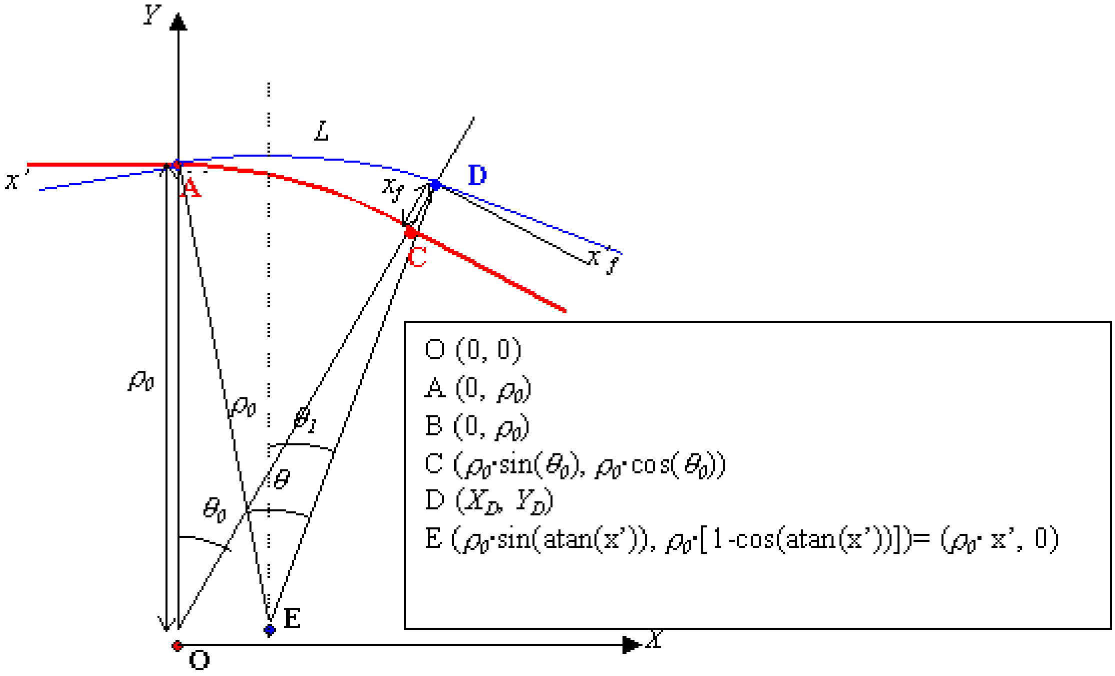
with: $d\theta = \cos^2\theta_0 \cdot d(\tan\theta) = \cos^2\theta_0 \cdot (\tan\theta - \tan\theta_0) = \sin\theta_0 \cdot \left(\dfrac{x - d\rho}{\rho_0}\right).$

One obtains finally:

$$\boxed{\varphi = \frac{2\pi \cdot f_{RF}}{\beta_0 c} \cdot \left(\sin\theta_0 + (\theta_0 - \sin\theta_0) \cdot \frac{\rho_0 \sin\theta_s}{X_s}\right) \cdot x = T_{5,1} \cdot x}.$$

The other terms ($T_{3,1}$, $T_{4,1}$,$T_{6,1}$) are equal to zero.

Matrix second column: variation with x’


Y
L
x'
A
D
x_f
x_f
C
ρ_0
ρ_0
θ_1
θ
θ_0
E
O
X
O (0, 0)
A (0, ρ_0)
B (0, ρ_0)
C (ρ_0·sin(θ_0), ρ_0·cos(θ_0))
D (X_D, Y_D)
E (ρ_0·sin(atan(x')), ρ_0·[1-cos(atan(x'))])= (ρ_0· x', 0)

Calculus of $T_{1,2}$

D is at the intersection of (O, C) and the circle with centre E and radius ρ. Its coordinates satisfy:

$$\begin{cases} X_D = \tan\theta_0 \cdot Y_D \\ \left(X_D - \rho_0 \cdot x'\right)^2 + Y_D^2 = \rho_0^2 \end{cases}.$$

Giving:

$$\left(1+\tan^2\theta_0\right)\cdot Y_D^2 - 2\cdot\rho_0\cdot x'\cdot\tan\theta_0\cdot Y_D - \rho_0^2\cdot\left(1-x'^2\right) = 0\,.$$

The solution of (X) is obtained giving a reduced discriminator:

$$\Delta' = \rho_0^2\cdot\tan^2\theta_0\cdot x'^2 + \frac{\rho_0^2\cdot\left(1-x'^2\right)}{\cos^2\theta_0}\,.$$

A first order development in $x' << 1$ gives:

$$\Delta' = \frac{\rho_0^2}{\cos^2\theta_0}$$

This gives the coordinates of D:

$$\begin{cases} X_D = \rho_0\cdot\sin\theta_0\cdot\left(1+\sin\theta_0\cdot x'\right) \\ Y_D = \rho_0\cdot\cos\theta_0\cdot\left(1+\sin\theta_0\cdot x'\right) \end{cases}$$

The final position of the particle in the moving frame is then:

$$\boxed{x_f = \frac{Y_D - Y_C}{\cos\theta_0} = \rho_0 \cdot \sin\theta_s \cdot x' = T_{1,2} \cdot x'}$$

Calculus of $T_{2,2}$

$$x'_f = \tan(\theta_0 - \theta_1) = \frac{\tan\theta_0 - \tan\theta_1}{1 + \tan\theta_0 \cdot \tan\theta_1}$$

with: $\tan\theta_1 = \dfrac{X_D - X_E}{Y_D - Y_E}$.

At first order, one has:

$$\tan\theta = \frac{X_D - X_E}{Y_D - Y_E},$$

giving : $\boxed{x'_f = \cos\theta_0 \cdot x' = T_{2,2} \cdot x'}$

Calculus of $T_{5,2}$

$$\varphi = \frac{2\pi \cdot f_{RF}}{\beta_0 c}(L - \rho_0\theta_0) = \frac{2\pi \cdot f_{RF}}{\beta_0 c}\rho_0 \cdot d\theta .$$

with : $d\theta = \cos^2\theta_0 \cdot d(\tan\theta) = \cos^2\theta_0 \cdot (\tan\theta - \tan\theta_0),$

as: $\tan\theta = \tan(\theta_1 + \theta_0) = \tan\theta_0 \cdot \left(1 + \dfrac{1 - \cos\theta_0}{\sin\theta_0 \cdot \cos\theta_0} \cdot x'\right),$

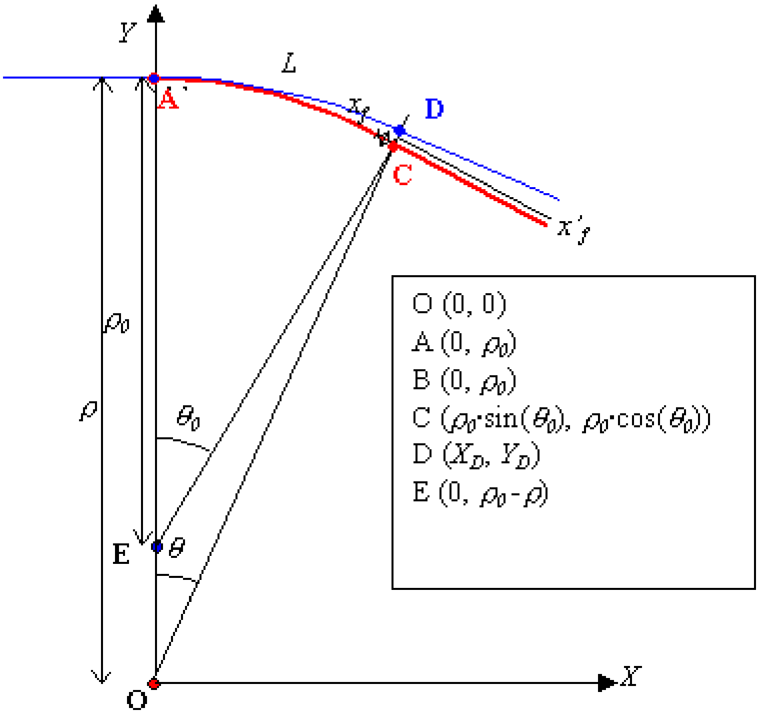
one has: $d\theta = (1 - \cos\theta_0) \cdot x'$.

One obtains finally:

$$\boxed{\varphi = \frac{2\pi \cdot f_{RF}}{\beta_0 c} \cdot \rho_0 \cdot (1 - \cos\theta_0) \cdot x' = T_{5,2} \cdot x'}.$$

The other terms ($T_{3,2}$, $T_{4,2}$,$T_{6,2}$) are equal to zero.

Sixth matrix column: variation with □



Calculus of $T_{1,6}$

D is at the intersection of (O, C) and the circle with center E and radius ρ. Its coordinates satisfy:

$$\begin{cases} X_D = \tan(\theta_0)\cdot Y_D \\ X_D^2 + \left(Y_D - (\rho_0 - \rho)\right)^2 = \rho^2 \end{cases}.$$

Giving:

$$\left(1+\tan^2\theta_0\right)\cdot Y_D^2 + 2\cdot d\rho\cdot Y_D - \rho_0\cdot\left(\rho_0 + 2\cdot d\rho\right) = 0$$

with: $\rho = \rho_0 + d\rho = \rho_0\cdot\left(1+\dfrac{d\rho}{\rho_0}\right).$

As: $\rho = \dfrac{p}{q\cdot B}$, one has: $\dfrac{d\rho}{\rho_0} = \delta$.

The solution of (X) is obtained giving a reduced discriminator:

$$\Delta' = d\rho^2 + \frac{\rho_0\cdot(\rho_0 + 2\cdot d\rho)}{\cos^2\theta_0}.$$

A first order development in $\dfrac{d\rho}{\rho_0} << 1$ gives:

$$\Delta' = \frac{\rho_0^2}{\cos^2\theta_0}\cdot\left[1 + 2\cdot\frac{d\rho}{\rho_0}\right]$$

This gives the coordinates of D:

$$\begin{cases} X_D = \rho_0 \cdot \sin\theta_0 \cdot \left(1 + (1 - \cos\theta_0) \cdot \dfrac{d\rho}{\rho_0}\right) \\ Y_D = \rho_0 \cdot \cos\theta_0 \cdot \left(1 + (1 - \cos\theta_0) \cdot \dfrac{d\rho}{\rho_0}\right) \end{cases}$$

The final position of the particle in the moving frame is then:

$$x_f = \frac{Y_D - Y_C}{\cos\theta_0} = \rho_0 \cdot (1 - \cos\theta_0) \cdot \delta = T_{1,6} \cdot \delta$$

Calcul of $T_{2,6}$

$$x'_f = \tan(\theta_0 - \theta) = \frac{\tan\theta_0 - \tan\theta}{1 + \tan\theta_0 \cdot \tan\theta}$$

With: $\tan\theta = \dfrac{X_D - X_E}{Y_D - Y_E}$.

At first order, one has:

$$\tan\theta = \tan\theta_0 \cdot \left(1 - \frac{d\rho}{\rho_0 \cos\theta_0}\right),$$

Giving:

$$\boxed{x'_f = \sin\theta_0 \cdot \delta = T_{5,5} \cdot \delta}$$

Calcul of $T_{5,6}$

$$\varphi = \frac{2\pi \cdot f_{RF}}{\beta_0 c}(L - \rho_0\theta_0) = \frac{2\pi \cdot f_{RF}}{\beta_0 c}\rho_0\theta_0 \cdot \left(\frac{d\rho}{\rho_0} + \frac{d\theta}{\theta_0}\right).$$

with: $d\theta = \cos^2\theta_0 \cdot d(\tan\theta) = \cos^2\theta_0 \cdot (\tan\theta - \tan\theta_0) = -\sin\theta_0 \cdot \dfrac{d\rho}{\rho_0}$,

One obtains finally:

$$\boxed{\varphi = \frac{2\pi \cdot f_{RF}}{\beta_0 c} \cdot \rho_0\theta_0 \cdot \left(1 - \frac{\sin\theta_0}{\theta_0}\right) \cdot \delta = T_{5,6} \cdot \delta}.$$

The terms ($T_{3,6}$, $T_{4,6}$) are equal to zero, the term $T_{6,6}$ is equal to 1.

Matrix fifth column: variation with φ

The output position, slope, energy do not depend on the input phase φ:
The terms ($T_{1,5}$, $T_{2,5}$, $T_{3,5}$, $T_{4,5}$, $T_{6,5}$) are equal to zero, the term $T_{5,5}$ is equal to 1.

Matrix third and fourth columns: motion in y (or Z)

The equation of motion along Z direction is:

$$\frac{dp_Z}{dt} = -q \cdot \frac{p_y}{\gamma m} \cdot B_X = -q \cdot \frac{p_y}{\gamma m} \cdot k \cdot Z \ .$$

For the matrix calculation, one uses a first order development of the force, giving:

$$\frac{dp_Z}{dt} = -q \cdot \frac{p_0 \cdot \sin\theta_0}{\gamma m} \cdot k \cdot Z \ .$$

This equation is the classical one in a quadrupole with gradient: $k \cdot \sin\theta_0$.
The associated matrix coefficients are:

$$T_{3,3} = T_{4,4} = \cos\left(\sqrt{K}\rho_0\theta_0\right),$$

$$T_{3,4} = \frac{\sin\left(\sqrt{K}\rho_0\theta_0\right)}{\sqrt{K}},$$

$$T_{4,3} = -\sqrt{K} \cdot \sin\left(\sqrt{K}\rho_0\theta_0\right),$$

With: $K = \dfrac{k \cdot \sin\theta_s}{p_0}$.

The other coefficients are equal to 0.

Alpha magnet matrix

The final matrix of a fraction of a alpha magnet (on which, X_s and θ_s are kept almost constant) :

$$\begin{pmatrix} \cos\theta_0+(1-\cos\theta_0)\cdot\dfrac{\rho_0\cdot\sin\theta_s}{X_s} & \rho_0\cdot\sin\theta_s & 0 & 0 & 0 & \rho_0\cdot(1-\cos\theta_0) \\ -\dfrac{\sin\theta_0}{\rho_0}\cdot\left(1-\dfrac{\rho_0\cdot\sin\theta_s}{X_s}\right) & \cos\theta_0 & 0 & 0 & 0 & \sin\theta_0 \\ 0 & 0 & \cos\left(\sqrt{K}\rho_0\theta_0\right) & \dfrac{\sin\left(\sqrt{K}\rho_0\theta_0\right)}{\sqrt{K}} & 0 & 0 \\ 0 & 0 & -\sqrt{K}\cdot\sin\left(\sqrt{K}\rho_0\theta_0\right) & \cos\left(\sqrt{K}\rho_0\theta_0\right) & 0 & 0 \\ K_\varphi\cdot\left(\sin\theta_0+(\theta_0-\sin\theta_0)\cdot\dfrac{\rho_0\sin\theta_s}{X_s}\right) & K_\varphi\cdot\rho_0\cdot(1-\cos\theta_0) & 0 & 0 & 1 & K_\varphi\cdot\rho_0\cdot(\theta_0-\sin\theta_0) \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

With: $K=\dfrac{k\cdot\sin\theta_s}{X_s}$,

And: $K_\varphi=\dfrac{2\pi\cdot f_{RF}}{\beta_0 c}$.

The matrix of the full element is a product of all matrixes for varying X_s and θ_s.

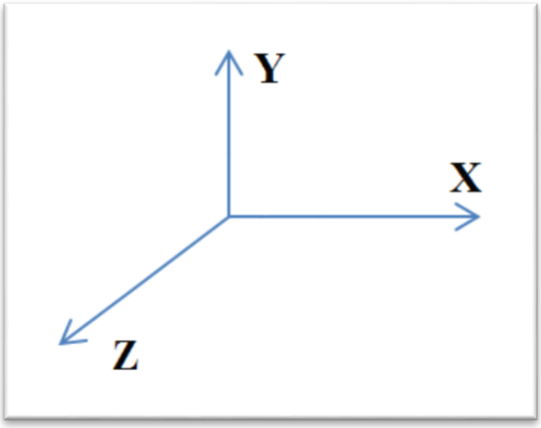
Explanation about the way to obtain this matrix

Dynamics calculations

General description

Frame convention

x, *y*, *z* is in a direct frame.



Usual formulas

$E=\gamma\cdot mc^2=1+W$, $p=\gamma\beta\cdot c$, $v=\beta\cdot c$

$\gamma^{-2}=1-\beta^2$ $\qquad \gamma=1-(\gamma\beta)^2$

$$\frac{dp}{p}=\frac{\gamma}{\gamma+1}\frac{dW}{W}$$

$$\lambda=\frac{c}{f}$$

v, and β are the particle **velocity** and reduced velocity,
mc^2, *E*, *W* and γ are the particle rest mass **energy**, total energy, kinetic energy and reduced energy,
p and $\gamma\beta$ are the particle **momentum** and reduced momentum.
c is the physical maximum velocity (speed of light in vacuum).
λ and *f* are respectively the free-space wavelength and the RF frequency of an electromagnetic field.

Description

In an accelerator, the transport of the beam particles is generally described as a function of the **abscissa *s***, on a **reference trajectory** followed by a **reference particle**.
Each particle is represented by a **6 coordinate vector** whose 3 coordinates represent the **position** and 3 represent the **motion** of the particle in the real space.

Where $\vec{x}(s)$ is a vector representing the particle position in the phase-space: $\vec{x} = \begin{pmatrix} x \\ x' = dx/ds \\ y \\ y' = dy/ds \\ z \\ \delta = \Delta p/p_s \end{pmatrix}$,

In linear forces, the phase-space coordinates of a particle at location s_2 can be deduced from those at the location s_1 along an accelerator, by a single matrix multiplication:

$\vec{x}(s_2) = R \cdot \vec{x}(s_1)$,

x, *y* and *z* being respectively the horizontal, vertical and longitudinal position of the particle in the bunch (relative to a synchronous particle). p_s is the synchronous particle momentum, and $\delta=(p-p_s)/p_s$, with *p* being the particle momentum.

R is the 6×6 transfer matrix between s_1 and s_2. In TraceWin this matrix is partitioned into 2x2 matrices to simplify and accelerate the calculations.

$$R = \left[\begin{array}{cc:cc:cc} a_{00} & a_{01} & a_{02} & a_{03} & a_{04} & a_{05} \\ a_{10} & a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ \hdashline a_{20} & a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\ a_{30} & a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \\ \hdashline a_{40} & a_{41} & a_{42} & a_{43} & a_{44} & a_{45} \\ a_{50} & a_{51} & a_{52} & a_{53} & a_{54} & a_{55} \end{array}\right] = \begin{bmatrix} R_{xx} & R_{xy} & R_{xz} \\ R_{yx} & R_{yy} & R_{yz} \\ R_{zx} & R_{zy} & R_{zz} \end{bmatrix}.$$

In order to be able to use this formalism, the space charge force is considered as linear. To calculate the space-charge effect, the real beam is replaced by an equivalent uniform beam having identical rms properties (sizes and emittances). The total emittance of the equivalent uniform bunched beam in each phase plane is then 5 times the rms emittance, and its envelope size is $\sqrt{5}$ times its rms size. For a continuous beam these factors are 4 for the emittance and 2($=\sqrt{4}$) for envelope.

Twiss parameters

Lets define $\langle w \rangle$ as the mean value of the w particle property over the beam at location *s*.

The beam phase-space position is defined by: $\begin{pmatrix} \bar{x} = \langle x \rangle \\ \overline{x'} = \langle x' \rangle \\ \bar{y} = \langle y \rangle \\ \overline{y'} = \langle y' \rangle \\ \bar{z} = \langle z \rangle \\ 0 \end{pmatrix}$

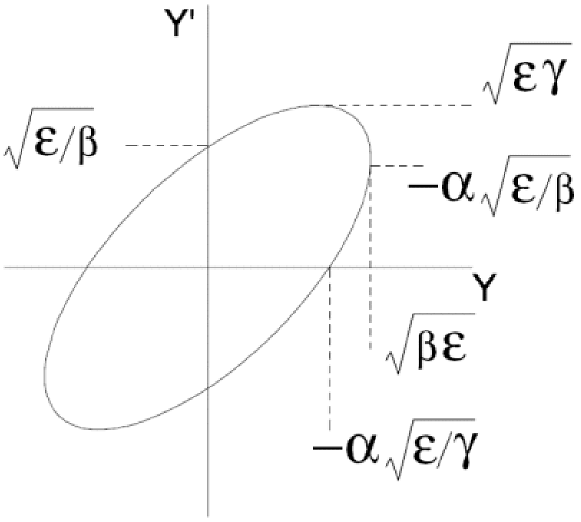
The beam rms sizes are defined by: $\tilde{w} = \sqrt{\left\langle (w-\overline{w})^2 \right\rangle}$, with w used for x, x', y, y', z or δ .

The beam rms correlation: $\overline{wv} = \left\langle (w-\overline{w}) \cdot (v-\bar{v}) \right\rangle$, with w and v used for x, x', y, y', z or δ.

The beam rms unnormalized emittances: $\tilde{\varepsilon}_w = \sqrt{\tilde{w}^2\, \tilde{w}'^2 - \langle w\, w' \rangle^2}$ with w used for x, y and z.

Note: In the preceding definition, *z'* is defined as $z' = (v_z - v_{zs})/v_{zs}$, *vz* and *vzδ* being the longitudinal velocities of respectively the particle and the synchronous particle. Most of the time, *δ* is used rather than *z'*. In that last case, the emittance is defined by the conversion from *εz* to *εzδ*.
In case of linear forces, the beam can be represented in sub phase planes by ellipses whose equation can be written: $\gamma_w w^2 + 2\alpha_w w w' + \beta_w w'^2 = \varepsilon_w$, where :

εw is the unnormalized beam effective emittance (which is the full emittance of a homogenous beam) define as 5 times the rms-emittance for bunched beam and 4 times that of a continuous beam.
$\alpha_w = -\overline{ww'}/\tilde{\varepsilon}_w$, $\beta_w = \tilde{w}^2/\tilde{\varepsilon}_w$, and $\gamma_w = \tilde{w}'^2/\tilde{\varepsilon}_w$ are the beam Twiss parameters satisfying the relationship: $\beta_w \gamma_w - \alpha_w^2 = 1$.


Y'
$\sqrt{\varepsilon\gamma}$
$\sqrt{\varepsilon/\beta}$
$-\alpha\sqrt{\varepsilon/\beta}$
Y
$\sqrt{\beta\varepsilon}$
$-\alpha\sqrt{\varepsilon/\gamma}$

The beam can be represented by a matrix, called the σ-matrix defined as:

$$[\sigma]=5\cdot\begin{bmatrix}\tilde{x}^2 & \overline{xx'} & \overline{xy} & \overline{xy'} & \overline{xz} & \overline{x\delta}\\ \overline{xx'} & \tilde{x}'^2 & \overline{x'y} & \overline{x'y'} & \overline{x'z} & \overline{x'\delta}\\ \overline{xy} & \overline{x'y} & \tilde{y}^2 & \overline{yy'} & \overline{yz} & \overline{y\delta}\\ \overline{xy'} & \overline{x'y'} & \overline{yy'} & \tilde{y}'^2 & \overline{y'z} & \overline{y'\delta}\\ \overline{xz} & \overline{x'z} & \overline{yz} & \overline{y'z} & \tilde{z}^2 & \overline{z\delta}\\ \overline{x\delta} & \overline{x'\delta} & \overline{y\delta} & \overline{y'\delta} & \overline{z\delta} & \tilde{\delta}^2\end{bmatrix}.$$

The evolution of the *σ -matrix* along the line from s_1 to s_2 can be calculated with the transfer matrix *R*:

$$[\sigma](s_2)=R\cdot[\sigma](s_2)\cdot R^T\ ,$$

Where R^T is the transpose of *R* and *[σ]* is the beam σ-matrix. Like with the transfer matrixes the *σ-matrixes* can be partitioned into 2x2 matrices:

$$[\sigma]=\begin{bmatrix}\sigma_{xx} & \sigma_{xy} & \sigma_{xz}\\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz}\\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz}\end{bmatrix}.$$

The elements are divided into small steps, whose transfer matrixes are used to transport the beam *σ-matrix*. The space-charge effect is applied at each step.

Definition of the matched beam

The 2x2 extracted *σ-matrix* can be written in terms of Twiss parameters.

$$\sigma_{ww}=\begin{bmatrix}\beta_w\varepsilon_w & -\alpha_w\varepsilon_w\\ -\alpha_w\varepsilon_w & \gamma_w\varepsilon_w\end{bmatrix}.$$

We observe that $\det(\sigma_{ww})=(\beta_w\gamma_w-\alpha_w^2)\cdot\varepsilon_w^2=\varepsilon_w^2$

Let's *R* be the transfer matrix of a lattice of a periodic structure: $R=\begin{bmatrix}R_{xx} & R_{xy} & R_{xz}\\ R_{yx} & R_{yy} & R_{yz}\\ R_{zx} & R_{zy} & R_{zz}\end{bmatrix}.$

The 2x2 extracted diagonal matrix can be written:

$$R_{ww}=\begin{bmatrix}\cos(\sigma_{w0})+\alpha_{w0}\sin(\sigma_{w0}) & \beta_{w0}\sin(\sigma_{w0})\\ -\gamma_{w0}\sin(\sigma_{w0}) & \cos(\sigma_{w0})-\alpha_{w0}\sin(\sigma_{w0})\end{bmatrix},$$

Where α_{wo}, β_{wo}, and γ_{wo} are the Twiss parameters of the beam matched to the lattice, and σ_{wo} is the zero-current phase advance per lattice in the *[w-w']* phase plane.

Mismatch factor

The mismatch factor (as defined above) between the two ellipses

$$\varepsilon = \gamma x^2 + 2\alpha x x' + \beta (x')^2$$

and

$$\varepsilon = \mathrm{G} x^2 + 2\mathrm{A} x x' + B (x')^2$$

is given by

$$\mathrm{M} = \sqrt{\left[\frac{1}{2}\left(R + \sqrt{R^2 - 4}\right)\right]} - 1$$

Where

$$\mathrm{R} = \beta G + B\gamma - 2\alpha A$$

For periodic structure, the chart “*mismatch factor*” form “*Charts*” tab-sheet compare for each period the input and ouput Twiss parameters.

Twiss parameters and acceleration

In case of acceleration, the determinant of the transfer matrix is not equal to 1, and the matrix cannot be written with the Twiss parameters, as defined before. To extract the Twiss parameters of the matched beam from the transfer matrix when there is acceleration, we use the matrix defined as below:

$$R^{'} = \left[\begin{array}{cc|cc|cc} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & \beta_o\gamma_o & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \beta_o\gamma_o & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & \beta_o\gamma_o \end{array}\right] \cdot R \cdot \left[\begin{array}{cc|cc|cc} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/\beta_i\gamma_i & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/\beta_i\gamma_i & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/\beta_i\gamma_i \end{array}\right],$$

Where β_i and γ_i are the relativistic parameters at the input and β_o and γ_o are the relativistic parameters at the output of the lattice. Now, the R' matrix determinant equals 1.

The Twiss parameters of the matched beam under acceleration conditions can then be deduced from the R' matrix:

$$\sigma_{x0} = \cos^{-1}\left(\frac{r'_{00} + r'_{11}}{2}\right), \quad \sigma_{y0} = \cos^{-1}\left(\frac{r'_{22} + r'_{33}}{2}\right), \quad \sigma_{z0} = \cos^{-1}\left(\frac{r'_{44} + r'_{55}}{2}\right),$$

$$\alpha_{x0} = \frac{(r'_{00} - r'_{11})}{2\sin(\sigma_{x0})}, \quad \alpha_{y0} = \frac{(r'_{22} - r'_{33})}{2\sin(\sigma_{y0})}, \quad \alpha_{z0} = \frac{(r'_{44} - r'_{55})}{2\sin(\sigma_{z0})},$$

$$\beta_{x0} = \beta_s\gamma_s \frac{r'_{01}}{\sin(\sigma_{x0})}, \quad \beta_{y0} = \beta_s\gamma_s \frac{r'_{23}}{\sin(\sigma_{y0})}, \quad \beta_{z0} = \beta_s\gamma_s^3 \frac{r'_{45}}{\sin(\sigma_{z0})},$$

$$\gamma_{x0} = \frac{-r'_{10}}{\beta_s \gamma_s \sin(\sigma_{x0})}, \qquad \gamma_{y0} = \frac{-r'_{32}}{\beta_s \gamma_s \sin(\sigma_{y0})}, \qquad \gamma_{z0} = \frac{-r'_{54}}{\beta_s \gamma_s^3 \sin(\sigma_{z0})},$$

Where, *r'ij* is a *R'* matrix coefficient (i^th^ row, j^th^ column) and α_{wo}, β_{wo}, and γ_{wo} are output Twiss parameters of the matched beam in the *[w-w']* phase plane.

Conversions between *[z-z'], [z-δ]* and *[Δφ-ΔW]* phase planes

β and *γ* being the beam reduced velocity and energy, *λ* the RF wavelength in vacuum, mc^2 the particle rest energy, we have in the paraxial approximation conditions the following relationship between parameters:

$$\Delta\varphi = -\frac{360°}{\beta\lambda} \cdot z \,,$$

But if the beam shows a divergence this relation becomes:

$$\Delta\varphi = -\frac{360°}{\beta\lambda} \cdot \sqrt{1 + \frac{(\Delta x')^2}{4} + \frac{(\Delta y')^2}{4}}\, z$$

Δφ and *z* being the RF phase and the position of a beam particle relative to the synchronous one.

$$\Delta W = \beta^2 \gamma^3 mc^2 \cdot z' = \beta^2 \gamma\, mc^2 \cdot \delta \,,$$

In the general case with a synchronous particle and a generator particle these relation become:

$$\delta = \frac{\Delta W \gamma_G + (\gamma_G - 1)(m_S c^2 - m_G c^2)}{m_G c^2 (\gamma_G^2 - 1)} \qquad \delta = z' \gamma_G^2 + \frac{(m_S c^2 - m_G c^2)}{m_G c^2}$$

ΔW, z' and *δ* being the energy, velocity and momentum of a beam particle relative to the synchronous one.

$$\varepsilon_w = \frac{360° \cdot mc^2}{\lambda} \cdot \varepsilon_{zn} = \frac{360° \cdot mc^2}{\lambda} \cdot \beta\gamma^3 \cdot \varepsilon_z = \frac{360° \cdot mc^2}{\lambda} \cdot \beta\gamma \cdot \varepsilon_{z\delta} \,,$$

εw and εzn being the normalized longitudinal emittances, εz and $\varepsilon z\delta$ being the unnormalized longitudinal emittances of the beam in respectively the *[z-z']* and the *[z- δ]* phase planes.

$$\beta_w = \frac{360°}{mc^2 \gamma^3 \beta^3 \lambda} \cdot \beta_z = \frac{360°}{mc^2 \gamma\, \beta^3 \lambda} \cdot \beta_{z\delta} \,,$$

βw, βz and $\beta z\delta$ are the *β*-Twiss parameters of the beam in respectively the *[Δφ-ΔW], [z-z']* and *[z-δ]* phase planes.

$$\alpha_w = -\alpha_z = -\alpha_{z\delta},$$

α_w, α_z and $\alpha_{z\delta}$ are the α-Twiss parameters of the beam in respectively *the [Δφ-ΔW], [z-z']* and *[z-δ]* phase planes.

$$\gamma_w = \frac{mc^2\gamma^3\beta^3\lambda}{360°}\cdot\gamma_z = \frac{mc^2\gamma\beta^3\lambda}{360°}\cdot\gamma_{z\delta},$$

γ_w, γ_z and $\gamma_{z\delta}$ are the γ-Twiss parameters of the beam in respectively the *[Δφ-ΔW], [z-z']* and *[z-δ]* phase planes.

Normalization of the emittance

$$\varepsilon_{tn} = \beta\gamma\cdot\varepsilon_t,$$

$$\varepsilon_{zn} = \beta\gamma^3\cdot\varepsilon_z = \beta\gamma\cdot\varepsilon_{z\delta}.$$

4D and 6D emittance definition

$$\varepsilon_{4D} = (\beta\gamma)^2\cdot\sqrt{Det\left[beam\_matrix(x,x',y,y')\right]}$$

$$\varepsilon_{6D} = (\beta\gamma)^3\cdot\sqrt{Det\left[beam\_matrix(x,x',y,y',z,\delta)\right]}$$

$$\varepsilon_t = \sqrt{\varepsilon_{4D}}$$

Beta X&Y function

The $\beta_{xx'}$ and $\beta_{yy'}$ function plotted in envelope charts are defined as following:

$$\beta_{xx'} = \frac{\sigma_{11} - \sigma_{66}\cdot T_{16}^2}{\varepsilon_{xx'}} \text{ and } \beta_{yy'} = \frac{\sigma_{33} - \sigma_{66}\cdot T_{46}^2}{\varepsilon_{yy'}}$$

σ, the beam matrix and *T*, the transfer matrix and ε is the non normalized emittance at the beginning of the structure.

Horizontal and vertical dispersion

The *H disp* and *V disp* function plotted in envelope charts are defined as following according to selected option:

The defaut dispersions are defined like following:

Hor_Disp = T_{16} / T_{66}

Ver_Disp = T_{36} / T_{66}

If option « *Use default dispersion def.* » from « *Main* » tab-sheet is not choosen then:

Hor_Disp = T_{16}
Ver_Disp = T_{36}

T is the transfer matrix.

In tracking way, dispersions are defined like following:

$$Hor_{disp} = \frac{< \frac{dp}{p} x >}{< (\frac{dp}{p})^2 >}$$

$$Ver_{disp} = \frac{< \frac{dp}{p} y >}{< (\frac{dp}{p})^2 >}$$

Synchronous phase definitions

The energy gain of a particle in a cavity is given by:

$$\Delta W = \int_{-\infty}^{+\infty} q\ E_z(s)\ cos\big(\Phi(s)\big)\ ds$$

Its depends on the cavity field amplitude $E_z(s)$ and RF phase seen by the particle $\Phi(s)$ when it is at position s. This RF phase is given by equation:

$$\Phi(s) = \Phi_0 + 2\pi \int_{s_0}^{s} \frac{ds}{\beta(s) \cdot \lambda_{RF}}$$

Where:

- Φ_0 is the RF phase when particle enters the cavity at position s_0. For a given particle, it varies linearly (of slope 1) with respect to the phase of the RF field in the cavity.
- $\beta(s)$ is the particle reduced velocity at position s,
- λ_{RF} is the wavelenght of the RF field in the cavity.

In TraceWIN, the beam dynamics in a cavity is numerically integrated in the field map starting at s_0.

Nevertheless, most of the accelerator physicists are used to designing linacs with cavities operating at a given:

- "***effective voltage***", V_0T and
- "***synchronous phase***", Φ_s.

as parameters of a simplified model of the energy gain in the cavity []:

$$\Delta W = q\ V_0T\ cos(\Phi_s)$$

This means that a "common language" has to be established in order to translate these parameters in real amplitude and phase of a field map.

TraceWIN proposed 2 models to define the *effective voltage* and the *synchronous phase* explained below.

NB1: The user can chose one or the other model, remembering that the beam dynamics is finally calculated in the field map whatever his choice. Nevertheless, the chosen model has an impact on the way the cavity is tuned for given *effective voltage* and *synchronous phase*.
NB2: The two models gives the same *effective voltage* and *synchronous phase* when the field is low (the velocity change is negligible in the cavity)

NB3: Both models give 0 energy gain for $\Phi_s = -90°$ but only model 2 gives maximum energy gain for $\Phi_s = 0°$.

NB4: The phase acceptance associated to a given *synchronous phase* depends on the chosen model and the acceleration conditions. It is recommended to check it with multiparticle simulations.

Historic model:

Starting from the exact calculation of energy gain:

$$\Delta W_{real} = \int_{-\infty}^{+\infty} q\, E_z(s)\, cos\big(\Phi(s)\big)\, ds$$

Adding and subtracting an arbitrary phase Φ_s in the cosine:

$$\Delta W_{real} = \int_{-\infty}^{+\infty} q\, E_z(s)\, cos(\Phi(s) - \Phi_s + \Phi_s)\, ds$$

Using trigonometric laws: $cos(a+b) = cos(a)\, cos(b) - sin(a)\, sin(b)$, one gets:

$$\Delta W_{real} = \int_{-\infty}^{+\infty} q\, E_z(s)[\, cos(\Phi(s) - \Phi_s)\, cos(\Phi_s) - sin(\Phi(s) - \Phi_s)\, sin(\Phi_s)]\, ds$$

$$\Delta W_{real} = \int_{-\infty}^{+\infty} q\, E_z(s)\, cos(\Phi(s) - \Phi_s)\, ds \cdot cos(\Phi_s) - \int_{-\infty}^{+\infty} q\, E_z(s)\, sin(\Phi(s) - \Phi_s)\, ds \cdot sin(\Phi_s)$$

This energy gain can then be “naturally” modelled by:

$$\Delta W_{mod} = q\, V_0 T\, cos(\Phi_s),$$

by choosing Φ_s cancelling the second term, i.e.:

$$\int_{-\infty}^{+\infty} q\, E_z(s)\, sin(\Phi(s) - \Phi_s)\, ds = 0$$

Using trigonometric laws: $sin(a+b) = sin(a)\, cos(b) + cos(a)\, sin(b)$, one gets:

$$\int_{-\infty}^{+\infty} q\, E_z(s)\, sin\big(\Phi(s)\big)\, ds \cdot cos(\Phi_s) - \int_{-\infty}^{+\infty} q\, E_z(s)\, cos\big(\Phi(s)\big)\, ds \cdot sin(\Phi_s) = 0$$

Leading to a definition of the synchronous phase Φ_s:

$$\boxed{tan[\Phi_s] = \frac{\int_{-\infty}^{+\infty} q\, E_z(s)\, sin\big(\Phi(s)\big)\, ds}{\int_{-\infty}^{+\infty} q\, E_z(s)\, cos\big(\Phi(s)\big)\, ds}}$$

and of the effective voltage:

$$\boxed{V_0 T = \frac{\Delta W_{real}}{q\, cos[\Phi_s]} = \int_{-\infty}^{+\infty} E_z(s)\, cos(\Phi(s) - \Phi_s)\, ds = \left|\int_{-\infty}^{+\infty} E_z(s)\, e^{j\cdot(\Phi(s))}\, ds\right|}$$

PS: one notes that, in these conditions:

$$\frac{\partial \mathit{\Delta} W_{mod}}{\partial \Phi_0} = \frac{\partial \mathit{\Delta} W_{real}}{\partial \Phi_0}$$

New model:

New model proposed by Jean-Michel LAGNIEL (GANIL).
See « *Longitudinal beam dynamics at high accelerating fields, what changes ?* », ROSCOFF 2021.

Equalizing the real energy gain of the synchronous particle with the one given by the model:

$$\mathit{\Delta} W_{real} = \int_{-\infty}^{+\infty} q\, E_z(s) \cos\big(\Phi(s)\big)\, ds$$

$$\mathit{\Delta} W_{mod} = q\, V_0 T \cos(\Phi_s)$$

$$\mathit{\Delta} W_{real} = \mathit{\Delta} W_{mod} \quad => \quad qV_0 T \cos(\Phi_s) = \mathit{\Delta} W_{real}$$

$\mathit{\Delta} W_{real}$ is calculated numerically from the field map.

Equalizing the variation with RF phase of the real energy gain with the variation with synchronous phase of the modelled energy gain around the synchronous particle (linearization):

$$\left.\frac{\partial \mathit{\Delta} W_{real}}{\partial \delta \Phi_0}\right]_{\Phi_s} = fm_{21real}$$

$$\left.\frac{\partial \mathit{\Delta} W_{mod}}{\partial \delta \Phi_0}\right]_{\Phi_s} = -q\, V_0 T\ \sin(\Phi_s)$$

$$\frac{\partial \mathit{\Delta} W_{real}}{\partial \Phi_0} = \frac{\partial \mathit{\Delta} W_{mod}}{\partial \Phi_s} \quad => \quad -q\, V_0 T \sin(\Phi_s) = fm_{21real}$$

fm_{21real} is calculated numerically from the field map.

Leading to a definition of the synchronous phase Φ_s:

$$\boxed{\tan(\Phi_s) = \frac{fm_{21real}}{\mathit{\Delta} W_{real}}}$$

and of the effective voltage:

$$\boxed{qV_0 T = \frac{\mathit{\Delta} W_{real}}{\cos[\Phi_s]} = \sqrt{(\mathit{\Delta} W_{real})^2 + {fm_{21real}}^2}}$$

Transit time factor definition

These following definitions are used in all accelerating element.
T: is the usual time factor transit given by SUPERFISH.
T': is the *T'* from SUPERFISH time -2π
T'': is the *T''* from SUPERFISH time -$4\ \pi^2$

$$T(\beta)=T_s+kT_s'\cdot(\kappa-1)+k^2T_s''\cdot\frac{(\kappa-1)^2}{2},$$

$$kT'(\beta)=kT_s'+k^2T_s''\cdot(\kappa-1)\ \ldots \qquad \text{With } \kappa=\frac{k_s}{k}=\frac{\beta_s}{\beta}.$$

The electric field is corrected according to *T(β)*.
Coordinate transformations are given in Wangler's book page 202:

Phase advance definition

TraceWin calculates the particle phase advance in two ways. The first one is the extraction of the phase advance from the transfer matrix of the lattice ($k_{0x,y}$, μ or σ_0). The second one is done by the beta function integration along a lattice giving the phase advance with (σ) or without σ_0 space charge:

$\sigma_x = \int_0^L \frac{dl}{\beta(l)}$ with L is the lattice length and $\beta = \frac{<x^2>}{\varepsilon_x}$.

x is the beam RMS or effective size and ε_x the unnormalized RMS or effective emittance.

with $M=\begin{bmatrix} M_{xx} & M_{xy} & M_{xz} \\ M_{yx} & M_{yy} & M_{yz} \\ M_{zx} & M_{zy} & M_{zz} \end{bmatrix}$ the transfer matrix of the lattice

$\sigma_{0x} = arcos\left(\frac{Tr[M_{xx}]}{2\sqrt{1-Det[M_{xy}]}}\right)$, $\sigma_{0y} = arcos\left(\frac{Tr[M_{yy}]}{2\sqrt{1-Det[M_{yx}]}}\right)$ and $\sigma_{0z} = arcos\left(\frac{Tr[M_{zz}]}{2}\right)$ with $k_0 = \frac{\sigma_0}{L}$

The first phase advance type can be plotted from the "***Phase advance”->”Structure”***.of "***Chart***" page
And the second one can be plotted from the "***Phase advance”->”Beam”*** of "***Chart***" page

Residual orbit

Use in the error studies in order to know the beam gravity evolution, It’s defined like below:

$$\bar{x}=\frac{1}{N}\sum_1^N x \qquad x_{rms}=\sqrt{\frac{1}{N}\sum_1^N x^2-\bar{x}^2}$$

$$\bar{y}=\frac{1}{N}\sum_1^N y \qquad y_{rms}=\sqrt{\frac{1}{N}\sum_1^N y^2-\bar{y}^2}$$

$$\bar{r} = \sqrt{\bar{x}^2 + \bar{y}^2} \qquad\qquad r_{rms} = \sqrt{\frac{1}{N}\sum_{1}^{N} r^2 - \bar{r}^2}$$

x & *y* are the beam gravity position and *N* is the number of run or linac.

Halo definition

In a one dimension linear motion (described in (p_i, q_i) phase-space), the following quantities are kinematic invariants:

$$I_2^i \equiv \langle q_i^2 \rangle \langle p_i^2 \rangle - \langle q_i\ p_i \rangle^2$$
$$I_4^i \equiv \langle q_i^4 \rangle \langle p_i^4 \rangle + 3\langle q_i^2 p_i^2 \rangle^2 - 4\langle q_i p_i^3 \rangle \langle q_i^3 p_i \rangle$$

< > is the average value over the beam distribution.

One defines the halo parameter, H_i (*i* for *x*, *y*, or *z* direction), as a ratio between a function of the fourth order momentum and the second order momentum:

$$H_i \equiv \frac{\sqrt{3I_4^i}}{2I_2^i} - 2 = \frac{\sqrt{3\langle q_i^4 \rangle \langle p_i^4 \rangle \ + 9\langle q_i^2 p_i^2 \rangle^2 - 12\langle q_i\ p_i^3 \rangle \langle q_i^3 p_i \rangle}}{2\langle q_i^2 \rangle \langle p_i^2 \rangle - 2\langle q_i\ p_i \rangle^2}$$

This quantity is **conserved in linear forces** and **by homothetic transformation** of the beam distribution. It is normalized and centered in order to have:

- $H_i = 0$, for uniform elliptical distribution,
- $H_i = 1$, for Gaussian elliptical distribution.

From ref: "PHYSICAL REVIEW SPECIAL TOPICS - ACCELERATORS AND BEAMS,VOLUME 5, 124202 (2002)", "Beam halo definitions based upon moments of the particle distribution (C. K. Allen and T. P.Wangler) "

Gradient definition

In TraceWin the gradient definition is defined as following:

$$G_n = \frac{B_0}{a^{(n-1)}} = \frac{1}{(n-1)!}\frac{\partial^{(n-1)} B_y}{\partial x^{(n-1)}}$$

Where:
n = 2 Quadrupole
n = 3 Sextupole
n = 4 Octupole

With: B_0, the magnetic field (T) on the pole and *a*, the half aperture.

This definition depends of computer programs:

$$g(TraceWin) = g(TRANSPORT) = \frac{1}{(n-1)!} g(MAD)$$

Space charge

Space Charge in envelope simulations

In the *envelop* simulation, the space-charge (SC) force is linearized assuming an equivalent uniform beam if the beam is respectively continuous or bunched. The number of SC kicks is given by the parameter "***Step of calculation per βλ***" in the "***main***" *sheet*.

In a free space, the motion equation of a particle, feeling only space-charge force, can be written along one direction in the **beam frame *R****:

$$\frac{d^2w^*}{dt^{*2}} = F_w^*,$$

F_w^* being the *w* component of the space-charge force (divided by the particle rest mass).

All the quantities with a star (*) are expressed in the **beam frame *R****, all quantities without star are expressed in the **laboratory frame *R***.
We have, from the Lorentz transform:

$$dt^* = dt/\gamma\,,\ x^* = x\,,\ y^* = y\,,\ z^* = \gamma \cdot z\,,$$

and:

$$ds = \beta c \cdot dt\,.$$

The derivation with *s*, the beam longitudinal position used as independent variable, gives:

$$\frac{d^2w^*}{ds^2} = F_w = \frac{F_w^*}{(\gamma\beta c)^2}\,.$$

In the longitudinal direction, one has:

$$\frac{d\delta}{ds} = \gamma^2 \frac{d^2z}{ds^2} = \gamma \frac{d^2z^*}{ds^2} = \gamma \frac{F_w^*}{(\gamma\beta c)^2} = \gamma F_w\,.$$

Continuous beam

In continuous beam, no space-charge force acts along the longitudinal direction, and:

$$F_x = \frac{2K}{a_x(a_x + a_y)} \cdot x\,, \qquad F_y = \frac{2K}{a_y(a_x + a_y)} \cdot y\,,$$

with: $K = \frac{|q| \cdot I}{\pi\varepsilon_0 mc^3\beta^3\gamma^3}$, the beam generalized perveance.

a_x

Bunched beam

The space-charge force *w*-component (*w* for *x*, *y* or) acting on the particles:

$$F_w = \frac{K_w}{a_x^* a_y^* a_z^*} \cdot w^* ,$$

with: $$K_w = \frac{3 \cdot |q| \cdot Q}{4\pi\varepsilon_0 mc^2 \beta^2 \gamma^2} \cdot f_w(a_x^*, a_y^*, a_z^*) ,$$

and $$f_w(a_x^*, a_y^*, a_z^*) = \int_0^\infty \frac{ds}{(a_w^{*2} + s)\sqrt{(a_x^{*2} + s)(a_y^{*2} + s)(a_z^{*2} + s)}}$$ is the **form factor** such as:

$$f_x + f_y + f_z = 1.$$

a_x^*, a_y^* and a_z^* are the beam semi-axes of a uniform ellipsoid (√5 times the rms beam sizes) <u>in the beam frame</u>, *Q* is the bunch charge, *ε₀* is the vacuum permittivity. Note that: $a_z^* = \gamma \cdot a_z$.

The form factor integral calculation depends on the ratios a_z^* / a_x^* and a_z^* / a_y^* . If they are lower than 12 the integral is calculated by the Gauss method with a very good precision. If they are greater than 12 (which happens obviously when the beam is ultra-relativistic due to the Lorentz transformation) an expended development is used and slightly reducing the result precision.

Space-charge application

Frame change

The space charge impulse should be applied in the beam frame. Before any application, the beam *σ-matrix* should be written from the laboratory frame to the beam frame by making the transformation:

$$[\sigma] = R_\gamma \cdot [\sigma] \cdot R_\gamma^{\ T} ,$$

with: $$R_\gamma = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \gamma & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} .$$

After the application of space-charge impulse (taking into account the beam coupling), the *σ-matrix* should be written back to the laboratory frame:

$$[\sigma] = R_{\gamma^{-1}} \cdot [\sigma] \cdot R_{\gamma^{-1}}^{\ T} ,$$

with: $R_{\gamma^{-1}} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \gamma^{-1} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$.

Beam coupling or tilted.

Due to the elements (magnetic coil or dipole) or initial conditions, the beam ellipsoid in *[x-y-z]* space can be tilted. In this case, the beam ellipsoid (in the beam frame) must first be transformed to a coordinate system in which it is upright before applying the space-charge impulses.
If the ellipsoid is tilted in the *[x-y]* plane, the angle between the x-axis and the axis of the elliptical projection on the *[x-y]* plane is:

$$\theta_{xy} = \frac{1}{2}\tan^{-1}\left(\frac{2\sigma_{02}}{\sigma_{22} - \sigma_{00}}\right),$$

with σ_{ij} , σ -matrix elements.

If the ellipsoid is tilted in the *[x-z]* plane:

$$\theta_{xz} = \frac{1}{2}\tan^{-1}\left(\frac{2\sigma_{04}}{\sigma_{44} - \sigma_{00}}\right).$$

If the ellipsoid is tilted in the *[y-z]* plane:

$$\theta_{yz} = \frac{1}{2}\tan^{-1}\left(\frac{2\sigma_{24}}{\sigma_{44} - \sigma_{22}}\right).$$

The ellipsoid can be brought upright by rotations of angles $-\theta_{xy}$ (or $-\theta_{xz}$ and $-\theta_{yz}$) accomplished by applying the transfer matrixes:

$$R_{xy} = \left[\begin{array}{cc:cc:cc} \cos(\theta_{xy}) & 0 & \sin(\theta_{xy}) & 0 & 0 & 0 \\ 0 & \cos(\theta_{xy}) & 0 & \sin(\theta_{xy}) & 0 & 0 \\ \hdashline -\sin(\theta_{xy}) & 0 & \cos(\theta_{xy}) & 0 & 0 & 0 \\ 0 & -\sin(\theta_{xy}) & 0 & \cos(\theta_{xy}) & 0 & 0 \\ \hdashline 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right],$$

R_{xz} and R_{yz} can be obtained the same way.

The rotation is then applied:

$$[\sigma] = R_{yz} \cdot R_{xz} \cdot R_{xy} \cdot [\sigma] \cdot R_{xy}{}^T \cdot R_{xz}{}^T \cdot R_{yz}{}^T ,$$
$$[\sigma] = Rot \cdot [\sigma] \cdot Rot^T .$$

When the ellipse is upright, the space-charge impulses can be applied. The three reverse rotations can then be applied:

$$[\sigma] = R_{xy}{}^T \cdot R_{xz}{}^T \cdot R_{yz}{}^T \cdot [\sigma] \cdot R_{yz} \cdot R_{xz} \cdot R_{xy} ,$$
$$[\sigma] = Rot^T \cdot [\sigma] \cdot Rot .$$

Space-charge impulse

The space-charge kick applied on distance Δs (the calculation step) uses the transfer matrix R_{ce}:

$$R_{ce} = \left[\begin{array}{cc:cc:cc} 1 & 0 & 0 & 0 & 0 & 0 \\ F_x\Delta s & 1 & 0 & 0 & 0 & 0 \\ \hdashline 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & F_y\Delta s & 1 & 0 & 0 \\ \hdashline 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & \gamma F_z\Delta s & 1 \end{array}\right].$$

The space-charge impulse is applied in the bunch frame, where the beam ellipse is upright:

$$[\sigma] = R_{ce} \cdot [\sigma] \cdot R_{ce}{}^T .$$

Finally, the total space charge effect is given by:

$$\boxed{[\sigma] = R_{\gamma^{-1}} \cdot Rot^T \cdot R_{ce} \cdot Rot \cdot R_\gamma \cdot [\sigma] \cdot R_\gamma{}^T \cdot Rot^T \cdot R_{ce}{}^T \cdot Rot \cdot R_{\gamma^{-1}}{}^T}.$$

Space Charge in Partran simulation

In the *Partran* simulation, the user can select its space-charge routine in the “***Multiparticle / Partran space charge options***” sheet.

- **PICNIR (2D) – (r, z)**, Particles In Cells Numerical Integration between Rings, is based on the SCHEFF model. It considers a beam with a transverse circular symmetry. If it is not the case, a correction is applied which is less and less adequate as much as the beam is not circular. The first parameter gives the total number of lattices in radial (r) direction, the second gives the total number of lattices in longitudinal (z) direction. The mesh size is adjusted from 0 to 3.5 times r-

rms in transverse, and to +/- 3.5 times z-rms in the longitudinal. The space-charge force outside the mesh is this of an equivalent gaussian beam.
Set number of lattice, Nz, lower than 0 allows to cancel to longitudinal forces.

- **PICNIC (3D) – (xy, z)**, Particles In Cells Numerical Integration between Cubes, is a fully 3D space-charge routine. The first parameter gives the half-number of lattices in horizontal (x) and vertical (y) directions, the second gives the half-number of lattices in longitudinal (z) direction. Its options are :
  - **Weight**: if 0, the particle charge is deposited in the lattice where it stands. If 1, the particle charge is distributed in the closest lattices (small smoothing). The nominal value is 0.
  - **Mesh/rms**: the mesh full size is adjusted around the beam to +/- *Mesh/rms* time rms sizes (x, y and z). The nominal value (if one sets 0) is 3.5. The space-charge force outside the mesh is this of an equivalent gaussian beam.
  - **Skip**: in order to slightly speed-up the space charge calculation, one can depose every Nth particle in the mesh. This parameter is *N*. The nominal value is 1.

- **Special**: written for dedicated user.
- **CE_CYL**: written for dedicated user.
- **My space-charge**: space charge routine defined from external library. See example shown in following chapter.

The number of SC kicks in field maps or in DRIFT is set in the "***Multiparticle / Partran step of calcul***" sheet. For all hard-edge elements (SOLENOID, BEND…), the SC kick is applied at its middle. If you want increase the SC kick rate, please cut the element in many pieces.

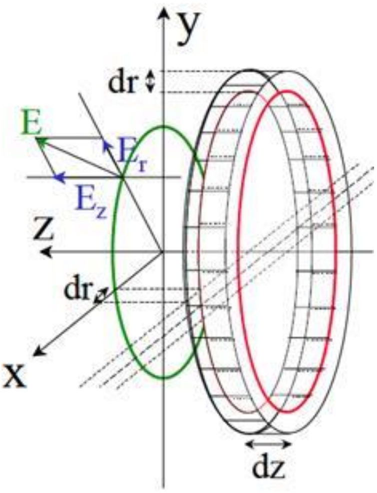
For **DTL cell, QUAD & QUAD_ELE** elements, multi SC kick can also be applied according to proposed options selected.

In all cases, the procedure is the following:

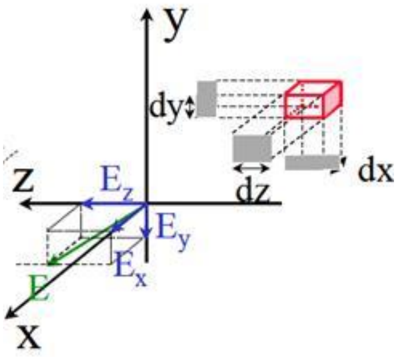
- the bunch, whose distribution is given at a given abscissa, is extended to the **time of the CoG** in bunch frame (where the average velocity is zero)e,
- the **mesh size** is calculated from the bunch RMS sizes,
- the only **electric field** is calculated in the bunch frame.
- the **SC kicks** are applied to the particles **momentum** in the bunch frame,
- the particles new angles and energy in **laboratory frame** are deduced.

The change of frames used Lorentz transformation. In case of ultra-relativist bunches in which particles can be relativistic even in bunch frame, the SC routine reaches one of its limits.

When the bunch is not short compared to the intra-bunches distance, effect of neighbor bunches are taken into account and particles out of +/- 180° are temporary "moved" into the bunch (with 360° jumps) by the SC routine to estimate their contribution to the SC and calculate the SC they fill.



Interaction between rings
Cylindrical symmetry
assumed with PICNIR



Interaction between cubes
NO symmetry assumed
with PICNIC

Recommendations for user:

- Have a look on http://accelconf.web.cern.ch/AccelConf/l98/PAPERS/MO4042.PDF to understand some subtleties of SC routines.
- PICNIR should be used with almost circular beams or to accelerate the calculus but to the expense or a lower accuracy.
- Use lattices with aspect ratio (in bunch frame) as close as 1 as possible.

Adjust the number of SC lattices with the number of particles. For example, in PICNIC, we recommend to use 7×7 cells on ±3.5σ mesh with 10k particles to tune a linac, 10×10 cells on ±4σ mesh with 100k particles to finalize the linac design and 15×15 cells on ±5σ with 1M particles to explore the tails and >10M particles for outside communication (in that case, choose the mesh size you want) !
Have a look on noise anaylisis: Phys. Rev. ST Accel. Beams 17, 124201 (2014)

Develop its own space-charge routine

This feature allows to each user to develop its own space-charge routine. A detailed example following explains how to perform it. Use the following '*main.cpp*' file and compile it as a dynamic library. This library has be located either in the structure (*.dat) directory or in the executable directory.

```
/*************************************************************************
main.cpp
Windows -> Dynamic library (dll)
Linux or MacOS -> Dynamic shared object (so, dylib)
-------------------
begin : Wed Dec 1 2010
copyright : (C) 2010 by URIOT Didier
email : duriot@cea.fr
*************************************************************************/

#ifdef _WIN32
  #include <windows.h>
  #define DLL_EXPORT __declspec(dllexport)
#else
  #define DLL_EXPORT
```

```
#endif
#include <cmath>
#include <cstdio>
#include <cstdlib>
#include <cstring>


#ifdef __cplusplus
extern "C" {
#endif

//----------------------------------------------------------------------
//-- MY_SC - MY_SC - MY_SC - MY_SC - MY_SC - MY_SC - MY_SC -----------
//----------------------------------------------------------------------
//
// You have to compile your space-charge routine as an dll
// - my_space_charge.dll          (for windows)
// - my_space_charge.so           (for linux)
// - my_space_charge.dylib        (for MacOS)
//
// Commands to compil and link GNU gcc compiler in Windows:
// g++.exe -m32 -Wall -c main.cpp -o main.o
// g++.exe -m32 -shared -Wl,--dll main.o -o my_space_charge.dll
// If you use 64bits TraceWin version replace both -m32 by -m64

// Commands to compil and link GNU gcc compiler in MAC OS:
// g++ -m32 -Wall -pedantic -c  main.c -o main.o
// g++ -m32 -Wall -shared -dynamiclib  main.o -o my_space_charge.dylib
// If you use 64bits TraceWin version replace both -m32 by -m64

// MY_SC syntax example
// repect the name of the routine

int DLL_EXPORT space_charge(double Zs, double ds,int mesh1,int mesh2,int Nele,int
npart,double *cord,double *loss,double freq,double mass0,int q,double *ws,double
*Ibeam,double *extra_param,char *error_mess)
{
  // Zs             : Current position in the element (from ds/2 to Length-ds/2) (m)
  // ds             : kick applied distance (m)
  // mesh1          : First parameter of the SC option in sheet “multiparticle” sheet
  // mesh2          : Second parameter of the SC option in sheet “multiparticle” sheet
  // cord           : See example (drift)
  // loss           : see example (drift)
  // freq           : beam frequency (Hz)
  // mass           : Particle mass (eV)
  // q              : particle charge state
  // *ws            : reference kinetic energy (eV) (can be modified)
  // *Ibeam         : Beam current (A) (can be modified)
 // *extra_param    : Not used
  // error_mess     : TraceWin stop and show this error message if this function return 0
  // if error_mess!="" and function return 1, this message is print to the standard console
without stop TraceWin
  double zzs,bgs,gams,betas;
  double x,y,xp,yp,z,dpsp,w,gamma;

  strcpy(error_mess,"");
  zzs=(*ws)/mass0;
  bgs=sqrt(zzs*(2.+zzs));
  gams=1.0+zzs;
  betas=bgs/gams;

  // drift treansport example, has to be replace by your SC routine
  for (int i=0;i<npart;i++) {
    if ((int)loss[i]==0) {
      x=cord[i*6];      // m
      xp=cord[i*6+1];   // rad
      y=cord[i*6+2];    // m
      yp=cord[i*6+3];   // rad
      z=cord[i*6+4];    // m
      dpsp=cord[i*6+5]; // dp/p
      x=x+ds*xp;
      y=y+ds*yp;
      w=dpsp*betas*betas*gams*mass0+(*ws);
      gamma = 1+w/mass0;
      z=z+dpsp*ds/(gamma*gamma);

      cord[i*6]=x;
      cord[i*6+2]=y;
      cord[i*6+4]=z;
    }
  }
  return(1);
}
#ifdef __cplusplus
}
#endif
```

Particle motion in electromagnetic field

The equations of particle motions in electromagnetic field (RF or static) defined below are mainly uses in FIED_MAP elements for each particle (Partran) and for beam centroid (Envelope) with linearization.

General equations of the dynamics

The variation of the amount of movement of a particle with charge *q* and mass *m* is within an electromagnetic field:

$$\frac{d\vec{p}}{dt} = q \cdot \left(\vec{E} + \frac{\vec{p}}{\gamma m} \times \vec{B} \right)$$

In Cartesian coordinates:

$$\begin{cases} \dot{p}_x = \dfrac{dp_x}{dt} = q \cdot \left(E_x + \dfrac{p_z}{\gamma m} \cdot \left(y' \cdot B_z - B_y \right) \right) \\ \dot{p}_y = \dfrac{dp_y}{dt} = q \cdot \left(E_y + \dfrac{p_z}{\gamma m} \cdot \left(B_x - x' \cdot B_z \right) \right) \\ \dot{p}_z = \dfrac{dp_z}{dt} = q \cdot \left(E_z + \dfrac{p_z}{\gamma m} \cdot \left(x' \cdot B_y - y' \cdot B_x \right) \right) \end{cases}$$

With:

$$\frac{dx}{ds} = x' = \frac{p_x}{p_z} \text{ et } \frac{dy}{ds} = y' = \frac{p_y}{p_z}$$

Then:

$$\frac{dx'}{ds} = \frac{d\left(p_x / p_z \right)}{dt} \cdot \frac{dt}{ds} = \frac{1}{\beta_z c} \cdot \frac{\dot{p}_x - \dot{p}_z \cdot x'}{p_z}$$

Finally, we get:

$$\frac{dx'}{ds} = \frac{1}{\gamma \beta_z^2 \left(mc^2 / q \right)} \cdot \left(E_x - x' \cdot E_z + \beta_z c \cdot \left(x'y' \cdot B_x - \left(1 + x'^2 \right) \cdot B_y + y' \cdot B_z \right) \right)$$

$$\frac{dy'}{ds} = \frac{1}{\gamma \beta_z^2 \left(mc^2 / q \right)} \cdot \left(E_y - y' \cdot E_z + \beta_z c \cdot \left(\left(1 + y'^2 \right) \cdot B_x - x'y' \cdot B_y - x' \cdot B_z \right) \right)$$

As regards the longitudinal dynamics, the variables used are generally either kinetic energy *T*, or the amount of movement of the particle *p*. The variation of these parameters is derived from the relationship:

$$p^2 = p_x^2 + p_y^2 + p_z^2 = p_z^2 \cdot \left(1 + x'^2 + y'^2 \right) = \frac{\left(E_0 + T \right)^2 - E_0^2}{c^2}.$$

Given:

$$\frac{dp}{ds}=\frac{\frac{dp_z}{ds}\cdot\left(1+x'^2+y'^2\right)+p_z\cdot\left(x'\cdot\frac{dx'}{ds}+y'\cdot\frac{dy'}{ds}\right)}{\sqrt{1+x'^2+y'^2}}$$

$$p^2=p_x^2+p_y^2+p_z^2=p_z^2\cdot\left(1+x'^2+y'^2\right)=\frac{\left(E_0+T\right)^2-E_0^2}{c^2}$$

Then:

$$\frac{dp}{ds}=\frac{q}{\beta\cdot c}\cdot\left(E_z+x'\cdot E_x+y'\cdot E_y\right)$$

And:

$$\frac{dT}{ds}=q\cdot\left(E_z+x'\cdot E_x+y'\cdot E_y\right)$$

In the PIC code, these equations must be integrated step by step

Linearization

Under the assumption of paraxiality, linearization of the equations gives us:

$$\frac{dx'}{ds}=\frac{1}{\gamma\beta^2\left(mc^2/q\right)}\cdot\left(E_x-x'\cdot E_z+\beta c\cdot\left(-B_y+y'\cdot B_z\right)\right)$$
$$\frac{dy'}{ds}=\frac{1}{\gamma\beta^2\left(mc^2/q\right)}\cdot\left(E_y-y'\cdot E_z+\beta c\cdot\left(B_x-x'\cdot B_z\right)\right)$$
$$\frac{dp}{ds}=\frac{1}{\beta\cdot c}\cdot\left(E_z+x'\cdot E_x+y'\cdot E_y\right)$$

In TraceWin, longitudinal variables are:

$$z=-\frac{\varphi}{2\pi}\cdot\beta\lambda \quad \text{and} \quad \delta=\frac{p-p_s}{p_s}$$

ps is the amount of movement of the synchronous particle.
The evolution of δ is then:

$$\frac{d\delta}{ds}=\frac{1}{p_s}\cdot\left(\frac{dp}{ds}-(1+\delta)\cdot\frac{dp_s}{ds}\right)\frac{dp}{ds}=\frac{1}{\beta\cdot c}\cdot\left(E_z+x'\cdot E_x+y'\cdot E_y\right)$$

The fields can be modeled by:

$$E_x=E_{x0}+\frac{\partial E_x}{\partial x}\cdot x+\frac{\partial E_x}{\partial y}\cdot y+\frac{\partial E_x}{\partial z}\cdot z\ \ldots$$

$$B_x = B_{x0} + \frac{\partial B_x}{\partial x}\cdot x + \frac{\partial B_x}{\partial y}\cdot y + \frac{\partial B_x}{\partial z}\cdot z \ldots$$

The linearized equations of motion become:

$$\frac{dx'}{ds} = \frac{q}{\gamma_s \beta_s^2 mc^2}\cdot\left(\begin{array}{l}\left(E_{x0} - \beta_s c\cdot B_{y0}\right) - E_{z0}\cdot x' + \beta_s c\cdot B_{z0}\cdot y' - \left[\left(2-\beta_s^2\right)\cdot E_{x0} + \beta_s c\cdot B_{y0}\right]\cdot\delta \\ \left(\frac{\partial E_x}{\partial x} - \beta_s c\cdot\frac{\partial B_y}{\partial x}\right)\cdot x + \left(\frac{\partial E_x}{\partial y} - \beta_s c\cdot\frac{\partial B_y}{\partial y}\right)\cdot y + \left(\frac{\partial E_x}{\partial z} - \beta_s c\cdot\frac{\partial B_y}{\partial z}\right)\cdot z\end{array}\right)$$

$$\frac{dy'}{ds} = \frac{q}{\gamma_s \beta_s^2 mc^2}\cdot\left(\begin{array}{l}\left(E_{y0} + \beta_s c\cdot B_{x0}\right) - \beta_s c\cdot B_{z0}\cdot x' - E_{z0}\cdot y' - \left[\left(2-\beta_s^2\right)\cdot E_{y0} - \beta_s c\cdot B_{x0}\right]\cdot\delta \\ \left(\frac{\partial E_y}{\partial x} + \beta_s c\cdot\frac{\partial B_x}{\partial x}\right)\cdot x + \left(\frac{\partial E_y}{\partial y} + \beta_s c\cdot\frac{\partial B_x}{\partial y}\right)\cdot y + \left(\frac{\partial E_y}{\partial z} + \beta_s c\cdot\frac{\partial B_x}{\partial z}\right)\cdot z\end{array}\right)$$

$$\frac{d\delta}{ds} = \frac{q}{\gamma_s \beta_s^2 mc^2}\cdot\left(\frac{\partial E_z}{\partial x}\cdot x + \frac{\partial E_z}{\partial y}\cdot y + \frac{\partial E_z}{\partial z}\cdot z + E_{x0}\cdot x' + E_{y0}\cdot y' - \left(2-\beta_s^2\right)\cdot E_{z0}\cdot\delta\right)$$

We used here:

$$d\left(\left(\gamma\beta^2\right)^{-1}\right) = -\frac{2-\beta_s^2}{\gamma_s\beta_s^2}\cdot\delta\,,$$

$$d\beta = \frac{\beta_s}{\gamma_s^2}\cdot\delta\,,$$

$$d\left(\beta^{-1}\right) = -\frac{\delta}{\beta_s\cdot\gamma_s^2}\,.$$

Some examples of element

Element with symmetry of revolution

Many elements (solenoids, Einzel lenses, RF cavities (DTL, CDC, supra-elliptical,)) have symmetry of revolution around the beam axis. In this case, the fields are represented in the cylindrical reference: (r, θ, z).

Then: $$\begin{pmatrix} V_x\left(r\cdot\cos\theta, r\cdot\sin\theta\right) \\ V_y\left(r\cdot\cos\theta, r\cdot\sin\theta\right)\end{pmatrix} = \begin{pmatrix}\cos\theta & -\sin\theta \\ \sin\theta & \cos\theta\end{pmatrix}\cdot\begin{pmatrix}V_r\left(r,\theta\right) \\ V_\theta\left(r,\theta\right)\end{pmatrix}$$

Magnetic solenoid

In solenoid we get at the first order:

$$\vec{E} = \vec{0},$$
$$B_\theta = 0,$$
$$B_r = \frac{\partial B_r}{\partial r} \cdot r,$$
$$B_z = B_{z0}.$$

The linearized equations of motion become:

$$\frac{dx'}{ds} = \frac{q}{\gamma_s \beta_s^2 mc^2} \cdot \left(\beta_s c \cdot B_{z0} \cdot y' - \beta_s c \cdot \frac{\partial B_r}{\partial r} \cdot y \right)$$

$$\frac{dy'}{ds} = \frac{q}{\gamma_s \beta_s^2 mc^2} \cdot \left(-\beta_s c \cdot B_{z0} \cdot x' + \beta_s c \cdot \frac{\partial B_r}{\partial r} \cdot x \right)$$

Electrostatic Einzel lens

In an electrostatic Einzel lens, we get, at the first order:

$$\vec{B} = \vec{0},$$
$$E_\theta = 0,$$
$$E_r = \frac{\partial E_r}{\partial r} \cdot r,$$
$$E_z = E_{z0}.$$

The linearized equations of motion become:

$$\frac{dx'}{ds} = \frac{q}{\gamma_s \beta_s^2 mc^2} \cdot \left(\frac{\partial E_r}{\partial r} \cdot x - E_{z0} \cdot x' \right)$$

$$\frac{dy'}{ds} = \frac{q}{\gamma_s \beta_s^2 mc^2} \cdot \left(\frac{\partial E_r}{\partial r} \cdot y - E_{z0} \cdot y' \right)$$

$$\frac{d\delta}{ds} = -\frac{q}{\gamma_s \beta_s^2 mc^2} \cdot \left(2 - \beta_s^2\right) \cdot E_{z0} \cdot \delta$$

In these equations, we must add the variation of the energy of the synchronous particle:

$$\frac{dT_s}{ds} = q \cdot E_{z0}.$$

RF accelerating cavities

In a RF accelerating cavity we get:

$$E_z(r,s,t)=E_{z0}(s)\cdot\left(1-\frac{1}{4}\cdot\left(\frac{1}{E_{z0}(s)}\cdot\frac{d^2E_{z0}(s)}{ds^2}+\frac{\omega^2}{c^2}\right)\cdot r^2\right)\cdot\cos(\omega\cdot t+\varphi),$$

$$E_r(r,s,t)=-\frac{1}{2}\cdot\frac{dE_{z0}(s)}{ds}\cdot\left(1-\frac{1}{8}\cdot\left(\frac{1}{E_{z0}(s)}\cdot\frac{d^2E_{z0}(s)}{ds^2}+\frac{\omega^2}{c^2}\right)\cdot r^2\right)\cdot r\cdot\cos(\omega\cdot t+\varphi),$$

$$B_\theta(r,s,t)=-\frac{\omega}{2\cdot c^2}\cdot E_{z0}(s)\cdot\left(1-\frac{1}{8}\cdot\left(\frac{1}{E_{z0}(s)}\cdot\frac{d^2E_{z0}(s)}{ds^2}+\frac{\omega^2}{c^2}\right)\cdot r^2\right)\cdot r\cdot\sin(\omega\cdot t+\varphi).$$

At the first ordre:

$$\begin{aligned}
E_z(r,s,t)&=E_{z0}(s)\cdot\cos(\omega\cdot t+\varphi),\\
E_r(r,s,t)&=-\frac{1}{2}\cdot\frac{dE_{z0}(s)}{ds}\cdot r\cdot\cos(\omega\cdot t+\varphi),\\
B_\theta(r,s,t)&=-\frac{\omega}{2\cdot c^2}\cdot E_{z0}(s)\cdot r\cdot\sin(\omega\cdot t+\varphi)\cdot
\end{aligned}$$

In Cartesian coordinates:

$$\begin{aligned}
E_x(r,s,t)&=-\frac{1}{2}\cdot\frac{dE_{z0}(s)}{ds}\cdot x\cdot\cos(\omega\cdot t+\varphi)=\frac{\partial E_x}{\partial x}\cdot x,\\
E_y(r,s,t)&=-\frac{1}{2}\cdot\frac{dE_{z0}(s)}{ds}\cdot y\cdot\cos(\omega\cdot t+\varphi)=\frac{\partial E_y}{\partial y}\cdot y,\\
B_x(r,s,t)&=\frac{\omega}{2\cdot c^2}\cdot E_{z0}(s)\cdot y\cdot\sin(\omega\cdot t+\varphi)=\frac{\partial B_x}{\partial y}\cdot y,\\
B_y(r,s,t)&=-\frac{\omega}{2\cdot c^2}\cdot E_{z0}(s)\cdot x\cdot\sin(\omega\cdot t+\varphi)=\frac{\partial B_y}{\partial x}\cdot x.
\end{aligned}$$

The linearized equations of motion become:

$$\begin{aligned}
\frac{dx'}{ds}&=\frac{q}{\gamma_s\beta_s^2mc^2}\cdot\left(-E_{z0}\cdot\cos\varphi\cdot x'+\left(\frac{\partial E_x}{\partial x}-\beta_sc\cdot\frac{\partial B_y}{\partial x}\right)\cdot x\right),\\
\frac{dy'}{ds}&=\frac{q}{\gamma_s\beta_s^2mc^2}\cdot\left(-E_{z0}\cdot\cos\varphi\cdot y'+\left(\frac{\partial E_y}{\partial y}+\beta_sc\cdot\frac{\partial B_x}{\partial y}\right)\cdot y\right),\\
\frac{d\delta}{ds}&=\frac{q}{\gamma_s\beta_s^2mc^2}\cdot\left(\frac{\partial E_z}{\partial z}\cdot z-\left(2-\beta_s^2\right)\cdot E_{z0}\cdot\cos\varphi\cdot\delta\right).
\end{aligned}$$

Element without special symmetry

Magnetic quad

In a magnetic quad:

$$\vec{E}=\vec{0}\,,\ B_x=G\cdot y \text{ and } B_y=G\cdot x$$

Then:

$$\frac{dx'}{ds}=-\frac{q}{\gamma_s\beta_s^2mc^2}\cdot\beta_s c\cdot G\cdot x\,,$$
$$\frac{dy'}{ds}=\frac{q}{\gamma_s\beta_s^2mc^2}\cdot\beta_s c\cdot G\cdot y\ .$$

Electrostatic quad

In an electrostatic quad:

$$\vec{B}=\vec{0}\,,\ E_x=-G\cdot x \text{ and } E_y=G\cdot y$$

Then:

$$\frac{dx'}{ds}=\frac{q}{\gamma_s\beta_s^2mc^2}\cdot G\cdot x$$
$$\frac{dy'}{ds}=\frac{q}{\gamma_s\beta_s^2mc^2}\cdot G\cdot y$$

CORE – HALO EVOLUTIONS ALONG the ACCELERATOR

1. Principle

A high intensity beam can be described as a combination of:
- the central core, very compact and dense, where linear forces are dominant, leaving the emittance unchanged
- the external halo, much less dense, where some particles have been sent after gaining extra energy and where nonlinear forces are dominant, leading to emittance increase.

Let us consider the case of a dense, uniform core where self-forces are strictly linear, surrounded by a non-uniform and very few dense halo. In such a configuration, the corehalo limit is clearly given by the location where the density gradient abruptly changes from small variations in the halo to a very steep (infinite) variation when arriving on the "wall" of the core uniform distribution.

For a more realistic distribution presenting a similar topology but where the density gradient continuously varies, the core-halo limit definition can be generalized as the location where there is the steepest density gradient variation, that is where the Laplacian of the density is maximum. In 1D, it corresponds to the second derivative's maximum (not to be confused with the second derivative's zero, which is the inflection point). **We will take this location of second derivative maximum as the core-halo limit**.

2. At a given position

At a given position along the accelerator and for a given coordinate (spatial or momentum), a histogram is calculated, allowing to obtain the beam density profile. From that, first and then second derivatives are calculated. In order to get rid of numerical noises and in the meantime preserving the abrupt variations of the profile that must be detected, smoothing of the original profile is not appropriate. The method of sliding derivative, calculated on the average of 10 points around the point of interest is used instead. As the halo is expected only at the external part of the beam, only positive maximas of the second derivative should be selected (For a constant profile, due to numerical noise, the second derivatives are close to zero but either positive or negative. That is why, only positive maximas that are much bigger (in absolute value) than negative minimas should be selected).

That procedure gives core-halo limits that correspond roughly to what can be seen intuitively when looking at a density profile. For a Gaussian profile characterised by the standard deviation σ, this limit is $\sigma\sqrt{3}$. For a pure square, triangular or parabolic profile, there is no halo as expected.

3. Along the accelerator

Once the core-halo limit is determined, the following quantities can be given along the accelerators:

- The core maximum size and the halo maximum size (to be considered instead of the classical beam envelope)
- The percentage of halo size $PHS = \frac{Halo\ size}{Total\ beam\ size}$
- The percentage of halo particles $PHP = \frac{Number\ of\ particles\ in\ the\ halo}{Total\ number\ of\ particles}$
- The halo density $HD = \frac{Number\ of\ particles\ in\ the\ halo}{Halo\ size}$
- Percentage of particles initially in the halo; for a given position considered as an initial one, this percentage is 100%, then decreases (while oscillating because of transfer with other planes), indicating the replenishment rate of halo particles.

*(Reference: Core-halo issues for a very high intensity beam, P. A. P. Nghiem, N. Chauvin, W. Simeoni Jr., D. Uriot, Applied Physics Letters, **104**, 074109 (2014).)*

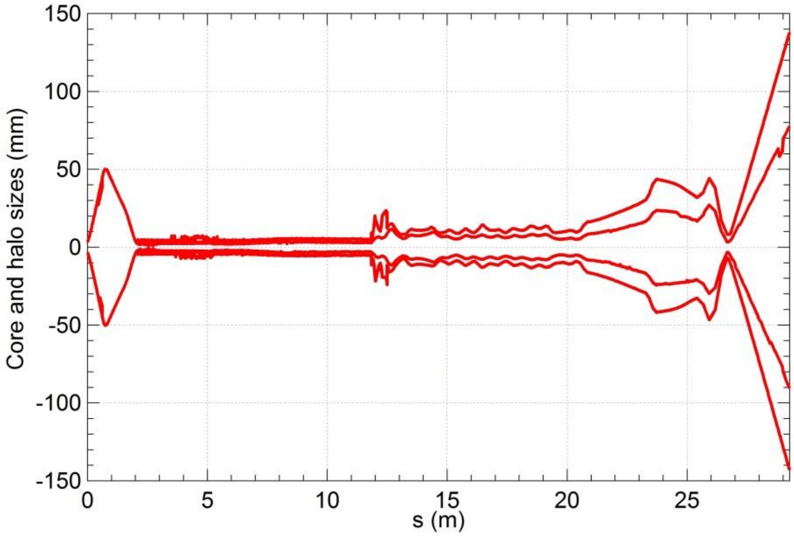


Figure: Core size (internal line) and halo size (external line) along the accelerator

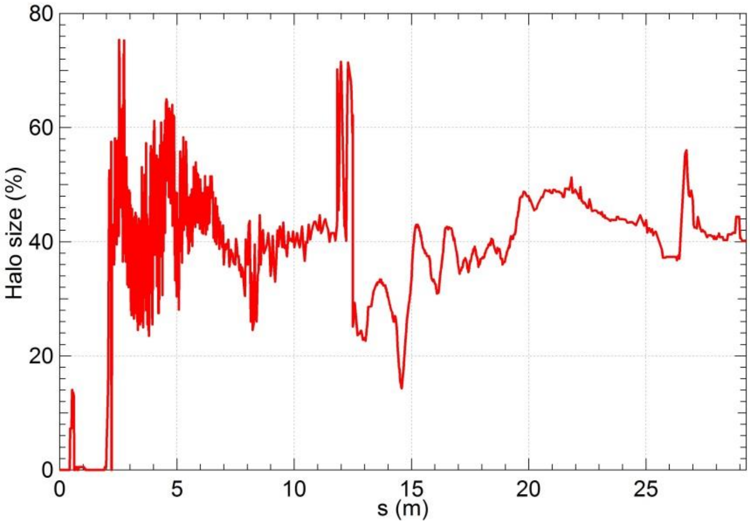


Figure: Percentage of halo size along the accelerator

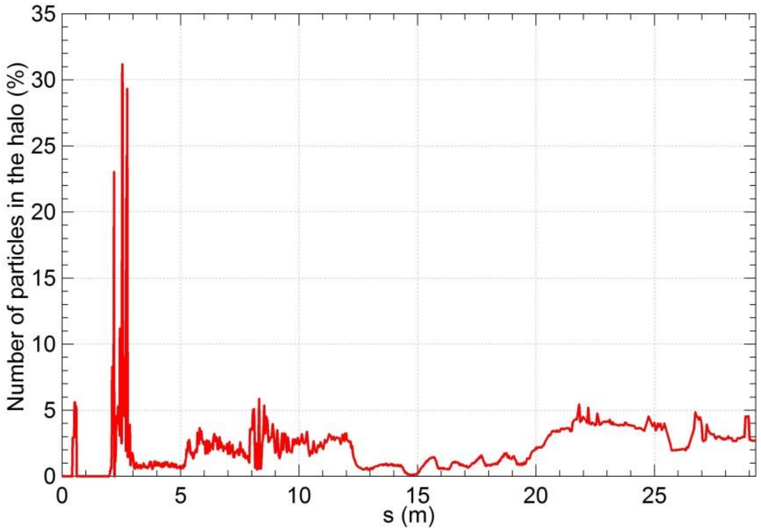


Figure: Percentage of halo particles along the accelerator

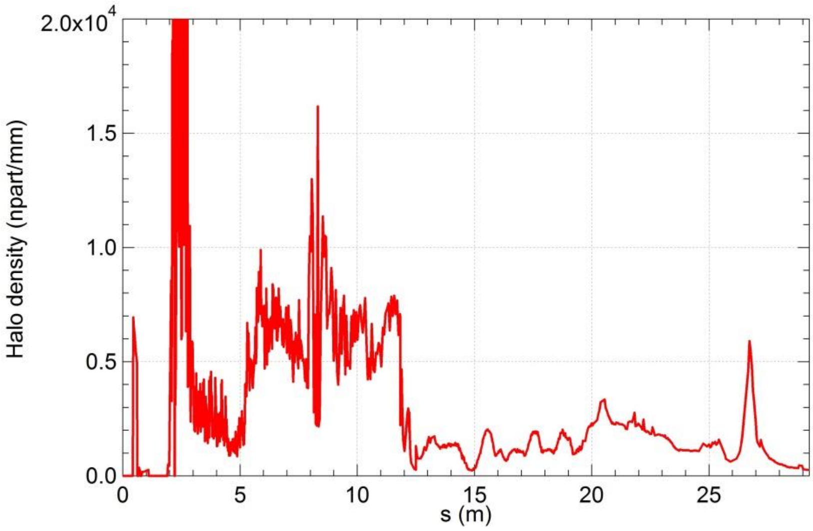


Figure: Halo density along the accelerator

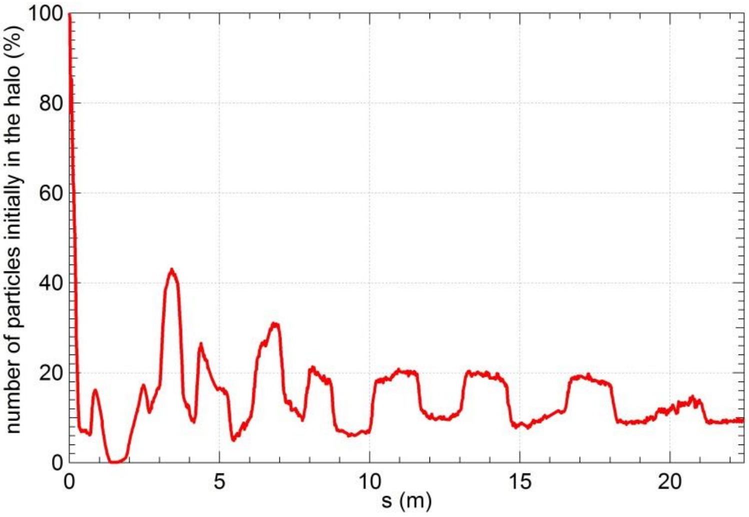


Figure: Percentage of particles initially in the halo along the SRF Linac

3D field development

(*Note written by Ciprian Plostinar*)

The 3D magnetic field components and their derivates for a multipole magnet in the region close to the axis can be expressed as given by the gradient of a scalar magnetic potential function, V.

The proposed solution in the literature for the scalar potential is of the form:

$$V_n(x,y,z)=(n!)^2\left(\sum_{q=0}^{\infty}(-1)^q\frac{G^{(2q)}(\mathrm{z})\left(x^2+y^2\right)^q}{4^q q!(n+q)!}\right)\left(\sum_{m=0}^{n}\frac{\sin\left(m\frac{\pi}{2}\right)x^{n-m}\ \mathrm{y}^{\mathrm{m}}}{m!(\mathrm{n-m})!}\right)$$

Where G(z) is the magnetic gradient along the longitudinal axis.

More explicitly, for a quadrupole (n=2), the scalar potential is:

$$V(x,y,z)=\left(G(z)-\frac{G''(\mathrm{z})\left(\mathrm{x}^2+y^2\right)}{12}+\frac{G''''(z)\left(\mathrm{x}^2+y^2\right)^2}{384}-\frac{G''''''(z)\left(\mathrm{x}^2+y^2\right)^3}{23040}\right)xy$$

And the three magnetic field components in Cartesian coordinates are given by:

$$B_x=\frac{\partial V}{\partial x}$$

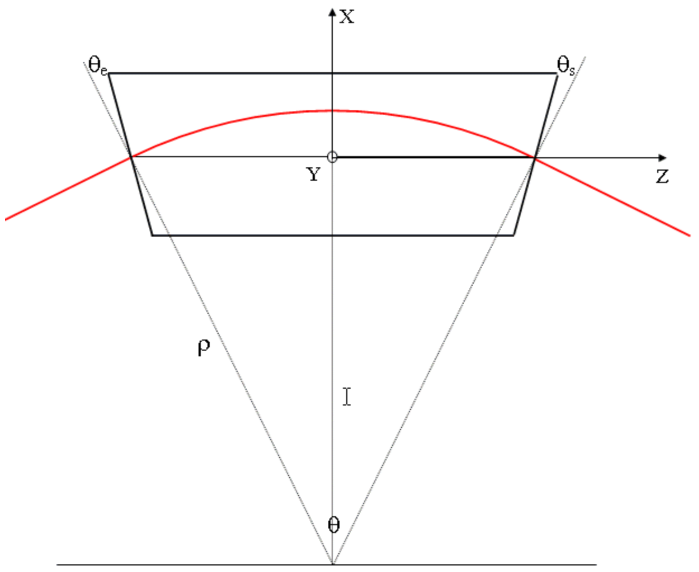
$$By=\frac{\partial V}{\partial y}$$

$$Bz=\frac{\partial V}{\partial z}$$

Bend error treatment

DY :

y_e -= DY
y_s += DY

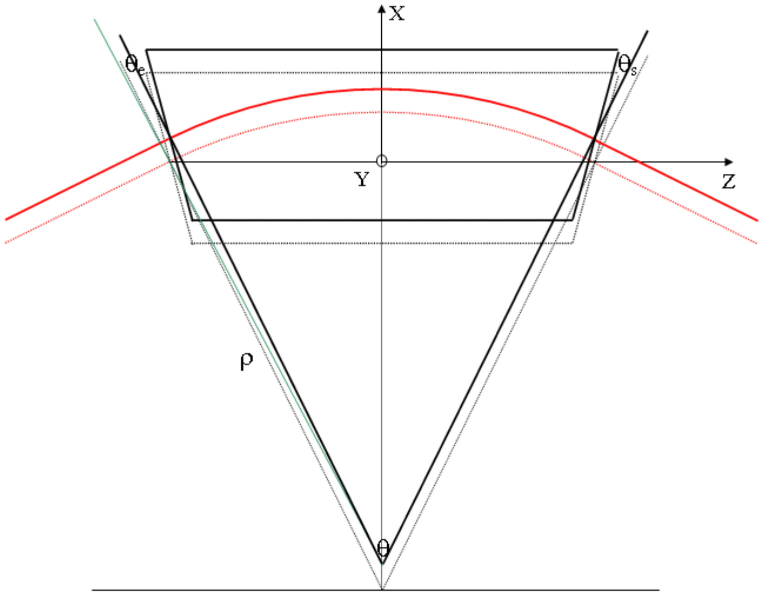


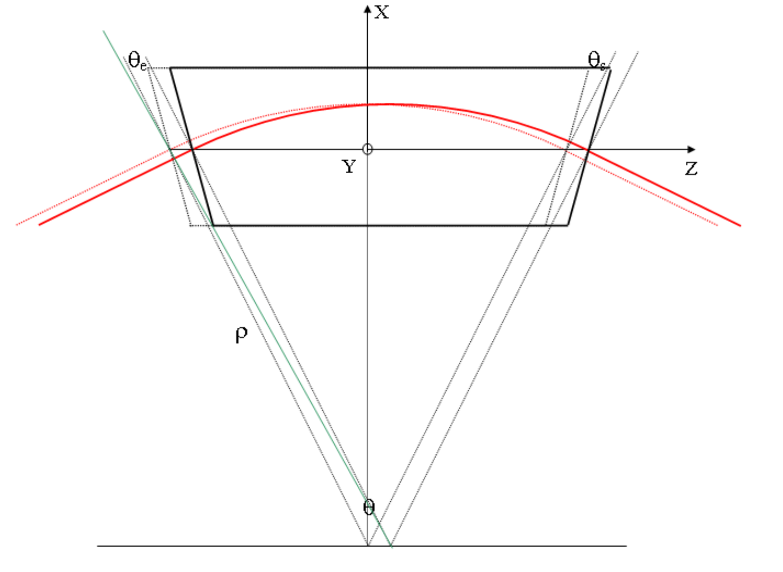
DX :

Input drift (*DX*·sin(*θ*/2)

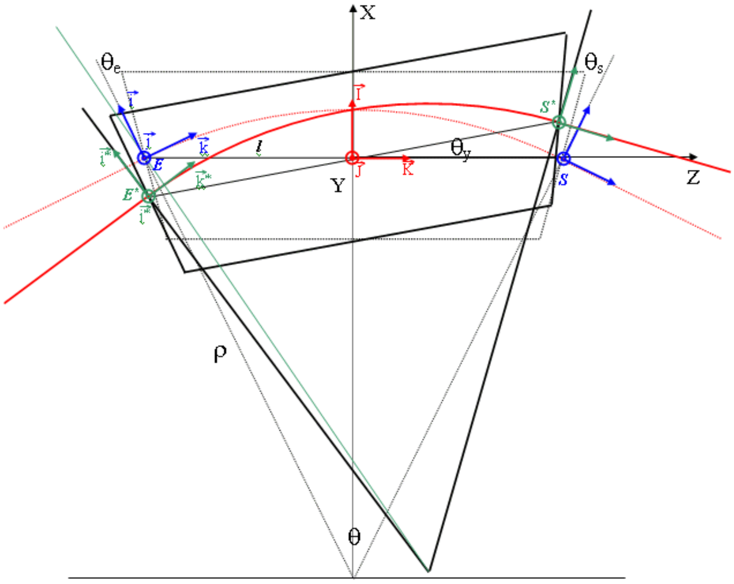
$$\begin{cases} x_e - = DX \cdot \cos(\theta/2) \\ x_s + = DX \cdot \cos(\theta/2) \end{cases}$$

Output drift (*DX*·sin(*θ* /2)



DZ : Input drift (*DZ*·cos(*θ* /2) $\begin{cases} x_e += DZ \cdot \sin(\theta/2) \\ x_s += DZ \cdot \sin(\theta/2) \end{cases}$ Output drfit (-*DZ*·cos(*θ*/2)	

Input of the dipole



At the input:

$$\begin{pmatrix} \vec{I} \\ \vec{J} \\ \vec{K} \end{pmatrix} = \begin{pmatrix} \cos\theta/2 & 0 & \sin\theta/2 \\ 0 & 1 & 0 \\ -\sin\theta/2 & 0 & \cos\theta/2 \end{pmatrix} \cdot \begin{pmatrix} \vec{i}_e \\ \vec{j}_e \\ \vec{k}_e \end{pmatrix} = T_{r_e \to R} \cdot \begin{pmatrix} \vec{i}_e \\ \vec{j}_e \\ \vec{k}_e \end{pmatrix}$$

Mθ, the rotation matrix of the dipole (around an axis, meaning given by the law of the corkscrew):

$$M_{\theta_x}=\begin{pmatrix}1 & 0 & 0\\ 0 & \cos\theta_x & \sin\theta_x\\ 0 & -\sin\theta_x & \cos\theta_x\end{pmatrix}\approx\begin{pmatrix}1 & 0 & 0\\ 0 & 1-\frac{1}{2}\theta_x^2 & \theta_x\\ 0 & -\theta_x & 1-\frac{1}{2}\theta_x^2\end{pmatrix}$$

$$M_{\theta_y}=\begin{pmatrix}\cos\theta_y & 0 & -\sin\theta_y\\ 0 & 1 & 0\\ \sin\theta_y & 0 & \cos\theta_y\end{pmatrix}\approx\begin{pmatrix}1-\frac{1}{2}\theta_y^2 & 0 & -\theta_y\\ 0 & 1 & 0\\ \theta_y & 0 & 1-\frac{1}{2}\theta_y^2\end{pmatrix}$$

$$M_{\theta_z}=\begin{pmatrix}\cos\theta_z & \sin\theta_z & 0\\ -\sin\theta_z & \cos\theta_z & 0\\ 0 & 0 & 1\end{pmatrix}\approx\begin{pmatrix}1-\frac{1}{2}\theta_z^2 & \theta_z & 0\\ -\theta_z & 1-\frac{1}{2}\theta_y^2 & 0\\ 0 & 0 & 1\end{pmatrix}$$

We also gave the second order development of the matrix (be careful, in this case, the determinant is not zero).
The matrix expressing the total rotation of the dipole is the product of the three rotation matrices. However, this product is not commutative.
On the other hand, assuming that the angles of rotation are small and remaining at the second order, we can then find a simplified matrix for the rotation of the elements:

$$M_\theta\approx\begin{pmatrix}1-\frac{1}{3}\cdot\left(\theta_x^2+\theta_y^2+\theta_z^2\right) & \theta_z & -\theta_y\\ -\theta_z & 1-\frac{1}{3}\cdot\left(\theta_x^2+\theta_y^2+\theta_z^2\right) & \theta_x\\ \theta_y & -\theta_x & 1-\frac{1}{3}\cdot\left(\theta_x^2+\theta_y^2+\theta_z^2\right)\end{pmatrix}$$

It is this matrix that we will use in PARTRAN. It is obviously approximated, but most certainly less than knowing the magnitude of the errors. Note, however, that its determinant is non-zero (in order 4). The choice of the diagonal is not unique. It is however the one that minimizes the determinant for equal angles of rotation in all directions.

The momemtum $\vec{p}$ in the fram $\left(\vec{i}_e,\vec{j}_e,\vec{k}_e\right)$ become then $\vec{p}^{\,*}$ in the frame $\left(\vec{i}_e^{\,*},\vec{j}_e^{\,*},\vec{k}_e^{\,*}\right)$:

$$\vec{p}^{\,*}=M_{-\theta}\cdot\vec{p}\,.\ (1)$$

A particle of coordinates $(x_e, y_e, 0)$ in the frame $(\vec{i}_e, \vec{j}_e, \vec{k}_e)$ centered on E get then coordinates (x_e^*, y_e^*, z_e^*) in the frame $(\vec{i}_e^*, \vec{j}_e^*, \vec{k}_e^*)$ centered on E^* such as :

$$\begin{pmatrix} x_e^* \\ y_e^* \\ z_e^* \end{pmatrix} = T_{R\to r_e} \cdot \left((Id - M_{-\theta}) \cdot \begin{pmatrix} 0 \\ 0 \\ -l \end{pmatrix} + M_{-\theta} \cdot T_{r_e \to R} \cdot \begin{pmatrix} x_e \\ y_e \\ 0 \end{pmatrix} \right)$$

l is the half length of the rope at the main path of the magnet.
We get : $l = \rho \cdot \sin \varphi/2 \cdot$

then :

$$\begin{pmatrix} x_e^* \\ y_e^* \\ z_e^* \end{pmatrix} = \begin{pmatrix} (1-\varepsilon) \cdot x_e - (\theta_z \cos\varphi/2 + \theta_x \sin\varphi/2) \cdot y_e - (\theta_y \cos\varphi/2 - \varepsilon \cos\varphi/2) \cdot l \\ (1-\varepsilon) \cdot y_e + (\theta_z \cos\varphi/2 + \theta_x \sin\varphi/2) \cdot x_e + \theta_x \cdot l \\ -\theta_y \cdot x_e - (\theta_z \sin\varphi/2 - \theta_x \cos\varphi/2) \cdot y_e - (\theta_y \sin\varphi/2 + \varepsilon \cos\varphi/2) \cdot l \end{pmatrix} \qquad (2)$$

with : $\varepsilon = \frac{1}{3} \cdot (\theta_x^2 + \theta_y^2 + \theta_z^2)$.

The input procedure is as follows::

A particle enters the dipole with the coordinates in the frame $(x_e, x_e', y_e, y_e', phi_e, E_e)$ in the frame $(\vec{i}_e, \vec{j}_e, \vec{k}_e)$.

Its reduced momemtum $\vec{p}_e = (x_e' \cdot p_{ze}, y_e' \cdot p_{ze}, p_{ze})$ in $(\vec{i}_e, \vec{j}_e, \vec{k}_e)$ is calculated with $p_{ze} = \sqrt{\frac{\gamma^2 - 1}{1 + x_e'^2 + y_e'^2}}$.

It is then transformed into $\vec{p}_e^*$ with (1), then we use: $x_e'^* = \frac{p_{xe}^*}{p_{ze}^*}$, …

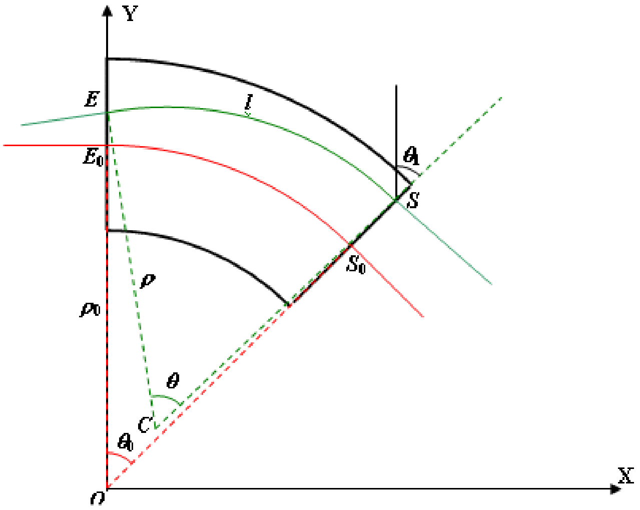
Its position in frame $(\vec{i}_e^*, \vec{j}_e^*, \vec{k}_e^*)$ i obtained with (2).

A drift length $-z_e^*$ has to be added in front of the bend matrix.

Transport in the bend

The entry corner is treated as a thin lens "like in PARMILA" taking into account the influence of the magnetic leakage field.

The dipole matrix as it was used before is replaced by a nonlinear transport in the frame $(\vec{i}_e^*, \vec{j}_e^*, \vec{k}_e^*)$. The calculation is detailed below:



The coordinates of the points are given in the coordinate frame (X, Y) centered on O.

$$E_0(0,\rho_0),\ E(0,\rho_0+x_e),\ S_0(\rho_0\cos\theta_0,\rho_0\sin\theta_0)$$

The trajectory of the particle is a circle of radius of curvature ρ :

$$\rho=\frac{p}{qB}=\rho_0\cdot\frac{1+\delta p/p_0}{1+\delta B/B_0}$$

Find the coordinates of the center of this circle C :

$$\begin{cases} x'_e=\dfrac{X_C}{Y_E-Y_C} \\ (X_E-X_C)^2+(Y_E-Y_C)^2=\rho^2 \end{cases}$$

$$\Rightarrow \qquad \begin{cases} X_C=x'_e\cdot\dfrac{\rho}{\sqrt{1+x'^2_e}} \\ Y_C=Y_E-\dfrac{\rho}{\sqrt{1+x'^2_e}} \end{cases}$$

Find the coordinates of the exit position S of the particle:

$$\begin{cases} X_S=Y_S\cdot\tan\theta_0 \\ (X_S-X_C)^2+(Y_S-Y_C)^2=\rho^2 \end{cases}$$

$$\Rightarrow \qquad \begin{cases} X_S=\sin\theta_0\cdot\left[X_C\sin\theta_0+Y_C\cos\theta_0+\sqrt{\rho^2-(X_C\cos\theta_0-Y_C\sin\theta_0)^2}\right] \\ Y_S=\cos\theta_0\cdot\left[X_C\sin\theta_0+Y_C\cos\theta_0+\sqrt{\rho^2-(X_C\cos\theta_0-Y_C\sin\theta_0)^2}\right] \end{cases}$$

We deduce its exit position, $x_s = SS_0$, of the particle in the reference frame linked to the reference trajectory:

$$x_s = X_C \sin\theta_0 + Y_C \cos\theta_0 + \sqrt{\rho^2 - (X_C \cos\theta_0 - Y_C \sin\theta_0)^2} - \rho_0$$

Let us calculate the angle of the particle at exit in the reference frame linked to the reference trajectory $x_s' = \arctan(\theta_0 - \theta_1)$:

$$\tan\theta_1 = \frac{X_S - X_C}{Y_S - Y_C},$$

$$\Rightarrow \qquad x_s' = \frac{(Y_S - Y_C)\cdot\tan\theta_0 - (X_S - X_C)}{(X_S - X_C)\cdot\tan\theta_0 + (Y_S - Y_C)}$$

Let us calculate the length of the trajectory: $l = \rho\cdot\theta$

$$\tan\theta = \frac{(X_S - X_C) + (Y_S - Y_C)\cdot x_e'}{(Y_S - Y_C) - (X_S - X_C)\cdot x_e'}$$

$$\Rightarrow \qquad l = \rho\cdot\arctan\left(\frac{(X_S - X_C) + (Y_S - Y_C)\cdot x_e'}{(Y_S - Y_C) - (X_S - X_C)\cdot x_e'}\right).$$

In the vertical plane, the magnet can be considered as a drift:

$$\begin{cases} y_s' = y_e' \\ y_s = y_e + y_e'\cdot l \end{cases}.$$

Bend output

At the exit of the dipole, the particle is at the position $(x_s^*, x_s'^*, y_s^*, y_s'^*, phi_s^*, E_s^*)$.

We get:

$$\begin{pmatrix} \vec{I}^* \\ \vec{J}^* \\ \vec{K}^* \end{pmatrix} = \begin{pmatrix} \cos\theta/2 & 0 & -\sin\theta/2 \\ 0 & 1 & 0 \\ \sin\theta/2 & 0 & \cos\theta/2 \end{pmatrix} \cdot \begin{pmatrix} \vec{i}_s^* \\ \vec{j}_s^* \\ \vec{k}_s^* \end{pmatrix} = T_{r_s^* \to R^*} \cdot \begin{pmatrix} \vec{i}_s^* \\ \vec{j}_s^* \\ \vec{k}_s^* \end{pmatrix}$$

Vector $\vec{p}^*$ in the ssytem $(\vec{i}_s^*, \vec{j}_s^*, \vec{k}_s^*)$ become the $\vec{p}$ in the frame $(\vec{i}_s, \vec{j}_s, \vec{k}_s)$:

$$\vec{p} = M_\theta \cdot \vec{p}^*. \qquad (3)$$

A particle of coordonate $(x_s^*, y_s^*, 0)$ in the frame $(\vec{i}_s^*, \vec{j}_s^*, \vec{k}_s^*)$ centered on S^* get then the coodonates (x_s, y_s, z_s) in the frame $(\vec{i}_s, \vec{j}_s, \vec{k}_s)$ centered onré S such as :

$$\begin{pmatrix} x_s \\ y_s \\ z_s \end{pmatrix} = T_{R\to r_s^*} \cdot \left(\left(Id - M_\theta\right) \cdot \begin{pmatrix} 0 \\ 0 \\ l \end{pmatrix} + M_\theta \cdot T_{r_s^*\to R} \cdot \begin{pmatrix} x_s^* \\ y_s^* \\ 0 \end{pmatrix} \right) \quad (4)$$

then :

$$\begin{pmatrix} x_s \\ y_s \\ z_s \end{pmatrix} = \begin{pmatrix} (1-\varepsilon)\cdot x_s^* + \left(\theta_z \cos\varphi/2 - \theta_x \sin\varphi/2\right)\cdot y_s^* - \left(\theta_y \cos\varphi/2 - \varepsilon\cos\varphi/2\right)\cdot l \\ (1-\varepsilon)\cdot y_s^* - \left(\theta_z \cos\varphi/2 - \theta_x \sin\varphi/2\right)\cdot x_s^* + \theta_x \cdot l \\ \theta_y \cdot x_s^* - \left(\theta_z \sin\varphi/2 + \theta_x \cos\varphi/2\right)\cdot y_s^* + \left(\theta_y \sin\varphi/2 + \varepsilon\cos\varphi/2\right)\cdot l \end{pmatrix} \qquad (2)$$

with : $\varepsilon = \frac{1}{3}\cdot\left(\theta_x^2 + \theta_y^2 + \theta_z^2\right)$.

The output procedure is as follows:
A particle leaves the dipole with the coordinates $\left(x_s^*, x_s'^*, y_s^*, y_s'^*, phi_s^*, E_s^*\right)$ in the frame $\left(\vec{i}_s^*, \vec{j}_s^*, \vec{k}_s^*\right)$.
Its reduced momentum $\vec{p}_s^* = \left(x_s'^* \cdot p_{zs}^*, y_s'^* \cdot p_{zs}^*, p_{zs}^*\right)$ dans $\left(\vec{i}_s^*, \vec{j}_s^*, \vec{k}_s^*\right)$ is calculated with $p_{zs}^* = \sqrt{\frac{\gamma^2 - 1}{1 + x_s'^{*2} + y_s'^{*2}}}$.

The it is transformed in $\vec{p}_s$ avec (3), then we take : $x_s' = \frac{p_{xs}}{p_{zs}}$, …

Its position in the frame $\left(\vec{i}_s, \vec{j}_s, \vec{k}_s\right)$ is obtained with (2).

A drift lentgh $-z_s$ has to be added.

RF cavity transient analysis with TraceWin

Some functionalities are available in TraceWin code in order to study the cavity behavior and more generally the linac behavior during the transient time of the cavity RF fields or of the beam current pulse. To have some more details about motivations, objectives and results of this functionality, have a look about the following publication:

Dynamic compensation of an rf cavity failure in a superconducting linac
Jean-Luc Biarrotte and Didier Uriot
PHYSICAL REVIEW SPECIAL TOPICS - ACCELERATORS AND BEAMS ***11****, 072803 (2008)*

First order model

The spoke and elliptical RF superconducting cavities operate with the $TM_{010\text{-}\pi}$ mode, which produces an accelerating RF voltage on the cavity axis. Using the RLC circuit analogy, the behavior of the “cavity + beam” system can be described to first order by the following equation:

$$(1)\qquad \frac{d\tilde{V}_C(t)}{dt} = \frac{\omega\,(r/Q)}{4}\left(2\tilde{I}_G(t)+\tilde{I}_B(t)\right) - \frac{\omega\,(1-j\,\tan\psi(\mathrm{t}))}{2\,Q_L}\tilde{V}_C(t)$$

$\tilde{V}_C$ represents the low frequency component of the accelerating voltage created in the cavity at the operating RF frequency $\omega = 2\pi f$. Its amplitude $V_C = \left|\tilde{V}_C\right| \approx \left|\int E_Z(z)\,e^{j\omega z/\beta c}dz\right|$ gives the voltage seen by a particle, with velocity β and optimal phase, while crossing the cavity (the Transit Time Factor is included). Its phase $\varphi_c = \arg(\tilde{V}_C)$ gives the phase of this accelerating voltage, compared to the reference phase of the system which is chosen here to be the phase giving a 0° synchronous phase. In the case where the cavities are modeled using punctual gaps, this phase is therefore directly equal to the synchronous phase.

$\tilde{I}_B$ represents the low frequency component of the beam current crossing the cavity. For short bunches, its amplitude is given by $I_B = \left|\tilde{I}_B\right| \approx 2\,I_0$, where *I0* is the beam mean current. Its phase $\varphi_B = \arg(\tilde{I}_B) = \pi$ by definition of the reference phase.

$\tilde{I}_G$ represents the low frequency component of the current created by the RF power generator. Its amplitude is given by $I_G = \left|\tilde{I}_G\right| = 2\,\sqrt{P_{inc}/((r/Q)\,Q_i)}$, where *Pinc* is the incident RF power, *(r/Q)* the cavity shunt impedance (linac definition), and *Qi* the incident coupling. Its phase is noted $\varphi_G = \arg(\tilde{I}_G)$

.

The resonant frequency of the cavity f_{cav} is always varying because of various perturbations (microphonics, Lorentz detuning…), and these perturbations cannot be neglected because of the narrow bandwidth of the loaded cavity. These fluctuations are included in (1) through the detuning angle of the cavity, $\tan(\Psi(t)) \cong 2Q_L\,(f_{cav}(t) - f)/f$, where QL is the quality factor of the loaded cavity $(1/Q_L = 1/Q_0 + 1/Q_i + 1/Q_t)$. We choose to describe these frequency fluctuations with a first order model using the following equation (2), where $\Delta f_{CTS}(t)$, $\Delta f_L(t)$, $\sum_i \Delta f_{MIC_i}(t)\sin(2\pi f_{MIC_i}t)$ are the detuning contributions from cold tuning system management, Lorentz forces, and microphonics respectively, k_L is the Lorentz force detuning coefficient in *Hz/(MV/m)*^{2}, and τ_m is the mechanical time constant of the cavity.

(2) $$f_{cav}(t) = f + \Delta f_L(t) + \Delta f_{CTS}(t) + \sum_i \Delta f_{MIC_i}(t)\sin\left(2\pi f_{MIC_i} t\right)$$ with

$$\frac{d\,\Delta f_L(t)}{dt} = \frac{1}{\tau_m}\left(\frac{10^{-12}k_L}{L_{acc}{}^2}V_C{}^2(t) - \Delta f_L(t)\right)$$

The solution of the coupled equations (1) and (2) gives the transient evolution of the accelerating voltage $\tilde{V}_C(t)$ in the cavity. More details can be found in.

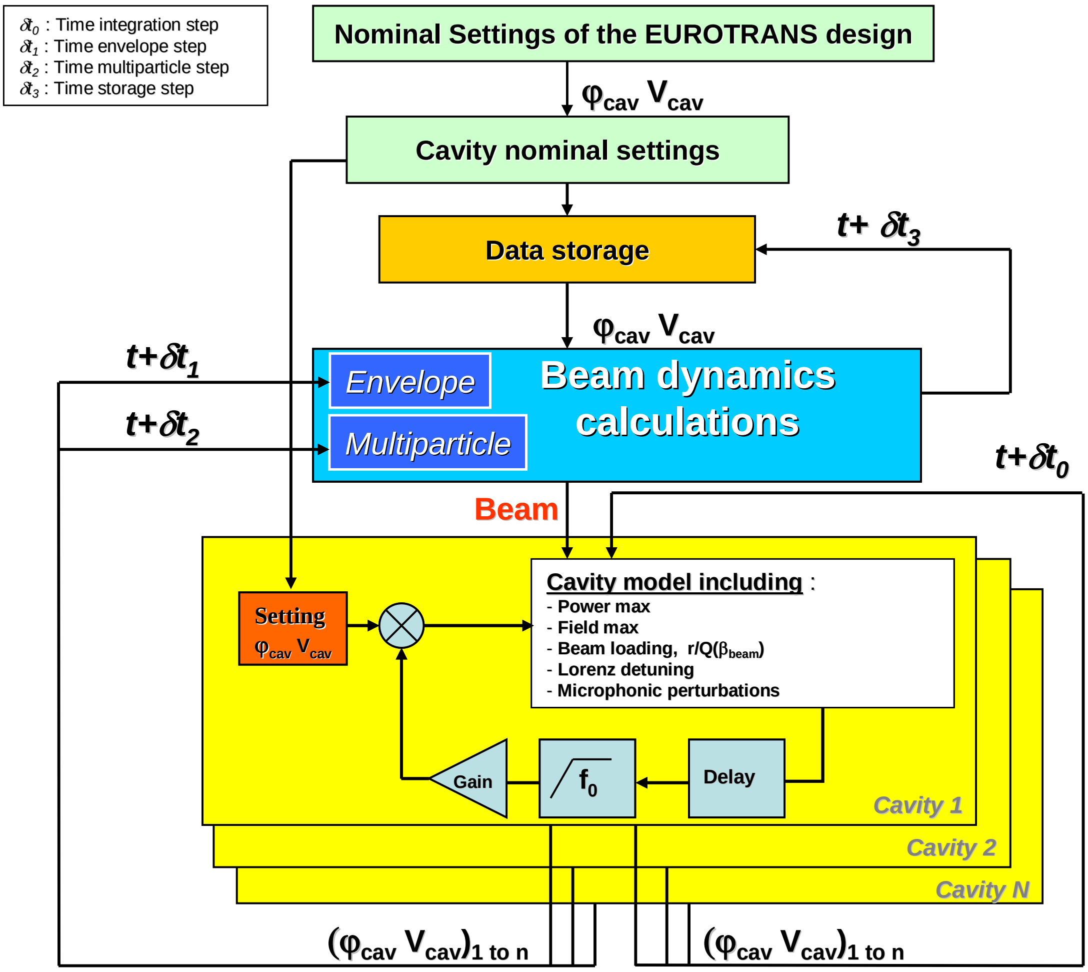
Modelling the RF regulation loop

Due to various perturbations (cavity frequency variations, beam transients…), the accelerating voltage $\tilde{V}_C(t)$ produced in the cavity from (1) + (2) is not stable at all. In order to regulate the accelerating field and phase, a regulation loop is required, called “Low-level RF” (LLRF) regulation loop. Its role is to monitor the field produced in the cavity using a capacitive probe, perform an adequate treatment of the signal, compare it to the desired (V_C, φ_C) set-point, and use the detected error to react on the RF high-power amplifier stage via a PID (Proportional / Integrative / Derivative) controller.
A rough but meaningful modeling of such a loop has been defined. The main elements are:

- a comparator that monitors the probe signal and compare it to the desired set-point; at this location of the loop, the signals have been digitalized, and the comparison is made using I/Q signals;
- a PID controller (we will here only use the gain “P” as a first approach);
- a delay or/and a low-pass filter to account for the bandwidth of the whole system; the typical order of magnitude is a few μs delays, and a few kHz cut-off frequency.

Simulation code development

The cavity model describing the transient RF cavity behaviour has first been included into the beam envelope and multiparticle TraceWin code and successful validate using tests performed to compare the results produced by this new cavity module with previous Simulink results.
TraceWin code calculates the transport of the reference particle (envelope) or the beam distribution (multiparticle) through the cavities of the linac, and is by default a “static” tool. It has thus been modified to be able to include the “time” variable, and therefore perform simulations at different times. The architecture of the TraceWin code “transient calculation” option is shown in fgure below. Several time steps are involved in the process. From the initial condition, at $t=0$, where all the cavities are set to their nominal RF fields and phases, different time-based iterations are performed:
every δt_0 (time integration step), a new couple of RF field amplitude and phase is evaluated in each cavity of the linac according to RF cavity model; every δt_1 (time envelope step, $\delta t_1 \geq \delta t_0$), a new beam transport calculation is performed through the linac (envelope transport), using the RF field characteristics (amplitude and phase) obtained at this time in each cavity (which can be modelled either by multi-gap or field map element); this calculation updates the beam characteristics at each linac location; every δt_2 (time multiparticle step, $\delta t_2 \geq \delta t_1$), a multiparticle transport simulation is performed; this calculation updates the beam characteristics at each linac location;
every δt_3 (time storage step, $\delta t_3 \geq \delta t_2$), all the linac and beam characteristics at each location are stored.



Code architecture.

The computation time strongly depends on the choice of these different time steps, and especially on δt_1 and δt_2; δt_3 is fixed by the available memory. Typically, to simulate accurately a 10 ms linac behavior, the time steps are respectively chosen to δt_0=1 ns, δt_1=1 µs, δt_2=10 µs and δt_3=10 µs..
Finally, let’s note that for each kind of cavity of the linac, a file has to be created in order to indicate to the transport code its main characteristics. These files have to contain also the feedback loop parameters, which can be different according to the cavity type. Some extra parameters as microphonic frequencies and amplitudes can be added if needed.

Cavity parameter files

Lacc 0.29792	*; Physical accelerating length of the cavity [m]*
rsqN 220	*; Optimal shunt impedance [Ω]*
rsQ0 4181.98	*;* $^{(*)}$
rsQ1 -81304.21	*;* $^{(*)}$
rsQ2 612270.40	*;* $^{(*)}$
rsQ3 -2303524.54	*;* $^{(*)}$
rsQ4 4695528.24	*;* $^{(*)}$
rsQ5 -4989523.30	*;* $^{(*)}$
rsQ6 2175215.22	*;* $^{(*)}$
Kl -8	*; Lorenz factor [Hz/(MV/m)*^{2}*]*

```
To_m 0.002                  ; Cavity mechanical constant
NMIC 0                      ; Number of microphonic frequency perturbations to consider
FMIC 100 600 1000           ; List of microphonic frequencies [Hz]
DFMIC 30 10 5               ; List of microphonic amplitudes
Rres 10e-9                  ; Niobium surface resistance [Ω]
T 4.2                       ; Cryogenic operating temperature [K]
Tc 9.2                      ; Nobium critical temperature [K]
G 100                       ; Form factor [Ω]
A 0                         ; To define Qo as function of accelerating field (**)
Qt 1e12                     ; Transmit couplage
Time_Start_Cav 0            ; Time [s] when the RF cavity is set on
Time_Stop_Cav 0.005         ; Time [s] when the RF cavity is set off
Time_Start_Beam 0.0         ; Time [s] when the beam pulse is started
Time_Stop_Beam 100.0        ; Time [s] when the beam pulse is stopped
Gain 20                     ; Feed-back gain
Freq_BP_loop 10000          ; Low filter frequency cut (Hz)
PowMax 30000                ; Max power from power supply (W)
Retard_Loop 0               ; Feed-back delay (s)
MargeField 0.7              ; field margin, here 70% available
MargePhase 10               ; RF phase margin, here RF phase can be adjusted of +/-10°
Fast_start                  ; Allow to avoid transient time need to start cavities at t=0
```

(*) *rsQ* = *rsqN*, but if *rsqN* equal to *0* then

$$rsQ = rsQ0 + \beta \cdot rsQ1 + \beta^2 \cdot rsQ2 + \beta^3 \cdot rsQ3 + \beta^4 \cdot rsQ4 + \beta^5 \cdot rsQ5 + \beta^6 \cdot rsQ6$$

With β, beam reduced velocity

(**) $Q_{olow} = Q_o - \left(A \,.\, Q_o \,.\, E_{acc}\right)$, with $Q_o = \dfrac{G}{Rs}$

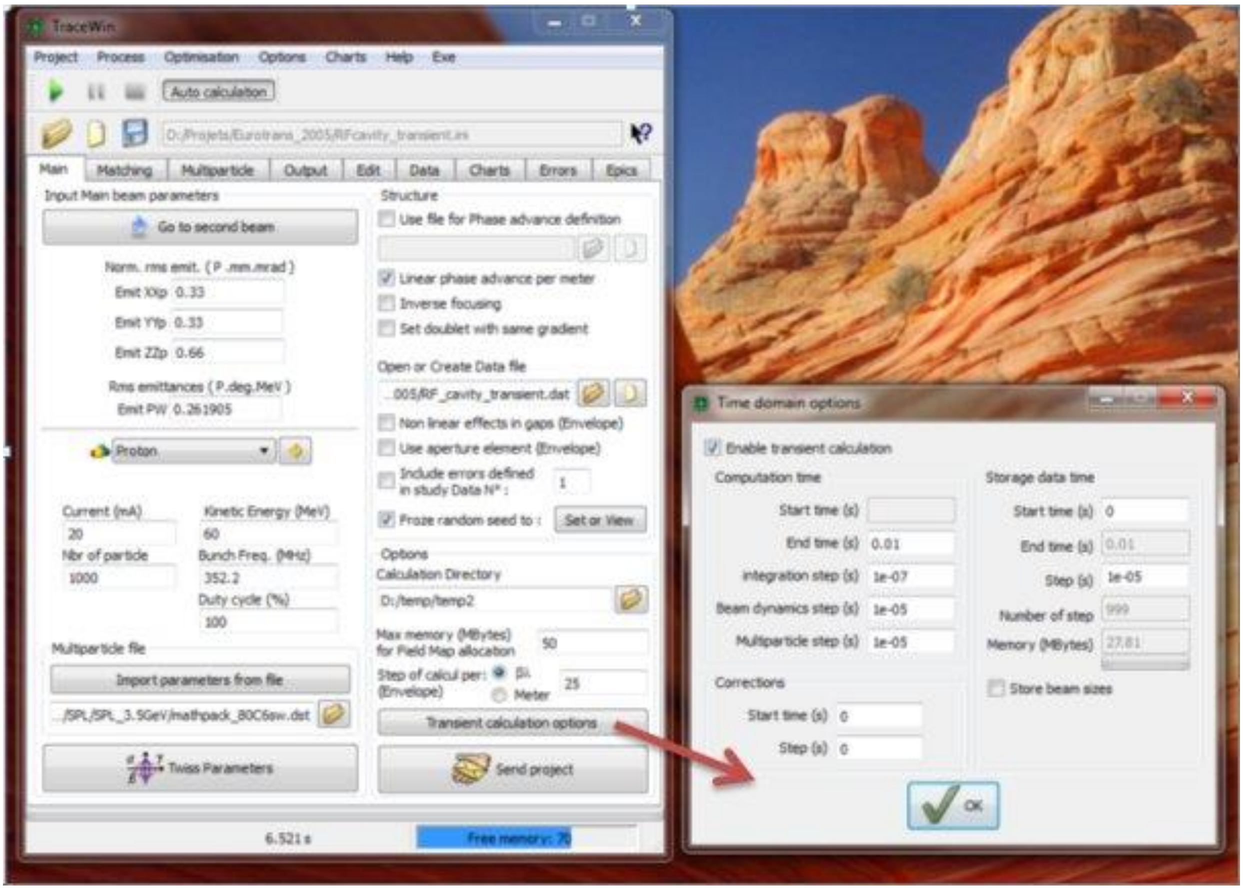
To cancel a line value put *';'* at the beginning

Examples

In this example (See "*HELP*" menu at the top then "*Open example files*" and open "*Cavity_transient*" project file), the second spoke cavity failed at 5 ms. The cavity failure parameters is set in "*spoke_25MV_failed.cav*" file and the nominal cavity is described in the "*spoke_25MV_nominal.cav*" file.

CAVITY_PARAMETERS spoke_25MV_nominal.cav
FIELD_MAP 100 597.92 -124.88 50 0 1 0 0 spoke_25MV

Cavity parameters file can be applied on FIELD_MAP, NCELLS, GAP elements.



D:/Projets/Eurotrans_2005/RFcavity_transient.ini

Input Main beam parameters

Norm. rms emit. (P .mm.mrad)

Emit XXp 0.33

Emit YYp 0.33

Emit ZZp 0.66

Rms emittances (P.deg.MeV)

Emit PW 0.261905

Current (mA) 20

Kinetic Energy (MeV) 60

Nbr of particle 1000

Bunch Freq. (MHz) 352.2

Duty cycle (%) 100

Multiparticle file

../SPL/SPL_3.5GeV/mathpack_80C6sw.dst

Structure

☐ Use file for Phase advance definition

☑ Linear phase advance per meter

☐ Inverse focusing

☐ Set doublet with same gradient

Open or Create Data file

...005/RF_cavity_transient.dat

☐ Non linear effects in gaps (Envelope)

☐ Use aperture element (Envelope)

☐ Include errors defined in study Data N° : 1

☑ Froze random seed to :

Options

Calculation Directory

D:/temp/temp2

Max memory (MBytes) for Field Map allocation 50

Step of calcul per: (Envelope) ● βλ ○ Meter 25

☑ Enable transient calculation

Computation time

Start time (s)

End time (s) 0.01

integration step (s) 1e-07

Beam dynamics step (s) 1e-05

Multiparticle step (s) 1e-05

Corrections

Start time (s) 0

Step (s) 0

Storage data time

Start time (s) 0

End time (s) 0.01

Step (s) 1e-05

Number of step 999

Memory (MBytes) 27.81

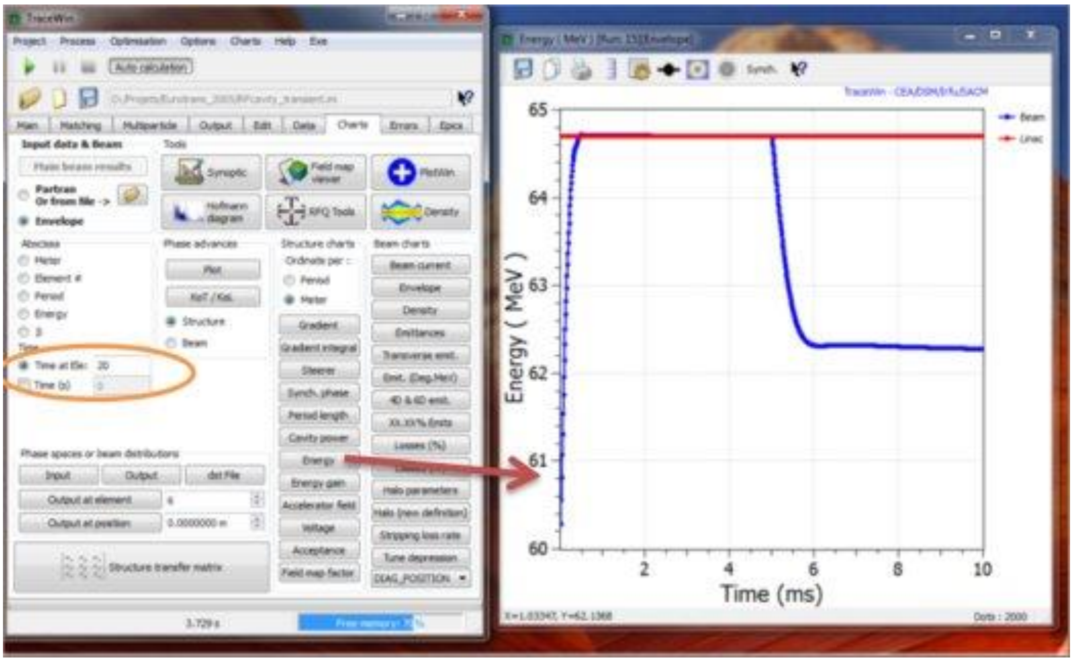
☐ Store beam sizes

In this box, all the parameters needed to perform the transient simulation have to be set. Don’t forget to “*Enable transient simulation*”

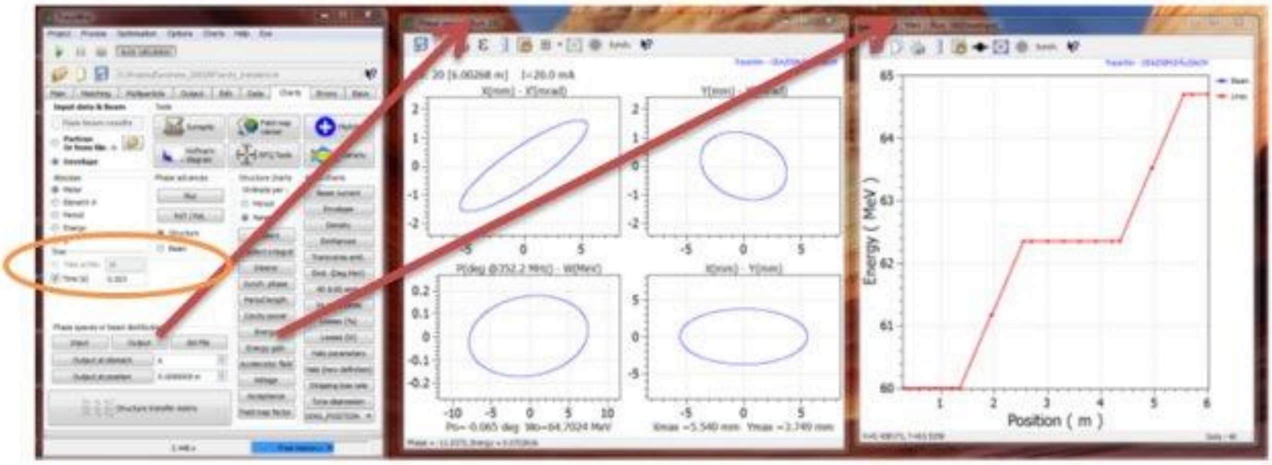
Simulations can be performed with or without multiparticle option, but in a first time it’s strongly recommended to start only in envelope mode.

Global results can be observe either as a function of the time of the simulation or as at a define time at position elements

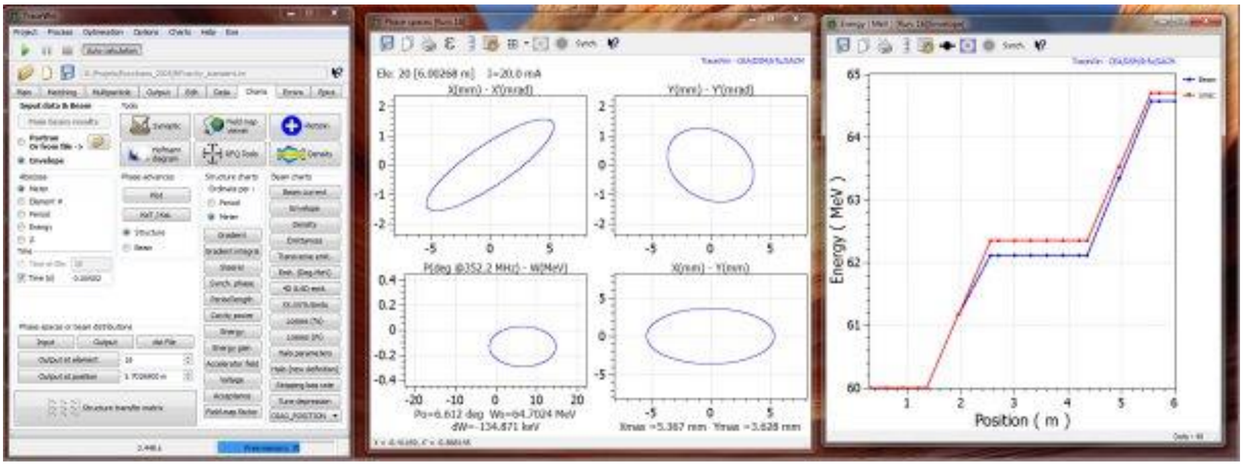
The following example shows the beam energy (blue) behavior during the 10 ms of simulation. The first slop at the beginning corresponds to the cavity filling. You can avoid this stage by using the “Fast_start” parameter of the cavity parameters file allowing to start simulation directly at the nominal field in the cavities.
You can see also, starting at 5 ms, the effect of the second cavity failure.



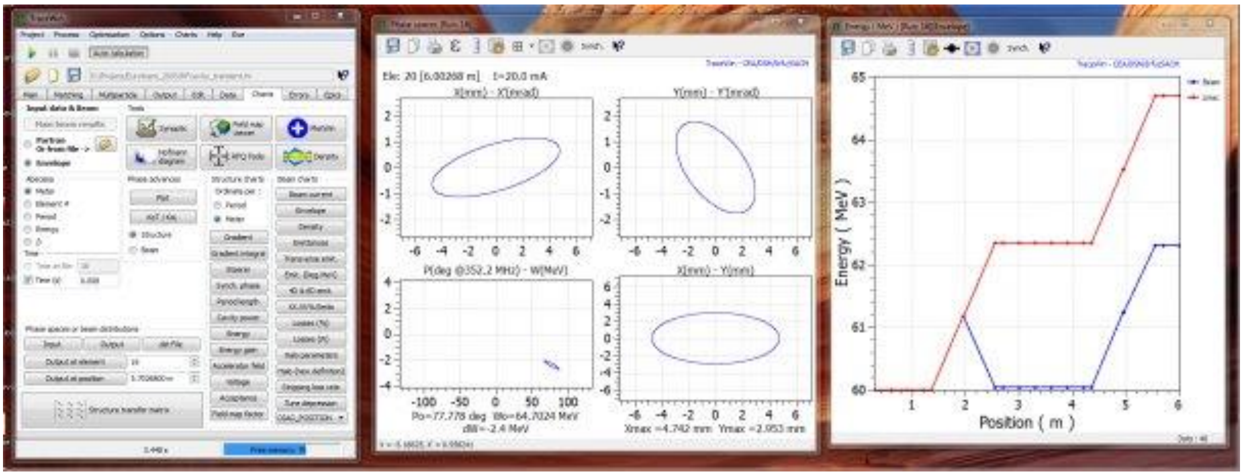
The other possibility is to show the beam behavior at a given time like following example, where at different time the beam envelope is shown. Bellow output beam en energy along the machine at t=3 ms.



Bellow output beam en energy along the machine at t=5.02 ms. The cavity start to failed and the first effects are visible on the output beam.


Ele: 20 [6.00268 m] I=20.0 mA
X(mm) - X'(mrad)
Y(mm) - Y'(mrad)
P(deg @352.2 MHz) - W(MeV)
X(mm) - Y(mm)
Po=6.612 deg Wo=64.7024 MeV
dW=-134.871 keV
Xmax =5.367 mm Ymax =3.628 mm
Energy (MeV)
Position (m)

Finally, bellow at 8 ms, almost at the end of the rtransient simulation


Ele: 20 [6.00268 m] I=20.0 mA
X(mm) - X'(mrad)
Y(mm) - Y'(mrad)
P(deg @352.2 MHz) - W(MeV)
X(mm) - Y(mm)
Po=77.778 deg Wo=64.7024 MeV
dW=-2.4 MeV
Xmax =4.742 mm Ymax =2.953 mm
Energy (MeV)
Position (m)

Errors study management

Remote & Local computers

For an error study (only statistical study) with PARTRAN or Toutatis or even in envelope, which spend a very long time, several computers can be used via client/server architecture (multiparameters scheme). These remote machines have to work under Window, Linux and MacOS operating system. You can decide to use these computers only during the night or weekend. Some computer can be add or remove during the error study process. (You have to install and launch into each computer the code "***twserver***".
For each computer you have to select the number of core you want to use.
If your run is very short (a few seconds), it's probably much better to use only your local computer (IP=0.0.0.0) and set a number of core bigger than the real available number of core. You should have to check that point in your own computer.

TraceWin server installation

For Windows you have to install "twserver.exe" and for Linux or MacOS install "twserver" code on your remote computers. You have just to copy the executable file somewhere and launch it. Setup is completely automatic and the final installation will be "*C:\TWSever*" for Windows and "*Home/.TWServer*" for Linux or MacOS. Print "*./twserver&*" for Linux or "*twserver.exe*" for Windows in a console. All new installation will destroy the running old version and will install the new one in the good directory. Don't try to start "twserver" directly in its final directory.
Sometime, some remote machines don't reply to TraceWin requests. A few reasons can explain that:
Check you firewall: The TraceWin server need access rights to TCP port between 1024 and 10000. Have a look on "*Remote computer*" box management in order to check if you select or not the option "*Work only out of working hours*". Check also the minimum "*Needed memory*" requirement.
Under Windows, remote server can be disabled locally (right click on right bottom tray icon). That means TraceWin will use this computer only if its CPU charge is very low.
Before any error study, test all your computers with "*Test computers*" button from "*Remote computers*" Box.

The twserver cannot launch more the *N* jobs in the same time. *N* is either defined by default by the number of core available or by the environment variable '*TWSERVER_N_THREAD=N*".

For CEA Saclay users only:
You can install "twserver" on the Irfu clusters called "DAPHPC" or "DAPINT", for that launch twserver by the command *"./twserver daphpc*" or *"./twserver dapint*" according to cluster where you are. Set the number of cores for the corresponding cluster

For ESS cluster users only:

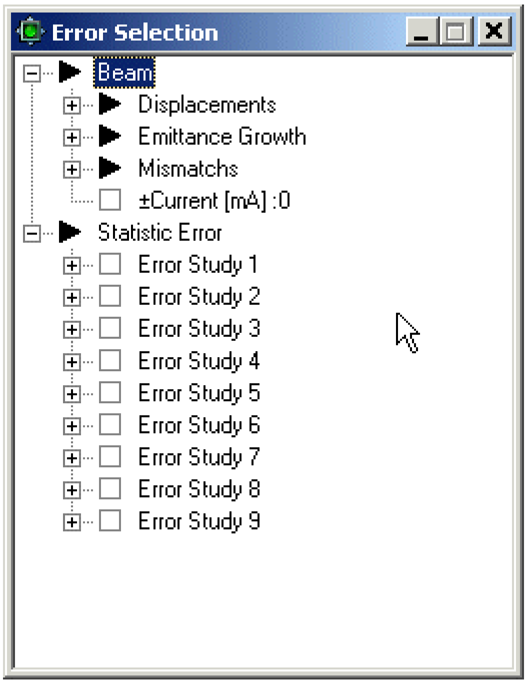
- Download "*twserver_without_X*" file and copy it to ESS cluster (user@ess01.dcsc.ku.dk).
- Dowload "*twservertunnel.sh*" file on your computer.
- On cluster:
    1. Rename "*twserver_without_X*" to "*twserver*"
    2. Type "*./twserver ess". This command will install twserver on cluster directory "/data/users/TWServer*". All users must have write and read rights. On the cluster only one occurance of twserve has to run. Each time a new user will type this command the old server will closed and the new one will replace it. (Avoid to do that when twserver is used, you'll stop a colleague study)
- On your local computer:

1. Type "*./twservertunnel open user@ess01.dcsc.ku.dk*" This command will create ssh tunnel between your local computer and ESS cluster. It's necessary because TraceWin communicates to twserver by FTP protocol using port 9090 to 9110 and SFTP is requiered. By this way the port 9090 to 9110 from your local computer will be directly connected to ESS cluster by a ssh tunnel. Unfortunallty for each port (20), you have to type your cluster password. (That can be avoid by creating a ssh key authorization, see your administrator)
2. Launch TraceWin and in remote computer box, add ESS cluster with IP address (127.0.0.1)
3. When you want stop your studies don't forget to closed port tunnels ("*./twservertunnel close user@ess01.dcsc.ku.dk").*

- An alternative simpler way is to directly use TraceWin code to ESS cluster (use "*TraceWin_libc_2-3-4*" version). In this case, after twserver installation you have to add ESS cluster IP to TraceWin remote computer list and you have also to transfer your project data to ESS cluster.

Generality

TraceWin allows to study many kinds of errors. They are separated in two types: The input "*beam errors*" and the "*statistical errors*". The first type concerns the input beam errors, like beam displacement, emittance growths, beam mismatches and so on. The second is based on a Monte Calo approach, it concerns quadrupole, cavity, RFQ and input beam, this method needs a statistical study with several runs.



The input beam errors are study one after the other. You have to select one or several errors and put the error amplitude values. There are 9 different statistical error studies, which are also study one after the other, but the errors defined inside are combined during a study including *N* run, ("***Number of run"***) and the error amplitude have to be insert in the data file by including the error commands.

Different tools are available in order to analyze the statistical results. All the results are located in

Input beam errors

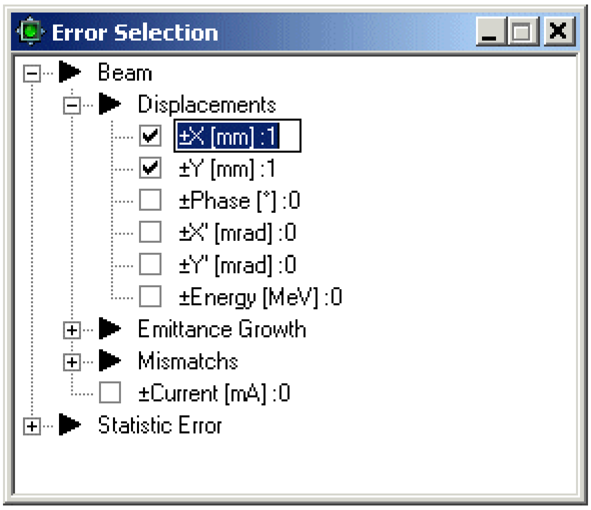
This kind of study, you have to use only one computer with one core.

Four kind of error:

- **Beam displacement**: The beam input position is not centered.
- **Emittance growth**: The input beam emittance is increased by a percentage.
- **Beam mismatch**: The input beam is mismatched by a percentage. A 20 % mismatch in x plane means α_x and β_x are multiplied by $(1.2)^2$.
- **Beam Current error**: Allows to study the effect of the input beam current variation.

The best way to understand the way of using is to look at an example:

Example: Know the linac behavior and losses when the input beam positions *X* and *Y* move from -1 mm to 1 mm in ten steps [-1.0, -0.9, -0.8,….,0.8, 0.9, 1.0]. You have to select these two kind of errors and put the error amplitude (Click two time one the item). Then, put "***Nbr of step***" to and to finally select "***Envelope error study***" or "***Particles error study***"



Error Selection

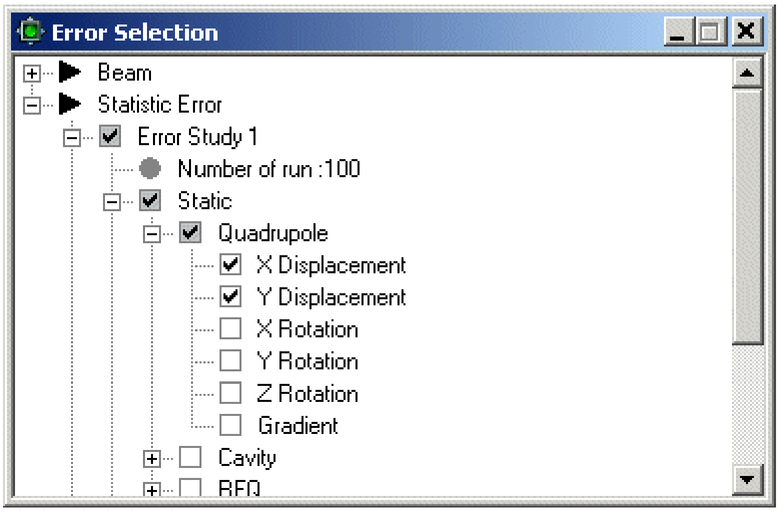
- Beam
  - Displacements
    - ☑ ±X [mm] :1
    - ☑ ±Y [mm] :1
    - □ ±Phase [°] :0
    - □ ±X' [mrad] :0
    - □ ±Y' [mrad] :0
    - □ ±Energy [MeV] :0
  - Emittance Growth
  - Mismatchs
  - □ ±Current [mA] :0
- Statistic Error

For the beam errors, in the box Remote and Local computer you have to select only your local computer (IP:0.0.0.0).

Now, you just have to run TraceWin, the two errors study will perform one after the other. At the end, you can look at the results like the output beam behavior according to the input beam errors by using the two file "*X_Displacement_ENV.txt*" and "*Y_Displacement_ENV.txt*" if "***Envelope study***" is selected or "*X_Displacement_PAR.txt*" and "*Y_Displacement_PAR.txt*" if "***Particles study***" is selected. You find them in the "***Calculation directory***" defined in the "Main" TraceWin page. One of these files has to be select in "***Studies results***" of the "Errors" page.

Statistical errors

The first stage is to include in the data file the different error commands with all the amplitude errors, the second one is to select the different kinds of error in the box "*Error selection*" of "*Error*" page, For example, if you want to include in the "*Error study 1*", some static quadrupole displacement and rotation errors you have to do the following selection.



- ⊞ ▶ Beam
- ⊟ ▶ Statistic Error
  - ⊟ ☑ Error Study 1
    - ● Number of run :100
    - ⊟ ☑ Static
      - ⊟ ☑ Quadrupole
        - ☑ X Displacement
        - ☑ Y Displacement
        - □ X Rotation
        - □ Y Rotation
        - □ Z Rotation
        - □ Gradient
      - ⊞ □ Cavity
      - ⊞ □ RFQ

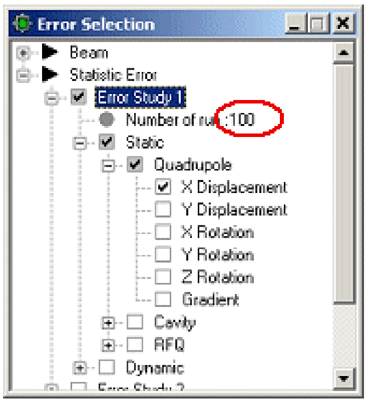
Third stage: Selects le number of linacs, which will be generated, here 100, "***Number of run***". Finally, select the number of step to reach the maximum amplitude of the errors, "***Nbr of step***" of "Error" page. For example 5 mean [20% of the max amplitudes, 40%, 60%…]. In this case 500 runs will be performed. All errors are randomly set in a uniform probability law *[-σ, □σ]*.

Correction scheme

If you have defined a correction scheme in you data file, this scheme will be applied to each machine like the following procedure:
Use your reference machine tuned with your correction scheme if you defined one (diagnostic associated with ADJUST command) and if you check "*Match using diagnostic*" in "*Matching*" page.
Static random errors are applied to your linac according to your static error commands.
Your correction scheme is performed.
Dynamic random errors are applied to your linac according to your dynamic error commands.
The final run is performed.

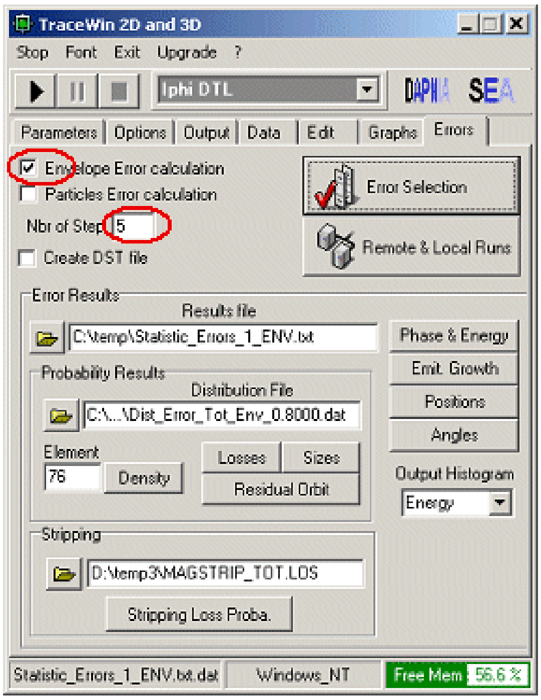
Error study example

A classical error study is concerning the quadrupole misalignments in several DTL tanks, for example: Quadrupoles misalignment study from 0 mm to 0.2 mm in 5 steps. 100 linacs with 1000 macro-particles for each of them.



Error Selection

- Beam
- Statistic Error
  - ☑ Error Study 1
    - Number of ru :100
    - ☑ Static
      - ☑ Quadrupole
        - ☑ X Displacement
        - □ Y Displacement
        - □ X Rotation
        - □ Y Rotation
        - □ Z Rotation
        - □ Gradient
      - □ Cavity
      - □ RFQ
    - □ Dynamic



TraceWin 2D and 3D

☑ Envelope Error calculation

□ Particles Error calculation

Nbr of Step 5

□ Create DST file

Error Results

Results file: C:\temp\Statistic_Errors_1_ENV.txt

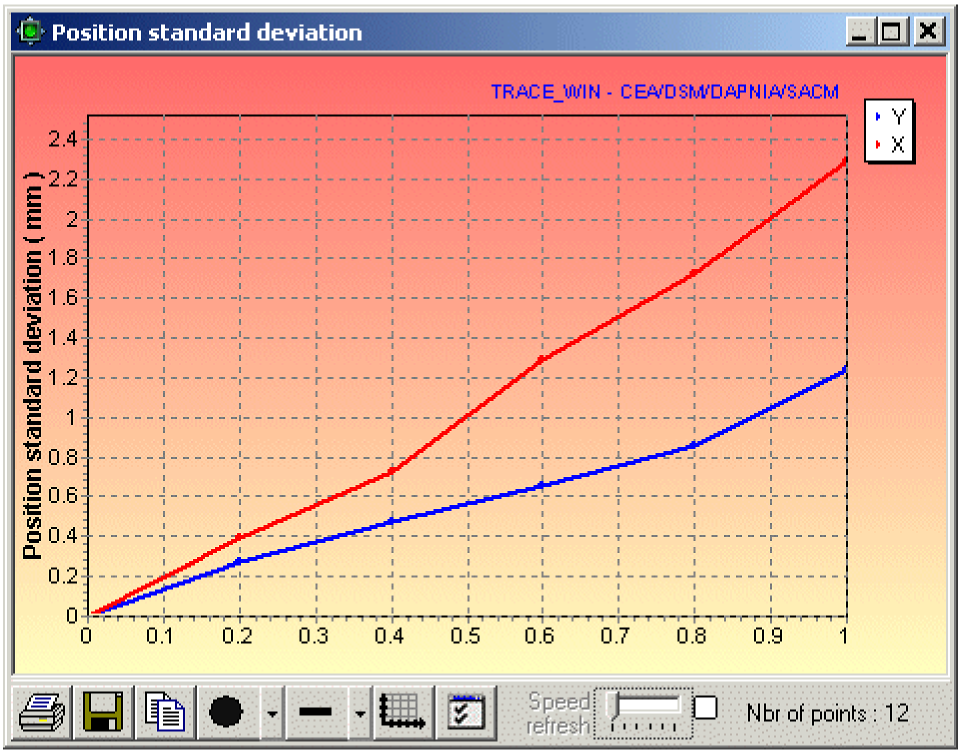
Probability Results

Distribution File: C:\...\Dist_Error_Tot_Env_0.8000.dat

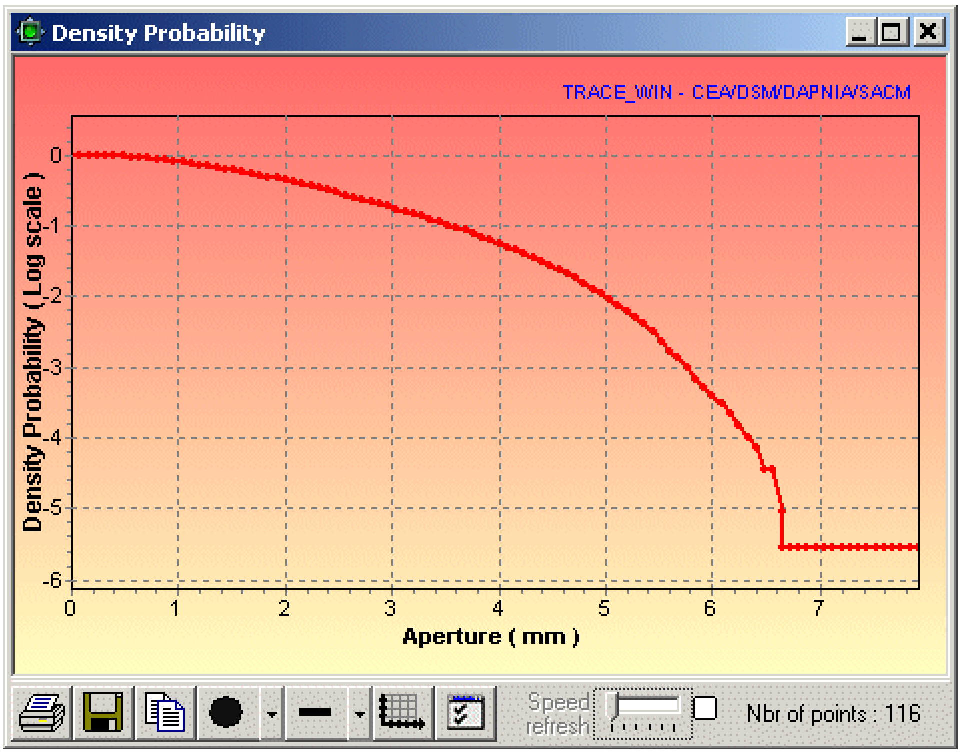
Element: 76

Stripping: D:\temp3\MAGSTRIP_TOT.LOS

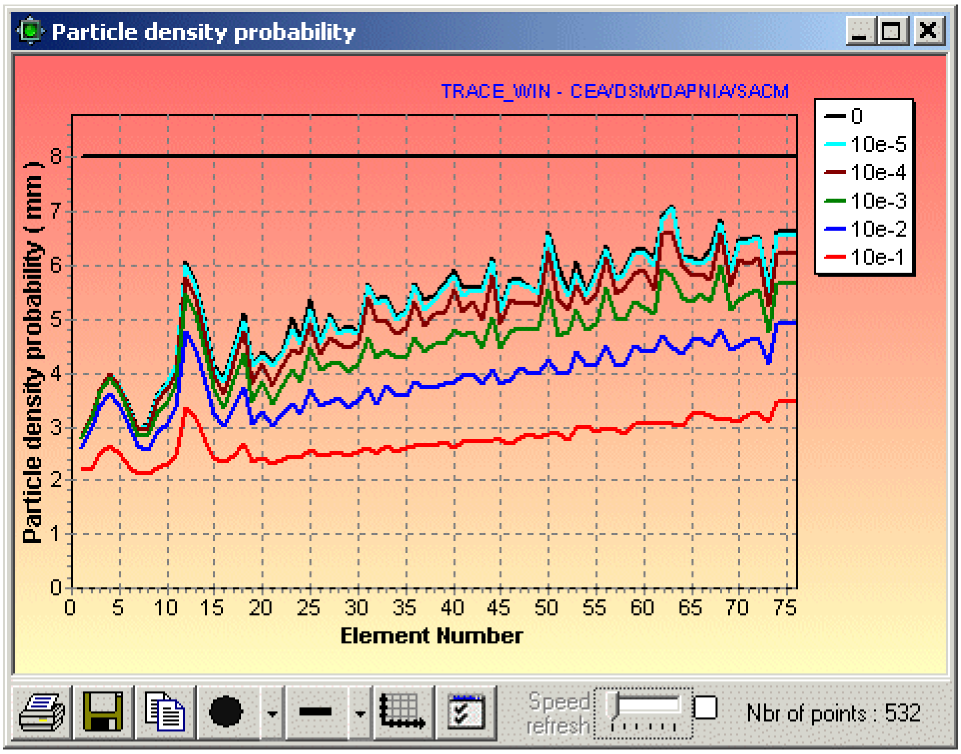
The result file is: "Statistic_Errors_1_ENV.txt"
The distribution file results are:"*Dist_Error_Tot_Env_0.2000.dat*" for 20%
"Dist_Error_Tot_Env_0.4000.dat" for 40%,
…
The rms output beam position for 100% of the error (0.2 mm) is about 2.2 mm for X.



The density probability of the beam located at the end of the element 76 shows a 6.6 mm maximum beam size.



The particle density probability repartition all along the structure show for example than the maximum size of 99% of the beam is lower than 5 mm when the errors are randomly distributed at 80% of the maximum amplitude. (Red curve: 90% of the 100.000 particles, 100 linacs of 1000 particles, Blue curve: 99%, Green curve 99.9% …)



If you want to combine other quadrupole errors, you just have to select them and restart the study. Obviously, cavity errors can be combined with quadrupole error. You could also insert in your data file a scheme of correction; steerers associated with beam monitor positions. Start a new study in order to see if your scheme is efficient.

Virtual accelerator

Implementation

Two different approaches to implement the virtual accelerator are possible, either we integrate the simulation model into the control system (like XAL @SNS or J-PARC example) or we integrate a part of the control system into the simulation code (like Traxce3D). This is the last approach which has been developed making the simulation code associated to control system as an independent high level application. The control system was based to EPICS (Experimental Physics and Industrial Control System), as many accelerator in the world.

EPICS

With EPICS is provided most of the gateways or programming interface to advanced programming languages. The database is accessed through the "Channel Access" protocol. Finally, the graphical user interfaces are various and many. Modern and useful, EPICS seems for the time being the ideal tool to perform this kind of development. Figure 5 represents the architecture of EPICS.

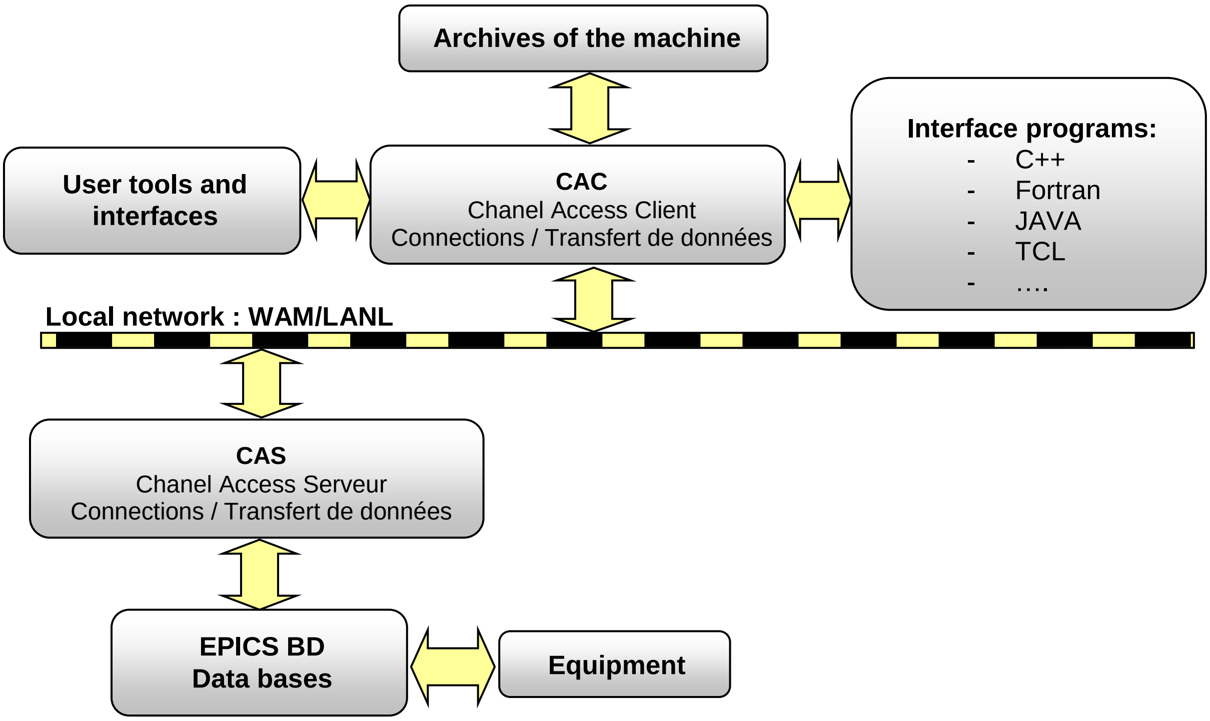
EPICS is based on a data base "EPICS BD" composed of recordings or "RECORD" objects representing equipment. Each of these RECORDs contains many fields representing the main parameters of the equipment. All these information are accessible for reading and writing via the Channel Access.
To simplify, we only considered for given equipment only 3 fields of the RECORD:

- **The command**: represents the value that we want to order to the equipment. For example, you wish to send 200A in the coil of a solenoid.

- **The virtual command**: represents the value that we want to order to the code for the simulation of this equipment. For example, you wish to simulate the machine with 200A in the solenoid coil.

- **The acquisition**: represents the acquisition of the value of this equipment. For example, we measure 200.3 A in the power supply connected to the solenoid.

With the structure of the RECORDs as it is defined in EPCIS, we can use therefore a single database to represent two accelerators. It should be noted that some components such as diagnostics will be used through the last field.
The different fields of the RECORDs are accessed via the 'Real Process Variable' or PV's variable.



Interface TraceWin ↔ EPICS

The dialogues between TraceWin and EPICS are made through "*EPICS Base*" library. It is the heart of EPICS, including code sources of the base system or kernel, libraries for interfacing with different operating systems, Channel Access libraries, the standard RECORD and many other tools. Among the different EPICS version, we used the **3.16.0.1**. This base allows to build a Chanel Access Client (ACC) for reading and writing in the database of EPICS. The process should use a database, to be defined, internal to the simulation code making the correspondences between the syntax and the physical quantities in TraceWin and EPICS such it is described in figure below.

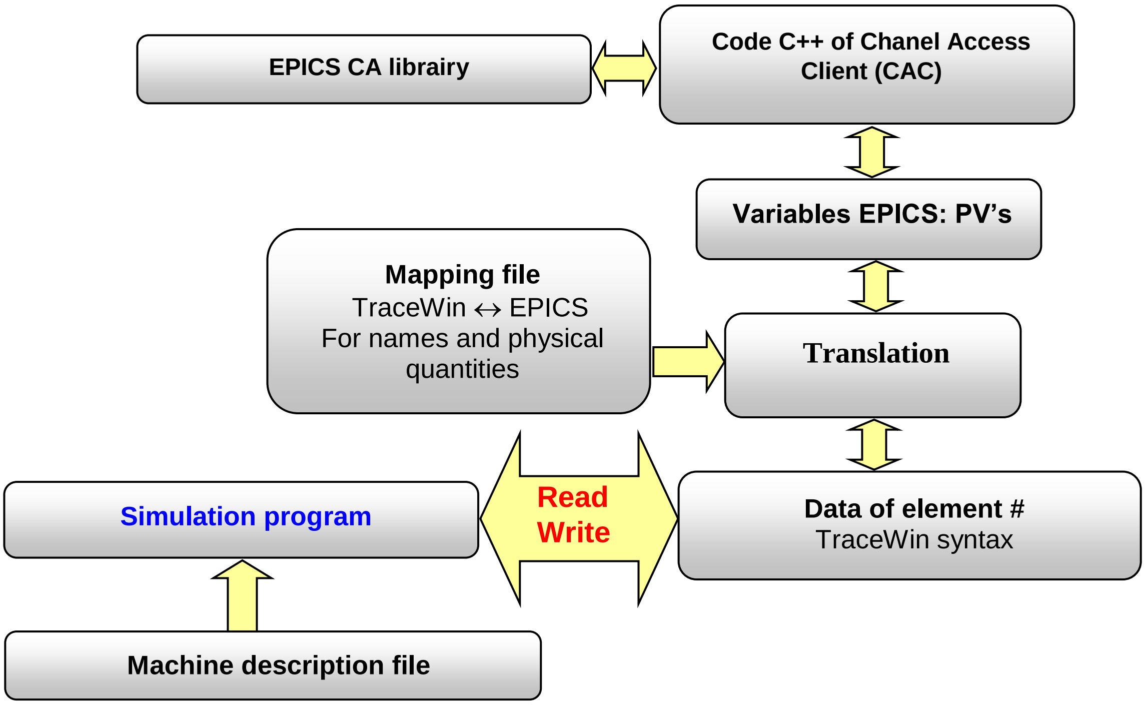
TraceWinVA

2 versions, named TraceWinVA, have been developed for Windows (64bits) and Linux (64bits) (*). All developments have been done in C++ language and the final architecture is visible below. Both codes are fully stand-alone and don't require external library installation. It is thus very easy to install them, usually the linux version, on a sufficiently powerful computer dedicated to the calculation in accelerator control rooms.

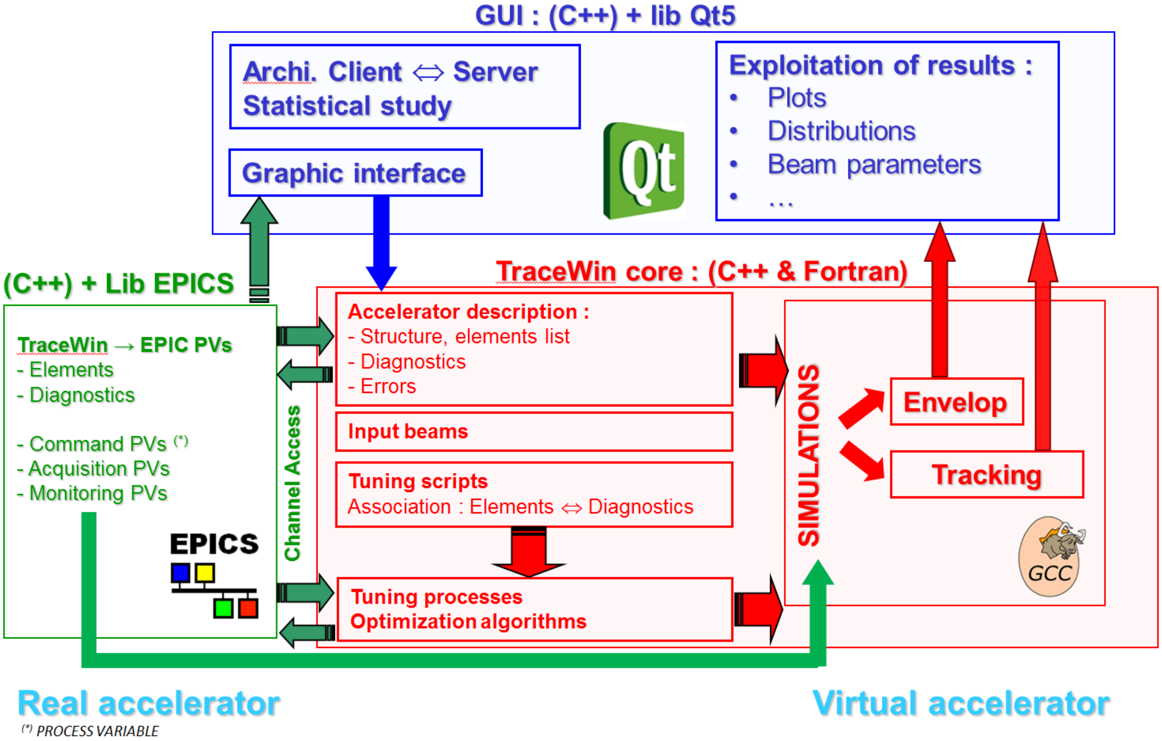
(*) *If your system requires a 32-bit version, it could be provided only on request.*

Even if “*TraceWinVA*” and “*TraceWin*” codes seem very similar, it’s not exactly true and we still recommande to use “*TraceWin*” code for pure simulations because a little bit more efficient.

TraceWinVA is a beta version in even if it contains most of the major features; it is not yet complete and tested.



Structure of the code of the virtual accelerator



Implementation in TraceWinVA code.

Data synchronization

One of the difficulties is the synchronization of the EPCIS commands with the code. In other words, how do you know that a command has gone to EPICS and then check that the concerned equipment has been put to value requested? The example of power supply for magnetic element equipment illustrates this difficulty. Already developed and tested on IPHI and SPIRAL2 projects, both required different approaches to control usual power supply. We considere today that the IPHI method is the most usual way and it's the default method use in TraceWinVA code, but probbalby new method more efficient could be implemented on another project according to machine specificities.

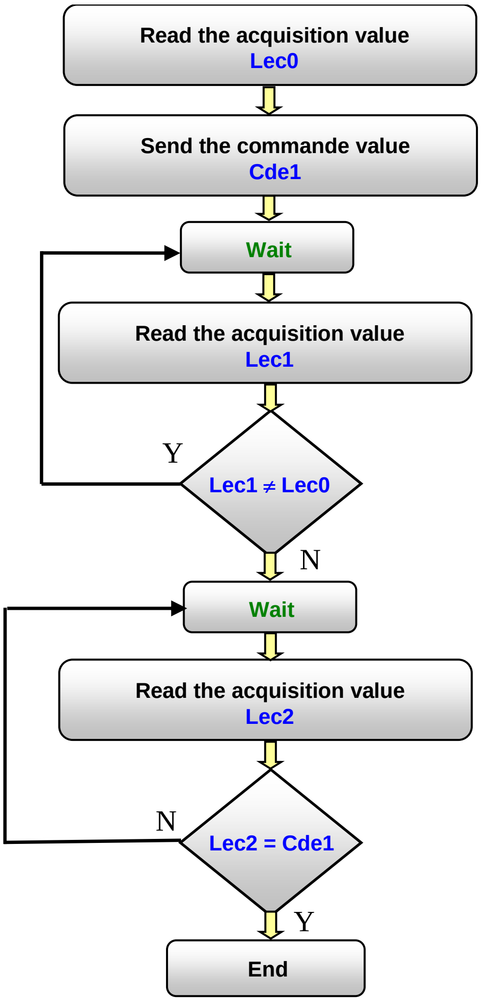
IPHI (Default)

Figue below presents the algorithm for synchronization of a command dedicated to a power supply of the IPHI injector. It is based on two waiting loops. The first is intended to await the arrival of the command on the equipment while the second waits for stabilization of the acquisition value returned by the power supply to EPCIS data base. That means that the sending of a data speed is therefore linked to the speed of response of the equipment.

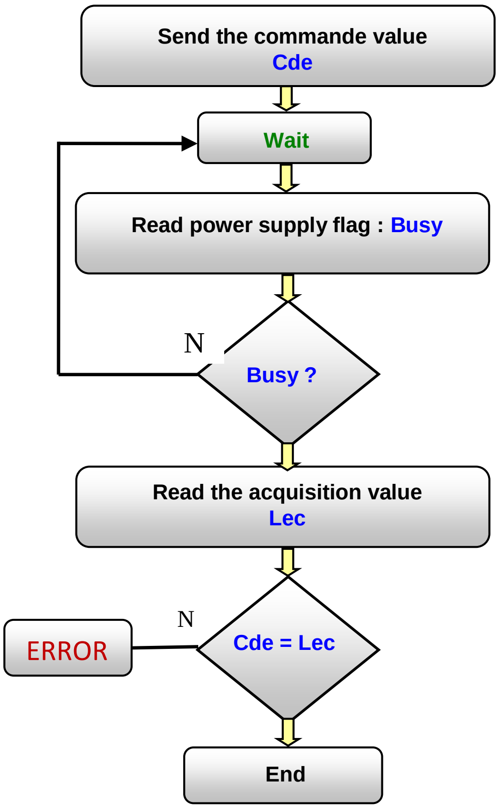
SPIRAL2

Unlike IPHI, power supplies of SPIRAL2 are much more recent and have a flag indicating that they are reaching the final setting. According to this specification a different and more efficient and simple scheme has been developed for SPIRAL2 project.

This example shows that it is not always possible to standardize the code for all equipment and some necessary adaptations for the specific features of the elements of each machine have to be considered and implemented.


Read the acquisition value
Lec0
Send the commande value
Cde1
Wait
Read the acquisition value
Lec1
Y
Lec1 ≠ Lec0
N
Wait
Read the acquisition value
Lec2
N
Lec2 = Cde1
Y
End

Writing algorithm for IPHI power supplies



Writing algorithm for SPIRAL2 power supplies

Obviously, in the VA software no waiting loop are used and all PV are monitoring in asynchrony way.

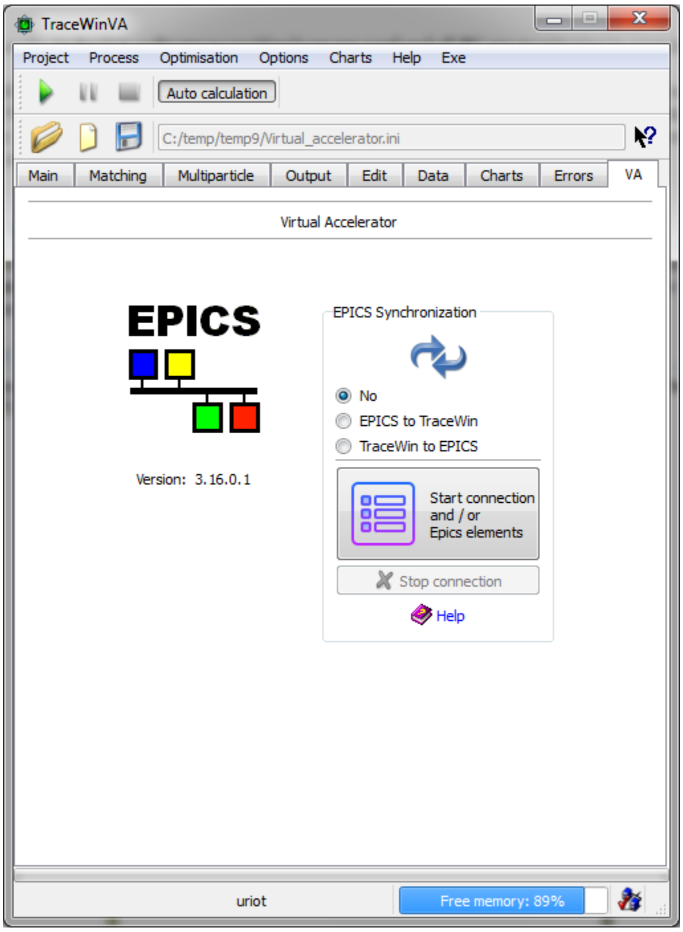
In addition, it is important to match the physical units used in TraceWin and EPICS. For example, TraceWin knows the magnetic field gradient G(T/m) of a quadrupole while EPICS works with the power supply current I(A). This necessarily implies to provide to TraceWinVA the corresponding G(I), see example.

Way of using

The next is based on the project example named "*Virtual_accelerator*" from the “*Help*” menu then “*Open examples*”.

Main interface

In TraceWinVA’s tabsheet “*Epics*”, 3 options are available to control synchronization between the EPICS data base and the simulation code:



EPICS Synchronization
● No
○ EPICS to TraceWin
○ TraceWin to EPICS

- **No** : means the code doesn’t provide connection with the EPICS data base.

- **EPICS to TraceWin:** In this mode, any action on the control system of the machine affects at the same time on the real accelerator and on the virtual one. The virtual Accelerator is directly related to the adjustment of the real machine parameters and simulates it. This simulation can be automatically started at each change of a setting or be done periodically. The machine is under the permanent supervision of the virtual machine.

The diagnostic values resulting from the simulation can be sent to the real machine, allowing the virtual machine to be used to test high-level control command procedures.

- **TraceWin to EPICS:** The first interest of this mode is its capacity to send the machine setting calculated by the simulation code into control system of the real machine. In this mode, algorithms for automatic adjustment of the simulation code can be used to adjust the real machine. Element setting and daignostics measurement come from real machine. The different procedures of tuning has be defined using DIAG_XXX element and ADJUST commands

You can find in "Help" menu then "Open example", a virtal machine project example.

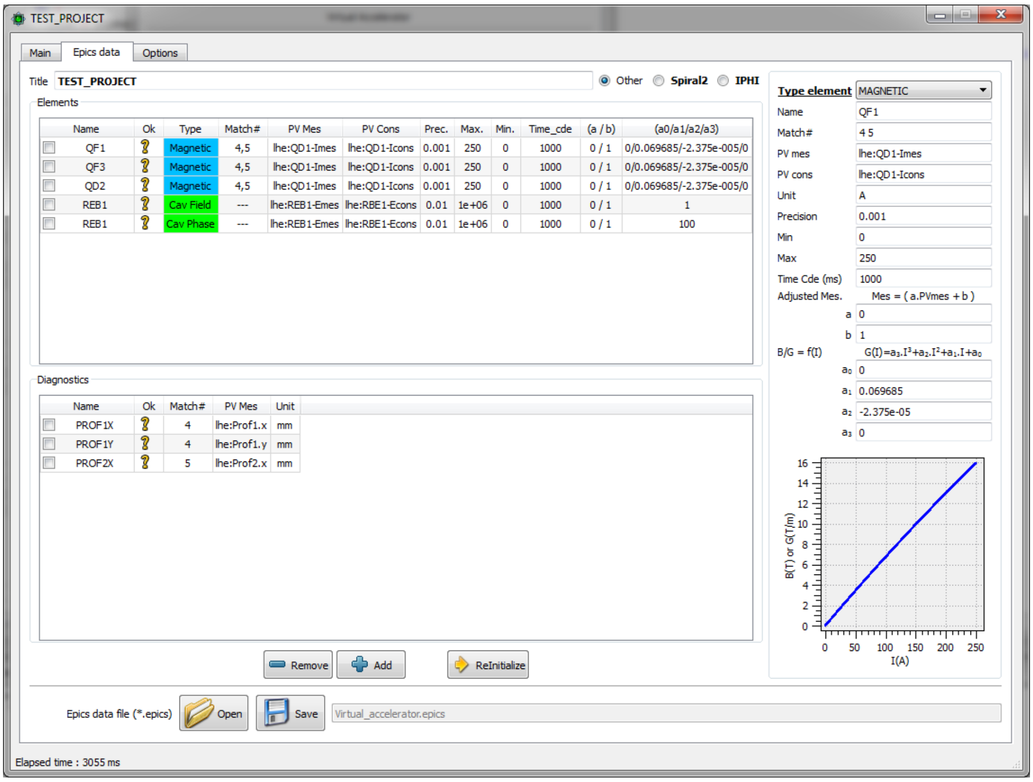
When a connected mode is selected a button "*Stop connection*" allows to break this link to come back to an usual standalone simulation code.

Before to select a connection mode, the liste of elements has to be defined using button "*Start connection and/or Epics Elements*". It provides the tools allowing to visualize and initialize the common elements defined in TraceWinVA code and in EPICS data base. Fist stage consists to go the the tab-sheet "*Epics data*", see below, in order to defined the liste of EPICS elements and their main parameters.

Epics tool – "Epics data"

The page contains all data needed to read and set the elements defined in the EPICS data base, including diagnostics, typically PV names and some other parameters.

- The name of element has to be similar to the name defined in the structure file page.
- G(I) has to be defined either in the GUI and in the structure file using the EXCITATION_CURVE command for magnetic elements.
- Today, recognized elements are only FIELD_MAP, but future extension to usual elements is possible.
- "*Precision*" parameter has to be chosen very carefully, because it is used to end an EPICS command before the time_out defined by "*Time Cde*" parameter.
- When you add, delete or change something a reinitialisation is mandatory.
- All data has to be saved in a "**.epics*" file.



Title TEST_PROJECT ● Other ○ **Spiral2** ○ **IPHI**

Elements

	Name	Ok	Type	Match#	PV Mes	PV Cons	Prec.	Max.	Min.	Time_cde	(a / b)	(a0/a1/a2/a3)
□	QF1	?	Magnetic	4,5	lhe:QD1-Imes	lhe:QD1-Icons	0.001	250	0	1000	0 / 1	0/0.069685/-2.375e-005/0
□	QF3	?	Magnetic	4,5	lhe:QD1-Imes	lhe:QD1-Icons	0.001	250	0	1000	0 / 1	0/0.069685/-2.375e-005/0
□	QD2	?	Magnetic	4,5	lhe:QD1-Imes	lhe:QD1-Icons	0.001	250	0	1000	0 / 1	0/0.069685/-2.375e-005/0
□	REB1	?	Cav Field	---	lhe:REB1-Emes	lhe:RBE1-Econs	0.01	1e+06	0	1000	0 / 1	1
□	REB1	?	Cav Phase	---	lhe:REB1-Emes	lhe:RBE1-Econs	0.01	1e+06	0	1000	0 / 1	100

Diagnostics

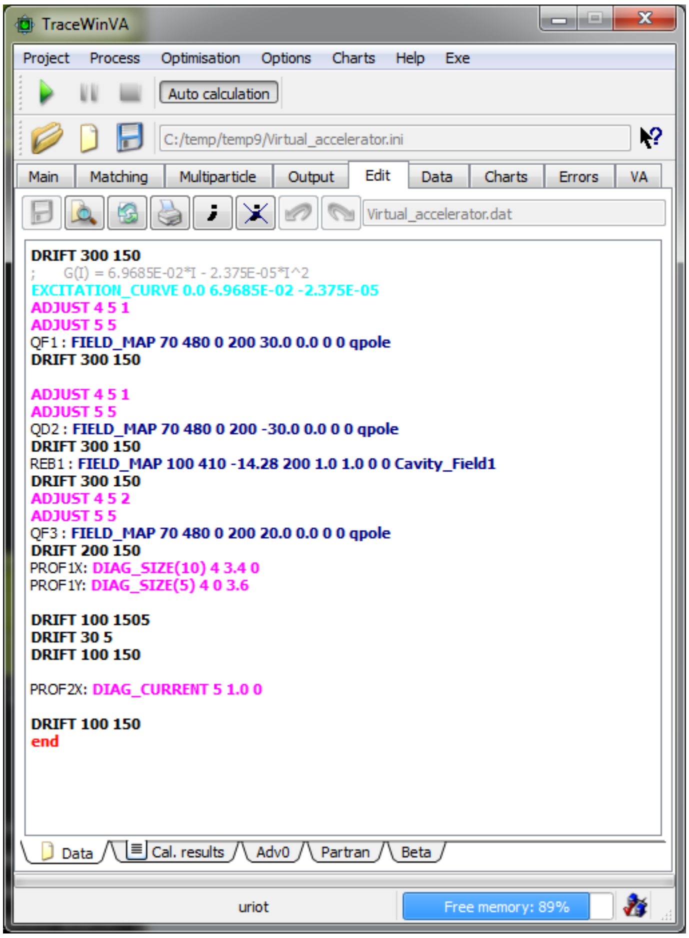
	Name	Ok	Match#	PV Mes	Unit
□	PROF1X	?	4	lhe:Prof1.x	mm
□	PROF1Y	?	4	lhe:Prof1.y	mm
□	PROF2X	?	5	lhe:Prof2.x	mm

Type element	MAGNETIC
Name	QF1
Match#	4 5
PV mes	lhe:QD1-Imes
PV cons	lhe:QD1-Icons
Unit	A
Precision	0.001
Min	0
Max	250
Time Cde (ms)	1000
Adjusted Mes.	Mes = (a.PVmes + b)
a	0
b	1
B/G = f(I)	$G(I)=a_3.I^3+a_2.I^2+a_1.I+a_0$
a_0	0
a_1	0.069685
a_2	-2.375e-05
a_3	0

Epics data file (*.epics) Virtual_accelerator.epics

Elapsed time : 3055 ms

List of elements and diagnostics used from EPICS data base



C:/temp/temp9/Virtual_accelerator.ini

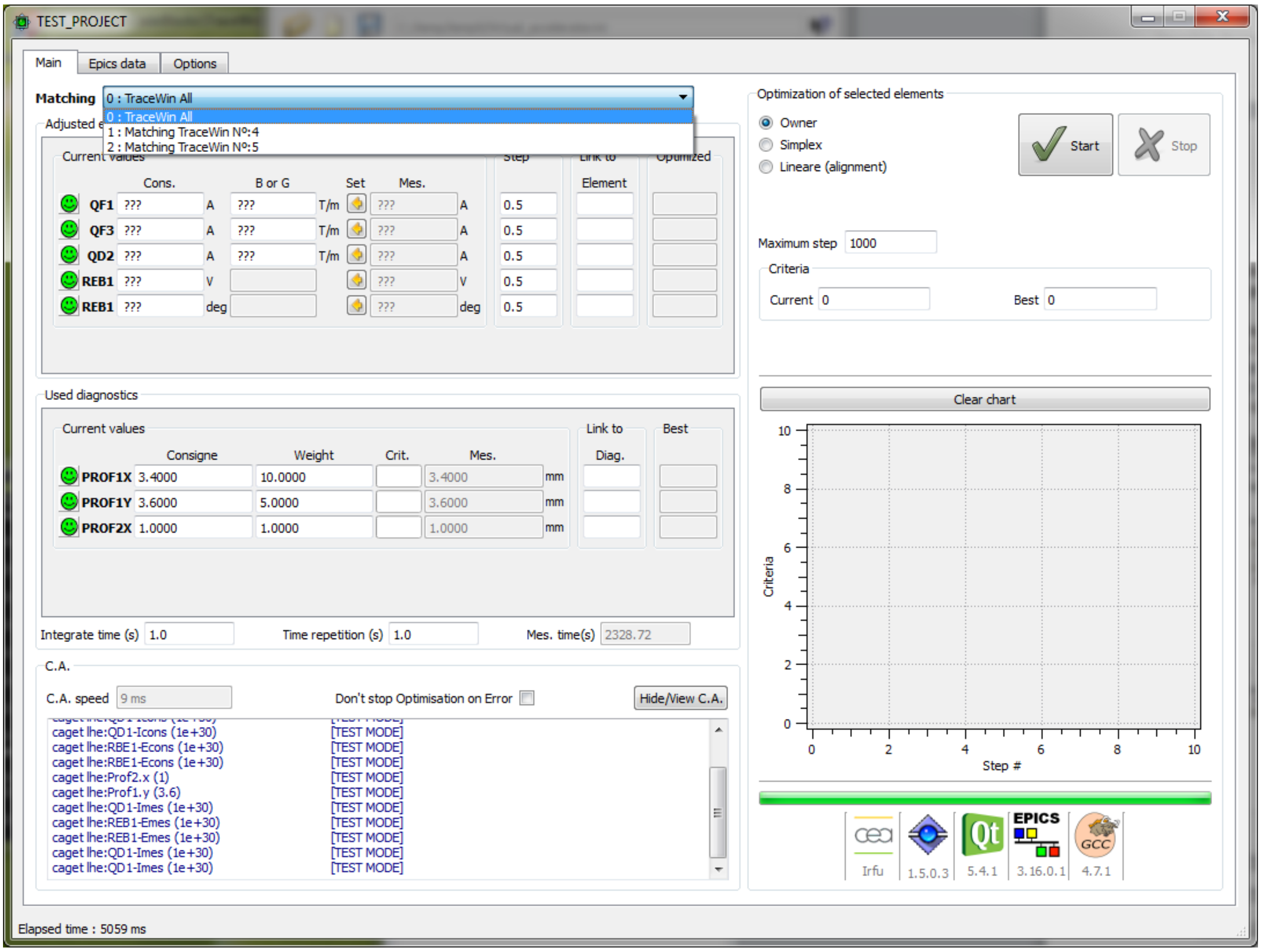
Virtual_accelerator.dat

```
DRIFT 300 150
;      G(I) = 6.9685E-02*I - 2.375E-05*I^2
EXCITATION_CURVE 0.0 6.9685E-02 -2.375E-05
ADJUST 4 5 1
ADJUST 5 5
QF1 : FIELD_MAP 70 480 0 200 30.0 0.0 0 0 qpole
DRIFT 300 150

ADJUST 4 5 1
ADJUST 5 5
QD2 : FIELD_MAP 70 480 0 200 -30.0 0.0 0 0 qpole
DRIFT 300 150
REB1 : FIELD_MAP 100 410 -14.28 200 1.0 1.0 0 0 Cavity_Field1
DRIFT 300 150
ADJUST 4 5 2
ADJUST 5 5
QF3 : FIELD_MAP 70 480 0 200 20.0 0.0 0 0 qpole
DRIFT 200 150
PROF1X: DIAG_SIZE(10) 4 3.4 0
PROF1Y: DIAG_SIZE(5) 4 0 3.6

DRIFT 100 1505
DRIFT 30 5
DRIFT 100 150

PROF2X: DIAG_CURRENT 5 1.0 0

DRIFT 100 150
end
```

Data file of TraceWin describing the structure of the simulated machine

Epics tool – “Main”

This main page, displays the synchronized PV values defined in the preceding page. All actions in the control system of the machine are automatically visible here. In addition, directly from this page it is possible to modify individually a PV value which causes a direct action on the machine. Only common elements and diagnostics of EPICS tools and structure file are visible on this page. The matching “*TraceWin_all*” shows all of them. The liste of “*Matching TraceWin N°#*” correspond to the different matchings defined in the structure file and an optimization can be launched directly on the real machine based on the measurements of the diagnostics.
Button “*Set*” allows to individually control an element on the real machine and also test the different EPCIS parameters of this element.



Matching: 0 : TraceWin All

Adjusted e… – Current values

	Cons.		B or G		Set	Mes.		Step	Link to Element	Optimized
QF1	???	A	???	T/m		???	A	0.5		
QF3	???	A	???	T/m		???	A	0.5		
QD2	???	A	???	T/m		???	A	0.5		
REB1	???	V				???	V	0.5		
REB1	???	deg				???	deg	0.5		

Used diagnostics – Current values

	Consigne	Weight	Crit.	Mes.		Link to Diag.	Best
PROF1X	3.4000	10.0000		3.4000	mm		
PROF1Y	3.6000	5.0000		3.6000	mm		
PROF2X	1.0000	1.0000		1.0000	mm		

Integrate time (s) 1.0 Time repetition (s) 1.0 Mes. time(s) 2328.72

C.A.

C.A. speed 9 ms Don't stop Optimisation on Error □

```
caget lhe:QD1-Icons (1e+30)           [TEST MODE]
caget lhe:RBE1-Econs (1e+30)          [TEST MODE]
caget lhe:RBE1-Econs (1e+30)          [TEST MODE]
caget lhe:Prof2.x (1)                 [TEST MODE]
caget lhe:Prof1.y (3.6)               [TEST MODE]
caget lhe:QD1-Imes (1e+30)            [TEST MODE]
caget lhe:REB1-Emes (1e+30)           [TEST MODE]
caget lhe:REB1-Emes (1e+30)           [TEST MODE]
caget lhe:QD1-Imes (1e+30)            [TEST MODE]
caget lhe:QD1-Imes (1e+30)            [TEST MODE]
```

Optimization of selected elements

● Owner
○ Simplex
○ Lineare (alignment)

Maximum step 1000

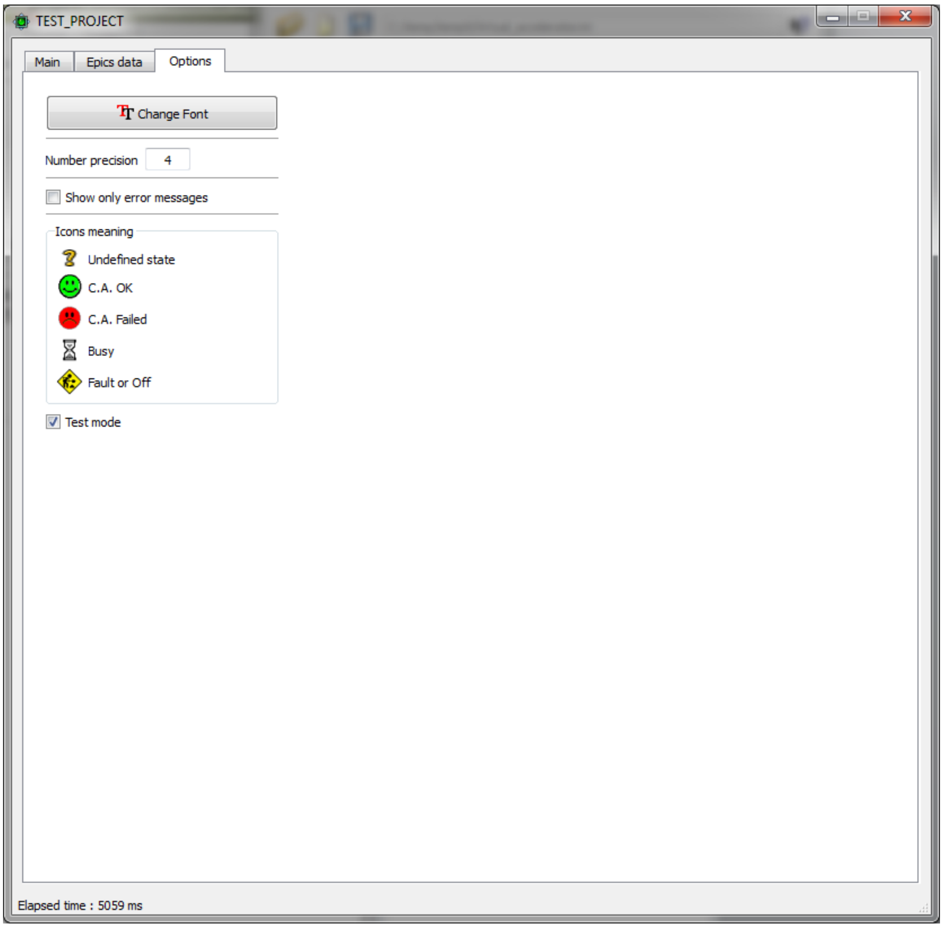
Criteria: Current 0 Best 0

Elapsed time : 5059 ms

Visualization and monitored elements.

Epics tool – “Options”

Some options are available here; the main one is the “*Test mode*”. Set by defaut in the example, it allows to simulate a EPICS connection. You have to remove it to really send command to the real machine.
“*Number precision*” set the number precision in the GUI dispaying.



Number precision 4

□ Show only error messages

Icons meaning

- Undefined state
- C.A. OK
- C.A. Failed
- Busy
- Fault or Off

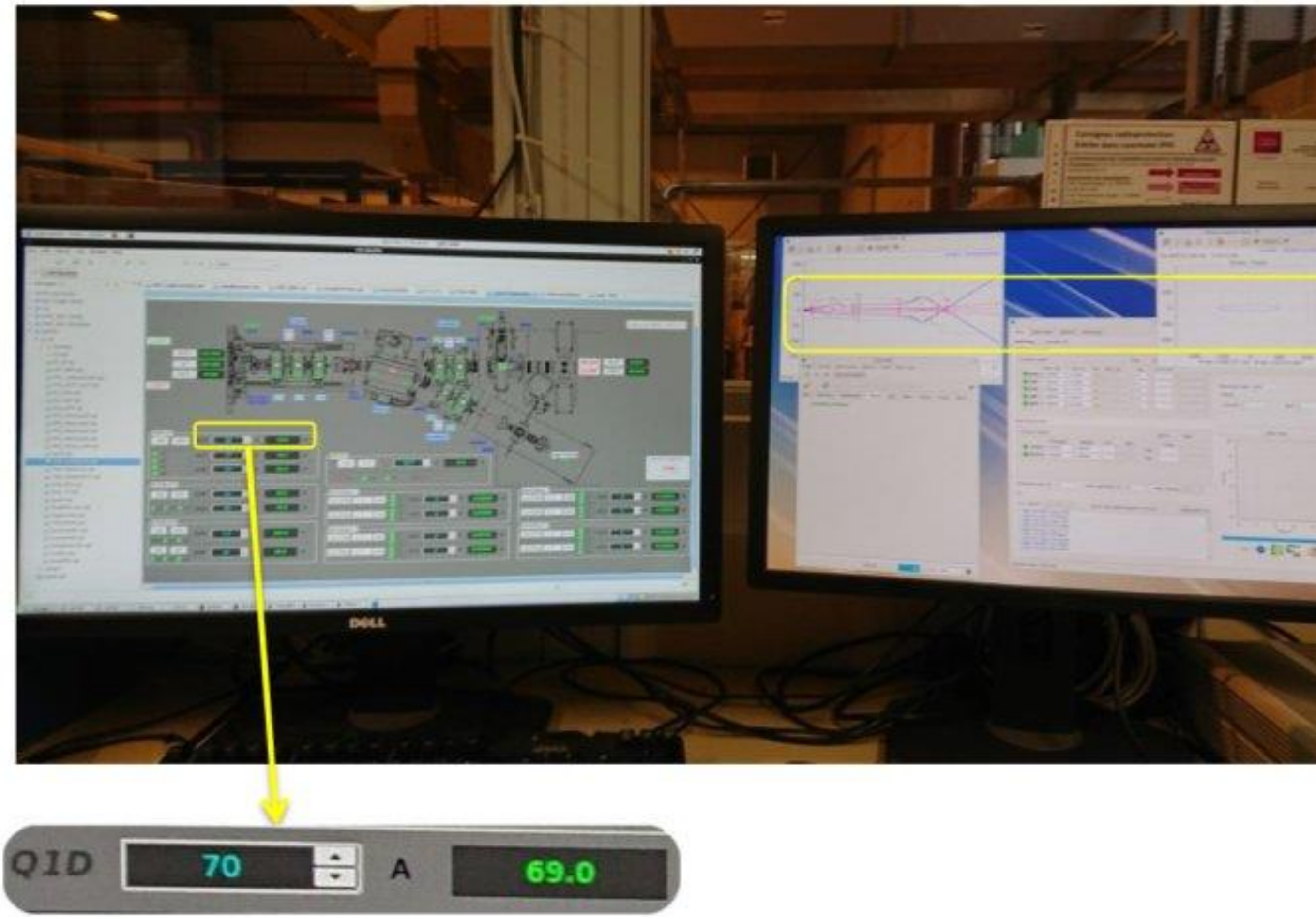
☑ Test mode

Elapsed time : 5059 ms

Synchronization of simulations

The virtual accelerator program has to be configured in "*EPICS to TraceWin*" mode. Then all simulation made use the data coming from EPICS data base. As all PVs are monitored and synchronized, when during operation of the real accelerator an element setting is changed in the control system, the simulation is automatically redone and the output charts are refreshed in the same time. In this way, an operator can continuously view the state of the beam and its main parameters wherever it is relevant. A comparison with measurements from diagnostics can thus allow a deep monitoring of the accelerator behavior. A significant discrepancy must alert the operator that something is wrong.

Both figures below show the control room of the IPHI injector before and after an operator has significantly changed a magnetic element of the line. We can immediately see an impact on the envelopes of the beam due to changes in the setting of the machine. Output chart has to set in "*Refresh*" mode to allow this feature.



Views of IPHI’s control room, QD1=70A, some outputs of corresponding VA simulation are shown on the right.

Views of IPHI’s control room. QD1 is set by operator to 10A, new VA simulation is launched and new output displays are synchronized.