

**Technical Note,
Updated for version 3.7
March 29, 2007**

**TRACK – a Code for Beam Dynamics Simulation in
Accelerators and Transport Lines with 3D Electric and
Magnetic Fields¹**

P.N. Ostroumov, V.N. Aseev, B. Mustapha
Argonne National Laboratory, Physics Division
E-mail: ostroumov@phy.anl.gov

Argonne National Laboratory, Argonne, IL, U.S.A.

¹ This work is supported by the U. S. Department of Energy, Office of Nuclear Physics, under contract No. DE-AC02-06CH11357.

SOFTWARE COPYRIGHT NOTIFICATION

© COPYRIGHT 2007 UChicago Argonne, LLC

THIS SOFTWARE DISCLOSES MATERIAL PROTECTED UNDER COPYRIGHT LAW AND FURTHER DISSEMINATION IS PROHIBITED WITHOUT PRIOR WRITTEN CONSENT OF THE PATENT COUNSEL OF ARGONNE NATIONAL LABORATORY.

ARGONNE NATIONAL LABORATORY, WITH A FACILITY IN THE STATE OF ILLINOIS, IS OWNED BY THE UNITED STATES GOVERNMENT, AND OPERATED BY UCHICAGO ARGONNE, LLC UNDER PROVISION OF A CONTRACT WITH THE DEPARTMENT OF ENERGY.

GOVERNMENT LICENSE AND DISCLAIMER

NOTICE: This data was produced by UChicago Argonne, LLC under Contract No. DE-AC02-06CH11357 with the Department of Energy. For 5 years from March, 2007, the Government is granted for itself and others acting on its behalf a nonexclusive, paid-up, irrevocable worldwide license in this data to reproduce, prepare derivative works, and perform publicly and display publicly, by or on behalf of the Government. There is provision for the possible extension of the term of this license. Subsequent to that period or any extension granted, the Government is granted for itself and others acting on its behalf a nonexclusive, paid-up, irrevocable worldwide license in this data to reproduce, prepare derivative works, distribute copies to the public, perform publicly and display publicly, and to permit others to do so. The specific term of the license can be identified by inquiry made to UChicago Argonne, LLC or DOE.

THIS SOFTWARE IS SUPPLIED "AS IS" WITHOUT WARRANTY OF ANY KIND.

NEITHER THE UNITED STATES GOVERNMENT, NOR THE UNITED STATES DEPARTMENT OF ENERGY, NOR UCHICAGO ARGONNE, LLC, NOR ANY OF THEIR EMPLOYEES, MAKES ANY WARRANTY, EXPRESS OR IMPLIED, OR ASSUMES ANY LEGAL LIABILITY OR RESPONSIBILITY FOR THE ACCURACY, COMPLETENESS, OR USEFULNESS OF ANY INFORMATION, DATA, APPARATUS, PRODUCT, OR PROCESS DISCLOSED, OR REPRESENTS THAT ITS USE WOULD NOT INFRINGE PRIVATELY OWNED RIGHTS.

LICENSING INQUIRIES MAY BE DIRECTED TO THE OFFICE OF TECHNOLOGY TRANSFER AT ARGONNE NATIONAL LABORATORY.

Table Of Content

1. Introduction	4
2. Code verification.....	5
3. Three-dimensional electromagnetic field distribution	5
4. Launching the code MWRead.....	6
4.1 Input Data.....	6
4.2 Output Data.....	9
5. TRACK Input Files.....	10
5.1 Track.dat	11
5.2 Sclinac.dat.....	15
5.3 Fi_in.dat	15
5.4 Solenoid.#01: Data file for 1D solenoid field profile	15
5.5 Stripper.#01: Data file for stripper parameters	16
5.6 RFQ.#01: Data file for RFQ parameters.....	17
5.7 Graph.cfg: Data file for graphics output.....	17
5.8 Maximum size of arrays in the code.....	18
6. Beam line elements supported by TRACK.....	19
6.1 RF devices.....	19
6.2 Magnetostatic devices.....	21
6.3 Electrostatic devices.....	27
6.4 Drift space and other elements of the beamline.....	29
6.5 Misalignments, errors and beam steering correction	33
7. Launching the code TRACK	38
8. TRACK Output data	38
9. Acknowledgements.....	42
10. References	42
Appendix 1. Simulation of element misalignments and field errors	43
A1.1 Main Cycle for generation of different accelerator seeds	47
Appendix 2. Optimization of the corrector files	48
Appendix 3. Acceptance calculation.....	52
Appendix 4. Potential expansion in a multi-cell RFQ	53
A4.1 Regular accelerating cells	54
A4.2 Transition cell	55
A4.3 Entrance and exit radial matchers	57
Appendix 5. Particle simulation in dipole magnets	60
A5.1 Structure line for the element “rounded DIPOLE”	64
Appendix 6. Creation of user defined initial distribution.	65
Appendix 7. A short manual for longitudinal corrections using TRACK	67

1. Introduction

The code TRACKv37¹ for Windows PC simulates beam dynamics of multi-component ion beams in linear accelerators and has the following features:

- multiparticle simulation of multiple component ion beams in 6D phase space;
- 3D electromagnetic fields in rf resonators are obtained with the CST MWS in rectangular mesh;
- Fringing fields of magnets and multipoles are approximated using Enge coefficients as in the RAYTRACE code;
- Realistic fields in solenoids;
- Integration of equations of motion by 4th order Runge-Kutta method;
- Misalignments and random errors;
- Two-dimensional (for dc beams) and three-dimensional space charge calculation. The test runs indicate that the space charge routine works well.
- Calculation of first and second order matrices of all elements.

Elements supported by TRACKv35:

- Any type of accelerating resonator with realistic 3D fields.
- Radio Frequency Quadrupole (RFQ) accelerators;
- Soft&hard edge solenoids;
- Bending magnets with the fringing fields;
- Electrostatic and magnetic multipoles (quadrupoles, sextupoles etc.) with fringing fields;
- Hard edge quadrupoles;
- Multi-harmonic bunchers;
- Axial-symmetric electrostatic lenses;
- Entrance and exit of the HV deck;
- Transverse beam steering "thin" elements;
- Stripping foils&films. (Currently the stripping elements are implemented for the RIA accelerators only.)
- Horizontal & vertical slits.
- Misalignment of bending magnets;
- Automatic multi-component beam center steering in transverse phase space;
- Automatic multi-component beam energy correction to compensate static errors of rf fields.

The following capabilities will be added soon:

- Built-in optimization of envelope matching in transverse and longitudinal phase space in realistic fields of solenoids and rf cavities.
- Combined SC magnet containing solenoid and dipole steering coils with realistic field distribution. This option existed in TRACKv21 but has to be properly incorporated into TRACKv35;
- Beam neutralization (This may be important in the LEBT);

The simulation includes two main steps:

¹ The version of TRACKv37 for multiprocessor platform operating on LINUX has been also developed.

1) Preparation of field maps. This step requires extensive use of the code Microwave studio (MWS) and Electromagnetic studio (EMS). The field maps can be also prepared using other codes.

2) Prior to the final multi-particle simulation, the elements of the transport system and accelerator must be defined by using first and higher order optimization codes. TRACK can also be used in some cases.

3) Tracking of large amount of multi-particles.

Important Note. This code is constantly being modified and verified. We realize that this manual has significant room for improvement. Please contact us to report any kind of problems encountered while running TRACKv37.

Currently TRACKv37 does not support the following:

- a) Space charge calculations in the transport system with “hard edge” solenoids, multipoles and bending magnets.
- b) Space charge of bunched beams in bending magnets.

2. Code verification

We have verified TRACKv35 calculations for common types of elements against other simulation codes such as LANA, DYNAMION and ELEGANT. The transport element calculations with fringing fields have been verified using the codes COSY, GIOS and TRANSPORT.

Depending on the complexity of the task, TRACKv35 can compile up to a total of 10^6 particles on a regular desktop PC. For 10^4 particles of the 200 driver linacs of RIA, it takes about 70 hours using a 1.7GHz processor speed. (This is more than 1200 elements including ~400 rf resonators and 16 bending magnets.) The multiprocessor version of the TRACKv35 simulates 10^6 particles for ~360 linacs with randomly seeded errors.

3. Three-dimensional electromagnetic field distribution

3D electromagnetic fields of resonators are extracted from CST MWS (or any other electrodynamics code) as ASCII files **field_e.txt**, **field_h.txt**. For example, MWS can extract 3D fields in the meshes of the aperture area as a Microsoft EXCEL *.xls files. The EXCEL file should be saved as a MS DOS txt file **field_e.txt** or **field_h.txt**. In TRACKv35, the transport of a charged particle is described by the equation of motion:

$$\frac{d\vec{p}}{dt} = q(\vec{E} + \vec{v} \times \vec{B}), \quad (1)$$

where \vec{p} is the particle momentum and q is its charge, $\vec{E} = \vec{E}_{ext} + \vec{E}_{int}$ and $\vec{B} = \vec{B}_{ext} + \vec{B}_{int}$ are the sums of the external and internal electric and magnetic fields, \vec{v} is the particle velocity. TRACKv35 integrates the equations of motion of all the tracked particles for a short distance and calculates the space charge fields. In TRACKv35, particle motion through each ion-optical device can generally be described in three different Cartesian coordinate systems(CCS). Two CCS correspond to the entrance and exit of each ion-optical element. The third CCS is used for the definition of the electromagnetic field distribution in the element as is seen from Fig.A5-1 (Appendix 5). Depending on the geometry and the

type of element, external fields in the code can be defined using any of the following formats:

1. Three-dimensional tables of the fields \vec{E} and \vec{B} in the element's CCS which are generated with the help of external codes. For calculation of the field value at the particle location, a quadratic interpolation routine is used.
2. Two-dimensional tables in the (r, z) plane for elements with axial symmetry such as solenoids or Einzel lenses. These elements can be also described by 3D fields.
3. Two dimensional tables of the B_y component of the magnetic field in the median plane $\{x, z\}$ for rectangular dipole magnets. The off-median component B_y and components B_x and B_z are evaluated using the method described in ref. 4 of Appendix 5.
4. The fringe field falloff for dipole and multipole elements is described by a six-parameter Enge function [see ref. 5-6 in Appendix 5]:

$$F(z) = \frac{1}{1 + \exp(a_0 + a_1(z/D)^1 + \dots + a_5(z/D)^5)}$$

where z is the distance along the line which is perpendicular to the effective field boundary, D is the full air-gap of the element.

4. Launching the code **MWSread**

The code **MWSread.exe** is a result of significant modifications of the code **TRANSFORM.EXE**. The latter has been written to support earlier versions of the TRACKv35. Both the codes **TRACKv37.exe** and **MWSread.exe** must be compiled with equal number of meshpoints NXmax, NYmax, NZmax. Typical numbers are NXmax=Nymin=25 and NZmax=201. The latter can be a larger number if necessary. The code MWSread.exe performs the following:

- a) Reads ASCII files produced by EM codes in the region of beam-device² interaction. If spatial symmetry conditions are applicable for the field calculations in the EM and MWS codes, use them. For the best accuracy of simulations, meshpoints in the beam-device interaction area of the EM or MWS model should be larger than NXmax, NYmax, NZmax. However, the code accepts lower number of mesh-points too.
- b) Interpolates the field in order to produce Ex,Ey,Ez,Hx,Hy,Hz fields on regular mesh to be used by TRACKv35. The output field file is named as **eh_MWS.#01** for RF devices or **eh_EMS.#01** for static devices. They are unformatted binary FORTRAN files.
- c) Code **MWSread.exe** produces four auxiliary output files: **output.dat**, **egarm.dat**, **bgarm.dat** and **check_field.dat**.

For each device one should have three (for static electric or magnetic device) or four (for electromagnetic and static combined field devices) files in the same directory: **MWSread.exe**, **input.dat** and two ASCII-files of electric and magnetic fields for RF or static electromagnetic devices or ASCII-file of electric field for electrostatic device or ASCII-file of magnetic field for magnetostatic device. Make a separate directory for each device. This directory consists of the executable file **MWSread.exe** which should be run prior to the code **TRACKv35.EXE**.

Launch the code **MWSread.exe**.

4.1 Input Data

² In this write-up the term “device” and “beam-line element” are equivalent.

Note: 1) The file **input.dat** prepared for the code **transform.exe** (previous version of **MWSread.exe**) is compatible with **MWSread.exe** only for RF cavities.
 2) In the earlier versions of **TRACK** (earlier than version 34), the '**TRACK+TRANSFORM**' array for electrostatic devices was generated assuming artificial zero magnetic field array. An artificial zero electric field array was generated for magnetostatic devices. The full geometry of the device without any spatial symmetry had to be defined. Only under these restrictions, static field files **eh_EMS.***** created by **transform.exe** are compatible with any version of the **TRACK** code.
 3) It is recommended to redefine all field files generated by the file **transform.exe** for static devices.

The file **input.dat** contains a string *namelist/fasa/*. The *namelist* variables are listed below.

field defines the type of the device field.
Field = 'rf' stands for RF fields (default value)
 'eh' stands for static electromagnetic field
 'e' stands for static electric field
 'h' stands for static magnetic field.
symmetry defines an electromagnetic symmetry of fields.
Symmetry = 'rf' for RF cavities (default value)
 'dipole' for rectangular dipole magnet with symmetry w.r.t. XZ coordinate plane
 'axial' for devices with axial symmetry w.r.t. Z-axis.
 'quad' for devices with quadrupole symmetry w.r.t. Z-axis.
 'nosym' means the lack of any spatial symmetry of the fields.
file_MWS_e and **file_MWS_h** are the file names for input ASCII files of electric and magnetic fields. It can be any name allowed by **FORTRAN**. For example: **E.fld**, **H.fld**, **einzel_from_opera.dat**, **eRF.MWS**, **efield.txt** etc.

RF electromagnetic fields:

file_MWS_e= file name of the ASCII file for electric field
file_MWS_h= file name of the ASCII file for magnetic field

Electrostatic field:

file_MWS_e= file name of the ASCII file for electric field
file_MWS_h= 'empty' (default, optional)

Magnetostatic field:

file_MWS_e='empty' (default, optional)
file_MWS_h= file name of the ASCII file for magnetic field

d_device - the length of the device, d, the field extension along the z-axis.
z_device - (default value is =0) z-coordinate of the device field CCS w.r.t. the CCS of external 3D-field code, see Fig **xxx**.
aperture = device half-width **a** in **x** and **y** directions.

key_1_e defines symmetry w.r.t. YZ coordinate plane w.r.t. both the device CCS and 3D-field code CCS.
key_2_e defines symmetry w.r.t. XZ coordinate plane w.r.t. both the device CCS and 3D-field code CCS.
key_3_e defines symmetry w.r.t. XY plane only when origins of the device CCS and 3D-field code CCS are the same,i.e. **z_device=0**. For **z_device≠0** **key_3_e** must be equal to 0.
key_1_e, **key_2_e**, **key_3_e** =1, there is symmetry.

`key_1_e, key_2_e, key_3_e =0`, there is no symmetry.

`te00` - defines field level of the RF device or inter-electrode voltage for the electrostatic devices
`freqb[Hz]=` operating frequency of the given type of RF cavities.
`CFE=(-1 or +1)`, defines the sign of the electric field.
`CFH=(-1 or +1)`, defines the sign of the magnetic field. `CFE` and `CFH` are the amplitudes of the RF electric and magnetic fields **respectively**, which extracted from the external 3D-field electromagnetic code output.

The signs of the electric and magnetic field amplitudes are defined from the time dependence of the RF fields and can be different in different electromagnetic codes. In **TRACKv35** all RF-fields have the following time dependence:

$$\vec{E}(\vec{r},t) = \vec{e}(\vec{r}) \cos(\omega t + \phi_0), \quad \vec{H}(\vec{r},t) = \vec{h}(\vec{r}) \sin(\omega t + \phi_0), \quad \omega = 2\pi \cdot freqb,$$

where ϕ_0 -is an initial phase. From Maxwell equation $\text{rot} \vec{H} = \epsilon_0 \partial \vec{E} / \partial t$ one can find:

$$\oint \vec{H} d\vec{l} = \sin \phi_0 \oint \vec{h} d\vec{l} = \epsilon_0 \oint d\vec{s} \partial \vec{E} / \partial t = -\epsilon_0 \omega \sin \phi_0 \oint \vec{e} d\vec{s} .$$

For the circle with the radius $r \leq 0.5 R_{\text{aperture}}$ (perpendicular to the cavity z -axes), one can find $eds = e_z(r, \theta, z) r dr d\theta \equiv e_z(0, 0, z) r dr d\theta$ and $hdl = h \theta r d\theta$ therefore

$$r \int_0^{2\pi} h_g(r, \theta, z) d\theta \equiv -\epsilon_0 (2\pi/f) \pi r^2 e_z(0, 0, z)$$

$$I_H(z) = -\frac{freqb}{2\pi^2 \epsilon_0 r} \int_0^{2\pi} h_g(r, \theta, z) d\theta \equiv e_z(0, 0, z) = I_E(z)$$

The electric field amplitude will be multiplied by `CFE*te00` in the **MWSread.exe** code. Similarly, magnetic field is multiplied by `CFH*te00`. Both functions $I_H(z)$ and $I_E(z)$ are evaluated for RF cavities and z , $I_H(z)$ and $I_E(z)$ are extracted in the file **check_field.dat**. The code **MWSread.exe** compares functions $I_H(z)$ and $I_E(z)$ and will ask to define necessary sign of electric or magnetic field using the parameters `CFE` and `CFH` if initial `CFE` and `CFH` are wrong. It can be necessary to change the sign either `CFE` or `CFH`.

Example of output information on the PC screen and in the **output.dat** file:

`CFE=1, CFH=-1`

Warning: CFE and CFH are inappropriate.

Please insert new CFE and CFH and press enter

`-1 -1` or

`1 1` are possible answers.

It is very useful to plot and compare functions $I_H(z)$ and $I_E(z)$.

Note: for the RF cavity with negligible magnetic field copy **e_field.dat** to **h_field.dat** and use `CFE=1, CFH=0`. Ignore the warning about `CFE` and `CFH` and insert 1 and 0 for `CFE`, `CFH` one more time.

Running **MWSread.exe** with different set of parameters `d_device` and `z_device` one can divide 3D-field table into several zones. This feature is useful for representation of the fringing fields. Another reason could be to insert long devices as separate zones.

`Ncells ≥ 1` (optional for **MWSread.exe**) → virtual number of "cells", have been used in **transform.exe**.

`Abeta` (optional) → virtual beta, has been used in **transform.exe**.

Three parameters `freqb`, `Ncells` and `abeta` have been used in the old code for calculation purpose only.

$$d_{\text{device}} = 0.5 * abeta * c_{\text{light}} * ncells / freqb.$$

In **MWSread.exe** parameters **Ncells** and **abeta** are also available but parameter **d_device** is more convenient for defining the extension of the field distribution along the z-axis.

Variables **NCAV**, **atp**, **ztp**, **targ**, **drft**, **tt0** are optional and included in **Input.dat** file only for compatibility with previous version **transform.exe**.
r_fourier (=0.5 by default) the relative radius for Fourier analysis of the fields at $r=r_{\text{fourier}} \times \text{aperture}$
Parameters **iread** (=0, default value), **iprint** (=0, default value), **Vf** and **Bf** are used to analyze the properties of the field distribution obtained by the external 3D-codes for the devices defined in **sclinac.dat**.

```
iread =0 the files eh_MWS.#01 or eh_EMS.#01 are created.
iread =n old files eh_MWS.#n or eh_EMS.#n are read and analyzed.
Vf      amplitude of the electric field.
Bf      amplitude of the magnetic field.
iprint=1 ASCII files with electric and magnetic fields on
          the uniform-mesh spatial grid of the TRACk code are extracted.
```

4.2 Output Data

The code **MWSread.exe** produces binary **eh_MWS.#01** file for the RF devices or **eh_EMS.#01** file for the static devices. The following line must be inserted into the **sclinac.dat** file to represent a device with realistic 3D field distributions:

```
n elem ... d_elem ... te00 ...      rap ... for RF devices
n elem ... d_elem ... Bf   ...      rap ... for magnetostatic devices
n elem ... d_elem ... Vf   ...      rap ... for electrostatic devices
n elem ... d_elem ... Bf,Vf ...    rap ... for static devices
```

where

n - is the number of file containing the field tables (consistent with the extension of the file **eh_MWS.#xx**).
elem - is the name of the element(**cav**, **eq3d**, **enzl** ...)
d_elem length of the device
te00, **Bf**, **Vf** amplitudes of the field.
The binary file **eh_MWS.#01** produced by the code **MWSread.exe** **eh_MWS.#01** must be renamed to

```
eh_MWS.#0n if 0<n<9
eh_MWS.#n  if 10<n<99
Examples: 5 cav ..      eh_MWS.#01 -> eh_MWS.#05
           13 cav ..     eh_MWS.#01 -> eh_MWS.#13
           96 eq3d ..    eh_EMS.#01 -> eh_EMS.#96
           1 eq3d ..    eh_EMS.#01 -> eh_EMS.#01
```

The code **MWSread.exe** produces **check_field.dat** for the set of mesh-points $z_i \{0 \leq z_i \leq d_{\text{device}}\}$
if **symmetry='rf'** , then $z_i, I_H(z_i), I_E(z_i)$;
if **symmetry='axial'** , then $z_i, V(z_i)$, where V is the normalized voltage along Z- axis for axially symmetric electrostatic fields;
if **symmetry='axial'** , then $z_i, B_z(z_i), B_{\text{hard}}(z_i)$, where $B_z(z_i)$ is the normalized z-component of the magnetic field $B_z(z_i)$, for axially symmetric magnetostatic fields.

Typical set of parameters in **input.dat** for RF cavity:
& freqb = 100d6,

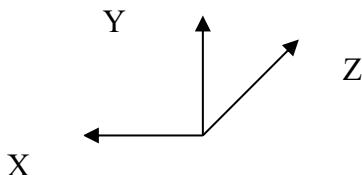
```

cfe=1,cfh=-1,te00=0.9,
d_device=50. ,aperture=1. ,
file_MWS_e='e.txt' , file_MWS_h='h.txt' ,
key_1_e=1, key_2_e=1, key_3_e=0
& end
Typical set of parameters for solenoid:
& symmetry='axial'
  d_device=50. ,aperture=1.
  file_MWS_h='solenoid.txt'
  key_1_e=1, key_2_e=1, key_3_e=1
& end
Typical set of parameters for electrostatic quad:
& symmetry='quad'
  te00=1
  d_device=20. ,aperture=3.
  file_MWS_e='equad.txt' ,
  key_1_e=0, key_2_e=0, key_3_e=0
& end
Typical set of parameters for Einzel lens:
& symmetry='axial'
  te00=1
  d_device=10. ,aperture=0.5
  file_MWS_e='e.txt'
  key_1_e=0, key_2_e=0, key_3_e=1
& end

```

5. TRACK Input Files

The simulation is performed in right-handed Cartesian coordinate system.



Z is the direction of the beam propagation.

To run this code the following files must be copied to a designated directory³:

```

TRACKv37.exe = simulation code
fi_in.dat = has the input "synchronous" phases of all SRF cavities
Track.dat = contains general input parameters of the beam and linac
sclinac.dat = file containing geometry of the Linac (focusing,
accelerating structure and other elements)
graph.cfg = input file to support on-line Windows graphs.
cavity.#01 = for the first type of cavity (not needed if iflag_cav=1), in
recent simulations using TRACKv35.exe we do not use this file.
eh_MWS.#01 = for the first type of cavity. If sclinac.dat contains more
types of cavities all files with appropriate extensions should exist.

```

³ LINUX version of the TRACKv37 exists and the Manual is being written.

```

solenoid.#01 = if isol=1 (file of realistic field distribution in the
solenoid)
deck.#01      = transition to/from HV deck, axial symmetric lens.
RFQ.#01       = RFQ data.
MHBrz.#01     = field distribution in rf buncher (2-gap resonator, can
operate at 2 frequencies simultaneously)
stripper.#01   = stripper data.

```

The number in the file extension corresponds to the successive number of similar elements along the linac.

5.1 Track.dat

The input data is defined using the FORTARN statement 'namelist'.

```

namelist /TRAN/ Win,atp,freqb,sfas,serf,npat
&      ,nqtot,part,qq,epsnx,alphx,betx,epsnsy,alphy,bety,DEESRF
&      ,epsnz,alphz,betz,aper,x00,xp00,y00,yp00,ph00,dww00,phi0
&      ,disp,dcav,phmax,dwwmax,db_b,Qavr,df_4D,Qdesign, Adesign
&      ,current,amass,dWWacc
&      ,n_tp,n_tt,ncells,np_set

```

The input file consists of the following data (this is an example of simulation of the whole RIA driver linac from 0.19 MeV/u to 400 MeV/u) :

```

&TRAN


```

Line

```



is given as an example. This line shows the directory where all files are located with the tables of device fields. All files with the extension .#xx are located in this directory. There is no need to copy these files into the directory where the input files with the extension .dat and TRACKv35.exe are located.


```

This line

```

work_dir = ' '

```

will serve similar function as **table_dir** for the output files. Will be modified in future.

```

WIN [eV/u]= input beam energy.
atp = mass number.
nqtot= number of different ion species.

```

`qq(1:nqtot) [Q/e]` - charge state of each species.
`npat(1:nqtot)` - number of macro-particles with given q/A. If beam `current` is not equal to zero, final adjustment of the number of particles takes place in the code: each multi-particle will carry the same charge for the space charge calculation routines.
`np_set` - number of particles in each charge state for the accelerator reference setting. This setting is required if the flag `iflag_corr=1` and the steering correction is activated. The steering correction of the beam centroid in transverse phase space is required to avoid emittance growth of multi-q beams and minimize the coherent oscillations in the transverse phase space. To search for optimum fields in the correctors several complete runs of the linac must be done. To run for this optimization procedure the setting of the linac must be defined.
`Qavr` - average charge state, it is used for 2-charge state injecton
`Qavr~0.5*(qq(1)+qq(2))`. This parameter is effective upstream of the RFQ, in the LEBT.
`Adesign` - mass number of the design particle for the accelerator or transport system.
`Qdesign` - charge state of the design particle for the accelerator or transport system.
`Amass(1:nqtot)` - ion mass number, useful for simulation of various masses exiting the ion source.
`Freqb(Hz)`, - fundamental frequency of the incoming beam. All rf resonators frequencies are set with respect to `freqb` using a harmonic number. This number can be an integer or a non-integer. For dc beams one can use the frequency of the first rf device located downstream of the ion source.
`current(1:nqtot) [mA]` electrical current of each heavy ion beam component.
`part <=1` for calculation of emittances of beam, containing '`part`' particles with respect to the total number of survived particles.
`df_4D` - phase width (+/-) of the initial 4-dimensional distribution (dc beam) if the flag `iflag2D=1` is used.
`epsnx,alphx,betx`; [cm*mrad], [unitless] [cm/rad] → initial transverse parameters of beam, emittance is total and normalized.
`epsny,alphy,bety`; [cm*mrad], [unitless] [cm/rad] → initial transverse parameters of beam, emittance is total and normalized.
`epsnz,alphz,betz` [deg*%], [unitless], [deg/%] → initial longitudinal parameters of beam, degrees at frequency `freqb`. `epsnx`, `epsny` are the full normalized emittances in $\pi \cdot \text{cm}^2 \cdot \text{mrad}$. `epsnz` is the longitudinal emittance in $dW/W(\%)$ and phase (degree of `freqb`) of current frequency `freqb` (this emittance is not invariant). Currently in TRACKv35 we generate 4D or 6D water bag distributions. In this case $\epsilon_{TOTAL} = 6\epsilon_{RMS}$ for the 4D Waterbag and $\epsilon_{TOTAL} = 8\epsilon_{RMS}$ for the 6D Waterbag.
`phmax` - is calculated using subroutine PHASSET
`dwwmax` - [%], amplitude of energy spread of the initial distribution for the acceptance calculation.
`Dwwacc` - [%], $\Delta W/W$ for the acceptance calculations, if particle energy is lower than `dwwacc` then this particle will be treated as an unaccelerated particle.
`x00 (1:nqtot) [cm]` - initial displacement of center of the beam
`xp00 (1:nqtot) [mrad]` - at entrance of accelerator for each
`y00 (1:nqtot) [cm]` - charge state
`yp00 (1:nqtot) [mrad]` - *-----*

```

ph00 (1:nqtot) [deg]-
Dww00(1:nqtot) [%]-
*----*
*----*
db_b - delta_beta/beta for dc beam exiting ion source.

```

DEESRF – the relative range (like 0.2, for example, which corresponds to $\pm 20\%$ field deviation from the design field) of SC cavity field with respect to the average design value. This feature can be activated to study beam evolution along the linac with different level of fields in the SC resonators. The field level in the resonators is generated using uniform random distribution within $\pm \text{DEESRF}$.

The track.dat file has a section with integer values and flags:

```

namelist /INDEX/ isol,lstep,iaccep,iaxial,NRZ,igraph,isrf
&,iflag_lev,IINT,iflag_env,iflag_cav,iflag_ell,nstep_cav
&,iflag_bc,iflag_dis,iflag_qq,iflag2D,iflag_fenv
&,iflag_mhb,iflag_upd,iflag_rms,iflag_tac,iflag_halo,iflag_corr
&,iwrite_dis,iread_dis,iRIARFQ,nrz_min

```

This is the example of the integer parameter list in the **track.dat** file:

```

&INDEX
NRZ=1, igraph=1, iaccep=0, isol=1, iflag_dis=0,iflag2D=0,iflag_qq=1,
iflag_rms=1, iint=100, nstep_cav=50
iflag_env=1, iflag_cav=1, iflag_ell=0, iflag_fenv=0
iflag_upd=0, iflag_halo=0, iflag_lev=1, isrf=100
iwrite_dis=1, iRIARFQ=0,iflag_corr=0
&END

```

The parameters given below are shown with default values.

```

NRZ=1      - number of seeds for error and misalignment simulations, for single
              simulations NRZ=1. If NRZ not equal 1 use igraph=0 just to save CPU
              time. The graphs are not completely supported for NRZ $\neq$ 1.
igraph=1   - show graphs on PC screen, =0 no graphic output
iaccep=0   - if 1 the longitudinal acceptance is calculated. See details in
              Appendix 3.
isol=0     - 1-realistic fields, 0-rectangular model. If isol=1 the code
              TRACKV35.exe will search for ASCII data file solenoid.#01 with the
              distribution of axial symmetric magnetic field along z. If solenoid
              is defined with 3d-field distribution and element ''sol3d'' then the
              parameter isol is ignored.
ISRF=0     - the seed of the random number generator ISRF must be in the range
              (0, 2147483646). If ISRF is zero, a value is computed using the
              system clock; and, hence, the result of the program using the IMSL
              random number generators will be different at different times.
IINT=50    - number of steps for Runge-Kutta integrator which is applied for
              drift spaces and solenoids.
iflag2D=1  starts with DC beam and calculates only 2D electric fields due to
              space charge, in z direction space charge field is equal to 0.
iflag_qq=1 will show phase space plots for indicated ion species or charge
              state of given ion species.
iflag_env=1 number of graphical outputs along one element (if  $\neq 1$  it is not
              supported in all elements)
iflag_fenv=1 extracts rms phase [deg] on the PC screen (shown in green color).
              Check vertical scale to see the curve.

```

```

iflag_cav=1 RK integration in the accelerating cavities, 0 - use iterative
               method (as in the LANA code). Do not use 0 - it is obsolete.
iflag_ell=1 Draw ellipses on the PC screen, if =0 do not draw ellipses.
nstep_cav=50 number of steps for RK integrator in the rf cavity.
iflag_dis=1 6D waterbag distribution, if =0 is a 4D waterbag distribution
iflag_upd=0 - this is an internal flag, not to be used.
iflag_rms=1 Show rms envelopes on the PC screen, =0 do not show rms envelopes
iflag_tac=0, Do not calculate transverse acceptance, if =1 calculate transverse
               acceptance. See details in Appendix 3.
iflag_halo=0 if =1, generates and simulates halo particles after the stripper
               #2. This feature is not ready to be used.
iflag_mhb=0 MHB is calculated on the base of 3d table
iflag_bc=0 phase setting is not adjusted for average bunch center before the
               each cavity. It is important that for the acceptance calculations
               iflag_bc must be 0.
iflag_lev=0 → do not calculate N/N0(relative beam intensity) as a function of
               emittance(f( $\epsilon$ )). If this parameter =1, then N/N0 as f( $\epsilon$ ) is reduced
               to the rms emittance for xx',yy' and the{ $\varphi$  -  $\Delta W/W$ }-planes will be
               extracted in the file level.dat. Use this flag with element 'prmtr'
               to extract 'emittance distribution' in given location along the
               structure.
iwrite_dis=1 - writes unformatted data file read_dis.dat of the particle
               distribution at the end of simulation (at the end of last element in
               sclinac.dat file) for future use as an initial distribution.
iread_dis=1 - reads file read_dis.dat of the particle distribution and starts
               simulation with this distribution ignoring generation of the initial
               distribution. Use this flag to read custom formed initial
               distribution (see Appendix 6).
iflag_corr=1 - Calls beam centroid steering subroutine to find corrector
               strength (see Appendix 7). iflag_corr =-1 is required to start
               simulations for many seeds. In addition see Appendix 1 to set up the
               simulations of errors and misalignments.
IRIARFQ=0 - simulates single or multiple charge state beam within 360° of the
               RFQ frequency (inside one bucket). This option is not valid for
               simulation of two charge-state beam with space charge as for RIA
               RFQ. If IRIARFQ=1, then simulation takes place within 720° and 2
               neighboring buckets can be populated with different charge states
               (or the same charge state). Use IRIARFQ=1, to simulate RIA driver
               linac RFQ in two-charge state mode with MHB and velocity equalizer.

```

The following values are set by default in the **TRACKv35** code:

```

isol=0
iaccep=0
i_step=0 index for the array of central trajectory in the bending magnet,
               it is used for the space charge calculations
isrf=0
part=0.
db_b=0.
Qdesign=28.5
Adesign=238.
nstep=100 This parameter depends on space charge calculations
               and it is better to define the integration step for each
               device in the structure file sclinac.dat.
nstep_cav=50 !Number of integration steps in resonators.

```

5.2 **sclinac.dat**

The file **sclinac.dat** defines the following elements of the accelerating-focusing channel or transport line. Each element in the TRACK code has own ID:

```
c      i_device=0      ! no device assigned
c      i_device=1      !*DRIFT *!
c      i_device=2      !*SOLENOID*, *HARD-EDGE SOLENOID*!!
c      i_device=3      !*Bending Magnet*!
c      i_device=4      !*QUAD **HARD-EDGE QUAD*!
c      i_device=5      !*BUNCHER*!
c      i_device=6      !*CAVITY *!
c      i_device=7      !*CORR*!
c      i_device=8      !*MULTIPOLE*!
c      i_device=9      !*RFQ*!
c      i_device=10     !*MHB*!
c      i_device=11     !*HV DECK*!
c      i_device=12     !*SLIT*!
c      i_device=13     !*F_HIDE*!
c      i_device=14     !*MONITOR*!
c      i_device=15     !*EQUAD3D*!      electrostatic quadrupole with 3D mesh
c                      from EM Studio
c      i_device=16     !*EINZEL*!      electrostatic lens with 3D mesh from EM
c                      Studio
c      i_device=17     !*EQUAD*!      electrostatic quadrupole
c      i_device=18     !*SOL3D *!      solenoid with 3D mesh from EM Studio
c      i_device=19     !*cdump*!      This marker defines correction section
c                      boundaries
c      i_device=21     !*extrc*!      A device with 3D electrostaic field
c      i_device=22     !*UDS*!      User Defined structure (multi-gap resonator
c                      like DTL )
c      i_device=23     !*shrt*!
c      i_device=24     !*strip*!  STRIPPER
c      i_device=25     !*EH3D*!  static field (combined electric and magnetic
c                      field)
```

5.3 **Fi_in.dat**

The file **fi_in.dat** for the **TRACKv35.exe** must be generated manually and it contains the phase angle with respect to the maximum energy gain in a given resonator. It should be negative for stable longitudinal motion. For the bunch rotator the phase is -90 deg (or close to this number, and can be adjusted using simulated results, for example, to keep average beam energy unchanged). For a chain of similar accelerating resonators the phase angle is practically equal to the synchronous phase. As soon as one runs **TRACKv35.exe**, the code produces file **linac.dat** which combines **sclinac.dat** and **fi_in.dat** files. Using the data in file **linac.dat**, one can check the final phases assigned to each resonator.

5.4 Solenoid.#01: Data file for 1D solenoid field profile

This file has been used together with element **sol** and the flag **isol=1** before the development of the element **sol3d**. The number in the file extension corresponds to the successive number of solenoids along the linac. If in **track.dat** the parameter **isol=1**, then **TRACKv35.exe** will search for ASCII data file **solenoid.#01** with the distribution of axial symmetric magnetic field along z. Below is the

typical field distribution of a solenoid with bucking coils. The first line is the number of mesh-points. The total length of the solenoid L_s is given in **sclinac.dat** file. The distribution is applied to this length L_s . This distribution should be smooth enough in order to avoid irregularities in higher order derivatives otherwise it can result in non-physical field distribution and emittance growth. The solenoid field map is calculated using 4th order derivatives.

37

```
-0.014143283
-0.025661771
-0.040576367
-0.054855518
-0.056903368
-0.027229919
0.053280013
0.191001852
0.369154663
0.547779134
0.692967978
0.798030467
0.870633884
0.920061736
0.953459121
0.975653094
0.98971838
0.99750644
1.000000009
0.99750644
0.98971838
0.975653094
0.953459121
0.920061736
0.870633884
0.798030467
0.692967978
0.547779134
0.369154663
0.191001852
0.053280013
-0.027229919
-0.056903368
-0.054855518
-0.040576367
-0.025661771
-0.014143283
```

5.5 Stripper.#01: Data file for stripper parameters

In addition to the stripper description in the **sclinac.dat** file there is a file **stripper.#01** (the number in the file extension corresponds to the successive number of similar elements along the linac). Below is the typical stripper file. First line: number of charge states after the stripper. Second line charge states, the first charge state will be used as a reference charge state for the phase setting in the resonators. The third line: percentage of particles at corresponding charge states. The total percentage is 100. The fourth line is equal to the second line (this line will be used in future).

The last line is the kinetic energy per nucleon after the stripper [keV/u].

```
5
72. 70. 71. 73. 74.
0.3 0.15 0.2 0.2 0.15
72. 70. 71. 73. 74.
10000.
```

5.6 RFQ.#01: Data file for RFQ parameters.

This is an ASCII file with the RFQ parameters on the base of a 2-term potential expansion. These parameters must be produced by RFQ optimization codes such as DESRFQ or some other code. It is important that the accelerating and focusing parameters are determined with the actual shape of vane modulations. The total number of RFQ accelerating cells nRFQ is given in the element "RFQ". The format for reading is:

```
read (1,*) (bRFQ(i),tRFQ(i),xRFQ(i),aRFQ(i),dRFQ(i),indRFQ(i),i=1,nRFQ)
```

There are 6 columns, the definitions corresponding to those in the original Kapchinsky-Teplyakov paper.

1: beta at the exit of the cell, this column is used just for reference, does not effect on beam parameters.

2: Parameter teta as is defined by Kapchinsky and $\text{teta}=\pi\text{A}/4$, where A is the accelerating efficiency defined by the LANL group.

3: Kappa;

4: Aperture radius [cm];

5: length of the accelerating cell [cm];

6: index, equal to 0 if the cell is of regular length $\beta\lambda/2$ otherwise =1.

5.7 Graph.cfg: Data file for graphics output.

The lines in this file are self-explanatory. In case of any doubts, just run **TRACKv35.exe** to identify the parameters of the graphics. Do not change the integer arrays. The graphics are not perfect: if particles are outside the graph boundary, they will be shown with "wrong" coordinates. In case of the RFQ, the particles are "filtered" by energy before sending for graphics.

```
&WINS
title = ' Superconducting Linac '
ix(1) = 114, iy(1) = 32, jx(1) = 322, jy(1) = 240, lx(1) = 4, ly(1) = 8,
xmin(1)= -1., xmax(1)= 1., ymin(1)= -0.01, ymax(1)= 0.01,
ix(2) = 394, iy(2) = 32, jx(2) = 602, jy(2) = 240, lx(2) = 4, ly(2) = 8,
xmin(2)= -1., xmax(2)= 1., ymin(2)= -0.01, ymax(2)= 0.01,
ix(3) = 674, iy(3) = 32, jx(3) = 882, jy(3) = 240, lx(3) = 4, ly(3) = 8,
xmin(3)= -30., xmax(3)=30., ymin(3)= -0.01, ymax(3)= 0.01,
ix(4) = 75, jx(4) = 950, iy(4) = 399, jy(4) = 599, lx(4) = 1, ly(4) = 5,
xmin(4) = 0., xmax(4) = 30000., ymin(4) = 0., ymax(4) =1.5
&end
```

5.8 Maximum size of arrays in the code.

Maximum possible numbers for the arrays in the TRACKv35 are given in FORTRAN Data file in the code. Depending on the code application and hardware parameters, the code can be compiled for different mesh size, number of macroparticles and number of ion species.

```

& NXmax=25 , NYmax=25 , NZmax=201 ! grid for dist. of field in rect reg.
&, KSTEPR= 8,JSTEPA= 16 ,LSTEPZ= 48 !max number of grids for field distribution
of axial-symmetric electromagnetic field of resonators.

&, PI= 3.141592653589793d0          ! [rad]
&, PIOVHR= PI/180d0                  ! [rad/ø]
&, HROVPI= 180d0/PI                  ! [rad/ø]
&, CC= 2.99792458d10                ! [cm/sec]
&, C_light= 2.99792458d8            ! [m/sec]
&, EPS0= 1/(4.d0*PI*1.d-9*CC**2)    ! [F/cm]
&, EE= 4.803242d-10                 ! [CGSe]
&, EV= 1.60217733d-12              ! [erg/eV]
&, AMU= 931.49432d6                 ! amu [eV]
&, W0= 1.0073*amu                   ! proton rest mass [eV]
&, emass= 510.079                   ! electron rest mass [KeV]
&, ch_to_m=3.2184034535d-3         ! [C*sec/kg] = [1/T]
&, mu0=4d0*pi*1d-7                 ! [T/(A/m)]
&, twopi=2d0*pi                     ! [rad]
&, tovpi=2d0/pi                     ! [rad]
&, degrad=pi/180d0                  ! [rad/deg]
&, ngridx=32, ngridy=32, ngridz=64, nm3=16, nm5=33 !For Space Charge field mesh.
&, MNTot = 100000                  ! maximum number of particles of each type.
&, Melem = 2000                     ! maximum number of elements (drifts, magnets...)
&, MQtot = 20                       ! total number of charges (masses)
&, CRO=4d0*pi**2/3.13d10          ! [1/milliamper], for space charge calculation

& NXmax=25 , NYmax=25 , NZmax=201 ! grid for dist. of field in rect reg.
! Attention: For simulation of long DTLs use NZmax=801 for compiling files
! This requires appropriate generation of field files by transform.exe
&, maxNseed=200                     ! max number of seeds
&, maxNCorr = 150, maxNMon = 100     ! max number of correctors and monitors
&, maxNSection = 30                  ! max number of correction sections.
&, Nbuf=200000                      ! max number of buffer size for
graphics, can be suspended for non-graph option
&, PI= 3.141592653589793d0          ! [rad]
&, PIOVHR= PI/180d0                  ! [rad/deg]
&, HROVPI= 180d0/PI                  ! [rad/deg]
&, CC= 2.99792458d10                ! [cm/sec]
&, C_light= 2.99792458d8            ! [m/sec]
&, EPS0= 1/(4.d0*PI*1.d-9*CC**2)    ! [F/cm]
&, EE= 4.803242d-10                 ! [CGSe]
&, EV= 1.60217733d-12              ! [erg/eV]
&, AMU= 931.49432d6                 ! amu [eV]
&, W0= 1.0073*amu                   ! proton rest mass [eV]
&, emass= 510.079                   ! electron rest mass [KeV]
&, ch_to_m=3.2184034535d-3         ! [C*sec/kg*(m/cm)] = [1/(T*cm)]
&, mu0=4d0*pi*1d-7                 ! [T/(A/m)]
&, twopi=2d0*pi                     ! [rad]
&, tovpi=2d0/pi                     ! [rad]
&, degrad=pi/180d0                  ! [rad/deg]
&, ngridx=32, ngridy=32, ngridz=64 !For space charge field grid

```

```

&, nm3=16, nm5=33           ! nm=max(nx,ny,nz/2)
                           nm3=nm/2, nm5=nm+1, For SC grid
&, MNTot = 250000          ! maximum number of particles
&, Melem = 2000            ! maximum number of elements (drifts, magnets...)
&, m_cell=15               ! total number of cells in the cavity
&, maxn_tp=99              ! number of different cavity types
&, MQtot = 5                ! total number of charges (masses)
&, nm_halo = 10000 )       ! max number of halo particle generator,

```

Attention: For simulation of long DTLs with large number of accelerating cells (more than 10) use NZmax=801 for compiling **TRACKv35.exe**. This requires appropriate generation of field files by **MWSread.exe**. The same mesh number along the z-axis must be used in the code **MWSread.exe** during the compiling. Depending on PC memory the code **TRACKv35.exe** can be compiled with required dimensions of the arrays.

6. Beam line elements supported by TRACK

6.1 RF devices

ACCELERATING CAVITY

n cav d_elem harm TE00 any nstep(optional)

n	type of cavity: each type of cavity must have field files with the name eh_EMS.#** .
d_elem [cm]	total length of cavity
harm	harmonic number of cavity with respect to the fundamental beam frequency freqb defined in the TRACK.dat file
TE00	field level of cavity. This parameter is equivalent to the parameter TE00 in the input.dat file for MWSread.exe . If used in the sclinac.dat file it can define the field level in a particular resonator.
any	no longer in use in TRACKv35. In previous versions it meant shift of the cavity in vertical direction. Use shift device for this purpose.
nstep	(optional)number of steps for integration of the cavity. If it is voided, then nstep = nstep_cav given from TRACK.dat .

Misalignment errors of cavity displacement in transverse directions and static and dynamic errors of the RF phase and amplitude are defined by the device 'align'.

Example:

```
5  cav    43.79542  3.   1.0   0.0      here nstep=nstep_cav
5  cav    43.79542  3.   1.0   0.0   500 here nstep=500
```

MULTI-CELL RFQ

ncells rfq Vf d_elem R0RFQ RFQ_ph0 harm nstep n ncoef

ncells	number of $\beta\lambda/2$ -length cells in the RFQ
Vf [kV]	Inter-vane Voltage
d_elem [cm]	total length of the RFQ
R0RFQ [cm]	average radius

```

RFQ_ph0 [deg] Phase of the RFQ field with respect to the incoming pre-bunched
beam.

harm harmonic number of the RFQ cavity with respect to the
freqb defined in the track.dat file.

nstep number of steps for integration per cell ( $d_{cell} = 0.5\beta\lambda$ ), were
 $\lambda=c/(harm*freqb)$  is current RF wave length and  $\beta_c$  - is velocity
of the reference particle

n (=1 by default, optional) number of the extension of the file
with name rfq.#n

ncoef (ncoef=2 by default, optional) number of field coefficient in Fourier-
Bessel expansion of the filed in RFQ cell.

```

Examples:

1) Two coefficients field distribution. Input file is **rfq.#1**
138 rfq 68.47 394.3084938 0.6 90. 2. 50

1) Eight coefficients field distribution. Input file is **rfq.#5**
138 rfq 68.47 394.3084938 0.6 90. 2. 50 5 8

1) Five coefficients field distribution. Input file is **rfq.#1**
138 rfq 68.47 394.3084938 0.6 90. 2. 50 1 5

More details of the potential expansion in the RFQ are given in Appendix 4.

Modifications, October 4, 2005

RFQ can consist of entrance and exit regions as 3D tables obtained from EM studio. In order to have possibility to optimize vane shapes in the end regions iterative procedure of EM calculations and TRACK simulation may be necessary. To do this there is an option to write a scratch file inside the RFQ. An example follows.

```

1 prmtr iflag_t=1
-68 scrch 260
263 rfq 90.4 303.442 .34 -90. 1.0 100 3
This line makes scratch file scrch.#68 after cell #260. The RFQ input file is
rfq.#03.

```

Reading of the scratch file.

```

1 prmtr iflag_t=1
68 scrch 260
263 rfq 90.4 303.442 .34 -90. 1.0 100 3

```

TWO-HARMONIC BUNCHER

```

n mhb Ef1 Ef2 d_elem rap MHB_ph0 any any MHBh1 MHBh2 nstep
n number of the extension of the file with name eh_EMS.## 3D table
field MHB or file with name eh_2d.## for 2D table field MHB
Ef1 [V] amplitude of the voltage of the first harmonic
Ef2 [V] amplitude of the voltage of the second harmonic
d_elem [cm] total length of the MHB
rap [cm] aperture radius
MHB_ph0[deg] initial phase set of the MHB
any not in use
any not in use
MHBh1 first harmonic number of MHB with respect to the fundamental

```

MHBh2 beam frequency `freqb` given in the **TRACK.dat** file
nstep second harmonic number of MHB with respect to the fundamental
 beam frequency `freqb` given in the **TRACK.dat** file
 (optional)number of steps for integration for the MHB.
 If this parameter is voided, `nstep = int` which is defined in the **TRACK.dat**.

Example:

```
1 mhb      2500.   0.  4.65728   2.0    14.5   257   16
```

Comment. The MHB can be defined as a sequence of resonators (element 'cav'). However, the element 'MHB' allows one to apply 2 harmonics of rf field with different amplitudes simultaneously.

Note:

This element is considered as an rf cavity with only electric field.

4-HARMONIC IDEAL BUNCHER

Element name: 'fhide'

```
c ctype      = el.n ! Type of the MHB to read the field distribution
c EMHB1    = el.p1 ! Amplitude of the ideal "Saw tooth" voltage
c FFAS     = el.p2 ! Phase of the ideal "Saw tooth" voltage (degree)
```

Example:

```
1 fhide    -0.015   40.
```

BUNCHER

This device is equivalent to the device *CAVITY* with `fi_in=-90°`.

n bunch d_elem harm TE00 nstep

```
n          type of cavity: each type of cavity must have field files with
           the name eh_EMS.#**.
d_elem [cm] total length of cavity
harm       harmonic number of cavity with respect to the fundamental
           beam frequency freqb given in the TRACK.dat file
TE00       field level of the cavity.
nstep      number of the integration step for the device
```

Example:

```
2 bunch    43.79542  3.   1.0    100
field distribution is defined in eh_EMS.#02
```

6.2 Magnetostatic devices

HARD-EDGE OR 1D TABLE FIELD SOLENOID

This element can be used to define either a hard-edge solenoid (`isol=0`) or a 1D table field solenoid (`isol=1`). 1D table format is described in chapter 5.4. We recommend using the element `sol3D` as a "soft edge" solenoid.

n sol Bf* d_elem heff rap rap*** nstep**

```
n=0 isol=0,1  hard-edge solenoid
n>0 isol=0    hard-edge solenoid
```

```

n>0 isol=1      n extension number of the file solenoid.#n
Bf      [G] field level in the solenoid
d_elem [cm] total length of solenoid
heff    [cm] effective length of solenoid, d_elem≥heff.
rap     [cm] aperture radius
rap***  [cm] placeholder
nstep*** (optional) number of integration steps for the 1D table
          field solenoid. If it is voided, nstep = iint given from
TRACK.dat.

```

Note:

- * Use normalization $B_z(0,0,d_{\text{elem}}/2)=1$ for solenoid field table.
 B_f is the field value in center of the solenoid.
- ** $heff$ has no meaning for 1D table field solenoid and is a placeholder.
- **** $nstep$ is placeholder for the hard-edge solenoid

Examples:

- 1) **isol=1.** Field is given in **solenoid.#01**. In this example **nstep = iint**.
The effective length is placeholder and can be used as a reference.
1 sol 110000. 30. 20. 3. 3.
- 2) **isol=1.** Field is given in **solenoid.#45**. In this example **nstep = 500**.
The effective length is placeholder and can be input for reference.
45 sol 110000. 30. 20. 3. 3. 500
- 3) **isol=0.** All lines below represent the same hard-edge solenoid.
 $heff$ must be defined as an input parameter. **rap**** and **nstep** are
used as the placeholders and can be skipped.

```

1 sol 110000. 30. 20. 3.
1 sol 110000. 30. 20. 3. 3.
1 sol 110000. 30. 20. 3. 3. 200
0 sol 110000. 30. 20. 3.

```

*** 2D OR 3D TABLE FIELD SOLENOID ***

The solenoid field is calculated by some external electromagnetic code.
2D and 3D table format is described in chapter 5.???

```

n sol3D Bf d_elem rap nstep

n>0      extension number of the file eh_EMS.#n
d_elem [cm] total length of the solenoid (the field extension along z)
Bf      [Gs] peak magnetic field at the center of the solenoid
rap     [cm] aperture radius
nstep    number of integration steps for the device.

```

Note:

The pre-processor code **TRANSFORM.exe** normalizes solenoid field table to provide
 $B_z(0,0,d_{\text{elem}}/2)=1$.
 B_f is the field in the center of the solenoid.

Examples:

- 1) Field is given in **eh_EMS.#01**
1 sol3d 110000. 30. 3. 100
- 2) Field is given in **eh_EMS.#67**

67 sol3d 110000. 30. 3. 50

* DIPOLE MAGNET, ROUNDED POLE FACE*

Internal or user defined set of Enge coefficients $c_0 \div c_5$ can be used for the fringe field calculation. User can define Enge coefficient using the commands **1 enge bmag c0 ..c5** and **2 enge bmag c0 ..c5**. The number 1 (2) in the first position defines coefficients for the entrance (exit) edge ???.

n bmag d_elem rbend theta airgap width bet1 bet2 r1_inv r2_inv nstep

n=0 the fringe field is calculated analytically, Eq. (A5-1,A5-2)
n>1 extension number of the file **eh_EMS.#n** containing
the entrance fringe field 2D table. A file **eh_EMS.#n+1** must
contains 2D table for the exit fringe field, see Fig. A5-4B
2D table format is described in chapter 5.???
d_elem [cm] the total length of the magnet, this length includes entrance
and exit fringe fields $L=2d+r_0\theta$ ($d \geq 3g$)
rbend [cm] r_0 is the bending radius
theta [deg] θ is the bending angle ($\theta > 0$ bend to the right, $\theta < 0$ bend to the
left)
airgap [cm] g is the gap width for field calculations
width [cm] w is the horizontal aperture of dipole
bet1 [deg] β_1 is the entrance edge angle
bet2 [deg] β_2 is the exit edge angle
r1_inv [1/cm] $1/R_1$ is the curvature of entrance pole face
r2_inv [1/cm] $1/R_2$ is the curvature of exit pole face
nstep number of integration steps for the device.

Note:

It is recommended to read Appendix A5 for more information.
Additional dipole parameters are δ_1 ($=0.9$ by default) and δ_2 ($=3$ by default).
The first parameter is defined a vertical vacuum chamber dimension,
see Fig. A5-3, the parameter δ_2 defines the fringe fields extensions
 $z_{11} = z_{21} = -\delta_2 g$ inside the dipole, see Fig. A5-4. TRACK terminates task for
the intersecting entrance and exit fringe fields. User can change this
parameters.

The command line **1 prmtr delta1=r** assigns r to δ_1 .

The command line **1 prmtr delta2=r** assigns r to δ_2 .

The command line **1 prmtr dev_field=** calculates dipole
parameters p , B_f , k_1 , x_{cr1} , x_{cr2} , z_{11} , z_{12} , x_{11} , x_{12} , z_{21} , z_{22} , x_{21} , x_{22} and
extracts them into the file **log.dat**. 1D table containing a field falloff
 $B_y(0,0,z)$, see Fig. A5-5A is also extracted in the file **log.dat**

The command **2 prmtr dev_field=** extracts 2D tables containing the entrance and
exit fringe field $B_y(x,0,z)$, see Fig. A5-4, into the file **dev_field.dat**.

Examples:

1) Enge coefficients are internal $c_0=-0.003183, c_1=2.32, c_2 \div c_5=0$.
0 **bmag** 108.432 40. 90. 7.6 10. 31. 31. 0.01 -0.01 200

2) Enge coefficients are user defined.

1 **enge bmag** 0.1756 2.4415 -0.0555 0.4293 -0.4706 0.1088

```
2 enge bmag 0.1756 2.4415 -0.0555 0.4293 -0.4706 0.1088
1 prmtr delta1=0.8
1 prmtr delta2=2.5
1 prmtr dev_field=
0 bmag 108.432 40. 90. 7.6 10. 31. 31. 0.01 -0.01 200
```

DIPOLE MAGNET, ANY POLE FACE

This element supports any shape of the poles (not only rounded one) and magnetic field with arbitrary field gradient. N is the field index. This element is more generalized bending magnet and supports all 5 types of dipoles defined in the code RAYTRACE [A5-4].

n	dipo	d_elem	rbend	theta	airgap	width	bet1	bet2	N	V	Γ	Δ	mtype	nstep
---	------	--------	-------	-------	--------	-------	------	------	---	---	----------	----------	-------	-------

<i>n</i>	extension number in the files eh_EMS.#n
	for 2D table fringe field of the dipole
<i>d_elem</i> [cm]	L-total length of the dipole, this length includes the entrance and exit fringe fields , $L=2d+\rho_0\theta$ ($d \geq 3g$)
<i>rbend</i> [cm]	ρ_0 -bending radius
<i>theta</i> [deg]	θ -bending angle ($\theta > 0$ right bend , $\theta < 0$ left bend)
<i>airgap</i> [cm]	g-gap width for field calculations
<i>width</i> [cm]	w-horizontal aperture of dipole
<i>bet1</i> [deg]	β_1 -entrance edge angle
<i>bet2</i> [deg]	β_2 exit edge angle

N

B N , B , Γ , and Δ are coefficient of the Taylor expansion of the
G magnetic field in a "uniform" field region, see Eq. (A5-4)

△

mtype type of the dipole magnet
mtype=1,2 homogenous field dipole
mtype=3 dipole with field gradient

`nstep` number of integration steps for the device

Note:

The dipole parameters are defined in Appendix 5. The parameters xcr1, xcr2 are output parameters. The parameters xmax, zmin, zmax are included in **eh_2d.#n**. The code "TRANSPORT" uses parameter k1, "GIOS" uses the function E(z), and "RAYTRACE" uses 2D tables with $B_x(x, 0, z)$ for fringe field description.

Subroutine `dipo` calculates k_1 , $E(z)$, and 2D tables with $B_x(x, 0, z)$. RAYTRACE can calculate 2D fringe field tables for different types of dipole magnets. These tables can be used for the element `dipo` after some modification.

Examples:

2D tables for fringe fields are in the file **eh_2d.#34**

$N = 0.5$, $B = 0$, $\Gamma = 0$, $A = 0$

MAGNETIC MULTIPOLE WITH FRINGE FIELDS

Internal or user defined set of Enge coefficients c0÷c5 can be used for the fringe field calculation. User can define Enge coefficient using the command lines **1 enge mult c0 ..c5** and **2 enge mult c0 ..c5**. The number 1 (2) in the first position defines coefficients for the entrance (exit) edge.

n mult d_elem Heff* Bq Bh** Bo** rap i_quad nstep**

n
d_elem[cm] total length of the multipole
Heff [cm] effective length of the multipole
Bq [G] quadrupole field component at r=Ra
Bh [G] hexapole field component at r=Ra
Bo [G] octupole component at r=Ra
Ra [cm] aperture radius
i_quad 0 - full lens, 1 - first half of lens, 2 -second half of lens
Note: when the element is divided to two halves, the total length of the element is equal to the length of two halves. However, the effective length is the same independently either it is full multipole or the half of the multipole.
nstep number of integration steps for the device.

Note.

- * The effective length `heff` and Enge coefficients are the same for all multipole components. A device `m6` for more precise description of the edge fields of the magnetic multipole is being developed.
- ** `Bq`, `Bh`, and `Bo` are some values of the field defined as $G = Bq/Ra$, $H = Bh/Ra^2$, and $O = Bo/Ra^3$ where G , H , and O are coefficients of multipole field expansion, see Appendix XXX (being developed).
`Bq > 0` provides focusing in the horizontal plane.

Examples:

1) Full lens. Enge coefficients are internal.

1 mult 115. 50. 4834. -350.0 0. 5. 0 200

2) The lens is divided into two halves. Enge parameters of each half are internal.

1 mult 57.5 50. 4834. -350.0 0. 5. 1 100
1 mult 57.5 50. 4834. -350.0 0. 5. 2 100

3) Enge coefficients are user defined. In this example we use the Enge coefficients which are equal to the internal coefficients.

1 enge mult -0.00004, 4.518219, 0, 0, 0
2 enge mult -0.00004, 4.518219, 0, 0, 0

1 mult 115. 50. 4834. -350.0 0. 5. 0 200

All these examples provide the same beam transformation.

MAGNETIC QUADRUPOLE WITH FRINGE FIELDS

HARD EDGE MAGNETIC QUADRUPOLE

Internal or user defined set of Enge coefficients `c0..c5` can be used for the fringe field calculation. User can define Enge coefficient using the commands `1 enge quad c0 ..c5` and `2 enge quad c0 ..c5`

The quadrupole with `d_elem=Heff` or `nstep=0` is treated as a hard edge quadrupole.

n quad Bq* d_elem Heff rap any nstep*****

n
Bq [G] quadrupole component at r=Ra
d_elem[cm] total length of the quadrupole
Heff [cm] effective length of the quadrupole

```

Ra      [cm]  aperture radius
any                placeholder, arbitrary number
nstep               number of integration steps for the device

```

Note.

- * B_q is an artificial number and defines $G = B_q/R_a$, where G is a coefficient of the quadrupole field expansion, see Appendix XXX (being developed).
 $B_q > 0$ provides focusing in a horizontal plane
- ** Sorry
- *** `nstep` is placeholder for the hard-edge quad

Example:

- 1) The quad with the fringe fields. Enge coefficients are internal.
`1 quad -1126.0 40. 25. 3. 0 200`
- 2) The quad with the fringe fields. Enge coefficients are user defined.
In this example we use the Enge coefficients which are equal to the internal coefficients.
`1 enge quad -0.00004, 4.518219, 0, 0, 0`
`2 enge quad -0.00004, 4.518219, 0, 0, 0`

`1 quad -1126.0 40. 25. 3. 0 200`
- 3) The hard edge quad (`nstep=0`).
`1 quad -1126.0 40. 25. 3. 0 0`
- 4) The hard edge quad (`d_elem=Heff`).
`1 drift 8.5 3. 3.`
`1 quad -1126.0 25. 25. 3. 0 200`
`1 drift 8.5 3. 3.`

The beam transformation is the same for examples 3) and 4).
The "soft-edge" quad in the example 1) or 2) is treated as "hard-edge" quad in the example 3) by applying `nstep=0`. It's a recommended way of switching between "soft-edge" and "hard-edge" quads.

3D-FIELD-MAP MAGNETOSTATIC QUADRUPOLE

The quad field is calculated by some external electromagnetic code.
3D table format is described in chapter 5.?

`n mq3d d_elem Bq * rap nstep`

```

n>0           extension number of the files eh_EMS.#n
Bq            [G]  quadrupole component at r=Ra
d_elem        [cm] total length of the quad
Ra            [cm] aperture radius
nstep         number of the integration steps for the device

```

Note.

- * B_q is an artificial number and defines $G = B_q/R_a$, where G is a coefficient of the quadrupole field expansion, see Appendix ??.
Positive field $B_q > 0$ implies focusing in a horizontal plane.
The field table **eh_EMS.#n** is normalized by **TRANSFORM.exe** so that $G=1$ in the quad center. **TRANSFORM.exe** evaluates the quad effective length **Heff** and Enge coefficients for the field table **eh_EMS.#n**

(see chapter 4 for further explanation).

Example:

```
1) field is given in eh_EMS.#04
   4  mq3d    6000.0  13.61   2.     40
2) field is given in eh_EMS.#67
   67  mq3d    6000.0  13.61   2.    140
```

6.3 Electrostatic devices

3D-FIELD-MAP ELECTROSTATIC QUADRUPOLE

n eq3d d_elem Vf rap nstep

```
n           extension number in the files eh_EMS.#n
d_elem [cm] total length of the quad
Vf [V]      inter-electrode voltage*
Ra [cm]      aperture radius
nstep       number of integration steps for the device
```

* The field table **eh_EMS.#n** is normalized by **MWSread.exe** so that the inter electrode voltage is equal to 1 (see chapter 4 for explanation). **Vf** is the inter electrode voltage of the quad.

Note.

- 1) **Vf>0** provide focusing in a horizontal plane.
- 2) Field gradient in the center of the 3D electrostatic quad is $G=k^*Vf/Ra^{**2}$, $k\neq 1$. **MWSread.exe** evaluates k and the quad effective length **Heff** and extracts this two parameters in the file **output.dat**. For linear calculations the device **eq3d** can be replaced by the **equad** with rhe aperture radius **Ra**, effective length **Heff**, and inter electrode voltage **Vf* = k*Vf**.

Examples:

```
4  eq3d    6000.0  13.61   2.     40 , field is given in eh_EMS.#04
67 eq3d   -1000.0  13.61   2.    140 , field is given in eh_EMS.#67
```

3D-FIELD-MAP THREE-ELECTRODE EINZEL LENS

n einz Vf d_elem rap nstep

```
n           extension number in the files eh_EMS.#n
d_elem [cm] total length of the quad
Vf [V]      voltage of the central electrode*, field level
Ra [cm]      aperture radius
nstep       number of integration steps for the device
```

*The field table **eh_EMS.#n** is normalized by **MWSread.exe** so that the voltage of the outer electrodes is equal to 0 and the voltage of the central electrode is equal to 1, see chapter 4 of this manual for the explanations. **Vf** is the central electrode voltage of the einzel lens.

Example:

```
9  einz    6000.0   13.61   2.     40    , field is given in eh_EMS.#09
97 einz   -1000.0   13.61   2.    140    , field is given in eh_EMS.#97
```

ELECTROSTATIC QUADRUPOLE WITH FRINGE FIELDS**n quad Vf d_elem Heff rap nstep**

n n=0 internal set of Enge coefficients c0÷c5 is used
 n=1 the Enge coefficients c0÷c5 are defined by the latest
 element 1 enge equad c0 ..c5 placed upstream the line describing
 equad in the sclinac.dat file.

Vf [V] inter-electrode voltage
d_elem[cm] actual length of the quadrupole
Heff [cm] effective length of the quadrupole
Ra [cm] aperture radius
nstep number of integration steps for the device**

Note.

- 1) **vf** >0 provides focusing in a horizontal plane
- 2) Field gradient in the center of the quad is $G=Vf/Ra^{**2}$
- 3) Hard edge electrostatic quadrupole is not applicapable in **TRACK**.

Example:

- 1) Enge coefficients are internal.
`0 equad 5126.0 40. 25. 3. 0 200`
- 2) Enge coefficients must be defined. In this example we use the Enge coefficients which are equal to internal ones.
`1 enge equad -0.00004, 4.518219, 0, 0, 0`

`1 equad -1126.0 40. 25. 3. 0 200`

Beam transformation is the same in the examples 1) and 2).

HV PLATFORM ENTRANCE&EXIT**Element name: 'deck'**

ntype = el.n ! Type of the lens, determines the file with input field distribution.
Vdeck = el.p1 ! Voltage with respect to the ground [kV]
 Positive voltage accelerates positively charged ions
d_elem = el.p2 !total length of the electric field distribution at the edge of the HV deck [cm]
rap = el.p3 ! aperture of the beam pipe [cm]
mesh_Z = el.n1 ! number of meshpoints along z
mesh_R = el.n2 ! number of meshpoints along r

Note.

- 1) This element is being reconstructed. Use the device 'extrc' instead of the 'deck'.

Example:

```

2 deck    -35.   20.  1.5  201    16
1 extrc   2000. (voltage,V)  78.(length,cm)  2. (aperture, cm)      100 (number
of integration steps)

```

6.4 Drift space and other elements of the beamline

DRIFT

n drift d_elem rapx rapy nstep

n -any integer number
d_elem [cm]-length of the drift space
Rx [cm] -horizontal half aperture (X direction)
Ry [cm] -vertical half aperture (Y direction)
nstep -(optional, ≥ 2) number of steps for integration along the -drift space. If this parameter is omitted,
nstep =max{2, d_elem/(0.25 $\beta\lambda$)}, where $\lambda=c/(\text{harm}*\text{freqb})$ is the current RF wave length and βc - is the velocity of the reference particle.

Note.

- 1) $Ry > 0$:the aperture of the drift is rectangular $\{-|Rx| \leq x \leq |Rx|, -Ry \leq y \leq Ry\}$.
- 2) $Ry < 0$:the aperture of the drift is elliptical $(x/Rx)^2 + (y/Ry)^2 < 1$
- 3) The drift space with the length close to zero can be used as a slit in **sclinac.dat** input.

Example:

- 1) 1 **drift** 9.0 3. 3. **nstep** is defined internally, the aperture is rectangular
- 2) 1 **drift** 9.0 3. 3. 45 **nstep=45**, the aperture is rectangular
- 3) 0 **drift** 9.0 3. -3. 245 **nstep=245**, the aperture is elliptical

SLIT

n slit rapx rapy

n = 1 defines element 'slit'
n = 0 defines element 'pseudo slit'
n = -1 no action
Rx [cm] half aperture in x direction
Ry [cm] half aperture in y direction

- 1) $Ry > 0$:the aperture of the slit is rectangular $\{-|Rx| \leq x \leq |Rx|, -Ry \leq y \leq Ry\}$.
- 2) $Ry < 0$:the aperture of the slit is elliptical: $(x/Rx)^2 + (y/Ry)^2 < 1$
- 3) The **slit** element has a zero length along z-coordinate.

Note:

Particle is marked as a 'lost' particle if it is outside the **slit** aperture. The 'lost' particle is not traced further. The coordinates of the 'lost' particle on the slit will be saved. All particles lost in 'pseudo slit' and upstream of the 'pseudo slit' will be removed from the array of particle coordinates. The latter is redefined and will consist of less or equal number of particles than upstream of the element 'pseudo slit'.

STRIPPER

n strip xWtarg dWtarg dxtarg dds

n number of the stripper film or foil and extension number of the file **stripper.#0n**
xWtarg energy loss (keV/u)
dWtarg RMS energy straggling in keV/u, rms value calculated by the external code, in **TRACK** we assume the distribution is Gaussian.
dxtarg RMS scattering in X' and Y', in radians.
dds fluctuation of the target thickness (relative unit), peak-to-peak amplitude. The number of charge states and the percentage of particle distribution in different charge states, as well as average beam energy after the stripper must be given in file **stripper.#0n**.

Example:

```
1 strip 10. 4.1 0.0002 0.05
```

*) Currently this element is suitable only for uranium at energies ~10-15 MeV/u at the first stripper and at ~85-90 MeV/u at the second stripper. The stripper subroutines in **TRACK** are based on SRIM simulations. The passage through the stripper was initially simulated using the SRIM code and the ion beam distribution has been parameterized. Detailed description of the procedure will follow. Generalized particle distribution after the passage of the stripper is being developed by B. Mustapha.

SHIFT-ROTATE

Element name: 'shrt'

ntype = el.n = 0 no meaning
fiz [deg] = el.p1 angle of rotation (positive for the counterclockwise rotation)
delx [cm] = el.p2 displacement in X-direction
dely [cm] = el.p3 displacement in Y-direction

Example:

```
1 shrt 45. 0.1 0.1
```

UDS

n uds d_elem Vf rap dz_elem nstep z1 z2 z3 z4 z5

User defined structure. This element can be used to define a gap inside the multi-gap resonator.

n extension number of the file **eh_EMS.###**. The field file defines 3D distribution of the electric field in the accelerating gap (no magnetic field is assumed). This field can be generated by an external electrostatic code.
d_elem[cm] total length of the user defined device
Vf[V] gap voltage
rap [cm] radius of aperture

```

dz_elem[cm] shortening of the second drift tube length to fit the reference
           particle phase in the following rf gap if such a gap exists.
nstep      number of the integration steps for the device
z1 z2 z3 z4 z5 (optional)   z-coordinates for extracting the reference
                           particle's (r.p.) phase along the element. Example: if one needs to
                           know phase of the r.p. one of z-coordinates should be equal to
                           d_elem/2. The phases of the r.p. will be extracted into the file
                           matrix.dat.

```

PRMTR

This element can be used to define a drift tube resonator containing many accelerating gaps. In addition, **prmtr** can be used to define or re-define any integer flag from the list given as input parameters in the file **track.dat**.

0 prmtr iflag_chain=phasa_in [deg]

```

n uds
n drift
n uds
n drift
.....

```

0 prmtr iflag_chain=0

Examples:

If **iflag_log=1**, it will extract major information about the element following the line "**prmtr**" into the file **log.dat**.
If **isol=0,1** it will introduce solenoid with out fringe fileds (**isol=0**) and with fring fields (requires **solenoid.dat** file).
If **current=0** it will introduce zero current and replaces the 'current' definition in the **track.dat** file.

SCRATCH

Element name: 'scrch'

Example:

```

-1 scrch
 1 scrch
-1 : the code writes all necessary information to temporary file scratch.#01.
+1 : the code reads all data and starts simulation from the intermediate
location along z where file scratch.#01 has been created.

```

STOP

n Stop

```

n          any
stop stops simulation at this given location along the z.

```

Example:

```

0 stop

```

SWTCH

```
n  swtch  coeff
n      -number of correction sections
coeff  -This element is used to change the field level in focusing elements
during the correction procedures. swtch acts on focusing devices (solenoids and
quadrupoles) located immediately after swtch
```

Example:

```
2 swtch  coeff
1 sol    B0 d Heff rap rap
```

UPDATE

When **0 updat** structure line appears in **scilinac.dat** the beam matrix for all charge states is calculated and then the transfer matrices of each device between **0 updat** and **1 updat** are calculated, both for reference particle with charge state Qdesign and mass Adesign and for each ion species with **Q(iq)**, **A(iq)**, where iq=1,nqtot. For charge state Qdesign and mass Adesign a cumulative matrix of all devices from **0 updat** to exit of this device is calculated. Beam matrix after each element is calculated as matrix production and from ray trace calculations.

Matrix of devices and cumulative matrix from **0 updat** to exit of this device and two kinds of beam matrices are extracted into the file **matrix.dat**.

It is possible to have any number of non-intersecting **0 updat ... 1 updat** blocs in the **scilinac.dat**. The matrix calculations do not include beam space charge yet(it is being developed).

FILTER

This element is used to filter unaccelerated particles after the RFQ by setting dw_w=0.04. (relative energy deviation from the reference particle energy). This mode is used for matching purposes. If dw_w=1.0 all particles are kept and the un-accelerated particles will be lost either on the aperture walls or on the slits.

Element name 'FILTR'

```
n  filtr  dw_w  QdesignNew  Index
n      =1  element is activated, if n=-1, element is de-activated
dw_w   energy cut-off boundary
QdesignNew - Charge of the reference particle after the filter, should be equal
            to the charge state of one of the charge states upstream of the
            element 'filtr'
index    If index=0, then no effect on bunch structure; if index=1, two
            different charge states are combined into one bucket artificially
            (space charge effect is ignored, current =0).
```

Example:

```
0  filtr  0.05  28  0.
```

6.5 Misalignments, errors and beam steering correction

ALIGN

```

n   align name  $\Delta_{xy}$   $\Delta_z$   $\phi_z$   $\Delta\Phi_j$   $\Delta F_j$   $\Delta\Phi_s$   $\Delta F_s$ 
n       = nt_cav - the SRF cavity type (name=cav)
n       = 1   for all other devices
n       ==-1  no action
name    the device name in sclinac.dat
 $\Delta_{xy}$  [cm] (=0 by default) the amplitude of the element displacement in x
                and y direction w.r.t. the entrance and exit coordinate systems
                of the device **.
 $\Delta_z$  [cm]   (=0 by default) the amplitude of the element displacement in z
                direction w.r.t. the entrance and exit coordinate systems of the
                element
 $\phi_z$  [mrad]  (=0 by default) the amplitude of the element rotation about
                z-axe of the entrance and exit coordinate system of the
                element.
 $\Delta\Phi_j$  [deg] (=0 by default) the rms fast error of the phase**.
 $\Delta F_j$  [%]   (=0 by default) the rms fast error of the field amplitude***.
 $\Delta\Phi_s$  [deg] (=0 by default) the amplitude of the element static phase error
 $\Delta F_s$  [%]   (=0 by default) the amplitude of the element static field
                error

```

Note:

1) The previous version of the code, TRACKv34, defined the misalignments of the solenoids and SRF cavities in the input file **track.dat**. **TRACKv35** define the misalignments all of the devices by the command **align** in the input file **sclinac.dat**

2) The line in the **sclinac.dat** file **1 align name..** defines the position of misalignments, phase and field errors for all devices, except cavities, with the given **name** located downstream of this line. The line **nt_cav align cav ..** defines the errors only for cavities of the type **nt_cav** i.e. only for devices **nt_cav cav ..** located downstream of this command.

3) The errors and misalignments of the device are described by 10 randomly generated numbers $R_1 \div R_{10}$. Particularly, $R_1 \div R_6$ defines the spatial misalignments of the device, see Appendix A1 for the explanations.

$x_{in} = \Delta_{xy} * R_1$ is the device displacement along x axis of the entrance coordinate system.

$y_{in} = \Delta_{xy} * R_2$ is the device displacement along y axis of the entrance coordinate system.

$z_{in} = \Delta_{xy} * R_3$ is the device displacement along z axis of the entrance coordinate system.

$\phi_{xin} = \phi_{xy} * R_4$ is the device rotation about x axis of the entrance coordinate system.

$\phi_{yin} = \phi_{xy} * R_5$ is the device rotation about y axis of the entrance coordinate system.

$\phi_{zin} = \phi_z * R_6$ is the device rotation about z axis of the entrance coordinate system.

Phase and amplitude fast errors are described by Gaussian distribution with standard deviation $\sigma=1.0$ and truncated at the 3σ -level. R_7, R_8 , $|R_{7,8}| \leq 3$. The fast errors are known as jitter or dynamic errors.

$\Delta\Phi_j * R_7$ is the fast phase error of the device

$\Delta F_j * R_8$ is the fast field error of the device

R_9 and R_{10} are uniformly distributed in interval [-1,1] and defined to be the static errors of the phase and amplitude of the rf field.

$\Delta\Phi_s * R_9$ is the static phase error of the device.

$\Delta F_s * R_{10}$ is the static field error of the device.

- 4) The line `1 prmtr msln_dstr=N` followed by a device evaluates N seeds for spatial misalignment of this device. The device displacements and rotation w.r.t. entrance coordinate system are extracted into the file `r_in.dat`. The file `r_in.dat` contains the title: name Xin[cm] Yin[cm] Zin[cm] RXin[mrad] RYin[mrad] RZin[mrad] and N line with the device coordinate w.r.t. entrance coordinate system.

The device displacement and rotation w.r.t. exit coordinate system is extracted in a file `r_out.dat`. The file `r_out.dat` has the heading:

name Xout[cm] Yout[cm] Zout[cm] RXout[mrad] RYout[mrad] RZout[mrad] and N line with the device coordinate w.r.t. exit coordinate system.

If the line `1 prmtr msln_dstr=N` appears in the `sclinac.dat` several times, only the last line will be executed.

- 5) In each accelerator seed 10 numbers $R_1 \div R_{10}$ are generated for each element defined in the file `sclinac.dat`. These data is extracted into the file `VRAND_nnn.dat`, where nnn is the number of seed.

Examples:

1) `1 align mult 0.03 0. 0. 0. 0.`

The misalignment is assigned or re-assigned to all multipoles (`n mult..`) located downstream of this line.

2) `23 align cav 0.03 0. 0. 0.1 0.1 1. 1.`

The misalignment is assigned or re-assigned to all cavities of the type 23 (`23 cav ..`) located downstream of this line.

The field in the cavity of type 23 is defined by the file `eh_MWS.#23`.

ARTIFICIAL STEERING CORRECTOR

n zero pos ang

`n` = 1 the "zero" corrector is activated

`=-1` no action (the "zero" corrector is not activated.)

`pos[cm]` error of beam center position after correction in the horizontal and vertical planes

`ang[mrad]` error of beam centroid angle after correction in the horizontal and vertical planes

Note:

This element assigns zero to the center of gravity (with some residual errors of positioning) of the beam in xx' and yy' planes. The new beam center positions Xc and Yc, and the beam center angles X'c and Y'c will be uniformly distributed in intervals
 $-pos \leq Xc \leq pos$, $-pos \leq Yc \leq pos$, $-ang \leq X'c \leq ang$, $-ang \leq Y'c \leq ang$.

Example:

```
1 zero 0.01 0.1
```

THIN-DIPOLE STEERING CORRECTOR

```

n CORR L FH FV rapC nstep FHKick FVkick
n =1 corrector is activated
n =-1 no action (corrector is not activated )
L[cm] (=0 by default), "thin"corrector has zero length
FH[mrad] -maximum horizontal corrector strength for transverse
           correction procedure ( FHKick>0)
           -actual corrector strength if FHKick=0. In this case it acts as an
           externally defined corrector with the kick FH (manual corrector).

FV[mrad] -maximum vertical corrector strength for transverse
           correction procedure ( FHKick>0)
           -actual corrector strength if FHKick=0. In this case it acts as an
           externally defined corrector with the kick FV (manual corrector).

RapC[cm] -(optional)aperture of the corrector, it is not applied to "thin"
           corrector
nstep -(optional) number of integration steps, it is not applied to "thin"
           corrector
FHKick[mrad] -vertical corrector strength for transfer function calculation
FVkick[mrad] -horizontal corrector strength for transfer function calculation

```

Note:

This element assigns thin-dipole corrections to the center of gravity of the beam in xx' and yy' planes. Strengths are determined by running the correction module of the code or are defined manually.

Details are given in Appendix 2.

Examples:

```

1) 0 corr 0.    2.   3.          - manual corrector
2) 0 corr 0.    2.   3.   0.  0. 1.  1.  - "calculated" correction strengths
3) 0 corr 0.    2.   3.   0.  0. 0.  0.  - manual corrector
4) 0 corr 0.    2.   3.   0.  0. 0.  1.  - manual horizontal corrector and the
                                             "calculated" correction strength in
                                             vertical plane

```

DIPOLE CORRECTOR

```

n CORR L FH FB rapC nstep FHKick FVkick
n>1          -number of the extension of the file with name eh_2d.#n
               for 2D table with  $B_y(x, 0, z)$  and  $B_z(0, y, z)$  of the corrector
L[cm]         -length of the corrector
FH=BYmax[G]  -maximum vertical field in the center of the corrector for
               transverse correction procedure ( FHKick>0)

```

```

-actual corrector strength (FHkick=0). In this case the corrector is
"manual".
    Vertical field bends beam in the horizontal plane.

FV=BXmax[G] -maximum horizontal field in the center of the corrector for
    transverse correction procedure (FVkick>0).
    -actual corrector strength (FVkick=0). In this case the corrector
is "manual".
    Horizontal field bends beam in the vertical plane

RapC [cm] - the corrector aperture
nstep -number of integration steps
FHkick[G] -vertical field in the center of the corrector for transfer
    function calculation
FVkick[G] -horizontal field in center of the corrector for transfer
    function calculation

```

Note:

This realistic dipole corrector subroutine uses 2D-field distribution in XZ and YZ planes.

DIPOLE CORRECTOR COMBINED WITH SOLENOID

```

n1 sol3D Bf d_elem rap nstep
& n CORR L FH FV rapC nstep FHkick FVkick
n>1 -extension number in the files eh_2d.#n
      for 2D table with  $B_y(x, 0, z)$  and  $B_z(0, y, z)$  with field of
      the corrector
L[cm] -length of the corrector*

FH=BYmax[G] -vertical field, bends beam in the horizontal plane
    -maximum vertical field in the center of the corrector for
        transverce correction procedure (FHkick>0)
    -actual corrector strength (FHkick=0). The corrector is
        "hmanual".
FV=BXmax[G] horizontal field, bends beam in the vertical plane
    -maximum horizontal field in the center of the corrector for
        transverce correction procedure (FVkick>0)
    -actual corrector strength (FVkick=0). The corrector is
        "manual".
RapC [cm] - the corrector aperture*
nstep -number of integration steps*
FHkick[G] -vertical field in center of the corrector for transfer
    function calculation
FVkick[G] -horizontal field in center of the corrector for transfer
    function calculation

```

* The length, aperture and number of integration steps of the combined corrector are defined in the line for the upstream solenoid.

Note:

- 1) This modification of the corrector subroutine is used for combined solenoid and dipole fields.
- 2) The center of the corrector coincides with the center of the solenoid which is defined in preceding line, see Fig???.

BEAM POSITION MONITOR

n MON L dx dy Wx Wy

n	-any integer, no meaning in this version of the TRACK
L [cm]	(=0, by default, optional)
dx[cm]	-accuracy of the BPMX
dy[cm]	-accuracy of the BPMY
Wx[cm]	- weight in the correction equation for the BPMX
Wy[cm]	- weight in the correction equation for the BPMY

Note:

- 1) This line defines two Beam Position Monitors (BPM) simultaneously:
BPMX measures the x-coordinate of the beam center and BPMY measures the y-coordinate of the beam center.
 - 2) L#0 defines a virtual monitor to measure the beam center angle. Accuracy of the angle measurement is $d\phi_x$ and $d\phi_y$, which are

$$d\varphi_x = \sqrt{2} dx/L, d\varphi_y = \sqrt{2} dy/L$$

CORRECTION SECTION MARKER

n CDUMP
n >0 number of the correction sections.

Note:

In TRACKv35 `cdump` is used like brackets.

The correction section number n is defined by two or three `n cdump` elements. The code uses correctors and monitors between 1st `n cdump` and 2nd `n cdump` when the correction section is marked by two `n cdump` elements. TRACK uses correctors between 1st `n cdump` and 2nd `n cdump` and the monitors between 1st `n cdump` and 3rd `n cdump` when the correction section is marked by three elements `cdump`. See Fig.???

Example:

Δ1(1*-- 1st section--->-->2nd section-->-->3rd sect.-->-->4st sect.-->----- 5th section----->

***** $(_1\Delta**=\Delta_1)$ ***** $(_2\Delta**=_1)$ **= $\Delta*_2$) *****
 <----- 1st correction section ----->
 <----- 2nd correction section ----->

(this figure needs some modifications)

Beam passes through the first section. The strengths of the correctors between $(\textcolor{red}{1} \quad \textcolor{red}{1})$ are defined in the first correction section using the BPM inside the brackets $(\textcolor{red}{1} \quad \textcolor{red}{1}) \quad \textcolor{red}{1}$

7. Launching the code TRACK

Collect all necessary files in one directory. Click TRACKv37.EXE, hit the letter 'g' and press ENTER. For a file manager & editor, we use a very convenient software "Windows Commander" v.4.52 purchased from Christian Ghisler (Switzerland).

TRACKv37.EXE simulates beam dynamics with the graphical windows. On running the code several output files are produced.

8. TRACK Output data

Beam parameters after each element of **sclinac.dat** are extracted into file **out.dat**. The first line in the file **out.dat** is the title of various beam parameters. The second line contains beam parameters w.r.t. Cartesian coordinate system (CCS) \vec{E}_{first} which is the entrance CCS of the accelerator. The subsequent lines contain beam parameters w.r.t. the device exit coordinate system CCS \vec{E}_{out} . Definitions of different CCS along the beam path are shown in Fig. 1 below.

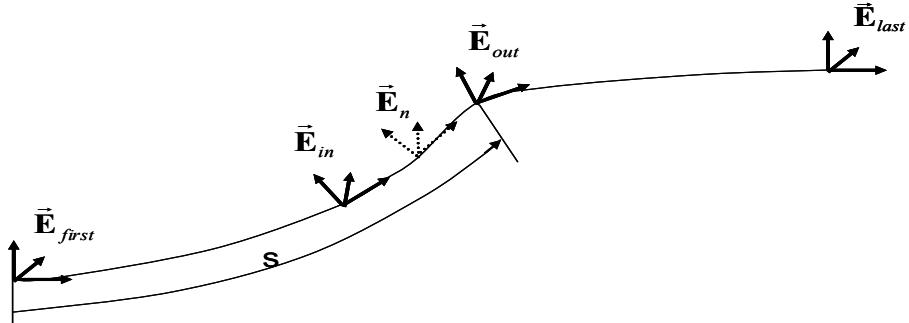


Figure 1. Trajectory of the reference particle.

For each device TRACK defines entrance CCS \vec{E}_{in} , exit CCS \vec{E}_{out} and **nstep**+1 intermediate CCS $\vec{E}_n, n=0, nstep$; $\vec{E}_0 = \vec{E}_{in}$, $\vec{E}_{nstep} = \vec{E}_{out}$. Three numbers (x, y, s) , where s is a distance along the reference particle path from accelerator entrance to origin of any CCS \vec{E} , x and y are given w.r.t. this CCS \vec{E} , generate a curvilinear beam optical coordinate of the particle.

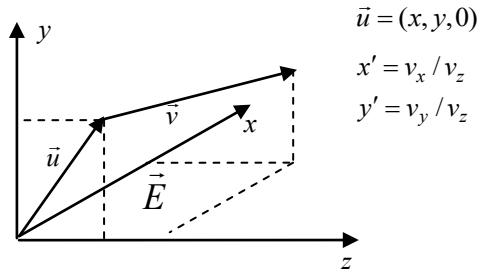


Figure 2. Cartesian coordinate system, \vec{v} is the velocity of the reference particle.

The column names in the file **out.dat** are:

<code>n_el</code>	- sequence number of the element in sclinac.dat
<code>name</code>	- name of the element(<code>cav</code> , <code>align ..</code>)
<code>dist[m]</code>	- distance along the accelerator beam optic axis from the beginning of the entrance to the exit of the given element
<code>Energy[MeV/u]</code>	- average kinetic energy of the beam
<code>x_rms [cm]</code>	$-x_{rms} = \sqrt{(x - \bar{x})^2}$, x rms envelope of beam
<code>y_rms [cm]</code>	$-y_{rms} = \sqrt{(y - \bar{y})^2}$, y rms envelope of beam
<code>Xmax [cm]</code>	$-x_{max}$ envelope of beam (100% of particles)
<code>Ymax [cm]</code>	$-y_{max}$ envelope of beam (100% of particles)
<code>phi_rms [deg]</code>	$-\varphi_{rms} = \sqrt{(\varphi - \bar{\varphi})^2}$ rms phase envelope of beam
<code>phi_max [deg]</code>	$-\varphi_{max}$ phase envelope of beam (100% of particles)
<code>DW/W[rel.u.]</code>	$-(dW/W)_{max}$ energy envelope of beam (100% of particles)
<code>4*exn_rms [cm*mrad]</code>	$-4\varepsilon_x$, where $\varepsilon_x = \beta\gamma\sqrt{(x^2 x'^2 - \bar{x}\bar{x}')^2}$ is normalized rms emittance in the (xx') phase plane
<code>ex##.# [cm*mrad]</code>	$\varepsilon_x(\#.\#%)$ normalized emittance in (xx') phase plane containing $\#.\#[\%]$ of the beam particles. $\#.\#$ is defined by variable <code>part</code> in the file track.dat . $\#.\#=100*\text{part}$, for example if <code>part=0.995</code> , then $\#.\#=99.5$)
<code>exn_max [cm*mrad]</code>	$\varepsilon_x(100\%)$ normalized emittance in (xx') phase plane containing 100% of particles.
<code>4*eyn_rms [cm*mrad]</code>	$-4\varepsilon_y$, where $\varepsilon_y = \beta\gamma\sqrt{(y^2 y'^2 - \bar{y}\bar{y}')^2}$ is normalized rms emittance in the (yy') phase plane
<code>ey##.# [cm*mrad]</code>	$-\varepsilon_y(\#.\#%)$ normalized emittance in (yy') phase plane containing $\#.\#[\%]$ of the beam particles. $\#.\#$ is defined by variable <code>part</code> in the file track.dat .
<code>eyn_max [cm*mrad]</code>	$-\varepsilon_y(100\%)$ normalized emittance in (yy') phase plane containing 100% of the beam particles.
<code>4*ezn_rms [keV/u*ns]</code>	$-4\varepsilon_z$, where $\varepsilon_z = \sqrt{\Delta t^2 (dW/W)^2 - \Delta t (dW/W)^2}$ is normalized rms emittance in longitudinal ($\Delta t=t-t_{RP}$, $\Delta W/W=(W-W_{RP})/W_{RP}$) phase plane. t_{RP} and W_{RP} is time of flight and kinetic energy of the reference particle.
<code>ezn##.# [keV/u*ns]</code>	$-\varepsilon_z(\#.\#%)$ normalized emittance in longitudinal ($\Delta t=t-t_{RP}$, $\Delta W/W=(W-W_{RP})/W_{RP}$) containing $\#.\#[\%]$ of the beam particles. $\#.\#$ is defined by variable <code>part</code> in the file track.dat .
<code>ezn_max [cm*mrad]</code>	$-\varepsilon_y(100\%)$ normalized emittance in longitudinal ($\Delta t=t-t_{RP}$, $\Delta W/W=(W-W_{RP})/W_{RP}$) containing of the beam particles.
<code>XPc [cm]</code>	\bar{x} - spatial beam center position in (xx') phase plane
<code>BXc [mrad]</code>	\bar{x}' - angular beam center position in (xx') phase plane

YPC[cm]	\bar{y} - spatial beam center position in (yy') phase plane
BYC[mrad]	\bar{y}' - angular beam center position in (yy') phase plane
btgm	β_Y , $\beta=v/c$ is velocity of the reference particle, $\gamma=1/\sqrt{1-\beta^2}$
zcn[deg]	$\bar{\varphi}$ phase beam center position in (φ dw/w) phase plane w.r.t reference particle
a_x	α_x -Twiss-parameter of the beam in (xx') phase plane
b_x[cm/mrad]	β_x -Twiss-parameter of the beam in (xx') phase plane
a_y	α_y -Twiss-parameter of the beam in (yy') phase plane
b_y[cm/mrad]	β_y -Twiss-parameter of the beam in (yy') phase plane
a_z	α_z -Twiss-parameter of the beam in (φ dw/w) phase plane
b_z[deg*(%ofD_W/W)]	β_z -Twiss-parameter of the beam in (φ dw/w) phase plane
#of_lost_part	number of the lost particles
total#of_part	total number of the survived particles

File **coord.dat** contains coordinates of particles transmitted to the line **0 stop** in the **sclinac.dat** file. The particle coordinates are calculated w.r.t. the CCS \vec{E}_{last} which is the exit CCS of the accelerator. The first line of this file is column names:

Nseed iq dt[nsec] dW[Mev/u] x[cm] x'[mrad] y[cm] y'[mrad].
Nseed is the seed number. For ideal machine without errors Nseed=0.

iq is the charge state number of ion. The charge and mass of the ion is equal $Q_{iq}=qq(iq)$ and $A_{iq}=amass(iq)$. The arrays **qq** and **amass** are defined in the file **track.dat**.

dt $dt=t-t_{RP}$, where t is the particle time-of-flight and t_{RP} is the reference particle time-of-flight.

dW $dW=W-W_{RP}$, where W is the energy of the particle, and W_{RP} is the energy of the reference particle.

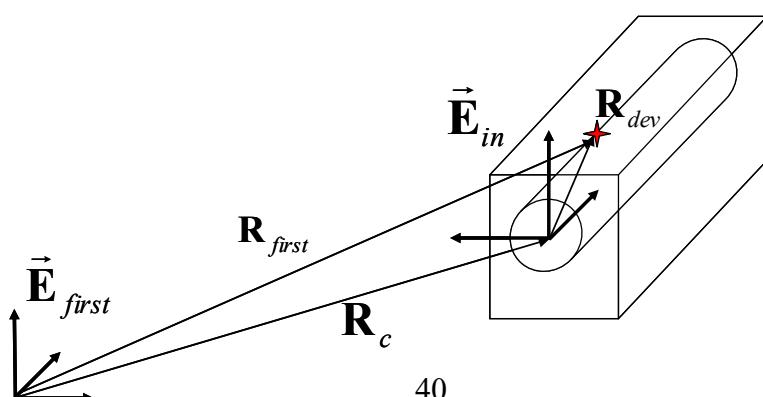
x is the x-coordinate of the particle w.r.t. CS \vec{E}_{last}

x' $x'=v_x/v_z$

y is the y-coordinate of the particle w.r.t. CS \vec{E}_{last}

y' $y'=v_y/v_z$

The file **lost .dat** contains coordinates of the particles which are lost in the accelerator or transport line. The coordinates and velocities of the particles lost in a given device are extracted in the first CCS \vec{E}_{first} of the accelerator and in the entrance CCS \vec{E}_{in} of the device.



The unit vectors of the device CCS $\vec{\mathbf{E}}_{in}$ are defined by the transformation matrix O and the unit vectors of the device CCS $\vec{\mathbf{E}}_{first}$ as

$$\begin{vmatrix} \mathbf{ex}_{dev} \\ \mathbf{ey}_{dev} \\ \mathbf{ez}_{dev} \end{vmatrix} = \begin{vmatrix} O_{11} & O_{12} & O_{13} \\ O_{21} & O_{22} & O_{23} \\ O_{31} & O_{32} & O_{33} \end{vmatrix} \begin{vmatrix} \mathbf{ex}_{first} \\ \mathbf{ey}_{first} \\ \mathbf{ez}_{first} \end{vmatrix} .$$

The transformation rule from the device CCS $\vec{\mathbf{E}}_{in}$ to the CCS $\vec{\mathbf{E}}_{first}$ for the particle coordinate and velocity is

$$R_{first}^T = R_{dev}^T O^{-1} + R_c^T, \quad V_{first}^T = V_{dev}^T O^{-1},$$

where T means transpose.

The column names of the file **lost.dat** are:

n_el	-sequence number of element in the file sclinac.dat
name	-name of the device(cav, align ..)
Q[e]	-the charge of the ion
A	-the mass number of the ion
Xfirst[cm]	-x-coordinate of the particle w.r.t. the CCS $\vec{\mathbf{E}}_{first}$
Yfirst[cm]	-y-coordinate of the particle w.r.t. the CCS $\vec{\mathbf{E}}_{first}$
Zfirst[cm]	-z-coordinate of the particle w.r.t. the CCS $\vec{\mathbf{E}}_{first}$
VXfirst	Vx/c-x-component of the particle velocity w.r.t. the CCS $\vec{\mathbf{E}}_{first}$
VYfirst	Vy/c-y-component of the particle velocity w.r.t. the CCS $\vec{\mathbf{E}}_{first}$
VZfirst	Vz/c- z-component of the particle velocity w.r.t. the CCS $\vec{\mathbf{E}}_{first}$
Xdev[cm]	- x-coordinate of the particle w.r.t. the device CCS $\vec{\mathbf{E}}_{in}$
Ydev[cm]	- y-coordinate of the particle w.r.t. the device CCS $\vec{\mathbf{E}}_{in}$
Zdev[cm]	- z-coordinate of the particle w.r.t. the device CCS $\vec{\mathbf{E}}_{in}$
VXdev	Vx/c- x-component of the particle velocity w.r.t. the CCS $\vec{\mathbf{E}}_{in}$
VYdev	Vy/c- y-component of the particle velocity w.r.t. the CCS $\vec{\mathbf{E}}_{in}$
VZdev	Vz/c- z-component of the particle velocity w.r.t. the CCS $\vec{\mathbf{E}}_{in}$
Xc[cm]	x-coordinate of the origin the CCS $\vec{\mathbf{E}}_{in}$ w.r.t. the CCS $\vec{\mathbf{E}}_{first}$
Yc[cm]	y-coordinate of the origin the CCS $\vec{\mathbf{E}}_{in}$ w.r.t. the CCS $\vec{\mathbf{E}}_{first}$
Zc[cm]	z-coordinate of the origin the CCS $\vec{\mathbf{E}}_{in}$ w.r.t. the CCS $\vec{\mathbf{E}}_{first}$

O(1,1)	O ₁₁	transformation matrix element
O(1,2)	O ₁₂	transformation matrix element
O(1,3)	O ₁₃	transformation matrix element
O(2,1)	O ₂₁	transformation matrix element
O(2,2)	O ₂₂	transformation matrix element
O(2,3)	O ₂₃	transformation matrix element
O(3,1)	O ₃₁	transformation matrix element
O(3,2)	O ₃₂	transformation matrix element
O(3,3)	O ₃₃	transformation matrix element

A flag `iflag_lost` provides two features.

9. Acknowledgements

We appreciate very much the continuous help from B. Mustapha who supports the UNIX version of TRACK. Many our colleagues contributed to the code development. Particularly, the following subroutines were developed with the help of our associates: a)the early version of the code transform.exe - A.A. Kolomiets; b) the 3D Poisson solver - V.A. Moiseev (INR, Moscow-Troitsk), c) the parametrization of the stripper on the base of SRIM code - B. Mustapha (ANL-PHY); d) the original version of automatic steering correction in transverse phase space - E. Lessner (ANL-PHY).

The authors thank M. Sengupta for careful reading of the Manual.

10. References

1. P. N. Ostroumov and K. W. Shepard, Correction of Beam Steering Effects in Low-Velocity Superconducting Quarter- Wave Cavities, Phys. Rev. ST. Accel. Beams **11**, 030101 (2001).
2. V.A. Moiseev and P. N. Ostroumov. High Intensity Beam Dynamics in the Ion Linear Accelerators. Proc. of the 1998 European Part. Accel. Conf., EPAC98, Stockholm, p.1216.
<http://accelconf.web.cern.ch/AccelConf/e98/PAPERS/THP10F.PDF>
3. P.N. Ostroumov, V. N. Aseev, B. Mustapha. Beam Loss Studies in High-Intensity Heavy-Ion Linacs. Phys. Rev. ST. Accel. Beams, Volume 7, 090101 (2004).

Appendix 1. Simulation of element misalignments and field errors

There are three groups of errors:

- 1) Misalignment error is a displacement of the device as a rigid body.
- 2) Field error is the field amplitude of the device.
- 3) Error of the rf field phase.

A rigid body requires six independent coordinates to specify its displacement. Usually a Cartesian set of coordinates (CSC) is fixed in the rigid body and the body motion is defined by three shifts x , y , z of the CSC origin and three angles φ_x , φ_y , φ_z which specify the rotation axes about the initial axes. In TRACK there are two equivalent sets of the coordinates for each device: the entrance CSC and the exit CSC, Figure 1a.

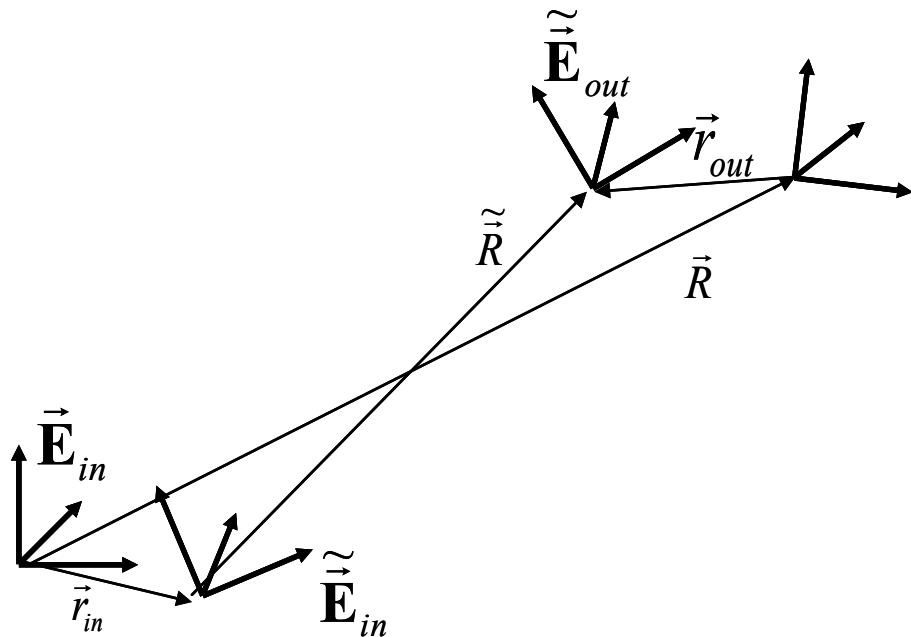


Figure 1a

A special set of the device coordinates is for providing equality of the device entrance and exit CSC's. This set contains the device shifts x_{in} , y_{in} and z_{in} and a rotation angle $\varphi_{z_{in}}$ about the z-direction of the entrance CSC, and the device shifts x_{out} , y_{out} and z_{out} w.r.t. exit SC, and rotation angle $\varphi_{z_{in}}$ about the z-direction of the exit SC. The actual position of the device is uncertain within the given tolerances (Δ_x , Δ_y , Δ_z , φ_z). In TRACK we use the uniform distribution of misalignments and randomly generate x_{in} , y_{in} , ... within the given tolerance amplitudes (Δ_x , Δ_y , Δ_z , φ_z).

A complete position of the misaligned device is determined by its displacements x_{in} , y_{in} , z_{in} and the rotation angles $\varphi_{x_{in}}$, $\varphi_{y_{in}}$, $\varphi_{z_{in}}$ w.r.t. the

entrance CSC. Tolerances φ_x and φ_y can be estimated as $\varphi_x = \varphi_y = 2 \frac{\sqrt{\Lambda_x^2 + \Lambda_y^2 + \Lambda_z^2}}{R}$,

where R is the distance between the origins of the entrance and exit CSC. They are the upper boundaries for φ_x , φ_y (a device with rectilinear optic axis has $\varphi_x = 2\Delta_y/R$, $\varphi_y = 2\Delta_x/R$). The set of values $\{x_{in}, y_{in}, z_{in}, \varphi_{x_{in}}, \varphi_{y_{in}}, \varphi_{z_{in}}\}$ unambiguously determine the device displacements $x_{out}, y_{out}, z_{out}$ and rotation angles $\varphi_{x_{out}}, \varphi_{y_{out}}, \varphi_{z_{out}}$ w.r.t. the exit CSC as is seen from the equations (A2-2)

One can obtain $x_{in}, y_{in}, z_{in}, \varphi_{x_{in}}, \varphi_{y_{in}}, \varphi_{z_{in}}$ uniformly distributed within their tolerances, calculate $x_{out}, y_{out}, z_{out}, \varphi_{z_{out}}$ and check conditions

$$-\Delta_x < x_{out} < \Delta_x, -\Delta_y < y_{out} < \Delta_y, -\Delta_z < z_{out} < \Delta_z, -\varphi_z < \varphi_{z_{out}} < \varphi_z. \quad (A1-1)$$

This procedure is continued until (A1-1) will be satisfied.

In this way a distribution function of each pair of the coordinates (x_{in}, x_{out}) , (y_{in}, y_{out}) , (z_{in}, z_{out}) , and $(\varphi_{z_{in}}, \varphi_{z_{out}})$ are the same and the equality of the entrance and exit CSC are provided.

Below we present mathematical formalism to support our conclusions. Let \vec{E} and \vec{r} be vectors:

$$\vec{E} = \begin{vmatrix} \vec{e}_x \\ \vec{e}_y \\ \vec{e}_z \end{vmatrix}, \vec{r} = \begin{vmatrix} r_x \\ r_y \\ r_z \end{vmatrix},$$

where $\vec{e}_x, \vec{e}_y, \vec{e}_z$ are basic unit vectors of an arbitrary Cartesian System of Coordinate (CSC) w.r.t. "master" or accelerator hall coordinate system. Any vector defined in CSC \vec{E} is a scalar product $\vec{r} = r^T \vec{E} = r_x \vec{e}_x + r_y \vec{e}_y + r_z \vec{e}_z$ where "T" means transpose. The entrance CSC \vec{E}_{in} , exit CSC $\vec{E}_{out} = O \vec{E}_{in}$, and vector $\vec{R} = R^T \vec{E}_{in}$ - a origin of the exit CSC w.r.t. entrance CSC are defined for any device as is presented in Fig1. The 3×3 matrix O and R^T are defined by the device setup. O is the 3×3 unit matrix and $R^T = (0, 0, d_elem)$ is valid for the devices with rectilinear central trajectory. For the dipole magnets one can obtain the following matrix O and vector R:

$$O = \begin{vmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{vmatrix}, \quad R = \begin{vmatrix} x_c \\ 0 \\ z_c \end{vmatrix}.$$

The entrance CSC $\tilde{\vec{E}}_{in} = \Omega_{in} \vec{E}_{in}$ of the misaligned device is defined to be 3×3 rotation matrix Ω_{in} and a shift vector \mathbf{r}_{in} as is seen in Fig 1a. The rotation matrix $\Omega_{in} = \Omega_x \Omega_y \Omega_z$, where

$$\Omega_x = \begin{vmatrix} 1 & 0 & 0 \\ 0 & \cos\phi_x & \sin\phi_x \\ 0 & -\sin\phi_x & \cos\phi_x \end{vmatrix}, \quad \Omega_y = \begin{vmatrix} \cos\phi_y & 0 & -\sin\phi_y \\ 0 & 1 & 0 \\ \sin\phi_y & 0 & \cos\phi_y \end{vmatrix}, \quad \Omega_z = \begin{vmatrix} \cos\phi_z & \sin\phi_z & 0 \\ -\sin\phi_z & \cos\phi_z & 0 \\ 0 & 0 & 1 \end{vmatrix}$$

For small angles the rotation matrix is

$$\Omega \approx \begin{vmatrix} 1 & \phi_z & -\phi_y \\ -\phi_z & 1 & \phi_x \\ \phi_y & -\phi_x & 1 \end{vmatrix}.$$

Because the device is a rigid body, the matrix Ω and vector R^T have the same numerical values for the unmoved and misaligned device: $\tilde{\mathbf{E}}_{out} = O\tilde{\mathbf{E}}_{in}$ $\tilde{\mathbf{R}} = R^T\tilde{\mathbf{E}}_{in}$.

We can define a rotation matrix Ω_{out} and a shift vector r_{out} for exit CSC:

$$\tilde{\mathbf{E}}_{out} = \Omega_{out}\bar{\mathbf{E}}_{out}, \quad \vec{r}_{out} = \vec{r}_{in} - \vec{R} + \tilde{\mathbf{R}}.$$

From the following equalities

$$\begin{aligned} \tilde{\mathbf{E}}_{out} &= O\tilde{\mathbf{E}}_{in} = O\Omega_{in}\bar{\mathbf{E}}_{in} = O\Omega_{in}O^{-1}\bar{\mathbf{E}}_{out} = \Omega_{out}\bar{\mathbf{E}}_{out} \text{ and} \\ r_{out}^T \tilde{\mathbf{E}}_{out} &= r_{in}^T \bar{\mathbf{E}}_{in} - R^T \bar{\mathbf{E}}_{in} + R^T \tilde{\mathbf{E}}_{in} = (r_{in}^T - R^T + R^T \Omega_{in}) \bar{\mathbf{E}}_{in} = (r_{in}^T - R^T + R^T \Omega_{in}) O^{-1} \bar{\mathbf{E}}_{in} \end{aligned}$$

One can obtain

$$\Omega_{out} = O\Omega_{in}O^{-1}, \quad r_{out}^T = (r_{in}^T + R^T \Omega_{in} - R^T)O^{-1} \quad (\text{A2-2})$$

Particle motion through the device starts in the plane $z=0$ of the entrance CSC.

The particle position $\vec{u}_0, u_0^T = (x_0, y_0, 0)$ and velocity $\vec{v}_0, v_0^T = (v_{ox}, v_{oy}, v_{oz})$ (see

Fig.2a) are transformed from the entrance CSC $\bar{\mathbf{E}}_{in}$ to csc $\tilde{\mathbf{E}}_{in}$ as

$$\tilde{u}_0^T = (u_0^T - r_{in}^T)\Omega_{in}^{-1}, \quad \tilde{v}_0^T = v_0^T\Omega_{in}^{-1}$$

Using the equation of motion in field free space

$$\tilde{x} = \tilde{x}_0 - \tilde{x}'_0 \tilde{z}_0, \quad \tilde{y} = \tilde{y}_0 - \tilde{y}'_0 \tilde{z}_0, \quad \tilde{x}'_0 = \tilde{v}_{0x} / \tilde{v}_{0z}, \quad \tilde{y}'_0 = \tilde{v}_{0y} / \tilde{v}_{0z}$$

the particle is transformed to the plane $\tilde{z}_0 = 0$ of the entrance CSC $\tilde{\mathbf{E}}_{in}$.

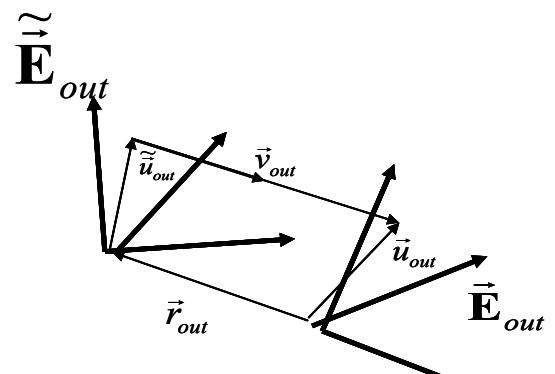
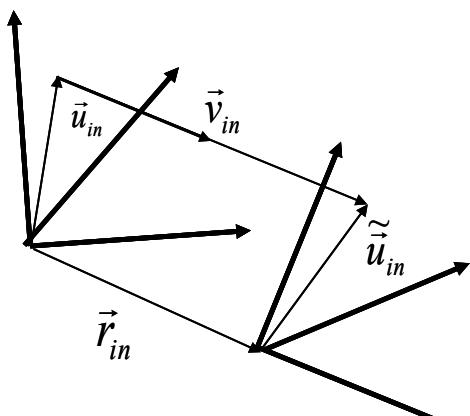


Figure 1b

Figure 1c

Then the particle is tracked through the device using the Lorenz equations and arrives at the plane $\tilde{z}_1 = 0$ of the exit CSC $\tilde{\bar{E}}_{out}$. Let us denote as $\tilde{\bar{u}}_i, \tilde{u}_{in}^T = (\tilde{x}_1, \tilde{y}_1, 0)$ the particle coordinate and velocity $\tilde{\bar{v}}_1, \tilde{v}_1^T = (\tilde{v}_{1x}, \tilde{v}_{1y}, \tilde{v}_{1z})$ on this plane, Fig.2b. The transformation rule from of the exit CSC $\tilde{\bar{E}}_{out}$ to the exit CSC \bar{E}_{out} for the particle coordinate and velocity is

$$u_1^T = \tilde{u}_1^T \Omega_{out} + r_{out}^T, \quad v_1^T = \tilde{v}_1^T \Omega_{out}.$$

Ultimately, particle is transported to the plane $z_1=0$ of the exit CSC \bar{E}_{out}

according to the equations

$$x_1 = x_1 - x'_1 z_1, \quad y = y_1 - y'_1 z_1, \quad x'_1 = v_{1x} / v_{1z}, \quad y'_1 = v_{1y} / v_{1z}$$

Misalignment errors are introduced in the file **sclinac.dat** by using the element **align**. All devices of the type **n** downstream of the line **align** experience position misalignments, phase and field errors within the tolerances defined in the line **align**.

The errors and misalignments of the device are described by 10 randomly generated numbers $R_1 \div R_{10}$. Particularly, $R_1 \div R_6$ define spatial misalignments of the device, see below for more details.

$x_{in} = \Delta_{xy} * R_1$ is the device displacement along x axis in the entrance coordinate system;

$y_{in} = \Delta_{xy} * R_2$ is the device displacement along y axis in the entrance coordinate system;

$z_{in} = \Delta_{xy} * R_3$ is the device displacement along z axis in the entrance coordinate system;

$\phi_{xin} = \phi_{xy} * R_4$ is the device rotation about x axis in the entrance coordinate system;

$\phi_{yin} = \phi_{xy} * R_5$ is the device rotation about y axis in the entrance coordinate system;

$\phi_{zin} = \phi_z * R_6$ is the device rotation about z axis in the entrance coordinate system;

Phase and amplitude fast errors are described by the Gaussian distribution with standard deviation $\sigma=1.0$ and truncated at the 3σ -level. $R_7, R_8, |R_{7,8}| \leq 3$. The fast errors are known as jitter or dynamic errors:

$\Delta\Phi_j * R_7$ is the fast phase error of the device;

$\Delta F_j * R_8$ is the fast field error of the device

R_9 and R_{10} are uniformly distributed in interval $[-1,1]$ and defined static errors of the phase and amplitude of the rf field.

$\Delta\Phi_s \cdot R_9$ is the static phase error of the device.

$\Delta F_s \cdot R_{10}$ is the static field error of the device.

In each accelerator seed 10 random numbers $R_1 \dots R_{10}$ are generated for each element defined in the file **sclinac.dat**. These data is stored in the file **VRAND_nnn.dat**, where **nnn** is the total number of accelerator seeds.

To start simulation of many seeds with errors the following steps should be applied:

- 1) In the **track.dat** file. Assign NRZ the number of required seeds, there are no misalignments for the first seed. Set **iflag_corr**=-1. Add the following lines;

```
&correct
    partCorr=1
&end
```
- 2) Add lines with the element **align** in the **sclinac.dat** file.

A1.1 Main Cycle for generation of different accelerator seeds

Before simulation with errors, TRACK performs calculation for ideal machine without errors to define phase setting of the accelerator devices and design energy of the reference particle for each device. After the first run of the whole machine without errors TRACK generates several samples of the accelerator with error. Each seed (or sample of the accelerator with error) have a number. This number is included in the correction procedure output files names. Usually, they are useful for comparison of the PC and multiprocessor calculations.

Two parameters define the number of the seeds:

```
nrz      (=1 by default) number of seeds in given run of the TRACK
nrz_min  (=1 by default) number of the first seed in a given run of the TRACK
```

The seeds have the numbers **nrz_min+nrz_min+nrz-1** in a given run.

Note:

Some seeds can have an error set that can result in unusual particle motion and extra losses. It can take place due to the resonance motion in the given set of fields or unsatisfactory work of the beam center correction procedure. The flag **iglag_vrand** can be used for the error set extraction and detailed investigation of such accelerator seed.

Examples:

1) **nrz_min=4, nrz=2, iflag_vrand=0**. The randomly generated errors are calculated for seeds #4÷5 in a given run.

2) **nrz_min=4, nrz=2, iflag_vrand=1**. The randomly generated errors are calculated for seeds #4÷5 in a given run. As a result of the run the files **vrand_004.dat** and **vrand_005.dat** will be created.

3) **nrz_min=4, nrz=2, iflag_vrand=-1**. Files **vrand_004.dat** and **vrand_005.dat** must be generated in some previous run and must exist in the directory to run the code successfully. The seeds #4÷5 are calculated with previously generated errors.

Appendix 2. Optimization of the corrector fields

The accelerator (or beam line) can be divided into several "correction sections", for correction of the spatial and angular beam center displacement from the accelerator axis due to misalignments of the elements.

Let us introduce the vectors

$$f^T = \{f_x(1), \dots, f_x(N_x), f_y(1), \dots, f_y(N_y)\}$$

$$F^T = \{F_x(1), \dots, F_x(N_x), F_y(1), \dots, F_y(N_y)\}'$$

where $f_x(i)$ and $f_y(i)$ are actual and $F_x(i)$ and $F_y(i)$ are maximum strength of the corrector number i . N_x and N_y are the number of the x -correctors and y -correctors in a given section as is seen in Fig. A2-1. This section is delimited by two first brackets as is shown in Fig. A2-1. The beam steering algorithm can be implemented in "correction sections", whereby N correctors and M monitors are related by the following two sets of equations:

$$\begin{aligned} \sum_{i=1}^{N_y} R_{12}(i, \hat{i}; \vec{B}) f_x(i) + \sum_{i=1}^{N_y} R_{14}(i, \hat{i}; \vec{B}) f_y(i) &= -x_{avr}(\hat{i}), \quad \hat{i} = 1, M \\ \sum_{i=1}^{N_x} R_{32}(i, \hat{i}; \vec{B}) f_x(i) + \sum_{i=1}^{N_y} R_{34}(i, \hat{i}; \vec{B}) f_y(i) &= -y_{avr}(\hat{i}), \quad \hat{i} = 1, M \end{aligned}, \quad (1)$$

where $x_{avr}(\hat{i})$ and $y_{avr}(\hat{i})$ are beam center displacement at the \hat{i}^{th} monitor.

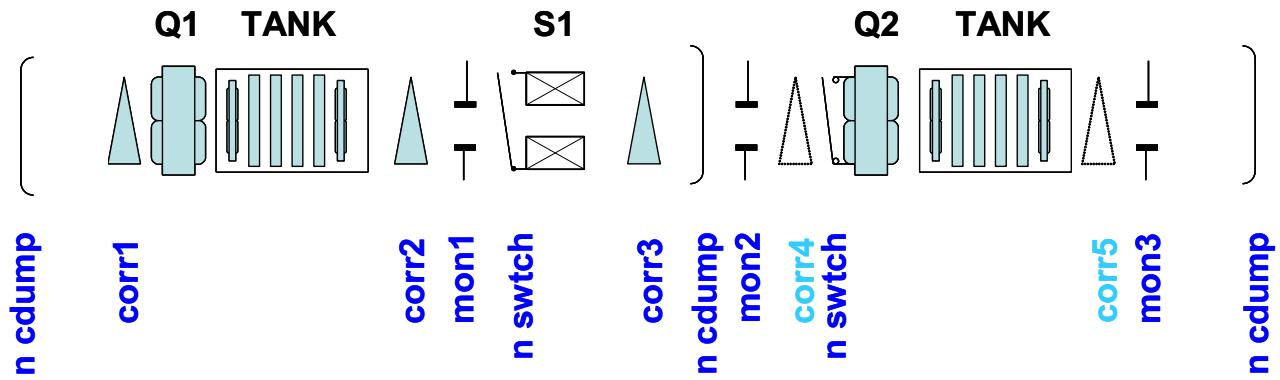


Figure A2-1.

Indexes i denote correctors, and 'covered' indexes \hat{i} denote monitors. Here $R_{12}(i, \hat{i}), R_{32}(i, \hat{i}), R_{14}(i, \hat{i}),$ and $R_{34}(i, \hat{i})$ are transfer functions from the corrector i to the monitor \hat{i} . \vec{B} denotes the vector containing device parameters in the given section.

The test particle with coordinates $\{x_{test} = 0, x'_{test} = 0, y_{test} = 0, y'_{test} = 0, \varphi_{test} = 0, W_{test} = \bar{W}\}$ at the entrance of each correction section is used to evaluate the transfer function. \bar{W} is an average beam energy. Beam center, $x_{test}(\hat{i})$, $y_{test}(\hat{i})$, at the monitor number \hat{i} is calculated for special sets of the correctors' strengths

$$\begin{aligned} f^T &= (0, \dots, \eta_x(i), \dots, 0, 0, \dots, 0) \quad i = 1, N_x, \text{ or} \\ f^T &= (0, \dots, 0, 0, \dots, \eta_y(i), \dots, 0) \quad i = N_x + 1, N_x + N_y \end{aligned} \quad (2)$$

The transfer functions are defined as

$$\begin{aligned}
R_{12}(i, \hat{i}) &= \frac{[x_{test}(\hat{i}) + \delta x(\hat{i})]}{\eta_x(i)}, R_{32}(i, \hat{i}) = \frac{[y_{test}(\hat{i}) + \delta y(\hat{i})]}{\eta_x(i)}, i = 1, N_x \\
R_{14}(i, \hat{i}) &= \frac{[x_{test}(\hat{i}) + \delta x(\hat{i})]}{\eta_y(i)}, R_{34}(i, \hat{i}) = \frac{[y_{test}(\hat{i}) + \delta y(\hat{i})]}{\eta_y(i)}, i = N_x + 1, N_x + N_y
\end{aligned} \tag{3}$$

$\delta x(\hat{i}), \delta y(\hat{i})$ are beam position measurement errors.

We can repeat calculations (or measurements with the real beam) with other set parameters \tilde{B} (for example, turn off the quad Q2 in Fig A2-1) and obtain the additional correction equations

$$\begin{aligned}
\sum_{i=1}^{N_y} R_{12}(i, \hat{i}; \tilde{B}) f_x(i) + \sum_{i=1}^{N_y} R_{14}(i, \hat{i}; \tilde{B}) f_y(i) &= -\tilde{x}_{avr}(\hat{i}), \hat{i} = 1, M \\
\sum_{i=1}^{N_x} R_{32}(i, \hat{i}; \tilde{B}) f_x(i) + \sum_{i=1}^{N_y} R_{34}(i, \hat{i}; \tilde{B}) f_y(i) &= -\tilde{y}_{avr}(\hat{i}), \hat{i} = 1, M
\end{aligned} \tag{4}$$

We can rewrite equations (3) in matrix form $(A + \Delta A)f = -(b + \Delta b)$, where transfer matrix $A + \Delta A$ is $2M \times (N_x + N_y)$ rectangular matrix and equals to

$$A + \Delta A = \begin{vmatrix} R_{12}(1, \hat{i}; \vec{B}) & \dots & R_{12}(N_x, \hat{i}; \vec{B}) & R_{14}(1, \hat{i}; \vec{B}) & \dots & R_{14}(N_y, \hat{i}; \vec{B}) \\ \dots & \dots & \dots & \dots & \dots & \dots \\ R_{12}(1, \hat{M}; \vec{B}) & \dots & R_{12}(N_x, \hat{M}; \vec{B}) & R_{14}(1, \hat{M}; \vec{B}) & \dots & R_{14}(N_y, \hat{M}; \vec{B}) \\ R_{32}(1, \hat{i}; \vec{B}) & \dots & R_{32}(N_x, \hat{i}; \vec{B}) & R_{34}(1, \hat{i}; \vec{B}) & \dots & R_{34}(N_y, \hat{i}; \vec{B}) \\ \dots & \dots & \dots & \dots & \dots & \dots \\ R_{32}(1, \hat{M}; \vec{B}) & \dots & R_{32}(N_x, \hat{M}; \vec{B}) & R_{34}(1, \hat{M}; \vec{B}) & \dots & R_{34}(N_y, \hat{M}; \vec{B}) \end{vmatrix}, \quad b + \Delta b = \begin{vmatrix} x_{avr}(\hat{i}) \\ x_{avr}(\hat{M}) \\ y_{avr}(\hat{i}) \\ y_{avr}(\hat{M}) \end{vmatrix} \tag{5}$$

In equation (5) transfer matrix and right hand vector are divided into two parts. A and b are calculated without beam position measurement errors ($\delta x(\hat{i}) = \delta y(\hat{i}) = 0$). ΔA and Δb are random numbers because of the devices' field errors and finite accuracy of the monitors. To find the correctors' strengths we minimize over the vector f the function

$$\Omega(f) = \sum_{i=1}^{2M} \left(\frac{\sum_{k=1}^{N_x+N_y} (A_{ik} + \Delta A_{ik}) f_k + b_i + \Delta b_i}{w_i} \right)^2 \quad \text{with restrictions } |f_k| \leq F_k, \tag{6}$$

where w_i is the 'weight' of the i^{th} monitor. Usually w_i is equal to the standard deviation of the measurement errors of the monitor. The residual beam displacement on the i^{th} monitors after correction is nonzero and equals to

$$\delta_i = -b_i + \sum_{k=1}^{N_x+N_y} A_{ik} f_k$$

We estimate the quality of the given correction procedure by solving (6) Nr times with monitor errors $\delta w(\bar{i})$ uniformly distributed in the interval $[-w_i, w_i]$ $i = 1, M$. The correlation matrix for the correctors' strength is

$$C_{ij}(f) = \frac{\sum_{m=1}^{Nr} [f_i(m) - \bar{f}_i(m)][f_j(m) - \bar{f}_j(m)]}{Nr-1}, \text{ where } \bar{f}_i = \frac{\sum_{m=1}^{Nr} f_i(m)}{Nr}, i, j = 1, N_y + N_y$$

and for residual beam positions on monitors is

$$C_{ij}(\delta) = \frac{\sum_{m=1}^{Nr} [\delta_i(m) - \bar{\delta}_i(m)][\delta_j(m) - \bar{\delta}_j(m)]}{Nr-1}, \text{ where } \bar{\delta}_i = \frac{\sum_{m=1}^{Nr} \delta_i(m)}{Nr}, i, j = 1, 2M.$$

Abnormal value $\sqrt{C_{ii}(f)} \approx F_i$ where F_i is the maximum strength of the i^{th} corrector or $\sqrt{C_{ii}(\delta)} > w_i$ gives warning of possible difficulties of the given correction procedure.

Useful information about the solution of the matrix A can be obtained from the covariance matrix of the set of solutions f_n for randomly defined matrix $A_n = A + \Delta A_n$ and right-hand side $b + b_n$ for N samples of the correction equations. The code TRACK provides four possibilities depending on parameter `MonErr` for calculations of the covariance matrix C. For each seed, the devices are misaligned and their phases and field amplitudes are deviated from the nominal values. The set of particles `npatC(iq)`, $iq=1, nqtot$, where $npatC(iq) \leq npat(iq)$, is transported through the correction section to calculate the vector b and the beam with zero transverse and longitudinal emittances is transported through the section N_{eq} times. Each time only one corrector is on (remember, that the device `corr` define simultaneously two correctors), the first deflects the beam in the horizontal plane and the second - in the vertical plane. The correctors' strength F_i for the correction matrix calculation is equal to `ThetaH` or `Bymax`.

- 1) `MonErr=1` (by default). In each seed errors and misalignments are generated for each device and BPMs measure beam position in each monitor. Matrix A_n and b_n are defined in each seed with the errors of the devices position and field errors, in this case $\Delta b=0$.
- 2) `MonErr=0` beam experiences errors only due to the device misalignments and field errors, in this case $\Delta b=0$ (no measurement errors)
- 3) `MonErr >1` The matrix A and b are calculated including randomly generated monitor errors $\delta w(\bar{i})$ with uniform distribution in the interval $[-dw(\bar{i}), dw(\bar{i})]$ $i=1, N_{mon}$. The minimization problem (6) is solved M=`MonErr` times for different sets of the monitor errors.

$$C_{ij}(f) = \frac{\sum_{m=1}^{Nr} [f_i(m) - \bar{f}_i(m)][f_j(m) - \bar{f}_j(m)]}{Nr-1}, \text{ where } \bar{f}_i = \frac{\sum_{m=1}^{Nr} f_i(m)}{Nr}, i, j = 1, N_y + N_y$$

$$R_{24}(k, \bar{i}) = \frac{[x_{test}(\bar{i}) + \delta x(\bar{i})]}{f_y(k)}, R_{34}(k, \bar{i}) = \frac{[y_{test}(\bar{i}) + \delta y(\bar{i})]}{f_y(k)}$$

$$f^T = (0, \dots, 0, f_y(k) = \eta_y(k)F_y(k), 0, \dots, 0), k = 1, N_y$$

$$\Omega(f) = \sum_{i=1}^{2M} \left(\frac{\sum_{k=1}^{N_x+N_y} A_{ik} f_k + b_i}{w_i} \right)^2$$

Efficiency of the correction procedure is determined by the condition number of the transfer matrix A . The large condition number indicates that small changes in A and b can lead to large changes in the solution f .

Appendix 3. Acceptance calculation

The code allows one to generate both transverse and longitudinal acceptance of the accelerating-focusing system. The acceptance picture at the beginning of the accelerator or transport system can be obtained as an area in the phase space occupied by survived particles which are tracked to the end of the system defined in the file **sclinac.dat**. To simulate acceptance two flags either **iaccep=1** for the longitudinal acceptance or **iflag_tac=1** for the transverse acceptance are used (see the description of the file **input.dat**). As a result two files **accep.dat** and **negative.dat** are generated. The coordinates of the survived particles are stored in **accep.dat**. Coordinates of the lost particles are stored in **negative.dat**. Options **iaccep=1** and **iflag_tac=1** don't work together.

For the acceptance calculations **NRZ** in the **input.dat** file must be equal to 1. The output data can be processed using EXCEL or ORIGIN to produce the acceptance picture. Acceptance calculation in one transverse phase space usually should be done with negligible emittance in another transverse plane.

Appendix 4. Potential expansion in a multi-cell RFQ

A Radio Frequency Quadrupole accelerator can be separated into three functional sections:

- 1) A regular bunching or accelerating section.
- 2) A transition cell between the radial matcher and regular cell.
- 3) Entrance and exit radial matchers.

For the beam dynamics simulation in the TRACK code we assume electrostatic approximation of the fields. The RFQ fields can be inserted into the code by using numerical 3D tables and/or by using the coefficients of Fourier-Bessel expansion of the electrostatic potential in each cell. The design of the vane profile must be known and an external code must be used to generate 3D fields in the RFQ sections and cells. Usually we use the code DESRFQ [1] which is available through contact to Dr. A.A. Kolomiets (ITEP, Moscow). Also the PARMTEQ code of LANL can be used to generate TRACK input files.

The transverse cross-section of the vanes has a shape shown in Fig. A4-1. We assume that the regular RFQ cells are sinusoidally shaped:

$$\begin{aligned} x_t &= r_0 \left(1 + \frac{m-1}{m+1} \cos(kz) \right) \\ y_t &= r_0 \left(1 - \frac{m-1}{m+1} \cos(kz) \right), \quad k = \pi/L, \quad -L/2 < z < L/2 \end{aligned}, \quad (1)$$

where a and ma are the minimum and maximum distance of the vane tips wrt the z -axes, x_t and y_t are the coordinates of the horizontal and vertical vanes,

$r_0 = \frac{a+ma}{2}$ is the average radius of the vanes and L is the cell length as shown in Fig. A4-2.

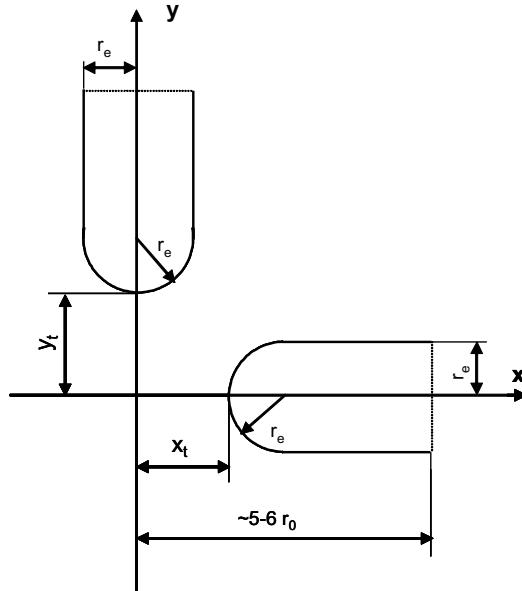


Fig. A4-1. Transverse cross-section of the RFQ vanes.

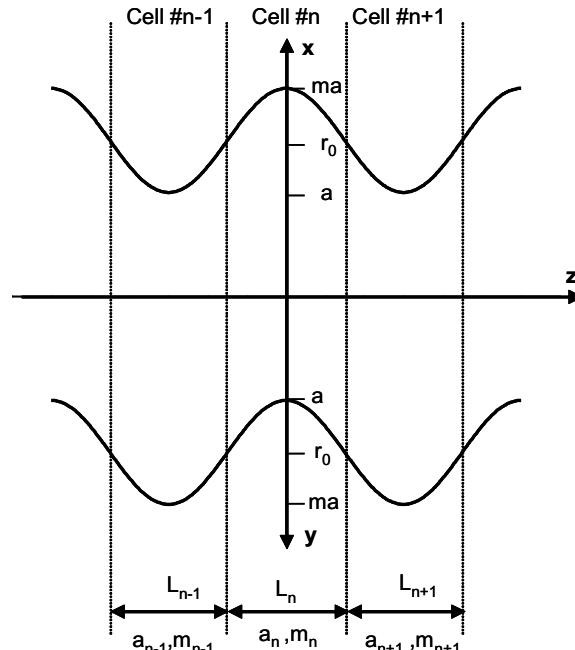


Fig. A4-2. Longitudinal profile of the RFQ vanes.

A4.1 Regular accelerating cells

In the TRACK code we assume that the cell begins at the point where the vane tips have exact quadrupole symmetry as is shown in Fig. A4-2. As usual, we assume that the parameters a, m, L are varied slowly along the RFQ and the potential in a regular cell can be found applying a mirror symmetry wrt the xy-plane at the beginning and the end of cell. Let's assume $z=0$ at the center of the cell then the potential function satisfies symmetry conditions which can be written in cylindrical coordinates as:

$$\begin{aligned} U(r, -\vartheta, z) &= U(r, \vartheta, z) \\ U(r, \pi/2 + \vartheta, z) &= -U(r, \pi/2 - \vartheta, z) \\ U(r, \vartheta, -z) &= U(r, \vartheta, z) \\ U(r, \vartheta + \pi/2, L - z) &= -U(r, \vartheta, z) \end{aligned} \quad (2)$$

The potential is:

$$\begin{aligned} U(r, \theta, z) &= \frac{|V|}{2} \sum_{m=0}^{\infty} \left\{ A_{0,2m+1} (r/R_0)^{2(2m+1)} \cos[2(2m+1)\theta] + \right. \\ &\quad \left. \sum_{n=1}^{\infty} A_{2n,2m+1} I_{2(2m+1)}(2nkr) \cos[2(2m+1)\theta] \cos(2nkz) + \right. \\ &\quad \left. \sum_{n=1}^{\infty} A_{2n-1,2m} I_{4m}[(2n-1)kr] \cos(4m\theta) \cos[(2n-1)kz] \right\} \end{aligned} \quad (3)$$

The potential expansion in a regular RFQ cell with an accuracy of the order r^7 contains eight terms:

$$\begin{aligned} U(r, \theta, z) &\approx \frac{|V|}{2} \left[A_{01}(r/r_0)^2 \cos(2\theta) + A_{10} I_0(kr) \cos(kz) \right] + \\ &\quad \frac{|V|}{2} \left[A_{03}(r/r_0)^6 \cos(6\theta) + A_{21} I_2(2kr) \cos(2\theta) \cos(2kz) + A_{12} I_4(kr) \cos(4\theta) \cos(kz) + \right. \\ &\quad \left. A_{30} I_0(3kr) \cos(3kz) + A_{23} I_6(2kr) \cos(6\theta) \cos(2kz) + A_{32} I_4(3kr) \cos(4\theta) \cos(3kz) \right] \end{aligned} \quad (4)$$

In the TRACK code, the line in the input file **RFQ.#nn** for regular RFQ cell may contain thirteen parameters

$$\{\beta, T, \kappa, a, L, index, A_{21}, A_{12}, A_{30}, A_{03}, A_{23}, A_{32}, eh_EM.#nn\}, \quad (5)$$

where $\beta=v/c$ is the reference particle velocity at the exit of the cell, $T=\pi A_{10}/4$, $\kappa=a^2 A_{01}$, $index=0$. First six parameters in this line define two-term potential approximation ($A_{01} \neq 0$, $A_{10} \neq 0$). The first column is used only for the graphic outputs and does not effect on particle tracking.

Six parameters A_{mn} and the file name $eh_EMS.#nn$ are optional. By default all 6 coefficients A_{mn} are equal to zero. These coefficients must be defined by an external code for the given design of the RFQ if one likes to track particles in 8-term potential. If the file name $eh_EMS.#nn$ is appeared in the line, the field in the cell must be given

as a 3D table field in the file eh_EMS.#nn. In this case TRACK reads file eh_EMS.#nn and uses this field to interpolate for trajectory integration.

The command line in the sclinac.dat file

```
1 prmtr ncoef=k01 k10 k21 k12 k30 k03 k23 k32
```

is used to turn off the effect of each term in series (4). By default all coefficients k_{mn} are equal to 1. Defining $k_{mn}=0$ one can make $A_{mn}=0$ for all regular cells even if these coefficients are not equal to zero in the line (5) in the **sclinac.dat** file.

A period of modulation with the length $2L$ consist of two cells. For certainty, we assume the vane tip in horizontal plane in the center of n -th cell is at the maximum and the vane tip in the vertical plane is at the minimum wrt the z -axis. In the $(n+1)$ -cell the location of H- and V-vane tips wrt the z -axis is opposite. For each cell, the potential expansion is performed assuming that the center of z -coordinate is located in the center of cell and H-vane tip reaches its minimum distance wrt the z -axis. Usually RFQ begins with the radial matcher which does not have any modulation. The first RFQ cell with modulation has maximum distance of the vane tip from the z -axis in horizontal plane at the center of the cell. In order to reverse this condition, one should define vane=-1 in the line

```
n rfq ....
```

A4.2 Transition cell

Transition cell is the cell between two sections of the RFQ with non-adiabatic change of modulation. For example, a cell between the uniform section of the RFQ without any modulation and the section with modulation is the transition cell. In the transition cell the potential function satisfies symmetry conditions:

$$\begin{aligned} U(r, -\theta, z) &= U(r, \theta, z) \\ U(r, \pi/2 - \theta, z) &= U(r, \pi/2 + \theta, z) \end{aligned} \quad (6)$$

The potential can be expanded into series

$$U(r, \theta, z) = \frac{|V|}{2} \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} B_{2m, 2m+2n}(z) r^{2m+2n} \cos(2m\theta) \quad (7)$$

with recursive relation between the series coefficients:

$$B_{2m, 2m+2n+2} = \frac{1}{(2m)^2 - [2m+2n+2]^2} B''_{2m, 2m+2n}(z) \quad (8)$$

The field distribution in the transition cell is essentially three dimensional. To define transition cell fields in TRACK, we use 3D field tables obtained from EM Studio. Also, an analytical field expansion in the transient cell is available in TRACK with an accuracy of order r^3

$$U(r, \vartheta, z) = B_{00}(z) - \frac{1}{4} B''_{00}(z) r^2 + B_{22} r^2 \cos 2\vartheta \quad (9)$$

Vane tip profiles and potential expansion in the transition cell between modulated and unmodulated vanes has been discussed by K. Crandall [2]. K. Crandall divides transition cell into two sections as is shown in Fig. A4-3. In the first section where $-L/2 < z < 0$, the vane tip modulation is similar as in a regular accelerating cell. At $z > 0$ the vane modulation is given by the expression:

$$\begin{aligned}x_t &= r_0 + (ma - r_0)Cr(Kz) \\y_t &= r_0 + (a - r_0)Cr(Kz) \quad 0 \leq z \leq L' , \\Cr(Kz) &= \frac{3}{4}\cos\frac{Kz}{2} + \frac{1}{4}\cos\frac{3Kz}{2}\end{aligned}\quad (10)$$

where $K = \pi/L'$ $L' = \sqrt{0.75}L$. In ref. [2] it is proposed to provide continuity of the functions $x_t(z)$ and $y_t(z)$ and its first and second derivatives in the points of conjugation at $z=0$ and $z=L'$ as is shown in Fig. A4-3. If the vane modulation is given by (10), a good approximation to the expansion coefficients in the expression (9) is:

$$\begin{aligned}B_{00} &= A_{10} \cos(kz) \quad , -L/2 < z < 0 \\B_{00} &= A_{10}Cr(Kz) \quad , \quad 0 < z < L' \\B_{22} &= A_{01}/r_0^2 \quad , -L/2 < z < L'\end{aligned}\quad (11)$$

where A_{01}, A_{10} are the expansion coefficients of the series (4) for the regular cell with the parameters a, m, L, r_e . Fig. A4-4 shows the longitudinal electric field obtained from numerical calculations with EM Studio and analytical expansion (9).

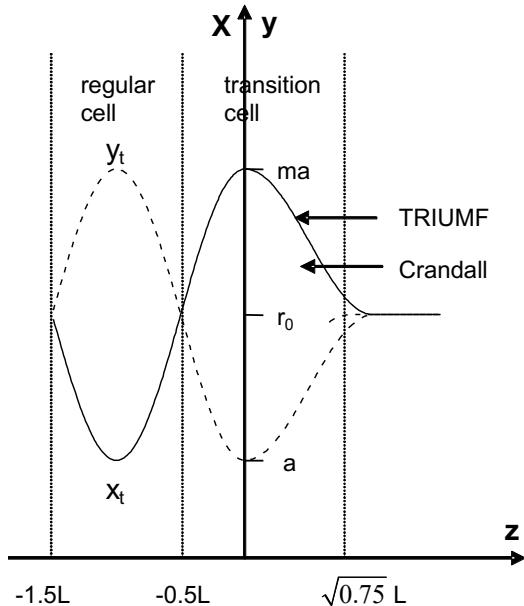


Fig. A4-3. Transition cell.

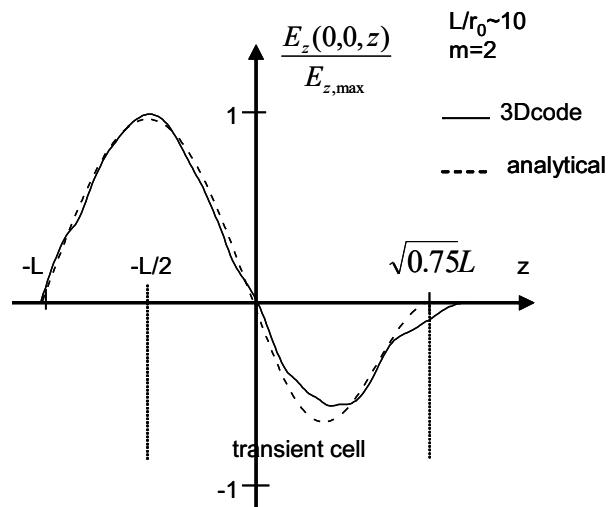


Fig. A4-4. Electric field distribution in the transition cell. The results were obtained by EM studio and analytically.

In the RFQ for ISAC at TRIUMF a sinusoidal modulation of the vanes for $z > 0$ has been applied [3]:

$$x_t = \frac{r_0 + a}{2} + \frac{r_0 - a}{2} \cos\left(\frac{\pi z}{L}\right)$$

$$y_t = \frac{r_0 + ma}{2} + \frac{r_0 - ma}{2} \cos\left(\frac{\pi z}{L}\right) \quad 0 < z < L$$

The line in the input file **RFQ.#nn** for transient RFQ cell may contain seven parameters

$$\{\beta, T, \kappa, a, L_T, index, eh_EM.\#nn\}, \quad (12)$$

where *index*=2 is for the entrance transition cell and *index*=-2 is for the exit transition cell and $L_T = (0.5 + \sqrt{0.75})L$. Parameter *eh_EM.#nn* appears if 3D-table of the electric field obtained from EM studio code will be used.

A4.3 Entrance and exit radial matchers

In radial matcher section (RMS) the potential function has symmetry planes $x=0$, $y=0$, $x=y$ and satisfies the symmetry conditions:

$$\begin{aligned} U(r, -\theta, z) &= U(r, \theta, z) \\ U(r, \pi/2 - \theta, z) &= U(r, \pi/2 + \theta, z) \\ U(r, \pi/2 - \theta, z) &= -U(r, \theta, z) \end{aligned} \quad (13)$$

The potential can be expanded into the series

$$U(r, \theta, z) = \frac{|V|}{2} \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} B_{2(2m+1), 2(2m+1)+2n}(z) r^{2(2m+1)+2n} \cos[2(2m+1)\theta] \quad (14)$$

with recursive relation

$$B_{2(2m+1), 2(2m+1)+2n+2} = \frac{1}{[2(2m+1)]^2 - [2(2m+1) + 2n + 2]^2} B''_{2(2m+1), 2(2m+1)+2n}(z). \quad (15)$$

With the accuracy of r^7 as in a regular cell, the potential expansion in the RMS is

$$U(r, \theta, z) = \frac{|V|}{2} B_{22} \left[g(z) r^2 - \frac{g''(z)}{12r_0^2} r^4 + \frac{g^{(4)}(z)}{384r_0^4} r^6 \right] \cos 2\theta + \frac{|V|}{2} B_{66} s(z) r^6 \cos 6\theta \quad (16)$$

Where $B_{22} = A_{01}/r_0^2$ and $B_{66} = A_{03}/r_0^6$, A_{01} and A_{03} is the expansion coefficients of the series (???) for the cell without modulation. Let's assume $z=0$ at the beginning of the RMS. The functional form of a falloff function $g(z)$ is

$$g(z) = \begin{cases} 1 & 0 \leq z < L_1 \\ \tilde{g}(z) & L_1 \leq z \leq L_1 + D \\ 0 & L_1 + D \leq z \leq L_m \end{cases} \quad (17)$$

Where L_m is the RMS length. As a $\tilde{g}(z)$ one can use

$$\tilde{g}(z) = (z - L_1)/D, \quad \text{linear function}$$

(18)

$$\tilde{g}(z) = \frac{3}{4} \cos\left[\frac{\pi(z - L_1)}{2D}\right] + \frac{1}{4} \cos\left[\frac{3\pi(z - L_1)}{2D}\right], \quad \text{Crandall function [2]} \quad (19)$$

$$\tilde{g}(z) = 0.5[1 + \cos[\pi(z - L_1)/D]], \quad \text{sinusoidal ramp [3]} \quad (20)$$

For the falloff function one can define two parameters: effective length of the field falloff and effective length of the transition region:

$$H_{eff} = \int_0^{L_m} g(z) dz, \quad D_{eff} = 1/|g'(z_D)|, \quad ,$$

(21)

where z_D is defined as a solution of the equation $g''(z_D) = 0$.

Let's define the vane tip profile in the RMS as

$$x_t(z) = y_t(z) = r_0 / \sqrt{g(z)}, \quad 0 \leq z \leq z_v, \quad (22)$$

where z_v is the endpoint of the vanes. If the condition $x_t(z_v)/r_0 = 1/\sqrt{g(z_v)} \geq 5$ is satisfied, the falloff function behavior does not depend from the location of the point z_v .

Fig. A4-5 and A4-6 show different falloff functions and vane profiles calculated by the expression (22) at the same values of parameters H_{eff} and D_{eff} for $x_t(z_v) = 8$. The vane profiles to produce the falloff functions (19) and (20) have smoother field transition compared with the low (1). Once there is no significant difference between falloff functions (19) and (20), in the TRACK code we use the sinusoidal ramp (20) for the analytical representation of the field in the entrance and exit RMS.

The RMS described by the expression (22) and (20) has been simulated by EM Studio which shown that the Fourier expansion of the numerically calculated potential reproduces the analytical as is shown in Fig. A4-7.

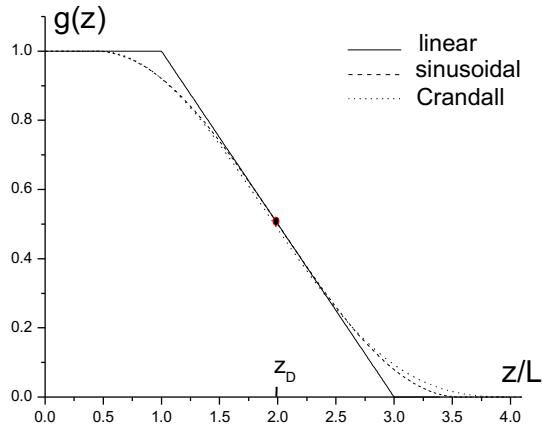


Fig. A4-5. Falloff fucntions.

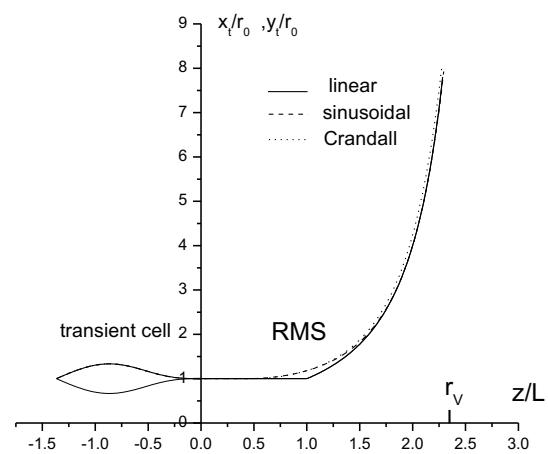


Fig. A4-6. Vane profiles corresponding to falloff functions given in Fig.5.

The line in the input file **RFQ.#nn** for RMS may contain 11 parameters

$$\{\beta, H_{\text{eff}}, D_{\text{eff}}, r_0, L_M, \text{index}, A_{01}, A_{03}, k_H, k_D, \text{eh_EM.\#nn}\}, \quad (23)$$

where H_{eff} , D_{eff} - effective length and the effective transient length of the quadrupole field component, $L_1 = H_{\text{eff}} - \pi D_{\text{eff}}/4$, $D = \pi D_{\text{eff}}/2$, see equation (9) A_{01} , A_{03} are the expansion coefficients of the series (2) for the unmodulated cell, k_H (=1 by default, optional)-correction coefficient for the effective length, of the duodecapole field component, k_V (=1 by default, optional)-correction coefficient for the effective transient length of the duodecapole component, $\text{index}=1$ for the entrance transient cell and $\text{index}=-1$ for the exit. Parameter eh_EM.\#nn is optional. Appearance of the parameter em_EM.\#nn in cells description string means that TRACK uses the 3D-table field approximation for this transition cell.

```
The code MWS_read.exe with input.dat file as defined below
&fasa
device='RFQM'
r_fourier=.6
Vf=1
file_MWS_e='cellM.dat',
file_MWS_h='empty',
key_1_e=1, key_2_e=1, key_3_e=0,
aperture=2*r0
d_device=L_M
z_device=
&end
```

produces a line (23) for given input 3D-table cellM.dat.

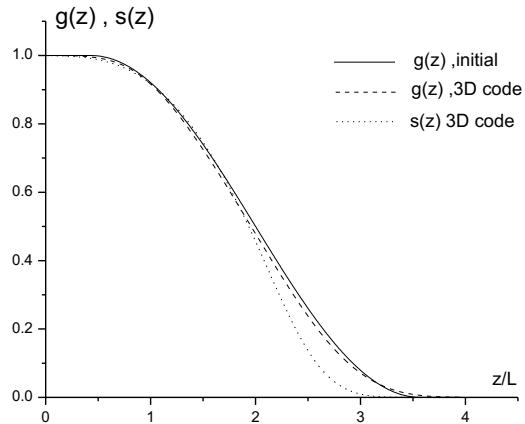


Fig. A4-7. Analytical and computer simulated falloff functions.

References

- A4-1. A.A. Kolomiets, T.E. Tretjakova and S.G. Yaramishev. DESRFQ – a code for Generation of Radio Frequency Quadrupole. ITEP report, 2001.
- A4-2. K. R. Crandall Effects of Vane-Tip geometry on the Electric Fields in Radio-Frequency Quadrupole Linacs. Technical report LA-9595, LANL, April 1983.
- A4-3. L. Root. TRIUMF report. 1996.

Appendix 5. Particle simulation in dipole magnets

Four coordinate systems (CS) are used for the dipole: an entrance CS (X_{in}, Z_{in}), an exit CS (X_{out}, Z_{out}), an entrance fringe field CS (X_1, Z_1) and an exit fringe field CS (X_2, Z_2). Axis Z_1 is normal to the entrance effective field boundary. Axis Z_2 is normal to the exit effective field boundary. The positive Y direction is out of the page, see Fig. A5-1. TRACK sign convention for parameters is identical to the sign convention in the TRANSPORT code. $\theta > 0$ means rotation of the beam direction clockwise about positive Y axis. Positive β_1, β_2 imply transverse focusing i. e. focusing in YZ plane. Positive R_1, R_2 (convex curvature) represent negative sextupole components. On Fig. A5-1 $\theta > 0$, $\beta_1 > 0$, $\beta_2 < 0$, $R_1 > 0$, $R_2 > 0$.

The central trajectory is defined using the "hard edge" approximation as a straight line, a circular arc and another straight line. There is no particle moving along this trajectory in realistic field. As a central trajectory we use the trajectory which coincides with the circular arc inside the dipole, see Fig. A5-2. We also suggest that the beam enters the dipole along Z_{in} -axis and exit along Z_{out} -axis. To provide these conditions, the dipole is artificially moved $\Delta x = -x_{cr1}$ and $\Delta z = -x_{cr2}/\sin(\theta)$ w.r.t. the entrance coordinate system (X_{in}, Z_{in}). XCR1 and XCR2 are parallel shifts of the realistic central trajectory w.r.t. the hard edge central trajectory at entry and exit fringe fields respectively [A5-1, A5-2]. The code TRACK evaluates XCR1 and XCR2 and shifts the dipole automatically.

TRACK dipole aperture is a curvilinear tube with the rectangular cross section. The shape of the tube is defined in Fig. A5-3. The vacuum tube has finite thickness and it is taken into account by the parameter δ_1 ($\delta_1=0.9$ by default).

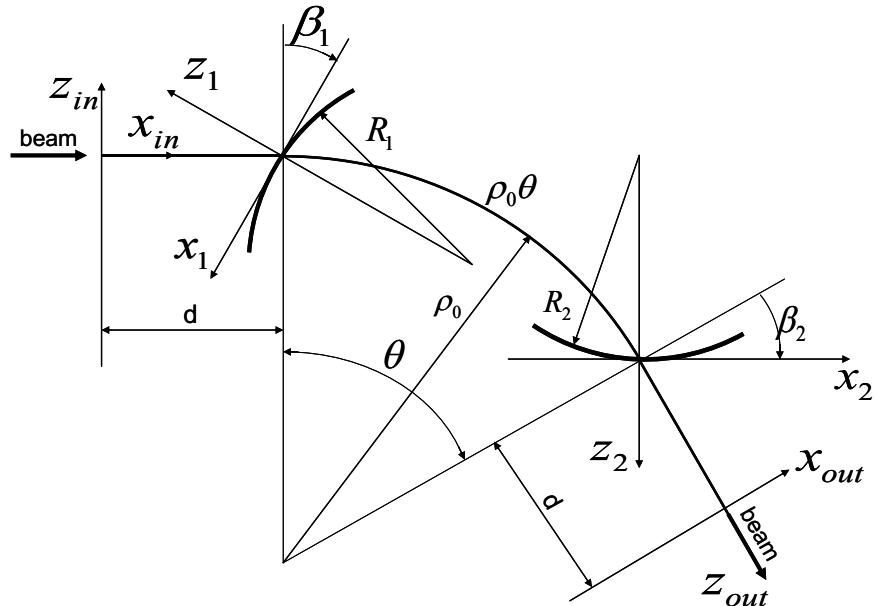


Fig. A5-1: Definition of dipole parameters in the TRACK code.

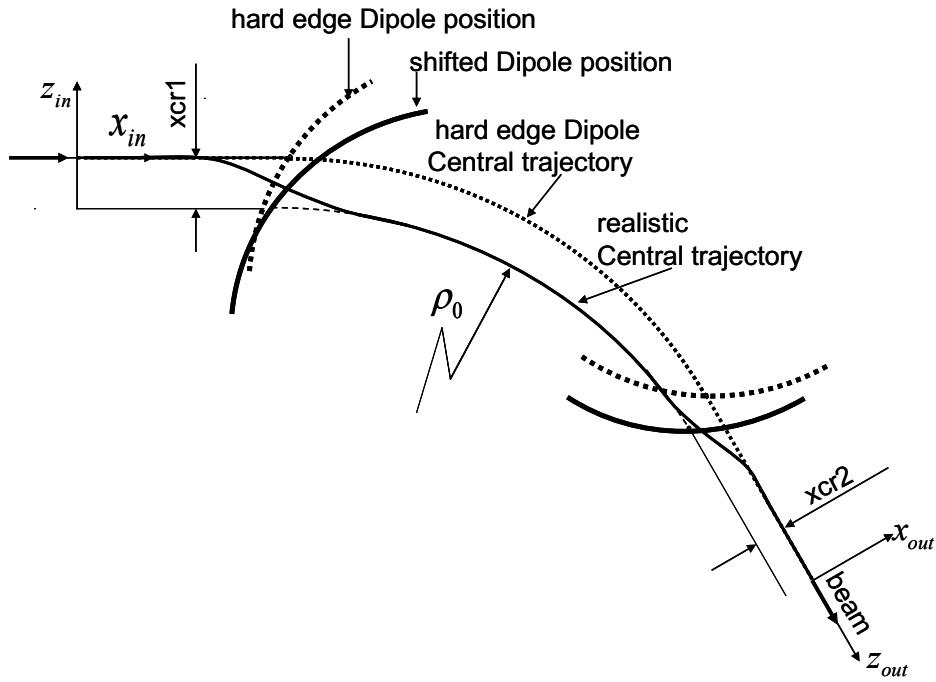


Fig. A5-2: The reference particle (central) trajectory.

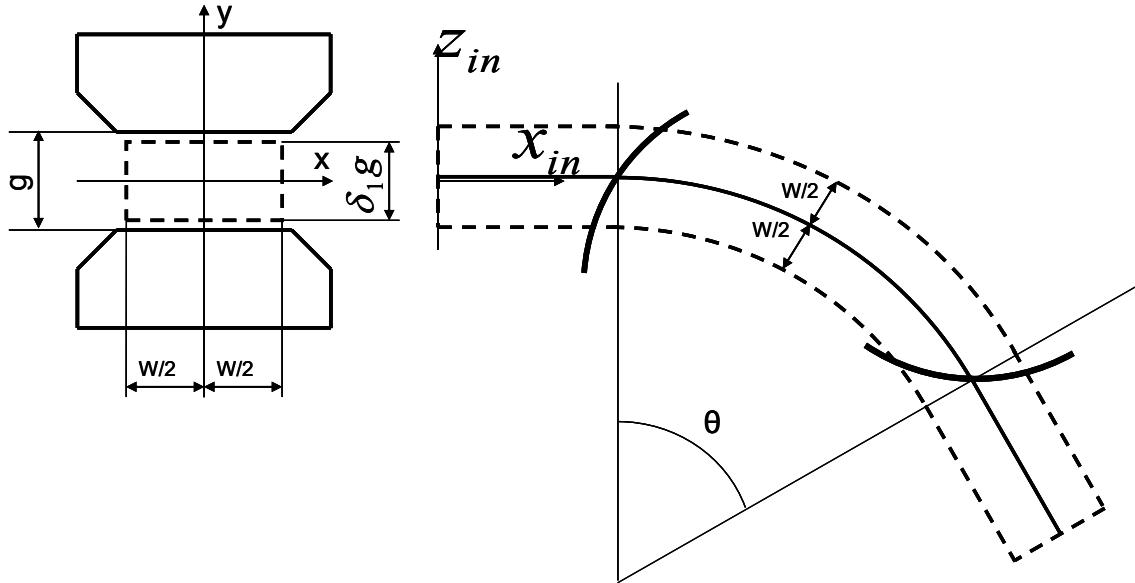


Fig. A5-3. The dipole aperture.

The element **bmag** is a dipole magnet with the homogenous magnetic field in the central region of the magnet and rounded pole tip edges. Due to rounded pole shape (see Fig A5-1), only the field components $B_y(0, y, z)$ and $B_z(0, y, z)$ are needed for the field calculations in the fringe field regions. We use the Enge type function $E(z)$ for magnetic field falloff at the dipole edges {A5-5, A5-6}, see Fig. A5-4.

$$E(z) = \frac{B_y(0, 0, z)}{B_f} = \frac{1}{1 + \exp(c_0 + c_1 s + c_2 s^2 + c_3 s^3 + c_4 s^4 + c_5 s^5)}, \quad s = \frac{z}{g} \quad (\text{A5-1})$$

$$B_y(0, y, z) = B_f \left[E(z) - \frac{E''(z)}{2} y^2 + \frac{E^{(4)}(z)}{24} y^4 - \frac{E^{(6)}(z)}{720} y^6 \right] \quad (\text{A5-2})$$

$$B_z(0, y, z) = B_f \left[E'(z)y - \frac{E'''(z)}{6} y^3 + \frac{E^{(5)}(z)}{120} y^5 \right]$$

The entrance fringe field covers an interval from $z11=-\delta_2 g$ to $z12=w\sin(|\beta_1|)+d\cos(\beta_1)$ and the exit fringe field covers an interval from $z21=-\delta_2 g$ to $z22=w\sin(|\beta_2|)+d\cos(\beta_2)$.

2D field tables are also available for the **bmag** fringe fields evaluation. These tables have to be defined on a rectangular grid $\{0 \leq y \leq g/2, z11 \leq z \leq z12\}$ for entrance fringe field and on a rectangular grid $\{0 \leq y \leq g/2, z21 \leq z \leq z22\}$ for the exit fringe field, see Fig. A5-5B

The codes "TRANSPORT" [A5-1] and "TRACE-3D" [A5-2] use the parameter

$$k1 = \frac{\int_{z11}^{z12} E(z)(1-E(z))dz}{z12-z11} \quad (\text{A5-3})$$

the code "GIOS" [A5-3] uses the function $E(z)$, and the code "RAYTRACE" [A5-4] uses 2D tables with $B_y(x, 0, z)$ for fringe field description.

The momentum p of the reference particle, magnetic field B_f in the uniform region, and $k1$, $xcr1$, $xcr2$, $x11$, $z11$, $x12$, $z12$, $x21$, $z21$, $x22$ and $z22$ are the output parameters of the TRACK code.

This section is the description of the element "dipo". Currently it is under development.

The "dipo" is a dipole with the homogenous field or field with a field gradient inside the magnet (the uniform field region in the element "bmag"). The field expansion in the median plane is given by:

$$B_y(\Delta\rho) = B_f \left[1 - N \frac{\Delta\rho}{\rho_0} + B \left(\frac{\Delta\rho}{\rho_0} \right)^2 + \Gamma \left(\frac{\Delta\rho}{\rho_0} \right)^3 + \Delta \left(\frac{\Delta\rho}{\rho_0} \right)^4 \right], \quad \Delta\rho = \rho - \rho_0, \quad (\text{A5-4})$$

where ρ is the distance between the point and the curvature center of the layout circular arc (see fig. A5-1). 2D tables are used for the definition of the magnetic field in the fringe field regions. These tables are defined in the Cartesian coordinates, see Fig. A5-4, $\{-x_{\max} \leq x \leq x_{\max}, z_{\min} \leq z \leq z_{\max}\}$, where $x_{\max} = \max(x_{11}, x_{12}, x_{21}, x_{22})$, for the entrance and exit fringe fields. The y-component of the magnetic field $B_y(x, 0, z)$ is given on the median plane.

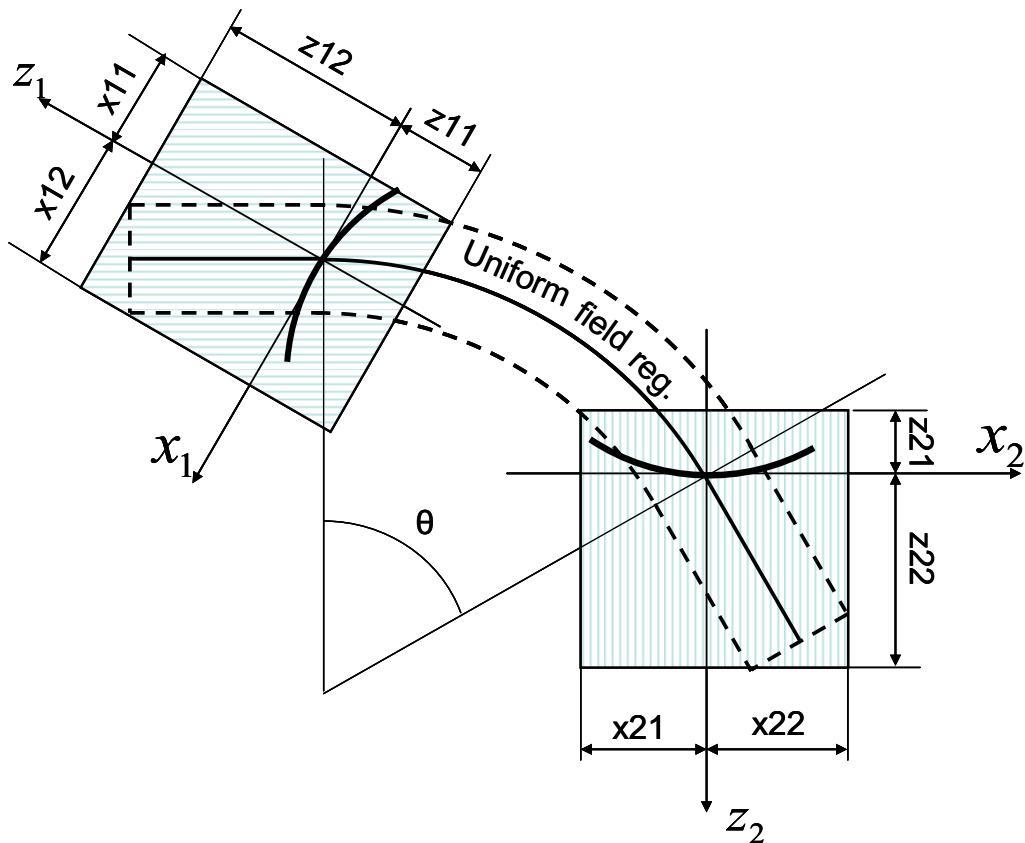


Fig. A5-4. Fringe fields extension in the median plane.

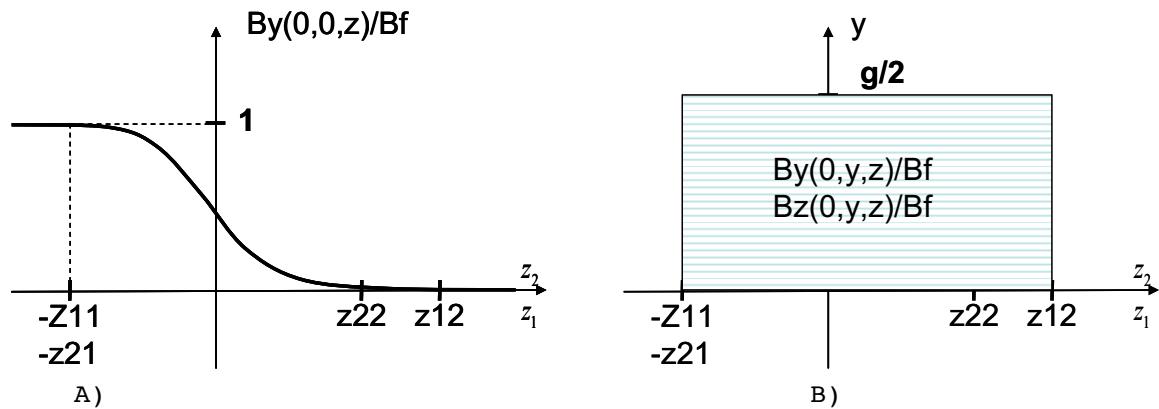


Fig. A5-5. Fringe fields extension in the $\{y, z\}$ plane.

References

- [A5-1] K.L. Brown, The ion optical program TRANSPORT. Technical Report 91, SLAC, 1979.
- [A5-2] K.R. Crandall, TRACE 3-D Documentation, Report LA-11054-MS, Los Alamos, 1987.
- [A5-3] H. Wollnik, J. Brezina and M. Berz, Nucl. Instr. and Meth. A 258 (1987) 408.
- [A5-4] S. Kowalsky and H.A. Enge, RAYTRACE, MIT Report, Cambridge, Massachusetts, July 1, 1987.
- [A5-5] H.A. Enge, Rev. of Sci. Instr., 34, 385 (1964)
- [A5-6] H.A. Enge, Rev. of Sci. Instr., 35, 278 (1964)

A5.1 Structure line for the element “rounded DIPOLE”

amu= 931.49432d3[keV/u]
 W is the kinetic energy per nucleon
 Q is the ion charge
 A is the ion mass
 $\gamma = 1 + W/\text{amu}$, $\beta = \sqrt{1 - 1/\gamma^2}$ $p[\text{GeV}/c] = 10^{-6} \cdot \text{amu} \cdot \beta \gamma / (Q/A)$ $B_f[kG] = \frac{0.3107131 \beta \gamma}{\rho[\text{cm}](Q/A)}$

0 bmag L ρ θ g w β₁ β₂ 1/ R₁ 1/R₂ nstep

The input file for the same element in the code **TRANSPORT**

```

' DIPOLE '
0
15. 1. 'mm'    0.1 ;
15. 5. 'mm'    0.1 ;
15. 6. 'P/10'   0.1 ;
1.    0. 0.    0. 0.    0. 0.    p ;
17.          '2ND' ;
16. 5. w/2 ;
16. 5. g/2 ;
16. 7. k1 ;
16. 8. 0. ;
16. 12. 1/R1 ;
16. 12. 1/R2 ;

20. 180. ; (if θ<0)
3. d=0.5(L- ρ|θ| ) ;
2. β1 ;
4. ρ|θ|, Bf  0.0           'BMAG' ;
2. β2 ;
3. d=0.5(L- ρ|θ| ) ;
20. -180. ; (if θ<0)
SENTINEL

```

Appendix 6. Creation of user defined initial distribution.

Frequently it is useful to start the simulation with the existing particle distribution, not from the initial distribution generated inside the code. A particular example is the simulation of the SC linac with the initial distribution obtained from TRACK or another code by simulation of the elements located upstream of the SC linac. For beam loss simulation on the SC linac it is preferable to start with the distribution formed in the RFQ. For this and similar cases we have developed a possibility to write a file with the information of total charge states nqtot, average energy Wtmp [keV/u], number of particles of each charge state npat(ird), charge states qq(ird) and 6 coordinates of each particle. The maximum size of the particle coordinates is usually 0:500000, however, it can be larger for specific cases. The particle phase csi(i,iq) is in radians at the fundamental frequency of the accelerator or front end that was used for generation of the distribution.

Example: In the front end of the RIA driver linac there is MHB operating at 28.75 MHz therefore the simulation starts with fundamental frequency 28.75 MHz. The particle distribution is written into the file after the RFQ which operates at 57.5 MHz. Before writing into the file the array is reduced to 28.75 MHz by dividing the array elements by harmonic number which is 2 in this case.

The following text is the excerpt from the FORTRAN program which shows format of the file.

```
if (iwrite_dis.eq.1) then
  open(1, file='read_dis.dat', status='unknown'
& ,access='SEQUENTIAL',form='UNFORMATTED')
  write(1) Wtmp,nqtot
  write(1) (npat(ird),ird=1,nqtot)
  write(1) (qq (ird),ird=1,nqtot)
  do iq = 1,nqtot
    do i = 0,npad(iq)
      write(1) x(i,iq) , xx(i,iq) , y(i,iq) , yy (i,iq),
&           csi(i,iq)/harm0, bb (i,iq) , spin(i,iq)
    enddo
  enddo
  close(1)
endif
```

Units:

x(i,iq)[cm], xx(i,iq)[rad], y(i,iq)[cm], yy(i,iq) [rad], csi(i,iq) [rad],
bb(i,iq) [relative velocity] , spin(i,iq) [1.d0 if particle is within the
acceptance of the following system, otherwise =-1d0]

To write a file with the particle distribution at the end of simulation the flag `iwrite_dis=1`.

Reading of the particle distribution will be done with the initial flag `iread_dis=1`.

By default these flags are equal to 0.

The following steps are necessary to provide correct simulation with the initial distribution which is read from the file `read_dis.dat`:

- 1) Make `track.dat` file assuming that the simulation starts with the internally generated distribution.

- 2) Be sure that files **track.dat** and **sclinac.dat** are compatible to start the simulation with fundamental frequency that was used for the phases in the file **read_dis.dat**.
- 3) In the "**INDEX**" section of the **track.dat** file set **iread_dis=1**.
- 4) Be sure that file **read_dis.dat** exists in the directory where all files related to the **TRACKv35.exe** code are located.
- 5) Check flag **isol**. For hard edge solenoids **isol=0**. Check parameter **current=0** for zero space charge calculations. Set **iflag_dis=1** for bunched beam structure, not for dc beam.

Appendix 7. A short manual for longitudinal corrections using TRACK

A correction section should be bracketed between two 'cdump' commands with the same index number 'n'. The index number 'n' refers to the number of the correction section; the first is 1, the second is 2, ...

An example of a correction section is given below

```
n cdump
..
..
.. here is the regular lattice
..
..
..
phcor i1 i2 r1 r2
n cdump
```

The command 'phcor' is the actual correction command, it should be bewteen two 'cdump' commands of the same index. The arguments of the command 'phcor' are:

```
i1: energy correction flag:
      i1=0 No correction;
      i1=1 correction using the reference charge state
average energy;
      i1=2 correction using the multiple charge state
beam average energy.
i2: phase correction flag:
      i2=0 No correction;
      i2=1 correction using the reference charge state
average phase;
      i2=2 correction using the multiple charge state
beam average phase.
r1: energy correction residual error in %
r2: phase correction residual error in deg
```