Contents

AGSMM : AGS main magnet

The AGS main magnet is a combined function dipole with straight axis (lines of constant field are straight lines).
The field computation routines for *AGSMM* are the same as for *MULTIPOL* (details in section **??**, page **??**), however *AGSMM* has the following four particularities :

- There are only three multipole components present in *AGSMM* : dipole, quadrupole and sextupole.
- The dipole field B_0 is drawn from the reference rigidity, $B\rho_{ref}$, and follows the latter so to preserve $\rho = B\rho_{ref}/B_0$ and the orbit deviation L/ρ. In particular,
  - in the absence of acceleration, $B\rho_{ref} \equiv BORO$, with $BORO$ the quantity appearing in the object definition using *[MC]OBJET*,
  - in presence of acceleration using *CAVITE*, $B\rho_{ref}$ is changed to $BORO \times D_{ref}$ at each passage in the cavity, with D_{ref} the relative synchronous momentum increase, a quantity that **zgoubi** updates at cavity traversal.
- The field indices, quadrupole $K1$ and sextupole $K2$, are derived from the reference rigidity, $B\rho_{ref}$, via momentum-dependent polynomials, taken from Ref. [**?**].
- The AGS main dipole has back-leg windings, used for instance for injection and extraction orbit bumps. The number of winding turns and the number of Ampere-turns are part of the data in the input data list. The intensity in the windings is accounted for in the conversion from total ampere-turns in the magnet to momentum and then to magnetic field.

Note : A consequence of items 2 and 3 is that no field value is required in defining the AGS main magnets in the zgoubi.dat input data list.

AGSMM	**AGS main magnet**		
IL	$IL = 1, 2[\times 10^n]$: print field and coordinates along trajectories.	0-2$[\times 10^n]$	I
MOD[.MOD2], dL, R_0, $dB1$, $dB2$, $dB3$	Type of magnet model [1] [type of back-leg winding model [2]] ; unused ; pole tip radius, 10 cm if set to zero ; relative error on dipole, quadrupole, sextupole component.	2*no dim., cm, 3*no dim.	I[.I], 5*E
NBLW, *NBLW times :* NW, I	Number of back-leg windings ; for each back-leg winding : number of windings, current.	$\leq$2, *NBLW*$\times$ (any, Amp.)	I, *NBLW*$\times$ (I, E)
	Entrance face		
X_E, λ_E, E_2, E_3	Integration zone ; fringe field extent : dipole fringe field extent = λ_E ; quadrupole fringe field extent = $\lambda_E * E_2$; sextuppole fringe field extent = $\lambda_E * E_3$ (sharp edge if field extent is zero)	2*cm, 2*no dim.	4*E
NCE, $C_0 - C_5$	same as *QUADRUPO*	0-6, 6*no dim.	I, 6*E
	Exit face		
X_S, λ_S, S_2, S_3	Integration zone ; as for entrance	2*cm, 2*no dim.	4*E
NCS, $C_0 - C_5$		0-6, 6*no dim.	I, 6*E
$R1$, $R2$, $R3$	Skew angles of field components	3*rad	10*E
XPAS	Integration step	cm	E
KPOS, *XCE*, *YCE*, *ALE*	*KPOS*=1 : element aligned, 2 : misaligned ; shifts, tilt (unused if *KPOS*=1). *KPOS* = 3 : effective only if $B1 \neq 0$: entrance and exit frames are shifted by *YCE* and tilted *wrt.* the magnet by an angle of • either ALE if ALE$\neq$0 • or 2 Arcsin$(B1\,XL\,/\,2BORO)$ if ALE=0 *KPOS* = 4 : same as *KPOS* = 3 however with possible X- or Y- or Z-misalignment or -rotation (under development)	1-4, 2*cm, rad	I, 3*E

[1] *MOD=1* : centered multipole model ; *MOD=2* : long-shifted dipole model ; *MOD=3* : short-shifted dipole model.

[2] *MOD2 = 0* (default) : user defined back-leg windings (defined in routine `agsblw.f`) ; *MOD2 = 1* : actual AGS data are taken, namely : MM_A16AD : NBLW = 1, SIGN = 1.D0, NW = 10 ; MM_A17CF : NBLW = 1, SIGN = 1.D0, NW = 10 ; MM_A18CF : NBLW = 1, SIGN = -1.D0, NW = 10 ; MM_A19BD : NBLW = 1, SIGN = -1.D0, NW = 12 ; MM_A20BD : NBLW = 1, SIGN = 1.D0, NW = 12 ; MM_B02BF : NBLW = 2, SIGN = 1.D0, NW1 = 12, SIGN = 1.D0, NW2 = 6 ; MM_B03CD : NBLW = 1, SIGN = 1.D0, NW = 10 ; MM_B04CD : NBLW = 1, SIGN = -1.D0, NW = 10 ; MM_B05A : NBLW = 1, SIGN = -1.D0, NW = 10 ; MM_K19BD : NBLW = 1, SIGN = 1.D0, NW = 6 ; MM_K20B : NBLW = 1, SIGN = 1.D0, NW = 6 ; MM_L13CF : NBLW = 1, SIGN = -1.D0, NW = 5 ; MM_L14C : NBLW = 1, SIGN = -1.D0, NW = 5 ; MM_A07CD : NBLW = 1, SIGN = -1.D0, NW = 5 ; MM_A08C : NBLW = 1, SIGN = -1.D0, NW = 5 ; MM_B01B : NBLW = 1, SIGN = 1.D0, NW = 6 ; MM_L06A : NBLW = 1, SIGN = 1.D0, NW = 5 ; MM_L07C : NBLW = 1, SIGN = 1.D0, NW = 5 ; MM_A14C : NBLW = 1, SIGN = -1.D0, NW = 5 ; MM_A15A : NBLW = 1, SIGN = -1.D0, NW = 5 ; MM_E06A : NBLW = 1, SIGN = -1.D0, NW = 5 ; MM_E07CD : NBLW = 1, SIGN = -1.D0, NW = 5 ; MM_E20BD : NBLW = 1, SIGN = 1.D0, NW = 6 ; MM_F01BF : NBLW = 1, SIGN = 1.D0, NW = 6 ; MM_F14CF : NBLW = 1, SIGN = 1.D0, NW = 5 ; MM_F15AD : NBLW = 1, SIGN = 1.D0, NW = 5 ; MM_G08CD : NBLW = 1, SIGN = -1.D0, NW = 5 ; MM_G09BF : NBLW = 1, SIGN = -1.D0, NW = 6.
MOD2 = 1 : User defined - implementation to be completed.

AGSQUAD : AGS quadrupole

The AGS quadrupoles are regular quadrupoles. The simulation of *AGSQUAD* uses the same field modelling as *MULTIPOL*, section **??**, page **??**. However amperes are provided as input to *AGSQUAD* rather than fields, the reason being that some of the AGS quadrupoles have two superimposed coil circuits, with separate power supplies. It has been dealt with this particularity by allowing for an additional set of multipole data in *AGSQUAD*, compared to *MULTIPOL*.

The field in *AGSQUAD* is computed using transfer functions from the ampere-turns in the coils to magnetic field that account for the non-linearity of the magnetic permeability [**?**].

AGSQUAD	**AGS quadrupole**		
IL	$IL = 1, 2[\times 10^n]$: print field and coordinates along trajectories.	0-2$[\times 10^n]$	I
XL, R_0, $IW1$, $IW2$, $IW3$, $dIW1$, $dIW2$, $dIW3$	Length of element ; radius at pole tip ; current in windings ; relative error on currents.	2*cm,3*A 3*no dim	5*E 3*E
	Entrance face		
X_E, λ_E	Integration zone ; fringe field extent. (sharp edge if field extent is zero)	2*cm,9*no dim.	11*E
NCE, $C_0 - C_5$	Same as *QUADRUPO*	0-6, 6*no dim.	I, 6*E
	Exit face		
X_S, λ_S	Integration zone ; as for entrance	2*cm, 9*no dim.	11*E
NCS, $C_0 - C_5$		0-6, 6*no dim.	I, 6*E
$R1$	Roll angle	10*rad	10*E
XPAS	Integration step	cm	E
KPOS, *XCE*, *YCE, ALE*	*KPOS*=1 : element aligned, 2 : misaligned ; shifts, tilt (unused if *KPOS*=1).	1-2, 2*cm, rad	I, 3*E

AIMANT : Generation of dipole mid-plane 2-D map, polar frame

The keyword *AIMANT* provides an automatic generation of a dipole median plane field map in polar coordinates.

A more recent and improved version will be found in *DIPOLE-M*. In addition, a similar modelling, that however skips the stage of an intermediate mid-plane field map, can be found in *DIPOLE[S]*.

The extent of the map is defined by the following parameters, as shown in Figs. 1A and 1B,

AT	:	total angular aperture
RM	:	mean radius used for the positioning of field boundaries
RMIN, RMAX	:	minimum and maximum radial boundaries of the map

The 2 or 3 effective field boundaries (EFB) inside the map are defined from geometric boundaries, the shape and position of which are determined by the following parameters,

ACENT	:	arbitrary angle, used for the positioning of the EFB's.
ω	:	azimuth of an EFB with respect to *ACENT*
θ	:	angle of a boundary with respect to its azimuth (wedge angle)
R_1, R_2	:	radius of curvature of an EFB
U_1, U_2	:	extent of the linear part of the EFB.

At any node of the map mesh, the value of the Z component of the field is calculated as

$$B_Z = \mathcal{F}(R,\theta) * B_0 * \left(1 + N * \left(\frac{R-RM}{RM}\right) + B * \left(\frac{R-RM}{RM}\right)^2 + G * \left(\frac{R-RM}{RM}\right)^3\right) \tag{1}$$

where N, B and G are respectively the first, second and third order field indices and $\mathcal{F}(R,\theta)$ is the fringe field coefficient (it determines the "flutter" in periodic structures).

Calculation of the Fringe Field Coefficient

With each EFB a realistic extent of the fringe field, λ, is associated (Figs. 1A and 1B), and a fringe field coefficient F is calculated. In the following λ stands for either λ_E (Entrance), λ_S (Exit) or λ_L (Lateral EFB).
If a node of the map mesh is at a distance of the EFB larger than λ, then $F = 0$ outside the field map and $F = 1$ inside. If a node is inside the fringe field zone, then F is calculated as follows.

Two options are available, for the calculation of F, depending on the value of ξ.

If $\xi \geq 0$, F is a second order type fringe field (Fig. 2) given by

$$F = \frac{1}{2}\frac{(\lambda - s)^2}{\lambda^2 - \xi^2} \qquad \text{if } \xi \leq s \leq \lambda \tag{2}$$

$$F = 1 - \frac{1}{2}\frac{(\lambda - s)^2}{\lambda^2 - \xi^2} \qquad \text{if } -\lambda \leq s \leq -\xi \tag{3}$$

where s is the distance to the EFB, and

$$F = \frac{1}{2} + \frac{s}{\lambda + \xi} \qquad \text{if } 0 \leq s \leq \xi \tag{4}$$

$$F = \frac{1}{2} - \frac{s}{\lambda + \xi} \qquad \text{if } -\xi \leq s \leq 0 \tag{5}$$

This simple model allows a rapid calculation of the fringe field, but may lead to erratic behavior of the field when extrapolating out of the median plane, due to the discontinuity of d^2B/ds^2, at $s = \pm\xi$ and $s = \pm\lambda$. For better accuracy it is advised to use the next option.

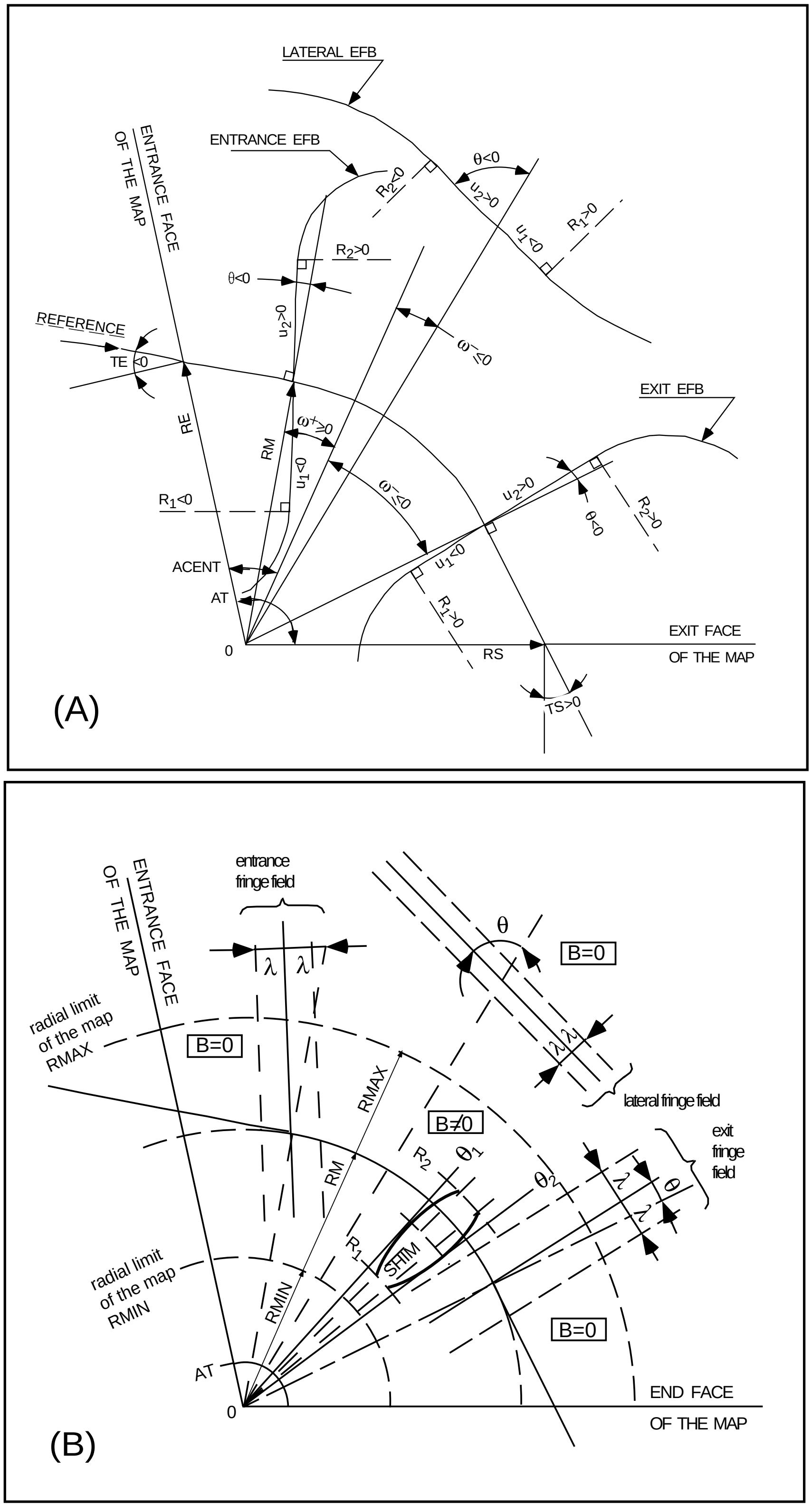


Figure 1: A : Parameters used to define the field map and geometrical boundaries.
B : Parameters used to define the field map and fringe fields.

If $\xi = -\mathbf{1}$, F is an exponential type fringe field (Fig. 2) given by [**?**]

$$F = \frac{1}{1 + \exp P(s)} \tag{6}$$

where s is the distance to the EFB, and

$$P(s) = C_0 + C_1\left(\frac{s}{\lambda}\right) + C_2\left(\frac{s}{\lambda}\right)^2 + C_3\left(\frac{s}{\lambda}\right)^3 + C_4\left(\frac{s}{\lambda}\right)^4 + C_5\left(\frac{s}{\lambda}\right)^5 \tag{7}$$

The values of the coefficients C_0 to C_5 should be such that the derivatives of B_Z with respect to s be negligible at $s = \pm\lambda$, so as not to perturb the extrapolation of $\vec{B}$ out of the median plane.
It is also possible to simulate a shift of the EFB, by giving a non zero value to the parameter *shift*. s is then changed to $s-$ *shift* in the previous equation. This allows small variations of the total magnetic length.
Let F_E (respectively F_S, F_L) be the fringe field coefficient attached to the entrance (respectively exit, lateral) EFB following the equations above. At any node of the map mesh, the resulting value of the fringe field coefficient (eq. 1) is (Fig. 3)

$$\mathcal{F}(R,\theta) = F_E * F_S * F_L$$

In particular, $F_L \equiv 1$ if no lateral EFB is requested.

The Mesh of the Field Map

The magnetic field is calculated at the nodes of a mesh with polar coordinates, in the median plane. The radial step is given by

$$\delta R = \frac{\textit{RMAX - RMIN}}{\textit{IRMAX} - 1}$$

and the angular step by

$$\delta\theta = \frac{AT}{\textit{IAMAX} - 1}$$

where, *RMIN* and *RMAX* are the lower and upper radial limits of the field map, and AT is its total angular aperture (Fig. 1B). *IRMAX* and *IAMAX* are the total number of nodes in the radial and angular directions.

Simulating Field Defects and Shims

Once the initial map is calculated, it is possible to perturb it by means of the parameter *NBS*, so as to simulate field defects or shims.

If NBS $= -\mathbf{2}$, the map is globally modified by a perturbation proportional to $R - R_0$, where R_0 is an arbitrary radius, with an amplitude $\Delta B_Z/B_0$, so that B_Z at the nodes of the mesh is replaced by

$$B_Z * \left(1 + \frac{\Delta B_Z}{B_0}\,\frac{R - R_0}{\textit{RMAX} - \textit{RMIN}}\right)$$

If NBS $= -\mathbf{1}$, the perturbation is proportional to $\theta - \theta_0$, and B_Z is replaced by

$$B_Z * \left(1 + \frac{\Delta B_Z}{B_0}\,\frac{\theta - \theta_0}{AT}\right)$$

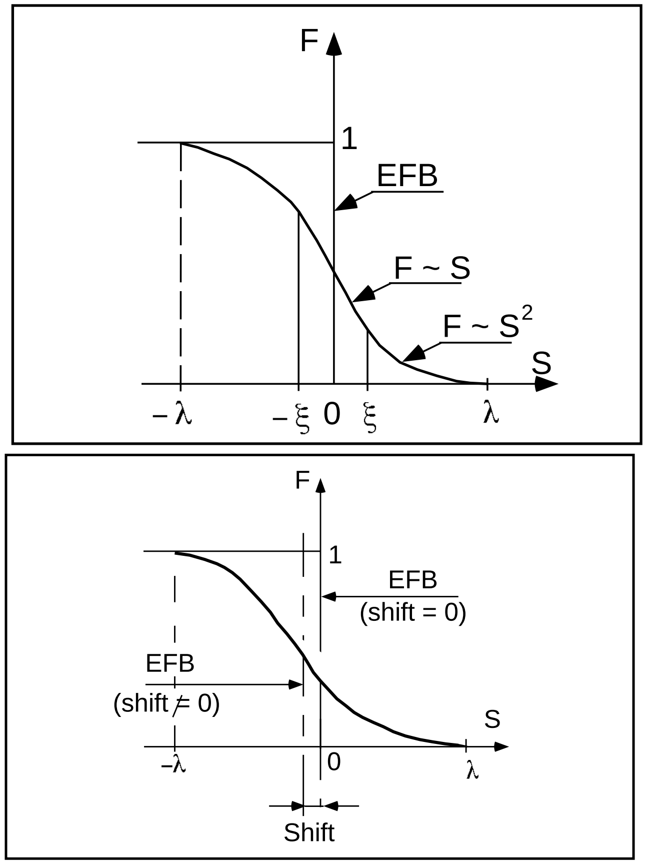


Figure 2: Second order type fringe field (upper plot) and exponential type fringe field (lower plot).

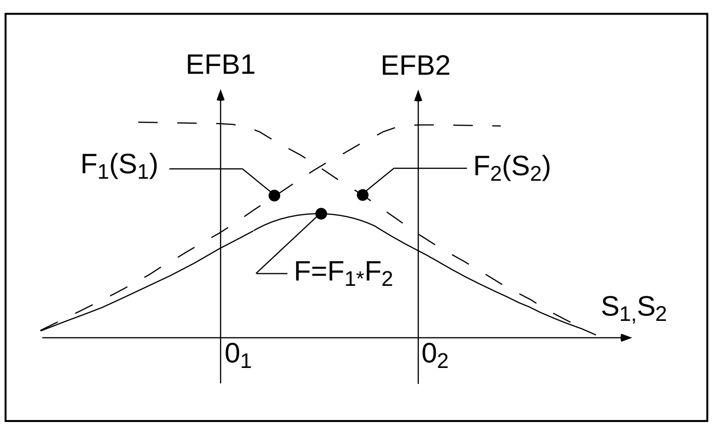


Figure 3: Effective value of $\mathcal{F}(R,\theta)$ for overlapping fringe fields F_1 and F_2 centered at O_1 and O_2.

If NBS $\geq$ 1, then *NBS* shims are introduced at positions $\frac{R_1+R_2}{2}$, $\frac{\theta_1+\theta_2}{2}$ (Fig. 4) [**?**]
The initial field map is modified by shims with second order profiles given by

$$\theta = \left(\gamma + \frac{\alpha}{\mu}\right) \beta \frac{X^2}{\rho^2}$$

where X is shown in Fig. 4, $\rho = \frac{R_1+R_2}{2}$ is the central radius, α and γ are the angular limits of the shim, β and μ are parameters.
At each shim, the value of B_Z at any node of the initial map is replaced by

$$B_Z * \left(1 + F\theta * FR * \frac{\Delta B_Z}{B_0}\right)$$

where $F\theta = 0$ or $FR = 0$ outside the shim, and $F\theta = 1$ and $FR = 1$ inside.

Extrapolation Off Median Plane

The vertical field $\vec{B}$ and its derivatives in the median plane are calculated by means of a second or fourth order polynomial interpolation, depending on the value of the parameter *IORDRE* (*IORDRE*=2, 25 or 4, see section **??**). The transformation from polar to Cartesian coordinates is performed following eqs. (**??** or **??**). Extrapolation off median plane is then performed by means of Taylor expansions following the procedure described in section **??**.

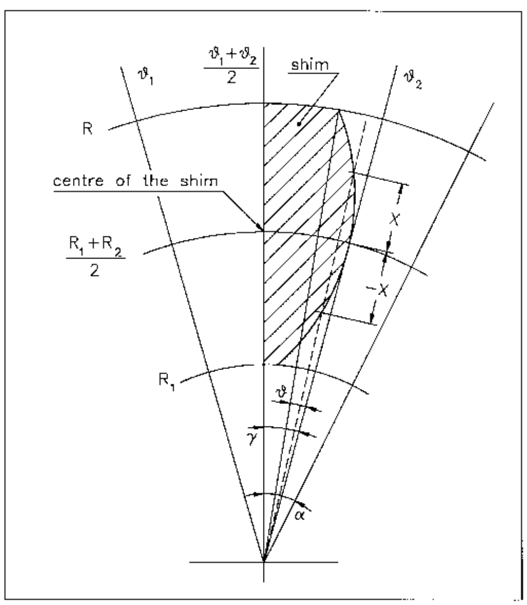


Figure 4: A second order profile shim. The shim is centered at $\frac{(R_1+R_2)}{2}$ and $\frac{(\theta_1+\theta_2)}{2}$.

AIMANT **Generation of dipole mid-plane 2-D map, polar frame**

$$B_Z = \mathcal{F} B_0 \left(1 - N\left(\frac{R-RM}{RM}\right) + B\left(\frac{R-RM}{RM}\right)^2 + G\left(\frac{R-RM}{RM}\right)^3\right)$$

NFACE, IC, IL	Number of field boundaries $IC = 1, 2$: print field map $IL = 1, 2[\times 10^n]$: print field and coordinates along trajectories.	2-3, 0-2, 0-2$[\times 10^n]$	3*I
IAMAX, IRMAX	Azimuthal and radial number of nodes of the mesh	≤ 400, $\leq 10^4$	2*I
B_0, *N*, *B*, *G*	Field and field indices	kG, 3*no dim.	4*E
AT, ACENT, RM, RMIN, RMAX	Mesh parameters : total angle of the map ; azimuth for EFBs positioning ; reference radius ; minimum and maximum radii	2*deg, 3*cm	5*E
	ENTRANCE FIELD BOUNDARY		
λ, ξ	Fringe field extent (normally $\simeq$ gap size) ; flag : - if $\xi \geq 0$: second order type fringe field with linear variation over distance ξ - if $\xi = -1$: exponential type fringe field : $F = (1 + \exp(P(s)))^{-1}$ $P(s) = C_0 + C_1(\frac{s}{\lambda}) + C_2(\frac{s}{\lambda})^2 + ... + C_5(\frac{s}{\lambda})^5$	cm, (cm)	2*E
NC, $C_0 - C_5$, *shift*	NC = 1 + degree of $P(s)$; C_0 to C_5 : see above ; EFB shift (ineffective if $\xi \geq 0$)	0-6, 6*no dim., cm	I, 7*E
ω^+, θ, R_1, U_1, U_2, R_2	Azimuth of entrance EFB with respect to *ACENT* ; wedge angle of EFB ; radii and linear extents of EFB (use $\mid U_{1,2} \mid = \infty$ when $R_{1,2} = \infty$) (Note : $\lambda = 0$, ω^+ = ACENT and $\theta = 0$ for sharp edge)	2*deg, 4*cm	6*E
	EXIT FIELD BOUNDARY (See ENTRANCE FIELD BOUNDARY)		
λ, ξ	Fringe field parameters	cm, (cm)	2*E
NC, $C_0 - C_5$, *shift*		0-6, 6*no dim., cm	1, 7*E
ω^-, θ, R_1, U_1, U_2, R_2	Positioning and shape of the exit EFB (Note : $\lambda = 0$, $\omega^- =$-AT+ACENT and $\theta = 0$ for sharp edge)	2*deg, 4*cm	6*E

If NFACE = 3	LATERAL FIELD BOUNDARY (See ENTRANCE FIELD BOUNDARY) Next 3 records *only* if *NFACE* = 3		
λ, ξ	Fringe field parameters	cm, (cm)	2*E
NC, $C_0 - C_5$, *shift*		0-6, 6*no dim., cm	I, 7*E
ω^-, θ, R_1, U_1, U_2, R_2, $RM3$	Positioning and shape of the lateral EFB ; RM3 is the radial position on azimuth *ACENT*	2*deg, 5*cm	7*E
NBS	Option index for perturbations to the field map	-2 $-$ 0 or ≥ 1	I
If NBS = 0	Normal value. No other record required		
If NBS = -2	The map is modified as follows :		
R_0, $\Delta B/B_0$	B transforms to $B * \left(1 + \frac{\Delta B}{B_0}\frac{R-R_0}{RMAX-RMIN}\right)$	cm, no dim.	2*E
If NBS = -1	the map is modified as follows :		
θ_0, $\Delta B/B_0$	B transforms to $B * \left(1 + \frac{\Delta B}{B_0}\frac{\theta-\theta_0}{AT}\right)$	deg, no dim.	2*E
If NBS ≥ 1	Introduction of NBS shims		
For I = 1, NBS	The following 2 records must be repeated NBS times		
R_1, R_2, θ_1, θ_2, λ	Radial and angular limits of the shim ; λ is unused	2*cm, 2*deg, cm	5*E
γ, α, μ, β	geometrical parameters of the shim	2*deg, 2*no dim.	4*E
IORDRE	Degree of interpolation polynomial : 2 = second degree, 9-point grid 25 = second degree, 25-point grid 4 = fourth degree, 25-point grid	2, 25 or 4	I
XPAS	Integration step	cm	E
KPOS	Positioning of the map, normally 2. Two options :	1-2	I
If KPOS = 2 RE, TE, RS, TS	Positioning as follows : Radius and angle of reference, respectively, at entrance and exit of the map.	cm, rad, cm, rad	4*E
If KPOS = 1 DP	Automatic positioning of the map, by means of reference relative momentum	no dim.	E

AUTOREF : Automatic transformation to a new reference frame

AUTOREF positions the new reference frame following 3 options :

If $\mathbf{I=1}$, *AUTOREF* is equivalent to

$$\textit{CHANGREF}[XCE=0, YCE=Y(1), ALE=T(1)]$$

so that the new reference frame is at the exit of the last element, with particle 1 at the origin with its horizontal angle set to $T=0$.

If $\mathbf{I=2}$, it is equivalent to

$$\textit{CHANGREF}[XW, YW, T(1)]$$

so that the new reference frame is at the position (XW, YW) of the waist (calculated automatically in the same way as for *IMAGE*) of the three rays number 1, 4 and 5 (compatible for instance with *OBJET*, *KOBJ* = 5, 6, together with the use of *MATRIX*) while $T(1)$, the horizontal angle of particle number $I1$, is set to zero.

If $\mathbf{I=3}$, it is equivalent to

$$\textit{CHANGREF}[XW, YW, T(I1)]$$

so that the new reference frame is at the position (XW, YW) of the waist (calculated automatically in the same way as for *IMAGE*) of the three rays number I1, I2 and I3 specified as data, while $T(I1)$ is set to zero.

AUTOREF **Automatic transformation to a new reference frame**

I	1 : Equivalent to *CHANGREF* ($XCE = 0, YCE = Y(1), ALE = T(1)$)	1-2	I
	2 : Equivalent to *CHANGREF* ($XW, YW, T(1)$), with (XW, YW) being the location of the intersection (waist) of particles 1, 4 and 5 (useful with *MATRIX*, for automatic positioning of the first order focus)		
	3 : Equivalent to *CHANGREF* ($XW, YW, T(I1)$), with (XW, YW) being the location of the intersection (waist) of particles $I1$, $I2$ and $I3$ (for instance : $I1$ = central trajectory, $I2$ and $I3$ = paraxial trajectories that intersect at the first order focus)		
If $I = 3$	Next record only if $I = 3$		
$I1$, $I2$, $I3$	Three particle numbers	3*(1-*IMAX*)	3*I

BEAMBEAM : Beam-beam lens

BEAMBEAM is a beam-beam lens simulation, a point transform [**?**].

Upon option using *SPNTRK*, *BEAMBEAM* will include spin kicks, after modelling as described in Ref. [**?**].

BEAMBEAM	**Beam-beam lens**		
SW, I	0/1 : off/on ; beam intensity. Use *SPNTRK* to activate spin kicks.	0-2, Amp	I, E
$\alpha_Y,\ \beta_Y,\ \epsilon_{Y,norm}/\pi$	Beam parameters, horizontal.	- , m, m.rad	3*E
$\alpha_Z,\ \beta_Z,\ \epsilon_{Z,norm}/\pi$	Beam parameters, vertical.	- , m, m.rad	3*E
$\sigma_X,\ \sigma_{dp/p}$	*rms* bunch length ; *rms* momentum spread.	m, -	2*E
$\mathcal{C},\ \alpha$	Ring circumference ; momentum compaction.	m, -	2*E
$Q_Y,\ Q_Z,\ Q_s$	Tunes, horizontal, vertical, synchrotron.	-, -, -	3*E
$A_Y,\ A_Z,\ A_X$	Amplitudes, horizontal, vertical, longitudinal.	-, -, -	3*E

BEND : Bending magnet, Cartesian frame

BEND is one of several keywords available for the simulation of dipole magnets. It presents the interest of easy handling, and is well adapted for the simulation of synchrotron dipoles and such other regular dipoles as sector magnets with wedge angles.

The field in *BEND* is defined in a Cartesian coordinate frame (unlike for instance *DIPOLE[S]* that uses a polar frame). As a consequence, having particle coordinates at entrance or exit of the magnet referring to the curved main direction of motion may require using *KPOS*, in particular *KPOS=3* (in a circular machine cell for instance, see section **??**, p. **??**).

The dipole simulation accounts for the magnet geometrical length *XL*, for a possible skew angle (X-rotation, useful for obtaining vertical deviation magnet), and for the field $B1$ such that in absence of fringe field the deviation θ satisfies $XL = 2\frac{BORO}{B1}\sin\theta/2$.

Then follows the description of the entrance and exit EFB's and fringe fields. The wedge angles W_E (entrance) and W_S (exit) are defined with respect to the sector angle, with the signs as described in Fig. 1. Within a distance $\pm X_E(\pm X_S)$ on both sides of the entrance (exit) EFB, the fringe field model is used (same as for *QUADRUPO*, Fig. **??**, p. **??**) ; elsewhere, the field is supposed to be uniform.

If λ_E (resp. λ_S) is zero sharp edge field model is assumed at entrance (resp. exit) of the magnet and X_E (resp. X_S) is forced to zero. In this case, the wedge angle vertical first order focusing effect (if $\vec{B}1$ is non zero) is simulated at magnet entrance and exit by a kick $P_2 = P_1 - Z_1\tan(\epsilon/\rho)$ applied to each particle (P_1, P_2 are the vertical angles upstream and downstream the EFB, Z_1 the vertical particle position at the EFB, ρ the local horizontal bending radius and ϵ the wedge angle experienced by the particle ; ϵ depends on the horizontal angle T).

Magnet (mis-)alignment is assured by *KPOS*. *KPOS* also allows some degrees of automatic alignment useful for periodic structures (section **??**).

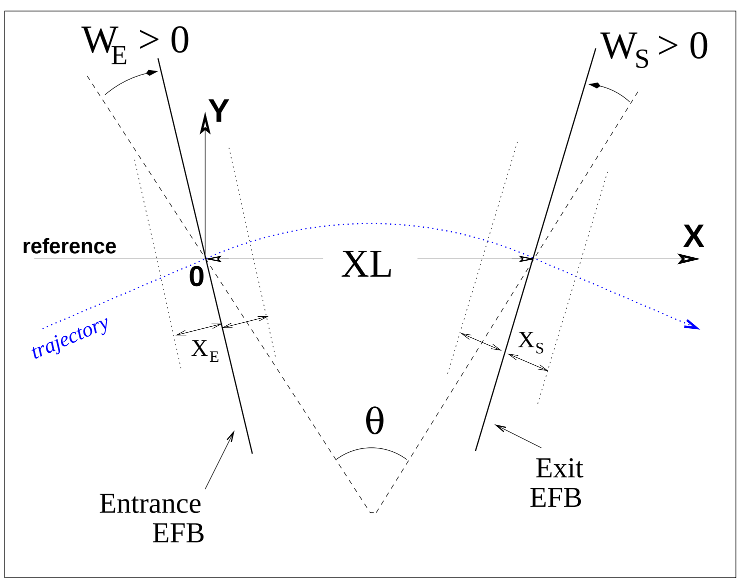
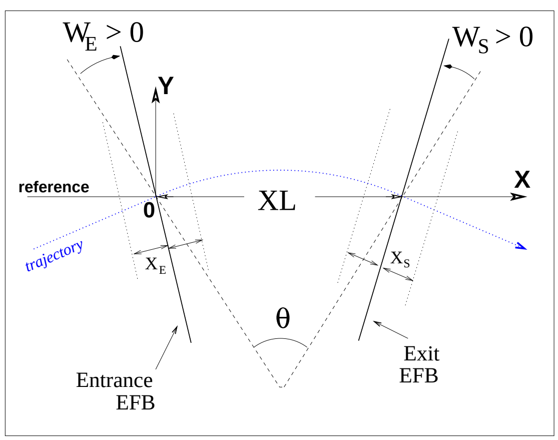


Figure 1: Geometry and parameters of *BEND* : *XL* = length, θ = deviation, W_E, W_S are the entrance and exit wedge angles. The motion is computed in the Cartesian frame (O, X, Y, Z)

BEND **Bending magnet, Cartesian frame**

IL	$IL = 1, 2[\times 10^n]$: print field and coordinates along trajectories.	0-2$[\times 10^n]$	I
XL, Sk, $B1$	Length ; skew angle ; field	cm, rad, kG	3*E
	Entrance face :		
X_E, λ_E, W_E	Integration zone extent ; fringe field extent (normally $\simeq$ gap height ; zero for sharp edge) ; wedge angle	cm, cm, rad	3*E
N, C_0–C_5	Unused ; fringe field coefficients : $B(s) = B1\,F(s)$ with $F(s) = 1/(1+\exp(P(s))$ and $P(s) = \sum_{i=0}^{5} C_i(s/\lambda)^i$	unused, 6*no dim.	I, 6*E
	Exit face :		
X_S, λ_S, W_S	See entrance face	cm, cm, rad	3*E
N, C_0–C_5		unused, 6*no dim.	I, 6*E
XPAS	Integration step	cm	E
KPOS, XCE, YCE, ALE	*KPOS*=1 : element aligned, 2 : misaligned ; shifts, tilt (unused if *KPOS*=1) *KPOS* = 3 : entrance and exit frames are shifted by *YCE* and tilted *wrt.* the magnet by an angle of • either ALE if ALE≠0 • or 2 Arcsin(*B1XL* / 2*BORO*) if ALE=0	1-2, 2*cm, rad	I, 3*E



Geometry and parameters of *BEND* : *XL* = length, θ = deviation, W_E, W_S are the entrance and exit wedge angles. The motion is computed in the Cartesian frame (O, X, Y, Z)

BINARY : *BINARY/FORMATTED* data converter

This procedure translates field map data files from “BINARY” to “FORMATTED” – in the *FORTRAN* sense, or the other way.

The keyword is followed by, next data line,

$$NF[.J], NCOL, NHDR$$

the number of files to be translated [`READ` format option, a single digit integer, optional], number of data columns in the file, number of header lines in the file.

If J is not given, the $NCOL$ arrangement should be consistent with the following *FORTRAN* READ statement :

```
READ (unit=ln,*) (X7(I),I=1,NCOL)
```

If $J = 1$, NCOL should be consistent with the following *FORTRAN* READ statement :

```
READ (unit=ln,fmt='(1x,ncol*E11.2)') (X7(I),I=1,NCOL)
```

Then follow, line by line, the NF names of the files to be translated.

Iff a file name begins with the prefix “B_” or “b_”, it is assumed “binary”, and hence converted to “formatted”, and given the same name after suppression of the prefix “B_” or “b_”. Conversely, *iff* the file name does not begin with “B_” or “b_”, the file is presumed “formatted” and hence translated to “binary”, and is given the same name after addition of the prefix “b_”.

In its present state, the procedure *BINARY* only supports a limited number of read/write formats. Details concerning I/O formatting can be found in the *FORTRAN* file `'binary.f'`.

BINARY ***BINARY/FORMATTED*** **data converter**

$NF[.J],\ NCol,\ NHDR$	Number of files to convert [READ format type, see below], of data columns, of header lines.	$\leq 20, \geq 1$, $0-9$	3*I1
The next NF **lines :**			
FNAME	Name of the file to be converted. File content is assumed binary *iff* name begins with "B_" or "b_", assumed formatted otherwise.		A80

READ format :

If FRM not given Format is '*'
If FRM=1 Format is '1X,7E11.*'

BREVOL : 1-D uniform mesh magnetic field map

BREVOL reads a 1-D axial field map from a storage data file, whose content must match the following *FORTRAN* reading sequence (possible FORMAT updates are to be found in `fmapw.f`).

```
OPEN (UNIT = NL, FILE = FNAME, STATUS = 'OLD' [,FORM='UNFORMATTED'])
DO 1 I = 1, IX
IF (BINARY) THEN
READ(NL) X(I), BX(I)
ELSE
READ(NL,*) X(I), BX(I)
ENDIF
1     CONTINUE
```

where *IX* is the number of nodes along the (symmetry) X-axis, $X(I)$ their coordinates, and *BX*(I) are the values of the X component of the field. *BX* is normalized with *BNORM* factor prior to ray-tracing, as well X is normalized with the coefficient *XNORM* (useful to convert to centimeters, the working units in **zgoubi**). For binary files, *FNAME* must begin with 'B_' or 'b_', a flag 'BINARY' will thus be set to '.TRUE.' by the *FORTRAN*.

X-cylindrical symmetry is assumed, resulting in BY and BZ taken to be zero on axis. $\vec{B}(X, Y, Z)$ and its derivatives along a particle trajectory are calculated by means of a 5-point polynomial interpolation followed by second order off-axis extrapolation (see sections **??**, **??**).

Entrance and/or exit integration boundaries may be defined in the same way as in *CARTEMES* by means of the flag ID and coefficients A, B, C, etc.

BREVOL	**1-D uniform mesh magnetic field map** X-axis cylindrical symmetry is assumed		
IC, *IL*	$IC = 1, 2$: print the map $IL = 1, 2[\times 10^n]$: print field and coordinates along trajectories.	0-2, 0-2$[\times 10^n]$	2*I
BNORM, XN	Field and X-coordinate normalization coeff.	2*no dim.	2*E
TITL	Title. Start with “FLIP” to get field map X-flipped.		A80
IX	Number of longitudinal nodes of the map	≤ 400	I
FNAME [, SUM] [1, 2]	File name		A80
ID, A, B, C [, A', B', C', B'', etc., if $ID \geq 2$]	Integration boundary. Ineffective when $ID = 0$. ID = -1, 1 or ≥ 2 : as for *CARTEMES*	≥ -1, 2*no dim., cm [,2*no dim., cm, etc.]	I,3*E [,3*E,etc.]
IORDRE	Unused	2, 25 or 4	I
XPAS	Integration step	cm	E
KPOS, *XCE*, *YCE, ALE*	*KPOS*=1 : element aligned, 2 : misaligned ; shifts, tilt (unused if *KPOS*=1)	1-2, 2*cm, rad	I, 3*E

[1] *FNAME* (*e.g.*, solenoid.map) contains the field data. These must be formatted according to the following *FORTRAN* sequence :

```
OPEN (UNIT = NL, FILE = FNAME, STATUS = ‘OLD’ [,FORM=’UNFORMATTED’])
DO 1 I = 1, IX
IF (BINARY) THEN
READ(NL) X(I), BX(I)
ELSE
READ(NL,*) X(I), BX(I)
ENDIF
1     CONTINUE
```

where $X(I)$ and $BX(I)$ are the longitudinal coordinate and field component at node (I) of the mesh. Binary file names must begin with *FNAME* ’B_’ or ’b_’. ‘Binary’ will then automatically be set to ‘.TRUE.’.

[2] Sumperimposing (summing) field maps is possible. To do so, pile up file names with ’SUM’ following each name but the last one. *e.g.*, in the following example, 3 field maps are read and summed :

```
myMapFile1  SUM
myMapFile2  SUM
myMapFile3
```

(all maps must all have their mesh defined in identical coordinate frame).

CARTEMES : 2-D Cartesian uniform mesh magnetic field map

CARTEMES was originally dedicated to the reading and processing of the measured median plane field maps of the QDD spectrometer SPES2 at Saclay, assuming mid-plane dipole symmetry. However, it can be used for the reading of any 2-D median plane maps, provided that the format of the field data storage file fits the following *FORTRAN* sequence

```
OPEN (UNIT = NL, FILE = FNAME, STATUS = 'OLD' [,FORM='UNFORMATTED'])
IF (BINARY) THEN
READ(NL) (Y(J), J=1, JY)
ELSE
READ(NL,FMT='(10F8.2)') (Y(J), J=1, JY)
ENDIF
DO 1 I=1, IX
IF (BINARY) THEN
READ(NL) X(I), (BMES(I,J), J=1, JY)
ELSE
READ(NL,FMT='(10F8.1)') X(I), (BMES(I,J), J=1, JY)
ENDIF
1     CONTINUE
```

where, IX and JY are the number of longitudinal and transverse horizontal nodes of the uniform mesh, and $X(I)$, $Y(J)$ their coordinates. *FNAME* is the name of the file containing the field data. For binary files, *FNAME* must begin with 'B_' or 'b_', a flag 'BINARY' will thus be set to '.TRUE.' by the *FORTRAN*.

The measured field *BMES* is normalized with *BNORM*,

$$B(I, J) = \textit{BMES}(I, J) \times \textit{BNORM}$$

As well the longitudinal coordinate X is normalized with a *XNORM* coefficient (useful to convert to centimeters, the working units in **zgoubi**).
The vector field, $\vec{B}$, and its derivatives out of the median plane are calculated by means of a second or fourth order polynomial interpolation, depending on the value of the parameter *IORDRE* (*IORDRE* = 2, 25 or 4, see section **??**).

In case a particle exits the mesh, its *IEX* flag is set to -1 (see section **??**, p. **??**), however it is still tracked with the field being *extrapolated* from the closest nodes of the mesh. Note that such extrapolation process may induce erratic behavior if the distance from the mesh gets too large.

Entrance and/or exit integration boundaries can be defined with the flag ID, as follows (Fig. 1).

If ID $=$ **1 :** the integration in the field is terminated on a boundary with equation $A'X + B'Y + C' = 0$, and then the trajectories are extrapolated linearly onto the exit border of the map.

If ID $= -$**1 :** an entrance boundary is defined, with equation $A'X + B'Y + C' = 0$, up to which trajectories are first extrapolated linearly from the map entrance border, prior to being integrated in the field.

If ID $\geq$ **2 :** one entrance boundary, and $ID - 1$ exit boundaries are defined, as above. The integration in the field terminates on the last $(ID - 1)$ exit boundary. No extrapolation onto the map exit border is performed in this case.

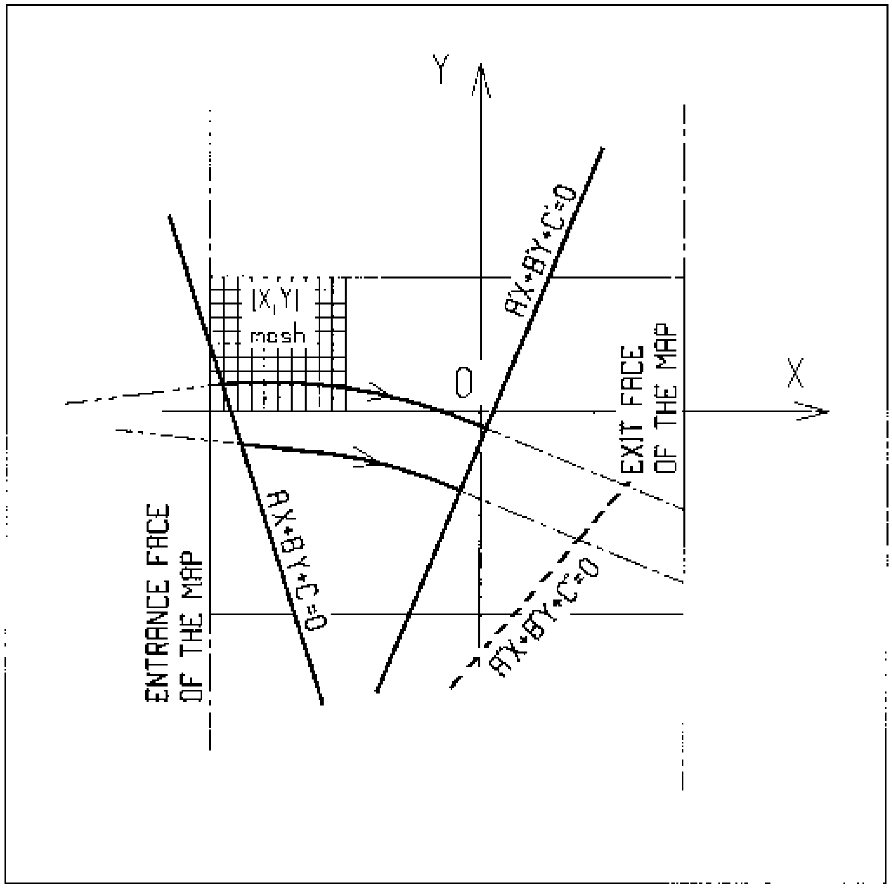


Figure 1: OXY is the coordinate system of the mesh. Integration boundaries may be defined, using $ID \neq 0$: particle coordinates are extrapolated linearly from the entrance face of the map, onto the boundary $A'X + B'Y + C' = 0$; after ray-tracing inside the map and terminating on the boundary $AX + BY + C = 0$, coordinates are extrapolated linearly onto the exit face of the map if $ID = 2$, or terminated on the last $(ID - 1)$ boundary if $ID > 2$.

CARTEMES — **2-D Cartesian uniform mesh magnetic field map**
mid-plane symmetry is assumed

IC, *IL*	$IC = 1, 2$: print the map $IL = 1, 2[\times 10^n]$: print field and coordinates along trajectories.	0-2, 0-2$[\times 10^n]$	2*I
BNORM, *XN*,*YN*	Field and X-,Y-coordinate normalization coeffs.	3*no dim.	3*E
TITL	Title. Start with "FLIP" to get field map X-flipped.		A80
IX, JY	Number of longitudinal (IX) and transverse (JY) nodes of the map	≤ 400, ≤ 200	2*I
FNAME [1]	File name		A80
ID, A, B, C [, A', B', C', A'', B'',etc., if $ID \geq 2$]	Integration boundary. Normally $ID = 0$. $ID = -1$: integration in the map begins at entrance boundary defined by $AX + BY + C = 0$. $ID = 1$: integration in the map is terminated at exit boundary defined by $AX + BY + C = 0$. $ID \geq 2$: entrance (A, B, C) and up to $ID - 1$ exit $(A', B', C', A'', B'', etc.)$ boundaries	≥ -1,2*no dim., cm [,2*no dim., cm, etc.]	I, 3*E [3*E,etc.]
IORDRE	Degree of interpolation polynomial (see *DIPOLE-M*)	2, 25 or 4	I
XPAS	Integration step	cm	E
KPOS, *XCE*, *YCE*, *ALE*	*KPOS*=1 : element aligned, 2 : misaligned ; shifts, tilt (unused if *KPOS*=1)	1-2, 2*cm, rad	I, 3*E

[2] *FNAME* (*e.g.*, spes2.map) contains the field data. These must be formatted according to the following *FORTRAN* sequence :

```
OPEN (UNIT = NL, FILE = FNAME, STATUS = 'OLD' [,FORM='UNFORMATTED'])
IF (BINARY) THEN
READ(NL) (Y(J), J=1, JY)
ELSE
READ(NL,100) (Y(J), J=1, JY)
ENDIF
100     FORMAT(10 F8.2)
DO 1 I=1,IX
IF (BINARY) THEN
READ(NL) X(I), (BMES(I,J), J=1, JY)
ELSE
READ(NL,101) X(I), (BMES(I,J), J=1, JY)
101  FORMAT(10 F8.2)
ENDIF
1       CONTINUE
```

where $X(I)$ and $Y(J)$ are the longitudinal and transverse coordinates and *BMES* is the Z field component at a node (I, J) of the mesh. For binary files, *FNAME* must begin with 'B_' or 'b_'.
'Binary' will then automatically be set to '.TRUE.'

CAVITE : Accelerating cavity

CAVITE provides a simulation of a (zero length) accelerating cavity ; it can be used in conjunction with keywords *REBELOTE* and *SCALING* for the simulation of multi-turn tracking with synchrotron or fixed field (FFAG, cyclotron) acceleration (see section **??**). It must be preceded by *PARTICUL* for the definition of mass M and charge q.

A major effect of *CAVITE* on optics settings is the following :

The reference rigidity of a problem, as used when computing optical strengths from field values, sections **??**-**??**, is specified in the object definition by *[MC]OBJET*. However, in many cases – options as described below – that reference rigidity will be updated upon crossing the cavity, by the amount of the synchronous rigidity increase as induced by the cavity, namely,

$$B\rho_{ref} = \mathit{BORO} \longrightarrow B\rho_{ref} = \mathit{BORO} + \delta B\rho_s$$

Note as an illustration of the process, that, in this case, a simple way to have the optical elements have their *strengths* maintained constant is to use *SCALING* with the option *NTIM* $= -1$.

If $\mathbf{IOPT} = \mathbf{0}$ **:** *CAVITE* is switched off.

If $\mathbf{IOPT} = \mathbf{1}$ **:** *CAVITE* simulates the RF cavity of a synchrotron accelerator : the periodic motion over $IP = 1$, *NPASS* + 1 turns (passes through the structure) is obtained using the keyword *REBELOTE*, option K = 99, while RF and optical elements time dependent functions are simulated by means of *SCALING* – see section **??**. *CAVITE* may conveniently be located *at the end* of the optical structure, otherwise its phasing has to be indicated. The synchrotron motion of any of the *IMAX* particles of a beam is obtained from the following mapping

$$\begin{cases} \phi_2 - \phi_1 = 2\pi\, f_{RF} \left(\dfrac{\ell}{\beta c} - \dfrac{\mathcal{L}}{\beta_s c} \right) \\ W_2 - W_1 = q\hat{V}\, \sin\phi_1 \end{cases}$$

where

- ϕ = RF phase ; $\phi_2 - \phi_1$ = variation of ϕ between two traversals
- W = kinetic energy ; $W_2 - W_1$ = energy gain at a traversal of *CAVITE*
- $\mathcal{L}$ = length of the synchronous closed orbit (to be calculated by prior ray-tracing, see the bottom NOTE)
- ℓ = orbit length of the particle between two traversals
- $\beta_s c$ = velocity of the (virtual) synchronous particle
- βc = velocity of the particle
- $\hat{V}$ = peak RF voltage
- q = particle electric charge.

The RF frequency f_{RF} is a multiple of the synchronous revolution frequency, and is obtained from the input data, following

$$f_{RF} = \frac{hc}{\mathcal{L}} \frac{q(B\rho)_s}{\sqrt{q^2(B\rho)_s^2 + (Mc)^2}}$$

where

- h = harmonic number of the R.F
- M = mass of the particle
- c = velocity of light.

The current rigidity $(B\rho)_s$ of the synchronous particle is obtained from the timing law specified by means of *SCALING* following $(B\rho)_s = \mathit{BORO} \cdot \mathit{SCALE(TIMING)}$ (see *SCALING* for the meaning and calculation of the scale factor *SCALE(TIMING)*). *If SCALING is not used,* $(B\rho)_s$ *is assumed to keep the constant value BORO* as given in the object description (see *OBJET* for instance).
The velocity βc of a particle is calculated from its current rigidity

$$\beta = \frac{q(B\rho)}{\sqrt{q^2(B\rho)^2 + (Mc)^2}}$$

The velocity $\beta_s c$ of the synchronous particle is obtained in the same way from

$$\beta_s = \frac{q(B\rho)_s}{\sqrt{q^2(B\rho)_s^2 + (Mc)^2}}$$

The kinetic energies and rigidities involved in these formulae are related by

$$q(B\rho) = \sqrt{W(W + 2Mc^2)}$$

Finally, the initial conditions for the mapping, at the first turn, are the following

- For the (virtual) synchronous particle

$$\phi_1 = \phi_s = \text{synchronous phase}$$
$$(B\rho)_{1s} = \mathit{BORO}$$

- For any of the $I = 1$, *IMAX* particles of the beam

$$\phi_{1I} = \phi_s = \text{synchronous phase}$$
$$(B\rho)_{1I} = \mathit{BORO} * D_I$$

where the quantities *BORO* and D_I are given in the object description.

Calculation of the Coordinates Let $p_I = \left[p_{XI}^2 + p_{YI}^2 + p_{ZI}^2\right]^{1/2}$ be the momentum of particle I at the exit of the cavity, while $p_{I_0} = \left[p_{XI_0}^2 + p_{YI_0}^2 + p_{ZI_0}^2\right]^{1/2}$ is its momentum at the entrance. The kick in momentum is assumed to be fully longitudinal, resulting in the following relations between the coordinates at the entrance (denoted by the index zero) and at the exit

$$p_{XI} = \left[p_I^2 - (p_{I_0}^2 - p_{XI_0}^2)\right]^{1/2}$$
$$p_{YI} = p_{YI_0}, \quad \text{and} \quad p_{ZI} = p_{ZI_0} \quad \text{(longitudinal kick)}$$
$$X_I = X_{I_0}, \quad Y_I = Y_{I_0} \quad \text{and} \quad Z_I = Z_{I_0} \quad \text{(zero length cavity)}$$

and for the angles (see Fig. **??**)

$$\left.\begin{aligned} T_I &= \text{Atg}\left(\frac{p_{YI}}{p_{XI}}\right) \\ P_I &= \text{Atg}\left(\frac{P_{ZI}}{(p_{XI}^2 + p_{YI}^2)^{1/2}}\right) \end{aligned}\right\} \quad \text{(damping of the transverse motion)}$$

If $\mathbf{IOPT} = \mathbf{2}$: the same simulation of a synchrotron RF cavity as for $\mathbf{IOPT} = \mathbf{1}$ is performed, except that the keyword *SCALING* (family *CAVITE*) is not taken into account in this option : the increase in kinetic energy at each traversal, for the synchronous particle, is

$$\Delta W_s = q\hat{V}\,\sin\phi_s$$

where the synchronous phase ϕ_s is given in the input data. From this, the calculation of the law $(B\rho)_s$ and the RF frequency f_{RF} follows, according to the formulae given in the *IOPT* = 1 case.

If $\mathbf{IOPT} = \mathbf{3}$: sine RF law, acceleration without synchrotron motion. Any particle will be given a kick

$$\Delta W = q\hat{V}\,\sin\phi_s$$

where $\hat{V}$ and ϕ_s are input data.

If $\mathbf{IOPT} = \mathbf{6}$: allows reading the RF frequency and/or phase law from an external file (wih name normally "zgoubi.freqLaw.In"). See routines `cavite.f` and `scalin.f` for details Was first used for acceleration in scaling FFAG [**?**].

If $\mathbf{IOPT} = \mathbf{7}$: fixed frequency RF, quasi- or isochronous acceleration. Was first used for quasi-isochronous, fixed frequency acceleration in the EMMA prototype linear FFAG [**?**, **?**].
Can be used for cyclotron acceleration.

NOTE. Calculation of the closed orbit :

Due to possible dipole type of optical defects (*e.g.*, fringe fields, straight axis combined function dipoles), the closed orbit may not coincide with the ideal axis of the optical elements (hence it will be almost everywhere non-zero). One way to calculate it at the beginning of the structure (*i.e.*, where the initial particle coordinates are defined) is to ray-trace a single particle over a sufficiently large number of turns, starting with initial conditions taken near the reference orbit, so as to obtain statistically well-defined transverse phase-space ellipses. The local closed orbit coincides with the coordinates Y_c, T_c, Z_c, P_c of the center of the ellipses. A few iterations are usually sufficient (avoid near-integer tunes) to ensure accuracy. Next, ray-tracing over one turn a particle starting with the initial condition (Y_c, T_c, Z_c, P_c) will provide the entire closed orbit, and as a sub-product its length $\mathcal{L}$ (the $F(6,1)$ coordinate in the *FORTRAN*).

CAVITE [1] | **Accelerating cavity**
$\Delta W = qV sin(2\pi h f \Delta t + \varphi_s)$ and other voltage and frequency laws.

IOPT[.i]	Option. $i = 1$ causes info output into `zgoubi.CAVITE.out`	0-7	I
If IOPT=0	Element inactive		
X, X	Unused		
X, X	Unused		
If IOPT=1 [2]	f_{RF} follows the timing law given by *SCALING*		
$\mathcal{L}$, h	Reference closed orbit length ; harmonic number	m, no dim.	2*E
$\hat{V}$, X	R.F. peak voltage ; unused	V, unused	2*E
If IOPT=2	f_{RF} follows $\Delta W_s = q\hat{V} sin\phi_s$		
$\mathcal{L}$, h	Reference closed orbit length ; harmonic number	m, no dim.	2*E
$\hat{V}$, ϕ_s	R.F. peak voltage ; synchronous phase	V, rad	2*E
If IOPT=3	No synchrotron motion : $\Delta W = q\hat{V} sin\phi_s$		
X, X	Unused ; unused	2*unused	2*E
$\hat{V}$, ϕ_s	R.F. peak voltage ; synchronous phase	V, rad	2*E
If IOPT=6	Read RF frequency and/or phase law from external file, “zgoubi.freqLaw.In”.		
$\mathcal{L}$, E_k	Orbit length and kinetic energy at start of acceleration.	m, MeV	2*E
$\hat{V}$, Φ_s	R.F. peak voltage ; synchronous phase.	V, rad	2*E
If IOPT=7	Quasi- or isochronous acceleration.		
X, E_k	Unused ; RF frequency ;	- , Hz	2*E
$\hat{V}$, Φ_s	R.F. peak voltage ; synchronous phase.	V, rad	2*E

[1] Use *PARTICUL* to declare mass and charge.
[2] For ramping the R.F. frequency following $B\rho(t)$, use *SCALING*, with family *CAVITE*.

CHAMBR : Long transverse aperture limitation

CHAMBR causes the identification, counting and stopping of particles that reach the transverse limits of the vacuum chamber. The chamber can be either rectangular (*IFORM* = 1) or elliptic (*IFORM* = 2). The chamber is centered at *YC*, *ZC* and has transverse dimensions $\pm YL$ and $\pm ZL$ such that any particle will be stopped if its coordinates Y, Z satisfy

$$(Y - \mathit{YC})^2 \geq YL^2 \text{ or } (Z - \mathit{ZC})^2 \geq ZL^2 \quad \text{if} \quad \mathit{IFORM} = 1$$
$$\frac{(Y - \mathit{YC})^2}{YL^2} + \frac{(Z - \mathit{ZC})^2}{ZL^2} \geq 1 \quad \text{if} \quad \mathit{IFORM} = 2$$

The conditions introduced with *CHAMBR* are valid along the optical structure until the next occurrence of the keyword *CHAMBR*. Then, if $IL = 1$ the aperture is possibly modified by introducing new values of *YC*, *ZC*, YL and ZL, or, if $IL = 2$ the chamber ends and information is printed concerning those particles that have been stopped.

The testing is done in optical elements at each integration step, between the *EFB*'s. For instance, in *QUADRUPO* there will be no testing from $-X_E$ to 0 and from *XL* to *XL* + X_S, but only from 0 to *XL* ; in *DIPOLE*, there is no testing as long as the *ENTRANCE EFB* is not reached, and testing is stopped as soon as the *EXIT* or *LATERAL EFB*'s are passed.

In optical elements defined in polar coordinates, Y stands for the radial coordinate (*e.g.*, *DIPOLE*, see Figs. **??**C, p. **??**, and **??**, p. **??**). Thus, centering *CHAMBR* at
$YC =$ *RM* simulates a chamber curved with radius *RM*, and having a radial acceptance *RM*± *YL*. In *DRIFT*, the testing is done at the beginning and at the end, and only for positive drifts. There is no testing in *CHANGREF*.

When a particle is stopped, its index *IEX* (see *OBJET* and section **??**) is set to the value -4, and its actual path length is stored in the array *SORT* for possible further use.

CHAMBR	**Long transverse aperture limitation** [1]		
IA	0 : element inactive 1 : (re)definition of the aperture 2 : stop testing and reset counters, print information on stopped particles.	0-2	I
IFORM[.J], $C1$, $C2$, $C3$, $C4$	*IFORM* = 1 : rectangular aperture ; *IFORM* = 2 : elliptical aperture. *J* = 0, default : opening is [2] $\pm YL = \pm C1$, $\pm ZL = \pm C2$, centered at $YC = C3$, $ZC = C4$. *J* = 1 : opening is [2], in Y : $[C1, C2]$, in Z : $[C3, C4]$	1-2[.0-1]	I[.I], 4*E

[1] Any particle out of limits is stopped.

[2] When used with an optical element defined in polar coordinates (*e.g.*, *DIPOLE*) YL is the radius and YC stands for the reference radius (normally, $YC \simeq RM$).

CHANGREF : Transformation to a new reference frame

CHANGREF transports particles from a reference plane (O, Y, Z) at path distance S, to a new one by a combination of translations and/or rotations. It essentially aims at positioning optical elements with respect to one another, as setting a reference frame at the entrance or exit of field maps, or to simulate misalignments (see also *KPOS* option). *CHANGREF* can be placed anywhere in a structure.

Spin tracking, particle decay and gas-scattering are taken into account in *CHANGREF*. Energy loss by synchrotron radiation (*SRLOSS* keyword) is not.

There are two "styles" of *CHANGREF*, as follows.

The "old style" *CHANGREF* requires the three data *XCE*, *YCE*, *ALE* and then gets the new particle coordinates Y_2, T_2, Z_2, P_2 and path length S_2 from the old ones Y_1, T_1, Z_1, P_1 and S_1 using

$$
\begin{aligned}
T_2 &= T_1 - ALE \\
Y_2 &= \frac{(Y_1 - YCE)\cos T_1 + XCE \sin T_1}{\cos T_2} \\
DL^2 &= (XCE - Y_2 \sin ALE)^2 + (YCE - Y_1 + Y_2 \cos ALE)^2 \\
Z_2 &= Z_1 + DL \,\mathrm{tg} P_1 \\
S_2 &= S_1 + \frac{DL}{\cos P_1} \\
P_2 &= P_1
\end{aligned}
$$

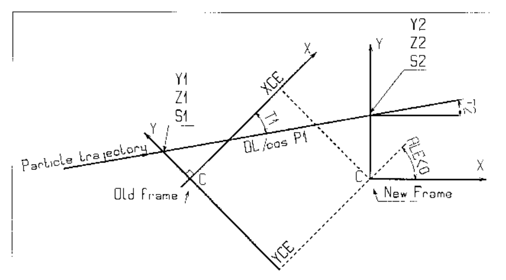


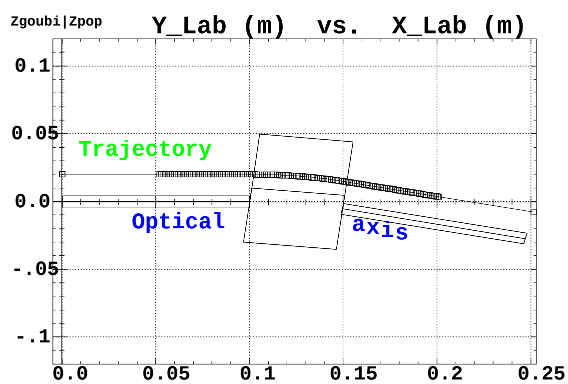
Figure 1: Scheme of the *CHANGREF* procedure.

where, XCE and YCE are shifts in the horizontal plane along, respectively, X- and Y-axis, and ALE is a rotation around the Z-axis. DL is given the sign of $XCE - Y_2 \sin(ALE)$.

The example below shows the use of *CHANGREF* for the symmetric positioning of a combined function dipole+quadrupole magnet in a drift-bend-drift geometry with 12.691 degrees deviation (obtained upon combined effect of a dipole component and of quadrupole axis shifted 1 cm off optical axis).

Zgoubi data file :

```
Using CHANGREF, "Old style"
'OBJET'
51.71103865921708                                   Electron, Ekin=15MeV.
2
1 1                                                    One particle, with
2. 0.  0.0 0.0 0.0 1. 'R'          Y_0=2 cm, other coordinates zero.
1 1 1 1 1 1 1
'MARKER'    BEG    .plt                          -> list  into zgoubi.plt.
'DRIFT'                                                    10 cm drift.
10.
'CHANGREF'
0. 0. -6.34165                                   First : half Z-rotation.
'CHANGREF'
0. 1. 0.                                                Next : Y-shift.
'MULTIPOL'       Combined function multipole, dipole + quadrupole.
2                                             -> list  into zgoubi.plt.
5  10. 2.064995867082342  2. 0. 0. 0. 0. 0. 0. 0. 0.
0 0  5. 1.1  1.00 1.00 1.00 1.00 1.00 1. 1. 1. 1.
4  .1455   2.2670  -.6395  1.1558  0. 0.  0.
0 0  5. 1.1  1.00 1.00 1.00 1.00 1.00 1. 1. 1. 1.
4  .1455   2.2670  -.6395  1.1558  0. 0.  0.
0 0 0 0 0 0 0 0 0 0
.1   step size
1  0. 0.  0.
'CHANGREF'
0. -1. -6.34165     First : Y-shift back ; next : half Z-rotation.
'DRIFT'                                                    10 cm drift.
10.
'MARKER'    END    .plt             ".plt" => list  into zgoubi.plt.
'FAISCEAU'
'END'
```



Note : The square markers scheme the stepwise integration in case of ±5 cm additional fringe field extent upstream and downstream of the 5 cm long multipole.

The "new style" *CHANGREF* allows all 6 degrees of freedom rather than just 3, namely, X-, Y-, Z-shift, X-, Y-, Z-rotation. In addition, *CHANGREF* "new style" allows up to 9 successive such elementary transformations, in arbitrary order. The "old style" example above is transposed into "new style", hereafter.

Zgoubi data file :

```
Using CHANGREF, "New Style"
'OBJET'
51.71103865921708                          Electron, Ekin=15MeV.
2
1 1                                            One particle, with
2. 0.  0.0 0.0 0.0 1. 'R'      Y_0=2 cm, other coordinates zero.
1 1 1 1 1 1 1
'MARKER'    BEG    .plt                -> list  into zgoubi.plt.
'DRIFT'                                              10 cm drift.
10.
'CHANGREF'
ZR -6.34165 YS 1.             First half Z-rotate, Next Y-shift.
'MULTIPOL'     Combined function multipole, dipole + quadrupole.
2                                   -> list  into zgoubi.plt.
5  10. 2.064995867082342  2. 0. 0. 0. 0. 0. 0. 0. 0.
0 0  5. 1.1  1.00 1.00 1.00 1.00 1.00 1. 1. 1. 1.
4  .1455   2.2670  -.6395  1.1558  0. 0.  0.
0 0  5. 1.1  1.00 1.00 1.00 1.00 1.00 1. 1. 1. 1.
4  .1455   2.2670  -.6395  1.1558  0. 0.  0.
0 0 0 0 0 0 0 0 0 0
.1   step size
1  0. 0.  0.
'CHANGREF'
YS -1. ZR -6.34165       First Y-shift back, next half Z-rotate.
'DRIFT'                                              10 cm drift.
10.
'FAISCEAU'
'END'
```

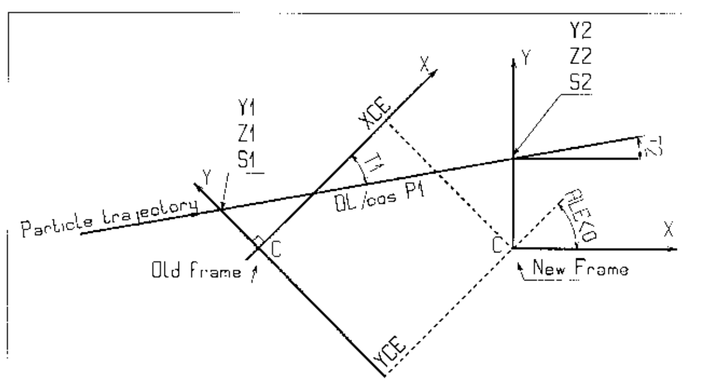
CHANGREF **Transformation to a new reference frame**

"Old Style" (Figure below) :

XCE, *YCE*, *ALE*	Longitudinal and transverse shifts, followed by Z-axis rotation	2*cm, deg	3*E

"New Style" (example below). In an arbitrary order, up to 9 occurrences of :

XS 'val', *YS 'val'*, *ZS 'val'*, *XR 'val'*, *YR 'val'*, *ZR 'val'*		cm or deg	up to 9*(A2,E)



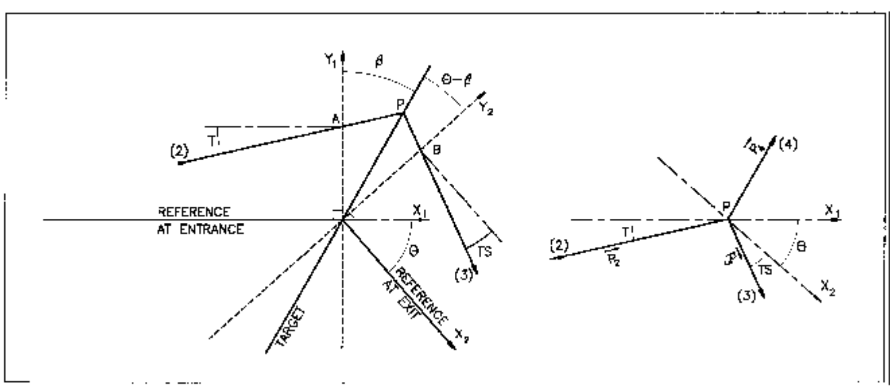
Parameters in the *CHANGREF* procedure.

Example :

```
Using CHANGREF "New Style
'OBJET'
51.71103865921708                           Electron, Ekin=15MeV.
2
1 1                                              One particle, with
2. 0.  0.0 0.0 0.0 1. 'R'      Y_0=2 cm, other coordinates zero.
1 1 1 1 1 1 1
'MARKER'    BEG    .plt                -> list  into zgoubi.plt.
'DRIFT'                                              10 cm drift.
10.
'CHANGREF'
ZR -6.34165 YS 1.            First half Z-rotate, Next Y-shift.
'CHANGREF'
0. 1. 0.
'MULTIPOL'     Combined function multipole, dipole + quadrupole.
2                                   -> list  into zgoubi.plt.
5  10. 2.064995867082342  2. 0. 0. 0. 0. 0. 0. 0. 0.
0 0  5. 1.1  1.00 1.00 1.00 1.00 1.00 1. 1. 1. 1.
4  .1455   2.2670  -.6395  1.1558  0. 0.  0.
0 0  5. 1.1  1.00 1.00 1.00 1.00 1.00 1. 1. 1. 1.
4  .1455   2.2670  -.6395  1.1558  0. 0.  0.
0 0 0 0 0 0 0 0 0 0
.1   step size
1  0. 0.  0.
'CHANGREF'
YS -1. ZR -6.341         First Y-shift back, next half Z-rotate.
'DRIFT'                                              10 cm drift.
10.
'FAISCEAU'
'END'
```

CIBLE, TARGET **Generate a secondary beam following target interaction**

M_1, M_2, M_3, Q T_2, θ, β	Target, incident and scattered particle masses ; Q of the reaction ; incident particle kinetic energy ; scattering angle ; angle of the target	5*$\frac{MeV}{c^2}$, 2*deg	7*E
NT, NP	Number of samples in T and P coordinates after *CIBLE*		2*I
TS, PS, DT	Sample step sizes ; tilt angle	3*mrad	3*E
BORO	New reference rigidity after *CIBLE*	kG.cm	E



Scheme of the principles of *CIBLE (TARGET)*

A, T = position, angle of incoming particle 2 in the entrance reference frame
P = position of the interaction
B, T = position, angle of the secondary particle in the exit reference frame
θ = angle between entrance and exit frames
β = tilt angle of the target

COLLIMA : Collimator

COLLIMA acts as a mathematical aperture of zero length. It causes the identification, counting and stopping of particles that reach the aperture limits.

Physical Aperture A physical aperture can be either rectangular (*IFORM* = 1) or elliptic (*IFORM* = 2). The collimator is centered at *YC*, *ZC* and has transverse dimensions $\pm YL$ and $\pm ZL$ such that any particle will be stopped if its coordinates Y, Z satisfy

$$(Y - \mathit{YC})^2 \geq YL^2 \text{ or } (Z - \mathit{ZC})^2 \geq ZL^2 \quad \text{if} \quad \mathit{IFORM} = 1$$
$$\frac{(Y - \mathit{YC})^2}{YL^2} + \frac{(Z - \mathit{ZC})^2}{ZL^2} \geq 1 \quad \text{if} \quad \mathit{IFORM} = 2$$

Longitudinal Phase-space Collimation *COLLIMA* can act as a longitudinal phase-space aperture, coordinates acted on are selected with *IFORM.J.* Any particle will be stopped if its horizontal (h) and vertical (v) coordinates satisfy

$$(h \leq h_{min} \text{ or } h \geq h_{max}) \text{ or } (v \leq v_{min} \text{ or } v \geq v_{max})$$

wherein, h is either path length S if *IFORM*=6 or time if *IFORM*=7, and v is either 1+DP/P if *J*=1 or kinetic energy if *J*=2 (provided mass and charge have been defined using the keyword *PARTICUL*).

Transverse Phase-space Collimation *COLLIMA* can act as a transverse phase-space aperture. Any particle will be stopped if its coordinates satisfy

$$\gamma_Y Y^2 + 2\alpha_Y YT + \beta_Y T^2 \geq \epsilon_Y/\pi \quad \text{if} \quad \mathit{IFORM} = 11 \text{ or } 14$$
$$\gamma_Z Z^2 + 2\alpha_Z ZP + \beta_Z P^2 \geq \epsilon_Z/\pi \quad \text{if} \quad \mathit{IFORM} = 12 \text{ or } 15$$

If *IFORM*=11 (respectively 12) then ϵ_Y/π (respectively ϵ_Z/π) is to be specified by the user as well as $\alpha_{Y,Z}$, $\beta_{Y,Z}$. If *IFORM*=14 (respectively 15) then α_Y and β_Y (respectively α_Z, β_Z) are determined by **zgoubi** by prior computation of the matched ellipse to the particle population, so only $\epsilon_{Y,Z}/\pi$ need be specified by the user.

When a particle is stopped, its index *IEX* (see *OBJET* and section **??**) is set to the value -4, and its actual path length is stored in the array *SORT* for possible further use with *HISTO*).

COLLIMA	**Collimator** [1]		
IA	0 : element inactive 1 : element active 2 : element active and print information on stopped particles	0-2	I
Physical-space collimation			
IFORM[.J], $C1$, $C2$, $C3$, $C4$	*IFORM* = 1 : rectangular aperture ; *IFORM* = 2 : elliptical aperture. *J* = 0, default : opening is $\pm YL = \pm C1$, $\pm ZL = \pm C2$, centered at $YC = C3$, $ZC = C4$. *J* = 1 : opening is, in Y : $[C1, C2]$, in Z : $[C3, C4]$	1-2[.0-1]	I[.I], 4*E
Longitudinal collimation			
IFORM.J, H_{min}, H_{max}, V_{min}, V_{max}	*IFORM* = 6 or 7 for horizontal variable resp[ly] S or Time, J=1 or 2 for vertical variable resp[ly] 1+dp/p, kinetic-E (MeV) ; horizontal and vertical limits	2*cm or 2*s, 2*no.dim or 2*MeV	I, 4*E
Phase-space collimation			
IFORM, α, β, ϵ/π, N_σ	*IFORM* = 11, 14 : horizontal collimation ; horizontal ellipse parameters (unused if 14) [2], emittance, cut-off *IFORM* = 12, 15 : vertical collimation ; vertical ellipse parameters (unused if 15) [2], emittance, cut-off *IFORM* = 13, 16 : longitudinal collimation ; *to be implemented*	11-16, no.dim, 2*m, no.dim	I, 4*E

[1] Any particle out of limits is stopped.
[2] The rejection boundary is the *rms* ellipse matched to the particle distribution.

DECAPOLE : Decapole magnet (Fig. 1)

The meaning of parameters for *DECAPOLE* is the same as for *QUADRUPO*.
In fringe field regions the magnetic field $\vec{B}(X,Y,Z)$ and its derivatives up to fourth order are derived from the scalar potential expressed to the 5th order in Y and Z

$$V(X,Y,Z) = G(X)\left(Y^4 Z - 2Y^2 Z^3 + \frac{Z^5}{5}\right)$$
$$\text{with } G_0 = \frac{B_0}{R_0^4}$$

The modelling of the fringe field form factor $G(X)$ is described under *QUADRUPO*, p. **??**.

Outside fringe field regions, or everywhere in sharp edge decapole ($\lambda_E = \lambda_S = 0$) , $\vec{B}(X,Y,Z)$ in the magnet is given by

$$B_X = 0$$
$$B_Y = 4G_0(Y^2 - Z^2)YZ$$
$$B_Z = G_0(Y^4 - 6Y^2 Z^2 + Z^4)$$

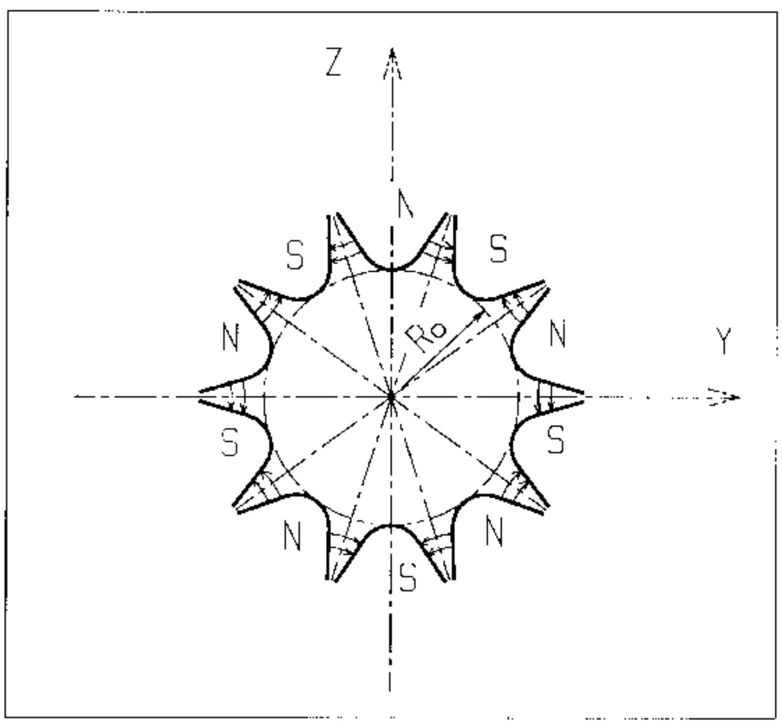
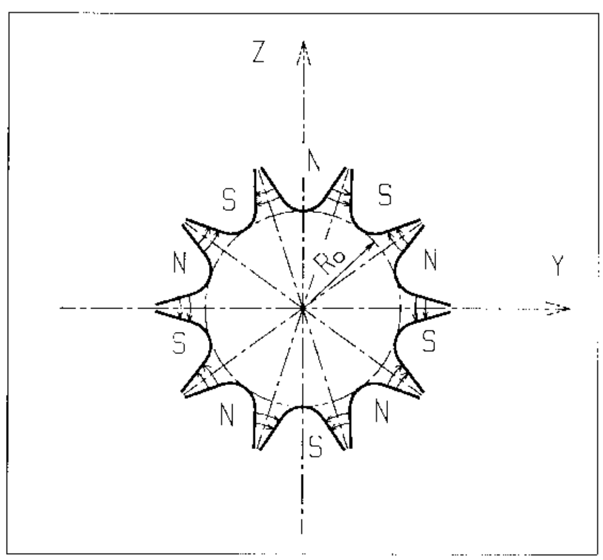


Figure 1: Decapole magnet

DECAPOLE **Decapole magnet**

IL	$IL = 1, 2[\times 10^n]$: print field and coordinates along trajectories.	0-2$[\times 10^n]$	I
XL, R_0, B_0	Length ; radius and field at pole tip	2*cm, kG	3*E
X_E, λ_E	Entrance face : Integration zone extent ; fringe field extent ($\lesssim 2R_0$, $\lambda_E = 0$ for sharp edge)	2*cm	2*E
NCE, $C_0 - C_5$	*NCE* = unused $C_0 - C_5$ = Fringe field coefficients such that $G(s) = G_0/(1 + \exp P(s))$, with $G_0 = B_0/R_0^4$ and $P(s) = \sum_{i=0}^{5} C_i(s/\lambda)^i$	unused, 6*no dim.	I, 6*E
X_S, λ_S *NCS*, $C_0 - C_5$	Exit face : see entrance face	2*cm 0-6, 6*no dim.	2*E I, 6*E
XPAS	Integration step	cm	E
KPOS, *XCE*, *YCE*, *ALE*	*KPOS*=1 : element aligned, 2 : misaligned ; shifts, tilt (unused if *KPOS*=1)	1-2, 2*cm, rad	I, 3*E



DIPOLE : Dipole magnet, polar frame

DIPOLE provides a model of a dipole field, possibly with transverse indices. The field along a particle trajectory is computed as the particle motion proceeds, straightforwardly from the dipole geometrical boundaries. Field simulation in *DIPOLE* is the same as used in *DIPOLE-M* and *AIMANT* for computing a field map ; the essential difference in *DIPOLE* is in its skipping that intermediate stage of field map generation found in *DIPOLE-M* and *AIMANT*.

DIPOLE has a version, *DIPOLES*, that allows overlapping of fringe fields in a configuration of neighboring magnets.

The dimensioning of the magnet is defined by (Fig. **??**, p. **??**)

AT	:	total angular aperture
RM	:	mean radius used for the positioning of field boundaries

The 2 or 3 effective field boundaries (EFB), from which the dipole field is drawn, are defined from geometric boundaries, the shape and position of which are determined by the following parameters.

ACENT	:	arbitrary inner angle, used for EFB's positioning
ω	:	azimuth of an EFB with respect to *ACENT*
θ	:	angle of an EFB with respect to its azimuth (wedge angle)
R_1, R_2	:	radius of curvature of an EFB
U_1, U_2	:	extent of the linear part of an EFB.

The magnetic field is calculated in polar coordinates. At any position (R, θ) along the particle trajectory the value of the vertical component of the mid-plane field is calculated using

$$B_Z(R,\theta) = \mathcal{F}(R,\theta) * B_0 * \left(1 + N * \left(\frac{R-RM}{RM}\right) + B * \left(\frac{R-RM}{RM}\right)^2 + G * \left(\frac{R-RM}{RM}\right)^3\right) \quad (1)$$

where N, B and G are respectively the first, second and third order field indices and $\mathcal{F}(R,\theta)$ is the fringe field coefficient (it determines the "flutter" in periodic structures).

Calculation of the Fringe Field Coefficient With each EFB a realistic extent of the fringe field, λ (normally equal to the gap size), is associated and a fringe field coefficient F is calculated. In the following λ stands for either λ_E (Entrance), λ_S (Exit) or λ_L (Lateral EFB).

F is an exponential type fringe field (Fig. **??**, p. **??**) given by [**?**]

$$F = \frac{1}{1 + \exp P(s)}$$

wherein s is the distance to the EFB and depends on (R,θ), and

$$P(s) = C_0 + C_1\left(\frac{s}{\lambda}\right) + C_2\left(\frac{s}{\lambda}\right)^2 + C_3\left(\frac{s}{\lambda}\right)^3 + C_4\left(\frac{s}{\lambda}\right)^4 + C_5\left(\frac{s}{\lambda}\right)^5$$

It is also possible to simulate a shift of the *EFB*, by giving a non zero value to the parameter *shift*. s is then changed to $s-$*shift* in the previous equation. This allows small variations of the magnetic length.

Let F_E (respectively F_S, F_L) be the fringe field coefficient attached to the entrance (respectively exit, lateral) EFB. At any position on a trajectory the resulting value of the fringe field coefficient (eq. 1) is

$$\mathcal{F}(R,\theta) = F_E * F_S * F_L$$

In particular, $F_L \equiv 1$ if no lateral EFB is requested.

Calculation of the Mid-plane Field and Derivatives $B_Z(R,\theta)$ in Eq. 1 is computed at the $n \times n$ nodes ($n = 3$ or 5 in practice) of a "flying" interpolation grid in the median plane centered on the projection m_0 of the actual particle position M_0 as schemed in Fig. 1. A polynomial interpolation is involved, of the form

$$B_Z(R,\theta) = A_{00} + A_{10}\theta + A_{01}R + A_{20}\theta^2 + A_{11}\theta R + A_{02}R^2$$

that yields the requested derivatives, using

$$A_{kl} = \frac{1}{k!l!}\frac{\partial^{k+l}B_Z}{\partial\theta^k\partial r^l}$$

Note that, the source code contains the explicit analytical expressions of the coefficients A_{kl} solutions of the normal equations, so that the operation *is not* CPU time consuming.

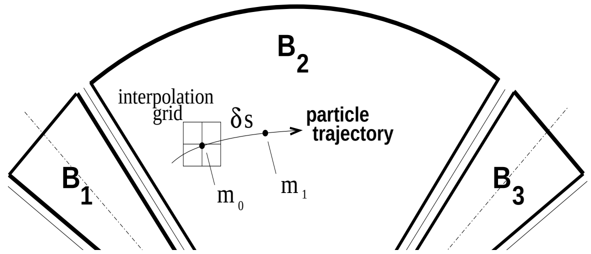


Figure 1: Interpolation method. m_0 and m_1 are the projections in the median plane of particle positions M_0 and M_1 and separated by δs, projection of the integration step.

Extrapolation Off Median Plane From the vertical field $\vec{B}$ and derivatives in the median plane, first a transformation from polar to Cartesian coordinates is performed, following eqs (**??** or **??**), then, extrapolation off median plane is performed by means of Taylor expansions, following the procedure described in section **??**.

DIPOLE	**Dipole magnet, polar frame** $B_Z = \mathcal{F}B_0\left(1 + N\left(\frac{R-RM}{RM}\right) + B\left(\frac{R-RM}{RM}\right)^2 + G\left(\frac{R-RM}{RM}\right)^3\right)$		
IL	$IL = 1, 2[\times 10^n]$: print field and coordinates along trajectories.	0-2$[\times 10^n]$	I
AT, RM	Total angular extent of the dipole ; reference radius	deg, cm	2*E
ACENT, B_0, N, B, G	Azimuth for positioning of EFBs ; field and field indices	deg., kG, 3*no dim.	5*E
	ENTRANCE FIELD BOUNDARY		
λ, ξ	Fringe field extent (normally $\simeq$ gap size) ; unused. Exponential type fringe field $F = 1 / (1 + \exp(P(s)))$ with $P(s) = C_0 + C_1(\frac{s}{\lambda}) + C_2(\frac{s}{\lambda})^2 + ... + C_5(\frac{s}{\lambda})^5$	cm, unused	2*E
NC, $C_0 - C_5$, shift	Unused ; C_0 to C_5 : see above ; EFB shift	0-6, 6*no dim., cm	I,7*E
ω^+, θ, R_1, U_1, U_2, R_2	Azimuth of entrance EFB with respect to *ACENT* ; wedge angle of EFB ; radii and linear extents of EFB (use $\mid U_{1,2} \mid = \infty$ when $R_{1,2} = \infty$)	2*deg, 4*cm	6*E
	EXIT FIELD BOUNDARY (See ENTRANCE FIELD BOUNDARY)		
λ, ξ NC, $C_0 - C_5$, shift	Fringe field parameters	cm, unused 0-6, 6*no dim., cm	2*E 1, 7*E
ω^-, θ, R_1, U_1, U_2, R_2	Positioning and shape of the exit EFB	2*deg, 4*cm	6*E
	LATERAL FIELD BOUNDARY (See ENTRANCE FIELD BOUNDARY)		
λ, ξ NC, $C_0 - C_5$, shift	LATERAL EFB is inhibited if $\xi = 0$	cm, unused 0-6, 6*no dim., cm	2*E 1, 7*E
ω^-, θ, R_1, U_1, U_2, R_2, $RM3$	Positioning and shape of the exit EFB	2*deg, 5*cm	7*E
IORDRE, Resol	Degree of interpolation polynomial : 2 = second degree, 9-point grid 25 = second degree, 25-point grid 4 = fourth degree, 25-point grid ; resolution of flying mesh is *XPAS/Resol*	2, 25 or 4 ; no dim.	I, E
XPAS	Integration step	cm	E
KPOS	Positioning of the map, normally 2. Two options :	1-2	I
If KPOS = 2 RE, TE, RS, TS	Positioning as follows : Radius and angle of reference, respectively, at entrance and exit of the map.	cm, rad, cm, rad	4*E
If KPOS = 1 DP	Automatic positioning of the map, by means of reference relative momentum	no dim.	E

DIPOLE-M : Generation of dipole mid-plane 2-D map, polar frame

DIPOLE-M is a more recent, simpler and improved version of *AIMANT*.

The keyword *DIPOLE-M* provides an automatic generation of a dipole field map in polar coordinates. The extent of the map is defined by the following parameters, as shown in Figs. **??**A and **??**B.

AT	:	total angular aperture
RM	:	mean radius used for the positioning of field boundaries
RMIN, RMAX	:	minimum and maximum radii

The 2 or 3 effective field boundaries (EFB) inside the map are defined from geometric boundaries, the shape and position of which are determined by the following parameters.

ACENT	:	arbitrary inner angle, used for EFB's positioning
ω	:	azimuth of an EFB with respect to *ACENT*
θ	:	angle of an EFB with respect to its azimuth (wedge angle)
R_1, R_2	:	radius of curvature of an EFB
U_1, U_2	:	extent of the linear part of an EFB.

At any node of the map mesh, the value of the field is calculated as

$$B_Z(R,\theta) = \mathcal{F}(R,\theta) * B_0 * \left(1 + N * \left(\frac{R-RM}{RM}\right) + B * \left(\frac{R-RM}{RM}\right)^2 + G * \left(\frac{R-RM}{RM}\right)^3\right) \tag{1}$$

where N, B and G are respectively the first, second and third order field indices and $\mathcal{F}$ is the fringe field coefficient.

Calculation of the Fringe Field Coefficient With each EFB a realistic extent of the fringe field, λ (normally equal to the gap size), is associated and a fringe field coefficient F is calculated. In the following λ stands for either λ_E (Entrance), λ_S (Exit) or λ_L (Lateral EFB).

F is an exponential type fringe field (Fig. **??**) given by [**?**]

$$F = \frac{1}{1 + \exp P(s)}$$

where s is the distance to the EFB, and

$$P(s) = C_0 + C_1\left(\frac{s}{\lambda}\right) + C_2\left(\frac{s}{\lambda}\right)^2 + C_3\left(\frac{s}{\lambda}\right)^3 + C_4\left(\frac{s}{\lambda}\right)^4 + C_5\left(\frac{s}{\lambda}\right)^5$$

It is also possible to simulate a shift of the *EFB*, by giving a non zero value to the parameter *shift*. s is then changed to $s-$*shift* in the previous equation. This allows small variations of the total magnetic length.

Let F_E (respectively F_S, F_L) be the fringe field coefficient attached to the entrance (respectively exit, lateral) EFB. At any node of the map mesh, the resulting value of the fringe field coefficient (eq. 1) is

$$\mathcal{F}(R,\theta) = F_E * F_S * F_L$$

In particular, $F_L \equiv 1$ if no lateral EFB is requested.

The Mesh of the Field Map The magnetic field is calculated at the nodes of a mesh with polar coordinates, in the median plane. The radial step is given by

$$\delta R = \frac{RMAX - RMIN}{IRMAX - 1}$$

and the angular step by

$$\delta\theta = \frac{AT}{IAMAX - 1}$$

where *RMIN*and *RMAX* are the lower and upper radial limits of the field map, and AT is its total angular aperture (Fig. **??**B). *IRMAX* and *IAMAX* are the total number of nodes in the radial and angular directions.

Simulating Field Defects and Shims Once the initial map is calculated, it is possible to modify it by means of the parameter *NBS*, so as to simulate field defects or shims.

If NBS $= -2$, the map is globally modified by a perturbation proportional to $R - R_0$, where R_0 is an arbitrary radius, with an amplitude $\Delta B_Z/B_0$, so that B_Z at the nodes of the mesh is replaced by

$$B_Z * \left(1 + \frac{\Delta B_Z}{B_0} \frac{R - R_0}{\textit{RMAX- RMIN}}\right)$$

If NBS $= -1$, the perturbation is proportional to $\theta - \theta_0$, and B_Z is replaced by

$$B_Z * \left(1 + \frac{\Delta B_Z}{B_0} \frac{\theta - \theta_0}{AT}\right)$$

If NBS ≥ 1, then *NBS* shims are introduced at positions $\frac{R_1 + R_2}{2}, \frac{\theta_1 + \theta_2}{2}$ (Fig. **??**) [**?**]
The initial field map is modified by shims with second order profiles given by

$$\theta = \left(\gamma + \frac{\alpha}{\mu}\right) \beta \frac{X^2}{\rho^2}$$

where X is shown in Fig. **??**, $\rho = \frac{R_1 + R_2}{2}$ is the central radius, α and γ are the angular limits of the shim, β and μ are parameters.
At each shim, the value of B_Z at any node of the initial map is replaced by

$$B_Z * \left(1 + F\theta * FR * \frac{\Delta B_Z}{B_0}\right)$$

where $F\theta = 0$ or $FR = 0$ outside the shim, and $F\theta = 1$ and $FR = 1$ inside.

Extrapolation Off Median Plane The vector field $\vec{B}$ and its derivatives in the median plane are calculated by means of a second or fourth order polynomial interpolation, depending on the value of the parameter *IORDRE* (*IORDRE*=2, 25 or 4, see section **??**). The transformation from polar to Cartesian coordinates is performed following eqs (**??** or **??**). Extrapolation off median plane is then performed by means of Taylor expansions, following the procedure described in section **??**.

DIPOLE-M	**Generation of dipole mid-plane 2-D map, polar frame** $B_Z = \mathcal{F} B_0 \left(1 + N\left(\frac{R-RM}{RM}\right) + B\left(\frac{R-RM}{RM}\right)^2 + G\left(\frac{R-RM}{RM}\right)^3\right)$		
NFACE, IC, IL	Number of field boundaries $IC = 1, 2$: print field map $IL = 1, 2$: print field and coordinates on trajectories	2-3, 0-2, 0-2$[\times 10^n]$	3*I
IAMAX, *IRMAX*	Azimuthal and radial number of nodes of the mesh	≤ 400, ≤ 200	2*I
B_0, N, B, G	Field and field indices	kG, 3*no dim.	4*E
AT, *ACENT*, RM, *RMIN*, *RMAX*	Mesh parameters : total angle of the map ; azimuth for positioning of EFBs ; reference radius ; minimum and maximum radii	2*deg, 3*cm	5*E
	ENTRANCE FIELD BOUNDARY		
λ, ξ	Fringe field extent (normally $\simeq$ gap size) ; unused. Exponential type fringe field $F = 1 \,/\, (1 + \exp(P(s)))$ with $P(s) = C_0 + C_1(\frac{s}{\lambda}) + C_2(\frac{s}{\lambda})^2 + ... + C_5(\frac{s}{\lambda})^5$	cm, unused	2*E
NC, $C_0 - C_5$, shift	Unused ; C_0 to C_5 : see above ; EFB shift	0-6, 6*no dim., cm	I,7*E
ω^+, θ, R_1, U_1, U_2, R_2	Azimuth of entrance EFB with respect to *ACENT* ; wedge angle of EFB ; radii and linear extents of EFB (use $\mid U_{1,2} \mid = \infty$ when $R_{1,2} = \infty$)	2*deg, 4*cm	6*E
	(Note : $\lambda = 0$, ω^+ = *ACENT* and $\theta = 0$ for <u>sharp edge</u>)		
	EXIT FIELD BOUNDARY (See ENTRANCE FIELD BOUNDARY)		
λ, ξ NC, $C_0 - C_5$, shift	Fringe field parameters	cm, unused 0-6, 6*nodim., cm	2*E 1, 7*E
ω^-, θ, R_1, U_1, U_2, R_2	Positioning and shape of the exit EFB	2*deg, 4*cm	6*E
	(Note : $\lambda = 0$, $\omega^- = -AT+$ *ACENT* and $\theta = 0$ for <u>sharp edge</u>)		
If NFACE = 3	LATERAL FIELD BOUNDARY (See ENTRANCE FIELD BOUNDARY) Next 3 records *only* if *NFACE* = 3		
λ, ξ	Fringe field parameters	cm, (cm)	2*E
NC, $C_0 - C_5$, shift		0-6, 6*no dim., cm	I, 7*E
ω^-, θ, R_1, U_1, U_2, R_2, $RM3$	Positioning and shape of the lateral EFB ; RM3 is the radial position on azimuth *ACENT*	2*deg, 5*cm	7*E
NBS	Option index for perturbations to the field map	normally 0	I
If NBS = 0	Normal value. No other record required		
If NBS = -2	The map is modified as follows :		
R_0, $\Delta B/B_0$	B transforms to $B * \left(1 + \frac{\Delta B}{B_0} \frac{R-R_0}{RMAX-RMIN}\right)$	cm, no dim.	2*E
If NBS = -1	the map is modified as follows :		
θ_0, $\Delta B/B_0$	B transforms to $B * \left(1 + \frac{\Delta B}{B_0} \frac{\theta-\theta_0}{AT}\right)$	deg, no dim.	2*E

If NBS ≥ 1	Introduction of NBS shims		
For I = 1, NBS	The following 2 records must be repeated NBS times		
R_1, R_2, θ_1, θ_2, λ	Radial and angular limits of the shim ; λ is unused	2*cm, 2*deg, cm	5*E
γ, α, μ, β	geometrical parameters of the shim	2*deg, 2*no dim.	4*E
IORDRE	Degree of interpolation polynomial : 2 = second degree, 9-point grid 25 = second degree, 25-point grid 4 = fourth degree, 25-point grid	2, 25 or 4	I
XPAS	Integration step	cm	E
KPOS	Positioning of the map, normally 2. Two options :	1-2	I
If KPOS = 2 RE, TE, RS, TS	Positioning as follows : Radius and angle of reference, respectively, at entrance and exit of the map.	cm, rad, cm, rad	4*E
If KPOS = 1 DP	Automatic positioning of the map, by means of reference relative momentum	no dim.	E

DIPOLES : Dipole magnet N-tuple, polar frame [?, ?]

DIPOLES works much like *DIPOLE* as to the field modelling, yet with the particularity that it allows positioning up to 5 such dipoles within the angular sector with full aperture AT thus allowing accounting for overlapping fringe fields. This is done in the following way[1].

The dimensioning of the magnet is defined by

AT	:	total angular aperture
RM	:	mean radius used for the positioning of field boundaries

For each one of the $N = 1$ to 5 dipoles of the N-tuple, the 2 effective field boundaries (entrance and exit EFBs) from which the dipole field (eqs. 1, 2) is drawn are defined from geometrical boundaries, the shape and position of which are determined by the following parameters (in the same manner as in *DIPOLE*, *DIPOLE-M*) (see Fig. **??**-A, p. **??**, and Fig. 1)

ACN_i	:	arbitrary inner angle, used for EFB's positioning
ω	:	azimuth of an EFB with respect to *ACN*
θ	:	angle of an EFB with respect to its azimuth (wedge angle)
R_1, R_2	:	radius of curvature of an EFB
U_1, U_2	:	extent of the linear part of an EFB

Calculation of the Field From a Single Dipole The magnetic field is calculated in polar coordinates. At all (R, θ) in the median plane ($Z = 0$), the magnetic field due a single one (index i) of the dipoles of a N-tuple magnet can take either form, upon option,

$$(i) \qquad B_{Z\,i}(R,\theta) = B_{Z0,i}\,\mathcal{F}_i(R,\theta)\,\left(1 + b_{1_i}(R - RM_i)/RM_i + b_{2_i}(R - RM_i)^2/RM_i^2 + ...\right) \tag{1}$$

$$(ii) \qquad B_Z(R,\theta) = B_{Z0,i} + \sum_{i=1}^{N} \mathcal{F}_i(R,\theta)\,\left(b_{1_i}(R - RM_i) + b_{2_i}(R - RM_i)^2 + ...\right) \tag{2}$$

wherein $B_{Z0,i}$ is a reference field, at reference radius RM_i, and $\mathcal{F}(R,\theta)$ is the fringe field coefficient, see below. This field model is proper to simulate for instance chicane dipoles, cyclotron or FFAG magnets, etc.

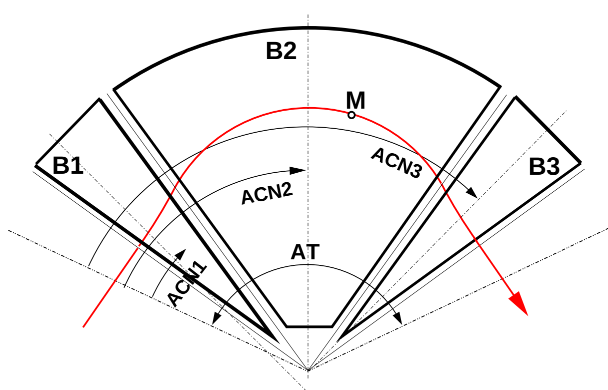


Figure 1: Definition of a dipole triplet using the *DIPOLES* or *FFAG* procedures.

Calculation of the Fringe Field Coefficient In a dipole, a realistic extent of the fringe field, g, is associated with each EFB, and a fringe field coefficient F is calculated.

F is an exponential type fringe field (Fig. **??**, page **??**) given by [**?**]

$$F = \frac{1}{1 + \exp P(d)}$$

wherein d is the distance to the EFB and depends on (R, θ), and

[1] *FFAG* can be referred to as another instance of a procedure based on such method.

$$P(d) = C_0 + C_1 \left(\frac{d}{g}\right) + C_2 \left(\frac{d}{g}\right)^2 + C_3 \left(\frac{d}{g}\right)^3 + C_4 \left(\frac{d}{g}\right)^4 + C_5 \left(\frac{d}{g}\right)^5$$

In addition, g is made dependent of R (a way to simulate the effect of variable gap size on fringe field extent), under the form

$$g(R) = g_0 \, (RM/R)^{\kappa}$$

This dependence is accounted for rigorously if the interpolation method (see below) is used, or else to order zero (derivatives of $g(R)$ are not considered) if the analytic method (below) is used.

Let F_E (respectively F_S) be the fringe field coefficient attached to the entrance (respectively exit) EFB ; at any position on a trajectory the resulting value of the fringe field coefficient is taken to be

$$\mathcal{F}_i(R,\theta) = F_E * F_S \tag{3}$$

Calculation of the Field Resulting From all N Dipoles Now, accounting for N neighboring dipoles in an N-tuple, the mid-plane field and field derivatives are obtained by addition of the contributions of the N dipoles taken separately, namely

$$B_Z(R,\theta) = \sum_{i=1,N} B_{Zi}(R,\theta) \tag{4}$$

$$\frac{\partial^{k+l} \vec{B_Z}(R,\theta)}{\partial\theta^k \partial r^l} = \sum_{i=1,N} \frac{\partial^{k+l} \vec{B_{Z}}_i(R,\theta)}{\partial\theta^k \partial r^l} \tag{5}$$

Note that, in doing so it is not meant that field superposition does apply in reality, it is just meant to provide the possibility of obtaining a realistic field shape, that would for instance closely match (using appropriate $C_0 - C_5$ sets of coefficients) 3-D field simulations obtained from magnet design codes.

Calculation of the Mid-plane Field Derivatives Two methods have been implemented to calculate the field derivatives in the median plane (Eq. 4), based on either analytical expressions derived from the magnet geometrical description, or classical numerical interpolation.

The first method has the merit of insuring best symplecticity in principle and fastest tracking. The interest of the second method is in its facilitating possible changes in the mid-plane magnetic field model $B_Z(R,\theta)$, for instance if simulations of shims, defects, or special R,θ field dependence need to be introduced.

*Analytical method [**?**] :*

The starting ingredients are, on the one hand distances to the EFBs,

$$d(R,\theta) = \sqrt{(x(R,\theta) - x_0(R,\theta))^2 + (y(R,\theta) - y_0(R,\theta))^2}$$

to be computed for the two cases d_{Entrance}, d_{Exit}, and on the other hand the expressions of the coordinates of particle position M and its projection P on the EFB in terms of the magnet geometrical parameters, namely

$$\begin{aligned}
x(R,\theta) &= \cos(ACN - \theta) - RM \\
y(R,\theta) &= R\sin(ACN - \theta) \\
x_P(R,\theta) &= \sin(u)\ (y(R,\theta) - y_b)/2 + x_b\ \sin^2(u) + x(R,\theta)\ \cos^2(u) \\
y_P(R,\theta) &= \sin(u)\ (x(R,\theta) - x_b)/2 + y_b\ \cos^2(u) + y(R,\theta)\ \sin^2(u)
\end{aligned}$$

with $x_b,\ y_b,\ u$ parameters drawn from the magnet geometry (sector angle, wedge angle, face curvatures, etc.).

These ingredients allow calculating the derivatives $\frac{\partial^{u+v} x(R,\theta)}{\partial\theta^u \partial r^v}$, $\frac{\partial^{u+v} y(R,\theta)}{\partial\theta^u \partial r^v}$, $\frac{\partial^{u+v} x_0(R,\theta)}{\partial\theta^u \partial r^v}$, $\frac{\partial^{u+v} y_0(R,\theta)}{\partial\theta^u \partial r^v}$, which, in turn, intervene in the derivatives of the compound functions $\frac{\partial^{u+v} F(R,\theta)}{\partial\theta^u \partial r^v}$, $\frac{\partial^{u+v} p(R,\theta)}{\partial\theta^u \partial r^v}$, $\frac{\partial^{u+v} d(R,\theta)}{\partial\theta^u \partial r^v}$.

Interpolation method :

The expression $B_Z(R,\theta)$ in Eq. 4 is, in this case, computed at the $n \times n$ nodes (n = 3 or 5 in practice) of a "flying" interpolation grid in the median plane centered on the projection m_0 of the actual particle position M_0 as schemed in Fig. 2. A polynomial interpolation is involved, of the form

$$B_Z(R,\theta) = A_{00} + A_{10}\theta + A_{01}R + A_{20}\theta^2 + A_{11}\theta R + A_{02}R^2$$

that yields the requested derivatives, using

$$A_{kl} = \frac{1}{k!l!}\,\frac{\partial^{k+l}B}{\partial\theta^k\partial r^l}$$

Note that, the source code contains the explicit analytical expressions of the coefficients A_{kl} solutions of the normal equations, so that the operation *is not* CPU time consuming.

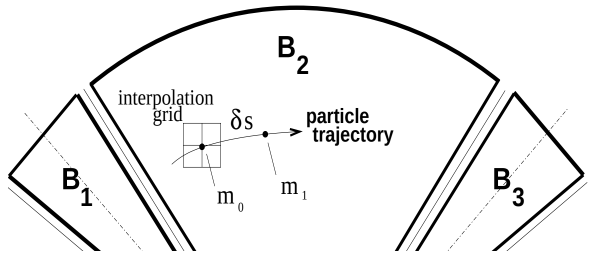


Figure 2: Interpolation method. m_0 and m_1 are the projections in the median plane of particle positions M_0 and M_1 and separated by δs, projection of the integration step.

Extrapolation Off Median Plane From the vertical field $\vec{B}$ and derivatives in the median plane, first a transformation from polar to Cartesian coordinates is performed, following eqs (**??** or **??**), then, extrapolation off median plane is performed by means of Taylor expansions, following the procedure described in section **??**.

Sharp Edge Sharp edge field fall-off at a field boundary can only be simulated if the following conditions are fulfilled :

- entrance (resp. exit) field boundary coincides with entrance (resp. exit) dipole limit (it means in particular, see Fig. **??**, $\omega^+ = ACENT$ (resp. $\omega^- = -(AT - ACENT)$), together with $\theta = 0$ at entrance (resp. exit) EFBs),
- analytical method for calculation of the mid-plane field derivatives is used.

DIPOLES	**Dipole magnet N-tuple, polar frame** (i) $B_Z = \sum_{i=1}^{N} B_{Z0,i}\, \mathcal{F}_i(R,\theta)\, \big(1 + b_{1_i}(R-RM_i)/RM_i + b_{2_i}(R-RM_i)^2/RM_i^2 + ...\big)$ (ii) $B_Z = B_{Z0,i} + \sum_{i=1}^{N} \mathcal{F}_i(R,\theta)\, \big(b_{1_i}(R-RM_i) + b_{2_i}(R-RM_i)^2 + ...\big)$		
IL	$IL = 1, 2[\times 10^n]$: print field and coordinates along trajectories.	0-2$[\times 10^n]$	I
N, AT, RM	Number of magnets in the N-tuple ; total angular extent of the dipole ; reference radius	no dim., deg, cm	I, 2*E
Repeat N times the following sequence			
ACN, δRM [1], B_0, ind, b_i, $(i = 1, ind)$	Positioning of EFBs : azimuth, $RM_i = RM + \delta RM$; field ; number of, and field coefficients	deg., cm, kG, $(ind+1)$*no dim.	3*E, I, ind*E
	ENTRANCE FIELD BOUNDARY		
g_0, κ	Fringe field extent ($g = g_0\,(RM/R)^\kappa$) Exponential type fringe field $F = 1\,/\,(1+\exp(P(s)))$ with $P(s) = C_0 + C_1(\frac{s}{g}) + C_2(\frac{s}{g})^2 + ... + C_5(\frac{s}{g})^5$	cm, no dim.	2*E
NC, $C_0 - C_5$, shift	Unused ; C_0 to C_5 : see above ; EFB shift	0-6, 6*no dim., cm	I,7*E
ω^+, θ, R_1, U_1, U_2, R_2	Azimuth of entrance EFB with respect to *ACN* ; wedge angle of EFB ; radii and linear extents of EFB (use $\mid U_{1,2} \mid = \infty$ when $R_{1,2} = \infty$)	2*deg, 4*cm	6*E
	(Note : $g_0 = 0$, ω^+ = *ACENT*, $\theta = 0$ and KIRD=0 for sharp edge)		
	EXIT FIELD BOUNDARY (See ENTRANCE FIELD BOUNDARY)		
g_0, κ		cm, no dim.	2*E
NC, $C_0 - C_5$, shift		$0-6$, 6*no dim., cm	1, 7*E
ω^-, θ, R_1, U_1, U_2, R_2		2*deg, 4*cm	6*E
	(Note : $g_0 = 0$, $\omega^- = -AT+$ *ACENT*, $\theta = 0$ and KIRD=0 for sharp edge)		
	LATERAL FIELD BOUNDARY **to be implemented - following data not used**		
g_0, κ		cm, no dim.	2*E
NC, $C_0 - C_5$, shift		0-6, 6*no dim., cm	1, 7*E
ω^-, θ, R_1, U_1, U_2, R_2, $R3$		2*deg, 5*cm	7*E
End of repeat			

[1] Non-zero δRM requires KIRD= $2, 4$ or 25.

KIRD[.n], Resol	KIRD=0 : analytical computation of field derivatives ; n=0 : default, B_Z formula (i) above, n=1 : B_Z formula (ii). Resol = 2/4 for 2nd/4th order field derivatives computation KIRD=2, 25 or 4 : numerical interpolation of field derivatives ; size of flying interpolation mesh is *XPAS/Resol* KIRD=2 or 25 : second degree, 9- or 25-point grid KIRD=4 : fourth degree, 25-point grid	0, 2, 25 or 4 ; no dim.	I, E
XPAS	Integration step	cm	E
KPOS	Positioning of the magnet, normally 2. Two options :	1-2	I
If KPOS = 2	Positioning as follows :		
RE, TE, RS, TS	Radius and angle of reference, respectively, at entrance and exit of the magnet	cm, rad, cm, rad	4*E
If KPOS = 1	Automatic positioning of the magnet, by means of		
DP	reference relative momentum	no dim.	E

DODECAPO : Dodecapole magnet (Fig. 1)

The meaning of parameters for *DODECAPO* is the same as for *QUADRUPO*.
In fringe field regions the magnetic field $\vec{B}(X,Y,Z)$ and its derivatives up to fourth order are derived from the scalar potential approximated to the 6th order in Y and Z

$$V(X,Y,Z) = G(X)\left(Y^4 - \frac{10}{3}Y^2Z^2 + Z^4\right)YZ$$

$$\text{with } G_0 = \frac{B_0}{R_0^5}$$

The modelling of the fringe field form factor $G(X)$ is described under *QUADRUPO*, p. **??**.

Outside fringe field regions, or everywhere in sharp edge dodecapole ($\lambda_E = \lambda_S = 0$) , $\vec{B}(X,Y,Z)$ in the magnet is given by

$$B_X = 0$$
$$B_Y = G_0(5Y^4 - 10Y^2Z^2 + Z^4)Z$$
$$B_Z = G_0(Y^4 - 10Y^2Z^2 + 5Z^4)Y$$

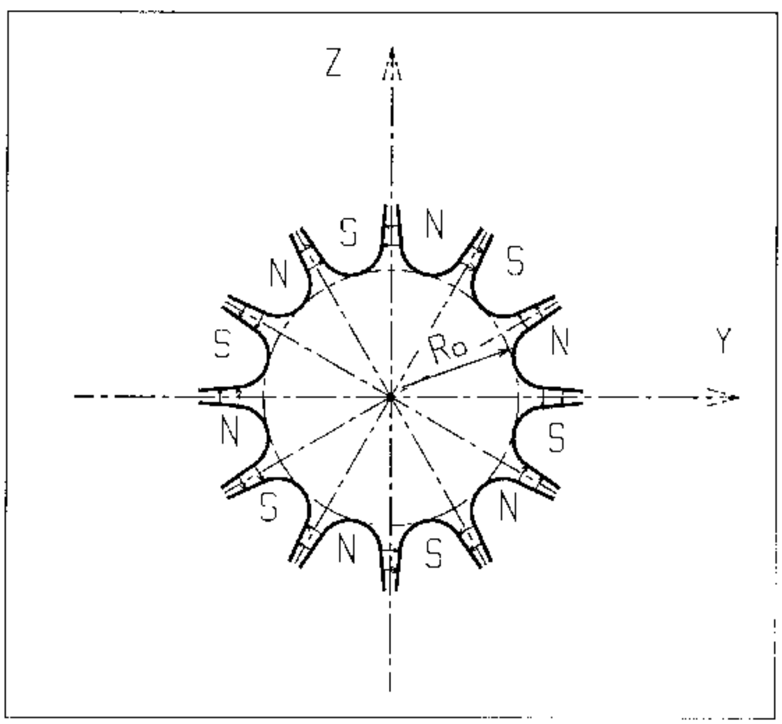
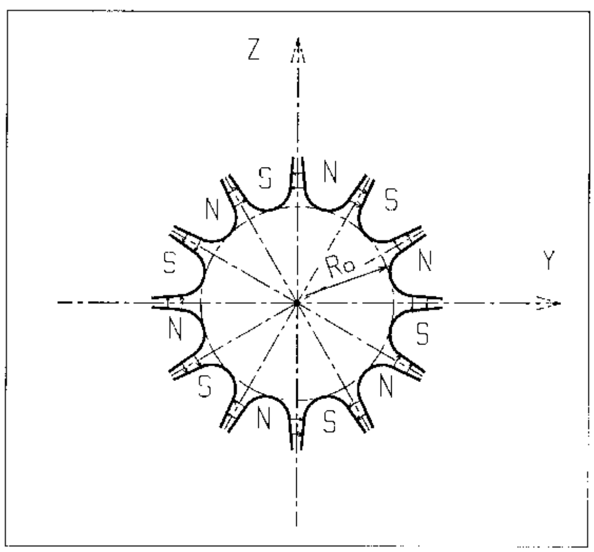


Figure 1: Dodecapole magnet

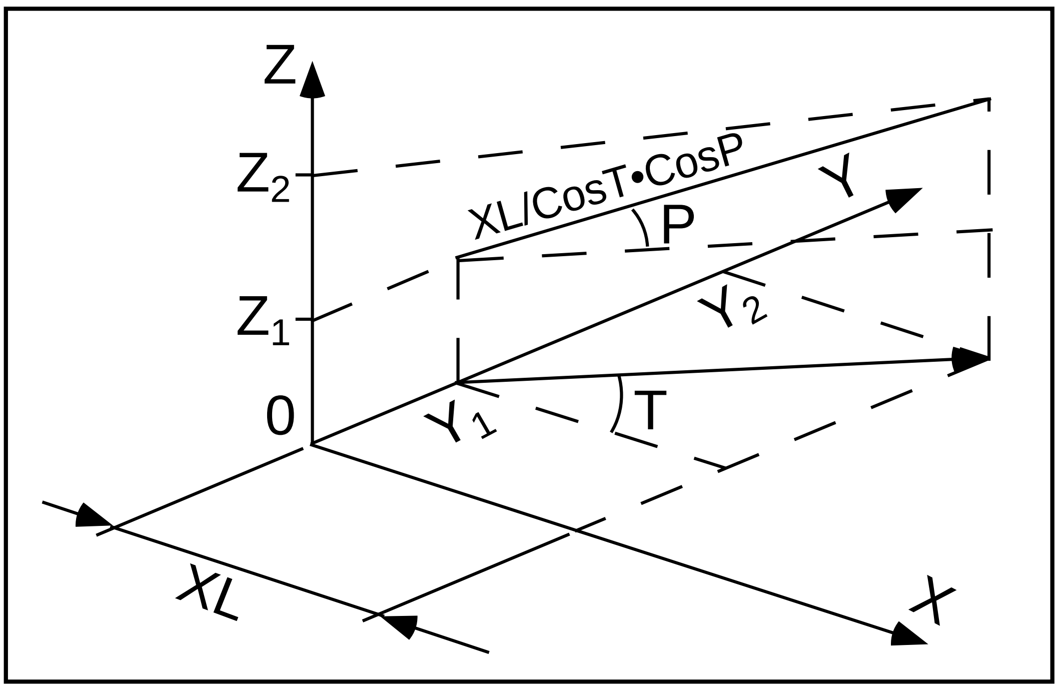
DODECAPO **Dodecapole magnet**

IL	$IL = 1, 2[\times 10^n]$: print field and coordinates along trajectories.	0-2$[\times 10^n]$	I
XL, R_0, B_0	Length ; radius and field at pole tip	2*cm, kG	3*E
X_E, λ_E	Entrance face : Integration zone extent ; fringe field extent ($\lesssim 2R_0$, $\lambda_E = 0$ for sharp edge)	2*cm	2*E
NCE, $C_0 - C_5$	*NCE* = unused $C_0 - C_5$ = Fringe field coefficients such that $G(s) = G_0/(1 + \exp P(s))$, with $G_0 = B_0/R_0^5$ and $P(s) = \sum_{i=0}^{5} C_i(s/\lambda)^i$	unused, 6*no dim.	I, 6*E
X_S, λ_S *NCS*, $C_0 - C_5$	Exit face : see entrance face	2*cm 0-6, 6*no dim.	2*E I, 6*E
XPAS	Integration step	cm	E
KPOS, *XCE*, *YCE*, *ALE*	*KPOS*=1 : element aligned, 2 : misaligned ; shifts, tilt (unused if *KPOS*=1)	1-2, 2*cm, rad	I, 3*E



DRIFT, ESL **Field free drift space**

XL	length	cm	E



EBMULT : Electro-magnetic multipole

EBMULT simulates an electro-magnetic multipole, by addition of electric $(\vec{E})$ and magnetic $(\vec{B})$ multipole components (dipole to 20-pole). $\vec{E}$ and its derivatives $\frac{\partial^{i+j+k}\vec{E}}{\partial X^i \partial Y^j \partial Z^k}$ $(i+j+k \leq 4)$ are derived from the general expression of the multipole scalar potential (eq. **??**), followed by a $\frac{\pi}{2n}$ rotation $(n = 1,2,3,...)$ (see also *ELMULT*). $\vec{B}$ and its derivatives are derived from the same general potential, as described in section **??** (see also *MULTIPOL*).

The entrance and exit fringe fields of the $\vec{E}$ and $\vec{B}$ components are treated separately, in the same way as described under *ELMULT* and *MULTIPOL*, for each one of these two fields. Wedge angle correction is applied in sharp edge field model if $\vec{B}1$ is non zero, as in *MULTIPOL*. Any of the $\vec{E}$ or $\vec{B}$ multipole field component can be X-rotated independently of the others.

Use *PARTICUL* prior to *EBMULT*, for the definition of particle mass and charge.

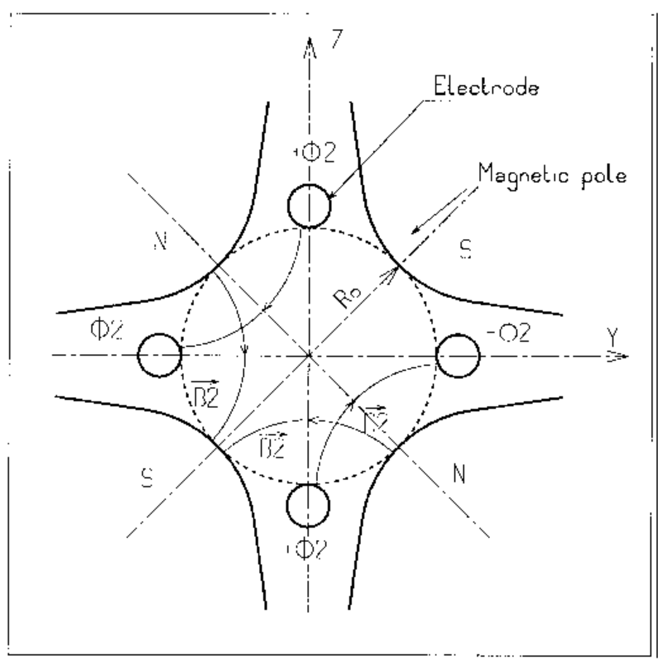
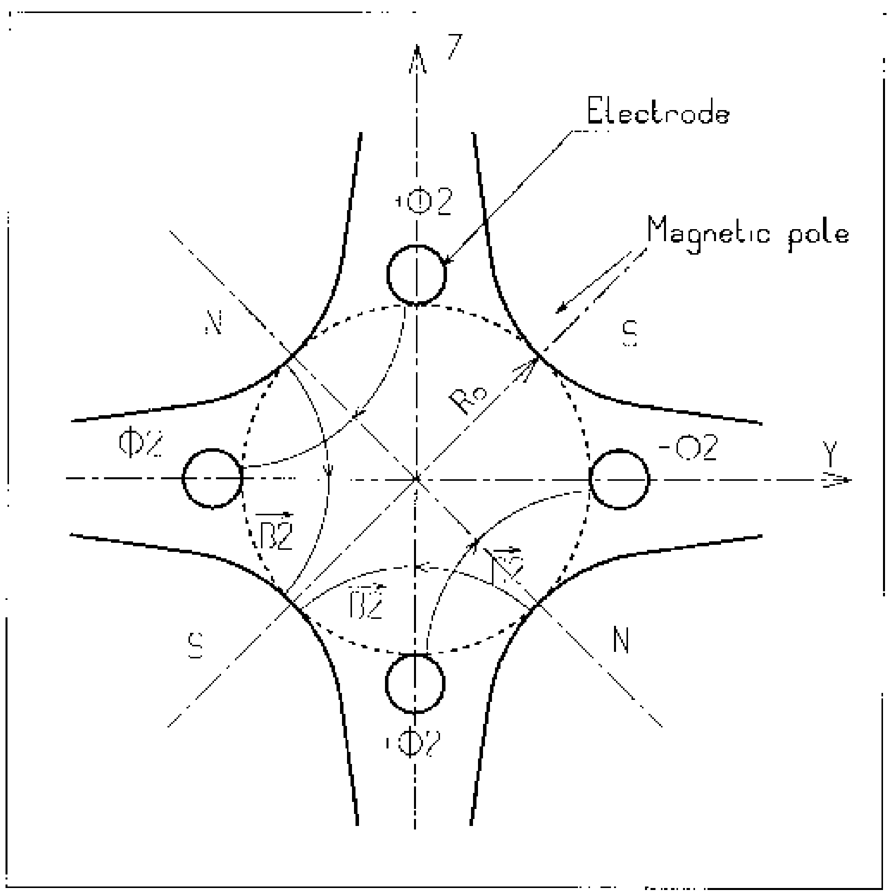


Figure 1: An example of $\vec{E}$, $\vec{B}$ multipole : the achromatic quadrupole
(known for its allowing null second order chromatic aberrations [**?**]).

	Exit face		
X_S, λ_S, S_2, ..., S_{10}	Integration zone ; as for entrance	2*cm, 9*no dim.	11*E
NCS, $C_0 - C_5$		0-6, 6*no dim.	I, 6*E
$R1$, $R2$, $R3$, ..., $R10$	Skew angles of magnetic field components	10*rad	10*E
XPAS	Integration step	cm	E
KPOS, *XCE*, *YCE*, *ALE*	*KPOS*=1 : element aligned, 2 : misaligned ; shifts, tilt (unused if *KPOS*=1)	1-2, 2*cm, rad	I, 3*E



EL2TUB : Two-tube electrostatic lens

The lens is cylindrically symmetric about the X-axis.
The length and potential of the first (resp. second) electrode are $X1$ and $V1$ ($X2$ and $V2$). The distance between the two electrodes is D, and their inner radius is R_0 (Fig. 1). The model for the electrostatic potential along the axis is [**?**]

$$V(X) = \frac{V_2 - V_1}{2}\,\mathrm{th}\,\frac{\omega x}{R_0}\left[+\frac{V_1+V_2}{2}\right] \qquad \text{if } D = 0$$

$$V(X) = \frac{V_2 - V_1}{2}\,\frac{1}{2\omega D/R_0}\,\ell n\,\frac{\mathrm{ch}\,\omega\,\dfrac{x+D}{R_0}}{\mathrm{ch}\,\omega\,\dfrac{x-D}{R_0}}\left[+\frac{V_1+V_2}{2}\right] \qquad \text{if } D \neq 0$$

(x = distance from half-way between the electrodes ; ω = 1.318 ; th = hyperbolic tangent ; ch = hyperbolic cosine) from which the field $\vec{E}(X,Y,Z)$ and its derivatives are derived following the procedure described in section **??** (note that they don't depend on the constant term $\left[\dfrac{V_1+V_2}{2}\right]$ which disappears when differentiating).

Use *PARTICUL* prior to *EL2TUB*, for the definition of particle mass and charge.

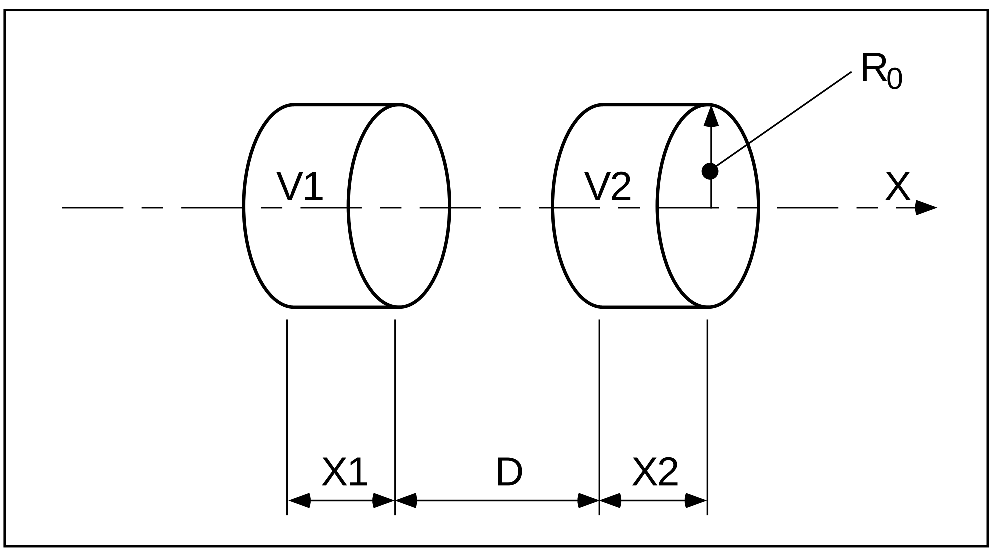
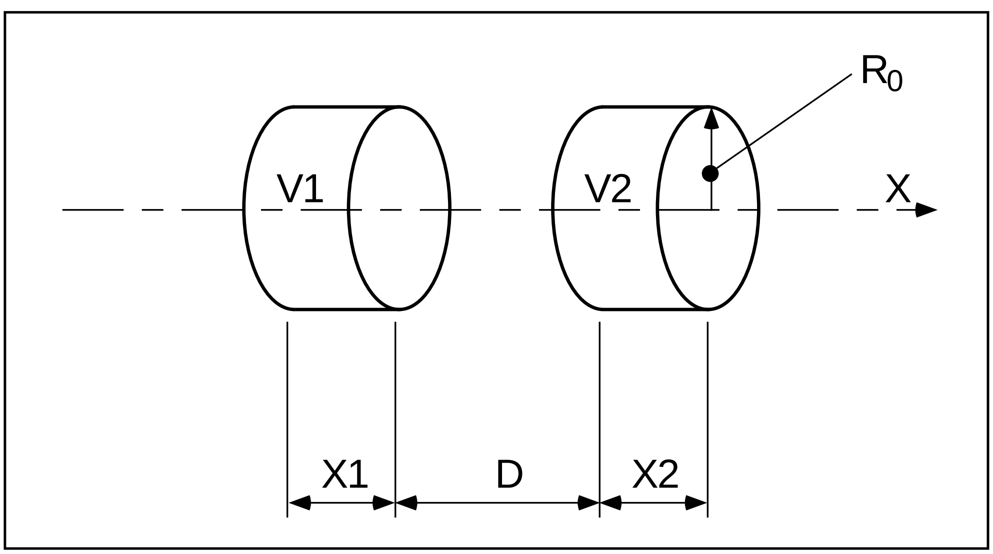


Figure 1: Two-electrode cylindrical electric lens.

EL2TUB [1]	**Two-tube electrostatic lens**		
IL	$IL = 1, 2[\times 10^n]$: print field and coordinates along trajectories.	0-2$[\times 10^n]$	I
X_1, D, X_2, R_0	Length of first tube ; distance between tubes ; length of second tube ; inner radius	3*m	4*E
V_1, V_2	Potentials	2*V	2*E
XPAS	Integration step	cm	E
KPOS, *XCE*, *YCE*, *ALE*	*KPOS*=1 : element aligned, 2 : misaligned ; shifts, tilt (unused if *KPOS*=1)	1-2, 2*cm, rad	I, 3*E



Two-electrode cylindrical electric lens.

[1] Use *PARTICUL* to declare mass and charge.

ELMIRC : Electrostatic N-electrode mirror/lens, circular slits [?]

The device works as mirror or lens, horizontal or vertical. It is made of N 2-plate electrodes and has mid-plane symmetry[1].

Electrode slits are circular, concentric with radii $R1$, $R2$, ..., RN-1, D is the mirror/lens gap. The model for the mid-plane ($Z = 0$) radial electrostatic potential is (after Ref. [?, p.443])

$$V(r) = \sum_{i=2}^{N} \frac{Vi - Vi{-1}}{\pi} \arctan\left(\sinh \frac{\pi(r - Ri{-1})}{D}\right)$$

where Vi are the potential at the N electrodes (and normally $V1 = 0$ refers to the incident beam energy). r is the current radius.

The mid-plane field $\vec{E}(r)$ and its r-derivatives are first derived by differentiation, then $\vec{E}(r, Z)$ and derivatives are obtained from Taylor expansions and Maxwell relations. Eventually a transformation to the rotating frame provides $\vec{E}(X, Y, Z)$ and derivatives as involved in eq. **??**.
Stepwise integration starts at entrance (defined by RE, TE) of the first electrode and terminates when rotation of the reference rotating frame (RM, X, Y) has reached the value AT. Normally, $R1 - RE$ and $R1 - RS$ should both exceed $3D$ (so that potential tails have negligible effect in terms of trajectory behavior).

Positioning of the element is performed by means of *KPOS* (see section **??**).

Use *PARTICUL* prior to *ELMIRC*, for the definition of particle mass and charge.

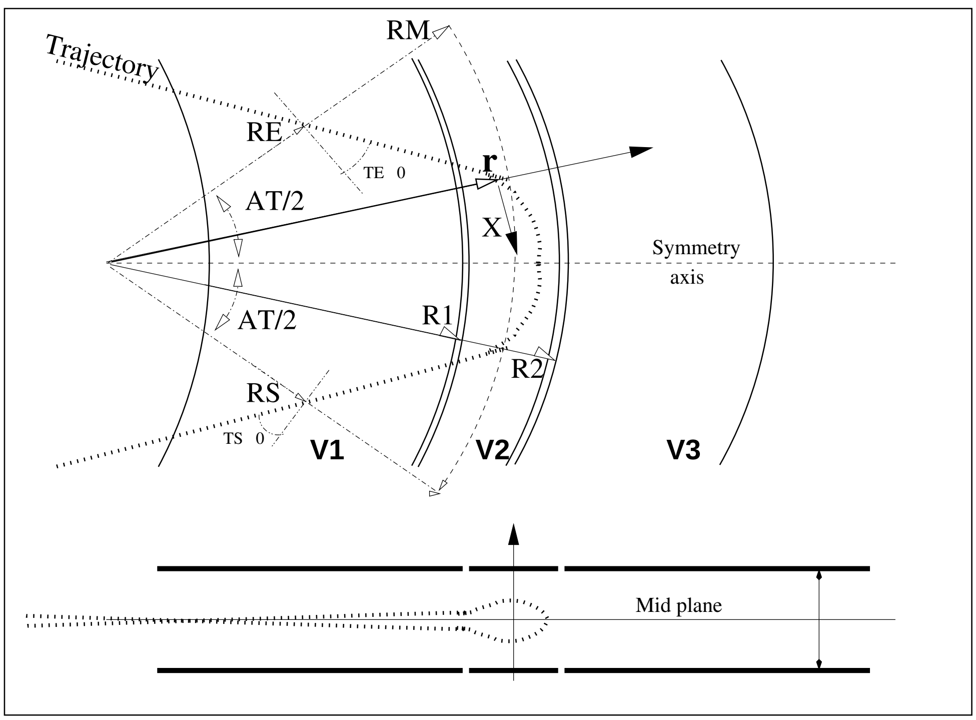
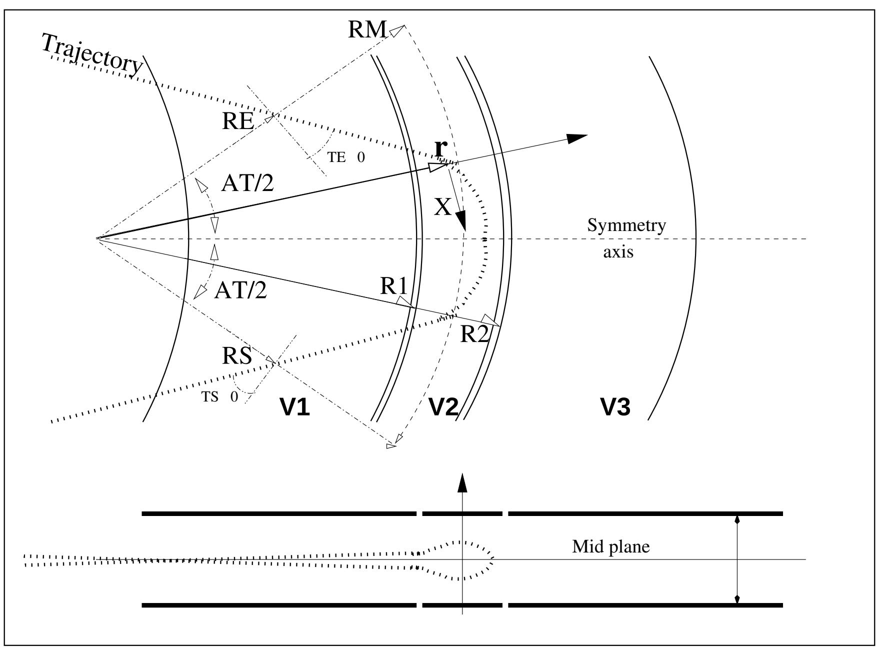


Figure 1: Electrostatic N-electrode mirror/lens, circular slits, in the case $N = 3$, in horizontal mirror mode.

[1]NOTE : in the present version of the code, the sole horizontal mirror mode is operational, and N is limited to 3.

ELMIRC **Electrostatic N-electrode mirror/lens, circular slits**

IL	$IL = 1, 2[\times 10^n]$: print field and coordinates along trajectories.	0-2$[\times 10^n]$	I
$R1$, $R2$, AT, D	Radius of first and second slits ; total deviation angle ; gap	4*m 2*m, rad, m	4*E 4*E
$V - VA$, $VB - V$	Potential difference	2*V	2*E
XPAS	Integration step	cm	E
KPOS	Normally $KPOS = 2$ for positioning ;	1-2	I
RE, TE, RS, TS	Radius and angle at respectively entrance and exit.	cm, rad, cm, rad	4*E



Electrostatic N-electrode mirror/lens, circular slits, in the case $N = 3$, in horizontal mirror mode.

ELMULT : Electric multipole

The simulation of multipolar electric field $\vec{M}_E$ proceeds by addition of the dipolar $(\vec{E}1)$, quadrupolar $(\vec{E}2)$, sextupolar $(\vec{E}3)$, etc., up to 20-polar $(\vec{E}10)$ components, and of their derivatives up to fourth order, following

$$\vec{M}_E = \vec{E}1 + \vec{E}2 + \vec{E}3 + \ ... \ + \vec{E}10$$
$$\frac{\partial \vec{M}_E}{\partial X} = \frac{\partial \vec{E}1}{\partial X} + \frac{\partial \vec{E}2}{\partial X} + \frac{\partial \vec{E}3}{\partial X} + \ ... \ + \frac{\partial \vec{E}10}{\partial X}$$
$$\frac{\partial^2 M_E}{\partial X \partial Z} = \frac{\partial^2 \vec{E}1}{\partial X \partial Z} + \frac{\partial^2 \vec{E}2}{\partial X \partial Z} + \frac{\partial^2 \vec{E}3}{\partial X \partial Z} + \ ... \ + \frac{\partial^2 \vec{E}10}{\partial X \partial Z}$$
etc.

The independent components $\vec{E}1$ to $\vec{E}10$ and their derivatives up to the fourth order are calculated by differentiating the general multipole potential given in eq. **??** (page **??**), followed by a $\frac{\pi}{2n}$ rotation about the X-axis, so that the so defined right electric multipole of order n, and of strength [**?**, **?**]

$$K_n = \frac{1}{2} \frac{\gamma}{\gamma^2 - 1} \frac{V_n}{R_0^n}$$

(V_n = potential at the electrode, R_0 = radius at pole tip, γ = relativistic Lorentz factor of the particle) has the same focusing effect as the right magnetic multipole of order n and strength $K_n = \frac{B_n}{R_0^{n-1} B\rho}$ (B_n = field at pole tip, $B\rho$ = particle rigidity, see *MULTIPOL*).

The entrance and exit fringe fields are treated separately. They are characterized by the integration zone X_E at entrance and X_S at exit, as for *QUADRUPO*, and by the extent λ_E at entrance, λ_S at exit. The fringe field extents for the dipole component are λ_E and λ_S. The fringe field extent for the quadrupolar (sextupolar, ..., 20-polar) component is given by a coefficient E_2 (E_3, ..., E_{10}) at entrance, and S_2 (S_3, ..., S_{10}) at exit, such that the fringe field extent is $\lambda_E * E_2$ ($\lambda_E * E_3$, ..., $\lambda_E * E_{10}$) at entrance and $\lambda_S * S_2$ ($\lambda_S * S_3$, ..., $\lambda_S * S_{10}$) at exit.

If $\lambda_E = 0$ ($\lambda_S = 0$) the multipole lens is considered to have a sharp edge field at entrance (exit), and then, X_E (X_S) is forced to zero (for the mere purpose of saving computing time).

If $E_i = 0$ ($S_i = 0$) ($i = 2,\ 10$), the entrance (exit) fringe field for multipole component i is considered as a sharp edge field.

Any multipole component $\vec{E}i$ can be rotated independently by an angle RXi around the longitudinal X-axis, for the simulation of positioning defects, as well as skew lenses.
Use *PARTICUL* prior to *ELMULT*, for the definition of particle mass and charge.

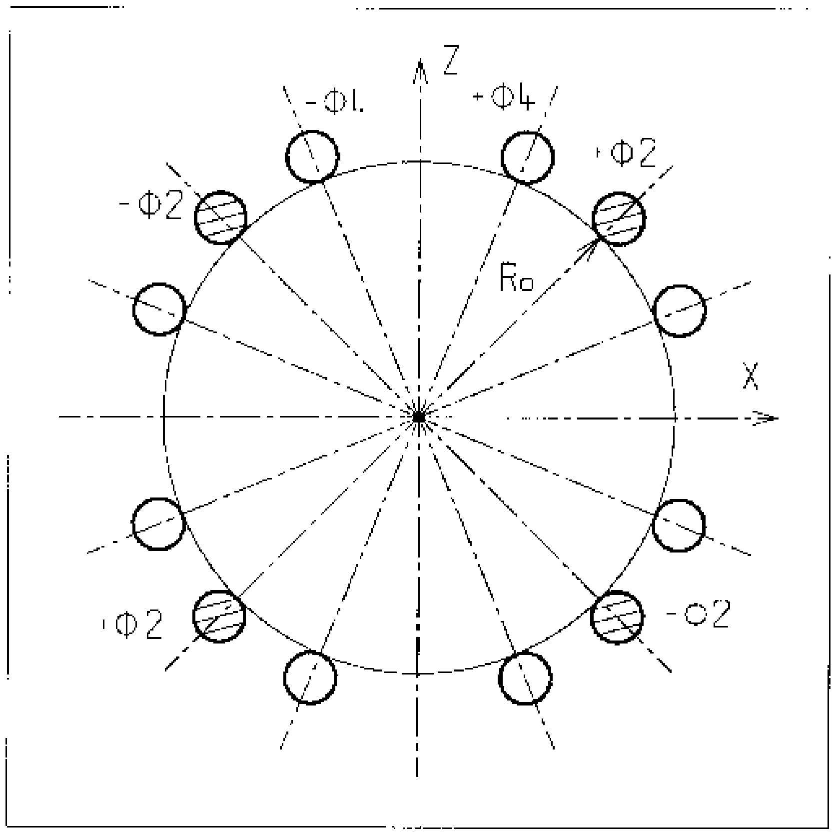
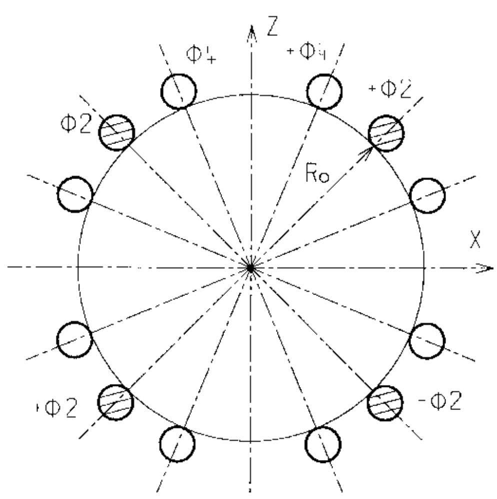


Figure 1: An electric multipole combining skew-quadrupole ($\vec{E}2 \neq \vec{0},\ \vec{R}2 = \pi/4$) and skew-octupole ($\vec{E}4 \neq \vec{0},\ \vec{R}4 = \pi/8$) components ($\vec{E}1 = \vec{E}3 = \vec{E}5 = \ ... \ = \vec{E}10 = \vec{0}$) [**?**].

ELMULT [1] **Electric multipole**

IL	$IL = 1, 2[\times 10^n]$: print field and coordinates along trajectories.	0-2$[\times 10^n]$	I
XL, R_0, $E1$, $E2$, ..., $E10$	Length of element ; radius at pole tip ; field at pole tip for dipole, quadrupole, ..., dodecapole components	2*cm, 10*V/m	12*E
	Entrance face		
X_E, λ_E, E_2, ..., E_{10}	Integration zone ; fringe field extent : dipole fringe field extent = λ_E ; quadrupole fringe field extent = $\lambda_E * E_2$; ... 20-pole fringe field extent = $\lambda_E * E_{10}$ (sharp edge if field extent is zero)	2*cm, 9*no dim.	11*E
NCE, $C_0 - C_5$	same as *QUADRUPO*	0-6, 6*no dim.	I, 6*E
	Exit face		
X_S, λ_S, S_2, ..., S_{10}	Integration zone ; as for entrance	2*cm, 9*no dim.	11*E
NCS, $C_0 - C_5$		0-6, 6*no dim.	I, 6*E
$R1$, $R2$, $R3$, ..., $R10$	Skew angles of field components	10*rad	10*E
XPAS	Integration step	cm	E
KPOS, *XCE*, *YCE*, *ALE*	*KPOS*=1 : element aligned, 2 : misaligned ; shifts, tilt (unused if *KPOS*=1)	1-2, 2*cm, rad	I, 3*E



[1] Use *PARTICUL* to declare mass and charge.

ELREVOL : 1-D uniform mesh electric field map

ELREVOL reads a 1-D axial field map from a storage data file, whose content must fit the following *FORTRAN* reading sequence

```
OPEN (UNIT = NL, FILE = FNAME, STATUS = 'OLD' [,FORM='UNFORMATTED'])
DO 1 I=1, IX
IF (BINARY) THEN
READ(NL) X(I), EX(I)
ELSE
READ(NL,*) X(I), EX(I)
ENDIF
1     CONTINUE
```

where IX is the number of nodes along the (symmetry) X-axis, $X(I)$ their coordinates, and $EX(I)$ are the values of the X component of the field. EX is normalized with *ENORM* prior to ray-tracing. As well the longitudinal coordinate X is normalized with a *XNORM* coefficient (useful to convert to centimeters, the working units in **zgoubi**).

X-cylindrical symmetry is assumed, resulting in EY and EZ taken to be zero on axis. $\vec{E}(X,Y,Z)$ and its derivatives along a particle trajectory are calculated by means of a 5-points polynomial interpolation followed by second order off-axis extrapolation (see sections **??** and **??**).

Entrance and/or exit integration boundaries may be defined in the same way as in *CARTEMES* by means of the flag ID and coefficients A, B, C, A', B', C'.

Use *PARTICUL* prior to *ELREVOL*, for the definition of particle mass and charge.

ELREVOL [1]	**1-D uniform mesh electric field map** X-axis cylindrical symmetry is assumed		
IC, *IL*	$IC = 1, 2$: print the map $IL = 1, 2[\times 10^n]$: print field and coordinates along trajectories.	0-2, 0-2$[\times 10^n]$	2*I
ENORM, X-NORM	Field and X-coordinate normalization coeff.	2*no dim.	2*E
TITL	Title. Start with "FLIP" to get field map X-flipped.		A80
IX	Number of longitudinal nodes of the map	≤ 400	I
FNAME [2]	File name		A80
ID, A, B, C [, A', B', C', B'', etc., if $ID \geq 2$]	Integration boundary. Ineffective when $ID = 0$. ID = -1, 1 or $\geq$ 2 : as for *CARTEMES*	≥ -1, 2*no dim., cm [,2*no dim., cm, etc.]	I,3*E [,3*E,etc.]
IORDRE	Unused	2, 25 or 4	I
XPAS	Integration step	cm	E
KPOS, XCE, *YCE, ALE*	*KPOS*=1 : element aligned, 2 : misaligned ; shifts, tilt (unused if *KPOS*=1)	1-2, 2*cm, rad	I, 3*E

[1] Use *PARTICUL* to declare mass and charge.

[2] *FNAME* (*e.g.*, e-lens.map) contains the field data. These must be formatted according to the following *FORTRAN* sequence :

```
OPEN (UNIT = NL, FILE = FNAME, STATUS = 'OLD' [,FORM='UNFORMATTED'])
DO 1 I = 1, IX
IF (BINARY) THEN
READ(NL) X(I), EX(I)
ELSE
READ(NL,*) X(I), EX(I)
ENDIF
1    CONTINUE
```

where $X(I)$ and $EX(I)$ are the longitudinal coordinate and field component at node (I) of the mesh. Binary file names *FNAME* must begin with 'B_' or 'b_'. 'Binary' will then automatically be set to '.TRUE.'

EMMA : 2-D Cartesian or cylindrical mesh field map for EMMA FFAG

EMMA is dedicated to the reading and treatment of 2-D or 3-D Cartesian mesh field maps representing the EMMA FFAG cell quadrupole doublet[1] [**?**, **?**].

EMMA can sum up independent field maps of each of the two quadrupoles, with each its scaling coefficient. The two maps can be radially positioned independently of one another at Y_F, Y_D respectively, just like the actual *EMMA* quadrupoles. In particular,

MOD : operational and map *FORMAT* reading mode ;
MOD$\leq$19 : Cartesian mesh ;
MOD$\geq$20 : cylindrical mesh.

MOD=0 : two 2D maps, one representing QF, one reprensenting QD. A single map, superimposition of both, is built prior to tracking and used for tracking.
MOD=1 : two 2D maps, one representing QF, one reprensenting QD, a resulting single map is devised in the following way : QF_new is interpolated from QF with dr=xF, QD_new is interpolated from QD with dr=xD. A single map, superimposition of both, is built prior to tracking and used for tracking.

The parameters that move/position the maps, as (Y_F, Y_D), are accessible from the FIT, allowing to adjust the cell tunes.

EMMA works much like *TOSCA*. Refer to that keyword, and to the *FORTRAN* file `emmac.f`, for details.

[1] The stepwise ray-tracing code Zgoubi is the on-line model code for the worlds first non-scaling FFAG experiment.

EMMA **2-D Cartesian or cylindrical mesh field map for EMMA FFAG**

IC, IL	see *CARTEMES*	0-2, 0-2$[\times 10^n]$	2*I
BNORM, XN,YN, ZN	Field and X-,Y-,Z-coordinate normalization coefficients	4*no dim.	4*E
TITL	Title. Start with "FLIP" to get field map X-flipped		A80
IX, IY, IZ, *MOD[.i]*	Number of nodes of the mesh in the X, Y and Z directions, $IZ = 1$ for single 2-D map ; *MOD* : operational and map *FORMAT* reading mode [1] *MOD*$\leq$19 : Cartesian mesh ; *MOD*$\geq$20 : cylindrical mesh ; *.i*, optional, tells the reading *FORMAT*, default is '*'.	≤ 400, ≤ 200, 1, $\geq$ 0[.1-9]	3*I
FNAME [1] $(K = 1, NF)$	Names of the NF files that contain the 2-D maps, ordered from $Z(1)$ to $Z(NF)$. If *MOD*=0 : a single map, superimposition of QF and QD ones, is built for tracking. If *MOD*=1 : a single map, *interpolated* from QF[x_F] and QD[x_D] ones, is built for tracking. If *MOD*=22 : a single map, superimposition of QF and QD ones, is built for tracking. If *MOD*=24 : field at particle is interpolated from a (QF,QD) pair of maps, closest to current (x_F, x_D) value, taken from of set of (QF,QD) pairs registered in FNAME...		A80
ID, A, B, C [, A', B', C', B'', etc., if $ID \geq 2$]	Integration boundary. Ineffective when $ID = 0$. ID = -1, 1 or ≥ 2 : as for *CARTEMES*	≥ -1, 2*no dim., cm [,2*no dim., cm, etc.]	I,3*E [,3*E,etc.]
IORDRE	If $IZ = 1$: as in *CARTEMES* If $IZ \neq 1$: unused	2, 25 or 4	I
XPAS	Integration step	cm	E
KPOS, XCE, *YCE, ALE*	*KPOS*=1 : element aligned, 2 : misaligned ; shifts, tilt (unused if *KPOS*=1)	1-2, 2*cm, rad	I, 3*E

[1] *FNAME* normally contains the field map data. If *MOD*=24 *FNAME(K)* contains the names of the QF maps and QD maps, as well as the QF-QD distance attached to each one of these pairs.

FAISCEAU **Print particle coordinates**

Print particle coordinates at the location where the keyword is introduced in the structure.

FAISCNL **Store particle coordinates in file FNAME**

FNAME[1]	Name of storage file (*e.g.*, zgoubi.fai, or b_zgoubi.fai for binary storage).	A80

FAISTORE **Store coordinates every IP other pass [, at elements with appropriate label]**

FNAME [1] [,*LABEL(s)*]	Name of storage file (*e.g.* zgoubi.fai) [; label(s) of the element(s) at the exit of which the store occurs (10 labels maximum)]. If either *FNAME* or first *LABEL* is 'none' then no storage occurs. Store occurs at all elements if first *LABEL* is 'all' or 'ALL'.	A80, [, 10*A10]
IP	Store every IP other pass (when using *REBELOTE* with *NPASS* $\geq IP - 1$).	I

[1] Stored data can be read back from *FNAME* using *OBJET*, *KOBJ* = 3.

FFAG : FFAG magnet, N-tuple [?, ?]

FFAG works much like *DIPOLES* as to the field modelling, apart from the radial dependence of the field, $B = B_0(r/r_0)^k$, so-called "scaling". Note that *DIPOLES* does similar job by using a Taylor r-expansion of $B_0(r/r_0)^k$.

The *FFAG* procedure allows overlapping of fringe fields of neighboring dipoles, thus simulating in some sort the field in a dipole N-tuple - as for instance in an FFAG doublet or triplet. A detailed application, with five dipoles, can be found in Ref. [**?**]. This is done in the way described below.

The dimensioning of the magnet is defined by

AT	:	total angular aperture
RM	:	mean radius used for the positioning of field boundaries

For each one of the $N = 1$ to (maximum) 5 dipoles of the N-tuple, the two effective field boundaries (entrance and exit EFBs) from which the dipole field is drawn are defined from geometric boundaries, the shape and position of which are determined by the following parameters (in the same manner as in *DIPOLE*, *DIPOLE-M*) (see Fig. **??**-A page **??**, and Fig. 1)

ACN_i	:	arbitrary inner angle, used for EFB's positioning
ω	:	azimuth of an EFB with respect to *ACN*
θ	:	angle of an EFB with respect to its azimuth (wedge angle)
R_1, R_2	:	radius of curvature of an EFB
U_1, U_2	:	extent of the linear part of an EFB

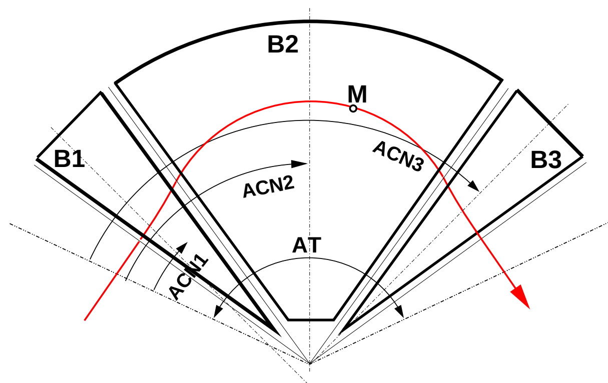


Figure 1: Definition of a dipole N-tuple ($N = 3$, a triplet here) using the *DIPOLES* or *FFAG* procedures.

Calculation of the Field From a Single Dipole The magnetic field is calculated in polar coordinates. At all (R, θ) in the median plane ($z = 0$), the magnetic field due a single one (index i) of the dipoles of a N-tuple FFAG magnet is written

$$B_{Z\,i}(R,\theta) = B_{Z0,i}\,\mathcal{F}_i(R,\theta)\;(R/R_M)^{K_i}$$

wherein $B_{Z0,i}$ is a reference field, at reference radius RM_i, whereas $\mathcal{F}(R,\theta)$ is calculated as described below.

Calculation of $\mathcal{F}_i(R,\theta)$ The fringe field coefficient $\mathcal{F}_i(R,\theta)$ associated with a dipole is computed as in the procedure *DIPOLES* (eq. **??**), including (rigorously if the interpolation method is used, see page **??**, or to order zero if the analytic method is used, see page **??**) radial dependence of the gap size

$$g(R) = g_0\,(RM/R)^{\kappa} \tag{1}$$

so to simulate the effect of gap shaping on $B_{Z\,i}(R,\theta)|_R$ field fall-off, over the all radial extent of a scaling FFAG dipole (with normally - but not necessarily in practice - $\kappa \approx K_i$).

Calculation of the Field Resulting From All N Dipoles For the rest, namely, calculation of the full field at particle position from the N dipoles, analytical calculation or numerical interpolation of the mid-plane field derivatives, extrapolation off median plane, etc., things are performed exactly as in the case of the *DIPOLES* procedure (see page **??**).

Sharp Edge Sharp edge field fall-off at a field boundary can only be simulated if the following conditions are fulfilled :

- entrance (resp. exit) field boundary coincides with entrance (resp. exit) dipole limit (it means in particular, see Fig. **??**, $\omega^+ = ACENT$ (resp. $\omega^- = -(AT - ACENT)$), together with $\theta = 0$ at entrance (resp. exit) EFBs),
- analytical method for calculation of the mid-plane field derivatives is used.

FFAG	**FFAG magnet, N-tuple** **UNDER DEVELOPMENT** $B_Z = \sum_{i=1}^{N} B_{Z0,i}\, \mathcal{F}_i(R,\theta)\ (R/R_{M,i})^{K_i}$		
IL	$IL = 1, 2[\times 10^n]$: print field and coordinates along trajectories.	0-2$[\times 10^n]$	I
N, AT, RM	Number of dipoles in the FFAG N-tuple ; total angular extent of the dipole ; reference radius	no dim., deg, cm	I, 2*E
Repeat N times the following sequence ________________________			
ACN, δRM, B_{Z0}, K	Azimuth for dipole positioning ; $R_{M,i} = RM + \delta RM$; field at $R_{M,i}$; index	deg, cm, kG, no dim.	4*E
	ENTRANCE FIELD BOUNDARY		
g_0, κ	Fringe field extent ($g = g_0\, (RM/R)^{\kappa}$)	cm, no dim.	2*E
NC, $C_0 - C_5$, shift	Unused ; C_0 to C_5 : fringe field coefficients ; EFB shift	0-6, 6*no dim, cm	I,7*E
ω^+, θ, R_1, U_1, U_2, R_2	Azimuth of entrance EFB with respect to *ACN* ; wedge angle of EFB ; radii and linear extents of EFB (use $\mid U_{1,2} \mid = \infty$ when $R_{1,2} = \infty$)	2*deg, 4*cm	6*E
	(Note : $g_0 = 0$, ω^+ = *ACENT*, $\theta = 0$ and KIRD=0 for sharp edge)		
	EXIT FIELD BOUNDARY (See ENTRANCE FIELD BOUNDARY)		
g_0, κ		cm, no dim	2*E
NC, $C_0 - C_5$, shift		0-6, 6*no dim, cm	1, 7*E
ω^-, θ, R_1, U_1, U_2, R_2		2*deg, 4*cm	6*E
	(Note : $g_0 = 0$, $\omega^- = -AT+$ *ACENT*, $\theta = 0$ and KIRD=0 for sharp edge)		
	LATERAL FIELD BOUNDARY **to be implemented - following data not used**		
g_0, κ		cm, no dim	2*E
NC, $C_0 - C_5$, shift		0-6, 6*no dim, cm	1, 7*E
ω^-, θ, R_1, U_1, U_2, R_2		2*deg, 4*cm	6*E
End of repeat ____			
KIRD, Resol	KIRD=0 : analytical computation of field derivatives ; Resol = 2/4 for 2nd/4th order field derivatives computation KIRD$2, 4$ or 25 : numerical interpolation of field derivatives ; size of flying interpolation mesh is *XPAS/Resol* KIRD=2 or 25 : second degree, 9- or 25-point grid KIRD=4 : fourth degree, 25-point grid	0, 2, 25 or 4 ; no dim.	I, E
XPAS	Integration step	cm	E
KPOS	Positioning of the magnet, normally 2. Two options :	1-2	I
If KPOS = 2 RE, TE, RS, TS	Positioning as follows : Radius and angle of reference, respectively, at entrance and exit of the magnet	 cm, rad, cm, rad	 4*E
If KPOS = 1 DP	Automatic positioning of the magnet, by means of reference relative momentum	 no dim.	 E

FFAG-SPI : Spiral FFAG magnet, N-tuple [?, ?]

FFAG-SPI works much like *FFAG* as to the field modelling, apart from the axial dependence of the field.

The *FFAG* procedure allows overlapping of fringe fields of neighboring dipoles, thus simulating in some sort the field in a dipole N-tuple - as for instance in an FFAG doublet or triplet (Fig. 1). This is done in the way described below.

The dimensioning of the magnet is defined by

AT : total angular aperture
RM : mean radius used for the positioning of field boundaries

For each one of the $N = 1$ to (maximum) 5 dipoles of the N-tuple, the two effective field boundaries (entrance and exit EFBs) from which the dipole field is drawn are defined from geometric boundaries, the shape and position of which are determined by the following parameters

ACN_i : arbitrary inner angle, used for EFB's positioning
ω : azimuth of an EFB with respect to *ACN*
ξ : spiral angle

with ACN_i and ω as defined in Fig. 1 (similar to what can be found in Figs. **??** and **??**-A).

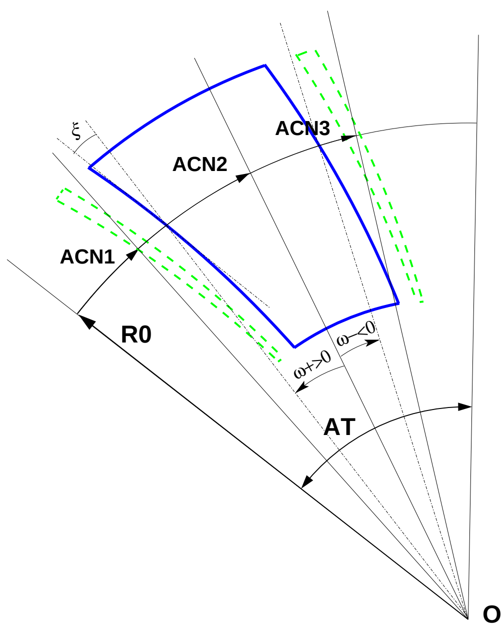


Figure 1: A N-tuple spiral sector FFAG magnet ($N = 3$ here, simulating active field clamps at entrance and exit side of a central dipole).

Calculation of the Field From a Single Dipole The magnetic field is calculated in polar coordinates. At all (R, θ) in the median plane ($Z = 0$), the magnetic field due a single one (index i) of the dipoles of a N-tuple FFAG magnet is written

$$B_{Z i}(R, \theta) = B_{Z0,i}\, \mathcal{F}_i(R, \theta)\ (R/R_M)^{K_i}$$

wherein $B_{Z0,i}$ is a reference field, at reference radius RM_i, whereas $\mathcal{F}(R, \theta)$ is calculated as described below.

Calculation of $\mathcal{F}_i(R, \theta)$ The fringe field coefficient $\mathcal{F}_i(R, \theta)$ associated with a dipole is computed as in the procedure *DIPOLES* (eq. **??**), including radial dependence of the gap size

$$g(R) = g_0\, (RM/R)^{\kappa} \tag{1}$$

so to simulate the effect of gap shaping on $B_{Z i}(R, \theta)|_R$ field fall-off, over the all radial extent of the dipole (with normally - yet not necessarily in practice - $\kappa \approx K_i$).

Calculation of the Full Field From All N Dipoles For the rest, namely calculation of the full field at particle position, as resulting from the N dipoles, calculation of the mid-plane field derivatives, extrapolation off median plane, etc., things are performed in the same manner as for the *DIPOLES* procedure (see page **??**).

FFAG-SPI	**Spiral FFAG magnet, N-tuple** **UNDER DEVELOPMENT** $B_Z = \sum_{i=1}^{N} B_{Z0,i}\,\mathcal{F}_i(R,\theta)\ (R/R_{M,i})^{K_i}$		
IL	$IL = 1, 2[\times 10^n]$: print field and coordinates along trajectories.	0-2$[\times 10^n]$	I
N, AT, RM	Number of dipoles in the FFAG N-tuple ; total angular extent of the dipole ; reference radius	no dim., deg, cm	I, 2*E
Repeat N times the following sequence			
ACN, δRM, B_{Z0}, K	Azimuth for dipole positioning ; $R_{M,i} = RM + \delta RM$; field at $R_{M,i}$; index	deg, cm, kG, no dim.	4*E
	ENTRANCE FIELD BOUNDARY		
g_0, κ	Fringe field extent ($g = g_0\,(RM/R)^{\kappa}$)	cm, no dim.	2*E
NC, $C_0 - C_5$, shift	Unused ; C_0 to C_5 : fringe field coefficients ; EFB shift	0-6, 6*no dim, cm	I,7*E
ω^+, ξ, 4 dummies	Azimuth of entrance EFB with respect to *ACN* ; spiral angle ; 4 × unused	2*deg, 4*unused	6*E
	EXIT FIELD BOUNDARY (See ENTRANCE FIELD BOUNDARY)		
g_0, κ		cm, no dim	2*E
NC, $C_0 - C_5$, shift		0-6, 6*no dim, cm	1, 7*E
ω^-, ξ, 4 dummies		2*deg, 4*unused	6*E
	LATERAL FIELD BOUNDARY **to be implemented - following data not used**		
g_0, κ		cm, no dim	2*E
NC, $C_0 - C_5$, shift		0-6, 6*no dim, cm	1, 7*E
ω^-, θ, R_1, U_1, U_2, R_2		2*deg, 4*cm	6*E
End of repeat			
KIRD, Resol	KIRD=0 : analytical computation of field derivatives ; Resol = 2/4 for 2nd/4th order field derivatives computation KIRD$2, 4$ or 25 : numerical interpolation of field derivatives ; size of flying interpolation mesh is *XPAS/Resol* KIRD=2 or 25 : second degree, 9- or 25-point grid KIRD=4 : fourth degree, 25-point grid	0, 2, 25 or 4 ; no dim.	I, E
XPAS	Integration step	cm	E
KPOS	Positioning of the magnet, normally 2. Two options :	1-2	I
If KPOS = 2 RE, TE, RS, TS	Positioning as follows : Radius and angle of reference, respectively, at entrance and exit of the magnet	 cm, rad, cm, rad	 4*E
If KPOS = 1 DP	Automatic positioning of the magnet, by means of reference relative momentum	 no dim.	 E

FIN, END **End of input data list**

Any information in zgoubi.dat following these keywords will be ignored

FIT, FIT2 : Fitting procedure

The keywords *FIT*, *FIT2* allow the automatic adjustment of up to 20 variables, for fitting up to 20 constraints.

FIT was drawn from the matrix transport code BETA [**?**]. *FIT2* is a simplex method (Nelder-Mead method), it has been implemented next [**?**]. One or the other may converge faster, or may have some advantages/disadvantages, depending on the problem.

Any physical parameter of any element in the zgoubi.dat data list may be varied. Examples of available constraints are, amongst others :
- trajectory coordinates in the $F(J,I)$ array, I = particle number, J = coordinate number = 1 to 7 for respectively D, Y, T, Z,P, S =path length, time ;
- spin coordinates ;
- any of the 6×6 coefficients of the first order transfer matrix $[R_{ij}]$ as defined in the keyword *MATRIX* ;
- any of the $6 \times 6 \times 6$ coefficients of the second order array $[T_{ijk}]$ as defined in *MATRIX* ;
- any of the 4×4 coefficients of the beam σ-matrix
- transmission efficiency of an optical channel.
- tunes $\nu_{Y,Z}$ and periodic betatron functions $\beta_{Y,Z}$, $\alpha_{Y,Z}$, $\gamma_{Y,Z}$, as computed in the coupled hypothesis [**?**].

A full list of the constraints available is given in the table page 5.

FIT, *FIT2* are compatible with the use of (*i.e.*, can be encompassed in) *REBELOTE* for successive fitting trials using various sets of parameters (option $K = 22$ in *REBELOTE*).

VARIABLES The first input data in *FIT[2]* is the number of variables *NV*. A variable is defined by a line of data comprised of

$IR =$ number of the varied element in the structure
$IP =$ number of the physical parameter to be varied in this element
$XC =$ coupling parameter. Normally $XC = 0$. If $XC \neq 0$, coupling will occur (see below)

followed by, either

$DV =$ allowed relative range of variation of the physical parameter IP

or

$[Vmin, Vmax] =$ allowed interval of variation of the physical parameter IP

Numbering of the Elements (*IR*) : The elements (*i.e.*, keywords *DIPOLE, QUADRUPO,* etc.) as read by **zgoubi** in the zgoubi.dat sequence are assigned a number. which follows their sequence in the data file. It is that very number, *IR*, that the *FIT[2]* procedure uses. A simple way to get IR once the zgoubi.dat file has been built, is to do a preliminary run, since the first thing **zgoubi** does is copy the sequence from zgoubi.dat into the result file zgoubi.res, with all elements numbered.

Numbering of the Physical Parameters (*IP*) : All the data that follow a keyword are numbered - except for *SCALING*, see below.
With most of the keywords, the numbering follows the principle hereafter :

Input data	**Numbering for FIT**
'KEYWORD'	
first line	1, 2, 3, ..., 9
second line	10, 11, 12, 13, ..., 19
this is a comment	a line of comments is skipped
next line	20, 21, 22, ..., 29
and so on...	

The examples of *QUADRUPO* (quadrupole) and *TOSCA* (Cartesian or cylindrical mesh field map) are as follows.

Input data	**Numbering for FIT**
'QUADRUPO'	
IL	1
XL, R_0, B	10, 11, 12
X_E, λ_E	20, 21
NCE, C_0, C_1, C_2, C_3, C_4, C_5	30, 31, 32, 33, 34, 35, 36
X_S, λ_S	40, 41
NCS, C_0, C_1, C_2, C_3, C_4, C_5	50, 51, 52, 53, 54, 55, 56
XPAS	60
KPOS, XCE, YCE, ALE	70, 71, 72, 73
TOSCA	
IC, IL	1, 2
BNORM, X- [, Y-, Z-]NORM	10, 11 [, 12, 13]
TIT	This is text
IX, IY, IZ, *MOD*	20, 21, 22, 23
FNAME	This is text
ID, A, B, C [A', B', C', etc. if $ID \geq 2$]	30, 31, 32, 33 [34, 35, 36 [, 37, 38, 39] if $ID \geq 2$]
IORDRE	40
XPAS	50
KPOS, XCE, YCE, ALE	60,61,62,63

A different numbering, fully sequential, has been adopted in the following elements :

AIMANT, *DIPOLE*, *EBMULT*, *ELMULT*, *MULTIPOL*.

It is illustrated here after in the case of *MULTIPOL* and *DIPOLE-M*.

Input data	**Numbering for FIT**
'MULTIPOL'	
0	1
365.760 10.0 7.5739 1.4939 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0	2, 3, 4, 5, ..., 13
10.0 4.0 0.80 0.0 0.0 0.0 0.0 0. 0. 0. 0.	14, 15, ..., 24
NC, C_0, C_1, C_2, C_3, C_4, C_5, shift	25, 26, 27, 28, 29, 30, 31, 32
10.0 4.0 0.80 0.0 0.0 0.0 0.0 0. 0. 0. 0.	33, 34, ..., 43
NC, C_0, C_1, C_2, C_3, C_4, C_5, shift	44, 45, 46, 47, ..., 51
0. 0. 0. 0. 0. 0. 0. 0. 0. 0.	52, 53, 54, ..., 61
step size	62
KPOS, XCE, YCE, ALE	63, 64, 65, 66

Input data	Numbering for FIT
'DIPOLE-M'	
NFACE, IC, IL	1, 2, 3
IAMAX, IRMAX	4, 5
B_0, N, B, G	6, 7, 8, 9
AT, ACENT, RM, RMIN, RMAX	10, 11, 12, 13, 14
λ , ξ	15,16
NC, C_0, C_1, C_2, C_3, C_4, C_5, shift	17, 18, 19, 20, 21, 22, 23, 24
ω, θ, R_1, U_1, U_2, R_2	25, 26, 27, 28, 29, 30
etc.	

Parameters in *SCALING* also have a sequential numbering, yet some positions are skipped, this is illustrated in the example hereafter which covers all possible working modes of *SCALING* (all details regarding the numbering can be found in the *FORTRAN* subroutine `rscal.f`) :

Input data	Numbering for FIT	Quantities to be varied (see SCALING for details)
'SCALING'		
$\boxed{1}$ $\boxed{9}$	1 2	Non relevant
AGSMM *AF *BF		Keywords concerned, their labels
-1 3 12 $\boxed{1.}$ 13 $\boxed{1.}$ 14 $\boxed{1.}$	3 4 5	dB1, dB2, dB3 parameters in AGSMM
$\boxed{7.2135}$	6	Field factor
$\boxed{1}$	7	Timing
AGSMM *AD *BD		
-1 3 12 $\boxed{1.}$ 13 $\boxed{1.}$ 14 $\boxed{1.}$	8 9 10	
$\boxed{7.2135}$	11	
$\boxed{1}$	12	
AGSMM *CF		
-1 3 12 $\boxed{1.}$ 13 $\boxed{1.}$ 14 $\boxed{1.}$	13 14 15	
$\boxed{7.2135}$	16	
$\boxed{1}$	17	
AGSQUAD QH_*		
3		
$\boxed{0.605}$ $\boxed{0.77}$ $\boxed{0.879}$	18 19 20	Field factor
$\boxed{1}$ $\boxed{2000}$ $\boxed{10000}$	21 22 23	Timing
AGSQUAD QV_*		
3		
$\boxed{0.587}$ $\boxed{0.83}$ $\boxed{0.83}$	24 25 26	
$\boxed{1}$ $\boxed{2000}$ $\boxed{10000}$	27 28 29	
MULTIPOL COH1		
1.10		No numbering if 1.10 type of option
./Csnk3D/bump_centered.scal		
1 2		
MULTIPOL COH2		
1.10		
./Csnk3D/bump_centered.scal		
1 4		
MULTIPOL KICKH KICKV		
2		
$\boxed{0.1}$ $\boxed{0.3}$	30 31	Field factor
$\boxed{1}$ $\boxed{10}$	32 33	Timing
MULTIPOL		
-1		
$\boxed{0.72135154291}$	34	
$\boxed{1}$	35	

Coupled Variables (XC) Coupling a variable parameter to any other parameter in the structure is possible. This is done by giving XC a value of the form $r.ppp$ where the integer part r is the number of the coupled element in the structure (equivalent to IR, see above), and the decimal part ppp is the number of its parameter of concern (equivalent to IP, see above) (if the parameter number is in the range 1, 2, ... ,9 (resp. 10, 11, ... 19 or 100, ...), then ppp must take the form $00p$ (resp. $0pp$, ppp)). For example, $XC = 20.010$ is a request for coupling with the parameter number 10 of element number 20 of the structure, while $XC = 20.100$ is a request for coupling with the parameter number 100 of element 20.

An element of the structure which is coupled (by means of $XC \neq 0$) to a variable declared in the data list of the *FIT[2]* keyword, needs not appear as one of the *NV* variables in that data list (this would be redundant information).

XC can be either positive or negative. If $XC > 0$, then the coupled parameter will be given the same value as the variable parameter (for example, symmetric quadrupoles in a lens triplet will be given the same field). If $XC < 0$, then the coupled parameter will be given a variation opposite to that of the variable, so that the sum of the two parameters stays constant (for example, an optical element can be shifted while preserving the length of the structure, by coupling together its upstream and downstream drift spaces).

Variation Range There are two ways to define the allowed range for a variable, as follows.

(i) DV : For a variable (parameter number IP under some keyword) with initial value v, the *FIT[2]* procedure is allowed to explore the range $v \times (1 \pm DV)$.

(i) $[v_{min}, v_{max}]$: This specifies the allowed interval of variation.

CONSTRAINTS The next input data in *FIT[2]* is the number of constraints, *NC*. A list of the available constraints is given in the table page 5 ; adding or changing a constraint resorts to the *FORTRAN* file `ff.f`.

Each constraint is defined by the following list of data :

IC	=	type of constraint (see table p. 5).
I, J	=	constraint (*i.e.*, R_{ij}, determinant, tune ; T_{ijk} ; σ_{ij} ; trajectory $\#I$ and coordinate $\#J$)
IR	=	number of the keyword at the exit of which the constraint applies
V	=	desired value of the constraint
W	=	weight of the constraint (smaller W for higher weight)
NP		*NP* values follow

IC=0 : The coefficients σ_{11} (σ_{33}) = horizontal (vertical) beta values and σ_{22} (σ_{44}) = horizontal (vertical) derivatives ($\alpha = -\beta'/2$) are obtained by transport of their initial values at line start as introduced using for instance *OBJET, KOBJ=5.1.*

IC=0.1 : Beam parameters : $\sigma_{11} = \beta_Y, \sigma_{12} = \sigma_{21} = -\alpha_Y, \sigma_{22} = \gamma_Y$, $\sigma_{33} = \beta_Z, \sigma_{34} = \sigma_{43} = -\alpha_Z, \sigma_{44} = \gamma_Z$; periodic dispersion : $\sigma_{16} = D_Y$, $\sigma_{26} = D'_Y$, $\sigma_{36} = D_Z, \sigma_{46} = D'_Z$, all quantities derived by assuming periodic structure and identifying the first order transfer matrix with the form $Icos\mu + Jsin\mu$.

IC=1, 2 : The coefficients R_{ij} and T_{ijk} are calculated following the procedures described in *MATRIX*, option *IFOC* = 0. The fitting of the $[R_{ij}]$ matrix coefficients supposes the tracking of particles with paraxial coordinates, normally defined using *OBJET* option *KOBJ* = 5 or 6.

Type of constraint	Parameters defining the constraints: IC	I	J	Constraint	Parameter(s): #	values			Object definition (recommended)
σ-matrix	0	1 - 6	1 - 6	σ_{IJ} $(\sigma_{11}=\beta_Y, \sigma_{12}=\sigma_{21}=\alpha_Y$, etc.)					*OBJET/KOBJ=5,6*
Periodic parameters	0.N	1 - 6	1 - 6	σ_{IJ} $(\sigma_{11}=\cos\mu_Y+\alpha_Y\sin\mu_Y$, etc.)					*OBJET/KOBJ=5.N*
		7	any	Y-tune = $\mu_Y/2\pi$					
(N=1-9 for *MATRIX*		8	any	Z-tune = $\mu_Z/2\pi$					
block 1-9))		9	any	$\cos(\mu_Y)$					
		10	any	$\cos(\mu_Z)$					
First order transport coeffs.	1	$1-6$	$1-6$	Transport coeff. R_{IJ}					*OBJET/KOBJ=5*
		7	i	$i\neq 8$: YY-determinant ; i=8 : YZ-det.					
		8	j	$j\neq 7$: ZZ-determinant ; j=7 : ZY-det.					
Second order transport coeffs.	2	$1-6$	$11-66$	Transport coeff. $T_{I,j,k}$ $(j=[J/10], k=J-10[J/10])$					*OBJET/KOBJ=6*
Trajectory coordinates	3	$1-IMAX$	$1-7$	$F(J,I)$					*[MC]OBJET*
		-1	$1-7$	$< F(J,i) >_{i=1,IMAX}$					
		-2	$1-7$	$Sup(\lvert F(J,i)\rvert)_{i=1,IMAX}$					
		-3	$1-7$	$Dist\lvert F(J,I)\rvert_{i=I1,I2,dI}$	3	I1	I2	dI	
	3.1	$1-IMAX$	$1-7$	$\lvert F(J,I)-FO(J,I)\rvert$					
	3.2	$1-IMAX$	$1-7$	$\lvert F(J,I)+FO(J,I)\rvert$					
	3.4	$1-IMAX$	$1-7$	$\lvert F(J,I)-F(J,K)\rvert$	1	$K\leq IMAX$			
	3.5	$1-IMAX$	$1-7$	$(F(J,I)-F(J,K))/F(J,K)$	1	$K\leq IMAX$			
Ellipse parameters	4	$1-6$	$1-6$	σ_{IJ} $(\sigma_{11}=\beta_Y$, $\sigma_{12}=\sigma_{21}=\alpha_Y$, etc.)					*OBJET/KOBJ=8 ; MCOBJET/KOBJ=3*
Number of particles	5	-1	any	$N_{survived}/IMAX$					*OBJET*
		$1-3$	any	$N_{in\ \epsilon_{Y,Z,X}}/N_{survived}$	1	ϵ/π			*MCOBJET*
		$4-6$	any	$N_{in\ best\ \epsilon_{Y,Z,X,rms}}/N_{survived}$					*MCOBJET*
Across optical elements,	7.1	$1-IMAX$	$1-7$	min. (1) or max. (2) of $F(J,I)$	1	1-2			*[MC]OBJET*
	7.2	$1-IMAX$	$1-7$	max($F(J,I)$) - min$F(J,I)$)					
	7.3	$1-IMAX$	$1-7$	min$F(J,I)$) + max($F(J,I)$)					
	7.6	$1-IMAX$	$1-7$	min. (1) or max. (2) value of B_J	1	1-2			
	7.7	$1-IMAX$	$1-7$	max(B_J) - min(B_J)					
	7.8	$1-IMAX$	$1-7$	min(B_J) + max(B_J)					
	7.9	$1-IMAX$	$1-7$	$\int B_J\,ds$					
Spin	10	$1-IMAX$	$1-4$	$S_{X,Y,Z}(I)$, $\lvert\vec{S}(I)\rvert$					*[MC]OBJET +SPNTRK*
	10.1	$1-IMAX$	$1-3$	$\lvert S_{X,Y,Z}(I)-SO_{X,Y,Z}(I)\rvert$					

IC=3 : If $1\leq I\leq IMAX$ then the value of coordinate type J ($J=1,6$ for respectively $D,\ Y,\ T,\ Z,\ P,\ S$) of particle number I ($1\leq I\leq IMAX$) is constrained. However I can take special meaning, as follows.

$I=-1$: the constraint is the mean value of coordinate of type J,

$I=-2$: the constraint is the maximum value of coordinate of type J,

$I=-3$: the constraint is the distance between two different particles.

IC=3.1 : Absolute value of the difference between local and initial J-coordinate of particle I (convenient *e.g.* for closed orbit search).

IC=3.2 : Absolute value of the sum of the local and initial J-coordinate of particle I.

IC=3.3 : Minimum (*NP*=1) or maximum value (*NP*=2) of the local J-coordinate of particle I.

IC=3.4 : Absolute value of the difference between local J-coordinates of particles respectively I and K.

IC=4 : The coefficients σ_{11} (σ_{33}) = horizontal (vertical) beta values and σ_{22} (σ_{44}) = horizontal (vertical) derivatives ($\alpha=-\beta'/2$) are derived from an ellipse match of the current particle population (as generated for instance using *MCOBJET, KOBJ=3*).

The fitting of the $[\sigma_{ij}]$ coefficients supposes the tracking of a relevant population of particles within an appropriate emittance.

IC=5 : The constraint value is the ratio of particles (over *IMAX*). Three cases possible :

$I=-1$, ratio of particles still on the run.

$I=1,\ 2,\ 3$, *maximization* of the number of particles encompassed within a given I-type (for respectively $Y,\ Z,\ D$) phase-space emittance value. Then, *NP*=1, followed by the emittance value. The center and shape of the ellipse are determined by a matching to the position and shape of the particle distribution.

$I=4,\ 5,\ 6$, same as previous case, except for the ellipse, taken to be the *rms* matched ellipse to the distribution. Thus *NP*=0.

IC=10 : If $1 \leq I \leq$ *IMAX* then the value of coordinate type J ($J = 1, 3$ for respectively S_X, S_Y, S_Z) of particle number I is constrained.

IC=10.1 : Difference between final and initial J-spin coordinate of particle I (convenient *e.g.* for $\vec{n}_0$ spin vector search).

OBJECT DEFINITION Depending on the type of constraint (see table p. 5), constraint calculations are performed either from transport coefficient calculation and in such case require *OBJET* with either *KOBJ* = 5 or *KOBJ* = 6, or from particle distributions and in this case need object definition using for instance *OBJET* with *KOBJ* = 8, *MCOBJET* with *KOBJ* = 3.

THE FITTING METHODS The *FIT* procedure was drawn from the matrix transport code BETA [**?**]. It is a direct sequential minimization of the quadratic sum of all errors (*i.e.*, differences between desired and actual values for the *NC* constraints), each normalized by its specified weight W (the smaller W, the stronger the constraint).
The step sizes for the variation of the physical parameters depend on their initial values, and cannot be accessed by the user. At each iteration, the optimum value of the step size, as well as the optimum direction of variation, is determined for each one of the *NV* variables. Then follows an iterative global variation of all *NV* variables, until the minimization fails which results in a next iteration on the optimization of the step sizes.

The *FIT2* procedure is based on the Nelder-Mead method, it has various specificities, details can be found in Ref. [**?**].

The optimization process may be stopped by means of a penalty value, or a maximum number of iterations on the step size or on the call to the function.

COMBINING FIT[2] AND REBELOTE *FIT[2]* may be followed by the keyword *REBELOTE*. This allows executing again the same fit procedure, after having changed the value of some parameter in zgoubi.dat data list. That's the role of *REBELOTE* in that game : it changes that parameter, and causes the fit to be executed again with that different value.

FIT, FIT2	**Fitting procedure**		
NV	Number of physical parameters to be varied	≤ 20	I
For I = 1, NV	*repeat NV times the following sequence*		
either :			
IR, *IP*, *XC*, *DV*	Number of the element in the structure ; number of the physical parameter in the element ; coupling switch (off = 0) ; variation range ($\pm$)	$\leq$MXL [1], $\leq$MXD, $\pm$ MXD.MXD [2], relative	2*I, 2*E
or :			
IR, *IP*, *XC*, $[V_{min}, V_{max}]$		$\leq$MXL, $\leq$MXD,	2*I, 3*E
NC [, *penalty* [,*ITER*]] [3]	Number of constraints [, penalty [, number of iterations]].	≤ 20 [,10^{-n} [,> 0]]	I [, E [, I]]
For I = 1, NC	*repeat NC times the following sequence :*		
IC, I, J, IR, V [4], WV, NP [, $p_i(i=1,NP)$]	IC, I and J define the type of constraint (see table below) ; IR : number of the element after which the constraint applies ; V : value ; W : weight (the stronger the lower WV) NP : number of parameters ; if $NP \geq 1$, $p_i(i=1,NP)$: parameter values.	0-5, 3*($>$0), current unit, 2*no dim., curr. units	4*I, 2*E, I, NP*E

[1] MXL value is set in include file `MXLD.H`.
[2] MXD value is set in include file `MXLD.H`. Data is of the form "integer.iii" with i a 1-digit integer.
[3] FIT[2] will stop when the sum of the squared residuals gets $<$*penalty*, or when the maximum allowed number of iterations is reached.
[4] V is in current **zgoubi** units in the case of particle coordinates (cm, mrad). It is in MKSA units (m, rad) in the case of matrix coefficients.

FOCALE	**Particle coordinates and horizontal beam size at distance XL**		
XL	Distance from the location of the keyword	cm	E

FOCALEZ	**Particle coordinates and vertical beam size at distance *XL***		
XL	Distance from the location of the keyword	cm	E

GASCAT : Gas scattering

Modification of particle momentum and velocity vector, performed at each integration step, under the effect of scattering by residual gas.

Installation is to be completed.

GASCAT **Gas scattering**

KGA	Off/On switch	0, 1	I
AI, DEN	Atomic number ; density		2*E

GETFITVAL : Get values of *variables* as saved from former FIT[2] run

This keyword allows reading, from a file whose name needs be specified, parameter values to be assigned to optical elements in zgoubi.dat.

That file is expected to contain a copy-paste of the data under the *FIT[2]* procedure as displayed in zgoubi.res, normally under the form

```
STATUS OF VARIABLES  (Iteration #    95)
LMNT  VAR PARAM  MINIMUM    INITIAL    FINAL      MAXIMUM     STEP    NAME       LBL1 LBL2
145    1   4    -3.000E+03    762.   761.9484791 3.000E+03  1.254E-05 MULTIPOL   HKIC DHCB02
182    2   4    -1.000E+03   -231.  -230.9846875 1.000E+03  4.182E-06 MULTIPOL   HKIC DHCB08
146    3   4    -1.000E+03   -320.  -319.8554171 1.000E+03  4.182E-06 MULTIPOL   VKIC DVCB02
183    4   4    -1.000E+03    528.   527.7249064 1.000E+03  4.182E-06 MULTIPOL   VKIC DVCB08
615    5   4    -3.000E+03    308.   307.6860565 3.000E+03  1.254E-05 MULTIPOL   HKIC DHCF02
651    6   4    -1.000E+03   -114.  -113.8490362 1.000E+03  4.182E-06 MULTIPOL   HKIC DHCF08
616    7   4    -1.000E+03   -78.9  -78.88730937 1.000E+03  4.182E-06 MULTIPOL   VKIC DVCF02
652    8   4    -1.000E+03    212.   211.8789183 1.000E+03  4.182E-06 MULTIPOL   VKIC DVCF08
# STATUS OF CONSTRAINTS
# TYPE  I   J  LMNT#     DESIRED       WEIGHT         REACHED         KI2     *  Parameter(s)
#   3   1   2   127   0.0000000E+00  1.0000E+00    1.0068088E-08   6.0335E-01 *  0 :
#   3   1   3   127   0.0000000E+00  1.0000E+00    7.0101405E-09   2.9250E-01 *  0 :
#   3   1   4   127   0.0000000E+00  1.0000E+00    2.9184383E-10   5.0696E-04 *  0 :
#   3   1   5   127   0.0000000E+00  1.0000E+00    3.1142381E-10   5.7727E-04 *  0 :
#   3   1   2   436   0.0000000E+00  1.0000E+00    3.8438378E-09   8.7944E-02 *  0 :
#   3   1   3   436   0.0000000E+00  1.0000E+00    1.5773011E-09   1.4808E-02 *  0 :
#   3   1   4   436   0.0000000E+00  1.0000E+00    2.2081272E-10   2.9022E-04 *  0 :
#   3   1   5   436   0.0000000E+00  1.0000E+00    5.7930552E-11   1.9975E-05 *  0 :
#   Function  called   1859 times
# Xi2 =    1.68006E-16   Busy...
```

A '#' at the beginning of a line means it is commented, thus it will not be taken into account. However a copy-paste from zgoubi.res (which is the case in the present example) would not not need any commenting.

Since some of the *FIT[2]* variables may belong in *[MC]OBJET*, *GETFITVAL* may appear right before *[MC]OBJET* in zgoubi.dat, to allow for its updating.

GETFITVAL	**Get values of *variables* as saved from former FIT[2] run**	
FNAME	Name of storage file. Zgoubi will proceed silently if not found.	A

HISTO : 1-D histogram

Any of the coordinates used in **zgoubi** may be histogrammed, namely initial Y_0, T_0, Z_0, P_0, S_0, D_0 or current Y, T, Z, P, S, D particle coordinates ($S =$ path length ; D may change in decay process simulation with *MCDESINT*, or when ray-tracing in $\vec{E}$ fields), and also spin coordinates and modulus S_X, S_Y, S_Z and $\left\|\vec{S}\right\|$.

HISTO can be used in conjunction with *MCDESINT*, for statistics on the decay process, by means of *TYP*. *TYP* is a one-character string. If it is set equal to 'S', only secondary particles (they are tagged with an 'S') will be histogrammed. If it is set equal to 'P', then only parent particles (non-'S') will be histogrammed. For no discrimination between S-econdary and P-arent particles, *TYP* = 'Q' must be used.

The dimensions of the histogram (number of lines and columns) may be modified. It can be normalized with *NORM* = 1, to avoid saturation.

Histograms are indexed with the parameter *NH*. This allows making independent histograms of the same coordinate at several locations in a structure. This is also useful when piling up problems in a single input data file (see also *RESET*). *NH* is in the range 1-5.

If *REBELOTE* is used, the statistics on the 1+*NPASS* runs in the structure will add up.

HISTO **1-D histogram**

J, $X_{\min}$, $X_{\max}$, *NBK*, *NH*	J = type of coordinate to be histogrammed ; the following are available : • current coordinates : $1(D)$, $2(Y)$, $3(T)$, $4(Z)$, $5(P)$, $6(S)$, • initial coordinates : $11(D_0)$, $12(Y_0)$, $13(T_0)$, $14(Z_0)$, $15(P_0)$, $16(S_0)$, • spin : $21(S_X)$, $22(S_Y)$, $23(S_Z)$, $24(< S >)$; $X_{\min}$, $X_{\max}$ = limits of the histogram, in units of the coordinate of concern ; *NBK* = number of channels ; *NH* = number of the histogram (for independence of histograms of the same coordinate)	1-24, 2* current units, < 120, 1-5	I, 2*E, 2*I
NBL, KAR, NORM, TYP	Number of lines (= vertical amplitude) ; alphanumeric character ; normalization if *NORM* = 1, otherwise *NORM* = 0 ; *TYP* = ‘P’ : primary particles are histogrammed, or ‘S’ : secondary, or Q : all particles - for use with *MCDESINT*	normally 10-40, char., 1-2, P-S-Q	I, A1, I, A1

IMAGE **Localization and size of horizontal waist**

IMAGES **Localization and size of horizontal waists**

For each momentum group, as classified by means of *OBJET*, *KOBJ* = 1, 2 or 4

IMAGESZ **Localization and size of vertical waists**

For each momentum group, as classified by means of *OBJET*, *KOBJ* = 1, 2 or 4

IMAGEZ **Localization and size of vertical waist**

MAP2D : 2-D Cartesian uniform mesh field map - arbitrary magnetic field [?]

MAP2D reads a 2-D field map that provides the three components B_X, B_Y, B_Z of the magnetic field at all nodes of a 2-D Cartesian uniform mesh in an (X, Y) plane. No particular symmetry is assumed, which allows the treatment of any type of field (*e.g.*, solenoidal, or dipole, helical dipole, at arbitrary Z elevation - the map needs not be a mid-plane map).

The field map data file has to be be filled with a format that satisfies the *FORTRAN* reading sequence below (in principle compatible with *TOSCA* code outputs), details and possible updates are to be found in the source file `'fmapw.f'` :

```
OPEN (UNIT = NL, FILE = FNAME, STATUS = 'OLD' [,FORM='UNFORMATTED'])
DO 1 J=1,JY
DO 1 I=1,IX
IF (BINARY) THEN
READ(NL) Y(J), Z, X(I), BY(I,J), BZ(I,J), BX(I,J)
ELSE
READ(NL,100) Y(J), Z, X(I), BY(I,J), BZ(I,J), BX(I,J)
100             FORMAT (1X, 6E11.4)
ENDIF
1         CONTINUE
```

IX (JY) is the number of longitudinal (transverse horizontal) nodes of the 2-D uniform mesh, Z is the considered Z-elevation of the map. For binary files, FNAME must begin with 'B_' or 'b_', a flag 'BINARY' will thus be set to '.TRUE.'. The field $\vec{B} = (B_X, B_Y, B_Z)$ is next normalized with BNORM, prior to ray-tracing. As well the coordinates X, Y are normalized with *X-*, *Y-NORM* coefficients (useful to convert to centimeters, the working units in **zgoubi**).

At each step of the trajectory of a particle, the field and its derivatives are calculated using a second or fourth degree polynomial interpolation followed by a Z extrapolation (see sections **??** page **??**, **??** page **??**). The interpolation grid is 3*3-node for 2nd order (option *IORDRE = 2*) or 5*5 for 4th order (option *IORDRE = 4*).

Entrance and/or exit integration boundaries may be defined, in the same way as for *CARTEMES*.

MAP2D **2-D Cartesian uniform mesh field map - arbitrary magnetic field**

IC, IL	$IC = 1, 2$: print the field map $IL = 1, 2[\times 10^n]$: print field and coordinates along trajectories	0-2, 0-2$[\times 10^n]$	2*I
BNORM, XN,YN	Field and X-,Y-coordinate normalization coeffs.	3*no dim.	3*E
TITL	Title. Start with “FLIP” to get field map X-flipped.		A80
IX, JY	Number of longitudinal and horizontal-transverse nodes of the mesh (the Z elevation is arbitrary)	≤ 400, ≤ 200	2*I
FNAME [1]	File name		A80
ID, A, B, C [, A', B', C', B'', etc., if $ID \geq 2$]	Integration boundary. Ineffective when $ID = 0$. ID = -1, 1 or ≥ 2 : as for *CARTEMES*	≥ -1, 2*no dim., cm [,2*no dim., cm, etc.]	I,3*E [,3*E,etc.]
IORDRE	Degree of polynomial interpolation, 2nd or 4th order.	2, 4	I
XPAS	Integration step	cm	E
KPOS, XCE, *YCE, ALE*	*KPOS*=1 : element aligned, 2 : misaligned ; shifts, tilt (unused if *KPOS*=1)	1-2, 2*cm, rad	I, 3*E

[1] *FNAME* (*e.g.*, magnet.map) contains the field map data.
These must be formatted according to the following *FORTRAN* read sequence (normally compatible with *TOSCA* code *OUTPUTS* - details and possible updates are to be found in the source file `'fmapw.f'`) :

```
OPEN (UNIT = NL, FILE = FNAME, STATUS = 'OLD')
DO 1 J = 1, JY
DO 1 I = 1, IX
IF (BINARY) THEN
READ(NL) Y(J), Z(1), X(I), BY(I,J), BZ(I,J), BX(I,J)
ELSE
READ(NL,100) Y(J), Z(1), X(I), BY(I,J), BZ(I,J), BX(I,J)
100       FORMAT (1X, 6E11.4)
ENDIF
1     CONTINUE
```

where $X(I)$, $Y(J)$ are the longitudinal, horizontal coordinates in the at nodes (I, J) of the mesh, $Z(1)$ is the vertical elevation of the map, and BX, BY, BZ are the components of the field.

For binary files, *FNAME* must begin with 'B_' or 'b_'˜; a logical flag 'Binary' will then automatically be set to '.TRUE.'

MAP2D-E : 2-D Cartesian uniform mesh field map - arbitrary electric field

MAP2D-E reads a 2-D field map that provides the three components E_X, E_Y, E_Z of the electric field at all nodes of a 2-D Cartesian uniform mesh in an (X, Y) plane. No particular symmetry is assumed, which allows the treatment of any type of field (*e.g.*, field of a parallel-plate mirror with arbitrary Z elevation - the map needs not be a mid-plane map).

The field map data file has to be be filled with a format that satisfies the *FORTRAN* reading sequence below (in principle compatible with *TOSCA* code outputs), details and possible updates are to be found in the source file `'fmapw.f'` :

```
OPEN (UNIT = NL, FILE = FNAME, STATUS = 'OLD' [,FORM='UNFORMATTED'])
DO 1 J=1,JY
DO 1 I=1,IX
IF (BINARY) THEN
READ(NL) Y(J), Z, X(I), EY(I,J), EZ(I,J), EX(I,J)
ELSE
READ(NL,100) Y(J), Z, X(I), EY(I,J), EZ(I,J), EX(I,J)
100             FORMAT (1X, 6E11.4)
ENDIF
1        CONTINUE
```

IX (JY) is the number of longitudinal (transverse horizontal) nodes of the 2-D uniform mesh, Z is the considered Z-elevation of the map. For binary files, FNAME must begin with 'E_' or 'b_', a flag 'BINARY' will thus be set to '.TRUE.'. The field $\vec{E} = (E_X, E_Y, E_Z)$ is next normalized with ENORM, prior to ray-tracing. As well the coordinates X, Y are normalized with *X-,Y-NORM* coefficients (useful to convert to centimeters, the working units in **zgoubi**.

At each step of the trajectory of a particle, the field and its derivatives are calculated using a second or fourth degree polynomial interpolation followed by a Z extrapolation (see sections **??** page **??**, **??** page **??**). The interpolation grid is 3*3-node for 2nd order (option *IORDRE = 2*) or 5*5 for 4th order (option *IORDRE = 4*).

Entrance and/or exit integration boundaries may be defined, in the same way as for *CARTEMES*.

MAP2D-E **2-D Cartesian uniform mesh field map - arbitrary electric field**

IC, *IL*	$IC = 1, 2$: print the field map $IL = 1, 2[\times 10^n]$: print field and coordinates along trajectories	0-2, 0-2$[\times 10^n]$	2*I
ENORM, X-,Y-NORM	Field and X-,Y-coordinate normalization coeffs.	2*no dim.	2*E
TITL	Title. Start with "FLIP" to get field map X-flipped.		A80
IX, JY	Number of longitudinal and horizontal-transverse nodes of the mesh (the Z elevation is arbitrary)	≤ 400, ≤ 200	2*I
FNAME [1]	File name		A80
ID, A, B, C [, A', B', C', B'', etc., if $ID \geq 2$]	Integration boundary. Ineffective when $ID = 0$. ID = -1, 1 or ≥ 2 : as for *CARTEMES*	≥ -1, 2*no dim., cm [,2*no dim., cm, etc.]	I,3*E [,3*E,etc.]
IORDRE	Degree of polynomial interpolation, 2nd or 4th order.	2, 4	I
XPAS	Integration step	cm	E
KPOS, *XCE*, *YCE, ALE*	*KPOS*=1 : element aligned, 2 : misaligned ; shifts, tilt (unused if *KPOS*=1)	1-2, 2*cm, rad	I, 3*E

[1] *FNAME* (*e.g.*, ''mirror.map'') contains the field map data.
These must be formatted according to the following *FORTRAN* read sequence - details and possible updates are to be found in the source file `'fmapw.f'` :

```
OPEN (UNIT = NL, FILE = FNAME, STATUS = 'OLD')
DO 1 J = 1, JY
DO 1 I = 1, IX
IF (BINARY) THEN
READ(NL) Y(J), Z(1), X(I), EY(I,J), EZ(I,J), EX(I,J)
ELSE
READ(NL,100) Y(J), Z(1), X(I), EY(I,J), EZ(I,J), EX(I,J)
100        FORMAT (1X, 6E11.4)
ENDIF
1     CONTINUE
```

where $X(I)$, $Y(J)$ are the longitudinal, horizontal coordinates in the at nodes (I, J) of the mesh, $Z(1)$ is the vertical elevation of the map, and EX, EY, EZ are the components of the field.

For binary files, *FNAME* must begin with 'B_' or 'b_'˜; a logical flag 'Binary' will then automatically be set to '.TRUE.'

MARKER : Marker

MARKER does nothing. Just a marker. No data.

As any other keyword, *MARKER* is allowed two *LABEL*s. Using '.plt' as a second *LABEL* will cause storage of current coordinates into zgoubi.plt.

MARKER **Marker**

Just a marker. No data
'.plt' as a second *LABEL* will cause storage of current coordinates into zgoubi.plt

MATRIX : Calculation of transfer coefficients, periodic parameters

MATRIX causes the calculation of the transfer coefficients through the optical structure, from the *OBJET* down to the location where *MATRIX* is introduced in the structure, or, upon option, down to the horizontal focus closest to that location. In this last case the position of the focus is calculated automatically in the same way as the position of the waist in *IMAGE*. Depending on option *IFOC*, *MATRIX* also delivers the beam matrix and betatron phase advances or (case of a periodic structure) periodic beam matrix and tunes, chromaticities and other global parameters.

Depending on the value of option *IORD*, different procedures follow

- If *IORD* = 0, *MATRIX* is inhibited (equivalent to *FAISCEAU*, whatever *IFOC*).
- If *IORD* = 1, the first order transfer matrix $[R_{ij}]$ is calculated, from a third order approximation of the coordinates. For instance

  $$Y^+ = \left(\frac{Y}{T_0}\right) T_0 + \left(\frac{Y}{T_0^2}\right) T_0^2 + \left(\frac{Y}{T_0^3}\right) T_0^3, \qquad Y^- = -\left(\frac{Y}{T_0}\right) T_0 + \left(\frac{Y}{T_0^2}\right) T_0^2 - \left(\frac{Y}{T_0^3}\right) T_0^3$$

  will yield, neglecting third order terms,

  $$R_{11} = \left(\frac{Y}{T_0}\right) = \frac{Y^+ - Y^-}{2T_0}$$

  In addition, if *OBJET*, *KOBJ* = 5.01 is used (hence introducing initial optical function values, $\alpha_{Y,Z}$, $\alpha_{Y,Z}$, $D_{Y,Z}$, $D'_{Y,Z}$), then, using the R_{ij} above, *MATRIX* will transport the optical functions and phase advances ϕ_Y, ϕ_Z, following

  $$\begin{pmatrix} \beta \\ \alpha \\ \gamma \end{pmatrix}_{at\ MATRIX} = \begin{pmatrix} R_{11}^2 & -2R_{11}R_{12} & R_{12}^2 \\ -R_{11}R_{21} & R_{12}R_{21} & R_{11}R_{12} \\ R_{21}^2 & -2R_{21}R_{22} & R_{22}^2 \end{pmatrix} \begin{pmatrix} \beta \\ \alpha \\ \gamma \end{pmatrix}_{at\ OBJET}$$

  $$\Delta\phi_Y = A\tan \frac{R_{12}}{(R_{11}\beta_{Y,objet} - R_{12}\alpha_{Y,objet})}, \quad \Delta\phi_Z = A\tan \frac{R_{34}}{(R_{33}\beta_{Z,objet} - R_{34}\alpha_{Z,objet})}, \tag{1}$$

  $$\phi_{Y,Z} \to \phi_{Y,Z} + 2\pi \quad \text{if } \phi_{Y,Z} < 0, \text{given } [0,\pi] \text{ Atan determination}$$

  and print these out.
- If *IORD* = 2, fifth order Taylor expansions are used for the calculation of the first order transfer matrix $[R_{ij}]$ and of the second order matrix $[T_{ijk}]$. Other higher order coefficients are also calculated.

An automatic generation of an appropriate object for the use of *MATRIX* can be obtained using the procedure *OBJET* (pages **??**, **??**), as follows
- if *IORD* = 1, use *OBJET*(KOBJ = 5[.NN, NN=01,99]), that generates up to 99*11 sets of initial coordinates. In this case, up to ninety nine matrices may be calculated, each one *wrt.* to the reference trajectory of concern.
- if *IORD* = 2, use *OBJET*(KOBJ = 6) that generates 61 sets of initial coordinates.

The next option, *IFOC*, acts as follows

- If *IFOC* = 0, the transfer coefficients are calculated at the location of *MATRIX*, and with respect to the reference trajectory. For instance, Y^+ and T^+ above are defined for particle number i as $Y^+ = Y^+(i) - Y(Ref)$, and $T^+ = T^+(i) - T(\text{ref.})$.
- If *IFOC* = 1, the transfer coefficients are calculated at the horizontal focus closest to *MATRIX* (determined automatically), while the reference direction is that of the reference particle. For instance, Y^+ is defined for particle number i as $Y^+ = Y^+(i) - Y_{\text{focus}}$, while T^+ is defined as $T^+ = T^+(i) - T(\text{ref.})$).
- If *IFOC* = 2, no change of reference frame is performed : the coordinates refer to the current frame. Namely, $Y^+ = Y^+(i)$, $T^+ = T^+(i)$, etc.

- If *IFOC* = 10 + *NPeriod*, then, from the 1-turn transport matrix as obtained in the way described above, *MATRIX* calculates periodic parameters characteristic of the structure such as optical functions and tune numbers, assuming that it is *NPeriod*-periodic, and in the coupled hypothesis, based on the Edwards-Teng method [**?**].

  If *IORD* = 2 additional periodic parameters are computed such as chromaticities, beta-function momentum dependence, etc.

Addition of *zgoubi.MATRIX.out* next to *IORD*, *IFOC* will cause stacking of *MATRIX* output data into zgoubi.MATRIX.out file (convenient for use with *e.g.* gnuplot type of data treatment software).

MATRIX **Calculation of transfer coefficients, periodic parameters**

IORD, *IFOC* *[, zgoubi.MATRIX.out]*	Options : *IORD* = 0 : Same effect as *FAISCEAU* *IORD* = 1 (normally using *OBJET*, *KOBJ* = 5) : First order transfer matrix ; beam matrix, phase advance if using *OBJET*, *KOBJ* = 5.01 ; if *IFOC* > 10 : periodic beam matrix, tune numbers *IORD* = 2 (normally using *OBJET*, *KOBJ* = 6) : First order transfer matrix $[R_{ij}]$, second order array $[T_{ijk}]$ and higher order transfer coefficients ; if *IFOC* > 10 : periodic parameters, *IFOC* = 0 : matrix at actual location, reference $\equiv$ particle # 1 *IFOC* = 1 : matrix at the closest first order horizontal focus, reference $\equiv$ particle # 1 *IFOC* = 10 + NPER : same as *IFOC* = 0, and also calculates the Twiss parameters, tune numbers, etc. (assuming that the DATA file describes one period of a NPER-period structure). Including 'zgoubi.MATRIX.out' will cause printout to zgoubi.MATRIX.out file	0-2, 0-1 or > 10	2*I [,A]

MCDESINT : Monte-Carlo simulation of in-flight decay[?]

As soon as *MCDESINT* appears in a structure (normally, after *OBJET* or after *CIBLE*), in-flight decay simulation starts. It must be preceded by *PARTICUL* for the definition of mass M_1 and *COM* lifetime τ_1.
The two-body decay simulated is

$$1 \longrightarrow 2 + 3$$

The decay is isotropic in the center of mass. 1 is the incoming particle, with mass M_1, momentum $p_1 = \gamma_1 M_1 \beta_1 c$ (relative momentum $D_1 = \dfrac{p_1}{q}\dfrac{1}{\mathit{BORO}}$ with *BORO* = reference rigidity, defined in *[MC]OBJET*), and position Y_1, Z_1 in the **zgoubi** frame. 2 and 3 are decay products with respective masses and momenta M_2, M_3 and $p_2 = \gamma_2 M_2 \beta_2 c$, $p_3 = \gamma_3 M_3 \beta_3 c$.
The decay length s_1 of particle 1 is related to its center of mass lifetime τ_1 by

$$s_1 = c\tau_1 \sqrt{\gamma_1^2 - 1}$$

The path length s up to the decay point is then calculated from a random number $0 < R_1 \leq 1$ by using the exponential decay formula

$$s = -s_1 \ell n R_1$$

After decay, particle 2 will be ray-traced with assumed positive charge, while particle 3 is discarded. Its scattering angles in the center of mass θ^* and ϕ are generated from two other random numbers R_2 and R_3.
ϕ is a relativistic invariant, and θ in the laboratory frame (Fig. 1) is given by

$$\tan\theta = \frac{1}{\gamma_1} \frac{\sin\theta^*}{\dfrac{\beta_1}{\beta_2^*} + \cos\theta^*}$$

β_2^* and momentum p_2 are given by

$$\gamma_2^* = \frac{M_1^2 + M_2^2 - M_3^2}{2M_1M_2}$$
$$\beta_2^* = \left(1 - \frac{1}{\gamma^2}\right)^{1/2}$$
$$\gamma_2 = \gamma_1 \gamma_2^* \left(1 + \beta_1 \beta_2^* \cos\theta^*\right)$$
$$p_2 = M_2 \sqrt{\gamma_2^2 - 1}$$

Finally, θ and ϕ are transformed into the angles T_2 and P_2 in the **zgoubi** frame, and the relative momentum takes the value $D_2 = \dfrac{p_2}{q}\dfrac{1}{\mathit{BORO}}$ (where *BORO* is the reference rigidity, see *OBJET*), while the starting position of M_2 is (Y_1, Z_1, s_1).

The decay simulation by **zgoubi** satisfies the following procedures. In optical elements and field maps, after each integration step *XPAS*, the actual path length of the particle, $F(6, I)$, is compared to its limit path length s. If s is passed, then the particle is considered as having decayed at $F(6, I) - \dfrac{\mathit{XPAS}}{2}$, at a position obtained by a linear translation from the position at $F(6, I)$. Presumably, the smaller *XPAS*, the smaller the error on position and angles at the decay point.

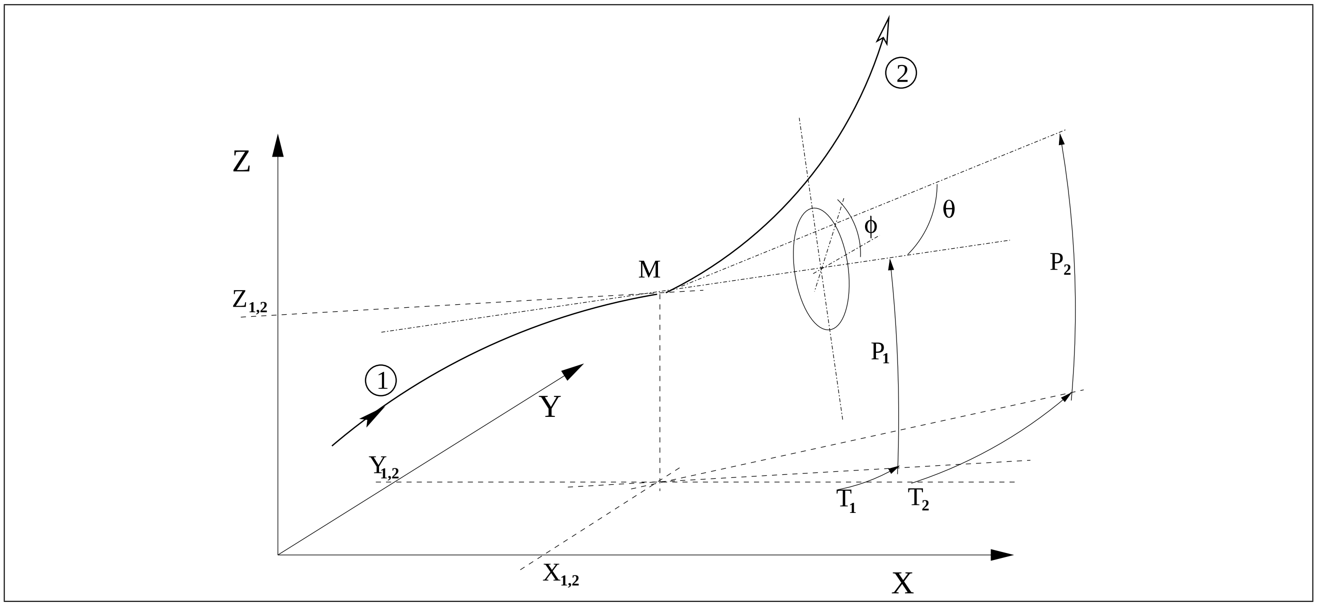


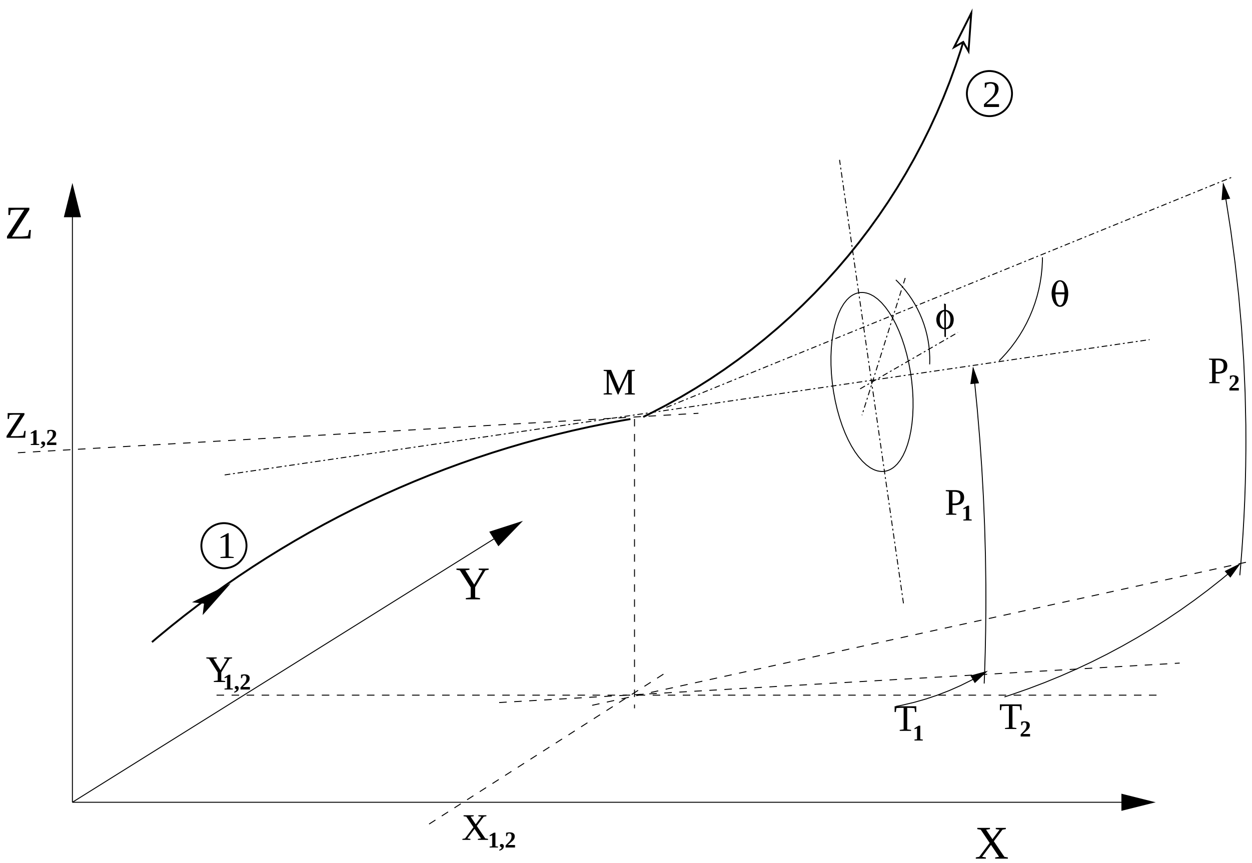
Figure 1: At position $M(X_1, Y_1, Z_1)$, particle 1 decays into 2 and 3 ; **zgoubi** then proceeds with the computation of the trajectory of 2, while 3 is discarded.
θ and ϕ are the scattering angles of particle 2 relative to the direction of the incoming particle 1 ; they transform to T_2 and P_2 in **zgoubi** frame.

In *ESL* and *CHANGREF*, $F(6, I)$ is compared to s at the end of the element. If the decay occurs inside the element, the particle is considered as having decayed at its actual limit path length s, thus its coordinates at s are recalculated by translation.

The limit path length of all particles ($I = 1$, *IMAX*) is stored in the array *FDES*$(6, I)$. For further statistical purposes (*e.g.*, use of *HISTO*) the daughter particle 2 is tagged with an $'S'$ standing for "secondary". When a particle decays, its coordinates D, Y, T, Z, P, s, *time* at the decay point are stored in *FDES*(J, I), $J = 1$, 7.

A note on negative drifts :
The use of negative drifts with *MCDESINT* is allowed and correct. For instance, negative drifts may occur in a structure for some of the particles when using *CHANGREF* (due to the Z-axis rotation or a negative *XCE)*, or when using *DRIFT* with *XL* < 0. Provision has been made to take it into account during the *MCDESINT* procedure, as follows.
If, due to a negative drift, a secondary particle reaches back the decay location of its parent particle, then the parent particle is "resurrected " with its original coordinates at that location, the secondary particle is discarded, and ray-tracing resumes in a regular way for the parent particle which is again allowed to decay, after the same path length. This procedure is made possible by prior storage of the coordinates of the parent particles (in array *FDES*(J, I)) each time a decay occurs.
Negative steps (*XPAS*< 0) in optical elements are not compatible with *MCDESINT*.

MCDESINT [1]	**Monte-Carlo simulation of in-flight decay** M1 $\rightarrow$ M2 + M3		
[INFO,] [2] $M2$, $M3$, $\tau 2$ [3]	[Switch,] ; masses of the two decay products; COM lifetime of particle 2	[-,] 2*MeV/c^2, s	[A4,] 3*E
$I1$, $I2$, $I3$	Seeds for random number generators	3*$\simeq 10^6$	3*I



Particle 1 decays into 2 and 3 ; **zgoubi** then calculates trajectory of 2, while 3 is discarded. θ and ϕ are the scattering angles of particle 2 relative to the direction of the incoming particle 1. They transform to T_2 and P_2 in Zgoubi frame.

---

[1] *MCDESINT* must be preceded by *PARTICUL*, for the definition of the mass and lifetime of the incoming particle M1.
[2] Presence of 'INFO' will cause more info on decay kinematics parameters to be printed into zgoubi.res at each decay.
[3] $\tau 2$ can be left blank, in which case the lifetime of particle 2 is set to zero (it decays immediately, which from a practical point of view means that it is not tracked).

MCOBJET : Monte-Carlo generation of a 6-D object

MCOBJET generates a set of *IMAX* random 6-D initial conditions (the maximum value for *IMAX* is defined in the include file `'MAXTRA.H'`). It can be used in conjunction with the keyword *REBELOTE* which either allows generating an arbitrarily high number of initial conditions, or, in the hypothesis of a periodic structure, allows multi-turn tracking with initial conditions at pass number *IPASS* identified with conditions at end of pass number *IPASS* -1.

The first datum in *MCOBJET* is the reference rigidity (a negative value is allowed)

$$BORO = \frac{p_0}{q} \text{ (kG.cm)}$$

Depending on the value of the next datum, *KOBJ*, the *IMAX* particles have their initial random conditions Y, T, Z, P, X and D (relative rigidity, $B\rho/BORO$) generated on 3 different types of supports, as described below.
Next come the data

$$KY, KT, KZ, KP, KX, KD$$

that specify the type of probability density for the 6 coordinates.
KY, *KT*, *KZ*, *KP*, *KX* can take the following values :

1. uniform density, $p(x) = 1/2\delta x$ if $-\delta x \le x \le \delta x$, $p(x) = 0$ elsewhere,
2. Gaussian density, $p(x) = \dfrac{1}{\delta x\sqrt{2\pi}} e^{-\dfrac{x^2}{2\delta x^2}}$,
3. parabolic density, $p(x) = \dfrac{3}{4\delta x}(1 - \dfrac{x^2}{\delta x^2})$ if $-\delta x \le x \le \delta x$, $p(x) = 0$ elsewhere.

KD can take the following values :

1. uniform density, $p(D) = 1/2\delta D$ if $-\delta D \le D \le \delta D$, $p(D) = 0$ elsewhere,
2. exponential density, $p(D) = N_0 \exp(C_0 + C_1 l + C_2 l^2 + C_3 l^3)$ with $0 \le l \le 1$ and $-\delta D \le D \le \delta D$,
3. $p(D)$ is determined by a kinematic relation, namely, with $T =$ horizontal angle, $D = \delta D * T$.

Next come the central values for the random sorting,

$$Y_0, \quad T_0, \quad Z_0, \quad P_0, \quad X_0, \quad D_0$$

namely, the probability density laws $p(x)$ ($x = Y, T, Z, P$ or X) and $p(D)$ described above apply to the variables $x - x_0$ ($\equiv Y - Y_0, T - T_0$, ...) and $D - D_0$ respectively. Negative value for D_0 is allowed (see section **??**, page **??**).

KOBJ = 1 : Random generation of *IMAX* particles in a hyper-window with widths (namely the half-extent for uniform or parabolic distributions ($KY, KT, ... = 1$ or 3), and the r.m.s. width for Gaussian distributions ($KY, KT, ... = 2$))

$$\delta Y, \quad \delta T, \quad \delta Z, \quad \delta P, \quad \delta X, \quad \delta D$$

Then follow the cut-off values, in units of the r.m.s. widths δY, δT, ... (used only for Gaussian distributions, $KY, KT, ... = 2$)

$$N_{\delta Y}, \quad N_{\delta T}, \quad N_{\delta Z}, \quad N_{\delta P}, \quad N_{\delta X}, \quad N_{\delta D}$$

The last data are the parameters

$$N_0, \quad C_0, \quad C_1, \quad C_2, \quad C_3$$

needed for generation of the D coordinate upon option $KD = 2$ (unused if $KD = 1,\ 3$) and a set of three integer seeds for initialization of random sequences,

$$IR1, \quad IR2, \quad IR3 \quad \text{(all} \simeq 10^6\text{)}$$

All particles generated by *MCOBJET* are tagged with a (non-S) character, for further statistic purposes (*e.g.*, with *HISTO*, *MCDESINT*).

KOBJ = 2 : Random generation of *IMAX* $= IY * IT * IZ * IP * IX * ID$ particles on a hyper-grid. The input data are the number of bars in each coordinate

$$IY, \quad IT, \quad IZ, \quad IP, \quad IX, \quad ID$$

the spacing of the bars

$$PY, \quad PT, \quad PZ, \quad PP, \quad PX, \quad PD$$

the width of each bar

$$\delta Y, \quad \delta T, \quad \delta Z, \quad \delta P, \quad \delta X, \quad \delta D$$

the cut-offs, used with Gaussian densities (in units of the r.m.s. widths)

$$N_{\delta Y}, \quad N_{\delta T}, \quad N_{\delta Z}, \quad N_{\delta P}, \quad N_{\delta X}, \quad N_{\delta D}$$

This is illustrated in Fig. 1.

The last two sets of data in this option are the parameters

$$N_0, \quad C_0, \quad C_1, \quad C_2, \quad C_3$$

needed for generation of the D coordinate upon option KD= 2 (unused if KD= 1, 3) and a set of three integer seeds for initialization of random sequences, $IR1$, $IR2$, and $IR3$ (all $\simeq 10^6$).
All particles generated by *MCOBJET* are tagged with a (non-S) character, for further statistic purposes (see *HISTO and MCDESINT*).

KOBJ = 3 : Distribution of *IMAX* particles inside a 6-D ellipsoid defined by the three sets of data (one set per 2-D phase-space)

$$\begin{array}{llll} \alpha_Y, & \beta_Y, & \dfrac{\varepsilon_Y}{\pi}, & N_{\varepsilon_Y} \ [, \quad N'_{\varepsilon_Y}, \text{ if } N_{\varepsilon_Y} < 0] \\ \alpha_Z, & \beta_Z, & \dfrac{\varepsilon_Z}{\pi}, & N_{\varepsilon_Z} \ [, \quad N'_{\varepsilon_Z}, \text{ if } N_{\varepsilon_Z} < 0] \\ \alpha_X, & \beta_X, & \dfrac{\varepsilon_X}{\pi}, & N_{\varepsilon_X} \ [, \quad N'_{\varepsilon_X}, \text{ if } N_{\varepsilon_X} < 0] \end{array}$$

where α, β are the ellipse parameters and ε/π the *rms* emittance, corresponding to an elliptical frontier $\dfrac{1+\alpha_Y^2}{\beta_Y}Y^2 + 2\alpha_Y YT + \beta_Y T^2 = \varepsilon_Y/\pi$ (*idem* for the (Z, P) or (X, D) planes). N_{ε_Y}, N_{ε_Z} and N_{ε_X} are the sorting cut-offs (used only for Gaussian distributions, $KY, KT, ... = 2$).
The sorting is uniform in surface (for $KY = 1$ or $KZ = 1$ or $KX = 1$) or Gaussian ($KY = 2$ or $KZ = 2$), and so on, as described above. A uniform sorting has the ellipse above for support. A Gaussian sorting has the ellipse above for r.m.s. frontier, leading to $\sigma_Y = \sqrt{\beta_Y \varepsilon_Y/\pi}$, $\sigma_T = \sqrt{\dfrac{(1+\alpha_Y^2)}{\beta_Y}\varepsilon_Y/\pi}$, and similar relations for $\sigma_Z, \sigma_P, \sigma_X, \sigma_D$.
If N_ε is negative, thus the sorting fills the elliptical ring that extends from $|N_\varepsilon|$ to N'_ε (rather than the inner region determined by the N_ε cut-off as discussed above).

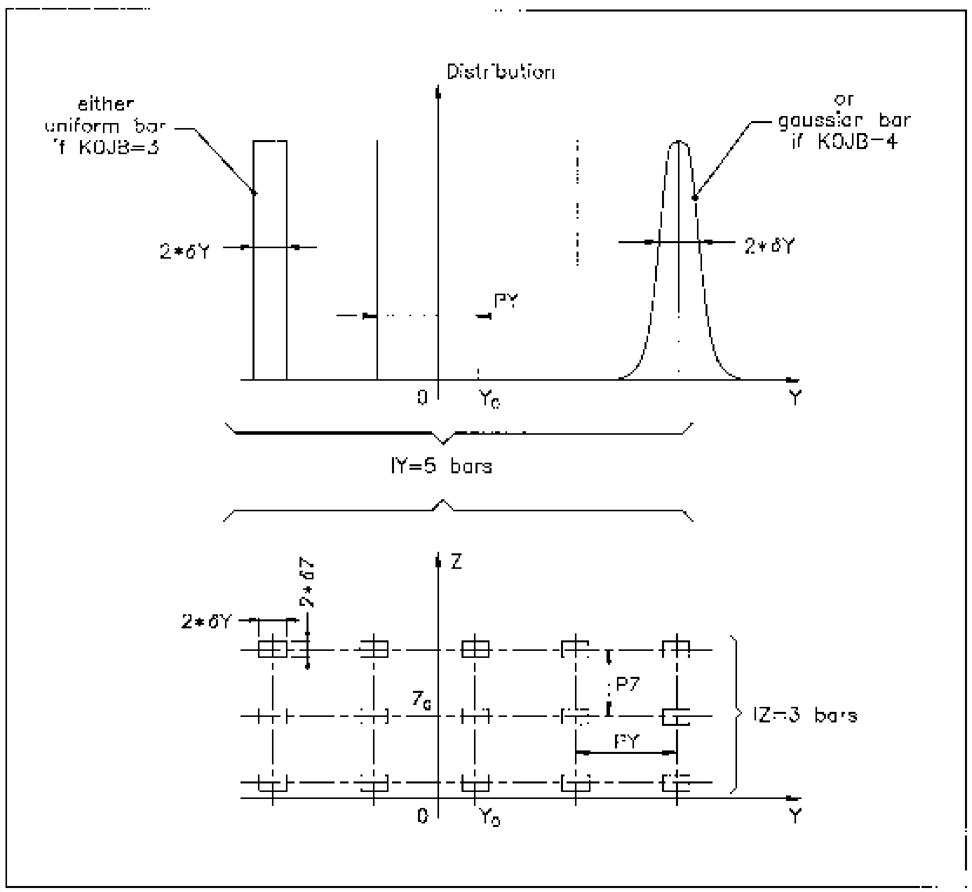


Figure 1: A scheme of input parameters to *MCOBJET* when *KOBJ*= 2.
Top : Possible distributions of the Y coordinate.
Bottom : A 2-D grid in (Y, Z) space.

MCOBJET **Monte-Carlo generation of a 6-D object**

BORO	Reference rigidity	kG.cm	E
KOBJ	Type of support of the random distribution *KOBJ*= 1 : window *KOBJ*= 2 : grid *KOBJ*= 3 : phase-space ellipses	1-3	I
IMAX	Number of particles to be generated	$\leq 10^4$	I
KY, KT, KZ, KP, KX, KD [1]	Type of probability density	6*(1-3)	6*I
Y_0, T_0, Z_0, P_0, X_0, D_0	Mean value of coordinates ($D_0 = B\rho/BORO$)	m, rad, m, rad, m, no dim.	6*E
If KOBJ = 1	**In a window**		
δY, δT, δZ, δP, δX, δD	Distribution widths, depending on KY, KT etc. [1]	m, rad, m, rad, m, no dim.	6*E
$N_{\delta Y}$, $N_{\delta T}$, $N_{\delta Z}$, $N_{\delta P}$, $N_{\delta X}$, $N_{\delta D}$	Sorting cut-offs (used only for Gaussian density)	units of σ_Y, σ_T, etc.	6*E
N_0, C_0, C_1, C_2, C_3	Parameters involved in calculation of P(D)	no dim.	5*E
$IR1$, $IR2$, $IR3$	Random sequence seeds	3*$\simeq 10^6$	3*I

[1] Let $x = Y, T, Z, P$ or X. KY, KT, KZ, KP and KX can take the values

1 : uniform, $p(x) = 1/2\delta x$ if $-\delta x \leq x \leq \delta x$

2 : Gaussian, $p(x) = \exp(-x^2/2\delta x^2)/\delta x\sqrt{2\pi}$

3 : parabolic, $p(x) = 3(1 - x^2/\delta x^2)/4\delta x$ if $-\delta x \leq x \leq \delta x$

KD can take the values

1 : uniform, $p(D) = 1/2\delta D$ if $-\delta D \leq x \leq \delta D$

2 : exponential, $p(D) = \text{No } \exp(C_0 + C_1 l + C_2 l^2 + C_3 l^3)$ if $-\delta D \leq x \leq \delta D$

3 : kinematic, $D = \delta D * T$

If KOBJ = 2	**On a grid**		
IY, IT, IZ, IP, IX, ID	Number of bars of the grid		6*I
PY, PT, PZ, PP, PX, PD	Distances between bars	m, rad, m rad, m, no dim.	6*E
$\delta Y, \delta T, \delta Z, \delta P, \delta X, \delta D$	Width of the bars ($\pm$) if uniform, Sigma value if Gaussian distribution	*ibidem*	6*E
$N_{\delta Y}, N_{\delta T}, N_{\delta Z}, N_{\delta P}, N_{\delta X}, N_{\delta D}$	Sorting cut-offs (used only for Gaussian density)	units of σ_Y, σ_T,etc.	6*E
N_0, C_0, C_1, C_2, C_3	Parameters involved in calculation of $P(D)$	no dim.	5*E
$IR1, IR2, IR3$	Random sequence seeds	3*$\simeq 10^6$	3*I
If KOBJ = 3	**On a phase-space ellipse** [1]		
$\alpha_Y, \beta_Y, \varepsilon_Y/\pi, N_{\sigma_{\epsilon_Y}}$ [, $N'_{\sigma_{\epsilon_Y}}$ if $N_{\sigma_{\epsilon_Y}} < 0$] [2]	Ellipse parameters and emittance, Y-T phase-space ; cut-off	no dim., m/rad, m, units of $\sigma(\varepsilon_Y)$	4*E [,E]
$\alpha_Z, \beta_Z, \varepsilon_Z/\pi, N_{\sigma_{\epsilon_Z}}$ [, $N'_{\sigma_{\epsilon_Z}}$ if $N_{\sigma_{\epsilon_Z}} < 0$] [2]	Ellipse parameters and emittance, Z-P phase-space ; cut-off	no dim., m/rad, m, units of $\sigma(\varepsilon_Z)$	4*E [,E]
$\alpha_X, \beta_X, \varepsilon_X/\pi, N_{\sigma_{\epsilon_X}}$ [, $N'_{\sigma_{\epsilon_X}}$ if $N_{\sigma_{\epsilon_X}} < 0$] [2]	Ellipse parameters and emittance, X-D phase-space ; cut-off	no dim., m/rad, m, units of $\sigma(\varepsilon_X)$	4*E [,E]
$IR1, IR2, IR3$	Random sequence seeds	3*$\simeq 10^6$	3*I

[1] Similar possibilities, non-random, are offered with *OBJET*, KOBJ=8 (p. **??**)

[2] Works with Gaussian density type only : sorting within the ellipse frontier

$$\frac{1+\sigma_Y^2}{\beta_Y^2}Y^2 + 2\alpha_Y YT + \beta_Y T^2 = \frac{\varepsilon_Y}{\pi}$$

if $N_{\sigma_{\epsilon_Y}} > 0$, or, if $N_{\sigma_{\epsilon_Y}} < 0$ sorting within the ring

$$[\,|N_{\sigma_{\epsilon_Y}}|, N'_{\sigma_{\epsilon_Y}}\,]$$

MULTIPOL : Magnetic multipole

The simulation of multipolar magnetic field $\vec{M}$ by ***MULTIPOL*** proceeds by addition of the dipolar ($\vec{B}1$), quadrupolar ($\vec{B}2$), sextupolar ($\vec{B}3$), etc., up to 20-polar ($\vec{B}10$) components, and of their derivatives up to fourth order, following

$$\vec{M} = \vec{B}1 + \vec{B}2 + \vec{B}3 + \ ... \ + \vec{B}10$$

$$\frac{\partial \vec{M}}{\partial X} = \frac{\partial \vec{B}1}{\partial X} + \frac{\partial \vec{B}2}{\partial X} + \frac{\partial \vec{B}3}{\partial X} + \ ... \ + \frac{\partial \vec{B}10}{\partial X}$$

$$\frac{\partial^2 \vec{M}}{\partial X \partial Z} = \frac{\partial^2 \vec{B}1}{\partial X \partial Z} + \frac{\partial^2 \vec{B}2}{\partial X \partial Z} + \frac{\partial^2 \vec{B}3}{\partial X \partial Z} + \ ... \ + \frac{\partial^2 \vec{B}10}{\partial X \partial Z}$$

etc.

The independent components $\vec{B}1$, $\vec{B}2$, $\vec{B}3$, ..., $\vec{B}10$ and their derivatives up to the fourth order are calculated as described in section **??**.

The entrance and exit fringe fields are treated separately. They are characterized by the integration zone X_E at entrance and X_S at exit, as for ***QUADRUPO***, and by the extent λ_E at entrance, λ_S at exit. The fringe field extents for the dipole component are λ_E and λ_S. The fringe field for the quadrupolar (sextupolar, ..., 20-polar) component is given by a coefficient E_2 (E_3, ..., E_{10}) at entrance, and S_2 (S_3, ..., S_{10}) at exit, such that the extent is $\lambda_E * E_2$ ($\lambda_E * E_3$, ..., $\lambda_E * E_{10}$) at entrance and $\lambda_S * S_2$ ($\lambda_S * S_3$, ..., $\lambda_S * S_{10}$) at exit.

If $\lambda_E = 0$ ($\lambda_S = 0$) the multipole lens is considered to have a sharp edge field at entrance (exit), and then, X_E (X_S) is forced to zero (for the mere purpose of saving computing time). If $E_i = 0$ ($S_i = 0$) ($i = 2,\ 10$), the entrance (exit) fringe field for the multipole component i is considered as a sharp edge field. In sharp edge field model, the wedge angle vertical first order focusing effect (if $\vec{B}1$ is non zero) is simulated at magnet entrance and exit by a kick $P_2 = P_1 - Z_1 \tan(\epsilon/\rho)$ applied to each particle (P_1, P_2 are the vertical angles upstream and downstream of the EFB, Z_1 is the vertical particle position at the EFB, ρ the local horizontal bending radius and ϵ the wedge angle experienced by the particle ; ϵ depends on the horizontal angle T).

Any multipole component $\vec{B}i$ can be rotated independently by an angle RXi around the longitudinal X-axis, for the simulation of positioning defects, as well as skew lenses.

Magnet (mis-)alignment is assured by ***KPOS***. ***KPOS*** also allows some degrees of automatic alignment useful for periodic structures (section **??**).

MULTIPOL **Magnetic Multipole**

IL	$IL = 1, 2[\times 10^n]$: print field and coordinates along trajectories	0-2$[\times 10^n]$	I
XL, R_0, $B1$, $B2$, ..., $B10$,	Length of element ; radius at pole tip ; field at pole tip for dipole, quadrupole, ..., dodecapole components	2*cm,10*kG	12*E
	Entrance face		
X_E, λ_E, E_2, ..., E_{10}	Integration zone ; fringe field extent : dipole fringe field extent = λ_E ; quadrupole fringe field extent = $\lambda_E * E_2$; ... 20-pole fringe field extent = $\lambda_E * E_{10}$ (sharp edge if field extent is zero)	2*cm,9*no dim.	11*E
NCE, $C_0 - C_5$	same as *QUADRUPO*	0-6, 6*no dim.	I, 6*E
	Exit face		
X_S, λ_S, S_2, ..., S_{10}	Integration zone ; as for entrance	2*cm, 9*no dim.	11*E
NCS, $C_0 - C_5$		0-6, 6*no dim.	I, 6*E
$R1$, $R2$, $R3$, ..., $R10$	Skew angles of field components	10*rad	10*E
XPAS	Integration step	cm	E
KPOS, *XCE*, *YCE*, *ALE*	*KPOS*=1 : element aligned, 2 : misaligned ; shifts, tilt (unused if *KPOS*=1) for *QUADRUPO*. *KPOS* = 3 : effective only if $B1 \neq 0$: entrance and exit frames are shifted by *YCE* and tilted *wrt.* the magnet by an angle of • either ALE if ALE≠0 • or 2 Arcsin$(B1\,XL\,/\,2BORO)$ if ALE=0	1-3, 2*cm, rad	I, 3*E

OBJET : Generation of an object

OBJET is dedicated to the construction of sets of initial coordinates, in several ways.

The first datum is the reference rigidity (a negative value is allowed)

$$BORO = \frac{p_0}{q}$$

At the object, the beam is defined by a set of *IMAX* particles (the maximum value for *IMAX* is defined in the include file `'MAXTRA.H'`) with the initial conditions (Y, T, Z, P, X, D) with $D = B\rho/BORO$ the relative rigidity.
Depending on the value of the next datum *KOBJ*, these initial conditions may be generated in eight different ways :

KOBJ = 1 : Defines a grid in the Y, T, Z, P, X, D space. One gives the number of points desired

$$IY, \quad IT, \quad IZ, \quad IP, \quad IX, \quad ID$$

with $IY \leq n_Y \ldots ID \leq n_D$ such that $n_Y \times n_T \times ... \times n_D \leq max(IMAX)$. One defines the sampling range in each coordinate

$$PY, \quad PT, \quad PZ, \quad PP, \quad PX, \quad PD$$

zgoubi then generates $IMAX = IY * IT * IZ * IP * IX * ID$ particles with initial coordinates

$$\begin{array}{lllll}
0, & \pm PY, & \pm 2*PY, & \ldots, & \pm IY/2*PY, \\
0, & \pm PT, & \pm 2*PT, & \ldots, & \pm IT/2*PT, \\
0, & \pm PZ, & \pm 2*PZ, & \ldots, & \pm IZ/2*PZ, \\
0, & \pm PP, & \pm 2*PP, & \ldots, & \pm IP/2*PP, \\
0, & \pm PX, & \pm 2*PX, & \ldots, & \pm IX/2*PX, \\
0, & \pm PD, & \pm 2*PD, & \ldots, & \pm ID/2*PD,
\end{array}$$

In this option relative rigidities will be classified automatically in view of possible further use of *IMAGES*$[Z]$ for momentum analysis and image formation.
The particles are tagged with an index *IREP* possibly indicating a symmetry with respect to the (X,Y) plane, as explained in option *KOBJ*= 3. If two trajectories have mid-plane symmetry, only one will be ray-traced, while the other will be deduced using the mid-plane symmetries. This is done for the purpose of saving computing time. It may be incompatible with the use of some procedures (*e.g. MCDESINT*, which involves random processes).
The last datum is a reference in each coordinate, ***YR***, ***TR***, ***ZR***, ***PR***, ***XR***, ***DR***. For instance the reference rigidity is $DR*$ ***BORO***, resulting in the rigidity of a particle of initial condition $I * PD$ to be $(DR + I * PD) *$ ***BORO***.

KOBJ = 1.01: Same as *KOBJ*= 1 except for the Z symmetry. The initial Z and P conditions are the following

$$\begin{array}{lllll}
0, & PZ, & 2*PZ, & \ldots, & (IZ-1)*PZ, \\
0, & PP, & 2*PP, & \ldots, & (IP-1)*PP,
\end{array}$$

This object results in shorter outputs/CPU-time when studying problems with Z symmetry.

KOBJ = 2 : Next data : *IMAX*, *IDMAX*. Initial coordinates are entered explicitly for each trajectory. *IMAX* is the total number of particles. These may be classified in groups of equal number for each value of momentum, in order to fulfill the requirements of image calculations by *IMAGES*[*Z*]. *IDMAX* is the number of groups of momenta. The following initial conditions defining a particle are specified for each one of the *IMAX* particles

$$Y, \quad T, \quad Z, \quad P, \quad X, \quad D, \quad 'A'$$

where $D *$ *BORO* is the rigidity (negative value allowed) and $'A'$ is a (arbitrary) tagging character.

The last record *IEX* (I=1, *IMAX*) contains *IMAX* times either the character "1" to indicates that the particle has to be tracked, or "-9" to indicates that the particle should not be tracked.

This option *KOBJ*= 2 may be be useful for the definition of objects including kinematic effects.

KOBJ = 2.01: Same as *KOBJ*= 2 except for the units, meter and radian in that case.

KOBJ $= 3$ **:** This option allows the reading of initial conditions from an external input file *FNAME*.
The next three data lines are :

IT1, *IT2*, *ITStep*
IP1, *IP2*, *IPStep*

YF, *TF*, *ZF*, *PF*, *SF*, *DPF*, *TiF*, *TAG*
YR, *TR*, *ZR*, *PR*, *SR*, *DPR*, *TiR*
InitC

followed by the storage file name *FNAME*.

IT1, *IT2*, ITStep cause the code to read coordinates of particles number *IT1* through *IT2* by step *ITStep*.

IP1, *IP2*, *IPStep* cause the code to read coordinates belonging in the passes range *IP1* through *IP2*, step *IPStep*.

YF, *TF*, *ZF*, *PF*, *SF*, *DPF*, *TiF* are scaling factors whereas *YR*, *TR*, *ZR*, *PR*, *SR*, *DPR*, *TiR* are references added to the values of respectively Y, T, Z, P, S, DP as read in file *FNAME*, so that any coordinate $C = Y, T, Z...$ is changed into CF*C + CR. In addition a flag character TAG allows retaining only particles with identical tagging letter LET, unless TAG='*' in which case it has no selection effect - for instance TAG='S' can be used to retain only secondary particles following in-flight decay simulations.

If InitC$=1$ ray-tracing starts from the current coordinates$F(J, I)$,
if InitC$=0$ ray-tracing starts from the initial coordinates $FO(J, I)$, as read from file *FNAME*.

The file *FNAME* must be formatted in the appropriate manner. The following *FORTRAN* sequence is an instance, details and possible updates are to be found in the source file `'obj3.f'` :

```
OPEN (UNIT = NL, FILE = FNAME, STATUS = 'OLD')
DO I = 1, IMAX
READ (NL,100) LET (I), IEX(I), (FO(J,I),J=1,6), (F(J,I),J=1,6), I, IREP(I),
>      LET(I),IEX(I),-1.D0+FO(1,I),(FO(J,I),J=2,MXJ),
>      -1.D0+F(1,I),F(2,I),F(3,I),
>      (F(J,I),J=4,MXJ),ENEKI,
>      ID,I,IREP(I), SORT(I),D,D,D,D,RET(I),DPR(I),
>      D, D, D, BORO, IPASS, KLEY,LBL1,LBL2,NOEL
100        FORMAT(1X,
C1   LET(IT),KEX,   1.D0-FO(1,IT),(FO(J,IT),J=2,MXJ),
1           A1,1X,I2,1P,7E16.8,
C2   1.D0-F(1,IT),(FO(J,IT),J=2,MXJ),
2         /,3E24.16,
C3   Z,P*1.D3,SAR,     TAR,     DS,
3         /,4E24.16,E16.8,
C4   KART,  IT,IREP(IT),SORT(IT),X, BX,BY,BZ, RET(IT), DPR(IT),
4         /,I1,2I6,7E16.8,
C5         EX,EY,EZ, BORO,  IPASS,  KLEY,  (LABEL(NOEL,I),I=1,2),NOEL
5  /,4E16.8,          I6,1X,  A8,1X,  2A10,                I5)
ENDDO
```

where the meaning of the parameters (apart from D=dummy real, ID=dummy integer) is the following

LET(I)	:	one-character string (for tagging)
IEX(I)	:	flag, see *KOBJ*= 2 and page **??**
FO(1-6,I)	:	coordinates D, Y, T, Z, P and path length of particle number I, at the origin. $D *$ *BORO* = rigidity
F(1-6,I)	:	id, at the current position.

IREP is an index which indicates a symmetry with respect to the median plane. For instance, if $Z(I+1) = -Z(I)$, then normally *IREP*$(I+1) =$ *IREP*(I). Consequently the coordinates of particle $I+1$ will not be obtained from ray-tracing but instead deduced from those of particle I by simple symmetry. This saves on computing time.

KOBJ$=3$ can be used directly for reading files filled by *FAISCNL*, *FAISTORE*.
If more than *IMAX* particles are to be read from a file, use *REBELOTE*.

Note : In this option, one has to make sure that input data do not conflict with possible use of the keyword *PARTICUL* that assigns mass and charge.

KOBJ = 3.01: Same as **KOBJ = 3**, except for the formatting of trajectory coordinate data in *FNAME*, namely, according to the following *FORTRAN* sequence

```
OPEN (UNIT = NL, FILE = FNAME, STATUS = 'OLD')
1    CONTINUE
READ (NL,*,END=10,ERR=99) Y, T, Z, P, S, D
GOTO 1
10    CALL ENDFIL
99    CALL ERREAD
```

KOBJ = 3.02: As for **KOBJ=3.01**, except for the different format

READ(NL,*) X,Y,Z,PX,PY,PZ

where PX, PY, and PZ, are the momenta in MeV/c. Note that DPR will be ignored in this case.

KOBJ = 3.03: As for **KOBJ=3.01**, except for the different format :

READ(NL,*) DP,Y,T,Z,P,S,TIME,MASS,CHARGE

where MASS is the mass in MeV/c and CHARGE is the charge in units of the elementary charge.

Note : For details and possible updates in the formatted read of concern in the *FORTRAN*, regarding options 3.01-3.03, see the source file 'obj3.f'.

KOBJ = 5 : Mostly dedicated to the calculation of first order transfer matrix and various other optical parameters, using for instance *MATRIX* or *TWISS*. The input data are the coordinate sampling

$$PY, \quad PT, \quad PZ, \quad PP, \quad PX, \quad PD$$

The code generates 11 particles, with initial coordinates

$$0, \quad \pm PY, \quad \pm PT, \quad \pm PZ, \quad \pm PP, \quad \pm PX, \quad \pm PD$$

These values should be small enough, so that the paraxial ray approximation be valid. The last data line gives the reference

$$YR, \quad TR, \quad ZR, \quad PR, \quad XR, \quad DR$$

(with $DR *$ *BORO* the reference rigidity - negative value allowed), which adds to the previous coordinate values.

KOBJ = 5.01: Same as *KOBJ = 5*, except for an additional data line giving initial beam ellipse parameters and dispersions, $\alpha_Y, \beta_Y, \alpha_Z, \beta_Z, \alpha_X, \beta_X, D_Y$, D'_Y , D_Z , D'_Z, for further transport of these using *MATRIX*, or for possible use by the *FIT[2]* procedure.

KOBJ = 5.NN: Like *KOBJ = 5*, except for $NN = 02-99$ references needed in this case (thus *NN-1* additional input data lines), rather than just one. Zgoubi will generate *NN* sets of 11 particles with initial coordinates in each set taken *wrt.* one of the *NN* references.

A subsequent use of *MATRIX* would then cause the computation of *NN* transport matrices.

KOBJ = 6: Mostly dedicated to the calculation of first, second and other higher order transfer coefficients and various other optical parameters, using for instance *MATRIX*. The input data are the coordinate sampling (normally taken paraxial)

$$PY, \quad PT, \quad PZ, \quad PP, \quad PX, \quad PD$$

to allow the building up of an object containing 61 particles (note : their coordinates can be checked by printing out into zgoubi.res using *FAISCEAU*), whereas a last data line gives the reference

$$YR, \quad TR, \quad ZR, \quad PR, \quad XR, \quad DR$$

(with $DR *$ *BORO* the reference rigidity - negative value allowed), which adds to the previous coordinate values.

KOBJ = 7 : Object with kinematics
The data and functioning are the same as for *KOBJ*= 1, except for the following

- ID is not used,
- PD is the kinematic coefficient, such that for particle number I, the initial relative rigidity D_I is calculated from the initial angle T_I following

  $$D_I = DR + PD * T_I$$

  while T_I is in the range

  $$0, \quad \pm PT, \quad \pm 2 * PT, \quad \ldots, \quad \pm IT/2 * PT$$

  as stated under *KOBJ*= 1

KOBJ = 8 : Generation of phase-space coordinates on ellipses.
The ellipses are defined by the three sets of data (one set per ellipse)

$$\begin{array}{ccc} \alpha_Y, & \beta_Y, & \varepsilon_Y/\pi \\ \alpha_Z, & \beta_Z, & \varepsilon_Z/\pi \\ \alpha_X, & \beta_X, & \varepsilon_X/\pi \end{array}$$

where α, β are the ellipse parameters and $\varepsilon/$ is the ellipse surface, corresponding to an ellipse with equation

$$\frac{1+\alpha_Y^2}{\beta_Y}Y^2 + 2\alpha_Y YT + \beta_Y T^2 = \varepsilon_Y/\pi$$

(*idem* for the (Z, P) or (X, D) planes).

The ellipses are centered respectively on (Y_0, T_0), (Z_0, P_0), (X_0, D_0).

The number of samples per plane is respectively IX, IY, IZ. If that value is zero, the central value above is assigned.

OBJET **Generation of an object**

BORO	Reference rigidity	kG.cm	E
KOBJ[.K2]	Option index [.More options]	1-6	I
If KOBJ = 1[.01]	[Non-] Symmetric object		
IY, IT, IZ, IP, IX, ID	Ray-Tracing assumes mid-plane symmetry Total number of points in $\pm Y$, $\pm T$, $\pm Z$, $\pm P$ [$+Z$, $+P$ with KOBJ = 1.01], $\pm X$. and $\pm D$ coordinates ($IY \leq 20$,...,$ID \leq 20$)	IY*IT*IZ*IP*IX*ID $\leq 10^4$	6*1
PY, PT, PZ, PP, PX, PD	Step size in Y, T, Z, P, X and momentum ($PD = \delta B\rho/BORO$)	2(cm,mrad), cm, no dim.	6*E
YR, TR, ZR, PR, XR, DR	Reference ($DR = B\rho/BORO$)	2(cm,mrad), cm, no dim.	6*E
If KOBJ = 2[.01]	All the initial coordinates must be entered explicitly		
IMAX, IDMAX	total number of particles ; number of distinct momenta (if *IDMAX* > 1, group particles of same momentum)	*IMAX* $\leq 10^4$	2*I
For I = 1, IMAX	Repeat *IMAX* times the following line		
Y, T, Z, P, X, D, LET	Coordinates and tagging of the *IMAX* particles ; *If KOBJ = 2.01* input units are different :	2(cm,mrad), cm, no dim., 2(m,rad), m, no dim.,	6*E, A1
IEX($I = 1$, *IMAX*)	*IMAX* times 1 or -9. If *IEX*(I) $= 1$ trajectory I is ray-traced, it is not if *IEX*(I) $= -9$.	1 or -9	*IMAX*I
If KOBJ=3[.NN, NN=00…03]	Reads coordinates from a storage file NN=00 (default) : [b_]zgoubi.fai like data file FORMAT NN=01 : read FORMAT is ``READ(NL,*) Y,T,Z,P,S,DP'' NN=02 : read FORMAT is ``READ(NL,*) X,Y,Z,PX,PY,PZ'' NN=03 : read FORMAT is ``READ(NL,*) DP,Y,T,Z,P,S,TIME,MASS,CHARGE''		
IT1, IT2, ITStep	Read particles numbered IT1 to IT2, step ITStep (For more than 10^4 particles stored in *FNAME*, use '*REBELOTE*')	$\geq 1, \geq IT1, \geq 1$	3*I
IP1, IP2, IPStep	Read particles that belong in pass numbered IP1 to IP2, step IPStep	$\geq 1, \geq IP1, \geq 1$	3*I
YF, TF, ZF, PF, XF, DF, TF, TAG	Scaling factor. TAG-ing letter : no effect if '*', otherwise only particles with TAG≡LET are retained.	7*no.dim, char.	7*E, A1
YR, TR, ZR, PR, XR, DR, TR	Reference. Given the previous line of data, all coordinate C is transformed to C*CF+CR	2(cm, mrad), cm, no dim.,μs	7*E
InitC	0 : set $new\ \vec{R}_0 = old\ \vec{R}_0$, $new\ \vec{R} = old\ \vec{R}$; 1 : set $new\ \vec{R}_0 = old\ \vec{R}$, $new\ \vec{R} = old\ \vec{R}$; 2 : save $old\ \vec{R}$ in $new\ \vec{R}_0$, set $new\ \vec{R} = old\ \vec{R}_0$.	0-1	I
FNAME	File name (*e.g.*, zgoubi.fai) (NN in KOBJ=3.NN determines storage FORMAT)		A80
If KOBJ = 5[.NN, NN=01,99]	**Generation of 11 particles, or 11*NN if** $I \geq 2$ (for use with *MATRIX*, $IORD = 1$)		

PY, PT, PZ, PP, PX, PD	Step sizes in Y, T, Z, P, X and D	2(cm,mrad), cm, no dim.	6*E
YR, TR, ZR, PR, XR, DR	Reference trajectory ($DR = B\rho/BORO$)	2(cm,mrad), cm, no dim.	6*E
If KOBJ = 5.01	additional data line :		
$\alpha_Y, \beta_Y, \alpha_Z, \beta_Z, \alpha_X, \beta_X$, D_Y, D'_Y, D_Z, D'_Z	Initial beam ellipse parameters [1]	2(no dim.,m), ?, ?, 2(m,rad)	6*E, 4*E
If KOBJ = 5.NN, NN=02-99	i = 1 to 98 (if, resp^{ly}, NN=02 to 99) additional data lines :		
YR, TR, ZR, PR, XR, DR	Reference trajectory # i ($DR = B\rho/BORO$)	2(cm,mrad), cm, no dim.	6*E
If KOBJ = 6	**Generation of 61 particles** (for use with *MATRIX*, $IORD = 2$)		
PY, PT, PZ, PP, PX, PD	Step sizes in Y, T, Z, P, X and D	2(cm,mrad), cm, no dim.	6*E
YR, TR, ZR, PR, XR, DR	Reference trajectory ; $DR = B\rho/BORO$	2(cm,mrad), cm, no dim.	6*E
If KOBJ = 7	**Object with kinematics**		
IY, IT, IZ, IP, IX, ID	Number of points in $\pm Y$, $\pm T$,$\pm Z$, $\pm P$, $\pm X$; ID is not used	*IY*IT*IZ*IP*IX*ID*$\leq 10^4$	6*I
PY, PT, PZ, PP, PX, PD	Step sizes in Y, T, Z, P and X ; PD = kinematic coefficient, such that $D(T) = DR + PD * T$	2(cm,mrad), cm, mrad^{-1}	6*E
YR, TR, ZR, PR, XR, DR	Reference ($DR = B\rho/BORO$)	2(cm,mrad), cm, no dim.	6*E
If KOBJ = 8	**Generation of phase-space coordinates on ellipses** [2]		
IY, IZ, IX	Number of samples in each 2-D phase-space ; if zero the central value (below) is assigned	$0 \leq IX, IY, IZ \leq IMAX$, $1 \leq IX * IY * IZ \leq IMAX$	3*I
Y_0, T_0, Z_0, P_0, X_0, D_0	Central values ($D_0 = B\rho/BORO$)	m, rad, m, rad, m, no dim.	6*E
$\alpha_Y, \beta_Y, \varepsilon_Y/\pi$	ellipse parameters and emittances	no dim., m, m	3*E
$\alpha_Z, \beta_Z, \varepsilon_Z/\pi$		no dim., m, m	3*E
$\alpha_X, \beta_X, \varepsilon_X/\pi$		no dim., m, m	3*E

[1] They can be transported by using MATRIX
[2] Similar possibilities, random, are offered with *MCOBJET*, KOBJ=3 (p. **??**)

OBJETA : Object from Monte-Carlo simulation of decay reaction [?]

This generator simulates the reactions

$$M_1 + M_2 \longrightarrow M_3 + M_4$$

and then

$$M_4 \longrightarrow M_5 + M_6$$

where M_1 is the mass of the incoming body ; M_2 is the mass of the target ; M_3 is an outgoing body ; M_4 is the rest mass of the decaying body ; M_5 and M_6 are decay products. Example :

$$p + d \longrightarrow^3 \text{He} + \eta$$
$$\eta \longrightarrow \mu^+ + \mu^-$$

The first input data are the reference rigidity

$$\mathit{BORO} = p_0/q$$

an index *IBODY* which specifies the particle to be ray-traced, namely M3 (*IBODY* = 1), M5 (*IBODY* = 2) or M6 (*IBODY* = 3). In this last case, initial conditions for M6 must be generated by a first run of *OBJETA* with *IBODY* = 2 ; they are then stored in a buffer array, and restored as initial conditions at the next occurrence of *OBJETA* with *IBODY* = 3. Note that **zgoubi** by default assumes positively charged particles.

Another index, *KOBJ*, specifies the type of distribution for the initial transverse coordinates Y, Z ; namely either uniform (*KOBJ*= 1) or Gaussian (*KOBJ*= 2). The other three coordinates T, P and D are deduced from the kinematic of the reactions.

The next data are the number of particles to be generated, *IMAX*, the masses involved in the two previous reactions.

$$M_1, \quad M_2, \quad M_3, \quad M_4, \quad M_5, \quad M_6$$

and the kinetic energy T_1 of the incoming body (M_1).
Then one gives the central value of the distribution for each coordinate

$$Y_0, \quad T_0, \quad Z_0, \quad P_0, \quad D_0$$

and the width of the distribution around the central value

$$\delta Y, \quad \delta T, \quad \delta Z, \quad \delta P, \quad \delta D$$

so that only those particles in the range

$$Y_0 - \delta Y \le Y \le Y_0 + \delta Y \qquad \ldots \qquad D_0 - \delta D \le D \le D_0 + \delta D$$

will be retained. The longitudinal initial coordinate is uniformly sorted in the range

$$-\mathit{XL} \le X_0 \le \mathit{XL}$$

The random sequences involved may be initialized with different values of the two integer seeds IR_1 and IR_2 ($\simeq 10^6$).

Possible use of *PARTICUL* will have no effect : it will not change the mass and charge assumptions as set by *OBJETA*.

0.1 Declaring Options

A series of options are available which allow the control of various of the procedures and functionalities of the code.

Some of these options are normally declared right after the object definition, for instance
- *SPNTRK* : switch-on spin tracking,
- *PARTICUL* to declare particle mass and charge, if for instance tracking in electric fields, or tracking spin, or in presence of synchrotron radiation energy loss simulations,

some may appear further down in the structure (in zgoubi.dat), for instance
- *MCDESINT* : switch-on in-flight decay, could be after a target,
- *REBELOTE* : for multi-turn tracking, including an extraction line section for instance,

others may normally be declared at the end of zgoubi.dat data pile, for instance
- *END* : end of a problem,
- *FIT* : fitting procedure - can also appear before *REBELOTE*

GETFITVAL is an exception rule in that it may appear *before* the object definition (thus becoming the first keyword in zgoubi.dat data list). This is the case if variables prior saved following a '*FIT[2]*' procedure and then read using *GETFITVAL*, happen to belong in the object input data list.

SYSTEM as well, is liable to appear anywhere in the data list.

OBJETA	**Object from Monte-Carlo simulation of decay reaction**		
	$M1 + M2 \longrightarrow M3 + M4$ and $M4 \longrightarrow M5 + M6$		
BORO	Reference rigidity	kG.cm	E
IBODY, KOBJ	Body to be tracked : $M3$ (*IBODY*=1), $M5$ (*IBODY*=2) $M6$ (*IBODY*=3) ; type of distribution for Y_0 and Z_0 : uniform (*KOBJ*= 1) or Gaussian (*KOBJ*= 2)	1-3,1-2	2*I
IMAX	Number of particles to be generated (use '*REBELOTE*' for more)	$\leq 10^4$	I
$M_1 - M_6$	Rest masses of the bodies	6*GeV/c^2	6*E
T_1	Kinetic energy of incident body	GeV	E
Y_0, T_0, Z_0, P_0, D_0	Only those particles in the range $Y_0 - \delta Y \leq Y \leq Y_0 + \delta Y$ .... $D_0 - \delta D \leq D \leq D_0 + \delta D$ will be retained	2(cm,mrad), no dim.	5*E
δY, δT, δZ, δP, δD		2(cm,mrad), no dim.	5*E
XL	Half length of object : $-XL \leq X_0 \leq XL$ (uniform random distribution)	cm	E
$IR1$, $IR2$	Random sequence seeds	2*$\simeq 0^6$	2*I

OCTUPOLE : Octupole magnet (Fig. 1)

The meaning of parameters for *OCTUPOLE* is the same as for *QUADRUPO*. In fringe field regions the magnetic field $\vec{B}(X,Y,Z)$ and its derivatives up to fourth order are derived from the scalar potential approximated to the 8-th order in Y and Z

$$V(X,Y,Z) = \left(G - \frac{G''}{20}\,(Y^2+Z^2) + \frac{G''''}{960}\,(Y^2+Z^2)^2\right)(Y^3Z - YZ^3)$$
$$\text{with } G_0 = \frac{B_0}{R_0^3}$$

The modelling of the fringe field form factor $G(X)$ is described under *QUADRUPO*, p. **??**.

Outside fringe field regions, or everywhere in sharp edge dodecapole ($\lambda_E = \lambda_S = 0$) , $\vec{B}(X,Y,Z)$ in the magnet is given by

$$B_X = 0$$
$$B_Y = G_0(3Y^2 - Z^2)\,Z$$
$$B_Z = G_0(Y^2 - 3Z^2)\,Y$$

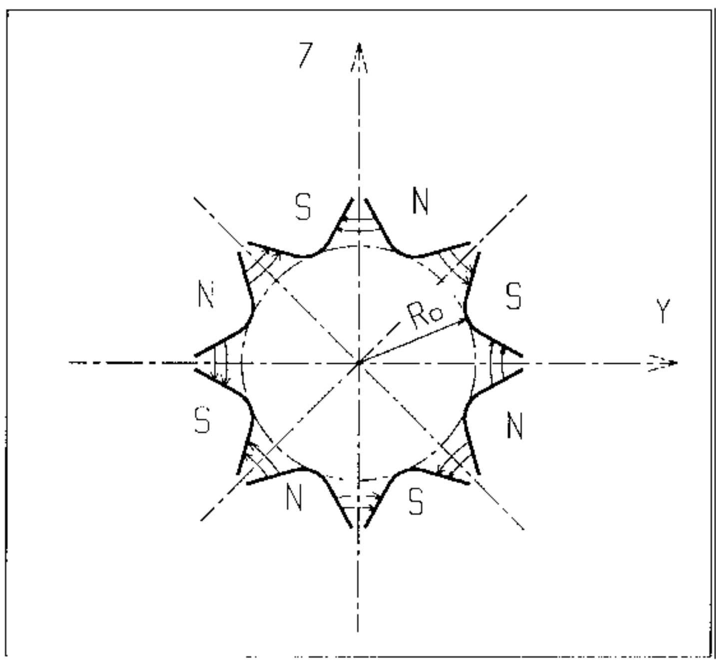
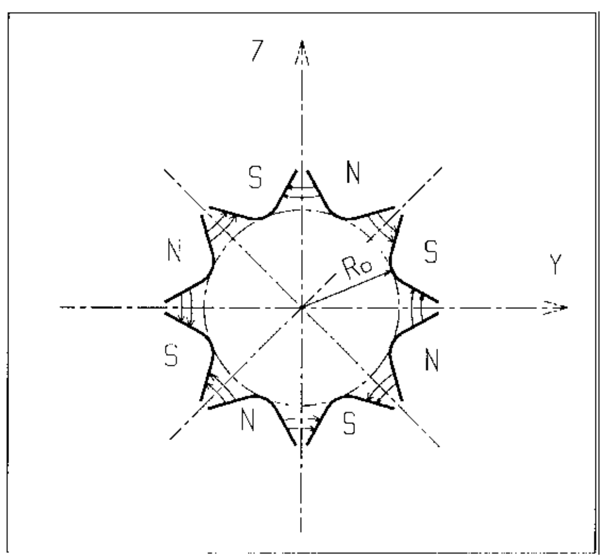


Figure 1: Octupole magnet

OCTUPOLE	**Octupole magnet**		
IL	$IL = 1, 2[\times 10^n]$: print field and coordinates along trajectories.	0-2$[\times 10^n]$	I
XL, R_0, B_0	Length ; radius and field at pole tip of the element	2*cm, kG	3*E
	Entrance face :		
X_E, λ_E	Integration zone ; Fringe field extent ($\lambda_E = 0$ for sharp edge)	2*cm	2*E
NCE, $C_0 - C_5$	*NCE* = unused $C_0 - C_5$ = fringe field coefficients such that : $G(s) = G_0/(1 + \exp\ P(s))$, with $G_0 = B_0/R_0^3$ and $P(s) = \sum_{i=0}^{5} C_i(s/\lambda)^i$	any, 6*no dim.	I, 6*E
	Exit face :		
X_S, λ_S	Parameters for the exit fringe field ; see entrance	2*cm	2*E
NCS, $C_0 - C_5$		0-6, 6*no dim.	I, 6*E
XPAS	Integration step	cm	E
KPOS, *XCE*, *YCE*, *ALE*	*KPOS*=1 : element aligned, 2 : misaligned ; shifts, tilt (unused if *KPOS*=1)	1-2, 2*cm, rad	I, 3*E



Octupole magnet

OPTICS : Write out optical functions

OPTICS normally appears next to object definition, it normally works in conjunction with element label(s).
OPTICS causes the transport and write out, in zgoubi.res, of the 6×6 beam matrix, following options *KOPT* and '*label*', below.

IF *KOPT=0* : Off
IF *KOPT=1* : Will transport the optical functions with initial values as specified in *OBJET*, option *KOBJ=5.01*.

Note : The initial values in *OBJET[KOBJ=5.01]* may be the periodic ones, as obtained, for instance, from a first run using *MATRIX[IFOC=11]*.

A second argument, '*label*', allows

- if *label = all* : printing out, into zgoubi.res, after all keywords of the zgoubi.dat structure,
- otherwise, printing out at all keyword featuring *LABEL* $\equiv$ *label* as a first label (see section **??**, page **??**, regarding the labelling of keywords).

A third argument, *IMP=1*, will cause saving of the transported beta functions into file zgoubi.OPTICS.out.

OPTICS **Write out optical functions**

IOPT, label, *IMP*	$IOPT = 0/1$: Off/On. Transport the beam matrix ; 'label' : Can be 'all', 'ALL', or existing 'LABEL_1(NOEL)' ; $IMP = 1$ causes storage of optical functions in zgoubi.OPTICS.out.	0-1, string, 0-1	I, A, I

OPTIONS : Global options

OPTIONS allows switching various options.

Available, for now :

- Inhibit (most of) write statements to zgoubi.res
  Form of the statement : "WRITE   -1"
  Back to normal : "WRITE   +1"

OPTIONS	**Global options**		
IOPT, NBOP	$IOPT = 0/1$: Off/On. NBOP : total number of options.	0-1, ≥ 0	2*I

NBOP lines should follow. Possible choices :

WRITE, ± 1	-1 inhibites (most of) write statements to zgoubi.res	'WRITE', ± 1	A,I

ORDRE : Taylor expansions order

The position $\vec{R}$ and velocity $\vec{u}$ of a particle are obtained from Taylor expansions as described in eq. (**??**). By default, these expansions are up to the fifth order derivative of $\vec{u}$,

$$\vec{R}_1 \approx \vec{R}_0 + \vec{u}\Delta s + ... + \vec{u}^{(5)}\,\frac{\Delta s^6}{6!}$$
$$\vec{u}_1 \approx \vec{u} + \vec{u}\,'\Delta s + \ldots + \vec{u}^{(5)}\,\frac{\Delta s^5}{5!}$$

which corresponds to fourth order derivatives of fields $\vec{B}$, eq. (**??**). and of $\vec{E}$, eq. (**??**).

However, third or higher order derivatives of fields may be zero in some optical elements, for instance in a sharp edge quadrupole. Also, in several elements, no more than first and second order field derivatives are implemented in the code. One may also wish to save on computation time by limiting the time-consuming calculation of lengthy (while possibly ineffective in terms of accuracy) Taylor expansions.

In that spirit, the purpose of *ORDRE*, option $IO = 2-5$, is to allow for expansions to the $\vec{u}^{(n)}$ term in eq. **??**. Default functioning is $IO = 4$, stated in *FORTRAN* file `block.f`.

Note the following :
As concerns the optical elements

DECAPOLE, DODECAPO, EBMULT, ELMULT, MULTIPOL, OCTUPOLE,
QUADRUPO, SEXTUPOL

field derivatives (see eq. **??** p. **??**, eq. **??** p. **??**,) have been installed in the code according to $\vec{u}^{(5)}$ Taylor development order ; it may not be as complete for other optical elements. In particular, in electric optical elements field derivatives (eq. **??**) are usually provided to no more than second order, which justifies saving on computing time by means of *ORDRE*, so to avoid pushing Taylor expansions as high as $\vec{u}^{(5)}$.

NOTE : see also the option *IORDRE* in field map declarations (*DIPOLE-M, TOSCA*, etc.).

ORDRE	**Taylor expansions order**		
IO	Taylor expansions of $\vec{R}$ and $\vec{u}$ up to $\vec{u}^{(IO)}$ (default is $IO = 4$)	2-5	I

PARTICUL : Particle characteristics

Since **zgoubi** works using the rigidity, (*BORO*, as declared in *[MC]OBJET*), *PARTICUL* only needs be introduced (normally, following *[MC]OBJET* in the input data file zgoubi.dat) when the definition of some characteristics of the particles (mass, charge, gyromagnetic factor, life-time in the center of mass) is needed, as is the case when using the following procedures :

CAVITE	: mass, charge
MCDESINT	: mass, COM life-time
SPNTRK	: mass, gyromagnetic factor
SRLOSS	: mass, charge
SYNRAD	: mass, charge
Electric and Electro-Magnetic elements	: mass, charge

The declaration of *PARTICUL* must **precede** these keywords.

If *PARTICUL* is omitted, which is in general the case when ray-tracing ions in purely magnetic optical assemblies, then **zgoubi**, since it only knows the rigidity, will skip the computation of such quantities as time of flight.

PARTICUL **Particle characteristics**

M, Q, G, τ, X	Mass ; charge ; gyromagnetic factor ; COM life-time ; unused	MeV/c^2, C, no dim., s	5*E

If M is of the form $\{$`M1 M2`$\}$, then when masses are assigned to particles from a previously defined object, the first half of the particles are given the mass `M1`, and the second half are given the mass `M2`.
If Q is zero, the reference charge is left unchanged.
NOTE : Only the parameters of concern need their value be specified (for instance M, Q when electric lenses are used) ; others can be set to zero.

PICKUPS : Beam centroid path; closed orbit

PICKUPS computes the coordinates of the beam centroid, at one or more *LABEL*'ed keyword(s). These coordinates are the average values of the coordinates of the particles in a bunch. That (list of) *LABEL*(s) is specified by the user, as part of the arguments under the keyword *PICKUPS*.

In conjunction with *REBELOTE* in the case of a periodic structure, *PICKUPS* thus effectively delivers the closed orbit coordinates.

PICKUPS **Beam centroid path; closed orbit**

N	0 : inactive ≥ 1 : number of *LABELs* at which beam centroid is computed	≥ 0	I
For I = 1, N	A list of N keywords' labels follows		
*LABEL*1 [,LABEL2, [...]]	The N labels at which beam centroid is to be computed/recorded. In case a "*LABELi*" in the list does not exist, it is peacefully ignored.	N string(s)	N*A10

Example

A trick :

```
'PICKUPS'
1
none labelA labelB ...
```

This is a possible way to inhibited an earlier use of *PICKUPS* with "labelA, labelB, ..." keyword list. It is sufficient (and necessary) for that, that no keyword in zgoubi.dat data list has "none" as a its first *LABEL*.

PLOTDATA : Intermediate output for the PLOTDATA graphic software [?]

PLOTDATA was at the origin implemented for the purpose of plotting particle coordinates using the TRIUMF PLOTDATA package. However nothing precludes using it with a different aim.

The *PLOTDATA* keyword can be introduced at up to 20 locations in zgoubi.dat. There, particle coordinates will be stored in a local array, `FF`. They are overwritten at each pass. Usage of `FF` is left to the user, see *FORTRAN* subroutine `pltdat.f`.

PLOTDATA **Intermediate output for the PLOTDATA graphic software**

To be documented.

POISSON :Read magnetic field data from *POISSON* output

This keyword allows reading a field profile $B(X)$ from *POISSON* output. Let *FNAME* be the name of this output file (normally, *FNAME* = outpoi.lis) ; the data are read following the *FORTRAN* statements here under

```
I = 0
11    CONTINUE
I = I + 1
READ(LUN,101,ERR=10,END=10) K, K, K, R, X(I), R, R, B(I)
101     FORMAT(I1, I3, I4, E15.6, 2F11.5, 2F12.3)
GOTO 11
10    CONTINUE
...
```

where X(I) is the longitudinal coordinate, and B(I) is the Z component of the field at a node (I) of the mesh. K's and R's are dummy variables appearing in the *POISSON* output file outpoi.lis but not used here.

From this field profile, a 2-D median plane map is built, with a rectangular and uniform mesh ; mid-plane symmetry is assumed. The field at each node (X_i, Y_j) of the map is $B(X_i)$, independent of Y_j (*i.e.*, the distribution is uniform in the Y direction).

For the rest, *POISSON* works in a way similar to *CARTEMES*.

POISSON **Read magnetic field data from *POISSON* output**

IC, IL	$IC = 1, 2$: print the field map $IL = 1, 2[\times 10^n]$: print field and coordinates along trajectories.	0-2, 0-2$[\times 10^n]$	2*I
BNORM, XN,YN	Field and X-,Y-coordinate normalization coeffs.	3*no dim.	3*E
TITL	Title. Start with "FLIP" to get field map X-flipped		A80
IX, IY	Number of longitudinal and transverse nodes of the uniform mesh	≤ 400, ≤ 200	2*I
FNAME [1]	File name		A80
ID, A, B, C [, A', B', C', B'', etc., if $ID \geq 2$]	Integration boundary. Ineffective when $ID = 0$. ID = -1, 1 or ≥ 2 : as for *CARTEMES*	≥ -1, 2*no dim., cm [,2*no dim., cm, etc.]	I,3*E [,3*E,etc.]
IORDRE	Degree of interpolation polynomial as for *DIPOLE-M*	2, 25 or 4	I
XPAS	Integration step	cm	E
KPOS, *XCE*, *YCE, ALE*	*KPOS*=1 : element aligned, 2 : misaligned ; shifts, tilt (unused if *KPOS*=1)	1-2, 2*cm, rad	I, 3*E

[1] *FNAME* (*e.g.*, ''outpoi.lis'') contains the field map data.
These must be formatted according to the following *FORTRAN* read sequence - details
and possible updates are to be found in the source file `'fmapw.f'` :

```
I = 0
11 CONTINUE
I = I+1
READ(LUN,101,ERR=99,END=10) K, K, K, R, X(I), R, R, B(I)
101    FORMAT(I1, I3, I4, E15.6, 2F11.5, 2F12.3)
GOTO II
10    CONTINUE
```

where $X(I)$ is the longitudinal coordinate, and $B(I)$ is the Z component of the field at a node (I) of the mesh.
K's and R's are variables appearing in the *POISSON* output file outpoi.lis, not used here.

POLARMES : 2-D polar mesh magnetic field map

Similar to *CARTEMES*, apart from the polar mesh frame : IX is the number of angular nodes, JY the number of radial nodes ; $X(I)$ and $Y(J)$ are respectively the angle and radius of a node (these parameters are similar to those entering in the definition of the field map in *DIPOLE-M*).

POLARMES **2-D polar mesh magnetic field map**
mid-plane symmetry is assumed

IC, IL	$IC = 1, 2$: print the map $IL = 1, 2[\times 10^n]$: print field and coordinates along trajectories.	0-2, 0-2$[\times 10^n]$	2*I
BNORM, AN,RN	Field and A-,R-coordinate normalization coeffs.	3*no dim.	3*E
TITL	Title. Start with “FLIP” to get field map X-flipped		A80
IA, JR	Number of angular and radial nodes of the mesh	≤ 400, ≤ 200	2*I
FNAME [1]	File name		A80
ID, A, B, C [, A', B', C', B'', etc., if $ID \geq 2$]	Integration boundary. Ineffective when $ID = 0$. ID = -1, 1 or ≥ 2 : as for *CARTEMES*	≥ -1, 2*no dim., cm [,2*no dim., cm, etc.]	I,3*E [,3*E,etc.]
IORDRE	Degree of interpolation polynomial (see *DIPOLE-M*)	2, 25 or 4	I
XPAS	Integration step	cm	E
KPOS	as for *DIPOLE-M*. Normally 2.	1-2	I
If KPOS = 2			
RE, TE, RS, TS		cm, rad, cm, rad	4*E
If KPOS = 1			
DP		no dim.	E

[1] *FNAME* (*e.g.*, spes2.map) contains the field data.
These must be formatted according to the following *FORTRAN* read sequence - details and possible updates are to be found in the source file `'fmapw.f'` :

```
OPEN (UNIT = NL, FILE = FNAME, STATUS = 'OLD' [,FORM='UNFORMATTED'])
IF (BINARY) THEN
READ(NL) (Y(J), J=1, JY)
ELSE
READ(NL,100) (Y(J), J=1, JY)
ENDIF
100     FORMAT(10 F8.2)
DO 1 I = 1,IX
IF (BINARY) THEN
READ (NL) X(I), (BMES(I,J), J=1, JY)
ELSE
READ(NL,101) X(I), (BMES(I,J), J=1, JY)
101   FORMAT(10 F8.1)
ENDIF
1    CONTINUE
```

where $X(I)$ and $Y(J)$ are the longitudinal and transverse coordinates and *BMES* is the Z field component at a node (I, J) of the mesh. For binary files, *FNAME* must begin with 'B_' or 'b_'. ‘Binary’ will then automatically be set to ‘.TRUE.’

PS170 : Simulation of a round shape dipole magnet

PS170 is dedicated to a 'rough' simulation of CERN *PS170* spectrometer dipole.

The field B_0 is constant inside the magnet, and zero outside. The pole is a circle of radius R_0, centered on the X axis. The output coordinates are generated at the distance *XL* from the entrance (Fig. 1).

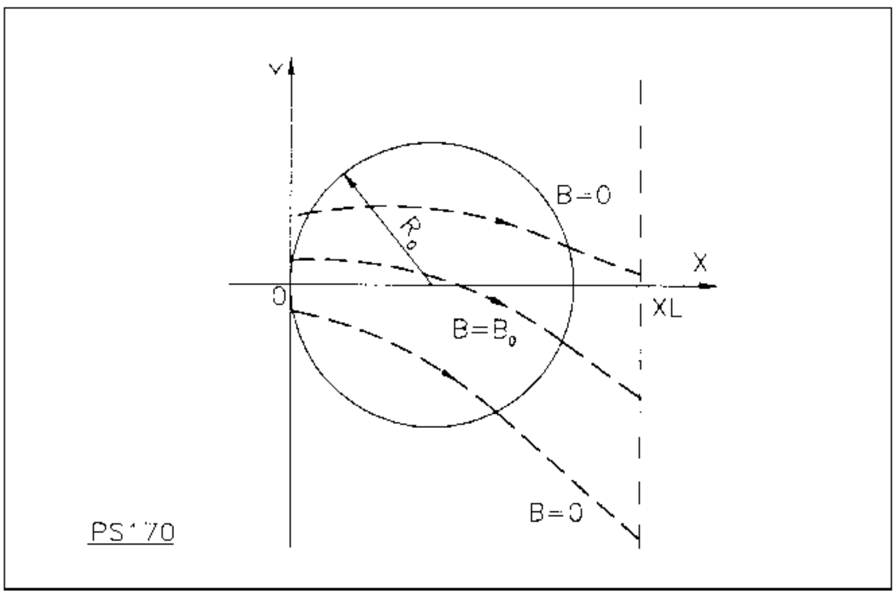
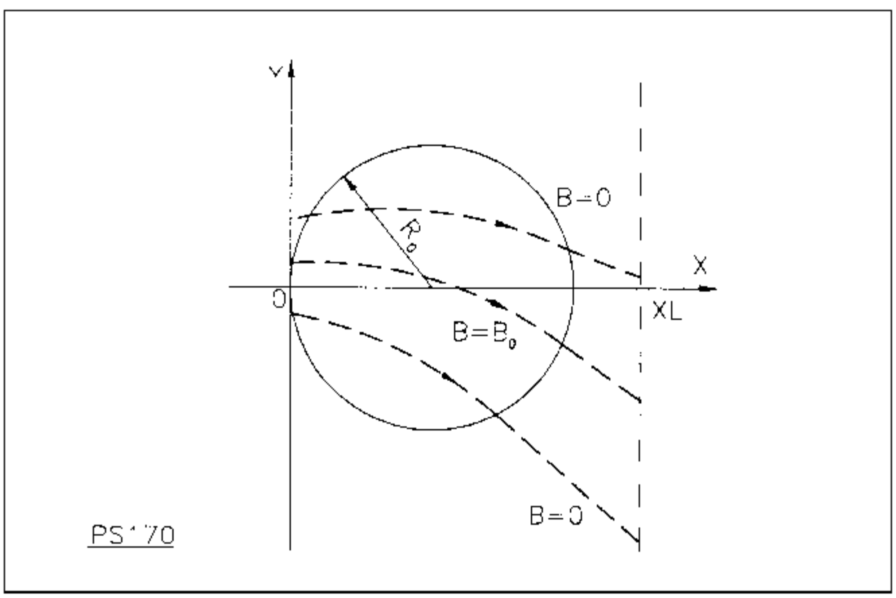


Figure 1: Scheme of the PS170 magnet simulation.

PS170 **Simulation of a round shape dipole magnet**

IL	$IL = 1, 2[\times 10^n]$: print field and coordinates along trajectories.	0-2$[\times 10^n]$	I
XL, R_0, B_0	Length of the element, radius of the circular dipole, field	2*cm, kG	3*E
XPAS	Integration step	cm	E
KPOS, *XCE*, *YCE*, *ALE*	*KPOS*=1 : element aligned, 2 : misaligned ; shifts, tilt (unused if *KPOS*=1)	1-2, 2*cm, rad	I, 3*E



Scheme of the PS170 magnet simulation.

QUADISEX **Sharp edge magnetic multipoles**

$B_Z \mid_{Z=0} = B_0 \left(1 + \frac{N}{R_0} Y + \frac{B}{R_0^2} Y^2 + \frac{G}{R_0^3} Y^3\right)$

IL	$IL = 1, 2[\times 10^n]$: print field and coordinates along trajectories.	0-2$[\times 10^n]$	I
XL, R_0, B_0	Length of the element ; normalization distance ; field	2*cm, kG	3*E
N, $EB1$, $EB2$, $EG1$, $EG2$	Coefficients for the calculation of B. if $Y > 0$: $B = EB1$ and $G = EG1$; if $Y < 0$: $B = EB2$ and $G = EG2$.	5*no dim.	5*E
XPAS	Integration step	cm	E
KPOS, *XCE*, *YCE*, *ALE*	*KPOS*=1 : element aligned, 2 : misaligned ; shifts, tilt (unused if *KPOS*=1)	1-2, 2*cm, rad	I, 3*E

QUADRUPO : Quadrupole magnet (Fig. 1)

The length of the magnet *XL* is the distance between the effective field boundaries (EFB), Fig. 2. The field at the pole tip R_0 is B_0.
The extent of the entrance (exit) fringe field is characterized by $\lambda_E(\lambda_S)$. The distance of ray-tracing on both sides of the EFB's, in the field fall off regions, will be $\pm\, X_E$ at the entrance, and $\pm\, X_S$ at the exit (Fig. 2), by prior and further automatic change of frame.
In the fringe field regions $[-X_E, X_E]$ and $[-X_S, X_S]$ on both sides of the EFB's, $\vec{B}(X,Y,Z)$ and its derivatives up to fourth order are calculated at each step of the trajectory from the analytical expressions of the three components B_X, B_Y, B_Z obtained by differentiation of the scalar potential (see section **??**) expressed to the 8th order in Y and Z.

$$V(X,Y,Z) = \left(G - \frac{G''}{12}\,(Y^2+Z^2) + \frac{G''''}{384}\,(Y^2+Z^2)^2 - \frac{G''''''}{23040}\,(Y^2+Z^2)^3 \right) YZ$$
$$(\quad G'' = d^2G/dX^2, ...)$$

where G is the gradient on axis [**?**] :

$$G(s) = \frac{G_0}{1+\exp P(s)} \qquad \text{with} \quad G_0 = \frac{B_0}{R_0}$$

and,

$$P(s) = C_0 + C_1\left(\frac{s}{\lambda}\right) + C_2\left(\frac{s}{\lambda}\right)^2 + C_3\left(\frac{s}{\lambda}\right)^3 + C_4\left(\frac{s}{\lambda}\right)^4 + C_5\left(\frac{s}{\lambda}\right)^5 P(s) = C_0 + C_1\left(\frac{s}{\lambda}\right) + C_2\left(\frac{s}{\lambda}\right)$$

where, s is the distance to the field boundary and λ stands for λ_E or λ_S (normally, $\lambda \simeq 2 * R_0$).
When fringe fields overlap inside the magnet (*XL* $\leq X_E + X_S$), the gradient G is expressed as

$$G = G_E + G_S - 1$$

where, G_E is the entrance gradient and G_S is the exit gradient.
If $\lambda_E = 0$ ($\lambda_S = 0$), the field at entrance (exit) is considered as sharp edged, and then $X_E(X_S)$ is forced to zero (for the mere purpose of saving computing time).
Outside of the fringe field regions (or everywhere when $\lambda_E = \lambda_S = 0$) $\vec{B}(X,Y,Z)$ in the magnet is given by

$$B_X = 0$$
$$B_Y = G_0 Z$$
$$B_Z = G_0 Y$$

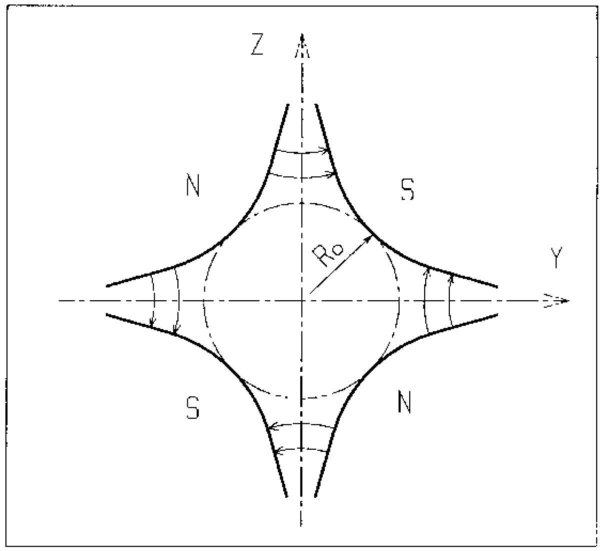


Figure 1: Quadrupole magnet

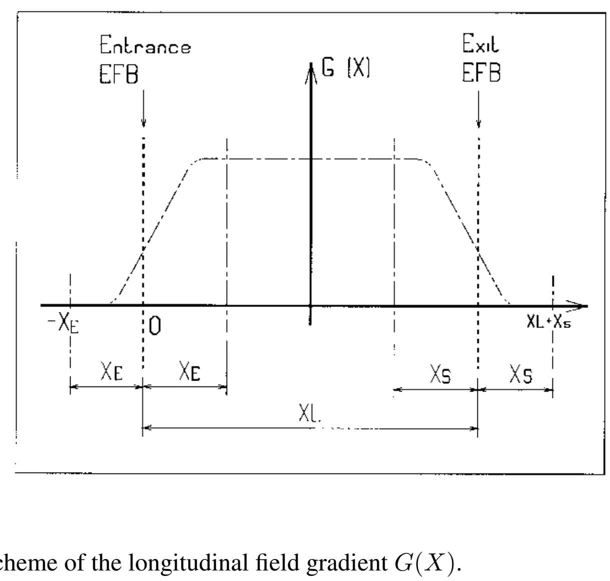


Figure 2: Scheme of the longitudinal field gradient $G(X)$.
(OX) is the longitudinal axis of the reference frame $(0, X, Y, Z)$ of **zgoubi**. The length of the element is *XL*. Trajectories are ray-traced from $-X_E$ to *XL*$+X_S$, by means of respectively prior and final automatic change of frame.

QUADRUPO **Quadrupole magnet**

IL	$IL = 1, 2[\times 10^n]$: print field and coordinates along trajectories.	0-2$[\times 10^n]$	I
XL, R_0,B_0	Length ; radius and field at pole tip	2*cm, kG	3*E
	Entrance face :		
X_E, λ_E	Integration zone extent ; fringe field extent ($\simeq 2R_0$, $\lambda_E = 0$ for sharp edge)	2*cm	2*E
NCE, $C_0 - C_5$	*NCE* = unused $C_0 - C_5$= Fringe field coefficients such that $G(s) = G_0/(1 + \exp P(s))$, with $G_0 = B_0/R_0$ and $P(s) = \sum_{i=0}^{5} C_i(s/\lambda)^i$	any, 6*no dim.	I, 6*E
	Exit face		
X_S, λ_S	See entrance face	2*cm	2*E
NCS, $C_0 - C_5$		0-6, 6*no dim.	I, 6*E
XPAS	Integration step	cm	E
KPOS, *XCE*, *YCE, ALE*	*KPOS*=1 : element aligned, 2 : misaligned ; shifts, tilt (unused if *KPOS*=1)	1-2, 2*cm, rad	I, 3*E

REBELOTE : 'Do it again'

When *REBELOTE* is encountered in the input data file, the code execution jumps,

- either back to the beginning of the data file - the default behavior,
- or (option *K=99.1* or *K=99.2*) back to a particular *LABEL*.

Then *NPASS-1* passes (from *LABEL* to *REBELOTE*) follow.
As to the last pass, number *NPASS+1*, there are two possibilities :

- either it also encompasses the whole *LABEL* to *REBELOTE* range,
- or, upon request (option *K=99.2*), execution may exit that final pass upstream of *REBELOTE*, at a location defined by a second dedicated *LABEL* placed between the first above mentioned *LABEL*, and *REBELOTE*. In both cases, following the end of this "multiple-pass" procedure, the execution continues from the keyword which follows *REBELOTE*, until *'END'* is encountered.

The two functionalities of *REBELOTE* are the following :

• *REBELOTE* can be used for Monte Carlo simulations when more than Max(*IMAX*) particles are to be tracked. Thus, when the following random procedures are used : *MCOBJET*, *OBJETA*, *MCDESINT*, *SPNTRK* (*KSO* = 5), their random seeds are not reset and independent statistics will add up.
This includes **Monte Carlo simulations**, in beam lines : normally $K = 0$. *NPASS* runs through the same structure, from *MCOBJET* to *REBELOTE* will follow, resulting in the calculation of $(1 + \textit{NPASS}) * \textit{IMAX}$ trajectories, with as many random initial coordinates.

• *REBELOTE* can be used for multi-turn ray-tracing in circular machines **circular machines** : normally $K = 99$ in that case. *NPASS* turns in the same structure will follow, resulting in the tracking of *IMAX* particles over $1 + \textit{NPASS}$ turns. For the simulation of pulsed power supplies, synchrotron motion, and other Q-jump manipulation, see *SCALING*.
For instance, using option described *K=99.2* above, a full "injection line + ring + extraction line" installation can be simulated - kicker firing and other magnet ramping can be simulated using *SCALING*.
Using the double-*LABEL* method discussed above with option *K=99.2*, it is possible to encompass the ring between an injection line section (namely, with the element sequence of the latter extending from *OBJET* to the first *LABEL*), and an extraction line (its description will then follow *REBELOTE*), whereas the ring description extends from to the first *LABEL* to *REBELOTE*, with possible extraction, at the last pass, at the location of the second *LABEL*, located between the first one and *REBELOTE*,

Output prints over *NPASS+1* passes might result in a prohibitively big zgoubi.res file. They may be switched on/off by means of the option *KWRIT*= $i.j$, with $i = 1/0$ respectively. The j flag commands printing pass number and some other information onto the video output, every 10^{j-1} turns if $j > 0$; output is switched off if $j = 0$.

REBELOTE also provides information : statistical calculations and related data regarding particle decay *(MCDESINT)*, spin tracking *(SPNTRK)*, stopped particles (*CHAMBR*, *COLLIMA*), etc.

COMBINING REBELOTE AND FIT[2] The keyword *REBELOTE* can follow *FIT[2]*. This allows executing again the same fit procedure, after having changed the value of some parameter in zgoubi.dat. That's the role of *REBELOTE* in that game : it changes that parameter, and then sends the zgoubi execution pointer back to the top of zgoubi.dat for a new run.

An example : see page **??**

REBELOTE **Jump to the beginning of zgoubiinput data file**

NPASS, *KWRIT*, K*[.n]*, *[, Label1 [, Label2]]*	NPASS : Number of runs ; *KWRIT* = 1.1 (resp. 0.0) switches (inhibits) *FORTRAN WRITE*s to .res and to screen ; K option : $K = 0$: initial conditions (coordinates and spins) are generated following the regular functioning of object definitions. If random generators are used (*e.g.* in *MCOBJET*) their seeds will not be reset. $K = 22$: next run will account for new parameter values in zgoubi.dat data list, see below. $K = 99$: coordinates at end of previous pass are used as initial coordinates for the next pass ; *idem* for spin components. $K = 99.1$: Label1 is expected, subsequent passes will start from element with Label1 down to *REBELOTE* and so forth ; $K = 99.2$: Label1 and Label2 are expected ; last pass (# NPASS+1) will end at element with Label2 whereupon execution will jump to the keyword next to *REBELOTE* and will be carried out down to *'END'*.	arbitrary ; 0-1 ; 0, 22, 99	3*I 2A10

If K = 22 [1]

NPRM	Number of parameters to be changed for next runs		I

Repeat NPR times the following sequence (tells parameters concerned, and for each its successive values) :

LMNT, *PRM*, NV*Val	Keyword # in zgoubi.dat list ; parameter # under that Keyword (same as for FIT[2], see page **??**) ; *NV* successive values (if $NV < NPASS$ then last value is maintained over remaining passes).	-, -, NV*dim[2]	2*I, NV*E

[1] K=22 is compatible with use of the *FIT[2]* procedure : *e.g.*, allows successive *FIT*s in a run, with successive sets of optical parameters.

[2] V is in current **zgoubi** units in the case of particle coordinates (cm, mrad). It is in MKSA units (m, rad) in the case of matrix coefficients.

RESET **Reset counters and flags**

Resets counters involved in *CHAMBR*, *COLLIMA* *HISTO* and *INTEG* procedures

Switches off *CHAMBR*, *MCDESINT*, *SCALING* and *SPNTRK* options

SCALING : Power supplies and R.F. function generator

SCALING acts as a function generator dedicated to varying fields in optical elements, potentials in electrostatic devices, RF parameters in *CAVITE*. It is normally intended to be declared right after the object definition, and used in conjunction with *REBELOTE*, for the simulation of multi-turn tracking - possibly including acceleration cycles.

SCALING acts on families of elements, a family being designated by its name that coincides with the keyword of the corresponding element. For instance, declaring *MULTIPOL* as to be varied will result in the same timing law being applied to all *MULTIPOL*'s in the **zgoubi** optical structure data file. Subsets can be selected by labeling keywords in the data file (section **??**, page **??**) and adding the corresponding *LABEL*(s) in the *SCALING* declarations (two *LABEL*'s maximum). The family name of concern, as well as the scaling function for that family, are given as input data to the keyword *SCALING*. There is an upper limit, *NFMAX*, to the number NF of families that can be declared as subject to a scaling law, *NFMAX* can be changed in the *FORTRAN* include file `MXFS.H`. A scaling law can be comprised of up to NT successive timings, between two successive timings, a linear interpolation law is used to determine the scaling factor.

An example of data formatting for the simulation of an acceleration cycle in a circular machine is given in the following.

SCALING		- Scaling
1 4		Active. $NF = 4$ families of elements are concerned, as listed below
QUADRUPO QFA QFB		- Quadrupoles labeled 'QFA' and Quadrupoles labeled 'QFB'
2		$NT = 2$ timings
18131.E-3	24176.E-3	The field increases (linearly) from 18131E-3$*B_0$ to 24176E-3$*B_0$
1	6379	from turn 1 to turn 6379
MULTIPOL QDA QDB		- Multipoles labeled 'QDA' and Multipoles labeled 'QDB'
2		$NT = 2$ timings
18131.E-3	24176.E-3	Fields increase from 18131E-3$*B_i$ to 24176E-3$*B_i$ ($\forall i = 1,\ 10$ poles)
1	6379	from turn 1 to turn 6379
BEND		- All *BEND*'s (regardless of any *LABEL*)
2		$NT = 2$ timings
18131.E-3	24176.E-3	As above
1	6379	
CAVITE		- Accelerating cavity
3		$NT = 3$ timings
1 1.22	1.33352	The synchronous rigidity $(B\rho)_s$ increases,
1 1200	6379	from $(B\rho)_{s_o}$ to 1.22 $*(B\rho)_{s_o}$ from turn 1 to 1200, and
		from 1.22 $*(B\rho)_{s_o}$ to 1.33352 $(B\rho)_{s_o}$ from turn 1200 to 6379

The timing is in unit of turns. In this example, TIMING = 1 to 6379 (turns). Therefore, at turn number N, B and B_i are updated in the following way. Let *SCALE*$(TIMING = N)$ be the updating scale factor

$$\text{SCALE}(N) = 18.131 \frac{24.176 - 18.131}{1 + 6379 - 1}(N-1)$$

and then

$$B(N) = \textit{SCALE}(N) B_0$$
$$B_i(N) = \textit{SCALE}(N) B_{i0}$$

The RF frequency is computed using

$$f_{RF} = \frac{hc}{\mathcal{L}} \frac{q(B\rho)_s}{(q^2(B\rho)_s^2 + (Mc^2)^2)^{1/2}}$$

where the rigidity is updated in the following way. Let $(B\rho)_{s_o}$ be the initial rigidity (namely, $(B\rho)_{s_o} =$ *BORO* as defined in the keyword *OBJET* for instance). Then, at turn number N,

$$\text{if } 1 \le N \le 1200 \text{ then, } \textit{SCALE}(N) = 1 + \frac{1.22 - 1}{1 + 1200 - 1}(N-1)$$
$$\text{if } 1200 \le N \le 6379 \text{ then, } \textit{SCALE}(N) = 1.22 + \frac{1.33352 - 1.22}{1 + 6379 - 1200}(N-1200)$$

and then,

$$(B\rho)_s(N) = \textit{SCALE}(N) \cdot (B\rho)_{s_o}$$

from which value the calculations of $f_{RF}(N)$ follow.

NT can take negative values, then acting as an option switch (rather than giving number of timings), as follows :

• $NT = -1$: this is convenient for synchrotron acceleration. In this case the next two lines both contain a single data (as for $NT = 1$), respectively the starting scaling factor value, and 1. The current field scaling factor will then be updated from the energy kick by the cavity if for instance *CAVITE/IOPT=2* is used, namely,

$$\mathrm{SCALE}(N) = \mathrm{SCALE}(N-1) * \frac{B\rho(N)}{B\rho(N-1)}$$

• $NT = -2$: this is convenient for reading an RF law for *CAVITE* from an external data file, including usage for acceleration in fixed field accelerators.

• $NT = 1.10$: allows taking the scaling law from an external data file, as in the following example :

MULTIPOL COH1	
1.10	
./Csnk3D/bump_centered.scal	File name
1 2	Column numbers in the file : col. 2 gives the scaling factor at rigidity given by col. 1.

Notes :

1. In causing, via *CAVITE*, a change of the synchronous rigidity, *SCALING* causes a change of the reference rigidity, following (see *CAVITE*)

$$B\rho_{ref} = \textit{BORO} \longrightarrow B\rho_{ref} = \textit{BORO} + \delta B\rho_s$$

2. It may happen that some optical elements won't scale, for source code development reasons. This should be paid attention to.

SCALING	**Power supplies and R.F. function generator**		
IOPT, NFAM	*IOPT* = 0 (inactive) or 1 (active) ; *NFAM* = number of families to be scaled	0-1 ; 1-9	2*I
For NF=1, NFAM :	repeat *NFAM* times the following sequence :		
NAMEF [, Lbl [, Lbl]]	Name of family (*i.e.*, keyword of concern) [, up to 2 labels]		A10 [,A10[,A10]]
NT	$NT > 0$: number of timings ; $NT = -1$: field scaling factor updated by *CAVITE* ; $NT = -2$: RF law in *CAVITE* is read from external data file.	-2, -1 or 1-10	I
$SCL(I), I = 1, NT$	Scaling values (a single one, normally 1, if $NT = -1$).	relative	NT*E
$TIM(I), I = 1, NT$	Corresponding timings, in units of turns (1 if $NT = -1$).	turn number	NT*I

SEPARA : Wien Filter - analytical simulation

Note : simulation by stepwise integration can be found in *WIENFILTER*.

SEPARA provides an analytic simulation of an electrostatic separator. Input data are the length L of the element, the electric field E and the magnetic field B. The mass m and charge q of the particles are entered by means of the keyword *PARTICUL*.

The subroutines involved in *SEPARA* solve the following system of three equations with three unknown variables S, Y, Z (while $X \equiv L$), that describe the cycloidal motion of a particle in $\vec{E}$, $\vec{B}$ static fields (Fig. 1).

$$
\begin{aligned}
X &= -R\cos\left(\frac{\omega S}{\beta c} + \epsilon\right) - \frac{\alpha S}{\omega \beta c} + \frac{C_1}{\omega} \\
Y &= R\sin\left(\frac{\omega S}{\beta c} + \epsilon\right) - \frac{\alpha}{\omega^2} - \frac{C_2}{\omega} + Y_0 \\
Z &= S\sin(P_0) + Z_0
\end{aligned}
$$

where, S is the path length in the separator, $\alpha = -\frac{Ec^2}{\gamma}$, $\omega = -\frac{Bc^2}{m\gamma}$, $C_1 = \beta\sin(T_0)\cos(P_0)$ and $C_2 = \beta c\cos(T_0)\cos(P_0)$ are initial conditions. c = velocity of light, βc = velocity of the particle, $\gamma = (1-\beta^2)^{-\frac{1}{2}}$ and $\tan\epsilon = (C_2 + \frac{\alpha}{\omega})/C_1$. Y_0, T_0, Z_0, P_0 are the initial coordinates of the particle in the **zgoubi** reference frame. Here βc and γ are assumed constant, which is true as long as the change of momentum due to the electric field remains negligible all along the separator.

The option index IA in the input data allows switching to inactive element (thus equivalent to *ESL*), horizontal or vertical separator. Normally, E, B and the value of β_W for wanted particles are related by

$$
B(T) = -\frac{E\left(\frac{V}{m}\right)}{\beta_W \cdot c\left(\frac{m}{s}\right)}
$$

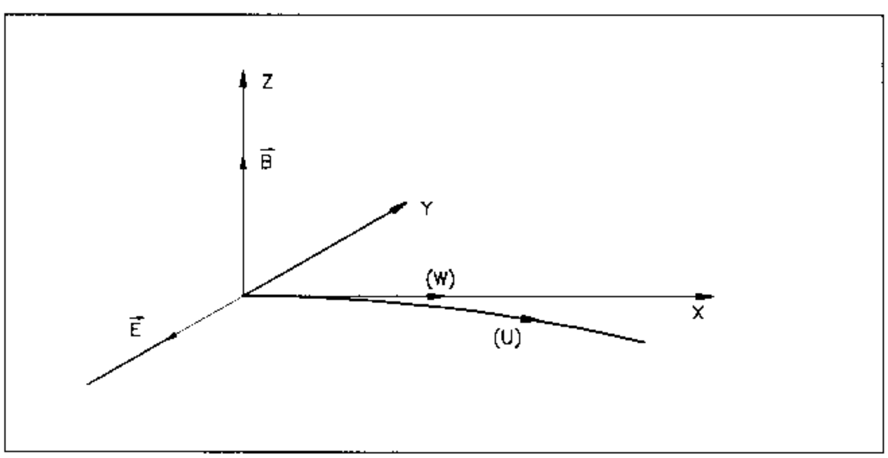
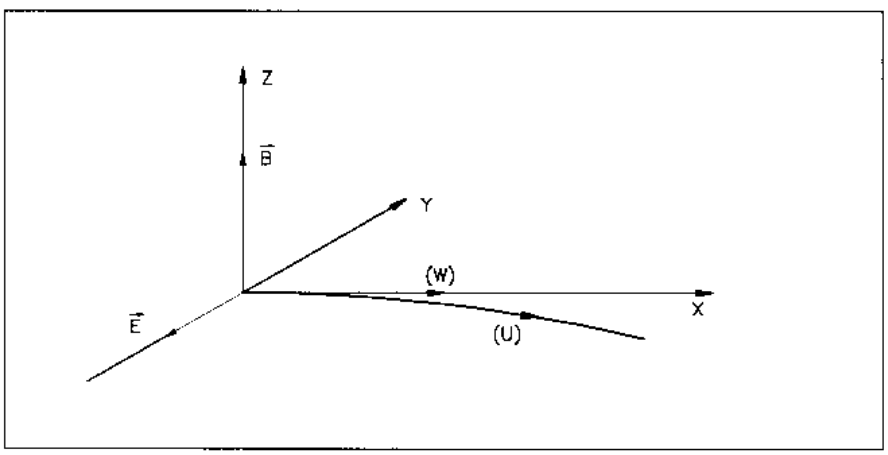


Figure 1: Horizontal separation between a wanted particle, (W), and an unwanted particle, (U). (W) undergoes a linear motion while (U) undergoes a cycloidal motion.

SEPARA [1] **Wien Filter - analytical simulation**

IA, *XL*, E, B,	$IA = 0$: element inactive $IA = 1$: horizontal separation $IA = 2$: vertical separation ; Length of the separator ; electric field ; magnetic field.	0-2, m, V/m, T	I, 3*E



Horizontal separation between a wanted particle, (W), and an unwanted particle, (U). (W) undergoes a linear motion while (U) undergoes a cycloidal motion.

[1] *SEPARA* must be preceded by *PARTICUL* for the definition of mass and charge of the particles.

SEXQUAD **Sharp edge magnetic multipole**

$B_Z\mid_{Z=0}= B_0\left(\frac{N}{R_0}Y+\frac{B}{R_0^2}Y^2+\frac{G}{R_0^3}Y^3\right)$

IL	$IL=1,2[\times10^n]$: print field and coordinates along trajectories.	0-2$[\times10^n]$	I
XL, R_0, B_0	Length of the element ; normalization distance ; field	2*cm, kG	3*E
N, $EB1$, $EB2$, $EG1$, $EG2$	Coefficients for the calculation of B. if $Y>0$: $B=EB1$ and $G=EG1$; if $Y<0$: $B=EB2$ and $G=EG2$.	5*no dim.	5*E
XPAS	Integration step	cm	E
KPOS, *XCE*, *YCE*, *ALE*	*KPOS*=1 : element aligned, 2 : misaligned ; shifts, tilt (unused if *KPOS*=1)	1-2, 2*cm, rad	I, 3*E

SEXTUPOL : Sextupole magnet (Fig. 1)

The meaning of parameters for *SEXTUPOL* is the same as for *QUADRUPO*.

In fringe field regions the magnetic field $\vec{B}(X,Y,Z)$ and its derivatives up to fourth order are derived from the scalar potential approximated to 7th order in Y and Z

$$V(X,Y,Z) = \left(G - \frac{G''}{16}(Y^2+Z^2) + \frac{G''''}{640}(Y^2+Z^2)^2\right)\left(Y^2 Z - \frac{Z^3}{3}\right)$$
$$\text{with } G_0 = \frac{B_0}{R_0^2}$$

The modelling of the fringe field form factor $G(X)$ is described under *QUADRUPO*, p. **??**.

Outside fringe field regions, or everywhere in sharp edge sextupole ($\lambda_E = \lambda_S = 0$), $\vec{B}(X,Y,Z)$ in the magnet is given by

$$B_X = 0$$
$$B_Y = 2G_0 YZ$$
$$B_Z = G_0(Y^2 - Z^2)$$

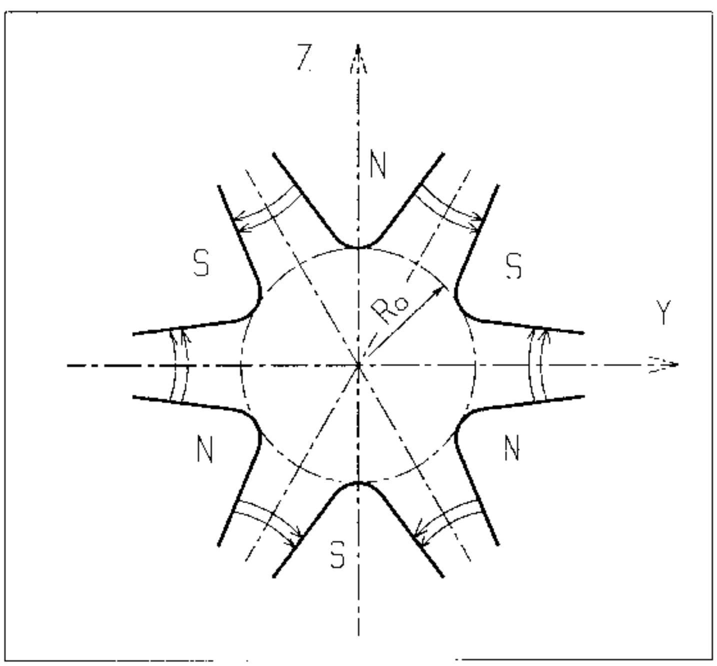
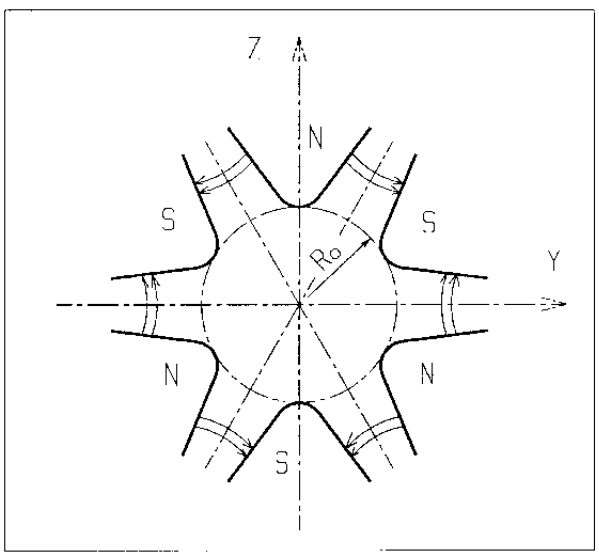


Figure 1: Sextupole magnet

SEXTUPOL	**Sextupole Magnet**		
IL	$IL = 1, 2[\times 10^n]$: print field and coordinates along trajectories.	0-2$[\times 10^n]$	I
XL, R_0,B_0	Length ; radius and field at pole tip of the element	2*cm, kG	3*E
	Entrance face :		
X_E, λ_E	Integration zone ; fringe field extent ($\lambda_E = 0$ for sharp edge)	2*cm	2*E
NCE, $C_0 - C_5$	*NCE* = unused $C_0 - C_5$ = Fringe field coefficients such that $G(s) = G_0/(1 + \exp P(s))$, with $G_0 = B_0/R_0^2$ and $P(s) = \sum_{i=0}^{5} C_i(s/\lambda)^i$	any, 6* no dim.	I, 6*E
	Exit face :		
X_S, λ_S	Parameters for the exit fringe field ; see entrance	2*cm	2*E
NCS, $C_0 - C_5$		0-6, 6*no dim.	I, 6*E
XPAS	Integration step	cm	E
KPOS, *XCE*, *YCE, ALE*	*KPOS*=1 : element aligned, 2 : misaligned ; shifts, tilt (unused if *KPOS*=1)	1-2, 2*cm, rad	I, 3*E



Sextupole magnet

SOLENOID : Solenoid (Fig. 1)

The solenoidal magnet has an effective length *XL*, a mean radius R_0 and an asymptotic field $B_0 = \mu_0 NI/XL$ (*i.e.*, $\int_{-\infty}^{\infty} B_X(X,r)dX = \mu_0 NI,\ \forall r < R_0$), wherein B_X=longitudinal field component, NI = number of Ampere-Turns, $\mu_0 = 4\pi 10^{-7}$.

The distance of ray-tracing beyond the effective length *XL*, is X_E at the entrance, and X_S at the exit (Fig. 1).

Two methods are available for the computation of the field $\vec{B}(X,r)$ and its derivatives.
Method 1 : $\vec{B}(X,r)$, $r = (Y^2+Z^2)^{1/2}$ and its derivatives up to second order at all (X,Y,Z) are calculated following Ref. [**?**], based on the three complete elliptic integrals K, E and Π. The latter are calculated with the algorithm proposed in the same reference, their derivatives are calculated by means of recursive relations [**?**].

This analytical model for the solenoidal field allows simulating an extended range of coil geometries (length, radius) provided that the coil thickness is small enough compared to the mean radius R_0.

In particular the field on-axis writes (taking $X = r = 0$ at the center of the solenoid)

$$B_X(X,r=0) = \frac{\mu_0 NI}{2XL}\left[\frac{XL/2 - X}{\sqrt{(XL/2-X)^2 + R_0^2}} + \frac{XL/2 + X}{\sqrt{(XL/2+X)^2 + R_0^2}}\right]$$

and yields the magnetic length

$$L_{mag} \equiv \frac{\int_{-\infty}^{\infty} B_X(X, r<R_0)dX}{B_X(X=r=0)} = XL\sqrt{1+\frac{4R_0^2}{XL^2}} > XL \tag{1}$$

with in addition

$$B_X(\text{center}) \equiv B_X(X=r=0) = \frac{\mu_0 NI}{XL\sqrt{1+\frac{4R_0^2}{XL^2}}}.$$

Method 2 : The second method available uses eq. 1 above as a 1-D model and uses off-axis extrapolation to derive the field and its derivatives at all (X,Y,Z), following the method described in section **??**.

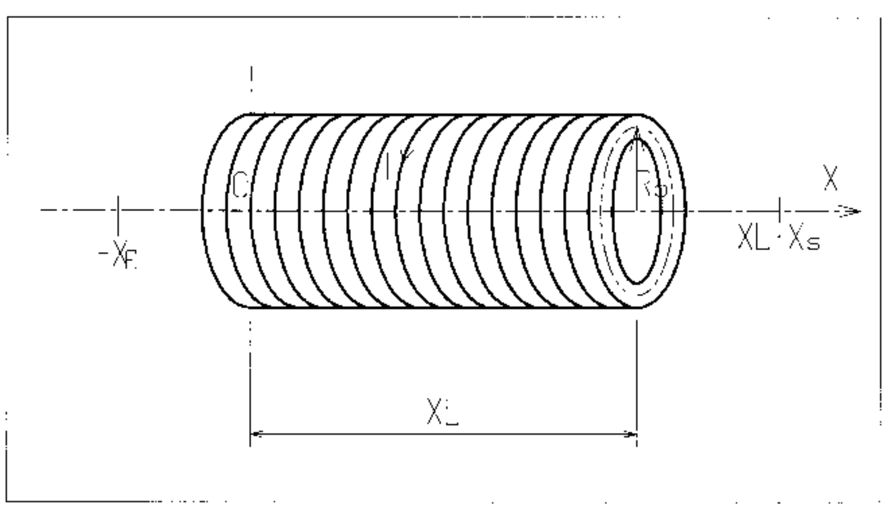
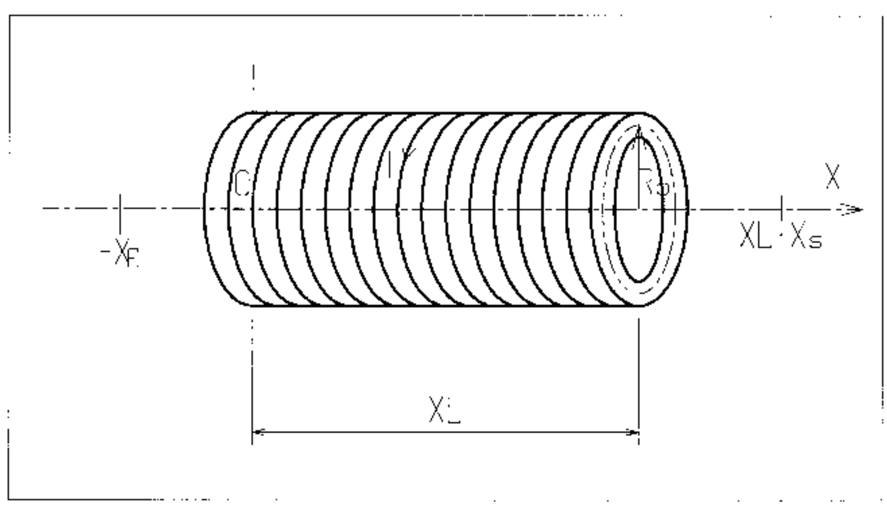


Figure 1: Solenoidal magnet.

SOLENOID **SOLENOIDTitl**

IL	$IL = 1, 2[\times 10^n]$: print field and coordinates along trajectories.	0-2$[\times 10^n]$	I
XL, R_0, B_0	Length ; radius ; asymptotic field (=$\mu_0 NI/XL$)	2*cm, kG	3*E
X_E, X_S	Entrance and exit integration zones	2*cm	2*E
XPAS	Integration step	cm	E
KPOS, *XCE*, *YCE*, *ALE*	*KPOS*=1 : element aligned, 2 : misaligned ; shifts, tilt (unused if *KPOS*=1)	1-2, 2*cm, rad	I, 3*E



SPINR : Spin rotation

Spin precession, a local transformation.

The precession is defined by its axis and its value.

SPINR	**SPINRTitl**		
Θ_x, Φ_X	Angles that define the spin precession axis.	2*rad	2*E
μ	Spin precession angle.	rad	E

SPNPRNL	**Store spin coordinates in file *FNAME***	
FNAME [1]	Name of storage file (*e.g.*, zgoubi.spn)	A80

SPNSTORE	**Store spin coordinates every IP other pass**	
FNAME [1] [,*LABEL(s)*] [2]	Name of storage file (*e.g.*, zgoubi.spn) [; label(s) of the element(s) at the exit of which the store occurs (10 labels maximum)].	A80 [, 10*A10]
IP	Store every IP other pass (when using *REBELOTE* with *NPASS* $\geq IP - 1$).	I

SPNPRT	**Print spin coordinates**
	Print spin coordinates into zgoubi.res, at the location where this keyword is introduced in the structure.

[1] *FNAME* = 'none' will inhibit printing.
[2] If first *LABEL* = 'none' then printing will be inhibited.

SPNTRK : Spin tracking

The keyword *SPNTRK* allows switching spin tracking on (index *KSO*=1) or off (*KSO*=0), or resuming (index *KSO*=-1, following an occurence *KSO*=0). It also permits the attribution of an initial spin to each one of the *IMAX* particles of the beam, following a distribution that depends on the option index *KSO*. It must be preceded by *PARTICUL* for the definition of mass and gyromagnetic factor.

KSO = 1 (respectively 2, 3) : the *IMAX* particles defined with *[MC]OBJET* are given a longitudinal (1,0,0) spin component (respectively transverse horizontal (0,1,0), vertical (0,0,1)).

KSO = 4 : initial spin components are entered explicitly for each one of the *IMAX* particles of the beam.

KSO = 4.1 : three initial spin components $S_X,\ S_Y,\ S_Z$ are entered explicitly just once, they are then assigned to each one of the *IMAX* particles of the beam.

KSO = 5 : random generation of *IMAX* initial spin conditions as described in Fig. 1. Given a mean polarization axis (S) defined by its angles T_0 and P_0, and a cone of angle A with respect to this axis, the *IMAX* spins are sorted randomly in a Gaussian distribution

$$p(a) = \exp\left[-\frac{(A-a)^2}{2\delta A^2}\right] / \delta A\,\sqrt{2\pi}$$

and within a cylindrical uniform distribution around the (S) axis. Examples of simple distributions available by this mean are given in Fig. 2.

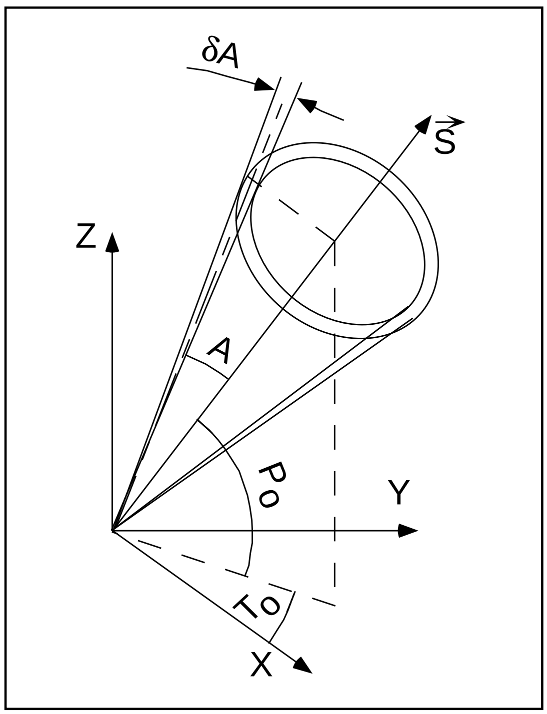


Figure 1: Spin distribution as obtained with option *KSO* = 5.
The spins are distributed within an annular strip δA (standard deviation) at an angle A with respect to the axis of mean polarization (S) defined by T_0 and P_0.

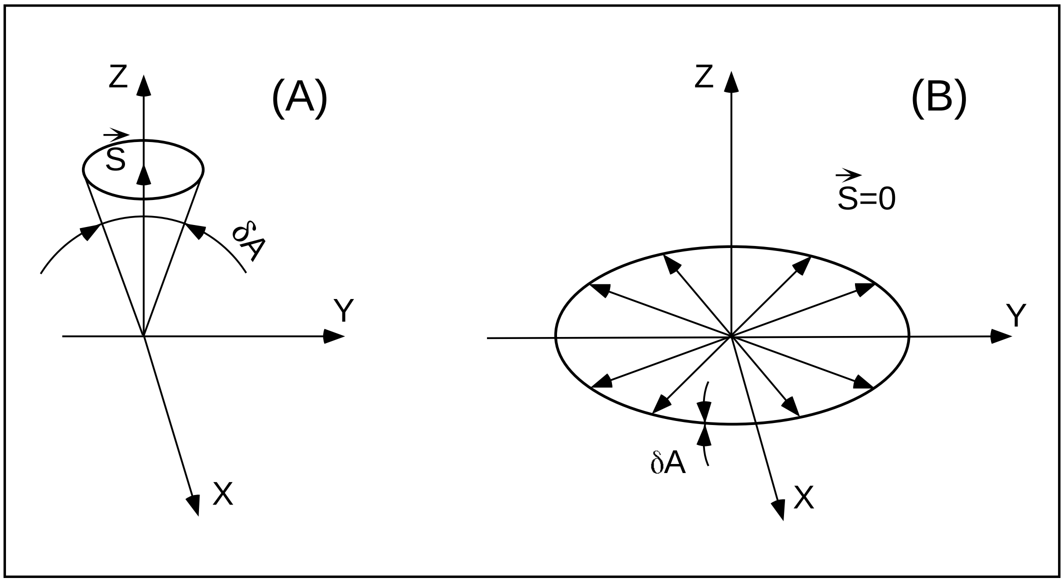


Figure 2: Examples of the use of *KSO* =5.

A : Gaussian distribution around a mean vertical polarization axis, obtained with T_0 = arbitrary, $P_0 = \pi/2$, $A = 0$ and $\delta A \neq 0$.

B : Isotropic distribution in the median plane, obtained with $P_0 = \pm\pi/2$, $A = \pi/2$, and $\delta A = 0$.

SPNTRK [1]	**Spin tracking**		
KSO	*KSO=0* : spin tracking [switched] off ; *KSO=-1* : spin tracking resumes. Otherwise : as stated below.	-1 or 0 or 1-3 0r 4[.1] or 5	I
If KSO = 1 – 3	*KSO* = 1 (respectively 2, 3) : all particles have their spin automatically set to (1,0,0), longitudinal [respectively (0,1,0), horizontal and (0,0,1), vertical]		
If KSO = 4	Repeat *IMAX* times (corresponding to the *IMAX* particles in '*OBJET*') the following sequence :		
S_X, S_Y, S_Z	X, Y and Z initial components of the initial spin.	3*no dim.	3*E
If KSO = 4.1			
S_X, S_Y, S_Z	X, Y and Z components of the initial spins. These will be assigned to all particles.	3*no dim.	3*E
If KSO = 5	Random distribution in a cone (see figure) Enter the following two sequences :		
TO, PO, A, δA	Angles of average polarization : A = angle of the cone ; δA = standard deviation of distribution around A	4*rad	4*E
IR	Random sequence seed	$\lesssim 10^6$	I

[1] *SPNTRK* must be preceded by *PARTICUL* for the definition of G and mass.

SRLOSS : Synchrotron radiation loss [?]

The keyword *SRLOSS* allows activating or stopping (option $KSR = 1, 0$ respectively) stepwise tracking of energy loss by stochastic emission of photons in magnetic fields, following the method described in section **??**.

It can be chosen to allow radiation in the sole dipole fields, or in all types of fields regardless of their multipole composition. It can also be chosen to allow for the radiation induced transverse kick.

SRLOSS must be preceded by *PARTICUL* for defining mass and charge values as they enter in the definition of SR parameters.

Statistics on SR parameters are computed and updated while tracking, the results of which can be obtained by means of the keyword *SRPRNT*.

SRLOSS	**Synchrotron radiation loss**		
KSR[.i]	Switch ; $i = 1$ causes info output into `zgoubi.SRLOSS.out`	$0-1$	2*I
STR1, *STR2*	Options : *STR1* = *'ALL'* or *'all'* or a particular KEYWORD ; *STR2* = *'scale'* will scale fields with energy loss.	2*A	
Option, seed	Option : 0 / inhibited, 1 / photon entails dp only, 2 / photon entails dp and angle kick.	$1-3, > 10^5$	I

SRPRNT : Print SR loss statistics

SRPRNT may be introduced anywhere in a structure. It allows switching on synchrotron radiation loss computation. It produces in addition a print out (to zgoubi.res) of current state of statistics on several parameters related to SR loss presumably activated beforehand with keyword *SRLOSS*.

SRPRNT **Print SR loss statistics** into zgoubi.res

SYNRAD : Synchrotron radiation spectral-angular densities

The keyword *SYNRAD* enables (or disables) the calculation of synchrotron radiation (SR) electric field and spectral angular energy density. It must be preceded by *PARTICUL* for defining mass and charge values, as they enter in the definition of SR parameters.

SYNRAD is supposed to appear a first time at the location where SR should start being taken into account, with the first data *KSR* set to 1. It results in on-line storage of the electric field vector and other relevant quantities in zgoubi.sre, as step by step integration proceeds. The observer position (XO, YO, ZO) is specified next to *KSR*.

Data stored in zgoubi.sre :

(ELx, ELy, ELz) : electric field vector $\vec{\mathcal{E}}$ (eq. **??**)
$(btx, bty, btz) = \vec{\beta} = \frac{1}{c} \times$ particle velocity
$(gx, gy, gz) = \frac{d\vec{\beta}}{dt} =$ particle acceleration (eq. **??**)
$\Delta\tau =$ observer time increment (eq. **??**)
$t' = \tau - r(t')/c =$ retarded (particle) time
(rtx, rty, rtz) : $\vec{R}(t)$, particle to observer vector (eq. **??**)
$(x, y, z) =$ particle coordinates
$\Delta s =$ step size in the magnet (fig. **??**)
$NS =$ step number
$I =$ particle number
$LET(I) =$ tagging letter
IEX$(I) =$ stop flag (see section **??**)

SYNRAD is supposed to appear a second time at the location where SR calculations should stop, with *KSR* set to 2. It results in the output of the angular energy density $\int_{\nu_1}^{\nu_2} \partial^3 W/\partial\phi\,\partial\psi\,\partial\nu$ (eq. **??**) as calculated from the Fourier transform of the electric field (eq. **??**). The spectral range of interest and frequency sampling (ν_1, ν_2, N) are specified next to *KSR*.

SYNRAD	**Synchrotron radiation spectral-angular densities**		
KSR	Switch 0 : inhibit SR calculations 1 : start 2 : stop	0-2	I
If KSR = 0			
$D1$, $D2$, $D3$	Dummies		3*E
If KSR = 1			
$X0$, $Y0$, $Z0$	Observer position in frame of magnet next to *SYNRAD*	3*m	3*E
If KSR = 2			
ν_1, ν_2, N	Frequency range and sampling	2*eV, no dim.	2*E, I

SYSTEM : System call

The keyword *SYSTEM* allows one or a series of system calls. It can appear anywhere, an arbitrary number of times, in the zgoubi.dat data list. It is effective at the very location where it appears.

SYSTEM keyword is followed by the list of the desired system commands. That can be saving zgoubi output files, calling again **zgoubi** at the end of a run so allowing dependent consecutive jobs, etc.

0.1 Optical Elements and Related Numerical Procedures

SYSTEM	**System call**		
NCMD	The number of calls to follow.	≤ 0	I

NCMD lines follow, one command per line.

TOSCA : 2-D and 3-D Cartesian or cylindrical mesh field map

TOSCA is dedicated to the reading and treatment of 2-D or 3-D Cartesian or cylindrical mesh field maps as delivered by the *TOSCA* magnet computer code standard output.

A pair of flags, *MOD*, *MOD2*, determine whether Cartesian or Z-axis cylindrical mesh is used, and the nature of the field map data set.

The total number of field data files to be read is determined by the *MOD* flag (see below) and by the parameter IZ that appears in the data list following the keyword. Each of these files contains the field components B_X, B_Y, B_Z on an (X, Y) mesh. $IZ = 1$ for a 2-D map, and in this case B_X and B_Y are assumed zero all over the map[1]. For a 3-D map with mid-plane symmetry, described with a set of 2-D maps at various Z, then *MOD=0* and $IZ \geq 2$, and thus, the first data file whose name follows in the data list is supposed to contain the median plane field (assuming $Z = 0$ and $B_X = B_Y = 0$), while the remaining $IZ - 1$ file(s) contain the $IZ - 1$ additional planes in increasing Z order. For arbitrary 3-D maps, no symmetry assumed, then *MOD=1* and the total number of maps (whose names follow in the data list) is IZ, such that map number $[IZ/2] + 1$ is the $Z = 0$ elevation one.

The field map data file has to be be filled with a format that fits the *FORTRAN* reading sequence. The following is an instance, details and possible updates are to be found in the source file `'fmapw.f'` :

```
DO  1  K = 1, KZ
OPEN (UNIT = NL, FILE = FNAME, STATUS = 'OLD' [,FORM='UNFORMATTED'])
DO  1  J = 1,  JY
DO  1 I = 1, IX
IF (BINARY) THEN
READ(NL) Y(J), Z(K), X(I), BY(J,K,I), BZ(J,K,I), BX(J,K,I)
ELSE
READ(NL,100) Y(J), Z(K), X(I), BY(J,K,I), BZ(J,K,I), BX(J,K,I)
100            FORMAT(1X,6E11.2)
ENDIF
1     CONTINUE
```

IX (JY, KZ) is the number of longitudinal (transverse horizontal, vertical) nodes of the 3-D uniform mesh. For letting **zgoubi** know that these are binary files, FNAME must begin with 'B_' or 'b_'.

In addition to the *MOD=1*, *2* cases above, one can have *MOD=12* and in that case a single file contains the all 3-D field map. See table below and the *FORTRAN* subroutine `fmapw.f` and its entries FMAPR, FMAPR2, for more details, in particular the formatting of the field map data file(s).

[1] Use *MAP2D* in case non-zero B_X, B_Y are to be taken into account in a 2-D map.

MOD	**MOD2**	
MOD $\leq$ 19 : Cartesian mesh		
0 and $IZ = 1$	none	2-D map, a single data file for $B_Z(X, Y)\|_{Z=0}$, mid-plane symmetry.
0 and $IZ > 1$	none	3-D map, 1+IZ/2 data files of upper half of magnet, one per (X,Y)$\|_{0\leq Z\leq Z_{max}}$ plane, mid-plane symmetry.
0	1, 2, 3	As previous case, just different reading formats.
1	none	2- or 3-D map, IZ data files, one per (X,Y) plane, no symmetry assumed.
1	1, 2, 3	As previous case, just different reading formats.
3	none, 1	AGS main magnet field map, 2-D, mid-plane symmetry assumed. MOD2=1 causes field to be perturbed by $(1 + n_1\, dY + n_2\, dY^2 + n_3\, dY^3)$ factor.
12	none	3-D map, single file, upper half of magnet, symmetry with respect to (X,Y) mid-plane.
12	1	3-D map, single file, whole magnet volume (thus no symmetry assumed).
12	2	3-D map, single file, 1/8th of the magnet, symmetry wrt. (X,Y), (X,Z), (Y,Z) planes.
15	1-6	3-D map, whole magnet volume (thus no symmetry assumed), up to 6 maps summed up : at all node, $\vec{B} = \sum_{i=1}^{i=MOD2} a_i\, \vec{B}_i$.
MOD $\geq$ 20 : Cylindrical mesh		
20, 21		3-D map, single file, half a magnet, cyl. symmetry with respect to (Y,Z) plane.
22, 24		3-D map, single file, half a magnet, symmetry with respect to (X,Y) mid-plane.

The field $\vec{B} = (B_X, B_Y, B_Z)$ is normalized by means of *BNORM* in a similar way as in *CARTEMES*. As well the coordinates X and Y (and Z in the case of a 3-D field map) are normalized by the *X-[, Y-, Z-]NORM* coefficient (useful to convert to centimeters, the working units in **zgoubi**).

At each step of the trajectory of a particle inside the map, the field and its derivatives are calculated

- in the case of 2-D map, by means of a second or fourth order polynomial interpolation, depending on *IORDRE* (*IORDRE* = 2, 25 or 4), as for *CARTEMES*,
- in the case of 3-D map, by means of a second order polynomial interpolation with a $3 \times 3 \times 3$-point parallelepipedic grid, as described in section **??**.

Entrance and/or exit integration boundaries between which the trajectories are integrated in the field may be defined, in the same way as in *CARTEMES*.

TOSCA	**2-D and 3-D Cartesian or cylindrical mesh field map**		
IC, IL	see *CARTEMES*	0-2, 0-2	2*I
BNORM, XN,YN, ZN	Field and X-,Y-,Z-coordinate normalization coefficients	4*no dim.	4*E
TITL	Title. Include "FLIP" to get field map X-flipped. Include "HEADER n" in case *FNAME* starts with $n \geq 1$ header lines.		A80
IX, IY, IZ, *MOD[.MOD2]*	Number of nodes of the mesh in the X, Y and Z directions, $IZ = 1$ for single 2-D map ; *MOD* : operational and map *FORMAT* reading mode [2]; *MOD*≤19 : Cartesian mesh ; *MOD*≥20 : cylindrical mesh. *MOD2*, optional, tells the reading *FORMAT*, default is '*'.	≤*MXX*[1], ≤*MXY*, ≤*IZ*, ≥ 0[.1-9]	3*I
FNAME [1] ($K = 1, NF$)	Names of the NF files that contain the 2-D maps, from $Z(1)$ to $Z(NF)$. If *MOD*=0 : $NF = 1 + [IZ/2]$, the NF 2-D maps are for $0 \leq Z \leq Z_{max}$, they are symmetrized with respect to the $Z(1) = 0$ plane. If *MOD*=1 : $NF = IZ$, no symmetry assumed ; $Z(1) = Z_{max}$, $Z(1 + [IZ/2]) = 0$ and $Z(NF) = -Z_{max}$. If *MOD*=12 : a single *FNAME* file contains the all 3-D volume. If *MOD*=20-22 : other symmetry options, see `toscap.f` routine...		A80
ID, A, B, C [, A', B', C', A'', etc., if $ID \geq 2$]	Integration boundary. Ineffective when $ID = 0$. ID = -1, 1 or ≥ 2 : as for *CARTEMES*	≥ -1, cm, 2*no dim. [,*idem*]	I,3*E [,3*E,etc.]
IORDRE	If $IZ = 1$: 3, 4, 25, as in *CARTEMES* ; unused if $IZ \neq 1$.	2, 25 or 4	I
XPAS	Integration step	cm	E
If Cartesian mesh (see MOD) :			
KPOS, XCE, YCE, ALE	*KPOS*=1 : element aligned, 2 : misaligned ; shifts, tilt	1-2, 2*cm, rad	I, 3*E
If polar mesh :			
KPOS	as for *POLARMES*. Normally 2.	1-2	I
If KPOS = 2			
RE, TE, RS, TS		cm, rad, cm, rad	4*E

[1] *MXX, MXY, IZ* may be changed, they are stated in the include file `PARIZ.H`.

[2] Each file *FNAME(K)* contains the field specific to elevation $Z(K)$ and must be formatted according to the following *FORTRAN* read sequence (that usually fits *TOSCA* code *OUTPUTS* - details and possible updates are to be found in the source file `'fmapw.f'`) :

```
DO K = 1, NF
OPEN (UNIT = NL, FILE = FNAME(K), STATUS = 'OLD' [,FORM='UNFORMATTED'])
DO J = 1, JY ;     DO I = 1, IX
IF (BINARY) THEN
READ(NL) Y(J), Z(K), X(I), BY(J,K,I), BZ(J,K,I), BX(J,K,I)        node coordinates, field components at node
ELSE
READ(NL,*) Y(J), Z(K), X(I), BY(J,K,I), BZ(J,K,I), BX(J,K,I)      node coordinates, field components at node
ENDIF
ENDDO ;   ENDDO
NL = NL + 1
ENDDO
```

For 2-D maps BX and BY are assumed zero at all nodes of the 2-D mesh, regardless of BX(J,1,I), BY(J,1,I) values. For binary files, *FNAME* must begin with 'B_' or 'b_'.

TRANSMAT : Matrix transfer

TRANSMAT performs a second order transport of the particle coordinates in the following way

$$X_i = \sum_j R_{ij} X_j^0 + \sum_{j,k} T_{ijk} X_j^0 X_k^0$$

where, X_i stands for any of the current coordinates Y, T, Z, P, path length and momentum dispersion, and X_i^0 stands for any of the initial coordinates. $[R_{ij}]$ ($[T_{ijk}]$) is the first order (second order) transfer matrix as usually involved in second order beam optics [**?**]. Second order transfer is optional. The length of the element represented by the matrix may be introduced for the purpose of path length updating.

Note : *MATRIX* delivers $[R_{ij}]$ and $[T_{ijk}]$ matrices in a format suitable for straightforward use with *TRANSMAT*.

TRANSMAT **Matrix transfer**

IORDRE	Transfer matrix order	1-2	I
XL	Length (ineffective, for updating)	m	E
For $IA = 1, 6$:			
$R(IA, IB)$, $IB = 1,\ 6$	First order matrix	m, rad	6 lines 6*E each
If IORDRE = 2	Following records *only* if *IORDRE* = 2		
$T(IA, IB, IC)$,	Second order matrix, six 6*6 blocks	m, rad	36 lines 6*E each

TRAROT : Translation-Rotation of the reference frame

UNDER DEVELOPMENT. Check before use.

This procedure transports particles into a new frame by translation and rotation. Effect on spin tracking, particle decay and gas-scattering are taken into account (but not on synchrotron radiation).

TRAROT	**Translation-Rotation**		
$TX, TY, TZ,$ RX, RY, RZ	Translations, rotations	3*m, 3*rad	6*E

TWISS : Calculation of periodic optical parameters

TWISS causes the calculation of transport coefficients and various other global parameters, in particular periodical quantities as tunes and optical functions, in the coupled hypothesis. *TWISS* is normally placed at the end of the structure ; it causes a series of up to 5 successive passes in the structure (at the manner of *REBELOTE*).

The object necessary for these calculations will be generated automatically if one uses *OBJET* with option *KOBJ*= 5.

TWISS works in a way similar to *MATRIX*, iterating the *MATRIX* process wherever necessary, changing for instance the reference trajectory in *OBJET* for dp/p related computations. In particular :

- It assumes that the reference particle (particle #1 of 11, when using *OBJET[KOBJ= 5]*) is located on the closed orbit. *This condition has to be satisfied for TWISS to work consistently.*
- A first pass (the only one if *KTW=1*) through the structure allows computing the periodic optical functions from the rays.
- The periodic dispersions are used to define chromatic closed orbits at $\pm\delta p/p$. A second and a third pass (which terminate the process if *KTW=2*) with chromatic objects centered respectively on $\pm\delta p/p$ chromatic orbits will then compute the chromatic first order transport matrices. From these the chromaticities are deduced.
- Anharmonicities need two additional passes (which terminate the process if *KTW=3*). They are deduced from the difference in tunes for particles tracked on different transverse invariants, horizontal or vertical.

0.1 Complements Regarding Various Functionalities

TWISS	Calculation of periodic optical parameters		
KTW[.KTW2], FacD, FacA	$KTW = 0/1/2/3$: Off / as *MATRIX* / add computation of chromaticities / add computation of anharmonicities. $KTW2 = 1$: long write-up to zgoubi.res. $FacD \times D = \delta p/p$ value applied, with D the momentum sampling in OBJET ; $FacA$: unused.	0-3[.1], 2*any	I,2*E

Example

```
'OBJET'
20015.55                          ! 6 GeV electrons.
5                                 ! Will generate 11 particles.
.001 .001 .001 .001 0. .0001     ! Coordinate sampling for matrix computation : $delta_Y,
0. 0. 0. 0. 0. 1.                 ! delta_T, delta_Z, delta_P, delta_S (unused), delta_D$.
................................
zgoubi.dat optics list in between
................................
'TWISS'
2 1. 1.                          ! KTW = 3, FacD = 1
'END'
```

"*KTW=3*" under *TWISS* will cause 3 successive executions of zgoubi.dat and will result in delivery (print out to zgoubi.res) of

- the on-momentum matrix of the optical structure,
- off-momentum matrices at $\frac{dp}{p} = \pm FacD * \delta D$,
- the Twiss parameters in the hypothesis of a stable periodic structure,
- the momentum compaction, chromaticities, etc.

UNDULATOR : Undulator magnet

UNDULATOR magnet. UNDER DEVELOPMENT.

UNDULATOR **Undulator magnet**

Under development, to be documented

UNIPOT : Unipotential cylindrical electrostatic lens

The lens is cylindrically symmetric about the X-axis.

The length of the first (resp. second, third) electrode is $X1$ (resp. $X2$, $X3$). The distance between the electrodes is D. The potentials are $V1$ and $V2$. The inner radius is R_0 (Fig. 1). The model for the electrostatic potential along the axis is [**?**]

$$V(x) = \frac{V2 - V1}{2\omega D}\left[\ln \frac{\cosh \dfrac{\omega\left(x + \dfrac{X2}{2} + D\right)}{R_0}}{\cosh \dfrac{\omega\left(x + \dfrac{X2}{2}\right)}{R_0}} + \ln \frac{\cosh \dfrac{\omega\left(x - \dfrac{X2}{2} - D\right)}{R_0}}{\cosh \dfrac{\omega\left(x - \dfrac{X2}{2}\right)}{R_0}}\right]$$

($x =$ distance from the center of the central electrode ; ω = 1,318 ; cosh = hyperbolic cosine), from which the field $\vec{E}(X, Y, Z)$ and its derivatives are deduced following the procedure described in section **??**.

Use *PARTICUL* prior to *UNIPOT*, for the definition of particle mass and charge.

The total length of the lens is $X1 + X2 + X3 + 2\,D$; stepwise integration starts at entrance of the first electrode and terminates at exit of the third one.

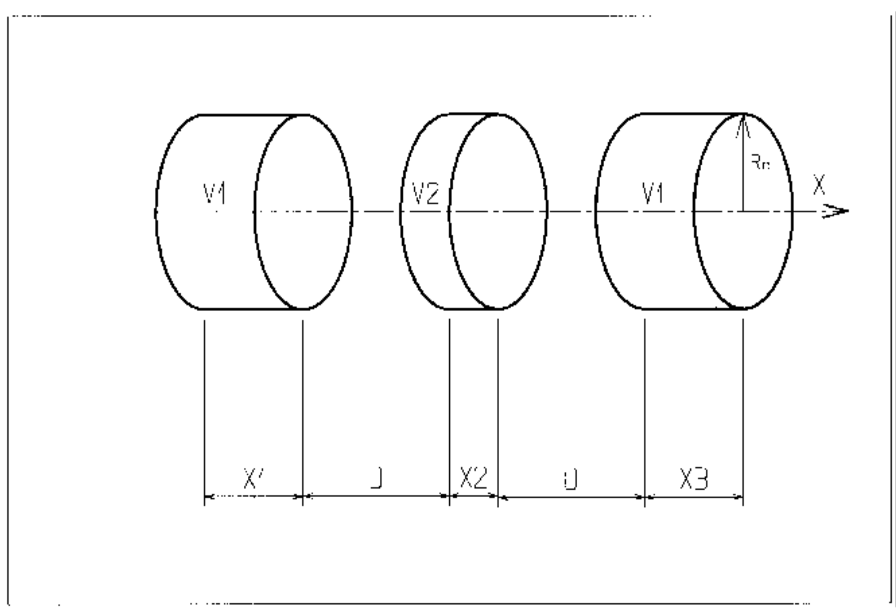
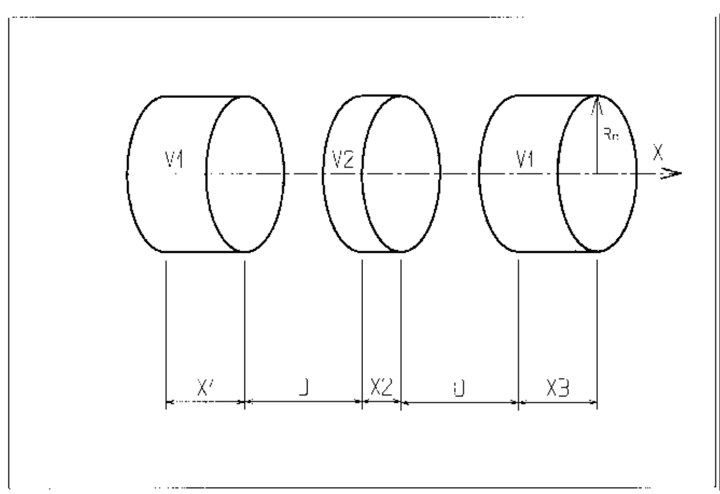


Figure 1: Three-electrode cylindrical unipotential lens.

UNIPOT **Unipotential electrostatic lens**

IL	$IL = 1, 2$: print field and coordinates along trajectories	0-2$[\times 10^n]$	
X_1, D, X_2, X_3, R_0	Length of first tube ; distance between tubes ; length of second and third tubes ; radius	5*m	5*E
V_1, V_2	Potentials	2*V	2*E
XPAS	Integration step	cm	E
KPOS, *XCE*, *YCE*, *ALE*	*KPOS*=1 : element aligned, 2 : misaligned ; shifts, tilt (unused if *KPOS*=1)	1-2, 2*cm, rad	I, 3*E



VENUS : Simulation of a rectangular shape dipole magnet

VENUS is dedicated to a 'rough' simulation of SATURNE Laboratory's *VENUS* dipole. The field B_0 is constant inside the magnet, with longitudinal extent *XL* and transverse extent $\pm YL$; outside these limits, $B_0 = 0$ (Fig. 1).

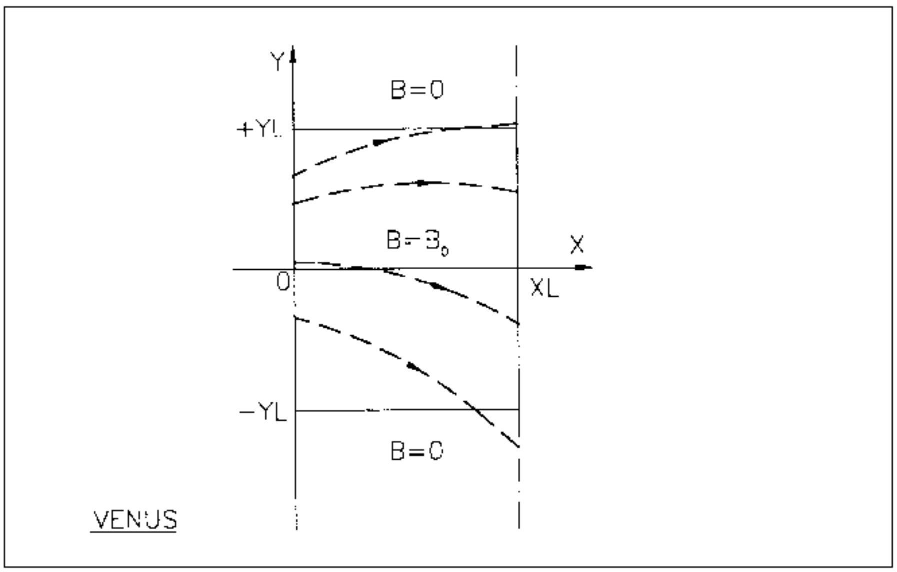
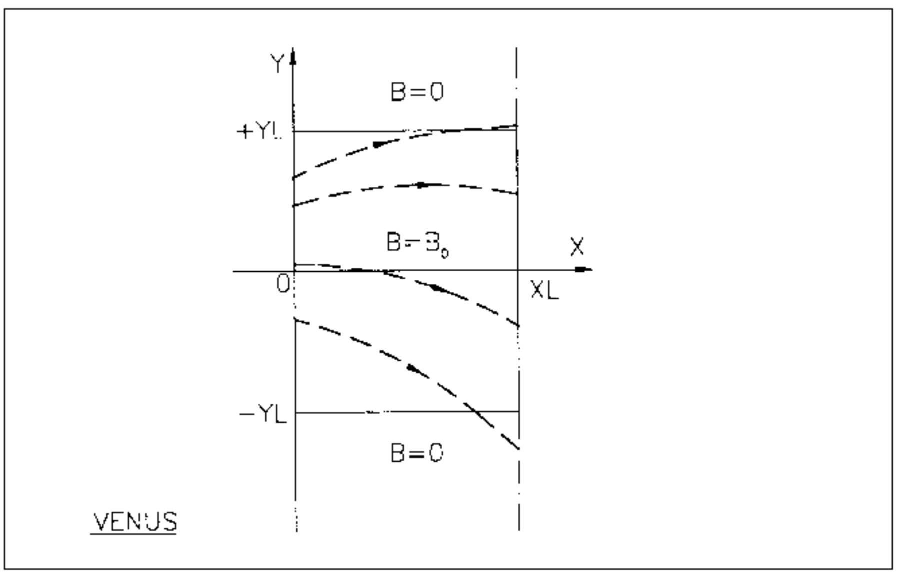


Figure 1: Scheme of *VENUS* rectangular dipole.

VENUS	**Simulation of a rectangular dipole magnet**		
IL	$IL = 1, 2[\times 10^n]$: print field and coordinates along trajectories.	0-2$[\times 10^n]$	I
XL, YL, B_0	Length ; width = $\pm YL$; field	2*cm, kG	3*E
XPAS	Integration step	cm	E
KPOS, *XCE*, *YCE*, *ALE*	*KPOS*=1 : element aligned, 2 : misaligned ; shifts, tilt (unused if *KPOS*=1)	1-2, 2*cm, rad	I, 3*E



Scheme of *VENUS* rectangular dipole.

WIENFILT : Wien filter

WIENFILT simulates a Wien filter, with transverse and orthogonal electric and magnetic fields $\vec{E}_Y$, $\vec{B}_Z$ or $\vec{E}_Z$, $\vec{B}_Y$ (Fig. **??**). It must be preceded by *PARTICUL* for the definition of particle mass and charge.

The length *XL* of the element is the distance between its entrance and exit EFB's. The electric and magnetic field intensities E_0 and B_0 in the central, uniform field region, normally satisfy the relation

$$B_0 = -\frac{E_0}{\beta_W c}$$

for the selection of " wanted" particles of velocity $\beta_W c$. Ray-tracing in field fall-off regions extends over a distance X_E (X_S) beyond the entrance (exit) EFB by means of prior and further automatic change of frame. Four sets of coefficients λ, $C_0 - C_5$ allow the description of the entrance and exit fringe fields outside the uniform field region, following the model [**?**]

$$F = \frac{1}{1 + \exp(P(s))}$$

where $P(s)$ is of the term

$$P(s) = C_0 + C_1 \left(\frac{s}{\lambda}\right) + C_2 \left(\frac{s}{\lambda}\right)^2 + C_3 \left(\frac{s}{\lambda}\right)^3 + C_4 \left(\frac{s}{\lambda}\right)^4 + C_5 \left(\frac{s}{\lambda}\right)^5$$

and s is the distance to the EFB. When fringe fields overlap inside the element (*i.e.*, *XL* $\leq X_E + X_S$), the field fall-off is expressed as

$$F = F_E + F_S - 1$$

where $F_E(F_S)$ is the value of the coefficient respective to the entrance (exit) EFB.
If $\lambda_E = 0$ $(\lambda_S = 0)$ for either the electric or magnetic component, then both are considered as sharp edge fields and $X_E(X_S)$ is forced to zero (for the purpose of saving computing time). In this case, the magnetic wedge angle vertical first order focusing effect is simulated at entrance and exit by a kick $P_2 = P_1 - Z_1 \tan(\epsilon/\rho)$ applied to each particle (P_1, P_2 are the vertical angles upstream and downstream the EFB, Z_1 the vertical particle position at the EFB, ρ the local horizontal bending radius and ϵ the wedge angle experienced by the particle ; ϵ depends on the horizontal angle T). This is not done for the electric field, however it is advised not to use a sharp edge electric dipole model since this entails non symplectic mapping, and in particular precludes accounting for momentum effects of the non zero longitudinal electric field component.

WIENFILT [1] **Wien filter**

IL	$IL = 1, 2[\times 10^n]$: print field and coordinates along trajectories.	0-2$[\times 10^n]$	I
XL, E, B, HV	Length ; electric field ; magnetic field ; option : element inactive ($HV = 0$) horizontal ($HV = 1$) or vertical ($HV = 2$) separation	m, V/m, T, 0-2	3*E, I
	Entrance face :		
X_{E}, λ_{E_E}, λ_{B_E}	Integration zone extent ; fringe field extents, E and B respectively ($\simeq$ gap height)	3*cm	3*E
C_{E0}–C_{E5}	Fringe field coefficients for E	6*no dim.	6*E
C_{B0}–C_{B5}	Fringe field coefficients for B	6*no dim.	6*E
	Exit face :		
X_S, λ_{E_S}, λ_{B_S}	See entrance face	3*cm	3*E
C_{E0}–C_{E5}		6*no dim.	6*E
C_{B0}–C_{B5}		6*no dim.	6*E
XPAS	Integration step	cm	E
KPOS, *XCE*, *YCE, ALE*	*KPOS*=1 : element aligned, 2 : misaligned ; shifts, tilt (unused if *KPOS*=1)	1-2, 2*cm, rad	I, 3*E

[1] Use *PARTICUL* to declare mass and charge.