

Source routine

An introduction to a new approach to source routines

Beginner course – INTA, April 2024

Why do we need a source routine

 The source routine is used to define complex sources, when the options provided in the BEAM, BEAMPOS and BEAMAXES cards are not enough.

- Most common use cases:
 - Mixed field
 - Beam with an energy spectrum
 - Complex beam shape
 - Second step of a two-step simulation



The "old" source routine

- Scary for beginners, limited documentation
- Written according the FORTRAN 77 standard, prone to errors

1	-			64 *	push many but this way we reserve a
2	*=	source ====================================			stack for the secondaries to be gene
3	*	source			Npflka is the stack counter: of cour
4	*	SUBROUTINE SOURCE (NOMORE)			must be =0
5		SUBRUCITINE SUBRCE (NUMORE)		68	NPFLKA = NPFLKA + 1
6		INCLUDE 'dblprc.inc'			Wt is the weight of the particle
7		INCLUDE 'dimpar.inc'		70	WTFLK (NPFLKA) = ONEONE
8		INCLUDE 'iounit.inc'		71	WEIPRI = WEIPRI + WTFLK (NPFLKA)
9	*	INCLUDE IOUNIC.INC			Particle type (1=proton). Ijbea
	*-			73 *	
10	*-				+
11					(Radioactive) isotope:
12	*	Copyright (C) 2003-2019: CERN & INFN		76	IF (IJBEAM .EQ2 .AND. LRDBEA
13	*	All Rights Reserved.		77	IARES = IPROA
14	*			78	IZRES = IPROZ
15		New source for FLUKA9x-FLUKA20xy:		79	IISRES = IPROZ
16	*			80	
17	*	Created on 07 January 1990 by Alfredo Ferrari & Paola Sala			CALL STISBM (IARES, IZRES, II
18	*	Infn - Milan		81	IJHION = IPROM * 100000 + MOD
19				82	IJHION = IJHION * 100 + KXH
20		This is just an example of a possible user written source routine.		83	IONID = IJHION
21	*	note that the beam card still has some meaning - in the scoring the		84	CALL DCDION (IONID)
22	*	maximum momentum used in deciding the binning is taken from the		85	CALL SETION (IONID)
23	*	beam momentum. Other beam card parameters are obsolete.		86	LFRPHN (NPFLKA) = .FALSE.
24	*			87 *	
25	*	Output variables:			+
26	*			89 *	Heavy ion:
27	*	Nomore = if > 0 the run will be terminated		90	ELSE IF (IJBEAM .EQ2) THEN
28	*			91	IJHION = IPROM * 100000 + MOD
29	*-			92	IJHION = IJHION * 100 + KXH
30	*			93	IONID = IJHION
31		INCLUDE 'beamcm.inc'		94	CALL DCDION (IONID)
32		INCLUDE 'fheavy.inc'		95	CALL SETION (IONID)
33		INCLUDE 'flkstk.inc'		96	ILOFLK (NPFLKA) = IJHION
34		INCLUDE 'ioiocm.inc'		97 *	Flag this is prompt radiation
35		INCLUDE 'ltclcm.inc'		98	LRADDC (NPFLKA) = .FALSE.
36		INCLUDE 'paprop.inc'		99 *	Group number for "low" energy neu
37		INCLUDE 'sourcm.inc'	1	00	IGROUP (NPFLKA) = 0
38		INCLUDE 'sumcou.inc'	1	01 *	Parent radioactive isotope:
39	*		1	02	IRDAZM (NPFLKA) = 0
40		LOGICAL LFIRST, LISNUT	1	03 *	Particle age (s)
41	*		1	04	AGESTK (NPFLKA) = +ZERZER
42		SAVE LFIRST	1	05 *	Kinetic energy of the particle (G
43		DATA LFIRST / .TRUE. /	1	06	TKEFLK (NPFLKA) = SQRT (PBEAM
44	*	Statement function:	1	07	8 - AM (IONID)
45		LISNUT (IJ) = INDEX (PRNAME (IJ), 'NEUTRI') .GT. 0	1	08 *	Particle momentum
46	*=		==* 1	09	PMOFLK (NPFLKA) = PBEAM
47				10 *	PMOFLK (NPFLKA) = SQRT (TKEFL
48	*	BASIC VERSION		11 *	& + TWOTW
49	*	SHOLE TENSION		12	LFRPHN (NPFLKA) = .FALSE.
50				13 *	1
51		NOMORE = 0			
52	-	+		15 *	Normal hadron:
53	*			16	ELSE
54	-	IF (LFIRST) THEN		17	IONID = IJBEAM
55	*	*** The following 3 cards are mandatory ***		18	ILOFLK (NPFLKA) = IJBEAM
56	*	TKESUM = ZERZER		19 *	Flag this is prompt radiation
57				20	LRADDC (NPFLKA) = .FALSE.
		LFIRST = .FALSE.		21 *	
58	*	LUSSRC = .TRUE.		22	IGROUP (NPFLKA) = 0
59 60	*	*** User initialization *** END IF		23 *	Parent radioactive isotope:
	*	CND IF		24	IRDAZM (NPFLKA) = 0
61	*				Particle age (s)
62	*	+	* 1	20 *	Furthere age (5)

```
63 * Push one source particle to the stack. Note that you could as well
                                                                               126
                                                                                            AGESTK (NPFLKA) = +ZERZER
                                reserve a maximum amount of space in the
                                                                              127 * | Kinetic energy of the particle (GeV)
                                to be generated
                                                                              128
                                                                                            TKEFLK (NPFLKA) = SQRT ( PBEAM**2 + AM (IONID)**2 )
                                r: of course any time source is called it
                                                                              129
                                                                                                           - AM (IONID)
                                                                               130
                                                                                   * | Particle momentum
                                                                               131
                                                                                            PMOFLK (NPFLKA) = PBEAM
                                                                               132
                                                                                            PMOFLK (NPFLKA) = SQRT ( TKEFLK (NPFLKA) * ( TKEFLK (NPFLKA)
                                                                              133 *
                                                                                                               + TWOTWO * AM (IONID) ) )
                                                                               134 *
                                ..). Ijbeam is the type set by the BEAM
                                                                              135 *
                                                                                            Check if it is a neutrino, if so force the interaction
                                                                               136
                                                                                            (unless the relevant flag has been disabled)
                                                                                   *
                                                                              137
                                                                                            IF ( LISNUT (IJBEAM) .AND. LNUFIN ) THEN
                                                                               138
                                                                                               LFRPHN (NPFLKA) = .TRUE.
                                 . LRDBEA ) THEN
                                                                              139
                                                                              140
                                                                                   *
                                                                              141 *
                                                                                         Not a neutrino
                                                                              142
                                                                                            ELSE
                                IZRES, IISRES )
                                                                              143
                                                                                              LFRPHN (NPFLKA) = .FALSE.
                                0000 + MOD ( IPROZ, 100 ) * 1000 + IPROA
                                                                              144
                                                                                            END IF
                                   + KXHEAV
                                                                              145 *
                                                                              146 *
                                                                              147
                                                                                         END IF
                                                                              148 +
                                                                              149 * +-----
                                                                               150 * From this point ....
                                                                              151 * Particle generation (1 for primaries)
                                                                              152
                                                                                        LOFLK (NPFLKA) = 1
                                                                               153
                                                                                   * User dependent flag:
                                0000 + MOD ( IPROZ, 100 ) * 1000 + IPROA
                                                                              154
                                                                                        LOUSE (NPFLKA) = 0
                                   + KXHEAV
                                                                              155 * No channeling:
                                                                                         KCHFLK (NPFLKA) = 0
                                                                               156
                                                                              157
                                                                                         ECRFLK (NPFLKA) = ZERZER
                                                                                   * Extra infos:
                                                                               158
                                                                              159
                                                                                        INFSTK (NPFLKA) = 0
                                                                                         LNFSTK (NPFLKA) = 0
                                                                               160
                                                                               161
                                                                                         ANFSTK (NPFLKA) = ZERZER
                                energy neutrons, set to 0 anyway
                                                                                   * Parent variables:
                                                                               162
                                                                               163
                                                                                         IPRSTK (NPFLKA) = 0
                                                                               164
                                                                                         EKPSTK (NPFLKA) = ZERZER
                                                                               165
                                                                                   * User dependent spare variables:
                                                                               166
                                                                                       DO 100 ISPR = 1, MKBMX1
                                                                              167
                                                                                            SPAREK (ISPR, NPFLKA) = ZERZER
                                article (GeV)
                                                                              168
                                                                                   100 CONTINUE
                                 T ( PBEAM**2 + AM (IONID)**2 )
                                                                                   * User dependent spare flags:
                                                                              169
                                                                              170
                                                                                        DO 200 ISPR = 1, MKBMX2
                                                                                            ISPARK (ISPR,NPFLKA) = 0
                                                                              171
                                                                              172 200 CONTINUE
                                RT ( TKEFLK (NPFLKA) * ( TKEFLK (NPFLKA)
                                                                              173 * Save the track number of the stack particle:
                                  + TWOTWO * AM (IONID) ) )
                                                                                         ISPARK (MKBMX2,NPFLKA) = NPFLKA
                                                                              174
                                                                               175
                                                                                         NPARMA = NPARMA + 1
                                                                                         NUMPAR (NPELKA) = NPARMA
                                                                              176
                                                                              177
                                                                                         NEVENT (NPFLKA) = 0
                                                                                         DENEAR (NPELKA) = +ZERZER
                                                                              178
                                                                              179 \star
                                                                                      ... to this point: don't change anything
                                                                              180
                                                                                         AKNSHR (NPFLKA) = -TWOTWO
                                                                               181
                                                                                   Cosines (tx,ty,tz)
                                                                               182
                                                                                         TXFLK (NPFLKA) = UBEAM
                                                                              183
                                                                                         TYFIK (NPFIKA) = VRFAM
                                energy neutrons, set to 0 anyway
                                                                              184
                                                                                         TZFLK (NPFLKA) = WBEAM
                                                                               185
                                                                                        TZFLK (NPFLKA) = SQRT ( ONEONE - TXFLK (NPFLKA)**2
                                                                              186 *
                                                                                                                - TYFLK (NPFLKA)**2 )
                                                                                       5
                                                                              187 * Polarization cosines:
                                                                               188
                                                                                         TXPOL (NPFLKA) = -TWOTWO
```

18	9	TYPOL (NPFLKA) = +ZERZER
	0	TZPOL (NPFLKA) = +ZERZER
19	1 *	Particle coordinates
19	2	XFLK (NPFLKA) = XBEAM
19	3	YFLK (NPFLKA) = YBEAM
19	4	ZFLK (NPFLKA) = ZBEAM
19	5 *	Calculate the total kinetic energy of the primaries: don't change
19		+*
19	7 *	(Radioactive) isotope:
19	в	IF (IJBEAM .EQ2 .AND. LRDBEA) THEN
19	9 *	
20	ð *	**
20	1 *	Heavy ion:
20	2	ELSE IF (ILOFLK (NPFLKA) .EQ2 .OR.
20	3	8 ILOFLK (NPFLKA) .GT. 100000) THEN
20	4	TKESUM = TKESUM + TKEFLK (NPFLKA) * WTFLK (NPFLKA)
20	5 *	
20	6 *	**
20	7 *	Standard particle:
20	в	ELSE IF (ILOFLK (NPFLKA) .NE. 0) THEN
20	9	TKESUM = TKESUM + (TKEFLK (NPFLKA) + AMDISC (ILOFLK(NPFLKA)))
21	0	★ WTFLK (NPFLKA)
21	1 *	
21	2 *	**
21	3 *	
21	4	ELSE
21	5	TKESUM = TKESUM + TKEFLK (NPFLKA) * WTFLK (NPFLKA)
21	5	END IF
21	7 *	
21	B *	+*
21	9	RADDLY (NPFLKA) = ZERZER
22	ð *	Here we ask for the region number of the hitting point.
22		NREG (NPFLKA) =
22		The following line makes the starting region search much more
		robust if particles are starting very close to a boundary:
22	4	CALL GEOCRS (TXFLK (NPFLKA), TYFLK (NPFLKA), TZFLK (NPFLKA))
22	5	CALL GEOREG (XFLK (NPFLKA), YFLK (NPFLKA), ZFLK (NPFLKA),
22		<pre>8 NRGFLK(NPFLKA), IDISC)</pre>
22	7 *	Do not change these cards:
22	В	CALL GEOHSM (NHSPNT (NPFLKA), 1, -11, MLATTC)
22		NLATTC (NPFLKA) = MLATTC
	0	CMPATH (NPFLKA) = ZERZER
	1	CALL SOEVSV
23		RETURN
		💳 End of subroutine Source ====================================
23		END
23		
23	6	



The "new" source routine

- Simplified appearance
- Long & meaningful names for variables and routines
- Forced declaration of variables Use of **implicit none**
- Documented by comments and in the manual
- Variables for user's usage clearly indicated
- Lines not to be edited are "hidden" in routines in the source_library.inc library file
- Old source routines can still be used



The "new" source routine

subroutine SOURCE (momore)

Copyright (C) 2020: CERN Source routine or FLUKA 4: Created on 24 September 2020 by David Hervath & Roberts Versac FLI Bearlines Modified on 17 November 2020 by David Horvath 0 Roberto Versaci ELI Beamlines this is a simplified user written source routine utilizing a separate source voutine library. It is intended as an alternative new-user-friendly version of the swarte.f routime. Existing FLMAA 4 source routines remain commative. Note that the beam card still has some meaning - in the scoring maximum momentum used in deciding the binning is taken from the beam momentum. Other beam card parameters are obsolete.

Output variables: nomore - if > 0 the run will be terminated

Quick start guide:

This user source routine template mins to modernize the legacy routine by implementing modern Fortram conventions and to provide built in sampling functions. The users only need to change / add code between the BEGISWEMG and END marks, one soction for doclaration of user variables, and one for assigning values to the beam parameters.

By default there is no user variable defined, and all code lines for parameter assignment are communication. These communic start with the symbol: " $^{++}$. To enable one, the $^{++}$ needs to be deleted.

(Note: In Fortran each code line should start in column 7 or further in.) Every beam parameter has a default value based on the FLURA input file. A parameter assignment should only be used if the default value has to be

There are three ways to assign a value to a parameter: 1. Direct assignment: A parameter is equal to a value. For example: scentum, energy = 8.100

If the parameter defined as a double precision, then the assigned value should be represented as double precision as well so as to not loose numerical precision. To do this a "D" exponential mark must be used.

Using a sampling function: A parameter is assigned to a value, which is calculated by a separate function. For example:

coordinate x = sample flat distribution([min], [max]) The parameters between the '[' and ']' brackets need to be replaced with numbers, or user variables containing the desired values.

 Using a sampling subroutine: They are similar to function, but they are not returning values directly; instead they modify the variables in their argument list. For example: call sample_annular_distribution(Inmin], [rmax], coordinate_x, con

The example above has two input parameters between brackets, and two output parameters (without bracket). The input parameters have to be provided, as for functions. The output parameter names usually don't need to be changed, but there are cases, where a subset of possible output parameters has to be selected.

for further details see the flink manual.

module source_variables implicit none

integer, save :: particle_code integer, save :: heavyion_stomic_number, heavyion_mass_number, heavyion_isomer logical, save :: radioactive_isotope

double precision, save :: nonestum_energy, particle_weight legical, save :: energy_logical_flag

double precision, save :: divergence_x, divergence_y logical, save :: gaussian_divergence_logical_flag

double precision, save :: coordinate_s, coordinate_y, coordinate_r

integer, save :: direction_flag double precision, save :: direction_cosx, direction_cosy, direction_cosx

double precision, save :: polarization_cosy, polarization_cosy, polarization_cosy 223 + radioactive_isotope - .trve.

double precision, save :: particle_age double precision, save :: kshort_component double precision, save :: delayed_radioactive_decay end module source_variables

Include 'source_library.isc'

FLUKA

use source_library use source_variables implicit none logical lfirst save lfirst data lfirst / .true. / integer nomore logical debug_logical_flag
data debug_logical_flag / .false. / integer debug_lines data debug_lines / 100 / double precision adamny twoe(chase space) phase space entry

double precision sample_flat_momentum_energy double precision sample_gaussian_momentum_energy double precision sample_initegram_momentum_energy double precision sample_initegram_momentum_energy double precision sample_gaussian_distribution double precision sample_flat_distribution

double precision FLENDM

END of user declared variables nomore = 0

if (lfirst) them
 call initialization()
 lfirst = .false.
end if

BEGINNING of customizable code

1 1. Accessing variables from the SOURCE card tvalues set on the SOURCE card can be accessed with the following variables: mumerical values (souble precision): muHSOU(), wwFSOU(), ... wHSOU(18) SOURT text (& character): fontoor

1 2. Primary particle

1 2.1. Particle code FILME particle code of the primary See section 5.1 of the FILMA menual for the list of particle code ! Defaulti # Farticle code of the primary defined on the BEAM card if present, otherwise 1 (proton) 195 196 • particle_code = ...

1 2.2. Heavy ion 1F the HLAWYON particle type (particle_code - -2) has been selected on the BLAW card, the ion can be specified with the following parameters: 1 Specified on HL-MONE card (if present), otherwise 2-6, A-12, 1-8 (12C)

 heavyion_atomic_number = ...
 heavyion_mass_number = ...
 heavyion_isomer = ... | 2.3. Radioactive isotope

i Selects if the specified ion is to be used as a radioactive source I Default: remail() .true. : if ISOTOPE particle type is selected on the BEAM card, .false. : otherwise

! Note: ! Requires RADDECAY (semi-analogue) and DCYSCDRE cards ! Direction sampling is always isotropic ! Romentum / emergy settings are disregarded

3. Particle momentum / energy and weight

1 3.1. Nomentum & Energy Set the momentum [GeV/c] or the kinetic energy [GeV] of the primary particle for heavy ions values are per nuclear mass unit. 233 | Default: 234 | Momentum calculate 235 236 * momentum_energy = ... ! Default: ! Romentum calculated from values set on the BEAM card (if present), 200 GeV/c otherwise

1 3.2. Energy flag

I Solici becomendia and energy I fibe energy flg is: - faits: The monoton_energy worlable contains the momentum of the particle I fibe monoton_energy worlable contains the kisetic energy of the particle I effective I faits: energy_logical_flag = .true

237 238 239 240 241 242 243 244 245 246 245 246 247 248 * 249 1 3.3. Particle weight

f Sets the initial weight of the primary Default: 1.000 particle_weight = ...

3.4. Sampling functions and subroutines

3.4.1. Flat distribution momentum_energy = sample_flat_momentum_energy([min], [max])

3.4.2. Gaussian distribution

271 | -----272 273 * momentum_energy = sample_gaussian_momentum_energy([mean], [fwhm]) 3.4.3. Maxwell-Boltzmann distribution

Tengerature is given in GeV, energy flag must be ,true, momentum energy = sample maxwell boltzmann energy([temperature])

1 3.4.4. Sampling from histogram Possible [usit]s: "ter", "Gev", "Mev", "kev" "ev" "ter/c", "Gev(c", "Mev/c", "kev/c", "kev/c", "kev/c", Mistogram file in the bin - Thmac (of the bin) - GW/GE (bin beight; NOTE: doesn't need to be normalized)

momentum energy = sample histogram momentum energy([filename], [unit]) 1 3.4.5. Exponential distribution

------Ingut variables: - e_min [GeV] - e_max (GeV] - intensity_ratio = (int_e_max / int_e_min) Outgut variables: - maxetian_energy - particle_mergy call sample exponential energy weight([e min], [e max], [intensity ratio], momentum energy, particle weight)

4. Beam angular divergence

1 4.1. Divergence value 1 Sets the beam divergence in the X-2 (divergence_x) and Y-2 (divergence_y) planes [rad] Divergences are applied before beam direction Defaults: Set on the BEAM card if present (converted to radians), 0.0 otherwise
 315
 ! Divergences are appl

 316
 ! Defaults:

 317
 ! Set on the BEAM

 318
 * divergence_x = ...

 319
 * divergence_y = ...

 320
 * divergence_y = ...

1 4.2. Divergence type 1. Setto: bettern fits and Gaussian divergence. 15 fits is at the 17 fit is at the 1. the. I fit divergence is forth. Divergence values are taken as full opening angle 1. the. I fit divergence is downlow - Revenue values are taken as PANN of the distribution lefthalt; 1. Set in the RADM card (if present), false, otherwise
 15
 1 Selects between flat and Gaussian divergence

 16
 17 is is not too.

 17
 1 sets to too.

 18
 1 felster.

 18
 1 true.

 19
 1 felster.

 19
 1 felster.

 10
 1 felster.

 10
 1 felster.

 10
 1 felster.

 11
 1 felster.

 120
 1 felster.

 120
 1 felster.

 121
 1 felster.

 122
 1 felster.

 123
 1 felster.

 124
 1 felster.

 125
 1 felster.

 126
 1 felster.

 127
 1 felster.

 128
 1 felster.

 129
 1 felster.

 120
 1 felster.

 121
 1 felster.

 122
 1 felster.

 1231
 1 felster.

5. Beam starting position

1 5.1. Coordinates

Note: the snapshot is not meant to be read – Detailed view will follow

Sets the starting coordinates (x,y,z) of the beam [cm] locality: Coordinates set on the BEAMPOS card if present, (0.000, 0.000, 0.000) otherwise

5.2. Sampling functions and subroutines 1 5.2.1. Flat distribution

Replace [a] with "x", "y", or "z". 357
358 * coordinate [a] s sample flat distribution([min], [max]) 1 5.2.2. Gaussian distribution

Replace [a] with fst, fyt, or fst, coordinate [a] + sample gaussian distribution([mean], [fwhm])

1.5.2.3. Annular distribution

 Applies an anniher distribution to any two convinctos
 Topol variables:
 Two contributes of the contex of the annular distribution (coordinate_(a/b)) [cn]
 Two contributes of the contex of the annular distribution (coordinate_(a/b)) [cn]
 mapping contributes of the contex of the annular distribution (coordinate_(a/b)) [cn] supplate [a] and [a] match A , , ,
 cutput variables:
 - Modified coordinates of the sampled location (input values have been overwritten)

call sample annular distribution([rmin], [rmax], coordinate [a], coordinate [b]) 6. Beam direction

1 6.1. Direction cosines 1 Sets the direction cosines of the beam with respect to the X,Y and Z-axis Defaults: Set on the BEAMPOS card if present, otherwise (0.000, 0.000, 1.000)

1 6.2. Direction flag and the second second second second (with a still be sound(ind))
 a (1) since in a state into account (with a still be sound(ind))
 a (1) since into account (with a still be sound(ind))
 a (1) since into account (with a still be sound(ind))
 a (1) since into account (with a still be sound (ind))
 a (1) since into account (with a still be sound (ind))
 a (1) since into account (with a still be sound (ind))
 a (1) since into account (with a still be sound (ind))
 a (1) since into account (with a still be sound (ind))
 a (1) since into account (with a still be sound (ind))
 a (1) since into account (ind))

direction_flag = ...

1 6.3. Sampling functions and subroutines 4.3.1. Isotropic distribution Cutput variables: - direction_cosx - direction_cosy - direction_cosz

call sample_isotropic_direction(direction_cosx, direction_cosy, direction_cosz) 1 7. Other changeable parameters

I for most of the uses none of these parameters should be changed from the defaults 1.7.1. Polarization coninan The three inputs indicate the direction cosines of the particle polarization Defaults: (-2.000, 0.000, 0.000)

polarization_cosx = ...
polarization_cosy = ...
polarization_cosz = ... 1 7.2. Particle age

Sets the starting age of the primary particle in seconds Default: 0.000 640 s particle are s ...

1 7.3. Kshort compo Sets The Kshort component of the KO/K8ber Default: -2.00

Source routine

i55 i56 * kshort_component = . 1 7.4. Delayed radioactive decay

Sets the delay for the radioactive decay with respect to the standard primary zero time Default: 0.000

465 * delayed_radioactive_decay = ...

1 8. Sampling from phase space file Allows to read particle information from a phase space file and sets the primary accordingly

! 8.1. Input variables

- [filename] - [filename] - [energy_umil]: - Possible [energy unil]s: "TeV", "GeV", "MeV/c", "keV/c", "eV/c" - "TeV/c", "GeV/c", "MeV/c", "keV/c", "eV/c"

Plankie terming multiple "Terric", "Well", "Well", "Well", "Well", "Well",
 Plankie (Territoria), "Well", "We

1 8.2. File format The phase space file has to contain the following columns in this order:

! - Particle code [integer]

1 - Particle momentum / energy [double precision]

! - Starting X coordinate [double precision] ! - Starting Y coordinate [double precision] ! - Starting Z coordinate [double precision]

 Starting X direction cosine [double precision Starting Y direction cosine [double precision Starting Z direction cosine [double precision] I - Particle weight [double precision]

1 8.3. Output variables

phase_space_entry: Variable containing information of a single particle from the phase space file
 nomore: Fleg to indicate that all particles has been read

1 8.4. Subroutine call

call read_phase_space_file([filename], [energy_unit], [length_unit], phase_space_entry, [sequential_logical_flag], nomore 1 8.5. Reading information from 'phase space entry' variable

The information stored in the 'phase space entry' has to be copied to the appropriate variables particle_code = phase_space_entry%pc momentum_energy = phase_space_entry%m_e

emergy_logical_flag = .true. coordinate_x = phase_space_entry% coordinate_y = phase_space_entry% coordinate_z = phase_space_entry%

direction_cosx = phase_space_entry%
direction_cosy = phase_space_entry%
direction_cosz = phase_space_entry%

535 * particle_weight = phase_space_entry%we

9. Debugging

1 9.1. Debug logical flag Enables or disables the printout of the beam parameters for debugging Possible values: . false. : Debug output disabled (Default) . true. : Debug output enabled

debug_logical_flag = .true. 1 9.2. Debug lines

Sets the maximum number of lines printed in the debug output Default: 100

if (debug_logical_flag) call print_primary(debug_lines)

5

58 * debug_lines = 100

END of customizable code - Do not change below

call set primary()

573 return 574 see End of subroutine Source ------

end if

User declaration

! BEGINNING of user declared variables

• Dedicated space for the declaration of additional variables (and functions)



User declaration (example)

! ______ BEGINNING of user declared variables

integer :: counter
double precision :: energy
logical :: flag

END of user declared variables



Initialization

 Initialization of internal variables Runs every time, resetting their values to the defaults

Custom initialization block

Runs only the first time the source routine is used

call initialization()

if (first_run) then

BEGINNING of custom initialization

END of custom initialization

first_run = .false.
end if



Main section

- For setting the internal variables directly, or using one of the sampling routines
- To enable a line, remove the `*` (The command should start on the 7th column)
- The variables with `[``]` brackets and ... are placeholders, they need to be replaced with values or user variables (Brackets should be deleted)
- Always use double precision format for floating point numbers (1.0d0)

```
BEGINNING of customizable code
particle_code = ...
heavyion_atomic_number = ...
heavyion_mass_number = ...
heavyion isomer = ...
momentum_energy = ...
energy_logical_flag = .true.
particle_weight = ...
momentum_energy = sample_flat_momentum_energy( [min], [max] )
momentum_energy = sample_gaussian_momentum_energy( [mean], [fwhm] )
momentum_energy = sample_maxwell_boltzmann_energy( [temperature] )
momentum_energy = sample_histogram_momentum_energy( [filename], [unit] )
momentum_energy = sample_spectrum_momentum_energy( [filename], [unit] )
call sample_exponential_energy_weight( [e_min], [e_max], [intensity_ratio], momentum_energy, particle_weight )
divergence_x = ...
divergence_y = ...
gaussian_divergence_logical_flag = .true.
coordinate x = ...
coordinate_y = ...
coordinate_z = ...
coordinate_[a] = sample_flat_distribution( [min], [max] )
coordinate [a] = sample gaussian distribution( [mean], [fwhm] )
call sample_annular_distribution( [rmin], [rmax], coordinate_[a], coordinate_[b] )
direction_cosx = ...
direction_cosy = ...
direction_cosz = ...
direction_flag = ...
call sample_isotropic_direction( direction_cosx, direction_cosy, direction_cosz )
polarization_cosx = ...
polarization_cosy = ...
polarization_cosz = ...
particle_age = ...
kshort component = ...
delayed_radioactive_decay = ...
call read_phase_space_file( [filename], [energy_unit], [length_unit], phase_space_entry, [sequential_logical_flag], nomore )
particle_code = phase_space_entry%pc
momentum_energy = phase_space_entry%m_e
energy_logical_flag = .true.
coordinate_x = phase_space_entry%x
coordinate_y = phase_space_entry%y
coordinate_z = phase_space_entry%z
direction_cosx = phase_space_entry%u
direction_cosy = phase_space_entry%v
direction_cosz = phase_space_entry%w
particle_weight = phase_space_entry%wei
```

debug_logical_flag = .true.

```
* debug_lines = ...
```



Primary particle

- particle_code = ...
- By default, the particle type given in the **BEAM** card is taken
- Particle codes explained in FLUKA manual section 5.1

*

- Possible application: beam made of more than one type of particles
 - * heavyion_atomic_number = ... * heavyion_mass_number = ... * heavyion_isomer = ...
- Only used if primary particle is set to HEAVYION or ISOTOPE on the **BEAM** card
- Default values are set on the **HI-PROPE** card, or for ¹²C if the card is missing



Energy / momentum

* momentum_energy = ...

- By default, the particle <u>momentum</u> is expected
- The default value is based on the **BEAM** card (Automatically converted into momentum if energy is given in the **BEAM** card)
- If energy is specified in the source routine, the following logical value must be set .true.

* energy_logical_flag = .true.



Energy / momentum

- The momentum divergence set on the **BEAM** card is not retained
- It in necessary to specify it in the source routine
- It is easy with the supplied functions / subroutine

Sampling functions:

Analytical

- Flat (Uniform)
- Gaussian
- Maxwell-Boltzmann
- Exponential

From an external file

- Histogram
- Continuous spectrum
- Discrete spectrum



Energy / Momentum – Analytical samplings

- Flat / uniform:
 - Samples uniformly between two momentum [GeV/c] or, energy [GeV] values
 - * momentum_energy = sample_flat_momentum_energy([min], [max])
- Gaussian:
 - Samples from a Gaussian distribution with a given mean and FWHM value ([GeV/c] or [GeV])
 - * momentum_energy = sample_gaussian_momentum_energy([mean], [fwhm])
- Maxwell-Boltzmann:
 - Samples from a Maxwell-Boltzmann energy distribution with a given temperature [GeV]
 - It is only meaningful if the energy_logical_flag is set to .true.
 - * momentum_energy = sample_maxwell_boltzmann_energy([temperature])



Energy / Momentum – Analytical samplings

• Exponential:

• Samples according the exponential distribution between two energies [GeV], with a given intensity ration at the specified energies

```
* call sample_exponential_energy_weight(
       [e_min], [e_max], [intensity_ratio],
       momentum_energy, particle_weight
     )
```

- It is only meaningful if the energy_logical_flag is set to .true.
- Note 1: Different syntax used (function vs. subroutine)
- Note 2: The return values momentum_energy and particle_weight are among the arguments
- Note 3: This is a biased sampling! It is not suitable for cases where fully analogue simulation is required (E.g. scoring with DETECT card)



Energy / Momentum – Sampling from file

• Histogram:

• Samples from a histrogram specified in an external file

```
* momentum_energy = sample_histogram_momentum_energy(
    [filename], [unit]
)
```

- The external file needs to have 3 columns:
 - Lower energy boundary of the histogram's bins
 - Upper energy boundary of the histogram's bins
 - Intensity per energy unit (dN/dp or dN/dE)
- The particle momentum / energy is sampled uniformly withing a bin
- [unit] variable is the unit used in the external file. Possible values:
 - TeV/c, GeV/c, MeV/c, keV/c, J
 - TeV, GeV, MeV, keV, eV, J



Energy / Momentum – Sampling from file

• Continuous spectrum:

• Samples from a continuous spectrum specified in an external file

```
* momentum_energy = sample_spectrum_momentum_energy(
       [filename], [unit]
)
```

• Samples from a discrete (line) spectrum specified in an external file

```
* momentum_energy = sample_discrete_momentum_energy(
    [filename], [unit]
)
```

• The external file needs to have 2 columns:

• Energy

- Intensity at the specified energy
- With the continuous spectrum, the intensity is linearly interpolated



Energy / Momentum – Examples

- Setting energy flag to true: energy_logical_flag = .true.
- Monoenergetic beam:

 $momentum_energy = 1.0d-1$

Gaussian beam:

```
momentum_energy = sample_gaussian_momentum_energy(1.0d-1, 1.0d-2)
```



Energy / Momentum – Examples

• Sampling from an exponential distribution:

call sample_exponential_energy_weight(1.0d-6, 1.0d-3, 0.01d0,

- & momentum_energy, particle_weight)
- Sampling from an external spectrum:

```
momentum_energy = sample_spectrum_momentum_energy(
& 'spectrum.txt', 'MeV')
```

Note: The & character is for the line continuations, it should be always in column 6.



Source routine – Particle weight

- * particle_weight = ...
- To create biased sources (see Monte Carlo lecture)
- Usually needed only for complex source definitions
- Note: The exponential spectrum sampling subroutine uses variable particle weight, but in that case, it is automatically set



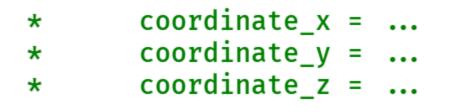
Source routine – Beam divergence

* divergence_x = ...
* divergence_y = ...

- By default:
 - values are taken from the **BEAM** card
 - It is assumed to be a flat angular distribution
- For Gaussian divergence the following logical value must be set .true.
 - * gaussian_divergence_logical_flag = .true.



Source routine – Beam starting position



- By default, values are taken from the **BEAMPOS** card
- Beam shape set in the **BEAM** card, and
- extended sources specified in additional **BEAMPOS** cards are not implemented



Source routine – Beam starting position

Some predefined routines (2 functions and 1 subroutine) are already available:
 Flat distribution:

* coordinate_[a] = sample_flat_distribution([min], [max])

Gaussian distribution:

* coordinate_[a] = sample_gaussian_distribution([mean], [fwhm])

Annular distribution:

* call sample_annular_distribution([rmin], [rmax], coordinate_[a],

Remember the values must be in double precision (1.0d0).

Note: If annular sampling is used, the coordinates have to be selected as well.



Source routine – Beam direction

- * direction_cosx = ...
 - direction_cosy = ...
- * direction_cosz = ...
- By default, values are taken from the **BEAMPOS** card

*

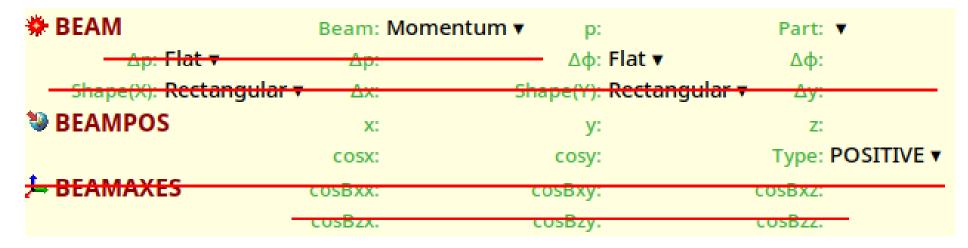
• If the direction_flag is set to: * direction_flag = ...

- 0 : All three values are considered and the they are normalized automatically (Default)
- 1 : The manually set value of the z direction is disregarded. Instead, it is calculated from the x and y direction cosines with a positive sign.
- 2 : As with option 1, but negative sign is used.
- A predefined subroutine is are already available for isotropic direction sampling
 - * call sample_isotropic_direction(direction_cosx, direction_cosy, direction_cosz)



Source routine – Unused values

- It is important to remember, not all values used in the FLUKA input are used in the source routine:
 - The beam momentum distribution
 - The shape of the extended beam / volumetric sources
 - The separate coordinate system set up for the beam



• If one of these features is required, it needs to be programmed in the source routine as well by using the available sampling procedures or by custom code.



Source routine – Phase-space sampling

- Used for the second step in a two-step simulation
- It reads a file containing information on individual particles:
 - Particle code
 - Momentum / energy
 - Starting coordinate
 - Starting direction
 - Weight
- Can replay the particles sequentially, or select from them randomly

*	call read_phase_space_file([filename], [en
* *	particle_code
*	<pre>energy_logical_flag = .true.</pre>
* * *	coordinate_x = phase_space_entry%x coordinate_y = phase_space_entry%y coordinate_z = phase_space_entry%z
* * *	direction_cosx = phase_space_entry%u direction_cosy = phase_space_entry%v direction_cosz = phase_space_entry%w
*	particle weight = phase space entry%wei



Source routine – Debugging

- To help debug the source routine, the major particle parameters can be printed
- To enable this feature, set

* debug_logical_flag = .true.

- The printed parameters:
 - Energy / momentum
 - Coordinates
 - Direction
 - Weight
- The number of primaries printed can be set with:
 - * debug_lines = 100



SOURCE card and passing parameters

- To invoke a source routine, it is necessary to add a **SOURCE** card
- A **SOURCE** card can be empty or can be used to pass parameters to the routine
 - Max. 18 numerical values (WHASOU (ii)) and 1 string (max. 8 characters) (SDUSOU)

SOURCE	#1:7.	#2: 250.	#3: 12.5
sdum: linksour	#4: 3.75	#5:	#6:
	#7:	#8:	#9:
	#10:	#11:	#12:
	#13:	#14:	#15:
	#16:	#17:	#18:

Good practice:

Even if the beam energy / momentum is defined in the source routine, specify it in the **BEAM** card as it is used for internal initialization. Set a momentum value higher than the maximum possible one.



Adding the user routine to the project folder

- 1. Open [Compile] tab
- 2. It is maybe hidden in the dropdown menu
- Click the [Database] button (Use [Add] for an existing file)
- 4. Select the user routine you want to use
- 5. Click [Copy to Project]

The copied user routine will be in the Flair projects directory

🚆 + /mnt/d/WSL/Run/course/sour 🖥 🦘 🔻 险 🛛 🗮 Flair 🛛 🔯 Inj		Compile 🛛 🖧 Rui	n 🛄 Plot	- □ ×
🕘 _ 📖 🛄	Add V Database V Remove Add V Database V Move	o 🛛 🖲 🌽 🖉	ika 💽 Clean Build Action	2
	-	Compile	Action	▼ ×
xecutable:	×	Options:		
File	Туре	Size	Date	
X FLUKA User routir	nes			×
File Size	Date		Desc	
rflctv.f 1149	2021.02.12 08:12:4 refl	ectivity (for optical g	photons)	
rfrndx.f 1441	2021.02.12 08:12:4 refr			
soevsv.f 2554	2021.02.12 08:12:4 sav		•	
source.f 9203	2021.02.12 08:12:4 to c	enerate any distribu	tion for source particles	
4 source_new(1912	4 2021.02.12 08:12:4			
stupre.f 3909	2021.02.12 08:12:4 set			
stuprf.f 1667			ons, muons and neutrinos)	
ubsset.f 5265	2021.02.12 08:12:4 to c		g parameters	
udcdrl.f 2107	2021.02.12 08:12:4 dec			
usimbs.f 3017	2021.02.12 08:12:4 use		e biasing	
usrein.f 1233	2021.02.12 08:12:4 eve			
usreou.f 1463	2021.02.12 08:12:4 pos 2021.02.12 08:12:4 use			
usrglo.f 1460				
usrhsc.f 2700	2021.02.12 08:12:4 USe		isity) SCaling factors	
usrini.f 1314	2021.02.12 08:12:4 use			
usrmed.f 2438			biasing and/or particle select	tions on a
usrout.f 1225	2021.02.12 08:12:4 use			
usrmc.f 1128	2021.02.12 08:12:4 use		esiqual nuclei scoring	
ustckv.f 908	2021.02.12 08:12:4 Use		hatan production	
wvlnsh.f 2489	2021.02.12 08:12:4 Wa	ve Length Shifted p	noton production	
5 <u>C</u> opy to Project	<u>S</u> can Input <u>V</u> iew	<u>C</u> lose		
		Files: 1		



Compiling a custom FLUKA executable

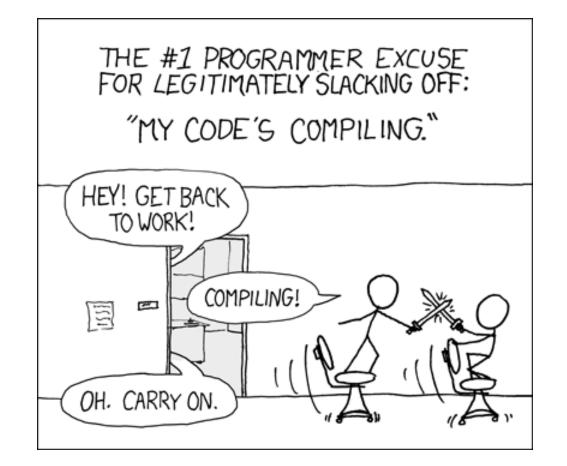
- Verify that the user routine is in the list
- 2. Name your custom executable
- 3. Select the appropriate linker:
 - a. Use Ifluka by default
 - b. Use *ldpmqmd* if DPMJET or RQMD models are needed
- 4. Compile the executable

The custom executable should be set default on the [Run] tab automatically

🚆 + /mnt/d/WSL/Run/course/sour	ce_routine.flair - flair			- 🗆	×
🔜 🧐 👻 📔 🌺 Flair – 🔯 Ing	out 🛛 🙀 Geometry 🛛 🛕			🗘 Viewer	v 🗊
	Add Add Files	00 🥒 📩	3 uka Clean Build Action	Ļ	
0		Compile			▼ ×
Executable: myfluka.exe 2	× .	Options:			
File	Туре	Size	Date		
source_newgen.f	Fortran	19124	2021.06.02 13:25:11		\land
1					
		Files: 1			×



Time for an exercise!



xkcd.com/303



Source routine

