Joint ICTP-IAEA Workshop on Monte Carlo Radiation Transport and Associated Data Needs for Medical Applications

28 October – 8 November 2024

ICTP, Trieste, Italy

Lecture 20

DOSXYZnrc advanced sources: phase-space sources and shared libraries

Reid Townson

Metrology Research Centre
National Research Council Canada

DOSXYZnrc advanced sources

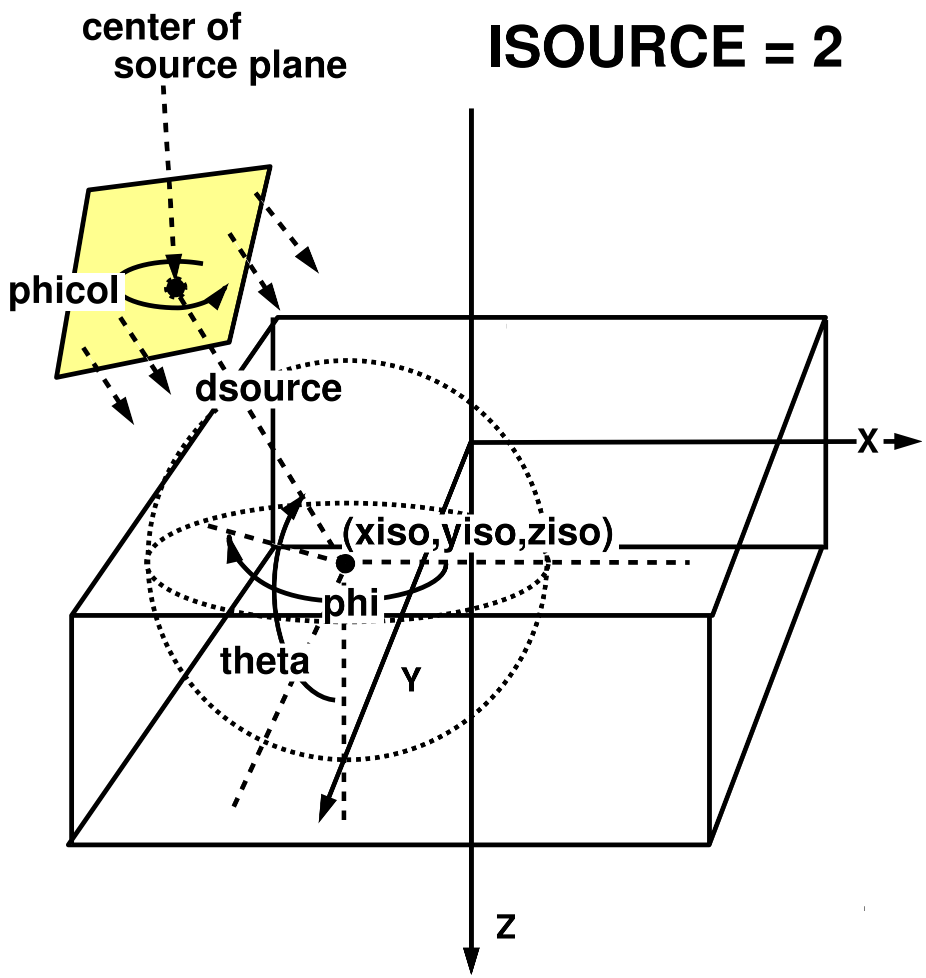
- DOSXYZnrc is often paired with BEAMnrc (e.g. radiotherapy simulation)
  - BEAMnrc performs LINAC modelling
  - DOSXYZnrc calculates dose to patient

How do we interface the two applications?

- Phase-space: save the particles to a file between applications
- Shared library: call BEAMnrc directly from DOSXYZnrc

Phase-space incident from any direction–source 2

(or from multiple, user-defined directions–source 8)



Inputs for phase-space sources

`iqin`: charge of particles to use

- Normally set to 2 to use all particles in phase-space

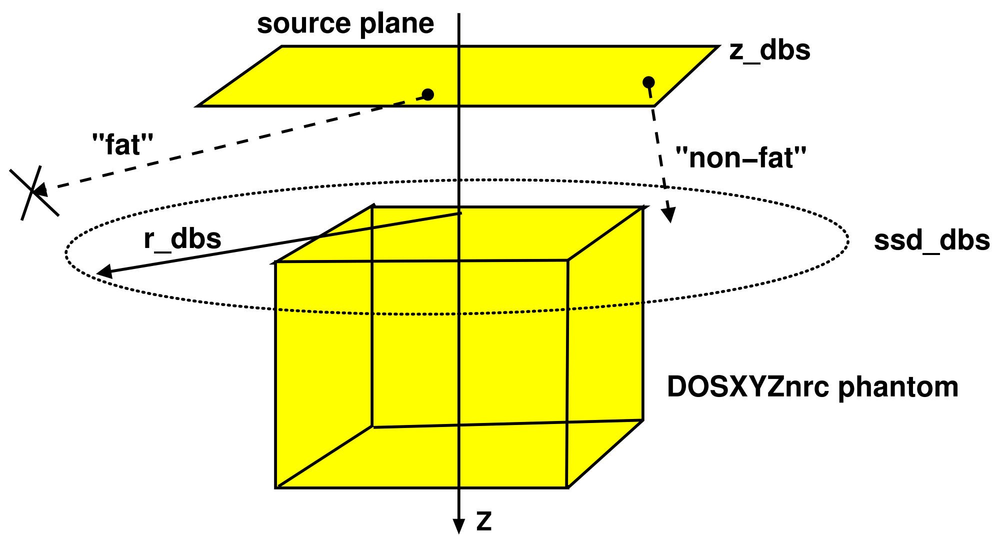
Positioning the source

- `xiso(i)`, `yiso(i)`, `ziso(i)`: isocentre coordinates
- `theta(i)`, `phi(i)`, `phicol(i)`: source direction
- `dsource(i)`: source to isocentre distance
  - Absolute distance from isocentre to source centre (origin of phase-space)
  - For IAEA phase-space, distance from isocentre to the primary source (SAD)

Inputs for phase-space sources (cont.)

If directional bremsstrahlung splitting (DBS) was used to generate the phase-space:

- “Fat” particles will compromise statistics
- Set `i_dbs` to 1
- `r_dbs`: DBS splitting radius
- `ssd_dbs`: SSD at which splitting radius was defined
- `z_dbs`: Z (in accelerator frame) where phase-space file was scored



Inputs for phase-space sources (cont.)

e_split: charged particle splitting

- Only used with photon splitting (n_split $>$ 1)
- Charged particles split e_split times (with weight reduced by 1/e_split) as soon as they enter the phantom geometry
- Prevents high-weight contaminant e^-'s from compromising dose statistics
- For maximum efficiency, set e_split=n_split

Particle redistribution (ISMOOTH)

- On re-use, each particle's (X,Y) and (U,V) are shifted to:
  1. (-X,Y) with (-U,V)
  2. (X,-Y) with (U,-V)
  3. (-X,-Y) with (-U,-V)
- Only valid for symmetric accelerator geometry with treatment field centered on beam axis!

Inputs for phase-space sources (cont.)

Particle recycling (NRCYCL)

- The number of times to recycle each particle before moving on to the next one
- Avoids phase-space restarts $\Rightarrow$ **essential for accurate statistics if no. of histories $>$ no. of particles in phase-space source!**
- If set to 0 (and not using only positrons OR filtering incident particles based on LATCH), automatically calculated using:

$$\mathtt{NRCYCL} = \mathtt{INT}\left(\frac{\text{no. of histories}}{\text{no. of particles with selected charge}}\right) \tag{1.1}$$

- If you find that >1 restart occurs or $>$ 50% of source gets resampled in restart:
  - Particles are missing geometry, or rejected because they are multiple passers or outside BEAM_SIZE
  - Recalculate NRCYCL using formula in manual PIRS-794.

IAEA format phase-space sources

- IAEA format is commonly provided by linac manufacturers
- `dsource` is instead used to define distance from the primary source to the phantom isocentre (SAD).
- Unlike `*.egsphsp`, the incident Z-position of particles is either in the phase-space data (3D phase-space) or header file (planar phase-space).

DOSXYZnrc advanced sources: BEAMnrc shared library

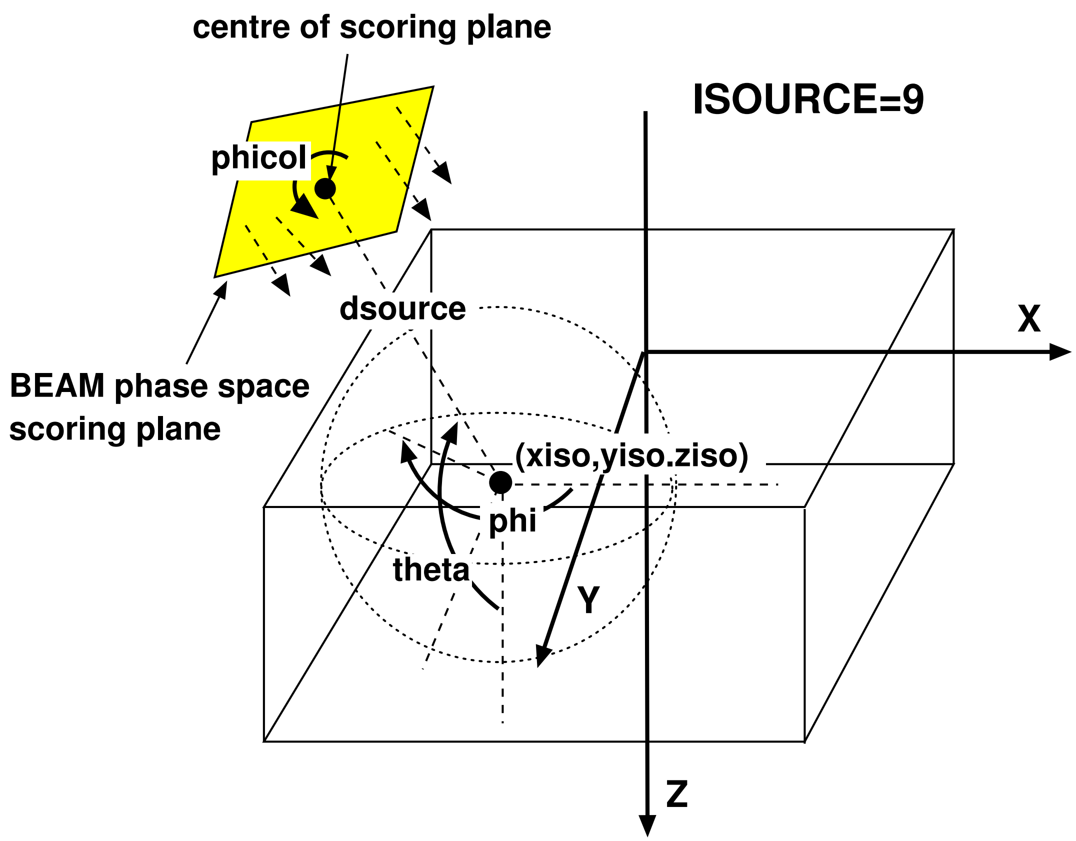
- Instead of saving particles in a phase-space, BEAMnrc can be directly integrated into DOSXYZnrc

- Start by building a shared library for a BEAMnrc accelerator and creating the relevant input file

- Each time DOSXYZnrc asks for a source particle, BEAMnrc is used to generate particles that reach the phantom

Full BEAM simulation incident from any direction–source 9

(or from multiple, user-defined directions–source 10)



Inputs for BEAM simulation source

`BEAM_accelname`: the accelerator to use

- Must be compiled as a shared library:
  - `libBEAM_accelname.so` for Unix/Linux
  - `BEAM_accelname.dll` for Windows

in your `$EGS_HOME/bin/config` directory.

`inputfile` (no `.egsinp` ext.): the BEAMnrc input file

- Must exist in your `$EGS_HOME/BEAM_accelname` directory.
- Must be set up to output phase-space data at a single scoring plane. Instead of being written to a phase-space file, these particles will be used directly.

Inputs for BEAM simulation source (cont.)

`pegsfile` (no `.pegs4inp` ext.): PEGS4 data to be used in the BEAM simulation.

- Must exist in either `$HEN_HOUSE/pegs4/data` or `$EGS_HOME/pegs4/data`.

Inputs for BEAM simulation source (cont.)

If directional bremsstrahlung splitting (DBS) was used:

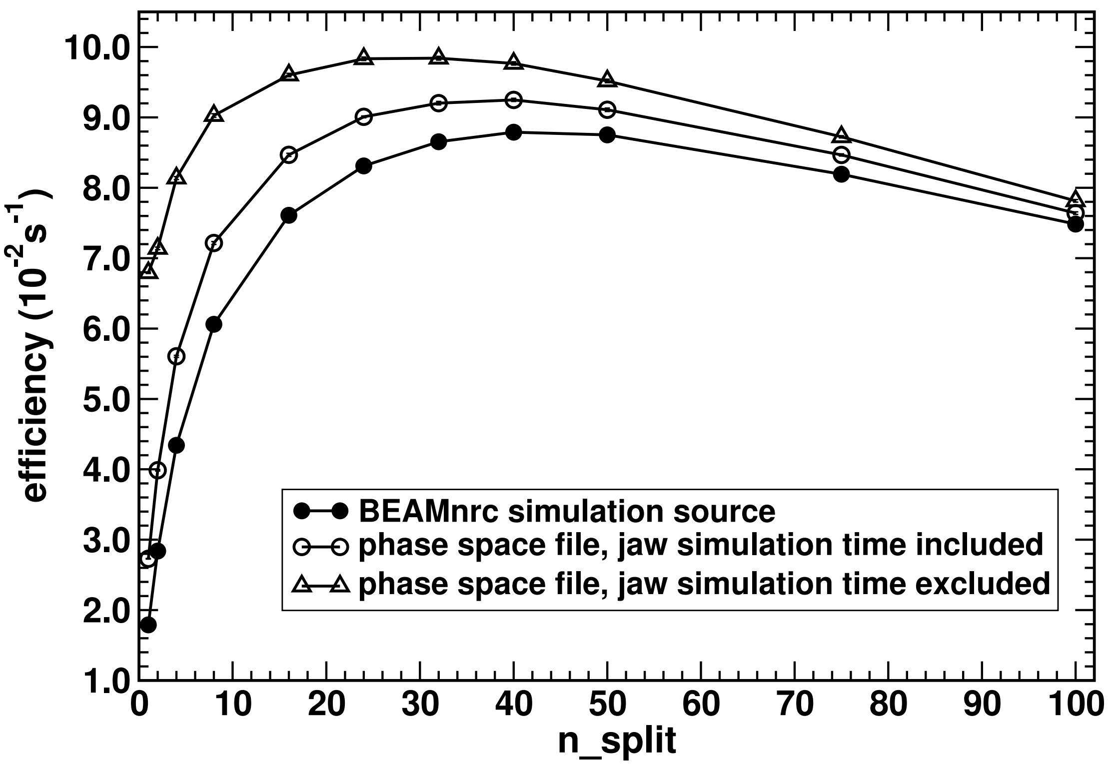
- Set `i_dbs` to 1
- Particle still carries flag indicating if it's "fat" or not, so this is all we need

`e_split`: charged particle splitting

- Same as for phase-space source (2,8)

Efficiency compared to phase-space source

- 6 MV Elekta photon beam, 10×10 cm^2 field (SSD=100 cm), $0.25 \times 0.25 \times 0.25$ cm^3 phantom voxels, doses $> 0.5\text{D}_{\max}$
- 10 GByte phase-space above jaws, 500 MByte below (used as source).



System requirements for BEAM simulation source

- **Unix/Linux**
  1. A working C or C++ compiler.
  2. The `dl` system library.
- **Windows**
  1. Nuttin'
     - We supply a pre-compiled object file for accessing the BEAM shared library (`$HEN_HOUSE/lib/config/load_beamlib.obj`).

The installation should automatically determine whether your system can support BEAM simulation sources and will set it up for you.

Compiling BEAMnrc as a shared library

Go into your `$EGS_HOME/BEAM_accelname` directory and type:

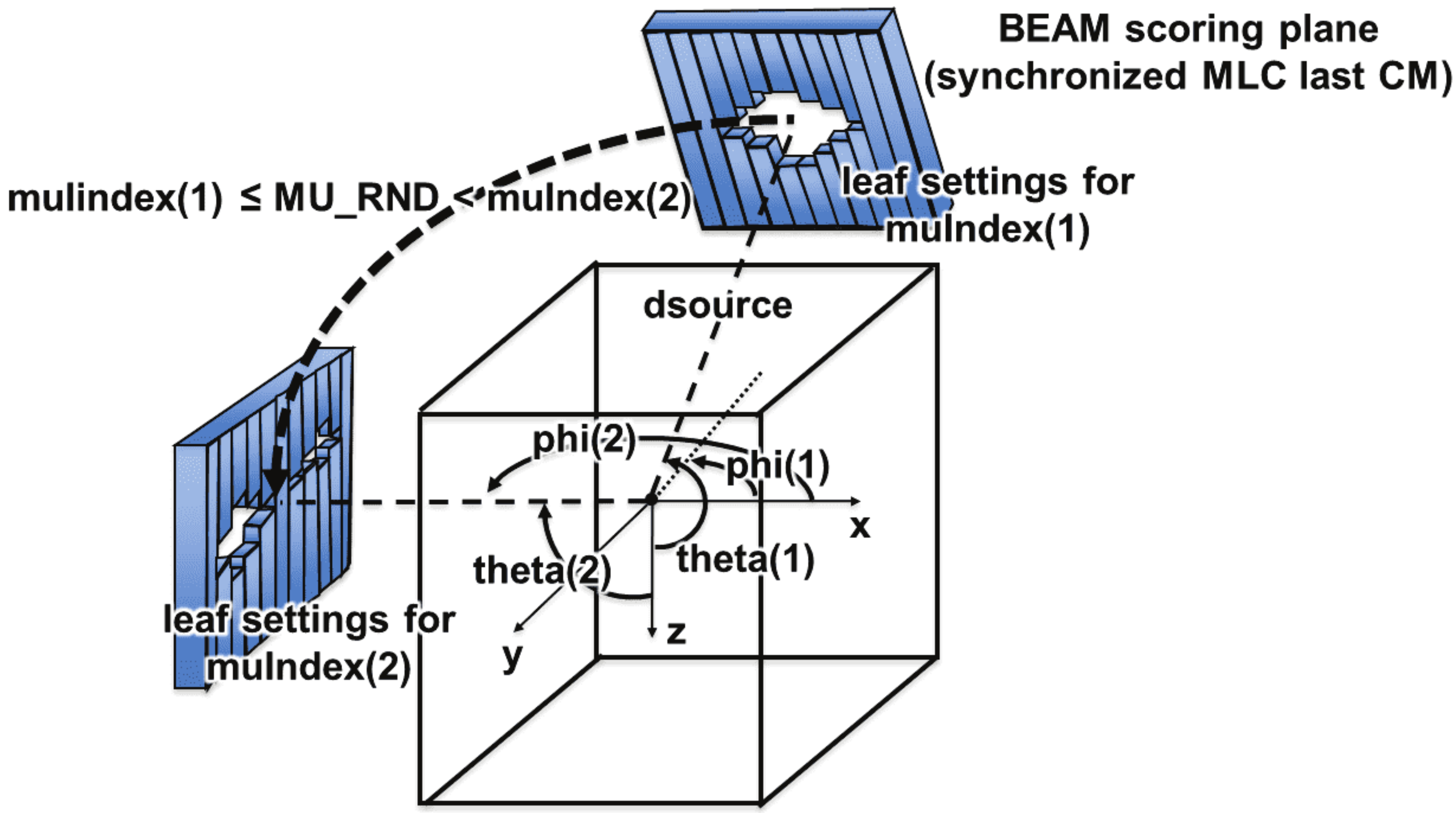
```
make library
```

- `beam_lib.mortran` replaces `beam_main.mortran` in the `mortjob.mortran` for your accelerator.

- `beam_lib.mortran`
  1. Defines particle source “container"
  2. Redefines phase-space macros so that particle data is written to container and not a phase-space file
  3. Defines `beamlib_sample`.
     - Called when the container is empty
     - Runs the BEAM simulation until there is a particle in the container
  4. Calls required subroutines at end of run.

Synchronized phase-space (source 20)

- Simulates continuous motion of the source plane between user-defined control points
- Developed by Tony Popescu and Julio Lobo (PMB 55 (2010) 4431–4443)



Other inputs for synchronized phase-space source

If directional bremsstrahlung splitting (DBS) was used to generate the phase-space:

- Same as for phase-space source (2,8)

`e_split`: charged particle splitting

- Same as for phase-space source (2,8)

Other inputs for synchronized phase-space source (cont.)

`FILNAM`: the phase-space file (path & extension)

`nset`: the number of control points

- Control points define the beginning and end points of ranges of:
  - incident direction angles
  - SSDs
  - isocentre coordinates
- Continuous motion is simulated between control points
- $2 \leq \texttt{nset} \leq \texttt{MAXANG}$, where `MAXANG` is defined in `$EGS_HOME/dosxyznrc/dosxyznrc_user_macros.mortran`

Control point parameters

For $i = 2, ..., \texttt{nset}$, input the following parameters:

- `xiso(i)`, `yiso(i)`, `ziso(i)`: isocentre coordinates
- `theta(i)`, `phi(i)`, `phicol(i)`: source direction
- `dsource(i)`: source to isocentre distance
- `muIndex(i)`: monitor unit index in $[0, 1]$, defining the fraction of the total number of incident particles delivered up to control point i.
  - $\texttt{muIndex}(i) \geq \texttt{muIndex}(i-1)$
  - $\texttt{muIndex}(1) = 0.0$ and $\texttt{muIndex}(\texttt{nset}) = 1.0$

Control point parameters (cont.)

An simple example input segment for nset=7 is:

```
# xiso, yiso, ziso, theta, phi, phicol, dsource, muIndex
0,      0,    -9,   90,    0,   0,      15,      0.0
0,      0,    -9,   90,    360, 0,      15,      0.1
0,      0,    -5,   90,    0,   45,     15,      0.1
0,      0,    -5,   90,    360, 45,     15,      0.5
0,      0,     5,   90,    0,   90,     15,      0.5
0,      0,     5,   90,    360, 90,     15,      0.9
0,      0,     9,   90,    360, 135,    15,      1.0
```

The first two control points ($\texttt{muIndex}(1) = 0.0$ and $\texttt{muIndex}(2) = 0.1$) define 10% of incident particles, with `phi` evenly distributed over $[0, 360°]$

MU_RND per incident particle

- Each incident particle is assigned MU_RND $\epsilon[0, 1]$
- MU_RND is compared with muIndex(i) to determine incident geometry
- If $\texttt{muIndex}(i-1) \leq \texttt{MU\_RND} < \texttt{muIndex}(i)$, then parameters are interpolated:

$$\texttt{PARAM} = \texttt{PARAM}(i-1) + \frac{\texttt{PARAM}(i) - \texttt{PARAM}(i-1)}{\texttt{muIndex}(i) - \texttt{muIndex}(i-1)}(\texttt{MU\_RND} - \texttt{muIndex}(i-1)) \quad (1.2)$$

where PARAM is the value of a source parameter (xiso, yiso, ziso, theta, phi, etc.)

Note: Generating MU_RND per incident particle means that particles arising from the *same primary history* will be incident from different source orientations

4D phase-space output

`i_muidx_out`: whether to output MU

- Set `i_muidx_out` to 1 to include the fractional monitor unit index in phase-space output
- `i_phsp_out` must be set to 1 or 2
- Outputs a 4D IAEA format phase-space, containing particles leaving the phantom geometry

Sync PHSP: Intermediate shared library

`the_shared_lib`

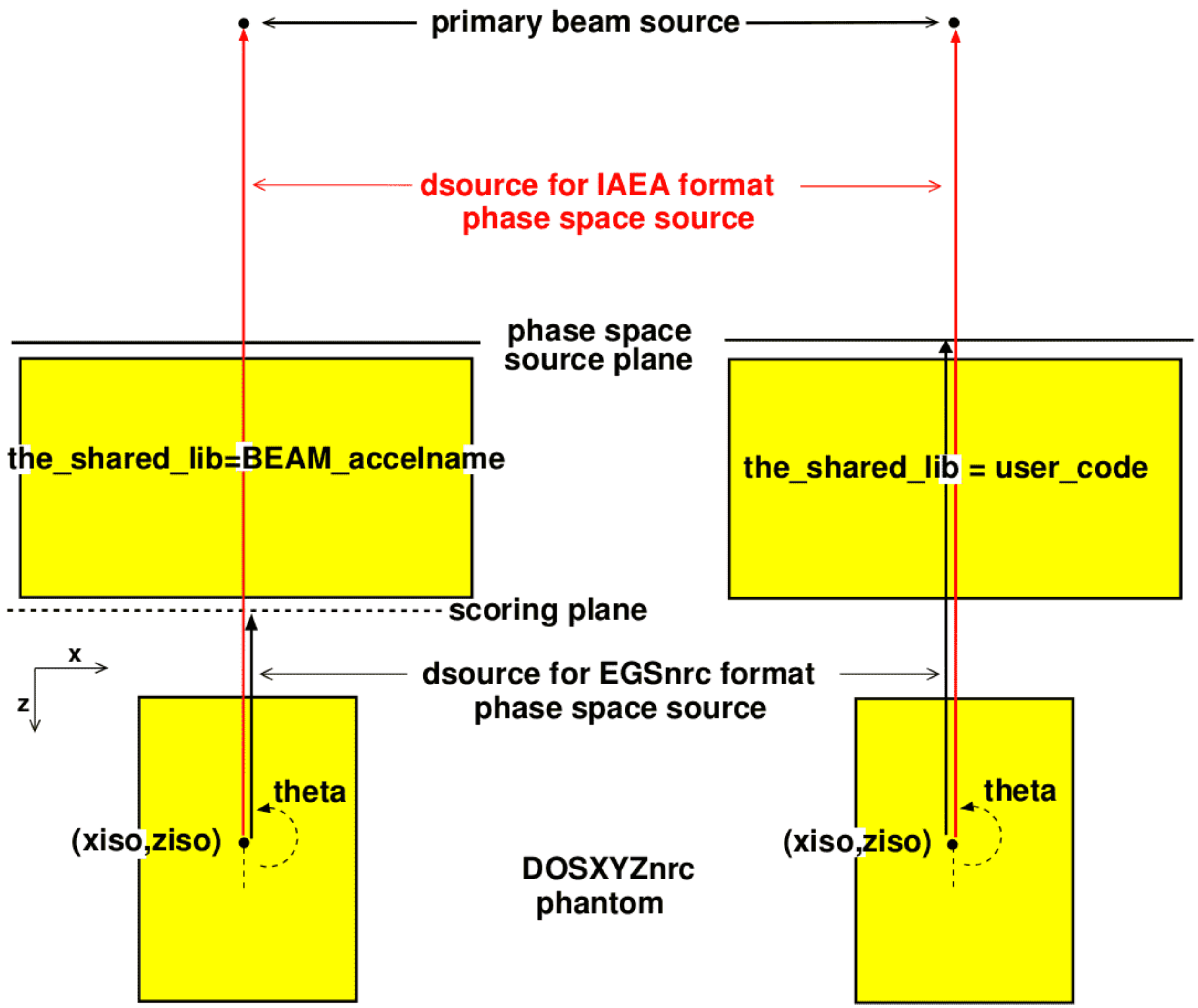
- The name of the BEAM accelerator (`BEAM_accelname`) or non-EGSnrc MLC simulation code compiled as a shared library

`the_input_file`

- Input file for the BEAMnrc or non-EGSnrc code defining the shared library geometry interposed between the phase-space source and DOSXYZnrc phantom
- For BEAMnrc, the file must be in `$EGS_HOME/BEAM_accelname` and specify phase-space output at the bottom of the accelerator

Any BEAMnrc synchronized CMs are automatically passed the MU index of each particle (`MU_RND`)

Sync PHSP: Intermediate shared library (cont.)



Sync PHSP: Synchronized intermediate BEAMnrc shared library

- Particle data at the output plane is stored in an internal array for eventual DOSXYZnrc transport
- Do not specify the same phase-space source in BEAMnrc as in DOSXYZnrc for source 20!
  - Some versions of Fortran fail by a file opening conflict
  - A **dummy source** of any type can be input, because it is not used
- The `PEGS4` data used in the accelerator simulation is not an input variable and must be the same as that used in DOSXYZnrc

Sync PHSP: Synchronized intermediate non-EGSnrc shared library

- We provide an example of non-EGSnrc shared library integration in `$HEN_HOUSE/cutils/load_vculib.c`

- It is designed for `particleDMLC`, developed at Virginia Commonwealth University (VCU) by Lobo and Popescu

- DOSXYZnrc interfaces with the shared library by three functions:

1. `initvcu(char *the input file, float *survival ratio)`
    - Reads the input file
    - Generates an estimate of the survival ratio of particles between the phase-space source and the bottom of the application geometry (usually an MLC). This is used to calculate `NRCYCL`

Synchronized non-EGSnrc shared library (cont.)

2. `vculib_sample(float *e, float *x, float *y, float *z, float *u, float *v,float *w, float *wt, int *iq, int *latch, int *nhist, int *more_in_container)`
   - Runs the code and samples particles reaching the bottom of the geometry
   - Returns the phase-space data for a particle, the number of primary histories to run at that point (`nhist`), and whether or not there are particles remaining in the container (`more_in_container`)
3. `vculib finish()`
   - Called at the end of simulation to close files, free memory, etc.

Sync PHSP: Calibration runs for shared libraries

`calflag`: whether to do calibration run

- On by default if shared library is present
- Determines the number of times to recycle (`NRCYCL`)
- Accounts for the ratio of the number of particles emerging from the library geometry to the number of indicent particles (survival ratio)
- Set `calflag`=1 to skip the calibration run
  - The calibration can consume significant CPU time
  - If there are sufficient particles in the phase-space source such that restarting will not be an issue, set `calflag`=1

Synchronized BEAMnrc simulation (source 21)

Similar inputs to source 20 (`nset`, `i_muidx_out`, etc.)

`the_beam_code`

- The name of the BEAM accelerator (`BEAM_accelname`)

`the_input_file`

- Input file for the BEAMnrc, in `$EGS_HOME/BEAM_accelname` and specifying a scoring plane

`the_pegs_file`

- The `PEGS` data to be used in the BEAMnrc simulation

Intermediate shared library

`the_vcu_code`

- The non-EGSnrc MLC simulation code compiled as a shared library

`the_vcu_input_file`

- Defines a geometry between the bottom of the treatment head and DOSXYZnrc phantom

MU_RND per primary history

- Unlike source 20, MU_RND is chosen only for each primary history (not for each incident particle)
- Thus, the muIndex(i) for the control points are traceable to actual monitor units in the treatment head

DOSXYZnrc advanced sources are powerful!

- Source `2, 8`: Phase-space
- Source `9, 10`: BEAMnrc shared library
- Source `20`: Synchronized phase-space
  - Intermediate shared library possible
- Source `21`: Synchronized BEAMnrc shared library
  - Intermediate shared library possible