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 24

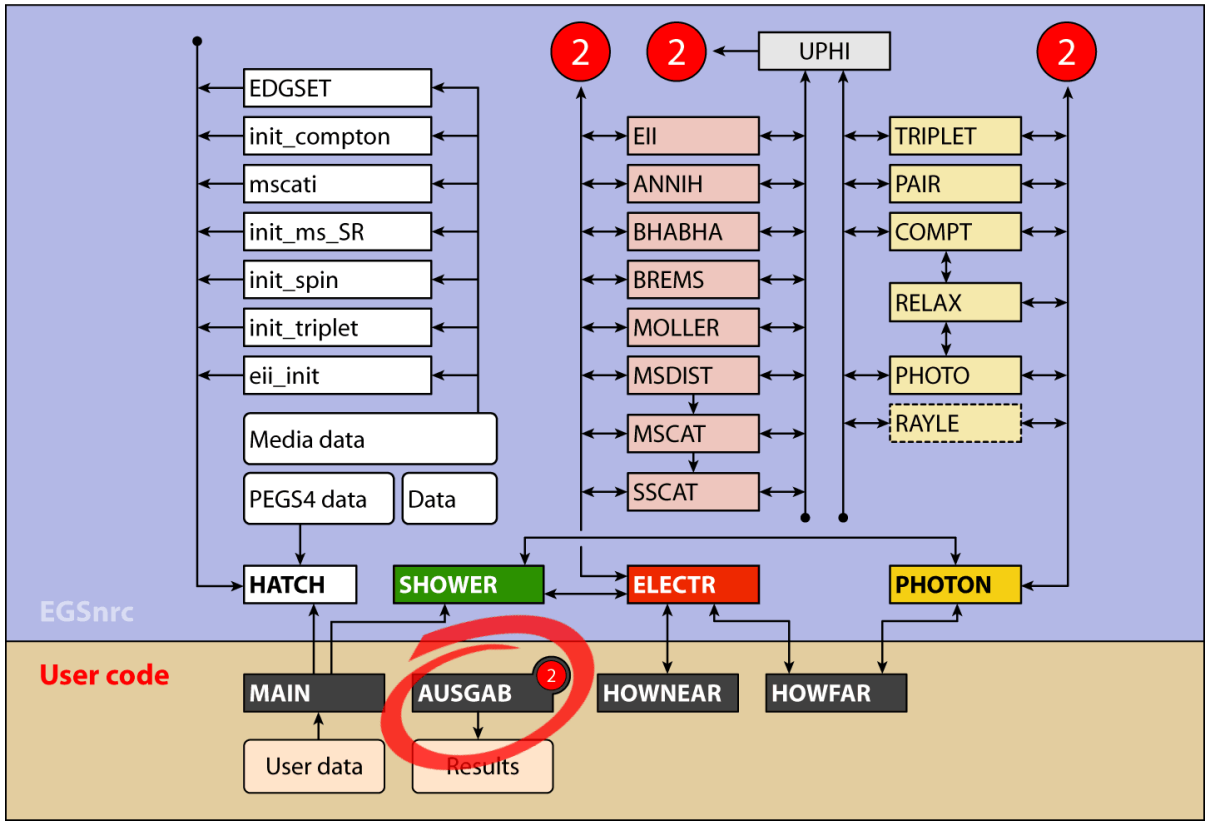
EGSnrc scoring and egs++ ausgab objects

Ernesto Mainegra-Hing

Metrology Research Centre
National Research Council Canada

EGSnrc AUSGAB routine

The AUSGAB routine (from the German *ausgeben*, meaning *to output*) is a user-supplied callback which is invoked at critical points during the particle shower, such as before and after any **interaction**, particle **transport**, or particle **termination** (because it is leaving the geometry or its energy falls below an energy cutoff).



AUSGAB is called with argument IARG

- The `IARG` argument to `AUSGAB(IARG)` indicates what triggered the call.
- `IARG` can take on 31 integer values values from `IARG=0` to `IARG=30`, although only `0, 1, 2, 3, 4` are turned on by default in EGSnrc.

TABLE 1.1 The five default AUSGAB triggers in EGSnrc

IARG	What triggered the AUSGAB call
0	Particle about to be transported by a distance TVSTEP.
1	Particle about to be discarded because its energy is below the cutoff`ECUT` (charged particles) or `PCUT` (photons)—but its energy is larger than the corresponding PEGS cutoff `AE` or `AP`.
2	Particle about to be discarded because its energy is below both `ECUT` and `AE` (or `PCUT` and `AP`).
3	Particle about to be discarded at the user's request (usually in `HOWFAR`, or by range rejection).
4	The difference between the energy of the incident particle and all of the final products is being deposited locally. This energy is due to sub-threshold relaxation events.

Additional IARG values (off by default)

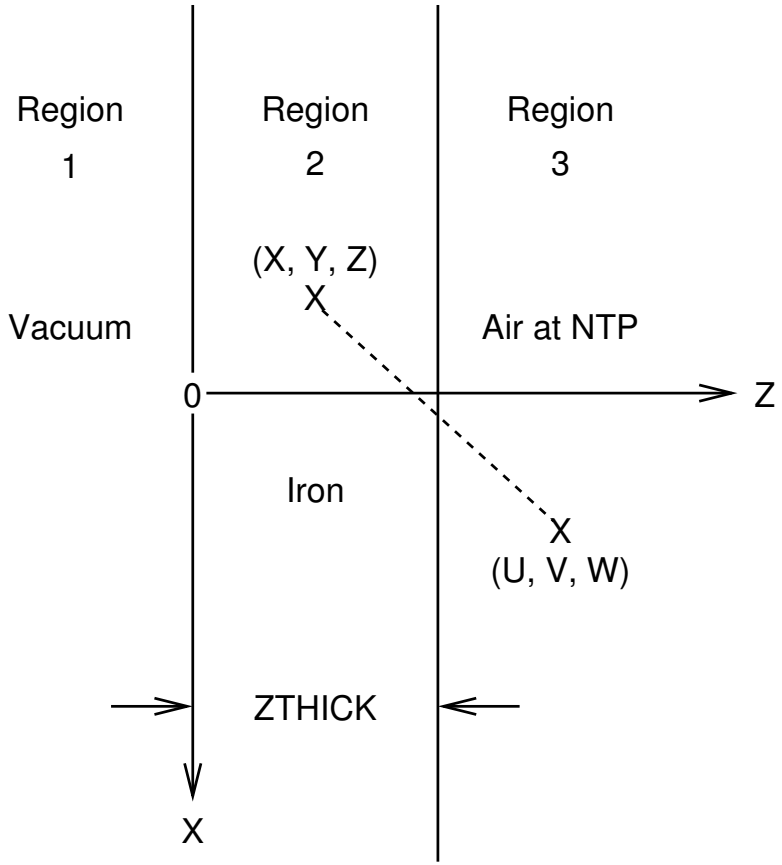
The remaining 26 IARG values must be *turned on* by setting to 1 the associated entry in the IAUSFL array.

TABLE 1.2 A few other non-default AUSGAB triggers in EGSnrc

IARG	IAUSFL	What triggered the AUSGAB call
5	6	Particle was transported by distance TVSTEP.
6	7	A bremsstrahlung interaction about to occur (calling BREMS from within ELECTR).
7	8	Just returned to ELECTR routine from a BREMS call.
...	...	
14	15	A positron has just annihilated at rest.
17	18	A Compton interaction about to occur (calling COMPT from within PHOTON.
18	19	Just returned to PHOTON routine from a COMPT call.
...	...	

Example 1: a three region geometry

As an example of how to write an AUSGAB subprogram, consider the geometry depicted below, comprising only three regions separated by the vertical planes. Suppose that we are only interested in photons that go from Region 2 into Region 3.



Example 1: a three region geometry

The AUSGAB routine that will accomplish this is listed below. In this example we print out the stack variables as well as IARG.

```
SUBROUTINE AUSGAB(IARG);

COMIN/STACK/;

"only output information for photons that are discarded"
"(by the user) in region 3"

IF (IARG = 3 & IQ(NP) = 0 & IR(NP) = 3) [
   OUTPUT E(NP),X(NP),Y(NP),Z(NP),U(NP),V(NP),W(NP),
          IQ(NP),IR(NP),IARG;  (7G15.7,3I5);
]

RETURN;
END;
```

Example 2: ion chamber dosimetry in egs++

The meaning of `iarg` carries over to applications built with egs++ such as the `cavity` app:

```
int ausgab (int iarg) {
    int np = the_stack->np-1;
    ...
    if (iarg <= 4) {
        int ir = the_stack->ir[np]-2;
        if (ir >= 0 && is_cavity[ig][ir]) {
            EGS_Float aux = the_epcont->edep*the_stack->wt[np];
            if (aux > 0) {
                dose->score(ig,aux);
            }
        }
     }
}
```

The egs++ ausgab object (AO)

- `EGS_AdvancedApplication` calls the `EGS_Application` method `userScoring(iarg)` instead of `ausgab(iarg)`. This method has a generic implementation to first call the scoring methods of ausgab objects (`EGS_AusgabObject` class) registered with the application and to only then call the `ausgab()` function.

- AOs are *defined by input* as follows:

```
:start ausgab object definition:
   :start ausgab object:
     ...
   :stop ausgab object:
     ...
:stop ausgab object definition:
```

- AOs *created by input* avoid the need to code, which is an attractive option for new adopters of EGSnrc.

Particle Tracks

- First EGSnrc ausgab object (AO), developed by Iwan Kawrakow, based on Georgi Gerganov's track scoring mechanism implemented in `egs_pet` application
- input file definition
- Tracks stored in a `*.ptracks` file
- Track visualization with `egs_view` using:

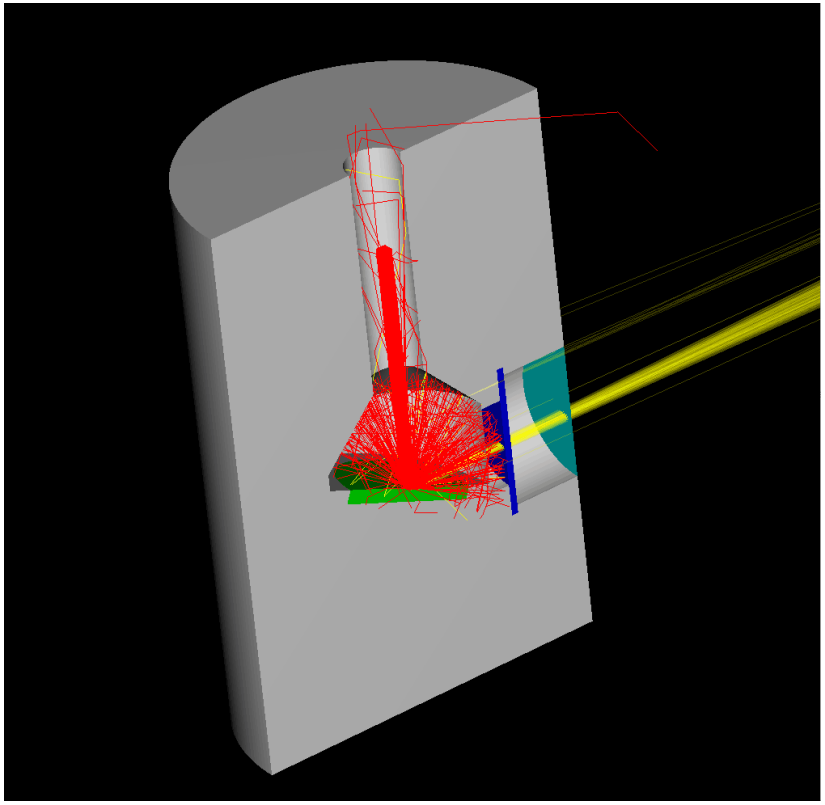
```
egs_view input_file tracks_file
```

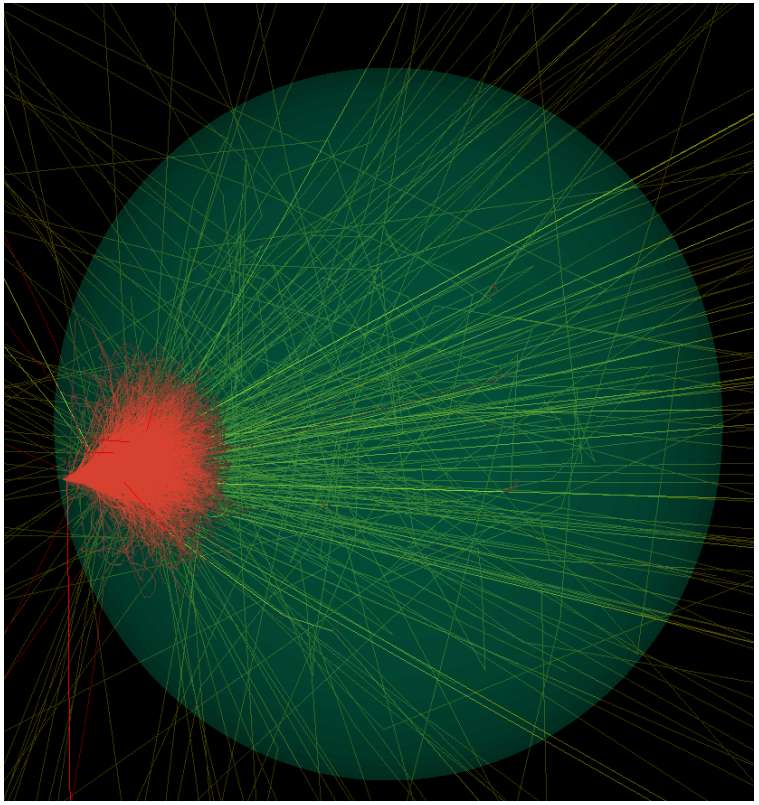
Track scoring object definition

```
:start ausgab object definition:
   :start ausgab object:
        library = egs_track_scoring
        name    = some_name
        score photons   = yes or no # optional, yes by default
        score electrons = yes or no # optional, yes by default
        score positrons = yes or no # optional, yes by default
        start scoring   = event_number # optional, 0 by default
        stop  scoring   = event_number # optional, 1024 by default
        buffer size     = size         # optional, 1024 by default
        file name addition = some_string # optional, empty by default
   :stop ausgab object:
:stop ausgab object definition:
```

Particle Tracks Visualization

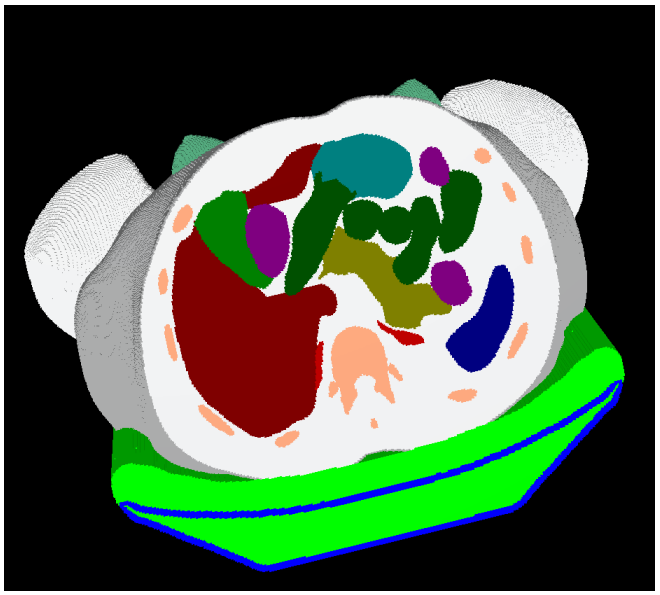
- Tracks stored in a `*.ptracks` file
- Visualization with `egs_view` using: `egs_view input_file tracks_file`





Dose scoring

Motivation: Despite the availability of dose scoring applications such as DOSRZnrc and DOSXYZnrc, calculating dose to specific media such as dose to organs in a voxelized geometry is not possible without modifying these applications.



Purpose: generic dose scoring AO that could be used with **any** C++ EGSnrc application and provide both, dose to arbitrary regions and to all media in the geometry.

Available in EGSnrc since 2013 release.

Dose scoring AO definition

```
:start ausgab object definition:
   :start ausgab object:
      library      = egs_dose_scoring
      name         = some_name
      medium dose  = no  # (default)
      region dose  = yes # (default)
      volume       =  v1 ... vn  # defaults to v1 or 1 cm^3
      dose regions = ir1 ... irn # individual regions
   :stop ausgab object:
:stop ausgab object definition:
```

- Volume needed to compute the mass for each dose scoring zone.
- A unique volume indicates identical volumes for all scoring regions
- If volume for scoring region missing, first entry or a default of 1 g/cm^3 is used.
- Option to calculate dose deposited **in each medium**. Useful for organ dose calculations.

Dose scoring AO definition: groups of regions

```
:start ausgab object definition:
   :start ausgab object:
      library      = egs_dose_scoring
      name         = some_name
      medium dose  = no  # (default)
      region dose  = yes # (default)
      volume       =  v1 ... vn  # defaults to v1 or 1 cm^3
      dose start region = iri_1 ... iri_N
      dose stop region  = ire_1 ... ire_N
   :stop ausgab object:
:stop ausgab object definition:
```

Enter the same number of volumes as groups of regions or individual volumes for each region. Defaults to first entry or 1 g/cm^3.

Dose scoring AO definition: normalization

```
:start ausgab object definition:
   :start ausgab object:
      library      = egs_dose_scoring
      name         = some_name
      medium dose  = no  # (default)
      region dose  = yes # (default)
      volume       =  v1 ... vn  # defaults to v1 or 1 cm^3
      dose regions = ir1 ... irn # individual regions
      # or alternatively: provide groups of consecutive regions
      # dose start region = iri_1 ... iri_N
      # dose stop region  = ire_1 ... ire_N
      normalization = C
   :stop ausgab object:
:stop ausgab object definition:
```

One can also provide an arbitrary dose normalization constant.

Dose scoring AO screen output

```
==> Summary of media dosimetry (per fluence)
  medium                Edep/[MeV*cm2]                  D/[Gy*cm2]
-------------------------------------------------------------------------
ADIPOSETISSUE     9.3196e-01 +/- 0.193  % 3.4037e-16 +/- 0.193  %
MUSCLETISSUE      3.3014e+00 +/- 0.113  % 3.8776e-16 +/- 0.113  %
SOFTTISSUE        2.3984e+00 +/- 0.153  % 2.5052e-16 +/- 0.153  %
SKINTISSUE        9.3428e-01 +/- 0.178  % 8.0702e-16 +/- 0.178  %
LUNGTISSUE        8.7853e-01 +/- 0.242  % 3.1761e-16 +/- 0.242  %
SPONGIOSA         6.1030e-01 +/- 0.312  % 4.5978e-16 +/- 0.312  %
CORTICALBONE      2.2731e+00 +/- 0.163  % 1.2130e-15 +/- 0.163  %
CARTILAGE         8.3502e-02 +/- 0.789  % 2.2347e-16 +/- 0.789  %
-------------------------------------------------------------------------
```

- Results output to the screen by default
- Might be enough for few regions and media
- Output can be redirected to a file, e.g., using
  `app -i inp_file -p pegs4_file | tee output_file`

Dose scoring AO screen output

```
==> Summary of region dosimetry (per fluence)
     ir            medium  rho (g/cm^3)  Volume (cm^3)   Edep (MeV*cm^2)              D (Gy*cm^2)
-------------------------------------------------------------------------------------------------------
4500000  MUSCLETISSUE       1.05000000       0.002160    3.6855e-05  +/-  8.693  %    2.6032e-12  +/-  8.693  %
4500001  MUSCLETISSUE       1.05000000       0.002160    2.2087e-05  +/-  11.605 %    1.5601e-12  +/-  11.605 %
4500002  MUSCLETISSUE       1.05000000       0.002160    1.5648e-05  +/-  14.753 %    1.1053e-12  +/-  14.753 %
4500003  SOFTTISSUE         1.05000000       0.002160    1.0831e-05  +/-  17.558 %    7.6507e-13  +/-  17.558 %
4500004  SOFTTISSUE         1.05000000       0.002160    1.0602e-05  +/-  17.348 %    7.4887e-13  +/-  17.348 %
4500005  SOFTTISSUE         1.05000000       0.002160    1.2167e-05  +/-  17.724 %    8.5943e-13  +/-  17.724 %
4500006  MUSCLETISSUE       1.05000000       0.002160    7.7660e-06  +/-  22.672 %    5.4855e-13  +/-  22.672 %
4500007  MUSCLETISSUE       1.05000000       0.002160    1.0582e-05  +/-  23.083 %    7.4746e-13  +/-  23.083 %
4500008  MUSCLETISSUE       1.05000000       0.002160    7.9323e-06  +/-  22.728 %    5.6030e-13  +/-  22.728 %
4500009  MUSCLETISSUE       1.05000000       0.002160    9.7780e-06  +/-  20.923 %    6.9067e-13  +/-  20.923 %
4500010  MUSCLETISSUE       1.05000000       0.002160    6.7862e-06  +/-  23.265 %    4.7934e-13  +/-  23.265 %
7500000  Air                0.00120500       0.002160    0.0000e+00  +/-  100.000%    0.0000e+00  +/-  100.000%
7500001  Air                0.00120500       0.002160    1.1904e-07  +/-  100.000%    7.3270e-12  +/-  100.000%
7500002  Air                0.00120500       0.002160    1.5650e-07  +/-  100.000%    9.6326e-12  +/-  100.000%
7500003  SKINTISSUE         1.09000000       0.002160    3.4521e-05  +/-  8.618  %    2.3489e-12  +/-  8.618  %
7500004  SKINTISSUE         1.09000000       0.002160    2.4548e-05  +/-  11.392 %    1.6703e-12  +/-  11.392 %
7500005  ADIPOSETISSUE      0.95000000       0.002160    1.1480e-05  +/-  16.333 %    8.9623e-13  +/-  16.333 %
7500006  ADIPOSETISSUE      0.95000000       0.002160    7.5614e-06  +/-  19.027 %    5.9032e-13  +/-  19.027 %
7500007  ADIPOSETISSUE      0.95000000       0.002160    4.7755e-06  +/-  25.912 %    3.7282e-13  +/-  25.912 %
7500008  LUNGTISSUE         0.26000000       0.002160    1.8732e-06  +/-  40.567 %    5.3435e-13  +/-  40.567 %
7500009  LUNGTISSUE         0.26000000       0.002160    3.4283e-06  +/-  28.968 %    9.7795e-13  +/-  28.968 %
7500010  LUNGTISSUE         0.26000000       0.002160    1.4758e-06  +/-  48.542 %    4.2099e-13  +/-  48.542 %
-------------------------------------------------------------------------------------------------------


==> Summary of media dosimetry (per fluence)
  medium              Edep/[MeV*cm2]              D/[Gy*cm2]
-----------------------------------------------------------------------
Air              5.0402e-02 +/- 0.211  % 1.5564e-12 +/- 0.211  %
ADIPOSETISSUE    9.2948e-01 +/- 0.061  % 1.5716e-13 +/- 0.061  %
MUSCLETISSUE     3.2943e+00 +/- 0.036  % 1.7913e-13 +/- 0.036  %
SOFTTISSUE       2.3983e+00 +/- 0.048  % 1.1598e-13 +/- 0.048  %
SKINTISSUE       9.2520e-01 +/- 0.057  % 3.6999e-13 +/- 0.057  %
LUNGTISSUE       8.8238e-01 +/- 0.076  % 1.4769e-13 +/- 0.076  %
SPONGIOSA        6.1227e-01 +/- 0.099  % 2.1355e-13 +/- 0.099  %
CORTICALBONE     2.2780e+00 +/- 0.051  % 5.6280e-13 +/- 0.051  %
CARTILAGE        8.3856e-02 +/- 0.250  % 1.0390e-13 +/- 0.250  %
-----------------------------------------------------------------------
```

Dose scoring AO file output

If one of the geometries in the simulation is an `EGS_XYZGeometry` :

- Option to output the dose in the voxels of this geometry to a `.3ddose` file.
- Familiar to users of `DOSXYZnrc` (see DOSXYZnrc users manual for more details).
- In this case, masses of each voxel are available through the member function getVolume of `EGS_XYZGeometry`.

Dose scoring AO file output: example

An example input to obtain a `.3ddose` file for an `EGS_XYZGeometry` is given below:

```
:start ausgab object:
  library            = egs_dose_scoring
  name               = some_name
  medium dose = no
  region dose = no
  :start output dose file:
     geometry name = must name an EGS_XYZGeometry
     file type = 3ddose (currently the default and only format available)
  :stop output dose file:
:stop ausgab object:
```

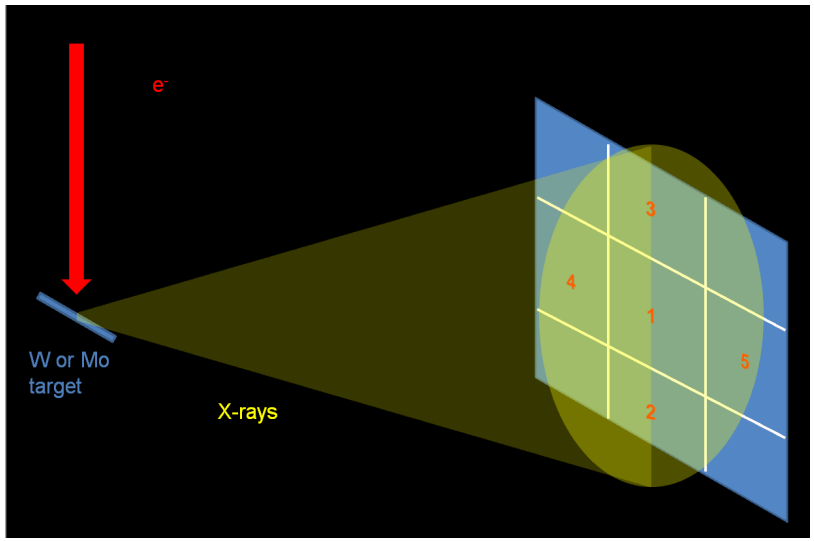
Output is to the file `some_name.3ddose`.

Note: Region doses for the simulation geometry have been turned off to avoid outputting the dose for every voxel in the `EGS_XYZGeometry` to the screen/.egslog file.

Fluence scoring

Motivation: AAPM TG-195

MC Reference Data Sets for Imaging Research Case 6: Estimate x-ray spectrum for a 30 kV Mo and a 100 kV W x-ray beam and energy fluence at the scoring plane in different ROIs.

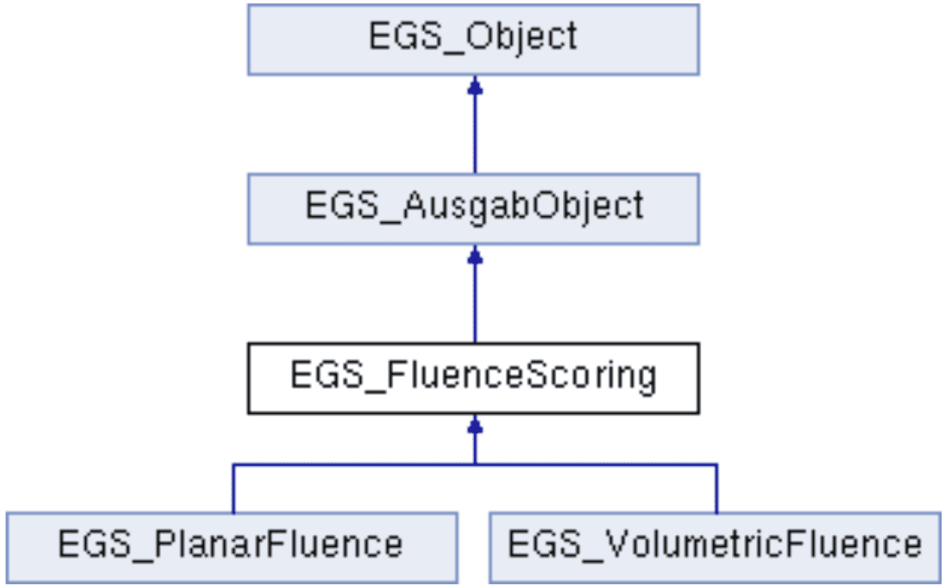


Purpose: generic fluence scoring AO that could be used with **any** C++ EGSnrc application.

Available in EGSnrc since 2023 release.

Fluence scoring: types

- Base class
  - Basic ingredients such as energy grid, particle type, scoring regions,
  - Flag to score primary fluence.
  - Provides the method for defining primary and secondary particles.
- `Planar`: scoring on circular or rectangular fields
- `Volumetric`: scoring in arbitrary geometrical regions



Note: If fluence for more than one particle type is desired, multiple AOs are required.

Fluence scoring AO definition (common inputs)

```
:start ausgab object:
   name    = id-string            # Arbitrary identifying string
   library = egs_fluence_scoring  # Library name
   type    = planar               # or volumetric
   scoring particle = photon, or electron, or positron
   score primaries = yes or no    # Defaults to `no`.
   score spectrum  = yes or no    # Defaults to `no`.
   verbose         = yes or no    # Defaults to `no`. Provide more details.
   normalization   = norm         # User-requested normalization. Defaults to 1.

   #########
   # If scoring spectrum, define energy grid
   # Default: 128 linear energy bins between 1 keV and 1 MeV
   #########
   :start energy grid:
     number of bins = nbins
     minimum kinetic energy = Emin
     maximum kinetic energy = Emax
     scale = linear or logarithmic # Defaults to `linear`.
   :stop energy grid:

:stop ausgab object:
```

Fluence scoring AO: primary particle definition

For photons:

- Flag scattered photons, secondaries, and relaxation particles as secondaries

For charged particles:

- DEFAULT:
  - Secondary charged particles defined as those resulting from charged particle interactions, atomic relaxations following EII, brems and annihilation photons.

Fluence scoring AO: primary particle definition

For charged particles:

- OPTIONAL:
  - Set ‘source particle‘ to ’photon’ in the input file to not flag brems photons so that when scoring charged particle fluence in photon beams, first generation e- are primaries. This is implicit for photon interactions when scoring charged particle fluence. But one must explicitly account for that during brems events.

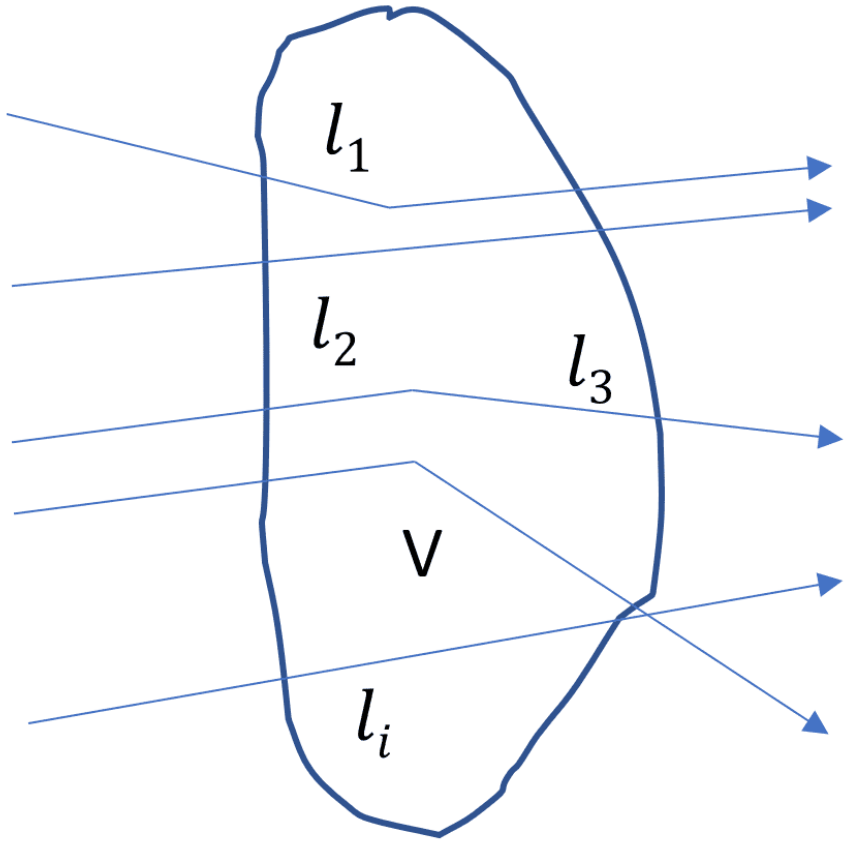
```
:start ausgab object:
   .
   score primaries = yes          # Defaults to `no`.
   .
   # Currently used ONLY for photons and bremsstrahlung targets.
   source particle  = photon # or electron, or positron
                             # Defaults to source-emitted particles if only one type,
                             #  or to scoring particle if multiple particle types.
                             # Useful for bremsstrahlung targets and radioactive sources.
   .
:stop ausgab object:
```

Volumetric fluence scoring

A linear track-length estimator used to compute fluence in arbitrary geometrical regions.

Fluence definition:

$$\Phi = \frac{1}{V} \sum_i l_i$$



Volumetric fluence scoring definition: by regions

```
:start ausgab object:
    name    = id-string            # Arbitrary identifying string
    library = egs_fluence_scoring  # Library name
    type    = volumetric           # Score in a volume
    scoring particle = photon, or electron, or positron

    :start volumetric scoring:

        scoring regions = ir1 ir2 ... irn # defaults to NONE

        ### Alternatively:
        #start region = iri_1, iri_2, ..., iri_n
        #stop region  = irf_1, irf_2, ..., irf_n
        ###

        volumes = V1, V2, ..., VN # Enter as many as scoring regions. If same number
                                  # of entries as group of regions, assumes groups of
                                  # equal volume regions. If only one entry, assumes
                                  # equal volumes in all regions. Defaults to 1 g/cm^3.

    :stop volumetric scoring:

:stop ausgab object:
```

Volumetric fluence scoring definition: using labels

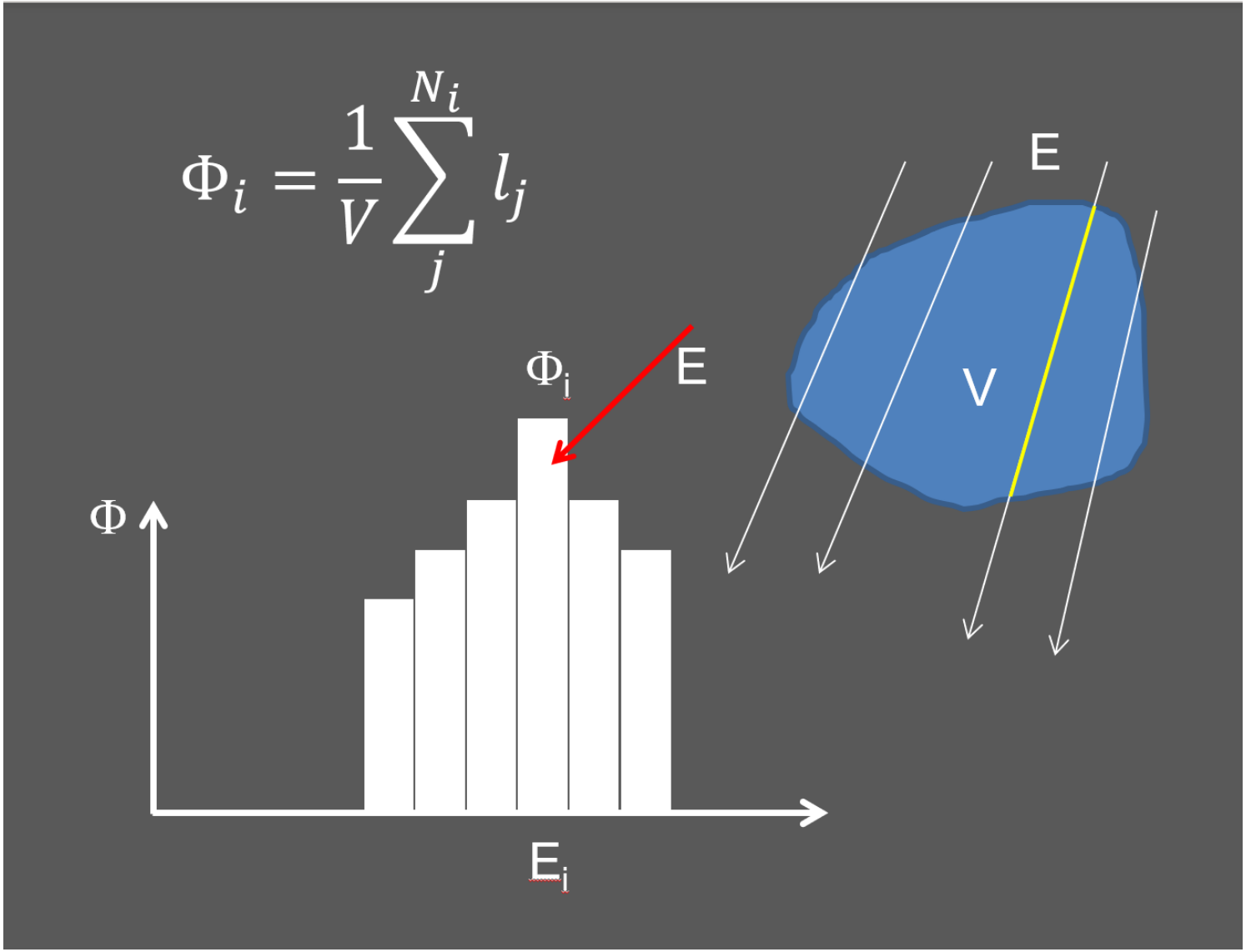
```
:start ausgab object:
    name    = id-string            # Arbitrary identifying string
    library = egs_fluence_scoring  # Library name
    type    = volumetric           # Score in a volume
    scoring particle = photon, or electron, or positron

    :start volumetric scoring:

         # Define contributing regions using labels, use keyword ALL to score in all regions
         scoring regions = label_1 label_2 ... label_n # defaults to NONE


        volumes = V1, V2, ..., VN # Enter as many as scoring regions. If same number
                                  # of entries as group of regions, assumes groups of
                                  # equal volume regions. If only one entry, assumes
                                  # equal volumes in all regions. Defaults to 1 g/cm^3.

    :stop volumetric scoring:

:stop ausgab object:
```

Volumetric charged particle differential fluence methods

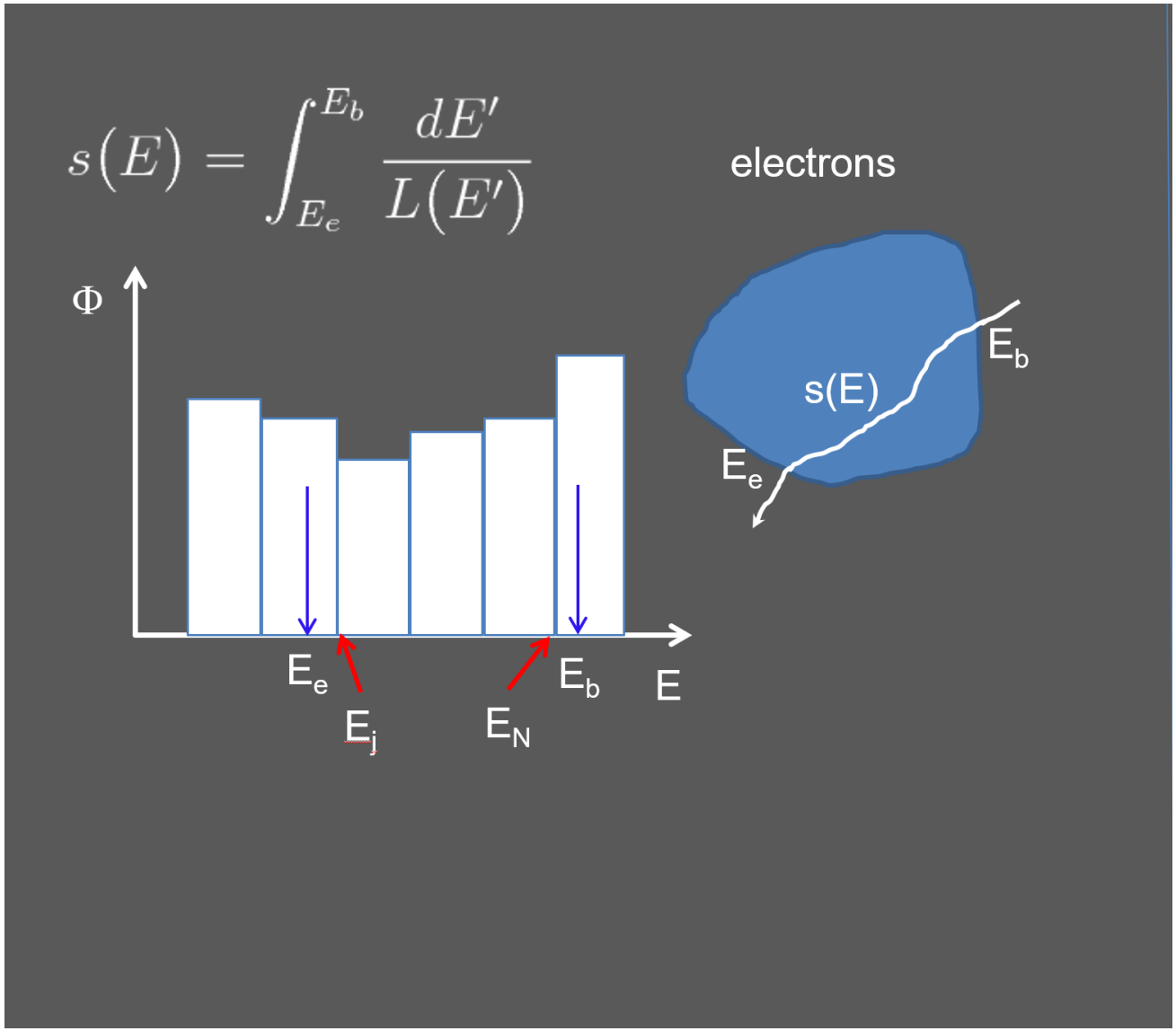
```
:start ausgab object:
    .
    type    = volumetric         # Score in a volume
    scoring particle = electron or positron
    score spectrum  = yes        # Defaults to `no`

    :start volumetric scoring:
        .
        method  = flurz or stpwr or stpwr05 # Charged particle differential fluence
                                            # Defaults to `stpwr`.
            .
    :stop volumetric scoring:
    .
:stop ausgab object:
```

Differential fluence: photons

$$\Phi_i = \frac{1}{V}\sum_{j}^{N_i} l_j$$

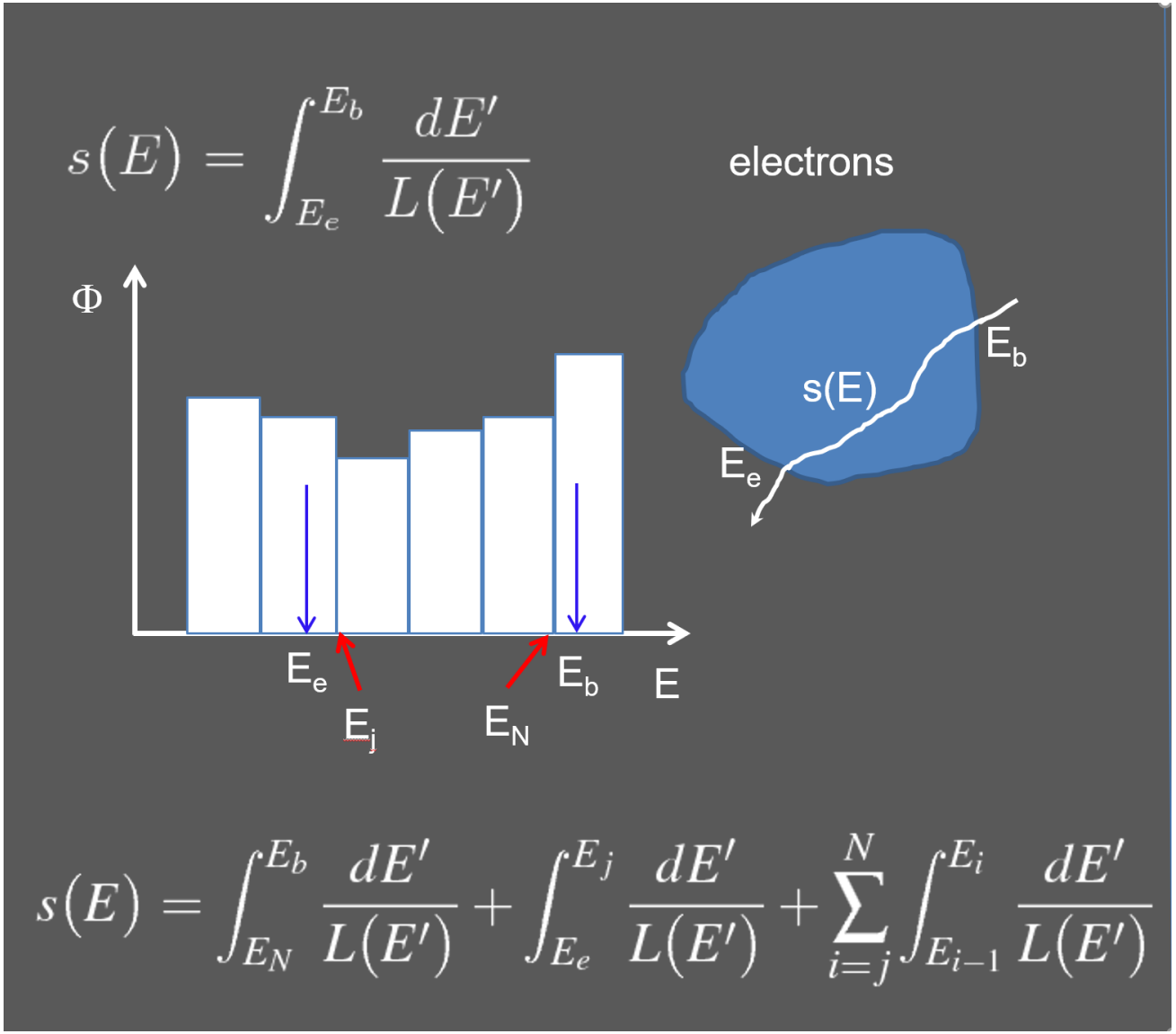


Differential fluence: electrons



Differential fluence: electrons

$$s(E) = \int_{E_e}^{E_b} \frac{dE'}{L(E')}$$

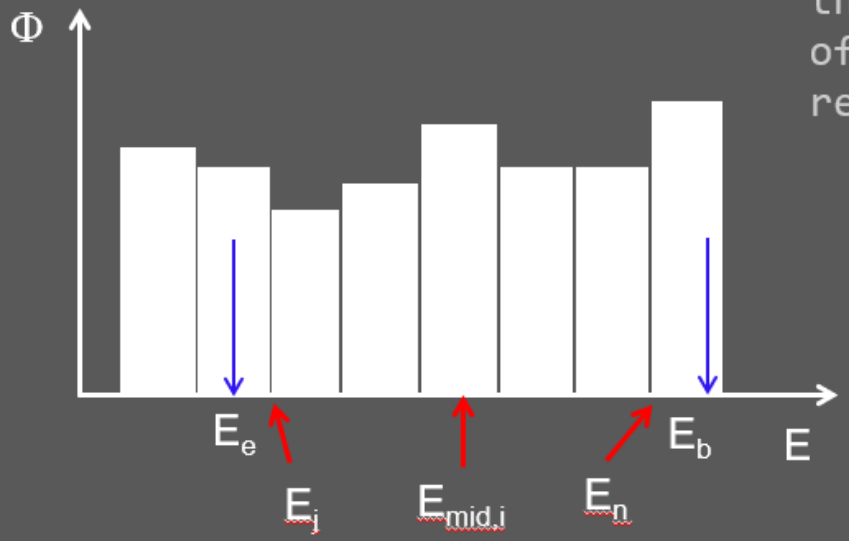


$$s(E) = \int_{E_N}^{E_b} \frac{dE'}{L(E')} + \int_{E_e}^{E_j} \frac{dE'}{L(E')} + \sum_{i=j}^{N} \int_{E_{i-1}}^{E_i} \frac{dE'}{L(E')}$$

Differential fluence: electrons

$$L(E) = a_i + b_i \ln E \quad \text{for} \quad E_i \leq E < E_{i+1}$$

O(3) series expansion of the integral of the inverse of the stopping power with respect to energy.



$$s(E) = \frac{(E_b - E_N)}{L\left(\frac{E_b + E_N}{2}\right)} + \frac{(E_j - E_e)}{L\left(\frac{E_j + E_e}{2}\right)} + \sum_{i=j}^{n} \frac{\Delta E_i}{L(E_{mid,i})}$$

Volumetric charged particle differential fluence methods

```
:start ausgab object:
  .
  :start volumetric scoring:
    #
    # stpwr => Accounts for stopping power variation along the particle's step.
    #          More accurate than method used in FLURZnrc albeit about
    #          about 10% slower in electron beam cases.
    # Uses an O(3) series expansion of the integral of the
    # inverse of the stopping power with respect to energy.
    # Stopping power is represented as a linear interpolation
    # over a log energy grid.
    #
    # stpwrO5 => Uses an O(5) series expansion. Slightly slower.
    #
    # flurz   => FLURZnrc algorithm
    # Path length at each energy interval from energy deposited EDEP
    # and total particle step TVSTEP. Assumes stopping power constancy
    # along the particle's step. Might introduce artifacts if
    # ESTEPE or the scoring bin width are too large.
  :stop volumetric scoring:
:stop ausgab object:
```

Planar fluence scoring

A linear track-length estimator used to compute fluence for either a circular field or a rectangular screen of arbitrary resolution (pixels).

From fluence definition:

$$\Phi = \frac{1}{V}\sum_i l_i$$

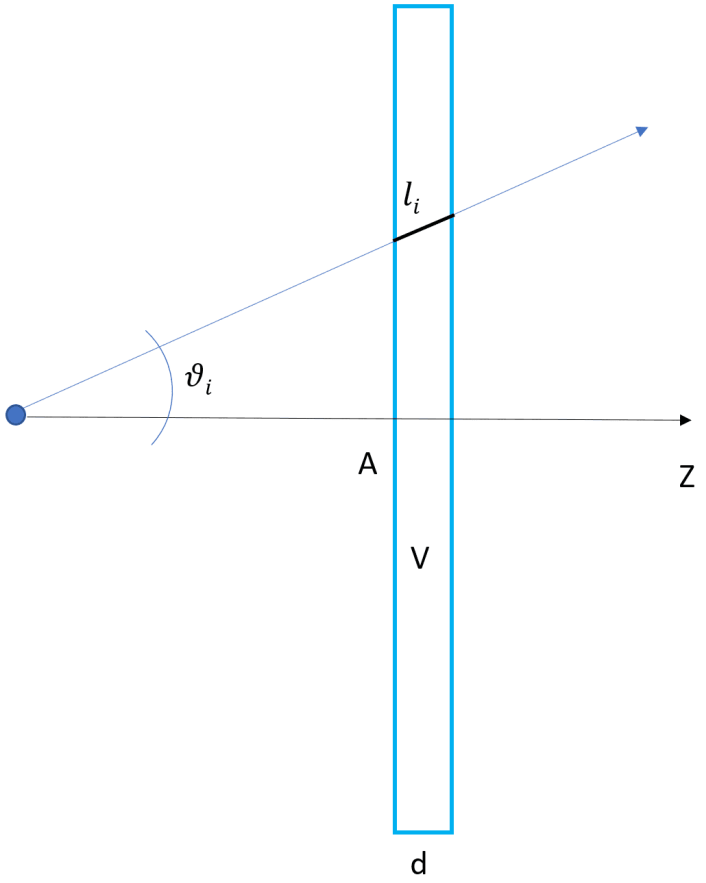
If particle crosses V in a straight line

$$l_i = \frac{d}{\cos\theta}$$

then

$$\Phi = \frac{1}{A}\sum_i \frac{1}{\cos\theta}$$

This is always valid in the limit $d \to 0$



Planar fluence scoring definition (circular field)

```
:start ausgab object:

    name    = id-string            # Arbitrary identifying string
    library = egs_fluence_scoring  # Library name
    type    = planar               # Score on circular or square field
    scoring particle = photon, or electron, or positron

    :start planar scoring:

       # Define contributing regions, recommended for charged particles
       contributing regions = ir1 ir2 ... irn # defaults to ALL regions

       ### Alternatively:
       # start contributing region = iri_1, iri_2, ..., iri_n
       # stop contributing region =  irf_1, irf_2, ..., irf_n
       ###

       scoring circle = x y z R
       scoring plane normal = ux uy uz

    :stop planar scoring:

:stop ausgab object:
```

Planar fluence scoring definition (circular field)

```
:start ausgab object:

    name    = id-string            # Arbitrary identifying string
    library = egs_fluence_scoring  # Library name
    type    = planar               # Score on circular or square field
    scoring particle = photon, or electron, or positron

    :start planar scoring:

       # Define contributing regions using labels,
       # use keyword ALL to include all regions
       contributing regions = label_1 label_2 ... label_n

       scoring circle = x y z R
       scoring plane normal = ux uy uz

    :stop planar scoring:

:stop ausgab object:
```

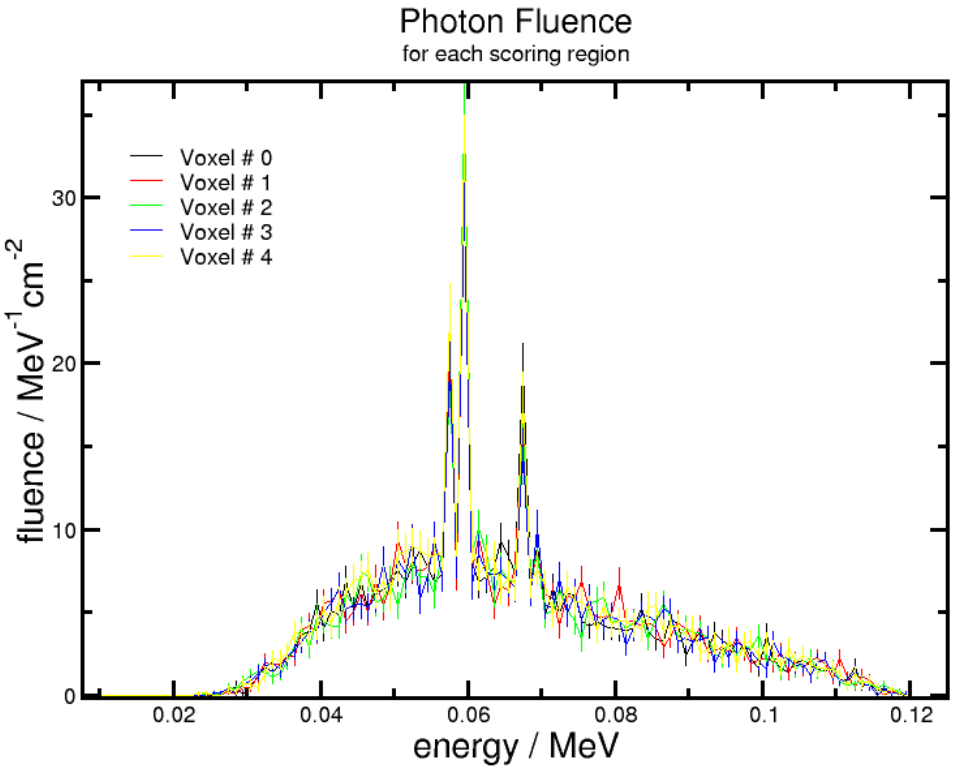
Planar fluence scoring definition (rectangular field)

```
:start ausgab object:
    .
    type    = planar               # Score on circular or square field
    scoring particle = photon, or electron, or positron
    .
    :start planar scoring:
       .
       scoring rectangle = xmin xmax ymin ymax

       resolution = Nx Ny # Number of pixels in X and Y direction

       ##############################################
       # See documentation for EGS_AffineTransform
       ##############################################
       :start transformation:
          rotation = 2, 3 or 9 floating point numbers
          translation = tx, ty, tz
       :stop transformation:
       ##############################################

    :stop planar scoring:

:stop ausgab object:
```

Fluence scoring object output

- Total fluence on each scoring region output to standard output for an interactive execution or the `*.egslog` file if running in parallel.

- Spectrum for each scoring region output to an xmgrace file `input_file_name.agr`



Phase space scoring AO

Scores phase space data for particles using one of two methods:

- Particles exiting and/or entering all surfaces of a predefined geometry. The geometry can be a component of the simulation geometry or coincident with a surface in the simulation geometry.

- Particles on exiting one user-specified region and entering another. The user can specify multiple exit/entry region pairs.

Can be used in any C++ user code by entering the proper input block in the ausgab object definition block.

Note on parallel runs: During a parallel run, each job outputs its phase space data to a file. These phase space files are not added automatically, hence one must either use the addphsp tool or program their own concatenation routine.

Phase space scoring AO: Formats

Phase space data can be scored in one of 2 possible formats:

- EGSnrc format: E,x,y,u,v,wt,latch
  - option to include multiple crossers and their descendents.
- IAEA format: iq,E,[x],[y],[z],u,v,wt,latch,[mu]
  - option of specifying a fixed x, y, and/or z coordinate of the scoring plane/line/point,
  - fixed coordinates not scored for each particle but specified in the header (.IAEAheader) file.
  - option of scoring the synchronization parameter, MU, passed from the source.
  - multiple crossers (and their descendents) NOT scored according to the IAEA protocol to format phase space files.

Phase space scoring AO definition

```
:start ausgab object:
   library                 = egs_phsp_scoring
   name                    = some_name
   output format           = EGSnrc or IAEA
   constant X              = X value (cm) at which all particles are scored (IAEA format only)
   constant Y              = Y value (cm) at which all particles are scored (IAEA format only)
   constant Z              = Z value (cm) at which all particles are scored (IAEA format only)
   particle type           = all, photons, or charged
   score mu                = yes or no [default] (IAEA format only)
   score multiple crossers = no (default) or yes (EGSnrc format only)
   output directory        = name of output directory

   # Use method 1: score particles on entry to/exit from a predefined geometry
       phase space geometry     = name of previously defined geometry
       score particles on       = entry, exit, entry and exit [default]

   # Or method 2: score particles on exiting one region and entering another
   #    from regions              = list of exit region numbers
   #    to regions                = list of entry region numbers
:stop ausgab object:
```

Radiative splitting VRT AO

Increases the efficiency of radiative events by setting EGSnrc internal radiative splitting number. This ausgab object is specified via:

```
:start ausgab object:
    library   = egs_radiative_splitting
    name      = some_name
    splitting = n_split
:stop ausgab object:
```

Similar to the Uniform Bremsstrahlung splitting (UBS) technique implemented in BEAMnrc.

TODO: Add directional radiative splitting (DRS)

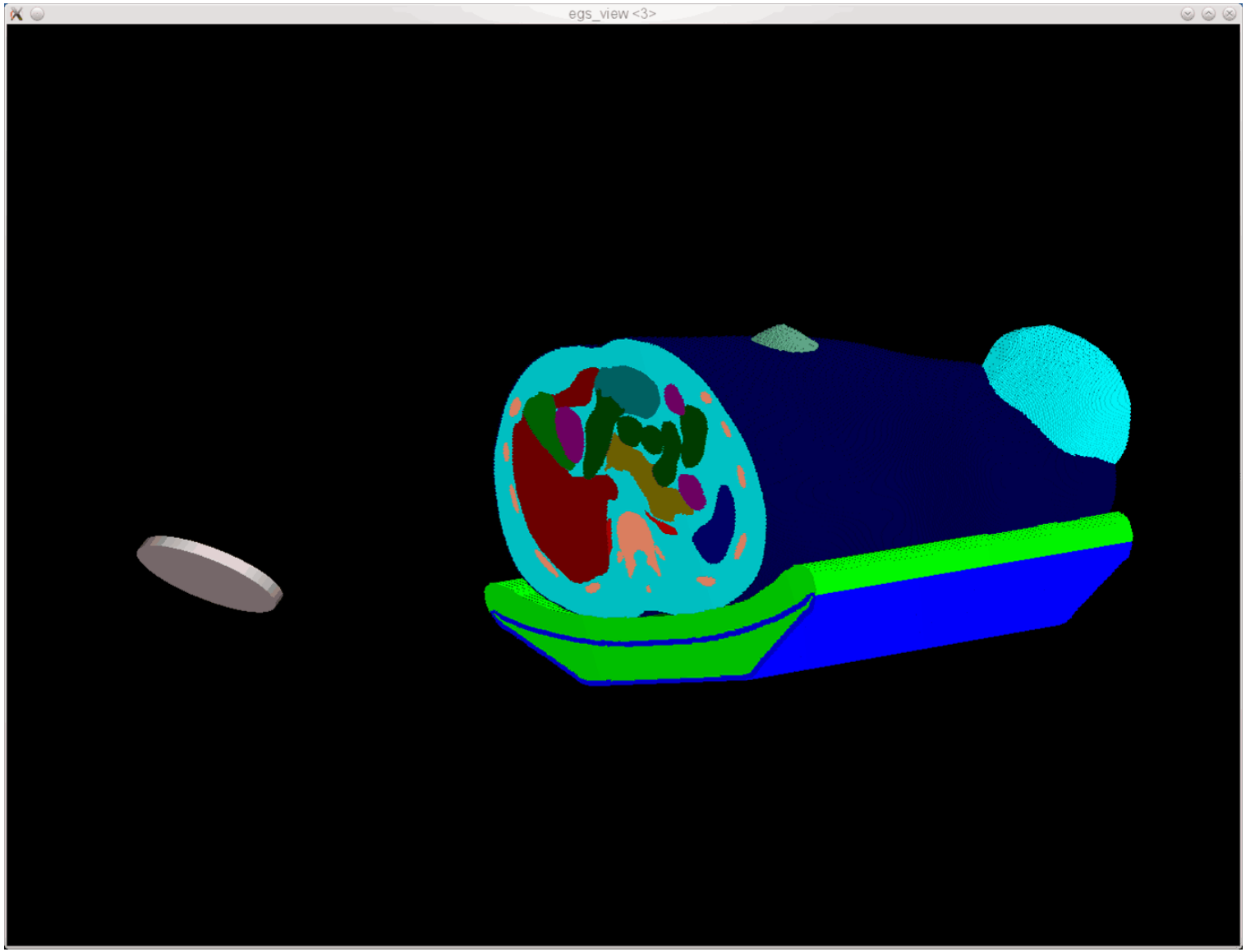
The smallest EGSnrc application

```
#include "egs_advanced_application.h"
int main(int argc, char **argv) {
    EGS_AdvancedApplication app(argc,argv);
    int err = app.initSimulation();
    if( err ) return err;
    err = app.runSimulation();
    if( err < 0 ) return err;
    return app.finishSimulation();
}
```

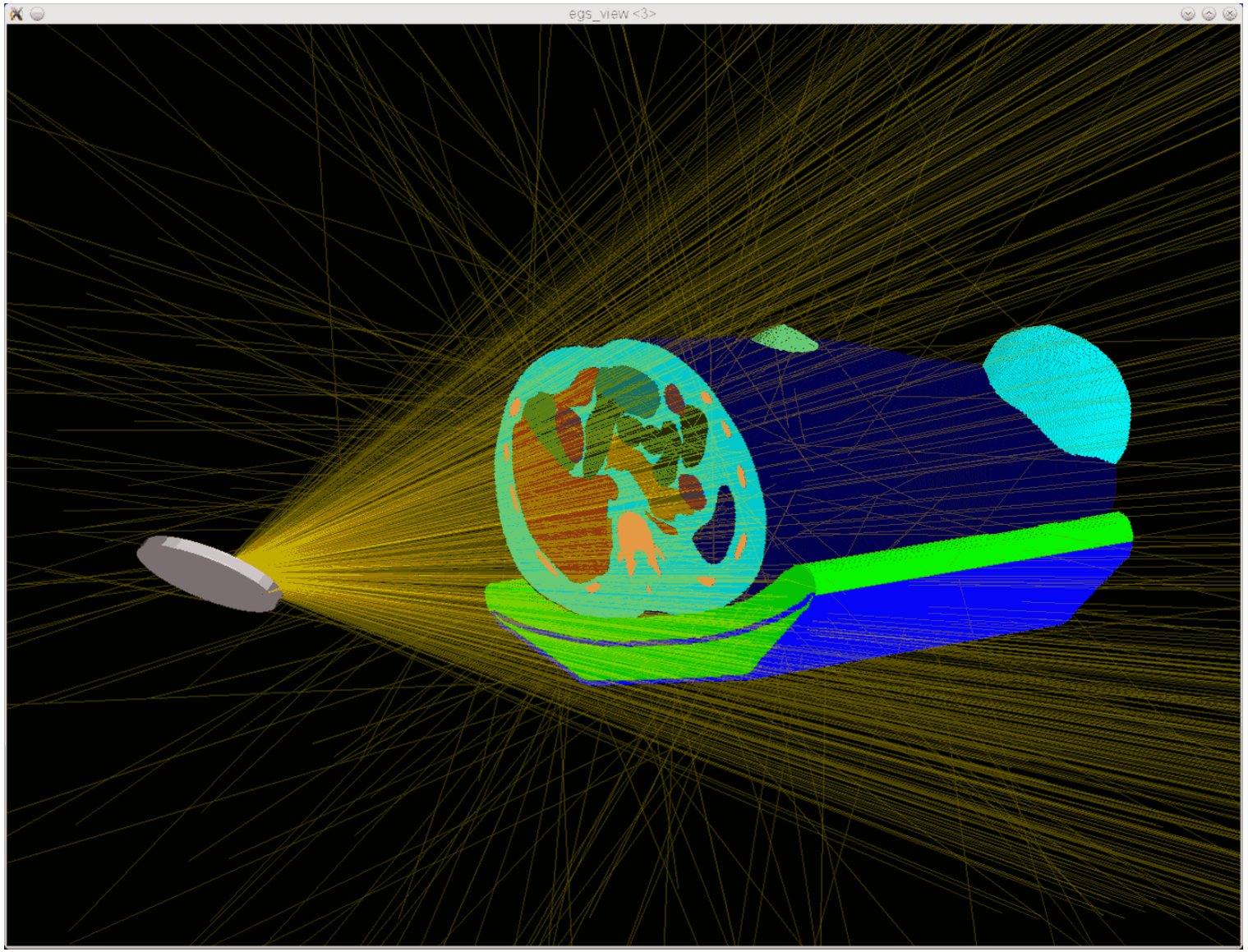
The smallest EGSnrc application: input file

```
:start geometry definition:
   # define the simulation geometry
:stop geometry definition:
:start source definition:
   # define the simulation source
:stop source definition:
:start run control:
   # histories, calculation type
:stop run control:
:start MC transport parameter:
   # energy cut-offs, photon and electron transport parameters
:stop MC transport parameter:
:start ausgab object definition:
    :start ausgab object: # AO 1
    :stop ausgab object:
    .
    :start ausgab object: # AO N
    :stop ausgab object:
:stop ausgab object definition:
```

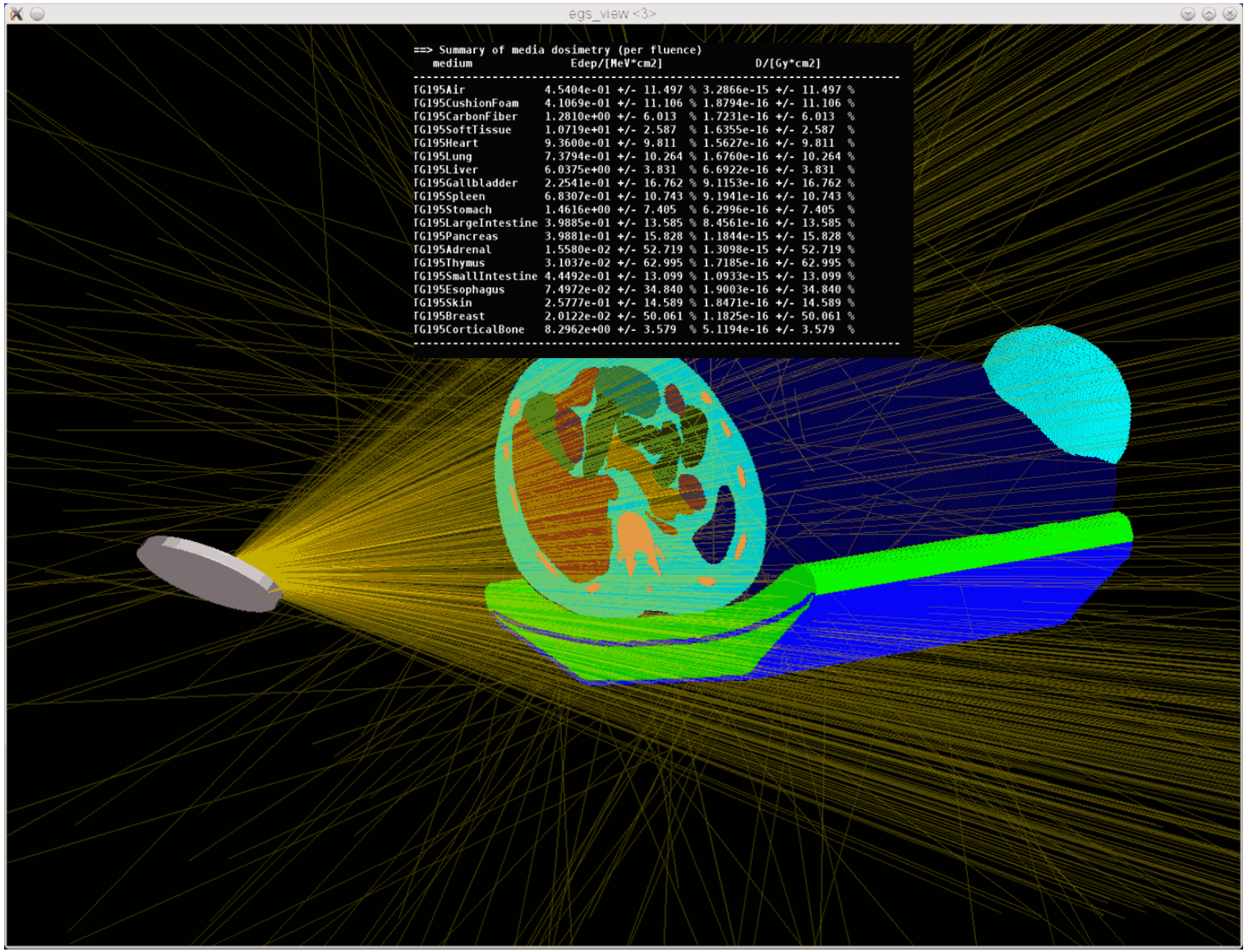
Chest phantom



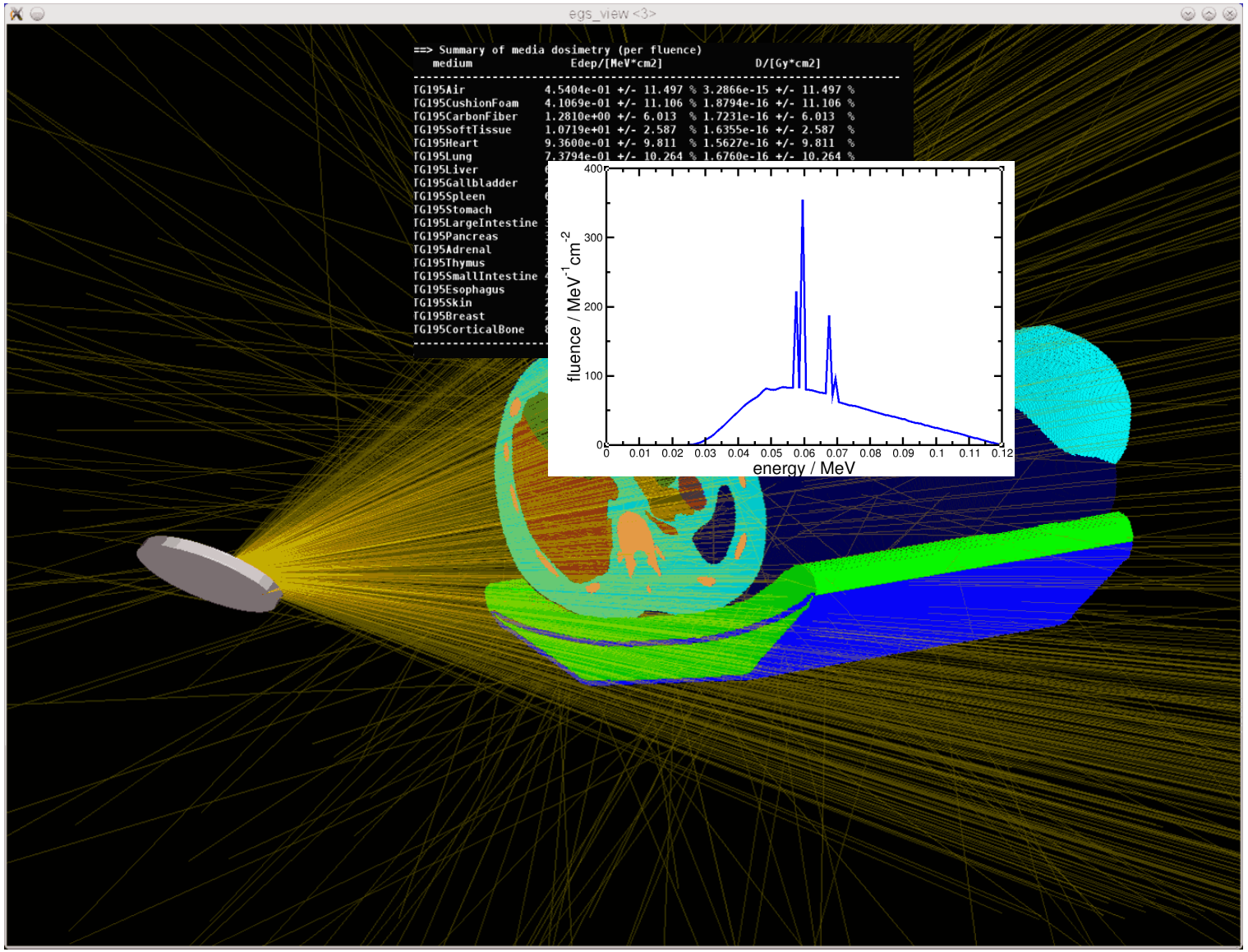
Chest phantom



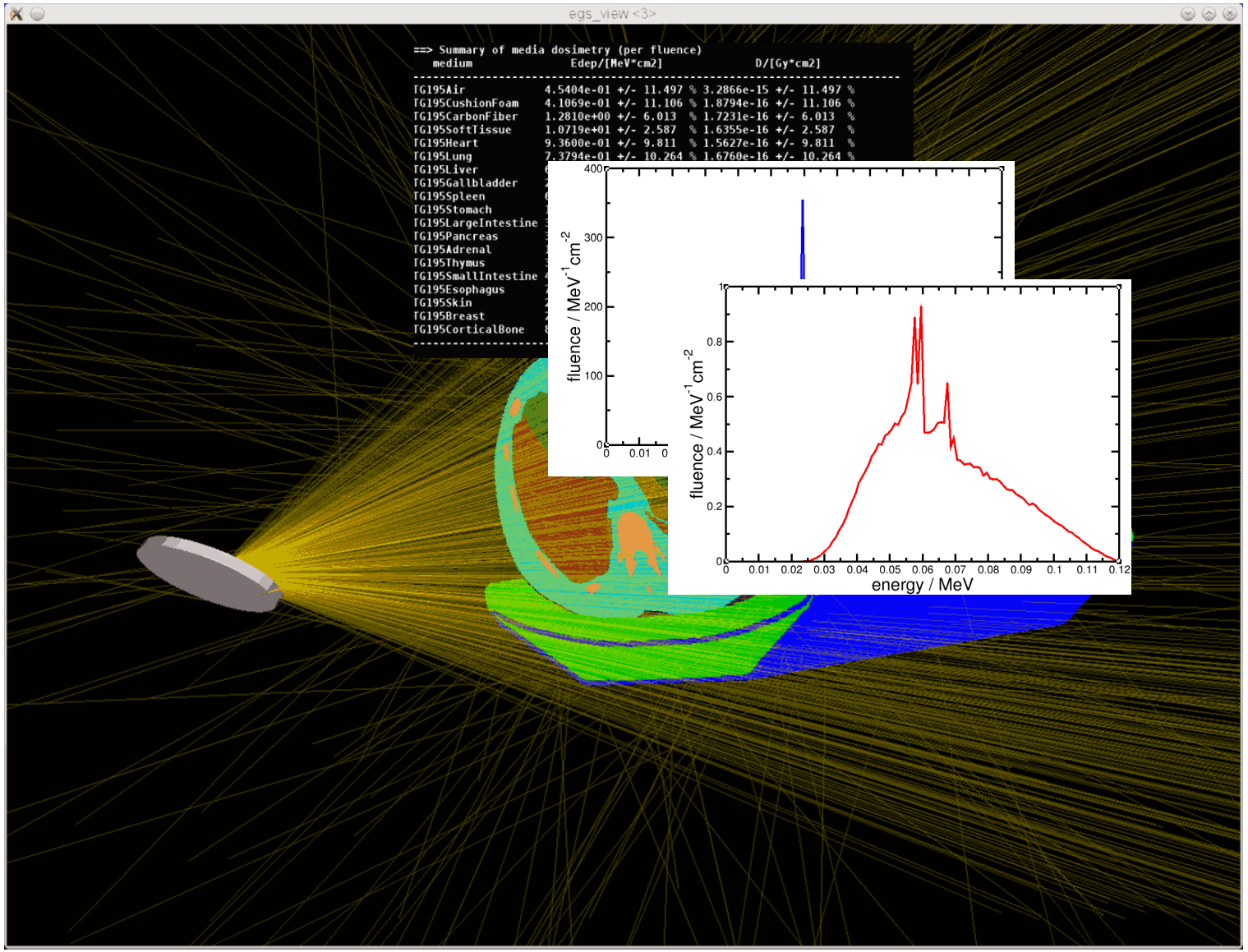
Chest phantom



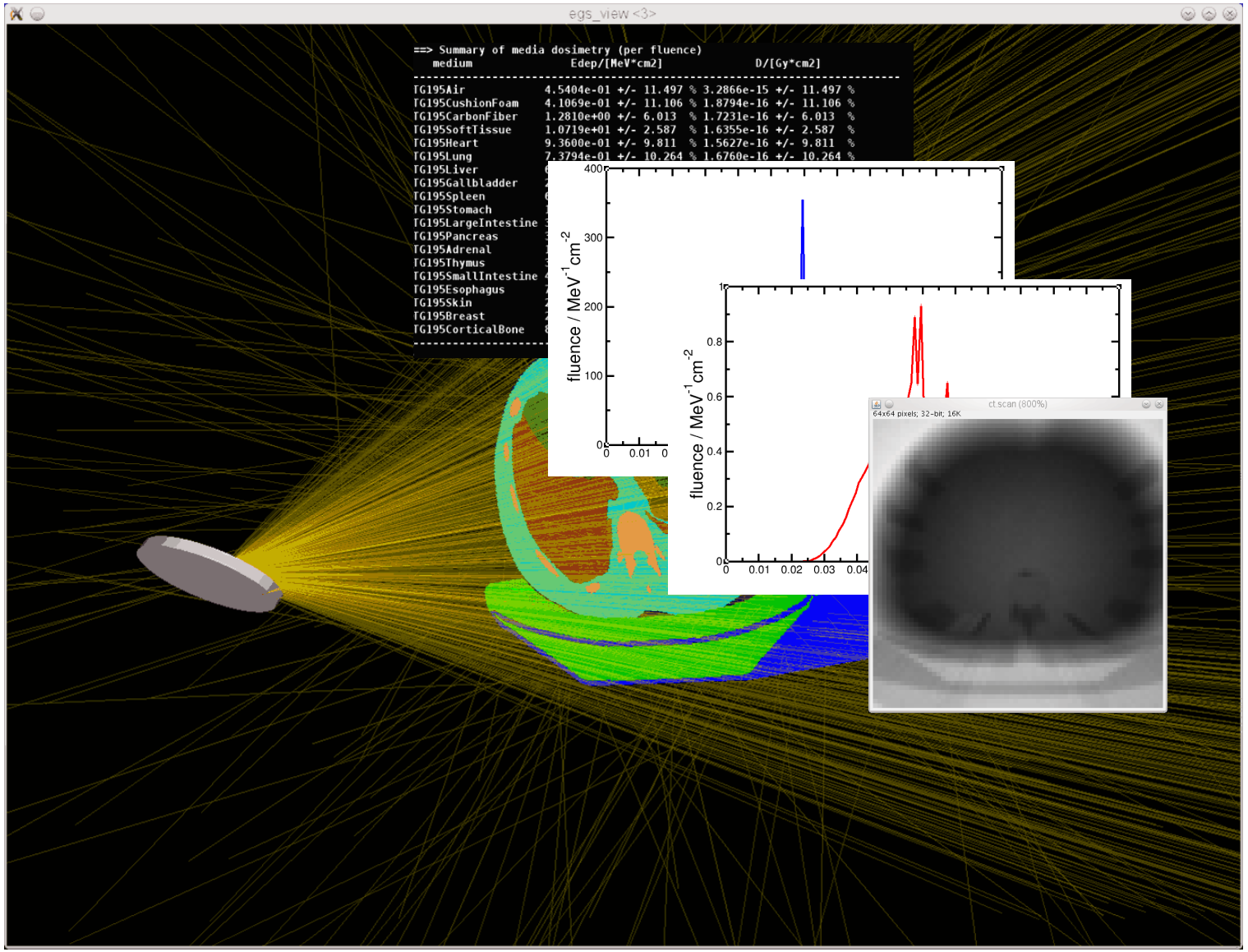
Chest phantom



Chest phantom



Chest phantom



Summarizing

- With the help of scoring AOs, a very basic C++ application becomes a powerful input-based modelling tool.

- Quick start for beginners or users of other MC codes wanting to compare.

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 24

EGSnrc scoring and egs++ ausgab objects

Ernesto Mainegra-Hing

Metrology Research Centre
National Research Council Canada