Lecture 18

Dose analysis and visualization tools

Reid Townson

Metrology Research Centre
National Research Council Canada

The *.3ddose format

- Plain text file

- Dose and uncertainty are slice-by-slice (X-Y planes)

1. Number of voxels in X, Y, Z directions (e.g., n_x, n_y, n_z)

2. Voxel boundaries (cm) in X direction ($n_x + 1$ values)

3. Voxel boundaries (cm) in Y direction ($n_y + 1$ values)

4. Voxel boundaries (cm) in Z direction ($n_z + 1$ values)

5. Dose values array ($n_x n_y n_z$ values)

6. Statistical uncertainty values array (relative errors, $n_x n_y n_z$ values)

Reading *.3ddose is easy

- Read one by one (across columns) to get dose readings in X direction

- Read every (n_x)-th value to get readings in Y direction

- Read every $(n_x n_y)$-th value to get readings in Z direction

Row (block) Number	Column Number				
	1	2	3	4	5
1	4	3	2		
2	-2.0000	-1.0000	0.0000	1.0000	2.0000
3	-2.0000	-1.0000	1.0000	2.0000	
4	0.0000	1.0000	2.0000		
5	1.0e-12	2.1e-12	2.0e-12	1.1e-12	
	0.9e-12	0.9e-12	0.9e-12	0.8e-12	
	0.9e-12	0.9e-12	0.9e-12	0.8e-12	
6	1.0e-12	2.1e-12	2.0e-12	1.1e-12	
	0.9e-12	0.9e-12	0.9e-12	0.8e-12	
	0.9e-12	0.9e-12	0.9e-12	0.8e-12	
7	1.0000E-01	1.0000E-01	1.0000E-01	1.0000E-01	
	1.0000E-01	1.0000E-01	1.0000E-01	1.0000E-01	
	1.0000E-01	1.0000E-01	1.0000E-01	1.0000E-01	
8	1.0000E-01	1.0000E-01	1.0000E-01	1.0000E-01	
	1.0000E-01	1.0000E-01	1.0000E-01	1.0000E-01	
	1.0000E-01	1.0000E-01	1.0000E-01	1.0000E-01	

Reading *.3ddose in c++

```c++
// Read in number of voxels
fscanf(f, "%d %d %d\n", &x_num, &y_num, &z_num);

// Read in boundaries
for(i = 0; i < x_num+1; i++) {
    fscanf(f, "%f", &x_bound[i]);
}
for(i = 0; i < y_num+1; i++) {
    fscanf(f, "%f", &y_bound[i]);
}
for(i = 0; i < z_num+1; i++) {
    fscanf(f, "%f", &z_bound[i]);
}
```

Reading *.3ddose in c++ (cont.)

```cpp
// Read in the dose data
for(k = 0; k < z_num; k++) {
    for(j = 0; j < y_num; j++) {
        for(i = 0; i < x_num; i++) {
            fscanf(f, "%f", &data[i][j][k]);
        }
    }
}

// Read in the dose error
for(k = 0; k < z_num; k++) {
    for(j = 0; j < y_num; j++) {
        for(i = 0; i < x_num; i++) {
            fscanf(f, "%f", &error[i][j][k]);
        }
    }
}
```

The `*.egsphant` format

1. The number of media in the phantom

2. The names of the media

3. The ESTEPE value for each medium (now a dummy input)

4. The number of voxels in the X, Y and Z directions

5. A list of the voxel boundaries in the X direction

6. A list of the voxel boundaries in the Y direction

7. A list of the voxel boundaries in the Z direction

8. For each Z slice, an X-Y array containing the medium number in each voxel

9. For each slice in the Z direction, an X-Y array containing the densities in each voxel

```
1111111111111111111111111111111111111111111111111111111111111111
1111111111111111111111111111111111111111111111111111111111111111
11111111111111111111222222221311111111111111111111111111111111
11111111111111222222222223333333222222111111111111111111111111
111111112223333333333333333333333333333333333222111111111111
11122333333333333333334444444443433333333333333333333221111111
333333333333333333333344333333333344333333333333333333333322221
333333333333333332222233333333333333333222333333333333333333333
33333333334332222222233333333333333333222222222233333333333333
33333333333222222222233333333333333333333222222222233333333333333
33333333333222222222233333333333333333333222222222233333333333333
33333333332222222222322223332223333333333232222222222333333333333
33333333332222222222332223333333333333333322222222223333333334
4333333333222222222322233334443333333333323222222222343333334
4433333443222222222332223344444432222222223222222234433333344
343333334432222222222223444444443222222222222222233333333443
334333333333332222222223444443344443333333333322233333333334433
3334333333333333333334434444444443443322233333334433333344333
333333443333444433333333333433334333333333344433333443333
33333333344333333333333333333333333333333333333334433333333333
33333333333343333333333333333333333333333333333334433333333333
```
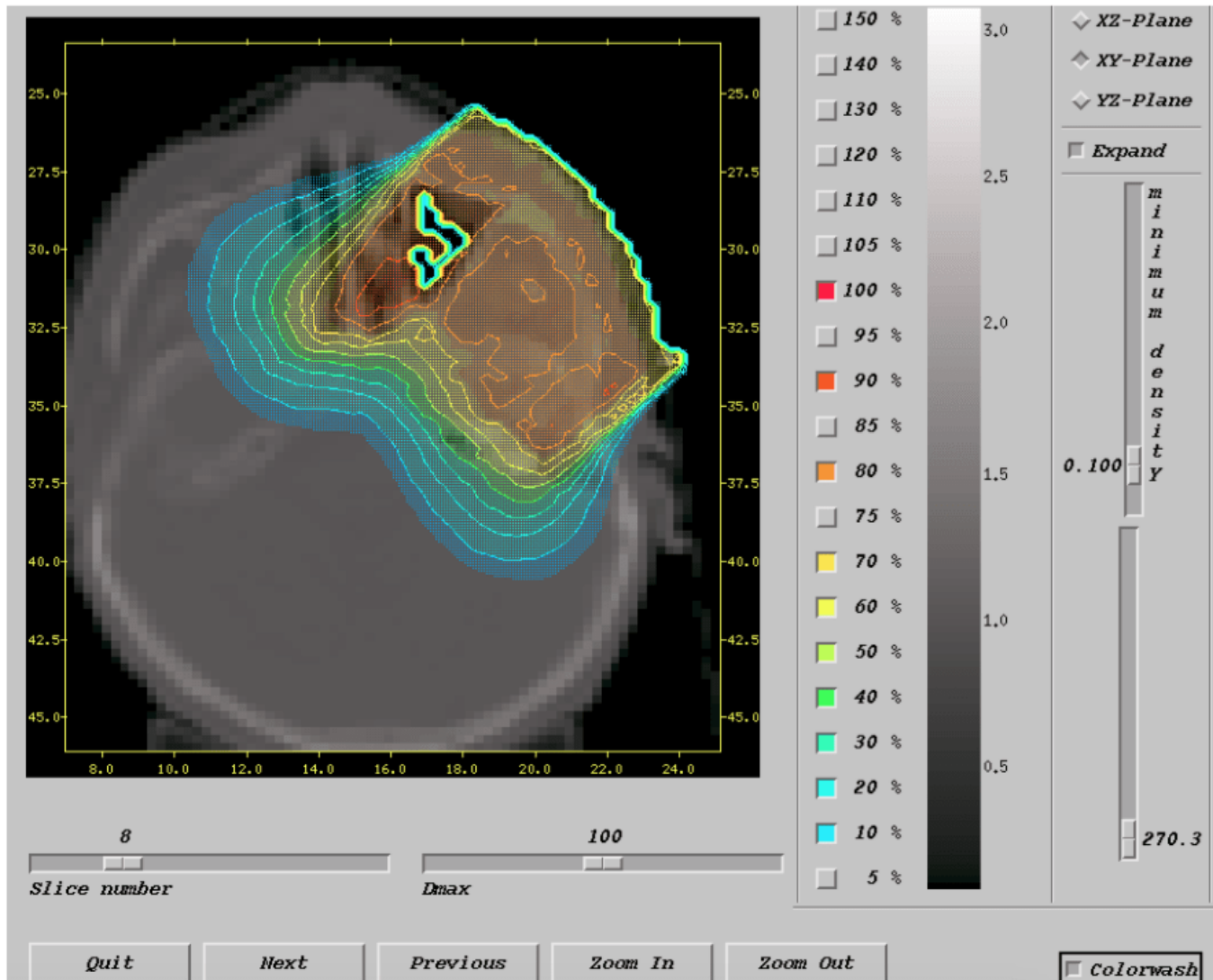
Visualizing dose with dosxyz_show

```
dosxyz_show density_file [dose_file]
```

- Displays slices of a phantom (.egsphant)

- Overlays dose on each slice (.3ddose)

- The dose and density files must be on the same grid

- Open Motif is required to compile (e.g. libmotif-dev)
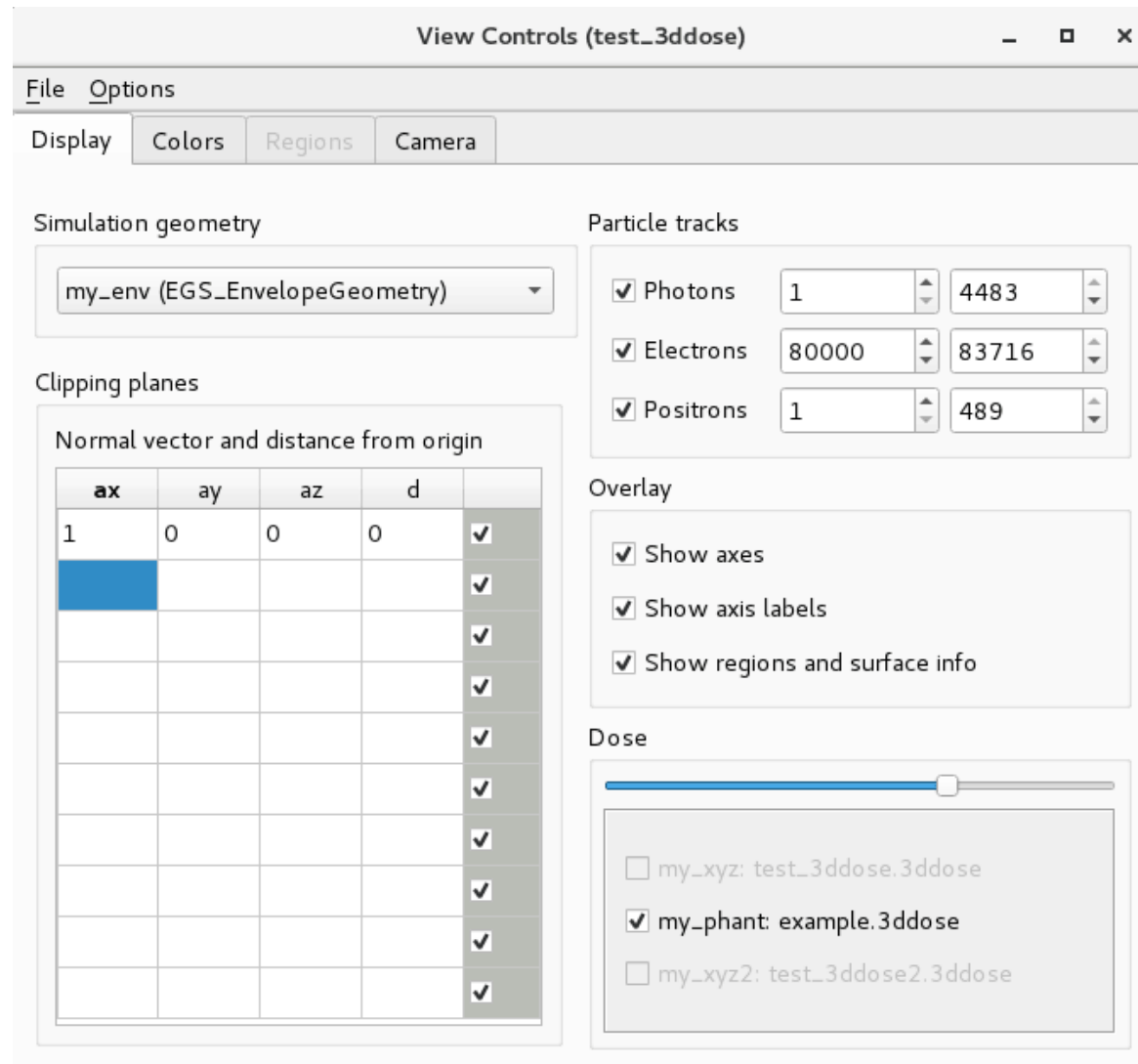
Visualizing dose with dosxyz_show

Dose display in `egs_view`

- Recent developments have added dose display to `egs_view`

- Only 3ddose files supported for XYZ geometries

- Currently no isodose lines or dose smoothing
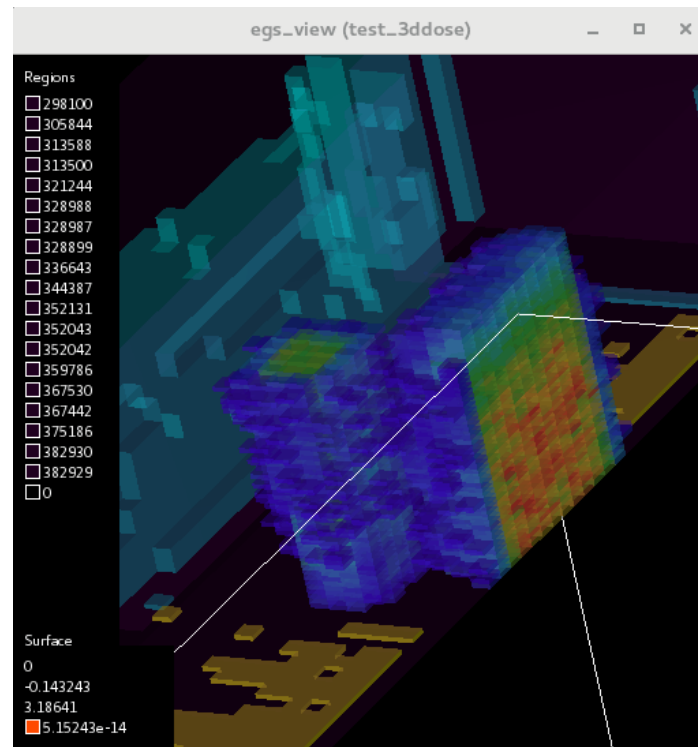
Dose display in `egs_view`

egsphant and 3ddose in egs_view

```
:stop geometry definition:
  :start geometry:
        name = my_phant
        library  = egs_ndgeometry
        type = EGS_XYZGeometry
        egsphant file = slice_files.egsphant
        ct ramp = 5MeV.ramp # this requirement will be removed in future PR
  :stop geometry:
  simulation geometry = my_phant
:stop geometry definition:

:start ausgab object definition:
  :start ausgab object:
        library = egs_dose_scoring
        name = example
        medium dose = no
        region dose = no
        :start output dose file:
            geometry name = my_phant
            file type = 3ddose
        :stop output dose file:
  :stop ausgab object:
:stop ausgab object definition:
```

VICTORIA: 3ddose, egsphant and DICOM viewer
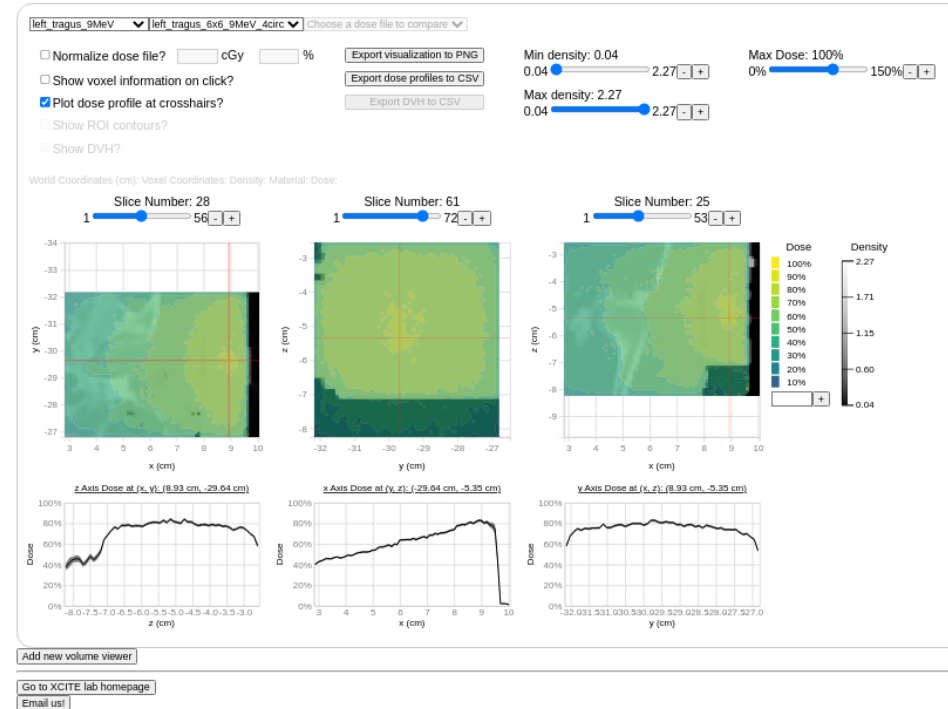
http://web.uvic.ca/ bazalova/VICTORIA/

Importing *.3ddose to a TPS

- Import into a TPS by overwriting DICOM RD dose values

- Requires a DICOM library (e.g. pydicom, CERR) and some coding!

- Update DICOM image parameters:

    - Image Position (0x00200032)

    - Image Orientation (0x00200037)

    - Number of Frames (0x00280008)

    - Rows (0x00280010)

    - Columns (0x00280011)

    - Pixel Spacing (0x00280030)

    - Dose Units item (0x30040002)

    - Grid Offset Vector (0x3004000C)

    - Dose Grid Scaling (0x3004000E)

    - Pixel Data (0x7FE00010)

STATDOSE for Analyzing .3ddose **files**

- Compiled automatically with BEAMnrc installation

- Generates dose profiles as xmgrace plots

- Dose distributions can be normalized, rebinned

- Statistical comparisons can be performed if only two or more dose distributions have been read in, and the voxel geometries are identical

STATDOSE voxel grid dimensions

For larger voxel grids, adjust the dimensions in
`$HEN_HOUSE/omega/progs/statdose/statdose.mortran` and recompile.

```
REPLACE {$MAXVOXX} WITH {128};
REPLACE {$MAXVOXY} WITH {128};
REPLACE {$MAXVOXZ} WITH {128};
```

STATDOSE Example

Begin by reading in one or more `.3ddose` files.

```
*****************************************
        STATDOSE.MORTRAN(Rev 1.3)
        Max array dimensions:  128  128 128
        Max # data sets: 15
*****************************************

 MAIN MENU
 --------
 0 - Exit
 1 - Read dose distributions
 2 - Statistical analysis
 3 - Normalization
 4 - Rebinning
 5 - Plot
 6 - Save

 Selection: 1      # Begin by reading dose files
```

STATDOSE: Assign each file a reference

```
READ DOSE DISTRIBUTIONS
-----------------------
 1 16MVp_h2o_phantom_phsp_example
 2 5MeV-2
 3 5MeV

Input file number to Read in: (1- 7 or 0-Main Menu): 3
File number for temporary storage: (1-* or 0-Main Menu): 1

Number of voxels in X,Y,Z directions:        51        50        51
Voxel-boundary values in X-direction:      -5.10  -   5.10
Voxel-boundary values in Y-direction:       0.00  -  10.00
Voxel-boundary values in Z-direction:      -5.10  -   5.10
Reading dose distribution...
Reading in the uncertainties on DOSE Distribution...

Have read dose distributions
```

STATDOSE: Statistical analysis

```
STATS MENU
---------
0 - Return to main menu
1 - Plot frequency vs. D1-D2
2 - Plot frequency vs. (D1-D2)/[(D1+D2)/2] * 100%
3 - Plot frequency vs. (D1-D2)/max central axis dose * 100%
4 - Plot frequency vs. (D1-D2)/sqrt(err1**2+err2**2)
```

STATDOSE: Normalization

```
NORM MENU
--------
0 - Main menu
1 - Apply scaling factor
2 - Normalize to average dose
3 - Normalize to maximum dose along central axis
4 - Normalize to dose in specific voxel
5 - Denormalize
```

STATDOSE: Rebinning

```
REBINNING
----------

Files currently loaded:


1 - 5MeV
2 - 5MeV-2


Number of file to Rebin (0-MainMenu): 1
Number of Rebinned File (1-5,0-MainMenu): 3
Name of rebinned file (default=5MeV.rb):
Axis for rebinning (1-X,2-Y,3-Z): 1
Rebinning factor: 2
REBINNING......


NUMBER OF VOXELS IN X,Y,Z DIRECTIONS:          26         50          51
Voxel-boundary values in X-direction:       -5.10  -   5.10
Voxel-boundary values in Y-direction:        0.00  -  10.00
Voxel-boundary values in Z-direction:       -5.10  -   5.10
```

STATDOSE: Plotting

If plot axis falls on voxel boundaries, then the plotted dose is the voxel with the lower index (lower X and/or Y value)

```
PLOT MENU
--------
0 - Main menu
1 - Plot profiles
2 - Comparison plot
```

STATDOSE: Plotting

```
COMPARE-PLOT
------------
Axis for Compare-plot (0-PlotMenu,1-X,2-Y,3-Z): 1
Coordinates of Axis (y,z): 0,0
Graph Title (default=Compareplot at  (0.0,0.0)):
Output Filename (default=Compareplot.agr):    # Remember what you put here!

Generate Automatic Offset? (y/n) [n]:

Files currently loaded:

1 - 5MeV
2 - 5MeV-2

Input curve # to be plotted in this graph (0=>View Graph): 1

3D-dose distribution  1: 5MeV

Files currently loaded:

1 - 5MeV
2 - 5MeV-2

Input curve # to be plotted in this graph (0=>View Graph): 2

3D-dose distribution  2: 5MeV
```

STATDOSE: Plotting

5MeV depth dose curves