Anthropomorphic Phantoms in Geant4

Geant4 Training Course in Medicine 2023
Hokkaido University

J. W. Archer[1] *, M. J. Large[1], J. M. C. Brown[2,1], S. Guatelli[1]

[1]Centre for Medical Radiation Physics, University of Wollongong

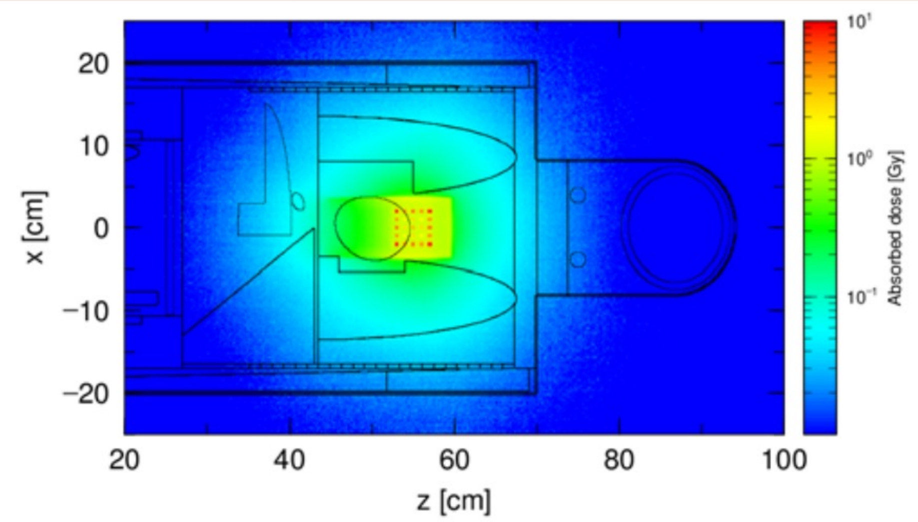
[2]Department of Physics and Astronomy, Swinburne University of Technology

*email: archerj@uow.edu.au

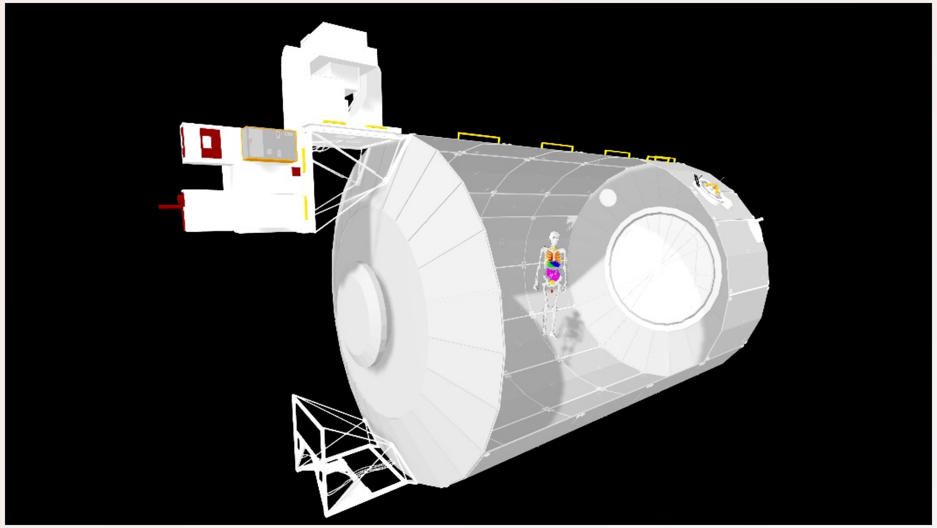
UNIVERSITY OF WOLLONGONG AUSTRALIA

Why Human Phantoms?

- Many different applications of anthropomorphic phantom models:
  - Reference individuals are required for radiological protection and monitoring
  - Required for internal dosimetry (e.g. radionuclides in nuclear medicine and targeted therapies)
  - General radiation transport calculations



MIRD phantoms used for high dose rate superficial brachytherapy[2]



Space radiation protection in the ISS

[1] ICRP, 1975. *ICRP Publication 23*
[2] Ohta et. al., 2017. *Energy Procedia* **131**

Why Human Phantoms?

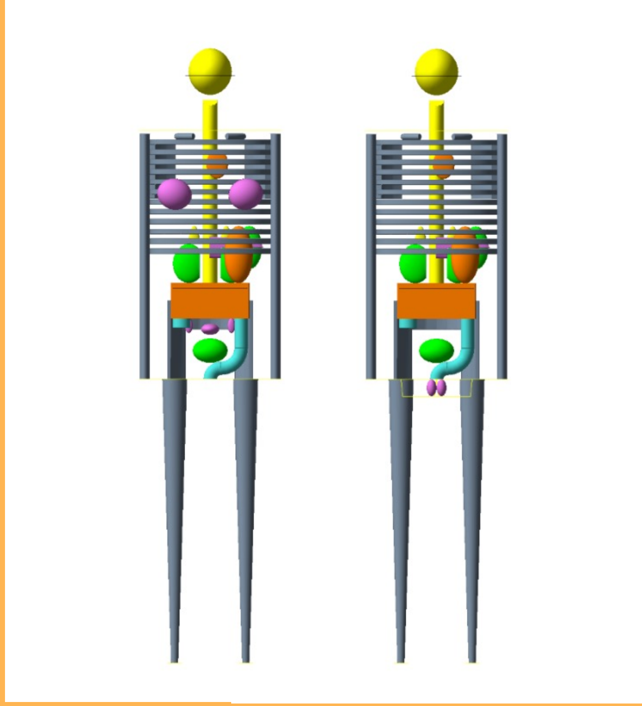
- Whilst there are several publically available phantoms, they all have limitations:
  - Variability in structure due to age, race and other factors
  - Other factors contribute to the risk assessment:
    - differing iodine metabolism
    - water balance depends on environmental factors

[1] ICRP, 1975. *ICRP Publication 23*

Types of Human Phantoms in Geant4

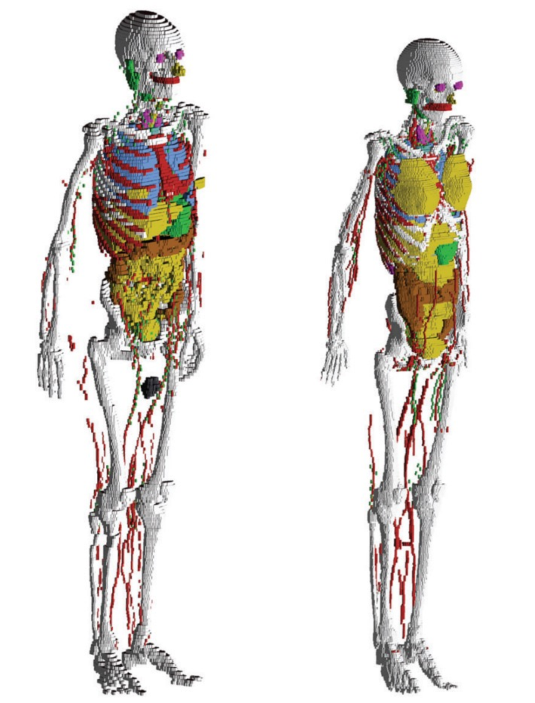
- There are several different phantom of varying complexity included in Geant4:

MIRD/ORNL



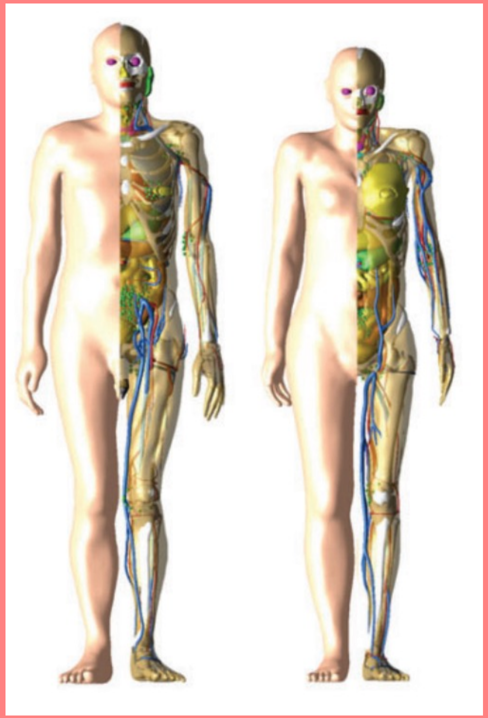
advanced/human_phantom

ICRP110



advanced/ICRP110_HumanPhantoms

ICRP145



advanced/ICRP145_HumanPhantoms

- All available as [advanced examples]()

MIRD Phantoms

- The Medical Internal Radiation Dose (MIRD) committee developed the adult MIRD phantom [3, 4]
  - Extended to children of different ages
  - Extended to male, female and pregnant female models

- Male and female phantoms are included in the *human_phantoms* advanced example

[3] ICRP, 1975. *ICRP Publication 23*
[4] Snyder et. al., 1969. *Journal of Nuclear Medicine Supplemental* **3**

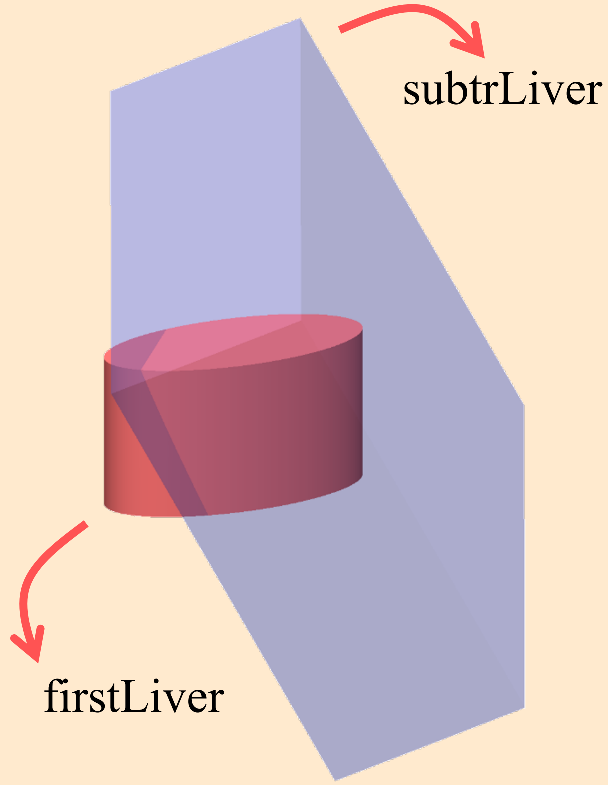
MIRD Phantoms

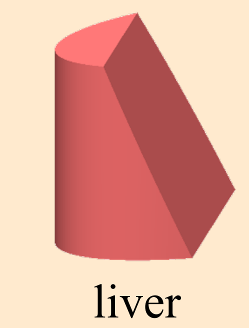
- The MIRD phantom contains *hard-coded* volumes using CSG solids

Example: Liver

G4MirdLiver.cc

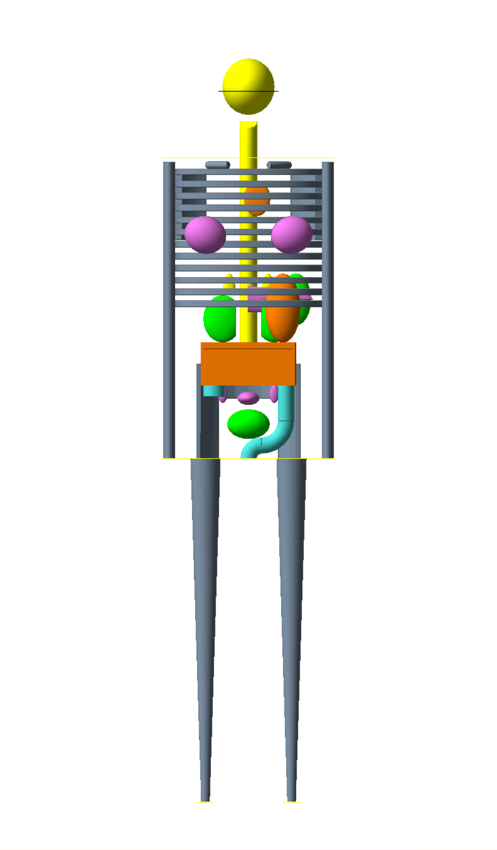
```
G4VPhysicalVolume* G4MIRDLiver::Construct(const G4String& volumeName,
G4VPhysicalVolume* mother, const G4String& colourName, G4bool wireFrame, G4bool)
{
  ...

  G4EllipticalTube* firstLiver = new G4EllipticalTube("FirstLiver", dx, dy, dz);

  G4Box* subtrLiver = new G4Box("SubtrLiver", xx/2., yy/2., zz/2.);

  G4RotationMatrix* rm_relative = new G4RotationMatrix();
  rm_relative -> rotateY(32.* degree);
  rm_relative -> rotateZ(40.9* degree);

  G4SubtractionSolid* liver = new G4SubtractionSolid("Liver",
               firstLiver,subtrLiver,
               rm_relative,
               G4ThreeVector(10.0*cm,0.0*cm,0.0 *cm));

  ...

  return physLiver;
}
```

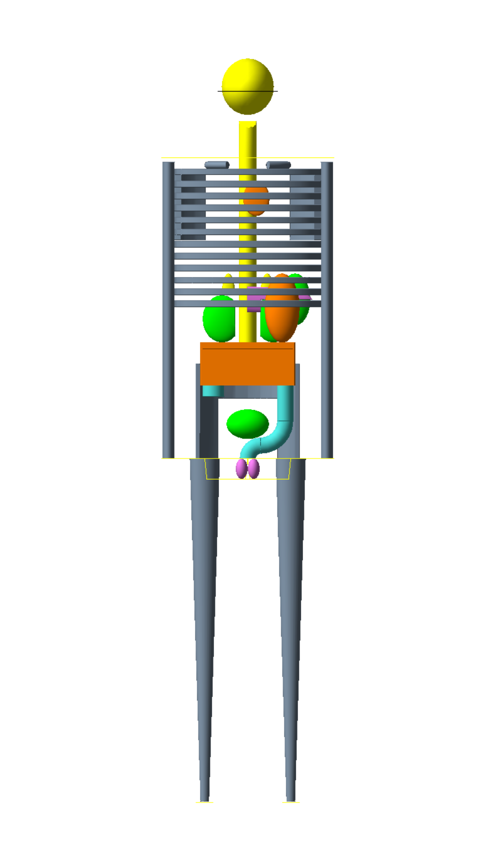




MIRD Phantoms

- Geometry Description Markup Language (GDML) [5] files are available for reference, but are not used for the simulation





[5] cern.ch/gdml

ORNL Phantoms

- The Oak Ridge National Laboratory (ORNL) developed early anatomical phantoms from Boolean operations on geometric shapes [6]
- Whilst phantoms of newborns through to an adult is available, the male and female adult phantoms are only available in Geant4
- The organs of the ORNL phantoms are imported directly from GDML files

Example macro file

```
/phantom/setPhantomModel ORNLFemale # Can also be ORNLMale, MIRD, MIRDHead, ORNLHead

/phantom/setPhantomSex Female # Can also be Male

/phantom/buildNewPhantom

/run/initialize
```

[6] Cristy and Eckerman, 1987. *ORNL/TM-8381/VI*

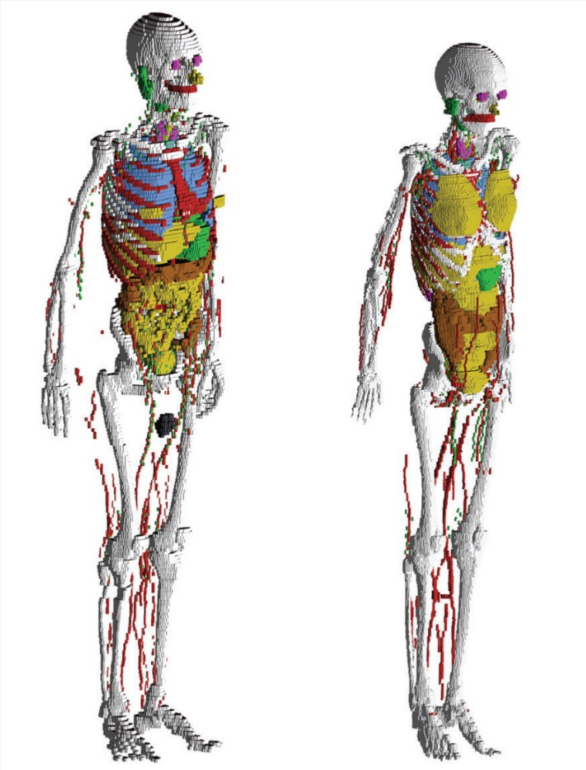
MIRD/ORNL: Advanced Analysis

- By default, the simulation outputs the energy deposited in each organ
- To perform custom analysis, the organ volume at a given step point can be obtained from the *G4LogicalVolume*:

```
void SteppingAction::UserSteppingAction(const G4Step* step)
{
    G4String bodyPartName = step->GetPreStepPoint()->GetTouchable()->GetVolume()->GetLogicalVolume()->GetName();

    if (bodyPartName == "Liver")
    {
        // Do something...
    }
}
```

ICRP110 Voxelised Phantoms

- ICRP Report 110 [7] outlined a *voxel* phantom based on medical imaging
- Most organs relevant for radiological assessment are described
- The height of the voxels (*z*) were scaled to a reference height, whilst the width (*x, y*) were scaled to the skeletal mass
- Male and female phantoms included:



[7] ICRP, 2009. *Annals of the ICRP ICRP* **39**(2)

ICRP110 Voxelised Phantoms

Property	Male	Female
Height (m)	1.76	1.63
Mass (kg)	73.0	60.0
Number of tissue voxels	1,946,375	3,886,020
Slice thickness (voxel height, mm)	8.0	4.84
Voxel in-plane resolution (mm)	2.137	1.775
Voxel volume (mm^3)	36.54	15.25
Number of columns	254	299
Number of rows	127	137
Number of slices	220 (+2)*	346 (+2)*

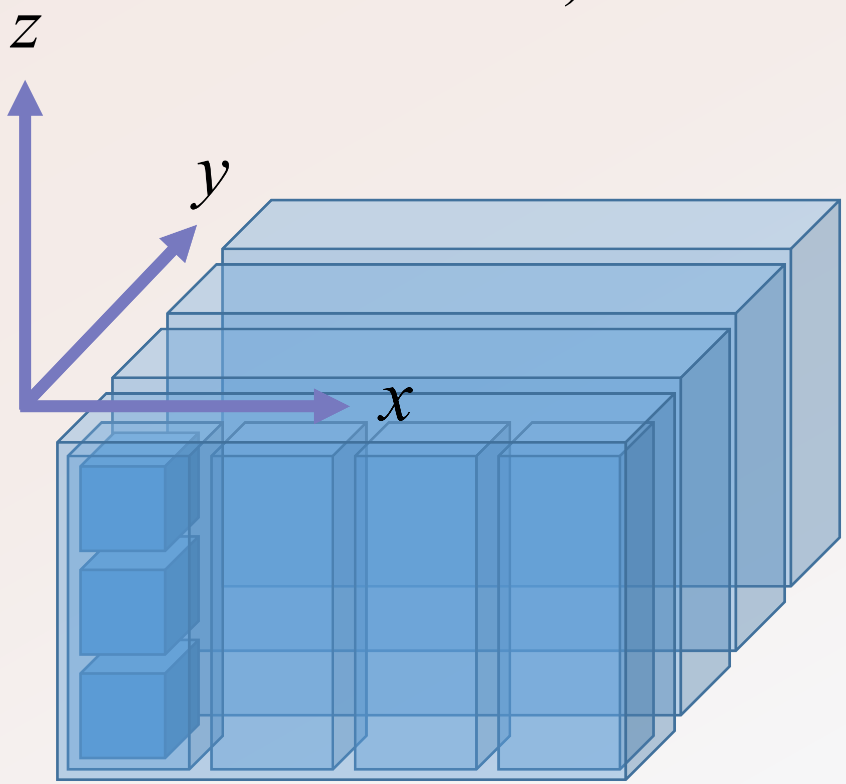
[7] ICRP, 2009. *Annals of the ICRP ICRP* **39**(2)

ICRP110 Phantom Implementation

- Implemented in a Geant4 advanced example *ICRP110_HumanPhantoms* [8]
- Created using a parameterised geometry (*G4VNestedParameterisation*)

ICRP110PhantomConstruction.cc

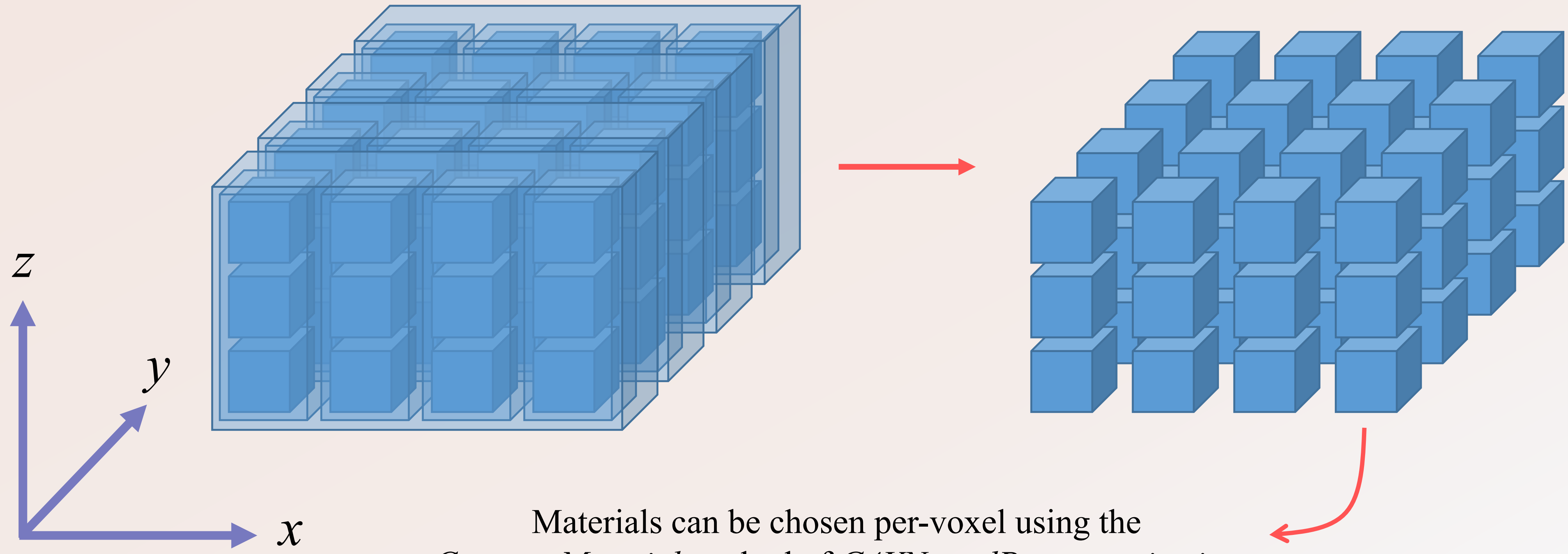
```
G4VPhysicalVolume* ICRP110PhantomConstruction::Construct()
{

   new G4PVReplica(yRepName, logYRep, fContainer_logic, kYAxis, fNVoxelY,
fVoxelHalfDimY*2.);

   new G4PVReplica(xRepName, logXRep, logYRep, kXAxis, fNVoxelX,
fVoxelHalfDimX*2.);

   ICRP110PhantomNestedParameterisation* param =  new
ICRP110PhantomNestedParameterisation(halfVoxelSize, pMaterials);

    new G4PVParameterised("phantom",     // their name
                          logicVoxel,   // their logical volume
                          logXRep,      // Mother logical volume
                          kZAxis,       // Are placed along this axis
                          fNVoxelZ,     // Number of cells
                          param);       // Parameterisation
}
```



[8] Large et. al., 2020. *Journal of Physics: Conference Series (MMND ITRO 2020)*

ICRP110 Phantom Implementation

- Due to the nested parameterisation, the result is a *voxel* structure:

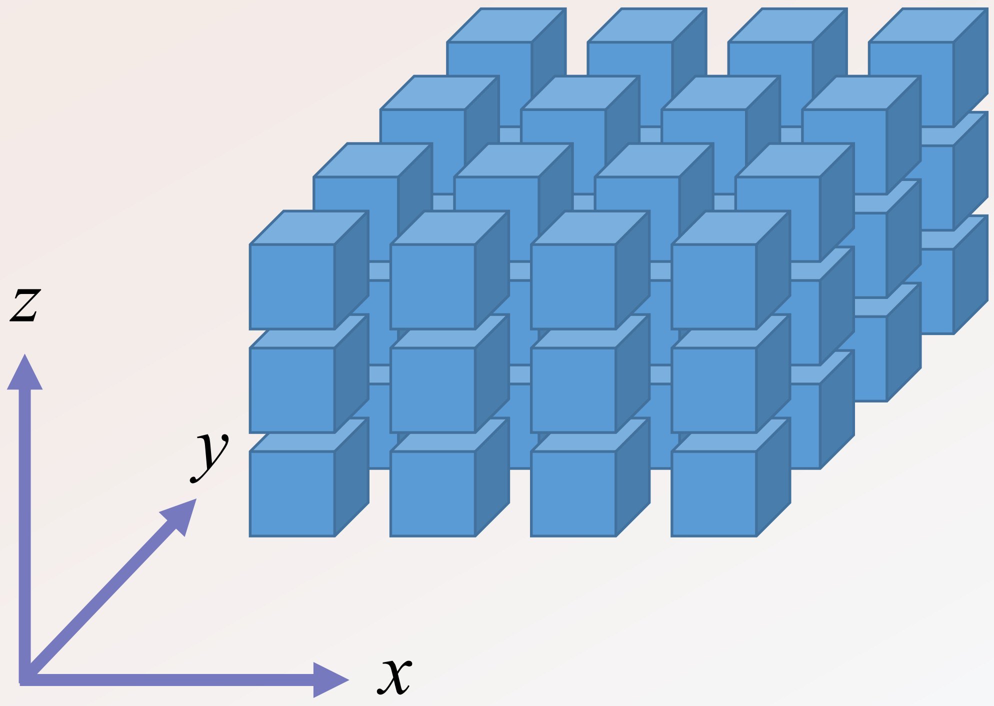


Materials can be chosen per-voxel using the *ComputeMaterial* method of *G4VNestedParameterisation*

ICRP110 Phantom Implementation

- The voxels can be identified using the copy number of each volume

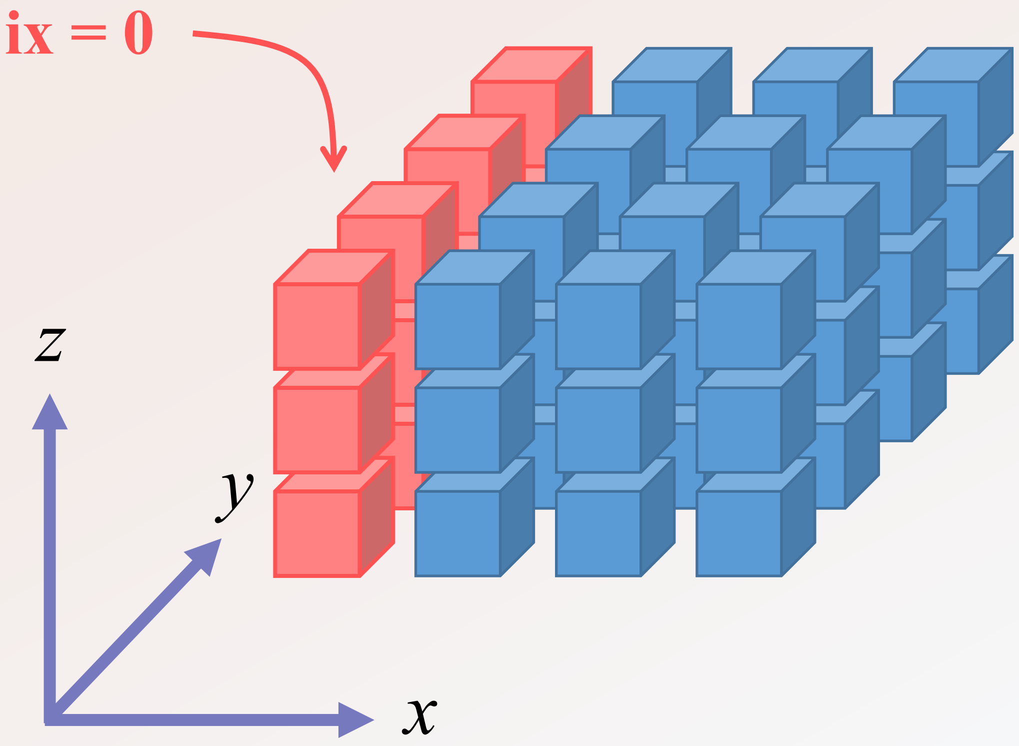
```
G4Material* ICRP110PhantomNestedParameterisation::
ComputeMaterial(G4VPhysicalVolume* physVol, const G4int
iz,const G4VTouchable* parentTouch)
{
    G4int ix = parentTouch->GetCopyNumber(0);
    G4int iy = parentTouch->GetCopyNumber(1);
    G4int iz = physVol->GetCopyNo();
    ...
    return material;
}
```



ICRP110 Phantom Implementation

- The voxels can be identified using the copy number of each volume

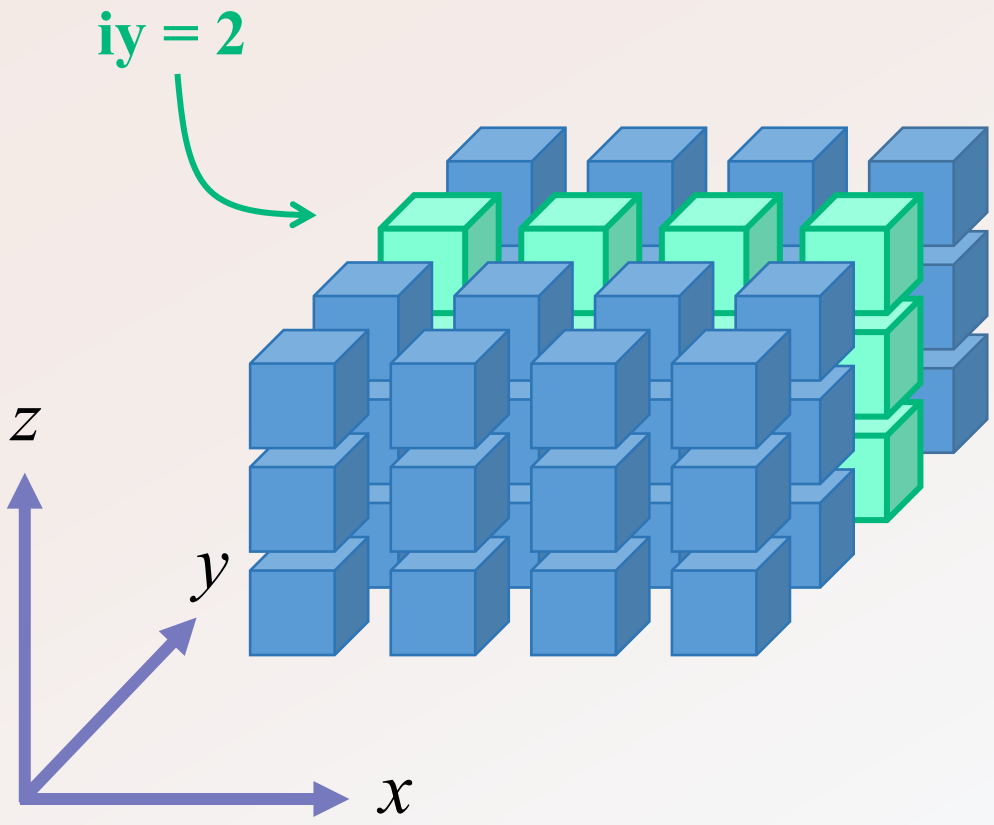
```
G4Material* ICRP110PhantomNestedParameterisation::
ComputeMaterial(G4VPhysicalVolume* physVol, const G4int
iz,const G4VTouchable* parentTouch)
{
    G4int ix = parentTouch->GetCopyNumber(0);
    G4int iy = parentTouch->GetCopyNumber(1);
    G4int iz = physVol->GetCopyNo();
    ...
    return material;
}
```



ICRP110 Phantom Implementation

- The voxels can be identified using the copy number of each volume

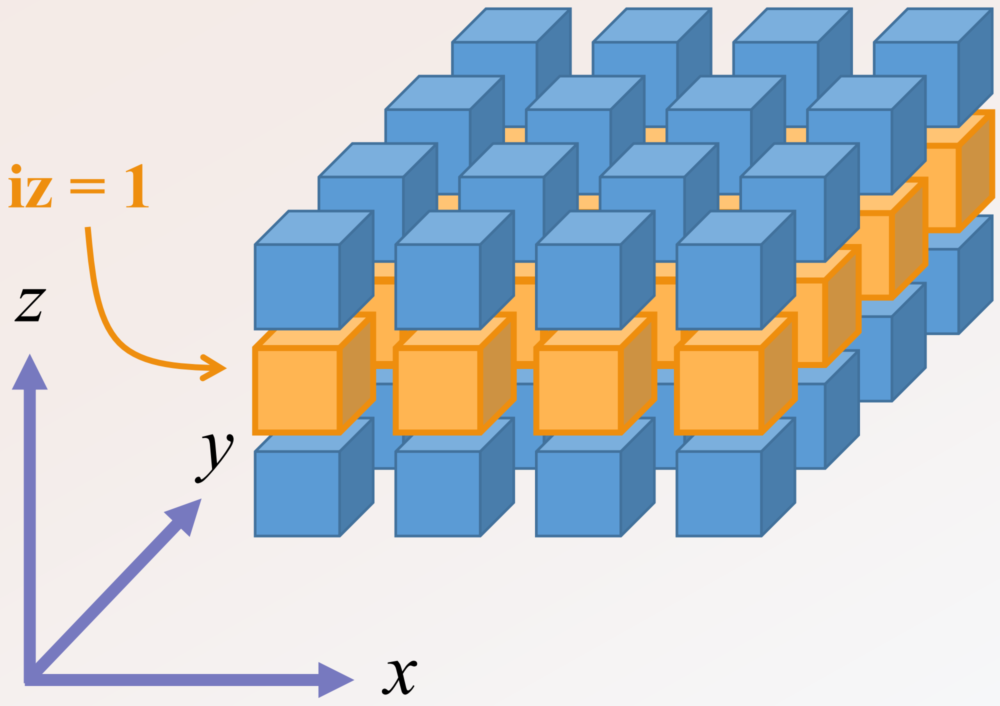
```
G4Material* ICRP110PhantomNestedParameterisation::
ComputeMaterial(G4VPhysicalVolume* physVol, const G4int
iz,const G4VTouchable* parentTouch)
{
    G4int ix = parentTouch->GetCopyNumber(0);
    G4int iy = parentTouch->GetCopyNumber(1);
    G4int iz = physVol->GetCopyNo();
    ...
    return material;
}
```



ICRP110 Phantom Implementation

- The voxels can be identified using the copy number of each volume

```
G4Material* ICRP110PhantomNestedParameterisation::
ComputeMaterial(G4VPhysicalVolume* physVol, const G4int
iz,const G4VTouchable* parentTouch)
{
    G4int ix = parentTouch->GetCopyNumber(0);
    G4int iy = parentTouch->GetCopyNumber(1);
    G4int iz = physVol->GetCopyNo();
    ...
    return material;
}
```



ICRP110 Phantom Implementation

The material of the individual voxels is handled by *ICRP110PhantomNestedParameterisation*

ICRP110PhantomNestedParameterisation.cc

```
G4Material* ICRP110PhantomNestedParameterisation::
ComputeMaterial(G4VPhysicalVolume* physVol, const G4int iz,const G4VTouchable*
parentTouch)
{
    G4int ix = parentTouch -> GetReplicaNumber(0);
    G4int iy = parentTouch -> GetReplicaNumber(1);

    G4int copyID = ix + fnX*iy + fnX*fnY*iz;
    std::size_t matIndex = GetMaterialIndex(copyID);
    static G4Material* mate = nullptr;
    mate = fMaterials[matIndex];

    return mate;
}
```

This is a unique integer for each voxel

fMaterials stores the materials for each organ

ICRP110 Data Definitions

- The data files consist of:

Metadata File

MaleData.dat / FemaleData.dat

```
222
254
127
1.0685
1.0685
4
53
0          air
1          teeth
2          bone
3          humeri_upper
```

- 222, 254, 127: Number voxels in *z*, *x*, *y*
- 1.0685, 1.0685, 4: Voxel half-widths (mm)
- 0 air … 3 humeri_upper: Organs

```
AM_slice1.g4dat
AM_slice2.g4dat
AM_slice3.g4dat
```

- Slice files

Slice Files

MaleData.dat / FemaleData.dat

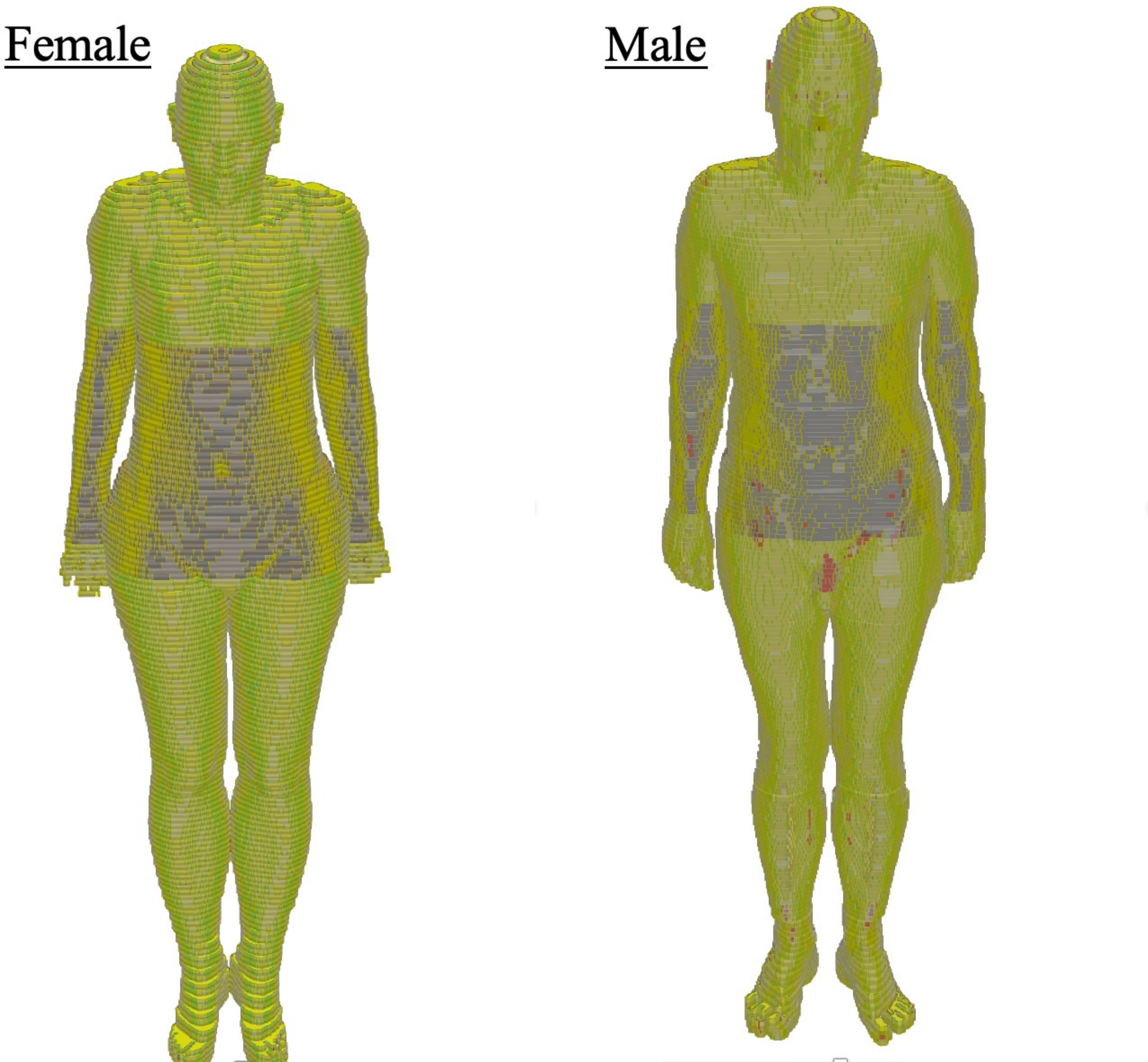
```
254
127
222
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
```

- 254, 127, 222: Number voxels in *z*, *x*, *y*
- 0 0 0 0 0 …: Material identifier for each voxel

ICRP110 Data

- Each slice file is a different layer of voxels

- The voxel nature of the phantom makes it easy to select specific slices



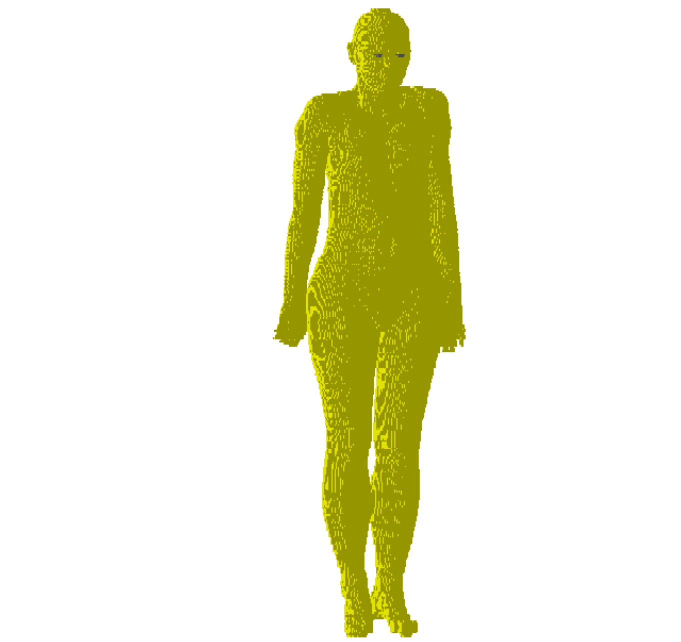
ICRP110 Sections

- The included macro files allow for easy implementation of specific sections:

```
/phantom/setPhantomSection head
```

```
/phantom/setPhantomSection trunk
```

```
/phantom/setPhantomSection full
```



- Can also select custom slices by modifying the number of z slices and the data files in the metadata file

ICRP145 Phantom

- To overcome limited voxel resolution, tetrahedral mesh phantoms were introduced in ICRP Report 145 [8]
- In this phantom:
  - All organs are completely enclosed (including the skin)
  - Detailed models of tubular intestines, spinal structures, lymphatic nodes, eye lens', skin layers

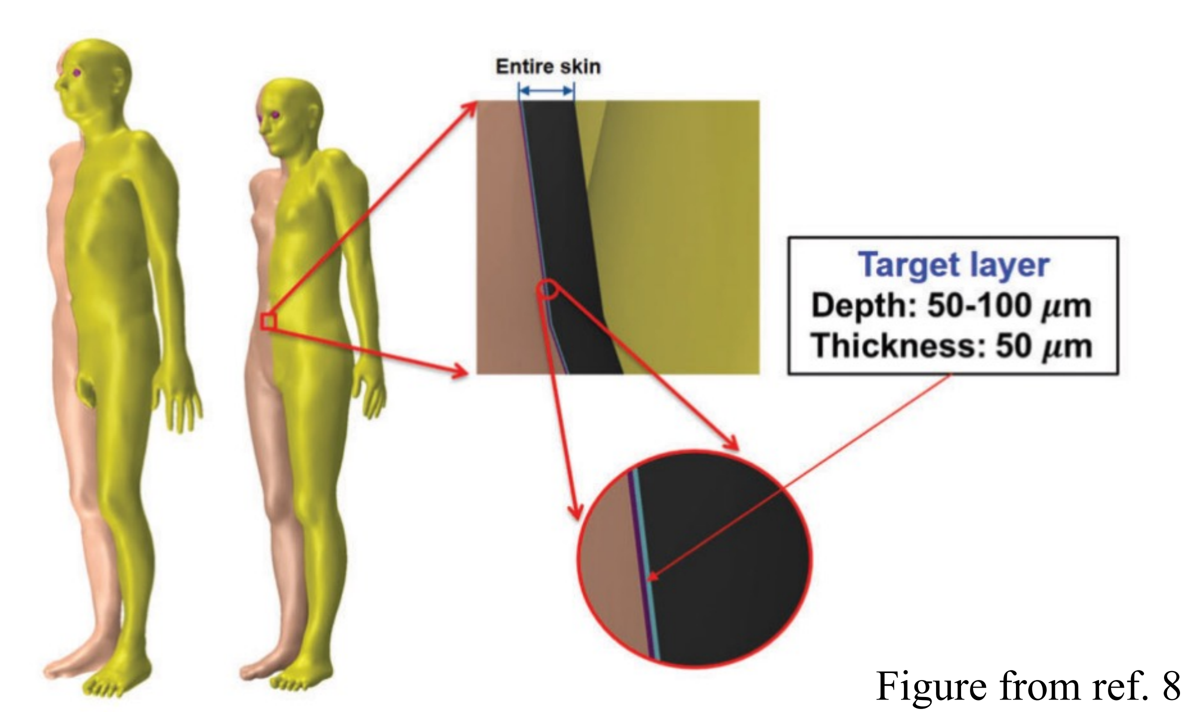


Figure from ref. 8

[8] ICRP, 2009. *Annals of the ICRP ICRP* **49**(3)

ICRP145 Phantom

The tetrahedral mesh phantoms were generated from the voxel phantoms using a smoothing and adjustment process:

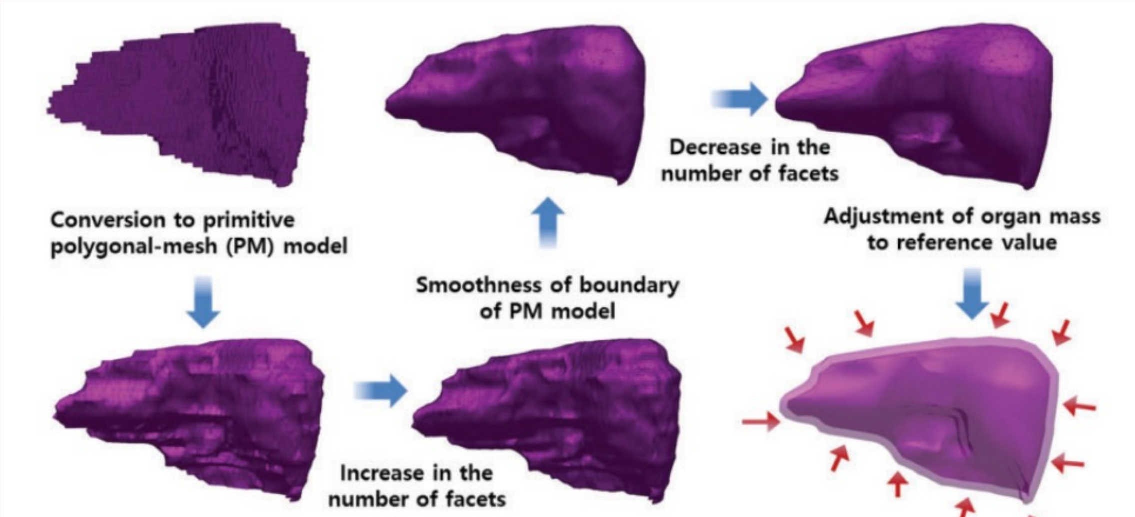


Figure from ref. 8

Doses in each organ computed as standard output:

```
==========================================================================
 Run #0 / Number of event processed : 10000000
==========================================================================
organ ID|        Organ Mass (g)    Dose (Gy/source)      Relative Error
     100|                 8.683           1.823e-17               0.214
     200|                 8.683           2.226e-17               0.184
     300|                 0.022           1.212e-17               0.623
     301|                 0.090           1.298e-17               0.594
     302|                 0.028           1.613e-17               0.553
     303|                11.291           2.867e-17               0.163
     400|                 0.141           3.189e-17               0.582
     401|                 0.390           1.916e-17               0.400
     402|                 0.098           1.795e-17               0.367
     403|                 0.049           2.251e-17               0.450
     404|                 0.098           2.337e-17               0.340
     405|                28.808           2.232e-17               0.105
```

[8] ICRP, 2009. *Annals of the ICRP* **49**(3)

ICRP145 Data Definitions

- The data files consist of:

Vertices File

.node

```
1279642  3  0  0
   0    1.728173  0.274099  33.475464
   1    1.550969  0.481751  33.402969
   2    1.788876  0.210670  33.374435
   3    1.819830  0.170031  33.444263
   4    2.168805  3.718461  33.734196
   5    2.124705  3.613879  33.695755
   6    2.047916  3.682885  33.755173
   7    2.105053  3.762564  33.829369
   8    2.268390  3.746360  33.721901
   9    2.237944  3.665057  33.678017
  10    1.512970  3.144052  34.097740
  11    1.445530  2.994499  34.030819
  12    1.479287  3.046279  34.258648
```

Vertex ID — *x*, *y* and *z* position

Tetrahedra File

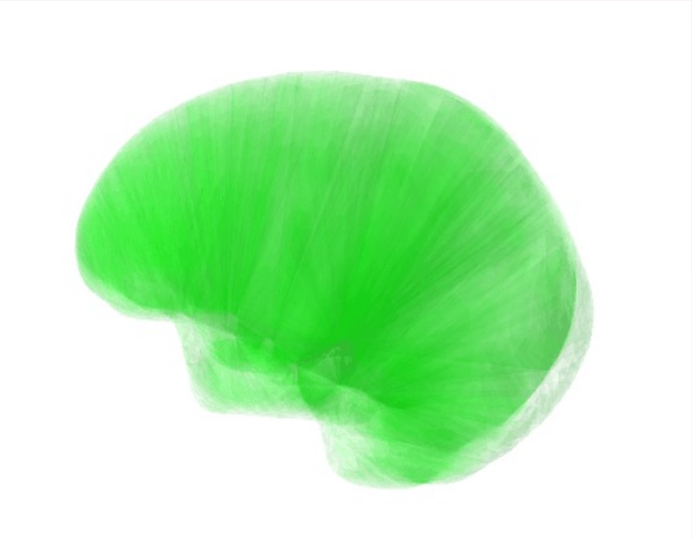
.ele

```
8582677  4  1
0   1226007 1148977 1225948 1149037    12501
1   901017 459351 901018 459350    11700
2   669624 649462 669602 649482    7401
3   588413 589616 588414 589617    5500
4   1253585 73801 1253586 1253584    11700
5   797418 642003 797417 642234    11700
6   712743 707964 712738 707966    11700
7   1136520 1092753 1136510 1135500    12400
8   896817 896836 896823 854111    10700
9   248688 248687 868295 248689    11600
10   804387 802611 804381 804383    9900
11   527654 471133 527560 527563    4900
12   960713 975965 961002 961003    10900
```

Tet. ID — IDs of the vertices of the tetrahedra — Organ ID

ICRP145 Data

- The data is imported through the *TETModelImport* class
- Through small modifications, the import can be restricted from tetrahedra of certain organ ids or positions
- The *organ id* of a volume can be obtained using the tetrahedron *copy number*



```
void SteppingAction::UserSteppingAction(const G4Step* step)
{
    ...
    G4int copyNo = step->GetPreStepPoint()->GetTouchable()->GetCopyNumber();
    G4int organId = fTETData->GetMaterialIndex(copyNo);

    if (organId == 6100)
    {
        // This is in the brain, do something...
    }
}
```

Bone Dose Computations

- All phantoms listed do not directly segment active (red) bone marrow
- Instead, the ICRP Report 116 [9] describes an estimation of the dose to active marrow using doses to the spongiosa:

- For each bone site *x* the total active marrow dose $D_{skel}(AM)$ is the total dose to the spongiosa at each site $D(SP, x)$ scaled by the fractional mass of active marrow at each site $m(AM, x)/m(AM)$:

$$D_{skel}(AM) = \sum_{x} \frac{m(AM, x)}{m(AM)} D(SP, x)$$

[9] ICRP, 2010. *Annals of the ICRP* **40**(2-5)

Bone Dose Computations

- Active marrow masses available in ICRP 110 and ICRP 116:

Table 3.2. Skeletal tissue masses in the ICRP reference adult male and adult female computational phantoms.

Organ	Bone site	Reference adult male				Reference adult female			
		Active marrow		Endosteum		Active marrow		Endosteum	
		Mass (g)	Mass (%)	Mass (g)	Mass (%)	Mass (g)	Mass (%)	Mass (g)	Mass (%)
14	Humeri, upper half – spongiosa	26.9	2.3	9.41	1.7	20.7	2.3	7.16	1.8
15	Humeri, upper half – medullary cavities			0.19	0.0			0.14	0.0
17	Humeri, lower half – spongiosa			11.25	2.1			8.32	2.0
18	Humeri, lower half – medullary cavities			0.25	0.0			0.19	0.0
20	Lower arm bones – spongiosa			16.31	3.0			12.03	3.0
21	Lower arm bones – medullary cavities			0.09	0.0			0.07	0.0
23	Wrists and hands – spongiosa			12.50	2.3			7.10	1.7
25	Clavicles – spongiosa	9.3	0.8	2.50	0.5	7.2	0.8	1.90	0.5
27	Cranium – spongiosa	88.9	7.6	83.40	15.3	68.4	7.6	64.20	15.8
29	Femora, upper half – spongiosa	78.4	6.7	43.34	8.0	60.3	6.7	33.53	8.2
30	Femora, upper half – medullary cavities			0.86	0.2			0.67	0.2
32	Femora, lower half – spongiosa			47.83	8.8			23.67	5.8
33	Femora, lower half – medullary cavities			0.67	0.1			0.33	0.1
35	Lower leg bones – spongiosa			87.38	16.1			79.91	19.6
36	Lower leg bones – medullary cavities			5.02	0.9			4.59	1.1
38	Ankles and feet – spongiosa			42.20	7.8			24.40	6.0
40	Mandible – spongiosa	9.4	0.8	2.00	0.4	7.2	0.8	1.60	0.4
42	Pelvis – spongiosa	205.2	17.5	51.70	9.5	157.5	17.5	39.70	9.7
44	Ribs – spongiosa	188.8	16.1	29.80	5.5	144.9	16.1	22.90	5.6
46	Scapulae – spongiosa	32.8	2.8	9.80	1.8	25.2	2.8	7.60	1.9
48	Cervical spine – spongiosa	45.6	3.9	11.50	2.1	35.1	3.9	8.80	2.2
50	Thoracic spine – spongiosa	188.8	16.1	26.90	4.9	144.9	16.1	20.60	5.1
52	Lumbar spine – spongiosa	143.9	12.3	23.40	4.3	110.7	12.3	18.00	4.4
54	Sacrum – spongiosa	115.9	9.9	20.60	3.8	89.1	9.9	15.80	3.9
56	Sternum – spongiosa	36.3	3.1	5.50	1.0	27.9	3.1	4.30	1.1
	Totals	1170.2	100	544.4	100	899.1	100	407.50	100

Source: ICRP, 2009. Adult reference computational phantoms. ICRP Publication 110. Ann. ICRP 39(2).

[9] ICRP, 2010. *Annals of the ICRP* **40**(2-5)

Extra Considerations

- The tetrahedral (volume) phantom geometry has large *memory* requirements (~ 11 GB [10])
- Supplementary materials provided with the ICRP145 report also gives a *polygonal* (surface) mesh
  - Can be useful for visualisation or manipulation using other software
- The tetrahedral mesh compared to the polygonal mesh [10]:
  - More efficient for particle transport (~2600 times faster)
  - Less time for initialisation (~ 0.5 times)
  - Requires more memory (11 GB vs 1.3 GB)

[10] Han et. al., 2020. *Journal of Radiological Protection*, **40**(4)

Extra Considerations

- Polygonal meshes visualized:

- Since the tetrahedral mesh is generated from the polygonal mesh, the polygonal mesh can be used for surface calculations