## Patient (VoxelizedPhantom) positioning in Gate

Gate allows CTs to be imported directly as a VoxelizedPhantom object. For dose calculations using the TPSPencilBeam source, and plan / source description file formalism, we need the isocentre to be at the World origin. There is a command to shift the plan’s isocentre to World origin (/TranslateTheImageAtThisIsoCenter) but I found that this did strange things when combined with patient rotations. We thus need to calculate the required shifts and rotations to apply to the image. Important points to note:

1. Gate will place the geometric centre of the image at the World origin.
2. Gate will import the patient such that their dicom coordinates align with the Gate World coordinates (see Figure below). Note that this is always true – for different patient setups (HFS, FFP etc.) the Offset and TransformMatrix fields in the mhd file, read from the dicom images, change sign such that it gives a consistent alignment.
3. Rotations of the image are specified in the image, not World coordinates. They can be specified around any axis that passes through the image origin (centre) but rotations about another point (i.e. the isocentre) are not possible.
4. Gate only allows one translation and one rotation to be specified in a mac file; they cannot be chained together. This means that if we need to rotate the patient multiple times (180 about z to prone position, 180 about y to feet first and a 20 degree couch kick) then we need to express this as a rotation by a single angle about a single axis.

The above means that we must take care in the order of our operations. If we shift the patient to the isocentre and then rotate, since we can only rotate about the image centre we will subsequently rotate the isocentre off of the world origin. Hence we must:

1. Calculate the translation required to position the isocentre at the World origin without any patient rotations
2. Calculate the net rotation matrix of all rotations
3. Calculate from this a single angle and axis of rotation
4. Perform the rotation in Gate
5. Transform the shift vector calculated in step 1 by our rotation matrix of step 2
6. Apply this transformed shift so as to translate the isocentre to the World origin

This may seem convoluted but it’s the only way I found how to do what we needed within Gate. Another option would be to generate the mhd file and then modify the Offset and TransformMatrix fields appropriately depending on the set up. This would mean we would only have to perform rotations for couch kicks.

### Calculating the axis and angle of rotation

We have the usual 3D rotation matrices:

corresponding to the axes in the Figure below. Hence if we wish to simulate a patient in a feet first prone (FFP) position, with a couch angle of 20 degrees, the total rotation would be given by:

where the first 180 rotation about *y* flips the patient to feet first, the 180 about *z* flips them to prone, and the final rotation about *y* is the couch kick.

Every 3D rotation is defined by its axis and its angle; in Gate we may only specify a single rotation. From <https://en.wikipedia.org/wiki/Rotation_matrix>, if we have a rotation matrix *R*,

then some vetor **u** which is parallel to the rotation axis can be calculated as

where

Hence in our Gate simulation we can use the normalised vector **u**/|**u**| with the /setRotationAxis command and with the /setRotationAngle command. Note that this method for determining will not work if the rotation matrix is symmetric. To avoid this being the case, whenever I perform any rotation I never use angles of 0 or 180 exactly, but add 0.001 degrees to each rotation on each axis. An alternative method to calculate the roatation angle is from the trace of the rotation matrix; care is needed with the sign of the angle:

### Translating isocentre to centre of World volume

Using the TPSPencilBeam source functionality of Gate, we must position the plan’s isocentre at the centre of our Gate world. The patient image will be imported automatically such that its geometric centre is positioned at the centre of the world. Hence we need to calculate the vector connecting the plan’s isocentre (green circle in Figure) to the image geometric centre (red circle). Hence our translation vector, is simply:

The coordinates of the isocentre are contained in the plan dicom file. We can find the coordinates of the image centre as:

where contains the dimensions of the voxels, the dimensions of the full image and where is the coordinate of the centre of the voxel with the smallest x, y and z values. The values for the can be taken from the dicom file (PixelSpacing and SliceThickness) or from the mhd file (ElementSpacing) tags. The is calculated simply by multiplying the voxel dimensions by the number of voxels in each direction.

Note that the values for need a little more care. They are calculated from the ImagePositionPatient (dicom) or Origin (mhd) tags but depend on the patient orientation. For instance, the Origin of a patient in HFS position will correspond to the centre of the voxel with the minimum x, y and z coordinates, hence will be identical to the Origin tag. In contrast, the Origin of a patient in HFP will correspond to the centre of the voxel with maximum x and y, and minimum z coordinates and hence we will need to determine manually. This is done in the corner\_voxel\_centres() method in generatefiles.py, which returns min and max *x*, *y* and *z* coordinates of the centre of all corner voxels.

Since we perform the rotation in Gate before the translation, the final translation vector to be used in the Gate mac file, , must account for all prior rotations, hence:

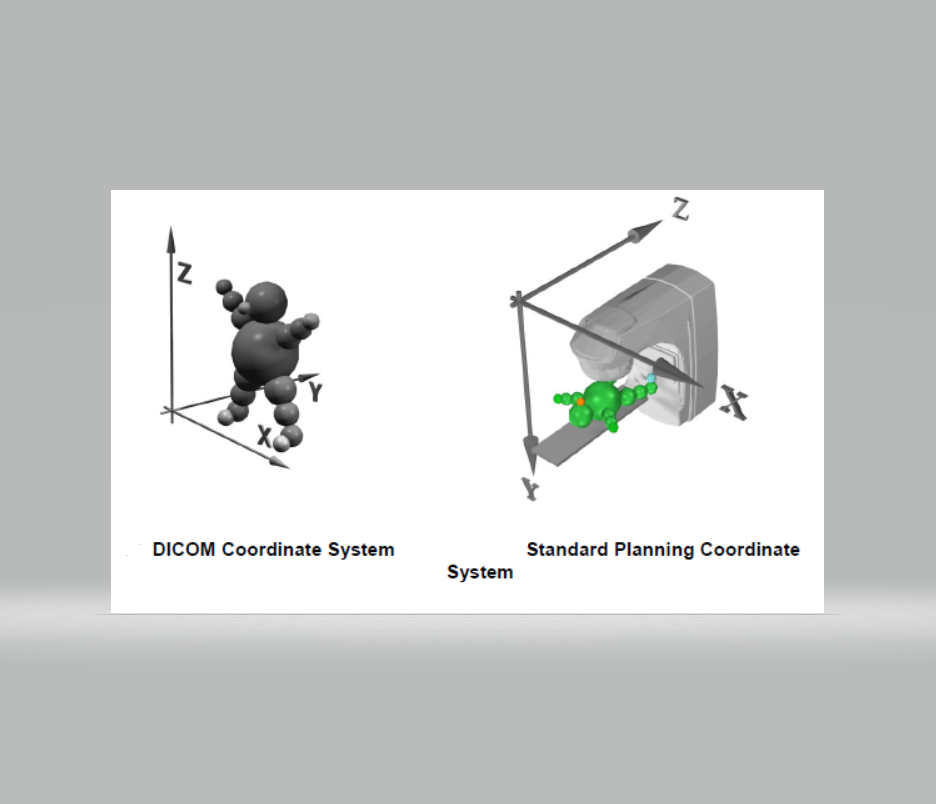


Figure: left image shows how the Gate World coordinates relate to the patient when importing any CT image as a VoxelizedPhantom. Note this is true irrespective of the patient set-up (HFS, HFP etc.) by virtue of the differing transform matrices and offsets.

Centre

Isocentre

VoxDim

ImgDim

MinVoxel

**T**

Figure: illustrating the translation vector required to position the plan’s isocentre at the World origin (assuming no image rotations).