Computational assignment 7

due on November 10, 2025

Computation of the photon dose-influence matrix

For treatment plan optimization we will discretize the radiation field into beam segments of 5 mm in size, called beamlets. The overall goal of this assignment is to generate the dose-influence matrix D_{ij} which stores the dose contributions of all beamlets j to all voxels i in the patient. This mostly amounts to running the function

`calculate_pencil_beam_dose(angle,latpos,raddepth)`

many times and rearranging the data.

1) Write a function

`create_beams(angles)`

which takes a vector of beam angles as input and returns a dictionary `beams` with the following fields

```
beams[beamNo] = {
    'angle': angle, # the incident beam angle
    'nBeamlets': len(beamletpos), # number of beamlets
    'beamletpos': beamletpos, # array of beamlet positions
    'raddepth': raddepth, # radiological depth of all voxels
    'pb': pb # dictionary of pencil beams
}
```

where `beamNo` is the number of the beam.

The beamlet positions `beamletpos` refer to the center of a 5 mm wide photon beam. The beamlet that hits the isocenter is at position 0. Beamlets should be spaced 5 mm apart, corresponding to the size of the beamlet. The number of beamlets should be large enough so that the beam covers the entire tumor. While in practice one would include only those beamlets that hit the tumor to avoid dose calculation for unnecessary beamlets, you do not have to be that efficient here. You can eyeball how many beamlets are needed and use the same number of beamlets for every beam angle.

2) Write a function

`create_Dij_matrix(beams)`

which returns a matrix D whose elements D_{ij} store the dose contributions of beamlet j to voxel i. This mostly amounts to rearranging the data in the dictionary `beams`. The dose-influence matrix D_{ij} should be arranged such that you can calculate the dose distribution d in the patient via a simple matrix multiplication

$$d = Dx \tag{1}$$

where d is a vector that concatenates the dose values in all the voxels. The size of d is thus the number of voxels. x is a vector containing the fluence of all beamlets in all beams.

Hint:

The 2D dose distribution can be converted into a dose vector using the `numpy.reshape` function.

3) Calculate the dose-influence matrix for 9 equally spaced beams. Plot the dose distribution for uniform fluence, i.e. each beamlet has the same fluence. You can then try to tweak the fluence vector x manually to improve the dose distribution.