# Example Proton Treatment Plan with Manipulated CT values

## Table of Contents





In this example we will show how to load patient data into matRad and how to setup a photon dose calculation including optimizing the beamlet intensities. Next we will apply a sequezing algorithm with a subsequent direct aperture optimization.

# Patient Data Import

Let's begin with a clear Matlab environment. Next, import the protstate case into your workspace. The patient is comprised of a 'ct' and 'cst' structure defining the CT images and the structure set. Make sure the matRad root directy with all its subdirectories is added to the Matlab search path.

```
clc,clear,close all
load('PROSTATE.mat');
```

Let's check the two variables, we have just imported. First, the 'ct' variable comprises the ct cube along with some meta information describing properties of the ct cube (cube dimensions, resolution, number of CT scenarios). Please note that mutiple ct cubes (e.g. 4D CT) can be stored in the cell array ct.cube{}

```
ct
```

```
ct =

  struct with fields:
```

```
         cube: {[184×184×90 double]}
   resolution: [1×1 struct]
      cubeDim: [184 184 90]
  numOfCtScen: 1
```

The 'cst' cell array defines volumes of interest along with information required for optimization. Each row belongs to one certain VOI, whereas each column defines different proprties. Specifically, the second and third column show the name and the type of the structure. The tpe can be set to OAR, TARGET or IGNORED. The fourth column depicts a linear index vector depicting voxels in the CT cube that are covered by the VOI. In total, 17 structures are defined in the cst

```
cst
```

```
cst =

  10×6 cell array

  Columns 1 through 5

    [0]    'BODY'               'OAR'       {1×1 cell}    [1×1 struct]
    [1]    'Bladder'            'OAR'       {1×1 cell}    [1×1 struct]
    [2]    'Lt femoral head'    'OAR'       {1×1 cell}    [1×1 struct]
    [3]    'Lymph Nodes'        'OAR'       {1×1 cell}    [1×1 struct]
    [4]    'PTV 56'             'TARGET'    {1×1 cell}    [1×1 struct]
    [5]    'PTV 68'             'TARGET'    {1×1 cell}    [1×1 struct]
    [6]    'Penile bulb'        'OAR'       {1×1 cell}    [1×1 struct]
    [7]    'Rectum'             'OAR'       {1×1 cell}    [1×1 struct]
    [8]    'Rt femoral head'    'OAR'       {1×1 cell}    [1×1 struct]
    [9]    'prostate bed'       'TARGET'    {1×1 cell}    [1×1 struct]

  Column 6

    [1×1 struct]
    [1×1 struct]
              []
              []
    [1×1 struct]
    [1×1 struct]
              []
    [1×1 struct]
              []
              []
```

# Treatment Plan

The next step is to define your treatment plan labeld as 'pln'. This structure requires input from the treatment planner and defines the most important cornerstones of your treatment plan.

First of all, we need to define what kind of radiation modality we would like to use. Possible values are photons, protons or carbon. In this case we want to use photons. Then, we need to define a treatment machine to correctly load the corresponding base data. Since we provide generic base data we set the machine to 'Genereric. By this means matRad will look for 'proton_Generic.mat' in our root directory and will use the data provided in there for dose calculation

```
pln.radiationMode = 'photons';   % either photons / protons / carbon
pln.machine       = 'Generic';
```

Define the flavour of biological optimization for treatment planning along with the quantity that should be used for optimizaion. Possible values are (none: physical optimization; const_RBExD: constant RBE of 1.1; LEMIV_effect: effect-based optimization; LEMIV_RBExD: optimization of RBE-weighted dose. As we are using photons, simply set the parameter to 'none' thereby indicating the physical dose should be optimized.

```
pln.bioOptimization = 'none';
```

Now we have to set some beam parameters. We can define multiple beam angles for the treatment and pass these to the plan as a vector.matRad will then interpret the vector as multiple beams. In this case, we define linear spaced beams from 0 degree to 359 degree in 30 degree steps. This results in 12 beams. All corresponding couch angles are set to 0 at this point. Moreover, we set the bixelWidth to 5, which results in a beamlet size of 5 x 5 mm. The number of fractions is set to 30. Be advised that matRad is always optimizing the fraction dose.

```
pln.gantryAngles    = [0:40:359];
pln.couchAngles     = [0 0 0 0 0 0 0 0 0];
pln.bixelWidth      = 5;
pln.numOfFractions  = 30;
```

Obtain the number of beams and voxels from the existing variables and calculate the iso-center which is per default the mass of gravity of all target voxels.

```
pln.numOfBeams      = numel(pln.gantryAngles);
pln.numOfVoxels     = prod(ct.cubeDim);
pln.voxelDimensions = ct.cubeDim;
pln.isoCenter       = ones(pln.numOfBeams,1) *
 matRad_getIsoCenter(cst,ct,0);
```

Enable sequencing and direct aperture optimization (DAO).

```
pln.runSequencing = 1;
pln.runDAO        = 1;
```

y listo our treatment plan is ready. Lets have a look at it:

```
pln
```

```
pln =

  struct with fields:

      radiationMode: 'photons'
            machine: 'Generic'
    bioOptimization: 'none'
       gantryAngles: [0 40 80 120 160 200 240 280 320]
        couchAngles: [0 0 0 0 0 0 0 0 0]
```

```
        bixelWidth: 5
    numOfFractions: 30
        numOfBeams: 9
       numOfVoxels: 3047040
   voxelDimensions: [184 184 90]
         isoCenter: [9×3 double]
     runSequencing: 1
            runDAO: 1
```

# Generatet Beam Geometry STF

This acronym stands for steering file and comprises the beam geomtry along with the ray and pencil beam positions

```
stf = matRad_generateStf(ct,cst,pln);
```

```
matRad: Generating stf struct... Progress: 100.00 %
```

# Dose Calculation

Lets generate dosimetric information by pre-computing dose influence matrices for inverse planning.

```
dij = matRad_calcPhotonDose(ct,stf,pln,cst);
```

```
matRad: Photon dose calculation...
Beam 1 of 9:
matRad: calculate radiological depth cube...done
                   SSD = 870mm
matRad: Uniform primary photon fluence -> pre-compute kernel
 convolution for SSD = 870 mm ...
Progress: 100.00 %
Beam 2 of 9:
matRad: calculate radiological depth cube...done
                   SSD = 858mm
matRad: Uniform primary photon fluence -> pre-compute kernel
 convolution for SSD = 858 mm ...
Progress: 100.00 %
Beam 3 of 9:
matRad: calculate radiological depth cube...done
                   SSD = 830mm
matRad: Uniform primary photon fluence -> pre-compute kernel
 convolution for SSD = 830 mm ...
Progress: 100.00 %
Beam 4 of 9:
matRad: calculate radiological depth cube...done
                   SSD = 821mm
matRad: Uniform primary photon fluence -> pre-compute kernel
 convolution for SSD = 821 mm ...
Progress: 100.00 %
Beam 5 of 9:
matRad: calculate radiological depth cube...done
                   SSD = 898mm
```

```
matRad: Uniform primary photon fluence -> pre-compute kernel
 convolution for SSD = 898 mm ...
Progress: 100.00 %
Beam 6 of 9:
matRad: calculate radiological depth cube...done
                   SSD = 898mm
matRad: Uniform primary photon fluence -> pre-compute kernel
 convolution for SSD = 898 mm ...
Progress: 100.00 %
Beam 7 of 9:
matRad: calculate radiological depth cube...done
                   SSD = 821mm
matRad: Uniform primary photon fluence -> pre-compute kernel
 convolution for SSD = 821 mm ...
Progress: 100.00 %
Beam 8 of 9:
matRad: calculate radiological depth cube...done
                   SSD = 832mm
matRad: Uniform primary photon fluence -> pre-compute kernel
 convolution for SSD = 832 mm ...
Progress: 100.00 %
Beam 9 of 9:
matRad: calculate radiological depth cube...done
                   SSD = 858mm
matRad: Uniform primary photon fluence -> pre-compute kernel
 convolution for SSD = 858 mm ...
Progress: 100.00 %
```

# Inverse Planning for IMRT

This function optimizes the fluence of the beam, giving back the weights used to scale the number of photons in every ray of the beam, improving the accuracy of the simulation

```
resultGUI = matRad_fluenceOptimization(dij,cst,pln);
```

```

******************************************************************************
This program contains Ipopt, a library for large-scale nonlinear
 optimization.
 Ipopt is released as open source code under the Eclipse Public
 License (EPL).
         For more information visit http://projects.coin-or.org/Ipopt
******************************************************************************

This is Ipopt version 3.11.8, running with linear solver ma57.

Number of nonzeros in equality constraint Jacobian...:        0
Number of nonzeros in inequality constraint Jacobian.:        0
Number of nonzeros in Lagrangian Hessian.............:        0

Total number of variables............................:     4159
                     variables with only lower bounds:     4159
                variables with lower and upper bounds:        0
                     variables with only upper bounds:        0
```

```
Total number of equality constraints.................:        0
Total number of inequality constraints...............:        0
        inequality constraints with only lower bounds:        0
   inequality constraints with lower and upper bounds:        0
        inequality constraints with only upper bounds:        0

iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du
 alpha_pr  ls
   0 3.4661412e+002 0.00e+000 7.16e+000   0.0 0.00e+000    -  0.00e
+000 0.00e+000   0
   1 1.1285377e+002 0.00e+000 4.43e+000  -0.2 1.75e+000    -
 7.80e-001 1.07e-001f  2
   2 5.1987150e+001 0.00e+000 1.81e+000  -1.2 8.75e-002    -
 9.85e-001 1.00e+000f  1
   3 3.8558482e+001 0.00e+000 1.47e+000  -2.2 4.43e-002    -
 9.98e-001 1.00e+000f  1
   4 2.2821592e+001 0.00e+000 7.99e-001  -2.8 1.05e-001    -  1.00e
+000 1.00e+000f  1
   5 1.4778224e+001 0.00e+000 4.22e-001  -3.6 1.08e-001    -  1.00e
+000 1.00e+000f  1
   6 1.2551932e+001 0.00e+000 5.96e-001  -4.3 8.34e-002    -  1.00e
+000 1.00e+000f  1
   7 1.0923807e+001 0.00e+000 2.79e-001  -5.2 3.76e-002    -  1.00e
+000 1.00e+000f  1
   8 1.0250391e+001 0.00e+000 2.77e-001  -6.3 4.15e-002    -  1.00e
+000 6.86e-001f  1
   9 1.0218997e+001 0.00e+000 4.01e-001  -7.3 7.94e-002    -  1.00e
+000 1.97e-002f  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du
 alpha_pr  ls
  10 9.8446556e+000 0.00e+000 2.34e-001  -7.9 1.49e-001    -  1.00e
+000 1.25e-001f  1
  11 9.7032685e+000 0.00e+000 1.05e+000  -4.9 1.54e-001    -
 6.56e-001 5.13e-002f  1
  12 9.1408187e+000 0.00e+000 2.36e-001  -5.4 1.80e-001    -  1.00e
+000 2.05e-001f  1
  13 9.0369091e+000 0.00e+000 1.25e+000  -6.2 1.88e-001    -  1.00e
+000 3.98e-002f  1
  14 8.7695825e+000 0.00e+000 3.72e-001  -6.6 1.89e-001    -  1.00e
+000 1.15e-001f  1
  15 8.2816682e+000 0.00e+000 5.21e-001  -7.0 2.46e-001    -  1.00e
+000 2.05e-001f  1
  16 8.2668034e+000 0.00e+000 9.13e-001  -7.4 1.47e-001    -  1.00e
+000 1.06e-002f  1
  17 8.0245735e+000 0.00e+000 3.76e-001  -7.9 1.90e-001    -  1.00e
+000 1.52e-001f  1
  18 7.9100127e+000 0.00e+000 9.21e-001  -5.5 2.30e-001    -  1.00e
+000 6.08e-002f  1
  19 7.7503472e+000 0.00e+000 4.52e-001  -3.4 1.76e-001    -
 5.29e-001 1.22e-001f  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du
 alpha_pr  ls
  20 7.6365791e+000 0.00e+000 9.69e-001  -3.6 1.73e-001    -
 9.99e-001 1.03e-001f  1
```

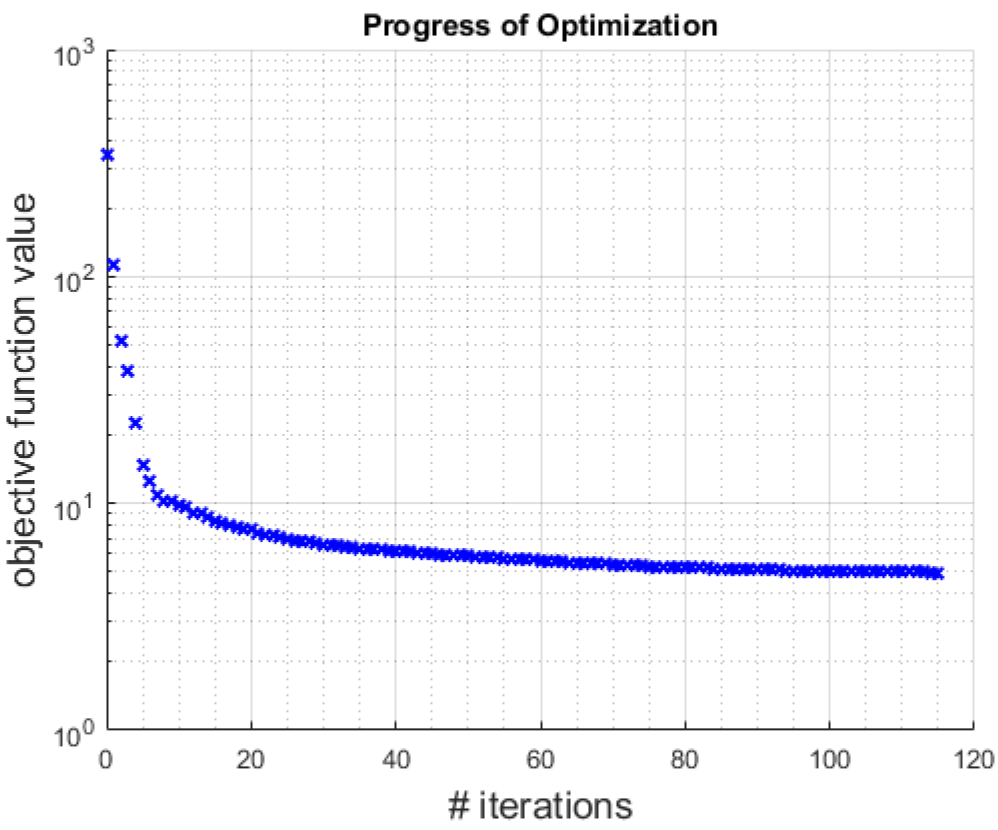
```
  21 7.4291240e+000 0.00e+000 1.01e+000  -2.5 2.24e-001    -
 7.76e-001 4.64e-001f  1
  22 7.2749461e+000 0.00e+000 1.07e+000  -3.0 3.53e-002    -
 9.59e-001 7.29e-001f  1
  23 7.1787521e+000 0.00e+000 7.74e-001  -3.0 4.57e-002    -
 9.36e-001 5.25e-001f  1
  24 7.0891227e+000 0.00e+000 6.03e-001  -3.8 7.14e-002    -
 8.98e-001 3.67e-001f  1
  25 6.9572054e+000 0.00e+000 5.63e-001  -4.1 9.50e-002    -
 9.53e-001 4.72e-001f  1
  26 6.8498252e+000 0.00e+000 1.60e-001  -4.8 1.34e-001    -
 8.93e-001 3.18e-001f  1
  27 6.8238374e+000 0.00e+000 4.42e-001  -4.1 7.68e-002    -
 6.77e-001 1.25e-001f  1
  28 6.7606796e+000 0.00e+000 1.78e+000  -2.9 4.58e-001    -
 5.45e-001 5.22e-001f  1
  29 6.6997401e+000 0.00e+000 4.02e-001  -4.0 7.31e-002    -
 9.22e-001 3.01e-001f  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du
 alpha_pr  ls
  30 6.6007548e+000 0.00e+000 2.31e-001  -3.0 2.23e-001    -
 5.89e-001 1.00e+000f  1
  31 6.6056241e+000 0.00e+000 3.68e-001  -2.9 1.51e-001    -
 7.52e-001 1.00e+000f  1
  32 6.5451434e+000 0.00e+000 1.49e-001  -3.0 1.47e-001    -
 8.18e-001 1.00e+000f  1
  33 6.4695686e+000 0.00e+000 1.02e-001  -3.2 2.32e-001    -
 9.94e-001 7.84e-001f  1
  34 6.3839884e+000 0.00e+000 4.03e-001  -3.5 1.09e-001    -
 9.98e-001 6.47e-001f  1
  35 6.3498844e+000 0.00e+000 3.84e-001  -4.1 5.68e-002    -
 8.56e-001 3.12e-001f  1
  36 6.2748030e+000 0.00e+000 2.70e-001  -4.3 8.67e-002    -
 7.92e-001 5.02e-001f  1
  37 6.2496128e+000 0.00e+000 2.66e-001  -4.4 7.02e-002    -
 9.92e-001 1.89e-001f  1
  38 6.2377272e+000 0.00e+000 3.22e-001  -3.1 1.76e-001    -
 3.74e-001 8.72e-001f  1
  39 6.1926934e+000 0.00e+000 4.01e-001  -3.6 7.29e-002    -
 9.38e-001 3.10e-001f  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du
 alpha_pr  ls
  40 6.1501301e+000 0.00e+000 1.79e-001  -3.1 5.00e-002    -
 8.75e-001 1.00e+000f  1
  41 6.1374788e+000 0.00e+000 1.26e-001  -3.1 4.75e-002    -
 7.95e-001 1.00e+000f  1
  42 6.1049866e+000 0.00e+000 1.19e-001  -4.0 6.05e-002    -  1.00e
+000 3.99e-001f  1
  43 6.0590251e+000 0.00e+000 1.75e-001  -4.2 1.30e-001    -
 9.99e-001 3.67e-001f  1
  44 6.0421920e+000 0.00e+000 4.28e-001  -5.3 3.40e-002    -  1.00e
+000 1.99e-001f  1
  45 5.9901741e+000 0.00e+000 3.07e-001  -3.9 6.74e-002    -
 4.61e-001 3.55e-001f  1
```

```
 46 5.9533565e+000 0.00e+000 2.27e-001  -4.0 5.12e-002    -
5.16e-001 3.62e-001f  1
 47 5.9191000e+000 0.00e+000 1.97e-001  -3.7 3.71e-002    -
4.02e-001 6.44e-001f  1
 48 5.9054118e+000 0.00e+000 1.84e-001  -4.6 4.62e-002    -
5.57e-001 1.97e-001f  1
 49 5.8865224e+000 0.00e+000 1.83e-001  -4.7 6.49e-002    -  1.00e
+000 1.83e-001f  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du
alpha_pr  ls
 50 5.8623836e+000 0.00e+000 2.23e-001  -4.8 8.15e-002    -
8.68e-001 1.90e-001f  1
 51 5.8313624e+000 0.00e+000 1.98e-001  -5.1 1.04e-001    -
4.34e-001 1.97e-001f  1
 52 5.8166188e+000 0.00e+000 2.25e-001  -6.1 9.40e-002    -
4.91e-001 1.05e-001f  1
 53 5.7719550e+000 0.00e+000 1.50e-001  -4.1 1.21e-001    -
9.64e-001 2.43e-001f  1
 54 5.7573151e+000 0.00e+000 1.85e-001  -3.9 5.99e-002    -
4.38e-001 1.36e-001f  1
 55 5.7285898e+000 0.00e+000 1.73e-001  -4.1 7.20e-002    -
3.12e-001 2.44e-001f  1
 56 5.7022704e+000 0.00e+000 1.14e-001  -3.7 6.78e-002    -
4.53e-001 2.30e-001f  1
 57 5.6837254e+000 0.00e+000 1.20e-001  -9.8 9.03e-002    -
3.25e-001 1.30e-001f  1
 58 5.6549900e+000 0.00e+000 2.20e-001  -4.5 1.01e-001    -
6.55e-001 1.91e-001f  1
 59 5.6272405e+000 0.00e+000 1.93e-001  -4.4 9.19e-002    -
8.97e-001 2.13e-001f  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du
alpha_pr  ls
 60 5.6207900e+000 0.00e+000 1.94e-001  -4.7 4.98e-002    -
2.98e-001 7.86e-002f  1
 61 5.5860577e+000 0.00e+000 1.59e-001  -4.2 7.76e-002    -
7.96e-001 3.09e-001f  1
 62 5.5534160e+000 0.00e+000 7.26e-002  -3.9 6.38e-002    -
7.58e-001 3.95e-001f  1
 63 5.5350914e+000 0.00e+000 1.21e-001  -4.0 5.54e-002    -
4.39e-001 2.34e-001f  1
 64 5.5090555e+000 0.00e+000 8.39e-002  -3.5 4.53e-002    -
3.90e-001 4.91e-001f  1
 65 5.4918630e+000 0.00e+000 1.11e-001  -4.4 6.67e-002    -
5.38e-001 1.97e-001f  1
 66 5.4773816e+000 0.00e+000 2.16e-001  -5.0 5.71e-002    -
5.04e-001 2.13e-001f  1
 67 5.4554934e+000 0.00e+000 1.29e-001  -4.7 7.95e-002    -
3.98e-001 2.46e-001f  1
 68 5.4425196e+000 0.00e+000 1.56e-001  -4.5 1.06e-001    -
7.11e-001 1.07e-001f  1
 69 5.4166954e+000 0.00e+000 1.93e-001  -5.2 8.28e-002    -
2.23e-001 2.98e-001f  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du
alpha_pr  ls
```

```
 70 5.3972462e+000 0.00e+000 1.17e-001  -4.6 1.16e-001    -
9.76e-001 1.76e-001f  1
 71 5.3801751e+000 0.00e+000 3.47e-001  -3.7 7.46e-002    -
3.74e-001 4.70e-001f  1
 72 5.3718011e+000 0.00e+000 1.18e-001  -3.5 1.94e-002    -
6.48e-001 1.00e+000f  1
 73 5.3466649e+000 0.00e+000 1.18e-001  -4.0 7.45e-002    -
5.09e-001 5.75e-001f  1
 74 5.3104917e+000 0.00e+000 5.85e-002  -3.9 1.24e-001    -
7.40e-001 4.45e-001f  1
 75 5.2886999e+000 0.00e+000 5.37e-002  -3.9 1.14e-001    -
7.49e-001 3.12e-001f  1
 76 5.2719956e+000 0.00e+000 1.16e-001  -3.9 6.57e-002    -
5.26e-001 4.35e-001f  1
 77 5.2552196e+000 0.00e+000 1.80e-001  -4.4 7.26e-002    -
9.22e-001 3.10e-001f  1
 78 5.2393772e+000 0.00e+000 1.40e-001  -4.3 5.93e-002    -
7.06e-001 3.18e-001f  1
 79 5.2324785e+000 0.00e+000 7.10e-002  -3.7 2.84e-002    -
6.40e-001 8.27e-001f  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du
alpha_pr  ls
 80 5.2220768e+000 0.00e+000 1.43e-001  -3.8 2.04e-002    -
6.20e-001 5.85e-001f  1
 81 5.2114431e+000 0.00e+000 1.66e-001  -5.2 2.70e-002    -
9.17e-001 4.25e-001f  1
 82 5.2008648e+000 0.00e+000 5.57e-002  -3.9 1.53e-002    -
8.17e-001 1.00e+000f  1
 83 5.1845365e+000 0.00e+000 4.57e-002  -3.9 3.29e-002    -
6.11e-001 6.44e-001f  1
 84 5.1666598e+000 0.00e+000 1.18e-001  -4.1 2.33e-002    -
7.73e-001 7.19e-001f  1
 85 5.1535799e+000 0.00e+000 1.37e-001  -4.8 3.26e-002    -
9.34e-001 3.60e-001f  1
 86 5.1459587e+000 0.00e+000 1.16e-001  -5.4 5.07e-002    -
7.26e-001 1.39e-001f  1
 87 5.1321947e+000 0.00e+000 7.98e-002  -5.1 5.43e-002    -
3.51e-001 2.45e-001f  1
 88 5.1212454e+000 0.00e+000 1.07e-001  -5.5 6.08e-002    -
7.72e-001 1.90e-001f  1
 89 5.1188207e+000 0.00e+000 1.71e-001  -3.9 8.56e-002    -
1.48e-001 6.91e-001f  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du
alpha_pr  ls
 90 5.1080404e+000 0.00e+000 7.61e-002  -5.0 3.92e-002    -  1.00e
+000 3.21e-001f  1
 91 5.0986904e+000 0.00e+000 1.16e-001  -4.2 1.67e-002    -
6.53e-001 9.76e-001f  1
 92 5.0928655e+000 0.00e+000 7.00e-002  -4.3 1.43e-002    -
7.53e-001 6.53e-001f  1
 93 5.0844718e+000 0.00e+000 9.84e-002  -4.3 1.94e-002    -
5.73e-001 7.35e-001f  1
 94 5.0785926e+000 0.00e+000 6.07e-002  -4.5 4.36e-002    -
7.61e-001 2.16e-001f  1
```

```
  95 5.0682354e+000 0.00e+000 6.40e-002  -4.8 6.90e-002    -
 6.17e-001 2.43e-001f  1
  96 5.0635705e+000 0.00e+000 6.35e-002  -6.7 4.29e-002    -
 2.44e-001 1.63e-001f  1
  97 5.0555460e+000 0.00e+000 5.98e-002  -4.9 7.56e-002    -
 5.93e-001 1.68e-001f  1
  98 5.0561657e+000 0.00e+000 1.28e-001  -3.9 4.66e-002    -
 4.21e-001 4.76e-001f  1
  99 5.0489086e+000 0.00e+000 1.25e-001  -4.3 3.29e-002    -
 8.03e-001 4.35e-001f  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du
 alpha_pr  ls
 100 5.0440120e+000 0.00e+000 8.87e-002  -4.4 2.64e-002    -
 6.25e-001 4.82e-001f  1
 101 5.0365704e+000 0.00e+000 5.85e-002  -4.4 4.47e-002    -  1.00e
+000 4.02e-001f  1
 102 5.0265466e+000 0.00e+000 3.93e-002  -4.4 6.01e-002    -
 7.06e-001 4.34e-001f  1
 103 5.0198731e+000 0.00e+000 7.88e-002  -4.5 3.63e-002    -
 5.13e-001 4.70e-001f  1
 104 5.0136560e+000 0.00e+000 3.47e-002  -4.2 5.11e-002    -
 4.73e-001 4.59e-001f  1
 105 5.0103070e+000 0.00e+000 5.37e-002  -4.7 3.86e-002    -
 4.64e-001 2.35e-001f  1
 106 5.0044447e+000 0.00e+000 7.06e-002  -4.8 5.08e-002    -
 5.40e-001 3.28e-001f  1
 107 4.9997208e+000 0.00e+000 6.20e-002  -4.8 5.71e-002    -
 8.00e-001 2.39e-001f  1
 108 4.9963062e+000 0.00e+000 7.39e-002 -10.7 5.83e-002    -
 3.65e-001 1.59e-001f  1
 109 4.9906377e+000 0.00e+000 8.38e-002  -5.8 9.08e-002    -
 6.99e-001 1.76e-001f  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du
 alpha_pr  ls
 110 4.9857837e+000 0.00e+000 7.71e-002  -5.0 5.06e-002    -
 4.29e-001 2.52e-001f  1
 111 4.9874976e+000 0.00e+000 1.52e-001  -4.2 2.30e-002    -
 6.17e-001 1.00e+000f  1
 112 4.9811097e+000 0.00e+000 3.12e-002  -4.5 3.87e-002    -
 7.60e-001 5.43e-001f  1
 113 4.9782173e+000 0.00e+000 3.99e-002  -4.4 2.27e-002    -
 6.57e-001 5.59e-001f  1
 114 4.9755139e+000 0.00e+000 6.65e-002  -4.7 3.54e-002    -
 7.97e-001 3.03e-001f  1
 115 4.9714511e+000 0.00e+000 4.96e-002  -4.3 5.26e-002    -
 4.82e-001 4.47e-001f  1

Number of Iterations....: 115

                                   (scaled)                 (unscaled)
Objective...............:  4.9714510585804295e+000
 4.9714510585804295e+000
Dual infeasibility......:  4.9615059103684862e-002
 4.9615059103684862e-002
```

```
Constraint violation....:  0.0000000000000000e+000
 0.0000000000000000e+000
Complementarity.........:  1.1423591918356079e-004
 1.1423591918356079e-004
Overall NLP error.......:  4.9615059103684862e-002
 4.9615059103684862e-002


Number of objective function evaluations             = 121
Number of objective gradient evaluations             = 116
Number of equality constraint evaluations            = 0
Number of inequality constraint evaluations          = 0
Number of equality constraint Jacobian evaluations   = 0
Number of inequality constraint Jacobian evaluations = 0
Number of Lagrangian Hessian evaluations             = 0
Total CPU secs in IPOPT (w/o function evaluations)   =      3.041
Total CPU secs in NLP function evaluations           =     80.099

EXIT: Solved To Acceptable Level.
Calculating final cubes...
```




# Sequencing

This is a multileaf collimator leaf sequencing algorithm that is used in order to modulate the intensity of the beams with multiple static segments, so that translates each intensity map into a set of deliverable aperture shapes; according to Siochi (1999).

```
resultGUI = matRad_siochiLeafSequencing(resultGUI,stf,dij,5);
```

# DAO - Direct Aperture Optimization

The Direct Aperture Optimization is an automated planning system, only possibble for photons in which we bypass the traditional intensity optimization, and instead directly optimize the shapes and the weights of the apertures. This technique allows the user to specify the maximum number of apertures per beam direction, and hence provides significant control over the complexity of the treatment delivery.

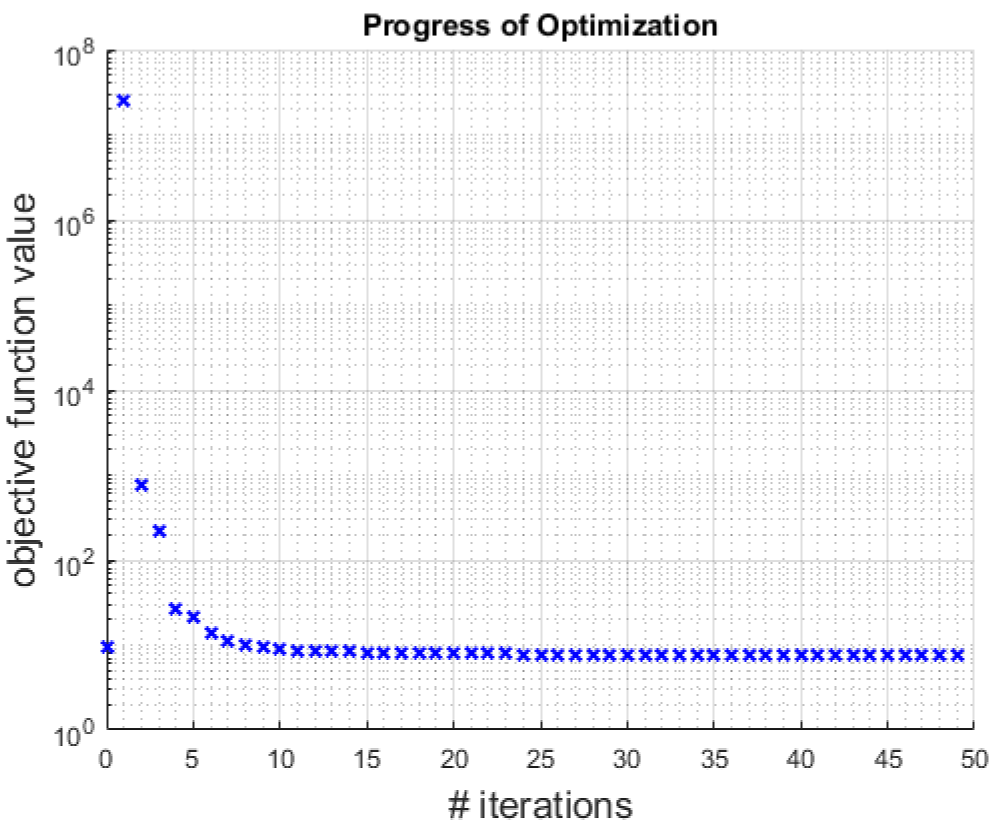
```
resultGUI =
 matRad_directApertureOptimization(dij,cst,resultGUI.apertureInfo,resultGUI,pln);
matRad_visApertureInfo(resultGUI.apertureInfo);
```

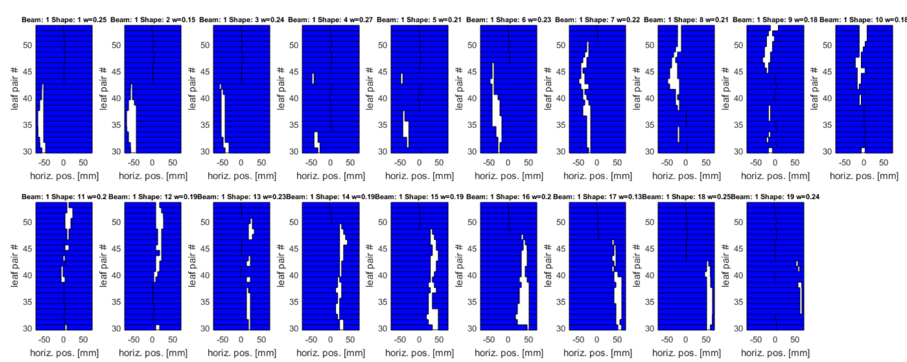
```

******************************************************************************
This program contains Ipopt, a library for large-scale nonlinear
 optimization.
 Ipopt is released as open source code under the Eclipse Public
 License (EPL).
         For more information visit http://projects.coin-or.org/Ipopt
******************************************************************************

This is Ipopt version 3.11.8, running with linear solver ma57.

Number of nonzeros in equality constraint Jacobian...:        0
Number of nonzeros in inequality constraint Jacobian.:     7536
Number of nonzeros in Lagrangian Hessian.............:        0

Total number of variables............................:     7693
                     variables with only lower bounds:      157
                variables with lower and upper bounds:     7536
                     variables with only upper bounds:        0
Total number of equality constraints.................:        0
Total number of inequality constraints...............:     3768
        inequality constraints with only lower bounds:     3768
   inequality constraints with lower and upper bounds:        0
        inequality constraints with only upper bounds:        0

iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du
 alpha_pr  ls
   0 9.3201067e+000 0.00e+000 1.24e+001   0.0 0.00e+000    -  0.00e
+000 0.00e+000   0
   1 2.4980518e+007 0.00e+000 1.20e+005   0.9 8.08e+000    -  1.00e
+000 1.00e+000h  1
   2 7.7707542e+002 0.00e+000 6.27e+002   1.3 8.07e+000    -  1.00e
+000 9.93e-001f  1
   3 2.1535242e+002 0.00e+000 3.09e+002  -0.9 5.63e-002    -
 9.90e-001 1.00e+000f  1
   4 2.6416144e+001 0.00e+000 4.00e+001  -1.8 5.38e-002    -  1.00e
+000 1.00e+000f  1
   5 2.1819260e+001 0.00e+000 3.06e+001  -2.2 7.91e-003    -  1.00e
+000 1.00e+000f  1
   6 1.3604299e+001 0.00e+000 1.83e+001  -3.3 2.78e-002    -  1.00e
+000 1.00e+000f  1
```

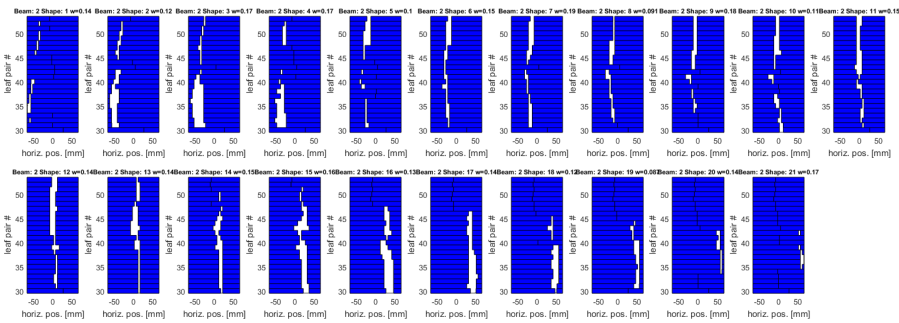
```
   7 1.1284182e+001 0.00e+000 1.06e+001  -4.1 1.38e-002    -  1.00e
+000 1.00e+000f  1
   8 9.9544180e+000 0.00e+000 1.14e+001  -5.0 1.64e-002    -  1.00e
+000 1.00e+000f  1
   9 9.3680205e+000 0.00e+000 5.98e+000  -6.2 8.06e-003    -  1.00e
+000 1.00e+000f  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du
 alpha_pr  ls
  10 9.0706510e+000 0.00e+000 5.27e+000  -7.5 5.49e-003    -  1.00e
+000 1.00e+000f  1
  11 8.6269650e+000 0.00e+000 4.09e+000  -8.9 1.15e-002    -  1.00e
+000 1.00e+000f  1
  12 8.5862352e+000 0.00e+000 7.71e+000 -10.1 1.85e-002    -  1.00e
+000 2.50e-001f  3
  13 8.4335081e+000 0.00e+000 3.66e+000 -11.0 4.49e-003    -  1.00e
+000 1.00e+000f  1
  14 8.3471026e+000 0.00e+000 2.96e+000 -11.0 3.85e-003    -  1.00e
+000 1.00e+000f  1
  15 8.2287985e+000 0.00e+000 2.86e+000 -11.0 6.44e-003    -  1.00e
+000 1.00e+000f  1
  16 8.1201189e+000 0.00e+000 2.53e+000 -11.0 6.43e-003    -  1.00e
+000 1.00e+000f  1
  17 8.1192926e+000 0.00e+000 6.24e+000 -11.0 1.12e-002    -  1.00e
+000 1.00e+000f  1
  18 8.0007011e+000 0.00e+000 1.67e+000 -11.0 3.14e-003    -  1.00e
+000 1.00e+000f  1
  19 7.9743609e+000 0.00e+000 2.02e+000 -11.0 1.34e-003    -  1.00e
+000 1.00e+000f  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du
 alpha_pr  ls
  20 7.9274457e+000 0.00e+000 1.72e+000 -11.0 3.83e-003    -  1.00e
+000 1.00e+000f  1
  21 7.9067815e+000 0.00e+000 4.15e+000 -11.0 1.02e-002    -  1.00e
+000 5.00e-001f  2
  22 7.8633135e+000 0.00e+000 1.92e+000 -11.0 3.18e-003    -  1.00e
+000 1.00e+000f  1
  23 7.8385506e+000 0.00e+000 1.12e+000 -11.0 4.02e-003    -  1.00e
+000 1.00e+000f  1
  24 7.8266177e+000 0.00e+000 1.29e+000 -11.0 2.42e-003    -  1.00e
+000 1.00e+000f  1
  25 7.7988129e+000 0.00e+000 1.51e+000 -11.0 4.59e-003    -  1.00e
+000 1.00e+000f  1
  26 7.7573793e+000 0.00e+000 2.21e+000 -11.0 7.15e-003    -  1.00e
+000 1.00e+000f  1
  27 7.7379835e+000 0.00e+000 1.40e+000 -11.0 9.69e-003    -  1.00e
+000 5.00e-001f  2
  28 7.7225740e+000 0.00e+000 7.74e-001 -11.0 2.07e-003    -  1.00e
+000 1.00e+000f  1
  29 7.7067105e+000 0.00e+000 9.10e-001 -11.0 3.01e-003    -  1.00e
+000 1.00e+000f  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du
 alpha_pr  ls
  30 7.6901640e+000 0.00e+000 1.05e+000 -11.0 3.44e-003    -  1.00e
+000 1.00e+000f  1
```

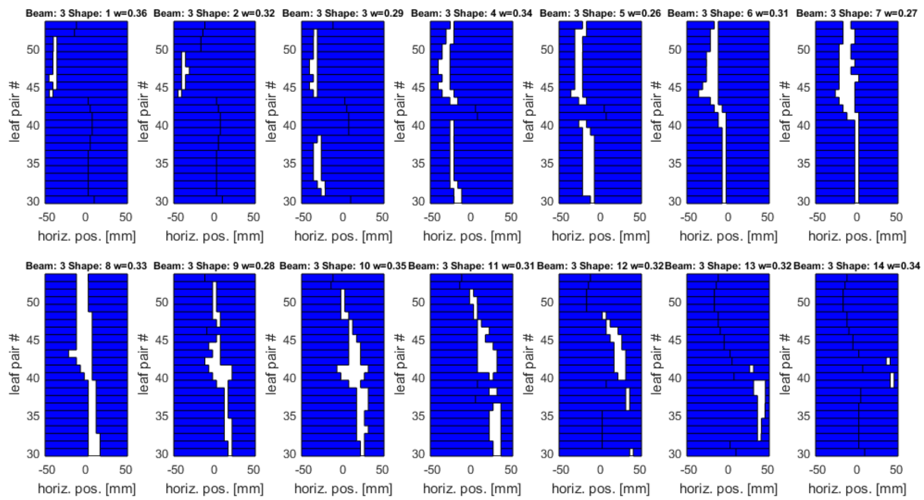
```
  31 7.6832826e+000 0.00e+000 2.86e+000 -11.0 7.84e-003    -  1.00e
+000 1.00e+000f  1
  32 7.6621433e+000 0.00e+000 7.76e-001 -11.0 1.95e-003    -  1.00e
+000 1.00e+000f  1
  33 7.6557887e+000 0.00e+000 9.23e-001 -11.0 1.72e-003    -  1.00e
+000 1.00e+000f  1
  34 7.6486952e+000 0.00e+000 1.14e+000 -11.0 2.19e-003    -  1.00e
+000 1.00e+000f  1
  35 7.6307143e+000 0.00e+000 1.05e+000 -11.0 5.53e-003    -  1.00e
+000 1.00e+000f  1
  36 7.6239447e+000 0.00e+000 1.81e+000 -11.0 1.20e-002    -  1.00e
+000 1.00e+000f  1
  37 7.6050561e+000 0.00e+000 7.90e-001 -11.0 1.71e-003    -  1.00e
+000 1.00e+000f  1
  38 7.5983235e+000 0.00e+000 7.66e-001 -11.0 1.66e-003    -  1.00e
+000 1.00e+000f  1
  39 7.5876272e+000 0.00e+000 1.11e+000 -11.0 4.88e-003    -  1.00e
+000 1.00e+000f  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du
 alpha_pr  ls
  40 7.5807086e+000 0.00e+000 1.55e+000 -11.0 4.69e-003    -  1.00e
+000 1.00e+000f  1
  41 7.5736854e+000 0.00e+000 8.92e-001 -11.0 1.89e-003    -  1.00e
+000 1.00e+000f  1
  42 7.5635183e+000 0.00e+000 6.24e-001 -11.0 5.34e-003    -  1.00e
+000 1.00e+000f  1
  43 7.5554298e+000 0.00e+000 8.68e-001 -11.0 4.19e-003    -  1.00e
+000 1.00e+000f  1
  44 7.5349268e+000 0.00e+000 2.19e+000 -11.0 1.42e-002    -  1.00e
+000 1.00e+000f  1
  45 7.5286340e+000 0.00e+000 1.49e+000 -11.0 1.14e-002    -  1.00e
+000 1.00e+000f  1
  46 7.5179916e+000 0.00e+000 4.80e-001 -11.0 4.92e-003    -  1.00e
+000 1.00e+000f  1
  47 7.5149416e+000 0.00e+000 5.49e-001 -11.0 1.19e-003    -  1.00e
+000 1.00e+000f  1
  48 7.5112564e+000 0.00e+000 1.20e+000 -11.0 2.77e-003    -  1.00e
+000 1.00e+000f  1
  49 7.5068477e+000 0.00e+000 4.38e-001 -11.0 1.58e-003    -  1.00e
+000 1.00e+000f  1

Number of Iterations....: 49

                                   (scaled)                 (unscaled)
Objective...............:  7.5068477378563934e+000
 7.5068477378563934e+000
Dual infeasibility......:  4.3820603272578390e-001
 4.3820603272578390e-001
Constraint violation....:  0.0000000000000000e+000
 0.0000000000000000e+000
Complementarity.........:  1.0000000000557194e-011
 1.0000000000557194e-011
Overall NLP error.......:  4.3820603272578390e-001
 4.3820603272578390e-001
```

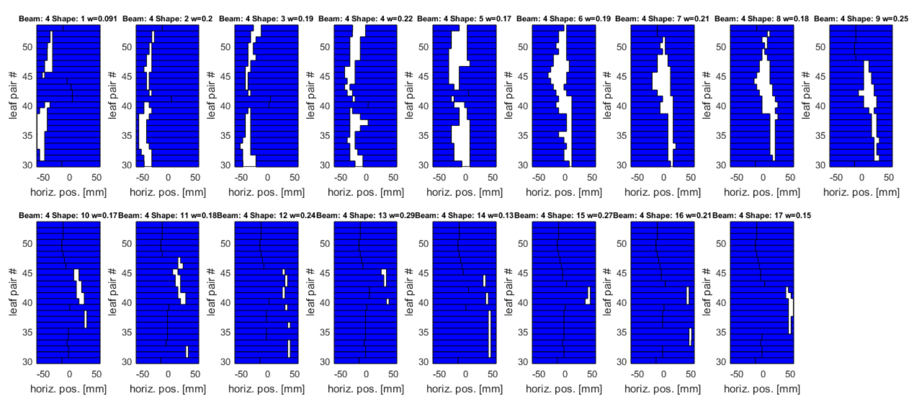
```
Number of objective function evaluations             = 55
Number of objective gradient evaluations             = 50
Number of equality constraint evaluations            = 0
Number of inequality constraint evaluations          = 55
Number of equality constraint Jacobian evaluations   = 0
Number of inequality constraint Jacobian evaluations = 50
Number of Lagrangian Hessian evaluations             = 0
Total CPU secs in IPOPT (w/o function evaluations)   =      2.047
Total CPU secs in NLP function evaluations           =     54.668

EXIT: Solved To Acceptable Level.
```

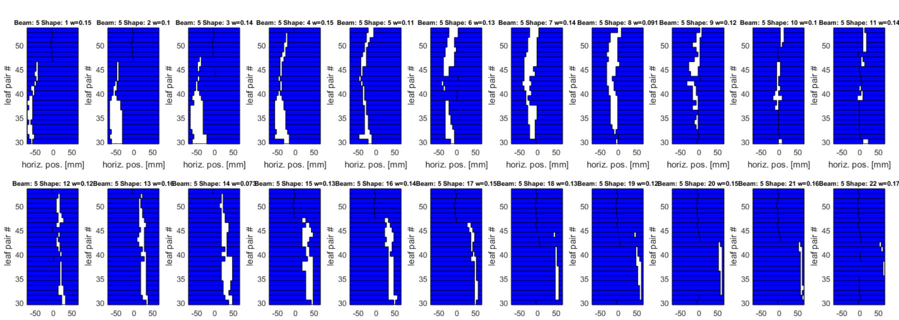


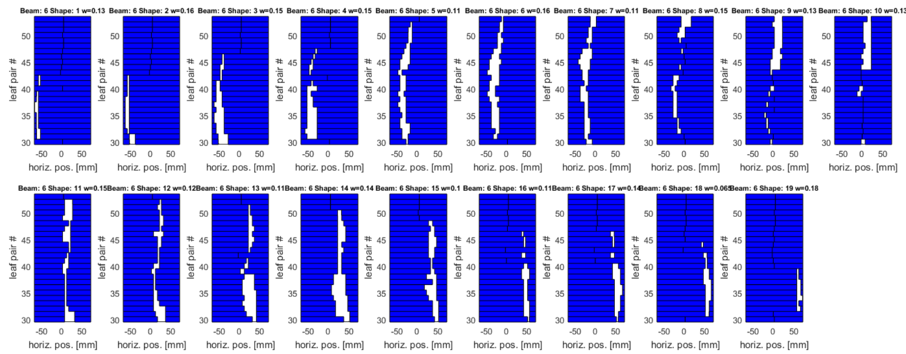



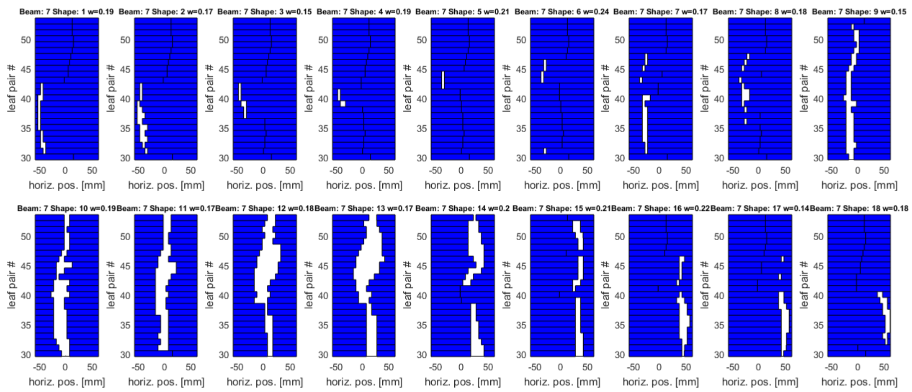

Beam: 2 Shape: 1 w=0.14
Beam: 2 Shape: 2 w=0.12
Beam: 2 Shape: 3 w=0.17
Beam: 2 Shape: 4 w=0.17
Beam: 2 Shape: 5 w=0.1
Beam: 2 Shape: 6 w=0.15
Beam: 2 Shape: 7 w=0.19
Beam: 2 Shape: 8 w=0.091
Beam: 2 Shape: 9 w=0.18
Beam: 2 Shape: 10 w=0.11
Beam: 2 Shape: 11 w=0.15
Beam: 2 Shape: 12 w=0.14
Beam: 2 Shape: 13 w=0.14
Beam: 2 Shape: 14 w=0.15
Beam: 2 Shape: 15 w=0.16
Beam: 2 Shape: 16 w=0.13
Beam: 2 Shape: 17 w=0.14
Beam: 2 Shape: 18 w=0.12
Beam: 2 Shape: 19 w=0.087
Beam: 2 Shape: 20 w=0.14
Beam: 2 Shape: 21 w=0.17
leaf pair #
horiz. pos. [mm]

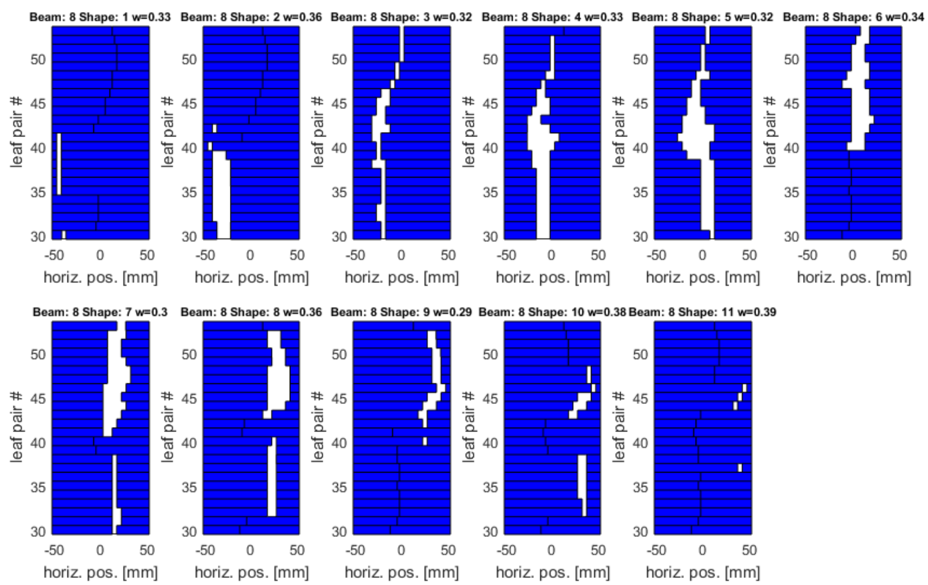

Beam: 3 Shape: 1 w=0.36
Beam: 3 Shape: 2 w=0.32
Beam: 3 Shape: 3 w=0.29
Beam: 3 Shape: 4 w=0.34
Beam: 3 Shape: 5 w=0.26
Beam: 3 Shape: 6 w=0.31
Beam: 3 Shape: 7 w=0.27
Beam: 3 Shape: 8 w=0.33
Beam: 3 Shape: 9 w=0.28
Beam: 3 Shape: 10 w=0.35
Beam: 3 Shape: 11 w=0.31
Beam: 3 Shape: 12 w=0.32
Beam: 3 Shape: 13 w=0.32
Beam: 3 Shape: 14 w=0.34
leaf pair #
horiz. pos. [mm]

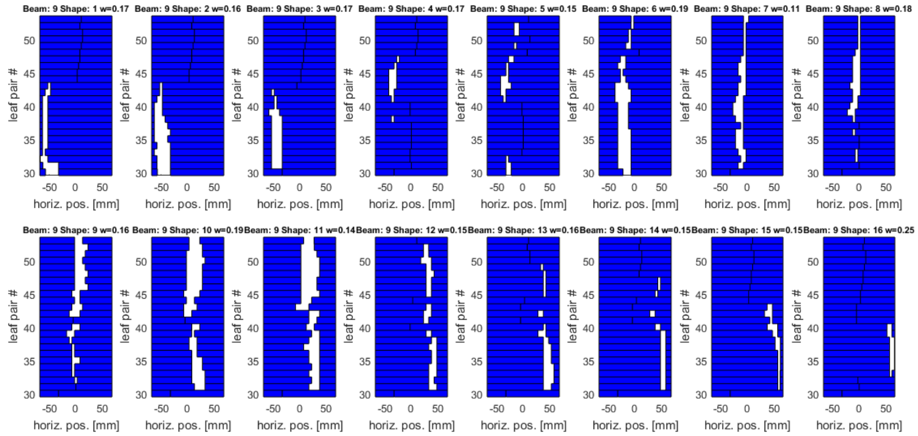

Beam: 4 Shape: 1 w=0.091
Beam: 4 Shape: 2 w=0.2
Beam: 4 Shape: 3 w=0.19
Beam: 4 Shape: 4 w=0.22
Beam: 4 Shape: 5 w=0.17
Beam: 4 Shape: 6 w=0.19
Beam: 4 Shape: 7 w=0.21
Beam: 4 Shape: 8 w=0.18
Beam: 4 Shape: 9 w=0.25
Beam: 4 Shape: 10 w=0.17
Beam: 4 Shape: 11 w=0.18
Beam: 4 Shape: 12 w=0.24
Beam: 4 Shape: 13 w=0.29
Beam: 4 Shape: 14 w=0.13
Beam: 4 Shape: 15 w=0.27
Beam: 4 Shape: 16 w=0.21
Beam: 4 Shape: 17 w=0.15
leaf pair #
horiz. pos. [mm]


Beam: 5 Shape: 1 w=0.15
Beam: 5 Shape: 2 w=0.1
Beam: 5 Shape: 3 w=0.14
Beam: 5 Shape: 4 w=0.15
Beam: 5 Shape: 5 w=0.11
Beam: 5 Shape: 6 w=0.13
Beam: 5 Shape: 7 w=0.14
Beam: 5 Shape: 8 w=0.091
Beam: 5 Shape: 9 w=0.12
Beam: 5 Shape: 10 w=0.1
Beam: 5 Shape: 11 w=0.14
Beam: 5 Shape: 12 w=0.12
Beam: 5 Shape: 13 w=0.16
Beam: 5 Shape: 14 w=0.073
Beam: 5 Shape: 15 w=0.13
Beam: 5 Shape: 16 w=0.14
Beam: 5 Shape: 17 w=0.15
Beam: 5 Shape: 18 w=0.13
Beam: 5 Shape: 19 w=0.12
Beam: 5 Shape: 20 w=0.15
Beam: 5 Shape: 21 w=0.16
Beam: 5 Shape: 22 w=0.17
leaf pair #
horiz. pos. [mm]


Beam: 6 Shape: 1 w=0.13
Beam: 6 Shape: 2 w=0.16
Beam: 6 Shape: 3 w=0.15
Beam: 6 Shape: 4 w=0.15
Beam: 6 Shape: 5 w=0.11
Beam: 6 Shape: 6 w=0.16
Beam: 6 Shape: 7 w=0.11
Beam: 6 Shape: 8 w=0.15
Beam: 6 Shape: 9 w=0.13
Beam: 6 Shape: 10 w=0.13
Beam: 6 Shape: 11 w=0.15
Beam: 6 Shape: 12 w=0.12
Beam: 6 Shape: 13 w=0.11
Beam: 6 Shape: 14 w=0.14
Beam: 6 Shape: 15 w=0.1
Beam: 6 Shape: 16 w=0.11
Beam: 6 Shape: 17 w=0.14
Beam: 6 Shape: 18 w=0.065
Beam: 6 Shape: 19 w=0.18
leaf pair #
horiz. pos. [mm]


Beam: 7 Shape: 1 w=0.19
Beam: 7 Shape: 2 w=0.17
Beam: 7 Shape: 3 w=0.15
Beam: 7 Shape: 4 w=0.19
Beam: 7 Shape: 5 w=0.21
Beam: 7 Shape: 6 w=0.24
Beam: 7 Shape: 7 w=0.17
Beam: 7 Shape: 8 w=0.18
Beam: 7 Shape: 9 w=0.15
Beam: 7 Shape: 10 w=0.19
Beam: 7 Shape: 11 w=0.17
Beam: 7 Shape: 12 w=0.18
Beam: 7 Shape: 13 w=0.17
Beam: 7 Shape: 14 w=0.2
Beam: 7 Shape: 15 w=0.21
Beam: 7 Shape: 16 w=0.22
Beam: 7 Shape: 17 w=0.14
Beam: 7 Shape: 18 w=0.18
leaf pair #
horiz. pos. [mm]


Beam: 8 Shape: 1 w=0.33
Beam: 8 Shape: 2 w=0.36
Beam: 8 Shape: 3 w=0.32
Beam: 8 Shape: 4 w=0.33
Beam: 8 Shape: 5 w=0.32
Beam: 8 Shape: 6 w=0.34
Beam: 8 Shape: 7 w=0.3
Beam: 8 Shape: 8 w=0.36
Beam: 8 Shape: 9 w=0.29
Beam: 8 Shape: 10 w=0.38
Beam: 8 Shape: 11 w=0.39
leaf pair #
horiz. pos. [mm]




# Indicator Calculation

Here we call the indicator wrapper function that is just a wrapper function to calculate multiple quality indicators like D98, D95. One basically calculates default quality indicators for a given dose distribution. If you don't see clearly the images, go in the folder 'html' in your working folder and you should look at the images, starting with the same name of this file, numbered from 3 to 11.

```
cst = matRad_indicatorWrapper(cst,pln,resultGUI);
```

```
 0                 BODY - Mean dose =  0.32 Gy +/-  0.48 Gy (Max dose
=  2.42 Gy, Min dose =  0.00 Gy)
                          D2% =  1.86 Gy, D5% =  1.32 Gy, D98% =
0.00 Gy, D95% =  0.00 Gy,
                          V0Gy = 100.00%, V0.483Gy =  27.96%,
V0.967Gy =  10.53%, V1.45Gy =   4.08%, V1.93Gy =   1.31%, V2.42Gy =
0.00%,

 1              Bladder - Mean dose =  1.37 Gy +/-  0.47 Gy (Max dose
=  2.39 Gy, Min dose =  0.35 Gy)
                          D2% =  2.31 Gy, D5% =  2.28 Gy, D98% =
0.66 Gy, D95% =  0.75 Gy,
                          V0Gy = 100.00%, V0.483Gy =  99.80%,
V0.967Gy =  82.94%, V1.45Gy =  30.66%, V1.93Gy =  16.50%, V2.42Gy =
0.00%,

 2       Lt femoral head - Mean dose =  0.45 Gy +/-  0.43 Gy (Max dose
=  1.51 Gy, Min dose =  0.00 Gy)
                          D2% =  1.26 Gy, D5% =  1.16 Gy, D98% =
0.00 Gy, D95% =  0.01 Gy,
                          V0Gy = 100.00%, V0.483Gy =  37.79%,
V0.967Gy =  17.86%, V1.45Gy =   0.08%, V1.93Gy =   0.00%, V2.42Gy =
0.00%,

 3          Lymph Nodes - Mean dose =  1.90 Gy +/-  0.10 Gy (Max dose
=  2.38 Gy, Min dose =  1.72 Gy)
                          D2% =  2.29 Gy, D5% =  2.18 Gy, D98% =
1.79 Gy, D95% =  1.81 Gy,
```

*V0Gy = 100.00%, V0.483Gy = 100.00%, V0.967Gy = 100.00%, V1.45Gy = 100.00%, V1.93Gy = 12.91%, V2.42Gy = 0.00%,*

*4 PTV 56 - Mean dose = 1.91 Gy +/- 0.13 Gy (Max dose = 2.42 Gy, Min dose = 1.66 Gy)*
*D2% = 2.30 Gy, D5% = 2.26 Gy, D98% = 1.78 Gy, D95% = 1.80 Gy,*
*V0Gy = 100.00%, V0.483Gy = 100.00%, V0.967Gy = 100.00%, V1.45Gy = 100.00%, V1.93Gy = 16.38%, V2.42Gy = 0.01%,*
*CI = 0.4824, HI = 24.48 for reference dose of 1.9 Gy*

*5 PTV 68 - Mean dose = 2.26 Gy +/- 0.05 Gy (Max dose = 2.42 Gy, Min dose = 1.94 Gy)*
*D2% = 2.35 Gy, D5% = 2.33 Gy, D98% = 2.11 Gy, D95% = 2.16 Gy,*
*V0Gy = 100.00%, V0.483Gy = 100.00%, V0.967Gy = 100.00%, V1.45Gy = 100.00%, V1.93Gy = 100.00%, V2.42Gy = 0.01%,*
*CI = 0.8942, HI = 7.74 for reference dose of 2.3 Gy*

*6 Penile bulb - Mean dose = 0.17 Gy +/- 0.06 Gy (Max dose = 0.27 Gy, Min dose = 0.10 Gy)*
*D2% = 0.26 Gy, D5% = 0.26 Gy, D98% = 0.10 Gy, D95% = 0.10 Gy,*
*V0Gy = 100.00%, V0.483Gy = 0.00%, V0.967Gy = 0.00%, V1.45Gy = 0.00%, V1.93Gy = 0.00%, V2.42Gy = 0.00%,*

*7 Rectum - Mean dose = 1.22 Gy +/- 0.46 Gy (Max dose = 2.41 Gy, Min dose = 0.04 Gy)*
*D2% = 2.24 Gy, D5% = 2.10 Gy, D98% = 0.06 Gy, D95% = 0.14 Gy,*
*V0Gy = 100.00%, V0.483Gy = 92.29%, V0.967Gy = 80.44%, V1.45Gy = 22.79%, V1.93Gy = 6.80%, V2.42Gy = 0.00%,*

*8 Rt femoral head - Mean dose = 0.47 Gy +/- 0.43 Gy (Max dose = 1.58 Gy, Min dose = 0.00 Gy)*
*D2% = 1.32 Gy, D5% = 1.20 Gy, D98% = 0.01 Gy, D95% = 0.01 Gy,*
*V0Gy = 100.00%, V0.483Gy = 41.55%, V0.967Gy = 16.69%, V1.45Gy = 0.45%, V1.93Gy = 0.00%, V2.42Gy = 0.00%,*

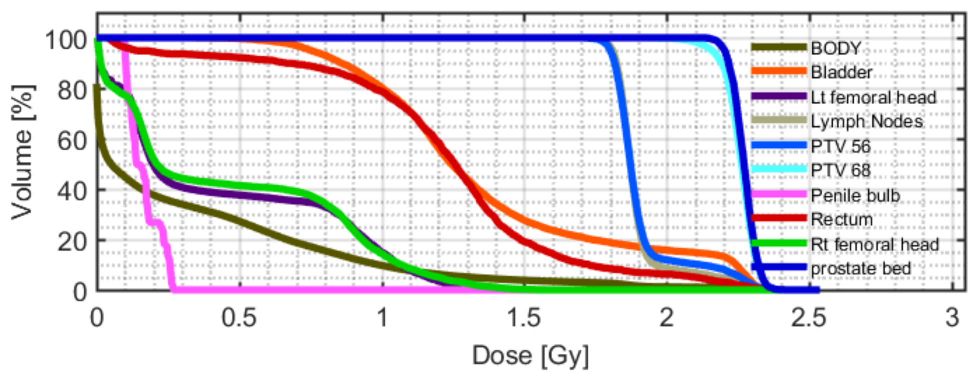
*9 prostate bed - Mean dose = 2.27 Gy +/- 0.04 Gy (Max dose = 2.42 Gy, Min dose = 2.13 Gy)*
*D2% = 2.35 Gy, D5% = 2.34 Gy, D98% = 2.18 Gy, D95% = 2.20 Gy,*

```
                          V0Gy = 100.00%, V0.483Gy = 100.00%,
V0.967Gy = 100.00%, V1.45Gy = 100.00%, V1.93Gy = 100.00%, V2.42Gy =
0.04%,
                          Warning: target has no objective that
penalizes underdosage,
```

# Show DVH and QI

This last function allow you to plot the Dose Volume Histogram and the dose statistics of all the VOIs in side the ct.

```
matRad_showDVH(cst,pln)
```




| VOIs | mean | std | max | min | D2 | D5 |
|---|---|---|---|---|---|---|
| BODY | 0.3244 | 0.4817 | 2.4166 | 0 | 1.8624 | 1.3175 |
| Bladder | 1.3661 | 0.4650 | 2.3895 | 0.3472 | 2.3089 | 2.2831 |
| Lt femoral head | 0.4468 | 0.4254 | 1.5059 | 0.0021 | 1.2590 | 1.1603 |
| Lymph Nodes | 1.8969 | 0.1034 | 2.3830 | 1.7197 | 2.2860 | 2.1835 |
| PTV 56 | 1.9081 | 0.1252 | 2.4166 | 1.6617 | 2.3018 | 2.2589 |
| PTV 68 | 2.2570 | 0.0549 | 2.4166 | 1.9356 | 2.3508 | 2.3340 |
| Penile bulb | 0.1651 | 0.0571 | 0.2690 | 0.0970 | 0.2625 | 0.2598 |
| Rectum | 1.2242 | 0.4587 | 2.4147 | 0.0382 | 2.2444 | 2.0999 |
| Rt femoral head | 0.4715 | 0.4339 | 1.5752 | 0.0020 | 1.3245 | 1.1997 |

dose statistics

*Published with MATLAB® R2017a*