Example: Proton Treatment Plan with subsequent Isocenter shift

Table of Contents

In this example we will show (i) how to load patient data into matRad (ii) how to setup a proton dose calculation (iii) how to inversely optimize the pencil beam intensities directly from command window in MATLAB. (iv) how to simulate a lateral patient displacement by shifting the iso-center (v) how to recalculated the dose considering the shifted geometry and the previously optimized pencil beam intensities (vi) how to compare the two results

Patient Data Import

Let's begin with a clear Matlab environment and import the prostate patient into your workspace

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

Treatment Plan

The next step is to define your treatment plan labeled 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 example we would like to use protons for treatment planning. Next, we need to define a treatment machine to correctly load the corresponding base data. matRad features generic base data in the file 'proton_Generic.mat'; consequently the machine has to be set accordingly

```
pln.radiationMode = 'protons';
pln.machine       = 'Generic';
```

Define the flavor of biological optimization for treatment planning along with the quantity that should be used for optimization. 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 use protons, we follow here the clinical standard and use a constant relative biological effectiveness of 1.1. Therefore we set bioOptimization to const_RBExD

```
pln.propOpt.bioOptimization = 'const_RBExD';
```

for particles it is possible to also calculate the LET disutribution alongside the physical dose. Therefore you need to activate the corresponding option during dose calculcation

```
pln.propDoseCalc.calcLET = 1;
```

Now we have to set the remaining plan parameters.

```
pln.numOfFractions        = 30;
pln.propStf.gantryAngles  = [90 270];
pln.propStf.couchAngles   = [0 0];
pln.propStf.bixelWidth    = 3;
pln.propStf.numOfBeams    = numel(pln.propStf.gantryAngles);
pln.propStf.isoCenter     = ones(pln.propStf.numOfBeams,1) *
 matRad_getIsoCenter(cst,ct,0);
pln.propOpt.runDAO        = 0;
pln.propOpt.runSequencing = 0;

% dose calculation settings
pln.propDoseCalc.doseGrid.resolution.x = 3; % [mm]
pln.propDoseCalc.doseGrid.resolution.y = 3; % [mm]
pln.propDoseCalc.doseGrid.resolution.z = 3; % [mm]
```

Generate Beam Geometry STF

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

```
matRad: Generating stf struct... Warning: Could not find HLUT
Philips-AcQSimCT-ConvolutionKernel-000000_protons.hlut in hlutLibrary
 folder.
matRad default HLUT loaded
Progress: 100.00 %
```

Dose Calculation

Lets generate dosimetric information by pre-computing dose influence matrices for unit beamlet intensities. Having dose influences available allows for subsequent inverse optimization.

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

```
Warning: Could not find HLUT
Philips-AcQSimCT-ConvolutionKernel-000000_protons.hlut in hlutLibrary
 folder.
matRad default HLUT loaded
matRad: Using a constant RBE of 1.1
matRad: Particle dose calculation...
Beam 1 of 2:
matRad: calculate radiological depth cube...done.
matRad: calculate lateral cutoff...done.
Progress: 100.00 %
Beam 2 of 2:
matRad: calculate radiological depth cube...done.
matRad: calculate lateral cutoff...done.
Progress: 100.00 %
```

Inverse Optimization for IMPT

The goal of the fluence optimization is to find a set of bixel/spot weights which yield the best possible dose distribution according to the clinical objectives and constraints underlying the radiation treatment

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

```
Optimzation initiating...
Press q to terminate the optimization...

******************************************************************************
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............................:    45367
                     variables with only lower bounds:    45367
                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 4.3797572e+002 0.00e+000 1.06e+000   0.0 0.00e+000    -  0.00e
+000 0.00e+000   0
```

```
   1 4.1078869e+002 0.00e+000 6.83e-002  -1.1 7.03e-002    -
 9.93e-001 1.00e+000f  1
   2 7.5086992e+001 0.00e+000 1.81e-002  -1.7 1.38e+000    -  1.00e
+000 1.00e+000f  1
   3 3.8671035e+001 0.00e+000 1.21e-002  -3.4 3.71e-001    -
 9.79e-001 1.00e+000f  1
   4 3.1272594e+001 0.00e+000 1.01e-002  -3.9 2.84e-001    -
 9.87e-001 1.00e+000f  1
   5 2.5003599e+001 0.00e+000 9.35e-003  -4.8 4.36e-001    -
 9.99e-001 1.00e+000f  1
   6 2.0846199e+001 0.00e+000 1.23e-002  -5.5 7.04e-001    -  1.00e
+000 1.00e+000f  1
   7 1.7591884e+001 0.00e+000 6.94e-003  -6.0 2.83e-001    -  1.00e
+000 1.00e+000f  1
   8 1.6326338e+001 0.00e+000 5.67e-003  -7.2 2.37e-001    -  1.00e
+000 1.00e+000f  1
   9 1.4806683e+001 0.00e+000 4.46e-003  -8.5 4.12e-001    -  1.00e
+000 1.00e+000f  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du
 alpha_pr  ls
  10 1.2847415e+001 0.00e+000 3.64e-003  -9.4 6.27e-001    -  1.00e
+000 1.00e+000f  1
  11 1.2215055e+001 0.00e+000 5.30e-003 -10.0 8.57e-001    -  1.00e
+000 4.32e-001f  1
  12 1.2209742e+001 0.00e+000 5.29e-003 -11.0 3.82e-001    -  1.00e
+000 3.47e-003f  1
  13 1.2205224e+001 0.00e+000 1.68e-002 -11.0 5.32e-001    -  1.00e
+000 1.87e-003f  1
  14 1.1927952e+001 0.00e+000 4.79e-003  -8.7 6.61e-001    -
 9.96e-001 9.68e-002f  1
  15 1.1917877e+001 0.00e+000 1.22e-002  -9.3 6.82e-001    -  1.00e
+000 3.48e-003f  1
  16 1.1184059e+001 0.00e+000 5.15e-003 -10.3 8.20e-001    -  1.00e
+000 2.53e-001f  1
  17 1.1175054e+001 0.00e+000 3.38e-003  -8.2 7.06e-001    -
 9.28e-001 3.97e-003f  1
  18 1.1104923e+001 0.00e+000 9.96e-003  -6.7 8.32e-001    -  1.00e
+000 2.76e-002f  1
  19 1.0978977e+001 0.00e+000 8.25e-003  -6.4 9.25e-001    -  1.00e
+000 4.83e-002f  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du
 alpha_pr  ls
  20 1.0861417e+001 0.00e+000 1.05e-002  -4.9 9.85e-001    -
 9.98e-001 4.48e-002f  1
  21 1.0512857e+001 0.00e+000 1.39e-002  -5.3 1.10e+000    -  1.00e
+000 1.36e-001f  1
  22 1.0232965e+001 0.00e+000 2.36e-002  -4.0 1.91e+000    -
 7.07e-001 1.27e-001f  1
  23 1.0021448e+001 0.00e+000 9.38e-003  -6.3 1.09e+000    -
 3.74e-001 1.07e-001f  1
  24 9.5832724e+000 0.00e+000 6.53e-003  -3.9 1.13e+000    -
 9.00e-001 2.86e-001f  1
  25 9.4425217e+000 0.00e+000 8.83e-003  -4.9 8.39e-001    -
 6.07e-001 1.33e-001f  1
```

```
  26 9.2721003e+000 0.00e+000 1.31e-002  -4.6 6.98e-001    -
 9.46e-001 2.44e-001f  1
  27 9.0298465e+000 0.00e+000 3.82e-003  -4.3 8.57e-001    -  1.00e
+000 4.26e-001f  1
  28 8.8806041e+000 0.00e+000 1.26e-002  -4.0 5.19e-001    -
 8.42e-001 3.52e-001f  1
  29 8.7654375e+000 0.00e+000 4.62e-003  -4.1 4.16e-001    -
 5.65e-001 3.60e-001f  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du
 alpha_pr  ls
  30 8.6062147e+000 0.00e+000 6.24e-003  -4.1 5.72e-001    -
 6.78e-001 3.74e-001f  1
  31 8.5131785e+000 0.00e+000 5.41e-003  -4.1 6.41e-001    -
 3.72e-001 1.70e-001f  1
  32 8.3595389e+000 0.00e+000 4.83e-003  -4.3 7.22e-001    -
 2.75e-001 2.77e-001f  1
  33 8.1685741e+000 0.00e+000 4.60e-003  -4.1 7.79e-001    -
 4.64e-001 5.53e-001f  1
  34 8.0508215e+000 0.00e+000 1.17e-002  -3.4 4.84e-001    -
 5.14e-001 1.00e+000f  1
  35 7.9345906e+000 0.00e+000 1.37e-003  -4.3 2.32e-001    -
 9.54e-001 1.00e+000f  1
  36 7.8499631e+000 0.00e+000 9.59e-004  -5.0 1.94e-001    -
 9.87e-001 1.00e+000f  1
  37 7.7721251e+000 0.00e+000 1.05e-003  -4.8 1.93e-001    -
 9.43e-001 8.12e-001f  1
  38 7.6778951e+000 0.00e+000 3.94e-003  -5.0 3.20e-001    -
 9.98e-001 5.09e-001f  1
  39 7.6329474e+000 0.00e+000 6.16e-003  -5.3 5.41e-001    -
 9.18e-001 1.30e-001f  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du
 alpha_pr  ls
  40 7.5158258e+000 0.00e+000 9.85e-003  -4.1 6.00e-001    -
 6.09e-001 3.23e-001f  1
  41 7.4610116e+000 0.00e+000 2.81e-003  -5.8 9.18e-001    -
 7.10e-001 1.24e-001f  1
  42 7.3606139e+000 0.00e+000 5.82e-003  -4.1 5.33e-001    -
 3.63e-001 3.14e-001f  1
  43 7.2971402e+000 0.00e+000 3.10e-003  -4.7 6.48e-001    -
 4.13e-001 1.84e-001f  1
  44 7.2251428e+000 0.00e+000 4.20e-003  -4.3 5.35e-001    -
 4.94e-001 2.57e-001f  1
  45 7.1573042e+000 0.00e+000 5.76e-003  -4.4 5.28e-001    -
 9.00e-001 3.03e-001f  1
  46 7.0772400e+000 0.00e+000 2.55e-003  -4.7 8.17e-001    -
 6.37e-001 3.30e-001f  1
  47 7.0552974e+000 0.00e+000 4.08e-003 -10.6 4.24e-001    -
 3.78e-001 1.37e-001f  1
  48 7.7538384e+000 0.00e+000 5.52e-003  -2.8 1.07e+001    -
 3.80e-002 2.50e-001f  1
  49 7.1955782e+000 0.00e+000 4.61e-003  -4.2 2.19e+000    -  1.00e
+000 5.36e-001f  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du
 alpha_pr  ls
```

```
  50 6.9497764e+000 0.00e+000 9.67e-003  -4.2 6.84e-001    -  1.00e
+000 7.00e-001f  1
  51 6.8964778e+000 0.00e+000 5.85e-003  -4.7 5.27e-001    -
 8.88e-001 3.32e-001f  1
  52 6.8436481e+000 0.00e+000 4.56e-003  -5.5 6.25e-001    -
 8.31e-001 3.23e-001f  1
  53 6.8060607e+000 0.00e+000 5.08e-003  -4.9 5.26e-001    -
 6.07e-001 2.58e-001f  1
  54 6.7560011e+000 0.00e+000 1.18e-002  -4.3 4.92e-001    -
 5.02e-001 6.39e-001f  1
  55 6.7222675e+000 0.00e+000 2.60e-003  -4.8 6.01e-001    -
 4.96e-001 2.32e-001f  1
  56 6.6746449e+000 0.00e+000 3.10e-003  -4.2 5.90e-001    -
 4.52e-001 2.82e-001f  1
  57 6.6225337e+000 0.00e+000 2.46e-003  -4.4 6.42e-001    -
 4.69e-001 2.76e-001f  1
  58 8.2500213e+000 0.00e+000 5.30e-003  -2.4 1.83e+001    -
 1.29e-002 1.58e-001f  1
  59 6.6454438e+000 0.00e+000 2.99e-003  -4.1 3.07e+000    -
 2.02e-001 8.59e-001f  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du
 alpha_pr  ls
  60 6.5771759e+000 0.00e+000 5.09e-003  -4.1 4.99e-001    -  1.00e
+000 4.43e-001f  1
  61 6.5336113e+000 0.00e+000 1.52e-002  -4.8 6.41e-001    -
 9.88e-001 3.49e-001f  1
  62 6.4775403e+000 0.00e+000 3.52e-003  -5.8 8.64e-001    -  1.00e
+000 3.72e-001f  1
  63 6.4449004e+000 0.00e+000 1.16e-003  -4.1 4.05e-001    -
 4.98e-001 5.00e-001f  2
  64 6.4079727e+000 0.00e+000 2.04e-003  -4.1 9.41e-002    -  1.00e
+000 1.00e+000f  1
  65 6.3207261e+000 0.00e+000 8.75e-004  -5.0 9.01e-001    -
 2.89e-001 4.27e-001f  1
  66 6.3019422e+000 0.00e+000 3.06e-003  -4.8 6.05e-001    -
 4.83e-001 1.34e-001f  1
  67 6.2640220e+000 0.00e+000 5.83e-003  -4.9 7.61e-001    -
 8.39e-001 2.39e-001f  1
  68 6.2259820e+000 0.00e+000 4.14e-003  -4.9 7.61e-001    -
 8.14e-001 2.43e-001f  1
  69 6.2080493e+000 0.00e+000 3.86e-003 -10.8 1.05e+000    -
 3.96e-001 8.82e-002f  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du
 alpha_pr  ls
  70 6.1657456e+000 0.00e+000 5.84e-003  -5.5 1.08e+000    -
 3.68e-001 2.04e-001f  1
  71 6.1370482e+000 0.00e+000 7.76e-003  -5.5 1.04e+000    -
 8.72e-001 1.58e-001f  1
  72 6.0859933e+000 0.00e+000 3.35e-003  -4.4 8.28e-001    -
 4.81e-001 3.65e-001f  1
  73 6.0582691e+000 0.00e+000 4.07e-003  -4.4 4.89e-001    -
 6.27e-001 2.95e-001f  1
  74 6.0340044e+000 0.00e+000 2.01e-003  -4.5 7.03e-001    -
 3.79e-001 1.81e-001f  1
```

```
  75 6.0067968e+000 0.00e+000 3.99e-003  -6.1 9.32e-001    -
 2.18e-001 1.66e-001f  1
  76 5.9761879e+000 0.00e+000 2.84e-003  -4.9 9.62e-001    -
 7.22e-001 1.94e-001f  1
  77 5.9514259e+000 0.00e+000 2.06e-003  -4.3 4.72e-001    -
 2.58e-001 2.51e-001f  1
  78 5.9301538e+000 0.00e+000 3.76e-003  -4.8 6.46e-001    -
 3.49e-001 1.92e-001f  1
  79 5.9019870e+000 0.00e+000 2.60e-003  -4.7 8.02e-001    -
 3.68e-001 2.34e-001f  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du
 alpha_pr  ls
  80 5.8874722e+000 0.00e+000 8.73e-003  -5.0 6.37e-001    -
 7.07e-001 1.50e-001f  1
  81 5.8363212e+000 0.00e+000 3.03e-003  -4.9 8.38e-001    -
 7.90e-001 5.02e-001f  1
  82 5.8327784e+000 0.00e+000 6.00e-003  -6.8 7.40e-001    -
 6.14e-001 3.30e-002f  1
  83 5.8173565e+000 0.00e+000 4.21e-003  -4.1 3.19e-001    -
 5.30e-001 1.00e+000f  1
  84 5.7847139e+000 0.00e+000 1.97e-003  -4.7 5.54e-001    -
 5.71e-001 4.47e-001f  1
  85 5.7499361e+000 0.00e+000 2.99e-003  -4.7 6.04e-001    -
 6.67e-001 5.06e-001f  1
  86 5.7336923e+000 0.00e+000 1.61e-003  -4.8 7.17e-001    -
 4.68e-001 1.59e-001f  1
  87 5.7253575e+000 0.00e+000 7.80e-003  -6.1 5.68e-001    -
 5.37e-001 1.19e-001f  1
  88 5.6948046e+000 0.00e+000 2.88e-003  -5.7 1.12e+000    -
 7.93e-001 2.37e-001f  1
  89 5.6855603e+000 0.00e+000 3.69e-003  -5.2 8.07e-001    -
 2.18e-001 8.10e-002f  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du
 alpha_pr  ls
  90 5.6929520e+000 0.00e+000 4.48e-003  -4.0 9.04e-001    -
 3.84e-001 1.00e+000f  1
  91 5.6654813e+000 0.00e+000 1.24e-003  -4.5 6.48e-001    -
 7.18e-001 2.63e-001f  1
  92 5.6386652e+000 0.00e+000 5.33e-003  -4.5 5.53e-001    -
 8.61e-001 3.76e-001f  1
  93 5.6056039e+000 0.00e+000 4.42e-003  -4.6 5.78e-001    -
 9.37e-001 5.01e-001f  1
  94 5.5906091e+000 0.00e+000 7.93e-003  -4.9 7.07e-001    -
 9.29e-001 2.00e-001f  1
  95 5.5715780e+000 0.00e+000 4.40e-003  -5.5 9.45e-001    -
 6.58e-001 1.92e-001f  1
  96 5.5526547e+000 0.00e+000 2.98e-003  -4.8 7.18e-001    -
 3.58e-001 2.36e-001f  1
  97 5.5319082e+000 0.00e+000 3.38e-003  -5.0 7.86e-001    -
 4.31e-001 2.42e-001f  1
  98 5.5109597e+000 0.00e+000 2.16e-003  -5.1 1.02e+000    -
 5.59e-001 2.08e-001f  1
  99 5.5074125e+000 0.00e+000 3.71e-003 -11.0 8.31e-001    -
 2.24e-001 3.91e-002f  1
```

```
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du
 alpha_pr  ls
 100 5.4795582e+000 0.00e+000 2.37e-003  -6.2 1.23e+000    -
 2.30e-001 2.17e-001f  1
 101 5.4730004e+000 0.00e+000 2.73e-003  -5.5 1.25e+000    -
 8.62e-001 4.46e-002f  1
 102 5.4536776e+000 0.00e+000 4.88e-003  -4.8 5.97e-001    -
 3.83e-001 2.65e-001f  1
 103 5.6283167e+000 0.00e+000 4.17e-003  -3.2 1.15e+001    -
 1.65e-002 1.12e-001f  1
 104 5.4248219e+000 0.00e+000 4.41e-003  -4.8 2.17e+000    -
 1.36e-001 7.07e-001f  1
 105 5.4049961e+000 0.00e+000 2.04e-003  -4.8 6.80e-001    -
 5.05e-001 1.43e-001f  1
 106 5.3995461e+000 0.00e+000 4.41e-003  -6.4 6.10e-001    -
 6.68e-001 8.96e-002f  1
 107 5.3804253e+000 0.00e+000 4.01e-003  -5.2 7.00e-001    -
 9.46e-001 2.51e-001f  1
 108 5.3657193e+000 0.00e+000 3.02e-003  -5.3 9.02e-001    -
 4.51e-001 1.35e-001f  1
 109 5.3558920e+000 0.00e+000 3.91e-003 -11.0 7.10e-001    -
 1.27e-001 1.21e-001f  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du
 alpha_pr  ls
 110 5.3331654e+000 0.00e+000 2.63e-003  -5.6 1.08e+000    -
 4.26e-001 1.91e-001f  1
 111 5.3164685e+000 0.00e+000 1.87e-003  -5.9 1.52e+000    -
 3.16e-001 9.83e-002f  1
 112 5.3138576e+000 0.00e+000 4.91e-003 -11.0 7.43e-001    -
 2.09e-001 3.23e-002f  1
 113 5.2866218e+000 0.00e+000 2.05e-003  -5.7 1.44e+000    -
 6.32e-001 1.87e-001f  1
 114 5.3187769e+000 0.00e+000 1.15e-003  -4.3 3.22e-001    -
 2.24e-001 1.00e+000f  1
 115 5.2958090e+000 0.00e+000 9.72e-004  -4.9 9.95e-001    -
 3.45e-001 1.91e-001f  1
 116 5.2529006e+000 0.00e+000 1.87e-003  -4.9 1.06e+000    -
 5.46e-001 3.35e-001f  1
 117 5.2305804e+000 0.00e+000 3.85e-003  -4.6 5.56e-001    -
 6.24e-001 3.25e-001f  1
 118 5.2241578e+000 0.00e+000 5.17e-003  -5.1 4.32e-001    -
 7.41e-001 1.41e-001f  1
 119 5.2041025e+000 0.00e+000 1.89e-003  -4.7 5.32e-001    -
 6.69e-001 4.60e-001f  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du
 alpha_pr  ls
 120 5.1971050e+000 0.00e+000 2.24e-003  -5.9 7.37e-001    -
 5.66e-001 9.48e-002f  1
 121 5.1931953e+000 0.00e+000 7.09e-003  -7.2 6.80e-001    -
 4.79e-001 5.76e-002f  1
 122 5.1662009e+000 0.00e+000 3.82e-003  -5.1 1.06e+000    -
 4.43e-001 2.64e-001f  1
 123 5.1855544e+000 0.00e+000 2.08e-003  -4.3 2.93e-001    -
 4.25e-001 1.00e+000f  1
```

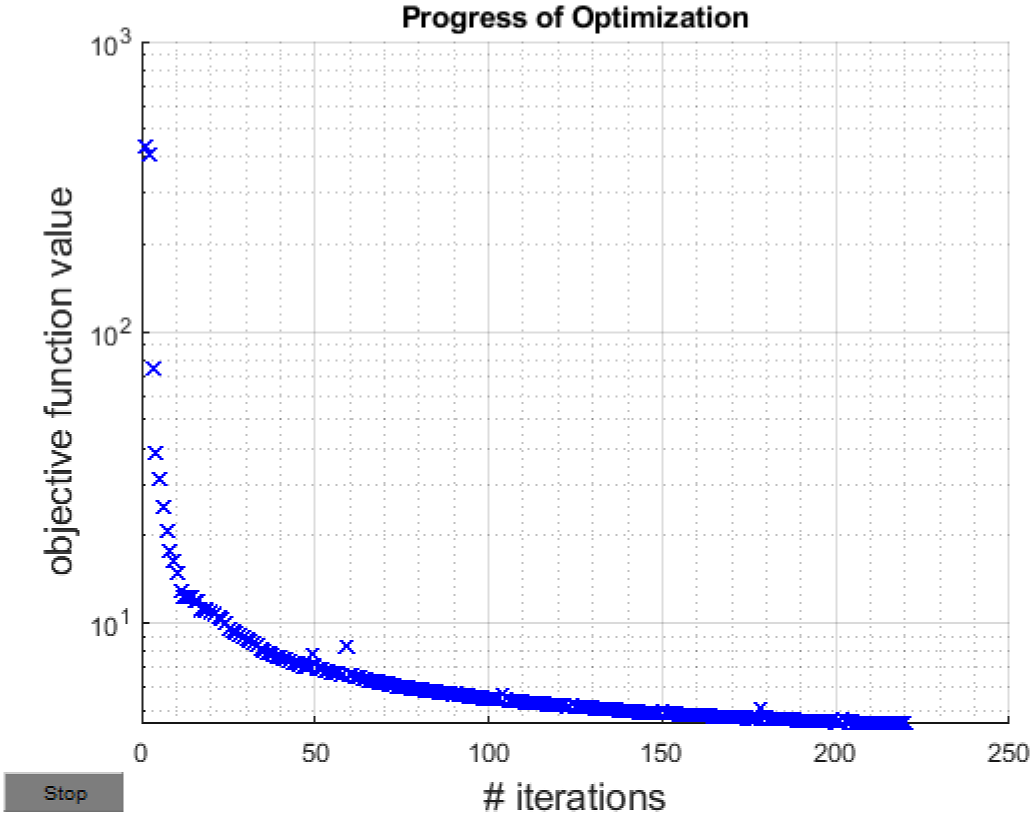
```
 124 5.1665936e+000 0.00e+000 1.79e-003  -4.7 5.79e-001    -
 4.57e-001 2.92e-001f  1
 125 5.1404920e+000 0.00e+000 2.37e-003  -4.7 5.12e-001    -
 5.74e-001 4.32e-001f  1
 126 5.1333176e+000 0.00e+000 2.13e-003  -4.8 3.93e-001    -
 4.80e-001 1.55e-001f  1
 127 5.1165762e+000 0.00e+000 8.45e-003  -5.1 5.01e-001    -
 7.10e-001 3.17e-001f  1
 128 5.1096427e+000 0.00e+000 7.93e-003  -5.7 5.42e-001    -
 5.92e-001 1.31e-001f  1
 129 5.0957870e+000 0.00e+000 5.99e-003  -5.9 7.84e-001    -
 8.70e-001 1.94e-001f  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du
 alpha_pr  ls
 130 5.0774987e+000 0.00e+000 3.86e-003  -6.0 8.29e-001    -
 8.13e-001 2.64e-001f  1
 131 5.0675379e+000 0.00e+000 3.36e-003  -5.3 6.30e-001    -
 7.34e-001 1.87e-001f  1
 132 5.0563330e+000 0.00e+000 5.22e-003  -4.7 2.73e-001    -
 7.01e-001 5.58e-001f  1
 133 5.0460255e+000 0.00e+000 2.46e-003  -4.8 3.68e-001    -
 3.66e-001 3.04e-001f  1
 134 5.0359030e+000 0.00e+000 1.20e-003  -4.9 5.46e-001    -
 3.07e-001 2.03e-001f  1
 135 5.0255534e+000 0.00e+000 1.16e-003  -4.9 8.91e-001    -
 6.18e-001 1.51e-001f  1
 136 5.0175313e+000 0.00e+000 4.48e-003  -5.2 9.16e-001    -
 3.43e-001 1.20e-001f  1
 137 4.9935769e+000 0.00e+000 3.12e-003  -5.2 1.25e+000    -
 8.81e-001 2.90e-001f  1
 138 4.9855568e+000 0.00e+000 2.33e-003  -5.9 1.09e+000    -
 1.68e-001 1.12e-001f  1
 139 4.9694039e+000 0.00e+000 2.31e-003  -5.5 1.17e+000    -
 5.14e-001 2.23e-001f  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du
 alpha_pr  ls
 140 4.9603623e+000 0.00e+000 1.77e-003 -11.0 1.46e+000    -
 3.97e-001 1.03e-001f  1
 141 4.9446081e+000 0.00e+000 1.04e-003  -4.6 2.53e-001    -
 3.41e-001 1.00e+000f  1
 142 4.9380850e+000 0.00e+000 1.71e-003  -5.5 9.58e-001    -
 4.63e-001 1.16e-001f  1
 143 4.9216076e+000 0.00e+000 3.37e-003  -4.9 8.19e-001    -
 6.27e-001 3.82e-001f  1
 144 4.9149058e+000 0.00e+000 5.37e-003  -5.2 5.81e-001    -
 4.35e-001 1.80e-001f  1
 145 4.9075415e+000 0.00e+000 2.79e-003  -5.4 8.23e-001    -
 6.80e-001 1.41e-001f  1
 146 4.8916279e+000 0.00e+000 3.41e-003  -5.2 8.31e-001    -
 3.70e-001 3.08e-001f  1
 147 4.8882254e+000 0.00e+000 3.68e-003 -11.0 7.99e-001    -
 4.47e-001 7.04e-002f  1
 148 4.8765019e+000 0.00e+000 2.35e-003  -7.3 9.92e-001    -
 5.03e-001 2.06e-001f  1
```

```
 149 5.0009188e+000 0.00e+000 2.04e-003  -3.9 4.08e+000    -
 2.11e-002 4.74e-001f  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du
 alpha_pr  ls
 150 4.8935944e+000 0.00e+000 1.98e-003  -5.4 2.65e+000    -
 4.78e-002 4.95e-001f  1
 151 4.8912896e+000 0.00e+000 3.23e-003  -5.4 1.26e+000    -  1.00e
+000 2.40e-002f  1
 152 4.8626442e+000 0.00e+000 7.21e-003  -5.4 1.10e+000    -
 6.31e-001 3.85e-001f  1
 153 4.8549814e+000 0.00e+000 7.76e-003  -5.6 7.11e-001    -
 9.97e-001 1.78e-001f  1
 154 4.8446526e+000 0.00e+000 2.93e-003  -5.2 4.96e-001    -
 7.15e-001 3.39e-001f  1
 155 4.8410230e+000 0.00e+000 6.75e-003  -5.9 4.81e-001    -
 7.41e-001 1.30e-001f  1
 156 4.8346397e+000 0.00e+000 5.65e-003  -6.9 6.05e-001    -
 4.35e-001 1.86e-001f  1
 157 4.8267163e+000 0.00e+000 5.43e-003  -6.3 8.08e-001    -
 7.47e-001 1.74e-001f  1
 158 4.8190755e+000 0.00e+000 3.28e-003  -6.2 8.35e-001    -
 4.63e-001 1.63e-001f  1
 159 4.8132600e+000 0.00e+000 2.30e-003  -5.3 7.23e-001    -
 2.53e-001 1.31e-001f  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du
 alpha_pr  ls
 160 4.8049667e+000 0.00e+000 9.16e-004  -4.8 1.95e-001    -
 3.18e-001 1.00e+000f  1
 161 4.7994937e+000 0.00e+000 8.13e-004  -4.9 5.44e-001    -
 4.20e-001 2.36e-001f  1
 162 4.7857813e+000 0.00e+000 7.53e-004  -5.0 1.11e+000    -
 3.95e-001 2.88e-001f  1
 163 4.7799209e+000 0.00e+000 2.31e-003  -5.3 1.08e+000    -
 5.00e-001 1.41e-001f  1
 164 4.7695435e+000 0.00e+000 1.52e-003  -5.7 1.22e+000    -
 5.59e-001 2.02e-001f  1
 165 4.7635091e+000 0.00e+000 2.62e-003  -6.6 1.40e+000    -
 5.40e-001 1.02e-001f  1
 166 4.7501624e+000 0.00e+000 1.66e-003  -4.7 6.98e-001    -
 5.89e-001 5.39e-001f  1
 167 4.7438780e+000 0.00e+000 1.53e-003  -4.9 6.37e-001    -
 4.62e-001 2.49e-001f  1
 168 4.7388600e+000 0.00e+000 2.27e-003  -4.9 6.10e-001    -
 3.39e-001 2.44e-001f  1
 169 4.7323564e+000 0.00e+000 1.30e-003  -4.9 6.79e-001    -
 4.39e-001 2.88e-001f  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du
 alpha_pr  ls
 170 4.7305070e+000 0.00e+000 1.82e-003  -5.4 9.51e-001    -
 4.30e-001 5.42e-002f  1
 171 4.7184328e+000 0.00e+000 6.89e-003  -6.1 1.12e+000    -
 4.87e-001 3.12e-001f  1
 172 4.7071066e+000 0.00e+000 5.39e-003  -6.6 1.55e+000    -
 1.99e-001 2.02e-001f  1
```

```
 173 4.7058131e+000 0.00e+000 6.97e-003  -5.4 8.04e-001    -
 6.30e-001 4.19e-002f  1
 174 4.6935621e+000 0.00e+000 2.17e-003  -4.9 1.14e+000    -
 6.91e-001 2.94e-001f  1
 175 4.6875891e+000 0.00e+000 2.53e-003  -4.8 4.09e-001    -
 3.98e-001 3.16e-001f  1
 176 4.6799790e+000 0.00e+000 1.60e-003  -5.0 6.00e-001    -
 2.59e-001 2.71e-001f  1
 177 5.0853764e+000 0.00e+000 4.14e-003  -2.9 2.67e+001    -
 3.98e-003 1.17e-001f  1
 178 4.7143241e+000 0.00e+000 1.06e-003  -4.8 3.24e+000    -
 5.77e-003 8.09e-001f  1
 179 4.7145642e+000 0.00e+000 2.08e-002  -4.8 1.17e+001    -
 3.48e-002 5.28e-003f  7
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du
 alpha_pr  ls
 180 4.6959088e+000 0.00e+000 7.71e-004  -4.8 7.61e-001    -
 9.57e-001 3.39e-001f  1
 181 4.6727963e+000 0.00e+000 6.17e-003  -4.8 7.67e-001    -  1.00e
+000 8.67e-001f  1
 182 4.6695099e+000 0.00e+000 5.98e-003  -5.3 3.91e-001    -
 8.14e-001 2.00e-001f  1
 183 4.6651104e+000 0.00e+000 3.99e-003  -5.7 5.42e-001    -
 8.28e-001 1.81e-001f  1
 184 4.6558348e+000 0.00e+000 2.99e-003  -5.9 7.23e-001    -
 9.55e-001 2.98e-001f  1
 185 4.6470223e+000 0.00e+000 1.83e-003  -6.1 8.62e-001    -
 8.02e-001 2.81e-001f  1
 186 4.6413442e+000 0.00e+000 1.90e-003  -5.3 3.96e-001    -
 6.06e-001 4.20e-001f  1
 187 4.6314216e+000 0.00e+000 2.51e-003  -5.1 3.95e-001    -
 5.90e-001 1.00e+000f  1
 188 4.6261166e+000 0.00e+000 3.88e-004  -5.2 7.67e-001    -
 3.86e-001 2.84e-001f  1
 189 4.6188570e+000 0.00e+000 2.72e-003  -5.4 1.35e+000    -
 4.53e-001 2.37e-001f  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du
 alpha_pr  ls
 190 4.6128095e+000 0.00e+000 1.20e-003  -5.6 1.99e+000    -
 7.03e-001 1.39e-001f  1
 191 4.6062801e+000 0.00e+000 2.69e-003  -5.5 1.08e+000    -
 1.81e-001 2.96e-001f  1
 192 4.6005286e+000 0.00e+000 1.40e-003  -5.5 1.83e+000    -
 9.63e-001 1.45e-001f  1
 193 4.5924764e+000 0.00e+000 1.79e-003  -5.5 2.13e+000    -
 2.63e-001 1.70e-001f  1
 194 4.5883923e+000 0.00e+000 2.40e-003  -5.5 2.18e+000    -
 4.34e-001 8.11e-002f  1
 195 4.5765015e+000 0.00e+000 1.80e-003  -5.5 2.48e+000    -
 1.98e-001 2.04e-001f  1
 196 4.5732161e+000 0.00e+000 1.47e-003  -6.1 2.46e+000    -
 2.44e-001 5.55e-002f  1
 197 4.5682429e+000 0.00e+000 1.93e-003  -6.3 2.56e+000    -
 4.26e-001 7.89e-002f  1
```

```
 198 4.5604844e+000 0.00e+000 2.70e-003  -5.9 2.49e+000    -
 3.70e-001 1.27e-001f  1
 199 4.5513251e+000 0.00e+000 1.90e-003  -5.7 2.56e+000    -
 1.51e-001 1.38e-001f  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du
 alpha_pr  ls
 200 4.5482728e+000 0.00e+000 4.44e-003  -5.2 1.19e+000    -
 5.08e-001 1.02e-001f  1
 201 4.6292471e+000 0.00e+000 4.05e-003  -3.4 6.24e+001    -
 1.07e-002 3.64e-002f  1
 202 4.6221226e+000 0.00e+000 3.90e-003  -5.2 3.15e+000    -
 1.66e-002 5.25e-002f  1
 203 4.5706463e+000 0.00e+000 5.69e-003  -5.2 2.89e+000    -
 4.51e-001 4.42e-001f  1
 204 4.5582694e+000 0.00e+000 2.64e-003  -5.2 1.84e+000    -
 4.28e-001 2.09e-001f  1
 205 4.5441603e+000 0.00e+000 3.61e-003  -5.2 1.45e+000    -
 7.52e-001 3.35e-001f  1
 206 4.5386418e+000 0.00e+000 3.57e-003  -5.1 7.98e-001    -
 7.23e-001 2.44e-001f  1
 207 4.5391802e+000 0.00e+000 8.09e-004  -4.9 5.12e-001    -
 5.09e-001 1.00e+000f  1
 208 4.5369142e+000 0.00e+000 1.84e-003  -5.1 3.96e-001    -  1.00e
+000 2.49e-001f  1
 209 4.5287469e+000 0.00e+000 6.69e-003  -5.7 1.09e+000    -
 9.01e-001 3.11e-001f  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du
 alpha_pr  ls
 210 4.5224960e+000 0.00e+000 1.92e-003  -5.3 9.36e-001    -
 6.42e-001 2.85e-001f  1
 211 4.5204834e+000 0.00e+000 1.27e-003  -5.1 4.74e-001    -
 4.18e-001 1.91e-001f  1
 212 4.5102765e+000 0.00e+000 1.81e-003  -5.3 1.23e+000    -
 3.55e-001 3.47e-001f  1
 213 4.5059642e+000 0.00e+000 9.76e-004  -5.3 1.02e+000    -
 3.67e-001 1.77e-001f  1
 214 4.5051061e+000 0.00e+000 2.93e-003  -6.8 1.32e+000    -
 2.74e-001 2.48e-002f  1
 215 4.4981644e+000 0.00e+000 1.54e-003  -6.5 2.22e+000    -
 3.03e-001 1.17e-001f  1
 216 4.4925760e+000 0.00e+000 1.84e-003  -5.3 1.13e+000    -
 2.23e-001 1.74e-001f  1
 217 4.4903224e+000 0.00e+000 1.83e-003  -6.4 1.55e+000    -
 1.83e-001 5.10e-002f  1
 218 4.4871978e+000 0.00e+000 2.01e-003  -5.3 1.21e+000    -
 3.11e-001 8.82e-002f  1
 219 4.4846916e+000 0.00e+000 1.12e-003  -4.9 1.46e+000    -
 2.15e-001 4.38e-001f  1

Number of Iterations....: 219

                                   (scaled)                 (unscaled)
Objective...............:  4.4846916045641736e+000
 4.4846916045641736e+000
```

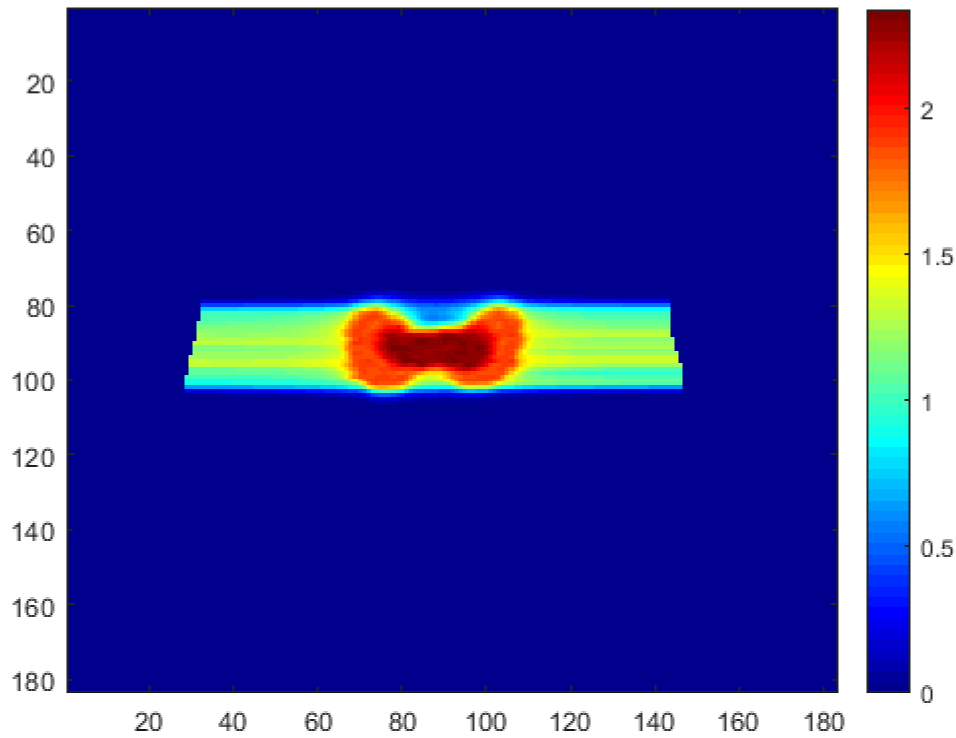
```
Dual infeasibility......:  1.1246414861891072e-003
 1.1246414861891072e-003
Constraint violation....:  0.0000000000000000e+000
 0.0000000000000000e+000
Complementarity.........:  4.0607728791104100e-004
 4.0607728791104100e-004
Overall NLP error.......:  1.1246414861891072e-003
 1.1246414861891072e-003


Number of objective function evaluations             = 235
Number of objective gradient evaluations             = 220
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)   =     69.821
Total CPU secs in NLP function evaluations           =    185.803

EXIT: Solved To Acceptable Level.
matRad: applying a constant RBE of 1.1
```



Plot the Resulting Dose Slice

Let's plot the transversal iso-center dose slice

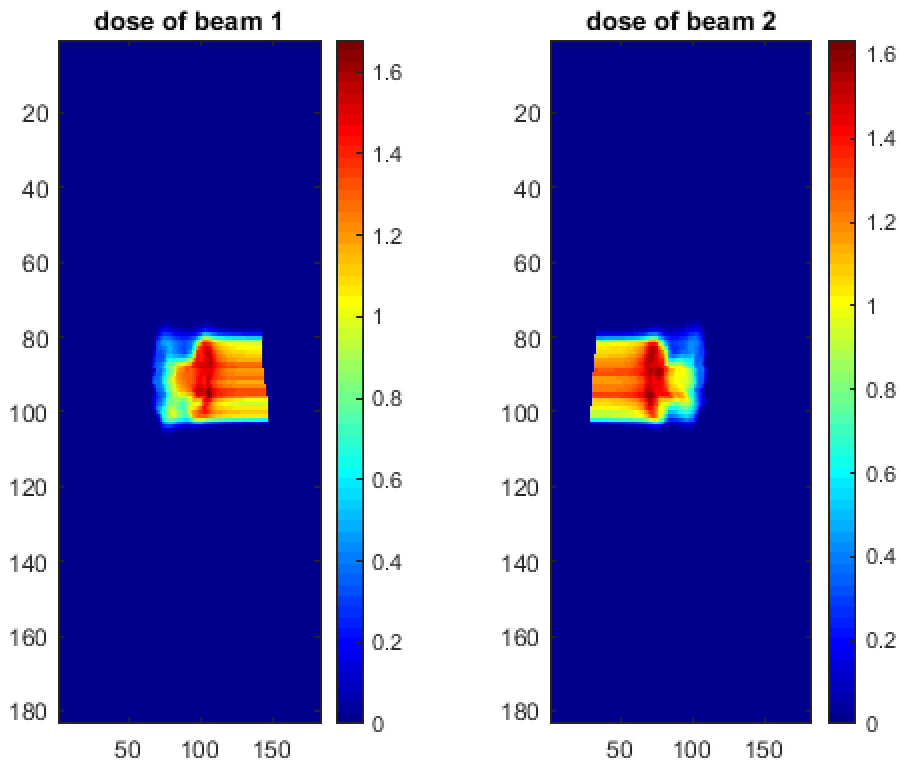
```
slice = round(pln.propStf.isoCenter(1,3)./ct.resolution.z);
figure
imagesc(resultGUI.RBExDose(:,:,slice)),colorbar,colormap(jet)
```



Plot the Resulting Beam Dose Slice

Let's plot the transversal iso-center dose slice of beam 1 and beam 2 separately

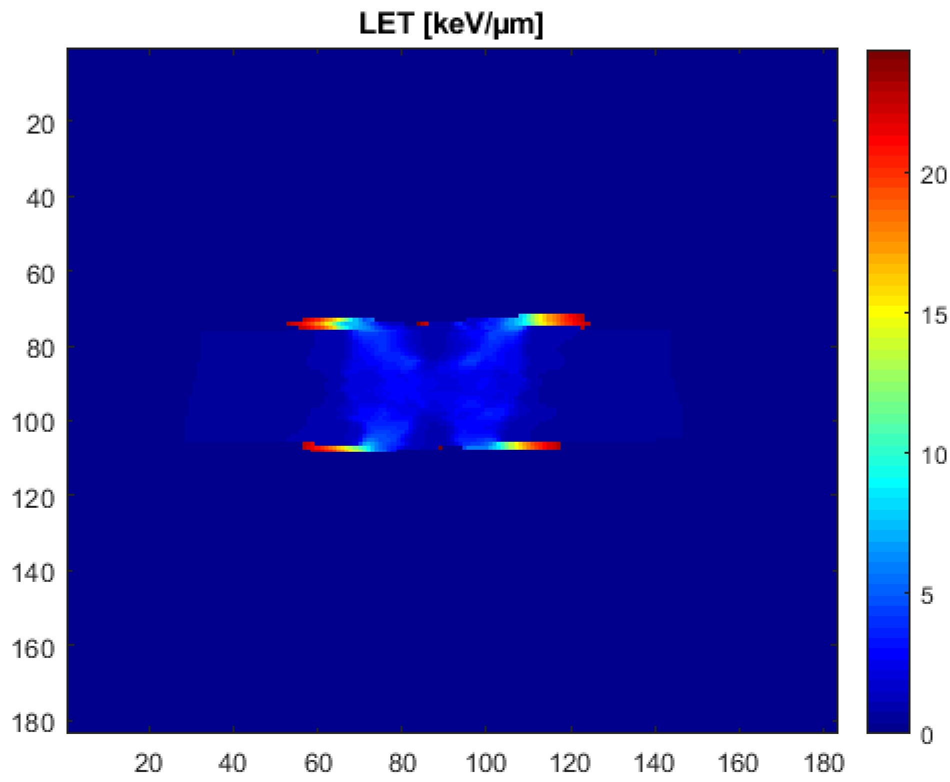
```
figure
subplot(121),imagesc(resultGUI.RBExDose_beam1(:,:,slice)),colorbar,colormap(jet),t
 of beam 1')
subplot(122),imagesc(resultGUI.RBExDose_beam2(:,:,slice)),colorbar,colormap(jet),t
 of beam 2')
```



and the corresponding LET distribution

Transversal iso-center slice

```
figure
imagesc(resultGUI.LET(:,:,slice)),colormap(jet),colorbar,title('LET
 [keV/µm]')
```



Now let's simulate a patient shift in y direction for both beams

```
stf(1).isoCenter(2) = stf(1).isoCenter(2) - 4;
stf(2).isoCenter(2) = stf(2).isoCenter(2) - 4;
pln.propStf.isoCenter       = reshape([stf.isoCenter],[3
 pln.propStf.numOfBeams])';
```

Recalculate Plan

Let's use the existing optimized pencil beam weights and recalculate the RBE weighted dose

```
resultGUI_isoShift =
 matRad_calcDoseDirect(ct,stf,pln,cst,resultGUI.w);
```

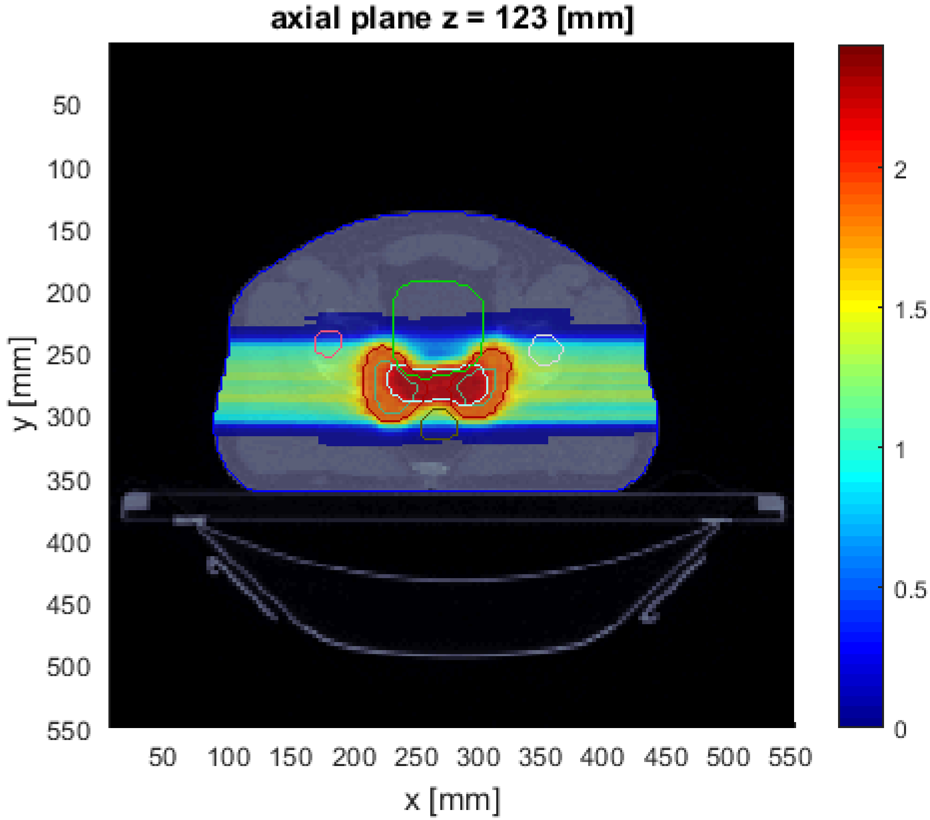
```
Warning: Could not find HLUT
Philips-AcQSimCT-ConvolutionKernel-000000_protons.hlut in hlutLibrary
 folder.
matRad default HLUT loaded
matRad: Using a constant RBE of 1.1
matRad: Particle dose calculation...
Beam 1 of 2:
matRad: calculate radiological depth cube...done.
matRad: calculate lateral cutoff...done.
Progress: 100.00 %
Beam 2 of 2:
matRad: calculate radiological depth cube...done.
```

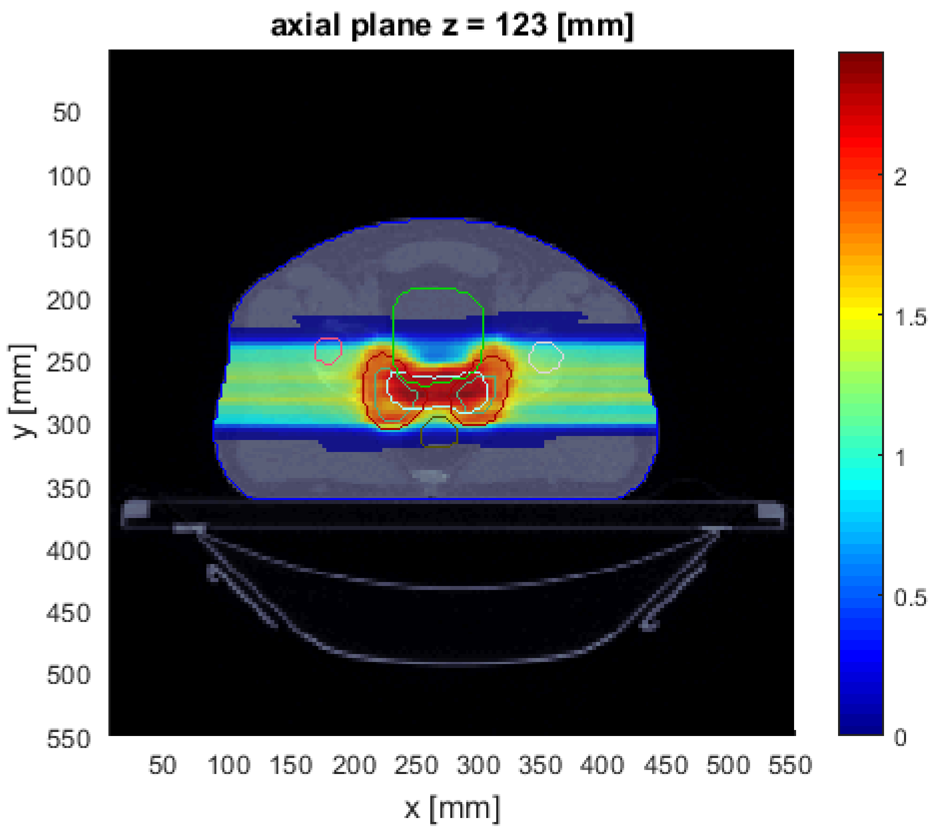
```
matRad: calculate lateral cutoff...done.
Progress: 100.00 %
matRad: applying a constant RBE of 1.1
```

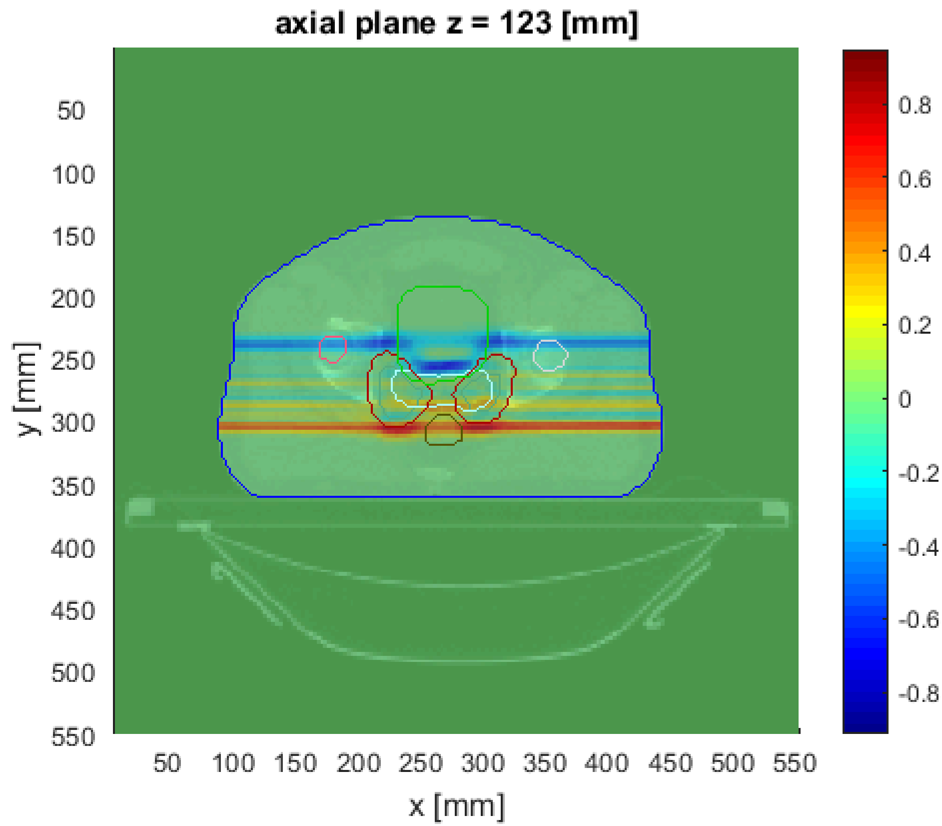
Visual Comparison of results

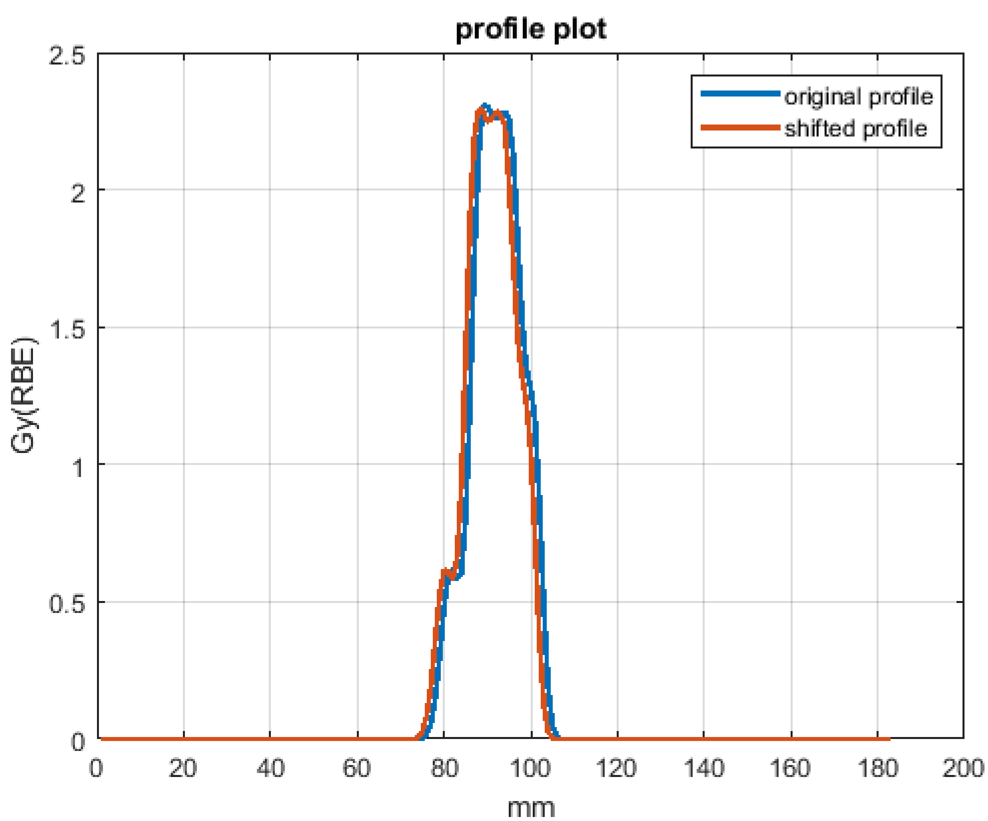
Let's compare the new recalculation against the optimization result.

```
plane = 3;
doseWindow = [0 max([resultGUI.RBExDose(:);
 resultGUI_isoShift.RBExDose(:)])];

figure,title('original plan')
matRad_plotSliceWrapper(gca,ct,cst,1,resultGUI.RBExDose,plane,slice,
[],0.75,colorcube,[],doseWindow,[]);
figure,title('shifted plan')
matRad_plotSliceWrapper(gca,ct,cst,1,resultGUI_isoShift.RBExDose,plane,slice,
[],0.75,colorcube,[],doseWindow,[]);

absDiffCube = resultGUI.RBExDose-resultGUI_isoShift.RBExDose;
figure,title('absolute difference')
matRad_plotSliceWrapper(gca,ct,cst,1,absDiffCube,plane,slice,[],
[],colorcube);

% Let's plot single profiles that are perpendicular to the beam
 direction
ixProfileY = round(pln.propStf.isoCenter(1,2)./ct.resolution.y);

profileOrginal = resultGUI.RBExDose(:,ixProfileY,slice);
profileShifted = resultGUI_isoShift.RBExDose(:,ixProfileY,slice);

figure,plot(profileOrginal,'LineWidth',2),grid on,hold on,
       plot(profileShifted,'LineWidth',2),legend({'original
 profile','shifted profile'}),
       xlabel('mm'),ylabel('Gy(RBE)'),title('profile plot')
```


axial plane z = 123 [mm]
y [mm]
50
100
150
200
250
300
350
400
450
500
550
x [mm]
2
1.5
1
0.5
0


axial plane z = 123 [mm]
y [mm]
50
100
150
200
250
300
350
400
450
500
550
x [mm]
2
1.5
1
0.5
0

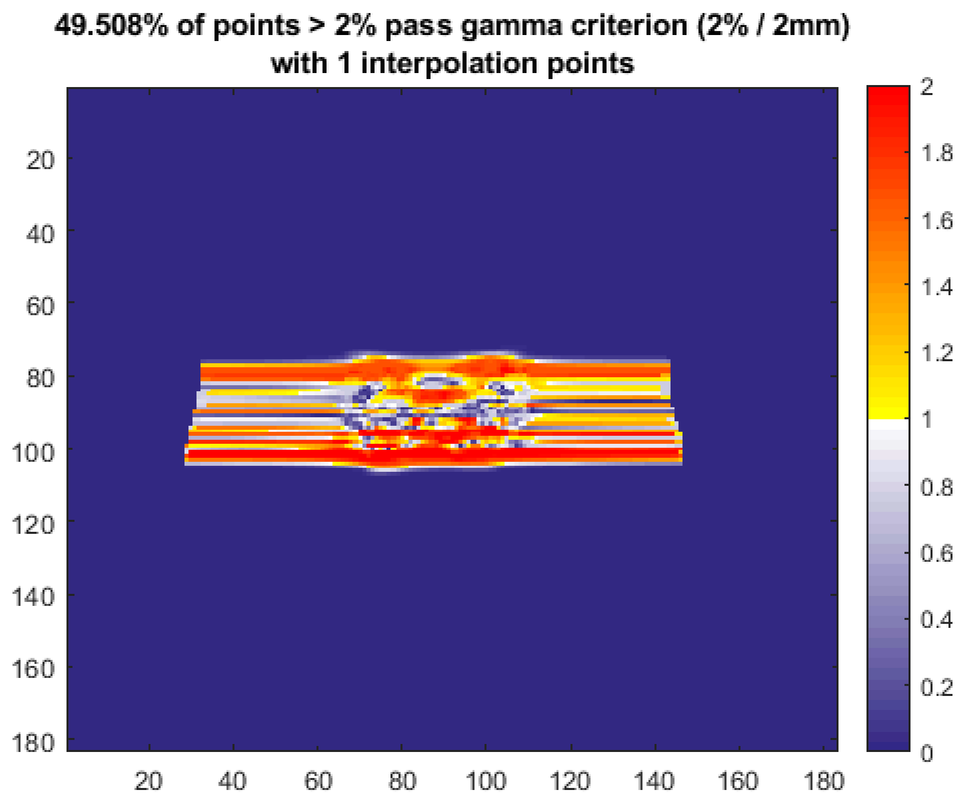

axial plane z = 123 [mm]
y [mm]
x [mm]
50
100
150
200
250
300
350
400
450
500
550
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8

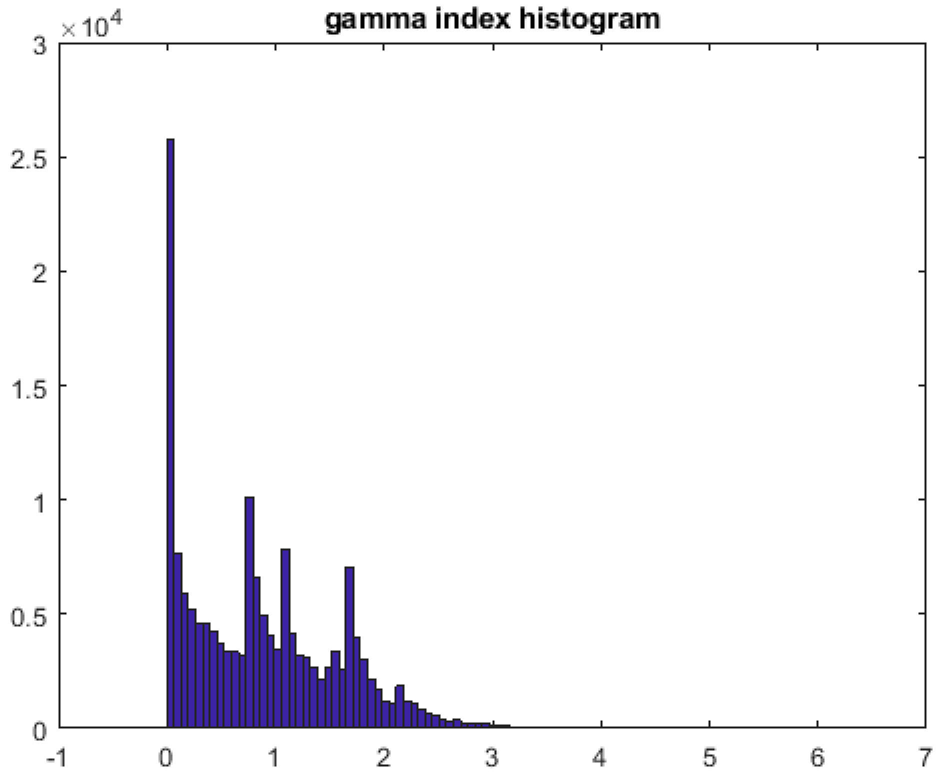

profile plot
original profile
shifted profile
Gy(RBE)
mm
2.5
2
1.5
1
0.5
0
0
20
40
60
80
100
120
140
160
180
200

Quantitative Comparison of results

Compare the two dose cubes using a gamma-index analysis. The gamma index is a composite quality distribution equally taking into account a dose difference and a distance to agreement criterion in order to quantify differences between two dose cubes. A gamma-index value of smaller than 1 indicates a successful test and a value greater than 1 illustrates a failed test.

```
doseDifference  = 2;
distToAgreement = 2;
n               = 1;

[gammaCube,gammaPassRateCell] = matRad_gammaIndex(...
    resultGUI_isoShift.RBExDose,resultGUI.RBExDose,...
    [ct.resolution.x, ct.resolution.y, ct.resolution.z],...
    [doseDifference distToAgreement],slice,n,'global',cst);


[env, ~] = matRad_getEnvironment();
% Let's plot the gamma index histogram
figure
hist(gammaCube(gammaCube>0),100)
title('gamma index histogram')
```





Published with MATLAB® R2018a