Example: Carbon Ion Treatment Plan

Table of Contents

In this example we will show (i) how to load patient data into matRad (ii) how to setup a carbon ion dose calculation plan including variable RBE optimization (iii) how to inversely optimize the pencil beam intensities based on the RBE-weighted dose (iv) how to inversely optimize the pencil beam intensities based on the biological effect (v) how to change the tissues' radiobiological characteristics (vi) how to recalculated the dose considering the previously optimized pencil beam intensities (vii) how to compare the two results

Patient Data Import

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

```
clc,clear,close all;
load('LIVER.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 carbon ions 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 'carbon_Generic.mat'; consequently the machine has to be set accordingly

```
pln.radiationMode = 'carbon';
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 carbon ions, we decide to use base data from the local effect model IV and want to optimize the RBE-weighted dose. Therefore we set bioOptimization to LEMIV_RBExD

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

The remaining plan parameters are set like in the previous example files

```
pln.numOfFractions        = 30;
pln.propStf.gantryAngles  = 315;
pln.propStf.couchAngles   = 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;
```

Generate Beam Geometry STF

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

```
matRad: Generating stf struct... Warning: Could not find HLUT GE
 MEDICAL SYSTEMS-Discovery CT590
RT-ConvolutionKernel-STANDARD_carbon.hlut in hlutLibrary folder.
 matRad default
HLUT loaded
Progress: 100.00 %
```

Let's have a closer look on the stf.ray sub-structure which contains the actual beam/ray geometry information. For illustration purposes we want to show ray # 100. Besides geometrical information about the position and orientation of the ray, we can also find pencil beam information. If the ray coincides with the target, pencil beams were defined along the ray from target entry to target exit.

```
display(stf.ray(100));
```

```
         rayPos_bev: [-21 0 27]
    targetPoint_bev: [-42 10000 54]
             rayPos: [-14.8492 14.8492 27]
        targetPoint: [7.0414e+03 7.1008e+03 54]
             energy: [1×8 double]
            focusIx: [1 1 1 1 1 1 1 1]
       rangeShifter: [1×8 struct]
```

Here are the energies selected on ray # 100:

```
display(stf.ray(100).energy);
```

```
  Columns 1 through 7

  157.1200   160.2600   163.3500   166.4100   169.4300   172.4100   175.3700
```

```
  Column 8

  178.2800
```

Dose Calculation

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

```
Warning: Could not find HLUT GE MEDICAL SYSTEMS-Discovery CT590
RT-ConvolutionKernel-STANDARD_carbon.hlut in hlutLibrary folder.
 matRad default
HLUT loaded
matRad: loading biological base data... done.
matRad: Particle dose calculation...
Beam 1 of 1:
matRad: calculate radiological depth cube...done.
matRad: calculate lateral cutoff...done.
Progress: 100.00 %
```

Inverse Optimization for IMPT based on RBE-weighted dose

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............................:    11746
                     variables with only lower bounds:    11746
                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 4.3212467e+002 0.00e+000 9.11e+000   0.0 0.00e+000    -  0.00e
+000 0.00e+000   0
   1 7.9203913e+003 0.00e+000 9.42e+000  -0.4 5.54e-001    -
 1.16e-001 5.00e-001f  2
   2 1.5932000e+003 0.00e+000 5.67e+000  -0.9 1.46e-001    -  1.00e
+000 1.00e+000f  1
   3 8.1545425e+001 0.00e+000 3.26e+000  -0.9 2.02e-001    -  1.00e
+000 6.85e-001f  1
   4 9.3140111e+001 0.00e+000 1.29e+000  -1.3 3.73e-002    -  1.00e
+000 1.00e+000f  1
   5 5.9937370e+001 0.00e+000 5.25e-001  -1.5 1.07e-002    -  1.00e
+000 1.00e+000f  1
   6 1.4922628e+001 0.00e+000 3.88e-001  -2.6 2.47e-002    -  1.00e
+000 1.00e+000f  1
   7 1.0529189e+001 0.00e+000 3.51e-001  -3.2 9.84e-003    -  1.00e
+000 1.00e+000f  1
   8 7.3029975e+000 0.00e+000 2.82e-001  -4.0 1.30e-002    -  1.00e
+000 1.00e+000f  1
   9 5.1175861e+000 0.00e+000 2.10e-001  -3.5 1.78e-002    -  1.00e
+000 1.00e+000f  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du
 alpha_pr  ls
  10 4.0143011e+000 0.00e+000 2.80e-001  -3.6 2.38e-002    -  1.00e
+000 9.32e-001f  1
  11 3.4530331e+000 0.00e+000 2.63e-001  -4.0 1.71e-002    -  1.00e
+000 6.08e-001f  1
  12 5.8197111e+000 0.00e+000 6.74e-001  -2.4 3.72e-002    -
 7.42e-001 1.00e+000f  1
  13 3.9196656e+000 0.00e+000 1.34e-001  -2.6 1.90e-002    -  1.00e
+000 1.00e+000f  1
  14 3.3376736e+000 0.00e+000 1.15e-001  -2.6 1.27e-002    -  1.00e
+000 1.00e+000f  1
  15 2.3218205e+000 0.00e+000 1.15e-001  -3.3 2.68e-002    -  1.00e
+000 1.00e+000f  1
  16 2.0802587e+000 0.00e+000 8.57e-002  -4.2 1.56e-002    -  1.00e
+000 1.00e+000f  1
  17 2.0056115e+000 0.00e+000 1.98e-001  -4.8 1.68e-002    -  1.00e
+000 3.60e-001f  1
  18 1.8672808e+000 0.00e+000 2.04e-001  -5.3 3.66e-002    -  1.00e
+000 3.03e-001f  1
  19 1.6511289e+000 0.00e+000 1.48e-001  -3.5 2.24e-002    -
 8.03e-001 8.70e-001f  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du
 alpha_pr  ls
  20 1.5413994e+000 0.00e+000 1.28e-001  -3.8 7.43e-003    -  1.00e
+000 9.31e-001f  1
  21 1.4943856e+000 0.00e+000 3.08e-001  -4.8 1.04e-002    -  1.00e
+000 5.48e-001f  1
```

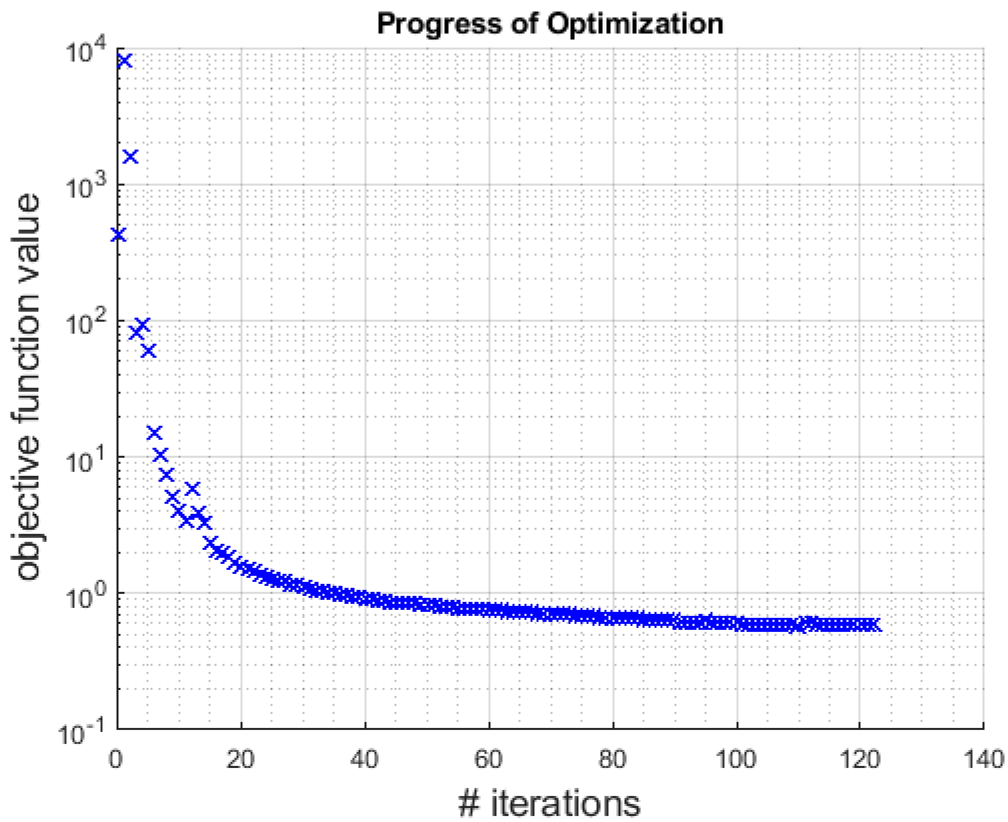
```
 22 1.4507023e+000 0.00e+000 2.42e-001  -5.8 2.12e-002    -
 8.63e-001 2.61e-001f  1
 23 1.3811969e+000 0.00e+000 1.77e-001  -4.2 2.74e-002    -
 7.22e-001 3.37e-001f  1
 24 1.3229031e+000 0.00e+000 2.64e-001  -3.6 2.76e-002    -
 4.00e-001 9.42e-001f  1
 25 1.2805318e+000 0.00e+000 3.08e-001  -4.6 1.90e-002    -
 9.36e-001 3.12e-001f  1
 26 1.2515544e+000 0.00e+000 2.56e-001  -4.4 1.24e-002    -
 8.09e-001 3.70e-001f  1
 27 1.2234903e+000 0.00e+000 1.79e-001  -4.8 1.98e-002    -
 7.49e-001 2.33e-001f  1
 28 1.1634019e+000 0.00e+000 1.34e-001  -4.7 2.75e-002    -
 9.40e-001 3.77e-001f  1
 29 1.1415768e+000 0.00e+000 3.71e-001  -3.9 3.88e-002    -
 4.78e-001 6.43e-001f  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du
 alpha_pr  ls
 30 1.1220968e+000 0.00e+000 2.99e-001  -4.2 1.02e-002    -
 7.56e-001 2.71e-001f  1
 31 1.0839633e+000 0.00e+000 1.55e-001  -4.4 1.50e-002    -
 7.43e-001 4.13e-001f  1
 32 1.0609757e+000 0.00e+000 1.27e-001  -5.0 1.77e-002    -
 8.96e-001 2.38e-001f  1
 33 1.0379480e+000 0.00e+000 2.83e-001  -5.6 2.14e-002    -
 9.77e-001 2.10e-001f  1
 34 1.0078392e+000 0.00e+000 2.42e-001  -6.0 2.34e-002    -
 6.96e-001 2.88e-001f  1
 35 9.9621414e-001 0.00e+000 3.01e-001  -5.0 1.67e-002    -
 9.06e-001 1.50e-001f  1
 36 9.7632027e-001 0.00e+000 1.89e-001  -4.4 1.43e-002    -
 5.72e-001 2.92e-001f  1
 37 9.7460739e-001 0.00e+000 8.09e-002  -3.9 1.19e-002    -
 5.45e-001 1.00e+000f  1
 38 9.5899063e-001 0.00e+000 1.01e-001  -4.4 1.06e-002    -
 8.34e-001 2.77e-001f  1
 39 9.3896502e-001 0.00e+000 1.80e-001  -4.6 1.32e-002    -  1.00e
+000 2.93e-001f  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du
 alpha_pr  ls
 40 9.2275621e-001 0.00e+000 2.27e-001  -5.2 1.45e-002    -
 8.88e-001 2.36e-001f  1
 41 9.0825277e-001 0.00e+000 1.73e-001  -6.3 1.43e-002    -
 7.03e-001 2.36e-001f  1
 42 8.9438531e-001 0.00e+000 2.76e-001  -6.9 1.69e-002    -
 8.33e-001 2.18e-001f  1
 43 8.8217010e-001 0.00e+000 1.60e-001  -7.6 2.17e-002    -
 8.63e-001 1.59e-001f  1
 44 8.6481469e-001 0.00e+000 2.80e-001  -8.3 2.63e-002    -
 8.70e-001 2.13e-001f  1
 45 8.5759196e-001 0.00e+000 1.91e-001  -5.1 2.21e-002    -
 8.53e-001 1.05e-001f  1
 46 8.4569914e-001 0.00e+000 1.94e-001  -4.7 1.78e-002    -
 6.09e-001 2.27e-001f  1
```

```
  47 8.6196740e-001 0.00e+000 9.31e-002  -4.1 1.12e-002    -
 5.54e-001 1.00e+000f  1
  48 8.5268512e-001 0.00e+000 4.30e-002  -4.3 1.24e-002    -
 8.41e-001 1.99e-001f  1
  49 8.2862807e-001 0.00e+000 1.08e-001  -4.3 1.26e-002    -
 5.90e-001 5.80e-001f  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du
 alpha_pr  ls
  50 8.1935375e-001 0.00e+000 3.51e-001  -5.2 1.00e-002    -
 8.50e-001 2.85e-001f  1
  51 8.1222059e-001 0.00e+000 2.66e-001  -5.2 1.24e-002    -
 8.75e-001 2.07e-001f  1
  52 8.0047092e-001 0.00e+000 3.91e-001  -5.8 1.57e-002    -
 9.64e-001 2.95e-001f  1
  53 7.9288222e-001 0.00e+000 2.18e-001  -5.1 1.56e-002    -
 9.14e-001 2.17e-001f  1
  54 7.8768612e-001 0.00e+000 1.31e-001 -11.0 2.45e-002    -
 6.70e-001 9.95e-002f  1
  55 7.7511800e-001 0.00e+000 1.26e-001  -6.1 3.21e-002    -  1.00e
+000 1.90e-001f  1
  56 7.6760504e-001 0.00e+000 1.61e-001  -4.9 2.82e-002    -  1.00e
+000 1.35e-001f  1
  57 7.6161699e-001 0.00e+000 1.90e-001 -10.9 2.04e-002    -
 3.78e-001 1.49e-001f  1
  58 7.7368687e-001 0.00e+000 9.10e-002  -4.2 1.81e-002    -
 5.61e-001 9.06e-001f  1
  59 7.6143215e-001 0.00e+000 8.92e-002  -4.5 7.85e-003    -
 2.97e-001 3.74e-001f  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du
 alpha_pr  ls
  60 7.5286735e-001 0.00e+000 1.24e-001  -4.7 1.01e-002    -
 6.11e-001 2.52e-001f  1
  61 7.4507468e-001 0.00e+000 1.38e-001  -4.7 8.69e-003    -
 6.06e-001 3.17e-001f  1
  62 7.4128976e-001 0.00e+000 2.63e-001  -5.0 7.32e-003    -
 8.56e-001 1.98e-001f  1
  63 7.3229960e-001 0.00e+000 2.25e-001  -5.1 9.65e-003    -
 9.94e-001 4.53e-001f  1
  64 7.2734268e-001 0.00e+000 1.70e-001  -5.4 9.81e-003    -  1.00e
+000 3.18e-001f  1
  65 7.2584387e-001 0.00e+000 2.95e-001  -5.7 1.17e-002    -  1.00e
+000 8.69e-002f  1
  66 7.1777429e-001 0.00e+000 1.93e-001  -6.0 2.13e-002    -
 9.17e-001 2.76e-001f  1
  67 7.1205463e-001 0.00e+000 2.08e-001  -6.2 2.28e-002    -
 9.46e-001 2.00e-001f  1
  68 7.0659682e-001 0.00e+000 1.84e-001  -6.4 3.41e-002    -
 9.72e-001 1.30e-001f  1
  69 7.0107370e-001 0.00e+000 1.70e-001  -5.6 3.58e-002    -
 5.82e-001 1.32e-001f  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du
 alpha_pr  ls
  70 6.9558441e-001 0.00e+000 1.84e-001  -4.8 1.87e-002    -
 7.88e-001 2.78e-001f  1
```

```
  71 6.9277075e-001 0.00e+000 1.39e-001 -10.9 1.85e-002    -
 4.66e-001 1.38e-001f  1
  72 6.8968311e-001 0.00e+000 1.10e-001  -4.9 2.09e-002    -
 4.23e-001 1.50e-001f  1
  73 6.8649099e-001 0.00e+000 1.13e-001 -11.0 2.97e-002    -
 3.14e-001 1.08e-001f  1
  74 6.8147525e-001 0.00e+000 1.51e-001  -5.5 4.10e-002    -
 7.60e-001 1.33e-001f  1
  75 6.7383574e-001 0.00e+000 1.27e-001  -6.2 4.02e-002    -
 4.70e-001 2.17e-001f  1
  76 6.7032228e-001 0.00e+000 6.87e-002  -5.6 3.73e-002    -
 2.92e-001 1.14e-001f  1
  77 6.6624527e-001 0.00e+000 7.15e-002 -11.0 4.83e-002    -
 2.36e-001 1.06e-001f  1
  78 6.5994405e-001 0.00e+000 8.63e-002  -5.2 6.62e-002    -
 9.27e-001 1.76e-001f  1
  79 6.5810764e-001 0.00e+000 8.88e-002  -5.2 1.51e-002    -
 1.39e-001 1.56e-001f  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du
 alpha_pr  ls
  80 6.5448753e-001 0.00e+000 7.14e-002 -11.0 4.74e-002    -
 1.35e-001 9.43e-002f  1
  81 6.4953972e-001 0.00e+000 9.28e-002  -5.3 5.09e-002    -
 4.12e-001 1.31e-001f  1
  82 6.4738530e-001 0.00e+000 1.00e-001  -5.4 3.23e-002    -
 4.01e-001 8.68e-002f  1
  83 6.4410659e-001 0.00e+000 7.72e-002  -6.0 3.71e-002    -
 3.11e-001 1.18e-001f  1
  84 6.4190966e-001 0.00e+000 1.21e-001  -5.2 2.58e-002    -
 3.66e-001 1.17e-001f  1
  85 6.3801796e-001 0.00e+000 1.24e-001  -5.8 2.98e-002    -
 2.28e-001 1.88e-001f  1
  86 6.3662816e-001 0.00e+000 1.42e-001  -5.5 3.00e-002    -
 5.92e-001 7.02e-002f  1
  87 6.3090687e-001 0.00e+000 1.37e-001  -5.2 4.52e-002    -
 6.91e-001 2.10e-001f  1
  88 6.2784489e-001 0.00e+000 1.10e-001  -5.0 3.28e-002    -
 4.13e-001 1.72e-001f  1
  89 6.2538138e-001 0.00e+000 1.09e-001  -7.2 3.77e-002    -
 3.72e-001 1.16e-001f  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du
 alpha_pr  ls
  90 6.2243814e-001 0.00e+000 1.23e-001  -5.9 5.85e-002    -
 5.59e-001 9.48e-002f  1
  91 6.1827962e-001 0.00e+000 1.45e-001  -5.6 3.99e-002    -
 5.61e-001 1.95e-001f  1
  92 6.1440236e-001 0.00e+000 1.12e-001  -5.9 5.50e-002    -
 6.34e-001 1.39e-001f  1
  93 6.1255129e-001 0.00e+000 9.60e-002  -5.3 5.32e-002    -
 4.46e-001 6.96e-002f  1
  94 6.0873153e-001 0.00e+000 1.97e-001  -5.2 3.84e-002    -
 5.96e-001 2.13e-001f  1
  95 6.2372616e-001 0.00e+000 1.15e-001  -4.4 3.61e-002    -
 6.53e-001 4.52e-001f  1
```

```
 96 6.1698162e-001 0.00e+000 8.32e-002  -4.6 8.11e-003    -
 2.67e-001 3.62e-001f  1
 97 6.0798584e-001 0.00e+000 6.81e-002  -4.6 1.14e-002    -
 2.31e-001 5.50e-001f  1
 98 6.0641974e-001 0.00e+000 1.20e-001  -4.9 8.50e-003    -
 8.40e-001 1.72e-001f  1
 99 6.0153746e-001 0.00e+000 1.45e-001  -5.0 9.85e-003    -
 8.82e-001 6.10e-001f  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du
 alpha_pr  ls
 100 6.0052076e-001 0.00e+000 1.54e-001  -5.6 8.20e-003    -  1.00e
+000 1.87e-001f  1
 101 5.9854053e-001 0.00e+000 1.15e-001  -5.2 1.08e-002    -  1.00e
+000 2.90e-001f  1
 102 5.9627202e-001 0.00e+000 1.55e-001  -5.4 1.34e-002    -  1.00e
+000 2.86e-001f  1
 103 5.9476305e-001 0.00e+000 1.04e-001  -5.6 1.83e-002    -
 9.77e-001 1.55e-001f  1
 104 5.9197521e-001 0.00e+000 1.77e-001  -5.2 1.99e-002    -
 9.78e-001 2.87e-001f  1
 105 5.9007188e-001 0.00e+000 9.47e-002  -4.8 2.09e-002    -
 6.85e-001 2.51e-001f  1
 106 5.8910072e-001 0.00e+000 1.57e-001 -10.9 2.39e-002    -
 3.68e-001 9.73e-002f  1
 107 5.8519508e-001 0.00e+000 1.03e-001  -5.5 5.44e-002    -
 3.75e-001 1.93e-001f  1
 108 5.8383070e-001 0.00e+000 8.44e-002  -5.6 5.43e-002    -
 7.31e-001 5.73e-002f  1
 109 5.8217955e-001 0.00e+000 1.67e-001  -7.0 2.80e-002    -
 3.30e-001 1.33e-001f  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du
 alpha_pr  ls
 110 5.7866264e-001 0.00e+000 1.23e-001  -5.4 4.72e-002    -
 5.01e-001 1.89e-001f  1
 111 6.1250506e-001 0.00e+000 3.36e-001  -4.2 3.84e-002    -
 5.60e-001 1.00e+000f  1
 112 5.9933955e-001 0.00e+000 4.23e-002  -4.5 2.81e-002    -
 8.89e-001 4.51e-001f  1
 113 5.9179881e-001 0.00e+000 1.21e-001  -4.5 1.72e-002    -
 8.70e-001 3.46e-001f  1
 114 5.8450074e-001 0.00e+000 7.25e-002  -4.5 1.48e-002    -
 9.29e-001 5.73e-001f  1
 115 5.8290619e-001 0.00e+000 1.13e-001  -4.5 7.07e-003    -
 5.71e-001 8.56e-001f  1
 116 5.8436066e-001 0.00e+000 1.12e-001  -4.5 9.06e-003    -
 7.29e-001 1.00e+000f  1
 117 5.8684706e-001 0.00e+000 4.21e-002  -4.5 1.43e-002    -
 7.19e-001 9.92e-001f  1
 118 5.9120456e-001 0.00e+000 7.84e-002  -4.5 1.77e-002    -
 9.59e-001 1.00e+000f  1
 119 5.8861106e-001 0.00e+000 3.85e-002  -4.5 3.84e-003    -  1.00e
+000 1.00e+000f  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du
 alpha_pr  ls
```

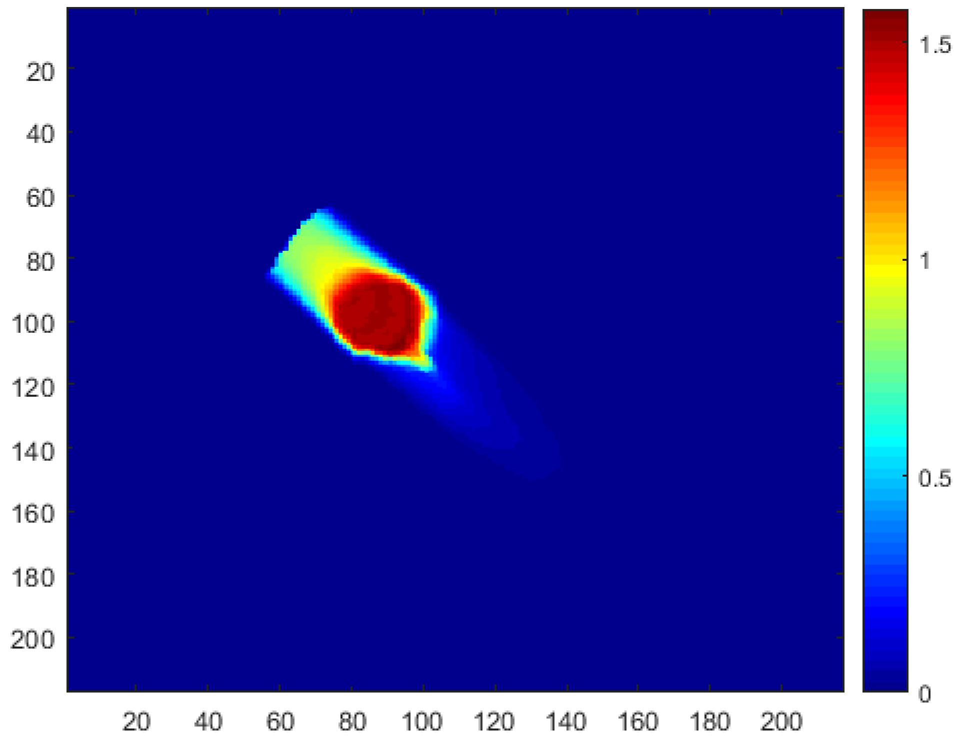
```
 120 5.8771886e-001 0.00e+000 2.64e-002  -4.5 4.26e-003    -
 7.55e-001 1.00e+000f  1
 121 5.8752368e-001 0.00e+000 1.44e-002  -4.5 3.58e-003    -  1.00e
+000 1.00e+000f  1
 122 5.8690086e-001 0.00e+000 1.81e-002  -4.5 7.30e-003    -
 9.61e-001 1.00e+000f  1

Number of Iterations....: 122

                                   (scaled)                 (unscaled)
Objective...............:  5.8690086123591856e-001
 5.8690086123591856e-001
Dual infeasibility......:  1.8066950608328639e-002
 1.8066950608328639e-002
Constraint violation....:  0.0000000000000000e+000
 0.0000000000000000e+000
Complementarity.........:  3.4099294541645421e-005
 3.4099294541645421e-005
Overall NLP error.......:  1.8066950608328639e-002
 1.8066950608328639e-002


Number of objective function evaluations             = 128
Number of objective gradient evaluations             = 123
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)   =     12.795
Total CPU secs in NLP function evaluations           =    203.121

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



Plot the Resulting Dose Slice

Let's plot the transversal iso-center dose slice

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



Inverse Optimization for IMPT based on biological effect

To perform a dose optimization for carbon ions we can also use the biological effect instead of the RBE-weighted dose. Therefore we have to change the optimization mode and restart the optimization

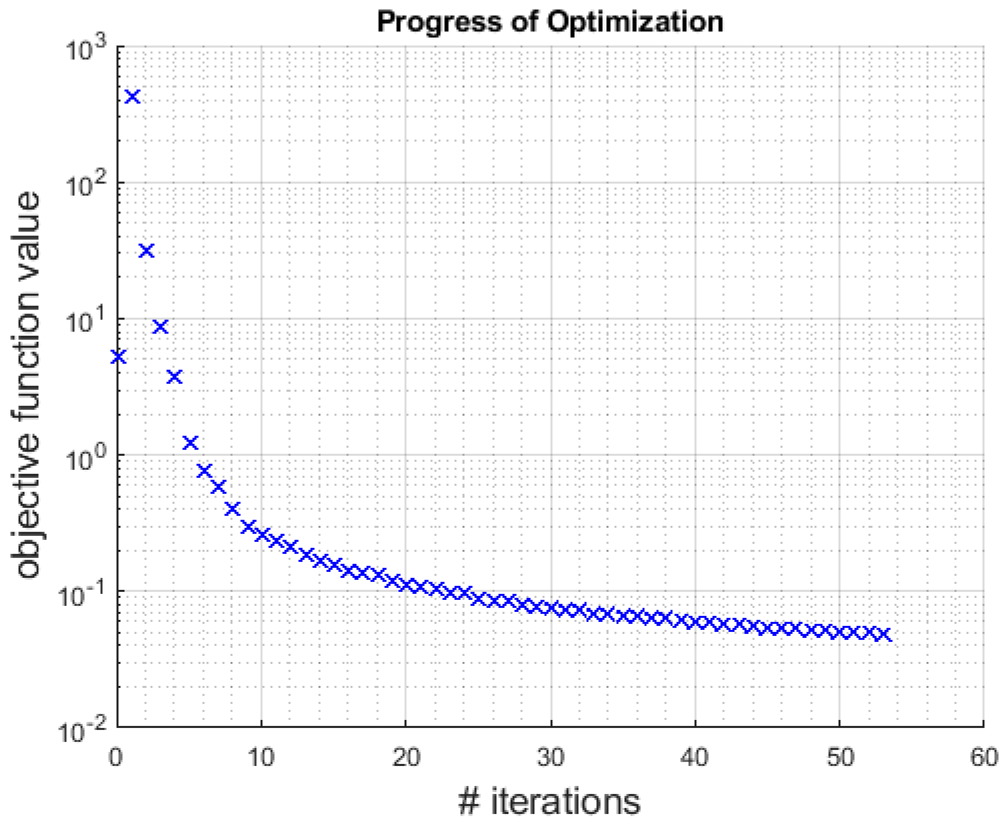
```
pln.propOpt.bioOptimization = 'LEMIV_effect';
resultGUI_effect = matRad_fluenceOptimization(dij,cst,pln);
```

```
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............................:    11746
                     variables with only lower bounds:    11746
                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 5.1855427e+000 0.00e+000 1.11e+000   0.0 0.00e+000    -  0.00e
+000 0.00e+000   0
   1 4.2755000e+002 0.00e+000 1.06e+000  -0.9 1.18e-001    -
 9.98e-001 1.00e+000f  1
   2 3.1348617e+001 0.00e+000 1.41e-001  -1.8 9.66e-002    -  1.00e
+000 1.00e+000f  1
   3 8.6317660e+000 0.00e+000 4.55e-002  -2.7 2.35e-002    -  1.00e
+000 1.00e+000f  1
   4 3.7591916e+000 0.00e+000 4.35e-002  -2.7 1.31e-002    -  1.00e
+000 1.00e+000f  1
   5 1.2108609e+000 0.00e+000 2.57e-002  -3.6 1.68e-002    -
 9.99e-001 1.00e+000f  1
   6 7.6516399e-001 0.00e+000 2.46e-002  -4.7 1.04e-002    -  1.00e
+000 1.00e+000f  1
   7 5.8442629e-001 0.00e+000 1.96e-002  -5.3 1.00e-002    -  1.00e
+000 1.00e+000f  1
   8 4.0530974e-001 0.00e+000 1.44e-002  -5.2 1.73e-002    -  1.00e
+000 1.00e+000f  1
   9 3.0181281e-001 0.00e+000 1.91e-002  -5.3 2.44e-002    -  1.00e
+000 9.90e-001f  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du
 alpha_pr  ls
  10 2.5782560e-001 0.00e+000 1.27e-002  -5.1 2.28e-002    -  1.00e
+000 5.96e-001f  1
  11 2.3743469e-001 0.00e+000 3.37e-002  -4.5 8.54e-003    -  1.00e
+000 1.00e+000f  1
  12 2.0958394e-001 0.00e+000 7.36e-003  -4.6 9.43e-003    -
 7.98e-001 1.00e+000f  1
  13 1.8488168e-001 0.00e+000 9.89e-003  -4.6 1.81e-002    -  1.00e
+000 1.00e+000f  1
  14 1.6957651e-001 0.00e+000 1.29e-002  -5.3 1.65e-002    -  1.00e
+000 3.68e-001f  1
  15 1.5725484e-001 0.00e+000 3.16e-002  -5.3 1.60e-002    -  1.00e
+000 3.77e-001f  1
  16 1.4098214e-001 0.00e+000 1.54e-002  -5.7 1.78e-002    -  1.00e
+000 6.89e-001f  1
  17 1.3508370e-001 0.00e+000 7.60e-003  -6.2 1.44e-002    -  1.00e
+000 4.05e-001f  1
  18 1.3057202e-001 0.00e+000 8.91e-003  -6.4 2.25e-002    -  1.00e
+000 1.80e-001f  1
```

```
 19 1.2056887e-001 0.00e+000 1.37e-002  -4.7 2.31e-002    -
 6.74e-001 4.33e-001f  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du
 alpha_pr  ls
 20 1.1218996e-001 0.00e+000 1.44e-002  -4.6 1.09e-002    -
 6.23e-001 1.00e+000f  1
 21 1.0733473e-001 0.00e+000 6.24e-003  -5.5 1.45e-002    -
 7.85e-001 3.73e-001f  1
 22 1.0342589e-001 0.00e+000 2.40e-002  -5.7 1.94e-002    -  1.00e
+000 2.76e-001f  1
 23 9.6066702e-002 0.00e+000 1.09e-002  -5.9 3.03e-002    -  1.00e
+000 3.95e-001f  1
 24 9.7944339e-002 0.00e+000 2.72e-002  -4.7 5.26e-002    -
 7.94e-001 9.23e-001f  1
 25 8.9174618e-002 0.00e+000 2.59e-002  -4.8 8.72e-003    -  1.00e
+000 1.00e+000f  1
 26 8.5897334e-002 0.00e+000 5.83e-003  -5.0 6.20e-003    -
 8.86e-001 1.00e+000f  1
 27 8.4348766e-002 0.00e+000 1.34e-002  -5.7 1.21e-002    -  1.00e
+000 3.22e-001f  1
 28 8.0838902e-002 0.00e+000 1.54e-002  -6.0 2.38e-002    -  1.00e
+000 3.76e-001f  1
 29 7.7778555e-002 0.00e+000 1.25e-002  -6.5 3.09e-002    -
 8.85e-001 2.66e-001f  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du
 alpha_pr  ls
 30 7.5379852e-002 0.00e+000 1.68e-002  -7.8 3.82e-002    -
 7.59e-001 1.78e-001f  1
 31 7.2002404e-002 0.00e+000 1.39e-002  -5.5 2.94e-002    -
 7.25e-001 3.32e-001f  1
 32 7.1195720e-002 0.00e+000 9.70e-003  -5.0 1.54e-002    -
 6.17e-001 1.00e+000f  1
 33 6.8157527e-002 0.00e+000 3.62e-003  -5.5 1.15e-002    -
 8.11e-001 6.99e-001f  1
 34 6.7523722e-002 0.00e+000 9.14e-003  -5.5 8.03e-003    -  1.00e
+000 2.07e-001f  1
 35 6.5282594e-002 0.00e+000 1.38e-002  -6.1 1.43e-002    -
 7.41e-001 4.49e-001f  1
 36 6.4681890e-002 0.00e+000 1.52e-002  -6.6 1.39e-002    -
 9.73e-001 1.42e-001f  1
 37 6.3038959e-002 0.00e+000 1.12e-002  -5.8 1.65e-002    -
 7.60e-001 3.20e-001f  1
 38 6.2313039e-002 0.00e+000 8.17e-003 -11.0 1.93e-002    -
 5.15e-001 1.28e-001f  1
 39 6.1420010e-002 0.00e+000 2.11e-002  -6.9 2.42e-002    -
 8.25e-001 1.25e-001f  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du
 alpha_pr  ls
 40 5.9678287e-002 0.00e+000 1.04e-002  -5.9 3.05e-002    -
 7.52e-001 1.88e-001f  1
 41 5.8747349e-002 0.00e+000 1.40e-002  -7.1 2.00e-002    -
 4.51e-001 1.56e-001f  1
 42 5.7550921e-002 0.00e+000 1.11e-002  -6.4 2.85e-002    -
 7.54e-001 1.44e-001f  1
```

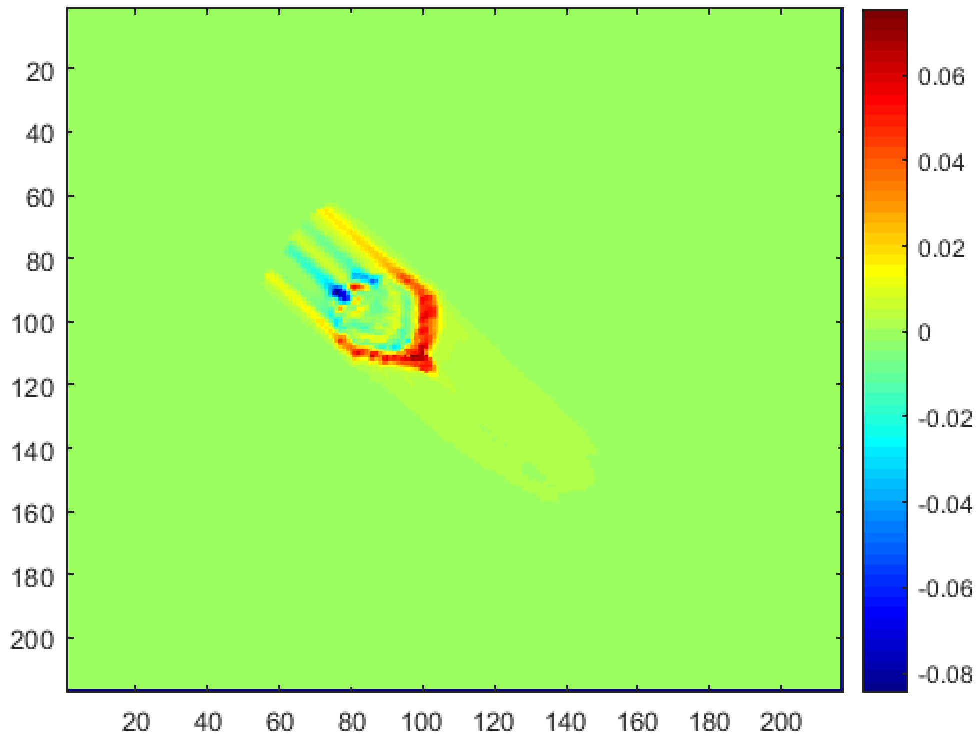
```
  43 5.6762409e-002 0.00e+000 1.12e-002  -7.3 1.98e-002    -
 2.77e-001 1.26e-001f  1
  44 5.5009283e-002 0.00e+000 8.54e-003  -6.4 3.11e-002    -
 8.78e-001 1.94e-001f  1
  45 5.3956283e-002 0.00e+000 7.45e-003  -6.4 3.70e-002    -
 5.84e-001 9.41e-002f  1
  46 5.3573327e-002 0.00e+000 8.65e-003 -11.0 1.13e-002    -
 1.16e-001 1.17e-001f  1
  47 5.2681151e-002 0.00e+000 6.65e-003  -7.2 2.42e-002    -
 3.18e-001 1.34e-001f  1
  48 5.1599911e-002 0.00e+000 5.76e-003  -5.9 3.38e-002    -
 7.09e-001 1.56e-001f  1
  49 5.1089502e-002 0.00e+000 1.46e-002  -6.0 1.36e-002    -
 3.20e-001 1.62e-001f  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du
 alpha_pr  ls
  50 4.9985884e-002 0.00e+000 1.02e-002  -5.4 2.55e-002    -
 6.94e-001 3.89e-001f  1
  51 4.9650930e-002 0.00e+000 1.17e-002  -6.5 1.40e-002    -
 6.12e-001 1.14e-001f  1
  52 4.9066823e-002 0.00e+000 7.84e-003  -7.4 1.95e-002    -
 5.56e-001 1.57e-001f  1
  53 4.8105177e-002 0.00e+000 1.13e-002  -6.4 2.22e-002    -
 5.29e-001 2.65e-001f  1

Number of Iterations....: 53

                                   (scaled)                 (unscaled)
Objective...............:  4.8105176846303319e-002
 4.8105176846303319e-002
Dual infeasibility......:  1.1262259994036907e-002
 1.1262259994036907e-002
Constraint violation....:  0.0000000000000000e+000
 0.0000000000000000e+000
Complementarity.........:  3.0220184161354036e-006
 3.0220184161354036e-006
Overall NLP error.......:  1.1262259994036907e-002
 1.1262259994036907e-002


Number of objective function evaluations             = 54
Number of objective gradient evaluations             = 54
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)   =      5.378
Total CPU secs in NLP function evaluations           =     61.932

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



Visualize differences

Through optimzation based on the biological effect we obtain a slightly different dose distribution as visualized by the following dose difference map

```
figure;
imagesc(resultGUI.RBExDose (:,:,slice)-
resultGUI_effect.RBExDose(:,:,slice));
colorbar;
colormap(jet);
```



Change Radiosensitivity

The previous treatment plan was optimized using an photon alpha-beta ratio of 2 for all tissues. Now, Let's change the radiosensitivity by adapting alphaX. This will change the photon alpha-beta ratio from 2 to 10.

```
for i = 1:size(cst,1)
    cst{i,5}.alphaX      = 0.5;
    cst{i,5}.TissueClass = 2;
end
```

Recalculate Plan

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

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

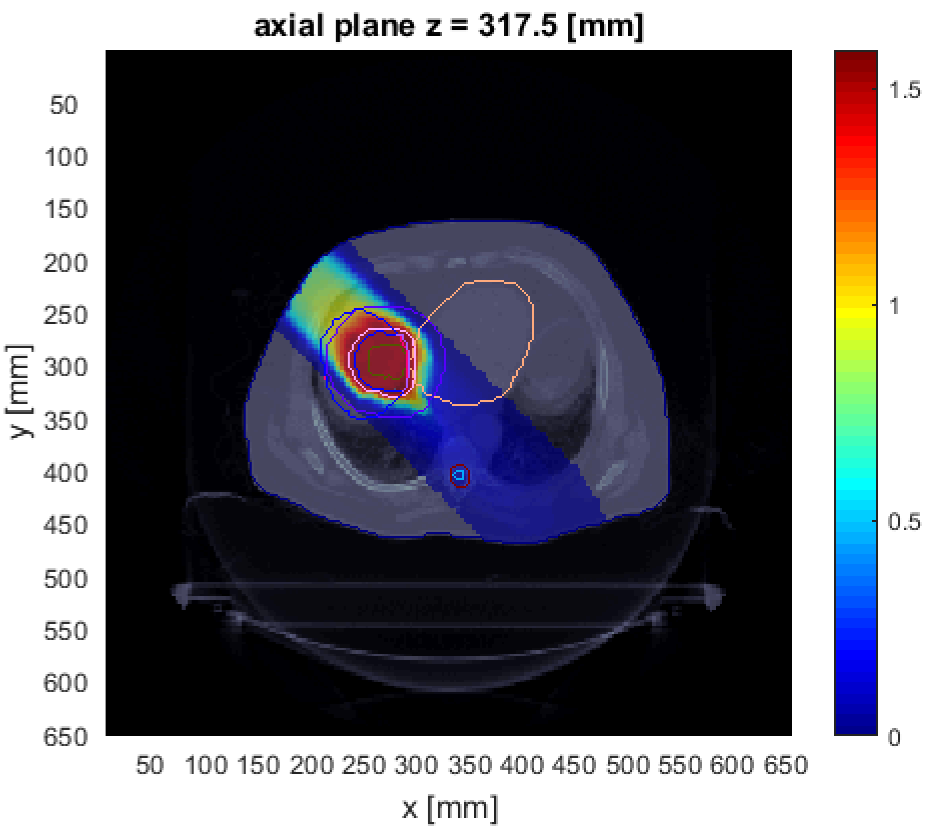
```
Warning: Could not find HLUT GE MEDICAL SYSTEMS-Discovery CT590
RT-ConvolutionKernel-STANDARD_carbon.hlut in hlutLibrary folder.
 matRad default
HLUT loaded
matRad: loading biological base data... done.
matRad: Particle dose calculation...
Beam 1 of 1:
matRad: calculate radiological depth cube...done.
matRad: calculate lateral cutoff...done.
```

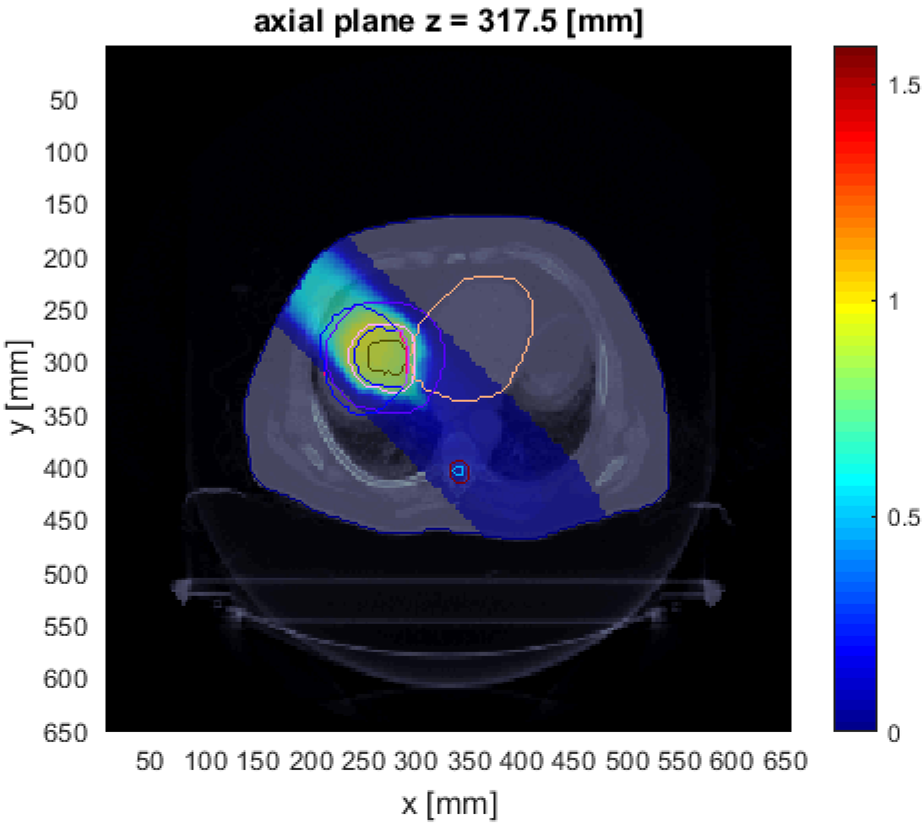
Progress: 100.00 %

Result Comparison

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

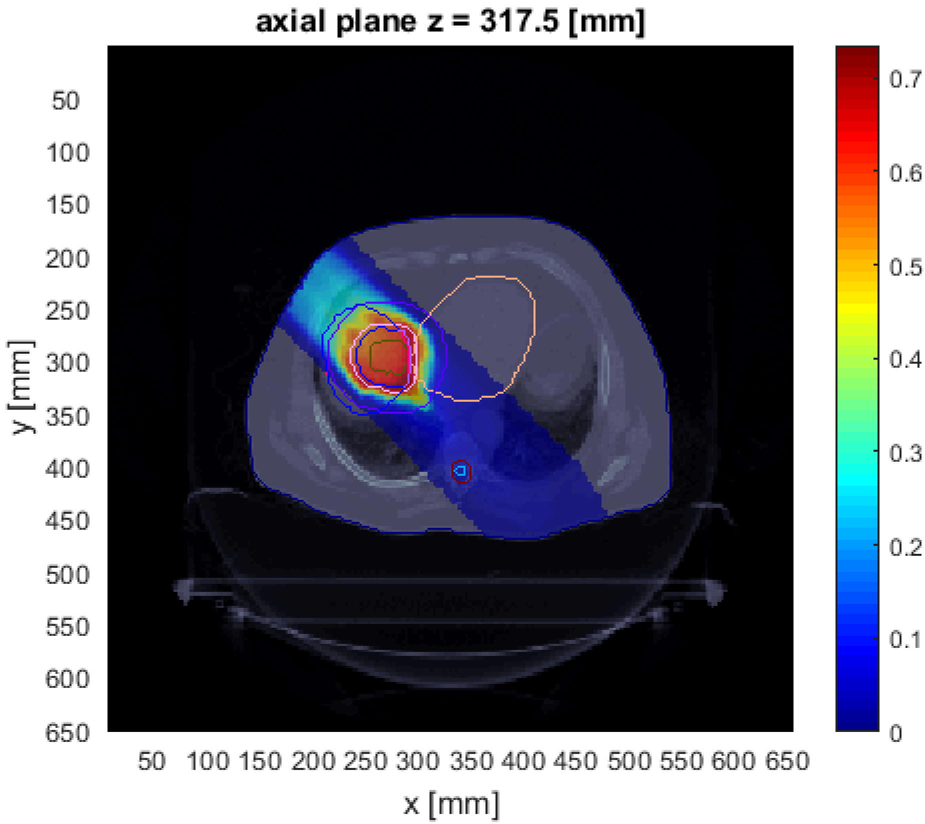
```
plane = 3;
doseWindow = [0 max([resultGUI_effect.RBExDose(:);
 resultGUI_tissue.RBExDose(:)])];

figure,title('original plan')
matRad_plotSliceWrapper(gca,ct,cst,1,resultGUI_effect.RBExDose,plane,slice,
[],[],colorcube,[],doseWindow,[]);
figure,title('manipulated plan')
matRad_plotSliceWrapper(gca,ct,cst,1,resultGUI_tissue.RBExDose,plane,slice,
[],[],colorcube,[],doseWindow,[]);
```





At this point we would like to see the absolute difference of the original optimization and the recalculation.

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



Published with MATLAB® R2018a