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 inversly 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 optimzed pencil beam intensities (vi) how to compare the two results

Patient Data Import

Let's begin with a clear Matlab environment. First, import the prostate patient into your workspace. The phantom is comprised of a 'ct' and 'cst' structure defining the CT images and the structure set. Make sure the matRad root directory 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 interests 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 corresponding 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]
              []
              []
```

The fifth column represents meta parameters used for optimization such as the overlap priority, which can be specified in double presision. A lower overlap priority indicates increased importance. In contrast, a higher overlap priority indicatets a strcture with lower importance. The parameters alphaX and betaX depict the tissue's photon-radiosensitivity parameter of the linear quadratic model. These parameter are required for biological treatment planning using a variable RBE. Let's output the meta optimization parameter of the rectum, which is stored in the eighth row:

```
cst{8,2}
cst{8,5}
```

```
ans =

    'Rectum'


ans =

  struct with fields:

    TissueClass: 1
         alphaX: 0.1000
          betaX: 0.0500
       Priority: 3
        Visible: 1
```

The sixth column contains optimization information such as objectives and constraints which are required to calculate the objective function value. Please note, that multiple objectives/constraints can be defined for individual structures. As the rectum is an OAR, we have defined and squared overdosing objective so that it is considered to be expensive/costly for the optimizer delivering more than 50 Gy to the rectum.

```
cst{8,6}
```

```
ans =

  struct with fields:

          type: 'square overdosing'
       penalty: 300
          dose: 50
           EUD: NaN
        volume: NaN
    robustness: 'none'
```

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 example we would like to use protons for treatment planning. 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 = 'protons';
```

```
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 use protons, we follow here the clinical standard and use a constant relative biological effectivness of 1.1. Therefore we set bioOptimization to const_RBExD

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

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. We define two opposing beams. For the first beam we set the gantry angle to 90 degree and the corresponding couch angle to 0 degree. The second beam possess a gantry angle of 270 degree and a couch angle of 0 degree. Furthermore, we want the lateral pencil beam spacing in x and y to be 3 mm in the iso-center plane which is perpendicular to the central beam ray. In total we are using 30 fractions. It is noteworthy that matRad is always optimizing the fraction dose.

```
pln.gantryAngles    = [90 270];
pln.couchAngles     = [0 0];
pln.bixelWidth      = 3;
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);
```

Disable sequencing and direct aperture optimization, since we have a particle plan.

```
pln.runDAO        = 0;
pln.runSequencing = 0;
```

and ZACK! our treatment plan is ready. Lets have a look at it:

```
pln
```

```
pln =

  struct with fields:

      radiationMode: 'protons'
            machine: 'Generic'
    bioOptimization: 'const_RBExD'
       gantryAngles: [90 270]
        couchAngles: [0 0]
         bixelWidth: 3
     numOfFractions: 30
         numOfBeams: 2
        numOfVoxels: 3047040
    voxelDimensions: [184 184 90]
```

```
          isoCenter: [2×3 double]
             runDAO: 0
      runSequencing: 0
```

Generatet Beam Geometry STF

This acronym stands for steering file and comprises the complet beam geomtry along with ray position, pencil beam positions and energies, source to axis distance (SAD) etc.

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

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

Let's display the beam geomtry information of the second beam

```
stf(2)
```

```

ans = 

  struct with fields:

           gantryAngle: 270
            couchAngle: 0
            bixelWidth: 3
         radiationMode: 'protons'
                   SAD: 10000
             isoCenter: [263.2719 266.0622 124.0277]
             numOfRays: 843
                   ray: [1×843 struct]
       sourcePoint_bev: [0 -10000 0]
           sourcePoint: [-10000 0 0]
    numOfBixelsPerRay: [1×843 double]
      totalNumOfBixels: 22774
```

Dose Calculation

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

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

```
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 Optimizaiton 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);
```

```
******************************************************************************
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............................:    45574
                     variables with only lower bounds:    45574
                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 4.3490711e+002 0.00e+000 1.07e+000   0.0 0.00e+000    -  0.00e
+000 0.00e+000   0
   1 4.0427541e+002 0.00e+000 7.37e-002  -1.1 7.77e-002    -
 9.91e-001 1.00e+000f  1
   2 7.0982048e+001 0.00e+000 1.97e-002  -1.7 1.37e+000    -
 9.96e-001 1.00e+000f  1
   3 3.6109959e+001 0.00e+000 1.27e-002  -3.4 3.84e-001    -
 9.75e-001 1.00e+000f  1
   4 2.9072905e+001 0.00e+000 1.06e-002  -3.9 2.77e-001    -
 9.87e-001 1.00e+000f  1
   5 2.3067104e+001 0.00e+000 1.03e-002  -4.7 4.22e-001    -
 9.99e-001 1.00e+000f  1
   6 1.9295712e+001 0.00e+000 1.36e-002  -5.5 6.66e-001    -  1.00e
+000 1.00e+000f  1
   7 1.6242709e+001 0.00e+000 7.13e-003  -6.0 2.70e-001    -  1.00e
+000 1.00e+000f  1
   8 1.5129109e+001 0.00e+000 5.99e-003  -7.2 2.09e-001    -  1.00e
+000 1.00e+000f  1
```

```
   9 1.3740190e+001 0.00e+000 4.75e-003  -8.5 3.81e-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 1.1965389e+001 0.00e+000 3.76e-003  -9.5 6.21e-001    -  1.00e
+000 1.00e+000f  1
  11 1.1479588e+001 0.00e+000 7.70e-003  -9.9 8.60e-001    -  1.00e
+000 6.53e-001f  1
  12 1.1466560e+001 0.00e+000 7.66e-003 -11.0 2.60e-001    -  1.00e
+000 7.48e-003f  1
  13 1.1465561e+001 0.00e+000 1.45e-002 -11.0 3.56e-001    -  1.00e
+000 4.09e-004f  1
  14 1.0213007e+001 0.00e+000 3.25e-003 -11.0 4.72e-001    -  1.00e
+000 6.09e-001f  1
  15 1.0205353e+001 0.00e+000 3.21e-003 -11.0 2.79e-001    -  1.00e
+000 1.07e-002f  1
  16 1.0205149e+001 0.00e+000 1.21e-002 -11.0 4.04e-001    -  1.00e
+000 1.94e-004f  1
  17 1.0116738e+001 0.00e+000 2.99e-003 -11.0 5.25e-001    -  1.00e
+000 6.49e-002f  1
  18 1.0105841e+001 0.00e+000 7.65e-003  -8.8 5.86e-001    -
 8.87e-001 6.98e-003f  1
  19 1.0022232e+001 0.00e+000 1.17e-002  -9.4 7.34e-001    -  1.00e
+000 4.42e-002f  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du
 alpha_pr  ls
  20 9.8338336e+000 0.00e+000 7.44e-003 -10.4 8.83e-001    -  1.00e
+000 9.00e-002f  1
  21 9.8218623e+000 0.00e+000 1.35e-002 -11.0 9.07e-001    -  1.00e
+000 6.01e-003f  1
  22 9.7326587e+000 0.00e+000 7.49e-003 -11.0 1.05e+000    -  1.00e
+000 4.17e-002f  1
  23 9.5583562e+000 0.00e+000 7.91e-003 -11.0 1.25e+000    -  1.00e
+000 7.39e-002f  1
  24 9.5339743e+000 0.00e+000 1.43e-002  -7.1 1.55e+000    -
 7.60e-001 8.64e-003f  1
  25 9.4272717e+000 0.00e+000 2.80e-002  -5.2 1.51e+000    -
 8.36e-001 4.07e-002f  1
  26 9.2245906e+000 0.00e+000 1.35e-002  -4.3 1.89e+000    -
 7.03e-001 7.85e-002f  1
  27 9.0902825e+000 0.00e+000 8.31e-003  -6.3 1.51e+000    -
 2.57e-001 5.81e-002f  1
  28 8.7127219e+000 0.00e+000 7.13e-003  -4.6 1.58e+000    -
 8.37e-001 1.87e-001f  1
  29 8.4541155e+000 0.00e+000 5.79e-003  -4.3 1.41e+000    -
 4.99e-001 1.54e-001f  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du
 alpha_pr  ls
  30 8.3483122e+000 0.00e+000 8.72e-003 -10.4 1.07e+000    -
 3.79e-001 1.02e-001f  1
  31 8.2324208e+000 0.00e+000 6.02e-003  -4.9 1.72e+000    -
 9.95e-001 2.60e-001f  1
  32 8.2055840e+000 0.00e+000 3.46e-002  -4.6 5.96e-001    -  1.00e
+000 3.25e-002f  1
```

```
 33 7.8920138e+000 0.00e+000 2.55e-002  -3.7 6.32e-001    -
9.74e-001 4.28e-001f  1
 34 7.7603808e+000 0.00e+000 1.13e-002  -4.4 4.73e-001    -
5.53e-001 3.40e-001f  1
 35 7.6868406e+000 0.00e+000 9.15e-003  -4.0 4.11e-001    -
7.93e-001 2.47e-001f  1
 36 7.5730659e+000 0.00e+000 7.71e-003  -4.0 4.69e-001    -
7.89e-001 3.97e-001f  1
 37 7.5090417e+000 0.00e+000 6.82e-003  -5.0 5.12e-001    -
5.23e-001 2.53e-001f  1
 38 7.3849350e+000 0.00e+000 4.00e-003  -4.3 4.83e-001    -
5.13e-001 5.06e-001f  1
 39 7.3504179e+000 0.00e+000 6.71e-003  -4.6 4.77e-001    -
8.12e-001 1.52e-001f  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du
alpha_pr  ls
 40 7.2463583e+000 0.00e+000 1.18e-002  -4.8 5.95e-001    -
6.44e-001 4.41e-001f  1
 41 7.1774628e+000 0.00e+000 2.89e-003  -4.9 7.24e-001    -  1.00e
+000 2.49e-001f  1
 42 7.2768275e+000 0.00e+000 1.46e-002  -3.2 9.85e-001    -
3.86e-001 1.00e+000f  1
 43 7.0024445e+000 0.00e+000 2.76e-003  -3.7 6.44e-001    -
7.57e-001 1.00e+000f  1
 44 6.8758809e+000 0.00e+000 1.14e-003  -4.6 3.65e-001    -
9.96e-001 8.71e-001f  1
 45 6.8630812e+000 0.00e+000 7.60e-003  -5.4 2.81e-001    -
9.97e-001 1.61e-001f  1
 46 6.8078588e+000 0.00e+000 4.32e-003  -6.1 4.51e-001    -
7.84e-001 4.48e-001f  1
 47 6.7824785e+000 0.00e+000 4.09e-003  -7.0 4.57e-001    -
8.08e-001 1.84e-001f  1
 48 6.7318712e+000 0.00e+000 2.51e-003  -6.1 6.76e-001    -
4.80e-001 2.58e-001f  1
 49 6.7006803e+000 0.00e+000 2.88e-003  -4.5 5.87e-001    -
5.63e-001 1.48e-001f  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du
alpha_pr  ls
 50 6.6807285e+000 0.00e+000 2.71e-003  -4.8 6.32e-001    -
2.42e-001 7.51e-002f  1
 51 6.6000364e+000 0.00e+000 2.58e-003  -3.9 5.76e-001    -
2.76e-001 3.21e-001f  1
 52 7.1705913e+000 0.00e+000 2.39e-003  -2.3 1.19e+001    -
1.20e-002 9.40e-002f  1
 53 6.5623589e+000 0.00e+000 2.01e-003  -4.1 1.59e+000    -
1.32e-001 7.09e-001f  1
 54 6.5162253e+000 0.00e+000 8.59e-003  -4.5 6.85e-001    -
9.91e-001 1.12e-001f  1
 55 6.4492613e+000 0.00e+000 1.18e-002  -4.8 5.09e-001    -
9.97e-001 3.79e-001f  1
 56 6.3993245e+000 0.00e+000 5.81e-003  -5.2 4.43e-001    -
9.94e-001 4.20e-001f  1
 57 6.3584296e+000 0.00e+000 3.35e-003  -5.7 4.54e-001    -
9.74e-001 4.18e-001f  1
```

```
  58 6.3336977e+000 0.00e+000 4.89e-003  -6.8 5.67e-001    -
 7.77e-001 2.27e-001f  1
  59 6.3114075e+000 0.00e+000 5.24e-003  -4.6 4.63e-001    -
 4.32e-001 2.36e-001f  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du
 alpha_pr  ls
  60 6.2655808e+000 0.00e+000 2.20e-003  -4.4 4.84e-001    -
 3.71e-001 4.40e-001f  1
  61 6.2300813e+000 0.00e+000 1.96e-003  -4.4 6.70e-001    -
 4.09e-001 2.48e-001f  1
  62 6.2161174e+000 0.00e+000 5.53e-003  -5.4 7.32e-001    -
 3.62e-001 8.21e-002f  1
  63 6.1611347e+000 0.00e+000 2.34e-003  -4.4 1.08e+000    -
 3.95e-001 2.85e-001f  1
  64 6.1553828e+000 0.00e+000 5.53e-003 -10.6 6.96e-001    -
 3.03e-001 3.10e-002f  1
  65 6.0956673e+000 0.00e+000 6.09e-003  -4.5 7.10e-001    -
 5.09e-001 2.92e-001f  1
  66 6.0717380e+000 0.00e+000 3.78e-003  -4.8 6.81e-001    -
 3.91e-001 1.29e-001f  1
  67 6.0206649e+000 0.00e+000 2.27e-003  -4.1 4.73e-001    -
 4.17e-001 3.71e-001f  1
  68 6.0000064e+000 0.00e+000 3.29e-003 -10.3 5.01e-001    -
 4.01e-001 1.87e-001f  1
  69 5.9666369e+000 0.00e+000 3.56e-003  -5.1 5.42e-001    -
 7.43e-001 3.02e-001f  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du
 alpha_pr  ls
  70 5.9492554e+000 0.00e+000 4.28e-003  -5.2 5.31e-001    -
 8.06e-001 1.56e-001f  1
  71 5.9137852e+000 0.00e+000 5.99e-003  -5.1 6.17e-001    -
 9.63e-001 3.33e-001f  1
  72 5.8903674e+000 0.00e+000 1.80e-003  -4.8 5.61e-001    -
 7.76e-001 1.98e-001f  1
  73 5.8631878e+000 0.00e+000 8.91e-003  -4.3 6.28e-001    -
 7.41e-001 2.02e-001f  1
  74 6.7925850e+000 0.00e+000 1.12e-002  -2.4 4.55e+001    -
 3.32e-002 6.90e-002f  1
  75 5.9297116e+000 0.00e+000 1.08e-002  -3.9 2.26e+000    -
 7.67e-002 7.85e-001f  1
  76 5.8122623e+000 0.00e+000 6.73e-003  -3.9 3.75e-001    -
 5.94e-001 8.19e-001f  1
  77 5.7843265e+000 0.00e+000 1.04e-002  -4.7 4.33e-001    -
 7.15e-001 3.20e-001f  1
  78 5.7493480e+000 0.00e+000 5.44e-003  -4.9 5.16e-001    -
 9.54e-001 4.26e-001f  1
  79 5.7247047e+000 0.00e+000 3.97e-003  -5.5 5.74e-001    -
 9.63e-001 3.12e-001f  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du
 alpha_pr  ls
  80 5.7009930e+000 0.00e+000 4.50e-003  -4.6 4.89e-001    -
 5.76e-001 3.67e-001f  1
  81 5.6781298e+000 0.00e+000 4.94e-003  -4.5 2.95e-001    -
 4.73e-001 5.67e-001f  1
```

```
 82 5.6530706e+000 0.00e+000 2.11e-003  -4.2 2.78e-001    -
4.75e-001 5.01e-001f  1
 83 5.6504971e+000 0.00e+000 6.65e-003 -10.4 8.13e-001    -
2.72e-001 1.85e-002f  1
 84 5.5936357e+000 0.00e+000 2.01e-003  -4.7 1.12e+000    -
4.92e-001 3.27e-001f  1
 85 5.5779807e+000 0.00e+000 5.62e-003  -4.6 9.93e-001    -
5.82e-001 8.92e-002f  1
 86 5.5401800e+000 0.00e+000 3.77e-003  -4.6 7.45e-001    -
3.59e-001 3.27e-001f  1
 87 5.5152974e+000 0.00e+000 2.75e-003  -4.9 7.53e-001    -
6.03e-001 2.51e-001f  1
 88 5.5005425e+000 0.00e+000 3.37e-003  -5.0 6.70e-001    -
3.96e-001 1.78e-001f  1
 89 5.4777625e+000 0.00e+000 3.83e-003  -5.2 7.06e-001    -
7.40e-001 2.76e-001f  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du
alpha_pr  ls
 90 5.4595663e+000 0.00e+000 3.62e-003  -7.1 8.75e-001    -
3.65e-001 1.98e-001f  1
 91 5.4342043e+000 0.00e+000 2.37e-003  -4.7 6.15e-001    -
4.37e-001 3.81e-001f  1
 92 5.4070272e+000 0.00e+000 3.61e-003  -4.3 2.62e-001    -
4.87e-001 8.77e-001f  1
 93 5.4006834e+000 0.00e+000 2.06e-003  -4.8 6.12e-001    -
4.55e-001 8.82e-002f  1
 94 5.3797485e+000 0.00e+000 3.30e-003 -10.6 7.27e-001    -
3.07e-001 2.53e-001f  1
 95 5.3596690e+000 0.00e+000 1.28e-003  -4.9 9.10e-001    -
3.21e-001 2.14e-001f  1
 96 5.3565448e+000 0.00e+000 4.75e-003 -10.8 5.35e-001    -
3.38e-001 5.24e-002f  1
 97 5.3343806e+000 0.00e+000 4.09e-003  -5.2 6.79e-001    -
7.61e-001 2.73e-001f  1
 98 5.6135977e+000 0.00e+000 3.98e-003  -3.4 5.23e+000    -
5.08e-002 4.52e-001f  1
 99 5.3707802e+000 0.00e+000 3.98e-003  -4.7 2.11e+000    -
1.27e-002 6.02e-001f  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du
alpha_pr  ls
100 5.3460347e+000 0.00e+000 4.59e-003  -4.7 7.46e-001    -
6.64e-001 1.90e-001f  1
101 5.3032521e+000 0.00e+000 8.05e-003  -4.7 6.62e-001    -
7.48e-001 4.35e-001f  1
102 5.2777204e+000 0.00e+000 7.33e-003  -4.9 5.53e-001    -
8.71e-001 4.52e-001f  1
103 5.2701740e+000 0.00e+000 5.41e-003  -5.4 4.37e-001    -
8.95e-001 1.92e-001f  1
104 5.2612836e+000 0.00e+000 6.24e-003  -6.4 5.40e-001    -
7.66e-001 1.90e-001f  1
105 5.2475871e+000 0.00e+000 6.79e-003  -6.2 6.65e-001    -
8.31e-001 2.48e-001f  1
106 5.2402506e+000 0.00e+000 5.62e-003  -5.5 7.18e-001    -
7.63e-001 1.20e-001f  1
```

```
 107 5.2202164e+000 0.00e+000 3.49e-003  -4.5 3.75e-001    -
 3.18e-001 6.81e-001f  1
 108 5.2090157e+000 0.00e+000 2.54e-003  -4.4 3.05e-001    -
 4.79e-001 1.00e+000f  1
 109 5.1724303e+000 0.00e+000 1.25e-003  -4.3 1.08e+000    -
 5.26e-001 7.04e-001f  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du
 alpha_pr  ls
 110 5.1576221e+000 0.00e+000 5.17e-003  -4.7 6.40e-001    -
 9.06e-001 3.86e-001f  1
 111 5.1482006e+000 0.00e+000 7.88e-003  -5.1 5.61e-001    -
 9.52e-001 2.57e-001f  1
 112 5.1254942e+000 0.00e+000 2.28e-003  -5.6 1.11e+000    -
 7.39e-001 2.87e-001f  1
 113 5.1076357e+000 0.00e+000 3.40e-003  -5.5 2.10e+000    -
 7.31e-001 1.38e-001f  1
 114 5.0889109e+000 0.00e+000 5.70e-003  -4.5 3.60e-001    -
 3.38e-001 5.41e-001f  1
 115 5.0676087e+000 0.00e+000 1.22e-003  -4.4 3.45e-001    -
 3.74e-001 1.00e+000f  1
 116 5.0527819e+000 0.00e+000 1.21e-003  -4.6 4.90e-001    -
 5.34e-001 4.44e-001f  1
 117 5.0426034e+000 0.00e+000 4.93e-003  -5.0 6.24e-001    -
 9.85e-001 3.11e-001f  1
 118 5.0300487e+000 0.00e+000 3.21e-003  -5.3 7.98e-001    -
 8.92e-001 3.02e-001f  1
 119 5.0221356e+000 0.00e+000 5.64e-003  -6.5 1.00e+000    -
 6.67e-001 1.40e-001f  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du
 alpha_pr  ls
 120 5.1321508e+000 0.00e+000 8.13e-003  -3.7 5.49e+000    -
 4.68e-002 2.69e-001f  1
 121 5.0681609e+000 0.00e+000 8.17e-003  -5.0 2.14e+000    -
 1.17e-002 3.20e-001f  1
 122 5.0261287e+000 0.00e+000 4.55e-003  -5.0 1.79e+000    -
 6.71e-001 2.72e-001f  1
 123 5.0046005e+000 0.00e+000 4.55e-003  -5.0 1.23e+000    -
 2.17e-001 2.37e-001f  1
 124 4.9931713e+000 0.00e+000 2.22e-003  -5.1 1.29e+000    -
 6.99e-001 1.30e-001f  1
 125 4.9665455e+000 0.00e+000 1.54e-003  -4.6 5.42e-001    -
 5.84e-001 8.38e-001f  1
 126 4.9596673e+000 0.00e+000 6.42e-003  -4.9 4.11e-001    -
 9.82e-001 2.00e-001f  1
 127 4.9470793e+000 0.00e+000 4.10e-003  -5.1 7.10e-001    -
 4.72e-001 2.04e-001f  1
 128 4.9339181e+000 0.00e+000 3.71e-003  -6.9 8.61e-001    -
 3.41e-001 1.91e-001f  1
 129 4.9245220e+000 0.00e+000 3.68e-003  -5.7 8.59e-001    -
 5.27e-001 1.44e-001f  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du
 alpha_pr  ls
 130 4.9143430e+000 0.00e+000 3.64e-003  -5.4 8.44e-001    -
 5.53e-001 1.62e-001f  1
```

```
 131 4.9014864e+000 0.00e+000 2.62e-003  -4.9 6.94e-001    -
 5.17e-001 2.48e-001f  1
 132 4.8878477e+000 0.00e+000 3.15e-003  -5.1 7.73e-001    -
 3.57e-001 2.62e-001f  1
 133 4.8812957e+000 0.00e+000 4.70e-003  -5.8 9.62e-001    -
 4.88e-001 1.07e-001f  1
 134 4.8745146e+000 0.00e+000 6.13e-003  -6.1 1.00e+000    -
 7.11e-001 1.04e-001f  1
 135 4.8594474e+000 0.00e+000 3.21e-003  -5.7 1.19e+000    -
 5.42e-001 1.94e-001f  1
 136 4.8502984e+000 0.00e+000 2.65e-003  -7.3 1.11e+000    -
 3.06e-001 1.22e-001f  1
 137 4.8397409e+000 0.00e+000 3.19e-003  -4.8 6.46e-001    -
 5.66e-001 2.29e-001f  1
 138 5.0246298e+000 0.00e+000 3.44e-003  -3.3 1.42e+001    -
 1.27e-002 1.28e-001f  1
 139 4.9052149e+000 0.00e+000 2.87e-003  -4.8 2.73e+000    -
 4.26e-002 3.86e-001f  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du
 alpha_pr  ls
 140 4.8352429e+000 0.00e+000 2.47e-003  -4.8 1.71e+000    -
 6.25e-001 4.32e-001f  1
 141 4.9899454e+000 0.00e+000 5.58e-003  -3.9 4.10e+000    -
 5.76e-002 4.11e-001f  1
 142 4.8930395e+000 0.00e+000 5.77e-003  -4.7 2.52e+000    -
 3.91e-001 4.67e-001f  1
 143 4.8593134e+000 0.00e+000 1.17e-002  -4.7 1.24e+000    -  1.00e
+000 2.42e-001f  1
 144 4.8312118e+000 0.00e+000 5.21e-003  -4.7 7.80e-001    -
 8.12e-001 3.24e-001f  1
 145 4.8227437e+000 0.00e+000 9.05e-003  -5.3 8.00e-001    -
 9.28e-001 1.12e-001f  1
 146 4.7992027e+000 0.00e+000 8.71e-003  -6.7 9.03e-001    -
 7.75e-001 3.02e-001f  1
 147 4.7950905e+000 0.00e+000 8.35e-003  -7.0 6.21e-001    -
 6.10e-001 8.11e-002f  1
 148 4.7769291e+000 0.00e+000 8.36e-003  -7.0 9.31e-001    -
 7.74e-001 2.60e-001f  1
 149 4.7659430e+000 0.00e+000 5.89e-003  -6.1 8.29e-001    -
 6.81e-001 1.94e-001f  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du
 alpha_pr  ls
 150 4.7536114e+000 0.00e+000 5.08e-003  -5.8 8.96e-001    -
 6.94e-001 2.16e-001f  1
 151 4.7478525e+000 0.00e+000 3.96e-003 -11.0 9.91e-001    -
 7.07e-001 9.39e-002f  1
 152 4.7376081e+000 0.00e+000 7.16e-003  -5.6 7.87e-001    -
 6.51e-001 2.12e-001f  1
 153 4.7293792e+000 0.00e+000 2.55e-003  -5.3 9.94e-001    -
 7.43e-001 1.36e-001f  1
 154 4.9080425e+000 0.00e+000 2.35e-003  -3.9 4.14e+000    -
 1.23e-002 5.29e-001f  1
 155 4.8175318e+000 0.00e+000 1.33e-003  -5.3 4.63e+000    -
 2.02e-001 2.38e-001f  1
```

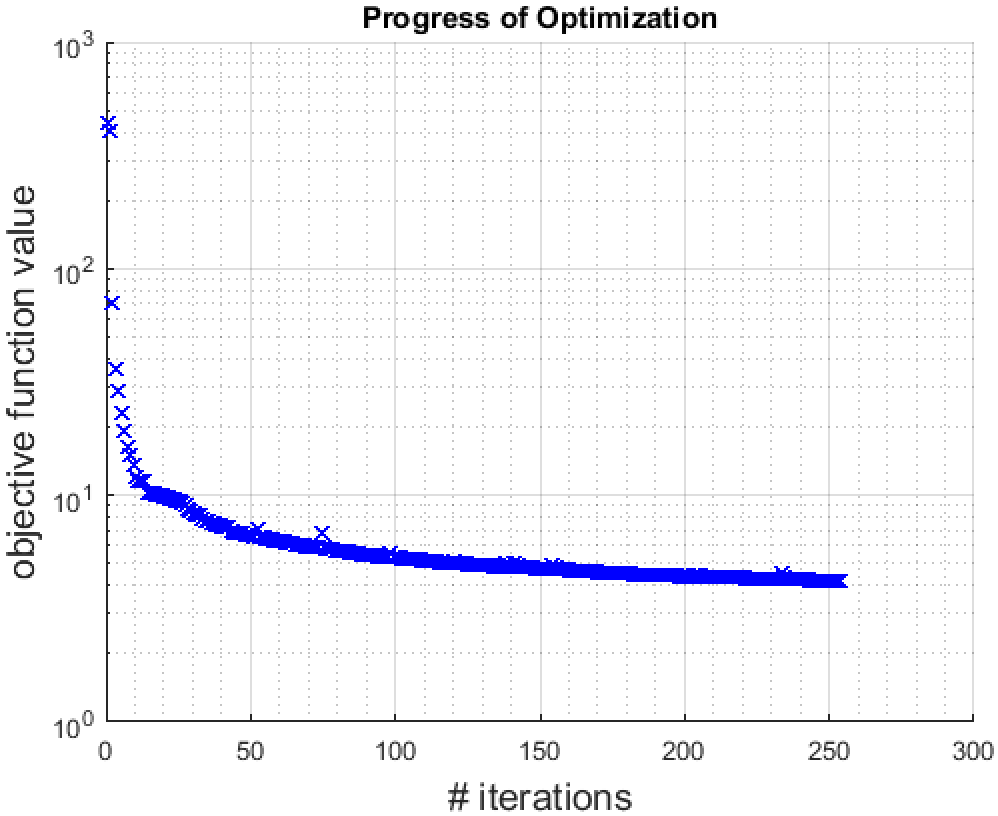
```
 156 4.7293387e+000 0.00e+000 3.56e-003  -5.3 3.82e+000    -
 7.87e-001 2.71e-001f  1
 157 4.7077953e+000 0.00e+000 1.18e-003  -4.7 4.23e-001    -
 6.52e-001 7.70e-001f  1
 158 4.7009183e+000 0.00e+000 3.86e-003  -4.8 3.39e-001    -
 7.14e-001 3.19e-001f  1
 159 4.6855947e+000 0.00e+000 2.05e-003  -4.8 5.53e-001    -
 4.37e-001 4.42e-001f  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du
 alpha_pr  ls
 160 4.6789128e+000 0.00e+000 5.59e-003  -5.1 6.85e-001    -
 5.66e-001 1.53e-001f  1
 161 4.6638348e+000 0.00e+000 2.95e-003  -5.3 1.04e+000    -
 5.17e-001 2.39e-001f  1
 162 4.6562587e+000 0.00e+000 2.11e-003  -6.4 1.29e+000    -
 4.40e-001 9.60e-002f  1
 163 4.6518499e+000 0.00e+000 5.28e-003 -11.0 1.06e+000    -
 4.60e-001 6.93e-002f  1
 164 4.6310441e+000 0.00e+000 4.82e-003  -6.7 1.59e+000    -
 2.55e-001 2.31e-001f  1
 165 4.6249851e+000 0.00e+000 1.06e-002  -4.8 3.69e-001    -
 5.45e-001 3.17e-001f  1
 166 4.6232901e+000 0.00e+000 2.29e-003  -4.6 1.05e-001    -
 4.65e-001 1.00e+000f  1
 167 4.6174714e+000 0.00e+000 3.00e-003  -5.6 7.45e-001    -
 4.93e-001 1.45e-001f  1
 168 4.6059326e+000 0.00e+000 1.88e-003  -5.1 1.17e+000    -
 4.79e-001 1.99e-001f  1
 169 4.6023980e+000 0.00e+000 4.17e-003  -5.1 8.41e-001    -
 6.23e-001 8.07e-002f  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du
 alpha_pr  ls
 170 4.5837393e+000 0.00e+000 4.10e-003  -5.1 1.23e+000    -
 6.50e-001 2.88e-001f  1
 171 4.5768137e+000 0.00e+000 2.48e-003  -7.2 1.60e+000    -
 3.01e-001 7.87e-002f  1
 172 4.5658820e+000 0.00e+000 2.45e-003 -11.0 1.28e+000    -
 6.60e-002 1.62e-001f  1
 173 4.5575992e+000 0.00e+000 2.16e-003  -6.4 1.36e+000    -
 3.66e-001 1.17e-001f  1
 174 4.5473314e+000 0.00e+000 4.42e-003  -5.2 1.10e+000    -
 4.99e-001 1.72e-001f  1
 175 4.5812473e+000 0.00e+000 4.24e-003  -3.8 5.91e+000    -
 1.73e-002 1.07e-001f  1
 176 4.5579659e+000 0.00e+000 3.81e-003  -5.2 2.17e+000    -
 6.21e-002 1.98e-001f  1
 177 4.5461480e+000 0.00e+000 1.13e-002  -5.2 1.60e+000    -
 4.99e-001 1.32e-001f  1
 178 4.5424873e+000 0.00e+000 3.91e-003 -11.0 1.50e+000    -
 3.72e-001 4.55e-002f  1
 179 4.5176030e+000 0.00e+000 1.29e-003  -4.9 1.14e+000    -
 7.31e-001 4.44e-001f  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du
 alpha_pr  ls
```

```
 180 4.5094690e+000 0.00e+000 1.88e-003  -4.7 5.30e-001    -
 5.18e-001 7.23e-001f  1
 181 4.5075355e+000 0.00e+000 5.65e-003  -6.4 6.40e-001    -
 5.32e-001 6.57e-002f  1
 182 4.4970194e+000 0.00e+000 2.26e-003  -5.3 1.20e+000    -
 8.45e-001 2.41e-001f  1
 183 4.4872889e+000 0.00e+000 2.01e-003  -5.2 1.69e+000    -
 3.84e-001 1.94e-001f  1
 184 4.4839699e+000 0.00e+000 4.69e-003 -11.0 1.19e+000    -
 4.42e-001 8.24e-002f  1
 185 4.4764595e+000 0.00e+000 3.29e-003  -7.0 1.54e+000    -
 3.46e-001 1.27e-001f  1
 186 4.4703044e+000 0.00e+000 2.00e-003  -5.6 1.81e+000    -
 2.94e-001 9.48e-002f  1
 187 4.4636883e+000 0.00e+000 3.83e-003 -11.0 1.60e+000    -
 2.39e-001 1.13e-001f  1
 188 4.4554679e+000 0.00e+000 2.92e-003  -5.8 2.07e+000    -
 4.00e-001 1.14e-001f  1
 189 4.4486254e+000 0.00e+000 3.12e-003  -4.9 6.53e-001    -
 4.14e-001 2.80e-001f  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du
 alpha_pr  ls
 190 4.4423426e+000 0.00e+000 9.88e-004  -4.7 1.81e-001    -
 3.00e-001 1.00e+000f  1
 191 4.4418497e+000 0.00e+000 6.62e-003  -5.3 6.35e-001    -
 5.50e-001 2.21e-002f  1
 192 4.4359968e+000 0.00e+000 2.44e-003  -4.8 7.21e-001    -
 5.65e-001 3.88e-001f  1
 193 4.4328624e+000 0.00e+000 9.28e-003  -5.2 5.15e-001    -
 7.88e-001 1.54e-001f  1
 194 4.4249194e+000 0.00e+000 7.45e-003  -5.6 8.07e-001    -
 7.44e-001 2.43e-001f  1
 195 4.4190046e+000 0.00e+000 5.66e-003  -5.0 7.17e-001    -
 1.54e-001 2.36e-001f  1
 196 4.4168992e+000 0.00e+000 2.60e-003  -5.3 7.15e-001    -
 4.54e-001 7.92e-002f  1
 197 4.4092729e+000 0.00e+000 4.37e-003  -5.0 5.88e-001    -
 2.95e-001 3.71e-001f  1
 198 4.4053230e+000 0.00e+000 3.22e-003  -5.2 5.48e-001    -
 5.30e-001 2.13e-001f  1
 199 4.4019562e+000 0.00e+000 3.15e-003  -5.3 5.98e-001    -
 6.55e-001 1.63e-001f  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du
 alpha_pr  ls
 200 4.3960399e+000 0.00e+000 3.28e-003  -5.3 6.43e-001    -
 4.88e-001 2.65e-001f  1
 201 4.3928333e+000 0.00e+000 2.25e-003  -6.2 7.02e-001    -
 4.32e-001 1.27e-001f  1
 202 4.3883469e+000 0.00e+000 3.19e-003  -6.1 9.59e-001    -
 6.43e-001 1.25e-001f  1
 203 4.4526888e+000 0.00e+000 3.28e-003  -3.7 8.99e+000    -
 2.11e-002 1.63e-001f  1
 204 4.4291752e+000 0.00e+000 3.24e-003  -5.3 2.00e+000    -
 8.26e-003 1.88e-001f  1
```

```
 205 4.4186553e+000 0.00e+000 2.78e-003  -5.3 1.78e+000    -
 4.51e-001 9.95e-002f  1
 206 4.3876022e+000 0.00e+000 6.17e-003  -5.3 1.64e+000    -
 4.47e-001 3.70e-001f  1
 207 4.3766507e+000 0.00e+000 6.25e-003  -4.9 5.88e-001    -
 9.27e-001 3.85e-001f  1
 208 4.3703600e+000 0.00e+000 8.91e-003  -5.0 2.30e-001    -
 8.93e-001 6.56e-001f  1
 209 4.3669725e+000 0.00e+000 3.82e-003  -5.1 2.40e-001    -
 8.20e-001 3.68e-001f  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du
 alpha_pr  ls
 210 4.3607758e+000 0.00e+000 1.99e-003  -5.2 4.85e-001    -
 5.58e-001 3.49e-001f  1
 211 4.3570178e+000 0.00e+000 2.27e-003  -5.9 6.64e-001    -
 6.29e-001 1.53e-001f  1
 212 4.3505635e+000 0.00e+000 2.43e-003  -6.2 1.09e+000    -
 5.43e-001 1.57e-001f  1
 213 4.3530460e+000 0.00e+000 1.24e-003  -4.7 1.80e-001    -
 3.67e-001 1.00e+000f  1
 214 4.3497240e+000 0.00e+000 1.28e-003  -5.2 6.47e-001    -
 5.79e-001 1.24e-001f  1
 215 4.3363643e+000 0.00e+000 1.44e-003  -5.8 1.30e+000    -
 3.88e-001 2.56e-001f  1
 216 4.3334083e+000 0.00e+000 5.60e-004  -4.7 9.83e-001    -
 3.39e-001 5.90e-001f  1
 217 4.3307562e+000 0.00e+000 2.19e-003  -5.7 6.47e-001    -
 5.54e-001 1.02e-001f  1
 218 4.3258191e+000 0.00e+000 3.83e-003  -6.3 7.69e-001    -
 4.06e-001 1.54e-001f  1
 219 4.3206024e+000 0.00e+000 4.37e-003  -5.5 7.33e-001    -
 4.87e-001 1.72e-001f  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du
 alpha_pr  ls
 220 4.3209135e+000 0.00e+000 2.73e-003  -4.8 8.03e-001    -
 2.13e-001 8.35e-001f  1
 221 4.3172121e+000 0.00e+000 1.71e-003  -5.3 7.49e-001    -
 5.32e-001 1.15e-001f  1
 222 4.3097042e+000 0.00e+000 3.82e-003  -5.0 3.51e-001    -
 4.61e-001 4.94e-001f  1
 223 4.3018036e+000 0.00e+000 2.77e-003  -5.1 6.50e-001    -
 5.61e-001 2.93e-001f  1
 224 4.2934939e+000 0.00e+000 2.67e-003  -5.5 8.45e-001    -
 4.78e-001 2.43e-001f  1
 225 4.2920178e+000 0.00e+000 3.15e-003  -5.7 6.97e-001    -
 4.91e-001 5.10e-002f  1
 226 4.2831729e+000 0.00e+000 2.36e-003  -5.8 1.21e+000    -
 4.53e-001 1.88e-001f  1
 227 4.2802664e+000 0.00e+000 1.75e-003  -5.0 5.75e-001    -
 1.45e-001 1.28e-001f  1
 228 4.2709377e+000 0.00e+000 1.31e-003  -4.8 3.31e-001    -
 1.43e-001 8.83e-001f  1
 229 4.2681270e+000 0.00e+000 1.48e-003  -5.0 3.27e-001    -
 6.95e-001 3.11e-001f  1
```

```
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du
 alpha_pr  ls
 230 4.2657188e+000 0.00e+000 3.89e-003  -5.3 4.79e-001    -
 8.35e-001 1.86e-001f  1
 231 4.2606683e+000 0.00e+000 2.41e-003  -5.7 7.64e-001    -
 7.47e-001 2.16e-001f  1
 232 4.2591376e+000 0.00e+000 3.75e-003  -6.0 7.85e-001    -
 5.67e-001 6.01e-002f  1
 233 4.5007957e+000 0.00e+000 4.13e-003  -3.7 1.50e+001    -
 1.55e-002 2.47e-001f  1
 234 4.3938442e+000 0.00e+000 4.14e-003  -5.2 3.80e+000    -
 4.02e-002 2.83e-001f  1
 235 4.3166292e+000 0.00e+000 3.76e-003  -5.2 2.87e+000    -
 5.35e-001 3.49e-001f  1
 236 4.2774569e+000 0.00e+000 7.44e-003  -5.2 1.68e+000    -
 5.50e-001 3.60e-001f  1
 237 4.2594959e+000 0.00e+000 7.01e-003  -5.2 1.01e+000    -
 7.85e-001 3.19e-001f  1
 238 4.2555723e+000 0.00e+000 4.24e-003  -5.2 5.23e-001    -
 5.20e-001 1.40e-001f  1
 239 4.2504925e+000 0.00e+000 1.10e-003  -4.9 1.44e-001    -
 3.86e-001 1.00e+000f  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du
 alpha_pr  ls
 240 4.2483438e+000 0.00e+000 6.08e-003  -5.6 4.94e-001    -  1.00e
+000 1.09e-001f  1
 241 4.2420273e+000 0.00e+000 6.51e-003  -5.5 5.84e-001    -
 8.93e-001 2.79e-001f  1
 242 4.2354161e+000 0.00e+000 1.58e-003  -5.0 1.80e-001    -
 7.94e-001 1.00e+000f  1
 243 4.2328254e+000 0.00e+000 1.38e-003  -5.2 2.60e-001    -
 4.84e-001 2.78e-001f  1
 244 4.2284939e+000 0.00e+000 1.87e-003  -5.3 6.59e-001    -
 4.52e-001 1.78e-001f  1
 245 4.2203052e+000 0.00e+000 8.72e-004  -5.2 9.14e-001    -
 7.50e-001 2.78e-001f  1
 246 4.2188917e+000 0.00e+000 2.00e-003 -11.0 7.81e-001    -
 1.76e-001 5.17e-002f  1
 247 4.2145975e+000 0.00e+000 1.27e-003  -5.6 1.12e+000    -
 4.23e-001 1.15e-001f  1
 248 4.2075353e+000 0.00e+000 2.62e-003 -11.0 1.52e+000    -
 3.18e-001 1.36e-001f  1
 249 4.2064371e+000 0.00e+000 3.83e-003  -7.7 9.33e-001    -
 2.67e-001 3.65e-002f  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du
 alpha_pr  ls
 250 4.1991106e+000 0.00e+000 2.87e-003  -5.8 1.11e+000    -
 4.65e-001 2.17e-001f  1
 251 4.1945354e+000 0.00e+000 1.53e-003  -5.4 7.75e-001    -
 3.18e-001 1.93e-001f  1
 252 4.1930819e+000 0.00e+000 3.12e-003  -5.8 6.20e-001    -
 4.26e-001 8.10e-002f  1
 253 4.1889720e+000 0.00e+000 2.01e-003  -5.1 2.22e-001    -
 3.49e-001 6.53e-001f  1
```

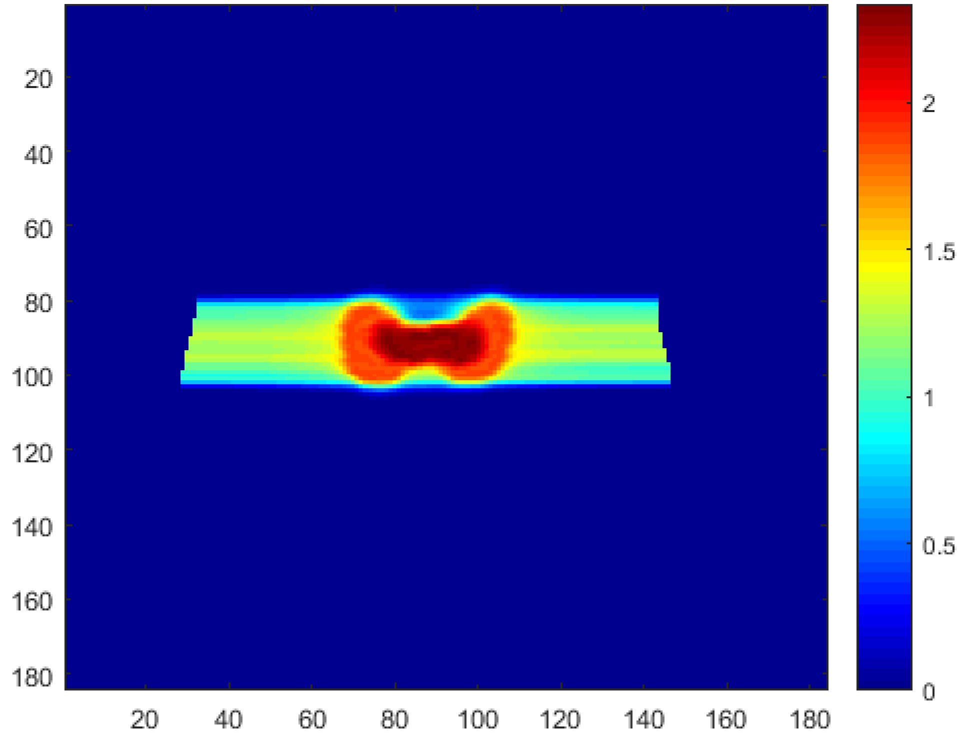
```
 254 4.1870701e+000 0.00e+000 1.80e-003  -5.1 3.28e-001    -
 4.70e-001 2.25e-001f  1

Number of Iterations....: 254

                                   (scaled)                 (unscaled)
Objective...............:  4.1870701025738564e+000
 4.1870701025738564e+000
Dual infeasibility......:  1.8038287480841529e-003
 1.8038287480841529e-003
Constraint violation....:  0.0000000000000000e+000
 0.0000000000000000e+000
Complementarity.........:  8.4111383756812913e-006
 8.4111383756812913e-006
Overall NLP error.......:  1.8038287480841529e-003
 1.8038287480841529e-003


Number of objective function evaluations             = 255
Number of objective gradient evaluations             = 255
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)   =     13.811
Total CPU secs in NLP function evaluations           =     67.812

EXIT: Solved To Acceptable Level.
Calculating final cubes...
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.isoCenter(1,3)./ct.resolution.z);
figure
imagesc(resultGUI.RBExDose(:,:,slice)),colorbar,colormap(jet)
```



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.isoCenter       = reshape([stf.isoCenter],[3 pln.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);
```

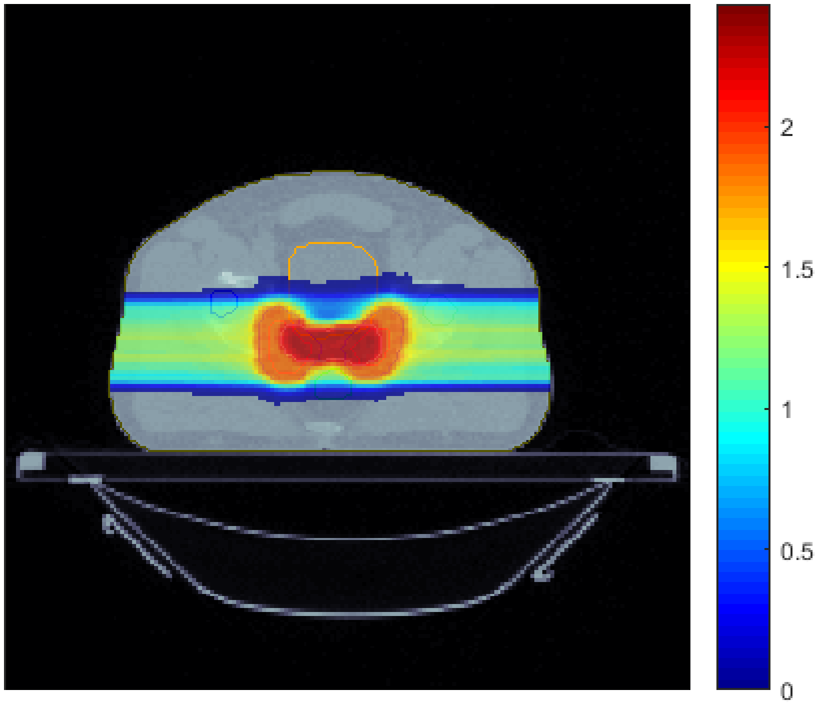
```
matRad: Using a constant RBE of 1.1
matRad: Particle dose calculation...
Beam 1 of 2:
matRad: calculate radiological depth cube...done.
matRad: calculate lateral cutoff...done.
Progress: 100.00 %
Beam 2 of 2:
matRad: calculate radiological depth cube...done.
matRad: calculate lateral cutoff...done.
Progress: 100.00 %
```

Visual Comparison of results

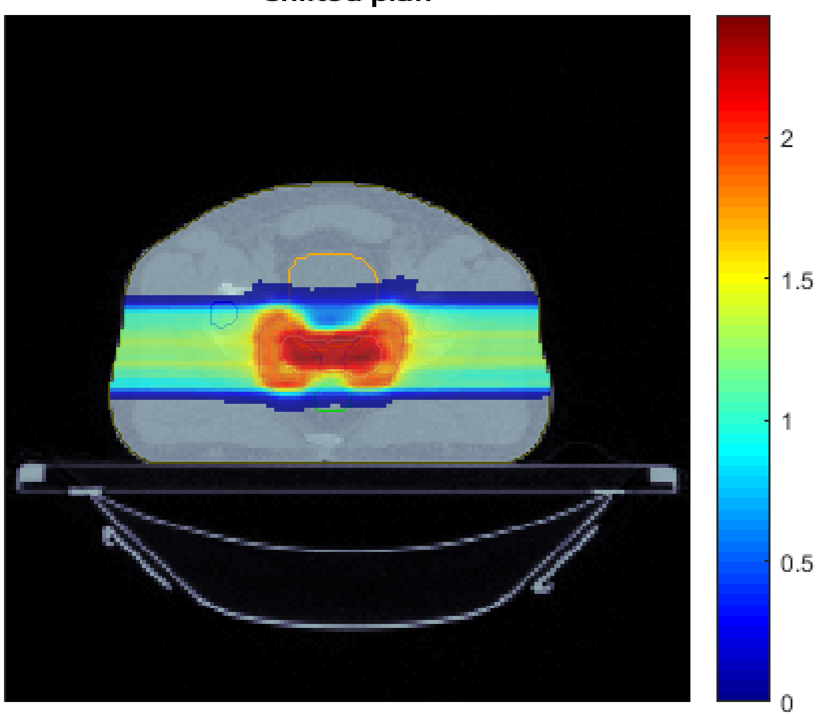
Let's compare the new recalculation against the optimization result.

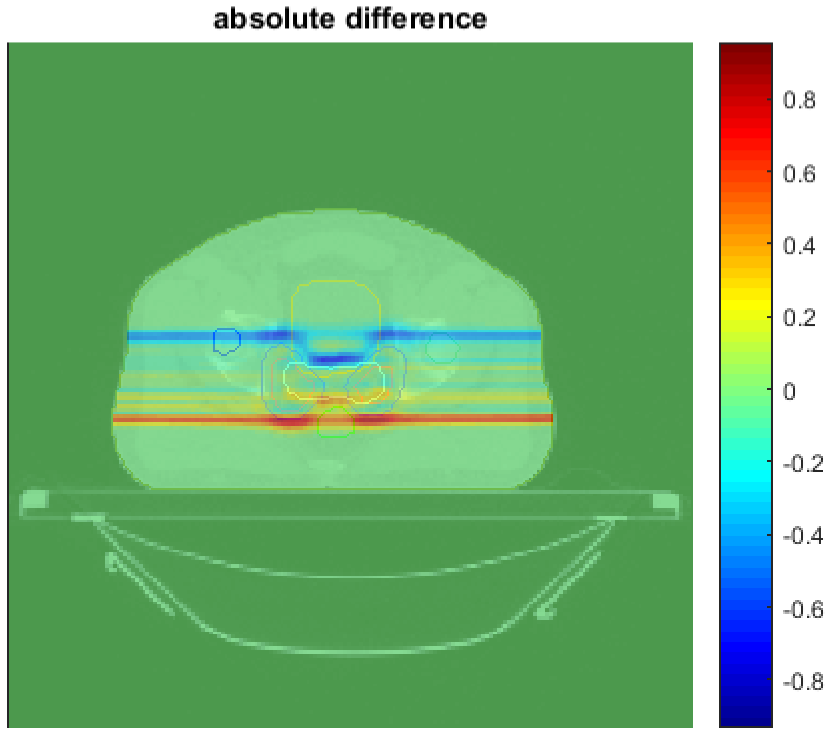
```
plane = 3;
doseWindow = [0 max([resultGUI.RBExDose(:);
 resultGUI_isoShift.RBExDose(:)])];

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.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')
```

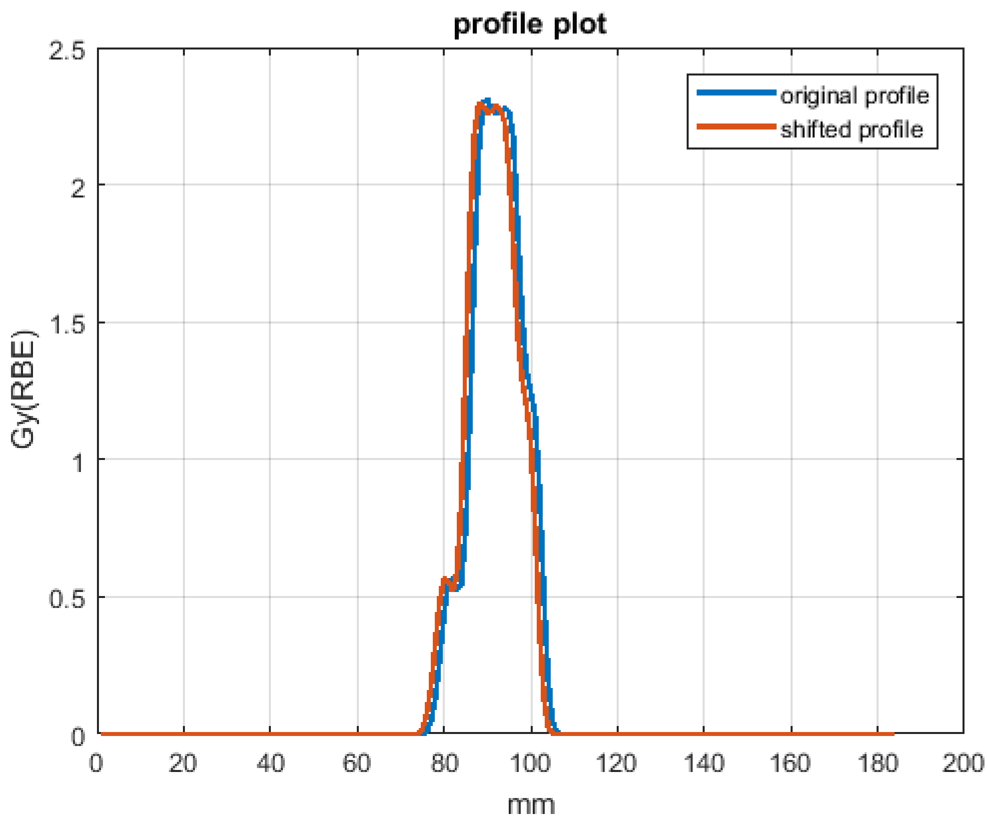
original plan



shifted plan




absolute difference
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8

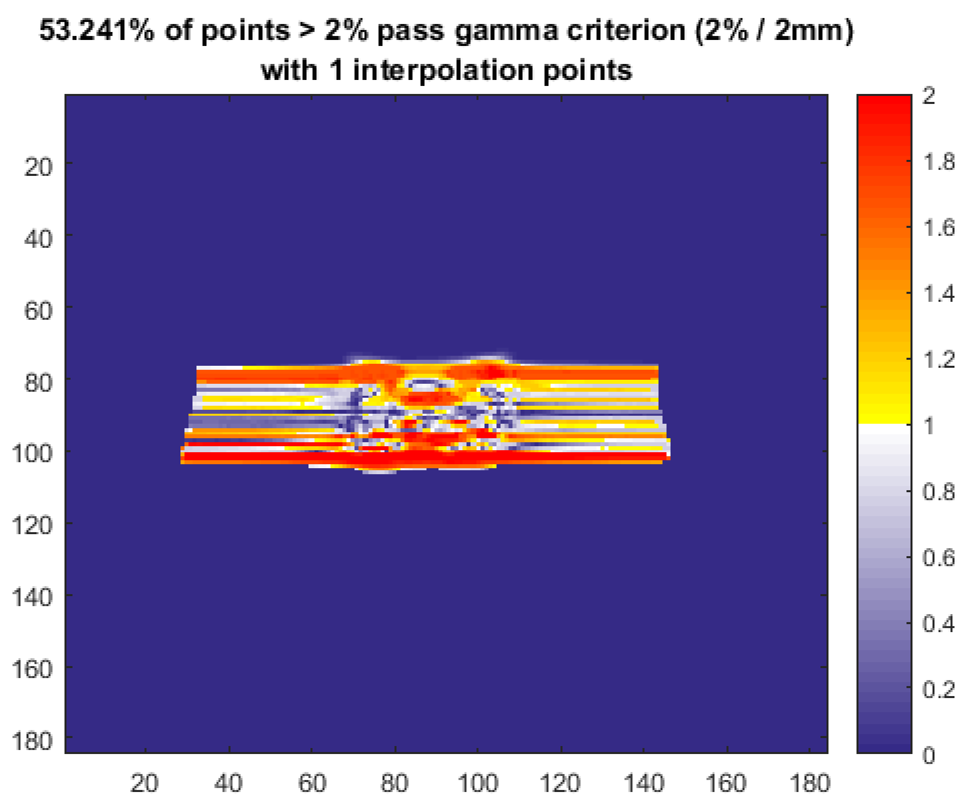

profile plot
original profile
shifted profile
Gy(RBE)
mm
2.5
2
1.5
1
0.5
0
0
20
40
60
80
100
120
140
160
180
200

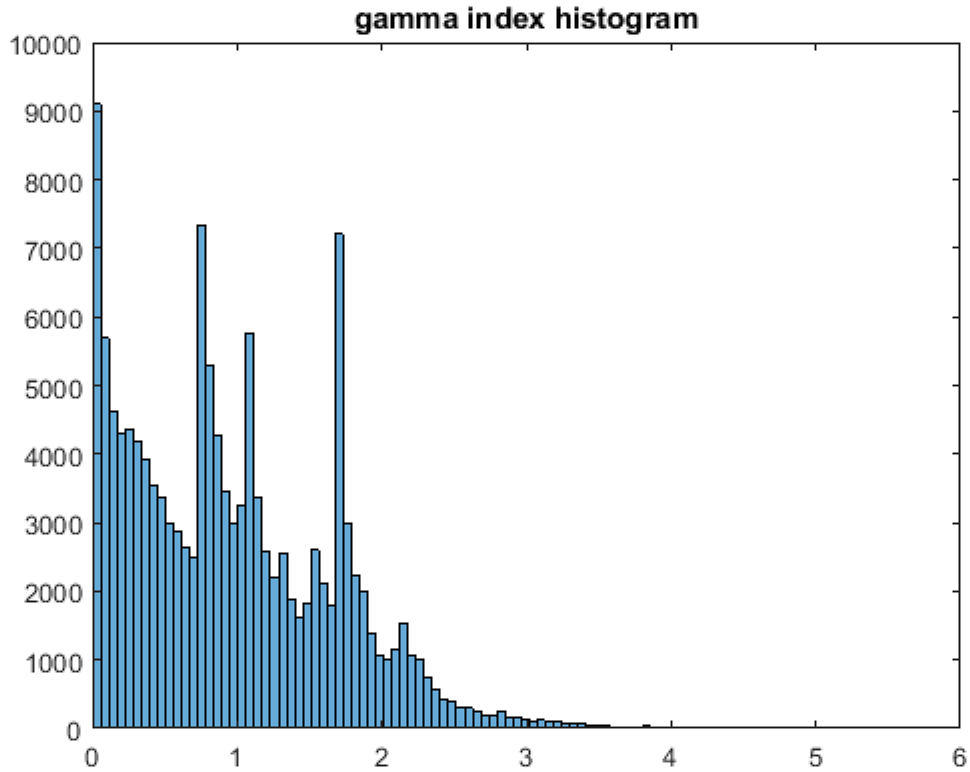
Quantitative Comparison of results

Compare the two dose cubes using a gamma-index analysis. The gamma index is a composite quality distribution equaly taking into account a dose difference and a distance to criteria in order to reveal differences betweent two dose cubes. A gamma-index value of smaller than 1 indicates a successfull 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],...
    slice,[doseDifference distToAgreement],n,'global',cst);


% Let's plot the gamma index histogram
figure,histogram(gammaCube(gammaCube>0),100),title('gamma index
 histogram')
```

.
.
.
.
.





Published with MATLAB® R2017a