

PRIEDAI

```
1 %PROJECT_PATH = ''; %Define Project path
2 SCAN_DATA_FOLDER = 'ScanData\';
3 PTV_NAME = 'RS.PTV.mat';
4 OD_NAME = 'RS.OD.mat';
5 SCAN_NAME = 'CT';
6
7 %Read data
8 if ~exist('excelDataLoaded', 'var') || excelDataLoaded == false
9     fprintf('BEGIN: read excel data \n');
10    [xStopPower, yStopPower] = readExcel('Data\StoppingPowerRatio4
        ', 1, 'B2:C12');
11    [xScatterPower, yScatterPower] = readExcel('Data\
        ScatteringPowerH2O ',1, 'A2:C34');
12
13    [xProfile1cm, yProfile1cm] = readExcel('Data\Profiliai ', 2, '
        A3:B52');
14    [xProfile2cm, yProfile2cm] = readExcel('Data\Profiliai ', 2, '
        E3:F53');
15
16    [xDepthDose, yDepthDose] = readExcel('Data\GylioDoze ', 1, 'A2:
        B37');
17
18    excelDataLoaded = true;
19    fprintf('END: read excel data \n');
20 else
21    fprintf('Excel data already loaded\n');
22 end
23
24 %CSAPI Cubic spline interpolant with not-a-knot end condition.
25 deapthDose = csapi(xDepthDose, yDepthDose);
26
```

```

27 length1Spline = csapi(xProfile1cm , yProfile1cm);
28 length2Spline = csapi(xProfile2cm , yProfile2cm);
29
30 %used for halfshadow length
31 length1cm = fnval(length1Spline , 0.1) - fnval(length1Spline , 0.9);
32 length2cm = fnval(length2Spline , 0.1) - fnval(length2Spline , 0.9);
33
34 stoppingPowerRatio = csapi(xStopPower, yStopPower);
35 scaterPowerH20 = csapi(xScatterPower , yScatterPower);
36 H2OScatterRatio = 1;
37
38 sliceNo=185;
39 fprintf('BEGIN: get scan images \n');
40 SCAN_FILE = strcat(SCAN_NAME, '. ', int2str(sliceNo) , '.dem');
41 [vezysFull , saugaFull , scanImageFull] = getScanImages(strcat(
    PROJECT_PATH,SCAN_DATA_FOLDER), PTV_NAME, OD_NAME, SCAN_FILE,
    sliceNo);
42 fprintf('END: get scan images \n');
43
44 fprintf('BEGIN: crop data \n');
45 [vezys , sauga , patientCT , xDomain, zDomain] = cropScanImages(
    vezysFull , saugaFull , scanImageFull );
46 fprintf('END: crop data \n');
47
48 fprintf('BEGIN: get CT Value \n');
49 [X, Z, vezysValue , saugaValue , CTValue] = getCTValue( xDomain,
    zDomain, vezys , sauga , patientCT );
50 fprintf('END: get CT Value \n');
51
52 [m, n] = size(X);
53 RatioValue = fnval(stoppingPowerRatio , CTValue);
54 z_effValue = z_effFunc( RatioValue , X, Z );
55

```

```

56 T_0 = 6;
57 R_p = 3.8;
58 sigmaValue = sigmaCalc( X, Z, z_effValue, RatioValue,
    scatterPowerH20, T_0, R_p );
59
60 center = 0;
61 tangentShadow = (length2cm-length1cm)/(2-1);
62 L0 = 1;
63 SSD = 100;
64 intSteps = 200;
65
66 %Integravimo rezis nurodomas funkcijoje
67 [probValue, sigma_medValue, integralValue] = ...
68     prob_sigmaMed_integralValues( sigmaValue, X, Z, L0,
        tangentShadow, center, intSteps);
69 %
70 [ValueDoseMeas, ValueZDose] = doseCalc( X, Z, L0, SSD,
    tangentShadow, depthDose, sigmaValue, intSteps);
71
72 doseMeasSpline = csapi(ValueZDose, ValueDoseMeas);
73 measuredAreaDose = doseValueCalc(z_effValue, integralValue,
    doseMeasSpline, SSD);
74
75 critGoal = 10;
76 cancerGoal = 80;
77 otherGoal = 30;
78
79 [critOld, cancerOld, score] = getScores( saugaValue, vezysValue,
    ...
    measuredAreaDose, cancerGoal, critGoal, otherGoal);
80
81
82 skinEdge = getSkinEdge(CTValue);
83

```

```

84 numberOfPoints = 12;
85 [bolusPoints, edgePoints] = getBolusPoints( X, Z, skinEdge,
        numberOfPoints);
86
87 GA_solution = GeneticOptimization( 10, X, Z, ratioValue,
        deapthDose, ...
88     H2OScatterRatio, scatterPowerH20, ...
89     skinEdge, edgePoints, A, numberOfPoints, dx, dz, L0, SSD, T_0,
        R_p, ...
90     tangentShadow, center, intSteps, ...
91     criticalValue, tumorValue, tumorGoal, criticalGoal, otherGoal)
        ;
92
93 SA_solution = SimulatedAnealing( X, Z, RatioValue, saugaValue,
        vezysValue, bolusPoints, numberOfPoints, edgePoints, ...
94     scaterPowerH20, tangentShadow, deapthDose, score, dx, dz,
        center, intSteps, ...
95     L0, T_0, R_p, SSD, ...
96     cancerGoal, critGoal, otherGoal);

1 function [ vezys, sauga, CTImage ] = getScanImages( path, PTVName,
        ODName, scanName, sliceNo )
2     PTV = load(strcat(path,PTVName));
3     vezys = double(rot90(PTV.contours.Segmentation(:, :, sliceNo),
        3));
4
5     OD = load(strcat(path,ODName));
6     sauga = double(rot90(OD.contours.Segmentation(:, :, sliceNo), 3)
        );
7
8     CTImageFresh = dicomread(strcat(path,scanName));
9     CTInfo = dicominfo(strcat(path,scanName));
10
11     CTImage = double(CTImageFresh)*CTInfo.RescaleSlope+CTInfo.

```

```

        RescaleIntercept;
12 end

1 function [ vezysCrop, saugaCrop, scanImageCrop, xDomain, zDomain ]
    = cropScanImages( vezysFull, saugaFull, scanImageFull )
2     centerZ=128;
3     centerX=286;
4     cropSizeZ=30;
5     cropSizeX=37;
6
7     pixelSize=0.125;
8
9     vezysCrop = vezysFull(centerZ - cropSizeZ : centerZ +
        cropSizeZ, centerX - cropSizeX : centerX + cropSizeX);
10    saugaCrop = saugaFull(centerZ - cropSizeZ : centerZ +
        cropSizeZ, centerX - cropSizeX : centerX + cropSizeX);
11    scanImageCrop = scanImageFull(centerZ - cropSizeZ : centerZ +
        cropSizeZ, centerX - cropSizeX : centerX + cropSizeX);
12
13    cropWidth = cropSizeX*pixelSize;
14    cropHeight = cropSizeZ*pixelSize;
15
16    X = -cropWidth:pixelSize:cropWidth;
17    Z = 0:-pixelSize:-2 * cropHeight;
18
19    [xDomain, zDomain] = meshgrid (X, Z);
20 end

1 function [ X, Z, ValueCancer, ValueSave, ValueCT ] = getCTValue(
    xDomain, zDomain, cancer, save, patientCT )
2
3     L0 = 1;
4     SSD = 100;
5
6     dx = 0.125;

```

```

7     dz = 0.125;
8
9     %Half of horizontal area length
10    A = (size(xDomain, 2) - 1) * dx / 2.;
11    %Full area depth
12    B = (size(xDomain, 2) - 1) * dz;
13
14    X = zeros(size(-A:dx:A, 2));
15    Z = zeros(size(0:dz:B, 2));
16
17    i = 0;
18    for step_x = -A:dx:A
19        i = i + 1;
20        j = 1;
21        for step_z = 0:dz:B
22            X(i, j) = step_x*step_z/(SSD - L0) + step_x;
23            Z(i, j) = L0 - step_z;
24            j = j + 1;
25        end
26    end
27
28    ValueCT = griddata(xDomain(1, :), zDomain(:, 1), patientCT, X,
29        Z);
30
31    ValueCT(isnan(ValueCT)) = -1000;
32    %
33    ValueSave = griddata(xDomain(1, :), zDomain(:, 1), save, X, Z)
34        ;
35    ValueSave(isnan(ValueSave)) = 0;
36    ValueSave(ValueSave <= 0.9) = 0;
37    ValueSave(ValueSave > 0.9) = 1;
38    %
39    ValueCancer = griddata(xDomain(1, :), zDomain(:, 1), cancer, X
40        , Z);

```

```

37     ValueCancer(isnan(ValueCancer)) = 0;
38     ValueCancer(ValueCancer <= 0.9) = 0;
39     ValueCancer(ValueCancer > 0.9) = 2;
40
41 end

1 function [ z_effValue ] = z_effFunc( RatioValue , X, Z )
2     [m, n] = size(RatioValue);
3     z_effValue = zeros(m, n);
4
5     for i = 1:m
6         dist=((X(i,1)-X(i,2))^2+(Z(i,1)-Z(i,2))^2)^(0.5);
7         sum = 0;
8         for j = 1:n
9             if j>1
10                 sum = sum + RatioValue(i,j-1)*dist;
11             end
12             z_effValue(i,j) = sum + (dist*RatioValue(i,j))/2;
13         end
14     end
15 end

1 function [ sigmaValue ] = sigmaCalc( X, Z, z_effValue , RatioValue ,
    scaterPowerH20 , T_0, R_p )
2
3     [m, n] = size(RatioValue);
4     sigmaValue = zeros(m, n);
5
6     z_eff_temp(:, 2:n) = z_effValue(:, 1:n-1);
7     ratioValue_temp(:, 2:n) = RatioValue(:, 1:n-1);
8
9     energy = T_0 .* (1 - z_eff_temp ./ R_p);
10    energy(:, 1) = 0;
11    scatterPower = fnval(scaterPowerH20 , energy) .*
        ratioValue_temp;

```

```

12
13     for i = 1:m
14         dist = sqrt((X(i, 1)-X(i, 2))^2 + (Z(i, 1)-Z(i, 2))^2);
15         M0 = 0; M1 = 0; M2 = 0;
16         for j = 1:n
17             sigma = 0;
18             if j>1
19                 M2 = M2 - 2*M1 + M0 + dist*scatterPower(i, j);
20                 M1 = M1 - M0 - dist*scatterPower(i, j);
21                 M0 = M0 + dist*scatterPower(i, j);
22                 sigma = ( M2 + M1 + (M0/3))*dist*dist;
23             end
24             sigmaValue(i, j) = sigma;
25         end
26     end
27 end

1 function [ probValue, sigma_medValue, integralValue ] =
    prob_sigmaMed_integralValues( sigmaValue, X, Z, L0,
    tangentShadow, center, intSteps )
2
3     [m, n] = size(sigmaValue);
4     b = (size(X, 2) - 1) * 0.125 / 2.;
5     a = -b;
6
7     integralValue = zeros(m, n);
8
9     sigma_medValue = (abs(Z - L0) .* (tangentShadow * 0.391 *
    0.25)).^2 + sigmaValue;
10
11     probValue = exp(-(center - X).^2 ./ (2 .* sigma_medValue)) ./
    (2 * pi .* sigma_medValue).^0.5;
12     probValue(isnan(probValue)) = 0;
13     probValue(probValue > 1) = 1;

```

```

14
15     for i=1:m
16         for j=1:n
17             sum = 0;
18             for k = 1:intSteps-1
19                 val = a + (k*(b-a)/intSteps);
20                 sum = sum + gaussian(sigma_medValue(i, j), val, X(
                    i, j));
21             end
22
23             integralValue(i, j) = 1 / (2*pi*sigma_medValue(i, j))
                ^0.5*...
24             ((sum+(gaussian(sigma_medValue(i, j), a, X(i, j))
                +...
25             gaussian(sigma_medValue(i, j), b, X(i, j))))/2)*...
26             (b-a)/intSteps);
27             if isnan(integralValue(i, j))
28                 integralValue(i, j) = 0;
29             end
30             if integralValue(i, j) > 1
31                 integralValue(i, j) = 1;
32             end
33         end
34     end
35 end

1 function [ ValueDoseMeas, ValueZStep ] = doseCalc( X, Z, L_0, SSD,
    tangentShadow, doseMeasured, sigmaValue, intSteps )
2
3     [m, n] = size(X);
4     dz = 0.125;
5     b = (size(X, 2) - 1) * 0.125 / 2.;
6     a = -b;
7     midLengthPosition = m/2 + 0.5;

```

```

8     zeroDepthPosition = L_0/dz;
9
10    ValueZStep= zeros(1, n-zeroDepthPosition);
11    z_step = abs(Z(midLengthPosition, 1:n-zeroDepthPosition) - L_0
12               );
13
14    ValueDoseMeas = ((SSD + z_step) ./ SSD).^2 .* fnval(
15                   doseMeasured, z_step) ./ 100 - 0.05;
16
17
18    for j = 1:n-zeroDepthPosition
19        z = abs(Z(midLengthPosition, j) - L_0);
20        x = X(midLengthPosition, zeroDepthPosition+j);
21
22        sigmaM = (abs(Z(midLengthPosition, zeroDepthPosition+j) -
23                     L_0) *...
24                  tangentShadow * 0.391 * 0.25)^2 + sigmaValue(
25                      midLengthPosition, zeroDepthPosition+j);
26
27        sum = 0;
28        for k = 1:intSteps-1
29            value = a + (k * (b-a) / intSteps);
30            sum = sum + gaussian(sigmaM, value, x);
31        end
32
33        answer = 1/((2*pi*sigmaM)^0.5) *...
34                ((sum + (gaussian(sigmaM, a, x) + gaussian(sigmaM, b, x
35                    ))/2)*(b-a)/intSteps);
36
37        if isnan(answer)
38            answer = 1;
39        end
40
41        ValueZStep(1, j) = z;
42        ValueDoseMeas(1, j) = ValueDoseMeas(1, j) / answer;

```

```

36
37         if z > 3.5 || ValueDoseMeas(1, j) < 0
38             ValueDoseMeas(1, j) = 0;
39         end
40
41     end
42 end

1 function [ doseValue ] = doseValueCalc(z_effValue, integralValue
    , ...
2     doseMeasSpline, SSD )
3
4     doseValue = integralValue .* fnval(doseMeasSpline, z_effValue)
        .* (SSD ./ (SSD + z_effValue)).^2;
5     doseValue(doseValue < 0.1) = 0;
6     doseValue = doseValue .* 100;
7 end

1 function [ criticalScore, cancerScore, score ] = getScores(
    saugaValue, vezysValue, doseValue, cancerGoal, critGoal,
    otherGoal )
2
3     cancerScore=0;
4     criticalScore=0;
5     cancerNumber=0;
6     critNumber=0;
7     otherNumber = 0;
8     otherScore = 0;
9
10    [m, n] = size(doseValue);
11    for i = 1:m
12        for j = 1:n
13            flag = 0;
14            if saugaValue(i, j) > 0
15                critNumber = critNumber+1;

```

```

16         %criticalScore = criticalScore + ((doseValue(i,j)
           - critGoal)^2)^0.5;
17         criticalScore = criticalScore + doseValue(i,j);
18         flag = 1;
19     end
20     if vezysValue(i,j) > 1
21         cancerNumber = cancerNumber + 1;
22         %cancerScore = cancerScore + ((doseValue(i,j) -
           cancerGoal)^2)^0.5;
23         cancerScore = cancerScore + doseValue(i,j);
24         flag = 1;
25     end
26     if flag == 0
27         otherNumber = otherNumber + 1;
28         %otherScore = otherScore + ((doseValue(i,j) -
           otherGoal)^2)^0.5;
29         otherScore = otherScore + doseValue(i,j);
30     end
31 end
32 end
33
34 criticalScore = 1*criticalScore/critNumber;
35 cancerScore = -1*cancerScore/cancerNumber;
36 otherScore = 1*otherScore/otherNumber;
37
38 score = criticalScore + cancerScore + otherScore;
39 end

1 function [ edge ] = getSkinEdge( CTImage )
2
3     imSize = size(CTImage);
4     edgeMatrix = zeros(imSize);
5     for i = 1:imSize(1)
6         flag = 0;

```

```

7         j = 1;
8         while (j < imSize(2)) && (flag < 1)
9             %2 consecutive points required in case CT of rounding
               errors
10            if CTImage(i , j) > -248 && CTImage(i , j+1) > -248
11                edgeMatrix(i , j) = 1;
12                edgeMatrix(i , j + 1) = 1;
13                edgeMatrix(i , j + 2) = 1;
14                flag = 1;
15            end
16            j=j+1;
17        end
18    end
19    edge=edgeMatrix;
20 end

1 function [ BolusPoints , edgePoints ] = getBolusPoints( X, Z, edge ,
    numberOfPoints)
2     [m, n] = size(X);
3     func1 = csapi(X(:,1) , 1:m);
4     %%
5     A = 3;
6     bolusPoint = [0 , 0];
7     bolusPoints = zeros(numberOfPoints , 2);
8
9     distanceBetweenPoints= 2*A/numberOfPoints;
10
11    for i = 0:numberOfPoints
12        bolusPoint(1) = -A+distanceBetweenPoints*i;
13        eile = fnval(func1 , bolusPoint(1));
14        bolusPoint(2) = Z(eile , fixEdge(edge(eile , :)));
15        bolusPoints(i+1,:) = bolusPoint;
16    end
17

```

```

18     BolusPoints = bolusPoints;
19     edgePoints = checkBolus(X, Z, edge);
20 end

1 function [m, b] = getBolusLines(numberOfPoints, Points)
2     bolusLineM = zeros(1, numberOfPoints);
3     bolusLineB = zeros(1, numberOfPoints);
4
5     for i=1:numberOfPoints
6         bolusLineM(i) = (Points(i+1, 2)-Points(i, 2))/(Points(i+1,
            1)-Points(i, 1));
7         bolusLineB(i) = Points(i, 2)-bolusLineM(i)*Points(i, 1);
8     end
9     m = bolusLineM;
10    b = bolusLineB;
11 end

1 function y = checkPointPosition(x, z, edgePoints,dx)
2     n = size(edgePoints, 1);
3     y = 0;
4     for i = 1:n
5         if(dx >= abs(x-edgePoints(i,1))) && z > edgePoints(i,2)
6             y = 1;
7             break;
8         end
9     end
10 end

1 function [ solution ] = SimulatedAnealing( X, Z, RatioValue,
    saugaValue, vezysValue, ...
2     bolusPoints, numberOfPoints, edgePoints, ...
3     scaterPowerH20, tangentShadow, deapthDose, score, ...
4     dx, dz, center, intSteps, ...
5     L0, T_0, R_p, SSD, ...
6     cancerGoal, critGoal, otherGoal)

```

```

7
8     [m, n] = size(RatioValue);
9     H2OScatterRatio = 1;
10
11     maxSteps = 5;
12     currentStep = 1;
13     T0 = 1;
14     T = T0;
15     Tf = 1e-3;
16     alfa = 0.95;
17
18     guessBolusPoints = bolusPoints;
19
20     bestScore = score;
21     bestBolusPoints = bolusPoints;
22     %bestRatioValue = RatioValue;
23
24     tic
25     while (T > Tf) && (currentStep <= maxSteps)
26         fprintf('Iteration no: %i \n', currentStep);
27         heightToAdd = round(rand*1) +1;
28         basePoint = 6 + round(rand*7);
29         check = guessBolusPoints(basePoint , 2) + heightToAdd*dz;
30
31         if (check >= L0)
32             guessBolusPoints(basePoint , 2) = L0;
33         else
34             guessBolusPoints(basePoint ,2) = bestBolusPoints(
35                 basePoint , 2) + heightToAdd*dz;
36         end
37
38     [bolusLineM , bolusLineB] = getBolusLines(numberOfPoints ,
39         guessBolusPoints);

```

```

38     guessRatioValue = RatioValue;
39
40     for i = 1:m
41         for j = 1:n
42             if checkPointPosition(X(i,j), Z(i,j), edgePoints,
43                                     dx) > 0
44                 flag = 0;
45                 k = 1;
46                 while k < numberOfPoints && flag == 0
47                     if guessBolusPoints(k, 1) <= X(i,j) &&
48                         guessBolusPoints(k+1,1) >= X(i,j)
49                         if ( bolusLineM(k) * X(i,j) +
50                             bolusLineB(k) - Z(i,j)) >=0
51                             guessRatioValue(i,j) =
52                                 H2OScatterRatio;
53                             flag = 1;
54                         end
55                     end
56                     k = k+1;
57                 end
58             end
59         end
60     end
61
62     z_effValue = z_effFunc(guessRatioValue, X, Z );
63     sigmaValue = sigmaCalc( X, Z, z_effValue, guessRatioValue,
64                             scatterPowerH20, T_0, R_p );
65     [probValue, sigma_medValue, integralValue] = ...
66         prob_sigmaMed_integralValues( sigmaValue, X, Z, L0,
67                                         tangentShadow, center, intSteps );
68
69     [ValueDoseMeas, ValueZDose] = doseCalc(X, Z, L0, SSD,
70                                             tangentShadow, deapthDose, sigmaValue, intSteps);

```

```

64     doseMeasSpline = csapi( ValueZDose , ValueDoseMeas );
65     doseValue = doseValueCalc( z_effValue , integralValue ,
        doseMeasSpline , SSD );
66
67     [critOld , cancerOld , guessScore] = getScores( saugaValue ,
        vezysValue , doseValue , cancerGoal , critGoal , otherGoal)
        ;
68
69     df = guessScore - bestScore;
70     if df < 0
71         bestScore = guessScore;
72         bestBolusPoints = guessBolusPoints;
73         %bestRatioValue = guessRatioValue;
74         bestDose = doseValue;
75     elseif (rand < probability(df, T) && df > 1e-2)
76         bestScore = guessScore;
77         bestBolusPoints = guessBolusPoints;
78     end
79     fprintf( 'best score: %f \n' , bestScore );
80     T=T0*alfa^currentStep;
81     currentStep = currentStep + 1;
82     fprintf( '-----\n' );
83
84     end
85     toc
86
87     solution = bestDose;
88 end

1 function [ candidate ] = GeneticOptimization( N, X, Z, ratioValue ,
    deapthDose , ...
2     H2OScatterRatio , scatterPowerH20 , ...
3     skinEdge , edgePoints , A, numPoints , dx, dz , L0, SSD, T_0, R_p,
    ...

```

```

4     tangentShadow, center, intSteps, ...
5     criticalValue, tumorValue, tumorGoal, criticalGoal, otherGoal)
6     tic
7
8     func1 = csapi(X(:,1), 1:size(X, 1));
9     edgeInterpol = csapi(edgePoints(:,1), edgePoints(:,2));
10
11     distanceBetweenPoints= 2*A/numPoints;
12
13     bolusArray(1:N) = GA_Bolus;
14     % generating initial bolus population
15     for i = 1:N
16         bolusPoint = [0, 0];
17         bolusPoints = zeros(numPoints, 2);
18         for j = 0:numPoints
19             bolusPoint(1) = -A + (distanceBetweenPoints*j);
20             eile = fnval(func1, bolusPoint(1));
21             bolusPoint(2) = Z(eile, fixEdge(skinEdge(eile, :)));
22
23             bolusPoint(2) = bolusPoint(2) + dz*(round((rand*(L0 -
24                 bolusPoint(2)))/dz)-1);
25             bolusPoints(j+1,:) = bolusPoint;
26         end
27         bolusArray(i).BolusPoints = bolusPoints;
28
29         bolusRatioValue = getBolusRatio(X, Z, H2OScatterRatio,
30             ratioValue, bolusArray(i).BolusPoints, ...
31             edgePoints, dx );
32
33         doseDistribution = getDoseDistribution(X, Z,
34             bolusRatioValue, scatterPowerH20, ...
35             deapthDose, L0, SSD, T_0, R_p, tangentShadow, center,
36             intSteps);

```

```

33
34
35     [critOld, cancerOld, bolusScore] = getScores(
        criticalValue, tumorValue, doseDistribution, ...
36         tumorGoal, criticalGoal, otherGoal);
37
38     bolusArray(i).Fitness = bolusScore;
39 end
40
41 for i = 1:N
42     %figure
43     %scatter(bolusArray(i).BolusPoints(:,1), bolusArray(i).
        BolusPoints(:,2))
44     fprintf('Fitness: %f\n', bolusArray(i).Fitness)
45 end
46
47     currentPopulation = bolusArray;
48     newPopulation(1:N) = GA_Bolus;
49     fitnessSort = zeros(1, N);
50 for i = 1:5
51     fprintf('Generation: %i\n', i)
52     if N > 3
53         %selecting
54         for j = 1:N
55             fitnessSort(j) = currentPopulation(j).Fitness;
56         end
57         fitnessSort = sort(fitnessSort);
58
59         for j = 1:round(N/2)
60             for k = 1:N
61                 if fitnessSort(j) == currentPopulation(k).
                    Fitness
62                     newPopulation(j) = currentPopulation(k);

```

```

63             break ;
64         end
65     end
66 end
67 %crossing
68 for j = 1:(round(N/2) - 1)
69     newPopulation(round(N/2)+j).BolusPoints = vertcat
70         (...
            newPopulation(j).BolusPoints(1:round(size(
                newPopulation(j).BolusPoints, 1)/2),:),
            ....
            newPopulation(j+1).BolusPoints(round(size(
                newPopulation(j+1).BolusPoints, 1)/2)+1:
                size(newPopulation(j+1).BolusPoints, 1),:))
            ;
71
72 %Mutate
73 for k = 1:numPoints
74     if rand < 0.1
75         newPopulation(round(N/2)+j).BolusPoints(k,
76             2) = ...
77             fnval(edgeInterpol, newPopulation(
78                 round(N/2)+j).BolusPoints(k, 1)) +
79                 ...
80                 rand*(L0 - fnval(edgeInterpol,
81                     newPopulation(round(N/2)+j).
82                     BolusPoints(k, 1)));
83     end
84 end
85
86 %Calculate fitness
87 bolusRatioValue = getBolusRatio(X, Z,
88     H2OScatterRatio, ratioValue, ...
89     newPopulation(round(N/2)+j).BolusPoints,

```

```

        edgePoints , dx);
84     doseDistribution = getDoseDistribution(X, Z,
        bolusRatioValue , scatterPowerH20 , ...
85     deapthDose , L0, SSD, T_0, R_p, tangentShadow ,
        center , intSteps);
86     [critOld , cancerOld , bolusScore] = getScores(
        criticalValue , tumorValue , doseDistribution ,
        ...
87     tumorGoal , criticalGoal , otherGoal);
88
89     newPopulation(round(N/2)+j).Fitness = bolusScore;
90     newPopulation(round(N/2)+j).DoseValue =
        doseDistribution;
91     newPopulation(round(N/2)+j).RatioValue =
        bolusRatioValue;
92     end
93
94     if rem(N, 2) == 0
95         newPopulation(N).BolusPoints = vertcat(...
96             currentPopulation(N).BolusPoints(1:round(size(
                currentPopulation(N).BolusPoints , 1)/2) ,:)) ,
                ....
97             currentPopulation(1).BolusPoints(round(size(
                currentPopulation(1).BolusPoints , 1)/2)+1:
                size(currentPopulation(1).BolusPoints , 1)
                ,:));
98
99     bolusRatioValue = getBolusRatio(X, Z,
        H2OScatterRatio , ratioValue , ...
100     newPopulation(N).BolusPoints , edgePoints , dx);
101     doseDistribution = getDoseDistribution(X, Z,
        bolusRatioValue , scatterPowerH20 , ...
102     deapthDose , L0, SSD, T_0, R_p, tangentShadow ,

```

```

        center , intSteps);
103         [critOld , cancerOld , bolusScore] = getScores(
            criticalValue , tumorValue , doseDistribution ,
            ...
104         tumorGoal , criticalGoal , otherGoal);
105         newPopulation(N).Fitness = bolusScore;
106         newPopulation(N).DoseValue = doseDistribution;
107         newPopulation(N).RatioValue = bolusRatioValue;
108     end
109     currentPopulation = newPopulation;
110
111     for j = 1:N
112         %figure
113         %scatter(currentPopulation(j).BolusPoints(:,1) ,
            currentPopulation(j).BolusPoints(:,2))
114         fprintf('Fitness: %f\n' , currentPopulation(j).
            Fitness)
115     end
116     else
117         fprintf('Need more samples')
118     end
119 end
120
121 for j = 1:N
122     fitnessSort(j) = currentPopulation(j).Fitness;
123 end
124 fitnessSort = sort(fitnessSort);
125 for k = 1:N
126     if fitnessSort(1) == currentPopulation(k).Fitness
127         bestCandidate = currentPopulation(k);
128         break;
129     end
130 end

```

```
131     toc
132     candidate = bestCandidate;
133 end

1 classdef GA_Bolus
2
3     properties
4         BolusPoints
5         RatioValue
6         Fitness
7         DoseValue
8     end
9
10    methods
11    end
12
13 end
```