MATH2390 Week 7 Hand in Lab

Table of Contents

Jason Thomas s3907634

Question 1

```
clear;

type Three_circles.m
```

```
function [c, ceq, gc, gceq] = Three_circles(rx)
    x = rx(4:6);
    y = rx(7:9);

    c(1) = (rx(1) + rx(2))^2 - (x(1) - x(2))^2 - (y(1) - y(2))^2;
    c(2) = (rx(1) + rx(3))^2 - (x(1) - x(3))^2 - (y(1) - y(3))^2;
    c(3) = (rx(2) + rx(3))^2 - (x(2) - x(3))^2 - (y(2) - y(3))^2;

    ceq = [ ];

    if nargout > 2

        gc(1:9,1) = [2*rx(1) + 2*rx(2);
                     2*rx(1) + 2*rx(2);
                     0;
                     -2*x(1) + 2*x(2);
                     -2*x(2) + 2*x(1);
                     0;
                     -2*y(1) + 2*y(2);
                     -2*y(2) + 2*y(1)
                     0];

        gc(1:9,2) = [2*rx(1) + 2*rx(3);
                     0;
                     2*rx(1) + 2*rx(3);
                     -2*x(1) + 2*x(3);
                     0;
```

```
                    -2*x(3) + 2*x(1);
                    -2*y(1) + 2*y(3);
                    0;
                    -2*y(3) + 2*y(1)];

        gc(1:9,3) = [0;
                    2*rx(2) + 2*rx(3);
                    2*rx(2) + 2*rx(3);
                    0;
                    -2*x(2) + 2*x(3);
                    -2*x(3) + 2*x(2);
                    0;
                    -2*y(2) + 2*y(3);
                    -2*y(3) + 2*y(2)];

        gceq = [];
    end
end
```

Question 2

```
type LinearConstraints.m
[A, b] = LinearConstraints();
```

```
function [A, b] = LinearConstraints()

    % A * rx <= b, for the upper bounds
    A_UB = [[eye(3); eye(3)], eye(6)];
    b_UB = [5;5;5;3;3;3];

    A_LB = [[eye(3); eye(3)], -eye(6)];
    b_LB = zeros(6,1);

    A = [A_UB; A_LB];
    b = [b_UB; b_LB];
end
```

Question 3

```
type Neg_Areas.m
```

```
function [value, grad]  = Neg_Areas(rx)
    switch nargout
        case 1
            % change sign to maximise
            value = -(rx(1) + rx(2) + rx(3));
        case 2
            value = -(rx(1) + rx(2) + rx(3));
            grad = [-1; -1; -1; 0; 0; 0; 0; 0; 0];
    end
end
```

Question 4

```
options = optimoptions('fmincon','Display','iter','GradObj','on',...
    'GradConstr','on','PlotFcns',{@optimplotx, @optimplotfval});
[rx, fval] = fmincon(@Neg_Areas, zeros(9,1), A, b,...
[], [], zeros(9),[inf; inf; inf; 5; 5; 5; 3; 3; 3],...
@Three_circles, options);
```

```
Warning: Length of lower bounds is > length(x); ignoring extra bounds.
Your initial point x0 is not between bounds lb and ub; FMINCON
shifted x0 to strictly satisfy the bounds.
```

Iter	F-count	f(x)	Feasibility	First-order optimality	Norm of step
0	1	-2.970000e+00	3.920e+00	1.261e-01	
1	2	-1.197007e+00	6.368e-01	1.814e-01	1.251e+00
2	3	-8.377784e-01	3.119e-01	3.665e-01	6.693e-01
3	4	-6.944139e-02	2.143e-03	7.723e-01	5.685e-01
4	6	-2.643893e-01	3.107e-02	5.755e-01	4.843e-01
5	8	-2.854639e-01	3.622e-02	2.166e-01	3.496e-01
6	10	-2.553685e-01	2.898e-02	9.004e-02	3.784e-01
7	11	-1.561887e-01	1.084e-02	9.704e-02	2.432e-01
8	12	-4.062923e-02	7.337e-04	1.073e+00	8.142e-01
9	14	-4.976430e-02	1.101e-03	5.221e-01	6.269e-03
10	16	-4.515835e-02	9.063e-04	2.939e-01	1.680e-02
11	18	-3.469794e-02	5.351e-04	3.535e-01	2.615e-02
12	19	-1.494049e-02	0.000e+00	5.933e+00	8.290e-02
13	20	-2.587037e-02	0.000e+00	2.964e+00	2.459e-02
14	21	-5.119465e-02	0.000e+00	1.807e+00	1.425e-01
15	22	-1.668202e-01	0.000e+00	8.580e-01	5.634e-01
16	23	-2.241510e-01	0.000e+00	8.244e-01	2.409e-01
17	24	-5.142488e-01	0.000e+00	1.075e+00	1.385e+00
18	25	-4.680245e-01	0.000e+00	1.027e+00	8.540e-01
19	26	-6.618632e-01	0.000e+00	9.881e-01	8.432e-01
20	27	-6.929768e-01	0.000e+00	9.144e-01	4.413e-01
21	28	-8.429688e-01	0.000e+00	9.132e-01	8.764e-01
22	29	-9.300596e-01	0.000e+00	9.126e-01	7.025e-01
23	30	-8.997934e-01	0.000e+00	9.132e-01	3.228e-01
24	31	-9.344756e-01	0.000e+00	8.949e-01	2.456e-01
25	32	-9.931126e-01	0.000e+00	8.951e-01	4.030e-01
26	33	-1.096989e+00	0.000e+00	8.949e-01	5.495e-01
27	34	-1.452557e+00	0.000e+00	8.940e-01	1.457e+00
28	35	-1.786681e+00	0.000e+00	8.932e-01	1.249e+00
29	36	-1.929504e+00	0.000e+00	8.063e-01	8.630e-01
30	37	-2.059108e+00	0.000e+00	6.832e-01	3.973e-01

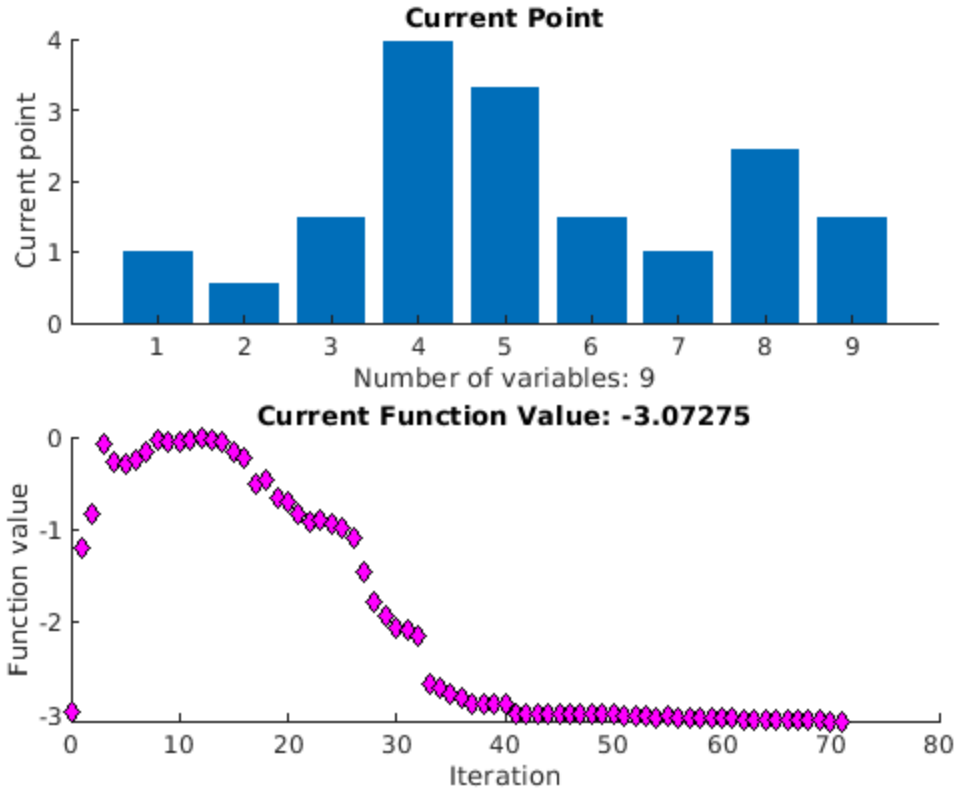
Iter	F-count	f(x)	Feasibility	First-order optimality	Norm of step
31	42	-2.094413e+00	0.000e+00	6.786e-01	3.927e-02
32	47	-2.142556e+00	0.000e+00	6.477e-01	4.775e-02
33	48	-2.675385e+00	0.000e+00	4.763e-01	5.852e-01
34	49	-2.726281e+00	0.000e+00	1.845e-01	4.985e-01
35	51	-2.773800e+00	0.000e+00	1.540e-01	1.466e-01

36	53	-2.817691e+00	0.000e+00	1.644e-01	1.259e-01
37	54	-2.878591e+00	0.000e+00	1.656e-01	2.824e-01
38	55	-2.890901e+00	0.000e+00	6.104e-02	2.202e-01
39	56	-2.884539e+00	0.000e+00	3.711e-02	1.402e-01
40	57	-2.882937e+00	0.000e+00	2.012e-02	5.710e-02
41	58	-2.986792e+00	0.000e+00	2.845e-02	9.336e-02
42	59	-2.990624e+00	0.000e+00	4.542e-02	1.565e-01
43	60	-2.999385e+00	0.000e+00	7.502e-02	1.991e-01
44	61	-3.002373e+00	0.000e+00	7.262e-02	4.052e-02
45	62	-3.002620e+00	0.000e+00	7.095e-02	5.620e-03
46	63	-3.002812e+00	0.000e+00	7.063e-02	3.819e-03
47	64	-3.003027e+00	0.000e+00	7.029e-02	3.766e-03
48	65	-3.003357e+00	0.000e+00	6.974e-02	4.485e-03
49	66	-3.004092e+00	0.000e+00	6.844e-02	6.403e-03
50	67	-3.006202e+00	0.000e+00	6.391e-02	1.378e-02
51	68	-3.010879e+00	0.000e+00	4.707e-02	3.940e-02
52	69	-3.014254e+00	0.000e+00	2.937e-02	1.102e-01
53	70	-3.019515e+00	0.000e+00	2.427e-02	2.984e-01
54	71	-3.034162e+00	0.000e+00	2.340e-02	6.438e-01
55	72	-3.028932e+00	0.000e+00	1.263e-02	2.698e-01
56	73	-3.033070e+00	0.000e+00	2.047e-02	1.182e-01
57	74	-3.034932e+00	0.000e+00	2.533e-02	5.743e-02
58	75	-3.035197e+00	0.000e+00	2.622e-02	5.933e-03
59	76	-3.035687e+00	0.000e+00	2.768e-02	6.383e-03
60	77	-3.036424e+00	0.000e+00	1.902e-02	9.237e-03

Iter	F-count	f(x)	Feasibility	First-order optimality	Norm of step
61	78	-3.036591e+00	0.000e+00	4.000e-03	4.678e-03
62	79	-3.064729e+00	0.000e+00	1.174e-02	6.496e-02
63	80	-3.065651e+00	0.000e+00	8.529e-03	2.227e-02
64	81	-3.065539e+00	0.000e+00	8.047e-04	1.731e-03
65	82	-3.071279e+00	0.000e+00	1.534e-03	1.469e-02
66	83	-3.071309e+00	0.000e+00	1.521e-03	1.960e-04
67	84	-3.071309e+00	0.000e+00	1.600e-04	1.466e-05
68	85	-3.072460e+00	0.000e+00	3.393e-04	2.862e-03
69	86	-3.072461e+00	0.000e+00	3.200e-05	5.207e-06
70	87	-3.072746e+00	0.000e+00	8.803e-05	7.055e-04
71	88	-3.072746e+00	0.000e+00	4.456e-07	2.968e-07

Local minimum found that satisfies the constraints.

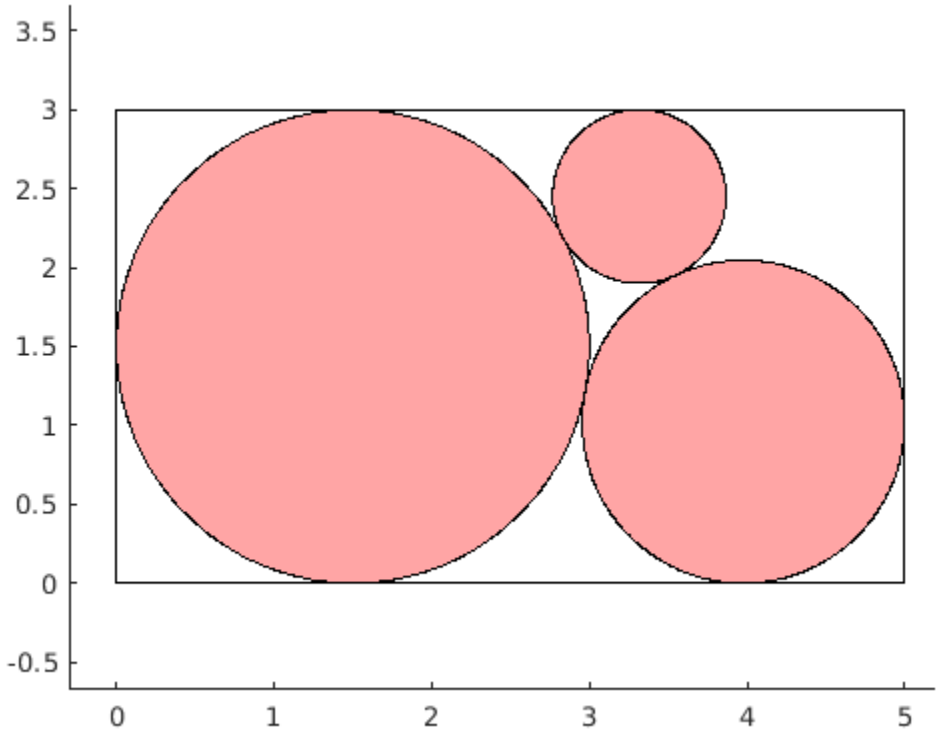
Optimization completed because the objective function is non-decreasing in feasible directions, to within the value of the optimality tolerance, and constraints are satisfied to within the value of the constraint tolerance.



Question 5

```
rectangleDim = [5, 3];

type maximalArea.m
maximalArea(rx, rectangleDim);

type checkGeometry.m
checkGeometry(rx(1:3), rx(4:6), rx(7:9), rectangleDim);

% To me, this looks successful


function maximalArea(rx, rectangleDim)
    circleArea = pi*rx(1)^2 + pi*rx(2)^2 + pi*rx(3)^2;
    rectangleArea = rectangleDim(1)*rectangleDim(2);

    fprintf("Circles have area: %f\n", circleArea);
    fprintf("Circles fill %f of rectangle\n", circleArea/rectangleArea);
end
Circles have area: 11.305119
Circles fill 0.753675 of rectangle

function checkGeometry(r, x, y, rectangleDim)
    % you should be able to look and see if the optimal solution makes sense
```

```
    polySides = 1000; % close enough to a circle

    figure

    rectangle('Position', [0, 0, rectangleDim]);

    hold on;
    axis equal;

    circle1 = nsidedpoly(polySides, 'Center', [x(1), y(1)], 'Radius', r(1));
    plot(circle1, 'FaceColor', 'r');
    circle2 = nsidedpoly(polySides, 'Center', [x(2), y(2)], 'Radius', r(2));
    plot(circle2, 'FaceColor', 'r');
    circle3 = nsidedpoly(polySides, 'Center', [x(3), y(3)], 'Radius', r(3));
    plot(circle3, 'FaceColor', 'r');

    hold off;
end
```



Published with MATLAB® R2023a