

**P1.****(A)**

Let  $\max(2x_1 + 3x_2 - 5x_3, x_1, x_2, 2) \leq t$ . We can then form the following linear program:

$$\begin{array}{llllll} \min & t & & & & \\ \text{s.t.} & +2x_1 & +3x_2 & -5x_3 & \leq & t \\ & +2x_1 & -x_2 & +x_3 & \leq & t \\ & x_1, & x_2 & & \leq & t \end{array}$$

**(B)**

Let  $t_1, t_2, t_3, t_4 \geq 0$ .

Let  $|x_1 + x_2| \leq t_1, |x_2 - x_3| \leq t_2, |x_3 - x_1| \leq t_3, |x_1 + x_2 + x_3| \leq t_4$ .

You now have the linear problem

$$\begin{array}{llllll} \min & +t_1 & +t_2 & +t_3 & +t_4 & \\ \text{s.t.} & +x_1 & +x_2 & & & \leq t_1 \\ & -x_1 & -x_2 & & & \leq t_1 \\ & & +x_2 & -x_3 & & \leq t_2 \\ & & -x_2 & +x_3 & & \leq t_2 \\ & -x_1 & & +x_3 & & \leq t_3 \\ & +x_1 & & -x_3 & & \leq t_3 \\ & +x_1 & +x_2 & +x_3 & & \leq t_4 \\ & -x_1 & -x_2 & -x_3 & & \leq t_4 \\ & t_1, & t_2 & t_3 & t_4 & \geq 0 \end{array}$$

**(C)**

Let  $\max(|x_1|, |x_2|, |x_3|, |x_1 + x_2|) \leq t$ .

You now have the linear problem

$$\begin{array}{llll} \min & t & & \\ \text{s.t.} & +x_1 & -x_2 & \leq 5 \\ & & +x_2 & \leq 3 \\ & +x_1 & & \leq t \\ & -x_1 & & \leq t \\ & & +x_2 & \leq t \\ & & -x_2 & \leq t \\ & & +x_3 & \leq t \\ & & -x_3 & \leq t \\ & +x_1 & +x_2 & \leq t \\ & -x_1 & -x_2 & \leq t \\ & t & & \geq 0 \end{array}$$

## P2.

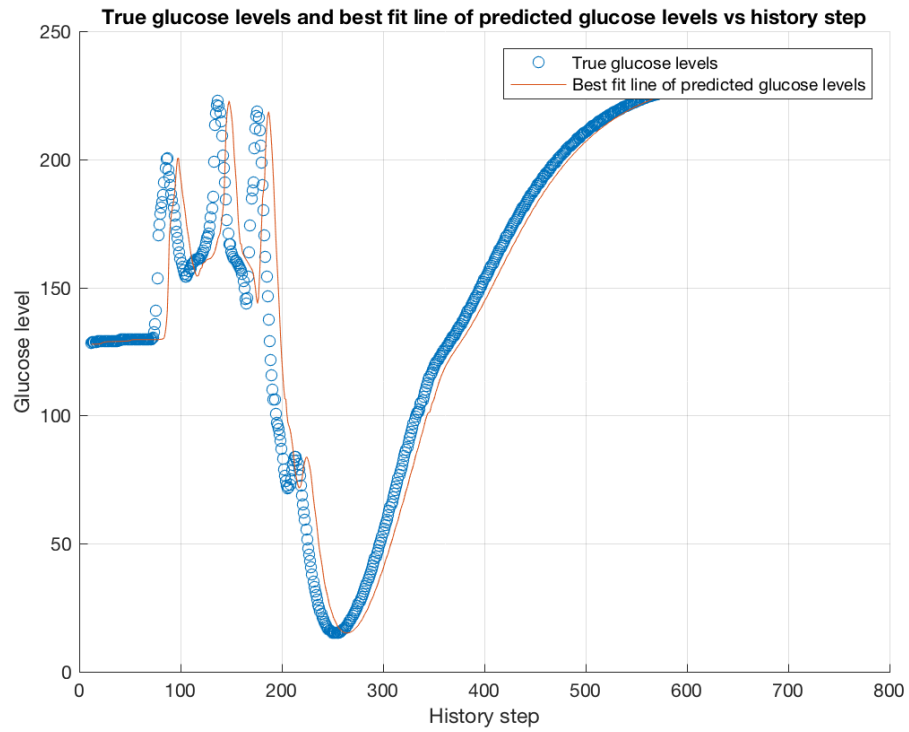
```
1 % Code for problem 2 of homework 4.
2 % Read in the csv file skipping the first row
3 M = csvread('insulinGlucose1.csv', 1);
4
5 % Insulin input values
6 u = M(:,1);
7 % Glucose levels
8 G = M(:,2);
9
10 A = [];
11 b = [];
12
13 [m,~] = size(G(12:end));
14 for i = 1:m
15     b = [b, G(i)];
16     b = [b, -G(i)];
17 end
18
19 b = b';
20
21 % Initialize the objective function which is the minimum of the sum
22 % of the absolute values of the residuals.
23 % Residuals have the form  $y - Ax$ .
24 f = ones(1,17);
25
26 % Create the constraint matrix A which is the residuals
27 % of glucose levels 12 -> 707.
28 % If  $u = |x|$  then  $u = x$  and  $u = -x$ 
29 [m,~] = size(G);
30 for t = 11:(m-1)
31     a = [G(t) G(t-1) G(t-2) G(t-3) G(t-4) G(t-5) G(t-6) G(t-7) G(t-8) G(t-9) G(t-10)];
32     a = horzcat(a, [u(t) u(t-1) u(t-2) u(t-3) u(t-4) u(t-5)]);
33     % Append the constraint ax.
34     A = [A;a];
35
36     % Append the constraint -ax.
37     a = times(a, -1);
38     A = [A;a];
39 end
40
41 % Solve for the coefficients.
42 % x will have an array of the coefficient values when it is solved.
43 [x,fval] = linprog(f,A,b)
44
45 % Recreate the matrix A without the additional absolute value constraints.
46 A = [];
47 for t = 11:(m-1)
48     a = [G(t) G(t-1) G(t-2) G(t-3) G(t-4) G(t-5) G(t-6) G(t-7) G(t-8) G(t-9) G(t-10)];
49     a = horzcat(a, [u(t) u(t-1) u(t-2) u(t-3) u(t-4) u(t-5)]);
50     A = [A;a];
51 end
52
53 % Create an array of the history steps.
54 t = [];
55 for i = 12:m
56     t = [t, i];
57 end
58
59 % Plot the true glucose levels at each history step t_i.
60 figure(1);
61 scatter(t,G(12:end));
62 hold on;
63
64 % Plot the best fit line of the predicted glucose levels at each
65 % history step t_i.
66 % x stores the estimated coefficients of the best fit line.
67 %  $|x| = \sqrt{x^2}$ 
68 A = (A*x).^2;
69 A = sqrt(A);
70 plot(t, A);
71 grid on;
72 legend('True glucose levels', 'Best fit line of predicted glucose levels');
73 title('True glucose levels and best fit line of predicted glucose levels vs history step');
```

```

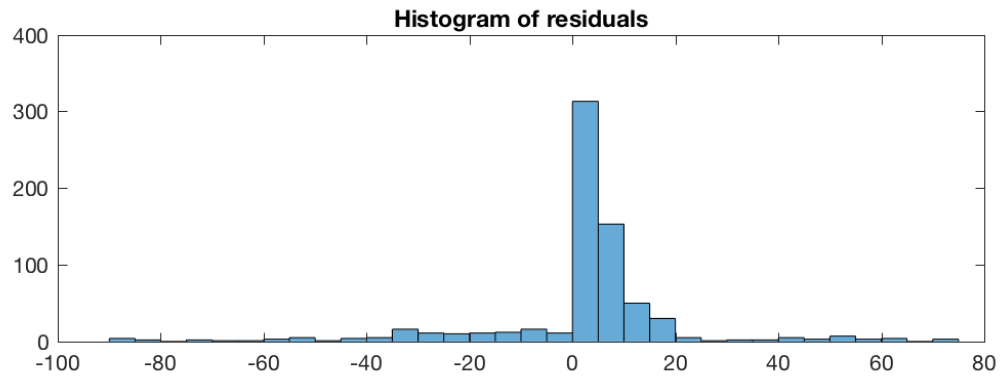
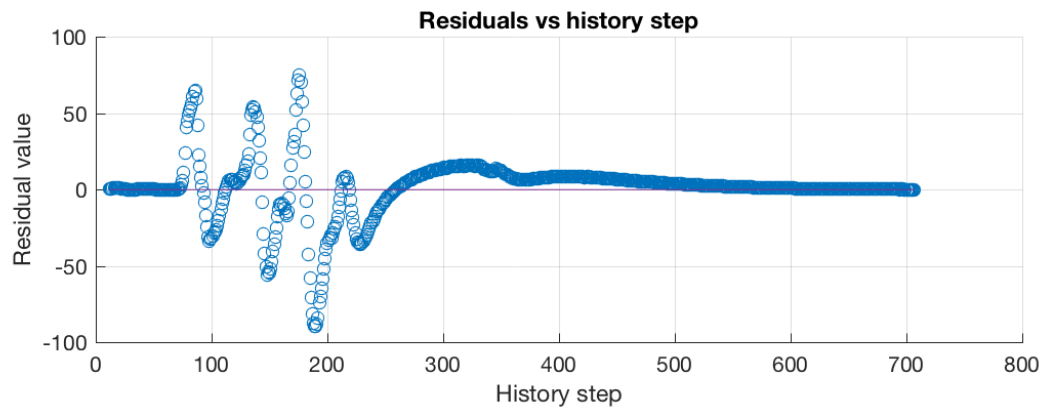
74 xlabel('History step');
75 ylabel('Glucose level');
76 saveas(1, 'predictions.png');
77
78 % Plot the residuals.
79 figure(2);
80 residuals = [];
81 trueGlucoseValues = G(12:end);
82 predictedGlucoseValues = A;
83 [m,~] = size(predictedGlucoseValues);
84 for i = 1:m
85     residuals = [residuals, trueGlucoseValues(i) - predictedGlucoseValues(i)];
86 end
87
88 subplot(2,1,1);
89 scatter(t, residuals);
90 hold on;
91 plot(t, zeros(m));
92 grid on;
93 title('Residuals vs history step');
94 xlabel('History step');
95 ylabel('Residual value');
96
97 subplot(2,1,2);
98 histogram(residuals);
99 title('Histogram of residuals');
100 saveas(2, 'residuals.png');

```

Coefficient	Value
$a_0$	8.704148513061227e-14
$a_1$	-3.375077994860476e-14
$a_2$	-1.474376176702208e-13
$a_3$	2.131628207280301e-13
$a_4$	-1.243449787580175e-13
$a_5$	-1.056932319443149e-13
$a_6$	1.376676550535194e-13
$a_7$	-3.979039320256561e-13
$a_8$	3.179678742526448e-13
$a_9$	6.430411758628907e-13
$a_{10}$	0.999999999999309
$b_0$	1.989519660128281e-13
$b_1$	-2.060573933704291e-13
$b_2$	1.847411112976261e-13
$b_3$	-8.526512829121202e-14
$b_4$	-1.207922650792170e-13
$b_5$	7.815970093361102e-14



(a) Linear Regression Predictions



(b) Residuals

**P3.**

**(A)**

1.

$x_1$	+2	$-x_4$	$+x_5$	$-x_6$			$+w_6$
$x_2$	-8		$+2x_5$	$+x_6$	$+w_4$	$+w_5$	$+w_6$
$x_3$	+4	$+x_4$	$-2x_5$			$-w_5$	$-w_6$
$w_1$	-7	$+x_4$	$+x_5$	$+3x_6$	$+w_4$	$+w_5$	
$w_2$	-3	$+2x_4$		$+x_6$			$-w_6$
$w_3$	2	$+x_4$	$-2x_5$	$+x_6$		$-w_5$	$-w_6$
$\zeta$	16		$-6x_5$		$-3w_4$	$-2w_5$	$-5w_6$

2.

$x_1$	-1	$-x_3$	$+x_4$	$+2x_6$	$-w_1$		$+w_4$
$x_2$	-4	$-x_3$	$+x_4$	$+x_6$			$+w_4$
$x_5$	0	$-x_3$		$+2x_6$	$-w_1$	$+w_2$	$+w_4$
$w_3$	-2	$+x_3$		$+x_6$			
$w_5$	7	$+x_3$	$-x_4$	$-5x_6$	$+2w_1$	$-w_2$	$-2w_4$
$w_6$	-3		$2x_4$	$+x_6$		$-w_2$	
$\zeta$	14	$+4x_3$	$-6x_4$	$-6x_6$	$+2w_1$		$-5w_4$

3.

$x_1$	-1		$+x_4$	$+x_5$		$-w_2$	
$x_2$	-6	$+x_3$	$+x_4$	$+x_5$	$+w_1$	$-w_2$	$-w_3$
$x_6$	2	$-x_3$					$+w_3$
$w_4$	-4	$+3x_3$		$+x_5$	$+w_1$	$-w_2$	$-2w_3$
$w_5$	5		$-x_4$	$-2x_5$		$+w_2$	$-w_3$
$w_6$	-1	$-x_3$	$+2x_4$			$-w_2$	$+w_3$
$\zeta$	22	$-5x_3$	$-6x_4$	$-5x_5$	$-3w_1$	$+5w_2$	$+4w_3$

**(B)**

1.

$x_3$	2			$-x_6$	$+w_3$		
$x_4$	6		$+x_2$	$-2x_6$	$+w_3$	$-w_4$	
$x_5$	4	$-x_1$			$-w_3$		$-w_5$
$w_1$	3	$-x_1$	$+x_2$	$+x_6$			
$w_2$	9	$-2x_1$	$+x_2$	$-2x_6$		$-w_4$	$-w_5$
$w_6$	0	$+2x_1$	$+x_2$	$-x_6$	$+2w_3$	$-w_4$	$+w_5$
$\zeta$	-8	$-2x_1$	$-4x_2$	$+4x_6$	$-2w_3$	$+w_4$	

2.  $x_6$  will be the entering variable because it has the largest coefficient.

3. The equation for the entering variable column  $a_i$  is  $a_i = -B^{-1}Ne_i$ .

4. The leaving variable will be  $w_6$  because it constrains the entering variable the most.

5. The basic variables will be  $x_3, x_4, x_5, w_1, w_2, x_6$ .

**(C)**

$x_5$	1	$+x_1$			$-x_4$	$+w_2$	
$x_6$	4		$+x_2$	$+x_3$	$-x_4$		$-w_4$
$w_1$	7	$-x_1$	$+2x_2$	$+x_3$	$-x_4$		$-w_4$
$w_3$	2		$+x_2$	$+2x_3$	$-x_4$		$-w_4$
$w_5$	1	$-2x_1$	$-x_2$	$-2x_3$	$+2x_4$	$-w_2$	$+w_4$
$w_6$	1		$+x_2$	$+x_3$	$+x_4$	$-w_2$	$-w_4$
$\zeta$	4	$-2x_1$	$-2x_2$		$-2x_4$		$-w_4$

**(D)**

This produces a new objective function:  $(-2+u)x_1 + (-3+u)x_2 - x_3 - x_4 + (1+u)x_6$ . To make sure the dictionary is final and feasible we have to ensure that the objective row coefficients are  $\leq 0$  and the constant column values are  $\geq 0$ . Trying different values of  $\mu$  I found a bound of  $-1 \leq \mu \leq 0$  as values of the  $\mu$  that kept the dictionary final and feasible.

**(E)**

We need to ensure that the constant column of the dictionary is  $\geq 0$  to ensure the dictionary is feasible. Trying different values of  $\lambda$  I found a bound of  $-1 \leq \lambda \leq +1$ .

```

1  % Code for problem 3 of homework 4.
2  A = [+1 -1 0 0 0 -1 +1 0 0 0 0 0;...
3      +1 0 0 -1 -1 0 0 +1 0 0 0 0;...
4      0 0 -1 0 0 -1 0 0 +1 0 0 0;...
5      0 -1 -1 +1 0 +1 0 0 0 +1 0 0;...
6      +1 0 +1 0 +1 +1 0 0 0 0 +1 0;...
7      -1 0 0 -1 +1 -1 0 0 0 0 0 +1];
8
9  b = [3 -1 -2 +4 +6 -2]';
10
11 c = [-2 -3 -1 -1 0 +1 0 0 0 0 0 0];
12
13 % P3 a1
14 xB = [1 2 3 7 8 9];
15 xN = [4 5 6 10 11 12];
16 B = A(:, xB);
17 N = A(:, xN);
18 Binv = inv(B);
19 % Dictionary constant value column.
20 Binv*b;
21 % Dictionary coefficient value rows.
22 -Binv*N;
23
24 cB = c(:, xB);
25 cN = c(:, xN);
26 % Objective function constant value.
27 cB*(Binv*b);
28 % Object function coefficient values.
29 cN - cB*(Binv*N);
30
31 % P3 a2
32 xB = [5 6 7 8 9 10];
33 xN = [1 2 3 4 11 12];
34
35
36 B = A(:, xB);
37 N = A(:, xN);
38 Binv = inv(B);
39 % Dictionary constant value column.
40 Binv*b;
41 % Dictionary coefficient value rows.
42 -Binv*N;
43
44 cB = c(:, xB);
45 cN = c(:, xN);
46 % Objective function constant value.
47 cB*(Binv*b);
48 % Object function coefficient values.
49 cN - cB*(Binv*N);
50
51 % P3 a3
52 xB = [1 2 6 10 11 12];
53 xN = [3 4 5 7 8 9];
54
55 B = A(:, xB);
56 N = A(:, xN);
57 Binv = inv(B);
58 % Dictionary constant value column.
59 Binv*b;
60 % Dictionary coefficient value rows.
61 -Binv*N;
62
63 cB = c(:, xB);
64 cN = c(:, xN);
65 % Objective function constant value.
66 cB*(Binv*b);
67 % Object function coefficient values.
68 cN - cB*(Binv*N);
69
70 % p3 b
71 xB = [3 4 5 7 8 12];
72 xN = [1 2 6 9 10 11];
73
74 B = A(:, xB);
75 N = A(:, xN);

```

```

76 Binv = inv(B);
77 % Dictionary constant value column.
78 Binv*b;
79 % Dictionary coefficient value rows.
80 -Binv*N;
81
82 cB = c(:, xB);
83 cN = c(:, xN);
84 % Objective function constant value.
85 cB*(Binv*b);
86 % Object function coefficient values.
87 cN - cB*(Binv*N);
88
89 % Choose the third variable as the entering variable.
90 % Get the column of the entering variable.
91 a_i = -Binv*N(:, 3);
92
93 % Determine the leaving variable row index by finding the value that
94 % constraints the entering variable the most.
95 -(Binv*b)./a_i;
96
97 % Get the leaving variable row.
98 a_j = -(Binv*N);
99 a_j = a_j(6,:);
100
101 % P3 c
102 f = [-2 -3 -1 -1 0 1];
103 Y = [1 -1 0 0 0 -1; 1 0 0 -1 -1 0; 0 0 -1 0 0 -1; 0 -1 -1 1 0 1; 1 0 1 0 1 1; -1 0 0 -1 1 -1];
104 x = [3 -1 -2 4 6 -2];
105 options = optimoptions('linprog','Algorithm','dual-simplex');
106 [x,fval] = linprog(-f, Y, x, [], [], zeros(size(f)), [], options)
107
108 xB = [5 6 7 9 11 12];
109 xN = [1 2 3 4 8 10];
110
111 B = A(:, xB);
112 N = A(:, xN);
113 Binv = inv(B);
114 % Dictionary constant value column.
115 Binv*b
116 % Dictionary coefficient value rows.
117 -Binv*N
118
119 cB = c(:, xB);
120 cN = c(:, xN);
121 % Objective function constant value.
122 cB*(Binv*b)
123 % Object function coefficient values.
124 cN - cB*(Binv*N)
125
126 % P3 d
127 %  $z = (-2+u)x_1 + (-3+u)x_2 - x_3 - x_4 + (1+u)x_6$ 
128 u = 0;
129 c = [(-2+u) (-3+u) -1 -1 0 (+1+u) 0 0 0 0 0 0];
130
131 xB = [5 6 7 9 11 12];
132 xN = [1 2 3 4 8 10];
133
134 B = A(:, xB);
135 N = A(:, xN);
136 Binv = inv(B);
137 % Dictionary constant value column.
138 Binv*b
139 % Dictionary coefficient value rows.
140 -Binv*N
141
142 cB = c(:, xB);
143 cN = c(:, xN);
144 % Objective function constant value.
145 cB*(Binv*b)
146
147 % Check if all values in constant column are  $\geq 0$ 
148 % This means the dictionary is feasible.
149 if all(Binv*b>=0)
150     disp('Dictionary is feasible.');
```



```

151 else
152     disp('Dictionary is not feasible.');
```

$$cN - cB*(B_{inv}*N)$$

```

153 end
154
155 % Object function coefficient values.
156 cN - cB*(B_{inv}*N)
157
158 % Check if all objective function coefficients are  $\leq 0$ . This means the dictionary is final.
159 if all(cN - cB*(B_{inv}*N)  $\leq 0$ )
160     disp('Dictionary is final.');
```

$$cN - cB*(B_{inv}*N)$$

```

161 else
162     disp('Dictionary is not final.');
```

$$cN - cB*(B_{inv}*N)$$

```

163 end
164
165 % P3 e
166 y = 1;
167 b = b + (y.*[1 1 0 0 0 -1])';
168 c = [-2 -3 -1 -1 0 +1 0 0 0 0 0 0];
169
170 xB = [5 6 7 9 11 12];
171 xN = [1 2 3 4 8 10];
172
173 B = A(:, xB);
174 N = A(:, xN);
175 B_{inv} = inv(B);
176 % Dictionary constant value column.
177 B_{inv}*b;
178 % Dictionary coefficient value rows.
179 -B_{inv}*N;
180
181 cB = c(:, xB);
182 cN = c(:, xN);
183 % Objective function constant value.
184 cB*(B_{inv}*b);
185
186 % Check if all values in constant column are  $\geq 0$ 
187 % This means the dictionary is feasible.
188 B_{inv}*b
189 if all(B_{inv}*b  $\geq 0$ )
190     disp('Dictionary is feasible.');
```

$$cN - cB*(B_{inv}*N)$$

```

191 else
192     disp('Dictionary is not feasible.');
```

$$cN - cB*(B_{inv}*N)$$

```

193 end
194
195 % Object function coefficient values.
196 cN - cB*(B_{inv}*N);
```