

## HW 5 Solutions

### Problems 1 and 2(a-e)

Problem	$n$	$f_1$	$f_2$	DOF
1	6	6	1	2
2a	8	8	1	4
2b	8	10	0	1
2b (alt)	7	8	1	1
2c	6	7	0	1
2d	12	15	0	3
2e	10	12	0	3

### Problem 3a

```
1  /*prob3a.ch*/
2
3  #include <stdio.h>
4  #include <numeric.h>
5
6  #define DEG2RAD(x) ((x)*M_PI/180.0)
7  #define RAD2DEG(x) ((x)*180.0/M_PI)
8
9  int main () {
10     double r1, r2, r3, r4;
11     double complex z3;
12     double theta1;
13     double rp;
14     double beta;
15     double theta2;
16
17     /* Two of each of these for 2 separate circuits */
18     double complex P[2];
19     double complex B[2];
20     double theta3[2];
21     double theta4[2];
22
23     r1 = 5.5;
24     r2 = 1.0;
25     r3 = 3.0;
26     r4 = 4.5;
27     theta1 = DEG2RAD(10.0);
28     rp = 2.5;
29     beta = DEG2RAD(30.0);
30     theta2 = DEG2RAD(45.0);
31
32     z3 = polar(r1, theta1) - polar(r2, theta2);
33     complexsolveRR(r3, -r4, z3, theta3[0], theta4[0], theta3[1],
34                   theta4[1]);
35
36     B[0] = polar(r2, theta2) + polar(r3, theta3[0]);
37     B[1] = polar(r2, theta2) + polar(r3, theta3[1]);
38
39     P[0] = polar(r2, theta2) + polar(rp, theta3[0] + beta);
```

```

39 P[1] = polar(r2, theta2) + polar(rp, theta3[1] + beta);
40
41 printf("CIRCUIT 1\n");
42 printf("theta3_1 = %7.4lf (radian) = %7.4lf (degree), theta4_1
= %7.4lf (radian) = %7.4lf (degree)\n", theta3[0], RAD2DEG(
theta3[0]), theta4[0], RAD2DEG(theta4[0]));
43 printf("B(%lf,%lf) \nP(%lf,%lf)\n", real(B[0]), imag(B[0]),
real(P[0]), imag(P[0]));
44
45
46 printf("\nCIRCUIT 2\n");
47 printf("theta3_1 = %7.4lf (radian) = %7.4lf (degree), theta4_1
= %7.4lf (radian) = %7.4lf (degree)\n", theta3[1], RAD2DEG(
theta3[1]), theta4[1], RAD2DEG(theta4[1]));
48 printf("B(%lf,%lf) \nP(%lf,%lf)\n", real(B[1]), imag(B[1]),
real(P[1]), imag(P[1]));
49
50 return 0;
51 }

```

### Problem 3b

```

1  /*prob3b.ch*/
2
3  #include <stdio.h>
4  #include <numeric.h>
5
6  #define DEG2RAD(x) ((x)*M_PI/180.0)
7  #define RAD2DEG(x) ((x)*180.0/M_PI)
8
9  int main () {
10     double r1, r2, r3, r4;
11     double complex z3;
12     double theta1;
13     double rp;
14     double beta;
15     double theta2;
16
17     /* Two of each of these for 2 separate circuits */
18     double complex P[2];
19     double complex B[2];
20     double theta3[2];
21     double theta4[2];
22
23     r1 = getnum("Please enter the length of R1 [5.5]: ", 5.5);
24     r2 = getnum("Please enter the length of R2 [1.0]: ", 1.0);
25     r3 = getnum("Please enter the length of R3 [3.0]: ", 3.0);
26     r4 = getnum("Please enter the length of R4 [4.5]: ", 4.5);
27     theta1 = getnum("Please enter the angle of theta1 in degrees
[10.0]: ", 10.0);
28     theta1 = DEG2RAD(theta1);
29     rp = getnum("Please enter the length r_p [2.5]: ", 2.5);
30     beta = getnum("Please enter the angle Beta in degrees [30.0]: "
, 30.0);
31     beta = DEG2RAD(beta);
32     theta2 = getnum("Please enter the angle theta2 in degrees
[45.0]: ", 45.0);

```

```

33     theta2 = DEG2RAD(theta2);
34     z3 = polar(r1, theta1) - polar(r2, theta2);
35     complexsolveRR(r3, -r4, z3, theta3[0], theta4[0], theta3[1],
        theta4[1]);
36
37     B[0] = polar(r2, theta2) + polar(r3, theta3[0]);
38     B[1] = polar(r2, theta2) + polar(r3, theta3[1]);
39
40     P[0] = polar(r2, theta2) + polar(rp, theta3[0] + beta);
41     P[1] = polar(r2, theta2) + polar(rp, theta3[1] + beta);
42
43     printf("CIRCUIT 1\n");
44     printf("theta3_1 = %7.4lf (radian) = %7.4lf (degree), theta4_1
        = %7.4lf (radian) = %7.4lf (degree)\n", theta3[0], RAD2DEG(
        theta3[0]), theta4[0], RAD2DEG(theta4[0]));
45     printf("B(%lf,%lf) \nP(%lf,%lf)\n", real(B[0]), imag(B[0]),
        real(P[0]), imag(P[0]));
46
47
48     printf("\nCIRCUIT 2\n");
49     printf("theta3_1 = %7.4lf (radian) = %7.4lf (degree), theta4_1
        = %7.4lf (radian) = %7.4lf (degree)\n", theta3[1], RAD2DEG(
        theta3[1]), theta4[1], RAD2DEG(theta4[1]));
50     printf("B(%lf,%lf) \nP(%lf,%lf)\n", real(B[1]), imag(B[1]),
        real(P[1]), imag(P[1]));
51
52     return 0;
53 }

```

### Problem 3c

```

1  /*prob3c.ch*/
2
3  #include <stdio.h>
4  #include <numeric.h>
5  #include <chplot.h>
6
7  #define DEG2RAD(x) ((x)*M_PI/180.0)
8  #define RAD2DEG(x) ((x)*180.0/M_PI)
9
10 #define N 361
11
12 int main () {
13     double r1, r2, r3, r4;
14     double complex z3;
15     double theta1;
16     double rp;
17     double beta;
18     double theta2[N];
19
20     /* Two of each of these for 2 separate circuits */
21     double complex P[2][N];
22     double P_x[2][N], P_y[2][N];
23     double complex B[2][N];
24     double B_x[2][N], B_y[2][N];
25     double theta3[2][N];
26     double theta4[2][N];

```

```

27 double theta3_u[2][N];
28 double theta4_u[2][N];
29
30 int i;
31
32 CPlot plot, *subplot, plot1, plot2, plot3, plot4, plot5, plot6;
33
34 r1 = getnum("Please enter the length of R1 [5.5]: ", 5.5);
35 r2 = getnum("Please enter the length of R2 [1.0]: ", 1.0);
36 r3 = getnum("Please enter the length of R3 [3.0]: ", 3.0);
37 r4 = getnum("Please enter the length of R4 [4.5]: ", 4.5);
38 theta1 = getnum("Please enter the angle of theta1 in degrees
39 [10.0]: ", 10.0);
40 theta1 = DEG2RAD(theta1);
41 rp = getnum("Please enter the length r_p [2.5]: ", 2.5);
42 beta = getnum("Please enter the angle Beta in degrees [30.0]: "
43 , 30.0);
44 beta = DEG2RAD(beta);
45
46 for(i = 0; i < N; i++) {
47     theta2[i] = i * (2.0*M_PI)/(double)(N-1);
48     z3 = polar(r1, theta1) - polar(r2, theta2[i]);
49     complexsolveRR(r3, -r4, z3, theta3[0][i], theta4[0][i],
50     theta3[1][i], theta4[1][i]);
51
52     B[0][i] = polar(r2, theta2[i]) + polar(r3, theta3[0][i]);
53     B_x[0][i] = real(B[0][i]);
54     B_y[0][i] = imag(B[0][i]);
55
56     B[1][i] = polar(r2, theta2[i]) + polar(r3, theta3[1][i]);
57     B_x[1][i] = real(B[1][i]);
58     B_y[1][i] = imag(B[1][i]);
59
60     P[0][i] = polar(r2, theta2[i]) + polar(rp, theta3[0][i] +
61     beta);
62     P_x[0][i] = real(P[0][i]);
63     P_y[0][i] = imag(P[0][i]);
64
65     P[1][i] = polar(r2, theta2[i]) + polar(rp, theta3[1][i] +
66     beta);
67     P_x[1][i] = real(P[1][i]);
68     P_y[1][i] = imag(P[1][i]);
69 }
70
71 unwrap(theta3_u[0], theta3[0]);
72 unwrap(theta3_u[1], theta3[1]);
73 unwrap(theta4_u[0], theta4[0]);
74 unwrap(theta4_u[1], theta4[1]);
75
76 /* Make Subplot stuff */
77
78 plot.subplot(3,2);
79 subplot = plot.getSubplot(0,0);
80 subplot->data2D(theta2, theta4_u[0]);
81 subplot->title("theta2 vs theta4, Ckt 1");
82
83 subplot = plot.getSubplot(0,1);

```

```

79 subplot->data2D(theta2, theta4_u[1]);
80 subplot->title("theta2 vs theta4, Ckt 2");
81
82 subplot = plot.getSubplot(1,0);
83 subplot->data2D(B_x[0], B_y[0]);
84 subplot->title("Bx vs By, Ckt 1");
85
86 subplot = plot.getSubplot(1,1);
87 subplot->data2D(B_x[1], B_y[1]);
88 subplot->title("Bx vs By, Ckt 2");
89
90 subplot = plot.getSubplot(2,0);
91 subplot->data2D(P_x[0], P_y[0]);
92 subplot->title("Px vs Py, Ckt 1");
93
94 subplot = plot.getSubplot(2,1);
95 subplot->data2D(P_x[1], P_y[1]);
96 subplot->title("Px vs Py, Ckt 2");
97
98 plot.plotting();
99
100
101 plot1.data2D(theta2, theta4_u[0]);
102 plot1.title("theta2 vs theta4, Ckt 1");
103 plot1.label(PLOT_AXIS_X, "theta2 (rad)");
104 plot1.label(PLOT_AXIS_Y, "theta4 (rad)");
105 plot1.plotting();
106
107 plot2.data2D(theta2, theta4_u[1]);
108 plot2.title("theta2 vs theta4, Ckt 2");
109 plot2.label(PLOT_AXIS_X, "theta2 (rad)");
110 plot2.label(PLOT_AXIS_Y, "theta4 (rad)");
111 plot2.plotting();
112
113 plot3.data2D(B_x[0], B_y[0]);
114 plot3.title("Bx vs By, Ckt 1");
115 plot3.sizeRatio(1);
116 plot3.axisRange(PLOT_AXIS_X, 0.0, 2.5);
117 plot3.axisRange(PLOT_AXIS_Y, 1.5, 4.0);
118 plot3.ticksRange(PLOT_AXIS_X, 0.5);
119 plot3.ticksRange(PLOT_AXIS_Y, 0.5);
120 plot3.plotting();
121
122 plot4.data2D(B_x[1], B_y[1]);
123 plot4.title("Bx vs By, Ckt 2");
124 plot4.sizeRatio(1);
125 plot4.axisRange(PLOT_AXIS_X, 1.0, 3.0);
126 plot4.axisRange(PLOT_AXIS_Y, -3.0, -1.0);
127 plot4.ticksRange(PLOT_AXIS_X, 0.5);
128 plot4.ticksRange(PLOT_AXIS_Y, 0.5);
129 plot4.plotting();
130
131 plot5.data2D(P_x[0], P_y[0]);
132 plot5.title("Px vs Py, Ckt 1");
133 plot5.sizeRatio(1);
134 plot5.axisRange(PLOT_AXIS_X, -1.5, 1.5);
135 plot5.axisRange(PLOT_AXIS_Y, 1.0, 4.0);

```

```
136     plot5.ticsRange(PLOT_AXIS_X, 0.5);
137     plot5.ticsRange(PLOT_AXIS_Y, 0.5);
138     plot5.plotting();
139
140     plot6.data2D(P_x[1], P_y[1]);
141     plot6.title("Px vs Py, Ckt 2");
142     plot6.sizeRatio(1);
143     plot6.axisRange(PLOT_AXIS_X, 1.0, 3.5);
144     plot6.axisRange(PLOT_AXIS_Y, -1.5, 1.0);
145     plot6.ticsRange(PLOT_AXIS_X, 0.5);
146     plot6.ticsRange(PLOT_AXIS_Y, 0.5);
147     plot6.plotting();
148
149     return 0;
150 }
```