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

```
/*prob3a.ch*/

#include <stdio.h>
#include <numeric.h>

#define DEG2RAD(x) ((x)*M_PI/180.0)
#define RAD2DEG(x) ((x)*180.0/M_PI)

int main () {
    double r1, r2, r3, r4;
    double complex z3;
    double theta1;
    double rp;
    double beta;
    double theta2;

    /* Two of each of these for 2 separate circuits */
    double complex P[2];
    double complex B[2];
    double theta3[2];
    double theta4[2];

    r1 = 5.5;
    r2 = 1.0;
    r3 = 3.0;
    r4 = 4.5;
    theta1 = DEG2RAD(10.0);
    rp = 2.5;
    beta = DEG2RAD(30.0);
    theta2 = DEG2RAD(45.0);

    z3 = polar(r1, theta1) - polar(r2, theta2);
    complexsolveRR(r3, -r4, z3, theta3[0], theta4[0], theta3[1], theta4[1]);

    B[0] = polar(r2, theta2) + polar(r3, theta3[0]);
    B[1] = polar(r2, theta2) + polar(r3, theta3[1]);

    P[0] = polar(r2, theta2) + polar(rp, theta3[0] + beta);
    P[1] = polar(r2, theta2) + polar(rp, theta3[1] + beta);

    printf("CIRCUIT 1\n");
    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]));
    printf("B(%lf,%lf) \nP(%lf,%lf)\n", real(B[0]), imag(B[0]), real(P[0]), imag(P[0]));


    printf("\nCIRCUIT 2\n");
    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]));
    printf("B(%lf,%lf) \nP(%lf,%lf)\n", real(B[1]), imag(B[1]), real(P[1]), imag(P[1]));

    return 0;
}
```

Problem 3b

```
/*prob3b.ch*/

#include <stdio.h>
#include <numeric.h>

#define DEG2RAD(x) ((x)*M_PI/180.0)
#define RAD2DEG(x) ((x)*180.0/M_PI)

int main () {
    double r1, r2, r3, r4;
    double complex z3;
    double theta1;
    double rp;
    double beta;
    double theta2;

    /* Two of each of these for 2 separate circuits */
    double complex P[2];
    double complex B[2];
    double theta3[2];
    double theta4[2];

    r1 = getnum("Please enter the length of R1 [5.5]: ", 5.5);
    r2 = getnum("Please enter the length of R2 [1.0]: ", 1.0);
    r3 = getnum("Please enter the length of R3 [3.0]: ", 3.0);
    r4 = getnum("Please enter the length of R4 [4.5]: ", 4.5);
    theta1 = getnum("Please enter the angle of theta1 in degrees [10.0]: ", 10.0);
    theta1 = DEG2RAD(theta1);
    rp = getnum("Please enter the length r_p [2.5]: ", 2.5);
    beta = getnum("Please enter the angle Beta in degrees [30.0]: ", 30.0);
    beta = DEG2RAD(beta);
    theta2 = getnum("Please enter the angle theta2 in degrees [45.0]: ", 45.0);
    theta2 = DEG2RAD(theta2);
    z3 = polar(r1, theta1) - polar(r2, theta2);
    complexsolveRR(r3, -r4, z3, theta3[0], theta4[0], theta3[1], theta4[1]);

    B[0] = polar(r2, theta2) + polar(r3, theta3[0]);
    B[1] = polar(r2, theta2) + polar(r3, theta3[1]);

    P[0] = polar(r2, theta2) + polar(rp, theta3[0] + beta);
    P[1] = polar(r2, theta2) + polar(rp, theta3[1] + beta);

    printf("CIRCUIT 1\n");
    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]));
    printf("B(%lf,%lf) \nP(%lf,%lf)\n", real(B[0]), imag(B[0]), real(P[0]), imag(P[0]));


    printf("\nCIRCUIT 2\n");
    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]));
    printf("B(%lf,%lf) \nP(%lf,%lf)\n", real(B[1]), imag(B[1]), real(P[1]), imag(P[1]));

    return 0;
}
```

Problem 3c

```
/*prob3c.ch*/

#include <stdio.h>
#include <numeric.h>
#include <chplot.h>

#define DEG2RAD(x) ((x)*M_PI/180.0)
#define RAD2DEG(x) ((x)*180.0/M_PI)

#define N 361

int main () {
    double r1, r2, r3, r4;
    double complex z3;
    double theta1;
    double rp;
    double beta;
    double theta2[N];

    /* Two of each of these for 2 separate circuits */
    double complex P[2][N];
    double P_x[2][N], P_y[2][N];
    double complex B[2][N];
    double B_x[2][N], B_y[2][N];
    double theta3[2][N];
    double theta4[2][N];
    double theta3_u[2][N];
    double theta4_u[2][N];

    int i;

    CPlot plot, *subplot, plot1, plot2, plot3, plot4, plot5, plot6;

    r1 = getnum("Please enter the length of R1 [5.5]: ", 5.5);
    r2 = getnum("Please enter the length of R2 [1.0]: ", 1.0);
    r3 = getnum("Please enter the length of R3 [3.0]: ", 3.0);
    r4 = getnum("Please enter the length of R4 [4.5]: ", 4.5);
    theta1 = getnum("Please enter the angle of theta1 in degrees [10.0]: ", 10.0);
    theta1 = DEG2RAD(theta1);
    rp = getnum("Please enter the length r_p [2.5]: ", 2.5);
    beta = getnum("Please enter the angle Beta in degrees [30.0]: ", 30.0);
    beta = DEG2RAD(beta);

    for(i = 0; i < N; i++) {
        theta2[i] = i * (2.0*M_PI)/(double)(N-1);
        z3 = polar(r1, theta1) - polar(r2, theta2[i]);
        complexsolveRR(r3, -r4, z3, theta3[0][i], theta4[0][i], theta3[1][i], theta4[1][i]);

        B[0][i] = polar(r2, theta2[i]) + polar(r3, theta3[0][i]);
        B_x[0][i] = real(B[0][i]);
        B_y[0][i] = imag(B[0][i]);

        B[1][i] = polar(r2, theta2[i]) + polar(r3, theta3[1][i]);
        B_x[1][i] = real(B[1][i]);
        B_y[1][i] = imag(B[1][i]);

        P[0][i] = polar(r2, theta2[i]) + polar(rp, theta3[0][i] + beta);
        P_x[0][i] = real(P[0][i]);
        P_y[0][i] = imag(P[0][i]);

        P[1][i] = polar(r2, theta2[i]) + polar(rp, theta3[1][i] + beta);
        P_x[1][i] = real(P[1][i]);
        P_y[1][i] = imag(P[1][i]);
    }

    unwrap(theta3_u[0], theta3[0]);
    unwrap(theta3_u[1], theta3[1]);
    unwrap(theta4_u[0], theta4[0]);
    unwrap(theta4_u[1], theta4[1]);

    /* Make Subplot stuff */

    plot.subplot(3,2);
    subplot = plot.getSubplot(0,0);
    subplot->data2D(theta2, theta4_u[0]);
    subplot->title("theta2 vs theta4, Ckt 1");

    subplot = plot.getSubplot(0,1);
    subplot->data2D(theta2, theta4_u[1]);
    subplot->title("theta2 vs theta4, Ckt 2");

    subplot = plot.getSubplot(1,0);
    subplot->data2D(B_x[0], B_y[0]);
    subplot->title("Bx vs By, Ckt 1");

    subplot = plot.getSubplot(1,1);
    subplot->data2D(B_x[1], B_y[1]);
    subplot->title("Bx vs By, Ckt 2");

    subplot = plot.getSubplot(2,0);
    subplot->data2D(P_x[0], P_y[0]);
    subplot->title("Px vs Py, Ckt 1");

    subplot = plot.getSubplot(2,1);
    subplot->data2D(P_x[1], P_y[1]);
    subplot->title("Px vs Py, Ckt 2");

    plot.plotting();


    plot1.data2D(theta2, theta4_u[0]);
    plot1.title("theta2 vs theta4, Ckt 1");
    plot1.label(PLOT_AXIS_X, "theta2 (rad)");
    plot1.label(PLOT_AXIS_Y, "theta4 (rad)");
    plot1.plotting();

    plot2.data2D(theta2, theta4_u[1]);
    plot2.title("theta2 vs theta4, Ckt 2");
    plot2.label(PLOT_AXIS_X, "theta2 (rad)");
    plot2.label(PLOT_AXIS_Y, "theta4 (rad)");
    plot2.plotting();

    plot3.data2D(B_x[0], B_y[0]);
    plot3.title("Bx vs By, Ckt 1");
    plot3.sizeRatio(1);
    plot3.axisRange(PLOT_AXIS_X, 0.0, 2.5);
    plot3.axisRange(PLOT_AXIS_Y, 1.5, 4.0);
    plot3.ticsRange(PLOT_AXIS_X, 0.5);
    plot3.ticsRange(PLOT_AXIS_Y, 0.5);
    plot3.plotting();

    plot4.data2D(B_x[1], B_y[1]);
    plot4.title("Bx vs By, Ckt 2");
    plot4.sizeRatio(1);
    plot4.axisRange(PLOT_AXIS_X, 1.0, 3.0);
    plot4.axisRange(PLOT_AXIS_Y, -3.0, -1.0);
    plot4.ticsRange(PLOT_AXIS_X, 0.5);
    plot4.ticsRange(PLOT_AXIS_Y, 0.5);
    plot4.plotting();

    plot5.data2D(P_x[0], P_y[0]);
    plot5.title("Px vs Py, Ckt 1");
    plot5.sizeRatio(1);
    plot5.axisRange(PLOT_AXIS_X, -1.5, 1.5);
    plot5.axisRange(PLOT_AXIS_Y, 1.0, 4.0);
    plot5.ticsRange(PLOT_AXIS_X, 0.5);
    plot5.ticsRange(PLOT_AXIS_Y, 0.5);
    plot5.plotting();

    plot6.data2D(P_x[1], P_y[1]);
    plot6.title("Px vs Py, Ckt 2");
    plot6.sizeRatio(1);
    plot6.axisRange(PLOT_AXIS_X, 1.0, 3.5);
    plot6.axisRange(PLOT_AXIS_Y, -1.5, 1.0);
    plot6.ticsRange(PLOT_AXIS_X, 0.5);
    plot6.ticsRange(PLOT_AXIS_Y, 0.5);
    plot6.plotting();

    return 0;
}
```