

## APPENDIX

### Phase Rotation Program

#### 1. Main program for phase rotation (protmn.cc)

```
// Filename: protmn.cc
// Main C++ program for phase rotation

#include <stdio.h>
#include <fstream.h>
#include <iostream.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>
#include <math.h>

extern "C" {
    extern void protsb_ (int*,int*,float*,float*,
        float*,float*,char(*)[60],char(*)[60]);
}

main()
{
    int n1, n2;
    float *a, *b, *rot3p, theta;
    char in[60], out[60];
    char ok;

    cout << "what is angle in degree (theta)? i.e 30";
```

```
cin >> theta;
cout << "What is number of time samples (n1)? i.e. 10 ";
cin >> n1;
cout << "What is number of traces (n2)? i.e. 10 ";
cin >> n2;
cout << "What is input file name (in)? i.e. in ";
cin >> in;
cout << "What is output file name (out)? i.e. out ";
cin >> out;

cout << "theta=" << theta << "\n";
cout << "n1=" << n1 << "\n";
cout << "n2=" << n2 << "\n";
cout << "input=" << in << "\n";
cout << "output=" << out << "\n";

cout << "Are this parameters ok? type yes or no ";
cin >> ok;
if (ok == 'n')
{
    cout << "Try again " << "\n";
    return 1;
}

a=(float*)malloc(3*n1*n2*sizeof(float));
b=(float*)malloc(3*n1*n2*sizeof(float));
rot3p=(float*)malloc(3*sizeof(float));

protsb_(&theta, &n1, &n2, a, b, rot3p, &in, &out);

return 0;
}
```

## 2. Subroutine program for phase rotation

### 2.1 protsb.f

```

C23456789012345678901234567890123456789012345678901234567890123456789
C
C Filename: protsb.f
C
C SUBROUTINE PROTSB(N1,N2,THETA,ROT3P,A,B,IN,OUT)
C
C SUBROUTINE TO CONVOLVE INPUT WITH A THREE-POINT PHASE ROTATOR
C N1 = NUMBER OF TIME SAMPLES
C N2 = NUMBER OF TRACES
C THETA = ANGLE OF PHASE ROTATION (DEGREE)
C ROT3P = THREE-POINT PHASE ROTATOR
C A = INPUT MATRIX
C B = OUTPUT MATRIX
C IN = INPUT FILE NAME
C OUT = OUTPUT FILE NAME
C
C REAL ROT3P(3),THETA,A(N1,N2),B(N1+2,N2)
C INTEGER N1,N2
C CHARACTER IN*60, OUT*60
C OPEN(7,FILE=IN,ACCESS='DIRECT',
C *RECL=N1*4,FORM='UNFORMATTED')
C WRITE(*,*) ' OPENED FILE 7 - INPUT'
C OPEN(8,FILE=OUT,ACCESS='DIRECT',
C *RECL=N1*4,FORM='UNFORMATTED')
C WRITE(*,*) ' OPENED FILE 8 - OUTPUT'
C
C GET A THREE-POINT PHASE ROTATION
C CALL ROTA3P(ROT3P,THETA)
C
C READ INPUT
C
C DO L=1,N2
C READ(7,REC=L) (A(I,L),I=1,N1)
C
C END DO

```

```

CALL ZERO2D(B,N1,N2)
C
C CONVOLVE INPUT WITH 3-POINT PHASE ROTATION
C
DO L=1,N2
  DO J=1,N1+2
    DO I=1,3
      K=I+J-1
      B(K,L)=B(K,L)+A(J,L)*ROT3P(I)
    END DO
  END DO
END DO
C
C WRITE OUTPUT
C
DO L=1,N2
  WRITE(8,REC=L) (B(I,L),I=2,N1+1)
END DO
WRITE(*,*) ' WROTE OUTPUT MATRIX'
CLOSE (7,STATUS='KEEP')
CLOSE (8,STATUS='KEEP')
RETURN
END

```

## 2.2 rota3p.f

C23456789012345678901234567890123456789012345678901234567890123456789

```

C File name : rota3p.f
C
SUBROUTINE ROTA3P(ROT3P,THETA)
C
C COMPUTE THREE-POINT PHASE ROTATION
C SPIKE = 3-POINT SPIKE INPUT
C QUAD = 3-POINT QUADRATURE INPUT
C THETA = ROTATION ANGLE IN DEGREE
C ROT3P = 3-POINT PHASE ROTATION OUTPUT
C
REAL SPIKE(3),QUAD(3),ROT3P(3),THETA

```

```

DATA SPIKE/0.0000,1.0000,0.0000/
DATA QUAD/-0.7071,0.0000,0.7071/
DO I=1,3
  ROT3P(I)=SPIKE(I)*COS(-0.017453*THETA)
*   -QUAD(I)*SIN(-0.017453*THETA)
  WRITE(*,100) THETA,SPIKE(I),QUAD(I),ROT3P(I)
END DO
100 FORMAT(F5.0,3F9.4)
RETURN
END

```

### 2.3 zero2d.f

```

C23456789012345678901234567890123456789012345678901234567890123456789
SUBROUTINE ZERO2D(X,N1,N2)
C
C INITIALIZE A 2D MATRIX X(N1+2,N2) WITH ZEROS
C
DIMENSION X(N1+2,N2)
DO I=1,N1+2
  DO J=1,N2
    X(I,J)=0.0
  END DO
END DO
RETURN
END

```

## Curriculum Vitae



**Name :** Miss Suwimon Udphuay

**Birth Date :** August 29, 1975

**Education :** 1997 Received Bachelor Degree (B.S.) in Physics from Chiang Mai University

**Scholarship :** 1994-1999 Received a scholarship from the Development and Promotion of Science and Technology Talents Project, DPST