REFERENCES

Billiotte, J., De Moegen, H., and Oren, P.E. (1993) Experimental micro-modeling and numerical simulation of gas/water injection/withdrawal cycles as applied to underground gas storage reservoir. SPE Advanced Technology Series,1(1), 133-139.

Bittkow, P., Kretzschmar, H-J., and Kana, J. (1991) Simulation of conversion of Lobodice underground gas storage facility from town gas to natural gas. Erdoel Erdgas Kohle, 107(3), 118-123.

Carnahan, B., Luther, H.A., and Wilkes, J.O. (1969). Applied Numerical Methods, New York: Wiley.

Chapra, S.C., Canale, R.P. (2002). Numerical Methods for Engineerings, McGraw-Hill.

Killincer, N. and Gumrah, F. (2000) Numerical simulation study on mixing of inert cushion gas with working gas in an underground gas storage reservoir. Energy Sources, 20(10), 869-879.

Misra, B.R., Foh, S.E., Shillari, Y.A., Berry,R.M., and Labaune, F. (1988). Use of inert base gas in underground natural gas storage. Special Report, 20, 162-169.

Park, Y.C., Syng, W.M., and Kim, S.J. (2002) Development of a FEM reservoir model equipped with an effective permeability tensor and its application to naturally fractured reservoirs. Energy Sources, 24(6), 531-542.

Rasin Tek, M. and Wilkes, J.O. (1966) New Concept in Underground Storage of Natural Gas. New York: American Gas Association Inc.

Sonier, F., Lehuen, P., and Nabil, R. (1993) Full-field gas storage simulation using a control-volume finite-element model. Proceedings - SPE Annual Technical Conference and Exhibition, Sigma, Reservoir Engineering, 10, 515-524.

Tan, Y., Chen, j., and Yu, Q. (2001) An optimization running analysis of bottom water drive underground natural gas storage in seasonal peak shaying. Natural Gas Industry, 21(3), 86-89.

Tan, Yu-Fei. and Chen, Jia-Xin. (2001). Optimization of underground natural gas storage reservoir in summer. Journal of Harbin Institute of Technology, 34(4), 525-528.

Wilkes, J.O. (1999) Fluid Mechanics for Chemical Engineering. New Jersey: Pretice-Hall PTR.

Yeh, Y.J. and Kuo, M.C.T. (1994) Design of tracer tests for natural gas underground storage operations - a simulation approach. Proceedings of the SPE Latin American and Caribbean Petroleum Engineering Conference, 3, 1231-1245.

Appendix A Source Code for Natural Gas Storage Problem .

```
      PROGRAM GAS STORAG
c     PRELIMINALY VERSION
c     Program for computing transient pressures in gas-storage field.

c      ALPHA            TEMP*PS/TS
c      BETA             POROS*SQRT(MU/2)
c      BIGQS            Current gas withdrawal rate, SCF/day
c      C                Conversion Factor, 6.327E-3 cp sq ft/md day psia
c      COUNT            Counter on number of time steps
c      CX(I,J)          C*K(I,j)*DT/(2*BETA*DX**2)
c      CY(I,J)          C*K(I,j)*DT/(2*BETA*DY**2)
c      DELTA(I,J)       Array of gas withdrawals.  The only nonzero value
c                       occurs at the single well, I = IW, J = JW
c      DT               Time step, days
c      DX,DY            Grid spacings in the x and y direction,ft
c      EPS              Porosity
c      FREQ             frequency of printing
c      H                Formation thickness, ft
c      I,J              Column and row indicates (x and y direction)
c      IW,JW            Indicates of block in which well is located
c      K                Permeability, md
c      L                Length of storage field,ft
c      M                Number of increments in x direction
c      MU               Gas viscosity,cp
c      N                Number of increments in y direction
c      P                Pressure,psia
c      PD               Minimun required delivery pressure into well,psia
c      PHI              Gas potential,(psia)^2/cp
c      PS,TS            standard pressurea(psia),standaard temperature (R)
c      PZERO            Initial Pressure (uniform).(psia)
c      Q                Gas withdrawal rate,SCF/day
c      QMAX             Maximun gas withdrawal rate,SCF/day
c      QS               Gas withdrawal rate,SCF/day cu ft =1/day
c      RE               Effective drainage radius of well, ft
c      RW               Wellbore radius, ft
c      T                Time,hr
c      TEMP             Reservoir temperature,R
c      TMAX             Maximun time for simulation,days
c      W                Width of storage field,ft
c     ................Declarations..............

      IMPLICIT NONE
      INTEGER   COUNT,FREQ,I,IW,J,JW,M,N,MP1,NP1
      REAL    ALPHA,BETA,BIGQS,C,CX,CY,DELTA,DT,DX,DY,EPS,H,K
      REAL  L,MU,P,PD,PHI,PI,PS,PZERO,PW,QMAX,QS,RE,RW,T,TEMP
```

```
        REAL   TMAX,TS,W
        DIMENSION CX(21,21),CY(21,21),DELTA(21, 21)
        DIMENSION K(21,21),P(21,21),PHI(21,21)
        OPEN   (4, FILE='CarbonateInput Data1.1.dat')
        OPEN   (6, FILE='CarbonateOutput1.1.dat')
        OPEN   (7, FILE='CarbonatePrint1.1.dat')

c       ................Read and check input parameters...............
        READ(4,*)DT,EPS,FREQ,H,IW,JW,L,M,MU,N,PD,PS,PZERO,
     *    QMAX,RW,TEMP,TMAX,TS,W
        MP1 = M+1
        NP1 = N+1
        DO 601 J=NP1,1,-1
    READ(4,*)(K(I,J),I=1,MP1)
601     CONTINUE
        DO 602 J=NP1,1,-1
    READ(4,*)(P(I,J),I=1,MP1)
602     CONTINUE
        WRITE (6,201)DT,EPS,FREQ,H,IW,JW,L,M,MU,N,PD,PS,PZERO,
     *        QMAX,RW,TEMP,TMAX,TS,W

c       ................Initialize value..............
         PI=3.14159
         C = 6.327E-3
         COUNT = 0
         DX = L/M
         DY = W/N
         QS = 0.
         RE = SQRT(DX*DY)
         RE = 1968.5
         WRITE (6,202) DX,DY,RE
         ALPHA = TEMP*PS/TS
         BETA = EPS*SQRT(MU/2.)
     T=0.
         MP1=M+1
         NP1=N+1
    DO 3 I=1,MP1
         DO 2 J=1,NP1
                CX(I,J)=C*K(I,J)*DT/(2.*BETA*DX**2)
                CY(I,J)=C*K(I,J)*DT/(2.*BETA*DY**2)
                DELTA(I,J) = 0.
c        P(I,J)=PZERO
        PHI(I,J)=P(I,J)**2/(2.*MU)
  2      END DO
  3 END DO

c       ...Inertial pertubation at center should declay with time..............
```

```
c         PHI(6,6)=2.* PHI(6,6)
c         P(6,6)= SQRT(2.*MU*PHI(6,6))

c        .......Print premiability (1 for k), and initial pressure
c                 (2 for P) and potential (3 for PHI) field..................
     CALL PRINT (1,M,N,K)
        WRITE (6,203) T,QS
        CALL PRINT (2,M,N,P)
        CALL PRINT (3,M,N,PHI)

c        ....Perform calculations over successive time steps...............
   4    COUNT=COUNT+1
        T=T+DT

c    ..... Update the value of the well withdrawal rate .....
     BIGQS = QMAX*(1. - ABS(2.*T/TMAX - 1.))
     QS = BIGQS/(DX*DY*H)
     DELTA(IW,JW) = ALPHA*SQRT(PHI(IW,JW))*QS*DT/(2.*BETA)
c        ................Update potentials...............
        CALL IAD (CX,CY,DELTA,M,N,PHI)
c        ....Print pressure and potential fields when request,
c          first converting potentail to pressure..............
         IF (COUNT/FREQ*FREQ.EQ.COUNT) THEN
        DO I=1,MP1
              DO J=1,NP1
                    P(I,J)=SQRT(2.*MU*ABS(PHI(I,J)))
              END DO
        END DO
              WRITE (6,203)T,QS
              CALL PRINT (2,M,N,P)
              CALL PRINT (3,M,N,PHI)
        END IF
c    ..... Stop if delivery pressure is unacceptable .....
        PW = SQRT(P(IW,JW)**2 - MU*BIGQS*ALPHA/(PI*K(IW,JW)*H)*
   *       (ALOG(RE/RW) - 0.5)/C)
     WRITE (6,204) PW
     IF (PW .LT. PD)  THEN
       WRITE (6,205)
      STOP
     END IF

c        ...............     Stop if time exceed upper limit..............
        IF (T.GT.TMAX-DT/2.) THEN
              STOP
        ELSE
              GOTO 4
        END IF
```

```
c        .........Formats for output statements............
 201    FORMAT (1X,'Simulation of gas-storage reservoir with'/
   *            'DT              =',F10.3,2X,'days'/
   *            'EPS       =',F10.3/
   *            'FREQ =',I6/
       *                'H      =',F10.3,2X,'ft'/
   *            'IW             =',I6/
   *            'JW             =',I6/
   *            'L              =',F10.3,2X,'ft'/
   *            'M              =',I6/
   *            'MU             =',F10.3,2X,'cp'/
   *            'N              =',I6/
   *      'PD           =',F10.3,2X,'psia'/
   *            'PS             =',F10.3,2X,'psia'/
   *            'PZERO          =',F10.3,2X,'psia'/
   *            'QMAX           =',E10.3,2X,'SCF/days'/
       *                'RW    =',F10.3,2X,'ft'/
   *            'TEMP=',F10.3,2X,'R'/
   *            'TMAX           =',F10.3,2X,'days'/
   *            'TS             =',F10.3,2X,'R'/
   *            'W              =',F10.3,2X,'ft')

 202    FORMAT ('DX                 =',F10.3,2X,'ft'/
   *            'DY             =',F10.3,2X,'ft'/
       *                'RE    =',F10.3,2X,'ft')

 203  FORMAT (' At time T  = ',F8.2,2X,'days,at a withdrawal rate'/
   *       '       QS = ',F8.2,2X,'1/days:')
 204  FORMAT (' PW    = ', F10.3, 2X, 'psia')
 205  FORMAT (' Delivery pressure has fallen to unacceptable value:'/
   *       ' STOP EXECUTION')
      STOP
         END


         SUBROUTINE IAD (CX,CY,DELTA,M,N,PHI)
C     Updates the gas potentials across a time step. Variables are same
C     as in main program, with addition of:
C       V       Vector of solutions returned by TRIDAG

      IMPLICIT NONE
      INTEGER I,J,MP1,NP1,M,N
      REAL A,B,C,D,CX,CY,DELTA,PHI,PHISTAR,V,LAMBDAX,LAMBDAY
      DIMENSION A(21),B(21),C(21),CX(21, 21),
     *           CY(21, 21),D(21),DELTA(21, 21),
     *           PHI(21, 21),PHISTAR(21, 21),V(21),
```

```
       *                LAMBDAX(21,21), LAMBDAY(21,21)
       MP1 = M+1
       NP1 = N+1

       DO 101 I= 1,MP1
            DO 102 J= 1,NP1
                 LAMBDAX(I,J) = CX(I,J)*SQRT(ABS(PHI(I,J)))
                 LAMBDAY(I,J) = CY(I,J)*SQRT(ABS(PHI(I,J)))
102         CONTINUE
101    CONTINUE


c      ....Compute temperature for first half time increment(implicit by
rows)..............

    DO 105 J=1,NP1
    B(1)=2.0*(1+LAMBDAX(1,J))
    C(1)=-2.0*LAMBDAX(1,J)
       A(MP1)=-2.0*LAMBDAX(MP1,J)
       B(MP1)=2.0*(1+LAMBDAX(MP1,J))
            IF(J.EQ.1)THEN
                 D(1) = 2.0*(1-LAMBDAY(1,1))*PHI(1,1)
   *               +2.0*LAMBDAY(1,2)*PHI(1,2)-DELTA(1,1)
                 D(MP1)=2.0*(1-LAMBDAY(MP1,1))*PHI(MP1,1)
   *               +2.0*LAMBDAY(MP1,2)*PHI(MP1,2)
   *               -DELTA(MP1,1)
            ELSE IF(J.EQ.NP1)THEN
                 D(1)=2.0*LAMBDAY(1,N)*PHI(1,N)
   *             +2.0*(1-LAMBDAY(1,NP1))*PHI(1,NP1)
   *             -DELTA(1,NP1)
          D(MP1)=2.0*LAMBDAY(MP1,N)*PHI(MP1,N)
   *               +2.0*(1-LAMBDAY(MP1,NP1))*PHI(MP1,NP1)
   *               -DELTA(MP1,NP1)
         ELSE
                 D(1)=LAMBDAY(1,J-1)*PHI(1,J-1)
   *               +2*(1-LAMBDAY(1,J))*PHI(1,J)
   *               +LAMBDAY(1,J+1)*PHI(1,J+1)-DELTA(1,J)
                 D(MP1)=LAMBDAY(MP1,J-1)*PHI(MP1,J-1)
   *               +2*(1-LAMBDAY(MP1,J))*PHI(MP1,J)
   *               +LAMBDAY(MP1,J)*PHI(MP1,J+1)
   *           -DELTA(MP1,J)
            END IF
            DO 103 I=2,M
                 A(I)=-LAMBDAX(I,J)
                 B(I)=2.0*(1+LAMBDAX(I,J))
                 C(I)=-LAMBDAX(I,J)
                 IF(J.EQ.1)THEN
```

```
                         D(I)=2.0*(1-LAMBDAY(I,1))*PHI(I,1)
     *                   +2.0*LAMBDAY(I,2)*PHI(I,2)-DELTA(I,1)
              ELSEIF(J.EQ.NP1)THEN
                         D(I)=2.0*(1-LAMBDAY(I,NP1))*PHI(I,NP1)
     *                   +2.0*LAMBDAY(I,N)*PHI(I,N)-DELTA(I,NP1)
              ELSE
                         D(I)=LAMBDAY(I,J-1)*PHI(I,J-1)
     *                   +2*(1-LAMBDAY(I,J))*PHI(I,J)
       *                   +LAMBDAY(I,J+1)*PHI(I,J+1)-DELTA(I,J)
                    END IF
 103          CONTINUE
              CALL TRIDAG (1,MP1,A,B,C,D,V)
              DO 104  I=1,MP1
                    PHISTAR(I,J)=V(I)
 104          CONTINUE
 105   CONTINUE
```

c ....Compute temperature for second half time increment(implicit by columns)...............

```
       DO 110 I=1,MP1
    B(1)=2.0*(1+LAMBDAY(I,1))
    C(1)=-2.0*LAMBDAY(I,1)
       A(NP1)=-2.0*LAMBDAY(I,NP1)
       B(NP1)=2.0*(1+LAMBDAY(I,NP1))
            IF(I.EQ.1)THEN
                   D(1)=2.0*(1-LAMBDAX(1,1))*PHISTAR(1,1)
   *               +2.0*LAMBDAX(2,1)*PHISTAR(2,1)-DELTA(1,1)
                   D(NP1)=2.0*(1-LAMBDAX(1,NP1))*PHISTAR(1,NP1)
   *                +2.0*LAMBDAX(2,NP1)*PHISTAR(2,NP1)-
DELTA(1,NP1)
            ELSE IF(I.EQ.MP1)THEN
                   D(1)=2.0*LAMBDAX(M,1)*PHISTAR(M,1)
   *               +2.0*(1-LAMBDAX(MP1,1))*PHISTAR(MP1,1)-
DELTA(MP1,1)
          D(NP1)=2.0*LAMBDAX(M,NP1)*PHISTAR(M,NP1)
   *                +2.0*(1-LAMBDAX(MP1,NP1))*PHISTAR(MP1,NP1)
   *                -DELTA(MP1,NP1)
        ELSE
                   D(1)=LAMBDAX(I-1,1)*PHISTAR(I-1,1)
   *               +2*(1-LAMBDAX(I,1))*PHISTAR(I,1)
   *               +LAMBDAX(I+1,1)*PHISTAR(I+1,1)-DELTA(I,1)
                   D(NP1)=LAMBDAX(I-1,NP1)*PHISTAR(I-1,NP1)
   *                +2*(1-LAMBDAX(I,NP1))*PHISTAR(I,NP1)
   *                +LAMBDAX(I+1,NP1)*PHISTAR(I+1,NP1)-
DELTA(I,NP1)
            END IF
```

```
              DO 108 J=2,N
                     A(J)=-LAMBDAY(I,J)
                     B(J)=2.0*(1+LAMBDAY(I,J))
                     C(J)=-LAMBDAY(I,J)
                     IF(I.EQ.1)THEN
                            D(J)=2.0*(1-LAMBDAX(1,J))*PHISTAR(1,J)
   *                        +2.0*LAMBDAX(2,J)*PHISTAR(2,J)-DELTA(1,J)
            ELSEIF(I.EQ.MP1)THEN
                            D(J)=2.0*(1-LAMBDAX(MP1,J))*PHISTAR(MP1,J)
   *                        +2.0*LAMBDAX(M,J)*PHISTAR(M,J)-
DELTA(MP1,J)
            ELSE
                            D(J)=LAMBDAX(I-1,J)*PHISTAR(I-1,J)
   *                        +2*(1-LAMBDAX(I,J))*PHISTAR(I,J)
       *                      +LAMBDAX(I+1,J)*PHI(I+1,J)-DELTA(I,J)
                     END IF
 108          CONTINUE
              CALL TRIDAG (1,NP1,A,B,C,D,V)
              DO 109  J=1,NP1
                     PHI(I,J)=V(J)
 109          CONTINUE
 110   CONTINUE
    RETURN
       END

       SUBROUTINE PRINT (CODE,M,N,X)
c      For printing the fields of permeabillity,pressure,potential, or
c      potential at the end of a haif time-step,depending on the vallue
c      of CODE being 1,2,3,4.
       IMPLICIT NONE
       INTEGER CODE,I,J,M,N,MP1,NP1
       REAL X
       DIMENSION X(21,21)
       MP1 = M+1
       NP1 = N+1
       SELECT CASE (CODE)
       CASE(1)
              WRITE (6,201)
201           FORMAT (' The permeability field md'/
   *       'standting with the row J=NP1 is:'/)
              DO J=NP1,1,-1
                     WRITE (6,202)(X(I,J),I=1,MP1)
202                  FORMAT ('',11F7.1)
              END DO
       CASE(2)
              WRITE (6,203)
203           FORMAT (' The current pressure field (psia) is:'/)
```

```
          DO J=NP1,1,-1
                WRITE (6,204)(X(I,J),I=1,MP1)
204             FORMAT ('',11F7.1)
          END DO
      CASE(3)
          WRITE (6,205)
205       FORMAT (' The current potential field,MM(psia^2/cp)is:'/)
          DO J=NP1,1,-1
                WRITE (6,206)(X(I,J)/1.E6,I=1,MP1)
206             FORMAT ('',11F7.1)
          END DO
      CASE(4)
          WRITE (6,207)
207       FORMAT (' The current pressure field,MM(psia^2/cp)is:'/)
          DO J=NP1,1,-1
                WRITE (6,208)(X(I,J)/1.E6,I=1,MP1)
208             FORMAT ('',11F7.1)
          END DO
      END SELECT
      RETURN
      END


      SUBROUTINE TRIDAG (FIRST,LAST,A,B,C,D,V)
c     Procedure for solving a system of simultaneous
c     linear equation with a triagonal coefficient matrix
      IMPLICIT NONE
      INTEGER FIRST,I,LAST
      REAL A,B,BETA,C,D,GAMMA,V
      DIMENSION A(21),B(21),BETA(21),C(21),D(21)
      DIMENSION GAMMA(21),V(21)
c     ...........Compute intermediate arrays BETA and GAMMA ...............
      BETA(FIRST)=B(FIRST)
      GAMMA(FIRST)=D(FIRST)/BETA(FIRST)
      DO I=FIRST+1,LAST
          BETA(I)=B(I)-A(I)*C(I-1)/BETA(I-1)
          GAMMA(I)=(D(I)-A(I)*GAMMA(I-1))/BETA(I)
      END DO
c     ...........Compute final solution vector V ..............

      V(LAST)=GAMMA(LAST)
      DO I=LAST-1,FIRST,-1
        V(I)=GAMMA(I)-C(I)*V(I+1)/BETA(I)
      END DO
      RETURN
      END
```

Appendix B The relation between Φ and P.

Phi*1.E6	P	P/Z
0.00	0	0
0.10	7.071068	7.071068
0.20	17.07107	17.07107
0.30	27.07107	27.07107
0.40	37.07107	37.07107
0.50	47.07107	47.07107
0.60	57.07107	57.07107
0.70	67.07107	67.07107
0.80	77.07107	77.07107
0.90	87.07107	87.07107
1.00	97.07107	97.07107
1.10	107.0711	107.0711
1.20	117.0711	117.0711
1.30	127.0711	127.0711
1.40	137.0711	137.0711
1.50	147.0711	147.0711
1.60	157.0711	157.0711
1.70	167.0711	167.0711
1.80	177.0711	177.0711
1.90	187.0711	187.0711
2.00	197.0711	197.0711
2.10	207.0711	207.0711
2.20	217.0711	217.0711
2.30	227.0711	227.0711
2.40	237.0711	237.0711
2.50	247.0711	247.0711
2.60	257.0711	257.0711
2.70	267.0711	267.0711
2.80	277.0711	277.0711
2.90	287.0711	287.0711
3.00	297.0711	297.0711
3.10	307.0711	307.0711
3.20	317.0711	317.0711
3.30	327.0711	327.0711
3.40	337.0711	337.0711
3.50	347.0711	347.0711
3.60	357.0711	357.0711
3.70	367.0711	367.0711
3.80	377.0711	377.0711
3.90	387.0711	387.0711
4.00	397.0711	397.0711
4.10	407.0711	407.0711
4.20	417.0711	417.0711
4.30	427.0711	427.0711
4.40	437.0711	437.0711
4.50	447.0711	447.0711
4.60	457.0711	457.0711
4.70	467.0711	467.0711
4.80	477.0711	477.0711
4.90	487.0711	487.0711
5.00	497.0711	497.0711

Phi*1.E6	P	P/Z
5.10	507.0711	507.0711
5.20	517.0711	517.0711
5.30	527.0711	527.0711
5.40	537.0711	537.0711
5.50	547.0711	547.0711
5.60	557.0711	557.0711
5.70	567.0711	567.0711
5.80	577.0711	577.0711
5.90	587.0711	587.0711
6.00	597.0711	597.0711
6.10	607.0711	607.0711
6.20	617.0711	617.0711
6.30	627.0711	627.0711
6.40	637.0711	637.0711
6.50	647.0711	647.0711
6.60	657.0711	657.0711
6.70	667.0711	667.0711
6.80	677.0711	677.0711
6.90	687.0711	687.0711
7.00	697.0711	697.0711
7.10	707.0711	707.0711
7.20	717.0711	717.0711
7.30	727.0711	727.0711
7.40	737.0711	737.0711
7.50	747.0711	747.0711
7.60	757.0711	757.0711
7.70	767.0711	767.0711
7.80	777.0711	777.0711
7.90	787.0711	787.0711
8.00	797.0711	797.0711
8.10	807.0711	807.0711
8.20	817.0711	817.0711
8.30	827.0711	827.0711
8.40	837.0711	837.0711
8.50	847.0711	847.0711
8.60	857.0711	857.0711
8.70	867.0711	867.0711
8.80	877.0711	877.0711
8.90	887.0711	887.0711
9.00	897.0711	897.0711
9.10	907.0711	907.0711
9.20	917.0711	917.0711
9.30	927.0711	927.0711
9.40	937.0711	937.0711
9.50	947.0711	947.0711
9.60	957.0711	957.0711
9.70	967.0711	967.0711
9.80	977.0711	977.0711
9.90	987.0711	987.0711
10.00	997.0711	997.0711

Appendix C The relation between P and Φ.

P	P2	Phi
0	0	0
10	100	500
20	400	3500
30	900	8500
40	1600	15500
50	2500	24500
60	3600	35500
70	4900	48500
80	6400	63500
90	8100	80500
100	10000	99500
110	12100	120500
120	14400	143500
130	16900	168500
140	19600	195500
150	22500	224500
160	25600	255500
170	28900	288500
180	32400	323500
190	36100	360500
200	40000	399500
210	44100	440500
220	48400	483500
230	52900	528500
240	57600	575500
250	62500	624500
260	67600	675500
270	72900	728500
280	78400	783500
290	84100	840500
300	90000	899500
310	96100	960500
320	102400	1023500
330	108900	1088500
340	115600	1155500
350	122500	1224500
360	129600	1295500
370	136900	1368500
380	144400	1443500
390	152100	1520500
400	160000	1599500
410	168100	1680500
420	176400	1763500
430	184900	1848500
440	193600	1935500
450	202500	2024500
460	211600	2115500
470	220900	2208500
480	230400	2303500
490	240100	2400500
500	250000	2499500

P	P2	Phi
510	260100	2600500
520	270400	2703500
530	280900	2808500
540	291600	2915500
550	302500	3024500
560	313600	3135500
570	324900	3248500
580	336400	3363500
590	348100	3480500
600	360000	3599500
610	372100	3720500
620	384400	3843500
630	396900	3968500
640	409600	4095500
650	422500	4224500
660	435600	4355500
670	448900	4488500
680	462400	4623500
690	476100	4760500
700	490000	4899500
710	504100	5040500
720	518400	5183500
730	532900	5328500
740	547600	5475500
750	562500	5624500
760	577600	5775500
770	592900	5928500
780	608400	6083500
790	624100	6240500
800	640000	6399500
810	656100	6560500
820	672400	6723500
830	688900	6888500
840	705600	7055500
850	722500	7224500
860	739600	7395500
870	756900	7568500
880	774400	7743500
890	792100	7920500
900	810000	8099500
910	828100	8280500
920	846400	8463500
930	864900	8648500
940	883600	8835500
950	902500	9024500
960	921600	9215500
970	940900	9408500
980	960400	9603500
990	980100	9800500
1000	1000000	9999500

CURRICULUM VITAE

Name: Mr. Khomsak Chamnakyut

Date of Birth: September 3, 1979

Nationality: Thai

University Education:

1998-2002 Bachelor Degree of Engineering in Chemical Engineering, Faculty of Engineering, Thammasat University, Bangkok, Thailand.