Global adjustments of Geoid Data   MRN 017   (1)

let there be N tracks

the i-th track has a offset correction c_i

~~the i-th track~~

let there be M track intersections. Each intersection has two tracks. for the j-th intersection these are tracks L_j and R_j (L="left" R="right") order them so L has the smaller track number)

the geoid height ~~diff~~ error at the j-th intersection is

$$E_j = [g^L_j + c_{L_j}] - [g^R_j + c_{R_j}] \quad ; \; j = 1, M$$

we want this error to be zero.

$$E_j \approx 0 \quad \text{so} \quad [c_{L_j} - c_{R_j}] = [g^R_j - g^L_j] \quad ; \; j = 1, M$$

write these as a matrix equation $A_{jk} c_k = [g_{R_j} - g_{L_j}] = y_j$

$; \; k = 1, N$

$$A_{jk} = \delta_{kL_j} - \delta_{kR_j}$$

$$[A^T A]_{rs} = A_{ir} A_{is} = [\delta_{rL_i} - \delta_{rR_i}][\delta_{sL_i} - \delta_{sR_i}]$$

$$= \delta_{rL_i}\delta_{sL_i} - \delta_{rL_i}\delta_{sR_i} - \delta_{rR_i}\delta_{sL_i} + \delta_{rR_i}\delta_{sR_i}$$

note $[A^T A]$ has rank < N since adding constant to c_i's doesn't change residuals. perhaps damped least squares best soln?

note L_i never equal R_i since track intersections always 2 different tracks

case 1 on diagonal elements $r=s$

$$[A^TA]_{rr} = \sum_i \{ \delta_{rL_i}\delta_{rL_i} - \delta_{rL_i}\delta_{rR_i} - \delta_{rR_i}\delta_{rL_i} + \delta_{rR_i}\delta_{rR_i} \}$$

contributes whenever the left track is r — zero since left and right track never both r — contributes whenever the right track is r

now suppose the r-th track is intersected p times. then for each of these intersections the track will appear in L_p or R_p but not both. Therefore:

$$[A^TA]_{rr} = p = \text{number of times the } r\text{-th track is intersected}$$

case 2 off diagonal elements $r \neq s$

$$[A^TA]_{rs} = \sum_i \{ \delta_{rL_i}\delta_{sL_i} - \delta_{rL_i}\delta_{sR_i} - \delta_{rR_i}\delta_{sL_i} + \delta_{rR_i}\delta_{sR_i} \}$$

zero since $r \neq s$ — contributes whenever $L_i=r$ and $R_i=s$ — contributes whenever $R_i=r$ and $L_i=s$ — zero since $r \neq s$

suppose tracks r and s intersect q times. at each of these intersections either ($L_q=r$ and $R_q=s$) or ($L_q=s$ and $R_q=r$). therefore

$$[A^TA]_{rs} = -q = -1 \times \text{the number of times tracks } r \text{ and } s \text{ intersect.} \quad (r \neq s)$$

$$[A^T y]_r = A_{ir} y_i = [\delta_{rL_i} - \delta_{rR_i}]\, y_i$$

since $L_i \neq R_i$ this contributes a -1 or $+1$. } contributes whenever the i-th intersection lies on the r-th track.

set all $[A^T y]_r$ to zero. now consider each intersection (i) this intersection is the intersections of tracks L_i and R_i. add y_i to $[A^T y]_{r=L_i}$ and subtract it from $[A^T y]_{r=R_i}$.

(STEP 1) prepare table:

intersection number i	number of L track L_i	number of R track R_i	difference in geoid height $y_i = g_{R_i} - g_{L_i}$

(step 2) zero $[A^T A]_{rs}$ and $[A^T y]_r$

(step 3) read table of intersections to build $A^T A$ and $A^T y$

ADD 1 to $[A^T A]_{r=L_i, s=L_i}$

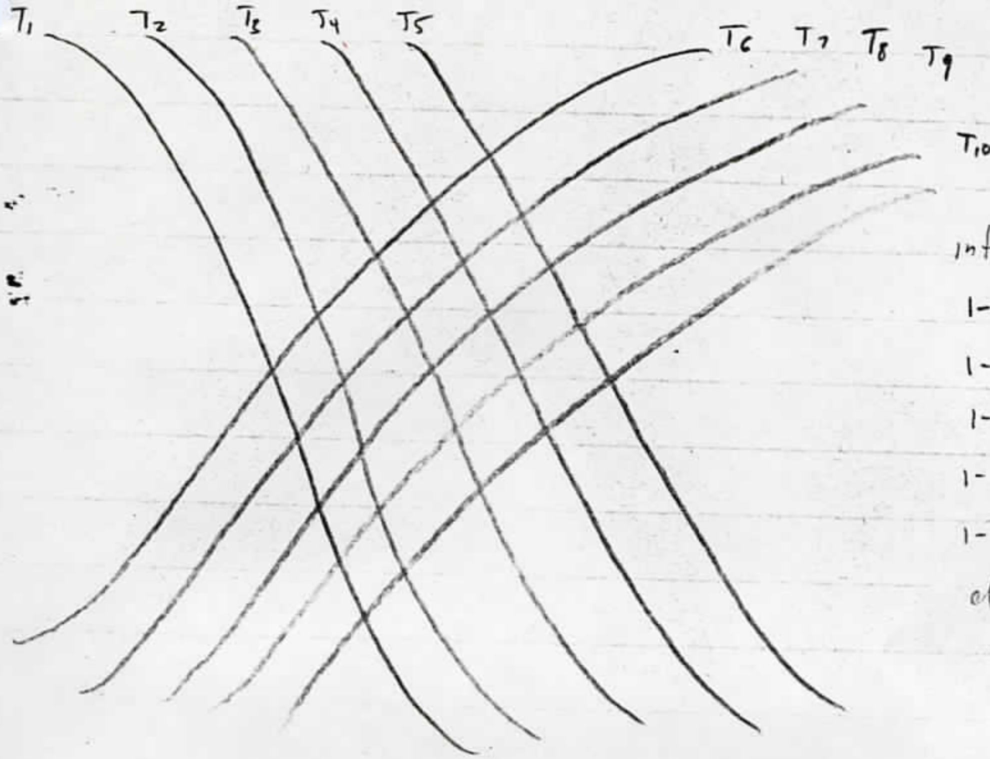
ADD 1 to $[A^T A]_{r=R_i, s=R_i}$

subtract 1 from $[A^T A]_{r=L_i, s=R_i}$

subtract 1 from $[A^T A]_{r=R_i, L_i=s_i}$

Add y_i to $[A^T y]_{r=L_i}$

subtract y_i from $[A^T y]_{r=R_i}$



intersections

1-6

1-7

1-8

1-9

1-10

etc

$$A^TA = \begin{pmatrix} 5 & 0 & 0 & 0 & 0 & -1 & -1 & -1 & -1 & -1 \\ & 5 & 0 & 0 & 0 & -1 & -1 & -1 & -1 & -1 \\ & & 5 & 0 & 0 & -1 & -1 & -1 & -1 & -1 \\ & & & 5 & 0 & -1 & -1 & -1 & -1 & -1 \\ & & & & 5 & -1 & -1 & -1 & -1 & -1 \\ & & & & & 5 & 0 & 0 & 0 & 0 \\ & & & & & & 8 & 0 & 0 & 0 \\ & & & & & & & 5 & 0 & 0 \\ & & & & & & & & 5 & 0 \\ & & & & & & & & & 5 \end{pmatrix}$$

```
c program to do global adjustment of profiles by minimizing cross-over
c errors. w menke dec/81
c this version a test program with 4 ascending and 4 descending tracks
      dimension itl(100), itr(100), gl(100), gr(100), y(100)
c itl, itr ... i-th crossover has ascending track number itl(i) and
c               descending itr(i)
c gl, gr ..... i-th crossover. ascending track geoid height gl(i)
c               descending gr(i)
c y .......... crossover error at i-th intersection is y(i)
      dimension a(8,8), b(8), z(8), zn(8)
c a, b, z .... normal equations are A*z=b where z are offset corrections
      dimension ap(8,8), bp(8)
c ap, bp .... copy of normal equations, since one is destroyed during
c               solution by Gauss-Jordan reduction
      character*40 title

      nstore=8
      iprnt=6
c     data in this file
      open(1,file='test.in',status='old')
      rewind(1)
      read(1,"(i10)") n
      write(6,"('data for each crossover:')")
      write(6,"(' asc #    desc #     asc geoid      desc geoid')")
      do 10 i=1,100
              read(1,"(2i10,2f20.0)",end=11) itl(i),itr(i),gl(i),gr(i)
              write(6,"(2i10,2f20.0)") itl(i),itr(i),gl(i),gr(i)
              y(i) = gr(i)-gl(i)
10            continue
11    continue
      m=i-1
      write(6,"(i8,' intersections read')")

c     zero normal equations prior to building them
      do 15 i=1,n
              b(i)=0.0
              do 14 j=1,n
                      a(i,j)=0.0
14                    continue
15            continue

c     build normal equations, crossover by crossover
      do 20 i=1,m
              a(itl(i),itl(i)) = a(itl(i),itl(i)) + 1.0
              a(itr(i),itr(i)) = a(itr(i),itr(i)) + 1.0
              a(itl(i),itr(i)) = a(itl(i),itr(i)) - 1.0
              a(itr(i),itl(i)) = a(itr(i),itl(i)) - 1.0
              b(itl(i)) = b(itl(i)) + y(i)
              b(itr(i)) = b(itr(i)) - y(i)
20            continue

      write(6,"('normal equation matrix:')")
      do 25 i=1,n
              write(6,"(8f4.1)") (a(i,j),j=1,n)
25            continue
      write(6,"('normal equation vector')")
      do 30 i=1,n
              write(6,"(f10.5)") b(i)
30            continue

      write(6,"('enter sigma')")
       d(5,"(f20.0)") sigma
       te(6,"('damping coefficient is',e13.5)") sigma
c add damping coefficient to diagonal of normal equation matrix
```

```
      do 35 i=1,n
            a(i,i) = a(i,i) + sigma
35          continue
c     solve by gauss-jordan reduction
      idet=Ø
      xtest=1.Øe-6
      itriag=Ø
      do 36 i=1,n
            bp(i)=b(i)
            do 36 j=1,n
                  ap(i,j)=a(i,j)
36          continue
      call gauss(ap,bp,n,nstore,det,idet,xtest,ierror,itriag)
      write(6,"('solution by Gauss-Jordan reduction. error status:',i8)")
    *ierror
      write(6,"(8e13.5)") (bp(i),i=1,n)
c compute improvement in variance
      sum=Ø.Ø
      sum1=Ø.Ø
      do 46 i=1,m
            sum = sum + (g1(i)+bp(it1(i))-gr(i)-bp(itr(i)))**2
            sum1 = sum1 + (g1(i)-gr(i))**2
46          continue
      write(6,"('residual sum of squares ',e13.5)") sum
      f = 1ØØ.Ø - 1ØØ.Ø*sum/sum1
      write(6,"('percent improvement',f1Ø.5)") f
c     solution by Gauss-Siedel iteration
      itmax=5
      test=1.e-6
      call gsiter(a,b,z,n,nstore,itmax,itact,test,ierror)
      write(6,"(' G-S iteration, error = ',i8,' iterations = ',i8)")
    * ierror, itact
      write(6,"(8e13.5)") (z(i),i=1,n)
c     compute improvement in variance
      sum=Ø.Ø
      sum1=Ø.Ø
      do 646 i=1,m
            sum = sum + (g1(i)+z(it1(i))-gr(i)-z(itr(i)))**2
            sum1 = sum1 + (g1(i)-gr(i))**2

646         continue
      write(6,"('residual sum of squares ',e13.5)") sum
      f = 1ØØ.Ø - 1ØØ.Ø*sum/sum1
      write(6,"('percent improvement',f1Ø.5)") f
      stop
      end
```

```
      subroutine gauss(a,vec,n,nstore,det,idet,test,ierror,itriag)
c
c a subroutine to solve a system of n linear equations in n
c unknowns, where n doesn't exceed 1ØØ. all parameters real.
c
c gaussian reduction with partial pivoting is used. the
c original matrix a(n,n) is first triangularized. its zero
c elements are replaced by coefficients which allow the trans-
c forming of the vector vec(n).
c
c if itriag is set to 1 before the subroutine call, the tri-
c angularization proceedure is ommitted and a is assumed to
c have been triangularized on a previous subroutine call. after
c the call itriag is automatically set to 1.
c
c the matrix a (nxn) and the vector vec (n) are altered
c during the subroutine call. the solution is returned in vec.
c
c test is a real positive number specified by the user which is
c used in testing for division by near zero numbers. if the absolute
c value of a number is .le. test an error condition
c will result.
c
c the error conditions are returned in ierror. they are :
c                     Ø : no error
c                     1 : division condition violated
c                         during triangularization of a
c                     2 : division condition violated
c                         during back solving
c                     3 : division condition violated
c                         at both places
c
c the determinant of a is calculated during triangularization and
c returned in det, if idet is set to one before the call.
c
c nstore is the size to which a and vec were dimensioned in the maine
c program, whereas n is the size of the used portion of a and vec
c
c warning : if the itriag=1 option is used, the array isub
c must have been preserved from the call which triangularized the
c matrix a
c
      dimension a(nstore,nstore),vec(nstore),line(1ØØ),isub(1ØØ)
c
      n3=n-1
      iet=Ø
      ieb=Ø
c
c ****** triangularize the matrix a, replacing the zero elements
c        of the triangularized matrix with the coefficients needed
c        to transform the vector vec
c
      if(itriag.eq.1) go to 3ØØ
c
      if(idet.eq.1) det=1.Ø
      do 1 j=1,n
1     line(j)=Ø
c
      do 3Ø j=1,n3
      big=Ø.Ø
      do 4Ø 11=1,n
      if(line(11).eq.1) go to 4Ø
      testa=abs(a(11,j))
      if(testa.lt.big) go to 4Ø
```

```
      i=11
      big=testa
4Ø    continue
      if(big.le.test) iet=1
      line(i)=1
      isub(j)=i
      sum=1.Ø/a(i,j)
      if(idet.eq.1) det=det*a(i,j)
      do 1Ø k=1,n
      if(line(k).eq.1) go to 1Ø
      b=a(k,j)*sum
      do 2Ø l=j+1,n
      a(k,l)=a(k,l)-b*a(i,l)
2Ø    continue
      a(k,j)=b
1Ø    continue
3Ø    continue
c
c ****** find the row of the triangularized matrix that has
c        been reduced to having only one non-zero element.
c        this has order number n
c
      do 32 ll=1,n
      if(line(ll).ne.1) go to 35
32    continue
35    isub(n)=ll
      b=a(ll,n)
      if(idet.eq.1) det=det*b
c
c ****** setup array line so that line(l) gives the order number
c        of the i-th row of storage matrix a
c
3ØØ   do 1ØØ i=1,n
1ØØ   line(isub(i))=i
c
c ****** transform the vector into one corresponding to the
c        triangularized matrix
c
      do 32Ø j=1,n3
      b=vec(isub(j))
      do 31Ø k=1,n
      if(line(k).le.j) go to 31Ø
      vec(k)=vec(k)-a(k,j)*b
31Ø   continue
32Ø   continue
c
c ****** back-solve the triangularized equations
c
      b=a(ll,n)
      testa=abs(b)
      if(testa.le.test) ieb=2
      vec(isub(n))=vec(isub(n))/b
      do 5Ø j=n-1,1,-1
      sum=vec(isub(j))
      do 6Ø j2=j+1,n
      sum=sum-vec(isub(j2))*a(isub(j),j2)
6Ø    continue
      b=a(isub(j),j)
      testa=abs(b)
      if(testa.le.test) ieb=2
      vec(isub(j))=sum/b
5Ø    continue
c
c ****** put the solution vector into the proper order
```

```
c
      do 230 i=1,n
      do 210 j=i,n
      if(line(j).eq.i) go to 220
210   continue
220   b=vec(j)
      vec(j)=vec(i)
      vec(i)=b
      line(j)=line(i)
230   continue
c
c ****** set error parameters
c
      ierror=iet+ieb
      itriag=1
      return
      end
        subroutine gsiter( a, b, z, n, nstore, itmax, itact, test, ierror )
        real*4 a(nstore,nstore), b(n), z(n), test
        integer n, nstore, itmax, itact
c
c       subroutine gsiter, by William Menke, January, 1982
c       a simple implementation of Gauss-Seidel iteration for
c       the solution of the linear equations AZ=B.
c       Note that this implementation does no pivoting, so that
c       it will fail if there is a zero on the diagonal of a.
c
c       a       input matrix, n by n is size
c       b       input vector of length n
c       z       solution vector, length n
c       n       size of a, etc
c       nstore  dimensioned size of a
c       itmax   maximum number of iterations
c       itact   actual number of iterations
c       test    iterations stopped if RMS change in z
c               is less than test
c       ierror  error flag. 0=no error 1=zero divide 2=max iteration reached
c
        dimension zn(100)
        test2=float(n)*test**2
c
c       initial guess for z all zeroes
c
        do 1 i=1,n
                z(i)=0.0
1               continue
c
c       each iteration builds a better estimate of z, called vn
c
        do 10 it=1, itmax
                do 2 i=1,n
                        sum = 0.0
                        do 3 j=1,i-1
                                sum=sum+a(i,j)*zn(j)
3                               continue
                        do 4 j=i+1,n
                                sum=sum+a(i,j)*z(j)
4                               continue
                        if( a(i,i).eq.0 ) then
                                ierror=1
                                return
                        end if
                        zn(i) = (-sum+b(i))/a(i,i)
2                       continue
```

```
          sum = Ø.Ø
          do 5 i=1,n
                  sum = sum + (z(i)-zn(i))**2
                  z(i)=zn(i)
5                 continue
          if( sum.le.test2 ) then
                  ierror=Ø
                  itact=it
                  return
          end if
1Ø        continue
      ierror=2
      itact=itmax
      return
      end
```

% Reduction = 98.3%

```
    4         8                   2.ØØØ                2.ØØØ
normal eq  ion matrix:
 4.Ø  .Ø  .Ø  .Ø-1.Ø-1.Ø-1.Ø-1.Ø
  .Ø 4.Ø  .Ø  .Ø-1.Ø-1.Ø-1.Ø-1.Ø
  .Ø  .Ø 4.Ø  .Ø-1.Ø-1.Ø-1.Ø-1.Ø
  .Ø  .Ø  .Ø 4.Ø-1.Ø-1.Ø-1.Ø-1.Ø
-1.Ø-1.Ø-1.Ø-1.Ø 4.Ø  .Ø  .Ø  .Ø
-1.Ø-1.Ø-1.Ø-1.Ø  .Ø 4.Ø  .Ø  .Ø
-1.Ø-1.Ø-1.Ø-1.Ø  .Ø  .Ø 4.Ø  .Ø
-1.Ø-1.Ø-1.Ø-1.Ø  .Ø  .Ø  .Ø 4.Ø
normal equation vector
   4.4ØØØØ
  -3.7ØØØØ
   4.2ØØØØ
  -3.9ØØØØ
   3.1ØØØØ
  -4.6ØØØØ
   4.5ØØØØ
  -4.ØØØØØ
enter sigma
damping coefficient is   .1ØØØØe+ØØ
solution by Gauss-Jordan reduction. error status:      Ø
   .1Ø431e+Ø1  -.93255e+ØØ   .99428e+ØØ  -.98133e+ØØ   .78621e+ØØ  -.1Ø918e+Ø1   .11277e+Ø1  -.9455Øe+ØØ
residual sum of squares    .5522Øe+ØØ
percent improvement  98.36141
 G-S iteration, error =        2 iterations =           2
   .1Ø717e+Ø1  -.9Ø389e+ØØ   .1Ø229e+Ø1  -.95267e+ØØ   .81417e+ØØ  -.1Ø639e+Ø1   .11556e+Ø1  -.91754e+ØØ
residual sum of squares    .55219e+ØØ
percent improvement  98.36144
data for each crossover:
 asc #     desc #     asc geoid     desc geoid
        1         5             1.ØØØ                1.4ØØ
        1         6              .ØØØ                2.ØØØ
        1         7            -1.ØØØ               -1.ØØØ
        1         8             2.ØØØ                4.ØØØ
        2         5             2.ØØØ                 .3ØØ
        2         6             1.ØØØ                1.ØØØ
        2         7             2.ØØØ                 .ØØØ
        2         8             4.ØØØ                4.ØØØ
        3         5             3.ØØØ                3.2ØØ
        3         6             1.ØØØ                3.ØØØ
        3         7            -1.ØØØ               -1.ØØØ
        3         8              .ØØØ                2.ØØØ
        4         5             4.ØØØ                2.ØØØ
        4         6             2.ØØØ                2.6ØØ
        4         7             1.ØØØ               -1.5ØØ
        4         8             2.ØØØ                2.ØØØ

normal equation matrix:
 4.Ø  .Ø  .Ø  .Ø-1.Ø-1.Ø-1.Ø-1.Ø
  .Ø 4.Ø  .Ø  .Ø-1.Ø-1.Ø-1.Ø-1.Ø
  .Ø  .Ø 4.Ø  .Ø-1.Ø-1.Ø-1.Ø-1.Ø
  .Ø  .Ø  .Ø 4.Ø-1.Ø-1.Ø-1.Ø-1.Ø
-1.Ø-1.Ø-1.Ø-1.Ø 4.Ø  .Ø  .Ø  .Ø
-1.Ø-1.Ø-1.Ø-1.Ø  .Ø 4.Ø  .Ø  .Ø
-1.Ø-1.Ø-1.Ø-1.Ø  .Ø  .Ø 4.Ø  .Ø
-1.Ø-1.Ø-1.Ø-1.Ø  .Ø  .Ø  .Ø 4.Ø
normal equation vector
   4.4ØØØØ
  -3.7ØØØØ
   4.2ØØØØ
  -3.9ØØØØ
   3.1ØØØØ
```

Damping = 1.0

% Reduction = 94.5%

069

```
data for each crossover:
 asc #     desc #      asc geoid      desc geoid
         1         5              1.000                1.400
         1         6               .000                2.000
         1         7             -1.000               -1.000
         1         8              2.000                4.000
         2         5              2.000                 .300
         2         6              1.000                1.000
         2         7              2.000                 .000
         2         8              4.000                4.000
         3         5              3.000                3.200
         3         6              1.000                3.000
         3         7             -1.000               -1.000
         3         8               .000                2.000
         4         5              4.000                2.000
         4         6              2.000                2.600
         4         7              1.000               -1.500
         4         8              2.000                2.000

normal equation matrix:
 4.0  .0  .0  .0-1.0-1.0-1.0-1.0
  .0 4.0  .0  .0-1.0-1.0-1.0-1.0
  .0  .0 4.0  .0-1.0-1.0-1.0-1.0
  .0  .0  .0 4.0-1.0-1.0-1.0-1.0
-1.0-1.0-1.0-1.0 4.0  .0  .0  .0
-1.0-1.0-1.0-1.0  .0 4.0  .0  .0
-1.0-1.0-1.0-1.0  .0  .0 4.0  .0
-1.0-1.0-1.0-1.0  .0  .0  .0 4.0
normal equation vector
   4.40000
  -3.70000
   4.20000
  -3.90000
   3.10000
  -4.60000
   4.50000
  -4.00000
enter sigma
damping coefficient is   .10000e-01
solution by Gauss-Jordan reduction. error status:        0
   .10661e+01  -.95382e+00   .10163e+01  -.10037e+01   .80421e+00  -.11160e+01   .11533e+01  -.96637e+00
residual sum of squares    .53271e+00
percent improvement  98.41927
 G-S iteration, error =        2 iterations =         2
   .10971e+01  -.92285e+00   .10472e+01  -.97272e+00   .83510e+00  -.10851e+01   .11842e+01  -.93547e+00
residual sum of squares    .53271e+00
percent improvement  98.41927
data for each crossover:
 asc #     desc #      asc geoid      desc geoid
         1         5              1.000                1.400
         1         6               .000                2.000
         1         7             -1.000               -1.000
         1         8              2.000                4.000
         2         5              2.000                 .300
         2         6              1.000                1.000
         2         7              2.000                 .000
         2         8              4.000                4.000
         3         5              3.000                3.200
         3         6              1.000                3.000
         3         7             -1.000               -1.000
         3         8               .000                2.000
                   5              4.000                2.000
                   6              2.000                2.600
         4         7              1.000               -1.500
```

TRIAL RUNS

Damping = 0.01

% reduction 98.4%

Damping = 0.1

689

```
  -4.60000
   4.50000
  -4.00000
enter sigma
damping coefficient is    .10000e+01
solution by Gauss-Jordan reduction. error status:          0
    .85778e+00  -.76222e+00   .81778e+00  -.80222e+00   .64222e+00  -.89778e+00   .92222e+00  -.77778e+00
residual sum of squares    .18575e+01
percent improvement  94.48821
 G-S iteration, error =       2 iterations =          2
    .87200e+00  -.74800e+00   .83200e+00  -.78800e+00   .65360e+00  -.88640e+00   .93360e+00  -.76640e+00
residual sum of squares    .18570e+01
percent improvement  94.48971
```