CHEBYSHEVLJEVO RJEŠENJE SUSTAVA LINEARNIH JEDNADŽBI PREKOBROJNIH PODATAKA

izradila: ***Margareta Devčić***

Da bi se moglo razumjeti Chebyshevljevo rješenje sustava linearnih jednadžbi prekobrojnih podataka, trebaju se prvo objasniti pojmovi "linearno programiranje", "simpleks metoda", "sustavi prekobrojnih podataka".

Linearno programiranje

Pojam "linearno programiranje" obrađen je u području numeričke analize, dakle dio je matematike, međutim svoju primjenu našao je i u geodeziji. Riječ "programiranje", koja se u literaruri obično navodi vezano uz ovo područje, nije nužno povezana sa komputorskim programiranjem, već se upotrebljava u smislu "planiranje".

Problem linearnog programiranja zahtijeva da ciljna funkcija ili funkcija kriterija

$$H = c_1x_1 + \dots + c_nx_n$$

bude minimalizirana ili maksimalizirana ovisno o skupu ograničavajućih uvjeta te o uvjetu nenegativnosti

$$x_j \geq 0$$

gdje je i = 1, . . . , m, te j = 1, . . . , n. U matričnom obliku problem se može napisati kao:

$$H(x) = c^T x = \text{minimum}$$
$$Ax \leq b$$
$$x \geq 0$$

Teorem linearnog programiranja navodi da se traženi minimum ili maksimum javlja u ekstremnmoj mogućoj točki.

Simpleks metoda

Simpleks metoda koristi se za rješavanje problema linearnog programiranja. Poslije njenog otkrića (Dantzig, oko 1940-e god.) simpleks metoda je bila bez konkurencije, sve do kasnih 1980-ih, zbog svoje prednosti u rješavanju praktičnih problema linearnog programiranja. Razvojem praktične primjene linearnog programiranja u zadnjih dvadesetak godina, te napredovanjem kompjutorske tehnologije, ono je našlo primjenu u geodeziji.

Budući da se rješenje pojavljuje u ekstremnoj mogućoj točki, primjena simpleks metode počinje u takvoj točki. Računa se vrijednost funkcije H. Zatim se zamijenjuje ta ekstremna točka za slijedeću na drugom kraju ruba i to tako da se postigne manja (u slučaju minimiziranja) vrijednost funkcije H. Ovaj proces izmjene ekstremnih točaka se nastavlja sve dok se vrijednost funkcije H više ne može smanjiti.

Sustavi prekobrojnih podataka (mjerenja)

Sustav linearnih jednadžbi prekobrojnih podataka Ax = b je onaj kod kojeg matrica A ima više redova nego stupaca, tj. broj jednadžbi "m" je veći od broja nepoznanica "n". Uobičajeno bi bilo da vektor rješenja x ne postoji, te da je takav sustav linearnih jednadžbi besmislen. Takav sustav se još naziva nedosljedan (nepostojan). Sustavi prekobrojnih podataka se pojavljuju uvijek kada postoji više rezultata nego što bi se zahtjevalo za određenu preciznost. Međutim, mnoštvo netočnih, proturječnih informacija postaje zamjena za nekoliko perfektnih rezultata, te se iz tih proturječja može izvući dobra aproksimacija točnih rezultata. Postoje dvije metode rješavanja prekobrojnih sustava linearnih jednadžbi. To su rješenje najmanjih kvadrata i Chebyshevljevo rješenje. Ovdje će biti opisano Chebyshevljevo rješenje. Objema metodama zajednički je vektor ostataka R = Ax – b. Budući da se vektor R ne može svesti na nul-vektor, nastoji se odabrati vektor rješenja x tako da se R minimalizira.

CHEBYSHEVLJEVO RJEŠENJE

Sustav linearnih jednadžbi prekobrojnih podataka Ax = b može se napisati u obliku:

$$\sum_{j=1}^{n} a_{i,j} x_j = b_i,$$

gdje je i=1, . . . ,m, te j=1, . . . ,n, a vrijedi m$\geq$n. Chebyshevljevo ($l\infty$ ili mini-maks) rješenje mxn sustava linearnih jednadžbi prekobrojnih podataka određuje vektor x koji minimalizira maksimalnu apsolutnu vrijednost ostataka

$$r = \max_{1 \leq i \leq m} \left| b_i - \sum_{j=1}^{n} a_{i,j} x_j \right|.$$

U geodeziji ovi ostaci se mogu poistovjetiti sa popravkama. Polazi se od sustava jednadžbi popravaka (svakom mjerenju pripada a priori poznata težina)

$$v = A\,x - l$$

Neka je "v" ona popravka od koje ne postoji po apsolutnoj vrijednosti veća popravka, tj.

$$|v_i| \leq v, \qquad i=1, \dots, n$$

Dakle bit ce:

$$v \ \min$$

Opis algoritma u Fortranu

Ovdje se opisuje algoritam koji rješava sustav linearnih jednadžbi prekobrojnih podataka u Chebyshevljevoj normi. Autori ovog algoritma su I. Barrodale i C. Phillips. Algoritam je objavljen u časopisu CACM (Communications of the ACM) 1975.god. u obliku potprograma, a ovdje je prilagođen geodetskim potrebama. Predstavlja modifikaciju simpleks metode linearnog programiranja.

Približno rješenje određeno je tokom prvih (k+1) iteracija, gdje "k" označava rang matrice A. U slijedećim iteracijama ovo rješenje se poboljšava, sve dok se ne dobije Chebyshevljevo rješenje x koje minimalizira maksimalnu apsolutnu vrijednost popravaka ili dok se ne pronađe približno rješenje x* za koje je:

$$\frac{v^{*} - v}{v} < \text{RELLER},$$

gdje je RELLER realna varijabla koju zadaje korisnik. Parametar M označava broj jednadžbi, a N broj nepoznanica. Simpleks iteracije izvršavaju se na matrici *A* dimenzija (NDIM, MDIM), gdje je NDIM=N+3, a MDIM=M+1. Transponirana matrica A se pohranjuje u prvih N redaka i M stupaca matrice *A*. Transponirani vektor jedinica e^T dimenzija (1,N) pohranjen je u (N+1)–i redak matrice *A*. Transponirani vektor b sa suprotnim predznakom, dimenzija (1,N) pohranjen je u (N+2)–i redak matrice *A*. Ovi inicijalni koeficijenti matrice *A* kasnije se unište. TOL je realna pozitivna varijabla koja se treba postaviti na malu pozitivnu vrijednost. Program ne može razlikovati nulu od bilo koje vrijednosti čija veličina nije veća od TOL. Dakle, program se neće vrtjeti niti oko jednog broja čija je veličina manja od TOL, te može završiti prerano ako se TOL loše zada. Parametar RELLER se postavlja na 0.0, ukoliko se traži Chebyshevljevo rješenje x. Ako se traži približno rješenje x* koje zadovoljava gornju nejednadžbu, RELLER se postavi na pozitivnu vrijednost (npr. 0.1). Uobičajeni efekt ove mogućnosti je smanjenje broja traženih simpleks iteracija. RESMAX predstavlja apsolutnu vrijednost najvećeg popravka, pa je na izlazu označen sa "| v | max". ITER je broj iteracija obavljenih simpleks metodom. OCODE je izlazni kod, može biti sa vrijednošću 1, ako je rješenje optimalno i jedinstveno, sa vrijednošću 2, ako su računanja okončana prerano (to je u slučaju kada program naiđe na veličinu čija je vrijednost manja od TOL, tada se sve izlazne informacije odnose na zadnju obavljenu iteraciju), te sa vrijednošću 0, ako je rješenje optimalno, ali nije jedinstveno.

Sam program kompajliran je i linkan pomoću Microsoft-ovog Fortrana, verzija 5.0. Unose se matrica A^T i vektor l, a ako postoje težine, unosi se P. Pri samom unosu ulaznih podataka može se birati da li se želi unositi elemente matrica A, l i P (program tada stvara datoteke a.dat, l.dat i p.dat) ili se želi učitati A, l i P iz već postojećih datoteka a.dat, l.dat i p.dat (u slučaju da se u unosu podataka pogriješilo, te se editiranjem popravilo pogrešku u a.dat, l.dat ili p.dat). Također se pri unosu ulaznih podatka određuje da li postoje težine. Program stvara datoteku x.dat, u kojoj se nalazi rješenje, te datoteku v.dat, u kojoj se nalazi vektor popravaka. Veličina parametra TOL pri njegovom zadavanju ne smije biti manja od .1E-07, te se i zadaje u obliku .1E-.Osim varijable TOL, svi podaci unose se na tri decimale. Pri unosu elemenata matrice A, l i P treba obavezno uvijek upisati decimalnu točku (npr. pogrešno bi bilo unijeti 1,već 1.).

U programu se koristi postupak pivotiranja. Pivot element $x_{p,q}$ uvijek je različit od nule,a pivot jednadžbe su :

$$x'_{p,j} = \frac{x_{p,j}}{x_{p,q}}, \qquad \text{za } i=p$$

$$x'_{p,j} = x_{i,j} - x'_{p,j} \cdot x_{i,q}, \qquad \text{za } i \neq p$$

gdje je $1 \leq p \leq m$ i $(m+1) \leq q \leq n$.

ALGORITAM U FORTRANU

```
      PROGRAM CHEBYSHEV
       IMPLICIT DOUBLE PRECISION
(A-Y)
       INTEGER
PROW,PCOL,RANK,RANKP1,OCODE,I,J,MM1,K,KP1,MP1,NP1,NP2,NP3
       INTEGER
MODE,NP1MK,LEV,NP1MR,M,N,MDIM,NDIM,ITER,S,R
       REAL*4
BIG,D,DD,PIVOT,RELIMP,VAL,TPIVOT,RELERR,TOL,RESMAX
       DIMENSION
A(100,100),B(100),X(100),P(100)
       DATA BIG/1.E+38/
     1 FORMAT(A30,$,/,/)
       WRITE(*,1)'PROGRAM
CHEBYSHEV'
       WRITE(*,3)'BROJ JEDNADZBI
JE=  '
     3 FORMAT(A30,$)
       READ(*,'(I4)') M
       WRITE(*,3)'BROJ NEPOZNANICA
JE=  '
       READ(*,'(I4)') N
       WRITE(*,3)'TOL=  '
       READ(*,'(E12.3)') TOL
       WRITE(*,3)'RELERR=  '
       READ(*,'(E12.3)') RELERR
       WRITE(*,3)'DIALOG
INPUT(0=DA,1=NE) '
       READ(*,'(I1)') S
       WRITE(*,3)'TEZINE(0=DA,1=NE)
'
       READ(*,'(I1)') R
     4 FORMAT(1X,'A','(',I2,',',\)
     5 FORMAT(1X,I2,')','=',\)
       OPEN(UNIT=3,FILE='A.DAT')
       DO 7 I=1,N
         DO 6 J=1,M
           IF (S.EQ.0) THEN
           WRITE(*,4)I
           WRITE(*,5)J
           READ(*,'(F12.4)')A(I,J)
           WRITE(3,'(F12.4)')A(I,J)
           ENDIF
           IF (S.EQ.1) THEN
           READ(3,'(F12.4)')A(I,J)
           ENDIF
     6 CONTINUE
     7 CONTINUE
       OPEN(UNIT=5,FILE='L.DAT')
       DO 8 J=1,M
         IF (S.EQ.0) THEN
         WRITE(*,9)J
         READ(*,'(F12.4)')B(J)
         WRITE(5,'(F12.4)')B(J)
         ENDIF
         IF (S.EQ.1) THEN
         READ(5,'(F12.4)')B(J)
         ENDIF
     8 CONTINUE
     9
FORMAT(1X,'l','(',I2,')',\,'=')
       IF (R.EQ.0) THEN
       OPEN(UNIT=6,FILE='P.DAT')
       DO 10 J=1,M
         IF (S.EQ.0) THEN
         WRITE(*,11)J
         READ(*,'(F12.4)')P(J)
         WRITE(6,'(F12.4)')P(J)
         ENDIF
         IF (S.EQ.1) THEN
         READ(6,'(F12.4)')P(J)
         ENDIF
    10 CONTINUE
    11
FORMAT(1X,'P','(',I2,')',\,'=')
       DO 13 I=1,N
         DO 12 J=1,M
           A(I,J)=A(I,J)*P(J)
    12 CONTINUE
    13 CONTINUE
       DO 15 J=1,M
         B(J)=B(J)*P(J)
    15 CONTINUE
       ENDIF
C  INICIJALIZACIJA
       MP1=M+1
       MDIM=MP1
       NP1=N+1
       NP2=N+2
       NP3=N+3
       NDIM=NP3
       NP1MR=1
       RANK=N
       RELTMP=RELERR
       RELERR=0.
       DO 19 J=1,M
         A(NP1,J)=1.
         A(NP2,J)=-B(J)
         A(NP3,J)=N+J
    19 CONTINUE
       A(NP1,MP1)=0.
       ITER=0
       OCODE=1
       DO 20 I=1,N
         X(I)=0.
         A(I,MP1)=I
```

```
   20 CONTINUE
C NIVO 1.
      LEV=1
      K=0
   30 K=K+1
      KP1=K+1
      NP1MK=NP1-K
      MODE=0
      DO 40 J=K,M
        B(J)=1.
   40 CONTINUE
C ODREDI VECTOR ZA ULAZ U BAZU
   50 D=-BIG
      DO 60 J=K,M
        IF (B(J).EQ.0.) GO TO 60
        DD=ABS(A(NP2,J))
        IF (DD.LE.D) GO TO 60
        PCOL=J
        D=DD
   60 CONTINUE
      IF (K.GT.1) GO TO 70
C TEST ZA NULE PO PRAVILU DESNE
RUKE
      IF (D.GT.TOL) GO TO 70
      RESMAX=0.
      MODE=2
      GO TO 380
C ODREDI VEKTOR ZA IZLAZ IZ BAZE
   70 D=TOL
      DO 80 I=1,NP1MK
      DD=ABS(A(I,PCOL))
      IF (DD.LE.D) GO TO 80
      PROW=I
      D=DD
   80 CONTINUE
      IF (D.GT.TOL) GO TO 330
C PROVJERI LINEARNU OVISNOST U
NIVOU 1.
      B(PCOL)=0.
      IF (MODE.EQ.1) GO TO 50
      DO 100 J=K,M
        IF (B(J).EQ.0.) GO TO 100
        DO 90 I=1,NP1MK
          IF (ABS(A(I,J)).LE.TOL)
GO TO 90
          MODE=1
          GO TO 50
   90 CONTINUE
  100 CONTINUE
      RANK=K-1
      NP1MR=NP1-RANK
      OCODE=0
      GO TO 160
  110 IF (PCOL.EQ.K) GO TO 130
C ZAMJENA STUPACA U NIVOU 1.
      DO 120 I=1,NP3
        D=A(I,PCOL)
        A(I,PCOL)=A(I,K)
        A(I,K)=D
  120 CONTINUE
  130 IF (PROW.EQ.NP1MK) GO TO 150
C ZAMJENA REDAKA U NIVOU 1.
      DO 140 J=1,MP1
        D=A(PROW,J)
        A(PROW,J)=A(NP1MK,J)
        A(NP1MK,J)=D
  140 CONTINUE
  150 IF (K.LT.N) GO TO 30
  160 IF (RANK.EQ.M) GO TO 380
      RANKP1=RANK+1
C NIVO 2.
      LEV=2
C ODREDI VEKTOR ZA ULAZ U BAZU
      D=TOL
      DO 170 J=RANKP1,M
        DD=ABS(A(NP2,J))
        IF (DD.LE.D) GO TO 170
        PCOL=J
        D=DD
  170 CONTINUE
C USPOREDI CHEBISHEVU POGRESKU SA
TOL
      IF (D.GT.TOL) GO TO 180
      RESMAX=0
      MODE=3
      GO TO 380
  180 IF (A(NP2,PCOL).LT.-TOL) GO
TO 200
      A(NP1,PCOL)=2.-A(NP1,PCOL)
      DO 190 I=NP1MR,NP3
        IF (I.EQ.NP1) GO TO 190
        A(I,PCOL)=-A(I,PCOL)
  190 CONTINUE
C PRIPREMI SVE ULAZE U PIVOT
STUPAC
C (OSIM PIVOT) DA BUDU NEGATIVNI
  200 DO 220 I=NP1MR,N
        IF (A(I,PCOL).LT.TOL) GO
TO 220
        DO 210 J=1,M
        A(NP1,J)=A(NP1,J)+2.*A(I,J)
          A(I,J)=-A(I,J)
  210 CONTINUE
        A(I,MP1)=-A(I,MP1)
  220 CONTINUE
      PROW=NP1
      GO TO 330
  230 IF (RANKP1.EQ.M) GO TO 380
      IF (PCOL.EQ.M) GO TO 250
C ZAMJENA STUPACA U NIVOU 2.
      DO 240 I=NP1MR,NP3
        D=A(I,PCOL)
        A(I,PCOL)=A(I,M)
        A(I,M)=D
  240 CONTINUE
  250 MM1=M-1
C NIVO 3.
      LEV=3
C ODREDI VEKTOR ZA ULAZ U BAZU
  260 D=-TOL
      VAL=2.*A(NP2,M)
      DO 280 J=RANKP1,MM1
        IF (A(NP2,J).GE.D) GO TO
270
        PCOL=J
        D=A(NP2,J)
        MODE=0
        GO TO 280
  270   DD=VAL-A(NP2,J)
        IF (DD.GE.D) GO TO 280
        MODE=1
        PCOL=J
        D=DD
  280 CONTINUE
      IF (D.GE.-TOL) GO TO 380
      DD=-D/A(NP2,M)
      IF (DD.GE.RELTMP) GO TO 290
      RELERR=DD
      MODE=4
      GO TO 380
  290 IF (MODE.EQ.0) GO TO 310
      DO 300 I=NP1MR,NP1
        A(I,PCOL)=2.*A(I,M)-
A(I,PCOL)
  300 CONTINUE
      A(NP2,PCOL)=D
      A(NP3,PCOL)=-A(NP3,PCOL)
C ODREDI VEKTOR ZA IZLAZ IZ BAZE
  310 D=BIG
      DO 320 I=NP1MR,NP1
        IF (A(I,PCOL).LE.TOL) GO
TO 320
        DD=A(I,M)/A(I,PCOL)
        IF (DD.GE.D) GO TO 320
        PROW=I
        D=DD
  320 CONTINUE
      IF (D.LT.BIG) GO TO 330
      OCODE = 2
      GO TO 380
C PIVOTIRANJE PO A(PROW,PCOL)
  330 PIVOT = A(PROW,PCOL)
      DO 340 J=1,M
        A(PROW,J)=A(PROW,J)/PIVOT
  340 CONTINUE
      DO 360 J=1,M
        IF (J.EQ.PCOL) GO TO 360
        D=A(PROW,J)
        DO 350 I=NP1MR,NP2
          IF (I.EQ.PROW) GO TO 350
          A(I,J)=A(I,J)-
D*A(I,PCOL)
  350   CONTINUE
  360 CONTINUE
      TPIVOT=-PIVOT
      DO 370 I=NP1MR,NP2
        A(I,PCOL)=A(I,PCOL)/TPIVOT
  370 CONTINUE
      A(PROW,PCOL)=1./PIVOT
      D=A(PROW,MP1)
      A(PROW,MP1)=A(NP3,PCOL)
      A(NP3,PCOL)=D
      ITER=ITER+1
      GO TO (110,230,260),LEV
C PRIPREMI IZLAZ
  380 DO 390 J=1,M
        B(J)=0.
  390 CONTINUE
      IF (MODE.EQ.2) GO TO 450
      DO 400 J=1,RANK
        K=A(NP3,J)
        X(K)=A(NP2,J)
  400 CONTINUE
      IF (MODE.EQ.3 .OR.
RANK.EQ.M) GO TO 450
      DO 410 I=NP1MR,NP1
        K=ABS(A(I,MP1))-FLOAT(N)
      B(K)=A(NP2,M)*SIGN(1.,A(I,MP1))
  410 CONTINUE
      IF (RANKP1.EQ.M) GO TO 430
      DO 420 J=RANKP1,MM1
        K=ABS(A(NP3,J))-FLOAT(N)
        B(K)=(A(NP2,M)-
A(NP2,J))*SIGN(1.,A(NP3,J))
  420 CONTINUE
C TESTIRAJ ZA NEJEDINSTVENO
RJESENJE
  430 DO 440 I=NP1MR,NP1
        IF (ABS(A(I,M)).GT.TOL) GO
TO 440
        OCODE=0
        GO TO 450
  440 CONTINUE
  450 IF (MODE.NE.2 .AND.
MODE.NE.3) RESMAX=A(NP2,M)
      IF (RANK.EQ.M) RESMAX=0
      IF (MODE.EQ.4)
RESMAX=RESMAX-D
  455 FORMAT(A30,$,/)
      WRITE(*,455)'RJESENJE:'
      OPEN(UNIT=7,FILE='X.DAT')
      DO 457 I=1,N
      WRITE(7,456)'X',I,X(I)
  456
FORMAT(A1,$,1X,'(',I2,')','=',F12.4)
  457 CONTINUE
      IF (R.EQ.1) THEN
      OPEN(UNIT=8,FILE='V.DAT')
      DO 465 J=1,M
        WRITE(8,469)'v',J,-B(J)
  465 CONTINUE
      ENDIF
      IF (R.EQ.0) THEN
      OPEN(UNIT=9,FILE='V.DAT')
      D=TOL
      DO 468 J=1,M
        B(J)=B(J)/P(J)
        WRITE(9,469)'v',J,-B(J)
        DD=ABS(B(J))
        IF (DD.LE.D) GO TO 468
        D=DD
  468 CONTINUE
      WRITE(*,489) D
      ENDIF
  469
FORMAT(A1,$,1X,'(',I2,')','=',F12.4)
  470 FORMAT(1X,'BROJ
ITERACIJA=',I4)
      WRITE(*,470)ITER
      IF (R.EQ.1) THEN
      WRITE(*,489) RESMAX
      ENDIF
  489 FORMAT(1X,'łvłmax=',F12.4)
  490 FORMAT(1X,'RANG=',I4)
      WRITE(*,490) RANK
  500 FORMAT(1X,'OCODE=',I4)
      WRITE(*,500) OCODE
      END
```