

RAČUNALNIŠKI PODPROGRAMI ZA RAČUNANJE LASTNIH VREDNOSTI IN LASTNIH VEKTORJEV

INFORMATICA 2/91

Keywords: principal components analysis, eigenvalues, eigenvectors, diagonalization method, subroutine

Vesna Čančer
Ekonomsko-poslovna fakulteta v Mariboru
Razlagova 14, 62000 Maribor

V članku obravnavamo računalniške podprograme za računanje lastnih vrednosti in lastnih vektorjev simetrične korelacijske matrike, ki jih potrebujemo pri analizi glavnih komponent. V njem je uporabljena modifikacija Jacobijevega postopka, imenovana "Pope in Tompkins", ki je primerna za računalniško programiranje. Čeprav se uporabljajo hitrejše metode s tridiagonalizirano korelacijsko matriko, se Jacobijeva metoda se pojavlja v sodobnih podprogramskih paketih, saj omogoča natančne izračune lastnih vrednosti in njim pripadajočih lastnih vektorjev. Delovanje podprograma smo ponazorili s primerom in izpisali končni rezultat.

ABSTRACT

The following paper deals with the fortran computer subroutines for computing eigenvalues and eigenvectors of a real symmetric matrix, which are necessary for the principal components analysis. Because of its convenience for computer programming, diagonalization method originated by Jacobi and modified by the iterative "Pope and Tompkins" technique is used. In spite of some quicker methods, based on the threedagonalization matrix, Jacobi's method still appears in the modern subroutine packages because it provides precise definitions of eigenvalues and associated eigenvectors. A numeric example illustrating the application of the described procedure and given computer subroutine is presented.

1. UVOD

Analiza glavnih komponent je statistična tehnika, s katero n statističnih znakov x_1, x_2, \dots, x_n linearno in ortogonalno nadomestimo z enakim številom nekoreliranih glavnih komponent y_1, y_2, \dots, y_n [3], [8]. Transformacija temelji na iskanju lastnih vrednosti in lastnih vektorjev kovariančne matrike ali, kot v našem primeru, korelacijske matrike A , ki je kovariančna matrika standardiziranih spremenljivk x_1, x_2, \dots, x_n [3], [4], [6].

Vzemimo n -razsežni vektor z elementi r_1, r_2, \dots, r_n in poskusajmo rešiti matrično enačbo

$$\begin{bmatrix} 1 & a_{12} & \dots & a_{1n} \\ a_{21} & 1 & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & 1 \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_n \end{bmatrix} = \lambda \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_n \end{bmatrix}, \quad (1.1)$$

kjer je A matrika korelacijskih koeficientov in λ poljubno število [8].

To enačbo je mogoče zapisati v obliki sistema n linearnih enačb, ki jih uredimo tako, da vse člene z neznankami r_1, r_2, \dots, r_n prenesemo na levo stran. Dobimo homogen sistem enačb; desne strani vseh enačb so namreč enake nič [8], [10]. Tak sistem ima pri poljubni vrednosti λ trivialno rešitev $r_1 = r_2 = \dots = r_n = 0$. Netrivialna rešitev eksistira kvečjemu tedaj, kadar je

determinanta koeficientov $A = 0$ [10]. Če za λ vzamemo tako vrednost, da enačbe tvorijo odvisen sistem, dobimo neskončno mnogo rešitev. Sistem homogenih enačb je torej odvisen, če je determinanta sistema enaka nič:

$$\begin{vmatrix} 1-\lambda & a_{12} & \dots & a_{1n} \\ a_{21} & 1-\lambda & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & 1-\lambda \end{vmatrix} = 0 \quad (1.2)$$

To je karakteristična enačba korelacijske matrike, ki jo lahko zapišemo v obliki algebrajske enačbe n -te stopnje [8]. Rešitve karakteristične enačbe (1.2) $\lambda_1, \lambda_2, \dots, \lambda_n$ imenujemo lastne vrednosti korelacijske matrike. Predpostavimo, da so lastne vrednosti realne in različne. Uredimo jih po velikosti:

$$\lambda_1 > \lambda_2 > \dots > \lambda_n. \quad (1.3)$$

Vsaki lastni vrednosti $\lambda_i, i = 1, 2, \dots, n$, pripada lastni vektor r_i , ki je rešitev matricne enačbe:

$$A r_i = \lambda_i r_i.$$

Lastnosti lastnih vrednosti in lastnih vektorjev korelacijske matrike so podrobneje opisane npr. v [3], [4], [8] in [10].

S podprogramom EIGEN [5] se računajo lastne vrednosti in lastni vektorji simetrične korelacijske matrike, katere elementi so v podprogramu urejeni v vektor, označen s formalno spremenljivko A . V tej spremenljivki se izračunajo tudi lastne vrednosti. Tvori se matrika lastnih vektorjev, katere elementi so v podprogramu urejeni v vektorju R .

2. OSNOVE UPORABLJENE METODE

Uporabljena je za računalniško programiranje primerna modifikacija Jacobijeve metode za določanje lastnih vrednosti, t. j. algoritem Pope in Tompkins [4], ki ga je za računalnike priredil von Neumann. Bistvo Jacobijevega postopka za ugotavljanje lastnih vrednosti je, da se z zaporedjem ortogonalnih transformacij korelacijska matrika pretvori v diagonalno matriko lastnih vrednosti:

$$\begin{aligned} A_1 &= R_1' A R_1 \\ A_2 &= R_2' A_1 R_2 \\ &\vdots \\ A_g &= R_g' A_{g-1} R_g = A, \end{aligned}$$

kjer je A simetrična korelacijska matrika,

R transformacijska matrika ortogonalnih transformacij

in A diagonalna matrika lastnih vrednosti.

Z vsako ortogonalno transformacijo lahko en nediagonalni element reduciramo na nič. Proces se prične z redukcijo največjega nediagonalnega elementa, nato se reducira drugi največji nediagonalni element itd., dokler niso vsi nediagonalni elementi reducirani na nič. V vsakem zaporedju moramo poznati elementarno transformacijsko matriko, s katero bomo izbrani element v matriki reducirali na nič [10], [11]:

$$R_l = \begin{bmatrix} \cos \Theta & \sin \Theta & 0 & \dots & 0 \\ -\sin \Theta & \cos \Theta & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix} \quad (2.1)$$

Kot Θ , za katerega moramo zarotirati korelacijsko matriko A , da bo element a_{lm} postal nič, je podan z izrazom:

$$\tan 2 \Theta = -2 a_{lm} / (a_{ll} - a_{mm}). \quad (2.2)$$

Ortogonalna transformacije je z geometričnega vidika podrobneje razložena v [3], [4], [10] in [11]; v članku navajamo le ključne enačbe, ki so uporabljene v podprogramu EIGEN [5]. Kot Θ je določen s predmultiplikacijo in postmultiplikacijo matrike A s transformacijsko matriko R in pogojem, da mora biti rezultat taksne multiplikacije nič:

$$(a_{ll} - a_{mm}) \sin \Theta \cos \Theta + a_{lm} (\cos^2 \Theta - \sin^2 \Theta) = 0. \quad (2.3)$$

Upoštevajmo vrednosti za $\sin 2 \Theta$ in $\cos 2 \Theta$:

$$1/2 (a_{ll} - a_{mm}) \sin 2 \Theta + a_{lm} \cos 2 \Theta = 0, \quad (2.4)$$

iz česar sledi:

$$-2 a_{lm} / (a_{ll} - a_{mm}) = \sin 2 \Theta / \cos 2 \Theta = \tan 2 \Theta. \quad (2.5)$$

Nato iz izraza za tangens dobimo vrednosti $\sin \Theta$ in $\cos \Theta$, t. j. elemente transformacijske matrike R . Bistvena razlika med izvorno Jacobijevo metodo in algoritmom Pope in Tompkins je v tem, da se z njim na nič ne reducira le en element, pač pa se z določitvijo minimalne meje tolerance zaporedno eliminirajo vsi nediagonalni elementi, ki so večji od te meje. Nato se določi nova minimalna meja tolerance, ki je nižja od prejšnje meje. Proces se nadaljuje vse do končne meje tolerance napake za lastne vrednosti. Algoritem tvori osem korakov:

V 1. koraku poiščemo vsote kvadratov vseh nediagonalnih elementov korelacijske matrike in kvadratni koren te vsote, t. j. začetno normo

$$\nu_1 = \left(\sum_{i \leq k} 2 a_{ik}^2 \right)^{1/2}, \quad (2.6)$$

ki ji je v podprogramu prirejena formalna spremenljivka ANORM.

V 2. koraku določimo končno mejo tolerance napake za lastne vrednosti:

$$\nu_F = \frac{\nu_1 \cdot 10^{-6}}{N}, \quad (2.7)$$

kjer je N rang korelacijske matrike. V podprogramu je tej meji prirejena formalna spremenljivka ANRMX.

V 3. koraku tvorimo enotsko transformacijsko matriko R.

V 4. koraku določimo začetno minimalno mejo tolerance, tako da bodo vsi nedijagonalni elementi, večji od te meje, rotirani na nič. V podprogramu ji je prirejena formalna spremenljivka THR.

V 5. koraku iščemo nedijagonalne elemente, ki so po absolutni vrednosti večji od minimalne meje tolerance. Iskanje lahko omejimo na desno od glavne diagonale, ker je korelacijska matrika simetrična glede na glavno diagonalo. Vsak tak element je zreduciran na nič. Iskanje in reduciranje nadaljujemo, dokler vsi elementi, ki niso na diagonali, niso manjši ali enaki dani meji. Tedaj definiramo novo minimalno mejo tolerance in nadaljujemo opisani postopek, dokler ne dosežemo končne meje tolerance.

V 6. koraku določimo elemente zaporedne transformacijske matrike A po vsaki iteraciji; v tej matriki se izračunajo tudi lastne vrednosti korelacijske matrike.

V 7. koraku z ortogonalno transformacijo zajamemo tudi enotsko transformacijsko matriko R; v tej matriki se izračunajo tudi lastni vektorji korelacijske matrike.

V 8. koraku ugotavljamo lastne vektorje korelacijske matrike; lastne vrednosti in njim pripadajoče lastne vektorje uredimo po velikosti lastnih vrednosti.

V zbirki podprogramov IBM Corporation [5] so 5., 6. in 7. korak opisanega algoritma zaradi lažjega razumevanja podprograma EIGEN strnjeni v enačbe, ki jih bomo uporabili pri razlagi delovanja obravnavanega podprograma:

$$\lambda = -a_{lm} \quad (2.8)$$

$$\mu = 1/2 (a_{ll} - a_{mm}) \quad (2.9)$$

$$\omega = \text{sign}(\mu) \frac{\lambda}{\sqrt{(\lambda^2 + \mu^2)}} \quad (2.10)$$

$$\sin \theta = \frac{\omega}{\sqrt{2(1 + \sqrt{1 - \omega^2})}} \quad (2.11)$$

$$\cos \theta = \sqrt{1 - \sin^2 \theta} \quad (2.12)$$

$$B = a_{ll} \cos \theta - a_{lm} \sin \theta \quad (2.13)$$

$$C = a_{ll} \sin \theta + a_{lm} \cos \theta \quad (2.14)$$

$$B = r_{ll} \cos \theta - r_{lm} \sin \theta \quad (2.15)$$

$$r_{im} = r_{il} \sin \theta + r_{im} \cos \theta \quad (2.16)$$

$$r_{il} = B \quad (2.17)$$

$$a_{ll} = a_{ll} \cos^2 \theta + a_{mm} \sin^2 \theta - 2a_{lm} \sin \theta \cos \theta \quad (2.18)$$

$$a_{mm} = a_{ll} \sin^2 \theta + a_{mm} \cos^2 \theta + 2a_{lm} \sin \theta \cos \theta \quad (2.19)$$

$$a_{lm} = (a_{ll} - a_{mm}) \sin \theta \cos \theta + a_{lm} (\cos^2 \theta - \sin^2 \theta) \quad (2.20)$$

3. OPIS DELOVANJA PODPROGRAMA EIGEN S PRIMEROM

V glavnem programu [2] kličemo podprogram EIGEN z ukazom CALL, s katerim se dejanski parametri iz glavnega programa priredijo formalnim parametrom v podprogramu:

| formalni parametri v podprog. | p o m e n |
|-------------------------------|---|
| A | vhodna simetrična korelacijska matrika, urejena kot vektor, ki se med računanjem v podprogramu unici; izhodni vektor lastnih vrednosti, ki so elementi matrike lastnih vrednosti |
| R | izhodna matrika lastnih vektorjev, urejena kot vektor |
| N | rang matrik A in R, enak redu teh matrik zaradi neodvisnosti statističnih znakov |
| MV | vhodna spremenljivka za EIGEN, ki ji priredimo vrednost z ukazom DATA v glavnem programu in je lahko 0 za računanje lastnih vrednosti in vektorjev ali 1 za računanje zgolj lastnih vrednosti |

V glavnem programu smo spremenljivki MV priredili vrednost 0.

V EIGEN-u niso potrebni drugi podprogrami ali funkcijski podprogrami. Za ponazoritev delovanja podprograma bomo navedli vmesne in izpisali končne rezultate. Za lažje razumevanje uporabljenega algoritma izpisimo podprogram EIGEN [5].

```

SUBROUTINE EIGEN(A,R,N,MV)
DIMENSION A(1),R(1)
C   TVORJENJE ENOTSKE MATRIKE
5   RANGE=1.0E-6
   IF(MV-1) 10,25,10
10  IQ=-N
   DO 20 J=1,N
   IQ=IQ+N
   DO 20 I=1,N
   IJ=IQ+I
   R(IJ)=0.0
   IF(I-J) 20,15,20
15  R(IJ)=1.0
20  CONTINUE
C   RACUNANJE ZACETNE IN KONCNE NORME (ANORM IN
C   ANRMX)
25  ANORM=0.0
   DO 35 I=1,N
   DO 35 J=I,N
   IF(I-J) 30,35,30
30  IA=I+(J-J)/2
   ANORM=ANORM+A(IA)*A(IA)
35  CONTINUE
   IF(ANORM) 165,165,40
40  ANORM=1.414*SQRT(ANORM)
   ANRMX=ANORM*RANGE/FLOAT(N)
C   INICIALIZACIJA INDIKATORJA IN RACUNANJE
C   INPUTA (THR)
   IND=0

```

```

      THR=ANORM
45  THR=THR/FLOAT(N)
50  L=1
55  M=L+1
C   RACUNANJE SIN IN COS
60  MQ=(M*M-M)/2
      LQ=(L*L-L)/2
      LM=L+MQ
62  IF( ABS(A(LM))-THR ) 130,65,65
65  IND=1
      LL=L+LQ
      MM=M+MQ
      X=0.5*(A(LL)-A(MM))
68  Y=-A(LM)/ SQRT(A(LM)*A(LM)+X*X)
      IF(X) 70,75,75
70  Y=-Y
75  SINX=Y/ SQRT(2.0*(1.0+( SQRT(1.0-Y*Y))))
      SINX2=SINX*SINX
78  COSX= SQRT(1.0-SINX2)
      COSX2=COSX*COSX
      SINCS =SINX*COSX
C   ROTIRANJE L IN M STOLPCEV
      ILQ=N*(L-1)
      IMQ=N*(M-1)
      DO 125 I=1,N
      IQ=(I*I-I)/2
      IF(I-L) 80,115,80
80  IF(I-M) 85,115,90
85  IM=I+MQ
      GO TO 95
90  IM=M+IQ
95  IF(I-L) 100,105,105
100 IL=I+LQ
      GO TO 110
105 IL=L+IQ
110 X=A(IL)*COSX-A(IM)*SINX
      A(IM)=A(IL)*SINX+A(IM)*COSX
      A(IL)=X
115 IF(MV-1) 120,125,120
120 ILR=ILQ+I
      IMR=IMQ+I
      X=R(ILR)*COSX-R(IMR)*SINX
      R(IMR)=R(ILR)*SINX+R(IMR)*COSX
      R(ILR)=X
125 CONTINUE
      X=2.0*A(LM)*SINCS
      Y=A(LL)*COSX2+A(MM)*SINX2-X
      X=A(LL)*SINX2+A(MM)*COSX2+X
      A(LM)=(A(LL)-A(MM))*SINCS+A(LM)*(COSX2-SINX2)
      A(LL)=Y
      A(MM)=X
C   KONCNI TESTI
C   TEST, DA JE M ZADNI STOLPEC
130 IF(M-N) 135,140,135
135 M=M+1
      GO TO 80
C   TEST, DA JE L PREDZADNI STOLPEC
140 IF(L-(N-1)) 145,150,145
145 L=L+1
      GO TO 55
150 IF(IND-1) 160,155,160
155 IND=0
      GO TO 50
C   PRIMERJANJE INPUTA S KONCNO NORMO
160 IF(THR-ANRNX) 165,165,45
C   SORTIRANJE LASTNIH VREDNOSTI IN LASTNIH VEKTOV
165 IQ=-N
      DO 185 I=1,N
      IQ=IQ+N
      LL=I+(I*I-I)/2
      JQ=N*(I-2)
      DO 185 J=I,N
      JQ=JQ+N
      MM=J+(J*J-J)/2
      IF(A(LL)-A(MM)) 170,185,185
170 X=A(LL)
      A(LL)=A(MM)
      A(MM)=X
      IF(MV-1) 175,185,175
175 DO 180 K=1,N
      ILR=IQ+K
      IMR=JQ+K
      X=R(ILR)
      R(ILR)=R(IMR)

```

```

180 R(IMR)=X
185 CONTINUE
      RETURN
      END

```

Simetrična vhodna korelacijska matrika, ki vsebuje korelacijske koeficiente med 3 statističnimi znaki, ki smo jih poprej standardizirali (družbenim produktom na prebivalca, izdatki za osebno potrošnjo na prebivalca in izdatki za investicije na prebivalca, podanimi v cenah leta 1972 in opazovanimi v Jugoslaviji v letih 1974 do 1988) [9]:

$$A = \begin{bmatrix} 1.000 & 0.887 & -0.125 \\ 0.887 & 1.000 & 0.307 \\ -0.125 & 0.307 & 1.000 \end{bmatrix} \quad (3.1)$$

se v podprogramu EIGEN priredi kot vektor v formalni spremenljivki A z naslednjim zaporedjem elementov: 1., 0.887, 1., -0.125, 0.307, 1., torej je njihovo število $N*(N+1)/2 = 6$.

V 1. zapisu je definiran podprogram EIGEN in izhodni spremenljivki A in R ter vhodne spremenljivke A, N in MV. V podprogramu je upoštevana enojna natančnost. Če bi zeleli dvojno natančnost, bi to definirali z ukazom
 DOUBLE PRECISION A,R,ANORM,ANRNX,THR,X,Y,SINX,SIX2,
 COSX,COSX2,SINCS,RANGE,DSQRT,DABS
 Tudi fortranske funkcije morajo biti dvojne natančnosti. SQRT v ukazih 40,68,75 in 78 moramo zamenjati z DSQRT. ABS v ukazu 62 moramo zamenjati v DABS. Popraviti moramo tudi konstanto v ukazu 5, in sicer na 1.0D-12.

Z ukazi 10-1 do 20 se v primeru, ko smo v glavnem programu spremenljivki MV dodelili vrednost 0, tvori enotska transformacijska matrika (2.1), katere elementi so urejeni v vektor R(IJ). Izvede se 3. korak algoritma Pope in Tompkins.

Z ukazi 25 do 40 se tvori začetna norma ANORM (2.6). Izvede se 1. korak algoritma Pope in Tompkins. V našem primeru je začetna norma 1.3389340.

V ukazu 40+1 se tvori končna meja tolerance napake za lastne vrednosti oz. končna norma ANRNX (2.7). Izvaja se torej 2. korak obravnavanega algoritma. V našem primeru je končna norma 0.0000004.

Z ukazi 45-2 do 55 poteka inicializacija indikatorjev IND in določanje začetne minimalne meje tolerance, kar sodi v 4. korak v algoritmu Pope in Tompkins. V našem primeru se je izračunalo 14 minimalnih mej tolerance THR; prva je bila 0.4463114, zadnja pa 0.0000003.

Z ukazi od 60 do 78+2 se računajo sinusi in kosinusi, nato pa se do 130-1 zamenjujejo L in M stolpci. V vsaki iteraciji se namreč

selekcioniраjo nediagonalni elementi. To sodi v 5., 6. in 7. korak obravnavanega algoritma. Iz ukaza 62 je razvidno: če je nediagonalni element korelacijske matrike po absolutni vrednosti manjši od prej določene minimalne meje tolerance THR, ni reduciran na nič; če so nediagonalni elementi po absolutni vrednosti enaki ali večji od prej določene minimalne meje tolerance, pa se začne reduciranje tega elementa na nič. V našem primeru je v prvi iteraciji $A(2)$, ki je 0.887, večji od THR, ki je 0.446, zato sledi reduciranje $A(2)$ na nič. V ukazu 68-1 se izračuna X po (2.9). X predstavlja del izraza (2.4). V ukazu 68 se po (2.10), v katerem nastopa (2.8), izračuna Y , ki je potreben pri računanju vrednosti elementov transformacijske matrike R in A : $\sin \Theta$ in $\cos \Theta$. V ukazu 75 se po (2.11) izračuna element transformacijske matrike $R \sin \Theta$ in se zapomni v formalni spremenljivki $SINX$. V naslednjem ukazu se izračuna $\sin^2 \Theta$ in se zapomni v formalni spremenljivki $SINX2$. V ukazu 78 se po (2.12) izračuna $\cos \Theta$ in se zapomni v $COSX$. V naslednjih dveh ukazih se izračunata $\cos^2 \Theta$ in $\sin \Theta \cos \Theta$ in se zapomnita v spremenljivkah $COSX2$ in $SINCS$.

V 6. koraku se določijo elementi zaporednih transformacijskih matrik po vsaki iteraciji. V ukazu 110 se izračuna X po (2.13). V naslednjem ukazu se izračuna $A(IM)$ po (2.14). Nato se $A(IL)$ priredi X . Če se računajo le lastne vrednosti, se 7. korak algoritma preskoči, sicer pa se na ukazu 120 prične 7. korak v algoritmu Pope in Tompkins, v katerem je z ortogonalno transformacijo zajeta tudi transformacijska matrika R , ki je v podprogramu urejena v vektor. V ukazu 120+2 se po (2.15) izračuna X . V naslednjem ukazu se po (2.16) izračuna $R(IMR)$ in nato po (2.17) se $R(ILR)$. V našem primeru se v zadnjih dveh iteracijah v spremenljivkah $R(1)-R(9)$ izračunajo elementi lastnih vektorjev, ki se niso dokončno urejeni. Ko se postopek izvede za N -ti I , se v ukazu 125+1 zaradi poenostavitve naslednjih dveh izrazov izračuna tretji element v izrazih (2.18) in (2.19). V navedenih izrazih se izračunata Y in X ; to sodi v 6. korak obravnavanega algoritma. V naslednjem ukazu se po (2.20) izračuna $A(LM)$, ki ustreza levi strani (2.3); izvede se torej 5. korak obravnavanega algoritma. Nato se v 6. koraku $A(LL)$ priredi Y , izračunan po (2.18), $A(MM)$ pa se priredi X , izračunan po (2.19). V našem primeru se v zadnjih dveh iteracijah izračunajo od nič različne lastne vrednosti, ki se niso urejene po velikosti, in se zapomnijo v spremenljivkah $A(LL)$ in $A(MM)$, torej v $A(1)$, $A(3)$ in $A(6)$.

Sledijo testi za dokončanje postopka. Če je element $A(LM)$ manjši od minimalne meje tolerance THR, se v ukazu 160 THR primerja s končno mejo tolerance napake za lastne vrednosti ANRMX. Če je

minimalna meja tolerance THR manjša ali enaka končni meji tolerance napake za lastne vrednosti ANRMX, se nadaljuje z 8. korakom, sicer pa se računa nova minimalna meja tolerance THR in proces reduciranja se nadaljuje, vse dokler ni dosežena končna meja tolerance ANRMX. V našem primeru se proces reduciranja konča, ko je minimalna meja tolerance THR z vrednostjo 0.0000003 manjša od končne meje tolerance ANRMX z vrednostjo 0.0000004.

Sledi urejanje lastnih vrednosti po (1.3) in urejanje lastnim vrednostim pripadajočih lastnih vektorjev, kar sodi v 8. korak obravnavanega algoritma. Najprej se po velikosti uredijo lastne vrednosti. V našem primeru je lastna vrednost z indeksom 2 manjša od lastne vrednosti z indeksom 3, zato se indeksu z manjšo absolutno vrednostjo priredi večja lastna vrednost. Prvi, tretji in sesti element so diagonalni elementi v matriki lastnih vrednosti, ki je urejena v vektor, in so različni od nič: $\lambda_1 = 1.906$, $\lambda_2 = 1.076$ in $\lambda_3 = 0.017$. Če se računajo tudi lastni vektorji, je urejanje lastnih vektorjev odvisno od ureditve lastnih vrednosti. V našem primeru so se zamenjala zaporedna mesta 4. in 7., 5. in 8. ter 6. in 9. elementa R . Pri urejanju lastnih vrednosti je pomembna velikost lastne vrednosti, pri urejanju lastnih vektorjev pa je pomemben indeks, katerega vrednost je odvisna od ureditve lastnih vrednosti po velikosti. Z ukazoma RETURN in END se podprogram konča, izvajanje glavnega programa pa se nadaljuje z ukazom, ki sledi ukazu CALL v glavnem programu.

V tabeli 1 so urejene lastne vrednosti in njim pripadajoči lastni vektorji, izračunani za primer (3.1):

TABELA 1: Lastne vrednosti lastni vektorji

| | LASTNE VREDNOSTI | LASTNI VEKTORJI | | |
|---|------------------|-----------------|-------|--------|
| | | 1 | 2 | 3 |
| 1 | 1.906 | 0.681 | 0.717 | 0.149 |
| 2 | 1.076 | -0.317 | 0.105 | 0.942 |
| 3 | 0.017 | -0.660 | 0.689 | -0.299 |

4. ZAKLJUČEK

V nekaterih novejših računalniških podprogramih za računanje lastnih vrednosti in lastnih vektorjev korelacijske matrike se upoštevajo postopki, ki temeljijo na tridiagonalizirani korelacijski matriki, kamor sodijo Givensov, Wilkinsov in Francisov postopek diagonalizacije [4]. Iz tridiagonaliziranih matrik hitreje ugotovimo lastne vrednosti kot iz netridiagonaliziranih matrik. Znana je Kaiserjeva modifikacija Francisove metode, na kateri temelji

npr. podprogram XKISR [1]. Čeprav so ti postopki hitrejši od Jacobijeve metode, pa prav s slednjo dobimo zelo natančne izračune lastnih vrednosti. Zaradi primernosti za računalniško programiranje se Jacobijeva metoda se vedno uporablja. Novejši računalniški podprogrami, npr. XSMEIG [1], v katerih se tudi upošteva algoritem Pope in Tompkins, se od podprograma EIGEN razlikujejo le po uporabi novejših verzij FORTRAN-a, logika delovanja teh podprogramov pa je ista kot v podprogramu EIGEN [5].

LITERATURA

1. CHAGHAGHI, S. Francois: Time Series Package (TSPACK). G. Goos & J. Hartmanis, Springer-Verlag 1985.
2. ČANCER, Vesna: Analiza statističnega vzorca po glavnih komponentah. (Diplomsko delo) EPF, Maribor 1990.
3. DUNTEMAN, H. George: Principal Components Analysis. Sage Publications, Newbury Park 1989.
4. FULGOSI, Ante: Faktorska analiza. Školska knjiga, Zagreb 1984.
5. IBM Corporation: Scientific Subroutine Package GH20-0252--4. IBM 1968.
6. JAMNIK, Rajko: Uvod v matematično statistiko. DMFA Slovenije, Ljubljana 1976.
7. KAVKLER, Ivan: Teoretične osnove in primeri uporabe faktorske analize. (Magistrsko delo) Sveučilište u Zagrebu, Zagreb 1975.
8. MESKO, Ivan: Izbrana poglavja iz kvantitativne analize. VEKS, Maribor 1978.
9. Statistički godišnjak Jugoslavije 1990. Savezni zavod za statistiku, Beograd 1990.
10. VIDAČ, Ivan: Visja matematika I. DZS, Ljubljana 1973.
11. VIDAČ, Ivan: Visja matematika II. DZS, Ljubljana 1975.