

Az előző részben kitűzött feladatok megoldása:

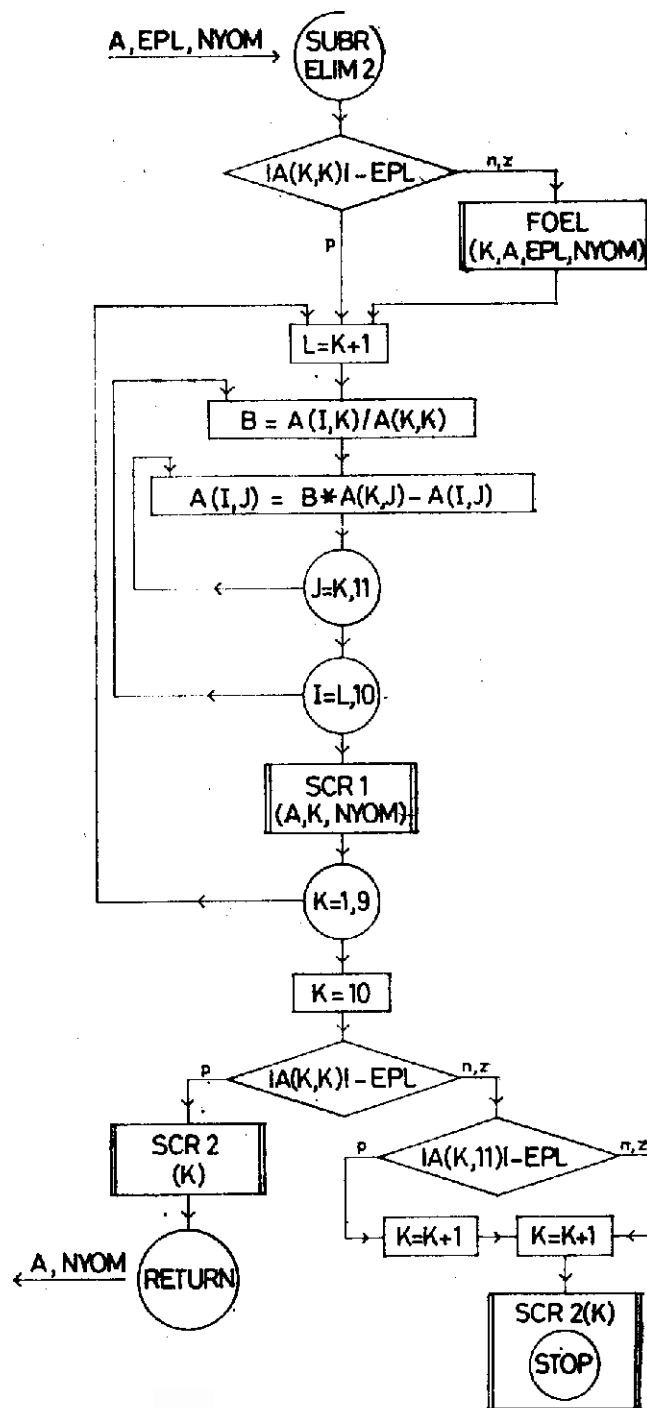
1. *Feladat:* Blokkdiagram készítendő a 10 ismeretlenes egyenletrendszer eliminációs eljárása során alkalmazandó főelem keresésére.

Megoldás: új főelem keresésére, mint említettük, akkor kerül sor, ha az közel zérus értékű, emiatt a vele való osztás túlsordulást, a program futásának leállítását eredményezné. A megfelelő főelem megtalálása után oszlopkeresést hajtunk végre és folytathatjuk az eliminációt. Azt azonban „fől kell íratnunk”, hogy az oszlopok sorrendje miként változott, mivel a visszahelyettesítésnél a kapott gyökök sorrendje a cserélt oszlopok sorrendjével lesz azonos. A gyököket az eredeti sorrendbe csak ennek segítségével rendezhetjük.

Miután így a Gauss-féle eliminációs módszerre épülő program a visszahelyettesítési és ellenőrzési résztől eltekintve elkészült, bemutatjuk a teljes program első részének egy lehetséges formáját. Ezzel kapcsolatos megjegyzéseink a következők:

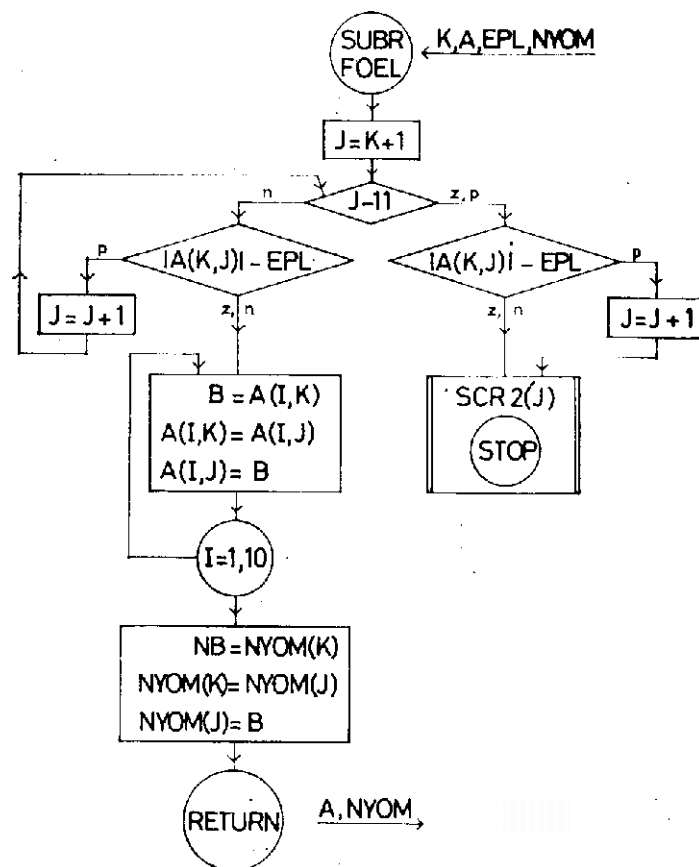
– A **MASTER** szegmens az **A** és **B** tömböket tölti fel az eredeti együttható mátrixszal (utóbbit az ellenőrző részben használjuk fel). \times lesz az eredmény vektor, a **NYOM** vektor az oszlopok sorrendjét tárolja, indulásnál az 1-től 10-ig terjedő egész számokkal töltjük fel. **LX** az **SCR1** szubrutinnak, **NX** az **SCR3** szubrutinnak ad jelzést. **SCR1** az eredeti, ill. a 9-dik menet után kapott együttható mátrixot írja ki. **ELIM2** a főelemkereséssel módosított eliminációs szubrutin. **VISSZ** a visszahelyettesítő, **CONTR** az ellenőrző, **SCR3** az eredményt kijelző szubrutinok, melyekről a következő részben esik szó.

– **ELIM2**-ben **L** a menet számát jelzi. A főelemet a **FOEL** szubrutin keresi ki. A **K = 10** utasítás után a 9-edik menetet követő egyenletrendszer megoldhatóságát vizsgáljuk. Ha **K** értéke 10, az **SCR2** kiírja az elimináció elkészültét, és visszatér a hívó programba, de ha **K** értéke 11 vagy 12, akkor jelzi az egyenletrendszer megoldhatatlanságának okát és a **STOP**-ra fut. A szubrutin blokkdiagramját az 1. ábra mutatja.



1. ábra

– A **FOEL** a főelemet egy k -dik menetben a $(k+1)$ -edik (J -edik) oszloptól a 10-edik oszlopig keresi. Ha a sorban legalább 10 együtttható zérus, akkor az **SCR2** rutint hívja és ott nyomtatás után megáll. Az oszlopcsere a 3-as címkéjű ciklusutasítással kezdődik. A cserélt oszlopok indexeit a **NYOM** vektor megfelelő komponenseinek cseréjével jegyezzük fel. Blokkdiagramját a 2. ábra mutatja.



2. ábra

– A **SCR2** csak akkor engedi vissza a program végrehajtást a hívó szegmensbe, ha **K=10**, azaz ha az eliminációban mindig van zérushoz nem közel eső főelem. Ha a sorban mind a 11 együttható zérussá válik, akkor valamelyik egyenlet előállítható a többiből, ha csak az első tíz zérus, de a 11-edik nem az, akkor van két egymásnak ellentmondó egyenlet.

Ezek után a Gauss-féle eliminációs program első része az alábbi:

```

MASTER PR13
DIMENSION A(10,11),B(10,11),X(14),NYOM(10)
1 READ(1,1)((A(I,J),J=1,11),I=1,10)
FORMAT(9(6F12.2/5F12.2/),6F12.2/5F12.2)
DO 3 I=1,10
NYOM(I)=I
DO 3 J=1,11
3 B(I,J)=A(I,J)
EPL=10.**(-3)
LX=0
NX=-1
CALL SCR1(A,LX,NYOM)
CALL ELIM2(A,EPL,NYOM)
CALL VISSZ(A,X,NYOM)
CALL SCR3(X,NX)
CALL CONTR(B,X)
STOP
END
SUBROUTINE SCR1(A,L,NYOM)
DIMENSION A(10,11),NYOM(10)
IF(L-1) 0,6,6
WRITE(3,1)
1 FORMAT(1H1///20X,28HAZ EREDETI EGYUETTHATOO MAAT,
× 4HRIX:///)
GO TO 4
6 IF(L-9) 8,0,8
WRITE(3,5) L
5 FORMAT(1H1///20X,i2,10H-DIK MENET///)
4 WRITE(3,7)((A(I,J),J=1,11),I=1,10),(NYOM(K),K=1,10)
7 FORMAT(10(/7X,11 F12.2),///20X,
× 21HAZ OSZLOPOK HELYZETE:./15X,10I12/)
8 RETURN
END
SUBROUTINE ELIM2(A,EPL,NYOM)
DIMENSION A(10,11),NYOM(10)
DO 3 K=1,9
IF(ABS(A(K,K))-EPL) 0,0,4
CALL FOEL(K,A,EPL,NYOM)
4 L=K+1
DO 2 I=L,10
B=A(I,K)/A(K,K)
DO 1 J=K,11
1 A(I,J)=B*A(K,J)-A(I,J)
2 CONTINUE
CALL SCR1(A,K,NYOM)
3 CONTINUE
K=10
IF(ABS(A(K,K))-EPL) 5,5,0
CALL SCR2(K)
RETURN
5 IF(ABS(A(K,11))-EPL) 7,7,0
K=K+1
7 K=K+1
CALL SCR2(K)
END
SUBROUTINE SCR2(K)
WRITE(3,2)
IF(K-11) 0,5,7
RETURN

```

```

5 | WRITE(3,6)
  | WRITE(3,9)
  | STOP
7 | WRITE(3,6)
  | WRITE(3,8)
  | STOP
2 | FORMAT(///50X,18HELMINACIOO KEESZ/)
6 | FORMAT(///50X,19HAZ EGYENLETRENDSZER/)
8 | FORMAT(5X,14HELLENTONDASOS)
9 | FORMAT(5X,12HHATAROZATLAN)
  | END
  | SUBROUTINE FOEL(K,A,EPL,NYOM)
  | DIMENSION A(10,11),NYOM(10)
  | J=K+1
4 | IF(J-11) 1,0,0
  | IF(ABS(A(K, J))-EPL) 6,6,0
  | J=J+1
6 | CALL SCR2(J)
1 | IF(ABS(A(K,J))-EPL) 0,0,3
  | J=J+1
3 | DO 5 I=1,10
  | B=A(I,K)
  | A(I,K)=A(I,J)
5 | A(I,J)=B
  | NB=NYOM(K)
  | NYOM(K)=NYOM(J)
  | NYOM(J)=NB
  | RETURN
  | END

```

2. feladat. Program készítendő egy 5×5 -ös méretű bővös négyzet betöltésére és kinyomtatására.

Megoldás: egy lehetséges program az alábbi:

```

  | MASTER PR14
  | DIMENSION A(5,5)
  | READ(1,9) AK,D
  | M=1
  | N=3
  | DO 8 J=1,5
  | DO 7 I=1,5
  | A(M,N)=AK
  | AK=AK+D
  | IF(M-5) 0,1,0
  | M=M+1
  | GO TO 2
1 | M=1
2 | IF(N-5) 0,3,0
  | N=N+1
  | GO TO 7
3 | N=1
7 | CONTINUE
  | IF(N-1) 0,4,0
  | N=N-1
  | GO TO 8
4 | N=5
8 | CONTINUE
  | OESSZ=0.
  | DO 5 I=1,5
5 | OESSZ=OESSZ+A(I,3)
  | WRITE(3,6) ((A(I,J),J=1,5),I=1,5),OESSZ
6 | FORMAT(1H1,20(/),5(50X,5F12.2//),50X,
  | × 40HA SOR- ILL. OSZLOPOSSZEGEK KOZOS ERTEKE:,
  | × 3X,F12.2)
9 | FORMAT(2F12.2)
  | STOP
  | END

```

A feladatban nem adtuk meg az elsőként betöltendő „kezdő” mátrix elem indexeit, emiatt az tetszőleges. A közölt programban a kezdő elem sor-, ill. oszlopindexeit az **M**, ill. **N** azonosítók értékadással kapják. Ha megfelelően adjuk meg az indexeket, akkor nem csak a sorok és oszlopok összege „bővös” (ugyanaz a szám), hanem a fő-, ill. mellékátlókra eső elemek összege is. Ilyenkor diagonálisan bővös a négyzet, ami a példában adott kezdőelem indexekkel is bekövetkezik. A bővös négyzet további speciális tulajdonsága lehet, hogy a fő-, ill. mellékátlókkal párhuzamos „tört átlók” mentén elhelyezkedő elemek összege is bővös, azaz pándiagonális a négyzet. A példánkban a mellékátlókkal párhuzamos tört átlókra ez is fennáll. Pl. $a_{2,1} + a_{1,2} + a_{5,3} + a_{4,4} + a_{3,5}$ egy ilyen mellékátlóval párhuzamos tört átló elemei. A kitöltés más „szabály” alapján is történhet, pl. lóugrás szerint. Bizonyos páros rendű mátrixok is lehetnek bővös négyzetek, ezek kitöltése bonyolultabb. Bővös négyzetekkel rovatunk tovább nem foglalkozik.

6.2. folytatás. A főelem keresése a megoldás pontosságával is kapcsolatos. Ha mindig a legnagyobb abszolút értékű sor-elemet választjuk főelemnek, akkor növelni tudjuk a pontosságot. Sajnos még ilyen módszer mellett is előfordul-

hat, hogy az eljárás használhatatlanul pontatlan gyökökhöz vezet, ami a kiindulási mátrix bizonyos tulajdonságaival magyarázható. Ennek kifejtésébe nem bocsátkozhatunk.

6.3. Visszahelyettesítés, ellenőrzés

A visszahelyettesítő szubrutin az eliminált együtthatójú egyenletrendszerből „alulról fölfelé haladva” kiszámítja a gyököket. Ezek sorrendje azonban a megcserélt oszlopok sorrendjével egyezik. Hogy végül tudjuk, melyik az első, második stb. gyök, a **NYOM** vektor segítségével helyre kell állítani sorrendjüket. Ezt is a visszahelyettesítő szubrutin végzi. A k -adik gyök egy olyan tört, melynek nevezője a sor fődiagonálisba eső eleme. A számlálót kéttagú kifejezésnek tekinthetjük, első tagja a k -adik sor 11-edik eleme, a második tagja egy **C** szám, melynek számítása a $\mathbf{C} = \mathbf{C} + \mathbf{A}(\mathbf{L} + \mathbf{1}, \mathbf{K}) * \mathbf{Y}(\mathbf{K})$ utasítással történhet, ha **C** kezdőértéke zérus, és **L** értékét 10-től visszafelé $(k + 1)$ -ig változtatjuk.

A pontosság bizonytalansága miatt a **B** tömbben eltett eredeti együttható mátrix felhasználásával visszahelyettesítjük az eredmény vektor komponenseit az egyenletek bal oldalába, és a helyettesítési értékeket kivonjuk a jobb oldali számértékekből. Ha a különbségek egy kívánt pontossági intervallumba esnek, az eredmény vektort jónak mondjuk.

Feladatok

1. Írjuk meg a **PR13** hiányzó szubrutinjait. (Az ellenőrző vektort is nyomtassa!)
2. A KÖMAL 55. kötet 2. száma 63. oldalán levő 2077-es feladatot módosítsuk az alábbi szerint: Két db egységnyi sugarú körlemez úgy fekszik egymáson, hogy együttvéve 6π egységnyi területet fednek le. Szorítsuk olyan két tört közé a középpontok távolságát, amelynek nevezője 1000, a számlálók egészek, és a számlálók különbsége 1. A feladat számítására készítsünk programot.