

NAGY HÁLÓZATOK MEGOLDÁSI MÓDSZEREIRŐL

SINGER DÉNES

A MŰSZAKI TUDOMÁNYOK DOKTORA

[Beérkezett: 1973. nov. 29-én]

A tanulmány ismerteti a nagy hálózatok számítására számba jövő módszereket és ezeket gazdaságosságuk szempontjából vizsgálja. Részletes elemzés tárgyává teszi a hálózati mátrix ritkaságát kihasználó módszereket, melyek a többi módszert hatékonyságban lényegesen felülmúlják. A továbbiakban a „ritka mátrix technika” szempontjából elemzi a dekompozíciós eljárásokat és rámutat azokra az eddig kiaknázatlan lehetőségekre, melyek a dekompozíció és a ritka mátrix technika együttes alkalmazásából adódhatnak.

Bevezetés

A nagy rendszerek problematikája szoros kapcsolatban áll a numerikus megoldási módszerek gazdaságosságának kérdésével. Különösen élesen vetődik fel ez a hálózatelméletben, mivel itt már viszonylag egyszerű gyakorlati feladatok megoldása is komoly igényeket támaszt mind a számítási idő, mind a memóriaszükségletet illetően. A kérdés fontossága nemcsak a nagykiterjedésű energiaelosztó hálózatoknál nyilvánvaló, hanem minden olyan fizikai rendszernél is, amely mint hálózat értelmezhető.

A nagy hálózatok számításánál számbajövő módszerek a következő három osztályba sorolhatók:

- a) hálózatredukciós módszerek
- b) dekompozíciós (diakoptikai) módszerek
- c) a hálózati mátrix „ritkaságát” (sparsity) kihasználó módszerek.

Ez a felosztás lényegében megfelel a történelmi fejlődésnek. A hálózat-redukciós módszerek nagy részben még a számítógépek megjelenése előtt lettek kialakítva. A G. Kron kezdeményezte diakoptikai módszerek a számítógépek megjelenésével váltak időszerűvé. A hálózati mátrix ritkaságát kihasználó módszereket a számítógépek által teremtett lehetőségek születték. Ezeknek köszönhető, hogy az utolsó tíz évben a hálózatszámítások gépidő és memóriai igénye 5–10-ed részére csökkent, (azonos géptípust feltételezve). A módszerek említett három osztálya között természetesen nem lehet éles határvonalat húzni. Ez különösen a b) és c) osztályra vonatkozik.

A továbbiakban elsősorban a hálózati mátrix ritkaságát kihasználó módszerekkel foglalkozunk, mivel ezek hatékonyságban a többi módszereket

* Dr. Singer Dénes, 1021 Budapest, Furulya u. 5

lényegesen felülmúlják és rámutatunk arra, hogy a régebbi módszerek — speciális alkalmazásoktól eltekintve, nem gazdaságosak. Új lehetőségeket lát-szanak feltárni a dekompozíciós és a hálózati mátrix ritkaságát kihasználó módszerek kombinációjával adódó számítási eljárások.

Szükségesnek tartjuk hangsúlyozni, hogy az ismertetendő módszerek elsődlegesen lineáris, időben invariáns reciprok hálózatokra, ill. ezek stacionárius állapotainak számítására vonatkoznak. Mint ismeretes a nemlineáris hálózatok széles osztályának számítása ezen feladatokra visszavezethető. Ugyanez vonatkozik a reciprok lineáris és nemlineáris hálózatok tranzienseinek számítására is. Ezért az ismertetendő módszerek jelentősége messze túlmutat elsődleges alkalmazásuk körén.

A hálózati egyenletek megválasztásáról

Nagy hálózatok számításának gazdaságossága nagy mértékben függ a hálózati egyenletek *alakjának* célszerű megválasztásától. Általában nem célszerű közvetlenül a két Kirchhoff-féle törvényből és a hálózatelemek végponti egyenleteiből kiindulni. Megfelelőbbek az ezekből különböző transzformációk útján adódó egyenletek. A hálózati egyenleteknek ily módon adódó két legfontosabb alakja a csomóponti, illetve hurokalak, melyeket vektoriálisan a következőképpen írhatunk:

$$A'YAe' = A'[I - YE] \quad (1)$$

$$C'ZCi' = C'[E - ZI] \quad (2)$$

A jelöléseket illetően lásd az 1. táblázatot.

A hálózati egyenletek más alakúak lesznek, ha az (1) és (2) egyenletekben A , illetve C topológiai mátrixokat más mátrixokkal fejezzük ki. További

I. táblázat

Jelölés	Megnevezés*
e	ágfeszültség vektor
e'	csomóponti feszültség vektor
E	forrásfeszültség vektor
i	ágáram vektor
i'	hurokáram vektor
I	ágforrásáram vektor
Z	impedancia mátrix
Y	admittancia mátrix
A	ág-csomópont mátrix
C	ág-hurok mátrix
Z_L	hurok impedancia mátrix; $Z_L = C'ZC$
Y_N	csomóponti admittancia mátrix; $Y_N = A'YA$

* Megjegyzés: A jelölések feszültségek és áramok helyett bármely intenzív, illetve extenzív változót is jelenthetnek.

alakok származnak az A és C mátrixoknak, valamint a többi változónak a fa-ágak, illetve hídágak szerinti particionálásával, stb. Lásd erre vonatkozólag pl. [1]-et.

A hálózati egyenletek alakjának megválasztásánál a következő szempontok játszanak szerepet:

a) a hálózat topológiai viszonyai: pl., hogy fa-szerű vagy erősen hurkolt-e a hálózat,

b) a hálózat szimmetria tulajdonságai,

c) a hálózat nagysága,

d) az ismert, illetve ismeretlen változók jellege,

e) a számítások célkitűzései, pl., hogy egyedi számításról van-e szó, vagy hogy változatlan struktúra mellett csak a forrásparaméterek változnak.

Emellett, a feladat természetéből folyó szempontok mellett a hálózati egyenletek alakjának megválasztásánál egy általános szempontot is szem előtt kell tartani, a szimultán megoldandó egyenletek számának minimalizálását. Ha eltekintünk a feladat jellegétől, általában az az alak a legmegfelelőbb, melynél az egyenletrendszer mátrixának dimenziója minimális. Pl. kevésbé hurkolt, lényegében fa-struktúrájú hálózatnál célszerű a (2) hurokalak alkalmazása, mivel itt az invertálandó Z_L mátrix dimenziója $[l \times l]$; l a független hurkok száma. Az (1) csomóponti alak használatánál az invertálandó Y_N mátrix $[(n-1) \times (n-1)]$ dimenziójú; n a csomópontszám. Mivel feltételezésünk szerint $l \ll n$, a (2) alak használatánál itt tetemes idő és memória megtakarítás érhető el: első közelítésben az idő- és memóriaigény $(n/l)^3$, illetve $(n/l)^2$ arányban csökken.

A nagy hálózatok számítására szolgáló módszerekkel szemben a következő általános követelmények támaszthatók:

a) a megoldandó egyenletek mátrixának mérete minimális kell hogy legyen,

b) hasznosítsa az egyenletmátrix ritkaságát,

c) hasznosítsa a hálózat szimmetria tulajdonságait.

Emellett a matematikai követelmények mellett szükséges, hogy:

d) a megoldás alkalmazkodjék a feladatosztály jellegéhez,

e) a gépi adatbevitelhez, az adatok tárolásához és visszakereséséhez alkalmazza a leghatékonyabb módszereket.

Nagy lineáris egyenletrendszerek numerikus megoldásánál számbajövő módszerekről

A lineáris egyenletrendszerek megoldására rendelkezésre álló számos numerikus módszer közül nagy hálózatok egyenleteinek megoldására első sorban a következő három módszer jön tekintetbe:

- a) LU faktorizáció
 b) az inverz mátrix szorzat mátrix alakjában való előállítás
 c) Hardy-Cross féle iterációs módszer.

A *LU-faktorizációnál* az egyenletrendszer G mátrixát az L alsó, illetve U felső háromszögmátrix szorzataként állítjuk elő, vagyis:

$$Gx = LUx = b \quad (4)$$

b az ismert, x az ismeretlen változó vektora: Ha g_{ij} -vel, l_{ij} -vel, illetve u_{ij} -vel jelöljük a G mátrix, ill. L és U háromszögmátrixok elemeit, ezek előállítása a következő algoritmusokkal történhet [2]

$$u_{ij} = \left(g_{ij} - \sum_{k=1}^{j-1} l_{ik} u_{kj} \right) / l_{ii} \quad i < j \quad (5)$$

$$l_{ij} = g_{ij} - \sum_{k=1}^{j-1} l_{ik} u_{kj} \quad i \geq j \quad (6)$$

A (4) egyenletrendszer megoldása az L és U elemeinek kiszámítása után csupán egy előremenő és egy visszafelémelő szubsztitúciót igényel. A (4) következőképpen írható:

$$Ly = b \quad (7)$$

ahol

$$Ux = y \quad (8)$$

Három ismeretlenes egyenletrendszer esetében:

$$\begin{bmatrix} l_{11} & & \\ l_{21} & l_{22} & \\ l_{31} & l_{32} & l_{33} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \quad (9)$$

Előremenő szubsztitúciót alkalmazva:

$$\begin{aligned} y_1 &= b_1 / l_{11} \\ y_2 &= b_2 - b_1 \frac{l_{21}}{l_{11}} = b_2 - l_{21} y_1 \\ y_3 &= b_3 - b_1 \frac{l_{31}}{l_{11}} - \left(b_2 - b_1 \frac{l_{21}}{l_{11}} \right) \frac{l_{32}}{l_{22}} = b_3 - l_{31} y_1 - \frac{l_{32}}{l_{22}} y_2 \end{aligned} \quad (10)$$

Az y_1, y_2, y_3 ezen értékeit az

$$\begin{bmatrix} 1 & u_{12} & u_{13} \\ & 1 & u_{23} \\ & & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} \quad (11)$$

egyenletbe helyettesítve az x_1, x_2, x_3 ismeretlenek számára a következő értékek adódnak:

$$\begin{aligned} x_3 &= y_3 \\ x_2 &= y_2 - u_{23}y_3 \\ x_1 &= y_1 - u_{12}y_2 + (u_{12}u_{23} - u_{13})y_3 \end{aligned} \tag{12}$$

Szükséges megjegyezni, hogy az algoritmus csak akkor működik, ha valamennyi $l_{ii} \neq 0$, ellenkező esetben az u_{ij} értékek végtelennek adódnak; lásd az (5) kifejezést. Ez a nehézség a G mátrix sorainak, illetve oszlopainak megfelelő cserélésével megkerülhető.

Az inverz mátrixnak *mátrixszorzat alakjában* való előállítására több lehetőség kínálkozik. A módszernek a következőkben ismertetendő variánsa a többiekkel szemben bizonyos számítástechnikai előnyöket látszik biztosítani [3]. Hozzuk első sorban a $Gx = b$ egyenletet, illetve ennek G mátrixát az ismert Gauss-féle eliminációs módszerrel háromszög alakra. Az egyenletrendszer kiszélesített mátrixából:

$$\begin{bmatrix} g_{11} & g_{12} & \dots & g_{1n} & b_1 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ g_{n1} & g_{n2} & \dots & g_{nn} & b_n \end{bmatrix} = H \tag{13}$$

a sorok lineáris kombinációjával fokozatosan a $H^{(1)}, H^{(2)} \dots H^{(k)} H^{(u)}$ mátrixokat nyerjük, ahol H a keresett háromszögmátrix. A $H^{(k)}$ mátrix a következő alakú:

$$H^{(k)} = \begin{bmatrix} 1 & g_{12}^{(1)} & g_{13}^{(1)} & \dots & g_{1n}^{(1)} & b_1^{(1)} \\ & 1 & g_{23}^{(2)} & \dots & g_{2n}^{(2)} & b_2^{(2)} \\ & \cdot & \cdot & \cdot & \cdot & \cdot \\ & \cdot & \cdot & \cdot & \cdot & \cdot \\ g_{k1} & g_{k2} & g_{k3} & \dots & g_{kn} & b_k \\ g_{(k+1)1} & g_{(k+1)2} & g_{(k+1)3} & \dots & g_{(k+1)n} & b_{(k+1)} \\ & \cdot & \cdot & \cdot & \cdot & \cdot \\ & \cdot & \cdot & \cdot & \cdot & \cdot \\ & \cdot & \cdot & \cdot & \cdot & \cdot \\ g_{n1} & g_{n2} & g_{n3} & \dots & g_{nn} & b_n \end{bmatrix}$$

A $H^{(k)}$; $k = 1, 2, \dots, n$ mátrixok elemei segítségével képezzük most a

$$\begin{bmatrix} d_{11} & u_{12} & u_{13} & \dots & u_{1n} \\ l_{21} & d_{22} & u_{23} & \dots & u_{2n} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ l_{n1} & l_{n2} & l_{n3} & \dots & d_{nn} \end{bmatrix} \tag{15}$$

mátrixot, ahol a d_{ij} , l_{ij} , illetve u_{ij} elemek következőképpen vannak definiálva:

$$\begin{aligned} d_{ii} &= 1/g_{ii}^{(i-1)} & g_{ii} &\neq 0 \\ u_{ij} &= g_{ij}^{(i)} & i &< j \\ l_{ij} &= g_{ij}^{(j-1)} & i &> j \end{aligned} \quad (16)$$

A g_{ij} elemek *felső* indexe itt a $\mathbf{H}^{(k)}$ $k = 1, 2, \dots, n$ indexére; utal; pl. a $g_{ij}^{(i-1)}$ a $\mathbf{H}^{(i-1)}$ mátrix i -edik diagonális elemét jelenti.

Definiáljuk a \mathbf{D} , \mathbf{L} , \mathbf{L}^* , \mathbf{U} , \mathbf{U}^* mátrixokat, melyek az egységmátrixtól csak az i -edik sorban, ill. oszlopban különböznek. Az egyszerűség kedvéért a (17)-ben a teljes mátrixok helyett csak a kérdéses sor, ill. oszlopvektorokat adjuk fel.

$$\begin{aligned} D_i &= [0, 0, \dots, 0, d_{ii}, 0 \dots 0, 0] \\ L_i &= [0, 0, \dots, 0, 1, -l_{(i+1)i}, -l_{(i+2)i}, \dots, -l_{(n-1)i} - l_{ni}] \\ L_i^* &= [-l_{i1}, -l_{i2}, \dots, -l_{i(i-1)}, 1, 0, \dots, 0, 0] \\ U_i &= [0, 0, \dots, 0, 1, -u_{i(i+1)}, -u_{i(i+2)}, \dots, -u_{i(n-1)}, -u_{in}] \\ U_i^* &= [-u_{1i}, -u_{2i} \dots u_{(i-1)i}, 1, 0, \dots, 0, 0]^t \end{aligned} \quad (17)$$

Így a \mathbf{D} az egységmátrixtól csak az i -edik sorban különbözik, melynek helyébe a D_i sorvektor lép. Ugyanígy az \mathbf{L} az egységmátrixtól az i -edik oszlopban levő L_i oszlopvektorban különbözik.

Az így definiált mátrixok jelentősége abban áll, hogy *inverzük* egyszerűen felírható. A D_i inverzének elemei a d_{ii} reciprok értékeként adódnak. Az L_i , L_i^* , U_i és U_i^* invertálásához csupán csak a nemdiagonális elemek előjelét kell megfordítani.

A (17) mátrixok segítségével a (4) egyenlet megoldása:

$$x = \mathbf{G}^{-1}b \quad (18)$$

a következőképpen írható:

$$x = \mathbf{U}_1 \mathbf{U}_2 \dots \mathbf{U}_{(n-1)} \mathbf{D}_n \mathbf{L}_{(n-1)} \mathbf{D}_{(n-1)} \mathbf{L}_{(n-2)} \dots \mathbf{L}_2 \mathbf{D}_2 \mathbf{L}_1 \mathbf{D}_1 b \quad (19)$$

A \mathbf{G} mátrix inverziója így egyszerű ritka mátrixok szorzására lett visszavezetve. Szükséges megjegyezni, hogy a *közvetlen* mátrix-inverzió ritka mátrixoknál rendkívül előnytelen, mivel az invertálás felszámolja a nulla elemeket és így elvesznek azon előnyök, amelyeket a ritka mátrix számítástechnikailag jelent.

A hálózatszámításnál az inverz mátrix szorzat alakban való előállítás akkor jelent előnyt a LU-faktorizációval szemben, ha a hálózat topológiája és paraméterei változatlanok, csupán b forrásparaméterek változnak.

A reciprokok hálózatok mátrixa mindig szimmetrikus, ezeknél a (19) kifejezés egyszerűsödik. A (18) egyenlet megoldása ebben az esetben:

$$x = U_1 U_2 \dots U_{(n-1)} D U_{(n-1)}^t \dots U_2^t U_1^t b, \quad (20)$$

ahol

$$D = D_1 D_2 \dots D_n$$

Szimmetria esetében a műveletek száma is majdnem felére csökken, mivel a $g_{ij}^{(j-1)}$; $i > j$ elemek értékének kiszámításához nem szükségesek a mátrix diagonálisától balra levő elemek. A kiértékelés itt a

$$g_{ij}^{(j-1)} = g_{jj}^{(j-1)} g_{ji}^{(j)} \quad (21)$$

összefüggés alapján történhet.

Megemlítjük, hogy a (17) mátrixok birtokában a (18) egyenletrendszer megoldása még más mátrix-szorzatok alakjában is nyerhető, pl. az

$$x = U_2^* U_3^* \dots U_{n-1}^* U_n^* D_n L_n^* D_{(n-1)} L_{(n-1)}^* \dots D_2 L_3^* D b \quad (22)$$

alakban.

Iterációs módszerek nagy hálózatok egyenleteinek megoldására numerikus instabilitásuk miatt általában kevésbé előnyösek, mint a véges módszerek. Alkalmazásuk így csak akkor célszerű, ha az egyenletrendszer megoldásáról eleve közelítő információval rendelkezünk, vagyis midőn az iterációs műveleteket „jó” induló értékekkel kezdhetjük el, pl. kis számú hurkot tartalmazó hálózatoknál. Ez az eset áll fenn akkor is, ha a hálózat ismert állapotától csak *kevés*sé eltérő állapot számításáról van szó.

Iterációs megoldási módszerek alkalmazhatók akkor is, ha a hálózat nagyobb számú hurkot tartalmaz, de a hídágak ellenállása nagy és így a megfelelő áramok viszonylag kicsinyek. Ez a feltétel sok esetben mesterségesen is biztosítható a hálózat fájának megfelelő kiválasztásával úgy, hogy a hálózat legnagyobb ellenállású ágai szerepeljenek hídágakként. A módszer, amelyet az építészeti statikában való alkalmazása alapján „törzstartó módszernek” nevezhetnénk el, azzal az előnnyel rendelkezik, hogy egyaránt alkalmazható úgy lineáris, mint nem-lineáris hálózatokra. A módszer leglényegesebb része egy, az ágak megfelelő rendezésére szolgáló algoritmus [6].

Ritka mátrixú egyenletrendszerek megoldásáról

Nézzük meg, hogy az előző megoldási módszerek szempontjából mit jelent az egyenlet mátrixának ritkasága, illetve hogyan hasznosítható ezeknél a ritkaság a számítási idő és a memóriaigény csökkentésére? Nagy hálózatok esetében ez a kérdés különösen lényeges, mivel ezeknél a 0-elemek száma ál-

talában sokszorososan meghaladja a nem nulla elemeket. Ez könnyen belátható a hálózati egyenletek (1) csomóponti alakja alapján.

A csomóponti alakban szereplő Y_N csomóponti admittanciamátrixban csak az

$$Y_{Nii} = \sum_j y_{ij}$$

diagonális és az

$$Y_{Nij} = -y_{ij} \quad j = k, l, m \dots$$

nemdiagonális elemek különböznek nullától. y_{ij} az Y primitív admittanciamátrix elemei; $k, l, m \dots$ az i -edik csomóponttal ténylegesen összekapcsolt csomópont indexeit jelenti.

A gyakorlatban előforduló elosztóhálózatoknál (és itt elsősorban villamos, gáz és víz-hálózatokra gondolunk), az egy csomópontba befutó ágak száma átlagban 2–3. A csomóponti admittancia-mátrix „relatív ritkaságát”, η -át, a következőképpen definiáljuk:

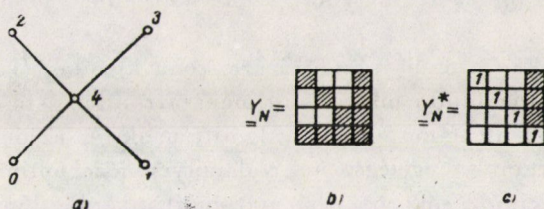
$$\eta = 100 \left| \frac{n - \bar{e} - 2}{n - 1} \right| \quad (23)$$

ahol \bar{e} az egyes csomópontokhoz kötött ágak számának átlagértéke; n a csomópontok száma. Az előbbi $\bar{e} = 2-3$ átlagértékkel számítva egy 100, illetve 1000 csomóponttal rendelkező hálózat csomóponti admittancia mátrixának relatív ritkasága 95–96%, illetve 99,5–99,6%-os! A mondottak alapján nyilvánvaló, hogy a mátrix ritkaságának kihasználásával biztosítható elsősorban a nagy hálózatok számításához szükséges tetemes gépido és memória-szükséglet radikális csökkentése.

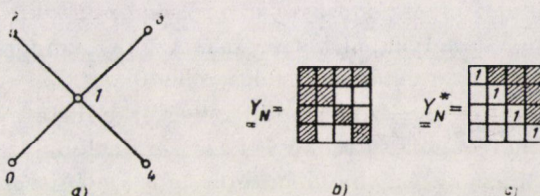
A mátrix ritkaságának kihasználása elsősorban megfelelő programozási technika alkalmazásával érhető el, ami lehetővé teszi, hogy a tárolásnál, illetve az egyenletrendszer megoldásánál eleve csak a nem nulla elemek legyenek figyelembe véve. Ezekre a kérdésekre még a továbbiakban visszatérünk. Itt elsősorban arra szeretnénk rámutatni, milyen szerepet játszik a mátrix ritkaságának kihasználása szempontjából a megoldási módszer megválasztása és az egyenletrendszer belső elrendezése (pivoting). Ez utóbbi a hálózati egyenletek esetében lényegében a hálózat csomópontjainak, illetve ágainak megfelelő sorszámozását jelenti.

Az egyenletrendszer rendezésének kérdése a lineáris algebraiban eredetileg a számítási hibák analízise kapcsán merül fel. A kérdés jelentőségére a mátrix ritkaságának kihasználása szempontjából először Sato és Tiney mutatott rá [4]. Itt szükséges megjegyezni, hogy olyan elrendezés, mely előnyös a mátrix ritkaságának kihasználása szempontjából nem feltétlenül az a számítási hibák minimalizálásához. A hálózatszámítási célokra eddig kidolgozott programok csupán az első szempontot veszik figyelembe és hallgatólagosan fel-

tételezik, hogy a számítási hibák a toleranciákon belül maradnak. A szerző véleménye szerint ez az állapot nem megnyugtató és szükséges a kérdés beható elemzése alapján mindkét szempontból *közel* optimális rendezési sémák kialakítása.



1. ábra



2. ábra

A hálózat csomópontjainak számozása és az Y_N csomóponti admittancia-mátrix 0-elemeinek elrendezése közötti összefüggést az 1. ábrán látható egyszerű hálózat példája illusztrálja; lásd az 1a–b, illetve 2a–b ábrákat. Az egyébként azonos struktúrájú 1a, illetve 2a hálózat csupán abban különbözik egymástól, hogy az 1 és 4 csomópont számozása fellett cserélve. Az 1b, illetve 2b mátrixok háromszögesítésével (Gauss-féle eliminációjával) nyert mátrixot, Y_N^* -t az 1c, illetve 2c ábrák érzékeltetik. Míg az eredeti számozásnál az Y_N^* felső háromszögmátrix telt, az átszámozás után 30%-ban 0-elemeket tartalmaz. Ugyanilyen arányban csökkent a mátrix tárolásához szükséges memória.

A csomópontok számozási módjának nemcsak a memóriaigényre, hanem a szükséges műveletek számára és így a gépidőre is befolyása van. Így az 1. ábrának megfelelő számozásnál, az $Y_N \rightarrow Y_N^*$ transzformációhoz (a műveletek optimális megszervezésénél), 4 osztás, 6 szorzás és 10 kombinált szorzás-összeadás szükséges. A 2. ábrának megfelelő számozásnál ezen átalakításnál a műveletek száma 4 osztásra, 3 szorzásra és 3 kombinált szorzásra-összeadásra redukálódik.

A csomópontok különböző számozásánál tehát a Gauss-féle eliminációnál általában különbözőnek adódik a 0-elemek és a szükséges műveletek száma. Ugyanez vonatkozik a csomóponti admittancia-mátrix LU faktorizációjára, illetve a mátrix inverzének szorzatként való előállítására is.

A mátrix 0-elemeinek száma ezen transzformációknál a csomópontok kedvezőtlen számozása esetén esetleg meg is nőhet, ami kedvezőtlen. A sor-számozás hatékonyságát az eredeti és a transzformáció végrehajtása után meglevő 0-elemek számának hányadosával mérhetjük. *Optimálisnak* nevezhetünk olyan számozást, melynél ezen hányados minimális értéket vesz fel.

A csomópontok optimális számozására vonatkozólag (vagy ami ezzel egyenértékű, a faktorizálandó mátrix optimális átrendezésére vonatkozólag) eddig nem ismeretesek általános érvényű algoritmusok. Az ez ideig szerkesztett hálózatszámító programok megelégszenek valamilyen *közel* optimális számozással. A számbajövő számozások közül hármat említünk meg [5]:

1) A csomópontok számozása a kiinduló ágak száma alapján történik, tehát először kerülnek megszámozásra azon csomópontok, melyekből egy ág indul ki, majd azok, amelyekből kettő, stb. Azonos számú ággal rendelkező csomópontok sorszámai egymás közt felcserélhetők.

2) A csomópontok úgy számozandók, hogy a hálózati egyenlet mátrixának faktorizációjánál ennek sorra következő sora a lehetől egkevesebb nemnulla elemet tartalmazzon. Amennyiben több sor is kielégíti ezt a követelményt, bármelyik választható.

3) A csomópontok úgy számozandók, hogy a hálózati egyenlet mátrixának faktorizációjánál az egyes lépéseknél a legkisebb számú *uj* nem-zéró elem keletkezzék.

A legegyszerűbb az 1. számozás, mivel csupán az egyes csomópontokban összefutó ágak számáról kíván információt. A 2., illetve 3. számozási sémák alkalmazása lényegesen körülményesebb, mivel eleve az egész faktorizációs eljárás szimulációját igényli. Ennek munkai igényessége viszont általában kiegyenlíti azokat az előnyöket, amelyeket alkalmazásuk az 1. számozással szemben nyújthat.

A felsorolt számozási sémák a számítás gazdaságossága szempontjából nem minden esetben előnyösek. Sokszor a feladat speciális adottságait figyelembe vevő számozást célszerű alkalmazni. (Itt eltekintünk attól, hogy *kész* általános célú program alkalmazása, még abban az esetben is előnyösebb lehet, mint a feladat speciális adottságait figyelembe vevő eljárásé, amelyhez külön programot kell szerkeszteni!) Néhány ilyen eset a következő:

1. Az új feladatnál az előzőleg megoldotthoz képest csupán a hálózati egyenletek néhány *forrásparamétere* változik meg. Ilyenkor célszerű a számozást úgy végezni, hogy a hálózati egyenletek változást nem szenvedő sorai kerüljenek előre, mivel így a faktorizációnál a megfelelő előretartó műveleteket nem kell újra elvégezni.

2. Az új feladatnál az előzőhöz képest csupán a *hálózati mátrix* néhány eleme változik meg. Itt olyan számozást célszerű alkalmazni, hogy a hálózati mátrix utolsó soraiként szerepeljenek a változást nem szenvedő sorok.

3. A hálózat speciális topológiai tulajdonságokkal rendelkezik. Így pl. elosztóhálózatoknál gyakran előfordul, hogy ez több egymással csupán kis-számú ággal összekötött részhálózatra bontható szét. Itt célszerű úgy eljárni, hogy először az első, majd a második, stb. részhálózat csomópontjait látjuk el sorszámmal.

Az említett esetekben a hálózati egyenletek megoldásához célszerűek a különböző dekompozíciós módszerek.

Programozási technikák a mátrixok ritkaságának kihasználására

Az előbb ismertetett faktorizációs eljárások önmagukban nem biztosítanak a nagy hálózatok egyenleteinek gazdaságos megoldásához szükséges hatékonyságot, ha ebben megfelelő programozási módszerek nem segítenének. A legutóbbi időben kifejlesztett „ritkamátrix technikák” (sparse matrix techniques) lényegesen lecsökkentik a tárolandó adatok számát és így a szükséges memóriakapacitást. Ugyanekkor az egyenletmegoldással kapcsolatos műveleteket szintén a „komprimált” adatrendszerben végzik és így menet közben is a leggazdaságosabban használják ki az operatív memóriát.

Itt — anélkül, hogy a kérdésbe részletesen belemennénk — csupán vázolni kívánjuk ezen módszereknek alapul szolgáló elveket. Részletesebb információt nyújtanak a közzétett programleírások [7–9].

Ritka mátrixoknál csupán a nemzéró elemek kerülnek tárolásra. Egy-idejűleg tárolódnak a nemzéró elemeknek a mátrixban való elhelyezkedésére vonatkozó információk is. Itt a következő lehetőségek állnak fenn:

a) a mátrixelemek három vektorban kerülnek tárolásra, ahol az első kettő a sor, illetve oszlopindexet tartalmazza, a harmadik pedig az elemek értékeiből áll. Ennél a tárolási módnál az ún. $i-j$ tárolásnál az elemek sorrendje *tetszőleges* lehet.

b) a nemnulla elemek oszlop-, illetve sorfolytonosan vannak egyetlen vektorban (arrayben) tárolva. Az elemeknek a sorokban (illetve oszlopokban) elfoglalt helyére vonatkozó információkat egy második vektor tartalmazza. Egy harmadik vektor járulékos információkat szolgáltat arra vonatkozóan, hogy melyik a következő oszlop (illetve sor) első eleme. A tárolásnak ez a módja — az *oszlop*, illetve *sorfolytonos* tárolás —, különösen előnyös a faktorizációs eljárások szempontjából, mivel az utóbbiaknál egyszerre a mátrix egész során vagy oszlopán végzünk műveleteket.

c) az ún. bit-tárolásnál a nem-nulla elemek értékei hasonlóan tárolódnak, mint az előző esetben (oszlop-, illetve sorfolytonosan). Az elem helyzetének megjelölésére egy bit-lánc (bit-string) szolgál. Amennyiben az elem zérus, a bit-lánc a kérdéses helyen 0, amennyiben nem nulla, 1-es áll.

A G egyenletmátrix ily módon történő „komprimálásával” nyert vektorból az LU faktorizált mátrix előállítása mindenesetre bizonyos nehézségekbe ütközik. A LU elemeinek képzésére az (5) és (6) kifejezések szolgálnak. A LU j -edik oszlopának előállításához, mint az (5) és (6)-ból látható, az oszlopfolytonosan rendezett L mátrix egy egész sora szükséges. Ez nem könnyű címzési problémát jelent.

A nehézség megkerülésére több mód lehetséges: Az egyik módszer az (5), illetve a (6)-ban rögzíti a k értékét és képezi az L és U j -edik oszlopát a k rögzített értékénél. Tételezzük fel, hogy az LU mátrix $1, 2, \dots, j-1$ -edik oszlopa ily módon meghatározást nyert és pedig *komprimált* alakban, (tehát a 0-értékű elemek elhagyásával), ezeket tároltuk az l_1, l_2, \dots, l_{j-1} vektorban. A k rögzített értékére az (5) és (6)-ból a következő kifejezések adódnak:

$$\begin{aligned} u_{ij}^{(k)} &= u_{ij}^{(k-1)} / l_{kk} & i &= k \\ u_{ij}^{(k)} &= u_{ij}^{(k-1)} - l_{ik} u_{kj}^{(k)} & k < i < j \\ l_{ij}^{(k)} &= l_{ij}^{(k-1)} - l_{ik} u_{kj}^{(k)} & i < j \end{aligned} \quad (24)$$

Az i itt az l_k vektorban az elem sorszáma.

$$\begin{aligned} u_{ij}^{(0)} &= a_{ij} & i < j; \\ l_{ij}^{(0)} &= a_{ij} & i \geq j \end{aligned} \quad (24')$$

Az eljárás lehetővé teszi az LU mátrix *komprimált* oszlopmátrixának, l_k -nak egyetlen menetben való számítását.

Nagy hálózatok számítására szolgáló eddig kidolgozott programok közös jellemzőit a következőkben foglalhatjuk össze:

a) az egyenletrendszer megoldására általában *véges* faktorizációs módszereket használnak,

b) automatikus főelemkiválasztással és közel optimális rendezéssel dolgoznak,

c) komprimált formában tárolják a hálózatra vonatkozó adatokat és közbülső eredményeket. Az algebrai műveleteket közvetlenül a komprimált adatokon végzik.

A hálózatok dekompozíciója

Dekompozíciós módszereken olyan módszereket értünk, amelyek a teljes hálózat egyenleteinek megoldását a részhálózatra vonatkozó megoldások összekapcsolásával szolgáltatják. Amennyiben a dekompozíciónak tisztán matematikai megfontolások szolgálnak alapul, „particionálásról” beszélhetünk.

Abban az esetben, midőn a matematikaik mellett fizikai megfontolások is szerepet játszanak, G. Kron terminológiáját használva „diakoptikai módszerekről” beszélhetünk [10]. Ilyen fizikai megfontolás lehet a Kirchhoff-féle törvényeknek és az energiamegmaradás elvének explicit alkalmazása mellett, virtuális forrásáramok és kapcsoláselemek, stb. bevezetése. Anélkül, hogy a lehetséges megoldásokba mélyebben belemerülnénk, a két módszer típusra egy-egy reprezentatív példát ismertetünk.

A csomóponti egyenlet particionálása a következőképpen történhet:

Az (1) egyenlet

$$A'I = I', \quad A'YA = Y_N$$

jelölések bevezetésével, a következőképpen írható ($E = O$)

$$Y_N e' = I' \quad (25)$$

Ezt a következőképpen írhatjuk:

$$\begin{bmatrix} Y_{11} & Y_{12} \\ Y_{21} & Y_{22} \end{bmatrix} \begin{bmatrix} e'_1 \\ e'_2 \end{bmatrix} = \begin{bmatrix} I'_1 \\ I'_2 \end{bmatrix} \quad (26)$$

ahol az egyszerűség kedvéért feltételeztük, hogy a hálózatot két részre bontottuk: az Y_{11} és Y_{22} a részhálózatok csomóponti admittancia-mátrixai, Y_{21} , Y_{12} kapcsoló mátrixok; e'_1 , e'_2 , I'_1 , I'_2 a részhálózatok csomóponti feszültség, illetve csomóponti forrásáram vektorai.

A (7)-et kifejtve, az e'_1 számára az

$$e'_1 = Y_{11}^{-1} [I'_1 - Y_{12} e'_2] \quad (27)$$

kifejezés adódik. A (26) kifejtésével nyert második egyenletbe, az e_{11} ezen kifejezését visszahelyettesítve az e'_2 számára az

$$[Y_{22} - Y_{21} Y_{11}^{-1} Y_{12}] e'_2 = I'_2 - Y_{21} Y_{11}^{-1} I'_1 \quad (28)$$

egyenletet nyerjük. Hasonlóan az e'_1 számára az

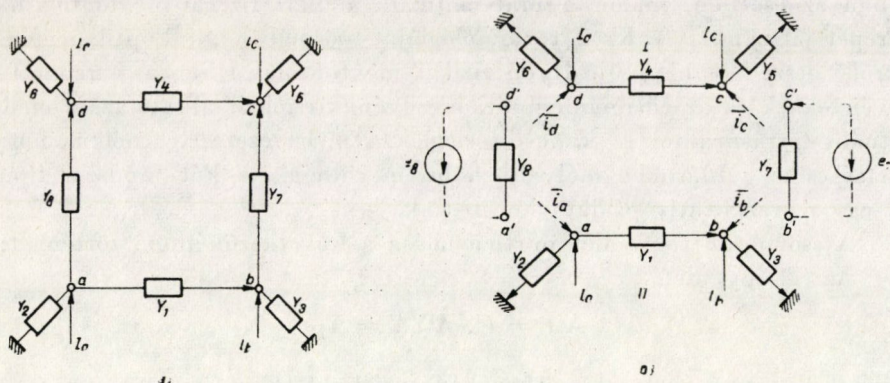
$$[Y_{11} - Y_{12} Y_{22}^{-1} Y_{21}] e'_1 = I'_1 - Y_{12} Y_{22}^{-1} I'_2 \quad (29)$$

egyenlet adódik.

A (28) és (29) egyenletek alkalmazásával a hálózati egyenlet megoldását az eredeti Y_N mátrixnál lényegesen kisebb mátrixok invertálásával sikerül végrehajtani.

A hálózat (25) csomóponti egyenletének egy *diakoptikai* megfontolások alapján történő felbontását a 3. ábrán szemléltetjük [11].

Bontsuk a hálózatot a 7 és 8 ágak kiiktatásával az I és II részhálózatokra. A 7 és 8 ágak kiiktatásának ellensúlyozására az a , b , c , d csomópontokban



3. ábra

i_a, i_b, i_c, i_d hipotetikus áramokat tételezhetünk fel, melyek nagyságát és irányát úgy vesszük fel, hogy az e'_a, e'_b, e'_c, e'_d csomóponti feszültségek ne szenvedjenek változást.

Az $i = [i_a, i_b, i_c, i_d]$ csomóponti és az $i_k = [i_7, i_8]$ kapcsolóág áramvektorok összefüggését az

$$i = C_k i_k \quad (30)$$

egyenlettel fejezhetjük ki, ahol C_k egy kapcsoló mátrix. Az $e_k = [e_7, e_8]$ kapcsolóágfeszültség és a $e' = [e'_a, e'_b, e'_c, e'_d]$ csomóponti feszültségek vektorainak összefüggését a

$$e' = -C'_k e' \quad (31)$$

kifejezéssel adhatjuk meg.

A két részhálózat csomóponti egyenletei

$$Y_{N1} e'_1 = I'_1 + e_1 \quad (32)$$

$$Y_{N2} e'_2 = I'_2 + e_2 \quad (33)$$

alakban írhatók, ahol az 1 és 2 index az I, illetve II részhálózatokra utal. A (32) és (33) egyetlen hipermátrix egyenletként is felírható:

$$Y_D e' = I' + i, \quad (33)$$

ahol

$$Y_D = \begin{bmatrix} Y_{N1} & \mathbf{0} \\ \mathbf{0} & Y_{N2} \end{bmatrix} \quad (34)$$

A kapcsolóágakra vonatkozólag a következő egyenlet adható meg:

$$Y_k^{-1} i_k = Z_k i_k = e_k \quad (35)$$

A (34) és (35) egyenletekből a (30) és (31) helyettesítésével az

$$\mathbf{X}_D e' = I' + \mathbf{C}_k i_k \quad (36)$$

$$\mathbf{Z}_k i_k = -\mathbf{C}_k^t e' \quad (37)$$

egyenletek adódnak. A (36) és a (37)-ből az i_k eliminálásával az e -ismeretlen csomóponti feszültségekre az

$$[\mathbf{Y}_D^{-1} - \mathbf{Y}_D^{-1} \mathbf{C}_k \mathbf{W}^{-1} \mathbf{C}_k^t \mathbf{Y}_D^{-1}] e' = I' \quad (38)$$

egyenlet adódik, ahol

$$\mathbf{W} = \mathbf{Z}_k + \mathbf{C}_k^t \mathbf{Y}_D^{-1} \mathbf{C}_k \quad (39)$$

A (38) egyenlet jelentősége főleg abban áll, hogy a csomóponti admittancia-mátrix helyett csak egy diagonális hipermátrixot kell invertálni, mely lényegesen könnyebben végrehajtható.

A matematikai particionálás és a diakoptikai módszerek előnyök a következő esetekben:

a) ha a hálózati egyenlet mátrixa viszonylag kevés 0-elemet tartalmaz. Ezen módszerek ugyanis általában az eredeti mátrixnál lényegesen kisebb mátrixok invertálását igénylik. Mivel a mátrixinverzió időszükséglete első közelítésben n^3 -al arányos, lényeges időmegtakarítás érhető el. (A valóságos idő-megtakarítás felmérésénél számba kell venni a dekompozícióval járó járulékos szorzási, összeadási, átrendezési, stb. műveleteket.)

b) sorozatszámításoknál, amennyiben csupán a hálózat egyes részei szenvednek változást,

c) amennyiben a hálózat nagyszámú, azonos struktúrájú részhálózatokból épült fel, mint pl. parciális differenciálegyenletek hálózati módszerrel történő megoldásánál.

A dekompozíciós módszerek értékelését a különböző szerzők — beleértve ebbe G. Kront is — ezen speciális feltételek tükrében adták meg és így túlértékelték ezen módszerek jelentőségét. Ha a dekompozíciós módszereket a mátrixok ritkaságát kihasználó legújabb módszerek fényében nézzük, a régebbi értékelések csak részben állják meg helyüket.

Az a) alatti előnyök illuzorikusak, mivel nagy hálózatoknál, mint ezt a bevezetésben láttuk, a 0-elemek száma 95% felett van. Másrészt a mátrix ritkaságát kihasználó legújabb számítógépes módszerek időszükséglete a csomópontok, illetve hurkok számával arányos, szemben a ritkaságot ki nem használó, közel ezek köbével arányos időigényű módszerekkel.

A régebbi dekompozíciós módszerek általában nem célszerűek a hálózati egyenletek megoldására, mivel olyan kifejezésekhez vezetnek, melyben mátrixinverziók szerepelnek; lásd a (28), (29) és (39) kifejezéseket. Mivel az inverz mátrixok általában *teltek*, a megoldandó egyenletek mátrixában megnő a nemnulla elemek száma és így elveszhetnek azon előnyök, amelyek a mátrixok

ritkaságát kihasználó módszerek alkalmazásából adódhatnak. Ez a hátrány általában felülmúlja azon előnyöket, melyek a megoldandó egyenletek *méretének*, a dekompozíciós módszer alkalmazásából adódó csökkenéséből származnak.

Megállapításunk nem vonatkozik azon újabb kísérletekre, melyek a hálózati egyenletek dekompozícióját ezek faktorizációjával kapcsolják össze. A hálózat egyenlet (25) csomóponti alakja a következőképpen particionálható:

$$\begin{bmatrix} \mathbf{Y}_{11} & \mathbf{Y}_{12} & \mathbf{Y}_{13} \\ \mathbf{Y}_{21} & \mathbf{Y}_{22} & \mathbf{Y}_{23} \\ \mathbf{Y}_{31} & \mathbf{Y}_{32} & \mathbf{Y}_{33} \end{bmatrix} \begin{bmatrix} e'_1 \\ e'_i \\ e'_3 \end{bmatrix} = \begin{bmatrix} I'_1 \\ I'_2 \\ I'_3 \end{bmatrix} \quad (39)$$

A baloldali hipermatrix, hasonlóan, mint közönséges mátrixoknál felírható LU-faktorizált alakban

$$\begin{bmatrix} \mathbf{L}_{11} & & \mathbf{I}^* & \mathbf{U}_{21} & \mathbf{U}_{13} \\ \mathbf{L}_{21} & \mathbf{L}_{22} & & \mathbf{I}^* & \mathbf{U}_{23} \\ \mathbf{L}_{31} & \mathbf{L}_{32} & \mathbf{L}_{33} & & \mathbf{I}^* \end{bmatrix} \begin{bmatrix} e'_1 \\ e'_2 \\ e'_3 \end{bmatrix} = \begin{bmatrix} I'_1 \\ I'_2 \\ I'_3 \end{bmatrix} \quad (40)$$

ahol \mathbf{L}_{ij} , \mathbf{U}_{ij} négyzetes mátrixok, \mathbf{I}^* egységmatrixok. A (39) megoldás formálisan egy előremenő és egy hátrafelemenő szubsztitúcióval egyenértékű.

Bevezetve az $y = [y_1, y_2, y_3]$ hipervektort, melyet a következőképpen definiálunk:

$$\begin{bmatrix} \mathbf{I}^* & \mathbf{U}_{12} & \mathbf{U}_{13} \\ & \mathbf{I}^* & \mathbf{U}_{23} \\ & & \mathbf{I}^* \end{bmatrix} \begin{bmatrix} e'_1 \\ e'_2 \\ e'_3 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} \quad (41)$$

a (39) az

$$\begin{bmatrix} \mathbf{L}_{11} & & \\ \mathbf{L}_{21} & \mathbf{L}_{22} & \\ \mathbf{L}_{31} & \mathbf{L}_{32} & \mathbf{L}_{33} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} I'_1 \\ I'_2 \\ I'_3 \end{bmatrix} \quad (42)$$

alakba írható. Az

$$\mathbf{L}_{11}y_1 = I'_1 \quad (43)$$

egyenlet formálisan a (4) egyenlettel azonos. Az \mathbf{L}_{11} LU-faktorizációjával, melyet az (5) és (6) alapján minden további nélkül végrehajthatunk és a mátrix ürességét kihasználó programozási technika alkalmazásával a (43) egyenlet az y_1 -re nézve megoldható. Az y_1 értékeinek az

$$\mathbf{L}_{22}y_2 = I'_2 - \mathbf{L}_{21}y_1 \quad (44)$$

egyenletbe való helyettesítésével a (44) egyenlet hasonlóan az y_2 -re nézve megoldható. Az

$$\mathbf{L}_{33}y_3 = I'_3 - \mathbf{L}_{31}y_1 - \mathbf{L}_{32}y_2 \quad (45)$$

egyenlet az y_3 -at szolgáltatja.

Az y_1, y_2, y_3 értékeinek a (41)-be való visszahelyettesítésével és az így adódó egyenletek megoldásával a keresett e'_1, e'_2, e' csomóponti feszültségvektorok meghatározhatók.

Könnyen belátható, hogy ez az eljárás biztosítja azokat az előnyöket, amit a dekompozíció és az ehhez kapcsolt — a mátrix ritkaságát kihasználó programozási technika együttesen jelent. Ez abból következik, hogy a folytatólagos mátrixszubsztitúció eredményeként adódó (43), (44), (45) egyenletek ugyanazon, — az L_{11}, L_{22}, L_{33} mátrixok ritkaságát kihasználó módszerrel oldhatók meg, mint a nemparticionált csomóponti egyenlet. A (43), (44), és (45) egyenletek dimenziói viszont az utóbbinál lényegesen kisebbek.

IRODALOM

1. G. W. STAGG, A. H. EL-ABIAD: *Computer Methods in Power System Analysis*, McGraw-Hill, New York, 1968.
2. A. RALSTON: *A First Course in Numerical Analysis*, McGraw Hill, New York, 1965, pp. 398—415.
3. W. F. TINNEY, I. W. WALKER: Direct Solutions of Sparse Network Equations by Optimally Ordered Triangular Factorization. *Proc. IEEE* 55 (1967), 11, 1801—1809.
4. N. SATO, W. F. TINNEY: Techniques for Exploiting the Sparsity of the Network Admittance Matrix, *IEEE Trans. Power System*, 82 (1963) pp. 944—950.
5. E. C. OGBUOBIRI, W. F. TINNEY, I. W. WALKER: Sparsity-Directed Decomposition for Gaussian Elimination of Matrices. *IEEE Trans. Power Syst.*, 89 (1970), 1, 141—150.
6. ELEK I., BOROSSAY JÓZSEFNÉ: Program gázhálózatok számítására a CDC 3300 gépre. Szóbeli közlés.
7. E. C. OGLUOBIRI: Dynamic Storage and Retrieval in Sparsity Programming. *IEEE Trans. Power Syst.*, 89 (1970), 1, 150—155.
8. M. E. CHURCHILL: A Sparse Matrix Procedure for Power System Analysis. *Proc. Oxford Conf. on Large Sparse Sets of Lin. Equations*, Academic Press, London, 1971, April, 1970.
9. K. ZOLLENKOPF: Bi-Factorisation — Basic Computational Algorithm and Programming Techniques. *Proc. Oxford Conf. on Large Sparse Sets of Lin. Equations*, Academic Press, London, 1971, April 1970, pp. 75—97.
10. G. KRON, *Diakoptics*. MacDonald, London, 1961.
11. D. SINGER: Network Theory of Bar Structures, *Acta Techn Hung.*, 73 (1972) pp. 217—235.

On Methods for Calculating Large Networks. The paper deals with methods applicable for the calculation of large networks and examines them from the point of view of economy. It investigates methods using „sparse matrix techniques” which surpass the other ones in efficiency. Furthermore it investigates decomposition methods and points out the advantages of combining decomposition and sparse matrix techniques.

Über Methoden für die Berechnung von großen Netzen. Der Aufsatz untersucht die für die Berechnung von großen Netzen in Betracht kommenden Methoden vom Standpunkt ihrer Wirtschaftlichkeit. Eingehend werden Verfahren, analysiert welche von der „sparse” Matrix-technik Gebrauch machen. Des weiteren beschäftigt sich die Arbeit mit Zerlegungsmethoden unter dem Gesichtspunkt der „sparse” Matrixtechnik, und weist auf die bisher unausgenutzten Möglichkeiten hin, welche sich aus der kombinierten Anwendung der „sparse” Matrixtechnik und der Zerlegung ergeben.