

Stora glesa matriser

I många tillämpningar behöver man hantera mycket stora matriser. Ofta så stora att hela matrisen inte får plats i primärminnet på en gång.

Om de flesta elementen i en matris är 0 kallas matrisen för gles. När man lagrar en gles matris lagrar man ofta bara de nollskilda elementen och dess positioner i matrisen.

Exempel: Låt

$$\mathbf{A}_{11} = \begin{bmatrix} 4 & -1 & 0 & 0 \\ -1 & 4 & -1 & 0 \\ 0 & -1 & 4 & -1 \\ 0 & 0 & -1 & 4 \end{bmatrix}, \mathbf{A}_{12} = \begin{bmatrix} -1 & -1 & -1 & -1 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

och

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{12}^T & \mathbf{A}_{11} \end{bmatrix} = \begin{bmatrix} 4 & -1 & 0 & 0 & -1 & -1 & -1 & -1 \\ -1 & 4 & -1 & 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & 4 & -1 & 0 & 0 & -1 & 0 \\ 0 & 0 & -1 & 4 & 0 & 0 & 0 & -1 \\ -1 & 0 & 0 & 0 & 4 & -1 & 0 & 0 \\ -1 & -1 & 0 & 0 & -1 & 4 & -1 & 0 \\ -1 & 0 & -1 & 0 & 0 & -1 & 4 & -1 \\ -1 & 0 & 0 & -1 & 0 & 0 & -1 & 4 \end{bmatrix}$$

$\mathbf{A}$ symmetrisk, 8×8 matris och 34 av elementen är nollskilda.

Om matrisen är $n \times n$ och symmetrisk kan man tolka den som en oriktad graf där noder och kanter ges av

$$V = \{1, 2, \dots, n\} \\ E = \{(i, j) \text{ om } a_{ij} \neq 0, i > j\}$$

Exempel: För $\mathbf{A}_{11}$ har vi

$$V = \{1, 2, 3, 4\} \\ E = \{(2, 1), (3, 2), (4, 3)\}$$

Antalet kanter som innehåller nod v_i kallas graden för nod v_i

Exempel: Graderna för noderna 1, 2, 3 och 4 i $\mathbf{A}_{11}$ är 1, 2, 2 och 1.

Man vill försöka behålla så mycket som möjligt av matrisens gleshet vid Gausseliminering.

Då man nollställer första kolumnen i $\mathbf{A}$ ovan ger elementen på första raden, i 5:e till 8:e kolumnen upphov till att nya nollskilda element bildas. När element som tidigare var 0 blir nollskilda då man tex. Gausseliminerar kallas det för att man fått fill in. Man vill få så lite fill in som möjligt.

Positionerna på de nollskilda elementen i en matris bestämmer hur mycket fill in man får vid Gausseliminering. Antag till exempel att vi i Gausselimineringen tänker nollställa elementen under a_{jj} (elementen under diagonalen i kolumn j) och att a_{jj}, a_{ij}, a_{jk} alla är nollskilda. Ett steg i Gausselimineringen:

$$\begin{bmatrix} a_{jj} & \dots & a_{jk} \\ \vdots & & \\ a_{ij} & \dots & 0 \end{bmatrix} \sim \begin{bmatrix} a_{jj} & \dots & a_{jk} \\ \vdots & & \\ 0 & \dots & F \end{bmatrix}$$

där $a_{ik} = F \neq 0$. Grafiskt motsvaras det av att man tagit bort kanten mellan nod j och i , och istället infört en kant mellan nod i och k .

Matrisen med följande struktur ger ingen fill in om den Gausselimineras (med x menas nollskilt element. Nollorna har inte markerats)

$$\begin{bmatrix} x & & & & & x \\ & x & & & & x \\ & & x & & & x \\ & & & x & & x \\ & & & & x & x \\ x & x & x & x & x & x \end{bmatrix}$$

Om man byter plats på 2:a och 6:e raden, 2:a och 6:e kolumnen i matrisen ovan får man

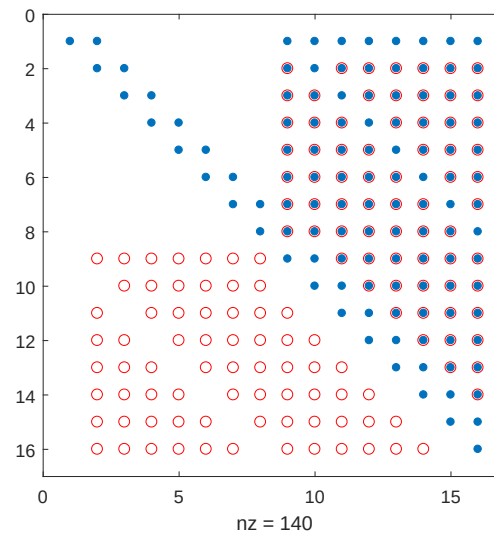
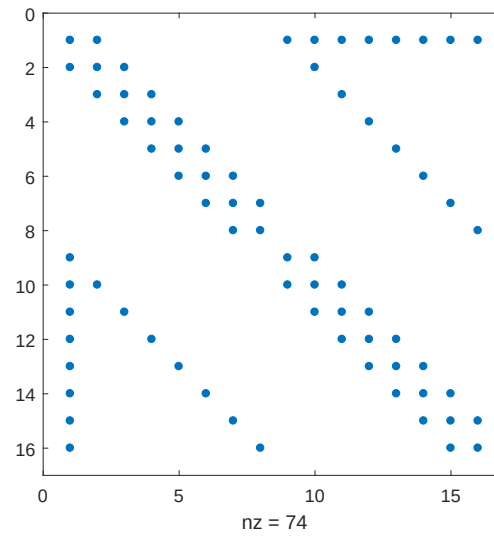
$$\begin{bmatrix} x & x & & & & \\ x & x & x & x & x & x \\ & x & x & & & \\ & x & & x & & \\ & x & & & x & \\ & x & & & & x \end{bmatrix}$$

Nu är det bara nollställning av elementen i den första kolumnen som inte ger fill in.

Det finns många algoritmer för att sortera om rader och kolumner i en matris för att minska fill in vid Gausseliminering. En av algoritmerna heter minimum degree.

Minimum degree: Välj nästa pivot element (nästa rad att eliminera från) som den nod i grafen med lägst grad (om flera noder har samma grad, välj en av dem).

I figuren nedan ser vi strukturen av en matris $\mathbf{A}$. Matrisen har skapats på samma sätt som i det första exemplet med delmatriser $\mathbf{A}_{11}$ och $\mathbf{A}_{12}$, fast vi har gjort delmatriserna 8×8 . I den översta figuren strukturen av $\mathbf{A}$ före Gausseliminering (endast nollskilda element har markerats), nedersta figuren efter Gausseliminering till trappstegsform. De röda ringarna markerar var det någon gång under Gausselimineringen bildats ett element $\neq 0$



Nedan och på nästa sida strukturen av $\mathbf{A}$, permuterad med Matlab's minimum degree. Bilden nedan före Gausseliminering och bilden på nästa sida visar efter Gausseliminering till trappstegsform.

