

DOMAIN DECOMPOSITION METHOD WITH SPARSE DIRECT SOLVER

J. Kruis¹, R. Vondráček¹, Z. Bittnar¹

Summary: *Contribution deals with the application of the sparse direct solver in methods of domain decomposition which are used for solution of large systems of equations. The DP-FETI method is assumed and shortly described. The classical LDL^T factorization used in domain decomposition methods is based on the skyline storage. The sparse direct solver is based on the compressed row storage. The elimination graph is replaced by the quotient graph. The factorization of the matrix contains two phases, symbolic and real factorization. The proposed application of the sparse direct solver reduces memory requirements and leads to shorter CPU times.*

1. Úvod

Řešení rozsáhlých inženýrských a vědeckých úloh numerickými metodami vede na soustavy algebraických rovnic s mnoha neznámými. Jedním ze způsobů efektivního řešení takových soustav jsou metody rozložení oblasti na podoblasti, nazývané též domenové dekompozice. Metod rozložení oblasti na podoblasti je celá řada, proto se v tomto příspěvku uvažují jen metody bez překryvu (non-overlapping) a metody dělicí síť konečných prvků (dělí se diskretizovaná úloha, ne spojitá formulace).

2. Metoda DP-FETI

DP-FETI je zkratka anglického názvu Dual-Primal Finite Element Tearing and Interconnecting. Metoda byla popsána např. v (Farhat a spol., 2001) a vychází z metody FETI, kde se zavádějí některá vylepšení. Metoda bude popsána na mechanické úloze. Rozdělením oblasti na podoblasti vzniknou nové hranice a všechny uzly je třeba roztřídit na tzv. vnitřní a hraniční. V metodě DP-FETI se navíc třídí i hraniční uzly na rohové a zbývající. Všechny neznámé definované v rohových uzlech (z celé konstrukce) jsou umístěny ve vektoru $\mathbf{d}^{[c]}$. Ve zbývajících hraničních uzlech se kvůli zajištění spojitosti definují Lagrangeovy multiplikátory, které jsou umístěny ve vektoru $\boldsymbol{\lambda}$. Počet všech rohových uzlových neznámých je n_c , počet všech multiplikátorů je n_m .

Vektor uzlových posunutí na j -té podoblasti je vyjádřen ve tvaru

$$\mathbf{d}_j = \mathbf{E}_j \mathbf{d}_j^{[r]} + \mathbf{F}_j \mathbf{d}^{[c]}, \quad (1)$$

¹Ing. Jaroslav Kruis, Ph.D., Ing. Richard Vondráček, Prof. Ing. Zdeněk Bittnar, DrSc., Katedra stavební mechaniky, Stavební fakulta ČVUT, Thákurova 7, 166 29 Praha 6; tel. +420 2 2435 4369, tel. +420 2 2435 4309, tel. +420 2 2435 3869, e-mail: jk@cml.fsv.cvut.cz, e-mail: bittnar@fsv.cvut.cz.

kde $\mathbf{d}_j^{[r]}$ obsahuje uzlová posunutí bez rohových posunutí, \mathbf{E}_j a \mathbf{F}_j jsou lokalizační matice a $\mathbf{d}^{[c]}$ je vektor všech rohových posunutí.

Funkcionál energie po diskretizaci má tvar

$$\Pi = \sum_{j=1}^m \left(\frac{1}{2} \mathbf{d}_j^{[r]T} \mathbf{K}_j \mathbf{d}_j - \mathbf{d}_j^{[r]T} \mathbf{f}_j + \boldsymbol{\lambda}^T \mathbf{G}_j \mathbf{d}_j^{[r]} \right), \quad (2)$$

kde se celková energie získává jako součet příspěvků z jednotlivých podoblastí. Ve funkcionalu je použito standardní označení: \mathbf{K}_j je matice tuhosti, \mathbf{f}_j je vektor uzlových zatížení a \mathbf{G}_j je lokalizační matice. Dosazením výrazu (1) do funkcionalu energie (2) vede na tvar

$$\begin{aligned} \Pi &= \sum_{j=1}^m \left(\frac{1}{2} (\mathbf{d}_j^{[r]})^T \mathbf{E}_j^T \mathbf{K}_j \mathbf{E}_j \mathbf{d}_j^{[r]} + \frac{1}{2} (\mathbf{d}_j^{[r]})^T \mathbf{E}_j^T \mathbf{K}_j \mathbf{F}_j \mathbf{d}^{[c]} + \right. \\ &+ \frac{1}{2} (\mathbf{d}^{[c]})^T \mathbf{F}_j^T \mathbf{K}_j \mathbf{E}_j \mathbf{d}_j^{[r]} + \frac{1}{2} (\mathbf{d}^{[c]})^T \mathbf{F}_j^T \mathbf{K}_j \mathbf{F}_j \mathbf{d}^{[c]} - \\ &\left. - (\mathbf{d}_j^{[r]})^T \mathbf{E}_j^T \mathbf{f}_j - (\mathbf{d}^{[c]})^T \mathbf{F}_j^T \mathbf{f}_j + \boldsymbol{\lambda}^T \mathbf{G}_j \mathbf{d}_j^{[r]} \right). \end{aligned} \quad (3)$$

Ve funkcionalu (3) je $m + 2$ neznámých vektorů. Podmínky stacionarity se získají derivováním podle jednotlivých neznámých. Podmínka rovnováhy na j -té podoblasti má tvar

$$\begin{aligned} \frac{\partial \Pi}{\partial \mathbf{d}_j^{[r]}} &= \mathbf{E}_j^T \mathbf{K}_j \mathbf{E}_j \mathbf{d}_j^{[r]} + \mathbf{E}_j^T \mathbf{K}_j \mathbf{F}_j \mathbf{d}^{[c]} - \\ &- \mathbf{E}_j^T \mathbf{f}_j + \mathbf{G}_j^T \boldsymbol{\lambda} = \mathbf{0}, \end{aligned} \quad (4)$$

jedna podmínka pro všechny rohové uzly má tvar

$$\frac{\partial \Pi}{\partial \mathbf{d}^{[c]}} = \sum_{j=1}^m \left(\mathbf{F}_j^T \mathbf{K}_j \mathbf{E}_j \mathbf{d}_j^{[r]} + \mathbf{F}_j^T \mathbf{K}_j \mathbf{F}_j \mathbf{d}^{[c]} - \mathbf{F}_j^T \mathbf{f}_j \right) = \mathbf{0} \quad (5)$$

a podmínka spojitosti je

$$\frac{\partial \Pi}{\partial \boldsymbol{\lambda}} = \sum_{j=1}^m \mathbf{G}_j \mathbf{d}_j^{[r]} = \mathbf{0}. \quad (6)$$

Vzhledem ke speciálním tvarům matic

$$\mathbf{E}_j = \begin{pmatrix} \mathbf{I} \\ \mathbf{0} \end{pmatrix}, \quad \mathbf{F}_j = \begin{pmatrix} \mathbf{0} \\ (\mathbf{F}_j^{[c]}) \end{pmatrix} \quad (7)$$

mají maticové součiny z předcházejících vztahů tvar

$$\mathbf{E}_j^T \mathbf{K}_j \mathbf{E}_j = \mathbf{K}_j^{[rr]}, \quad (8)$$

$$\mathbf{E}_j^T \mathbf{K}_j \mathbf{F}_j = \mathbf{K}_j^{[rc]} \mathbf{F}_j^{[c]}, \quad (9)$$

$$\mathbf{F}_j^T \mathbf{K}_j \mathbf{E}_j = (\mathbf{F}_j^{[c]})^T \mathbf{K}_j^{[cr]}, \quad (10)$$

$$\mathbf{F}_j^T \mathbf{K}_j \mathbf{F}_j = (\mathbf{F}_j^{[c]})^T \mathbf{K}_j^{[cc]} \mathbf{F}_j^{[c]} \quad (11)$$

a pro vektory platí

$$\mathbf{f}_j = \begin{pmatrix} \mathbf{f}_j^{[r]} \\ \mathbf{f}_j^{[c]} \end{pmatrix}, \quad (12)$$

kde

$$\mathbf{f}_j^{[r]} = \mathbf{E}_j^T \mathbf{f}_j. \quad (13)$$

Vektor $\mathbf{d}_j^{[r]}$ může být vyjádřen na každé podoblasti ve tvaru

$$\mathbf{d}_j^{[r]} = (\mathbf{K}_j^{[rr]})^{-1} \left(\mathbf{f}_j^{[r]} - \mathbf{G}_j^T \boldsymbol{\lambda} - \mathbf{K}_j^{[rc]} \mathbf{F}_j^{[c]} \mathbf{d}_j^{[c]} \right). \quad (14)$$

Pokud jsou rohové uzly zvoleny správně, matice $(\mathbf{K}_j^{[rr]})^{-1}$ existuje. Po dosazení (14) do podmínky (5) vychází

$$\begin{aligned} \sum_{j=1}^m \left(-(\mathbf{F}_j^{[c]})^T \mathbf{K}_j^{[cr]} (\mathbf{K}_j^{[rr]})^{-1} \mathbf{G}_j^T \boldsymbol{\lambda} - (\mathbf{F}_j^{[c]})^T \mathbf{K}_j^{[cr]} (\mathbf{K}_j^{[rr]})^{-1} \mathbf{K}_j^{[rc]} \mathbf{F}_j^{[c]} \mathbf{d}_j^{[c]} + \right. \\ \left. + (\mathbf{F}_j^{[c]})^T \mathbf{K}_j^{[cr]} (\mathbf{K}_j^{[rr]})^{-1} \mathbf{f}_j^{[r]} + (\mathbf{F}_j^{[c]})^T \mathbf{K}_j^{[cc]} \mathbf{F}_j^{[c]} \mathbf{d}_j^{[c]} - (\mathbf{F}_j^{[c]})^T \mathbf{f}_j^{[c]} \right) = \mathbf{0} \end{aligned} \quad (15)$$

a po dosazení do (6) vychází

$$\begin{aligned} \sum_{j=1}^m \left(\mathbf{G}_j (\mathbf{K}_j^{[rr]})^{-1} \mathbf{f}_j^{[r]} - \mathbf{G}_j (\mathbf{K}_j^{[rr]})^{-1} \mathbf{G}_j^T \boldsymbol{\lambda} - \right. \\ \left. - \mathbf{G}_j (\mathbf{K}_j^{[rr]})^{-1} \mathbf{K}_j^{[rc]} \mathbf{F}_j^{[c]} \mathbf{d}_j^{[c]} \right) = \mathbf{0}. \end{aligned} \quad (16)$$

Pro snadnější manipulaci je zavedeno následující označení:

$$\mathbf{A}_{RR} = \sum_{j=1}^m (\mathbf{F}_j^{[c]})^T \mathbf{K}_j^{[cc]} \mathbf{F}_j^{[c]} - \sum_{j=1}^m (\mathbf{F}_j^{[c]})^T \mathbf{K}_j^{[cr]} (\mathbf{K}_j^{[rr]})^{-1} \mathbf{K}_j^{[rc]} \mathbf{F}_j^{[c]}, \quad (17)$$

$$\mathbf{A}_{RM} = \sum_{j=1}^m (\mathbf{F}_j^{[c]})^T \mathbf{K}_j^{[cr]} (\mathbf{K}_j^{[rr]})^{-1} \mathbf{G}_j^T, \quad (18)$$

$$\mathbf{A}_{MR} = \sum_{j=1}^m \mathbf{G}_j (\mathbf{K}_j^{[rr]})^{-1} \mathbf{K}_j^{[rc]} \mathbf{F}_j^{[c]}, \quad (19)$$

$$\mathbf{A}_{MM} = \sum_{j=1}^m \mathbf{G}_j (\mathbf{K}_j^{[rr]})^{-1} \mathbf{G}_j^T, \quad (20)$$

$$\mathbf{b}_R = \sum_{j=1}^m (\mathbf{F}_j^{[c]})^T \mathbf{f}_j^{[c]} - \sum_{j=1}^m (\mathbf{F}_j^{[c]})^T \mathbf{K}_j^{[cr]} (\mathbf{K}_j^{[rr]})^{-1} \mathbf{f}_j^{[r]}, \quad (21)$$

$$\mathbf{b}_M = \sum_{j=1}^m \mathbf{G}_j (\mathbf{K}_j^{[rr]})^{-1} \mathbf{f}_j^{[r]}. \quad (22)$$

S použitím zavedeného označení mají podmínky (15) a (16) tvar

$$\begin{pmatrix} -\mathbf{A}_{RR} & \mathbf{A}_{RM} \\ \mathbf{A}_{MR} & \mathbf{A}_{MM} \end{pmatrix} \begin{pmatrix} \mathbf{d}^{[c]} \\ \boldsymbol{\lambda} \end{pmatrix} = \begin{pmatrix} -\mathbf{b}_R \\ \mathbf{b}_M \end{pmatrix}. \quad (23)$$

Rovnice (23) definuje redukovaný problém, ve kterém se objevují jen neznámé definované na hranici. Vektor rohových neznámých $\mathbf{d}^{[c]}$ může být vyjádřen ve tvaru

$$\mathbf{d}^{[c]} = -(\mathbf{A}_{RR})^{-1}(-\mathbf{b}_R - \mathbf{A}_{RM}\boldsymbol{\lambda}) . \quad (24)$$

Opět platí, že pro správnou volbu rohových uzlů matice $(\mathbf{A}_{RR})^{-1}$ existuje. Druhá rovnice v (23) má po dosazení (24) tvar

$$\left(\mathbf{A}_{MM} + \mathbf{A}_{MR}(\mathbf{A}_{RR})^{-1}\mathbf{A}_{RM}\right)\boldsymbol{\lambda} = \mathbf{b}_M - \mathbf{A}_{MR}(\mathbf{A}_{RR})^{-1}\mathbf{b}_R . \quad (25)$$

Soustava rovnic (25) se řeší metodou sdružených gradientů, pro kterou se matice soustavy nemusí sestavovat.

3. Řídký přímý řešič

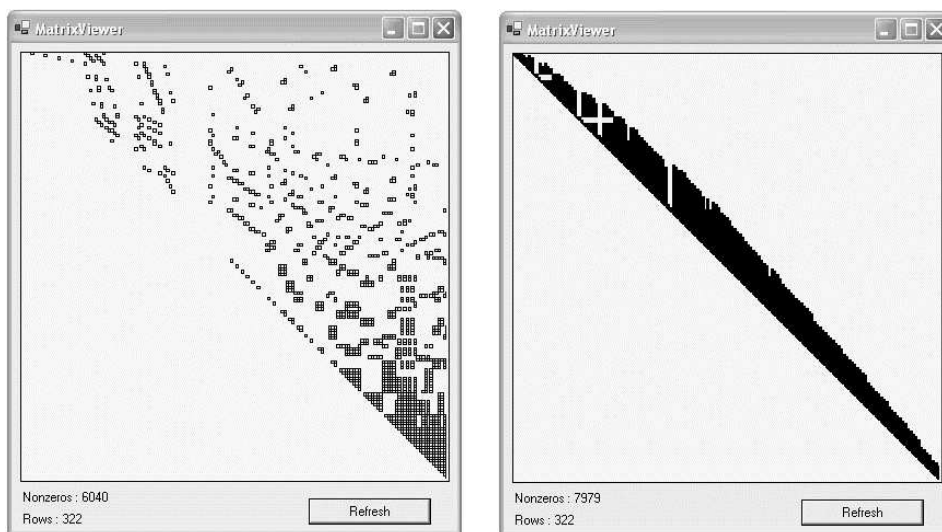
Nechť je dána soustava lineárních rovnic $\mathbf{A}\mathbf{x} = \mathbf{b}$, kde \mathbf{A} je matice odpovídající diskrétnímu lineárnímu operátoru vycházejícímu z metody konečných prvků. Tato matice je typicky řídká. Pokud je matice \mathbf{A} symetrická, lze provést dekompozici $\mathbf{A} = \mathbf{L}\mathbf{D}\mathbf{L}^T$, kde \mathbf{D} je diagonální matice a \mathbf{L} je dolní trojúhelníková matice s jedničkami na diagonále. Důležitým faktem je, že \mathbf{L} má více nenulových hodnot než \mathbf{A} . V průběhu přímého chodu Gaussovy eliminace totiž nevyhnutelně dojde k jistému částečnému zaplnění matice novými nenulovými hodnotami.

Obzvláště zajímavým faktem však je to, že rozsah zaplnění přímo souvisí s primárním setříděním neznámých v matici soustavy. Běžné řešiče založené na uložení matice ve formátu skyline se snaží minimalizovat zaplnění umístěním všech nenulových hodnot do profilu skyline. Jedná se vlastně o diagonální pás proměnné šířky, který by měl být co možná nejmenší. Tento způsob uložení vyžaduje pouze jedno celočíselné pole uchovávající hodnoty výšky jednotlivých sloupců. Hodnoty ve sloupcích jsou uloženy jednoduše jako vektory, což umožňuje velmi efektivní běh vnitřní smyčky Gaussovy eliminace. Nicméně, setřídění matice s ohledem na minimální profil je obvykle něco jiného, než setřídění, které vede na minimální zaplnění.

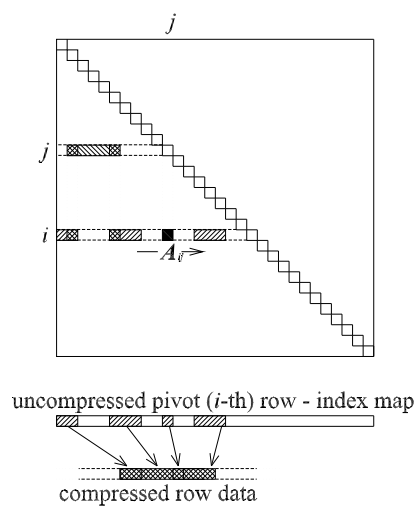
Řešiče založené na uložení matice ve formátu kompresovaných řádků naopak umožňují použití setřídění s ohledem na minimální zaplnění. Toto schéma používá o něco složitější datovou strukturu, která vyžaduje uložení sloupcových indexů nenulových hodnot v každém řádku. Také běh vnitřní smyčky je, narozdíl od skyline, zatížen jistou režií spojenou s kontrolou indexů. Od určité velikosti úlohy se však začne projevovat řádově menší množství nenulových hodnot, což vede k podstatně nižším nárokům na paměť a výrazné redukci aritmetických operací s reálnými čísly. Výsledkem je značné urychlení faktorizace, stejně jako následné zpětné substituce.

3.1 Symbolická faktorizace

Nalezení optimálního setřídění matice vzhledem k minimálnímu zaplnění je naneštěstí NP-úplný problém. Prakticky to znamená, že pro hledání optimálního setřídění s ohledem na minimální zaplnění jsme nuceni používat nějaké heuristické algoritmy. Řídký přímý řešič používá algoritmus Approximate Minimum Degree (AMD), jak jej navrhli P. Amestoy, T. A. Davis, a I. S. Duff v (Duff a spol., 1994).



Obrázek 1: Rozdílné uložení pomocí kompresovaných sloupců a skyline.



Obrázek 2: Schéma vnitřní smyčky faktorizace na řídké matici.

Struktura nenulových čísel v řídké symterické matici může být znázorněna jako neorientovaný graf $G = (V, E)$, kde vrcholy V znázorňují jednotlivé neznámé a hrany E odpovídají nenulovým hodnotám v jednotlivých řádcích. V průběhu přímého chodu Gaussovy eliminace se postupně z grafu odebírají jednotlivé vrcholy a přidávají se nové hrany. Tyto nově přidané hrany odpovídají zaplnění. Ve snaze maximálně omezit výsledné zaplnění se vybírá vždy jako nový pivot ten vrchol, který má v daný okamžik nejmenší množství přilehlých hran.

Tento model faktorizace je založen na myšlence eliminačního grafu. Nepříjemná vlastnost tohoto přístupu tkví v předem neodhadnutelném růstu spotřebované paměti spolu s poměrně vysokými nároky na počet logických operací.

Kvocientní graf (quotient graph) nabízí lepší variantu uložení nenulové struktury řídké matice. Kvocientní graf $\mathcal{G} = (V, \overline{V}, E, \overline{E})$ obsahuje množinu proměnných V (neodeeliminované vrcholy), množinu elementů \overline{V} (odeliminované vrcholy), množinu hran mezi proměnnými E a množinu hran mezi proměnnými a elementy \overline{E} .

Eliminací jednoho vrcholu grafu vznikne ze všech vrcholů jemu přilehlých plně propojený podgraf (clique). Zatímco je tato struktura v eliminačním grafu reprezentována skutečným přidáním všech hran ($O(n_v^2)$ záznamů) v kvocientním grafu jsou uloženy pouze příslušné vrcholy ($O(n_v)$ záznamů).

Kvocientní graf je úsporná reprezentace eliminačního grafu. Zatímco eliminační graf obsahuje množinu $Adj(i)$ vrcholů přilehlých k vrcholu i explicitně, kvocientní graf vyžaduje definici následujících množin:

$$\begin{aligned}\mathcal{A}_i &\equiv \{j : (i, j) \in E\} \subseteq V, \\ \mathcal{E}_i &\equiv \{e : (i, e) \in \overline{E}\} \subseteq \overline{V}, \\ \mathcal{L}_e &\equiv \{i : (i, e) \in \overline{E}\} \subseteq V.\end{aligned}$$

Množina \mathcal{A}_i obsahuje všechny proměnné přilehlé k proměnné i . Použitá notace naznačuje, že \mathcal{A} odpovídá nenulovým hodnotám v i -tém řádku matice \mathbf{A} . \mathcal{E}_i je množina všech elementů přilehlých k proměnné i . Množina \mathcal{L}_e obsahuje všechny proměnné přilehlé k elementu e . Prvky této množiny souvisí s nenulovými hodnotami i -tého řádku výsledné matice \mathbf{L} .

Pomocí těchto množin můžeme vyjádřit množinu $Adj_G(i)$ všech proměnných přilehlých k proměnné i . Velikost $|Adj_G(i)|$ množiny všech proměnných přilehlých proměnné i úzce souvisí se skutečným stupněm proměnné i .

$$Adj_G(i) = \left(\mathcal{A}_i \cup \bigcup_{e \in \mathcal{E}_i} \mathcal{L}_e \right) \setminus \{i\}. \quad (26)$$

Toto znamená, že ostatní proměnné jsou v kvocientním grafu spojené s proměnnou i buď přímo přes množinu \mathcal{A}_i , nebo přes sjednocení množin \mathcal{L}_e pro každý element $e \in \mathcal{E}_i$, který přiléhá k proměnné i .

Pomocí vzorce (26) se nalezne proměnná $p \in V$ minimalizující $|Adj_G(p)|$. Tato je vybrána jako pivot. Nejprve je odstraněna z množiny V a potom přidána do množiny \overline{V}

```

for i=1 to n
  for j=1 to i-1
     $\mathbf{A}_{ij} = \mathbf{A}_{ij} - \sum_{k=1}^{j-1} \mathbf{A}_{ik} \mathbf{A}_{jk}^T$ 
     $\mathbf{A}_{ii} = \mathbf{A}_{ii} - \sum_{k=1}^{i-1} \mathbf{A}_{ik} \mathbf{D}_{kk}^{-1} \mathbf{A}_{ik}^T$ 
  for k=1 to i-1
     $\mathbf{A}_{ik} = \mathbf{A}_{ik} \mathbf{D}_{kk}^{-1}$ 

```

Tabulka 1: Algoritmus blokové \mathbf{LDL}^T faktorizace.

jakožto nový element. Přičemž hrany vzájemně spojující proměnné z množiny \mathcal{A}_p jsou odstraněny jako nadále zbytečné, neboť jejich propojení je od nyní realizováno přes nově založený element $p \in \bar{V}$. Na základě (26) je vytvořena nová množina \mathcal{L}_p . Ze vztahu (26) též plyne, že všechny elementy z množiny \mathcal{E}_p už teď obsahují duplicitní informace o konektivitě a jakožto zbytečné jsou odstraněny včetně všech referencí na ně. Odstraněné elementy jsou absorbovány novým elementem p . Množiny \mathcal{A}_p a \mathcal{E}_p jsou zrušeny a algoritmus pokračuje hledáním nového pivotu.

Výsledná implementace AMD obsahuje ještě koncept superproměnných i a odhad horní meze stupně proměnné. Superproměnná je množina navzájem nerozlišitelných proměnných, přičemž i a j jsou nerozlišitelné právě tehdy, když $\text{Adj}_G(i) \cup \{i\} = \text{Adj}_G(j) \cup \{j\}$. Oba přístupy dramaticky urychlují běh celého algoritmu.

Detailnější popis celého algoritmu výrazně přesahuje rámec vymezený tomuto textu a lze ho nalézt např. v (Duff a spol., 1994).

3.2 Uložení matice pomocí kompresovaných řádků

Vzhledem k povaze některých fyzikálních úloh popsanych metodou konečných prvků můžeme matici \mathbf{A} ukládat jako blokově řídkou matici. Nenulový vzor matice \mathbf{A} je ovlivněn konektivitou na úrovni uzlů sítě a protože v jednom uzlu sítě může být m vzájemně svázaných stupňů volnosti, má smysl uvažovat jednotlivé nenulové komponenty matice jako bloky $\mathbf{A}_{ij}[m \times m]$.

Je-li určeno pořadí, ve kterém se budou eliminovat jednotlivé neznámé a je-li známa výsledná struktura matice \mathbf{L} , dekompozice matice se hledá ve tvaru :

$$\begin{bmatrix} \mathbf{A}_{11} & \cdots & \mathbf{A}_{n1}^T \\ \vdots & \ddots & \vdots \\ \mathbf{A}_{n1} & \cdots & \mathbf{A}_{nn} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & & \mathbf{0} \\ & \ddots & \\ \mathbf{L}_{n1} & \cdots & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{D}_{11} & & \\ & \ddots & \\ & & \mathbf{D}_{nn} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \cdots & \mathbf{L}_{n1}^T \\ & \ddots & \vdots \\ \mathbf{0} & & \mathbf{I} \end{bmatrix} \quad (27)$$

Croutův algoritmus uvedený v tabulce 1 provádí požadovanou dekompozici. Lze jej odvodit ze vztahu (27) rozepsaného pro složku \mathbf{A}_{ij} .

Na vstupu je symetrická matice \mathbf{A} , z níž je třeba pouze její dolní polovina. Algoritmus postupně upravuje její hodnoty tak, že výsledkem je matice \mathbf{D} na diagonále a v dolním trojúhelníku je uložena matice \mathbf{L} . Což lze schematicky naznačit takto:

| | | skyline | | sparse direct | |
|-----|---------|-----------|---------|---------------|-----------|
| NSD | TNDOF | SKY | RED SKY | PNP před ŘŘ | PNP po ŘŘ |
| 2x2 | 115 680 | 7 245 827 | 70 | 272 996 | 2 261 380 |
| 3x3 | 259 920 | 7 245 827 | 200 | 272 996 | 2 261 380 |
| 4x4 | 461 760 | 7 245 827 | 424 | 272 996 | 2 261 380 |

Tabulka 2: Paměťové nároky metody DP-FETI.

$$\begin{bmatrix} \mathbf{A}_{11} & & & & \\ \mathbf{A}_{21} & \mathbf{A}_{22} & & & \\ \vdots & & \ddots & & \\ \mathbf{A}_{n1} & \mathbf{A}_{n2} & \cdots & \mathbf{A}_{nn} & \end{bmatrix} \xrightarrow{\text{faktorizace}} \begin{bmatrix} \mathbf{D}_{11} & & & & \\ \mathbf{L}_{21} & \mathbf{D}_{22} & & & \\ \vdots & & \ddots & & \\ \mathbf{L}_{n1} & \mathbf{L}_{n2} & \cdots & \mathbf{D}_{nn} & \end{bmatrix} \quad (28)$$

Při běhu tohoto algoritmu se nejvíce času stráví při počítání výrazu $\sum_{k=1}^{k < i} \mathbf{A}_{ik} \mathbf{A}_{jk}^T$. Zde se provádí součiny těch elementů, které jsou nenulové zároveň v i -tém a j -tém řádku. Vhodná implementace této smyčky je proto nezbytnou ingrediencí efektivního řídkého přímého řešiče.

K dosažení takové implementace je použito pomocné indexové pole délky n , do nějž jsou dočasně dekomprimovány sloupcové indexy i -tého řádku. Toto pole je vlastně indexová tabulka obsahující ukazatele na jednotlivé záznamy sloupcových indexů v poli kompresovaného řádku. Pokud řádek neobsahuje sloupcový index, je na jeho pozici hodnota -1.

Ovšem, tato režie spojená s nepřímou adresací dat při faktorizaci na řídké matici může být z velké části vykoupena blokovými operacemi. Zde se vlastně jedná o elementární operace s relativně malými plnými maticemi, jež mohou být implementovány velmi efektivně, jak s ohledem na paralelní zpracování aritmetických operací, tak i s ohledem na využití vyrovnávací paměti (cache).

4. Numerické příklady

Chování metody DP-FETI s využitím řídkého přímého řešiče bude předvedeno na deskové úloze definované na pravidelné oblasti rozdělené na pravidelné podoblasti. Všude je použita strukturovaná síť konečných prvků. Deska je modelována Mindlinovou teorií.

Tabulka 2 obsahuje základní informace o řešeném problému. Ve sloupcích jsou po řadě uvedeny tyto informace: počet podoblastí, celkový počet neznámých v celé úloze, počet uložených prvků v matici na jedné podoblasti metodou skyline, počet prvků v matici redukovaného problému, počet uložených nenulových prvků matice jedné podoblasti před spuštěním řídkého řešiče a počet uložených nenulových prvků (PNP) matice jedné podoblasti po proběhnutí řídkého řešiče (ŘŘ).

Výpočetní časy metody DP-FETI jsou uvedeny v tabulce 3. Pro každý výpočet jsou uvedeny dva časy. Čas potřebný pro eliminaci vnitřních neznámých a čas potřebný pro řešení redukovaného problému. První dvojice platí pro klasické využití \mathbf{LDL}^T rozkladu

| | LDL^T | | ŘŘ | |
|-----|---------|---------|------|---------|
| NSD | ELIM | RED SYS | ELIM | RED SYS |
| 2x2 | 15 | 37 | 11 | 21 |
| 3x3 | 15 | 61 | 11 | 33 |
| 4x4 | 16 | 90 | 11 | 53 |

Tabulka 3: Časové nároky metody DP-FETI.

ve skyline, druhá dvojice platí pro LDL^T rozklad využívající kompresovaných řádků.

5. Poděkování

Tato práce byla podporována projektem S 308 6102 AV ČR.

6. Literatura

Davis, T.A., Amestoy, P. & Duff, I.S. (1994) *An Approximate Minimum Degree Ordering Algorithm*. Computer and Information Sciences Dept., University of Florida, Technical Report TR-94-039.

Farhat, C., Lesoinne, M., LeTallec, P., Pierson, K. & Rixen D. (2001) FETI-DP: A Dual-Primal Unified FETI Method–Part I: A Faster Alternative to the Two-level FETI Method. *International Journal for Numerical Methods in Engineering* 50, 7, pp. 1523-1544.