

Zdeněk Dostál

Rozděl a slep aneb jak řešit soustavu s bilionem lineárních rovnic

Pokroky matematiky, fyziky a astronomie, Vol. 63 (2018), No. 1, 28–40

Persistent URL: <http://dml.cz/dmlcz/147207>

Terms of use:

© Jednota českých matematiků a fyziků, 2018

Institute of Mathematics of the Czech Academy of Sciences provides access to digitized documents strictly for personal use. Each copy of any part of this document must contain these *Terms of use*.



This document has been digitized, optimized for electronic delivery and stamped with digital signature within the project *DML-CZ: The Czech Digital Mathematics Library*
<http://dml.cz>

Rozděl a slep aneb jak řešit soustavu s bilionem lineárních rovnic

Zdeněk Dostál

Abstrakt. Cílem článku je naznačit úlohu matematiky a efektivnost nových algoritmů pro řešení rozsáhlých soustav lineárních rovnic na současných masívně paralelních superpočítačích. Na příkladu řešení Poissonovy rovnice je popsána základní varianta metody rozložení oblasti typu FETI (finite element tearing and interconnecting) s projektorem na přirozenou hrubou síť, jsou odvozeny základní kvalitativní výsledky demonstrující asymptoticky lineární (optimální) složitost řešení a jsou popsána prakticky důležitá zdokonalení. Slabá i silná numerická škálovatelnost je demonstrována numerickými experimenty s řešením soustav s více než dvěma sty miliardami neznámých.

Mnoho praktických úloh vede na řešení soustavy lineárních rovnic

$$\begin{array}{cccccc} a_{11}x_1 & + & \cdots & + & a_{1n}x_n & = & b_1, \\ \vdots & & \vdots & & \vdots & & \vdots \\ a_{n1}x_1 & + & \cdots & + & a_{nn}x_n & = & b_n, \end{array} \quad (1)$$

kteřou můžeme zapsat stručněji ve tvaru

$$Ax = b, \quad \text{kde} \quad A = [a_{ij}], \quad b = [b_i], \quad x = [x_i], \quad i, j = 1, \dots, n.$$

Zde se budeme zabývat řešením velkých ($n \gg 10^9$) soustav (1) pocházejících z diskretizace eliptických parciálních diferenciálních rovnic (PDR), které popisují například ustálené vedení tepla, rovnováhu pružných těles nebo potenciál elektromagnetického pole. Matice A je v našem případě řádká s malým počtem (do sta) nenulových prvků na řádku. Navíc je symetrická a pozitivně definitní (SPD), tj. $A = A^T$ a $x^T Ax > 0$ pro $x \neq 0$, takže řešení $\hat{x} \in \mathbb{R}^n$ soustavy (1) splňuje nutnou a postačující podmínku

$$f(\hat{x}) = \min_{x \in \mathbb{R}^n} f(x), \quad f(x) = \frac{1}{2}x^T Ax - b^T x. \quad (2)$$

Připomeňme, že gradient f v bodě x je vektor $\nabla f(x) = Ax - b$ a že podmínku minima f v bodě x můžeme psát ve tvaru

$$\nabla f(x)^T d = 0, \quad d \in \mathbb{R}^n.$$

Tuto podmínku minima můžeme snadno odvodit pomocí Taylorovy řady pro přírůstek f v bodě x , která se v našem případě redukuje na identitu

$$f(x+d) - f(x) = (Ax - b)^T d + \frac{1}{2}d^T Ad = \nabla f(x)^T d + \frac{1}{2}d^T \nabla^2 f(x) d. \quad (3)$$

Prof. RNDr. ZDENĚK DOSTÁL, DSc., Katedra aplikované matematiky a Národní superpočítačové centrum, VŠB-TU v Ostravě, 17. listopadu 15, 708 33 Ostrava-Poruba, e-mail: zdenek.dostal@vsb.cz

Řešení velkých soustav je aktuálním předmětem výzkumu — Google nabízí na dotaz „algorithms for large system of linear equations“ více než dvanáct milionů odkazů. Úspěšný řešič musí umět využít veškeré dostupné informace o matici A , zejména informace získané analýzou vztahu mezi (1) a původní PDR. Ideální algoritmus by rozkládal výpočet na velké množství nezávislých částí bez vzájemné komunikace, které by prováděly aritmetické operace s argumenty uloženými v paměti v souladu s hierarchickou strukturou moderních superpočítačů.

Zde se budeme věnovat variantě jedné z nejúspěšnějších metod rozložení oblasti bez překrytí typu FETI (Finite Element Tearing and Interconnecting), která do značné míry splňuje výše uvedené požadavky. Metodu poprvé publikovali Farhat a Roux [6] počátkem devadesátých let s cílem využít začínající paralelní počítače a usnadnit rozdělení práce na diskretizaci výpočetních modelů letadel; po doplnění projektorem na „přirozenou hrubou síť“ dokázali později Farhat, Mandel a Roux [5] její asymptoticky lineární složitost. Samotná myšlenka rozdělit oblast, na které je daná PDR definována, na podoblasti a iterovat je známa nejméně od roku 1881, kdy Schwarz publikoval vůbec první práci o metodách rozložení oblasti [11] s překrytím. Je pozoruhodné, že jeho práce, která vznikla mnoho let před rozšířením počítačů, se stala základem algoritmů vhodných pro paralelní řešení rozsáhlých soustav (1).

1. Finitní a iterační metody

Připomeňme nejprve hlavní charakteristiky dvou základních tříd metod řešení soustavy (1). Menší soustavy lze efektivně řešit pomocí vhodných finitních metod jako je Gaussova eliminace, s jejichž variantami se lze setkat už na střední škole. Počet aritmetických operací potřebných k získání řešení \hat{x} soustavy (1) je obecně omezen konstantním násobkem n^3 ; alternativně mluvíme o složitosti řešení $O(n^3)$. Tyto postupy lze pro naše soustavy upravit tak, že dosáhnou složitosti $O(n^2)$ — i to však znamená, že kdybychom vyřešili soustavu s milionem rovnic za jednu vteřinu, trvalo by řešení soustavy s bilionem rovnic na stejném počítači bilion vteřin, tedy více než 10 000 let.

Pro řešení větších soustav s řídkou maticí A je možné použít iterační metody. Připomeňme si, že základní iterační metody jsou založeny na tom, že nepotřebujeme znát řešení soustavy (1) přesně, takže vystačíme s algoritmy, které se k řešení dokážou přiblížit s libovolnou přesností. Pro ilustraci se nechme inspirovat vzorcem pro součet členů geometrické posloupnosti s kvocientem $q \neq 1$ ve tvaru

$$1 + q + q^2 + \dots + q^k = \frac{1 - q^{k+1}}{1 - q}. \quad (4)$$

Pro $q = 1 - \lambda$, kde $\lambda \in (0, 1]$, dostaneme

$$1 + (1 - \lambda) + (1 - \lambda)^2 + \dots + (1 - \lambda)^k = \lambda^{-1} - (1 - \lambda)^{k+1} \lambda^{-1}, \quad (5)$$

takže řešení skalární rovnice

$$\lambda x = b \quad (6)$$

můžeme dostat ve tvaru

$$\hat{x} = \lambda^{-1} b = b + (1 - \lambda)b + (1 - \lambda)^2 b + \dots + (1 - \lambda)^k b + r_k b, \quad r_k = (1 - \lambda)^{k+1} \lambda^{-1}.$$

Pro dostatečně velké k bude r_k malé a iterace x^k generované algoritmem

$$x^0 = b, \quad x^{i+1} = b + (1 - \lambda)x^i = b + (1 - \lambda)b + \dots + (1 - \lambda)^i b \quad (7)$$

budou aproximovat řešení \hat{x} s malou chybou

$$|x^k - \hat{x}| = |r_k||b|.$$

Iterace x^k se budou k \hat{x} blížit tím rychleji, čím je větší $\lambda \in (0, 1]$.

Iterační metoda (7) používá k výpočtu iterací pouze násobení a sčítání, což jsou operace, které můžeme provádět efektivně i s velkými řídkými maticemi. Abychom ukázali souvislost s řešením (1) detailněji, připomeňme větu o spektrálním rozkladu, podle které lze každou reálnou symetrickou matici A zapsat ve tvaru

$$A = U \Lambda U^T, \quad \Lambda = \text{diag}(\lambda_1, \dots, \lambda_n) = \begin{bmatrix} \lambda_1 & & O \\ & \ddots & \\ O & & \lambda_n \end{bmatrix}, \quad UU^T = U^T U = I,$$

takže

$$A^k = U \Lambda^k U^T, \quad A^{-1} = U \Lambda^{-1} U^T, \\ I = U I U^T, \quad (I - A)^k = U (I - \Lambda)^k U^T.$$

Množinu $\sigma(A) = \{\lambda_1, \dots, \lambda_n\}$ nazýváme spektrem matice A , neboť ji na A není vidět, ale s její pomocí lze mnohé vysvětlit — je jako duch na spiritistické seanci. Rovnici (1) můžeme přepsat do tvaru

$$U \Lambda U^T x = b. \quad (8)$$

Přenásobíme-li (8) zleva maticí U^T a označíme $y = U^T x$, $c = U^T b$, dostaneme

$$\Lambda y = c,$$

takže (8) se rozpadne na n skalárních rovnic

$$\lambda_j y_j = c_j, \quad j = 1, \dots, n. \quad (9)$$

Tyto skalární rovnice můžeme řešit algoritmem (7), pokud platí $\sigma(A) \subseteq (0, 1]$. Když algoritmus (7) přepíšeme pomocí diagonálních matic Λ a I , bude výsledný algoritmus generovat současně všechny komponenty řešení y_j . Dostaneme

$$y^0 = c, \quad y^{i+1} = c + (I - \Lambda)y^i,$$

$$\hat{y} = c + (I - \Lambda)c + \dots + (I - \Lambda)^k c + r^k c = y^k + r^k c, \quad r^k = (I - \Lambda)^{k+1} \Lambda^{-1}. \quad (10)$$

Získali jsme tak aproximaci y^k řešení \hat{y} soustavy (9). Jak ale získat aproximaci x^k řešení $\hat{x} = U \hat{y}$, když o U a Λ víme pouze to, že existují? Dosaďme $U^T b$ za c , vynásobme (10) zleva U a označme $x^k = U y^k$. Dostaneme

$$\hat{x} = b + U(I - \Lambda)U^T b + \dots + U(I - \Lambda)^k U^T b + R_k b, \quad R_k = U(I - \Lambda)^{k+1} U^T U \Lambda^{-1} U^T, \quad (11)$$

a použijeme-li ještě $UU^T = I$,

$$\hat{x} = b + (I - A)b + \dots + (I - A)^k b + R_k b = x^k + R_k b, \quad R_k = (I - A)^{k+1} A^{-1}. \quad (12)$$

Iterace x^k ovšem můžeme počítat induktivně pomocí vztahů

$$x^0 = b, \quad x^{i+1} = b + (I - A)x^i. \quad (13)$$

K získání přibližného řešení x^k je třeba k násobení maticí A . Snadno se ověří, že

$$\|x^k - \hat{x}\| = \|R_k b\| \leq (1 - \lambda_{\min})^{k+1} \lambda_{\min}^{-1} \|b\|, \quad \lambda_{\min} = \min\{\lambda_1, \dots, \lambda_n\}, \quad (14)$$

kde $\|\cdot\|$ značí eukleidovskou normu (platí $\|r^k\| = \|R_k\|$). Jestliže $\sigma(A) \not\subseteq (0, 1]$, stačí soustavu (1) přenásobit vhodnou kladnou konstantou, nejlépe $\|A\|^{-1} = \lambda_{\max}^{-1}$, kde $\lambda_{\max} = \max\{\lambda_1, \dots, \lambda_n\}$. Jelikož $\sigma(\|A\|^{-1}A) \subseteq (\lambda_{\min}/\lambda_{\max}, 1]$, rychlost konvergence bude v tomto případě klesat s rostoucím číslem podmíněnosti

$$\kappa(A) = \lambda_{\max}/\lambda_{\min}.$$

Číslo podmíněnosti je základní charakteristika rychlosti konvergence iteračních metod. Máme-li třídu úloh s SPD maticemi $A^t \in \mathcal{A}$, $t \in \mathcal{T}$, které splňují $\sigma(A^t) \subseteq [a, b]$, $[a, b] \subseteq (0, 1]$ nebo $\kappa(A^t) \leq C$, $t \in \mathcal{T}$, plyne z (14), že pro získání přibližného řešení jakékoliv soustavy s maticí $A^t \in \mathcal{A}$ s danou relativní přesností $\|R_k\|$ potřebujeme stejnoměrně omezený počet násobení maticí A^t . Když mají matice A^t navíc stejnoměrně (nezávisle na $t \in \mathcal{T}$) omezený počet nenulových prvků na řádku, pak vektor x^k můžeme dostat pomocí $O(n)$ aritmetických operací. Hlavní myšlenkou FETI je využít finitní metody a informace o příslušném spojitém problému k transformaci třídy špatně podmíněných soustav, jež vznikly diskretizací dané PDR, na třídu dobře podmíněných soustav s řídkou maticí a se strukturou, která je vhodná k efektivní paralelní implementaci.

2. Modelová úloha a rozložení oblasti

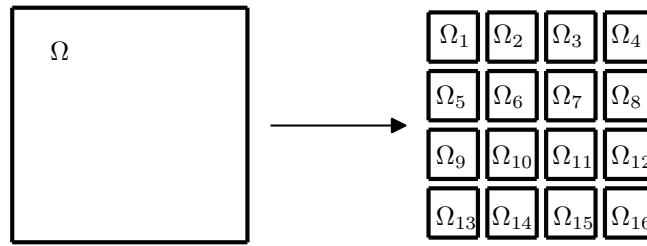
Pro jednoduchost se omezíme na Poissonovu rovnici (popisující ustálené vedení tepla)

$$-\Delta u = -\sum_{i=1}^d \frac{\partial^2 u}{\partial x_i^2} = f, \quad x \in \Omega, \quad u(x) = 0, \quad x \in \Gamma, \quad (15)$$

kde Γ je hranice oblasti $\Omega = (0, 1)^d$, $d \in \{1, 2, 3\}$. Experimenty budeme provádět pro $d = 3$, pro výklad a ilustraci budeme používat $d = 1, 2$.

První krok FETI spočívá v rozložení Ω do $s = 1/H^d$ podoblastí Ω_i se stranou (délkou) H a hranicí Γ_i , tak jako na obrázku 1 pro $d = 2$. Na každé podoblasti Ω_i zvolme pravidelnou síť se stejným diskretizačním parametrem (krokem) h a uzly x_j^i , takže sítě na sousedních podoblastech budou na sebe navazovat. Pro každou Ω_i sestavíme symetrické pozitivně semidefinitní (SPS) lokální matice tuhosti A_i a lokální vektory pravých stran b^i . Dále položíme

$$A = \text{diag}(A_1, \dots, A_s), \quad u = \begin{bmatrix} u^1 \\ \vdots \\ u^s \end{bmatrix}, \quad b = \begin{bmatrix} b^1 \\ \vdots \\ b^s \end{bmatrix}, \quad i = 1, \dots, s.$$



Obr. 1. Rozložení oblasti Ω na podoblasti

Matice A_i můžeme získat pomocí vhodné konečněprvkové báze nebo pomocí konečných diferencí. Například pro $d = 1$ tak v druhém případě využijeme aproximace

$$\Delta u^i(x_j^i) \cong \frac{1}{h^2} (u^i(x_j^i - h) - 2u^i(x_j^i) + u^i(x_j^i + h)) = \frac{1}{h^2} (u_{j-1}^i - 2u_j^i + u_{j+1}^i)$$

a další vlastnosti spojitě úlohy, takže dostaneme

$$A_i = \frac{1}{h^2} \begin{bmatrix} 1 & -1 & & & & \\ -1 & 2 & -1 & & & \\ & & \ddots & \ddots & \ddots & \\ & & & -1 & 2 & -1 \\ & & & & -1 & 1 \end{bmatrix}, \quad b_j^i = f(x_j^i); \quad i = 1, \dots, s.$$

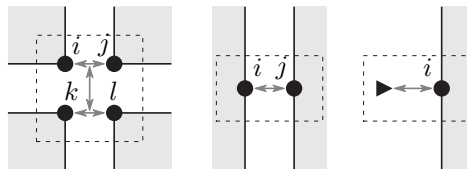
V našem případě jsme krajní řádky a sloupce matice A_i doplnili tak, aby A_i byla SPS s nulovým prostorem $\text{Ker } A_i$ (jádreem) tvořeným násobky

$$e_i = [1, \dots, 1]^T, \quad i = 1, \dots, s.$$

Řešení diskretizovaného problému (15) v minimalizační formulaci (2) se tak redukuje na řešení úlohy kvadratického programování najít

$$\min_{u \in \mathbb{R}^n} \left(\frac{1}{2} u^T A u - b^T u \right) \quad \text{pro } B u = o, \quad (16)$$

kde matice $B \in \mathbb{R}^{m \times n}$ reprezentuje „slepowání“ podoblastí a vynucení hodnot předepsaných okrajovou podmínkou (15). Typické situace pro $d = 2$ jsou na obrázku 2.



Obr. 2. Spojení podoblastí v rozích a na hranách a vynucení nulového posunutí

Předepsané hodnoty u_i na hranici Γ (jako $u_i = 0$ na obrázku 2 vpravo) lze vynutit pomocí nějakého řádku b_{ℓ^*} matice B , který je totožný s i -tým řádkem jednotkové matice příslušné dimenze, neboť v tomto případě $b_{\ell^*}u = 0$ je ekvivalentní $u_i = 0$. Podobně se ztotožní u_i a u_j ve vnitřních bodech společné hranice sousedních podoblastí pomocí řádku s vhodně umístěnou jedničkou a minus jedničkou. Ztotožnění řešení ve vnitřních rozích sousedících podoblastí (jako na obrázku 2 vlevo) lze zajistit submaticí $B_{\mathcal{I}^*}$ matice B s nenulovými prvky umístěnými ve třech řádcích a čtyřech sloupcích odpovídajících globálním indexům rohových složek, tedy

$$B_{\mathcal{I}^*} = \begin{bmatrix} & i & & j & & k & & l & & \\ \dots & 1 & \dots & -1 & \dots & 0 & \dots & 0 & \dots & \\ \dots & 0 & \dots & 0 & \dots & 1 & \dots & -1 & \dots & \\ \dots & 1 & \dots & 1 & \dots & -1 & \dots & -1 & \dots & \end{bmatrix}.$$

Poznamenejme, že v původní FETI dědily okrajové podoblasti okrajové podmínky od původní úlohy. Pro $d = 1$ bychom tak dostali bázi jádra (nulového prostoru) FETI matice A ve tvaru $\text{Ker } A = \text{Im } R$,

$$R = \text{diag}(o, e_2, \dots, e_{s-1}, o).$$

Pro komplikovanější úlohy je lepší vynutit předepsané hodnoty u_i tím, že přidáme do B příslušné řádky jednotkové matice tak, jak jsme to udělali výše. Tím jsme dosáhli toho, že všechny lokální matice tuhosti A_i mají předem známé jádro. Tato modifikace FETI se nazývá Total FETI či stručně TFETI [2]. Jestliže A je TFETI matice tuhosti, potom její jádro lze zapsat ve tvaru $\text{Ker } A = \text{Im } R$,

$$R = \text{diag}(e_1, \dots, e_s).$$

3. Redukce problému na hranice podoblastí

V dalším kroku vyloučíme primární neznámé u použitím Karush–Kuhn–Tuckerovy podmínky (metody Lagrangeových multiplikátorů) na řešení podmíněného extrému (16), čímž získáme

$$Au - b + B^T \lambda = o. \quad (17)$$

Rovnici (17) lze snadno odvodit nezávisle na obecné teorii. Pomocí identity (3) se můžeme přesvědčit, že řešení x úlohy (16) musí splňovat $\nabla f(x)^T d = 0$ jen pro přípustné přírůstky d , tedy pro $d \in \text{Ker } B$. Jelikož $Bd = o$ je ekvivalentní

$$(B^T z)^T d = z^T Bd = 0, \quad z \in \mathbb{R}^n,$$

platí, že $\text{Im } B^T$ je ortogonální doplněk $\text{Ker } B$. Odtud $\nabla f(x)^T d = 0$ pro všechna $d \in \text{Ker } B$ právě když $\nabla f(x) = Ax - b$ patří do $\text{Im } B^T$, tedy když existuje λ tak, že platí (17).

Jestliže $b - B^T \lambda \in \text{Im } A$, tedy jestliže existuje z tak, že $b - B^T \lambda = Az$, můžeme u vyjádřit ze (17) pomocí jakékoliv matice A^+ , která splňuje $AA^+A = A$. Snadno se ověří, že v tom případě pro každé u , které řeší (17), existuje vektor α tak, že

$$u = A^+(b - B^T \lambda) + R\alpha. \quad (18)$$

V našem případě můžeme ke specifikaci A^+ využít toho, že vynecháním posledního sloupce a posledního řádku A_i dostaneme regulární matici M_i , a zvolit

$$A_i^+ = \begin{bmatrix} M_i^{-1} & o^T \\ o & 0 \end{bmatrix}, \quad A^+ = \text{diag}(A_1^+, \dots, A_s^+).$$

Připomeňme, že R je báze jádra $\text{Ker } A$, které je ortogonálním doplňkem $\text{Im } A$, takže podmínku řešitelnosti $b - B^T \lambda \in \text{Im } A$ můžeme napsat ve tvaru

$$R^T(b - B^T \lambda) = o. \quad (19)$$

Po dosazení u z rovnice (18) do rovnice $Bu = o$ a elementárních úpravách dostaneme

$$BA^+B^T \lambda - BR\alpha = BA^+b. \quad (20)$$

Rovnice (19) a (20) určují λ i α . Než se dáme do jejich řešení, zavedeme označení

$$F = BA^+B^T, \quad G = -R^TB^T, \quad \tilde{d} = BA^+b, \quad e = -R^Tb,$$

takže rovnice (19) a (20) můžeme přepsat ve tvaru

$$\begin{aligned} F\lambda + G^T\alpha &= \tilde{d}, \\ G\lambda &= e. \end{aligned} \quad (21)$$

Pro iterační řešení (21) je vhodnější hledat λ ve vektorovém prostoru než v afinním prostoru. Tohoto cíle dosáhneme pomocí λ_0 , které splňuje $G\lambda_0 = e$, a substituce $\lambda = \lambda_0 + \mu$. Dostaneme

$$\begin{aligned} F\mu + G^T\alpha &= d, \quad d = \tilde{d} - F\lambda_0, \\ G\mu &= o. \end{aligned} \quad (22)$$

Poslední krok spočívá v „odfiltrování“ α z první rovnice soustavy (22). Povšimněme si, že (22) lze považovat za podmínky minima pro úlohu najít

$$\bar{\mu} = \arg \min_{\mu \in \text{Ker } G} \theta(\mu), \quad \theta(\mu) = \frac{1}{2}\mu^T F\mu - d^T \mu.$$

Jelikož $\text{Ker } G = \{P\mu : \mu \in \mathbb{R}^m\}$, $P = I - G^T(GG^T)^{-1}G$, můžeme $\bar{\mu}$ získat ve tvaru

$$\bar{\mu} = P \arg \min \theta(P\mu). \quad (23)$$

Z podmínky minima pro (23) tak dostaneme

$$PFP\mu = Pd. \quad (24)$$

K nalezení řešení (24) použijeme iterační řešič, typicky sdružené gradienty, který hledá řešení ve tvaru $\bar{\mu} = p(PFP)Pd$, kde p je polynom. V každé iteraci můžeme ušetřit jedno násobení projektorem P , neboť pro každý polynom p platí

$$p(PFP)Pd = p(PF)Pd.$$

K získání řešení stačí dosadit $\bar{\mu}$ do (22), vypočítat $\bar{\alpha} = (GG^T)^{-1}G(d - F\bar{\mu})$ a konečně

$$x = A^+(b - B^T(\lambda^0 + \bar{\mu})) + R\bar{\alpha}. \quad (25)$$

4. Co to přineslo?

Podívejme se na čísla podmíněnosti třídy matic

$$P^t F^t P^t, \quad t \in \mathcal{T}, \quad \mathcal{T} = \{(h, H) : H = 1/k, h = H/\ell, \quad k, \ell \in \mathbb{N}\}, \quad (26)$$

které vznikly rozložením oblasti Ω s parametry h a H , a na náročnost řešení jedné soustavy z třídy soustav rovnic závislejších na parametru $C > 0$,

$$P^t F^t P^t \mu^t = d^t, \quad t \in \mathcal{T}(C), \quad \mathcal{T}(C) = \{(h, H) \in \mathcal{T} : H/h \leq C\}. \quad (27)$$

Zvolme nejdříve pevně $t = (h, H)$ a zapišme $B = [B_1, \dots, B_s]$ tak, aby

$$F = BA^+B^T = [B_1, \dots, B_s] \begin{bmatrix} A_1^+ & O & \dots & O \\ O & A_2^+ & \dots & O \\ \vdots & \vdots & \dots & \vdots \\ O & O & \dots & A_s^+ \end{bmatrix} \begin{bmatrix} B_1^T \\ B_2^T \\ \vdots \\ B_s^T \end{bmatrix} = \sum_{i=1}^s B_i A_i^+ B_i^T.$$

Pokud c_B a C_B jsou takové konstanty, že pro všechna λ platí

$$c_B \|\lambda\|^2 \leq \|B^T \lambda\|^2 \leq C_B \|\lambda\|^2, \quad (28)$$

což budeme dále předpokládat, potom pro všechna $\lambda \in \text{Ker } G$ dostaneme

$$c_B \left(\max_{i=1, \dots, s} \|A_i\| \right)^{-1} \|\lambda\|^2 \leq \lambda^T F \lambda \leq C_B \left(\min_{i=1, \dots, s} \bar{\lambda}_{\min}(A_i) \right)^{-1} \|\lambda\|^2, \quad (29)$$

kde $\bar{\lambda}_{\min}(A_i)$ značí nejmenší nenulové vlastní číslo matice A_i . V našem případě jsou řádky matice B ortogonální, takže není těžké dosáhnout $c_B = C_B = 1$. V obecném případě můžeme provést velmi efektivně ortonormalizaci řádků matice B .

Vlastní čísla λ_{jk}^i diskrétního laplaciánu A_i jsou známa (viz [9]), např. pro $d = 2$ platí

$$\lambda_{jk}^i = \frac{4}{h^2} \left(\sin^2 \frac{j\pi}{2(H/h+1)} + \sin^2 \frac{k\pi}{2(H/h+1)} \right), \quad j, k = 0, \dots, H/h.$$

Odtud

$$\bar{\lambda}_{\min}(A_i) = \frac{4}{h^2} \sin^2 \frac{\pi}{2(H/h+1)} \cong \frac{\pi^2}{H^2}, \quad \lambda_{\max} = \frac{8}{h^2}, \quad \bar{\kappa}(A_i) \cong \frac{8H^2}{\pi^2 h^2} \lesssim \frac{H^2}{h^2}. \quad (30)$$

Symbol \cong zde znamená „s dostatečnou přesností“ a symbolu \lesssim rozumíme tak, že existuje konstanta $C > 1$ taková, že C -krát pravá strana je větší než levá strana pro všechna h, H . Poslední vztah na řádku (30) platí i pro $d = 1, 3$. Kombinací (29) a (30) dostaneme

$$\bar{\kappa}(PFP) = \lambda_{\max}(PFP) / \bar{\lambda}_{\min}(PFP) \lesssim H^2/h^2.$$

Je-li tedy $C > 0$ a $t \in \mathcal{T}(C)$, $t = (h, H)$, jsou poměr H/h a regulární číslo podmíněnosti $\bar{\kappa}(P^t F^t P^t)$ stejnoměrně omezené. Odtud plyne, že i počet iterací potřebných k získání přibližného řešení jakékoli soustavy (27) s předem danou relativní přesností

je stejnoměrně omezený. Odtud dále plyne, že složitost řešení je asymptoticky lineární, tedy $O(n)$. Takovou kvalitativní vlastnost má však i pouhé opisování řešení — *kvalitativně lepšího výsledku již nelze dosáhnout!* Nic podobného nelze říct o nerozložené soustavě $Ax = b$ získané pomocí konečných diferencí s krokem h , neboť z (30) plyne

$$1/h^2 \lesssim \bar{\kappa}(A).$$

Lineární složitost je velmi netriviální a nečekaný výsledek. Mimo jiné znamená, že i kdybychom měli k dispozici inverzní matici nebo trojúhelníkový rozklad A , nevyplatilo by se ho pro dostatečně velkou soustavu použít. První algoritmus s touto vlastností byl multigríd, navržený už v šedesátých letech minulého století Fedorenkem [7]. Multigríd je stále úspěšně používán pro řešení velkých soustav, jeho masivně paralelní implementace však je složitější ve srovnání s metodami rozložení oblasti. Myšlenka multigrídu je ve FETI zabudována prostřednictvím akce projektoru P na komplement prostoru tuhých pohybů „plovoucích“ podoblastí.

Většina nákladů iteračního kroku, násobení F a vektoru, spočívá v řešení s nezávislých soustav s SPS maticí, které lze provádět paralelně. Pokud bude poměr H/h omezený, bude při zjemňování diskretizace růst počet podoblastí a řešených problémů, které však budou mít stejnou dimenzi. Dokud bude možno přidělit každé podoblasti výpočetní jádro a nezačnou dominovat náklady na komunikaci, bude čas potřebný k řešení jakékoliv soustavy (27) téměř stejný pro všechna $t \in \mathcal{T}$, a tedy téměř nezávislý na dimenzi problému.

Nepříznivým důsledkem odhadu (30) je rychlý růst $\bar{\kappa}(PFP)$ se zjemňováním diskretizace podoblastí. Ukazuje se však, že odhad (30) je příliš pesimistický. K odvození realističtějšího odhadu si všimněme, že matice B má nenulové pouze sloupce odpovídající proměnným na hranici oblastí. Označme \tilde{B} matici, která vznikne z B vynecháním nulových sloupců, a necht \tilde{A} značí matici, která vznikne z A^+ vynecháním odpovídajících sloupců i řádků. Matice \tilde{A} zřejmě dědí blokovou strukturu z matice A^+ . Jelikož \tilde{A} je malý blok A^+ , lze očekávat, že $\bar{\kappa}(PFP)$ lze odhadnout mnohem lépe než ve (30). Podrobnější, ale netriviální analýza [12] sledující sestavení A ukazuje, že

$$\bar{\kappa}(PFP) = \bar{\kappa}(P\tilde{B}^T\tilde{A}\tilde{B}P) \lesssim H/h.$$

Dalšího zrychlení konvergence lze dosáhnout pomocí předpodmínění [12]. Necht \mathcal{I} a \mathcal{B} značí po řadě množinu indexů proměnných ve vnitřku a na hranici podoblastí a necht řádky a sloupce A jsou přeuspořádány tak, že platí

$$A = \begin{bmatrix} A_{\mathcal{I}\mathcal{I}} & A_{\mathcal{I}\mathcal{B}} \\ A_{\mathcal{B}\mathcal{I}} & A_{\mathcal{B}\mathcal{B}} \end{bmatrix}, \quad S = A_{\mathcal{B}\mathcal{B}} - A_{\mathcal{B}\mathcal{I}}A_{\mathcal{I}\mathcal{I}}^{-1}A_{\mathcal{I}\mathcal{B}}.$$

Potom pro vhodnou zobecněnou inverzi platí $\tilde{A} = S^+$ a pro matici předpodmíněného systému s předpodmiňovačem $P\tilde{B}^T\tilde{S}\tilde{B}P$ platí [12]

$$\bar{\kappa}(P\tilde{B}^T\tilde{S}\tilde{B}P) \lesssim 1 + \log^2 \frac{H}{h}.$$

5. Když FETI nestačí

Ukazuje se, že základní FETI metoda je efektivní do nejvýše desítek tisíc podoblastí. Kritickým činitelem je dimenze projektoru P , jehož akce s rostoucím počtem podoblastí dominuje výpočetní náročnosti. Aby překonali toto omezení, Farhat, Lesoinne, a Pierson [4] a Klawonn a Rheinbach [8] navrhli spojit skupiny podoblastí explicitně.

Například matici tuhosti klastru tvořeného čtveřicí podoblastí se společným bodem

$$x \in \bar{\Omega}_i \cap \bar{\Omega}_j \cap \bar{\Omega}_k \cap \bar{\Omega}_l,$$

lze získat částečným rozřezáním větších podoblastí nebo pomocí matice, která vznikne z jednotkové matice nahrazením čtyř sloupců, odpovídajících čtyřem rohovým uzlům totožným s x , jejich součtem. Když provedeme stejnou modifikaci se sloupci jednotkové matice řádu n pro všechny klastry, dostaneme matici L , která transformuje globální proměnné \tilde{x} na původní proměnné x prostřednictvím vztahu $x = L\tilde{x}$. Snadno se ověří, že blokově diagonální matice tuhosti $K = \text{diag}(K_1, \dots, K_{n_k})$ s bloky K_i odpovídajícími klastrům a matice C rovnostních vazeb, které spojují nové proměnné, jsou dány formulemi

$$K = L^T A L, \quad C = B L.$$

Jádro každého klastru K_i získaného slepením čtyř podoblastí je pouze jednodimenzionální, takže počet řádků matice G odpovídající rozložení na klastry je jen jedna čtvrtina počtu řádků původní matice G . Klaster, který vznikne slepením čtyř sousedních podoblastí v rozích [13], je na obrázku 3.

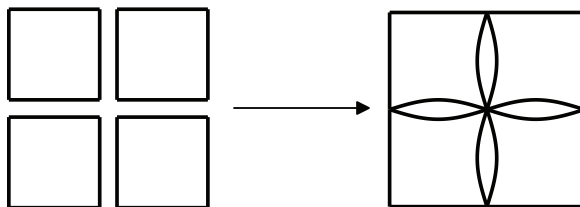
Když zaměníme podoblasti za klastry, dostaneme duální matici $F = C^T K + C$ téměř stejným postupem jako matici $F = B A^+ B^T$ v kap. 3. Jak se dá očekávat, regulární číslo podmíněnosti Schurova komplementu matice tuhosti klastru vznikajícího spojením v jednom či více rozích je větší než regulární číslo podoblasti; platí

$$\bar{\kappa}(C^T K + C) \lesssim \left(1 + \log \frac{H}{h}\right) \frac{H}{h}.$$

Místo spojení v rozích můžeme rovnostní vazby upravit a rozdělit do dvou bloků, B_P a B_D ,

$$B = \begin{bmatrix} B_P \\ B_D \end{bmatrix},$$

a použít B_P k eliminaci některých primárních proměnných. Matice B_P nemusí definovat jen „slepení“ několika rohů, ale může definovat i obecnější vazby jako nulový průměr součtu proměnných na sousedních hranách. Detaily implementace nejsou triviální,

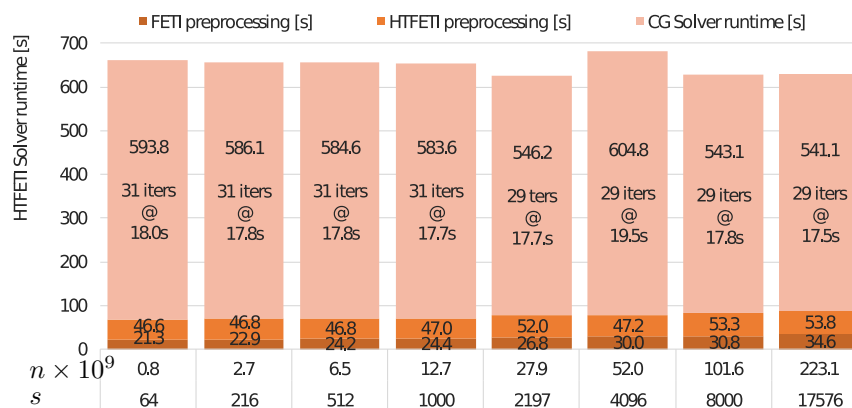


Obr. 3. H-TFETI klaster se spojením v rozích

Weak Scalability Test

Up to **223 billion** DOF on 17576 compute nodes (281 216 cores)

Heat transfer (Laplace equation)



Obr. 4. Slabá škálovatelnost. Ve sloupcích shora: čas iteračního řešiče, počet iterací, čas iterace, preprocessing. Poslední dva řádky obsahují počet neznámých n a počet podoblastí s .

ale dobře známé, viz např. Klawonn a Rheinbach [8] nebo Dostál et al. [3, Chap. 19]. Lze dokázat, že regulární čísla podmíněnosti matic sestavených z podoblastí i malých klastřů slepených nulovými hranovými průměry lze odhadnout kvalitativně stejně (viz Vodstrčil et al. [13]), tedy

$$\bar{\kappa}(C^T K^+ C) \lesssim H/h.$$

Dalšího zrychlení konvergence lze dosáhnout předpokládáním podobně jako u FETI.

6. Jak to skutečně funguje?

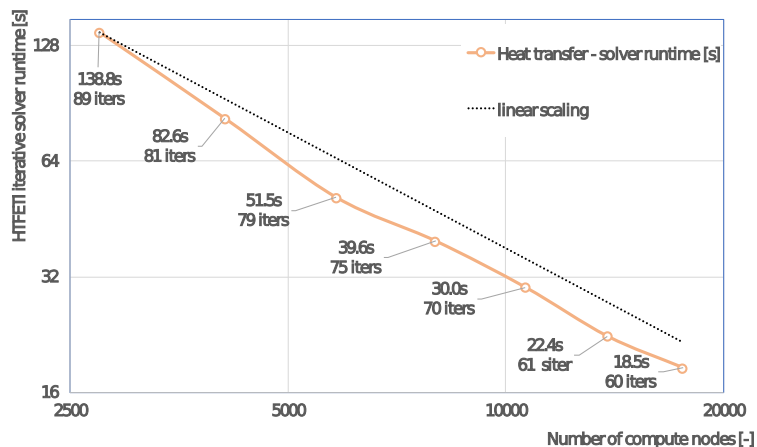
Zbývá ukázat, jak to skutečně funguje. K testování použijeme 3D verzi úlohy (15) s parametry H, h , $H/h = 250$. Nejprve budeme demonstrovat *slabou škálovatelnost*, tedy že lze skutečně řešit různé velké úlohy do dané relativní přesnosti (zde $10^{-6} \|b\|$) se stejnoměrně omezeným počtem iterací, které lze implementovat paralelně. Na obrázku 4 jsou výsledky dosažené HTFETI s předpokládáním se softwarem ESPRESO [10] na superpočítači TITAN v Oak Ridge National Laboratory (www.ornl.gov).

Vidíme, že v souladu s teorií se počty iterací prakticky nemění (dokonce mírně klesají) při řešení úloh diskretizovaných v rozsahu $8 \times 10^8 - 2.23 \times 10^{11}$ neznámých. Jelikož podoblasti jsou diskretizovány stejným počtem neznámých a většina manipulace s nimi se provádí současně, i časy zůstávají prakticky stejné (asi 10 minut).

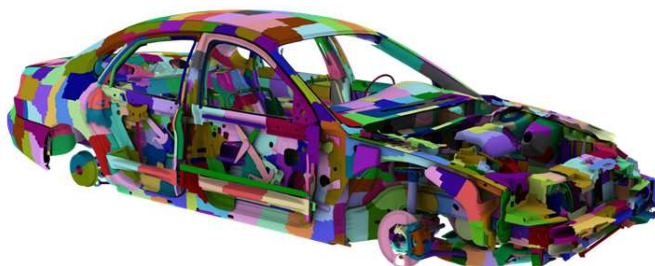
Na obrázku 5 jsou výsledky řešení stejné úlohy, tentokrát s pevně zvolenou diskretizací oblasti a s měnící se dekompozicí a s měnícím se počtem procesorů. Opět

Strong Scalability Test

Heat transfer 20 billion DOF on up to 17 576 Compute Nodes (281 216 cores)



Obr. 5. Silná škálovatelnost, $n = 20 \times 10^9$



Obr. 6. Karoserie automobilu rozložená na podoblasti (autor T. Brzobohatý)

vidíme, že algoritmus prošel úspěšně i tímto testem *silné škálovatelnosti*, neboť čas řešení klesá úměrně počtu procesorů. Počet iterací klesá v souladu s teorií.

Na závěr poznamenejme, že pro řešení praktických úloh se rozložení na podoblasti provádí automaticky. Ukázka takového rozkladu pro řešení realistické úlohy je na obrázku 6. K diskretizaci velmi rozsáhlých úloh je obtížné využít masivní paralelismus, takže svou náročností soupeří s řešením úlohy.

7. Závěr

Cílem článku bylo naznačit úlohu matematiky při řešení rozsáhlých soustav lineárních rovnic na současných masivně paralelních superpočítačích. I když jsme se zde omežili jen na eliptické PDR, podobné výsledky jsou známy i pro další typy úloh. Řešení

úloh v Lagrangeových multiplifikátorech je mimořádně výhodné pro řešení variačních nerovnic, neboť veškerá omezení ve formě nerovností se redukuje na prosté meze. Variační nerovnice, které popisují rovnováhu soustavy těles ve vzájemném kontaktu, je tak možné řešit s optimální složitostí stejně jako rovnice (např. [3]). S dalšími typy metod rozložení oblasti se lze seznámit v monografii Toselliho a Widlunda [12]. Představu o současném výzkumu lze získat ze stránek sborníků z DDM konferencí [1].

Poděkování. Autor děkuje J. Bouchalovi a své ženě Marii za podnětné připomínky a T. Brzobohatému a dalším autorům software ESPRESO [10] za numerické experimenty. Práce vznikla za podpory MŠMT ČR: The project „IT4Innovations excellence in science — LQ1602“ supported by National Programme of Sustainability (NPU II) and the Large Infrastructures for Research, Experimental Development and Innovations project „IT4Innovations National Supercomputing Center – LM2015070“.

L i t e r a t u r a

- [1] Domain Decomposition Proceedings [online]. Dostupné z: <http://www.ddm.org/conferences.html>
- [2] DOSTÁL, Z., HORÁK, D., KUČERA, R.: *Total FETI — an easier implementable variant of the FETI method for numerical solution of elliptic PDE*. Commun. Numer. Methods Eng. 22 (2006), 1155–1162.
- [3] DOSTÁL, Z., KOZUBEK, T., SADOWSKÁ, M., VONDRÁK, V.: *Scalable algorithms for contact problems*. AMM 36, Springer, New York, 2016.
- [4] FARHAT, C., LESOINNE, M., PIERSON K.: *A scalable dual-primal domain decomposition method*. Numer. Linear Algebra Appl. 7 (2000), 687–714.
- [5] FARHAT, C., MANDEL, J., ROUX, F.-X.: *Optimal convergence properties of the FETI domain decomposition method*. Comput. Methods Appl. Mech. Engrg. 115 (1994), 365–385.
- [6] FARHAT, C., ROUX, F.-X.: *A method of finite element tearing and interconnecting and its parallel solution algorithm*. Int. J. Numer. Methods Eng. 32 (1991), 1205–1227.
- [7] FEDORENKO, R. P. *The speed of convergence of one iterative process*. Soviet Comput. Math. Math. Phys. 4 (1964), 227–235.
- [8] KLAWONN, A., RHEINBACH, O.: *Highly scalable parallel domain decomposition methods with an application to biomechanics*. Z. Angew. Math. Mech. 90 (2010), 5–32.
- [9] MARČUK, G. I.: *Metody numerické matematiky*. Academia, Praha, 1987.
- [10] ŘÍHA, L., BRZOBOHATÝ, T., MARKOPOULOS, A., MECA, O.: *IT4I Espresso – fast solver for HPC users* [online]. Dostupné z: espresso.it4i.cz
- [11] SCHWARZ, H. A.: *Über einen Grenzübergang durch alternierendes Verfahren*. Vierteljahrsschrift der Naturforschenden Gesellschaft in Zürich 15 (1870), 272–286.
- [12] TOSELLI, A., WIDLUND O. B.: *Domain Decomposition Methods – Algorithms and Theory*. CM 34, Springer, Berlin, 2005.
- [13] VODSTRČIL, P., BOUCHALA, J., JAROŠOVÁ, M., DOSTÁL, Z.: *On conditioning of Schur complements of H-TFETI clusters for 2D problems governed by Laplacian*. Appl. Math. 62 (2017), 699–718.