VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra aplikované matematiky

Paralelizace řešení eliptických okrajových úloh pomocí TFETI metody rozložení oblastí

Parallelization of the solution of elliptic boundary value problems using TFETI domain decomposition method

2012 Pavla Jirůtková

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra aplikované matematiky

Zadání bakalářské práce

Student: **Pavla Jirůtková**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 1103R031 Výpočetní matematika

Téma: Paralelizace řešení eliptických okrajových úloh pomocí TFETI metody rozložení oblastí
Parallelization of the solution of elliptic boundary value problems using TFETI domain decomposition method

Zásady pro vypracování:

Cílem práce je seznámit se s metodou Total FETI (Total Finite Element Tearing and Interconnecting), provést její implementaci v Matlabu pro 2D Poissonovu úlohu a výslednou aplikaci paralelizovat pomocí Matlab Distributed Computing serveru. Metoda TFETI umožňuje rozdělit původní (počtem neznámých) rozsáhlou úlohu na úlohy menší, které mohou být řešeny současně na více procesorech. Zajištění konektivity řešení jednotlivých úloh se provádí pomocí Lagrangeových multiplikátorů, které představují síly lepící řešení dílčích problémů dohromady.

Seznam doporučené odborné literatury:

Publikace autorů Z. Dostál, T. Kozubek, V. Vondrák, Ch. Farhat, O. Widlund, A. Klawon.
http://am.vsb.cz/kozubek

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **doc. Ing. Tomáš Kozubek, Ph.D.**

Datum zadání: 18.11.2011
Datum odevzdání: 04.05.2012

VYSOKÁ ŠKOLA BÁŇSKÁ-TECHNICKÁ UNIVERZITA * Fakulta elektrotechniky a informatiky * OSTRAVA

doc. RNDr. Jiří Bouchala, Ph.D.
vedoucí katedry

prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlašuji, že jsem tuto bakalářskou práci vypracovala samostatně. Uvedla jsem všechny literární prameny a publikace, ze kterých jsem čerpala.

V Ostravě 4. května 2012 .....

Ráda bych na tomto místě poděkovala Doc. Ing. Tomášovi Kozubkovi, Ph.D. za mnohé rady, připomínky, ochotu a trpělivost při vedení mé bakalářské práce.

Abstrakt

Práce se zabývá paralelní implementací TFETI metody rozložení oblasti, která je vyvíjena na Katedře aplikované matematiky VŠB - TU Ostrava pro řešení rozsáhlých inženýrských problémů, a dále její aplikací na řešení eliptických okrajových úloh. Po diskretizaci oblasti pomocí metody konečných prvků se oblast rozdělí na nepřekrývající se podoblasti tvořené elementy sítě. Toto vede na soustavu rovnic s blokovou maticí tuhosti, což usnadňuje paralelizaci jejího řešení. Pomocí Lagrangeových multiplikátorů jsou pak vynucovány nejen lepící podmínky, ale i Dirichletovy okrajové podmínky. Metoda je tak ve srovnání s metodou FETI efektivnější a snažší na implementaci. Je zde také provedena efektivní regularizace matic tuhosti podoblastí a jsou zavedeny ortogonální projektory, které zlepšují podmíněnost duální úlohy s Lagrangeovými multiplikátory. Na numerických experimentech je ověřena numerická i paralelní škálovatelnost metody.

Klíčová slova: Lagrangeovy multiplikátory, metoda konečných prvků, metoda rozložení oblasti, paralelní implementace, PCGP, škálovatelnost, TFETI

Abstract

The bachelor thesis deals with parallel implementation of the TFETI domain decomposition method, which is developed at the Department of Applied Mathematics VŠB - TU Ostrava to solve large engineering problems. It deals also with application of TFETI on the solution of elliptic boundary value problems. After finite element space discretization we decompose the domain into nonoverlapping subdomains defined by mesh elements. This leads to a system of linear equations with block-diagonal system matrix which enables its effective parallel solution. The gluing conditions and even the Dirichlet boundary conditions are enforced by the Lagrange multipliers, so the method is more effective and simpler for implementation than FETI. The effective regularization of the stiffness matrices of the subdomains is performed and the condition number of the dual problem with Lagrange multipliers is improved by using orthogonal projectors. Both numerical and parallel scalability of TFETI are tested in the numerical experiments.

Keywords: Lagrange multipliers, finite element method, domain decomposition method, parallel implementation, PCGP, scalability, TFETI

Seznam použitých zkratek a symbolů

MKP	–	Metoda konečných prvků
PCGP	–	Projected Conjugate Gradient method with Preconditioning
QP	–	Quadratic Programming
TFETI	–	Total Finite Element Tearing and Interconnecting
$\partial\Omega$	–	Hranice oblasti Ω
$\overline{\Omega}$	–	Uzávěr oblasti Ω
$C(\Omega)$	–	Funkce spojitá na Ω
$C^1(\Omega)$	–	Funkce se spojitými prvními derivacemi na Ω
$C^2(\Omega)$	–	Funkce se spojitými druhými derivacemi na Ω
Δ	–	Laplaceův operátor
$\mathbf{I}$	–	Jednotková matice
$\mathbf{O}$	–	Nulová matice
$\mathbb{R}$	–	Množina všech reálných čísel
S	–	Schurův doplněk
λ	–	Lagrangeovy multiplikátory
∇	–	Gradient
$\mathbf{o}$	–	Nulový vektor

Obsah

Seznam tabulek

Seznam obrázků

1 Úvod

S rostoucím výkonem počítačů se nabízí možnost řešit i náročné inženýrské úlohy. K tomu je ovšem zapotřebí mít k dispozici efektivní numerické metody, které lze uplatnit i na paralelní počítače a získat tak řešení v mnohem kratším čase. Mezi takové metody patří i metody rozložení oblasti, které zadanou úlohu dělí na menší části a ty je pak možné řešit paralelně. Cílem bakalářské práce je popsat efektivní paralelní implementaci právě jedné z těchto metod, konkrétně metody TFETI (Total Finite Element Tearing and Interconnecting), a použít ji na řešení okrajové eliptické úlohy.

Nejprve je nutné provést diskretizaci zadané oblasti na určitý (konečný) počet prvků. Tu provádíme pomocí metody konečných prvků (MKP), která je v porovnání s metodou konečných diferencí vhodnější na náročné inženýrské úlohy, protože vychází z fyzikálního modelu a lze ji aplikovat i na složité geometrické tvary. Stručně se touto metodou zabýváme v kapitole 2, kde si nejprve v 1D, poté ve 2D, ukážeme podstatné vztahy, které jsou zapotřebí při samotné implementaci.

Hlavní částí je kapitola 3, kde se zabýváme samotnou metodou TFETI poprvé představenou Dostálem, Horákem a Kučerou v [1], a která pomocí Lagrangeových multiplikátorů vynucuje splnění lepících podmínek, ale také splnění Dirichletových okrajových podmínek. Krátce si představíme některé významné metody rozložení oblasti a hlavní rozdíly mezi nimi. Ukážeme si přechod původního QP (Quadratic Programming) problému na duální problém, efektivní regularizaci matice tuhosti, která vychází z [2, 3], a také aplikaci upravené metody sdružených gradientů s předpodmíněním. Vlastní paralelizace metody je pak naznačena v následující kapitole 4, kde je popsán způsob paralelní implementace v MATALBu.

Na závěr (kapitola 5) jsou předvedeny výsledky numerických experimentů, které byly provedeny na výpočetním clusteru ComSiO superpočítačového centra VŠB-TU Ostrava (SPC) [4]. Na vybraných příkladech je testována, a následně i ověřena, numerická a paralelní škálovatelnost algoritmu.

2 Stručně o metodě konečných prvků

Nejznámějšími numerickými metodami pro řešení eliptických parciálních diferenciálních rovnic jsou metoda konečných diferencí (metoda sítí) a metoda konečných prvků. Obě tyto metody převádí původní úlohu na soustavu algebraických lineárních rovnic, která má výrazně pásový charakter. Abychom mohli numericky vyřešit eliptickou okrajovou úlohu, musíme provést diskretizaci zadané oblasti. Tu provedeme právě pomocí MKP, která původní oblast rozděluje na tzv. konečné prvky a narozdíl od metody sítí diskretizuje tzv. slabou formulaci úlohy. V celé této kapitole vycházíme zejména z [5] .

2.1 MKP v 1D

Mějme strunu uchycenou na obou koncích, pro kterou platí

$$\begin{aligned} &-u^{''}(x) = f(x) \quad \text{pro } \forall x \in (a;b), \\ &u(a) = u(b) = 0, \\ &f \in C((a;b)),\ u \in C^2((a;b)). \end{aligned} \tag{2.1}$$

Definujeme si množinu testovacích funkcí V:

$$V = \{v \in C^1((a;b)) \,|\, v(a) = v(b) = 0\}.$$

Diferenciální rovnici (2.1) vynásobíme libovolnou testovací funkcí z množiny V, výraz zintegrujeme na intervalu $\langle a;b\rangle$ pomocí metody per partes a upravíme. Dostáváme tak slabou formulaci ve tvaru

$$\begin{cases} \text{nadji } u \in V \ \text{ takové, že} \\ a(u,v) = b(v) \quad \forall v \ \in V\ , \end{cases} \tag{2.2}$$

kde

$$a(u,v) = \int_a^b u^{'}(x)\, v^{'}(x)\, dx, \qquad b(v) = \int_a^b f(x)\, v(x)\, dx.$$

Přičemž $a(u,v)$ je bilineární forma a $b(v)$ je lineární funkcionál.

Poznámka 2.1 Slabá formulace je ekvivalentní s tzv. energetickou (variační) formulací, která hledá řešení $u(x)$ jako minimum funkce, tedy

$$u(x) = \arg\min_{u \in V} \frac{1}{2} a(u,u) - b(u).$$

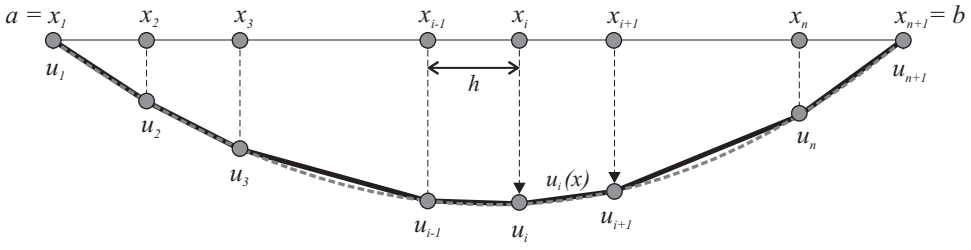
Diskretizace

Strunu zachycenou na obou koncích rozdělíme na n dílků se stejným krokem h. Dostáváme tak ekvidistantní síť na intervalu $\langle a; b\rangle$ s krokem $h = \frac{b-a}{n}$ a množinou uzlů $S_h = \{x_1, x_2, \ldots, x_{n+1}\}$, kde $x_{i+1} = x_i + h$, $i = 1, \ldots, n$. Síť je tedy složena z n konečných prvků, kde i-tý prvek je roven intervalu $\langle x_i, x_{i+1}\rangle$. Úlohu dále řešíme jako posunutí v jednotlivých uzlech a dostáváme tak aproximovaný průhyb struny $u(x)$ pomocí spojité po částech lineární funkce

$$u_i(x) = a_i x + b_i, \ \ \forall x \in \langle x_i, x_{i+1}\rangle, \ \ i = 1, \ldots, n. \tag{2.3}$$

Dosadíme-li souřadnice uzlů x_i a x_{i+1} do rovnice (2.3) získáme pro i-tý prvek posunutí v těchto uzlech. Označme si posunutí uzlu x_i jako u_i a posunutí uzlu x_{i+1} jako u_{i+1}. Dostáváme dvě lineární rovnice o dvou neznámých, ze kterých získáme konstanty a a b:

$$a_i = \frac{1}{h}\left(u_{i+1} - u_i\right), \, b_i = -\frac{1}{h}\left(x_i u_{i+1} - x_{i+1} u_i\right). \tag{2.4}$$



Obrázek 2.1: Diskretizace struny.

Matice tuhosti

Provedeme diskretizaci bilineární formy $a(u, u)$. To znamená, že integrál nad celým intervalem $\langle a; b\rangle$ rozdělíme na n integrálů, každý nad i-tým prvkem a sečteme je:

$$a(u, u) = \sum_{i=1}^{n} \int_{x_i}^{x_{i+1}} \left(u'(x)\right)^2 dx. \tag{2.5}$$

Člen $(u'(x))^2$ si vyjádříme jako $(u'(x))^T (u'(x))$, provedeme derivaci funkce posuvů (2.3) a celý výraz zintegrujeme. Po úpravě získáme

$$a\left(u_i(x), u_i(x)\right) = (u_i, u_{i+1}) \frac{1}{h} \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} \begin{pmatrix} u_i \\ u_{i+1} \end{pmatrix}, \tag{2.6}$$

kde

$$\mathbf{A_i} = \frac{1}{h}\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}. \tag{2.7}$$

Takto definovaná matice $\mathbf{A_i}$ je maticí tuhosti pro i-tý prvek sítě a označujeme ji pojmem lokální matice tuhosti.

Globální matici tuhosti pro celou síť pak postupně seskládáme z jednotlivých příspěvků $\mathbf{A_i}$, kde $i = 1, \ldots, n$. Pro strunu s krokem sítě h tak získáme globální matici tuhosti, která je pozitivně semidefinitní a má třídiagonální tvar:

$$\mathbf{A} = \sum_{i=1}^{n} \mathbf{A_i} = \frac{1}{h}\begin{pmatrix} 1 & -1 & 0 & \ldots & 0 & 0 \\ -1 & 2 & -1 & \ldots & 0 & 0 \\ 0 & -1 & 2 & \ddots & \vdots & \vdots \\ \vdots & \vdots & \ddots & \ddots & -1 & 0 \\ 0 & 0 & \ldots & -1 & 2 & -1 \\ 0 & 0 & \ldots & 0 & -1 & 1 \end{pmatrix}. \tag{2.8}$$

Vektor zatížení

Tak, jak jsme u matice tuhosti vycházeli z formy $a(u, u)$, bude vektor zatížení vycházet z lineárního funkcionálu $b(u)$:

$$b(u) = \sum_{i=1}^{n} \int_{x_i}^{x_{i+1}} u(x) f(x)\, dx. \tag{2.9}$$

Opět se zaměříme na i-tý prvek a provedeme jeho posunutí v bodě x (vycházíme ze vztahu (2.3) a (2.4)):

$$\begin{aligned} u_i(x) &= \frac{1}{h}\left((x_{i+1} - x)\, u_i + (x - x_i)\, u_{i+1}\right) \\ &= \frac{1}{h}\left(x_{i+1} - x, x - x_i\right)\begin{pmatrix} u_i \\ u_{i+1} \end{pmatrix}. \end{aligned}$$

Toto posunutí přenásobíme s $f(x)$ a celý výraz zintegrujeme nad i-tým prvkem. Dostáváme

$$\int_{x_i}^{x_{i+1}} u_i(x)^\top f(x) dx = \frac{1}{h}\left(u_i, u_{i+1}\right)\begin{pmatrix} \int_{x_i}^{x_{i+1}} f(x)(x_{i+1} - x)\, dx \\ \int_{x_i}^{x_{i+1}} f(x)(x - x_i)\, dx \end{pmatrix}, \tag{2.10}$$

kde

$$\mathbf{f_i} = \frac{1}{h}\begin{pmatrix} \int_{x_i}^{x_{i+1}} f(x)(x_{i+1} - x)\, dx \\ \int_{x_i}^{x_{i+1}} f(x)(x - x_i)\, dx \end{pmatrix}. \tag{2.11}$$

Vektor $\mathbf{f_i}$ je vektor zatížení i-tého prvku sítě a označujeme jej pojmem lokální vektor zatížení.

V našem případě nám však bude stačit pouze přibližný výpočet integrálu. Pomocí obdélníkového pravidla dostáváme aproximaci $\mathbf{f_i}$:

$$\mathbf{f_i} \approx \begin{pmatrix} f(x_s)(x_{i+1} - x_s) \\ f(x_s)(x_s - x_i) \end{pmatrix}, \tag{2.12}$$

kde $x_s = \frac{1}{2}(x_i + x_{i+1})$ je souřadnice těžiště i-tého prvku.

Obdobně jako u matice tuhosti sestavíme globální vektor zatížení z jednotlivých příspěvků $\mathbf{f_i}$, kde $i = 1, \ldots, n$, které usazujeme na pozice i a $i+1$.

$$\mathbf{f} = \sum_{i=1}^{n} \mathbf{f_i} = \frac{1}{h} \begin{pmatrix} \int_{x_1}^{x_2} f(x)(x_1 - x)\,dx \\ \vdots \\ \int_{x_{i-1}}^{x_i} f(x)(x - x_{i-1})\,dx + \int_{x_i}^{x_{i+1}} f(x)\ (x_{i+1} - x)\,dx \\ \vdots \\ \int_{x_n}^{x_{n+1}} f(x)(x - x_n)\,dx \end{pmatrix}. \tag{2.13}$$

2.2 MKP ve 2D

Mějme parciální diferenciální rovnici ohraničené membrány, pro kterou platí

$$\begin{aligned} &-\Delta u(x,y) = f(x,y) \quad \text{pro } \forall (x,y) \in \Omega, \\ &u(x,y) = 0 \quad \text{pro } \forall (x,y) \in \partial\Omega, \\ &f \in C(\Omega),\ u \in C^2(\Omega). \end{aligned} \tag{2.14}$$

Definujeme si množinu testovacích funkcí

$$V = \{v \in C^1(\Omega) \,|\, v = 0 \text{ na } \partial\Omega\}.$$

Parciální diferenciální rovnici (2.14) přenásobíme testovací funkcí $v \in V$, levou i pravou stranu výrazu zintegrujeme na oblasti Ω a pomocí Greenovy věty rovnici upravíme. Obdržíme slabou formulaci ve tvaru:

$$a(u,v) = b(v) \qquad \forall v\ \in V\ , \tag{2.15}$$

kde

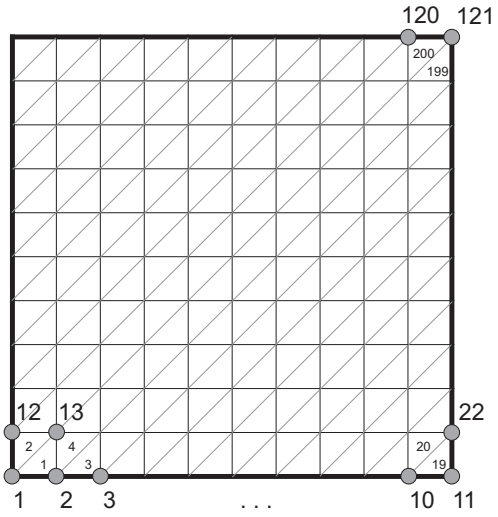
$$a(u,v) = \int_\Omega \left(\frac{\partial u \partial v}{\partial x^2} + \frac{\partial u \partial v}{\partial y^2} \right) \mathrm{d}\Omega, \qquad b(v) = \int_\Omega fv\ \mathrm{d}\Omega + \int_{\partial\Omega} \left(\frac{\partial u}{\partial x} n_x v + \frac{\partial u}{\partial y} n_y v \right) \mathrm{d}s.$$

Triangulace a interpolace řešení

Oblast $\Omega \subset \mathbb{R}^2$ rozdělíme na čtvercovou síť s krokem h. Dostáváme tak množinu uzlů $S_h = \{P_1, P_2, \ldots, P_n\}$. Nyní provedeme triangulaci, tedy oblast $T_h = \bar{\Omega}$ rozdělíme na konečný počet trojúhelníků t_k (viz obr. 2.2), pro které platí:

$$T_h = \bigcup_{k=1}^{n_T} t_k,\ t_k = \Delta_{P_{k1},P_{k2},P_{k3}},\ \ n_T = 2\,(n-1)^2\,, \tag{2.16}$$

kde n_T je počet všech prvků sítě. Průhyb k-tého uzlu označíme jako $u_k \approx u_{P_k}$, $k = 1, \ldots, n$.



Obrázek 2.2: Triangularizace.

Úlohu budeme dále řešit jako posunutí v jednotlivých uzlech. Funkce posuvů na k-tém prvku aproximujeme pomocí lineární funkce

$$\begin{aligned} &u_k\,(x,y) = a_k x + b_k y + c_k,\ \ \forall\,(x,y) \in t_k, \\ &u_k\,(x,y) = 0,\ \ \forall\,(x,y) \notin t_k. \end{aligned} \tag{2.17}$$

Rovnici (2.17) si přepíšeme do tvaru

$$u_k\,(x,y) = \varphi\,(x,y)\,\mathbf{a}_k = [x,y,1]\,[a_k,b_k,c_k]^\top\,, \tag{2.18}$$

kde φ je matice aproximačního polynomu a $\mathbf{a}_k$ je vektor konstant.

Dosazením souřadnic uzlů, které jsou zároveň vrcholy k-tého prvku, do rovnice (2.17) získáme soustavu tří rovnic, kterou lze zapsat v maticovém tvaru jako

$$\mathbf{D}_k \mathbf{a}_k = \mathbf{u}_k, \tag{2.19}$$

tedy

$$\begin{pmatrix} x_{k_1} & y_{k_1} & 1 \\ x_{k_2} & y_{k_2} & 1 \\ x_{k_3} & y_{k_3} & 1 \end{pmatrix} \begin{pmatrix} a_k \\ b_k \\ c_k \end{pmatrix} = \begin{pmatrix} u_{k_1} \\ u_{k_2} \\ u_{k_3} \end{pmatrix}.$$

Matice D_k je regulární a proto platí, že $\mathbf{a}_k = \mathbf{D}_k^{-1}\mathbf{u}_k$. Dosazením takto vyjádřeného a_k do rovnice (2.18) získáme předpis

$$u_k(x,y) = \varphi(x,y)\,\mathbf{D}_k^{-1}\mathbf{u}_k = \mathbf{N}_k\mathbf{u}_k, \tag{2.20}$$

kde $\mathbf{N}_k = \varphi(x,y)\,\mathbf{D}_k^{-1}$ je tzv. matice bázových funkcí.

Matice tuhosti

Vycházíme z levé strany rovnice slabé formulace (2.15):

$$a(u_k,u_k) = \int_\Omega \nabla u_k \left(\nabla u_k\right)^\top \,\mathrm{d}\Omega, \tag{2.21}$$

kde gradient vektoru posuvů $\left(\nabla u_k\right)^T = \nabla\varphi(x,y)\,\mathbf{D}_k^{-1}\mathbf{u}_k$. Označme $\mathbf{G}_k := \nabla\left(\varphi(x,y)\,\mathbf{D}_k^{-1}\right)$ jako transformační matici. Dosazením $\left(\nabla u_k\right)^T = \mathbf{G}_k\mathbf{u}_k$ do rovnice (2.21) obdržíme

$$\mathbf{u_k}^\top \int_{t_k} \mathbf{G}_k^\top\mathbf{G}_k \,\mathrm{d}\Omega\; \mathbf{u}_k = \mathbf{u}_k^\top\mathbf{A}_k\mathbf{u}_k, \tag{2.22}$$

kde

$$\mathbf{A}_k = \mathbf{G}_k^\top\mathbf{G}_k S_\Delta \tag{2.23}$$

je lokální matice tuhosti, přičemž $S_\Delta = \frac{1}{2}\left|\det\left(\mathbf{D}_k\right)\right|$ je plocha konečného prvku, tj. trojúhelníku.

Globálmí matici tuhosti $\mathbf{A}$ sestavíme z jednotlivých příspěvků $\mathbf{A}_k$. Tento příspěvek je definován jako $a_{k_i,k_j} = a_{k_i,k_j} + a_{i,j}^k$ pro $i,j = 1,2,3$. Označením k_1,k_2,k_3 máme na mysli globální čísla uzlů k-tého prvku.

$$\mathbf{A} = \sum_{k=1}^{n_T}\mathbf{A}_k = \sum_{k=1}^{n_T}\begin{pmatrix} 0 & \dots & 0 & \dots & 0 & \dots & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & & \vdots & & \vdots & & \vdots \\ 0 & \dots & a_{1,1}^k & \dots & a_{1,2}^k & \dots & a_{1,3}^k & \dots & 0 \\ \vdots & & \vdots & \ddots & \vdots & & \vdots & & \vdots \\ 0 & \dots & a_{2,1}^k & \dots & a_{2,2}^k & \dots & a_{2,3}^k & \dots & 0 \\ \vdots & & \vdots & & \vdots & \ddots & \vdots & & \vdots \\ 0 & \dots & a_{3,1}^k & \dots & a_{3,2}^k & \dots & a_{3,3}^k & \dots & 0 \\ \vdots & & \vdots & & \vdots & & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & \dots & 0 & \dots & 0 & \dots & 0 \end{pmatrix}. \tag{2.24}$$

Vektor zatížení

U vektoru zatížení vycházíme z pravé strany rovnice slabé formulace (2.15), která je rozdělena na dva příspěvky, a to:

- hustotu sil

$$\int_{t_k} u^\top f \, \mathrm{d}\Omega = \mathbf{u}_k^\top \mathbf{b}_k \ , \tag{2.25}$$

- zatížení na hranici

$$\oint_{\Gamma_T^k} u^\top T \, \mathrm{d}S = \mathbf{u}_k^\top \Big|_{\Gamma_T^k} \mathbf{T}_k \quad \text{, kde } \Gamma_T^k := \overline{P_{k_I} P_{k_{II}}},\ I, II \in \{1,2,3\}\,. \tag{2.26}$$

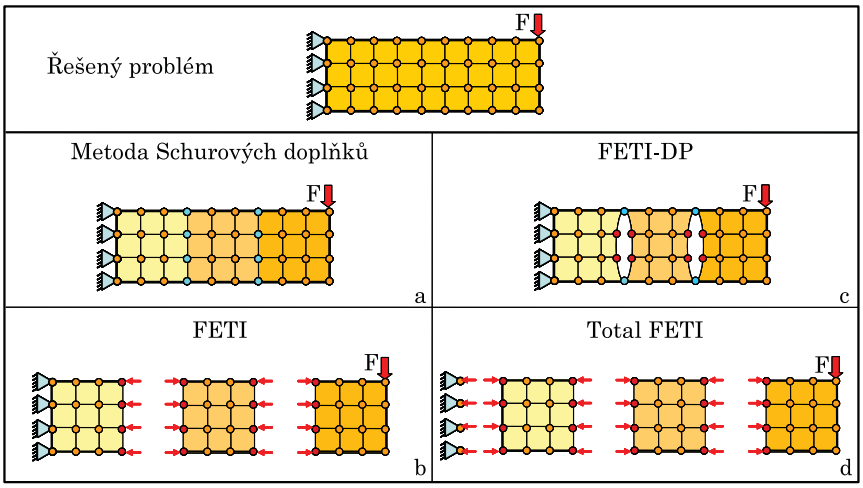
Jestliže provedeme sumarizaci příspěvků (2.25) a (2.26) přes všechny příslušné prvky a pak tyto příspěvky sečteme, dostaneme globální vektor zatížení.

$$\begin{aligned} &\mathbf{b}_k = \left(b_i^k\right)_{i=1,2,3},\ \mathbf{T}_k = \left(T_i^k\right)_{i=1,2}, \\ &\mathbf{b} = \sum_{k=1}^{n_T} \mathbf{b}_k + \sum_{k=1}^{n_{\Gamma_T}} \mathbf{T}_k = \sum_{k=1}^{n_T} \begin{pmatrix} b_{k_1} + b_1^k \\ b_{k_2} + b_2^k \\ b_{k_3} + b_3^k \end{pmatrix} + \sum_{k=1}^{n_{\Gamma_T}} \begin{pmatrix} b_{k_I} + T_1^k \\ b_{k_{II}} + T_2^k \end{pmatrix}. \end{aligned} \tag{2.27}$$

3 TFETI

3.1 Úvod do metod rozložení oblasti

Metody rozložení oblasti jsou založeny na principu rozděl a panuj. Původní rozsáhlou úlohu rozdělí na řadu menších úloh, které je možné řešit odděleně, nezávisle na ostatních a tedy i paralelně, což nám umožňuje rychleji nalézt řešení původní úlohy.



Obrázek 3.1: Metody rozložení oblasti [6].

Nyní si krátce představíme čtyři základní typy metod s nepřekrývajícími se podoblastmi [6] (viz obr. 3.1). První z těchto metod je metoda Schurových doplňků (někdy také označována jako metoda subkonstrukcí, nebo metoda statické kondenzace) [7] založená na pásové, řídké matici soustavy, jenž získáme speciálním očíslováním neznámých. Neznámé uvnitř podoblastí jsou eliminovány a zůstává pouze redukovaná soustava s neznámými na hranicích podoblastí, která se řeší pomocí sdružených gradientních metod.

Metoda FETI (Finite Element Tearing and Interconnecting) představena Farhatem a Rouxem v roce 1992 [8], narozdíl od předešlé metody, spočívá ve fyzickém oddělení jednotlivých podoblastí při zachování spojitosti řešení. Zavádíme nové tzv. „lepící“ podmínky na hranicích mezi podoblastmi, které jsou vynuceny Lagrangeovými multiplikátory. Původní primární proměnné jsou eliminovány a úloha tak přechází v hledání duálních proměnných.

Další metoda s názvem FETI-DP (Finite Element Tearing and Interconnecting-Dual Primal) v sobě kombinuje obě předešlé metody (Schurovy doplňky a FETI). Je založena na neúplné dekompozici, kdy podoblasti jsou odděleny pouze částečně a spojitost je zachována v tzv. rozích, jak je vidět na obr. 3.1 v bloku *c*.

Poslední metoda, kterou zde představíme je TFETI (nebo také Total FETI) [1]. Z obrázku 3.1 je patrná velká podobnost s metodou FETI. Jedinou odlišností je pojetí Dirichletových podmínek, které jsou stejně jako lepící podmínky vynucovány Lagrangeovými multiplikátory. Se všemi podoblastmi tak můžeme pracovat stejně, protože matice tuhosti

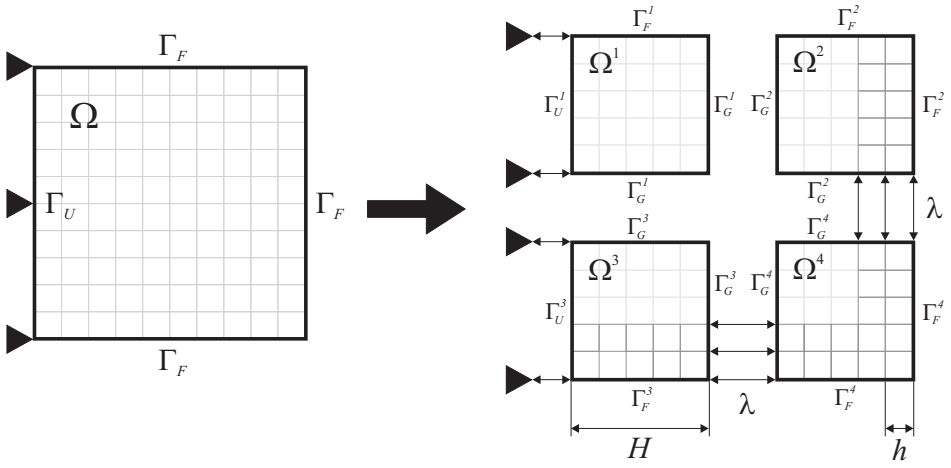
všech podoblastí mají na rozdíl od metody FETI předem známá jádra. Jediným problémem je singularita matice tuhosti, neexistuje k ní matice inverzní a je třeba hledat zobecněnou inverzi. Metoda je oproti FETI efektivnější, numericky stabilnější a navíc jednodušší na implementaci.

V následujících kapitolách se budeme podrobněji zabývat metodou TFETI tak, jak je popsána v [9], případně v [10]. Ukážeme si dekompozici původní oblasti (kapitola 3.2) a popíšeme přechod velkého primárního problému, kdy hledáme pole posunutí, na menší a lépe podmíněný duální problém s Lagrangeovými multiplikátory, jehož podmíněnost je dále zlepšena použitím projektorů (kapitola 3.3). Ve stejné kapitole si také ukážeme modifikaci metody sdružených gradientů s předpodmíněním pro řešení QP problému.

3.2 Dekompozice a primární QP problém

Prvním krokem metody TFETI je dekompozice původní spojité oblasti Ω, kterou odtrhneme od části hranice s Dirichletovými okrajovými podmínkami Γ_U. Oblast dále rozložíme na N podoblastí $\Omega^p \in \mathbb{R}^d$ s krokem H, v našem případě uvažujme $d = 2$. Každou z nich jednoznačně očíslujeme ($p = 1, 2, \ldots, N$) a zavedeme nové „lepící“ podmínky na uměle vytvořených hranicích mezi podoblastmi a na hranicích s předepsanými Dirichletovými podmínkami.

Každá hranice Γ^p podoblasti Ω^p se skládá ze tří disjunktních částí Γ_U^p, Γ_F^p a Γ_G^p, $\Gamma^p = \overline{\Gamma_U^p} \cup \overline{\Gamma_F^p} \cup \overline{\Gamma_G^p}$, kde na Γ_U^p jsou předepsané Dirichletovy okrajové podmínky reprezentující posunutí, na Γ_F^p Neumannovy okrajové podmínky reprezentující povrchové síly z původních předepsaných okrajových podmínek na Γ a nakonec Γ_G^p označuje část hranice, která je „lepena“ k sousedním podoblastem (viz obr. 3.2). Lepící podmínky vynucují spojitost posunutí na hranicích mezi podoblastmi, na hranicích s Dirichletovými podmínkami pak vynucují předepsaná posunutí.

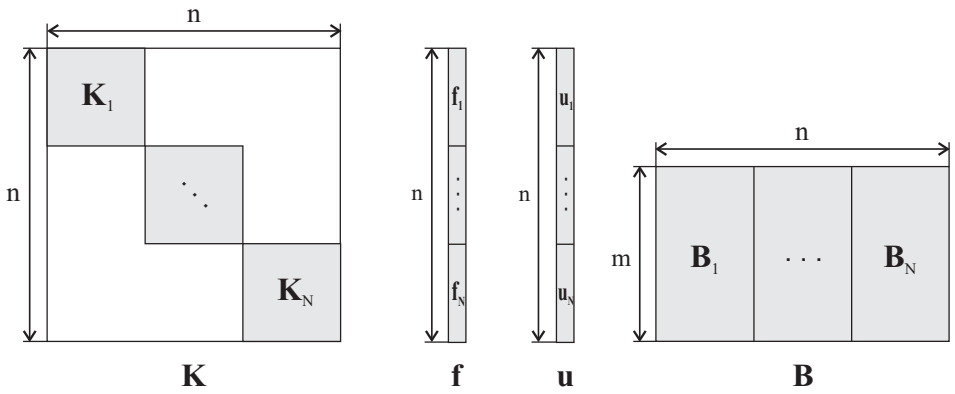


Obrázek 3.2: TFETI rozložení oblasti na očíslované podoblasti a jejich diskretizace.

Konečně-prvková diskretizace oblasti $\overline{\Omega} = \overline{\Omega^1} \cup \ldots \cup \overline{\Omega^N}$ s vhodným očíslováním uzlů vyústí v QP problém, kdy dostáváme energetickou formulaci (viz kapitola 2.1)

$$\min_u \frac{1}{2}\mathbf{u}^\top \mathbf{K}\mathbf{u} - \mathbf{f}^\top \mathbf{u} \quad \text{za podmínky } \mathbf{B}\mathbf{u} = \mathbf{c}, \tag{3.1}$$

kde $\mathbf{K} = \mathrm{diag}\,(\mathbf{K}_1, \ldots, \mathbf{K}_N)$ je symetrická pozitivně semidefinitní blokově diagonální řídká matice tuhosti rozměrů $n \times n$, $\mathbf{f}$ je sloupcový vektor zatížení velikosti n, $\mathbf{B}$ je matice vazebních podmínek plné hodnosti o rozměrech $m \times n$, $\mathbf{c}$ je vektor vazebních konstant velikosti m a $\mathbf{u}$ je vektor hledaného posunutí velikosti n. Přičemž n je celkový počet uzlů a m je celkový počet zadaných lepících a Dirichletových podmínek. Typicky je m mnohem menší než n.



Obrázek 3.3: Blokové znázornění matic a vektorů energetické formulace (3.1).

Diagonální bloky $\mathbf{K}_p$ matice $\mathbf{K}$ jsou pozitivně semidefinitní řídké matice tuhosti se známými jádry odpovídající jednotlivým podoblastem Ω^p. Všechny bloky tak mohou být efektivně zregularizovány a následně rozloženy pomocí klasické Choleského faktorizace pro regulární matice. Vektor zatížení $\mathbf{f}$ obsahuje celkové síly působící na jednotlivé uzly, které jsou výslednicemi objemových sil a/nebo jiných předepsaných napětí.

Matice $\mathbf{B}$ s řádky $\mathbf{b}_i$ a vektor $\mathbf{c}$ s prvky c_i vynucují spojitost posunutí na umělých hranicích mezi podoblastmi, tedy na Γ^p_G, a rovněž předepsaná posunutí na části hranice s předepsanými Dirichletovými podmínkami, tedy na Γ^p_U. Spojitost je vynucena podmínkou $\mathbf{b}_i\mathbf{u} = c_i = 0$, kde $\mathbf{b}_i$ jsou vektory řádu n s prvky 1 a -1 na pozicích příslušných spojovacích uzlů a s nulami na zbylých pozicích, c_i je nenulovým prvkem pouze na pozicích, kde umisťujeme Dirichletovy podmínky, na zbylých pozicích je nulový.

Matici vazebních podmínek $\mathbf{B}$ lze přímo sestavit tak, aby měla ortonormální řádky, musíme si ale dát pozor na uzly, které jsou sdíleny více než dvěma podoblastmi. V těchto případech nám totiž vznikají lineárně závislé řádky. Jestliže máme uzel sdílený dvěma podoblastmi a hranicí s předepsanými Dirichletovými podmínkami, vznikají nám tři lineárně závislé řádky $\mathbf{b}_i$. Avšak zrušením libovolného řádku tuto závislost odstraníme. Těžší

případ nastává, jestliže je uzel sdílen čtyřmi podoblastmi a vznikají tak čtyři lineárně závislé řádky $\mathbf{b}_i$. Opět vyškrtneme libovolný řádek jako v předchozím případě, zbylé řádky je ale navíc potřeba zortogonalizovat. Ukázka možného sestavení matice $\mathbf{B}$ je v příloze A.

Ačkoliv je (3.1) standardním problémem konvexního kvadratického programování, jeho formulace není vhodná pro numerické řešení. Důvody jsou typicky špatně podmíněná, singulární a rozměrná matice $\mathbf{K}$, ale také neefektivní, „drahé“ projekce na přípustnou množinu.

3.3 Přechod k duálnímu problému a předpodmínění

Výše zmíněné potíže lze redukovat použitím teorie duality konvexního programování [11], kde jsou všechna omezení vynucena Lagrangeovými multiplikátory λ. Lagrangián problému (3.1) je

$$L(\mathbf{u}, \lambda) = \frac{1}{2}\mathbf{u}^\top \mathbf{K}\mathbf{u} - \mathbf{f}^\top \mathbf{u} + \lambda^\top (\mathbf{B}\mathbf{u} - \mathbf{c}). \tag{3.2}$$

Je známo, že primární QP problém (3.1) je ekvivalentní sedlobodovému problému

$$L\left(\overline{\mathbf{u}}, \overline{\lambda}\right) = \sup_{\lambda} \inf_{u} L(\mathbf{u}, \lambda). \tag{3.3}$$

Řešení problému (3.3) vede na hledání $\left(\overline{\mathbf{u}}, \overline{\lambda}\right) \in \mathbb{R}^n \times \mathbb{R}^m$ takového, aby:

$$A \begin{pmatrix} \mathbf{u} \\ \lambda \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ \mathbf{c} \end{pmatrix}, \tag{3.4}$$

kde

$$A := \begin{pmatrix} \mathbf{K} & \mathbf{B}^\top \\ \mathbf{B} & \mathbf{O} \end{pmatrix}.$$

Předpokládáme, že (3.4) je jednoznačně řešitelná, což je zaručeno následující nutnou a postačující podmínkou:

$$Ker\, \mathbf{B}^\top = \{\mathbf{o}\}, \tag{3.5}$$

$$Ker\, \mathbf{K} \cap Ker\, \mathbf{B} = \{\mathbf{o}\}. \tag{3.6}$$

Všimněme si, že (3.5) je podmínka na matici $\mathbf{B}$ o plné řádkové hodnosti. Zaveďme matici $\mathbf{R}$ plné hodnosti a předpokládejme, že sloupce matice $\mathbf{R}$ jsou tvořeny vektory báze $Ker\, \mathbf{K}$. V našem případě je $\mathbf{R} \in \mathbb{R}^{n \times N}$ tvořena diagonálními bloky $\mathbf{R}_p$:

$$\mathbf{R}_p = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}. \tag{3.7}$$

První rovnice z (3.4) je splněna, jestliže

$$\mathbf{f} - \mathbf{B}^\top \overline{\lambda} \in Im\,\mathbf{K}, \tag{3.8}$$

což můžeme ekvivalentně zapsat jako

$$\mathbf{R}^\top \left(\mathbf{f} - \mathbf{B}^\top \overline{\lambda}\right) = \mathbf{o}, \tag{3.9}$$

a dále, jestliže

$$\overline{\mathbf{u}} = \mathbf{K}^\dagger \left(\mathbf{f} - \mathbf{B}^\top \overline{\lambda}\right) + \mathbf{R}\overline{\alpha}, \tag{3.10}$$

kde $\mathbf{K}^\dagger$ je libovolná zobecněná inverze splňující $\mathbf{K}\mathbf{K}^\dagger\mathbf{K} = \mathbf{K}$ a $\overline{\alpha} \in \mathbb{R}^N$. Nyní dosadíme (3.10) do druhé rovnice v (3.4) a dostáváme

$$-\mathbf{B}\mathbf{K}^\dagger\mathbf{B}^\top\overline{\lambda} + \mathbf{B}\mathbf{R}\overline{\alpha} = \mathbf{c} - \mathbf{B}\mathbf{K}^\dagger\mathbf{f}. \tag{3.11}$$

Rovnice (3.9) a (3.11) si přepíšeme do maticové podoby, jejíž řešením nalezneme dvojici $\left(\overline{\lambda}, \overline{\alpha}\right)$:

$$S \begin{pmatrix} \lambda \\ \alpha \end{pmatrix} = \begin{pmatrix} \mathbf{d} \\ \mathbf{e} \end{pmatrix}, \tag{3.12}$$

kde

$$S := \begin{pmatrix} \mathbf{B}\mathbf{K}^\dagger\mathbf{B}^\top & -\mathbf{B}\mathbf{R} \\ -\mathbf{R}^\top\mathbf{B}^\top & \mathbf{O} \end{pmatrix}$$

je (negativní) *Schurův doplněk* matice $\mathbf{K}$ v matici A, vektor $\mathbf{d} := \mathbf{B}\mathbf{K}^\dagger\mathbf{f} - \mathbf{c}$ a vektor $\mathbf{e} := -\mathbf{R}^\top\mathbf{f}$. Jakmile vypočteme $\left(\overline{\lambda}, \overline{\alpha}\right)$, hledané řešení $\overline{\mathbf{u}}$ pak jednoduše získáme z rovnice (3.10). Poznamenejme, že rovnice (3.12) má formálně stejnou sedlobodovou strukturu jako (3.4), nicméně její velikost je podstatně menší a diagonální blok $\mathbf{B}\mathbf{K}^\dagger\mathbf{B}^\top$ je mnohem lépe podmíněn než matice $\mathbf{K}$.

Podívejme se nyní na efektivní regularizaci matice tuhosti $\mathbf{K}$ [2, 3], která umožňuje následné rozložení pomocí klasického Choleského rozkladu pro regulární matice. To je velmi důležité právě pro efektivní vliv matice $\mathbf{K}^\dagger$ na vektor. Mějme diagonální bloky $\mathbf{K}_p$ matice $\mathbf{K}$. Přičteme-li libovolnou malou konstantu $c \in \mathbb{R}^+$ k nějákému prvku na diagonále každé matice $\mathbf{K}_p$, dostáváme zregularizované matice $\mathbf{K}_p$ a tedy i regulární matici $\mathbf{K}$, ke které existuje zobecněná inverze $\mathbf{K}^\dagger$. Nyní lze pomocí klasického Choleského rozkladu rozložit již regulární matici $\mathbf{K}$ na součin $\mathbf{L}\mathbf{L}^\top$, kde $\mathbf{L}$ je dolní trojúhelníková matice. Dále je třeba podotknout, že nemusíme provádět přímé vyčíslení $\mathbf{K}^\dagger$ ani její přímé násobení s jinou maticí, stačí pouze provést akci násobení vektorem zprava, a to (s použitím matlabovské syntaxe) následujícím způsobem:

$$\mathbf{x} = \mathbf{K}^\dagger\mathbf{b} = \left(\mathbf{L}\mathbf{L}^\top\right)^\dagger \mathbf{b} = \mathbf{L}^\top \backslash \left(\mathbf{L} \backslash \mathbf{b}\right).$$

Předtím než se podíváme na metody řešení rovnice (3.12) zavedeme nové značení

$$\mathbf{F} := \mathbf{B}\mathbf{K}^{\dagger}\mathbf{B}^{\top}, \quad \mathbf{G} := -\mathbf{R}^{\top}\mathbf{B}^{\top},$$

rovnici (3.12) pak můžeme přepsat do tvaru

$$\begin{pmatrix} \mathbf{F} & \mathbf{G}^{\top} \\ \mathbf{G} & \mathbf{O} \end{pmatrix} \begin{pmatrix} \lambda \\ \alpha \end{pmatrix} = \begin{pmatrix} \mathbf{d} \\ \mathbf{e} \end{pmatrix}. \tag{3.13}$$

Nyní rozdělíme (3.13) použitím ortogonálního projektoru $\mathbf{P_G}$ na tzv. hrubý problém $Ker\,\mathbf{G}$. Podmínka (3.5) naznačuje, že $\mathbf{G}$ je matice plné řádkové hodnosti, projektor $\mathbf{P_G}$ zavedeme následovně

$$\mathbf{P_G} := \mathbf{I} - \mathbf{Q_G} \text{ , kde } \mathbf{Q_G} := \mathbf{G}^{\top}\left(\mathbf{G}\mathbf{G}^{\top}\right)^{-1}\mathbf{G}.$$

Přičemž $\mathbf{Q_G}$ je ortogonální projektor na $Im\,\mathbf{G}^{\top}$ a $\mathbf{P_G}$ je ortogonální projektor na $Ker\,\mathbf{G}$. Aplikací projektoru $\mathbf{P_G}$ na první rovnici soustavy (3.13) obdržíme, že vektor $\overline{\lambda}$ splňuje:

$$\mathbf{P_G}\mathbf{F}\lambda = \mathbf{P_G}\mathbf{d}, \quad \mathbf{G}\lambda = \mathbf{e}. \tag{3.14}$$

Abychom (3.14) mohli upravit na jednu rovnici vektorového prostoru $Ker\,\mathbf{G}$, rozložíme řešení $\overline{\lambda}$ na $\overline{\lambda}_{Im} \in Im\,\mathbf{G}^{\top}$ a na $\overline{\lambda}_{Ker} \in Ker\,\mathbf{G}$, platí tedy:

$$\overline{\lambda} = \overline{\lambda}_{Im} + \overline{\lambda}_{Ker}\ . \tag{3.15}$$

Vektor $\overline{\lambda}_{Im}$ získáme jednoduše jako

$$\overline{\lambda}_{Im} = \mathbf{G}^{\top}\left(\mathbf{G}\mathbf{G}^{\top}\right)^{-1}\mathbf{e},$$

zbývá ukázat, jak dostaneme $\overline{\lambda}_{Ker}$. Dosazením (3.15) do (3.14) obdržíme:

$$\mathbf{P_G}\mathbf{F}\overline{\lambda}_{Ker} = \mathbf{P_G}\left(\mathbf{d} - \mathbf{F}\overline{\lambda}_{Im}\right), \quad \overline{\lambda}_{Ker} \in Ker\,\mathbf{G}. \tag{3.16}$$

Všimněme si, že tato rovnice je jednoznačně řesitelná, jelikož $\mathbf{P_G}\mathbf{F} : Ker\,\mathbf{G} \mapsto Ker\,\mathbf{G}$ je symetrická pozitivně definitní matice a existuje k ní inverze, pokud existuje inverze k A [12]. Nakonec poznamenejme, že jestliže je $\overline{\lambda}$ známá, část řešení $\overline{\alpha}$ je dána vztahem:

$$\overline{\alpha} = \left(\mathbf{G}\mathbf{G}^{\top}\right)^{-1}\mathbf{G}\left(\mathbf{d} - \mathbf{F}\overline{\lambda}\right). \tag{3.17}$$

Vliv matice $\mathbf{F}$ na vektor je velmi dobře paralelizovatelný díky blokově diagonální struktuře matice $\mathbf{K}^{\dagger}$ a blokové struktuře matice $\mathbf{B}$ respektující dekompozici na podoblasti. Na druhou stranu, vliv projektoru $\mathbf{P_G}$ nelze přímo paralelizovat. Předešlé výsledky nyní shrneme do algoritmického schématu.

Algoritmické schéma

1.1 Sestavit $\mathbf{G} := -\mathbf{R}^\top \mathbf{B}^\top$, $\mathbf{d} := \mathbf{B}\mathbf{K}^\dagger \mathbf{f} - \mathbf{c}$, $\mathbf{e} := -\mathbf{R}^\top \mathbf{f}$.
1.2 Vypočítat $\overline{\lambda}_{Im} = \mathbf{G}^\top \left(\mathbf{G}\mathbf{G}^\top\right)^{-1} \mathbf{e}$.
1.3 Sestavit $\widetilde{\mathbf{d}} := \mathbf{d} - \mathbf{F}\overline{\lambda}_{Im}$.
1.4 Vypočítat $\overline{\lambda}_{Ker}$ řešením $\mathbf{P_G}\mathbf{F}\overline{\lambda}_{Ker} = \mathbf{P_G}\widetilde{\mathbf{d}}$ na $Ker\,\mathbf{G}$.
1.5 Sestavit $\overline{\lambda} := \overline{\lambda}_{Im} + \overline{\lambda}_{Ker}$.
2 Vypočítat $\overline{\alpha} := \left(\mathbf{G}\mathbf{G}^\top\right)^{-1} \mathbf{G}\left(\mathbf{d} - \mathbf{F}\overline{\lambda}\right)$.
3 Vypočítat $\overline{\mathbf{u}} = \mathbf{K}^\dagger \left(\mathbf{f} - \mathbf{B}^\top \overline{\lambda}\right) + \mathbf{R}\overline{\alpha}$.

Nakonec si ukážeme modifikovanou metodu sdružených gradientů s předpodmíněním (PCGP) [13], kterou použijeme pro výpočet $\overline{\lambda}_{Ker}$ v kroku 1.4 algoritmického schématu. Chceme tedy vypočítat $\overline{\lambda}_{Ker}$ řešením soustavy $\mathbf{P_G}\mathbf{F}\overline{\lambda}_{Ker} = \mathbf{P_G}\widetilde{\mathbf{d}}$ na $Ker\,\mathbf{G}$ s „levným“ předpodmiňovačem $\overline{\mathbf{F}^{-1}}$ [13] k matici $\mathbf{F}$. Poznamenejme, že pro ortonormální matici $\mathbf{B}$ obdržíme

$$\overline{\mathbf{F}^{-1}} = \mathbf{B}\mathbf{K}\mathbf{B}^\top.$$

Algoritmus PCGP

1. Inicializace

   $\mathbf{r}^0 = \widetilde{\mathbf{d}}$, $\lambda^0_{Ker} = \mathbf{o}$.

2. Iterace $k = 1, 2, \ldots,$ až do konvergence

   Projekce $\mathbf{w}^{k-1} = \mathbf{P_G}\mathbf{r}^{k-1}$.
   Předpodmínění $\mathbf{z}^{k-1} = \overline{\mathbf{F}^{-1}}\mathbf{w}^{k-1}$.
   Projekce $\mathbf{y}^{k-1} = \mathbf{P_G}\mathbf{z}^{k-1}$.
   $\beta^k = \left(\mathbf{y}^{k-1}\right)^\top \mathbf{w}^{k-1} / \left(\mathbf{y}^{k-2}\right)^\top \mathbf{w}^{k-2}$; $\left(\beta^1 = 0\right)$.
   $\mathbf{p}^k = \mathbf{y}^{k-1} + \beta^k \mathbf{p}^{k-1}$; $\left(\mathbf{p}^1 = \mathbf{y}^0\right)$.
   $\alpha^k = \left(\mathbf{y}^{k-1}\right)^\top \mathbf{w}^{k-1} / \left(\mathbf{p}^k\right)^\top \mathbf{F}\mathbf{p}_k$.
   $\lambda^k_{Ker} = \lambda^{k-1}_{Ker} + \alpha^k \mathbf{p}^k$.
   $\mathbf{r}^k = \mathbf{r}^{k-1} - \alpha^k \mathbf{F}\mathbf{p}_k$.

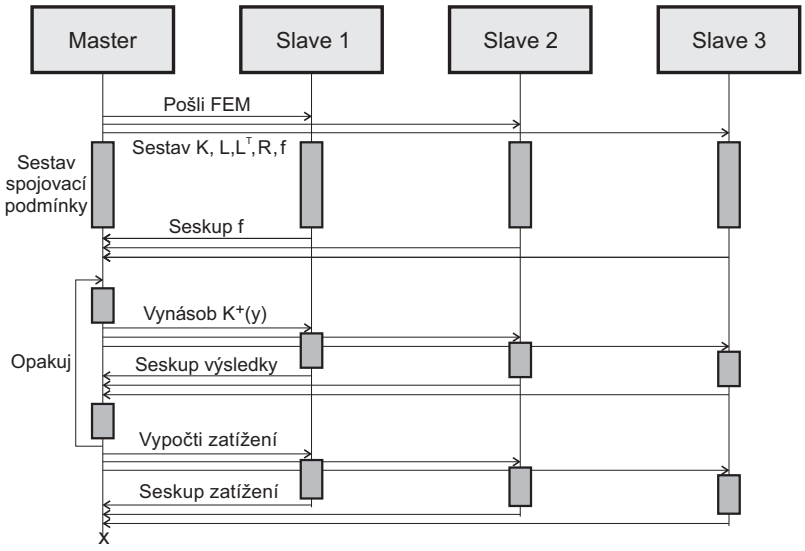
3. $\overline{\lambda}_{Ker} = \lambda^k_{Ker}$.

Použitím TFETI v kombinaci s algoritmem PCGP jsme schopni nalézt řešení původního problému v časové složitosti $O(1)$. Časová složitost maticovo-vektorového násobení nezávisle na velikosti problému daného poměrem mezi krokem dekompozice H a krokem diskretizace h zůstává stejná.

4 Paralelizace

Paralelizace výpočetně náročných algoritmů umožňuje výrazné zrychlení výpočtů a tedy i nalezení řešení v mnohem kratším čase. Původní problém řešený sekvenčním programovým kódem je totiž rozložen na řadu menších úloh, které lze řešit souběžně. V našem případě provádíme paralelizaci dat, kdy jedna výpočetní jednotka, tzv. *master*, má řídící úlohu a koordinuje průběh celého výpočtu a ostatní jednotky, označovány jako *slave*, pracují s různými částmi dat stejného kódu a jsou podřízeny masterovi.

Paralelní implementaci jsme dělali v MATLABu s využitím dvou toolboxů, *Parallel Computing Toolbox* a *Distributed Computing Server* [14]. *Parallel Computing Toolbox* nám umožňuje spouštět paralelní aplikace lokálně na vícejádrovém osobním počítači a rozložit tak početně a datově náročné problémy mezi více procesorů. *Distributed Computing Server* umožňuje spouštět aplikace na výpočetním clusteru, na němž máme možnost spustit větší počet paralelních procesů, a obsahuje základní plánovač úloh *job manager*. Dále používáme interaktivní paralelní prostředí *pmode*, které umožňuje interaktivně pracovat s paralelními procesy běžícími současně. Po zadání příkazu *pmode start local 4* se nám otevře paralelní příkazové okno se čtyřmi menšími okny představující tzv. *laby*, každý pro jeden proces. Základní komunikace mezi jednotlivými *laby* je prováděna příkazy *labSend()*, funkce pro posílání dat z jednoho *labu* na jiné, a *labReceive()*, funkce pro obdržení těchto dat. Velmi užitečné jsou také dodatkové funkce *numlabs()*, udávající celkový počet *labů*, a *labindex()*, označující index *labu*. V samotném kódu pak můžeme snadno oddělit část určenou *masterovi* a část určenou *slavům* pomocí logické podmínky typu *jestliže labindex = 1, tak ..., jinak ...*, což zachovává kód jednotným.



Obrázek 4.1: Schéma paralelního výpočtu TFETI.

Jak již bylo zmíněno na začátku předchozí kapitoly, metody rozložení oblasti jsou vhodné pro paralelní implementaci, jelikož výpočty na jednotlivých podoblastech je možné řešit odděleně a nezávisle na ostatních. U metody TFETI je navíc možné se všemi podoblastmi pracovat stejně díky odděleným Dirichletovým podmínkám, čímž je tato metoda na paralelní implementaci jednodušší [9]. Všimněme si struktury primárních dat, tedy $\mathbf{K}$, $\mathbf{B}$, $\mathbf{R}$, $\mathbf{f}$ a $\mathbf{u}$. Jsou rozděleny do bloků odpovídajících jednotlivým podoblastem, pro paralelizaci těchto dat tedy stačí, aby *master* sdělil svým *slavům*, které maticové bloky mají zpracovávat. Naopak kritickým bodem je aplikace projektoru $\mathbf{Q_G}$ a vhodné rozdělení matice $\mathbf{G}$ [9]. Obecné schéma znázorňující postup paralelního výpočtu metody TFETI vidíme na obrázku 4.1.

V našem případě jsme paralelizovali pouze matici tuhosti $\mathbf{K}$ a s ní spojené násobící funkce *multi_K* a *multi_Kplus* pro násobení příslušné matice s vektorem. Označme N_s jako počet podoblastí řešených na jednotlivých *slavech*. Nejprve je třeba pomocí příkazu *labSend* rozeslat *slavům* příslušné části dat fem struktury, které v sobě nesou informace o příslušných N_s podoblastech. Na každém *slavu* se pak sestaví N_s bloků $\mathbf{K}_p$, které dohromady tvoří matici tuhosti $\mathbf{K}$. Při použití násobících funkcí *multi_K* a *multi_Kplus* je v první fázi nutné na *masterovi* rozdělit vstupní vektor a rozeslat jednotlivé části *slavům*. V druhé fázi *slavové* provádějí samotné násobení N_s bloků matice $\mathbf{K}$ nebo $\mathbf{K}^\dagger$ s odpovídajícími částmi vstupního vektoru. Pomocí funkce *gather_vec* jsou pak jednotlivé výstupy násobení poskládány zpět do jediného výstupního vektoru, s kterým se na *masterovi* dále pracuje.

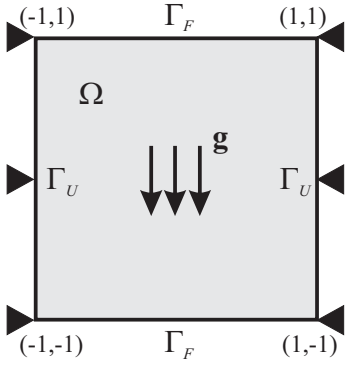
5 Numerické experimenty

K testování vytvořeného algoritmu jsme využili superpočítačové centrum VŠB-TU Ostrava (SPC) [4], které poskytuje výpočetní prostředí a výpočetní zdroje pro náročné výpočty. Využíváme zde výpočetní cluster ComSiO, který má osm výpočetních uzlů a devátý řídící uzel sloužící pro pre- a post- processing úloh. V clusteru je k dispozici 32 procesorů a maximální počet paralelně spuštěných úloh je tedy 32.

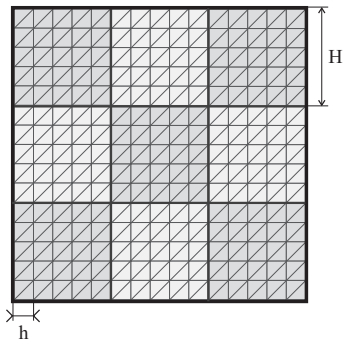
Na vybraných příkladech budeme zkoumat především numerickou a paralelní škálovatelnost metody TFETI. Numerická škálovatelnost je vlastnost, kdy počet iterací zůstává nezávislý (konstantní) na počtu neznámých. Paralelní škálovatelnost je pak vlastnost, kdy doba běhu algoritmu je nepřímo úměrná počtu využitých procesorů.

Následují dva příklady na výpočet prohnutí membrány definované na oblasti Ω s Dirichletovými a Neumannovými okrajovými podmínkami, která je zatížená silou g. Je třeba poznamenat, že následující výsledky jsou obdrženy z vlastního neoptimalizovaného kódu s tolerancí algoritmu PCGP 1E-05.

Příklad 5.1
Mějme oblast Ω o rozměru $(-1,1) \times (-1,1)$, na níž působí zatížení dané funkcí $g = -2(x^2 + y^2)$. Geometrie se vstupními daty je naznačena na obrázku 5.1. Oblast je dělena postupně na počet $1, 4, 9, \ldots, 49$ podoblastí, tj. $N = k^2, k = 1, 2, 3, \ldots, 7$. Pro testování numerické škálovatelnosti volíme počet prvků každé podoblasti vždy $2 \times (180 \times 180)$. Všechny podoblasti jsou diskretizovány stejnou trojúhelníkovou sítí s krokem h (viz obrázek 5.2) a poměr $\frac{H}{h} = 180$ zůstává konstantní. Při testování paralelní škálovatelnosti mějme oblast se stálým počtem $2 \times (540 \times 540)$ prvků, kterou následně dělíme na jednotlivé počty podoblastí. Poměr $\frac{H}{h}$ v tomto případě konstantní není. Výsledky obou testů jsou znázorněny v tabulkách 5.1 a 5.2.



Obrázek 5.1: Geometrie k příkladu 5.1.



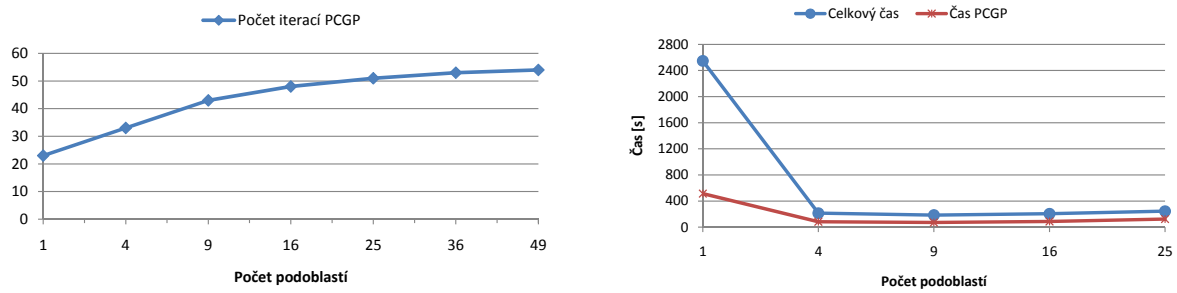
Obrázek 5.2: Ukázka diskretizace a dekompozice oblasti (9 podoblastí po $2 \times (5 \times 5)$ prvcích).

Tabulka 5.1: Numerická škálovatelnost příkladu 5.1.

Počet podoblastí	1	4	9	16	25	36	49
Primární proměnné	32 761	131 044	294 849	524 176	819 025	1 179 400	1 605 290
Duální proměnné	362	1 445	3 250	5 777	9 026	12 997	17 690
Dimenze $Ker\mathbf{K}$	1	4	9	16	25	36	49
Počet iterací PCGP	23	33	43	48	51	53	54
Čas PCGP [s]	12,64	26,12	73,12	189,9	635,8	1 569	3 240
Celkový čas [s]	33,83	80,67	189,4	403	991,4	2 121	4 022

Tabulka 5.2: Paralelní škálovatelnost příkladu 5.1.

Počet podoblastí	1	4	9	16	25
Primární proměnné	292 681	293 764	294 849	295 936	297 025
Duální proměnné	1 082	2 165	3 250	4 337	5 426
Dimenze $Ker\mathbf{K}$	1	4	9	16	25
Počet iterací PCGP	33	37	43	42	43
Čas PCGP [s]	515	82,84	71,54	88,39	124,9
Celkový čas [s]	2 547	214,5	185,3	205,8	244,9



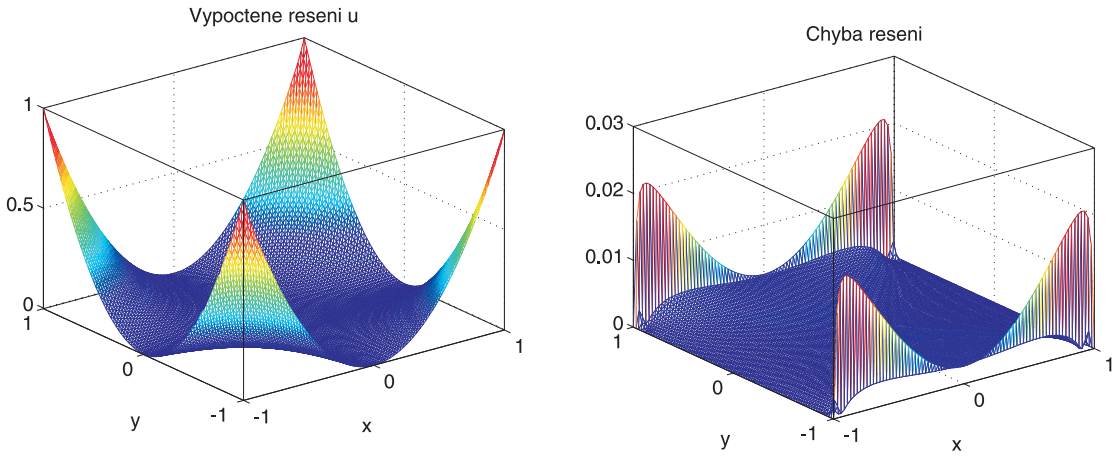
Obrázek 5.3: Vývoj počtu iterací PCGP (vlevo) a vybraných časů (vpravo) vůči vzrůstajícímu počtu podoblastí (příklad 5.1).

Na obrázku 5.3 jsou v levém grafu znázorněny počty iterací PCGP pro všechny varianty počtu podoblastí. Počet iterací se nejprve zvýší na dvojnásobek, pak ale dochází k ustálení. Počet kroků se dále nenavyšuje a je shora ohraničen. Všimněme si, že u první varianty máme téměř 33 tisíc neznámých a u poslední pak více než 1,6 milionů (viz tabulka 5.1). Můžeme tedy bezpečně prohlásit, že numerická škálovatelnost byla potvrzena.

Na obrázku 5.3 v pravém grafu je zaznamenán celkový čas a čas algoritmu PCGP v závislosti na vzrůstajícím počtu podoblastí. Při rozdělení oblasti na 4 podoblasti se celkový

čas výpočtu snížil více než 10x a čas algoritmu PCGP více než 6x oproti první variantě. Při dalším dělení dochází už pouze k malému snížení obou časů a následně dokonce k mírnému zvyšování. Pro splnění paralelní škálovatelnosti by mělo být zachované klesání časů až do 25 podoblastí. Odchylka je zřejmě způsobena právě použitím vlastního neoptimalizovaného algoritmu.

Grafické výstupy příkladu 5.1 jsou znázorněny na obrázku 5.4. Všimněme si, že chyba řešení nejvíce narůstá na hranicích s Neumannovou okrajovou podmínkou. Tyto podmínky jsou totiž obecně mnohem hůře splněny než Dirichletovy okrajové podmínky.

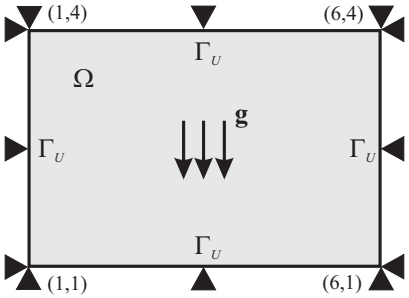


Obrázek 5.4: Vypočtený průhyb membrány (vlevo) a chyba (vpravo) příkladu 5.1 (oblast dělená na 16 podoblastí po $2 \times (20 \times 20)$ prvcích).

■

Příklad 5.2

Oblast Ω je rozměru $(1,6) \times (1,4)$ a působí na ni síla daná funkcí $g = 2\sin(x)\cos(y)$. Narozdíl od předchozího příkladu je oblast uchycena na všech stranách (viz obrázek 5.5), diskretizace podoblastí ale zůstává stejná (viz obrázek 5.2). Oblast opět dělíme na $1, 4, 9, \ldots, 49$ podoblastí, tj. $N = k^2$, $k = 1, 2, 3, \ldots, 7$. Při testování numerické škálovatelnosti je pak na podoblasti stálý počet prvků $2 \times (180 \times 180)$ a je zachován pevný poměr $\frac{H}{h} = 180$. Při testování paralelní škálovatelnosti dělíme, stejně jako v předchozím příkladě 5.1, oblast s počtem $2 \times (540 \times 540)$ prvků postupně na jednotlivé počty podoblastí. Výsledky obou testů jsou shrnuty v tabulkách 5.3 a 5.4.



Obrázek 5.5: Geometrie k příkladu 5.2

Tabulka 5.3: Numerická škálovatelnost příkladu 5.2.

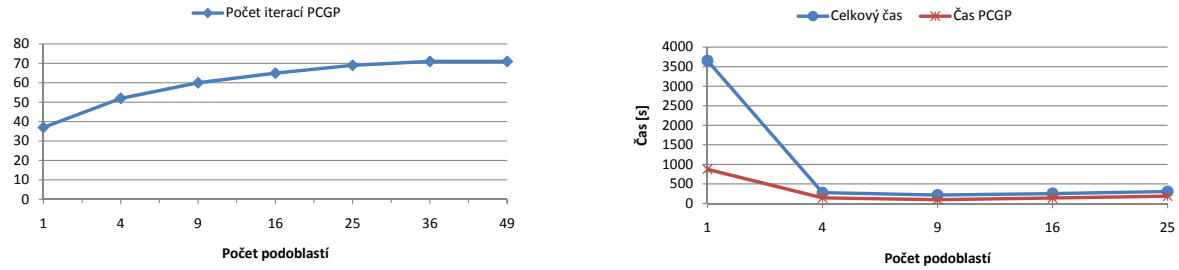
Počet podoblastí	1	4	9	16	25	36	49
Primární proměnné	32 761	131 044	294 849	524 176	819 025	1 179 400	1 605 290
Duální proměnné	720	2 163	4 328	7 215	10 824	15 155	20 208
Dimenze $Ker\mathbf{K}$	1	4	9	16	25	36	49
Počet iterací PCGP	37	52	60	65	69	71	71
Čas PCGP [s]	20,6	42,59	102,3	258,9	963	2 114	4 136
Celkový čas [s]	39,6	91,93	216	470	1 318	2 656	4 891
Chyba řešení	5,09E-05	1,28E-05	5,68E-06	3,19E-06	2,04E-06	1,42E-06	1,05E-06

Tabulka 5.4: Paralelní škálovatelnost příkladu 5.2.

Počet podoblastí	1	4	9	16	25
Primární proměnné	292 681	293 764	294 849	295 936	297 025
Duální proměnné	2 160	3 243	4 328	5 415	6 504
Dimenze $Ker\mathbf{K}$	1	4	9	16	25
Počet iterací PCGP	57	62	60	59	56
Čas PCGP [s]	874,7	143,1	98,01	139,3	186,3
Celkový čas [s]	3 653	276,3	214,5	253,8	305,3

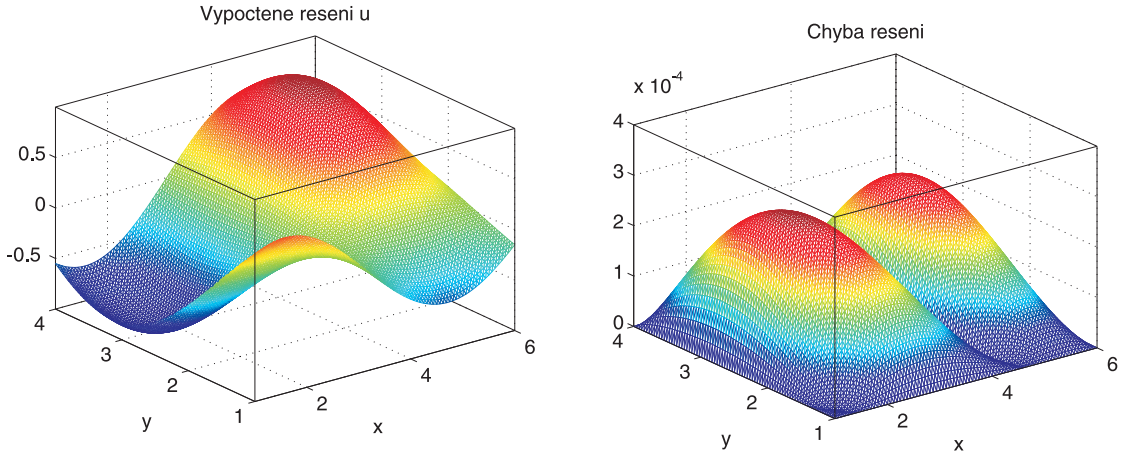
Počty iterací PCGP v závislosti na počtu podoblastí jsou znázorněny v levém grafu obrázku 5.6. Podobně jako u předchozího příkladu počet iterací nejprve vzroste, v tomto případě zhruba 1,8x, a následně se ustálí. Pro 36 a 49 podoblastí je pak tento počet stejný (viz tabulka 5.3) a můžeme předpokládat, že při dalším dělení oblasti nepřesáhne hranici 80 iterací. Numerická škálovatelnost tak byla opět potvrzena.

Na obrázku 5.6 v pravém grafu je zachycen celkový čas a čas algoritmu PCGP vůči vzrůstajícímu počtu podoblastí. Vývoj obou časů je v podstatě stejný jako v předchozím příkladě 5.1.



Obrázek 5.6: Vývoj počtu iterací PCGP (vlevo) a vybraných časů (vpravo) vůči vzrůstajícímu počtu podoblastí (příklad 5.2).

Grafické výstupy příkladu 5.2 jsou znázorněny na následujícím obrázku 5.7. Celková chyba řešení se pohybuje v řádu 10^{-5} až 10^{-6} (viz tabulka 5.3).



Obrázek 5.7: Vypočtený průhyb membrány (vlevo) a chyba (vpravo) příkladu 5.2 (oblast dělená na 16 podoblastí po $2 \times (20 \times 20)$ prvcích).

■

6 Závěr

V práci byla popsána metoda rozložení oblasti TFETI a její paralelní implementace. Byla ukázána diskretizace oblasti pomocí MKP a následně provedena dekompozice na nepřekrývající se podoblasti. Dále byl popsán přechod k duálnímu problému, kdy hledáme Lagrangeovy multiplikátory λ, které vynucují lepící podmínky mezi podoblastmi a také Dirichletovy okrajové podmínky na hranici Γ_u. Problém singularity matice tuhosti byl vyřešen její efektivní regularizací, která umožňuje použití klasického Choleského rozkladu pro regulární matice.

V rámci numerických experimentů jsme testovali a ověřovali numerickou a paralelní škálovatelnost TFETI na dvou úlohách zatížení membrány. Pro tyto výpočty jsme využili vlastní neoptimalizovanou implementaci dané metody. Počítali jsme na 28 procesorech výpočetního clusteru ComSiO až do 1 605 290 neznámých. V našem případě se jednalo pouze o jednoduché úlohy, metoda se ale plně využívá pro náročné inženýrské problémy, např. pro řešení nelineárních problémů mechaniky [6, 15], které spadají do výzkumných aktivit Katedry aplikované matematiky a rovněž do projektu IT4Innovations.

Vlastní implementace by mohla být samozřejmě dále vylepšena o celkovou paralelizaci, která je nyní omezena pouze na matici tuhosti a s ní spojené násobící akce. Rovněž by bylo zapotřebí optimalizovat kód pro zvýšení efektivity metody a získání tak ještě lepších výsledků.

Díky této práci jsem měla možnost seznámit se s metodou TFETI, jak po stránce teoretické, tak i po stránce praktické. Dále se seznámit s paralelním programováním v MATLABu a rovněž si vyzkoušet práci v prostředí superpočítačového centra VŠB - TU Ostrava.

Pavla Jirůtková

7 Reference

[1] Z. Dostál, D. Horák, R. Kučera, *Total FETI - an easier implementable variant of the FETI method for numerical solution of elliptic PDE.* Communications in Numerical Methods in Engineering, 22(2006), 12, s. 1155-1162.

[2] Z. Dostál, T. Kozubek, A. Markopoulos, M. Mensik, *Cholesky decdomposition and a generalized inverse of the stiffness matrix of a floating structure with known null space.* Applied Mathematics and Computation, 2011; **217**:6067-6077.

[3] T. Brzobohatý, Z. Dostál, P. Kovář, T. Kozubek, A. Markopoulos, *Cholesky decomposition with fixing nodes to stable evaluation of a generalized inverse of the stiffness matrix of a floating structure.* IJNME, DOI: 10.1002/nme.3187.

[4] VŠB - TU Ostrava, SPC VŠB - TU Ostrava [online], <http://spc.vsb.cz>.

[5] T. Kozubek, T. Brzobohatý, V. Hapla, M. Jarošová, A. Markopoulos, *Lineární algebra s Matlabem.* Matematika pro inženýry 21. století, reg. č. CZ.1.07/2.2.00/07.0332, 2011.

[6] A. Markopoulos, *Škálovatelné metody rozložení oblasti k řešení statických úloh mechaniky.* Dizertační práce, VŠB - TU Ostrava, 2009. Kap. 5, 8.

[7] J. Kruis, *Metoda Schurových doplňků.* ČVUT Praha, 2006. Dostupné z <http://www.cvut.cz/pracoviste/odbor-rozvoje/dokumenty/hab_inaug/hp/2006/>

[8] C. Farhat, F-X. Roux, *An Unconventional Domain Decomposition Method for an Efficient Parallel Solution of Large-Scale Finite Element Systems.* SIAM Journal on Sci. and Stat. Computing 13, 1992; s. 379–396.

[9] T. Kozubek, V. Vondrak, M. Mensik, D. Horak, Z. Dostal, V. Hapla, P. Kabelikova, M. Cermak, *Total FETI domain decomposition method and its massively parallel implementation.* Zasláno k publikaci, 2011.

[10] V. Hapla, *Vytvoření rozhraní knihovny MatSol pro paralelní řešení kontaktních úloh.* Diplomová práce, VŠB - TU Ostrava, 2010.

[11] Z. Dostal, *Optimal Quadratic Programming Algorithms, with Applications to Variational Inequalities.* 1. vydání, SOIA 23, Springer US, New York, 2009.

[12] J. Haslinger, T. Kozubek, R. Kučera, G. Peichl, *Projected Schur complement method for solving non-symmetric saddle-point systems arising from fictitious domain approach.* Numerical Linear Algebra with Applications, 2007; 14(9): s. 713–739.

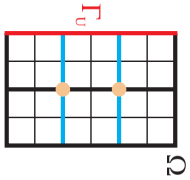
[13] C. Farhat, J. Mandel, F-X. Roux, *Optimal convergence properties of the FETI domain decomposition method.* Comput. Methods Appl. Mech. and Eng. 115, 1994; s. 365–385.

[14] The MathWorks, Inc., Products and Services [online], <http://www.mathworks.com/products/>.

[15] Z. Dostál, T. Kozubek, V. Vondrák, T. Brzobohatý, A. Markopoulos, *Scalable TFETI algorithm for the solution of multibody contact problems of elasticity.* Int. J. Numer. Meth. Engng. 82(11), 1384–1405, 2010.

A Matice B

Ukázka sestavení matice **B** oblasti Ω, která je složena z 2×3 podoblastí, s počtem prvků 2×2 na každé podoblasti a s Dirichletovými okrajovými podmínkami na jedné straně. Řádky matice odpovídající případu, kdy je uzel sdílen čtyřmi podoblastmi, jsou vyznačeny a ponechány bez úprav, tj. jsou lineárně závislé a nezortogonalizované.



	1. podoblast									2. podoblast									3. podoblast									4. podoblast									5. podoblast									6. podoblast								
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	-1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	-1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	1	0	0	0	0	0
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	1	0	0	0	0

Obrázek A.1: Ukázka sestavení matice **B**.