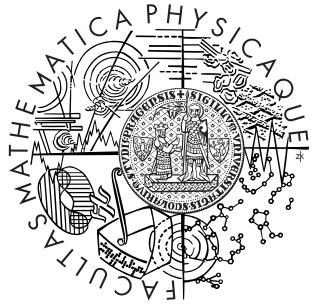
Univerzita Karlova v Praze

Matematicko-fyzikální fakulta

RIGORÓZNÍ PRÁCE



Mgr. Jitka Štrausová

Řešení soustav lineárních rovnic s obroubenou maticí

Katedra numerické matematiky

Studijní program: výpočtová matematika, software

Děkuji panu Doc. RNDr. Vladimíru Janovskému, DrSc. a panu Doc. RNDr. Janu Zítkovi, CSc. za věnovaný čas a cenné připomínky. Dále pak děkuji panu Ondřeji Liberdovi, Ph.D., který mi významně pomohl s programovací částí rigorózní práce. Také bych chtěla poděkovat slečně Jaroslavě Schovancové za pomoc při práci s typografickým systémem LaTeX.

Prohlašuji, že jsem svou rigorózní práci napsala samostatně a výhradně s použitím citovaných pramenů. Souhlasím se zapůjčováním práce.

V Praze dne 29. srpna 2007 Mgr. Jitka Štrausová

Obsah

Název práce: *Řešení soustav lineárních rovnic s obroubenou maticí*
Autor: *Mgr. Jitka Štrausová*
Katedra (ústav): *Katedra numerické matematiky*
Abstrakt: *Rigorózní práce se zabývá srovnáním dvou algoritmů pro řešení soustav lineárních rovnic s obroubenou maticí M. Tato matice se skládá ze čtyř bloků (matic A, B, C, D), z nichž levý horní blok představuje řídká, špatně podmíněná a strukturovaná matice A. Ostatní bloky (B, C, D) mají ve srovnání s A malé rozměry a jsou husté. Říkáme, že matice A je obroubená maticemi B, C, D. Je výhodné zachovat tuto blokovou strukturu matice M a využít tak řídkost matice A. K tomu lze použít jeden ze dvou algoritmů.*

Prvním z algoritmů je přímá metoda BEM pro matice s jednoduchým vroubením a její rekurzivní varianta BEMW pro matice s širším vroubením. Druhým algoritmem je iterační metoda. Oba algoritmy jsou založeny na řešení soustavy lineárních rovnic pomocí blokového LU rozkladu matice M. Každý z algoritmů však používá jiný LU rozklad.
Přínos práce, původní výsledky: *detailní numerická analýza zpětné stability uvedených algoritmů, implementace a numerické testy*
Klíčová slova: *soustava lineárních rovnic, obroubená matice, zpětná stabilita*

Title: *Solving bordered linear systems*
Author: *Mgr. Jitka Štrausová*
Department: *Department of Numerical Mathematics*
Abstract: *The comparison of two algorithms for solving bordered linear systems is considered. The matrix of this system consists of four blocks (matrices A, B, C, D), the upper left one is a sparse matrix A, which is ill-conditioned and structured. The other blocks (B, C, D) are dense. We say that the matrix A is bordered with the matrices B, C, D. It is desirable to preserve the block structure of the matrix and take advantage of sparsity and structure of the matrix A. The literature suggests to use two different algorithms:*

The first one is the method BEM for matrices with the borders of width equal to one. The recursive alternative for matrices with wider borders is called BEMW. The second algorithm is an iterative method. Both techniques are based on different variants of the block LU-decomposition.
Contribution: *detailed backward analysis of the above algorithms, implementation and numerical tests*
Keywords: *linear system, bordered matrix, backward stability*

Úvod

Rigorózní práce se zabývá řešením soustavy lineárních rovnic $Mz = b$ s maticí

$$M = \begin{pmatrix} A & B \\ C^T & D \end{pmatrix},$$

kde $A \in \mathbb{R}^{n\times n}$, $B, C \in \mathbb{R}^{n\times m}$, $D \in \mathbb{R}^{m\times m}$. Říkáme, že matice A je obroubená maticemi B, C, D.

Tyto soustavy se často vyskytují v numerické kontinuaci a bifurkačních problémech. Obvykle $n \gg m$, v mnoha aplikacích je dokonce $m = 1$. Protože matice A vzniká z diskretizace nějakého problému, je obvykle řídká a strukturovaná a mnohdy špatně podmíněná. Naopak matice B, C, D jsou husté, proto struktura celé matice M je mnohem složitější než struktura matice A. Je výhodnější zachovat tuto 'obroubenou strukturu' matice M a použít speciální algoritmus pro řešení soustav s obroubenou maticí, protože pak můžeme využít řídkost a strukturovanost matice A a ušetřit tak výpočetní čas.

Předpokládejme, že matice M je regulární a matice A je řídká a špatně podmíněná (případně singulární) a $n \gg m$. Matice M nechť je dobře podmíněná.

Pro řešení těchto soustav existují dva algoritmy. Prvním je přímá metoda BEM (pro $m = 1$), resp. BEMW (pro $m > 1$), jejímiž autory jsou W. Govaerts a J. D. Pryce. Druhý (iterační) algoritmus představili M. Paprzycki a P. Y. Yalamov. Cílem rigorózní práce je srovnání těchto dvou algoritmů.

První kapitola představuje pojmy zaokrouhlovací chyba, citlivost úlohy (v našem případě tedy podmíněnost matice) a zpětná stabilita. Ta je důležitým kritériem při posuzování kvality algoritmu a tedy i při srovnávání algoritmů.

Ve druhé kapitole je pomocí dvou dílčích algoritmů představen algoritmus BEM pro matici A obroubenou pouze vektory. Tuto metodu ale obvykle nelze použít pro matice s širším vroubením ($m > 1$). Zdůvodnění tohoto tvrzení najdeme také ve druhé kapitole. Následuje popis algoritmu BEMW pro matici A s širším vroubením, který je koncipován jako rekurzivní varianta algoritmu BEM. Závěr kapitoly je věnován zpětné stabilitě algoritmů BEM a BEMW.

Druhý algoritmus je iterační. Jde o klasický LU rozklad s částečnou pivotací. Případná singularita matice A je odstraněna umělou perturbací. Řešení získané pomocí LU rozkladu se ještě iteračně zpřesňuje.

Poznatky týkající se zpětné stability obou algoritmů byly převzaty z [3] a [4] pro algoritmus BEM, resp. BEMW a z [9] pro iterační algoritmus.

Poslední, čtvrtá kapitola obsahuje výsledky numerických testů, srovnává oba algoritmy z hlediska výpočetního času a přesnosti vypočteného řešení. Oba algoritmy se ukazují jako téměř stejně přesné, co se týká rychlosti, vychází z provedených testů lépe algoritmus iterační.

Kapitola 1

Numerická stabilita

Rigorózní práce se zabývá řešením soustavy lineárních rovnic. Řešení soustavy lineárních rovnic pomocí numerické metody se ale neobejde bez zaokrouhlovacích chyb.

Vstupní data mohou být zatížena nepřesností a při výpočtu se v každé aritmetické operaci provádí další zaokrouhlení. Chyby ve vstupních datech a dále chyby vzniklé zaokrouhlováním během výpočtu se projeví v konečném výsledku. Proto chceme, aby numerická metoda byla stabilní. Kdybychom totiž použili numerickou metodu, která není stabilní, řešení, které bychom dostali, by vlivem zaokrouhlovacích chyb pravděpodobně nebylo příliš blízko řešení přesnému.

Stabilita algoritmu je tedy nepochybně důležitým kritériem určujícím kvalitu algoritmu. Cílem rigorózní práce je srovnání dvou algoritmů. Proto první kapitolu věnujeme numerické stabilitě.

1.1 Zaokrouhlovací chyba

Jakákoliv operace prováděná pomocí počítače je zatížena zaokrouhlovacími chybami. Je to způsobeno tím, že v počítači lze reprezentovat pouze konečnou podmnožinu reálných čísel. Každé číslo reprezentovatelné v počítači má pouze konečný počet cifer. Hovoříme o tzv. konečné aritmetice.

Nechť $\beta \in \mathbb{N}$, $\beta \geq 2$ a nechť x je reálné číslo s konečným počtem cifer a_k, přičemž $0 \leq a_k < \beta$ pro $k = -m, \ldots, n$. Množina čísel

$$x_\beta = (-1)^s \left[a_n a_{n-1} \ldots a_1 a_0 . a_{-1} a_{-2} \ldots a_{-m}\right], \quad a_n \neq 0$$

se nazývá *poziční číselná soustava se základem* β. Symbol mezi a_0 a a_{-1} se nazývá *řádová tečka*. (V případě $\beta = 10$ hovoříme o desetinné tečce, v případě $\beta = 2$ jde o binární tečku.) Číslo s určuje znaménko čísla x a nabývá hodnoty $s = 0$ (pro x kladné) nebo $s = 1$ (pro x záporné).

Přirozeným základem číselné soustavy je $\beta = 10$. V počítačích se obvykle používá číselná soustava se základem $\beta = 2$ (tzv. binární soustava). V tomto případě v zápisu čísla vystupují pouze cifry 0 a 1, které se nazývají bity.

Reálné číslo lze tedy v konečné aritmetice aproximovat pomocí čísla s konečným počtem cifer. Předpokládejme, že v počítači je pro každé číslo vyhrazeno N paměťových míst. (Přesněji, pro $\beta = 2$, se jedná o N bitů operační paměti.) Pro každé reálné číslo lze jeho aproximaci uložit dvěma způsoby.

První možností je, že jedno místo je věnováno znaménku, $N-k-1$ míst zabírají cifry před řádovou tečkou a k míst zabírají cifry za řádovou tečkou. Tedy

$$x = (-1)^s[a_{N-2}a_{N-3}\dots a_k.a_{k-1}\dots a_0],$$

kde

a) $s = 1$ nebo $s = 0$,
b) cifry a_i splňují $0 \leq a_i < \beta$ pro $i = 0 \dots N$,
c) $\beta \in \mathbb{N}$ a $\beta \geq 2$ je základ číselné soustavy.

Množina takovýchto čísel se nazývá *systém s pevnou řádovou tečkou* (fixed-point system). V tomto systému počet zobrazitelných čísel není příliš velký, vezmeme-li v úvahu velký počet paměťových míst, které každé číslo zabírá. To není příliš výhodné, proto se častěji užívá druhá možnost reprezentace čísla v počítači.

Druhou možností, jak reprezentovat číslo v počítači, je *systém s pohyblivou řádovou tečkou* (floating-point system). V tomto případě je aproximace čísla x uložena následujícím způsobem:

$$x = (-1)^s \cdot (0.a_1a_2\dots a_t) \cdot \beta^e = (-1)^s \cdot m \cdot \beta^{e-t},$$

kde

a) $t \in \mathbb{N}$ je počet cifer,
b) $\beta \in \mathbb{N}$, $\beta \geq 2$ je základ číselné soustavy,
c) $0 \leq a_i \leq \beta - 1$, $i = 1, \dots, t$ jsou cifry,
d) $m = a_1 \dots a_t$, $m \leq \beta^t - 1$ je tzv. mantisa,
e) $e \in \mathbb{Z}$, $L \leq e \leq U$ se nazývá exponent,
f) $L \in \mathbb{Z}$ a $U \in \mathbb{Z}$ je horní a dolní mez exponentu e.

V této reprezentaci nejsou čísla rozmístěna rovnoměrně. Vzdálenost mezi dvěma po sobě jdoucími čísly je nejméně $\beta^{-1}eps\,|x|$ a nejvíce $eps\,|x|$, kde $eps = \beta^{1-t}$ je *strojové epsilon*.

Strojové epsilon je definováno jako vzdálenost mezi číslem 1 a jeho nejbližším sousedem. Tedy eps je nejmenší takové číslo, že

$$1 + eps > 1.$$

Označme systém s pohyblivou řádovou tečkou $\mathbb{F}(\beta, t, U, L) = \mathbb{F}$. Je-li $x \in \mathbb{R}$, pak nemusí platit $x \in \mathbb{F}$. V počítači ukládáme pouze aproximaci čísla x, která vznikne tzv. zaokrouhlením. Označíme-li zaokrouhlenou hodnotu reálného čísla x symbolem $fl(x)$, pak $x - fl(x)$ je tzv. *zaokrouhlovací chyba*. Platí následující tvrzení.

Tvrzení 1.1. *Jestliže $x \in \mathbb{R}$ je takové, že $x_{min} \leq |x| \leq x_{max}$ (kde x_{min} a x_{max} je nejmenší, resp. největší zobrazitelné číslo), pak*

$$fl(x) = x(1 + \delta), \quad |\delta| \leq \mathbf{u},$$

kde

$$\mathbf{u} = \frac{1}{2}\beta^{1-t} = \frac{1}{2}eps$$

se nazývá zaokrouhlovací jednotka.

Tvrzení vlastně říká, že pro zaokrouhlovací chybu platí: $|x - fl(x)| \leq \mathbf{u}$.

Podobně ani výsledek nějaké aritmetické operace prvků z $\mathbb{F}$ nemusí ležet v $\mathbb{F}$. Je-li $op : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$, $op \in \{+, -, \cdot, :\}$ aritmetická operace definovaná na množině reálných čísel, definujeme $\widetilde{op} : \mathbb{F} \times \mathbb{F} \to \mathbb{F}$ následujícím způsobem:

$$x \ \widetilde{op} \ y = fl(x \ op \ y)$$

Zaokrouhlovací chyby tedy vznikají i při provádění aritmetických operací. Abychom je mohli popsat, zavedeme následující označení. (Toto označení budeme potřebovat později.)

Symboly $\theta_1, \theta_2, \ldots$ budou značit skalár nebo matici $n \times n$, která je 'blízko' matice jednotkové. Ve skalárním případě $\theta \in 1(\delta)$ (kde δ je nezáporná konstanta) znamená, že $\theta = e^{\epsilon}$, kde $|\epsilon| \leq \delta$. V případě, že θ je matice, $\theta \in \ 1(\delta)$ znamená, že θ je součin konečného počtu matic $exp(E_i)$, kde $\sum_i \|E_i\| \leq \delta$.

Nechť $x, y \in \mathbb{F}$. Potom platí (viz [3])

$$fl(x \ op \ y) = \theta(x \ op \ y), \quad \theta \in 1(\mathbf{u}).$$

Je-li $x, y \in \mathbb{R}^n$, pak existuje konstanta C_{IP} taková, že

$$fl(x^T y) = x^T \theta y, \quad \theta \in (C_{IP}\mathbf{u}),$$

kde θ je diagonální matice a $C_{IP} \leq n$.

1.2 Citlivost úlohy

Citlivost úlohy vyjadřuje, jak se malá změna ve vstupních datech projeví na řešení úlohy, bez ohledu na výběr algoritmu.

Definice 1.1. *Číslem podmíněnosti matice $A \in \mathbb{R}^{n\times n}$ rozumíme číslo*

$$\kappa(A) = \|A\| \left\|A^{-1}\right\|,$$

kde $\|\cdot\|$ je maticová norma generovaná nějakou vektorovou normou.

Jestliže číslo podmíněnosti matice je malé, pak relativně malé změny ve vstupních datech způsobí relativně malé změny v řešení. (Proč, to si ukážeme ve větě 1.1.) V tomto případě hovoříme o dobře podmíněné matici.

Pokud je naopak číslo podmíněnosti velké (t.j. relativně malé změny vstupních dat způsobí relativně velké změny řešení), říkáme, že matice je špatně podmíněná.

Uvažujme soustavu

$$Ax = b, \tag{1.1}$$

kde $A \in \mathbb{R}^{n\times n}$, $x, b \in \mathbb{R}^n$. Nechť $x + \Delta x$ je řešení soustavy s porušenou maticí a s porušenou pravou stranou, tedy

$$(A + \Delta A)(x + \Delta x) = b + \Delta b. \tag{1.2}$$

Následující věta ukazuje, jak se změní řešení v závislosti na porušení koeficientů problému. K jejímu důkazu budeme potřebovat toto pomocné lemma.

Lemma 1.1. *Nechť $A \in \mathbb{R}^{n\times n}$ je matice taková, že $\|A\| < 1$. Pak platí:*

1. $I + A$ je regulární;

2. $\|(I + A)^{-1}\| \leq \frac{1}{1-\|A\|}$;

3. $\|(I + A)^{-1} - I\| \leq \frac{\|A\|}{1-\|A\|}$.

Důkaz. 1. Předpokládejme, že matice $I + A$ je singulární. To znamená, že existuje vektor $y \neq 0$ takový, že $y + Ay = 0$. Z toho ale plyne, že matice A má vlastní číslo $\lambda = -1$, a tím pádem je $1 \leq \rho(A) \leq \|A\|$. To je však spor s předpokladem, že $\|A\| < 1$.
2. Označme $Q = (I + A)^{-1}$. Odtud plyne $I = Q + QA$. Víme, že $\|I\| = 1$. Tedy

$$1 = \|(Q + QA)\| \geq \|Q\| - \|QA\| \geq \|Q\| - \|Q\| \|A\|.$$

Odtud máme

$$\|Q\| \leq \frac{1}{1 - \|A\|}. \tag{1.3}$$

3. Z rovnosti $I = Q + QA$ a z nerovnosti (1.3) plyne

$$\|I - Q\| = \|QA\| \leq \|Q\| \, \|A\| \leq \frac{\|A\|}{1 - \|A\|}$$

□

Věta 1.1. *Nechť $A \in \mathbb{R}^{n\times n}$ je regulární matice, x je přesné řešení soustavy (1.1) a $x+\Delta x$ je přesné řešení soustavy (1.2). Nechť matice ΔA je taková, že $\|A^{-1}\Delta A\| < 1$. Pak platí*

$$\frac{\|\Delta x\|}{\|x\|} \leq \frac{\kappa(A)}{1 - \|A^{-1}\Delta A\|} \left(\frac{\|\Delta A\|}{\|A\|} + \frac{\|\Delta b\|}{\|b\|} \right). \tag{1.4}$$

Důkaz. Protože $A+\Delta A = A\,(I + A^{-1}\Delta A)$ a matice A je regulární podle předpokladu a matice $I + A^{-1}\Delta A$ je regulární podle lemmatu 1.1, je i matice $A + \Delta A$ regulární a můžeme tedy psát

$$\Delta x = (x + \Delta x) - x = (A + \Delta A)^{-1}(b + \Delta b) - x =$$

$$= \left[A(I + A^{-1}\Delta A)\right]^{-1} (b + \Delta b) - x = (I + A^{-1}\Delta A)^{-1}A^{-1}(b + \Delta b) - x =$$

$$= \left[(I + A^{-1}\Delta A)^{-1} - I\right] x + (I + A^{-1}\Delta A)^{-1}A^{-1}\Delta b.$$

Z lemmatu 1.1 (bod 3) plyne

$$\|\Delta x\| \leq \frac{1}{1 - \|A^{-1}\Delta A\|} \left(\left\|A^{-1}\Delta A\right\| \|x\| + \left\|A^{-1}\Delta b\right\| \right). \tag{1.5}$$

Vynásobíme-li (1.5) číslem $\frac{1}{\|x\|}$ a použijeme-li, že $\|XY\| \leq \|X\| \, \|Y\|$, dostaneme

$$\frac{\|\Delta x\|}{\|x\|} \leq \frac{1}{1 - \|A^{-1}\Delta A\|} \left(\left\|A^{-1}\right\| \|A\| \frac{\|\Delta A\|}{\|A\|} + \frac{\|A^{-1}\| \, \|\Delta b\|}{\|x\|} \right). \tag{1.6}$$

Protože platí $\frac{1}{\|A\|\|x\|} \leq \frac{1}{\|Ax\|}$, máme

$$\frac{\|\Delta x\|}{\|x\|} \leq \frac{\|A^{-1}\| \, \|A\|}{1 - \|A^{-1}\Delta A\|} \left(\frac{\|\Delta A\|}{\|A\|} + \frac{\|\Delta b\|}{\|b\|} \right) = \frac{\kappa(A)}{1 - \|A^{-1}\Delta A\|} \left(\frac{\|\Delta A\|}{\|A\|} + \frac{\|\Delta b\|}{\|b\|} \right)$$

□

Vidíme, že relativní změna řešení závisí na relativních změnách koeficientů matice a pravé strany. Hlavní odhad závisí přímo na čísle podmíněnosti $\kappa(A)$.

Předpokládejme, že platí $\|\Delta A\| \leq \gamma \, \|A\|$ a $\|\Delta b\| \leq \gamma \, \|b\|$, kde γ je kladné číslo, které nějakým způsobem závisí na zaokrouhlovací jednotce $\mathbf{u}$ (například můžeme volit $\gamma = \beta^{1-t} = 2\mathbf{u}$). To znamená, že poruchy dat $\|\Delta A\|$ a $\|\Delta b\|$ jsou omezené pomocí $\|A\|$, resp. $\|b\|$ a pomocí charakteristiky systému s pohyblivou řádovou tečkou, který používáme. Nerovnost (1.1) tedy ještě rozšíříme v následující větě.

Věta 1.2. *Předpokládejme, že* $\|\Delta A\| \leq \gamma \|A\|$, $\|\Delta b\| \leq \gamma \|b\|$, *kde* $\gamma \in \mathbb{R}^+$ *a* $\Delta A \in \mathbb{R}^{n\times n}$, $\Delta b \in \mathbb{R}^n$. *Jestliže* $\gamma\kappa(A) < 1$, *pak platí následující nerovnosti*

$$\frac{\|x + \Delta x\|}{\|x\|} \leq \frac{1 + \gamma\kappa(A)}{1 - \gamma\kappa(A)}, \tag{1.7}$$

$$\frac{\|\Delta x\|}{\|x\|} \leq \frac{2\gamma}{1 - \gamma\kappa(A)}\kappa(A). \tag{1.8}$$

Důkaz. Z (1.2) plyne, že $(I + A^{-1}\Delta A)(x + \Delta x) = x + A^{-1}\Delta b$. Protože

$$\left\|A^{-1}\Delta A\right\| \leq \gamma \|A\| \left\|A^{-1}\right\| \leq \gamma\kappa(A) < 1, \tag{1.9}$$

pak podle lemmatu 1.1 je matice $I + A^{-1}\Delta A$ regulární. Vezmeme-li inverzní matici k této, dostaneme

$$\|x + \Delta x\| \leq \left\|(I + A^{-1}\Delta A)^{-1}\right\| \left(\|x\| + \gamma \left\|A^{-1}\right\| \|b\|\right).$$

Z lemmatu 1.1 pak plyne, že

$$\|x + \Delta x\| \leq \frac{1}{1 - \|A^{-1}\Delta A\|} \left(\|x\| + \gamma \left\|A^{-1}\right\| \|b\|\right).$$

Podle (1.9) a faktu, že $\|b\| \leq \|A\| \|x\|$, je

$$\|x + \Delta x\| \leq \frac{1}{1 - \gamma\kappa(A)} \left(\|x\| + \gamma \left\|A^{-1}\right\| \|A\| \|x\|\right). \tag{1.10}$$

Pro dosažení nerovnosti (1.7) již stačí vydělit (1.10) číslem $\|x\|$.

Dokážeme nyní nerovnost (1.8). Odečtením (1.1) a (1.2) máme

$$A\Delta x = -\Delta A(x + \Delta x) + \Delta b. \tag{1.11}$$

Vynásobíme-li (1.11) zleva maticí A^{-1} a použijeme-li (1.9), dostaneme

$$\|\Delta x\| \leq \left\|A^{-1}\Delta A\right\| \|x + \Delta x\| + \left\|A^{-1}\right\| \|\Delta b\| \leq$$

$$\leq \gamma\kappa(A) \|x + \Delta x\| + \gamma \left\|A^{-1}\right\| \|b\|.$$

Vydělíme-li obě strany prvkem $\|x\|$ a použijeme-li trojúhelníkovou nerovnost $\|x + \Delta x\| \leq \|x\| + \|\Delta x\|$ a fakt, že $\|b\| \leq \|A\| \|x\|$, dostaneme

$$\frac{\|\Delta x\|}{\|x\|} \leq \left(\frac{\|x\| + \|\Delta x\|}{\|x\|} + \frac{\|x\|}{\|x\|}\right) \gamma\kappa(A),$$

neboli

$$\frac{\|\Delta x\|}{\|x\|} \leq 2\gamma\kappa(A) + \frac{\|\Delta x\|}{\|x\|}\gamma\kappa(A).$$

Převedeme-li $\frac{\|\Delta x\|}{\|x\|}\gamma\kappa(A)$ na druhou stranu, máme

$$\frac{\|\Delta x\|}{\|x\|}\left(1-\gamma\kappa(A)\right) \leq 2\gamma\kappa(A).$$

Z toho již plyne nerovnost (1.8). □

V souladu s předpokladem, že $\gamma = \beta^{1-t} = 2\mathbf{u}$, z věty 1.2 plyne, že

$$\frac{\|\Delta x\|}{\|x\|} \approx \mathbf{u}\kappa(A),$$

kde $\mathbf{u}$ je zaokrouhlovací jednotka.

Definice 1.2. *Číslo* $\mathbf{u}\kappa(A)$ *se nazývá nevyhnutelná chyba.*

1.3 Zpětná stabilita

V předchozím odstavci jsme se zabývali citlivostí úlohy. Stabilita je na rozdíl od citlivosti vlastností algoritmu. Při řešení problému pomocí nějaké numerické metody obdržíme místo přesného řešení řešení přibližné. Zajímá nás, zda je toto řešení dostatečně blízko řešení přesnému, tedy zjišťujeme velikost chyby řešení.

Ke zkoumání velikosti chyby jsou dva přístupy - přímá (též dopředná) a zpětná analýza.

Přímá analýza zaokrouhlovacích chyb spočívá v tom, že provedeme odhad chyby postupně, v každé jednotlivé operaci během výpočtu, a dostaneme tak velmi pracně celkový odhad chyby výsledku, který bude značně pesimistický.

V numerických výpočtech oboru lineární algebry se častěji užívá tzv. *zpětná analýza zaokrouhlovacích chyb*. V tomto postupu zahrneme všechny nepřesnosti ve výpočtu pouze do vstupních dat, t.j. hledáme porušení vstupních dat tak, aby vypočtené (nepřesné) řešení bylo přesným řešením problému s těmito porušenými vstupními daty.

Nechť $S : \mathbb{R}^{n\times n} \times \mathbb{R}^n \to \mathbb{R}^n$ je operátor, který dvojici $S(A, b)$ přiřadí aproximaci řešení x soustavy $Ax = b$. Pak S se nazývá metodou řešení problému $Ax = b$, dvojici (A, b) nazýváme daty úlohy.

Definice 1.3. *Metoda se nazývá numericky stabilní, jestliže pro všechny matice* A *a pro všechny vektory pravé strany* b *platí*

$$\frac{\|\Delta x\|}{\|x\|} \approx \mathbf{u}\kappa(A),$$

kde $\mathbf{u}$ *je zaokrouhlovací jednotka.*

Poznámka 1.1. Předchozí definice říká, že metoda se nazývá numericky stabilní, jestliže pro všechny matice A a pro všechny vektory pravé strany b je relativní chyba řešení omezená malým násobkem nevyhnutelné chyby $\mathbf{u}\kappa(A)$.

Definice 1.4. *Metoda S se nazývá zpětně stabilní, jestliže řešení, vypočtené touto numerickou metodou, je přesným řešením soustavy s porušenými daty, t.j. $S(A, b)$ splňuje*

$$(A + \Delta A)S(A, b) = b + \Delta b, \tag{1.12}$$

kde $\|\Delta A\| \leq \mathbf{u}C_S \|A\|$ a $\|\Delta b\| \leq \mathbf{u}C_S \|b\|$. Konstanta C_S se nazývá konstanta stability a může záviset na n.

Zpětná stabilita je důležitou vlastností numerické metody pro řešení soustav lineárních rovnic. Vstupní data problému (A, b) obvykle obsahují chyby (t.j. místo soustavy $Ax = b$ řešíme porušenou soustavu). Zpětná stabilita zajišťuje, že i přes porušená vstupní data obdržíme dobrou aproximaci přesného řešení.

Kapitola 2

Govaerts - Pryce

V této kapitole představíme první z algoritmů pro řešení soustav lineárních rovnic. Začneme jeho jednodušší variantou, totiž metodou pro řešení soustavy s maticí obroubenou pouze vektory. Pomocí této metody pak ukážeme algoritmus pro obecnější případ (širší vroubení).

Na závěr této kapitoly se budeme zabývat stabilitou těchto dvou algoritmů.

Mějme soustavu lineárních rovnic

$$M \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix} \tag{2.1}$$

s blokovou maticí

$$M = \begin{pmatrix} A & B \\ C^T & D \end{pmatrix}, \tag{2.2}$$

kde $A \in \mathbb{R}^{n\times n},\ B, C \in \mathbb{R}^{n\times m},\ D \in \mathbb{R}^{m\times m},\ x, f \in \mathbb{R}^n,\ y, g \in \mathbb{R}^m$.

2.1 Algoritmus BEM

Řešíme nejprve soustavu (2.1) s maticí M obroubenou pouze vektory, t.j. $m = 1$. (B, C jsou tedy vektory a D je skalár, proto je pro lepší přehlednost označíme po řadě b, c, d.) Úlohu lze řešit pomocí dvou různých blokových LU rozkladů matice M, které zachovávají její strukturu.

První se nazývá Doolittlův rozklad:

$$\begin{pmatrix} A & b \\ c^T & d \end{pmatrix} = \begin{pmatrix} I_n & 0 \\ w^T & 1 \end{pmatrix} \begin{pmatrix} A & b \\ 0 & \delta^* \end{pmatrix}, \tag{2.3}$$

kde w^T a δ^* dostaneme ze vztahů

$$c^T = \begin{pmatrix} w^T & 1 \end{pmatrix} \begin{pmatrix} A \\ 0 \end{pmatrix} = w^T A, \quad d = \begin{pmatrix} w^T & 1 \end{pmatrix} \begin{pmatrix} b \\ \delta^* \end{pmatrix} = w^T b + \delta^*.$$

Tedy $c = A^T w$ a $\delta^* = d - w^T b$.
Dále položíme

$$\begin{pmatrix} \gamma_1 \\ \gamma_2 \end{pmatrix} = \begin{pmatrix} A & b \\ 0 & \delta^* \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

a řešíme soustavu

$$\begin{pmatrix} I_n & 0 \\ w^T & 1 \end{pmatrix} \begin{pmatrix} A & b \\ 0 & \delta^* \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} I_n & 0 \\ w^T & 1 \end{pmatrix} \begin{pmatrix} \gamma_1 \\ \gamma_2 \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix} \tag{2.4}$$

(tzv. substituce vpřed). Z (2.4) je $\gamma_1 = f$, $w^T\gamma_1 + \gamma_2 = w^T f + \gamma_2 = g$ a tedy $\gamma_2 = g - w^T f$.
Potom provedeme tzv. zpětnou substituci

$$\begin{pmatrix} A & b \\ 0 & \delta^* \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \gamma_1 \\ \gamma_2 \end{pmatrix} = \begin{pmatrix} f \\ g - w^T f \end{pmatrix}, \tag{2.5}$$

která dává $Ax + by = f$ a $\delta^* y = g - w^T f$. Pomocí těchto dvou rovností spočítáme x a y.

Nyní tyto poznatky můžeme shrnout do algoritmu.

Algoritmus BED (*Block Elimination Doolittle*):

1. $A^T w = c$
2. $\delta^* = d - w^T b$
3. $y = (g - w^T f)/\delta^*$
4. $Ax = f - by$

Druhým blokovým LU rozkladem matice M je tzv. Croutův rozklad:

$$\begin{pmatrix} A & b \\ c^T & d \end{pmatrix} = \begin{pmatrix} A & 0 \\ c^T & \delta \end{pmatrix} \begin{pmatrix} I_n & v \\ 0 & 1 \end{pmatrix} \tag{2.6}$$

Prvky v a δ získáme opět ze vztahů

$$b = \begin{pmatrix} A & 0 \end{pmatrix} \begin{pmatrix} v \\ 1 \end{pmatrix} = Av, \quad d = \begin{pmatrix} c^T & \delta \end{pmatrix} \begin{pmatrix} v \\ 1 \end{pmatrix} = c^T v + \delta.$$

Máme tedy $Av = b$ a $\delta = d - c^T v$.
Tentokrát položíme

$$\begin{pmatrix} \xi \\ y \end{pmatrix} = \begin{pmatrix} I & v \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

a provedeme substituci vpřed

$$\begin{pmatrix} A & 0 \\ c^T & \delta \end{pmatrix} \begin{pmatrix} \xi \\ y \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix}. \tag{2.7}$$

Z (2.7) dostáváme $A\xi = f$, $c^T\xi + \delta y = g$, t.j. $y = (g - c^T\xi)/\delta$.
Zpětná substituce

$$\begin{pmatrix} I & v \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \xi \\ y \end{pmatrix} \tag{2.8}$$

dává $x + vy = \xi$ a tedy $x = \xi - vy$.

Opět výše odvozené shrneme do algoritmu.

Algoritmus BEC (*Block Elimination Crout*):

1. $Av = b$
2. $\delta = d - c^T v$
3. $A\xi = f$
4. $y = (g - c^T\xi)/\delta$
5. $x = \xi - vy$

Předpokládejme, že máme zpětně stabilní metodu pro řešení soustavy s maticí A, resp. s maticí A^T. Jestliže matice A a M jsou dobře podmíněné, algoritmy BED a BEC fungují dobře.

Pokud ale matice A je špatně podmíněná, nabízí se na obdržený výsledek aplikovat algoritmus iteračního zpřesnění.

Algoritmus iteračního zpřesnění

Algoritmus iteračního zpřesnění řešení je založen na jednoduché myšlence. Nechť z_c je řešení soustavy s maticí M spočtené pomocí nějaké numerické metody a z_t nechť je přesné řešení této soustavy. Spočítáme tzv. vektor rezidua

$$r = h - Mz_c,$$

kde $h = (f^T, g^T)^T$ je pravá strana soustavy (2.1). Označme w řešení soustavy $My = r$. Pokud je toto řešení přesné, pak zřejmě

$$M(z_c + w) = h \quad a \quad z_t = z_c + w.$$

Je jasné, že w přesně nezískáme, ale tato myšlenka dává základ algoritmu iteračního zpřesnění řešení.

Algoritmus iteračního zpřesnění

1. $z = z_c$
2. $r = h - Mz$
3. w se spočte jako řešení soustavy $My = r$ užitím numerické metody

4. $z^* = z + w$
5. $r^* = h - Mz^*$
6. je-li $\|r^*\|_2 < \|r\|_2$, jdi na 7, jinak jdi na 8
7. $r = r^*, z = z^*$, jdi na 3
8. konec, z je hledané zpřesnění řešení.

Aplikujeme nyní iterační zpřesnění řešení přímo na algoritmus BED, resp. BEC.

Nechť $(x_1^T, y_1^T)^T$ je numerické řešení problému (2.1) s pravou stranou $(f^T, g^T)^T$, vypočtené pomocí algoritmu BED. Definujme algoritmus BED+k.

Algoritmus BED+k

Pro $i = 1, 2, ..., k$

1. spočítáme rezidua $f_1 = f - Ax_1 - by_1,\ g_1 = g - cx_1 - dy_1$
2. spočítáme x_2, y_2 pomocí algoritmu BED (s pravou stranou $h_1 = (f_1^T, g_1^T)^T$)
3. $x_1 = x_1 + x_2, y_1 = y_1 + y_2$

Analogicky definujeme také algoritmus BEC+k.

Numerické testy provedené v [3] ukazují, že použijeme-li algoritmus BED, je y spočteno správně bez jakéhokoliv iteračního zpřesnění. Abychom ale získali dostatečně přesně i složku x, je nutné aplikovat algoritmus BED+k pro $k > 1$. Ukážeme proč.

Pro jednoduchost předpokládejme, že chyby vznikají pouze při řešení soustav s maticemi A a A^T, t.j. v krocích 1 a 4 algoritmu BED. Zanedbáme tedy zaokrouhlovací chyby při ostatních výpočtech. Označíme-li $\overline{w}$, $\overline{\delta^*}$, $\overline{y}$ jako spočtené hodnoty, máme

$$(A + \Delta A)^T \overline{w} = c, \quad \overline{\delta^*} = d - \overline{w}^T b, \quad \overline{\delta^*}\overline{y} = g - \overline{w}^T f. \tag{2.9}$$

Pro matici ΔA platí $\|\Delta A\| \leq \mathbf{u} C_S \|A\|$, kde $\mathbf{u}$ je zaokrouhlovací jednotka a C_S je konstanta stability metody pro řešení soustavy s maticí A, resp. A^T (tu podle předpokladu máme k dispozici). Označme dále x_1 (neznámé) přesné řešení soustavy

$$(A + \Delta A)\, x_1 = f - b\overline{y}. \tag{2.10}$$

Z (2.9) a (2.10) dostáváme

$$\begin{pmatrix} A + \Delta A & b \\ c^T & d \end{pmatrix} \begin{pmatrix} x_1 \\ \overline{y} \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix}. \tag{2.11}$$

Protože (2.11) představuje malou změnu dobře podmíněného (původního) systému (2.1) s maticí M, musí $\overline{y}$ být dobrá aproximace y.

Dále numerické testy ukazují, že použijeme-li algoritmus BEC, je nutné pro dostatečně přesný výpočet obou složek řešení provést algoritmus BEC+ k pro $k > 1$. Ukazuje se ale také, že relativní chyba x, spočteného pomocí algoritmu BEC+k+1, je řádu relativní chyby y, spočteného pomocí algoritmu BEC+k. To tedy znamená, že pokud y je řádu $\mathbf{u}\left\|(x^T\ y^T)^T\right\|$, pak algoritmus BEC počítá x správně, t.j. s chybou řádu $\mathbf{u}\kappa(M)\left\|(x^T\ y^T)^T\right\|$ ($\mathbf{u}\kappa(M)$ je nevyhnutelná chyba). Opět ukážeme, proč tomu tak je.

Opět budeme předpokládat, že chyby vznikají pouze při řešení soustav s maticí A a A^T (tedy v krocích 1 a 3 algoritmu BEC). Nechť $\overline{v},\ \overline{\delta},\ \overline{\xi},\ \overline{x},\ \overline{y}$ označují vypočtené hodnoty. Máme

$$(A + \Delta_1 A)\,\overline{v} = b, \tag{2.12}$$

$$\overline{\delta} = d - c^T\overline{v}, \tag{2.13}$$

$$(A + \Delta_2 A)\,\overline{\xi} = f, \tag{2.14}$$

$$\overline{\delta}\overline{y} = g - c^T\overline{\xi}, \tag{2.15}$$

$$\overline{x} = \overline{\xi} - \overline{v}\overline{y}, \tag{2.16}$$

kde $\|\Delta_1 A\| \le \mathbf{u}C_S\,\|A\|\,,\ \|\Delta_2 A\| \le \mathbf{u}C_S\,\|A\|$.

Pomocí (2.16) máme

$$A\overline{x} = A\overline{\xi} - A\overline{v}\overline{y},$$

pomocí (2.12) je

$$b\overline{y} = A\overline{v}\overline{y} + \Delta_1 A\overline{v}\overline{y},$$

tedy

$$A\overline{x} + b\overline{y} = A\overline{\xi} + \Delta_1 A\overline{v}\overline{y}. \tag{2.17}$$

Dosadíme-li ještě do (2.17) za $A\overline{\xi}$ z (2.14), dostáváme

$$A\overline{x} + b\overline{y} = f - \Delta_2 A\overline{\xi} + \Delta_1 A\overline{v}\overline{y}. \tag{2.18}$$

Podobně z (2.16) dostaneme

$$c^T\overline{x} = c^T\overline{\xi} - c^T\overline{v}\overline{y}$$

a z (2.13) dostaneme

$$d\overline{y} = \overline{\delta}\overline{y} + c^T\overline{v}\overline{y}.$$

Použijeme-li ještě (2.15), máme

$$c^T\overline{x} + d\overline{y} = c^T\overline{\xi} + g - c^T\overline{\xi} = g. \tag{2.19}$$

Zapíšeme-li rovnosti(2.18) a (2.19) maticově, máme

$$\begin{pmatrix} A & b \\ c^T & d \end{pmatrix} \begin{pmatrix} \overline{x} \\ \overline{y} \end{pmatrix} = \begin{pmatrix} f - (\Delta_2 A)\overline{\xi} + (\Delta_1 A)\overline{vy} \\ g \end{pmatrix}. \tag{2.20}$$

Nyní potřebujeme odhadnout $\left\|-(\Delta_2 A)\overline{\xi} + (\Delta_1 A)\overline{vy}\right\|$ pomocí $\|f\|$ a $\|g\|$. V článku [3] je ukázáno, že

$$\left\|\Delta_2 A\overline{\xi}\right\| \leq \mathbf{u} C_S \|f\|, \quad \|\Delta_1 A\overline{vy}\| \leq \mathbf{u} C_S \|g\|, \tag{2.21}$$

tedy

$$\left\|\Delta_2 A\overline{\xi}\right\| \approx \mathbf{u} \|f\|, \quad \|\Delta_1 A\overline{vy}\| \approx \mathbf{u} \|g\|.$$

Máme tedy

$$\left\| \begin{pmatrix} -(\Delta_2 A)\overline{\xi} + (\Delta_1 A)\overline{vy} \\ 0 \end{pmatrix} \right\| \leq \mathbf{u} C_S \left\| \begin{pmatrix} f \\ g \end{pmatrix} \right\|, \tag{2.22}$$

kde C_S je konstanta stability metody pro řešení soustavy s maticí A.

Protože platí

$$\begin{pmatrix} A & b \\ c^T & d \end{pmatrix} \begin{pmatrix} \overline{x} \\ \overline{y} \end{pmatrix} = \begin{pmatrix} f - (\Delta_2 A)\overline{\xi} + (\Delta_1 A)\overline{vy} \\ g \end{pmatrix},$$

kde

$$\left\| \begin{pmatrix} -(\Delta_2 A)\overline{\xi} + (\Delta_1 A)\overline{vy} \\ 0 \end{pmatrix} \right\| \approx \mathbf{u} \left\| \begin{pmatrix} f \\ g \end{pmatrix} \right\|,$$

je $(\overline{x}^T\ \overline{y}^T)^T$ aproximace řešení $(x^T\ y^T)^T$ s relativní chybou řádu $\mathbf{u}\kappa(M)$.

Vzhledem k těmto dvěma faktům je přirozené spočítat y pomocí algoritmu BED a potom udělat krok iteračního zpřesnění pomocí algoritmu BEC, kde jako počáteční přiblížení použijeme správně vypočtenou hodnotu y a počáteční hodnotu vektoru x položíme rovnu nulovému vektoru.

Na základě této myšlenky vznikl algoritmus BEM.

Algoritmus BEM (*Block Elimination Mixed*):

1. $A^T w = c$
2. $\delta^* = d - w^T b$
3. $Av = b$
4. $\delta = d - c^T v$
5. $y_1 = (g - w^T f)/\delta^*$
6. $f_1 = f - by_1$

7. $g_1 = g - dy_1$
8. $A\xi = f_1$
9. $y_2 = (g_1 - c^T\xi)/\delta$
10. $x = \xi - vy_2$
11. $y = y_1 + y_2$

V tomto algoritmu se nejprve v krocích 1 - 2 provede Doolittlův LU rozklad a v krocích 3 - 4 Croutův LU rozklad matice M. V kroku 5 se pomocí zpětné substituce vypočítá y_1. Kroky 6 a 7 představují výpočet reziduí. V krocích 8, 9 a 10 se pomocí Croutova rozkladu řeší soustava

$$\begin{pmatrix} A & b \\ c^T & d \end{pmatrix} \begin{pmatrix} x + 0 \\ y_2 + y_1 \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix}.$$

Jedná se vlastně o krok iteračního zpřesnění, kde pravá strana je nahrazena vektory f_1, g_1 a jako počáteční přiblížení se uvažuje vypočtené y_1 (z kroku 5) a $x = 0$. Krok 11 je už jen aktualizace vektoru y.

Metoda BEM dává správné výsledky, pokud matice M je dobře podmíněná, i když matice A je téměř singulární. Je možné uměle zkonstruovat takový případ, kdy tato metoda selže, ale v praxi se toto stává velmi zřídka.

Metoda BEM je zpětně stabilní pro $m = 1$. Toto tvrzení bude podrobněji zdůvodněno ve třetí kapitole.

Na závěr tohoto odstavce ještě uvedeme poznámku, která se ukáže jako velmi užitečná pro případ $m > 1$.

Poznámka 2.1. Pokud v krocích 1 - 4 zaměníme v a w a dále δ a δ^*, dostaneme metodu BEM pro soustavu s transponovanou maticí M^T.

2.2 Bloková alternativa algoritmu BEM

Použijeme-li algoritmus BEM i pro blokové vroubení (t.j. pro $m > 1$), pak δ a δ^* jsou nahrazeny malými maticemi typu $(m \times m)$.

Definice 2.1. *Nechť A, B, C, D jsou matice o rozměrech $p \times p$, $p \times q$, $p \times q$, $q \times q$ a nechť A je invertibilní. Nechť*

$$M = \begin{pmatrix} A & B \\ C^T & D \end{pmatrix}.$$

Pak Schurův doplněk matice A v matici M je matice $D - C^T A^{-1} B$ typu $q \times q$.

Nahradíme-li v algoritmu vektory b, c, d, v, w po řadě maticemi B, C, D, V, W příslušných rozměrů, dostaneme $\delta = D - C^T V$, resp. $\delta^* = D - WB$. Protože $AV = B$ a $A^T W = C$, máme

$$\delta = \delta^* = D - C^T A^{-1} B. \tag{2.23}$$

Tedy podle definice 2.1 jak δ tak δ^* jsou vlastně Schurovým doplňkem matice A v matici M. Kdybychom použili algoritmus BEM, museli bychom kromě řešení soustavy s maticí A a A^T řešit v krocích 5 a 9 také soustavu s maticí δ, resp. δ^*.

V tomto odstavci ukážeme, proč takováto bloková metoda BEM může selhat. K tomu budeme potřebovat definici singulárních čísel a několik jejich vlastností.

Definice 2.2. *Nechť $A \in \mathbb{R}^{m\times n}$. Pak $A = USV^T$, kde U a V jsou ortogonální matice typu $m \times m$, resp. $n \times n$, a $S \in \mathbb{R}^{m\times n}$ je diagonální matice tvaru*

$$S = \begin{pmatrix} \Sigma \\ 0 \end{pmatrix} \text{ jestliže } m \geq n, \qquad \text{nebo } S = \begin{pmatrix} \Sigma & 0 \end{pmatrix} \text{ jestliže } m < n,$$

kde

$$\Sigma = \begin{pmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_{min(m,n)} \end{pmatrix}. \tag{2.24}$$

Čísla $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_{min(m,n)} \geq 0$ se nazývají singulární čisla matice A.

Pro každou matici $A \in \mathbb{R}^{m\times n}$ položíme $\sigma_i = 0$ pro $i > min(m, n)$.

Lemma 2.1. *1. Nechť $\lambda_i(X)$ jsou vlastní čísla pozitivně semidefinitní matice $X \in \mathbb{R}^{k\times k}$ taková, že $\lambda_1(X) \geq \ldots \geq \lambda_k(X)$. Pak*

$$\sigma_i(A)^2 = \lambda_i(A^T A) = \lambda_i(AA^T).$$

2. $\sigma_i(A) = \sigma_i(A^T)$ pro všechna i.

3. Pro každé i je $\sigma_i(A) = min\left\{\left\|A - \tilde{A}\right\| : \text{rank}(\tilde{A}) = i\right\}$.

4. Singulární čísla matice se nemění vynásobením matice unitární maticí, permutací řádků nebo sloupců, změnou znaménka řádků nebo sloupců, ani přidáním nulového řádku či sloupce.

5. Jestliže B je podmatice matice A, pak $\sigma_i(B) \leq \sigma_i(A)$ pro všechna i.

6. Jestliže $B = XAY$, pak $\sigma_i(B) \leq \|X\| \, \|Y\| \, \sigma_i(A)$ pro všechna i.

7. Nechť $A \in \mathbb{R}^{m\times n}$ a $B = \begin{pmatrix} I_n \\ A \end{pmatrix}$. Pak $\sigma_i(B)^2 = 1 + \sigma_i(A)^2$ pro $1 \leq i \leq n$.

Důkaz. Důkaz lze nalézt v [2], str. 53. □

Věta 2.1. *Nechť*

$$M = \begin{pmatrix} A & B \\ C^T & D \end{pmatrix}$$

je regulární bloková matice, kde $A, B, C, D \in \mathbb{R}^{n\times n}$. *Nechť její inverze je*

$$M^{-1} = \begin{pmatrix} P & Q \\ R^T & S \end{pmatrix}.$$

Pak

$$\|M\|^{-2}\,\sigma_i(A) \le \sigma_i(S) \le \left\|M^{-1}\right\|^2 \sigma_i(A) \tag{2.25}$$

pro každé i.

Důkaz. Nejprve dokážeme druhou nerovnost. Bez újmy na obecnosti předpokládejme, že matice A je regulární. (To je možné, vzhledem k tomu, že všechny členy v nerovnosti (2.25) jsou spojité funkce A a libovolně malé změny v singulární matici z ní učiní matici regulární.)

Položme $A^{-1}B = V$. Pak

$$\begin{pmatrix} A^{-1} & 0 \end{pmatrix} M = \begin{pmatrix} I_n & V \end{pmatrix}. \tag{2.26}$$

Z (2.26) a lemmatu 2.1 (body 5, 6) máme

$$\sigma_i(A^{-1}) \le \left\|M^{-1}\right\| \sigma_i \begin{pmatrix} I_n & V \end{pmatrix} \quad \forall i. \tag{2.27}$$

$$\begin{aligned} \left(\sigma_i \begin{pmatrix} I_n & V \end{pmatrix}\right)^2 &= \left(\sigma_i \begin{pmatrix} I_n \\ V^T \end{pmatrix}\right)^2 \qquad [\textit{lemma } 2.1 \ (\textit{bod } 2)] \\ &= 1 + (\sigma_i(V))^2 \qquad [\textit{lemma } 2.1 \ (\textit{body } 2, 7)] \\ &= \left(\sigma_i \begin{pmatrix} -V \\ I_n \end{pmatrix}\right)^2, \qquad [\textit{lemma } 2.1 \ (\textit{body } 4, 7)] \end{aligned}$$

čili

$$\sigma_i \begin{pmatrix} I_n & V \end{pmatrix} = \sigma_i \begin{pmatrix} -V \\ I_n \end{pmatrix} \quad \forall i. \tag{2.28}$$

Na druhé straně je

$$M \begin{pmatrix} -V \\ I_n \end{pmatrix} = \begin{pmatrix} 0 \\ S^{-1} \end{pmatrix}. \tag{2.29}$$

Pomocí lemmatu 2.1 (body 4, 6) dostáváme

$$\sigma_i \begin{pmatrix} -V \\ I_n \end{pmatrix} \le \left\|M^{-1}\right\| \sigma_i(S^{-1}). \tag{2.30}$$

Spojíme-li (2.27), (2.29), (2.30), dostaneme

$$\sigma_i(A^{-1}) \leq \left\|M^{-1}\right\|^2 \sigma_i(S^{-1}). \tag{2.31}$$

Použitím nerovnosti (2.31) a skutečnosti, že singulární čísla inverzní matice k matici typu $(k \times k)$ jsou pro $i \leq k$ převrácené hodnoty singuláních čísel původní matice, v obráceném pořadí, dostáváme druhou nerovnost tvrzení věty.

První nerovnost dostaneme obdobným postupem s tím rozdílem, že v rovnostech (2.26) a (2.29) ponecháme matici M na levé straně. □

Věta 2.2. *Nechť*

$$M = \begin{pmatrix} A & B \\ C^T & D \end{pmatrix}$$

je regulární bloková matice typu $(n+m) \times (n+m)$, *kde* $A \in \mathbb{R}^{n\times n}$, $B, C \in \mathbb{R}^{n\times m}$, $D \in \mathbb{R}^{m\times m}$. *Nechť její inverze je*

$$M^{-1} = \begin{pmatrix} P & Q \\ R^T & S \end{pmatrix}.$$

Položme $p = min(n, m)$. *Pak*

$$\|M\|^{-2} \sigma_{n-j}(A) \leq \sigma_{m-j}(S) \leq \left\|M^{-1}\right\|^2 \sigma_{n-j}(A) \quad pro\ 0 \leq j < p. \tag{2.32}$$

Důkaz. Nechť $m < n$. Označme

$$\tilde{M} = \begin{pmatrix} A & B & 0 \\ C^T & D & 0 \\ 0 & 0 & \mu I_{n-m,n-m} \end{pmatrix},$$

kde $\mu = \|M\|$. Pak

$$\tilde{M}^{-1} = \begin{pmatrix} P & Q & 0 \\ R^T & S & 0 \\ 0 & 0 & \mu^{-1} I_{n-m,n-m} \end{pmatrix}.$$

Na matici $\tilde{M}$ lze pohlížet jako na matici o čtyřech blocích o velikosti $n \times n$, lze tedy aplikovat větu 2.1. Dostáváme

$$\left\|\tilde{M}\right\|^{-2} \sigma_i(A) \leq \sigma_i \begin{pmatrix} S & 0 \\ 0 & \mu^{-1} I_{n-m,n-m} \end{pmatrix} \leq \left\|\tilde{M}^{-1}\right\|^2 \sigma_i(A). \tag{2.33}$$

Podle lemmatu 2.1 bod 4 je

$$\sigma_i \begin{pmatrix} S & 0 \\ 0 & \mu^{-1} I_{n-m,n-m} \end{pmatrix} = \sigma_i \begin{pmatrix} \mu^{-1} I_{n-m,n-m} & 0 \\ 0 & S \end{pmatrix}.$$

Platí

$$\sigma_i \begin{pmatrix} \mu^{-1} I_{n-m,n-m} & 0 \\ 0 & S \end{pmatrix} = \|M\|^{-1} \quad pro\ 1 \le i \le n-m \tag{2.34}$$

$$= \sigma_{m-n+i}(S) \quad pro\ n-m < i \le n. \tag{2.35}$$

Dále platí

$$\left\|\tilde{M}\right\| = max\left(\mu, \|M\|\right) = \|M\|, \tag{2.36}$$

$$\left\|\tilde{M}^{-1}\right\| = max\left(\mu^{-1}, \left\|M^{-1}\right\|\right) = \left\|M^{-1}\right\|. \tag{2.37}$$

Kombinací (2.35), (2.36) a (2.37) dostáváme

$$\|M\|^{-2} \sigma_i(A) \le \sigma_{m-n+i}(S) \le \left\|M^{-1}\right\|^2 \sigma_i(A)$$

pro $n-m < i \le n$. Položíme-li $j = n-i$, dostaneme tvrzení věty. □

Nás bude zajímat podmíněnost matice δ (resp. δ^*), t.j. Schurova doplňku matice A v matici M. Tím ale není nic jiného než matice S^{-1}.

Pro číslo podmíněnosti matice S platí $\kappa(S) = \|S\|\,\|S^{-1}\|$. Uvažujme spektrální normu $\|S\| = \sqrt{\rho(S^T S)}$. Máme tedy

$$\kappa(S^{-1}) = \kappa(S) = \|S\| \left\|S^{-1}\right\| = \sqrt{\rho(S^T S)}\sqrt{\rho((S^T S)^{-1})} =$$

$$= \sqrt{\lambda_{max}(S^T S)}\sqrt{\lambda_{max}((S^T S)^{-1})} = \frac{\sqrt{\lambda_{max}(S^T S)}}{\sqrt{\lambda_{min}(S^T S)}},$$

kde $\lambda_{max}(S^T S)$, resp. $\lambda_{min}(S^T S)$ značí největší, resp. nejmenší vlastní číslo matice $S^T S$.

Podle lemmatu 2.1 (bod 1) je tedy

$$\kappa(S^{-1}) = \kappa(S) = \frac{\sigma_{max}(S)}{\sigma_{min}(S)} = \frac{\sigma_1(S)}{\sigma_m(S)}. \tag{2.38}$$

Aplikací Věty 2.2 pro $j = 0$ a $j = m-1$ na naši matici M dostaneme

$$\|M\|^{-2} \sigma_n(A) \le \sigma_m(S) \le \left\|M^{-1}\right\|^2 \sigma_n(A) \tag{2.39}$$

$$\|M\|^{-2} \sigma_{n-m+1}(A) \le \sigma_1(S) \le \left\|M^{-1}\right\|^2 \sigma_{n-m+1}(A), \tag{2.40}$$

čili

$$\frac{\|M\|^{-2} \sigma_{n-m+1}(A)}{\left\|M^{-1}\right\|^2 \sigma_n(A)} \le \frac{\sigma_1(S)}{\sigma_m(S)} \le \frac{\left\|M^{-1}\right\|^2 \sigma_{n-m+1}(A)}{\|M\|^{-2} \sigma_n(A)}. \tag{2.41}$$

Píšeme-li δ místo S^{-1}, jednoduchými úpravami a použitím (2.38) nakonec dostaneme

$$\frac{1}{(\kappa(M))^2} \frac{\sigma_{n-m+1}(A)}{\sigma_n(A)} \le \kappa(\delta) \le \frac{\sigma_{n-m+1}(A)}{\sigma_n(A)} (\kappa(M))^2. \tag{2.42}$$

Pokud tedy M je dobře podmíněná, t.j. v nejlepším případě $\kappa(M) = 1$, vidíme, že podmíněnost matice $\delta = \delta^*$ je určena m nejmenšími singulárními čísly matice A. Je-li matice A špatně podmíněná (a to podle předpokladu je), pak protože

$$\kappa(A) = \frac{\sigma_1(A)}{\sigma_n(A)} \geq \frac{\sigma_{n-m+1}(A)}{\sigma_n(A)}, \tag{2.43}$$

může i matice δ být špatně podmíněná. Přesněji, pro malá m může být odhad (2.43) dobrý, naopak pro velká m a špatně podmíněnou matici A dostaneme velkou horní mez a tudíž bloková varianta algoritmu BEM by mohla selhat.

2.3 Algoritmus BEMW

V předchozím odstavci jsme ukázali, proč nelze použít algoritmus BEM pro širší vroubení. Algoritmus BEMW je koncipován jinak - rekurzivně. Matice B, C, D, jsou 'rozkouskovány' do několika vektorů $b_1,\ c_1 \in \mathbb{R}^n, \ldots, b_m,\ c_m \in \mathbb{R}^{n+m-1}$ a skalárů $d_{11}, \ldots, d_{mm}$. Obrázek ukazuje příklad pro $m = 3$.

$$\left(\begin{array}{cc|c|c} A & b_1 & b_2 & \\ \hline c_1 & d_{11} & & b_3 \\ \hline c_2 & & d_{22} & \\ \hline \multicolumn{2}{c|}{c_3} & & d_{33} \end{array}\right) \tag{2.44}$$

Jak jsme již zmínili v prvním odstavci této kapitoly, zaměníme-li v algoritmu BEM v za w a δ za δ^*, dostaneme algoritmus pro řešení soustavy s maticí M^T, který je stejně stabilní jako algoritmus původní. Označme algoritmus BEM pro soustavu s maticí M

$$S_1 = BEM(b, c, d, S, S^T) \tag{2.45}$$

a algoritmus BEM pro soustavu s maticí M^T označme

$$S_1^T = BEM(c, b, d, S^T, S), \tag{2.46}$$

kde S, resp. S^T označuje zpětně stabilní řešič pro soustavu s maticí A, resp. A^T.

Označme A_k $(0 \leq k \leq m)$ podmatici matice M řádu $(n + k)$ takovou, že

$$A_0 = A, \quad A_m = M$$

a

$$A_k = \begin{pmatrix} A_{k-1} & b_k \\ c_k^T & d_k \end{pmatrix}. \tag{2.47}$$

Vektory b_k a c_k jsou sloupcové vektory o délce $(n + k - 1)$ a d_k je skalár.

Algoritmus BEMW se skládá z rekurzivní konstrukce stabilních řešičů pro matice A_k a A_k^T.

Algoritmus BEMW (*Block Elimination Mixed for Wider Borders*):

Vstup: S_0, S_0^T (zpětně stabilní řešiče pro matice A_0, A_0^T)

1. pro $k = 1, 2, \ldots, m$
2. $S_k = BEM(b_k, c_k, d_k, S_{k-1}, S_{k-1}^T)$
3. $S_k^T = BEM(c_k, b_k, d_k, S_{k-1}^T, S_{k-1})$

2.4 Zpětná stabilita algoritmů BEM a BEMW

V tomto odstavci nejprve ukážeme stabilitu algoritmu BEM. Hlavní myšlenkou důkazu stability algoritmu BEMW je potom fakt, že se příslušné vlastnosti přenášejí od jednoho dílčího algoritmu k druhému.

Předtím, než uvedeme několik tvrzení, která ukazují zpětnou stabilitu algoritmu BEM, zavedeme označení, které budeme v tomto odstavci používat.

Symbolem $\overline{a}$ budeme značit vypočtenou hodnotu a. Číslo $\mathbf{u}$ je opět zaokrouhlovací jednotka. Symboly $\theta_1, \theta_2, \ldots$ jsou stejné jako v kapitole 1.

Připomeňme ještě, že

$$fl(x\ op\ y) = \theta(x\ op\ y), \quad \theta \in 1(\mathbf{u}),$$

a že existuje konstanta C_{IP} taková, že

$$fl(x^T y) = x^T \theta y, \quad \theta \in (C_{IP}\mathbf{u}),$$

kde θ je diagonální matice, $x, y \in \mathbb{R}^n$ a $C_{IP} \leq n$.

V důkazech budeme používat tyto nerovnosti:

$$\left\|x^\theta\right\| \leq e^{\|\theta\|}, \quad \left\|e^\theta - I\right\| \leq \|\theta\|\, e^{\|\theta\|}.$$

Věta 2.3. *Nechť S je zpětně stabilní metoda pro řešení soustavy s maticí A^T s konstantou stability C_S. Nechť $\overline{y}$ je vypočtené v kroku 3 algoritmu BEM. Pak $\overline{y}$ je y-ová komponenta přesného řešení soustavy (2.1). Jinými slovy, existují ΔA, Δb, Δc, Δd, Δf, Δg a x_∞ tak, že*

$$\begin{pmatrix} A + \Delta A & b + \Delta b \\ (c + \Delta c)^T & d + \Delta d \end{pmatrix} \begin{pmatrix} x_\infty \\ \overline{y} \end{pmatrix} = \begin{pmatrix} f + \Delta f \\ g + \Delta g \end{pmatrix} \tag{2.48}$$

a

$$b + \Delta b = \theta_b b, \quad \theta_b \in 1((1 + C_{IP})\mathbf{u}),$$
$$d + \Delta d = \theta_d d, \quad \theta_d \in 1(\mathbf{u}),$$
$$f + \Delta f = \theta_f f, \quad \theta_f \in 1((2 + C_{IP})\mathbf{u}),$$

$$g + \Delta g = \theta_g g, \quad \theta_g \in 1(2\mathbf{u}),$$
$$\|\Delta A\| \leq C_S \mathbf{u} \|A\|,$$
$$\|\Delta c\| \leq C_S \mathbf{u} \|c\|.$$

Důkaz. Máme

$$(A + \Delta A)^T \overline{w} = c + \Delta c, \quad \|\Delta A\| \leq C_S \mathbf{u} \|A\|, \quad \|\Delta c\| \leq C_S \mathbf{u} \|c\|, \tag{2.49}$$

$$\theta_1 \overline{\delta}^* = d - \overline{w}^T \theta_2 b, \quad \theta_1 \in 1(\mathbf{u}), \quad \theta_2 \in 1(C_{IP}\mathbf{u}), \tag{2.50}$$

$$\theta_3 \overline{(g - w^T f)} = g - \overline{w}^T \theta_4 f, \quad \theta_3 \in 1(\mathbf{u}), \quad \theta_4 \in 1(C_{IP}\mathbf{u}), \tag{2.51}$$

$$\theta_5 \overline{y} = \overline{(g - w^T f)}/\delta^*, \quad \theta_5 \in 1(\mathbf{u}). \tag{2.52}$$

Pomocí (2.50), (2.51) a (2.52) dostaneme

$$\overline{y} = \frac{\theta_5^{-1}\theta_3^{-1} g - \overline{w}^T \theta_5^{-1}\theta_3^{-1}\theta_4 f}{\theta_1^{-1} d - \overline{w}^T \theta_1^{-1}\theta_2 b}. \tag{2.53}$$

Tedy $\overline{y}$ je přesná y-ová komponenta řešení soustavy

$$\begin{pmatrix} A + \Delta A & \theta_1^{-1}\theta_2 b \\ (c + \Delta c)^T & \theta_1^{-1} d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \theta_5^{-1}\theta_3^{-1}\theta_4 f \\ \theta_5^{-1}\theta_3^{-1} g \end{pmatrix}, \tag{2.54}$$

z čehož plyne tvrzení věty. □

Věta 2.4. *Nechť S je zpětně stabilní metoda pro řešení soustavy s maticí A s konstantou stability C_S a nechť $\overline{z} = (\overline{x}^T, \overline{y}^T)^T$ je řešení soustavy (2.1) vypočtené pomocí algoritmu BEC. Pak $\overline{x}, \overline{y}$ splňují rovnost*

$$\begin{pmatrix} A + \Delta A & b + \Delta b \\ (c + \Delta c)^T & d + \Delta d \end{pmatrix} \begin{pmatrix} \overline{x} \\ \overline{y} \end{pmatrix} = \begin{pmatrix} f + \Delta f \\ g \end{pmatrix} + \begin{pmatrix} T \\ U \end{pmatrix} \overline{\xi}, \tag{2.55}$$

kde

$$\|\Delta A\| \leq (2 + C_S)\mathbf{u}\ exp(2u) \|A\|,$$
$$\|\Delta b\| \leq C_S \mathbf{u} \|b\|,$$
$$\|\Delta c\| \leq (5 + C_{IP})\mathbf{u}\ exp((5 + C_{IP})u) \|c\|,$$
$$\|\Delta d\| \leq 3\mathbf{u}\ exp(3\mathbf{u}) \|d\|,$$
$$\|\Delta f\| \leq C_S \mathbf{u} \|f\|,$$
$$\|T\| \leq (2C_S + (1 + C_S\mathbf{u})\mathbf{u}\ exp(\mathbf{u}))\mathbf{u} \|A\|,$$
$$\|U\| \leq (4 + 2C_{IP})\mathbf{u}\ exp((6 + C_{IP})\mathbf{u}) \|c\|.$$

Důkaz. Vypočtené hodnoty $\overline{v}, \overline{\xi}, \overline{y}, \overline{x}$ splňují

$$(A + \Delta_v A)\overline{v} = b + \Delta b, \quad \|\Delta_v A\| \leq C_S \mathbf{u} \|A\|, \quad \|\Delta b\| \leq C_S \mathbf{u} \|b\|, \tag{2.56}$$

$$(A + \Delta_\xi A)\overline{\xi} = f + \Delta f, \quad \|\Delta_\xi A\| \leq C_S \mathbf{u} \|A\|, \quad \|\Delta f\| \leq C_S \mathbf{u} \|f\|, \tag{2.57}$$

$$\theta_6 \overline{y} = \frac{g - c^T \theta_7 \overline{\xi}}{d - c^T \theta_8 \overline{v}}, \quad \theta_6 \in 1(3\mathbf{u}), \quad \theta_7 \in 1(C_{IP}\mathbf{u}), \quad \theta_8 \in 1(C_{IP}\mathbf{u}), \tag{2.58}$$

$$\theta_9 \overline{x} = \overline{\xi} - \theta_{10} \overline{vy}, \quad \theta_9 \in 1(\mathbf{u}), \quad \theta_{10} \in 1(\mathbf{u}). \tag{2.59}$$

Dosazením za $\overline{vy}$ v (2.58) pomocí (2.59) dostaneme

$$\theta_6 d\overline{y} + \theta_6 c \theta_8 \theta_{10}^{-1} \theta_9 \overline{x} = g + c^T(\theta_6 \theta_8 \theta_{10}^{-1} - \theta_7)\overline{\xi}. \tag{2.60}$$

Spojením rovností (2.59), (2.56) a (2.57) dostaneme

$$(A + \Delta_v A)\theta_{10}^{-1}\theta_9 \overline{x} + (b + \Delta b)\overline{y} = f + \Delta f + \left[(\Delta_v A - \Delta_\xi A) + (A + \Delta_v A)(\theta_{10}^{-1} - I)\right]\overline{\xi}. \tag{2.61}$$

Nyní můžeme (2.60) a (2.61) přepsat jako

$$\begin{pmatrix} A + \Delta A & b + \Delta b \\ (c + \Delta c)^T & d + \Delta d \end{pmatrix} \begin{pmatrix} \overline{x} \\ \overline{y} \end{pmatrix} = \begin{pmatrix} f + \Delta f \\ g \end{pmatrix} + \begin{pmatrix} T \\ U \end{pmatrix} \overline{\xi}, \tag{2.62}$$

kde meze pro $\|\Delta A\|, \|\Delta b\|, \|\Delta c\|, \|\Delta d\|, \|\Delta f\|, \|T\|$ a $\|U\|$ lze spočítat pomocí mezí v (2.56) - (2.59). □

Věta 2.3 představuje zpětnou analýzu kroků 1 - 3 v algoritmu BEM. Důležitým výsledkem je, že y spočtené v kroku 3 je správné, i když matice A je velmi špatně podmíněná.

Věta 2.4 je zpětnou analýzou algoritmu BEC.

Podle věty 2.4 je

$$(M + \Delta M)\begin{pmatrix} \overline{x} \\ \overline{y} \end{pmatrix} = \begin{pmatrix} f + \Delta f \\ g \end{pmatrix} + \begin{pmatrix} T \\ U \end{pmatrix} \overline{\xi},$$

kde, zanedbáme-li členy vyšších řádů,

$$\|\Delta M\| \leq \mathbf{u} C_2 \|M\|, \quad C_2 = C_S + C_{IP} + 10, \tag{2.63}$$

$$\|T\| \leq \mathbf{u} C_3 \|A\|, \quad C_3 = 2C_S + 1, \tag{2.64}$$

$$\|U\| \leq \mathbf{u} C_4 \|c\|, \quad C_4 = 2C_{IP} + 4. \tag{2.65}$$

Lze ukázat (viz [3]), že algoritmus BEM je zpětně stabilní a že pro konstantu stability platí

$$C_{ST} = C_S + C_{IP} + 20 + (2C_S + 2C_{IP} + 5)(1 + C_P(2C_S + 2C_{IP} + 5)), \tag{2.66}$$

kde C_P je konstanta omezující $\mathbf{u}\,\|(A+E)^{-1}\|\,\|M\|$ pro všechny poruchy E matice A. V rovnosti (2.66) jsou členy řádu $\mathbf{u}\kappa(A)$ zanedbány.

Nyní si ukážeme, jak je to se stabilitou algoritmu BEMW v závislosti na rostoucím m.

V souladu se značením používaném v odstavci 2.3 je S_1 algoritmus pro řešení soustavy s maticí A_1. Z předchozího plyne, že tento algoritmus je zpětně stabilní s konstantou stability $C_{S_1} = C_{ST}$. Totéž samozřejmě platí i pro algoritmus S_1^T.

Z předchozí zpětné analýzy algoritmu BEM ale rekurzivně plyne i stabilita algoritmu S_2 a S_2^T atd., pokud konstanty $C_{S_1}, C_{S_2}, \ldots$ nerostou příliš rychle.

Uvažujme pevné k, $0 \le k \le m-1$ a předpokládejme, že $\mathbf{u}\left\|A_k^{-1}\right\|\,\|A_{k+1}\| \ll 1$, $\mathbf{u}\kappa(A_{k+1}) \ll 1$. Pak podle (2.66) máme přibližně

$$C_{S_{k+1}} = 3C_{S_k} + 3C_{IP} + 25. \tag{2.67}$$

Z tohoto lze usoudit, že růst konstant stability je exponenciální. V takovém případě bychom však mohli použít algoritmus BEMW pouze pro malá m.

Ve skutečnosti je růst těchto konstant mnohem pomalejší. Člen $3C_{S_k}$ vznikl z nerovností (2.63) a (2.64), které jsou původně odvozeny v důkazu věty 2.4 (rovnost (2.61)). Velikost $\overline{\xi}$, které se vyskytuje v této rovnosti, je v našem případě už aproximací $\overline{x}$. Proto se poruchy A_k řádu $\mathbf{u}C_{Sk}\,\|A_k\|$, které odpovídají členům $(\Delta_v A)\theta_{10}^{-1}\theta_9\overline{x}$ a $(\Delta_v A)\overline{\xi}$ v rovnosti (2.61), vyruší. (Opět zanedbáme členy vyššího řádu.)

Tedy místo (2.67) máme

$$C_{S_{k+1}} = C_{S_k} + 3C_{IP} + 25, \tag{2.68}$$

což představuje (podstatně pomalejší) lineární růst konstant stability.

Ukázali jsme, že algoritmus BEM je stabilní pro $m = 1$ a závislost konstant stability algoritmu BEMW na rostoucím m je lineární. To je vzhledem k tomu, že m v praxi nabývá často pouze malých hodnot (např. $m = 3$), více než uspokojivý výsledek. (Podrobněji se lze o zpětné stabilitě algoritmů BEM a BEMW dočíst v [3] a [4].)

Kapitola 3

Paprzycki - Yalamov

V této kapitole nejdříve představíme druhý z algoritmů pro řešení soustavy (2.1) s obroubenou maticí

$$M = \begin{pmatrix} A & B \\ C^T & D \end{pmatrix},$$

který je založen na perturbacích trojúhelníkového rozkladu. (Opět $A \in \mathbb{R}^{n\times n}$, $B, C \in \mathbb{R}^{n\times m}$, $D \in \mathbb{R}^{m\times m}$.) Tento algoritmus představili v roce 1999 Plamen Y. Yalamov a Marcin Paprzycki.

Druhá část této kapitoly je věnována zaokrouhlovacím chybám a zpětné stabilitě tohoto algoritmu.

3.1 Popis algoritmu

Základní myšlenka algoritmu spočívá v tom, že provedeme malou změnu prvků matice, kdykoliv hrozí dělení malým číslem. Řešení, které získáme, je pak ale řešením problému perturbovaného. Řešení původního problému získáme aplikací algoritmu iteračního zpřesnění řešení.

Celý postup je (stejně jako u algoritmu BEM) založen na blokovém LU rozkladu matice M:

$$M = \begin{pmatrix} PLU & 0 \\ C^T & I_m \end{pmatrix} \begin{pmatrix} I_n & V \\ 0 & W \end{pmatrix}, \tag{3.1}$$

kde $V = A^{-1}B$, $W = D - C^TV$. Spočítáme LU rozklad matice A s částečnou pivotací (t.j. $A = PLU$, kde P je permutační matice). Pokud matice A je špatně podmíněná (nebo singulární), matice U má na diagonále malé (nebo nulové) prvky. Dělení těmito prvky by vedlo k velkým zaokrouhlovacím chybám v konečném řešení.

Abychom překonali tento problém, zvětšíme diagonální prvky u_{ii} matice U o malé číslo η, které bude specifikováno později.

Tedy

$$\widetilde{u_{ii}} = u_{ii} + \delta u_{ii} = u_{ii} + \mathrm{Sgn}(u_{ii})\eta, \tag{3.2}$$

kde

$$\mathrm{Sgn}(a) = \begin{cases} \mathrm{sign}(a), & \text{pro } a \neq 0, \\ 1, & \text{pro } a = 0. \end{cases}$$

Po takovéto úpravě diagonálních prvků matice U můžeme řešení soustavy spočítat zřejmým způsobem pomocí blokového LU rozkladu (3.1).

Položíme

$$\begin{pmatrix} x_1 \\ y_1 \end{pmatrix} = \begin{pmatrix} I_n & V \\ 0 & W \end{pmatrix} \begin{pmatrix} x_\eta \\ y_\eta \end{pmatrix},$$

a řešíme soustavu

$$\begin{pmatrix} PLU & 0 \\ C^T & I_n \end{pmatrix} \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix}.$$

Dostaneme $Ax_1 = f$ a $C^T x_1 + y_1 = g$, tedy $y_1 = g - C^T x_1$.

Nyní provedeme zpětnou substituci

$$\begin{pmatrix} I_n & V \\ 0 & W \end{pmatrix} \begin{pmatrix} x_\eta \\ y_\eta \end{pmatrix} = \begin{pmatrix} x_1 \\ y_1 \end{pmatrix}$$

a máme $x_\eta = V y_\eta = x_1$, tedy $x_\eta = x_1 - V y_\eta$, a $W y_\eta = y_1$.

Algoritmus 3.1.

1. spočítáme $PLU = A$
2. pokud $|u_{ii}| < \eta$, položíme $u_{ii} = u_{ii} + \mathrm{Sgn}(u_{ii})\eta$
3. vyřešíme $AV = B$ pomocí LU rozkladu
4. $W = D - C^T V$
5. vyřešíme $Ax_1 = f$ pomocí LU rozkladu
6. $y_1 = g - C^T x_1$
7. vyřešíme $W y_\eta = y_1$
8. $x_\eta = x_1 - V y_\eta$

Poznámka 3.1. V kroku 3 představují V a B matice typu $(m \times n)$. Soustavu $AV = B$ řešíme tedy po složkách. Nechť V_i je i-tý sloupec matice V a B_i nechť je i-tý sloupec matice B. Pro vyřešení soustavy $AV = B$ musíme vyřešit soustavy $AV_i = B_i$ pro $i = 1 \ldots m$.

Výsledkem algoritmu je perturbované řešení $z_\eta = (x_\eta^T, y_\eta^T)^T$. Protože byl ale problém perturbován, nejsme pravděpodobně příliš blízko přesnému řešení. V numerických testech se ukazuje, že pro získání řešení s přesností blízké strojové přesnosti

obvykle stačí pouze jeden krok algoritmu iteračního zpřesnění řešení.

Myšlenka algoritmu iteračního zpřesnění řešení je uvedena v kapitole 2. Algoritmus samotný pro lepší přehlednost ještě zopakujeme.

Řešíme soustavu $Mz = h$. Označme symbolem z přesné řešení této soustavy a symbolem z_c řešení vypočtené pomocí algoritmu 3.1.

Algoritmus 3.2.

1. $z = z_c$
2. $r = h - Mz$
3. w se spočte jako řešení soustavy $My = r$ užitím algoritmu 3.1.
4. $z^* = z + w$
5. $r^* = h - Mz^*$
6. je-li $\|r^*\|_2 < \|r\|_2$, jdi na 7, jinak jdi na 8
7. $r = r^*, z = z^*$, jdi na 3
8. konec, z je hledané zpřesnění řešení.

Pracná část tohoto postupu je řešení soustavy s maticí M, tedy aplikace algoritmu 3.1, kde je nejpracnějším krokem LU rozklad matice A. Ten se však provede pouze jednou, takže algoritmus 3.2 není náročný na počet operací. Proto i nejnáročnější částí celého postupu pro řešení soustavy s obroubenou matící je výpočet LU rozkladu.

Nyní se budeme zabývat tím, jaký vliv má perturbace diagonálních prvků matice U na konečné řešení. Vliv perturbace není na první pohled vidět, protože jsme během výpočtu měnili průběžné výsledky. Ale z rovnice

$$u_{ii} = a_{ii} - \sum_{j=1}^{i-1} l_{ij} u_{ji}, \tag{3.3}$$

která definuje prvek u_{ii}, plyne

$$\widetilde{u_{ii}} = u_{ii} + \delta u_{ii} = a_{ii} + \delta a_{ii} - \sum_{j=1}^{i-1} l_{ij} u_{ji}. \tag{3.4}$$

Tedy změna prvků u_{ii} je ekvivalentí stejné změně v prvcích a_{ii}. (Poznamenejme, že a_{ii} je diagonální prvek v již permutované matici A, přesto jej ale pro jednoduchost budeme značit a_{ii}.)

Ještě než tyto myšlenky shrneme, uvedeme definici maximové normy.

Definice 3.1. *Nechť $A \in \mathbb{R}^{m\times n}$ je matice. Pak maximová norma matice A je definována*

$$\|A\|_\infty = \max_{i=1,\dots,m} \sum_{j=1}^{n} |a_{ij}| .$$

Změna diagonálních prvků u_{ii} trojúhelníkové matice U má tedy stejný efekt, jako kdybychom řešili původní problém, ale s maticí $\widetilde{M} = M + \delta M$, kde

$$\|\delta M\|_\infty \le \eta. \tag{3.5}$$

Matice δM je diagonální a její diagonální prvky jsou buď nulové nebo rovné $\pm\eta$.

3.2 Analýza zaokrouhlovacích chyb

Nyní provedeme zpětnou analýzu zaokrouhlovacích chyb. Budeme předpokládat, že máme zpětně stabilní metody pro řešení soustav s maticí A a W. (Předpokládáme zpětnou stabilitu po složkách, ale předpoklad zpětné stability v normě dává obdobné výsledky.)

Nechť $X = (x_{ij})_{i,j=1}^n$ je matice typu $(n \times n)$. Pak symbolem $|X|$ budeme značit matici

$$Y = (|x_{ij}|)_{i,j=1}^n .$$

Protože podle předpokladu máme zpětně stabilní metody pro řešení soustav s maticí A a W, pro porušenou soustavu s maticí A, resp. W tedy platí

$$(A + \epsilon_A)\overline{x} = c, \quad |\epsilon_A| \le K_1 \mathbf{u} \, |A| , \tag{3.6}$$

$$(\overline{W} + \epsilon_W)\overline{y} = d, \quad |\epsilon_W| \le K_2 \mathbf{u} \, \left|\overline{W}\right| , \tag{3.7}$$

kde K_1, K_2 jsou konstanty, $\overline{x}, \overline{y}$ jsou vypočtená řešení, $\overline{W}$ je vypočtené W a $\mathbf{u}$ je zaokrouhlovací jednotka.

Provedeme nejprve zpětnou analýzu jednotlivých kroků algoritmu, t.j. analýzu blokového LU rozkladu a analýzu řešení soustavy s trojúhelníkovou maticí. Složením těchto výsledků pak dostaneme zpětnou analýzu celého algoritmu.

V dalším bude symbol $\overline{X}$ opět značit vypočtené X.

Lemma 3.1. *Vezmeme-li v úvahu zaokrouhlovací chyby, máme $L\overline{U} = M + E$, kde*

$$L = \begin{pmatrix} A & 0 \\ C^T & I \end{pmatrix}, \quad \overline{U} = \begin{pmatrix} I & \overline{V} \\ 0 & \overline{W} \end{pmatrix},$$

a

$$|E| \le |M| \begin{pmatrix} 0 & \left|\overline{V}\right| \\ 0 & I \end{pmatrix} \mu_1, \tag{3.8}$$

kde

$$\mu_1 = max\left\{K_1\mathbf{u}, \gamma_{n+1}\right\} \ a \ \gamma_{n+1} = \frac{(n+1)\mathbf{u}}{1-(n+1)\mathbf{u}}.$$

Důkaz. Je

$$L\overline{U} = \begin{pmatrix} A & 0 \\ C^T & I \end{pmatrix}\begin{pmatrix} I & \overline{V} \\ 0 & \overline{W} \end{pmatrix} = \begin{pmatrix} A & A\overline{V} \\ C^T & \overline{W}+C\overline{V} \end{pmatrix} =$$

$$= \begin{pmatrix} A & B+A\delta V \\ C^T & D+\delta W \end{pmatrix} = M + \begin{pmatrix} 0 & A\delta V \\ 0 & \delta W \end{pmatrix}.$$

Z toho máme

$$E = \begin{pmatrix} 0 & A\delta V \\ 0 & \delta W \end{pmatrix},$$

kde δV je chyba $\overline{V} - V$ a δW je chyba výpočtu $\overline{W}$, t.j. $\delta W = \overline{W} - D + C\overline{V}$.

Nyní omezíme prvky matice E. Označíme i-tý sloupec matice V, resp. B jako V_i, resp. B_i. Potom z (3.6) je

$$(A+\epsilon_A)\overline{V}_i = B_i,$$

a jednoduchou úpravou dostaneme

$$A\delta V_i = -\epsilon_A\overline{V}_i, \quad |A\delta V_i| \leq K_1\mathbf{u}\,|A|\left|\overline{V}_i\right|,$$

kde $\delta V_i = \overline{V}_i - V_i$. Tedy

$$|A\delta V| \leq K_1\mathbf{u}\,|A|\left|\overline{V}\right|.$$

Chyba δW je výsledek jednoho maticového násobení a jednoho sčítání matic. Standardní analýza zaokrouhlovacích chyb dává

$$|\delta W| \leq \left(|D| + |C|\left|\overline{V}\right|\right)\gamma_{n+1}, \quad \gamma_{n+1} = \frac{(n+1)\mathbf{u}}{1-(n+1)\mathbf{u}}$$

(viz [9]). Nakonec dostaneme

$$|E| \leq \begin{pmatrix} 0 & K_1\mathbf{u}\,|A|\left|\overline{V}\right| \\ 0 & \left(|D|+|C|\left|\overline{V}\right|\right)\gamma_{n+1} \end{pmatrix} \leq |M|\begin{pmatrix} 0 & \overline{V} \\ 0 & I \end{pmatrix}\mu_1.$$

□

Lemma 3.2. *Vypočtené řešení soustavy s blokově trojúhelníkovou maticí $Lu = b$ splňuje $(L+\Delta L)\overline{u} = b$, kde*

$$|\Delta L| \leq |L|\,\mu_2, \quad \mu_2 = max\left\{K_1\mathbf{u}, \gamma_n\right\}, \quad \gamma_n = \frac{n\mathbf{u}}{1-n\mathbf{u}}.$$

Důkaz. Řešení soustavy s blokově trojúhelníkovou maticí má dvě části. První je řešení soustavy s maticí A, druhou částí je zpětná substituce. Pro první část platí podmínka (3.6). Pro druhou máme

$$\left|\Delta C^T\right| \leq \left|C^T\right| \gamma_n, \quad |\Delta I| \leq I\gamma_n,$$

tedy

$$\Delta L = \begin{pmatrix} \Delta A & 0 \\ \Delta C^T & \Delta I \end{pmatrix}$$

a

$$|\Delta L| \leq \begin{pmatrix} K_1\mathbf{u}\,|A| & 0 \\ \left|C^T\right|\gamma_n & I\gamma_n \end{pmatrix} \leq |L|\,\mu_2.$$

□

Lemma 3.3. *Vypočtené řešení soustavy $\overline{U}z = \overline{u}$ splňuje $(\overline{U} + \Delta U)\overline{z} = \overline{u}$, kde*

$$|\Delta U| \leq \left|\overline{U}\right| \mu_3, \quad \mu_3 = max\left\{K_2\mathbf{u}, \gamma_n\right\}, \quad \gamma_n = \frac{n\mathbf{u}}{1 - n\mathbf{u}}.$$

Důkaz. Důkaz je podobný jako v případě lemmatu 3.2. Místo podmínky (3.6) použijeme podmínku (3.7). □

Věta 3.1. *Vypočtené řešení soustavy $Mz = b$ splňuje $(M + \Delta M)\overline{z} = b$, kde*

$$|\Delta M| \leq 2\,|M| \begin{pmatrix} I & \left|\overline{V}\right| \\ 0 & I \end{pmatrix} \mu, \quad \mu = \mu_1 + \mu_2 + \mu_3 + \mu_2\mu_3.$$

Důkaz. Spojením lemmat 3.1, 3.2 a 3.3 dostaneme

$$(M + \Delta M)\overline{z} = (M + E + \Delta L\overline{U} + L\Delta U + \Delta L\Delta U)\overline{z} = b.$$

Použijeme dříve spočítaná omezení pro E, ΔL a ΔU a dostaneme

$$|\Delta M| \leq |M| \begin{pmatrix} I & \left|\overline{V}\right| \\ 0 & I \end{pmatrix} \mu_1 + |L|\left|\overline{U}\right|(\mu_2 + \mu_3 + \mu_2\mu_3).$$

Nyní

$$\begin{aligned}
|L|\left|\overline{U}\right| &= \begin{pmatrix} |A| & 0 \\ \left|C^T\right| & I \end{pmatrix} \begin{pmatrix} I & \left|\overline{V}\right| \\ 0 & I\overline{\Delta} \end{pmatrix} \\
&= \begin{pmatrix} |A| & |A|\left|\overline{V}\right| \\ \left|C^T\right| & |\overline{\Delta}| + \left|C^T\right|\left|\overline{V}\right| \end{pmatrix} \\
&\leq \begin{pmatrix} |A| & |A|\left|\overline{V}\right| \\ \left|C^T\right| & |D| + 2\left|C^T\right|\left|\overline{V}\right| \end{pmatrix} \\
&\leq 2\begin{pmatrix} |A| & |B| \\ \left|C^T\right| & |D| \end{pmatrix} \begin{pmatrix} I & \left|\overline{V}\right| \\ 0 & I \end{pmatrix} = 2\,|M| \begin{pmatrix} I & \left|\overline{V}\right| \\ 0 & I \end{pmatrix}.
\end{aligned}$$

□

Zpětná analýza ve větě 3.1 platí pro každou matici M. V našem případě řešíme soustavu s maticí $M + \delta M$, kde $\|\delta M\| \leq \eta$. Máme tedy

$$(M + \delta M + \Delta M)\overline{z}_\eta = b, \tag{3.9}$$

kde $\overline{z}_\eta$ je řešení vypočtené v přítomnosti poruch δM.

Věta 3.2. *Relativní chyba řešení soustavy (3.9) splňuje*

$$\frac{\|\overline{z}_\eta - z\|_\infty}{\|\overline{z}_\eta\|_\infty} \leq 2\kappa(M)(\eta + r\mu), \tag{3.10}$$

kde

$$r = \left\| \begin{array}{cc} I & \left|\overline{V}\right| \\ 0 & I \end{array} \right\|_\infty,$$

číslo podmíněnosti $\kappa(M)$ matice M je definováno vztahem

$$\kappa(M) = max\left\{\left\|M^{-1}\right\|_\infty, \left\|\left|M^{-1}\right|\,|M|\right\|_\infty\right\},$$

a μ je stejné jako ve větě 3.1.

Důkaz. Z rovnosti (3.9) máme

$$\overline{z}_\eta - z = -M^{-1}(\delta M + \Delta M)\overline{z}_\eta.$$

Protože $\|\delta M\|_\infty \leq \eta$, je

$$\|\overline{z}_\eta - z\|_\infty \leq \left(\left\|M^{-1}\right\|_\infty \eta + \left\|\left|M^{-1}\right|\,|\Delta M|\right\|_\infty\right)\|\overline{z}_\eta\|_\infty. \tag{3.11}$$

Podle věty 3.1 a nerovnosti (3.11) máme

$$\begin{array}{rcl} \|\overline{z}_\eta - z\|_\infty & \leq & \left(\|M^{-1}\|_\infty \eta + 2\,\||M^{-1}|\,|M|\|_\infty\, r\mu\right)\|\overline{z}_\eta\|_\infty \\ & \leq & 2\kappa(M)(\eta + r\mu)\,\|\overline{z}_\eta\|_\infty. \end{array}$$

□

Věta 3.2 ukazuje, že velikost chyby $\|\overline{z}_\eta - z\|_\infty$ je určena hodnotou $\eta + r\mu$. Číslo r (*faktor růstu*) udává růst průběžných výsledků při výpočtu $\overline{V}$. Protože jsme předpokládali zpětnou stabilitu metody pro řešení soustavy s maticí A, konstanta μ je řádu $n\mathbf{u}$. V tomto případě musí být zdrojem jakékoliv velké chyby velká hodnota r, což se může stát, jestliže matice A je špatně podmíněná.

Předpokládejme, že matice A neobsahuje velké prvky. Řešíme soustavu s maticí A pomocí LU rozkladu s částečnou pivotací. V obecném případě má rovnice (3.6) tvar

(viz [9], podrobnosti o analýze zaokrouhlovacích chyb při řešení soustavy lineárních rovnic pomocí LU rozkladu lze nalézt v [1], str. 104 - 108)

$$(A + \Delta A)\overline{x} = b, \quad |\Delta A| \leq 2\gamma_n \left|\overline{L}\right| \left|\overline{U}\right|.$$

Protože metoda je zpětně stabilní, součin $\left|\overline{L}\right| \left|\overline{U}\right|$ není o moc větší než $|A|$ a tedy $\left|\overline{L}\right|$ a $\left|\overline{U}\right|$ nemá příliš velké prvky.

Prvky matice A v průběhu řešení soustavy s maticí A rostou v důsledku dělení malými diagonálními prvky matice U.

Faktor růstu r můžeme přibližně popsat jako:

$$r \cong \frac{1}{\eta^s}, \tag{3.12}$$

kde s je těžké odhadnout. (O faktoru růstu se lze podrobněji dočíst v [1], str. 116.) Volba $s = 1$ se v praxi ukazuje jako velmi dobrá.

Nyní můžeme určit hodnotu η. Z (3.10) a (3.12) je

$$\frac{\|\bar{z}_\eta - z\|_\infty}{\|\bar{z}_\eta\|_\infty} \leq 2\kappa(M)(\eta + \frac{\mu}{\eta^s}). \tag{3.13}$$

To znamená, že pro dosažení minimálního omezení relativní chyby řešení soustavy (3.9) musí $\eta + \mu/\eta^s$ být minimální. Tento výraz nabývá minima pro $\eta = (s\mu)^{\frac{1}{s+1}}$. Při volbě $s = 1$ máme tedy

$$\eta \approx \sqrt{\mu} \approx \sqrt{n\mathbf{u}}.$$

Ukázali jsme, že algoritmus je zpětně stabilní, vhodnou volbou η je $\sqrt{n\mathbf{u}}$.

Kapitola 4

Numerické testy

V této, poslední kapitole jsou uvedeny výsledky numerických testů provedených na počítači s procesorem Intel Celeron CPU 2.40 GHz a s operačním systémem MS Windows XP. Všechny výpočty byly provedeny ve dvojité přesnosti.

Tedy

$$eps = 10^{-16}, \quad \mathbf{u} = \frac{1}{2}eps \approx 10^{-16}.$$

Oba algoritmy jsou naprogramovány v programovacím jazyce C a jako dílčí kroky jsou použity funkce z balíku `LAPACK` a lineárně algebraické operace z balíku `BLAS`.

Srovnávací testy byly provedeny pro dvě různé matice A - řídkou a hustou.

Prvním typem matice A je Jakobián standardního dynamického systému, který se v literatuře označuje jako *Bruselátor*, viz [5]. Na matici A je aplikován posun $A - \lambda I$, kde λ je vybrané vlastní číslo matice A, tedy $A := A - \lambda I$.

Výsledkem této konstrukce je tedy špatně podmíněná matice A, která je řídká, konkrétně $(2, 2)$ - pásová, t.j.

$$(i - j > 2) \vee (j - i > 2) \quad \Longrightarrow \quad a_{ij} = 0 \quad \forall i, j = 1, \ldots, n,$$

kde $A = (a_{ij})_{i,j=1}^{n}$.

Hustá matice A je konstruována následujícím způsobem. Nejprve položme

$$A_0 = \mathrm{diag}(0, 0, 0, 0.7 + 0.04n, 0.7 + 0.04(n-1), \ldots, 0.86).$$

Definujme Householderovu matici

$$H_i = I - 2h_i h_i^T,$$

kde h_i je jednotkový, náhodně generovaný vektor takový, že všechny jeho složky leží v intervalu $(-1, 1)$.

Potom matice A je dána předpisem

$$A = H_1 H_2 \dots H_{100} A_0 H_{101} H_{102} \dots H_{200}.$$

Matice A je velmi špatně podmíněná, její singulární čísla jsou $0, 0, 0, 0.7 + 0.04n$, $0.7+0.04(n-1), \dots, 0.86$. To, že matice A není singulární, je způsobeno pouze zaokrouhlovacími chybami.

Volba matic B, C, D a vektorů pravé strany f, g je v obou případech (pro A řídkou i husou) stejná.

Vektory pravé strany f, g se získají tak, že se zvolí přesné řešení x_p, y_p a vektory f a g se vypočtou ze vztahů

$$\begin{aligned} f &= Ax_p + By_p, \\ g &= C^T x_p + Dy_p. \end{aligned}$$

Matice B, C, D a vektory x_p, y_p jsou voleny náhodně tak, že všechny jejich prvky leží v intervalu $(-1, 1)$.

První z algoritmů - tedy algoritmus BEMW - budeme v dalším nazývat *algoritmem Govaerts - Pryce* (Gov. - Pryce), druhý (iterační) algoritmus budeme nazývat *algoritmem Paprzycki - Yalamov* (Pap. - Yal.).

V algoritmu Paprzycki - Yalamov je hodnota η volena

$$\eta = 10^{-8} \approx \sqrt{\mathbf{u}}.$$

V algoritmu Govaerts - Pryce je jako zpětně stabilní metoda pro řešení soustavy s maticí A použit LU rozklad s částečnou pivotací (funkce z balíku `LAPACK`).

Srovnávacím kritériem je vypočetní čas v sekundách a absolutní dopředná chyba, tedy $\|z_p - z\|_2$, kde $z_p = \left(x_p^T, y_p^T\right)^T$ je přesné (předem zvolené) řešení a $z = \left(x^T, y^T\right)^T$ je řešení vypočtené.

Testy byly prováděny pro řídkou matici A o rozměrech 100×100 a 500×500 a pro hustou matici A o rozměrech 100×100 a 200×200.

V tabulkách 4.1 a 4.2 je vidět závislost výpočetního času a dopředné absolutní chyby na rostoucím m nejprve pro řídkou matici A.

	Čas		Chyba	
m	Gov. - Pryce	Pap. - Yal.	Gov. - Pryce	Pap. - Yal.
1	0.015	0.015	$3.2 \cdot 10^{-6}$	$6.4 \cdot 10^{-6}$
2	0.015	0.031	$6.9 \cdot 10^{-6}$	$8.6 \cdot 10^{-6}$
4	0.015	0.015	$2.1 \cdot 10^{-6}$	$4.7 \cdot 10^{-6}$
6	0.015	0.031	$1.1 \cdot 10^{-5}$	$3.6 \cdot 10^{-6}$
8	0.015	0.031	$5.5 \cdot 10^{-6}$	$2.5 \cdot 10^{-6}$
10	0.031	0.031	$3.2 \cdot 10^{-6}$	$7.1 \cdot 10^{-6}$
14	0.031	0.015	$4.6 \cdot 10^{-5}$	$8.2 \cdot 10^{-6}$
18	0.062	0.046	$1.2 \cdot 10^{-5}$	$8.7 \cdot 10^{-6}$
22	0.062	0.046	$6.9 \cdot 10^{-6}$	$5.9 \cdot 10^{-6}$
25	0.062	0.062	$2.4 \cdot 10^{-5}$	$4.5 \cdot 10^{-6}$
30	0.078	0.062	$1.3 \cdot 10^{-5}$	$2.3 \cdot 10^{-6}$

Tabulka 4.1: A - řídká matice (100×100), závislost výp. času a chyby na m

	Čas		Chyba	
m	Gov. - Pryce	Pap. - Yal.	Gov. - Pryce	Pap. - Yal.
1	0.046	0.031	$3.7 \cdot 10^{-4}$	$5.1 \cdot 10^{-4}$
5	0.062	0.046	$1.2 \cdot 10^{-4}$	$4.8 \cdot 10^{-4}$
10	0.109	0.046	$2.6 \cdot 10^{-4}$	$2.6 \cdot 10^{-4}$
15	0.140	0.062	$5.7 \cdot 10^{-3}$	$7.1 \cdot 10^{-4}$
20	0.187	0.062	$4.8 \cdot 10^{-4}$	$6.0 \cdot 10^{-4}$
30	0.256	0.092	$1.7 \cdot 10^{-3}$	$5.4 \cdot 10^{-4}$
40	0.359	0.087	$4.2 \cdot 10^{-3}$	$1.1 \cdot 10^{-4}$

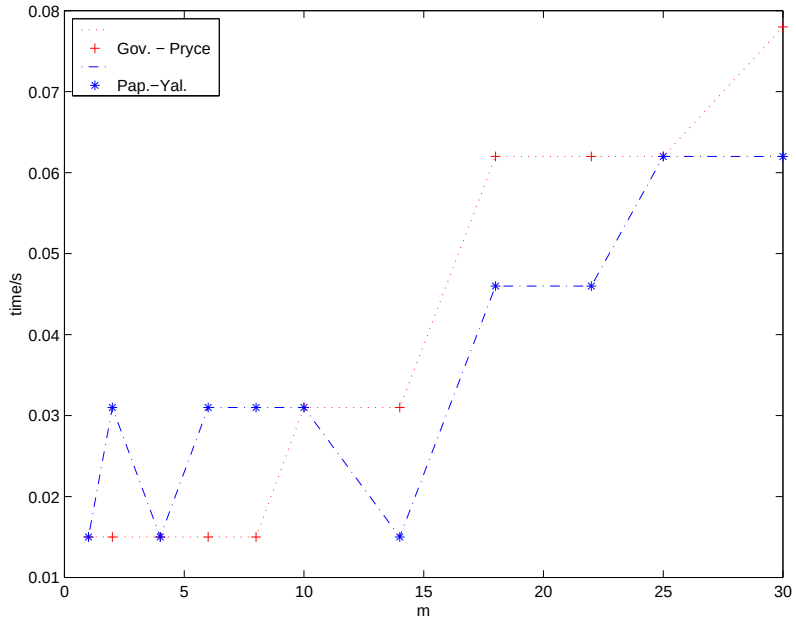
Tabulka 4.2: A - řídká matice (500×500), závislost výp. času a chyby na m

Výsledky jsou ještě pro lepší přehlednost znázorněny na obrázcích 4.1 - 4.4 v podobě grafů závislosti výpočetního času, resp. chyby na rostoucím m.

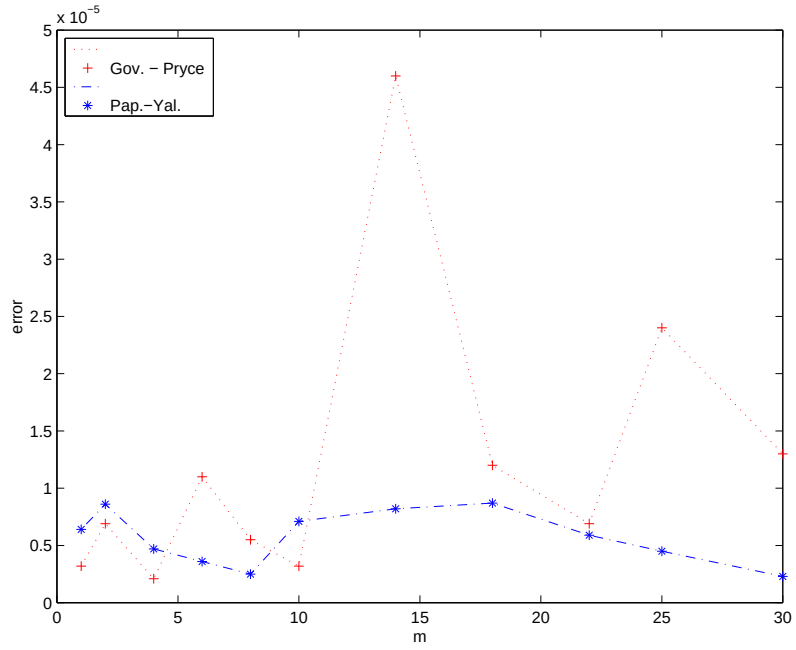
Při všech výpočtech pomocí algoritmu Paprzycki - Yalamov stačil pouze jeden krok iteračního zpřesnění. (Iterační zpřesnění bylo prováděno, dokud se norma rezidua zmenšovala alespoň o řád.)

Algoritmus Paprzycki - Yalamov se ukázal jako o něco rychlejší. Pro matici (100×100) byl růst výpočetního času u obou algoritmů v podstatě stejný. Pro matici (100×100) byl již rozdíl v růstu výpočetního času výraznější, výpočetní čas u algoritmu Paprzycki - Yalamov je téměř konstatní.

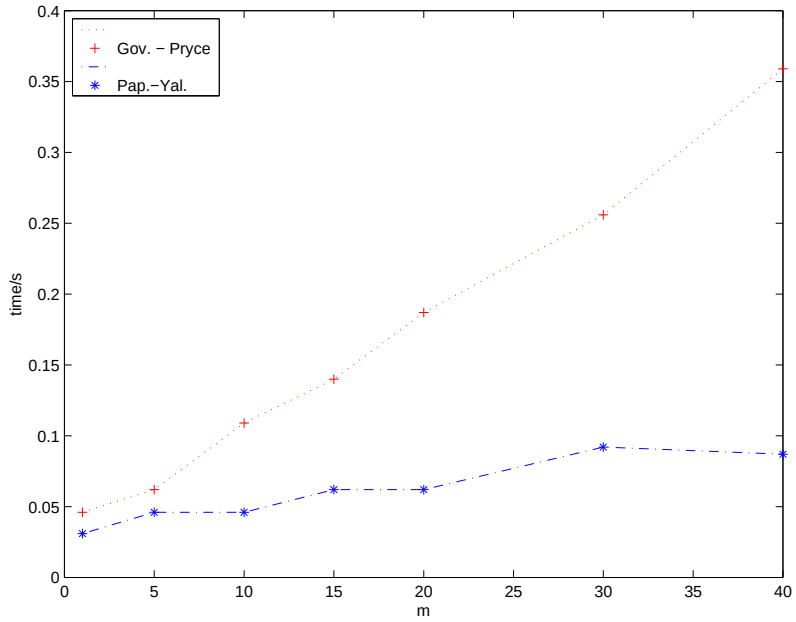
Oba algoritmy počítají v podstatě stejně přesně, občas se stane, že chyba algoritmu Govaerts - Pryce 'poskočí' o řád nahoru.



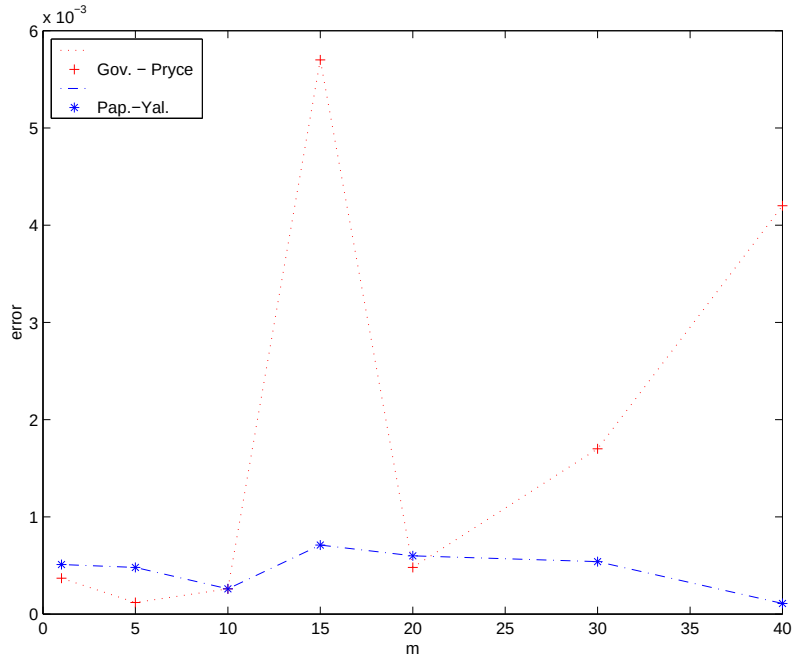
Obrázek 4.1: A - řídká matice (100×100), závislost výp. času na m



Obrázek 4.2: A - řídká matice (100×100), závislost chyby na m



Obrázek 4.3: A - řídká matice (500×500), závislost výp. času na m



Obrázek 4.4: A - řídká matice (500×500), závislost chyby na m

Tabulky 4.3 a 4.4 ukazují závislost výpočetního času a absolutní dopředné chyby na rostoucím m, tentokrát pro hustou matici A.

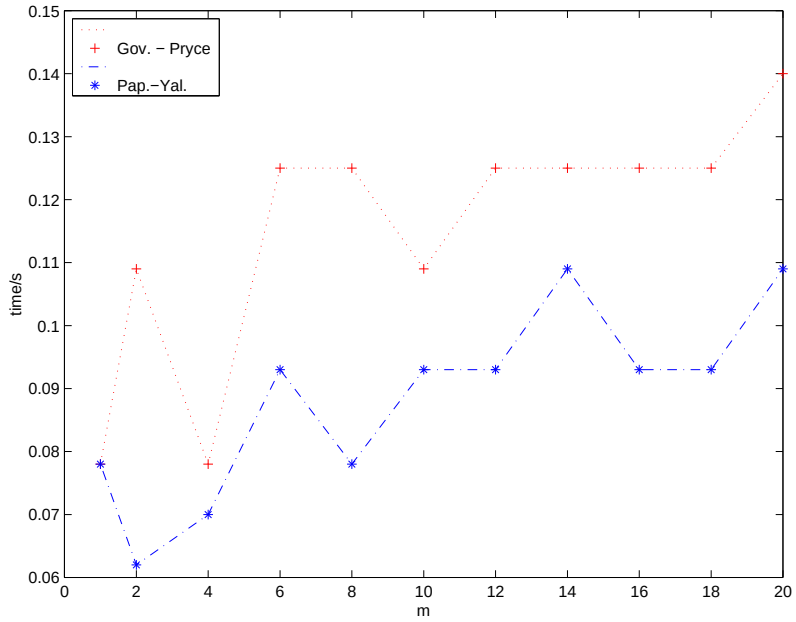
	Čas		Chyba	
m	Gov. - Pryce	Pap. - Yal.	Gov. - Pryce	Pap. - Yal.
1	0.078	0.078	7.5^*	$2.4 \cdot 10^{-6}$
2	0.109	0.062	29.9^*	$9.9 \cdot 10^{-7}$
4	0.078	0.070	$1.3 \cdot 10^{-6}$	$3.1 \cdot 10^{-6}$
6	0.125	0.093	$1.1 \cdot 10^{-6}$	$2.1 \cdot 10^{-6}$
8	0.125	0.078	$1.2 \cdot 10^{-6}$	$2.6 \cdot 10^{-6}$
10	0.109	0.093	$8.1 \cdot 10^{-6}$	$4.6 \cdot 10^{-6}$
12	0.125	0.093	$7.0 \cdot 10^{-5}$	$4.8 \cdot 10^{-6}$
14	0.125	0.109	$1.9 \cdot 10^{-5}$	$3.9 \cdot 10^{-6}$
16	0.125	0.093	$1.5 \cdot 10^{-5}$	$5.2 \cdot 10^{-6}$
18	0.125	0.093	$8.9 \cdot 10^{-6}$	$5.3 \cdot 10^{-6}$
20	0.140	0.109	$1.3 \cdot 10^{-5}$	$5.8 \cdot 10^{-6}$

Tabulka 4.3: A - hustá matice (100×100), závislost výp. času a chyby na m

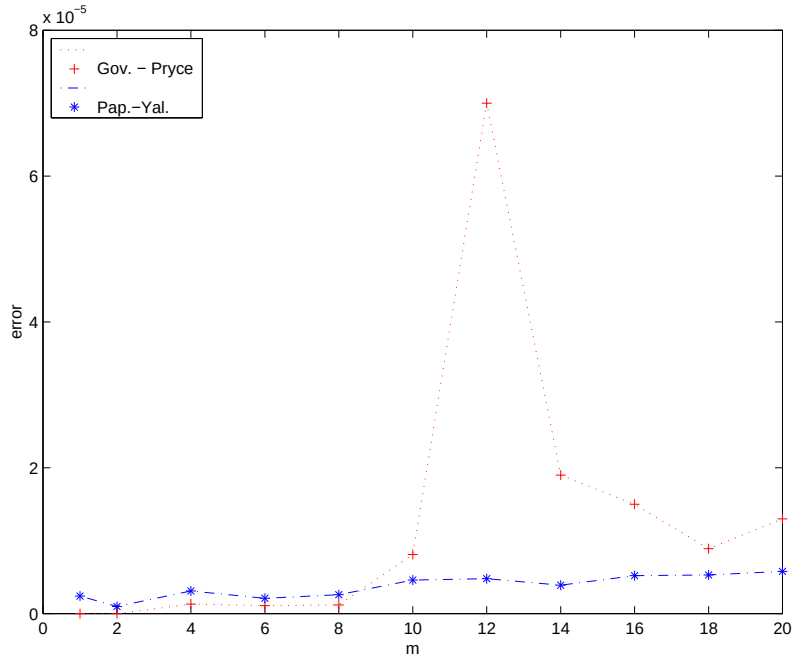
	Čas		Chyba	
m	Gov. - Pryce	Pap. - Yal.	Gov. - Pryce	Pap. - Yal.
1	0.375	0.312	30.4^*	$5.2 \cdot 10^{-6}$
5	0.515	0.328	$1.5 \cdot 10^{-5}$	$9.7 \cdot 10^{-6}$
10	0.390	0.359	$1.2 \cdot 10^{-5}$	$1.2 \cdot 10^{-5}$
15	0.468	0.343	$3.1 \cdot 10^{-5}$	$3.8 \cdot 10^{-5}$
20	0.968	0.500	$1.3 \cdot 10^{-5}$	$8.5 \cdot 10^{-5}$
25	0.968	0.423	$2.4 \cdot 10^{-5}$	$2.3 \cdot 10^{-5}$
30	1.078	0.521	$3.4 \cdot 10^{-4}$	$1.1 \cdot 10^{-5}$

Tabulka 4.4: A - hustá matice (200×200), závislost výp. času a chyby na m

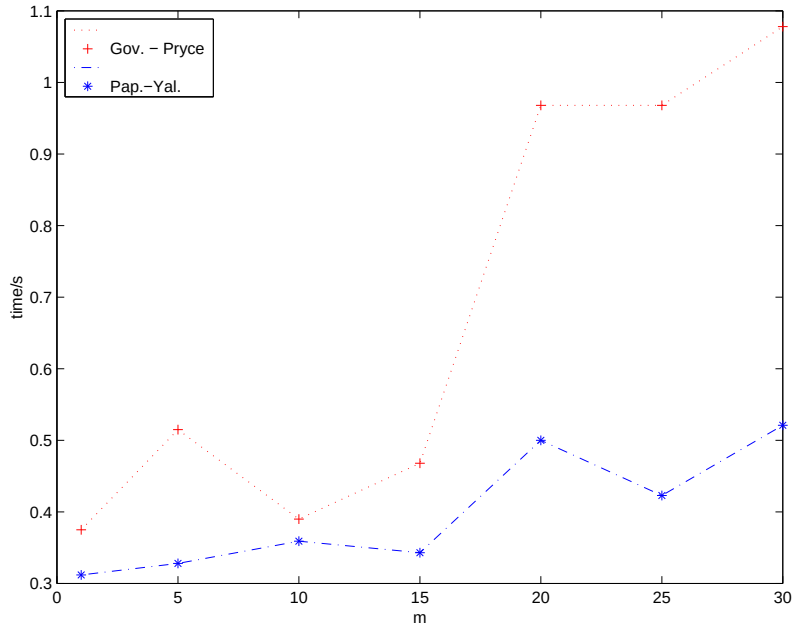
Na obrázcích 4.5 - 4.8 je opět tato závislost vyjádřena graficky.



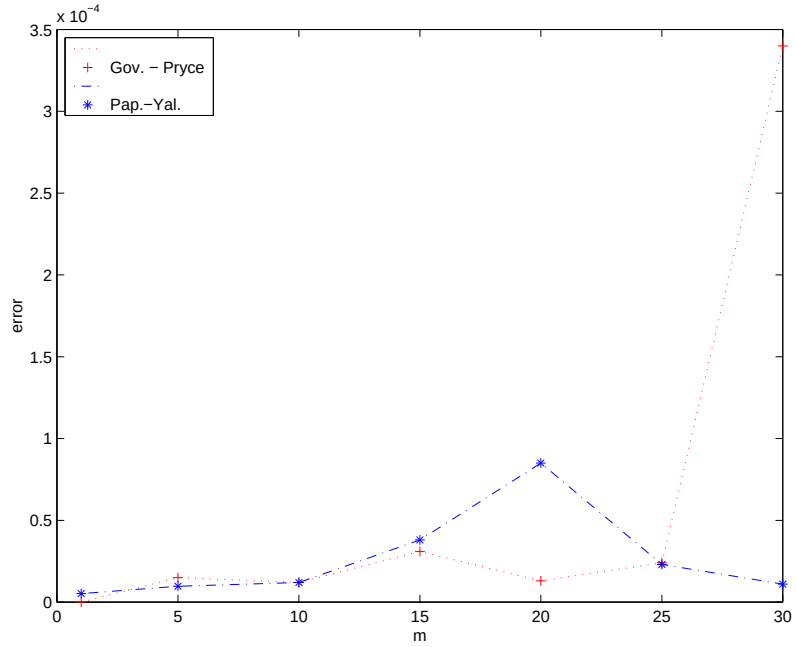
Obrázek 4.5: A - hustá matice (100×100), závislost výp. času na m



Obrázek 4.6: A - hustá matice (100×100), závislost chyby na m



Obrázek 4.7: A - hustá matice (200×200), závislost výp. času na m



Obrázek 4.8: A - hustá matice (200×200), závislost chyby na m

Při výpočtech s maticí A typu (100×100) bylo občas třeba udělat 2 kroky iteračního zpřesnění. (Bez něj byla chyba obvykle 100-krát větší.) V ostatních případech stačil opět pouze jeden krok iteračního zpřesnění.

Pro hustou matici A o rozměrech (100×100) je algoritmus Paprzycki - Yalamov opět o něco rychlejší, výpočetní čas ale roste u obou algoritmů stejně rychle. Pro matici A o rozměrech (200×200) je růst výpočetního času v závislosti na m u algoritmu Govaerts - Pryce podstatně rychlejší než u algoritmu Paprzycki - Yalamov.

Zajímavé je, že algoritmus Govaerts - Pryce selhává pro špatně podmíněnou matici M ($\kappa(M) \approx 10^{15}$). Absolutní dopředná chyba je potom řádu 10^0 až 10^1. V tabulkách jsou tyto velké hodnoty absolutní chyby označeny *, v grafech jsou vynechány (resp. nahrazeny nulou).

Algoritmus Paprzycki - Yalamov však počítá soustavy i s touto špatně podmíněnou maticí M správně.

Algoritmus Paprzycki - Yalamov se ukázal jako o něco efektivnější, pro větší matice s větším vroubením podstatně rychlejší a celkově poněkud přesnější.

Literatura

[1] Golub, G. H. van Loan, C. F. *Matrix computations*, The Johns Hopkins University Press, 1996.

[2] Govaerts, W. *Numerical Methods for Bifurcation of Dynamical Equilibria*, SIAM, Philadelphia, 2000.

[3] Govaerts, W. *Stable solvers and block elimination for bordered systems*, SIAM J. Matrix Anal. Appl. 12(1991) 469-483.

[4] Govaerts, W. and Pryce, J. D. *Mixed block elimination for linear systems with wider borders*, IMA J. Numer. Anal. 13(1993) 161-180.

[5] Hairer, E., Wanner, G. *Solving Ordinary Differential Equations II, Stiff and Differential-Algebraic Problems*, Springer-Verlag, Berlin, 1991

[6] Práger, M. *Numerická matematika I.*, SPN, Praha, 1981.

[7] Quarteroni, A., Sacco, R., Saleri, F. *Numerical mathematics*, Springer Verlag, New York, 2000.

[8] Segethová, J. *Základy numerické matematiky*, Univerzita Karlova v Praze - Nakladatelství Karolinum, Praha, 2002.

[9] Yalamov, P. Y., Paprzycki, M. *Stability and performance analysis of block elimination solver for bordered linear system*, IMA J. Numer. Anal. 19:335-348, 1999.