Pokroky matematiky, fyziky a astronomie

Jan Šípek; Jan Zítko

Algoritmy na výpočet kořenů polynomu

Pokroky matematiky, fyziky a astronomie, Vol. 46 (2001), No. 1, 33--42

Persistent URL: `http://dml.cz/dmlcz/141061`

Algoritmy na výpočet kořenů polynomu

Jan Šípek a Jan Zítko, Praha

Úvod

Již Babylóňané, řečeno v termínech současného myšlení, řešili kvadratické rovnice. Tyto poznatky spadají do období 2. tisíciletí před naším letopočtem. Trvalo pak celá tisíciletí, než se dospělo k zavedení moderního matematického formalismu, tj. k řešení polynomiální rovnice ve tvaru

$$p(z) = a_n z^n + a_{n-1} z^{n-1} + \cdots + a_0 = 0.$$

Naším cílem není popsat historický vývoj formulace tohoto problému. Ten čtenář může nalézt např. v přehledu, který je v článku V. Y. Pana [Pv97], kde jsou navíc uvedeny další citace z historie matematiky [Be40], [Bo68], [Ev83]. Chceme tím jenom dokumentovat, že řešení polynomiálních rovnic patří k jedné z nejstarších matematických úloh, které stojí v současné době trochu stranou pozornosti, i když se občas objeví nové a zajímavé články z této oblasti. Výzkum v oboru numerických metod je v současnosti soustředěn na numerické řešení diferenciálních rovnic, numerickou lineární a nelineární algebru, optimalizační problémy, a to vše s přihlédnutím na hekticky se rozvíjející počítačovou techniku.

Vraťme se však k polynomům. Formule na výpočet kořenů polynomu obsahující odmocniny byla známa pro polynomy stupně $n \leqq 4$. Ruffini v roce 1813 a Abel v roce 1827 dokázali neexistenci analogické formule pro polynomy stupně $n \geqq 5$. Tento výzkum završil Galois v roce 1832. Podle toho se tudíž kořeny obecného polynomu stupně většího než čtyři počítají numericky, přičemž do dnešních dnů bylo sestaveno několik stovek iteračních metod. Z tohoto důvodu se na jedné straně úloha najít kořeny polynomů považuje za uzavřenou. Mnohdy není ani zájem rozvíjet dále teorii. Případný projekt by asi nebyl finančně podporován, a to nejen u nás. Inženýři dříve počítali vlastní čísla matic jako kořeny charakteristického polynomu. Avšak s rozvojem iteračních metod na výpočet vlastních čísel matic byly sestaveny rychlé algoritmy, kterým, pokud srovnáváme rychlost, mohou klasické algoritmy pracující s charakteristickým polynomem jen těžko konkurovat. Podívejme se však na celou věc s trochou abstrakce. Budeme-li počítat obecně komplexní vlastní čísla komplexní matice a budou-li tato vlastní čísla vytvářet shluky nebo budou-li násobná, pak

---

Mgr. Jan Šípek (1975) a doc. RNDr. Jan Zítko, CSc. (1940), katedra numerické matematiky, Matematicko-fyzikální fakulta Univerzity Karlovy, Sokolovská 83, 186 75 Praha 8.

Tento článek byl napsán díky podpoře grantů MSM 113200007 a GAČR 201/00/0080.

maticová iterační metoda nepočítá přesně, anebo zcela selhává. Vyvstává tedy otázka, zdali bychom nenašli spolehlivě vlastní čísla, vrátíme-li se k polynomům. Uvědomíme-li si dále fakt, že polynomiální rovnice je speciálním případem nelineární rovnice, umožňuje aplikace postupů z teorie nelineárních rovnic na rovnice polynomiální další rozvoj a optimalizaci algoritmů pro výpočet kořenů polynomu (viz např. kniha [Ku98]).

Jak je vidět, i v této oblasti je možné najít nové a zajímavé otázky. Uvažovali jsme postupy v poslední době nejvíce propagované a zejména jejich aplikace na počítači. Domníváme se, že je užitečné seznámit širší matematickou veřejnost s moderními postupy a upozornit ji na „robustní“ program, který byl na základě získaných poznatků sestaven a odzkoušen a může sloužit pro výuku i pro praxi.

V současné literatuře se cituje nejvíce metoda založená na Lehmerově-Schurově větě, Bairstowova, Laguerrova, Newtonova, Aberthova a Jenkinsova-Traubova metoda a postupy vzniklé jejich kombinací. Numerické experimenty ukázaly, že v době počítačů nemá význam se zabývat metodami, jako je Graeffova či Bernoulliova, i přes jejich nesporný přínos do teorie numerické matematiky. Vyzkoušet všechny postupy, které se kdy v literatuře objevily, je nereálné a v současné době, která klade důraz především na ekonomickou stránku řešených úkolů, i neuskutečnitelné. Cílem výzkumu v rámci studia bylo nalézt kombinaci metod z již citovaných, která by spolehlivě řešila i „problémové“ polynomy. Zde půjdou proti sobě dva faktory, jak se ostatně dá očekávat, a to spolehlivost a rychlost výpočtu.

Uvedené metody jsme naprogramovali a otestovali navíc při použití násobné aritmetiky. Tímto postupem jsme dosáhli zajímavých výsledků a není nám známo, že by někdo tento postup zkoušel. Studium numerických výsledků pro různý počet platných míst bylo zajímavé a ukázalo, že i pro polynomy stupně 200 a více s nepříznivým rozložením kořenů je možné nalézt všechny kořeny s předepsanou přesností. Do článku jsme zařadili již uvedené metody. Podrobněji pojednáme o nejspolehlivější metodě, která je založena na Lehmerově-Schurově větě.

V závěru pojednání uvádíme numerické výsledky, které byly získány naším programem. Ten byl navržen tak, aby bylo možno aritmetiku násobnou a vestavěnou kombinovat, tj. některé části výpočtu proběhnou ve vestavěné, některé v násobné aritmetice. Volba je ponechána na uživateli. Všechny výpočty lze samozřejmě provést pro obecné polynomy s komplexními koeficienty i kořeny. Tento program je k dispozici na `www` adrese `http://artax.karlin.mff.cuni.cz/~honza/polynom/` a domníváme se, že může být řadě čtenářů užitečný. Dotazy je možné posílat na e-mailovou adresu `honza@atrey.karlin.mff.cuni.cz`.

V celé práci budeme vyšetřovat polynom

$$p(z) = a_n z^n + a_{n-1} z^{n-1} + \cdots + a_0 \tag{1}$$

s komplexními koeficienty a budeme předpokládat, že $a_n a_0 \neq 0$.

Bairstowova metoda

Myšlenka této metody spočívá v počítání kvadratických faktorů. Dělení polynomu $p(z)$ polynomem $z^2 + uz + v$, kde $u, v \in \mathbb{C}$ [1]), zapišme ve tvaru

$$p(z) = (z^2 + uz + v)(b_n z^{n-2} + b_{n-1} z^{n-3} + \cdots + b_2) + b_1(z + u) + b_0, \qquad (2)$$

kde $b_i \in \mathbb{C}$. Je ihned vidět, že polynom $z^2 + uz + v$ dělí polynom $p(z) = a_n z^n + \cdots + a_0$ stupně $n > 2$, právě když $b_1 = b_0 = 0$. Snadno nahlédneme, že b_0 a b_1 jsou polynomiálními funkcemi u a v. Pišme dále $b_0 = b_0(u, v)$, $b_1 = b_1(u, v)$. Nalezení kvadratického faktoru jsme tímto převedli na řešení systému dvou nelineárních rovnic

$$b_0(u, v) = 0, \quad b_1(u, v) = 0.$$

K řešení systému použijeme Newtonovu-Kantorovičovu metodu. Tím obdržíme posloupnosti $\{u_j\}_{j=0}^{\infty}$ a $\{v_j\}_{j=0}^{\infty}$; pokud konvergují k $\tilde{u}$, resp. $\tilde{v}$, pak $z^2 + \tilde{u}z + \tilde{v}$ je kvadratickým faktorem polynomu $p(z)$. Spočteme jeho kořeny, a tím dostaneme dva kořeny polynomu $p(z)$. Získaným kvadratickým faktorem vydělíme polynom $p(z)$ a dosadíme za polynom původní. Proces opakujeme, pokud stupeň uvažovaného polynomu je větší než 2. Vidíme, že celá metodika je postavena na Newtonově-Kantorovičově metodě. Lze ukázat, že je-li $z^2 + \tilde{u}z + \tilde{v}$ kvadratický faktor polynomu $p(z)$ a jsou-li kořeny tohoto faktoru dva navzájem různé a jednoduché kořeny polynomu $p(z)$, pak existuje okolí řešení $(\tilde{u}, \tilde{v})$, ve kterém Newtonova-Kantorovičova metoda neselže. Odtud jsou zároveň patrné možnosti použití Bairstowovy metody.

Laguerrova metoda

Předpokládejme nejprve, že koeficienty polynomu p jsou reálné a jeho kořeny $z_1, z_2, \ldots, z_n$ jsou reálné a navzájem různé. Budiž $z_1 < z_2 < \cdots < z_n$. Definujme z důvodů jednoduchosti zápisu $z_0 = -\infty$ a $z_{n+1} = +\infty$ a dále budiž $\boldsymbol{I}_i = \langle z_i, z_{i+1} \rangle$ pro $i = 0, \ldots, n$. Zvolíme-li reálné číslo $z \neq z_j$ pro všechna j, existuje index i tak, že $z \in \boldsymbol{I}_i$. Sestrojme kvadratickou funkci, která bude mít oba své kořeny v intervalu $\boldsymbol{I}_i$ a v bodě z bude mít zápornou hodnotu. Představíme-li si to geometricky, pak sestrojíme parabolu, která má oba své průsečíky s osou x v intervalu $\boldsymbol{I}_i$ a její graf probíhá v okolí bodu z pod osou x. Parabol s výše popsanou vlastností můžeme sestrojit nekonečně mnoho. Myšlenka Laguerrovy metody spočívá v sestrojení takové paraboly, která má průsečíky s osou x co nejblíže ke krajním bodům intervalu $\boldsymbol{I}_i$, což jsou kořeny polynomu p. Podle pravidla, které uvedeme za chvíli, z těchto dvou průsečíků vybereme jeden (označme jej na chvíli stejným písmenem z) a opět konstruujeme parabolu, která probíhá v okolí nového bodu z pod osou x a má dva průsečíky v intervalu $\boldsymbol{I}_i$. A opět vybereme jeden z průsečíků a tento postup opakujeme. Touto úvahou jsme naznačili iterační postup, který jak tušíme, konverguje k jednomu z krajních bodů intervalu $\boldsymbol{I}_i$,

[1]) Obor komplexních čísel.

tj. ke kořenu polynomu p. Tento postup lze zapsat jako jednokrokovou stacionární iterační metodu (viz [Ra65])

$$z^{k+1} = z^k - \frac{np(z^k)}{p'(z^k) \pm \sqrt{\mathbf{H}(z^k)}}\,, \tag{3}$$

kde $\mathbf{H}(z) = (n-1)\{(n-1)(p'(z))^2 - np(z)p''(z)\}$. Iteraci (3) nazveme Laguerrovou metodou. Za počáteční přiblížení z^0 volíme libovolné reálné číslo. Pokud by náhodou bylo kořenem polynomu, vydělíme polynom lineárním faktorem $(z - z^0)$ a pracujeme dále s vyděleným polynomem. O volbě znaménka před odmocninou (v tomto znaménku je zakódováno, který z průsečíků paraboly vezmeme) a o konvergenci Laguerrovy metody hovoří následující věta.

Věta 1. *Nechť p je reálný polynom stupně $n \geqq 1$, jehož všechny kořeny jsou reálné a navzájem různé. Nechť $z^0 \in \mathbb{R}$ je počáteční aproximace a $p(z^0) \neq 0$. Pak iterační postup* (3)*, kde znaménko před odmocninou volíme rovné* $\operatorname{sign} p'(z^k)$*, vytváří posloupnost $\{z^k\}_{k=0}^{\infty}$, která konverguje monotónně a kubicky k některému z kořenů polynomu p. Je-li pro počáteční hodnotu $p'(z^0) = 0$, volíme znaménko před odmocninou v první iteraci libovolně. Přitom $p'(z^k) \neq 0$ pro $k = 1, 2, \ldots$*

Z monotónní konvergence plyne, že je-li $z^0 \in \boldsymbol{I}_i$, pak posloupnost aproximací konverguje buď k z_i, nebo k z_{i+1}, kde $i \in \{1, 2, \ldots, n\}$. Je-li $i = 0$ nebo $i = n+1$, pak posloupnost $\{z^k\}_{k=0}^{\infty}$ konverguje k z_1, resp. k z_n. Volba znaménka před odmocninou je rozhodující pro rychlost konvergence. Kubická konvergence znamená, že v okolí kořenu z_i se iterace, které k němu konvergují, chovají tak, že $|z^{k+1} - z_i| \sim |z^k - z_i|^3$. V praxi to znamená, že je-li absolutní hodnota $|z^k - z_i|$ rovna např. jedné desetině, pak $|z^{k+1} - z_i|$ je řádově jedna tisícina. Vidíme, že máme k dispozici velmi rychlou metodu, jejíž fungování nezávisí na rozložení kořenů, pokud jsou reálné a navzájem různé. Připomeňme si, že např. rychlost konvergence Bernoulliovy nebo Graeffovy metody závisí na rozložení kořenů, i když jsou všechny reálné a navzájem různé.

Formulujme si nyní úlohu co nejobecněji. Mějme polynom s komplexními koeficienty a s komplexními kořeny a aplikujme Laguerrovu metodu na tento polynom. Takový pokus si v počítači můžeme lehce provést. Stačí změnit deklarace proměnných. V tomto případě budeme znaménko před odmocninou volit tak, aby absolutní hodnota výrazu ve jmenovateli byla největší. Je to analogické k reálnému případu, kdy volba znaménka nám dává větší absolutní hodnotu jmenovatele. A budeme mile překvapeni. Metoda funguje výborně, jak se ukázalo v celé řadě příkladů, které jsme spočítali.

Pro komplexní polynomy s komplexními kořeny se metoda odvozuje jiným postupem. Ten je možné najít v práci B. Parletta [Pa64]. Výsledný iterační vzorec je stejný. Řekněme si ještě, co je o této metodě známo.

Jsou-li všechny kořeny polynomu reálné, ale některé z nich nejsou jednoduché, metoda sice konverguje, ale její konvergence je jen lineární v okolí násobného kořene. V případě, že metoda konverguje k jednoduchému komplexnímu kořenu, je její konvergence kubická. Případy, kdy konvergence selhává, jsou nesmírně řídké. Uveďme si na závěr příklad polynomu a volby počáteční aproximace tak, aby Laguerrova metoda selhala. Mějme polynom $p(z) = z(\frac{1}{3}z^2 + a^2)$, kde $a > 0$, tedy $p(a) \neq 0$. Zvolíme-li $z^0 = a$, pak iterace Laguerrovy metody vypadají následovně: $z^{\text{sudé číslo}} = a$, $z^{\text{liché číslo}} = -a$. Ke konvergenci stačí zvolit jiné počáteční přiblížení. S podobným jevem jsme se setkali u Newtonovy metody, kterou v tomto pojednání nerozvádíme, protože ji považujeme za všeobecně známou. Poznamenejme jenom, že vzorce pro Laguerrovu a Newtonovu metodu se jen nepatrně liší v malém okolí kořenu.

Praxe však od nás požaduje takovou metodu, která konverguje rychle a spolehlivě, přičemž s rozvojem hodně rychlých počítačů se zdůrazňuje spolehlivost. To znamená, že zadáme polynom a chceme od počítače kořeny, eventuálně kořeny v zadané oblasti komplexní roviny bez jakéhokoliv zásahu. Maximálně můžeme upozornit uživatele na to, že při změně toho či onoho parametru lze očekávat zrychlení výpočtu. Je možné například pomocí dané proměnné říci počítači, že koeficienty polynomu jsou reálné nebo že polynom má všechny kořeny reálné. Podrobně je vše uvedeno v popisu programu na `www` stránce.

Vidíme, že z tohoto hlediska samotná Laguerrova metoda neuspokojuje a že ji bude nutné kombinovat s dalším, asi pomalejším, avšak spolehlivějším postupem. Tomu bude věnován následující odstavec.

Lehmerova-Schurova metoda

V tomto odstavci se budeme zabývat separací kořenů komplexního polynomu, a to hlavně postupy založenými na Rouchéově větě.

Uvažujme opět polynom (1) s komplexními koeficienty. Označme

$$\begin{aligned} p^\star(z) &= \overline{a}_0 z^n + \overline{a}_1 z^{n-1} + \cdots + \overline{a}_n, \quad \text{tj.} \\ p^\star(z) &= z^n \overline{p(1/\overline{z})}, \end{aligned} \tag{3}$$

a definujme operátor T následovně:

$$\begin{aligned} T[p] &= \overline{a}_0 p - a_n p^\star, \quad \text{tj.} \\ T[p](z) &= \overline{a}_0 p(z) - a_n p^\star(z), \end{aligned} \tag{4}$$

a dále rekurentně

$$T^j[p] = T\bigl[T^{j-1}[p]\bigr]. \tag{5}$$

Nechť ℓ je nejmenší přirozené číslo, pro které platí

$$T^\ell[p](0) = 0. \tag{6}$$

Věta 2 (Lehmerova-Schurova). *Nechť p je polynom stupně $n \geqq 1$, $p(0) \neq 0$. Pak platí:*

(a) *Jestliže existuje přirozené číslo k takové, že $0 < k < \ell$ a $T^k[p](0) < 0$, pak polynom p má alespoň jeden kořen uvnitř jednotkového kruhu.*

(b) *Jestliže $T^j[p](0) > 0$ pro všechna $j = 1, 2, \ldots, \ell - 1$ a $T^{\ell-1}[p]$ je nenulová konstanta, pak žádný kořen neleží uvnitř jednotkového kruhu.*

Důkaz této zajímavé věty je možné najít například v [Ma49], [Ra65], [Zi75].

Poznámka. Pokud má polynom $p(z)$ kořen uvnitř kruhu o středu $c \in \mathbb{C}$ a poloměru $r \in \mathbb{R}$, pak polynom $h(z) = p(rz + c)$ má kořen uvnitř kruhu o středu 0 a poloměru 1.

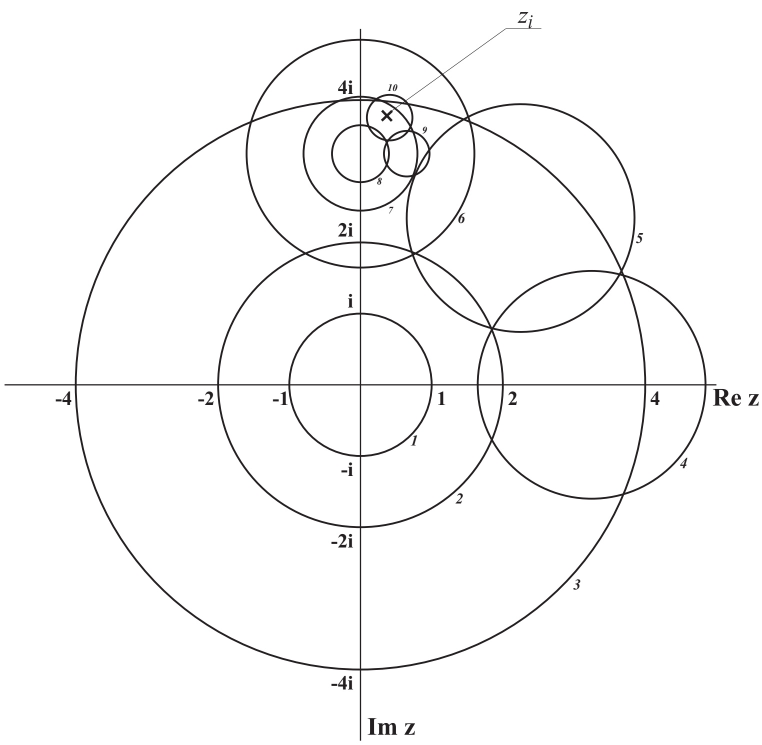
Není tudíž problém zjistit, zda kořen polynomu leží nebo neleží uvnitř libovolného kruhu v komplexní rovině. Na tom je postaven celý algoritmus. A nyní si řekneme, jak metoda funguje, a to formou „komunikace“ s počítačem, který nám bude odpovídat buď ANO, nebo NE. Je tedy zadán polynom a začneme komunikaci, která by mohla probíhat např. takto:

Leží kořen uvnitř jednotkového kruhu?	NE
Leží kořen v kruhu o poloměru 2*?*	NE
Leží kořen v kruhu o poloměru 2^2*?*	ANO

Kořen tedy leží v mezikruží o vnitřním poloměru $R = 2$ a o vnějším poloměru $2R$. Toto mezikruží pokryjeme osmi stejnými kruhy o středech

$$\frac{3R}{2\cos(\frac{1}{8}\pi)}\,\mathrm{e}^{\frac{1}{4}k\pi\mathrm{i}} \quad \text{pro} \quad k = 0, 1, \ldots, 7$$

a o poloměru $\frac{4}{5}R$. Je snadné ukázat, že to jde a že to je skoro nejmenší možný poloměr zajišťující úplné pokrytí. Výpočtem bychom zjistili, že minimální poloměr je roven číslu $\bigl(1 + \sqrt{1 + 9/(3 + 2\sqrt{2})}\bigr)\, R \doteq 0{,}7975\, R$, avšak s číslem $\frac{4}{5}R$ se nám bude lépe počítat. Provedeme přiřazení $R := \frac{4}{5}R$.



A nyní budeme probíhat postupně všech osm kruhů. Nechť pro třetí kruh nám počítač odpoví ANO, pro předchozí dva NE. Pro tento kruh pokračujeme v komunikaci. Střed pro všechny další kruhy zůstává stejný, zmenšujeme pouze poloměr.

Leží kořen uvnitř uvažovaného kruhu o poloměru R?	ANO
Leží kořen v kruhu o poloměru $R/2$?	ANO
Leží kořen v kruhu o poloměru $R/2^2$?	NE

Položíme $R := R/2^2$ a pokryjeme mezikruží o vnitřním poloměru R a o vnějším poloměru $2R$, ve kterém evidentně leží kořen, osmi stejně velkými kruhy podle již popsaného pravidla. Je vidět, že poloměry uvažovaných kruhů konvergují k nule. Jestliže tedy zvolíme poloměr nejmenšího kruhu např. 10^{-7} a provádíme-li již popsaný postup až do kruhu o takto malém poloměru, pak střed tohoto kruhu je aproximace ke kořenu polynomu s přesností 10^{-7}. Je nasnadě od tohoto středu nastartovat některou z rychlých metod, např. metodu Newtonovu nebo Laguerrovu. Praxe dokonce ukazuje, že stačí zachytit kořen v kruhu o poloměru 10^{-1} až 10^{-2}. Máme-li spočtenu aproximaci kořene, vydělíme polynom lineárním faktorem a pokračujeme s vyděleným polynomem od začátku.

Shrňme nyní již odvozené vlastnosti:

1. Rychlost konvergence Lehmerovy-Schurovy metody není žádným způsobem ovlivněna násobností kořenů nebo jejich shluky. U řady iteračních metod se tento problém vyskytuje.
2. V každém kroku můžeme přejít k rychleji konvergující metodě (např. k Newtonově metodě). Konvergence této metody bude záviset hlavně na tom, jak blízko jsme se dostali ke kořenu. Pokud metoda konvergovat nebude, můžeme se vrátit k Lehmerově-Schurově metodě a pokusit se zlepšit aproximaci kořene.
3. Po nalezení kořene vydělíme polynom lineárním faktorem, který tento kořen obsahuje, a můžeme pokračovat v hledání dalších.

Díky těmto vlastnostem je Lehmerova-Schurova metoda ideální jako základ adaptivního algoritmu pro hledání všech kořenů komplexního polynomu.

Zbývá poznamenat, co dělat v případě, že nejsou splněny předpoklady věty 2. Je dokázáno, že pokud leží kořen polynomu na jednotkové kružnici nebo existují-li reciproké kořeny ($z_i = 1/\overline{z}_j$) polynomu, pak $T^{\ell-1}[p](z)$ nevyjde jako nenulová konstanta. V tomto případě poloměr vyšetřovaného kruhu o málo zvětšíme nebo zmenšíme a celý postup opakujeme od začátku.

Aberthova metoda

Uvažujme polynom (1) stupně n s jednoduchými kořeny $z_1, \ldots, z_n$. Nechť z_i^k, kde $i = 1, \ldots, n$, jsou navzájem různé dostatečně blízké aproximace kořenů polynomu (1).

Potom pomocí logaritmických derivací[2]) polynomu můžeme odvodit iterační proces

$$z_i^{k+1} = z_i^k - \frac{p(z_i^k)}{p'(z_i^k) - p(z_i^k)\sum\limits_{j\neq i}^{n} \frac{1}{z_i^k - z_j^k}}, \quad i = 1, 2, \ldots, n, \quad k = 0, 1, 2, \ldots,$$

který konverguje kubicky ke všem kořenům polynomu. Tento iterační proces se nazývá Aberthova metoda. Více informací je možné nalézt např. v [Ku98].

Jenkinsův-Traubův algoritmus

Tento algoritmus jsme vybrali hlavně pro srovnání, protože se stal jádrem většiny „blackbox" řešičů, jako například *Mathematica*.

Je známo, že Newtonova metoda je dána předpisem

$$z^{j+1} = z^j - \frac{p(z^j)}{p'(z^j)}, \quad j = 0, 1, \ldots$$

Jenkinsův-Traubův algoritmus nahradí derivaci $p'(z^j)$ polynomu $p(z)$ hodnotou $H^j(z^j)$ polynomu $H^j(z)$ sestrojeného v následujících třech stupních.

Algoritmus nalezne jeden kořen polynomu. Kořen je odseparován a algoritmus se použije na polynom nižšího stupně. Nechť obecný polynom p představuje původní polynom definovaný vztahem (1), kde $p(0) \neq 0$, nebo polynom po provedení $k-1$ kroků algoritmu.

Stupeň 1 (proces bez posunů).

$$\begin{aligned} H^0(z) &= p'(z), \\ H^{j+1}(z) &= \frac{1}{z}\left[H^j(z) - \frac{H^j(0)}{p(0)}\,p(z)\right], \quad j = 0, 1, \ldots, M-1. \end{aligned}$$

V praxi se přirozené číslo M většinou volí pevné již na začátku výpočtu v závislosti na přesnosti aritmetiky, např. pro vestavěnou aritmetiku $M = 5$.

Stupeň 2 (proces s pevnými posuny). Zvolme β kladné číslo takové, že

$$\beta \leqq \min_{i=1,\ldots,k-1} |z_i|,$$

kde z_i jsou kořeny nalezené v předcházejících krocích, a zvolme s tak, že $|s| = \beta$ a platí $|s - z_1| < |s - z_i|$ pro $i = 2, \ldots, k-1$. Další iterace polynomu $H^j(z)$ vypadají následovně

$$H^{j+1}(z) = \frac{1}{z-s}\left[H^j(z) - \frac{H^j(s)}{p(s)}p(z)\right], \quad j = M, M+1, \ldots, L-1,$$

kde přirozené číslo L opět volíme.

[2]) $(\log p(z))'$

Stupeň 3 (proces s proměnnými posuny). Položme

$$z^L = s - \frac{p(s)}{\tilde{H}^L(s)}$$

a proveďme

$$H^{j+1}(z) = \frac{1}{z - z^j}\left[H^j(z) - \frac{H^j(z^j)}{p(z^j)}\,p(z)\right],$$
$$z^{j+1} = z^j - \frac{p(z^j)}{H^{j+1}(z^j)}, \quad j = L, L+1, \dots,$$

kde $\tilde{H}^j$ je polynom H^j vydělený koeficientem u nejvyšší mocniny.

Více informací o tomto algoritmu naleznete např. v článcích [Je70], [Je72].

Poznámka k numerickým výsledkům

Provedli jsme testování 100 polynomů stupně nejvýše 64. Soustředili jsme se přitom na polynomy, se kterými mají školní programy na výpočet kořenů problémy.

Jsou to násobné kořeny, shluky kořenů, polynomy s velkými kořeny i koeficienty v absolutní hodnotě (např. Wilkinsonův polynom nebo polynomy s geometrickým rozložením kořenů).

Pro tento článek jsme vybrali 50 polynomů s komplexními a 50 s reálnými koeficienty. Kořeny těchto polynomů jsme poté počítali použitím Aberthovy, Jenkinsovy-Traubovy, Laguerrovy a Lehmerovy-Schurovy metody v kombinaci buď s Newtonovou, nebo Laguerrovou metodou ve vestavěné aritmetice.

Výsledky jsme zaznamenali do následujících dvou tabulek:

TABULKA 1. Polynomy s reálnými koeficienty

Metoda	Vyřešil [%]	Iterace [%]	Kořeny [%]	Koeficienty [%]	Čas [%]
Jenkinsova-Traubova	98.00	0.00	22.00	14.00	84.00
Aberthova	98.00	98.00	22.00	6.00	16.00
Laguerrova	100.00	0.00	24.00	30.00	0.00
Leh.-Schur + Lag.	100.00	0.00	16.00	30.00	0.00
Leh.-Schur + New.	100.00	2.00	16.00	20.00	0.00

TABULKA 2. Polynomy s komplexními koeficienty

Metoda	Vyřešil [%]	Iterace [%]	Kořeny [%]	Koeficienty [%]	Čas [%]
Jenkinsova-Traubova	98.00	0.00	20.00	18.00	80.00
Aberthova	98.00	98.00	28.00	8.00	20.00
Laguerrova	100.00	0.00	14.00	32.00	0.00
Leh.-Schur + Lag.	100.00	0.00	20.00	20.00	0.00
Leh.-Schur + New.	100.00	2.00	18.00	22.00	0.00

Obě tabulky mají stejnou strukturu. Ve sloupci **Vyřešil** je uvedeno procento polynomů, u kterých byly nalezeny všechny kořeny. Ve sloupci **Iterace** je uvedeno, v kolika procentech případů měla příslušná metoda nejmenší průměrný počet iterací k nalezení jednoho kořenu. Sloupec **Kořeny** ukazuje, v kolika případech byla u dané metody hodnota $\max\bigl(|p(z_i)|\bigr)$ nejmenší. Sloupec **Koeficienty** ukazuje, v kolika případech byla u dané metody hodnota $\max(|a'_i - a_i|)$ nejmenší, kde a'_i jsou koeficienty polynomu sestaveného z nalezených kořenů a a_i jsou koeficienty zadaného polynomu. Poslední sloupec ukazuje, v kolika případech byla metoda rychlejší než ostatní.

Jak je vidět z obou tabulek, Jenkinsova-Traubova metoda společně s Aberthovou metodou se zdají jako velmi rychlé a přesné, pokud vezmeme v úvahu pouze funkční hodnoty v kořenech. Na druhou stranu, pokud vezmeme v úvahu i zrekonstruované koeficienty, podává v reálném případě Laguerrova a v komplexním případě Lehmerova-Schurova metoda lepší výsledky.

Na závěr je možné říci, že pokud nám jde hlavně o přesnost všech nalezených řešení, je Lehmerova-Schurova metoda vhodnější.

L i t e r a t u r a

[Ba95] Bailey, D. H.: *A Fortran Based Multiprecision System.* Tech. Report RNR-94-013, 1995.

[Be40] Bell, E. T.: *The Development of Mathematics.* McGraw–Hill, New York 1940.

[Bo68] Boyer, C. A.: *A History of Mathematics.* John Wiley & Sons, New York 1968.

[Br78] Brent, R. P.: *A Fortran Multiple Precision Arithmetic Package.* ACM Trans. on Math. Software *4* (1978), 57–70.

[Ev83] Eves, H.: *An Introduction to the History of Mathematics.* Saunders College Publishing 1983.

[He64] Henrici, P.: *Elements of Numerical Analysis.* John Wiley & Sons, New York 1964.

[Je70] Jenkins, M. A., Traub, J. F.: *A three-stage variable-shift iteration for polynomial zeros and its relation to generalized Rayleigh iteration.* Numer. Math. *14* (1970), 252–263.

[Je72] Jenkins, M. A., Traub, J. F.: *Algorithm 419: Zeros of a Complex Polynomial.* Communications of the ACM *15* (1972), 97–110.

[Kn69] Knuth, D. E.: *The Art of Computer Programming.* Volume 2 / Seminumerical Algorithms. Addison–Wesley publishing company, Inc. 1969.

[Ku98] Kyurkchiev, N. V.: *Initial Approximations and Root Finding Methods.* 1998.

[Ma49] Marden, M.: *The Geometry of the Zeros of a Polynomial.* Amer. Math. Soc., Providence, R. I. 1949.

[Nr95] Press, W. H., Teukolsky, S. A., Vetterling, W. T., Flannery, B. P.: *Numerical Recipes in FORTRAN.* The Art of Scientific Computing, Second Edition. Cambridge University Press 1995.

[Pa64] Parlett, B.: *Laguerre's Method Applied to the Matrix Eigenvalue Problem.* Math. Comp. *18* (1964), 466–485.

[Pr81] Práger, M.: *Numerická matematika.* SPN, Praha 1981.

[Pv97] Pan, V. Y.: *Solving a polynomial equation: Some history and recent progress.* SIAM Rev. *39* (1997), 2, 187–220.

[Ra65] Ralston, A.: *A first course in numerical analysis.* McGraw-Hill, New York 1965.

[Vi87] Vitásek, E.: *Numerické metody.* SNTL, Praha 1987.

[Wi63] Wilkinson, J. H.: *Rounding Errors in Algebraic Processes.* Notes on App. Sci. No. 32. Her Majesty's Stationery Office 1963.

[Zi75] Zítko, J.: *Úvod do numerické matematiky.* SPN, Praha 1975.