

POSUDEK OPONENTA DIPLOMOVÉ PRÁCE

Student: Bc. Michal Andresík

Oponent: Ing. Jaroslav Bušek, Ph.D.

Studijní program: Informační technologie

Specializace: Softwarové inženýrství

Akademický rok: 2024/2025

Téma diplomové práce: Funkce QPmR pro výpočet kořenů kvazi-polynomu v jazyku C++

Hodnocení práce:

Zadání:

Vzhledem k tomu, že hlavním (a jediným) tvůrčím úkolem této práce byla (re)implementace algoritmu pro hledání kořenů kvazi-polynomu, původně realizovaného v prostředí MATLAB, hodnotím náročnost zadání jako průměrnou. Za výzvu lze považovat především specifické prostředí cílového programovacího jazyka (C++) a případné úpravy původního kódu za účelem zvýšení efektivity, což však nebylo součástí zadání. Zda se autor rozhodne algoritmus optimalizovat, tak záleželo výhradně na jeho iniciativě.

Splnění zadání:

V rámci zadané práce byl algoritmus QPmR pro hledání kořenů kvazi-polynomu implementován v jazyce C++ v obou požadovaných verzích, čímž bylo formálně naplněno zadání. Přesto mám výhrady ke zpracování některých dílčích úkolů, zejména k zásadnímu bodu 4. V kontextu tohoto bodu je dokumentace velmi stručná a svou podobou připomíná spíše uživatelskou příručku. Očekával bych podrobnější popis implementovaných postupů, a to i nad rámec těch, které byly převzaty z původní verze QPmR a popsány v textu práce. Například výpočty kontur, které jsou sice zmíněny v teoretické části, však nejsou v dokumentaci detailněji rozvedeny – jejich implementaci je nutné dohledávat přímo ve zdrojovém kódu.

Zvolený postup řešení:

Autor provedl rešerši na téma lineárních systémů bez zpoždění i se zpožděním. Dále popsal původní implementaci algoritmu QPmR (v2) a stručně se zmínil o obdobných algoritmech pro určování kořenů kvazi-polynomu. Věnoval se také specifickým metodám určování kontur, které jsou vhodné pro implementaci v jazyce C++. V závěru práce realizoval implementaci algoritmu QPmR (v2) v jazyce C++.

Zvolený postup řešení považuji za adekvátní vzhledem k zadání práce, nicméně mám k němu několik praktických připomínek. Popis praktické části se omezuje převážně na uživatelskou příručku k vytvořené aplikaci. Chybí mi podrobnější rozbor samotného procesu vývoje programu, včetně popisu odlišností a případných úskalí oproti původní implementaci.

Zvolený způsob datové výměny působí velmi přímočaře; očekával bych alespoň základní úvahu o jeho vhodnosti a případném využití v kontextu dalších aplikací.

Dále mám určité výhrady k metodě ověřování správnosti implementovaného algoritmu. Ta byla založena na porovnání výsledků s původní implementací v prostředí MATLAB, a to pouze na několika specifických příkladech. Tento přístup sice naznačuje shodu, avšak neprokazuje jednoznačně správnost nové implementace — ta je předpokládána na základě správnosti původní verze algoritmu. Bylo by vhodné doplnit



ověření jiným způsobem, například analytickými řešeními pro jednodušší případy nebo srovnáním s výsledky alternativních metod.

Porovnání výpočetních časů v kapitole 10 je sice relevantní jako forma benchmarku, ale vzhledem k povaze aplikace bych očekával také analýzu dosahované numerické přesnosti. Chybí rovněž ověření úplnosti – tedy schopnosti algoritmu nalézt všechny kořeny v definované oblasti.

Odborná úroveň:

Z odborného hlediska text práce postrádá dostatečnou hloubku. Autor se věnoval rozsáhlému teoretickému úvodu, který je však příliš obecný a k samotným systémům se zpožděním se dostává až v pozdější části textu. V této souvislosti mám několik připomínek.

V kontextu zadání je kapitola 1 příliš obecná. Široce pojatý úvod do dynamických systémů není pro řešení zadaných úkolů nezbytný. Bylo by vhodnější zaměřit se přímo na systémy s dopravním zpožděním a obecné informace ponechat na odborné publikace. Do klasifikace dynamických systémů v kapitole 1.1.1 bych doporučil zahrnout i pojem časové invariantnosti, která je dále zmíněna v souvislosti s lineárními systémy.

Kapitola 1.1.2 přináší minimální až žádný přínos vzhledem k řešení zadaného úkolu. Autor by se měl opět více soustředit na systémy s dopravním zpožděním a vynechat zbytečné rozšiřování textu. Kapitola 3.4, která se zabývá stabilitou systémů se zpožděním, není pro řešení algoritmu hledání nul kvazi-polynomu klíčová a nebyla ani vyžadována zadáním.

V kapitole 3 je pro označení kořenů (kvazi-)polynomu použito označení „nuly“, které není chybné, avšak v kapitole 4 autor přechází na označení „kořeny“. Doporučuji zachovat konzistentní terminologii v celé práci.

Rozsah kapitoly 4 by mohl být rozšířen na úkor nadbytečných informací uvedených v úvodu práce. Informace v této kapitole jsou relevantní a poskytují základní přehled o ustáleném portfoliu numerických metod pro hledání kořenů kvazi-polynomů. K samotným systémům s dopravním zpožděním se text dostává až po třech stránkách obecného úvodu, což je vzhledem k zadání práce pozdě. Teoretická část by mohla začínat rovnou kapitolou 2, aniž by to výrazně ovlivnilo její kvalitu.

Největší teoretický přínos spatřuji v kapitole 6, která se zabývá algoritmy pro určení konturních křivek s důrazem na implementaci v jazyce C++. Předchozí kapitoly lze považovat spíše za shrnutí, které bylo sice požadováno zadáním, ale nemuselo nabývat takového rozsahu. V kapitole 7 pak považuji rozvahy o výběru programovacího jazyka za zbytečné, jelikož zadání jasně stanovuje použití C++. Historický exkurz v kapitole 7.1 je rovněž nadbytečný.

V rovnici 2.3 a 2.4 není správné pořadí členů. Popis procesu převodu rovnice 2.6 na rovnice 2.10 je zbytečně zdoluhavý, přičemž rovnice 2.8 a 2.9 společně s komentářem jsou nadbytečné.

Tvůrčí výstup práce je popsán v kapitole 9 na pouhých šesti stranách, doplněných mnoha ilustracemi. Následující kapitola 10 obsahuje čtyři stránky příkladů. Z tohoto pohledu je výstup autora popsán velmi stručně ve srovnání s teoretickou částí, která tvoří většinu práce. Očekával bych podrobnější a odbornější popis implementace v C++, který by reflektoval poznatky získané rešerší původního kódu či alternativních postupů specifických pro tento jazyk, jak je uvedeno v kapitole 6.

Není například jasné, jak autor vyřešil určování kontur nebo zda využil nějakou formu paralelismu, o kterých se zmiňoval v teoretické části. Způsob generování kontur lze odvodit až z příkladu 10.3, kde je stručně zmíněno, že existují dvě verze programu — jedna pracuje s konturami z programu gnuplot, pro druhou verzi však text způsob generování kontur nepopisuje (nebo jsem tuto informaci přehlédl).

Popis struktury aplikace v kapitole 9.2 je velmi stručný a pouze rámcový, postrádá bližší informace o samotné implementaci, která není dále v textu rozvedena.

Formální a jazyková úroveň, rozsah práce:

Po formální stránce je text napsán dobře. Obsahuje průměrné množství překlepů a drobných chyb, například „lkdyž“, „účey“, „podporu podporu“, „meu“ či ve větě „Z obrázku 6.1 zřejmé, že...“ chybí přísudek. Struktura textu je zvolena rozumně a odpovídá typu práce. V této oblasti však postrádám lepší návaznost



implementační části na poznatky uvedené v teoretických kapitolách. Některé kapitoly by mohly být stručnější a méně rozvěklé. Další připomínky se týkají konkrétních částí textu.

Poděkování by bylo vhodné uvést alespoň jako samostatný malý nadpis, protože v současné podobě působí v textu poněkud ledabyle. Je poněkud neobvyklé, že úvod začíná až na straně 13. V obsahu navíc chybí informace o textech na předchozích stránkách, což je sice běžné, protože se tyto části do obsahu obvykle nezahrnují, ale je vhodné o tom být alespoň informován.

Úvod práce se věnuje výhradně popisu struktury práce, což není dostačující. Chybí zde motivace a obecný kontext, vymezení problému a stanovení cílů práce. Rovněž by zde měly být uvedeny použité metody a přístupy, i když je lze částečně odvodit z popisu struktury.

Některé obrázky jsou bohužel nižší kvality (mají nízké DPI) a jsou zrnité s viditelnými pixely. U převzatých obrázků je to dáno původním zdrojem, nicméně u vlastních obrázků by bylo vhodné exportovat je ve vyšším rozlišení, například obrázek 5.1.

V kapitole 5.3.1 se nachází neplatný odkaz na zdroj, což je nutné opravit. Zkratky DFS a BFS v kapitole 6.3 nejsou vysvětleny a chybí i ve výčtu zkratk v závěru práce.

Závěr je velmi strohý a měl by obsahovat komentář k dosaženým výsledkům v kontextu cílů uvedených v úvodu či zadání práce.

Vzhledem k tomu, že zásadní součástí práce je odevzdaný zdrojový kód aplikace, doporučuji v příloze uvést i strukturu souborů, což by usnadnilo orientaci v projektu.

Výběr zdrojů, korektnost citací:

Množství i kvalita vybraných zdrojů jsou na velmi dobré úrovni a autor zvolil relevantní odbornou literaturu. Výtku mám však k použití výukových prezentací v teoretické části práce, kde by bylo vhodnější opřít se o recenzované zdroje, například odborné knihy.

Namísto zdroje [3], který je pouze prezentací (i když pocházející z prestižní univerzity), bych doporučil použít silnější, recenzovaný zdroj – ideálně knihu či monografii. Zdroj [4] je rovněž pouze prezentací, přičemž stejné informace by jistě bylo možné nalézt v kvalitnější odborné publikaci.

Dále je třeba upozornit na nevhodné umístění referencí. Uvádění citací až na konci odstavce, zejména pokud jsou takto shluknuty dohromady, není vhodné. Lepší je uvádět odkazy na zdroje bezprostředně u konkrétních informací, které z nich byly čerpány.

Vkládání citací přímo do rovnic je nevhodné, protože to může být snadno zaměněno za matematickou nebo jinou operaci.

Další komentáře a hodnocení:

Namísto rozsáhlého slovního popisu kódu v kapitole 8 bych uvítal spíše vývojový diagram nebo jinou vizuální reprezentaci, která by přehledně vysvětlila algoritmus QPmR (verze 2). V kapitole 5 byl k tomuto účelu použit pseudokód, což je vhodné. Alternativně by bylo možné jednotlivé obrázky nahradit souhrnným schématem, kde by bloky reprezentovaly jednotlivé části celého kódu. Stejný přístup by se dal využít i u rozsáhlejšího kódu QPmR verze 2.

Aplikace při zadání nesprávných vstupních hodnot nenahlásí chybu ani neupozorní uživatele. Při výpočtu kořenů kvazi-polynomu, který může trvat i několik sekund, chybí indikace průběhu (status bar nebo alespoň textová informace o probíhající výpočtu), takže uživatel nemá jistotu, zda aplikace stále běží, či zda došlo k jejímu zaseknutí.

Pro nastavení programu bych spíše očekával použití INI souboru, který by byl univerzálnější než specifický textový formát použitý v současnosti. Parametry kvazi-polynomu jsou zadávány prostřednictvím textového souboru, což samo o sobě není problém, nicméně jeho formát je aplikaci vlastní a specifický. Pro snadnější datovou výměnu s jinými programy bych doporučil standardizované formáty jako CSV, XML nebo JSON.

Stejná výtku platí pro textový výstup aplikace, jehož strojové zpracování sice možné je, ale vyžaduje tvorbu vlastního parseru. Datová výměna parametrů a výsledků výhradně prostřednictvím textových souborů se jeví



jako málo efektivní. Rovněž bych uvítal možnost zadat parametry přímo přes příkazovou řádku, což by umožnilo externí volání aplikace jinými programy.

Součástí práce byla odevzdána složka projektu se zdrojovými soubory a přeloženou aplikací, přičemž velikost složky se zdrojovými soubory přesahuje 2,7 GB. Důvodem je přítomnost cache souborů z vývojového prostředí Visual Studio, které však pro samotný projekt nejsou relevantní a měly by být z odevzdání vynechány.

Celkové hodnocení:

Diplomová práce Bc. Michala Andresíka na téma Funkce QPmR pro výpočet kořenů kvazi-polynomu v jazyku C++ splňuje zadání po formální stránce, avšak s několika výhradami. Autor sice úspěšně implementoval požadovaný algoritmus ve dvou verzích a provedl základní ověření jeho funkčnosti, nicméně práce postrádá hlubší analytický vhled do samotného procesu implementace, praktických rozhodnutí i odlišností oproti původní verzi v MATLABu. Teoretická část je rozsáhlá, ale místy zbytečně obecná, a ne vždy přímá vzhledem k zadání. Praktická část je naopak stručná a často odkazuje na zdrojový kód bez podrobného rozboru. Přes dílčí nedostatky v odborné i jazykové rovině je práce zpracována v odpovídající kvalitě a svědčí o schopnosti autora řešit zadaný problém.

Otázky k obhajobě:

- 1) Jakým způsobem byla ověřena správnost nové implementace algoritmu QPmR v jazyce C++ a jak by šlo toto ověření doplnit pro zvýšení důvěryhodnosti výsledků?
- 2) Jaké změny by bylo vhodné provést v přístupu k datové výměně parametrů a výsledků aplikace, aby byla lépe kompatibilní s jinými programy a usnadnila automatizované zpracování?
- 3) Jaké algoritmy pro hledání kontur byly implementovány v C++ aplikaci (mimo prostředí gnuplot)?
- 4) Byl použit nějaký typ paralelismu pro urychlení ve výsledné implementaci? Pokud ne, kterou část byste tímto způsobem doporučil urychlit? Popište možná úskalí.

Celkové hodnocení práce:

Známku uvede oponent dle svého uvážení dle klasifikační stupnice ECTS:

A – výborně, B – velmi dobře, C – dobře, D – uspokojivě, E – dostatečně, F – nedostatečně.

Stupeň F znamená též „nedoporučuji práci k obhajobě“.

Předloženou diplomovou práci doporučuji k obhajobě a navrhuji hodnocení

C - dobře.

V případě hodnocení stupněm „F – nedostatečně“ uveďte do připomínek a slovního vyjádření hlavní nedostatky práce a důvody tohoto hodnocení.

Datum

9. 6. 2025

Podpis oponenta diplomové práce