České vysoké učení technické v Praze
Fakulta elektrotechnická
Katedra řídící techniky

Bakalářská práce

Možnosti využití PAS při výpočtech s n–D polynomy

Praha, 2007 Jiří Mikolášek

České vysoké učení technické v Praze, fakulta elektrotechnická

Katedra řídicí techniky

Školní rok: 2006/2007

Zadání bakalářské práce

Student: Jiří Mikolášek

Obor: Kybernetika a měření

Název tématu: Možnosti využití PAS při výpočtech s *n*-D polynomy

Zásady pro vypracování:

1. Seznamte se s pojmem počítačový algebraický systém (PAS).
2. Vypracujte přehled volně dostupných PAS, s informacemi např. o podporovaných operačních systémech, oblastech výpočtů, pro které je PAS určen, atp.
3. PAS, které se Vám zdají být nejlepší, si po dohodě s vedoucím práce nainstalujte a vyzkoušejte. Otestujte algoritmy výpočtu největšího společného dělitele polynomů o více proměnných, o výsledcích testů napište zprávu.

Seznam odborné literatury: Dodá vedoucí práce

Vedoucí bakalářské práce: Ing. Peter Augusta

Datum zadání bakalářské práce: zimní semestr 2006/07

Termín odevzdání bakalářské práce: 15. 8. 2007

Prof. Ing. Michael Šebek, DrSc.
vedoucí katedry

Prof. Ing. Zbyněk Škvor, CSc.
děkan

V Praze, dne 6. 3. 2007

Prohlášení

Prohlašuji, že jsem svou bakalářskou práci vypracoval samostatně a použil jsem pouze podklady (literaturu, projekty, SW atd.) uvedené v přiloženém seznamu.

Nemám závažný důvod proti užití tohoto školního díla ve smyslu § 60 Zákona č.121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

V Praze dne: 23. srpna 2007

podpis

Poděkování

Děkuji především vedoucímu bakalářské práce Petru Augustovi

Anotace: Práce se zabývá počítačovou algebrou a využitím počítačových algebraických systémů při výpočtech největšího společného dělitele.

Poskytuje přehled volně dostupných počítačových algebraických systémů, nároky na operační systém v němž je lze provozovat, případně hardware, přehled webových stránek systémů a možnosti využití počítačových algebraických systémů při výpočtu největšího společného dělitele polynomů s více proměnnými.

Dále se zaměřuje na použití algoritmů pro výpočet největšího společného dělitele v teorii řízení, hlavně v oblasti algebraických (polynomiálních) metod řízení při řešení diofantické rovnice.

Největší prostor byl vyhrazen pro testování algoritmů největšího společného dělitele ve vybraných počítačových algebraických systémech z hlediska rychlosti při výpočtu největšího společného dělitele polynomů stupně 1 až 5000 s počtem proměnných 1 až 5 s ohledem na časovou náročnost a porovnání systémů na základě testů a grafické znázornění časových výsledků.

Annotation: This thesis deals with computer algebra and using them to compute greatest common divisor.

It offers an overview of free available computer algebra systems, requirements to operating system, eventually hardware, an overview of web sites these systems and possibilities for using them to compute greatest common divisor of multivariable polynomials.

It is focusing on using algorithms to compute greatest common divisor in control theory mainly in way algebraic (polynomial) methods of control solution to diophantine equations.

Main contens is testing algorithms using to compute greatest common divisor, testing some computer algebra systems in term of rate during compute greatest common divisor of polynomials degree 1 up to 5000 and 1 up to 5 variables and comparison systems on the basis of tests and graphical illustration.

Obsah

1 Počítačová algebra a počítačový algebraický systém **2**
1.1 Počítačová algebra . . . 2
1.2 Symbolické operace . . . 3
1.2.1 Historie stručně . . . 4
1.3 Počítačové algebraické systémy . . . 5
1.3.1 Vývoj počítačových algebraických systémů . . . 5
1.4 Algoritmy pro symbolické výpočty . . . 6
1.4.1 Algebraické struktury . . . 6
1.4.2 Aritmetika v základních algebraických oborech . . . 6
1.4.3 Okruhy, tělesa a polynomy . . . 6
1.4.4 Největší společný dělitel . . . 7
1.4.4.1 Největší společný dělitel celých čísel . . . 7
1.4.4.2 Největší společný dělitel polynomů s racionálními koeficienty . . . 8
1.4.4.3 Největší společný dělitel polynomů s celočíselnými koeficienty . . . 9
1.4.4.4 Další používané algoritmy . . . 11
1.4.4.5 Ukázky možností naprogramování procedur pro výpočet gcd celých čísel a jednoduchých polynomů v systému Maple . . . 11

2 Největší společný dělitel a teorie řízení **13**
2.1 Lineární diofantická rovnice a její řešení . . . 13
2.1.1 Řešení diofantické rovnice metodou neurčitých koeficientů . . . 14
2.2 Řešení obecné n-D rovnice . . . 14
2.3 Další možnosti využití GCD v teorii řízení . . . 14

3 Přehled počítačových algebraických systémů **15**

4 Testování algoritmů GCD ve vybraných PAS **23**
4.1 Metodika testování . . . 23
4.1.1 Hardwarové a softwarové vybavení . . . 23
4.1.2 Testovací polynomy . . . 24
4.1.3 Průběh testů . . . 24
4.2 Maple 9.52 . . . 24
4.2.1 Generování testovacích polynomů . . . 26
4.2.2 Možnosti výpočtu největšího společného dělitele . . . 27
4.2.3 Výsledky testů největšího společného dělitele . . . 28
4.3 Maxima . . . 28
4.3.1 Historie . . . 30
4.3.2 Generování testovacích polynomů . . . 30

4.3.3 Možnosti výpočtu největšího společného dělitele 30
4.3.4 Výsledky testů největšího společného dělitele 30
4.4 SINGULAR 31
4.4.1 Generování testovacích polynomů 32
4.4.2 Možnosti výpočtu největšího společného dělitele 33
4.4.3 Výsledky testů největšího společného dělitele 33
4.5 Giac/Xcas 33
4.5.1 Generování testovacích polynomů 35
4.5.2 Možnosti výpočtu největšího společného dělitele 35
4.5.3 Výsledky testů největšího společného dělitele 36
4.6 CoCoA 37
4.6.1 Generování testovacích polynomů 37
4.6.2 Možnosti výpočtu největšího společného dělitele 38
4.6.3 Výsledky testů největšího společného dělitele 38

5 Závěr a porovnání jednotlivých systémů 40

Úvod

Výkon současných počítačů je na takové úrovni, že matematické výpočty lze provádět na téměř každém počítači a téměř ve všech počítačových algebraických systémech v přijatelném časovém rozmezí. Pokud je potřeba provádět výpočty bez zaokrouhlování a používat symbolické operace nad rozsáhlými strukturami je to práce právě pro systémy počítačové algebry. Většinu z operací, které jsou obsaženy v systémech počítačové algebry, můžeme provádět i ručně, avšak s mnohem větším úsilím a stráveným časem. Jedním z používaných algoritmů je i algoritmus pro výpočet největšího společného dělitele. Algoritmus pro výpočet největšího společného dělitele je ve většině systémů používán přímo skrytě při operacích s polynomy a racionálními funkcemi–racionální funkce se reprezentují jako podíl polynomů ve zkrácené formě. Zaměříme se hlavně na použití největšího společného dělitele v teorii řízení a řešení tzv. diofantické rovnice. Největší společný dělitel je základním výpočetním mechanismem pro řadu dalších algoritmů, a to nejen v teorii řízení, ale i v řadě dalších.

Tato práce by měla čtenáři poskytnout přehled počítačových algebraických systémů běžnému uživateli dostupných a dále porovnání některých těchto systémů z hlediska rychlosti výpočtů algoritmů největšího společného dělitele. V oblasti počítačové algebry dochází k vývoji stejně jako v jiných oborech, proto nelze tento přehled brát jako neměnný a vždy je vhodné použít některý informační zdroj a ověřit, zda daný systém ještě existuje, není k dispozici nová verze nebo se neobjevil nějaký lepší. Tento přehled lze nalézt například v [9] nebo přímo na webových stránkách systémů.

V první části práce je nastíněna teorie počítačové algebry a počítačových algebraických systémů a definice některých použitých termínů, které budou potřebné pro další části. Některé termíny jsou uvedeny pouze jako přehled bez vysvětlení dalších souvislostí, neboť není možné obsáhnout celou teorii a ani to nebylo předmětem této práce. Více informací lze nalézt v odborné literatuře. Dále je uvedena kapitola pojednávající o využití algoritmu největšího společného dělitele v teorii řízení a řešení diofantické rovnice.

Další část je věnována přehledu počítačových algebraických systémů, které využívají algoritmů největšího společného dělitele. Jsou zde uvedeny hlavně informace ohledně vývoje a možností daného systému, protože to, jak se systém ovládá a další technické informace je možné nalézt v nápovědě k jednotlivým systémům nebo na jejich webových stránkách, kde lze ve většině případů nalézt množství příkladů jak se systémem zacházet.

Poslední část se zaměřuje na testování vybraných počítačových algebraických systémů a jejich porovnání z hlediska rychlosti výpočtu největšího společného dělitele. Výsledky testů jsou použitelné pokud se čtenář bude zabývat právě výpočtem největšího společného dělitele a bude hledat nejrychlejší systém počítačové algebry.

Kapitola 1

Počítačová algebra a počítačový algebraický systém

Tato kapitola podává přehled možností a definici počítačové algebry a počítačových algebraických systémů. Vlastnosti jsou vypsány pouze v bodech, neboť je dostupno množství literatury na toto téma a předmětem této práce nebylo řešit teoretické problémy. Zájemce o tuto problematiku může další nalézt např. v [2, 13, 11].

1.1 Počítačová algebra

Počítačová algebra (někdy též nazývaná algebraické výpočty nebo symbolické manipulace) je obor vědeckého počítání, který vyvíjí, analyzuje, implementuje a používá algebraické algoritmy.

číselné	symbolické
$\frac{2}{6} \rightarrow 0.33333$	$\frac{2}{6} \rightarrow \frac{1}{3}$
$2+3=5$	$x+2x=3x$
$\cos(3.14159) \rightarrow -0.999999$	$\cos(\Pi) \rightarrow -1$
	$\sin(2x) = 2\sin x \cos x$
	$\frac{dx^2}{dx} \rightarrow 2x$
$\int_0^{\frac{1}{2}} \frac{x}{x^2-1} \rightarrow 0.1438$	$\int \frac{x}{x^2-1} \rightarrow \frac{\ln\lvert x^2-1\rvert}{2}$
	$a^2-b^2 \rightarrow (a+b)(a-b)$
numerické	algebraické
vyhodnocení	zjednodušení

Tabulka 1.1: Srovnání číselných a symbolických výpočtů.

Symbolické výpočty

- výpočty s čísly s absolutní přesností, bez zaokrouhlování,
- výpočty se symboly, neznámými (x,y),
- výpočty s funkcemi (sin, cos, f),

- manipulace se vzorci,
- symbolické operace (derivování, integrování, faktorizace, ...).

Použití algebraických výpočtů

- na rozdíl od člověka počítač chyby nedělá (je-li vstupní informace zadána správně dostaneme správný výsledek),
- ve vědeckých problémech je často potřeba zpracovávat velice rozsáhlé algebraické výrazy (statisíce a více členů) nebo provádět dlouhé analytické operace,
- některé algoritmy člověk ručně s tužkou na papíře není schopen provádět (např. faktorizace[1], eliminace kvantifikátorů[2]).

1.2 Symbolické operace

Symbolické operace slouží pro řešení problémů v oblasti počítačové algebry. Tyto operace zahrnují například:

- zjednodušování, automatické zjednodušování s odhadem a zjednodušování s omezením,
- substituce symbolů, funkčních nebo numerických hodnot,
- přepsání trigonometrických funkcí do exponenciální podoby,
- parciální a totální derivace,
- parciální a celková faktorizace,
- řešení lineárních a některých nelineárních rovnic v několika dimenzích,
- řešení diferenciálů a diferenciálních rovnic,
- řešení limit,
- určité a neurčité integrály, výcerozměrové integrály,
- integrální transformace,
- libovolně přesné numerické operace,
- řady operací za sebou (rozvoje a součty),
- maticové operace (součin, inverzní matice, ...),
- zobrazení matematických výrazů ve 2–D formě, často se používá sázecí systémy jako je TEX (nebo také Prettyprint).

Dalšími vlastnostnostmi mohou být:

[1] Jako faktorizace se v matematice a jejích aplikacích označuje problém rozložení čísla na součin menších čísel, v nejčastější podobě pak rozklad celého čísla na součin prvočísel.

[2] Kvantifikovaný vzorec v prenex tvaru v oboru reálných čísel má tvar $G = (Q_{f+1}x_{f+1})...(Q_r x_r)F(x_1, ..., x_r)$. Eliminace kvantifikátorů je proces, který z kvantifikovaného vzorce G najde nekvantifikovaný vzorec H ve volných proměnných $x_1, ..., x_f$ ekvivalentní vzorci G.

- počítání s velkou přesností (od 21/3 do 10000 desetiných míst),
- přídavné knihovny pro použití v aplikované matematice (fyzikální přídavné knihovny pro fyzikální výpočty),
- sestrojování grafů a parametrické kreslení funkcí ve dvou a třech dimenzích a jejich animace,
- rozhraní (API – Aplication Interface) pro připojování k externím programům (databáze), nebo používání různých programovacích jazyků,
- kreslení grafů, schémat a digramů,
- manipulace s textovými řetězci (porovnávání a vyhledávání),
- statistické výpočty,
- důkazy a ověření,
- zpracování signálů jako obrázků,
- syntéza zvuku,
- mnoho PAS také obsahuje programovací jazyk vyšší úrovně, který dovoluje uživatelům implementovat vlastní algoritmy, programovací jazyk může být podobný jako tradiční programovací jazyky,
- při studování algoritmů užívaných pro počítačové algebraické systémy mluvíme o počítačové algebře, symbolických výpočtech, algebraických výpočtech nebo, méně všeobecně, symbolických manipulacích, symbolickém zpracování, symbolické matematice nebo symbolické algebře,
- doba běhu numerických programů implementovaných v typických PAS je obvykle delší než v rovnocenných programech implementovaných v systémech jako je MATLAB, GNU Octave nebo přímo v C, neboť jsou naprogramovány pro symbolické operace a tudíž nemohou pro většinu funkcí používat strojové numerické operace přímo.

1.2.1 Historie stručně

- původně vznikl obor hlavně pro potřeby fyziky,
- 1955–první programy pro derivování vzorců,
- 1965–první obecně použitelné systémy programů pro práci s algebraickými výrazy,
- 1975–nový vědecký obor s vlastními konferencemi a časopisy,
- 1990–obecné rozšíření integrovaných matematických systémů do téměř všech vědních oborů.

1.3 Počítačové algebraické systémy

Počítačový algebraický systém (PAS) je software, který umožňuje symbolické matematické výpočty. Funkce PAS spočívá v manipulaci s matematickými výrazy v symbolické formě.

1.3.1 Vývoj počítačových algebraických systémů

Jméno systému	Rok	Příbuzné systémy	e-mail adresa, tel.
ALPAK	1964	ALTRAN	(Bell Labs)
ALTRAN	1968		(Bell Labs)
FORMULA (Algol).			
FORMAC		FORMAC (PL/I)	(IBM)
FORMAC (PL/I)			(IBM)
MATHLAB (DECUS)	1968	MACSYMA	(DEC)
CAMAL			
REDUCE	1968		reduce-netlib@rand.org
MACSYMA	1970	Symbolics Macsyma, ALJABR, ParaMacs, VAXIMA, DOE-Macsyma	(See Below)
SchoonShip	1971		archive.umich.edu (FTP)
muMath	1979	Derive	
VAXIMA	1980		(312) 972-7250
SMP	1982	Mathematica	NOT ON MARKET
Symbolics MACSYMA	1983		macsyma-service@symbolics.com
DOE-Macsyma	1984	ALJABR	gcook@llnl.gov
Maple	1985		wmsi@daisy.waterloo.edu
MathCAD	1985(?)	Mathcad	1-800-MATHCAD
Powermath	1985		NOT ON MARKET
REDUCE/PC	1986		reduce-netlib@rand.org
Derive	1988		(Soft Warehouse Inc.)
Mathematica	1988		info@wri.com
Theorist	1988		(415) 543-2252 (Prescience Corp)
PARI	1988(?)		ftp to math.ucla.edu
FORM	1989		form@can.nl
MACSYMA/PC	1989		macsyma-service@symbolics.com
ALJABR	1991		aljabr@fpr.com
Mathcad	1991		1-800-MATHCAD
SymbMath	1991		chen@deakin.oz.au
Axiom	1991		(708) 971-2337
ParaMacs	1991		lph@paradigm.com
SIMATH	1992		marc@math.uni-sb.de

Tabulka 1.3: Tabulka od Briana Evanse <evans@eedsp.gatech.edu> [2].

1.4 Algoritmy pro symbolické výpočty

1.4.1 Algebraické struktury

Požadavky

- přesná reprezentace algebraických struktur,
- přesná aritmetika $(+,-,\cdot,\div)$ s algebraickými strukturami,
- další analytické operace s těmito strukturami (derivování, integrování, ...).

Reprezentace algebraických struktur

Abychom mohli s algebraickými strukturami pracovat na počítači musíme je reprezentovat datovými strukturami, reprezentace je velice důležitá, protože na ní často závisí efektivnost implementovaných algoritmů.

1.4.2 Aritmetika v základních algebraických oborech

Pod aritmetikou rozumíme operace sčítání, odčítání, násobení a dělení.

Zjednodušování

- většina operací v počítačové algebře je jistou formou zjednodušení,
- otázka pro reprezentaci algebraických výrazů,
- otázkou zůstává, který tvar je jednodušší (záleží na tom, jak daný výraz budeme zpracovávat)

$$(a+b)^2 \Leftrightarrow a^2 + 2ab + b^2$$

1.4.3 Okruhy, tělesa a polynomy

Definice. [9] *Grupou* $(G, \circ)$ se nazývá neprázdná množina G uzavřená vzhledem k binární operaci $\circ$, takovou, že $\circ$ je asociativní operací, v G existuje jediný *neutrální prvek* e a pro každý prvek $a \in G$ existuje prvek $a^{-1} \in G$ tak, že $a \circ a^{-1} = a^{-1} \circ a = e$.

Grupu G nazýváme *abelovskou* (*komutativní*), je-li navíc operace $\circ$ komutattivní vzhledem ke grupě G.

Okruhem $(R, +, \times)$ nazýváme neprázdnou množinu R uzavřenou na dvě binární operace $+$ a $\times$, která je komutativní grupou vzhledem k operaci $+$, operace $\times$ je vzhledem k prvkům R asociativní a v R existuje vzhledem k operaci $\times$ neutrální prvek (*jednotkový prvek*), splňující navíc levý a pravý distributivní zákon.

Komutativním okruhem nazveme okruh, vzhledem ke kterému je i operace $\times$ komutativní.

Definice. [9] Pro libovolný komutativní okruh R definujeme okruh polynomů $R[x]$ nad R jako množinu formálních výrazů

$$P(x) = a_n x^n + a_{n-1} x^{n-1} + ... + a_1 x + a_0,$$

kde n je libovolné přirozené číslo a $a_i \in R$ pro $0 \leq i \leq n$. Součet polynomů $P(x) = \sum_{k=0}^{n} b_n x^n$ definujeme předpisem

$$P(x) + Q(x) = \sum_{k=0}^{n} r_k x^k,$$

kde $c_k = a_k + b_k$ pro $0 \leq k \leq \max(m, n)$, přitom nedefinované hodnoty a_k nebo b_k uvažujeme jako nulové. Součin polynomů $P(x)$ a $Q(x)$ definujeme vztahem

$$P(x) \cdot Q(x) = \sum_{k=0}^{m+n} c_k x^k,$$

kde $c_k = \sum_{i+j=k} a_i b_j$. Neutrálním prvkem je tzv. *nulový polynom*, který má všechny koeficienty nulové (značíme 0), jednotkovým prvkem pak polynom, který má jediný nenulový koeficient $a_0 = 1$.

Stupněm $\deg(P(x))$ nenulového polynomu $P(x) = \sum_{k=0}^{m} a_k x^k$ nazýváme největší přirozené číslo n takové, že $a_n \neq 0$.

Je důležité si uvědomit, že okruhem R v předcházející definici může být rovněž okruh polynomů. V tomto případě mluvíme o *okruhu polynomů více proměnných* a místo $((R[x_1])[x_2]...)[x_n]$ píšeme $R[x_1, x_2, ..., x_n]$ (nebo jen $R[x]$).

1.4.4 Největší společný dělitel

Algoritmus pro výpočet GCD (greatest common divisor) je důležitý algoritmus užívaný řadou dalších algoritmů. Zaměříme se hlavně na jednoduché ukázky výpočtů největšího společného dělitele pomocí Euklidova algoritmu. Další používané algoritmy jsou pouze zmíněny bez hlubšího rozboru definic, které lze nalézt např. v [9].

Na konci kapitoly je ukázka použití Euklidova algoritmu naprogramovaného v systému Maple.

1.4.4.1 Největší společný dělitel celých čísel

Mějme čísla a, $b \in Z$, pro která platí $a \geq b$. Potom definujeme jejich podíl $\mathrm{quot}(a, b)$ a zbytek po dělení $\mathrm{rem}(a, b)$ jako celá čísla, pro něž

$$a = \mathrm{quot}(a, b)b + \mathrm{rem}(a, b)$$

a $0 \leq \mathrm{rem}(a, b) < b$. Označme $\gcd(a, b)$ největší společný dělitel čísel a, b. Potom pokud $\mathrm{rem}(a, b)$ je nenulové platí

$$\gcd(a, b) = \gcd(b, \mathrm{rem}(a, b)),$$

$\gcd(a, b)$ totiž dělí jak a tak b, dělí také b $\mathrm{quot}(a, b)$ a musí tedy dělit i $\mathrm{rem}(a, b)$. Na této identitě je založen *Euklidův algoritmus* pro výpočet největšího společného dělitele. *Euklidův algoritmus* [2] pro výpočet největšího společného dělitele dvou celých čísel:

```
gcd:=GCDI(a,b)
 [a, b jsou celá čísla
  použité algoritmy:
   rem(a,b) - zbytek po dělení čísla a číslem b ]
```

```
1. if a < b then
    r:=a;
    a:=b;
    b:=r;
   fi
2. while not(b=0) do
    r:=rem(a,b);
    a:=b;
    b:=r;
   do
3. gcd:=a;
```

1.4.4.2 Největší společný dělitel polynomů s racionálními koeficienty

Mějme polynomy z $Q[x]$ v jedné proměnné x s racionálními koeficienty. Stupeň polynomu $a(x)$ označíme jako $\deg(a(x))$.

Pro dva polynomy $a(x)$, $b(x)$ z $Q[x]$, pro které platí $\deg(a(x)) \geq \deg(b(x))$, definujeme jejich podíl $\mathrm{quot}(a(x), b(x))$ a zbytek po dělení $\mathrm{rem}(a(x), b(x))$ jako polynomy z $Q[x]$, pro které platí

$$a(x) = \mathrm{quot}(a(x), b(x))b(x) + \mathrm{rem}(a(x), b(x)),$$

přitom $0 \leq \deg(\mathrm{rem}(a(x), b(x))) < \deg(b(x))$. Stejně jako pro celá čísla platí pro největší společný dělitel polynomů

$$\gcd(a(x), b(x)) = \gcd(b(x), \mathrm{rem}(a(x), b(x))),$$

Euklidův algoritmus pro výpočet největšího společného dělitele dvou polynomů s racionálními koeficienty [2]:

```
gcd:=GCDPQ(a(x),b(x))
        [a, b jsou polynomy z Q[x]
         použité algoritmy:
           deg(a)   - stupeň polynomu a
           rem(a,b) - zbytek po dělení polynomu a polynomem b]

      1. if deg(a) < deg(b) then
            r:=a;
            a:=b;
            b:=r;
         fi

      2. while not(b=0) do
           r:=rem(a,b);
           a:=b;
           b:=r;
         do
      3. gcd:=a;
```

- algoritmus vyžaduje řadu výpočtů největšího společného dělitele dvou celých čísel při výpočtech s racionálními čísly, přičemž celá čísla mohou být značně velká,

- tyto výpočty jsou časově náročné, algoritmus není efektivní,
- výpočtů s racionálními čísly se můžeme zbavit tím, že budeme počítat pouze v oboru polynomů s celočíselnými koeficienty.

Příklad gcd v Q[x] [2]

Máme spočítat $\gcd(a, b)$, kde

$$a = x^8 + x^6 - 3x^4 - 3x^3 + 8x^2 + 2x - 5$$

$$b = 3x^6 + 5x^4 - 4x^2 - 9x + 21$$

použitím algoritmu GCDPQ dostaneme následující zbytky r

$$r_1 = -\frac{5}{9}x^4 + \frac{1}{9}x^2 - \frac{1}{3}$$

$$r_2 = -\frac{117}{25}x^2 - 9x + \frac{441}{25}$$

$$r_3 = \frac{233150}{6591}x - \frac{102500}{2197}$$

$$r_4 = \frac{1288744821}{543589225}$$

čili polynomy a, b jsou nesoudělné, jejich největším společným dělitelem je 1.

1.4.4.3 Největší společný dělitel polynomů s celočíselnými koeficienty

Mějme polynomy z $Z[x]$ v jedné proměnné x s celočíselnými koeficienty. Pseudodělení polynomů $a(x)$, $b(x)$, pro které $\deg(a(x)) \geq \deg(b(x))$ je definováno

$$\mathrm{lcof}(b(x))^{m-n+1}a(x) = \mathrm{pquot}(a(x), b(x))b(x) + \mathrm{prem}(a(x), b(x)),$$

$\mathrm{cont}(a(x))$ je největší společný dělitel všech koeficientů polynomu a $\mathrm{pp}(a(x))$ je primitivní část polynomu a. Euklidův algoritmus posloupnosti polynomiálních zbytků (polynomial reminder sequence) [2]:

```
gcd:=GCDPRS(a(x),b(x))
        [předpokládá, že stupeň polynomu a je větší nebo roven stupni
         polynomu b, tj. deg(a) >= deg(b)
         použité algoritmy:
           prem(a,b) - pseudo-zbytek po dělení polynomu a polynomem b
           pp(a)     - primitivní část polynomu a
           gcdi(j,k) - největší společný dělitel dvou celých čísel ]

       1. A:=pp(a);
```

```
   B:=pp(b);
2. while not B =0 do
     r:=prem(A,B);
     A:=B;
     B:=r;
   od;
3. gcd:=gcdi(cont(a),cont(b)) pp(A);
```

- koeficienty a mezivýsledky rostou velice rychle, neefektivní algoritmus,
- je možné na každém kroku počítat primitivní část zbytku a pokračovat ve výpočtu s primitivní částí; výpočet primitivní části však vyžaduje výpočet největších společných dělitelů koeficientů, celých čísel, který je pro velká čísla časově náročný,
- modulární přístup; výpočty prováděny modulo prvočíslo p :
  - aplikovat homomorfismus[3] na koeficienty,
  - spočítat největší společný dělitel transformovaných polynomů modulo p,
  - použít Čínský zbytkový algoritmus[4] pro rekonstrukci koeficientů v největším společném děliteli.

Příklad gcd Z[x] [2]

Máme spočítat $\gcd(a, b)$, kde

$$a = x^8 + x^6 - 3x^4 - 3x^3 + 8x^2 + 2x - 5$$

$$b = 3x^6 + 5x^4 - 4x^2 - 9x + 21$$

použitím algoritmu GCDPRS dostaneme následující pseudo-zbytky r.

pseudo zbytek	primitivní část
$r_1 = -15x^4 + 3x^2 - 9$	$-5x^4 + x^2 - 3$
$r_2 = 15795x^2 + 30375x - 59535$	$13x^2 + 25x - 49$
$r_3 = 1254542875143750x^2 - 1654608338437500$	$4663x - 6150$
$r_4 = 12593338795500743100931141992187500$	1

[3]Homomorfismus je zobrazení z jedné algebraické struktury do jiné stejného typu, které zachovává veškerou důležitou strukturu.

Každý typ algebraické struktury má svůj typ homomorfismu. Obecně je homomorfismus zobrazení $\phi : A \to B$ mezi dvěma algebraickými strukturami stejného typu takové, že pro každou definovanou operaci f a pro všechna x_i v A platí $\phi(f_A(x_1, ..., x_n)) = f_B(\phi(x_1), ..., {}_n))$.

[4]Čínská věta o zbytcích (také známa jako Čínská věta o zbytku nebo Čínská zbytková věta) je matematické tvrzení z modulární aritmetiky, tj. oblasti nacházející se na pomezí elementární teorie čísel a algebry. Pojednává o vlastnostech čísel v okruzích kongruence modulo n (okruhy Z_n). Využívá se v algoritmech pro zpracování velkých čísel nebo v kryptografii.

1.4.4.4 Další používané algoritmy

Heuristický algoritmus

je založen na podobném principu jako *modulární metody*. Jsou-li $a(x)$, $b(x)$ například polynomy jedné proměnné nad okruhem Z celých čísel, pak se můžeme domnívat, že $\gcd(a(\xi), b(\xi))$ je nějak svázán s $\gcd(a(x), b(x))$, kde $\xi \in Z$ je vhodné číslo. Dá se dokonce předpokládat, že by bylo možné ze znalosti $\gcd(a(\xi), b(\xi))$ zkonstruovat největšího společného dělitele polynomů $a(x)$, $b(x)$. Tento předpoklad se ukazuje pravdivým při vhodné volbě čísla ξ.

Věta. Nechť $a(x), b(x) \in Z[x_1, x_2, ..., x_k]$ jsou nenulové polynomy, které jsou primitivní vzhledem k Z. Nechť dále $\xi \in Z$ je číslo splňující

$$\xi > 1 + 2\min(\|a(x)\|, \|b(x)\|).$$

Položme $\gamma(x) = \gcd(a(x_1, ..., x_{k-1}, \xi), b(x_1, ..., x_{k-1}, \xi)) \in Z[x_1, ..., x_{k-1}]$ a jako $c(x)$ označme polynom vytvořený pomocí ξ- adické[5] reprezentace $\gamma(x)$. Označíme-li jako $\mathrm{pp}(c(x))$ polynom vzniklý vydělením polynomu $c(x)$ největším společným dělitelem jeho celočíselných koeficientů, pak je $\mathrm{pp}(c(x))$ největším společným dělitelem polynomů $a(x), b(x)$ právě tehdy, když $\mathrm{pp}(c(x)) \mid a(x)$ a $\mathrm{pp}(c(x)) \mid b(x)$.

Henselův (EZGCD) algoritmus

Algoritmy *modulárního* (resp. pravděpodobnostního) *typu* bývají zpravidla úspěšnější než *algoritmy založené na pseudodělení polynomů* a *zbytkových posloupnostech*. Ve většině výpočtů s malým počtem proměnných dobře vyhoví *heuristická metoda* a v případě, že tato selže, je možné aplikovat *modulární metodu*. Jak je však z tvaru této metody vidět, v případě mnoha proměnných, jde-li navíc o řídké polynomy, provádí se zde příliš mnoho (jednoduchých) výpočtů v euklidovských okruzích. Jedním z možných překlenutí tohoto problému je *algoritmus řídkého moduláriho GCD*. Jinou metodou (která je v Maple použita kdykoliv selže heuristická metoda) je algoritmus, který bývá nazýván *Extended Zassenhaus GCD* nebo *Henselův algoritmus*.

1.4.4.5 Ukázky možností naprogramování procedur pro výpočet gcd celých čísel a jednoduchých polynomů v systému Maple

Rekurzivní procedura pro výpočet gcd dvou celých čísel.

```
> mygcd := proc(a::nonnegint,b::nonnegint)
> # assume a >= b.
>   if b = 0 then return a fi;
>   return mygcd(b, a mod b);
>  end;
```

Výpis po krocích.

```
trace(mygcd);
> mygcd(8,4);
{--> enter mygcd, args = 8, 4
{--> enter mygcd, args = 4, 0
<-- exit mygcd (now in mygcd) = 4}
```

[5]Číselná soustava o základu Z (Z-adická soustava) má Z číslic: 0,1, ..., Z-1.

```
<-- exit mygcd (now at top level) = 4}
```

Rozšířený Euklidův algoritmus.

```
> mygcdex := proc(a::nonnegint,b::nonnegint,x::name,y::name)
>   local a1,a2,a3,x1,x2,x3,y1,y2,y3,q;
>   a1 := a;  a2 := b;
>   x1 := 1;  y1 := 0;
>   x2 := 0;  y2 := 1;
>   while (a2 <> 0) do
>     a3 := a1 mod a2;
>     q := floor(a1/a2);
>     x3 := x1 - q*x2;
>     y3 := y1 - q*y2;
>     a1 := a2;  a2 := a3;
>     x1 := x2;  x2 := x3;
>     y1 := y2;  y2 := y3;
>   od;
>   x := x1;  y := y1;
>   return a1;
> end;
```

Zadání polynomu o dvou neznámých.

```
> mygcdex(3,6,'x','y');
3
> x;
0
> y;
1
> 3*x + 6*y;
3
```

Rekurzivní procedura pro počítání gcd.

```
> mygcdexrec := proc(a::nonnegint,b::nonnegint,x::name,y::name)
> #  a >= b >= 0.
> local xp, yp, d, q, r;
>   if b = 0 then
>     x := 1;  y := 0;  return a;
>   else
>     r := irem(a,b,q);
>     d := mygcdexrec(b,r,xp,yp);
>     x := yp;  y := xp - q*yp;  return d;
>   fi;
>  end;
```

Zadání polynomu o dvou neznámých:

```
> mygcdexrec(3,6,'x','y');
```

Kapitola 2

Největší společný dělitel a teorie řízení

Při návrhu řízení často pracujeme s lineárními polynomiálními rovnicemi. Při použití algebraických (polynomiálních) metod řízení, které vycházejí z vnějšího popisu systému můžeme chápat systém jako algebraický objekt. Řešení algebraických rovnic je důležité právě při návrhu řízení algebraickými (polynomiálními) metodami, kdy je systém popsán jako podíl dvou polynomů (pro MIMO systémy polynomiálním maticovým zlomkem) a regulátor je potom popsán polynomy $X(p)$ a $Y(p)$, které jsou řešením tzv. diofantické rovnice (2.1).

Základní *diofantická rovnice* má tvar

$$A(p)X(p) + B(p)Y(p) = C(p) \tag{2.1}$$

kde A, B, C jsou zadané polynomiální rovnice s proměnnou p, zatímco X a Y jsou neznámé polynomiální matice.

Rovnice může být přepsána do tvaru

$$X(p)A(p) = B(p) \tag{2.2}$$

Polynomiální rovnice se typicky řeší pomocí Euklidova algoritmu nebo metodou neurčitých koeficientů.

2.1 Lineární diofantická rovnice a její řešení

Nechť A, B, C jsou dané polynomy stupňů $\delta A, \delta B, \delta C$. Pak lineární diofantickou rovnicí rozumíme rovnici (2.1) kde X, Y jsou hledané polynomy. Rovnice (2.1) má řešení X, Y právě tehdy, jestliže společný dělitel $D = (A, B)$ dělí polynom C. Má-li rovnice (2.1) řešení, má jich pak nekonečně mnoho. Nechť X_p, Y_p je libovolné partikulární řešení rovnice (2.1), pak obecné řešení je:

$$X = X_p + \frac{B}{(A,B)}F, Y = Y_p - \frac{A}{(A,B)}F$$

kde F je libovolný polynom. Úloha (2.1) se zpravidla řeší nalezením některého partikulárního řešení, např. metodou neurčitých koeficientů Euklidovým algoritmem nebo jinými metodami, a dále nalezením koeficientů polynomů vytvářející neurčitost obecného řešení.

2.1.1 Řešení diofantické rovnice metodou neurčitých koeficientů

Partikulární řešení $X = X_p, Y = Y_p$ rovnice (2.1) je možno určit přímo porovnáním koeficientů u stejných mocnin na levé a pravé straně rovnosti (2.1). Stupně hledaných polynomů je třeba určit následovně:

$\delta X = \delta B - 1, \delta Y = \delta A - 1$ pro $\delta A + \delta B > \delta C$

$\delta X = \delta B - 1, \delta Y = \delta C - \delta B$ pro ostatní případy

Soustava rovnic má řešení právě tehdy, jestliže společný dělitel $D = (A, B)$ dělí polynom C. Má jediné řešení, jestliže společný dělitel polynomů A, B je roven 1.

2.2 Řešení obecné n-D rovnice

V některých případech je nutné řešit rovnice s obecně n proměnnými, potom řešíme rovnici ve tvaru

$$A(p_1, ..., p_n)X(p_1, ..., p_n) + B(p_1, ..., p_n)Y(p_1, ..., p_n) = C(p_1, ..., p_n) \tag{2.3}$$

kde A, B, C jsou zadané polynomiální rovnice s proměnými $p_1, ..., p_n$, zatímco X a Y jsou neznámé polynomiální matice.

Věta (řešitelnost). [12] gcd(a,b)|c *kdykoliv* (2.3) *je řešitelná.*

2.3 Další možnosti využití GCD v teorii řízení

Existují i další možnosti využití výpočtů největšího společného dělitele. Pro doplnění lze uvést použití GCD například pro faktorizaci n–D polynomiálních matic.

Kapitola 3

Přehled počítačových algebraických systémů

V této kapitole se zaměříme na přehled a stručný popis počítačových algebraických systémů, které lze v současnosti používat. Ve většině případů se jedná o aplikace open-source[1].

Tabulky obsahují stručný přehled vlastností systémů. Dále následuje rozsáhlejší popis jednotlivých systémů.

Axiom

Používá se ve výzkumu a vývoji matematických algoritmů. Při práci s Axiomem je možné využít matematických objektů jako jsou okruhy, pole nebo polynomy. Stejně tak můžeme používat datové struktury známé z počítačů (seznamy, stromy, hash tabulky), které mají automaticky přiřazené hodnotové typy.

Axiom používá programovací jazyk známý jako A#.

Původně byl Axiom vyvíjen v IBM pod názvem Scratchpad, pod názvem Axiom se vyvíjí od roku 1973. Původně byl komerčním produktem, ale v současné době je licencován jako open-source software.

CoCoA

(COmputations in COmmutative Algebra) je volně dostupný počítačový algebraický systém používaný pro počítání s čísly a polynomy. Knihovna CoCoA je dostupná pod licencí GNU General Public License. CoCoA je k dispozici pro mnoho operačních systémů a to například i pro Macintosh s procesorem PPC a x86, Linux s procesory x86, x86-64 & PPC, Sun Solaris s procesorem SPARC a Windows. Knihovna CoCoA je využívána hlavně vědeckými pracovníky, ale může být použita i pro jednoduché výpočty.

Používá se pro počítání a řešení problémů s velkými celými čísly, racionálními čísly, polynomy, lineárními systémy, při řešení rovnic s nezápornými kořeny, logickými výrazy, při obarvování geografických map, pro řešení Heronovy formule[3] a při používání Gröbnerových bází.

Derive

[1] Open source nebo také open-source software (OSS) je počítačový software s otevřeným zdrojovým kódem. Otevřenost zde znamená jak technickou dostupnost kódu, tak legální dostupnost - licenci software, která umožňuje, při dodržení jistých podmínek, uživatelům zdrojový kód využívat, například prohlížet a upravovat.

[3] Důležitá v rovinné geometrii. Je dán trojúhelník ABC o stranách délek a, b, c. Obsah p trojúhelníku ABC je potom $p = \sqrt{s(s-a)(s-b)(s-c)}$, kde $s = \frac{1}{2}(a+b+c)$.

Název	Web	Poznámky
Axiom	http://wiki.axiom-developer.org/	
CoCoA	http://cocoa.dima.unige.it/	počítání s polynomy
Derive	http://education.ti.com/	vývoj ukončen
DCAS	http://sourceforge.net/projects/dcas/	
DoCon	http://www.haskell.org/docon/	
Eigenmath	http://www.eigenmath.net/	
Fermat	http://www.bway.net/~lewis	polynomy, matice
GAP	http://www.gap-system.org/	grupy, diskrétní matematika
GiNaC	http://www.ginac.de/	C++ knihovna
Macaulay	http://www.math.columbia.edu/~bayer/Macaulay/	polynomiální
Magma	http://magma.maths.usyd.edu.au/magma/	algebra, kryptografie, teorie množin a čísel
Maple	http://www.maplesoft.com/	
MathCad	http://www.mathsoft.com/	WYSIWYG rozhraní
Mathematica	http://www.wolfram.com/	
Mathomatic	http://www.mathomatic.org/	obecná algebra
Maxima	http://sourceforge.net/projects/maxima	
Meditor	http://jscl-meditor.sourceforge.net/	symbolická matematická knihovna v Javě a matematický editor
MuMATH	http://en.wikipedia.org/wiki/MuMATH	
MuPAD	http://www.mupad.de/	
PARI/GP	http://pari.math.u-bordeaux.fr/	teorie čísel
REDUCE	http://reduce-algebra.com/	
SAGE	http://sagemath.com/	
Singular	http://www.singular.uni-kl.de/	polynomiální
Yacas	http://www.singular.uni-kl.de/	

Tabulka 3.1: Počítačové algebraické systémy a jejich webové stránky.

Derive byl vyvíjen jako nástupce pro muMATH firmou Soft Warehouse v Honolulu (Hawaii), v současnosti převzala Derive firma Texas Instruments. Derive je naprogramován v jazyce muLISP. První verze byla vydána v roce 1988.

Vzhledem k tomu, že Derive požaduje poměrně málo operační paměti je použitelný i na starších a méně výkoných počítačích. Derive je dostupný pro operační systém Windows a pro DOS. Ve velké míře se používá při výuce.

Současná verze je Derive 6.1. Symbolická agebra, která se používá v přenosných kalkulátorech TI (Texas Instruments) vychází právě z Derive.

DCAS

Dcas je dynamický počítačový algebraický systém založený na myšlence používat identity[4] jako pravidla pro algebraické manipulace. V 70. letech 20. století vytvořil Robert Fenichel

[4]Identita, nebo také identické zobrazení, je matematické zobrazení, které přiřazuje prvku množiny ten samý prvek stejné množiny. Aplikací identity se tedy nic nezmění, výsledkem je opět vstupní hodnota. Značí se Id nebo I.

Název	Požadavky na operační systém	Licence(cena)
Axiom	Windows, Mac OS ?[2], Linux, BSD ?, Unix ?	upravená BSD licence, free
CoCoA	Windows, Mac OS, Linux, BSD, Unix	GPL,free
Derive	Windows	$99 Educational, $199 Professional
DCAS	Windows, Mac OS ?, Linux ?, BSD ?, Unix ?	non-OSI approved license
DoCon	Windows ?, Mac OS ?, Linux, BSD ?, Unix ?	GPL
Eigenmath	Windows, Mac OS, Linux, BSD, Unix	
Fermat	Windows, Mac OS, Linux, BSD, Unix	shareware, $50
GAP	Windows, Mac OS, Linux, BSD, Unix	GPL, free
GiNaC	Windows, Mac OS ?, Linux, BSD, Unix	GPL, free
Macaulay	Windows, Mac OS, Linux, BSD ?, Unix ?	GPL, free
Magma	Windows, Mac OS, Linux, BSD, Unix	komerční
Maple	Windows, Mac OS, Linux, BSD ?, Unix	komerční
MathCad	Windows	komerční
Mathematica	Windows, Mac OS, Linux, Unix	komerční
Mathomatic	Windows, Mac OS, Linux, BSD, Unix	komerční
Maxima	Windows, Mac OS, Linux, BSD, Unix	GPL, free
Meditor	Windows, Mac OS, Linux, BSD, Unix	GPL, free
MuMATH		komerční
MuPAD	Windows, Mac OS, Linux	komerční
PARI/GP	Windows, Mac OS, Linux, BSD, Unix	GPL, free
REDUCE	Windows, Mac OS, Linux, BSD, Unix	komerční
SAGE	Windows, Mac OS, Linux, BSD, Unix	komerční
Singular	Windows, Mac OS, Linux, BSD, Unix	komerční
WIRIS	Windows ?, Mac OS, Linux, BSD, Unix	komerční
Yacas	Windows, Mac OS, Linux, BSD, Unix	GPL, free

Tabulka 3.2: Počítačové algebraické systémy jejich požadavky na systém a licence.

systém pojmenovaný FAMOUS, který používal pro tyto operace LISP. FAMOUS se stal později základem pro DCAS.

Moderní systém DCAS byl vytvořen Martinem Johansenem a byl znám pod označením DCAS Ether. Systém pracoval pomocí voleb tříd identit zakládajích se na formách vstupních výrazů.

DoCon

DoCon je počítačový algebraický systém napsaný v programovacím jazyce Haskell. Jedná se o knihovnu, jejíž funkce jsou jednoduše přístupné pro uživatele, kteří je chtějí použít ve svých aplikacích.

Eigenmath

Eigenmath je volně dostupný, snadno použitelný, multiplatformní počítačový algebraický systém od George Weigta napsaný v jazyce C.

Identitou se také v jiném významu myslí rovnice, která je splněna ve všech případech, tzn. její levá a pravá strana jsou identické, mají pouze jiný tvar.

Název	Podpora pro počítání GCD
Axiom	ANO, dostatečné možnosti
CoCoA	ANO, dostatečné možnosti
Derive	ANO, základní možnosti
DCAS	není známo
DoCon	není známo
Eigenmath	není známo
Fermat	není známo
GAP	není známo
GiNaC	není známo
Macaulay	není známo
Magma	není známo
Maple	ANO, rozsáhlé možnosti
MathCad	není známo
Mathematica	ANO, rozsáhlé možnosti
Mathomatic	není známo
Maxima	ANO, dostatečné možnosti
Meditor	není známo
MuMATH	není známo
MuPAD	není známo
PARI/GP	není známo
REDUCE	ANO, dostatečné možnosti
SAGE	není známo
Singular	ANO, rozsáhlé možnosti
Yacas	není známo

Tabulka 3.3: Počítačové algebraické systémy a počítání GCD.

Je dostupný pro Windows a Mac OS X, zdrojový kód je dostupný taktéž.

Fermat

Fermat je počítačový algebraický systém vyvíjený Prof. Robertem H. Lewisem z Fordhamské univerzity. Je použitelný při počítání s celými čísly (libovolné velikosti), racionálními čísly, reálnými čísly, komplexními čísly, poly konečných elementů, polynomy o více proměnných, racionálními funkcemi nebo polynomy modulo[5] jiných polynomů. Hlavní oblast aplikací je aritmetika racionálních funkcí s více proměnnými a maticová algebra přes okruhy polynomů s více proměnnými nebo racionální funkce. Fermat neumí zjednodušování transcendentních funkcí[6] nebo symbolické integrace.

GAP

GAP (Groups, Algorithms and Programming) je počítačový algebraický systém pro výpočty zaměřené na oblast diskrétní algebry, ale můžeme využít též počítání v oblasti teorie množin. GAP

[5]Zbytek po dělení nebo také modulo je početní operace související s operací celočíselného dělení. Například 7 mod 3 = 1.

[6]Funkci označujeme jako algebraickou, pokud ji lze vyjádřit ve tvaru polynomu, např. pokud lze funkci $y = \mathrm{f}(x)$ vyjádřit jako $\mathrm{P}(x, y) = 0$, kde P je polynom, pak se jedná o algebraickou funkci. Stupeň polynomu P pak určuje stupeň funkce. Funkce, které nejsou algebraické, označujeme jako transcendentní.

byl v 1986–1997 vyvíjen na Lehrstuhl D für Mathematik (LDFM), RWTH Aachen v Německu. Později na School of Mathematical and Computational Sciences na University of St. Andrews ve Skotsku.

GAP je dostupný pro systémy Unix, Windows a Macintosh.

Pro GAP existuje mnoho rozšiřujích funkcí ve formě balíčků. Uživatel si tedy může upravit funkcionalitu programu podle sebe. V roce 2006 bylo distribuováno 58 balíčků.

Je též dostupné rozhraní pro použití balíčků v systému SINGULAR bez použití GAP.

Giac/Xcas

Více je uvedeno v kapitole 4.5.

GiNaC

GiNaC je free počítačový algebraický systém uvolňovaný pod GNU General Public License. Název pochází z akronymu GiNaC is Not a CAS. Toto jméno slouží jako narážka, že se jedná o GNU projekt.

Nemá uživatelsky přívětivé rozhraní. Symbolické algoritmy jsou zapisovány uživatelem přímo ve vlastní upravené implementaci jazyka C++.

Macaulay

Macaulay je počítačový algebraický systém pro počítání s polynomy, obzvlášť pro výpočty Gröbnerových bází. Macaulay je navržen pro řešení problémů komutativní algebry a algebraické geometrie[7] a má poměrně jednoduchou syntax často popisovanou jako "jazyk algebraických systémů". Je pojmenován podle F.S. Macaulay, který pracoval na eliminační teorii.

Macaulay byl vyvinut Daveem Bayerem a Mikem Stillmanem. Je volně dostupný pro Macintosh, Linux a Windows.

Většina uživatelů používá Macaulay2, což je nová verze od Dana Graysona a Mikea Stillmana.

Magma

Magma je vysoce výkoný počítačový algebraický systém navržený pro řešení problémů algebry, teorii čísel, geometrie a kombinatoriky. Je pojmenován podle algebraické struktury magma[8]. Jedná se o nekomerční proprietární software[9], distribuovaný pod cost recovery licencí. Lze ho používat hlavně na systémech na bázi Unix a Linux, ale podporuje i Windows.

Maple

Více je uvedeno v kapitole 4.2.

---

[7] Algebraická geometrie je matematická disciplína nacházející se, jak už název napovídá, na rozhraní algebry a geometrie. Používá metody komutativní algebry pro řešení geometricky formulovaných problémů. Je jednou z nejrozvíjenějších disciplín moderní matematiky a má řadu styčných bodů s ostatními disciplínami matematiky – zejména komplexní analýzou, topologií a teorií čísel.

Impulsem k rozvoji algebraické geometrie byly pokusy řešit polynomiální rovnice více proměnných. Algebraická geometrie se zabývá obecnými vlastnostmi řešení těchto rovnic a jejich soustav.

[8] Magma je jiný výraz pro grupoid. V algebře je grupoid základní algebraická struktura s jednou operací. Je to množina, na které je definována jedna binární operace tak, aby byla na této množině neomezeně proveditelná - tj. výsledkem operace provedené na libovolných prvcích množiny byl prvek z této množiny.

[9] Proprietární software je takový software, kde jeho autor upravuje licencí (typicky EULA) či jiným způsobem možnosti jeho používání. K takovému software nejsou zpravidla k dispozici volně zdrojové kódy či v nich nelze svobodně dělat úpravy a výsledné dílo distribuovat. Takový software obvykle spadá do kategorie komerčního software, který jeho autor prodává.

MathCad

Mathcad (původně MathCAD) je software pro provádění a dokumentaci inženýrských a vědeckých výpočtů. Poprvé se objevil v roce 1986 ve verzi pro operační systém DOS. Byl to první systém jenž zavedl přímou matematickou notaci kombinovanou s automatickým výpočtem. Byl také prvním systémem který automaticky vyhodnocoval výrazy a zároveň kontroloval správnost jednotek v rámci systému SI (International system of units). V současnosti se Mathcad vyrovná některým počítačovým algebraickým systémům jako například Mathematica nebo Maple. Mathcad se však orientuje spíše na snadné používání a na numerické aplikace. Mathcad byl vypracován a původně napsán Allenem Razdowem (z MIT), spoluzakladatelem Mathsoftu, který je v současnosti součástí Parametric Technology Corporation.

Mathcad ukázal cestu mnoha rozdílným matematickým nástrojům. Mathcad je orientován na worksheety, v nichž jsou rovnice a výrazy zobrazovány graficky a ne jen textově.

Mathematica

Mathematica je počítačový algebraický systém původně vytvořený Stephenem Wolframem, zdokonalený skupinou matematiků a programátorů, které Wolfram vedl. Dnes je prodáván firmou Wolfram Research.

Mathematica je velmi často používána ve vědě, studenty na vysokých školách, profesionály, výzkumníky a učiteli. Mathematica je také vysokoúrovňový programovací jazyk.

Mathomatic

Mathomatic je volně dostupný, přenosný, obecně použitelný počítačový algebraický systém, který může symbolicky řešit, zjednodušovat, slučovat a porovnávat algebraické rovnice, pracovat s komplexními čísly a polynomy. Provádí symbolické výpočty (limity, derivace, extrémy, Taylorovy řady a polynomiální integrace a Laplaceovu transformaci) a zvládá veškerou algebru, mimo trigonometrie a logaritmů. Mathomatic je výhradně prací jednoho vývojáře a je napsán v optimalizovaném jazyce C. Nedovoluje výpočet velmi rozsáhlých výrazů, které by mohly zabrat hodně času a operační paměti.

Mathomatic může být použit pro generování kódu, konverzi rovnic do optimalizovaného tvaru pro programovací jazyky Python, C a Java.

Mathomatic je konzolový program s barevnou příkazovou řádkou běžící pod libovolným operačním systémem. Konzole je velmi jednoduchá a prakticky nevyžaduje nic nového se učit.

Není potřeba umět programovat, Mathomatic pracuje jako algebraický kalkulátor. Výrazy a rovnice jsou vkládány ve standardní infixové notaci[10]. Operace nad nimi jsou prováděny vkládáním jednoduchých příkazů. Všechny výpočty mají dvojitou přesnost v plovoucí řádové čárce s přesností 14 desetinných míst. Numerické operace obsahují výpočty, sčítání a numerické integrace využívající Simpsonovo pravidlo[11].

Maxima

Více je uvedeno v kapitole 4.3.

[10] Infixová notace je nejběžnější způsob matematického zápisu, ve kterém jsou operátory napsány mezi operandy, se kterými pracují (např. 3 + 4). Její počítačová analýza není tak jednoduchá jako v případě prefixové notace (+ 3 4) nebo postfixové notace (3 4 +), ale je používána ve spoustě programovacích jazycích pro její rozšířenost.

V infixové notaci jsou narozdíl od prefixové nebo postfixové notaci závorky nutností. V případě, že závorky chybějí, záleží na prioritě početních operací.

[11] Simpsonovo pravidlo – výpočet plochy pod křivkou f(x) je nahrazen výpočtem plochy pod parabolou proložené třemi body funkce f(x).

MuMATH

MuMATH je počítačový algebraický systém, který byl vyvinut na konci 70. let 20. století Albertem D. Richem a Davidem Stoutemyerem ze Soft Warehouse v Honolulu, Hawaii. Byl napsán v programovacím jazyku muSIMP, jenž byl vystavěn na LISPu a jeho obdobě muLISP. Platformy, které podporoval byly CP/M a TRS-DOS (od verze muMATH-79), Apple II (od verze muMATH-80) a MS-DOS (od verze muMATH-83, je i zároveň verzí poslední).

Firma Soft Warehouse později začala vyvíjet Derive a v roce 1999 byla prodána firmě Texas Instruments.

MuPAD

MuPAD byl původně vyvíjen výzkumnou skupinou MuPAD na University of Paderborn, od roku 1997 na něm pracuje firma SciFace Software GmbH & Co. KG ve spolupráci se skupinou MuPAD a partnery z dalších univerzit .

Až do podzimu 2005 byl MuPAD nabízen pro výzkum a vzdělání zdarma, ale důsledkem ukončení činnosti hlavní vývojové skupiny je v současnosi dostupný jen za poplatek.

Jádro MuPADu je vázáno na Scientific Notebook. Dřívější verze MuPADu byly vázány na SciLab.

PARI/GP

PARI/GP je počítačový algebraický systém s hlavním cílem usnadnit počítání v teorii čísel. Od začátku října 2006 je dostupná stabilní verze 2.3.1 a je vyvíjena verze 2.4.0. PARI/GP je od verze 2.1.0 free software a je distribuován pod GNU General Public License. Je možné ho spustit na většině operačních systémů.

REDUCE

REDUCE je obecně použitelný počítačový algebraický systém používaný zejména k aplikacím ve fyzice.

Vývoj REDUCE započal v 60. letech 20.století Anthony C. Hearnem.

REDUCE je napsán výhradně ve vlastním dialektu jazyka LISP nazývaném Standard LISP, vyjádřeném v syntaxi blízké Algolu nazývané RLISP, jenž je využíván jako základ programovacího jazyka při zadávání uživatelských příkazů.

Implementace REDUCE je dostupná pro velké množství počítačových systémů, operačních systémů a na LISPu založených systémech. Současně je dostupná pro většinu distribucí systémů Unixu, Linuxu, Microsoft Windows nebo Apple Macintosh.

REDUCE je distribuována jako cost-recovery[12] s níž byl dodáván obvykle i plný zdrojový kód, což vedlo k tomu, že REDUCE byla velmi populárním nástrojem při výzkumech na poli počítačových algebraických systémů.

SAGE

Software for Algebra and Geometry Experimentation (zkráceně SAGE) je distribuce Pythonu a Pyrexu pro použití jako počítačový algebraický systém a systém pro tvorbu grafů. Je podporován Linuxem na procesorech AMD, x86, a Itanium; Mac OS X na PowerPC a Intel; a Cygwinem v prostředí Microsoft Windows.

[12] zákazník jednou zaplatí a aktualizace má zdarma

SAGE poskytuje rozhraní pro začlenění ostatních matematických software a knihoven. Rozhraní je použitelné například pro gnuplot, Magma, Maple a Mathematica, které dovolují uživateli spojit tyto systémy a používat je naráz. Je tak vstupem pro ostatní matematické nástroje jako je GNU TEXmacs.

SAGE poskytuje web servery pro grafické rozhraní MoinMoin, distribuované výpočty a Trac.

Většina vývoje SAGE je podporována granty. Hlavní vývojář William Stein je matematik na University of Washington.

SAGE je volně dostupný software pod GNU GPL.

Singular

Více je uvedeno v kapitole 4.4.

Yacas

Yacas je open source, obecně použitelný a snadno použitelný počítačový algebraický systém. Jméno je zkratka z Yet Another Computer Algebra System.

Yacas je postaven na vlastním programovacím jazyce navrženým pro tyto účely v němž může být jednoduše implementovaný jakýkoliv nový algoritmus. Jazyk je velmi blízký LISPu a programovacímu jazyku systému Mathematica.

Yacas zvládá vstupy a výstupy v plain ASCII nebo v OpenMath, interaktivně nebo v dávkovém módu.

Kapitola 4

Testování algoritmů GCD ve vybraných PAS

Vzhledem k tomu, že počítačových algebraických systémů je velké množství, rychle vznikají nové jednoúčelové nástroje a opět zanikají pro svou zastaralost, bylo třeba zvolit pouze takové počítačové algebraické systémy u nichž je určitá jistota, že obsahují algoritmus pro výpočet největšího společného dělitele, stále se vyvíjejí a jsou dostupné pro různé platformy operačních systémů a hardware. Během výběru se ukázalo, že různé PAS přistupují k výpočtu největšího společného dělitele rozdílně. Někde si lze zvolit z několika algoritmů pro různou rychlost výpočtu, ale například u komerčně dostupného Maple 9.5 toto možné není. V tomto ohledu spíše zvítězily PAS z řad open-source jako například Maxima, které dovolují nastavit poměrně velké množství parametrů. Avšak je třeba si uvědomit, že Maple je velice komplexní balík, jenž se nespecializuje pouze na jednu oblast výpočtů, proto se mohou jevit jeho možnosti v některých ohledech horší než u specializovaných nástrojů.

Jako testované systémy byly nakonec s ohledem na použitelnost vybrány systémy: Maple, Maxima, SINGULAR, Giac/XCAS a CoCoA. Co autora vedlo k výběru těchto systémů je uvedeno v kapitolách přislušných jednotlivým systémům.

4.1 Metodika testování

4.1.1 Hardwarové a softwarové vybavení

Všechny testy probíhaly na počítači typu notebook s dvoujádrovým procesorem a 512MB operační paměti

- CPU typ Mobile DualCore Intel Pentium M, 1600 MHz (12 x 133)
- Základní deska FUJITSU SIEMENS AMILO Pro Edition V3505
- Čipová sada základní desky Intel Calistoga i945GM/PM
- Pracovní paměť 512 MB (DDR2-533 DDR2 SDRAM)

Operační systém

- Microsoft Windows XP Home Edition Service Pack 2

- Linux openSUSE 10.2[1]

Zabrané prostředky jinými aplikacemi po spuštění systémů (hrubý odhad měnící se velmi často)

- Microsoft Windows - 30% operační paměti, využití CPU 5%
- Linux - při správném nastavení jsou zabrané prostředky zanedbatelné

4.1.2 Testovací polynomy

Pokud chceme testovat algoritmus největšího společného dělitele polynomu o více proměnných potřebujeme získat sadu polynomů, jež by k těmto testům posloužila. Jako testovací polynomy byly zvoleny polynomy obsahující všechny mocniny daných řádů (dense) s počtem proměnných 1 až 5, řádu 1 až 5000 podle délky trvání výpočtu a možností daného počítačového algebraického systému.

Většina počítačových algebraických systémů obsahuje nástroje a knihovny pro generování náhodných polynomů. Pokud to daný počítačový algebraický systém umožňoval byly testovací polynomy generovány přímo v jednotlivých testovaných počítačových algebraických systémech, v opačném případě byly využity polynomy vygenerované pomocí Maple 9.52.

S ohledem na časovou náročnost a využitelnost výpočtů v dalším praktickém zpracování bylo třeba vybrat odpovídající polynomy, které byly generovány náhodnou funkcí. Jako hranice, kdy ještě mělo cenu čekat na výpočet byla stanovena doba 10 minut, tomuto musel být přizpůsoben stupeň polynomu a počet proměnných polynomu. Toto omezení bylo zavedeno z důvodů časové náročnosti takto prováděných testů.

Tyto parametry samozřejmě velmi ovlivňují dobu výpočtu a ani se nepředpokládá, že bude nutné počítat největší společné dělitele vysokých řádů a nebo polynomů s hodně proměnnými. Vzhledem k tomu byly použity polynomy maximálně řádu 5000 pro polynomy jedné proměnné až řádu 10 pro polynomy pěti proměnných, jejichž největší společný dělitel byl spočítán v době řádově desítek sekund až jednotek minut.

Při testování nebylo možné použít zcela náhodné polynomy, neboť systémy počítačové algebry velmi rychle odhalí skutečnost, že většina polynomů náhodně vygenerovaných má největší společný dělitel roven 1. Proto byly vygenerovány dva náhodné polynomy a výsledný největší společný dělitel. Polynomy k testování potom byly součinem jednoho náhodného polynomu a náhodně vygenerovaného největšího společného dělitele. Vznikly tak dva polynomy, jejichž stupeň byl roven součtu stupňů násobených polynomů (stupeň náhodných polynomů a náhodného největšího společného dělitele byly stejné).

4.1.3 Průběh testů

Výběr polynomů k testování uvádí kapitola 4.1.2. Pokud byla doba výpočtu větší než zvolených 10 minut byl test přerušen. Každý jednotlivý test s polynomem daného řádu a počtem proměnných byl prováděn desetkrát u každého systému se všemy polynomy. Výsledná hodnota byla zvolena jako průměr ze všech deseti výsledných hodnot.

4.2 Maple 9.52

Systém Maple je v současné době asi nejpoužívanějším komerčním počítačovým algebraickým systémem, proto nemohl chybět ani při testování algoritmů největšího společného dělitele v této

[1] Pouze pro ověření některých výsledků a porovnání, ukázalo se, že doby výpočtu jsou srovnatelné na obou platformách.

práci. Jeho uživatelské rozhraní je graficky přívětivé a naučit se s ním pracovat není nijak složité. Obsahuje funkce a nástroje pro širokou skupinu operací, avšak některé specializované funkce jsou na úkor jeho rozsáhlosti hůře nastavitelné uživatelem a ten se musí spolehnout na výchozí nastavení Maple nebo automatické funkce rozpoznání. To byl také případ výpočtu největšího společného dělitele, kdy uživatel nemůže nastavit jakým algoritmem bude počítán.

Přestože v současné době je k dispozici již Maple verze 11, byla použita verze 9.52 a to proto, že Maple je produkt komerční a licence jenž byla k dispozici byla právě pro verzi starší. Nicméně je velice pravděpodobné, že algoritmy pro výpočet největšího společného dělitele budou v obou verzích stejně účinné.

Základní vlastnosti systému Maple 9.52 [14]:

- vícedokumentové grafické uživatelské rozhraní s hypertextovým propojením jednotlivých pracovních listů,
- hierarchická struktura, formátovaná textová pole, různé styly znaků, kontextová nápověda umoňující přípravu profesionálních dokumentů pro přímé prezentace nebo tisk,
- interaktivní práce v matematickém prostředí (možnost sdílení proměnných mezi pracovními listy),
- vylepšené menu a obohacená lišta nástrojů,
- OLE2 klient - zápisník může obsahovat dynamicky připojené či vložené objekty z ostatních aplikací (z těch, které mohou vystupovat jako OLE servery), pouze klasický worksheet,
- matematický asistent - velké množství matematických rutin, které však nelze používat bez matematických znalostí, celkem více než 3500 příkazů,
- matematický server - využití matematických schopností externími aplikacemi,
- generování zdrojových programů ve FORTRANu nebo v C,
- generování zdrojového textu do LaTeXu, html, rtf, xml a txt formátu,
- generování obrázků v různých grafických formátech,
- použití MapleNet umožňuje přístup k výpočetním možnostem pomocí webového prohlížeče,
- Maplets - definování vlastního grafického uživatelského rozhraní pro příkazy a procedury Maple,
- obohacení o Maplets jako součást speciální knihovny pro názornou výuku matematiky,
- program obsahuje velmi kvalitně zpracovaný a podrobný HELP v nové struktuře, novinka - Math Dictionary,
- přeprogramování jádra v jazyku Java (částečné zpomalení a výrazně větší nároky na paměť - trochu se zlepšilo v posledních verzích), nový design interface, zachována možnost klasického worksheetu (úspornější).

Hardwarové a softwarové nároky

- Intel Pentium III 650 MHz
- většina operačních systémů
- minimálně 64 - 256 MB RAM
- 350 - 485 MB disk
- CD ROM
- TCP/IP internet protokol

4.2.1 Generování testovacích polynomů

Pro generování náhodných polynomů slouží v systému Maple 9.52 funkce:

randpoly(vars, opts)

vars - proměnné nebo seznam proměnných
opts - (volitelně), zádání tvaru polynomu pomocí předpisu (rovnice) nebo bližší specifikace pro způsob generování polynomu

Při zavolání funkce *randpoly(vars, opts)* dojde k vygenerování náhodného polynomu s počtem proměnných, jenž byly zadány jako vstupní parametr *vars*. Můžeme zvolit i tvar polynomu, který má být generován a to pomocí nepovinných parametrů.

Prvním parametrem je již zmiňovaný *vars*, který ovlivňuje kolik neznámých bude polynom obsahovat. Další možné parametry jsou:

Parametr	Použití	Výchozí hodnota
coeffs	generování koeficientů	rand(-99..99)
expons	generování exponentů jako koeficientů	rand(6)
terms	počet generovaných členů	6
degree	stupeň polynomu	5
dense	polynom bude obsahovat všechny mocniny	sparse
homogeneous	polynom je homogenní	

Příklady použití:

$>randpoly(x);$
$==>50 - 85x^5 - 55x^4 - 37x^{32} + 97x$
$>randpoly([x, y]);$
$==>45yx - 8y^2 - 93x^4 + 92x^5 + 43x^4y - 62x^3y^2$
$>randpoly([x, y], terms = 10);$
$==>-62 + x - 47y - 91y^2 - 47x^3 - 61y^3 + 41x^2y^2 - 58xy^3 - 90y^4 + 53x^2y^3$
$>randpoly([x, y], dense, degree = 4);$
$==>72 + x^4 - 94x^3y + 83x^3 - 86x^2y^2 + 23x^2y - 84x^2 + 19xy^3 - 50xy^2 + 88xy - 53x + 85y^4 + 49y^3 + 78y^2 + 17y$
$>randpoly([x, y], homogeneous);$
$==>-78x^2y^3 - 48x^4y - 47y^5$
$>randpoly([x, y, z], dense, homogeneous, degree = 2);$

$==>68x^2 - 72zx - 87xy + 79z^2 + 43yz - 66y^2$
$>randpoly([x, sin(x), cos(x)]);$
$==>-42sin(x)^2+88cos(x)^2-76cos(x)^3-65x^3sin(x)^2+25x^2sin(x)^2cos(x)+28cos(x)^4$
$>randpoly(z, expons = rand(-5..5));$
$==>-\frac{61}{z^5} + \frac{9}{z^4} - 83z + 78z^4 + 94z^2$
$>randpoly([x], coeffs = proc()randpoly(y)endproc);$
$==>(-43-36y^5+40y^4+22y^3+5y^2-88y)x^5+(-55-73y^5+25y^4+4y^3-59y^2+62y)x^4+$
$==>(-78+25y^5+9y^4+40y^3+61y^2+40y)x^3+(81+62y^5+11y^4+88y^3+y^2+30y)x^2+$
$==>(57 - 5y^5 - 28y^4 + 4y^3 - 11y^2 + 10y)x + 58 - 82y^5 - 48y^4 - 11y^3 + 38y^2 - 7y$
$>RandomTools[Generate](polynom(integer(range = -10..10), x, degree = 4));$
$==>8 - x - x^2 + x^3 - x^4$

Pro generování náhodných polynomů zvoleného stupně a počtu proměnných byla využita funkce `randpoly()`.

```
> g := randpoly([x, y], degree=10,dense):
> f1:=expand(g * randpoly([x,y], degree=100,dense)):
> f2:=expand(g * randpoly([x, y], degree=100,dense)):
> start:=time():
> gcdm:=gcd(f1,f2):
> dobaBehu:=time()-start;
```

4.2.2 Možnosti výpočtu největšího společného dělitele

Maple obsahuje funkci pro výpočet největšího společného dělitele polynomů `gcd` a funkci pojmenovanou `igcd` pro počítání největšího společného dělitele celých čísel. Maple taktéž nabízí rozšířenou funkci (`gcdex`, `igcdex`), která počítá v součtu vstupy `gcd a` a `b`, `x` a `y` způsobem `ax + by = gcd(a,b)`.

Pro testy byla použita funkce `gcd`. Tato funkce volí typ použitého algoritmu podle vstupních hodnot.

`gcd(a,b,'cofa','cofb')`

`a,b` - polynomy s více proměnnými zadáváné jako pole

`cofa,cofb` - (volitelně) nevyhodnotitelné výrazy

Algoritmus `gcdex` - extended Euclidean algorithm for polynomials

`gcdex(A,B,x,'s','t')`

`gcdex(A,B,C,x,'s','t')`

`A,B,C` - polynomy s proměnnou x

`x` - jméno proměnné

`s,t` - (volitelně) nevyhodnotitelné výrazy

Příklad:

```
> gcdex(x^3-1,x^2-1,x,'s','t');
```
==>$-1+x$
```
> s,t;
```
==>$1, -x$
```
> gcdex(x^2+a,x^2-1,x^2-a,x,'s','t');
> s,t;
```
==>$\frac{-(a-1)}{a+1}, 2\frac{a}{a+1}$
```
> gcdex(1,x,1-2*x+4*x^2,x,'s','t');
> s,t;
```
==>$1, 4x-2$

4.2.3 Výsledky testů největšího společného dělitele

Tabulka 4.1 shrnuje provedené testy výpočtu největšího společného dělitele. Pokud je v některém poli černá tečka znamená to, že test probíhal příliš dlouho viz kapitola 4.1.2.

Porovnání a zhodnocení s ostatními testovanými systémy je uvedeno v poslední kapitole.

4.3 Maxima

Maxima je často označována jako open-source alternativa systému Maple. Ve skutečnosti sice stačí na většinu běžných výpočtů, ale Maple jako produkt komerční a využívaný v mnohem větší míře obsahuje mnoho drobných vylepšení, která uživateli pomohou a zjednodušší práci. I přes některé nedostatky je jeho ovládání poměrně jednoduché a pro uživatele přístupné. Pro testování největšího společného dělitele byl zvolen jako možná náhrada za komerční Maple, když uživatel nebude mít Maple k dispozici.

Maxima je systém nejen pro symbolické ale i numerické výpočty výrazů zahrnující:

- derivace, integrace, řešení diferenciálních rovnic,
- Tailorovy řady,
- Laplaceovu transformaci,
- soustavy lineárních rovnic, polynomy,
- práci s množinami, seznamy, vektory, maticemi a tenzory.

Další vlastnosti:

- vysoká přesnost numerických výsledků zajištěná používáním přesných zlomků, libovolně přesných celých čísel a libovolně přesných čísel v plovoucí desetinné řádové čárce[2],
- vykreslování funkcí a dat ve 2–D nebo 3–D,
- různá grafická uživatelská rozhraní, například wxMaxima je jedno z nich založené na platformě wxWidgets GUI,
- matematický editor GNU TEXmacs může být použit tak, aby poskytoval interaktivní GUI pro Maximu

[2] závisí samozřejmě na výkonu hardware

proměnné	stupeň	t[s]	průběžné hodnoty t[s]									
	1	0,000	0	0	0	0	0	0	0	0	0	0
	2	0,002	0	0	0	0	0	0	0,015	0	0	0
	5	0,000	0	0	0	0	0	0	0	0	0	0
	10	0,000	0	0	0	0	0	0	0	0	0	0
	50	0,006	0,016	0	0,016	0	0	0,015	0	0,016	0	0
	100	0,002	0	0	0	0	0	0	0	0,016	0	0
	200	0,016	0,032	0,046	0,016	0,016	0,015	0,016	0	0	0,015	0
	300	0,020	0,047	0,016	0,016	0,015	0,016	0,016	0,015	0,015	0,016	0,032
	400	0,025	0,078	0,032	0,015	0,016	0,016	0,016	0,015	0,031	0,015	0,016
	500	0,027	0,078	0,031	0,016	0,015	0,032	0,016	0,016	0,015	0,031	0,015
	1000	0,077	0,157	0,063	0,063	0,062	0,094	0,062	0,063	0,063	0,063	0,078
1	2000	0,358	0,765	0,282	0,328	0,328	0,282	0,313	0,328	0,344	0,282	0,328
	1	0,019	0,188	0	0	0	0	0	0	0	0	0
	2	0,002	0	0	0,016	0	0	0	0	0	0	0
	5	0,014	0,093	0	0	0	0,016	0	0		0,016	0
	10	0,016	0,062	0	0	0,016	0,015	0,016	0,015	0,016	0,016	0
	20	0,042	0,25	0,032	0,031	0,016	0,016	0,016	0,016	0,015	0,015	0,016
	50	0,175	0,297	0,187	0,172	0,14	0,172	0,172	0,141	0,157	0,156	0,156
	100	0,757	0,969	0,875	0,734	0,719	0,75	0,75	0,75	0,75	0,75	0,734
	160	2,880	3,203	3,031	2,844	2,906	2,86	2,875	2,843	2,844	2,86	2,86
	200	7,220	6,968	7,219	7,297	7,233	7,219	7,156	7,422	7,235	7,235	7,219
	300	21,039	17,702	21,454	21,358	21,359	21,5	21,376	21,392	21,376	21,36	21,514
	400	45,322	44,484	46,484	47,422	46,922	44,891	44,515	44,547	44,563	44,844	44,547
2	500	105,783	100,797	99,766	99,218	144,093	101,172	105,359	101,936	99,53	100,172	
	1	0,017	0,156	0	0	0	0	0	0	0,016	0	0
	2	0,003	0,016	0	0	0,015	0	0	0	0	0	0
	5	0,020	0,125	0	0,015	0,016	0	0,016	0,016	0	0	0,015
	10	0,278	1,047	0,187	0,188	0,218	0,171	0,204	0,171	0,172	0,203	0,219
	16	2,458	2,469	3,032	2,64	2,609	2,578	2,452	2,453	1,86	1,812	2,678
	20	3,673	3,78	4,375	3,86	3,188	3,671	3,828	3,547	3,203	3,266	4,016
	30	44,612	52,579	49,456	40,047	41,156	48,484	46,642	38,234	39,86	39,048	50,61
3	50	594,791	591,757	597,825	•	•	•	•	•	•	•	•
	1	0,023	0,219	0	0	0	0	0	0	0	0,015	0
	2	0,003	0	0	0,016	0	0	0,016	0	0	0	0
	5	0,244	1,202	0,172	0,125	0,109	0,141	0,14	0,125	0,14	0,141	0,14
	10	1,272	1,156	0,953	1,437	0,969	1,329	1,36	0,844	1,562	1,64	1,469
4	20	62,518	56,078	74,609	73,984	52,564	45,922	78,33	53,471	75,717	73,14	41,36
	1	0,005	0	0,015	0	0	0,016	0	0,015	0	0	0
	2	0,009	0,015	0,016	0,016	0,015	0,016	0,015	0	0	0	0
	5	0,369	0,531	0,312	0,312	0,422	0,344	0,375	0,328	0,422	0,281	0,359
5	10	7,564	7,109	7,578	7,515	8,172	9,125	5,469	6,452	10,172	6,625	7,422

Tabulka 4.1: Výsledky testů GCD v systému Maple

- obsahuje kompletní programovací jazyk se syntaxí podobnou ALGOLu, ale se sémantikou podobnou Lispu, proto může být Maxima snadno použitelná pro výuku programování a počítačové algebry,
- vzhledem k tomu, že Maxima je napsána v Common Lisp, dá se snadno rozšiřovat pomocí vlastních úprav kódu a dokonce programy napsané v Lispu mohou být volány přímo z Maximy,
- možnost generovat kód v jiných programovacích jazycích (např. Fortran), které mohou výpočty provést mnohem efektivněji,
- některé speciální výpočty jako je faktorizace velkých čísel, manipulace s velmi velkými

polynomy, atd. je většinou lepší provádět ve specializovaných systémech.

Harwarové a softwarové nároky

- Hardwarové nároky distributor neuvádí, doba výpočtu je potom závislá na použitém hardware.
- Zdrojový kód Maximy lze kompilovat na většině systémů, jako je Windows, Linux a MacOS X. Zdrojový kód pro všechny systémy a předkompilované binární balíky pro Windows a Linux jsou dostupné na web stránkách SourceForge - `http://maxima.sourceforge.net/download.shtml`.

4.3.1 Historie

Maxima je následníkem systému Macsyma, legendárního počítačového algebraického systému vyvinutého na konci 60. let 20. století na Massachusetts Institute of Technology. Maxima má stálou aktivní uživatelskou komunitu, k čemuž přispívá její dostupnost jako open source. Macsyma byl revoluční systém a mnoho dalších systémů, jako Maple a Mathematica se jím nechaly inspirovat.

Když se Maxima začala vyvíjet odděleně od Macsymy, jejím prvním tvůrcem byl William Schelter a to od roku 1982 až do jeho smrti v roce 2001. V roce 1998 dostal povolení k vydání zdrojových kódů pod licencí GNU General Public License (GPL). Když zemřel byla utvořena skupina uživatelů a vývojářů, jenž dokázala otevřít Maximě cestu k široké veřejnosti.

4.3.2 Generování testovacích polynomů

Maxima neumožňuje generovat náhodné polynomy, které by byly vhodné pro účely testování tak, jak je uvedeno v této práci, viz kapitola 4.1.2. Pro generování polynomů byl tedy použit systém Maple, viz kapitola 4.2.1 a výsledné polynomy exportovány do systému Maxima.

4.3.3 Možnosti výpočtu největšího společného dělitele

Maxima nabízí většinu používaných algoritmů pro výpočet největšího společného dělitele. Pomocí funkce

```
gcd (p_1, p_2, x_1, ...)
```

a jejích parametrů lze specifikovat co a jakým algoritmem bude počítáno. `Funkce gcd()` počítá největšího společného dělitele polynomů `p_1` a `p_2`. Uživatel může ovlivnit jaký algoritmus bude použit. Nastavením `gcd` na `ez`, `subres`, `red`, nebo `spmod` volíme ezgcd, počítání pomocí subresultantů, redukčním algoritmem nebo modulárním algoritmem. Jako výchozí a pro testy použitý je nastaven `subres`.

4.3.4 Výsledky testů největšího společného dělitele

Tabulka 4.2 shrnuje provedené testy výpočtu největšího společného dělitele. Pokud je v některém poli černá tečka znamená to, že test probíhal příliš dlouho viz kapitola 4.1.2.

Porovnání a zhodnocení s ostatními testovanými systémy je uvedeno v poslední kapitole.

proměnné	stupeň	t[s]	průběžné hodnoty t[s]									
1	1	0,000	0	0	0	0	0	0	0	0	0	0
1	2	0,000	0	0	0	0	0	0	0	0	0	0
1	5	0,030	0	0,1	0	0	0	0	0	0	0	0,2
1	10	0,030	0	0	0	0,2	0	0	0	0	0,1	0
1	50	0,550	0,6	0,5	0,6	0,5	0,5	0,6	0,6	0,6	0,4	0,6
1	100	0,503	0,65	0,5	0,51	0,5	0,52	0,5	0,46	0,5	0,48	0,41
1	200	5,911	5,93	5,95	5,89	5,89	5,86	5,88	5,9	5,93	5,92	5,96
1	300	23,704	23,93	23,69	23,62	23,65	23,78	23,64	23,73	23,64	23,67	23,69
1	400	44,484	44,43	44,44	44,46	44,47	44,49	44,53	44,57	44,47	44,48	44,5
1	500	114,424	114,47	114,39	114,5	114,38	114,42	114,36	114,48	114,41	114,51	114,32
1	1000	677,760	677,33	677,11	677,27	677,33	678,18	677,36	677,89	678,67	677,75	678,71
2	1	0,360	0,9	0,8	0	0,2	0	0,1	0	0,8	0	0,8
2	2	0,210	0	0,1	0,3	0,3	0	0,3	0	0,2	0,9	0
2	5	0,302	0,17	0,25	0,13	0,17	0,6	0,1	0,5	0,3	0,5	0,3
2	10	0,519	0,69	0,47	0,49	0,54	0,53	0,54	0,49	0,5	0,5	0,44
2	20	28,258	28,31	28,2	28,18	28,26	28,28	28,39	28,14	28,37	28,25	28,2
2	50	více jak 10 min	•	•	•	•	•	•	•	•	•	•
2	100	více jak 10 min	•	•	•	•	•	•	•	•	•	•
2	160	více jak 10 min	•	•	•	•	•	•	•	•	•	•
2	200	více jak 10 min	•	•	•	•	•	•	•	•	•	•
2	300	více jak 10 min	•	•	•	•	•	•	•	•	•	•
2	400	více jak 10 min	•	•	•	•	•	•	•	•	•	•
2	500	více jak 10 min	•	•	•	•	•	•	•	•	•	•
3	1	0,010	0	0,1	0	0	0	0	0	0	0	0
3	2	0,060	0,1	0	0,2	0	0	0,1	0	0,2	0	0
3	5	0,305	0,29	0,34	0,33	0,25	0,31	0,33	0,32	0,32	0,25	0,31
3	10	16,815	16,7	16,8	16,82	16,76	16,78	16,91	16,76	16,86	16,84	16,92
3	16	více jak 10 min	•	•	•	•	•	•	•	•	•	•
3	20	více jak 10 min	•	•	•	•	•	•	•	•	•	•
3	30	více jak 10 min	•	•	•	•	•	•	•	•	•	•
3	50	více jak 10 min	•	•	•	•	•	•	•	•	•	•
4	1	0,224	0,14	0,3	0,3	0,3	0	0,1	0	0	0,8	0,3
4	2	0,252	0,21	0,17	0,24	0,19	0,22	0,19	0,3	0,4	0,3	0,3
4	5	3,494	3,8	3,56	3,67	3,63	3,44	3,38	3,39	3,37	3,31	3,39
4	10	více jak 10 min	•	•	•	•	•	•	•	•	•	•
4	20	více jak 10 min	•	•	•	•	•	•	•	•	•	•
5	1	0,141	0,11	0,3	0,2	0	0	0,2	0	0,2	0,2	0,2
5	2	0,446	0,14	0,5	0,42	0,5	0,5	0,6	0,4	0,2	0,6	0,6
5	5	11,594	11,94	11,53	11,57	11,46	11,54	11,6	11,41	11,7	11,56	11,63
5	10	více jak 10 min	•	•	•	•	•	•	•	•	•	•

Tabulka 4.2: Výsledky testů GCD v systému Maxima

4.4 SINGULAR

SINGULAR je počítačový algebraický systém pro polynomiální výpočty se speciálním důrazem na komutativní algebru, algebraickou geometrii a teorii singularity. Má poměrně velkou uži-

vatelskou základnu a je jedním z nejrozšířenějších open source produktů počítačové algebry mezi odborníky. Z těchto důvodů byl také zvolen pro testování největšího společného dělitele.

Pro řešení problémů nekomutativní algebry lze využít následníka SINGULARu nazývaném PLURAL. SINGULAR byl navržený Gertem-Martinem Greuelem, Gerhardem Pfisterem a Hansem Schönemannem.

Hlavní vlastnosti:

- používané objekty - ideální/modely od velmi obecných polynomiálních okruhů po různá pole prvků,
- velká rozmanitost algoritmů implementovaných v jádru (napsaných v C/C++),
- intuitivní programovací jazyk podobný programovacímu jazyku C,
- mnoho dalších implementovaných algoritmů je dostupných v podobě knihoven SINGULARu,
- spolupracuje s dalšími PAS jako jsou MMA nebo MuPAD,
- rozsáhlá dostupná dokumentace na webu `http://www.singular.uni-kl.de`,
- vývoj od roku 1984; v současnosti je dostupná aktuální verze 3.0.2.

Hardwarové a softwarové nároky

- Volně dostupný pro většinu hardware a software platforem (Unix, Windows 95/NT, Macintosh) - přehled verzí ke stažení je na `ftp://www.mathematik.uni-kl.de/pub/Math/Singular`.
- Volně dostupný jako binární program pro Windows 95/98/ME/NT/2K/XP, Unix: Linux (PCs, DEC-Alpha), HP-UX (Hewlett-Packard), Solaris (Sun), IRIX (SGI), AIX (IBM), OSF (DEC), FreeBSD (PCs) Macintosh: MacOS X.

4.4.1 Generování testovacích polynomů

Pro generování polynomů byla použita knihovna `random.lib` a její procedura

```
sparsepoly(u[,o,p,b]); u,o,p,b integers
```

Výstupem procedury jsou polynomy pouze s termy stupně d, $u \leq d \leq o$, kde p procento termů stupně d je 0, zbývající mají náhodné koeficienty v rozmezí [1,b), (výchozí hodnoty: o=u, p=75, b=30000). Při vhodném nastavení lze získat polynomy blížící se polynomům, které by byly vhodné pro testovací účely viz kapitola 4.1.2 i když z názvu funkce se zdá, že generuje polynomy řídké (sparse). Příklad:

```
LIB "random.lib";
ring r=0,(x,y,z),dp;
sparsepoly(5);
"";
==> 24263xy4+24170x4z+21962x3yz+26642xy3z+5664xy2z2+17904xz4
==> sparsepoly(3,5,90,9);
==> 8x3z2+2y3z2+3xyz3+2xy3+yz3+xy2
```

Před každým použitím polynomu musí být nadefinován polynomiální okruh pomocí:

```
ring name = (coefficient_field), ( names_of_ring_variables ), ( ordering );
```

Testovací kód:

```
> int stupen=20;
> ring r=0,(x),lp;
> LIB "random.lib";
> poly g=sparsepoly(0,stupen,0,5000);
> poly f1=g*sparsepoly(0,stupen,0,5000);
> poly f2=g*sparsepoly(0,stupen,0,5000);
> rtimer=1;
> int cas=timer; poly gcdm=gcd(f1,f2);cas=timer-cas;cas;
> cas
```

Poznámka. Funkce `rtimer` slouží k nastavení zobrazení doby výpočtu.

Nastavení timeru SINGULARu pro větší přesnost: `Singular [options] [file1 [file2 ...]]`

`--min-time=SECS` Udává minimální zobrazovaný čas výpočtu (jako výchozí je nastaveno 0,5 s. Pro testování bylo nastaveno 0,001 s.)

`--ticks-per-sec=TICKS` Udává počet tiků za 1 sekundu (jako výchozí je nastaveno 1. Pro testování bylo nastaveno 1000.)

4.4.2 Možnosti výpočtu největšího společného dělitele

Singular nabízí nejpoužívanější algoritmy pro výpočet největšího společného dělitele. Pomocí funkce

```
gcd ( int_expression, int_expression )
gcd ( number_expression, number_expression )
gcd ( poly_expression, poly_expression )
```

můžeme získat největší společné dělitele požadovaných vstupů. Algoritmus `gcd` počítá největší společné dělitele polynomů jedné nebo více proměnných. Algoritmus výpočtu největšího společného dělitele nemůže uživatel zvolit, ale je volen přímo SINGULARem. Pokud je exponent prvočíslo je použita metoda subresultantů. Pro exponent 0 a v případě jedné proměnné je použit modulární algoritmus. Pro více proměnných je použit EZGCD. Nelze počítat `gcd` polynomů s racionálními čísly Q.

4.4.3 Výsledky testů největšího společného dělitele

Tabulka 4.3 shrnuje provedené testy výpočtu největšího společného dělitele. Pokud je v některém poli černá tečka znamená to, že test probíhal příliš dlouho viz kapitola 4.1.2.

Porovnání a zhodnocení s ostatními testovanými systémy je uvedeno v poslední kapitole.

4.5 Giac/Xcas

Protože Giac/Xcas je kombinace vlastně dvou nástrojů/knihoven byl záměr testování výpočtů největšího společného dělitele takový, aby se zjistilo, zda poměrně jednoduchá, ale zato modulární knihovna použitelná v dalších aplikacích naprogramovaných uživatelem, bude srovnatelná nebo se bude aspoň přibližovat některému "kompletnímu" počítačovému algebraickému systému.

proměnné	stupeň	t[s] průměr	průběžné hodnoty									
1	1	0,000	0	0	0	0	0	0	0	0	0	0
1	2	0,000	0	0	0	0	0	0	0	0	0	0
1	5	0,000	0	0	0	0	0	0	0	0	0	0
1	10	0,000	0	0	0	0	0	0	0	0	0	0
1	50	0,002	0,016	0	0	0	0	0	0	0	0	0
1	100	0,016	0,016	0,016	0,016	0,015	0,016	0,015	0,016	0,016	0,016	0,016
1	200	0,047	0,047	0,047	0,046	0,047	0,046	0,047	0,047	0,047	0,047	0,046
1	300	0,095	0,09	0,109	0,094	0,094	0,094	0,094	0,094	0,094	0,094	0,094
1	400	0,158	0,156	0,172	0,157	0,157	0,156	0,156	0,156	0,157	0,157	0,156
1	500	0,195	0,187	0,187	0,188	0,219	0,204	0,203	0,188	0,188	0,187	0,203
1	1000	0,461	0,5	0,453	0,468	0,469	0,453	0,453	0,453	0,453	0,469	0,437
1	2000	1,428	1,391	1,438	1,422	1,421	1,438	1,437	1,454	1,422	1,453	1,406
1	5000	7,891	7,641	7,828	7,922	7,907	7,891	7,937	7,922	7,953	7,969	7,937
2	1	0,000	0	0	0	0	0	0	0	0	0	0
2	2	0,000	0	0	0	0	0	0	0	0	0	0
2	5	0,016	0,016	0,016	0,015	0,015	0,015	0,016	0,016	0,015	0,016	0,015
2	10	0,125	0,125	0,125	0,125	0,125	0,125	0,125	0,125	0,125	0,125	0,125
2	20	0,908	0,906	0,906	0,906	0,906	0,922	0,906	0,907	0,906	0,906	0,907
2	50	51,806	51,734	51,703	51,641	52,266	51,656	51,953	51,625	51,656	51,907	51,922
2	100	1481,625	1481,625	•	•	•	•	•	•	•	•	•
2	160	více jak 10 min	•	•	•	•	•	•	•	•	•	•
2	200	více jak 10 min	•	•	•	•	•	•	•	•	•	•
2	300	více jak 10 min	•	•	•	•	•	•	•	•	•	•
2	400	více jak 10 min	•	•	•	•	•	•	•	•	•	•
2	500	více jak 10 min	•	•	•	•	•	•	•	•	•	•
3	1	0,000	0	0	0	0	0	0	0	0	0	0
3	2	0,000	0	0	0	0	0	0	0	0	0	0
3	5	0,080	0,078	0,078	0,078	0,078	0,078	0,078	0,078	0,078	0,094	0,079
3	10	1,016	1,032	1,015	1,016	1,015	1,015	1,015	1,016	1,016	1	1,015
3	16	13,239	13,156	13,156	13,219	13,266	13,281	13,313	13,234	13,265	13,266	13,235
3	20	50,142	50,016	49,953	49,859	50,156	49,937	50,375	50,391	50,281	50,156	50,297
3	50	635,103	635,531	634,313	634,469	634,61	634,282	636,188	634,907	635,078	637,406	634,25
4	1	0,000	0	0	0	0	0	0	0	0	0	0
4	2	0,010	0,016	0,015	0,016	0,016	0,016	0	0	0,016	0	0
4	5	0,266	0,265	0,266	0,266	0,265	0,266	0,265	0,266	0,266	0,266	0,265
4	10	12,386	12,421	12,375	12,391	12,391	12,375	12,375	12,391	12,375	12,391	12,375
4	20	více jak 10 min	•	•	•	•	•	•	•	•	•	•
5	1	0,000	0	0	0	0	0	0	0	0	0	0
5	2	0,017	0,031	0,016	0,016	0,015	0,015	0,015	0,015	0,015	0,015	0,015
5	5	0,872	0,875	0,89	0,86	0,859	0,875	0,859	0,859	0,859	0,875	0,906
5	10	124,937	124,687	124,718	124,875	124,703	124,657	124,734	125,984	124,578	125,609	124,828

Tabulka 4.3: Výsledky testů GCD v systému SINGULAR

Rozhraní Xcas bylo zvoleno jako jedno z jednodušších a přehledných rozhraní pro Giac přidávájící navíc tabulkový procesor.

Giac/Xcas je free počítačový algebraický systém. Vlastnosti:

- mód kompatibility pro Maple, MuPAD a TI89,
- Xcas je rozhraní, jenž má za cíl zlepšit výkonnost a uživatelskou přívětivost programu,

- Giac/Xcas lze použít pro počítačovou algebru, geometrii (2–D a 3–D) a jako tabulkový procesor,
- Giac je free (GPL) C++ knihovna, jenž provádí všechny výpočty, může být použita v různých programech vytvořených v jazyce C++.

Výpočty:

- počítačová algebra: libovolná přesnost celých čísel a čísel v plovoucí řádové čárce, celých čísel a polynomiální aritmetika (gcd, extended gcd, faktorizace, ...), zjednodušování, řešení rovnic, částečná dekompozice zlomků, kalkulus (derivace, integrace, limity, rozvoj řad), lineární algebra (vektory, matice, redukce řádků do echelonovy formy, vlastní čísla a vlastní vektory),
- 2–D geometrie: bod, úsečka, přímka, trojúhelník, mnohoúhelník, kružnice, parametrické křivky, průsečíky, tečny... dynamická geometrie, všechny geometrické příkazy jsou programovatelné,
- 3–D geometrie: bod, úsečka, přímka, rovina, trojúhelník, mnohoúhelník, kružnice, parametrické křivky, povrchy, průsečíky, tečny..., dynamická geometrie, všechny geometrické příkazy jsou programovatelné,
- tabulkový procesor : relativní a absolutní odkazy, buňky mohou obsahovat formální objekty (např. 1/2, sin(x), ...), 1–D a 2–D statistika (střední hodnota - mean, stddev, hgistogram, kovariance - covariance, regrese - regressions, ...),
- programování : funkce, lokální proměnné, testy, cykly, možnost volby syntaxe (C, Maple, MuPAD, TI89), většinou je možné spustit Maple, MuPAD nebo TI89 bez zásahu uživatele do Xcasu, programový editor, interaktivní debugger.

Hardwarové a softwarové nároky

Dostupný pro všechny běžné platformy a systémy Windows, Mac OS X a Linux/Unix.

4.5.1 Generování testovacích polynomů

Systém obsahuje funkci pouze pro generování náhodných polynomů jedné proměnné, v ostatních případech musel být použit export polynomů ze systému Maple. Funkce pro generování náhodných polynomů:

```
randPoly([Var(var)],Intg(n))
```

`Var(var)` - označení proměnné
`Intg(n)` - stupeň polynomu

4.5.2 Možnosti výpočtu největšího společného dělitele

Jsou dostupné funkce `modgcd` pro počítání pomocí modulárního algoritmu, `ezgcd` pro výpočet největšího společného dělitele polynomů s nejméně dvěma proměnnými pomocí algoritmu ezgcd, `psrgcd` pro počítání pomocí subresultantů, `heugcd` pro počítání pomocí algoritmu nazývaném heuristická pgcd. Jaký algoritmus je použit při volání funkce `gcd(poly1,poly2)`se nepodařilo zjistit, avšak lze předpokládat, že je volen nejvhodnější algoritmus[3] pro dané vstupy.

[3]nejvhodnější z pohledu daného systému, což neznamená, že je pro danou aplikaci nejrychlejší

4.5.3 Výsledky testů největšího společného dělitele

Tabulka 4.4 shrnuje provedené testy výpočtu největšího společného dělitele. Pokud je v některém poli černá tečka znamená to, že test probíhal příliš dlouho viz kapitola 4.1.2.

Porovnání a zhodnocení s ostatními testovanými systémy je uvedeno v poslední kapitole.

proměnné	stupeň	t[s] průměr	průběžné hodnoty									
1	1	0,008	0,0086	0,007	0,0078	0,00785	0,0086	0,00785	0,0078	0,00545	0,0086	0,00855
1	2	0,015	0,0125	0,022	0,014	0,0125	0,0141	0,0141	0,0141	0,0141	0,022	0,0141
1	5	0,033	0,0219	0,0219	0,0312	0,039	0,055	0,0314	0,0282	0,039	0,0282	0,0344
1	10	0,057	0,062	0,0545	0,0625	0,028	0,054	0,0625	0,07	0,039	0,055	0,0775
1	50	0,200	0,171	0,234	0,235	0,235	0,234	0,218	0,188	0,155	0,14	0,187
1	100	0,302	0,296	0,297	0,36	0,297	0,282	0,296	0,249	0,313	0,313	0,313
1	200	0,622	0,671	0,546	0,657	0,593	0,703	0,579	0,626	0,578	0,751	0,516
1	300	1,072	1,093	1,11	1,047	1,14	1,078	0,999	1	1,094	1,062	1,094
1	400	1,438	1,501	1,734	1,376	1,422	1,407	1,359	1,407	1,313	1,437	1,421
1	500	1,899	1,798	1,844	1,954	1,875	1,968	1,922	1,939	1,906	1,828	1,954
1	1000	5,095	5,797	4,922	4,953	5,001	5,094	5,078	4,984	5,016	5,109	5
1	2000	15,473	15,827	16	15,704	15,343	15,89	15,938	14,562	15,719	15,031	14,719
2	1	0,027	0,028	0,028	0,035	0,0282	0,028	0,0282	0,0282	0,028	0,025	0,0172
2	2	0,098	0,109	0,124	0,14	0,078	0,125	0,0545	0,11	0,055	0,0705	0,109
2	5	0,253	0,188	0,218	0,233	0,281	0,329	0,218	0,282	0,234	0,281	0,266
2	10	0,688	0,719	0,671	0,657	0,656	0,765	0,656	0,673	0,671	0,688	0,719
2	20	2,553	2,484	2,515	2,624	2,516	2,608	2,469	2,641	2,515	2,625	2,531
2	50	7,439	7,78	7,313	7,578	6,938	7,547	7,344	7,359	7,61	7,422	7,5
2	100	málo paměti	•	•	•	•	•	•	•	•	•	•
2	160	více jak 10 min	•	•	•	•	•	•	•	•	•	•
2	200	více jak 10 min	•	•	•	•	•	•	•	•	•	•
2	300	více jak 10 min	•	•	•	•	•	•	•	•	•	•
2	400	více jak 10 min	•	•	•	•	•	•	•	•	•	•
2	500	více jak 10 min	•	•	•	•	•	•	•	•	•	•
3	1	0,042	0,039	0,0406	0,0374	0,0625	0,0376	0,039	0,039	0,0346	0,0344	0,055
3	2	0,249	0,391	0,235	0,219	0,266	0,265	0,219	0,219	0,234	0,251	0,187
3	5	1,444	1,344	1,329	1,469	1,484	1,422	1,656	1,343	1,516	1,437	1,437
3	10	9,194	8,906	9,188	8,766	9,5	9,155	9,203	9,626	8,891	9,125	9,578
3	16	10,454	11,906	9,812	9,109	9	10,844	11,765	10,109	11,656	9,297	11,046
3	20	více jak 10 min	•	•	•	•	•	•	•	•	•	•
3	30	více jak 10 min	•	•	•	•	•	•	•	•	•	•
3	50	více jak 10 min	•	•	•	•	•	•	•	•	•	•
4	1	0,141	0,125	0,155	0,156	0,11	0,124	0,125	0,157	0,157	0,156	0,14
4	2	0,224	0,236	0,156	0,187	0,281	0,187	0,234	0,282	0,266	0,156	0,25
4	5	6,736	6,594	6,625	6,905	6,875	6,703	6,75	6,673	6,875	6,641	6,719
4	10	více jak 10 min	•	•	•	•	•	•	•	•	•	•
4	20	více jak 10 min	•	•	•	•	•	•	•	•	•	•
5	1	0,147	0,172	0,156	0,141	0,171	0,156	0,14	0,14	0,124	0,156	0,11
5	2	0,247	0,265	0,25	0,281	0,296	0,297	0,171	0,203	0,234	0,203	0,265
5	5	27,530	26,907	26,64	28,703	27,048	27,235	26,656	28,609	28,406	26,546	28,546
5	10	více jak 10 min	•	•	•	•	•	•	•	•	•	•

Tabulka 4.4: Výsledky testů GCD v systému Giac/Xcas

4.6 CoCoA

CoCoA je volně dostupný systém pro počítání s polynomy o více proměnných. Je to poměrně nově vzniklý systém, aspoň co se týče jeho grafického uživatelského rozhraní, jenž je snadno ovladatelné a přehledné s rozdělenou částí pro zadání vstupů a vypočítaných výstupů. Práce s polynomy o více proměnných je důležitou vlastností k počítání největšího společného dělitele těchto polynomů.

Speciálně se CoCoA zabývá počítáním okruhů polynomů s více proměnnými nad racionálními nebo modulárními celými čísly a jejich ideály a moduly. Implementace teorie operací ideálů/modulů se spoléhá na teorii Gröbnerových bází.

Jedna z nejdůležitějších vlastností CoCoA je její vysokoúrovňový programovací jazyk jenž umožňuje uživateli napsat své vlastní funkce a směrovat systém postupně k řešení komplikovaných a složitých výpočtů. Tento jazyk byl navržen tak, aby byl přirozený a intuitivní, proto je jednoduché se ho naučit. Jazyk je interpretovaný; jeho syntax je podobná Pascalu.

Hlavními uživateli CoCoA jsou výzkumníci v komutativní algebře a algebraické geometrii. Nicméně výpočetní algebraické techniky jsou rozšířeny v mnoha oborech (například numerická analýza, kryptografie, statistika a dynamické systémy), a proto jsou také oblastí kde lze CoCoA využít.

Systém obsahuje kompletní on-line nápovědu (taktéž dostupnou jako html manuál). CoCoA je volně dostupný software pro výzkumné a vzdělávací účely.

Pro pohodlnou práci nabízí systém textový interface, Emacs mód a grafické uživatelské rozhraní společné pro většinu platforem.

CoCoA je speciálně navržena pro následující výpočty:

- Gröbnerovy báze,
- polynomiální faktorizace,
- exaktní lineární algebra,
- Hilbertovy funkce,
- ideály bodů,
- teorie ideálů.

Většina verzí CoCoA má grafické uživatelské rozhraní, využívající programovací jazyk CoCoALanguage. S knihovnou CoCoALib, a C++ library, je možné použít výpočetního jádra CoCoA pro vývoj vlastních aplikací.

4.6.1 Generování testovacích polynomů

Před použitím polynomiálních výpočtů s polynomy více než tří neznámých je nutno nadefinovat nový polynomiální okruh pomocí `Use R::=Q[var1,var2,..,varn]`, náhodné polynomy jsou generovány pomocí funkce `Randomized(expression)`. Proměnná `S` označuje výsledný stupeň polynomu. Použití:

```
Use R::=Q[x,y,z,u];
S:=1;
T:=Randomized((x+y+z+u+1)^S);
F:=T*Randomized((x+y+z+u+1)^S);
G:=T*Randomized((x+y+z+u+1)^S);
Time H:=GCD(F,G);
```

4.6.2 Možnosti výpočtu největšího společného dělitele

CoCoA obsahuje jedinou funkci pro počítání největšího společného dělitele:

```
GCD(F_1:INT,...,F_n:INT):INT
GCD(L:LIST of INT):INT
GCD(F_1:POLY,...,F_n:POLY):POLY
GCD (L:LIST of POLY):POLY
```

Algoritmus jaký při výpočtech uplatňuje se nepodařilo zjistit. Lze předpokládat, že zná většinu algoritmů a použije vždy nejvhodnější podle charakteru vstupů.

4.6.3 Výsledky testů největšího společného dělitele

Tabulka 4.5 shrnuje provedené testy výpočtu největšího společného dělitele. Pokud je v některém poli černá tečka znamená to že test probíhal příliš dlouho viz kapitola 4.1.2.

Porovnání a zhodnocení s ostatními testovanými systémy je uvedeno v poslední kapitole.

proměnné	stupeň	t[s] průměr	průběžné hodnoty									
1	1	0,006	0,01	0	0	0,01	0	0,01	0	0	0	0,03
1	2	0,006	0	0,01	0,01	0,01	0,01	0	0	0	0,01	0,01
1	5	0,007	0,01	0,01	0,01	0,01	0	0,01	0	0	0,01	0,01
1	10	0,005	0,01	0	0	0	0	0,01	0,01	0	0,01	0,01
1	50	0,008	0,01	0	0,01	0,01	0	0,01	0,01	0,01	0,01	0,01
1	100	0,010	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01
1	200	0,037	0,04	0,03	0,04	0,04	0,04	0,03	0,04	0,04	0,03	0,04
1	300	0,071	0,070	0,07	0,06	0,09	0,07	0,07	0,07	0,07	0,07	0,07
1	400	0,130	0,12	0,12	0,12	0,12	0,14	0,14	0,14	0,12	0,14	0,14
1	500	0,266	0,25	0,28	0,26	0,26	0,28	0,28	0,26	0,25	0,28	0,26
1	1000	0,816	0,81	0,82	0,82	0,82	0,82	0,82	0,81	0,82	0,81	0,81
1	2000	2,967	2,95	2,96	3,04	2,93	2,96	2,96	2,98	2,98	2,95	2,96
2	1	0,010	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01
2	2	0,026	0,03	0,01	0,03	0,01	0,03	0,03	0,03	0,03	0,03	0,03
2	5	0,061	0,06	0,06	0,06	0,06	0,06	0,06	0,06	0,06	0,07	0,06
2	10	0,263	0,28	0,25	0,26	0,26	0,26	0,26	0,26	0,28	0,26	0,26
2	20	2,258	2,34	2,21	2,23	2,2	2,23	2,21	2,23	2,32	2,35	2,26
2	50	97,110	95,28	92,65	95,54	101,32	93	101,34	101,81	101,4	95,7	93,06
2	100	více jak 10 min	1777,79	•	•	•	•	•	•	•	•	•
2	160	více jak 10 min	•	•	•	•	•	•	•	•	•	•
2	200	více jak 10 min	•	•	•	•	•	•	•	•	•	•
2	300	více jak 10 min	•	•	•	•	•	•	•	•	•	•
2	400	více jak 10 min	•	•	•	•	•	•	•	•	•	•
2	500	více jak 10 min	•	•	•	•	•	•	•	•	•	•
3	1	0,030	0,03	0,03	0,03	0,03	0,03	0,03	0,03	0,03	0,03	0,03
3	2	0,054	0,06	0,06	0,06	0,06	0,04	0,06	0,04	0,06	0,04	0,06
3	5	0,331	0,32	0,34	0,28	0,34	0,34	0,32	0,34	0,35	0,34	0,34
3	10	8,588	8,59	8,5	8,5	8,64	8,62	8,56	8,67	8,64	8,57	8,59
3	16	136,200	134,6	135,53	134,89	136,48	137,37	137,39	136.36	136,35	136,23	136,96
3	20	501,045	488,73	500,14	498,56	499,23	502	505,31	505,37	506,65	497,04	507,42
3	30	více jak 10 min	•	•	•	•	•	•	•	•	•	•
3	50	více jak 10 min	•	•	•	•	•	•	•	•	•	•
4	1	0,053	0,06	0,04	0,06	0,06	0,06	0,04	0,07	0,04	0,06	0,04
4	2	0,124	0,12	0,12	0,12	0,12	0,14	0,14	0,12	0,14	0,1	0,12
4	5	2,784	2,81	2,82	2,81	2,81	2,78	2,78	2,76	2,76	2,76	2,75
4	10	375,924	367,6	378,75	373,9	378,53	368,26	378	379,29	376,81	380,12	377,98
4	20	více jak 10 min	•	•	•	•	•	•	•	•	•	•
5	1	0,096	0,1	0,1	0,1	0,09	0,09	0,1	0,11	0,09	0,09	0,09
5	2	0,206	0,2	0,2	0,21	0,18	0,17	0,21	0,21	0,21	0,21	0,26
5	5	25,304	25,18	25,23	25,32	25,2	25,29	25,31	25,4	25,39	25,4	25,32
5	10	více jak 10 min	•	•	•	•	•	•	•	•	•	•

Tabulka 4.5: Výsledky testů GCD v systému CoCoA

Kapitola 5

Závěr a porovnání jednotlivých systémů

Jak je z tabulek a grafů níže vidět, některé systémy svým výkonem téměř zcela propadly. Například systém **Maxima**, jenž je mnohdy porovnáván se systémem **Maple** byl nejpomalejší z testovaných systémů a to ve všech ohledech. Možná příčina je v tom, že vývojáři se snaží do systému zabudovat co možná nejvíc funkcí, ale nekoukají na odladění systému a mnohdy použití neefektivních algoritmů. Je sice pravda, že uživatel může nastavit algoritmus výpočtu největšího společného dělitele podle sebe, avšak jako výchozí je použita metoda subresultantů, která je pro polynomy poměrně pomalá. Pokud však uživatel se systémem začne pracovat není jisté zda ihned bude znát všechna nastavení, proto byly systémy testovány s nastavením výchozím. Ve všech ostatních systémech je použitý algoritmus volen automaticky podle vstupních hodnot.

Lze vidět, že naopak nejrychlejším systémem je **Maple**. Jeho rychlost je dána také tím, že jako jediný dokázal využít potenciál harware na němž byly systémy testovány a to dvoujádrový procesor. Jediný **Maple** zatěžoval při výpočtech obě jádra, ostatní systémy využívaly jen jedno jádro a výkonnost se tedy snížila téměř o 50 %. Pokud je uživatel ochoten zaplatit částky v řádech desetitisíců je Maple nejlepší volba. **Maple** je využitelný v řadě dalších oblastech, takže investice se jistě vyplatí.

Příjemným překvapením byl systém **SINGULAR**, který byl druhý nejlepší. Ve většině případů byl sice až desetkrát pomalejší než **Maple**, avšak při počítání nižších stupňů byly časy srovnatelné. Je vidět, že **SINGULAR** si zaslouží být mezi nejpoužívanějšímy open-source systémy počítačové algebry.

Systém **CoCoA** má, jak se zdá, velký potenciál se rozšířit, protože jeho uživatelské rozhraní je přehledné a uživatel se s ním seznámí rychle. Nápověda poskytovaná jako html dokumentace je komplexní a lze v ní najít vše, co systém nabízí přehledně vysvětleno. Největší jeho výhodou při počítání největšího společného dělitele je jeho rychlost, která je srovnatelná se systémem **SINGULAR**.

Knihovna **Giac** je ve výsledku taktéž použitelnou alternativou a vzhledem k tomu, že může být použita ve vlastních aplikacích je její výkonnost dostačující. Rychlost výpočtů byla také ovlivněna právě použitým rozhraním **Xcas**, ale i tak jsou časy mnohem menší než v případě **Maximy.**

Z tabulky 5.1 je vidět, že při testování polynomů jedné proměnné byl čas téměř zanedbatelný, což se projevilo i na dobách výpočtů s polynomy řádu 1 až 100, kdy systém nebyl ani schopen čas přesně měřit (tak malé hodnoty měřit nedokáže). Ve většině případů se projevila závislost

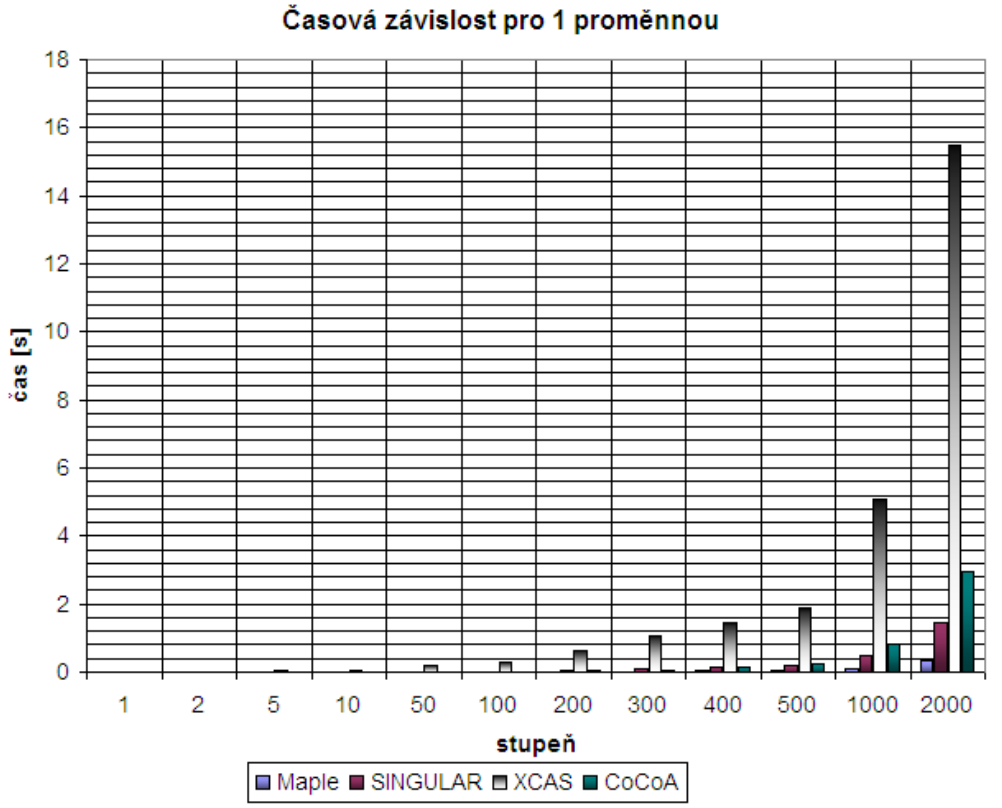
mezi pořadím provádění výpočtu, kdy první výpočet trval téměř vždy delší dobu.

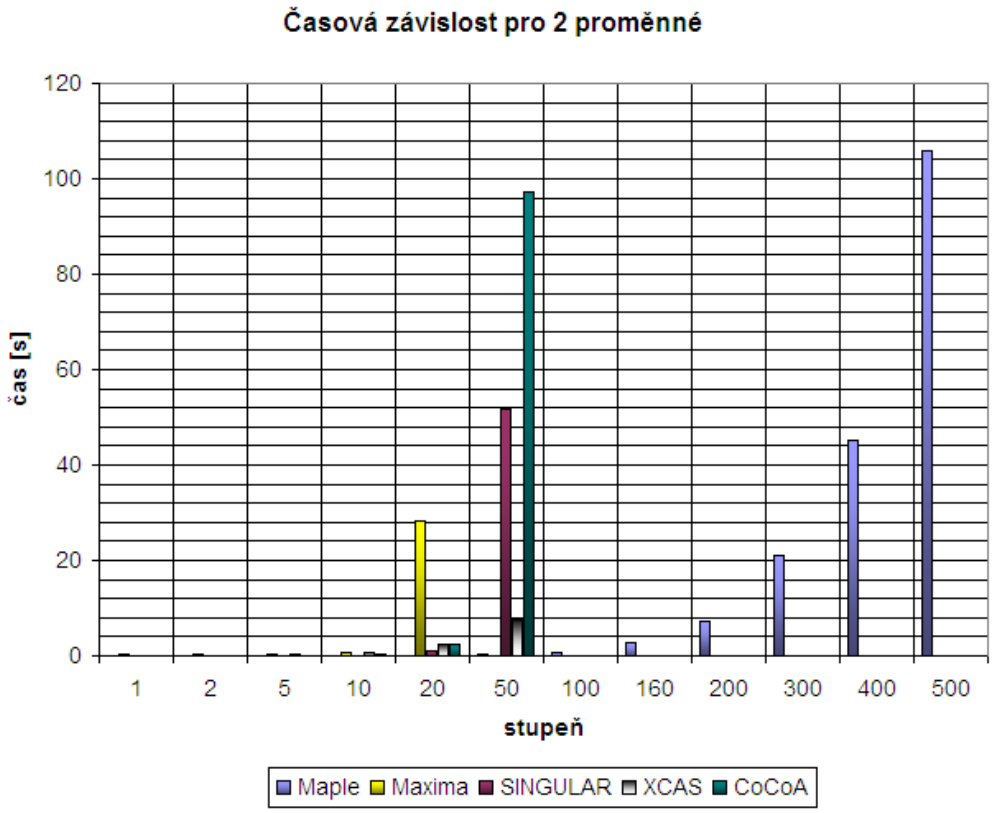
Pokud je v některém poli černá tečka znamená to, že test nebyl dokončen, neboť doba testu byla delší než stanovená hranice 10 minut.

		t[s] průměr				
proměnné	stupeň	Maple	Maxima	SINGULAR	XCAS/GINAC	CoCoA
	1	0,000	0,000	0,000	0,008	0,006
	2	0,002	0,000	0,000	0,015	0,006
	5	0,000	0,030	0,000	0,033	0,007
	10	0,000	0,030	0,000	0,057	0,005
	50	0,006	0,550	0,002	0,200	0,008
	100	0,002	0,503	0,016	0,302	0,010
	200	0,016	5,911	0,047	0,622	0,037
	300	0,020	23,704	0,095	1,072	0,071
	400	0,025	44,484	0,158	1,438	0,130
	500	0,027	114,424	0,195	1,899	0,266
	1000	0,077	677,760	0,461	5,095	0,816
1	2000	0,358	•	1,428	15,473	2,967
	1	0,019	0,360	0,000	0,027	0,010
	2	0,002	0,210	0,000	0,098	0,026
	5	0,014	0,302	0,016	0,253	0,061
	10	0,016	0,519	0,125	0,688	0,263
	20	0,042	28,258	0,908	2,553	2,258
	50	0,175	•	51,806	7,439	97,110
	100	0,757	•	1481,625	•	•
	160	2,880	•	•	•	•
	200	7,220	•	•	•	•
	300	21,039	•	•	•	•
	400	45,322	•	•	•	•
2	500	105,783	•	•	•	•
	1	0,017	0,010	0,000	0,042	0,030
	2	0,003	0,060	0,000	0,249	0,054
	5	0,020	0,305	0,080	1,444	0,331
	10	0,278	16,815	1,016	9,194	8,588
	16	2,458	•	13,239	10,454	136,200
	20	3,673	•	50,142	•	501,045
	30	44,612	•	635,103	•	•
3	50	594,791	•	•	•	•
	1	0,023	0,224	0,000	0,141	0,053
	2	0,003	0,252	0,010	0,224	0,124
	5	0,244	3,494	0,266	6,736	2,784
	10	1,272	•	12,386	•	375,924
4	20	62,518	•	•	•	•
	1	0,005	0,141	0,000	0,147	0,096
	2	0,009	0,446	0,017	0,247	0,206
	5	0,369	11,594	0,872	27,530	25,304
5	10	7,564	•	124,937	•	•

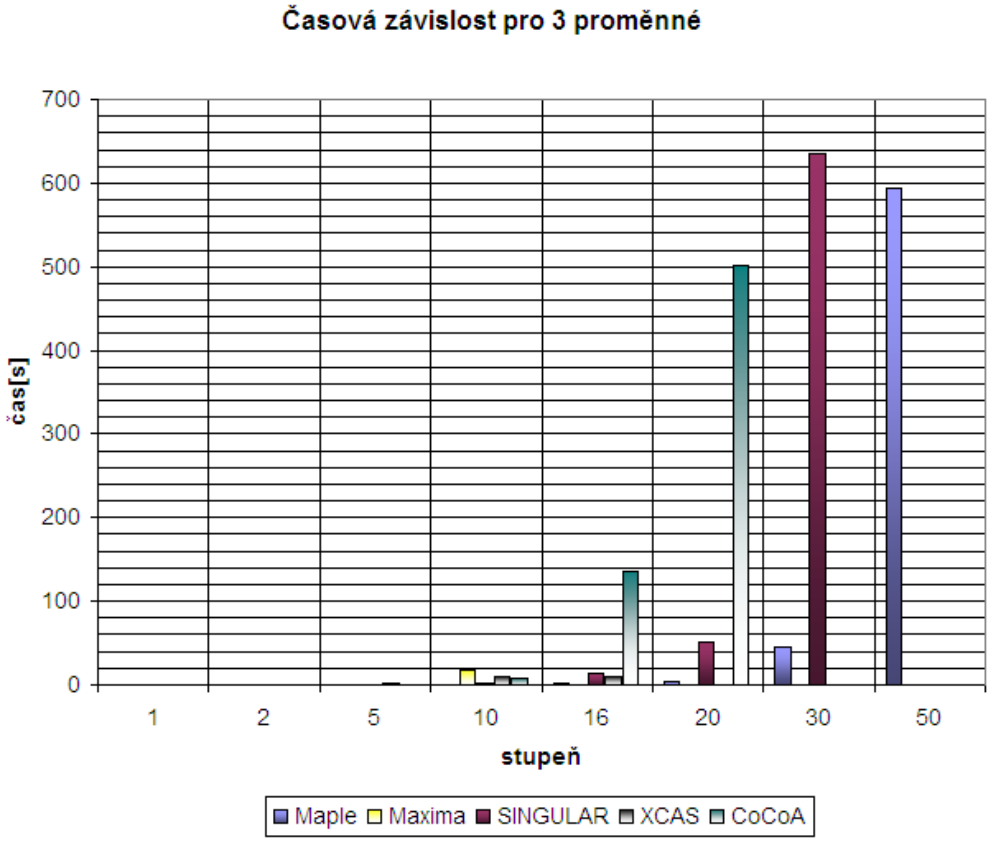
Tabulka 5.1: Porovnání času výpočtů GCD na jednotlivých systémech.

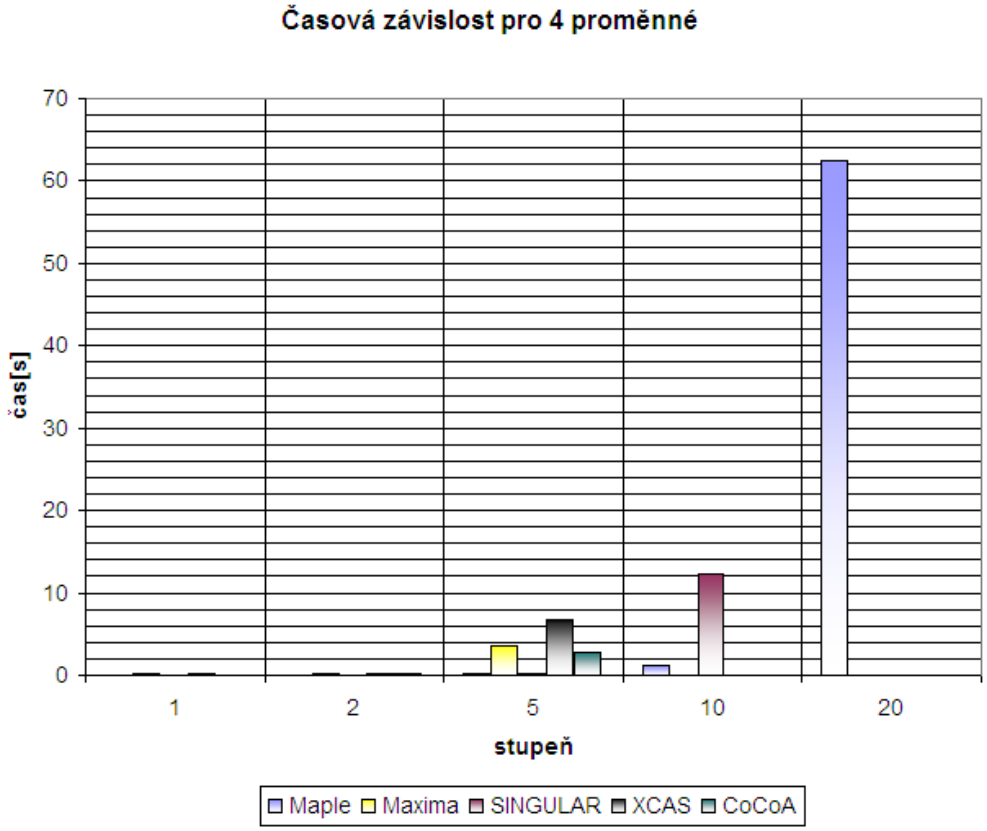
Poznámka. V prvním grafu pro časovou závislost polynomů s 1 proměnnou není zahrnut systém Maxima, neboť hodnoty časů jsou mnohem větší než ostatních systémů a graf by tak ztratil vypovídací hodnotu. To je patrné i u ostatních grafů, ale v nich už byly použity všechny výsledky. U polynomů s více jak 1 proměnnou některé grafy neobsahují všechny hodnoty, neboť tyto časy byly delší jak 10 minut. Pokud některý systém není ve vyšším stupni zastoupen, ale v předchozím ano, neznamená to, že čas byl ve vyšším stupni nulový.



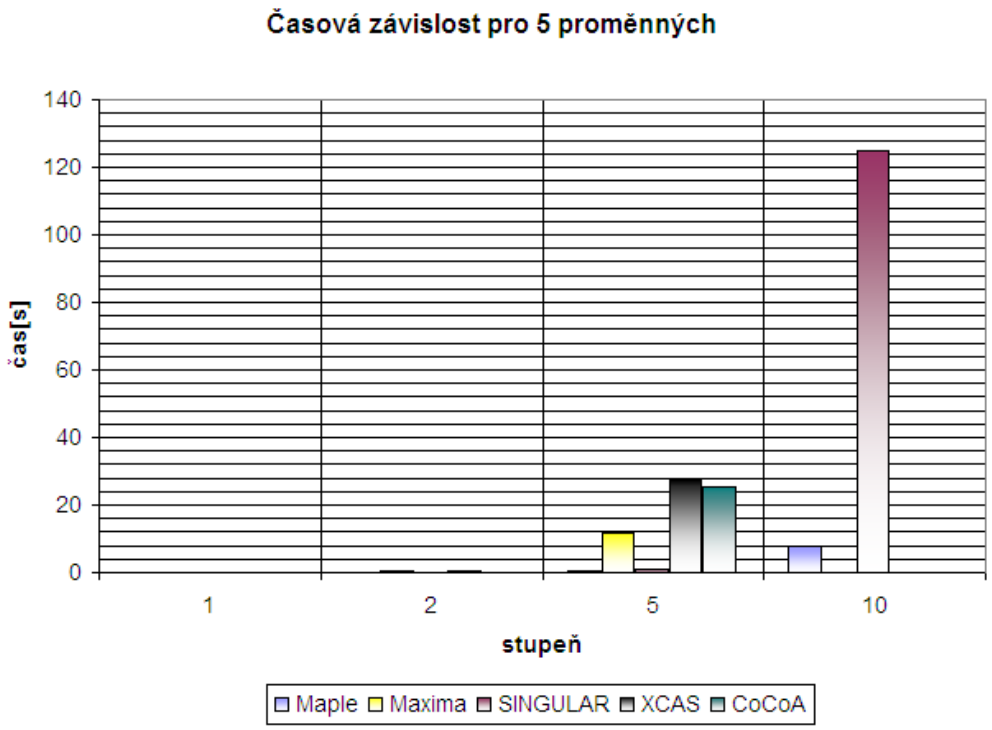


Obrázek 5.1: Grafy zobrazující dobu výpočtu GCD na jednotlivých systémech.





Obrázek 5.2: Grafy zobrazující dobu výpočtu GCD na jednotlivých systémech.



Obrázek 5.3: Grafy zobrazující dobu výpočtu GCD na jednotlivých systémech.

Literatura

[1] odkazy na webové stránky počítačových algebraických systémů
http://directory.google.com/Top/Science/Math/Algebra/Software/
http://www.cs.ru.nl/~freek/digimath/xindex.html

[2] R. Liška et al. *Počítačová Algebra, Algoritmy, Systémy a Aplikace.* Říjen 29, 1998
http://kfe.fjfi.cvut.cz/~liska/

[3] M. Lavaud. *Thema Free Software and Research. Univ. Orléans, LSM 2003*
http://www.univ-orleans.fr/EXT/ASTEX/astex/doc/en/congres/rmll2003/html/a1rll.htm

[4] Wolfram Mathworld. *Greatest Common Divisor.*
http://mathworld.wolfram.com/GreatestCommonDivisor.html

[5] CS Wikipedia. *Euklidův algoritmus.*
http://cs.wikipedia.org/wiki/Euklidův_algoritmus

[6] R. H. Lewis, M. Wester. *Comparison of Polynomial-Oriented Computer Algebra Systems (Preliminary Report).*
http://www.fordham.edu/lewis/cacomp.html

[7] Wikipedia. *Comparison of computer algebra systems.*
http://en.wikipedia.org/wiki/Comparison_of_computer_algebra_systems

[8] O. Modrlák. *Číslicové řízení - Syntéza číslicových regulátorů - Studijní materiály.* Katedra řídicí techniky - Fakulta mechatroniky a mezioborových inženýrských studií - TECHNICKÁ UNIVERZITA V LIBERCI, Květen 2004

[9] M. Bulant. *Diplomová práce - Systémy počítačové algebry a algebra - Algoritmy na výpočet největšího společného dělitele.* Brno, duben 1997

[10] A. Klimentová, M. Šebek. *Řešení symbolických polynomiálních rovnic.* Fakulta elektrotechnická České vysoké učení technické v Praze

[11] M. Bulant. *Algebra 2 - Teorie čísel.* katedra matematiky, Přírodovědecká fakulta, Masarykova univerzita, Brno

[12] M. Šebek. *Multi - dimensional systems: Control via polynomial techniques.* Dr.Sc. thesis, Czech Academy of Sciences, Prague, 1994

[13] M. Demlová, J. Nagy. *Algebra.* SNTL Praha, 1985

[14] A. Němeček, *Přednáškové slidy předmětu X01PAS.* Fakulta elektrotechnická České vysoké učení technické v Praze, 2007