西安电子科技大学

硕士学位论文

Grobner基生成算法的并行

姓名：狄鹏

申请学位级别：硕士

专业：通信与信息系统

指导教师：马文平

20080501

摘 要

Gröbner 基（Gröbner Bases）理论是计算机代数的一个基石，因为不仅可以知道 Gröbner 基存在性，而且更为关键的是提出了计算 Gröbner 基的可行性算法，所以无论是在理论上还是在计算上 Gröbner 基都起着巨大的作用，近年来 Gröbner 基理论的应用已经愈来愈广泛。实际上，对于与多项式相关的或者是能够将问题转化为与多项式相关的问题，Gröbner 基的理论和技巧就能发挥重要做用，于是 Gröbner 基做为一种强有力的辅助工具而被广泛应用于密码学及其相关领域。然而基本的计算 Gröbner 基的生成算法的计算效率是很低的，因此在一定程度上影响了 Gröbner 基的实用价值。由此我们提出在现有生成算法的基础之上以并行化的方式提高其中间计算效率。

本文从介绍 Gröbner 基入手，首先简要介绍了求解 Gröbner 基的基本理论、基本生成算法及其作用域；进而从现有的求解 Gröbner 基的主流生成算法出发，先对求解算法进行分析，再介绍算法的并行相关问题及影响算法效率的关键——中间项的约化，这也是我们所主要关注的地方，我们在此过程中采用 C+MPI 来进行并行处理。文中首先采用结构化高斯消元法对可能产生大型稀疏矩阵予以简化，而后采用并行高斯全选主元消去法对中间项进行约化，以达到提高算法实现效率的目的；最后将 Gröbner 基方法应用于零知识证明方式的身份认证，提出以采用并行的基于 Gröbner 基的零知识证明与部分盲签名相结合的方式进行安全电子支付的模型，并对其交易过程和安全性进行了分析。

关键词：**Gröbner 基，MPI，高斯消元法，零知识证明**

Abstract

The theory of Gröbner Bases is one of the basis for Computer—Algebra. Gröbner Bases is great significance in theory and computation. It is not only for being achieved the existence, but the more important is to be presented feasible algorithms of computing Gröbner Bases. In the past years, the theory of Gröbner Bases had been found more and more applications. In fact, theory of Gröbner Bases and its technology will play a big role to solve questions, which are on the polynomials or which can be converted into ones on the polynomials. Therefore, Gröbner Bases has been widely applied as a kind of powerful tool in Cryptography and relevant fields. It is very low of the computational efficiency of basic computing Gröbner Bases algorithms to diminish the applications of Gröbner Bases. So we propose to do parallelling based on existing generating algorithms to raise the efficiency of computation.

In this thesis, firstly, we briefly introduce theory of Gröbner Bases: basic theory and algorithms to achieve Gröbner Bases, also with application domain. Then with the mainstream generating algorithms, we make analysis of these algorithms, introduce the associative paralleling problems and the key factor influencing the algorithms efficiency----to reduce the middle items, which is what we mainly focus on. In the paralleling process we use C+MPI to execute. At the beginning, Structured Gaussian Elimination method is used to reduce the probably existing huge sparse matrices, then Parallel Complete Gaussian Pivoting Elimination is used to reduce the middle items, and both for improving the algorithms efficiency. Next we apply Gröbner Bases method into ZKP(Zero—knowledge Proof) for identity authority, and we propose Electronic Cash Payment System model to use parallelling ZKP identity authority system based on Gröbner Bases and Part Blind Signature, Finally with analysis of its transactions courses and security.

Keyword: Gröbner Bases, MPI (Message Passing Interface), Gauss Elimination, ZKP (Zero-Knowledge Proof)

参考符号

a|b	a 整除 b
$\sqrt{I}$	理想 I 的根理想
<S>	由集合 S 中元素生成的子群或者理想
K	域
I：J	理想 I 被理想 J 除的商理想
lt(f)	多项式 f 的首项
lc(f)	多项式 f 的首项的系数
lp(f)	多项式 f 的首项的幂积
$k[x_1,x_2,\cdots,x_n]$	域 k 上 n 变元 $x_1,x_2,\cdots,x_n$ 的多项式环
$<f_1,f_2,\cdots,f_r>$	由多项式 $f_1,f_2,\cdots,f_r$ 生成的理想或子模
gcd	最大公因子
lcm	最小公倍式
$f \xrightarrow{G} h$	模 G 约化到 h
$S(f,g)$	多项式 f 和 g 的 S-多项式
$\deg(f)$	多项式 f 的次数
X	向量 $(x_1,x_2,\cdots,x_n)$
GB（I）	理想 I 的 Gröbner 基
RGB（I）	I 的既约 Gröbner 基
$\ker\varphi$	表示映射 φ 核
∂(f)	表示多项式 f 的次数

创新性声明

秉承学校严谨的学风和优良的科学道德，本人声明所呈交的论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢中所罗列的内容以外，论文中不包含其他人已经发表或撰写过的研究成果；也不包含为获得西安电子科技大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中做了明确的说明并表示了谢意。

申请学位论文与资料若有不实之处，本人承担一切相关责任。

本人签名：张鹏　　日期：2008.7.3

关于论文使用授权的说明

本人完全了解西安电子科技大学有关保留和使用学位论文的规定，即：研究生在校攻读学位期间论文工作的知识产权单位属西安电子科技大学。学校有权保留送交论文的复印件，允许查阅和借阅论文；学校可以公布论文的全部或部分内容，可以允许采用影印、缩印或其他复制手段保存论文。（保密的论文在解密后遵守此规定）

本人签名：张鹏　　日期：2008.7.3

导师签名：　　日期：2008.7.5

第一章 绪 论

随着通讯网络特别是 INTERNET 的高速发展，利用网络作为信息交流和信息处理变得越来越普遍。目前，无论是国家政府还是企业都正融入到网络通信的大潮之中，从其原来的传统经营模式向网络化模式演进。电子政务、电子商务及其它的各类电子业务将成为不可逆转的发展趋势。伴随着与日俱增的海量相关网络活动的发生，人们越来越多地将其注意的焦点投向信息安全这个问题。信息安全问题主要集中体现在以下方面：

(1)身份认证——确认网络客户的真实身份；

(2)信息和数据的保密性——个人或系统机密信息和数据保护；

(3)信息和数据的完整性——防止不合法的数据修改；

(4)不可抵赖性——网络环境下行为的事后的不可抵赖(数字签名)；

信息安全中最核心的技术是密码技术。密码技术是研究对传送信息采取何种的变换以防止第三者对信息的窃取，因而密码技术是实现所有安全服务的重要基础。密码体制从原理上可分为两大类，即单钥体制（One-key System）和双钥体制（Two-key System）。

密码技术经过几十年的发展已经趋于成熟，从应用方面来看大体分为两类：对称密码技术和非对称密码技术。非对称密码技术是支撑解决上述所涉及的四个关键方面的问题的核心。目前越来越流行的是基于 PKI 体系模型的解决方案。公钥密码的安全性都是基于复杂的数学难题。根据所基于的数学难题来分类，有以下三类系统目前被认为是安全和有效的：(1)大整数因子分解系统(代表性的有 RSA)；(2)有限域离散对数系统(代表性的有 DSA)；(3)有限域椭园曲线离散对数系统(ECC)。在有限域上，结合 Gröbner 基的一些良好的性质，可以在信息安全领域中的很多方面进行有效工作，如：密钥分发、加密、解密、零知识证明、数字签名、不可否认消息认证等。而且随着相关算法的进步与完善、计算机速度的提高和计算机网络的发展 Gröbner 基的实用价值进一步得到体现，其应用范围也进一步扩展，由于其良好的性质，Gröbner 基已经成为了一种进行密码分析的有力工具。其中目前 Gröbner 基最主要的应用之一是其可以辅助解决密码学中的多变元多项式方程系统的问题。

下面我们就来回顾一下 Gröbner 基相关理论的发展简史。Gröbner 基理论的形成经历了几十年的时间，最早可追溯到 1927 年 F.S.Macauly 的工作，其主要工作是将全序的概念引入到由多变元多项式环中单项式全体组成的集合内，其目的是

研究理想的某些不变量。随后 H.Hironaka 于 1964 年在研究奇性分解（resolution of singularities）时，引入了多变元多项式的除法算法。1965 年奥地利数学家 Bruno Buchberger 使用除法算法系统地研究了域上多变元多项式环的理想生成元问题。其基本思想是在单项式的集合中引入保持单项式的乘法运算的全序，称为项序，以保证多项式相处后所得余多项式的唯一性。他引入 S-多项式，使得对多项式环中的任一给定的理想，从它的一组生成元出发，可计算得到一组特殊的生成元，即 Gröbner 基（这是 Bruno Buchberger 用其导师的名字命名的）。B.Buchberger 最大的贡献在于使 Gröbner 基可以计算，可以真正求出来。自从 B.Buchberger 发现 Gröbner 基（Gröbner Bases 或 Groebner Bases）至今 Gröbner 基已经在计算代数领域内，不论是在理论研究还是实际应用中，都已经成为了一种不可或缺的强有力的计算工具，而且被视为一种可以进行密码分析的有力工具。此后，H.Grauert，G.Beman，D.Lazard，L.Robbiano 等人又分别先后对 Gröbner 基理论进行了进一步深入的研究。近十几年来，Gröbner 基的应用研究发展迅速，这包括给出切实可行的域上的多项式环中的理想的准素分解算法，其中零维理想的准素分解的研究比较彻底。

我们所研究的 Gröbner 基是一类基于有限域 $\mathbb{F}$ 的多变元多项式环 $\mathbb{F}[X]$上的非零理想的特殊基。Gröbner 基方法是求解非线性代数系统的一种非数值迭代的代数方法。其基本思想是在原非线性多项式代数系统所构成的多项式环中，通过对变量和多项式项的适当排序，对原系统进行约简，最后生成一个与原系统等价且便于直接求解的标准基（Standard Bases，即 Gröbner 基）。B.Buchberger 在其论文中有如下的关于算法描述：假定有属于有限域 $\mathbb{F}$ 上的多变元多项式环 $\mathbb{F}[X]$中的一个多项式有限序列 F，即 $F\subseteq\mathbb{F}[X]$，计算一组属于有限域 $\mathbb{F}$ 上的多变元多项式环 $\mathbb{F}[X]$中的 Gröbner 基 G，即 $G\subseteq\mathbb{F}[X]$，则可由多项式有限序列 F 和 Gröbner 基 G 生成相同的理想 I。

Gröbner 基方法（理论和算法）能够提供一种标准的方法，这种方法能很好的解决可以被表示成由多变元多项式集合中项的形式所构成的很多问题，这些问题包括：代数几何、交换代数和多项式理想理论，非变量理论，自动化几何定理证明，编码理论，整数程序设计，偏微分方程，超几何函数，统计学，非交换代数系统理论等。

因为 Gröbner 基拥有比其他任意多项式序列更好的特性，所以很多涉及理想 I 的问题，只要有 Gröbner 基 G，即可获得 $G\subseteq\mathbb{F}[X]$，则可使得计算多项式有限序列 F 的过程变得简单。例如理想 I 的构成问题：对于任意多项式 $f\in\mathbb{F}[X]$，判断是否存在关系 $f\in I$。当然 Gröbner 基最主要的应用之一是其可以辅助解决密码学领域中的多变元多项式方程系统的问题。事实上计算相关系统的 Gröbner 基的工作量也是相当大的。在最坏的情况下，Gröbner 基的运算量呈双幂指数复杂度增长。但实际

中许多应用 Gröbner 基例子都可以很快的获得结果，而且在很多的计算机系统的代数计算子系统中有一条类似于 GröbnerBases 或 GroebnerBases 的指令以便用户可以较方便的进行 Gröbner 基的相关计算。目前已经有了一些可以较为方便的直接计算 Gröbner 基的数学软件，如 Mathematica，Maple，Singular 等（参见附录），这些软件包中含有相关程序，虽然其算法不尽相同，效率也大都不太高，但随着 Gröbner 基求解效率的不断提升，其良好的应用前景仍然是可以预见。

因为基本的求解 Gröbner 基的 Buchberger 算法的计算效率是很低的，因而出现了许多改进 Buchberger 算法的新方法。现在主流的生成算法主要包括是 F4 算法和 F5 算法。F4 算法是目前一种有效的而且应用较为广泛的计算 Gröbner 基的方法。作为对 F4 算法的一种改进，F5 算法避免了 Gröbner 基的计算中的一些冗余计算。该算法还特别加入了一些附加标准用以检测无效运算，以便减少这些运算，从而可以在一定程度上加速整个算法的实现，但其整体执行效率并不是很高。

本文从介绍 Gröbner 基入手，首先介绍了相关的代数学知识，建立项序的概念，随后介绍了求解 Gröbner 基的基本理论和基本算法及其作用域，包括多项式除法及其算法、最基本的 Buchberger 算法、扩展的环上及主理想环上的相关算法等；继而从现有的求解 Gröbner 基的主流算法出发，先对求解算法进行分析，介绍了算法的并行相关问题及影响算法效率的关键问题——中间项的约化，这也是我们所主要关注的地方，我们在此过程中采用 C+MPI 技术来进行并行处理，首先采用结构化高斯消元法对可能产生大型稀疏矩阵予以简化，而后采用并行高斯全选主元消去法对中间项进行约化，以达到提高算法实现效率的目的，最后选取一个简单的实例进行验证；最后我们将 Gröbner 基方法应用于零知识证明方式的身份认证，提出以采用并行的基于 Gröbner 基的零知识证明与部分盲签名相结合的方式进行安全电子支付的模型，并对其交易过程和安全性进行了分析。当然我们的工作还主要停留在较浅的层面上，在今后的工作中我们将进一步对其进行深入与完善并争取将其实现。

由于本人的知识水平有限，文中难免有缺点和错误之处，敬请批评指正。

第二章 Gröbner 基理论

Gröbner基是由奥地利数学家Bruno Buchberger首先在其博士论文中介绍的。Gröbner基方法是求解非线性代数系统的一种非数值迭代的代数方法。其基本思想是在原非线性多项式代数系统所构成的多项式环内，通过对变量和多项式的项的适当排序，对原系统进行约简，最后生成一个与原系统等价且便于直接求解的标准基（Standard Bases，即Gröbner基）。传统的结式消元法也可以是求解符号形式非线性代数系统的代数法，但它需要高度依赖于具体问题的消元技巧，并且会产生增根，这是我们在本文中主要改进的方面之一。

本章中我们讲述Gröbner基的基本理论和求Gröbner基的基本算法。首先给出建立Gröbner基理论所必须的概念----项序，然后分别讲述域上和环上的Gröbner基及其相关的基本求解算法。

2.1 算术代数知识

本节中我们将讲述 Gröbner 基理论和应用所需的最基本的算术代数知识，但是我们并未对其作系统地介绍，而仅仅是讲述最基本的、本文所涉及的、非讲不可的内容，借此可以很快的了解 Gröbner 基理论本身，所以对所有文中提及的定义、定理未作证明。本节中的定义及定理主要摘自参考文献[9]和[10]。

因为环具有两个代数系统，其中第一种运算是交换群的运算，第二种运算是半群的运算。所以首先我们介绍幺半群（monoid），进而引出 Dickson 定理。Dickson 定理是建立 Gröbner 基理论的基石之一。

定义 2.1.1（半群、幺半群）一个半群（semigroup）是指一个非空集合 M 和 M 上的满足结合律的二元运算“•”，记为（M，•），即 $M\neq\varnothing$，对任何 a，b∈M，有 a•b∈M，亦可简记 a•b=ab，并对任何 a，b，c∈M，有 a(bc)=(ab)c。M 中的一个元素 e 称为幺元素或单位元（unit），如果对于 a∈M，那么都有 ea=ae=a。如果半群 M 含有元素，则称 M 为幺半群（monoid）。如果对于任何 a • b∈M，都有 ab=ba，那么称 M 为交换半群。

定理 2.1.1（Dickson 定理）交换幺半群 $\mathbb{N}^n$ 中的任意一个理想都是有限生成的。

在后面的研究中我们将看到，在多项式环中单项式生成理想时，Dickson 引理将起到关键作用。

定义 2.1.2（主理想环）如果环 R 中的每个理想都是主理想，则称 R 为主理想环（principal ideal ring）。

定义 2.1.3（诺特环）如果 R 中的每个理想都是有限生成的，则称 R 为诺特环（Noetherian ring），有时亦记为 Noether 环。

定义 2.1.4（准素理想）设 I 是环 R 中的理想。则 I 称为准素理想(primary ideal)，是指 I≠R，和对任何 x，y∈R，如果 xy∈R，那么或者 x∈I，或者有某个正整数 n，使得 $y^n \in I$，即 I 是准素理想当且仅当 R / I≠0 和 R / I 中每个零因子都是幂零元。

一般说来，理想的极小准素分解并不是唯一的。但是，在理想的极小准素分解中，属于它的素理想和素理想的个数是唯一确定的。

定理 2.1.2 诺特环 R 中的每个理想都有极小准素分解。

在本章中我们总设R表示含有单位元的诺特(Noether)交换环，k表示一般的域，设A=k[$x_1,x_2,\cdots,x_n$]表示域k上的n变元多项式环，或者A=R[$x_1,x_2,\cdots,x_n$]=R[X]表示n变元多项式环。

2.2 项 序

Gröbner 基理论的本质是从多变元多项式环中任意一个理想的一组生成元出发，描述和计算出一组具有“良好”性质的生成元，而具有这种良好性质的生成元，可以帮助我们研究理想的结构和进行某些理想运算。由理想的一组生成元出发求出另一组生成元，很自然的想到，其过程必定含有多变元多项式的除法运算，或确切的说，含有一个多项式被一个非零多项式去除，或被多个非零多项式去除的运算。对于域上的单变元的情形，利用长除法便可做到。但对环上的多变元情形，问题变得复杂得多。即便是域上的情形，如果变元个数大于或等于 2，按通常的办法作除法，其商多项式和余多项式都不保证唯一。而如果每次运算的结果不同，这种运算就没有多大的意义了。对于环上的情形，即便是单变元，也相当困难。

我们先研究域上的多变元的情形。对域上的两个单变元多项式可以进行除法运算的关键是在于每次运算都可用消最高次项进行，而在用消元法解域上的线性方程组时，需要先确定变元的消去顺序，而后再根据高斯消去进行运算。由此在要做多变元多项式除法时，应在单项式集合或变元的幂积的集合中引进序，当然是全序。进而，由于单项式间可进行乘法运算，因此希望引进的序能够与乘法运算相容，或者说保持乘法运算。通常具有这种性质的序被称为项序。本节的主要内容就是给出项序的确切定义和基本性质。

令 N 是非负整数集合，n 是一个给定的正整数，$x_1,x_2,\cdots,x_n$ 表示环 R 上的 n 个变元。令集合

$$T^n=\{x_1^{\alpha_1},x_2^{\alpha_2},\cdots,x_n^{\alpha_n}\,|\,\alpha_i\in\mathbb{N},i=1,2,\cdots,n\},$$

即 T^n 是 n 个变元 $x_1,x_2,\cdots,x_n$ 的幂积的集合。简记 $x_1^{\alpha_1}x_2^{\alpha_2}\cdots x_n^{\alpha_n}=X^{\alpha}$，其中

$X=(x_1,x_2,\cdots,x_n)$，$\alpha=(\alpha_1,\alpha_2,\cdots,\alpha_n)\in\mathbb{N}^n$。对于$T^n$中的任一两个元素$X^\alpha=x_1^{\alpha_1}x_2^{\alpha_2}\cdots x_n^{\alpha_n}$和$X^\beta=x_1^{\beta_1}x_2^{\beta_2}\cdots x_n^{\beta_n}$，定义它们的乘积为：$X^\alpha\cdot X^\beta=X^\alpha X^\beta=x_1^{\alpha_1+\beta}x_2^{\alpha_2+\beta_2}\cdots x_n^{\alpha_n+\beta_n}$，或者由环$R[x_1,x_2,\cdots,x_n]$中的乘法得到上式，结果$T^n$都是一个幺半群。

所谓σ是集合T^n上的一个全序(total order)，是指对任意给定的T^n中的两个元素X^α和X^β，下面的三个关系之一必须成立，而且只有一个成立：

$X^\alpha<_\sigma X^\beta$，$X^\alpha=X^\beta$，$X^\beta<_\sigma X^\alpha$。

如果无须特别标注σ，上式可简记为：

$X^\alpha<X^\beta$，$X^\alpha=X^\beta$，$X^\beta<X^\alpha$。

集合T^n上的全序σ称为良序(well-ordering)，如果T^n的每个非空子集合都有最小元，即对T^n的任何非空子集A，必存在元素$X^\alpha\in A$，使得对所有$X^\beta\in A$，$X^\alpha\leqslant_\sigma X^\beta$。

以下给出项序的定义和基本性质：

定义 2.2.1 集合T^n上的一个全序“<”（全序σ的$<_\sigma$简记）称为一个项序(term order)，如果其同时满足如下的两个条件：

（1）对所有$X^\beta\in T^n$和$X^\beta\neq 1$，都有$1<X^\beta$；

（2）对任何X^α，X^β，$X^\gamma\in T^n$，如果$X^\alpha<X^\beta$，则$X^\alpha X^\gamma<X^\beta X^\gamma$。

定理 2.2.1 设σ是集合T^n上的项序，则σ是T^n上的良序。

推论 2.2.1 设σ是集合T^n上的项序当且仅当它是T^n上的良序，而且保持乘法运算。

T^n上的相对于$x_1<x_2<\cdots<x_n$的字典项序(lexicographical order)是T^n上的如下一个项序：$1<x_1<x_1^2<x_1^3<\cdots<x_2<x_1x_2<\cdots<x_n<x_1x_n<\cdots<x_1x_1^2\cdots x_n<\cdots$

T^n上的相对于$x_1<x_2<\cdots<x_n$的全次数字典项序(total degree lexicographical order)定义如下：

对$i=\{i_1,i_2,\cdots,i_n\}$，$j=\{j_1,j_2,\cdots,j_n\}\in\mathbb{N}^n$，则

$$X^i<X^j\Leftrightarrow\sum_{s=1}^{n}i_s<\sum_{s=1}^{n}j_s\text{，或者}\sum_{s=1}^{n}i_s=\sum_{s=1}^{n}j_s\text{，}$$

且相对于$x_1<x_2<\cdots<x_n$的字典项序下满足$X^i<X^j$。于是

$$1<x_1<x_2<\cdots<x_n<x_1^2<x_1x_2<\cdots<x_{n-1}x_n<x_n^2<x_1^3<x_1^2x_2<\cdots<x_n^3<\cdots$$

有时，也把T^n上的项序称为R[x]上的项序。固定的R[x]上的一个项序，则对任意$f\in R[x]$，其中$f\neq 0$，可表示为

$$f=\alpha_1X^{\alpha_1}+\alpha_2X^{\alpha_2}+\cdots+\alpha_rX^{\alpha_r}$$

其中$0\neq\alpha_i\in R$，$X^{\alpha_i}=x_1^{\alpha_{i1}}x_2^{\alpha_{i2}}\cdots x_n^{\alpha_{ir}}$，$X^{\alpha_i}\in T^n$，$1\leqslant i\leqslant r$，且$X^{\alpha_1}>X^{\alpha_2}>\cdots>X^{\alpha_r}$。

下面给出一些常用的记号，我们定义：

$lp(f)=X^{\alpha_1}$，即 f 的首项幂积；

$lc(f)=\alpha_1$，即 f 的首项系数；

$lt(f)=\alpha_1 X^{\alpha_1}$，即 f 的首项；

$lp(0)=lc(0)=lt(0)=0$；

$c(f,\ X^{\alpha})$为幂积 X^{α} 在 f 中的系数；

$Supp(f)=\{X^{\alpha}\in T^{n}\mid c(f,\ X^{\alpha})\neq 0\}$，为在 f 中出现的幂积的集合；

$\log(X^{\alpha})=\alpha$，为幂积 X^{α} 的幂指数。

2.3 除法算法

本节中我们讨论求域上的多项式环中的理想的 Gröbner 基的 Buchberger 算法所依赖的除法算法。

本节中总设 $A=k[x_1,x_2,\cdots,x_n]$表示域 k 上的 n 变元多项式环，σ 是任意给定的环 A 上的一个项序，并简记为<。

定义 2.3.1 对于环 A 中的三个多项式 f，g，h，其中 $g\neq 0$，我们称 f 模 g 一步约化为 h，用 $f\xrightarrow{g}h$ 表示，当且仅当 lp(g)是 f 中某一非零单项式 X 的因子，且

$$h=f-\frac{X}{lt(g)}g。$$

这个约化过程，就是将 f 中的一个项用严格比它小的一些项的和来代替。

定义 2.3.2 令 $f,h,f_1,\cdots,f_s$ 是环 A 中的多项式，且对 i=1,2,…,s, $f_i\neq 0$。令集合 $F=\{f_1,\cdots,f_s\}$。我们说 f 模 F 约化为 h，用 $f\xrightarrow{F}_{+}h$ 表示，当且仅当下式成立：

$$f\xrightarrow{f_{i_1}}h_1\xrightarrow{f_{i_2}}h_2\xrightarrow{f_{i_3}}\cdots\xrightarrow{f_{i_t}}h_t=h$$

其中对 $j=1,2,\cdots,t$，$f_{i_j}\in F,h_j\in A$。

定义 2.3.3 如果 R[x]中的非零的多项式 f 不能被 R[x]的非零多项式子集 $F=\{f_1,\cdots,f_s\}$约化，则称 f 是相对于 F 的既约标准型，或称 f 相对于 F 是既约的(reduced)。

定理 2.3.1（参考文献[1]，P206）设 R 是 Noether 交换环，设 f 是 R[X]的非零多项式，F 是 R[X]的非零多项式集合。则 f 是相对于 F 的既约标准型，当且仅当 $lt(f)\notin lt(F)$。

下面给出域 k 上的 n 变元多项式之间的除法算法，其既可用于上机使用，也

可用于手工演算。

算法 2.1 多变元多项式除法算法

输入：$f_0, f_1, \cdots, f_s \in \mathbb{F}[x_1, x_2, \cdots, x_n]$，其中，$f_i \neq 0 (0 \leqslant i \leqslant s)$。

输出：$u_1, u_2, \cdots, u_s$，r，使得 $f = u_1 f_1 + u_2 f_2 + \cdots + u_s f_s + r = \sum_{i=1}^{s} u_i f_i + r$，且 r 相对于$\{f_1, f_2, \cdots, f_s\}$是既约的，$lp(f)=\max\{lp(u_1)lp(f_1), \cdots, lp(u_s)lp(f_s), lp(r)\}$。

初始化：$u_1 := 0, u_2 := 0, \cdots, u_s := 0, r := 0, h := f$。

WHILE $h \neq 0$ DO

IF 存在 i 使得 $lp(f_i) | lp(h)$ THEN

$$u_i := u_i + \frac{lt(h)}{lt(f_i)}$$

$$h := h - \frac{lt(h)}{lt(f_i)} f_i$$

ELSE

$$r := r + lt(h)$$

$$h := h - lt(h)$$

注意：（1）该算法中得到的多项式 $u_1, u_2, \cdots, u_s$ 和余多项式 r 都依赖于 $f_1, f_2, \cdots, f_s$ 的排列顺序；

（2）如果 $f \in \langle f_1, f_2, \cdots, f_s \rangle = \langle F \rangle$，则该算法产生的 r 不一定为 0。

定理 2.3.2（参考文献[10]）给定环 A 中非零多项式集合 $F=\{f_1, f_2, \cdots, f_s\} \subseteq A \backslash \{0\}$ 和多项式 $f \in A$，及环 A 上的项序 $<$，则算法 2.1 产生的输出多项式 $u_1, u_2, \cdots, u_s$ 和 r，满足下面的性质：

（1）$f = \sum_{i=1}^{s} u_i f_i + r$，其中 u_i，$r \in A$，且 r 相对 F 是既约的；

（2）$lp(f)=\max\{lp(u_1)lp(f_1), \cdots, lp(u_s)lp(f_s), lp(r)\}$。

2.4 域上的 Gröbner 基

本节中我们将给出域 k 上的 n 变元多项式环 $A= k[x_1, x_2, \cdots, x_n]$ 中理想的 Gröbner 基的定义。

计算域上的多项式环中理想的 Gröbner 基的算法要注意以下两点：（1）算法的基础是一个多项式模一个非零多项式的约化算法，这需要涉及到域中元素的除法；

（2）算法是可由计算机程序实现的。

定理 2.4.1 （参考文献[1]）设 A 是 Noether 交换环，I 是 R[x]的理想，$G=\{g_1,g_2,\cdots,g_t\}$是 I 的一组生成元。则下述论断等价：

（1）lt(G)=lt(I)；

（2）若 f 是 R[x]中的任意多项式，则 $f\in I\Leftrightarrow f\xrightarrow{G}_+0$；

（3）对所有 $f\in I$，则 f 可表示成

$$f=h_1g_1+h_2g_2+\cdots+h_tg_t \qquad (3.3\text{-}1)$$

其中，$h_1,h_2,\cdots,h_t\in R[x]$，满足 $lp(f)=\max_{1\le i\le t}\{lp(h_i)lp(g_i)\}$。这时称式（3.3-1）是 f 的 G-表示。

定义 2.4.1 （Gröbner 基） 设 I 是环 A 中任意给定的一个非零理想，$G=\{g_1,g_2,\cdots,g_t\}$是 I 中非零多项式的有限集合。则称 G 是理想 I 的 Gröbner 基(Gröbner Bases，有时亦称为标准基（standard bases）)，当且仅当对 I 中的每个非零多项式 f，存在 i，$1\leqslant i\leqslant t$，使得 $lp(g_i)\mid lp(f)$。

定义 2.4.2（既约 Gröbner 基） 设 R 是诺特环，I 是 R[x]的理想，$G=\{g_1,g_2,\cdots,g_r\}$是 I 的 Gröbner 基，如果对每个 $j=1,2,\cdots,r, lt(g_j)\notin lt(G\backslash\{g_j\})$，则称 G 是既约 Gröbner 基(reduced Gröbner Bases)，简记为 RGB。

定理 2.4.2 设 I 为环 A 上的理想，$G=\{g_1,g_2,\cdots,g_r\}$是 I 中非零元集合。如果 G 是 I 的 Gröbner 基，那么对任何 $f\in I$，如果 $f\xrightarrow{G}_+r$ 和 r 相对 G 是既约的，则必有 r=0。

定理 2.4.3 令 I 为环 $A=k[x_1,\cdots,x_n]$上的非零理想，$G=\{g_1,g_2,\cdots,g_r\}\subseteq I\backslash\{0\}$。则下面的叙述使等价的：

（1）G 是 I 的 Gröbner 基；

（2）$f\in I$ 当且仅当 $f\xrightarrow{G}_+0$；

（3）$f\in I$ 当且仅当存在 $h_1,h_2,\cdots,h_t\in A$，使得 $f=\sum_{i=1}^{t}h_ig_i$，

$lp(f)=\max\{lp(h_i)lp(g_i)\mid i=1,2,\cdots,t\}$；

（4）lt(G)=lt(I)。

定义 2.4.3 设环 A 中的非零多项式的有限集 G 是 Gröbner 基，是指 G 生成的理想$\langle G\rangle$的 Gröbner 基。

定理 2.4.4 设 G 为环 A 上的 Gröbner 基，则对任何 $f\in A$，$f\xrightarrow{G}_+r$ 其中 r 相对 G 是既约的，则 r 由 f 和 G 惟一确定。

定理 2.4.5 域 k 上 n 变元多项式环 A，集合 $G=\{g_1,g_2,\cdots,g_r\}\subseteq I\backslash\{0\}$。则 G 是 Gröbner 基当且仅当对任何多项式 $f\in A$，f 模 G 的余项是唯一的。

任何理想都存在 Gröbner 基，但是 Gröbner 基理论的关键是提出可行算法，即由理想的任何一组生成元出发，计算出该理想的 Gröbner 基的算法，而且由目前的计算机的计算能力，该算法是可实现的。本节我们介绍由 B. Buchberger 提出的 Buchberger 算法，此算法的核心是引入了 S–多项式的概念。

定义 2.4.4 设环 A 是域 k 上 n 变元多项式环，设 $f,g\in A\backslash\{0\}$，L=lcm(lp(f)，lp(g))，其中 lcm 表示最小公倍。令

$$S(f,g)=\frac{L}{lt(f)}f-\frac{L}{lt(g)}g。$$

多项式 $S(f,g)$ 称为 f 和 g 的 S-多项式（S-Polynomials）（参考文献[1]、[10]）。

定理 2.4.6 设 $f_1,f_2,\cdots,f_s\in\mathbb{F}[x_1,\cdots,x_n]$，并且 lp($f_1$)=lp($f_2$)=…=lp($f_s$)=X≠0。设 $f=\sum_{i=1}^{s}c_if_i$，其中 $c_i\in$F，（i=1,2,…,s），并使得 lp(f)<X。则 f 可以表示成 $S(f_i,f_{i+1})$（$1\le i\le s-1$）在 F 上的线性组合。

证明：记 $f_i=a_iX+g_i$，其中 g_i 是 $\mathbb{F}[x_1,\cdots,x_n]$中的多项式，且 g_i 的每个单项<X。由于 $f=\sum_{i=1}^{s}c_if_i$，且 lp(f)<X，所以 $\sum_{i=1}^{s}c_ia_i=0$。由此得到

$$\begin{aligned}f&=\sum_{i=1}^{s}c_if_i\\&=\sum_{i=1}^{s}c_ia_i\frac{f_i}{a_i}\\&=\sum_{j=1}^{s-1}(\sum_{i=1}^{j}c_ia_i)(\frac{Xf_j}{a_jX}-\frac{Xf_{j+1}}{a_{j+1}X})+(c_1a_1+c_2a_2+\cdots+c_sa_s)\frac{f_s}{a_s}\\&=\sum_{j=1}^{s-1}(\sum_{i=1}^{j}c_ia_i)S(f_j,f_{j+1})\end{aligned}$$

因此，f 可表示成 F 上的线性组合。 □

定理 2.4.7（Buchberger）（参考文献[1]）令 G=$\{g_1,g_2,\cdots,g_t\}\subseteq A\backslash\{0\}$，则 G 是理想 I=$\langle G\rangle$ 的 Gröbner 基，当且仅当对所有 $i\neq j,1\le i,j\le t$，有

$$S(g_i,g_j)\xrightarrow{G}_{+}0。$$

下面介绍最基本的求解 Gröbner 基的算法—Buchberger 算法，它是基于定理 2.4.7 的求解 Gröbner 基的算法。这种算法由 Buchberger 于 1965 年提出。此算法在可交换代数集合上采用一种计算的方式处理问题，类似于判定一个多项式是否属于由某些有顺序的多项式生成的理想一样。

算法 2.2 计算理想的 Gröbner 基 Buchberger 算法

输入：H=$\{h_1,h_2,\cdots,h_s\}\subseteq\mathbb{F}[x_1,\cdots,x_n]$，其中，$h_i\neq0$（$1\le i\le s$）。

输出：G=$\{g_1,g_2,\cdots,g_t\}$，一个 Gröbner 基，使得$\langle G\rangle=\langle H\rangle$。

初始化：G:=H，$\Psi:=\{(h_i,h_j)\mid h_i\neq h_j,h_i,h_j\in G\}$

WHILE $\Psi\neq\varnothing$ DO

任意选择 $(f,g)\in\Psi$

$\Psi:=\Psi\setminus\{(f,g)\}$

$S(f,g)\xrightarrow{G}_+h$，这里 h 已不能被 G 简化

IF $h_i\neq 0$ THEN

$\Psi:=\Psi\cup\{(u,h)\mid$对所有u $\in$ G$\}$

$G:=G\cup\{h\}$

证明：在此简要证明 Buchberger 算法的正确性。

如果算法是有限步停止的，则输出的有限集合 G 必然能够满足如下性质：（1）$\langle G\rangle=\langle H\rangle$；（2）对任意 $f,g\in$ G，则$S(g_i,g_j)\xrightarrow{G}_+0$。因此，由定理 2.4.7 可知，G 是 Gröbner 基。所以需要证明 Buchberger 算法是有限步停止的。设算法中第 i 步的集合 G 是 G_i（i=0,1,$\cdots$,$G_0=H$）。其中 G_{i+1} 是 G_i 中添加了一个不能被 G_i 约化的元素。因此有

$$lt(G_0)\subset lt(G_1)\subset\cdots\subset lt(G_i)\subset lt(G_{i+1})\subset\cdots$$

由 Hilbert 基定理（参考文献[1]），存在 t≥0 使得 $lt(G_t)=lt(G_{t+1})=\cdots$。因此，Buchberger 算法到第 t 步就停止了。 □

Buchberger 算法为以后许多 Gröbner 基算法奠定了思想基础。

上述算法适用范围非常广，可能由于这个原因，该算法的效率不高。对不同的具体问题，需要给出更具体的快速求解 Gröbner 基的算法。

实验显示本算法中计算量最大的部分是由多项式 s 化简为标准形式 h 的部分。在最初的 Buchberger 算法中大多数情况下 h 总是可以被化简为 0。这个过程将花费大量的时间，特别是如果在给定一些可以很容易就能看出是否通过模$\{g_1,\ldots,g_n\}$可以化简为 0 的多项式 s 时，此过程就将不再进行。

然而 Gröbner 基算法在其实现中有一个限制因素，就是在实际应用中通常会占用大量的内存空间，因此以后出现的新算法都将考虑的一个重要问题就是如何提高内存效率。

同时 Buchberger 算法所计算出的 Gröbner 基并不唯一，我们希望能找到与 Gröbner 基相关的不变量，为此我们需要使用既约 Gröbner 基。

定义 2.4.5（极小 Gröbner 基）环 A 中的 Gröbner 基 G=$\{g_1,g_2,\cdots,g_t\}\subseteq$A\{0} 称为极小的（minimal），如果对每个 i，$1\leq i\leq t$，$lc(g_i)=1$，而且对任何 i≠j,$1\leq i,j\leq t$，都有 $lp(g_i)\nmid lp(g_j)$。

定理 2.4.8 设 G=$\{g_1,g_2,\cdots,g_t\}$是环 A 中的理想 I 的 Gröbner 基。如果 $lp(g_i)\mid lp(g_j)$，$1\le i,j\le t$，则$\{g_1,\cdots,g_{j-1},g_{j+1},\cdots,g_t\}$仍是 I 的 Gröbner 基。

利用定理 2.4.8 从任何一组 Gröbner 基出发，不难求得极小 Gröbner 基。极小 Gröbner 基也不是唯一，但它蕴含着某些不变量。极小性是相对 Gröbner 基中元素个数而言。

环 **A** 中的 Gröbner 基 G=$\{g_1,g_2,\cdots,g_t\}$为既约 Gröbner 基，如果对所有 i，$1\le i\le t$，$lc(g_i)=1$，而且 g_i 相对 $G_i\setminus\{g_i\}$是既约的，即对任何 i，g_i 中没有非零项可被任何 $lp(g_j)$ 除，$i\ne j$。

既约 Gröbner 基当然是极小的，但反之不成立。然而可由一组极小 Gröbner 基出发求出既约 Gröbner 基（RGB）。

2.5 环上的 Gröbner 基

本节主要是将域上的 Gröbner 基理论推广到环上去。域和环的本质差别是在于能否作除法。因此要将域上 Gröbner 基算法的基础，即约化过程推广到环上，就需要找到既能消项，又能避免除法运算的算法，这就是本节的出发点。

定义 2.5.1（合冲-可解环）（参考文献[10]）令 R 是含有单位元的交换环，如果对 R 中的任何有限集$\{r_1,r_2,\cdots,r_t\}$可计算出 R-模

$$R(r_1,r_2,\cdots,r_t)=\{(s_1,s_2,\cdots,s_t)\in R^t\,\big|\,\textstyle\sum_{i=1}^{t}s_ir_i=0\}$$

的有限基（即有限生成元组）$\overrightarrow{\omega_1},\overrightarrow{\omega_2},\cdots,\overrightarrow{\omega_q}\in R^t$，则称 $\overrightarrow{\omega_1},\overrightarrow{\omega_2},\cdots,\overrightarrow{\omega_q}$ 为模 $R(r_1,r_2,\cdots,r_t)$ 的有限合冲基，R-模 $R(r_1,r_2,\cdots,r_t)$ 中的元素称为合冲(Syzygy)，$R(r_1,r_2,\cdots,r_t)$ 为合冲模(Syzygy module)，常用 $Syz_R(r_1,r_2,\cdots,r_t)$，R 称为合冲-可解环(Syzygy-solvable ring)。

若 $\overrightarrow{\omega_i}=(\omega_{i1},\omega_{i2},\cdots,\omega_{it})$，$1\le i\le q$ 是 $Syz_R(r_1,r_2,\cdots,r_t)$ 的有限基，则 $\overrightarrow{\omega_i}$ 满足：

（1）对任何 i，$1\le i\le q$，$\sum_{j=1}^{t}\omega_{ij}r_j=0$，

（2）对任何 $\vec{u}=(u_1,u_2,\cdots,u_t)\in R^t$，$\sum_{i=1}^{t}u_ir_i=0$，存在 $\vec{v}=(v_1,v_2,\cdots,v_q)\in R^q$，使得 $\vec{u}=\sum_{i=1}^{q}v_i\overrightarrow{\omega_i}$。

实际上，合冲基可以看作以 $r_1,r_2,\cdots,r_t$ 为系数的环 R 上齐次线性方程的解析所构成的 R-模的一组有限生成元。

定义 2.5.2（强可计算环）令 R 是含有单位元的交换环，如果 R 满足下面的 4 个条件：

（1）R 是诺特环；

（2）R 是可计算的；

（3）R 是可分离的；

（4）R 是合冲-可解的；

则称 R 为强可计算环(strong computable ring)。

定义 2.5.3 对任意给定的环 A 中的两个多项式 f 和 h，及非零多项式集合 $F=\{f_1,f_2,\cdots,f_s\}\subseteq A\setminus\{0\}$，称 f 模 F 可约化到 h，用 $f\xrightarrow{F}_+h$ 表示，如果存在有限个多项式 $h_1,h_2,\cdots,h_{s-1}\in A$ 使得

$$f\xrightarrow{F}h_1\xrightarrow{F}h_2\xrightarrow{F}h_{s-1}\xrightarrow{F}h$$

由此知，如果 $f\xrightarrow{F}_+h$，则 $f-h\in\langle F\rangle$。

设 r 是环 A 中的多项式，$F=\{f_1,f_2,\cdots,f_s\}\subseteq A\setminus\{0\}$。称 r 为相对 F 的极小(minimal)元，如果 r=0 或者 r≠0 及 r 模 F 不能约化。

引理 2.5.1（参考文献[10]）环 A 中非零多项式 r 相对集合 $F=\{f_1,f_2,\cdots,f_s\}\subseteq A\setminus\{0\}$ 是极小的，当且仅当 $lt(r)\notin lt(F)$。

定理 2.5.1（参考文献[10]）令 $f\in A$ 和集合 $F=\{f_1,f_2,\cdots,f_s\}\subseteq A\setminus\{0\}$，则存在相对 F 的极小元 r，使得 $f\xrightarrow{F}_+r$，而且存在多项式 $h_1,h_2,\cdots,h_t\in A$，使得

$$f=\sum_{i=1}^{t}h_if_i+r;$$

$$lp(f)=\max\{\max_{1\le i\le t}\{lp(h_i)lp(f_i)\},lp(r)\}$$

算法 2.3　环上除法算法

输入：$f,f_1,\cdots,f_t$，且 $f_i\neq0$（$1\le i\le t$）。

输出：$h_1,h_2,\cdots,h_t$，r，使得 $f=\sum_{i=1}^{t}h_if_i+r$，r 相对 $\{f_1,f_2,\cdots,f_t\}$ 是极小的，$lp(f)=\max\{\max_{1\le i\le t}\{lp(h_i)lp(f_i)\},lp(r)\}$。

初始化：$h_1:=0,h_2:=0,\cdots,h_t:=0,r:=f$。

IF　∃i，i:$1\le i\le t$，使得 $lp(f_i)\mid lp(r)$，

　And　$\exists c_1,c_2,\cdots,c_t\in R$，$X_1,X_2,\cdots,X_t\in T^n$ 使得 $lt(r)=\sum_{i=1}^{t}c_iX_ilt(f_i)$，

　And　$\forall c_i\neq0$，$lp(r)=X_ilp(f_i)$。

DO　$r:=r-\sum_{i=1}^{t}c_iX_if_i$

FOR　i:=1　TO　t

DO $h_i := h_i + c_i X_i$

定理 2.5.2 令 I 是环 A 中的理想，G=$\{g_1, g_2, \cdots, g_t\} \subseteq I \backslash \{0\}$，<是 T^n 上的项序。则下述条件等价：

（1）lt(G)=lt(I)；

（2）$\forall f \in A$，有 $f \in I$ 当且仅当 $f \xrightarrow{G}_{+} 0$；

（3）$\forall f \in I$，存在 $h_1, h_2, \cdots, h_t \in A$，使得 $f = h_1 g_1 + h_2 g_2 + \cdots + h_t g_t$；

$lp(f) = \max_{1 \le i \le t} \{lp(h_i) lp(g_i)\}$。

定义 2.5.4（Gröbner 基）（参考文献[10]）设 I 是环 A 中任意给定的一个非零理想，有限集合 G⊆A\{0}。则称 G 是理想 I 的一组 Gröbner 基(Gröbner Bases)，简记为 G-基（GB），如果 G 满足定理 2.5.2 中的任何一个条件，有限集和 G⊆A\{0}称为 Gröbner 基，如果 G 是理想$\langle G \rangle$的 Gröbner 基。

定理 2.5.3 设 R 是强可计算环，A=R[$x_1, \cdots, x_n$]是 R 上的 n 个变元 $x_1, \cdots, x_n$ 的多项式环，I 是 A 中任意给定的一个理想。则 I 有 Gröbner 基，即环 A 中的任一理想都存在 Gröbner 基。

定义 2.5.5（极小 Gröbner 基）（参考文献[10]）设 R 是含单位元的诺特交换环，A=R[$x_1, \cdots, x_n$]，I⊆A\{0}是环 A 中的非零理想，G 是 I 的 Gröbner 基。称 G 是 I 的极小 Gröbner 基(minimal Gröbner Basis)，如果对所有 g∈G，g 相对 G{g}是极小的。

定理 2.5.4 令 G=$\{g_1, g_2, \cdots, g_t\}$是环 A 中非零元集合，H 是合冲 A-模 Syz(lt(g_1),lt(g_2),⋯,lt(g_t))的齐次生成元集合，则 G 是理想$\langle g_1, g_2, \cdots, g_t \rangle = \langle G \rangle$的 Gröbner 基当且仅当对任何$\langle h_1, h_2, \cdots, h_t \rangle \in \langle H \rangle$有 $\sum_{i=1}^{t} h_i g_i \xrightarrow{G}_{+} 0$。

算法 2.4　环上 Gröbner 基算法

输入：$F = \{f_1, \cdots, f_s\} \subseteq A \backslash \{0\} = R[x_1, \cdots, x_n] \backslash \{0\}$。

输出：G=$\{g_1, g_2, \cdots, g_t\}$为理想$\langle f_1, f_2, \cdots, f_s \rangle$的 Gröbner 基。

初始化：$G := \varnothing$，$G' = F$。

WHILE　$G' \neq G$　DO

$G := G'$

设 G=$\{g_1, g_2, \cdots, g_t\}$

计算 A-模 Syz(lt(g_1),lt(g_2),⋯,lt(g_t))的齐次生成元集合 H

FOR $\forall \mathrm{h}=(h_1,h_2,\cdots,h_t)\in \mathrm{H}$ DO

约化 $\sum_{i=1}^{t}\mathrm{h_i g_i}\xrightarrow{G'}_{+} r$，r 相对 G' 极小

IF r≠0 THEN

$G:=G'\cup\{\mathrm{r}\}$

定理 2.5.5 设 R 是强可计算环，则算法 2.4 产生的 G 是理想的 $\langle f_1,f_2,\cdots,f_s\rangle$ 的 Gröbner 基。

在上述算法中，需要进行多次合冲模生成元的计算。Möller 给出一个上述算法中需要的求合冲模的生成元的计算方法，基本思路是充分利用已有的合冲模的生成元来计算新的合冲模的生成元，实质上是一个递归算法，避免了不必要的计算，提高了算法的效率。

设 $c_1\mathrm{X}_1,c_2\mathrm{X}_2,\cdots,c_t\mathrm{X}_t$ 为环 A=R[$x_1,\cdots,x_n$] 中的 t 个非零单项式。令 $\mathrm{Syz}_\sigma=\mathrm{Syz}(c_1\mathrm{X}_1,c_2\mathrm{X}_2,\cdots,c_\sigma\mathrm{X}_\sigma)$，其中 $1\le\sigma\le t$。为了计算 $\mathrm{Syz}(c_1\mathrm{X}_1,c_2\mathrm{X}_2,\cdots,c_t\mathrm{X}_t)=\mathrm{Syz}_t$ 的齐次生成元集合，采用归纳方法计算 $\mathrm{Syz}_1,\mathrm{Syz}_2,\cdots,\mathrm{Syz}_{t-1},\mathrm{Syz}_t$ 的齐次生成元集合。

算法 2.5　用 Möller 技巧的环 A 中 Gröbner 基算法　（参考文献[10]）

输入：$F=\{f_1,\cdots,f_t\}\subseteq \mathrm{A}\backslash\{0\}$，A=R[$x_1,\cdots,x_t$]。

输出：理想 $\langle f_1,f_2,\cdots,f_t\rangle$ 的 Gröbner 基 G。

初始化：$G:=F$，σ:=1，m:=t。

WHILE $\sigma\le m$ DO

COMPUTE $S=\{\{1,2,\ldots,\sigma\}$的相对$\mathrm{lp}(\mathrm{f}_1),\cdots,\mathrm{lp}(\mathrm{f}_\sigma)$且包含$\sigma$的饱和子集$\}$

FOR $\forall \mathrm{J}\in S$ DO

$\mathrm{X_J}:=lcm(\mathrm{lp(f_j)}\,|\,j\in J)$

COMPUTE 理想 $(\langle \mathrm{lc(f_j)}\,|\,j\in J,j\ne\sigma\rangle_R:\langle \mathrm{lc(f_\sigma)}\rangle_R)$ 的生成元

$$b_{iJ},i=1,\cdots,\mu_J$$

FOR i:=1, … , μ_J DO

COMPUTE $b_j\in R,j\in J,j\ne\sigma$，使 $\sum_{j\in J,j\ne\sigma} b_j lc(\mathrm{f_j})+b_{iJ}lc(\mathrm{f}_\sigma)=0$

约化 $\sum_{j\in J,j\ne\sigma} b_j\frac{\mathrm{X_J}}{lp(\mathrm{f_j})}\mathrm{f_j}+b_{iJ}\frac{\mathrm{X_J}}{lp(\mathrm{f}_\sigma)}\mathrm{f}_\sigma\xrightarrow{G}_{+} r$，r 相对 G 是极小的。

IF $r\ne 0$ THEN

$$f_{m+1} := r, G := G \cup \{f_{m+1}\}, m := m+1, \sigma := \sigma + 1$$

2.6 主理想环上的 Gröbner 基

环上 Gröbner 基理论是域上 Gröbner 基理论的推广，最接近域的环可以说是主理想环（除过欧几里德环以外）。因此，我们认为主理想整环，即 PID（Principal Ideal Domain）上的 Gröbner 基的计算将会与域上的 Gröbner 基的计算最相似。另外，在实际问题中常用的环，如整数环，域上单变元多项式环和欧式环都是主理想整环。因此，就促使我们进一步关注主理想整环上的 Gröbner 基理论。

本节中设 R 是主理想整环，多项式环 A=R[$x_1,\cdots,x_n$]。为研究 A 上的 Gröbner 基，下面引入 S-基和强 Gröbner 基的概念。

定义 2.6.1（参考文献[10]）设环 A=R[$x_1,\cdots,x_n$]，$f_1,f_2,\cdots,f_t$ 是 A 中非零多项式，而且 t≥2。称合冲 A-模 Syz(lt(f_1),lt(f_2),…,lt(f_t))的生成元集合 B 为 S-基（S-bases），如果 B 的每个变元都是齐次的，而且恰有 2 个非零分量。

算法 2.6 PID 上的 Gröbner 基算法

输入：$F=\{f_1,\cdots,f_t\}\subseteq R[x_1,\cdots,x_t]\setminus\{0\}$，R 是 PID。

输出：G={$g_1,g_2,\cdots,g_t$}，理想$\langle F\rangle$的 Gröbner 基。

初始化：$G := F$，$\xi=\{\{f,g\}\,|\,f\neq g, f,g\in G\}$。

WHILE $\xi\neq\varnothing$ DO

任选$\{f,g\}\in\xi$，令$lt(f)=c_fX_f$，$lt(g)=c_gX_g$

$\xi := \xi\setminus\{\{f,g\}\}$

COMPUTE $c=\mathrm{lcm}(c_f,c_g)$，$X=\mathrm{lcm}(X_f,X_g)$，

$$S(f,g)=\frac{c}{c_f}\frac{X}{X_f}f+\frac{c}{c_g}\frac{X}{X_g}g\xrightarrow{G}{}_{+}h$$，h 相对 G 是极小的

IF $h\neq 0$ THEN

$\xi := \xi\cup\{\{u,h\}\,|\,\forall u\in G\}$

$G := G\cup\{h\}$

定义 2.6.2（强 Gröbner 基，极小强 Gröbner 基）（参考文献[10]）设 R 是 PID，G={$g_1,g_2,\cdots,g_t$} $\subseteq R[x_1,\cdots,x_n]\setminus\{0\}$。称 G 是理想 I=$\langle g_1,g_2,\cdots,g_t\rangle$的强 Gröbner 基（strong Gröbner bases），如果对每个$f\in I$，都存在某个 i，$1\le i\le t$，使得$lt(g_i)\,|\,lt(f)$。

称 G 是理想 I 的极小强 Gröbner 基（minimal strong Gröbner bases），如果 G 是强极小强 Gröbner 基，而且 $i \neq j, 1 \leq i, j \leq t$，$lt(g_i) \nmid lt(g_j)$。

定理 2.6.1 令 R 是含有单位元的交换环，$A = R[x_1, \cdots, x_n]$。设 I 是环 A 中的理想和 $G=\{g_1, g_2, \cdots, g_t\} \subseteq I$ 是一个有限集，则下列叙述等价：

（1）lt(G)=lt(I)；

（2）$\forall f \in I$，有 $f \xrightarrow{G}_+ 0$；

（3）$\langle G \rangle = I$ 和对 G 的每个子集 F，F 的 S-多项式集和 SP(F)的每个多项式 h，$h \in SP(F)$，有 $h \xrightarrow{G}_+ 0$。

定理 2.6.1 给出了计算 Gröbner 基的算法。即我们可由理想 I 的一组生成元 G 出发，计算它们的 S-多项式，再检验它是否模 G 约化为零。如果不是，将其相应的极小元添加到 G 上，重复此过程，直到每个 S-多项式模 G 都约化到零为止。实际上，这里求环上 Gröbner 基的算法与求域上 Gröbner 基的算法完全平行。对于环的情况，为了得到 S-多项式，我们需要基环是合冲-可解的。整个算法要求是强可计算环。

在本章之中我们讲述了 Gröbner 基的最基本的概念与性质和求解 Gröbner 基的算法。实际上这些算法都是最基本的，其执行效率一般比较低。现在已有许多的文章讨论过如何使算法的效率进一步提高，或者说将算法改进，其目的只有一个：进一步提高 Gröbner 基的实用价值，使其应用范围进一步扩展、使用效率进一步提升，这也是我们开展工作的原因和实现目标。

本章主要参考涉及到的参考文献包括[1]、[2]、[9]、[10]、[11]、[12]、[14]、[28]及[34]等。

第三章 Gröbner 基的生成算法及其并行

Gröbner 基方法求解的一般应用思路如下：设多项式集合 $F\in k[x_1,x_2,\cdots,x_n]$，其中 $k[x_1,x_2,\cdots,x_n]$表示域 k 上的 n 变元多项式环，我们将 F 变换到另一个由多项式构成的具有某种良好性能的集合 G（即 Gröbner 基）上，使 F 和 G“相等”，即生成了相同的理想。正是由于 Gröbner 基的特性，可以使很多在集合 F 上难以解决的问题转换到集合 G 上进行求解，从而通过 Gröbner 基实现简化。

在应用 Gröbner 基求解的过程中，要注意以下的问题：

1）化简多变元多项式：设多项式 $f_i,g,h,i=1,\cdots,s$，$F=\{f_1,\cdots,f_s\}$，有 f 模 F 约化为 h，即 $f\xrightarrow{F}_+h$。

2）多变元多项式一般总是可以进行化简的，同时最终化简式可能并不唯一。要使 F 成为 Gröbner 基，则最终化简式必须是唯一的。

3）化简结束标志：即满足 Gröbner 基基本定理，将运算算法 $Alg(F,Spol(f_i,f_j))$ 的输出值化为非平凡解，同时随着不能化简为零的 S-多项式的增加，G 也不断增大，从而使需要重复化简的可选 S-多项式的数目也进一步增大，这时需要应用 Dickson 引理来最终结束化简过程。

4）算法正确性检验标准：如果 F 是 Gröbner 基，则必有运算算法 $Alg(F,Spol(f_i,f_j))=0,\forall f_i\in F,\forall f_j\in F$ 成立。

在前面的章节中我们介绍了几种 Gröbner 基的生成算法，本章中我们将引入当前以 F4 和 F5 算法为代表的主流生成算法，并对其中间项进行并行约化。主要思路是：在算法中采用结构化高斯消去法对可能产生大型稀疏矩阵予以简化，进而主要采用并行高斯全选主元消去法进行约化，以达到提高算法效率的目的。

3.1 并行计算简介

定义 3.1.1 并行计算（Parallel Computing）（参考文献[32]）是指同时使用多种计算资源解决计算问题的过程。

为执行并行计算，计算资源应包括一台配有多处理机（并行处理）的计算机、一个与网络相连的计算机专有编号，或者两者结合使用。并行计算的主要目的是快速解决大型且复杂的计算问题。

并行计算是在串行计算的基础上演变而来，它仿真自然世界中的事务状态：一个序列中众多同时发生的、复杂且相关的事件。

为利用并行计算，通常计算问题表现为以下特征：

（1）将工作分离成离散部分，有助于同时解决；

（2）随时并及时地执行多个程序指令；

（3）多计算资源下解决问题的耗时要少于单个计算资源下的耗时。

并行计算是相对于串行计算来说的，所谓并行计算分为时间上的并行和空间上的并行。时间上的并行就是指流水线技术，而空间上的并行则是指用多个处理器并发的执行计算。

并行算法的研究主要分为并行计算理论（计算模型、下界，问题可并行性，NC 类和 P-完全性）、并行算法的设计与分析（NC 类问题的有效并行算法的设计和分析方法）和并行算法的实现（硬件平台与软件支撑）三个层次。

为了获得高性能，并行实现中必须考虑以下几个问题：

（1）必须考虑负载平衡，从而保证没有某个计算部分占据大部分运行时间；

（2）必须在并行实现中加入技巧才能将计算扩展到其他并行处理机上；

（3）问题的某些部分虽然是串行的，但可以通过流水线并行策略加快执行；

（4）可能需要特殊的策略来处理不规则的计算。

本文中我们不讨论并行过程中的访问数据和传输中间结果所需的通信量的最小化程度问题，而将重点放在数据并行和任务并行实现方面上。

定义 3.1.2 MPI（Message Passing Interface）（参考文献[30]） MPI 是消息传递并行程序设计的标准之一，当前通用的是 MPI1.1 规范。正在制定的 MPI2.0 规范除支持消息传递外，还支持 MPI 的 I/O 规范和进程管理规范。MPI 正成为并行程序设计事实上的工业标准。 MPI 的实现包括 MPICH、LAM、IBM MPL 等多个版本，最常用和稳定的是 MPICH.MPICH 含三层结构，最上层是 MPI 的 API，基本是点到点通信，和在点到点通信基础上构造的集群通信（Collective Communication）；中间层是 ADI 层（Abstract Device Interface），其中 device 可以简单地理解为某一种底层通信库，ADI 就是对各种不同的底层通信库的不同接口的统一标准；底层是具体的底层通信库。

MPI 环境的初始化和结束流程如下：在调用 MPI 例程之前，各个进程都应该执行 MPI_INIT，接着调用 MPI_COMM_SIZE 获取缺省组(group)的大小，调用 MPI_COMM_RANK 获取调用进程在缺省组中的逻辑编号（从 0 开始）。然后，进程可以根据需要，向其它节点发送消息或接收其它节点的消息，经常调用的函数是 MPI_SEND 和 MPI_RECV。最后，当不需要调用任何 MPI 例程后，调用 MPI_FINALIZE 消除 MPI 环境，进程此时可以结束，也可以继续执行与 MPI 无关的语句。

实际上构成了编写一个完整的 MPI 程序所需例程的最小集要涉及到六个函数：MPI_INIT，MPI_COMM_SIZE，MPI_COMM_RANK，MPI_SEND，MPI_RECV，MPI_FINALIZE。

MPI 的有四个重要特征：Communicator（通信空间），Group（进程组），Context_id（上下文标识），Data Types（数据类型）。

MPI 提供 Communicator 来指定通信操作的上下文，提供了通信操作的执行空间。在某个通信空间（或上下文）中发送的消息必须在相同的空间中接收，不同空间中的消息互不干扰。定义一个 Communicator，也就指定了一组共享该空间的进程，这些进程组成了该 Communicator 的 Group。

Communicator 通过其特征属性 Context_id 来区分；同一个进程不同的 Communicator 有不同的 Context_id。因此 Context_id 是另一个区分消息的标志。

MPI 引入消息的 Data Type 属性的目的有两个：一是支持异构系统计算；二是允许消息来自不连续的或类型不一致的存储区，例如，可以传送数组的一列，或传送一个结构值，而该结构的每个元素的类型不同。Data Types 定义了消息中不连续的数据项及其可能不同的数据类型。Data Type 由应用程序在执行时通过基本的数据类型创建。

3.2 生成算法伪代码

一个完整的Gröbner 基的生成算法应该主要包含以下几个部分：

1）主函数算法：给定任意有限多变元多项式集合F，找到一个多项式集合G，使$\langle F\rangle=\langle G\rangle$，则G就是Gröbner基。

2）运算法则算法：

初始化：
G:=F
$\forall f_i \in G, f_j \in G, i \in (1,\cdots,n), j \in (1,\cdots,n)$

运算：由f_i，f_j生成的 S-多项式，并将其关于 G 化简到 h。

如果 $h=0$，则选择下一个多项式对；

如果 $h\neq 0$，则将h添加到G中，并重复以上步骤。

3）求S-多项式：$f,g\in k[x_1,\cdots,x_n]\backslash\{0\}$，L=lcm(lp($f$)，lp($f$))，其中lcm表示最小公倍。令

$$S(f,g)=\frac{L}{lt(f)}f-\frac{L}{lt(g)}g$$

多项式$S(f,g)$称为f和g的S-多项式（S-Polynomials）（参考文献[1]）。

这里需要注意：S-多项式虽然是Gröbner 基理论的算法中的核心部分，但 Gröbner 基与S-多项式并没有直接关系。在本文中我们所主要关注的正是S-多项式的并行约化问题。

4）化简方法：Gröbner 基算法针对不同应用领域有不同的化简方式：对线性多项式采用高斯消元算法；对单变元多项式采用欧几里得算法。我们在本文中主要采用并行高斯消元算法。

本文中我们所研究的主要内容就是算法的化简问题。本节我们将以伪代码的形式来描述Gröbner基生成算法及其并行改进。在此我们强调本文中所改进的原始算法来主要自于参考文献[3]、[4]、[5]、[6]、[7]、[8]、[11]、[14]、[26]和[35]中所描述的当前主流生成算法。

下面是主要算法的伪代码：

首先我们将主函数的算法描述如下：

算法3.1 主函数算法

假设：全局变量 r　　//多项式阵列

　　全局变量 Rules　//简化标准阵列

输入：齐次多项式序列F=($f_1,\cdots,f_m$)$\subseteq\mathbb{F}[x_1,\cdots,x_n]$，其中，$f_i\neq 0\,(0\leqslant i\leqslant m)$

输出：理想$\langle F\rangle$的Gröbner基

Main(F)

初始化：

m:=|F|； Rules:=$(\varnothing)_{j=1}^{m}$； r:=∅；

$r_m:=(e_m,HC(f_m)^{-1}f_m)$；　　//$e_m$表示第 m 个标准基向量

$G:=(\varnothing)_{i=1}^{m}$； $G_m:=\{m\}$；

FOR i=m-1 to 1 DO

　　$G_i:=$ Alg_Func(f_i,i,G)

　　IF ($\exists k\in G_i$) poly(r_k)=1 THEN

　　return {1}

　　END IF

END FOR

return {poly(r_k)|$k\in G_i$}

下面我们将运算法则算法描述如下：

算法3.2 Alg_Func 算法

输入：i∈N，多项式$f\in\mathbb{F}[X]$，有限子集$G_{i+1}\in R$，其中多项式(G_{i+1})是由($f_{i+1},\cdots,f_m$)所构成的Gröbner基

输出：由全局变量r_i，$i\in G_i$所构成的集合G_i

Alg_Func(f，i，G_{i+1})

初始化：

$r_i:=(e_i, HC(f)^{-1}f)$；　　//e_i表示第i个标准基向量

$G_i := G_{i+1} \cup \{i\}$

$P := \varnothing$

FOR $j \in G_{i+1}$ DO

$P := P \cup Crit\_Pair(r_i, j, i, G_{i+1})$

END FOR

WHILE $P \neq \varnothing$ DO

$d := \min\{deg(p) \mid p \in P\}$

$P_d := \{p \in P \mid deg(p) = d\}$

$P := P \setminus P_d$

$S_d := S\text{-}Pols(P_d)$

$R_d := Reduction(S_d, G_i, G_{i+1})$

FOR $k \in R_d$ DO

$P := P \cup \{Crit\_Pair(k, l, i, G_{i+1}) \mid l \in G_i\}$

$G_i := G_i \cup \{k\}$

END FOR

END WHILE

return G_i

本算法中使用到了3个辅助中间变量函数：Crit_Pair函数，即构造关键对；Spols函数，构造S-多项式；Reduction函数，用以简化当前列表中的多项式。

下面首先我们将给出Crit_Pair关键对函数的概念：

定义3.2.1（参考文献[5]）设多项式$(r_k, r_l) \in R^2$的关键对为

$Crit\_Pair(r_k, r_l) = (lcm(HT(r_k), HT(r_l)), t \setminus HT(r_k), r_k, t \setminus HT(r_l), r_l)$

同时满足 $S(u_1 r_k) \succ S(u_2 r_l)$ 。

我们称这样的一个关键对的度为$deg(lcm(HT(r_k), HT(r_l)))$。

以下是求关键对的算法—Crit_Pair算法：

算法3.3 Crit_Pair算法

输入： 定义多项式r_k，$r_l \in R$，$i \in \mathbb{N}$

输出： 对应的关键对，如果(r_k, r_l)或(r_l, r_k)是标准化的

Crit_Pair (k，l，i，G)

初始化：

$t := lcm(HT(r_k), HT(r_l))$

$u_1 := t \setminus HT(r_k)$

$u_2 := t \setminus HT(r_l)$

IF $S(u_1 r_k) \prec S(u_2 r_l)$ THEN

$k' := k$

$k := l$ //交换 k 和 l

$l := k'$

$u_1 := t \setminus HT(r_k)$

$u_2 := t \setminus HT(r_l)$

END IF

$(t_1, e_{k_1}) := S(r_k)$

$(t_2, e_{k_2}) := S(r_l)$

IF $u_1 t_1$ 由 G_{k_1+1} 最优化简 THEN

return $\varnothing$

END IF

IF $u_2 t_2$ 由 G_{k_2+1} 最优化简 THEN

return $\varnothing$

END IF

return $\{(t, u_1, k, u_2, l)\}$

S-多项式是Gröbner 基求解算法中的核心部分，下面我们将其介绍如下：

算法3.4 SPols算法 // SPols函数指求S-多项式

输入： 关键对的集合P=[$p_1, p_2, \cdots, p_h$]

输出： 由全局变量r_i, $i \in \mathbb{N}$所构成的序列F

SPols(B)

初始化： $p_i = (t_i, u_i, k_i, v_i, l_i)$， 其中i=1,…, h

$F := \varnothing$

FOR i=1 TO h DO

$c_1 := HC(r_{k_i})$

$c_2 := HC(r_{l_i})$

$s := u_i poly(r_{k_i}) - \frac{c_1}{c_2} v_i poly(r_{l_i})$

IF $s \neq 0 \,\&\, (\neg Rewritable(u_i, r_{k_i})) \,\&\, (\neg Rewritable(v_i, r_{l_i}))$ THEN

$N := |r| + 1$

$r_N := (u_i S(r_{i_1}), s)$

Add_Rules(N)

$F_{|F|+1} := N$

END IF

END FOR

sort F 使得 i<j ⇒ $S(r_{F_i}) \prec S(r_{F_j})$

return F

本算法与以前的主要算法（如Buchberger算法等）的主要区别就在以下算法：化简列表中的多项式并返回相应值。因此我们对Reduction算法进行改进：采用辅助函数Top_Reduction，进行基本的化简步骤。Top_Reduction函数的结果是返回（r，F′）对，其中$r \in R$，F′为多项式列表。F′=∅意味着r不可约（或为0）。如果$F' \neq \varnothing$且r=∅，则表示需要再次对F′中的所有元素再次运行Top_Reduction函数。

下面将介绍多项式化简算法—Reduction算法。

算法3.5 Reduction算法

输入：由全局变量r_i，$i \in N$所构成的集合T和G′，集合序列$G=(G_i)_{i=1}^m$，$\exists k \in N$

输出：由全局变量r_i，$i \in N$所构成的集合D

Reduction (T，G′，G)

初始化：D := ∅

WHILE T≠0 DO

选则$k \in T$使$S(r_k)$为序列$\{S(r_{k'}) | k' \in T\}$中的最小项，即为$\prec$-minimal

$T := T \setminus \{k\}$

$h := poly(r_k)$的标准形式，记作$\{poly(r_j) | j \in G'\}$

$r_k := \{S(r_k)，h\}$

K, T′ := TopReduction(k，G′∪D，G)

$D := D \cup K$

$T := T \cup T'$

END WHILE

return D

为了应用Top_Reduction函数我们需要一个函数来测试多项式列表中的多项式的关键项是否可分解。当不能进一步约分化简时，返回一个化简值或是∅。

算法3.6 Seek_Reductor算法

输入：由全局变量r_i，$i \in N$所构成的集合G′，集合序列$G=(G_i)_{i=1}^m$，$\exists k \in N$

输出：由全局变量r_i，$i \in N$所构成的集合R（可能为空集∅）

Seek_Reductor (k，G′，G)

初始化： $t := HT(r_k)$

FOR $j \in G'$ DO

$t' := HT(r_j)$

$v_j e_{k_j} = S(r_j)$ // e_{k_j} 表示第 r_j 个标准基向量

IF $t' | t$ THEN

$u := t/t'$ -

IF $uS(r_j) = S(r_k)$ or Rewritable(u,j,k) or uv_j 由 G_{k_j+1} 最优简化THEN

continue

ELSE return {j}

END IF

END IF

END FOR

return ∅

算法3.7 Top_Reduction算法

输入：由全局变量 r_i ,$i \in N$所构成的集合G′，集合序列 $G=(G_i)_{i=1}^m$，$\exists k \in N$

输出：由全局变量 r_i ,$i \in N$所构成的集合D和T（可能为空集∅）

Top_Reduction (k， G′， G)

IF $poly(r_k)=0$ THEN

printf("化简过程将成为死循环，无法结束！")

return ∅,∅

END IF

初始化：

$p := poly(r_k)$

$J := Seek\_Reductor(k, G', G)$

IF $J = \varnothing$ THEN

$p := p/HC(p)$

$r_k := (S(r_k), p)$

return {k},∅

ELSE

设 $j \in J$ 则

$q := poly(r_j)$

$u := HT(p)/HT(q)$

$p := p - HC(p)/HC(q)uq$

IF $uS(r_j) \prec S(r_k)$ THEN

$r_k := (S(r_k), p)$

return $\varnothing, \{k\}$

ELSE

$N := |r| + 1$

$r_N := (uS(r_j), p)$

Add_Rules(N)

return $\varnothing, \{k, N\}$

END IF

END IF

以下是各化简规则算法的伪代码：

我们应用阵列Rule存储变量Rules，阵列Rule的每个元素都是T×N的劣表中的一个元素。在进行化简之前，我们首先进行一下初始化：

FOR i=1 TO m DO

$Rules_i := \varnothing$

下面的算法用以合并Rule阵列。

算法3.8 Add_Rules算法

输入： $j \in \mathbb{N}$

输出：全局变量$Rules_i$

Add_Rules (j)

$(t, e_i) := S(r_j)$

$Rules_i := concat(((t, j)), Rules_i)$

//设 $a=(a_1,\cdots,a_t), b=(b_1,\cdots,b_j)$，若 c: =concat(a, b)，则有 $c=(a_1,\cdots,a_t,b_1,\cdots,b_j)$

算法3.9过程是经Rewrite算法处理后（u，k）发生变化，则返回值是真（true）。

算法3.9 Rewritable算法

输入： $u \in \mathcal{T}$, $\exists k \in \mathbb{N}$

输出：true 或者 false

Rewritable （u，k）

return $k' \neq k$

算法3.10过程是要化简（u, k）：

算法3.10 Rewrite算法

输入： $u \in \mathcal{T}$, $\exists k \in \mathbb{N}$

输出：k

Rewrite (u，k)

```
(v, e_l) := S(r_k)
FOR  j=1  TO  |Rules_l|  DO
      (t, j') := Rules_{l,j}
      IF  t| uv  THEN
          return  j'
      END IF
END FOR
return k
```

3.3 算法并行及分析

在实际工作开始的时候先要考虑的是如何选择策略。如果对于所有 $k\neq\varnothing$，size(Select(k))=1，那么改进的生成算法即等价为 Buchberger 算法，Select 函数对应着 Buchberger 界法的选择策略。选择策略，即 Select 函数的选择对算法的性能是十分重要的，这是 Gröbner 基计算过程中是最难的一步(其他的步骤是理想的消去和分解)。其中一个原因是算法的输入只是没有数学结构的 $R[x]$ 的一个子集。我们想在计算的开始给这些多项式一些结构，我们用 d-Gröbner 基的定义。

算法 3.11 Select(P)算法

输入： 一个关键对链表 P

输出：证实确实是一个关键对的链表

Select (P)

d:=min{deg(lcm(p)), $p\in$ P }

P_d :={ $p\in$ P | deg(lcm(p))=d }

return P_d

根据前人的经验我们知道，经过检验的最好的 Select 函数实现是取全序最小的所有关键对，它的实现可以用上面的算法表示出来。我们把这种策略称为正规策略，则对应每次选择计算的结果即为所给多项式集合的 d-Gröbner 基。

通过实验观察可以看出，在算法的符号处理过程中由多项式生成的矩阵多是大型稀疏矩阵，约化这样的大型稀疏矩阵将成为我们要面临的主要问题之一，这也是我们要并行实现的主要部分。由前人的工作我们知道，矩阵的约化问题可以转化为解方程组问题，有几种解大型线性方程组系统的方法，以下我们将选择高斯全主元消去法给予介绍，这也是我们所采用的主要消元法。

高斯全主元消去法是一种常用的解线性方程组的算法。它具有较高的精度和稳定性，但时间复杂度也较高，并且随着方程组阶数增长，时间复杂度增长较快。因此，将其并行化改进是非常有必要的。

假设我们要处理的多项式生成矩阵可以化为方程组 Ax=b，式中：A 是一个 n*n 的矩阵；b 是一个 n*1 的列向量。

首先我们来分析一下输入数据时的情况。输入数据一般较多，因此从文件输入。对于传统串行算法，这一步需串行的将 A 与 b 中的数据依次读入内存，总时间为 $t_{in}=n(n+1)t_r$，式中：t_r 为程序读取单位数据的时间；n 为启动并行程序中的进程数。

本文中采用 MPI 并行 I/O 输入数据。这种技术在 MPI-2 标准中实现，由许多快速读写文件的 API 组成，是一种比较新的技术。MPI 并行 I/O 的特点是，各进程并发地从文件的不同位置读入数据。所以 MPI 并行 I/O 与传统方法相比，无论是运行时间还是对内存的需求量都相应减小，此时的执行时间为 $t_{in}=\frac{1}{N}n(n+1)t_r$。

相应的对于那些没有采用 MPI 并行 I/O 技术的并行实现，一般先由主进程将所有数据读入自己的内存，再将数据发送到相应的其他进程，所消耗的总时间为

$$t_{in}=n(n+1)t_r+\frac{N-1}{N}n(n+1)t_t,$$

式中 t_t 为进程之间传送单位数据的时间（参考文献[30]）。

可见，当输入数据时采用 MPI 并行 I/O 技术节省了时间。在内存需求方面，如果未采用 MPI 并行 I/O 技术，需要的内存是(n+1)ns，而当采用 MPI 并行 I/O 技术改进后，单个节点内存的需求只有(n+1)ns/N，式中 s 为程序中所用的类型单位数据占用的内存空间。

下面我们来看看选主元的情况。假设选主元执行到第 i 步，对于传统串行算法，这一步需要在系数矩阵右下角的(n-i+1)*(n-i+1)子阵中寻找主元，因此一步执行时间大约是 $t_{pi}(i)=(n-i+1)^2t_{a,s}$，总的执行时间是(进程内部内存中交换行的时间由于较短，我们不予考虑)：

$$t_{pi}=\sum_{i=1}^{n}(n-i+1)^2t_{a,s}=\sum_{i=1}^{n}i^2t_{a,s}=\frac{n(n+1)(2n+1)}{6}t_{a,s},$$

式中 $t_{a,s}$ 为机器执行一次加减法的时间。

并行高斯全选主元消去法中各进程先独立的选出本地主元，然后通过全局规约的方法选出全局的主元。在选定全局主元之后，按照并行算法，要将主元所在行与第 i 行相互交换(假设当时是进行第 i 步选主元)，将主元所在列与第 i 列交换。在并行算法中，由于采用了面向行的分割法，交换列还是在同一个进程中进行，而交换行将涉及到进程之间的通信。为了避免进程间通信所带来的开销，对算法进行了改进，在程序中引入了标志数组 num[]，记录了每一行在矩阵中对应的行号，这样当需要交换第 i 与第 j 行的时候，只需交换 num[i]与 num[j]的值，而不需要将整行数据全部进行交换，省去了大量的数据通信。在对算法进行改进后，这一步

的执行时间减少为 $t_{pi}=\frac{n(n+1)(2n+1)}{6N}t_{a,s}$。

如果不进行这一步优化，在最优的情况下(每次消元都不需要交换行)，时间与该优化的算法相当，但在最坏情况下(每次消元都需要换行)，这一步的时间将变为：

$$t_{pi}=\frac{n(n+1)(2n+1)}{6N}t_{a,s}+2n(n+1)t_t。$$

假设数据是随机分布的，那么这一步的执行时间的数学期望将是：

$$t_{pi}=\frac{n(n+1)(2n+1)}{6N}t_{a,s}+\frac{N-1}{N}2n(n+1)t_t。$$

可见，算法的并行改进确实缩短了程序运行时间。

接下来我们将分析消去过程。这里我们采用多线程消去。

实验中可以看出 Gröbner 基的计算过程中产生的多项式对应的系数矩阵大多是稀疏矩阵。对于稀疏方程组系统存在一些非常简单的解法，其中的某些方法可以用作 MPI 并行算法应用之前的预处理步骤，如结构化高斯消去法。下面将简要介绍一下我们所采用的结构化高斯消去法。因为该方法的目的是用于减小要处理的稀疏矩阵的维数。在某些情况下，该方法本身就可以作为一种高效的解稀疏线性方程组系统的方法。

这种方法的产生是基于对稀疏矩阵的一种简单的观察。通过观察可以看出在稀疏矩阵的某些列上只有一个非零元，则用其余的行对该行作约化时，对该列上的这一非零元素不会产生影响。我们的想法是依次去掉含单一非零元素的那些行，即去掉对应这一非零元素的变量，先将其保存在另外的地方，对矩阵其余部分作约化，当去掉这些行以后发现出现了更多的只含一个非零元素的列，则重复上面的过程直到矩阵中不再有只含单一非零元的列，这样由矩阵的稀疏性，就大大的减小了矩阵的维数。然后对矩阵的其余部分作约化，最后还要恢复被删去的行。我们所采用的方法是将先前删除的行按与被删除的顺序相反的顺序添加到经过约化的矩阵的后边，用已经是约化的部分对这些先前被删除的行作约化来实现矩阵的整体约化。其中对剩下部分作约化时必然会由矩阵的行初等变换增加矩阵中的非零元的个数，可以取使得增加的非零元个数最少的行对其余的行作约化。由于本人经验有限所以在实现中，忽略了这一步，只应用了结构化高斯消去法的第一步。结构化高斯消去法的算法（参考文献[45]）简介如下：

算法 3.12　结构化高斯消去法

第一步：　从稀疏矩阵中删除只含单一非零元的列中非零元所有的行。重复这一步骤直到矩阵中没有只含单一非零元的列。

第二步：选择非零元数最多的αM 列($\alpha>0$)，称这些列为“Heavy”，其余的列为“Light”。典型的α值可以是 1/32，矩阵中的行的权重定义为在它的 Light 列中的非零元素的个数。

第三步：消去权重为 1 的行的第一个非零系数对应的变量，也就是从其余的对应这一变量有非零系数的那些行减去权重为 1 的行的恰当倍数。从这些行中消去这一变量，这就出现了引入非零元的问题。由这一过程引入的非零元数为 $(\omega_i-1)(k-1)-\omega_i-(k-1)=(\omega_i-2)(k-1)-\omega_i$，这里 ω_i 为权重，k 为含这一非零元的行数。

第四步：如果产生的矩阵的行数比列数多 r 行，则丢弃权重最大的 r 行。

第五步：消去引入最少非零元的变量。

目前使用多线程通常有 2 种选择：OpenMP 和 Pthreads。OpenMP 容易使用，但灵活性较差，不能控制线程的创建与销毁，不能控制具体任务的划分等，在性能上也有一定的损失；Pthreads 虽然使用较为复杂，但可以完全控制线程的生命周期，性能上的损失也较小。文中采用 Pthreads 多线程。

采用多线程会带来一些额外的开销，例如线程的创建、销毁和等待线程等。在算法的消去阶段，每一步的计算量是不同的，第 1 步消去的计算量最大，最后一步消去计算量最小，总之，第 i 步消去需要处理一个(n-i+1) * (n-i+2)的矩阵。由此可知，当消元刚开始进行，计算量较大，线程数应尽量大一些，以便提高性能；但随着消去步数的增加，计算量逐渐减小，线程带来的开销所占的比重会增大，线程数应当小一些。因此应根据消元步数的不同选择最适合的线程个数。

假设消元执行到第 i 步，线程个数为 m。那么如果创建一个线程的时间是 t_e，消灭一个线程的时间是 t_k，线程之间等待的平均时间是 t_ω (如果负载平衡较好，t_ω 可以接近零)，m 个线程带来的的总开销是 $(t_e+t_k+t_\omega)(m-1)$，执行一步消去需要一次乘法和一次加法，如果 m 个线程的任务量相同，并且系数矩阵的数据完全随机分步，消去计算所用的时间是

$$\frac{(n-i)(n-i+1)}{Nm}(t_{a,s}+t_{m,d})。$$

那么在 m 个线程的情况下，第 i 步 $(0\le i\le n-1)$ 消去的总时间

$$t_e(i,m)=(t_e+t_k+t_\omega)(m-1)+\frac{(n-i)(n-i+1)}{Nm}(t_{a,s}+t_{m,d}),$$

式中 $t_{m,d}$ 为执行一次乘除法所用的时间。

这样要解决的问题可归纳为选择合适的 m，使得第 i 步的总时间 $t_e(i,m)$ 的值最小。为此，对 $t_e(i,m)$ 求 m 的偏导数得

$$\frac{\partial t_e(i,m)}{\partial m}=(t_e+t_k+t_\omega)$$

令偏导数的值等于 0，解得当 $t_e(i,m)$ 值最小时，m 的取值为

$$m_o=\sqrt{\frac{(n-i)(n-i+1)}{N}\frac{t_{a,s}+t_{m,d}}{t_e+t_k+t_\omega}}$$

其中 $\frac{t_{a,s}+t_{m,d}}{t_e+t_k+t_\omega}$ 可以看作是计算时间与创建线程时间的比值，这个值与 CPU 速度以及所用的操作系统等有关。用实验的方法可测得在并行机上的值，因此该值在程序一开始就可以得到，每一步不用重新计算。

对于串行算法，这一步的执行时间是 $(n-i)(n-i+1)(t_{a,s}+t_{m,d})$。

对于并行算法，如果不用多线程，第 i 步消元所用的时间(m=1 时)为

$$t_e(i,1)=\frac{(n-i)(n-i+1)}{N}(t_{a,s}+t_{m,d}),$$

很明显相对于串行算法减少运行时间。如果使用多线程，第 i 步消元所用的时间是

$$t_e(i,M)=(t_e+t_k+t_\omega)(M-1)+\frac{(n-i)(n-i+1)}{NM}(t_{a,s}+t_{m,d}),$$

其中 M 表示并行机每个节点的 CPU 数目。

经过以上三步之后再进行回代求解，从而可以完成对稀疏矩阵的约化。

我们采用的编程语言是 C+MPI，由于考虑到计算的精度，避免计算过程中可能出现的大整数计算和算法执行过程中的通信要求，因而我们采用在有限域上实现算法。

首先对多项式进行处理，然后由所得的多项式集合生成系数矩阵。对矩阵的处理过程，先用结构化高斯消去法进行预处理，再用高斯全选主元消去法进行并行约化，最后将在最开始被删去的行添加到已经被约化的矩阵的后边，用单进程进行一次串行约化得到最终的约化结果。再按算法描述的步骤进行运算，最终得到给定的多项式集合生成的理想的 Gröbner 基。

由于考虑到算法执行过程中生成的矩阵的稀疏性和规模，对于生成的系数矩阵采用稀疏存储方式，用二维动态数组表示矩阵：每一行的第一个元素记录该行在矩阵中的顺序，第二个元素记录该行元素的个数，以后的每相邻的两个元素的第一个元素记录第二个出现在系数矩阵中该行中的位置，第二个元素为该行中第一个元素表示的位置处的非零元素值。在应用结构化高斯消去法时用变量 dd 表示结构化高斯消去法的第一步重复的次数，数组 ds[dd]记录前 dd 重约化中被删去的行数，最后在恢复的过程中按与被删去时的相反的顺序将这些行添加到经约化的矩阵的后边，再经过一次约化。因为考虑到矩阵的结构，我们用经过并行约化的矩阵先生成多项式链表，然后提取其系数形成矩阵，形成上三角阵，最后对矩阵的下部进行串行约化。

需要说明的是，当多项式数量不大时用并行约化效率并不一定会比进行串行约化时的效率高，因此我们按多项式的数量分不同的情况进行处理：当多项式数大于等于 M1 时应用结构化高斯消去法，当剩余的矩阵的行数大于等于 M2 时应用并行约化，否则用单进程进行运算。对于一般情况，取 M1=5000 与 M2=2000。 而

且考虑到负载平衡问题，我们选择的循环分配的处理器映射方式，即由主进程负责进行文件的读取，然后将任务以循环分配的方式分配给其它的进程。

程序中应用的各函数和与各个结构有关的函数分别在各自的头文件和实现文件中声明和实现。对于初始多项式集合的输入采用文件输入方式。由于存取方式问题和时间有限，没能实现文件的共享读取，只用主进程进行文件存取。所以我们的算法效率还有待进一步改进。

我们所采用的方法是将先前删除的行按与被删除的顺序相反的顺序添加到经过约化的矩阵的后边，用已经是约化的部分对这些先前被删除的行作约化来实现矩阵的整体约化。下面是恢复被删去的行时的部分程序清单：

```
while(dd>0){
    rewind (fp);      //指针fp的位置倒回到文件的开头
    s=0;
    k=i;
    if(dd-2>=0 ){
        l=ds[dd-1]-ds[dd-2];
        while(s<ds[dd-2]){ inputAF(t, fp);
                           s++; }
        }
   else l=ds[dd-1];
   for( ;i<k+l; i++){
        inputAF(t ,fp);
       MF->ets[i] = (int *)calloc (2*(t[1]+1), sizeof(int));
       memcpy (MF->ets[i], t, 2*(t[1]+1)*sizeof(int));
       }
  dd--;
}
```

我们实现并行处理还包括经结构化高斯消去法处理之后剩余的稀疏矩阵的化简部分，其应用了并行高斯全选主元消去法。

从进程对应主进程设计，只是当约化结束后将本进程中的约化结果发送至主进程以供继续处理。

虽然在算法的执行过程中生成的稀疏矩阵不具有一定的结构，但是在结构化矩阵处理技术中提到可以通过一定的调整策略使矩阵具有一定的结构，因此在算法的实现中可以应用结构化矩阵计算技术进行处理，如果能实现的话将大大提高算法的效率。而且如果要使算法有意义，应使算法能处理大整数情况，如果能够应用 Hensel 提升将有限域中计算得到的 Gröbner 基提升到大整数情况将是另一种计算实数域上的 Gröbner 基的有效的途径。

由于使用结构化高斯消去法使其在恢复过程中可能会引入多余的非零元，所以对于算法的实际效率还有待进一步考察。由于时间的限制，这些都是我们下一

步实验的目标。

3.4 实验实例结果检验

我们给出一组多项式例子来求其Gröbner基：设 $F=[f_1,f_2,f_3,f_4]$，其中 $f_1=xyzt-1, f_2=xyz+xyt+xzt+yzt, f_3=xy+yz+xt+zt, f_4=x+y+z+t$，并设 $x>y>z>t$。

写成文件为：

4, 4

2：1(1,1,1,1)-1(0,0,0,0)

4：1(1,1,1,0)+1(1,1,0,1)+1(1,0,1,1)+1(0,1,1,1)

4：1(1,1,0,0)+1(0,1,1,0)+1(1,0,0,1)+1(0,0,1,1)

4：1(1,0,0,0)+1(0,1,0,0)+1(0,0,1,0)+1(0,0,0,1)

这里前两个数分别表示多项式数和变量数，下面每行的第一个元素表示该多项式中的单项式数，“:”后是系数与项对。经过计算我们得到此多项式集合生成的理想的Gröbner基为：

List of MP:

1(0,0,2,4)+1(0,1,1,0)-1(0,1,0,1)+1(0,0,1,1)-2(0,0,0,2)

1(0,0,3,2)+1(0,0,2,3)-1(1,0,0,0)-1(0,1,0,0)-2(0,0,1,0)-2(0,0,0,1)

1(0,1,0,4)+1(0,0,0,5)-1(0,1,0,0)-1(0,0,0,1)

1(0,1,1,2)+1(0,0,2,2)-1(0,1,0,3)+1(0,0,1,3)-1(0,0,0,4)-1(0,0,0,0)

1(0,1,2,0)+1(0,0,2,1)+1(1,0,0,2)+1(0,0,1,2)

1(0,2,0,0)+2(0,1,0,1)+1(0,0,0,2)

1(1,0,0,0)+1(0,1,0,0)-1(0,0,1,0)+1(0,0,0,1)

即多项式集合为：$G=[f_1',f_2',f_3',f_4',f_5',f_6',f_7']$，其中 $f_1'=z^2t^4+yz-yt+zt-2t^2$，$f_2'=z^3t^2+z^2t^3-x-y-2z-2t$，$f_3'=yt^4+t^5-y-t$，$f_4'=yzt^2+z^2t^2-yt^3+zt^3-t^4-1$，$f_5'=yz^2+z^2t+xt^2+zt^2$，$f_6'=y^2+2yt+t^2$，$f_7'=x+y+z+t$。

可以看出我们得出的Gröbner基为非约化的，所以要得到既约Gröbner基还要对我们的结果再做一次约化，这样就可以保证我们最后所得的Gröbner基为既约的。这可以看作对原算法的一点改进。但这又引入了多余的多项式，所以在最后的约化过程中当有新的项加入时，给新加入的多项式加上标记，约化结束后，将新加入的多项式去掉，则得到既约Gröbner基（RGB）。

经过化简之后，我们最终可以得到的Gröbner基的一组标准基：

$G=\{xt+xy+yz+zt, xyt+xyz+xzt+yzt, -1+xyzt, t+x+y+z\}$，其项序为 $x>y>z>t$。

本运行结果可以参考数学软件Mathematica的运行结果进行检验，我们所使用

的是Mathematica 5.0版本（参见附录）。

Mathematica是一个集成化的计算机软件系统，它的主要功能包括三个方面：符号运算、数值计算和图形功能，其中符号计算能力最强。Mathematica的基本符号计算功能表现在能作如下的符号运算：多项式的各种运算(四则运算、展开和因式分解等)、有理函数的各种运算、求多项式方程、有理方程的精确解、符号向量及符号矩阵的各种运算、集合、表的各种运算、求极限、导数、不定积分、幂级数的展开和求解某些微分方程等。此外还有一些能处理特殊领域符号计算的程序包。具体指令为：

输入：GroebnerBasis[$\{f_1,f_2,...,f_m\},\{x_1,x_2,...,x_n\}$]

操作符为“Shift+Enter”

其中，$\{f_1,f_2,...,f_m\}$是一组多元多次多项式，$\{x_1,x_2,...,x_n\}$是这些多项式的未知数，且以 lex order 为项序，即要求$x_1>x_2>...>x_n$。

其他数学软件如Maple也有类似的指令可以求得化简过的Gröbner基。

第四章 Gröbner 基的并行化应用

随着社会信息化的发展，信息安全为保障社会信息的安全、维护社会的安定和团结，发挥着极其重要的作用。信息安全核心的技术是信息的加密解密技术，而加密解密技术都是建立在一定的数学难题之上的。为了保护数据在传递过程中不被别人窃听或修改，必须对数据进行加密(加密后的数据称为密文)，这样，即使别人窃取了数据(密文)，由于没有密钥而无法将之还原成明文(未经加密数据)，从而保证了数据的安全性，接收方因有正确的密钥，因此可以将密文还原成正确的明文。本文中应用 Gröbner 基方法来辅助进行信息的保密工作，提出了将基于 Gröbner 基方法的零知识证明用于身份认证。

4.1 信息安全

信息安全（Information Security，InfoSec）其本身包括的范围很大，主要任务就是要采取措施（技术手段及有效管理）让这些信息资产免遭威胁，或者将威胁带来的后果降到最低程度，以此维护组织的正常运作。

凡是涉及到保密性、完整性、可用性、可追溯性、真实性和可靠性保护等方面的技术和理论，都是信息安全所要研究的范畴，也是信息安全所要实现的目标。信息安全通常强调所谓 CIA 三元组的目标，即保密性、完整性和可用性。网络环境下的信息安全体系是保证信息安全的关键，包括计算机安全操作系统、各种安全协议、安全机制（数字签名，信息认证，数据加密等），直至安全系统，其中任何一个安全漏洞便可以威胁全局安全。信息安全服务至少应该包括支持信息网络安全服务的基本理论，以及基于新一代信息网络体系结构的网络安全服务体系等结构。

信息安全是指信息网络的硬件、软件及其系统中的数据受到保护，免受偶然的或者恶意的原因而遭到破坏、更改、泄露，系统可以连续可靠正常地运行，信息服务不中断。

信息安全主要涉及计算机科学、网络技术、通信技术、密码技术、信息安全技术、应用数学、数论、信息论等。

信息安全在社会信息化的今天有着极其重要作用和重要应用，其应用范围非常广泛。目前所研究的范围主要包括：信息的加密和解密，信息的数字签名，密钥分发、管理机制，信息隐匿，零知识证明等安全机制。

目前 Gröbner 基方法已经在信息安全方面有了一定的应用，主要包括：应用 Gröbner 基来做信息的加密和解密；基于 Gröbner 基的数字签名；基于 Gröbner 基

建立的公共密钥分发、管理机制等诸多方面。

目前 Gröbner 基方法已经在对称加密和非对称加密方面有了一定的应用，如 RSA 算法中密钥的计算等；离散对数密码方面，如 DSA 数字签名；椭圆曲线加密解密中也有了一定的应用。

4.2 零知识证明

零知识证明(zero-knowledge proof)的思想是证明自己有某些知识而不将该知识的内容泄露给对话者。零知识证明的目的是在不给予别人对秘密信息的访问权限的前提下，证明自己有对该秘密信息的访问权限。

定义4.2.1（参考文献[44]）设（G,•）和（G',∘）是两个群，映射 $f:G\to G'$ 叫做群G到群 G' 的同态(homomorphism)，是指对任意a, b∈G，$f(a\bullet b)=f(a)\bullet f(b)$（或简记为 $f(ab)=f(a)f(b)$）。如果同态 f 是满的（surjective），即 f 是映上，称 f 为满同态（epimorphism）。如果同态 f 是单射（injective），即为一一映射，称 f 为单同态（monomorphism）。如果同态 f 是单射同时是满射，称f为G到 G' 的同构（isomorphism），此时称群G到 G' 是同构的（isomorphic），用G≃ G' 表示。

同构群本质上是相同的，即它们具有完全相同的群结构。

图同构是 NP 完整(NP-complete)的。即使构造一个问题只需要适量的时间和空间，解决它却需要花费几百万台计算机几百万年的时间。图是由边连接起来的顶点的集合。每条边严格地连接两个顶点，但并不是每一对顶点(一定)由一条边连接。有些图同其它图是同构的，即：

对于同构的图 G 和 G'，存在一个一一对应的函数关系 F：

（1）F 的定义域是 G 的顶点集；

（2）F 的值域是 G' 的顶点集；

（3）当且仅当[g_1,g_2]是 G 中的一条边，[F(g_1),F(g_2)]才是 G' 中的一条边。

假定示证者 P 声称知道图 G 和 G' 之间是图同构。实际上这意味着他自己构造了这两幅同构的图，或者至少是构造图的人提供了同构性。如果以前公布过：示证者 P 是知道 G 和 G' 之间的同构的人，那么知道这两幅图同构可能是他证明自己身份的方式。显然，仅仅通过直接显示同构会允许任何观察者冒充示证者 P，所以采用这样的方式并不好。

下面是示证者 P 所做的用以证明其自己知道同构的步骤：

（1）示证者 P 随机的改变图 G 的次序以产生另外一幅图 H。由于示证者 P 知道 G 和 G' 之间的同构，因此她很容易同时找到 G' 和 H 之间的同构；

（2）示证者 P 将 H 发送给验证者 V；

（3）验证者 V 可能要求示证者 P 证明：要么(a)H 和 G 同构；要么(b)H 和 G'

同构。但是，验证者 V 可以不要求两者都证明(如果他同时获得了对两者的证明，那么他可以自己证明 G 和G'之间的同构了)：

（4）示证者 P 提供验证者 V 所要求的证明。

上述验证过程表明：如果一个假冒示证者 P 的人不知道 G 和G'之间的同构，冒充者能做的最多是试图假冒与 G 同构的 H(同验证者 V 一样知道 G 和G')，并希望验证者 V 不要求G'和 H 之间的同构。或者，假冒示证者 P 的人可以试图假冒从G'构造的 H，并且希望相反的情况。不论如何，冒充者有 50%的可能性被上面的协议抓住。

然而验证者 V 可能发现 1/2 的可信度并不足以让示证者 P 证明她知道同构。幸运的是，验证者 V 能够简单地要求示证者 P 现在生成H'并再次经受该协议的考验。如果其现在通过了，那么验证者 V 对示证者 P 的可信度可达到 3/4。如果这还不够好，那么她可以使用H''第三次经受协议的考验，并获得 7/8 的可信度，15/16 的可信度，31/32 的可信度，等等。通过重复该协议，示证者 P 可以证明：他知道验证者 V 所要求的任意可信度的同构(但总是比 100%要小一些)。不管对该协议重复了多少次。验证者 V 都不会获得有助于构造他自己的 G/G'同构的知识，因此“零知识”泄露给了验证者 V。

4.3 基于 Gröbner 基的零知识证明

在实际工作开始的时候先要考虑的是如何选择策略，我们所采用的基于 Gröbner 基的零知识证明的思路如下：设 G 是$k[x_1,x_2,\cdots,x_n]$的有限子集，G 是 Gröbner 基，当且仅当对每个$f\in k[x_1,x_2,\cdots,x_n]$，$f$模 G 的范式是唯一的。基于此的零知识证明是很容易实现的，如果 Bob 知道某个 n 元多项式理想的基于给定单项式序的最简 Gröbner 基，则他可以声明他知道这个基于给定单项式序的最简 Gröbner 基，而 Alice 可能要他证明给自己看。因为 Alice 不知道单项式序，所以他无法计算出最简 Gröbner 基。如果穷举的话，难度将是指数级的。

Bob 证明给 Alice 看说他知道某个单项式序下的最简 Gröbner 基的证明过程(协议)如下：

（1）Bob 随机取出多项式环空间的一个多项式f;

（2）Bob 将多项式f发送给 Alice;

（3）Alice 要求 Bob 证明f是多项式理想空间下的多项式;

（4）Bob 根据多项式环下的最简 Gröbner 基的性质，f模此 Gröbner 基的范式是唯一的，而且如果$f\in H$，f模 G 的范式还为 0。由此 Bob 可以证明他知道这个 G 为 Gröbner 基。

由以上协议知，因为知道这个多项式理想的最简 Gröbner 基，如果不知道，

要证明某个随机任意的 $f \in H$ 是困难的，而且从概率上讲，如果每一个选取的 $f \in H$，他都能证明(即可以模 G 的范式唯一且为 0)的话，我们有理由相信 Bob 知道基于给定单项式序的最简 Gröbner 基。

由此 Bob 向 Alice 证明了他知道基于某个单项式序下的最简 Gröbner 基，而并没有告诉 Alice 具体的基。即实现了“零知识”泄露。

4.4 盲签名与部分盲签名

数字签名是一项重要的计算机安全技术，它的基本作用是保证传送的信息不被篡改和伪造，并确认签名者的身份。盲签名是一种特殊的数字签名。1983 年，Chuam 首先提出了盲签名[参考文献 56]的概念。之后，对盲签名的研究不断深入，强盲签名、部分盲签名[参考文献 24]的概念陆续被提出。

盲签名是指签名者并不知道所签文件或消息的具体内容，而文件或消息的拥有者又可以签名得到签名人关于真实文件或消息的签名。Chaum 曾给出了关于盲签名更直观的说明：所谓盲签名，就是先将要隐蔽的文件放入信封，再将一张复写纸也放入信封，签名的过程就是签名者将名字签在信封上，他的签名便透过复写纸签到了文件上。基于盲签名的特点，盲签名技术在电子货币、电子投票、电子支付等应用中的匿名性方面起着重要作用，国内外学者就此进行了许多相关研究并取得了一定成果。

盲数字签名方案具有的特性：消息的内容对签名者是盲的。

盲签名的简单实现过程如下：由 Alice 发送盲消息 m'给 Bob，由 Bob 对盲消息 m'进行签名，并发送（m'，sign（m'））给 Alice，Alice 利用（m'，sign（m'））推得对消息 m 的签名 sign（m），得到（m，sign（m））。当 Bob 看到（m，sign（m））时可以验证它是自己对盲消息 m'的有效签名。

盲签名可以保护签名接受者的隐私（如身份，签名内容等），并使得签名者无法追踪自己的签名。正是这种不可追踪性使得盲签名可以用于电子现金系统和匿名的电子选举。然而，为了确保电子支付的安全性和可控性，要求电子支付是可审核的，这与盲签名的不可追踪性相矛盾。理想的盲签名系统将致力于解决这两方面的问题。1996 年 Abe 等人首次提出了部分盲签名[参考文献 24]的概念，其克服了完全盲签名的缺点。

部分盲签名方案的基本思想如下：

部分盲签名方案可以看作一个集合 {x，f(x)，c，S（），V（），B（），U（）}。其中，x 和 f (x)分别是签名者的私钥和公钥。c 是签名者将在不泄露给发送者的前提下加入签名中的信息；S（x，c，m）是签名者用私钥 x 对信息 m 的签名；V（）是签名验证函数，它使得 {f（x），m，S（x，c，m）} 满足 V（f（x），m，

S（x，c，m））；B（）是致盲函数，它使得B（m，r）与消息m 及致盲因子r 统计无关；U（）是脱盲函数，它使得U（S，r'）是脱盲后用户取得的最终签名，而且在脱盲因子r' 不泄露的前提下U（S，r'）与S 统计无关。

部分盲签名的基本协议如下：

（1）发送方将致盲后的信息B（m，r）发送给签名方。

（2）签名方用其私钥对信息进行签名，然后将签名S（x，c，B（m，r））发送给发送方。

（3）发送方检查签名是否满足验证函数 V（），接着对签名进行脱盲，即计算U（S（x，c，B（m，r）），r'），从而计算得S（x，c，m）。然后可以将签名和被签名信息m 发送给签名方。

（4）签名方可以检查 S（x，c，m）和 m 是否满足验证函数 V（），但无法获取任何有关用户的身份c 的信息。

4.5 实 例

本节我们将并行的基于Gröbner基的零知识证明应用于电子商务模型之中，作为身份认证的方式进行讨论，提出了一种带有并行的零知识身份认证系统以及部分盲签名技术的电子支付安全模型，并对其安全性能进行了系统的分析。

电子支付是指单位、个人通过电子终端，直接或间接地向银行业金融机构发出支付指令，实现货币支付与资金转移。电子支付是指单位、个人通过电子终端，直接或间接向银行业金融机构发出支付指令，实现货币支付与资金转移。电子支付的业务类型按电子支付指令发起方式分为网上支付、电话支付、移动支付、销售点终端交易、自动柜员机交易和其他电子支付。而在电子商务支付系统当中电子现金是我们最常选取的支付方式。电子现金（E-cash）是一种非常重要的电子支付系统，它可以被看作是现实货币的电子或数字模拟，电子现金以数字信息形式存在，通过互联网流通。但比现实货币更加方便、经济。它最简单的形式包括三个主体：商家，用户，银行和四个安全协议过程：初始化协议，提款协议，支付协议，存款协议。

电子现金在其生命周期中要经过提取、支付和存款3个过程，涉及用户、商家和银行等三方。用户与银行执行提取协议从银行提取电子现金；用户与商家执行支付协议支付电子现金；商家与银行执行存款协议，将交易所得的电子现金存入银行。图1所示即为电子支付的一般模型。

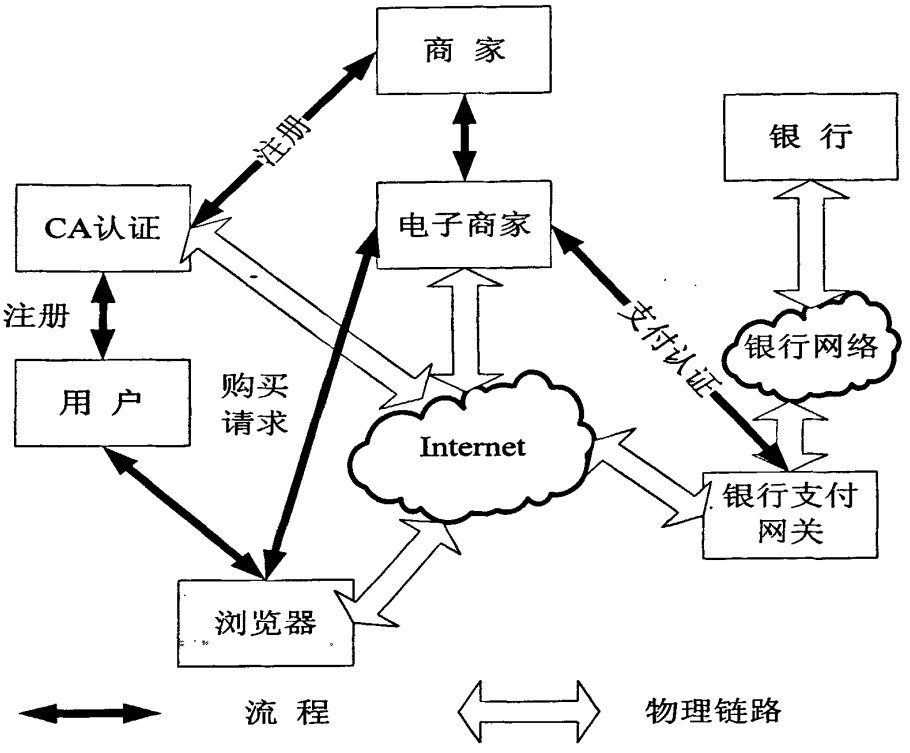

图 1 电子支付模型

由传统电子支付模型可以发现，在电子现金支付的过程中用户与银行之间的信息传递是最为重要的一环。用户在进行电子交易之前必须与自己的代理银行之间通过注册建立一种安全协议，以便于确认在之后的电子交易中用户的真实身份。因此，电子支付需要一个安全的系统来确认用户身份并保障交易是用户本人完成。

由此本文中提出了一种采用基于Gröbner基的零知识证明的身份验证系统，它将大大提高这一环节的安全性能。

基本的基于Gröbner基的零知识证明的相关内容，我们在前面的章节中已经作过讨论，此处不再重复，基本的零知识证明协议包括示证者P和验证者V之间的n次交换，可以把它们全部转换为并行完成：

(1) 示证者P使用他的信息和n个随机数把一个难题变成n个不同的同构难题，然后用它的信息和随机数解决这n个新难题。

(2) 示证者P提交这n个新难题的解法。

(3) 示证者P向验证者V透露n个新难题。验证者V无法利用这些新难题得到关于原问题或其解法的任何信息。

(4) 对这n个新难题中的每一个，验证者V要求示证者P：a) 向他证明新旧难题是同构的，或：b) 公开她在第(2)步中提交的解法，并证明它是这个新难题的解。

(5) 示证者P对这n个新难题中的每一个都表示同意。

基于并行零知识证明的身份验证系统包括身份证明信息(如：口令)、随机数和N个不同的同构难题。这种身份验证系统模式不是直接输入口令让系统鉴别，而是用随机数生成N个不同的同构难题，知道口令就可以给出这N个难题的正确解从而通过系统验证。不知道口令则不可能给出N个难题全部正确的解，从而无法通过系统的验证。同时，不知道口令者既使知道N个同构难题及其解，也无法得到口令信息。根据并行零知识证明可以设计出满足上述要求的身份验证系统。注册用户验证过程如图2所示。

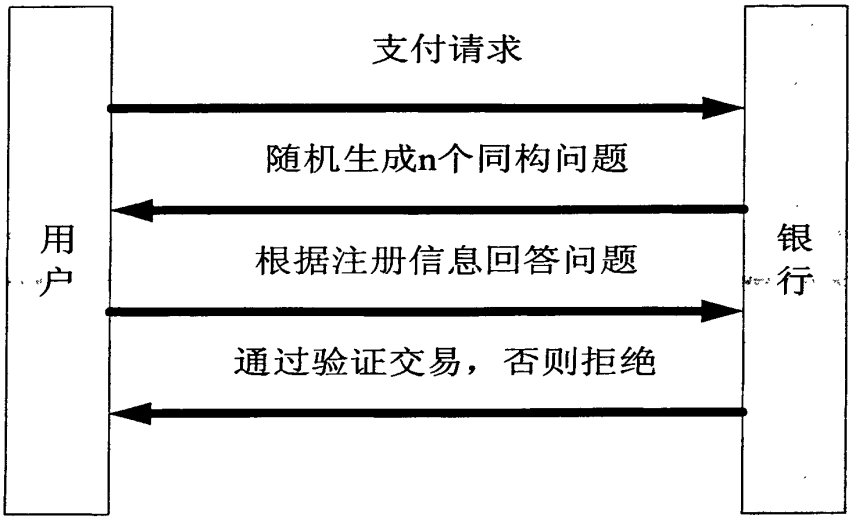

图 2 系统验证过程示意图

部分盲签名方案是由Abe和Fujisaki[参考文献24]提出的。它允许签名者添加一些接收者同意的限制性条款。这样，可以对接收者有所限制，在签名者不知所签署的消息具体内容的情况下，有效地保护签名者的合法权益。

一个部分盲签名方案有三个参与者：签名者、签名接收者、验证者。它有三种算法：密钥生成算法、部分盲签名问题算法、验证算法。

(1) 密钥生成算法是一个概率多项式时间算法，只需要输入一个安全参数，它就输出一个公钥、私钥对。

(2) 部分盲签名问题是一个签名者和接收者之间的交互协议。接收者的公开输入包括签名者与他约定的公开信息。签名者的公开输入包括公开信息和他的公钥。签名者的私人输入包括他的私钥。签名接收者的私人输入包括待签名的消息。当协议中止时，接收者的公开输入显示“完成”或者“未完”，接收者的私人信息输出显示“失败”或输出签名者对消息的签名信息。

(3) 验证算法也是一个多项式时间算法，输入签名者的公钥、公开信息和签名信息，输出“接受”或“拒绝”。

随着电子商务的迅速发展，人们越发热衷于网上购物，即用户通过注册代理

银行的网站，然后在网上以电子现金的方式与供货商家进行交易。由于这种交易方式是建立在开放的Internet之上的，所以在电子支付的过程中系统地安全性能成为人们关注的问题。以下提出带有基于Gröbner基的零知识证明身份认证系统以及部分盲签名技术的电子支付模型，并对其安全性进行分析。

该模型是针对Web系统网上电子交易提出的，由两部分组成：并行的基于Gröbner基的零知识证明身份认证系统以及部分盲签名的电子现金方案。

（1）初始化协议

首先，在银行Web系统中注册用户。注册过程分为两部分完成：

①身份认证。该认证需要用户向银行系统提供一种确认用户身份的证明，如电子版身份证、驾照、护照、有效证件等等。银行系统将会按照此证明核查用户身份并判断是否可以注册。

②注册。银行选择大素数p，q，满足 $q \mid p-1$，$g \in Z_p^*$ 且它的阶为q；银行的私钥为 $x \in Z_p^*$，相应的公钥为 $y=g^x \bmod p$；H() 是输出在 Z_q 上的公开单向函数。

（2）提款协议

用户想在开通电子支付服务的商家网站中购买商品，商家通过网站建立用户订单并随机选择 $k \in Z_q$，将它通过秘密信道传送给用户，然后用户向银行提出支付请求，并用 $z \in Z_q$ 代表取款金额。银行收到请求后随机生成n个同构问题传给用户，用户则按照注册信息回答问题并将答案反馈给银行，当银行确认n个问题全部正确后通过用户的身份验证。以上验证过程采用基于Gröbner基的零知识证明来实现，基本思想在前面的章节中已有介绍，此处不再重复。在银行确认用户的身份之后，用户与银行执行以下协议：

①银行选取随机数 $u,s,d \in Z_p^*$，计算 $a=g^u \bmod p$，$b=g^s z^d \bmod p$，再将a，b发送给用户；

②用户选取随机数 $t_1,t_2,t_3,t_4 \in Z_p^*$，计算 $\eta=g^k \bmod p$，$\alpha=ag^{t_1}y^{t_2} \bmod p$，$\beta=bg^{t_3}y^{t_4} \bmod p$，$\varepsilon=H(\alpha \| \beta \| z \| \eta)$，$e=\varepsilon-t_2-t_4 \bmod q$ 最后将e发送给银行；

③银行计算 $c=e-d \bmod q$，$r=r-cx \bmod q$ 再将签名 (r,c,s,d) 发送给用户；

④用户收到签名 (r,c,s,d) 后，计算 $\theta=r+t_1 \bmod q$，$\omega=c+t_2 \bmod q$，$\delta=s+t_3 \bmod q$，$\mu=d+t_4 \bmod q$，随后验证 $\omega+\mu=H(g^\theta y^\omega \| g^\delta y^\mu \| z \| \eta)$ 是否成立。成立则将 $\{z,\theta,\omega,\delta,\mu,\eta\}$ 作为电子现金。

（3）支付协议

①用户提供给商家电子现金 $\{z,\theta,\omega,\delta,\mu,\eta\}$，并用零知识向商家证明他知道k；

②商家验证 $\omega+\mu=H(g^\theta y^\omega \| g^\delta y^\mu \| z \| \eta)$ 是否成立，来确定该电子现金的有效性。同时由商家自己选取的随机数k可以保证商家知道此电子现金是否花费过。

（4）存款协议

商家把 $\{z,\theta,\omega,\delta,\mu,\eta\}$ 传输给银行，银行检测电子现金的合法性，如果该现金合

法则在商家的帐户上添加电子现金中的面额，完成整个电子现金支付过程。

下面我们将简要地分析一下该模型的特点。上述的电子支付模型在身份验证与电子现金交易的过程中具有较高的安全性能。

首先，基于零知识的身份验证系统可以保障用户能够重复在别人面前登录银行系统，而不必担心口令泄露。因为同构的难题及其解并未给出关于用户口令的准确信息故增加了口令破解的难度。虽然，这样会增加用户进入系统的复杂性，但是采用并行基于Gröbner基的零知识证明方式可以降低计算复杂度，因而这种代价是值得的。

其次，电子交易的过程运用了部分盲签名的技术，是基于Abe—Okamoto部分盲签名方案设计的一种离线电子现金方案。它在支付时不要求银行同时在线，这是在线电子现金不具备的优点。

下面对此方案的性质和功能进行简要分析：

(1)它满足盲签名的四条性质。不可伪造性、不可抵赖性、盲性和不可跟踪性，同时可以有效地保持签署消息的盲性；

(2)这种部分盲签名方案是基于Okamoto—Schnorr盲签名提出的，签名的长度较短，实现速度相对较快；

(3)因为签名中加入了限制性条款即共识信息，它可以对签名者和接受者的权利和义务以及签署消息的性质加以说明。而且共识信息不容修改，一旦改动签名将不会成立。

该电子现金是不可分的，因为采用部分盲签名，所取的电子现金面额可以在现金数据中标明，因而比用普通盲签名构造的方案具有更大的灵活性。

我们所介绍的这种并行的基于Gröbner基的零知识证明身份验证系统和部分盲签名的电子现金支付模型，它利用了已有的零知识证明系统和盲签名方案来构筑电子现金，采用并行Gröbner基实现来提高效率，总体来说对拓宽研究电子现金的视野具有一定的价值。

4.6 小 结

随着社会的信息化步伐的加快及电子商务、电子政务等的发展，人们对信息安全、信息认证投入了极大的关注。要求在信息系统中建立一套完善的系统安全解决方案。现有的传统的信息安全机制在随着硬件性能的大步提升后，使得以前认为安全的一些加密解密算法在目前看来越来越有被破译的可能，因此产生的代数曲线加密思想。该思想是基于代数学的交换群思想的椭圆曲线加密解密算法及建立在其上的信息安全机制正得到更多的认可和应用。在此基础之上本章利用前面章节中的 Gröbner 基并行化知识，对基于 Gröbner 基的信息安全机制中的一种

----零知识证明作了简单的探讨，并在此基础上提出基于 Gröbner 基的并行的零知识身份认证与部分盲签名技术相结合的电子支付安全模型。

因为所有的加密解密算法都必须建立在一定的数学难题及其一些特殊性质之上，所以我们在这里是基于单项式序的排列组合数问题和多项式计算难度问题。在计算复杂度上还有待提高，而且所基于的多项式问题在理论上还不是不可破解的，因而我所做的过程中主要仍然是在有限域上进行的。所以我考虑在以后的工作中继续探索和寻找，争取能有所突破。

第五章 结束语

Gröbner 基，在某些文章中也称之为 standard 基，它具有“唯一性”的良好性质，利用 Gröbner 基，理想成员的判断及许多问题都可得到解决。Gröbner 基方法（理论和算法）能够提供一种标准的方法，这种方法能很好的解决可以被表示成由多变元多项式集合中项的形式所构成的很多问题，这些问题包括：代数几何、交换代数和多项式理想理论，非变量理论，自动化几何定理证明，编码理论，整数程序设计，偏微分方程，超几何函数，统计学，非交换代数系统理论等。在交换代数中 Gröbner 基方法可以解决以下问题：代数系统方程的可解性判断与解答，理想元素与根元素判决，对多项式理想取模所生成的剩余类环上的有效计算，Hilbert 方程，多项式间的代数关系，多项式映射的逆映射和由多项式系数（”syzygies”）构成的线性 Diophantine 方程等。

随着Gröbner 基求解算法的不断改进，该方法的应用领域越来越广，同时随着Gröbner 基求解算法效率的不断提高，其性能也逐渐能够满足人们的对求解 Gröbner 基求解要求。因而对Gröbner 基方法（理论和算法）的研究有着非常广阔的前景。在本文中我们所研究的主要内容仅仅是对算法的并行与约化，还有更多的复杂而又有意义的工作等待着我们去进一步完成。

5.1 小 结

本文中我们所研究的主要内容就是算法的并行相关问题及影响算法效率的关键，即中间项的约化，这是使 Gröbner 基更有效和更实用的一个重要条件。本文由当前的主流算法出发，先对求解算法进行分析，进而介绍算法的并行及对影响算法效率的中间项的约化，在此过程中首先采用结构化高斯消元法对可能产生大型稀疏矩阵予以减化，进而主要采用并行高斯全选主元消去法进行约化，以达到提高算法效率的目的，最后我们将 Gröbner 基方法应用于零知识证明方式的身份认证，提出以采用并行的基于 Gröbner 基的零知识证明与部分盲签名相结合的方式进行安全电子支付的模型。

现有的能够进行求解多变元多项式的数学软件有很多了，我们在进行算法分析和并行的时候，对一些现有的软件程序进行了参考和分析。需要指出，当计算量达到一定程度时，由于内存大小的限制，当多项式数多到一定程度时，其实现输出到文件、实现文件处理方、实现文件的共享读取等功能还不是很完备。在一般的 PC 机上很难以运行 Mathematica 里的 Gröbner 基软件包对高阶幂的多项式组

的求解，由此可见我们的工作还仅仅只是一个开头，更多更有价值的工作正等待着我们去完成。

5.2 发展趋势

Gröbner 基这种方法主要用于可由计算机算法实现的方式来解决的一些交换代数中的基本问题，如多项式理想理论，代数几何等，现在在信息安全领域广泛受到重视。同时随着研究的深入，这种可编程实现的方法将被进一步广泛适用于各个的领域，如：多变元多项式矩阵的因式分解，离散系统的柯西问题解决，最小左（右）零化子计算，单双边多项式矩阵方程，Bezout恒等式等的可行性测试与解决方法构建，可观测性测试，FIR / IIR多维滤波器组的设计等等。

Gröbner 基方法就是这样一种能够快速的求解上述问题的方法，同时当算法的实现效率随着算法自身的改进，计算机软件与硬件的扩展带来的计算速率的提升而进一步提高之后，其在工程领域方面使用的前景不可估量。

致 谢

首先深深感谢我的导师马文平教授！在我攻读硕士学位期间马老师为我创造了良好的学习环境和学习氛围，在学习和生活上始终得到马老师的亲切关怀和悉心的指导。马老师渊博的知识、一丝不苟的工作作风以及敏锐的洞察力都给我的学习和研究以莫大的帮助和启发。马老师提倡学术自由，善于引导学生进行创造性思维，鼓励学生发表个人见解。他严谨的学风、豁达的胸襟和诲人不倦的精神令我终身难忘。正是马老师使我懂得了什么样的人是真正的科研工作者。马老师以正直无私的品格、豁达的胸襟和哲人的风度为我树立了榜样。再次感谢马老师对我谆谆的教导和孜孜不倦的言传身教，这一切都将让我终生受益！

在此特别感谢张鸿燕老师！衷心的感谢他给予我学习上的指导和生活上的关心！他所给予我的东西是我要学习和领会一生的，从他身上我看到了作为科研工作者所必须的气质和内涵，他以正直无私的品格为我树立了一个学习的榜样。感谢像他这样一位哥哥的无私关心与真诚帮助，让我找回了许多自己已经丢失已久的东西。

衷心的感谢刘景伟老师、刘振华老师、史耀媛老师、陈兴老师等所给予我学习上的指导和生活上的关心与帮助！

感谢实验室的何叶锋博士、杨元原博士、杨晨博士、闫琪博士、余旺科博士、白晓峰硕士、江伟湘硕士、陈薇硕士、苟杰斌硕士、李靖岚硕士、曹凯华硕士、张荣硕士等所有的师兄师姐师弟师妹们，感谢王磊硕士、王国争硕士、范乐伟硕士、李强硕士等同学朋友的关怀与帮助，感谢他(她)们在学习上对我的帮助、在生活上给我带来的欢乐！

深深感谢我的父母对我的理解与支持，他们对我无私的关爱和付出为我营造出一个良好的生活和学习环境！

攻读硕士的两年半是我人生中的一个重要阶段，其中有成功与喜悦也有挫折与失败，我将永远怀念这段人生历程。

对于所有关心和帮助过我的人，我无以为报，只希望自己在毕业后更加努力地学习和工作不辜负他们对我的期望。

参考文献

[1]Adams W W, Loustaunau P, An Introduction to Gröbner Bases, American Mathematical Society. 1994.

[2]Atiyah M F, Macdonald I G., Introduction to Commutative Algebra, - Addison-Wesley Publishing Company, 1969.

[3]Faugère, J.-C., A new efficient algorithm for computing Gröbner bases without reduction to zero (F5), in: ISSAC '02: Proceedings from the International Symposium on Symbolic and Algebraic Computation (2002), pp. 75–83, revised version 1.2 available at the URL below. http://www-calfor.lip6.fr/»jcf/Papers/@papers/f5.pdf

[4]Faugère, J.-C., A new efficient algorithm for computing Gröbner bases (F4), Journal of Pure and Applied Algebra 139 (1999), pp. 61–88.

[5]Faugère, J.-C., A new efficient algorithm for computing Gröbner bases without reduction to zero (F5), ISSAC 2002, July 7-10, 2002, pp.75-83, Lille, France.

[6]Katsusuke Nabeshima, A Direct Products of Fields Approach to Comprehensive Gröbner Bases over Finite Fields, Proceedings of the Seventh International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC'05).

[7]Iyad A. Ajwa, A case study of Grid Computing and computer algebra: parallel Gröbner Bases and Characteristic Sets, The Journal of Supercomputing, v.41 n.1, p.53-62, July 2007.

[8]Iyad A. Ajwa , Paul S. Wang, Parallel algorithms and implementations for the grobner bases algorithm and the characteristic sets method, Kent State University, Kent, OH, 1998.

[9]何青，计算代数。北京：北京师范大学出版社，1997。

[10]刘木兰，Gröbner 基础论及其应用。北京：科学出版社, 2000。

[11]Iyad A. Ajwa, Zhuojun Liu, Paul S. Wang. Gröbner Bases Algorithm, ICM Technical Reports Series (ICM-199502-00), February 1995.

[12]Bruno Buchberger, Gröbner Bases:A Short Introduction for Systems Theorists, R.Moreno-Diaz et al. (Eds.): EUROCAST 2001, LNCS 2178, pp. 1-19, 2001.

[13]Dongming Wang, On the Parallelization of Characteristic-Set-Based Algorithms, Proceedings of the First International ACPC Conference on Parallel Computation, pp.338-349, September 30-October 02, 1991.

[14]T. Becker and V. Weispfenning, Gröbner Bases, Springer-Verlag, New York, 1993.

[15]Marc Moreno Maza , Yuzhen Xie, An implementation report for parallel triangular decompositions, Proceedings of the eighteenth annual ACM symposium on Parallelism in algorithms and architectures, July 30-August 02, 2006, Cambridge, Massachusetts, USA.

[16]Bruno Buchberger, The Parallel L-Machine for Symbolic Computation, Research Contributions from the European Conference on Computer Algebra-Volume 2, pp.541-542, April 01-03, 1985.

[17]Caboara, M., M. Kreuzer and L. Robbiano, Efficiently computing minimal sets of critical pairs, Journal of Symbolic Computation 38 (2004), pp. 1169–1190.

[18]Ars, G., J.-C. Faugère, H. Imai, M. Kawazoe and M. Sugita, Comparison between XL and Gröbner basis algorithms, in: P. J. Lee, editor, ASIACRYPT 2004, Lecture Notes in Computer Science 3329 (2004), pp. 338–353.

[19]Bardet, M., J.-C. Faugère and B. Salvy, Complexity of Gröbner basis computation for semi-regular overdetermined sequences over F2 with solutions in F2, rapport de recherche 5049, Institut National de Recherche en Informatique et en Automatique, Lorraine (2003).

[20]Winfried Just and Brandilyn Stigler, Computing Gröbner Bases of Ideals of Few Points in High Dimensions, ACM Communications in Computer Algebra, Vol 40, No. 3, Sep. 2006.

[21]Weiwei Lin, Changgeng Guo, Deyu Qi, Yuehong Chen, and Zhang Zhili, Implementations of Grid-Based Distributed Parallel Computing, Proceedings of the First International Multi-Symposiums on Computer and Computational Sciences (IMSCCS'06).

[22]Ning Lu, Z. Todd Taylor, Dave P. Chassin, Ross Guttromson, and Scott Studham, Parallel Computing Environments and Methods for Power Distribution System Simulation. http://arxiv.org/ftp/cs/papers/0409/0409035.pdf

[23]Eugenio Roanes-Lozano, Eugenio Roanes-Macías, and Luis M. Laita. Some Applications of Gröbner Basis, May/June 2004, Computing in Science & Engineering, pp. 56-60.

[24]Abe M, Fujisaki E, How to Date Blind Signatures[A], Advances in Cryptology-Asiacrypt'96 Proceedings[C], Berlin:Springer-Verlag,1996, pp. 244-251.

[25]Meng Zhou and Franz Winkler, Gröbner Basis in Difference-Differential Modules, ISSAC'06, July 9–12, 2006, Genova, Italy.

[26]Till Stegers, Faugère's F5 Algorithm Revisited, Thesis For The Degree Of Diplom-Mathematiker, September 2005.

[27]Jean-Charles Faugère and Antoine Joux, Algebraic Cryptanalysis of Hidden Field Equation (HFE) Cryptosystems Using Gröbner Bases, D. Boneh (Ed.): CRYPTO 2003, LNCS 2729, pp. 44 - 60, 2003.

[28]李源顺, 浅谈多元多次方程组, 数学传播, 27 卷 1 期, 2003 年 3 月, pp. 57-67。

[29]H. Michael Möller, Teo Mora, Carlo Traverso, Gröbner bases computation using syzygies, ISSAC 92, July 1992 , Berkeley, California, United States pp. 320 – 328.

[30]都志辉, 高性能计算并行编程技术一一 MPI 并行程序设计, 北京：清华大学出版社, 2001。

[31]Barry Wilkinson and Michael Allen 著, 陆鑫达等译, 并行程序设计, 北京：机械工业出版社, 2002。

[32]陈国良, 并行算法的设计与分析（修订版）。北京: 高等教育出版社，2002。

[33]J. Dongarra, I. Foster, G. Fox, W. Gropp, K. Kennedy, L. Torczon, and A. White 编著, 莫则尧, 陈军, 曹小林等译，李晓梅审校，并行计算综论，北京：电子工业出版社, 2005。

[34]陆佩忠，Gröbner 基与环上线性递归阵列，北京：高等教育出版社, 2002。

[35]A. Montes, A new algorithm for discussing Gröbner basis with parameters, Journal of Symbolic Computation, 33/1-2:183–208, 2002.

[36]K. Nabeshima, A Computation Method for ACGB-V, In A. Dolzmann, A. Seidl, and T. Sturm, editors, Conference in Honor of the 60th Birthday of Volker Weispfenning, pp. 173–180. BOD Norderstedt, 2005.

[37]Attardi G. and Traverso C., Homogeneous Parallel Algorithm, Journal of Symbolic Computation, 1996.

[38]陈国良, 并行计算一一结构算法编程。北京：高等教育出版社，1999。

[39]何元清, 孙世新, 傅彦，并行编程模式及分析, 第 31 卷, 第 2 期, 电子科技大学学报, 2004, pp173-175。

[40]孙济洲, 樊莉亚, 孙敏, 于策, 张绍敏, 改进的并行高斯全主元消去法，第 39 卷, 第 9 期, 天津大学学报, 2006 年 9 月, pp.1115-1119。

[41]Quinn M J. Parallel Programming in C with MPI and OpenMP[M].Beijing: Tsinghua University Press, 2005.

[42]万哲先, 代数与编码，北京: 科学出版社.1981。

[43]McGinn S F, Shaw R E. Paralel Gaussian elimination using openMP and MPI[C] //Proceedings of the 16th Annual International Symposium on High Performance

Computing Systems and Applications. Moncton, NB, Canada, 2002:169-174.

[44]张禾瑞，近世代数基础。北京：高等教育出版社，1978 年修订版。

[45]柴凤娟, 快速 Gröbner Basis 算法的并行实现, 硕士学位论文, 2005。

[46]彭丰富，Gröbner 基与约化的研究及其应用，硕士学位论文，2001。

[47]杜瑞昌，关于 Groebner 基算法复杂性及其应用的若干问题，硕士学位论文，2001。

[48]张普朝，基于 Gröbner Bases 的信息安全技术，硕士学位论文，2002。

[49]张博，试验设计中的 Gröbner 基方法，硕士学位论文，2004。

[50]彭丰富，陈小松，Gröbner 基优化算法，武汉科技大学学报(自然科学版)，2003 年 9 月，第 26 卷第 3 期，pp.320-322。

[51]韩然，周梦，线性变换下 Gröbner 基的转换问题，北京电子科技学院学报，2003 年 6 月，第 11 卷第 1 期，pp.1-7。

[52]王平水，基于独立集问题的零知识证明研究，计算机技术与发展，2007 年 9 月，第 17 卷第 9 期，pp.55-57。

[53]张跃宇，陈杰，苏万力，王育民，一种 IND-CCA2 完全匿名的短群签名，计算机学报，第 30 卷第 10 期，2007 年 10 月，pp.1865-1871。

[54]冯修玉，施荣华，彭艳，一种零知识证明的群签名方案，计算机工程与应用，2005.33，pp. 122-123。

[55]张虎强，洪佩琳，李津生， 熊继平，一种零知识证明协议的安全分析与改进，信息安全与通信保密，2006.11，pp.163-166。

[56]Chaum D., Blind Signature for Untraceable payment[C], Proc. Of Advance in Cryptology-Crypto'82,1983:199-203.

[57]许静，冯伟成，周莲英，孙晓明，基于部分盲签名的新型电子现金安全系统研究，计算机工程，2006 年 10 月，第 32 卷第 19 期，pp.157-158。

[58] Zhang F, Kim K., ID-based Blind Signature and Ring Signature fromPai rings[C], Pr oc.of th eA siacryptL NCS,20 02:53 3-547.

[59]辛向军，李发根，肖国镇，对几种部分盲签名方案的安全性分析与改进，西安电子科技大学学报（自然科学版），2006 年 12 月，第 33 卷第 6 期，pp.953-955。

[60]田秀健，曹珍富，基于身份的可认证的盲签名方案，计算机工程，2006 年 7 月，第 32 卷第 14 期，pp.136-137。

[61]张彤，王育民，几种部分盲签名的算法设计及其安全性分析，西安电子科技大学学报（自然科学版），2004 年 12 月第 31 卷第 6 期，pp963-966。

[62]Oded Goldreich，密码学基础（第一、二卷）。北京：电子工业出版社，2005。

[63] Robert Sedgewick, 算法 I-IV（C 实现）——基础、数据结构、排序和搜索（第三版·影印版），北京：中国电力出版社, 2003。

[64]王育民，刘建伟，通信网的安全——理论与技术。西安：西安电子科技大学出版社，1999。

[65]叶琳，韩建，洪志全,盲签名机制的性能分析,信息技术，2006 年第 10 期，pp.16-19。

[66]http://www.symbolicdata.org

[67]http://fgbrs.lip6.fr/jcf/Benchs/

[68]http://www.math.msu.edu/»jan/Demo/TIMINGS.html

[69]http://www.cdc.informatik.tu-darmstadt.de/»stegers/

[70]http://www.singular.uni-kl.de/

[71]http://www.cs.hku.hk/cluster2003/tutorials.htm#infiniband

[72]www.rocksclusters.org

[73]http://www-unix.mcs.anl.gov/mpi/mpich

[74]http://Iiinwww.ira.uka.de/courses/prakt/mpi/mpi2-html-doc/node306.html

[75]http://www.rnpi.nd.edu/lam/download/

附 录

一般可查到的与 Gröbner 基相关的开放源代码主要有：

Macaulay。Macaulay 计算机代数系统对于多项式计算非常有用，并重点强调 Gröbner 基计算。它旨在解决具有简单语法并且已描述为代数机器语言 (algebraic machine language) 的问题。

Magma。Magma 在成本回收许可证下进行分发，是一个旨在解决代数问题的高性能系统。它突出体现了用于群论的功能以及群数据库、用于整数和多项式算术的渐近快速算法和几个用于高级运算的前沿库。

Mathomatic。此程序没有内置的编程功能，旨在用作简单的符号数学计算器。它可以在任何系统上使用 C 编译器、标准 C 库和 UNIX make 实用程序进行编译。

我们参考：Mathematica5.0 中的 Algebraic Operations on Polynomials 一节内容，以下是参考到的相关指令的简要内容，具体内容可登陆 Mathematica 主页或参考 Mathematica 全书查看。

GroebnerBasis[{ $poly_1, poly_2, \cdots$}, { $x_1, x_2, \cdots$}]

gives a list of polynomials that form a Gröbner basis for the set of polynomials $poly_i$.

GroebnerBasis[{ $poly_1, poly_2, \cdots$}, { $x_1, x_2, \cdots$}, { $y_1, y_2, \cdots$}]

finds a Gröbner basis in which the y_i have been eliminated.

PolynomialReduce[poly, { $poly_1, poly_2, \cdots$}, { $x_1, x_2, \cdots$}]

finds a minimal representation of $poly$ in terms of the $poly_i$.

The set of polynomials in a Gröbner basis have the same collection of roots as the original polynomials.

For polynomials in one variable, GroebnerBasis reduces to Polynomial GCD.

For linear functions in any number of variables, GroebnerBasis is equivalent to Gaussian elimination.

The Gröbner basis in general depends on the ordering assigned to monomials. This ordering is affected by the ordering of the x_i.

The following options can be given:

MonomialOrder	Lexicographic	the criterion used for ordering monomials
CoefficientDomain	Automatic	the type of objects assumed to be coefficients
Modulus	0	The modulus for numerical coefficients

Possible settings for MonomialOrder are Lexicographic, DegreeLexicographic, DegreeReverseLexicographic or an explicit weight matrix. Monomials are specified for the

purpose of MonomialOrder by lists of the exponents with which the x_i appear in them.

The ordering of the x_i and the setting for MonomialOrder can substantially affect the efficiency of GroebnerBasis.

Possible settings for CoefficientDomain are InexactNumbers, Rationals, RationalFunctions and Polynomials[x].

.....

For systems of polynomial equations, Solve constructs a Gröbner basis.

Solve and GroebnerBasis use an efficient version of the Buchberger algorithm.

.....

另外一种数学软件Singular 3.0的实现，其命令为slimgb，所采用算法是F4算法的一个变种，可参见：Greuel GM, Pfister G, Schonemann H. Singular 3.0. Centre for Computer Algebra A Computer Algebra System for Polynomial Computations: University of Kaiserslautern, 2001. http://www.singular.uni-kl.de/

作者读研期间发表的论文和参与的科研项目

论文发表：

狄鹏，何业锋，杨元源。MP2P 中一种简单安全身份认证，西安邮电学院学报(自然科学版)，2008 年 5 月第 13 卷第 3 期，pp9-12。

Grobner基生成算法的并行

作者： 狄鹏
学位授予单位： 西安电子科技大学