线性代数与多项式方程组求解

清华大学计算机系 计 14 班
吴育昕(2011011271) 李铁峥(2011011268)

目录

1 概述

解多项式方程组一直是代数学的中心问题之一. 几何与分析中的许多问题,当抽象出代数结构后,最后常常归结为多项式方程组的求解. 然而时至今日，这个问题的解决依然不尽人意. 对于简单的方程组，可以利用传统的换元，降次，消元的方法解决，但对于复杂的多项式方程组，现有的有效算法的理论复杂度都至少在双指数级别.

本文中,作者利用所学的线性代数知识,研究了多项式方程组求解中的结式方法, 同时通过阅读文献,了解并简要介绍了多项式方程组求解的抽象代数手段, 并举出了多项式方程组求解在计算机专业领域应用的几个例子.

2 结式方法

2.1 原理

消元是解方程组的基本原理. 对于解线性方程组,由于方程只有一次,可以直接将每个方程中的一变元用其余变元表出,再代入消元. 但对于多项式方程组,由于很多类型的高次方程无根式解,即使有也大多极其复杂,难以使用传统的代入消元手段进行消元, 而基于换元或方程相加减等的消元方法太过于依赖结构,不具有计算机处理所需的通用性. 因此需要引入其他的消元方法,结式(Resultant)就是一种.

首先考虑复数域上关于 x 的两个多项式方程构成的方程组

$$\begin{cases} f(x)=\sum_{i=0}^{n} a_i x^{n-i}=0 \\ g(x)=\sum_{i=0}^{m} b_i x^{m-i}=0 \end{cases}, \text{其中} a_0 b_0 \neq 0$$

这个方程组的解,实际上就是指两多项式的公共复根.

$f(x)=0, g(x)=0$ 有公共复根

$\Leftrightarrow \exists f_1(x), g_1(x), \deg f_1(x)<n, \deg g_1(x)<m, \dfrac{f(x)}{f_1(x)}=\dfrac{g(x)}{g_1(x)}$

$\Leftrightarrow \exists f_1(x), g_1(x), \deg f_1(x)<n, \deg g_1(x)<m, f(x)g_1(x)=g(x)f_1(x)$

于是设 $f_1(x)=\sum_{i=0}^{n-1} u_i x^{n-1-i}, g_1(x)=\sum_{i=0}^{m-1} v_i x^{m-1-i}$, 代入上式,比较左右两边所有 $m+n$ 个项的系数,可得 $m+n$ 个线性方程.

$$\begin{cases} a_0 v_0 & & & = & b_0 u_0 & & \\ a_1 v_0 + a_0 v_1 & & & = & b_1 u_0 + b_0 u_1 & & \\ \cdots\cdots & & \cdots & = & \cdots\cdots & & \cdots \\ & a_n v_{m-2} + a_{n-1} v_{m-1} & = & & & b_m u_{n-2} + b_{m-1} u_{n-1} & \\ & a_n v_{m-1} & = & & & b_m u_{n-1} & \end{cases}$$

这个关于 $(v_0, \cdots, v_{m-1}, u_0, \cdots, u_{n-1})$ 的 $m+n$ 元齐次线性方程组必须有非零解,于是其系数行列式

$$A=\begin{vmatrix} a_0 & & & & & & b_0 & & & \\ a_1 & a_0 & & & & & b_1 & b_0 & & \\ \cdots & \cdots & \cdots & \cdots & \cdots & & \cdots & \cdots & \cdots & \\ & & & & a_n & a_{n-1} & & & b_m & b_{m-1} \\ & & & & & a_n & & & & b_m \end{vmatrix}=0$$

也即其转置

$$A^T=\begin{vmatrix} a_0 & a_1 & \cdots & \cdots & \cdots & \cdots & a_n & & & \\ & a_0 & a_1 & \cdots & \cdots & \cdots & \cdots & a_n & & \\ & & \cdots & \cdots & \cdots & \cdots & \cdots & & & \\ & & & a_0 & a_1 & \cdots & \cdots & \cdots & & a_n \\ b_0 & b_1 & \cdots & b_{m-1} & b_m & & & & & \\ & b_0 & b_1 & \cdots & \cdots & b_m & & & & \\ & & \cdots & \cdots & \cdots & \cdots & \cdots & & & \\ & & & & & b_0 & b_1 & b_2 & \cdots & b_m \end{vmatrix}=0$$

注意到 A^T 仅由多项式 f, g 的系数决定,可记 $Res_x(f,g)=|A^T|$, 称其为多项式 f, g 的**Sylvester 结式(Sylvester's Resultant)** ,简称结式.

根据上述讨论,我们有结论:$f(x), g(x)$ 有重根 $\Leftrightarrow Res_x(f,g)=0$

再考虑二元多项式方程组:

$\begin{cases} f(x,y)=0 \\ g(x,y)=0 \end{cases}$,以 x 为主元,视 y 为常数,可求得结式 $P(y)=R_x(f,g)$,是 y 的多项式.

由之前对一元方程组的讨论可知,对 $P(y)$ 的每个零点 $y_0, f(x,y_0), g(x,y_0)$ 有公共复根. 反之,对于原方程组的每个解 (x_0,y_0), 必有 $P(y_0)=0$

于是,对原方程组的求解转化为对一元方程 $P(y)=0$ 的求解,成功实现了消元,降低了问题的难度.

对两个方程构成的含有多个变元的方程组,可以使用相同手段消去其中某个变元. 而对于超过两个方程构成的方程组,可以每次取其中两个方程计算结式,例如解如下方程组:

$$\begin{cases} f_1 = xy+z-5=0 \\ f_2 = x+y+z-6=0 \\ f_3 = x^2-2xy+y^2-2z=0 \end{cases}$$

$Res_x(f_1,f_2)=y^2+(z-6)y-z+5$
$Res_x(f_2,f_3)=4y^2+(4z-24)y+z^2-14z+36$
$Res_y(Res_x(f_1,f_2),Res_x(f_2,f_3))=(z-2)^2(z-8)^2$
于是消去了 y,z,得到 $z=2,z=8$, 代回原方程可求得完整解.
对于多个方程构成的方程组,也有多元结式理论可以用于解决消元问题.[2]

2.2 应用

对于一元方程组,结式方法的一个重要应用是判断多项式是否有重根. 考虑多项式 $f(x)=\prod_{i=1}^{k}(x-x_i)^{r_i}$, 显然对于每个 i,有 $r_i>1 \Leftrightarrow f'(x_i)=0 \Leftrightarrow x_i$ 是 $f(x)$ 与 $f'(x)$ 的公共复根. 所以, $f(x)=0$ 有重根 $\Leftrightarrow Res(f,f')=0$.

另外,上述的二元方程组的结式求解方法可应用于物理解题中常常能遇到的一类代数曲面/曲线隐式化问题. 例如,用参数方程表示的曲线 $\begin{cases} x=\dfrac{-t^2+2t}{t^2+1} \\ y=\dfrac{2t^2+2t}{t^2+1} \end{cases}$

将两式都写为关于 t 的多项式: $\begin{cases} f(t)=(x+1)t^2-2t+x \\ g(t)=(y-2)t^2-2t+y \end{cases}$

排除掉 $x+1=y-2=0$ 的情形,计算结式 $Res_t(f,g)=8x^2-4xy+5y^2+12x-12y$

于是 $8x^2-4xy+5y^2+12x-12y=0, (x,y)\neq(-1,2)$ 为曲线隐式化后的结果.

3 抽象代数方法简述

本节主要将抽象代数方法里的一个基本概念与线性代数进行类比.
为了求解线性方程组,我们首先学习了秩,线性相关等概念.

对于方程组 $\begin{cases} f_1(x_1,\cdots,x_n)=0 \\ f_2(x_1,\cdots,x_n)=0 \\ f_3(x_1,\cdots,x_n)=0 \\ \quad\vdots \end{cases}$

它是一个线性方程组时,若 $f_3=p_1f_1+p_2f_2, p_1,p_2\in\mathbb{C}$,那么在原方程组中,$f_3=0$ 是一个“没用”的方程. 反映在对应的系数矩阵中, 设 $\overrightarrow{b}_1$, $\overrightarrow{b}_2,\cdots$ 为行向量,则 $\overrightarrow{b}_3$ 可由 $\overrightarrow{b}_1$, $\overrightarrow{b}_2$ 线性表出,或称 $\overrightarrow{b}_3$ 在 $\overrightarrow{b}_1$, $\overrightarrow{b}_2$ 生成的线性空间中.

线性相关的概念描述了在线性方程组中哪些方程是“没用”的,秩的概念进一步给出了线性方程组中“有用”的方程个数. 搞清楚这一点,是系统化地解方程组的基本思路. 多项式方程组求解首先也需解决这一问题.

若将上述方程组视作一个多项式方程组,设 $K[x_1,\cdots,x_n]$ 为关于 $(x_1,\cdots,x_n)$ 的多项式集合. 则若有 $f_3=p_1f_1+p_2f_2, p_1,p_2\in K[x_1,\cdots,x_n]$, 那么 $f_3=0$ 是一个“没用”的方程. 于是,类比线性表出与线性相关的概念,我们很自然的给出了多项式“理想(Ideals)”的定义.

$f_1,\cdots,f_s$ 的生成理想定义为:

$$I=\langle f_1,\cdots,f_s\rangle=\{p_1f_1+\cdots+p_sf_s, p_i\in K[x_1,\cdots,x_n]\}.$$

于是,$f=0$ 是一个“没用”的方程当且仅当它在其余多项式的生成理想中. 因此,只需将方程组中全部多项式的生成理想用简单的形式表出,或者说,用一组性质较好的多项式生成,就容易将问题简化.

显然,理想具有一些基本的性质,例如其中的元素对于加法,数乘封闭.事实上,一个理想本身可作为一个线性空间. 在线性空间中我们通过找一组基来描述整个空间的性质,在理想中,可以通过 Grobner 基这种特殊的基,来描述整个理想.

Grobner 基是对理想的一种很好的表示,具有很多极好的性质,因此用于多项式方程组求解中的效率更高. 当前的许多符号计算软件,如 Maple ,Mathematica maTHμ 都使用 Grobner 基方法实现多项式方程组求解[1] [2].

以下为清华大学 maTHμ 协会自主开发的符号计算软件 maTHμ[3] 中 Grobner 基部分的代码接口:

res/multigroebner.cpp

```
namespace mU {
    namespace{
        order_q com_ptr;
        uint n;
        typedef std::pair<uint,uint> tuple2;
        typedef std::set<tuple2> tuple2set;

        bool groebner_less(const sparse_q & g, const sparse_q & h);

        void groebner_reduce(sparse_q & r, const sparse_q & f,
                const std::vector<sparse_q> polylist);

        void groebner_reduce(std::vector<sparse_q> & polylist, uint index);

        void mono_q_lcm(mono_q & r, const mono_q & f, const mono_q & g);

        void mono_q_div(mono_q & r, const mono_q & f, const mono_q & g);
```

[1] Maple 官方对 Solve 命令实现方式的介绍: http://www.maplesoft.com/support/help/Maple/view.aspx?path=Groebner%2fSolve

[2] Wolfram 官方对 Solve 命令实现方式的介绍: http://reference.wolfram.com/mathematica/tutorial/SomeNotesOnInternalImplementation.html#24488

[3] 主页http://mathmu.cn/

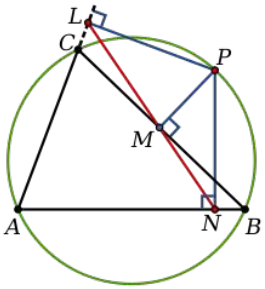
```
18
19          bool mono_q_divisible(const mono_q & f, const mono_q & g);
20
21          bool crit1(const mono_q & f, const mono_q & g);
22
23          bool for_crit2(const mono_q & h, const mono_q & f, const mono_q & g);
24
25          void crit2(tuple2set & NC, tuple2set & C, uint s, sparse_q & lms);
26
27          void syl(sparse_q & r, const sparse_q & f, const sparse_q & g,
28                  const mono_q & lmf, const mono_q & lmg);
29
30          void groebner_buchberger(std::vector<sparse_q> & gb, sparse_q & lms,
31                  const std::vector<sparse_q> & polylist);
32
33          void minimal_groebner(std::vector<sparse_q> & gb, sparse_q & lms);
34
35          void reduced_groebner(std::vector<sparse_q> & gb);
36      }
37      void MultiGroebnerBasisQ(std::vector<sparse_q> & gb,
38              const std::vector<sparse_q> & polylist, uint totalvar, order_q order);
39  }
```

4 多项式方程组求解的实际应用

4.1 几何定理的机器证明

平面几何定理的机器证明,已有的有效方法主要有两种. 一类是由张景中院士提出的基于面积的消点算法[6], 一类是由吴文俊院士提出的基于代数方程的消元算法[7], 因为构造性平面几何命题的条件和结论可以表示为一系列次数不高的代数方程, 因此代数方程组求解的理论对几何定理机器证明十分有用.

如著名的 Simson 定理[4]: 过 P 点向 ΔABC 三边做垂线,若 P 在 ΔABC 的外接圆上,则 LMN 三点共线.



以 AB 中点为原点,AB 方向为正方向,设 $A(-x_a,0)$, $B(x_a,0)$, $C(x_c,y_c)$,

[4]详见http://en.wikipedia.org/wiki/Simson_line

$P(x_p, y_p)$, $N(x_p, 0)$, $M(x_m, y_m)$, $L(x_l, y_l)$. 其条件可以化为如下代数方程组:

$$\begin{cases} PABC\text{共圆} & \Leftrightarrow y_c y_p^2 - (y_c^2 + x_c^2 - x_a^2)y_p + y_c(x_p^2 - x_a^2) = 0 \\ PM \perp BC & \Leftrightarrow (x_c + x_a)(x_m - x_p) + y_c(y_m - y_p) = 0 \\ M \in BC & \Leftrightarrow (x_c + x_a)y_m - y_c(x_m + x_a) = 0 \\ PL \perp AC & \Leftrightarrow (x_c - x_a)(x_l - x_p) + y_c(y_l - y_p) = 0 \\ L \in AC & \Leftrightarrow (x_c - x_a)y_l - y_c(x_l - x_a) = 0 \end{cases}$$

结论可写为: $(x_n - x_p)y_l - y_c(x_l - x_p) = 0$

要证明原题结论,即证明上式在原方程组的生成理想中. 吴文俊院士提出的机器证明方法,就是针对平面几何问题的特殊性提出的判断一多项式是否属于一个理想的快速方法.

4.2 计算机图形学中的曲面拼接

在本学期选修的计算机系"计算机图形学基础"课程中,作者了解到多项式方程组求解在计算机几何造型的曲面拼接中的重要作用.

对于一个参数曲面,传统的分析学中对连续性的定义不适用于图形学中,考虑参数方程表示的以下曲线[5]:

$$f(t) = \begin{cases} \dfrac{\overrightarrow{X} - \overrightarrow{Y}}{3} t, 0 \le t \le 1 \\ \dfrac{\overrightarrow{X} - \overrightarrow{Y}}{3} + (t-1)\dfrac{2(\overrightarrow{X} - \overrightarrow{Y})}{3}, 1 \le t \le 2 \end{cases}$$

参数连续性(Parametric Continuity)C^n 要求在各点的 $\dfrac{d^n f}{dt^n}$ 连续[6]. 在此例中,$f'(1^-) \neq f'(1^+)$,因此在 1 处不满足 C^1 连续,但事实上此函数图像为一直线.

几何连续性(Geometric Continuity)G^n 忽略了参数在曲线上变化的速率,更适合描述图形学中的连续性. 设 P 点是曲面 $F_1 = 0, F_2 = 0$ 的一个公共点, 若存在 P 处非零的多项式 A, B 使得 $AF_1 - BF_2$ 的前 k 阶导数均为 0,则称两曲面在 P 处 G^k 连续.

曲面拼接问题,就是对曲面 F_1 与一辅助平面 $H = 0$ 的交线 C,求曲面 F_2,使得 F_2 与 F_1 在 C 上 G^k 连续. 其等价条件为, $\exists$多项式$A, B, F_2 = AF + BH^{k+1}$,即 F_2 在 F_1 与 H^{k+1} 的生成理想中[8]. 因此这仍然是一个多项式代数的问题.

5 总结与感悟

本篇报告介绍了多项式方程组求解的结式方法, 利用线性代数中的一些简单手段,就可以通过构造矩阵实现两个方程间的联合消元, 进一步可以实现方程组的消元. 随后,我们简要提及了更高效的抽象代数方法中的几个有价值的概念, 将其与线性代数中的一些基本概念进行类比,加深了我们对这些代数概念的理解.

20 世纪以来现代科学技术突飞猛进,尤其是计算机科学的兴起为数学插上了腾飞的翅膀, 数学理论同时也为计算机科学技术提供了广阔的舞台. 现代的计算机技术为大型的符号计算提供了可能性, 关键的问题就在于如何把抽象的代数理论算法化, 使计算机高效地处理形形色色的代数问题. 如今强大的计算机代数系统不仅是各类工程技术的助手, 对纯粹科学研究也起着不可忽略的推动作用. 在

[5]此例改编自胡事民老师"计算机图形学基础"课件

[6]详见http://en.wikipedia.org/wiki/Smooth_function#Parametric_continuity

多项式代数理论上发展的计算机图形学,机器人运动学,密码学等一系列计算机专业相关领域和理论应运而生, 也大规模的应用在我们的日常生活中. 数学的规模之大,影响之深远,已经超越了所有的时代. 虚拟现实,三维真实感动画,智能机械手臂等华丽实用的科技让人们切实感受到了数学博大精深, 以及那种浑然天成的和谐美与对称美.

这次接触计算机与数学领域重叠部分的前沿应用, 既为未来的我们在计算机系的符号计算领域的学习打下基础, 也有益于我们对已有的线性代数知识的理解和巩固. 同时,在尝试用宏观和类比的眼光透视数学问题的过程中,我们也更加体会到了数学的美感, 以及从变化中抓住不变,从紊乱中归纳条理,在偶然中发现必然,从混沌中整理秩序的科研精神. 在完成这篇作品的过程中, 从确定方向,查阅文献,类比研讨,抽象方法,再到研究实践应用的一系列过程里,我们感受到：在处理数学问题时要善于类比联想；由想激疑,在释疑中启悟；由疑反思,在思辨中省悟；由思导验,在体验中领悟;由验致用,在应用中彻悟。

6 参考文献

[1] 丘维声. *高等代数(下册)*——大学高等代数课程创新教材. 清华大学出版社, 2010. ISBN: 978-7-302-23759-4.

[2] D.A. Cox, J.B. Little, and D. O'Shea. *Ideals, varieties, and algorithms: an introduction to computational algebraic geometry and commutative algebra*. 2nd ed. Springer Verlag, 2007. ISBN: 0-387-94680-2.

[3] D.A. Cox, J.B. Little, and D. O'shea. *Using algebraic geometry*. Springer Verlag, 2005. ISBN: 0-387-98487-9.

[4] 李超 等. *计算机代数系统的数学原理*. 清华大学出版社, 2010. ISBN: 978-7-30-223010-6. URL: http://msc.tsinghua.edu.cn/NWMA2010/theses/bachelor/Bachelor-ChaoLi.pdf.

[5] 王东明等. *多项式代数*. 高等教育出版社, 2011. ISBN: 978-7-04-031698-8.

[6] 张景中. *几何新方法和新体系*. 科学出版社, 2009. ISBN: 978-7-03-025042-1.

[7] 吴文俊. *几何定理机器证明的基本原理(初等几何部分)*. 科学出版社, 1984. ISBN: 978-7-03-028377-1.

[8] J. Warren. "Blending algebraic surfaces". In: *ACM Transactions on Graphics (TOG)* 8.4 (1989), pp. 263–278.