Classes structure for ParCA-2 system

G.I.Malaschonok

1. Introduction

The project Parallel Computer Algebra (ParCA) have been started in Tambov University in 2003. During the last 5 years we have obtained an extensive collection of useful classes, polynomial and matrix algorithms, etc. We have done a lot of experiments on cluster machines: a Myrinet-IBM cluster machine with 16 processors of Tambov University and 4000-processors cluster machine of Russian Academy of Sciences Joint Supercomputer Center (JSCC) for which we have access thanks to the RAS. These experiments demonstrated the acceleration of matrix and polynomial multiplication operations more then 70% and the acceleration of matrix operations of inversion family more then 50%. We say that acceleration is equal 100% if the real time of some operation is n-times bigger for one machine than for the cluster of the n-machines. These experiments let us understand that parallel computer algebra may become a realy efficient system.

A new stage of the ParCA project is connected with two problems.

First problem is the problem of complicated class structure. We have to duplicate many algorithms for different number rings.

The second problem is an absense of large objects of data. Each object in ParCA must be less than core memory. But for the big cluster machine we want to have the big objects. Just in this case we have a chance to get an efficient computational process for big cluster machine.

That is why at the end of 2008 year we have started the ParCA-2 project. Two main new features of this project are a new classes structure a new file-oriented data.

2. Class structure of ParCA-2

The main abstract class for ParCA-2 is a Scalar class. The abstract methods of this class cover all possible scalar operations. The methods are: add, subtract, multiply,

This work was completed with the support of our RFBR grant N11111.

divide, power, GCD, extendedGCD, mod, modInvers, modPower, divideAndRemainder, random, a family of "valueOf"methods for creating a definite scalar type from another scalar type. There are no objects in this class.

Daughter classes of the Scalar class are devided into several groups. There are numbers, polynomials, rationales and functions.

We have 12 classes for numbers. There are 6 exact number classes and 6 rounded number classes. The exact sets of numbers are: Z, Q, Cz, Cq, Zp and Zp32. We denote $Cz = \{a+ib : a, b \in Z, i^2 = -1\}$ and $Cq = \{a+ib : a, b \in Q, i^2 = -1\}$, Zp=Z/pZ, with arbitrary prime number $p \in$Z and $Zp32 = Z/pZ$, with prime number p less than maximal positive number of 4-bite word. The last class of numbers is the most fast and the most often used in computer algebra applications.

The rounded sets of numbers are: $R, C, R64, C64, R128$ and $C128$. Here we use the next notation. $R64$ is a standard 64 bit-length floating point real number. R is an arbitrary-length real number. $R128$ is a pare of a 64 bit-length floating point real number and an additional 64 bit-length word for an order. Three complex classes C, $C64$ and $C128$ are obtained from the classes R, $R64$ and $R128$, consequently.

The Polynomial class of many variables is an only one for any of 12 number types.

The class of Fractions is a daughter class of scalars. The numerator and the denominator of fractions are scalars.

The class of Rationals is a daughter class of Fractions. The numerator and denominator of rational element are polynomials. The other daughter class of Fractions is Q-number class. The numerator and denominator of Q-number are Z-numbers.

The other daughter class of Scalar is the class of Functions. Class Function is obtained as a composition of elementary transcendental functions and polynomial functions.

Therefore the total number of scalar classes is 60: 12 numbers classes, 12 polynomial classes, 12 rational classes and 12 function classes.

The other daughter class of Scalar is the class of Function. Class Function is obtained as a composition of elementary transcendental functions and polynomial functions.

We think that later the scalar class of infinite series may be done too.

The classes of Matrix, Vector and Tensor have the Scalar type elements. So matrix method may be applied to any Scalar class.

The main matrix methods are: “adjoint”, “det”, “kernel”, “toEchelonForm”, “inverse”. The main method is the recursive block method “adjDet”. This method returns the matrix determinant together with adjoint and echelon matrices. It is easy to obtain the kernel matrix from echelon matrix and inverse matrix from adjoin matrix and determinant.

We have started to construct the classes of file-matrices and file-polynomials. We expect that with these new file-classes we will have possibility to solve problems with a very large input data.

Список литературы

[1] Г.И.Малашонок, А.И. Авитесян, Ю.Д.Валеев, М.С. Зуев. Параллельные алгоритмы компьютерной алгебры. Труды Института системного программирования , 2004 г., т.8, ч.2, с.169-180.

[2] Г.И.Малашонок, Ю.Д.Валеев. О некоторых подходах к построению параллельных программ. Вестник тамбовского университета, Т.10, N.1. 2005, 154-156.

[3] Г.И.Малашонок, М.С. Зуев. О представлении матриц кватернарными деревьями. Вестник тамбовского университета, Т.10, N.1. 2005, 157-160.

[4] Г.И.Малашонок, Ю.Д.Валеев, М.С. Зуев. О параллельных матричных алгоритмах в компьютерной алгебре. Вестник тамбовского университета, Т.10, N.1. 2005, 161-163.

[5] G.I.Malaschonok. In the Direction of Parallel Computer Algebra System. Computer Science and Information Technologies. Proc. Conf. (Sept.19-23, 2005. Acad. Sci. of Armenia.) Yerevan, 2005, 451-453.

[6] Г.И.Малашонок. Об одном подходе к построению параллельной системы компьютерной алгебры Дифференциальные уравнения и системы компьютерной алгебры. Сб. научн. статей международной конф. (Брест, 5-8 сент. 2005 г.). Ч. 1. - Минск, БГПУ, 2005, 306-307.

[7] Г.И.Малашонок, Ю.Д.Валеев. Рекурсивное распараллеливание символьно-численных алгоритмов. Вестник Тамб. ун-та. Сер. Естеств. и техн. науки. Тамбов, 2006. Т.11, вып. 4, С. 536-549

[8] Г.И.Малашонок, Ю.Д.Валеев. Параллельные полиномиальные рекурсивные алгоритмы. International conference Polynomial Computer Algebra. St.Petersburg, PDMI RAS, 2008, С. 41-45.

[9] Г.И.Малашонок. О параллельной конструктивной математике. Международная научная конференция "Современное математическое образование и проблемы истории и методологии математики". Тамбов, 2008. С.194-195.

[10] Г.И.Малашонок, Ю.Д.Валеев. Организация параллельных вычислений в рекурсивных символьно-численных алгоритмах. Труды конференции ПаВТ'2008 (Санкт-Петербург). Челябинск: Изд-во ЮУрГУ, 2008. С. 153-165

G.I.Malaschonok
Tambov State University
Internatsionalnaya, 33
392000 Tambov
Russia
e-mail: malschonok@ya.ru

Effective normalization of a nonsingular in codimension one projective algebraic variety

Alexander L. Chistov

St. Petersburg Department of Steklov Mathematical Institute
of the Academy of Sciences of Russia
Fontanka 27, St. Petersburg 191023, Russia,
e-mail: alch@pdmi.ras.ru

2009

Abstract

Let V be a nonsingular in codimension one projective algebraic variety of degree D and of dimension n. Then the construction of the normalization of V can be reduced canonically within the time polynomial in the size of the input and $D^{n^{O(1)}}$ to solving a linear equation $aX + bY + cZ = 0$ over a polynomial ring.

Let k be a field with algebraic closure $\overline{k}$. Let $V \subset \mathbb{P}^N(\overline{k})$ be a projective algebraic variety defined over k and irreducible over $\overline{k}$ of dimension n and degree D. The the construction of the normalization of V can be divided into two parts:

(I) construction of a finite birational isomorphism $V' \to V$ such that the variety V' is nonsingular in codimension one; this means by definition that the dimension $\dim \mathrm{Sing}(V') \leqslant n-2$, where $\mathrm{Sing}(V')$ is the variety of singular points of V',

(II) construction of the normalization of V'.

In the case of zero–characteristic $\mathrm{char}(k) = 0$ one can solve (I) within the polynomial time using the ideas from [?]. Everything is reduced here to the case of the curves $n = 1$. We hope to describe this construction in detail in another paper. To obtain a similar effective algorithm for (I) in the case of nonzero characteristic is an open problem.

In this paper we concentrate ourself on problem (II). So we shall suppose in what follows that $V = V'$ is nonsingular in codimension one. In the case of surfaces $n = 2$ problem (II) can be solved within the polynomial time without difficulties (it is not our aim now). Besides that, the case $n = 2$ can be easily reduced to $n = 3$. So we shall suppose in what follows without loss of generality that $n \geqslant 3$. We assume that V is given by its generic point, see [?]. Notice also here that constructing an appropriate linear projection one can always reduce the general case to the particular one: $N = n + 2$ (but we don't give the details now). So we shall suppose in what follows that $N = O(n)$.