

УДК 519.688

PROJECT OF PARALLEL COMPUTER ALGEBRA

© G. I. Malaschonok

Key words: parallel computer algebra system; Laboratory of Algebraic Computations; parallel computations; symbolic computations; computer algebra.

The main features of current version of parallel computer algebra system developed in the Laboratory of Algebraic Computations of Tambov university together with MSoft Center are presented.

1. Introduction

Contemporary software for supercomputer supports numerical computations. But contemporary mathematics is a symbolic mathematics. The elementary mathematical objects are not numbers, but functions, operators, functional matrices and other objects. Therefore the supercomputer software must supports *symbolical-numerical computations*.

The well known and the most popular mathematical library for supercomputer is Intel Math Cernal Library (MKL). The Intel MKL consists of the following packages:

- linear algebra: dense matrices (BLAS, LAPACK, ScaLAPACK);
- linear algebra: sparse matrices (Sparse BLAS);
- fast Fourier Transforms (SMP и MPI);
- Vector Math Library (VML);
- Vector Statistical Library (VSL);
- Optimization Solvers (TRS).

All this packages are numerical. The next step of Math Kernel Library must be Math Symbolic Library - The Library of «Packages of Parallel Symbolic Mathematics» (MSL).

The project Parallel Computer Algebra has started in Tambov University in 2003 in collaboration with Institute for System Programming RAS. We carried out experiments on cluster of ISP RAS, then on Myrinet-IBM cluster of Tambov University and 8000-processors cluster of Joint Supercomputer Center (JSCC RAS).

In 2008 we started the ParCA-2 project [1] and in 2009 together with MSoft Center we started to realize the main part of ParCA-3 system.

2. The current tasks of the project

We fulfill the following symbolical-numerical computations:

- computations with polynomials of several variables,
- computations with symbolic compositions of elementary functions,
- symbolic integration and differentiation,
- trigonometric, logarithmic and exponential simplifications,
- solving some classes of differential equations and their systems in symbolic form,

- FFT for polynomials and integer numbers and multiplication on the base of FFT,
- computation with matrices over the functional compositions,
- computation of inverse, adjoint and echelon matrices and the kernel of a linear operator,
- computation of characteristic polynomial, determinant and spectrum of operator.

3. Structure of ParCA-3 system

Three parts of the ParCA-3 system are Front End, Kernel and Parallel part. Each part consists of several packages.

(1) The Front End part consists of two packages: *frontend* and *visualmath*. The frontend package is a collection of classes which support the low-level notebook programming for users of ParCA system. The *visualmath* package is devoted to the visualization of functions and geometric shapes.

(2) The Kernel part consists of four packages: *number*, *polynom*, *func*, and *matrix*. These packages support corresponding mathematical objects: numbers, polynomials, functions and matrices.

(3) The Parallel part consists of two packages: *parjava* and *parca*. The *parjava* package is a Java-MPI interface. The *parca* package consists of all parallel software of the ParCA system.

4. The computational paradigm

Principles of algebraic structures.

1. Each mathematical object is an element of some algebra (algebraic structure).
2. Each algebra is represented by some class in corresponding package. Therefore each element of algebra may be defined by the value of *dynamical fields of its class* and each operation of this algebra is represented by some *method of its class*.
3. In the top of the algebraic hierarchy we put some abstract algebra and call it «*element*». The algebraic hierarchy is represented by the corresponding *class hierarchy* with *Element class* in the top. Each class which represents some real algebraic structure has to extend this abstract class of *Elements*. There are no dynamical fields in this class but all possible operations for the main types of usually used algebraic structures with one and two binary operations must be defined in this class of *Elements*.

Principles of working context.

1. We name the direct product of several numerical algebras an algebraic space. For example, we can talk about the algebraic space $\mathbb{R}^n \mathbb{C}^m \mathbb{Z}^k$.
2. To carry out some symbolical-numerical computations we have to fix an algebraic space. We name a set of mathematical expressions which are defined on one algebraic space the *page*. We name a set of pages a *notebook*.
3. *Page* is a class. To create some page, or more correctly, to create an instance of the page class, we have to fix some algebraic space. All expressions at one page are considered as functions on this space.
3. To map this algebraic space V_1 into another algebraic space V_2 we have to make a map from this page on another page with this algebraic space V_2 . We may perform this map in another way. We have to fix the direct product $V_1 \times V_2$ as a algebraic space of page and then we may take any map from V_1 to V_2 in the same page.
4. To fix the algebraic space we must define the set of numbers and the names of variables, for example: $V_1 = \{u, v \in \mathbb{C}, x, y, z \in \mathbb{R}, n, m \in \mathbb{Z}\}$.

5. Current algebraic objects

There are defined standard **numerical algebras**: Z , Z_p and Z_{p32} . We denote $Z_p = Z/pZ$, with arbitrary prime number $p \in Z$ and $Z_{p32} = Z/pZ$, with a prime number p less than maximal positive number of 32-bit word.

Several approximate numerical algebras are defined for the sets of rounded numbers: R , $R64$, $R128$. The $R64$ is a standard 64 bit-length floating point real number, R is an arbitrary-length real number, $R128$ is a pair of a 64 bit-length floating point real number and an additional 64 bit-length word for an order.

The field of fractions may be obtained by means of Fraction class from any integer domain like Z , $R[x, y]$, etc.

The algebra of complex numbers may be obtained from any algebra due to the class Complex by means of standard complexification procedure. For example, three complex algebras C , $C64$ and $C128$ may be obtained from the algebras R , $R64$ and $R128$, correspondingly.

6. Current functions

The func package (F) contains the class of functions and supplementary classes. Each function of this class contains two dynamical fields: *name* – for titles of this function and *X* – for the array of Elements, which are arguments of this function. This class permits to construct any composition of transcendental and rational functions.

Modern computer systems like Mathematica has hundreds of mathematical functions and operators. We call them «calculus related functionality». The list of pairs with corresponding names, that are used in ParCA and Mathematica systems is presented below.

Functions: ID[x] = x, ABS[x] = Abs[x], LN[x] = Log[x], LG[x] = Log10[x], EXP[x] = Exp[x], Sqrt[x] = Sqrt[x], SIN[x] = Sin[x], COS[x] = Cos[x], TG[x] = Tan[x], CTG[x] = Cot[x], ARCSIN[x] = ArcSin[x], ARCCOS[x] = ArcCos[x], ARCTG[x] = ArcTan[x], ARCCTG[x] = ArcCot[x], SH[x] = Sinh[x], CH[x] = Cosh[x], TH[x] = Tanh[x], CTH[x] = Coth[x], ARCSH[x] = ArcSinh[x], ARCCH[x] = ArcCosh[x], ARCTGH[x] = ArcTanh[x], ARCCTGH[x] = ArcCoth[x], UNITSTEP[x] = UnitStep[x], FACT[x] = Factorial[x], GAMMA[x] = Gamma[x], SC[x] = Sec[x], CSC[x] = Csc[x], SIGN[x] = Sign[x], CUBRT[x] = Power[x,1/3], ARCSEC[x] = ArcSec[x], ARCCSC[x] = ArcCsc[x], GAMMA2[a,x] = Gamma[a,x], BINOMIAL[m,n] = Binomial[m,n], LOG[a,x] = Log[a,x], POW[x,a] = Power[a,x];

Operations: ADD = Plus, MULTIPLY = Times, SUBTRACT = Subtract, DIVIDE = Divide, D = D, INT = Integrate, SUM = Sum, PROD = Prod, LIM = Lim, SERIES = SeriesCoefficient, VECTOR = List, PLOT = Plot, PARAMPLOT = ParametricPlot, TABLEPLOT = ListPlot, SHOWGRAF = Show;

Others (Mathematica does not have it): DEFINITION, ALGEBRAIC_EQUATION, EQUATION, SEQUENCE, SYSTLAE(System of linear algebraic equations), SYSTLDE (System of linear differential equations), INITCOND (Initial conditions for system of differential equations), DOMAIN_OF_FUNC (Analog of Mathematica's Assumptions).

7. The matrix package

The **matrix package** contains the classes of dense and sparse matrices in two forms: in-core and out-of-core. The out-of-core form of matrices is necessary for very large matrices, which cannot be disposed in core. The main matrix methods are: «adjoint», «det», «kernel», «toEchelonForm», «inverse».

Computation algorithms for symbolical-numerical matrices are the most interesting objects

for parallelization.

The following matrix algorithms were obtained in parallel form.

1. Matrix multiplication.
2. Computation of adjoint, inverse and echelon form of matrix.
3. Computation of determinant and kernel.
4. Computation of matrix characteristic polynomial.
5. Computation of adjoint matrix for the polynomial matrix on the base of polynomial FFT algorithm.
6. Solving system of linear equations in the fraction field of integer number on the base of p-adic lifting.

8. The polynomial package

The polynomial package (Polynom) consists of classes of polynomials of several variables, rationals in factorized form (FactorPol), and sums of factorized rationals (FactorPolSum).

The class *Ring* of polynomial package defines the algebraic space in the form of polynomial ring. The algebraic space $V_1 = \{u, v \in \mathbb{C}, x, y, z \in \mathbb{R}, n, m \in \mathbb{Z}\}$ is defined by the ring $R_1 = \mathbb{C}[u, v]\mathbb{R}[x, y, z]\mathbb{Z}[n, m]$.

The most complicated and important problem among the polynomial problems is the Groebner basis computation.

Next polynomial algorithms were obtained in parallel form:

1. Polynomial multiplication.
2. Computation of the Groebner basis for a given ideal.

9. Online access to the website with ParCA

The universities and research institutes of the Academy of Sciences which participate in the programme «University Cluster» will have online access to the Cluster with ParCA System.

First experiments have showed that the system demonstrates different scalability for the problems of different sizes. So, for the best efficiency the number of processors have to be chosen according to the size of the problem. Future experiments will permit us to make a table which will show how many processors are needed for the defined problems.

Execution of computations.

An interface is realized as a Web-application. We use the following Web-technologies: JavaScript; Cascading Style Sheets CSS Java Servlet; DOM — Document Object Model; XMLHttpRequest; XML; Ajax. Apache Tomcat is a web server and MySQL is a database management system.

The *ParCA* system provides the control of task launching at cluster computer systems. The following toolware is exploited:

- 1) starting the task from the shape;
- 2) starting the task from the file loaded onto the server;
- 3) obtaining the result on-line;
- 4) control the queue of started tasks;
- 5) management by results;

- 6) loading of files onto the server;
- 7) management by files which were loaded onto the server.

10. Tasks

To start the task we have to implement the following: (1) Fill the main shape. (2) Fill the shape of parameters (3) Start the task.

We have two different possibilities for starting the task: (1) to start the task with the input data getting from the shape (2) to start the task with input data getting from the file. In the first case we have to write the title of the task and input data onto the shape.

In the current version a user can do one of 8 types of the tasks:

- 1) *gbasis* — computation of a Groebner basys;
- 2) *adjoint* — computation of an adjoin matrix;
- 3) *adjointDet* — computation of an adjoin matrix and a determinant;
- 4) *echelonForm* — computation of the echelon form of a matrix;
- 5) *inverse* — computation of the inverse matrix;
- 6) *det* — computation of the matrix determinant;
- 7) *kernel* — computation of the kernel of a linear operator;
- 8) *charPol* — computation of the characteristic polynomial.

To start the task from a file we have to write the file name with input data at the shape. The system makes it possible to start several tasks from one shape.

If it is necessary to add additional parameters the next steps must be implemented: (1) Click the button *Add*. (2) After appearing of a new window write the parameter. (3) Confirm the corrections by clicking *OK*. (4) Input the new parameters in the new window.

If you want to know the status of the started tasks choose *Queue*. You will see the shape, which allows a user to fulfill the following two functions: (1) to give an actual for this moment information about all the tasks of all the users; (2) to delete one or several tasks from the queue. In order to delete the tasks from the queue select them in the left column and press the button *Kill*.

View the list of results. You have to select in menu the item *Show Result*. You can see a shape with results. This shape let a user fulfill the following: to view the results of computations, to save the results at local computer, to delete the results from server.

View the list of files. To look through the list of files at server choose the item *Show Files* in the menu. This shape let a user fulfill the following: to see the contents of the file, to save the file at the local computer, to delete the file from the server.

Loading files onto the server. In order to load files onto the server it is necessary to do the following: (1) click the button *Upload File*. (2) Then press the button «Search». The dialog window will open. Select the file and click «open», then click the button «Upload» and wait for the end of this process.

REFERENCES

1. *Malaschonok G.I.* Classes structure of ParCA-2 system // International conference Polynomial Computer Algebra. St. Petersburg, PDMI RAS, 2009. P. 66-68.
2. *Malaschonok G.I., Avetisan A.I., Valeev Yu.D., Zuev M.S.* Parallel algorithms of computer algebra // Proceeding of the institute of system programming. 2004. V. 8. Issue. 2. P. 169-180.
3. *Malaschonok G.I.* 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. P. 451-453.
4. *Malashonok G.I.* Control of parallel computing process // Tambov University Reports. Series Natural and Technical Sciences. Tambov, 2009. V. 14, Issue. 1. 269-274. (Russian).
5. *Malaschonok G.I., Valeev Yu.D., Lapaev A.O.* On the choice of multiplication algorithm for polynomials and polynomial matrices // Zapiski POMI. 2009. V. 373. P. 157-188. (Russian).
6. *Malashonok G.I.* On the project of parallel computer algebra // Tambov University Reports. Series Natural and Technical Sciences. Tambov, 2009. V. 14. Issue. 4. P. 744-748. (Russian).
7. *Malaschonok G.I.* Architecture of ParCA3 project // International conference Polynomial Computer Algebra. St.Petersburg, PDMI RAS, 2010. P. 44-47.
8. *Malaschonok G.I., Starov M.V., Betin A.A., Pereslavytseva O.N., Pozdnykin A.G.* Parallel computer algebra. Part 1. The textbook. Tambov: Publishing house of TSU, 2010. (Russian).
9. *Malaschonok G.I.* Computer mathematics for computational network // Tambov University Reports. Series Natural and Technical Sciences. Tambov, 2010. V. 15. Issue 1. P. 322-327. (Russian).

БЛАГОДАРНОСТИ: Работа выполнена при финансовой поддержке АБЦП «Развитие научного потенциала высшей школы (2009-2010 годы)» (проект № 2.1.1/1853); темплана 1.12.09.

Поступила в редакцию 25 августа 2010 г.

Малашонок Г. И. Проект параллельной компьютерной алгебры.

В работе представлены основные характеристики системы параллельной компьютерной алгебры, разрабатываемой лабораторией алгебраических вычислений Тамбовского государственного университета совместно с МСофт Центром.

Ключевые слова: система параллельной компьютерной алгебры; лаборатория алгебраических вычислений; параллельные вычисления; символьные вычисления; компьютерная алгебра.