Available online @ www.iaraindia.com
RESEARCH EXPLORER-A Blind Review & Refereed Quarterly International Journal
ISSN: 2250-1940 (P) 2349-1647 (O)
Impact Factor: 3.655 (CIF), 2.78 (IRJIF), 2.62 (NAAS)
Volume VIII, Issue 26
January - March 2020
Formally UGC Approved Journal (63185), © Author

THE UTILIZE OF COMPUTER ALGEBRA AND ITS APPLICATIONS SOLVING SOME ALGEBRAIC EQUATIONS WITH MATLAB CODE

DR. MAHESH BOHRA
Department of Mathematics
Govt. Women Engineering College,
Ajmer, Rajasthan, India

RISHABH CHAUDHARY
Department of Mathematics
Bhagwant University, Ajmer, Rajasthan, India
&
MANOJ KUMAR
Department of Mathematics
Bhgwant University, Ajmer, Rajasthan, India

Abstract

In this paper we are discusses some applications of computer algebra and its application in solving algebraic equations with Matlab code. Computer algebra is an area of computer science and mathematics where you can calculate symbols representing mathematical objects instead of their numeric values. The purpose of computer algebra systems using computers is to symbolically manipulate a formula. For example, we see that some algebraic polynomial has some common uses of expansion, factoring, root finding, or simplification. As we see, some systems also provide us with numerical computation, graphics and simulation other functionalities. We have surveyed in this paper the problem of implantation and which currently examines its available methods. We have implemented and tested using methods of algorithms for practical examples.

Keywords: Root, Algorithm, Application, Computer, Algebra, Numeric Value.

I. Introduction

A computer can be any algebraic system (CAS) with the ability to manipulate mathematical expressions similar to manual calculations of scientists and mathematical mathematicians. "Computer algebra" or "symbolic computation" is part of the development of computer algebra systems in the late 20th century that have accelerated the work of algorithms on mathematical objects such as polynomials. Here we can divide computer algebra systems into two parts. Which is thus special and general purpose? Are devoted to a specific part of specific mathematics, such as number theory, group theory, or elementary

mathematics teaching. The purpose of a general-purpose computer algebra system should be useful to a user working in any scientific field that requires manipulation of mathematical expressions. To be useful, a general-purpose computer algebra system must include various features such as [1]:

- Allows the user interface to enter and display mathematical formulas. And usually from a menu selection, keyboard, mouse, or stylus.
- A simplified, a rewrite system to simplify mathematical formulas,
- Here is a large library of special functions and mathematical algorithms
- A convincing-exact numeral may be required for the larger size of the integer [1],

For the needs of users, the library should provide not only simplification needs but also [3]. For example, we can see that the calculation of the largest common divisors of a polynomial can be used to simplify expressions involving fractions systematically [2].

Other fields have a range of applications, such as in algebraic geometry in polynomial systems, applications in proving automated geometric theorems, computer aided design (CAD) and computer aided manufacturing (CAM) systems, computer graphics, virtual reality investigations. We have raised this problem to implicate [4].

These are proven and implemented using the software package Matlab, some samples run to present their applicability. And other algorithms have also been suggested [5].

II. Algebraic calculations

Symbol manipulation or algebraic computation of computer algebra is an area of scientific computation that develops, analyzes, implements and uses algebraic algorithms. The characteristics of algebraic computation are as follows [6,7,8]:

- Compute with precise exact numbers - no rounding.
- Computation between variables and symbols (eg, a,b).
- Computations between functions (eg sin a, cos b).
- Manipulate of formulas.
- Operations of symbolic (differentiation, factorization, integration etc.) [9,10]

Unlike numeric differentials, the following are symbolic differentials follows as [11,12]:

Computation of Numerical	**Computation of Symbolic**
Here we are computing only with numbers. e.g.: 4 + 7 = 11	Computing in algebraic number domains algebraic structures such as polynomial rings, finite fields, etc. with computing. Eg.: a+4b=6a, 1/12+1/8=5/24
Approximate results E.g.: 0.86602…, cos(3.1415)= -133….	Accurate representation of results E.g.:√3/2, Cos(√3/2)=-1
Numeric evaluation e.g.: x^2-y^2 = ?? (undefined x,y)	Symbolic simplification e.g.: x^2-y^2 = (x-y)(x+y)

III. Manipulation of Symbolic

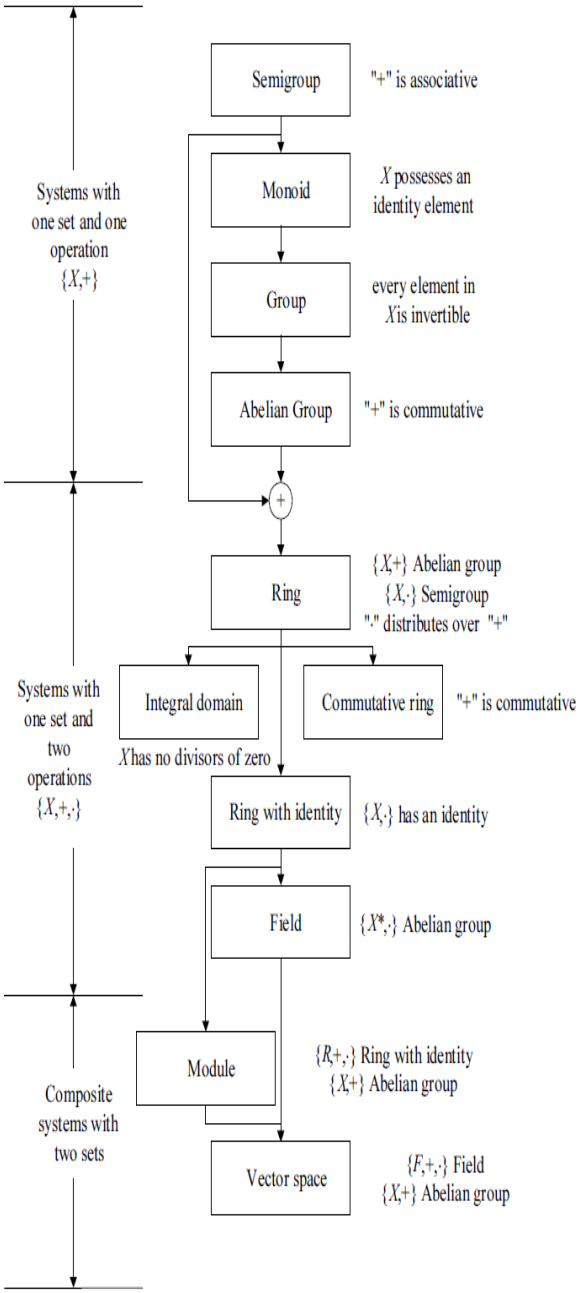
Commonly supported symbolic manipulations include [13]:

- statistical computation
- integral transforms
- taking some limits
- Some differences and differential equations are solved [14].
- There are some solutions to linear and some non-linear equations on different domains.
- partial and total differentiation
- Is a short expression or some standard form of simplification and assumptions including automatic simplification and simplification with constraints [15]

IV. computer algebra systems used in mathematics

- Knuth–Bendix completion algorithm[2]
- Root-finding algorithms[2]
- Polynomial factorization via e.g., over finite fields,[3] Berlekamp's algorithm or Cantor–Zassenhaus algorithm.
- Gaussian elimination[4]
- Greatest common divisor via e.g. Euclidean algorithm[5]
- Chinese remainder theorem[5]
- Diophantine equations[5]
- Hypergeometric summation via e.g. Gosper's algorithm[5]
- Limit computation via e.g. Gruntz's algorithm[5]
- Cylindrical algebraic decomposition[5]
- Padé approximant[5]
- Schwartz–Zippel lemma and testing polynomial identities[5]

V. Relationship between basic algebraic systems [1]



VI. Solving algebraic equations with matlab CODE and VI. The resultant method [1][7]

Suppose you have the system [6]

$$x^2y^2=0$$

$$x-y/2=\alpha$$

and solve for *x* and *y*. First, create symbolic objects [6].

syms x y a

There are ways to output of solve [6].

[solx,soly] = solve(x^2*y^2 == 0, x-y/2 == a)

The call returns the following [6].

solx =

0

a

soly =

-2*a

0

Modify the equation to $x^2y^2 = 1$. The new system has more solutions [6].

[solx,soly] = solve(x^2*y^2 == 1, x-y/2 == a)

Four distinct solutions are produced [6].

solx =

a/2 - (a^2 - 2)^(1/2)/2

a/2 - (a^2 + 2)^(1/2)/2

a/2 + (a^2 - 2)^(1/2)/2

a/2 + (a^2 + 2)^(1/2)/2

soly =

- a - (a^2 - 2)^(1/2)

- a - (a^2 + 2)^(1/2)

(a^2 - 2)^(1/2) - a

(a^2 + 2)^(1/2) - a

Since not specify the dependent variables, solve uses symvar to determine the variables [6].

This way is assigning output from solve is quite successful for "small" systems. For instance, if you have a 10-by-10 system of equations, typing the following is both awkward and time consuming [6].

[x1,x2,x3,x4,x5,x6,x7,x8,x9,x10] = solve(...)

To circumvent this difficulty, solve can return a structure whose fields are the solutions [6]. For example, solve the system of equations u^2 - v^2 = a^2, u + v = 1, a^2 - 2*a = 3.

syms u v a

S = solve(u^2 - v^2 == a^2, u + v == 1, a^2 - 2*a == 3)

The solver returns its results enclosed in this structure.

S =

struct with fields:

a: [2×1 sym]

u: [2×1 sym]

v: [2×1 sym]

The solutions for a reside in the "a-field" of S [6].

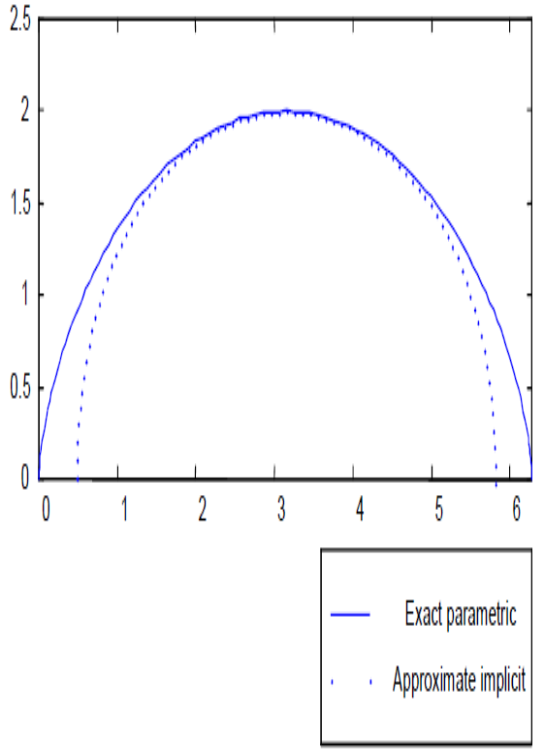
S.a

Sample run

```
» t=sym('t');
» x=1-t^2;
» y=2*t;
» w=1+t^2;
» pretty(x)
 1 - t²
» pretty(y)
2 t
» pretty(w)
1 + t²
» imp=implicit(x,y,w);
» pretty(imp)
x² - 1 + y²
```

Answer =
-1
3

Exact and approximate plot figures are shown [1].



REFERENCES

[1] Hazem Mohamed El-Alfy" Computer Algebra And Its Applications" Alexandria University Faculty Of Engineering Department Of Engineering Mathematics And Physics, *Alexandria University, 1997.*

[2] B. Buchberger; G.E. Collins; R. Loos (2013-06-29). Computer Algebra: Symbolic and Algebraic Computation. Springer Science & Business Media. *ISBN 978-3-7091-3406-1*.

[3] Joachim von zur Gathen; Jürgen Gerhard (2013-04-25). Modern Computer Algebra. Cambridge University Press. *ISBN 978-1-107-03903-2*.

[4] Keith O. Geddes; Stephen R. Czapor; George Labahn (2007-06-30). Algorithms for Computer Algebra. Springer Science & Business Media. *ISBN 978-0-585-33247-5*.

[5] https://en.wikipedia.org/wiki/Computer_algebra_system

[6] https://www.mathworks.com/help/symbolic/solve-a-system-of-algebraic-equations.html

[7] Tom Sederberg, Ron Goldman and Hang Du "Implicitizing Rational Curves by the Method of Moving Algebraic Curve". J. Symbolic *Computation 23, 1997.*

[8] R. N. Goldman, E. W. Chionh, and M. Zhang "On a Relationship between the Moving Line and Moving Conic Coefficient Matrices". Computer Aided *Geometric Design -16, 1999.*

[9] R. N. Goldman, E. W. Chionh and M. Zhang "Transformations and Transitions from the Sylvester to the Bézout Resultant". Technical report TR-99, 1999.

[10] Sandra Licciardi and Teo Mora" Implicitization of Hypersurfaces and Curves by the Primbasissatz and Basis Conversion". Presented at the Meeting Algebraic Algorithms for *Geometric Reasoning, Linz, 1992.*

[11] J. Rafael Sendra and Franz Winkler "Parameterization of Algebraic Curves Over Optimal Field Extensions". *J. Symbolic Computation-23, 1995.*

[12] Joseph Schicho " The Parameterization Problem for Algebraic Surfaces". Research Institute for Symbolic Computation, *Linz, Austria, 1999.*

[13] Dimitrios Poulakis and Evaggelos Voskos "On the Practical Solution of Genus Zero Diophantine Equations". J. Symbolic Computation -30, 2000.

[14] Erik Hillgarter "The Classification Problem for Point Symmetries of 2nd Order PDE's in one Dependent and two Independent Variables", *1999.*

[15] Günter Czichowski "Lie Theory of Differential Equations and Computer Algebra". *Seminar Sophus Lie, 1991.*