doi: 10.21495/71-0-109

25th International Conference
ENGINEERING MECHANICS 2019
Svratka, Czech Republic, 13 – 16 May 2019

RAPID RESEARCH WITH COMPUTER ALGEBRA SYSTEMS

C. Fischer*

Abstract: *Computer algebra systems (CAS) are gaining popularity not only among young students and scholars but also as a tool for serious work. These highly complicated software systems, which used to just be regarded as toys for computer enthusiasts, have reached maturity. Nowadays such systems are available on a variety of computer platforms, starting from freely-available on-line services up to complex and expensive software packages. The aim of this review paper is to show some selected capabilities of CAS and point out some problems with their usage from the point of view of 25 years of experience.*

Keywords: Computer algebra system, Methodology, Wolfram Mathematica

1. Introduction

The Wikipedia page (Wikipedia contributors, 2019a) defines CAS as *a package comprising a set of algorithms for performing symbolic manipulations on algebraic objects, a language to implement them, and an environment in which to use the language.* There are 35 different systems listed on the page, four of them discontinued. The oldest one, Reduce, was publicly released in 1968 (Hearn, 2005) and is still available as an open-source project. Maple (2019a) is among the most popular CAS. It was first publicly released in 1984 (Maple, 2019b) and is still popular, also among users in the Czech Republic. PTC Mathcad (2019) was published in 1986 in DOS as an engineering calculation solution, and gained popularity for its ability to work with typeset mathematical notation in combination with automatic computations. Unlike other listed systems, the symbolic capabilities here are limited; Mathcad remains oriented towards numerical engineering applications namely the verification, validation, documentation and re-use of engineering calculations (Wikipedia contributors, 2019b). Unlisted there however are on-line graphics calculators such as GeoGebra (2019), cloud computing services such as WolframAlpha (2019) and other teaching tools.

More than half of the listed CAS are open source projects, most of them actively developed. An interesting example is MuPAD (Wikipedia contributors, 2019c). At the turn of the century it was a very powerful system, easy to use and freely available for a variety of operating systems. It was originally developed by the MuPAD research group at the University of Paderborn. The successful product was taken over several times and was finally purchased by MathWorks. MuPAD became MuPAD® and was included in the Symbolic Toolbox in Matlab. Eleven of the twelve most recent CAS projects are open source.

The history of Wolfram Mathematica dates back to 1988 when the first version was publicly released together with a 750-page book (Wolfram, 1988), for particular details see (Wolfram.com, 2019). The author's personal experience begins in 1994 with version 2.0, which was actually released three years earlier. His first impression of the symbolic computational engine was very positive, even though the PC versions from that time (running on Windows 3.1) were, to put it cautiously, less than stable. Fortunately, the stability issues significantly diminished after switching to the linux version, in 1998, and the good impression survived.

2. Language and Usage

The user interfaces, or front-ends, of most of the contemporary systems are more or less similar to one another. They generally offer a WYSIWYG graphical interface resembling a standard text document.

* RNDr. Cyril Fischer, Ph.D., Institute of Theoretical and Applied Mechanics,
Prosecká 76, 190 00 Prague 9, tel. +420 225 443 310, e-mail fischerc@itam.cas.cz

However, ease or difficulty of working with a particular CAS depends mostly on the language used for communication between the user and the computational back-end. Specific needs implied by the symbolic nature of particular solved problems lead to the fact that every CAS defined its own programming language, which in some cases was later extended with a graphical interface. For example, the oldest CAS, Reduce, is based on an ALGOL 60-type syntax modification of the LISP language. Maple uses a language similar to Pascal. MuPAD also uses a similar language, and in addition it provides support for object oriented programming. All the mentioned languages proceed from the standard programming methodology introduced in mid-1950s. Most technical school graduates are familiar with such an approach.

The Wolfram Mathematica programming language was developed with the intention to allow its users greater variability. Usually it is described as a *multi-paradigm computational communication language*. Although most of the advanced properties of the software package are in fact application libraries, the possibility of building such a wide spectrum of applications confirms the variability of the language core. Its main designer, Stephen Wolfram, is acknowledged for his scientific works in computer science, mathematics, and theoretical physics. Thus, the language was written by a mathematician for mathematicians. It contains most of the elements of traditional programming languages (even the `Goto[]` statement), however, the more the user learns the language, the less he uses traditional constructions. *The Wolfram Language provides a unique opportunity not only to introduce someone to programming, but to quickly get them to the very frontiers of what can be done with computation today* (Wolfram, 2019). A few features of the Wolfram Language are shown in Table 1.

3. Examples for Practice

This short contribution cannot substitute a proper introduction to the Wolfram Mathematica language. For that purpose the reader is kindly referred to many available guides and books (Wolfram, 2019; Mangano, 2011; Trott, 2006, 2007, 2013; Trott and Trott, 2017). Only several simple examples will be presented here.

3.1. Natural Symbolic Derivation

Consider the mechanical system of a mass at the end of a spring. The kinetic and potential energies are $T = {}^1\!/_2\, m\dot{x}^2$ and $V = {}^1\!/_2\, kx^2$ (m – mass, k – stiffness). The corresponding Euler-Lagrange equation reads

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial}{\partial \dot{x}}(T-V)\right) - \frac{\partial}{\partial x}(T-V) = 0 \tag{1}$$

whose form can be evaluated in Mathematica using a single command:

input:
```
T = 1/2 m x'[t]^2; V = 1/2 k x[t]^2;
eq = ∂t (∂x'[t] (T - V)) - ∂x[t] T + ∂x[t] V == 0
```
output:
```
k x[t] + m x''[t] == 0
```

3.2. Graphics

The graphics capabilities of contemporary CAS are usually on a very high level. Visually attractive images created using Mathematica are easy to find on the internet. The Mathematica graphics engine is quite powerful and comprises a wide range of plotting and image manipulation functions, but some intuitive operations require a rather particular approach. Fine tuning of plots, annotating and the composition of several graphs into one picture in Mathematica is possible, but for the inexperienced user it is often preferable to use an external graphics editor. Moreover, the syntax of graphics commands has changed from version to version quite often.

Table 1: Selected features of the Wolfram Language

	input	output
Prefix, infix or postfix notation:	`{Factorial[5], 5!, 5 // Factorial}`	`{120, 120, 120}`
User definable symbols and operators:	`a_List_i := a[[i]]; s = {2, 3, 4}; s₂`	`3`
Pure functions:	`2^# &[8]`	`256`
Working with lists:	`Map[f, {a, b, c, d, e}]` `f /@ {a, b, c, d, e}`	`{f[a], f[b], f[c], f[d], f[e]}`

To illustrate the plotting performance a very simple plot of the solution to Eq. (1) is presented in Fig. 1. Using a single command, the plotted curve changes colour and style from solid black to dashed red for sections with curvature higher than 0.5.

```
nsol = First@NDSolve[{eq /. {m → 1, k → 1}, x'[0] == 0, x[0] == 1}, x, {t, 0, 10}];
Curvature[t_] := (Abs[x''[t]] / (1 + x'[t]^2)^(3/2) /. nsol);
Show[Plot[If[#[[1]], x[t] /. nsol], {t, 0, 7.15},
    PlotStyle → #[[2]] & /@ {
   {Curvature[t] > .5, {Dashed, Red}},
   {Curvature[t] <= .5, Automatic}}]
```

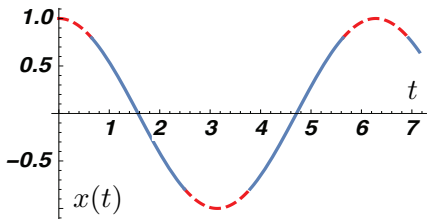

Figure 1: Sample plot of $x(t)$ in Eq. (1).

3.3. Numerical Computation and Parallelization

The majority of AMD and Intel processors contain several cores. Thus, the hardware for SMP-based parallel processing is easily available. Apart from a specialized version for grid computing, Wolfram Mathematica includes parallel versions for most naturally paralellizable commands, e.g., `ParallelTable` or `ParallelSum`, and it also provides support for GPU computation based on CUDA or OpenCL.

Regarding numerical computation, there are two reasons the performance of general CAS is weaker compared to traditional programming. The first originates from the interpreting character of the programming language; this effect is common for all interpreted systems. The second one stems from symbolic evaluation. Computation involving exact values always requires more time, see Table 2. Both problems can be avoided within the Mathematica environment using the command `Compile`. This compiles the numerical expression into machine code and provides processing speed without any additional overhead. When necessary, an SIMD paralellization evaluating the function for a complete list is a matter of a single option. The concept of external linkable procedures, similar to .mex files in Matlab, is available as well.

3.4. Low Level Code Generation

A natural extension of the symbolic manipulation capabilities is the ability to convert the results to different output formats. The conversion to TEX allows the user to directly typeset the resulting expression, and the commands `CForm[]` or `FortranForm[]` convert their argument to a C or Fortran code fragment. This feature is particularly useful because it eliminates typing errors.

4. Selected Problems

At first glance it appears that everything in the realm of technical mathematics is instantly available and that a thorough knowledge is not necessary. However, experience shows that the opposite is true. Consider a simple integration example:

(a) input: $\int_0^{\frac{2\pi}{\omega}} \text{Cos}[n\,\omega\,x]\,dx$ // Simplify — output: $\frac{\text{Sin}[2\,n\,\pi]}{n\,\omega}$

(b) $\text{Simplify}\left[\int_0^{\frac{2\pi}{\omega}} \text{Cos}[n\,\omega\,x]\,dx, \text{Assumptions} \to n \in \text{Integers}\right]$ — output: 0

In case (a), Mathematica is not concerned with the particular value of parameter n, whereas in certain other cases it is. The result is obviously correct, but the specific solutions for the particular values of n may

Table 2: Timing of symbolic and numerical operations

symbolic operation	timing	numerical operation	timing
$\sum_{k=0}^{100\,000} \text{Sin}\left[\frac{1}{100\,000}\,k\,\pi\right]$	1.251s	$\sum_{k=0}^{100\,000} \text{Sin}[0.00001\,k\,\pi]$	0.013s

remain hidden in more complicated expressions. The user must always know what the computer is doing and what is to be expected. Without exact knowledge one could end up with false or incomplete results.

Users also quite often copy and paste expressions obtained from a CAS into their publications. In this way readers may encounter unnecessarily complicated or even misleading formulas. The following fragment is taken from a paper submitted to a scientific journal; ω_o is frequency and D is a positive constant:

$$\frac{1}{D}\left(-D^{\frac{3}{2}}\sqrt{\frac{3\zeta_3\omega_0^4}{D}}+D\sqrt{3\zeta_3\omega_0^4}\,\mathrm{erf}\left(\frac{1}{8}\frac{K_0}{\sqrt{3D\zeta_3\omega_0^4}}\right)\right) \tag{2}$$

It is apparent that the author of Eq. (2) did not try to understand nor simplify his result.

5. Conclusions

Computer algebra systems allow researchers to concentrate on the core of their theories or projects. If properly used, they can significantly simplify the computationally demanding aspects of research. Their primary area of use seems to be non-routine derivations and computations. Their role in verification of particular procedures is undoubtedly indispensable. Mastering of such advanced tools places high demands on their users. The programming language of Wolfram Mathematica is conceptually different from traditional programming languages and, consequently, less understandable for technical experts. On the other hand, it enables a faster description and solution of mathematical and general computational problems, it helps eliminate certain types of errors and it allows instant verification of results.

Acknowledgement

The author would like to thank the developers of Wolfram *Mathematica*® for 25 years of their software helping his scientific work. The institutional support RVO 68378297 in these years is gratefully acknowledged. The licence, upgrades and support fees for the software were partially covered by various Czech Science Foundation projects, the most recent being No. 19-21817S. The author declares no personal profit from preferring, referring to or promoting Wolfram *Mathematica*® or any other software.

References

GeoGebra.org (2019) GeoGebra — free math apps. Available at: https://www.geogebra.org. Accessed: 2019-04-01.

Hearn, A. C. (2005) Reduce: The first forty years. In Dolzmann, A., edt., *Algorithmic Algebra and Logic: Proceedings of the A3L 2005, April 3 - 6, Passau, Germany*. Available at: http://www.reduce-algebra.com/reduce40.pdf

Mangano, S. (2011) *Mathematica cookbook*. O'Reilly, Beijing.

Maplesoft.com (2019a) Maple — software for mathematics, online learning, engineering. Online, available at: https://www.maplesoft.com. Accessed: 2019-04-01.

Maplesoft.com (2019b) Maple product history. Available at: https://www.maplesoft.com/products/maple/history/. Accessed: 2019-04-01.

PTC.com (2019) PTC Mathcad. Available at: https://www.ptc.com/en/products/mathcad/. Accessed: 2019-04-01.

Trott, M. (2006) *The Mathematica Guidebook for Symbolics*. Springer, New York.

Trott, M. (2007) *The Mathematica Guidebook for Numerics*. Springer, New York.

Trott, M. (2013) *The Mathematica Guidebook for Programming*. Springer, New York.

Trott, M. and Trott, M. (2017) *The Mathematica GuideBook for Graphics*. Springer, New York.

Wikipedia contributors (2019a) List of computer algebra systems — Wikipedia, the free encyclopedia. Online, available at: https://en.wikipedia.org/w/index.php?title=List_of_computer_algebra_systems&oldid=892811417. Accessed: 2019-04-01.

Wikipedia contributors (2019b) Mathcad — Wikipedia, the free encyclopedia. Online, available at: https://en.wikipedia.org/w/index.php?title=Mathcad&oldid=890586127. Accessed: 2019-04-01.

Wikipedia contributors (2019c) MuPAD — Wikipedia, the free encyclopedia. Online, available at: https://en.wikipedia.org/w/index.php?title=MuPAD&oldid=884753278. Accessed: 2019-04-01.

Wolfram, S. (1988) *Mathematica: A System for Doing Mathematics by Computer*. Addison-Wesley Publishing Company, Redwood City, first edition.

Wolfram, S. (2019) An elementary introduction to the Wolfram language, second edition. Online, available at: https://www.wolfram.com/language/elementary-introduction/2nd-ed . Accessed: 2019-04-01.

WolframAlpha.com (2019) WolframAlpha®: Making the world's knowledge computable. Online, available at: https://www.wolframalpha.com/. Accessed: 2019-04-01.

Wolfram.com (2019) Mathematica — Three decades of contributions, invention, discovery, and education. Online, available at: http://www.wolfram.com/mathematica/scrapbook/. Accessed: 2019-04-01.