

WHY COMPUTATIONAL TEACHING? THE ADVANTAGE OF USING THE MAPLE SOFTWARE IN TEACHING EXACT SCIENCES

ADELA IONESCU *

ABSTRACT. The computational thinking becomes more and more an essential part of modern teaching. In this context, the resources, including the collection of online lessons plan, videos and other resources were created to provide a better understanding of CT. But the most important are software resources, because of their flexibility and wide application tools. The aim of the present paper is to have an introduction trip focusing on the advantages of using powerful software like MAPLE in computational teaching. Some examples are focused on representing the objects and basic algebraic manipulation commands, very successfully used in teaching sciences.

1. Introduction

First released in 1981 by Waterloo Maple, Inc., Maple is a system for doing mathematics on a computer (<http://www.maplesoft.com/products> n.d.). Maple combines symbolic manipulation, numerical mathematics, outstanding graphics, and a sophisticated programming language. Because of its versatility, Maple has established itself as the computer algebra system of choice for many computer users including commercial and government scientists and engineers, mathematics, science, and engineering teachers and researchers, and students enrolled in mathematics, physics, and engineering courses. However, due to its special nature and sophistication, beginning users need to be aware of the special syntax required to make Maple perform in the way intended.

One can find just on the Maple site ¹ top ten reasons for which students must choose this soft in their education.

- Maple costs less than many scientific calculators, is far more functional, and will grow with you as you go on to take more advanced courses.
- Maple includes thousands of features and commands that make it useful in all of your math, engineering and science classes, from high-school and beyond.

¹<http://www.maplesoft.com/products>

- Maple's Clickable Calculus capabilities allow even novice users to perform complex operations without knowing any special commands or syntax.
- Thousands of commercial organizations, government research labs and academic institutions use it as a core part of their toolchain, so using Maple now will help you get a head-start on your career.
- Built-in tutors and task templates offer step-by-step and visual learning environments for many important math topics, including concepts in precalculus and calculus.
- Via MapleCloud, you can access interactive tutors and calculators 24/7 from your computer or mobile device.
- A continually growing library of Quick Start Tutorials, training videos, and recorded webinars make it easy to master the skills that will help you succeed in the studies.
- Have a question or a problem? Help is always just a click away via MaplePrimes.com, or Maple's extensive online help documentation.
- You get access to thousands of free Maple applications, examples and tips from the Maple Application Center.
- Hundreds of Math Apps make it easy to visually explore a broad set of fundamental concepts from functions and graphing, calculus, statistics, physics, and more.

One of the basic mathematical domains which help solving problems in all other mathematical (and not only) sciences is algebra. With the computational algebra, we can concentrate more on the ideas (the model and its formulation) instead of on the algebraic manipulations; we can explore the mathematics surrounding the problem and easily extend the results.

After learning the basics, the speed at which algebraic results are obtained with the computer compensates with dramatic advantage the extra time invested to formulate the problem in the computer. This is why we can say the computer algebra can enrich the student's working experience in a significant way.

Thus, learning computing algebra is just learning another language to do the same computations we do using pencil and paper:

- It is our mathematical / physical language, that we understand and use doing computations with paper and pencil;
- We can see it like a "dictionary" (help pages) that translate the computer algebra language into our mathematical / physical language;
- In the computer algebra language we want to express those algebraic computations we do with paper and pencil using our mathematical / physical language.

For these purposes we can state that the computer algebra language is as close as possible to our mathematical / physical language. We can access the power of Maple computational engine through a variety of interfaces.

1. *Standard worksheet.* This is a full-featured graphical user interface offering features that help create electronic documents that show all your assumptions, the calculations, and any margin of error in your results; or hide the computations to allow the reader to focus on the problem setup and final results. The advanced formatting features help you

create the customized document needed. Because the documents are *live*, we can edit the parameters and, with the click of a button, compute the new results.

2. *Classic worksheet*. This is Basic worksheet environment for older computers with limited memory.
3. *Command-line version*. This is command-line interface, without graphical user interfaces features, for solving very large complex problems or batch processing with scripts.

2. Some basic appliances in MAPLE

2.1. Introduction in the MAPLE language. In the first part of this section, we present the basic of computing algebra in teaching with Maple: some elementary functions. The table 1 synthesizes the basic operators, the basic elementary functions and also some important manipulation commands.

Operators	$+, -, *, /, ^$
Functions	exp, ln, sin, cos, tan, csc, sec, cot, arcsin, arccos, arctan, arccsc, arcsec, arccot. For the hyperbolic functions put an h at the end as in sinh, arctanh, etc.
Manipulation commands	Related to numerical evaluation: <i>evalf</i> , <i>Digits</i> . The complex components: <i>Re</i> , <i>Im</i> , <i>conjugate</i> , <i>abs</i> , <i>argument</i> Related to functions: series, convert (any function to any other one when possible), <i>FunctionAdvisor</i> Related to plotting: <i>plot</i> , <i>plot3d</i> , <i>plots:-plotcompare</i>

TABLE 1. Arithmetic operations and elementary functions.

Maple is very rich in all types of functions. A function application is represented with rounded parenthesis (), like in $f(x)$. Indexation, as used in tensors, is represented with squared brackets [], as in $A[\mu]$, displayed as A_{μ} for A_{μ} .

Numerical approximation is obtained applying *evalf*. The two functions *evalf* and *FunctionAdvisor*, need a special attention.

In all the following examples throughout the paper, it is exposed the command and the result produced by it:

- A. *Evalf*(Pi):
3.141592654
- *Digits*:
10
- *Evalf*[50](Pi):
3.1415926535897932384626433832795028841971693993751
- B. > *FunctionAdvisor*()
The usage is at follows.
- *FunctionAdvisor*(topic, function, ...):
where 'topic' indicates the subject on which advice is required, 'function' is the

name of a Maple function, and '...' represents possible additional input depending on the topic chosen. To list the possible topics:

- *FunctionAdvisor(topics)*.

A short form usage is:

- *FunctionAdvisor(function)*

where just the name of the function is also available and it displays a summary of information about the function.

With “*FunctionAdvisor*” we can call for use special classes of functions, depending on the aim of the calculations. For example

- *FunctionAdvisor(classes)*

[*trig, trigh, arctrig, arctrigh, elementary, GAMMA_related, Psi_related, Kelvin, Airy, Hankel, Bessel_related, OF1, orthogonal_polynomials, Ei_related, erf_related, Kummer, Whittaker, Cylinder, IF1, Elliptic_related, Legendre, Chebyshev, 2F1, Lommel, Struve_related, hypergeometric, Jacobi_related, InverseJacobi_related, Elliptic_doubly_periodic, Weierstrass_related, Zeta_related, Other, Bell, Heun, trigall, arctrigall, piecewise_related, complex_components, integral_transforms*]

and we get in this way a large panel of special functions for different use. We have to mention on one hand, that this is a small part of the very few functions we can find in Maple, and on the other hand, the advantage of this command in Maple.

In what follows the Table 2 presents algebraic expressions, equations and functions which are basic in teaching (Abell and Braselton 2005).

Algebraic expression	any mathematical object built with numbers, symbols and functions combined using arithmetic operations
Equation	a construction using the = sign, typically with algebraic expressions on the left-hand and right-hand sides
Function	It can be a known function as $\ln(z)$ or an unknown function, for example $f(x,y,z,t)$
Mapping	maps variables into constructions that involve these variables, typically algebraic expressions, for example $f := (x, y, z) \rightarrow \sqrt{x^2 + y^2 + z^2}$
Manipulation commands	<ul style="list-style-type: none"> - To represent function application use $()$, as in $f(x)$ - To construct expressions, equations and mappings use: $=$, $:=$, $->$, and <i>unapply</i> to convert an expression into a mapping - Related to expressions: <i>numer</i>, <i>denom</i>, <i>collect</i>, <i>coeff</i>, <i>degree</i> - Related to equations and inequations: $=$, $<>$, $<=$, $>=$ and to get each side use <i>lhs rhs</i> (from left hand side, right hand side) - Basic manipulation of expressions / equations: <i>subs</i>, <i>eval</i>, <i>map</i>, <i>collect</i>, <i>isolate</i>, <i>solve</i>

TABLE 2. Algebraic expressions, equations and functions.

The following examples show how fast and comfortable is to use these simple but basic commands.

Let us note that always is good to use “*restart*” in order to tell the soft not to use the previous operations / calculus.

- *restart*

$$ax^2 + \frac{e^x}{b} \quad (1)$$

We can give any name to an expression (assign a name to it) in order to refer to it, and also if the expression is not displayed. We do that by using the assign operator :=

- $f := (1)$

$$f := ax^2 + \frac{e^x}{b} \quad (2)$$

Now we can refer to the expression (1) using the given name

- $f;$

$$ax^2 + \frac{e^x}{b} \quad (3)$$

Different from an expression, an equation always has left and right-hand sides with the ‘=’ operator in between. For example

- $f = 0;$

$$ax^2 + \frac{e^x}{b} = 0 \quad (4)$$

Now we can assign names to everything, also to an equation:

- $h := f = g$

$$h := ax^2 + \frac{e^x}{b} = g \quad (5)$$

What we call “the function of x equal to...” is implemented in the computer as a *mapping*, using the arrow operator \rightarrow ,

- $x \rightarrow (1);$

$$x \rightarrow ax^2 + \frac{e^x}{b} \quad (6)$$

To use a mapping it is also practical to assign a name to it:

- $h := x \rightarrow (1);$

$$h := x \rightarrow ax^2 + \frac{e^x}{b} \quad (7)$$

Note however that the *mapping* h not really a function of x , but also of whatever argument you pass to it, as in

- $h(y)$

$$ay^2 + \frac{e^y}{b} \quad (8)$$

- $h(\alpha)$

$$a\alpha^2 + \frac{e^\alpha}{b} \quad (9)$$

We can convert an expression or equation into a mapping using “unapply”:

- $unapply(\%, x);$

$$x \rightarrow ax^2 + \frac{e^x}{b} \quad (10)$$

- $unapply((1), x, a, b);$

$$(x, a, b) \rightarrow ax^2 + \frac{e^x}{b} \quad (11)$$

Of course we can solve expressions or equations or systems of them. When solving, an expression is considered an equation with right-hand side equal to zero.

- $f;$

$$ax^2 + \frac{e^x}{b} \quad (12)$$

- $isolate(f, b);$

$$b = -\frac{e^x}{ax^2} \quad (13)$$

- $solve(f, b);$

$$\left\{ b = -\frac{e^x}{ax^2} \right\} \quad (14)$$

It is important to note that “isolate” however only returns one solution. To get all the solutions use “solve” as follows:

- $isolate(f, x);$

$$x = -2W\left(-\frac{\sqrt{-\frac{1}{ab}}}{2}\right) \quad (15)$$

- $solve(f, \{x\});$

$$\left\{ x = -2W\left(-\frac{\sqrt{-\frac{1}{ab}}}{2}\right) \right\}, \left\{ x = -2W\left(\frac{\sqrt{-\frac{1}{ab}}}{2}\right) \right\} \quad (16)$$

It is easy to observe that we have in fact a *family of solutions*. For a complete solving we use

- $Solve(f, \{x\}, AllSolutions);$

$$\left\{ x = -2W\left(-Z2, -\frac{\sqrt{-\frac{1}{ab}}}{2}\right) \right\}, \left\{ x = -2W\left(-Z3, \frac{\sqrt{-\frac{1}{ab}}}{2}\right) \right\} \quad (17)$$

- C. The *LambertW* function.

Like *evalf* and *FunctionAdvisor*, the *LambertW* function needs a special attention because of its importance. This function satisfies the relation:

$$LambertW(x)e^{LambertW(x)} = x \quad (18)$$

Since the equation $ye^y = x$ has an infinite number of solutions y for each (non-zero) value of x , **LambertW** has an infinite number of branches. Exactly one of these branches is analytic at 0. In Maple this branch is referred to as the *principal branch of LambertW*, and is denoted by **LambertW(x)**. The other branches all have a branch point at 0, and these branches are denoted in Maple by **LambertW(k,x)**, where k is any non-zero integer. More details on it can be found in Korless *et al.* (1996).

2.2. Some basic algebraic manipulations on matrices. What we generally call “simplification” is not really a well-defined operation, but is one based on the common sense, and the desired result sometimes depends on particularities of the problem. Also, it is well-known that this operation can be often difficult – for example in engineering and physics. The Table 3 is significant for the synthesis of the MAPLE commands and manipulations.

Commands	Matrix, Vector is the same as Vector[column], Vector[row], or matrix and vector. Use + and . for operations
Manipulation commands	<i>LinearAlgebra</i> package: conjugate, Transpose, HermitianTranspose, Determinant, Trace, Eigenvalues, Eigenvectors, MatrixExponential, LinearSolve <i>linalg</i> package: conjugate, transpose, htranspose, det, trace, eigenvalues, eigenvectors, exponential, linsolve

TABLE 3. Linear algebra package.

The *LinearAlgebra package* is rich, with 130 commands to manipulate Matrices and solve linear algebra problems. The other package, *linalg*, is older and contains 114 matrix algebra commands. We can do operations with both packages, but the new one is faster. In what follows we restrict to a small subset of matrix commands that are used more frequently.

In Abell and Braselton (2005) can be found more details and examples.

For historical and other reasons, there are two kinds of matrices in Maple:

- a) The old ones, represented by the lowercase word *matrix* have the advantage that we can compute with them without displaying their contents.
 b) The new ones, represented by the word *Matrix* have the advantage of performing component computations faster.

a) The case of *matrix*

- $A := \text{matrix}(2, 2, [a, b, c, d]);$

$$W = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \quad (19)$$

Invoking the matrix does not show its components.

- $A;$

$$A \quad (20)$$

We can refer to an unspecified component (this is useful when setting brackets rules in Quantum Mechanics), as in

- $A[i, j];$

$$A[i, j] \quad (21)$$

We can also specify the component of the matrix to visualize:

- $\text{eval}((21), [i = 1, j = 2]);$

$$b \quad (22)$$

b) The same happens for *Matrix*

- $B := \text{Matrix}(2, 2, [a, b, c, d]);$

$$B = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \quad (23)$$

But, we cannot refer in this case, to an unspecified component:

- $B[i, j];$
`Error, bad index into Matrix!!`

So, the *LinearAlgebra* package is all about *Matrix*, while there also exists the old *linalg* package about *matrix*. So we can do operations with both packages according to whether we need more symbolic capabilities (*linalg*) or faster computations (*LinearAlgebra*).

Vectors can be represented using the *vector* and *Vector* commands.

We can also specify the component of the matrix to visualize:

- $v := \text{vector}([v_a, v_b]);$

$$v := [v_a \ v_b] \quad (24)$$

- v ;

$$v \quad (25)$$

- $v[j]$;

$$v_j \quad (26)$$

- $v[2]$;

$$v_b \quad (27)$$

- $V := \text{vector}([V_a, V_b]);$

$$V := \begin{bmatrix} V_a \\ V_b \end{bmatrix} \quad (28)$$

When using *Matrix* and *Vector*, summation and product are performed using '+' and '·'. When using *matrix* and *vector*, it is the same but we need to enclose the operation with *evalm*.

- vA ;

$$v \cdot A \quad (29)$$

- $\text{evalm}((24));$

$$[v_a \cdot a + v_b \cdot c \quad v_a \cdot b + v_b \cdot d] \quad (30)$$

Is important to note that for *Vector* there are row and column vectors, so

- VB ;

Error, (in LinearAlgebra:-Multiply) cannot multiply a column Vector and a Matrix

The typical operations: *conjugate*, *Transpose*, *HermitianTranspose*, *Determinant*, *Trace*, *Eigenvalues* work also easy (Kurosh 1980; Abell and Braselton 2005).

3. Applications. Working with tensors

It is well-known that superior physics – and let's consider tensors – works with indexed objects, and that's why the calculus is often heavy. Maple comes and makes the life easier for those who choose to use the algebraic appliances to represent and solve problems.

The Table 4 below presents the basics for algebraic representation and subsequent commands in Maple.

Algebraic Representation	Indexed objects, as $A[x]$, can include functionality as in Ax, must be defined using the <i>Physics:-Define</i> command and can have (anti)symmetry properties of any particular kind regarding permutations of indices. We can also define tensors using tensorial equations
Commands	<i>Physics:-Define</i> , <i>Physics:-Simplify</i> , and <i>Physics:-`</i> to "multiply and simplify" in one step - useful to directly perform a contraction instead of just representing it.

TABLE 4. Tensors.

- *restart*
- *with(Physics)*
- *Setup(mathematicalnotation = true)*

$$[\textit{mathematicalnotation} = \textit{true}] \tag{31}$$

To define a system of references (coordinates system) and the related space-time vector we can use *Setup* or *Coordinates*.

- *Coordinates(X)*;
Default differentiation variables for d_, D_ and dAlembertian are:

$$\{X = (x1, x2, x3, x4)\} \tag{32}$$

Systems of spacetime Coordinates are:

$$\{X = (x1, x2, x3, x4)\}$$

$$\{X\}$$

We can now use X to represent function dependency, as in $F(X)$, equivalent to writing $F(x1, x2, x3, x4)$, and also as the spacetime vector X^{mu} (see *SpaceTimeVector* package). To indicate that an index is contravariant, prefix it with \sim (tilde):

- $X[\sim mu]$, $X[mu]$;

$$X^\mu, X_\mu \tag{33}$$

The label X can also be used to select each of the components of the space-time vector:

- $X[1]$, $X[3]$;

$$x1, x3 \tag{34}$$

and $x0$ is always mapped into $x4$:

- $x0$;

$$x4$$

We can also set the coordinates to be any sequence of four names. Three predefined sets are 'cartesian', 'cylindrical' and 'spherical'.

- *Coordinates(X = cartesian);*

Default differentiation variables for d_ , D_ and dAlembertian are:

$$\{X = (x,y,z,t)\} \tag{35}$$

Systems of spacetime Coordinates are:

$$\{X = (x,y,z,t)\}$$

$$\{X\}$$

We can set many systems of coordinates simultaneously - although only one is considered the 'default differentiation variables' for the d_ dAlembertian and D_ operators (and that is the system of references were all the general relativity tensors are defined).

- *Coordinates(Y)*

Systems of spacetime Coordinates are:

$$\{X = (x,y,z,t), Y = (y1,y2,y3,y4)\} \tag{36}$$

$$\{X,Y\}$$

- *Setup(diff);*

Partial match of 'Physics:-diff' against keyword 'differentiationvariables'

Default differentiation variables for d_ , D_ and dAlembertian are:

$$\{X = (x,y,z,t)\} \tag{37}$$

$$[differentiationvariables = [X]]$$

We can change the differentiation variables, for this - enter for instance **Setup(diff = Y)**;

The default dimension of space-time is

- *Setup(dimension);*

$$[dimension = 4] \tag{38}$$

To change the value of the spacetime dimension use **Setup(dimension = N)** where *N* is any positive integer greater than 1.

To see the values of the components of the metric you can enter the spacetime metric *g_* without indices; when we load the package it is of Minkowski type:

- *g_[];*

$$g_{\mu,\theta} = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \tag{39}$$

The space-time indexed differentiation operator is obtained by the command:

- $d_{[mu]}(F(X));$

$$\partial_{\mu}(F(X)) \quad (40)$$

This operator is also a representation for the *differential* of a function; for that purpose we enter it without indices:

- $d_{-}(F(X));$

$$\left(\partial_{\mu}(F(X))\right)\left(\partial(X^{\mu})\right) \quad (41)$$

Using d_{-} you can express the differential of any coordinate defined using *Coordinates*. It is also important to note that the *diff* and d_{-} operators are interconnected:

- $diff(F(X), X[\sim mu]);$

$$\partial_{\mu}(F(X)) \quad (42)$$

- $diff(F(X), X[mu]);$

$$\partial^{\mu}(F(X)) \quad (43)$$

By applying it two times you get the dAlembertian:

- $d_{[mu]}(d_{[mu]}(F(X)));$

$$\square F(X) \quad (44)$$

4. Conclusion

The present paper has realized an introductory review through the most powerful module of the Maple soft, the computational algebra module. We hope to make more familiar the computer algebra to all who want to use it in physics and applied sciences. Computer algebra can transform the algebraic computation part of physics into interesting discoveries. Of course this is only a small part; the beauty of this software is still to be explored, in order to discover how Maple enhances the learning experience for both teachers and students.

There are many ways to bring Maplesoft technology to our math courses, from adding new energy to traditional lectures, to fundamentally transforming the curriculum to reflect a new, innovative teaching style:

- *Increase student understanding* with illuminating explanations and visualizations that help make math more tangible;
- *Help students build confidence* through interactive explorations and immediate feedback;
- *Keep students engaged* with motivating examples and real-world applications that would be too difficult or time-consuming to do by hand.

References

- Abell, M. and Braselton, J. (2005). *Maple by example, 3rd Edition*. Amsterdam Boston: Elsevier Academic Press, San Diego, California.
<http://www.maplesoft.com/products> (n.d.).
- Korless, R. M., Gonnet, G. H., Hare, D., Jeffrey, D., and Knuth, D. (1996). "On the Lambert W Function". *Advances in Computational Mathematics* **5**, 329–359.
- Kurosh, A. (1980). *Higher algebra*. Mir Editions, Moscow.

* University of Craiova
Department of Applied Mathematics
Al. I. Cuza st., 13, Craiova, RO-200585, Romania

Email: adelajaneta2015@gmail.com

Paper contributed to the international workshop entitled "New Horizons in Teaching Science", which was held in Messina, Italy (18–19 november 2018), under the patronage of the *Accademia Peloritana dei Pericolanti*

Manuscript received 06 June 2020; published online 30 September 2021



© 2021 by the author(s); licensee *Accademia Peloritana dei Pericolanti* (Messina, Italy). This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/) (<https://creativecommons.org/licenses/by/4.0/>).