

Engineering Mathematics and Computer Algebra

T.J. Fairclough,
Department of Electrical and Electronic Engineering,
Institut Teknologi Brunei.

Abstract.

Computer Algebra software packages for microcomputers provide an opportunity to extend the range and complexity of the material normally offered in traditional Engineering Mathematics courses. This article reviews the typical capabilities of Computer Algebra systems and discusses some of their uses in the teaching of Engineering Mathematics.

1. Introduction

The development of Computer Algebra systems began in the 1960's on large mainframe computers, they were used mainly by research scientists to simplify calculations in general relativity, celestial mechanics and quantum mechanics. These systems were able to manipulate symbols as well as numbers and made it possible for digital computers to carry out the algorithms of algebra as readily as those of arithmetic.

An outstanding example of the successful application of symbolic techniques to a problem in algebra occurred in 1970 when scientists at the Boeing Scientific Research laboratories in Seattle reproduced, in about twenty hours of computer time, Delauney's immense calculation of the motion of the moon around the earth under the perturbing influence of the sun's gravitational field. These calculations had originally been carried out by hand over a twenty-year period, starting in 1847, by the French astronomer Charles Delauney. His original results had been published in two volumes and remained unchallenged for more than a hundred years. The computer discovered one major error in volume II of Delauney's work which gave rise to two consequent minor errors. Delauney's corrected calculation can now be used to test new computer algebra systems for accuracy and speed.

During the 1970's and early 1980's the availability of computer algebra systems remained largely restricted to those users having access to mainframes but some progress was made on systems developed specifically for personal computers. On mainframe systems the program Macsyma was developed at M.I.T. from 1969 onwards and in the late 1970's the less powerful program muMATH (succeeded by Derive) became available for microcomputers. In the 1980's along with the widespread availability of personal computers came a wide range of new computer algebra packages and today such systems as Derive, Maple, Reduce, Mathematica and the microcomputer version of Macsyma are potentially as much a part of the Engineer's mathematical toolkit as an electronic calculator.

The existence of computer algebra packages has implications for the teaching of courses in Engineering Mathematics, however, just as a numerical calculator is no use to a person not understanding the difference between multiplication and division, a computer algebra system still requires the user to understand the meaning of the processes the system carries out i.e. factorisation, differentiation, integration, Fourier and Laplace transforms, Taylor series etc. In this respect it is still necessary to teach appropriate mathematical methods but the use of a computer algebra package makes it easier to get the right answer and makes it feasible for the student to tackle more complex problems.

At Institut Teknologi Brunei students in Electrical and Electronic Engineering are introduced to a computer algebra package in the first semester of Engineering Mathematics and they continue its use in subsequent semesters. The package is treated as a 'mathematical assistant' and is a useful tool for enhancing the student's mathematical scope.

2. What Computer Algebra Can Do.

Computer algebra systems are not only useful for sophisticated calculations but may also be used to replace pencil-and-paper methods for more straightforward non-numerical calculations; most computer algebra systems can be used as a symbolic calculator where, by issuing a simple command, a moderately difficult calculation such as a symbolic (or numerical) integration can be carried out. Moreover it is quite likely that the system used will incorporate more methods and branches of mathematics than any one user knows, since it is effectively an Expert System.

When used as a symbolic calculator the question must first be typed in and submitted for evaluation, the answer is displayed as soon as it has been obtained. In some cases no answer can be obtained and the machine may eventually return the original question as the answer. The command syntax varies according to the package used, for example to evaluate the indefinite integral

$$\int \cos(x)dx$$

the following commands must be entered:

Macsyma	<code>integrate(cos(x),x)</code>
Mathematica	<code>Integrate[Cos[x],x]</code>
Maple	<code>int(cos(x),x)</code>
Reduce	<code>int(cos(x),x)</code>
Derive	<code>INT(COS(x),x)</code>

In each case the answer is returned as the function $\sin(x)$.

The basic mathematical facilities available on most packages typically include:

Arithmetic:	Exact and approximate rational and complex arithmetic. Control of radix base. Floating point calculations to any accuracy specified by the user.
Algebra:	Manipulation, expansion and factorisation of polynomials. Simplifying expressions. Partial fraction expansions. Substituting finite and infinite values. Defining functions of several variables. Exact solution of quadratic, cubic and quartic polynomial equations. Approximate solution of algebraic equations including linear simultaneous equations.
Calculus:	Summation and products over infinite and indefinite ranges. Derivatives of order n and partial derivatives. Definite, indefinite and multiple integrals. Special functions. Solution of ordinary differential equations.
Graphics:	Graph plotting in two and three dimensions.
Matrices:	Vector and matrix operations and algebra. Eigenvalues and eigenvectors. Inverses.

The more general purpose powerful systems, such as Macsyma, Mathematica, Maple and Reduce require machines with several megabytes of random access memory (RAM), take up several more megabytes of Hard-Disk space and offer more facilities than those outlined above. Derive, on the other hand, requires only 512 kilobytes of RAM and can be run from a single

floppy disk. Despite its small size, Derive is a flexible and reasonably powerful tool but for more difficult problems or advanced use the more powerful systems are required. In the next section we shall describe some of the basic features of a symbolic calculator using the Mathematica system as an example.

3. Mathematica.

Mathematica [1] is a system for doing mathematics by computer developed in USA by Wolfram Research Inc. It has a front end, which handles interaction with the user, and the kernel, which actually carries out the calculations. In a typical session dialogue the front end generates input line prompts; the n th input line starts with the prompt 'In[n]:= ', this is not typed in by the user and n is incremented automatically as the session proceeds. After typing the question on the input line the user presses Shift-Enter to submit the question to the kernel. When the calculation is complete the answer is printed on the screen as an output line starting with 'Out[n]: = ' and Mathematica is then ready for the next question.

Some typical manipulations are illustrated below, in each case the question to be answered is given in mathematical terms, this is followed by the appropriate input code and the consequent output rewritten in normal mathematical form:

- (a) Factorise $x^4 + 10ax^3 + 35a^2x^2 + 50a^3x + 24a^4$.

```
In[1]:= Factor[x^4+10*a*x^3+35*a^2*x^2+50*a^3*x+24*a^4]
```

```
Out[1]:= (x+a)(x+2a)(x+3a)(x+4a)
```

- (b) Simplify $10\sec^4(a)\tan(a) + \sec^5(a)\sin(5a) - 5\sec^5(a)\sin(3a)$.

```
In[2]:= Simplify[10*Sec[a]^4*Tan[a]+Sec[a]^5*Sin[5*a]-5*Sec[a]^5*Sin[3*a]]
```

```
Out[2]:= 16tan^5 ( a)
```

- (c) Differentiate the function $-\frac{\sqrt{a^2+x^2}}{x} + \ln(x + \sqrt{a^2+x^2})$ with respect to x .

In general to differentiate y with respect to x type $D[y,x]$, in this example the answer does not come out in its simplest form so we pass the answer through the simplifier by nesting the D and $Simplify$ commands

```
In[3]:= Simplify[ D[ -Sqrt[a^2+x^2]/x+Log[x+Sqrt[a^2+x^2]],x] ]
```

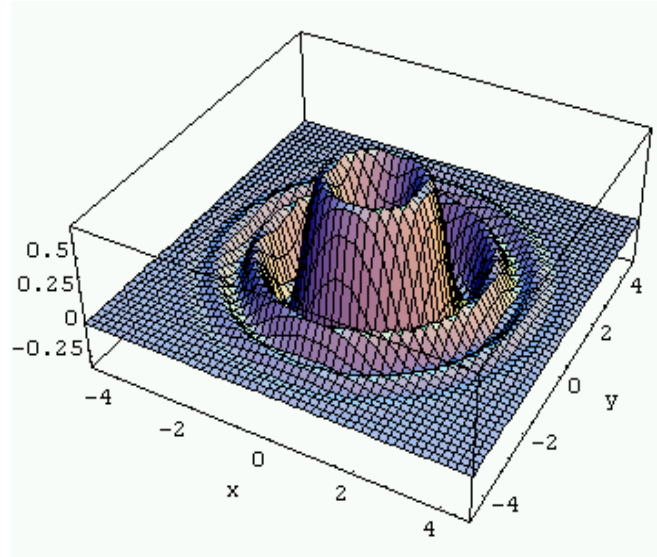
```
Out[3]:=  $\frac{\sqrt{a^2+x^2}}{x^2}$ 
```

- (d) Plot the surface $z = e^{-(x^2+y^2)/4} \sin(x^2 + y^2)$.

For plotting in three dimensions Mathematica has the versatile $Plot3D$ command. After the plot has been obtained using the default viewpoint it can be redrawn and shown again from a different viewpoint specified by the user.

```
In[4]:= Plot3D[E^(-(x^2+y^2)/4)*Sin[x^2+y^2],{x,-3*Pi/2,3*Pi/2},{y,-3*Pi/2,3*Pi/2},
PlotPoints -> 50, PlotRange -> All]
```

Out[4]:=



(e) Evaluate

$$\int_0^{\pi} \frac{[\cos(\frac{\pi}{2} \cos(\theta))]^2 d\theta}{\sin(\theta)}$$

This integral occurs in the calculation of the radiation resistance of a half-wave dipole; it can be manually evaluated only with some difficulty and requires advanced knowledge. In Mathematica the integral may be evaluated analytically by entering the command

```
In[5] := Integrate[(Cos[Pi/2*Cos[theta]])^2/Sin[theta], {theta, 0, Pi}]
```

```
Out[5]:= 1/2*(gamma - Ci(2pi) - 2*ln(1/2) + ln(pi/2))
```

In Out[5] γ is Euler's constant ($\gamma \approx 0.577216$) and $Ci(x)$ is the cosine integral [2], a special function of mathematical physics defined by

$$Ci(x) = \gamma + \ln(x) + \int_0^x \frac{[\cos(t) - 1]dt}{t}$$

To obtain a numerical value for the integral we may use the command `N[%]`; `%` is shorthand for 'the last answer', which in this case is Out[5], and `N[%]` returns the numerical value of this last answer as 1.21883.

```
In[6]:= N[%]  
Out[6]:= 1.21883
```

By using a computer algebra package in this way a student is able to evaluate a complex expression that would normally be out of reach. The necessity for using tables of integrals disappears and the complex mechanics of integration ceases to be a barrier to obtaining the correct answer.

4. Applications in Engineering Mathematics.

Laplace Transforms and Fourier Series are important mathematical tools used by electrical engineers. Mathematica comes with additional packages that, when loaded, allow these tools to be used and considerably ease the burden of calculation as described in the following sections 4.1 and 4.2

4.1. Transient analysis and Laplace Transforms.

Electrical transients are described by differential equations that can be solved by a variety of methods, although engineers frequently solve them using tables of Laplace Transforms. The manual solution procedure can be time consuming and is error prone. With a computer algebra package the differential equation can be quickly solved, either by Laplace Transforms or directly in the time domain. The transient behaviour can be plotted and the maximum overshoot calculated all with a few simple commands and with less chance of error. As an example consider the transient voltage, v in a series L-C-R circuit driven by a voltage step applied at time $t = 0$ and described by the differential equation

$$\frac{d^2 v}{dt^2} + 2 \frac{dv}{dt} + 17v = 34 \quad (1)$$

with the initial condition that both v and $\frac{dv}{dt}$ are zero when $t = 0$.

To solve the equation using Laplace Transforms it is first necessary to load the appropriate package; in Mathematica this is done by entering the command

```
In[7]:= <<Calculus`LaplaceTransform`
```

No output line is generated but the package is loaded, this package allows us to find the Laplace Transform of a function $f(t)$ in terms of s by means of a command of the form `LaplaceTransform[f,t,s]`. As a demonstration the command below obtains the Laplace Transform of the Bessel function $J_0(z)$

```
In[8]:= LaplaceTransform[ BesselJ[ 0, z ], z, s ]
```

```
Out[8]:=  $\frac{1}{\sqrt{1+s^2}}$ 
```

Returning to the problem of solving the differential equation (1) we must first take the Laplace Transform of both sides of (1) using the initial conditions and then solve the resulting algebraic equation to find an expression for the Laplace Transform of the voltage v . This Laplace Transform is then inverted to obtain the solution for v in the time domain. The procedure is accomplished as follows:

```
In[9]:= LaplaceTransform[v''[t]+2*v'[t]+17*v[t]= 34, t, s] /. {v[0] -> 0,v'[0] -> 0}
```

```
Out[9]:= s^2*LaplaceTransform[v[t],t,s]+2 s LaplaceTransform[v[t],t,s] +
17 LaplaceTransform[v[t],t,s] = 34/s
```

```
In[10]:= Solve[%,LaplaceTransform[v[t],t,s] ]
```

```
Out[10]:= {{ LaplaceTransform[v[t],t,s] -> 34/(s*(17+2*s+s^2)) }}
```

```
In[11]:= InverseLaplaceTransform[ 34/(s*(17+2*s+s^2)),s,t]
```

$$\text{Out}[11]:= 2 - 2e^{-t}(\cos(4t) + \frac{1}{4}\sin(4t))$$

We can plot the solution in a frame over a suitable time interval with gridlines using the command

```
In[12]:= Plot[%, {t, 0, 4}, PlotRange -> All, FrameLabel -> {"time", "voltage"},
GridLines -> Automatic, Frame -> True]
```

Out[12]:-

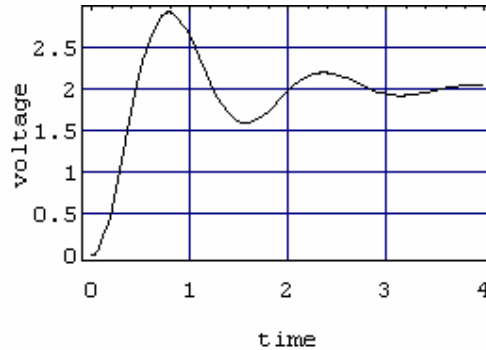


Figure 1. Transient voltage

Figure 1 shows a typical underdamped oscillation that appears to settle at about 2 volts. To confirm this we can use the limit command

```
In[13]:= Limit[Out[11], t -> Infinity]
```

Out[13]:= 2

To find the maximum overshoot we must determine the time at which it occurs, it will be useful to have an expression for the first derivative function

```
In[14]:= Simplify[D[Out[11], t]]
```

Out[14]:= $8.5 e^{-t} \sin(4t)$

By inspection this is first zero when $t = \pi/4$ hence

```
In[15]:= maxovershootpercent = (Out[11] - Out[13])/Out[13]*100 /. t -> Pi/4
```

Out[15]:= 45.6 (percent)

The preceding dialogue illustrates the solution procedure using Laplace Transforms, it is also possible to solve the differential equation directly in the time domain using the special command of the form `Dsolve[equations,variable]`

```
In[16]:= DSolve[{v''[t]+2*v'[t]+17*v[t]==34,v[0]==0,v'[0]==0},v[t],t]
```

Out[16]:= $v[t] \rightarrow 2 - 2e^{-t}(\cos(4t) + \frac{1}{4}\sin(4t))$

In this section we have seen some of the advantages of a computer algebra package over the more traditional pencil-and-paper method. The differential equation has been solved and plotted by means of a few straight-forward commands without the need for tables of Laplace Transforms

and with less chance of error. The session could be saved as a file and edited to be used again for a different equation if required.

4.2 Periodic waveforms and Fourier series.

A periodic waveform may be expressed as a Fourier series containing an infinite number of harmonic components. To obtain the Fourier series manually is time consuming and involves the evaluation of integrals or, sometimes, use of tables. In Mathematica the Fourier Transform package contains functions dealing with a wide range of Fourier methods including the trigonometric Fourier Series. The package is loaded as follows

```
In[17]:= <<Calculus`FourierTransform`
```

As an example we calculate below the first few terms of the Fourier series for a full-wave rectified cosine wave, the manual calculation of this result is tedious but the machine calculation presents no problem

```
In[18]:= FourierTrigSeries[ Cos[ωt], {ωt, -Pi/2,Pi/2}, 3]
```

$$\text{Out[18]} := \frac{2}{\pi} + \frac{4 \cos(2\omega t)}{3\pi} - \frac{4 \cos(4\omega t)}{15\pi} + \frac{4 \cos(6\omega t)}{35\pi}$$

A Fourier series truncated at the n th harmonic is known as the n th partial sum. As the value of n is increased the partial sum approximates more closely to the waveform. This is illustrated below for a sawtooth wave. In the ensuing dialogue two functions are defined by the user, a function named `partialsum[n]`, which calculates the n th partial sum for the sawtooth, and a function called `graph[n]`, which plots the graph of the n th partial sum.

To specify the user -defined function `partialsum[n]` for the sawtooth wave the appropriate syntax is

```
In[19]:= partialsum[n_] := FourierTrigSeries[ x , { x, 0, 1} , n]
```

There is no output line for a definition but we can check that the definition has been accepted by asking for the third partial sum

```
In[20]:= partialsum [3]
```

$$\text{Out[20]} := \frac{1}{2} - \frac{\sin(2\pi x)}{\pi} - \frac{\sin(4\pi x)}{2\pi} - \frac{\sin(6\pi x)}{3\pi}$$

Next we define a function `graph[n]` that plots the graph of the n th partial sum

```
In[21]:= graph[n_] := Plot[Evaluate[partialsum[n]], {x,0,3}]
```

We can use this graph function to synthesise the waveform using $n = 2,3,4$ and 30 harmonics and display the result in a graphics array as follows

```
In[22]:= Show[GraphicsArray[{{graph[2],graph[3]},{graph[4],graph[30]}]]]
```

Out[22]:=

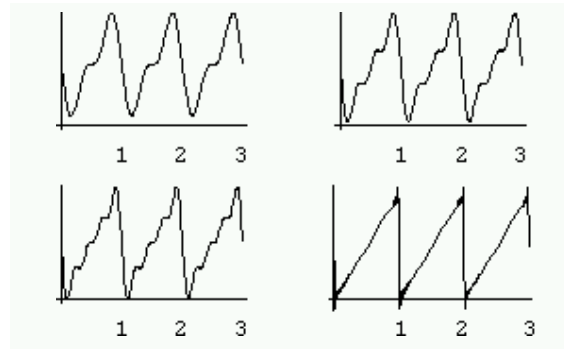


Figure 2. Partial Sums

In Figure 2 the irregularities in the partial sum on the bottom right are due to the Gibbs' phenomenon that occurs in the Fourier Series for discontinuous waveforms.

The preceding dialogue only touches on some of the capabilities of the Fourier Transform package and the usefulness of user-defined functions. It is also possible to program in Mathematica and for an advanced user to write new packages for any specific purpose not already provided.

5. Implications for the Teaching of Engineering Mathematics.

Courses in Engineering Mathematics need to be modified to take advantage of computer algebra as a means of increasing the scope and complexity of the type of mathematical problems that can be included in the course. This must not be done at the expense of basic conceptual content or pencil-and paper methods but rather as a means of reinforcing basic understanding and to eliminate drudgery.

It is advisable that a package be used on a regular basis with frequent exposure in a variety of contexts, otherwise it is all too easy to forget the command syntax and lose proficiency. The use of a computer algebra package adds interest to the classes and enables a student to produce graphical and symbolic output quickly and easily. With a computer algebra package a student can explore mathematical avenues that would otherwise remain unexplored.

There is a need to promote awareness of computer algebra generally. For a variety of reasons, its use does not seem to have spread as quickly as did the use of the numerical calculator and even now many potential users are unaware of the possibilities.

Computer algebra packages do not generally receive the same level of publicity and advertising as some other software applications such as word-processors and spreadsheets for example, but to a would-be user of mathematics they can provide a valuable new perspective and save many hours of work.

6. Conclusion.

Computer algebra is here to stay. Packages are getting cheaper and better all the time and even migrating to hand-held calculators. It is becoming increasingly important for teachers at all levels of the education system to be aware of the potential uses of computer algebra and to make appropriate modifications to their teaching strategies so as to take advantage of these new facilities.

In Engineering Mathematics computer algebra is particularly important and can be used to increase the scope and complexity of the problems tackled by students as well as to enhance their basic understanding of the fundamental principles.

References

- [1] Stephen Wolfram, Mathematica - A System for Doing Mathematics by Computer, Addison-Wesley, 1991.
- [2] M.Abramowitz and I. Stegun, Handbook of Mathematical Functions, Dover Publications, 1972.