# WHATEVER HAPPENED TO COMPUTER ALGEBRA SYSTEMS IN THE ENGINEERING CLASSROOM?

*M. Jankowski*

Department of Electrical Engineering
University of Southern Maine
Portland, ME 040104
email: mjankowski@usm.maine.edu

## ABSTRACT

In the 1980s and early 1990s, computers and computer algebra (CA) systems were used to transform how undergraduate calculus was taught and learned. Experiments in using these systems in electrical engineering followed rapidly. However, present day reality in electrical engineering education and signal processing education in particular, suggests that use of CA systems may be very limited despite significant gains in functionality and performance over the past decade. This paper describes how Mathematica$^{TM}$ is used in teaching an undergraduate linear signals and systems course at the University of Southern Maine. In the process I will highlight some of the present-day capabilities of a modern CA system and demonstrate its useful pedagogic role in teaching and learning electrical engineering.

***Index Terms*** - Computer algebra, *Mathematica*, signal processing education.

## 1. INTRODUCTION

Computer algebra (CA) systems began to appear in the early 1970s, and evolved out of research into artificial intelligence. The first popular systems were Reduce, Derive, and Macsyma. Today, two computer algebra systems that are commonly used by research mathematicians, scientists, and engineers are *Mathematica* and Maple. In the 1980s and early 1990s, as part of a nation-wide calculus reform initiative several major projects were funded with the goal of transforming how undergraduate calculus was taught by integrating computers and CA systems into the teaching process [1, 2]. Some of the stated benefits included ability to consider more complicated problems because students no longer had to do all the mathematical manipulations by hand and the use of visualization to improve understanding of calculus concepts. Clearly, these benefits were equally important in engineering education and so the use of CA systems in electrical engineering followed rapidly [3, 4, 5]. However, present day reality in electrical engineering education suggests that although computers and software may indeed be used widely, the use of CA systems may be very limited. For example, none of the leading textbooks on analog circuits, signals and signals, and digital signal processing make any use of the computer algebra systems available today. From a strictly pedagogic perspective this is a surprising phenomenon considering the extensive amount of algebraic calculations needed in exploring any of the fundamental concepts presented in these textbooks. It is also interesting to note that this phenomenon cannot be explained by any actual limitations of the available software systems.

The typical courses on circuits, signals and systems, and digital signal processing require of the students significant algebraic skills in manipulating a variety of mathematical expressions. In fact, recently published data gathered in the Signals and Systems Concepts Inventory project [6] confirms that a student's mathematical skills are a dominant factor in predicting the student's success in learning key concepts of signals and systems. Conversely, the inability to perform these manipulations may be a significant barrier. As a result, signal and systems textbooks devote whole sections and appendices to such topics as finding solutions to systems of ordinary constant-coefficient difference or differential equations, partial fraction expansion, the evaluation of many types of integrals and sums, and complex numbers and functions. These are precisely the type of computations that can be accomplished quite easily with *Mathematica* using familiar mathematical notation. As an example consider the evaluation of a convolution integral, specifically, the response of a first-order RC circuit to a pulse input (Example 2.7 in [7]). This defines the input and impulse response signals.

$h[\text{t\_}]\text{:=Piecewise}\left[\{\{e^{-t}, t \geq 0\}\}\right];$
$x[\text{t\_}]\text{:=Piecewise}[\{\{1, 0 \leq t \leq 2\}\}];$

This defines the calculation and returns the result. Significantly, note that integration over piecewise functions is now fully supported.

$y[\text{t\_}]\text{:=} \int_{-\infty}^{\infty} h[\tau] x[t-\tau] d\tau; y[t]$

$$\begin{cases} e^{-t}\left(-1+e^{2}\right) & t>2 \\ e^{-t}\left(-1+e^{t}\right) & 0<t\leq 2 \end{cases}$$

## 2. COURSE OUTLINE

The Department of Engineering (Electrical) at the University of Southern Maine has a fairly traditional curriculum in which a circuits course is followed by linear signals and systems and courses on electronics, electromagnetism, materials, and more. The linear signals and systems class therefore covers both discrete-time and continuous-time topics in roughly equal proportions. The current textbook for the course is "Signals and Systems" by Haykin and Van Veen [7]. Here is a list of the main topics of the course (the numbers in brackets indicate number of contact hours in lectures and laboratories combined):

1. Basic signals and introduction to LTI system properties (7.5)
2. Time-domain representations of LTI systems (15)
3. Fourier analysis of LTI systems (20)
4. Applications: sampling (5), filters and filter design (10), and communications (5)
5. Laplace and z-transforms (7.5)

The course departs from the norm in its implementation. Until fairly recently, the problem of translating standard mathematical notation into traditional high-level programming languages made it difficult to use programming, visualization and computers in the undergraduate classroom. This however is no longer the case, on the contrary, the rapid advances in the power of computer software and hardware have made it possible to use computation and scientific visualization as a learning tool without many of the difficulties associated with earlier programming environments. To benefit from these advances *Mathematica* was chosen as the department's primary software resource and is currently used in the signals and systems course and a few others. It was selected for its unique mix of excellent algebraic and numeric functionality combined with a highly interactive environment for computation and presentation. It is the latter feature, the notebook interface, that enables successful classroom use of the software. It allows the instructor to seamlessly mix presentation with computation in a computer-equipped classroom. This has resulted in a more participative and interesting learning environment that naturally reduces listener passivity so typical in a traditional lecture. *Mathematica* is ideally suited for interactive classroom use since its powerful typesetting features break the syntax barrier present in most other computational environments. Any mathematical formula may be written in its native, familiar form. Additionally, if desired mathematical operators commonly encountered in a given discipline may be associated with their standard operator symbols. The development of course materials that make use of the powerful computation and visualization features of *Mathematica* has further enriched the course. The *Mathematica* notebooks used to support the course are available online at http://www.usm.maine.edu/∼mjankowski/docs/ele314/labs/.

## 3. ELECTRONIC COURSEWARE

A *Mathematica* notebook is simultaneously a platform for computation and a technical document. These two forms complement themselves in a classroom setting allowing the instructor to freely mix two types of classroom activities. Notebooks, like standard textbooks, may include explanatory text, typeset mathematical equations and graphics, but unlike textbooks may also include evaluatable mathematical expressions and programs. In the signals and systems course, each notebook is written so as to maximize the opportunities for live computation during a class session. The computational capabilities in *Mathematica* are used in a variety of ways. At its most basic level *Mathematica* may be used as a simple, but powerful calculator which students can use to solve many of the problems found in standard undergraduate textbooks. The student benefits by spending more time formulating and understanding the problem than on algebraic manipulation. Examples of algebraic evaluation common to many introductory undergraduate courses include:

- solutions to constant coefficient difference and differential equations,
- solutions of systems of (non)linear equations,
- evaluation of convolution sums and integrals,
- evaluation of Fourier, Laplace and z-transform integrals and sums,
- partial fraction expansion of rational polynomials.

Here are a few examples of typical calculations. First, the discrete-time Fourier transform (DTFT) of a causal, real exponential sequence.

$$\sum_{n=0}^{\infty} r^n e^{-i\omega n}$$

$$\frac{e^{i\omega}}{e^{i\omega}-r}$$

Here is the inverse DTFT of an ideal, full-band differentiator.

$$\int_{-\pi}^{\pi} i\omega e^{i\omega n} d\omega$$

$$\frac{2n\pi\text{Cos}[n\pi]-2\text{Sin}[n\pi]}{n^2}$$

This simplifies the expression using a symbolic rule and a replace operation. (Symbol % indicates the most recent output.)

$$\%/.\text{Sin}[n\pi] \to 0$$

$$\frac{2\pi\text{Cos}[n\pi]}{n}$$

This returns the unit impulse response of a second-order discrete-time system (Example 8.15 in [8]).

**RSolve**$[\{y[n]-\frac{5}{6}y[n-1]+\frac{1}{6}y[n-2]==$
**KroneckerDelta**$[n]$,$y[-1]==0,y[-2]==0\},y[n],n]$**//First**

$$\{y[n]\to -6^{-n}\left(2^{1+n}-3^{1+n}\right)\text{UnitStep}[1+n]\}$$

This evaluates the response for a range of values of n=-2,-1,...,10.

**Table[Evaluate**$[\{n,y[n]/.\%\}],\{n,-2,10\}]$

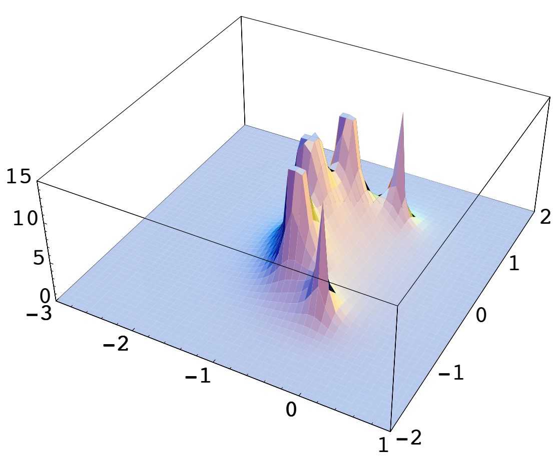
$$\{\{-2,0\},\{-1,0\},\{0,1\},\{1,\tfrac{5}{6}\},\{2,\tfrac{19}{36}\},\{3,\tfrac{65}{216}\},\{4,\tfrac{211}{1296}\},\{5,\tfrac{665}{7776}\},\{6,\tfrac{2059}{46656}\},\{7,\tfrac{6305}{279936}\},\{8,\tfrac{19171}{1679616}\},\{9,\tfrac{58025}{10077696}\},\{10,\tfrac{175099}{60466176}\}\}$$

*Mathematica* has excellent graphical capabilities which help the student visualize and the instructor demonstrate many important phenomena, thus assisting in the learning process. As an example, consider the ease with which we can obtain the plot of the transfer function and frequency response of a 5th-order Butterworth filter system. This defines the transfer function.

$H$**[s_]:=**
$1/(1.0+3.23607s+5.23607s^2+5.23607s^3+3.23607s^4$
$+s^5)$;

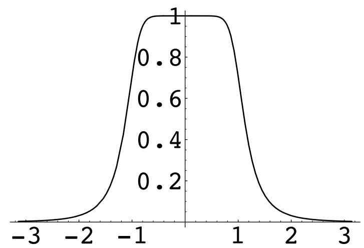
Here is a surface plot of |H(s)|, the magnitude of the system transfer function.

**Plot3D[Abs**$[H[\sigma+i\omega]],\{\sigma,-3,1\},\{\omega,-2,2\}$,
**PlotRange** $\to\{0,15\}$, **PlotPoints** $\to 50$, **Mesh** $\to$ **False**];



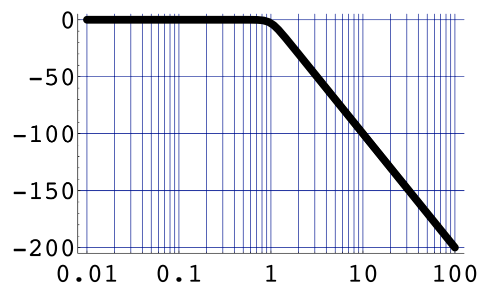

Here is the magnitude of the frequency response |H(j $\omega$)|. Note that the frequency response is obtained by evaluating the system transfer function H(s) on the imaginary axis.

**Plot[Abs**$[H[i\omega]],\{\omega,-\pi,\pi\}]$;




Finally, here is the so-called Bode plot of the magnitude spectrum.

**LogLinearPlot[20Log[10, Abs**$[H[i\omega]]],\{\omega,0.01,100\}$,
**PlotStyle** $\to$ **Thickness[0.02], GridLines** $\to$ **Automatic**];




## 4. ASSIGNMENTS AND PROGRAMMING PROJECTS

To deepen understanding and further promote active learning, "take-home" assignments and computer projects complement such simple exercises. It has been our experience that projects are an important learning instrument. Given a capable computational tool that makes mathematics more accessible and programming easier, projects help to motivate the student, they reward initiative and creativity, and prepare for independent learning. In addition to all the features described thus far, *Mathematica* has a very versatile and powerful programming language and code development environment. It provides many different programming paradigms from which to choose. Below are four code segments illustrating alternative implementations of a convolution sum.

1. For-loop method - typical procedural code for a *Mathematica* novice.

**nx = Length[sig]; nh = Length[ker];**
$y$ **= Table[0,** $\{n$, **nx − nh + 1}];**
**For[**$n=1,n\leq$ **nx − nh + 1,** $n$**++,**
**For[**$i=1,i\leq$ **nh,** $i$**++,**
$y[[n]]$**+=sig**$[[n+i-1]]$ **ker[[nh** $-i+1$**]]]]**

2. Sum method.

**nx = Length[sig]; nh = Length[ker];**
**Table[**$\sum_{i=1}^{\mathrm{nh}}$ **sig**$[[n+i-1]]$ **ker[[nh** $-i+1$**]],**

$\{n, \mathrm{nx} - \mathrm{nh} + 1\}]$

3. Dot product method.

```
nx = Length[sig]; nh = Length[ker];
Table[Take[sig, {n, n + nh − 1}].Reverse[ker],
{n, nx − nh + 1}]
```

4. Map method - typical functional coding style.

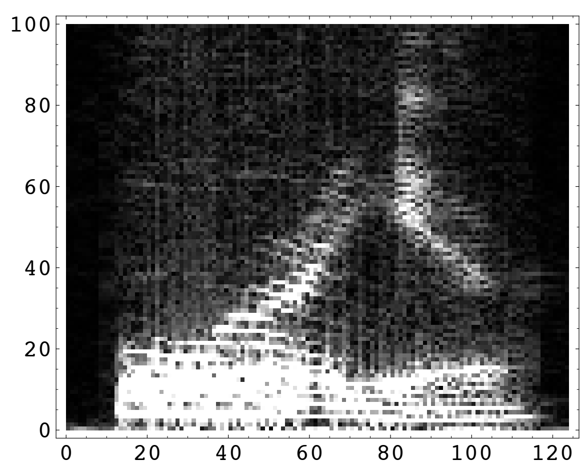
```
nh = Length[ker];
Reverse[ker].#&/@Partition[sig, nh, 1]
```

The convolution project is typically the first programming assignment during the course. In recent semesters other programming projects included:

- blurring and edge detection of grayscale images,
- lowpass FIR filter design and testing, filtering of audio data,
- FIR differentiator design and filtering of noise-free and noisy data,
- implementation of a spectrogram plot.

The spectrogram is simply a plot of the magnitude of the DFT of fixed-length segments (possibly overlapping) of some signal. The implementation of this calculation is particularly simple: function Partition breaks a list (i.e., vector) into segments of length given by the second argument with an offset specified by the third argument. Function Fourier is then applied to each segment resulting in a matrix of coefficients. The spectrogram is a plot of the magnitude of this matrix. The following code shows a somewhat simplified version of the spectrogram algorithm. This highlights the remarkable fact that the whole program can be written with a short, single line of nested function calls. The output shows the spectrogram of the word “Fourier”.

```
ListDensityPlot[
Transpose[Abs[Drop[Fourier[#], −100]]&/@
Partition[sig, 200, 50]], Mesh → False,
AspectRatio → Automatic];
```




## 5. CONCLUSION

The use of *Mathematica*, a leading CA system, in an undergraduate signals and systems course was presented. The article highlights the potential benefits of using *Mathematica* in an engineering classroom. The *Mathematica* notebook permits live use in the classroom. Excellent mathematical typesetting features eliminate the typical syntax barrier encountered when using a new system. The programming language allows for rapid prototyping of algorithms. Finally, most, if not all, algebraic or numeric calculations needed in an undergraduate engineering program can be obtained with usually no more than a single line of *Mathematica* code.

## 6. REFERENCES


[1] K.E. Schwingendorf and E. Dubinsky, “Purdue University: Calculus, concepts, and computers: Innovations in learning,” in T. W. Tucker (Ed.), Priming the calculus pump: Innovations and resources, Washington, DC: The Mathematical Association of America, pp. 175-198, 1990.

[2] D. Brown, H.A. Porta, and J.J. Uhl, “Calculus and *Mathematica*: A laboratory course for learning by doing,” in L. C. Leinbach (Ed.), The laboratory approach to teaching calculus, Washington, DC: The Mathematical Association of America, pp. 99-110, 1991.

[3] M. Yoder, “The use of symbolic algebra in electrical engineering,” Comput. Educ. Division ASEE, vol. 1, pp. 56-60, 1991.

[4] B.L. Evans, L.J. Karam, K.A West, and J.H. McClellan, “Learning Signals and Systems with *Mathematica,*” IEEE Trans. Education, Vol. 36., No. 1, pp 72-78, 1993.

[5] M. Jankowski, “A Computer Classroom for Electrical Engineering: enhancing undergraduate teaching and learning,” NSF DUE-9650253, 1996.

[6] K.E. Wage, J.R. Buck, C.H. Wright, and T.B. Welch, “The Signals and Systems Concepts Inventory ,” IEEE Transactions on Education, Vol. 48, No. 3, pp. 448 - 460, 2005.

[7] S. Haykin and B. Van Veen, Signals and Systems, John Wiley & Sons, Inc., 2003.

[8] J. H. McClellan, R. W. Schafer, and M. A. Yoder, “Signal Processing First,” Pearson/Prentice Hall, 2003.