

The "Gray Box" approach

The potential of symbolic mathematical software in
engineering analysis and education

Tom Lee Ph.D.
Maplesoft, Waterloo ON, Canada
tlee@maplesoft.com

▼ Abstract

The proverbial "black box" refers to computing tools where the user is shielded from the inner algorithms and interacts with the system from a purely and input and output point of view. Most modern engineering modeling and simulation software are based on this notion. The "gray box" is a term that describes the symbolic approach to engineering computation that offers distinct benefits in education and research. This paper offers an introduction to the tools and techniques of successfully deploying symbolic tools. The Maple system will be used as the software context.

▼ The challenge of modern engineering design education

Engineering, by definition, is the integration of the theoretical sciences and arts, and the practical techniques of application. This, indeed, is the source of many key challenges that face educational institutions today. The theoretical elements of engineering has historically been the emphasis in most undergraduate curricula with the "softer" or human side of engineering often addressed through internships or in many cases relegated to the post-academic phase in one's career. Over the recent decades, of course, the academic community has actively developed various techniques to complement the traditional rigor with context. Laboratories, capstone design projects, engineering competitions, and many innovative uses of technology are some examples.

▼ Impact of technology in education

Today, it would be challenging to find an engineering program that does not deploy technology to assist in education. From virtual laboratories, to modeling and simulation, to basic research and knowledge exploration, technology and engineering education are, forever, intimately bonded. In most cases, it appears that the role that any technology plays in the pedagogy arguably coalesces into one of two perspectives:

- Increase the efficiency of information flow – i.e. access more information, access different types of information, finish a problem faster, iterate the problem, etc.
- Introduce a new dimension in the educational mix – i.e. view the problem from a different angle, enrich the learning experience, etc.

In addition, a third strong trend has also emerged

- Use the technology as an end in itself. For example, the use of industry standard engineering software familiarizes the student with a tool of the profession.

The author submits that no one element of the above is clearly more important than another. Indeed, the most successful educational technology (i.e. as indicated by adoption trends, and longevity) generally shows the first or third point as being the most influential. For example general professional computation aids, solid modeling packages, commercial simulation tools, etc. are all highly successful in education though none were explicitly designed for teaching or learning. This might seem counterintuitive to educators who perceive the enrichment goals of the second point to have the more sound pedagogical framework.

▼ The Maple worldview

One class of software technology that has managed a unique balance of the three main perspectives, and reconciled the practical with the pedagogically enriched, are the general-purpose math systems with strong symbolic origins. Maple is a leading example of such systems. Today, these systems have integrated computations that make them substantially more than pure symbolic tools, but it is the core symbolic element that offers some of the most significant advantages for engineering education.

Most users of symbolic software are initially attracted to the technology that can compute closed-form or analytical solutions to mathematical problems. At the simple level, day-to-day tasks such as polynomial manipulation, basic differential and integral calculus, etc. can be virtually instantaneous and error free. For example, given two rational expressions of polynomials,

$$G(s) = \frac{G_N(s)}{G_D(s)} \text{ and } H(s) = \frac{H_N(s)}{H_D(s)},$$

derive the expression,

$$\frac{G(s)}{1 + G(s)H(s)}.$$

This is a common symbolic derivation encountered in a typical introductory linear control systems course, and is used to derive the closed-loop transfer function for a particular canonical form. A typical rational expression would be a constant or first order numerator and perhaps a second order denominator for $G(s)$, as an example.

The algebraic manipulation would not be beyond the limits of a typical undergraduate but it would be tedious and error-prone in the least. Many would also question what the educational value would be following some initial manual attempts during which a student would experience the underlying mechanics. Following the introductory phase, (i.e. during the application phase), it is a topic of debate whether such steps have any continuing contribution to learning.

Symbolic algorithms essentially automate that manual process. For example, for the following Maple sequence performs this particular type of computation (Note: in Maple computation examples, red expressions denote a user's input and blue expressions denote the results of the computation).

$$G := \frac{s + 1}{s^2 - 2s + 3} = \frac{s + 1}{s^2 - 2s + 3}$$

$$H := \frac{1}{s + 2} = \frac{1}{s + 2}$$

$$Y := \frac{G}{1 + G \cdot H} = \frac{s + 1}{(s^2 - 2s + 3) \left(1 + \frac{s + 1}{(s^2 - 2s + 3)(s + 2)} \right)}$$

$$\text{simplify}(Y) = \frac{(s + 2)(s + 1)}{s^3 + 7}$$

The time savings for such a calculation is likely on the order of 10 to 15 minutes for the most algebraically dexterous student and would likely be more for the typical student.

Time savings notwithstanding, the key observation here is that the theoretical steps were, in fact, maintained – i.e. the steps of

- define the functions
- apply the formula
- algebraically consolidate and simplify the result

are common whether done manually or through technology. This is the essence of the “Gray Box” perspective and indicates the primary advantage that commercial symbolic systems have over virtually all other forms of technology deployed in engineering education.

▼ The Gray Box concept

The gray box label is derived from the proverbial “black box” approaches to computation. The black box, of course, implies a hidden machinery that will take input information and produce output information with where the intermediate steps bear little or no resemblance to the problem domain. Numerical solution algorithms are examples of black box approaches. For example, the Runge-Kutta algorithms are famous and relatively easy to derive and present but they have no connection to the meaning of the differential equation that they try to solve. In engineering, differential equations carry physical or real systems meaning during derivation. Solving them numerically, removes that meaning.

The principal benefits of gray box technology are:

- The time savings efficiency are comparable to the black box. Steps are automated and error is reduced or eliminated. Solutions can be quickly computed for a wide range of problems. In fact, systems like Maple can easily be configured as black box systems and compute with the same
- The structure of the mathematics is maintained. Whenever possible, symbolic technology maintains the underlying algebraic structure. The immediate implication is flexibility. Parameters can remain undefined and mathematical information can be obtained in more natural form. For example, in the case of differential equation, a traditional numerical solver would force all equations to be expressed as a system of first order DEs isolating the derivative terms onto the left hand side (i.e. define in explicit form). Symbolic systems are more forgiving and allows the representation in implicit form – which is often the form encountered in reference material and map better to the equilibrium or balance concepts that often drive the formulation of the DE models. In addition, symbolic systems naturally support more flexible exploration of the parameter space.

For example, the following brief Maple session,

- Expresses a first order DE in natural, implicit form. The parameter a is left in symbolic form and does not need a numerical value for solution.
- Solves the DE analytically to generate a closed form solution (initial condition was supplied). The solution is $y(t, a)$.
- Studies the solution $y(t, a)$ graphically to determine the influence of a on the solution
- Evaluates the solution for different numerical values of a .
- Revert a to pure undefined (symbolic) form and the original solution structure is immediately recovered.

Define implicit DE

$$de := \dot{y} + a \cdot y + \frac{1}{2} = \sin(t) = \frac{d}{dt} y(t) + a y(t) + \frac{1}{2} = \sin(t).$$

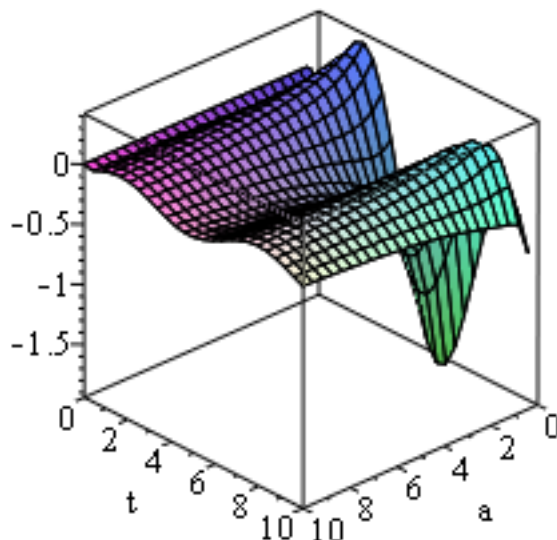
Using the right-click menu item *Solve ODE interactively* (launches Maple's interactive ODE solution assistant) with the initial condition $y(0) = 0$, the following solution can be computed (note that the traditional *dsolve* command could have also been used).

$$y(t) = -\frac{1}{2} \frac{e^{-at} (-a^2 - 1 - 2a)}{a(a^2 + 1)} + \frac{1}{2} \frac{-a^2 - 1 - 2 \cos(t) a + 2 a^2 \sin(t)}{a(a^2 + 1)}$$

Using the right-click menu item *Right-hand side*, extract the right hand side of the expression. Then using the menu item *Assign to a name*, give it the label, *sol*. Then use the menu item *Plot builder* (launches Maple's interactive assistant for composing complex plots) and choose *3D plot*

$$-\frac{1}{2} \frac{e^{-at} (-a^2 - 1 - 2a)}{a(a^2 + 1)} + \frac{1}{2} \frac{-a^2 - 1 - 2 \cos(t) a + 2 a^2 \sin(t)}{a(a^2 + 1)}$$

$$sol := -\frac{1}{2} \frac{e^{-at} (-a^2 - 1 - 2a)}{a(a^2 + 1)} + \frac{1}{2} \frac{-a^2 - 1 - 2 \cos(t) a + 2 a^2 \sin(t)}{a(a^2 + 1)} \rightarrow$$



Referring back to the result contained in *sol*, using the right-click menu item *Evaluation at a point*, specify $a = \frac{1}{3}$

$$\frac{12}{5} e^{-\frac{1}{3}t} - \frac{3}{2} - \frac{9}{10} \cos(t) + \frac{3}{10} \sin(t)$$

Try *Evaluation at a point* with $a = 0.3$

$$2.584097859 e^{-0.3t} - 1.666666666 - 0.9174311925 \cos(t) + 0.2752293578 \sin(t)$$

Some observations or benefits should be noted.

The mathematical structure assists in interpretation of the results. For example the expression,

$$2.584097859 e^{-0.3t} - 1.666666666 - 0.9174311925 \cos(t) + 0.2752293578 \sin(t)$$

immediately conveys some principal characteristics:

- positive exponential term \rightarrow instability or exponential growth
- constant term \rightarrow offset
- cosine and sine terms \rightarrow oscillatory behavior

Of course with further exploration, the overall magnitude and severity of these elements would be determined. Having a surface representation of the solution space (as in the above example), is a quick way to summarize the total impact of the parameter a in a concise form.

Another observation is that if a is defined exactly (i.e. integer or fraction), the solution remains exact. If a is a floating point value, then the expression adjusts accordingly. Floating point number by definition are approximations and floating point computations, by definition eliminate meaning. For example the value 2.2214 would generally not mean much to most observers and very few could guess the origin of the number. However, the number

$\frac{\pi}{\sqrt{2}}$ is exact and depending on the sequence of operations and context, one can often deduce its

origins. Systems such as Maple lets you easily manage either form. By default, they tend to prefer the exact form as in the case of $a = \frac{1}{3}$ in the above example. Maintaining such a form means that

theoretically your computations will never degrade due to round-off or propagation errors. From an education perspective, the greater significance will likely be the preservation of meaningful mathematical information, including numerical elements, in your work.

▼ Example – Control systems education

The purpose of the foregoing example was to illustrate that natural mappings between systems like Maple and the theoretical steps and frameworks within applied mathematics. The following is a more substantial example where these points are amplified greatly through a focused context.

Control theory is an ideal context for introducing symbolic computation techniques because it is an integrative course that ties in theoretical elements of calculus, algebra, differential equations, complex numbers, special functions and transforms, linear systems theory, etc. There is a theoretical richness and complexity that provides the greatest opportunities for education but also challenges commensurate with the opportunities.

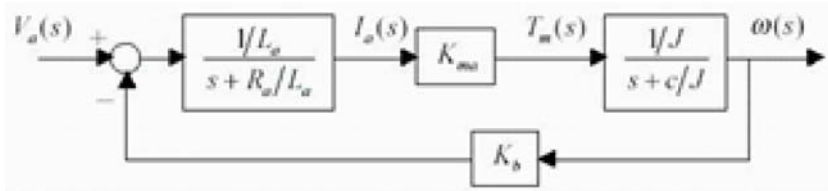
Secondly, it is a course that is inherently design-centric. More so than most mathematics-intensive courses, it is relatively easy to cast a problem in control as a design problem. “We wish to eliminate the level of oscillation and decrease reaction time for this system ... how do we do this?” is a design problem and one that naturally fits into even an introductory course. The challenge that many instructors would have, however, is efficiently reconciling the richness of the theory and the design elements within the constraints of a finite semester.

Gray box approaches are ideal for bringing the theory in focus in a design setting. By maintaining the mathematical structure, one can easily manipulate mathematical primitives of the theory: differential equations, complex arithmetic, linear systems theory, etc. More importantly, one can quickly convey first principles in strong design contexts.

An example is root locus analysis. In conventional control education, and indeed in most conventional professional tools, the root locus analysis technique is presented as the path that the poles take as the forward path gain is varied in a system in a particular canonical form. Within such a canonical form, one can then derive the classic “plotting” rules that often bewilder and confuse students. A more general definition of a root locus could be the path that the system poles (roots of the characteristic polynomial) carve as you vary a system parameter. Conceptually this could be any system parameter that is algebraically tied to the characteristic polynomial. From a design perspective, this broader interpretation could be much more important as it makes the connection between parameters, pole positions, and overall system performance.

For example, the following simple example places control mathematics in a design-oriented problem context that would generally not be attempted without appropriate technology support. Using Maple, a Laplace domain model of a control system (DC motor), is defined block by block, then combined and simplified. From this basic model, certain parameter studies are conducted including the root locus analysis for the parameter τ , the feedback loop parameter. Such a root locus can only be derived from the more general first principles interpretation as opposed to the drawing rules based analysis.

Initializations and block definitions



$$G1 := \frac{\frac{1}{L_a}}{s + \frac{R_a}{L_a}} = \frac{1}{L_a \left(s + \frac{R_a}{L_a} \right)}$$

$$G2 := K_{ma} = K_{ma}$$

$$G3 := \frac{\frac{1}{J}}{s + \frac{c}{J}} = \frac{1}{J \left(s + \frac{c}{J} \right)}$$

$$H1 := K_b = K_b$$

Compute closed loop transfer function and *Simplify* using the right click menu

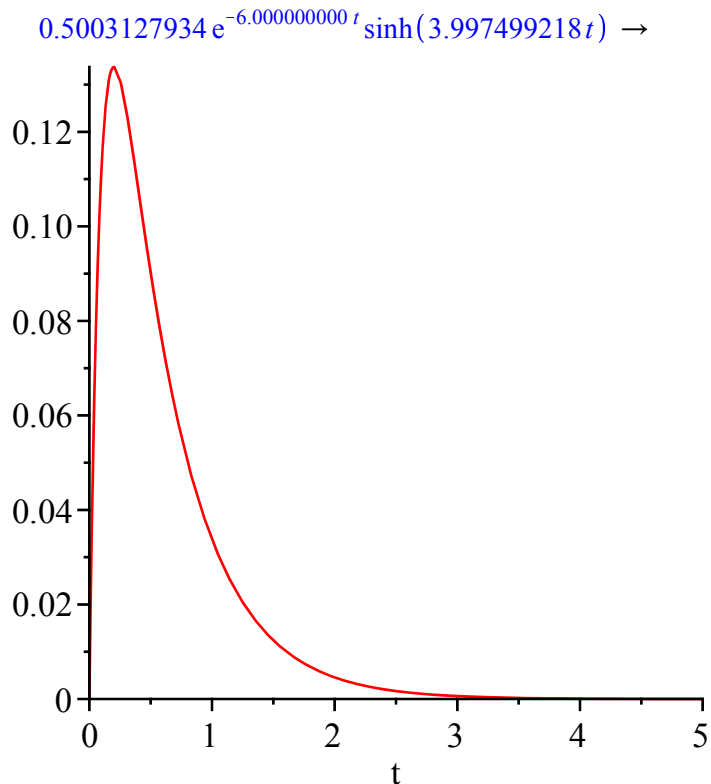
$$TF := \frac{G1 \cdot G2 \cdot G3}{1 + H1 \cdot G1 \cdot G2 \cdot G3} = \frac{K_{ma}}{L_a \left(s + \frac{R_a}{L_a} \right) J \left(s + \frac{c}{J} \right) \left(1 + \frac{K_b K_{ma}}{L_a \left(s + \frac{R_a}{L_a} \right) J \left(s + \frac{c}{J} \right)} \right)}$$

$$\frac{K_{ma}}{s^2 L_a J + s L_a c + R_a s J + R_a c + K_b K_{ma}} \rightarrow$$

Compute inverse Laplace transform to produced closed-form expression for the impulse response of the system. Use menu option *Integral transforms* → *Inverse Laplace transform*

$$\frac{2 K_{ma} e^{-\frac{1}{2} \frac{(L_a c + R_a J) t}{L_a J}} \sinh \left(\frac{1}{2} \frac{t \sqrt{L_a^2 c^2 - 2 J (R_a c + 2 K_b K_{ma}) L_a + R_a^2 J^2}}{L_a J} \right)}{\sqrt{L_a^2 c^2 - 2 J (R_a c + 2 K_b K_{ma}) L_a + R_a^2 J^2}} \rightarrow$$

Try some particular values and plot the response. Use *Evaluate at a point* and *Plot builder* → *2D Plot* respectively. Use the values: $R_a = 1.0$, $c = 0.1$, $J = 0.01$, $K_b = 0.01$, $K_{ma} = 0.1$, and $L_a = 0.5$.



Use *Evaluate at a point* on the transfer function definition. Enter same parameter values as before but omit the value for K_b . Give the result the name *TF2*.

$$\frac{K_{ma}}{s^2 L_a J + s L_a c + R_a s J + R_a c + K_b K_{ma}} \rightarrow \frac{0.01}{0.005 s^2 + 0.060 s + 0.10 + 0.01 K_b}$$

$$TF2 := \frac{0.01000}{0.00500 s^2 + 0.06000 s + 0.10000 + 0.01000 K_b}$$

Extract the denominator (i.e. obtain the characteristic equation).

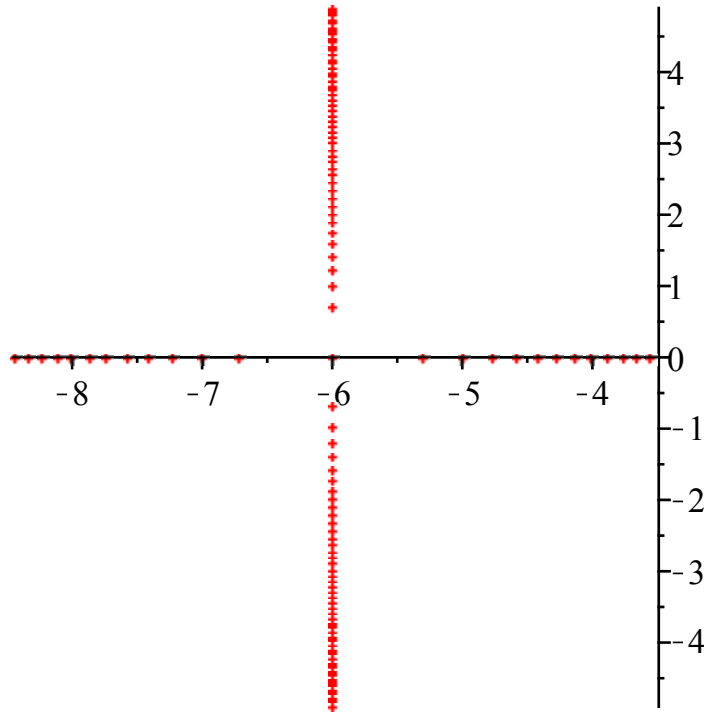
$$ceq := \text{denom}(TF2) = 0.005 s^2 + 0.060 s + 0.10 + 0.01 K_b$$

Vary K_b from $K_b = 5$ to $K_b = 20$ and solve the characteristic equations for its roots (including complex). Collect the real and imaginary parts for all of the roots and plot with real part on horizontal axis and complex parts on vertical axes. Use a Maple "for" loop to perform the iteration.

```

rldata := NULL :
for  $K_b$  from 5. by .25 to 20. do
  rldata := rldata, fsolve(ceq, s, complex)
od:
plots[complexplot]([rldata], style = point, symbol = cross)

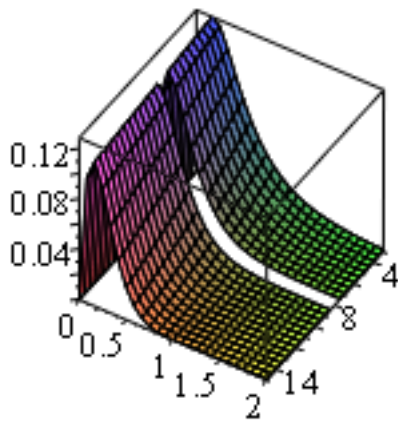
```



The above is essentially is a special form of a Root Locus plot where a non-standard parameter is varied and the resulting closed loop poles are plotted. In most applications of the root locus technique (educational or otherwise), the system is assumed to be in a unity feedback canonical form and the forward path gain is varied. This illustrates the extension of the root locus concept that was facilitated by Maple's flexible handling of the mathematical definitions.

Now plot the impulse response expression as a surface. To do this, simply reapply the inverse Laplace transform to the simplified transfer function and then use the Plot Builder. Note the gap in the surface. This is an opportunity to explore some critical values of interest for the design parameter.

$$\frac{0.01000}{0.00500s^2 + 0.06000s + 0.10000 + 0.01000K_b} \rightarrow \frac{2 \cdot e^{-6 \cdot t} \sinh\left(t \sqrt{16 - 2 \cdot K_b}\right)}{\sqrt{16 - 2 \cdot K_b}} \rightarrow$$



The most significant element of the foregoing example is the reconciliation of techniques and concepts from areas of control theory that often are not necessarily taught or learned in a cohesive context. The relationships among time response expressions, system transfer function, characteristic equations, pole values, parameter values are all interrelated and through the symbolic framework, can be explored interactively and iteratively.

In the final portion of the foregoing example, we note a discontinuity in the response surface, investigating the value of K_b corresponding to that gap will quickly illustrate that that gap corresponds with the value of K_b where poles transition from purely real to complex conjugates, or where the discriminants (the argument of the radicals) in the time response expressions transition from positive to negative, or where the system transitions from over to critically to under damped. Such concept connections are essential to the development of design intuition and for gaining maximum benefit from the rich traditions of engineering theory. Furthermore, these connections could not be as effectively or efficiently presented without computing tools that automate the techniques while maintaining the conceptual elements within the theory.

▼ Conclusions

This paper presented the educational benefits of symbolic computation technology in engineering education. The case was made that systems such as Maple were professional tools that had the capacity to save time and automate key techniques (as all strong engineering software packages do), but did it in a way that allowed for the theoretical elements were preserved. This was termed, the “gray box” – referring to the convenience of the “black box” without sacrificing control and insight into the domain. The benefits were presented in context of a control theory example where the technology facilitated the conceptual reconciliation of topics that are often not well integrated in the curriculum. This integrated view of the theory allow for richer exercises and establishing design orientation.

▼ References

- [1] Lee, Thomas 1990: Computer assisted methodologies for control systems education, University of Waterloo MSc Thesis, Faculty of Engineering.
- [2] Lee, Thomas; Heppler, G.R.; Wills, B.L. 1990: “Computer algebra systems for enhancing the learning environment for control systems education”, Proc. ASEE Annual Conference, Toronto, pp. 113-119
- [3] Porciello, G. Pino; Lee, T. 1990: “Issues in user interface design for control systems simulation software in engineering education”, Proc. 21st University of Pittsburgh Conference on Modeling and Simulation
- [4] Lee, Thomas; Heppler, G.R.; Wills, B.L.; Graham, E.D. 1989: “An interactive simulation environment for learning linear control systems”, Proc. Simulation in Engineering Education, SCS Multiconference, San Diego, pp. 110-114