

ALGEBRAIC SOLUTION OF MODELS WITH A QUICK THRUST BY MEANS OF A COMPUTER ALGEBRA SYSTEM

M. LEGUA¹

I. MORALES²

L.M. Sánchez RUIZ³

Abstract

Quick thrusts or pushes do appear in a number of engineering situations which are usually modeled by means of delta generalized functions. Some of these may be properly manipulated with computer algebraic systems such as Mathematica. The aim of this note is showing how this may be carried out in a number of specific examples corresponding to some dynamic systems.

Keywords: *Heaviside step function, Delta generalized function, Mathematica.*

1.INTRODUCTION

In many practical situations such when we face as a momentary sharp increase and fall in electric potential or when a quick thrust happens to appear we wish to make the mathematical modeling of the corresponding Engineering situation it is commonly done by considering limits in which a fixed amount of some physical fact is concentrated into an infinitesimal region. This situation leads to introduce *generalized functions* or *distributions* which can represent these limits in integrals and other types of calculations.

In this note we address this topic with Mathematica[®] ¹one of the most efficient computer algebra systems (CAS) in what concerns symbolic manipulation and many other features.

¹ CPS-Departamento de Matemática Aplicada, Universidad de Zaragoza

² IUMPA, Universidad Politécnica de Valencia, ETSIA-Matemática Aplicada

³ IUMPA, Universidad Politécnica de Valencia, ETSID-Matemática Aplicada

This mathematical software, whose first version came out in 1988, has indeed been commonly used in research and applications, cf. ² ⁵This CAS and another commonly used mathematical software, Matlab[®] ⁶, cf. ⁷ ¹⁰ make easy to solve initial value problems with a continuous or non-continuous input but both oblige to take into account some precautionary measures. In fact the authors studied the surprising and diverse results obtained by different Matlab versions when distributions happened to appear as inputs in dynamic systems ¹¹

2. QUICK THRUSTS MODELING

The unit step function $H(t)$, defined by

$$H(t) = 0 \text{ for } t < 0 \text{ and } H(t) = 1 \text{ for } t \geq 0, \quad (1)$$

is used in engineering problems to represent functions that appear for a limited time period. ⁴

¹ <http://reference.wolfram.com/mathematica/guide/Mathematica.html>

² Eppstein D., "Zonohedra and Zonotopes," *Mathematica in Education and Research* 5, 4 (1996), 15-21.

⁵ Weisstein E.W., "Making MathWorld." *The Mathematica Journal* 10 (2007), 474-488

⁶ <http://www.mathworks.com/products/matlab/whatsnew>

⁷ Bhardwaj R. and Kaur P. Ntaganda J.M., Mampassi B., Some B., Seck D and Samb A..

¹⁰ Legua M., Morales I. and Sánchez Ruiz L.M. "Initial value problems and casual functions with MATLAB"; *WSEAS Trans. Math.* 6, 2 (2007), 303-308.

¹¹ Ntaganda J.M., Mampassi B., Some B., Seck D and Samb A.. "Design of the human cardiovascular-respiratory control system using MATLAB"; *Far East J. Appl. Math.* 22 (2006), 137-153.

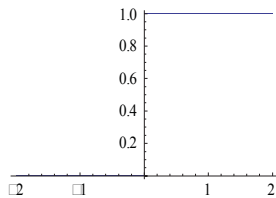
For instance if a function f just appears during an interval $[a,b]$ and vanishes outside we may formalize it by just considering the function

$$g(t) = f(t) (H(t - a) - H(t - b)). \quad (2)$$

The step function has no derivative at 0, in fact it is not even continuous at 0. However by using distributions, or generalized functions, the Dirac delta δ , can be viewed as its derivative.

These facts are handled by Mathematica with some caution. We recall that the unit step function and Dirac delta are represented in Mathematica by UnitStep and DiracDelta, respectively.

In: Plot[UnitStep[t], {t, -2, 2}]



In: D[UnitStep[t], t]

Out: (-1-DiscreteDelta[t]+UnitStep[-t]+UnitStep[t])/(1-DiscreteDelta[t

In: D[UnitStep[t], t]//Simplify

Out: Indeterminate $t=0$ This result is an unexpected. In fact Mathematica has also got another function, HeavisideTheta, which coincides with UnitStep for all numeric value different from 0 but remains undefined at 0. The derivative of this one provides the expected DiracDelta.

In: D[HeavisideTheta[t], t]

Out: DiracDelta[t]

In: {UnitStep[0],HeavisideTheta[0]}

Out: {1,HeavisideTheta[0]}

Let us note that simplifications are straightforwardly obtained,

In: Simplify[{t^2 HeavisideTheta[t], HeavisideTheta [3 t-1]},t>1]

Out: {t^2,1}

Example 1. (*Step functions and Laplace transforms with Mathematica*)

Find the Laplace transform of the casual function f defined by

$$f(t)=1/2 \text{ for } 3/4 \leq t < 1, f(t)=1 \text{ for } t \geq 1. \quad (3)$$

Sol.: The step function H enables to write

$$f(t) = 1/4(H(t)-H(t-3/4))+1/2(H(t-3/4)-H(t-1))+H(t-1).$$

Thus for our purpose we may proceed as follows:

In: LaplaceTransform[1/4 (UnitStep[t]-UnitStep[t-3/4]) +1/2 (UnitStep[t-3/4] - UnitStep[t-1])+UnitStep[t-1],t,s]

Out: $1/(4 s)+^{-s}/(2 s)+^{-3 s/4}/(4 s).$

Finding out the Inverse of the output of Example 1:

In: InverseLaplaceTransform[s,t]

Out: $1/4 (1+2 \text{ HeavisideTheta}[-1+t] + \text{ HeavisideTheta}[-3/4+t])$

We note that the UnitStep function has been replaced by the HeavisideTheta function, both of which indeed enjoy the same Laplace Transform:

In: LaplaceTransform[1/4(HeavisideTheta[t]HeavisideTheta[t3/4])+1/2(HeavisideTheta[t-3/4]-HeavisideTheta[t-1])+ HeavisideTheta[t-1],t,s]
Out: 1/(4s) + (2s) + (4s)

Searching for the Inverse Laplace Transform,

In: InverseLaplaceTransform[s,t]
Out: 1/4 (1+2 HeavisideTheta[-1+t] + HeavisideTheta[-3/4+t])

Thus even in this case this inverse Laplace transform does not provide the original function since the program has replaced a HeavisideTheta function in the origin function by the unit.

By means of the Simplify command we may easily check out this issue:

In: 1/4 (HeavisideTheta[t]-HeavisideTheta[t-3/4])+1/2 (HeavisideTheta[t-3/4]-HeavisideTheta[t-1])+ HeavisideTheta[t-1]//Simplify
Out: 1/4 (2 HeavisideTheta[-1+t]+HeavisideTheta[-3/4+t] +HeavisideTheta[t]).

Hence the program systematically replies the HeavisideTheta function when finding Inverse Laplace Transform but does not provide this one when there is some delay:

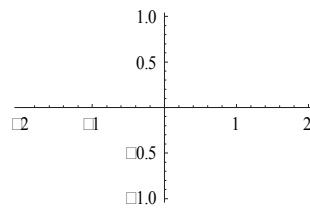
In: {LaplaceTransform[1,t,s],LaplaceTransform[UnitStep[t],t,s],LaplaceTransform[HeavisideTheta[t],x,s]}
Out: {1/s,1/s,1/s}

In: InverseLaplaceTransform[s,t]
Out: {1,1,1}

In what concerns DiracDelta[t] it returns 0 for all value t other than 0. It does not reply Infinity as its value at 0 but handles the DiracDelta properly in integrals.

In fact this is what has happened with UnitStep and HeavisideTheta as shown by finding the above Laplace Transforms.

In: Plot[DiracDelta[x], {x, -2, 2}]



Out:

In: DiracDelta[0]

Out: DiracDelta[0]

In: Integrate[DiracDelta[x], {x, -Infinity, Infinity}]

Out: 1

If we are to find the Laplace transform of a generalized function such as $e^{2t} \delta(t-\pi)$ we may write

In: LaplaceTransform[Exp[2t]*DiracDelta[t-Pi],t,s]

Reversely, the program should provide the corresponding Inverse Laplace Transform. However we get the unpleasant surprise that the program can only detect the DiracDelta input when there is no delay

In: InverseLaplaceTransform[Exp[-Pi (-2+s)],s,t]

Out: InverseLaplaceTransform[Exp[-Pi (-2+s)],s,t]

Note 1. Let us point out the above by finding the Laplace Transform of the standard DiracDelta and the DiracDelta with one unit delay, and subsequently asking Mathematica to obtain the corresponding Inverse Laplace Transforms,

In: LaplaceTransform[{DiracDelta[t],DiracDelta[t-1]}, t,s]

Out: {1^{-s}}

In: InverseLaplaceTransform[s,t]
Out: {DiracDelta[t], InverseLaplaceTransform[s,t]}.

Note 2. Mathematica can manipulate the derivative of the Dirac delta Dirac(1,t) which is a distribution δ' that satisfies

$$t\delta'(t) = -\delta(t). \quad (4)$$

In: D[DiracDelta[t],t]
Out: DiracDelta[t]

In: t*D[DiracDelta[t],t]//Simplify
Out: - DiracDelta[t].

3.INITIAL VALUE PROBLEMS

Laplace transforms are useful to solve initial value problems modelled by (a system of) linear differential equations with constant coefficients.

Next we provide some examples which show that the solution provided by Mathematica has to be adequately manipulated in order to check that it satisfies the dynamic systems when they correspond to the modeling of spike inputs.

Example 2. (*A Dirac delta input*) Solve

$$y'' + 5y' + 6y = \delta(t), y(0) = 0, y'(0) = 0. \quad (5)$$

Sol.: We find the solution by means of the DSolve Mathematica command

In: DSolve[{y''[t]+5*y'[t]+6*y[t]==DiracDelta[t], y[0]==0,y'[0]==0},y[t],t]
Out: {{y[t]- (-3t(-1+t) (1-HeavisideTheta[t]))}.

If we wish to check the answer,

In: z[t]=(-Exp[-3*t] (-1+ Exp[t]) (1-HeavisideTheta[t])
Out: $^{-2t}(-1+t)(1-HeavisideTheta[t])$

In: FullSimplify[D[z[t],{t,2}]+5*D[z[t],t]+6*z[t]]
Out: $-2^t(-1+\text{HeavisideTheta}[t]) - 3^t (-1-\text{DiracDelta}[t]-1\text{HeavisideTheta}[t])$

We have not obtained the expected DiracDelta. However, proceeding step by step.

In: D[z[t],t]//Simplify
Out: $-2^t(-3+2^t)(-1+\text{HeavisideTheta}[t])$

In: D[t]+5+6*z[t]//Simplify
Out: DiracDelta[t]

Finally the initial conditions may just partially be checked because of the no determination of the HeavisideTheta function at 0,

In: {z[0],D[z[t],t]/.t0}
Out: {0,-1+HeavisideTheta[0]}.

Despite the remark made at Note 1, the solution is much easily checked when there is some delay in the spike of the dynamic system. For instance in front of a delay of two units in the thrust of Example 2 the output and its checking are more easily handled.

Example 3. (*A delayed Dirac delta input*) Solve

$$y'' + 5y' + 6y = \delta(t-5), y(0) = 0, y'(0) = 0. \quad (6)$$

Sol.: We find the solution by means of the DSolve Mathematica command

In: DSolve[{y''[t]+5*y'[t]+6*y[t]==DiracDelta[t-5],y[0]==0,y'[0]==0}, y[t],t]
Out: {{y[t]^{10-3t} (-5+^t) HeavisideTheta[-5+t]}}

If we wish to check the answer,

In: z[t_]=Exp[10-3t](-Exp[5]+Exp[t])

Out: $10^{-3t}(-5+t)\text{HeavisideTheta}[-5+t]$

In: FullSimplify[D[z[t],{t,2}]+5*D[z[t],t]+6*z[t]]
Out: DiracDelta[-5+t]

In this case the initial conditions are easily checked,

In: {z[0],D[z[t],t]/.t0}
Out: {0,0}.

Example 4. (*A Step function input*) Solve

$$y'' + 5y' + 6y = H(t), y(0) = 0, y'(0) = 0. \quad (7)$$

Sol.: The solution is easily obtained with Mathematica,

In: DSolve[{y''[t] + 5 y'[t] + 6 y[t] == UnitStep[t], y[0]==0, y'[0]==0},y[t],t]
Out: {{y[t]1/6(-1)^2 (2) UnitStep[t]}}

If we try to check this is the right answer, the solution is not fully satisfactory but may be considered helpful

In: z[t_]= 1/6 Exp[-3t]-1+Exp[t]^2 (2+Exp[t]) UnitStep[t]
Out: 1/6 $^{-3t}(-1+t)^2 (2+t)$

In: FullSimplify[D[z[t],{t,2}]+5*D[z[t],t]+6*z[t]]
Out: $\begin{cases} 1 & t > 0 \\ \text{ComplexInfinity} & t == 0 \end{cases}$

In this case the initial conditions cannot be fully verified,

In: {z[t],D[z[t],t]/.t0}
Out: {0,Indeterminate}.

4.CONCLUSIONS

Mathematica is a computational software program commonly used in scientific, engineering, and mathematical fields. As all computer algebra systems Mathematica saves time in routine calculations and application of standard techniques. In fact their capabilities to deal with symbolic calculations have been continuously enhanced and simplified at the same time.

From the very first release Mathematica was split into two parts, the *kernel* and the *front end*. The former interprets the input expressions and returns the output expressions. The latter allows the creation and handling of the *Notebook* document which contains the program code and formatted text as well as the results. Documents can be structured using a hierarchy of cells, which allow outlining and sectioning of the file.

The continuous enhance of its capabilities for high-performance computing include the introduction of packed arrays, sparse matrices, and libraries designed to evaluate with high-precision arithmetic.

All the above has made Mathematica one of the strongest computer algebra systems in what refers to algebraic manipulation and by the number of examples shown in this note it can properly handle some distributions such as Dirac delta though we have had the chance to check that sometimes the answers are not fully satisfactory.

As a general rule, checking the answer provided by Mathematica may prove to be more than just an exercise of dealing with the program, it may show that something is wrong or confirm the answer is right.

Then knowledge and experience from the user will be required in order to avoid taking for good the program answer and perhaps decide whether using the `UnitStep` or `HeavisideTheta` function if needed, both equal to each other but at the origin and with a different behavior in the derivatives.

5. REFERENCES

<http://reference.wolfram.com/mathematica/guide/Mathematica.html>

Eppstein D., "Zonohedra and Zonotopes," *Mathematica in Education and Research* 5, 4 (1996), 15-21.

Moraño J.A. and Sánchez Ruiz L.M. *Cálculo y Álgebra con Mathematica 7*. Editorial de la UPV 2010.

<http://www.wolfram.com/>

Weisstein E.W., "Making MathWorld." *The Mathematica Journal* 10 (2007), 474-488.

<http://www.mathworks.com/products/matlab/whatsnew>

Bahriawati C. and Carstensen C., "Three MATLAB implementations of the lowest-order Raviart-Thomas MFEM with a posteriori error control"; *Comput. Methods Appl. Math.* 5 (2005), 333-361.

Bhardwaj R. and Kaur P. "Chaos using MATLAB in the motion of a satellite under the influence of magnetic torque"; pp. 337-372 in *Mathematical models and methods for real world systems* (Ed.: K. M. Furati, Z. Nashed and A.H. Siddiqi). LNPAM 247. Chapman & Hall, 2006.

Lin X., Liu A., Li Y and Wu P. "A MATLAB programming for simulation of X-ray capillaries"; *Appl. Math. Comput.* 172 (2006), 188-197.

Ntaganda J.M., Mampassi B., Some B., Seck D and Samb A.. "Design of the human cardiovascular-respiratory control system using MATLAB"; *Far East J. Appl. Math.* 22 (2006), 137-153.

M. LEGUA, I. MORALES
L.M. Sánchez RUIZ

Legua M., Morales I. and Sánchez Ruiz L.M. “Initial value problems and casual functions with MATLAB”; WSEAS Trans. Math. 6, 2 (2007), 303-308.

Bracewell R. “The Impulse Symbol”; pp. 69-97 in The Fourier Transform and its Applications, Ch. 5. McGraw-Hill, 1999