Computer Aided Laplace Transforms Calculus

Matilde Legua, Centro Politécnico Superior, Universidad de Zaragoza, E-50015 Zaragoza (Spain), mlegua@posta.unizar.es
José Antonio Moraño, Universidad Politécnica de Valencia, ETSID, E-46022 Valencia (Spain), jomofer@mat.upv.es
Luis Manuel Sánchez Ruiz, Universidad Politécnica de Valencia, ETSID, E-46022 Valencia (Spain), lmsr@mat.upv.es

***Abstract* —** *Nowadays it is quite common to organize laboratory sessions to complement theoretical classes of Mathematics in the Engineering Curriculum. The authors have been harmonizing them at Escuela Técnica Superior de Ingeniería del Diseño of Universidad Politécnica de Valencia (UPV) and Centro Politécnico Superior of Universidad de Zaragoza. In this way, the classes of Mathematics become more appealing to engineering students, which feel that mathematical tedious calculations may be smoothened and the subject becomes a friendly tool that allows them to solve engineering problems. The aim of this paper is to present how different mathematical software commonly used in engineering courses handles the calculation of Laplace Transforms. This commonly used operator has many engineering applications and we will survey different techniques that enable to take advantage of the capabilities of the software available to engineering students.*

Index Terms *— Laplace Transforms, Computer Aided Mathematics.*

ENGINEERING NEEDS AND SPANISH SITUATION

If we were to define the present situation of the world with one word, quite likely we would choose changing, which in fact is not far away from challenging. The needs of future Engineering professionals do not escape of the general situation: their needs change through time and adaptability and constant update is nowadays more important than ever. Justo Nieto, UPV Rector, has stated in some of his speeches that he would be happy if universities might leave the last semester in engineering studies in blank, to be decided when the students were actually to follow it.

We do think that he is right in the sense that we must be aware that a lot of our teaching will become obsolete in the next future and we cannot know at this moment what the needs of our students will be when they graduate.

The *Ley de Reforma Universitaria* in the early eighties gave room throughout Spanish universities to the development of different syllabuses in order to prepare them to face the 21st Century within a more advanced and flexible framework. This general law has been updated very recently in December 2001 by means of *the Ley Orgánica 6/2001 de Universidades*.

Because of them, engineering education moved into a credit system, with the theoretical contents of many subjects greatly reduced, increasing the student's capability to choose and design part of his curricula. This meant reviewing the contents of all the subjects and changing the way of teaching if just compressing previous teaching programmes was to be avoided. The solutions found in the different Spanish universities were far away from being uniform.

Here we expose the general orientation given at Escuela Técnica Superior de Ingeniería del Diseño (ETSID) and Centro Politécnico Superior de Zaragoza, the former emerged from Escuela Universitaria de Ingeniería Técnica Industrial (EUITI) de Valencia which was the earlier name of ETSID. We will explain how we have focussed our undergraduate teaching in Mathematics subjects and will particularize in a special topic, the Laplace Transform, one of the most useful tools in Engineering.

Since the late eighties, this school favoured a model of teaching that encouraged the student to develop a series of attitudes within the pedagogical process in such a way that they were strongly interested in their own training.

EUITI laid its foundations for this aim rooting its model on three main bets, [1]:

- Methodology innovation programmes and efficient use of multimedia technology used as a training tool.
- Involvement in international collaborative and exchange programmes.
- Relationship with industry.

This paper can find somehow its origin in the first of the above bets. What at first sight some mathematicians might see as a waste of time by taking part of their classes to the use of computers instead of delivering *magisterial* classes, has later proved to be an efficient way of not wasting time in routine calculations, so getting to recover some of the time lost for theoretical expositions.

Let us mention that Collaborative Programs such as ERASMUS, SOCRATES, LEONARDO, TEMPUS, ALFA and with US institutions have clearly helped to develop and update our subjects, on occasions just to avoid lack of knowledge of some specific issue by our students that were to spend part of their educational period in some partner institution.

On the other hand, our computer classrooms are shared with other subjects: Physics, Computer Fundamentals, Technical Drawing and Languages. There are technicians taking care of them and lecturers just have to worry for the mathematical content of the software used by the students. In addition, if our way of teaching has evolved, so have the exams that the students must pass. Up to the moment, we have not dared to abandon completely classical exams with theoretical/practical requirements, but we have also incorporated the requirement of compulsorily passing exams done with computers.

COMPUTER AIDED MATHEMATICS

In addition, when at the end the decision of incorporating some mathematical software into the classroom is taken, comes a difficult question: Which package should we choose?

As almost everything in life, and teaching is not an exception, the answer is not simple. It depends on what we aim to do, the time we can spend on and of course, thought it should not influence, the preferences of the teacher.

As previously mentioned, in the paper we are going to address a subject commonly studied at Engineering courses, the Laplace Transform and some of its main applications.

Within this context, the topics usually aimed to cover include:

- The step and impulse functions
- The use of convolution functions to study dynamical linear systems
- To solve initial value problems
- The transfer function to study the stability of systems

We expose how three very commonly used mathematical programs can help us by means of its implemented functions. We will explain with some detail the MATLAB® program possibilities, [5], and give the parallel behaviour with the MATHEMATICA© and DERIVE™ programs, [7] and [2]. See [3] and [4] for other related topics.

LAPLACE TRANSFORM MATLAB RELATED COMMANDS

Matlab is equipped with several commands that enable to find direct and inverse Laplace transforms of symbolic expressions. In particular:

- `laplace(f)` finds the Laplace transform of the symbolic function f giving the answer as a function of s whenever the function f depends on t. If the provided function f depends on s, then the Laplace transform is given as a function depending on t.
- `laplace(f,t)` gives the answer as a function depending on t.
- `laplace(f,w,z)` fixes that the function f depend on w and gives the answer as a function depending on z.
- `ilaplace(L)` finds the inverse Laplace transform of the symbolic function L that depends on s giving the answer as a function depending on t. In case that L depends on t, then the answer is given as a function depending on x.
- `ilaplace(L,y)` makes the program to give the answer as a function depending on y.
- `ilaplace(L,y,x)` is used to make clear that L depends on y and we wish an answer as a function depending on x.

Let us see by means of an example how this works to find the Laplace transform of the function $f(t) = \sinh(t)+e^{2t}\cos(3t)$. Later we will check the result by finding the inverse transform.

```
syms t
laplace(sinh(t)+exp(2*t)*cos(3*t))
ans =
1/(s^2-1)+(s-2)/((s-2)^2+9)
pretty(ans)

              1              s - 2
          ---------- + ------------
            2                  2
           s    - 1      (s - 2) + 9
```

```
syms s
ilaplace(1/(s^2-1)+(s-2)/((s-2)^2+9))
ans =
sinh(t)+exp(2*t)*cos(3*t)
```

The step function $H(t\text{-}a)$, defined by $H(t\text{-}a) = 0$ for $t < a$ and $H(t\text{-}a) = 1$ for $t \geq a$, is used in engineering problems and enables an easy representation of functions that appear for a limited time period. In this way if we need to find the Laplace transform of the function $f(t)$ defined by

$$f(t) = 3\sin(t) \text{ for } 0 \leq t < 2\pi, \quad f(t) = 0 \text{ for } t \geq 2\pi,$$

we may write that $f(t) = 3\sin(t)\text{-}3H(t\text{-}2\pi)\sin(t)$. Really we should use

$$f(t) = 3\sin(t)\,(H(t) - H(t\text{-}2\pi)),$$

but for our purpose the former is enough, and we may proceed as follows:

```
syms t
u=sym('Heaviside(t-2*pi)')
u =
Heaviside(t-2*pi)
L=laplace(3*sin(t)*(1-u))
L =
3/(s^2+1)-3*exp(-2*s*pi)/(s^2+1)
```

The Dirac impulse δ is a *distribution* and though it is not a proper function may be conveniently handled taking into account that

$$\delta(t-a) = lim_{\varepsilon\to 0^+}\delta_\varepsilon(t-a), \quad \delta_\varepsilon(t-a) = (1/\varepsilon)(H(t-a) - H(t-(a+\varepsilon)))$$

and that whenever f is a function continuous at an interval I and a belongs to I, $\int_{-\infty}^{\infty} f(t)\delta(t-a)dt = f(a)$. It satisfies that $L(\delta(t)) = 1$. These facts are implemented in the program. For instance, if we have to find the Laplace transform of the function

$$f(t) = e^t\,\delta(t - 3)$$

we may proceed as follows:

```
d=sym('Dirac(t-3)')
d =
Dirac(t-3)
laplace(exp(t)*d)
ans =
exp(3)*exp(-3*s)
```

And if we wish to evaluate the integral

$$\int_{-\infty}^{\infty} \sin(3t)\delta(t - \frac{\pi}{2})\,dt$$

we might just do:

```
pretty(int('sin(3*t)*Dirac(t-pi/2)',t,'-inf','inf'))
                                        -1
```

Another common task when working with Laplace transform is to find the convolution h of two functions f and g, which are piecewise continuous on each interval $[0,b]$, $b>0$,

$$h(t) = \int_0^t f(t-\tau)g(\tau)d\tau = \int_0^t f(\tau)g(t-\tau)d\tau.$$

It may be convenient to make the students to check that the Laplace transform of a convolution is the product of Laplace transforms. For instance with the functions $f(t) = e^t$, $g(t) = \sin(t)$, this means to do the following:

```
pretty(simple(laplace(int('(exp(u)*sin(t-u))','u','0','t'))))
                        1
              ----------------
                              2
              (s - 1) (s  + 1)

pretty(laplace(exp(t))*laplace(sin(t)))
                        1
              ----------------
                           2
              (s - 1) (s  + 1)
```

We have to insist our students on how easy is to take advantage of the above and provide applications of the Laplace transform to solve initial value problems modelled by (a system of) linear differential equations with constant coefficients. We must not leave out any opportunity that the examples we present fit within their studies: electronic, mechanical or chemical engineering. For the sake of briefness we will not do this here. But usually it will not be hard, and yes clearly convenient, to provide a similar example that fulfils the above. Thus if we were to solve a physical system modelled by the initial value problem

$$y''-2*y'+5*y=8*e^{-t},\ \ y(0)=2,\ y'(0)=12, \tag{1}$$

we would proceed as follows:

```
L1=laplace(diff(sym('y(t)'),2)-2*diff(sym('y(t)'))+5*sym('y(t)'))
L1 =
s*(s*laplace(y(t),t,s)-y(0))-D(y)(0)-
2*s*laplace(y(t),t,s)+2*y(0)+5*laplace(y(t),t,s)
L10=simple(subs(L1,{'laplace(y(t),t,s)','y(0)','D(y)(0)'},{'Ly',2,12}))
syms t
L20=laplace(8*exp(-t))
L10 =
s*(s*Ly-2)-8-2*s*Ly+5*Ly
L20 =
8/(s+1)
Ly=solve(L10-L20,'Ly')
Ly =
2*(s^2+5*s+8)/(s+1)/(s^2-2*s+5)
y=ilaplace(Ly)
y =
exp(-t)+exp(t)*cos(2*t)+6*exp(t)*sin(2*t)
```

Let us remark that MATLAB has a MAPLE command that enables to solve (a system of) differential equations by forcing the use of the Laplace transform. For instance in the above we just might have used the following

```
pretty(simple(sym(maple('dsolve({diff(y(t),t$2)-2*diff(y(t),t)+5*y(t)=8*exp(-
t),y(0)=2,(D(y))(0)=12},y(t),method=laplace)'))))
               y(t) = (exp(-2 t) + cos(2 t) + 6 sin(2 t)) exp(t)
```

Solving a system of differential equations is not much more complicated. In this way, if we are to solve

$$\begin{cases} x'(t) + x(t) = y(t) + e^t \\ y'(t) + y(t) = x(t) + e^t \end{cases}, \quad x(0) = 1, \quad y(0) = 1.$$

We might proceed as follows:

```
clear all
syms t
Lx1=laplace(diff(sym('x(t)'))+sym('x(t)'))
Lx1 =
s*laplace(x(t),t,s)-x(0)+laplace(x(t),t,s)
Lx10= subs (Lx1,{'laplace(x(t),t,s)','x(0)'},{'Lx',1})
Lx10 =
s*Lx-1+Lx
Ly1=laplace(diff(sym('y(t)'))+sym('y(t)'))
Ly1 =
s*laplace(y(t),t,s)-y(0)+laplace(y(t),t,s)
Ly10= subs (Ly1,{'laplace(y(t),t,s)','y(0)'},{'Ly',1})
Ly10 =
s*Ly-1+Ly
Lx2= laplace(sym('y(t)')+exp(t))
Lx2 =
laplace(y(t),t,s)+1/(s-1)
Lx20= subs (Lx2, 'laplace(y(t),t,s)', 'Ly')
Lx20 =
(Ly)+1/(s-1)
Ly2= laplace(sym('x(t)')+exp(t))
Ly2 =
laplace(x(t),t,s)+1/(s-1)
Ly20= subs (Ly2, {'laplace(x(t),t,s)'}, {'Lx'})
Ly20 =
Lx+1/(s-1)
syms Lx Ly
[Lx,Ly]=solve(Lx10-Lx20,Ly10-Ly20,Lx,Ly)
Lx =
1/(s-1)
Ly =
1/(s-1)
x=ilaplace(Lx) , y=ilaplace(Ly)
y =
exp(t)
x =
exp(t)
```

Finally we provide an example where the input function requires the use of the step function. Let us solve

$y''-y = f(t)$, $y(0)=1$, $y'(0)=2$, where $f(t)=1$ for $t<3$, $f(t)=t$ for $t>3$.

```
clear all
syms t
L1m=laplace(diff(sym('y(t)'),2)-sym('y(t)'))
L1m0= subs (L1m,{'laplace(y(t),t,s)','y(0)','D(y)(0)'},{'Ly',1,2})
L2m=laplace(1+(t-1)*'Heaviside(t-3)')
L1m =
s*(s*laplace(y(t),t,s)-y(0))-D(y)(0)-laplace(y(t),t,s)
L1m0 =
s*(s*Ly-1)-2-Ly
```

```
L2m =
1/s+2*exp(-3*s)/s+exp(-3*s)/s^2
Ly=simplify(solve(L1m0-L2m,'Ly'));
pretty(Ly)
                     3      2
                    s  + 2 s  + s + 2 exp(-3 s) s + exp(-3 s)
                    -----------------------------------------
                                    2   2
                                   s  (s  - 1)
y=ilaplace(Ly);
pretty(collect(y,'Heaviside(t-3)'))

  (1 + 3/2 exp(t - 3) + 1/2 exp(-t + 3) - t) Heaviside(t - 3) + 2 exp(t) - 1
```

LAPLACE TRANSFORM MATHEMATICA COMMANDS

Mathematica has got several commands that enable a behaviour similar to Matlab. In particular:

- **`LaplaceTransform(f, t, s)`** finds the Laplace transform of the symbolic function *f* giving the answer as a function of s whenever *f* depends on *t*.
- **`InverseLaplaceTransform(L, s, t)`** makes the program to find the inverse transform of *L*, which depends on *s*, giving the answer as a function depending on *t*.

Mathematica also has implemented the Heaviside function $H(t-a)$ by means of **`UnitStep[t-a]`**. A special care must be taken when writing with this program. In this way if we wish to find the Laplace transform of $f(t) = 3\sin(t)-3H(t-2\pi)\sin(t)$ we should write:

```
LaplaceTransform(3 Sin[t] (1-UnitStep[t-2Pi]), t, s)
```

As expected the answer of the program is

$$3(\frac{1}{1+s^2}-\frac{e^{-2\pi s}}{1+s^2})$$

The *Dirac function* is also implemented and thus we will get 1 as answer if we ask the program to evaluate **`LaplaceTransform(DiracDelta[t], t, s)`**.

This program is also efficient dealing with the properties of the Laplace transform in such a way that if we ask it to evaluate

```
LaplaceTransform(y''[t], t, s)
```

we will get as answer

$$s^2\text{LaplaceTransform}(y[t], t, s) – s\ y[0] – y'[0].$$

This fact and an adequate use of the "**`/.`**" command that makes the program to substitute in the expression in front of / the values given after "." allow to solve initial value problems in a way similar to the previously exposed. For instance with (1) we might have proceeded as follows:

```
DE = y''[t]-2y'[t]+5y[t]
5y[t]-2y'[t]+y''[t]
FIN = 8Exp[-t]
8e^-t
IV = {y[0]->2, y'[0]->12}
{y[0]->2, y'[0]->12}
LDE = LaplaceTransform[DE,t,s]/.IV
- 12 - 2s + 5 LaplaceTransform[y[t],t,s]+
s^2 LaplaceTransform[y[t],t,s] -
2 (-2 + s LaplaceTransform[y[t],t,s])
LFIN = LaplaceTransform[FIN,t,s]
 8
---
1+s
Ly = LaplaceTransform[y[t],t,s]
```

```
LaplaceTransform[y[t],t,s]
Solve[LDE == LFIN,Ly]
```

$$\{\{\text{LaplaceTransform}[y[t],t,s]\to \frac{2(8+5s+s^2)}{(1+s)\ (5-2s+s^2)}\}\}$$

```
Y[s_] = %[[1,1,2]]
```

$$\frac{2(8+5s+s^2)}{(1+s)\ (5-2s+s^2)}$$

```
y[t_] = InverseLaplaceTransform[%,s,t]
```

$$\frac{1}{2}\ e^{-t}(2+(1+6i)e^{(2-2i)t}+(1-6i)e^{(2+2i)t})$$

```
Expand[DE]==FIN
True
```

LAPLACE TRANSFORM DERIVE COMMANDS

The DERIVE program may find Laplace transforms $F(s) = L(f(t))$ but in order to avoid receiving a question mark as answer to any simplification, the domain of the variable s must be declared to be strictly bigger than the convergence order of $L(f(t))$. This is a constrain as well as the non-existence of a command to find inverse Laplace transforms.

This means a handicap and the need of knowing the most commonly used inverse transforms to be able to take advantage of the program, [6]. In this point the use of the **`Simplify Expand`** command of a rational expression to expanded it into a sum of simple fractions may be helpful.

Another inconvenient in some extreme situation is the non-existence (up to now) of an implemented Dirac impulse. However the Heaviside function $H(t)$ is available by means of the **`STEP(t)`** command.

Particularly interesting are the **`CHI(a,t,b)`** function defined as $H(t\text{-}b)\text{-}H(t\text{-}a)$ which enables an easy representation of functions defined on finite intervals, and the **`WAVE_SQUARE(t)`** function defined as (-1)^FLOOR(t), where FLOOR(t) stands for the integer part of t.

If we wish to find Laplace transforms with the DERIVE program we will have to load previously the INT_APPS.MTH file. This file includes other commands, all of them related with applications of the integrals, such as Fourier series, arc lengths, areas, volumes and centroids to name some of them. Once loaded the file we will have available the command

- **`LAPLACE(f, t, s)`** which simplifies to the Laplace transform of the expression f depending on t, giving the answer with s as independent variable.

For example, LAPLACE(sinht, t, s) simplifies to $1/(s^2\text{-}1)$, if s has been declared to be a variable whose domain is $]1,+\infty[$. Note that if later we need to find the Laplace transform of sinh(2t), LAPLACE(sinh(2t), t, s) would simplify to "?" unless the variable s is declared to belong to some interval $]a,+\infty[$ with $a \geq 2$ since this is the convergence order of the function sinh(2t). Once this is done the above simplification would give $2/(s^2\text{-}4)$.

Let us remark that for solving initial value problems with a first or second order ordinary differential equation, it may be more convenient to use some of the functions defined in the ODE1.MTH and ODE2.MTH files.

CONCLUSION

We are pretty convinced that once we get our students have got a solid foundation of the fundamentals of Mathematics, its teaching is facilitated with the use of computers. In this paper we have fixed our attention on the Laplace transform and some of its applications, leaving out some of its applications such as the study of the stability of dynamical systems that escaped of the scope that we had fixed ourselves and however are very important in engineering.

After the comparison of methods, MATLAB and MATHEMATICA seem to be the most efficient programs in order to solve initial value problems with the Laplace transforms. DERIVE has its strength in its easiness to be implemented and an easier syntax than the ones used by MATLAB and MATHEMATICA.

ACKNOWLEDGEMENT

The authors are indebted to ETSID for its challenging environment motivating the use of new technologies.

REFERENCES

1. Ballester, E., Sánchez Ruiz, L.M. and Seiz, R. "Teaching Innovation through Collaborative Programs at EUITI Valencia (Spain)", *CD International Conference on Engineering Education,* Oslo 2001, Paper no. 174.
2. DERIVE^{TM} 5. The Mathematical Assistant for Your Personal Computer, *Texas Instruments*, 2000.
3. Legua, M., Moraño, J.A. and Sánchez Ruiz, L.M., "An Iterative Calculus Programming Application", *CD International Conference on Engineering Education,* Taipei 2000 , Paper no. WC1-2.
4. Legua, M., Moraño, J.A. and Sánchez Ruiz, L.M., "Computer Aided Line Integral Calculus in Engineering", *CD International Conference on Engineering Education,* Session 7D5-7 to 7D5-12. EP Innovations, Oslo/Bergen (Norway) 2001.
5. MATLAB®. The language of technical computing, Help Browser, Version 6.5. *The MATHWORKS,* 2002.
6. Sánchez Ruiz, L.M., Legua Fernández, M., and Moraño, J.A., "Matemáticas con DERIVE", *Editorial Universidad Politécnica de Valencia*, 2001.
7. Wolfram, S., " The MATHEMATICA® Book", *Wolfram Media/Cambridge University Press,* 1999.