Review of the book

> U. Graf: **Applied Laplace Transforms and z-Transforms for Scientists and Engineers**. A Computational Approach using a *Mathematica* Package. Basel: Birkhäuser Verlag 2004, x + 500 pp., ISBN 3-7643-2427-9.

for the journal „Zeitschrift für Analysis und ihre Anwendungen".

Laplace transformations play a key role in solving ordinary differential equations (ODE) and hence are an indispensable tool for scientists and engineers that work on practical problems leading to ODE. The Laplace transform translates ODE's into algebraic expressions that can be handled and solved by algebraic techniques. The inverse Laplace transform translates such results back to the original setting.

For theorists there are plenty of questions about existence and proper interpretation of the resulting functions of such direct and inverse transforms. For practical purposes the transformation itself and the proper interpretation of the algebraic results are more central – and it is that matter the book is about.

An *Old Fashioned Approach* (OFA) to these transforms uses long transfer tables, as, e.g., [1, ch. 0.10] or [3, ch. 6], together with a set of properties or rules to relate even more general functions to those in the tables. Nowadays such tedious and error prone computations are more and more done with the support of computational tools.

But using symbolic computational tools, e.g., the framework of a general purpose Computer Algebra System as *Mathematica*, allow also for another approach: Huge tables of „precompiled" transfer results are (re)substituted by the algorithms themselves that were used to compute these precompiled tables. Since this *Computer Supported Approach* (CSA) leads to much more flexibly tailored tools it continues to require a good algorithmic and mathematical understanding to be applied appropriately.

Graf's book is a very valuable exposition of different aspects of such an algorithmic approach to Laplace transforms; and in one count also to z-transforms as a certain discrete variant since the methods are similar. It is based on – and written by the author of – the *Mathematica* package LaplaceAndzTransform (in the following – as in the book – „the Package") that significantly enhances the built-in capabilities of *Mathematica* to perform direct and inverse Laplace transforms. The Package was developed by the author during the last ten years and contains much of his algorithmic expertise. *Mathematica*'s function concept based on pattern matching and a rule based evaluation engine are best suited to add such expertise to the core facilities provided by *Mathematica* itself.

The main difference between OFA and CSA is the number of transformation rules that are applied. In the OFA the number of such rules is restricted to a small list of basic ones that can be (valuably) taught to scientists and engineers to be recognized by their „built-in pattern matcher". In the CSA much of that is done by the computer and can easily be extended to more evolved rules. It is one of the advantages of Graf's package compared to the built-in Laplace transform of *Mathematica* to provide explicit access to these rule based transformations. This allows for much more flexibility and makes the Package a real supporting tool for the experienced „Laplace transformer". Together with the excellent integrating facilities of *Mathematica* such an experienced user can compute direct and inverse Laplace transforms of a huge number of functions also beyond the scope of traditional applications.

Of course, such computations are by no means „automatic", but require methodic competency. It's that kind of „doing mathematics by computer" that Wu Wen-Tsun, the head of the Mathematics Mechanization Research Center of the Chinese Academy of Sciences, called „Mechanized Mathematics" in [2]. Graf points out another aspect of the necessity of such competency as an „important remark" at p. 123: „... The Package can do valuable and very helpful things but also stupid things, as for example computing a result that does not exist. In general it cannot decide whether a certain computation makes sense or not, hence the user should know what he is doing."

Such an understanding of CSA defines the structure of the book. It is a mixture between more theoretical chapters that explain the different concepts and ideas about Laplace transforms and demonstrate their usage and counterpart in the Package by small examples, and more practical ones that give a detailed presentation how Laplace transforms appear in practical applications from automatic control (ch. 5), transmission lines and more general electrical networks (ch. 8), heat conduction and vibration problems (ch. 10) and what tools are supplied by the Package to handle such problems.

In ch. 1 (*Laplace Transformation*, 75 p.) and 2 (*z-Transformation*, 38 p.) Graf explains the theoretical foundation of the basic transformation tools that are implemented in the Package. The explanation uses a sound mathematical language, concentrates on practically relevant topics and requires basic knowledge about analysis in general and the Laplace transform in particular. A main difference to standard texts is the inclusion of Efros' theorem as „a generalization of the convolution theorem that paves the way for many interesting new correspondences" (and the implementation of that meta rule in the Package). A big part of these chapters is devoted to ordinary linear differential and difference equations and introduces a „box supported" notation as common in engineering texts. It reappears in extended form in ch. 5 where Graf discusses *Applications to Automatic Control* in great detail. More advanced topics about Laplace and z-transforms (the complex inversion formula, residue calculations, contour deformation techniques, asymptotic analysis, Laplace's method, sampled data systems, discrete Laplace transforms) and differential and difference equations (anomalous systems, backward initial value problems) are presented in ch. 6 and 7. Even more advanced topics (Duhamel's formulas, Green's functions, fundamental solutions for parabolic PDE, finite Fourier transforms) and theoretical and practical questions about the numerical computation of inverse Laplace transforms are discussed in ch. 11 and 12 resp.

In ch. 3 (*Laplace Transforms with the Package*, 38 p.) and 4 (*z-Transformation with the Package*, 12 p.) Graf explains by references and examples how the Package can and should be used to support practical computations. Beyond tools for direct and inverse Laplace transforms there are also numerous functions to conveniently handle rational expressions (e.g., AmplifyBy, SimplifyRImageFunction, NormalRImageFunction) where *Mathematica* sometimes behaves very rigid due to its overall simplification strategy, to define practically important piecewise functions (e.g., PiecewiseFunction, PeriodicFunction, StepFunction) that extend the *Mathematica* built-in UnitStep and DiracDelta functions (note that piecewise functions defined by the built-in If- or Which-constructs are nicely plotted but *Mathematica* is unable to determine their analytical properties), and to handle differential and difference systems given in a notation with polynomial operator expressions that is best suited for the application of Laplace and z-transforms (e.g., LinearDOpSolve, DiscreteDOpSolve).

Laplace transforms of families of functions in many cases depend on the range of the parameters. The Package offers also functions (e.g., DefPositive, DefInteger) to set the range of such parameters and compute proper results of Laplace transforms that are undefined without such a restriction. Unfortunately these definitions are not (yet) fully integrated with the relatively new built-in assumption mechanism of *Mathematica*. For a long time *Mathematica* supported only a very restricted set of assumptions and only for the Integrate command. With *Mathematica* 5 the concept of assumptions was extended and added as a local option for simplification, e.g., with Simplify and FullSimplify. Graf's assumptions are defined globally (as Maple and MuPAD do). Since a local assumption concept is much more powerful (as local value assignments through substitution are compared to global assignments) it would be valuable to change that part of the Package in a forthcoming release.

For people familiar with theoretical and practical aspects of Laplace transforms the most valuable part are probably the chapters about practical applications to engineering problems in the book and the tools in the Package developed for such applications.

The central application, discussed in the book in ch. 5 (*Applications to Automatic Control*, 50 p.), is about an area where Laplace and z-transformations are widespreadly used. The reader is assumed to be familiar with basic concepts of automatic control theory on the level of an advanced engineer.

In section 5.1 Graf introduces in great detail the standardized box supported graphical notation of controller configurations and relates that notion to its mathematical counterparts – the input-output system and its Laplace transform, the rational transfer function. The Package offers tools to compute these rational transfer functions for a great number of different controller configurations. Section 5.2 is about state-variable analysis for systems with time invariant coupling that are given in matrix notation. The Package contains valuable tools to deal with such transfer matrices in continuous and also discrete contexts. The remaining sections of that chapter address the special but practically relevant case of second order differential systems (5.3), questions of stability (5.4), frequency analysis and filters (5.5) and discrete control systems related with sampled data (5.6).

The other application chapters 8 (*Examples from Electricity*, 30 p.), 9 (*Examples from Control Engineering*, 40 p.) and 10 (*Heat Conduction and Vibration Problems*, 32 p.) discuss different problems about transmission lines with induction and leakage conduction, more general cascading electrical networks and filters, control of pendulums, a DC motor, a magnet-ball-suspension system, sampled data state variable control systems, heat conduction and vibration problems.

Let me conclude this review with some „technical remarks“. First, I had trouble to install the Package from the CD both under Linux (SuSE 9.0) and Windows (XP) since the file names on the CD appeared in lower case (Linux) and upper case (Windows) but the Package requires the proper mixture of lower and upper case letters of various files during loading. For an experienced user it is not difficult but nasty to overcome such trouble.

A second and more serious remark is about „programming style“ used in the examples during the book and also on the CD. One may ask why to bother about that if the examples work and demonstrate the desired effects – and they do very well. But a book as Graf's will teach new-bees not only Laplace transforms but also proper usage of a CA system as *Mathematica*. Hence it should be judged about the quality of presentation of the examples not from a sloppy engineer's point of view but from a strong teacher's one: do the examples demonstrate good programming style, are they composed taking into consideration the full „state of the art“ – even not speaking about that explicitly in the text. Let me give some arguments that Graf's book is slightly weak in this direction.

My first objection is to the widespread use of % to refer to „previous result“. This is very nasty especially within the Plot command if the user is not satisfied with a first setting of the plot parameters and tries to redo the plot. New-bees should be teached not to use % but to store any valuable result in a variable.

My second objection is to the often unmotivated change of notation between functions (as f) and function expressions (as f[x]). It is a great difficulty for most new-bees really to understand *that* there is a difference if it was teached to them on a basis of „you once heared it and now forget it“. But *if* such a sloppy approach can be justified in a math. curriculum for engineers („There are more profound questions that students do not understand“) it *cannot* in a CSA environment, since Diff[f[x],x] and f'[x] (internally spelled as Derivative[1][f][x]) are *very* different statements. f[x]', as used on p. 136, is even syntactically wrong. In this point the author is sometimes very sloppy, e.g., in formula (2.12.) (p. 89), the definition of the Dirac impulse (p. 56–58) or with an expression like $(u(t)f(t) * u(t)g(t))(t)$ (p. 53), to name only some places in the text. Very puzzling is also the notion in formula (1.75) at p. 75, that could be given as easy as $\delta * f = f$. This notational confusion can be traced also in the Package, e.g., if LinearDOpSolve uses a very suspicious mixture of functional and expression notations as in the examples on p. 149.

A third objection is to the output format of the various Solve operators in the Package that is different to the output format of other Solve operators in *Mathematica*. The latter return a nested list of lists of substitution rules (to be interpreted – mathematically sound – as a set of solution tuples), the former only a flat list. It is one of the great advantages of *Mathematica* (and Reduce, Maxima and Axiom) compared to Maple and MuPAD to offer a common format for solutions of single equations and systems of equations. This coherence is broken within the Package.

A third remark is about the comparison between the LaplaceTransform built-in with *Mathematica* and the transformation tools in the Package. I started this review with the remark „*Mathematica*'s function concept ... is best suited to add such expertise to the core facilities provided by *Mathematica* itself". This is, of course, only part of the truth. Development of the Package over more that ten years means also ten years coevolution of *Mathematica* and the Package. During such a long time questions about conceptual flexibility, bifurcation points in the development and coherence do accumulate. Above I already addressed my desire to have the simplification concept of the Package more tightly integrated with that of *Mathematica*, in particular with the new Assumption handling available since v. 5. One of the weak points of Graf's book is that it does not address questions of comparison with the built-in concepts of *Mathematica* at all. Even not for the LaplaceTransform itself. In many places you can do the examples with the InverseLaplaceTransform or other built-in commands (I tried this with *Mathematica* 4.1) not loading the Package.

The reader (and user) may ask at that point „So what? Do I need this Package? Does it really help for my calculations beyond the built-in capabilities of *Mathematica*?". This is a very reasonable question with no global answer. Since the package is not freely available (even not from the author's home page) but only if purchased together with the book, it cannot be examined in advance if it meets special problem solving requirements. There is also no reference to it on the *Mathematica* Information Center at `http://library.wolfram.com/infocenter`. I doubt that this author's policy promotes the widespread use of the Package. The source code of the Package is on the CD, hence the interested reader can at least learn from it, but there are plenty of arguments against the usage of code that is available but not free, see, e.g., `http://creativecommons.org`.

To give an overall judgement of the book: It is a very well written exposition and useful guide for scientists and engineers to basic theoretical and advanced practical topics about Laplace transforms around the software on the CD. A little bit unsatisfactory for the reviewer is the tight bundling of the software and the book since I think that both have value in their own.

A comprehensive overview about the Package and the content of this book is available from the author's home page `http://www.hta-bi.bfh.ch/~gru`.

Literatur

[1] Teubner-Taschenbuch der Mathematik (Hrsg. E. Zeidler). B. G. Teubner Verlag, Stuttgart and Leipzig, 1996.

[2] Wu W.-T.: Mathematics Mechanization. Math. and its Applications, vol. 489. Science Press, Beijing and Kluwer Acad. Publ., Dordrecht, 2000.

[3] Standard Mathematical Tables and Formulae (ed. D. Zwillinger), 30^{th} edition. CRC Press, Boca Raton, 1996.

Leipzig H.-G. Gräbe