Improving CAS Capabilities: New Rules for Computing Improper Integrals

José L. Galán-García*, Gabriel Aguilera-Venegas, María Á. Galán-García, Pedro Rodríguez-Cielos, Iván Atencia-Mc.Killop

Department of Applied Mathematics
University of Málaga (Spain)

Abstract

Different Engineering applications require dealing with improper integral on unbounded domains (improper integrals of the first kind). The classical way for solving these integrals is by means of elementary Calculus (antiderivatives and limit computations) or using numerical approaches. In both situations different problems can arise. For example: the non-existence of antiderivative, the corresponding limit does not exist or integrals depending on parameters which complicate the use of numerical approaches. In order to solve this situation, Advanced Calculus techniques, such as Laplace or Fourier Transforms and the Residue Theorem can be applied. A brief review of the corresponding theoretical frame is included in this paper.

Computer Algebra Systems (Cas) are pieces of software that allow symbolic computations. Nowadays, there are many Cas in the market with an increasing level of sophistication which make them a very powerful tool in Engineering. In this paper, a brief review on the history of Cas evolution is introduced.

In this work, some Advanced Calculus techniques for computing improper integrals of the first kind which cannot be solved with standard procedures are described. These techniques could be easily integrated in almost every Cas. However, we have detected some lacks of these techniques in many widely used Cas.

In this paper, some tests involving improper integrals have been developed. These tests have been used to check the capabilities of some Cas and the results have provided a classification of the Cas with respect to this field.

One of the main contributions in this work is the generation of different rules to compute improper integrals of the first kind. The rules have been classified in specific rules (those coming from the tests) and general rules (those coming from theoretical frames and generalizations of the specific rules). These rules are easy to include in a Cas increasing the facilities of the Cas in the field of improper integral computation.

Keywords: CAS, Improper integrals, Integration rules.

1. Introduction

There are different applications in Engineering that require computing improper integrals of the first kind (integrals defined on an unbounded domain) such as: the work required to move an object from the surface of the earth to infinity (Kynetic Energy), the electric potential created by a charged sphere, the probability density function or the cumulative distribution function in Probability Theory, the values of the Gamma (Γ) function (which is useful to compute the Beta (β) function used to compute trigonometric integrals), Laplace and Fourier Transforms (very useful for example in Differential Equations), ...

Therefore, these applications need to deal with the following types of improper integrals:

$$\int_0^\infty f(t)\,\mathrm{d}t \quad ; \quad \int_{-\infty}^0 f(t)\,\mathrm{d}t \quad \text{and} \quad \int_{-\infty}^\infty f(t)\,\mathrm{d}t$$

*Corresponding author.

Email addresses: jlgalan@uma.es (José L. Galán-García*), gabri@ctima.uma.es (Gabriel Aguilera-Venegas), magalan@ctima.uma.es (María Á. Galán-García), prodriguez@uma.es (Pedro Rodríguez-Cielos), iatencia@ctima.uma.es (Iván Atencia-Mc.Killop)

One possible way of addressing the computation of these improper integrals is using a specific software for symbolic computation in order to obtain the exact result instead of using a numerical software which would provide an approximation. This kind of software which works in an exact and symbolic way is called a CAS (Computer Algebra System).

Usually, CAS use different rules for computing integrals. For example RUBI system, a **ru**le-**b**ased **i**ntegrator developed by Albert Rich [1], is a very powerful system for computing integrals using rules. But when computing improper integrals of the first kind, normally, the basic procedure is used (computing an antiderivative and some limits in infinite). That is:

$$\begin{aligned}
\int_0^{\infty} f(t)\,\mathrm{d}t &= \lim_{m\to\infty}\int_0^{m} f(t)\,\mathrm{d}t = \lim_{m\to\infty}\Big(F(m)-F(0)\Big)\\
\int_{-\infty}^{0} f(t)\,\mathrm{d}t &= \lim_{m\to-\infty}\int_m^{0} f(t)\,\mathrm{d}t = \lim_{m\to-\infty}\Big(F(0)-F(m)\Big)\\
\int_{-\infty}^{\infty} f(t)\,\mathrm{d}t &= \int_{-\infty}^{0} f(t)\,\mathrm{d}t + \int_0^{\infty} f(t)\,\mathrm{d}t \qquad \text{or, in case of convergence,}\\
\int_{-\infty}^{\infty} f(t)\,\mathrm{d}t &= \lim_{m\to\infty}\int_{-m}^{m} f(t)\,\mathrm{d}t = \lim_{m\to\infty}\Big(F(m)-F(-m)\Big) \qquad \text{(Cauchy principal value)}
\end{aligned}$$

where F is an antiderivative of f.

But, what happens if an antiderivative F for f or the above limits do not exist? The standard answer is to use a numerical approach. But, what if, in addition, the integrand f has one or more parameters? In this case, numerical techniques are more difficult to be used.

For example, for $\displaystyle\int_0^{\infty}\frac{\sin(at)}{t}\,\mathrm{d}t$; $\displaystyle\int_0^{\infty}\frac{\cos(at)-\cos(bt)}{t}\,\mathrm{d}t$ or $\displaystyle\int_{-\infty}^{\infty}\frac{\cos(bt)}{(t^2+a^2)^2}\,\mathrm{d}t$ the antiderivatives can not be computed and the integrands depend on different parameters. Hence, the above procedures cannot be used for these examples.

Some of these improper integrals that can not be solved using the basic procedures described above, can be computed using advanced techniques from the Analysis Theory, such as Laplace and Fourier transforms, or the Residue Theorem from the Complex Analysis Theory. In order to make the article to be self-contained, Section 2 is devoted to summarize the theoretical aspects of these techniques which lead to compute some improper integrals. Using these theories, in this paper some rules which allow to compute some improper integrals of the first kind are provided.

Since some CAS cannot compute some improper integrals of the first kind, the application of the provided rules can make a specific CAS to improve its capabilities being able to compute these improper integrals. Therefore, these rules can extend the types of improper integrals that CAS can compute.

In order to check if some CAS can be improved using the above mentioned rules, we have developed some tests with a selection of improper integrals and a selection of CAS.

After a brief historical overview of CAS evolution (Section 3), a description of the tests and the corresponding results are shown in Section 4, while in Section 5 the rules are enumerated.

Finally, conclusions of the developed work are stated in Section 6.

2. Theoretical frame

This section is included in order the article to be self-contained and can be skipped by an acquainted reader. For a deeper study of Laplace and Fourier Transforms and the Residue Theorem and its applications, there are many text books and math encyclopedias on Advanced Calculus such as [2, 3].

As a general remark, in addition of the specific conditions required by Laplace or Fourier Transforms or the Residue Theorem, the convergence of the improper integral to be computed is also a precondition for applying these advanced techniques.

2.1. Laplace Transform

The *Laplace Transform* of a function $f : \mathbb{R}^+ \longrightarrow \mathbb{R}$ is defined as

$$\mathcal{L}\Big[f(t)\Big] = F_{\mathcal{L}}(s) = \int_0^\infty \mathrm{e}^{-st} f(t)\,\mathrm{d}t$$

if such integral exists.

When $F_{\mathcal{L}}(s)$ exists for a given function f, it is valid in the domain $s > a$ for some $a \in \mathbb{R}$, being $s > 0$ the most common case.

See Appendix A for a list of elementary Laplace Transforms and main properties.

2.1.1. Computing improper integrals with Laplace Transforms

The Laplace Transform $F_{\mathcal{L}}(s) = \int_0^\infty \mathrm{e}^{-st} f(t)\,\mathrm{d}t$ is usually valid when $s > 0$ unless an exponential function appears in f, that is: $f(t) = \mathrm{e}^{at} g(t)$ (with g "free" of exponential functions). In this case (exponential case), the domain would be $s > a$.

According with the Laplace Transform definition, evaluating in $s = 0$:

$$\mathcal{L}\Big[f(t)\Big]\Big|_{s=0} = F_{\mathcal{L}}(0) = \int_0^\infty \mathrm{e}^{-st} f(t)\,\mathrm{d}t\Big|_{s=0} = \int_0^\infty f(t)\,\mathrm{d}t$$

Since Laplace Transform usually exists for $s > 0$, in order to compute an improper integral we can use the following result:

$$\int_0^\infty f(t)\,\mathrm{d}t = \lim_{s\to 0^+} \mathcal{L}\Big[f(t)\Big] \tag{1}$$

if both, the Laplace Transform and the limit, exist.

A particular formula can be obtained considering the formula (1) and property $\mathcal{L}7$ (see Appendix A) as follows:

$$\int_0^\infty \frac{f(t)}{t}\,\mathrm{d}t = \lim_{s\to 0^+} \mathcal{L}\left[\frac{f(t)}{t}\right] = \lim_{s\to 0^+} \int_s^\infty \mathcal{L}\Big[f(t)\Big]\,\mathrm{d}u = \int_0^\infty \mathcal{L}\Big[f(t)\Big]\,\mathrm{d}u \tag{2}$$

Remarks:

- If the Laplace Transform exists for $s > a$ (the exponential case previously described), the formulae (1) and (2) will be only applicable when $a < 0$.
- The existence of the Laplace Transform and the corresponding limit when $s \to 0^+$ does not imply the convergence of the improper integral to be computed. That is: in addition of the existence of both, the Laplace Transform and the limit, the convergence of the improper integral must be checked previously. In this case, the formulae (1) and (2) hold.

2.2. Fourier Transform

The *Fourier Transform* (non-unitary, angular frequency) of a function $f : \mathbb{R} \longrightarrow \mathbb{R}$ is defined as

$$\mathcal{F}\Big[f(t)\Big] = F_{\mathcal{F}}(s) = \int_{-\infty}^\infty f(t)\,\mathrm{e}^{-i\,st}\,\mathrm{d}t \qquad\qquad s \in \mathbb{R}$$

if such integral exists.

See Appendix B for a list of elementary Fourier Transforms and main properties.

2.2.1. Computing improper integrals with Fourier Transforms

According with the definition of the Fourier Transform of $f(t)$, evaluating in $s = 0$:

$$\int_{-\infty}^{\infty} f(t)\,\mathrm{d}t = \mathcal{F}\Big[f(t)\Big]\Big|_{s=0} = F_{\mathcal{F}}(0) \tag{3}$$

if the Fourier Transform exists for $s = 0$ and the improper integral $\displaystyle\int_{-\infty}^{\infty} f(t)\,\mathrm{d}t$ is convergent.

2.3. The Residue Theorem

Let C be a closed piecewise smooth positive oriented curve.

Let $f : \mathbb{C} \longrightarrow \mathbb{C}$ be an analytic function with a finite number of isolated singularities $z_1, z_2, \ldots, z_n$ inside C.

Let $\operatorname*{Res}_{z=z_k} f(z)$ be the residue of f in z_k.

Then:

$$\oint_C f(z)\,\mathrm{d}z = 2\pi\,i\left(\operatorname*{Res}_{z=z_1} f(z) + \operatorname*{Res}_{z=z_2} f(z) + \cdots + \operatorname*{Res}_{z=z_n} f(z)\right) = 2\pi\,i\sum_{k=1}^{n}\operatorname*{Res}_{z=z_k} f(z)$$

The residue $\operatorname*{Res}_{z=z_k} g(z)$ in a pole z_k of order n for a function $g(z)$, can be computed by the formula:

$$\operatorname*{Res}_{z=z_k} g(z) = \frac{1}{(n-1)!}\lim_{z\to z_k}\frac{\mathrm{d}^{n-1)}}{\mathrm{d}z^{n-1)}}\Big(\big(z-z_k\big)^n\,g(z)\Big)$$

Let f be an analytic function with a finite set of isolated singularities $S_{\mathcal{P}}$ inside $\mathcal{P} \equiv \mathrm{Im}(z) \geq 0$ none of them being on the real axis.

Let $\mathrm{CPV}(I)$ the Cauchy Principal Value of integral I

Then:

1. If $\displaystyle\lim_{z\to\infty} z\,f(z) = 0 \implies \mathrm{CPV}\left(\int_{-\infty}^{\infty} f(x)\,\mathrm{d}x\right) = \oint_{\mathcal{P}} f(z)\,\mathrm{d}z = 2\pi\,i\sum_{z_k\in S_{\mathcal{P}}}\operatorname*{Res}_{z=z_k} f(z).$

2. If $\displaystyle\lim_{z\to\infty} f(z) = 0 \implies$

$$\begin{cases} \mathrm{CPV}\left(\displaystyle\int_{-\infty}^{\infty} f(x)\cos(ax)\,\mathrm{d}x\right) &= \mathrm{Re}\left(\displaystyle\oint_{\mathcal{P}} f(z)\,\mathrm{e}^{i\,az}\,\mathrm{d}z\right) \\ &= \mathrm{Re}\left(2\pi\,i\displaystyle\sum_{z_k\in S_{\mathcal{P}}}\operatorname*{Res}_{z=z_k} f(z)\,\mathrm{e}^{i\,az}\right) \qquad a>0 \\ \mathrm{CPV}\left(\displaystyle\int_{-\infty}^{\infty} f(x)\sin(ax)\,\mathrm{d}x\right) &= \mathrm{Im}\left(\displaystyle\oint_{\mathcal{P}} f(z)\,\mathrm{e}^{i\,az}\,\mathrm{d}z\right) \\ &= \mathrm{Im}\left(2\pi\,i\displaystyle\sum_{z_k\in S_{\mathcal{P}}}\operatorname*{Res}_{z=z_k} f(z)\,\mathrm{e}^{i\,az}\right) \qquad a>0 \end{cases}$$

Since if an improper integral $I = \displaystyle\int_{-\infty}^{\infty} g(x)\,\mathrm{d}x$ is convergent, its value is equal to $\mathrm{CPV}(I)$, under the above conditions, the residue theorem can be used to compute convergent improper integrals as follows:

$$\int_{-\infty}^{\infty} f(x)\,\mathrm{d}x = 2\pi\,i\sum_{z_k\in S_{\mathcal{P}}}\operatorname*{Res}_{z=z_k} f(z) \tag{4}$$

$$\int_{-\infty}^{\infty} f(x)\cos(ax)\,\mathrm{d}x = \mathrm{Re}\left(2\pi\,i\sum_{z_k\in S_{\mathcal{P}}}\operatorname*{Res}_{z=z_k} f(z)\,\mathrm{e}^{i\,az}\right) \tag{5}$$

$$\int_{-\infty}^{\infty} f(x)\sin(ax)\,\mathrm{d}x = \mathrm{Im}\left(2\pi\, i \sum_{z_k \in S_{\mathcal{P}}} \operatorname*{Res}_{z=z_k} f(z)\,\mathrm{e}^{i\,az}\right) \tag{6}$$

3. CAS

Computer Algebra Systems (Cas) are software packages which deal with algebraical expressions symbolically and in an exact manner. Against of Numerical Software, Cas can expand expressions such as $(a+b)^2$ without needing to instance the values of a and b. That is, a Cas will return $a^2 + 2ab + b^2$ when expanding the previous expression while a Numerical Software would need to know the values of a and b. Regarding the dealing in an exact manner, a Cas would return the Euler's constant e for $\lim_{x\to\infty}\left(1+\frac{1}{x}\right)^x$, while when using a Numerical Software, the solution would be an approximation of e.

In addition of handing symbolic and exact computations, some Cas are also able to deal, for example, with factorization (numbers, polynomials, partial fraction decomposition, ...); symbolic partial and total differentiation; limits and series computation; solution of linear and non-linear numeric equations and inequations (and systems of them); solution of differential and partial differential equations (and systems of them); and many other symbolic computations. In particular, in this paper we are interested in Cas that can manage both indefinite and definite integration.

However, some Cas have limitations on the above described skills. In this paper, we introduce different rules, using Advanced Calculus Theory, which can improve the integration capabilities of some Cas.

Nowadays, Cas are broadly used in many areas such as Mathematics, Engineering or Physics, and in different tasks such as Education or Research and Development. Furthermore, some Numerical Software package have improved their capabilities including Cas modules. For example, MATLAB, the well-known Numerical Software, includes Symbolic Math Toolboxes (based on Cas such as MuPAD or Maple).

3.1. Brief historical review of Cas

About 120 years before the first Cas appears, Augusta Ada Byron King, Countess of Lovelace stated that Maths could be treated by engines not only numerically but also analytically. Specifically, she said:

> "Many persons who are not conversant with mathematical studies imagine that because the business of [Babbage's Analytical Engine] is to give its results in numerical notation, the nature of its processes must consequently be arithmetical and numerical, rather than algebraical and analytical. This is an error. The engine can arrange and combine its numerical quantities exactly as if they were letters or any other general symbols; and in fact it might bring out its results in algebraical notation, were provisions made accordingly."

This cite was added by Ada in Note E of her English translation [4] of the paper [5], originally written in French.

The following is a subjective list of the most relevant Cas ordered by date of first official distribution [6]:

Saint (Symbolic Automatic INTegrator) was developed in Lisp by James Slagle under the supervision of Marvin Minsky in 1961 [7, 8].

Schoonship, developed by Martinus J.G. Veltman in 1963 [9], was one of the first Cas. It ran on an IBM 7094 in December, 1963. The main objective of this Cas was to deal with particle physics.

MATHLAB is a Cas created by C. Engelman in 1964 using Lisp. MATHLAB 68 appeared in 1967 [10]. One of its applications was in Linear electrical circuits by means of symbolic circuit analysis [11].

Reduce was one of the first general purpose Cas. It was developed by Anthony C. Hearn in 1963 and was first distributed in 1968 [12]. 40 years later, in 2008, Anthony decided to make the software freely available under a modified BSD license, available at [13].

Formac (FORmula MAnipulation Compiler), developed by a research group leaded by Jean Sammet of IBM [7], appeared as an extension of Fortran IV, providing the capability of performing both numeric and non-numeric computations and manipulations in the same program [14].

Scratchpad started to be developed in IBM Research by Jim Griesmer and Dick Jenks in 1965 [7]. Scratchpad was implemented in the Lisp programming language using the 360 Lisp system [15].

Alpak was developed by Stan Brown at Bell Labs and was able to deal with rational functions [7, 16].

Sin (Symbolic INtegration) was developed by Joel Moses between 1965 and 1967 using Lisp [7, 17].

Macsyma (Project MAC's SYmbolic MAnipulator), developed by the MIT Project MAC (leaded by Carl Engelman, William Martin and Joel Moses [7]) in the late 60's, was first commercialized in 1978.

Scratchpad II was an abstract datatype language and system developed in 1977 by the Computer Algebra Group, Mathematical Sciences Department, at the IBM Thomas J. Watson Research Center, leaded by R. D. Jenks [18].

muMATH was developed by Albert D. Rich and David R. Stoutemyer in the late 70's [19] using muSIMP (a surface language for LISP. It was first released in 1979 (muMATH-79) and ran on 8080 and Z80 computers while muMATH-80 and muMATH-83 ran on Apple II and DOS respectively.

Maple (MAthematic PLEasure) is one of the most famous general purpose Cas. It started to be developed in 1980 by the Symbolic Calculus Group (with Gaston Gonnet and Keith Geddes, among others) at the University of Waterloo (Ontario, Canada). Its original goal was as a new Educational oriented software which could be run in small computers. The first public available version was Maple 3.3 in 1985. Since 1988, Maple is distributed by Maplesoft as a proprietary software [20].

Mathcad is mainly oriented to Engineering. Its Wysiwyg (What You See Is What You Get) interface makes it possible to handle with mathematical expressions in a graphical format. It was first developed in 1985 by Allen Razdow [21]. Nowadays, it is a proprietary software distributed by Ptc.

GAP (Group, Algorithms, Programming: a system for computational discrete algebra) was started at Lehrstuhl D für Mathematik (LDFM), RWTH Aachen, Germany in 1986. As stated in [22], "Gap is used in research and teaching for studying groups and their representations, rings, vector spaces, algebras, combinatorial structures, and more. The system, including source, is distributed freely".

MathHandbook, former symbMath (Symbolic Mathematics, VisualMath) is an online symbolic math calculator, math handbook and computer algebra system [23]. It was developed by Dr. Weiguang Huang (Sydney, Australia) since 1987.

Mathomatic was a free, portable, general-purpose Cas developed by George Gesslein II in 1986 and discontinued in 2012 [6]. "The Mathomatic source code can be compiled as a symbolic math library with a very simple API, which can be linked to C compatible programs that need to use the Mathomatic symbolic math engine".

Derive is the successor of muMATH, also developed by Albert D. Rich and David R. Stoutemyer. The first release for MS-DOS appeared in 1988 while the first version for Windows systems, in 1996. It was written in muLISP [19]. Although Derive was discontinued in 2007, it is still used widely, mainly in education, since its flexible syntax make it easy to by used by students. Derive was a proprietary software that required very few memory and storage resources. The DUG (Derive User Group) is still alive and the Derive Newsletter is a quarterly journal, edited by Josef Böhm, which still publishes many papers dealing with Derive.

Mathematica is one of the most widely used general purpose Cas developed by Stephen Wolfram in 1986. The first release, Mathematica 1.0, was launched in 1988 [24]. It is a proprietary software distributed by Wolfram Research Inc. and written in Wolfram Language. Its connectivity with many others applications (in both directions) together with the edition of *The Mathematica journal* make Mathematica be a very important tool for Engineering, Education and Research. Since 2009, Wolfram Research Inc. launched Wolfram Alpha (a computational knowledge engine) which is integrated with Mathematica.

Form is a symbolic manipulation system developed by Jos Vermaseren on Nikhef (Dutch National Institute for Subatomic Physics) in 1984 [25]. Form is very popular in the theoretical particle physics community.

PARI/GP is the combination of a C library mainly oriented for Algebraic Number Theory (PARI) and the command line interface (GP). It was developed in 1985 in the mathematics department of Bordeaux I University by a team leaded by Henri Cohen, the co-author of ISABELLE, the antecesor of PARI [26]. PARI/GP is copyrighted but freely available at [27].

FERMAT, developed by R. H. Lewis of Fordham University in 1993, is a CAS mainly oriented toward polynomial and matrix algebra over the rationals $\mathbb{Q}$ and finite fields [28].

MAGMA, the successor of CAYLEY system, was first launched in 1993. It is mainly designed to solve problems in algebra, number theory, algebraic geometry and algebraic combinatorics [29]. It is a proprietary software maintained and distributed by the Computational Algebra Group, School of Mathematics and Statistics, University of Sidney.

KANT/KASH. KANT is a specialized C-library for computations in algebraic number fields. It was developed at the University of Düsseldorf from 1987 until 1993 and at the Technical University Berlin afterwards by a group leaded by Michael Pohst [30]. KASH is a command line interface for KANT. It can be freely downloaded at [31].

SYMBOLIC MATH TOOLBOX, firstly released using MAPLE in 1993, is a toolbox which provides MATLAB (MATRIX LABORATORY, one of the most widely used Numerical software) with CAS funtionalities [32]. MATLAB, a proprietary software by MATHWORKS, has released new versions of SYMBOLIC MATH TOOLBOX using MUPAD since 2008 [33].

AXIOM is a general purpose CAS which evolved from SCRATCHPAD II. IBM sold it to NAG. Later, in 2001, NAG allowed a free software release which is available at [34]. In 2007, AXIOM forked in two open-source branches: FRICAS and OPENAXIOM available at [35] and [36] respectively.

MACAULAY2, developed by Michael Stillman and Daniel Grayson, is a complete rewrite of MACAULAY (by David Bayer and Michael Stillman, from 1983 till 1993) [37]. It was funded by the National Science Foundation since 1992. MACAULAY2 is a CAS mainly oriented to algebraic geometry and commutative algebra and it is freely available at [38].

CoCoA (COmputations in COmmutative Algebra) is a specialized CAS for commutative algebra [39]. One of its most important features is the way it deals with Gröbner bases. This capability makes CoCoA suitable to develop Knowledge-Based Systems (KBS), for example the one described in [40]. CoCoA-4 (first distribution in 1995) and CoCoA-5 (2011) are freely available for research and educational purposes at [41].

MAGNUS is a specific purpose CAS focused on Infinite Group Theory [42]. It was developed under the direction of Gilbert Baumslag in 1997 and discontinued in 2005.

SINGULAR (former BUCHMORA) is a CAS devoted to polynomial computations [43]. Its first public release appeared in 1997. SINGULAR can be freely downloaded at [44].

SYMBOLICC++ uses C++ to develop a general purspose CAS [45]. It was developed by W. H. Steeb in 1997 and can be freely downloaded at [46].

MAXIMA is a branch of MACSYMA maintained by William F. Schelter from 1982 until he passed away in 2001 [47]. MAXIMA is distributed under the GPL license since 1998. Nowadays, a group of users and developers maintain MAXIMA with several updates per year. MAXIMA is available at [47].

GINAC (iterated and recursive acronym for "Ginac Is Not A Cas") is a C++ library that allows symbolic manipulations together with other computations not typically classified in CAS [48]. It was developed in 1999 and can be freely downloaded at [48].

YACAS (Yet Another CAS) is a general purpose CAS. Its first release appeared in 1999. The development was leaded by Ayal Z. Pinkus [49] using its own programming language. It is a free software (GNU GPL) available at [50].

XCAS/GIAC. XCAS is a interface for the C++ library GIAC which is a free (GPL) (CAS) available at [51]. Its first release appeared in 2000, and it was developed by Bernard Parisse et al. GIAC/XCAS is used as a CAS kernel for GEOGEBRA and other CAS and has many interfaces with different environments and C/C++ libraries [51].

SAGEMATH (former SAGE, System for Algebra and Geometry Experimentation) was first released in 2005 [52]. SAGEMATH has an online version (SAGEMATH CLOUD). It uses other open-source package such as NUMPY, SCIPY, MATPLOTLIB, SYMPY, MAXIMA, GAP, FLINT, R and more. It also support programming in PHYTON. SAGEMATH is a free software (GNU GPL) available at [53].

SMATH STUDIO is a CAS with a WYSIWYG (paper-like) editor developed by A. Ivashov [54]. Its first release appeared in 2005. It can be freely downloaded at [54].

RUBI (RUle–Based Integrator) is developed and maintained by Albert D. Rich, the coauthor of MUMATH and DERIVE. After DERIVE was discontinued, Albert started the development of RUBI with the principal aim that this new system will survive changing technology. Nowadays, RUBI 4.9 consists of more than 6200 integration rules in a decision tree [1]. RUBI is freely available at [55]. MATHEMATICA needs to be installed on a computer in order to run RUBI.

CADABRA, designed specifically for the solution of problems in field theory by K. Peeters, was first released in 2007. Its input and output use TEX notation. It has special features for tensor theory used for example in high energy phisics [56]. It is freely (GNU GLP) available at [57].

SYMPY is a free and open source CAS started in 2005 by O. Čertík [58] and first released in 2007. It is a Python library used by many other systems (such as SAGEMATH. SYMPY is freely available (under the "modified BSD" license) at [59].

MUPAD is a CAS developed by the MUPAD research group at the University of Paderbon, Germany. In 2008 MUPAD was purchased by MATHWORKS and its code was included in the SYMBOLIC MATH TOOLBOX for MATLAB [33].

TI-NSPIRE CAS, a proprietary software, is the successor of DERIVE. It is the kernel of the Texas Instruments CAS graphing calculators and also, there exist versions which run in WINDOWS and MAC OS X systems. Its first release was launched in 2007 [60].

MATHICS, created by J. Pöschko in 2011 and maintained by Angus Griffith since 2013, is a free general-purpose online CAS which can also be run locally [61]. It has a MATHEMATICA-like syntax and can export results to LATEX. MATHICS is freely available (GPL) at [61].

SYMJA is a general purpose Java library for symbolic mathematics developed by A. Kramer. Its first public release was launched in 2011 and it is freely available (LGPL) at [62].

SYMBOLISM, created by E. Cavazos in 2013, is a C library for automatic simplification of algebraic expressions. It is freely available (modified BSD license) at [63].

DATAMELT (DMELT) is a free mathematics software that can be used for numeric computation, statistics, symbolic calculations, data analysis and data visualization. It is developed by S. Chekanov who also developed the predecessors JHEPWORK (2005-2013) and SCAVIS (2013-2015). The first release appeared in 2015. DMELT is freely available (GNU GPL, only for non-commercial purposes) at [64].

SYMAT is a computer science and math system written in Java. It supports programming in JAVASCRIPT, PYTHON and JAVA. It was developed by Netsyms Technologies in 2015 and it is freely available for personal use (modified BSD license) at [65].

CALCINATOR, designed by G. J. Paulos, is a CAS for desktop and mobile devices that runs exclusively on the local device browser. The first public release was launched in 2016. In addition of its capabilities as a scientific calculator or math solver, CALCINATOR has a html and LATEX math editor. It can be used online at [66].

3.2. CAS used for the tests

In order to develop our improper integral tests (detailed in section 4), among the CAS described in the previous subsection, we have chosen those ones that we have been able to work with, discarding the CAS with no integration facilities. Specifically, we have developed our tests with the following CAS classified in three categories:

- Discontinued CAS: DERIVE 6.1. This is the only discontinued CAS used in our tests. The main reason for using DERIVE is that the authors have been using this software for years in education [67] and in different applications (such as [68, 69]). Furthermore, we have published different software packages in the User Contributed Math Packages module from DERIVE 6 such as [70].
- Proprietary CAS: MATHEMATICA 11.0.0.0, MAPLE 2015, SYMBOLIC MATH TOOLBOX (in MATLAB R2016a).
- Free CAS: WOLFRAM ALPHA (NO PRO), MAXIMA, SAGEMATH, XCAS.

4. Tests

4.1. Usefulness of the tests

As stated before (section 1), CAS usually computes improper integrals using limits of antiderivatives. This is an effective procedure when both, the antiderivative and the limit, exist. But when working with a function whose antiderivative or the limit can not be computed, this classical approach is not applicable. The advanced techniques described in section 2 can then be used for some kind of improper integrals. With the tests we have developed, we are interested in checking which CAS can compute these special improper integrals and, if not, we provide some rules which allow CAS to compute them. This fact will increase the capabilities of such CAS.

4.2. Description of the tests

In this section we describe the developed tests (computation of improper integrals). For each one, we compute the exact value using the advances techniques described in section 2 in order to check if the different CAS considered achieve the solution. For each test, we will score all CAS with the following values:

- **3 points** if the CAS provide the optimal solution for the test.
- **2 points** if the CAS provide a right solution but non-optimal for the test.
- **1 point** if the CAS fails to provide a right solution for the general constants involved in the test, but it provides a right solution when the constants are instanced.
- **0 points** if the CAS fails in the general and instanced cases of the test.

The tests have been classified in three different tables depending on the theory needed to obtain their results (Laplace Transform, Fourier Transform and the Residue Theorem). Each row in these tables contents the following five elements: a numbered label for the test, the improper integral to compute, the result of the improper integral, specific observations for the test (mainly, the domain for the parameters involved in the improper integral) and information on how the result is derived from formulas (1) to (6) and the elementary Laplace and Fourier transforms and their properties (Appendix A and Appendix B).

Table 1 contains the tests related with Laplace Transform theory.

Table 1: Tests using Laplace Transform.

Test	Integral	Result	Observations	Derived from
T1	$\int_0^\infty \mathrm{e}^{-at} t^x \,\mathrm{d}t$	$\frac{\Gamma(x+1)}{a^{x+1}}$	$x \in (-1,\infty)$; $a \in \mathbb{R}^+$	(1), $\mathcal{L}8$
T2	$\int_0^\infty \left(\mathrm{e}^{-bt} \int_0^t \mathrm{e}^{-au} u^x \,\mathrm{d}u\right) \mathrm{d}t$	$\frac{\Gamma(x+1)}{b(a+b)^{x+1}}$	$x \in (-1,\infty)$; $a, b \in \mathbb{R}^+$	(1), $\mathcal{L}5$, $\mathcal{L}8$
T3	$\int_0^\infty \frac{\sin(at)}{t} \,\mathrm{d}t$	$\frac{\pi}{2}\,\mathrm{sign}(a)$	$a \in \mathbb{R}$	(2)
T4	$\int_0^\infty \mathrm{e}^{-bt} \frac{\sin(at)}{t} \,\mathrm{d}t$	$\arctan\left(\frac{a}{b}\right)$	$a \in \mathbb{R}$; $b \in \mathbb{R}^+$	(2), $\mathcal{L}8$
T5	$\int_0^\infty \left(\mathrm{e}^{-ct} \int_0^t \mathrm{e}^{-bu} \frac{\sin(au)}{u} \mathrm{d}u\right) \mathrm{d}t$	$\frac{\arctan\left(\frac{a}{b+c}\right)}{c}$	$a \in \mathbb{R}$; $b, c \in \mathbb{R}^+$	(1), $\mathcal{L}5$, $\mathcal{L}7$, $\mathcal{L}8$
T6	$\int_0^\infty \frac{\cos(at) - \cos(bt)}{t} \,\mathrm{d}t$	$\ln\left(\left\lvert\frac{b}{a}\right\rvert\right)$	$a, b \in \mathbb{R}^*$	(2), $\mathcal{L}1$
T7	$\int_0^\infty \mathrm{e}^{-ct} \frac{\cos(at) - \cos(bt)}{t} \,\mathrm{d}t$	$\frac{1}{2}\ln\left(\frac{b^2+c^2}{a^2+c^2}\right)$	$a, b \in \mathbb{R}$; $c > 0$	(2), $\mathcal{L}1$, $\mathcal{L}8$
T8	$\int_0^\infty \left(\mathrm{e}^{-dt} \int_0^t \mathrm{e}^{-cu} \frac{\cos(au) - \cos(bu)}{u} \,\mathrm{d}u\right) \mathrm{d}t$	$\frac{1}{2d}\ln\left(\frac{(d+c)^2+b^2}{(d+c)^2+a^2}\right)$	$a, b \in \mathbb{R}$; $c, d \in \mathbb{R}^+$	(1), $\mathcal{L}1$, $\mathcal{L}5$, $\mathcal{L}7$, $\mathcal{L}8$
T9	$\int_0^\infty \frac{\mathrm{e}^{-at} - \cos(bt)}{t} \,\mathrm{d}t$	$\ln\left(\frac{\lvert b\rvert}{a}\right)$	$a \in \mathbb{R}^+$; $b \in \mathbb{R}^*$	(2), $\mathcal{L}1$
T10	$\int_0^\infty \mathrm{e}^{-ct} \frac{\mathrm{e}^{-at} - \cos(bt)}{t} \,\mathrm{d}t$	$\ln\left(\frac{\sqrt{b^2+c^2}}{a+c}\right)$	$a, c > 0$; $b \in \mathbb{R}$	(2), $\mathcal{L}1$, $\mathcal{L}8$
T11	$\int_0^\infty \left(\mathrm{e}^{-dt} \int_0^t \mathrm{e}^{-cu} \frac{\mathrm{e}^{-au} - \cos(bu)}{u} \,\mathrm{d}u\right) \mathrm{d}t$	$\frac{1}{d}\ln\left(\frac{\sqrt{(d+c)^2+b^2}}{a+d+c}\right)$	$b \in \mathbb{R}$; $a, c, d \in \mathbb{R}^+$	(1), $\mathcal{L}1$, $\mathcal{L}5$, $\mathcal{L}7$, $\mathcal{L}8$

Table 1: Continuation of Tests using Laplace Transform.

Test	Integral	Result	Observations	Derived from
T12	$\int_0^\infty \frac{e^{-at} - e^{-bt}}{t}\, dt$	$\ln\left(\frac{b}{a}\right)$	$a, b \in \mathbb{R}^+$	(2), $\mathcal{L}1$
T13	$\int_0^\infty \left(e^{-ct} \int_0^t \frac{e^{-au} - e^{-bu}}{u}\, du\right) dt$	$\frac{1}{c} \ln\left(\frac{b+c}{a+c}\right)$	$a, b \geq 0$; $c \in \mathbb{R}^+$	(1), $\mathcal{L}1$, $\mathcal{L}5$, $\mathcal{L}7$, $\mathcal{L}8$

Table 2 contains the tests related with Fourier Transform theory.

Table 2: Tests using Fourier Transform.

Test	Integral	Result	Observations	Derived from
T14	$\int_{-\infty}^\infty e^{-at^2}\, dt$	$\sqrt{\frac{\pi}{a}}$	$a \in \mathbb{R}^+$	(3)
T15	$\int_{-\infty}^\infty e^{-at^2} \cos(bt)\, dt$	$\sqrt{\frac{\pi}{a}}\, e^{-\frac{b^2}{4a}}$	$a \in \mathbb{R}^+$; $b \in \mathbb{R}$	$\cos(bt) = \frac{e^{ibt} + e^{-ibt}}{2}$, (3), $\mathcal{F}1$, $\mathcal{F}6$
T16	$\int_{-\infty}^\infty e^{-at^2} \cos^2(bt)\, dt$	$\frac{1}{2}\sqrt{\frac{\pi}{a}}\left(1 + e^{-\frac{b^2}{a}}\right)$	$a \in \mathbb{R}^+$; $b \in \mathbb{R}$	$\cos(bt) = \frac{e^{ibt} + e^{-ibt}}{2}$, (3), $\mathcal{F}1$, $\mathcal{F}6$
T17	$\int_{-\infty}^\infty e^{-at^2} \sin^2(bt)\, dt$	$\frac{1}{2}\sqrt{\frac{\pi}{a}}\left(1 - e^{-\frac{b^2}{a}}\right)$	$a \in \mathbb{R}^+$; $b \in \mathbb{R}$	$\sin(bt) = \frac{e^{ibt} - e^{-ibt}}{2i}$, (3), $\mathcal{F}1$, $\mathcal{F}6$
T18	$\int_{-\infty}^\infty \frac{\cos(bt)}{a^2 + t^2}\, dt$	$\frac{\pi\, e^{-a\lvert b\rvert}}{a}$	$a \in \mathbb{R}^+$; $b \in \mathbb{R}$	$\cos(bt) = \frac{e^{ibt} + e^{-ibt}}{2}$, (3), $\mathcal{F}1$, $\mathcal{F}6$
T19	$\int_{-\infty}^\infty \frac{\cos^2(bt)}{a^2 + t^2}\, dt$	$\frac{\pi}{2a}\left(1 + e^{-2a\lvert b\rvert}\right)$	$a \in \mathbb{R}^+$; $b \in \mathbb{R}$	$\cos(bt) = \frac{e^{ibt} + e^{-ibt}}{2}$, (3), $\mathcal{F}1$, $\mathcal{F}6$
T20	$\int_{-\infty}^\infty \frac{\sin^2(bt)}{a^2 + t^2}\, dt$	$\frac{\pi}{2a}\left(1 - e^{-2a\lvert b\rvert}\right)$	$a \in \mathbb{R}^+$; $b \in \mathbb{R}$	$\sin(bt) = \frac{e^{ibt} - e^{-ibt}}{2i}$, (3), $\mathcal{F}1$, $\mathcal{F}6$

Table 2: Continuation of Tests using Fourier Transform.

Test	Integral	Result	Observations	Derived from
T21	$\int_{-\infty}^{\infty} \cos\left(at^2\right)\,\mathrm{d}t$	$\sqrt{\frac{\pi}{2\lvert a\rvert}}$	$a \in \mathbb{R}^*$	(3)
T22	$\int_{-\infty}^{\infty} \cos\left(at^2\right)\cos(bt)\,\mathrm{d}t$	$\sqrt{\frac{\pi}{2\lvert a\rvert}}\left(\cos\left(\frac{b^2}{4a}\right)+\sin\left(\frac{b^2}{4\lvert a\rvert}\right)\right)$	$a \in \mathbb{R}^*$; $b \in \mathbb{R}$	$\cos(bt)=\frac{\mathrm{e}^{i\,bt}+\mathrm{e}^{-i\,bt}}{2}$, (3), $\mathcal{F}1, \mathcal{F}6$
T23	$\int_{-\infty}^{\infty} \sin\left(at^2\right)\,\mathrm{d}t$	$\operatorname{sign}(a)\sqrt{\frac{\pi}{2\lvert a\rvert}}$	$a \in \mathbb{R}^*$	(3)
T24	$\int_{-\infty}^{\infty} \sin\left(at^2\right)\cos(bt)\,\mathrm{d}t$	$\operatorname{sign}(a)\sqrt{\frac{\pi}{2\lvert a\rvert}}\left(\cos\left(\frac{b^2}{4a}\right)-\sin\left(\frac{b^2}{4\lvert a\rvert}\right)\right)$	$a \in \mathbb{R}^*$; $b \in \mathbb{R}$	$\cos(bt)=\frac{\mathrm{e}^{i\,bt}+\mathrm{e}^{-i\,bt}}{2}$, (3), $\mathcal{F}1, \mathcal{F}6$
T25	$\int_{-\infty}^{\infty} \frac{\cos(bt)}{\sqrt{\lvert t\rvert}}\,\mathrm{d}t$	$\sqrt{\frac{2\pi}{\lvert b\rvert}}$	$b \in \mathbb{R}^*$;	$\cos(bt)=\frac{\mathrm{e}^{i\,bt}+\mathrm{e}^{-i\,bt}}{2}$, (3), $\mathcal{F}1, \mathcal{F}6$
T26	$\int_{-\infty}^{\infty} \lvert t\rvert^a \cos(bt)\,\mathrm{d}t$	$\frac{-2\sin\left(\frac{\pi}{2}a\right)\Gamma(a+1)}{\lvert b\rvert^{a+1}}$	$a \in (-1,0)$; $b \in \mathbb{R}^*$	$\cos(bt)=\frac{\mathrm{e}^{i\,bt}+\mathrm{e}^{-i\,bt}}{2}$, (3), $\mathcal{F}1, \mathcal{F}6$

Finally, Table 3 contains the tests related with the Residue Theorem theory, using formulas (4), (5) or (6).

Table 3: Tests using the Residue Theorem.

Test	Integral	Result	Observations	Derived from
T27	$\int_{-\infty}^{\infty} \frac{mx+n}{\left(x^2+a^2\right)^2}\,\mathrm{d}x$	$\frac{n}{2\lvert a\rvert^3}\,\pi$	$a \in \mathbb{R}^*$; $m,n \in \mathbb{R}$	(4)
T28	$\int_{-\infty}^{\infty} \frac{mx+n}{\left(ax^2+bx+c\right)^2}\,\mathrm{d}x$	$\frac{2(2an-mb)}{(4ac-b^2)^{3/2}}\,\pi$	$a \in \mathbb{R}^+$; $m,n \in \mathbb{R}$; $4ac > b^2$	(4)
T29	$\int_{-\infty}^{\infty} \frac{n}{x^2+a^2}\cos(bx)\,\mathrm{d}x$	$\frac{n}{\lvert a\rvert\,\mathrm{e}^{\lvert ab\rvert}}\,\pi$	$a \in \mathbb{R}^*$; $n,b \in \mathbb{R}$	(5)

Table 3: Continuation of Tests using the Residue Theorem.

Test	Integral	Result	Observations	Derived from
T30	$\int_{-\infty}^{\infty} \frac{mx+n}{\left(x^2+a^2\right)^2} \cos(bx)\,\mathrm{d}x$	$\frac{n\,(1+\lvert ab\rvert)}{2\lvert a\rvert^3\,\mathrm{e}^{\lvert ab\rvert}}\,\pi$	$a \in \mathbb{R}^*$; $m, n, b \in \mathbb{R}$	(5)
T31	$\int_{-\infty}^{\infty} \frac{mx+n}{x^2+a^2} \sin(bx)\,\mathrm{d}x$	$\operatorname{sign}(b)\,\frac{m}{\mathrm{e}^{\lvert ab\rvert}}\,\pi$	$a \in \mathbb{R}^*$; $m, n, b \in \mathbb{R}$	(6)
T32	$\int_{-\infty}^{\infty} \frac{mx+n}{\left(x^2+a^2\right)^2} \sin(bx)\,\mathrm{d}x$	$\frac{m\,b}{2\lvert a\rvert\,\mathrm{e}^{\lvert ab\rvert}}\,\pi$	$a \in \mathbb{R}^*$; $m, n, b \in \mathbb{R}$	(6)

4.3. Summary of the test results

The tests described in the previous tables have been run in the selected CAS detailed in subsection 3.2. For each test and each CAS a score of 0, 1, 2 or 3 points have been assigned according to the indications given in subsection 4.2.

The following three tables show the results obtained after running the tests. The lines in the table are ordered by the final score of each CAS.

Table 4 summarizes the result of the tests involving Laplace Transform theory.

Table 4: Scores for tests using Laplace Transforms

Test / CAS	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	Total
MATHEMATICA 11.0.0.0	3	3	3	3	0	3	3	0	3	2	0	3	2	28
MAPLE 2015	3	0	3	3	0	3	3	0	3	3	0	3	3	27
SYMBOLIC MATH TOOLBOX (MATLAB R2016A)	3	3	2	3	0	2	0	0	2	0	0	3	0	18
WOLFRAM ALPHA (NO PRO)	3	1	3	3	0	3	1	0	0	0	0	1	0	15
DERIVE 6.1	3	0	3	3	0	0	0	0	0	0	0	3	0	12
XCAS	1	0	1	0	0	2	0	0	2	0	0	1	0	7
WXMAXIMA 16.04.0	3	0	3	0	0	0	0	0	0	0	0	0	0	6
SAGEMATH	3	0	3	0	0	0	0	0	0	0	0	0	0	6

Table 5 summarizes the result of the tests involving Fourier Transform theory.

Table 5: Scores for tests using Fourier Transforms

Test / CAS	T14	T15	T16	T17	T18	T19	T20	T21	T22	T23	T24	T25	T26	Total
MATHEMATICA 11.0.0.0	3	3	3	3	3	3	2	3	3	3	3	3	3	38
WXMAXIMA 16.04.0	3	3	3	3	3	3	3	2	2	2	2	2	2	33
SAGEMATH	3	3	3	3	3	3	3	2	2	2	2	2	2	33
MAPLE 2015	3	3	3	3	3	3	3	2	2	2	2	2	0	31
WOLFRAM ALPHA (NO PRO)	3	3	3	3	3	0	2	3	1	3	1	3	1	29
SYMBOLIC MATH TOOLBOX (MATLAB R2016A)	3	3	0	0	1	0	0	1	1	1	1	2	2	15
DERIVE 6.1	3	0	0	0	0	0	0	2	0	2	0	3	1	11
XCAS	3	0	0	0	1	1	1	0	0	0	0	1	0	7

Table 6 summarizes the result of the tests involving the Residue Theorem theory.

Table 6: Scores for tests using the Residue Theorem

Test / CAS	T27	T28	T29	T30	T31	T32	Total
MATHEMATICA 11.0.0.0	3	3	3	3	3	3	18
WOLFRAM ALPHA (NO PRO)	3	3	3	3	3	3	18
WXMAXIMA 16.04.0	3	3	3	3	3	3	18
SAGEMATH	3	3	3	3	3	3	18
MAPLE 2015	3	2	2	2	2	2	13
XCAS	1	3	2	2	2	2	12
SYMBOLIC MATH TOOLBOX (MATLAB R2016A)	3	3	1	1	1	1	10
DERIVE 6.1	3	3	0	0	0	0	6

Finally, Table 7 shows the final scores by adding the partial scores obtained in the previous three tables. This final table is ordered by the total score achieved by each CAS and also includes the success percentages obtained by the different CAS. These success percentages are also graphically shown in Figure 1.

Table 7: Total scores

Method / Cas	Laplace Transform	Fourier Transform	The Residue Theorem	Total	Success percentage
Mathematica 11.0.0.0	28	38	18	84	87,5 %
Maple 2015	27	31	13	71	74 %
Wolfram Alpha (no Pro)	15	29	18	62	64,6 %
wxMaxima 16.04.0	6	33	18	57	59,4 %
SageMath	6	33	18	57	59,4 %
Symbolic Math ToolBOX (MATLAB R2016a)	18	15	10	43	44,8 %
Derive 6.1	12	11	6	29	30,2 %
Xcas	7	7	12	26	27,1 %

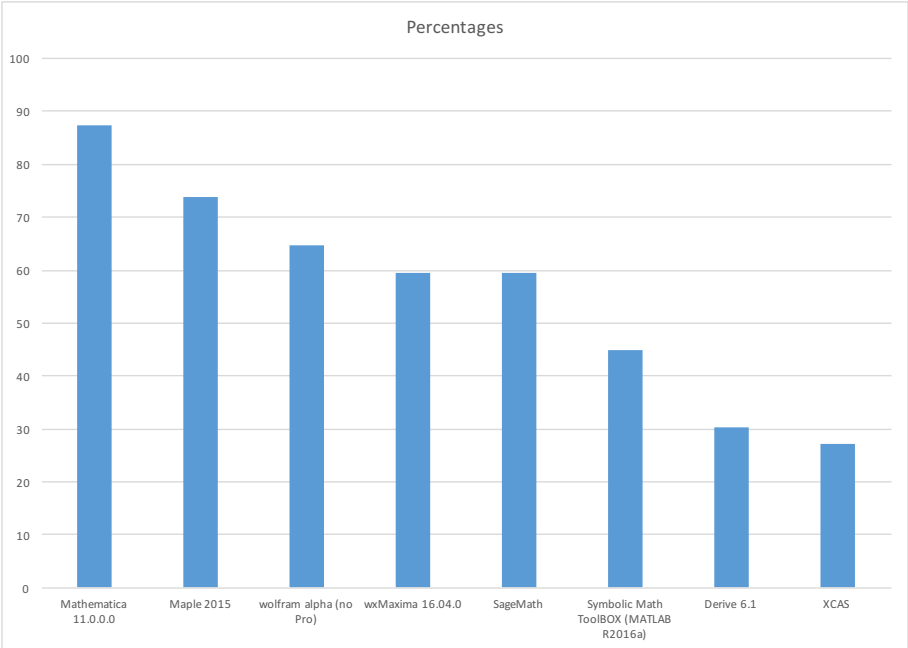

Figure 1: Success percentages

Remark: The final scores and success percentages obtained by the different Cas should not be considered as any kind of evaluation of these Cas. These scores and percentages just show the behavior of the Cas for the very specific improper integrals evaluated in the tests.

5. Rules

In this section a list of rules that can be included in any CAS in order to compute improper integrals are described. These rules constitute one of the most important contributions in this work. Any CAS can use them in order to improve its capabilities since none of the CAS considered for the previous tests have been able to compute all the improper integrals involved. Furthermore, even if a CAS can compute an improper integral without using these rules, the run-time spent by the CAS could be improved when using the appropriate rule.

Please note that convergence of the improper integral must be checked previously to the use of a rule. If the CAS lacks of a procedure to check the convergence, a warning message should be displayed together with the result. That is, the answer to a non-convergence checked improper integral should add to the corresponding result a warning message like, for example: "... if the improper integral is convergent".

The following example remarks the importance of checking the convergence previously to the application of a rule:

The improper integral $\int_0^\infty \sin(ax)\,\mathrm{d}x$ is not convergent for all $a \in \mathbb{R}^*$ since the limit of the antiderivative in infinite, $\lim_{x\to\infty} -\frac{\cos(ax)}{a}$, does not exists. But the Laplace transform of $\sin(ax)$ is $\mathcal{L}\Big[\sin(ax)\Big] = \frac{a}{s^2+a^2}$ and, applying formula (1), the improper integral would be equal to $\lim_{s\to 0^+} \frac{a}{s^2+a^2} = \frac{1}{a}$. Therefore, if the convergence of the improper integral is not previously checked, a CAS could return $\frac{1}{a}$ as the result of a non-convergent improper integral.

To begin with, each test developed in section 4 provides a rule to compute the corresponding improper integral. That is, the tests considered in tables 1, 2 and 3 provide 32 specific rules to compute improper integrals.

In addition, in this section more generic rules for improper integral computation are introduced. Specifically, the following three tables provide some general rules which can be used in more general cases than the ones dealt in the 32 previous tests.

For each rule (a row in the tables), not only the improper integral and its result is shown but also some observations are considered. These observations are mainly related with convergence and with the domain of the parameters involved in the improper integral.

Table 8 shows the general rules that can be obtained directly from the theoretical formulas (1) to (6) stated in section 2.

Table 8: General rules using theoretical results.

Integral	Result	Observations
$\int_0^\infty f(t)\,\mathrm{d}t$	$\lim_{s\to 0^+} \mathcal{L}\Big[f(t)\Big]$	Improper integral must converge; $\mathcal{L}\Big[f(t)\Big]$ must exist; $\lim_{s\to 0^+} \mathcal{L}\Big[f(t)\Big]$ must exist.
$\int_0^\infty \frac{f(t)}{t}\,\mathrm{d}t$	$\int_0^\infty \mathcal{L}\Big[f(t)\Big]\,\mathrm{d}u$	Improper integral must converge; $\lim_{t\to 0^+} \frac{f(t)}{t}$ must exist; $\mathcal{L}\Big[f(t)\Big]$ must exist.
$\int_{-\infty}^\infty f(t)\,\mathrm{d}t$	$\mathcal{F}\Big[f(t)\Big]\Big\|_{s=0} = F_{\mathcal{F}}(0)$	Improper integrals must converge; $\mathcal{F}\Big[f(t)\Big]$ must exist for $s = 0$.

Table 8: Continuation of general rules using theoretical results.

Integral	Result	Observations
$\int_{-\infty}^{\infty} f(x)\,\mathrm{d}x$	$2\pi i \sum_{z_k \in S_{\mathcal{P}}} \operatorname*{Res}_{z=z_k} f(z)$	Improper integrals must converge; $f(z)$ must not have real singularities; $\mathcal{P} \equiv \operatorname{Im}(z) > 0$; $S_{\mathcal{P}} \equiv$ singularities of $f(z)$ in $\mathcal{P}$.
$\int_{-\infty}^{\infty} f(x)\cos(ax)\,\mathrm{d}x$	$\operatorname{Re}\left(2\pi i \sum_{z_k \in S_{\mathcal{P}}} \operatorname*{Res}_{z=z_k} f(z)\,\mathrm{e}^{iaz}\right)$	Improper integrals must converge; $f(z)\,\mathrm{e}^{iaz}$ must not have real singularities; $\mathcal{P} \equiv \operatorname{Im}(z) > 0$; $S_{\mathcal{P}} \equiv$ singularities of $f(z)\,\mathrm{e}^{iaz}$ in $\mathcal{P}$.
$\int_{-\infty}^{\infty} f(x)\sin(ax)\,\mathrm{d}x$	$\operatorname{Im}\left(2\pi i \sum_{z_k \in S_{\mathcal{P}}} \operatorname*{Res}_{z=z_k} f(z)\,\mathrm{e}^{iaz}\right)$	Improper integrals must converge; $f(z)\,\mathrm{e}^{iaz}$ must not have real singularities; $\mathcal{P} \equiv \operatorname{Im}(z) > 0$; $S_{\mathcal{P}} \equiv$ singularities of $f(z)\,\mathrm{e}^{iaz}$ in $\mathcal{P}$.

As stated in section 2, the Laplace and Fourier transforms of many functions cannot be computed directly from their definition, but can be easily computed using their properties. For this reason, in order to use the three first general rules of the previous table in an appropriate way, the Cas also should implement rules for the properties of Laplace and Fourier transforms to enable their computations.

Table 9 shows some general rules involving Fourier Transform.

Table 9: General rules using Fourier Transform.

Integral	Result	Observations	Derived from
$\int_{-\infty}^{\infty} \mathrm{e}^{-at^2}\cos^n(bt)\,\mathrm{d}t$	$\frac{\sqrt{\pi}}{2^n\sqrt{a}} \sum_{k=0}^{n} \binom{n}{k} \mathrm{e}^{-\frac{b^2(n-2k)^2}{4a}}$	$a \in \mathbb{R}^+$; $b \in \mathbb{R}$; $n \in \mathbb{N}$	$\cos(bt) = \frac{\mathrm{e}^{ibt} + \mathrm{e}^{-ibt}}{2}$, $(3), \mathcal{F}1, \mathcal{F}6$
$\int_{-\infty}^{\infty} \mathrm{e}^{-at^2}\sin^n(bt)\,\mathrm{d}t$	$\begin{cases} 0 & \text{if } n \text{ odd} \\ \frac{\sqrt{\pi}}{2^n\sqrt{a}} \sum_{k=0}^{n} \binom{n}{k} (-1)^k \mathrm{e}^{-\frac{b^2(n-2k)^2}{4a}} & \text{if } n \text{ even} \end{cases}$	$a \in \mathbb{R}^+$; $b \in \mathbb{R}$; $n \in \mathbb{N}$	$\sin(bt) = \frac{\mathrm{e}^{ibt} - \mathrm{e}^{-ibt}}{2i}$, $(3), \mathcal{F}1, \mathcal{F}6$
$\int_{-\infty}^{\infty} \frac{\cos^n(bt)}{a^2+t^2}\,\mathrm{d}t$	$\frac{\pi}{2^n a} \sum_{k=0}^{n} \binom{n}{k} \mathrm{e}^{-a\lvert b(n-2k)\rvert}$	$a \in \mathbb{R}^+$; $b \in \mathbb{R}$; $n \in \mathbb{N}$	$\cos(bt) = \frac{\mathrm{e}^{ibt} + \mathrm{e}^{-ibt}}{2}$, $(3), \mathcal{F}1, \mathcal{F}6$
$\int_{-\infty}^{\infty} \frac{\sin^n(bt)}{a^2+t^2}\,\mathrm{d}t$	$\begin{cases} 0 & \text{if } n \text{ odd} \\ \frac{\pi}{2^n a} \sum_{k=0}^{n} \binom{n}{k} (-1)^k \mathrm{e}^{-a\lvert b(n-2k)\rvert} & \text{if } n \text{ even} \end{cases}$	$a \in \mathbb{R}^+$; $b \in \mathbb{R}$; $n \in \mathbb{N}$	$\sin(bt) = \frac{\mathrm{e}^{ibt} - \mathrm{e}^{-ibt}}{2i}$, $(3), \mathcal{F}1, \mathcal{F}6$
$\int_{-\infty}^{\infty} \lvert t\rvert^a \cos^{2n+1}(bt)\,\mathrm{d}t$	$\frac{-2\sin\left(\frac{\pi}{2}a\right)\Gamma(a+1)}{4^n\lvert b\rvert^{a+1}} \sum_{k=0}^{n} \binom{2n+1}{k} \frac{1}{\lvert 2n+1-2k\rvert^{a+1}}$	$a \in (-1, 0)$; $b \in \mathbb{R}^*$; $n \in \mathbb{N}$	$\cos(bt) = \frac{\mathrm{e}^{ibt} + \mathrm{e}^{-ibt}}{2}$, $(3), \mathcal{F}1, \mathcal{F}6$

Finally, table 10 shows some general rules involving the Residue Theorem.

Table 10: General rules using the Residue Theorem.

Integral	Result	Observations	Derived from
$\int_{-\infty}^{\infty} \frac{x^{2m}}{x^{2n}+a^{2n}}\,\mathrm{d}x$	$\frac{\pi\,\csc\left(\frac{\pi(2m+1)}{2n}\right)}{n\,\lvert a\rvert^{2n-2m-1}}$	$a \in \mathbb{R}^*$; $n \in \mathbb{N}^*$; $m \in \mathbb{N}$; $n > m$.	(4)
$\int_{-\infty}^{\infty} \frac{P_k(x)}{(x^2+a^2)^n}\,\mathrm{d}x$	$\frac{2\pi i}{(n-1)!}\lim\limits_{x\to ai}\left[\frac{\mathrm{d}^{n-1)}}{\mathrm{d}x^{n-1)}}\left(\frac{P_k(x)}{(x+ai)^n}\right)\right]$	$P_k(x)$ polynomial of degree $k \leq 2n-2$; $a \in \mathbb{R}^+$; $n \in \mathbb{N}^*$	(4)
$\int_{-\infty}^{\infty} \frac{P_k(x)}{(ax^2+bx+c)^n}\,\mathrm{d}x$	$\frac{2\pi i}{a^n(n-1)!}\lim\limits_{x\to x_1}\left[\frac{\mathrm{d}^{n-1)}}{\mathrm{d}x^{n-1)}}\left(\frac{P_k(x)}{(x-\overline{x_1})^n}\right)\right]$	$P_k(x)$ polynomial of degree $k \leq 2n-2$; $a \in \mathbb{R}^*$; $4ac > b^2$; $x_1 = \frac{-b+\sqrt{4ac-b^2}\,i}{2a}$; $n \in \mathbb{N}^*$	(4)
$\int_{-\infty}^{\infty} \frac{P_k(x)\cos(bx)}{(x^2+a^2)^n}\,\mathrm{d}x$	$\mathrm{Re}\left(\frac{2\pi i}{(n-1)!}\lim\limits_{x\to ai}\left[\frac{\mathrm{d}^{n-1)}}{\mathrm{d}x^{n-1)}}\left(\frac{P_k(x)\,\mathrm{e}^{ibx}}{(x+ai)^n}\right)\right]\right)$	$P_k(x)$ polynomial of degree $k \leq 2n-1$; $a, b \in \mathbb{R}^+$; $n \in \mathbb{N}^*$	(5)
$\int_{-\infty}^{\infty} \frac{P_k(x)\cos(\beta x)}{(ax^2+bx+c)^n}\,\mathrm{d}x$	$\mathrm{Re}\left(\frac{2\pi i}{a^n(n-1)!}\lim\limits_{x\to x_1}\left[\frac{\mathrm{d}^{n-1)}}{\mathrm{d}x^{n-1)}}\left(\frac{P_k(x)\,\mathrm{e}^{i\beta x}}{(x-\overline{x_1})^n}\right)\right]\right)$	$P_k(x)$ polynomial of degree $k \leq 2n-2$; $a \in \mathbb{R}^*$; $\beta \in \mathbb{R}^+$; $4ac > b^2$; $x_1 = \frac{-b+\sqrt{4ac-b^2}\,i}{2a}$; $n \in \mathbb{N}^*$	(5)
$\int_{-\infty}^{\infty} \frac{P_k(x)\sin(bx)}{(x^2+a^2)^n}\,\mathrm{d}x$	$\mathrm{Im}\left(\frac{2\pi i}{(n-1)!}\lim\limits_{x\to ai}\left[\frac{\mathrm{d}^{n-1)}}{\mathrm{d}x^{n-1)}}\left(\frac{P_k(x)\,\mathrm{e}^{ibx}}{(x+ai)^n}\right)\right]\right)$	$P_k(x)$ polynomial of degree $k \leq 2n-1$; $a, b \in \mathbb{R}^+$; $n \in \mathbb{N}^*$	(6)
$\int_{-\infty}^{\infty} \frac{P_k(x)\sin(\beta x)}{(ax^2+bx+c)^n}\,\mathrm{d}x$	$\mathrm{Im}\left(\frac{2\pi i}{a^n(n-1)!}\lim\limits_{x\to x_1}\left[\frac{\mathrm{d}^{n-1)}}{\mathrm{d}x^{n-1)}}\left(\frac{P_k(x)\,\mathrm{e}^{i\beta x}}{(x-\overline{x_1})^n}\right)\right]\right)$	$P_k(x)$ polynomial of degree $k \leq 2n-2$; $a \in \mathbb{R}^*$; $\beta \in \mathbb{R}^+$; $4ac > b^2$; $x_1 = \frac{-b+\sqrt{4ac-b^2}\,i}{2a}$; $n \in \mathbb{N}^*$	(6)

Please, note that some of the general rules described in the previous three tables, generalize some specific rules considered in the tests 1 to 32.

6. Conclusions

The necessity of dealing with improper integrals in different applications for Engineering together with the difficulty of computing them in some cases has been stated in Section 1.

In order the article to be self-contained, Section 2 where the basic theoretical concepts on Laplace and Fourier Transform and the Residue Theorem has been included. This theoretical concepts have been used to compute some improper integrals for which the classical procedures of computations fail.

In Section 3 a brief review of the evolution of CAS has been considered. This brief review includes a short description of each of the most relevant and popular CAS historically ordered.

The use of the theoretical concepts has led to the development of several tests on improper integral computation which have been used to check the capabilities of different CAS in this field. The results obtained have provided a classification of this CAS attending to the scores obtained by each of them in three different blocks: improper integrals computations using Laplace Transform, Fourier Transform or the Residue Theorem. Joining these three blocks, a final score has been assigned to each CAS providing a final classification of them within this field (Section 4) .

The development of these tests has also provided a list of specific rules that CAS could integrate in their system in order to improve their capabilities on improper integral computations. In addition to these specifics rules, in Section 5 a new set of more general rules has been developed. These general rules are classified in three blocks: those obtained from the general theoretical frame, using Fourier Transform and using the Residue Theorem. These general rules are even more relevant for CAS than the specific ones since none of the CAS has all these general rules integrated and, therefore, all CAS could improve even more their capabilities.

The results stated in Section 4 show that none of the CAS considered has been able to solve all the tests. Therefore, the integration of the rules would increase the range of improper integrals that all these CAS can compute. Furthermore, even in the case that a CAS successes in the execution of a test, the run-time spent to get the result can be reduced when using the appropriate rule.

Acknowledgments

This work was partially supported by the national Spanish projects: TIN2015–70266–C2–1–P and TIN2014–59471–P.

We thank the anonymous reviewers for their suggestions and comments, which have improved the quality of the paper.

Appendix A. Elementary Laplace Transforms and properties

In order to compute the Laplace Transform of a function, since its definition may leads to a "difficult" integral, the following elementary Laplace Transforms and properties are used:

$$\mathcal{L}\Big[1\Big] = \frac{1}{s} \qquad s>0 \qquad\qquad \mathcal{L}\Big[e^{at}\Big] = \frac{1}{s-a} \qquad s>a$$

$$\mathcal{L}\Big[t^n\Big] = \frac{n!}{s^{n+1}} \qquad n\in\mathbb{N}^*\ ;\ s>0 \qquad\qquad \mathcal{L}\Big[t^x\Big] = \frac{\Gamma(x+1)}{s^{x+1}} \qquad x>-1\ ;\ s>0$$

$$\mathcal{L}\Big[\sin(at)\Big] = \frac{a}{s^2+a^2} \qquad s>0 \qquad\qquad \mathcal{L}\Big[\cos(at)\Big] = \frac{s}{s^2+a^2} \qquad s>0$$

Let $F(s)=\mathcal{L}[f(t)]\ ; G(s)=\mathcal{L}[g(t)]\ ; a,b\in\mathbb{R}$:

$\mathcal{L}$1: *Linearity* $\quad \mathcal{L}\left[a\cdot f(t)+b\cdot g(t)\right] = a\cdot F(s)+b\cdot G(s)$

$\mathcal{L}$2: *Translation* $\quad$ If $h(t)=\begin{cases} f(t-a) & t>a \\ 0 & t<a \end{cases}$ then $\mathcal{L}\Big[h(t)\Big] = e^{-as}F(s)$

$\mathcal{L}$3: *Scale change* $\quad \mathcal{L}\Big[f(at)\Big] = \frac{1}{a}F\left(\frac{s}{a}\right) \quad \forall\, a>0$

$\mathcal{L}$4: *Laplace Transform of derivatives* $\mathcal{L}\Big[f'(t)\Big] = sF(s) - f(0)$ and, generalizing for higher order derivatives:

$$\mathcal{L}\Big[f^{(n)}(t)\Big] = s^n F(s) - s^{n-1} f(0) - s^{n-2} f'(0) - \cdots - s f^{(n-2)}(0) - f^{(n-1)}(0)$$

$\mathcal{L}$5: *Laplace Transform of integrals* $\mathcal{L}\left[\int_0^t f(u)\,\mathrm{d}u\right] = \dfrac{F(s)}{s}$
(This property adds the restriction $s > 0$ to the restriction of $F(s)$).

$\mathcal{L}$6: *Polynomial multiplication* $\mathcal{L}\Big[t^n f(t)\Big] = (-1)^n \dfrac{\mathrm{d}^n}{\mathrm{d}s^n} F(s)$

$\mathcal{L}$7: *Division by t* if $\lim\limits_{t\to 0^+} \dfrac{f(t)}{t}$ exists, then $\mathcal{L}\left[\dfrac{f(t)}{t}\right] = \int_s^\infty F(u)\,\mathrm{d}u$

$\mathcal{L}$8: *Multiplication by exponentials* if $\alpha \in \mathbb{R}$, then $\mathcal{L}\Big[\mathrm{e}^{\alpha t} f(t)\Big] = F(s-\alpha)$
(The shift in the variable s is also applicable to the domain of the Laplace Transform. That is: if $F(s)$ is valid in $s > a$, the new Laplace Transform is valid in $s - \alpha > a \iff s > a + \alpha$).

$\mathcal{L}$9: *Convolution product* If $f(t) * g(t) = \int_0^t f(u)g(t-u)\,\mathrm{d}u$ then $\mathcal{L}\Big[f(t) * g(t)\Big] = F(s) \cdot G(s)$

$\mathcal{L}$10: *Initial value theorem* If $\lim\limits_{t\to 0^+} f(t)$ exists, then $\lim\limits_{t\to 0^+} f(t) = \lim\limits_{s\to\infty} sF(s)$

$\mathcal{L}$11: *Final value theorem* If $\lim\limits_{t\to\infty} f(t)$ exists, then $\lim\limits_{t\to\infty} f(t) = \lim\limits_{s\to 0^+} sF(s)$

Appendix B. Elementary Fourier Transforms and properties

As for Laplace Transform, in order to compute the Fourier Transform of a function, since its definition may leads to a "difficult" integral, the following elementary Fourier Transforms and properties are used:

$$\mathcal{F}\Big[\mathcal{X}_{[a,b]}(t)\Big] = \frac{\mathrm{e}^{-i\,a s} - \mathrm{e}^{-i\,b s}}{i\,s} \qquad \mathcal{F}\Big[\mathrm{e}^{ct}\,\mathcal{X}_{[a,b]}(t)\Big] = \frac{\mathrm{e}^{b(c-i\,s)} - \mathrm{e}^{a(c-i\,s)}}{c - i\,s} \qquad \mathrm{Re}(c) < 0$$

$$\mathcal{F}\Big[\mathcal{X}_{[-a,a]}(t)\Big] = \frac{2\sin(as)}{s} \qquad \mathcal{F}\Big[\mathrm{e}^{ct}\,\mathcal{X}_{[0,\infty)}(t)\Big] = \frac{1}{i\,s - c} \qquad \mathrm{Re}(c) < 0$$

$$\mathcal{F}\Big[\mathrm{sign}(t)\Big] = \frac{2}{i\,s} \qquad \mathcal{F}\left[\frac{1}{t^2 + c^2}\right] = -\frac{\pi}{c}\,\mathrm{e}^{c|s|} \qquad \mathrm{Re}(c) < 0$$

$$\mathcal{F}\Big[\mathrm{e}^{-at^2}\Big] = \sqrt{\frac{\pi}{a}}\,\mathrm{e}^{-\frac{s^2}{4a}} \qquad a > 0 \qquad \mathcal{F}\Big[\cos\left(at^2\right)\Big] = \sqrt{\frac{\pi}{|a|}}\,\cos\left(\frac{s^2}{4|a|} - \frac{\pi}{4}\right) \qquad a \in \mathbb{R}^*$$

$$\mathcal{F}\Big[\mathrm{e}^{c|t|}\Big] = \frac{-2c}{s^2 + c^2} \qquad \mathrm{Re}(c) < 0 \qquad \mathcal{F}\Big[\sin\left(at^2\right)\Big] = -\mathrm{sign}(a)\sqrt{\frac{\pi}{|a|}}\,\sin\left(\frac{s^2}{4|a|} - \frac{\pi}{4}\right) \qquad a \in \mathbb{R}^*$$

$$\mathcal{F}\left[\frac{1}{\sqrt{|t|}}\right] = \sqrt{\frac{2\pi}{|s|}} \qquad \mathcal{F}\Big[|t|^\alpha\Big] = \frac{-2\sin\left(\dfrac{\pi\alpha}{2}\right)\Gamma(\alpha+1)}{|s|^{\alpha+1}} \qquad -1 < \alpha < 0$$

where $\mathcal{X}_{[a,b]}(t) = \begin{cases} 1 & \text{if } t \in [a,b] \\ 0 & \text{if } t \notin [a,b] \end{cases}$ and $\mathrm{sign}(t) = \begin{cases} 1 & \text{if } t > 0 \\ 0 & \text{if } t = 0 \\ -1 & \text{if } t < 0 \end{cases}$

Let $F(s) = \mathcal{F}[f(t)]$; $G(s) = \mathcal{F}[g(t)]$:

$\mathcal{F}$**1:** *Linearity* $\mathcal{F}\Big[a \cdot f(t) + b \cdot g(t)\Big] = a \cdot F(s) + b \cdot G(s)$ with $a, b \in \mathbb{C}$

$\mathcal{F}$**2:** *Translation* $\mathcal{F}\Big[f(t-a)\Big] = \mathrm{e}^{-i\,a s}\, F(s)$ with $a \in \mathbb{R}$

$\mathcal{F}$**3:** *Scale change* $\mathcal{F}\Big[f(at)\Big] = \dfrac{1}{|a|}\, F\left(\dfrac{s}{a}\right)$ with $a \in \mathbb{R}$

$\mathcal{F}$**4:** *Inversion* $\mathcal{F}\Big[F(t)\Big] = 2\pi\, f(-s)$ if $f(s)$ is an integrable function.

$\mathcal{F}$**5:** *Fourier Transform of derivatives* $\mathcal{F}\Big[f'(t)\Big] = i\, s\, F(s)$ if $f(t)$ and $f'(t)$ are absolutely integrable functions and $\lim\limits_{t\to-\infty} f(t) = \lim\limits_{t\to\infty} f(t) = 0$. Generalizing for higher order derivatives:

$$\mathcal{F}\Big[F^{(n)}(t)\Big] = (i\,s)^n\, \mathcal{F}\Big[F(t)\Big] = (i\,s)^n\, f(s) \qquad \text{if } \lim_{t\to-\infty} f^{(k)}(t) = \lim_{t\to\infty} f^{(k)}(t) = 0 \quad \text{for} \quad k = 1, 2, \ldots, n-1$$

$\mathcal{F}$**6:** *Multiplication by exponentials* $\mathcal{F}\Big[\mathrm{e}^{i\,\alpha t}\, f(t)\Big] = F(s-\alpha)$ with $\alpha \in \mathbb{R}$

$\mathcal{F}$**7:** *Convolution product* if $f(t) * g(t) = \displaystyle\int_{-\infty}^{\infty} f(u)g(t-u)\,\mathrm{d}u$ then $\mathcal{F}\Big[f(t) * g(t)\Big] = F(s) \cdot G(s)$

$\mathcal{F}$**8:** *Parseval's Identity* $\displaystyle\int_{-\infty}^{\infty} |F(s)|^2\,\mathrm{d}s = 2\pi \int_{-\infty}^{\infty} |f(t)|^2\,\mathrm{d}t$

$\mathcal{F}$**9:** *Real function test* $F(t)$ is a function of real values $\iff$ $F(-s) = \overline{F(s)}$

References

References

[1] A. D. Rich, A Rule-based Revolution. Organizing Mathematical Knowledge as a Rule-based Decision Tree, DERIVE Newsletter 100 (2016) 3–6.

[2] M. Hazewinkel, Encyclopedia of Mathematics, Springer, ISBN 978-1-55608-010-4 (2001), on-line version: https://www.encyclopediaofmath.org/index.php/Main_Page (accessed August 29, 2016).

[3] A. Jeffrey, Advanced Engineering Mathematics, Academic Press, ISBN 0–12–382592–X (2002).

[4] L. F. Menabrea, Sketch of the Analytical Engine invented by Charles Babbage, Scientific Memoirs 3 (1843) 666–731.

[5] L. F. Menabrea, Notations sur la machine analytique de Charles Babbage, Bibliothèque Universelle de Genève 82 (1842).

[6] Wikipedia contributors, List of computer algebra systems, Wikipedia, The Free Encyclopedia, https://en.wikipedia.org/wiki/List_of_computer_algebra_systems (accessed December 9, 2016).

[7] J. Moses, Macsyma: A Personal History, Invited Presentation in Milestones in Computer Algebra, Tobago (2008).

[8] J. R. Slagle, A heuristic program that solves symbolic integration problems in freshman calculus: symbolic automatic integrator (SAINT), Ph.D. dissertation, MIT, Cambridge, MA (1961).

[9] M. J. G. Veltman, D. N. Williams, Schoonship '91, arXiv:hep-ph/9306228v2 (1993).

[10] C. Engelman, The legacy of MATHLAB 68, in: Proceedings of the second ACM symposium on Symbolic and algebraic manipulation (SYMSAC '71), (1971) 29–41. DOI: http://dx.doi.org/10.1145/800204.806265

[11] Wikipedia contributors, MATHLAB, Wikipedia, The Free Encyclopedia, https://en.wikipedia.org/wiki/MATHLAB (accessed August 29, 2016).

[12] A. C. Hearn, REDUCE: The First Forty Years, Invited paper presented at the A3L Conference in Honor of the 60th Birthday of Volker Weispfenning (2005).

[13] Reduce, http://reduce-algebra.sourceforge.net/ (accessed August 29, 2016).

[14] E. Bond et al. FORMAC – an experimental formula manipulation compiler, in: Proceedings of the 1964 19th ACM National Conference (1964) 112.101–112.1019. DOI: http://dx.doi.org/10.1145/800257.808916

[15] J. H. Griesmer, R. D. Jenks, SCRATCHPAD/1: An interactive facility for symbolic mathematics, in: SYMSAC '71 Proceedings of the second ACM symposium on Symbolic and algebraic manipulation (1971) 42–58. DOI: http://dx.doi.org/10.1145/800204.806266

[16] W. S. Brown, A language and system for symbolic algebra on a digital computer, in: Proceedings of the First ACM Symposium on Symbolic and Algebraic Manipulation (1966) 501–540. DOI: http://dx.doi.org/10.1145/800005.807955

[17] J. Moses, Symbolic Integration, MAC-TR-47, MIT, Cambridge, MA. (1967).

[18] R. D. Jenks, R. S. Sutor, S. M. Watt, Scratchpad II: An abstract datatype system for mathematical computation, Lecture Notes in Computer Science 296 (1988) 12–37. DOI: http://dx.doi.org/10.1007/3-540-18928-9_3

[19] A. D. Rich, A Brief History of the muMATH/DERIVE Cass, DERIVE Newsletter 40 (2000) 5.

[20] B. W. Char et al. A tutorial introduction to Maple, Journal of Symbolic Computation 2 (2) (1986) 179–200. DOI: http://dx.doi.org/10.1016/S0747-7171(86)80021-9

[21] N. Kajler, N. Soiffer, A Survey of User Interfaces for Computer Algebra Systems, Journal of Symbolic Computation 25 (2) (1998) 127–159. DOI: http://dx.doi.org/10.1006/jsco.1997.0170

[22] The GAP Group, GAP – Groups, Algorithms, and Programming, Version 4.8.4. (2016), http://www.gap-system.org (accessed August 29, 2016).
[23] MATHHANDBOOK, http://mathhandbook.com/ (accessed August 29, 2016).
[24] S. Wolfram, Computer algebra: a 32-year update, in: ISSAC '13 Proceedings of the 38th International Symposium on Symbolic and Algebraic Computation (2013) 7–8.
[25] FORM, http://www.nikhef.nl/~form/ (accessed August 29, 2016).
[26] J. Grabmeier, E. Kaltofen, V. Weispfenning, Computer Algebra Handbook, Springer, ISBN 978–3–642–62988–4 (2003).
[27] PARI/GP, http://pari.math.u-bordeaux.fr/ (accessed August 29, 2016).
[28] FERMAT, http://home.bway.net/lewis/ (accessed August 29, 2016).
[29] MAGMA, http://magma.maths.usyd.edu.au/magma/ (accessed August 29, 2016).
[30] M. Daberkow et al. KANT V4, Journal of Symbolic Computation 24 (3–4) (1997) 267–283. DOI: http://dx.doi.org/10.1006/jsco.1996.0126
[31] KANT/KASH, http://page.math.tu-berlin.de/~kant/kash.html (accessed August 29, 2016).
[32] The MATHWORKS Inc., Symbolic Math Toolbox User's Guide 2.0 (1997).
[33] Wikipedia contributors, MUPAD, Wikipedia, The Free Encyclopedia, https://en.wikipedia.org/wiki/MuPAD (accessed August 29, 2016).
[34] AXIOM, http://www.axiom-developer.org/ (accessed August 29, 2016).
[35] FRICAS, http://fricas.sourceforge.net/ (accessed August 29, 2016).
[36] OPENAXIOM, http://www.open-axiom.org/ (accessed August 29, 2016).
[37] D. Eisenbud et al. Computations in algebraic geometry with Macaulay 2, Algorithms and Computations in Mathematics (8), Springer-Verlag, ISBN 3–540–42230–7 (2001).
[38] MACAULAY2, http://www.math.uiuc.edu/Macaulay2/ (accessed August 29, 2016).
[39] J. Abbott, A. Bigatti, M. Caboara, L. Robbiano, COCOA: Computations in Commutative Algebra, ACM Communications in Computer Algebra 41 (3) (2007) 111–112. DOI: http://dx.doi.org/10.1145/1358190.1358204
[40] E. Roanes-Lozano, J. L. Galán-García, G. Aguilera-Venegas, A portable knowledge-based system for car breakdown evaluation, Applied Mathematics and Computation 267 (2015) 758–770. DOI: http://dx.doi.org/10.1016/j.amc.2014.12.001
[41] COCOA, http://cocoa.dima.unige.it/ (accessed August 29, 2016).
[42] G. Baumslag, C. F. Miller III, Experimenting and computing with infinite groups, DIMACS Series in Discrete Mathematics and Theoretical Computer Science (28) (1997) 19–30.
[43] G. M. Greuel, Description of Singular: A Computer Algebra System for Singularity Theory, Algebraic Geometry and Commutative Algebra, Euromath Bulletin 2 (1996) 161–172. DOI: http://dx.doi.org/10.1145/271130.271203
[44] W. Decker, G. M. Greuel, G. Pfister, H. Schönemann, SINGULAR 4-0-2 — A computer algebra system for polynomial computations (2015), http://www.singular.uni-kl.de (accessed August 29, 2016).
[45] Y. Hardy, K. S. Tan, W. H. Steeb, Computer Algebra with SymbolicC++. World Scientific, Singapore, ISBN: 978–981–283–360–0 (2008).
[46] SYMBOLICC++, http://issc.uj.ac.za/symbolic/symbolic.html (accessed August 29, 2016).
[47] MAXIMA, http://maxima.sourceforge.net/ (accessed August 29, 2016).
[48] GINAC, http://www.ginac.de/ (accessed August 29, 2016).
[49] A. Z. Pinkus, S. Winitzki, YACAS: A Do-It-Yourself Symbolic Algebra Environment, Artificial Intelligence, Automated Reasoning, and Symbolic Computation, Lecture Notes in Computer Science 2385 (2002) 332–336. DOI: http://dx.doi.org/10.1007/3-540-45470-5_29
[50] YACAS, http://www.yacas.org/ (accessed August 29, 2016).
[51] B. Parisse, R. De Graeve, GIAC/XCAS version 1.2.2 (2016), http://www-fourier.ujf-grenoble.fr/~parisse/giac.html (accessed August 29, 2016).
[52] W. Stein, Sage: creating a viable free open source alternative to Magma, Maple, Mathematica, and MATLAB, London Mathematical Society Lecture Note Series 403 (2013) 230–238.
[53] SAGEMATH, http://www.sagemath.org/ (accessed August 29, 2016).
[54] SMATH STUDIO, http://en.smath.info/ (accessed August 29, 2016).
[55] RUBI, http://www.apmaths.uwo.ca/~arich/index.html (accessed August 29, 2016).
[56] L. Brewin, A brief introduction to Cadabra: A tool for tensor computations in General Relativity, Computer Physics Communications 181(3) (2010) 489–498. DOI: http://dx.doi.org/10.1016/j.cpc.2009.10.020
[57] CADABRA, http://cadabra.phi-sci.com/ (accessed August 29, 2016).
[58] D. Joyner, O. Čertík, A. Meurer, B. E. Granger, Open Source Computer Algebra Systems: SYMPY, ACM Commun. Comput. Algebra 45 (3/4) (2012) 225–234. DOI: http://dx.doi.org/10.1145/2110170.2110185
[59] SYMPY, http://www.sympy.org/ (accessed August 29, 2016).
[60] Wikipedia contributors, TI-Nspire series, Wikipedia, The Free Encyclopedia, TI–NSPIRE, https://en.wikipedia.org/wiki/TI-Nspire_series (accessed December 9, 2016).
[61] MATHICS, http://mathics.github.io/ (accessed August 29, 2016).
[62] SYMJA, https://bitbucket.org/axelclk/symja_android_library/ (accessed August 29, 2016).
[63] SYMBOLISM, https://github.com/dharmatech/Symbolism (accessed August 29, 2016).
[64] DATAMELT (DMELT), http://jwork.org/dmelt/ (accessed August 29, 2016).
[65] SYMAT, http://symatapp.com/ (accessed August 29, 2016).
[66] CALCINATOR, http://calcinator.com/ (accessed August 29, 2016).
[67] G. Aguilera-Venegas, J. L. Galán-García, M. Á. Galán-García, P. Rodríguez-Cielos, Teaching semantic tableaux method for propositional classical logic with a CAS, International Journal for Technology in Mathematics Education 22 (2) (2015) 85–92. DOI: http://dx.doi.org/10.1564/tme_v22.2.07
[68] G. Aguilera, J. L. Galán, R. Madrid, A. M. Martínez, Y. Padilla, P. Rodríguez, Automated generation of contrapuntal musical com-

positions using probabilistic logic in derive, Journal of Mathematic and Computer in Simulation 80 (6) (2010) 1200–1211. DOI: http://dx.doi.org/10.1016/j.matcom.2009.04.012

[69] G. Aguilera, J. L. Galán, J. M. García, E. Mérida, P. Rodríguez, An accelerated-time simulation of car traffic on a motorway using a CAS, Journal of Mathematic and Computer in Simulation 104 (2014) 21–30. DOI: http://dx.doi.org/10.1016/j.matcom.2012.03.010

[70] J. L. Galán, P. Rodríguez., M. Á. Galán, Y. Padilla, E. Sánchez, J. Sánchez, Complex analysis.dfw: Complex numbers and functions, User Contributed Math Packages of software DERIVE 6, Texas Instruments (2004).