The function wizard project: Computer Algebra Handbook of Special Functions

E.S. Cheb-Terrab[a,b,c]

[a] *CECM, Department of Mathematics Simon Fraser University, Vancouver, Canada.*

[b] *Department of Theoretical Physics State University of Rio de Janeiro, Brazil.*

[c] *Waterloo Maple Inc.*

Abstract

The computational capabilities of the Maple system concerning special functions are evolving rapidly. The requirement concerning math functions, however, is not just computational: typically, one also needs information on identities, alternative definitions and mathematical properties in general. We usually look for that information in handbooks like *Abramowitz & Stegun*.

It is clear however that part of this information found in textbooks is already in the internal Maple subroutines since the computational capabilities implemented rely on that information. This leads naturally to the idea of a "function wizard" project, whose main purpose is to make this information complete and provide access to each piece of it through a simple interface.

Such a "computer algebra handbook of mathematical functions" - a concept close to that of an alive *function wizard* - can naturally provide more information than that conveyed in textbooks since it can respond to each request by processing whatever (growing number of) mathematical information using (a growing number of) mathematical algorithms.

The *function wizard* project is presently under development and this paper presents a demo of the first prototype with basic functionality already in place.

Introduction

The idea of a computer algebra handbook of special functions is by now relatively old and there is current activity around the idea in various places - see for instance the IMA 2002 conference "Special Functions in the Digital Age" (http://www.ima.umn.edu/digital-age/). Non-interactive information on Special Functions is also available on the web, e.g., the "Automatically Generated Encyclopedia of Special Functions" (2001, http://algo.inria.fr/esf/) and also the recent "Mathematical Functions" website (2001, http://functions.wolfram.com/). With a slightly different concept there is also the "SFTools database of special functions" (2000), more focused on hypergeometric special functions and based on singularity analysis, as well as the "Network of conversion routines for mathematical functions"[1] included in Maple's latest release 8 (2002). At the core of the motivation for all this activity there is the growing relevance of special functions in modeling. Quoting M.Berry,

> *" ... Several years ago I was invited to contemplate being marooned on the proverbial desert island. What book would I most wish to have there, in addition to the Bible and the complete works of Shakespeare? My immediate answer was: Abramowitz and Stegun's Handbook of Mathematical Functions. If I could substitute for the Bible, I would choose Gradsteyn and Ryzhik's Table of Integrals, Series and Products.4 Compounding the impiety, I would give up Shakespeare in favor of Prudnikov, Brychkov and Marichev's of Integrals and Series ... On the island, there would be much time to think about waves on the water that carve ridges on the sand beneath and focus sunlight there; shapes of clouds; subtle tints in the sky... With the arrogance that keeps us theorists going, I harbor the delusion that it would be not too difficult to guess the underlying physics and formulate the governing equations. It is when contemplating how to solve these equations - to convert formulations into explanations - that humility sets in. Then, compendia of formulas become indispensable."*
>
> **Michael Berry, *"Why are special functions special?"*, Physics Today, April 2001**

The other ingredient key to this activity is that computer algebra tools have been developing fast during the last couple of years, up to a point where the idea of a computer handbook of mathematical functions - behaving more like an alive function wizard - looks now like an ambitious though really feasible project. The keyword of such a project is its *interactiviness* with an alive mathematical worksheet, as for instance any GUI of a computer algebra system is. Such a computational function wizard however presents subtle challenges, mainly:

- Taking into account the wide interactive potential a computer has if compared with a paper handbook, what would be an interesting way of taking advantage of this? For instance, what would be the type of information request one would like to addressed?
- As it happens with words in spoken languages, in the mathematical language many functions can be expressed in terms of other functions through appropriate identities. So what would be an appropriate manner of organizing this information avoiding repetition - so that maintenance and extensions are feasible - and at the same time having all the useful relationships available at the tip of our fingers?

Regarding what *"type"* of information request is targeted by this project, our idea is to provide:

- "Human readable" information one would expect to *read* in a text book or help page. Apart from the necessary interactive routines for this purpose, the goal also implies on more than 100 new help pages with an appropriate design and organized summary of information.

[1]This network of conversion routines is also available for Maple 7 on the web as "The Mathematical functions Library" (2001, http://lie.uwaterloo.ca/pub/mathfuncs).

- "Computer readable" information, from higher level requests to tiny bits of static mathematical information. The output should be available in Maple language to any Maple subroutine.
- Answers to "Human meaningful" questions related to possible non-obvious links between pieces of information, where the answer requires processing information through different algorithms.

Regarding the computational strategy to organize the information, our idea is:

- Separate, completely, the information from its client. This is different from the current situation, where for historical reasons the Maple internal subroutines have coded inside the mathematical information they use, repeating it in different places, and not making it available to other routines. The idea instead is to have a clean set of "information bricks" and implement the functionality routines as clients of this new network of information.
- Implement a *single and simple* interface - the ***function_wizard*** command - to accept both human or computer information requests in a form as close as possible to a human spoken request.
- Introduce heuristic algorithms, able to process non-obvious requests typically present in a *human information* requests.
- Move away from the concept of a *help facility* and implement the ***function_wizard*** as something in the middle of a *help* and a *computational* facility.

By now this project has advanced to a point where the skeleton of the interface - the ***function_wizard*** command, as well as the underlying *Dictionary* concept (function conversion algorithms and routines) are in place, ready. Regarding mathematical information completeness, the project just started and there is still a long way to go. The Display concept is still draft (Maple output is available but related useful Maplets are not ready). Regarding a new network of hyperlinked help pages we are considering the use of scripts taking advantage of the new Worksheet Maple 8 package, in order to permit extensions and improvements to proceed in a natural manner - this is a key issue in the project.
A prototype of the ***function_wizard*** is available on the web at http://lie.uwaterloo.ca/pub/mathfuncs, as part of the "Mathematical Functions" library there available. What follows is a commented demo illustrating basic functionality already in place.

Demo

The main concept implemented in this first prototype is a Maple prompt command, ***function_wizard***, able to ***display readable information*** on the screen as well as to ***return computational results*** understandable by other Maple routines. At this point the output is displayed as standard Maple output (GUI java Maplets on the way ..) The goal is - from the scratch - that of an *intuitive routine* which could be used almost without having to *study* help pages.

The *function_wizard* interface

The first thing to consider then was that one should be able to learn about the ***function_wizard*** *by interacting with it*; for instance starting with "no arguments"[2]:

[2]In what follows, the *input* can be recognized by the Maple prompt `>` and Maple's output is displayed using fixed size fonts.

```
> function_wizard( );

The usage is as follows:

> function_wizard( topic, function, ... );

where 'topic' indicates the subject on which advice is required, 'function' is the name of
a Maple function, and '...' represents possible additional input depending on the 'topic'
chosen. To list the possible topics:

> function_wizard( topics );

A short form usage,

> function_wizard( function );

with just the name of the function is also available and displays a summary of information about
the function.
```

Following the information above, the next natural thing to ask is what would be these "`topics`":

```
> function_wizard( topics );

In general or for a given a function, the topics on which information is available are:

    ["ODE", "asymptotic_expansion", "class_info", "classify_function", "differentiation_rule",
    "form", "all forms", "identities", "integral_form", "known_functions", "relate",
    "series", "usage"]

For the functions on which information is available see

> function_wizard( known_functions );
```

Following this advise again, mathematical information is then available for these functions (note ***function_wizard*** understands requests regardless of the case and of the arguments being incompletely written)

```
> function_wizard( known_fu );

* Partial match of "known_fu" against topic "known_functions".

The functions on which information is available via

> function_wizard(function_name);

are:

    [AiryAi, AiryBi, AngerJ, BesselI, BesselJ, BesselK, BesselY, Beta,
    ChebyshevT, ChebyshevU, Chi, Ci, CylinderD, CylinderU, CylinderV, Ei,
    EllipticCE, EllipticCK, EllipticCPi, EllipticE, EllipticF, EllipticK,
    EllipticModulus, EllipticNome, EllipticPi, FresnelC, FresnelS, Fresnelf,
    Fresnelg, GAMMA, GaussAGM, GegenbauerC, HankelH1, HankelH2, HermiteH,
    Hypergeom, JacobiAM, JacobiCD, JacobiCN, JacobiCS, JacobiDC, JacobiDN,
    JacobiDS, JacobiNC, JacobiND, JacobiNS, JacobiP, JacobiSC, JacobiSD,
    JacobiSN, JacobiTheta1, JacobiTheta2, JacobiTheta3, JacobiTheta4, JacobiZeta,
    KelvinBei, KelvinBer, KelvinHei, KelvinHer, KelvinKei, KelvinKer, KummerM,
    KummerU, LaguerreL, LambertW, LegendreP, LegendreQ, LerchPhi, Li,
    LommelS1, LommelS2, MeijerG, Psi, Shi, Si, Ssi, StruveH, StruveL, Sum,
    WeberE, WeierstrassP, WeierstrassPPrime, WeierstrassSigma, WeierstrassZeta,
```

```
WhittakerM, WhittakerW, Zeta, arccos, arccosh, arccot, arccoth, arccsc,
arccsch, arcsec, arcsech, arcsin, arcsinh, arctan, arctanh, bernoulli,
binomial, cos, cosh, cot, coth, csc, csch, dawson, dilog, erf, erfc,
erfi, euler, exp, factorial, harmonic, hypergeom, ln, lnGAMMA,
pochhammer, polylog, sec, sech, sin, sinh, sum, tan, tanh]
```

That is, ***function_wizard*** knows about 128 mathematical functions. The idea is also that, instead of interrupting with "wrong arguments" messages, whenever it is possible gently redirect its user to the topics ***function_wizard*** understands:

```
> function_wizard("something_else...");

No information is available on the topic: "something_else...".
To see the topics for which information is available input:

> function_wizard(topics);
```

Some other interface features are illustrated in what follows, together with the presentation of some of the ***function_wizard*** functionality already implemented.

The "form" of the mathematical functions implemented in Maple

The Maple form of any of these 128 Maple functions can be obtained via

```
> function_wizard(form,BesselJ);
```

$$\mathrm{BesselJ}(a,\, z)$$

The output above uses Maple local variables *a,z*.

```
> has(%,[a,z]);
```

$$\mathit{false}$$

For computational purposes one can pass a list of Maple global variables to be used

```
> function_wizard(form,BesselJ,[A,Z]);
```

$$\mathrm{BesselJ}(A,\, Z)$$

```
> function_wizard(form,BesselJ,[Z]);
```

$$\mathrm{BesselJ}(a,\, Z)$$

```
> has(%,a), has(%,Z);
```

$$\mathit{false},\ \mathit{true}$$

Regarding some functions which admit different number of parameters:

```
> function_wizard("all_forms",Ei);
```

$$\mathrm{Ei}(a,\, z), \mathrm{Ei}(z)$$

Topics are also accepted either as Maple strings or Maple symbols, conveying the idea that for us humans thes requests have a unique meaning in both cases:

```
> function_wizard("form",hypergeom);
```

$$\mathrm{2F1}([a,\, b],\, [c],\, z)$$

In all cases two "text search" rounds - one sensitive and one insensitive - are performed when no exact topic match happens. Only the "best match" (possibly more than one) is displayed on the screen. In these two examples the best match happen for case sensitive

```
> function_wizard( fo, hypergeom);
```

```
* Partial match of "fo" against topic "form".
```

$$2F1([a, b], [c], z)$$

```
> function_wizard( F ,hypergeom);
```

```
Partial or misspelled indication of topic matches more than one available topic. Please select
the correct one from below and call function_wizard again.
```

FresnelC, FresnelS, Fresnelf, Fresnelg

Classifying the mathematical functions

A classification of functions according to the literature is available

```
> function_wizard(classify_function,BesselJ);
```

```
BesselJ belongs to the subclass "Bessel_related" of the class "0F1" and so, in principle, it
can be related to various of the 26 functions of those classes - see
function_wizard("Bessel_related"); and function_wizard("0F1");
```

"Bessel_related", "0F1"

A *quiet* optional argument allows one to get the output avoiding verbosity

```
> function_wizard(classify,BesselJ,quiet);
```

"Bessel_related", "0F1"

Information on function classes is available

```
> function_wizard("0F1");
```

```
The 26 functions in the "0F1" class are particular cases of the
hypergeometric function and are given by:
```

[BesselI, BesselJ, BesselK, BesselY, *HankelH1*, *HankelH2*, *AiryAi*, *AiryBi*, *KelvinBer*, *KelvinBei*, *KelvinKer*, *KelvinKei*, *KelvinHer*, *KelvinHei*, sin, cos, tan, csc, sec, cot, sinh, cosh, tanh, csch, sech, coth]

```
> function_wizard(‘1F1‘);
```

```
The 25 functions in the "1F1" class are particular cases of the
hypergeometric function and are given by:
```

[*KummerM*, *KummerU*, *WhittakerM*, *WhittakerW*, *GAMMA*, *CylinderD*, *CylinderU*, *CylinderV*, *LaguerreL*, *HermiteH*, erf, erfc, *erfi*, FresnelC, FresnelS, Fresnelf, Fresnelg, *dawson*, Ei, *Li*, Si, Shi, Ci, Chi, *Ssi*]

```
> function_wizard("2F1");
```

```
The 24 functions in the "2F1" class are particular cases of the
hypergeometric function and are given by:
```

[*JacobiP*, *GegenbauerC*, *LegendreP*, *LegendreQ*, *ChebyshevT*, *ChebyshevU*, *EllipticE*, *EllipticCE*, *EllipticK*, *EllipticCK*, *LerchPhi*, ln, arcsin, arccos, arctan, arccsc, arcsec, arccot, arcsinh, arccosh, arctanh, arccsch, arcsech, arccoth]

```
> function_wizard("trig");
```

```
The 6 functions in the "trig" class are:
```

$$[\sin, \cos, \tan, \csc, \sec, \cot]$$

```
> function_wizard(classify,exp);
```

```
* Partial match of "classify" against topic "classify_function".

exp belongs to the class "elementary" and so, in principle, it can be related to various of the
26 functions of that class - see function_wizard("elementary");
```

$$\text{“elementary”}$$

The differentiation rules

The differentiation rule of each mathematical function is available via:

```
> function_wizard(diff,BesselJ);
```

```
* Partial match of "diff" against topic "differentiation_rule".
```

$$\frac{\partial}{\partial z}\,\mathrm{BesselJ}(a,\,\mathrm{f}(z)) = (-\mathrm{BesselJ}(a+1,\,\mathrm{f}(z)) + \frac{a\,\mathrm{BesselJ}(a,\,\mathrm{f}(z))}{\mathrm{f}(z)})\,(\frac{d}{dz}\,\mathrm{f}(z))$$

```
> function_wizard("differentiation_rule",Zeta);
```

$$\frac{\partial}{\partial z}\,\zeta(a,\,b,\,\mathrm{f}(z)) = -b\,\zeta(a,\,b+1,\,\mathrm{f}(z))\,(\frac{d}{dz}\,\mathrm{f}(z))$$

The differentiation rule for *argument* is not implemented in Maple:

```
> function_wizard(differentiation_rule,"argument");
```

```
Differentiation rule for "argument" is unknown
```

The ***function_wizard*** is expected to handle such a request of “not available” information, for instance, deriving it, when possible, departing from the definitions:

```
> z = abs(z)*exp(I*argument(z));
```

$$z = |z|\,e^{(\mathrm{argument}(z)\,I)}$$

```
> diff(%,z):
> isolate(%,diff(argument(z),z)):
> expand(simplify(subs(exp(I*argument(z)) = z/abs(z),%))):
> rule := %;
```

$$\mathit{rule} := \frac{d}{dz}\,\mathrm{argument}(z) = -\frac{I}{z} + \frac{\mathrm{abs}(1,\,z)\,I}{|z|}$$

The ODE satisfied by a mathematical function

Besides some exceptions - e.g., the Gamma function - most of the mathematical functions admit a differential representation which is polynomial in the unknown and its derivatives. This differential polynomial representation could be from an ODE to a PDE system as is the case of some elliptic functions. The ***function_wizard*** can be queried about the “ODE satisfied by a given function” via

```
> function_wizard(ODE,BesselJ);
```

$$\mathrm{f}(z) = \mathrm{BesselJ}(a,\,z),\ \left[\frac{d^2}{dz^2}\,\mathrm{f}(z) = -\frac{\frac{d}{dz}\,\mathrm{f}(z)}{z} + \frac{(-z^2+a^2)\,\mathrm{f}(z)}{z^2}\right]$$

In above, z is a local variable. The organization of the output allows testing it directly using Maple's **odetest** command:

```
> odetest(%);
```

$$[0]$$

For computational purposes one can pass a global variable list to be used when construting the ODE

```
> function_wizard(ODE,BesselJ,[x]);
```

$$\mathrm{f}(x)=\mathrm{BesselJ}(a,\,x),\ \left[\frac{d^2}{dx^2}\,\mathrm{f}(x)=-\frac{\frac{d}{dx}\,\mathrm{f}(x)}{x}+\frac{(-x^2+a^2)\,\mathrm{f}(x)}{x^2}\right]$$

```
> has([%%],x);
```

$$\mathit{true}$$

For the same reasons ***function_wizard*** accepts as well an unknown function - say $y(x)$ - as extra argument, in which case the output will be as in

```
> function_wizard(ODE,BesselJ,y(x));
```

$$\mathrm{y}(x)=\mathrm{BesselJ}(a,\,x),\ \left[\frac{d^2}{dx^2}\,\mathrm{y}(x)=-\frac{\frac{d}{dx}\,\mathrm{y}(x)}{x}+\frac{(-x^2+a^2)\,\mathrm{y}(x)}{x^2}\right]$$

If instead of calling ***function_wizard*** with the name of a mathematical function one calls it with a complete math-function call, the request is reinterpreted:

```
> function_wizard( ODE, BesselJ(A,g(z)) );
```

$$\mathrm{f}(z)=\mathrm{BesselJ}(A,\,\mathrm{g}(z)),$$
$$\left[\frac{d^2}{dz^2}\,\mathrm{f}(z)=\frac{(\frac{d}{dz}\,\mathrm{f}(z))\,(\frac{d^2}{dz^2}\,\mathrm{g}(z))}{\frac{d}{dz}\,\mathrm{g}(z)}+(\frac{A^2}{\mathrm{g}(z)^2}-1)\,\mathrm{f}(z)\,(\frac{d}{dz}\,\mathrm{g}(z))^2-\frac{(\frac{d}{dz}\,\mathrm{g}(z))\,(\frac{d}{dz}\,\mathrm{f}(z))}{\mathrm{g}(z)}\right]$$

The integral form of a function

Information on the parameters come together with the output, in a format suitable for use by other routines

```
> function_wizard(integral, BesselJ);

* Partial match of "integral" against topic "integral_form".
```

$$\left[\mathrm{BesselJ}(a,\,z)=\frac{1}{2}\left(\frac{1}{\pi}\int_{-\pi}^{\pi}e^{(-a\,t\,I+z\,\sin(t)\,I)}\,dt\right),\ a{::}\mathit{integer}\right],$$
$$\left[\mathrm{BesselJ}(a,\,z)=\frac{2}{\pi}\int_{0}^{\infty}-\sin(-z\cosh(t)+\frac{a\,\pi}{2})\cosh(a\,t)\,dt,\ z{::}\mathit{real}\right],$$
$$\left[\mathrm{BesselJ}(a,\,z)=\frac{1}{\pi}\int_{0}^{\pi}\cos(a\,t-z\sin(t))\,dt-\left(\frac{\sin(a\,\pi)}{\pi}\int_{0}^{\infty}e^{(-t\,I-z\sinh(t))}\,dt\right),\ 0<\Re(z)\right]$$

Identities satisfied by a function

Most of the information on identities is already organized in the new net of conversion routines of Maple 8. The wizard can access this information in details or in general

```
> function_wizard(identities,KummerM);
```

$$[\text{“raise a”}, \mathrm{KummerM}(a,\, b,\, z) = \frac{(b-2-2\,a-z)\,\mathrm{KummerM}(a+1,\, b,\, z) + (a+1)\,\mathrm{KummerM}(a+2,\, b,\, z)}{b-1-a}], [$$

$$\text{“lower a”}, \mathrm{KummerM}(a,\, b,\, z) = \frac{(b-a+1)\,\mathrm{KummerM}(a-2,\, b,\, z) + (-b+2\,a-2+z)\,\mathrm{KummerM}(a-1,\, b,\, z)}{a-1}], \Bigg[$$

$$\text{“raise b”}, \mathrm{KummerM}(a,\, b,\, z) = \frac{(b+z)\,\mathrm{KummerM}(a,\, b+1,\, z) - \dfrac{z\,\mathrm{KummerM}(a,\, b+2,\, z)\,(b-a+1)}{b+1}}{b}\Bigg], [\text{“lower b”},$$

$$\mathrm{KummerM}(a,\, b,\, z) = \frac{((-2+b+z)\,\mathrm{KummerM}(a,\, b-1,\, z) + \mathrm{KummerM}(a,\, b-2,\, z)\,(-b+2))\,(b-1)}{(b-1-a)\,z}],$$

$$[\text{“mix a and b”},\ \mathrm{KummerM}(a,\, b,\, z) = e^z\,\mathrm{KummerM}(b-a,\, b,\, -z)]$$

```
> function_wizard(identities,LaguerreL);
```

$$[\text{“raise a”}, \mathrm{LaguerreL}(a,\, b,\, z) = \frac{(2\,a+3+b-z)\,\mathrm{LaguerreL}(a+1,\, b,\, z) + (-a-2)\,\mathrm{LaguerreL}(a+2,\, b,\, z)}{a+b+1}], [$$

$$\text{“lower a”}, \mathrm{LaguerreL}(a,\, b,\, z) = \frac{(b-z+2\,a-1)\,\mathrm{LaguerreL}(a-1,\, b,\, z) + (1-a-b)\,\mathrm{LaguerreL}(a-2,\, b,\, z)}{a}], [$$

$$\text{“raise b”}, \mathrm{LaguerreL}(a,\, b,\, z) = \frac{(1+b+z)\,\mathrm{LaguerreL}(a,\, b+1,\, z) - z\,\mathrm{LaguerreL}(a,\, b+2,\, z)}{a+b+1}],$$

$$[\text{“lower b”}, \mathrm{LaguerreL}(a,\, b,\, z) = \frac{(b-1+z)\,\mathrm{LaguerreL}(a,\, b-1,\, z) + (1-a-b)\,\mathrm{LaguerreL}(a,\, b-2,\, z)}{z}],$$

$$[\text{“mix a and b”},\ \mathrm{LaguerreL}(a,\, b,\, z) = \frac{e^z\sin(a\,\pi)\,\mathrm{LaguerreL}(-a-b-1,\, b,\, -z)}{\sin((a+b)\,\pi)}]$$

```
> function_wizard(identities,HermiteH);
```

$$[\text{“raise a”},\ \mathrm{HermiteH}(a,\, z) = \frac{2\,z\,\mathrm{HermiteH}(a+1,\, z) - \mathrm{HermiteH}(a+2,\, z)}{2\,a+2}], [\text{“lower a”},$$

$$\mathrm{HermiteH}(a,\, z) = 2\,z\,\mathrm{HermiteH}(a-1,\, z) + 2\,\mathrm{HermiteH}(a-2,\, z) - 2\,\mathrm{HermiteH}(a-2,\, z)\,a]$$

Series expansion of mathematical functions

Information concerning series expansion is found in the system since previous releases, though in incomplete form. The project includes updating these routines and basing ***function_wizard***'s output on that of the Maple **series** command. So currently ***function_wizard*** returns as in the following examples:

```
> function_wizard(series,BesselJ);
```

```
The system is unable to compute the "series" for BesselJ
```

```
> function_wizard(series,exp);
```

$$\mathrm{series}(e^z,\, z,\, 4) = 1 + z + \frac{1}{2}\, z^2 + \frac{1}{6}\, z^3 + \mathrm{O}(z^4)$$

Series are sometimes available regarding more than one parameter

```
> function_wizard(series,EllipticF);
```

$$\mathrm{series}(\mathrm{EllipticF}(a,\, z),\, a,\, 4) = a + (\frac{z^2}{6} + \frac{1}{6})\, a^3 + \mathrm{O}(a^5),$$

$$\mathrm{series}(\mathrm{EllipticF}(a,\, z),\, z,\, 4) = \arcsin(a) + \frac{1}{6}\, a^3\, \mathrm{2F1}([\frac{1}{2},\, \frac{3}{2}],\, [\frac{5}{2}],\, a^2)\, z^2 + \mathrm{O}(z^4)$$

```
> function_wizard(series,"hypergeom");
```

$$\mathrm{series}(\mathrm{2F1}([a,\, b],\, [c],\, z),\, z,\, 4) =$$
$$1 + \frac{a\, b}{c}\, z + \frac{a\, b\, (a+1)\, (b+1)}{2\, c\, (c+1)}\, z^2 + \frac{a\, b\, (a+1)\, (b+1)\, (a+2)\, (b+2)}{6\, c\, (c+1)\, (c+2)}\, z^3 + \mathrm{O}(z^4)$$

Asymptotic series expansion are known to ***function_wizard*** through the **asympt** Maple command; the project includes updating this command too

```
> function_wizard(series,Ei);
```

```
The system is unable to compute the "series" for Ei
```

```
> function_wizard(asymptotic_expansion,Ei);
```

$$\mathrm{asympt}(\mathrm{Ei}(a,\, z),\, z,\, 4) = \frac{\frac{1}{z} - \frac{a!}{(a-1)!\, z^2} + \frac{(a+1)!}{(a-1)!\, z^3} + \mathrm{O}(\frac{1}{z^4})}{e^z}$$

For the Hyperbolic cosine integral both *series* and *asympt* work fine

```
> function_wizard(series,Chi);
```

$$\mathrm{series}(\mathrm{Chi}(z),\, z,\, 4) = (\gamma + \ln(z)) + \frac{1}{4}\, z^2 + \mathrm{O}(z^4)$$

```
> function_wizard(asym,Chi);
```

```
* Partial match of "asym" against topic "asymptotic_expansion".
```

$$\mathrm{asympt}(\mathrm{Chi}(z),\, z,\, 4) = \frac{\sinh(z)}{z} + \frac{\cosh(z)}{z^2} + \frac{2\sinh(z)}{z^3} + \mathrm{O}(\frac{1}{z^4})$$

Relating two mathematical functions

At present the relation between two mathematical functions, when it exists, can be computed depending on the case.

```
> function_wizard(relate,sin,hypergeom);
```

$$\sin(z) = z\, \mathrm{0F1}([],\, [\frac{3}{2}],\, -\frac{z^2}{4})$$

Sometimes there are problems with the automatic conversions happening when a function is called. For example, in Maple the Bessel functions are automatically converted to trigonometric form whenever that is possible; this makes ***function_wizard*** fail in relating **sin** to **BesselJ**

```
> function_wizard(relate, sin, BesselJ );
```

```
Unable to establish a relation between sin and BesselJ
```

If we "avoid" this automatic conversion, for instance, by turning **BesselJ** inert, the relation is established

```
> unprotect(BesselJ); unassign(BesselJ);
> function_wizard(relate,sin,BesselJ);
```

$$\sin(z) = \frac{1}{2}\sqrt{z}\,\sqrt{\pi}\,\sqrt{2}\,\mathrm{BesselJ}(\frac{1}{2},\, z)$$

On the other hand, most well known relations between special functions can be computed normally. For instance in the group of functions admitting a 0F1 hypergeometric representation,

```
> function_wizard(relate,AiryAi,BesselJ);
```

$$\mathrm{AiryAi}(z) = \frac{1}{3}\,((-z)^{(3/2)})^{(1/3)}\,\mathrm{BesselJ}(\frac{-1}{3},\, \frac{2\,(-z)^{(3/2)}}{3}) - \frac{1}{3}\,\frac{z\,\mathrm{BesselJ}(\frac{1}{3},\, \frac{2\,(-z)^{(3/2)}}{3})}{((-z)^{(3/2)})^{(1/3)}}$$

```
> function_wizard(relate,KummerU,BesselK);
```

$$\mathrm{BesselK}(a,\, z) = \frac{\sqrt{\pi}\,(2\,z)^a\,\mathrm{KummerU}(a+\frac{1}{2},\, 2\,a+1,\, 2\,z)}{e^z}$$

Regarding functions admitting a 1F1 hypergeometric representation,

```
> function_wizard(relate,HermiteH,erf);
```

$$\mathrm{erf}(z) = \frac{-\dfrac{2\,\mathrm{HermiteH}(-1,\, z)}{e^{(z^2)}} + \sqrt{\pi}}{\sqrt{\pi}}$$

```
> function_wizard(relate,HermiteH,LaguerreL);
```

$$\mathrm{HermiteH}(a,\, z) = -\left(-\frac{\mathrm{LaguerreL}(\frac{a}{2},\, \frac{-1}{2},\, z^2)}{\mathrm{binomial}(\frac{a}{2}-\frac{1}{2},\, \frac{a}{2})\,\Gamma(\frac{1}{2}-\frac{a}{2})} + \frac{2\,z\,\mathrm{LaguerreL}(\frac{a}{2}-\frac{1}{2},\, \frac{1}{2},\, z^2)}{\mathrm{binomial}(\frac{a}{2},\, \frac{a}{2}-\frac{1}{2})\,\Gamma(-\frac{a}{2})}\right)\sqrt{\pi}\,2^a$$

Some examples of the 2F1 hypergeometric group are

```
> function_wizard(relate,JacobiP,LegendreP);
```

$$\mathrm{LegendreP}(a,\, z) = \mathrm{JacobiP}(-a-1,\, 0,\, 0,\, z)$$

```
> function_wizard(relate,GegenbauerC,LegendreP);
```

$$\mathrm{LegendreP}(a,\, z) = \mathrm{GegenbauerC}(a,\, \frac{1}{2},\, z)$$

```
> function_wizard(relate,hypergeom,arcsin);
```

$$\arcsin(z) = z\,\mathrm{2F1}([\frac{1}{2},\, \frac{1}{2}],\, [\frac{3}{2}],\, z^2)$$

```
> function_wizard(relate,GegenbauerC, hypergeom);
```

$$\mathrm{GegenbauerC}(a,\, b,\, z) = \frac{\Gamma(a+2\,b)\,\mathrm{2F1}([-a,\, a+2\,b],\, [\frac{1}{2}+b],\, -\frac{z}{2}+\frac{1}{2})}{\Gamma(1+a)\,\Gamma(2\,b)}$$

Concluding remarks

The ***function_wizard*** is an ambitious long term project which could become a quite useful tool when finished. As mentioned in the introduction, the idea is not new, although its implementation in the framework of a computer algebra system gives to it another dimension. Also, the possibility of developing this project fully integrated with the system's funtionality, e.g., the conversion network, the series and simplification code for special functions etc. brings renewed momentum to the activity in the special function area of the Maple library.

A key issue in this project is that, as the mathematical language itself, the ***function_wizard*** is a project to be in constant development. Hence, all its structure is being developed with that idea in mind, so that the set of routines could be updated with ease and new sectors plugged into its skeleton without having to adjust the structure or other sectors.

The other crucial idea is that of separating "mathematical pieces of information" from the "computational client routines" implementing functionality. This approach also requires adjusting a relevant portion of the existing (old) Maple library, so that information formerly hard-coded in so called "client routines" - e.g., the **series** or **simplify** Maple commands - is detached and made available to other routines in an organized manner.

Another challenge of this project consists of making all the functionality sensitive to assumptions on the parameters of the special functions, for instance when ***function_wizard*** is called using the **assuming** Maple facility. Moreover, the idea is to also convey "what information on the parameters is understood by ***function_wizard***", as for instance illustrated with the output of the *integral form* for **BesselJ**.

One important topic not currently implemented in Maple and to be covered by ***function_wizard*** is related to the possibility of expanding any mathematical expression using any especified orthonormal set of functions as a base. This type of expansion is typically used when solving partial differential equations and will probably require developing some other Maple commands implementing the functionality at user level. Regarding the ***function_wizard*** side, to have access to the the orthonormal properties of any special function would permit to chose a suitable base, for instance, to expand exact PDE solutions according to the particular boundary conditions of a given problem, or for tackling many other problems as well; this functionality is relevant in Physics in general.

The help pages to be produced during the ***function_wizard*** project also require particular attention: it is not a matter of just replicating what is found in standard Handbooks of mathematical functions, but to develope these pages *taking into account the computer algebra system framing them.* So, we want to take advantage of the possible interactivines with the whole ***function_wizard*** functionality and so avoid common redundancies or cluttering of information with excess of particular cases etc. At the same time we want these help pages to provide open-GL visualization as well as access to *user parameterized* information taking advantage of the new Maplets Maple facility. A valuable design for such "alive help pages" is in fact viewed as a project in itself.

Finally, as with all first prototypes of long term computational project, there is a great deal of scope for changing and enhancing things. You are very welcome to contribute ideas for improvements.

Acknowledgments

This work was partly supported by the MITACS project at the Centre of Experimental and Constructive Mathematics, Simon Fraser University, Canada.