

JTC1/SC22/WG14 - C

Considerations about new complex mathematical functions in the C language

Vincent Reverdy (vince.rev@gmail.com)

Motivations

➔ Why?

- **Some complex mathematical functions are pending in the « future library directions » since 1999 (more than 14 years ago)!**
- **Real-valued mathematical functions are about to be updated according to ISO/IEC/IEEE 60559-2011: complex functions should be considered too.**
- **There is a real need for the complex gamma function (« the gamma function appears everywhere »: that is also the case for the complex one!)**

List of functions

➔ **Functions pending since C99 plus the complex counterpart of the new functions from ISO/IEC/IEEE 60559-2011:**

- **cexpm1**
- **cexp2**
- **cexp2m1**
- **cexp10**
- **cexp10m1**

- **clog1p**
- **clog2**
- **clog2p1**
- **clog10**
- **clog10p1**

- **cerf**
- **cerfc**
- **clgamma**
- **ctgamma**


- **crsqr**
- **ccomound**
- **crootn**
- **cpown**
- **cpowr**

- **csinpi**
- **ccospi**
- **catan2pi**

Possible approaches

 Three approaches are possible:

- **1 - No change:** the remaining complex mathematical functions are a « future library direction » (for how long?)
- **2 - All the functions of the previous page that have a clear mathematical definition/meaning will be included in the next standard:** this leads to have a completely consistent header `<complex.h>` with `<math.h>` and nothing is postponed to another version of the C standard.
- **3 - Only the functions that cannot be « easily » implemented by users using the current set of complex functions will be included in the next version of the standard and the others will be reserved for a future update. These functions are:** `cerf`, `cerfc`, `clgamma` and `ctgamma`.

 If the committee is in favor of the 2nd approach (which has the great advantage to definitely close the problem), I would be happy to work on the specifications. Otherwise, the reasons why the 3rd one should be considered is presented in the next slides.

Complex error and gamma functions

cerf

cerfc

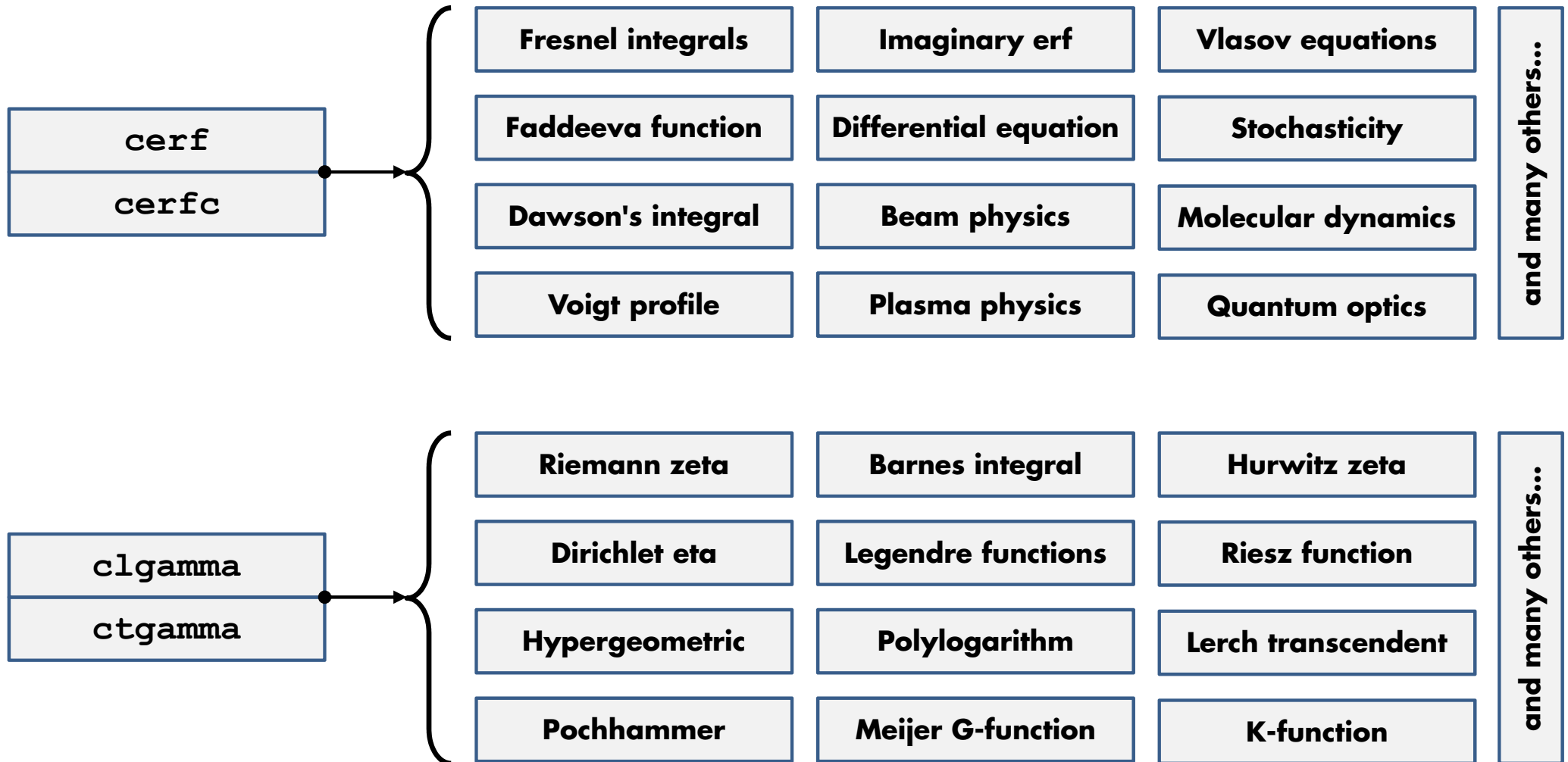
clgamma

ctgamma

➔ If the 2nd approach is not chosen, standardizing at least the complex error and gamma functions should be considered. Why ?

- They are difficult to implement from scratch for users.
- They cannot be easily implemented in terms of complex functions currently available in `<complex.h>`.
- They (especially the gamma functions) enter in a lot of definition of other mathematical functions.
- As a consequence of the preceding points, users are currently forced to switch to another language when they want to use complex special functions. Providing the error and gamma functions could solve this issue.
- Their names are already present in the standard as reserved names for future library directions.

Example of applications



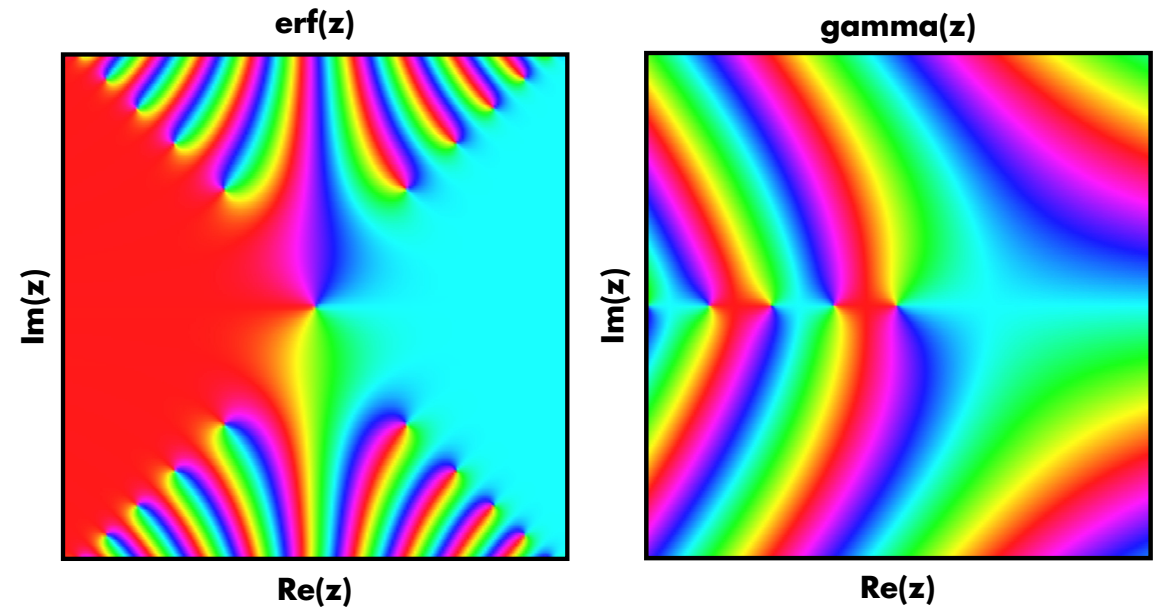
Some last words

➔ Open problems:

- **Exact definition of `c1gamma`?**
(branch cuts)
- **Specification of special values?**
(with infinity/NaN arguments)

➔ Existing implementations:

- **The Faddeeva package**
- **The `gamerf` package**



Straw polls

➔ **Should we consider the 2nd approach as a direction for the next C standard (standardizing all the meaningful complex counterpart of the `<math.h>` functions) ?**

Strongly in favor	Weakly in favor	Neutral	Weakly against	Strongly against

➔ **If not, should we consider at least the standardization of `cerf`, `cerfc`, `clgamma` and `ctgamma` as a direction for the next C standard ?**

Strongly in favor	Weakly in favor	Neutral	Weakly against	Strongly against