

Introduction to the `gfun` Package

Calling Sequence

`function(args)`

`gfun[function](args)`

Description

The `gfun` package has been designed as a help for the manipulation and discovery of functions or sequences satisfying linear differential or recurrence equations. The name of the package comes from its combinatorial application to generating functions.

The basic principle of the package is that linear differential equations or recurrences can be used as data-structures to represent their solutions. Procedures that convert to such a representation are:

[algfuntoalgeq](#) [algeqtodiffeq](#) [holexprtodiffeq](#)

The differential equations and recurrences can then be manipulated by

[algebraicsubs](#) [diffeqtorec](#) [rectodiffeq](#)
[poltodiffeq](#) [poltorec](#) [reducerecorder](#)

and the following ones that perform more elementary operations

[borel](#) [cauchyproduct](#) [diffeq+diffeq](#)
[diffeq*diffeq](#) [diffeqtohomdiffeq](#) [hadamardproduct](#)
[invborel](#) [Laplace](#)
[rec+rec](#) [rec*rec](#) [rectohomrec](#)
[minimizediffeq](#) (new) [istranscendental](#) (new) [indicialpolynomial](#) (new)

Useful information can then be extracted from these equations by one of

[algeqtoseries](#) [ratpolytocoef](#) [rectoproc](#) [the NumGfun subpackage](#)
[the ContFrac subpackage](#)

Given the first terms of the sequence, the `gfun` package also contains functions that will help conjecture what the generating function is. In some cases, this answer will be "explicit". In most cases though, such an explicit expression will not exist, and the answer will be an equation (either differential or algebraic) satisfied by the generating function. The functions dealing with numbers and series are:

[listtodiffeq](#) [seriestodiffeq](#)
[listtorec](#) [seriestorec](#)
[listtoalgeq](#) [seriestoalgeq](#)
[listtoratpoly](#) [seriestoratpoly](#)
[listtohypergeom](#) [seriestohypergeom](#)
[guesseqn](#) [guessgf](#)

All these guessing functions are based on a fast implementation that can be called directly, also for nonlinear problems: [pade2](#)

The way the guessing proceeds can be controlled by modifying [Parameters](#).

There are different types of generating functions that can be manipulated in `gfun`, such as ordinary (ogf) and exponential (egf).

The following conversion routines transform from one type to another:

[listtolist](#) [listtoseries](#)
[seriestolist](#) [seriestoseries](#)

Information about the computations that are being done can be obtained by setting `infolevel[gfun]` to anything between 1 and 5.

References

More information (on an early version) can be found in: "Gfun: a Maple package for the manipulation of generating and holonomic functions in one variable.", B. Salvy and P. Zimmermann, ACM Transactions on Mathematical Software, 20 (2), 163--177 (1994).

See also the web page of gfun: <http://perso.ens-lyon.fr/bruno.salvy/software/the-gfun-package/>

See Also

[with gfun\[Parameters\]](#)