NumGfun User Manual

Marc Mezzarobba

Manual for NumGfun v. 1.0
Last updated 2014-09-03

NumGfun User Manual

by Marc Mezzarobba

Contents

Overview of the gfun[NumGfun] Package

Calling Sequence

gfun[NumGfun][**command**](**arguments**)

command(**arguments**)

Description

The **NumGfun** package provides tools to perform "analytic" and numerical computations with power series given by *linear differential equations with polynomial coefficients*, analytic functions defined by convergent series of the same kind, and sequences given by recurrence relations with polynomial coefficients.

Its main features include the ability to compute:

- *numerical values* of analytic solutions of ODEs with polynomial coefficients, and *transition matrices* between ordinary or *regular singular points* of such equations, with guaranteed accuracy (that is, using rigorous error bounds),
- various kinds of *symbolic bounds* on the general terms/coefficients of solutions.

NumGfun is a subpackage of gfun.

List of NumGfun Package Commands

- Numerical evaluation and analytic continuation:
  `analytic_continuation` `diffeqtoproc` `evaldiffeq` `transition_matrix`
- Symbolic bounds:
  `bound_diffeq` `bound_diffeq_tail` `bound_ratpoly` `bound_rec` `bound_rec_tail`
- Utilities:
  `dominant_root` `fnth_term` `local_basis` `plot_path`

Informational Messages and Settings

- The verbosity level of NumGfun commands is determined by the value of infolevel[gfun]. Levels 1 to 5 correspond to informational messages. Levels 6 and higher additionally turn on debugging information.
- The **Settings** submodule provides a number of tuning parameters that influence the behaviour of NumGfun. As most of them require a detailed understanding of the algorithms and their implementation, the settings are only documented in the package's source code. Yet, error messages sometimes suggest changing a particular setting when a computation fails. This can be done by assigning a value to a member of **Settings**, as in: **NumGfun:-Settings:-default_eval_precision := 100**.

Examples

```
> with(gfun): with(NumGfun):
  fnth_term({(3*n+3)*u(n+1)=(3*n+5)*u(n), u(0)=1}, u(n), 2000, 50);
```

$$175.89036294166519099188900014849043485353660447070009 \tag{1.1}$$

```
> deq := holexprtodiffeq(arctan(z), y(z));
  evaldiffeq(deq, y(z), 1/2, 50);
```

$$deq := \left\{ \left(z^2+1\right)\left(\frac{\mathrm{d}}{\mathrm{d}z}\,y(z)\right) - 1,\ y(0)=0 \right\}$$

$$0.46364760900080611621425623146121440202853705428612 \tag{1.2}$$

```
> deq := (z^2+1)*diff(y(z),z,z) + (3*z+1)*diff(y(z),z) + z^2*y(z);
```

```
analytic_continuation(deq, y(z), [0, 1+I, 2], 50);
```

$$deq := (z^2+1)\left(\frac{d^2}{dz^2}y(z)\right) + (3z+1)\left(\frac{d}{dz}y(z)\right) + y(z)z^2$$

$$(0.72678326528197350565935299733280205125629707790244)\,\_C_0 + (0.43578845882065137070719121052451044561729148136090)\,\_C_1 \tag{1.3}$$

```
> deq := {(z^2-1)*diff(y(z),z,z)+(z^3+3)*y(z), y(0)=1, D(y)(0)=0};
  analytic_continuation(deq, y(z), [0, 1], 10, 'ord'=3);
```

$$deq := \left\{(z^2-1)\left(\frac{d^2}{dz^2}y(z)\right) + (z^3+3)\,y(z), y(0)=1, D(y)(0)=0\right\}$$

$$(4.7335543398)\left(1-2(z-1)\ln(z-1)+\left(-\frac{13}{4}+2\ln(z-1)\right)(z-1)^2\right) + (-8.3339545763+29.7417990787\,I)\left(z-1-(z-1)^2\right) \tag{1.4}$$

```
> deq := op(select(has, deq, z));
  transition_matrix(deq, y(z), [0, 1], 10);
```

$$deq := (z^2-1)\left(\frac{d^2}{dz^2}y(z)\right) + (z^3+3)\,y(z)$$

$$\begin{bmatrix} 4.7335543398 & 2.4577355851 \\ -8.3339545763+29.7417990787\,I & -4.1158620623+15.4424081173\,I \end{bmatrix} \tag{1.5}$$

```
> bound_ratpoly((z^7+3*z^2+z+1)/((z-2)^3*(z^3-3)*(z-I)^2), z);
```

$$\frac{1794743353}{25000000000\,(1-z)^2} + \frac{137935633}{5000000000}z^2 + \frac{2309624043}{50000000000}z^3 \tag{1.6}$$

```
> deq := holexprtodiffeq(arctan(z), y(z));
  bound_diffeq(deq, y(z));
```

$$deq := \left\{(z^2+1)\left(\frac{d}{dz}y(z)\right) - 1, y(0)=0\right\}$$

$$\frac{1}{2(1-z)^2} \tag{1.7}$$

Licence and Contact Information

- NumGfun is part of Algolib (http://algo.inria.fr/libraries/) available under the GNU Lesser General Public Licence, version 2.1 or, at your option, any later version. See the file COPYING for details.
- The source code for NumGfun can be downloaded from http://marc.mezzarobba.net#code-NumGfun.
- Please send your comments and bug reports to <marc@mezzarobba.net>.

References

The primary reference to use when citing NumGfun is:

- Marc Mezzarobba. NumGfun: a Package for Numerical and Analytic Computation with D-finite Functions. In *Proceedings of the 2010 International Symposium on Symbolic and Algebraic Computation (ISSAC 2010)*, pages 139–145. ACM, 2010. (`arXiv:1002.3077`, `doi:10.1145/1837934.1837965`).

A more detailed and up-to-date description as well as an in-depth discussion of many of the underlying algorithms appear in:

- Marc Mezzarobba. *Autour de l'évaluation numérique des fonctions D-finies*. Thèse de doctorat, École polytechnique, 2011. (In French.)

Original references for the algorithms implemented in NumGfun include the following:

- David V. Chudnovsky & Gregory V. Chudnovsky. Approximations and complex multiplication according to Ramanujan. *Ramanujan revisited*, Academic Press, 1988, 375-472.
- Marc Mezzarobba & Bruno Salvy. Effective Bounds for P-Recursive Sequences. *Journal of Symbolic Computation* 45(10):1075–1096, 2010. (`doi:10.1016/j.jsc.2010.06.024`)
- Joris van der Hoeven. Fast evaluation of holonomic functions. *Theoretical Computer Science*, 1999, 210, 199-216.

See Also

gfun, DEtools, UsingPackages, with

NumGfun[bound_diffeq] - majorant series for D-finite functions; NumGfun[bound_diffeq_tail] - bound the tails of the power series expansion of a D-finite function

Calling Sequences

bound_diffeq(**eq**, **y(z)**)

bound_diffeq_tail(**eq**, **y(z)**, **n**)

Parameters

`eq` - linear differential equation with polynomial coefficients, with initial values at origin

`y` - name; function name

`z` - name; variable of the function **y**

`n` - name; starting index of the tails

Description

- The **bound_diffeq** command computes a *majorant series* for the power series expansion at 0 of a formal power series specified as the solution of a linear differential equation with polynomial coefficients along with initial values. A majorant series of a formal series f with complex coefficients is a series g with nonnegative coefficients such that for all n, the coefficients f[n] and g[n] of z^n in f(z) and g(z) respectively satisfy abs(f[n]) <= g(n). The majorant series is a "tight" bound in the sense that its disk of convergence extends to the nearest singularity of the differential equation.
- The **bound_diffeq_tail** command computes a bound for the tails Sum(y[k]*z^k, k=n..infinity) of the power series expansion at 0 of an analytic function given as the solution of a linear differential equation with polynomial coefficients along with initial values. The output is formula involving the starting index n of the summation range.
- Differential equations with no initial values are also allowed on input. In this case, the output is only determined up to a constant factor, and is such that all formal power series (resp. convergent power series) solutions admit a majorant series (resp. a tail bound) of the given form for a suitable choice of constant. The constant depends on the particular choice of solution.
- Some intermediate computations are performed numerically, at a precision determined by the **Digits** environment variable. In particular, the value of **Digits** influences the precision at which some (rational) constants appearing in the bounds are computed. In rare cases, these functions may fail to produce a finite bound although one exists, and increasing **Digits** can help. (It is a bug, however, if an incorrect bound is returned.)

Examples

```
> with(gfun:-NumGfun):
> bound_diffeq({diff(y(z),z)=y(z), y(0)=1}, y(z));
```

$$\sum_{n=0}^{\infty} \frac{z^n}{\Gamma(n+1)} \tag{2.1}$$

```
> bound_diffeq({(1+z)*diff(y(z),z)=y(z), y(0)=1}, y(z));
```

$$\frac{1}{-z+1} \tag{2.2}$$

```
> bound_diffeq((1+z)*diff(y(z),z)=y(z), y(z));
```

```
Warning, incomplete initial conditions. The returned bound will hold (for a suitable
 choice of _C) for all *power series* solutions.
```

$$\frac{\_C}{-z+1} \tag{2.3}$$

```
> bound_diffeq({diff(y(z),z,z)=z*y(z), y(0)=1, D(y)(0)=1}, y(z));
```

$$\frac{1}{5040}\sum_{n=0}^{\infty}\frac{(n+1)\,(n+2)\,(n+3)\,(n+4)\,(n+5)\,(n+6)\,(n+7)\,3^{-\frac{2}{3}n}\,z^n}{\Gamma\left(\frac{1}{3}n+1\right)^2} \tag{2.4}$$

```
> bound_diffeq_tail({diff(y(z),z)=y(z), y(0)=1}, y(z), n);
```

$$\begin{cases} e^{|z|} & n \le \dfrac{|z|}{1-\dfrac{1}{|z|+3}} \\ \dfrac{(n+3)\left(\dfrac{|z|}{1-\dfrac{1}{n+3}}\right)^n}{\Gamma(n+1)} & otherwise \end{cases} \tag{2.5}$$

```
> bound_diffeq_tail({(1+z)*diff(y(z),z)=y(z), y(0)=1}, y(z), n);
```

$$-\frac{|z|^n}{|z|-1} \tag{2.6}$$

```
> bound_diffeq_tail(diff(y(z),z)=y(z), y(z), n);
```

```
Warning, incomplete initial conditions. The returned bound will hold (for a suitable
 choice of _C) for all *power series* solutions.
```

$$\begin{cases} \_C\,e^{|z|} & n \le \dfrac{|z|}{1-\dfrac{1}{|z|+3}} \\ \dfrac{\_C\,(n+3)\left(\dfrac{|z|}{1-\dfrac{1}{n+3}}\right)^n}{\Gamma(n+1)} & otherwise \end{cases} \tag{2.7}$$

See Also

gfun, NumGfun, bound_ratpoly, bound_rec

NumGfun[bound_ratpoly] - majorant series for rational functions

Calling Sequence

bound_ratpoly(**rat**, **z**)

Parameters

`rat` - rational function

`z` - name; variable of the rational function **rat**

Description

- The **bound_ratpoly** command computes a majorant series for the series expansion at 0 of a rational function.
- A majorant series of a formal Laurent series f with complex coefficients is a series g with nonnegative coefficients such that for all n, the the coefficients f[n] and g[n] of z^n respectively in f(z) and g(z) satisfy abs(f[n]) <= g[n].
- The majorant series returned by **bound_ratpoly** is of the form A/((z^j)*(1-u*z)^m)+P(z) where A and u are nonnegative constants, j and m are nonnegative integers, and P is a Laurent polynomial with nonnegative coefficients. It is a "tight" bound in the sense that the radius of convergence of its power series expansion matches that of the power series expansion of **rat**.

Examples

```
> with(gfun:-NumGfun):
> bound_ratpoly(1/(z+5), z);
```

$$\frac{1}{5\left(1-\frac{1}{5}z\right)} \tag{3.1}$$

```
> bound_ratpoly((5*z^6+z+2)/(z^3*(z^3+1)^2*(z+I)), z);
```

$$\frac{16147093}{5000000\,z^3\,(1-z)^2} \tag{3.2}$$

See Also

gfun, NumGfun, ratpolytocoeff, bound_diffeq

NumGfun[bound_rec] - bound a sequence given by a recurrence relation; NumGfun[bound_rec_tail] - bound the tails of a series whose general term satisfies a recurrence relation

Calling Sequences

bound_rec(**rec**, **u(n)**)

bound_rec_tail(**rec**, **u(n)**)

Parameters

`rec` - linear recurrence relation with polynomial coefficients, along with initial values

`u` - name; sequence name

`n` - name; variable of the sequence **u**

Description

- The **bound_rec** command computes a bound for the absolute value of the solution of a sequence given as the solution of a recurrence relation. The output is a formula involving the index **n** of the sequence.
- The **bound_rec_tail** command computes a bound for the series tails Sum(u(k), k=n..infinity), given a recurrence relation satisfied by u.
- Recurrences with no initial values are also allowed on input. In this case, the output is only determined up to a constant factor, and is such that solutions *defined for all nonnegative n* admit a bound of the given form for a suitable choice of constant. The constant depends on the particular choice of solution.
- Some intermediate computations are performed numerically, at a precision determined by the **Digits** environment variable. In particular, the value of **Digits** influences the precision at which some (rational) constants appearing in the bounds is computed. In rare cases, these functions may fail to produce a finite bound although one exists, and increasing **Digits** can help. (It is a bug, however, if an incorrect bound is returned.)

Examples

```
> with(gfun:-NumGfun):
```

```
> bound_rec({I*u(n+1) = (n+1)*u(n), u(0)=3}, u(n));
```

$$\frac{18849555927}{1000000000}\,(n+2)\,n! \tag{4.1}$$

```
> bound_rec({(2*n+2)^2*u(n+1) = -(n+1)*u(n), u(0)=1}, u(n));
```

$$\frac{(n+2)\,(n+1)\left(\frac{1}{4}\right)^{n}}{n!} \tag{4.2}$$

```
> # constant coefficients
```

```
bound_rec({15*u(n)-4*u(n+1)-13*u(n+2)+5*u(n+3), u(0) = 17/5, u(1) = 3, u(2) =
 3/12, u(3) = 0, u(4) = 5, u(5) = 0}, u(n));
```

$$\frac{3650256453}{1000000000}\,RootOf\left(15\,\_Z^3-4\,\_Z^2-13\,\_Z+5,\,index=1\right)^{-n}+\left\{\begin{array}{ll}\frac{9497435481}{10000000000} & n=7\\ \frac{1349743549}{1000000000} & n=4\\ \frac{3497435481}{10000000000} & n=6\\ 0 & otherwise\end{array}\right. \tag{4.3}$$

```
> bound_rec_tail({I*(n+1)*u(n+1) = u(n), u(0)=1}, u(n));
```

$$\left\{\begin{array}{ll}\mathrm{e} & n\le 2\\ \frac{(n+3)\left(\frac{1}{1-\frac{1}{n+3}}\right)^n}{\Gamma(n+1)} & otherwise\end{array}\right. \tag{4.4}$$

See Also

gfun, NumGfun, bound_diffeq

NumGfun[diffeqtoproc] - create a Maple procedure from a differential equation

Calling Sequence

diffeqtoproc(**eq**, **y(z)**, [prec=**precision**, disks=**disk_list**])

Parameters

`eq` - linear differential equation with polynomial coefficients

`y` - name; function name

`z` - name; variable of the function **y**

`precision` - (optional) positive integer; number of digits (of *absolute* precision)

`disks_list` - (optional) list of lists of the form [**path**, **radius**]

Description

- The **diffeqtoproc(eq, y(z))** command returns a Maple procedure **p** such that **p(u, [precision])** (where **u** may be a point or a path) evaluates **y** at **u** to the absolute precision precision. More precisely, **p(u, [precision])** is equivalent to **evaldiffeq(eq, y(z), u, [precision])**; see the help page of evaldiffeq for details.
- If the options **precision** and **disks** are given, **diffeqtoproc** performs precomputations that make subsequent evaluations to precisions up to **precision** at points within one of the disks given in **disk_list** faster. The elements of **disk_list** are lists of the form [**path**, **radius**] where **path** is a path (starting at the origin and avoiding the singular points of **eq**) whose endpoint gives the center of the disk. The precomputed data is used, if possible, for calls of the form **p(u, [precision])** where **u** is a single point (as opposed to a path), and only in this case. (Thus, the choice of the determination for multivalued functions **y** is done at precomputation time.)

Examples

```
> with(gfun): with(NumGfun):
```

```
> eq := diffeqtohomdiffeq(holexprtodiffeq(arctan(z),y(z)),y(z));
```

$$eq := \left\{-2\,z\left(\frac{\mathrm{d}}{\mathrm{d}z}\,y(z)\right)+\left(-z^2-1\right)\left(\frac{\mathrm{d}^2}{\mathrm{d}z^2}\,y(z)\right),\ y(0)=0,\ \mathrm{D}(y)(0)=1\right\} \tag{5.1}$$

```
> p := diffeqtoproc(eq, y(z)):
  [ p([0, 1+I, 2*I], 30), p([0, -1+I, 2*I], 30) ];
```

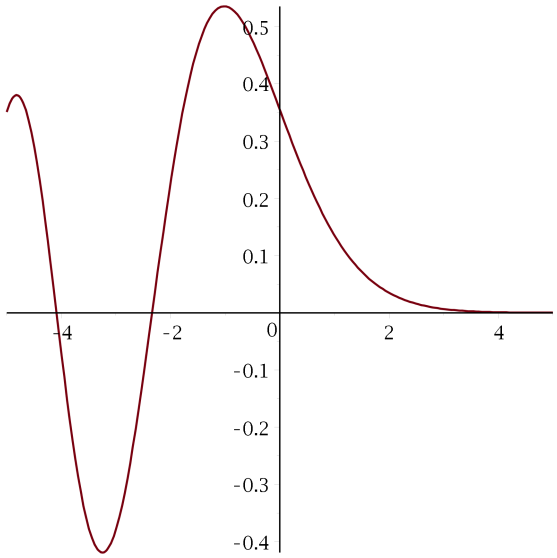
$$\begin{aligned}&[1.5707963267948966192313216916\\&\quad+0.54930614433405484569762261846\,\mathrm{I},\\&\quad-1.5707963267948966192313216916\\&\quad+0.54930614433405484569762261846\,\mathrm{I}]\end{aligned} \tag{5.2}$$

```
> eq := holexprtodiffeq(AiryAi(z), y(z));
```

$$eq := \left\{\frac{\mathrm{d}^2}{\mathrm{d}z^2}\,y(z)-z\,y(z),\ y(0)=\frac{1}{3}\,\frac{3^{1/3}}{\Gamma\left(\frac{2}{3}\right)},\ \mathrm{D}(y)(0)=-\frac{1}{2}\,\frac{3^{1/6}\,\Gamma\left(\frac{2}{3}\right)}{\pi}\right\} \tag{5.3}$$

```
> p := diffeqtoproc(eq, y(z), 'prec'=12, 'disks'=[[-2.5,3],[2.5,3]]):
```

```
> plot(p, -5..5);
```




```
> infolevel[gfun] := 2:
> p(-3, 10);
```

$$-0.3788142937 \tag{5.4}$$

```
> p(-3, 30);
p: using multiple-precision analytic continuation
rewrite_path: final analytic continuation path is [0 0 -3]
ordinary_step_transition_matrix: 0 --> -3 ord=1, prec~=.54e-31, terms=88
Recall that gfun:-NumGfun uses *absolute* error.
```

$$-0.378814293677658074347243916500 \tag{5.5}$$

```
> p(-42);
p: using multiple-precision analytic continuation
rewrite_path: final analytic continuation path is [0 0 -42]
ordinary_step_transition_matrix: 0 --> -42 ord=1, prec~=.54e-11, terms=840
Recall that gfun:-NumGfun uses *absolute* error.
```

$$0.0073966067 \tag{5.6}$$

See Also

gfun, NumGfun, rectoproc, evaldiffeq

NumGfun[dominant_root] - dominant root of a polynomial

Calling Sequence

dominant_root(**pol**, **z**, [`rootof`])

Parameters

`pol` - polynomial with algebraic coefficients and nonzero constant term

`z` - name; variable

`rootof` - optional keyword

Description

- The **dominant_root** command computes one of the roots of **pol** of maximum multiplicity among those of minimum absolute value.
- The constant term of **pol** must be nonzero.
- The output is a list of the form [root, multiplicity].
- With option `rootof`, if the dominant root is not rational, it is returned as a RootOf.

Examples

```
> with(gfun:-NumGfun):
> dominant_root((z-1)*(z^2+1)^2, z);
```

$$[I, 2] \tag{6.1}$$

```
> dominant_root((z-1)*(z^2+1)^2, z, 'rootof');
```

$$[RootOf(\_Z^2 + 1, index = 1), 2] \tag{6.2}$$

See Also

gfun, NumGfun, bound_ratpoly

NumGfun[evaldiffeq], NumGfun[analytic_continuation] - numerical evaluation of D-finite functions; NumGfun[transition_matrix] - numerical connection between regular points

Calling Sequences

evaldiffeq(**eq**, **y(z)**, **point**, [**precision**])

evaldiffeq(**eq**, **y(z)**, **path**, [**precision**], [ord=**order**], [monomials])

analytic_continuation(**...**)

transition_matrix(**eq**, **y(z)**, **path**, [**precision**])

Parameters

`eq -` linear differential equation with polynomial coefficients

`y -` name; function name

`z -` name; variable of the function **y**

`point -` complex number; evaluation point

`path -` list of complex numbers; analytic continuation path

`precision -` (optional) positive integer; required absolute accuracy (in decimal digits)

`ord -` (optional) positive integer; number of terms of local expansion to return

Description

Basic Usage

- The **evaldiffeq(eq, y(z), point, precision)** command evaluates the solution of the differential equation **eq** at **point** with an *absolute error* bounded by 10^(-**precision**). The evaluation point must lie closer to zero than any of the singular points of the differential equation. (The origin itself must be an ordinary point.)
- The more general calling sequence **evaldiffeq(eq, y(z), path, precision)** evaluates the function defined by analytic continuation of the solution along a broken-line path starting at 0 and avoiding the singular points of the equation, given as a list of vertices. In particular, **evaldiffeq(eq, y(z), [point], precision)** is equivalent to **evaldiffeq(eq, y(z), point, precision)** if **point** lies within the disk where the latter is defined, and gives the value of the solution defined on the Mittag-Leffler star with center at 0 of the differential equation **eq** otherwise.
- **analytic_continuation** is synonymous with **evaldiffeq**.
- Instead of the value of a single solution, **transition_matrix(eq, y(z), path, [precision])** computes the fundamental matrix at the endpoint of the path, corresponding to initial conditions Y(z0)=Id at the starting point. The i-th row of Y(z) contains the value of the coefficient of z^i in the Taylor expansion of each fundamental solution. Initial values given in the differential equation are ignored. The path does not need to start at 0. The meaning of the other parameters is unchanged.
- In all these cases, **point** and the elements of **path** are expected to be of type complex(numeric).

Regular Singular Points

- The endpoints of an analytic continuation path may also be regular singular points of the differential equation. If z0 and z1 are regular (i.e., ordinary or regular singular) points, the entries of the transition matrix from z0

to z1 along a certain path are the coefficients of the expression in a certain “local basis at z1” of the elements of the “local basis at z0” extended by analytic continuation along that path. In other words, the transition matrix sends the coefficients of the decomposition of a given solution on the first local basis to the coefficients of the expression of the same solution on the second local basis. Local bases are defined in such a way that this generalizes the above definition of transition matrices between ordinary point.

- Local bases can be computed using the local_basis command.
- When the keyword parameter `ord` is set to a positive integer **order**, **evaldiffeq** returns the first **order** terms of the generalized series expansion of the solution of interest at the endpoint of the path, in a format illustrated below. Alternatively, the keyword `monomials` can be specified to annotate each numerical connection constant with the leading term of the generalized series expansion of the corresponding local solution. If none of these options is given, only the connection constant corresponding to the first element of the local basis (corresponding in some sense to the “asymptotically dominant” local solution) is returned.
- There is no syntax for specifying “initial values” at regular singular points as part of the equation **eq**. When the origin is a regular singular point, **evaldiffeq** returns an expression involving symbolic constants _C[0], _C[1], ... representing the coefficients of the decomposition of y(z) in the canonical basis at the origin.
- Arbitrary algebraic numbers are allowed as part of analytic continuation paths when they correspond to regular singular points of the differential equation. They must be specified as indexed RootOfs.

Accuracy

- All these functions are designed to ensure the accuracy of the output: it is a bug if the result is not within 10^(-**precision**) of the exact value of the function.
- As an exception to the previous rule, *regular singular* connection problems for differential equations with *a single* finite singular point currently rely on heuristic error estimates. (A warning is issued when such an estimate is used.)
- When the result is not a single floating-point number, the floating-point coefficients appearing in it are “quoted” using the empty symbol `` to prevent automatic floating-point simplification of the output. This happens in particular when the initial values contain symbolic parameters.
- The Digits environment variable has no influence on the accuracy of the results. However, some internal computations, including that of intermediate error bounds, are performed at precision Digits (typically, using interval arithmetic to ensure that the final result remains rigorous). In rare cases (e.g., equations with singularities at distance about 10^(-Digits) from each other), it may be necessary to increase Digits for the computation to succeed.
- Likewise, evaluation points given in floating-point format are interpreted as exact rational numbers (or complex numbers with rational real and imaginary parts) regardless of the setting of Digits.

Performance

- These commands implement asymptotically fast algorithms, allowing in principle for evaluations at very high precisions (up to millions of digits). However, the constant factors involved are comparatively large and the cost grow fast with the complexity (order, degree, growth of a generic solution) of the equation.
- They are less suitable to be called repeatedly, even at moderate precisions. In particular, potentially costly bound computations are performed before each evaluation. For repeated evaluations at moderate precision in a known domain, try using diffeqtoproc instead.

Examples

```
> restart; with(gfun:-NumGfun):
```

```
> evaldiffeq({diff(y(z),z)-y(z), y(0)=1}, y(z), 1, 50);
```

$$2.71828182845904523536028747135266249775724709369996 \tag{7.1}$$

```
> analytic_continuation({(1+z^2)*diff(y(z),z,z)-(2*z+3)*y(z)+I*y(z), y(0)=Pi, D(y)
  (0)=-I}, y(z), [0, 2]);
```

$$37.6769250446 - 22.6773530382\,I \tag{7.2}$$

Computation of a local monodromy matrix by analytic continuation along a closed path around a singular point:

```
> eq := gfun:-diffeqtohomdiffeq(gfun:-holexprtodiffeq(arctan(z),y(z)),y(z));
```

$$eq := \left\{-2\,z\left(\frac{\mathrm{d}}{\mathrm{d}z}\,y(z)\right)+\left(-z^2-1\right)\left(\frac{\mathrm{d}^2}{\mathrm{d}z^2}\,y(z)\right),\ y(0)=0,\ \mathrm{D}(y)(0)=1\right\} \tag{7.3}$$

```
> transition_matrix(eq, y(z), [2*I, -1+I, 0, 1+I, 2*I], 20);
Warning, initial conditions {y(0) = 0, (D(y))(0) = 1} will be ignored
```

$$\begin{bmatrix} 1.0000000000000000000 & -9.4247779607693797153 \\ 0. & 1.0000000000000000000 \end{bmatrix} \tag{7.4}$$

Numerical connection between an ordinary point and a regular singular point:

```
> evaldiffeq(eq, y(z), [0, I], 30, 'ord'=3);
```

$$\begin{aligned}&(-0.50000000000000000000000000000\,\mathrm{I})\left(\ln(z-\mathrm{I})+\frac{1}{2}\,\mathrm{I}\,(z-\mathrm{I})-\frac{1}{8}\,(z-\mathrm{I})^2\right)\\&\quad+(0.78539816339744830961566084582\\&\quad+0.34657359027997265470861606072\,9\,\mathrm{I})\end{aligned} \tag{7.5}$$

```
> evaldiffeq(eq, y(z), [0, I], 30, 'monomials');
```

$$\begin{aligned}&(-0.50000000000000000000000000000\,\mathrm{I})\,(\ln(z-\mathrm{I})+\ldots)\\&\quad+(0.78539816339744830961566084582\\&\quad+0.34657359027997265470861606072\,9\,\mathrm{I})\,(1+\ldots)\end{aligned} \tag{7.6}$$

```
> evaldiffeq(eq, y(z), [0, I], 30);
```

$$-0.50000000000000000000000000000\,\mathrm{I} \tag{7.7}$$

```
> eq := op(select(has, eq, z));
```

$$eq := -2\,z\left(\frac{\mathrm{d}}{\mathrm{d}z}\,y(z)\right)+\left(-z^2-1\right)\left(\frac{\mathrm{d}^2}{\mathrm{d}z^2}\,y(z)\right) \tag{7.8}$$

```
> evaldiffeq(eq, y(z), [0, I], 10, 'ord'=3);
```

$$\begin{aligned}&\left((0.)\,\_C_0+(-0.5000000000\,\mathrm{I})\,\_C_1\right)\left(\ln(z-\mathrm{I})+\frac{1}{2}\,\mathrm{I}\,(z-\mathrm{I})-\frac{1}{8}\,(z-\mathrm{I})^2\right)\\&\quad+\left((1.0000000000)\,\_C_0+(0.7853981634+0.3465735903\,\mathrm{I})\,\_C_1\right)\end{aligned} \tag{7.9}$$

The same result expressed as a transition matrix:

```
> local_basis(eq, y(z), 0), local_basis(eq, y(z), I);
```

$$\begin{aligned}&\left[1,\ z-\frac{1}{3}\,z^3+\frac{1}{5}\,z^5\right],\left[\ln(z-\mathrm{I})+\frac{1}{2}\,\mathrm{I}\,(z-\mathrm{I})-\frac{1}{8}\,(z-\mathrm{I})^2-\frac{1}{24}\,\mathrm{I}\,(z-\mathrm{I})^3+\frac{1}{64}\,(z\right.\\&\quad\left.-\mathrm{I})^4+\frac{1}{160}\,\mathrm{I}\,(z-\mathrm{I})^5,\ 1\right]\end{aligned} \tag{7.10}$$

```
> transition_matrix(eq, y(z), [0, I], 10);
```

$$\begin{bmatrix} 0. & -0.5000000000\,\mathrm{I} \\ 1.0000000000 & 0.7853981634+0.3465735903\,\mathrm{I} \end{bmatrix} \tag{7.11}$$

Algebraic numbers

```
> eq := (z^2-2)*diff(y(z),z,z) + z*y(z) + y(z);
```

$$eq := \left(z^2-2\right)\left(\frac{\mathrm{d}^2}{\mathrm{d}z^2}\,y(z)\right)+y(z)\,z+y(z) \tag{7.12}$$

```
> alias(alpha=convert(sqrt(2), 'RootOf'));
```

$$\alpha \tag{7.13}$$

```
> local_basis(eq, y(z), alpha, 3);
```

$$\left[1-\frac{1}{4}\left(2+\sqrt{2}\right)\left(z-\sqrt{2}\right)\ln\left(z-\sqrt{2}\right)+\frac{1}{32}\left(3+2\sqrt{2}\right)\left(11-10\sqrt{2}+2\ln\left(z-\sqrt{2}\right)\right)\left(z-\sqrt{2}\right)^2, z-\sqrt{2}+\left(-\frac{1}{8}\sqrt{2}-\frac{1}{4}\right)\left(z-\sqrt{2}\right)^2\right] \tag{7.14}$$

```
> # [0,1,alpha] rather than [0,alpha] to work around a weakness
  # of evalrC
  transition_matrix(eq, y(z), [0, 1, alpha]);
```

$$\begin{bmatrix} 2.4938814696 & 2.4089417847 \\ -0.2035417759+6.6873857098\,\mathrm{I} & 0.2043720671+6.4596184954\,\mathrm{I} \end{bmatrix} \tag{7.15}$$

A case where only heuristic error control is currently implemented:

```
> eq := holexprtodiffeq(Ei(z), y(z));
```

$$eq := (-z+1)\left(\frac{\mathrm{d}}{\mathrm{d}z}y(z)\right)+z\left(\frac{\mathrm{d}^2}{\mathrm{d}z^2}y(z)\right) \tag{7.16}$$

```
> evaldiffeq(eq, y(z), [0, 1], 20);
Warning, infinite radius of convergence at regular singular point 0: rigorous error
 bounds are not implemented, falling back on heuristics
```

$$(1.31790215145440389486)\,\_C_0+(1.00000000000000000000)\,\_C_1 \tag{7.17}$$

See Also

gfun, NumGfun, diffeqtoproc, local_basis, nth_term

NumGfun[local_basis] - “canonical” local basis of the solution space of a linear ODE

Calling Sequence

local_basis(**eq**, **y(z)**, **point**, [**order**])

Parameters

`eq` - linear differential equation with polynomial coefficients

`y` - name; function name

`z` - name; variable of the function **y**

`point` - complex number; point at which to compute the local basis

`order` - integer, truncation order

Description

- The **local_basis** command returns the first few terms of the local basis of generalized series solutions at a given point of a differential equation used by other NumGfun commands.
- The expansion point must be a regular (i.e., ordinary or regular singular) point, and can be specified as an expression of type complex(numeric) or as an indexed RootOf.

Examples

```
> with(gfun:-NumGfun):
```

Ordinary points:

```
> eq := diff(y(z),z,z,z)+2*(diff(y(z), z))+y(z);
```

$$eq := \frac{d^3}{dz^3} y(z) + 2\left(\frac{d}{dz} y(z)\right) + y(z) \tag{8.1}$$

```
> local_basis(eq, y(z), 0);
```

$$\left[1 - \frac{1}{6} z^3 + \frac{1}{60} z^5, z - \frac{1}{3} z^3 - \frac{1}{24} z^4 + \frac{1}{30} z^5, z^2 - \frac{1}{6} z^4 - \frac{1}{60} z^5\right] \tag{8.2}$$

```
> local_basis(eq, y(z), I);
```

$$\left[1 - \frac{1}{6}(z-I)^3 + \frac{1}{60}(z-I)^5, z - I - \frac{1}{3}(z-I)^3 - \frac{1}{24}(z-I)^4 + \frac{1}{30}(z-I)^5, (z-I)^2 - \frac{1}{6}(z-I)^4 - \frac{1}{60}(z-I)^5\right] \tag{8.3}$$

Regular singular point:

```
> eq := z*diff(y(z),z,z,z)+2*y(z);
```

$$eq := z\left(\frac{d^3}{dz^3} y(z)\right) + 2\, y(z) \tag{8.4}$$

```
> local_basis(eq, y(z), 0);
```

$$\left[1 - z^2 \ln(z) + \left(-\frac{13}{144} + \frac{1}{12}\ln(z)\right) z^4, z - \frac{1}{3} z^3 + \frac{1}{90} z^5, z^2 - \frac{1}{12} z^4\right] \tag{8.5}$$

Algebraic exponents and expansion points:

```
> eq := z^4*(diff(y(z), z$4))+2*z^3*(diff(y(z), z$3))-3*z^2*(diff(y(z), z,
   z))+3*z*(diff(y(z), z))+(z+1)*y(z);
```

$$eq := z^4\left(\frac{d^4}{dz^4}y(z)\right)+2z^3\left(\frac{d^3}{dz^3}y(z)\right)-3z^2\left(\frac{d^2}{dz^2}y(z)\right)+3z\left(\frac{d}{dz}y(z)\right)+(z+1)y(z) \tag{8.6}$$

```
> local_basis(eq, y(z), 0, 2);
```

$$\left[z^{1-\sqrt{2}}\ln(z)-\frac{1}{343}\left(-12+4\sqrt{2}+7\ln(z)\right)z^{2-\sqrt{2}}\left(9+4\sqrt{2}\right), z^{1-\sqrt{2}}-\frac{1}{49}z^{2-\sqrt{2}}\left(9+4\sqrt{2}\right), z^{1+\sqrt{2}}\ln(z)+\frac{1}{343}z^{2+\sqrt{2}}\left(-12-4\sqrt{2}+7\ln(z)\right)\left(-9+4\sqrt{2}\right), z^{1+\sqrt{2}}+\frac{1}{49}z^{2+\sqrt{2}}\left(-9+4\sqrt{2}\right)\right] \tag{8.7}$$

```
> local_basis(eq, y(z), RootOf(_Z^3-1, 'index'=2));
```

$$\left[1+\left(\frac{1}{48}I\sqrt{3}-\frac{1}{48}\right)\left(z-(-1)^{2/3}\right)^4+\left(-\frac{11}{240}+\frac{1}{240}I\sqrt{3}\right)\left(z-(-1)^{2/3}\right)^5, z-(-1)^{2/3}-\frac{1}{8}\left(z-(-1)^{2/3}\right)^4+\left(-\frac{1}{15}-\frac{7}{120}I\sqrt{3}\right)\left(z-(-1)^{2/3}\right)^5, \left(z-(-1)^{2/3}\right)^2+\left(-\frac{1}{8}+\frac{1}{8}I\sqrt{3}\right)\left(z-(-1)^{2/3}\right)^4-\frac{1}{4}\left(z-(-1)^{2/3}\right)^5, \left(z-(-1)^{2/3}\right)^3+\left(\frac{1}{4}I\sqrt{3}+\frac{1}{4}\right)\left(z-(-1)^{2/3}\right)^4+\left(\frac{9}{40}I\sqrt{3}-\frac{9}{40}\right)\left(z-(-1)^{2/3}\right)^5\right] \tag{8.8}$$

See Also

gfun, NumGfun, evaldiffeq, DEtools[formal_sol]

NumGfun[plot_path] - display an analytic continuation path

Calling Sequences

plot_path(**eq**, **y(z)**, **path**)

plot_path(**eq**, **y(z)**, **path**, 'rewrite')

Parameters

`eq` - linear differential equation with polynomial coefficients

`y` - name; function name

`z` - name; variable of the function **y**

`path` - list of complex numbers; analytic continuation path

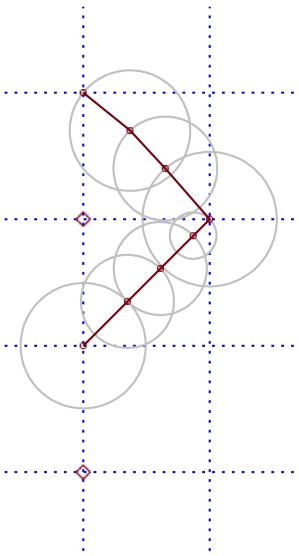
Description

Plot the analytic continuation path **path** along with the singular points of **eq**.

With option **rewrite**, first subdivide **path** to plot an approximation of the analytic continuation path that evaldiffeq would really use (by default) when asked to perform analytic continuation along that path.

Examples

```
> with(gfun:-NumGfun):
> gfun:-NumGfun:-plot_path((1+z^2)*diff(y(z),z,z)+y(z), y(z), [0, 1+I, 2*I],
   'rewrite');
```



See Also

NumGfun, evaldiffeq