

## 16.19 DESIR: Differential linear homogeneous equation solutions in the neighborhood of irregular and regular singular points

This package enables the basis of formal solutions to be computed for an ordinary homogeneous differential equation with polynomial coefficients over  $\mathbb{Q}$  of any order, in the neighborhood of zero (regular or irregular singular point, or ordinary point).

Authors: C. Dicrescenzo, F. Richard-Jung, E. Tournier.

Differential linear homogenous Equation Solutions in the neighbourhood of Irregular and Regular singular points

Version 3.1 - Septembre 89

Groupe de Calcul Formel de Grenoble  
laboratoire TIM3

(C. Dicrescenzo, F. Richard-Jung, E. Tournier)

E-mail: dicresc@afp.imag.fr

### 16.19.1 INTRODUCTION

This software enables the basis of formal solutions to be computed for an ordinary homogeneous differential equation with polynomial coefficients over  $\mathbb{Q}$  of any order, in the neighbourhood of zero ( regular or irregular singular point, or ordinary point ).

Tools have been added to deal with equations with a polynomial right-hand side, parameters and a singular point not to be found at zero.

This software can be used in two ways :

- direct ( DELIRE procedure )
- interactive ( DESIR procedure)

The basic procedure is the DELIRE procedure which enables the solutions of a linear homogeneous differential equation to be computed in the neighbourhood

of zero.

The DESIR procedure is a procedure without argument whereby DELIRE can be called without preliminary treatment to the data, that is to say, in an interactive autonomous way. This procedure also proposes some transformations on the initial equation. This allows one to start comfortably with an equation which has a non zero singular point, a polynomial right-hand side and parameters.

This document is a succinct user manual. For more details on the underlying mathematics and the algorithms used, the reader can refer to :

**E. Tournier** : Solutions formelles d'équations différentielles - Le logiciel de calcul formel DESIR.

These d'Etat de l'Université Joseph Fourier (Grenoble - avril 87).

He will find more precision on use of parameters in :

**F. Richard-Jung** : Representation graphique de solutions d'équations différentielles dans le champ complexe.

These de l'Université Louis Pasteur (Strasbourg - septembre 88).

### 16.19.2 FORMS OF SOLUTIONS

We have tried to represent solutions in the simplest form possible. For that, we have had to choose different forms according to the complexity of the equation (parameters) and the later use we shall have of these solutions.

"**general solution**" = { ....., { split\_sol , cond }, .....

cond = list of conditions or empty list (if there is no condition) that parameters have to verify such that split\_sol is in the basis of solutions. In fact, if there are parameters, basis of solutions can have different expressions according to the values of parameters. ( Note : if cond={}, the list "general solution" has one element only.)

split\_sol = { *q, ram, polysol, r* }  
 (" split solution " enables precise information on the solution to be obtained immediately )

The variable in the differential operator being  $x$ , solutions are expressed in respect to a new variable  $x_t$ , which is a fractional power of  $x$ , in the following way :

$q$  : polynomial in  $1/xt$  with complex coefficients  
 $ram$  :  $xt = x^{ram}$  ( $1/ram$  is an integer)  
 $polysol$  : polynomial in  $\log(xt)$  with formal series in  $xt$  coefficients  
 $r$  : root of a complex coefficient polynomial ("indicial equation").

"standard solution" =  $e^{qx} x^{r*ram} polysolx$

$qx$  and  $polysolx$  are  $q$  and  $polysol$  expressions in which  $xt$  has been replaced by  $x^{ram}$

N.B. : the form of these solutions is simplified according to the nature of the point zero.

- if 0 is a regular singular point : the series appearing in  $polysol$  are convergent,  $ram = 1$  and  $q = 0$ .
- if 0 is a regular point, we also have :  $polysol$  is constant in  $\log(xt)$  (no logarithmic terms).

### 16.19.3 INTERACTIVE USE

To call the procedure : `desir();`  
`solution:=desir();`

The DESIR procedure computes formal solutions of a linear homogeneous differential equation in an interactive way.

In this equation the variable *must be*  $x$ .

The procedure requires the order and the coefficients of the equation, the names of parameters if there are any, then if the user wants to transform this equation and how ( for example to bring back a singular point to zero see procedures `changehom`, `changevar`, `changeonc` - ).

This procedure DISPLAYS the solutions and RETURNS a list of general term { `lcoeff`, { ..., { `general_solution` }, ... } }. The number of elements in this list is linked to the number of transformations requested :

- \* `lcoeff` : list of coefficients of the differential equation
- \* `general_solution` : solution written in the general form

### 16.19.4 DIRECT USE

**procedure delire**( $x, k, grille, lcoef, param$ );

This procedure computes formal solutions of a linear homogeneous differential equation with polynomial coefficients over  $\mathbb{Q}$  and of any order, in the neighborhood of zero, regular or irregular singular point. In fact it initializes the call of the NEWTON procedure that is a recursive procedure (algorithm of NEWTON-RAMIS-MALGRANGE)

$x$  : variable  
 $k$  : "number of desired terms".  
 For each formal series in  $xt$  appearing in  $polysol$ ,  
 $a_0 + a_1xt + a_2xt^2 + \dots + a_nxt^n + \dots$ , we compute the  $k + 1$  first  
 coefficients  $a_0, a_1, \dots, a_k$ .  
 $grille$  : the coefficients of the differential operator are polynomial in  $x^{grille}$  (in  
 general  $grille = 1$ )  
 $lcoef$  : list of coefficients of the differential operator (in increasing order of  
 differentiation)  
 $param$  : list of parameters

This procedure RETURNS the list of general solutions.

### 16.19.5 USEFUL FUNCTIONS

#### Reading of equation coefficients

**procedure lectabcoef**( );

This procedure is called by DESIR to read the coefficients of an equation, in *increasing order of differentiation*, but can be used independently.

reading of  $n$  : order of the equation.

reading of parameters (only if a variable other than  $x$  appears in the coefficients)

this procedure returns the list  $\{ lcoef, param \}$  made up of the list of coefficients and the list of parameters (which can be empty).

#### Verification of results

**procedure solvalide**( $solutions, solk, k$ );

This procedure enables the validity of the solution number *sol*k in the list "solutions" to be verified.

*solutions* = {*lcoeff*, {..., {*general\_solution*}, ...}} is any element of the list returned by DESIR or is {*lcoeff*, *sol*} where *sol* is the list returned by DELIRE.

If we carry over the solution  $e^{qx} x^{r*ram} poly sol x$  in the equation, the result has the form  $e^{qx} x^{r*ram} reste$ , where *reste* is a polynomial in  $\log(xt)$ , with polynomial coefficients in  $xt$ . This procedure computes the minimal valuation *V* of *reste* as polynomial in  $xt$ , using *k* "number of desired terms" asked for at the call of DESIR or DELIRE, and DISPLAYS the "theoretical" size order of the regular part of the result :  $x^{ram*(r+v)}$ .

On the other hand, this procedure carries over the solution in the equation and DISPLAYS the significative term of the result. This is of the form :

$$e^{qx} x^a polynomial(\log(xt)), \quad \text{with } a \geq ram * (r + v).$$

Finally this procedure RETURNS the complete result of the carry over of the solution in the equation.

This procedure cannot be used if the solution number *sol*k is linked to a condition.

### Writing of different forms of results

#### **procedure standsol(solutions);**

This procedure enables the simplified form of each solution to be obtained from the list "solutions", {*lcoeff*, {..., {*general\_solution*}, ...}} which is one of the elements of the list returned by DESIR, or {*lcoeff*, *sol*} where *sol* is the list returned by DELIRE.

This procedure RETURNS a list of 3 elements : { *lcoeff*, *solstand*, *solcond* }

*lcoeff* = list of differential equation coefficients  
*solstand* = list of solutions written in standard form  
*solcond* = list of conditional solutions that have not been written in standard form. This solutions remain in general form.

This procedure has no meaning for "conditional" solutions. In case, a value has to be given to the parameters, that can be done either by calling the procedure SORPARAM that displays and returns these solutions in the standard form, either by calling the procedure SOLPARAM which returns these solutions in general form.

#### **procedure sorsol(sol);**

This procedure is called by DESIR to write the solution *sol*, given in general form, in standard form with enumeration of different conditions (if there are any). It can be used independently.

### Writing of solutions after the choice of parameters

**procedure sorparam**(*solutions, param*);

This is an interactive procedure which displays the solutions evaluated : the value of parameters is requested.

*solutions* : {*lcoeff*, {..., {*general\_solution*}, ...}}  
*param* : list of parameters.

It returns the list formed of 2 elements :

- list of evaluated coefficients of the equation
- list of standard solutions evaluated for the value of parameters.

**procedure solparam**(*solutions, param, valparam*);

This procedure evaluates the general solutions for the value of parameters given by *valparam* and returns these solutions in general form.

*solutions* : {*lcoeff*, {..., {*general\_solution*}, ...}}  
*param* : list of parameters  
*valparam* : list of parameters values

It returns the list formed of 2 elements :

- list of evaluated coefficients of the equation
- list of solutions in general form, evaluated for the value of parameters.

### Transformations

**procedure changehom**(*lcoeff, x, secmember, id*);

Differentiation of an equation with right-hand side.

*lcoeff* : list of coefficients of the equation  
*x* : variable  
*secmember* : right-hand side  
*id* : order of the differentiation.

It returns the list of coefficients of the differentiated equation. It enables an equation with polynomial right-hand side to be transformed into a homogeneous equation by differentiating  $id$  times,  $id = \text{degre}(\text{secmember}) + 1$ .

**procedure changevar**(*lcoeff*, *x*, *v*, *fct*);

Changing of variable in the homogeneous equation defined by the list, *lcoeff* of its coefficients : the old variable  $x$  and the new variable  $v$  are linked by the relation  $x = fct(v)$ .

It returns the list of coefficients in respect to the variable  $v$  of the new equation.

examples of use :

- translation enabling a rational singularity to be brought back to zero.
- $x = 1/v$  brings the infinity to 0.

**procedure changefonc**(*lcoeff*, *x*, *q*, *fct*);

Changing of unknown function in the homogeneous equation defined by the list *lcoeff* of its coefficients :

*lcoeff* : list of coefficients of the initial equation  
*x* : variable  
*q* : new unknown function  
*fct* :  $y$  being the unknown function  $y = fct(q)$

It returns the list of coefficients of the new equation.

Example of use :

this procedure enables the computation, in the neighbourhood of an irregular singularity, of the "reduced" equation associated to one of the slopes (the Newton polygon having a null slope of no null length). This equation gives much informations on the associated divergent series.

### Optional writing of intermediary results

**switch trdesir** : when it is ON, at each step of the Newton algorithm, a description of the Newton polygon is displayed (it is possible to follow the break of slopes), and at each call of the FROBENIUS procedure ( case of a null slope ) the corresponding indicial equation is displayed.

By default, this switch is OFF.

**16.19.6 LIMITATIONS**

1. This DESIR version is limited to differential equations leading to indicial equations of degree  $\leq 3$ . To pass beyond this limit, a further version written in the D5 environment of the computation with algebraic numbers has to be used.
2. The computation of a basis of solutions for an equation depending on parameters is assured only when the indicial equations are of degree  $\leq 2$ .