Dsolv Manual

Edition : auto generated by oxgentexi on 13 February 2017

OpenXM.org

1 DSOLV Functions

This section is a collection of functions to solve regular holonomic systems in terms of series. Algorithms are explained in the book [SST]. You can load this package by the command `load("dsolv.rr")$` This package requires `Diff` and `dmodule`.

To use the functions of the package `dsolv` in OpenXM/Risa/Asir, executing the command `load("dsolv.rr")$` is necessary at first.

This package uses `ox_sm1`, so the variables you can use is as same as those you can use in the package `sm1`.

1.1 Functions

1.1.1 dsolv_dual

`dsolv_dual(`*f*`,`*v*`)`
:: Grobner dual of *f*.

return List

f, *v* List

- It returns the Grobner dual of f in the ring of polynomials with variables v.
- The ideal generated by f must be primary to the maximal ideal generated by v. If it is not primary to the maximal ideal, then this function falls into an infinite loop.

Algorithm: This is an implementation of Algorithm 2.3.14 of the book [SST]. If we replace variables x, y, ... in the output by log(x), log(y), ..., then these polynomials in log are solutions of the system of differential equations f`_(x->x*dx, y->y*dy, ...)`.

```
[435] dsolv_dual([y-x^2,y+x^2],[x,y]);
[x,1]
[436] dsolv_act(y*dy-sm1.mul(x*dx,x*dx,[x,y]),log(x),[x,y]);
0
[437] dsolv_act(y*dy+sm1.mul(x*dx,x*dx,[x,y]),log(x),[x,y]);
0

[439] primadec([y^2-x^3,x^2*y^2],[x,y]);
[[[y^2-x^3,y^4,x^2*y^2],[y,x]]]
[440] dsolv_dual([y^2-x^3,x^2*y^2],[x,y]);
[x*y^3+1/4*x^4*y, x^2*y, x*y^2+1/12*x^4, y^3+x^3*y,
 x^2, x*y, y^2+1/3*x^3, x, y, 1]

[441] dsolv_test_dual();
  Output is  omitted.
```

1.1.2 dsolv_starting_term

dsolv_starting_term(*f*,*v*,*w*)
:: Find the starting term of the solutions of the regular holonomic system *f* to the direction *w*.

return List

f, *v*, *w* List

- Find the starting term of the solutions of the regular holonomic system *f* to the direction *w*.
- The return value is of the form [[*e1*, *e2*, ...], [*s1*, *s2*, ...]] where *e1* is an exponent vector and *s1* is the corresponding solution set, and so on.
- If you set `Dsolv_message_starting_term` to 1, then this function outputs messages during the computation.

Algorithm: Saito, Sturmfels, Takayama, Grobner Deformations of Hypergeometric Differential Equations ([SST]), Chapter 2.

```
[1076]   F = sm1.gkz( [ [[1,1,1,1,1],[1,1,0,-1,0],[0,1,1,-1,0]], [1,0,0]]);
[[x5*dx5+x4*dx4+x3*dx3+x2*dx2+x1*dx1-1,-x4*dx4+x2*dx2+x1*dx1,
  -x4*dx4+x3*dx3+x2*dx2,
  -dx2*dx5+dx1*dx3,dx5^2-dx2*dx4],[x1,x2,x3,x4,x5]]
[1077]  A= dsolv_starting_term(F[0],F[1],[1,1,1,1,0])$
Computing the initial ideal.
Done.
Computing a primary ideal decomposition.
Primary ideal decomposition of the initial Frobenius ideal
to the direction [1,1,1,1,0] is
[[[x5+2*x4+x3-1,x5+3*x4-x2-1,x5+2*x4+x1-1,3*x5^2+(8*x4-6)*x5-8*x4+3,
   x5^2-2*x5-8*x4^2+1,x5^3-3*x5^2+3*x5-1],
 [x5-1,x4,x3,x2,x1]]]

----------- root is [ 0 0 0 0 1 ]
----------- dual system is
[x5^2+(-3/4*x4-1/2*x3-1/4*x2-1/2*x1)*x5+1/8*x4^2
 +(1/4*x3+1/4*x1)*x4+1/4*x2*x3-1/8*x2^2+1/4*x1*x2,
 x4-2*x3+3*x2-2*x1,x5-x3+x2-x1,1]

[1078] A[0];
[[ 0 0 0 0 1 ]]
[1079] map(fctr,A[1][0]);
[[[1/8,1],[x5,1],[log(x2)+log(x4)-2*log(x5),1],
          [2*log(x1)-log(x2)+2*log(x3)+log(x4)-4*log(x5),1]],
 [[1,1],[x5,1],[-2*log(x1)+3*log(x2)-2*log(x3)+log(x4),1]],
 [[1,1],[x5,1],[-log(x1)+log(x2)-log(x3)+log(x5),1]],
 [[1,1],[x5,1]]]
```

Index

(Index is nonexistent)

(Index is nonexistent)

Short Contents

Table of Contents