Introduction to the Janet package

Calling Sequence:

Janet[<function>](args)
<function>(args)

Description:

- The Janet package provides routines for dealing with linear partial differential equations and with polynomial equations. The routines for the polynomial equations are described in the package Involutive, which is a proper subset of the Janet package. Therefore only the differential part will be described here.
- The main algorithm is Janet's algorithm for analysing systems of linear partial differential equations. Given a homogeneous system of linear partial differential equations with analytic coefficients, the command JanetBasis produces an equivalent system by manipulating the original equations and their derivatives such that the new system has the same set of analytic solutions and the Taylor coefficients of these solutions can be computed in a straightforward manner, cf. InvReduce. Also the number of free Taylor coefficients of each order can be read off, cf. HilbertSeries.
- The approach chosen is that of involutive division whereby (independent) variables are separated into multiplicative and non-multiplicative ones. Any Taylor coefficient can uniquely be expressed by the so called parametric derivatives by taking partial derivatives with respect to so called multiplicative variables at the relevant point. The choice of the parametric derivatives at a fixed point, i.e. the parametric Taylor coefficients, can be chosen arbitrarily as the name says, and the Janet basis allows to express any other Taylor coefficient at this point in terms of the parametric ones.
- To use a function of the Janet package, either define that function alone using the command with(Janet, <function>), or define all Janet functions using the command with(Janet). Alternatively, invoke the function using the long form Janet[<function>].
- The functions available in the Janet package are the following:

Basic commands:

JanetBasis	Denominators	ZeroSets
PrincDeriv	ParamDeriv	PDEBasis
InvReduce	HilbertSeries	
SolSeries	PolySol	

Commands for special applications:

CompCond	CompCondBasis	SyzOp
Resolution	ResolutionDim	EulerChar
ShorterResolution	ShortestResolution	SubFactor
WeightedHilbertSeries		
Linearize	Parametrization	Torsion
AutonomEq		
Intersection	LeftInverse	RightInverse
PDEFactorize	CoefficientMatrix	AnnihilatingSystem
BaseChg	ParamBaseChg	GenCoeff

Various invariants related to the `HilbertSeries`:

CartanCharacter	IndexRegularity
HilbertPolynomial	HilbertFunction
HP	HF
PDEHilbertPolynomial	PDEHilbertFunction
PDEHP	PDEHF
PDEHilbertSeries	

Commands to pass back and forth between polynomials and differential expressions:

Diff2Pol	Pol2Diff

Commands dealing with operators in matrix form:

CmpOp	JAdjoint
Diff2Op	AppOp

AddOp SubOp
Pol2Op

Commands dealing with operators represented by polynomials:
Diff2D D2Diff
Op2D D2Op
CmpD

Commands involving jet notation:
Diff2Ind Ind2Diff
Pol2Ind AppOpInd

Alternate Groebner basis command:
DiffGroebnerBasis

Auxiliary commands:
AssertJanetBasis LeadingDeriv AffEqn
JanetOptions JanetStats Jpdesolv

Commands dealing with polynomial rather than differential equations (cf. also Involutive):

InvolutiveBasis	InvolutiveBasisFast	InvolutiveBasisGINV
PolInvReduce	PolInvReduceFast	PolInvReduceGINV
PolTabVar	PolHilbertSeries	FactorModuleBasis
PolZeroSets	PolMinPoly	PolRepres
InvolutivePreprocess	Substitute	Syzygies
SyzygyModule	SyzygyModuleFast	SyzygyModuleGINV
PolResolution	Annihilator	PolCoeff
PolLeftInverse	PolRightInverse	CoeffList
PolWeightedHilbertSeries	Repres	NotHas/Has
PolIndexRegularity	PolDimension	PolCartanCharacter
PolHilbertPolynomial	PolHilbertFunction	
PolHP	PolHF	
LeadingMonomial	AddRhs	AssertInvBasis
InvolutiveOptions	Stats	

- For a description of the basic algorithms, see V. P. Gerdt, "Involutive Algorithms for Computing Groebner Bases", in: S. Cojocaru, G. Pfister, V. Ufnarovski, "Computational Commutative and Non-Commutative Algebraic Geometry", NATO Science Series, IOS Press, 2005, pp. 199-225; or V. P. Gerdt, "Involutive Division Technique: Some Generalizations and Optimizations", Journal of Mathematical Sciences 108(6), 2002, pp. 1034-1051.
- For a description of the packages Involutive and Janet, see Y. A. Blinkov, C. F. Cid, V. P. Gerdt, W. Plesken, D. Robertz, "The MAPLE package 'Janet': I. Polynomial Systems, II. Linear Partial Differential Equations", in: V. G. Ganzha, E. W. Mayr, E. V. Vorozhtsov (eds.), Proceedings of Computer Algebra in Scientific Computing CASC 2003, Passau, pp. 31-40 resp. 41-54.
- For a more general description of Janet's philosophy, see W. Plesken, D. Robertz, "Janet's approach to presentations and resolutions for polynomials and linear pdes", Archiv der Mathematik, 84(1), 2005, pp. 22-37.

Examples:

```
> with(Janet):
```

(Cauchy-Riemann differential equations)

```
> ivar := [x,y]; dvar := [u,v];
```

$$ivar := [x, y]$$

$$dvar := [u, v]$$

```
> L := [diff(u(x,y),x) - diff(v(x,y),y),diff(u(x,y),y) + diff(v(x,y),x)];
```

$$L := \left[\left(\frac{\partial}{\partial x}\mathrm{u}(x,y)\right) - \left(\frac{\partial}{\partial y}\mathrm{v}(x,y)\right), \left(\frac{\partial}{\partial y}\mathrm{u}(x,y)\right) + \left(\frac{\partial}{\partial x}\mathrm{v}(x,y)\right)\right]$$

```
> JB := JanetBasis(L, ivar, dvar);
```

$$JB := \left[\left[\left(\frac{\partial}{\partial y}\mathrm{u}(x,y)\right) + \left(\frac{\partial}{\partial x}\mathrm{v}(x,y)\right), \left(\frac{\partial}{\partial x}\mathrm{u}(x,y)\right) - \left(\frac{\partial}{\partial y}\mathrm{v}(x,y)\right)\right], [x, y], [u, v]\right]$$

```
> PrincDeriv();
```

$$\left[\left(\frac{\partial}{\partial y}\mathrm{u}(x,y)\right) + \left(\frac{\partial}{\partial x}\mathrm{v}(x,y)\right), [x, y], \frac{\partial}{\partial x}\mathrm{v}(x,y)\right]$$

$$\left[\left(\frac{\partial}{\partial x}\mathrm{u}(x,y)\right)-\left(\frac{\partial}{\partial y}\mathrm{v}(x,y)\right),[x,y],\frac{\partial}{\partial x}\mathrm{u}(x,y)\right]$$

```
> f := diff(u(x,y),x$2,y$2);
```

$$f := \frac{\partial^4}{\partial y^2\,\partial x^2}\mathrm{u}(x,y)$$

```
> InvReduce(f, JB);
```

$$-\left(\frac{\partial^4}{\partial y^4}\mathrm{u}(x,y)\right)$$

```
> HilbertSeries(t);
```

$$2+2\frac{t}{1-t}$$

```
> SolSeries(JB,3);
```

$$\left[\mathrm{u}(x,y)=C1_{0,0}+C2_{0,1}x+C1_{0,1}y-\frac{1}{2}C1_{0,2}x^2+C2_{0,2}xy+\frac{1}{2}C1_{0,2}y^2-\frac{1}{6}C2_{0,3}x^3-\frac{1}{2}C1_{0,3}x^2y+\frac{1}{2}C2_{0,3}xy^2+\frac{1}{6}C1_{0,3}y^3,\right.$$
$$\left.\mathrm{v}(x,y)=C2_{0,0}-C1_{0,1}x+C2_{0,1}y-\frac{1}{2}C2_{0,2}x^2-C1_{0,2}xy+\frac{1}{2}C2_{0,2}y^2+\frac{1}{6}C1_{0,3}x^3-\frac{1}{2}C2_{0,3}x^2y-\frac{1}{2}C1_{0,3}xy^2+\frac{1}{6}C2_{0,3}y^3\right],$$
$$[C1_{0,0},C2_{0,1},C1_{0,1},C1_{0,2},C2_{0,2},C2_{0,3},C1_{0,3},C2_{0,0}]$$

See Also:

with, Involutive, JanetOre.

Janet[AddOp] - add two linear differential operators in matrix form

Calling Sequence:

AddOp(oper1,oper2,ivar)

Parameters:

`oper1` - linear differential operator in matrix form
`oper2` - linear differential operator in matrix form
`ivar` - list of independent variables

Description:

- ***AddOp*** forms the sum of the linear differential operators given by the matrices **`oper1`** and **`oper2`**, the entries of which are described below.
- The entries of the input and output matrices have the form $[[c_1, [...]], ..., [c_r, [...]]]$, where the dots in the innermost brackets are sequences of independent variables from **`ivar`** representing partial derivatives w. r. t. these variables, and the c_i are the coefficients for these, i. e. functions in **`ivar`**. The whole bracket represents the sum of these items, i. e. the linear combination of the above operators. Note, if an innermost bracket is empty, it represents the identity operator. (For a convenient way to produce such a matrix use Diff2Op as in the example below.)

Examples:

```
> with(Janet):
```

Example 1:

```
> ivar := [x]; dvar := [u];
```

$$ivar := [x]$$

$$dvar := [u]$$

```
> o1 := Diff2Op([diff(u(x), x)+u(x)], ivar, dvar);
```

$$o1 := [[[1, [x]], [1, [\ ]]]]$$

```
> o2 := Diff2Op([x*diff(u(x), x)], ivar, dvar);
```

$$o2 := [[[x, [x]]]]$$

```
> AddOp(o1, o2, ivar);
```

$$[[[1+x, [x]], [1, [\ ]]]]$$

Example 2:

```
> ivar := [x,y]; dvar := [u,v];
```

$$ivar := [x, y]$$

$$dvar := [u, v]$$

```
> L1 := [y*diff(u(x,y),y)+diff(v(x,y),x$2), diff(v(x,y),x)+2*x^2*v(x,y)];
```

$$L1 := \left[y\left(\frac{\partial}{\partial y}\mathrm{u}(x,y)\right)+\left(\frac{\partial^2}{\partial x^2}\mathrm{v}(x,y)\right), \left(\frac{\partial}{\partial x}\mathrm{v}(x,y)\right)+2x^2\,\mathrm{v}(x,y)\right]$$

```
> o1 := Diff2Op(L1, ivar, dvar);
```

$$o1 := \begin{bmatrix} [[y, [y]]] & [[1, [x, x]]] \\ 0 & [[1, [x]], [2x^2, [\ ]]] \end{bmatrix}$$

```
> L2 := [u(x,y)+diff(u(x,y),x), diff(v(x,y),y)];
```

$$L2 := \left[\mathrm{u}(x,y)+\left(\frac{\partial}{\partial x}\mathrm{u}(x,y)\right), \frac{\partial}{\partial y}\mathrm{v}(x,y)\right]$$

```
> o2 := Diff2Op(L2, ivar, dvar);
```

$$o2 := \begin{bmatrix} [[1, [x]], [1, [\ ]]] & 0 \\ 0 & [[1, [y]]] \end{bmatrix}$$

```
> AddOp(o1, o2, ivar);
```

$$\begin{bmatrix} [[1, [x]], [y, [y]], [1, [\ ]]] & [[1, [x, x]]] \\ 0 & [[1, [x]], [1, [y]], [2\,x^2, [\ ]]] \end{bmatrix}$$

See Also:

Diff2Op, SubOp, AppOp, AppOpInd, AddOp, SubOp, JAdjoint, Op2D, D2Op, CmpD.

Janet[AffEqn] - convert linear homogeneous PDEs to inhomogeneous ones

Calling Sequence:

AffEqn(L,ivar,rhs)

Parameters:

`L` - list (of arbitrary entries), usually list of differential expressions
`ivar` - list of independent variables
`rhs` - list (same length as **`L`**) of right hand sides, i. e. the affine part of the inhomogeneous system

Description:

- ***AffEqn*** substitutes each entry **`m`** of **`L`** by **`m-e(op(ivar))`**, where **`e`** is the *i*-th entry of **`rhs`**. The command serves similar purposes as AddRhs in the polynomial context.
- If the input represents a linear homogeneous PDE system, the output can be viewed as an affine PDE system and can be processed by JanetBasis thus yielding equations for **`rhs`** which have to be satisfied and can be displayed by CompCond.

Example:

```
> with(Janet):
> ivar := [x,y,t];
```

$$ivar := [x, y, t]$$

```
> L := [diff(u(x,y,t),x$2) - y*diff(u(x,y,t),t$2), diff(u(x,y,t),y$2)];
```

$$L := \left[\left(\frac{\partial^2}{\partial x^2}\,u(x,y,t)\right) - y\left(\frac{\partial^2}{\partial t^2}\,u(x,y,t)\right), \frac{\partial^2}{\partial y^2}\,u(x,y,t)\right]$$

```
> AffEqn(L, ivar, [a1, a2]);
```

$$\left[\left(\frac{\partial^2}{\partial x^2}\,u(x,y,t)\right) - y\left(\frac{\partial^2}{\partial t^2}\,u(x,y,t)\right) - a1(x,y,t), \left(\frac{\partial^2}{\partial y^2}\,u(x,y,t)\right) - a2(x,y,t)\right]$$

```
> AffEqn(L, ivar, [t^2, 0]);
```

$$\left[\left(\frac{\partial^2}{\partial x^2}\,u(x,y,t)\right) - y\left(\frac{\partial^2}{\partial t^2}\,u(x,y,t)\right) - t^2, \frac{\partial^2}{\partial y^2}\,u(x,y,t)\right]$$

See Also:

JanetBasis, CompCond, CompCondBasis, AddRhs.

Janet[AnnihilatingSystem] - Janet basis for linear differential relations satisfied by a linear differential expression up to a certain order

Calling Sequence:

AnnihilatingSystem(U,ivar,dvar,unkn,der,uabl,dabl)

Parameters:

`U`	-	differential expression which is linear in the dependent variables
`ivar`	-	list of independent variables
`dvar`	-	list of dependent variables
`unkn`	-	list of one variable (being the dependent variable for the resulting Janet basis)
`der`	-	list of monomials in the independent variables, or a non-negative integer
`uabl`	-	(optional) unevaluated name for output list indicating partial derivatives of **U**
`dabl`	-	(optional) unevaluated name for output list indicating partial derivatives of the dependent variables

Description:

- ***AnnihilatingSystem*** returns a Janet basis (in the unknown function given in **unkn**) for the linear differential relations up to a certain order which are satisfied by **U**.
- The differential expression **U** is assumed to be linear in (analytic) functions which are listed as dependent variables in **dvar**. The arguments of these functions may be different from the given list **ivar** of independent variables in the sense that these functions may be composed with other known or unknown (analytic) functions of **ivar**, arbitrary in number.
- We outline some typical cases of application: Assume **U** is not a sum, but only a (certain multiple of a) function *f* listed in **dvar** (possibly composed with other (analytic) functions of **ivar**). If the arguments of this function *f* are functionally independent, then, around almost all points in the domain of definition of **U**, coordinates can be chosen such that the arguments of *f* are expressed as distinct coordinates in the new coordinate system. If now the number of arguments of *f* is less than the number of independent variables, then *f* does not depend on certain of the new coordinates, which gives rise to linear partial differential equations of first order satisfied by *f* in the given coordinates. If **U** equals *g f*, where *g* is a given (analytic) function of the independent variables, then linear partial differential equations of first order satisfied by **U** can be obtained by the same process when *f* is replaced by **U**/*g* (cf. Example 1 below). Locally, i.e. around almost all points in the domain of definition of **U**, these linear PDEs of first order characterize **U** in the sense that their common (analytic) solutions are exactly the (analytic) functions of the form prescribed by **U**.
- If **U** is a linear combination of (possibly composed) functions listed in **dvar** with coefficients that are given (analytic) functions, then the left ideal of differential operators which annihilate (locally) exactly the (analytic) functions of the form prescribed by **U** equals the intersection of the left ideals that are generated by the PDEs of first order obtained for the summands of **U** in the way explained above (cf. Example 2 below).
- ***AnnihilatingSystem*** does not exactly follow the above explanations, but computes linear PDEs satisfied by **U** as linear relations satisfied by the rows of a matrix representing partial derivatives of **U** as constructed by CoefficientMatrix. An entry of this matrix is the coefficient of the partial derivative of a function listed in **dvar** (referred to by the column index) in a partial derivative of **U** (referred to by the row index); cf. CoefficientMatrix for more details.
- If **der** is given as non-negative integer, then ***AnnihilatingSystem*** represents all partial derivatives of **U** up to order **der**-1 as a matrix and computes the linear relations among its rows. (In fact, ***AnnihilatingSystem*** takes advantage of the block structure of these matrices when performing Gaussian elimination.) If **der** is given as a list of monomials, then only the partial derivatives of **U** corresponding to the given monomials are taken into account (cf. Example 3).
- The result of ***AnnihilatingSystem*** is a Janet basis in the unknown function given in **unkn** in the same format as the result of JanetBasis .
- Since ***AnnihilatingSystem*** stops the computation of linear relations at a certain differential order, the returned Janet basis might not generate the left ideal of all differential operators which annihilate (analytic) functions of the form prescribed by **U**. A separate proof that it is a generating set has to be given if this is true. Sometimes such a proof can be given using Hilbert series, see Example 4

below.

- The optional arguments **uabl** and **dabl** can used to obtain the lists of partial derivatives corresponding to the rows resp. the columns of the matrix used by ***AnnihilatingSystem*** (cf. Example 5 below), see also CoefficientMatrix. In both cases the argument is expected to be an unevaluated name to which ***AnnihilatingSystem*** assigns the corresponding list. These lists are structured as lists of lists according to increasing differential order. The argument **uabl** may be given without providing a name for **dabl**. However, **uabl** must be given if **dabl** is present in the list of arguments.
- For more information about the systems of partial differential equations arising in the above way, see W. Plesken, D. Robertz, "Linear differential elimination for analytic functions", preprint, RWTH Aachen University, 2010.

Examples:

```
> with(Janet):
```

Example 1: a few small examples

```
> ivar := [x,y];
```

$$ivar := [x, y]$$

```
> AnnihilatingSystem(f(y), ivar, [f], [u], 2);
```

$$\left[\left[\frac{\partial}{\partial x}\mathrm{u}(x,y)\right], [x,y], [u]\right]$$

```
> AnnihilatingSystem(f(x,y), ivar, [f], [u], 2);
```

$$[[0], [x,y], [u]]$$

```
> AnnihilatingSystem(exp(x)*f(y), ivar, [f], [u], 2);
```

$$\left[\left[-\mathrm{u}(x,y)+\left(\frac{\partial}{\partial x}\mathrm{u}(x,y)\right)\right], [x,y], [u]\right]$$

```
> AnnihilatingSystem(exp(x-y)*f(x+y), ivar, [f], [u], 2);
```

$$\left[\left[-2\,\mathrm{u}(x,y)-\left(\frac{\partial}{\partial y}\mathrm{u}(x,y)\right)+\left(\frac{\partial}{\partial x}\mathrm{u}(x,y)\right)\right], [x,y], [u]\right]$$

Example 2:

```
> ivar := [x,y];
```

$$ivar := [x, y]$$

```
> U := f1(x)*y+f2(y)*x;
```

$$U := \mathrm{f1}(x)\,y+\mathrm{f2}(y)\,x$$

```
> J := AnnihilatingSystem(U, ivar, [f1,f2], [u], 3);
```

$$J := \left[\left[\mathrm{u}(x,y)-\left(\frac{\partial}{\partial y}\mathrm{u}(x,y)\right)y-\left(\frac{\partial}{\partial x}\mathrm{u}(x,y)\right)x+\left(\frac{\partial^2}{\partial y\,\partial x}\mathrm{u}(x,y)\right)yx\right], [x,y], [u]\right]$$

```
> J1 := AnnihilatingSystem(f1(x)*y, ivar, [f1], [u], 2);
```

$$J1 := \left[\left[-\mathrm{u}(x,y)+\left(\frac{\partial}{\partial y}\mathrm{u}(x,y)\right)y\right], [x,y], [u]\right]$$

```
> J2 := AnnihilatingSystem(f2(y)*x, ivar, [f2], [u], 2);
```

$$J2 := \left[\left[-\mathrm{u}(x,y)+\left(\frac{\partial}{\partial x}\mathrm{u}(x,y)\right)x\right], [x,y], [u]\right]$$

```
> Intersection(J1[1], J2[1], ivar, [u]);
```

$$\left[\left[\mathrm{u}(x,y)-\left(\frac{\partial}{\partial y}\mathrm{u}(x,y)\right)y-\left(\frac{\partial}{\partial x}\mathrm{u}(x,y)\right)x+\left(\frac{\partial^2}{\partial y\,\partial x}\mathrm{u}(x,y)\right)yx\right], [x,y], [u]\right]$$

Example 3:

```
> ivar := [x,y];
```

$$ivar := [x, y]$$

```
> U := f1(x^2-y)+f2(y^2-x);
```

$$U := \mathrm{f1}(x^2-y)+\mathrm{f2}(y^2-x)$$

```
> AnnihilatingSystem(U, ivar, [f1,f2], [u], 3);
```

$$\left[\left[\left(-\frac{1}{2}+8y^2x^2\right)\left(\frac{\partial^2}{\partial y\,\partial x}u(x,y)\right)+(-2x-4y^2)\left(\frac{\partial}{\partial x}u(x,y)\right)+(-2y-4x^2)\left(\frac{\partial}{\partial y}u(x,y)\right)+(-x+4x^2y)\left(\frac{\partial^2}{\partial y^2}u(x,y)\right)+(-y+4y^2x)\left(\frac{\partial^2}{\partial x^2}u(x,y)\right),[x,y],[u]\right]\right]$$

```
> AnnihilatingSystem(U, ivar, [f1,f2], [u], [y^2,x*y,x^2]);
```

$$[[0],[x,y],[u]]$$

```
> AnnihilatingSystem(U, ivar, [f1,f2], [u], [y,x,y^2,x*y,x^2]);
```

$$\left[\left[\left(-\frac{1}{2}+8y^2x^2\right)\left(\frac{\partial^2}{\partial y\,\partial x}u(x,y)\right)+(-2x-4y^2)\left(\frac{\partial}{\partial x}u(x,y)\right)+(-2y-4x^2)\left(\frac{\partial}{\partial y}u(x,y)\right)+(-x+4x^2y)\left(\frac{\partial^2}{\partial y^2}u(x,y)\right)+(-y+4y^2x)\left(\frac{\partial^2}{\partial x^2}u(x,y)\right),[x,y],[u]\right]\right]$$

Example 4: a proof that the annihilating system is complete using Hilbert series

```
> ivar := [x,y];
```

$$ivar := [x,y]$$

```
> U := f1(x)*y+f2(y)*x;
```

$$U := f1(x)\,y+f2(y)\,x$$

```
> J := AnnihilatingSystem(U, ivar, [f1,f2], [u], 3);
```

$$J := \left[\left[u(x,y)-\left(\frac{\partial}{\partial y}u(x,y)\right)y-\left(\frac{\partial}{\partial x}u(x,y)\right)x+\left(\frac{\partial^2}{\partial y\,\partial x}u(x,y)\right)yx\right],[x,y],[u]\right]$$

The annihilating system computed up to differential order 3 has the following Hilbert series:

```
> AssertJanetBasis(op(J)); H := HilbertSeries(t);
```

$$\left[\left[u(x,y)-\left(\frac{\partial}{\partial y}u(x,y)\right)y-\left(\frac{\partial}{\partial x}u(x,y)\right)x+\left(\frac{\partial^2}{\partial y\,\partial x}u(x,y)\right)yx\right],[x,y],[u]\right]$$

$$H := 1+2\,t+\frac{2\,t^2}{1-t}$$

We compute an annihilating system for the functions that can be expressed in terms of both summands of **U** (separately):

```
> J1 := AnnihilatingSystem(f1(x)*y, ivar, [f1], [u], 2);
```

$$J1 := \left[\left[-u(x,y)+\left(\frac{\partial}{\partial y}u(x,y)\right)y\right],[x,y],[u]\right]$$

```
> J2 := AnnihilatingSystem(f2(y)*x, ivar, [f2], [u], 2);
```

$$J2 := \left[\left[-u(x,y)+\left(\frac{\partial}{\partial x}u(x,y)\right)x\right],[x,y],[u]\right]$$

```
> JanetBasis([op(J1[1]), op(J2[1])], ivar, [u]);
```

$$\left[\left[-u(x,y)+\left(\frac{\partial}{\partial y}u(x,y)\right)y,-u(x,y)+\left(\frac{\partial}{\partial x}u(x,y)\right)x\right],[x,y],[u]\right]$$

```
> pdsolve(%[1]);
```

$$\{u(x,y)=\_C1\,y\,x\}$$

The Hilbert series for the intersection of the two function spaces is:

```
> Hcommon := HilbertSeries(t);
```

$$Hcommon := 1$$

If the function spaces given by the two summands in **U** had a trivial intersection, then the sum of these spaces would have the following Hilbert series (two functions of one argument):

```
> HU := 2/(1-t);
```

$$HU := \frac{2}{1-t}$$

```
> simplify(H - (HU - Hcommon));
```

$$0$$

This shows that the above annihilating system for **U** is complete.

Example 5:

```
> ivar := [x,y];
```

$$ivar := [x, y]$$

```
> U := cos(x)*f1(sin(x+y)) + f2(cos(x-y));
```

$$U := \cos(x)\,\mathrm{f1}(\sin(x+y)) + \mathrm{f2}(\cos(x-y))$$

```
> J := AnnihilatingSystem(U, ivar, [f1,f2], [u], 4, 'uabl', 'dabl');
```

$$J := \left[\left[\left(\frac{1}{2}\cos(x)\cos(y)+\frac{1}{2}\cos(x)^3\cos(y)+\frac{1}{2}\cos(x)^2\sin(x)\sin(y)\right)\left(\frac{\partial}{\partial x}\mathrm{u}(x,y)\right)\right.\right.$$
$$+\left(\frac{1}{2}\cos(x)\cos(y)+\frac{1}{2}\cos(x)^3\cos(y)+\frac{1}{2}\cos(x)^2\sin(x)\sin(y)\right)\left(\frac{\partial}{\partial y}\mathrm{u}(x,y)\right)$$
$$+(-\cos(x)^3\sin(y)+\sin(x)\cos(y)+\cos(x)^2\sin(x)\cos(y)+\cos(x)\sin(y))\left(\frac{\partial^2}{\partial y\,\partial x}\mathrm{u}(x,y)\right)$$
$$+\left(\frac{3}{2}\cos(x)^2\sin(x)\cos(y)+\frac{3}{2}\cos(x)\sin(y)+\frac{3}{2}\sin(x)\cos(y)-\frac{3}{2}\cos(x)^3\sin(y)\right)\left(\frac{\partial^2}{\partial y^2}\mathrm{u}(x,y)\right)$$
$$+(\cos(x)^2\sin(x)\sin(y)+\cos(x)\cos(y)+\cos(x)^3\cos(y))\left(\frac{\partial^3}{\partial y\,\partial x^2}\mathrm{u}(x,y)\right)$$
$$+\left(-\frac{1}{2}\cos(x)\sin(y)+\frac{1}{2}\cos(x)^3\sin(y)-\frac{1}{2}\cos(x)^2\sin(x)\cos(y)-\frac{1}{2}\sin(x)\cos(y)\right)\left(\frac{\partial^2}{\partial x^2}\mathrm{u}(x,y)\right)$$
$$+(-\cos(x)\cos(y)-\cos(x)^2\sin(x)\sin(y)-\cos(x)^3\cos(y))\left(\frac{\partial^3}{\partial y^3}\mathrm{u}(x,y)\right),$$
$$\left(\frac{3}{2}\cos(x)^2\sin(x)\sin(y)+\frac{3}{2}\cos(x)^3\cos(y)+\frac{3}{2}\cos(x)\cos(y)\right)\left(\frac{\partial}{\partial x}\mathrm{u}(x,y)\right)$$
$$+\left(\frac{3}{2}\cos(x)^2\sin(x)\sin(y)+\frac{3}{2}\cos(x)^3\cos(y)+\frac{3}{2}\cos(x)\cos(y)\right)\left(\frac{\partial}{\partial y}\mathrm{u}(x,y)\right)$$
$$+(-\cos(x)^3\sin(y)+\sin(x)\cos(y)+\cos(x)^2\sin(x)\cos(y)+\cos(x)\sin(y))\left(\frac{\partial^2}{\partial y\,\partial x}\mathrm{u}(x,y)\right)$$
$$+\left(\frac{3}{2}\cos(x)^2\sin(x)\cos(y)+\frac{3}{2}\cos(x)\sin(y)+\frac{3}{2}\sin(x)\cos(y)-\frac{3}{2}\cos(x)^3\sin(y)\right)\left(\frac{\partial^2}{\partial y^2}\mathrm{u}(x,y)\right)$$
$$+(\cos(x)^2\sin(x)\sin(y)+\cos(x)\cos(y)+\cos(x)^3\cos(y))\left(\frac{\partial^3}{\partial x^3}\mathrm{u}(x,y)\right)$$
$$+(-\cos(x)\cos(y)-\cos(x)^2\sin(x)\sin(y)-\cos(x)^3\cos(y))\left(\frac{\partial^3}{\partial y^2\,\partial x}\mathrm{u}(x,y)\right)$$
$$\left.\left.+\left(-\frac{1}{2}\cos(x)\sin(y)+\frac{1}{2}\cos(x)^3\sin(y)-\frac{1}{2}\cos(x)^2\sin(x)\cos(y)-\frac{1}{2}\sin(x)\cos(y)\right)\left(\frac{\partial^2}{\partial x^2}\mathrm{u}(x,y)\right)\right],[x,y],[u]\right]$$

```
> uabl; dabl;
```

$$[[1],[y,x],[y^2,yx,x^2],[y^3,y^2x,yx^2,x^3]]$$

$$[[\mathrm{f1}(\sin(x+y)),\mathrm{f2}(\cos(x-y))],[\mathrm{D}(f1)(\sin(x+y)),\mathrm{D}(f2)(\cos(x-y))],[(\mathrm{D}^{(2)})(f1)(\sin(x+y)),(\mathrm{D}^{(2)})(f2)(\cos(x-y))],$$
$$[(\mathrm{D}^{(3)})(f1)(\sin(x+y)),(\mathrm{D}^{(3)})(f2)(\cos(x-y))]]$$

Janet[AppOp] - apply differential operator

Calling Sequence:

AppOp(oper,e,ivar,dvar)

Parameters:

`oper` - differential operator (matrix with entries as described below)
`e` - list of differential expressions (of length equal to the number of columns of **`oper`**)
`ivar` - list of independent variables
`dvar` - list of dependent variables

Description:

- ***AppOp*** applies the differential operator represented by the matrix **`oper`** to the tuple of differential expressions represented by **`e`** (in column convention).
- The entries of the matrix **`oper`** must have the form $[[c_1, [...]], ..., [c_r, [...]]]$, where the dots in the innermost brackets are sequences of independent variables from **`ivar`** representing partial derivatives w. r. t. these variables, and the c_i are the coefficients for these, i. e. functions in **`ivar`**. The whole bracket represents the sum of these items, i. e. the linear combination of the above operators. Note, if an innermost bracket is empty, it represents the identity operator. (For a convenient way to produce such a matrix use Diff2Op as in the example below.)
- In the special case where **`e`** consists of the functions represented by **`dvar`** the command ***AppOp*** is inverse to Diff2Op.

Examples:

```
> with(Janet):
> ivar := [x,y,z]; dvar := [u,v];
```

$$ivar := [x, y, z]$$

$$dvar := [u, v]$$

```
> [diff(u(x,y,z),y)+diff(v(x,y,z),'$'(x,2)), diff(v(x,y,z),z)+diff(u(x,y,z),'$'(y,2))];
```

$$\left[\left(\frac{\partial}{\partial y}\mathrm{u}(x,y,z)\right)+\left(\frac{\partial^2}{\partial x^2}\mathrm{v}(x,y,z)\right),\left(\frac{\partial}{\partial z}\mathrm{v}(x,y,z)\right)+\left(\frac{\partial^2}{\partial y^2}\mathrm{u}(x,y,z)\right)\right]$$

```
> oper := Diff2Op(%, ivar, dvar);
```

$$oper := \begin{bmatrix} [[1,[y]]] & [[1,[x,x]]] \\ [[1,[y,y]]] & [[1,[z]]] \end{bmatrix}$$

```
> e := [u(x,y,z), v(x,y,z)];
```

$$e := [\mathrm{u}(x,y,z), \mathrm{v}(x,y,z)]$$

```
> AppOp(oper, e, ivar, dvar);
```

$$\left[\left(\frac{\partial}{\partial y}\mathrm{u}(x,y,z)\right)+\left(\frac{\partial^2}{\partial x^2}\mathrm{v}(x,y,z)\right),\left(\frac{\partial}{\partial z}\mathrm{v}(x,y,z)\right)+\left(\frac{\partial^2}{\partial y^2}\mathrm{u}(x,y,z)\right)\right]$$

The following is a possible abbreviation (not quite according to the description above):

```
> AppOp(oper, dvar, ivar, dvar);
```

$$\left[\left(\frac{\partial}{\partial y}\mathrm{u}(x,y,z)\right)+\left(\frac{\partial^2}{\partial x^2}\mathrm{v}(x,y,z)\right),\left(\frac{\partial}{\partial z}\mathrm{v}(x,y,z)\right)+\left(\frac{\partial^2}{\partial y^2}\mathrm{u}(x,y,z)\right)\right]$$

```
> e2 := [diff(u(x,y,z), x)+v(x,y,z), 0];
```

$$e2 := \left[\left(\frac{\partial}{\partial x}\mathrm{u}(x,y,z)\right)+\mathrm{v}(x,y,z), 0\right]$$

```
> AppOp(oper, e2, ivar, dvar);
```

$$\left[\left(\frac{\partial^2}{\partial y\,\partial x}\mathrm{u}(x,y,z)\right)+\left(\frac{\partial}{\partial y}\mathrm{v}(x,y,z)\right),\left(\frac{\partial^3}{\partial y^2\,\partial x}\mathrm{u}(x,y,z)\right)+\left(\frac{\partial^2}{\partial y^2}\mathrm{v}(x,y,z)\right)\right]$$

See Also:

JanetBasis, Diff2Op, Pol2Op, CmpOp, AddOp, SubOp, JAdjoint, AppOpInd, Diff2Ind, Ind2Diff, Diff2Pol, Pol2Diff, Op2D, Diff2D, D2Diff.

Janet[AppOpInd] - apply linear differential operator to jet expressions

Calling Sequence:

AppOpInd(oper,e,ivar,dvar)

Parameters:

`oper` - differential operator in matrix form (as described below)
`e` - list of differential expressions in jet notation (of length equal to the number of columns of **`oper`**)
`ivar` - list of indeterminates of the polynomial ring (which become the independent variables)
`dvar` - list of dependent variables

Description:

- ***AppOpInd*** applies the differential operator represented by the matrix **`oper`** to the tuple of differential expressions in jet notation represented by **`e`** (in column convention). The same is achieved by AppOp for Maple differential expression input and output.
- The entries of the matrix **`oper`** must have the form [[c_1, [...]], ..., [c_r, [...]]], where the dots in the innermost brackets are sequences of independent variables from **`ivar`** representing partial derivatives w. r. t. these variables, and the c_i are the coefficients for these, i. e. functions in **`ivar`**. The whole bracket represents the sum of these items, i. e. the linear combination of the above operators. Note, if an innermost bracket is empty, it represents the identity operator. (For a convenient way to produce such a matrix use Diff2Op as in the example below.)

Examples:

```
> with(Janet):
> ivar := [x,y,z]; dvar := [u,v];
```

$$ivar := [x, y, z]$$

$$dvar := [u, v]$$

```
> L := [u[y]+v[x,x],u[y,y]+v[z]];
```

$$L := [u_y + v_{x,x}, u_{y,y} + v_z]$$

```
> Lex := Ind2Diff(L, ivar, dvar);
```

$$Lex := \left[\left(\frac{\partial}{\partial y}\mathrm{u}(x,y,z)\right)+\left(\frac{\partial^2}{\partial x^2}\mathrm{v}(x,y,z)\right),\left(\frac{\partial^2}{\partial y^2}\mathrm{u}(x,y,z)\right)+\left(\frac{\partial}{\partial z}\mathrm{v}(x,y,z)\right)\right]$$

```
> oper := Diff2Op(Lex, ivar, dvar);
```

$$oper := \begin{bmatrix} [[1,[y]]] & [[1,[x,x]]] \\ [[1,[y,y]]] & [[1,[z]]] \end{bmatrix}$$

```
> AppOpInd(oper, [u[x,x],v], ivar, dvar);
```

$$[u_{x,x,y} + v_{x,x}, u_{x,x,y,y} + v_z]$$

```
> AppOp(oper, [diff(u(x,y,z), x, x), v(x,y,z)], ivar, dvar);
```

$$\left[\left(\frac{\partial^3}{\partial y\,\partial x^2}\mathrm{u}(x,y,z)\right)+\left(\frac{\partial^2}{\partial x^2}\mathrm{v}(x,y,z)\right),\left(\frac{\partial^4}{\partial y^2\,\partial x^2}\mathrm{u}(x,y,z)\right)+\left(\frac{\partial}{\partial z}\mathrm{v}(x,y,z)\right)\right]$$

See Also:

Diff2Ind, Ind2Diff, Pol2Ind, Diff2Op, Pol2Op, AppOp, Diff2Pol, Pol2Diff, Op2D, D2Op, Diff2D, D2Diff

Janet[AssertJanetBasis] - assure the system that differential expressions form a Janet basis

Calling Sequence:

AssertJanetBasis(L,ivar,dvar,oivar,ord)

Parameters:

`L` - list of linear differential expressions
`ivar` - list of independent variables
`dvar` - list of dependent variables
`oivar` - (optional) list of the independent variables in varied order
`ord` - (optional) change of ordering for differential expressions

Description:

- The internal result of the command ***AssertJanetBasis*** is that the data structure for the current Janet basis is set up such that commands like HilbertSeries, ParamDeriv, PrincDeriv, etc. can be invoked.
- All parameters to ***AssertJanetBasis*** have the same meaning as in JanetBasis.
- The output is a list containing **`L`** as first entry, the parameters **`ivar`**, **`dvar`**, and **`oivar`** (optional) being appended.
- One typical situation where ***AssertJanetBasis*** is used is to make an earlier computed Janet basis after at least one further call of ***JanetBasis*** again to the current Janet basis without recomputing it.
- Another, usually more important, use for ***AssertJanetBasis*** is for big PDE systems with constant coefficients. In this case one can apply InvolutiveBasisFast to the corresponding polynomial system (cf. Diff2Pol), translate the resulting Janet basis to a PDE system by using Pol2Diff, and then apply ***AssertJanetBasis.***

Examples:

```
> with(Janet):
```

Example 1: Working with two Janet bases

```
> ivar := [x,y]: dvar := [u,v]:
> L1 := [diff(u(x,y),x) - diff(v(x,y),y)];
```

$$L1 := \left[\left(\frac{\partial}{\partial x}\mathrm{u}(x,y)\right) - \left(\frac{\partial}{\partial y}\mathrm{v}(x,y)\right)\right]$$

```
> J1 := JanetBasis(L1, ivar, dvar):
> HilbertSeries(t);
```

$$2 + t\left(2\frac{1}{1-t} + \frac{1}{(1-t)^2}\right)$$

```
> L2 := [diff(u(x,y),x) - diff(v(x,y),y),diff(u(x,y),y) + diff(v(x,y),x)];
```

$$L2 := \left[\left(\frac{\partial}{\partial x}\mathrm{u}(x,y)\right) - \left(\frac{\partial}{\partial y}\mathrm{v}(x,y)\right), \left(\frac{\partial}{\partial y}\mathrm{u}(x,y)\right) + \left(\frac{\partial}{\partial x}\mathrm{v}(x,y)\right)\right]$$

```
> J2 := JanetBasis(L2, ivar, dvar):
> HilbertSeries(t);
```

$$2 + 2\frac{t}{1-t}$$

```
> AssertJanetBasis(op(J1)):
> HilbertSeries(t);
```

$$2 + t\left(2\frac{1}{1-t} + \frac{1}{(1-t)^2}\right)$$

Example 2: Using results of C++ commands for the polynomial case

```
> with(Involutive):
```

```
> ivar := [x,y,z]: dvar := [u,v]:
> L := [diff(u(x,y,z),x$2) - diff(v(x,y,z),x,y),diff(u(x,y,z),y$3) +
  diff(v(x,y,z),x$2),diff(u(x,y,z),x,y,z) + diff(v(x,y,z),x,y$2)];
```

$$L := \left[\left(\frac{\partial^2}{\partial x^2}\,u(x,y,z)\right) - \left(\frac{\partial^2}{\partial y\,\partial x}\,v(x,y,z)\right), \left(\frac{\partial^3}{\partial y^3}\,u(x,y,z)\right) + \left(\frac{\partial^2}{\partial x^2}\,v(x,y,z)\right), \left(\frac{\partial^3}{\partial z\,\partial y\,\partial x}\,u(x,y,z)\right) + \left(\frac{\partial^3}{\partial y^2\,\partial x}\,v(x,y,z)\right)\right]$$

```
> P := Diff2Pol(L, ivar, dvar);
```

$$P := [[x^2, -xy], [y^3, x^2], [xyz, xy^2]]$$

```
> IB := InvolutiveBasisFast(P, ivar);
```

$$IB := [[x^2, -xy], [y^3, x^2], [xyz, xy^2], [xy^3, x^3], [-xyz^2, x^2 y^2], [0, x^4 + x^3 z], [xyz^3, x^3 y^2]]$$

```
> J := Pol2Diff(IB, ivar, dvar);
```

$$J := \left[\left(\frac{\partial^2}{\partial x^2}\,u(x,y,z)\right) - \left(\frac{\partial^2}{\partial y\,\partial x}\,v(x,y,z)\right), \left(\frac{\partial^3}{\partial y^3}\,u(x,y,z)\right) + \left(\frac{\partial^2}{\partial x^2}\,v(x,y,z)\right), \left(\frac{\partial^3}{\partial z\,\partial y\,\partial x}\,u(x,y,z)\right) + \left(\frac{\partial^3}{\partial y^2\,\partial x}\,v(x,y,z)\right), \left(\frac{\partial^4}{\partial y^3\,\partial x}\,u(x,y,z)\right) + \left(\frac{\partial^3}{\partial x^3}\,v(x,y,z)\right), -\left(\frac{\partial^4}{\partial z^2\,\partial y\,\partial x}\,u(x,y,z)\right) + \left(\frac{\partial^4}{\partial y^2\,\partial x^2}\,v(x,y,z)\right), \left(\frac{\partial^4}{\partial x^4}\,v(x,y,z)\right) + \left(\frac{\partial^4}{\partial z\,\partial x^3}\,v(x,y,z)\right), \left(\frac{\partial^5}{\partial z^3\,\partial y\,\partial x}\,u(x,y,z)\right) + \left(\frac{\partial^5}{\partial y^2\,\partial x^3}\,v(x,y,z)\right)\right]$$

```
> JB := AssertJanetBasis(J, ivar, dvar):
> ParamDeriv(ivar, dvar);
```

$$\left[\frac{y^2}{1-z} + \frac{y}{1-z} + \frac{1}{1-z} + \frac{xy^2}{1-z} + \frac{xy}{1-z} + \frac{x}{1-z}, \frac{xy}{1-z} + \frac{x}{1-z} + \frac{x^2 y}{1-z} + \frac{x^2}{1-z} + \frac{x^3 y}{1-z} + \frac{x^3}{1-z} + \frac{1}{(1-y)(1-z)}\right]$$

```
> HilbertSeries(t);
```

$$2 + 6\,t + 11\,t^2 + 15\,t^3 + 17\,t^4 + t^5\left(17\,\frac{1}{1-t} + \frac{1}{(1-t)^2}\right)$$

See Also:

JanetBasis, PrincDeriv, ParamDeriv, InvReduce, CompCond, CompCondBasis, HilbertSeries, Diff2Pol, Pol2Diff, InvolutiveBasisFast, AssertInvBasis.

Janet[AutonomEq] - find linear differential equations that an autonomous element satisfies

Calling Sequence:

AutonomEq(p,L,ivar,dvar)

Parameters:

`p` - linear differential expression
`L` - list of linear differential expressions (to be interpreted as module generators)
`ivar` - list of independent variables
`dvar` - list of dependent variables

Description:

- ***AutonomEq*** returns a list which contains a generating set for all linear differential equations that the residue class of **p** in the cokernel of **L** satisfies.
- The linear differential expression **p** is considered as a representative of an element *P* of the right D-module whose presentation is given by **L**. ***AutonomEq*** then yields a generating set for the annihilator of *P* in D.
- The output is a list of linear differential expressions in the dependent variable _A. Substituting **p** for _A in this list gives equations which are zero modulo **L** (cf. InvReduce).

Examples:

```
> with(Janet):
> ivar := [t]; dvar := [u,v,w];
```

$$ivar := [t]$$

$$dvar := [u, v, w]$$

```
> R := Ind2Diff([u[t]-v-w, -u+v[t]+w], ivar, dvar);
```

$$R := \left[\left(\frac{d}{dt}\mathrm{u}(t)\right) - \mathrm{v}(t) - \mathrm{w}(t),\ -\mathrm{u}(t) + \left(\frac{d}{dt}\mathrm{v}(t)\right) + \mathrm{w}(t)\right]$$

```
> AutonomEq(u(t) + v(t), R, ivar, dvar);
```

$$\left[-\_\mathrm{A}(t) + \left(\frac{d}{dt}\_\mathrm{A}(t)\right)\right]$$

```
> AutonomEq(u(t), R, ivar, dvar);
```

$$[\ ]$$

See Also:

JanetBasis, InvReduce, CompCond, CompCondBasis, SyzOp, Resolution, Torsion, Ext1, Extn, Ind2Diff, Diff2Op.

Janet[BaseChg] - rewrite any expression of the factor module in a new basis

Calling Sequence:

BaseChg(expr,bas,JB);
BaseChg(expr,bas,JB,nbas,"");

Parameters:

`expr` - the expression(s) to be rewritten
`bas` - a basis of the factor module
`JB` - the Janet basis of the factor module
`nbas` - (optional) new names for the basis elements
`""` - (optional) flag to suppress the second component of the output

Description:

- ***BaseChg*** rewrites any expression of the factor module in a given new basis. The input **`expr`** can be given as a list or as a single expression. The names for the elements of the new basis might be given as the optional fourth parameter (**`nbas`**).
- The output consists of two components: the transformed expression and a list with the new basis. In case the input has been a list, this is repeated for any list element.
- If an arbitrary string is assigned as the last argument, only the transformed expression is returned, i.e. the new basis is omitted. (This works only for single expressions, not for input lists.)
- In particular, if the parametric derivatives are chosen as expressions, then ***BaseChg(*****`expr`**, **`bas`**, **`JB`*****)*** performs the same actions as ***ParamBaseChg(*****`bas`**, **`JB`*****)***, but with a different output format (i.e. the parametric derivatives are rewritten in the new basis).
- This function also requires the jets package.

Examples: (a linearised model of the stepmotor):

```
> with(Janet): with(jets):
> ivar := [t];
```

$$ivar := [t]$$

```
> Dvar := [did, diq, dtheta, dvd, dvq];
```

$$Dvar := [did, diq, dtheta, dvd, dvq]$$

Define the module Σ (which describes the linearised model of a stepmotor) :

```
> Sigma :=
  [L*diff(did(t),t)+R*did(t)-Nr*L*diff(theta(t),t)*diq(t)-Nr*L*iq(t)*diff(dtheta(t),t)-dv
  d(t),
  Nr*L*diff(theta(t),t)*did(t)+L*diff(diq(t),t)+R*diq(t)+(Nr*L*id(t)+Km)*diff(dtheta(t),t
  )-dvq(t), -Km*diq(t)+J*diff(dtheta(t),'$'(t,2))+B*diff(dtheta(t),t)];
```

$$\Sigma := \left[L\left(\frac{d}{dt}\,\mathrm{did}(t)\right) + R\,\mathrm{did}(t) - Nr\,L\left(\frac{d}{dt}\,\theta(t)\right)\mathrm{diq}(t) - Nr\,L\,\mathrm{iq}(t)\left(\frac{d}{dt}\,\mathrm{dtheta}(t)\right) - \mathrm{dvd}(t),\right.$$
$$Nr\,L\left(\frac{d}{dt}\,\theta(t)\right)\mathrm{did}(t) + L\left(\frac{d}{dt}\,\mathrm{diq}(t)\right) + R\,\mathrm{diq}(t) + (Nr\,L\,\mathrm{id}(t) + Km)\left(\frac{d}{dt}\,\mathrm{dtheta}(t)\right) - \mathrm{dvq}(t),$$
$$\left. -Km\,\mathrm{diq}(t) + J\left(\frac{d^2}{dt^2}\,\mathrm{dtheta}(t)\right) + B\left(\frac{d}{dt}\,\mathrm{dtheta}(t)\right)\right]$$

```
> u := [dvd(t), dvq(t)];
```

$$u := [\mathrm{dvd}(t), \mathrm{dvq}(t)]$$

Specify the module Σ/[u] and compute its Janet basis:

```
> Lu := [op(Sigma), op(u)];
```

$$Lu := \left[L\left(\frac{d}{dt}\,\mathrm{did}(t)\right) + R\,\mathrm{did}(t) - Nr\,L\left(\frac{d}{dt}\,\theta(t)\right)\mathrm{diq}(t) - Nr\,L\,\mathrm{iq}(t)\left(\frac{d}{dt}\,\mathrm{dtheta}(t)\right) - \mathrm{dvd}(t),\right.$$

$$Nr\,L\left(\frac{d}{dt}\theta(t)\right)\mathrm{did}(t)+L\left(\frac{d}{dt}\mathrm{diq}(t)\right)+R\,\mathrm{diq}(t)+(Nr\,L\,\mathrm{id}(t)+Km)\left(\frac{d}{dt}\mathrm{dtheta}(t)\right)-\mathrm{dvq}(t),$$

$$-Km\,\mathrm{diq}(t)+J\left(\frac{d^2}{dt^2}\mathrm{dtheta}(t)\right)+B\left(\frac{d}{dt}\mathrm{dtheta}(t)\right),\mathrm{dvd}(t),\mathrm{dvq}(t)\Bigg]$$

```
> JLu := JanetBasis(Lu, ivar, Dvar);
```

$$JLu := \left[\left[\left[\mathrm{dvq}(t),\mathrm{dvd}(t),Nr\left(\frac{d}{dt}\theta(t)\right)\mathrm{did}(t)+\frac{R\,\mathrm{diq}(t)}{L}+\left(\frac{d}{dt}\mathrm{diq}(t)\right)+\frac{(Nr\,L\,\mathrm{id}(t)+Km)\left(\frac{d}{dt}\mathrm{dtheta}(t)\right)}{L},\right.\right.\right.$$

$$\left.\left.\left.\frac{R\,\mathrm{did}(t)}{L}+\left(\frac{d}{dt}\mathrm{did}(t)\right)-Nr\left(\frac{d}{dt}\theta(t)\right)\mathrm{diq}(t)-Nr\,\mathrm{iq}(t)\left(\frac{d}{dt}\mathrm{dtheta}(t)\right)-\frac{Km\,\mathrm{diq}(t)}{J}+\frac{B\left(\frac{d}{dt}\mathrm{dtheta}(t)\right)}{J}+\left(\frac{d^2}{dt^2}\mathrm{dtheta}(t)\right)\right],[t],\right.$$

$$[did, diq, dtheta, dvd, dvq]\Bigg]$$

Find a new basis for the factor module:

```
> bas := ParamDeriv(ivar, Dvar);
```

$$bas := \left[\mathrm{dtheta}(t),\frac{d}{dt}\mathrm{dtheta}(t),\mathrm{diq}(t),\mathrm{did}(t)\right]$$

```
> BaseChg(did(t), bas, JLu);
```

$$\_X4(t),\left[\_X1(t)=\mathrm{dtheta}(t),\_X2(t)=\frac{d}{dt}\mathrm{dtheta}(t),\_X3(t)=\mathrm{diq}(t),\_X4(t)=\mathrm{did}(t)\right]$$

Take any other basis of the factor module, for example one of the Brunovsky bases:

```
> nbas1 := [dtheta(t), diff(dtheta(t),t), did(t), diff(did(t),t)];
```

$$nbas1 := \left[\mathrm{dtheta}(t),\frac{d}{dt}\mathrm{dtheta}(t),\mathrm{did}(t),\frac{d}{dt}\mathrm{did}(t)\right]$$

```
> BaseChg(diq(t), nbas1, JLu);
```

$$-\frac{\mathrm{iq}(t)\,\_X2(t)}{\frac{d}{dt}\theta(t)}+\frac{R\,\_X3(t)}{L\,Nr\left(\frac{d}{dt}\theta(t)\right)}+\frac{\_X4(t)}{Nr\left(\frac{d}{dt}\theta(t)\right)},\left[\_X1(t)=\mathrm{dtheta}(t),\_X2(t)=\frac{d}{dt}\mathrm{dtheta}(t),\_X3(t)=\mathrm{did}(t),\_X4(t)=\frac{d}{dt}\mathrm{did}(t)\right]$$

```
> nbas3 := [dtheta(t), diff(dtheta(t),t), diff(dtheta(t),t$2), diff(dtheta(t),t$3)];
```

$$nbas3 := \left[\mathrm{dtheta}(t),\frac{d}{dt}\mathrm{dtheta}(t),\frac{d^2}{dt^2}\mathrm{dtheta}(t),\frac{d^3}{dt^3}\mathrm{dtheta}(t)\right]$$

```
> BaseChg(bas[3], nbas3, JLu);
```

$$\frac{B\,\_X2(t)}{Km}+\frac{J\,\_X3(t)}{Km},\left[\_X1(t)=\mathrm{dtheta}(t),\_X2(t)=\frac{d}{dt}\mathrm{dtheta}(t),\_X3(t)=\frac{d^2}{dt^2}\mathrm{dtheta}(t),\_X4(t)=\frac{d^3}{dt^3}\mathrm{dtheta}(t)\right]$$

```
> BaseChg(bas[3], nbas3, JLu, [a1,a2,a3,a4]);
```

$$\frac{B\,a2}{Km}+\frac{J\,a3}{Km},\left[a1=\mathrm{dtheta}(t),a2=\frac{d}{dt}\mathrm{dtheta}(t),a3=\frac{d^2}{dt^2}\mathrm{dtheta}(t),a4=\frac{d^3}{dt^3}\mathrm{dtheta}(t)\right]$$

```
> BaseChg(bas[3], nbas3, JLu, [a1,a2,a3,a4], "");
```

$$\frac{B\,a2}{Km}+\frac{J\,a3}{Km}$$

```
> BaseChg(bas[3], nbas3, JLu, "");
```

$$\frac{B\,\_X2(t)}{Km}+\frac{J\,\_X3(t)}{Km}$$

```
> BaseChg(bas, nbas3, JLu);
```

$$\left[\_X1(t),\left[\_X1(t)=\mathrm{dtheta}(t),\_X2(t)=\frac{d}{dt}\mathrm{dtheta}(t),\_X3(t)=\frac{d^2}{dt^2}\mathrm{dtheta}(t),\_X4(t)=\frac{d^3}{dt^3}\mathrm{dtheta}(t)\right],\_X2(t),\right.$$

$$\left.\left[\_X1(t)=\mathrm{dtheta}(t),\_X2(t)=\frac{d}{dt}\mathrm{dtheta}(t),\_X3(t)=\frac{d^2}{dt^2}\mathrm{dtheta}(t),\_X4(t)=\frac{d^3}{dt^3}\mathrm{dtheta}(t)\right],\frac{B\,\_X2(t)}{Km}+\frac{J\,\_X3(t)}{Km},\right.$$

$$\left[\_X1(t)=\mathrm{dtheta}(t),\ \_X2(t)=\frac{d}{dt}\mathrm{dtheta}(t),\ \_X3(t)=\frac{d^2}{dt^2}\mathrm{dtheta}(t),\ \_X4(t)=\frac{d^3}{dt^3}\mathrm{dtheta}(t)\right],$$

$$-\frac{(BR+Km^2+Nr\,\mathrm{id}(t)\,Km\,L)\,\_X2(t)}{Km\,L\,Nr\left(\frac{d}{dt}\theta(t)\right)}-\frac{(BL+RJ)\,\_X3(t)}{Km\,L\,Nr\left(\frac{d}{dt}\theta(t)\right)}-\frac{J\,\_X4(t)}{Km\,Nr\left(\frac{d}{dt}\theta(t)\right)},$$

$$\left[\_X1(t)=\mathrm{dtheta}(t),\ \_X2(t)=\frac{d}{dt}\mathrm{dtheta}(t),\ \_X3(t)=\frac{d^2}{dt^2}\mathrm{dtheta}(t),\ \_X4(t)=\frac{d^3}{dt^3}\mathrm{dtheta}(t)\right]\Bigg]$$

```
> BaseChg(bas, nbas3, JLu, "");
```

$$\Bigg[\_X1(t),\left[\_X1(t)=\mathrm{dtheta}(t),\ \_X2(t)=\frac{d}{dt}\mathrm{dtheta}(t),\ \_X3(t)=\frac{d^2}{dt^2}\mathrm{dtheta}(t),\ \_X4(t)=\frac{d^3}{dt^3}\mathrm{dtheta}(t)\right],\ \_X2(t),$$

$$\left[\_X1(t)=\mathrm{dtheta}(t),\ \_X2(t)=\frac{d}{dt}\mathrm{dtheta}(t),\ \_X3(t)=\frac{d^2}{dt^2}\mathrm{dtheta}(t),\ \_X4(t)=\frac{d^3}{dt^3}\mathrm{dtheta}(t)\right],\ \frac{B\,\_X2(t)}{Km}+\frac{J\,\_X3(t)}{Km},$$

$$\left[\_X1(t)=\mathrm{dtheta}(t),\ \_X2(t)=\frac{d}{dt}\mathrm{dtheta}(t),\ \_X3(t)=\frac{d^2}{dt^2}\mathrm{dtheta}(t),\ \_X4(t)=\frac{d^3}{dt^3}\mathrm{dtheta}(t)\right],$$

$$-\frac{(BR+Km^2+Nr\,\mathrm{id}(t)\,Km\,L)\,\_X2(t)}{Km\,L\,Nr\left(\frac{d}{dt}\theta(t)\right)}-\frac{(BL+RJ)\,\_X3(t)}{Km\,L\,Nr\left(\frac{d}{dt}\theta(t)\right)}-\frac{J\,\_X4(t)}{Km\,Nr\left(\frac{d}{dt}\theta(t)\right)},$$

$$\left[\_X1(t)=\mathrm{dtheta}(t),\ \_X2(t)=\frac{d}{dt}\mathrm{dtheta}(t),\ \_X3(t)=\frac{d^2}{dt^2}\mathrm{dtheta}(t),\ \_X4(t)=\frac{d^3}{dt^3}\mathrm{dtheta}(t)\right]\Bigg]$$

```
> ParamBaseChg(nbas3, JLu);
```

$$\Bigg[\mathrm{dtheta}(t)=\_X1(t),\ \frac{d}{dt}\mathrm{dtheta}(t)=\_X2(t),\ \mathrm{diq}(t)=\frac{B\,\_X2(t)}{Km}+\frac{J\,\_X3(t)}{Km},$$

$$\mathrm{did}(t)=-\frac{\_X2(t)\,BR}{Km\,L\,Nr\left(\frac{d}{dt}\theta(t)\right)}-\frac{Km\,\_X2(t)}{L\,Nr\left(\frac{d}{dt}\theta(t)\right)}-\frac{\_X2(t)\,\mathrm{id}(t)}{\frac{d}{dt}\theta(t)}-\frac{\_X3(t)\,B}{Km\,Nr\left(\frac{d}{dt}\theta(t)\right)}-\frac{\_X3(t)\,RJ}{Km\,L\,Nr\left(\frac{d}{dt}\theta(t)\right)}-\frac{J\,\_X4(t)}{Km\,Nr\left(\frac{d}{dt}\theta(t)\right)},$$

$$\left[\_X1(t)=\mathrm{dtheta}(t),\ \_X2(t)=\frac{d}{dt}\mathrm{dtheta}(t),\ \_X3(t)=\frac{d^2}{dt^2}\mathrm{dtheta}(t),\ \_X4(t)=\frac{d^3}{dt^3}\mathrm{dtheta}(t)\right]$$

See Also:

JanetBasis, ParamDeriv, InvReduce, ParamBaseChg, GenCoeff.

Janet[CartanCharacter] - compute Cartan characters of a linear system of PDEs

Calling Sequence:

CartanCharacter(i)
CartanCharacter()

Parameters:

`i` - " " (empty string) or natural number smaller or equal to the number of indeterminates

Description:

- ***CartanCharacter***() prints the highest differentiation order q occurring in the Janet basis and returns the list of Cartan characters [$\alpha(q, 1)$, ..., $\alpha(q, n)$] of the system of PDEs whose involutive basis has been computed last by JanetBasis.
- Note the same information can be extracted from the command HilbertSeries which returns the answer $p(\lambda) + \lambda^q(\alpha(q, 1)/(1-\lambda) + ... + \alpha(q, n)/(1-\lambda)^n)$ for some polynomial $p(\lambda)$ of degree smaller than q.
- Recall, a rough interpretation of $\alpha(q, i)$ is that the general solution depends on $\alpha(q, i)$ arbitrary (analytic) functions in i variables (and $p(1)$ parameters).
- ***CartanCharacter***(" ") simply prints the Cartan characters $\alpha(q, 1)$, ..., $\alpha(q, n)$, where q is the highest differentiation order occurring in the Janet basis, which is an upper bound for the index of regularity, cf. IndexRegularity.
- ***CartanCharacter***(`i`) returns the Cartan characters $\alpha(q, i)$, where q is as above.
- The command only uses the data displayable by TabVar and depends on the monomial ordering chosen for the previous involutive basis computation.

Examples:

```
> with(Janet):
> ivar := [x,y,z,v]; dvar := [u];
```

$$ivar := [x, y, z, v]$$

$$dvar := [u]$$

```
> L := [diff(u(x,y,z,v),x,z)+diff(u(x,y,z,v),y,z)+diff(u(x,y,z,v),x,y),
  diff(u(x,y,z,v),x,y,z)-diff(u(x,y,z,v),v)];
```

$$L := \left[\left(\frac{\partial^2}{\partial z\,\partial x}u(x,y,z,v)\right)+\left(\frac{\partial^2}{\partial z\,\partial y}u(x,y,z,v)\right)+\left(\frac{\partial^2}{\partial y\,\partial x}u(x,y,z,v)\right),\left(\frac{\partial^3}{\partial z\,\partial y\,\partial x}u(x,y,z,v)\right)-\left(\frac{\partial}{\partial v}u(x,y,z,v)\right)\right]$$

```
> JanetBasis(L, ivar, dvar);
```

$$\left[\left[\left(\frac{\partial^2}{\partial z\,\partial x}u(x,y,z,v)\right)+\left(\frac{\partial^2}{\partial z\,\partial y}u(x,y,z,v)\right)+\left(\frac{\partial^2}{\partial y\,\partial x}u(x,y,z,v)\right)\left(\frac{\partial^3}{\partial z^2\,\partial x}u(x,y,z,v)\right)+\left(\frac{\partial^3}{\partial z^2\,\partial y}u(x,y,z,v)\right)+\left(\frac{\partial}{\partial v}u(x,y,z,v)\right),\left(\frac{\partial^4}{\partial z^2\,\partial y^2}u(x,y,z,v)\right)+\left(\frac{\partial^2}{\partial y\,\partial v}u(x,y,z,v)\right)+\left(\frac{\partial^2}{\partial z\,\partial v}u(x,y,z,v)\right)\right],[x,y,z,v],[u]\right]$$

```
> CartanCharacter("");
alpha(4,1) = 15
alpha(4,2) = 6
alpha(4,3) = 0
alpha(4,4) = 0
> CartanCharacter();
Cartan Character for q = 4
```

$$[15, 6, 0, 0]$$

```
> CartanCharacter(2);
```

$$6$$

```
> HilbertSeries();
```

$$1 + 4\,s + 9\,s^2 + 15\,s^3 + s^4\left(15\,\frac{1}{1-s} + 6\,\frac{1}{(1-s)^2}\right)$$

See Also:

JanetBasis, PrincDeriv, ParamDeriv, HilbertSeries, HilbertPolynomial, HP, HilbertFunction, HF, IndexRegularity, PDEBasis, PDEHilbertSeries, PDEHilbertPolynomial, PDEHP, PDEHilbertFunction, PDEHF.

Janet[CmpD] - compose two linear differential operators

Calling Sequence:

CmpD(L1,L2,Dvar,ivar)

Parameters:

L1	-	polynomial or matrix of polynomials representing linear differential operator
L2	-	polynomial or matrix of polynomials representing linear differential operator
Dvar	-	list of indeterminates representing the partial derivative operators
ivar	-	list of independent variables

Description:

- ***CmpD*** forms the composition of the linear differential operators represented by **L1** and **L2** which are (matrices of) polynomials in **Dvar**.
- If **L1** and **L2** are polynomials in **Dvar**, then they represent differential operators which act on one dependent variable. Then, the result of ***CmpD*** is a polynomial in **Dvar** which represents the composition of these operators.
- If **L1** and **L2** are matrices of polynomials in **Dvar**, then the number of columns of **L1** must equal the number of rows of **L2** so that matrix multiplication is defined. The result of ***CmpD*** is a matrix of polynomials in **Dvar** whose number of rows equals the number of rows of **L1** and whose number of columns equals the number of columns of **L2**.
- If one of **L1** and **L2** is a matrix of polynomials in **Dvar** and the other one is a polynomial in **Dvar**, then the composition of the operators is computed analogously to matrix-scalar resp. scalar-matrix multiplication.
- Linear differential operators given in matrix form can be composed by using CmpOp.

Examples:

```
> with(Janet):
```

Example 1: Linear differential operators acting on one dependent variable

```
> Dvar := [Dx,Dy]; ivar := [x,y];
```

$$Dvar := [Dx, Dy]$$

$$ivar := [x, y]$$

```
> L1 := x*Dx+y*Dy;
```

$$L1 := Dx\,x + y\,Dy$$

```
> L2 := x^2*Dy;
```

$$L2 := x^2\,Dy$$

```
> CmpD(L1, L2, Dvar, ivar);
```

$$x^3\,Dy\,Dx + 2\,x^2\,Dy + y\,x^2\,Dy^2$$

```
> CmpD(L2, L1, Dvar, ivar);
```

$$x^3\,Dy\,Dx + y\,x^2\,Dy^2 + x^2\,Dy$$

Example 2: Linear differential operators acting on more than one dependent variable

```
> Dvar := [Dx,Dy,Dz]; ivar := [x,y,z];
```

$$Dvar := [Dx, Dy, Dz]$$

$$ivar := [x, y, z]$$

```
> L1 := matrix([[y*Dy, Dx^2, 0], [0, Dz, 2*x^2*Dy^2]]);
```

$$L1 := \begin{bmatrix} y\,Dy & Dx^2 & 0 \\ 0 & Dz & 2\,x^2\,Dy^2 \end{bmatrix}$$

```
> o1 := D2Op(L1, Dvar, ivar);
```

$$o1 := \begin{bmatrix} [[y, [y]]] & [[1, [x, x]]] & 0 \\ 0 & [[1, [z]]] & [[2\,x^2, [y, y]]] \end{bmatrix}$$

```
> o2 := JAdjoint(o1, ivar);
```

$$o2 := \begin{bmatrix} [[-y, [y]], [-1, [\,]]] & 0 \\ [[1, [x, x]]] & [[-1, [z]]] \\ 0 & [[2\,x^2, [y, y]]] \end{bmatrix}$$

```
> L2 := matrix(Op2D(o2, Dvar, ivar));
```

$$L2 := \begin{bmatrix} -y\,Dy - 1 & 0 \\ Dx^2 & -Dz \\ 0 & 2\,x^2\,Dy^2 \end{bmatrix}$$

```
> CmpD(L1, L2, Dvar, ivar);
```

$$\begin{bmatrix} -y^2\,Dy^2 - 2\,y\,Dy + Dx^4 & -Dx^2\,Dz \\ Dx^2\,Dz & -Dz^2 + 4\,x^4\,Dy^4 \end{bmatrix}$$

```
> matrix(Op2D(CmpOp(o1, o2, ivar), Dvar, ivar));
```

$$\begin{bmatrix} -y^2\,Dy^2 - 2\,y\,Dy + Dx^4 & -Dx^2\,Dz \\ Dx^2\,Dz & -Dz^2 + 4\,x^4\,Dy^4 \end{bmatrix}$$

```
> CmpD(L2, L1, Dvar, ivar);
```

$$\begin{bmatrix} -y^2\,Dy^2 - 2\,y\,Dy & -y\,Dx^2\,Dy - Dx^2 & 0 \\ y\,Dx^2\,Dy & Dx^4 - Dz^2 & -2\,x^2\,Dy^2\,Dz \\ 0 & 2\,x^2\,Dy^2\,Dz & 4\,x^4\,Dy^4 \end{bmatrix}$$

```
> matrix(Op2D(CmpOp(o2, o1, ivar), Dvar, ivar));
```

$$\begin{bmatrix} -y^2\,Dy^2 - 2\,y\,Dy & -y\,Dx^2\,Dy - Dx^2 & 0 \\ y\,Dx^2\,Dy & Dx^4 - Dz^2 & -2\,x^2\,Dy^2\,Dz \\ 0 & 2\,x^2\,Dy^2\,Dz & 4\,x^4\,Dy^4 \end{bmatrix}$$

Composition of a scalar differential operator with **L2** in both ways:

```
> evalm(L2);
```

$$\begin{bmatrix} -y\,Dy - 1 & 0 \\ Dx^2 & -Dz \\ 0 & 2\,x^2\,Dy^2 \end{bmatrix}$$

```
> CmpD(Dx, L2, Dvar, ivar);
```

$$\begin{bmatrix} -Dx\,y\,Dy - Dx & 0 \\ Dx^3 & -Dx\,Dz \\ 0 & 2\,Dx\,x^2\,Dy^2 + 4\,x\,Dy^2 \end{bmatrix}$$

```
> CmpD(L2, Dx, Dvar, ivar);
```

$$\begin{bmatrix} -Dx\,y\,Dy - Dx & 0 \\ Dx^3 & -Dx\,Dz \\ 0 & 2\,Dx\,x^2\,Dy^2 \end{bmatrix}$$

See Also:

Diff2D, D2Diff, Op2D, D2Op, Diff2Op, JAdjoint, AppOp, CmpOp.

Janet[CmpOp] - compose two linear differential operators in matrix form

Calling Sequence:

CmpOp(oper1,oper2,ivar)

Parameters:

`oper1` - linear differential operator in matrix form
`oper2` - linear differential operator in matrix form
`ivar` - list of independent variables

Description:

- ***CmpOp*** forms the product of the linear differential operators given by the matrices **`oper1`** and **`oper2`**, the entries of which are described below.
- The entries of the input and output matrices have the form $[[c_1, [...]], ..., [c_r, [...]]]$, where the dots in the innermost brackets are sequences of independent variables from **`ivar`** representing partial derivatives w. r. t. these variables, and the c_i are the coefficients for these, i. e. functions in **`ivar`**. The whole bracket represents the sum of these items, i. e. the linear combination of the above operators. Note, if an innermost bracket is empty, it represents the identity operator. (For a convenient way to produce such a matrix use Diff2Op as in the example below.)

Examples:

```
> with(Janet):
```

Example 1:

```
> ivar := [x]; dvar := [u];
```

$$ivar := [x]$$

$$dvar := [u]$$

```
> o1 := Diff2Op([diff(u(x), x)], ivar, dvar);
```

$$o1 := [[[1, [x]]]]$$

```
> o2 := Diff2Op([x*diff(u(x), x, x)], ivar, dvar);
```

$$o2 := [[[x, [x, x]]]]$$

```
> CmpOp(o1, o2, ivar, dvar);
```

$$[[[x, [x, x, x]], [1, [x, x]]]]$$

Example 2:

```
> ivar := [x,y,z]; dvar := [u,v,w];
```

$$ivar := [x, y, z]$$

$$dvar := [u, v, w]$$

```
> L := [y*diff(u(x,y,z),y)+diff(v(x,y,z),x$2),
  diff(v(x,y,z),z)+2*x^2*diff(w(x,y,z),y$2)];
```

$$L := \left[y\left(\frac{\partial}{\partial y}\,u(x,y,z)\right)+\left(\frac{\partial^2}{\partial x^2}\,v(x,y,z)\right),\left(\frac{\partial}{\partial z}\,v(x,y,z)\right)+2\,x^2\left(\frac{\partial^2}{\partial y^2}\,w(x,y,z)\right)\right]$$

```
> o1 := Diff2Op(L, ivar, dvar);
```

$$o1 := \begin{bmatrix} [[y, [y]]] & [[1, [x, x]]] & 0 \\ 0 & [[1, [z]]] & [[2\,x^2, [y, y]]] \end{bmatrix}$$

```
> o2 := JAdjoint(o1, ivar);
```

$$o2 := \begin{bmatrix} [[-y, [y]], [-1, [\,]]] & 0 \\ [[1, [x, x]]] & [[-1, [z]]] \\ 0 & [[2x^2, [y, y]]] \end{bmatrix}$$

```
> CmpOp(o1, o2, ivar);
```

$$\begin{bmatrix} [[1, [x, x, x, x]], [-y^2, [y, y]], [-2y, [y]]] & [[-1, [x, x, z]]] \\ [[1, [x, x, z]]] & [[4x^4, [y, y, y, y]], [-1, [z, z]]] \end{bmatrix}$$

```
> CmpOp(o2, o1, ivar);
```

$$\begin{bmatrix} [[-y^2, [y, y]], [-2y, [y]]] & [[-y, [x, x, y]], [-1, [x, x]]] & 0 \\ [[y, [x, x, y]]] & [[1, [x, x, x, x]], [-1, [z, z]]] & [[-2x^2, [y, y, z]]] \\ 0 & [[2x^2, [y, y, z]]] & [[4x^4, [y, y, y, y]]] \end{bmatrix}$$

See Also:

Diff2Op, AppOp, AppOpInd, AddOp, SubOp, JAdjoint, Op2D, D2Op, CmpD.

Janet[CoefficientMatrix] - matrix of coefficients of partial derivatives of a linear differential expression

Calling Sequence:

CoefficientMatrix(U,ivar,dvar,der,uabl,dabl,opt)

Parameters:

`U` - differential expression which is linear in the dependent variables
`ivar` - list of independent variables
`dvar` - list of dependent variables
`der` - list of monomials in the independent variables, or a non-negative integer
`uabl` - (optional) unevaluated name for output list indicating partial derivatives of **`U`**
`dabl` - (optional) unevaluated name for output list indicating partial derivatives of the dependent variables
`opt` - (optional) string or equation specifying options (e.g. "trigonometric", or "columns"="POT")

Description:

- ***CoefficientMatrix*** returns a representation of certain partial derivatives of **`U`** in terms of a matrix. The differential expression **`U`** is assumed to be linear in (analytic) functions which are listed as dependent variables in **`dvar`**. The arguments of these functions may be different from the given list **`ivar`** of independent variables in the sense that these functions may be composed with other known or unknown (analytic) functions of **`ivar`**, arbitrary in number. By assumption, all partial derivatives of **`U`** are again linear in certain partial derivatives of the functions named in **`dvar`**. Each row of the resulting matrix corresponds to a partial derivative of **`U`**, each column corresponds to a partial derivative of a function in **`dvar`** with respect to certain of its arguments. An entry in the resulting matrix is the coefficient of the partial derivative of a function referred to by the column, in the partial derivative of **`U`** referred to by the row. The columns are chosen by ***CoefficientMatrix*** to correspond to the set of partial derivatives of functions in **`dvar`** occurring in the requested derivatives of **`U`**.
- Note that, if some functions listed in **`dvar`** are composed with other functions of **`ivar`**, then the expressions in the latter functions occurring in partial derivatives of **`U`** because of the chain rule will contribute to the entries of the corresponding row (cf. e.g. Example 4 below).
- AnnihilatingSystem is a command that returns a Janet basis for the linear relations that are satisfied by the rows of the resulting coefficient matrix.
- If **`der`** is given as a list, then the partial derivatives of **`U`** with respect to the monomials in **`der`** are formed, and the rows of the resulting matrix correspond to these partial derivatives in the given order. If **`der`** is given as a non-negative integer, then all partial derivatives of **`U`** are formed up to differential order **`der`**-1, and the rows of the resulting matrix correspond to these, sorted with respect to the degree-reverse lexicographical ordering on the monomials in **`ivar`**.
- The columns are always sorted with respect to the degree-reverse lexicographical ordering on the multi-indices representing partial derivatives of the functions in **`dvar`** with respect to their arguments. In order to fix this ordering more precisely, the options "columns"="TOP" resp. "columns"="POT" can be used, see below.
- The optional arguments **`uabl`** and **`dabl`** can used to obtain the lists of partial derivatives corresponding to the rows resp. the columns of the resulting matrix, cf. Examples below. In both cases the argument is expected to be an unevaluated name to which ***CoefficientMatrix*** assigns the corresponding list. The argument **`uabl`** may be given without providing a name for **`dabl`**. However, **`uabl`** must be given if **`dabl`** is present in the list of arguments.
- If an equation with left hand side "columns" is given in **`opt`**, then the right hand side is expected to be either the string "TOP" or the string "POT" (cf. Example 2 below). These are abbreviations of "term over position" resp. "position over term". In the first case (which is the default), the columns are sorted with respect to degrevlex (see above) with priority given to the monomials representing partial derivatives and comparing the respective functions in the order given in **`dvar`** in case the two monomials are equal. In the second case, functions are compared first and degrevlex is used to compare partial derivatives of the same function.
- ***CoefficientMatrix*** applies simplify to the resulting matrix before returning it. If the entries of the result involve sin and cos, then e.g. the square of cos is usually expressed in terms of the square of sin, leading to expressions which are difficult to guess a pattern from. If the string "trigonometric" is given in **`opt`**, then ***CoefficientMatrix*** tries to produce smaller expressions of the resulting entries by

applying the above substitution on the result of simplify (cf. Example 4 below).

- For more information about the coefficient matrix, see W. Plesken, D. Robertz, "Linear differential elimination for analytic functions", preprint, RWTH Aachen University, 2010.

Examples:

```
> with(Janet):
```

Example 1:

```
> ivar := [x,y];
```

$$ivar := [x, y]$$

```
> U := f1(x)*y + f2(y)*x;
```

$$U := \mathrm{f1}(x)\,y + \mathrm{f2}(y)\,x$$

```
> CoefficientMatrix(U, ivar, [f1,f2], [1,y,x]);
```

$$\begin{bmatrix} y & x & 0 & 0 \\ 1 & 0 & 0 & x \\ 0 & 1 & y & 0 \end{bmatrix}$$

```
> CoefficientMatrix(U, ivar, [f1,f2], 3);
```

$$\begin{bmatrix} y & x & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & x & 0 & 0 \\ 0 & 1 & y & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & x \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & y & 0 \end{bmatrix}$$

```
> M := CoefficientMatrix(U, ivar, [f1,f2], 3, 'uabl', 'dabl');
```

$$M := \begin{bmatrix} y & x & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & x & 0 & 0 \\ 0 & 1 & y & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & x \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & y & 0 \end{bmatrix}$$

```
> uabl;
```

$$[1, y, x, y^2, yx, x^2]$$

```
> dabl;
```

$$[\mathrm{f1}(x), \mathrm{f2}(y), \mathrm{D}(f1)(x), \mathrm{D}(f2)(y), (\mathrm{D}^{(2)})(f1)(x), (\mathrm{D}^{(2)})(f2)(y)]$$

```
> add(M[5,i] * dabl[i], i=1..6);
```

$$\mathrm{D}(f1)(x) + \mathrm{D}(f2)(y)$$

```
> convert(diff(U, x, y), D);
```

$$\mathrm{D}(f1)(x) + \mathrm{D}(f2)(y)$$

Example 2:

```
> ivar := [x,y,z];
```

$$ivar := [x, y, z]$$

```
> U := f1(x+y)*z + f2(x,y-z);
```

$$U := \mathrm{f1}(x+y)\,z + \mathrm{f2}(x, y-z)$$

```
> CoefficientMatrix(U, ivar, [f1,f2], 3, 'uabl', 'dabl');
```

$$\begin{bmatrix} z & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & z & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & z & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & z & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & z & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & z & 0 & 0 & 1 \end{bmatrix}$$

```
> uabl;
```

$$[1, z, y, x, z^2, z\,y, z\,x, y^2, y\,x, x^2]$$

```
> dabl;
```

$$[\mathrm{f1}(x+y), \mathrm{f2}(x, y-z), \mathrm{D}(f1)(x+y), \mathrm{D}_2(f2)(x, y-z), \mathrm{D}_1(f2)(x, y-z), (\mathrm{D}^{(2)})(f1)(x+y), \mathrm{D}_{2,2}(f2)(x, y-z), \mathrm{D}_{1,2}(f2)(x, y-z), \mathrm{D}_{1,1}(f2)(x, y-z)]$$

```
> CoefficientMatrix(U, ivar, [f1,f2], 3, 'uabl', 'dabl', "columns"="POT");
```

$$\begin{bmatrix} z & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & z & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & z & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & z & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & z & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & z & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

```
> dabl;
```

$$[\mathrm{f1}(x+y), \mathrm{D}(f1)(x+y), (\mathrm{D}^{(2)})(f1)(x+y), \mathrm{f2}(x, y-z), \mathrm{D}_2(f2)(x, y-z), \mathrm{D}_1(f2)(x, y-z), \mathrm{D}_{2,2}(f2)(x, y-z), \mathrm{D}_{1,2}(f2)(x, y-z), \mathrm{D}_{1,1}(f2)(x, y-z)]$$

Example 3:

```
> ivar := [x,y,z];
```

$$ivar := [x, y, z]$$

```
> U := f1(x+y)*z + f2(x,y-z) + f3(x+z,y+z)*x;
```

$$U := \mathrm{f1}(x+y)\,z + \mathrm{f2}(x, y-z) + \mathrm{f3}(x+z, z+y)\,x$$

```
> CoefficientMatrix(U, ivar, [f1,f2,f3], 3, 'uabl', 'dabl'):
> dabl;
```

$$[\mathrm{f1}(x+y), \mathrm{f2}(x, y-z), \mathrm{f3}(x+z, z+y), \mathrm{D}(f1)(x+y), \mathrm{D}_2(f2)(x, y-z), \mathrm{D}_2(f3)(x+z, z+y), \mathrm{D}_1(f2)(x, y-z), \mathrm{D}_1(f3)(x+z, z+y), (\mathrm{D}^{(2)})(f1)(x+y),$$
$$\mathrm{D}_{2,2}(f2)(x, y-z), \mathrm{D}_{2,2}(f3)(x+z, z+y), \mathrm{D}_{1,2}(f2)(x, y-z), \mathrm{D}_{1,2}(f3)(x+z, z+y), \mathrm{D}_{1,1}(f2)(x, y-z), \mathrm{D}_{1,1}(f3)(x+z, z+y)]$$

```
> CoefficientMatrix(U, ivar, [f1,f2,f3], 3, 'uabl', 'dabl', "columns"="POT"):
> dabl;
```

$$[\mathrm{f1}(x+y), \mathrm{D}(f1)(x+y), (\mathrm{D}^{(2)})(f1)(x+y), \mathrm{f2}(x,y-z), \mathrm{D}_2(f2)(x,y-z), \mathrm{D}_1(f2)(x,y-z), \mathrm{D}_{2,2}(f2)(x,y-z), \mathrm{D}_{1,2}(f2)(x,y-z), \mathrm{D}_{1,1}(f2)(x,y-z),$$
$$\mathrm{f3}(x+z,z+y), \mathrm{D}_2(f3)(x+z,z+y), \mathrm{D}_1(f3)(x+z,z+y), \mathrm{D}_{2,2}(f3)(x+z,z+y), \mathrm{D}_{1,2}(f3)(x+z,z+y), \mathrm{D}_{1,1}(f3)(x+z,z+y)]$$

Example 4: improved representation of trigonometric entries in the coefficient matrix

```
> ivar := [x,y];
```

$$ivar := [x, y]$$

```
> U := f1(sin(x+y)) + f2(cos(x-y));
```

$$U := \mathrm{f1}(\sin(x+y)) + \mathrm{f2}(\cos(x-y))$$

```
> CoefficientMatrix(U, ivar, [f1,f2], 4);
```

$$\begin{bmatrix}
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \cos(x+y) & \sin(x-y) & 0 & 0 & 0 & 0 \\
0 & 0 & \cos(x+y) & -\sin(x-y) & 0 & 0 & 0 & 0 \\
0 & 0 & -\sin(x+y) & -\cos(x-y) & \cos(x+y)^2 & 1-\cos(x-y)^2 & 0 & 0 \\
0 & 0 & -\sin(x+y) & \cos(x-y) & \cos(x+y)^2 & -1+\cos(x-y)^2 & 0 & 0 \\
0 & 0 & -\sin(x+y) & -\cos(x-y) & \cos(x+y)^2 & 1-\cos(x-y)^2 & 0 & 0 \\
0 & 0 & -\cos(x+y) & -\sin(x-y) & -3\cos(x+y)\sin(x+y) & -3\sin(x-y)\cos(x-y) & \cos(x+y)^3 & -\sin(x-y)(-1+\cos(x-y)^2) \\
0 & 0 & -\cos(x+y) & \sin(x-y) & -3\cos(x+y)\sin(x+y) & 3\sin(x-y)\cos(x-y) & \cos(x+y)^3 & \sin(x-y)(-1+\cos(x-y)^2) \\
0 & 0 & -\cos(x+y) & -\sin(x-y) & -3\cos(x+y)\sin(x+y) & -3\sin(x-y)\cos(x-y) & \cos(x+y)^3 & -\sin(x-y)(-1+\cos(x-y)^2) \\
0 & 0 & -\cos(x+y) & \sin(x-y) & -3\cos(x+y)\sin(x+y) & 3\sin(x-y)\cos(x-y) & \cos(x+y)^3 & \sin(x-y)(-1+\cos(x-y)^2)
\end{bmatrix}$$

```
> CoefficientMatrix(U, ivar, [f1,f2], 4, "trigonometric");
```

$$\begin{bmatrix}
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \cos(x+y) & \sin(x-y) & 0 & 0 & 0 & 0 \\
0 & 0 & \cos(x+y) & -\sin(x-y) & 0 & 0 & 0 & 0 \\
0 & 0 & -\sin(x+y) & -\cos(x-y) & \cos(x+y)^2 & \sin(x-y)^2 & 0 & 0 \\
0 & 0 & -\sin(x+y) & \cos(x-y) & \cos(x+y)^2 & -\sin(x-y)^2 & 0 & 0 \\
0 & 0 & -\sin(x+y) & -\cos(x-y) & \cos(x+y)^2 & \sin(x-y)^2 & 0 & 0 \\
0 & 0 & -\cos(x+y) & -\sin(x-y) & -3\cos(x+y)\sin(x+y) & -3\sin(x-y)\cos(x-y) & \cos(x+y)^3 & \sin(x-y)^3 \\
0 & 0 & -\cos(x+y) & \sin(x-y) & -3\cos(x+y)\sin(x+y) & 3\sin(x-y)\cos(x-y) & \cos(x+y)^3 & -\sin(x-y)^3 \\
0 & 0 & -\cos(x+y) & -\sin(x-y) & -3\cos(x+y)\sin(x+y) & -3\sin(x-y)\cos(x-y) & \cos(x+y)^3 & \sin(x-y)^3 \\
0 & 0 & -\cos(x+y) & \sin(x-y) & -3\cos(x+y)\sin(x+y) & 3\sin(x-y)\cos(x-y) & \cos(x+y)^3 & -\sin(x-y)^3
\end{bmatrix}$$

See Also:

JanetBasis, PrincDeriv, ParamDeriv, HilbertSeries, AnnihilatingSystem.

Janet[CompCond] - return compatibility conditions

Calling Sequence:

CompCond(L,ivar,dvar,oivar,ord)

Parameters:

`L`	-	list of linear differential expressions
`ivar`	-	list of independent variables
`dvar`	-	list of dependent variables
`oivar`	-	(optional) list of the independent variables in varied order
`ord`	-	(optional) change of ordering for differential expressions

Description:

- ***CompCond*** is called when one has an affine instead of a linear system. It returns the list of expressions which occurred as right hand sides corresponding to zero left hand side during computation of the last call of JanetBasis or during the reduction of the original equations (in **L**) and non-multiplicative prolongations of the Janet basis elements. Note, **L** has to be the same as in the last call of ***JanetBasis***, i. e. an affine system. (The input is so that the right hand side is not given seperately, but subtracted from the left hand side. If one has the linear system already, one can quickly produce the affine system by using the command AffEqn.)
- The expressions in the output list can be interpreted as necessary conditions to be satisfied by the right hand sides for solvability of the linear inhomogeneous (i. e. affine) system of PDEs with the given left hand side.
- The parameters **`ivar`**, **`dvar`**, **`oivar`**, and **`ord`** have the same meaning as in JanetBasis.
- In terms of modules, ***CompCond*** constructs the syzygies among the generators **L** of the free left module of |**`dvar`**|-tuples over the ring of differential operators modulo the submodule generated by **L**. (Strictly speaking, the ring is the non commutative polynoimial ring in the partial derivatives with respect to **`ivar`** over a field of functions in the independent variables **`ivar`**.) From this point of view the command is completely analogous to the polynomial command Syzygies. In particular, for the constant coefficient case, ***CompCond*** produces the same answers as ***Syzygies***, i. e. the answers ought to be translateable into each other by Pol2Diff and Diff2Pol. For further details on the module point of view cf. Resolution.
- One also has the possibility via **`ivar`** and **`dvar`** to work with other than the standard degrees for the independent and dependent variables, provided one has done that already in JanetBasis.

Examples:

```
> with(Janet):
```

This example takes first general right hand sides $a(x,y)$, $b(x,y)$, $c(x,y)$ and computes the compatibility conditions for them as PDEs.

```
> ivar := [x,y]; dvar := [u];
```

$$ivar := [x, y]$$

$$dvar := [u]$$

```
> l := [diff(u(x,y),x,x), diff(u(x,y),x,y)-diff(u(x,y),x), diff(u(x,y),y,y)];
```

$$l := \left[\frac{\partial^2}{\partial x^2}u(x,y), \left(\frac{\partial^2}{\partial y\,\partial x}u(x,y)\right) - \left(\frac{\partial}{\partial x}u(x,y)\right), \frac{\partial^2}{\partial y^2}u(x,y)\right]$$

```
> L := AffEqn(l, ivar, [a,b,c]);
```

$$L := \left[\left(\frac{\partial^2}{\partial x^2}u(x,y)\right) - a(x,y), \left(\frac{\partial^2}{\partial y\,\partial x}u(x,y)\right) - \left(\frac{\partial}{\partial x}u(x,y)\right) - b(x,y), \left(\frac{\partial^2}{\partial y^2}u(x,y)\right) - c(x,y)\right]$$

```
> JanetBasis(L, ivar, dvar);
```

$$\left[\left[\left(\frac{\partial}{\partial x}u(x,y)\right) + b(x,y) + \left(\frac{\partial}{\partial y}b(x,y)\right) - \left(\frac{\partial}{\partial x}c(x,y)\right), \left(\frac{\partial^2}{\partial y^2}u(x,y)\right) - c(x,y)\right], [x,y], [u]\right]$$

```
> CompCond(L, ivar, dvar);
```

$$\left[-\left(\frac{\partial}{\partial x}c(x,y)\right) - \left(\frac{\partial^2}{\partial y^2}b(x,y)\right) + \left(\frac{\partial^2}{\partial y\,\partial x}c(x,y)\right), -\left(\frac{\partial}{\partial x}b(x,y)\right) - \left(\frac{\partial^2}{\partial y\,\partial x}b(x,y)\right) + \left(\frac{\partial^2}{\partial x^2}c(x,y)\right) - a(x,y),\right.$$

$$-\left(\frac{\partial^2}{\partial y^2}\,b(x,y)\right)-\left(\frac{\partial^3}{\partial y^3}\,b(x,y)\right)+\left(\frac{\partial^3}{\partial y^2\,\partial x}\,c(x,y)\right)-\left(\frac{\partial}{\partial x}\,c(x,y)\right)\Bigg]$$

Now the last example is modified: Instead of general right hand side, explicit ones are taken:

```
> ls := [diff(u(x,y),'$'(x,2)), diff(u(x,y),x,y)-diff(u(x,y),x), diff(u(x,y),'$'(y,2))];
```

$$ls := \left[\frac{\partial^2}{\partial x^2}\,u(x,y),\left(\frac{\partial^2}{\partial y\,\partial x}\,u(x,y)\right)-\left(\frac{\partial}{\partial x}\,u(x,y)\right)\frac{\partial^2}{\partial y^2}\,u(x,y)\right]$$

```
> Ls := AffEqn(ls, ivar, [0,0,x^2]);
```

$$Ls := \left[\frac{\partial^2}{\partial x^2}\,u(x,y),\left(\frac{\partial^2}{\partial y\,\partial x}\,u(x,y)\right)-\left(\frac{\partial}{\partial x}\,u(x,y)\right)\left(\frac{\partial^2}{\partial y^2}\,u(x,y)\right)-x^2\right]$$

```
> Js := JanetBasis(Ls, ivar, dvar);
```

$$Js := \left[\left[\left(\frac{\partial}{\partial x}\,u(x,y)\right)-2\,x,\left(\frac{\partial^2}{\partial y^2}\,u(x,y)\right)-x^2\right],[x,y],[u]\right]$$

```
> HilbertSeries(t);
```

$$1+t$$

```
> CompCond(Ls, ivar, dvar);
```

$$[2,x]$$

This means, there are no solutions.

See Also:

JanetBasis, PrincDeriv, ParamDeriv, CompCondBasis, Resolution, ResolutionDim, EulerChar, SyzOp, HilbertSeries, AffEqn, Syzygies, Pol2Diff, Diff2Pol.

Janet[CompCondBasis] - return minimal Janet basis of compatibility conditions

Calling Sequence:

CompCondBasis(L,ivar,dvar,oivar,ord,rel)

Parameters:

`L`	-	list of linear differential expressions
`ivar`	-	list of independent variables
`dvar`	-	list of dependent variables
`oivar`	-	(optional) list of the independent variables in varied order
`ord`	-	(optional) change of ordering for differential expressions
`rel`	-	(optional) equation "mod" = list of differential expressions

Description:

- ***CompCondBasis*** computes the minimal Janet basis of compatibility conditions for the given system of linear partial differential equations. This system is expected to be homogeneous (hence, in contrast to CompCond, one does not need to form an affine system, e.g. using AffEqn.) The compatibility conditions are given again by left hand sides of partial differential equations, where the new dependent variable $\_c_i$ stands for the i-th equation in **L**.
- If the optional parameter **rel** is present, then compatibility conditions of the system given by **L** are computed modulo the system of linear partial differential equations given by the right hand side of **rel**. Hence the right hand side of **rel** is expected to be a list of differential expressions in the same independent and dependent variables as **L**. For a module-theoretic description cf. SyzygyModule.
- The expressions in the output list can be interpreted as necessary conditions to be satisfied by any right hand sides for solvability of the corresponding linear inhomogeneous (i. e. affine) system of PDEs with the given left hand side.
- The parameters **ivar**, **dvar**, **oivar**, and **ord** and have the same meaning as in JanetBasis.
- The result is of the same format as the result of ***JanetBasis***, i. e. it is a list containing the Janet basis of the compatibility conditions, the list **ivar** of independent variables and the list of dependent variables $\_c_i$ which are used to express the compatibility conditions.
- Internally, the given system of linear partial differential equations is turned into an affine system by subtracting new functions $\_c_i$ from the left hand sides of the system given in **L**. Then a generating set of compatibility conditions is computed using CompCond. Finally, JanetBasis is applied to this generating set.

Examples:

```
> with(Janet):
```

Example 1:

```
> ivar := [x,y]; dvar := [u];
```

$$ivar := [x, y]$$

$$dvar := [u]$$

```
> L := [diff(u(x,y),x,x), diff(u(x,y),x,y)-diff(u(x,y),x), diff(u(x,y),y,y)];
```

$$L := \left[\frac{\partial^2}{\partial x^2}\mathrm{u}(x,y), \left(\frac{\partial^2}{\partial y\,\partial x}\mathrm{u}(x,y)\right) - \left(\frac{\partial}{\partial x}\mathrm{u}(x,y)\right), \frac{\partial^2}{\partial y^2}\mathrm{u}(x,y)\right]$$

```
> CompCondBasis(L, ivar, dvar);
```

$$\left[\left[-\left(\frac{\partial}{\partial y}\_\mathrm{c1}(x,y)\right) + \_\mathrm{c1}(x,y) + \left(\frac{\partial}{\partial x}\_\mathrm{c2}(x,y)\right) - \left(\frac{\partial}{\partial x}\_\mathrm{c3}(x,y)\right) - \left(\frac{\partial^2}{\partial y^2}\_\mathrm{c2}(x,y)\right) + \left(\frac{\partial^2}{\partial y\,\partial x}\_\mathrm{c3}(x,y)\right) - \left(\frac{\partial^2}{\partial y^2}\_\mathrm{c1}(x,y)\right) + \left(\frac{\partial^2}{\partial x^2}\_\mathrm{c3}(x,y)\right)\right], [x, y], [\_c1, \_c2, \_c3]\right]$$

We obtain the same result if we apply ***JanetBasis*** to the result of ***CompCond***:

```
> L2 := AffEqn(L, ivar, [_c1,_c2,_c3]);
```

$$L2 := \left[\left(\frac{\partial^2}{\partial x^2}\mathrm{u}(x,y)\right) - \_c1(x,y), \left(\frac{\partial^2}{\partial y\,\partial x}\mathrm{u}(x,y)\right) - \left(\frac{\partial}{\partial x}\mathrm{u}(x,y)\right) - \_c2(x,y), \left(\frac{\partial^2}{\partial y^2}\mathrm{u}(x,y)\right) - \_c3(x,y)\right]$$

```
> JanetBasis(L2, ivar, dvar):
> C := CompCond(L2, ivar, dvar);
```

$$C := \left[-\left(\frac{\partial}{\partial x}\_c3(x,y)\right) - \left(\frac{\partial^2}{\partial y^2}\_c2(x,y)\right) + \left(\frac{\partial^2}{\partial y\,\partial x}\_c3(x,y)\right), -\left(\frac{\partial}{\partial x}\_c2(x,y)\right) - \left(\frac{\partial^2}{\partial y\,\partial x}\_c2(x,y)\right) + \left(\frac{\partial^2}{\partial x^2}\_c3(x,y)\right) - \_c1(x,y), -\left(\frac{\partial^2}{\partial y^2}\_c2(x,y)\right) - \left(\frac{\partial^3}{\partial y^3}\_c2(x,y)\right) + \left(\frac{\partial^3}{\partial y^2\,\partial x}\_c3(x,y)\right) - \left(\frac{\partial}{\partial x}\_c3(x,y)\right)\right]$$

```
> JanetBasis(C, ivar, [_c1,_c2,_c3]);
```

$$\left[\left[-\left(\frac{\partial}{\partial y}\_c1(x,y)\right) + \_c1(x,y) + \left(\frac{\partial}{\partial x}\_c2(x,y)\right) - \left(\frac{\partial}{\partial x}\_c3(x,y)\right) - \left(\frac{\partial^2}{\partial y^2}\_c2(x,y)\right) + \left(\frac{\partial^2}{\partial y\,\partial x}\_c3(x,y)\right) - \left(\frac{\partial^2}{\partial y^2}\_c1(x,y)\right) + \left(\frac{\partial^2}{\partial x^2}\_c3(x,y)\right)\right], [x,y], [\_c1,\_c2,\_c3]\right]$$

Example 2: Computing compatibility conditions of a linear PDE system modulo another linear PDE system

```
> ivar := [x,y]; dvar := [u];
```

$$ivar := [x,y]$$

$$dvar := [u]$$

```
> L := [diff(u(x,y),x,x), diff(u(x,y),x,y)-diff(u(x,y),x), diff(u(x,y),y,y)];
```

$$L := \left[\frac{\partial^2}{\partial x^2}\mathrm{u}(x,y), \left(\frac{\partial^2}{\partial y\,\partial x}\mathrm{u}(x,y)\right) - \left(\frac{\partial}{\partial x}\mathrm{u}(x,y)\right), \frac{\partial^2}{\partial y^2}\mathrm{u}(x,y)\right]$$

```
> R := [diff(u(x,y),x)];
```

$$R := \left[\frac{\partial}{\partial x}\mathrm{u}(x,y)\right]$$

```
> CompCondBasis(L, ivar, dvar, "mod"=R);
```

$$\left[\left[\_c2(x,y), \_c1(x,y), \frac{\partial}{\partial x}\_c3(x,y)\right], [x,y], [\_c1,\_c2,\_c3]\right]$$

See Also:

JanetBasis, PrincDeriv, ParamDeriv, CompCond, Resolution, ResolutionDim, EulerChar, SyzOp, HilbertSeries, AffEqn, Syzygies, SyzygyModule, Pol2Diff, Diff2Pol.

Janet[D2Diff] - convert linear differential operators into differential expressions

Calling Sequence:

D2Diff(L,Dvar,ivar,dvar)

Parameters:

`L`	-	(list of (lists of)) polynomial(s) representing linear differential operator(s)
`Dvar`	-	list of indeterminates representing the partial derivative operators
`ivar`	-	list of independent variables
`dvar`	-	list of dependent variables

Description:

- ***D2Diff*** applies the linear differential operator which is represented by **`L`** to the list of dependent variables. The resulting (list of) linear differential expression(s) is returned.
- The first argument **`L`** is either a polynomial in **`Dvar`** which represents a linear differential operator acting on differential expressions in one dependent variable, or a list of polynomials in **`Dvar`** representing (depending on the list **`dvar`**) either a list of differential operators of the first kind or one differential operator acting on differential expressions in |**`dvar`**| dependent variables; or **`L`** is a list of lists of polynomials in **`Dvar`** or a matrix of polynomials in **`Dvar`** which again represents one differential operator acting on differential expressions in |**`dvar`**| dependent variables.
- If **`L`** is a list of lists of polynomials in **`Dvar`**, then the lists in **`L`** must be of the same length.
- In general, the result of ***D2Diff*** is a list of linear differential expressions in the dependent variables **`dvar`** and the independent variables **`ivar`**. Only if **`L`** is a polynomial in **`Dvar`** and **`dvar`** contains only one dependent variable, the result is just one linear differential expression.
- Linear differential expressions can be translated back into linear differential operators represented by (lists of (lists of)) polynomial(s) in **`Dvar`** by using Diff2D.

Examples:

```
> with(Janet):
```

Example 1: Examples of linear differential operators acting on differential expressions in one dependent variable:

```
> Dvar := [Dt]; ivar := [t]; dvar := [u];
```

$$Dvar := [Dt]$$

$$ivar := [t]$$

$$dvar := [u]$$

```
> L := Dt^3-Dt+1;
```

$$L := Dt^3 - Dt + 1$$

```
> D2Diff(L, Dvar, ivar, dvar);
```

$$-\left(\frac{d}{dt}\,u(t)\right)+\left(\frac{d^3}{dt^3}\,u(t)\right)+u(t)$$

```
> Dvar := [Ds,Dt]; ivar := [s,t]; dvar := [u];
```

$$Dvar := [Ds, Dt]$$

$$ivar := [s, t]$$

$$dvar := [u]$$

```
> L := [2*Dt+1, Dt^2-Ds];
```

$$L := [2\,Dt + 1, Dt^2 - Ds]$$

```
> D2Diff(L, Dvar, ivar, dvar);
```

$$\left[2\left(\frac{\partial}{\partial t}\,u(s,t)\right)+u(s,t), -\left(\frac{\partial}{\partial s}\,u(s,t)\right)+\left(\frac{\partial^2}{\partial t^2}\,u(s,t)\right)\right]$$

Note, if the number of dependent variables is two, then the meaning of **L** changes:

```
> D2Diff(L, Dvar, ivar, [u,v]);
```

$$\left[2\left(\frac{\partial}{\partial t}u(s,t)\right)+u(s,t)-\left(\frac{\partial}{\partial s}v(s,t)\right)+\left(\frac{\partial^2}{\partial t^2}v(s,t)\right)\right]$$

Example 2: Example of a differential operator acting on more than one dependent variable:

```
> Dvar := [Ds,Dt]; ivar := [s,t]; dvar := [u,v];
```

$$Dvar := [Ds, Dt]$$

$$ivar := [s, t]$$

$$dvar := [u, v]$$

```
> L := [[Ds^2+1, Dt-2], [Ds*Dt+1, Dt^3]];
```

$$L := [[Ds^2+1, Dt-2], [Ds\,Dt+1, Dt^3]]$$

```
> D2Diff(L, Dvar, ivar, dvar);
```

$$\left[\left(\frac{\partial^2}{\partial s^2}u(s,t)\right)+u(s,t)+\left(\frac{\partial}{\partial t}v(s,t)\right)-2\,v(s,t),\left(\frac{\partial^2}{\partial t\,\partial s}u(s,t)\right)+u(s,t)+\left(\frac{\partial^3}{\partial t^3}v(s,t)\right)\right]$$

The first argument may be a matrix instead of a list of lists of polynomials in **Dvar**:

```
> L := matrix(2, 2, [[Ds^2+1, Dt-2], [Ds*Dt+1, Dt^3]]);
```

$$L := \begin{bmatrix} Ds^2+1 & Dt-2 \\ Ds\,Dt+1 & Dt^3 \end{bmatrix}$$

```
> D2Diff(L, Dvar, ivar, [u,v]);
```

$$\left[\left(\frac{\partial^2}{\partial s^2}u(s,t)\right)+u(s,t)+\left(\frac{\partial}{\partial t}v(s,t)\right)-2\,v(s,t),\left(\frac{\partial^2}{\partial t\,\partial s}u(s,t)\right)+u(s,t)+\left(\frac{\partial^3}{\partial t^3}v(s,t)\right)\right]$$

See Also:

Diff2D, Op2D, D2Op, CmpD, Diff2Pol, Pol2Diff, Diff2Ind, Ind2Diff, AppOpInd, Pol2Ind, Diff2Op, AppOp, Pol2Op.

Janet[D2Op] - convert polynomial of differential operators into linear differential operator in matrix form

Calling Sequence:

D2Op(L,Dvar,ivar)

Parameters:

`L` - (list of (lists of)) polynomial(s) representing linear differential operator(s)
`Dvar` - list of indeterminates representing the partial derivative operators
`ivar` - list of independent variables

Description:

- ***D2Op*** translates **L** which is a (list of (lists of)) polynomials in **Dvar**, into the matrix which represents the same linear differential operator as **L**.
- If **L** is a list of lists of polynomials in **Dvar**, then the lists in **L** must be of the same length.
- The entries of the output matrix have the form $[[c_1, [...]], ..., [c_r, [...]]]$, where the dots in the innermost brackets are sequences of independent variables from **ivar** representing partial derivatives w. r. t. these variables, and the c_i are the coefficients for these, i. e. functions in **ivar**. The whole bracket represents the sum of these items, i. e. the linear combination of the above operators. Note, if an innermost bracket is empty, it represents the identity operator.
- If **L** is a polynomial in **Dvar**, then **L** represents a differential operator which acts on one dependent variable. Then, the result of ***D2Op*** is a single entry of a differential operator in matrix form as described in the preceding paragraph.
- If **L** is a list of polynomials in **Dvar**, then **L** represents a column of differential operators acting on one dependent variable. Then, the result of ***D2Op*** is a matrix with one column and as many rows as there are entries in **L**.
- If **L** is a list of lists of polynomials in **Dvar**, then the common length of the lists is the number of dependent variables on which the differential operator represented by **L** acts. The number of columns of the resulting matrix equals this number; the number rows of the result equals the number of lists in **L**.
- Linear differential operators in matrix form can be converted to (lists of lists of) polynomial(s) in **Dvar**, where the *i*-th indeterminate in **Dvar** represents differentiation with respect to the *i*-th independent variable, by using Op2D.

Examples:

```
> with(Janet):
```

Example 1: Examples of linear differential operators acting on differential expressions in one dependent variable:

Ordinary differential operators:

```
> Dvar := [Dt]; ivar := [t];
```

$$Dvar := [Dt]$$

$$ivar := [t]$$

```
> L := Dt^3-Dt+1;
```

$$L := Dt^3 - Dt + 1$$

```
> D2Op(L, Dvar, ivar);
```

$$[[1, [t, t, t]], [-1, [t]], [1, [\,]]]$$

Partial differential operators:

```
> Dvar := [Ds,Dt]; ivar := [s,t];
```

$$Dvar := [Ds, Dt]$$

$$ivar := [s, t]$$

A list of polynomials in **Dvar** is interpreted as a column of differential operators acting on one dependent variable:

```
> L := [2*Dt+1, Dt^2-Ds];
```

$$L := [2\,Dt + 1, Dt^2 - Ds]$$

```
> D2Op(L, Dvar, ivar);
```

$$\begin{bmatrix} [[2, [t]], [1, [\ ]]] \\ [[1, [t, t]], [-1, [s]]] \end{bmatrix}$$

The same applies to a list of lists of polynomials in **Dvar** consisting of one entry only:

```
> L := [[2*Dt+1], [Dt^2-Ds]];
```

$$L := [[2\,Dt + 1], [Dt^2 - Ds]]$$

```
> D2Op(L, Dvar, ivar);
```

$$\begin{bmatrix} [[2, [t]], [1, [\ ]]] \\ [[1, [t, t]], [-1, [s]]] \end{bmatrix}$$

However, a list of lists of polynomials in **Dvar** which consist of two entries represents a differential operator acting on two dependent variables:

```
> L := [2*Dt+1, Dt^2-Ds];
```

$$L := [2\,Dt + 1, Dt^2 - Ds]$$

```
> D2Op([L], Dvar, ivar);
```

$$[[[2, [t]], [1, [\ ]]] \quad [[1, [t, t]], [-1, [s]]]]$$

Example 2: Examples of linear differential operator acting on more than one dependent variable:

```
> Dvar := [Ds,Dt]; ivar := [s,t];
```

$$Dvar := [Ds, Dt]$$

$$ivar := [s, t]$$

```
> L := [[Ds^2+1, Dt-2], [Ds*Dt+1, Dt^3]];
```

$$L := [[Ds^2 + 1, Dt - 2], [Ds\,Dt + 1, Dt^3]]$$

```
> D2Op(L, Dvar, ivar);
```

$$\begin{bmatrix} [[1, [s, s]], [1, [\ ]]] & [[1, [t]], [-2, [\ ]]] \\ [[1, [s, t]], [1, [\ ]]] & [[1, [t, t, t]]] \end{bmatrix}$$

The first argument may be a matrix instead of a list of lists of polynomials in **Dvar**:

```
> L := matrix(2, 2, [[Ds^2+1, Dt-2], [Ds*Dt+1, Dt^3]]);
```

$$L := \begin{bmatrix} Ds^2 + 1 & Dt - 2 \\ Ds\,Dt + 1 & Dt^3 \end{bmatrix}$$

```
> D2Op(L, Dvar, ivar);
```

$$\begin{bmatrix} [[1, [s, s]], [1, [\ ]]] & [[1, [t]], [-2, [\ ]]] \\ [[1, [s, t]], [1, [\ ]]] & [[1, [t, t, t]]] \end{bmatrix}$$

See Also:

Diff2D, D2Diff, Op2D, CmpD, Diff2Pol, Pol2Diff, Diff2Ind, Ind2Diff, AppOpInd, Pol2Ind, Diff2Op, AppOp, Pol2Op.

Janet[Diff2D] - convert differential expression into linear differential operator

Calling Sequence:

Diff2D(L,Dvar,ivar,dvar)

Parameters:

`L`	-	differential expression or list of such
`Dvar`	-	list of indeterminates representing the partial derivative operators
`ivar`	-	list of independent variables
`dvar`	-	list of dependent variables

Description:

- ***Diff2D*** returns the linear differential operator, which yields **`L`** if it is applied to **`dvar`**, as a (list of (lists of)) polynomial(s) in **`Dvar`**, where the *i*-th indeterminate in **`Dvar`** represents differentiation with respect to the *i*-th independent variable.
- If **`L`** consists of affine differential expressions, then the affine part is ignored.
- If **`dvar`** contains only one dependent variable, then the differential expressions in **`L`** are translated into polynomials in **`Dvar`**. If more dependent variables are given in **`dvar`**, then the result consists of lists of polynomials in **`Dvar`**, the *i*-th entry of which corresponds to the *i*-th dependent variable. In general, the result of ***Diff2D*** is a list of polynomials or a list of lists of polynomials as described before. Only if **`L`** is a differential expression and **`dvar`** contains only one dependent variable, the result is just one polynomial in **`Dvar`**.
- (Lists of (lists of)) polynomials in **`Dvar`** are translated back into differential expressions by using D2Diff.

Examples:

```
> with(Janet):
```

Converting one differential expression:

```
> Dvar := [Dx]; ivar := [x]; dvar := [u];
```

$$Dvar := [Dx]$$

$$ivar := [x]$$

$$dvar := [u]$$

```
> L := diff(u(x), x$3)-2*diff(u(x), x)+u(x);
```

$$L := \left(\frac{d^3}{dx^3}\,\mathrm{u}(x)\right) - 2\left(\frac{d}{dx}\,\mathrm{u}(x)\right) + \mathrm{u}(x)$$

```
> Diff2D(L, Dvar, ivar, dvar);
```

$$Dx^3 - 2\,Dx + 1$$

Converting one differential expression in several dependent variables:

```
> Dvar := [Dx]; ivar := [x]; dvar := [u,v];
```

$$Dvar := [Dx]$$

$$ivar := [x]$$

$$dvar := [u, v]$$

```
> L := diff(u(x), x$3)-2*diff(v(x), x)+u(x);
```

$$L := \left(\frac{d^3}{dx^3}\,\mathrm{u}(x)\right) - 2\left(\frac{d}{dx}\,\mathrm{v}(x)\right) + \mathrm{u}(x)$$

```
> Diff2D(L, Dvar, ivar, dvar);
```

$$[[Dx^3 + 1, -2\,Dx]]$$

Converting a list of differential expressions in one dependent variable:

```
> Dvar := [Dx,Dy,Dz]; ivar := [x,y,z]; dvar := [u];
```

$$Dvar := [Dx, Dy, Dz]$$

$$ivar := [x, y, z]$$

$$dvar := [u]$$

```
> L := [diff(u(x,y,z),x,y)-2*diff(u(x,y,z),x,z)+u(x),
  diff(u(x,y,z),x$3)-2*diff(u(x,y,z),y,y)];
```

$$L := \left[\left(\frac{\partial^2}{\partial y\,\partial x}\,u(x,y,z)\right) - 2\left(\frac{\partial^2}{\partial z\,\partial x}\,u(x,y,z)\right) + u(x),\ \left(\frac{\partial^3}{\partial x^3}\,u(x,y,z)\right) - 2\left(\frac{\partial^2}{\partial y^2}\,u(x,y,z)\right)\right]$$

```
> Diff2D(L, Dvar, ivar, dvar);
```

$$[Dx\,Dy - 2\,Dx\,Dz,\ Dx^3 - 2\,Dy^2]$$

Converting a list of differential expressions in several dependent variables:

```
> Dvar := [Dx,Dy,Dz]; ivar := [x,y,z]; dvar := [u,v,w];
```

$$Dvar := [Dx, Dy, Dz]$$

$$ivar := [x, y, z]$$

$$dvar := [u, v, w]$$

```
> L := [3*y*diff(u(x,y,z),y)+exp(x+z)*diff(v(x,y,z),'$'(x,2)),
  diff(v(x,y,z),z)+sin(x)*diff(w(x,y,z),'$'(y,2))];
```

$$L := \left[3\,y\left(\frac{\partial}{\partial y}\,u(x,y,z)\right) + \mathbf{e}^{(x+z)}\left(\frac{\partial^2}{\partial x^2}\,v(x,y,z)\right),\ \left(\frac{\partial}{\partial z}\,v(x,y,z)\right) + \sin(x)\left(\frac{\partial^2}{\partial y^2}\,w(x,y,z)\right)\right]$$

```
> Diff2D(L, Dvar, ivar, dvar);
```

$$[[3\,y\,Dy,\ \mathbf{e}^{(x+z)}\,Dx^2,\ 0],\ [0,\ Dz,\ \sin(x)\,Dy^2]]$$

See Also:

D2Diff, Op2D, D2Op, CmpD, Diff2Pol, Pol2Diff, Diff2Ind, Ind2Diff, AppOpInd, Pol2Ind, Diff2Op, AppOp, Pol2Op.

Janet[Diff2Ind] - convert differential expression from Maple to jet notation

Calling Sequence:

Diff2Ind(L,ivar,dvar)

Parameters:

`L` - differential expression or list of differential expressions (in usual Maple notation)
`ivar` - list of independent variables
`dvar` - list of dependent variables

Description:

- ***Diff2Ind*** translates the differential expressions in **L** (given in the usual Maple notation) to jet notation.
- The jet notation is adapted from the Maple package jets. It is much more compact than the usual Maple notation for differential expressions. To fix ideas, let *u* be a dependent variable and *x,y,z* the independent variables. Then the jet notation for $\frac{\partial^4}{\partial z\,\partial y\,\partial y\,\partial x}\mathrm{u}(x,y,z)$ is $u_{x,y,y,z}$.
- The command Ind2Diff is inverse to ***Diff2Ind***.

Examples:

```
> with(Janet):
> Diff2Ind(diff(u(x,y,z), x$2), [x,y,z], [u]);
```

$$u_{x,x}$$

```
> ivar := [x,y,z]; dvar := [u,v];
```

$$ivar := [x, y, z]$$

$$dvar := [u, v]$$

```
> L := [diff(u(x,y,z),y)+diff(v(x,y,z),x$2), x*diff(u(x,y,z),y$2)+diff(v(x,y,z),z)];
```

$$L := \left[\left(\frac{\partial}{\partial y}\mathrm{u}(x,y,z)\right)+\left(\frac{\partial^2}{\partial x^2}\mathrm{v}(x,y,z)\right), x\left(\frac{\partial^2}{\partial y^2}\mathrm{u}(x,y,z)\right)+\left(\frac{\partial}{\partial z}\mathrm{v}(x,y,z)\right)\right]$$

```
> Lin := Diff2Ind(L, ivar, dvar);
```

$$Lin := [u_y + v_{x,x}, x\,u_{y,y} + v_z]$$

```
> Ind2Diff(Lin, ivar, dvar);
```

$$\left[\left(\frac{\partial}{\partial y}\mathrm{u}(x,y,z)\right)+\left(\frac{\partial^2}{\partial x^2}\mathrm{v}(x,y,z)\right), x\left(\frac{\partial^2}{\partial y^2}\mathrm{u}(x,y,z)\right)+\left(\frac{\partial}{\partial z}\mathrm{v}(x,y,z)\right)\right]$$

See Also:

Pol2Ind, Ind2Diff, AppOpInd, Diff2Op, Pol2Op, Diff2Pol, Pol2Diff, Op2D, D2Op, Diff2D, D2Diff.

Janet[Diff2Op] - turn differential expression into linear differential operator in matrix form

Calling Sequence:

Diff2Op(L,ivar,dvar)

Parameters:

`L`	-	differential expression or list of such
`ivar`	-	list of independent variables
`dvar`	-	list of dependent variables

Description:

- ***Diff2Op*** produces the linear differential operator in matrix form, which applied to **`dvar`** yields **`L`** again, cf. AppOp.
- If **`L`** consists of affine expressions, then the affine part is ignored.
- The entries of the output matrix have the form [[c_1, [...]], ..., [c_r, [...]]], where the dots in the innermost brackets are sequences of independent variables from **`ivar`** representing partial derivatives w. r. t. these variables, and the c_i are the coefficients for these, i. e. functions in **`ivar`**. The whole bracket represents the sum of these items, i. e. the linear combination of the above operators. Note, if an innermost bracket is empty, it represents the identity operator.

Examples:

```
> with(Janet):
> ivar := [x,y,z]; dvar := [u,v,w];
```

$$ivar := [x, y, z]$$

$$dvar := [u, v, w]$$

```
> L := [3*y*diff(u(x,y,z),y)+exp(x+z)*diff(v(x,y,z),'$'(x,2)),
  diff(v(x,y,z),z)+sin(x)*diff(w(x,y,z),'$'(y,2))];
```

$$L := \left[3\,y\left(\frac{\partial}{\partial y}\,\mathrm{u}(x,y,z)\right)+\mathbf{e}^{(x+z)}\left(\frac{\partial^2}{\partial x^2}\,\mathrm{v}(x,y,z)\right),\left(\frac{\partial}{\partial z}\,\mathrm{v}(x,y,z)\right)+\sin(x)\left(\frac{\partial^2}{\partial y^2}\,\mathrm{w}(x,y,z)\right)\right]$$

```
> Diff2Op(L, ivar, dvar);
```

$$\begin{bmatrix} [[3\,y, [y]]] & [[\mathbf{e}^{(x+z)}, [x, x]]] & 0 \\ 0 & [[1, [z]]] & [[\sin(x), [y, y]]] \end{bmatrix}$$

```
> Diff2Op(3*y*diff(u(x,y,z),y)+exp(x+z)*diff(v(x,y,z),x),ivar,dvar);
```

$$[[[3\,y, [y]]] \quad [[\mathbf{e}^{(x+z)}, [x]]] \quad 0]$$

See Also:

CmpOp, JAdjoint, AppOp, Pol2Op, Diff2Ind, Ind2Diff, AppOpInd, Pol2Ind, Diff2Pol, Pol2Diff, Op2D, D2Op, Diff2D, D2Diff.

Janet[Diff2Pol] - return polynomial expression corresponding to a linear differential expression (with constant coefficients)

Calling Sequence:

Diff2Pol(L,ivar,dvar)

Parameters:

L	-	differential expression with constant coefficients or list of such
ivar	-	list of independent variables (which become the indeterminates of the polynomial ring)
dvar	-	list of dependent variables

Description:

- ***Diff2Pol*** is the command inverse to Pol2Diff .

Examples:

```
> with(Janet):
> ivar := [x,y,z]; dvar := [u];
```

$$ivar := [x, y, z]$$

$$dvar := [u]$$

Translating a differential expression:

```
> L := diff(u(x,y,z), x);
```

$$L := \frac{\partial}{\partial x} u(x, y, z)$$

```
> Diff2Pol(L, ivar, dvar);
```

$$x$$

Translating a list of differential expressions:

```
> L := [diff(u(x,y,z),x$2) - 3*u(x,y,z), diff(u(x,y,z),y,z$2)+diff(u(x,y,z),x)];
```

$$L := \left[\left(\frac{\partial^2}{\partial x^2} u(x, y, z)\right) - 3\, u(x, y, z), \left(\frac{\partial^3}{\partial z^2\, \partial y} u(x, y, z)\right) + \left(\frac{\partial}{\partial x} u(x, y, z)\right)\right]$$

```
> Diff2Pol(L, ivar, dvar);
```

$$[x^2 - 3, yz^2 + x]$$

Translating a list of affine differential expressions:

```
> L := [diff(u(x,y,z),x$2) - 3*u(x,y,z) - a(x,y,z), diff(u(x,y,z),y,z$2)+diff(u(x,y,z),x)
  - b(x,y,z)];
```

$$L := \left[\left(\frac{\partial^2}{\partial x^2} u(x, y, z)\right) - 3\, u(x, y, z) - a(x, y, z), \left(\frac{\partial^3}{\partial z^2\, \partial y} u(x, y, z)\right) + \left(\frac{\partial}{\partial x} u(x, y, z)\right) - b(x, y, z)\right]$$

```
> JB := JanetBasis(L, ivar, dvar);
```

$$JB := \left[\left[\left(\frac{\partial^2}{\partial x^2} u(x, y, z)\right) - 3\, u(x, y, z) - a(x, y, z), \left(\frac{\partial^3}{\partial z^2\, \partial y} u(x, y, z)\right) + \left(\frac{\partial}{\partial x} u(x, y, z)\right) - b(x, y, z),\right.\right.$$
$$\left.\left. 3\, u(x, y, z) + \left(\frac{\partial^4}{\partial z^2\, \partial y\, \partial x} u(x, y, z)\right) + a(x, y, z) - \left(\frac{\partial}{\partial x} b(x, y, z)\right)\right], [x, y, z], [u]\right]$$

```
> Diff2Pol(JB[1], ivar, dvar);
```

$$[x^2 - 3 = [1, 0], yz^2 + x = [0, 1], xyz^2 + 3 = [-1, x]]$$

```
> Diff2Pol(x*diff(u(x,y,z),x$2), ivar, dvar);
Error, (in Janet/Diff2Pol) expecting differential expressions with constant coefficients.
```

See Also:

InvolutiveBasis, JanetBasis, Pol2Diff, Pol2Op, Diff2Op, AppOp, Diff2Ind, Ind2Diff, AppOpInd, Op2D, D2Op, Diff2D, D2Diff

Janet[DiffGroebnerBasis] - return minimal Groebner basis for a system of linear partial differential equations

Calling Sequence:

DiffGroebnerBasis(L,ivar,dvar,oivar,ord)

Parameters:

`L` - list of linear differential expressions
`ivar` - list of independent variables
`dvar` - list of dependent variables
`oivar` - (optional) list of the independent variables in varied order
`ord` - (optional) change of ordering for differential expressions

Description:

- ***DiffGroebnerBasis*** returns the minimal Groebner basis for a system of linear partial differential equations with respect to a certain ordering. The default ordering is degree reverse lexicographical.
- The Groebner basis is extracted from the involutive basis for the given system of linear partial differential equations (which is, in general, a redundant Groebner basis because of the separation of the independent variables into multiplicative and non-multiplicative ones). Hence, ***DiffGroebnerBasis*** passes its arguments to JanetBasis and extracts the minimal Groebner basis from this result. Therefore, Janet's data (see PrincDeriv) contains information about the involutive basis for the given PDE system.
- All parameters to ***DiffGroebnerBasis*** have the same meaning as in JanetBasis.
- The output is a list containing the Groebner basis for the system of linear partial differential equations given by **L** with respect to the chosen ordering.

Examples:

```
> with(Janet):
```

Example: Janet's example

```
> ivar := [x,y,t]: dvar := [u]:
> L := [diff(u(x,y,t),x$2) - y*diff(u(x,y,t),t$2), diff(u(x,y,t),y$2)];
```

$$L := \left[\left(\frac{\partial^2}{\partial x^2}\,u(x,y,t)\right) - y\left(\frac{\partial^2}{\partial t^2}\,u(x,y,t)\right), \frac{\partial^2}{\partial y^2}\,u(x,y,t)\right]$$

```
> DiffGroebnerBasis(L, ivar, dvar);
```

$$\left[\left[\frac{\partial^2}{\partial y^2}\,u(x,y,t), \left(\frac{\partial^2}{\partial x^2}\,u(x,y,t)\right) - y\left(\frac{\partial^2}{\partial t^2}\,u(x,y,t)\right), \frac{\partial^3}{\partial y\,\partial t^2}\,u(x,y,t), \frac{\partial^4}{\partial t^4}\,u(x,y,t)\right], [x,y,t], [u]\right]$$

```
> PrincDeriv();
```

$$\left[\frac{\partial^2}{\partial y^2}\,u(x,y,t), [*,y,t], \frac{\partial^2}{\partial y^2}\,u(x,y,t)\right]$$

$$\left[\left(\frac{\partial^2}{\partial x^2}\,u(x,y,t)\right) - y\left(\frac{\partial^2}{\partial t^2}\,u(x,y,t)\right), [x,y,t], \frac{\partial^2}{\partial x^2}\,u(x,y,t)\right]$$

$$\left[\frac{\partial^3}{\partial y\,\partial t^2}\,u(x,y,t), [*,*,t], \frac{\partial^3}{\partial y\,\partial t^2}\,u(x,y,t)\right]$$

$$\left[\frac{\partial^3}{\partial y^2\,\partial x}\,u(x,y,t), [*,y,t], \frac{\partial^3}{\partial y^2\,\partial x}\,u(x,y,t)\right]$$

$$\left[\frac{\partial^4}{\partial t^4}\,u(x,y,t), [*,*,t], \frac{\partial^4}{\partial t^4}\,u(x,y,t)\right]$$

$$\left[\frac{\partial^4}{\partial y\,\partial x\,\partial t^2}\,u(x,y,t), [*,*,t], \frac{\partial^4}{\partial y\,\partial x\,\partial t^2}\,u(x,y,t)\right]$$

$$\left[\frac{\partial^5}{\partial x \partial t^4} u(x, y, t), [*, *, t], \frac{\partial^5}{\partial x \partial t^4} u(x, y, t)\right]$$

See Also:

JanetBasis, PrincDeriv, JanetOptions, LeadingDeriv.

Janet[GenCoeff] - rewrite module expression in a new basis

Calling Sequence:

GenCoeff(expr,u,bas,JB);
GenCoeff(expr,u,bas,JB,uu,nbas,"");

Parameters:

`expr` - the expression(s) to be rewritten
`u` - the basis of the submodule
`bas` - the basis of the factor module
`JB` - Janet basis of the module
`uu` - (optional) names for the elements from **u**
`nbas` - (optional) names for the basis elements in **bas**
`""` - (optional)

Description:

- The output consists of three lists: transformated expressions, names of the submodule generators, and names of the factor module generators.
- The input **expr** can be given as a list or as a single expression.
- For technical reasons, the basis change is computed in the decomposition of the module into a free submodule and a torsion submodule $\Sigma = [u] + \Sigma/[u]$. Thus, the bases **u** and **bas** have to be given separately.
- If the optional argument **uu** is replaced by "", or as the last argument, is omitted, then the basis elements in **u** are denoted by _U1, _U2,...
- If the optional argument **nbas** is left out then the basis elements in **bas** are denoted by _X1, _X2,...
- If ***GenCoeff*** is used with the last optional argument "", it returns only the list of expressions in the new basis.

Examples:

```
> with(Janet):
> ivar := [t];
```

$$ivar := [t]$$

```
> Dvar := [did, diq, dtheta, dvd, dvq];
```

$$Dvar := [did, diq, dtheta, dvd, dvq]$$

Define the module Σ (which describes the linearised model of a stepmotor) :

```
> Sigma :=
  [L*diff(did(t),t)+R*did(t)-Nr*L*diff(theta(t),t)*diq(t)-Nr*L*iq(t)*diff(dtheta(t),t)-dv
  d(t),
  Nr*L*diff(theta(t),t)*did(t)+L*diff(diq(t),t)+R*diq(t)+(Nr*L*id(t)+Km)*diff(dtheta(t),t
  )-dvq(t), -Km*diq(t)+J*diff(dtheta(t),'$'(t,2))+B*diff(dtheta(t),t)];
```

$$\Sigma := \left[L\left(\frac{d}{dt}\,\mathrm{did}(t)\right) + R\,\mathrm{did}(t) - Nr\,L\left(\frac{d}{dt}\,\theta(t)\right)\mathrm{diq}(t) - Nr\,L\,\mathrm{iq}(t)\left(\frac{d}{dt}\,\mathrm{dtheta}(t)\right) - \mathrm{dvd}(t),\right.$$
$$Nr\,L\left(\frac{d}{dt}\,\theta(t)\right)\mathrm{did}(t) + L\left(\frac{d}{dt}\,\mathrm{diq}(t)\right) + R\,\mathrm{diq}(t) + (Nr\,L\,\mathrm{id}(t) + Km)\left(\frac{d}{dt}\,\mathrm{dtheta}(t)\right) - \mathrm{dvq}(t),$$
$$\left. -Km\,\mathrm{diq}(t) + J\left(\frac{d^2}{dt^2}\,\mathrm{dtheta}(t)\right) + B\left(\frac{d}{dt}\,\mathrm{dtheta}(t)\right)\right]$$

```
> JSigma := JanetBasis(Sigma, ivar, Dvar):
```

We define a free submodule of Σ:

```
> u := Ind2Diff([dvd,dvq], ivar, Dvar);
```

$$u := [\mathrm{dvd}(t), \mathrm{dvq}(t)]$$

```
> Lu := [op(Sigma),op(u)];
```

$$Lu := \left[L\left(\frac{d}{dt}\mathrm{did}(t)\right) + R\,\mathrm{did}(t) - Nr\,L\left(\frac{d}{dt}\theta(t)\right)\mathrm{diq}(t) - Nr\,L\,\mathrm{iq}(t)\left(\frac{d}{dt}\mathrm{dtheta}(t)\right) - \mathrm{dvd}(t),\right.$$
$$Nr\,L\left(\frac{d}{dt}\theta(t)\right)\mathrm{did}(t) + L\left(\frac{d}{dt}\mathrm{diq}(t)\right) + R\,\mathrm{diq}(t) + (Nr\,L\,\mathrm{id}(t) + Km)\left(\frac{d}{dt}\mathrm{dtheta}(t)\right) - \mathrm{dvq}(t),$$
$$\left. -Km\,\mathrm{diq}(t) + J\left(\frac{d^2}{dt^2}\mathrm{dtheta}(t)\right) + B\left(\frac{d}{dt}\mathrm{dtheta}(t)\right), \mathrm{dvd}(t), \mathrm{dvq}(t)\right]$$

Verify that $\Sigma/[u]$ is a torsion module (i.e. the Hilbert series is finite):

```
> JLu := JanetBasis(Lu,ivar,Dvar); HilbertSeries(s); eval(%,s=1);
```

$$JLu := \left[\left[\left[\mathrm{dvq}(t), \mathrm{dvd}(t), Nr\left(\frac{d}{dt}\theta(t)\right)\mathrm{did}(t) + \frac{R\,\mathrm{diq}(t)}{L} + \left(\frac{d}{dt}\mathrm{diq}(t)\right) + \frac{(Nr\,L\,\mathrm{id}(t) + Km)\left(\frac{d}{dt}\mathrm{dtheta}(t)\right)}{L},\right.\right.\right.$$
$$\left.\left.\left.\frac{R\,\mathrm{did}(t)}{L} + \left(\frac{d}{dt}\mathrm{did}(t)\right) - Nr\left(\frac{d}{dt}\theta(t)\right)\mathrm{diq}(t) - Nr\,\mathrm{iq}(t)\left(\frac{d}{dt}\mathrm{dtheta}(t)\right) - \frac{Km\,\mathrm{diq}(t)}{J} + \frac{B\left(\frac{d}{dt}\mathrm{dtheta}(t)\right)}{J} + \left(\frac{d^2}{dt^2}\mathrm{dtheta}(t)\right)\right], [t],\right.$$
$$\left.[did, diq, dtheta, dvd, dvq]\right]$$

$$\frac{3+s}{4}$$

As a (torsion) basis for $\Sigma/[u]$ take, for instance, a Brunovsky basis (or any arbitrary basis):

```
> bas := [dtheta(t), diff(dtheta(t),t), did(t), diff(did(t),t)];
```

$$bas := \left[\mathrm{dtheta}(t), \frac{d}{dt}\mathrm{dtheta}(t), \mathrm{did}(t), \frac{d}{dt}\mathrm{did}(t)\right]$$

```
> IsTorsionBase(bas, JLu);
```

$$true$$

Rewrite the expression(s) in the given basis **bas**:

```
> GenCoeff(diff(dtheta(t),t), u, bas, JSigma);
```

$$\_X2(t), [\_U1(t) = \mathrm{dvd}(t), \_U2(t) = \mathrm{dvq}(t)], \left[\_X1(t) = \mathrm{dtheta}(t), \_X2(t) = \frac{d}{dt}\mathrm{dtheta}(t), \_X3(t) = \mathrm{did}(t), \_X4(t) = \frac{d}{dt}\mathrm{did}(t)\right]$$

```
> GenCoeff(diff(dtheta(t),t,t), u, bas, JSigma);
```

$$-\frac{\left(Km\,\mathrm{iq}(t)\,L\,Nr + B\,L\,Nr\left(\frac{d}{dt}\theta(t)\right)\right)\_X2(t)}{JL\,Nr\left(\frac{d}{dt}\theta(t)\right)} + \frac{Km\,R\,\_X3(t)}{JL\,Nr\left(\frac{d}{dt}\theta(t)\right)} + \frac{Km\,\_X4(t)}{JNr\left(\frac{d}{dt}\theta(t)\right)} - \frac{\_U1(t)\,Km}{L\,JNr\left(\frac{d}{dt}\theta(t)\right)}, [\_U1(t) = \mathrm{dvd}(t), \_U2(t) = \mathrm{dvq}(t)],$$
$$\left[\_X1(t) = \mathrm{dtheta}(t), \_X2(t) = \frac{d}{dt}\mathrm{dtheta}(t), \_X3(t) = \mathrm{did}(t), \_X4(t) = \frac{d}{dt}\mathrm{did}(t)\right]$$

Assign names for the elements of **u** or **bas**:

```
> GenCoeff(diff(dtheta(t),t,t), u, bas, JSigma, [u1,u2]);
```

$$\frac{\left(-Km\,\mathrm{iq}(t)\,L\,Nr - B\,L\,Nr\left(\frac{d}{dt}\theta(t)\right)\right)\_X2(t)}{JL\,Nr\left(\frac{d}{dt}\theta(t)\right)} + \frac{Km\,R\,\_X3(t)}{JL\,Nr\left(\frac{d}{dt}\theta(t)\right)} + \frac{Km\,\_X4(t)}{JNr\left(\frac{d}{dt}\theta(t)\right)} - \frac{u1\,Km}{L\,JNr\left(\frac{d}{dt}\theta(t)\right)}, [u1 = \mathrm{dvd}(t), u2 = \mathrm{dvq}(t)],$$
$$\left[\_X1(t) = \mathrm{dtheta}(t), \_X2(t) = \frac{d}{dt}\mathrm{dtheta}(t), \_X3(t) = \mathrm{did}(t), \_X4(t) = \frac{d}{dt}\mathrm{did}(t)\right]$$

```
> GenCoeff(diff(dtheta(t),t,t), u, bas, JSigma, [u1,u2], [x1,x2,x3,x4]);
```

$$\frac{\left(-Km\,\mathrm{iq}(t)\,L\,Nr - B\,L\,Nr\left(\frac{d}{dt}\theta(t)\right)\right)x2}{JL\,Nr\left(\frac{d}{dt}\theta(t)\right)} + \frac{Km\,R\,x3}{JL\,Nr\left(\frac{d}{dt}\theta(t)\right)} + \frac{Km\,x4}{JNr\left(\frac{d}{dt}\theta(t)\right)} - \frac{u1\,Km}{L\,JNr\left(\frac{d}{dt}\theta(t)\right)}, [u1 = \mathrm{dvd}(t), u2 = \mathrm{dvq}(t)],$$

$$\left[x1 = \text{dtheta}(t),\ x2 = \frac{d}{dt}\text{dtheta}(t),\ x3 = \text{did}(t),\ x4 = \frac{d}{dt}\text{did}(t)\right]$$

```
> GenCoeff(diff(dtheta(t),t,t), u, bas, JSigma, "", [x1,x2,x3,x4]);
```

$$-\frac{\left(Km\,\text{iq}(t)\,L\,Nr + B\,L\,Nr\left(\frac{d}{dt}\theta(t)\right)\right)x2}{JL\,Nr\left(\frac{d}{dt}\theta(t)\right)} + \frac{Km\,R\,x3}{JL\,Nr\left(\frac{d}{dt}\theta(t)\right)} + \frac{Km\,x4}{JNr\left(\frac{d}{dt}\theta(t)\right)} - \frac{\_\text{U1}(t)\,Km}{L\,JNr\left(\frac{d}{dt}\theta(t)\right)},\ [\_\text{U1}(t) = \text{dvd}(t),\ \_\text{U2}(t) = \text{dvq}(t)],$$

$$\left[x1 = \text{dtheta}(t),\ x2 = \frac{d}{dt}\text{dtheta}(t),\ x3 = \text{did}(t),\ x4 = \frac{d}{dt}\text{did}(t)\right]$$

```
> GenCoeff(diff(dtheta(t),t,t), u, bas, JSigma, "");
```

$$\frac{\left(-Km\,\text{iq}(t)\,L\,Nr - B\,L\,Nr\left(\frac{d}{dt}\theta(t)\right)\right)\_\text{X2}(t)}{JL\,Nr\left(\frac{d}{dt}\theta(t)\right)} + \frac{Km\,R\,\_\text{X3}(t)}{JL\,Nr\left(\frac{d}{dt}\theta(t)\right)} + \frac{Km\,\_\text{X4}(t)}{JNr\left(\frac{d}{dt}\theta(t)\right)} - \frac{\_\text{U1}(t)\,Km}{L\,JNr\left(\frac{d}{dt}\theta(t)\right)}$$

See Also:

JanetBasis, PrincDeriv, ParamDeriv, InvReduce, HilbertSeries, Ind2Diff, BaseChg, ParamBaseChg.

Janet[HF] - compute the filtered Hilbert function of a linear system of PDEs

Calling Sequence:

HF(i)
HF()

Parameters:

`i` - " " (empty string) or non-negative integer

Description:

- ***HF***(**`i`**) returns the number of free Taylor coefficients (i. e. parametric derivatives) of order up to **`i`** of the general solution of a linear system of PDEs whose Janet basis has been computed last by JanetBasis. Note the same information can be extracted from the command HilbertSeries by summation. Cf. also HP.
- It only uses the data displayable by TabVar and depends on the monomial ordering chosen. In the standard case of degree reverse lexicographic ordering, the order of the indeterminates does not play a role unlike the pure lexicographic case.
- ***HF***() returns a function expecting one parameter **`i`** which computes ***HF***(**`i`**) .
- ***HF***(" ") prints the function ***HF***(). From this print out one can easily deduce the index of regularity, cf. IndexRegularity.

Examples:

```
> with(Janet):
> ivar := [x,y,z,v]; dvar := [u];
```

$$ivar := [x, y, z, v]$$

$$dvar := [u]$$

```
> L := [diff(u(x,y,z,v),x,z)+diff(u(x,y,z,v),y,z)+diff(u(x,y,z,v),x,y),
  diff(u(x,y,z,v),x,y,z)-diff(u(x,y,z,v),v)];
```

$$L := \left[\left(\frac{\partial^2}{\partial z\,\partial x}\,u(x,y,z,v)\right)+\left(\frac{\partial^2}{\partial z\,\partial y}\,u(x,y,z,v)\right)+\left(\frac{\partial^2}{\partial y\,\partial x}\,u(x,y,z,v)\right),\left(\frac{\partial^3}{\partial z\,\partial y\,\partial x}\,u(x,y,z,v)\right)-\left(\frac{\partial}{\partial v}\,u(x,y,z,v)\right)\right]$$

```
> B := JanetBasis(L, ivar, dvar);
```

$$B := \left[\left[\left(\frac{\partial^2}{\partial z\,\partial x}\,u(x,y,z,v)\right)+\left(\frac{\partial^2}{\partial z\,\partial y}\,u(x,y,z,v)\right)+\left(\frac{\partial^2}{\partial y\,\partial x}\,u(x,y,z,v)\right),-\left(\frac{\partial}{\partial v}\,u(x,y,z,v)\right)-\left(\frac{\partial^3}{\partial z^2\,\partial x}\,u(x,y,z,v)\right)-\left(\frac{\partial^3}{\partial z^2\,\partial y}\,u(x,y,z,v)\right)-\left(\frac{\partial^2}{\partial y\,\partial v}\,u(x,y,z,v)\right)-\left(\frac{\partial^4}{\partial z^2\,\partial y^2}\,u(x,y,z,v)\right)-\left(\frac{\partial^2}{\partial z\,\partial v}\,u(x,y,z,v)\right)\right], [x, y, z, v], [u]\right]$$

```
> HilbertSeries();
```

$$1+4\,s+9\,s^2+15\,s^3+s^4\left(15\,\frac{1}{1-s}+6\,\frac{1}{(1-s)^2}\right)$$

```
> f := HF();
```

$$f := HF$$

```
> f(2);
```

14

```
> f(20);
```

1202

```
> HF(20);
```

1202

```
> HF("");
s = 0: 1
s = 1: 5
s = 2: 14
s = 3: 29
s >= 4: 3*s^2+2
> HP();
```

$$3s^2 + 2$$

```
> HilbertFunction("");
Dim(M.0) = 1
Dim(M.1) = 4
Dim(M.2) = 9
Dim(M.3) = 15
Dim(M.s) = -3+6*s, for s >= 4
```

See Also:

JanetBasis, PrincDeriv, ParamDeriv, HilbertFunction, HilbertPolynomial, HP, HilbertSeries, CartanCharacter, PDEBasis, PDEHilbertFunction, PDEHilbertPolynomial, PDEHF, PDEHP, PDEHilbertSeries.

Janet[HP] - compute the filtered Hilbert polynomial of a linear system of PDEs

Calling Sequence:

HP(p)
HP()

Parameters:

p - natural number or name of an indeterminate

Description:

- ***HP***() returns the filtered Hilbert polynomial of a linear system of PDEs whose Janet basis has been computed last by JanetBasis. Note this same information can be extracted from the command HE, which also yields the index of regularity and the dimensions of the earlier sections, cf. also IndexRegularity.
- ***HP***(**p**) returns the value of the polynomial function ***HP*** at **p**. Note, **p** must be greater or equal to the index of regularity, or **p** must be the name of the indeterminate for the Hilbert polynomial.
- If **p** is a natural number greater or equal to the index of regularity, ***HP***(**p**) is the number of free Taylor coefficients of order up to **p** in the expansion of a general solution of the system of PDEs, i. e. the number of parametric derivatives of order up to **p**, up to some fixed additive constant.
- The command only uses the data displayable by TabVar and depends on the ordering of derivatives chosen for the last call of ***JanetBasis***.
- The default name of the indeterminate is 's'. It will not be affected by a subs command.

Examples:

```
> with(Janet):
> ivar := [x,y,z,v]; dvar := [u];
```

$$ivar := [x, y, z, v]$$

$$dvar := [u]$$

```
> L := [diff(u(x,y,z,v),x,z)+diff(u(x,y,z,v),y,z)+diff(u(x,y,z,v),x,y),
  diff(u(x,y,z,v),x,y,z)-diff(u(x,y,z,v),v)];
```

$$L := \left[\left(\frac{\partial^2}{\partial z\,\partial x}\mathrm{u}(x,y,z,v)\right)+\left(\frac{\partial^2}{\partial z\,\partial y}\mathrm{u}(x,y,z,v)\right)+\left(\frac{\partial^2}{\partial y\,\partial x}\mathrm{u}(x,y,z,v)\right),\left(\frac{\partial^3}{\partial z\,\partial y\,\partial x}\mathrm{u}(x,y,z,v)\right)-\left(\frac{\partial}{\partial v}\mathrm{u}(x,y,z,v)\right)\right]$$

```
> JanetBasis(L, ivar, dvar);
```

$$\left[\left[\left(\frac{\partial^2}{\partial z\,\partial x}\mathrm{u}(x,y,z,v)\right)+\left(\frac{\partial^2}{\partial z\,\partial y}\mathrm{u}(x,y,z,v)\right)+\left(\frac{\partial^2}{\partial y\,\partial x}\mathrm{u}(x,y,z,v)\right)\left(\frac{\partial}{\partial v}\mathrm{u}(x,y,z,v)\right)+\left(\frac{\partial^3}{\partial z^2\,\partial x}\mathrm{u}(x,y,z,v)\right)+\left(\frac{\partial^3}{\partial z^2\,\partial y}\mathrm{u}(x,y,z,v)\right),\left(\frac{\partial^4}{\partial z^2\,\partial y^2}\mathrm{u}(x,y,z,v)\right)+\left(\frac{\partial^2}{\partial y\,\partial v}\mathrm{u}(x,y,z,v)\right)+\left(\frac{\partial^2}{\partial z\,\partial v}\mathrm{u}(x,y,z,v)\right)\right],[x,y,z,v],[u]\right]$$

```
> HilbertSeries();
```

$$1+4\,s+9\,s^2+15\,s^3+s^4\left(\frac{15}{1-s}+\frac{6}{(1-s)^2}\right)$$

```
> HP();
```

$$3\,s^2+2$$

```
> HP(20);
```

$$1202$$

```
> HF("");
s = 0: 1
s = 1: 5
s = 2: 14
s = 3: 29
s >= 4: 3*s^2+2
> HP(lambda);
```

$$3\,\lambda^2+2$$

```
> subs(lambda=4, %);
```

$$50$$

```
> HilbertPolynomial(lambda);
```

$$-3+6\,\lambda$$

See Also:

JanetBasis, TabVar, ParamDeriv, HilbertPolynomial, HilbertFunction, HF, HilbertSeries, CartanCharacter, PDEBasis, PDEHilbertPolynomial, PDEHilbertFunction, PDEHP, PDEHF, PDEHilbertSeries.

Janet[HilbertFunction] - compute the graded Hilbert function of a linear system of PDEs

Calling Sequence:

HilbertFunction(i)
HilbertFunction()

Parameters:

`i` - "" (empty string) or non-negative integer

Description:

- ***HilbertFunction***(**`i`**) returns the number of free Taylor coefficients of order **`i`** of the general solution, also called the parametric derivatives of order **`i`**, of a linear system of PDEs whose Janet basis has been computed last by JanetBasis. This information can also be seen in the Hilbert series, cf. HilbertSeries.
- ***HilbertFunction***() returns a function expecting one parameter **`i`** which computes ***HilbertFunction***(**`i`**) .
- ***HilbertFunction***("") prints the function ***HilbertFunction***().
- The command only uses the data displayable by TabVar and depends on the ordering of derivatives chosen for the last call of ***JanetBasis***.

Examples:

```
> with(Janet):
> ivar := [x,y,z,v]; dvar := [u];
```

$$ivar := [x, y, z, v]$$

$$dvar := [u]$$

```
> L := [diff(u(x,y,z,v),x,z)+diff(u(x,y,z,v),y,z)+diff(u(x,y,z,v),x,y),
  diff(u(x,y,z,v),x,y,z)-diff(u(x,y,z,v),v$3)];
```

$$L := \left[\left(\frac{\partial^2}{\partial z\,\partial x}\mathrm{u}(x,y,z,v)\right)+\left(\frac{\partial^2}{\partial z\,\partial y}\mathrm{u}(x,y,z,v)\right)+\left(\frac{\partial^2}{\partial y\,\partial x}\mathrm{u}(x,y,z,v)\right)\left(\frac{\partial^3}{\partial z\,\partial y\,\partial x}\mathrm{u}(x,y,z,v)\right)-\left(\frac{\partial^3}{\partial v^3}\mathrm{u}(x,y,z,v)\right)\right]$$

```
> B := JanetBasis(L, ivar, dvar);
```

$$B := \left[\left[\left(\frac{\partial^2}{\partial z\,\partial x}\mathrm{u}(x,y,z,v)\right)+\left(\frac{\partial^2}{\partial z\,\partial y}\mathrm{u}(x,y,z,v)\right)+\left(\frac{\partial^2}{\partial y\,\partial x}\mathrm{u}(x,y,z,v)\right), \left(\frac{\partial^3}{\partial v^3}\mathrm{u}(x,y,z,v)\right)+\left(\frac{\partial^3}{\partial z^2\,\partial x}\mathrm{u}(x,y,z,v)\right)+\left(\frac{\partial^3}{\partial z^2\,\partial y}\mathrm{u}(x,y,z,v)\right), \left(\frac{\partial^4}{\partial z^2\,\partial y^2}\mathrm{u}(x,y,z,v)\right)+\left(\frac{\partial^4}{\partial y\,\partial v^3}\mathrm{u}(x,y,z,v)\right)+\left(\frac{\partial^4}{\partial z\,\partial v^3}\mathrm{u}(x,y,z,v)\right)\right], [x, y, z, v], [u]\right]$$

```
> HilbertSeries();
```

$$1+4\,s+9\,s^2+15\,s^3+s^4\left(15\,\frac{1}{1-s}+6\,\frac{1}{(1-s)^2}\right)$$

```
> f := HilbertFunction();
```

$$f := Janet/HilbertFunction$$

```
> f(2);
```

$$9$$

```
> f(20);
```

$$117$$

```
> HilbertFunction(20);
```

$$117$$

```
> HilbertFunction("");
Dim(M.0) = 1
Dim(M.1) = 4
Dim(M.2) = 9
Dim(M.3) = 15
Dim(M.s) = -3+6*s, for s >= 4
```

See Also:

JanetBasis, PrincDeriv, ParamDeriv, HF, HilbertPolynomial, HP, HilbertSeries, CartanCharacter, PDEBasis, PDEHilbertFunction, PDEHF
, PDEHilbertPolynomial, PDEHP, PDEHilbertSeries.

Janet[HilbertPolynomial] - compute the graded Hilbert polynomial of a linear system of PDEs

Calling Sequence:

HilbertPolynomial(p)

Parameters:

p - (optional) name of the indeterminate (default: 's') or non-negative integer

Description:

- ***HilbertPolynomial***() returns the graded Hilbert polynomial of the linear system of PDEs whose Janet basis has been computed last by JanetBasis. Note, the same information can be extracted from the command HilbertFunction.
- ***HilbertPolynomial***(**p**) returns the value of the polynomial function ***HilbertPolynomial*** at **p**, where **p** might also be the name of the indeterminate for the Hilbert polynomial. If **p** is a non-negative integer greater than or equal to the index of regularity, ***HilbertPolynomial***(**p**) returns the number of free Taylor coefficients of order **p** in the expansion of a general solution of the system of PDEs, i.e. the number of parametric derivatives of order **p**. The same information for all non-negative integers can be obtained from HilbertFunction or HilbertSeries.
- The command only uses the data displayable by TabVar and depends on the ordering of derivatives chosen for the last call of ***JanetBasis***.
- The default name of the indeterminate is 's'. It will not be affected by a subs command.

Examples:

```
> with(Janet):
> ivar := [x,y,z,v]; dvar := [u];
```

$$ivar := [x, y, z, v]$$

$$dvar := [u]$$

```
> L := [diff(u(x,y,z,v),x,z)+diff(u(x,y,z,v),y,z)+diff(u(x,y,z,v),x,y),
  diff(u(x,y,z,v),x,y,z)-diff(u(x,y,z,v),v$3)];
```

$$L := \left[\left(\frac{\partial^2}{\partial z\,\partial x}\,u(x,y,z,v)\right)+\left(\frac{\partial^2}{\partial z\,\partial y}\,u(x,y,z,v)\right)+\left(\frac{\partial^2}{\partial y\,\partial x}\,u(x,y,z,v)\right)\left(\frac{\partial^3}{\partial z\,\partial y\,\partial x}\,u(x,y,z,v)\right)-\left(\frac{\partial^3}{\partial v^3}\,u(x,y,z,v)\right)\right]$$

```
> JanetBasis(L, ivar, dvar);
```

$$\left[\left[\left(\frac{\partial^2}{\partial z\,\partial x}\,u(x,y,z,v)\right)+\left(\frac{\partial^2}{\partial z\,\partial y}\,u(x,y,z,v)\right)+\left(\frac{\partial^2}{\partial y\,\partial x}\,u(x,y,z,v)\right)\left(\frac{\partial^3}{\partial v^3}\,u(x,y,z,v)\right)+\left(\frac{\partial^3}{\partial z^2\,\partial x}\,u(x,y,z,v)\right)+\left(\frac{\partial^3}{\partial z^2\,\partial y}\,u(x,y,z,v)\right)\left(\frac{\partial^4}{\partial z^2\,\partial y^2}\,u(x,y,z,v)\right)+\left(\frac{\partial^4}{\partial y\,\partial v^3}\,u(x,y,z,v)\right)+\left(\frac{\partial^4}{\partial z\,\partial v^3}\,u(x,y,z,v)\right)\right], [x, y, z, v], [u]\right]$$

```
> HilbertSeries();
```

$$1+4\,s+9\,s^2+15\,s^3+s^4\left(\frac{15}{1-s}+\frac{6}{(1-s)^2}\right)$$

```
> HilbertPolynomial();
```

$$-3+6\,s$$

```
> HilbertPolynomial(6);
```

$$33$$

```
> HP(6);
```

$$110$$

```
> HilbertFunction("");
Dim(M.0) = 1
Dim(M.1) = 4
Dim(M.2) = 9
Dim(M.3) = 15
Dim(M.s) = -3+6*s, for s >= 4
> HilbertPolynomial(lambda);
```

$$-3+6\,\lambda$$

```
> subs(lambda=4, %);
```

21

See Also:

JanetBasis, PrincDeriv, ParamDeriv, HP, HilbertFunction, HF, HilbertSeries, PDEBasis, PDEHilbertPolynomial, PDEHP, PDEHilbertFunction, PDEHF, PDEHilbertSeries.

Janet[HilbertSeries] - generating function for the number of free Taylor coefficients for the general solution of a linear system of PDEs

Calling Sequence:

HilbertSeries(v)

Parameters:

v - (optional) name of the indeterminate (default: 's')

Description:

- ***HilbertSeries*** returns the Hilbert series of the Janet basis produced by the last call of JanetBasis. Therefore it is necessary to call JanetBasis first. The output is the generating function of the number of free Taylor coefficients, also called parametric derivatives, of the general solution of the system, whose Janet basis has been computed last. It only uses the data displayable by TabVar and depends on the ordering of derivatives chosen for the previous Janet basis computation.
- The output can be expanded in the form $\sum_{i=0}^{\infty} d_i v^i$, where d_i is the number of free Taylor coefficients (i. e. parametric derivatives) of order i as described above.
- The default name of the indeterminate is 's'. To use the subs command later on the indeterminate, one has to explicitly give a name to the indeterminate.
- To enumerate rather than count the parametric derivatives cf. ParamDeriv.

Examples:

```
> with(Janet):
> ivar := [x,y,t]; dvar := [u];
```

$$ivar := [x, y, t]$$

$$dvar := [u]$$

```
> L := [diff(u(x,y,t),x$2) - diff(u(x,y,t),t$2), diff(u(x,y,t),y)-u(x,y,t)];
```

$$L := \left[\left(\frac{\partial^2}{\partial x^2}\mathrm{u}(x,y,t)\right) - \left(\frac{\partial^2}{\partial t^2}\mathrm{u}(x,y,t)\right), \left(\frac{\partial}{\partial y}\mathrm{u}(x,y,t)\right) - \mathrm{u}(x,y,t)\right]$$

```
> B := JanetBasis(L, ivar, dvar);
```

$$B := \left[\left[\left(\frac{\partial}{\partial y}\mathrm{u}(x,y,t)\right) - \mathrm{u}(x,y,t), -\left(\frac{\partial}{\partial x}\mathrm{u}(x,y,t)\right) + \left(\frac{\partial^2}{\partial y\,\partial x}\mathrm{u}(x,y,t)\right), \left(\frac{\partial^2}{\partial x^2}\mathrm{u}(x,y,t)\right) - \left(\frac{\partial^2}{\partial t^2}\mathrm{u}(x,y,t)\right)\right], [x, y, t], [u]\right]$$

```
> HilbertSeries();
```

$$2s + 1 + 2\frac{s^2}{1-s}$$

```
> SolSeries(B, 3);
```

$$\left[\mathrm{u}(x,y,t) = C1_{0,0,0} + C1_{1,0,0}\,x + C1_{0,0,0}\,y + C1_{0,0,1}\,t + \frac{1}{2}C1_{0,0,2}\,t^2 + C1_{1,0,1}\,xt + C1_{0,0,1}\,yt + \frac{1}{2}C1_{0,0,2}\,x^2 + C1_{1,0,0}\,xy + \frac{1}{2}C1_{0,0,0}\,y^2 + \frac{1}{6}C1_{0,0,3}\,t^3 + \frac{1}{2}C1_{1,0,2}\,xt^2 + \frac{1}{2}C1_{0,0,2}\,yt^2 + \frac{1}{2}C1_{0,0,3}\,x^2 t + C1_{1,0,1}\,xyt + \frac{1}{2}C1_{0,0,1}\,y^2 t + \frac{1}{6}C1_{1,0,2}\,x^3 + \frac{1}{2}C1_{0,0,2}\,x^2 y + \frac{1}{2}C1_{1,0,0}\,xy^2 + \frac{1}{6}C1_{0,0,0}\,y^3\right]$$

```
> HilbertSeries(lambda);
```

$$2\lambda + 1 + 2\frac{\lambda^2}{1-\lambda}$$

```
> taylor(%, lambda=0, 6);
```

$$1 + 2\lambda + 2\lambda^2 + 2\lambda^3 + 2\lambda^4 + 2\lambda^5 + \mathrm{O}(\lambda^6)$$

See Also:

JanetBasis, PrincDeriv, ParamDeriv, HilbertPolynomial, HP, HilbertFunction, HF, IndexRegularity, CartanCharacter, PDEBasis, PDEHilbertSeries, PDEHilbertPolynomial, PDEHP, PDEHilbertFunction, PDEHF.

Janet[Ind2Diff] - convert differential expression from jet notation to Maple notation

Calling Sequence:

Ind2Diff(L,ivar,dvar)

Parameters:

`L` - differential expression or list of differential expressions in jet notation
`ivar` - list of independent variables
`dvar` - list of dependent variables

Description:

- ***Ind2Diff*** translates the differential expressions in **L** given in jet notation to the usual Maple notation.
- The jet notation is adapted from the Maple package jets. It is much more compact than the usual Maple notation for differential expressions. To fix ideas, let *u* be a dependent variable and *x,y,z* the independent variables. Then the jet notation for $\frac{\partial^4}{\partial z\,\partial y\,\partial y\,\partial x}\mathrm{u}(x,y,z)$ is $u_{x,y,y,z}$. The name Ind comes form index notation.
- The command Diff2Ind is inverse to ***Ind2Diff***.

Examples:

```
> with(Janet):
> Ind2Diff(u[x,y], [x,y], [u]);
```

$$\frac{\partial^2}{\partial y\,\partial x}\mathrm{u}(x,y)$$

```
> ivar := [x,y,z]; dvar := [u,v];
```

$$ivar := [x, y, z]$$

$$dvar := [u, v]$$

```
> L := [u[y]+v[x,x], x*u[y,y]+v[z]];
```

$$L := [u_y + v_{x,x},\ x\,u_{y,y} + v_z]$$

```
> Lex := Ind2Diff(L, ivar, dvar);
```

$$Lex := \left[\left(\frac{\partial}{\partial y}\mathrm{u}(x,y,z)\right)+\left(\frac{\partial^2}{\partial x^2}\mathrm{v}(x,y,z)\right),\ x\left(\frac{\partial^2}{\partial y^2}\mathrm{u}(x,y,z)\right)+\left(\frac{\partial}{\partial z}\mathrm{v}(x,y,z)\right)\right]$$

```
> Diff2Ind(Lex, ivar, dvar);
```

$$[u_y + v_{x,x},\ x\,u_{y,y} + v_z]$$

See Also:

Diff2Ind, Pol2Ind, AppOpInd, Diff2Pol, Pol2Diff, Diff2Op, AppOp, Op2D, D2Op, Diff2D, D2Diff

Janet[IndexRegularity] - return the biggest degree where Hilbert polynomial and Hilbert function differ

Calling Sequence:

IndexRegularity()

Parameters:

- none (assumes that the Janet basis has been computed before)

Description:

- ***IndexRegularity*** returns the biggest integer *i* on which the HilbertFunction and the HilbertPolynomial (both counting the generic number of parametric derivatives according to degree) return different values. In other words *i* is the smallest integer such that HP and HF agree on all values greater or equal to *i*.
- The command refers to the last call of JanetBasis.

Examples:

```
> with(Janet):
> ivar := [x,y,z,v]; dvar := [u];
```

$$ivar := [x, y, z, v]$$

$$dvar := [u]$$

```
> L := [diff(u(x,y,z,v),x,z)+diff(u(x,y,z,v),y,z)+diff(u(x,y,z,v),x,y),
  diff(u(x,y,z,v),x,y,z)-diff(u(x,y,z,v),v)];
```

$$L := \left[\left(\frac{\partial^2}{\partial z\,\partial x}u(x,y,z,v)\right)+\left(\frac{\partial^2}{\partial z\,\partial y}u(x,y,z,v)\right)+\left(\frac{\partial^2}{\partial y\,\partial x}u(x,y,z,v)\right),\left(\frac{\partial^3}{\partial z\,\partial y\,\partial x}u(x,y,z,v)\right)-\left(\frac{\partial}{\partial v}u(x,y,z,v)\right)\right]$$

```
> B := JanetBasis(L, ivar, dvar);
```

$$B := \left[\left[\left(\frac{\partial^2}{\partial z\,\partial x}u(x,y,z,v)\right)+\left(\frac{\partial^2}{\partial z\,\partial y}u(x,y,z,v)\right)+\left(\frac{\partial^2}{\partial y\,\partial x}u(x,y,z,v)\right),-\left(\frac{\partial}{\partial v}u(x,y,z,v)\right)-\left(\frac{\partial^3}{\partial z^2\,\partial x}u(x,y,z,v)\right)-\left(\frac{\partial^3}{\partial z^2\,\partial y}u(x,y,z,v)\right)\right.\right.$$
$$\left.\left.-\left(\frac{\partial^2}{\partial y\,\partial v}u(x,y,z,v)\right)-\left(\frac{\partial^4}{\partial z^2\,\partial y^2}u(x,y,z,v)\right)-\left(\frac{\partial^2}{\partial z\,\partial v}u(x,y,z,v)\right)\right],[x,y,z,v],[u]\right]$$

```
> IndexRegularity();
```

$$1$$

```
> HilbertSeries();
```

$$1+4\,s+9\,s^2+15\,s^3+s^4\left(15\,\frac{1}{1-s}+6\,\frac{1}{(1-s)^2}\right)$$

```
> HP();
```

$$3\,s^2+2$$

```
> HF("");
s = 0: 1
s = 1: 5
s = 2: 14
s = 3: 29
s >= 4: 3*s^2+2
> HilbertPolynomial();
```

$$-3+6\,s$$

```
> HilbertFunction("");
Dim(M.0) = 1
Dim(M.1) = 4
Dim(M.2) = 9
Dim(M.3) = 15
Dim(M.s) = -3+6*s, for s >= 4
```

See Also:

JanetBasis, PrincDeriv, ParamDeriv, HilbertPolynomial, HilbertFunction, HilbertSeries, HP, HF, PDEBasis, PDEHilbertPolynomial, PDEHilbertFunction, PDEHilbertSeries, PDEHP, PDEHF.

Janet[Intersection] - intersect two differential modules

Calling Sequence:

Intersection(L1,L2,ivar,dvar)

Parameters:

`L1` - list of linear differential expressions (to be interpreted as module generators)
`L2` - list of linear differential expressions (to be interpreted as module generators)
`ivar` - list of independent variables
`dvar` - list of dependent variables

Description:

- ***Intersection*** computes a Janet basis of the intersection of the differential modules generated by **`L1`** and **`L2`** (with respect to degrevlex).
- The elements in **`L1`** and **`L2`** are expected to be linear differential expressions in the dependent variables **`dvar`** and the independent variables **`ivar`**.
- The result of ***Intersection*** is a list of length 3 representing a Janet basis as explained in JanetBasis.

Examples:

```
> with(Janet):
```

Example 1: least common left multiple of two linear ordinary differential operators

```
> ivar := [x];
```

$$ivar := [x]$$

```
> dvar := [u];
```

$$dvar := [u]$$

```
> L1 := [diff(u(x), x) - 1/x * u(x)];
  L2 := [diff(u(x), x) - (1+2*x)/(x*(1+x)) * u(x)];
```

$$L1 := \left[\left(\frac{d}{dx}\,u(x)\right) - \frac{u(x)}{x}\right]$$

$$L2 := \left[\left(\frac{d}{dx}\,u(x)\right) - \frac{(1+2x)\,u(x)}{x(1+x)}\right]$$

```
> J := Intersection(L1, L2, ivar, dvar);
```

$$J := \left[\left[2\,u(x) - 2x\left(\frac{d}{dx}\,u(x)\right) + x^2\left(\frac{d^2}{dx^2}\,u(x)\right)\right], [x], [u]\right]$$

Example 2:

```
> ivar := [x,y];
```

$$ivar := [x, y]$$

```
> dvar := [u];
```

$$dvar := [u]$$

```
> L1 := [diff(u(x,y), x, x), diff(u(x,y), y, y)];
```

$$L1 := \left[\frac{\partial^2}{\partial x^2}\,u(x,y), \frac{\partial^2}{\partial y^2}\,u(x,y)\right]$$

```
> L2 := [diff(u(x,y), x)-diff(u(x,y), y)];
```

$$L2 := \left[\left(\frac{\partial}{\partial x}\,u(x,y)\right) - \left(\frac{\partial}{\partial y}\,u(x,y)\right)\right]$$

```
> J := Intersection(L1, L2, ivar, dvar);
```

$$J := \left[\left[\left(\frac{\partial^2}{\partial x^2}\,u(x,y)\right) - \left(\frac{\partial^2}{\partial y^2}\,u(x,y)\right), \left(\frac{\partial^3}{\partial y^2\,\partial x}\,u(x,y)\right) - \left(\frac{\partial^3}{\partial y^3}\,u(x,y)\right)\right], [x,y], [u]\right]$$

Example 3:

```
> ivar := [x,y];
```

$$ivar := [x, y]$$

```
> dvar := [u,v];
```

$$dvar := [u, v]$$

```
> L1 := [diff(u(x,y), x)-diff(v(x,y), y), diff(u(x,y), y)+diff(v(x,y), x)];
```

$$L1 := \left[\left(\frac{\partial}{\partial x}\,u(x, y)\right) - \left(\frac{\partial}{\partial y}\,v(x, y)\right), \left(\frac{\partial}{\partial y}\,u(x, y)\right) + \left(\frac{\partial}{\partial x}\,v(x, y)\right)\right]$$

```
> L2 := [diff(u(x,y), x, x)+diff(u(x,y), y, y)];
```

$$L2 := \left[\left(\frac{\partial^2}{\partial x^2}\,u(x, y)\right) + \left(\frac{\partial^2}{\partial y^2}\,u(x, y)\right)\right]$$

```
> J := Intersection(L1, L2, ivar, dvar);
```

$$J := \left[\left[\left(\frac{\partial^2}{\partial x^2}\,u(x, y)\right) + \left(\frac{\partial^2}{\partial y^2}\,u(x, y)\right)\right], [x, y], [u, v]\right]$$

Example 4:

```
> ivar := [x,y];
```

$$ivar := [x, y]$$

```
> dvar := [u];
```

$$dvar := [u]$$

```
> L1 := [u(x,y)-a(x,y)]; L2 := [u(x,y)-b(x,y)];
```

$$L1 := [u(x, y) - a(x, y)]$$

$$L2 := [u(x, y) - b(x, y)]$$

```
> Intersection(L1, L2, ivar, dvar);
```

$$[0, [x, y], [u]]$$

```
> ZeroSets(ivar);
```

$$[b(x, y) - a(x, y)]$$

See Also:

JanetBasis, PrincDeriv, ParamDeriv, InvReduce, HilbertSeries, ZeroSets, Diff2Op, AppOp.

Janet[InvReduce] - return the normal form of a differential expression with respect to a Janet basis

Calling Sequence:

InvReduce(f,JB,ord,cf)

Parameters:

`f` - (list of) linear differential expression(s) to be reduced
`JB` - Janet basis
`ord` - (optional) change of ordering for differential monomials
`cf` - (optional) string "C", return a differential operator which encodes the reduction process

Description:

- ***InvReduce*** returns the normal form of the linear differential expression **`f`** modified by the expressions in the Janet basis **`JB`** and their derivatives. This is done by involutive reduction. In particular the normal form contains no derivatives of leading terms of any element of the Janet basis. Note, this normal form is zero if and only if **`f`** can be obtained from the elements of the Janet basis by differentiation, addition and multiplication with scalar functions. If **`f`** is a list of differential expressions, then the list of the corresponding normal forms is returned.
- **`JB`** is a list which contains the Janet basis (a list of linear differential expressions) as first entry and the arguments to the call of JanetBasis which produced this Janet basis as following entries. Most commonly **`JB`** is the output of the previous call of ***JanetBasis.*** If this is the case, ***InvReduce*** performs involutive reduction on **`f`** with **`JB`**, more precisely with an internal data structure, constructed by JanetBasis and which can be displayed by the command PrincDeriv. Note, the program does not check whether **`JB`** actually is a Janet basis with respect to the variables and term order defined by the entries in **`JB`** (or determined by **`ord`**). Therefore, if **`JB`** is a Janet basis w.r.t. a non-standard ordering of differential monomials, then this term ordering must be selected as described next (cf. also Example 3 below).
- The order on differential monomials can be set either by the optional third argument **`ord`** or by the (also optional) fourth entry of **`JB`**. The values 1 to 4 are accepted. In case 1, the pure lexicographical ordering is chosen. In case 2, the degree reverse lexicographical order is selected. The values 3 and 4 select pure lexicographical ordering and degree reverse lexicographical ordering respectively, but change from "position over term" order to "term over position" order, i.e. giving priority to dependent variables or independent variables respectively. The default value is 4.
- If JanetBasis was called with user defined degrees for independent and / or dependent variables, the corresponding parameters **`ivar`** and **`dvar`** have to be specified in the second and third entry of **`JB`** in the same manner (cf. Example 4 below).
- If the optional argument **`cf`**, which equals the string "C", is present, then ***InvReduce*** returns additionally a differential operator by means of which one can reconstruct the reduction process. If one applies this operator to the Janet basis in **`JB`** and substracts the result from **`f`**, then one obtains the normal form representative of **`f`** with respect to the Janet basis.

Examples:

```
> with(Janet):
```

Example 1: Cauchy-Riemann differential equations

```
> ivar := [x,y]; dvar := [u,v];
```

$$ivar := [x, y]$$

$$dvar := [u, v]$$

```
> L := [diff(u(x,y),x) - diff(v(x,y),y), diff(u(x,y),y) + diff(v(x,y),x)];
```

$$L := \left[\left(\frac{\partial}{\partial x}\mathrm{u}(x,y)\right) - \left(\frac{\partial}{\partial y}\mathrm{v}(x,y)\right), \left(\frac{\partial}{\partial y}\mathrm{u}(x,y)\right) + \left(\frac{\partial}{\partial x}\mathrm{v}(x,y)\right)\right]$$

```
> JB := JanetBasis(L, ivar, dvar);
```

$$JB := \left[\left[\left(\frac{\partial}{\partial y}\mathrm{u}(x,y)\right) + \left(\frac{\partial}{\partial x}\mathrm{v}(x,y)\right), \left(\frac{\partial}{\partial x}\mathrm{u}(x,y)\right) - \left(\frac{\partial}{\partial y}\mathrm{v}(x,y)\right)\right], [x, y], [u, v]\right]$$

```
> PrincDeriv();
```

$$\left[\left(\frac{\partial}{\partial y}\mathrm{u}(x,y)\right)+\left(\frac{\partial}{\partial x}\mathrm{v}(x,y)\right),[x,y],\frac{\partial}{\partial x}\mathrm{v}(x,y)\right]$$

$$\left[\left(\frac{\partial}{\partial x}\mathrm{u}(x,y)\right)-\left(\frac{\partial}{\partial y}\mathrm{v}(x,y)\right),[x,y],\frac{\partial}{\partial x}\mathrm{u}(x,y)\right]$$

```
> f := diff(u(x,y), x$2, y$2);
```

$$f := \frac{\partial^4}{\partial y^2\,\partial x^2}\mathrm{u}(x,y)$$

```
> InvReduce(f, JB);
```

$$-\left(\frac{\partial^4}{\partial y^4}\mathrm{u}(x,y)\right)$$

```
> InvReduce([diff(u(x,y),x,y), diff(u(x,y),x$4)], JB);
```

$$\left[\frac{\partial^2}{\partial y^2}\mathrm{v}(x,y),\frac{\partial^4}{\partial y^4}\mathrm{u}(x,y)\right]$$

Example 2: Explicitly give the expression of the normal form of **f** in terms of the original system

```
> ivar := [x,y]; dvar := [u,v];
```

$$ivar := [x,y]$$

$$dvar := [u,v]$$

```
> L := [diff(u(x,y),x) - diff(v(x,y),y), diff(u(x,y),y) + diff(v(x,y),x)];
```

$$L := \left[\left(\frac{\partial}{\partial x}\mathrm{u}(x,y)\right)-\left(\frac{\partial}{\partial y}\mathrm{v}(x,y)\right),\left(\frac{\partial}{\partial y}\mathrm{u}(x,y)\right)+\left(\frac{\partial}{\partial x}\mathrm{v}(x,y)\right)\right]$$

```
> L2 := AffEqn(L, ivar, [a,b]);
```

$$L2 := \left[\left(\frac{\partial}{\partial x}\mathrm{u}(x,y)\right)-\left(\frac{\partial}{\partial y}\mathrm{v}(x,y)\right)-\mathrm{a}(x,y),\left(\frac{\partial}{\partial y}\mathrm{u}(x,y)\right)+\left(\frac{\partial}{\partial x}\mathrm{v}(x,y)\right)-\mathrm{b}(x,y)\right]$$

```
> JB := JanetBasis(L2, ivar, dvar);
```

$$JB := \left[\left[\left(\frac{\partial}{\partial y}\mathrm{u}(x,y)\right)+\left(\frac{\partial}{\partial x}\mathrm{v}(x,y)\right)-\mathrm{b}(x,y),\left(\frac{\partial}{\partial x}\mathrm{u}(x,y)\right)-\left(\frac{\partial}{\partial y}\mathrm{v}(x,y)\right)-\mathrm{a}(x,y)\right],[x,y],[u,v]\right]$$

```
> f := diff(u(x,y), x$2, y$2);
```

$$f := \frac{\partial^4}{\partial y^2\,\partial x^2}\mathrm{u}(x,y)$$

```
> InvReduce(f-F(x,y), JB);
```

$$-\left(\frac{\partial^4}{\partial y^4}\mathrm{u}(x,y)\right)-\mathrm{F}(x,y)+\left(\frac{\partial^3}{\partial y^2\,\partial x}\mathrm{a}(x,y)\right)+\left(\frac{\partial^3}{\partial y^3}\mathrm{b}(x,y)\right)$$

Alternative way of tracing the reduction steps of ***InvReduce***:

```
> JB := JanetBasis(L, ivar, dvar);
```

$$JB := \left[\left[\left(\frac{\partial}{\partial y}\mathrm{u}(x,y)\right)+\left(\frac{\partial}{\partial x}\mathrm{v}(x,y)\right),\left(\frac{\partial}{\partial x}\mathrm{u}(x,y)\right)-\left(\frac{\partial}{\partial y}\mathrm{v}(x,y)\right)\right],[x,y],[u,v]\right]$$

```
> InvReduce(f, JB, "C");
```

$$\left[-\left(\frac{\partial^4}{\partial y^4}\mathrm{u}(x,y)\right)[[[1,[y,y,y]]]\quad[[1,[x,y,y]]]]\right]$$

```
> AppOp(%[2], JB[1], ivar, dvar);
```

$$\left[\left(\frac{\partial^4}{\partial y^4}\mathrm{u}(x,y)\right)+\left(\frac{\partial^4}{\partial y^2\,\partial x^2}\mathrm{u}(x,y)\right)\right]$$

Subtracting this combination of the basis elements of **JB** yields the result of Example 1:

```
> f - op(%);
```

$$-\left(\frac{\partial^4}{\partial y^4}\mathrm{u}(x,y)\right)$$

Example 3: Influence of term ordering

```
> ivar := [x,y]; dvar := [u];
```

$$ivar := [x,y]$$

$$dvar := [u]$$

Compute a Janet basis w.r.t. the lexicographical term ordering on the differential monomials (indicated by 1 here):

```
> JB := JanetBasis([diff(u(x,y),y$2)-diff(u(x,y),x)], ivar, dvar, 1);
```

$$JB := \left[\left[\left(\frac{\partial}{\partial x}\,u(x,y)\right)-\left(\frac{\partial^2}{\partial y^2}\,u(x,y)\right)\right], [x,y], [u]\right]$$

```
> f := diff(u(x,y), x);
```

$$f := \frac{\partial}{\partial x}\,u(x,y)$$

```
> InvReduce(f, JB);
```

$$\frac{\partial}{\partial x}\,u(x,y)$$

```
> InvReduce(f, JB, 1);
```

$$\frac{\partial^2}{\partial y^2}\,u(x,y)$$

```
> InvReduce(f, [op(JB), 1]);
```

$$\frac{\partial^2}{\partial y^2}\,u(x,y)$$

JB is a Janet basis w.r.t. any term ordering (because it consists of one element only); but the normal form of an element of the ideal depends on the chosen term ordering.

```
> JB := JanetBasis([diff(u(x,y), y$3)-diff(u(x,y), x), diff(u(x,y), x$2)-diff(u(x,y),
  y)], ivar, dvar);
```

$$JB := \left[\left[\left(\frac{\partial^2}{\partial x^2}\,u(x,y)\right)-\left(\frac{\partial}{\partial y}\,u(x,y)\right)\left(\frac{\partial^3}{\partial y^3}\,u(x,y)\right)-\left(\frac{\partial}{\partial x}\,u(x,y)\right), -\left(\frac{\partial}{\partial y}\,u(x,y)\right)+\left(\frac{\partial^4}{\partial y^3\,\partial x}\,u(x,y)\right)\right], [x,y], [u]\right]$$

```
> f := diff(u(x,y), y$6)-diff(u(x,y), y);
```

$$f := \left(\frac{\partial^6}{\partial y^6}\,u(x,y)\right)-\left(\frac{\partial}{\partial y}\,u(x,y)\right)$$

```
> InvReduce(f, JB);
```

$$0$$

```
> InvReduce(f, JB, 1);
```

$$\left(\frac{\partial^6}{\partial y^6}\,u(x,y)\right)-\left(\frac{\partial}{\partial y}\,u(x,y)\right)$$

```
> InvReduce(f, [op(JB), 1]);
```

$$\left(\frac{\partial^6}{\partial y^6}\,u(x,y)\right)-\left(\frac{\partial}{\partial y}\,u(x,y)\right)$$

JB is not a Janet basis w.r.t. the lexicographical term ordering, leading to a wrong normal form here.

Example 4: Assigning degrees to the independent variables

```
> ivar := [x,t=2]; dvar := [u];
```

$$ivar := [x, t=2]$$

$$dvar := [u]$$

```
> L := [diff(u(x,t),x,x)-diff(u(x,t),t)];
```

$$L := \left[\left(\frac{\partial^2}{\partial x^2}\,u(x,t)\right)-\left(\frac{\partial}{\partial t}\,u(x,t)\right)\right]$$

```
> JB := JanetBasis(L, ivar, dvar);
```

$$JB := \left[\left[\left(\frac{\partial^2}{\partial x^2}\,u(x,t)\right)-\left(\frac{\partial}{\partial t}\,u(x,t)\right)\right], [x,t], [u]\right]$$

```
> JB[2] := ivar;
```

$$JB_2 := [x, t=2]$$

```
> f := diff(u(x,t), x$3, t$4);
```

$$f := \frac{\partial^7}{\partial x^3\,\partial t^4}\,u(x,t)$$

```
> InvReduce(f, JB);
```

$$\frac{\partial^6}{\partial x \partial t^5} u(x, t)$$

See Also:

JanetBasis, PrincDeriv, ParamDeriv, CompCond, CompCondBasis, Resolution, SyzOp, HilbertSeries, AffEqn, AppOp.

Janet[JAdjoint] - return the adjoint operator of a linear differential operator

Calling Sequence:

JAdjoint(oper,ivar)

Parameters:

`oper` - differential operator in matrix form
`ivar` - list of independent variables

Description:

- For the linear differential operator **`oper`** given as a matrix with entries described below ***JAdjoint*** transposes **`oper`** and replaces any partial derivative by its negative, i. e. produces the adjoint operator.
- The entries of the matrix **`oper`** must have the form $[[c_1, [...]], ..., [c_r, [...]]]$, where the dots in the innermost brackets are sequences of independent variables from **`ivar`** representing partial derivatives w. r. t. these variables, and the c_i are the coefficients for these, i. e. functions in **`ivar`**. The whole bracket represents the sum of these items, i. e. the linear combination of the above operators. Note, if an innermost bracket is empty, it represents the identity operator. (For a convenient way to produce such a matrix use Diff2Op as in the example below.)

Examples:

```
> with(Janet):
> ivar := [x,y,z]; dvar:=[u,v];
```

$$ivar := [x, y, z]$$

$$dvar := [u, v]$$

```
> L := [[x^2+y, y^2+z], [z^3,y+1]];
```

$$L := [[x^2 + y, y^2 + z], [z^3, y + 1]]$$

```
> Pol2Diff(L, ivar, [u,v]);
```

$$\left[\left[\left(\frac{\partial}{\partial y}\mathrm{u}(x,y,z)\right)+\left(\frac{\partial^2}{\partial x^2}\mathrm{u}(x,y,z)\right)+\left(\frac{\partial}{\partial z}\mathrm{v}(x,y,z)\right)+\left(\frac{\partial^2}{\partial y^2}\mathrm{v}(x,y,z)\right)\right],\left[\left(\frac{\partial^3}{\partial z^3}\mathrm{u}(x,y,z)\right)+\mathrm{v}(x,y,z)+\left(\frac{\partial}{\partial y}\mathrm{v}(x,y,z)\right)\right]\right]$$

```
> mat := Diff2Op(%,ivar,dvar);
```

$$mat := \begin{bmatrix} [[1, [x, x]], [1, [y]]] & [[1, [y, y]], [1, [z]]] \\ [[1, [z, z, z]]] & [[1, [y]], [1, [\ ]]] \end{bmatrix}$$

```
> JAdjoint(mat,ivar);
```

$$\begin{bmatrix} [[1, [x, x]], [-1, [y]]] & [[-1, [z, z, z]]] \\ [[1, [y, y]], [-1, [z]]] & [[-1, [y]], [1, [\ ]]] \end{bmatrix}$$

See Also:

CmpOp, AppOp, AppOpInd, Diff2Op, Op2D, D2Op.

Janet[JanetBasis] - return the (unique) minimal Janet basis for a system of linear partial differential equations

Calling Sequence:

JanetBasis(L,ivar,dvar,oivar,ord,opt)

Parameters:

`L` - list of linear differential expressions
`ivar` - list of independent variables
`dvar` - list of dependent variables
`oivar` - (optional) list of the independent variables in varied order
`ord` - (optional) change of ordering for differential monomials
`opt` - (optional) equation specifying options for the computation

Description:

- ***JanetBasis*** returns the (unique) minimal Janet basis for a system of linear partial differential equations with respect to a certain ordering on differential monomials. The input equations are given in **`L`** by their left hand sides in the homogeneous case and by their left hand sides minus their right hand sides in the inhomogeneous case. ***JanetBasis*** is the most important procedure in the package Janet.
- The elements in **`L`** are expected to be linear differential expressions in the dependent variables **`dvar`** and the independent variables **`ivar`**. Most commonly each entry of the list **`L`** is created by diff.
- The output is a list containing the Janet basis as first entry, the parameters **`ivar`**, **`dvar`**, and **`oivar`** (optional) being appended. This output is taken as input for the commands InvReduce, SolSeries, and PolySol. Note, the same output can also be taken as input for the command Jpdesolv which simply calls ***pdesolv*** of the package ***Desolv***.
- The optional fourth parameter **`oivar`** is a permutation of **`ivar`**. It determines the ordering of the independent variables which is extended to an ordering for differential monomials that is used to define the highest term in a linear differential equation, cf. also the next paragraph. By default the ordering is according to **`ivar`**. Note that in any case, the ordering of the independent variables in **`ivar`** is the same as in arguments of functions in the given differential equations.
- For the optional fifth argument **`ord`** the values 1, 2, 3, or 4 are accepted. This argument determines the ordering of differential monomials in analogy to that for polynomial expressions, cf. InvolutiveBasis, and may have decisive influence on the output and on the running time. The default is 4, the degree reverse lexicographical ordering for the independent variables. 2 also selects the degree reverse lexicographical ordering for the independent variables. These two options differ in the case of several dependent variables: 2 gives higher priority to dependent variables occurring ealier in **`dvar`**, whereas 4 selects according to the chosen ordering of independent variables after selecting highest differentiation order first, and uses **`dvar`** only in case it finds the same highest terms for different dependent variables. 1 and 3 use pure lexicographical ordering instead, otherwise proceed as in 2 resp. 4. To save this information about the ordering in the output of ***JanetBasis*** and use it in the commands InvReduce, SolSeries, and PolySol, the user has to manually append **`ord`** as fourth entry in the output list. (**`ord`** is not appended by default due to compatibility for ***pdesolv***, see above.)
- In addition to the orderings described in the preceding paragraph, a block (elimination) ordering can be selected by partitioning the list of independent variables **`ivar`**. In this case the argument **`ivar`** is a list of lists ("blocks") of variables and **`ord`** is a list of natural numbers from 1 to 4 whose length equals the number of blocks. Two differential monomials are compared w.r.t. the block ordering as follows: The independent variables of the first block are examined first. If the according parts of the two differential monomials are different, the ordering specified by the first number in **`ord`** decides which monomial is greater. If the monomials are equal when considering only the first block of independent variables, then the ordering specified by the second number in **`ord`** is applied to the second block of independent variables, if the corresponding parts of the monomials are different, and so on (cf. Example 3 below). Note that in the case of more than one dependent variable, orderings 2 and 4 cannot be mixed.
- If "time"=*t* is given in **`opt`**, then *t* is expected to be a non-negative integer. In this case, the Janet basis computation is stopped after *t* seconds. If the program was not able to construct the result before *t* seconds, then a warning is printed.
- ***JanetBasis*** also generates a globally defined list ***COMPA*** which contains the compatibility conditions for the non-homogeneous case. More precisely, ***JanetBasis*** interprets every term *f***`(ivar)`** in **`L`** as a function of **`ivar`** and yields a list ***COMPA*** of relations in the *f* to

be satisfied for the system to be solvable.

- ***JanetBasis*** saves the information (in particular, the separation of the independent variables into multiplicative and non-multiplicative ones) in an internal data structure which can be displayed by PrincDeriv.
- Another global variable used in ***JanetBasis*** is **`Nu_St`**, providing information by which function the alorithm has divided certain equations, cf. ZeroSets for details.
- It is possible to assign degrees other than 1 to the independent variables and degrees other than 0 to the dependent variables (cf. Example 2 below). Therefore the declaration of the dependent and independent variables in **`dvar`** and **`ivar`** resp. become equations: **`dvar`** has to be given in the format $[u_1=e_1, ..., u_k=e_k]$, where e_i are integers, and the format of **`ivar`** is then $[x_1=d_1, ..., x_n=d_n]$ with natural numbers d_i. Of course the Janet basis will in general be different from the standard one. The same applies to the induced filtration. To use these non-standard degrees in InvReduce, SolSeries or PolySol, one has to replace the second and third entry of the output list of ***JanetBasis*** by **`dvar`** and **`ivar`** resp. Instead of HilbertSeries one has to use WeightedHilbertSeries.
- For a description of the basic algorithms, see V. P. Gerdt, "Involutive Algorithms for Computing Groebner Bases", in: S. Cojocaru, G. Pfister, V. Ufnarovski, "Computational Commutative and Non-Commutative Algebraic Geometry", NATO Science Series, IOS Press, 2005, pp. 199-225; or V. P. Gerdt, "Involutive Division Technique: Some Generalizations and Optimizations", Journal of Mathematical Sciences 108(6), 2002, pp. 1034-1051 (cf. Janet).

Examples:

```
> with(Janet):
```

Example 1: Janet's example

```
> ivar := [x,y,t]: dvar := [u]:
> L := [diff(u(x,y,t),x$2) - y*diff(u(x,y,t),t$2), diff(u(x,y,t),y$2)];
```

$$L := \left[\left(\frac{\partial^2}{\partial x^2}\,u(x,y,t)\right) - y\left(\frac{\partial^2}{\partial t^2}\,u(x,y,t)\right), \frac{\partial^2}{\partial y^2}\,u(x,y,t)\right]$$

```
> JB := JanetBasis(L, ivar, dvar);
```

$$JB := \left[\left[\frac{\partial^2}{\partial y^2}\,u(x,y,t), \left(\frac{\partial^2}{\partial x^2}\,u(x,y,t)\right) - y\left(\frac{\partial^2}{\partial t^2}\,u(x,y,t)\right), \frac{\partial^3}{\partial y\,\partial t^2}\,u(x,y,t), \frac{\partial^3}{\partial y^2\,\partial x}\,u(x,y,t), \frac{\partial^4}{\partial t^4}\,u(x,y,t), \frac{\partial^4}{\partial y\,\partial x\,\partial t^2}\,u(x,y,t), \frac{\partial^5}{\partial x\,\partial t^4}\,u(x,y,t)\right], [x,y,t], [u]\right]$$

```
> PrincDeriv();
```

$$\left[\frac{\partial^2}{\partial y^2}\,u(x,y,t), [*,y,t], \frac{\partial^2}{\partial y^2}\,u(x,y,t)\right]$$

$$\left[\left(\frac{\partial^2}{\partial x^2}\,u(x,y,t)\right) - y\left(\frac{\partial^2}{\partial t^2}\,u(x,y,t)\right), [x,y,t], \frac{\partial^2}{\partial x^2}\,u(x,y,t)\right]$$

$$\left[\frac{\partial^3}{\partial y\,\partial t^2}\,u(x,y,t), [*,*,t], \frac{\partial^3}{\partial y\,\partial t^2}\,u(x,y,t)\right]$$

$$\left[\frac{\partial^3}{\partial y^2\,\partial x}\,u(x,y,t), [*,y,t], \frac{\partial^3}{\partial y^2\,\partial x}\,u(x,y,t)\right]$$

$$\left[\frac{\partial^4}{\partial t^4}\,u(x,y,t), [*,*,t], \frac{\partial^4}{\partial t^4}\,u(x,y,t)\right]$$

$$\left[\frac{\partial^4}{\partial y\,\partial x\,\partial t^2}\,u(x,y,t), [*,*,t], \frac{\partial^4}{\partial y\,\partial x\,\partial t^2}\,u(x,y,t)\right]$$

$$\left[\frac{\partial^5}{\partial x\,\partial t^4}\,u(x,y,t), [*,*,t], \frac{\partial^5}{\partial x\,\partial t^4}\,u(x,y,t)\right]$$

Example 2: Using two dependent variables and right hand sides

```
> ivar := [x,y,t]: dvar := [u,v]:
> L := [diff(u(x,y,t),x,x) - y*diff(v(x,y,t) - a(x,y,t),t,t), diff(v(x,y,t),y,y) -
```

```
b(x,y,t),diff(v(x,y,t),y,y)-y*(diff(u(x,y,t),t,t)-diff(c(x,y,t),t,t))];
```

$$L := \left[\left(\frac{\partial^2}{\partial x^2} u(x,y,t)\right) - y\left(\left(\frac{\partial^2}{\partial t^2} v(x,y,t)\right) - \left(\frac{\partial^2}{\partial t^2} a(x,y,t)\right)\right), \left(\frac{\partial^2}{\partial y^2} v(x,y,t)\right) - b(x,y,t), \left(\frac{\partial^2}{\partial y^2} v(x,y,t)\right) - y\left(\left(\frac{\partial^2}{\partial t^2} u(x,y,t)\right) - \left(\frac{\partial^2}{\partial t^2} c(x,y,t)\right)\right)\right]$$

```
> JanetBasis(L, ivar, dvar);
```

$$\left[\left[\left[\left(\frac{\partial^2}{\partial t^2} u(x,y,t)\right) - \frac{b(x,y,t) + y\left(\frac{\partial^2}{\partial t^2} c(x,y,t)\right)}{y}, \left(\frac{\partial^2}{\partial y^2} v(x,y,t)\right) - b(x,y,t), \left(\frac{\partial^2}{\partial x^2} u(x,y,t)\right) - y\left(\frac{\partial^2}{\partial t^2} v(x,y,t)\right) + y\left(\frac{\partial^2}{\partial t^2} a(x,y,t)\right)\right],\right.$$
$$\left[\left(\frac{\partial^3}{\partial x \partial t^2} u(x,y,t)\right) - \frac{\left(\frac{\partial}{\partial x} b(x,y,t)\right) + y\left(\frac{\partial^3}{\partial t^2 \partial x} c(x,y,t)\right)}{y}, \left(\frac{\partial^4}{\partial t^4} v(x,y,t)\right) - \frac{y^2\left(\frac{\partial^4}{\partial t^4} a(x,y,t)\right) + \left(\frac{\partial^2}{\partial x^2} b(x,y,t)\right) + y\left(\frac{\partial^4}{\partial t^2 \partial x^2} c(x,y,t)\right)}{y^2},\right.$$
$$\left.\left.\left(\frac{\partial^5}{\partial y \partial t^4} v(x,y,t)\right) - \frac{\left(\frac{\partial^5}{\partial t^4 \partial y} a(x,y,t)\right) y^3 + \left(\frac{\partial^3}{\partial y \partial x^2} b(x,y,t)\right) y - y\left(\frac{\partial^4}{\partial t^2 \partial x^2} c(x,y,t)\right) + \left(\frac{\partial^5}{\partial t^2 \partial y \partial x^2} c(x,y,t)\right) y^2 - 2\left(\frac{\partial^2}{\partial x^2} b(x,y,t)\right)}{y^3}\right], [x,y,t],\right.$$
$$\left.[u,v]\right]$$

Here are the compatibility conditions for the right hand side ($a(x,y,t)$, $b(x,y,t)$, $c(x,y,t)$):

```
> COMPA;
```

$$\left[-\left(\frac{\partial^4}{\partial t^4} b(x,y,t)\right) y^4 + \left(\frac{\partial^6}{\partial y^2 \partial t^4} a(x,y,t)\right) y^4 + \left(\frac{\partial^4}{\partial y^2 \partial x^2} b(x,y,t)\right) y^2 - 4\left(\frac{\partial^3}{\partial y \partial x^2} b(x,y,t)\right) y + 2y\left(\frac{\partial^4}{\partial x^2 \partial t^2} c(x,y,t)\right)\right.$$
$$\left. - 2\left(\frac{\partial^5}{\partial y \partial x^2 \partial t^2} c(x,y,t)\right) y^2 + \left(\frac{\partial^6}{\partial y^2 \partial x^2 \partial t^2} c(x,y,t)\right) y^3 + 6\left(\frac{\partial^2}{\partial x^2} b(x,y,t)\right)\right]$$

We rerun the last example with ordering $y > x > t$ and position over term ordering:

```
> JanetBasis(L, ivar, dvar, [y,x,t], 2);
```

$$\left[\left[\left[\left(\frac{\partial^2}{\partial y^2} v(x,y,t)\right) - b(x,y,t), \left(\frac{\partial^4}{\partial t^4} v(x,y,t)\right) - \frac{y^2\left(\frac{\partial^4}{\partial t^4} a(x,y,t)\right) + \left(\frac{\partial^2}{\partial x^2} b(x,y,t)\right) + y\left(\frac{\partial^4}{\partial t^2 \partial x^2} c(x,y,t)\right)}{y^2},\right.\right.$$
$$\left.\left(\frac{\partial^5}{\partial y \partial t^4} v(x,y,t)\right) + \frac{-\left(\frac{\partial^5}{\partial t^4 \partial y} a(x,y,t)\right) y^3 - \left(\frac{\partial^3}{\partial x^2 \partial y} b(x,y,t)\right) y + y\left(\frac{\partial^4}{\partial t^2 \partial x^2} c(x,y,t)\right) - \left(\frac{\partial^5}{\partial t^2 \partial x^2 \partial y} c(x,y,t)\right) y^2 + 2\left(\frac{\partial^2}{\partial x^2} b(x,y,t)\right)}{y^3},\right.$$
$$\left.\left(\frac{\partial^2}{\partial t^2} u(x,y,t)\right) - \frac{b(x,y,t) + y\left(\frac{\partial^2}{\partial t^2} c(x,y,t)\right)}{y}, \left(\frac{\partial^2}{\partial x^2} u(x,y,t)\right) - y\left(\frac{\partial^2}{\partial t^2} v(x,y,t)\right) + y\left(\frac{\partial^2}{\partial t^2} a(x,y,t)\right)\right.$$
$$\left.\left.\left(\frac{\partial^3}{\partial x \partial t^2} u(x,y,t)\right) - \frac{\left(\frac{\partial}{\partial x} b(x,y,t)\right) + y\left(\frac{\partial^3}{\partial t^2 \partial x} c(x,y,t)\right)}{y}\right], [x,y,t], [u,v], [y,x,t]\right]$$

We rerun the last example with the usual order modified by assigning degrees to the independent variables:

```
> JanetBasis(L, [x=1,y=2,t=3], dvar);
```

$$\left[\left[\left[\left(\frac{\partial^2}{\partial y^2} v(x,y,t)\right) - b(x,y,t), -\left(\frac{\partial^2}{\partial x^2} u(x,y,t)\right) + y\left(\frac{\partial^2}{\partial t^2} v(x,y,t)\right) - y\left(\frac{\partial^2}{\partial t^2} a(x,y,t)\right)\left(\frac{\partial^2}{\partial t^2} u(x,y,t)\right) - \frac{b(x,y,t) + y\left(\frac{\partial^2}{\partial t^2} c(x,y,t)\right)}{y},\right.\right.$$
$$\left.\left. -2\frac{\frac{\partial^3}{\partial y \partial x^2} u(x,y,t)}{y} + 2\frac{\frac{\partial^2}{\partial x^2} u(x,y,t)}{y^2} + \left(\frac{\partial^4}{\partial y^2 \partial x^2} u(x,y,t)\right) - \left(\frac{\partial^2}{\partial t^2} b(x,y,t)\right) y + y\left(\frac{\partial^4}{\partial t^2 \partial y^2} a(x,y,t)\right)\right.$$

$$\left[\left(\frac{\partial^3}{\partial x\,\partial t^2}\,u(x,y,t)\right)-\frac{\left(\frac{\partial}{\partial x}\,b(x,y,t)\right)+y\left(\frac{\partial^3}{\partial t^2\,\partial x}\,c(x,y,t)\right)}{y},-\left(\frac{\partial^3}{\partial y\,\partial x^2}\,u(x,y,t)\right)+\frac{\frac{\partial^2}{\partial x^2}\,u(x,y,t)}{y}+y\left(\frac{\partial^3}{\partial y\,\partial t^2}\,v(x,y,t)\right)-y\left(\frac{\partial^3}{\partial t^2\,\partial y}\,a(x,y,t)\right),\right.$$
$$\left.\left(\frac{\partial^4}{\partial x^2\,\partial t^2}\,u(x,y,t)\right)-\frac{\left(\frac{\partial^2}{\partial x^2}\,b(x,y,t)\right)+y\left(\frac{\partial^4}{\partial t^2\,\partial x^2}\,c(x,y,t)\right)}{y},\right.$$
$$\left.\left(\frac{\partial^5}{\partial y\,\partial x^2\,\partial t^2}\,u(x,y,t)\right)-\frac{\left(\frac{\partial^3}{\partial y\,\partial x^2}\,b(x,y,t)\right)y+\left(\frac{\partial^5}{\partial t^2\,\partial y\,\partial x^2}\,c(x,y,t)\right)y^2-\left(\frac{\partial^2}{\partial x^2}\,b(x,y,t)\right)}{y^2}\right],[x,y,t],[u,v]\Bigg]$$

Example 3: Block ordering

```
> ivar := [x,y,z]; dvar := [u];
```

$$ivar := [x, y, z]$$

$$dvar := [u]$$

```
> L := Pol2Diff([x*y-z^3, x*y*z-x^2*y^2], ivar, dvar);
```

$$L := \left[\left(\frac{\partial^2}{\partial y\,\partial x}\,u(x,y,z)\right)-\left(\frac{\partial^3}{\partial z^3}\,u(x,y,z)\right),\left(\frac{\partial^3}{\partial z\,\partial y\,\partial x}\,u(x,y,z)\right)-\left(\frac{\partial^4}{\partial y^2\,\partial x^2}\,u(x,y,z)\right)\right]$$

```
> JanetBasis(L, ivar, dvar);
```

$$\left[\left[\left(\frac{\partial^3}{\partial z^3}\,u(x,y,z)\right)-\left(\frac{\partial^2}{\partial y\,\partial x}\,u(x,y,z)\right),\left(\frac{\partial^4}{\partial z^3\,\partial x}\,u(x,y,z)\right)-\left(\frac{\partial^3}{\partial y\,\partial x^2}\,u(x,y,z)\right),\left(\frac{\partial^4}{\partial y^2\,\partial x^2}\,u(x,y,z)\right)-\left(\frac{\partial^3}{\partial z\,\partial y\,\partial x}\,u(x,y,z)\right),\right.\right.$$
$$\left.\left.-\left(\frac{\partial^4}{\partial y\,\partial x^3}\,u(x,y,z)\right)+\left(\frac{\partial^5}{\partial z^3\,\partial x^2}\,u(x,y,z)\right),-\left(\frac{\partial^4}{\partial z\,\partial y\,\partial x^2}\,u(x,y,z)\right)+\left(\frac{\partial^6}{\partial z^3\,\partial y\,\partial x^2}\,u(x,y,z)\right)\right],[x,y,z],[u]\right]$$

```
> JanetBasis(L, [[x,y],[z]], dvar, [4,4]);
```

$$\left[\left[\left(\frac{\partial^6}{\partial z^6}\,u(x,y,z)\right)-\left(\frac{\partial^4}{\partial z^4}\,u(x,y,z)\right),\left(\frac{\partial^7}{\partial z^6\,\partial x}\,u(x,y,z)\right)-\left(\frac{\partial^5}{\partial z^4\,\partial x}\,u(x,y,z)\right),\left(\frac{\partial^2}{\partial y\,\partial x}\,u(x,y,z)\right)-\left(\frac{\partial^3}{\partial z^3}\,u(x,y,z)\right)\right],[x,y,z],[u]\right]$$

See Also:

JanetOptions, PrincDeriv, Denominators, ZeroSets, InvReduce, ParamDeriv, CompCond, CompCondBasis, Resolution, HilbertSeries, CartanCharacter, SolSeries, PolySol, AssertJanetBasis, Pol2Diff.

Janet[JanetOptions] - set up the options for the current session of Janet

Calling Sequence:

JanetOptions(s,v)

Parameters:

s - string specifying the option to be affected
v - (optional) the option's new value

Description:

- ***JanetOptions*** sets up the options for the current session of Janet. The string **s** specifies the option which is to be modified. Up to now, there are only three possible values for **s**, namely:

```
"JanetLike"       "matrix"       "criteria"
```

- If **s** equals "JanetLike", then **v** is expected to be either true or false. If **v** is true then the command JanetBasis returns a Janet-like Groebner basis instead of an involutive basis. The default setting is false. The return value of ***JanetOptions*** is the former value of the option "JanetLike".
- If the parameter **s** is the string "matrix", then **v** is expected to be either the symbol 'matrix' or the symbol 'Matrix'. This option determines the type for matrices returned by the procedure InvReduce (cf. Example 2 below). This option is meaningful only for Maple versions that provide both the linalg and the LinearAlgebra package. The default value for this option is the symbol 'Matrix'.
- If **s** equals the string "criteria", then **v** is expected to be a list consisting of some of the integers 2, 3, 4 (or being the empty list). If the integer *i* is present in **v**, then involutive basis computations during the current session of Janet will apply the *i*-th involutive criterion to avoid unnecessary reductions. For more details about the involutive criteria, see V. P. Gerdt, D. A. Yanovich, "Experimental Analysis of Involutive Criteria", in: A. Dolzmann, A. Seidl, T. Sturm (eds.), Algorithmic Algebra and Logic, BOD Norderstedt, pp. 105-109.

Examples:

```
> with(Janet):
```

Example 1: Computing Janet-like Groebner bases

(see V. P. Gerdt, Y. A. Blinkov, *Janet-like Groebner Bases*, Proceedings of Computer Algebra in Scientific Computing, Springer, 2005, pp. 184-195)

```
> L := Pol2Diff([x^7-y^2*z, x^4*w-y^3, x^3*y-z*w], [x,y,z,w], [u]);
```

$$L := \left[-\left(\frac{\partial^3}{\partial z\,\partial y^2}u(x,y,z,w)\right)+\left(\frac{\partial^7}{\partial x^7}u(x,y,z,w)\right)\left(\frac{\partial^5}{\partial x^4\,\partial w}u(x,y,z,w)\right)-\left(\frac{\partial^3}{\partial y^3}u(x,y,z,w)\right)\left(\frac{\partial^4}{\partial y\,\partial x^3}u(x,y,z,w)\right)-\left(\frac{\partial^2}{\partial z\,\partial w}u(x,y,z,w)\right)\right]$$

```
> JanetOptions("JanetLike", true);
```

false

```
> JanetBasis(L, [x,y,z,w], [u]);
```

$$\left[\left[-\left(\frac{\partial^4}{\partial z\,\partial x\,\partial w^2}u(x,y,z,w)\right)+\left(\frac{\partial^4}{\partial y^4}u(x,y,z,w)\right)\left(\frac{\partial^4}{\partial y\,\partial x^3}u(x,y,z,w)\right)-\left(\frac{\partial^2}{\partial z\,\partial w}u(x,y,z,w)\right)\left(\frac{\partial^5}{\partial x^4\,\partial w}u(x,y,z,w)\right)-\left(\frac{\partial^3}{\partial y^3}u(x,y,z,w)\right)\right.\right.$$
$$\left.\left.-\left(\frac{\partial^3}{\partial z\,\partial x\,\partial w}u(x,y,z,w)\right)+\left(\frac{\partial^5}{\partial y\,\partial x^4}u(x,y,z,w)\right)-\left(\frac{\partial^3}{\partial z\,\partial y^2}u(x,y,z,w)\right)+\left(\frac{\partial^7}{\partial x^7}u(x,y,z,w)\right)\right],[x,y,z,w],[u]\right]$$

```
> JanetOptions("JanetLike", false);
```

true

```
> JanetBasis(L, [x,y,z,w], [u]);
```

$$\left[\left[-\left(\frac{\partial^4}{\partial z\,\partial x\,\partial w^2}u(x,y,z,w)\right)+\left(\frac{\partial^4}{\partial y^4}u(x,y,z,w)\right)\left(\frac{\partial^4}{\partial y\,\partial x^3}u(x,y,z,w)\right)-\left(\frac{\partial^2}{\partial z\,\partial w}u(x,y,z,w)\right)\left(\frac{\partial^5}{\partial x^4\,\partial w}u(x,y,z,w)\right)-\left(\frac{\partial^3}{\partial y^3}u(x,y,z,w)\right)\right.\right.$$
$$\left.\left.-\left(\frac{\partial^5}{\partial z\,\partial x^2\,\partial w^2}u(x,y,z,w)\right)+\left(\frac{\partial^5}{\partial y^4\,\partial x}u(x,y,z,w)\right)-\left(\frac{\partial^3}{\partial z\,\partial x\,\partial w}u(x,y,z,w)\right)+\left(\frac{\partial^5}{\partial y\,\partial x^4}u(x,y,z,w)\right)\right.$$

$$\left(\frac{\partial^6}{\partial x^5\,\partial w}\,u(x,y,z,w)\right)-\left(\frac{\partial^4}{\partial y^3\,\partial x}\,u(x,y,z,w)\right),-\left(\frac{\partial^6}{\partial z\,\partial x^3\,\partial w^2}\,u(x,y,z,w)\right)+\left(\frac{\partial^6}{\partial y^4\,\partial x^2}\,u(x,y,z,w)\right),$$
$$-\left(\frac{\partial^4}{\partial z\,\partial x^2\,\partial w}\,u(x,y,z,w)\right)+\left(\frac{\partial^6}{\partial y\,\partial x^5}\,u(x,y,z,w)\right),\left(\frac{\partial^7}{\partial x^6\,\partial w}\,u(x,y,z,w)\right)-\left(\frac{\partial^5}{\partial y^3\,\partial x^2}\,u(x,y,z,w)\right),$$
$$-\left(\frac{\partial^5}{\partial z\,\partial x^3\,\partial w}\,u(x,y,z,w)\right)+\left(\frac{\partial^7}{\partial y\,\partial x^6}\,u(x,y,z,w)\right),-\left(\frac{\partial^3}{\partial z\,\partial y^2}\,u(x,y,z,w)\right)+\left(\frac{\partial^7}{\partial x^7}\,u(x,y,z,w)\right)\Bigg],[x,y,z,w],[u]\Bigg]$$

```
> DiffGroebnerBasis(L, [x,y,z,w], [u]);
```

$$\Bigg[\Bigg[-\left(\frac{\partial^4}{\partial z\,\partial x\,\partial w^2}\,u(x,y,z,w)\right)+\left(\frac{\partial^4}{\partial y^4}\,u(x,y,z,w)\right)\left(\frac{\partial^4}{\partial y\,\partial x^3}\,u(x,y,z,w)\right)-\left(\frac{\partial^2}{\partial z\,\partial w}\,u(x,y,z,w)\right)\left(\frac{\partial^5}{\partial x^4\,\partial w}\,u(x,y,z,w)\right)-\left(\frac{\partial^3}{\partial y^3}\,u(x,y,z,w)\right),$$
$$-\left(\frac{\partial^3}{\partial z\,\partial y^2}\,u(x,y,z,w)\right)+\left(\frac{\partial^7}{\partial x^7}\,u(x,y,z,w)\right)\Bigg],[x,y,z,w],[u]\Bigg]$$

Example 2: Changing the matrix type of results of procedures

```
> ivar := [x,y]; dvar := [u];
```

$$ivar := [x, y]$$
$$dvar := [u]$$

```
> J := JanetBasis([diff(u(x,y),x,y)+diff(u(x,y),y), diff(u(x,y),x,x,y,y)], ivar, dvar);
```

$$J := \Bigg[\Bigg[\frac{\partial^2}{\partial y^2}\,u(x,y),\left(\frac{\partial^2}{\partial y\,\partial x}\,u(x,y)\right)+\left(\frac{\partial}{\partial y}\,u(x,y)\right)\Bigg],[x,y],[u]\Bigg]$$

```
> JanetOptions("matrix", matrix);
```

Matrix

```
> InvReduce(diff(u(x,y),x,x,y,y), J, "C");
```

$$[0, [[[1, [\,]]]\quad [[1, [x, y]], [-1, [y]]]]$$

```
> whattype(%[2]);
```

array

```
> JanetOptions("matrix", Matrix);
```

matrix

```
> InvReduce(diff(u(x,y),x,x,y,y), J, "C");
```

$$[0, [[[1, [\,]]]\quad [[1, [x, y]], [-1, [y]]]]$$

```
> whattype(%[2]);
```

Matrix

See Also:

JanetBasis, JanetStats, PrincDeriv, InvReduce, ParamDeriv, HilbertSeries, CompCond, CompCondBasis, DiffGroebnerBasis, Pol2Diff.

Janet[JanetStats] - display statistics of last application of JanetBasis

Calling Sequence:

JanetStats()

Parameters:

- - none (assumes that JanetBasis has been called before)

Description:

- ***JanetStats*** displays statistical information about the last run of ***JanetBasis***.

Examples:

```
> with(Janet):
> ivar := [x1,x2,x3,x4]; dvar := [u];
```

$$ivar := [x1, x2, x3, x4]$$

$$dvar := [u]$$

```
> L := Pol2Diff([x1+x2+x3+x4, x1*x2+x2*x3+x3*x4+x4*x1,
  x1*x2*x3+x2*x3*x4+x3*x4*x1+x4*x1*x2, x1*x2*x3*x4-1], ivar, dvar);
```

$$L := \left[\left(\frac{\partial}{\partial x1}u(x1,x2,x3,x4)\right)+\left(\frac{\partial}{\partial x4}u(x1,x2,x3,x4)\right)+\left(\frac{\partial}{\partial x3}u(x1,x2,x3,x4)\right)+\left(\frac{\partial}{\partial x2}u(x1,x2,x3,x4)\right),\right.$$
$$\left(\frac{\partial^2}{\partial x4\,\partial x3}u(x1,x2,x3,x4)\right)+\left(\frac{\partial^2}{\partial x4\,\partial x1}u(x1,x2,x3,x4)\right)+\left(\frac{\partial^2}{\partial x2\,\partial x1}u(x1,x2,x3,x4)\right)+\left(\frac{\partial^2}{\partial x3\,\partial x2}u(x1,x2,x3,x4)\right),$$
$$\left(\frac{\partial^3}{\partial x3\,\partial x2\,\partial x1}u(x1,x2,x3,x4)\right)+\left(\frac{\partial^3}{\partial x4\,\partial x2\,\partial x1}u(x1,x2,x3,x4)\right)+\left(\frac{\partial^3}{\partial x4\,\partial x3\,\partial x2}u(x1,x2,x3,x4)\right)+\left(\frac{\partial^3}{\partial x4\,\partial x3\,\partial x1}u(x1,x2,x3,x4)\right),$$
$$\left.\left(\frac{\partial^4}{\partial x4\,\partial x3\,\partial x2\,\partial x1}u(x1,x2,x3,x4)\right)-u(x1,x2,x3,x4)\right]$$

```
> JanetBasis(L, ivar, dvar);
```

$$\left[\left[\left(\frac{\partial}{\partial x1}u(x1,x2,x3,x4)\right)+\left(\frac{\partial}{\partial x4}u(x1,x2,x3,x4)\right)+\left(\frac{\partial}{\partial x3}u(x1,x2,x3,x4)\right)+\left(\frac{\partial}{\partial x2}u(x1,x2,x3,x4)\right),\right.\right.$$
$$2\left(\frac{\partial^2}{\partial x4\,\partial x2}u(x1,x2,x3,x4)\right)+\left(\frac{\partial^2}{\partial x4^2}u(x1,x2,x3,x4)\right)+\left(\frac{\partial^2}{\partial x2^2}u(x1,x2,x3,x4)\right),$$
$$\left(\frac{\partial^3}{\partial x4\,\partial x3^2}u(x1,x2,x3,x4)\right)-\left(\frac{\partial^3}{\partial x4^2\,\partial x2}u(x1,x2,x3,x4)\right)-\left(\frac{\partial^3}{\partial x4^3}u(x1,x2,x3,x4)\right)+\left(\frac{\partial^3}{\partial x3^2\,\partial x2}u(x1,x2,x3,x4)\right),$$
$$\left(\frac{\partial^4}{\partial x4^2\,\partial x3^2}u(x1,x2,x3,x4)\right)+\left(\frac{\partial^4}{\partial x4^3\,\partial x3}u(x1,x2,x3,x4)\right)-\left(\frac{\partial^4}{\partial x4^4}u(x1,x2,x3,x4)\right)+\left(\frac{\partial^4}{\partial x4^2\,\partial x3\,\partial x2}u(x1,x2,x3,x4)\right)$$
$$-u(x1,x2,x3,x4)-\left(\frac{\partial^4}{\partial x4^3\,\partial x2}u(x1,x2,x3,x4)\right),$$
$$-\left(\frac{\partial}{\partial x2}u(x1,x2,x3,x4)\right)-\left(\frac{\partial}{\partial x4}u(x1,x2,x3,x4)\right)+\left(\frac{\partial^5}{\partial x4^5}u(x1,x2,x3,x4)\right)+\left(\frac{\partial^5}{\partial x4^4\,\partial x2}u(x1,x2,x3,x4)\right),$$
$$-\left(\frac{\partial}{\partial x3}u(x1,x2,x3,x4)\right)-\left(\frac{\partial}{\partial x4}u(x1,x2,x3,x4)\right)+\left(\frac{\partial^5}{\partial x4^2\,\partial x3^3}u(x1,x2,x3,x4)\right)+\left(\frac{\partial^5}{\partial x4^3\,\partial x3^2}u(x1,x2,x3,x4)\right),$$
$$-\left(\frac{\partial^2}{\partial x4\,\partial x2}u(x1,x2,x3,x4)\right)-2\left(\frac{\partial^2}{\partial x4^2}u(x1,x2,x3,x4)\right)+\left(\frac{\partial^2}{\partial x3\,\partial x2}u(x1,x2,x3,x4)\right)+\left(\frac{\partial^6}{\partial x4^4\,\partial x3^2}u(x1,x2,x3,x4)\right)$$
$$\left.\left.+\left(\frac{\partial^2}{\partial x4\,\partial x3}u(x1,x2,x3,x4)\right)\right],[x1,x2,x3,x4],[u]\right]$$

```
> JanetStats();
```

Number of elements of involutive basis, 7

Use of normal form procedure, 41
Number of reductions performed, 94
Number of transfers, 0
Use of second criterion, 0
Use of third criterion, 0
Use of fourth criterion, 0
The involutive basis is also a reduced Groebner basis.

See Also:

JanetBasis, JanetOptions, PrincDeriv, Pol2Diff.

Janet[Jpdesolv] - invoke *pdesolv* from the package *Desolv*

Calling Sequence:

Jpdesolv(L,ivar,dvar)

Parameters:

`L`	-	list of linear differential expressions
`ivar`	-	list of independent variables
`dvar`	-	list of dependent variables

Description:

- ***Jpdesolv*** simplifies the simultaneous usage of the Maple packages Janet and ***Desolv***. It simply calls ***pdesolv***(**L**, **dvar**, **ivar**), where each entry *u* of **dvar** is replaced by *u*(op(**ivar**)) if necessary.
- The command produces solutions of the PDE system given by **L**.
- The main objective of this extra command is the switching of **ivar** and **dvar** in the list of parameters so that the output of JanetBasis can used as input for the solver immediately.
- Note also that ParamDeriv allows a qualitative check, whether ***pdesolv*** has found all solutions.

Examples:

```
> with(Janet):
> with(desolv):
> ivar := [x,y]; dvar := [u,v];
```

$$ivar := [x, y]$$

$$dvar := [u, v]$$

```
> L := [diff(u(x,y),x) - diff(v(x,y),y), diff(u(x,y),x$2),
  diff(v(x,y),x,y)-diff(u(x,y),y$2)];
```

$$L := \left[\left(\frac{\partial}{\partial x}\mathrm{u}(x,y)\right)-\left(\frac{\partial}{\partial y}\mathrm{v}(x,y)\right), \frac{\partial^2}{\partial x^2}\mathrm{u}(x,y), \left(\frac{\partial^2}{\partial y\,\partial x}\mathrm{v}(x,y)\right)-\left(\frac{\partial^2}{\partial y^2}\mathrm{u}(x,y)\right)\right]$$

```
> J := JanetBasis(L, ivar, dvar);
```

$$J := \left[\left[\left(\frac{\partial}{\partial x}\mathrm{u}(x,y)\right)-\left(\frac{\partial}{\partial y}\mathrm{v}(x,y)\right), \frac{\partial^2}{\partial y^2}\mathrm{u}(x,y), \frac{\partial^2}{\partial y\,\partial x}\mathrm{v}(x,y), \frac{\partial^3}{\partial y^3}\mathrm{v}(x,y)\right], [x, y], [u, v]\right]$$

```
> Jpdesolv(op(J));
```

$$[[\,], [\,], [\mathrm{u}(x,y) = C\_2\,x - C\_4 + 2\,y\,C\_3\,x - y\,C\_5, \mathrm{v}(x,y) = C\_1 + y\,C\_2 + y^2\,C\_3 + \mathrm{F\_4}(x)], [\mathrm{F\_4}(x), C\_1, C\_2, C\_3, C\_4, C\_5]]$$

Part of this general solution, namely the polynomial solutions up to some degree, say 3, can also be obtained by the Janet package command ***PolySol***.

```
> PolySol(J, 3);
```

$$\left[\mathrm{u}(x,y) = C1_{0,0} + C2_{0,1}\,x + C1_{0,1}\,y + C2_{0,2}\,xy, \mathrm{v}(x,y) = C2_{0,0} + C2_{1,0}\,x + C2_{0,1}\,y + \frac{1}{2}C2_{2,0}\,x^2 + \frac{1}{2}C2_{0,2}\,y^2 + \frac{1}{6}C2_{3,0}\,x^3\right],$$

$$[C1_{0,0}, C2_{0,1}, C1_{0,1}, C2_{0,2}, C2_{0,0}, C2_{1,0}, C2_{2,0}, C2_{3,0}]$$

See Also:

JanetBasis, PrincDeriv, ParamDeriv, SolSeries, PolySol, ZeroSets.

Janet[LeadingDeriv] - map each element of a list to its leading derivative

Calling Sequence:

LeadingDeriv(L,ivar,dvar,oivar,ord,cf)

Parameters:

`L`	-	list of linear differential expressions
`ivar`	-	list of independent variables
`dvar`	-	list of dependent variables
`oivar`	-	(optional) list of the independent variables in varied order
`ord`	-	(optional) change of ordering for differential expressions
`cf`	-	(optional) string "C", return leading derivatives with coefficients

Description:

- ***LeadingDeriv*** returns the leading derivatives of the elements in **`L`** w. r. t. the specified ordering.
- For a description of all possible values of the parameters **`ivar`**, **`dvar`**, **`oivar`**, and **`ord`** see the corresponding explanations in JanetBasis.
- If the string "C" is given as optional parameter **`cf`**, then the output list is the list of leading derivatives of the elements in **`L`** including their leading coefficients.
- The command is analogous to the command LeadingMonomial in the polynomial case.

Examples:

```
> with(Janet):
> ivar := [x,y,z]; dvar := [u];
```

$$ivar := [x, y, z]$$

$$dvar := [u]$$

```
> L := [x*diff(u(x,y,z),x,y,z)+y^2*diff(u(x,y,z),x,z$3),
  2*diff(u(x,y,z),y$2)+diff(u(x,y,z),x)];
```

$$L := \left[x\left(\frac{\partial^3}{\partial z\,\partial y\,\partial x}\mathrm{u}(x,y,z)\right)+y^2\left(\frac{\partial^4}{\partial z^3\,\partial x}\mathrm{u}(x,y,z)\right), 2\left(\frac{\partial^2}{\partial y^2}\mathrm{u}(x,y,z)\right)+\left(\frac{\partial}{\partial x}\mathrm{u}(x,y,z)\right)\right]$$

```
> LeadingDeriv(L, ivar, dvar);
```

$$\left[\frac{\partial^4}{\partial z^3\,\partial x}\mathrm{u}(x,y,z), \frac{\partial^2}{\partial y^2}\mathrm{u}(x,y,z)\right]$$

```
> LeadingDeriv(L, ivar, dvar, 1);
```

$$\left[\frac{\partial^3}{\partial z\,\partial y\,\partial x}\mathrm{u}(x,y,z), \frac{\partial}{\partial x}\mathrm{u}(x,y,z)\right]$$

```
> LeadingDeriv(L, ivar, dvar, "C");
```

$$\left[y^2\left(\frac{\partial^4}{\partial z^3\,\partial x}\mathrm{u}(x,y,z)\right), 2\left(\frac{\partial^2}{\partial y^2}\mathrm{u}(x,y,z)\right)\right]$$

See Also:

JanetBasis, PrincDeriv, LeadingMonomial.

Janet[LeftInverse] - compute a left inverse of a linear differential operator

Calling Sequence:

LeftInverse(M,ivar,dvar,rvar)

Parameters:

`M` - differential operator in matrix form or list of linear differential expressions
`ivar` - list of independent variables
`dvar` - (optional) list of dependent variables
`rvar` - (optional) list of dependent variables for the left inverse

Description:

- ***LeftInverse*** computes (if possible) a left inverse of a linear differential operator **`M`**, i.e. a linear differential operator L such that the composition of L and **`M`** is the identity.
- The first parameter **`M`** is expected to be a linear differential operator in matrix form (cf. AppOp) with independent variables **`ivar`** or a list of linear differential expressions with independent variables **`ivar`** and dependent variables **`dvar`**. In the second case, ***LeftInverse*** extracts the linear differential operator from **`M`** and computes a left inverse of this operator.
- If **`M`** is given as differential operator in matrix form, then the arguments **`dvar`** and **`rvar`** can be omitted.
- If no left inverse of **`M`** exists, ***LeftInverse*** returns FAIL.
- If a left inverse L of **`M`** exists, ***LeftInverse*** returns such a differential operator L if **`M`** is a differential operator in matrix form, or returns the list of linear differential expressions which is obtained by applying L to the list **`rvar`** (or to the list of serially numbered variables _A if **`rvar`** is not given) if **`M`** is a list of linear differential expressions.
- Right inverses of linear differential operators are computed by RightInverse.

Examples:

```
> with(Janet):
> P := [[2*x^2, 4*x^2-2, 0], [x^2, 0, x^2-1], [-1, 0, 0], [2*x^2, 2*x^2, x^2]];
```

$$P := [[2x^2, 4x^2 - 2, 0], [x^2, 0, x^2 - 1], [-1, 0, 0], [2x^2, 2x^2, x^2]]$$

```
> ivar := [x];
```

$$ivar := [x]$$

```
> M := Pol2Op(P, ivar);
```

$$M := \begin{bmatrix} [[2, [x, x]]] & [[4, [x, x]], [-2, [\ ]]] & 0 \\ [[1, [x, x]]] & 0 & [[1, [x, x]], [-1, [\ ]]] \\ [[-1, [\ ]]] & 0 & 0 \\ [[2, [x, x]]] & [[2, [x, x]]] & [[1, [x, x]]] \end{bmatrix}$$

```
> L := LeftInverse(M, ivar);
```

$$L := \begin{bmatrix} 0 & 0 & [[-1, [\ ]]] & 0 \\ \left[[1, [x, x]], \left[\frac{-1}{2}, [\ ]\right]\right] & [[2, [x, x]]] & [[3, [x, x]]] & [[-2, [x, x]], [2, [\ ]]] \\ [[-1, [x, x]]] & [[-2, [x, x]], [-1, [\ ]]] & [[-3, [x, x]]] & [[2, [x, x]], [-1, [\ ]]] \end{bmatrix}$$

```
> CmpOp(L, M, ivar);
```

$$\begin{bmatrix} [[1, [\ ]]] & 0 & 0 \\ 0 & [[1, [\ ]]] & 0 \\ 0 & 0 & [[1, [\ ]]] \end{bmatrix}$$

```
> RightInverse(M, ivar);
```

$$FAIL$$

```
> dvar := [u1,u2,u3];
```

$$dvar := [u1, u2, u3]$$

```
> M := Pol2Diff(P, ivar, dvar);
```

$$M := \left[2\left(\frac{d^2}{dx^2}\,\mathrm{u1}(x)\right)+4\left(\frac{d^2}{dx^2}\,\mathrm{u2}(x)\right)-2\,\mathrm{u2}(x),\left(\frac{d^2}{dx^2}\,\mathrm{u1}(x)\right)+\left(\frac{d^2}{dx^2}\,\mathrm{u3}(x)\right)-\mathrm{u3}(x),-\mathrm{u1}(x),2\left(\frac{d^2}{dx^2}\,\mathrm{u1}(x)\right)+2\left(\frac{d^2}{dx^2}\,\mathrm{u2}(x)\right)+\left(\frac{d^2}{dx^2}\,\mathrm{u3}(x)\right)\right]$$

```
> LeftInverse(M, ivar, dvar);
```

$$\left[-\_\mathrm{A3}(x),2\,\_\mathrm{A4}(x)+3\left(\frac{d^2}{dx^2}\,\_\mathrm{A3}(x)\right)+2\left(\frac{d^2}{dx^2}\,\_\mathrm{A2}(x)\right)+\left(\frac{d^2}{dx^2}\,\_\mathrm{A1}(x)\right)-2\left(\frac{d^2}{dx^2}\,\_\mathrm{A4}(x)\right)-\frac{1}{2}\,\_\mathrm{A1}(x),\right.$$
$$\left.-\_\mathrm{A2}(x)-\_\mathrm{A4}(x)-3\left(\frac{d^2}{dx^2}\,\_\mathrm{A3}(x)\right)-2\left(\frac{d^2}{dx^2}\,\_\mathrm{A2}(x)\right)-\left(\frac{d^2}{dx^2}\,\_\mathrm{A1}(x)\right)+2\left(\frac{d^2}{dx^2}\,\_\mathrm{A4}(x)\right)\right]$$

```
> LeftInverse(M, ivar, dvar, [v1,v2,v3,v4]);
```

$$\left[-\mathrm{v3}(x),2\,\mathrm{v4}(x)+3\left(\frac{d^2}{dx^2}\,\mathrm{v3}(x)\right)+2\left(\frac{d^2}{dx^2}\,\mathrm{v2}(x)\right)+\left(\frac{d^2}{dx^2}\,\mathrm{v1}(x)\right)-2\left(\frac{d^2}{dx^2}\,\mathrm{v4}(x)\right)-\frac{1}{2}\,\mathrm{v1}(x),\right.$$
$$\left.-\mathrm{v2}(x)-\mathrm{v4}(x)-3\left(\frac{d^2}{dx^2}\,\mathrm{v3}(x)\right)-2\left(\frac{d^2}{dx^2}\,\mathrm{v2}(x)\right)-\left(\frac{d^2}{dx^2}\,\mathrm{v1}(x)\right)+2\left(\frac{d^2}{dx^2}\,\mathrm{v4}(x)\right)\right]$$

See Also:

JanetBasis, PrincDeriv, InvReduce, RightInverse, PDEFactorize, AppOp, CmpOp, AddOp, SubOp, Diff2Op, Pol2Op, Pol2Diff, AffEqn.

Janet[Linearize] - linearize a system of differential equations along general trajectory

Calling Sequence:

Linearize(L,ivar,dvar,lvar)

Parameters:

`L` - list of differential expressions or jet expressions
`ivar` - list of independent variables
`dvar` - list of dependent variables
`lvar` - list of linearized dependent variables

Description:

- ***Linearize*** returns the linearization of the system of differential equations given by **`L`** along a general trajectory (i.e. the Frechet derivative of **`L`**).
- The parameters **`dvar`** and **`lvar`** must be lists of the same length. The *i*th element in **`lvar`** is the linearization of the *i*th variable in **`dvar`**.
- If **`ivar`** = $[x_1, .., x_n]$, **`dvar`** = $[u_1, .., u_m]$, and **`lvar`** = $[U_1, .., U_m]$, then for each differential expression F in **`L`** this procedure gives $\sum_{i=1}^{m}\left(\frac{\partial}{\partial u_i}F\right)U_i(x_1, .., x_n)$.
- For more information about linearizing jet expressions see jlin.

Examples:

```
> with(Janet):
```

```
> ivar := [x,y]; dvar := [u,v]; Dvar := [U,V];
```

$$ivar := [x, y]$$

$$dvar := [u, v]$$

$$Dvar := [U, V]$$

```
> L := [diff(u(x,y),x,x) - u(x,y)*diff(v(x,y),y)];
```

$$L := \left[\left(\frac{\partial^2}{\partial x^2}\mathrm{u}(x, y)\right) - \mathrm{u}(x, y)\left(\frac{\partial}{\partial y}\mathrm{v}(x, y)\right)\right]$$

```
> Linearize(L, ivar, dvar, Dvar);
```

$$\left[\left(\frac{\partial^2}{\partial x^2}\mathrm{U}(x, y)\right) - \left(\frac{\partial}{\partial y}\mathrm{v}(x, y)\right)\mathrm{U}(x, y) - \mathrm{u}(x, y)\left(\frac{\partial}{\partial y}\mathrm{V}(x, y)\right)\right]$$

```
> L := [u[x,y]+cos(v)];
```

$$L := [u_{x,y} + \cos(v)]$$

```
> Linearize(L, ivar, dvar, Dvar);
```

$$[U_{x,y} - \sin(v)\,V]$$

See Also:

JanetBasis, PrincDeriv, ParamDeriv, Resolution, CompCond, Torsion, Parametrization, jlin, Ind2Diff, Diff2Ind.

Janet[Op2D] - convert linear differential operator in matrix form into polynomial of differential operators

Calling Sequence:

Op2D(oper,Dvar,ivar)

Parameters:

`oper` - differential operator (matrix with entries as described below)
`Dvar` - list of indeterminates representing the partial derivative operators
`ivar` - list of independent variables

Description:

- ***Op2D*** translates the linear differential operator **`oper`** in matrix form to the (list of lists of) polynomial(s) in **`Dvar`** which represents the same operator as **`oper`**, where the *i*-th indeterminate in **`Dvar`** represents differentiation with respect to the *i*-th independent variable.
- The entries of the matrix **`oper`** must have the form [[c_1, [...]], ..., [c_r, [...]]], where the dots in the innermost brackets are sequences of independent variables from **`ivar`** representing partial derivatives w. r. t. these variables, and the c_i are the coefficients for these, i. e. functions in **`ivar`**. The whole bracket represents the sum of these items, i. e. the linear combination of the above operators. Note, if an innermost bracket is empty, it represents the identity operator. (For a convenient way to produce such a matrix use Diff2Op as in the example below.)
- In general, the result of ***Op2D*** is a list of lists of polynomials in **`Dvar`**. The entries of the result are lists whose length equals the number of columns of **`oper`**. The number of these entries equals the number of rows of **`oper`**. If **`oper`** is not a matrix, but a single entry of a differential operator in matrix form as described above, then the result of ***Op2D*** is just a polynomial in **`Dvar`** (cf. Example 1 below).
- (Lists of (lists of)) polynomials in **`Dvar`** can be translated back to linear differential operators in matrix form by using D2Op.

Examples:

```
> with(Janet):
```

Example 1: Examples of linear differential operators acting on differential expressions in one dependent variable:

Ordinary differential operators:

```
> Dvar := [Dt]; ivar := [t];
```

$$Dvar := [Dt]$$

$$ivar := [t]$$

```
> L := u(t)-diff(u(t),t)+diff(u(t),t$3);
```

$$L := \mathrm{u}(t) - \left(\frac{d}{dt}\mathrm{u}(t)\right) + \left(\frac{d^3}{dt^3}\mathrm{u}(t)\right)$$

Translate differential operator in matrix form:

```
> oper := Diff2Op([L], ivar, [u]);
```

$$oper := [[1, [t, t, t]], [-1, [t]], [1, [\,]]]$$

```
> Op2D(oper, Dvar, ivar);
```

$$[[Dt^3 - Dt + 1]]$$

Translate the single entry of the above matrix:

```
> oper := Diff2Op(L, ivar, [u]);
```

$$oper := [[1, [t, t, t]], [-1, [t]], [1, [\,]]]$$

```
> Op2D(oper, Dvar, ivar);
```

$$Dt^3 - Dt + 1$$

Partial differential operators:

```
> Dvar := [Ds,Dt]; ivar := [s,t];
```

$$Dvar := [Ds, Dt]$$

$$ivar := [s, t]$$

```
> L := [2*diff(u(s,t),t)+u(s,t), -diff(u(s,t),s)+diff(u(s,t),t$2)];
```

$$L := \left[2\left(\frac{\partial}{\partial t}u(s,t)\right)+u(s,t), -\left(\frac{\partial}{\partial s}u(s,t)\right)+\left(\frac{\partial^2}{\partial t^2}u(s,t)\right)\right]$$

```
> oper := Diff2Op(L, ivar, [u]);
```

$$oper := \begin{bmatrix} [[2, [t]], [1, [\ ]]] \\ [[1, [t, t]], [-1, [s]]] \end{bmatrix}$$

```
> Op2D(oper, Dvar, ivar);
```

$$[[2\,Dt + 1], [Dt^2 - Ds]]$$

Example 2: Example of a differential operator acting on more than one dependent variable:

```
> Dvar := [Ds,Dt]; ivar := [s,t];
```

$$Dvar := [Ds, Dt]$$

$$ivar := [s, t]$$

```
> L := [[1+s^2, -2+t], [1+s*t, t^3]];
```

$$L := [[1 + s^2, -2 + t], [1 + s\,t, t^3]]$$

```
> oper := Pol2Op(L, ivar);
```

$$oper := \begin{bmatrix} [[1, [s, s]], [1, [\ ]]] & [[1, [t]], [-2, [\ ]]] \\ [[1, [s, t]], [1, [\ ]]] & [[1, [t, t, t]]] \end{bmatrix}$$

```
> Op2D(oper, Dvar, ivar);
```

$$[[Ds^2 + 1, Dt - 2], [Ds\,Dt + 1, Dt^3]]$$

See Also:

Diff2D, D2Diff, D2Op, CmpD, Diff2Op, AppOp, Pol2Op, AddOp, SubOp, Diff2Pol, Pol2Diff, Diff2Ind, Ind2Diff, AppOpInd, Pol2Ind.

Janet[PDEBasis] - vector space basis for the module of equations generated by the last computed Janet basis

Calling Sequence:

PDEBasis(ivar,dvar,oivar,subs)

Parameters:

`ivar` - list of independent variables as in the last call of ***JanetBasis***
`dvar` - list of dependent variables as in the last call of ***JanetBasis***
`oivar` - (optional) list of the independent variables in varied order as in the last call of ***JanetBasis***
`subs` - (optional) equation "subs"=expression

Description:

- ***PDEBasis*** returns a generating function, also called generalized Hilbert series, which enumerates (the leading derivatives of) a vector space basis for the module spanned by the equations in the Janet basis of the last call of JanetBasis, i. e. the vector space generated by these equations and all their partial derivatives.
- A term of the form $m/((1-x_1)...(1-x_n))\ e_i$ in the result enumerates all linear pdes which are obtained as partial derivatives of the unique Janet basis element with leading derivative m (for the i-th unknown function), where differentiation is restricted to be taken with respect to independent variables $x_1, ..., x_n$. Here m stands for a monomial in the indeterminates **`ivar`** corresponding to the partial derivative with respect to the same variables, and e_i for the i-th standard basis vector. Note that if the number of dependent variables is greater than one, the result of ***PDEBasis*** is accordingly a list of generating functions.
- The result of ***PDEBasis*** can also be easily read off from the information given by PrincDeriv. It is just the sum of the leading derivatives of the Janet basis, translated to monomials in the independent variables, each multiplied by the geometric series $1/((1-x_{\mu_1})...(1-x_{\mu_k})$, where $\{ x_{\mu_1}, ..., x_{\mu_k} \}$ is the corresponding set of multiplicative variables for the respective Janet basis element.
- **`ivar`** and **`dvar`** are expected to be the list of independent resp. dependent variables that were given as parameters to ***JanetBasis*** before.
- If the user specified an ordering of the independent variables different from **`ivar`** in the last call of JanetBasis using the optional parameter **`oivar`**, then the same parameter **`oivar`** has to be given to ***PDEBasis***.
- If an optional equation "subs"=expression is provided, then ***PDEBasis*** substitutes 'expression' for all variables in **`ivar`** in the result (cf. Example 1 below).
- For more information about generalized Hilbert series, see W. Plesken, D. Robertz, "Janet's approach to presentations and resolutions for polynomials and linear pdes", Archiv der Mathematik, 84(1), 2005, 22-37.

Examples:

```
> with(Janet):
```

Example 1:

```
> ivar := [x,y,t]; dvar := [u];
```

$$ivar := [x, y, t]$$

$$dvar := [u]$$

```
> L := [diff(u(x,y,t),x$2) - diff(u(x,y,t),t$2), diff(u(x,y,t),y)-u(x,y,t)];
```

$$L := \left[\left(\frac{\partial^2}{\partial x^2}\,u(x,y,t)\right) - \left(\frac{\partial^2}{\partial t^2}\,u(x,y,t)\right),\ \left(\frac{\partial}{\partial y}\,u(x,y,t)\right) - u(x,y,t)\right]$$

```
> JanetBasis(L, ivar, dvar):
> PrincDeriv();
```

$$\left[\left(\frac{\partial}{\partial y}\,u(x,y,t)\right) - u(x,y,t),\ [*, y, t],\ \frac{\partial}{\partial y}\,u(x,y,t)\right]$$

$$\left[-\left(\frac{\partial}{\partial x}\mathrm{u}(x,y,t)\right)+\left(\frac{\partial^2}{\partial y\,\partial x}\mathrm{u}(x,y,t)\right),[*,y,t],\frac{\partial^2}{\partial y\,\partial x}\mathrm{u}(x,y,t)\right]$$

$$\left[\left(\frac{\partial^2}{\partial x^2}\mathrm{u}(x,y,t)\right)-\left(\frac{\partial^2}{\partial t^2}\mathrm{u}(x,y,t)\right),[x,y,t],\frac{\partial^2}{\partial x^2}\mathrm{u}(x,y,t)\right]$$

```
> PDEBasis(ivar, dvar);
```

$$\frac{y}{(1-y)(1-t)}+\frac{xy}{(1-y)(1-t)}+\frac{x^2}{(1-x)(1-y)(1-t)}$$

```
> PDEBasis(ivar, dvar, "subs"=t);
```

$$\frac{t}{(1-t)^2}+\frac{t^2}{(1-t)^2}+\frac{t^2}{(1-t)^3}$$

```
> PDEHilbertSeries("var"=t);
```

$$\frac{t}{(1-t)^2}+\frac{t^2}{(1-t)^2}+\frac{t^2}{(1-t)^3}$$

```
> taylor(%, t=0, 10);
```

$$t+4t^2+8t^3+13t^4+19t^5+26t^6+34t^7+43t^8+53t^9+\mathrm{O}(t^{10})$$

```
> PDEHilbertFunction(1);
```

$$1$$

```
> PDEHilbertFunction(2);
```

$$4$$

```
> PDEHilbertFunction("");
Dim(M.s) = 0, for s < 1
Dim(M.1) = 1
Dim(M.s) = 3/2*s-1+1/2*s^2, for s >= 2
> PDEHilbertPolynomial(s);
```

$$\frac{3}{2}s-1+\frac{1}{2}s^2$$

Example 2:

```
> ivar := [x,y]; dvar := [u,v];
```

$$ivar := [x,y]$$

$$dvar := [u,v]$$

```
> L := [diff(u(x,y),x$2) - diff(v(x,y),y$2), diff(u(x,y),y)-v(x,y)];
```

$$L := \left[\left(\frac{\partial^2}{\partial x^2}\mathrm{u}(x,y)\right)-\left(\frac{\partial^2}{\partial y^2}\mathrm{v}(x,y)\right),\left(\frac{\partial}{\partial y}\mathrm{u}(x,y)\right)-\mathrm{v}(x,y)\right]$$

```
> JanetBasis(L, ivar, dvar):
> PrincDeriv();
```

$$\left[\left(\frac{\partial}{\partial y}\mathrm{u}(x,y)\right)-\mathrm{v}(x,y),[*,y],\frac{\partial}{\partial y}\mathrm{u}(x,y)\right]$$

$$\left[\left(\frac{\partial^2}{\partial y\,\partial x}\mathrm{u}(x,y)\right)-\left(\frac{\partial}{\partial x}\mathrm{v}(x,y)\right),[*,y],\frac{\partial^2}{\partial y\,\partial x}\mathrm{u}(x,y)\right]$$

$$\left[\left(\frac{\partial^2}{\partial x^2}\mathrm{u}(x,y)\right)-\left(\frac{\partial^2}{\partial y^2}\mathrm{v}(x,y)\right),[x,y],\frac{\partial^2}{\partial x^2}\mathrm{u}(x,y)\right]$$

$$\left[-\left(\frac{\partial^2}{\partial x^2}\mathrm{v}(x,y)\right)+\left(\frac{\partial^3}{\partial y^3}\mathrm{v}(x,y)\right),[x,y],\frac{\partial^3}{\partial y^3}\mathrm{v}(x,y)\right]$$

```
> PDEBasis(ivar, dvar);
```

$$\left[\frac{x^2}{(1-x)(1-y)}+\frac{xy}{1-y}+\frac{y}{1-y},\frac{y^3}{(1-x)(1-y)}\right]$$

```
> PDEBasis(ivar, dvar, "subs"=t);
```

$$\left[\frac{t^2}{(1-t)^2}+\frac{t^2}{1-t}+\frac{t}{1-t},\frac{t^3}{(1-t)^2}\right]$$

```
> PDEHilbertSeries("var"=t);
```

$$\frac{t^2}{(1-t)^2}+\frac{t^2}{1-t}+\frac{t}{1-t}+\frac{t^3}{(1-t)^2}$$

```
> taylor(%, t=0, 10);
```

$$t+3\,t^2+5\,t^3+7\,t^4+9\,t^5+11\,t^6+13\,t^7+15\,t^8+17\,t^9+O(t^{10})$$

```
> PDEHilbertFunction(1);
```

$$1$$

```
> PDEHilbertFunction(2);
```

$$3$$

```
> PDEHilbertFunction("");
Dim(M.s) = 0, for s < 1
Dim(M.1) = 1
Dim(M.2) = 3
Dim(M.s) = -1+2*s, for s >= 3
> PDEHilbertPolynomial(s);
```

$$-1+2\,s$$

Example 3: (changing the order of the independent variables)

```
> ivar := [x,y,z]; dvar := [u];
```

$$ivar := [x, y, z]$$

$$dvar := [u]$$

```
> L := [diff(u(x,y,z),x$2) - diff(u(x,y,z),y), diff(u(x,y,z),y,z)-u(x,y,z)];
```

$$L := \left[\left(\frac{\partial^2}{\partial x^2}\,u(x,y,z)\right)-\left(\frac{\partial}{\partial y}\,u(x,y,z)\right),\left(\frac{\partial^2}{\partial z\,\partial y}\,u(x,y,z)\right)-u(x,y,z)\right]$$

```
> JanetBasis(L, ivar, dvar, [y,x,z]):
> PrincDeriv();
```

$$\left[\left(\frac{\partial^2}{\partial z\,\partial y}\,u(x,y,z)\right)-u(x,y,z),[*,y,z],\frac{\partial^2}{\partial z\,\partial y}\,u(x,y,z)\right]$$

$$\left[\left(\frac{\partial^2}{\partial x^2}\,u(x,y,z)\right)-\left(\frac{\partial}{\partial y}\,u(x,y,z)\right),[x,*,z],\frac{\partial^2}{\partial x^2}\,u(x,y,z)\right]$$

$$\left[\left(\frac{\partial^3}{\partial z\,\partial y\,\partial x}\,u(x,y,z)\right)-\left(\frac{\partial}{\partial x}\,u(x,y,z)\right),[*,y,z],\frac{\partial^3}{\partial z\,\partial y\,\partial x}\,u(x,y,z)\right]$$

$$\left[-\left(\frac{\partial^2}{\partial y^2}\,u(x,y,z)\right)+\left(\frac{\partial^3}{\partial y\,\partial x^2}\,u(x,y,z)\right),[x,y,z],\frac{\partial^3}{\partial y\,\partial x^2}\,u(x,y,z)\right]$$

```
> PDEBasis(ivar, dvar, [y,x,z]);
```

$$\frac{yz}{(1-y)(1-z)}+\frac{x^2}{(1-x)(1-z)}+\frac{xyz}{(1-y)(1-z)}+\frac{yx^2}{(1-x)(1-y)(1-z)}$$

See Also:

JanetBasis, PrincDeriv, ParamDeriv, HilbertSeries, HilbertPolynomial, HP, HilbertFunction, HF, IndexRegularity, CartanCharacter, PDEHilbertSeries, PDEHilbertPolynomial, PDEHP, PDEHilbertFunction, PDEHF.

Janet[PDEFactorize] - compute a left hand factor of a linear differential operator with given right hand factor

Calling Sequence:

PDEFactorize(M1,M2,ivar,dvar,rvar)

Parameters:

`M1` - differential operator in matrix form or list of linear differential expressions
`M2` - differential operator in matrix form or list of linear differential expressions
`ivar` - list of independent variables
`dvar` - (optional) list of dependent variables
`rvar` - (optional) list of dependent variables for the left hand factor

Description:

- ***PDEFactorize*** divides (if possible) the linear differential operator corresponding to **`M1`** by the linear differential operator corresponding to **`M2`** from the right, i.e. it computes (if possible) a linear differential operator *F* such that **`M1`** = *F* **`M2`**. Note that *F* is not unique in general.
- The parameters **`M1`** and **`M2`** are expected to be linear differential operators in matrix form (cf. AppOp) with independent variables **`ivar`** or lists of linear differential expressions with independent variables **`ivar`** and dependent variables **`dvar`**. In the second case, ***PDEFactorize*** extracts the linear differential operators from **`M1`** and **`M2`** and performs the division with these operators.
- If **`M1`** and **`M2`** are given as differential operators in matrix form, then the arguments **`dvar`** and **`rvar`** can be omitted.
- If no left hand factor *F* of **`M1`** with corresponding right hand factor **`M2`** exists, ***PDEFactorize*** returns FAIL.
- If a left hand factor *F* of **`M1`** with corresponding right hand factor **`M2`** exists, ***PDEFactorize*** returns such a differential operator *F* if **`M1`** and **`M2`** are differential operators in matrix form, or returns the list of linear differential expressions which is obtained by applying *F* to the list **`rvar`** (or to the list of serially numbered variables _A if **`rvar`** is not given) if **`M1`** and **`M2`** are lists of linear differential expressions.
- Composition of linear differential operators is performed by CmpOp.

Examples:

```
> with(Janet):
```

Example 1:

```
> ivar := [x,y]; dvar := [u];
```

$$ivar := [x, y]$$

$$dvar := [u]$$

```
> M1 := [diff(u(x,y), x, y)];
```

$$M1 := \left[\frac{\partial^2}{\partial y \partial x} u(x, y)\right]$$

```
> M2 := [diff(u(x,y), y)];
```

$$M2 := \left[\frac{\partial}{\partial y} u(x, y)\right]$$

```
> PDEFactorize(M1, M2, ivar, dvar);
```

$$\left[\frac{\partial}{\partial x} \_A1(x, y)\right]$$

```
> PDEFactorize(M1, M2, ivar, dvar, [v]);
```

$$\left[\frac{\partial}{\partial x} v(x, y)\right]$$

```
> M1 := [diff(u(x,y), x, x)];
```

$$M1 := \left[\frac{\partial^2}{\partial x^2}\,u(x,y)\right]$$

```
> PDEFactorize(M1, M2, ivar, dvar);
```

$$FAIL$$

Example 2:

```
> ivar := [x];
```

$$ivar := [x]$$

```
> M1 := matrix([[[[x, [x, x]]], [[1, [x, x]], [-2, []]], 0], [[[1, [x, x]]], 0, [[1, [x,
  x]], [-x, []]]]]);
```

$$M1 := \begin{bmatrix} [[x,[x,x]]] & [[1,[x,x]],[-2,[\ ]]] & 0 \\ [[1,[x,x]]] & 0 & [[1,[x,x]],[-x,[\ ]]] \end{bmatrix}$$

```
> M2 := matrix([[[[1, [x, x]], [-2, []]], 0], [[[1, [x, x]]], [[1, [x, x]], [-2, []]]],
  [[[1, [x, x]]], [[1, [x, x]], [-x, []]]]]);
```

$$M2 := \begin{bmatrix} [[1,[x,x]],[-2,[\ ]]] & 0 \\ [[1,[x,x]]] & [[1,[x,x]],[-2,[\ ]]] \\ [[1,[x,x]]] & [[1,[x,x]],[-x,[\ ]]] \end{bmatrix}$$

```
> MM := CmpOp(M1, M2, ivar);
```

$$MM := \begin{bmatrix} [[x+1,[x,x,x,x]],[-2x-2,[x,x]]] & [[1,[x,x,x,x]],[-4,[x,x]],[4,[\ ]]] \\ [[2,[x,x,x,x]],[-2-x,[x,x]]] & [[1,[x,x,x,x]],[-2x,[x,x]],[-2,[x]],[x^2,[\ ]]] \end{bmatrix}$$

```
> F := PDEFactorize(MM, M2, ivar);
```

$$F := \begin{bmatrix} [[x,[x,x]]] & [[1,[x,x]],[-2,[\ ]]] & 0 \\ [[1,[x,x]]] & 0 & [[1,[x,x]],[-x,[\ ]]] \end{bmatrix}$$

```
> SubOp(CmpOp(F, M2, ivar), MM, ivar);
```

$$\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

```
> dvar := [u1,u2];
```

$$dvar := [u1, u2]$$

```
> L2 := AppOp(M2, dvar, ivar, dvar);
```

$$L2 := \left[\left(\frac{d^2}{dx^2}\,u1(x)\right) - 2\,u1(x), \left(\frac{d^2}{dx^2}\,u1(x)\right) + \left(\frac{d^2}{dx^2}\,u2(x)\right) - 2\,u2(x), \left(\frac{d^2}{dx^2}\,u1(x)\right) + \left(\frac{d^2}{dx^2}\,u2(x)\right) - x\,u2(x)\right]$$

```
> LL := AppOp(MM, dvar, ivar, dvar);
```

$$LL := \left[\left(\frac{d^4}{dx^4}\,u1(x)\right)x + \left(\frac{d^4}{dx^4}\,u1(x)\right) - 2\left(\frac{d^2}{dx^2}\,u1(x)\right)x - 2\left(\frac{d^2}{dx^2}\,u1(x)\right) + \left(\frac{d^4}{dx^4}\,u2(x)\right) - 4\left(\frac{d^2}{dx^2}\,u2(x)\right) + 4\,u2(x),\right.$$
$$\left. 2\left(\frac{d^4}{dx^4}\,u1(x)\right) - 2\left(\frac{d^2}{dx^2}\,u1(x)\right) - \left(\frac{d^2}{dx^2}\,u1(x)\right)x + \left(\frac{d^4}{dx^4}\,u2(x)\right) - 2x\left(\frac{d^2}{dx^2}\,u2(x)\right) - 2\left(\frac{d}{dx}\,u2(x)\right) + x^2\,u2(x)\right]$$

```
> FF := PDEFactorize(LL, L2, ivar, dvar, [v1,v2,v3]);
```

$$FF := \left[x\left(\frac{d^2}{dx^2}\,v1(x)\right) - 2\,v2(x) + \left(\frac{d^2}{dx^2}\,v2(x)\right), \left(\frac{d^2}{dx^2}\,v1(x)\right) - v3(x)\,x + \left(\frac{d^2}{dx^2}\,v3(x)\right)\right]$$

```
> Diff2Op(FF, ivar, [v1,v2,v3]);
```

$$\begin{bmatrix} [[x,[x,x]]] & [[1,[x,x]],[-2,[\ ]]] & 0 \\ [[1,[x,x]]] & 0 & [[1,[x,x]],[-x,[\ ]]] \end{bmatrix}$$

```
> SubOp(CmpOp(%, M2, ivar), MM, ivar);
```

$$\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

See Also:

JanetBasis, PrincDeriv, InvReduce, LeftInverse, RightInverse, AppOp, CmpOp, AddOp, SubOp, Diff2Op, Pol2Op, Pol2Diff, AffEqn

Janet[PDEHF] - compute the filtered Hilbert function of the module of equations generated by the last computed Janet basis

Calling Sequence:

PDEHF(i)
PDEHF()

Parameters:

`i` - " " (empty string) or non-negative integer

Description:

- ***PDEHF***(**`i`**) returns the sum of the dimensions of the *j*-th graded components, where *j* runs from 0 to **`i`**, of the module of equations for the linear system of PDEs whose Janet basis has been computed last by JanetBasis. For more information about this module and its grading, see PDEHilbertSeries.
- ***PDEHF***() returns a function expecting one parameter **`i`** which computes ***PDEHF***(**`i`**) .
- ***PDEHF***("") prints the function ***PDEHF***().
- The command only uses the data displayable by TabVar and depends on the ordering of derivatives chosen for the last call of ***JanetBasis***.
- For the Hilbert function counting free Taylor coefficients up to order **`i`** in the expansion of a general solution of the system of PDEs, see HE.

Examples:

```
> with(Janet):
```

Example 1:

```
> ivar := [x,y,t]; dvar := [u];
```

$$ivar := [x, y, t]$$

$$dvar := [u]$$

```
> L := [diff(u(x,y,t),x$2) - diff(u(x,y,t),t$2), diff(u(x,y,t),y)-u(x,y,t)];
```

$$L := \left[\left(\frac{\partial^2}{\partial x^2}\,u(x,y,t)\right)-\left(\frac{\partial^2}{\partial t^2}\,u(x,y,t)\right),\left(\frac{\partial}{\partial y}\,u(x,y,t)\right)-u(x,y,t)\right]$$

```
> JanetBasis(L, ivar, dvar):
> PrincDeriv();
```

$$\left[\left(\frac{\partial}{\partial y}\,u(x,y,t)\right)-u(x,y,t),[*,y,t],\frac{\partial}{\partial y}\,u(x,y,t)\right]$$

$$\left[-\left(\frac{\partial}{\partial x}\,u(x,y,t)\right)+\left(\frac{\partial^2}{\partial y\,\partial x}\,u(x,y,t)\right),[*,y,t],\frac{\partial^2}{\partial y\,\partial x}\,u(x,y,t)\right]$$

$$\left[\left(\frac{\partial^2}{\partial x^2}\,u(x,y,t)\right)-\left(\frac{\partial^2}{\partial t^2}\,u(x,y,t)\right),[x,y,t],\frac{\partial^2}{\partial x^2}\,u(x,y,t)\right]$$

```
> PDEBasis(ivar, dvar);
```

$$\frac{y}{(1-y)(1-t)}+\frac{xy}{(1-y)(1-t)}+\frac{x^2}{(1-x)(1-y)(1-t)}$$

```
> PDEBasis(ivar, dvar, "subs"=t);
```

$$\frac{t}{(1-t)^2}+\frac{t^2}{(1-t)^2}+\frac{t^2}{(1-t)^3}$$

```
> PDEHilbertSeries("var"=t);
```

$$\frac{t}{(1-t)^2}+\frac{t^2}{(1-t)^2}+\frac{t^2}{(1-t)^3}$$

```
> taylor(%, t=0, 10);
```

$$t + 4\,t^2 + 8\,t^3 + 13\,t^4 + 19\,t^5 + 26\,t^6 + 34\,t^7 + 43\,t^8 + 53\,t^9 + O(t^{10})$$

```
> PDEHF(1);
```

$$1$$

```
> PDEHF(2);
```

$$5$$

```
> PDEHF("");
s < 1: 0
s = 1: 1
s >= 2: -1/6*s+s^2+1/6*s^3
> PDEHP(s);
```

$$-\frac{1}{6}s + s^2 + \frac{1}{6}s^3$$

Example 2:

```
> ivar := [x,y]; dvar := [u,v];
```

$$ivar := [x, y]$$

$$dvar := [u, v]$$

```
> L := [diff(u(x,y),x$2) - diff(v(x,y),y$2), diff(u(x,y),y)-v(x,y)];
```

$$L := \left[\left(\frac{\partial^2}{\partial x^2}\,u(x,y)\right) - \left(\frac{\partial^2}{\partial y^2}\,v(x,y)\right), \left(\frac{\partial}{\partial y}\,u(x,y)\right) - v(x,y)\right]$$

```
> JanetBasis(L, ivar, dvar):
> PrincDeriv();
```

$$\left[\left(\frac{\partial}{\partial y}\,u(x,y)\right) - v(x,y), [*, y], \frac{\partial}{\partial y}\,u(x,y)\right]$$

$$\left[\left(\frac{\partial^2}{\partial y\,\partial x}\,u(x,y)\right) - \left(\frac{\partial}{\partial x}\,v(x,y)\right), [*, y], \frac{\partial^2}{\partial y\,\partial x}\,u(x,y)\right]$$

$$\left[\left(\frac{\partial^2}{\partial x^2}\,u(x,y)\right) - \left(\frac{\partial^2}{\partial y^2}\,v(x,y)\right), [x, y], \frac{\partial^2}{\partial x^2}\,u(x,y)\right]$$

$$\left[-\left(\frac{\partial^2}{\partial x^2}\,v(x,y)\right) + \left(\frac{\partial^3}{\partial y^3}\,v(x,y)\right), [x, y], \frac{\partial^3}{\partial y^3}\,v(x,y)\right]$$

```
> PDEBasis(ivar, dvar);
```

$$\left[\frac{x^2}{(1-x)(1-y)} + \frac{xy}{1-y} + \frac{y}{1-y}, \frac{y^3}{(1-x)(1-y)}\right]$$

```
> PDEBasis(ivar, dvar, "subs"=t);
```

$$\left[\frac{t^2}{(1-t)^2} + \frac{t^2}{1-t} + \frac{t}{1-t}, \frac{t^3}{(1-t)^2}\right]$$

```
> PDEHilbertSeries("var"=t);
```

$$\frac{t^2}{(1-t)^2} + \frac{t^2}{1-t} + \frac{t}{1-t} + \frac{t^3}{(1-t)^2}$$

```
> taylor(%, t=0, 10);
```

$$t + 3\,t^2 + 5\,t^3 + 7\,t^4 + 9\,t^5 + 11\,t^6 + 13\,t^7 + 15\,t^8 + 17\,t^9 + O(t^{10})$$

```
> PDEHF(2);
```

$$4$$

```
> PDEHF(3);
```

$$9$$

```
> PDEHF("");
s < 1: 0
s = 1: 1
s = 2: 4
s >= 3: s^2
> PDEHP(s);
```

$$s^2$$

See Also:

JanetBasis, PrincDeriv, ParamDeriv, HilbertSeries, HilbertPolynomial, HP, HilbertFunction, HF, IndexRegularity, CartanCharacter, PDEBasis, PDEHilbertSeries, PDEHilbertPolynomial, PDEHP, PDEHilbertFunction.

Janet[PDEHP] - compute the filtered Hilbert polynomial of the module of equations generated by the last computed Janet basis

Calling Sequence:

PDEHP(p)
PDEHP()

Parameters:

p - natural number or name of an indeterminate

Description:

- ***PDEHP***() returns the filtered Hilbert polynomial of the module of equations for the linear system of PDEs whose Janet basis has been computed last by JanetBasis. For more information about this module, see PDEHilbertSeries.
- The same information can also be extracted from the command PDEHF.
- ***PDEHP***(**p**) returns the value of the polynomial function ***PDEHP*** at **p**, where **p** might also be the name of the indeterminate for the Hilbert polynomial.
- The command only uses the data displayable by TabVar and depends on the ordering of derivatives chosen for the last call of ***JanetBasis***.
- The default name of the indeterminate is 's'. To use the subs command later on the indeterminate, one has to explicitly give a name to the indeterminate.
- For the filtered Hilbert polynomial counting free Taylor coefficients in the expansion of a general solution of the system of PDEs beyond the index of regularity, see HP.

Examples:

```
> with(Janet):
```

Example 1:

```
> ivar := [x,y,t]; dvar := [u];
```

$$ivar := [x, y, t]$$

$$dvar := [u]$$

```
> L := [diff(u(x,y,t),x$2) - diff(u(x,y,t),t$2), diff(u(x,y,t),y)-u(x,y,t)];
```

$$L := \left[\left(\frac{\partial^2}{\partial x^2}\mathrm{u}(x,y,t)\right)-\left(\frac{\partial^2}{\partial t^2}\mathrm{u}(x,y,t)\right),\left(\frac{\partial}{\partial y}\mathrm{u}(x,y,t)\right)-\mathrm{u}(x,y,t)\right]$$

```
> JanetBasis(L, ivar, dvar):
> PrincDeriv();
```

$$\left[\left(\frac{\partial}{\partial y}\mathrm{u}(x,y,t)\right)-\mathrm{u}(x,y,t),[*,y,t],\frac{\partial}{\partial y}\mathrm{u}(x,y,t)\right]$$

$$\left[-\left(\frac{\partial}{\partial x}\mathrm{u}(x,y,t)\right)+\left(\frac{\partial^2}{\partial y\,\partial x}\mathrm{u}(x,y,t)\right),[*,y,t],\frac{\partial^2}{\partial y\,\partial x}\mathrm{u}(x,y,t)\right]$$

$$\left[\left(\frac{\partial^2}{\partial x^2}\mathrm{u}(x,y,t)\right)-\left(\frac{\partial^2}{\partial t^2}\mathrm{u}(x,y,t)\right),[x,y,t],\frac{\partial^2}{\partial x^2}\mathrm{u}(x,y,t)\right]$$

```
> PDEBasis(ivar, dvar);
```

$$\frac{y}{(1-y)(1-t)}+\frac{xy}{(1-y)(1-t)}+\frac{x^2}{(1-x)(1-y)(1-t)}$$

```
> PDEBasis(ivar, dvar, "subs"=t);
```

$$\frac{t}{(1-t)^2}+\frac{t^2}{(1-t)^2}+\frac{t^2}{(1-t)^3}$$

```
> PDEHilbertSeries("var"=t);
```

$$\frac{t}{(1-t)^2}+\frac{t^2}{(1-t)^2}+\frac{t^2}{(1-t)^3}$$

```
> taylor(%, t=0, 10);
```

$$t+4t^2+8t^3+13t^4+19t^5+26t^6+34t^7+43t^8+53t^9+O(t^{10})$$

```
> PDEHF(1);
```

$$1$$

```
> PDEHF(2);
```

$$5$$

```
> PDEHF("");
s < 1: 0
s = 1: 1
s >= 2: -1/6*s+s^2+1/6*s^3
> PDEHP(s);
```

$$-\frac{1}{6}s+s^2+\frac{1}{6}s^3$$

```
> PDEHP(5);
```

$$45$$

Example 2:

```
> ivar := [x,y]; dvar := [u,v];
```

$$ivar := [x, y]$$

$$dvar := [u, v]$$

```
> L := [diff(u(x,y),x$2) - diff(v(x,y),y$2), diff(u(x,y),y)-v(x,y)];
```

$$L := \left[\left(\frac{\partial^2}{\partial x^2}u(x,y)\right)-\left(\frac{\partial^2}{\partial y^2}v(x,y)\right), \left(\frac{\partial}{\partial y}u(x,y)\right)-v(x,y)\right]$$

```
> JanetBasis(L, ivar, dvar):
> PrincDeriv();
```

$$\left[\left(\frac{\partial}{\partial y}u(x,y)\right)-v(x,y), [*,y], \frac{\partial}{\partial y}u(x,y)\right]$$

$$\left[\left(\frac{\partial^2}{\partial y\partial x}u(x,y)\right)-\left(\frac{\partial}{\partial x}v(x,y)\right), [*,y], \frac{\partial^2}{\partial y\partial x}u(x,y)\right]$$

$$\left[\left(\frac{\partial^2}{\partial x^2}u(x,y)\right)-\left(\frac{\partial^2}{\partial y^2}v(x,y)\right), [x,y], \frac{\partial^2}{\partial x^2}u(x,y)\right]$$

$$\left[-\left(\frac{\partial^2}{\partial x^2}v(x,y)\right)+\left(\frac{\partial^3}{\partial y^3}v(x,y)\right), [x,y], \frac{\partial^3}{\partial y^3}v(x,y)\right]$$

```
> PDEBasis(ivar, dvar);
```

$$\left[\frac{x^2}{(1-x)(1-y)}+\frac{xy}{1-y}+\frac{y}{1-y}, \frac{y^3}{(1-x)(1-y)}\right]$$

```
> PDEBasis(ivar, dvar, "subs"=t);
```

$$\left[\frac{t^2}{(1-t)^2}+\frac{t^2}{1-t}+\frac{t}{1-t}, \frac{t^3}{(1-t)^2}\right]$$

```
> PDEHilbertSeries("var"=t);
```

$$\frac{t^2}{(1-t)^2}+\frac{t^2}{1-t}+\frac{t}{1-t}+\frac{t^3}{(1-t)^2}$$

```
> taylor(%, t=0, 10);
```

$$t+3t^2+5t^3+7t^4+9t^5+11t^6+13t^7+15t^8+17t^9+O(t^{10})$$

```
> PDEHF(1);
```

$$1$$

```
> PDEHF(2);
```

$$4$$

```
> PDEHF("");
s < 1: 0
s = 1: 1
s = 2: 4
s >= 3: s^2
> PDEHP(s);
```

$$s^2$$

```
> PDEHP(4);
```

$$16$$

See Also:

JanetBasis, PrincDeriv, ParamDeriv, HilbertSeries, HilbertPolynomial, HP, HilbertFunction, HF, IndexRegularity, CartanCharacter, PDEBasis, PDEHilbertSeries, PDEHilbertPolynomial, PDEHilbertFunction, PDEHF.

Janet[PDEHilbertFunction] - graded Hilbert function of the module of equations generated by the last computed Janet basis

Calling Sequence:

PDEHilbertFunction(i)
PDEHilbertFunction()

Parameters:

`i` - "" (empty string) or non-negative integer

Description:

- ***PDEHilbertFunction***(**`i`**) returns the dimension of the **`i`**-th graded component of the module of equations for the linear system of PDEs whose Janet basis has been computed last by JanetBasis. For more information about this module, see PDEHilbertSeries. This information can also be obtained from the Hilbert series for this module of equations, cf. again PDEHilbertSeries.
- ***PDEHilbertFunction***() returns a function expecting one parameter **`i`** which computes ***PDEHilbertFunction***(**`i`**) .
- ***PDEHilbertFunction***("") prints the function ***PDEHilbertFunction***().
- The command only uses the data displayable by TabVar and depends on the ordering of derivatives chosen for the last call of ***JanetBasis***.
- For the Hilbert function counting free Taylor coefficients of order **`i`** in the expansion of a general solution of the system of PDEs, see HilbertFunction.

Examples:

```
> with(Janet):
```

Example 1:

```
> ivar := [x,y,t]; dvar := [u];
```

$$ivar := [x, y, t]$$

$$dvar := [u]$$

```
> L := [diff(u(x,y,t),x$2) - diff(u(x,y,t),t$2), diff(u(x,y,t),y)-u(x,y,t)];
```

$$L := \left[\left(\frac{\partial^2}{\partial x^2}\,u(x,y,t)\right)-\left(\frac{\partial^2}{\partial t^2}\,u(x,y,t)\right),\left(\frac{\partial}{\partial y}\,u(x,y,t)\right)-u(x,y,t)\right]$$

```
> JanetBasis(L, ivar, dvar):
> PrincDeriv();
```

$$\left[\left(\frac{\partial}{\partial y}\,u(x,y,t)\right)-u(x,y,t),[*,y,t],\frac{\partial}{\partial y}\,u(x,y,t)\right]$$

$$\left[-\left(\frac{\partial}{\partial x}\,u(x,y,t)\right)+\left(\frac{\partial^2}{\partial y\,\partial x}\,u(x,y,t)\right),[*,y,t],\frac{\partial^2}{\partial y\,\partial x}\,u(x,y,t)\right]$$

$$\left[\left(\frac{\partial^2}{\partial x^2}\,u(x,y,t)\right)-\left(\frac{\partial^2}{\partial t^2}\,u(x,y,t)\right),[x,y,t],\frac{\partial^2}{\partial x^2}\,u(x,y,t)\right]$$

```
> PDEHilbertSeries("var"=t);
```

$$\frac{t}{(1-t)^2}+\frac{t^2}{(1-t)^2}+\frac{t^2}{(1-t)^3}$$

```
> taylor(%, t=0, 10);
```

$$t+4\,t^2+8\,t^3+13\,t^4+19\,t^5+26\,t^6+34\,t^7+43\,t^8+53\,t^9+O(t^{10})$$

```
> PDEHilbertFunction(1);
```

$$1$$

```
> PDEHilbertFunction(2);
```

$$4$$

```
> PDEHilbertFunction("");
Dim(M.s) = 0, for s < 1
Dim(M.1) = 1
Dim(M.s) = 3/2*s-1+1/2*s^2, for s >= 2
> PDEHilbertPolynomial(s);
```

$$\frac{3}{2}s - 1 + \frac{1}{2}s^2$$

```
> PDEHilbertPolynomial(2);
```

$$4$$

Example 2:

```
> ivar := [x,y]; dvar := [u,v];
```

$$ivar := [x, y]$$

$$dvar := [u, v]$$

```
> L := [diff(u(x,y),x$2) - diff(v(x,y),y$2), diff(u(x,y),y)-v(x,y)];
```

$$L := \left[\left(\frac{\partial^2}{\partial x^2}\mathrm{u}(x,y)\right) - \left(\frac{\partial^2}{\partial y^2}\mathrm{v}(x,y)\right), \left(\frac{\partial}{\partial y}\mathrm{u}(x,y)\right) - \mathrm{v}(x,y)\right]$$

```
> JanetBasis(L, ivar, dvar):
> PrincDeriv();
```

$$\left[\left(\frac{\partial}{\partial y}\mathrm{u}(x,y)\right) - \mathrm{v}(x,y), [*, y], \frac{\partial}{\partial y}\mathrm{u}(x,y)\right]$$

$$\left[\left(\frac{\partial^2}{\partial y \partial x}\mathrm{u}(x,y)\right) - \left(\frac{\partial}{\partial x}\mathrm{v}(x,y)\right), [*, y], \frac{\partial^2}{\partial y \partial x}\mathrm{u}(x,y)\right]$$

$$\left[\left(\frac{\partial^2}{\partial x^2}\mathrm{u}(x,y)\right) - \left(\frac{\partial^2}{\partial y^2}\mathrm{v}(x,y)\right), [x, y], \frac{\partial^2}{\partial x^2}\mathrm{u}(x,y)\right]$$

$$\left[-\left(\frac{\partial^2}{\partial x^2}\mathrm{v}(x,y)\right) + \left(\frac{\partial^3}{\partial y^3}\mathrm{v}(x,y)\right), [x, y], \frac{\partial^3}{\partial y^3}\mathrm{v}(x,y)\right]$$

```
> PDEHilbertSeries("var"=t);
```

$$\frac{t^2}{(1-t)^2} + \frac{t^2}{1-t} + \frac{t}{1-t} + \frac{t^3}{(1-t)^2}$$

```
> taylor(%, t=0, 10);
```

$$t + 3t^2 + 5t^3 + 7t^4 + 9t^5 + 11t^6 + 13t^7 + 15t^8 + 17t^9 + \mathrm{O}(t^{10})$$

```
> PDEHilbertFunction(1);
```

$$1$$

```
> PDEHilbertFunction(2);
```

$$3$$

```
> PDEHilbertFunction("");
Dim(M.s) = 0, for s < 1
Dim(M.1) = 1
Dim(M.2) = 3
Dim(M.s) = -1+2*s, for s >= 3
> PDEHilbertPolynomial(s);
```

$$-1 + 2s$$

```
> PDEHilbertPolynomial(2);
```

$$3$$

See Also:

JanetBasis, PrincDeriv, ParamDeriv, HilbertSeries, HilbertPolynomial, HP, HilbertFunction, HF, IndexRegularity, CartanCharacter, PDEBasis, PDEHilbertSeries, PDEHilbertPolynomial, PDEHP, PDEHF.

Janet[PDEHilbertPolynomial] - graded Hilbert polynomial of the module of equations generated by the last computed Janet basis

Calling Sequence:

PDEHilbertPolynomial(p)

Parameters:

p - (optional) name of the indeterminate (default: 's') or non-negative integer

Description:

- ***PDEHilbertPolynomial***() returns the graded Hilbert polynomial of the module of equations for the linear system of PDEs whose Janet basis has been computed last by JanetBasis. For more information about this module, see PDEHilbertSeries.
- The same information can also be extracted from the command PDEHilbertFunction.
- ***PDEHilbertPolynomial***(**p**) returns the value of the polynomial function ***PDEHilbertPolynomial*** at **p**, where **p** might also be the name of the indeterminate for the Hilbert polynomial.
- The command only uses the data displayable by TabVar and depends on the ordering of derivatives chosen for the last call of ***JanetBasis***.
- The default name of the indeterminate is 's'. To use the subs command later on the indeterminate, one has to explicitly give a name to the indeterminate.
- For the Hilbert polynomial counting free Taylor coefficients in the expansion of a general solution of the system of PDEs beyond the index of regularity, see HilbertPolynomial.

Examples:

```
> with(Janet):
```

Example 1:

```
> ivar := [x,y,t]; dvar := [u];
```

$$ivar := [x, y, t]$$

$$dvar := [u]$$

```
> L := [diff(u(x,y,t),x$2) - diff(u(x,y,t),t$2), diff(u(x,y,t),y)-u(x,y,t)];
```

$$L := \left[\left(\frac{\partial^2}{\partial x^2}\,u(x,y,t)\right)-\left(\frac{\partial^2}{\partial t^2}\,u(x,y,t)\right),\left(\frac{\partial}{\partial y}\,u(x,y,t)\right)-u(x,y,t)\right]$$

```
> JanetBasis(L, ivar, dvar):
> PrincDeriv();
```

$$\left[\left(\frac{\partial}{\partial y}\,u(x,y,t)\right)-u(x,y,t),[*,y,t],\frac{\partial}{\partial y}\,u(x,y,t)\right]$$

$$\left[-\left(\frac{\partial}{\partial x}\,u(x,y,t)\right)+\left(\frac{\partial^2}{\partial y\,\partial x}\,u(x,y,t)\right),[*,y,t],\frac{\partial^2}{\partial y\,\partial x}\,u(x,y,t)\right]$$

$$\left[\left(\frac{\partial^2}{\partial x^2}\,u(x,y,t)\right)-\left(\frac{\partial^2}{\partial t^2}\,u(x,y,t)\right),[x,y,t],\frac{\partial^2}{\partial x^2}\,u(x,y,t)\right]$$

```
> PDEHilbertSeries("var"=t);
```

$$\frac{t}{(1-t)^2}+\frac{t^2}{(1-t)^2}+\frac{t^2}{(1-t)^3}$$

```
> taylor(%, t=0, 10);
```

$$t+4\,t^2+8\,t^3+13\,t^4+19\,t^5+26\,t^6+34\,t^7+43\,t^8+53\,t^9+O(t^{10})$$

```
> PDEHilbertFunction(1);
```



```
> PDEHilbertFunction(2);
```

$$4$$

```
> PDEHilbertFunction("");
Dim(M.s) = 0, for s < 1
Dim(M.1) = 1
Dim(M.s) = 3/2*s-1+1/2*s^2, for s >= 2
```

```
> PDEHilbertPolynomial(s);
```

$$\frac{3}{2}s - 1 + \frac{1}{2}s^2$$

```
> PDEHilbertPolynomial(3);
```

$$8$$

Example 2:

```
> ivar := [x,y]; dvar := [u,v];
```

$$ivar := [x, y]$$

$$dvar := [u, v]$$

```
> L := [diff(u(x,y),x$2) - diff(v(x,y),y$2), diff(u(x,y),y)-v(x,y)];
```

$$L := \left[\left(\frac{\partial^2}{\partial x^2}\mathrm{u}(x,y)\right) - \left(\frac{\partial^2}{\partial y^2}\mathrm{v}(x,y)\right), \left(\frac{\partial}{\partial y}\mathrm{u}(x,y)\right) - \mathrm{v}(x,y)\right]$$

```
> JanetBasis(L, ivar, dvar):
> PrincDeriv();
```

$$\left[\left(\frac{\partial}{\partial y}\mathrm{u}(x,y)\right) - \mathrm{v}(x,y), [*, y], \frac{\partial}{\partial y}\mathrm{u}(x,y)\right]$$

$$\left[\left(\frac{\partial^2}{\partial y \partial x}\mathrm{u}(x,y)\right) - \left(\frac{\partial}{\partial x}\mathrm{v}(x,y)\right), [*, y], \frac{\partial^2}{\partial y \partial x}\mathrm{u}(x,y)\right]$$

$$\left[\left(\frac{\partial^2}{\partial x^2}\mathrm{u}(x,y)\right) - \left(\frac{\partial^2}{\partial y^2}\mathrm{v}(x,y)\right), [x, y], \frac{\partial^2}{\partial x^2}\mathrm{u}(x,y)\right]$$

$$\left[-\left(\frac{\partial^2}{\partial x^2}\mathrm{v}(x,y)\right) + \left(\frac{\partial^3}{\partial y^3}\mathrm{v}(x,y)\right), [x, y], \frac{\partial^3}{\partial y^3}\mathrm{v}(x,y)\right]$$

```
> PDEHilbertSeries("var"=t);
```

$$\frac{t^2}{(1-t)^2} + \frac{t^2}{1-t} + \frac{t}{1-t} + \frac{t^3}{(1-t)^2}$$

```
> taylor(%, t=0, 10);
```

$$t + 3t^2 + 5t^3 + 7t^4 + 9t^5 + 11t^6 + 13t^7 + 15t^8 + 17t^9 + \mathrm{O}(t^{10})$$

```
> PDEHilbertFunction(1);
```

$$1$$

```
> PDEHilbertFunction(2);
```

$$3$$

```
> PDEHilbertFunction("");
Dim(M.s) = 0, for s < 1
Dim(M.1) = 1
Dim(M.2) = 3
Dim(M.s) = -1+2*s, for s >= 3
```

```
> PDEHilbertPolynomial(s);
```

$$-1 + 2s$$

```
> PDEHilbertPolynomial(4);
```

$$7$$

See Also:

JanetBasis, PrincDeriv, ParamDeriv, HilbertSeries, HilbertPolynomial, HP, HilbertFunction, HF, IndexRegularity, CartanCharacter, PDEBasis, PDEHilbertSeries, PDEHP, PDEHilbertFunction, PDEHF.

Janet[PDEHilbertSeries] - Hilbert series of the module of equations generated by the last computed Janet basis

Calling Sequence:

PDEHilbertSeries(v)

Parameters:

v - (optional) name of the indeterminate (default: 's')

Description:

- ***PDEHilbertSeries*** returns a generating function counting - according to the standard degrees - the leading derivatives of the module *M* formed by all linear combinations of the equations in the Janet basis produced by the last call of JanetBasis, and all their partial derivatives. The leading derivatives are determined according to the ordering of derivatives chosen for the last run of ***JanetBasis***.
- The module *M* can be considered as a submodule of the free left module of *m*-tuples over the ring of polynomial partial differential operators, where *m* is the number of unknown functions of the linear system of PDEs. A partial derivative of an unknown function occurring in a linear PDE is then given by the corresponding partial differential operator in the respective entry of a tuple. This free module of *m*-tuples is graded by the standard grading (maximal degree of the components) and the submodule of the leading derivatives of elements in *M* inherits a grading from this graded free module. Note, this submodule, and therefore also its Hilbert series, depends on the ordering of derivatives chosen in the call of JanetBasis. ***PDEHilbertSeries*** returns the Hilbert series of the submodule of leading derivatives of elements in *M*.
- The output has the form $\sum_{i=0}^{\infty} d_i v^i$, where the d_i are the dimensions of the homogeneous components of the module of leading derivatives of *M*.
- The default name of the indeterminate in the Hilbert series is 's'. To use the subs command later on the indeterminate, one has to explicitly give a name to the indeterminate.
- To enumerate rather than count the the leading derivatives, cf. PDEBasis. For the Hilbert series counting free Taylor coefficients of the general solution of the linear PDE system, cf. HilbertSeries.

Examples:

```
> with(Janet):
```

Example 1:

```
> ivar := [x,y,t]; dvar := [u];
```

$$ivar := [x, y, t]$$

$$dvar := [u]$$

```
> L := [diff(u(x,y,t),x$2) - diff(u(x,y,t),t$2), diff(u(x,y,t),y)-u(x,y,t)];
```

$$L := \left[\left(\frac{\partial^2}{\partial x^2}\,u(x,y,t)\right)-\left(\frac{\partial^2}{\partial t^2}\,u(x,y,t)\right),\left(\frac{\partial}{\partial y}\,u(x,y,t)\right)-u(x,y,t)\right]$$

```
> JanetBasis(L, ivar, dvar):
> PrincDeriv();
```

$$\left[\left(\frac{\partial}{\partial y}\,u(x,y,t)\right)-u(x,y,t),[*,y,t],\frac{\partial}{\partial y}\,u(x,y,t)\right]$$

$$\left[-\left(\frac{\partial}{\partial x}\,u(x,y,t)\right)+\left(\frac{\partial^2}{\partial y\,\partial x}\,u(x,y,t)\right),[*,y,t],\frac{\partial^2}{\partial y\,\partial x}\,u(x,y,t)\right]$$

$$\left[\left(\frac{\partial^2}{\partial x^2}\,u(x,y,t)\right)-\left(\frac{\partial^2}{\partial t^2}\,u(x,y,t)\right),[x,y,t],\frac{\partial^2}{\partial x^2}\,u(x,y,t)\right]$$

```
> PDEBasis(ivar, dvar);
```

$$\frac{y}{(1-y)(1-t)}+\frac{xy}{(1-y)(1-t)}+\frac{x^2}{(1-x)(1-y)(1-t)}$$

```
> PDEBasis(ivar, dvar, "subs"=t);
```

$$\frac{t}{(1-t)^2}+\frac{t^2}{(1-t)^2}+\frac{t^2}{(1-t)^3}$$

```
> PDEHilbertSeries("var"=t);
```

$$\frac{t}{(1-t)^2}+\frac{t^2}{(1-t)^2}+\frac{t^2}{(1-t)^3}$$

```
> taylor(%, t=0, 10);
```

$$t+4t^2+8t^3+13t^4+19t^5+26t^6+34t^7+43t^8+53t^9+O(t^{10})$$

```
> PDEHilbertFunction(1);
```

$$1$$

```
> PDEHilbertFunction(2);
```

$$4$$

```
> PDEHilbertFunction("");
Dim(M.s) = 0, for s < 1
Dim(M.1) = 1
Dim(M.s) = 3/2*s-1+1/2*s^2, for s >= 2
> PDEHilbertPolynomial(s);
```

$$\frac{3}{2}s-1+\frac{1}{2}s^2$$

Example 2:

```
> ivar := [x,y]; dvar := [u,v];
```

$$ivar := [x, y]$$

$$dvar := [u, v]$$

```
> L := [diff(u(x,y),x$2) - diff(v(x,y),y$2), diff(u(x,y),y)-v(x,y)];
```

$$L := \left[\left(\frac{\partial^2}{\partial x^2}u(x,y)\right)-\left(\frac{\partial^2}{\partial y^2}v(x,y)\right),\left(\frac{\partial}{\partial y}u(x,y)\right)-v(x,y)\right]$$

```
> JanetBasis(L, ivar, dvar):
> PrincDeriv();
```

$$\left[\left(\frac{\partial}{\partial y}u(x,y)\right)-v(x,y),[*,y],\frac{\partial}{\partial y}u(x,y)\right]$$

$$\left[\left(\frac{\partial^2}{\partial y\,\partial x}u(x,y)\right)-\left(\frac{\partial}{\partial x}v(x,y)\right),[*,y],\frac{\partial^2}{\partial y\,\partial x}u(x,y)\right]$$

$$\left[\left(\frac{\partial^2}{\partial x^2}u(x,y)\right)-\left(\frac{\partial^2}{\partial y^2}v(x,y)\right),[x,y],\frac{\partial^2}{\partial x^2}u(x,y)\right]$$

$$\left[-\left(\frac{\partial^2}{\partial x^2}v(x,y)\right)+\left(\frac{\partial^3}{\partial y^3}v(x,y)\right),[x,y],\frac{\partial^3}{\partial y^3}v(x,y)\right]$$

```
> PDEBasis(ivar, dvar);
```

$$\left[\frac{x^2}{(1-x)(1-y)}+\frac{xy}{1-y}+\frac{y}{1-y},\frac{y^3}{(1-x)(1-y)}\right]$$

```
> PDEBasis(ivar, dvar, "subs"=t);
```

$$\left[\frac{t^2}{(1-t)^2}+\frac{t^2}{1-t}+\frac{t}{1-t},\frac{t^3}{(1-t)^2}\right]$$

```
> PDEHilbertSeries("var"=t);
```

$$\frac{t^2}{(1-t)^2}+\frac{t^2}{1-t}+\frac{t}{1-t}+\frac{t^3}{(1-t)^2}$$

```
> taylor(%, t=0, 10);
```

$$t+3t^2+5t^3+7t^4+9t^5+11t^6+13t^7+15t^8+17t^9+O(t^{10})$$

```
> PDEHilbertFunction(1);
```

$$1$$

```
> PDEHilbertFunction(2);
```

$$3$$

```
> PDEHilbertFunction("");
Dim(M.s) = 0, for s < 1
```

```
Dim(M.1) = 1
Dim(M.2) = 3
Dim(M.s) = -1+2*s, for s >= 3
```

```
> PDEHilbertPolynomial(s);
```

$$-1+2\,s$$

See Also:

JanetBasis, PrincDeriv, ParamDeriv, HilbertSeries, HilbertPolynomial, HP, HilbertFunction, HF, IndexRegularity, CartanCharacter, PDEBasis, PDEHilbertPolynomial, PDEHP, PDEHilbertFunction, PDEHF.

Janet[ParamBaseChg] - rewrite the parametric derivatives in the new basis of the module

Calling Sequence:

ParamBaseChg(bas,JB);
ParamBaseChg(bas,JB,nbas, "");

Parameters:

`bas` - the new basis of the factor module
`JB` - the Janet basis of the factor module
`nbas` - (optional) new names for the new basis elements
`""` - (optional) flag to suppress the second component of the output

Description:

- ***ParamBaseChg*** rewrites the parametric derivatives (i.e. those which can occur in a differential expression reduced with respect to the Janet basis **`JB`**, see ParamDeriv) in the new basis **`bas`** of the factor module.
- The output consists of the following two lists: the first with the parametric derivatives expressed in the new basis, and the second with the new basis.
- The names for the elements of the new basis might be given as the optional third parameter (**`nbas`**).
- If the optional last parameter "" is assigned, the second list in the output is omitted.
- Note that the function is only applicable for finite bases **`bas`**.

Examples:

```
> with(Janet):
```

Example 1: The linearised model of a stepmotor

```
> ivar := [t];
```

$$ivar := [t]$$

```
> Dvar := [did, diq, dtheta, dvd, dvq];
```

$$Dvar := [did, diq, dtheta, dvd, dvq]$$

Define the module Σ, the submodule $[u]$ and the factor module $\Sigma/[u]$:

```
> Sigma :=
  [L*diff(did(t),t)+R*did(t)-Nr*L*diff(theta(t),t)*diq(t)-Nr*L*iq(t)*diff(dtheta(t),t)-dvd(
  t),
  Nr*L*diff(theta(t),t)*did(t)+L*diff(diq(t),t)+R*diq(t)+(Nr*L*id(t)+Km)*diff(dtheta(t),t)-
  dvq(t), -Km*diq(t)+J*diff(dtheta(t),'$'(t,2))+B*diff(dtheta(t),t)];
```

$$\Sigma := \left[L\left(\frac{d}{dt}\,did(t)\right) + R\,did(t) - Nr\,L\left(\frac{d}{dt}\,\theta(t)\right) diq(t) - Nr\,L\,iq(t)\left(\frac{d}{dt}\,dtheta(t)\right) - dvd(t),\right.$$
$$\left. Nr\,L\left(\frac{d}{dt}\,\theta(t)\right) did(t) + L\left(\frac{d}{dt}\,diq(t)\right) + R\,diq(t) + (Nr\,L\,id(t) + Km)\left(\frac{d}{dt}\,dtheta(t)\right) - dvq(t),\ -Km\,diq(t) + J\left(\frac{d^2}{dt^2}\,dtheta(t)\right) + B\left(\frac{d}{dt}\,dtheta(t)\right)\right]$$

```
> u := [dvd(t),dvq(t)];
```

$$u := [dvd(t), dvq(t)]$$

```
> Lu := [op(Sigma),op(u)];
```

$$Lu := \left[L\left(\frac{d}{dt}\,did(t)\right) + R\,did(t) - Nr\,L\left(\frac{d}{dt}\,\theta(t)\right) diq(t) - Nr\,L\,iq(t)\left(\frac{d}{dt}\,dtheta(t)\right) - dvd(t),\right.$$
$$\left. Nr\,L\left(\frac{d}{dt}\,\theta(t)\right) did(t) + L\left(\frac{d}{dt}\,diq(t)\right) + R\,diq(t) + (Nr\,L\,id(t) + Km)\left(\frac{d}{dt}\,dtheta(t)\right) - dvq(t),\ -Km\,diq(t) + J\left(\frac{d^2}{dt^2}\,dtheta(t)\right) + B\left(\frac{d}{dt}\,dtheta(t)\right)\right.$$

$$\left[\mathrm{dvd}(t), \mathrm{dvq}(t)\right]$$

```
> JLu := JanetBasis(Lu, ivar, Dvar);
```

$$JLu := \left[\left[\left[\mathrm{dvq}(t), \mathrm{dvd}(t), Nr\left(\frac{d}{dt}\theta(t)\right)\mathrm{did}(t) + \frac{R\,\mathrm{diq}(t)}{L} + \left(\frac{d}{dt}\mathrm{diq}(t)\right) + \frac{(Nr\,L\,\mathrm{id}(t) + Km)\left(\frac{d}{dt}\mathrm{dtheta}(t)\right)}{L},\right.\right.\right.$$

$$\left.\left.\left.\frac{R\,\mathrm{did}(t)}{L} + \left(\frac{d}{dt}\mathrm{did}(t)\right) - Nr\left(\frac{d}{dt}\theta(t)\right)\mathrm{diq}(t) - Nr\,\mathrm{iq}(t)\left(\frac{d}{dt}\mathrm{dtheta}(t)\right), -\frac{Km\,\mathrm{diq}(t)}{J} + \frac{B\left(\frac{d}{dt}\mathrm{dtheta}(t)\right)}{J} + \left(\frac{d^2}{dt^2}\mathrm{dtheta}(t)\right)\right], [t],\right.$$

$$\left.[did, diq, dtheta, dvd, dvq]\right]$$

Define **bas**, and verify that it is a basis of the factor module $\Sigma/[u]$:

```
> bas := [dtheta(t), diff(dtheta(t),t), did(t), diff(did(t),t)];
```

$$bas := \left[\mathrm{dtheta}(t), \frac{d}{dt}\mathrm{dtheta}(t), \mathrm{did}(t), \frac{d}{dt}\mathrm{did}(t)\right]$$

```
> Lbas := [op(JLu[1]), op(bas)];
```

$$Lbas := \left[\mathrm{dvq}(t), \mathrm{dvd}(t), Nr\left(\frac{d}{dt}\theta(t)\right)\mathrm{did}(t) + \frac{R\,\mathrm{diq}(t)}{L} + \left(\frac{d}{dt}\mathrm{diq}(t)\right) + \frac{(Nr\,L\,\mathrm{id}(t) + Km)\left(\frac{d}{dt}\mathrm{dtheta}(t)\right)}{L},\right.$$

$$\frac{R\,\mathrm{did}(t)}{L} + \left(\frac{d}{dt}\mathrm{did}(t)\right) - Nr\left(\frac{d}{dt}\theta(t)\right)\mathrm{diq}(t) - Nr\,\mathrm{iq}(t)\left(\frac{d}{dt}\mathrm{dtheta}(t)\right), -\frac{Km\,\mathrm{diq}(t)}{J} + \frac{B\left(\frac{d}{dt}\mathrm{dtheta}(t)\right)}{J} + \left(\frac{d^2}{dt^2}\mathrm{dtheta}(t)\right), \mathrm{dtheta}(t), \frac{d}{dt}\mathrm{dtheta}(t),$$

$$\left.\mathrm{did}(t), \frac{d}{dt}\mathrm{did}(t)\right]$$

```
> JanetBasis(Lbas, ivar, Dvar):
> HilbertSeries(s);
```

$$0$$

Take the parametric derivatives of $\Sigma/[u]$:

```
> AssertJanetBasis(op(JLu));
```

$$\left[\left[\left[\mathrm{dvq}(t), \mathrm{dvd}(t), Nr\left(\frac{d}{dt}\theta(t)\right)\mathrm{did}(t) + \frac{R\,\mathrm{diq}(t)}{L} + \left(\frac{d}{dt}\mathrm{diq}(t)\right) + \left(\frac{Km}{L} + Nr\,\mathrm{id}(t)\right)\left(\frac{d}{dt}\mathrm{dtheta}(t)\right)\right.\right.\right.$$

$$\left.\left.\left.\frac{R\,\mathrm{did}(t)}{L} + \left(\frac{d}{dt}\mathrm{did}(t)\right) - Nr\left(\frac{d}{dt}\theta(t)\right)\mathrm{diq}(t) - Nr\,\mathrm{iq}(t)\left(\frac{d}{dt}\mathrm{dtheta}(t)\right), -\frac{Km\,\mathrm{diq}(t)}{J} + \frac{B\left(\frac{d}{dt}\mathrm{dtheta}(t)\right)}{J} + \left(\frac{d^2}{dt^2}\mathrm{dtheta}(t)\right)\right], [t],\right.$$

$$\left.[did, diq, dtheta, dvd, dvq]\right]$$

```
> PD := ParamDeriv(ivar, Dvar);
```

$$PD := \left[\mathrm{dtheta}(t), \frac{d}{dt}\mathrm{dtheta}(t), \mathrm{diq}(t), \mathrm{did}(t)\right]$$

Represent the parametric derivatives in the new basis **bas** of the factor module $\Sigma/[u]$:

```
> ParamBaseChg(bas, JLu);
```

$$\left[\mathrm{dtheta}(t) = \_X1(t), \frac{d}{dt}\mathrm{dtheta}(t) = \_X2(t), \mathrm{diq}(t) = -\frac{\mathrm{iq}(t)\,\_X2(t)}{\frac{d}{dt}\theta(t)} + \frac{R\,\_X3(t)}{L\,Nr\left(\frac{d}{dt}\theta(t)\right)} + \frac{\_X4(t)}{Nr\left(\frac{d}{dt}\theta(t)\right)}, \mathrm{did}(t) = \_X3(t)\right]$$

$$\left[\_X1(t)=\mathrm{dtheta}(t),\ \_X2(t)=\frac{d}{dt}\mathrm{dtheta}(t),\ \_X3(t)=\mathrm{did}(t),\ \_X4(t)=\frac{d}{dt}\mathrm{did}(t)\right]$$

```
> ParamBaseChg(bas, JLu, [a1,a2,a3,a4]);
```

$$\left[\mathrm{dtheta}(t)=a1,\ \frac{d}{dt}\mathrm{dtheta}(t)=a2,\ \mathrm{diq}(t)=-\frac{\mathrm{iq}(t)\,a2}{\frac{d}{dt}\theta(t)}+\frac{R\,a3}{L\,Nr\left(\frac{d}{dt}\theta(t)\right)}+\frac{a4}{Nr\left(\frac{d}{dt}\theta(t)\right)},\ \mathrm{did}(t)=a3\right],$$

$$\left[a1=\mathrm{dtheta}(t),\ a2=\frac{d}{dt}\mathrm{dtheta}(t),\ a3=\mathrm{did}(t),\ a4=\frac{d}{dt}\mathrm{did}(t)\right]$$

```
> ParamBaseChg(bas, JLu, "");
```

$$\left[\mathrm{dtheta}(t)=\_X1(t),\ \frac{d}{dt}\mathrm{dtheta}(t)=\_X2(t),\ \mathrm{diq}(t)=-\frac{\mathrm{iq}(t)\,\_X2(t)}{\frac{d}{dt}\theta(t)}+\frac{R\,\_X3(t)}{L\,Nr\left(\frac{d}{dt}\theta(t)\right)}+\frac{\_X4(t)}{Nr\left(\frac{d}{dt}\theta(t)\right)},\ \mathrm{did}(t)=\_X3(t)\right]$$

Example 2:

```
> restart;
> with(Janet):
> ivar := [x,y]; dvar := [u,v];
```

$$ivar := [x, y]$$

$$dvar := [u, v]$$

```
> L := [diff(u(x,y),y)+diff(v(x,y),x), diff(u(x,y),x)-diff(v(x,y),y), diff(u(x,y),y$3)];
```

$$L := \left[\left(\frac{\partial}{\partial y}\mathrm{u}(x,y)\right)+\left(\frac{\partial}{\partial x}\mathrm{v}(x,y)\right),\ \left(\frac{\partial}{\partial x}\mathrm{u}(x,y)\right)-\left(\frac{\partial}{\partial y}\mathrm{v}(x,y)\right),\ \frac{\partial^3}{\partial y^3}\mathrm{u}(x,y)\right]$$

```
> JL := JanetBasis(L, ivar, dvar);
```

$$JL := \left[\left[\left(\frac{\partial}{\partial y}\mathrm{u}(x,y)\right)+\left(\frac{\partial}{\partial x}\mathrm{v}(x,y)\right),\ \left(\frac{\partial}{\partial x}\mathrm{u}(x,y)\right)-\left(\frac{\partial}{\partial y}\mathrm{v}(x,y)\right),\ \frac{\partial^3}{\partial y^3}\mathrm{u}(x,y),\ \frac{\partial^4}{\partial y^4}\mathrm{v}(x,y)\right],\ [x,y],\ [u,v]\right]$$

```
> HilbertSeries(t);
```

$$2+2\,t+2\,t^2+t^3$$

```
> PD := ParamDeriv(ivar, dvar);
```

$$PD := \left[\mathrm{v}(x,y),\ \frac{\partial}{\partial y}\mathrm{v}(x,y),\ \frac{\partial^2}{\partial y^2}\mathrm{v}(x,y),\ \frac{\partial^3}{\partial y^3}\mathrm{v}(x,y),\ \mathrm{u}(x,y),\ \frac{\partial}{\partial y}\mathrm{u}(x,y),\ \frac{\partial^2}{\partial y^2}\mathrm{u}(x,y)\right]$$

Take another basis of the factor module:

```
> NB := [v(x,y), -2*diff(v(x,y),y), diff(v(x,y),y$2), diff(v(x,y)-3*diff(v(x,y),y),y$3),
  u(x,y), diff(u(x,y),y),diff(u(x,y),y$2)];
```

$$NB := \left[\mathrm{v}(x,y),\ -2\left(\frac{\partial}{\partial y}\mathrm{v}(x,y)\right),\ \frac{\partial^2}{\partial y^2}\mathrm{v}(x,y),\ \left(\frac{\partial^3}{\partial y^3}\mathrm{v}(x,y)\right)-3\left(\frac{\partial^4}{\partial y^4}\mathrm{v}(x,y)\right),\ \mathrm{u}(x,y),\ \frac{\partial}{\partial y}\mathrm{u}(x,y),\ \frac{\partial^2}{\partial y^2}\mathrm{u}(x,y)\right]$$

```
> LNB := [op(L), op(NB)];
```

$$LNB := \left[\left(\frac{\partial}{\partial y}\mathrm{u}(x,y)\right)+\left(\frac{\partial}{\partial x}\mathrm{v}(x,y)\right),\ \left(\frac{\partial}{\partial x}\mathrm{u}(x,y)\right)-\left(\frac{\partial}{\partial y}\mathrm{v}(x,y)\right),\ \frac{\partial^3}{\partial y^3}\mathrm{u}(x,y),\ \mathrm{v}(x,y),\ -2\left(\frac{\partial}{\partial y}\mathrm{v}(x,y)\right),\ \frac{\partial^2}{\partial y^2}\mathrm{v}(x,y),\ \left(\frac{\partial^3}{\partial y^3}\mathrm{v}(x,y)\right)-3\left(\frac{\partial^4}{\partial y^4}\mathrm{v}(x,y)\right),\ \mathrm{u}(x,y),\ \frac{\partial}{\partial y}\mathrm{u}(x,y),\ \frac{\partial^2}{\partial y^2}\mathrm{u}(x,y)\right]$$

```
> JLNB := JanetBasis(LNB, ivar, dvar);
```

$$JLNB := [[\mathrm{v}(x,y),\ \mathrm{u}(x,y)],\ [x,y],\ [u,v]]$$

```
> HilbertSeries(t);
```

$$0$$

Rewrite the parameric derivatives in the new basis **NB**:

```
> ParamBaseChg(NB, JL);
```

$$\left[\mathrm{v}(x,y)=\_X1(x,y),\ \frac{\partial}{\partial y}\mathrm{v}(x,y)=-\frac{1}{2}\_X2(x,y),\ \frac{\partial^2}{\partial y^2}\mathrm{v}(x,y)=\_X3(x,y),\ \frac{\partial^3}{\partial y^3}\mathrm{v}(x,y)=\_X4(x,y),\ \mathrm{u}(x,y)=\_X5(x,y),\ \frac{\partial}{\partial y}\mathrm{u}(x,y)=\_X6(x,y),\ \frac{\partial^2}{\partial y^2}\mathrm{u}(x,y)=\_X7(x,y)\right],\left[\_X1(x,y)=\mathrm{v}(x,y),\ \_X2(x,y)=-2\left(\frac{\partial}{\partial y}\mathrm{v}(x,y)\right),\ \_X3(x,y)=\frac{\partial^2}{\partial y^2}\mathrm{v}(x,y),\ \_X4(x,y)=\left(\frac{\partial^3}{\partial y^3}\mathrm{v}(x,y)\right)-3\left(\frac{\partial^4}{\partial y^4}\mathrm{v}(x,y)\right)\right.$$

$$\_X5(x, y) = u(x, y),\ \_X6(x, y) = \frac{\partial}{\partial y} u(x, y),\ \_X7(x, y) = \frac{\partial^2}{\partial y^2} u(x, y)$$

See Also:

JanetBasis, PrincDeriv, ParamDeriv, InvReduce, HilbertSeries, AssertJanetBasis, BaseChg, GenCoeff.

Janet[ParamDeriv] - return the parametric derivatives of the linear PDE system of the last call of JanetBasis

Calling Sequence:

ParamDeriv(ivar,dvar,oivar,mode)

Parameters:

`ivar` - list of independent variables as in the last call of ***JanetBasis***
`dvar` - list of dependent variables as in the last call of ***JanetBasis***
`oivar` - (optional) list of the independent variables in varied order as in the last call of ***JanetBasis***
`mode` - (optional) string specifying options

Description:

- ***ParamDeriv*** returns the parametric derivatives, i.e. those derivatives whose specification at some fixed point yield the free Taylor coefficients of the solutions of the linear PDE system treated by the last call of JanetBasis. It only uses the data displayable by PrincDeriv.
- In case there are infinitely many parametric derivatives or **`mode`** contains the letter "G", a |dvar|-tuple of generating functions of the form $\frac{m}{d}$ is returned, where d is a product of some $(1-x)$, x an independent variable, and m a sum of different monomials in the x's. The geometric series expansion of this enumerates the parametric derivatives of the i-th dependent variable u, if $\frac{m}{d}$ is the i-th component, in the following sense: $x_1\text{^}a_1 \ldots x_n\text{^}a_n$ occurs in the expansion if and only if $\mathrm{diff}(u(x_1,\ldots,x_n),x_1\$a_1,\ldots,x_n\$a_n)$ is a parametric derivative. If **`dvar`** contains only one dependent variable, then the generating function is returned instead of the 1-tuple.
- In case there are only finitely many parametric derivatives and **`mode`** is omitted or does not contain the letter "G", only the list of these is returned. It is sorted using the degree reverse lexicographical ordering (higher priority for comparison is given to the dependent variables).
- If the user specified an ordering of the independent variables different from **`ivar`** in the last call of JanetBasis using the optional parameter **`oivar`**, then the same parameter **`oivar`** has to be given to ***ParamDeriv***.
- The optional argument **`mode`** is a string which may contain the letters "C" and "G".
- If the letter "G" is present in **`mode`**, then ***ParamDeriv*** is forced to return a generating function as described above even if there are only finitely many parametric derivatives (cf. Example 2 below).
- If there are infinitely many parametric derivatives and the letter "C" is contained in **`mode`**, then the parametric derivatives corresponding to the numerators m of the resulting generating function are returned in a list (cf. Example 3 below).
- ***ParamDeriv*** corresponds to the command FactorModuleBasis in the polynomial case.

Examples:

```
> with(Janet):
```

Example 1:

```
> ivar := [x,y]; dvar := [u,v];
```

$$ivar := [x, y]$$

$$dvar := [u, v]$$

```
> L := [diff(u(x,y),x)-diff(v(x,y),y),diff(u(x,y),y)+diff(v(x,y),x)];
```

$$L := \left[\left(\frac{\partial}{\partial x}\mathrm{u}(x,y)\right)-\left(\frac{\partial}{\partial y}\mathrm{v}(x,y)\right),\left(\frac{\partial}{\partial y}\mathrm{u}(x,y)\right)+\left(\frac{\partial}{\partial x}\mathrm{v}(x,y)\right)\right]$$

```
> JanetBasis(L, ivar, dvar);
```

$$\left[\left[\left(\frac{\partial}{\partial y}\mathrm{u}(x,y)\right)+\left(\frac{\partial}{\partial x}\mathrm{v}(x,y)\right),\left(\frac{\partial}{\partial x}\mathrm{u}(x,y)\right)-\left(\frac{\partial}{\partial y}\mathrm{v}(x,y)\right)\right],[x,y],[u,v]\right]$$

```
> ParamDeriv(ivar, dvar);
```

$$\left[\frac{1}{1-y}, \frac{1}{1-y}\right]$$

This result means that the equations determine *u* and *v* uniquely given the values of all derivatives of *u* and *v* with respect to *y* at some fixed point. In other words the partial derivatives of both *u* and *v* with respect to *y* of all orders are parametric. This agrees with the count given by

```
> HilbertSeries();
```

$$2+2\frac{s}{1-s}$$

Example 2:

Here comes an example with finite dimensional space of solutions:

```
> ivar := [x,y]; dvar := [u,v];
```

$$ivar := [x, y]$$

$$dvar := [u, v]$$

```
> L := [diff(u(x,y),y)+diff(v(x,y),x), diff(u(x,y),x)-diff(v(x,y),y), diff(u(x,y),y$3)];
```

$$L := \left[\left(\frac{\partial}{\partial y}\mathrm{u}(x,y)\right)+\left(\frac{\partial}{\partial x}\mathrm{v}(x,y)\right), \left(\frac{\partial}{\partial x}\mathrm{u}(x,y)\right)-\left(\frac{\partial}{\partial y}\mathrm{v}(x,y)\right), \frac{\partial^3}{\partial y^3}\mathrm{u}(x,y)\right]$$

```
> JanetBasis(L, ivar, dvar);
```

$$\left[\left[\left(\frac{\partial}{\partial y}\mathrm{u}(x,y)\right)+\left(\frac{\partial}{\partial x}\mathrm{v}(x,y)\right), \left(\frac{\partial}{\partial x}\mathrm{u}(x,y)\right)-\left(\frac{\partial}{\partial y}\mathrm{v}(x,y)\right), \frac{\partial^3}{\partial y^3}\mathrm{u}(x,y), \frac{\partial^4}{\partial y^4}\mathrm{v}(x,y)\right], [x, y], [u, v]\right]$$

```
> ParamDeriv(ivar, dvar);
```

$$\left[\mathrm{v}(x,y), \frac{\partial}{\partial y}\mathrm{v}(x,y), \frac{\partial^2}{\partial y^2}\mathrm{v}(x,y), \frac{\partial^3}{\partial y^3}\mathrm{v}(x,y), \mathrm{u}(x,y), \frac{\partial}{\partial y}\mathrm{u}(x,y), \frac{\partial^2}{\partial y^2}\mathrm{u}(x,y)\right]$$

```
> HilbertSeries(t);
```

$$2+2t+2t^2+t^3$$

```
> ParamDeriv(ivar, dvar, "G");
```

$$[y^2+y+1, y^3+y^2+y+1]$$

Example 3:

```
> ivar := [x,y]; dvar := [u];
```

$$ivar := [x, y]$$

$$dvar := [u]$$

```
> JanetBasis([diff(u(x,y),x,y)], ivar, dvar);
```

$$\left[\left[\frac{\partial^2}{\partial y \partial x}\mathrm{u}(x,y)\right], [x, y], [u]\right]$$

```
> ParamDeriv(ivar, dvar);
```

$$\frac{1}{1-y}+\frac{x}{1-x}$$

```
> ParamDeriv(ivar, dvar, "C");
```

$$\left[\mathrm{u}(x,y), \frac{\partial}{\partial x}\mathrm{u}(x,y)\right]$$

See Also:

JanetBasis, PrincDeriv, HilbertSeries, WeightedHilbertSeries, PDEBasis, PDEHilbertSeries, SolSeries, PolySol, FactorModuleBasis.

Janet[Parametrization] - return a parametrization of the kernel of a linear differential operator

Calling Sequence:

Parametrization(M,ivar,dvar,p)

Parameters:

`M` - differential operator in matrix form or list of linear differential expressions
`ivar` - list of independent variables
`dvar` - list of dependent variables
`p` - (optional) equation "parameters" = name or list of names for free parameters

Description:

- ***Parametrization*** constructs a parametrization of the kernel of the linear differential operator given by **M**, i.e. another linear differential operator such that the composition (cf. CmpOp) of the first operator with the second operator is zero.
- The first parameter **M** is expected to be a linear differential operator in matrix form (cf. AppOp) with independent variables **ivar** or a list of linear differential expressions with independent variables **ivar** and dependent variables **dvar**. In the second case, ***Parametrization*** extracts the linear differential operator from **M** and computes a parametrization of the kernel of this operator.
- If **M** is given as differential operator in matrix form, then the argument **dvar** can be omitted. In this case, ***Parametrization*** returns a differential operator in matrix form.
- If **M** is a list of linear differential expressions, then the constructed differential operator is applied (cf. AppOp) to a column vector of free parameters and the resulting list of linear differential expressions is returned. The names for these free parameters can be specified via the optional fourth parameter **p**. The default name is ϕ. If the number r of free parameters is greater than one, then the vector of free parameters is defined by the list of serially numbered names $n1, ..., nr$, where n is the name given as right hand side of **p**. Finally, if one happens to know the number of parameters in advance, then the name of every free parameter can be specified individually by passing a list of names as right hand side of the equation **p**.

Examples:

```
> with(Janet):
```

Example 1: Poincare sequence

```
> ivar := [x,y,z];
```

$$ivar := [x, y, z]$$

```
> M1 := Pol2Op([[x,y,z]], ivar);
```

$$M1 := [[[1, [x]]] \quad [[1, [y]]] \quad [[1, [z]]]]$$

```
> M2 := Parametrization(M1, ivar);
```

$$M2 := \begin{bmatrix} [[1, [y]]] & [[1, [z]]] & 0 \\ [[-1, [x]]] & 0 & [[1, [z]]] \\ 0 & [[-1, [x]]] & [[-1, [y]]] \end{bmatrix}$$

```
> M3 := Parametrization(M2, ivar);
```

$$M3 := \begin{bmatrix} [[1, [z]]] \\ [[-1, [y]]] \\ [[1, [x]]] \end{bmatrix}$$

```
> CmpOp(M1, M2, ivar);
```

$$[0 \quad 0 \quad 0]$$

```
> CmpOp(M2, M3, ivar);
```

$$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Example 2: Maxwell equations

```
> ivar := [t,x,y,z]; dvar := [B1,B2,B3,E1,E2,E3];
```

$$ivar := [t, x, y, z]$$

$$dvar := [B1, B2, B3, E1, E2, E3]$$

```
> M := Pol2Op([[t,0,0,0,-z,y], [0,t,0,z,0,-x], [0,0,t,-y,x,0], [x,y,z,0,0,0]], ivar);
```

$$M := \begin{bmatrix} [[1,[t]]] & 0 & 0 & 0 & [[-1,[z]]] & [[1,[y]]] \\ 0 & [[1,[t]]] & 0 & [[1,[z]]] & 0 & [[-1,[x]]] \\ 0 & 0 & [[1,[t]]] & [[-1,[y]]] & [[1,[x]]] & 0 \\ [[1,[x]]] & [[1,[y]]] & [[1,[z]]] & 0 & 0 & 0 \end{bmatrix}$$

```
> Parametrization(M, ivar);
```

$$\begin{bmatrix} [[-1,[y]]] & [[-1,[z]]] & 0 & 0 \\ [[1,[x]]] & 0 & [[-1,[z]]] & 0 \\ 0 & [[1,[x]]] & [[1,[y]]] & 0 \\ 0 & 0 & [[1,[t]]] & [[1,[x]]] \\ 0 & [[-1,[t]]] & 0 & [[1,[y]]] \\ [[1,[t]]] & 0 & 0 & [[1,[z]]] \end{bmatrix}$$

Repeat the same computation with a list of linear differential expressions as input:

```
> L := AppOp(M, dvar, ivar, dvar);
```

$$L := \left[\left(\frac{\partial}{\partial t}\mathrm{B1}(t,x,y,z)\right)-\left(\frac{\partial}{\partial z}\mathrm{E2}(t,x,y,z)\right)+\left(\frac{\partial}{\partial y}\mathrm{E3}(t,x,y,z)\right),\left(\frac{\partial}{\partial t}\mathrm{B2}(t,x,y,z)\right)+\left(\frac{\partial}{\partial z}\mathrm{E1}(t,x,y,z)\right)-\left(\frac{\partial}{\partial x}\mathrm{E3}(t,x,y,z)\right),\right.$$
$$\left.\left(\frac{\partial}{\partial t}\mathrm{B3}(t,x,y,z)\right)-\left(\frac{\partial}{\partial y}\mathrm{E1}(t,x,y,z)\right)+\left(\frac{\partial}{\partial x}\mathrm{E2}(t,x,y,z)\right),\left(\frac{\partial}{\partial x}\mathrm{B1}(t,x,y,z)\right)+\left(\frac{\partial}{\partial y}\mathrm{B2}(t,x,y,z)\right)+\left(\frac{\partial}{\partial z}\mathrm{B3}(t,x,y,z)\right)\right]$$

Specify the names for the free parameters:

```
> Parametrization(L, ivar, dvar, "parameters"=[A1,A2,A3,V]);
```

$$\left[-\left(\frac{\partial}{\partial y}\mathrm{A1}(t,x,y,z)\right)-\left(\frac{\partial}{\partial z}\mathrm{A2}(t,x,y,z)\right),\left(\frac{\partial}{\partial x}\mathrm{A1}(t,x,y,z)\right)-\left(\frac{\partial}{\partial z}\mathrm{A3}(t,x,y,z)\right),\left(\frac{\partial}{\partial x}\mathrm{A2}(t,x,y,z)\right)+\left(\frac{\partial}{\partial y}\mathrm{A3}(t,x,y,z)\right)\right.$$
$$\left.\left(\frac{\partial}{\partial t}\mathrm{A3}(t,x,y,z)\right)+\left(\frac{\partial}{\partial x}\mathrm{V}(t,x,y,z)\right),-\left(\frac{\partial}{\partial t}\mathrm{A2}(t,x,y,z)\right)+\left(\frac{\partial}{\partial y}\mathrm{V}(t,x,y,z)\right),\left(\frac{\partial}{\partial t}\mathrm{A1}(t,x,y,z)\right)+\left(\frac{\partial}{\partial z}\mathrm{V}(t,x,y,z)\right)\right]$$

```
> Parametrization(L, ivar, dvar, "parameters"=psi);
```

$$\left[-\left(\frac{\partial}{\partial y}\psi 1(t,x,y,z)\right)-\left(\frac{\partial}{\partial z}\psi 2(t,x,y,z)\right),\left(\frac{\partial}{\partial x}\psi 1(t,x,y,z)\right)-\left(\frac{\partial}{\partial z}\psi 3(t,x,y,z)\right),\left(\frac{\partial}{\partial x}\psi 2(t,x,y,z)\right)+\left(\frac{\partial}{\partial y}\psi 3(t,x,y,z)\right)\right.$$
$$\left.\left(\frac{\partial}{\partial t}\psi 3(t,x,y,z)\right)+\left(\frac{\partial}{\partial x}\psi 4(t,x,y,z)\right),-\left(\frac{\partial}{\partial t}\psi 2(t,x,y,z)\right)+\left(\frac{\partial}{\partial y}\psi 4(t,x,y,z)\right),\left(\frac{\partial}{\partial t}\psi 1(t,x,y,z)\right)+\left(\frac{\partial}{\partial z}\psi 4(t,x,y,z)\right)\right]$$

Example 3:

```
> ivar := [x,y]; dvar := [u,v];
```

$$ivar := [x, y]$$

$$dvar := [u, v]$$

```
> L := [diff(u(x,y),y)+diff(v(x,y),x)];
```

$$L := \left[\left(\frac{\partial}{\partial y}\mathrm{u}(x,y)\right)+\left(\frac{\partial}{\partial x}\mathrm{v}(x,y)\right)\right]$$

The default name for free parameters is ϕ:

```
> Parametrization(L, ivar, dvar);
```

$$\left[-\left(\frac{\partial}{\partial x}\phi(x,y)\right), \frac{\partial}{\partial y}\phi(x,y)\right]$$

An example with non-constant coefficients:

```
> L := [x*diff(u(x,y),y)+y*diff(v(x,y),x)];
```

$$L := \left[x\left(\frac{\partial}{\partial y}\mathrm{u}(x,y)\right) + y\left(\frac{\partial}{\partial x}\mathrm{v}(x,y)\right)\right]$$

```
> L2 := Parametrization(L, ivar, dvar);
```

$$L2 := \left[-y^2 x\left(\frac{\partial}{\partial x}\phi(x,y)\right) - 2y^2\phi(x,y), y x^2\left(\frac{\partial}{\partial y}\phi(x,y)\right) + 2x^2\phi(x,y)\right]$$

Verify that *L2* gives a parametrization of the kernel of the linear differential operator defined by *L*:

```
> M := Diff2Op(L, ivar, dvar);
```

$$M := [[[x, [y]]] \quad [[y, [x]]]]$$

```
> P := Diff2Op(L2, ivar, [phi]);
```

$$P := \begin{bmatrix} [[-y^2 x, [x]], [-2y^2, [\ ]]] \\ [[y x^2, [y]], [2x^2, [\ ]]] \end{bmatrix}$$

```
> CmpOp(M, P, ivar);
```

$$[\ 0]$$

See Also:

JanetBasis, PrincDeriv, InvReduce, CompCond, CompCondBasis, SyzOp, Resolution, Torsion, Ext1, Extn, AutonomEq, FlatOutput, Diff2Op, Pol2Op, AppOp, CmpOp, Diff2D, D2Diff

Janet[Pol2Diff] - return the linear differential expression corresponding to a polynomial expression

Calling Sequence:

Pol2Diff(L,ivar,dvar,rvar)

Parameters:

L	-	polynomial, list of polynomials, or list of tuples of polynomials (possibly extended by right hand sides)
ivar	-	list of indeterminates of the polynomial ring (which become the independent variables)
dvar	-	list of dependent variables
rvar	-	(optional) list of symbols

Description:

- In the simplest case, where **L** is a polynomial in **ivar** and **dvar** consists of one variable *u* only, ***Pol2Diff*** translates **L** into the result of applying the linear partial differential operator associated with **L** to *u*, where the operator associated with **L** is obtained from **L** by substituting each independent variable *x* by the partial derivative with respect to *x*.
- All other cases are variations of the above: **dvar** must match the length of the tuples of polynomials in **L**. In this case ***Pol2Diff*** does componentwise the same as the simple version, i. e. applies the differential operator obtained from the *i*-th component of some tuple in **L** to the *i*-th dependent variable in **dvar**. Note, in case **L** is a *k*-tuple of polynomials, **dvar** decides whether this is translated into a *k*-tuple of differential expressions (in case **dvar** consists of one element) or into one differential expression with *k* dependent variables (in case **dvar** has *k* elements).
- A further variant for ***Pol2Diff*** is to input an equation or a list of equations, the left hand sides being as above (a polynomial or a tuple of polynomials in **ivar**) and the right hand sides being either linear combinations of new variables a_i with polynomials in **ivar** as coefficients or tuples of polynomials in **ivar**. These equations are translated into differences of the above output by derivatives of new functions corresponding to the a_i resp. to the positions in the tuples on the right hand side, the derivatives of course in accordance to the polynomial coefficients. The list of names for the new variables must be provided as fourth parameter **rvar** if the right hand sides are given as linear combinations of these variables; it is optional in case the right hand sides are tuples of polynomials in **ivar** (cf. Examples 3 and 4 below).
- A typical use of ***Pol2Diff*** is to apply the command to an "expensive" involutive basis obtained by InvolutiveBasisFast to get the Janet basis for the corresponding linear PDE system with constant coefficients quickly, cf. AssertJanetBasis.
- There is a command Diff2Pol which is inverse to ***Pol2Diff***.

Examples:

```
> with(Janet):
```

Example 1:

```
> ivar := [x,y,z];
```

$$ivar := [x, y, z]$$

```
> L := [x^2+y, y^2+z];
```

$$L := [x^2 + y, y^2 + z]$$

If **dvar** consists of one variable only, then **L** is translated into a tuple of differential expressions:

```
> Pol2Diff(L, ivar, [u]);
```

$$\left[\left(\frac{\partial^2}{\partial x^2}\,u(x,y,z)\right)+\left(\frac{\partial}{\partial y}\,u(x,y,z)\right),\left(\frac{\partial}{\partial z}\,u(x,y,z)\right)+\left(\frac{\partial^2}{\partial y^2}\,u(x,y,z)\right)\right]$$

If the number of variables in **dvar** equals the number of entries in **L**, then **L** is translated into one differential expression:

```
> Pol2Diff(L, ivar, [u,v]);
```

$$\left[\left(\frac{\partial^2}{\partial x^2}\,u(x,y,z)\right)+\left(\frac{\partial}{\partial y}\,u(x,y,z)\right)+\left(\frac{\partial}{\partial z}\,v(x,y,z)\right)+\left(\frac{\partial^2}{\partial y^2}\,v(x,y,z)\right)\right]$$

Example 2: Translating lists of lists of polynomials

```
> ivar := [x,y,z];
```

$$ivar := [x, y, z]$$

```
> L := [[x^2+y, y^2+z], [y+z, x]];
```

$$L := [[x^2 + y, y^2 + z], [y + z, x]]$$

```
> Pol2Diff(L, ivar, [u,v]);
```

$$\left[\left(\frac{\partial^2}{\partial x^2}\,u(x,y,z)\right)+\left(\frac{\partial}{\partial y}\,u(x,y,z)\right)+\left(\frac{\partial}{\partial z}\,v(x,y,z)\right)+\left(\frac{\partial^2}{\partial y^2}\,v(x,y,z)\right),\left(\frac{\partial}{\partial z}\,u(x,y,z)\right)+\left(\frac{\partial}{\partial y}\,u(x,y,z)\right)+\left(\frac{\partial}{\partial x}\,v(x,y,z)\right)\right]$$

Example 3: Translating equations into affine differential expressions

```
> ivar := [x,y,z];
```

$$ivar := [x, y, z]$$

```
> L := [x^2+y=a, y^2+z=b];
```

$$L := [x^2 + y = a, y^2 + z = b]$$

```
> Pol2Diff(L, ivar, [u], [a,b]);
```

$$\left[\left(\frac{\partial^2}{\partial x^2}\,u(x,y,z)\right)+\left(\frac{\partial}{\partial y}\,u(x,y,z)\right)-a(x,y,z),\left(\frac{\partial}{\partial z}\,u(x,y,z)\right)+\left(\frac{\partial^2}{\partial y^2}\,u(x,y,z)\right)-b(x,y,z)\right]$$

```
> L := [x^2+y=[2*y,0], y^2+z=[0,x]];
```

$$L := [x^2 + y = [2y, 0], y^2 + z = [0, x]]$$

```
> Pol2Diff(L, ivar, [u]);
```

$$\left[\left(\frac{\partial^2}{\partial x^2}\,u(x,y,z)\right)+\left(\frac{\partial}{\partial y}\,u(x,y,z)\right)-2\left(\frac{\partial}{\partial y}\,a1(x,y,z)\right)\left(\frac{\partial}{\partial z}\,u(x,y,z)\right)+\left(\frac{\partial^2}{\partial y^2}\,u(x,y,z)\right)-\left(\frac{\partial}{\partial x}\,a2(x,y,z)\right)\right]$$

```
> Pol2Diff(L, ivar, [u], [a,b]);
```

$$\left[\left(\frac{\partial^2}{\partial x^2}\,u(x,y,z)\right)+\left(\frac{\partial}{\partial y}\,u(x,y,z)\right)-2\left(\frac{\partial}{\partial y}\,a(x,y,z)\right)\left(\frac{\partial}{\partial z}\,u(x,y,z)\right)+\left(\frac{\partial^2}{\partial y^2}\,u(x,y,z)\right)-\left(\frac{\partial}{\partial x}\,b(x,y,z)\right)\right]$$

Example 4: Using ***InvolutiveBasisFast*** for systems with constant coefficients

```
> ivar := [x,y,z];
```

$$ivar := [x, y, z]$$

```
> L := [diff(u(x,y,z),y)+diff(u(x,y,z),x$2)+diff(v(x,y,z),z)-b(x,y,z),
  diff(u(x,y,z),z)+diff(u(x,y,z),y$2)-a(x,y,z)];
```

$$L := \left[\left(\frac{\partial^2}{\partial x^2}\,u(x,y,z)\right)+\left(\frac{\partial}{\partial y}\,u(x,y,z)\right)+\left(\frac{\partial}{\partial z}\,v(x,y,z)\right)-b(x,y,z),\left(\frac{\partial}{\partial z}\,u(x,y,z)\right)+\left(\frac{\partial^2}{\partial y^2}\,u(x,y,z)\right)-a(x,y,z)\right]$$

```
> P := Diff2Pol(L, ivar, [u,v]);
```

$$P := [[y + x^2, z] = [0, 1], [z + y^2, 0] = [1, 0]]$$

```
> with(Involutive):
> J := InvolutiveBasisFast(P, ivar);
```

$$J := [[z + y^2, 0] = [1, 0], [y + x^2, z] = [0, 1], [0, y^2 z + z^2] = [-x^2 - y, z + y^2], [xy^2 + xz, 0] = [x, 0]]$$

```
> Pol2Diff(J, ivar, [u,v]);
```

$$\left[\left(\frac{\partial}{\partial z}\,u(x,y,z)\right)+\left(\frac{\partial^2}{\partial y^2}\,u(x,y,z)\right)-a1(x,y,z),\left(\frac{\partial^2}{\partial x^2}\,u(x,y,z)\right)+\left(\frac{\partial}{\partial y}\,u(x,y,z)\right)+\left(\frac{\partial}{\partial z}\,v(x,y,z)\right)-a2(x,y,z),\right.$$
$$\left(\frac{\partial^3}{\partial z\,\partial y^2}\,v(x,y,z)\right)+\left(\frac{\partial^2}{\partial z^2}\,v(x,y,z)\right)+\left(\frac{\partial^2}{\partial x^2}\,a1(x,y,z)\right)+\left(\frac{\partial}{\partial y}\,a1(x,y,z)\right)-\left(\frac{\partial}{\partial z}\,a2(x,y,z)\right)-\left(\frac{\partial^2}{\partial y^2}\,a2(x,y,z)\right),$$
$$\left.\left(\frac{\partial^3}{\partial y^2\,\partial x}\,u(x,y,z)\right)+\left(\frac{\partial^2}{\partial z\,\partial x}\,u(x,y,z)\right)-\left(\frac{\partial}{\partial x}\,a1(x,y,z)\right)\right]$$

```
> Pol2Diff(J, ivar, [u,v], [a,b]);
```

$$\left[\left(\frac{\partial}{\partial z}\,u(x,y,z)\right)+\left(\frac{\partial^2}{\partial y^2}\,u(x,y,z)\right)-a(x,y,z),\left(\frac{\partial^2}{\partial x^2}\,u(x,y,z)\right)+\left(\frac{\partial}{\partial y}\,u(x,y,z)\right)+\left(\frac{\partial}{\partial z}\,v(x,y,z)\right)-b(x,y,z),\right.$$
$$\left.\left(\frac{\partial^3}{\partial z\,\partial y^2}\,v(x,y,z)\right)+\left(\frac{\partial^2}{\partial z^2}\,v(x,y,z)\right)+\left(\frac{\partial^2}{\partial x^2}\,a(x,y,z)\right)+\left(\frac{\partial}{\partial y}\,a(x,y,z)\right)-\left(\frac{\partial}{\partial z}\,b(x,y,z)\right)-\left(\frac{\partial^2}{\partial y^2}\,b(x,y,z)\right),\right.$$

$$\left[\left[\left(\frac{\partial^3}{\partial y^2\,\partial x}\,u(x,y,z)\right)+\left(\frac{\partial^2}{\partial z\,\partial x}\,u(x,y,z)\right)-\left(\frac{\partial}{\partial x}\,a(x,y,z)\right)\right]\right]$$

See Also:

InvolutiveBasis, InvolutiveBasisFast, JanetBasis, Diff2Pol, Pol2Op, Diff2Op, AppOp, Diff2Ind, Ind2Diff, AppOpInd, Pol2Ind, Op2D, D2Op, Diff2D, D2Diff.

Janet[Pol2Ind] - translate polynomial into linear differential expression in jet notation

Calling Sequence:

Pol2Ind(L,ivar,dvar)

Parameters:

`L` - polynomial, list of polynomials, or list of tuples of polynomials
`ivar` - list of indeterminates of the polynomial ring (which become the independent variables)
`dvar` - list of dependent variables

Description:

- In the simplest case, where **L** is a polynomial in **ivar** and **dvar** consists of one variable *u* only, ***Pol2Ind*** translates **L** into the linear partial differential operator applied to *u* in jet notation. The command Pol2Diff gives the analogous output as differential expression.
- All other cases are variations of the above: **dvar** must match the length of the tuples of polynomials in **L**. In this case ***Pol2Ind*** does componentwise the same as the simple version, i. e. applies the differential operator obtained from the *i*-th component of some tuple in **L** to the *i*-th dependent variable in **dvar**. Note, in case **L** is a *k*-tuple of polynomials, **dvar** decides whether this is translated into a *k*-tuple of differential expressions (in case **dvar** consists of one element) or into one differential expression with *k* dependent variables (in case **dvar** has *k* elements).
- The composition of the commands Ind2Diff and Diff2Pol is inverse to ***Pol2Ind***.
- The jet notation is adapted from the Maple package jets. It is much more compact than the usual Maple notation for differential expressions. To fix ideas, let *u* be a dependent variable and *x,y,z* the independent variables. Then the jet notation for $\frac{\partial^4}{\partial z\,\partial y\,\partial y\,\partial x}\mathrm{u}(x,y,z)$ is $u_{x,y,y,z}$.

Examples:

```
> with(Janet):
> ivar := [x,y,z]; dvar := [u,v];
```

$$ivar := [x, y, z]$$

$$dvar := [u, v]$$

```
> Pol2Ind(x^2, ivar, [u]);
```

$$u_{x,x}$$

```
> L := [x^2+y, y^2+z];
```

$$L := [x^2 + y, y^2 + z]$$

```
> Pol2Ind(L, ivar, [u]);
```

$$[u_{x,x} + u_y, u_{y,y} + u_z]$$

```
> Pol2Ind(L, ivar, dvar);
```

$$u_{x,x} + u_y + v_{y,y} + v_z$$

```
> Pol2Diff(L, ivar, dvar);
```

$$\left[\left(\frac{\partial^2}{\partial x^2}\mathrm{u}(x,y,z)\right)+\left(\frac{\partial}{\partial y}\mathrm{u}(x,y,z)\right)+\left(\frac{\partial^2}{\partial y^2}\mathrm{v}(x,y,z)\right)+\left(\frac{\partial}{\partial z}\mathrm{v}(x,y,z)\right)\right]$$

```
> L := [[x^2+y, x], [z^2, y^2+z]];
```

$$L := [[x^2 + y, x], [z^2, y^2 + z]]$$

```
> Lin := Pol2Ind(L, ivar, dvar);
```

$$Lin := [u_{x,x} + u_y + v_x, u_{z,z} + v_{y,y} + v_z]$$

```
> Lex := Ind2Diff(Lin, ivar, dvar);
```

$$Lex := \left[\left(\frac{\partial^2}{\partial x^2}\mathrm{u}(x,y,z)\right)+\left(\frac{\partial}{\partial y}\mathrm{u}(x,y,z)\right)+\left(\frac{\partial}{\partial x}\mathrm{v}(x,y,z)\right),\left(\frac{\partial^2}{\partial z^2}\mathrm{u}(x,y,z)\right)+\left(\frac{\partial^2}{\partial y^2}\mathrm{v}(x,y,z)\right)+\left(\frac{\partial}{\partial z}\mathrm{v}(x,y,z)\right)\right]$$

```
> Pol2Diff(L, ivar, dvar);
```

$$\left[\left(\frac{\partial^2}{\partial x^2}\,u(x,y,z)\right)+\left(\frac{\partial}{\partial y}\,u(x,y,z)\right)+\left(\frac{\partial}{\partial x}\,v(x,y,z)\right),\left(\frac{\partial^2}{\partial z^2}\,u(x,y,z)\right)+\left(\frac{\partial^2}{\partial y^2}\,v(x,y,z)\right)+\left(\frac{\partial}{\partial z}\,v(x,y,z)\right)\right]$$

```
> Diff2Ind(Lex, ivar, dvar);
```

$$[u_{x,x}+u_y+v_x,\ u_{z,z}+v_{y,y}+v_z]$$

```
> Diff2Pol(Lex, ivar, dvar);
```

$$[[x^2+y,\ x],\ [z^2,\ y^2+z]]$$

See Also:

Diff2Ind, Ind2Diff, AppOpInd, Pol2Op, Diff2Op, AppOp, Diff2Pol, Pol2Diff, Op2D, D2Op, Diff2D, D2Diff.

Janet[Pol2Op] - turn polynomial into linear differential operator with constant coefficients

Calling Sequence:

Pol2Op(L,ivar,dvar)

Parameters:

`L`	-	polynomial, list of polynomials, or list of tuples of polynomials
`ivar`	-	list of independent variables
`dvar`	-	(optional) list of dependent variables

Description:

- ***Pol2Op*** returns a linear differential operator in matrix form by plugging in D_{x_i} for each independent variable x_i in **L**, where D_{x_i} is the partial differential operator w. r. t. x_i.
- If **L** consists of inhomogeneous equations, then the right hand sides are ignored.
- The entries of the output matrix have the form $[[c_1, [...]], ..., [c_r, [...]]]$, where the dots in the innermost brackets are sequences of independent variables from **ivar** representing partial derivatives w. r. t. these variables, and the c_i are the coefficients for these, i. e. functions in **ivar**. The whole bracket represents the sum of these items, i. e. the linear combination of the above operators. Note, if an innermost bracket is empty, it represents the identity operator.
- This command is realized as the composition of Pol2Diff and Diff2Op. The list **dvar** of dependent variables is used to call these commands. If **dvar** is not given, ***Pol2Op*** chooses a suitable list **dvar**.

Examples:

```
> with(Janet):
> ivar := [x,y,z];
```

$$ivar := [x, y, z]$$

```
> Pol2Op(x^2+y^2+z^2, ivar);
```

$$[[1, [x, x]], [1, [y, y]], [1, [z, z]]]$$

```
> Pol2Op(x^2+y^2+z^2-r, ivar);
```

$$[[1, [x, x]], [1, [y, y]], [1, [z, z]], [-r, [\ ]]]$$

```
> L := [[0,-z,y],[z,0,-x],[-y,x,0]];
```

$$L := [[0, -z, y], [z, 0, -x], [-y, x, 0]]$$

```
> Pol2Op(L, ivar);
```

$$\begin{bmatrix} 0 & [[-1, [z]]] & [[1, [y]]] \\ [[1, [z]]] & 0 & [[-1, [x]]] \\ [[-1, [y]]] & [[1, [x]]] & 0 \end{bmatrix}$$

```
> AppOp(%, [u,v,w], ivar, [u,v,w]);
```

$$\left[-\left(\frac{\partial}{\partial z} v(x, y, z)\right)+\left(\frac{\partial}{\partial y} w(x, y, z)\right), \left(\frac{\partial}{\partial z} u(x, y, z)\right)-\left(\frac{\partial}{\partial x} w(x, y, z)\right), -\left(\frac{\partial}{\partial y} u(x, y, z)\right)+\left(\frac{\partial}{\partial x} v(x, y, z)\right)\right]$$

```
> Diff2Pol(%, ivar, [u,v,w]);
```

$$[[0, -z, y], [z, 0, -x], [-y, x, 0]]$$

See Also:

Pol2Diff, Diff2Pol, Diff2Op, AppOp, CmpOp, JAdjoint, Pol2Ind, Diff2Ind, Ind2Diff, AppOpInd, Op2D, D2Op, Diff2D, D2Diff

Janet[PolySol] - polynomial solutions of a linear system of partial differential equations

Calling Sequence:

PolySol(G,deg,p,b,opt)

Parameters:

G	-	list of Janet basis, independent variables, dependent variables (i. e. the output of JanetBasis)
deg	-	natural number bounding the order or list of bounds (according to dependent variables)
p	-	(optional) list of values for the independent variables fixing the centre of expansion
b	-	(optional) unevaluated name for output in alternative form (as basis)
opt	-	(optional) equation "constname"=beginning of name for constants

Description:

- ***PolySol*** returns the polynomial solutions up to degree **deg** of the linear system of partial differential equations given by its Janet basis in **G**.
- The result of ***PolySol*** is a sequence of two lists, where the first list contains the polynomial solutions up to the specified degree for all dependent variables and the second list contains the free coefficients $Cj_{i_1 .. i_n}$ which occur in these polynomials. The name *C* is changed to *X*, say, if the equation "constname"=*X* is given in **opt**.
- One can specify the degree of the solutions by specifying **deg** either as a natural number, or as a list of *k* natural numbers, where *k* is the number of dependent variables, or by a *k*-tuple of *n*-tuples, where *n* is the number of independent variables. In the first case the degree is less than or equal to **deg**. In the second case the degrees are bounded by op(**deg**), where the *i*-th component of **deg** specifies the maximal degree of the *i*-th component function of the solution. Finally the third case refines the second case insofar as an upper bound for the degrees in each independent variable for each dependent variable is specified by **deg**. In this case **deg** is a list of length *k* of lists of length *n*.
- Note, usually the degrees are taken in the ordinary sense, but like in the command JanetBasis the user can also specify the degree for each independent variable taken from there. In this case, the output of ***JanetBasis*** cannot be taken as it comes out of the program but the third entry must be taken over from the input of ***JanetBasis***, i. e. it has the form $[x_1=d_1, ..., x_n=d_n]$. Also a fourth parameter in **G** ranging between 1 and 4 might be given to change the order of the derivatives in the internal computation (using SolSeries).
- If the center **p** of expansion is not specified, it is zero by default. If **p** is specified, say **p**=$[p_1, ..., p_n]$, where *n* is the number of independent variables, then the expansion is taken around $(p_1, ..., p_n)$. The admissible points **p** can be determined from the result of the command ZeroSets: any point outside of the ouput set of ZeroSets is admissible.
- In case the fourth parameter **b** is given, specific solutions are stored in **b**. They are obtained from the general solutions by setting all constants (parametric derivatives at the point **p**) except for one equal to 0 and the remaining one equal to 1 in all possible ways. In case one has an inhomoneous system of linear partial differential equations, also the solution where all constants are set to zero is given (as first solution).
- If an equation "constname"=*X* is given in **opt**, then the name for the constants appearing in the polynomial solutions is changed from *C* to *X*.

Examples:

```
> with(Janet):
```

Example 1:

```
> ivar := [x,y,t]; dvar := [u];
```

$$ivar := [x, y, t]$$

$$dvar := [u]$$

```
> L := [diff(u(x,y,t),x$2) - diff(u(x,y,t),t$2), diff(u(x,y,t),x,y)-diff(u(x,y,t),y)];
```

$$L := \left[\left(\frac{\partial^2}{\partial x^2}\mathrm{u}(x,y,t)\right) - \left(\frac{\partial^2}{\partial t^2}\mathrm{u}(x,y,t)\right), \left(\frac{\partial^2}{\partial y\,\partial x}\mathrm{u}(x,y,t)\right) - \left(\frac{\partial}{\partial y}\mathrm{u}(x,y,t)\right)\right]$$

```
> J := JanetBasis(L, ivar, dvar);
```

$$J := \left[\left[\left(\frac{\partial^2}{\partial y\,\partial x}\mathrm{u}(x,y,t)\right) - \left(\frac{\partial}{\partial y}\mathrm{u}(x,y,t)\right), \left(\frac{\partial^2}{\partial x^2}\mathrm{u}(x,y,t)\right) - \left(\frac{\partial^2}{\partial t^2}\mathrm{u}(x,y,t)\right), -\left(\frac{\partial}{\partial y}\mathrm{u}(x,y,t)\right) + \left(\frac{\partial^3}{\partial y\,\partial t^2}\mathrm{u}(x,y,t)\right)\right], [x,y,t], [u]\right]$$

```
> HilbertSeries();
```

$$1 + 3s + 4s^2 + \frac{4s^3}{1-s}$$

```
> PolySol(J, 3, 'b');
```

$$\left[\mathrm{u}(x,y,t) = C1_{0,0,0} + C1_{1,0,0}\,x + C1_{0,0,1}\,t + \frac{1}{2}C1_{0,0,2}\,t^2 + C1_{1,0,1}\,xt + \frac{1}{2}C1_{0,0,2}\,x^2 + \frac{1}{6}C1_{0,0,3}\,t^3 + \frac{1}{2}C1_{1,0,2}\,xt^2 + \frac{1}{2}C1_{0,0,3}\,x^2 t + \frac{1}{6}C1_{1,0,2}\,x^3\right], [C1_{0,0,0}, C1_{1,0,0}, C1_{0,0,1}, C1_{0,0,2}, C1_{1,0,1}, C1_{0,0,3}, C1_{1,0,2}]$$

```
> b;
```

$$\left[[\mathrm{u}(x,y,t)=1], [\mathrm{u}(x,y,t)=x], [\mathrm{u}(x,y,t)=t], \left[\mathrm{u}(x,y,t)=\frac{t^2}{2}+\frac{x^2}{2}\right], [\mathrm{u}(x,y,t)=xt], \left[\mathrm{u}(x,y,t)=\frac{1}{6}t^3+\frac{1}{2}x^2 t\right], \left[\mathrm{u}(x,y,t)=\frac{1}{2}xt^2+\frac{1}{6}x^3\right]\right]$$

```
> PolySol(J, 3, [1,2,-1]);
```

$$\left[\mathrm{u}(x,y,t) = C1_{0,0,2} - C1_{1,0,1} + \frac{2}{3}C1_{0,0,3} - \frac{2}{3}C1_{1,0,2} + C1_{0,0,1} + C1_{0,0,0} - C1_{1,0,0} + \frac{1}{6}C1_{1,0,2}\,x^3 + \frac{1}{2}C1_{1,0,2}\,xt^2 + (C1_{0,0,3} + C1_{0,0,2} + C1_{0,0,1} - C1_{1,0,2} - C1_{1,0,1})t + (C1_{1,0,0} + C1_{1,0,2} - C1_{0,0,2} + C1_{1,0,1} - C1_{0,0,3})x + \frac{1}{6}C1_{0,0,3}\,t^3 + \frac{1}{2}C1_{0,0,3}\,x^2 t + \left(\frac{1}{2}C1_{0,0,3} + \frac{1}{2}C1_{0,0,2} - \frac{1}{2}C1_{1,0,2}\right)t^2 + (C1_{1,0,1} - C1_{0,0,3} + C1_{1,0,2})xt + \left(\frac{1}{2}C1_{0,0,3} + \frac{1}{2}C1_{0,0,2} - \frac{1}{2}C1_{1,0,2}\right)x^2\right], [C1_{0,0,0}, C1_{1,0,0}, C1_{0,0,1}, C1_{0,0,2}, C1_{1,0,1}, C1_{0,0,3}, C1_{1,0,2}]$$

Example 2:

```
> ivar := [x,t=2]; dvar := [u];
```

$$ivar := [x, t=2]$$

$$dvar := [u]$$

```
> L := [diff(u(x,t),x,x)-diff(u(x,t),t)-t];
```

$$L := \left[\left(\frac{\partial^2}{\partial x^2}\mathrm{u}(x,t)\right) - \left(\frac{\partial}{\partial t}\mathrm{u}(x,t)\right) - t\right]$$

```
> J := JanetBasis(L, ivar, dvar);
```

$$J := \left[\left[\left(\frac{\partial^2}{\partial x^2}\mathrm{u}(x,t)\right) - \left(\frac{\partial}{\partial t}\mathrm{u}(x,t)\right) - t\right], [x,t], [u]\right]$$

```
> PolySol([J[1],ivar,J[3]], [[4,4]], 'b');
```

$$\left[\mathrm{u}(x,t) = C1_{0,0} + \frac{1}{6}C1_{1,1}\,x^3 + C1_{0,1}\,t + C1_{1,0}\,x + \left(\frac{1}{2} + \frac{1}{2}C1_{0,2}\right)x^2 t + \frac{1}{2}C1_{0,2}\,t^2 + C1_{1,1}\,xt + \frac{1}{2}C1_{0,1}\,x^2 + \left(\frac{1}{24}C1_{0,2} + \frac{1}{24}\right)x^4\right], [C1_{0,0}, C1_{1,0}, C1_{0,1}, C1_{0,2}, C1_{1,1}]$$

```
> b;
```

$$\left[\left[\mathrm{u}(x,t)=\frac{1}{2}x^2 t+\frac{1}{24}x^4\right], \left[\mathrm{u}(x,t)=1+\frac{1}{2}x^2 t+\frac{1}{24}x^4\right], \left[\mathrm{u}(x,t)=x+\frac{1}{2}x^2 t+\frac{1}{24}x^4\right], \left[\mathrm{u}(x,t)=t+\frac{1}{2}x^2 t+\frac{1}{2}x^2+\frac{1}{24}x^4\right], \left[\mathrm{u}(x,t)=x^2 t+\frac{1}{2}t^2+\frac{1}{12}x^4\right], \left[\mathrm{u}(x,t)=\frac{1}{6}x^3+\frac{1}{2}x^2 t+xt+\frac{1}{24}x^4\right]\right]$$

```
> PolySol([J[1],ivar,J[3]], 4, 'b'):
> b;
```

$$\left[\left[\mathrm{u}(x,t)=\frac{1}{2}x^2 t+\frac{1}{24}x^4\right], \left[\mathrm{u}(x,t)=1+\frac{1}{2}x^2 t+\frac{1}{24}x^4\right], \left[\mathrm{u}(x,t)=x+\frac{1}{2}x^2 t+\frac{1}{24}x^4\right], \left[\mathrm{u}(x,t)=t+\frac{1}{2}x^2 t+\frac{1}{2}x^2+\frac{1}{24}x^4\right],$$

$$\left[\mathrm{u}(x,t)=x^2 t+\frac{1}{2}t^2+\frac{1}{12}x^4\right],\left[\mathrm{u}(x,t)=\frac{1}{6}x^3+\frac{1}{2}x^2 t+xt+\frac{1}{24}x^4\right]$$

Note, the equations in this example were inhomogeneous. In case one has polynomial solutions, the program lists one polynomial solution more than the number of free parameters, the first solution having all parameters equal to zero.

Example 3:

```
> L := [diff(u(x,t),x,x)-diff(u(x,t),t)-sin(t)];
```

$$L:=\left[\left(\frac{\partial^2}{\partial x^2}\mathrm{u}(x,t)\right)-\left(\frac{\partial}{\partial t}\mathrm{u}(x,t)\right)-\sin(t)\right]$$

```
> J := JanetBasis(L, ivar, dvar);
```

$$J:=\left[\left[\left(\frac{\partial^2}{\partial x^2}\mathrm{u}(x,t)\right)-\left(\frac{\partial}{\partial t}\mathrm{u}(x,t)\right)-\sin(t)\right],[x,t],[u]\right]$$

```
> PolySol([J[1],ivar,J[3]], 4);
```

There is no polynomial solution up to degree 4.

Example 4: (change the name for constants)

```
> J := JanetBasis([diff(u(x),x$5)], [x], [u]);
```

$$J:=\left[\left[\frac{d^5}{dx^5}\mathrm{u}(x)\right],[x],[u]\right]$$

```
> PolySol(J, 3, "constname"=c);
```

$$\left[\mathrm{u}(x)=c1_0+c1_1 x+\frac{1}{2}c1_2 x^2+\frac{1}{6}c1_3 x^3\right],[c1_0,c1_1,c1_2,c1_3]$$

See Also:

JanetBasis, PrincDeriv, ParamDeriv, InvReduce, CompCond, CompCondBasis, HilbertSeries, HilbertPolynomial, HilbertFunction, SolSeries, ZeroSets.

Janet[Resolution] - return free resolution of a left module over a ring of differential operators

Calling Sequence:

Resolution(L,ivar,dvar,mode,tr)

Parameters:

`L` - list of linear differential expressions (to be interpreted as module generators)
`ivar` - list of independent variables
`dvar` - list of dependent variables
`mode` - (optional) string specifying the type of information to be returned
`tr` - (optional) positive integer (truncate resolution to length **tr**)

Description:

- We first take the point of view of left modules over the ring *R* which is the non-commutative polynomial ring in the partial derivatives with respect to **ivar** over a field of (analytic) functions in the independent variables **ivar**. The list **L** is interpreted as a subset of the left *R*-module *R^m* of *m*-tuples, where *m* is the number of dependent variables in **dvar**. As such, **L** generates an *R*-submodule of *R^m*. ***Resolution*** computes a free resolution of *R^m*/<**L**> by first computing the minimal Janet basis for <**L**> say of *k*(1) elements. These elements are given in form of a matrix describing a homomorphism *R^k*(1) -> *R^m* with cokernel *R^m*/<**L**>. It then computes a generating set **L**(1) of the kernel of this homomorphism and proceeds with *R^k*(1) and **L**(1) in the same way as it did with *R^m* and **L**. It terminates once the kernel is trivial.
- In the default case the output of ***Resolution*** is a list of differential operators given in matrix form. These matrices are to be multiplied from the right to the rows in *R^k*(*i*).
- The entries of these matrices have the form [[c_1, [...]], ..., [c_r, [...]]], where the dots in the innermost brackets are sequences of independent variables from **ivar** representing partial derivatives with respect to these variables, and the c_i are the coefficients for these, i.e. functions in **ivar**. The whole bracket represents the sum of these items, i.e. the linear combination of the above operators. Note, if an innermost bracket is empty, it represents the identity operator. To pass from a differential operator as described here to a differential expression, cf. AppOp, for the opposite direction cf. Diff2Op.
- Note, for matrices the row convention is used, i. e. *R^m* is identified with *R^*{1 x *m*} and the matrices are multiplied to rows from the right.
- As optional fourth parameter a string consisting of letters "C", "D", "G", "M", and "O" is accepted that does not contain "D" and "M" at the same time.
- If **mode** contains the letter "M", then the output is the list of matrices that were computed as kernels of the above homomorphisms (in the reversed order they were constructed). This is the default mode. If **mode** contains the letter "D" then the output is a list containing lists of integers [[$d_{r,1}$, ..., d_{r,n_r}], ..., [$d_{1,1}$, ..., d_{1,n_1}]], where $d_{i,j}$ is the degree of the *j*-th generator (row in matrix) of the *i*-th free module in the free resolution.
- If the letter "O" is present in **mode**, minimal Groebner bases are computed instead of minimal Janet bases in each step.
- If **mode** contains the letter "G", then the first matrix (i.e. the last one in the output) is formed using the given generating set **L**, i.e. the computation of a Janet basis is suppressed in the first step. If also the letter "C" is present in **mode**, then the minimal Janet basis is still computed in the first step and the first matrix is formed using the smaller generating set of **L** and its minimal Janet basis.
- If the optional parameter **tr** is supplied, ***Resolution*** stops after having computed **tr** kernels as described above.
- Taking the PDE point of view, the command ***Resolution*** creates in the first step a linear PDE system describing those right hand sides of the PDE system represented by the Janet basis of **L**, for which the corresponding affine system is solvable. In other words, the first step of ***Resolution*** is nothing else but the application of CompCond to the Janet basis of **L**. This process is iterated, e.g. in the second step the procedure is applied to the output of the first step, i.e. one obtains compatibility conditions for the compatibility conditions.
- For more information about Janet bases and resolutions, see W. Plesken, D. Robertz, "Janet's approach to presentations and

resolutions for polynomials and linear pdes", Archiv der Mathematik, 84(1), 2005, 22-37.

Examples:

```
> with(Janet):
```

Example 1: (Poincare sequence up to signs)

```
> ivar := [x,y,z]; dvar := [u];
```

$$ivar := [x, y, z]$$

$$dvar := [u]$$

```
> L := [diff(u(x,y,z), x), diff(u(x,y,z), y), diff(u(x,y,z), z)];
```

$$L := \left[\frac{\partial}{\partial x}\mathrm{u}(x,y,z), \frac{\partial}{\partial y}\mathrm{u}(x,y,z), \frac{\partial}{\partial z}\mathrm{u}(x,y,z)\right]$$

```
> Resolution(L, ivar, dvar);
```

$$\left[\begin{bmatrix}[[1,[x]]] & [[1,[z]]] & [[-1,[y]]]\end{bmatrix}, \begin{bmatrix}[[1,[y]]] & [[-1,[z]]] & 0\\ 0 & [[1,[x]]] & [[-1,[y]]]\\ [[1,[x]]] & 0 & [[-1,[z]]]\end{bmatrix}, \begin{bmatrix}[[1,[z]]]\\ [[1,[y]]]\\ [[1,[x]]]\end{bmatrix}\right]$$

```
> Resolution(L, ivar, dvar, "D");
```

$$[[3], [2, 2, 2], [1, 1, 1]]$$

Example 2: This example compares PolResolution with ***Resolution*** for an example with constant coefficients.

```
> with(Involutive):
```

We start with the polynomial case.

```
> L2 := [x^2,y^2,z^2];
```

$$L2 := [x^2, y^2, z^2]$$

```
> PolResolution(L2, [x,y,z]);
```

$$\left[\begin{bmatrix}0 & -1 & 0 & x & 0 & 0 & 0 & 1 & 0 & -y\\ -1 & 0 & x & 0 & 0 & 0 & 0 & -y & z^2 & 0\\ 0 & x & 0 & 0 & 1 & 0 & -y & 0 & 0 & 0\\ x & 0 & 0 & 0 & -y & z^2 & 0 & 0 & 0 & 0\end{bmatrix}, \begin{bmatrix}0 & 0 & 0 & 0 & 0 & -z^2 & y\\ 0 & 0 & 0 & 0 & y & 0 & -1\\ 0 & -z^2 & 0 & y & 0 & 0 & 0\\ y & 0 & 0 & -1 & 0 & 0 & 0\\ 0 & 0 & -z^2 y & 0 & 0 & 0 & x\\ 0 & 0 & -y^2 & 0 & 0 & x & 0\\ 0 & 0 & -z^2 & 0 & x & 0 & 0\\ 0 & 0 & 0 & x & 0 & 0 & -1\\ 0 & x & 0 & 0 & 0 & -1 & 0\\ x & 0 & 0 & 0 & -1 & 0 & 0\end{bmatrix}, \begin{bmatrix}z^2\\ y^2\\ x^2\\ z^2 y\\ z^2 x\\ y^2 x\\ z^2 yx\end{bmatrix}\right]$$

We now translate to the PDE case.

```
> LL2 := Pol2Diff(L2, ivar, dvar);
```

$$LL2 := \left[\frac{\partial^2}{\partial x^2}\mathrm{u}(x,y,z), \frac{\partial^2}{\partial y^2}\mathrm{u}(x,y,z), \frac{\partial^2}{\partial z^2}\mathrm{u}(x,y,z)\right]$$

```
> Resolution(LL2, ivar, dvar);
```

$$\left[\begin{bmatrix} 0 & [[-1,[\ ]]] & 0 & [[1,[x]]] & 0 & 0 & 0 & [[1,[\ ]]] & 0 & [[-1,[y]]] \\ [[-1,[\ ]]] & 0 & [[1,[x]]] & 0 & 0 & 0 & 0 & [[-1,[y]]] & [[1,[z,z]]] & 0 \\ 0 & [[1,[x]]] & 0 & 0 & [[1,[\ ]]] & 0 & [[-1,[y]]] & 0 & 0 & 0 \\ [[1,[x]]] & 0 & 0 & 0 & [[-1,[y]]] & [[1,[z,z]]] & 0 & 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & [[-1,[z,z]]] & [[1,[y]]] \\ 0 & 0 & 0 & 0 & [[1,[y]]] & 0 & [[-1,[\ ]]] \\ 0 & [[-1,[z,z]]] & 0 & [[1,[y]]] & 0 & 0 & 0 \\ [[1,[y]]] & 0 & 0 & [[-1,[\ ]]] & 0 & 0 & 0 \\ 0 & 0 & [[-1,[y,z,z]]] & 0 & 0 & 0 & [[1,[x]]] \\ 0 & 0 & [[-1,[y,y]]] & 0 & 0 & [[1,[x]]] & 0 \\ 0 & 0 & [[-1,[z,z]]] & 0 & [[1,[x]]] & 0 & 0 \\ 0 & 0 & 0 & [[1,[x]]] & 0 & 0 & [[-1,[\ ]]] \\ 0 & [[1,[x]]] & 0 & 0 & 0 & [[-1,[\ ]]] & 0 \\ [[1,[x]]] & 0 & 0 & 0 & [[-1,[\ ]]] & 0 & 0 \end{bmatrix}, \begin{bmatrix} [[1,[z,z]]] \\ [[1,[y,y]]] \\ [[1,[x,x]]] \\ [[1,[y,z,z]]] \\ [[1,[x,z,z]]] \\ [[1,[x,y,y]]] \\ [[1,[x,y,z,z]]] \end{bmatrix}\right]$$

Example 3: (Demonstrating the PDE point of view)

```
> ivar := [x,y,z]; dvar := [u];
```

$$ivar := [x, y, z]$$

$$dvar := [u]$$

```
> L := [exp(y)*diff(u(x,y,z),x)-y^2*diff(u(x,y,z),y,z),diff(u(x,y,z),x,z)];
```

$$L := \left[\mathbf{e}^{y}\left(\frac{\partial}{\partial x}\mathrm{u}(x,y,z)\right) - y^{2}\left(\frac{\partial^{2}}{\partial z\,\partial y}\mathrm{u}(x,y,z)\right), \frac{\partial^{2}}{\partial z\,\partial x}\mathrm{u}(x,y,z)\right]$$

```
> r := Resolution(L, ivar, dvar);
```

$$r := \left[\begin{bmatrix} 0 & [[1,[x]]] & [[-1,[z]]] \\ [[1,[x]]] & [[-y^{2},[y]]] & [[\mathbf{e}^{y},[\ ]]] \end{bmatrix}, \begin{bmatrix} [[y^{2},[y,z]],[-\mathbf{e}^{y},[x]]] \\ [[1,[x,z]]] \\ [[1,[x,x]]] \end{bmatrix}\right]$$

Compare the rightmost matrix to the Janet basis of **L**:

```
> AppOp(r[2], [u(x,y,z)], ivar, dvar);
```

$$\left[-\mathbf{e}^{y}\left(\frac{\partial}{\partial x}\mathrm{u}(x,y,z)\right) + y^{2}\left(\frac{\partial^{2}}{\partial z\,\partial y}\mathrm{u}(x,y,z)\right), \frac{\partial^{2}}{\partial z\,\partial x}\mathrm{u}(x,y,z), \frac{\partial^{2}}{\partial x^{2}}\mathrm{u}(x,y,z)\right]$$

```
> JanetBasis(L, ivar, dvar);
```

$$\left[\left[-\mathbf{e}^{y}\left(\frac{\partial}{\partial x}\,u(x,y,z)\right)+y^{2}\left(\frac{\partial^{2}}{\partial z\,\partial y}\,u(x,y,z)\right),\ \frac{\partial^{2}}{\partial z\,\partial x}\,u(x,y,z),\ \frac{\partial^{2}}{\partial x^{2}}\,u(x,y,z)\right],\ [x,y,z],\ [u]\right]$$

The compatibility conditions for the right hand side of the Janet basis, say $a(x,y,z)$, $b(x,y,z)$, $c(x,y,z)$ comes as follows from the second matrix from the right in the resolution:

```
> AppOp(r[1], [a,b,c], ivar, [a,b,c]);
```

$$\left[\left(\frac{\partial}{\partial x}\,b(x,y,z)\right)-\left(\frac{\partial}{\partial z}\,c(x,y,z)\right),\ \left(\frac{\partial}{\partial x}\,a(x,y,z)\right)-y^{2}\left(\frac{\partial}{\partial y}\,b(x,y,z)\right)+\mathbf{e}^{y}\,c(x,y,z)\right]$$

Since the resolution terminates here, this system allows arbitrary right hand sides.

See Also:

JanetBasis, PrincDeriv, ParamDeriv, InvReduce, CompCond, CompCondBasis, ResolutionDim, EulerChar, SyzOp, Diff2Op, AppOp, Pol2Diff, Diff2Pol, PolResolution.

Janet[RightInverse] - compute a right inverse of a linear differential operator

Calling Sequence:

RightInverse(M,ivar,dvar,rvar)

Parameters:

`M` - differential operator in matrix form or list of linear differential expressions
`ivar` - list of independent variables
`dvar` - (optional) list of dependent variables
`rvar` - (optional) list of dependent variables for the right inverse

Description:

- ***RightInverse*** computes (if possible) a right inverse of a linear differential operator **`M`**, i.e. a linear differential operator *R* such that the composition of **`M`** and *R* is the identity.
- The first parameter **`M`** is expected to be a linear differential operator in matrix form (cf. AppOp) with independent variables **`ivar`** or a list of linear differential expressions with independent variables **`ivar`** and dependent variables **`dvar`**. In the second case, ***RightInverse*** extracts the linear differential operator from **`M`** and computes a right inverse of this operator.
- If **`M`** is given as differential operator in matrix form, then the arguments **`dvar`** and **`rvar`** can be omitted.
- If no right inverse of **`M`** exists, ***RightInverse*** returns FAIL.
- If a right inverse *R* of **`M`** exists, ***RightInverse*** returns such a differential operator *R* if **`M`** is a differential operator in matrix form, or returns the list of linear differential expressions which is obtained by applying *R* to the list **`rvar`** (or to the list of serially numbered variables _A if **`rvar`** is not given) if **`M`** is a list of linear differential expressions.
- Left inverses of linear differential operators are computed by LeftInverse.

Examples:

```
> with(Janet):
> P := [[2*x^2, x^2, -1, 2*x^2], [4*x^2-2, 0, 0, 2*x^2], [0, x^2-1, 0, x^2]];
```

$$P := [[2x^2, x^2, -1, 2x^2], [4x^2-2, 0, 0, 2x^2], [0, x^2-1, 0, x^2]]$$

```
> ivar := [x];
```

$$ivar := [x]$$

```
> M := Pol2Op(P, ivar);
```

$$M := \begin{bmatrix} [[2,[x,x]]] & [[1,[x,x]]] & [[-1,[\,]]] & [[2,[x,x]]] \\ [[4,[x,x]],[-2,[\,]]] & 0 & 0 & [[2,[x,x]]] \\ 0 & [[1,[x,x]],[-1,[\,]]] & 0 & [[1,[x,x]]] \end{bmatrix}$$

```
> R := RightInverse(M, ivar);
```

$$R := \begin{bmatrix} 0 & \left[[1,[x,x]],\left[\frac{-1}{2},[\,]\right]\right] & [[-1,[x,x]]] \\ 0 & [[2,[x,x]]] & [[-2,[x,x]],[-1,[\,]]] \\ [[-1,[\,]]] & [[3,[x,x]]] & [[-3,[x,x]]] \\ 0 & [[-2,[x,x]],[2,[\,]]] & [[2,[x,x]],[-1,[\,]]] \end{bmatrix}$$

```
> CmpOp(M, R, ivar);
```

$$\begin{bmatrix} [[1,[\,]]] & 0 & 0 \\ 0 & [[1,[\,]]] & 0 \\ 0 & 0 & [[1,[\,]]] \end{bmatrix}$$

```
> LeftInverse(M, ivar);
```

$$FAIL$$

```
> dvar := [u1,u2,u3,u4];
```

$$dvar := [u1, u2, u3, u4]$$

```
> M := Pol2Diff(P, ivar, dvar);
```

$$M := \left[2\left(\frac{d^2}{dx^2}\,\mathrm{u1}(x)\right)+\left(\frac{d^2}{dx^2}\,\mathrm{u2}(x)\right)-\mathrm{u3}(x)+2\left(\frac{d^2}{dx^2}\,\mathrm{u4}(x)\right),\ 4\left(\frac{d^2}{dx^2}\,\mathrm{u1}(x)\right)-2\,\mathrm{u1}(x)+2\left(\frac{d^2}{dx^2}\,\mathrm{u4}(x)\right),\left(\frac{d^2}{dx^2}\,\mathrm{u2}(x)\right)-\mathrm{u2}(x)+\left(\frac{d^2}{dx^2}\,\mathrm{u4}(x)\right)\right]$$

```
> RightInverse(M, ivar, dvar);
```

$$\left[\left(\frac{d^2}{dx^2}\,\_\mathrm{A2}(x)\right)-\frac{1}{2}\,\_\mathrm{A2}(x)-\left(\frac{d^2}{dx^2}\,\_\mathrm{A3}(x)\right),\ 2\left(\frac{d^2}{dx^2}\,\_\mathrm{A2}(x)\right)-2\left(\frac{d^2}{dx^2}\,\_\mathrm{A3}(x)\right)-\_\mathrm{A3}(x),\ -\_\mathrm{A1}(x)+3\left(\frac{d^2}{dx^2}\,\_\mathrm{A2}(x)\right)-3\left(\frac{d^2}{dx^2}\,\_\mathrm{A3}(x)\right),\right.$$
$$\left.-2\left(\frac{d^2}{dx^2}\,\_\mathrm{A2}(x)\right)+2\,\_\mathrm{A2}(x)+2\left(\frac{d^2}{dx^2}\,\_\mathrm{A3}(x)\right)-\_\mathrm{A3}(x)\right]$$

```
> RightInverse(M, ivar, dvar, [v1,v2,v3]);
```

$$\left[\left(\frac{d^2}{dx^2}\,\mathrm{v2}(x)\right)-\frac{1}{2}\,\mathrm{v2}(x)-\left(\frac{d^2}{dx^2}\,\mathrm{v3}(x)\right),\ 2\left(\frac{d^2}{dx^2}\,\mathrm{v2}(x)\right)-2\left(\frac{d^2}{dx^2}\,\mathrm{v3}(x)\right)-\mathrm{v3}(x),\ -\mathrm{v1}(x)+3\left(\frac{d^2}{dx^2}\,\mathrm{v2}(x)\right)-3\left(\frac{d^2}{dx^2}\,\mathrm{v3}(x)\right),\right.$$
$$\left.-2\left(\frac{d^2}{dx^2}\,\mathrm{v2}(x)\right)+2\,\mathrm{v2}(x)+2\left(\frac{d^2}{dx^2}\,\mathrm{v3}(x)\right)-\mathrm{v3}(x)\right]$$

See Also:

JanetBasis, PrincDeriv, InvReduce, LeftInverse, PDEFactorize, AppOp, CmpOp, AddOp, SubOp, Diff2Op, Pol2Op, Pol2Diff, AffEqn.

Janet[ShorterResolution] - shorten (if possible) a free resolution of a left module over a ring of differential operators

Calling Sequence:

ShorterResolution(F,ivar,dvar)

Parameters:

`F`	-	list of matrices representing linear differential operators
`ivar`	-	list of independent variables
`dvar`	-	(optional) list of dependent variables

Description:

- Let R be the non-commutative polynomial ring in the partial derivatives with respect to **`ivar`** over a field of (analytic) functions in the independent variables **`ivar`**.
- Given a (finite) free resolution of a finitely presented left module over R, ***ShorterResolution*** tries to construct a shorter free resolution of the same module. This is possible whenever the last homomorphism between free modules in this free resolution admits a right inverse (see RightInverse).
- If the length m of the free resolution given by **F** is at least 3 and if the last homomorphism R_m between free modules given in **F** admits a right inverse S_m, then a shorter free resolution is obtained by removing the last free module, augmenting the last but first homomorphism R_{m-1} with S_m, i.e. replacing it by $(R_{m-1}\ S_m)$, and replacing the last but second homomorphism R_{m-2} by the transpose of $(R_{m-2}\ 0)$ in a compatible way (note also that the last but second free module in the given free resolution must be adjusted).
- If the length m of the free resolution given by **F** equals 2 and if the last homomorphism R_2 between free modules given in **F** admits a right inverse S_2, then a presentation of the module resolved by **F** is obtained by removing the last free module and augmenting the last but first homomorphism R_1 with S_2, i.e. by defining the presentation matrix $(R_1\ S_2)$.
- If the length m of the free resolution given by **F** is less than 2, then ***ShorterResolution*** returns **F**.
- **F** is a list of matrices representing a free resolution of a finitely presented left module over R. Most commonly, **F** is the result of Resolution.
- The result of ***ShorterResolution*** is of the same format as the input **F**, i.e. a list representing a free resolution of the finitely presented left module, which is shorter than the given one or equals the given one.
- The procedure described above can be iterated using the command ShortestResolution.
- For more details, see A. Quadrat, D. Robertz, "Computation of bases of free modules over the Weyl algebras", Journal of Symbolic Computation 42 (11-12), 2007, pp. 1113-1141.

Examples:

```
> with(Janet):
```

Example:

(see J.-F. Pommaret, *Partial Differential Equations and Group Theory: New Perspectives for Applications Kluwer*, 1994, p. 162)

```
> ivar := [x,y,z]; dvar := [u];
```

$$ivar := [x, y, z]$$

$$dvar := [u]$$

```
> L := [u(x,y,z), diff(u(x,y,z),x),  diff(u(x,y,z),y), diff(u(x,y,z),z)];
```

$$L := \left[u(x,y,z), \frac{\partial}{\partial x}u(x,y,z), \frac{\partial}{\partial y}u(x,y,z), \frac{\partial}{\partial z}u(x,y,z)\right]$$

```
> F1 := Resolution(L, ivar, dvar, "G");
```

$$F1 := \left[\begin{bmatrix} [[1,[x]]] & [[-1,[\ ]]] & [[1,[z]]] & [[-1,[y]]] \end{bmatrix}, \begin{bmatrix} [[-1,[\ ]]] & 0 & 0 & [[1,[y]]] & [[-1,[z]]] & 0 \\ [[-1,[x]]] & [[1,[y]]] & [[-1,[z]]] & 0 & 0 & 0 \\ 0 & 0 & [[-1,[\ ]]] & 0 & [[1,[x]]] & [[-1,[y]]] \\ 0 & [[-1,[\ ]]] & 0 & [[1,[x]]] & 0 & [[-1,[z]]] \end{bmatrix}, \begin{bmatrix} 0 & 0 & [[1,[z]]] & [[-1,[y]]] \\ 0 & [[1,[z]]] & 0 & [[-1,[x]]] \\ 0 & [[1,[y]]] & [[-1,[x]]] & 0 \\ [[1,[z]]] & 0 & 0 & [[-1,[\ ]]] \\ [[1,[y]]] & 0 & [[-1,[\ ]]] & 0 \\ [[1,[x]]] & [[-1,[\ ]]] & 0 & 0 \end{bmatrix}, \begin{bmatrix} [[1,[\ ]]] \\ [[1,[x]]] \\ [[1,[y]]] \\ [[1,[z]]] \end{bmatrix} \right]$$

```
> F2 := ShorterResolution(F1, ivar);
```

$$F2 := \left[\begin{bmatrix} [[-1,[\ ]]] & 0 & 0 & [[1,[y]]] & [[-1,[z]]] & 0 & 0 \\ [[-1,[x]]] & [[1,[y]]] & [[-1,[z]]] & 0 & 0 & 0 & [[-1,[\ ]]] \\ 0 & 0 & [[-1,[\ ]]] & 0 & [[1,[x]]] & [[-1,[y]]] & 0 \\ 0 & [[-1,[\ ]]] & 0 & [[1,[x]]] & 0 & [[-1,[z]]] & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & [[1,[z]]] & [[-1,[y]]] \\ 0 & [[1,[z]]] & 0 & [[-1,[x]]] \\ 0 & [[1,[y]]] & [[-1,[x]]] & 0 \\ [[1,[z]]] & 0 & 0 & [[-1,[\ ]]] \\ [[1,[y]]] & 0 & [[-1,[\ ]]] & 0 \\ [[1,[x]]] & [[-1,[\ ]]] & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} [[1,[\ ]]] \\ [[1,[x]]] \\ [[1,[y]]] \\ [[1,[z]]] \end{bmatrix} \right]$$

```
> F3 := ShorterResolution(F2, ivar);
```

$$F3 := \left[\begin{bmatrix} 0 & 0 & [[1,[z]]] & [[-1,[y]]] & [[-1,[\ ]]] & 0 & 0 & 0 \\ 0 & [[1,[z]]] & 0 & [[-1,[x]]] & 0 & 0 & 0 & [[-1,[\ ]]] \\ 0 & [[1,[y]]] & [[-1,[x]]] & 0 & 0 & 0 & [[-1,[\ ]]] & 0 \\ [[1,[z]]] & 0 & 0 & [[-1,[\ ]]] & 0 & 0 & 0 & 0 \\ [[1,[y]]] & 0 & [[-1,[\ ]]] & 0 & 0 & 0 & 0 & 0 \\ [[1,[x]]] & [[-1,[\ ]]] & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & [[1,[x]]] & [[-1,[\ ]]] & [[1,[z]]] & [[-1,[y]]] \end{bmatrix}, \begin{bmatrix} [[1,[\ ]]] \\ [[1,[x]]] \\ [[1,[y]]] \\ [[1,[z]]] \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}\right]$$

```
> F4 := ShorterResolution(F3, ivar);
```

$$F4 := \left[\begin{bmatrix} [[1,[\ ]]] & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ [[1,[x]]] & 0 & 0 & 0 & 0 & 0 & [[-1,[\ ]]] & 0 \\ [[1,[y]]] & 0 & 0 & 0 & 0 & [[-1,[\ ]]] & 0 & 0 \\ [[1,[z]]] & 0 & 0 & 0 & [[-1,[\ ]]] & 0 & 0 & 0 \\ 0 & [[-1,[\ ]]] & 0 & 0 & [[1,[y]]] & [[-1,[z]]] & 0 & 0 \\ 0 & [[-1,[x]]] & [[1,[y]]] & [[-1,[z]]] & 0 & 0 & 0 & [[-1,[\ ]]] \\ 0 & 0 & 0 & [[-1,[\ ]]] & 0 & [[1,[x]]] & [[-1,[y]]] & 0 \\ 0 & 0 & [[-1,[\ ]]] & 0 & [[1,[x]]] & 0 & [[-1,[z]]] & 0 \end{bmatrix}\right]$$

```
> ShorterResolution(F4, ivar);
```

$$\left[\begin{bmatrix} [[1,[\ ]]] & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ [[1,[x]]] & 0 & 0 & 0 & 0 & 0 & [[-1,[\ ]]] & 0 \\ [[1,[y]]] & 0 & 0 & 0 & 0 & [[-1,[\ ]]] & 0 & 0 \\ [[1,[z]]] & 0 & 0 & 0 & [[-1,[\ ]]] & 0 & 0 & 0 \\ 0 & [[-1,[\ ]]] & 0 & 0 & [[1,[y]]] & [[-1,[z]]] & 0 & 0 \\ 0 & [[-1,[x]]] & [[1,[y]]] & [[-1,[z]]] & 0 & 0 & 0 & [[-1,[\ ]]] \\ 0 & 0 & 0 & [[-1,[\ ]]] & 0 & [[1,[x]]] & [[-1,[y]]] & 0 \\ 0 & 0 & [[-1,[\ ]]] & 0 & [[1,[x]]] & 0 & [[-1,[z]]] & 0 \end{bmatrix}\right]$$

Hence, it was possible to reduce the length of the free resolution represented by **F1** in each step, finally arriving at a free resolution of length 1. These steps can be done at once by calling ShortestResolution:

```
> F := ShortestResolution(F1, ivar);
```

$$F := \left[\begin{bmatrix} [[1,[\ ]]] & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ [[1,[x]]] & 0 & 0 & 0 & 0 & 0 & [[-1,[\ ]]] & 0 \\ [[1,[y]]] & 0 & 0 & 0 & 0 & [[-1,[\ ]]] & 0 & 0 \\ [[1,[z]]] & 0 & 0 & 0 & [[-1,[\ ]]] & 0 & 0 & 0 \\ 0 & [[-1,[\ ]]] & 0 & 0 & [[1,[y]]] & [[-1,[z]]] & 0 & 0 \\ 0 & [[-1,[x]]] & [[1,[y]]] & [[-1,[z]]] & 0 & 0 & 0 & [[-1,[\ ]]] \\ 0 & 0 & 0 & [[-1,[\ ]]] & 0 & [[1,[x]]] & [[-1,[y]]] & 0 \\ 0 & 0 & [[-1,[\ ]]] & 0 & [[1,[x]]] & 0 & [[-1,[z]]] & 0 \end{bmatrix}\right]$$

In fact, the module presented by L is stably free because a right inverse of the presentation matrix obtained by ShortestResolution admits a right inverse:

```
> RightInverse(F[1], ivar);
```

$$\begin{bmatrix} [[1,[\ ]]] & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & [[1,[z]]] & [[-1,[y]]] & [[-1,[\ ]]] & 0 & 0 & 0 \\ 0 & [[1,[z]]] & 0 & [[-1,[x]]] & 0 & 0 & 0 & [[-1,[\ ]]] \\ 0 & [[1,[y]]] & [[-1,[x]]] & 0 & 0 & 0 & [[-1,[\ ]]] & 0 \\ [[1,[z]]] & 0 & 0 & [[-1,[\ ]]] & 0 & 0 & 0 & 0 \\ [[1,[y]]] & 0 & [[-1,[\ ]]] & 0 & 0 & 0 & 0 & 0 \\ [[1,[x]]] & [[-1,[\ ]]] & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & [[1,[x]]] & [[-1,[\ ]]] & [[1,[z]]] & [[-1,[y]]] \end{bmatrix}$$

See Also:

JanetBasis, PrincDeriv, InvReduce, CompCond, CompCondBasis, Resolution, ShortestResolution, ResolutionDim, EulerChar.

Janet[ShortestResolution] - return a shortest free resolution of a finitely presented module over a ring of differential operators

Calling Sequence:

ShortestResolution(F,ivar,dvar)

Parameters:

`F`	-	list of matrices representing linear differential operators
`ivar`	-	list of independent variables
`dvar`	-	(optional) list of dependent variables

Description:

- Let *R* be the non-commutative polynomial ring in the partial derivatives with respect to **`ivar`** over a field of (analytic) functions in the independent variables **`ivar`**.
- ***ShortestFreeResolution*** iterates the application of ShorterResolution to a (finite) free resolution of a finitely presented left module over *R* and returns a free resolution of the same module which cannot be shortened in this way anymore.
- **F** is either a matrix with polynomial entries or a list of matrices representing a free resolution of a finitely presented left module over *R* . In the first case, a free resolution of the left module presented by **F** is computed first. In the second case, most commonly, **F** is the result of Resolution. Then, in both cases, ShorterResolution is applied repeatedly to the resolution until ShorterResolution does not change the resolution anymore.
- The result of ***ShortestFreeResolution*** is a list representing a free resolution of the finitely presented left module.
- For more details, see A. Quadrat, D. Robertz, "Computation of bases of free modules over the Weyl algebras", Journal of Symbolic Computation 42 (11-12), 2007, pp. 1113-1141.

Examples:

```
> with(Janet):
```

Example:

(see J.-F. Pommaret, *Partial Differential Equations and Group Theory: New Perspectives for Applications Kluwer*, 1994, p. 162)

```
> ivar := [x,y,z]; dvar := [u];
```

$$ivar := [x, y, z]$$

$$dvar := [u]$$

```
> L := [u(x,y,z), diff(u(x,y,z),x),  diff(u(x,y,z),y), diff(u(x,y,z),z)];
```

$$L := \left[u(x,y,z), \frac{\partial}{\partial x}u(x,y,z), \frac{\partial}{\partial y}u(x,y,z), \frac{\partial}{\partial z}u(x,y,z)\right]$$

```
> F1 := Resolution(L, ivar, dvar, "G");
```

$$F1 := \left[\begin{bmatrix} [[1,[x]]] & [[-1,[\,]]] & [[1,[z]]] & [[-1,[y]]] \end{bmatrix}, \begin{bmatrix} [[-1,[\,]]] & 0 & 0 & [[1,[y]]] & [[-1,[z]]] & 0 \\ [[-1,[x]]] & [[1,[y]]] & [[-1,[z]]] & 0 & 0 & 0 \\ 0 & 0 & [[-1,[\,]]] & 0 & [[1,[x]]] & [[-1,[y]]] \\ 0 & [[-1,[\,]]] & 0 & [[1,[x]]] & 0 & [[-1,[z]]] \end{bmatrix}\right],$$

$$\begin{bmatrix} 0 & 0 & [[1,[z]]] & [[-1,[y]]] \\ 0 & [[1,[z]]] & 0 & [[-1,[x]]] \\ 0 & [[1,[y]]] & [[-1,[x]]] & 0 \\ [[1,[z]]] & 0 & 0 & [[-1,[\ ]]] \\ [[1,[y]]] & 0 & [[-1,[\ ]]] & 0 \\ [[1,[x]]] & [[-1,[\ ]]] & 0 & 0 \end{bmatrix}, \begin{bmatrix} [[1,[\ ]]] \\ [[1,[x]]] \\ [[1,[y]]] \\ [[1,[z]]] \end{bmatrix}$$

```
> F2 := ShorterResolution(F1, ivar);
```

$$F2 := \begin{bmatrix} \begin{bmatrix} [[-1,[\ ]]] & 0 & 0 & [[1,[y]]] & [[-1,[z]]] & 0 & 0 \\ [[-1,[x]]] & [[1,[y]]] & [[-1,[z]]] & 0 & 0 & 0 & [[-1,[\ ]]] \\ 0 & 0 & [[-1,[\ ]]] & 0 & [[1,[x]]] & [[-1,[y]]] & 0 \\ 0 & [[-1,[\ ]]] & 0 & [[1,[x]]] & 0 & [[-1,[z]]] & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & [[1,[z]]] & [[-1,[y]]] \\ 0 & [[1,[z]]] & 0 & [[-1,[x]]] \\ 0 & [[1,[y]]] & [[-1,[x]]] & 0 \\ [[1,[z]]] & 0 & 0 & [[-1,[\ ]]] \\ [[1,[y]]] & 0 & [[-1,[\ ]]] & 0 \\ [[1,[x]]] & [[-1,[\ ]]] & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \\ \begin{bmatrix} [[1,[\ ]]] \\ [[1,[x]]] \\ [[1,[y]]] \\ [[1,[z]]] \end{bmatrix} \end{bmatrix}$$

```
> F3 := ShorterResolution(F2, ivar);
```

$$F3 := \begin{bmatrix} \begin{bmatrix} 0 & 0 & [[1,[z]]] & [[-1,[y]]] & [[-1,[\ ]]] & 0 & 0 & 0 \\ 0 & [[1,[z]]] & 0 & [[-1,[x]]] & 0 & 0 & 0 & [[-1,[\ ]]] \\ 0 & [[1,[y]]] & [[-1,[x]]] & 0 & 0 & 0 & [[-1,[\ ]]] & 0 \\ [[1,[z]]] & 0 & 0 & [[-1,[\ ]]] & 0 & 0 & 0 & 0 \\ [[1,[y]]] & 0 & [[-1,[\ ]]] & 0 & 0 & 0 & 0 & 0 \\ [[1,[x]]] & [[-1,[\ ]]] & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & [[1,[x]]] & [[-1,[\ ]]] & [[1,[z]]] & [[-1,[y]]] \end{bmatrix}, \begin{bmatrix} [[1,[\ ]]] \\ [[1,[x]]] \\ [[1,[y]]] \\ [[1,[z]]] \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \end{bmatrix}$$

```
> F4 := ShorterResolution(F3, ivar);
```

$$F4 := \begin{bmatrix} \begin{bmatrix} [[1,[\ ]]] & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ [[1,[x]]] & 0 & 0 & 0 & 0 & 0 & [[-1,[\ ]]] & 0 \\ [[1,[y]]] & 0 & 0 & 0 & 0 & [[-1,[\ ]]] & 0 & 0 \\ [[1,[z]]] & 0 & 0 & 0 & [[-1,[\ ]]] & 0 & 0 & 0 \\ 0 & [[-1,[\ ]]] & 0 & 0 & [[1,[y]]] & [[-1,[z]]] & 0 & 0 \\ 0 & [[-1,[x]]] & [[1,[y]]] & [[-1,[z]]] & 0 & 0 & 0 & [[-1,[\ ]]] \\ 0 & 0 & 0 & [[-1,[\ ]]] & 0 & [[1,[x]]] & [[-1,[y]]] & 0 \\ 0 & 0 & [[-1,[\ ]]] & 0 & [[1,[x]]] & 0 & [[-1,[z]]] & 0 \end{bmatrix} \end{bmatrix}$$

```
> ShorterResolution(F4, ivar);
```

$$\begin{bmatrix} \begin{bmatrix} [[1,[\ ]]] & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ [[1,[x]]] & 0 & 0 & 0 & 0 & 0 & [[-1,[\ ]]] & 0 \\ [[1,[y]]] & 0 & 0 & 0 & 0 & [[-1,[\ ]]] & 0 & 0 \\ [[1,[z]]] & 0 & 0 & 0 & [[-1,[\ ]]] & 0 & 0 & 0 \\ 0 & [[-1,[\ ]]] & 0 & 0 & [[1,[y]]] & [[-1,[z]]] & 0 & 0 \\ 0 & [[-1,[x]]] & [[1,[y]]] & [[-1,[z]]] & 0 & 0 & 0 & [[-1,[\ ]]] \\ 0 & 0 & 0 & [[-1,[\ ]]] & 0 & [[1,[x]]] & [[-1,[y]]] & 0 \\ 0 & 0 & [[-1,[\ ]]] & 0 & [[1,[x]]] & 0 & [[-1,[z]]] & 0 \end{bmatrix} \end{bmatrix}$$

Hence, it was possible to reduce the length of the free resolution represented by **F1** in each step, finally arriving at a free resolution of length 1. These steps can be done at once by calling ***ShortestResolution***:

```
> F := ShortestResolution(F1, ivar);
```

$$F := \begin{bmatrix} \begin{bmatrix} [[1,[\ ]]] & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ [[1,[x]]] & 0 & 0 & 0 & 0 & 0 & [[-1,[\ ]]] & 0 \\ [[1,[y]]] & 0 & 0 & 0 & 0 & [[-1,[\ ]]] & 0 & 0 \\ [[1,[z]]] & 0 & 0 & 0 & [[-1,[\ ]]] & 0 & 0 & 0 \\ 0 & [[-1,[\ ]]] & 0 & 0 & [[1,[y]]] & [[-1,[z]]] & 0 & 0 \\ 0 & [[-1,[x]]] & [[1,[y]]] & [[-1,[z]]] & 0 & 0 & 0 & [[-1,[\ ]]] \\ 0 & 0 & 0 & [[-1,[\ ]]] & 0 & [[1,[x]]] & [[-1,[y]]] & 0 \\ 0 & 0 & [[-1,[\ ]]] & 0 & [[1,[x]]] & 0 & [[-1,[z]]] & 0 \end{bmatrix} \end{bmatrix}$$

In fact, the module presented by L is stably free because a right inverse of the presentation matrix obtained by ***ShortestResolution*** admits a right inverse:

```
> RightInverse(F[1], ivar);
```

$$\begin{bmatrix} [[1, [\ ]]] & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & [[1, [z]]] & [[-1, [y]]] & [[-1, [\ ]]] & 0 & 0 & 0 \\ 0 & [[1, [z]]] & 0 & [[-1, [x]]] & 0 & 0 & 0 & [[-1, [\ ]]] \\ 0 & [[1, [y]]] & [[-1, [x]]] & 0 & 0 & 0 & [[-1, [\ ]]] & 0 \\ [[1, [z]]] & 0 & 0 & [[-1, [\ ]]] & 0 & 0 & 0 & 0 \\ [[1, [y]]] & 0 & [[-1, [\ ]]] & 0 & 0 & 0 & 0 & 0 \\ [[1, [x]]] & [[-1, [\ ]]] & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & [[1, [x]]] & [[-1, [\ ]]] & [[1, [z]]] & [[-1, [y]]] \end{bmatrix}$$

See Also:

JanetBasis, PrincDeriv, InvReduce, CompCond, CompCondBasis, Resolution, ShorterResolution, ResolutionDim, EulerChar.

Janet[SolSeries] - Taylor expansion for the general solution of a linear system of partial differential equations

Calling Sequence:

SolSeries(G,deg,p,b,opt)

Parameters:

G	-	list of Janet basis, independent variables, dependent variables (i. e. the output of JanetBasis)
deg	-	natural number bounding the order or list of natural numbers bounding the order for each dependent variable separately
p	-	(optional) list of values for the independent variables fixing the centre of expansion
b	-	(optional) unevaluated name for output in alternative form (as basis)
opt	-	(optional) equation "constname"=beginning of name for constants

Description:

- ***SolSeries*** returns the Taylor expansion of the most general solution of the linear system of partial differential equation given by its Janet basis in **G**. The free Taylor coefficients of this expansion (cf. also ParamDeriv) are named $Cj_{i_1 .. i_n}$ by default (the name *C* is changed to *X*, say, if the equation "constname"=*X* is given in **opt**). Here $Cj_{i_1 .. i_n}$ is the coefficient of $x_1^{i_1} ... x_n^{i_n} / (i_1! ... i_n!)$ in the expansion of u_j, whenever $x_1, ..., x_n$ are chosen to be the independent variables and the u_j are the dependent variables.
- The result of ***SolSeries*** is a sequence of two lists, where the first list contains the Taylor expansions for all dependent variables and the second list contains all free Taylor coefficients which occur in these expansions.
- One can specify the extent of the expansion by specifying **deg** either as a natural number, or as a list of *k* natural numbers, where *k* is the number of dependent variables, or by a *k*-tuple of *n*-tuples, where *n* is the number of independent variables. In the first case the expansion is computed up to order **deg**. In the second case it is computed up to orders op(**deg**), where the *i*-th component of **deg** specifies the order to which the *i*-th component function of the solution is expanded. Finally the third case refines the second case insofar as an upper bound for the order in each independent variable for each dependent variable is specified by **deg**. In this case **deg** is a list of length *k* of lists of length *n*.
- Note, usually the orders are taken in the ordinary sense, but like in the command JanetBasis the user can also specify the degree for each independent variable taken from there. In this case, the output of ***JanetBasis*** cannot be taken as it comes out of the program but the second entry must be taken over from the input of ***JanetBasis***, i. e. it has the form $[x_1=d_1, ..., x_n=d_n]$. Also a fourth parameter in **G** ranging between 1 and 4 might be given to change the order of the derivatives.
- If the center **p** of expansion is not specified, it is taken to be zero. If **p** is specified, say **p**=$[p_1, ..., p_n]$, where *n* is the number of independent variables, then the expansion is taken around $(p_1, ..., p_n)$. The admissible points **p** can be determined from the result of the command ZeroSets: any point outside of the ouput set of ZeroSets is admissible.
- ***SolSeries*** checks first whether it has already been called before for the same system with lower orders for the expansion. If this is the case then it continues its calculation from there. This is to save time for big examples, if one starts with small degrees and repeatedly increases the degrees.
- In case the fourth parameter **b** is given, specific solutions are stored in **b**. They are obtained from the general solutions by setting all constants (parametric derivatives at the point **p**) except for one equal to 0 and the remaining one equal to 1 in all possible ways. In case one has an inhomoneous system of linear pdes, also the solution where all constants are set to zero is given (as first solution).
- If an equation "constname"=*X* is given in **opt**, then the name for the constants appearing in the expansion is changed from *C* to *X*.

Examples:

```
> with(Janet):
```

Example 1:

```
> ivar := [x,y,t]; dvar := [u];
```

$$ivar := [x, y, t]$$

$$dvar := [u]$$

```
> L := [diff(u(x,y,t),x$2) - diff(u(x,y,t),t$2), diff(u(x,y,t),y)-u(x,y,t)];
```

$$L := \left[\left(\frac{\partial^2}{\partial x^2}\mathrm{u}(x,y,t)\right) - \left(\frac{\partial^2}{\partial t^2}\mathrm{u}(x,y,t)\right), \left(\frac{\partial}{\partial y}\mathrm{u}(x,y,t)\right) - \mathrm{u}(x,y,t)\right]$$

```
> J := JanetBasis(L, ivar, dvar);
```

$$J := \left[\left[\left(\frac{\partial}{\partial y}\mathrm{u}(x,y,t)\right) - \mathrm{u}(x,y,t), -\left(\frac{\partial}{\partial x}\mathrm{u}(x,y,t)\right) + \left(\frac{\partial^2}{\partial y\,\partial x}\mathrm{u}(x,y,t)\right), \left(\frac{\partial^2}{\partial x^2}\mathrm{u}(x,y,t)\right) - \left(\frac{\partial^2}{\partial t^2}\mathrm{u}(x,y,t)\right)\right], [x,y,t], [u]\right]$$

```
> HilbertSeries();
```

$$2s + 1 + \frac{2s^2}{1-s}$$

```
> SolSeries(J, 3, 'b');
```

$$\left[\mathrm{u}(x,y,t) = C1_{0,0,0} + C1_{1,0,0}\,x + C1_{0,0,0}\,y + C1_{0,0,1}\,t + \frac{1}{2}C1_{0,0,2}\,t^2 + C1_{1,0,1}\,xt + C1_{0,0,1}\,yt + \frac{1}{2}C1_{0,0,2}\,x^2 + C1_{1,0,0}\,xy + \frac{1}{2}C1_{0,0,0}\,y^2 + \frac{1}{6}C1_{0,0,3}\,t^3 + \frac{1}{2}C1_{1,0,2}\,xt^2 + \frac{1}{2}C1_{0,0,2}\,yt^2 + \frac{1}{2}C1_{0,0,3}\,x^2 t + C1_{1,0,1}\,xyt + \frac{1}{2}C1_{0,0,1}\,y^2 t + \frac{1}{6}C1_{1,0,2}\,x^3 + \frac{1}{2}C1_{0,0,2}\,x^2 y + \frac{1}{2}C1_{1,0,0}\,xy^2 + \frac{1}{6}C1_{0,0,0}\,y^3\right], [C1_{0,0,0}, C1_{1,0,0}, C1_{0,0,1}, C1_{0,0,2}, C1_{1,0,1}, C1_{0,0,3}, C1_{1,0,2}]$$

```
> b;
```

$$\left[\left[\mathrm{u}(x,y,t) = 1 + y + \frac{1}{2}y^2 + \frac{1}{6}y^3\right], \left[\mathrm{u}(x,y,t) = x + xy + \frac{1}{2}xy^2\right], \left[\mathrm{u}(x,y,t) = t + yt + \frac{1}{2}y^2 t\right], \left[\mathrm{u}(x,y,t) = \frac{1}{2}t^2 + \frac{1}{2}x^2 + \frac{1}{2}yt^2 + \frac{1}{2}x^2 y\right], [\mathrm{u}(x,y,t) = xt + xyt], \left[\mathrm{u}(x,y,t) = \frac{1}{6}t^3 + \frac{1}{2}x^2 t\right], \left[\mathrm{u}(x,y,t) = \frac{1}{2}xt^2 + \frac{1}{6}x^3\right]\right]$$

```
> SolSeries(J, 3, [1,2,-1]);
```

$$\left[\mathrm{u}(x,y,t) = C1_{0,0,0} + C1_{1,0,0}\,(x-1) + C1_{0,0,0}\,(y-2) + C1_{0,0,1}\,(t+1) + \frac{1}{2}C1_{0,0,2}\,(t+1)^2 + C1_{1,0,1}\,(x-1)(t+1) + C1_{0,0,1}\,(y-2)(t+1) + \frac{1}{2}C1_{0,0,2}\,(x-1)^2 + C1_{1,0,0}\,(x-1)(y-2) + \frac{1}{2}C1_{0,0,0}\,(y-2)^2 + \frac{1}{6}C1_{0,0,3}\,(t+1)^3 + \frac{1}{2}C1_{1,0,2}\,(x-1)(t+1)^2 + \frac{1}{2}C1_{0,0,2}\,(y-2)(t+1)^2 + \frac{1}{2}C1_{0,0,3}\,(x-1)^2(t+1) + C1_{1,0,1}\,(x-1)(y-2)(t+1) + \frac{1}{2}C1_{0,0,1}\,(y-2)^2(t+1) + \frac{1}{6}C1_{1,0,2}\,(x-1)^3 + \frac{1}{2}C1_{0,0,2}\,(x-1)^2(y-2) + \frac{1}{2}C1_{1,0,0}\,(x-1)(y-2)^2 + \frac{1}{6}C1_{0,0,0}\,(y-2)^3\right], [C1_{0,0,0}, C1_{1,0,0}, C1_{0,0,1}, C1_{0,0,2}, C1_{1,0,1}, C1_{0,0,3}, C1_{1,0,2}]$$

Example 2:

```
> ivar := [x,t=2]; dvar := [u];
```

$$ivar := [x, t=2]$$

$$dvar := [u]$$

```
> L := [diff(u(x,t),x,x)-diff(u(x,t),t)-sin(t)];
```

$$L := \left[\left(\frac{\partial^2}{\partial x^2}\mathrm{u}(x,t)\right) - \left(\frac{\partial}{\partial t}\mathrm{u}(x,t)\right) - \sin(t)\right]$$

```
> J := JanetBasis(L, ivar, dvar);
```

$$J := \left[\left[\left(\frac{\partial^2}{\partial x^2}\mathrm{u}(x,t)\right) - \left(\frac{\partial}{\partial t}\mathrm{u}(x,t)\right) - \sin(t)\right], [x,t], [u]\right]$$

```
> SolSeries([J[1],ivar,J[3]], [4], 'b');
```

$$\left[\mathrm{u}(x,t) = C1_{0,0} + C1_{1,0}\,x + C1_{0,1}\,t + \frac{1}{2}C1_{0,2}\,t^2 + C1_{1,1}\,xt + \frac{1}{2}C1_{0,1}\,x^2 + \frac{1}{2}C1_{0,2}\,x^2 t + \frac{x^2 t}{2} + \frac{1}{6}C1_{1,1}\,x^3 + \frac{1}{24}C1_{0,2}\,x^4 + \frac{x^4}{24}\right], [C1_{0,0}, C1_{1,0}, C1_{0,1}, C1_{0,2}, C1_{1,1}]$$

```
> b;
```

$$\left[\left[\mathrm{u}(x,t) = \frac{1}{2}x^2 t + \frac{1}{24}x^4\right], \left[\mathrm{u}(x,t) = 1 + \frac{1}{2}x^2 t + \frac{1}{24}x^4\right], \left[\mathrm{u}(x,t) = x + \frac{1}{2}x^2 t + \frac{1}{24}x^4\right], \left[\mathrm{u}(x,t) = t + \frac{1}{2}x^2 + \frac{1}{2}x^2 t + \frac{1}{24}x^4\right],\right.$$

$$\left[\mathrm{u}(x,t)=\frac{1}{2}t^2+x^2t+\frac{1}{12}x^4\right],\left[\mathrm{u}(x,t)=xt+\frac{1}{2}x^2t+\frac{1}{6}x^3+\frac{1}{24}x^4\right]\Bigg]$$

```
> SolSeries([J[1],ivar,J[3]], [[4,4]], 'b');
```

$$\left[\mathrm{u}(x,t)=C1_{0,0}+C1_{1,0}x+C1_{0,1}t+\frac{1}{2}C1_{0,2}t^2+C1_{1,1}xt+\frac{1}{2}C1_{0,1}x^2+\frac{1}{2}C1_{1,2}xt^2+\frac{1}{2}C1_{0,2}x^2t+\frac{x^2t}{2}+\frac{1}{6}C1_{1,1}x^3+\frac{1}{4}C1_{0,3}x^2t^2\right.$$
$$\left.+\frac{1}{6}C1_{1,2}x^3t+\frac{1}{24}C1_{0,2}x^4+\frac{x^4}{24}+\frac{1}{12}C1_{1,3}x^3t^2+\frac{1}{24}C1_{0,3}x^4t+\frac{1}{48}C1_{0,4}x^4t^2-\frac{x^4t^2}{48}\right],$$
$$[C1_{0,0},C1_{1,0},C1_{0,1},C1_{0,2},C1_{1,1},C1_{1,2},C1_{0,3},C1_{1,3},C1_{0,4}]$$

```
> b;
```

$$\left[\left[\mathrm{u}(x,t)=\frac{1}{2}x^2t+\frac{1}{24}x^4-\frac{1}{48}x^4t^2\right],\left[\mathrm{u}(x,t)=1+\frac{1}{2}x^2t+\frac{1}{24}x^4-\frac{1}{48}x^4t^2\right],\left[\mathrm{u}(x,t)=x+\frac{1}{2}x^2t+\frac{1}{24}x^4-\frac{1}{48}x^4t^2\right],\right.$$
$$\left[\mathrm{u}(x,t)=t+\frac{1}{2}x^2+\frac{1}{2}x^2t+\frac{1}{24}x^4-\frac{1}{48}x^4t^2\right],\left[\mathrm{u}(x,t)=\frac{1}{2}t^2+x^2t+\frac{1}{12}x^4-\frac{1}{48}x^4t^2\right],\left[\mathrm{u}(x,t)=xt+\frac{1}{2}x^2t+\frac{1}{6}x^3+\frac{1}{24}x^4-\frac{1}{48}x^4t^2\right],$$
$$\left[\mathrm{u}(x,t)=\frac{1}{2}xt^2+\frac{1}{2}x^2t+\frac{1}{6}x^3t+\frac{1}{24}x^4-\frac{1}{48}x^4t^2\right],\left[\mathrm{u}(x,t)=\frac{1}{2}x^2t+\frac{1}{4}x^2t^2+\frac{1}{24}x^4+\frac{1}{24}x^4t-\frac{1}{48}x^4t^2\right],$$
$$\left.\left[\mathrm{u}(x,t)=\frac{1}{2}x^2t+\frac{1}{24}x^4+\frac{1}{12}x^3t^2-\frac{1}{48}x^4t^2\right],\left[\mathrm{u}(x,t)=\frac{1}{2}x^2t+\frac{1}{24}x^4\right]\right]$$

Example 3: (change the name for constants)

```
> J := JanetBasis([diff(u(x),x$5)], [x], [u]);
```

$$J:=\left[\left[\frac{d^5}{dx^5}\mathrm{u}(x)\right],[x],[u]\right]$$

```
> SolSeries(J, 3, "constname"=c);
```

$$\left[\mathrm{u}(x)=c1_0+c1_1x+\frac{1}{2}c1_2x^2+\frac{1}{6}c1_3x^3\right],[c1_0,c1_1,c1_2,c1_3]$$

See Also:

JanetBasis, PrincDeriv, ParamDeriv, InvReduce, CompCond, CompCondBasis, HilbertSeries, HilbertPolynomial, HilbertFunction, PolySol, ZeroSets.

Janet[SubOp] - compute the difference of two linear differential operators in matrix form

Calling Sequence:

SubOp(oper1,oper2,ivar)

Parameters:

`oper1` - linear differential operator in matrix form
`oper2` - linear differential operator in matrix form
`ivar` - list of independent variables

Description:

- ***SubOp*** forms the difference of the linear differential operators given by the matrices **`oper1`** and **`oper2`**, the entries of which are described below.
- The entries of the input and output matrices have the form $[[c_1, [...]], ..., [c_r, [...]]]$, where the dots in the innermost brackets are sequences of independent variables from **`ivar`** representing partial derivatives w. r. t. these variables, and the c_i are the coefficients for these, i. e. functions in **`ivar`**. The whole bracket represents the sum of these items, i. e. the linear combination of the above operators. Note, if an innermost bracket is empty, it represents the identity operator. (For a convenient way to produce such a matrix use Diff2Op as in the example below.)

Examples:

```
> with(Janet):
```

Example 1:

```
> ivar := [x]; dvar := [u];
```

$$ivar := [x]$$

$$dvar := [u]$$

```
> o1 := Diff2Op([diff(u(x), x)+u(x)], ivar, dvar);
```

$$o1 := [[[1, [x]], [1, [\ ]]]]$$

```
> o2 := Diff2Op([x*diff(u(x), x)], ivar, dvar);
```

$$o2 := [[[x, [x]]]]$$

```
> SubOp(o1, o2, ivar);
```

$$[[[1 - x, [x]], [1, [\ ]]]]$$

Example 2:

```
> ivar := [x,y]; dvar := [u,v];
```

$$ivar := [x, y]$$

$$dvar := [u, v]$$

```
> L1 := [y*diff(u(x,y),y)+diff(v(x,y),x$2), diff(v(x,y),x)+2*x^2*v(x,y)];
```

$$L1 := \left[y\left(\frac{\partial}{\partial y} u(x,y)\right) + \left(\frac{\partial^2}{\partial x^2} v(x,y)\right), \left(\frac{\partial}{\partial x} v(x,y)\right) + 2x^2 v(x,y) \right]$$

```
> o1 := Diff2Op(L1, ivar, dvar);
```

$$o1 := \begin{bmatrix} [[y, [y]]] & [[1, [x, x]]] \\ 0 & [[1, [x]], [2x^2, [\ ]]] \end{bmatrix}$$

```
> L2 := [u(x,y)+diff(u(x,y),x), diff(v(x,y),y)];
```

$$L2 := \left[u(x,y) + \left(\frac{\partial}{\partial x} u(x,y)\right), \frac{\partial}{\partial y} v(x,y) \right]$$

```
> o2 := Diff2Op(L2, ivar, dvar);
```

$$o2 := \begin{bmatrix} [[1, [x]], [1, [\ ]]] & 0 \\ 0 & [[1, [y]]] \end{bmatrix}$$

```
> SubOp(o1, o2, ivar);
```

$$\begin{bmatrix} [[-1, [x]], [y, [y]], [-1, [\ ]]] & [[1, [x, x]]] \\ 0 & [[1, [x]], [-1, [y]], [2\,x^2, [\ ]]] \end{bmatrix}$$

See Also:

Diff2Op, AddOp, AppOp, AppOpInd, AddOp, SubOp, JAdjoint, Op2D, D2Op, CmpD.

Janet[SyzOp] - return compatibility conditions in operator language

Calling Sequence:

SyzOp(oper,ivar)

Parameters:

`oper` - differential operator in matrix form
`ivar` - list of independent variables

Description:

- ***SyzOp*** is synonymous to CompCond with general right hand sides, but uses operator terminology for input and output rather than the usual differential expression terminology.
- Input and output are matrices. Their entries have the form $[[c_1, [...]], ..., [c_r, [...]]]$, where the dots in the innermost brackets are sequences of independent variables from **`ivar`** representing partial derivatives w. r. t. these variables, and the c_i are the coefficients for these, i. e. functions in **`ivar`**. The whole bracket represents the sum of these items, i. e. the linear combination of the above operators. Note, if an innermost bracket is empty, it represents the identity operator. (For a convenient way to produce such a matrix use Diff2Op as in the example below.)
- In the terminolgy of differentiation and functions ***SyzOp*** computes the possible right hand sides of the inhomogeneous PDE system whose associated homogeneous system is represented by **`oper`** (applied to a column of indeterminate functions). For these right hand sides ***SyzOp*** produces a linear PDE system in operator terminology, i. e. the solutions of the PDE system form the possible right hand sides.
- In the terminology of modules over the ring D of differential operators with respect to the independent variables from **`ivar`**, the matrices involved represent module homomorphisms of the free left D-modules of rows with entries in D of appropriate length and are understood as multiplications from the right. In this language, ***SyzOp*** computes a homomorphism from the input **`oper`** such that the kernel of **`oper`** is equal to the image of the resulting operator.
- Internally CompCond is used.
- Iterated use of ***SyzOp*** produces a free resolution of the cokernel of **`oper`** as left D-modules.

Examples:

```
> with(Janet):
> ivar := [x,y]; dvar := [u];
```

$$ivar := [x, y]$$

$$dvar := [u]$$

```
> L := [diff(u(x,y),x,x), diff(u(x,y),x,y)-diff(u(x,y),x), diff(u(x,y),y,y)];
```

$$L := \left[\frac{\partial^2}{\partial x^2}u(x,y), \left(\frac{\partial^2}{\partial y\,\partial x}u(x,y)\right) - \left(\frac{\partial}{\partial x}u(x,y)\right), \frac{\partial^2}{\partial y^2}u(x,y)\right]$$

```
> oper := Diff2Op(L, ivar, dvar);
```

$$oper := \begin{bmatrix} [[1, [x, x]]] \\ [[1, [x, y]], [-1, [x]]] \\ [[1, [y, y]]] \end{bmatrix}$$

```
> sop := SyzOp(oper, ivar);
```

$$sop := \begin{bmatrix} 0 & [[-1, [y, y]]] & [[1, [x, y]], [-1, [x]]] \\ [[-1, [\,]]] & [[-1, [x, y]], [-1, [x]]] & [[1, [x, x]]] \\ 0 & [[-1, [y, y, y]], [-1, [y, y]]] & [[1, [x, y, y]], [-1, [x]]] \end{bmatrix}$$

```
> CmpOp(sop, oper, ivar);
```

$$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

```
> sop1:=SyzOp(sop,ivar);
```

$$sop1 := [[[1, [y]], [1, [\ ]]] \quad 0 \quad [[-1, [\ ]]]]$$

```
> sop2:=SyzOp(sop1,ivar);
```

$$sop2 := \mathrm{array}(1 .. 0, 1 .. 1, [\ ])$$

Hence we have constructed a free resolution for the cokernel of **oper** of length 2. Here comes an example with non-constant coefficients:

```
> ivar := [x,y]; dvar := [u,v];
```

$$ivar := [x, y]$$

$$dvar := [u, v]$$

```
> L := [x*diff(u(x,y),x)-diff(v(x,y),y), y*diff(u(x,y),y)-diff(v(x,y),x), diff(u(x,y),
  x,y)];
```

$$L := \left[x\left(\frac{\partial}{\partial x}\mathrm{u}(x,y)\right) - \left(\frac{\partial}{\partial y}\mathrm{v}(x,y)\right), y\left(\frac{\partial}{\partial y}\mathrm{u}(x,y)\right) - \left(\frac{\partial}{\partial x}\mathrm{v}(x,y)\right), \frac{\partial^2}{\partial y\,\partial x}\mathrm{u}(x,y) \right]$$

```
> oper := Diff2Op(L, ivar, dvar);
```

$$oper := \begin{bmatrix} [[x, [x]]] & [[-1, [y]]] \\ [[y, [y]]] & [[-1, [x]]] \\ [[1, [x, y]]] & 0 \end{bmatrix}$$

```
> sop := SyzOp(oper, ivar);
```

$$sop := [[[1, [x, x, y]]] \quad [[-1, [x, y, y]]] \quad [[-x, [x, x]], [y, [y, y]], [-2, [x]], [2, [y]]]]$$

```
> CmpOp(sop, oper, ivar);
```

$$[\ 0 \quad 0]$$

See Also:

JanetBasis, PrincDeriv, ParamDeriv, CompCond, CompCondBasis, Resolution, AffEqn, AppOp, CmpOp, Diff2Op.

Janet[PrincDeriv],

Janet[TabVar] - **display Janet's data, i. e. all equations, leading derivatives, multiplicative variables etc.**

Calling Sequence:

PrincDeriv()
TabVar()

Parameters:

- - none (assumes that the Janet basis has been computed before)

Description:

- ***PrincDeriv*** displays the data constructed by JanetBasis. Therefore it is necessary to call ***JanetBasis*** first. The data structure is a list of lists each corresponding to an element of the Janet basis. ***TabVar*** is a synonym for ***PrincDeriv***.
- The entries of each list are the basis element, the list of multiplicative / non-multiplicative variables and the leading derivative. The variables in the second entry are the multiplicative variables of the respective leading derivative. Non-multiplicative variables are represented as '*'.
- Note, the data determine directly all possible principal derivatives, i. e. those derivatives which cannot occur in a reduced differential expression. These are the derivatives of the leading monomials in the Janet basis by multiplicative variables.

Examples:

```
> with(Janet):
> ivar := [x,y,t]; dvar := [u];
```

$$ivar := [x, y, t]$$

$$dvar := [u]$$

```
> L := [diff(u(x,y,t),x$2) - y*diff(u(x,y,t),t$2), diff(u(x,y,t),y$2)];
```

$$L := \left[\left(\frac{\partial^2}{\partial x^2}\,u(x,y,t)\right) - y\left(\frac{\partial^2}{\partial t^2}\,u(x,y,t)\right), \frac{\partial^2}{\partial y^2}\,u(x,y,t)\right]$$

```
> JanetBasis(L, ivar, dvar):
> PrincDeriv();
```

$$\left[\frac{\partial^2}{\partial y^2}\,u(x,y,t), [*, y, t], \frac{\partial^2}{\partial y^2}\,u(x,y,t)\right]$$

$$\left[\left(\frac{\partial^2}{\partial x^2}\,u(x,y,t)\right) - y\left(\frac{\partial^2}{\partial t^2}\,u(x,y,t)\right), [x, y, t], \frac{\partial^2}{\partial x^2}\,u(x,y,t)\right]$$

$$\left[\frac{\partial^3}{\partial y\,\partial t^2}\,u(x,y,t), [*, *, t], \frac{\partial^3}{\partial y\,\partial t^2}\,u(x,y,t)\right]$$

$$\left[\frac{\partial^3}{\partial y^2\,\partial x}\,u(x,y,t), [*, y, t], \frac{\partial^3}{\partial y^2\,\partial x}\,u(x,y,t)\right]$$

$$\left[\frac{\partial^4}{\partial t^4}\,u(x,y,t), [*, *, t], \frac{\partial^4}{\partial t^4}\,u(x,y,t)\right]$$

$$\left[\frac{\partial^4}{\partial y\,\partial x\,\partial t^2}\,u(x,y,t), [*, *, t], \frac{\partial^4}{\partial y\,\partial x\,\partial t^2}\,u(x,y,t)\right]$$

$$\left[\frac{\partial^5}{\partial x\,\partial t^4}\,u(x,y,t), [*, *, t], \frac{\partial^5}{\partial x\,\partial t^4}\,u(x,y,t)\right]$$

See Also:

JanetBasis, ParamDeriv, InvReduce, HilbertSeries, PDEBasis, PDEHilbertSeries.

Janet[Autonom],

Janet[Torsion] - return torsion module of a differential module

Calling Sequence:

Autonom(L,ivar,dvar)
Torsion(L,ivar,dvar)

Parameters:

`L` - list of linear differential expressions (to be interpreted as module generators)
`ivar` - list of independent variables
`dvar` - list of dependent variables

Description:

- In the language of right D-modules, the command ***Torsion*** computes the torsion submodule of the right D-module whose presentation is given by **L**. It lists generators for the torsion submodule in terms of the original generators, a presentation of the torsion module with respect to its generators, and finally the Hilbert series (see HilbertSeries) for the torsion module with respect to degrevlex.
- ***Autonom*** is a synonym for ***Torsion***.

Examples:

```
> with(Janet):
```

Example 1:

```
> ivar := [t]; dvar := [u,v,w];
```

$$ivar := [t]$$

$$dvar := [u, v, w]$$

```
> R := Ind2Diff([u[t]-v-w, -u+v[t]+w], ivar, dvar);
```

$$R := \left[\left[\left(\frac{d}{dt}\,u(t)\right) - v(t) - w(t),\ -u(t) + \left(\frac{d}{dt}\,v(t)\right) + w(t)\right]\right]$$

```
> Torsion(R, ivar, dvar);
```

$$\left[[\_T1(t) = u(t) + v(t)],\ \left[-\_T1(t) + \left(\frac{d}{dt}\,\_T1(t)\right)\right],\ 1\right]$$

The torsion submodule is generated by $\_T1(t) = u(t) + v(t)$ which satisfies the equation $\frac{d}{dt}\,\_T1(t) = \_T1(t)$.

Example 2:

```
> ivar := [x,y]; dvar := [u,v];
```

$$ivar := [x, y]$$

$$dvar := [u, v]$$

```
> R := [diff(u(x,y),x)+diff(v(x,y),y)+y*u(x,y)];
```

$$R := \left[\left[\left(\frac{\partial}{\partial x}\,u(x,y)\right) + \left(\frac{\partial}{\partial y}\,v(x,y)\right) + y\,u(x,y)\right]\right]$$

```
> Torsion(R, ivar, dvar);
```

$$[[\_T1(x,y) = 0],\ [\_T1(x,y)],\ 0]$$

The torsion submodule is trivial.

See Also:

JanetBasis, PrincDeriv, InvReduce, HilbertSeries, CompCond, CompCondBasis, SyzOp, Resolution, Ext1, Extn, AutonomEq, Diff2Op, AppOp.

Janet[WeightedHilbertSeries] - generalized Hilbert series (weighted version)

Calling Sequence:

WeightedHilbertSeries(ivar,dvar,v)

Parameters:

`ivar` - list of independent variables (with degrees)
`dvar` - list of dependent variables (with degrees)
`v` - (optional) name of the indeterminate (default 's')

Description:

- ***WeightedHilbertSeries*** returns the Hilbert series of the Janet basis produced by the last call of JanetBasis with respect to specially assigned degrees. Therefore it is necessary to call JanetBasis first. The output is the generating function of the number of free Taylor coefficients of the general solution of the system, whose Janet basis has been computed last where degrees are taken as follows:
- The parameters **`dvar`** and **`ivar`** are of the form $[u_1 = e_1, ..., u_k = e_k]$ and $[x_1 = d_1, ..., x_n = d_n]$ resp., where k is the number of dependent variables and n is the number of independent variables. The variable u_i and x_i is assigned non negative integers e_i and natural numbers d_i resp. as degrees. If an entry of these lists is not an equation, but consists of the variable only, this variable gets the default degree, which is 0 for dependent variables and 1 for independent variables.
- The output can be expanded in the form $\sum_{i=0}^{\infty} d_i v^i$, where d_i is the number of parametric derivatives of weighted order i as described above.
- The default name of the indeterminate is 's'. To use the subs command later on the indeterminate, one has to explicitly give a name to the indeterminate.
- To enumerate rather than count the parametric derivatives cf. ParamDeriv.
- The special case where all degrees e_i are equal to 1 and all degrees d_j are equal to 0 yields the same result (in a different expansion) as HilbertSeries.

Examples:

```
> with(Janet):
> dvar := [u]:
> L := [diff(u(x,t), x$2)-diff(u(x,t), t)];
```

$$L := \left[\left(\frac{\partial^2}{\partial x^2}\,u(x,t)\right) - \left(\frac{\partial}{\partial t}\,u(x,t)\right)\right]$$

```
> J := JanetBasis(L, [x,t=2], dvar);
```

$$J := \left[\left[\left(\frac{\partial^2}{\partial x^2}\,u(x,t)\right) - \left(\frac{\partial}{\partial t}\,u(x,t)\right)\right], [x,t], [u]\right]$$

The classical Hilbert series

```
> HilbertSeries(lambda);
```

$$1 + 2\lambda + 2\frac{\lambda^2}{1-\lambda}$$

differs from the one, where t gets the weight 2:

```
> WeightedHilbertSeries([x,t=2], dvar, lambda);
```

$$\frac{1}{1-\lambda^2} + \frac{\lambda}{1-\lambda^2}$$

```
> ParamDeriv([x,t], dvar);
```

$$\frac{1}{1-t} + \frac{x}{1-t}$$

See Also:

JanetBasis, PrincDeriv, ParamDeriv, HilbertSeries, HilbertPolynomial, HilbertFunction, CartanCharacters.

Janet[ZeroSets] - return the functions and their zeros by which the Janet basis algorithm had to divide

Calling Sequence:

ZeroSets(ivar,v)

Parameters:

`ivar` - list of independent variables
`v` - (optional) an independent variable

Description:

- When calculating a Janet basis, one might have to perform divisions by functions in the independent variables **`ivar`**. ***ZeroSets*** returns a list of these functions together with their zero sets. Note, the command SolSeries can only be used for centres of expansion outside the union of these zero sets. There might be additional restrictions from functional coefficients of leading terms in the Janet basis, which the Janet algorithm left untouched. In case an argument **`v`** is used, it only returns the subset of those functions which depend on **`v`** and those **`v`**-values that can be complemented to zeros of these functions. In any case the command refers to the last call of JanetBasis.
- Caution: ***ZeroSets*** uses the Maple function solve for computing the zero sets from the functions.

Examples:

```
> with(Janet):
> ivar := [x,y]: dvar := [u,v]:
> L := [diff(u(x,y),x) - y*diff(v(x,y),y),cos(y)*diff(u(x,y),y) + x^2*diff(v(x,y),x),
  diff(u(x,y),x,x) - y*x];
```

$$L := \left[\left(\frac{\partial}{\partial x}\mathrm{u}(x,y)\right) - y\left(\frac{\partial}{\partial y}\mathrm{v}(x,y)\right), \cos(y)\left(\frac{\partial}{\partial y}\mathrm{u}(x,y)\right) + x^2\left(\frac{\partial}{\partial x}\mathrm{v}(x,y)\right), \left(\frac{\partial^2}{\partial x^2}\mathrm{u}(x,y)\right) - yx\right]$$

```
> JB := JanetBasis(L, ivar, dvar);
```

$$JB := \left[\left[\cos(y)\left(\frac{\partial}{\partial y}\mathrm{u}(x,y)\right) + x^2\left(\frac{\partial}{\partial x}\mathrm{v}(x,y)\right)\left(\frac{\partial}{\partial x}\mathrm{u}(x,y)\right) - y\left(\frac{\partial}{\partial y}\mathrm{v}(x,y)\right), \cos(y)\left(\frac{\partial^2}{\partial y^2}\mathrm{u}(x,y)\right) - \sin(y)\left(\frac{\partial}{\partial y}\mathrm{u}(x,y)\right) + x^3,\right.$$
$$\left. y\cos(y)\left(\frac{\partial^3}{\partial y^3}\mathrm{v}(x,y)\right) + (2\cos(y) - y\sin(y))\left(\frac{\partial^2}{\partial y^2}\mathrm{v}(x,y)\right) - \sin(y)\left(\frac{\partial}{\partial y}\mathrm{v}(x,y)\right) + 3x^2\right], [x,y], [u,v]\right]$$

```
> ZeroSets(ivar);
```

$$\left[[x^2, \{x = 0\}], [y, \{y = 0\}], \left[\cos(y), \{y = \frac{1}{2}\pi\}\right]\right]$$

```
> ZeroSets(ivar, x);
```

$$[[x^2, \{x = 0\}]]$$

See Also:

JanetBasis, PrincDeriv, CompCond, CompCondBasis, SolSeries, PolySol.