

```
In[1]:= << HolonomicFunctions.m
```

```
HolonomicFunctions package by Christoph Koutschan, RISC-Linz, Version 1.4 (10.11.2010)
→ Type ?HolonomicFunctions for help
```

```
In[2]:= (* General information on HolonomicFunctions *)
```

```
In[3]:= ?HolonomicFunctions
```

The main objective of this package is the algorithmic manipulation of ∂ -finite (holonomic) functions. This includes (but is not restricted to) proving special function identities, finding recurrences, differential equations or relations of mixed type for ∂ -finite functions, and computing definite sums and integrals of ∂ -finite functions. Type ?DFinite to get the definition and a short introduction to ∂ -finite functions.

The following commands serve the above objectives: Annihilator, CreativeTelescoping, FindRelation, FindSupport, Takayama, ApplyOreOperator, UnderTheStaircase, AnnihilatorDimension.

The closure properties of ∂ -finite functions are implicitly executed in Annihilator. To execute them explicitly, use the commands DFinitePlus, DFiniteTimes, DFiniteSubstitute, DFiniteOreAction, DFiniteTimesHyper, DFiniteDE2RE, DFiniteRE2DE.

An important ingredient are Groebner bases in (noncommutative) Ore algebras: OreGroebnerBasis, OreReduce, GBEqual, FGLM.

A common subtask in the above algorithms is finding rational solutions of P-finite recurrences / differential equations or of coupled systems of such equations. The following commands address these purposes: RSolvePolynomial, RSolveRational, DSolvePolynomial, DSolveRational, QSolvePolynomial, QSolveRational, SolveOreSys, SolveCoupledSystem.

An element of an Ore algebra is called an Ore polynomial; the following commands explain the data type OrePolynomial that is introduced in this package and how to deal with it: OrePolynomial, ToOrePolynomial, OrePolynomialZeroQ, LeadingPowerProduct, LeadingExponent, LeadingCoefficient, LeadingTerm, OrePolynomialListCoefficients, NormalizeCoefficients, OrePlus, OreTimes, OrePower, ApplyOreOperator, ChangeOreAlgebra, ChangeMonomialOrder, OrePolynomialSubstitute, OrePolynomialDegree, Support.

In order to define own Ore algebras use the commands OreAlgebra, OreAlgebraGenerators, OreAlgebraOperators, OreAlgebraPolynomialVariables, OreOperators, OreOperatorQ, OreSigma, OreDelta, OreAction, Der, S, Delta, Euler, QS.

Some other functions that might be useful: Printlevel, RandomPolynomial.

If this package was useful in your scientific work, proper citation would be appreciated very much. Please use the following reference for this purpose:

```
@phdthesis{Koutschan09,
  author = {Christoph Koutschan},
  title = {Advanced Applications of the Holonomic Systems Approach},
  school = {RISC, Johannes Kepler University},
  address = {Linz, Austria},
  year = {2009}
}
```

```
In[4]:= ann = Annihilator[Sum[Binomial[n, k] LegendreP[k, x], {k, 0, n}], {S[n]}]
```

```
Out[4]= { (2 + n) S_n^2 + (-3 - 2 n - 3 x - 2 n x) S_n + (2 + 2 n + 2 x + 2 n x) }
```

```
In[5]:= Factor[ann]
```

```
Out[5]= { (2 + n) S_n^2 - (3 + 2 n) (1 + x) S_n + 2 (1 + n) (1 + x) }
```

```
In[6]:= (* executing closure properties *)
```

```
In[7]:= ann1 = Annihilator[LegendreP[k, x], {S[k]}]
```

```
Out[7]= { (2 + k) S_k^2 + (-3 x - 2 k x) S_k + (1 + k) }
```

```

In[8]:= ann2 = Annihilator[Binomial[n, k], {S[k]}]
Out[8]= {(1 + k) Sk + (k - n)}

In[9]:= ann = Factor[DFiniteTimes[ann1, ann2]]
Out[9]= {(2 + k)2 Sk2 + (3 + 2 k) (1 + k - n) x Sk + (k - n) (1 + k - n)}

In[10]:= (* available routines executing closure properties *)
In[11]:= ?DFinite*

▼ HolonomicFunctions'


| DFinite      | DFiniteOreAction | DFiniteRE2DE      | DFiniteTimes      |
|--------------|------------------|-------------------|-------------------|
| DFiniteDE2RE | DFinitePlus      | DFiniteSubstitute | DFiniteTimesHyper |



In[12]:= (* creative telescoping in HolonomicFunctions *)
In[13]:= CreativeTelescoping[Binomial[n, k] LegendreP[k, x], S[k] - 1, S[n]]
Out[13]= { { (2 + n) Sn2 + (-3 - 2 n - 3 x - 2 n x) Sn + (2 + 2 n + 2 x + 2 n x) },
  {  $\frac{1 + 2 k + k^2 + n + 2 k n + k^2 n}{-k + k^2 + n - 2 k n + n^2} S_k + \frac{k^2 + k^2 n - 2 x - 3 k x + 2 k^2 x - 3 n x - 5 k n x + 2 k^2 n x - n^2 x - 2 k n^2 x}{2 - 3 k + k^2 + 3 n - 2 k n + n^2}$  } }

In[14]:= (* recurrence for L2-orthogonal basis functions on triangles *)
In[15]:= ann = Factor[
  Annihilator[LegendreP[i, 2 x / (1 - y)] ((1 - y) / 2)i JacobiP[j, 2 i + 1, 0, y], {S[i], S[j]}]];
In[16]:= Support[ann]
Out[16]= { {Sj2, Sj, 1}, {Si2, Si Sj, Si, Sj, 1} }

In[17]:= (* the first part is merely the three term
  recurrence for Jacobi polynomials JacobiP[j, 2i+1, 0, y] *)
In[18]:= First[ann]
Out[18]= (2 + j) (3 + 2 i + j) (3 + 2 i + 2 j) Sj2 -
  (2 + i + j) (1 + 4 i + 4 i2 + 15 y + 16 i y + 4 i2 y + 16 j y + 8 i j y + 4 j2 y) Sj +
  (1 + j) (2 + 2 i + j) (5 + 2 i + 2 j)

In[19]:= ann = Factor[Annihilator[LegendreP[i, 2 x / (1 - y)] ((1 - y) / 2)i JacobiP[j, 2 i + 1, 0, y],
  {S[i], S[j], Der[x], Der[y]}]];
In[20]:= Support[ann]
Out[20]= { {Sj, Dx, Dy, 1}, {Dy2, Si, Dx, Dy, 1}, {Dx Dy, Si, Dx, Dy, 1},
  {Dx2, Dx, 1}, {Si Dy, Si, Dx, Dy, 1}, {Si Dx, Si, Dx, Dy, 1}, {Si2, Si, Dx, Dy, 1} }

In[21]:= (* conversion monomial basis to Legendre polynomials *)
In[22]:= ann = Annihilator[Integrate[xk LegendreP[n, x], {x, -1, 1}],
  {S[k], S[n]}, Assumptions → Element[k, Integers] && Element[n, Integers]]
Out[22]= {(1 + k - n) Sk + (-1 - k) Sn, (-3 - k - n) Sn2 + (k - n)}

```

In[23]:= **ApplyOreOperator[ann, a[k, n]]**

Out[23]= $\{(-1-k) a[k, 1+n] + (1+k-n) a[1+k, n], (k-n) a[k, n] + (-3-k-n) a[k, 2+n]\}$

In[24]:= **(* without the assumptions the delta part cannot be evaluated *)**

In[25]:= **ann = Annihilator[Integrate[x^k LegendreP[n, x], {x, -1, 1}], {S[k], S[n]}]**

FullSimplify::infd :

Expression $(-1)^k (\text{LegendreP}[n, -1] + \text{LegendreP}[1+n, -1])$ simplified to Indeterminate. >>

Annihilator::fail : Cannot evaluate the inhomogeneous part of an integral.

Out[25]= \$Failed

In[26]:= **(* one option is to ask to return the delta part
(as earlier in the CreativeTelescoping example) and do the evaluation oneself *)**

In[27]:= **ann =**

Annihilator[Integrate[x^k LegendreP[n, x], {x, -1, 1}], {S[k], S[n]}, Inhomogeneous -> True]

Out[27]= $\left\{ \left\{ (1+k-n) S_k + (-1-k) S_n, (-3-k-n) S_n^2 + (k-n) \right\}, \right.$
 $\left. \left\{ (-1)^{1+k} (-\text{LegendreP}[n, -1] - \text{LegendreP}[1+n, -1]), \right. \right.$
 $\left. \left. - \frac{(-1)^{1+k} (3+2n) (-\text{LegendreP}[n, -1] - \text{LegendreP}[1+n, -1])}{2+n} \right\} \right\}$

In[28]:= **rem = Last[%]**

Out[28]= $\left\{ (-1)^{1+k} (-\text{LegendreP}[n, -1] - \text{LegendreP}[1+n, -1]), \right.$
 $\left. - \frac{(-1)^{1+k} (3+2n) (-\text{LegendreP}[n, -1] - \text{LegendreP}[1+n, -1])}{2+n} \right\}$

In[29]:= **(* use the knowledge that Legendre polynomials at x=-1 evaluate to (-1)^n *)**

In[30]:= **rem /. LegendreP[n_, -1] -> (-1)^n**

Out[30]= $\left\{ (-1)^{1+k} (-(-1)^n - (-1)^{1+n}), - \frac{(-1)^{1+k} (-(-1)^n - (-1)^{1+n}) (3+2n)}{2+n} \right\}$

In[31]:= **% // FullSimplify**

Out[31]= $\{0, 0\}$