Appendix A

Maple library: GiosTools

The GiosTools maple library is a collection of tools useful for working with recurrence relations, hypergeometric functions, differential equations and other objects related to building the helium wave function. It includes some simple shortcuts to frequently-used maple commands, as well as a number of more involved, original functionalities such as coordinate system handling (type/coordsys), &>, assumecoords), identification of sequences and series (IdSeq and IdSer), automatic derivation of recurrence relations (RecRel), inversion of the Laplacian on a given polynomial term (invertpolynomial), collection of terms in arbitrary rational powers (Collect), and translation of maple code to LaTeX (getTeX). In the following, the usage of the functionalities provided by the library will be explained. A compact cheat sheet listing all calling sequences is available on Page 144.

A.1 Installation

Installation wizard (Windows only)

Execute `GiosTools.exe`. All steps described under "Manual installation" will be performed automatically for the main Maple installation (i.e. the program assigned to open `.mw` files).

Manual installation

First, copy the binary file containing the library `GiosTools.mla` to any location (e.g. `"C:\\Program Files (x86)\MapleLibraries\"`). It is advisable to use some location outside of the Maple installation directory, as this will preserve the library even if Maple is reinstalled. At startup, Maple needs to be informed about the location of the `GiosTools.mla` file. For this purpose, locate the `maple.ini` file in the main Maple installation directory (e.g. `"C:\\Program Files (x86)\Maple 16\lib\maple.ini"`); if no such file exists, simply create a new file of this name. Append to the `maple.ini` file the line `libname := PathToLibrary, libname:` (in the above example, this line would read `libname := "C:\\Program Files (x86)\MapleLibraries\", libname:`).

Usage

After installation, the GiosTools library can be loaded into any worksheet via the command

```
with(GiosTools).
```

A.2 Functionalities

A.2.1 Predefined coordinate systems: type/coordsys

Parameters

coordsys: A Coordinate system specified in any of the following forms:

1. A name of a predefined coordinate system.
2. A set or list of three expressions of either of the following forms:
   a. A predefined coordinate.
   b. An expression defining a coordinate in terms of predefined coordinates.
   c. An equation name=expression defining a named coordinate in terms of predefined coordinates.

Description

A number of frequently used coordinate systems have been predefined and can be accessed via their names. For example, ri refers to the interparticle coordinate system $\{r_1, r_2, r_{12}\}$, rad refers to the coordinate system $\{r, a, d\}$ containing hyperradius, cosine of the hyperangle, and scaled inter-electron distance, and so on. The list of all 21 predefined coordinate systems can be displayed with the command showcoord(systems):

```
> showcoord(systems);
```

"Name"	"Coordinates"	"r1r2r12 -> Coordinate system"	"Coordinate system -> r1r2r12"
ralphad	[r, α, d]	$\left[r=\sqrt{r1^2+r2^2},\ \alpha=\arccos\left(\frac{r1^2-r2^2}{r1^2+r2^2}\right),\ d=\frac{r12}{\sqrt{r1^2+r2^2}}\right]$	$\left[r1=\frac{1}{2}\sqrt{2}\,r\sqrt{1+\cos(\alpha)},\ r2=\frac{1}{2}\sqrt{2}\,r\sqrt{1-\cos(\alpha)},\ r12=r\,d\right]$
rad	[r, a, d]	$\left[r=\sqrt{r1^2+r2^2},\ a=\frac{r1^2-r2^2}{r1^2+r2^2},\ d=\frac{r12}{\sqrt{r1^2+r2^2}}\right]$	$\left[r1=\frac{1}{2}\sqrt{2}\,r\sqrt{1+a},\ r2=\frac{1}{2}\sqrt{2}\,r\sqrt{1-a},\ r12=r\,d\right]$
ralphatheta	[r, α, θ]	$\left[r=\sqrt{r1^2+r2^2},\ \alpha=\arccos\left(\frac{r1^2-r2^2}{r1^2+r2^2}\right),\ \theta=\arccos\left(\frac{1}{2}\,\frac{r1^2-r12^2+r2^2}{r1\,r2}\right)\right]$	$\left[r1=\frac{1}{2}\sqrt{2}\,r\sqrt{1+\cos(\alpha)},\ r2=\frac{1}{2}\sqrt{2}\,r\sqrt{1-\cos(\alpha)},\ r12=r\sqrt{1-\sin(\alpha)\cos(\theta)}\right]$
ralphav	[r, α, v]	$\left[r=\sqrt{r1^2+r2^2},\ \alpha=\arccos\left(\frac{r1^2-r2^2}{r1^2+r2^2}\right),\ v=\frac{1}{2}\,\frac{r1^2-r12^2+r2^2}{r1\,r2}\right]$	$\left[r1=\frac{1}{2}\sqrt{2}\,r\sqrt{1+\cos(\alpha)},\ r2=\frac{1}{2}\sqrt{2}\,r\sqrt{1-\cos(\alpha)},\ r12=r\sqrt{1-v\sin(\alpha)}\right]$
rav	[r, a, v]	$\left[r=\sqrt{r1^2+r2^2},\ a=\frac{r1^2-r2^2}{r1^2+r2^2},\ v=\frac{1}{2}\,\frac{r1^2-r12^2+r2^2}{r1\,r2}\right]$	$\left[r1=\frac{1}{2}\sqrt{2}\,r\sqrt{1+a},\ r2=\frac{1}{2}\sqrt{2}\,r\sqrt{1-a},\ r12=r\sqrt{1-v\sqrt{1-a^2}}\right]$
rbv	[r, b, v]	$\left[r=\sqrt{r1^2+r2^2},\ b=\frac{2\,r1\,r2}{r1^2+r2^2},\ v=\frac{1}{2}\,\frac{r1^2-r12^2+r2^2}{r1\,r2}\right]$	$\left[r1=\frac{r\,b}{\sqrt{1+b}+\sqrt{1-b}},\ r2=\frac{1}{2}\,r\left(\sqrt{1+b}+\sqrt{1-b}\right),\ r12=r\sqrt{1-v\,b}\right]$
raw	[r, a, w]	$\left[r=\sqrt{r1^2+r2^2},\ a=\frac{r1^2-r2^2}{r1^2+r2^2},\ w=\frac{r1^2-r12^2+r2^2}{r1^2+r2^2}\right]$	$\left[r1=\frac{1}{2}\sqrt{2}\,r\sqrt{1+a},\ r2=\frac{1}{2}\sqrt{2}\,r\sqrt{1-a},\ r12=r\sqrt{1-w}\right]$

...

Alternatively, one can specify a coordinate system by providing a list or set of three coordinates. Again, many frequently used coordinates are directly accessible through their names; for example, r1 stands for the interparticle distance r_1, alpha stands for the hyperangle α, etc. The list of all 27 predefined coordinates can be displayed with the command showcoord(coords):

```
> showcoord(coords);
```

Coordinate definitions	Relations etc.
$r = \sqrt{r1^2 + r2^2}$	Hyperradius
$\alpha = \arccos\left(\frac{r1^2 - r2^2}{r1^2 + r2^2}\right)$	Hyperangle
$d = \frac{r12}{\sqrt{r1^2 + r2^2}}$	$d = \frac{r12}{r}$
$a = \frac{r1^2 - r2^2}{r1^2 + r2^2}$	$a = \cos(\alpha)$

...

A coordinate that has not been predefined may be given via its relation to known coordinates, such as `z=sqrt(2-d^2)`. The left-hand side is optional; if it is missing, a new variable name will be automatically supplied. If the specified variable is, in fact, already known, its usual name will be used. For example, `[r,cos(alpha),1-d^2]` is recognized to be identical to `raw`:

```
> raw&>ri;
  [r,cos(alpha),1-d^2]&>ri;
```

$$\left[r = \sqrt{r1^2 + r2^2},\ a = \frac{r1^2 - r2^2}{r1^2 + r2^2},\ w = \frac{r1^2 - r12^2 + r2^2}{r1^2 + r2^2}\right]$$
$$\left[r = \sqrt{r1^2 + r2^2},\ a = \frac{r1^2 - r2^2}{r1^2 + r2^2},\ w = \frac{r1^2 - r12^2 + r2^2}{r1^2 + r2^2}\right]$$

Objects of `type/coordsys` can be used in a number of ways explained in detail on the following pages, such as producing the equations necessary to change from one coordinate system to another (using the operator `&>`), controlling the coordinates in which the Laplacian (`L(coordsys)`), potential energy operator (`VOP(coordsys)`) or hyperspherical harmonics (`HH(coordsys)`) are represented, and building corresponding recurrence relations in the given coordinates (`RecRel(coordsys)`). Note that for a change of coordinates in differential equations using predefined coordinate systems, the procedure `Dchange(coordsys1 &> coordsys2,PDE)` should be used instead of `PDEtools:-dchange` in order to avoid a number of potential problems.

A.2.2 Transformation equations between coordinate systems: `coordsys1 &> coordsys2`

Parameters

`coordsys1`: (optional) Origin coordinate system. The default value is `ri` $= \{r_1, r_2, r_{12}\}$.

`coordsys2`: Target coordinate system.

Description

The operator `&>` provides a list of equations transforming one coordinate system into another. The coordinate systems should be specified as objects of `type/coordsys` as described above :

```
> &>rad;
```

$$\left[r1 = \frac{1}{2}\sqrt{2}\, r\sqrt{1+a},\ r2 = \frac{1}{2}\sqrt{2}\, r\sqrt{1-a},\ r12 = r\, d\right]$$

```
> rad&>ri;
```

$$\left[r = \sqrt{r1^2 + r2^2},\ a = \frac{r1^2 - r2^2}{r1^2 + r2^2},\ d = \frac{r12}{\sqrt{r1^2 + r2^2}}\right]$$

```
> raw&>[r,alpha,theta];
```

$$[r = r,\ a = \cos(\alpha),\ w = \sin(\alpha)\cos(\theta)]$$

A.2.3 Coordinate change: Dchange(e,t,options)

Parameters

e: Set or list of equations corresponding to the transformation from the old variables on the left hand side of the equations to the new variables on the right hand side.

t: Algebraic expression or procedure (program).

opts: All options work exactly like for the PDEtools:-dchange command.

Description

Since the coordinate system operator &> occasionally returns equations which contain the same variable names on the LHS and RHS, such as

```
> raw&>[r,a,theta];
```

$$\left[r = r,\ a = a,\ w = \sqrt{1 - a^2}\,\cos(\theta)\right]$$

the procedure PDEtools:-dchange provided by maple for variable change is not entirely compatible with &>. The Dchange command is in extension of PDEtools:-dchange in that it also accepts lists as the first argument, and accepts transformation equations with variables of the same name on both sides, which makes it compatible with results of &>. Furthermore, Dchange already collects in differentials and integrals and simplifies terms typically appearing when dealing with the $a = \cos(\alpha)$ coordinate.

```
> proc (f) options operator, arrow; 2*a*(d-1)*(d+1)*(diff(diff(f, a), d))/(r^2*d); end proc;
  Dchange(rad&>raw,%,simplify);
```

$$f \to \frac{2\,a\,(d-1)\,(d+1)\left(\frac{\partial}{\partial d}\left(\frac{\partial}{\partial a} f\right)\right)}{r^2\,d}$$

$$f \to \frac{4\,a\,w\left(\frac{\partial}{\partial w}\left(\frac{\partial}{\partial a} f\right)\right)}{r^2}$$

```
> 2*a*(d-1)*(d+1)*(diff(f(r, a, d), d, a))/(r^2*d);
  Dchange(rad&>ralphatheta,%,factor@simplify);
```

$$\frac{2\,a\,(d-1)\,(d+1)\left(\frac{\partial^2}{\partial d\,\partial a} f(r, a, d)\right)}{r^2\,d}$$

$$-\frac{4\cos(\alpha)^2\cos(\theta)\left(\frac{\partial}{\partial \theta} f(\alpha, r, \theta)\right)}{r^2\sin(\alpha)^2\sin(\theta)^3} + \frac{4\cos(\alpha)\cos(\theta)\left(\frac{\partial^2}{\partial \theta\,\partial \alpha} f(\alpha, r, \theta)\right)}{r^2\sin(\theta)\sin(\alpha)} + \frac{4\cos(\alpha)^2\cos(\theta)^2\left(\frac{\partial^2}{\partial \theta^2} f(\alpha, r, \theta)\right)}{r^2\sin(\theta)^2\sin(\alpha)^2}$$

A.2.4 Turn assumptions on coordinate ranges on or off: assumecoords(state)

Parameters

state: Either on or off.

Description

The command assumecoordranges(on) results in worksheet-wide assumptions (using the assume facility) on the physical ranges of the frequently used coordinates r_1, r_2, r_{12}, r_z, r, a, d, z, w, v, α, θ, s, t, u. The command assumecoordranges(off) removes these assumptions again. Note that it is possible to switch off the tilde with which maple usually marks assumed variables (e.g. x~) by selecting Tools→Options→Display→Assumed variables: "No Annotation". The following examples demonstrate the kinds of simplifications that are possible through these assumptions; in each case, the result without and with assumptions is displayed.

```
> assumecoords(off):
  expand(ln((r1+r2-r12)/(r1+r2+r12)));
  assumecoords(on):
  expand(ln((r1+r2-r12)/(r1+r2+r12)));
```

$$\ln\left(\frac{r1+r2-r12}{r1+r2+r12}\right)$$

$$\ln(r1+r2-r12)-\ln(r1+r2+r12)$$

```
> assumecoords(off):
  simplify(combine(sqrt(1+a)*sqrt(1-a)));
  assumecoords(on):
  simplify(combine(sqrt(1+a)*sqrt(1-a)));
```

$$\sqrt{a+1}\,\sqrt{1-a}$$

$$\sqrt{-a^2+1}$$

Do however proceed with caution, it appears that active assumptions can cause some routines (such as RecRel) to fail.

A.2.5 Turn assumptions on indices on or off: assumeinds(state)

Parameters

state: Either on or off.

Description

The command assumeinds(on) triggers the assumption that the typical index names i, j, k, l, m and n are of type/nonnegint. These assumptions can be turned off again by assumeinds(off). Only affects

names that are not currently assigned a value.

A.2.6 Plot functions in $\{\alpha, \theta\}$ coordinates: `ratplot(f,{alpha,theta,options})`

The procedure `ratplot` changes coordinates of the function `f` to $\{r, \alpha, \theta\}$ coordinates, evaluates at $r = 1$, $Z = 2$ and produces a 3D plot with respect to α and θ.

Parameters

`f`: function or list of functions in $\{r_1, r_2, r_{12}\}$, $\{r, a, d\}$, $\{r, a, w\}$ or $\{s, t, u\}$ coordinates

`alpha`: (optional) Plot range of α. Default value is `alpha=0.001..Pi-0.001`.

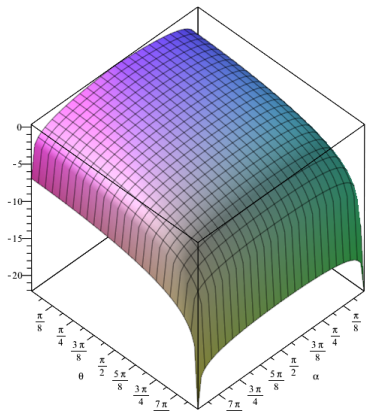
`theta`: (optional) Plot range of θ. Default value is `theta=0.001..Pi-0.001`.

`options`: (optional) Any additional keywords that can be applied to `plot3d`.

Description

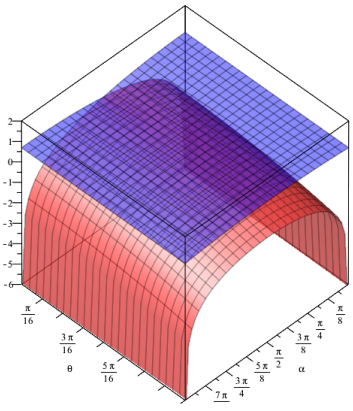
`ratplot` can be applied to quickly plot a function without having to manually change coordinates or entering the typical parameters for `plot3d`:

```
> ratplot(ln(r1+r2-r12));
```



However, additional parameters can easily be supplied:

```
> ratplot([ln(r1+r2-r12),ln(r1+r2+r12)],theta=0..Pi/2,view=-6..2,transparency=0.5,
  lightmodel=light4,color=[orange,blue]);
```



A.2.7 Laplacian: L(coordsys)(term)

Parameters

coordsys: (optional) Coordinate system given as type/coordsys. The default value is ri= $\{r_1, r_2, r_{12}\}$.

Description

The L command returns the Laplacian in the given coordinate system as a procedure, which may then directly be applied to some term:

```
> L();
```

$$f\to -\frac{1}{2}\frac{\partial}{\partial r1}\left(\frac{\partial}{\partial r1}f\right)-\frac{1}{2}\frac{\left(r1^2-r2^2+r12^2\right)\left(\frac{\partial}{\partial r12}\left(\frac{\partial}{\partial r1}f\right)\right)}{r1\,r12}-\left(\frac{\partial}{\partial r12}\left(\frac{\partial}{\partial r12}f\right)\right)$$
$$+\frac{1}{2}\frac{\left(-r2^2+r1^2-r12^2\right)\left(\frac{\partial}{\partial r2}\left(\frac{\partial}{\partial r12}f\right)\right)}{r2\,r12}-\frac{2\left(\frac{\partial}{\partial r12}f\right)}{r12}-\frac{\frac{\partial}{\partial r1}f}{r1}-\frac{1}{2}\frac{\partial}{\partial r2}\left(\frac{\partial}{\partial r2}f\right)-\frac{\frac{\partial}{\partial r2}f}{r2}$$

```
> L(raw);
```

$$f\to\frac{2\,(1+a)\,(-1+a)\left(\frac{\partial}{\partial a}\left(\frac{\partial}{\partial a}f\right)\right)}{r^2}+\frac{4\,w\,a\left(\frac{\partial}{\partial w}\left(\frac{\partial}{\partial a}f\right)\right)}{r^2}-\frac{1}{2}\frac{\partial}{\partial r}\left(\frac{\partial}{\partial r}f\right)$$
$$+\frac{2\,(-1+w)\,(w+1)\left(\frac{\partial}{\partial w}\left(\frac{\partial}{\partial w}f\right)\right)}{r^2}+\frac{6\,a\left(\frac{\partial}{\partial a}f\right)}{r^2}-\frac{5}{2}\frac{\frac{\partial}{\partial r}f}{r}+\frac{6\,w\left(\frac{\partial}{\partial w}f\right)}{r^2}$$

```
> L([r,a,d])(r^2*f(a,d));
```

$$6\,a\left(\frac{\partial}{\partial a}f(a,d)\right)+\frac{1}{2}\frac{\left(5\,d^2-4\right)\left(\frac{\partial}{\partial d}f(a,d)\right)}{d}-6f(a,d)+2\,(1+a)\,(-1+a)\left(\frac{\partial^2}{\partial a^2}f(a,d)\right)$$
$$+\frac{2\,a\,(d-1)\,(d+1)\left(\frac{\partial^2}{\partial d\,\partial a}f(a,d)\right)}{d}+\frac{1}{2}\left(-2+d^2\right)\left(\frac{\partial^2}{\partial d^2}f(a,d)\right)$$

A.2.8 Helium potential: VOP(coordsys),VOP0(coordsys)

Parameters

coordsys: (optional) Coordinate system given as type/coordsys. The default value is ri= $\{r_1, r_2, r_{12}\}$.

Description

These commands are simple shortcuts for the Coulomb potential energy of two interacting (VOP) or non-interacting (VOP0) particles with a charge of $-Y$ and one particle with a charge of Z:

```
> VOP();
```

$$-\frac{Z}{r1}-\frac{Z}{r2}+\frac{Y}{r12}$$

```
> VOP0(ralphatheta);
```

$$-\frac{2\,Z}{r\sqrt{2+2\cos(\alpha)}}-\frac{2\,Z}{r\sqrt{2-2\cos(\alpha)}}$$

A.2.9 Hyperspherical harmonics Y_l^n: HH(coordsys)(n,l)

Parameters

coordsys: (optional) Coordinate system given as type/coordsys. The default value is ri= $\{r_1, r_2, r_{12}\}$.

Description

HH(coordsys) returns the 3D hyperspherical harmonic Y_l^n in the given coordinate system as a procedure expecting the parameters n and l.

```
> HH(rad)(n,l);
```

$$r^n\left(1-a^2\right)^{\frac{1}{2}l}\,\text{GegenbauerC}\left(\frac{1}{2}\,n-l,\,l+1,\,a\right)\text{LegendreP}\left(l,\,-\frac{-1+a^2}{\sqrt{1-a^2}}\right)$$

```
> HH()(4,1);
```

$$4\left(r1^2-r2^2\right)\left(r1^2-r12^2+r2^2\right)$$

```
> HH({r,alpha,theta})(2,0);
```

$$2\,r^2\cos(\alpha)$$

A.2.10 Deriving Recurrence Relations: RecRel(coordsys)(ansatz)

Parameters

coordsys: Coordinate system specified as type/coordsys.

ansatz: Term specifying a power series ansatz for f in $\hat{L}(f) = \rho$ as explained below.

Description

The problem $\hat{L}(f) = \rho$ can be reformulated in any given coordinate system $\{\nu_1, \nu_2, \nu_3\}$ by expressing the function ρ as a power series in the ν_i, and accordingly making an ansatz for f:

$$\hat{L}\left(\sum_{i_1}\sum_{i_2}\sum_{i_3} c_{i_1,i_2,i_3} \cdot \nu_1^{i_1}\nu_2^{i_2}\nu_3^{i_3}\right) = \sum_{i_1}\sum_{i_2}\sum_{i_3} \rho_{i_1,i_2,i_3} \cdot \nu_1^{i_1}\nu_2^{i_2}\nu_3^{i_3}. \tag{A.1}$$

Evaluating the Laplacian and collecting in equal powers of the coordinates ν_i, one arrives at some recurrence relation. This recurrence relation can be obtained through the procedure RecRel(coordsys)(ansatz) where coordsys specifies the coordinate system $\{\nu_1, \nu_2, \nu_3\}$ as an object of type type/coordsys, and ansatz is the argument $c_{i_1,i_2,i_3} \cdot \nu_1^{i_1}\nu_2^{i_2}\nu_3^{i_3}$ of the left hand side sum in Eq. (A.1):

```
> RecRel(rad)(r^h*c[n,k]*a^n*d^k);
```

$$-\frac{1}{2}c_{n,k}(h+4+k+2n)(h-k-2n) - c_{n,k+2}(k+2)(k+3+2n) - 2c_{n+2,k}(n+2)(n+1) = Coeff(\rho, r^{h-2}a^n d^k)$$

```
> RecRel(rad)(r^3*c[n,k]*a^(2*n)*d^(2*k+1));
```

$$2c_{n,k}(k+2n+4)(k+2n-1) - 2c_{n,k+1}(2k+3)(k+2+2n) - 4c_{n+1,k}(n+1)(1+2n) = Coeff(\rho, r\,a^{2n}d^{2k+1})$$

```
> RecRel(ri)(c[i,j,k]*r1^i*r2^j*r12^k);
```

$$\frac{1}{2}c_{i-2,j+2,k+2}(j+2)(k+2) - \frac{1}{2}c_{i,j+2,k}(j+2)(3+j+k) - \frac{1}{2}c_{i,j,k+2}(k+2)(2k+6+i+j) - \frac{1}{2}c_{i+2,j,k}(i+2)(k+i+3) + \frac{1}{2}c_{i+2,j-2,k+2}(i+2)(k+2) = Coeff(\rho, r1^i r12^k r2^j)$$

As shown in the examples above, making an ansatz for just the even or odd powers of some variable is possible by simply adjusting the exponents.

A.2.11 Propagating coefficients w.r.t. one index: nextcoeff(RR,i)(inicoeff)

Parameters

RR: Recurrence relation relating coefficients $c_{n,k}$, $c_{n+1,k}$, $c_{n,k+1}$.

i: Name of index indicating propagation direction.

inicoeff: Term specifying a closed-form coefficient for a given value of i.

Description

Assume one has a three-term recurrence relation between coefficients $c_{n,k}$, $c_{n+1,k}$, $c_{n,k+1}$, and a closed-form expression inicoeff of the coefficients for some given value of n or k. For example, consider the recurrence relation in $\{r, a, d\}$ coordinates

```
> RR:=lhs(RecRel(rad)(c[n,k]*a^(2*n)*d^(2*k)*r^2))=0;
```

$$RR := 2\,c_{n,k}\,(k+2n+3)\,(k-1+2n) - 2\,c_{n,k+1}\,(k+1)\,(2k+3+4n) - 4\,c_{n+1,k}\,(n+1)\,(1+2n) = 0$$

and the initial coefficients at $k_0 = 0$,

```
> inicoeff:=c[n,1] = pochhammer(1/2,n)/n!;
```

$$inicoeff := c_{n,1} = \frac{\text{pochhammer}\left(\frac{1}{2}, n\right)}{n!}$$

In this situation, the command nextcoeff(RR,k) returns a procedure which calculates the closed-form of the next row of coefficients at $k = k_0 + 1$:

```
> nc:=nextcoeff(RR,k):
  nc(inicoeff);
```

$$c_{n,2} = \frac{1}{2}\,\frac{(4n-1)\,\text{pochhammer}\left(\frac{1}{2}, n\right)}{(4n+5)\,n!}$$

The following example uses the nextcoeff procedure for calculating the coefficients of $\hat{L}^{-1}\left(\frac{r_1 r_2}{r_{12}}\right)$:

```
> RecRel(rad)(c[n,k]*a^(2*n)*d^(2*k+1)*r^3): # Calculating the recurrence relation
  RR:=lhs(%) = -Physics[KroneckerDelta][-1, k]/2*pochhammer(1/2, n)/(factorial(n)*(-1+2*n)):
  nc:=nextcoeff(RR,k):                       # Procedure to calculate the next (wrt k) coefficient
  c[n,-1] = 0;                               # Iteratively solve the recurrence relation
  for ii from 0 to 4 do nc(%); end do;
```

$$c_{n,-1} = 0$$

$$c_{n,0} = \frac{1}{4}\,\frac{\text{pochhammer}\left(\frac{1}{2}, n\right)}{n!\,(-1+2n)\,(1+2n)}$$

$$c_{n,1} = \frac{1}{24}\,\frac{(10n+11)\,\text{pochhammer}\left(\frac{1}{2}, n\right)}{(3+2n)\,(n+1)\,(1+2n)\,n!}$$

$$c_{n,2} = \frac{1}{120}\,\frac{\left(140n^3+448n^2+319n-7\right)\,\text{pochhammer}\left(\frac{1}{2}, n\right)}{(n+1)\,(n+2)\,(5+2n)\,(1+2n)\,(3+2n)\,n!}$$

$$c_{n,3} = \frac{3}{560}\,\frac{\left(280n^4+1764n^3+3398n^2+1895n-99\right)\,\text{pochhammer}\left(\frac{1}{2}, n\right)}{(n+1)\,(7+2n)\,(n+3)\,(3+2n)\,(5+2n)\,(n+2)\,n!}$$

$$c_{n,4} = \frac{1}{1680}\,\frac{\left(6160n^5+64064n^4+233288n^3+343992n^2+159853n-14322\right)\,\text{pochhammer}\left(\frac{1}{2}, n\right)}{(7+2n)\,(n+4)\,(9+2n)\,(n+3)\,(3+2n)\,(5+2n)\,(n+2)\,n!}$$

A.2.12 Inverting the Laplacian on polynomials: `invertpolynomial(p,{h},{nonsingular})`

Parameters

`p`: Polynomial in $\{r_1, r_2, r_{12}\}$ coordinates.

`h`: Integer specifying the homogeneity if `p` $= 0$.

`nonsingular`: Keyword requesting the non-singular solution (rather than the version based on arctanh functions)

Description

For any polynomial `p` consisting of terms $c \cdot r_1^i r_2^j r_{12}^k$, `invertpolynomial(p)` returns $\hat{L}^{-1}(p)$, as far as possible, i.e. for all terms in which i, j and k are not all odd. If `p` is (anti)symmetric, then so is the result of `invertpolynomial(p)`:

```
> invertpolynomial(r1*r2);
```

$$\frac{1}{180}\,\frac{(r1^2+r2^2)\,(r2^4-16\,r1^2\,r2^2+r1^4)}{r1\,r2}$$

```
> invertpolynomial(r1+r2+r12);
```

$$-\frac{1}{12}\,r12^3-\frac{1}{6}\,r1^3-\frac{1}{6}\,r2^3$$

```
> invertpolynomial((r1-r2)*r12^2);
```

$$\frac{1}{90}\,(r1-r2)\,\left(7\,r1^4+7\,r1^3\,r2-10\,r1^2\,r12^2+2\,r1^2\,r2^2+7\,r1\,r2^3-10\,r1\,r12^2\,r2+7\,r2^4-10\,r12^2\,r2^2\right)$$

```
> invertpolynomial((r1+r2)/r12);
```

$$\left(-\frac{1}{3}\,r1^2+\frac{1}{3}\,r12^2-\frac{1}{3}\,r2^2\right)\,\mathrm{arctanh}\left(\frac{r12}{r1+r2}\right)-\frac{1}{3}\,(r1+r2)\,r12$$

The parameter `p` may also be zero, in which case the second parameter `h` needs to be given in order to specify the homogeneity:

```
> invertpolynomial(0,2);
```

$$2\,c_0\,(r1-r2)\,(r1+r2)+c_1\,\left(r1^2-r12^2+r2^2\right)$$

Using this procedure together with the other shortcuts provided by the `GiosTools` library, (the singular version of) Ψ_2 can be calculated very quickly:

```
> eval(eval(SERR,n=0),[Psi[-1]=0,Psi[-2]=0]);
  op(1,lhs(%))=eval(invertpolynomial(rhs(%),0),c=1);
```

$$T\left(\Psi_0\right) = 0$$

$$\Psi_0 = 1$$

```
> eval(eval(eval(SERR,n=1),[Psi[-1]=0]),[V=VOP(),%]);
  op(1,lhs(%))=invertpolynomial(rhs(%));
```

$$T\left(\Psi_1\right) = \frac{Z}{r1} + \frac{Z}{r2} - \frac{Y}{r12}$$

$$\Psi_1 = \frac{1}{2}\, Y\, r12 + (-r1 - r2)\, Z$$

```
> eval(lhs(SERR),n=2)=eval(eval(rhs(SERR),n=2),[V=VOP(),Psi[0] = 1,%]);
  op(1,lhs(%))=collect(invertpolynomial(rhs(%)),[E,Z,Y,arctanh],factor);
```

$$T\left(\Psi_2\right) = -\left(-\frac{Z}{r1} - \frac{Z}{r2} + \frac{Y}{r12}\right)\left(\frac{1}{2}\, Y\, r12 + (-r1 - r2)\, Z\right) + E$$

$$\Psi_2 = \left(-\frac{1}{6}\, r1^2 - \frac{1}{6}\, r2^2\right) E + \left(\frac{1}{12}\, r1^2 + \frac{1}{12}\, r2^2\right) Y^2 + \left(\left(\frac{1}{3}\, r12^2 - \frac{1}{3}\, r1^2 - \frac{1}{3}\, r2^2\right) \operatorname{arctanh}\left(\frac{r12}{r1 + r2}\right)\right.$$
$$\left. + \frac{1}{3}\, (r1 - r2)\, (r1 + r2) \operatorname{arctanh}\left(\frac{r1 - r2}{r12}\right) - \frac{2}{3}\, (r1 + r2)\, r12\right) Z\, Y + \left(r1\, r2 + \frac{1}{3}\, r1^2 + \frac{1}{3}\, r2^2\right) Z^2$$

In order to obtain the non-singular solution, use the keyword `nonsingular`:

```
> (r1+r2)/r12;
  invertpolynomial(%,nonsingular);
```

$$\frac{r1 + r2}{r12}$$

$$-2\left(\frac{1}{6}\, r1^2 - \frac{1}{6}\, r12^2 + \frac{1}{6}\, r2^2\right) \ln(r1 + r2 + r12) + \frac{1}{3}\, r1\, r2 - \frac{1}{3}\, r1\, r12 - \frac{1}{3}\, r2\, r12$$

The block B_- (which takes the place of $b_- = \operatorname{arctanh}\left(\frac{r_1 - r_2}{r_{12}}\right)$) as well as the corresponding tails are returned in $\{r, a, d\}$ coordinates, since they would be too clunky in interparticle coordinates:

```
> (r12/r1+r12/r2);
  invertpolynomial(%,nonsingular);
```

$$\frac{r12}{r1} + \frac{r12}{r2}$$

$$-\frac{1}{3}\, (r1 - r2)\, (r1 + r2)\left(\sum_{n=0}^{\infty}\sum_{k=0}^{\infty} \frac{\left(\frac{1}{2}\, a\right)^{2n+1} \Gamma\left(\frac{1}{2}\, k + 1 + 2\, n\right)\left(-\frac{1}{2}\sqrt{2}\, d\right)^k}{\Gamma\left(\frac{1}{2}\, k + \frac{3}{2} + n\right)\Gamma\left(n + \frac{3}{2}\right)}\right)$$
$$- \frac{4}{3}\, \frac{r^2\, d \arccos\left(\frac{1}{2}\sqrt{2}\, d\right)\sqrt{-d^2 + 2}}{\pi} + \frac{4}{3}\, \frac{r^2\left(d^2 - 1\right)\ln(r)}{\pi} - \frac{2}{3}\, r1\, r12 - \frac{2}{3}\, r2\, r12$$

A.2.13 Verifying identity of two terms: `compare(t1,t2,{s,v,onlyeven,onlyodd})`

Parameters

`t1`, `t2`: Terms to be compared.

`s`: (optional) Integer specifying the series expansion length.

`v`: (optional) List or set of variables to be expanded in.

`onlyeven=...`: (optional) List or set of variables of which only even powers should be considered.

`onlyodd=...`: (optional) List or set of variables of which only odd powers should be considered.

Description

Verifies whether the two terms `t1` and `t2` are identical by returning their difference in form of a series expansion. The series expansion is performed up to at least the power specified by `s`. If `v` is not given, series expansions are done in all variables found in `t1` and `t2`; otherwise, only the variables in `v` will be considered. For example, the following output verifies that the terms in the first and second line have identical series expansions up to power 20 of all three variables:

```
> ln((r1+r2-r12)/(r1+r2+r12))/2;
  -arctanh(r12/(r1+r2));
  compare(%%,%,20);
```

$$\frac{1}{2}\ln\left(\frac{r1+r2-r12}{r1+r2+r12}\right)$$
$$-\operatorname{arctanh}\left(\frac{r12}{r1+r2}\right)$$
$$O(r1^{20})\,O(r12^{20})\,O(r2^{20})$$

If two terms are not identical, a series expansion of their difference is returned:

```
> t1:=ln((r1+r2-r12));
  t2:=-arctanh(r12/(r1+r2));
  compare(t1,t2,8,[r12]);
```

$$t1 := \ln(r1+r2-r12)$$
$$t2 := -\operatorname{arctanh}\left(\frac{r12}{r1+r2}\right)$$
$$\ln(r1+r2)-\frac{1}{2}\frac{r12^2}{(r1+r2)^2}-\frac{1}{4}\frac{r12^4}{(r1+r2)^4}-\frac{1}{6}\frac{r12^6}{(r1+r2)^6}+O(r12^8)$$

One may also choose to only compare even or odd powers of a certain variable. For the terms in the previous example, this results in confirmation that their odd powers are indeed identical:

```
> compare(t1,t2,10,[r12],onlyodd={r12});
  Considering only odd powers of: , r12,   Warning: csgn and signum functions will be ignored.
```

$$O(r12^{10})$$

A.2.14 Translating maple to LaTeX math code: getTeX(t,{clip,cdots})

Parameters

t: Expression to be translated.

clip=true/false: (optional) (Linux only) Boolean, triggers copying of the result to the clipboard.

cdots=true/false: (optional) Boolean, triggers the use of \cdots between products. Default is false.

Description

The procedure getTeX(t) (or equivalently gettex(t)) translates a maple expression t into LaTeX code. When used with "red" (non-classic) Maple, the output can be directly copied into a LaTeX file. (Classic Maple treats the backslashes and quotes in the output differently and therefore makes post-processing (in form of removal of superfluous characters) necessary.) Maple objects that can be translated include sums, integrals, derivatives, hypergeometric and other special functions and Greek letters. For example, the output

```
> hypergeom([1/2,3/2],[2],a^2):
  %=convert(%,sum,a,n);
  getTeX(%);
```

$$\text{hypergeom}\left(\left[\frac{1}{2}, \frac{3}{2}\right], [2], a^2\right) = \sum_{n=0}^{\infty} \frac{\text{pochhammer}\left(\frac{1}{2}, n\right) \text{pochhammer}\left(\frac{3}{2}, n\right) a^{2n}}{\text{pochhammer}(2, n)\, n!}$$

```
\pFq{2}{1}{{1 \over 2},{3 \over 2}}{2}{{ a ^ {2}}}=\sum _{n = 0} ^ {\infty} {\left( {1 \over 2} \right) _ {n} \left( {3 \over 2}
    \right) _ {n}  a ^ {2 n} \over \left( 2 \right) _ {n}  n! }
```

can be copied directly into LaTeX to obtain

$$ {}_2\mathrm{F}_1\left[\begin{matrix}\frac{1}{2}, \frac{3}{2} \\ 2\end{matrix}; a^2\right] = \sum_{i=0}^{\infty} \frac{\left(\frac{1}{2}\right)_i \left(\frac{3}{2}\right)_i a^{2i}}{(2)_i\, i!}, $$

and

```
> 2*(1-cos(alpha)^2)^2*(1-cos(theta)^2)*(diff(f(alpha, theta), alpha, alpha))/(sin
  (theta)^2*sin(alpha)^4);
  gettex(%,cdots);
```

$$\frac{2\left(1-\cos(\alpha)^2\right)^2\left(1-\cos(\theta)^2\right)\left(\frac{\partial^2}{\partial\alpha^2} f(\alpha, \theta)\right)}{\sin(\theta)^2 \sin(\alpha)^4}$$

```
{2 (1 -  \text{cos} \left( \alpha \right)  ^ {2} ) ^ {2} \cdot \left(1 -  \text{cos} \left( \theta \right)  ^ {2} \right) \over  \text{sin}
    \left( \theta \right)  ^ {2} \cdot  \text{sin} \left( \alpha \right)  ^ {4}} \cdot {\partial ^ 2 \over  \partial \alpha ^ 2} f \left(
    \alpha,\theta \right)
```

translates to the formula (this time using \cdots)

$$\frac{2(1-\cos(\alpha)^2)^2 \cdot \left(1-\cos(\theta)^2\right)}{\sin(\theta)^2 \cdot \sin(\alpha)^4} \cdot \frac{\partial^2}{\partial\alpha^2} \mathrm{f}(\alpha, \theta)\,.$$

A.2.15 Identifying sequences of numbers: IdSeq(seq,{ind})

Parameters

seq: List of numbers.

ind: (optional) Index name or equation of the form index=integer.

Description

The procedure IdSeq (equivalently idseq) identifies a closed-form formula which produces a sequence seq (given as a list):

```
> n!*pochhammer(1/2,n)/pochhammer(3/4,n);
  [seq(%,n=1..10)];
  IdSeq(%);
```

$$\frac{n!\ \mathrm{pochhammer}\left(\frac{1}{2}, n\right)}{\mathrm{pochhammer}\left(\frac{3}{4}, n\right)}$$

$$\left[\frac{2}{3}, \frac{8}{7}, \frac{240}{77}, \frac{128}{11}, \frac{11520}{209}, \frac{138240}{437}, \frac{931840}{437}, \frac{223641600}{13547}, \frac{1955266560}{13547}, \frac{1002700800}{713}\right]$$

$$\frac{\mathrm{pochhammer}\left(\frac{1}{2}, n\right) \mathrm{pochhammer}(1, n)}{\mathrm{pochhammer}\left(\frac{3}{4}, n\right)}$$

The running index used to express this formula can be given as the second argument ind, and will usually be assumed to start from 1. If the first sequence element corresponds to an index other than 1, one may specify the starting index in the form index=integer:

```
> (-1)^n*GAMMA(2*n);
  [seq(%,n=3..10)];
  IdSeq(%,i=3);
```

$$(-1)^n \Gamma(2n)$$

$$[-120, 5040, -362880, 39916800, -6227020800, 1307674368000, -355687428096000, 121645100408832000]$$

$$\frac{1}{2} 4^i \Gamma(i)\ (-1)^i \mathrm{pochhammer}\left(\frac{1}{2}, i\right)$$

Sequence identification proceeds more smoothly the more sequence elements are given. Depending on the sequence, 10-15 elements give a good chance of success.

A.2.16 List storage tools: app2seq(ele,{i}), setseq(seq,{i}), resetseq({i}), getseq({i})

Parameters

i: (optional) Integer from 1 to 100, selecting the sequence to be manipulated. Default is 1.

ele: Number or list of numbers.

seq: List of numbers.

Description

Often one wants to progressively build lists. One such case would be an iterative process which, one by one, produces elements of some sequence, which one wants to store in a list for later identification using the IdSeq procedure. In order to avoid having to manually define, initialize and fill and keep track of lists, the GiosTools library provides 100 pre-defined lists, which can be accessed and manipulated through the procedures presented here. In each case, i specifies the number of the target list; the default is 1. The procedure app2seq(ele,{i}) appends the element(s) ele to the specified list. Alternatively, setseq(seq,{i}) sets the entire specified list to seq, forgetting all previously contained elements. A list can be reset completely by resetseq({i}). In order to simply output a list, use getseq({i}). The following example illustrates the usage of these list storage tools at the recursive calculation of the binomial coefficients $\binom{n}{1}$, $\binom{n}{2}$, $\binom{n}{3}$ and $\binom{n}{4}$ for $n = 1..10$, together with a confirmation via IdSeq:

```
> setseq([1],1):
  setseq([1],2):
  setseq([1],3):
  setseq([1],4):
  for i from 1 to 10 do
    app2seq(1,1);
    app2seq(getseq(1)[-1]+getseq(2)[-1],2);
    app2seq(getseq(2)[-1]+getseq(3)[-1],3);
    app2seq(getseq(3)[-1]+getseq(4)[-1],4);
  end do:
  getseq(1);
  getseq(2);
  getseq(3);
  getseq(4);
  IdSeq(%);
```

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

$$[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]$$

$$[1, 3, 6, 10, 15, 21, 28, 36, 45, 55, 66]$$

$$[1, 4, 10, 20, 35, 56, 84, 120, 165, 220, 286]$$

$$\frac{1}{6} n\,(n+2)\,(n+1)$$

A.2.17 Identifying series in one variable: IdSer(ser,{ind})

Parameters

ser: Series expansion in one variable.

ind: (optional) Index name.

Description

The procedure IdSer (or equivalently idser) is a version of IdSeq which takes as input a series expansion ser with numerical coefficients, identifies the sequence of coefficients, and returns a closed form for ser in form of an infinite summation:

```
> a+(1/6)*a^3+(3/40)*a^5+(5/112)*a^7+(35/1152)*a^9+(63/2816)*a^11+(231/13312)*a^13+
  (143/10240)*a^15+(6435/557056)*a^17+O(a^19);
  IdSer(%);
  simplify(value(%));
```

$$a+\frac{1}{6}a^3+\frac{3}{40}a^5+\frac{5}{112}a^7+\frac{35}{1152}a^9+\frac{63}{2816}a^{11}+\frac{231}{13312}a^{13}+\frac{143}{10240}a^{15}+\frac{6435}{557056}a^{17}+O\left(a^{19}\right)$$

$$\sum_{n=0}^{\infty}\frac{\text{pochhammer}\left(\frac{1}{2},n\right)a^{1+2n}}{(1+2n)\,n!}$$

$$\arcsin(a)$$

One possible application of this feature is performing simplifications by expanding some term into a series, followed by identification and resummation, as in the following simple example:

```
> sqrt(1+sqrt(1-a^2));
  series(%,a,12);
  IdSer(%);
  simplify(value(%));
```

$$\sqrt{1+\sqrt{1-a^2}}$$

$$\sqrt{2}-\frac{1}{8}\sqrt{2}\,a^2-\frac{5}{128}\sqrt{2}\,a^4-\frac{21}{1024}\sqrt{2}\,a^6-\frac{429}{32768}\sqrt{2}\,a^8-\frac{2431}{262144}\sqrt{2}\,a^{10}+O\left(a^{12}\right)$$

$$\sum_{n=0}^{\infty}\frac{\sqrt{2}\,\text{pochhammer}\left(\frac{1}{4},n\right)\text{pochhammer}\left(-\frac{1}{4},n\right)a^{2n}}{n!\,\text{pochhammer}\left(\frac{1}{2},n\right)}$$

$$\frac{1}{2}\sqrt{2}\left(\sqrt{1+a}+\sqrt{1-a}\right)$$

A.2.18 Converting expressions to hypergeometric functions: `convert(t,hyper)`

Parameters

`t`: Term to be converted into a hypergeometric function.

Description

This procedure is a shorthand for the command `convert(convert(expr,hypergeom,include=all, exclude=powers),hypergeom,include=radicals)`, which converts any applicable expression (except for integer powers) into a hypergeometric function:

```
> arcsin(a)=convert(arcsin(a),hyper);
```

$$\arcsin(a) = a\,\mathrm{hypergeom}\left(\left[\frac{1}{2},\frac{1}{2}\right],\left[\frac{3}{2}\right],a^2\right)$$

```
> 1/sqrt(1-a^2)=convert(1/sqrt(1-a^2),hyper);
```

$$\frac{1}{\sqrt{1-a^2}} = \mathrm{hypergeom}\left(\left[\frac{1}{2}\right],[\,],a^2\right)$$

```
> Sum(pochhammer(3/4,n)*pochhammer(3/4,n)/pochhammer(1/2,n)/n!
  *a^(2*n),n=0..infinity):
  %=convert(%,hyper);
```

$$\sum_{n=0}^{\infty} \frac{\mathrm{pochhammer}\left(\frac{3}{4},n\right)^2 a^{2n}}{\mathrm{pochhammer}\left(\frac{1}{2},n\right) n!} = \mathrm{hypergeom}\left(\left[\frac{3}{4},\frac{3}{4}\right],\left[\frac{1}{2}\right],a^2\right)$$

A.2.19 Converting expressions to power series: convert(t,sum,{v:=auto,i,options})

Parameters

t: Term to be converted into a power series expansion.

v: (optional) Variable to be expanded in. If one wishes to specify i but not v, the keyword auto may be used here.

i: (optional) Summation index or list of summation indices to be used.

options: (optional) Sequence of options, accepts the same keywords as convert/FormalPowerSeries.

Description

Hypergeometric functions are directly rewritten as sums, preserving the pochhammers:

```
> hypergeom([1/2,3/2],[2],a^2);
  %=convert(%,sum);
  %%=convert(%%,sum,a,n);
```

$$\text{hypergeom}\left(\left[\frac{1}{2}, \frac{3}{2}\right], [2], a^2\right)$$

$$\text{hypergeom}\left(\left[\frac{1}{2}, \frac{3}{2}\right], [2], a^2\right) = \sum_{i=0}^{\infty} \frac{\text{pochhammer}\left(\frac{1}{2}, i\right) \text{pochhammer}\left(\frac{3}{2}, i\right) a^{2\,i}}{\text{pochhammer}(2, i)\; i!}$$

$$\text{hypergeom}\left(\left[\frac{1}{2}, \frac{3}{2}\right], [2], a^2\right) = \sum_{n=0}^{\infty} \frac{\text{pochhammer}\left(\frac{1}{2}, n\right) \text{pochhammer}\left(\frac{3}{2}, n\right) a^{2\,n}}{\text{pochhammer}(2, n)\; n!}$$

When applied to other functions, this procedure applies convert\FormalPowerSeries. If variable and/or index are not given, they are automatically determined:

```
> sin(x)=convert(sin(x),sum,auto,k);
```

$$\sin(x) = \sum_{k=0}^{\infty} \frac{(-1)^k x^{2\,k+1}}{(2\,k+1)!}$$

When applied to finite sums or products, the procedure first attempts to apply convert\FormalPowerSeries to the entire term; if this fails, it maps onto the summands and factors:

```
> (-1*x/2+x^3/6)*arctan(x)+(-x^2/4+1/12)*ln(x^2+1)+5*
  x^2/12+1/4;
  `   `=convert(%,sum);
```

$$\left(-\frac{1}{2}x+\frac{1}{6}x^3\right)\arctan(x)+\left(-\frac{1}{4}x^2+\frac{1}{12}\right)\ln\left(x^2+1\right)+\frac{5}{12}x^2+\frac{1}{4}$$

$$=\sum_{i=0}^{\infty}\frac{\cos\left(\frac{1}{2}\pi i\right)x^{i+4}}{(i+4)\,(i+3)\,(i+2)\,(i+1)}$$

```
> sin(x)*exp(y)+arctanh(z):
  %=convert(%,sum);
```

$$\sin(x)\,e^{y}+\operatorname{arctanh}(z)=\left(\sum_{i=0}^{\infty}\frac{y^i}{i!}\right)\left(\sum_{j=0}^{\infty}\frac{(-1)^j x^{2j+1}}{(2j+1)!}\right)+\sum_{i=0}^{\infty}\frac{z^{2i+1}}{2i+1}$$

Multiple summation indices may be given in form of a list:

```
> sin(x)*cos(y):
  %=convert(%,sum,auto,[k,l]);
```

$$\sin(x)\cos(y)=\left(\sum_{k=0}^{\infty}\frac{(-1)^k x^{2k+1}}{(2k+1)!}\right)\left(\sum_{l=0}^{\infty}\frac{(-1)^l y^{2l}}{(2l)!}\right)$$

The argument of the power series may be an expression containing several variables:

```
> arctanh((r1-r2)/r12):
  %=convert(%,sum,(r1-r2)/r12,k);
```

$$\operatorname{arctanh}\left(\frac{r1-r2}{r12}\right)=\sum_{k=0}^{\infty}\frac{\left(\frac{r1-r2}{r12}\right)^{2k+1}}{2k+1}$$

The variable to be expanded in, if not given (i.e. if the third parameter is left empty or equal to the keyword `auto`), is determined as follows. If the expression `t` only contains a single variable, this variable is used. If `t` is a function of a single argument, `t` is expanded in this argument. This still works if the argument is a more complicated expression containing several variables. In the remaining cases, a variable appearing in `t` is randomly chosen (with preference for variable names that are not typically indices) and a warning is displayed.

```
> r*exp(-r)=convert(r*exp(-r),sum);
```

$$r\,e^{-r}=\sum_{i=0}^{\infty}\frac{(-1)^i r^{i+1}}{i!}$$

```
> x*sin(y)=convert(x*sin(y),sum);
```

$$x\sin(y)=x\left(\sum_{j=0}^{\infty}\frac{(-1)^j y^{2j+1}}{(2j+1)!}\right)$$

```
> ln((r1+r2-r12)/(r1+r2+r12)):
  %=convert(symplify(convert(%,arctanh)),sum,auto,k);
```

$$\ln\left(\frac{r1+r2-r12}{r1+r2+r12}\right)=-2\left(\sum_{k=0}^{\infty}\frac{\left(\frac{r12}{r1+r2}\right)^{2k+1}}{2k+1}\right)$$

A.2.20 Collecting in sqrt and fractional powers: Collect(t,c,{form,func})

Parameters

t: Expression.

c: Indeterminate (including square roots), unevaluated function, keyword

sqrt for simple collection in all detected square roots,

ratpower≡rp for advanced collection in rational powers,

ratpower(base)≡rp(base) for advanced collection in rational powers of the given base,

ratpower(base,exp)≡rp(base,exp) for advanced collection in base$^{\text{exp}}$,

or a list or set thereof.

form: (optional) name; 'recursive' (default) or 'distributed'.

func: (optional) procedure.

Description

The standard maple procedure collect refuses to collect in square roots and other terms. In such cases, one may use Collect, which temporarily freezes the collectible(s) c. This allows collection in sqrt(expr), powers, and so on. The optional arguments func and form work exactly as in the standard collect. If sqrt is specified without an argument, all square roots (more precisely, all half-integer powers) in t are detected, frozen and collected in; priority is given to square roots of sums and products.

```
> sqrt(r1+r2)*r1^2+sqrt(r1+r2)*r2^2+sqrt(r1+r2)*r12^2+sqrt(r1-
  r2)*r1^2-sqrt(r1-r2)*r2^2;
  Collect(%,sqrt);
```

$$\sqrt{r1+r2}\,r1^2+\sqrt{r1+r2}\,r2^2+\sqrt{r1+r2}\,r12^2+\sqrt{r1-r2}\,r1^2-\sqrt{r1-r2}\,r2^2$$

$$\left(r1^2+r2^2+r12^2\right)\sqrt{r1+r2}+\left(r1^2-r2^2\right)\sqrt{r1-r2}$$

Of course, one needs to proceed with caution, since terms such as $\sqrt{a+b}$ and $\frac{a+b}{\sqrt{a+b}}$ are not recognized to be identical, which sometimes results in incomplete simplification:

```
> a/sqrt(a + b) - b/sqrt(a + b) + sqrt(a + b);
  Collect(%,sqrt);
```

$$\frac{a}{\sqrt{a+b}}-\frac{b}{\sqrt{a+b}}+\sqrt{a+b}$$

$$\frac{a-b}{\sqrt{a+b}}+\sqrt{a+b}$$

The Collect procedure is also capable of identifying such situations and collecting in arbitrary rational powers while treating associated powers (such as $x^p, x^{p+1}, x^{p+2}, \ldots$) as a single object. This is done using the keyword ratpower({base,exp}) (rp for short). If ratpower is given without an argument, all rational powers of non-numerical bases in the term t are detected; and powers that have the same base and exponents differing by integers are grouped together. For each such group, the power with the

lowest exponent is frozen and collected in. For example, if `t` contains the square roots $\sqrt{a+b}$ and $\frac{1}{\sqrt{a+b}}$, $\sqrt{a+b}$ is represented as $\frac{a+b}{\sqrt{a+b}}$, and `t` is collected in $\frac{1}{\sqrt{a+b}}$:

```
> a/sqrt(a + b) - b/sqrt(a + b) + sqrt(a + b);
  Collect(%,ratpower);
```

$$\frac{a}{\sqrt{a+b}} - \frac{b}{\sqrt{a+b}} + \sqrt{a+b}$$

$$\frac{2\,a}{\sqrt{a+b}}$$

If `ratpower(base)` is given with one argument, the process described above will only be applied for powers of the specified `base`. If `ratpower(base,exp)` is given with two arguments, only powers of `base` with exponents differing from `exp` by an integer are collected. The detection of related rational powers also works for bases (take for example $1-a$) that are obscured inside products (such as in $\sqrt{a-1}$, $\sqrt{2-2a}$ or $\sqrt{1-a^2}$). This is achieved in the following way. If no base is given, all powers of factorizable terms $(x_1x_2\cdots)^p$ are rewritten as products of powers $x_1^p x_2^p \cdots$, and the numerical factors are made consistent (default behaviour is to make all trailing coefficients of the bases positive):

```
> (1-x)^(1/2)/sqrt(1+x)+(1-x^2)^(1/2)+sqrt(2*x-2)*sqrt(-2*x-2);
  Collect(%,rp,simplify);
```

$$\frac{\sqrt{1-x}}{\sqrt{1+x}} + \sqrt{-x^2+1} + \sqrt{-2-2\,x}\,\sqrt{2\,x-2}$$

$$-\frac{x\sqrt{1-x}}{\sqrt{1+x}}$$

If a base b is given, all powers $(xb)^p$ of bases divisible by b are split into powers of their factors, and the numerical factors are adjusted to agree with the given base:

```
> sqrt(1-a^2)+a*sqrt(2-2*a)*sqrt(2+2*a)+sqrt(1+a)*(1-a)^(3/2);
  Collect(%,rp(1-a),simplify);
  Collect(%%,rp(2-2*a),simplify);
```

$$\sqrt{-a^2+1} + a\sqrt{-2\,a+2}\,\sqrt{2+2\,a} + \sqrt{1+a}\,(1-a)^{3/2}$$

$$(a+2)\sqrt{1+a}\,\sqrt{1-a}$$

$$\frac{\sqrt{1+a}\,\sqrt{2}\,(a+2)\sqrt{-2\,a+2}}{2}$$

The procedures documented on the following pages deal with factorial-like terms, including pochhammer symbols, Γ functions, binomials, doublefactorials and of course factorials, and with general simplifications. All of these procedures take as their only argument(s) the terms to be acted upon, therefore the parameters will not be listed explicitly.

A.2.21 Conversions to Pochhammer: `convert(t,poch)`, `convert(t,pochfac)`

Converts a term `t` containing Γ functions, binomials, factorials, and doublefactorials to either pochhammer symbols (argument `poch`) or into expressions using both pochhammers and (wherever suitable) factorials (argument `pochfac`). For example:

```
> binomial(n,k)=convert(binomial(n,k),poch);
```

$$\mathrm{binomial}(n,k) = \frac{\mathrm{pochhammer}(-n,k)}{\mathrm{pochhammer}(1,k)\ (-1)^k}$$

```
> binomial(n,k)=convert(binomial(n,k),pochfac);
```

$$\mathrm{binomial}(n,k) = \frac{\mathrm{pochhammer}(-n,k)}{k!\ (-1)^k}$$

```
> GAMMA(2*n+3)=convert(GAMMA(2*n+3),poch);
```

$$\Gamma(2n+3) = 2\ \mathrm{pochhammer}(3, 2n)$$

```
> doublefactorial(2*n+3)=convert(doublefactorial(2*n+3),pochfac);
```

$$\mathrm{doublefactorial}(2n+3) = \frac{3}{4}\ 2^{n+2}\ \mathrm{pochhammer}\left(\frac{5}{2}, n\right)$$

```
> doublefactorial(2*n+4)=convert(doublefactorial(2*n+4),pochfac);
```

$$\mathrm{doublefactorial}(2n+4) = 2^{n+2}\ (n+2)!$$

A.2.22 Trimming pochhammers and factorials: `trimpoch(t)`, `trimfac(t)`, `trimpochfac(t)`

The procedure `trimpoch(t)` turns pochhammers of the form $(a)_{bn+c}$ into $(a+c)_{bn}$ by appending or detaching factors such that purely numerical terms c are removed from the second argument. Similarly, the procedure `trimfac(t)` turns $(an-b)!$ into $(an)!$ by attaching additional factors to remove negative numerical terms $-b$, which is often useful for making terms well-defined at $n = 0$. The procedure `trimpochfac(t)` does both at once. For example:

```
> pochhammer(5,k-3)=trimpoch(pochhammer(5,k-3));
```

$$\text{pochhammer}(5, k-3) = \frac{1}{24}\,\text{pochhammer}(2, k)$$

```
> pochhammer(x,y+3)=trimpoch(pochhammer(x,y+3));
```

$$\text{pochhammer}(x, y+3) = x\,(x+1)\,(x+2)\,\text{pochhammer}(x+3, y)$$

```
> pochhammer(n+2,2*k-2)=trimpoch(pochhammer(n+2,2*k-2));
```

$$\text{pochhammer}(n+2, 2k-2) = \frac{\text{pochhammer}(n, 2k)}{n\,(n+1)}$$

```
> (n-5)!=trimfac((n-5)!);
```

$$(n-5)! = \frac{n!}{(n-4)\,(n-3)\,(-2+n)\,(-1+n)\,n}$$

```
> (n-5)!*pochhammer(n-4,2*n+5)=trimpochfac((n-5)!*pochhammer(n-4,2*n+5));
```

$$(n-5)!\,\text{pochhammer}(n-4, 2n+5) = n!\,\text{pochhammer}(n+1, 2n)$$

A.2.23 Simplification of factorial-like terms: `gsimplify(t)`

The procedure `gsimplify(t)` is a shorthand for `simplify(simplify(t,GAMMA))`:

```
> n!*pochhammer(1/2, n)=gsimplify(n!*pochhammer(1/2, n));
```

$$n!\ \mathrm{pochhammer}\left(\frac{1}{2}, n\right) = 4^{-n}\,\Gamma(2n+1)$$

A.2.24 Changing one factorial-like term into another: `gchange(t1,t2)`

Often one wishes to replace one representation of some factorial-like term by another, for example $(k+1)_n \cdot \left(k+\frac{1}{2}\right)_n$ by $(2k)_{2n}$, but is unsure what extra factors have to be included. The procedure `gchange(t1,t2)` provides an equation expressing `t1` in terms of `t2`. For example:

```
> gchange(pochhammer(k+1,n)*pochhammer(k+1/2,n),pochhammer(2*k,2*n));
```

$$\mathrm{pochhammer}(1+k, n)\ \mathrm{pochhammer}\left(k+\frac{1}{2}, n\right) = \frac{4^{-n}\,(k+n)\ \mathrm{pochhammer}(2k, 2n)}{k}$$

A.2.25 Splitting factorial-like terms: `gsplit(t)`

This procedure finds the factorial-like term in the expression `t` which has the (multiplicity-wise) largest argument (e.g. $(2n)!$ in the example below), and splits it into two terms, one containing all even factors, and the other containing all odd factors (e.g. $(2n)! = 4^n n!\left(\frac{1}{2}\right)_n$):

```
> convert(sqrt(1-a^2),FormalPowerSeries,a,n):
  %=gsplit(%);
```

$$\sum_{n=0}^{\infty}\left(-\frac{4^{-n}\,(2n)!\,a^{2n}}{n!^2\,(2n-1)}\right) = \sum_{n=0}^{\infty}\left(-\frac{\mathrm{pochhammer}\left(\frac{1}{2}, n\right)a^{2n}}{n!\,(2n-1)}\right)$$

A.2.26 Symbolic simplification: `symplify(t)`

The procedure `symplify(t)` is a shorthand for `simplify(t,symbolic)`. It is e.g. safe to use with $\{r_1, r_2, r_{12}\}$ coordinates:

```
> sqrt((r1+r2)^2):
  %=symplify(%);
```

$$\sqrt{(r1+r2)^2} = r1+r2$$

```
> sqrt(r1^2+r2^2)*sqrt(2-2*sqrt(1-(r1^2-r2^2)^2/(r1^2+r2^2)^2)):
  simplify(%)=symplify(%);
```

$$\sqrt{r1^2+r2^2}\sqrt{2-4\sqrt{\frac{r1^2\,r2^2}{\left(r1^2+r2^2\right)^2}}} = \sqrt{2}\,(r1-r2)$$

A.2.27 Simplification via multi- and half-angle formulas: multiangle(t,{'symbolic'})

Sometimes, the result of a computation can be simplified significantly by removing numerical factors from the occurring trigonometric functions. The procedure multiangle automatically detects such numerical factors of the form $\frac{m}{2^n}$, where both m and n are integers, and applies first multi- and then half-angle formulas to remove them. The keyword symbolic can be used to suppress the exact treatment of the sign. Note that in some special cases, when the result is a polynomial, convert/FormalPowerSeries can be used to achieve the same simplification; however, multiangle works in more situations:

```
> sin(3*arcsin(a)):
  %=multiangle(%);
  %%=convert(%%,FormalPowerSeries,a,n);
```

$$\sin(3\arcsin(a)) = -4a^3 + 3a$$

$$\sin(3\arcsin(a)) = -4a^3 + 3a$$

```
> cos(1/2*x):
  %=multiangle(%);
```

$$\cos\left(\frac{1}{2}x\right) = \frac{1}{2}(-1)^{\mathrm{floor}\left(\frac{1}{2}\frac{x+\pi}{\pi}\right)}\sqrt{2+2\cos(x)}$$

```
> sin(3/2*arcsin(a)):
  %=multiangle(%,symbolic);
```

$$\sin\left(\frac{3}{2}\arcsin(a)\right) = -\frac{1}{2}\left(2-2\sqrt{1-a^2}\right)^{3/2} + \frac{3}{2}\sqrt{2-2\sqrt{1-a^2}}$$

A.2.28 Factorizing with positive trailing coefficients: posfactor(t)

The procedure posfactor works exactly like factor, except that it gives priority to making the trailing coefficients of all factors positive:

```
> (1-a^2);
  factor(%);
  posfactor(%%);
```

$$1-a^2$$

$$-(a-1)(a+1)$$

$$(-a+1)(a+1)$$

A.2.29 Combining powers: combinepowers(t)

Consolidates powers of the same term into a single expression:

```
> (x^k)^2*x:
  %=combinepowers(%);
```

$$(x^k)^2 x = x^{2k+1}$$

This also works inside more complicated expressions, while affecting only powers:

```
> Sum(Sum(c[n,m]*a^m*a^n/a,n=0..infinity),m=0..infinity):
  %=combinepowers(%);
```

$$\sum_{m=0}^{\infty}\left(\sum_{n=0}^{\infty}\frac{c_{n,m}\,a^{m}\,a^{n}}{a}\right)=\sum_{m=0}^{\infty}\left(\sum_{n=0}^{\infty}c_{n,m}\,a^{m+n-1}\right)$$

A.2.30 Expanding sums: `Expand(t)`

The `Expand` command expands `Sum`s involved in `t`:

```
> Sum(c*a[n]+d*b[n],n=0..infinity);
  Expand(%);
```

$$\sum_{n=0}^{\infty}\left(c\,a_{n}+d\,b_{n}\right)$$

$$c\left(\sum_{n=0}^{\infty}a_{n}\right)+d\left(\sum_{n=0}^{\infty}b_{n}\right)$$

It also fully expands more involved terms:

```
> Sum(x[k]+y^k*Sum((b[n]+c[k])*y^(k+n),n=0..infinity),k=2..K);
  Expand(%);
```

$$\sum_{k=2}^{K}\left(x_{k}+y^{k}\left(\sum_{n=0}^{\infty}\left(b_{n}+c_{k}\right)y^{k+n}\right)\right)$$

$$\sum_{k=2}^{K}x_{k}+\left(\sum_{n=0}^{\infty}y^{n}\,b_{n}\right)\left(\sum_{k=2}^{K}y^{2\,k}\right)+\left(\sum_{n=0}^{\infty}y^{n}\right)\left(\sum_{k=2}^{K}y^{2\,k}\,c_{k}\right)$$

A.2.31 Mapping into sums: `summap(proc,t,{i})`

The `summap` command maps the procedure `proc` to the inside of a sum in `t`. If `i` is not specified, the deepest-level sum is targeted. If `i` is given, the sum with `i` as its summation index is targeted:

```
> Sum(Sum(c[n,k]*a^(2*n)*d^k,n=0..infinity),k=0..infinity);
  summap(P,%);
  summap(P,%%,k);
```

$$\sum_{k=0}^{\infty}\left(\sum_{n=0}^{\infty}c_{n,k}\,a^{2\,n}\,d^{k}\right)$$

$$\sum_{k=0}^{\infty}\left(\sum_{n=0}^{\infty}P\left(c_{n,k}\,a^{2\,n}\,d^{k}\right)\right)$$

$$\sum_{k=0}^{\infty}P\left(\sum_{n=0}^{\infty}c_{n,k}\,a^{2\,n}\,d^{k}\right)$$

A.3 GiosTools cheat sheet

Show coordinate systems	`showcoord(systems/coords)`
Coordinate transformation equations	`coordsys1 &> coordsys2`
Coordinate change	`Dchange(eqns,term,options)`
Coordinate range assumptions	`assumecoords(‘on‘/‘off‘)`
Assume indices to be non-negative integers	`assumeinds(‘on‘/‘off‘)`
Plot in $\{\alpha, \theta\}$ coordinates	`ratplot(f,{‘alpha=‘...,‘theta‘=...,options})`
Laplacian	`L(coordsys)(term)`
Potential	`VOP(coordsys),VOPO(coordsys)`
Hyperspherical harmonics Y_l^n	`HH(coordsys)(n,l)`
Derive Recurrence Relations	`RecRel(coordsys)(ansatz)`
Invert the Laplacian on polynomials	`invertpolynomial(poly,{h})`
Verify identity of two terms	`compare(t1,t2,{s,vars,‘onlyeven‘,‘onlyodd‘})`
Translate maple to LaTeX	`getTeX(term,{‘clip‘,‘cdots‘})`
Identify sequences	`IdSeq(seq{ind})`
List storage tools	`app2seq(ele,{i})`, `setseq(seq,{i})`, `resetseq({i})`, `getseq({i})`
Identify series	`IdSer(ser,{ind})`
Convert → hypergeometric function	`convert(term,‘hyper‘)`
Convert → power series	`convert(term,‘sum‘,{vars:=auto,index,options})`
Collect in `sqrt` and fractional powers	`Collect(term,c,{form,func})`, e.g. `c=sqrt`, `c=ratpower({base,exp})`
Convert → pochhammer	`convert(term,‘poch‘)`
Convert → pochhammer & factorial	`convert(term,‘pochfac‘)`
Trim pochhammers & factorials	`trimpoch(term)`, `trimfac(term)`, `trimpochfac(term)`
Simplify of factorial-like terms	`gsimplify(term)`
Relate factorial-like terms	`gchange(t1,t2)`
Split factorial-like terms	`gsplit(term)`
Symbolic simplification	`symplify(term)`
Simplify via multi- and half-angle formulas	`multiangle(term,{‘symbolic‘})`
Factorize with positive trailing coefficients	`posfactor(term)`
Combine powers	`combinepowers(term)`
Expand sums	`Expand(term)`
Map into sums	`summap(proc,term,{i})`