The current version of Physics under development at Maplesoft has this datestamp:

```
> restart; Physics:-Version( )[2];
```

2014, November 13, 19:34 hours **(1)**

If you installed Physics.mla successfully (see installation instructions in the zip containing Physics.mla) you will see the date shown above. The following computations are reproducible only with this current version.

NOTE: This version of Physics only works in Maple 18. For the last update produced for Maple 17 see the Maple Physics: Research & Development webpage.

The update information in this worksheet referes to changes that happened between January and the end of July of 2014. All these changes are included in Maple 18.02 just released.

The information regarding the updates of Physics from August 1 till today is present in a separate worksheet, PhysicsUpdates18 (II).mw to be included by November 15 in the zip that contains this worksheet.

July 30: *all the changes till this date* are included in Maple 18.02.

- Avoid noncommutative products of tensors whithin Physics:-Library:-SubstituteTensor
- Fix in the tracking of algebra rules for interrelated commutators that involve vectorial quantum operators
- Fix in the computation of commutators involving cross products of quantum vectorial operators (typically related to angular momentum in quantum mechanics)
- Several adjustments supporting the implementation of "powers of vectors" reinterpreted as in $\vec{A}^{2} = \vec{A} \cdot \vec{A}$ and $\vec{A}^{3} = \vec{A} \cdot \vec{A} \cdot \vec{A} = (\vec{A} \cdot \vec{A}) \cdot \vec{A}$, that is left-associative
- Fix a case of cross product of non-projected quantum vectorial operators
- Tell type about the vectorial component (Vectors:-Component command) of a quantum vectorial operator

Examples

July 21

- Fix weakness in the normalization of tensor indices for some tensor cases where the the tensor is antisymmetric under permutation of these indices.
- Fix in Library:-Commute related to functions of noncommutative operators
- Enhance the computation of multiparameter algebra rules

Examples

July 14

- Implement automatic determination of symmetry under permutation of tensor indices when the tensor is defined as a matrix
- Implement a new routine to use, in a given tensorial expression, the same repeated indices used in another tensorial expression
- Improvements in the simplification of tensorial expressions

Examples

July 7

- Fix the identification of the vectorial character of a commutator involving vectors
- Improve the simplification with respect to side relations (equations) in the presence of quantum vectorial equations
- Fix the Dagger of inert Kets and Bras (so represented prefixed by %, as in %Ket, %Bra)
- Handle Inverse(A) . Ket the same as 1/A . Ket
- Implement a new mechanism to have more than one algebra rule related to the same function (e.g. a function of two arguments that come in different order)
- Fix bug in Library:-SubstituteTensor, discussed in two posts in Mapleprimes the first week of July .
- Make Library:-FlipCharacter to automatically map over equations, the same way Covariant and Contravariant do.
- Add new Library:-RepositionRepeatedIndicesAsIn routine, to reposition the repeated indices of a tensor according to given pattern of repetitions. This routine is rather relevant within SubstituteTensor and in the Simplification of tensorial expressions writen using Einstein's sum for repeated indices.
- Relevant change to handle, within Library:-ToCovariant and Library:-ToContravariant, when the spacetime is curved, the mapping of objects that are not really tensors in curved spacetimes like d_, dAlembertian, diff and Fundiff.

Examples

July 1

- Allow multiplying (dot product) to a list automatically distributing, as in $A \cdot [a, b, c] = [A \cdot a, A \cdot b, A \cdot c]$
- Implement $[A, B^k]_{-} = k\, C\, B^{k-1}$ when $[A, B]_{-} = C$ and it is assumed that k is an integer
- Fix in the application of the rule $[A, F(B)]_{-} = [A, B]_{-}\, \mathrm{D}(F)(B)$ when A and B are actually quantum ***vectorial*** operators (the dot product is not associative)
- Adjust the computation of "integer powers of a vector" as the dot product applied an integer number of times

▶ Examples

June 24

- Allow multiplying equations also when the product operator is in scalar and vector products (Vectors:-`.` and Vectors:-`&x`)
- Allow computing Commutators and Anticommutators of an expression with an equation

▶ Examples

June 19

- Implement $f(A) \mid A_a \rangle = f(a) \mid A_a \rangle$, including cases like for instance $f(\alpha A) \mid A_a \rangle = f(\alpha a) \mid A_a \rangle$, or $e^{\frac{I f(t) A}{\hbar}} \mid A_a \rangle = e^{\frac{I f(t) a}{\hbar}} \mid A_a \rangle$
- Implement that an operatorial function $F(H)$ is Hermitian if H is Hermitian and F is assumed to be real, via *assume*, *assuming* or *Setup*(*realobjects* = F).
- Implement that $F(H)$ is Hermitian if H is Hermitian and F is a mathematical real function, that is, one that maps real objects into real objects; in this change only *exp*, the trigonometric functions and their inert forms are included.
- Implement the expansion of the square a vectorial expression as the scalar (dot) product of the expression with itself, including the case of a vectorial quantum operator expression

▶ Examples

June 17

- Improve new functionality allowing parameterizing Commutators with more than one parameter, including the neutral cases, i.e. 0 for sum and 1 for values of the parameters. Also, with this change, only commutative parameters are allowed - see example below.
- Add a few previously missing Unitary and Hermitian operator cases:

  a) if U and V are unitary, the U*V is also unitary

  b) if A is Hermitian then exp(I A) is unitary

  c) if U is unitary and A is Hermitian, then U A Dagger(U) is also Hermitian

▶ Examples

June 10 and 14

- Implement Physics:-Library:-Commute(A, F(A)) = true whenver A is a quantum operator and F is a commutative mapping (see Cohen-Tannoudji, Quantum Mechanics, **(40)**, pag. 171)
- Allow differentiating with respet to a noncommutative variable whenever all the variables present in

the derivand commute with the differentiation variable.

- Implement the automatic computation of $F(X) \bullet Ket(X, x) = F(x)\ Ket(X, x)$, that is the automatic computation of a function of an operator applied to its eigenkets (see Cohen-Tannoudji, Quantum Mechanics, page 171)
- Implement parameterized commutators; for example: when setting the rule $Commutator\left(A, e^{\lambda B}\right) = \lambda\ C$, take lambda as a parameter, so $Commutator\left(A, e^{\alpha B}\right)$ now returns $\alpha\ C$, not $\lambda\ C$.
- Implement the automatic derivation of a commutator rule: Commutator(A, F(B)) = F '(B) when Commutator(A, C) = Commutator(B, C) = 0 and C = Commutator(A, B), as shown in (see Cohen-Tannoudji, Quantum Mechanics, **(51)**, pag. 171)
- Add new Physics:-Library:-GetByMatchingPattern routine (see example below)
- Fix issue regarding the product of a generic function of an operator times a Ket and equivalent constructions with Bras
- Make more precise the type definition for *ExtendedQuantumOperator* to include as such any arbitrary function of an *ExtendedQuantumOperator*
- Fix issue in the product of a matricial Vector[row] and a Matrix in the presence of noncommutative Vector and/or Matricial components
- Allow to sum over a list of objects, or over `in` structures like '$j \in [a, b, c]$' when redefining *sum*, and also in Physics:-Library:-Add

▶ Examples

June 5

- Implement, within the `*` operator, both the global and the Physics one, the product of equations as the product of left-hand sides equal the product of right-hand sides, eliminating the frequently tedius typing lhs(eq[1])*lhs(eq[2]) = rlhs(eq[1])*rhs(eq[2])

▶ Examples

June 3

- Fix issue with the expansion of Vectors:-Norm in the presence of the new optional argument *conjugate = true*
- Fix issue in convert/abs of Vector:-Norm in the presence of conjugate = true
- Fix issue in Fundiff (functional differentiation) of constructions that involve Vector:-Norm in the presence of conjugate = true
- Fix issue with combination of integrals involving Dirac functions
- Fix issue in the conversion to *g* (the spacetime metric) of products of Christoffel symbols that have contracted indices between themselves
- Implement display of inert derivatives of Bras and Kets when the compact mathematical display of PDEtools:-declare is ON

Related to changes of basis and of coordinates used in vectorial expressions:

- $\arctan(\sin(y), \cos(y))$ now returns y when $-\pi < y \leq \pi$, as well as returns 2 Pi − y when $\pi < y \leq 2\pi$
- $sin(arg(z))$ and $cos(arg(z))$ now respectively return as $sin(arg(z))$ and $cos(arg(z))$ when $\Re(z)$ is computable into something different from $'\Re(z)'$
- Fix issue: *csgn(0, z, value_at_0)* now passes the information on the value_at_0 when it returns in terms of signum, and that in turn always happens when *z::real*
- Three different algorithms got plugged into ChangeBasis in order to
  a) For $\alpha = \phi$ or θ, eliminate the constructions $sin(\alpha)$ and $cos(\alpha)$ in terms of products of cartesian coordinates with the cylindrical *r* or ρ before eliminating α using $arctan(y, x)$ or $arctan(\rho, z)$
  b) Eliminate the constructions $\sqrt{x^2 + y^2} = \rho$ and $\sqrt{x^2 + y^2 + z^2} = r$ before eliminating x, y and z in favor of cylindrical or spherical coordinates
  c) Correct the quadrant of $arccos(cos(..,.))$ and $arccos(sin(..,.))$ that may had been introduced in the intermediate steps within ChangeBasis.

▶ Examples

May 25

- Add support for the simplification and integration of spherical harmonics (Maple's SphericalY) relevant in quantum mechanics

▶ Examples

May 22 and 23

- Fix issue in simplify/Dirac
- Change default behavior in Physics:-FeynmanDiagrams in order to discard, now by default, all terms that include tadpoles, and add an option, includetadpoles, to have these terms included.
- Fix typesetting issue for normal products: the typesetting was not taking into account the compact display set with PDEtools:-declare

▶ Examples

May 15

- Fix an issue where operations on dot products, typically involving Kets, Bras and Quantum Operators, were wrongly applied to vector scalar products when Physics:-Vectors was also loaded

▶ Examples

May 7

- Fix issue in 1D notation, with assuming and the shortcut syntax for transpose and Hermitian transpose, respectively A^{+} and A^{*} when A is a Matrix and the .
- Adjusment in the simplification algorithm of tensorial expressions regarding when to recurse one of the internal subroutines recognizing patterns of contracted indices

▶ Examples

May 2 and 6

- Fix an issue with the internal representation of nested covariant derivatives in curved spacetimes that was preventing its rewriting in terms of non-covariant derivatives using d_ (convert/d_)
- Adjust the algorithm that decides the internal ordering used to simplify contracted products of tensors, in this way improving the simplification capabilities
- Fix typo that, when A was a Matrix, prevented the A^{-1} notation for computing the inverse of a Matrix to be fully evaluated

▶ Examples

April 28

- Fix issue with the reporting of free indices of derivatives, where covariant indices in the denominator represent contravariant indices of the tensorial structure and viceversa
- Implement recursive tensor simplification of mathematical functions whose arguments are tensorial expressions.
- Add new Physics:-Library:-FlipCharacterOfFreeIndices for tensorial manipulations
- Improve Library:-IsTensorialSymmetric and IsTensorialAntisymmetric to handle many lists of indices at the same time (symmetry/antisymmetry with respect to all of the lists given)

▶ Examples

April 23, 24 and 25

- Fix issue with the scalar product of non-real vectors: the Norm should (or not) include the optional argument conjugate = true, according to whether you enter A_ . A_ or A_ . conjugate(A_)
- Fix issue with the expansion of an expression involving inert derivatives when some functions of the expression are indicated to be frozen
- Fix issue with automatic simplification of the Weyl tensor in curved spaces

▶ Examples

April 21

- Fix issue in the simplification of tensorial expressions: the contravariant character of a contracted

index within tensorial nested contractions was not restored properly

▶ Examples

April 17 and 18

- Two fixes in Library:-SubstituteTensor
- Improvements in Library:-SubstituteTensor

▶ Examples

April 16

- Fix itempotency in expand/Int
- Large speedup in int/Dirac in the presence of multiple integrals
- Fix weakness in functional differentiation (Fundiff) in the case of multiple integrals, partly related to the problem fixed in expand/Int (first item above).

▶ Examples

April 15

- Fix an unexpected error interruption in Library:-SubstituteTensor when the target expression was of type relation
- Fix an unexpected error interruption in Simplify for some particular expression

▶ Examples

April 10

- Fix issue with the display of derivatives of Dagger

▶ Examples

April 5, 6 and 8

- Further adjustments in the Setup applet: a) fix an unexpected error interruption; b) fix the clearing of algebra rules through the setup applet when reloading the default setup after having loaded the advanced setup.
- Fix issue with the display of derivatives where the derivand uses special typesetting rules
- Fix issue with the global saving of a Physics setup
- Fix issue reported in Mapleprimes regarding the product of a non-square Matrix and a Vector
- Implement the ability in Fundiff to compute functional derivatives by passing only a function name

as second argument; it works OK when the derivand contains this function with only one dependency (may be many variables), say X. This permits variating a function quite like we do with paper and pencil, instead of the formerly orthodox approach of differentiating with respect to a function with different functionality, say Y, to then replace in the output all occurrences of Y by X. Nice simplification of the usage. On the way, remove occurrence of =~ to keep compatibility with Maple 17 while we update for it no the web.

- Avoid placing assumptions on annihilation and creation operators when using assuming
- Rewrite some remaining old differentiation routines to assure their integration with Physics in the presence of noncommutative variables
- Adjustments in the conversion from D_[j] to d_[j] when the index j has a numerical value
- Add new option changefreeindices to the Library:-ToCovariant, and Library:-ToContravariant to actually lower and raise the free indices instead of returning an expression that is mathematically equivalent (free indices unchanged). (This new functionality may end up detached into two new routines Library:-LowerFreeIndices and Library:-RaiseFreeIndices.)
- A couple of bug fixes in the typesetting of mcomplete structures
- Fix derivatives of inert %Re and %Im when the argument is a sum

▶ Examples

March 17

- Change default in Physics:-Vectors:-Norm so that it now returns the Euclidean real norm by default, i.e.: $Norm(\vec{v}) = \vec{v} \cdot \vec{v}$, and only return using conjugate, as in $Norm(\vec{v}) = \vec{v} \cdot \overline{\vec{v}}$, when the option conjugate is passed, or the setting *normusesconjugate* is set using Physics:-Setup
- Add a Setup option *normusesconjugate* so that the default = false can be changed to be equal to true.
- Update the Setup applet to include this option as a mode of operation.

▶ Examples

February 20

- Step 1) of 2-steps change in design:
  1) do not introduce KroneckerDelta as a tensor in results by Physics commands;
  2) change in Setup so that KroneckerDelta is not considered a tensor anymore

This change addresses the ambiguity that happens when the spacetime is not Euclidean (as it happens by default when loading Physics) and KroneckerDelta is indexed with both indices covariant: as a tensor, $\delta_{mu,\,nu} = g_{mu,\,nu}$ so that $\delta_{1,\,1} = -1$ and $delta_{alpha}{}^{alpha} = 4$, while taking KroneckerDelta as the symbol that returns 1 or according to whether their indices are equal or different, both computations should return 1. The ambiguity does not turn evident until the indices are given numerical values and both are covariant or contravariant, or when the the indices are equal, one covariant and the other contravariant, and the index is a spacetime index.

▶ Examples

February 17, 18 and 19

- Fix issue with the new Library:-GetTensorDependency, when acting on inert covariant derivatives
- Fix issue when defining tensor functions in terms of other tensor functions involving inert covariant derivatives
- Check for an unallowed redefinition of general relativity tensors
- Fix an issue with value of indexed objects: activate them the same way inert symbols get activated
- Prevent SubstituteTensorIndices to transform into all covariant or all contravariant the one covariant and one contravariant indices of KroneckerDelta when the spacetime metric is not Euclidean

▶ Examples

February 7 and 12

- Fix issue within Define: when, in a tensor equation definition, a free index also appears as a function name, construct the tensor-procedure definition using dummies as parameters to avoid replacing both the index and the function name when the tensor assumes numerical values.
- Fix issue with Define history: avoid returning some rare cases of tensors in internal representation
- Redirect the flow of differentiation rules in order to interact correctly in the presence of anti and non commutative variables

▶ Examples

January 26

- Fix, within TensorArray and SumOverRepeatedIndices, the ambiguity of Kronecker[i,j] in non-Euclidean spacetimes when i and j are spacetime indices
- Fix the simplification of products of LeviCivita tensors in curvedspacetimes when LeviCivita represents the galilean pseudo-tensor (related to Setup(levicivita = galilean)), instead of its generalization to curved spaces (related to Setup(levicivita = nongalilean))
- Fix Library:-ToContravariant and Library:-ToCovariant when acting over LeviCivita and the spacetime is curved: was missing a factor related to the determinant of the spacetime metric

▶ Examples

January 22

- Fix the copy and paste of annihilation/creation operators (restriction applies) including the use of equation labels with expressions containing them

▶ Examples

January 21

- Fix issue with inert vectorial differential operations on projected vectors constructed with non-projected vectors

Examples