

Maple 17 の [Physics](#) パッケージには、テンソル解析、特殊相対性理論および一般相対性理論用の 7 個の新しいコマンド、Physics 用の 96 個のプログラミングコマンドと 58 個の特殊なタイプを含む新しいライブラリ、ベクトル解析から量子力学までの全体的な機能を強化する数百に及ぶ変更が追加されました。[Physics](#) は Maple 16 で機能が倍になり、Maple 17 でさらに 33% 増強され、物理学における代数計算の最先端環境を提供するという Maplesoft の目標を実現しています。

物理と数学の機能に加え、他にも多くの変更がありました。計算での自然な操作性が強化されて、紙と鉛筆を使った計算方法に近い使用感が得られるようになり、結果表示はテキストブックのような品質に達しています。これには、ベクトル、テンソルインデックス、反交換的または非交換的な変数/関数を表すための新しい [リテラル下付き文字変数](#) の使用などが含まれます。

テンソルと相対性

- テンソル表記を使用した相対性で機能する 5 つの新しいコマンドセットが追加されました。[ExteriorDerivative](#)、[Geodesics](#)、[KillingVectors](#)、[LieDerivative](#)、および [LieBracket](#) というコマンドは、概念的にすべて相互に関連しています。このセットにより、[Physics](#) で用いられる自然な代数テンソル表記を使用した一般相対性での操作に関して Maple 16 での不足が補われ、微分形式を使用した再計算を実行する必要がなくなります。
- 2 つの新しいコマンド、[Antisymmetrize](#) と [Symmetrize](#) では、テンソル式を受け取り、インデックスフリーですべての順列に対する平均を計算して得られる、完全な対称テンソルと完全な非対称テンソルをそれぞれ返します。[Antisymmetrize](#) の場合、順列が偶数が奇数かに応じて符号 +1 または -1 を付けます。特に、[Antisymmetrize](#) に渡される式が非対称性テンソルの積の場合、出力もまた外積のテンソルになります。
- [Physics](#) では、紙と鉛筆を使って解く場合と同じように、共変および反変インデックスを使用して、これらのテンソルを代数計算を使って解くことができます。その際 Maple が自動的に対称性プロパティを考慮し、反復されるインデックスにアインシュタインの縮約記法を使用するとともに、縮約インデックスの各ペアの 1 つの反変インデックスに自動設定することで、式のテンソルインデックスの[チェック](#)、[区別](#)、[簡単化](#)を行います。Maple 17 では、積で反復されるインデックスの簡単化は、構成された積での対称性または非対称性副次式の特定、およびユーザー定義の対称性プロパティを考慮した自動正規化の導入という複数の方法で実行できます。
- Maple 17 では、[テンソルを定義](#)する場合 ($F^{\mu\nu}$ など)、その他のテンソルを含む任意のテンソル式として定義できます。たとえば、 $F^{\mu\nu} = \partial^\mu(A^\nu) - \partial^\nu(A^\mu)$ を渡して [定義](#)したり、テンソル式の対称性プロパティを自動的に使用して、 $F^{\mu\nu}$ 構成されたインデックスの簡単化 [を行う場合に](#)を代数式で正規化します。さらに、Physics のすべての内部ルーチンは、Physics で事前定義されたテンソルと同じ条件で新しいテンソルの処理を行います (行列形式の表示オプション、共変および反変インデックスが数値を仮定する場合の評価、行列または非ゼロコンポーネントの表示オプションなど)。詳細は [g](#) を参照してください。
- 計量のキーワードの一部を直接計量 [g](#) に渡すことで、すべてのデータベースでの計量の一致をリストし、詳細な検索を行い、必要な検索結果を選択できます。Maple 17 では、計量のデータベースが大幅に拡張されました ([参考文献](#) 参照)。
- [Riemann](#) テンソルコマンドで時空間計量セットに対するリーマン不変量を返すことが可能になりました。これには、[Riemann\[invariants\]](#) でのテンソルインデックスの代わりにキーワード [invariantsRiemann](#) を使用します。

- 時空間座標系を表現するため、[Physics:-Vectors](#) サブパッケージでの定義済み直交座標系、円筒座標系および球面座標系などの座標系を示す名前リストと、時間を表す t を使用できるようになりました。また、多くの座標系を同時に組み合わせて操作することができます。Maple 17 では、これらの座標系で作られた [SpaceTimeVector](#) に機能依存性があります。たとえば、すべての座標に対する固有時の τ です。これは特に、[Geodesics](#) を表現または計算する場合か、座標の 3-D 空間のサブセットを操作する場合に役立ちます。
- [Setup](#) を使用して、すべての [Physics](#) 設定をセッション中に保存し、同じセッションや新しいセッションに読み込むことが可能になりました。この機能は、[Setup](#) グラフィックアプレットから直接利用することもできます。この場合は [Setup\(\)](#) のように引数なしで [Setup](#) をコールすると、すべての [Physics](#) 設定が含まれます。
- [96 個のプログラミングコマンドと 58 個の特殊な Physics タイプを含む Physics の新しいライブラリ](#) には、テンソルを扱う対話式計算でも有用なコマンド (たとえば、[IsTensorialAntisymmetric](#)、[IsTensorialSymmetric](#)、and [TensorComponents](#) など) も含まれていますが、数は多くありません。

ディラック行列、交換子/反交換子、および代数

- [Trace](#) コマンドは 2 つの異なるモードで操作します。デフォルトでは (新機能)、[ディラック](#) 行列や [パウリ](#) 行列に対する代数的表現を除いて、スカラーを 1 つのインデックスのテンソル、または明示的に matrix または Matrix (Array) 型 オブジェクトとみなしません。従来の動作は、定数がトレース可能なオブジェクトとみなされる場合を除いて、新しい [Setup](#) オプション [traceonlymatrices = false](#) (デフォルト値は [true](#)) を使ってリストアできます。これにより、係数が単に定数ではない場合の [ディラック行列](#) を含む式のトレース計算という一般的な例で、無理なく [Trace](#) を使用できます。
- [Setup](#) を使用して [交換子](#) および [反交換子](#) 代数が設定されている場合、代数ルールセットの一貫性についての基本的なテストが実行され、不一致が検出された場合には、対話式のメッセージ表示とともに処理を中断します。同様に、一部の代数ルールには関連する要素の反交換性または非交換性が含まれるため、これらの要素は、これに応じて反交換的または非交換的と自動設定されます。
- 量子およびエルミート演算子と同じ条件でユニタリ演算子を定義できます。ユニタリ演算子が設定されると、[Setup](#) によって代数ルールが設定で自動的に作成されます。これらのルールは、この演算子を含む積を単純化する場合に [Simplify](#) で自動的に考慮されます。
- 代数ルールにテンソルの [共役](#) および [ダガー](#)、量子演算子 (たとえば、 A) が含まれる場合、[Setup](#) で決められた A の代数ルールを考慮して処理されます。[FeynmanDiagrams](#) で、共役スピノルは反交換的フィールドの [共役](#) または [ダガー](#) を使用して表現できます。同様に、[ブラケット](#)、[共役](#)、[ダガー](#)、[トレース](#)、および量子演算子の [逆行列](#) を含む関数の合成について、処理方法をより高度に認識できるようになりました。

ベクトル解析

- [Physics:-Vectors](#) のすべてのベクトル解析ルーチンで、 $\theta(t)$ や単位ベクトル $\hat{\rho}(t)$ のような、機能依存性のある、単位ベクトルとジオメトリ座標を使った計算が可能になり、数学表現機能が拡張されました。Maple 17 では新たに、直交座標系、円筒座標系、球座標系間の関係、3 つの座標系とこれらの座標系に関連する単位ベクトルとの関係が、微分を実行する際に自動的に考慮されるようになりました。さらに、これらの幾何学的な関係における機能の伝搬も、紙と鉛筆を使って解く際に自然に行われるように、コンピュータで実行されます。簡単な例を挙げると、 $x(t)$ を含む式の場合、 ϕ に関して微分するときに考慮される関係は、 $x = \rho \cos(\phi)$ ではなく $x(t) = \rho(t) \cos(\phi(t))$ になります。
- これらの変更に関連して、Maple 16 のコマンド [Physic:-Vectors:-VectorDiff](#) は [Physic:-Vectors:-diff](#) に名前変更され、[Physic:-diff](#) と同じものになっています。また、この [diff](#) コマンドで、上述の座標系と単位ベクトル間の幾何学的な関係を考慮するために、オプションの引数 [geometricdifferentiation](#) を使用できるようになりました。この

[diff](#) コマンドの新機能は、[Vectors](#) が読み込まれると、自動的にオンに設定されます。ただし、[設定](#) を使用するといつでもオンとオフを切り替えられます (キーワード [geometricdifferentiation](#))。

Physics 用の新しいプログラミングコマンドのライブラリ

- 96 個のコマンドと 58 個の特殊な物理タイプを含む新しい [プログラミングライブラリ](#) が [Physics](#) に追加され、Physics コマンドをプログラミングするために Maplesoft で使用される内部ルーチンとタイプがユーザーレベルで選択しやすくなりました。これらのプログラミングツールを使用すると、Physics パッケージと完全に互換性がある方法で、独自のプログラムを作成できるほか、[Physics](#) の機能を拡張することもできるため、最初からすべてのプログラムを作成する必要はありません。このようにして、ユーザー独自のプログラムは [Physics](#) のプログラムと直接やりとりを行うことができます。
- Physics:-Library の多くのプログラミングツールは、対話型計算にも役立ちます。いくつか例を挙げると、[AntiCommute](#) と [Commute](#) はそれぞれ 2 つのオブジェクトが反交換か交換かを識別し、[IsTensorialAntisymmetric](#) と [IsTensorialSymmetric](#) はそれぞれ、与えられたテンソル式が自由インデックスの順列の観点から (または自由インデックスの特定のペアに関して) 全非対称性か全対称性かを識別します。[Backup](#) と [Restore](#) はそれぞれ、Physics の全設定のバックアップをある時点で作成し、調査用の計算を実行した後、設定の変更を行ってればこの設定を元に戻します (代数ルールを使用した量子演算子の再定義、時空間またはその計量の次元の変更など)。

参照

[Maple 17 の新機能一覧](#)、[Physics](#)