*"Is there a convert command that converts a matrix to a tensor"*

*with Physics* :

First of all, even working with matrices, you are working in some *space* and, in general, using some coordinates, and if working with tensors you use certain type of letters as *tensor indices*. Suppose it is a 3D Euclidean space, your coordinates are Cartesian, and you want to use lowercase letters as indices; set that:

*Setup dimension* = 3, *metric* = *euclidean*, *coordinates* = *cartesian*, *spacetimeindices*= *lowercase*

*The dimension and signature of the tensor space are set to* 3, CCC

 *Systems of spacetime coordinates are:* *X* = *x*, *y*, *z*

 *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

 *The Euclidean metric in coordinates*  *x*, *y*, *z*

 *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

1 0 0

*g* =0 1 0

μ, ν

 0 0 1

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *coordinatesystems* = *X* , *dimension* = 3, *metric* = *spacetimeindices*= *lowercaselatin*Suppose you have the matrix *Matrix* 3, *symbol* = *m* | 1, 1 | = 1,  | 2, 2 | = 1,  | 3, 3 | = 1, | **(1)** |

 *m m m*

 1, 1 1, 2 1, 3

 *m m m***(2)**

2, 1 2, 2 2, 3 *m m m*

 3, 1 3, 2 3, 3

|  |  |
| --- | --- |
| To transform it into a tensor, just use the Define command *Define M i*, *j* = **(2)***Defined objects with tensor properties* γ*a*, *M* , σ*a*, v*a*, *g* , e*a b c*, *X*  | **(3)** |

 *i*, *j a*, *b* , , *a*

That is all. You can perform all tensor computations with such so defined tensor *M* . Start with its *i*, *j* definition

*M definition*

 *m m m*

 1, 1 1, 2 1, 3

*Mi*, *j* =*m*2, 1 *m*2, 2 *m*2, 3**(4)** *m m m*

 3, 1 3, 2 3, 3

Then itscomponents (default is the *covariant* ones)

*M*

 *m m m*

 1, 1 1, 2 1, 3

*Ma*, *b* =*m*2, 1 *m*2, 2 *m*2, 3**(5)** *m m m*

 3, 1 3, 2 3, 3

The *contravariant* components are the same as the covariant ones because the space is Euclidean

*M* ~

 *m m m*

 1, 1 1, 2 1, 3

*M a*, *b* =*m*2, 1 *m*2, 2 *m*2, 3**(6)** *m m m*

 3, 1 3, 2 3, 3

The trace

*M trace*

 *m* C*m* C*m* **(7)**

 1, 1 2, 2 3, 3

Also via

*M j*, *j*

*M* **(8)** *j*, *j*

*SumOverRepeatedIndices* **(8)**

 *m* C*m* C*m* **(9)**

 1, 1 2, 2 3, 3

The determinant *M determinant*

*m*  *m*  *m* L*m*  *m*  *m* L*m*  *m*  *m* C*m*  *m*  *m* C*m*  *m*  *m* L*m*  *m*  *m* **(10)**

 1, 1 2, 2 3, 3 1, 1 2, 3 3, 2 1, 2 2, 1 3, 3 1, 2 2, 3 3, 1 1, 3 2, 1 3, 2 1, 3 2, 2 3, 1

The inert representation of this determinant

*%M determinant*

  **M (11)**

*value* **(11)**

*m*  *m*  *m* L*m*  *m*  *m* L*m*  *m*  *m* C*m*  *m*  *m* C*m*  *m*  *m* L*m*  *m*  *m* **(12)**

 1, 1 2, 2 3, 3 1, 1 2, 3 3, 2 1, 2 2, 1 3, 3 1, 2 2, 3 3, 1 1, 3 2, 1 3, 2 1, 3 2, 2 3, 1

The non-zero components *M nonzero*

*M* = = *m* , 2, 1 = *m* , 2, 2 = *m* , 2, 3 = *m* , 3, **(13)** *a*, *b*1, 32, 12, 22, 3

,

1

1

=

*m*

,

1

1

2

,

1

,

=

*m*

1

,

2

3

,

1

,

=

*m*

1

3

,

,

,

2

3

=

*m*

3

,

2

,

3

,

3

=

*m*

3

,

3

1

 *"for example to perform basic operations like omega\*J\*omega where J is the inertia tensor and omega the vector of angular velocity?"*

OK. Define your *vector* now. You can do that directly with Define, or also starting with a Maple *Vector* constructor

*Vector* 3, *symbol*= *v*

  **(14)**

*v*

1

*v*

2

*v*

3

|  |  |
| --- | --- |
| *Define V j* = **(14)***Defined objects with tensor properties*γ*a*, *Mi*, *j*, σ*a*, *Vj*, v*a*, *ga*, *b*, e*a*, *b*, *c*, *Xa* For example, the computation you asked for*V i* *M i*, *j* *V j V* *M* *V i i*, *j j**SumOverRepeatedIndices* **(16)** 2 2 | **(15)****(16)** |
| *m*  *v* C*m*  *v* *v* C*m*  *v* *v* C*m*  *v* *v* C*m*  *v* C*m*  *v* *v* C*m*  *v* *v* C*m*  *v* *v* C*m*  **(17)**1, 1 1 1, 2 1 2 1, 3 1 3 2, 1 1 2 2, 2 2 2, 3 2 3 3, 1 1 3 3, 2 2 3 3, 3 *v*2 3Another example: operations with the metric |

*V a* *g\_ a*, *i* *M i*, *j* *g\_ j*, *b V b*

*Simplify*

**(18)**

*V* *g* *M* *g* *V* **(18)** *a a*, *i i*, *j b*, *j b*

 *V* *M* *V* **(19)**

*i i*, *j j*

Or products of tensor (underlying matrices) that take the symmetry properties into account

*LeviCivita a*, *b*, *c* *V a* *V b*

 e*a*, *b*, *c* *Va* *Vb* **(20)**

*Simplify* **(20)**

 0 **(21)**

Or other tensor commands *Symmetrize M a*, *b*