Deriving the mathematical form of 4D relativistic Lorentz transformations

Freddy Baudine(1), Edgardo S. Cheb-Terrab(2)

(1) Retired, passionate about Mathematics and Physics
(2) Physics, Differential Equations and Mathematical Functions, Maplesoft

Lorentz transformations are a six-parameter family of linear transformations $\Lambda$ that relate the values of the coordinates $x, y, z, t$ of an event in one inertial reference system to the coordinates $x', y', z', t'$ of the same event in another inertial system that moves at a constant velocity relative to the former. An explicit form of $\Lambda$ can be derived from physics principles, or in a purely algebraic mathematical manner. A derivation from physics principles is done in an upcoming post about relativistic dynamics, while in this post we derive the form of $\Lambda$ mathematically, as rotations in a (pseudo) Euclidean 4 dimensional space. Most of the presentation below follows the one found in Jackson's book on Classical Electrodynamics [1].

The computations below in Maple 2022 make use of the Maplesoft Physics Updates v.1283 or newer.

**Formulation of the problem and ansatz $\Lambda = e^\Lambda$**

The problem is to find a group of linear transformations, 

$$x'\,^\mu = \Lambda\,^\mu_\nu \, x^\nu$$

that represent rotations in a 4D (pseudo) Euclidean spacetime, and so they leave invariant the norm of the 4D position vector $x^\mu$; that is,

$$x'\,^\mu x'\,^\nu = x^\mu x^\nu$$

For the purpose of deriving the form of $\Lambda\,^\mu_\nu$, a relevant property for it can be inferred by rewriting the invariance of the norm in terms of $\Lambda\,^\mu_\nu$. In steps, from the above,

$$g_{\alpha, \beta} x'\,^\alpha x'\,^\beta = g_{\mu, \nu} x^\mu x^\nu$$

$$g_{\alpha, \beta} \Lambda\,^\alpha_\mu \Lambda\,^\beta_\nu x^\mu x^\nu = g_{\mu, \nu} x^\mu x^\nu$$

$$(\Lambda\,^\alpha_\mu g_{\alpha, \beta} \Lambda\,^\beta_\nu) x^\mu x^\nu = g_{\mu, \nu} x^\mu x^\nu$$

from where,

$$(\Lambda\,^\alpha_\mu g_{\alpha, \beta} \Lambda\,^\beta_\nu) = g_{\mu, \nu}$$

or in matrix (4 x 4) form, $\Lambda\,^\alpha_\mu \equiv \Lambda, g_{\alpha, \beta} \equiv g$

$$\Lambda^T \, g \, \Lambda = g$$

where $\Lambda^T$ is the transpose of $\Lambda$. Taking the determinant of both sides of this equation, and recalling that
\[ \det(\Lambda^T) = \det(\Lambda), \] we get

\[ \det(\Lambda) = \pm 1 \]

The determination of \( \Lambda \) is analogous to the determination of the matrix \( R \) (3D tensor \( R_{i,j} \)) representing rotations in the 3D space, where the same line of reasoning leads to \( \det(R) = \pm 1 \). To exclude reflection transformations, that have \( \det(\Lambda) = -1 \) and cannot be obtained through any sequence of rotations, because they do not preserve the relative orientation of the axes, the sign that represents our problem is +. To explicitly construct the transformation matrix \( \Lambda \), Jackson proposes the ansatz

\[ \Lambda = e^L \]

Summarizing: the determination of \( \Lambda_{\mu}^{\nu} \) consists of determining \( L_{\mu}^{\nu} \) entering \( \Lambda = e^L \) such that

\[ \det(\Lambda) = + 1 \]

**Determination of \( L_{\mu}^{\nu} \)**

In order to compare results with Jackson's book, we use the same signature he uses, \( (+---) \), and lowercase Latin letters to represent space tensor indices, while spacetime indices are represented using Greek letters, which is already Physics' default.

```
restart;
with(Physics):
Setup(signature = "+-+++", spaceindices = lowercaselatin)

\[ \text{signature = + - - - , spaceindices = lowercaselatin} \] (1)
```

Start by defining the tensor \( L_{\mu}^{\nu} \) whose components are to be determined. For practical purposes, define a macro \( LM = L \) to represent the tensor and use \( L \) to represent its components

```
macro(LM = L, %LM = %L) : Define(\Lambda, LM, quiet) :
LM[~mu, nu] = Matrix(4, symbol = L)
```

\[ L_{\mu}^{\nu} = \begin{bmatrix}
L_{1,1} & L_{1,2} & L_{1,3} & L_{1,4} \\
L_{2,1} & L_{2,2} & L_{2,3} & L_{2,4} \\
L_{3,1} & L_{3,2} & L_{3,3} & L_{3,4} \\
L_{4,1} & L_{4,2} & L_{4,3} & L_{4,4}
\end{bmatrix} \] (2)

Define((2))

**Defined objects with tensor properties**

\[ \{ \Lambda, L, g, \sigma, \mu, \nu, \gamma_{\alpha}, \beta, \epsilon_{\mu, \nu} \} \] (3)

Next, from \( \Lambda^T g \Lambda = g \) (see above in **Formulation of the problem**) one can derive the form of \( L \). To work algebraically with \( L, \Lambda, g \) representing matrices, set these symbols as noncommutative

```
Setup(noncommutativeprefix = \{\Lambda, LM, g\})
```

\[ \text{noncommutativeprefix = \{L, \Lambda, g\}} \] (4)
From 

\[ \Lambda^T \cdot g \cdot \Lambda = g \]  

it follows that

\[ g^{-1} \cdot (5) \cdot \Lambda^{-1} \]  

\[ g^{-1} \Lambda^T g = \Lambda^{-1} \]  

\[ \text{eval}((6), \Lambda = \exp(LM)) \]  

\[ g^{-1} \left( e^k \right)^T g = (e^k)^{-1} \]  

Expanding the exponential using 

\[ e^k = \sum_{k=0}^{\infty} \frac{k^k}{k!} \], and taking into account that the matrix product 

\[ g^{-1} \cdot k \cdot g \]  

can be rewritten as \( (g^{-1} \cdot k \cdot g)^k \), the left-hand side of (7) can be written as 

\[ e^{g^{-1} \cdot k \cdot g} \exp(g^{-1} \cdot LM \cdot g) = \text{rhs}((7)) \]  

\[ e^{g^{-1} \cdot k \cdot g} = (e^k)^{-1} \]  

Multiplying by \( e^k \) 

\[ (8) \cdot \exp(LM) \]  

Recalling that 

\[ g^{-1} = g^{\mu, \alpha}, \quad g = g_{\beta, \nu} \]  

and that for any matrix 

\[ \ell, \quad (\ell \cdot T)^{\beta} = \ell^{\beta}_{\alpha}, \]  

\[ g^{-1} \cdot \ell \cdot g = \text{sub}(\ell, LM = LM[\sim \mu, \sim \alpha] \cdot LM[\sim \beta, \sim \nu]) \]  

\[ \exp(g^{-1} \cdot \ell \cdot g) = e^{g^{-1} \cdot \ell \cdot g} \exp(g^{-1} \cdot LM \cdot g) = \text{rhs}((7)) \]  

\[ e^{g^{-1} \cdot \ell \cdot g} = (e^k)^{-1} \]  

To allow for the combination of the exponentials, now that everything is in tensor notation, remove the noncommutative character of \( \ell \). 

\[ \text{Setup}(\text{clear, noncommutativeprefix}) \]  

\[ \text{combine}((11)) \]  

\[ e^{g^{\alpha, \mu}_{\beta, \nu} \cdot \ell^{\mu} + \ell^{\beta}} = 1 \]  

Since every tensor component of this expression is real, taking the logarithm at both sides and simplifying tensor indices 

\[ \text{map}(\ln, (13)) \text{ assuming real} \]  

\[ g^{\alpha, \mu}_{\beta, \nu} \ell^{\beta}_{\alpha} + \ell^{\mu}_{\nu} = 0 \]  

\[ \text{Simplify}((14)) \]  

\[ \ell^{\mu}_{\nu} + \ell^{\mu}_{\nu} = 0 \]  

So the components of \( \ell \)
LM \[ \sim \mu, v, \text{matrix} \]

\[
\mathcal{L}^\mu_v = \begin{bmatrix}
L_{1,1} & L_{1,2} & L_{1,3} & L_{1,4} \\
L_{2,1} & L_{2,2} & L_{2,3} & L_{2,4} \\
L_{3,1} & L_{3,2} & L_{3,3} & L_{3,4} \\
L_{4,1} & L_{4,2} & L_{4,3} & L_{4,4}
\end{bmatrix}
\] (16)

satisfy (15). Using TensorArray the components of that tensorial equation are

\[
\text{TensorArray}(\{2 L_{1,1} = 0, 2 L_{2,2} = 0, 2 L_{3,3} = 0, 2 L_{4,4} = 0, -L_{1,2} + L_{2,1} = 0, L_{1,2} - L_{2,1} = 0, -L_{1,3} + L_{3,1} = 0, L_{1,3} - L_{3,1} = 0, -L_{1,4} + L_{4,1} = 0, L_{1,4} - L_{4,1} = 0, L_{3,2} + L_{2,3} = 0, L_{4,2} + L_{2,4} = 0, L_{4,3} + L_{3,4} = 0\})
\] (17)

Simplifying taking these equations into account results in the form of \( \mathcal{L}^\mu_v \) we were looking for

\[
\text{simplify}((16), (17))
\]

\[
\mathcal{L}^\mu_v = \begin{bmatrix}
0 & L_{1,2} & L_{1,3} & L_{1,4} \\
L_{1,2} & 0 & L_{2,3} & L_{2,4} \\
L_{1,3} & -L_{2,3} & 0 & L_{3,4} \\
L_{1,4} & -L_{2,4} & -L_{3,4} & 0
\end{bmatrix}
\] (18)

This is equation (11.90) in Jackson's book [1]. By eye we see there are only six independent parameters in \( \mathcal{L}^\mu_v \), or via

\[
\text{indets}(\text{rhs}((18)), \text{name})
\]

\[
\{L_{1,2}, L_{1,3}, L_{1,4}, L_{2,3}, L_{2,4}, L_{3,4}\}
\] (19)

\[
\text{nops}(19) = 6
\] (20)

This number is expected: a rotation in 3D space can always be represented as the composition of three rotations, and so, characterized by 3 parameters: the rotation angles measured on each of the space planes \((x, y), (y, z), (z, x)\). Likewise, a rotation in 4D space is characterized by 6 parameters: rotations on each of the three space planes, parameters \(L_{2,3}, L_{2,4}\) and \(L_{3,4}\), and rotations on the spacetime planes \((t, x), (t, y), (t, z)\), parameters \(L_{1,j}\). Define now \( \mathcal{L}^\mu_v \) using (18) for further computing with it in the next section

\[
\text{Define}(18)
\]

\[
\text{Defined objects with tensor properties}
\]

\[
\{\Lambda, \mathcal{L}^\mu_v, \gamma^\mu_v, \sigma^\mu_v, \partial^\mu, g_{\mu, v}, \gamma_{\alpha, \beta, \mu, v}\}
\] (21)

**Determination of** \( \Lambda^\mu_v \)
From the components of $\ell^{\mu}_{\nu}$ in (18), the components of $\Lambda^{\mu}_{\nu} = \exp(\ell^{\mu}_{\nu})$ can be computed directly using the LinearAlgebra:-MatrixExponential command. Then, following Jackson's book, in what follows we also derive a general formula for $\ell^{\mu}_{\nu}$ in terms of $\beta = \frac{v}{c}$ and $\gamma = \frac{1}{\sqrt{-\beta^2 + 1}}$ shown in [1] as equation (11.98), finally showing the form of $\Lambda^{\mu}_{\nu}$ as a function of the relative velocity of the two inertial systems of references.

An explicit form of $\Lambda^{\mu}_{\nu}$ in the case of a rotation on the $(t, x)$ plane can be computed by taking equal to zero all the parameters in (19) but for $L_{1, 2}$ and substituting in $\equiv \ell^{\mu}_{\nu}$

$$L_{1, 3} = 0, L_{1, 4} = 0, L_{2, 3} = 0, L_{2, 4} = 0, L_{3, 4} = 0$$

(19) minus $\{L[1, 2]\} = -0$

subs((22), (18))

$$\ell^{\mu}_{\nu} = \begin{bmatrix} 0 & L_{1, 2} & 0 & 0 \\ L_{1, 2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

(23)

Computing the matrix exponential,

$$\Lambda[-\mu, \nu] = \text{LinearAlgebra:-MatrixExponential}(\text{rhs}((23)))$$

$$\Lambda^{\mu}_{\nu} = \begin{bmatrix} \frac{L_{1, 2}^2}{2} + \frac{e^{-L_{1, 2}}}{2} & -\frac{L_{1, 2}}{2} + \frac{e^{L_{1, 2}}}{2} & 0 & 0 \\ -\frac{L_{1, 2}}{2} + \frac{e^{L_{1, 2}}}{2} & \frac{L_{1, 2}^2}{2} + \frac{e^{-L_{1, 2}}}{2} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(24)

convert((24), trigh)

$$\Lambda^{\mu}_{\nu} = \begin{bmatrix} \cosh(L_{1, 2}) & \sinh(L_{1, 2}) & 0 & 0 \\ \sinh(L_{1, 2}) & \cosh(L_{1, 2}) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(25)

This is formula (4.2) in Landau & Lifshitz book [2]. An explicit form of $\Lambda^{\mu}_{\nu}$ in the case of a rotation on the $(x, y)$ plane can be computed by taking equal to zero all the parameters in (19) but for $L_{2, 3}$

(19) minus $\{L[2, 3]\} = -0$

$$L_{1, 2} = 0, L_{1, 3} = 0, L_{1, 4} = 0, L_{2, 4} = 0, L_{3, 4} = 0$$

(26)
subs ((26), (18))

\[
\mathbb{L}^\mu_v = \begin{bmatrix}
0 & 0 & 0 & 0 \\
0 & 0 & L_{2,3} & 0 \\
0 & -L_{2,3} & 0 & 0 \\
0 & 0 & 0 & 0 \\
\end{bmatrix}
\]  

(27)

\[\Lambda^{\mu}_v = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & \cos(L_{2,3}) & \sin(L_{2,3}) & 0 \\
0 & -\sin(L_{2,3}) & \cos(L_{2,3}) & 0 \\
0 & 0 & 0 & 1 \\
\end{bmatrix}\]  

(28)

\(\Lambda[\sim\mu, v] = \text{LinearAlgebra:-MatrixExponential}(\text{rhs} ((27)))\)

\textbf{Rewriting} \(\mathbb{L}^\mu_v = \zeta^i_i K^i + \omega^i_i S^i\)

Following Jackson's notation, for readability, redefine the 6 parameters entering \(\mathbb{L}^\mu_v\) as

\[
\{LM[1, 2] = \zeta_1, LM[1, 3] = \zeta_2, LM[1, 4] = \zeta_3, LM[2, 3] = \omega_3, LM[2, 4] = -\omega_2, LM[3, 4] = \omega_1\}\}

\[
\{\zeta_1, 2 = \zeta_1, 3 = \zeta_2, 4 = \zeta_3, \omega_1, 2 = \omega_3, 3 = -\omega_2, 4 = \omega_1\}\}

(29)

(Note in the above the surrounding backquotes '...' to prevent a premature evaluation of the left-hand sides; that is necessary when using the \textit{Library:-RedefineTensorComponent} command.) With this redefinition, \(\mathbb{L}^\mu_v\) becomes

\textit{Library:-RedefineTensorComponent((29)) :}

\(LM[\sim\mu, v, \text{matrix}]\)

\[
\mathbb{L}^\mu_v = \begin{bmatrix}
0 & \zeta_1 & \zeta_2 & \zeta_3 \\
\zeta_1 & 0 & -\omega_3 & \omega_2 \\
\zeta_2 & \omega_3 & 0 & -\omega_1 \\
\zeta_3 & -\omega_2 & \omega_1 & 0 \\
\end{bmatrix}\]  

(30)

where each parameter represents a rotation angle on one plane. Any Lorentz transformation (rotation in 4D pseudo-Euclidean space) can be represented as the composition of these six rotations, and to each rotation, corresponds the matrix that results from taking equal to zero all of the six parameters but one.

The set of six parameters can be split into two sets of three parameters each, one representing rotations on the \((t, x_j)\) planes, parameters \(\zeta_j\), and the other representing rotations on the \((x_i, x_j)\) planes, parameters \(\omega_j\). With that, following \[1\], (30) can be rewritten in terms of four 3D tensors, two of them with the parameters as components, the other two with matrix as components, as follows:
\[ \zeta[i] = [\zeta_1, \zeta_2, \zeta_3], \omega[i] = [\omega_1, \omega_2, \omega_3], K[i] = [K_1, K_2, K_3], S[i] = [S_1, S_2, S_3] \]

\[ \zeta_i = [\zeta_1, \zeta_2, \zeta_3], \omega_i = [\omega_1, \omega_2, \omega_3], K_i = [K_1, K_2, K_3], S_i = [S_1, S_2, S_3] \]  

(31)

Define (31)

Defined objects with tensor properties

\[ \{ \Lambda, g, \mu, \nu, \gamma, \sigma, \zeta, \varepsilon, \partial, \omega, i, \alpha, \beta, \mu, v \} \]

The 3D tensors \( K_i \) and \( S_i \) satisfy the commutation relations

Setup (noncommutativeprefix = \{S, K\})


Commutator \((S[i], S[j]) = \text{LeviCivita}[i, j, k] S[k] \)

\[ [S_i, S_j] = \epsilon_{i, j, k} S^k \]

(34)

Commutator \((S[i], K[j]) = \text{LeviCivita}[i, j, k] K[k] \)

\[ [S_i, K_j] = \epsilon_{i, j, k} K^k \]

(35)

Commutator \((K[i], K[j]) = -\text{LeviCivita}[i, j, k] S[k] \)

\[ [K_i, K_j] = -\epsilon_{i, j, k} S^k \]

(36)

The matrix components of the 3D tensor \( K_i \), related to rotations on the \((t, x_j)\) planes, are

\[ K_1 := \text{matrix}([[0, 1, 0, 0], [1, 0, 0, 0], [0, 0, 0, 0], [0, 0, 0, 0]]) \]

\[ K_1 := \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \]

(37)

\[ K_2 := \text{matrix}([[0, 0, 1, 0], [0, 0, 0, 0], [1, 0, 0, 0], [0, 0, 0, 0]]) \]

\[ K_2 := \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \]

(38)

\[ K_3 := \text{matrix}([[0, 0, 0, 1], [0, 0, 0, 0], [0, 0, 0, 0], [1, 0, 0, 0]]) \]

\[ K_3 := \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \]

(39)

The matrix components of the 3D tensor \( S_i \), related to rotations on the \((x_i, x_j)\) 3D space planes, are

\[ S_1 := \text{matrix}([[0, 0, 0, 0], [0, 0, 0, 0], [0, 0, 0, -1], [0, 0, 1, 0]]) \]

\[ S_1 := \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \]
\[ S_i := \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \] (40)

\[ S_2 := \text{matrix}(\begin{bmatrix} [0, 0, 0, 0], [0, 0, 0, 1], [0, 0, 0, 0], [0, -1, 0, 0] \end{bmatrix}) \]

\[ S_2 := \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix} \] (41)

\[ S_3 := \text{matrix}(\begin{bmatrix} [0, 0, 0, 0], [0, 0, -1, 0], [0, 1, 0, 0], [0, 0, 0, 0] \end{bmatrix}) \]

\[ S_3 := \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \] (42)

*Verifying the commutation relations between \( S_i \) and \( K_j \)*

One can verify these matrices satisfy the commutation relations (34), (35), and (36) using `TensorArray`:

\[ [S_i, S_j] = \epsilon_{i, j, k} S^k \] (43)

`TensorArray((34), mm)`

*Partial match of 'mm' against keyword 'performmatrixoperations'*

\[
\begin{bmatrix}
0 = 0, & \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, & \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \\
\end{bmatrix}
\] (44)

\[
\begin{bmatrix}
0 = 0, & \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, & \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \\
\end{bmatrix}
\]
\[
\begin{bmatrix}
0 & 0 & 0 \\
0 & 0 & 1 \\
0 & 0 & 0 \\
0 & -1 & 0
\end{bmatrix}
= 
\begin{bmatrix}
0 & 0 & 0 \\
0 & 0 & 1 \\
0 & 0 & 0 \\
0 & -1 & 0
\end{bmatrix}
= 
\begin{bmatrix}
0 & 0 & 0 \\
0 & 0 & 1 \\
0 & 0 & 0 \\
0 & 0 & 1
\end{bmatrix},
0 = 0
\]

(35)

\[
[S^i, K^j] = \varepsilon^k_{i,j,k}
\]

(45)

TensorArray((35), mm)

* Partial match of 'mm' against keyword 'performmatrixoperations'

\[
\begin{bmatrix}
0 & 0 & 0 \\
0 & 0 & 1 \\
0 & 0 & 0 \\
1 & 0 & 0
\end{bmatrix}
= 
\begin{bmatrix}
0 & 0 & 0 \\
0 & 0 & 1 \\
0 & 0 & 0 \\
1 & 0 & 0
\end{bmatrix}
= 
\begin{bmatrix}
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0
\end{bmatrix},
0 = 0
\]

(46)

\[
[K^i, K^j] = -\varepsilon^k_{i,j,k} S^k
\]

(47)

TensorArray((36), mm)

* Partial match of 'mm' against keyword 'performmatrixoperations'

\[
\begin{bmatrix}
0 & 0 & 0 \\
0 & 0 & 1 \\
0 & 0 & 0 \\
0 & -1 & 0
\end{bmatrix}
= 
\begin{bmatrix}
0 & 0 & 0 \\
0 & 0 & 1 \\
0 & 0 & 0 \\
0 & -1 & 0
\end{bmatrix}
= 
\begin{bmatrix}
0 & 0 & 0 \\
0 & 0 & 1 \\
0 & 0 & 0 \\
0 & 0 & 1
\end{bmatrix},
0 = 0
\]

(48)
To avoid comparing by eye, one can always take the left-hand side minus the right-hand side, resulting in everything equal to 0

\[
\begin{bmatrix}
0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 \\
0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0
\end{bmatrix}
= 
\begin{bmatrix}
0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 \\
0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0
\end{bmatrix}
, 
0 = 0
\]

\[
\text{TensorArray}((\text{lhs} - \text{rhs})((36)), \text{mm})
\]

\[
\text{* Partial match of 'mm' against keyword 'performmatrixoperations'}
\]

\[
\begin{bmatrix}
0 \\
0 \\
\vdots \\
0
\end{bmatrix}
\begin{bmatrix}
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0
\end{bmatrix}
= 
\begin{bmatrix}
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0
\end{bmatrix}
(49)
\]

The \( \mathbf{L}^\mu_\nu \) tensor is now expressed in terms of these objects as

\[
\%LM[\sim{\mu, \nu}]=\omega[i] \cdot S[i] + \zeta[i] \cdot K[i]
\]

\[
\mathbf{L}^\mu_\nu = \zeta[i] K^i + \omega[i] S^i
\]

(50)

where the right-hand side, without free indices, represents the matrix form of \( \mathbf{L}^\mu_\nu \). This notation makes explicit the fact that any Lorentz transformation can always be written as the composition of six rotations \( \text{SumOverRepeatedIndices} ((50)) \)

\[
\]

(51)

\text{Library:}-\text{RewriteInMatrixForm} ((51))
\[
\mathbf{L}^\mu_v = \begin{bmatrix}
0 & \zeta_1 & 0 & 0 \\
\zeta_1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0
\end{bmatrix} + \begin{bmatrix}
0 & 0 & \zeta_2 & 0 \\
0 & 0 & 0 & 0 \\
\zeta_2 & 0 & 0 & 0 \\
0 & 0 & 0 & 0
\end{bmatrix} + \begin{bmatrix}
0 & 0 & 0 & \zeta_3 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
\zeta_3 & 0 & 0 & 0
\end{bmatrix} + \begin{bmatrix}
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & -\omega_i & 0 \\
0 & 0 & 0 & 0
\end{bmatrix}
\]

\[
\mathbf{L}^\mu_v = \begin{bmatrix}
0 & \zeta_1 & \zeta_2 & \zeta_3 \\
\zeta_1 & 0 & -\omega_3 & \omega_2 \\
\zeta_2 & \omega_3 & 0 & -\omega_1 \\
\zeta_3 & -\omega_2 & \omega_1 & 0
\end{bmatrix}
\]

which is the same as the starting point (30)

**The transformation** \(\mathbf{L}^\mu_v = \exp \left( \mathbf{L}^\mu_v \right)\), where \(\mathbf{L}^\mu_v = \zeta_i K^i\), as a function of the relative velocity of two inertial systems

As seen in the previous subsection, in \(\mathbf{L}^\mu_v = \zeta_i K^i + \omega_i S^i\), the second term, \(\omega_i S^i\), corresponds to 3D rotations embedded in the general form of 4D Lorentz transformations, and \(\zeta_i K^i\) is the term that relates the coordinates of two inertial systems of reference that move with respect to each other at constant velocity \(v\). In this section, \(\zeta_i K^i\) is rewritten in terms of that velocity, arriving at equation (11.98) of Jackson's book [1]. The key observation is that the 3D vector \(\zeta_j\), can be rewritten in terms of \(\tanh^{-1}(\beta)\), where \(\beta = \frac{v}{c}\) and \(c\) is the velocity of light (for the rationale of that relation, see [2], sec 4, discussion before formula (4.3)).

Use a macro - say \(ub\) - to represent the atomic variable \(\hat{\beta}\) (this variable can be entered as `\#mover(mi("&beta;"),mo("&circ;")`). In general, to create atomic variables, see the section on Atomic Variables of the page 2DMathDetails).

\[
\text{macro}(ub = \hat{\beta}) : \quad ub[j] = [ub[1], ub[2], ub[3]], \quad \zeta[j] = ub[j] \cdot \tanh^{-1}(\beta)
\]
\begin{equation}
\hat{\beta}_j = [\hat{\beta}_1, \hat{\beta}_2, \hat{\beta}_3], \zeta_j = \hat{\beta}_j \text{arctanh}(\beta)
\end{equation}

Define (54)

Defined objects with tensor properties

\begin{equation}
\{\Lambda, \xi^\mu_{\mu'}, \gamma_{\mu', \nu}, K^i_{\mu}, \sigma^i_{\mu}, S^i_{\rho}, \zeta_i, \partial^i_{\mu}, g_{\mu', \nu, \rho}, \gamma_{a, b}, \omega_i, \epsilon_{\alpha, \beta, \mu, \nu}, \hat{\beta}_j\}
\end{equation}

With these two definitions, and excluding the rotation term $\omega_i S^i$, we have

\[\%LM[\sim \mu, \nu] = \zeta[j] K[j]\]

\[\mathbb{L}^\mu_{\nu} = \zeta_j K^j\]

SumOverRepeatedIndices (56)

\[\mathbb{L}^\mu_{\nu} = \text{arctanh}(\beta) \left( K^1 \hat{\beta}_1 + K^2 \hat{\beta}_2 + K^3 \hat{\beta}_3 \right)\]

Library:-PerformMatrixOperations (57)

\[\mathbb{L}^\mu_{\nu} = \begin{bmatrix}
0 & \beta_1 \text{arctanh}(\beta) & \beta_2 \text{arctanh}(\beta) & \beta_3 \text{arctanh}(\beta) \\
\beta_1 \text{arctanh}(\beta) & 0 & 0 & 0 \\
\beta_2 \text{arctanh}(\beta) & 0 & 0 & 0 \\
\beta_3 \text{arctanh}(\beta) & 0 & 0 & 0
\end{bmatrix}\]

From this expression, the form of $\Lambda^\mu_{\nu}$ can be obtained as in (24) using

\[\text{LinearAlgebra:-MatrixExponential}\] and simplifying the result taking into account that $\hat{\beta}_j$ is a unit vector

SumOverRepeatedIndices ($ub[j]^2$) = 1

\[\hat{\beta}_1^2 + \hat{\beta}_2^2 + \hat{\beta}_3^2 = 1\]

\[\exp(\text{lhs (58)}) = \text{simplify(LinearAlgebra:-MatrixExponential(rhs (58))), (59))}\]

\[e^{\mathbb{L}^\mu_{\nu}} = \begin{bmatrix}
\frac{1}{\sqrt{-\beta^2 + 1}}, & \frac{\beta_1}{\sqrt{-\beta^2 + 1}}, & \frac{\beta_2}{\sqrt{-\beta^2 + 1}}, & \frac{\beta_3}{\sqrt{-\beta^2 + 1}} \\
\frac{\hat{\beta}_1 \beta}{\sqrt{-\beta^2 + 1}}, & -\frac{\sqrt{-\beta^2 + 1}}{\sqrt{-\beta^2 + 1}}, & -\frac{\hat{\beta}_1^2 - \sqrt{-\beta^2 + 1}}{\sqrt{-\beta^2 + 1}}, & -\frac{\beta_1 \hat{\beta}_2}{\sqrt{-\beta^2 + 1}} \left( \frac{\sqrt{-\beta^2 + 1}}{\sqrt{-\beta^2 + 1}} - 1 \right) \\
\frac{\hat{\beta}_1 \hat{\beta}_3}{\sqrt{-\beta^2 + 1}}, & -\frac{\hat{\beta}_2}{\sqrt{-\beta^2 + 1}}, & -\frac{\sqrt{-\beta^2 + 1}}{\sqrt{-\beta^2 + 1}}, & -\frac{\hat{\beta}_1 \hat{\beta}_2}{\sqrt{-\beta^2 + 1}} \left( \frac{\sqrt{-\beta^2 + 1}}{\sqrt{-\beta^2 + 1}} - 1 \right) \\
\frac{\beta_2 \beta}{\sqrt{-\beta^2 + 1}}, & -\frac{\hat{\beta}_1 \beta}{\sqrt{-\beta^2 + 1}}, & -\frac{\sqrt{-\beta^2 + 1}}{\sqrt{-\beta^2 + 1}}, & -\frac{\beta_2 - \hat{\beta}_2}{\sqrt{-\beta^2 + 1}} \left( \frac{\sqrt{-\beta^2 + 1}}{\sqrt{-\beta^2 + 1}} - 1 \right)
\end{bmatrix}\]
\[
- \frac{\hat{\beta}_2 \hat{\beta}_3 \left( \sqrt{-\beta^2 + 1} - 1 \right)}{\sqrt{-\beta^2 + 1}},
\frac{\hat{\beta}_3 \beta - \hat{\beta}_1 \hat{\beta}_3 \left( \sqrt{-\beta^2 + 1} - 1 \right)}{\sqrt{-\beta^2 + 1}},
- \frac{\hat{\beta}_2 \hat{\beta}_3 \left( \sqrt{-\beta^2 + 1} - 1 \right)}{\sqrt{-\beta^2 + 1}},
\frac{(\hat{\beta}_1^2 + \hat{\beta}_2^2) \sqrt{-\beta^2 + 1} - \hat{\beta}_1^2 - \hat{\beta}_2^2 + 1}{\sqrt{-\beta^2 + 1}}
\]

It is useful at this point to analyze the dependency on the components of \( \hat{\beta}_j \) of this matrix

\[
map(u \rightarrow \text{indets}(u, \text{specindex}(ub)), \text{rhs}(60))
\]

\[
\begin{array}{cccc}
\emptyset & \{\hat{\beta}_1\} & \{\hat{\beta}_2\} & \{\hat{\beta}_3\} \\
\{\hat{\beta}_1\} & \{\hat{\beta}_1\} & \{\hat{\beta}_1, \hat{\beta}_2\} & \{\hat{\beta}_1, \hat{\beta}_3\} \\
\{\hat{\beta}_2\} & \{\hat{\beta}_1, \hat{\beta}_2\} & \{\hat{\beta}_2\} & \{\hat{\beta}_2, \hat{\beta}_3\} \\
\{\hat{\beta}_3\} & \{\hat{\beta}_1, \hat{\beta}_3\} & \{\hat{\beta}_2, \hat{\beta}_3\} & \{\hat{\beta}_1, \hat{\beta}_2\}
\end{array}
\]

(61)

We see that the diagonal element [4, 4] depends on two instead of only one component of \( \hat{\beta}_j \). That is due to the simplification with respect to side relations, performed in (60), that constructs an elimination Groebner Basis that cannot reduce at once, using the single equation (59), the dependency of all of the elements [2, 2], [3, 3] and [4, 4] to a single component of \( \hat{\beta}_j \). So, to reduce further the dependency of the [4, 4] element, this component of (60) requires one more simplification step, using a different elimination strategy, explicitly requesting the elimination of \( \{\hat{\beta}_1, \hat{\beta}_2\} \)

\[
\text{rhs}(60)[4, 4]
\]

\[
\frac{(\hat{\beta}_1^2 + \hat{\beta}_2^2) \sqrt{-\beta^2 + 1} - \hat{\beta}_1^2 - \hat{\beta}_2^2 + 1}{\sqrt{-\beta^2 + 1}}
\]

(62)

\[
simplify(62, \{59\}, \{\hat{\beta}_1, \hat{\beta}_2\})
\]

\[
-\sqrt{-\beta^2 + 1} \hat{\beta}_3 + \sqrt{-\beta^2 + 1}
\]

(63)

This result involves only \( \hat{\beta}_3 \), and with it the form of \( \Lambda^\mu_v = \exp(\mathcal{L}^\mu_v) \) becomes

\[
\text{subs}(62) = (63), (60)
\]

\[
\mathcal{L}^\mu_v = \begin{bmatrix}
\frac{1}{\sqrt{-\beta^2 + 1}}, & \hat{\beta}_1 \beta, & \hat{\beta}_2 \beta, & \hat{\beta}_3 \beta
\end{bmatrix}
\]

(64)
Replacing now the components of the unit vector $\hat{\beta}_j$ by the components of the vector $\vec{\beta}$ divided by its modulus $\beta$

$$seq\left(ub[j]=\frac{\beta[j]}{\beta}, j=1..3\right)$$

$$\hat{\beta}_1 = \frac{\beta_1}{\beta}, \hat{\beta}_2 = \frac{\beta_2}{\beta}, \hat{\beta}_3 = \frac{\beta_3}{\beta}$$

(65)

and recalling that

$$e^{-\mu, v} = \Lambda_{\mu}^{\frac{\mu}{v}}$$

(66)

to get equation (11.98) in Jackson's book it suffices to introduce (the customary notation)

$$\frac{1}{\sqrt{1 - \beta^2}} = \gamma$$

$$\frac{1}{\sqrt{-\beta^2 + 1}} = \gamma$$

(67)

simplify\left(subs\left( (65), (66), (67), (67)^{-1}, (64) \right) \right)
\[
\Lambda^\mu_v = \begin{bmatrix}
\gamma & \beta_1 \gamma & \beta_2 \gamma & \beta_3 \gamma \\
\beta_1 \gamma & \frac{1}{\beta^2} (1 + \gamma) \beta_1^2 + \beta_2^2 & \frac{1}{\beta^2} (1 + \gamma) \beta_2^2 & \frac{1}{\beta^2} (1 + \gamma) \beta_3^2 \\
\beta_2 \gamma & \frac{1}{\beta^2} (1 + \gamma) \beta_1^2 & \frac{1}{\beta^2} (1 + \gamma) \beta_2^2 + \beta_3^2 & \frac{1}{\beta^2} (1 + \gamma) \beta_3^2 \\
\beta_3 \gamma & \frac{1}{\beta^2} (1 + \gamma) \beta_1^2 & \frac{1}{\beta^2} (1 + \gamma) \beta_2^2 & \frac{1}{\beta^2} (1 + \gamma) \beta_3^2 + \beta_3^2 
\end{bmatrix}
\]

(68)

This is equation (11.98) in Jackson's book.

Finally, to get the form of this general Lorentz transformation excluding 3D rotations, directly expressed in terms of the relative velocity \( v \) of the two inertial systems of references, introduce

\[
v[i] = [v_x, v_y, v_z], \quad B[i] = \frac{v[i]}{c}
\]

At this point it suffices to Define (69) as tensors

\[
\text{Define}(69)
\]

**Defined objects with tensor properties**

\[
\{\Lambda, \Lambda_{\mu \nu}, v, y, K, \sigma, S, \xi, \beta, \partial, g, a, b, \omega, \gamma, \nu, \epsilon, \alpha, \beta, \mu, v, \hat{\beta}\}
\]

and remove \( \beta \) and \( \gamma \) from the formulation using

\[
\text{(rhs = lhs)((67)), } \beta = \frac{v}{c}
\]

\[
\gamma = \frac{1}{\sqrt{-c^2 \beta^2 + 1}}, \quad \beta = \frac{v}{c}
\]

(71)

\[
\text{simplify(subs((71), simplify((68)))), size)
\]

\[
\Lambda^\mu_v = \begin{bmatrix}
\frac{1}{\sqrt{-v^2 + 1}}, & \frac{v_x}{c \sqrt{-v^2 + 1}}, & \frac{v_y}{c \sqrt{-v^2 + 1}}, & \frac{v_z}{c \sqrt{-v^2 + 1}} \\
\end{bmatrix}
\]

(72)
References