Mixing parameter tools

Internal documentation

S. Antusch, J. Kersten, M. Lindner, M. Ratz and M. Schmidt

December 5, 2008

This is an internal documentation of the `MixingParameterTools` add-on for Mathematica. We describe the functions which allow to extract the mixing parameters from mass and Yukawa matrices in some detail. There is also some information concerning the installation.

Contents

1 Short description and comments

This is the documentation of the `MixingParametersTools` add-on. It contains the `MPT3x3.m` package which provides various tools allowing for the extraction of physical parameters from mass and Yukawa matrices.

Note that we do not adopt the naming conventions of the related `SolveNeutrinoRGEs` add-on since the present add-on is intended to be a 'stand-alone' application, i.e. one may use it without loading the full set of `RGE...` packages. This is because it might be useful in order to study textures without running, and it is not bound to be applied to the analysis of neutrino masses only but may be used for quark and superpartner mass matrices as well.

The `MixingParameterTools` add-on is meant to replace the `ExtractMixingAngles.m` package. It provides tools to extract mixing parameters relating 3×3 mass matrices. It offers the treatment of both Dirac and Majorana neutrino masses. In addition, functions evaluating quark mixing parameters are also implemented.

2 Installation

2.1 UNIX/Linux

(i) Automatic installation

Unpack the archive MixingParameterTools.tar.bz2.

```
tar -xvjf MixingParameterTools.tar.bz2
```

Then go to the directory MixingParameterTools and execute the script install.sh in this directory. The script copies the Mathematica package to the .Mathematica folder in your home directory.

```
cd MixingParameterTools
./install.sh
```

(ii) Installation by hand

In order to install the package(s), one has to switch to the directory where the add-ons are, e.g.

```
cd .Mathematica/Applications/
```

and create a directory for the mixing parameter tools:

```
mkdir MixingParameterTools/
```

Then one has to move the `MPT3x3.m` package to the new directory:

```
cd
mv MPT3x3.m .Mathematica/Applications/MixingParameterTools/
```

3 Functions

We divide the functions into two classes:

- 'public' functions which will be explicitly mentioned in the publication, and
- 'private' functions which are useful, but have not necessarily to be invoked explicitly in order to extract the running mixing parameters.

Note that the terminology 'public' or 'private' is in quotation because it does not correspond to what is public or private in the context of Mathematica packages. The aim is to present only the absolutely necessary functions in the publication so that the effort for maintenance is as low as possible. These are the 'public' functions. The 'private' functions are those which will be useful in several applications for us, but where there is no need to tell the rest of the world that they exist (people familiar with Mathematica will notice their existence anyway).

`MNSMatrix` ('public')

`MNSMatrix[`m, Y_e`]` returns the MNS matrix, i.e. the matrix U_{MNS} which diagonalizes the (neutrino mass) matrix m in the basis where the (charged lepton Yukawa coupling) matrix Y_e is diagonal. By convention, the parameters of U_{MNS} fulfill $0 \leq \theta_{12} \leq \pi/4$, $0 \leq \theta_{13}, \theta_{23} \leq \pi/2$ and all other parameters range from 0 to 2π. It is possible to fix the hierarchy to be inverted by calling `MNSMatrix[`m, Y_e`,‘‘i’’]`. Note that the input matrices m and Y_e must be numeric.

`MNSParameters` ('public')

`MNSParameters[`m, Y_e`]` returns the MNS mixing and mass parameters $\{\{\theta_{12}, \theta_{13}, \theta_{23}, \delta, \delta_e, \delta_\mu, \delta_\tau, \varphi_1$
for a Majorana neutrino matrix m and a Yukawa coupling matrix Y_e. The returned parameters obey the conventions $0 \leq \theta_{12} \leq \pi/4$, $0 \leq \theta_{13}, \theta_{23} \leq \pi/2$ and all other parameters range from 0 to 2π. It is possible to fix an inverted hierarchy to be inverted by calling `MNSParameters[`m, Y_e`,‘‘i’’]`. Note that the input matrices m and Y_e must be numeric.

`DiracMNSMatrix` ('public')

`DiracMNSMatrix[`Y_ν, Y_e`]` returns the MNS matrix for Dirac neutrinos with Yukawa coupling Y_ν. It is the inverse of `CKMMatrix[`Y_ν, Y_e`]` (see below).

`DiracMNSParameters` ('public')

`DiracMNSParameters[`Y_ν, Y_e`]` returns the MNS mixing parameters $\{\theta_{12}, \theta_{13}, \theta_{23}, \delta\}$, $\{y_1, y_2, y_3\}$ (with $y_i = m_i/v$) and $\{y_e, y_\mu, y_\tau\}$ for neutrino and charged lepton Yukawa matrices Y_ν and Y_e. Note that these parameters are not sufficient to determine the unitary matrix which diagonalizes $Y_\nu^\dagger Y_\nu$ in the basis where $Y_e^\dagger Y_e$ is diagonal. The additional parameters, required to reconstruct $U_{\mathrm{MNS}}^{\mathrm{Dirac}}$, are unphysical.

`CKMMatrix` (‘public’)

`CKMMatrix[`Y_u, Y_d`]` returns the MNS matrix, i.e. the matrix U_{CKM} which diagonalizes the (down-type quark Yukawa) matrix Y_d in the basis where the (up-type quark Yukawa) matrix Y_u is diagonal. Note that the input matrices Y_u and Y_d must be numeric. Note also that this function can be used to extract the neutrino mixing parameters in the case of Dirac neutrinos (see above).

`CKMParameters` (‘public’)

`CKMParameters[`Y_u, Y_d`]` returns the CKM mixing parameters $\{\theta_{12}, \theta_{13}, \theta_{23}, \delta\}$, as well as the Yukawa couplings $\{y_u, y_c, y_t\}$ and $\{y_d, y_s, y_b\}$, for up- and down-type Yukawa matrices Y_u and Y_d. Note that these parameters are not sufficient to determine the unitary matrix which diagonalizes $Y_d^\dagger Y_d$ in the basis where $Y_u^\dagger Y_u$ is diagonal. The additional parameters, required to reconstruct U_{CKM}, are unphysical.

`CKMReplacementRules` (‘private’)

`CKMReplacementRules[`Y_u, Y_d`]` returns a list of replacement rules which replaces the quark parameter by the numerical value calculated from Y_u and Y_d.

`MPT3x3OrthogonalMatrix` (‘private’)

`MPT3x3OrthogonalMatrix[`$\theta_{12}, \theta_{13}, \theta_{23}, \delta$`]` returns the standard parametrized matrix

$$V = \begin{pmatrix} c_{12}c_{13} & s_{12}c_{13} & s_{13}e^{-\mathrm{i}\delta} \\ -c_{23}s_{12} - s_{23}s_{13}c_{12}e^{\mathrm{i}\delta} & c_{23}c_{12} - s_{23}s_{13}s_{12}e^{\mathrm{i}\delta} & s_{23}c_{13} \\ s_{23}s_{12} - c_{23}s_{13}c_{12}e^{\mathrm{i}\delta} & -s_{23}c_{12} - c_{23}s_{13}s_{12}e^{\mathrm{i}\delta} & c_{23}c_{13} \end{pmatrix}, \tag{3.1}$$

with c_{ij} and s_{ij} defined as $\cos\theta_{ij}$ and $\sin\theta_{ij}$, respectively. If δ is omitted, V with $\delta = 0$, i.e. a really orthogonal matrix, is returned.

`MPT3x3UnitaryMatrix` (‘private’)

`MPT3x3UnitaryMatrix[` $\theta_{12}, \theta_{13}, \theta_{23}, \delta, \delta_e, \delta_\mu, \delta_\tau, \varphi_1, \varphi_2$`]` returns the unitary matrix in standard parametrization

$$U = \mathrm{diag}(e^{\mathrm{i}\delta_e}, e^{\mathrm{i}\delta_\mu}, e^{\mathrm{i}\delta_\tau}) \cdot V \cdot \mathrm{diag}(e^{-\mathrm{i}\varphi_1/2}, e^{-\mathrm{i}\varphi_2/2}, 1) \tag{3.2}$$

with V being defined by Eq. (3.1). Instead of 9 arguments one can invoke `StandardUnitaryMatrix` with a list as argument. The entries of the list are then interpreted as mixing parameters $\{\theta_{12}, \theta_{13}, \theta_{23}, \delta, \delta_e, \delta_\mu, \delta_\tau, \varphi_1, \varphi_2\}$, and if the length of the list is less than 9, the missing parameters are interpreted as 0. If the list is longer than 9, the over-abundant entries are ignored.

`MPT3x3MixingMatrixL` (‘private’)

`MPT3x3MixingMatrixL[`M`,`*`options`*`]` returns the matrix U_L which is used for diagonalizing a general complex matrix M, i.e.

$$U_\mathrm{R}^\dagger\, M\, U_\mathrm{L} \;=\; \mathrm{diag}(M_1, M_2, M_3)\;, \tag{3.3}$$

with $M_1 \le M_2 \le M_3$. U_L is one representative of the matrices being suitable for fulfilling (3.3). If the ‘eigenvalues’ M_i are non-degenerate, different representatives are related by multiplication with unphysical phases. Note that in the case of degenerate M_i mixing parameters which are otherwise physical become unphysical. The label ‘L’ refers to the fact that this matrix rotates left-handed fields rather than the position in the diagonalization formula (3.3). The option is *`MPTTolerance`* which has as default value 10^{-6}. It controls the degree by which two numbers can disagree and still be considered equal. Depending on the agreement, the diagonalization process is interpreted to be successful or not.

`MPT3x3MixingMatrixR` (‘private’)

`MPT3x3MixingMatrixR[`M`,`*`options`*`]` returns the corresponding matrix U_R (see above). The label ‘R’ refers to the fact that this matrix rotates right-handed fields rather than the position in the diagonalization formula (3.3). The option is *`MPTTolerance`* which has as default value 10^{-6}. It controls the degree by which two numbers can disagree and still be considered equal. Depending on the agreement, the diagonalization process is interpreted to be successful or not.

`MPT3x3NeutrinoMixingMatrix` (‘private’)

`MPT3x3NeutrinoMixingMatrix[`m`,`*`MPTTolerance`*`]` returns the matrix U which is used for diagonalizing a general complex symmetric Matrix m, i.e.

$$U^T\, m\, U \;=\; \mathrm{diag}(m_1, m_2, m_3)\;, \tag{3.4}$$

and $|\Delta m^2_{32}| \ge |\Delta m^2_{21}|$ (with the usual definitions of Δm^2_{ij}) and $U_{12} \le U_{11}$.[1]
`MPT3x3NeutrinoMixingMatrix[`m, S`,`*`MPTTolerance`*`]` does the same only that the mass hierarchy can be fixed to be inverted by setting $S =$ “i”. Any other S leads to a **fixed** regular mass hierarchy. The option is *`MPTTolerance`* which has as default value 10^{-6}. It controls the degree by which two numbers can disagree and still be considered equal. Depending on the agreement, the diagonalization process is interpreted to be successful or not.

[1] The last relation is convention. It is implemented in order to read off $\theta_{12} \le \pi/4$ finally.

`MPT3x3MixingParameters` ('private')

`MPT3x3MixingParameters[`U`]` inverts (3.2) for a given unitary U. More specifically, `MPT3x3MixingParameters[`U`]` returns the mixing angles and phases θ_{12}, θ_{13}, θ_{23}, δ, δ_e, δ_μ, δ_τ, φ_1 and φ_2 of U in standard parametrization where $0 \leq \theta_{ij} \leq \pi/2$ holds.
Note: In order to reproduce an arbitrary unitary matrix, one must allow the 'Majorana phases' φ_i to range from 0 to 4π (rather than 2π). However, physically Majorana phases differing by 2π are ambiguous. That is, if a unitary matrix with a certain value of φ_i diagonalizes a symmetric mass matrix, the unitary matrix with $\varphi_i + 2\pi$ also does the job. As the function `MPT3x3MixingParameters` is intended to invert `MPT3x3UnitaryMatrix`, it returns the mathematical rather than the physical parameters.

> In some special cases, the mixing parameters are not defined uniquely. In this case, a warning is printed and one set of possible parameters is returned. As a general rule, as many of the physical parameters as possible are set to zero. More details can be found in sec. 5.4.

4 Remarks

4.1 Remarks on the calculation of the CKM matrix

The input parameters are the Yukawa couplings $Y = (Y_{fg})$ (Y_u and Y_d) which are defined via the Lagrangean

$$\mathscr{L}_{\text{Yukawa}} = \overline{\psi^f_{\text{R}}}\, Y_{fg}\, \psi^g_{\text{L}} + \text{h.c.}\ , \tag{4.1}$$

with R and L indicating right- and left-chiral fields, respectively. Y can always be diagonalized by a bi-unitary transformation

$$\begin{aligned} \psi_{\text{R}} &\to U_{\text{R}}^\dagger\, \psi_{\text{R}}\ , && (4.2\text{a}) \\ \psi_{\text{L}} &\to U_{\text{L}}^\dagger\, \psi_{\text{L}}\ , && (4.2\text{b}) \\ Y &\to U_{\text{R}}^\dagger\, Y\, U_{\text{L}} = \operatorname{diag}(y_1, y_2, y_3)\ , && (4.2\text{c}) \end{aligned}$$

with $y_1 \leq y_2 \leq y_3$ being the 'eigenvalues' of Y. Here, U_{L} and U_{R} are defined (or: can be computed) via

$$\begin{aligned} U_{\text{L}}^\dagger\, Y^\dagger\, Y\, U_{\text{L}} &\stackrel{!}{=} \operatorname{diag}\left(|y_1|^2, |y_2|^2, |y_3|^2\right)\ , && (4.3\text{a}) \\ U_{\text{R}}^\dagger\, Y\, Y^\dagger\, U_{\text{R}} &\stackrel{!}{=} \operatorname{diag}\left(|y_1|^2, |y_2|^2, |y_3|^2\right)\ , && (4.3\text{b}) \end{aligned}$$

respectively. For most applications, U_{R} is irrelevant.
The CKM matrix is calculated as follows:

(1) Switch to the basis where Y_u is diagonal, i.e.

$$\begin{aligned} Y_u &\to (U_{\text{R}}^{(u)})^\dagger\, Y_u\, U_{\text{L}}^{(u)} = \operatorname{diag}(y_u, y_c, y_t)\ , && (4.4\text{a}) \\ Y_d &\to (U_{\text{R}}^{(u)})^\dagger\, Y_d\, U_{\text{L}}^{(u)} =: Y_d'\ . && (4.4\text{b}) \end{aligned}$$

(2) Calculate U_{L} for Y_d'. This is U_{CKM}.

4.2 Remarks on the calculation of the MNS matrix

For the MNS matrix, switch to the basis where Y_e is diagonal,

$$Y_e \quad\to\quad U_\mathrm{R}^\dagger\, Y_e\, U_\mathrm{L} \;=\; \mathrm{diag}(y_e, y_\mu, y_\tau)\;, \tag{4.5a}$$

$$m_\nu \quad\to\quad U_\mathrm{L}^T\, m_\nu\, U_\mathrm{L} \;=:\; m'_\nu\;. \tag{4.5b}$$

The MNS matrix has to fulfill

$$U_\mathrm{MNS}^T\, m'_\nu\, U_\mathrm{MNS} \;=\; \mathrm{diag}(m_1, m_2, m_3)\;, \tag{4.6}$$

where the m_i are real and positive. However, this does not fix U_MNS entirely. First of all, there is the obvious ambiguity of ordering the mass eigenvalues m_i. In order to obtain a mixing matrix which can be compared with the experimental data, the choice of the prescription is somewhat subtle. From experiment we know that there is a small mass difference, called $\Delta m^2_\mathrm{sol} = m_i^2 - m_j^2$, and a larger one, referred to as $\Delta m^2_\mathrm{atm} = m_k^2 - m_\ell^2$. By convention, the masses are labeled such that $i, j \neq 3$ while either k or ℓ equals 3. The mass label 2 is attached to the eigenvector with the lower modulus of the first component. We are doing this since we want to read off a mixing angle θ_{12} less than $45°$. If it then turns out that $m_1 > m_2$, the corresponding mass matrix is most likely not physical.

5 Definition and Extraction of Mixing Parameters

5.1 Standard Parametrization

In this section we describe our conventions and how mixing angles and phases can be extracted from mass matrices. For a general unitary matrix we choose the so-called standard parametrization

$$U \;=\; \mathrm{diag}(e^{\mathrm{i}\delta_e}, e^{\mathrm{i}\delta_\mu}, e^{\mathrm{i}\delta_\tau}) \cdot V \cdot \mathrm{diag}(e^{-\mathrm{i}\varphi_1/2}, e^{-\mathrm{i}\varphi_2/2}, 1) \;=:\; K_\delta \cdot V \cdot K_\varphi\;, \tag{5.1}$$

where

$$V = \begin{pmatrix} c_{12}c_{13} & s_{12}c_{13} & s_{13}e^{-\mathrm{i}\delta} \\ -c_{23}s_{12} - s_{23}s_{13}c_{12}e^{\mathrm{i}\delta} & c_{23}c_{12} - s_{23}s_{13}s_{12}e^{\mathrm{i}\delta} & s_{23}c_{13} \\ s_{23}s_{12} - c_{23}s_{13}c_{12}e^{\mathrm{i}\delta} & -s_{23}c_{12} - c_{23}s_{13}s_{12}e^{\mathrm{i}\delta} & c_{23}c_{13} \end{pmatrix} \tag{5.2}$$

with c_{ij} and s_{ij} defined as $\cos\theta_{ij}$ and $\sin\theta_{ij}$, respectively.

5.2 Physical vs. unphysical parameters

In the literature, the following parameters in standard parametrization (cf. Eq. (5.1)) are called 'physical':

$$\{\theta_{12}, \theta_{13}, \theta_{23}, \delta\} \quad:\quad \text{for the quark sector,} \tag{5.3a}$$

$$\{\theta_{12}, \theta_{13}, \theta_{23}, \delta, \varphi_1, \varphi_2\} \quad:\quad \text{for the lepton sector.} \tag{5.3b}$$

The parameters $\{\theta_{12}, \theta_{13}, \theta_{23}, \delta\}$ for the quarks are measurable in weak processes, i.e. in processes where $W^\pm$ is involved. The parameters $\{\theta_{12}, \theta_{13}, \theta_{23}, \delta\}$ for the leptons are accessible in neutrino oscillations, in addition a combination of φ_1 and φ_2 can, in principle, be determined by neutrinoless double β decay experiments if neutrino masses are Majorana.

Let us work in the basis where $Y_e^\dagger Y_e$ is diagonal. Then

$$m_\nu \;=\; U_{\mathrm{MNS}}^* \,\mathrm{diag}(m_1, m_2, m_3)\, U_{\mathrm{MNS}}^\dagger \;, \tag{5.4}$$

i.e. the 12 parameters of m_ν can be decomposed into 3 eigenvalues, 6 physical rotations and 3 unphysical rotations according to the classification of the literature. In other words, mass matrices m_ν and m_ν' are equivalent if there exists a matrix $K_\delta = \mathrm{diag}(e^{\mathrm{i}\delta_e}, e^{\mathrm{i}\delta_\mu}, e^{\mathrm{i}\delta_\tau})$ such that

$$m_\nu' \;=\; K_\delta^*\, m_\nu\, K_\delta^\dagger \;=\; K_\delta^{-1}\, m_\nu\, K_\delta^{-1} \;. \tag{5.5}$$

For instance, we can always choose m_ν such that the δ_f phases vanish.

Above the see-saw scales, we have to deal in addition with $Y_\nu^\dagger Y_\nu$ which can be written as

$$Y_\nu^\dagger Y_\nu \;=\; U_\nu^\dagger \,\mathrm{diag}(y_1^2, y_2^2, y_3^2)\, U_\nu \;, \tag{5.6}$$

i.e. it can be parametrized by three eigenvalues and 6 rotation parameters (the parameters φ_i of the standard parametrization and an overall phase are obviously redundant). The crucial observation is now that a transformation between equivalent effective mass matrices,

$$m_\nu \;\to\; K_\delta\, m_\nu\, K_\delta \;, \tag{5.7}$$

which can be interpreted as a rotation of the lepton doublets according to

$$\vec{\ell} \to K_\delta^{-1}\, \vec{\ell}\,, \tag{5.8}$$

(obviously) implies a transformation of $Y_\nu^\dagger Y_\nu$,

$$Y_\nu^\dagger Y_\nu \;\to\; K_\delta^{-1}\, Y_\nu^\dagger Y_\nu\, K_\delta \;. \tag{5.9}$$

The set of transformations (5.7), (5.8) and (5.9) corresponds to a symmetry of the theory, and leaves in particular the RGEs invariant.

The transformation (5.7) (or (5.9)) alone, however, is **not** a symmetry. That is to say that changing the unphysical phases without changing the phases of the off-diagonal elements of $Y_\nu^\dagger Y_\nu$ (or vice versa) leads to a different model,[2] i.e. different measurable parameters (at low energies). In order to define a model, we hence need to specify both the δ phases and the off-diagonal elements of $Y_\nu^\dagger Y_\nu$ at the same time. In other words, due to the running, two models (in the 'full' theory) with the same parameters except for the

[2] It would be interesting to check if there is a transformation in M which corresponds to this change of the effective neutrino mass operatore alone.

δ_f phases do not describe the same physics. Hence, the δ_f phases are not 'unphysical' in the see-saw model, but they can always be traded for parameters of $Y_\nu^\dagger Y_\nu$.

One may wonder whether a similar thing occurs in the quark sector. This is because the physical objects are $Y_u^\dagger Y_u$ and $Y_d^\dagger Y_d$, and there is no physical object analogous to the effective neurino mass operator. In the basis where $Y_u^\dagger Y_u$ is diagonal, we can make the off-diagonal elements of $Y_d^\dagger Y_d$ real and positive by the transformation

$$\vec{q} \to K_\delta^{-1}\,\vec{q}\,, \tag{5.10}$$

which does not change $Y_u^\dagger Y_u$. This transformation does not change the physical parameters but only the unphysical ones. Because it is a symmetry if we impose simultaneously

$$Y_u \to Y_u\,K_\delta\,, \quad Y_d \to Y_d\,K_\delta\,, \tag{5.11}$$

the RGEs have to be invariant under it. This means that to just change the phases of the off-diagonal elements of $Y_d^\dagger Y_d$ without changing $Y_u^\dagger Y_u$ is a symmetry of the theory, hence the RGEs in the quark sector neither depend on the phases of the off-diagonal elements of $Y_d^\dagger Y_d$ nor the unphysical phases.

5.3 Extracting Mixing Angles and Phases

In the standard parametrization, the mixing angles θ_{13} and θ_{23} can be chosen to lie between 0 and $\frac{\pi}{2}$, and in the lepton sector by reordering the masses, θ_{12} can be restricted to $0 \le \theta_{12} \le \frac{\pi}{4}$. For the phases the range between 0 and 2π is required. In order to read off the mixing parameters in the generic case, i.e. for none of the angles θ_{ij} equal to 0 or $\pi/2$, we use the following procedure:

(1) $\theta_{13} = \arcsin(|U_{13}|)$.

(2) $\theta_{12} = \begin{cases} \arctan\left(\frac{|U_{12}|}{|U_{11}|}\right) & \text{if } U_{11} \neq 0 \\ \frac{\pi}{2} & \text{else} \end{cases}$

(3) $\theta_{23} = \begin{cases} \arctan\left(\frac{|U_{23}|}{|U_{33}|}\right) & \text{if } U_{33} \neq 0 \\ \frac{\pi}{2} & \text{else} \end{cases}$

(4) $\delta_\mu = \arg(U_{23})$

(5) $\delta_\tau = \arg(U_{33})$

(6) $\delta = -\arg\left(\dfrac{\dfrac{U_{11}^* U_{13} U_{31} U_{33}^*}{c_{12}\,c_{13}^2\,c_{23}\,s_{13}} + c_{12}\,c_{23}\,s_{13}}{s_{12}\,s_{23}}\right)$.

(7) $\delta_e = \arg(e^{\mathrm{i}\delta}\,U_{13})$

(8) $\varphi_1 = 2\arg(e^{\mathrm{i}\delta_e}\, U_{11}^*)$

(9) $\varphi_2 = 2\arg(e^{\mathrm{i}\delta_e}\, U_{12}^*)$

Here we used the relation[3]

$$\mathrm{Im}\left(U_{11}^* U_{13} U_{31} U_{33}^*\right) \quad = \quad c_{12}\, c_{13}^2\, c_{23}\, s_{13} \left(e^{-\mathrm{i}\delta}\, s_{12}\, s_{23} - c_{12}\, c_{23}\, s_{13}\right) \ .$$

Note that this relation is often used in order to introduce the Jarlskog invariants [?]

$$\begin{aligned} J_{\mathrm{CP}} &= \frac{1}{2}\left|\mathrm{Im}(U_{11}^* U_{12} U_{21} U_{22}^*)\right| = \frac{1}{2}\left|\mathrm{Im}(U_{11}^* U_{13} U_{31} U_{33}^*)\right| \\ &= \frac{1}{2}\left|\mathrm{Im}(U_{22}^* U_{23} U_{32} U_{33}^*)\right| = \frac{1}{2}\left|c_{12}\, c_{13}^2\, c_{23}\, \sin\delta\, s_{12}\, s_{13}\, s_{23}\right| \ . \end{aligned} \tag{5.12}$$

Another comment: by definition, θ_{13} is always positive. Technically this is achieved by an approriate choice of δ.

5.4 Remarks on special (degenerate) cases

(i) Diagonal mass matrices

If Y_e and m_ν are diagonal, the phases are mathematically not well-defined. However, even though one may trade δ_e for φ_1, none of the phases is physical. Hence, `MNSParameters` returns only non-zero δ_e, δ_μ and δ_τ in this case.[4]

(ii) Degenerate masses

In the case of degenerate masses, mixings are undefinded, because by definition mixings only occur between mass eigenstates which are not fixed in this case. `MNSParameters` will return arbitrary mixing angles and print a warning.

[3] There was an error in an earlier version of this relation. We are grateful to Yang Bai for pointing it out to us.

[4] For the quark sector, i.e. `CKMParameters`, a similar thing is not implemented at the moment. . .

(iii) $\boldsymbol{\theta_{13} = 0}$

For a zero CHOOZ angle, the Dirac phase is undefined. Besides, the determination of φ_2 and δ_e has to be modified. In general, we use

$$\delta = 0\,, \tag{5.13a}$$
$$\varphi_2 = 2\arg(e^{\mathrm{i}\delta_\mu} U_{22}^*)\,, \tag{5.13b}$$
$$\delta_e = \arg(e^{\mathrm{i}\frac{\varphi_2}{2}} U_{12})\,. \tag{5.13c}$$

For special values of θ_{12} and θ_{23}, we apply the following modifications:

$\theta_{12} = 0$**:**

$$\varphi_1 = 0\,, \tag{5.14a}$$
$$\delta_e = \arg(U_{11})\,. \tag{5.14b}$$

$\theta_{12} = \pi/2$**:** This case is not considered yet.

$\theta_{23} = 0$**:** The phases δ_e, δ_μ, φ_1 and φ_2 are linearly dependent due to the orthonormality of U (cf. the case $\theta_{13} = \pi/2$ below), so that we can choose one of them to be zero.

$$\varphi_1 = 0\,, \tag{5.15a}$$
$$\delta_e = \arg(U_{11})\,, \tag{5.15b}$$
$$\varphi_2 = 2\arg(e^{\mathrm{i}\delta_e} U_{12}^*)\,, \tag{5.15c}$$
$$\delta_\mu = \arg(-U_{21})\,. \tag{5.15d}$$

$\theta_{23} = \pi/2$**:**

$$\varphi_1 = 0\,, \tag{5.16a}$$
$$\delta_e = \arg(U_{11})\,, \tag{5.16b}$$
$$\varphi_2 = 2\arg(e^{\mathrm{i}\delta_e} U_{12}^*)\,, \tag{5.16c}$$
$$\delta_\tau = \arg(-e^{\mathrm{i}\frac{\varphi_2}{2}} U_{32})\,. \tag{5.16d}$$

(iv) $\theta_{13} = \pi/2$

This case often occurs when the neutrino mass hierarchy changes, i.e. Δm^2_{sol} overtakes Δm^2_{atm} during the running. As the rows and columns of the mixing matrix have to be normalized, it can be written as

$$U = \begin{pmatrix} 0 & 0 & e^{\mathrm{i}\delta_1} \\ -e^{\mathrm{i}\varphi_{21}} \sin\theta & e^{\mathrm{i}\varphi_{22}} \cos\theta & 0 \\ -e^{\mathrm{i}\varphi_{31}} \cos\theta & -e^{\mathrm{i}\varphi_{32}} \sin\theta & 0 \end{pmatrix} , \tag{5.17}$$

where the positions of sin and cos and of the minus signs are arbitrary, of course. We choose the angle θ to equal θ_{23} and the phase of the 13-element to equal δ_e. This implies $\theta_{12} = 0$ and $\delta = 0$. Furthermore, from the orthogonality of the rows we find $U^*_{21} U_{22} + U^*_{31} U_{32} = 0$, which means that $\varphi_{21} - \varphi_{22} - \varphi_{31} + \varphi_{32} = n\,2\pi$ ($n \in \mathbb{Z}$). Consequently, one of the remaining four phases is arbitrary, and we choose $\varphi_1 = 0$. Comparing eq. (5.17) to the standard parametrization now leads to the parametrization of the MNS matrix for $\theta_{13} = \pi/2$,

$$U = \begin{pmatrix} 0 & 0 & e^{\mathrm{i}\delta_e} \\ -e^{\mathrm{i}\delta_\mu} s_{23} & e^{\mathrm{i}(\delta_\mu - \varphi_2/2)} c_{23} & 0 \\ -e^{\mathrm{i}\delta_\tau} c_{23} & -e^{\mathrm{i}(\delta_\tau - \varphi_2/2)} s_{23} & 0 \end{pmatrix} . \tag{5.18}$$

Thus, the mixing parameters are determined as follows:

$$\begin{aligned}
\delta &= 0 \,, && (5.19\text{a}) \\
\delta_e &= \arg(U_{13}) \,, && (5.19\text{b}) \\
\theta_{12} &= 0 \,, && (5.19\text{c}) \\
\theta_{23} &= \arctan(|U_{21}/U_{31}|) \,, && (5.19\text{d}) \\
\varphi_1 &= 0 \,, && (5.19\text{e}) \\
\delta_\mu &= \arg(-U_{21}) \,, && (5.19\text{f}) \\
\varphi_2 &= 2\,(\delta_\mu - \arg(U_{22})) \,, && (5.19\text{g}) \\
\delta_\tau &= \arg(-U_{31}) \,. && (5.19\text{h})
\end{aligned}$$

For $U_{21} = 0$ ($\theta_{23} = 0$), we use $\delta_\mu = \arg(U_{22})$, $\varphi_2 = 0$. For $U_{31} = 0$ ($\theta_{23} = \pi/2$), we can also set $\varphi_2 = 0$; the phase δ_τ is then determined from $\delta_\tau = \arg(-U_{32})$. This also affects functions based on `MPT3x3MixingParameters[`U`]`, such as `MNSParameters` and `CKMParameters`. *…under construction… Ist dieser Kommentar noch aktuell? (J.)*

(v) $\theta_{13} \neq 0, \pi/2$, $\theta_{12} = 0$ or $\theta_{23} = 0, \pi/2$

For special values of θ_{12} or θ_{23}, we use the following modifications after the standard procedure of Sec. 5.3:

$\theta_{23} = 0$:

$$\delta_\mu = \arg(e^{i\frac{\varphi_2}{2}} U_{22}) \ . \tag{5.20}$$

$\theta_{23} = \pi/2$:

$$\delta_\tau = \arg(-e^{i\frac{\varphi_2}{2}} U_{32}) \ . \tag{5.21}$$

$\theta_{12} = 0$:

$$\varphi_2 = 2\arg(e^{i\delta_\mu} U_{22}^*) \ . \tag{5.22}$$

$\theta_{12} = \pi/2$: Not implemented yet.

The combination $\theta_{12} = \theta_{23} = 0$ does not require a separate If-query in the code: In step (9), of the standard procedure, φ_2 is set to 0, since $U_{12} = 0$. Afterwards, Eq. (5.20) yields $\delta_\mu = \arg(U_{22})$. Finally, φ_2 is again set to 0, this time because of Eq. (5.22).
The case $\theta_{12} = 0$, $\theta_{23} = \pi/2$ is also ok: First, step (9) of the standard procedure yields $\varphi_2 = 0$ due to $U_{12} = 0$. Next, Eq. (5.21) yields the correct value $\delta_\tau = \arg(-U_{32})$. Afterwards, Eq. (5.22) gives φ_2 once again, since δ_μ has been determined in step (4) of the standard procedure, so that $e^{i\delta_\mu} U_{22}^*$ is real.

A Theorems on Matrix-Diagonalization

Hermitian Matrices

A.1 Theorem:
Hermitian matrices M can be diagonalized by unitary transformations,

$$U^\dagger M U = \mathrm{diag}(M_1, \ldots, M_n) \ , \tag{1.1}$$

where U is unitary and the eigenvalues M_i are real. The columns of U contain the eigenvectors of M.

Proof. *See the standard textbooks on linear algebra.* □

General Matrices (Biunitary Diagonalization)

A.2 Theorem:
A general, non-singular matrix M can be diagonalized by a biunitary *transformation,*

$$U_{\rm L}^\dagger M U_{\rm R} = \mathrm{diag}(M_1, \dots, M_n)\ , \tag{1.2}$$

if none of the eigenvalues of $M^\dagger M$ equals zero. $U_{\rm L}$ and $U_{\rm R}$ are unitary, and M_i are real and positive. The matrices $U_{\rm L}$ and $U_{\rm R}$ can be found by determining the unitary transformations which diagonalize $MM^\dagger$ and $M^\dagger M$, respectively, i.e.

$$\begin{aligned} U_{\rm L}^\dagger\, MM^\dagger\, U_{\rm L} &= \mathrm{diag}(M_1^2, \dots, M_n^2)\ , & (1.3a)\\ U_{\rm R}^\dagger\, M^\dagger M\, U_{\rm R} &= \mathrm{diag}(M_1^2, \dots, M_n^2)\ . & (1.3b) \end{aligned}$$

Proof. *Define*

$$H^2 := MM^\dagger\ , \tag{1.4}$$

which is obviously Hermitian and can therefore be diagonalized by a unitary tranformation,

$$U_{\rm L}^\dagger\, MM^\dagger\, U_{\rm L} = \mathrm{diag}(M_1^2, \dots, M_n^2) =: D^2\ , \tag{1.5}$$

where M_i are real and positive. Define D as the diagonal matrix containing the squareroots of D^2. Then obviously

$$H := U_{\rm L} D U_{\rm L}^\dagger \tag{1.6}$$

satisfies equation (1.4). With $V := H^{-1}M$, which is unitary because

$$V^\dagger V \overset{H^\dagger = H}{=} M^\dagger H^{-1} H^{-1} M \overset{(1.4)}{=} M^\dagger (MM^\dagger)^{-1} M = \mathbb{1}\ , \tag{1.7}$$

we find

$$M = HV \overset{(1.6)}{=} U_{\rm L} D U_{\rm R}^\dagger\ , \tag{1.8}$$

where $U_{\rm R} := V^\dagger U_{\rm L}$ is unitary, so that equation (1.2) is proven. Furthermore, $U_{\rm R}$ diagonalizes $M^\dagger M$, since

$$U_{\rm R}^\dagger M^\dagger M U_{\rm R} \overset{(1.8)}{=} U_{\rm R}^\dagger\, U_{\rm R} D U_{\rm L}^\dagger\, U_{\rm L} D U_{\rm R}^\dagger\, U_{\rm R} = D^2\ , \tag{1.9}$$

which proves equation (1.3b). □

Symmetric Matrices

A.3 Theorem:
Complex symmetric matrices can be diagonalized by a unitary matrix U,

$$U^T M U = \mathrm{diag}(M_1, \ldots, M_n) := D \;, \tag{1.10}$$

where

$$U^\dagger\, M^\dagger M\, U = D^2 \;, \tag{1.11}$$

i.e. the real numbers M_i are the square roots of the eigenvalues of $M^\dagger M$.

Proof. *From theorem A.2 we know that*

$$M = U_\mathrm{L} D U_\mathrm{R}^\dagger \;, \tag{1.12}$$

where U_L, U_R and D are uniquely determined.[5] *As M is symmetric, it follows that*

$$M = M^T = U_\mathrm{R}^* D U_\mathrm{L}^T \;. \tag{1.13}$$

On the other hand, we can view the last equation as the diagonalization of M^T, which is uniquely determined as well according to theorem A.2. Hence, we conclude $U_\mathrm{L} = U_\mathrm{R}^*$, which completes the proof if we set $U := U_\mathrm{R}$ and take into account equation (1.3b). □

B Some thoughts during implementation

θ_{13} problem

if $\theta_{13} = \pi/2$ is encountered. there are 'angle and phase moduli', i.e. combinations of angles and phases which do not change the resulting unitary matrix when plugged in. This is obvious since the only non-trivial information is encoded in the bottom-left 2×2 sub-block (and the phase of U_{13}). This sub-block can be parametrized by one angle.

degenerate eigenvalues

Mathematica has the function `SingularValueDecomposition[`m`]` which returns three matrices U, V and W such that W is diagonal, U and V are unitary and

$$m \;=\; U\, W\, V^\dagger \;. \tag{2.1}$$

- This function is obviously very useful for matrix diagonalization (it almost does everything we need).

[5] Note that U_L, U_R are not always unique: If the eigenvalues of D are degenerate, there exist matrices U which diagonalize $M^\dagger M$, i.e. $U^\dagger\, M^\dagger M\, U = D$, which however do not diagonalize M. In this case, M can still be diagonalized, but the matrix which does the job can not simply be obtained by calculating the eigenvectors of $M^\dagger M$.

- For non-degenerate eigenvalues and symmetric m we have due to A.3 that $U^* = V$.
- However, for degenerate mass eigenvalues, `SingularValueDecomposition[`m`]` returns still U, V and W such that (2.1) holds, but $U^* \neq V$ in general. In this case, one (combination of) mixing angle(s) is unphysical. U^* and V differ by an unphysical rotation. If one takes the average, one can construct a matrix V such that $V^T M V$ is diagonal.
- Unfortunately, when trying to identify these unphysical parameters, one runs into the $\theta_{13} = \pi/2$ problem. So, this has to be solved first.