

Mixing parameter tools 1.0

Documentation

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

December 5, 2008

This is a short 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, and briefly explain our conventions. There is also some information concerning the installation.

Contents

1	Short description and comments	3
2	Installation	3
2.1	Automatic installation (UNIX/Linux only)	3
2.2	Semi-automatic installation (UNIX/Linux only)	3
2.3	Installation by hand	3
3	Quick start	5
4	Description of the functions	7
	<code>MNSMatrix</code>	7
	<code>MNSParameters</code>	7
	<code>DiracMNSMatrix</code>	7
	<code>DiracMNSParameters</code>	7
	<code>CKMMatrix</code>	7
	<code>CKMParameters</code>	8

5	Remarks	8
5.1	Remarks on the calculation of the CKM matrix	8
5.2	Remarks on the calculation of the MNS matrix	9
5.3	Definition and Extraction of Mixing Parameters	9
	(i) Standard Parametrization	9
	(ii) Extracting Mixing Angles and Phases	9
5.4	Remarks on special (degenerate) cases	10
6	Trouble-Shooting	11
A	Theorems on Matrix-Diagonalization	12

1 Short description and comments

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

This package comes along with the paper [hep-ph/0501272](https://arxiv.org/abs/hep-ph/0501272).

2 Installation

2.1 Automatic installation (UNIX/Linux only)

Execute `MixingParameterTools.installer` and you are done.

```
sh MixingParameterTools.installer
```

The package is copied to `~/.Mathematica/Applications/MixingParameterTools`. The documentation and an example notebook are placed in a subdirectory of the working directory, which is called `MixingParameterTools`.

2.2 Semi-automatic installation (UNIX/Linux only)

Unpack the archive `MixingParameterTools.tar.gz`.

```
tar -xvzf MixingParameterTools.tar.gz
```

Then go to the directory `MixingParameterToolsInstall` and execute the script `install.sh`.

```
cd MixingParameterToolsInstall
./install.sh
```

This copies the package to `~/.Mathematica/Applications/MixingParameterTools`. The documentation and an example notebook are placed in a new subdirectory of the working directory called `MixingParameterTools`. Hence, the folder `MixingParameterToolsInstall` can be deleted now.

2.3 Installation by hand

In order to install the package manually, the archive `MixingParameterTools.tar.gz` has to be unpacked first. Under UNIX/Linux, type

```
tar -xvzf MixingParameterTools.tar.gz
```

On Windows systems, a program like WinZip can be used. Then one has to move the directory `MixingParameterTools` from the folder `MixingParameterToolsInstall` to the directory where the `Mathematica` add-ons are located, e.g.

```
mv MixingParameterToolsInstall/MixingParameterTools \  
~/.Mathematica/Applications/
```

Under Windows XP, the path to the add-on directory should be something like `Application Data\Mathematica\Applications`. The documentation and an example notebook can be found in

`MixingParameterToolsInstall/Doc/MixingParameterTools/`.

3 Quick start

After installation, the only thing one has to do in order to have access to the functions is to load the package:

```
Needs["MixingParameterTools`MPT3x3`"];
```

Let's try if the package is really loaded. We can just ask *Mathematica* if it knows about some functions, e.g.

```
?CKMParameters
```

```
CKMParameters[Yu,Yd] returns the CKM mixing parameters  $\{\theta_{12}, \theta_{13}, \theta_{23}, \delta\}$  for up- and down-type Yukawa matrices Yu and Yd, as well as the Yukawa couplings  $\{Y_u, Y_c, Y_t\}$  and  $\{Y_d, Y_s, Y_b\}$ .
```

If this works, *Mathematica* will return a short description of this command *CKMParameters*. Now we can start playing. Just for fun, let us generate 'predictions' randomly, e.g. by

```
ClearAll[rndEntry, rndMatrix];
rndEntry := Exp[Random[Complex, {-6, 0.2 + 2 π I}]];
rndMatrix :=  $\begin{pmatrix} \text{rndEntry} & \text{rndEntry} & \text{rndEntry} \\ \text{rndEntry} & \text{rndEntry} & \text{rndEntry} \\ \text{rndEntry} & \text{rndEntry} & \text{rndEntry} \end{pmatrix}$ ;
```

Let's see what the 'prediction' of the above ansatz is:

```
ClearAll[Yd, Yu, UCKM];
Yd = rndMatrix;
Yu = rndMatrix;
UCKM = Catch[CKMMatrix[Yu, Yd]];
Print["UCKM="];
Print[MatrixForm[UCKM]];
```

```
UCKM =
```

```
 $\begin{pmatrix} 0.0334209 + 0.362879 i & -0.0692623 - 0.115267 i & 0.168338 + 0.905969 i \\ -0.76602 - 0.522548 i & 0.0123705 + 0.0569267 i & 0.319403 + 0.186401 i \\ -0.0728749 + 0.0452632 i & -0.923052 + 0.355664 i & -0.117296 - 0.0189635 i \end{pmatrix}$ 
```

OK, this is a 3×3 matrix. Fine. But suppose we're interested in the Cabbibo angle. Let us try

```
Print["CKM parameters: ", Catch[CKMParameters[Yu, Yd]]];
```

```
CKM parameters: {{0.353517, 1.17186, 1.25992, 0.189274},
 {0.00327824, 0.544895, 0.564447}, {0.00586943, 0.0820568, 0.163917}}
```

This looks better. According to the above short description, the first list contains the three angles and the Dirac phase, and the first entry of the first list is the Cabbibo angle. Let's see...

```
Print[" $\theta_{\text{Cabbibo}}$ =", Catch[CKMParameters[Yu, Yd]][[1, 1]] / Degree];
```

```
 $\theta_{\text{Cabbibo}}$ =20.2551
```

All right, our texture is probably only semi-realistic. Maybe we are more lucky with neutrinos.

```
ClearAll[Ye, tmpF, m];
Ye = rndMatrix;
tmpF = rndMatrix;
m = tmpF + Transpose[tmpF];
Print["MNS parameters: ", Catch[MNSParameters[m, Ye]]];
```

```
MNS parameters: {{0.3749, 0.128276, 0.231185,
 0.328005, 1.15855, 1.38587, 0.986024, 1.26561, 3.53476},
 {0.889961, 0.0452562, 1.89558}, {0.0429663, 0.143401, 0.236882}}
```

Now a strange thing happened. The entries of the second list, the neutrino mass eigenvalues, are not ordered ascendingly. This is because it always given in a form in which it is easy to compare with experiments (cf. Sec. 5.2).

We could be especially interested in the solar mixing angle:

```
Print[" $\theta_{12}$ =", Catch[MNSParameters[m, Ye]][[1, 1]] / Degree]
```

```
 $\theta_{12}$ =21.4802
```

Well, also not too close to the central experimental value. Let's try another texture...

At this point, we decide to leave the 'quick start' section, and turn to the description of the functions.

4 Description of the functions

MNSMatrix

`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 normal or inverted by calling `MNSMatrix`[$m, Y_e, \text{'n'}$] or `MNSMatrix`[$m, Y_e, \text{'i'}$], respectively. This option is useful if the hierarchy changes due to RG evolution. Note that the input matrices m and Y_e must be numeric.

MNSParameters

`MNSParameters`[m, Y_e] returns the MNS mixing and mass parameters $\{\{\theta_{12}, \theta_{13}, \theta_{23}, \delta, \delta_e, \delta_\mu, \delta_\tau, \varphi_1, \varphi_2\}, \{m_1, m_2, m_3\}, \{y_e, y_\mu, y_\tau\}\}$ 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 the hierarchy to be normal or inverted by calling `MNSMatrix`[$m, Y_e, \text{'n'}$] or `MNSMatrix`[$m, Y_e, \text{'i'}$], respectively. Note that the input matrices m and Y_e must be numeric. Furthermore, if parameters are undefined, some viable choice is returned. For instance, if $\theta_{13} = 0$, the function returns $\delta = 0$.

DiracMNSMatrix

`DiracMNSMatrix`[Y_ν, Y_e] returns the MNS matrix for Dirac neutrinos with Yukawa coupling Y_ν .

DiracMNSParameters

`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 $U_{\text{MNS}}^{\text{Dirac}}$, are unphysical. Note that the input matrices Y_ν and Y_e must be numeric. Furthermore, if parameters are undefined, some viable choice is returned. For instance, if $\theta_{13} = 0$, the function returns $\delta = 0$.

CKMMatrix

`CKMMatrix`[Y_u, Y_d] returns the CKM 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.

CKMParameters

`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. Note that the input matrices Y_u and Y_d must be numeric. Furthermore, if parameters are undefined, some viable choice is returned. For instance, if $\theta_{13} = 0$, the function returns $\delta = 0$.

5 Remarks

5.1 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

$$\mathcal{L}_{\text{Yukawa}} = \overline{\psi_R^f} Y_{fg} \psi_L^g + \text{h.c.} , \quad (5.1)$$

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

$$\psi_R \rightarrow U_R^\dagger \psi_R , \quad (5.2a)$$

$$\psi_L \rightarrow U_L^\dagger \psi_L , \quad (5.2b)$$

$$Y \rightarrow U_R^\dagger Y U_L = \text{diag}(y_1, y_2, y_3) , \quad (5.2c)$$

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

$$U_L^\dagger Y^\dagger Y U_L \stackrel{!}{=} \text{diag}(|y_1|^2, |y_2|^2, |y_3|^2) , \quad (5.3a)$$

$$U_R^\dagger Y Y^\dagger U_R \stackrel{!}{=} \text{diag}(|y_1|^2, |y_2|^2, |y_3|^2) , \quad (5.3b)$$

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.

$$Y_u \rightarrow (U_R^{(u)})^\dagger Y_u U_L^{(u)} = \text{diag}(y_u, y_c, y_t) , \quad (5.4a)$$

$$Y_d \rightarrow (U_R^{(u)})^\dagger Y_d U_L^{(u)} =: Y'_d . \quad (5.4b)$$

- (2) Calculate U_L for Y'_d . This is U_{CKM} .

5.2 Remarks on the calculation of the MNS matrix

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

$$Y_e \rightarrow U_R^\dagger Y_e U_L = \text{diag}(y_e, y_\mu, y_\tau), \quad (5.5a)$$

$$m_\nu \rightarrow U_L^T m_\nu U_L =: m'_\nu. \quad (5.5b)$$

The MNS matrix has to fulfill

$$U_{\text{MNS}}^T m'_\nu U_{\text{MNS}} = \text{diag}(m_1, m_2, m_3), \quad (5.6)$$

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 $\Delta m_{\text{sol}}^2 = m_i^2 - m_j^2$, and a larger one, referred to as $\Delta m_{\text{atm}}^2 = m_k^2 - m_\ell^2$. 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.

5.3 Definition and Extraction of Mixing Parameters

(i) 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

$$U = \text{diag}(e^{i\delta_e}, e^{i\delta_\mu}, e^{i\delta_\tau}) \cdot V \cdot \text{diag}(e^{-i\varphi_1/2}, e^{-i\varphi_2/2}, 1) =: K_\delta \cdot V \cdot K_\varphi, \quad (5.7)$$

where

$$V = \begin{pmatrix} c_{12}c_{13} & s_{12}c_{13} & s_{13}e^{-i\delta} \\ -c_{23}s_{12} - s_{23}s_{13}c_{12}e^{i\delta} & c_{23}c_{12} - s_{23}s_{13}s_{12}e^{i\delta} & s_{23}c_{13} \\ s_{23}s_{12} - c_{23}s_{13}c_{12}e^{i\delta} & -s_{23}c_{12} - c_{23}s_{13}s_{12}e^{i\delta} & c_{23}c_{13} \end{pmatrix} \quad (5.8)$$

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

(ii) 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 $0 \leq \theta_{12} \leq \frac{\pi}{4}$. 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:

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

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

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

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

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

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

where $i, j \in \{1, 2, 3\}$ and $i \neq j$.

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

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

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

Here we used the relation¹

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

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

$$\begin{aligned} J_{\text{CP}} &= \frac{1}{2} |\text{Im}(U_{11}^* U_{12} U_{21} U_{22}^*)| = \frac{1}{2} |\text{Im}(U_{11}^* U_{13} U_{31} U_{33}^*)| \\ &= \frac{1}{2} |\text{Im}(U_{22}^* U_{23} U_{32} U_{33}^*)| = \frac{1}{2} |c_{12} c_{13}^2 c_{23} \sin \delta s_{12} s_{13} s_{23}| . \end{aligned} \quad (5.9)$$

For the sake of a better numerical stability, one can choose any of the three combinations. In particular, if the modulus of one of the U_{ij} is very small, it turns out to be more accurate to choose a combination in which this specific U_{ij} does not appear.

5.4 Remarks on special (degenerate) cases

There are several cases where the mixing parameters are not uniquely defined, e.g. when the eigenvalues are degenerate. The package tries to return one set of possible mixing parameters then. In some cases, this is not successful, and an error message is produced. We keep on improving on these cases. Let us, however, stress that for phenomenologically viable mass matrices, and moderate deformations thereof, our functions work without problems.

¹There was an error in an earlier version of this relation. We are grateful to Yang Bai for pointing it out to us.

6 Trouble-Shooting

There may be bugs in the software. We will collect them on the web page

<http://www.ph.tum.de/~rge/MPT/bugs.html> .

In case you encounter a new bug, or if you have other suggestions, please write an email to `rge@ph.tum.de`.

A Theorems on Matrix-Diagonalization

Hermitean Matrices

A.1 Theorem:

Hermitean matrices M can be diagonalized by unitary transformations,

$$U^\dagger M U = \text{diag}(M_1, \dots, M_n), \quad (1.1)$$

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. \square

General Matrices (Biunitary Diagonalization)

A.2 Theorem:

A general, non-singular matrix M can be diagonalized by a biunitary transformation,

$$U_L^\dagger M U_R = \text{diag}(M_1, \dots, M_n), \quad (1.2)$$

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

$$U_L^\dagger M M^\dagger U_L = \text{diag}(M_1^2, \dots, M_n^2), \quad (1.3a)$$

$$U_R^\dagger M^\dagger M U_R = \text{diag}(M_1^2, \dots, M_n^2). \quad (1.3b)$$

Proof. Define

$$H^2 := M M^\dagger, \quad (1.4)$$

which is obviously Hermitean and can therefore be diagonalized by a unitary transformation,

$$U_L^\dagger M M^\dagger U_L = \text{diag}(M_1^2, \dots, M_n^2) =: D^2, \quad (1.5)$$

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

$$H := U_L D U_L^\dagger \quad (1.6)$$

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

$$V^\dagger V \stackrel{H^\dagger=H}{=} M^\dagger H^{-1} H^{-1} M \stackrel{(1.4)}{=} M^\dagger (M M^\dagger)^{-1} M = \mathbb{1}, \quad (1.7)$$

we find

$$M = H V \stackrel{(1.6)}{=} U_L D U_L^\dagger, \quad (1.8)$$

where $U_R := V^\dagger U_L$ is unitary, so that equation (1.2) is proven. Furthermore, U_R diagonalizes $M^\dagger M$, since

$$U_R^\dagger M^\dagger M U_R \stackrel{(1.8)}{=} U_R^\dagger U_R D U_L^\dagger U_L D U_R^\dagger U_R = D^2, \quad (1.9)$$

which proves equation (1.3b). \square

Symmetric Matrices

A.3 Theorem:

Complex symmetric matrices can be diagonalized by a unitary matrix U ,

$$U^T M U = \text{diag}(M_1, \dots, M_n) := D, \quad (1.10)$$

where

$$U^\dagger M^\dagger M U = D^2, \quad (1.11)$$

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

$$M = U_L D U_R^\dagger, \quad (1.12)$$

where U_L , U_R and D are uniquely determined.² As M is symmetric, it follows that

$$M = M^T = U_R^* D U_L^T. \quad (1.13)$$

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 $U_L = U_R^*$, which completes the proof if we set $U := U_R$ and take into account equation (1.3b). \square

²Note 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 $M^\dagger M$.