STATE RESEARCH CENTER OF RUSSIA
INSTITUTE FOR HIGH ENERGY PHYSICS

IHEP 2004–24

V.V. Ezhela, V.N. Larin

StandardPhysicalConstants 2.1 **Package**
(User's Guide)

Protvino 2004

UDK 539.1 M–24

Abstract

Ezhela V.V., Larin V.N. StandardPhysicalConstants 2.1 Package. (User's Guide): IHEP Preprint 2004–24. – Protvino, 2004. – p. 26.

The detailed user's guide of the StandardPhysicalConstants 2.1 package for *Mathematica* system is given. The package is to calculate average values and (co)variances of the functions dependent upon fundamental physical constants (FPC). The package set of FPC (113 constants) is combined from adjusted and recommended CODATA(2002) and PDG(2004) values, their variances, and correlations.

Аннотация

Ежела В.В., Ларин В.Н. Пакет StandardPhysicalConstants 2.1. (Руководство для пользователя): Препринт ИФВЭ 2004–24. – Протвино, 2004. – 26 с.

Представлено детальное описание пакета StandardPhysicalConstants версии 2.1 для вычислений средних значений и (ко)вариаций функций от фундаментальных физических постоянных (ФФП) в системе *Mathematica*. Пакет ориентирован на работу с объединенной выборкой (113 постоянных) из оцененных и рекомендованных CODATA(2002) и PDG(2004) значений ФФП с учетом оцененных и рекомендованных значений корреляций их неопределенностей.

Contents

1. Introduction **2**

2. The Package Structure and Initialization **2**

2.1. The Structure of Package 2

2.2. Package Initialization 2

3. Database **3**

3.1. Data Structure 3

3.2. Data Sources and Data Quality 4

4. Package Functionality **4**

4.1. Data Management Functions 4

4.1.1. NewListFPC 5

4.1.2. PrintDB 6

4.1.3. OutputDB 7

4.1.4. InsertFPC 7

4.1.5. DeleteFPC 9

4.1.6. Copy 9

4.1.7. Shift 10

4.2. Functions to Access to the Data 10

4.2.1. Access 11

4.2.2. PairCC 12

4.2.3. GroupCC 12

4.2.4. MakeCM 13

4.3. Calculators 14

4.3.1. ErrorPropagator 14

4.3.2. MatrixCharacteristics 17

4.3.3. Covariator 19

5. Summary **22**

6. Appendix **24**

1. Introduction

The results of the two recent re-evaluations and adjustments of the fundamental physical constants (FPC) in 1998 [1] and 2002 [2] performed at NIST show that the obtained correlation matrices are ill-conditioned matrices. To work with the ill-conditioned matrices one need to use the high accuracy to get reliable output from calculations. In particular, the ignorance of the correlations or their unjustified rounding can lead to the absurd results (see example in [3]).

However, the ignorance of the correlations in FPC turned to be the common practice. In all popular software systems for the scientific computations [4], [5], [6] the samples of FPC are extracted from the CODATA recommended FPC site but without correlations. Even there are no warnings there, informing users on the presence of large ($> 90\%$) correlations of the FPC uncertainties.

The proposed package StandardPhysicalConstants [3, 7] in the *Mathematica* [8] system is to remove this ignorance of correlations of the FPC in calculations for particle physics and related fields.

The package is still under development and some options for the comfortable usage are to be designed and implemented. Nevertheless, the package is elaborated enough to be used in tasks where the correlations are crucial for obtaining the meaningful results.

2. The Package Structure and Initialization

2.1. The Structure of Package

Package consists of two basic files (name.m) written on *Mathematica* language and three notebooks (name.nb):

Dbfpc113.m — is the database with information on the 113 physical constants.

Manager.m — is the package of the management functions for access, storage, and maintenance of data. It also contains modules to calculate average values, (co)variations, and correlations of functions depending on FPC and other random variables that stored in the working database.

FPalette.nb — is the palette of the package functions.

CPalette_113.nb — is the palette of the package FPC-names.

Read_me.nb — is the description of the package functionality.

2.2. Package Initialization

To initialize the package one should load two basic files in the current *Mathematica* session. This can be done by instruction Get[*file_name*] or <<*file_name*, where *file_name* — complete name of the file.

To be allowed to use "short" names the basic files should be located in the *Autoload* directory (one is automatically created at the *Mathematica* system installation). The other way is to use the command SetDirectory["*full_path*"].

If any of the above operations is performed then package will became on-line after instructions (in any order)

In[1] := **<< Dbfpc113.m**

In[2] := **<< Manager.m**

These instructions can be issued from the FPalette.nb palette also.

3. Database

Current version of the database Dbfpc113.m contains data on average values, corresponding standard uncertainties, and *known* correlation coefficients for 113 physical constants.

In particular all universal constants are in the database: speed of light in vacuum ($\boldsymbol{c_0}$), magnetic and electric constants ($\boldsymbol{\mu_0}$ и $\boldsymbol{\varepsilon_0}$), elementary charge ($\boldsymbol{e}$), Planck constant ($\boldsymbol{h}$), and others.

It also includes the set of most frequently used electromagnetic, atomic and nuclear constants: the Bohr magneton ($\boldsymbol{\mu_B}$); fine structure constant ($\boldsymbol{\alpha}$), masses, mass-ratios, and magnetic moments of some elementary particles ($\boldsymbol{m_e}$, $\boldsymbol{m_e/m_\mu}$, $\boldsymbol{\mu_p}$ etc.).

In addition there are a sample of astrophysical constants containing: Newtonian gravitational constant ($\boldsymbol{G}$), Planck mass ($\sqrt{\boldsymbol{\hbar c_0/G}}$), Hubble expansion rate ($\boldsymbol{H_0}$) etc. The complete FPC list included in the current version of the database (Dbfpc113.m) is displayed in **Appendix.**

3.1. Data Structure

All data placed in a list dataBase which has three components — three tables:

dataBase = { basicTable, standardNames, ccTable },

where:

basicTable is the two dimensional table, containing one row for each constant that is the list of data elements {*integer_ number, package_ name, constant_ symbol, value, units*}, that include the sequential number of the constant in this table (its local identifier *code* in the database), the *package_ name* that is the global constant identifier in any context. The rest data elements are the constant symbol, constant value in the NIST concise form

$$average(uncertainty) \times 10^{scale},$$

and units.

standardNames is the list of the FPC standard names (as they were defined in the data sources).

ccTable is the correlation matrix stored as two dimensional table with one row for every upper off diagonal correlation matrix element $r_{i<j}$:

$$\{code_1,\, code_2,\, constant\_symbol_1,\, constant\_symbol_2,\, r_{code_1,\, code_2}\}.$$

The values of all data elements, except the integer-type code identifier, have *character string* type.

The structures of these tables are presented in the examples in the next sections.

3.2. Data Sources and Data Quality

Data base Dbfpc113.m is assembled from samples extracted from two complementary resources [2, 9]. It should be noted that these resources are partly overlapping but not synchronised by the time of adjustments. The completeness of the data presentation is also different.

Re-evaluations and re-adjustments of the traditional CODATA set of FPC were episodic since 1950, and starting 1998 are performed every 4 years (see. [1], page 355). The last readjustment was performed using the data and theoretical formulae published before 2003 (see [2], where for the majority of the CODATA recommended FPC one can find average values, standard deviations, and pairwise correlation coefficients).

Unfortunately only a small subset of constants used in particle and astro-particle physics are included in the CODATA adjustments cycles. In the standard set of FPC (as of 2002) there are no data on the well established values of the masses of $\boldsymbol{W^{\pm}}$ and $\boldsymbol{Z^{0}}$ bosons, strong coupling constant $\boldsymbol{\alpha_s(M_Z)}$. The estimates for the constants: *weak mixing angle, Fermi coupling constant, tau mass [MeV]*, that are included in the CODATA-2002 FPC set [2] were extracted from the data recommended by the PDG collaboration in 2002 [10] that are partially obsolete.

The PDG collaboration performs readjustments on the particle physics and astro-particle physics constants every year with publication of the reports in even years. The last readjustment were performed in 2004 on the data and theoretical formulae published before 2004 (see RPP-2004 [9]). In this review some data from CODATA-2002 are reproduced but without correlations[1].

We see that for many applications in particle physics and astro-particle physics both resources: CODATA, and PDG are crucial. In the Dbfpc113.m all data which are "controlled" by PDG where replaced for the values from the PDG resource [9]. We assign the special value `"Null"` to all unknown correlation coefficients.

4. Package Functionality

The modules to manage data on FPC are organized in Manager.m package. They provide the following functions: data management, access to data, and calculation of the average values and variances for functions depending on FPC, as well as average values and covariance (correlation) matrices for the vector-functions depending on FPC.

4.1. Data Management Functions

This set of functions provides to create new databases, to add and delete constants, resort database, print database and elements of the database in tabular form, and dump of the created or modified database into the external file.

[1] In the CODATA-2002 recommended values of the FPC, the correlation coefficients for some constants that are heavily used in particle physics are huge, for example

$$\boldsymbol{r(m_\mu\,[MeV], m_e\,[MeV]) = 0.956, \qquad r(m_\mu\,[kg], m_e\,[kg]) = 0.988.}$$

NewListFPC	creates the new database from the existing one.
PrintDB	prints the database or indicated element of the database in tabular form.
OutputDB	outputs the indicated database into the external file with the standard name Dbfpc_n.m.
InsertFPC	inserts the new FPC to the indicated database.
DeleteFPC	deletes indicated list of FPC from the indicated database.
Copy	copies all data on the indicated constants from one database to another.
Shift	moves constant from one position to the other position in the indicated database.

4.1.1. NewListFPC

This function is to create new database using the data from the existing one (basic – dataBase or created earlier database). This function is useful if in the application the number of needed constant is small but calculations are bulky. In addition it helps to include into the temporal (working) database new constants that are absent in the dataBase or modify attributes and values of the existing constants, without changing the dataBase as it is stored in the file Dbfpc113.m).

The format of this function is as follows:

NewListFPC[*constant_ list, data_ base*],

where *constant_ list* is the list of FPC that should be copied from the *data_ base* into the new database. In the *constant_ list* the constants may be indicated by their codes in the *data_ base*, or by their package names, or in the mixture of these two possibilities:

In[3] := **wdb=NewListFPC[{102, 38, "planckConstant"}, dataBase]**

Out[3] = {{{1, fineStructureConstant, α, 7.297 352 568(24)$\times 10^{-3}$, },
{2, planckConstant, h, 6.626 0693(11)$\times 10^{-34}$, J s},
{3, speedOfLightInVacuum, c_0, 299 792 458, m s^{-1}}},
{fine-structure constant, Planck constant, speed of light in vacuum},
{{1, 2, α, h, 0.010}, {1, 3, α, c_0, 0.000}, {2, 3, h, c_0, 0.000}}}

Note that constants in the new database named **wdb** are ordered alphabetically and the local codes are changed to be 1, 2, and 3 (in the dataBase the corresponding codes are 38, 74, and 102). It is useful to remember that the code is the *local* identifier of a constant in each database, but the package name is the *global* identifier of a constant. The structure of any database created by the NewListFPC is identical to the main database (dataBase) even if the constants list is empty.

In[4] := **NewListFPC[{ }, dataBase]**

Out[4] = {{{Null, Null, Null, Null, Null}}, { }, {{Null, Null, Null, Null, Null}}}

4.1.2. PrintDB

This function is to present the indicated database or any indicated component of the database in the tabular form. The function formats are as follows:

PrintDB[*data_ base*], PrintDB[*element_ of_ data_ base*],

where *data_ base* — the name of database to be printed, a *element_ of_ data_ base* — is one of three elements of the database *data_ base[[i]]* ($i = 1, 2, 3$) or the name of the corresponding table, for example *cctab = data_ base[[3]]*.

Examples:

In[5] := **PrintDB[wdb]**

Constant Table

Code	Name	Symbol	Value	Units
1	fineStructureConstant	α	$7.297\ 352\ 568(24)\times10^{-3}$	
2	planckConstant	h	$6.626\ 0693(11)\ \times10^{-34}$	J s
3	speedOfLightInVacuum	c_0	299 792 458	m s^{-1}

Constant Standard Names

1	fine-structure constant
2	Planck constant
3	speed of light in vacuum

Correlation Coefficient Table

i	j	x_i	x_j	r_{ij}
1	2	α	h	0.010
1	3	α	c_0	0.000
2	3	h	c_0	0.000

To print only the first table, you type

In[6] := **PrintDB[wdb[[1]]]**

Constant Table

Code	Name	Symbol	Value	Units
1	fineStructureConstant	α	$7.297\ 352\ 568(24)\times10^{-3}$	
2	planckConstant	h	$6.626\ 0693(11)\ \times10^{-34}$	J s
3	speedOfLightInVacuum	c_0	299 792 458	m s^{-1}

4.1.3. OutputDB

This function is to dump the newly created database into the external file with the reserved name Dbfpc_n.m. This file will be placed into the current directory. The structure of the file Dbfpc_n.m is identical to the structure of the Dbfpc113.m and it has the same component names: *dataBase={basicTable, standardNames, ccTable}*.

This file may be edited (saving its structure), renamed and could be used in place of the basic file Dbfpc113.m as it is described in the subsection 2.2. The format of this function is as follows:

OutputDB[*data_base*],

where *data_base* is the name of the new database created by the function NewListFPC for example.

4.1.4. InsertFPC

This function is to add new (additional) constants to the database indicated as the function parameter. The formats of this function are as follows:

InsertFPC[*const_data_list, data_base*], InsertFPC[*const_data_list, data_base, initcc*],

where *const_data_list* is the list of data associated with added physical or mathematical constant, *data_base* is the name of the database to be modified, and *initcc* is the "initial" value of the correlation coefficients of the added constant with all constants already stored in the *data_base* (by default all of them have the "space" value " ").

The initial values of the correlations could be replaced later by the true ones in the *character string* format by the standard *Mathematica* options or by the external editing of the database saved to file (by OutputDB).

If the correlations of the newly entered constant are unknown the true values should be "Null". If the newly added constant is the exact one its true correlation coefficients with the other constants should be "0.000".

The first parameter of the function (*const_data_list*) is the list of five or six elements:

{code, "packagename", "symbol", "value", "units", "standardname"},

where the first element value is an integer number and the values of the rest elements have *character string* format.

The first and the last elements are the optional ones. If the first element (*code*) is omitted then the added constant will be placed in the alphabetic order in accordance with parameter "*packagename*". If the last parameter is omitted then into the table of the standard names the constant package name ("*packagename*") will be write as the standard name.

Example:

Let us add to the working database **wdb** the proton mass m_pc^2 in MeV (it is absent in the Dbfpc113.m). Data to be added are extracted from the site [2].

In[7] := **mydb = InsertFPC[{ , "protonMassInMeV", "$m_pc_0^2$", "938.272 029(80)", "MeV", "proton mass energy equivalent in MeV"}, wdb];**

We have created new working database **mydb**, and saved the "old" one (**wdb**). The obtained database is as follows:

In[8] := **PrintDB[mydb]**

Constant Table

Code	Name	Symbol	Value	Units
1	fineStructureConstant	α	$7.297\ 352\ 568(24)\times10^{-3}$	
2	planckConstant	h	$6.626\ 0693(11)\ \times10^{-34}$	J s
3	protonMassInMeV	$m_pc_0^2$	938.272 029(80)	MeV
4	speedOfLightInVacuum	c_0	299 792 458	m s^{-1}

Constant Standard Names

1	fine-structure constant
2	Planck constant
3	proton mass energy equivalent in MeV
4	speed of light in vacuum

Correlation Coefficient Table

i	j	x_i	x_j	r_{ij}
1	2	α	h	0.010
1	3	α	$m_pc_0^2$	
1	4	α	c_0	0.000
2	3	h	$m_pc_0^2$	
2	4	h	c_0	0.000
3	4	$m_pc_0^2$	c_0	

Now we fill in the empty fields of the correlation coefficients by the following assignments:

In[9] := **mydb[[3,2,5]] = "-0.086";**
mydb[[3,4,5]] = "0.995";
mydb[[3,6,5]] = "0.000";

and we get full table of the correlations:

In[10] := **PrintDB[mydb[[3]]]**

Correlation Coefficient Table

i	j	x_i	x_j	r_{ij}
1	2	α	h	0.010
1	3	α	$m_p c_0^2$	-0.086
1	4	α	c_0	0.000
2	3	h	$m_p c_0^2$	0.995
2	4	h	c_0	0.000
3	4	$m_p c_0^2$	c_0	0.000

4.1.5. DeleteFPC

This function is to remove all data associated with one or more constants from the indicated database. The format of this function is as follows:

DeleteFPC[*const, data_ base*],

where *const* is the identifier of one constant or the list of the constants identifiers to be deleted from the indicated database *data_ base*.

4.1.6. Copy

Sometimes it will be needed to add some constants from the one database (e.g. main one) to the other database. This can be done with Copy function which copies all data associated with indicated list of the constants from the "*source*" database to the "*receiver*" database. If in the *receiver* database there are constants not presented in the *source* database then their correlations with copied constants will be filled with the "Null" values. The function format is as follows:

Copy[*const, db_ source, db_ receiver*],

where *const* is constant identifier or the list of constants identifiers, the other parameters are the names of the source and receiver databases.

Example: Let us copy the Rydberg constant from the basic database to the **wdb**:

In[11] := **wdb = Copy["rydbergConstant", dataBase, wdb];**

In[12] := **PrintDB[wdb[[1]]]**

Constant Table

Code	Name	Symbol	Value	Units
1	fineStructureConstant	α	$7.297\ 352\ 568(24)\times10^{-3}$	
2	planckConstant	h	$6.626\ 0693(11)\ \times10^{-34}$	J s
3	rydbergConstant	R_∞	10 973 731.568 525(73)	m^{-1}
4	speedOfLightInVacuum	c_0	299 792 458	m s^{-1}

In[13] := **PrintDB[wdb[[3]]]**

Correlation Coefficient Table

i	j	x_i	x_j	r_{ij}
1	2	α	h	0.010
1	3	α	R_∞	–0.017
1	4	α	c_0	0.000
2	3	h	R_∞	0.000
2	4	h	c_0	0.000
3	4	R_∞	c_0	0.000

If the constant to be copied is contained in the receiver database, then it will not be copied and function Copy will print the message:

Copy:: *constant_ name* exists in DB-recipient already

In case of an attempt to copy the Rydberg constant to the **wdb** repeatedly the function Copy prints the message:

Copy:: rydbergConstant exists in DB-recipient already

4.1.7. Shift

The default alphabetical order by package names of the constants in the constants table of any database and corresponding ordering in the rest two components of this database may be changed by the function Shift with format:

Shift[*const, new_ position, data_ base*],

where *const* is the current value of the identifier of the moved constant, *new_ position* is the new code value for the moved constant, and *data_ base* is the database name under reordering. All data associated with the moved constant in all database tables will be moved coherently to preserve the integrity of the database.

4.2. Functions to Access to the Data

The functions of this set are to extract data on the indicated physical constants needed for calculational applications (average values, uncertainties and correlation coefficients).

Access	extracts and transforms to the real numbers average value, uncertainty, and relative uncertainty on the indicated constant.
PairCC	extracts the line from the correlation table corresponding to the indicated pair of constants.
GroupCC	extracts correlation sub-table for one indicated constant with the indicated group of constants.
MakeCM	transforms the correlation table of the indicated database with removed exact constants to the numerical correlation matrix.

4.2.1. Access

This function is to extract and transform to the real number format the average value and the standard deviation of the indicated constant from the indicated database. It also calculates and presents the relative uncertainty. Its format is as follows:

Access[*data_ base[[1]]*, "*const_ name*"],

where *data_ base[[1]]* is the first component of the database (constant table), "*const_ name*" is the constant package name in the *character string* format. The result is the list

{ *mean_ value, uncertainty, relative_ uncertainty* }.

The mining of the list components depends on the *value type* of the constant. In the current version of the database constants could be of four types which we describe as follows:

- *exact*, for example, the value of the magnetic constant ($\mu_0 = 4\pi \times 10^{-7}\ NA^{-2}$);
- *statistically estimated*, for example, the value of the Planck constant ($h = 6.6260693(11) \times 10^{-34}\ Js$, or $[6.6260693 \pm 0.0000011] \times 10^{-34}\ Js$);
- *interval estimated*, for example, the value of the local halo density ($\rho_{halo} = (2-13) \times 10^{-25} g\ cm^{-3}$);
- *approximate ("pseudo-exact")*, for example, the Earth radius ($R_\oplus = 6.378140 \times 10^6\ m$).

Examples:

In[14] := **Access[basicTable, "magneticConstant"]**

Out[14] = { $\frac{\pi\,\text{Second Volt}}{2500000\,\text{Ampere Meter}}$, Exact, Exact }

In[15] := **Access[mydb[[1]], "planckConstant"]**

Out[15] = { 6.6260693 $\times 10^{-34}$ Joule Second, 1.1 $\times 10^{-40}$ Joule Second, 1.7$\times 10^{-7}$ }

The values of the interval estimates are transformed by the Access function to the form of the statistical estimates in accordance with the rule:

$$V_{mean} = \frac{V_{max} + V_{min}}{2}; \qquad \sigma_V = \frac{V_{max} - V_{min}}{2}.$$

In[16] := **Access[basicTable, "localHaloDensity"]**

Out[16] = { $\frac{7.5\times 10^{-25}\,\text{Gram}}{\text{cMeter}^3}$, $\frac{5.5\times 10^{-25}\,\text{Gram}}{\text{cMeter}^3}$, 0.73 }

The approximate values are assigned the zero uncertainty.

In[17] := **Access[basicTable, "earthRadius"]**

Out[17] = { 6.378140 $\times 10^6$ Meter, 0, 0 }

4.2.2. PairCC

This function is to extract from the correlation table (third component of the database) the *line* corresponding to the indicated pair of constants. The format is as follows:

PairCC[*const1, const2, data_ base*].

Here the first two parameters are the identifiers of two constants and *data_ base* is the name of the database from which the line is to be extracted.

In[18] := **PairCC["bohrMagneton", "electronMass", dataBase]**

Out[18] = { 8, 30, μ_B, m_e, 0.991 }

In the output list the first pair of elements are the codes of the constants, the second pair are the corresponding constants symbols and the last element is the value of the correlation coefficient in the *character string* format. To transform it to the numerical format one can use the standard procedure:

In[19] := **ToExpression[Last[%]]**

Out[19] = 0.991

In[20] := **NumberQ[%]**

Out[20] = True

4.2.3. GroupCC

With the help this function one can obtain correlations of one constant with the indicated list of constants from the indicated database. The function formats are as follows:

GroupCC[*const, {c1, c2, ...}, data_ base*],
GroupCC[*const, {c1, c2, ...}, data_ base, "print"*].

Here the first parameter is the identifier of the physical constant, the second parameter is the list of constants identifiers the correlation coefficients of which with the constant indicated in the first parameter are to be extracted, and *data_ base* is the database name an extraction from which will be performed.

In[21] := **GroupCC[8, {"electronMass", 38, "boltzmannConstant", 89}, dataBase]**

Out[21] = {{8, 30, μ_B, m_e, 0.991}, {8, 38, μ_B, α, 0.106},
{8, 11, μ_B, k, 0.096}, {8, 89, μ_B, R_∞, –0.002}}

The third optional parameter (*"print"*) is used to obtain the extracted correlations in the tabular form.

In[22] := **GroupCC[8, {"electronMass", 38, "boltzmannConstant", 89},**
dataBase, "print"];

i	j	x_i	x_j	r_{ij}
8	30	μ_B	m_e	0.991
8	38	μ_B	α	0.106
8	11	μ_B	k	0.096
8	89	μ_B	R_∞	–0.002

4.2.4. MakeCM

This function transforms the correlation table of the indicated database into numerical correlation matrix for all non exact constants. The function formats are as follows:

MakeCM[*data_ base*]
MakeCM[*data_ base,* "*print*"],

where *data_ base* is the database name and the second optional parameter (*"print"*) used to present correlation matrix in the table form.

In[23] := **MakeCM[wdb]**

Out[23] = {{1., 0.01, –0.017}, {0.01, 1., 0.}, {–0.017, 0., 1.}}

Here is the same matrix in the tabular form:

In[24] := **MakeCM[wdb, "print"];**

	α	h	R_∞
α	1.000	0.010	–0.017
h	0.010	1.000	0.000
R_∞	–0.017	0.000	1.000

In addition MakeCM may be used to construct correlation matrix for the constants selected from the on-line main database (dataBase). For this the list of constants for which the correlation matrix is to be constructed should be indicated as the first parameter value.

In[25] := **MakeCM[{"muonMagneticMomentAnomaly",**
"electronMagneticMoment","muonMass"}, "print"];

	μ_e	a_μ	m_μ
μ_e	1.000	–0.001	–0.985
a_μ	–0.001	1.000	0.008
m_μ	–0.985	0.008	1.000

4.3. Calculators

This set of functions is to calculate: average values of the algebraic functions depending on the physical constants; estimates of their variances or covariances from the uncertainties of the constants and their correlations using the standard error propagation technique.

ErrorPropagator	calculates the average value, the uncertainty and relative uncertainty of the single algebraic function of the constants.
MatrixCharacteristics	calculates the eigenvalues, determinant and condition number, testing the positive definiteness of an indicated matrix; it calculates the threshold accuracy for the safety independent rounding of the positive definite matrix elements.
Covariator	calculates the covariance and correlation matrices for the algebraic vector functions depending upon the physical constants.

4.3.1. ErrorPropagator

This function is to calculate: the average value of an algebraic function depending on the physical constants, its uncertainty and relative uncertainty. In the beginning of calculation the ErrorPropagator tests a quality of the input correlation sub-matrix to be used in calculation and informs the user in case if it has negative eigenvalues. Function format is as follows:

ErrorPropagator[*expression, data_ base*],
ErrorPropagator[*expression, data_ base, {contr_ keys}*] ,

where *expression* is the expression or the symbol assigned to the expression earlier, *data_ base* is the name of the on-line database to be used in the calculations, and *{contr_ keys}* is the list of the control keys.

The average value of the function $\langle \boldsymbol{Q} \rangle$, represented by the (*expression*) is calculated by means of the standard *Mathematica* options on the basis of the average values of the FPC extracted in due course of calculations from the on-line database. Hence all FPC entering into the *expression* should be presented by their package names in *symbol* format.

Calculation of the standard deviation $\boldsymbol{u(Q)}$ is produced on the basis of the usual error propagation rule (see, e.g., [1], [9]):

$$u(Q)^2 = \sum_{i=1}^{N} \left(\frac{\partial Q}{\partial x_i}\right)^2 u(c_i)^2 + \sum_{i \neq j}^{N} \frac{\partial Q}{\partial x_i}\frac{\partial Q}{\partial x_j} u(c_i)u(c_j)r_{ij},$$

where the partial derivatives are obtained symbolically and calculated at values $x_i = c_i$; $u(c_i)$ is the standard deviation of the FPC c_i; r_{ij} is the corresponding element of the input correlation sub-matrix r.

If the input correlation sub-matrix is the positive semidefinite the function ErrorPropagator prints the standard representation of the calculated value of the function:

$$\{\ mean\_value,\ uncertainty,\ relative\_uncertainty\ \},$$

where the standard deviation (*uncertainty*) is calculated as $u_(Q) = \sqrt{u(Q)^2}$, and the relative standard deviation (*relative_uncertainty*) as $u_r(Q) = u(Q)/\langle Q\rangle$, $\langle Q\rangle \neq 0$.

If the input correlation matrix (r) has the negative eigenvalues the ErrorPropagator calculates two pairs of values one with input correlation sub-matrix and the other with identity correlation matrix. In this case the output will contain the input correlation matrix and both values:

$$\begin{array}{l} \{\ \{non\text{-}positive\ semidefinite\ submatrix\}, \\ \ \{\{mean\_value,\ uncertainty,\ relative\_uncertainty\}, \\ \ \ \{mean\_value,\ uncert\_1,\ relative\_uncert\_1\}\}\ \}. \end{array}$$

The third (optional) parameter of the function ErrorPropagator in general case is the two-elements list of the "control keys"

The first key controls the changes of the matrix elements of the input correlation matrix. The second element (key) to select the output forms. Their possible values are collected in the following table.

contr_keys[[1]]	**Action**
is empty	to all "Null"-valued matrix elements of the input correlation matrix the zero value (0) will be assigned.
0	the input correlation matrix is replaced by corresponding identity matrix.
"Null"	all matrix elements with values r_{ij} = "Null" are replaced by the special package function cC[i,j], for which any desirable value could be assigned.
{{i, j, value}, ...}	indicated matrix elements (r_{ij}) will be replaced by new indicated values.
contr_keys[[2]]	**Action**
is empty	standard output: {*mean_value, uncertainty, relative_uncertainty*}.
"CM"	the input correlation matrix will be added to the standard output.
"EM"	the "error matrix" will be added to the standard output; the matrix elements of this "error matrix" are the individual terms in the sum for the $u(Q)^2$.
"All"	the standard output will be added with both matrices.

Example:

Calculation of the values for the Bohr magneton by the ErrorPropagator using two equivalent formulae:

$$\mu_B = \left(\frac{c_0 \alpha^5 h}{32\pi^2 \mu_0 R_\infty^2}\right)^{1/2} = \frac{e\ h}{4\pi m_e}.$$

These calculations will illustrate a work of ErrorPropagator with the different input correlation matrices.

In[26] := **bm1 = Sqrt[speedOfLightInVacuum fineStructureConstant^5 * planckConstant / (32 π^2 magneticConstant rydbergConstant^2)];**

In[27] := **bv1 = PowerExpand[ErrorPropagator[bm1, dataBase] /. Volt -> Joule Ampere^{-1} Second^{-1} /. Ampere -> Joule Tesla^{-1} Meter^{-2}]**

Out[27] $= \{\ \frac{9.2740095\times10^{-24}\ \text{Joule}}{\text{Tesla}},\ \frac{7.74319\times10^{-31}\ \text{Joule}}{\text{Tesla}},\ 8.34935\times10^{-8}\ \}$

Hereinafter all numerical results are presented in the standard form of the *Mathematica* system. By the *Mathematica* standard option SetPrecision one can output the other number of the significant digits:

In[28] := **SetPrecision[First[bv1] * Tesla/Joule, 9] * Joule/Tesla**

Out[28] $= \frac{9.27400948\ \times\ 10^{-24}\ \text{Joule}}{\text{Tesla}}$

Calculations based on the second formula for Bohr magneton resulted in:

In[29] := **bm2 = elementaryCharge planckConstant / (4 π electronMass);**

In[30] := **bv2 = ErrorPropagator[bm2, dataBase] /. Coulomb -> kGram Tesla^{-1} Second^{-1}**

```
WARNING: correlation matrix is non-positive semidefinite.
         Minimum eigenvalue of this matrix is -0.000154812
```

Out[30] $= \{\ \{\{1.,\ 0.998,\ 0.999\},\ \{0.998,\ 1.,\ 1.\},\ \{0.999,\ 1.,\ 1.\}\},$
$\{\{\frac{9.274009\times10^{-24}\ \text{Joule}}{\text{Tesla}},\ \frac{7.28149\times10^{-31}\ \text{Joule}}{\text{Tesla}},\ 7.85151\times10^{-8}\},$
$\{\frac{9.274009\times10^{-24}\ \text{Joule}}{\text{Tesla}},\ \frac{2.38336\times10^{-30}\ \text{Joule}}{\text{Tesla}},\ 2.56993\times10^{-7}\}\}\ \}$

In[31] := **SetPrecision[First[First[Last[bv2]]] * Tesla/Joule, 9] * Joule/Tesla**

Out[31] $= \frac{9.27400941\ \times\ 10^{-24}\ \text{Joule}}{\text{Tesla}}$

Our calculations failed to reproduce the CODATA-2002 value of the Bohr magneton using data recommended by CODATA in 2002 release [2].

$$\boldsymbol{\mu_B = 9.274\,009\,49\,(80) \times 10^{-24}\,J\,T^{-1}.}$$

It should be noted that this failure could not be attributed to the problem of the calculation precision of the *Mathematica* numerical procedures. Most probably the origin of the failure is the lack of accuracy in the correlation coefficients recommended by CODATA [2]. This is demonstrated more sharply by the results of calculations with the second expression for the Bohr magneton.

In this case ErrorPropagator has informed us on the non positive semi-definiteness of the correlation sub-matrix. The output of the result in this case includes the correlation matrix and two values of the calculated quantity. The first value is obtained with the incorrect input matrix (we will not discuss it further), and the second value is obtained with identity matrix. In case of identity matrix the obtained uncertainty is three times larger than the recommended uncertainty.

With the help of the control keys one can to perform calculations with the identity matrix or output correlation matrix to learn it's properties:

In[32] := **bv1 = PowerExpand[ErrorPropagator[bm1, dataBase, {,"CM"}] /.**
Volt -> Joule Ampere^{-1} Second^{-1}/. Ampere -> Joule Tesla^{-1} Meter^{-2}]

Out[32] = {{{1., 0.01, –0.017}, {0.01, 1., 0.}, {–0.017, 0., 1.}},
$\{\frac{9.2740095\times10^{-24}\ \text{Joule}}{\text{Tesla}}, \frac{7.74319\times10^{-31}\ \text{Joule}}{\text{Tesla}}, 8.34935 \times 10^{-8}\}\}$

Examples how to investigate the properties of the correlation matrices, and particularly the used above matrices will be presented in the next section.

4.3.2. MatrixCharacteristics

This function is to clarify properties of the square numeric matrices such as: eigenvalues, condition number and its determinant that are crucial for the majority applications. It is called by the package function Covariator (the description of it is presented further) to control the quality of the correlation matrices but it might be useful in the other applications also. The formats of the MatrixCharacteristics function are as follows:

MatrixCharacteristics[*matrix*],
MatrixCharacteristics[*matrix,* "*matrix_name*"] .

The output information is presented by a list:

{ *condition_number, determinant, rounding_index, eigenvalues_list* }.

Using the list of the $n \times n$-matrix eigenvalues ($\lambda_i,\ i = \overline{1,n}$) which are produced by the *Mathematica* built-in function Eigenvalues[matrix], the MatrixCharacteristics function calculates: condition number $|\lambda_{max}|/|\lambda_{min}|$ and determinant $\prod_{i=1}^{n} \lambda_i$.

If the given matrix is positive definite the threshold accuracy for the safety independent rounding of the matrix elements [11] is also calculated as:

$$\boldsymbol{I_{min} = \left\lceil \log_{10} \left(\frac{n-1}{2\ \lambda_{min}} \right) \right\rceil .}$$

I_{min} gives the minimal accuracy for all off-diagonal matrix elements of the positive definite matrix that will guarantee the preservation of the positive definiteness when rounding the given matrix independently and uniformly. In case of non positive definite matrix this index did not calculated and the function prints the corresponding warning.

Examples:

The correlation matrix used in calculation the standard deviation of the Bohr magneton value with the first formula (see previous section) is as follows:

	α	h	R_∞
α	1.000	0.010	–0.017
h	0.010	1.000	0.000
R_∞	–0.017	0.000	1.000

It has the following characteristic numbers:

In[33] := **MatrixCharacteristics[First[bv1], "correlation matrix (1-st formula)"]**

Characteristics of correlation matrix (1-st formula)

1) Condition number = 1.04024
2) Determinant = 0.999611
3) Rounding criterion: Imin = 1

Out[33] = {1.04024, 0.999611, 1, {1.01972, 1., 0.980277}}

The input correlation matrix for the second formula reads:

	m_e	e	h
m_e	1.000	0.998	0.999
e	0.998	1.000	1.000
h	0.999	1.000	1.000

and the ErrorPropagator "informed" earlier that it is non positive definite.

In[34] := **MatrixCharacteristics[First[bv2], "correlation matrix (2-d formula)"];**

Characteristics of correlation matrix (2-d formula)

1) Condition number = 19365.5
2) Determinant = –1. × 10^{-6}
3) WARNING: the matrix is non-positive semidefinite
`(Minimum eigenvalue of this matrix is -0.000154812)`

In this example using the symbol "**;**" we have restricted the automatic output of text information only.

4.3.3. Covariator

This function is complementary to the ErrorPropagator. It calculates the covariance and correlation matrices for the vector functions depending on the FPC from the on-line FPC working database. The formats of the function Covariator are as follows:

Covariator[*expr_ list, data_ base*],
Covariator[*expr_ list, data_ base, {contr_ keys}*] ,

where *expr_ list* is the list of the algebraic expressions for the components of the vector functions for which the covariance (correlation) matrices are to be evaluated or the name assigned to the such list. The structure of the *expr_ list* is as follows:

$exprlist$ = { {$outputc_1$, $expr_1$}, {$outputc_2$, $expr_2$}, ..., {$outputc_n$, $expr_n$} }.

In other words this list is the chain of the relationships:

$outputc_1 = expr_1$, $outputc_2 = expr_2$, ...,

there may be identities among them ($constant_i \equiv constant_i$).

The second parameter (*data_ base*) is the name of the on-line database containing information on all input quantities: recommended FPC entering into the $expr_i$ and data on the quantities $outputc_i$ for which covariances (correlations) are to be calculated. Note that the $outputc_i$ in the working database should have all components filled except the correlations. Even if they are known it will be better to calculate them with the help of the ErrorPropagator function to control the calculational precision and then include the obtained values into the working FPC database [2].

The last (optional) parameter *{contr_ keys}* will be described in the text to follow.

The function Covariator corresponds to the standard "error propagation" procedure [1, 9]. It propagate covariance matrix of the input quantities $\boldsymbol{c_i}$ $\boldsymbol{(i = 1, 2, ..., M)}$ (independent variables) "into" the covariance matrix of the calculated physical quantities (dependent variables) $\boldsymbol{P_j(c_1, c_2, ..., c_M)}$, где $\boldsymbol{j = 1, 2, ..., N}$; $\boldsymbol{N \geq M}$:

$$\boldsymbol{u(P_k, P_l) = \sum_{i,j=1}^{M} \frac{\partial P_k}{\partial x_i} \frac{\partial P_l}{\partial x_j} u(c_i) u(c_j) r_{ij},}$$

[2] Discomfort caused by the obligatory insertion the additional (temporal) data into the working database before calculating the covariances (correlations) will be resolved in the next generations of the package.

Here the partial derivatives are calculated symbolically and numerically evaluated at values $\boldsymbol{x_i = c_i}$ of the input variables;
$\boldsymbol{u(c_i)}$ is the standard deviation of the FPC $\boldsymbol{c_i}$;
$\boldsymbol{r_{ij}}$ is the element of the input correlation matrix.

Actually Covariator function calculates the *relative* covariances ($\boldsymbol{u_r(P_k, P_l) = u(P_k, P_l)/(P_k\, P_l)}$) and variances ($\boldsymbol{u_r^2(P_k) = u_r(P_k, P_k) = u(P_k, P_k)/P_k^2}$), then the correlation coefficients:

$$\boldsymbol{r(P_k, P_l) = \frac{u_r(P_k, P_l)}{\sqrt{u_r^2(P_k)\, u_r^2(P_l)}}.}$$

The Covariator also checks if the input correlation matrix is the positive semi-definite matrix (using the MatrixCharacteristics).

If the input matrix is the positive semi-definite matrix then the Covariator performs the standard calculation of the uncertainty correlation matrix for the indicated dependent physical quantities from the indicated database.

Otherwise if the input matrix has negative eigenvalues the calculations of the covariances and correlations are performed twice (as in the case of the ErrorPropagator function): with input correlations and with the corresponding identity matrix. Hence in the output it will be two matrices. Besides Covariator gives by default the lists of the input and output quantities and their correlation matrices in the tabular forms.

Examples:
Calculations of the correlations for the Bohr magneton with the other FPC entering into the first formula (see Subsection 4.3.1) which in terms of the package names of the constants has the form:

In[35] := **bm = Sqrt[speedOfLightInVacuum fineStructureConstant^5 ***
planckConstant / (32 π^2 magneticConstant rydbergConstant^2)];

Definition of the list of the dependent vector-function components for which we want to calculate correlation matrix:

In[36] := **listbm = { {bohrMagneton, bm},**
{fineStructureConstant, fineStructureConstant},
{planckConstant, planckConstant},
{rydbergConstant, rydbergConstant} };

Finally let us initiate the calculation of the covariances (correlations):

In[37] := **cmbm = Covariator[listbm, dataBase]**

```
Input constants:  {fineStructureConstant, magneticConstant,
                   planckConstant, rydbergConstant, speedOfLightInVacuum}

Output constants:  {bohrMagneton, fineStructureConstant, planckConstant,
                    rydbergConstant}
```

	α	h	R_∞
α	1.000	0.010	–0.017
h	0.010	1.000	0.000
R_∞	–0.017	0.000	1.000

Characteristics of input correlation matrix
1) Condition number = 1.04024
2) Determinant = 0.999611
3) Rounding criterion: Imin = 1

	$\boldsymbol{\mu_B}$	$\boldsymbol{\alpha}$	$\boldsymbol{h}$	$\boldsymbol{R_\infty}$
$\boldsymbol{\mu_B}$	1.	0.10842	0.99514	–0.00175378
$\boldsymbol{\alpha}$	0.10842	1.	0.01	–0.017
$\boldsymbol{h}$	0.99514	0.01	1.	0.
$\boldsymbol{R_\infty}$	–0.00175378	–0.017	0.	1.

Characteristics of output correlation matrix
1) Condition number = 6.01129 × 10^{15}
2) Determinant = 6.65223 × 10^{-16}
3) Rounding criterion: Imin = 16

Out[37] = { {$\boldsymbol{\mu_B}$, $\boldsymbol{\alpha}$, $\boldsymbol{h}$, $\boldsymbol{R_\infty}$},
{{1., 0.10842, 0.99514, –0.00175378}, {0.10842, 1., 0.01, –0.017},
{0.99514, 0.01, 1., 0.}, {–0.00175378, –0.017, 0., 1.}} }

A few comments on the output should be made.

The list of the input constants turned to be larger then the list of the output constants as the two exact constants, participating in the expression of the dependent constant (Bohr magneton), were added to the first list.

After the correlation matrix of the input constants it follows the output correlation matrix that should be compared with the recommended sub-matrix

	$\boldsymbol{\mu_B}$	$\boldsymbol{\alpha}$	$\boldsymbol{h}$	$\boldsymbol{R_\infty}$
$\boldsymbol{\mu_B}$	1.	0.106	0.995	–0.002
$\boldsymbol{\alpha}$	0.106	1.	0.01	–0.017
$\boldsymbol{h}$	0.995	0.01	1.	0.
$\boldsymbol{R_\infty}$	–0.002	–0.017	0.	1.

Characteristics of the recommended correlation matrix
1) Condition number = 5343.7
2) Determinant = 0.000748049
3) Rounding criterion: Imin = 4

The corresponding matrix elements of these two matrices do not differ much (at the first glance), but characteristic numbers for the calculated matrix turned to be much worst that of the recommended one. We connect this with the difference in the orders of the evaluation and rounding of the correlation matrix elements in our case.

It should be noted that recommended sub-matrix turned to be over-rounded as the threshold accuracy is $I_{min} = 4$.

The third (optional) parameter (*contr_keys*) of the Covariator is the list of the two elements. The first key manages the values of the input correlation matrix (as it is in the case of the ErrorPropagator function). The second key is to control the output information and it can take the values:

contr_keys[[2]]	**Action**
is empty	the standard output of the correlation matrix.
"CVM"	the output of the matrix of relative covariances.
"Mix"	the output of the combined matrix with replacement of the down off-diagonal matrix elements of the relative covariances matrix with the correlation coefficients.
"All"	the output of all three above matrices.

Note that the matrix elements of the relative covariances are scaled to the its minimal element. The scale factor also is presented in the output giving the possibility to restore the true matrix of the relative covariances.

5. Summary

As it was emphasized in the Introduction the package StandardPhysicalConstants is under development. The short term plan for the upgrade and further development is to address the following problems:

- The problem of the quality assurance of the package FPC database is of high priority. It is the most hard problem, because it is out of our control for a while. Unfortunately we did not get any answers from the producers of the CODATA recommended FPC to our proposals to correct the recommended values of correlations and to optimize the web access to the numerical FPC data. It seems that the solution could be found on the way of "independent FPC evaluation and adjustments" as it is outlined in [11].
- The problem to organize the correct and comfortable handling the dependencies between dimensional quantities on the internal (package) and external (user) levels.
- The problem on the reliability of the calculations with the complex data structures used in the package.

References

[1] Mohr, P. and Taylor B. *CODATA recommended values of the fundamental physical constants: 1998*, in Rev. Mod. Phys., **72**, 351, (April 2000), The American Physical Society;

http://physics.nist.gov/cuu/Constants/archive1998.html

Unfortunately, in the archived version (V.3.2) the correlation coefficients are omitted.

[2] Mohr, P. and Taylor B. *CODATA recommended values of the fundamental physical constants: 2002*, http://physics.nist.gov/cuu/Constants/.

[3] Ezhela, V., Larin, V. *The Development of the Mathematica Package* 'StandardPhysicalConstants'*,* in Proc. of the 5th International *Mathematica* Symposium (eds. P.Mitic, Ph.Ramsden, J.Carne), Imperial College Press, London, 2003, pp. 207–214; IHEP Preprint 2003-17, Protvino, 2003.

[4] http://library.wolfram.com/howtos/constants/.

[5] http://www.mapleapps.com/categories/maple8/html/scientificconstants.html.

[6] The MAXIMA computer algebra system, "physconst" package. `http://maxima.sourceforge.net/.`

[7] Ezhela, V. and Larin, V. *The physical constants for Mathematica*, in Proc. of the fourth International *Mathematica* Symposium (eds. Y.Tazawa, S.Sakakibara, S.Ohashi, Y.Uemura) Tokyo Denki University Press, June 2001, pp. 263–268; IHEP Preprint 2001-27, Protvino, 2001.

[8] Wolfram, S. *The Mathematica Book*. 4th ed. Wolfram Media/Cambridge University Press, 1999.

[9] S. Eidelman, K.G. Hayes, K.A. Olive, et al., *Review of particle physics* [Particle Data Group], Phys. Lett. **B592** (2004) 1.

[10] Hagiwara, K. et al., *Review of particle physics.* [Particle Data Group], in Phys. Rev., **D66**, 010001-1, (July 2002), The American Physical Society.

[11] Siver, A.S. and Ezhela, V.V. *On the CODATA Recommended Values of the Fundamental Physical Constants: V.3.2(1998) & V.4.0(2002).* IHEP Preprint 2003-34, Protvino, 2003.

Received June 29, 2004.

6. Appendix

Names of the constants in the Dbfpc113.m

Code	**Package name** (*basicTable*)	**Standard names** (*standardNames*)
1	accelerationOfGravity	standard acceleration of gravity
2	ageOfUniverse	age of the universe
3	ageOfUniverseInGYear	age of the universe in Gyear
4	astronomicalUnit	astronomical unit (mean $\oplus - \odot$ distance)
5	atomicMassConstant	atomic mass constant
6	avogadroConstant	Avogadro constant
7	baryonDensityOfUniverse	baryon density of the universe
8	bohrMagneton	Bohr magneton
9	bohrMagnetonIneV	Bohr magneton in eV/T
10	bohrRadius	Bohr radius
11	boltzmannConstant	Boltzmann constant
12	classicalElectronRadius	classical electron radius
13	comptonWavelength	Compton wavelength
14	comptonWavelengthReduced	Compton wavelength over 2π
15	conductanceQuantum	conductance quantum
16	constantOfGravitation	Newtonian constant of gravitation
17	conversionConstant	Planck constant over 2π times c in $MeV\ fm$
18	conversionConstantSquare	conversion constant square
19	cosmicBackgroundTemperature	present day CBR temperature
20	deuteronMagneticMoment	deuteron magnetic moment
21	deuteronMass	deuteron mass
22	earthMass	Earth mass
23	earthRadius	Earth mean equatorial radius
24	earthSchwarzschildRadius	Schwarzschild radius of the Earth
25	electricConstant	electric constant
26	electronChargeToMassQuotient	electron charge to mass quotient
27	electronGFactor	electron g-factor
28	electronMagneticMoment	electron magnetic moment
29	electronMagneticMomentAnomaly	electron magnetic moment anomaly
30	electronMass	electron mass
31	electronMassInMeV	electron mass energy equivalent in MeV
32	electronMuonMagneticMomentRatio	electron-muon magnetic moment ratio
33	electronMuonMassRatio	electron-muon mass ratio
34	elementaryCharge	elementary charge
35	eVToKilogramRelationship	electron volt-kilogram relationship
36	faradayConstant	Faraday constant
37	fermiCouplingConstant	Fermi coupling constant
38	fineStructureConstant	fine-structure constant

Code	**Package name** (*basicTable*)	**Standard name** (*standardNames*)
39	galacticUnit	solar distance from galactic center
40	galacticUnitInParsec	solar distance from galactic center in parsec
41	galaxyVelocityWithRespectToCBR	Local group velocity with respect to CBR
42	hubbleConstant	present day Hubble expansion rate in SI
43	icePoint	ice point (standard temperature)
44	impedanceOfVacuum	characteristic impedance of vacuum
45	inverseFineStructureConstant	inverse fine-structure constant
46	jansky	Jansky
47	localDiskDensity	local disk density
48	localDiskDensityInGeV	local disk density in GeV
49	localHaloDensity	local halo density
50	localHaloDensityInGeV	local halo density in GeV
51	magneticConstant	magnetic constant
52	magneticFluxQuantum	magnetic flux quantum
53	meanSiderealDay	mean sidereal day (2005.0)
54	molarGasConstant	molar gas constant
55	molarVolume	molar volume of ideal gas ($273.15K$, $101.325\ kPa$)
56	muonComptonWavelength	muon Compton wavelength
57	muonComptonWavelengthReduced	muon Compton wavelength over 2π
58	muonElectronMassRatio	muon-electron mass ratio
59	muonGFactor	muon g-factor
60	muonMagneticMoment	muon magnetic moment
61	muonMagneticMomentAnomaly	muon magnetic moment anomaly
62	muonMagneticMomentToBohrMagneton	muon magnetic moment to Bohr magneton ratio
63	muonMass	muon mass
64	muonMassInMeV	muon mass energy equivalent in MeV
65	muonProtonMagneticMomentRatio	muon-proton magnetic moment ratio
66	muonProtonMassRatio	muon-proton mass ratio
67	neutronComptonWavelength	neutron Compton wavelength
68	neutronMagneticMoment	neutron magnetic moment
69	neutronMass	neutron mass
70	normalizedBaryonDensity	baryon-to-photon ratio
71	nuclearMagneton	nuclear magneton
72	nuclearMagnetonIneV	nuclear magneton in eV/T
73	parsec	parsec (1 AU/1 arc sec)
74	planckConstant	Planck constant
75	planckConstantInMeV	Planck constant in $eV\ s$
76	planckConstantReduced	Planck constant over 2π
77	planckConstantReducedInMeV	Planck constant over 2π in $eV\,s$

Code	**Package name** (*basicTable*)	**Standard name** (*standardNames*)
78	planckLength	Planck length
79	planckMass	Planck mass
80	planckMassInGeV	Planck mass energy equivalent in GeV
81	planckTime	Planck time
82	pressurelessMatterDensity	pressureless matter density of the universe
83	protonChargeToMassQuotient	proton charge to mass quotient
84	protonComptonWavelength	proton Compton wavelength
85	protonMagneticMoment	proton magnetic moment
86	protonMagneticMomentToBohrMagneton	proton magnetic moment to Bohr magneton ratio
87	protonMass	proton mass
88	protonNeutronMassRatio	proton-neutron mass ratio
89	rydbergConstant	Rydberg constant
90	rydbergEnergy	Rydberg constant times hc in eV
91	rydbergFrequency	Rydberg constant times c in Hz
92	sackurTetrodeConstant	Sackur-Tetrode constant $(1K, 101.325\,kPa)$
93	scaledCosmologicalConstant	dark energy density
94	siderealYear	sidereal year (fixed star to fixed star) (2005.0)
95	solarConstant	mean total solar irradiance (TSI)
96	solarLuminosity	solar luminosity
97	solarMass	solar mass
98	solarRadius	solar equatorial radius
99	solarSchwarzschildRadius	Schwarzschild radius of the Sun
100	solarVelocityAroundGalaxyCenter	solar velocity around center of Galaxy
101	solarVelocityWithRespectToCBR	solar velocity with respect to CBR
102	speedOfLightInVacuum	speed of light in vacuum
103	speedOfSound	speed of sound
104	stefanBoltzmannConstant	Stefan-Boltzmann constant
105	strongCouplingConstant	strong coupling constant
106	tauMass	tau mass
107	thomsonCrossSection	Thomson cross section
108	tropicalYear	tropical year (equinox to equinox) (2005.0)
109	wavelengthOfeVOverc0Particle	wavelength of $1eV/c$ particle
110	wBosonMass	$W^{\pm}$ boson mass
111	weakMixingAngle	weak mixing angle
112	wienDisplacementLawConstant	Wien displacement law constant
113	zBosonMass	Z^0 boson mass

Препринт отпечатан с оригинала-макета, подготовленного авторами.

В.В. Ежела, В.Н. Ларин.
Пакет StandardPhysicalConstants 2.1. (Руководство для пользователя).

Оригинал-макет подготовлен с помощью системы **LaTeX.**

Подписано к печати 30.06.2004 Формат 60 × 84/8.
Офсетная печать. Печ.л. 3,12. Уч.-изд.л. 2,5. Тираж 60. Заказ 282.
Индекс 3649.

ГНЦ РФ Институт физики высоких энергий
142284, Протвино Московской обл.

Индекс 3649

П Р Е П Р И Н Т 2004–24, И Ф В Э, 2004