DESY 16-019
January 2016

ISSN 0418-9833

mr: a C++ library for the matching and running of the Standard Model parameters

Bernd A. Kniehl[a], Andrey F. Pikelner[a,b,*], Oleg L. Veretin[a]

[a] *II. Institut für Theoretische Physik, Universität Hamburg, Luruper Chaussee 149, 22761 Hamburg, Germany*
[b] *Bogoliubov Laboratory of Theoretical Physics, Joint Institute for Nuclear Research, 141980 Dubna, Russia*

Abstract

We present the C++ program library mr that allows us to reliably calculate the values of the running parameters in the Standard Model at high energy scales. The initial conditions are obtained by relating the running parameters in the $\overline{\rm MS}$ renormalization scheme to observables at lower energies with full two-loop precision. The evolution is then performed in accordance with the renormalization group equations with full three-loop precision. Pure QCD corrections to the matching and running are included through four loops. We also provide a Mathematica interface for this program library.

Keywords: Particle physics, Standard Model, Running parameters, Renormalization group evolution, Threshold corrections

*Corresponding author
Email addresses: kniehl@desy.de (Bernd A. Kniehl), andrey.pikelner@desy.de (Andrey F. Pikelner), oleg.veretin@desy.de (Oleg L. Veretin)

PROGRAM SUMMARY

Manuscript title: `mr`: a `C++` library for the matching and running of the Standard Model parameters

Authors: B. A. Kniehl, A. F. Pikelner, O. L. Veretin

Program title: `mr`

Journal reference:

Catalogue identifier:

Licensing provisions: GPLv3

Programming language: `C++`

Computer: IBM PC

Operating system: Linux, Mac OS X

RAM: 1 GB

Number of processors used: available with `OpenMP`

Supplementary material:

Keywords: particle physics; Standard Model; running parameters; renormalization group evolution; threshold corrections

Classification: 11 Elementary Particle Physics, 11.1 General, High Energy Physics and Computing

External routines/libraries: `TSIL` [1], `OdeInt` [2]

Nature of problem:

The running parameters of the Standard Model renormalized in the $\overline{\rm MS}$ scheme at some high renormalization scale, which is chosen by the user, are evaluated in perturbation theory as precisely as possible in two steps. First, the initial conditions at the electroweak energy scale are evaluated from the Fermi constant G_F and the pole masses of the W, Z, and Higgs bosons and the bottom and top quarks including the full two-loop threshold corrections. Second, the evolution to the high energy scale is performed by numerically solving the renormalization group evolution equations through three loops. Pure QCD corrections to the matching and running are included through four loops.

Solution method:

numerical integration of analytic expressions

Restrictions:

Unusual features:

Additional comments:

Available for download from URL: http://apik.github.io/mr/

Running time:

less than 1 second

References

[1] S. P. Martin and D. G. Robertson, Comput. Phys. Commun. **174** (2006) 133–151 [hep-ph/0501132].

[2] K. Ahnert and M. Mulansky, AIP Conf. Proc. **1389** (2011) 1586–1589 [arxiv:1110.3397 [cs.MS]].

1. Introduction

The Standard Model (SM) exhibits excellent agreement with an enormous wealth of experimental data. It is supposed to describe particle physics from low energies up to tera-electron-volt scales and beyond. Moreover, the recent discovery of a Higgs-like particle [1, 2] makes the SM one of the best candidates for the theory of the electroweak (EW) and strong interactions. The absence of clear signals for *new physics*, e.g., supersymmetry, at the LHC leads us to the question: how far up in the energy scale can the SM be extrapolated? The answer to this question depends, among numerous factors, on whether the EW vacuum remains stable at ultimatively high energies or not. Recent studies of the scalar potential show that, with the Higgs boson mass being around 125 GeV, the SM can at most become metastable, in the sense that the lifetime of our EW vacuum exceeds the age of the Universe, when extrapolated up to the scales of the Planck mass [3, 4, 5, 6].

The analysis of the scalar potential at high energies basically requires two ingredients: (i) the renormalization group (RG) evolution of the running parameters and (ii) the initial conditions that relate the latter to the physical observables. These initial conditions, which are determined by the so-called threshold corrections, are usually taken at some lower energy scale, which is typically of the order of the masses of the weak gauge bosons or the top quark. For the consistent use of L-loop RG evolution, one should take into account at least $(L-1)$-loop matching.

It is the goal of this paper to introduce the `C++` program library `mr`, which is designed to conveniently perform the above analysis at the next-to-next-to-leading-order (NNLO) EW level. This means that we take into account the full two-loop threshold corrections and the full three-loop RG equations. In addition, we collect in our program library all the theoretical knowledge available to date, including the QCD β function and mass anomalous dimensions through four loops.

This paper is organized as follows. Section 2 contains the relevant definitions and notations concerning the parametrization of the SM. Section 3 provides a description of the `mr` program library. Section 4 explains how to use the `Mathematica` interface. Section 5 points the reader towards example programs. Section 6 contains our summary. Appendix A, Appendix B, and Appendix C list the methods needed for the on-shell (OS) mass input, running-parameter input, and the RG evolution, respectively, and Appendix D guides the reader through a typical example.

2. The physical setup

The SM can be described in terms of the following running parameters defined in the modified minimal-subtraction ($\overline{\text{MS}}$) renormalization scheme at the renormalization scale μ:

$$g_s(\mu), g(\mu), g'(\mu), y_t(\mu), y_b(\mu), \lambda(\mu), v(\mu), \tag{1}$$

where $g_s(\mu)$ is the strong gauge coupling, $g(\mu)$ and $g'(\mu)$ are the EW gauge couplings, $y_t(\mu)$ and $y_b(\mu)$ are Yukawa couplings of the top and bottom quarks, $\lambda(\mu)$ is the scalar self coupling, and the vacuum expectation value $v(\mu) = \sqrt{m_\phi^2(\mu)/(2\lambda(\mu))}$ is defined through $\lambda(\mu)$ and the Higgs mass parameter $m_\phi(\mu)$, which both appear in the scalar potential,

$$V(\phi) = -\frac{m_\phi^2}{2}\phi^\dagger\phi + \lambda(\phi^\dagger\phi)^2. \tag{2}$$

Here, the normalization of the quadratic term has been chosen so that $m_\phi^0 = m_H^0$ at tree level. In our analysis, we retain the Yukawa couplings only for the top and bottom quarks. The contributions due to the Yukawa couplings of the other fermions may be safely neglected; those fermions practically only contribute through their interactions with the gauge bosons. In addition to the parameters in Eq. (1), one may also consider the $\overline{\text{MS}}$ masses,

$$m_W(\mu), m_Z(\mu), m_H(\mu), m_t(\mu), m_b(\mu). \tag{3}$$

It is well known that, to preserve the gauge independence of the $\overline{\text{MS}}$ masses in Eq. (3), one has to systematically include the tadpole contributions (see, e.g., the detailed discussions in Refs. [7, 8, 9, 10, 11, 12]).

We first introduce the matching relations, which provide the initial conditions for the RG differential equations. These relations define the $\overline{\rm MS}$ parameters at some low-energy scale μ_0 in terms of appropriate input parameters. Specifically, our choice of input parameters is

$$\alpha_s(M_Z), G_F, M_W, M_Z, M_H, M_t, M_b, \tag{4}$$

where $\alpha_s(\mu) = g_s^2(\mu)/(4\pi)$ is the $\overline{\rm MS}$ strong-coupling constant, G_F is Fermi's constant, and M_W, M_Z, M_H, M_t, and M_b are the pole masses of the corresponding particles. At $\mu = M_Z$, $n_f = 5$ quark flavors are considered active.

It is also useful to introduce the $\overline{\rm MS}$ electromagnetic gauge coupling via [12]

$$\frac{1}{e^2(\mu)} = \frac{1}{g^2(\mu)} + \frac{1}{g'^2(\mu)}, \tag{5}$$

which in turn defines the $\overline{\rm MS}$ fine-structure constant as

$$\alpha(\mu) = \frac{e^2(\mu)}{4\pi}. \tag{6}$$

If the input in terms of the OS parameters in Eq. (4) is given, then the $\overline{\rm MS}$ couplings can be obtained at the matching scale μ_0. The corresponding matching relations are parametrized as

$$\begin{aligned} g^2(\mu_0) &= 2^{5/2} G_F M_W^2 [1 + \delta_W(\mu_0)], \\ g^2(\mu_0) + g'^2(\mu_0) &= 2^{5/2} G_F M_Z^2 [1 + \delta_Z(\mu_0)], \\ \lambda(\mu_0) &= 2^{-1/2} G_F M_H^2 [1 + \delta_H(\mu_0)], \\ y_f(\mu_0) &= 2^{3/4} G_F^{1/2} M_f [1 + \delta_f(\mu_0)], \end{aligned} \tag{7}$$

where $f = t, b$. In Eq. (7), $\delta_x(\mu)$ are complicated functions of the OS parameters in Eq. (4) and μ_0, which may be expanded as perturbation series,

$$\delta_x(\mu) = \sum_{i,j} \left(\frac{\alpha(\mu)}{4\pi}\right)^i \left(\frac{\alpha_s(\mu)}{4\pi}\right)^j Y_x^{i,j}(\mu). \tag{8}$$

The expansion coefficients $Y_x^{i,j}(\mu)$ are generally available for $i, j = 1, 2$, which corresponds to two-loop matching. Beyond that, the pure QCD corrections[1] are known through four loops and are given by $Y_f^{0,3}(\mu)$ and $Y_f^{0,4}(\mu)$.

[1] In the bosonic cases $x = W, Z, H$, the pure QCD contributions vanish identically, $Y_x^{0,j}(\mu) = 0$.

In addition to Eq. (7) for the couplings, we also present the matching relations for the masses, which we write as

$$\begin{aligned} m_B^2(\mu_0) &= M_B^2[1+\Delta_B(\mu_0)], \qquad B = W, Z, H, \\ m_f(\mu_0) &= M_f[1+\Delta_f(\mu_0)], \qquad f = t, b, \end{aligned} \tag{9}$$

for the bosons and fermions, respectively. The functions $\Delta_x(\mu)$ may also be expanded in perturbation series,

$$\Delta_x(\mu) = \sum_{i,j} \left(\frac{\alpha(\mu)}{4\pi}\right)^i \left(\frac{\alpha_s(\mu)}{4\pi}\right)^j X_x^{i,j}(\mu). \tag{10}$$

The comments made below Eq. (8) also apply to Eq. (10).

The pure QCD corrections to $m_t(\mu)$ and $y_t(\mu)$ are evaluated according to Eq. (59) in Ref. [12] with $n_l = 5$ massless quarks and $n_h = 1$ massive quark. For the pure QCD corrections to $m_b(\mu)$ and $y_b(\mu)$, to take into account diagrams with top-quark insertions, we treat the four lightest quarks as massless, but keep both the top and bottom quarks massive through two loops. The corresponding analytic formula reads [13, 14]:

$$\begin{aligned} \delta_{\rm QCD}(\mu) &= \frac{\alpha_s}{4\pi}\left(-\frac{16}{3} - 4l_{\mu M_b}\right) + \left(\frac{\alpha_s}{4\pi}\right)^2 \left\{-\frac{3305}{18} - \frac{64\zeta_2}{3} + \frac{8\zeta_3}{3}\right. \\ &\quad - \frac{32\zeta_2}{3}\ln 2 + n_h\left(\frac{143}{9} - \frac{32\zeta_2}{3}\right) + n_l\left(\frac{71}{9} + \frac{16\zeta_2}{3}\right) \\ &\quad + \frac{n_m}{9t^4}\left[72t^2 + 71t^4 - 48t^2 H_0(t) + 48(1+t+t^3+t^4)H_{-1,0}(t)\right. \\ &\quad \left. - 48H_{1,0}(t) - 96t^4 H_{0,0}(t) + 48t(1+t^2-t^3)H_{1,0}(t)\right] \\ &\quad + l_{\mu M_b}\left[-\frac{314}{3} + \frac{52}{9}(n_h+n_l+n_m)\right] \\ &\quad \left. + l_{\mu M_b}^2\left[-14 + \frac{4}{3}(n_h+n_l+n_m)\right]\right\}, \end{aligned} \tag{11}$$

where $n_l = 4$, $n_h = 1$, $n_m = 1$ refers to the massive bottom quark, $t = M_b/M_t$, $l_{\mu M_b} = \ln(\mu^2/M_b^2)$, and

$$\begin{aligned} H_0(t) &= \ln t, \qquad H_{-1,0}(t) = \ln t \ln(1+t) + {\rm Li}_2(-t), \\ H_{0,0}(t) &= \frac{\ln^2 t}{2}, \qquad H_{1,0}(t) = -\ln t \ln(1-t) - {\rm Li}_2(t) \end{aligned} \tag{12}$$

are harmonic polylogarithms as introduced in Ref. [15].

Finally, also the relations inverse to Eq. (9), which express the pole masses in terms of the $\overline{\rm MS}$ parameters, may be of interest [16, 17, 18],

$$\begin{aligned} M_b^2 &= m_B^2(\mu_0)\left[1+\overline{\Delta}_B(\mu_0)\right], \\ M_f &= m_f(\mu_0)\left[1+\overline{\Delta}_f(\mu_0)\right]. \end{aligned} \tag{13}$$

The functions $\overline{\Delta}_x(\mu)$ are parametrized similarly to Eq. (10), with expansion coefficients $\overline{X}_x^{i,j}(\mu)$. They are included in our `C++` program library as well.

The running of the $\overline{\rm MS}$ parameters in Eqs. (1) and (3) is governed by the RG equations,

$$\mu^2\frac{dx}{d\mu^2} = \beta_x, \qquad x = g_s, g, g', y_t, y_b, \lambda, \tag{14}$$

$$\mu^2\frac{d\ln x}{d\mu^2} = \gamma_x, \qquad x = m_W, m_Z, m_H, m_t, m_b, \tag{15}$$

with the respective β functions β_x and mass anomalous dimensions γ_x. Given the values of the parameters $x(\mu_0)$ at some initial scale μ_0, Eqs. (14) and (15) allow us to find their values at some high scale μ. Since β_x and γ_x are in turn functions of the $\overline{\rm MS}$ parameters in Eq. (1), the RG equations form a system of nonlinear differential equations to be solved simultaneously. The functions β_x and γ_x have been known through four loops in QCD for a long time [19, 20, 21, 22, 23] and have recently been computed through three loops in the full SM [24, 25, 26, 27, 28, 29, 30, 31].

3. `C++` library `mr`

The program library provides the following ingredients:

- evaluation of the coefficients $Y_x^{i,j}(\mu)$ and $X_x^{i,j}(\mu)$ in Eqs. (8) and (10), respectively, for given input values of the OS parameters in Eq. (4) and, inversely, of the coefficients $\bar{X}_x^{i,j}(\mu)$ for the given input values of the $\overline{\rm MS}$ parameters in Eqs. (1) and (3),

- evaluation of the $\overline{\rm MS}$ couplings and masses according to Eqs. (7) and (9), respectively, as well as evaluation of the inverse relations in Eq. (13),

- evolution of the $\overline{\rm MS}$ parameters in the scale μ using the RG equations in Eqs. (14) and (15).

First, in order to use the C++ interface, one has to include the corresponding header file. All parts of the program library are placed in the namespace mr:

```
#include "mr.hpp"
using namespace mr;
```

3.1. Input parameters

Before evaluating the coefficients $X_x^{i,j}(\mu)$ and $Y_x^{i,j}(\mu)$, the input parameter class must be created. If we wish to use the OS parameters as input, we have to use the class OSinput with a constructor taking as arguments the five pole masses M_b, M_W, M_Z, M_H, and M_t (in GeV):

```
//          Mb     MW       MZ       MH      Mt
OSinput oi(4.4, 80.385, 91.1876, 125.7, 173.2);
```

For the user's convenience, we have predefined sets of input parameters from different editions of the Review of Particle Physics by the Particle Data Group (PDG) tabulated in enums named pdg20xx, where 20xx denotes the year of the edition. For example, we may construct an object using the OS input from the 2014 issue as:

```
OSinput oi(pdg2014::Mb, pdg2014::MW, pdg2014::MZ, pdg2014::MH,
   pdg2014::Mt);
```

For a detailed description of the OSinput class methods, see Appendix A.1.

If we are interested in the inverse relations, defining the pole masses in terms of the $\overline{\text{MS}}$ parameters, we have to use the class MSinput. It constructs an object from the $\overline{\text{MS}}$ masses (in GeV) (for a detailed description, see Appendix B.1):

```
//                          mb      mW       mZ       mH      mt
mi = MSinput::fromMasses(4.4, 80.385, 91.1876, 125.7, 173.2);
```

or from the $\overline{\text{MS}}$ coupling constants:

```
//                        g1      g2       yb       yt      lambda
mi = MSinput::fromCouplings(4.4, 80.385, 91.1876, 125.7, 173.2);
```

3.2. Evaluation of coefficients $X_x^{i,j}(\mu)$ and $Y_x^{i,j}(\mu)$

Having at hand the `OSinput` class instance, we may construct instances of classes providing the $\overline{\rm MS}$ masses and couplings for given input values of the pole masses:

```
bb<OS>   db(oi, oi.MMt());
WW<OS>   dW(oi, oi.MMt());
ZZ<OS>   dZ(oi, oi.MMt());
HH<OS>   dH(oi, oi.MMt());
tt<OS>   dt(oi, oi.MMt());
```

Here, the second parameter is the square of the matching scale μ_0; in the previous example, we have used $\mu_0^2 = M_t^2$. For the b, W, Z, H, and t initialization `bb<OS>`, etc., this means that we calculate the $\overline{\rm MS}$ parameters in terms of the OS ones.

When we wish to calculate the inverse relations in Eq. (13), we may specify the input values of the $\overline{\rm MS}$ masses as:

```
MSinput mi(mb, mW, mZ, mH, mt);
```

In contrast to the OS masses, which are denoted by uppercase letters, $\overline{\rm MS}$ masses are denoted by lowercase letters. The input parameters `mb`, `mW`, `mZ`, `mH`, and `mt` implicitly dependent on the scale μ. When the coefficients $\bar{X}_x^{i,j}(\mu)$ are to be calculated from the $\overline{\rm MS}$ masses, the scale μ at which the latter are defined has to be specified:

```
bb<MS> xb(mi, mi.mmt());
WW<MS> xW(mi, mi.mmt());
ZZ<MS> xZ(mi, mi.mmt());
HH<MS> xH(mi, mi.mmt());
tt<MS> xt(mi, mi.mmt());
```

Note that the scale μ is common to all the $\overline{\rm MS}$ masses. Here, we have chosen $\mu^2 = m_t^2(\mu)$.

For each type of particle and set of input parameters, the construction of the class instances `db`, `dW`, `dZ`, `dH`, and `dt` or `xb`, `xW`, `xZ`, `xH`, and `xt` is the most time-consuming operation. At this step, the prototypes of all master integrals are evaluated for the predefined values in the input masses.

After initialization, the expansion coefficients in Eqs. (8) and (10) become available for each particle x. One may refer to the coefficients $X_x^{i,j}(\mu)$ and

$Y_x^{i,j}(\mu)$ through the following methods:

```
long double <particle>::x(size_t a_order, size_t as_order)
long double <particle>::y(size_t a_order, size_t as_order)
```

Alternatively the calls `xij()` or `yij()` are also available, where i and j are integers. For example, the call `dZ.x(1,1)` is equivalent to `dZ.x11()`, and the call `dt.y(0,3)` is equivalent to `dt.y03()`, and so on.

3.3. Evaluation of corrections to $\overline{MS}$ coupling constants and masses

For the evaluation of the $\overline{\text{MS}}$ parameters from Eqs. (7) and (9) as implemented in our code, $\alpha_s(\mu)$ and $\alpha(\mu)$ have to be supplied at the chosen scale μ.

Let us first consider $\alpha_s(\mu)$. We implemented a routine for the evolution of $\alpha_s(\mu)$ to the desired scale μ, starting from some initial scale μ_0 and the initial value $\alpha_s(\mu_0)$, which is $\alpha_s(M_Z)$ by default. Sometimes, however, it is of interest to perform the matching at some different scale μ_0. If $\mu_0 \geq M_t$, the corresponding decoupling at the top-quark threshold M_t has to be taken into account. This is implemented in our code through order $O(\alpha_s^3)$ [32, 33]. In the following example, we perform the evolution of $\alpha_s(\mu)$ from the initial PDG value at $\mu_0 = M_Z$ up to the scale $\mu = 1000$ GeV crossing the top-quark threshold:

```
// Initial scale is oi.MZ() 4-loop running
AlphaS aS(oi, pdg2014::asMZ, 4);
// Crossing Mt threshold
std::cout << aS(1000) << std::endl;
```

Let us now turn to $\alpha(\mu)$. In our approach, it is defined through Eqs. (5) and (6). If we introduce G_F as an additional input parameter, we may express $\alpha(\mu)$ as [12]

$$\alpha(\mu) = \frac{\sqrt{2} G_F M_W^2}{\pi} [1 + \delta_W(\mu)] \left[1 - \frac{M_W^2}{M_Z^2} \frac{1 + \delta_W(\mu)}{1 + \delta_Z(\mu)} \right]. \tag{16}$$

Since $\delta_Z(\mu)$ and $\delta_W(\mu)$ depend on $\alpha(\mu)$, Eq. (16) only provides an implicit definition of $\alpha(\mu)$. The `mr` program library provides two different classes, `AlphaSolve` and `AlphaGF`, to obtain $\alpha(\mu)$ (see Appendix A.3). Class `AlphaSolve` uses a numerical solution of Eq. (16), while class `AlphaGF` uses a perturbative reexpansion thereof.

Class `P2MS` allows us to evaluate all the $\overline{\rm MS}$ couplings and masses at a chosen matching scale μ_0. To use it, object `OSinput` has to be constructed first, by one of the methods described in Section 3.1. One also has to create the $\alpha_s(\mu)$ object `AlphaS` as described above.

Here is an example of defining two sets of $\overline{\rm MS}$ parameters, one with $\mu_0 = M_Z$ and the other one with $\mu_0 = M_t$, using the same set of OS input parameters:

```
// Input: Pole masses and Fermi constant in OS scheme
OSinput oi(pdg2014::Mb, pdg2014::MW, pdg2014::MZ, pdg2014::MH,
    pdg2014::Mt);
// $\overline{\mathrm{MS}}$ QCD coupling for as(Mt) from as(MZ)
AlphaS as(oi);
// Set of all $\overline{\mathrm{MS}}$ parameters at scale Mt
P2MS pMSmt(oi,pdg2014::Gf, as(oi.Mt()), oi.Mt(), order::all);
// Set of all $\overline{\mathrm{MS}}$ parameters at scale MZ
P2MS pMSmZ(oi,pdg2014::Gf, as(oi.MZ()), oi.MZ(), order::all);
```

The arguments in the constructor of `P2MS` include the `OSinput` object, the value of G_F, the value of $\alpha_s(\mu_0)$, and the scale μ_0. The last argument, `order`, is a mask indicating the orders to be included in Eqs. (8) and (10) (see classes `AlphaSolve` and `AlphaGF`).

3.4. RG evolution

For the evolution of the $\overline{\rm MS}$ couplings from the initial scale μ_0 up to the final scale μ, we implemented in our code the full three-loop results for the SM β functions. The pure QCD corrections to $\alpha_s(\mu)$ at four loops [19, 23, 34] and to $y_t(\mu)$ and $y_b(\mu)$ at four [20, 21, 22] and five [35] loops are also included. Using these β functions, we may construct the system of coupled differential equations, which can be solved numerically. For its numerical solution, we use the Cash–Karp modification of the Runge–Kutta method with adaptive grid step as implemented in the external routine `OdeInt` [36].

Instead of $g'(\mu)$, the rescaled constant

$$g_1(\mu) = \sqrt{\frac{3}{5}} g'(\mu), \tag{17}$$

which corresponds to the normalization familiar from grand unified theories (GUTs), is sometimes used in the literature. Following this tradition, we also

use $g_1(\mu)$ rather than $g'(\mu)$ in our code. For the purpose of the evolution, it is convenient to introduce the following $\overline{\rm MS}$ coupling constants related to the $\overline{\rm MS}$ parameters in Eq. (1):

$$\begin{aligned} a_1(\mu) &= \frac{5}{3}\frac{g'^2(\mu)}{16\pi^2}, \qquad a_2(\mu) = \frac{g^2(\mu)}{16\pi^2}, \qquad a_s(\mu) = \frac{g_s^2(\mu)}{16\pi^2}, \\ a_t(\mu) &= \frac{y_t^2(\mu)}{16\pi^2}, \qquad a_b(\mu) = \frac{y_b^2(\mu)}{16\pi^2}, \qquad a_\tau(\mu) = \frac{y_\tau^2(\mu)}{16\pi^2}, \\ a_\lambda(\mu) &= \frac{\lambda(\mu)}{16\pi^2}, \end{aligned} \tag{18}$$

where we have chosen the GUT normalization for $a_1(\mu)$ and introduced the Yukawa coupling $y_\tau(\mu)$ of the τ lepton. Then, Eq. (14) becomes the system of the seven coupled differential equations describing the RG evolution of the $\overline{\rm MS}$ coupling constants in Eq. (18),

$$\begin{aligned} \partial_t a_1 &= \beta_{a_1}(\mathbf{a}), \qquad \partial_t a_2 = \beta_{a_2}(\mathbf{a}), \qquad \partial_t a_s = \beta_{a_s}(\mathbf{a}), \\ \partial_t a_t &= \beta_{a_t}(\mathbf{a}), \qquad \partial_t a_b = \beta_{a_b}(\mathbf{a}), \qquad \partial_t a_\tau = \beta_{a_\tau}(\mathbf{a}), \\ \partial_t a_\lambda &= \beta_{a_\lambda}(\mathbf{a}), \end{aligned} \tag{19}$$

where $\mathbf{a}$ is a seven-component vector accommodating all the relevant $\overline{\rm MS}$ coupling constants in the SM, $\mathbf{a} = \{a_1, a_2, a_s, a_t, a_b, a_\tau, a_\lambda\}$, and ∂_t means differentiation with respect to $t = \ln\mu^2$. The evolution of $m_\phi(\mu)$ and $v(\mu)$ is controlled by the two additional differential equations,

$$\partial_t m_\phi = \beta_{m_\phi}(\mathbf{a}), \qquad \partial_t v = \beta_v(\mathbf{a}), \tag{20}$$

with

$$\beta_{m_\phi} = \frac{dm_\phi}{d\ln\mu^2} = \frac{m_\phi}{2}\gamma_{m_\phi^2}, \qquad \beta_v = \frac{dv}{d\ln\mu^2} = v\gamma_v, \tag{21}$$

where we have adopted the definitions of $\gamma_{m_\phi^2}$ and γ_v from Refs. [30, 31], respectively. The RG evolution of the $\overline{\rm MS}$ coupling constants according to Eq. (19) is implemented in class `CouplingsSM` and the extension by the RG evolution of the two additional $\overline{\rm MS}$ mass parameters according to Eq. (20) in class `ParametersSM`. Both of them are template classes. Using template parameters as `<a1,a2,as,at,ab,atau,alam,mphi,vev>`, it is possible to specify the orders of the β functions used for the solution of the differential equations.

The following example uses three-loop β functions for the EW gauge couplings and parameters, and four-loop β functions for the strong and Yukawa couplings, and the negative value `-1` at the sixth template parameter position means that the coupling $y_\tau(\mu)$ is completely eliminated from the SM:

```
ParametersSM<3,3,4,4,4,-1,3,3,3>
          p(a1, a2, as, at, ab, atau, lam, mphi, vev, NG)
```

If we are just interested in the solution of the coupled system of differential equations describing the evolution of the $\overline{\text{MS}}$ coupling constants in Eq. (19), we may use the short form:

```
CouplingsSM<3,3,4,4,4,-1,3>
          p(a1, a2, as, at, ab, atau, lam, NG)
```

If the object `P2MS` has already been constructed, as described in Section 3.3 and Appendix A.3, it is possible to create a solver using:

```
CouplingsSM<3,3,4,4,4,-1,3> p(p2ms, NG)
```

Here, `p2ms` is the previously constructed object of type `P2MS`. In all these examples, `mu0` is the initial scale (in GeV) for the RG evolution, and `NG` is the number of fermion generations in the SM, with default value `NG` $= 3$.

The actual RG evolution proceeds when `operator()` is called for one of the above-mentioned classes. Correspondingly, a seven-component vector containing the values of the $\overline{\text{MS}}$ coupling constants or a nine-component vector containing also the two $\overline{\text{MS}}$ mass parameters is returned. If, for example, we are interested in $m_\phi(\mu)$ at the scale $\mu = 1000$ GeV, the following sequence of commands has to be entered:

```
SMCouplings ai = p(1000);
// using enum couplings for more readable indexing
std::cout << ai[couplings::mphi] << std::endl;
// using numerical index, starting from 0
std::cout << ai[7] << std::endl;
```

We refer to Appendix C for further details.

4. `Mathematica` interface

It is possible to compile a `Mathematica` interface to the `mr` program library. If the `Mathematica` installation path is unusual, the path to the development utils `mprep` and `mcc` has to be specified at the configuration step, e.g., as:

```
./configure --with-mcc-path=<MATHDIR>/<ARCH>/CompilerAdditions
```

Here, `<MATHDIR>/<ARCH>/CompilerAdditions` is a path to the directory containing the `mprep` and `mcc` utils.

When the `Mathematica` interface is successfully compiled, it is possible to load it into `Mathematica`. If the `mr` mathlink executable is placed in the same directory and, for example, we wish to calculate $m_W^2(\mu)/M_W^2$ at scale $\mu = M_t$, then a typical session is:

```
Install["mr"];
(*      Mb   MW   MZ   MH   MT   mu *)
mmWMMW[4.4, 80, 91, 125, 173, 173]
```

The output, where the user has to supply the values of the $\overline{\rm MS}$ coupling constants at scale μ, `aEW[mu]` and `aQCD[mu]`, then reads:

```
{
  1 + 185.3545315192623037 aEW[173]
    +  563.188663413438646 aEW[173] aQCD[173]
    - 19207.76644304067414 aEW[173]^2
}
```

Alternatively, if we are just interested in the set of coefficients, the command:

```
Install["mr"];
(* Mb   MW   MZ   MH   MT   mu *)
Xt[4.4, 80, 91, 125, 173, 173]
```

returns a substitution list for the top-quark coefficients `yT` $= Y_t^{i,j}(\mu)$ in Eq. (8) and `xt` $= X_t^{i,j}(\mu)$ in Eq. (10):

```
{
  xt[1, 0] -> 108.36879199119430404,
  yT[1, 0] -> 2.1032409021919717513,
  xt[1, 1] -> -412.09726673991774271,
```

```
  yT[1, 1] -> -79.271276023543769118,
  xt[2, 0] -> -13725.00152946124629,
  yT[2, 0] -> 685.749556148969868
}
```

These functions, supplied with the `Mathematica` expressions for the SM β functions, which are available, for example, as ancillary files to the arXiv versions of Refs. [27, 28, 30], are sufficient for the analysis.

5. Example programs

We supply example programs with the code of the `mr` program library. For the compilation of the user's program using the `mr` program library, one has to type:

```
g++ -o userprog 'pkg-config --cflags --libs mr' userprog.cpp
```

A complete example is presented in Appendix D.

6. Conclusion

We presented the `C++` program library `mr` with a `Mathematica` interface for the evaluation of the $\overline{\text{MS}}$ couplings of the SM including the full set of two-loop threshold corrections and their three-loop evolution with high numerical precision. The source code including examples is available from the URL `http://apik.github.io/mr/`. A development version of this program library has already been used for the analysis of the EW vacuum stability in the SM [6].

7. Acknowledgments

We thank A. V. Bednyakov for useful discussions and for testing our program library. This work was supported in part by the German Federal Ministry for Education and Research BMBF through Grant No. 05H15GUCC1, by the German Research Foundation DFG through the Collaborative Research Centre No. SFB 676 *Particles, Strings and the Early Universe: the Structure of Matter and Space-Time*, by the Heisenberg–Landau Programme, and by the Dynasty Foundation.

Appendix A. Methods needed for OS mass input

Appendix A.1. Class `OSinput`

This class includes the following methods:

- `OSinput(long double Mb, MW, MZ, MH, Mt)`
  as the class constructor,
- `long double MMb(), MMW(), MMZ(), MMH(), MMt()`
  to get the squared masses M_b^2, M_W^2, M_Z^2, M_H^2, and M_t^2,
- `long double Mb(), MW(), MZ(), MH(), Mt()`
  to get the masses M_b, M_W, M_Z, M_H, and M_t,
- `OSinput setMb(long double), setMW(), setMZ(), setMH(), setMt()`
  to reset the values of the masses M_b, M_W, M_Z, M_H, and M_t in the OS input already initialized,
- `long double CW(), SW(), CCW(), SSW()`
  to get the trigonometric functions $\cos\theta_w$, $\sin\theta_w$, $\cos^2\theta_w$, and $\sin^2\theta_w$ of the weak mixing angle θ_w.

Appendix A.2. Classes `bb<OS>`, `WW<OS>`, `ZZ<OS>`, `HH<OS>`, *and* `tt<OS>`

The results for the coefficients $Y_x^{i,j}(\mu)$ and $X_x^{i,j}(\mu)$ in Eqs. (8) and (10), respectively, are organized as follows:

$$X^{i,j} = \texttt{nL} \cdot X_L + \texttt{nH} \cdot X_H + \texttt{boson} \cdot X_B, \tag{A.1}$$

and similarly for $Y_x^{i,j}(\mu)$, where `nL` and `nH` are the numbers of massless and massive fermion generations, respectively, and `boson` is a tag for the purely bosonic contributions. This splitting allows for the extraction of the individual contributions from the full results. The default values, corresponding to the case of the SM, read `nL` $= 2$, `nH` $= 1$, and `boson` $= 1$. The pure QCD corrections to $m_t(\mu)$ and $y_t(\mu)$ are evaluated using Eq. (59) in Ref. [12] with $n_l = 2\,\texttt{NL} + 1$ and $n_h = \texttt{NH}$ and those to $m_b(\mu)$ and $y_b(\mu)$ using Eq. (11) with $n_l = 2\,\texttt{NL}$ and $n_h = n_m = \texttt{NH}$.

The following methods are available:

- `long double x01(size_t nL = 2, size_t nH = 1, size_t boson=1)`
  `long double x02(size_t nL = 2, size_t nH = 1, size_t boson=1)`
  `long double x03(size_t nL = 2, size_t nH = 1, size_t boson=1)`
  `long double x04(size_t nL = 2, size_t nH = 1, size_t boson=1)`
  for the pure QCD corrections to $m_f(\mu)$ with $f = t, b$ (classes `tt<OS>` and `bb<OS>`),

- `long double y01(size_t nL = 2, size_t nH = 1, size_t boson=1)`
  `long double y02(size_t nL = 2, size_t nH = 1, size_t boson=1)`
  `long double y03(size_t nL = 2, size_t nH = 1, size_t boson=1)`
  `long double y04(size_t nL = 2, size_t nH = 1, size_t boson=1)`
  for the pure QCD corrections to $y_f(\mu)$ with $f = t, b$ (classes `tt<OS>` and `bb<OS>`),

- `long double x10(size_t nL = 2, size_t nH = 1, size_t boson=1)`
  `long double x11(size_t nL = 2, size_t nH = 1, size_t boson=1)`
  `long double x20(size_t nL = 2, size_t nH = 1, size_t boson=1)`
  for the corrections of orders $\mathcal{O}(\alpha)$, $\mathcal{O}(\alpha\alpha_s)$, and $\mathcal{O}(\alpha^2)$, respectively, to $m_B(\mu)$ with $B = W, Z, H$ and $m_f(\mu)$ with $f = t, b$,

- `long double y10(size_t nL = 2, size_t nH = 1, size_t boson=1)`
  `long double y11(size_t nL = 2, size_t nH = 1, size_t boson=1)`
  `long double y20(size_t nL = 2, size_t nH = 1, size_t boson=1)`
  for the corrections of orders $\mathcal{O}(\alpha)$, $\mathcal{O}(\alpha\alpha_s)$, and $\mathcal{O}(\alpha^2)$, respectively, to $\delta_f(\mu)$ with $x = W, Z, H, t, b$,

- `long double xgl10(size_t nL = 2, size_t nH = 1, size_t boson=1)`
  `long double xgl11(size_t nL = 2, size_t nH = 1, size_t boson=1)`
  `long double xgl20(size_t nL = 2, size_t nH = 1, size_t boson=1)`
  for the corrections of orders $\mathcal{O}(\alpha)$, $\mathcal{O}(\alpha\alpha_s)$, and $\mathcal{O}(\alpha^2)$, respectively, in the gaugeless limit to $m_B(\mu)$ with $B = W, Z, H$ and $m_f(\mu)$ with $f = t, b$, and

- `long double ygl10(size_t nL = 2, size_t nH = 1, size_t boson=1)`
  `long double ygl11(size_t nL = 2, size_t nH = 1, size_t boson=1)`
  `long double ygl20(size_t nL = 2, size_t nH = 1, size_t boson=1)`
  for the corrections of orders $\mathcal{O}(\alpha)$, $\mathcal{O}(\alpha\alpha_s)$, and $\mathcal{O}(\alpha^2)$, respectively, in the gaugeless limit to $\delta_x(\mu)$ with $x = W, Z, H, t, b$.

Instead of using `xij(...)`, one can also use the notation `x(i,j,...)`. For that, the following additional methods are available:

- `long double x(size_t apow, size_t aspow, size_t nL = 2, size_t nH = 1, size_t boson = 1)`
- `long double y(size_t apow, size_t aspow, size_t nL = 2, size_t nH = 1, size_t boson = 1)`
- `long double xgl(size_t apow, size_t aspow, size_t nL = 2, size_t nH = 1, size_t boson = 1)`
- `long double ygl(size_t apow, size_t aspow, size_t nL = 2, size_t nH = 1, size_t boson = 1)`

Appendix A.3. Class ***P2MS***

This is the main class to obtain the $\overline{\text{MS}}$ parameters in terms of the OS input. Let us first consider $\alpha(\mu)$, which is defined through G_F. Using fixed values of G_F and $\alpha_s(M_Z)$ together with `OSinput`, we may find $\alpha(\mu)$ as the numerical solution of Eq. (16):

```
AlphaSolve(const OSinput & in_, double tol_ = 10e-9,
           const long double & Gf0_ = pdg2014::Gf,
           const long double & as_ = pdg2014::asMZ,
           unsigned order_ =
           order::x01|order::x10|order::x02|
           order::x11|order::x20|order::x03)
```

Alternatively, we may solve Eq. (16) perturbatively, which gives another definition of $\alpha(\mu)$ at fixed order:

```
AlphaGF(const OSinput & in_, double tol_ = 10e-9,
        const long double & Gf0_ = pdg2014::Gf,
        const long double & as_ = pdg2014::asMZ,
        unsigned order_ =
        order::x01|order::x10|order::x02|
        order::x11|order::x20|order::x03)
```

In the above methods, the argument `order` is a bit mask with predefined values to enable or disable the corrections of given orders. The tolerance parameter `tol_` controls the accuracy of numerical solution.

Now we turn to the P2MS object, the declaration of which is:

```
P2MS<AlphaT>::P2MS(const OSinput & oi_, const long double & Gf_,
                   const long double & as_,
                   const long double & mu_,
                   unsigned ord_)
```

Here, AlphaT is one of possible types of solution, AlphaSolve or AlphaGF, at scale μ.

In the following, we use the following notations for the $\overline{\rm MS}$ couplings: $g_1(\mu)$ as defined in Eq. (17), $g_2(\mu) \equiv g(\mu)$, and similarly for the squares of the couplings, $a_1(\mu)$ and $a_2(\mu)$. The available methods of the P2SM class include the following:

- long double a1(), a2(), as(), at(), ab(), alam()
  to get $a_1(\mu)$, $a_2(\mu)$, $a_s(\mu)$, $a_t(\mu)$, $a_b(\mu)$, and $a_\lambda(\mu)$,
- long double g1(), g2(), gs(), yt(), yb(), lam()
  to get $g_1(\mu)$, $g(\mu)$, $g_s(\mu)$, $y_t(\mu)$, $y_b(\mu)$, and $\lambda(\mu)$,
- long double mphi(), vev()
  to get $m_\phi(\mu)$ and $v(\mu)$,
- MSinput getMSpar()
  to construct MSinput at scale μ as explained in Appendix B.1,
- SMCouplings runningCouplings()
  to get the vector {g1,g2,gs,yt,yb,ytau,lam,mphi,vev}, with the correction to $y_\tau(\mu)$ being always zero, and
- SMCouplings ai()
  to get the vector {a1,a2,as,at,ab,atau,alam,mphi,vev}, with the correction to $a_\tau(\mu)$ being always zero.

Appendix B. Methods needed for $\overline{\rm MS}$ parameter input

Appendix B.1. Class ***MSinput***

Here, we explain the methods of the classes needed if the initial input is given in terms of $\overline{\rm MS}$ masses or couplings. To highlight the difference in the construction between masses and couplings, we do not use constructors, but special functions which return carefully constructed objects:

- ```
  MSinput fromMasses(long double mb, long double mW,
                     long double mZ, long double mH,
                     long double mt)
  ```

  for the construction from the set of $\overline{\rm MS}$ masses,

- ```
  MSinput fromCouplings(long double g1, long double g2,
                        long double yb, long double yt,
                        long double lam, long double mphi,
                        long double scale)
  ```

  for the construction from the set of $\overline{\rm MS}$ couplings and the fixing of μ_0.

Appendix B.2. Classes `bb<MS>`, `WW<MS>`, `ZZ<MS>`, `HH<MS>`, *and* `tt<MS>`

Here, we explain the methods for the calculation of the QCD and EW corrections defined in Eq. (9):

- ```
  long double x01(size_t nL = 2, size_t nH = 1, size_t boson=1)
  long double x02(size_t nL = 2, size_t nH = 1, size_t boson=1)
  long double x03(size_t nL = 2, size_t nH = 1, size_t boson=1)
  long double x04(size_t nL = 2, size_t nH = 1, size_t boson=1)
  ```
  for the pure QCD corrections to $m_f(\mu)$ with $f = t, b$ (classes `tt<MS>` and `bb<MS>`),

- ```
  long double x10(size_t nL = 2, size_t nH = 1, size_t boson=1)
  long double x11(size_t nL = 2, size_t nH = 1, size_t boson=1)
  long double x20(size_t nL = 2, size_t nH = 1, size_t boson=1)
  ```
  for the corrections of orders $\mathcal{O}(\alpha)$, $\mathcal{O}(\alpha\alpha_s)$, and $\mathcal{O}(\alpha^2)$, respectively, to $m_B(\mu)$ with $B = W, Z, H$ and $m_f(\mu)$ with $f = t, b$.

Appendix C. Methods needed for RG evolution

Each β function is expressed in the form

$$\beta_i^{(l)} = \sum_{j_1,\ldots,j_7=0}^{j_1+\cdots+j_7\leq l+1} a_1^{j_1}\cdots a_7^{j_1} C(j_1,\ldots,j_7), \tag{C.1}$$

where l is a cutoff with respect to the loop order. It is possible to fix l for each β function separately using template parameters:

```
template < int pocoa1, int pocoa2, int pocoas,
           int pocoat, int pocoab, int pocoatau,
           int pocolam >
```

To eliminate a coupling constant from the theory, one may assign a negative value to the respective parameter. For example, the pure QCD β function at one loop is referred to as:

```
CouplingsSM<-1,-1,1,-1,-1,-1,-1>
```

The constructor of the appropriate object from the $\overline{\text{MS}}$ coupling constants at the initial scale μ_0 for `NG` fermion generations reads:

```
CouplingsSM(double a1, double a2, double as,
            double at, double ab, double atau,
            double lam,
            double mu0_, size_t NG_ = 3)
```

Here, we have used the input values of the $\overline{\text{MS}}$ coupling constants defined in Eq. (18). The analogous constructor that also includes $m_\phi(\mu)$ and $v(\mu)$ reads:

```
ParametersSM(double a1, double a2, double as,
             double at, double ab, double atau,
             double lam, double mphi, double vev,
             double mu0_, size_t NG_ = 3)
```

The constructor of the appropriate object from the `P2MS` object (see Appendix A.3) reads:

```
ParametersSM(const P2MS<T>& pi, size_t NG_ = 3)
```

The operator `()` is used for the evolution up to the final scale μ:

```
SMCouplings operator()(long double mu)
```

This method is implemented for both classes, `CouplingsSM` and `ParametersSM`, and returns the lists $\{a_1(\mu), a_2(\mu), a_s(\mu), a_t(\mu), a_b(\mu), a_\tau(\mu), a_\lambda(\mu)\}$ and $\{a_1(\mu), a_2(\mu), a_s(\mu), a_t(\mu), a_b(\mu), a_\tau(\mu), a_\lambda(\mu), m_\phi(\mu), v(\mu)\}$, respectively, at the final scale μ. The operator:

```
SMCouplings AandB(long double mu)
```

returns two lists of length nine with the values of the seven $\overline{\rm MS}$ coupling constants and the two additional $\overline{\rm MS}$ mass parameters together with their β functions at the final scale μ. These β functions are defined in Eqs. (19) and (20).

Appendix D. Complete example

Here, we present an example program, which calculates the $\overline{\rm MS}$ coupling constants at the initial scale $\mu_0 = M_t$ for given input values of the pole masses and evolves them up to the final scale μ. We use three-loop RG evolution for the gauge couplings, the top Yukawa coupling, and the Higgs self-coupling, and set the tau Yukawa coupling to zero everywhere.

```
// Example of Pole masses and Gf conversion to
// set of running couplings, running Higgs mass
// term and running vev in MS scheme

#include "mr.hpp"

int main (int argc, char *argv[])
{
  try
    {
      loglevel = logINFO;

      // Input: Pole masses and Fermi constant in OS scheme
      OSinput oi(pdg2014::Mb, pdg2014::MW, pdg2014::MZ,
          pdg2014::MH, pdg2014::Mt);

      // Running QCD coupling for as(Mt) from as(MZ)
      AlphaS as(oi);

      // Set of all running parameters at scale Mt
      P2MS<AlphaSolve> pMSmt(oi,pdg2014::Gf, as(oi.Mt()), oi.Mt(),
          order::all);

      // Initial values for running, input from pole masses
```

```
      ParametersSM<3,3,3,3,3,-1,3,3,0> avP2MS(pMSmt);

      std::cout << std::setprecision(3);

      for (size_t muPow = 3; muPow <= 20; muPow++)
        {

          SMCouplings av = avP2MS(pow(10,2*muPow));

          std::cout << " log10(mu) = " << muPow
                    << " a1  = " << av[couplings::g1]
                    << " a2  = " << av[couplings::g2]
                    << " a3  = " << av[couplings::gs]
                    << " at  = " << av[couplings::yt]
                    << " ab  = " << av[couplings::yb]
                    << " atau = " << av[couplings::ytau]
                    << " alam = " << av[couplings::lam]
                    << " mphi = " << av[couplings::mphi]
                    << " vev = " << av[couplings::vev] << std::endl;
        }
    }
  catch (std::exception &p)
    {
      std::cerr << p.what() << std::endl;
      return 1;
    }

  return 0;
}
```

References

[1] G. Aad, et al., Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC, Phys. Lett. B716 (2012) 1–29. arXiv:1207.7214, doi:10.1016/j.physletb.2012.08.020.

[2] S. Chatrchyan, et al., Observation of a new boson at a mass of 125 GeV

with the CMS experiment at the LHC, Phys. Lett. B716 (2012) 30–61. arXiv:1207.7235, doi:10.1016/j.physletb.2012.08.021.

[3] F. Bezrukov, M. Yu. Kalmykov, B. A. Kniehl, M. Shaposhnikov, Higgs boson mass and new physics, JHEP 10 (2012) 140. arXiv:1205.2893, doi:10.1007/JHEP10(2012)140.

[4] G. Degrassi, S. Di Vita, J. Elias-Miró, J. R. Espinosa, G. F. Giudice, G. Isidori, A. Strumia, Higgs mass and vacuum stability in the Standard Model at NNLO, JHEP 08 (2012) 098. arXiv:1205.6497, doi:10.1007/JHEP08(2012)098.

[5] D. Buttazzo, G. Degrassi, P. P. Giardino, G. F. Giudice, F. Sala, A. Salvio, A. Strumia, Investigating the near-criticality of the Higgs boson, JHEP 12 (2013) 089. arXiv:1307.3536, doi:10.1007/JHEP12(2013)089.

[6] A. V. Bednyakov, B. A. Kniehl, A. F. Pikelner, O. L. Veretin, Stability of the electroweak vacuum: Gauge independence and advanced precision, Phys. Rev. Lett. 115 (20) (2015) 201802. arXiv:1507.08833, doi:10.1103/PhysRevLett.115.201802.

[7] J. Fleischer, F. Jegerlehner, Radiative corrections to Higgs-boson decays in the Weinberg-Salam model, Phys. Rev. D23 (1981) 2001–2026. doi:10.1103/PhysRevD.23.2001.

[8] R. Hempfling, B. A. Kniehl, Relation between the fermion pole mass and $\overline{\text{MS}}$ Yukawa coupling in the standard model, Phys. Rev. D51 (1995) 1386–1394. arXiv:hep-ph/9408313, doi:10.1103/PhysRevD.51.1386.

[9] B. A. Kniehl, J. H. Piclum, M. Steinhauser, Relation between bottom-quark $\overline{\text{MS}}$ Yukawa coupling and pole mass, Nucl. Phys. B695 (2004) 199–216. arXiv:hep-ph/0406254, doi:10.1016/j.nuclphysb.2004.06.036.

[10] F. Jegerlehner, M. Yu. Kalmykov, B. A. Kniehl, On the difference between the pole and the $\overline{\text{MS}}$ masses of the top quark at the electroweak scale, Phys. Lett. B722 (2013) 123–129. arXiv:1212.4319, doi:10.1016/j.physletb.2013.04.012.

[11] B. A. Kniehl, O. L. Veretin, Two-loop electroweak threshold corrections to the bottom and top Yukawa couplings, Nucl. Phys. B885 (2014) 459–480, [Erratum: Nucl. Phys. B894 (2015) 56-57]. arXiv:1401.1844, doi:10.1016/j.nuclphysb.2015.02.012,10.1016/j.nuclphysb.2014.05.029.

[12] B. A. Kniehl, A. F. Pikelner, O. L. Veretin, Two-loop electroweak threshold corrections in the Standard Model, Nucl. Phys. B896 (2015) 19–51. arXiv:1503.02138, doi:10.1016/j.nuclphysb.2015.04.010.

[13] N. Gray, D. J. Broadhurst, W. Grafe, K. Schilcher, Three-loop relation of quark $\overline{\rm MS}$ and pole masses, Z. Phys. C48 (1990) 673–680. doi:10.1007/BF01614703.

[14] S. Bekavac, A. Grozin, D. Seidel, M. Steinhauser, Light quark mass effects in the on-shell renormalization constants, JHEP 10 (2007) 006. arXiv:0708.1729, doi:10.1088/1126-6708/2007/10/006.

[15] E. Remiddi, J. A. M. Vermaseren, Harmonic polylogarithms, Int. J. Mod. Phys. A15 (2000) 725–754. arXiv:hep-ph/9905237, doi:10.1142/S0217751X00000367.

[16] F. Jegerlehner, M. Yu. Kalmykov, O. Veretin, $\overline{\rm MS}$ vs. pole masses of gauge bosons: electroweak bosonic two-loop corrections, Nucl. Phys. B641 (2002) 285–326. arXiv:hep-ph/0105304, doi:10.1016/S0550-3213(02)00613-2.

[17] F. Jegerlehner, M. Yu. Kalmykov, O. Veretin, $\overline{\rm MS}$ vs. pole masses of gauge bosons II: two-loop electroweak fermion corrections, Nucl. Phys. B658 (2003) 49–112. arXiv:hep-ph/0212319, doi:10.1016/S0550-3213(03)00177-9.

[18] S. P. Martin, D. G. Robertson, Higgs boson mass in the Standard Model at two-loop order and beyond, Phys. Rev. D90 (7) (2014) 073010. arXiv:1407.4336, doi:10.1103/PhysRevD.90.073010.

[19] T. van Ritbergen, J. A. M. Vermaseren, S. A. Larin, The four-loop β-function in quantum chromodynamics, Phys. Lett. B400 (1997) 379–384. arXiv:hep-ph/9701390, doi:10.1016/S0370-2693(97)00370-5.

[20] K. G. Chetyrkin, Quark mass anomalous dimension to $O(\alpha_s^4)$, Phys. Lett. B404 (1997) 161–165. arXiv:hep-ph/9703278, doi:10.1016/S0370-2693(97)00535-2.

[21] J. A. M. Vermaseren, S. A. Larin, T. van Ritbergen, The 4-loop quark mass anomalous dimension and the invariant quark mass, Phys. Lett. B405 (1997) 327–333. arXiv:hep-ph/9703284, doi:10.1016/S0370-2693(97)00660-6.

[22] K. G. Chetyrkin, Four-loop renormalization of QCD: full set of renormalization constants and anomalous dimensions, Nucl. Phys. B710 (2005) 499–510. arXiv:hep-ph/0405193, doi:10.1016/j.nuclphysb.2005.01.011.

[23] M. Czakon, The four-loop QCD β-function and anomalous dimensions, Nucl. Phys. B710 (2005) 485–498. arXiv:hep-ph/0411261, doi:10.1016/j.nuclphysb.2005.01.012.

[24] L. N. Mihaila, J. Salomon, M. Steinhauser, Gauge coupling beta functions in the standard model to three loops, Phys. Rev. Lett. 108 (2012) 151602. arXiv:1201.5868, doi:10.1103/PhysRevLett.108.151602.

[25] K. G. Chetyrkin, M. F. Zoller, Three-loop β-functions for top-Yukawa and the Higgs self-interaction in the Standard Model, JHEP 06 (2012) 033. arXiv:1205.2892, doi:10.1007/JHEP06(2012)033.

[26] L. N. Mihaila, J. Salomon, M. Steinhauser, Renormalization constants and beta functions for the gauge couplings of the standard model to three-loop order, Phys. Rev. D86 (2012) 096008. arXiv:1208.3357, doi:10.1103/PhysRevD.86.096008.

[27] A. V. Bednyakov, A. F. Pikelner, V. N. Velizhanin, Anomalous dimensions of gauge fields and gauge coupling beta-functions in the Standard Model at three loops, JHEP 01 (2013) 017. arXiv:1210.6873, doi:10.1007/JHEP01(2013)017.

[28] A. V. Bednyakov, A. F. Pikelner, V. N. Velizhanin, Yukawa coupling beta-functions in the Standard Model at three loops, Phys. Lett. B722 (2013) 336–340. arXiv:1212.6829, doi:10.1016/j.physletb.2013.04.038.

[29] K. G. Chetyrkin, M. F. Zoller, β-function for the Higgs self-interaction in the Standard Model at three-loop level, JHEP 04 (2013) 091, [Erratum: JHEP09 (2013) 155]. arXiv:1303.2890, doi:10.1007/JHEP04(2013)091,10.1007/JHEP09(2013)155.

[30] A. V. Bednyakov, A. F. Pikelner, V. N. Velizhanin, Higgs self-coupling beta-function in the Standard Model at three loops, Nucl. Phys. B875 (2013) 552–565. arXiv:1303.4364, doi:10.1016/j.nuclphysb.2013.07.015.

[31] A. V. Bednyakov, A. F. Pikelner, V. N. Velizhanin, Three-loop Higgs self-coupling beta-function in the Standard Model with complex Yukawa matrices, Nucl. Phys. B879 (2014) 256–267. arXiv:1310.3806, doi:10.1016/j.nuclphysb.2013.12.012.

[32] K. G. Chetyrkin, B. A. Kniehl, M. Steinhauser, Strong Coupling Constant with Flavor Thresholds at Four Loops in the Modified Minimal-Subtraction Scheme, Phys. Rev. Lett. 79 (1997) 2184–2187. arXiv:hep-ph/9706430, doi:10.1103/PhysRevLett.79.2184.

[33] K. G. Chetyrkin, B. A. Kniehl, M. Steinhauser, Decoupling relations to $O(\alpha_s^3)$ and their connection to low-energy theorems, Nucl. Phys. B510 (1998) 61–87. arXiv:hep-ph/9708255, doi:10.1016/S0550-3213(97)00649-4.

[34] A. V. Bednyakov, A. F. Pikelner, Four-loop strong coupling beta-function in the Standard Model, arXiv:1508.02680.

[35] P. A. Baikov, K. G. Chetyrkin, J. H. Kühn, Quark mass and field anomalous dimensions to $\mathcal{O}(\alpha_s^5)$, JHEP 10 (2014) 076. arXiv:1402.6611, doi:10.1007/JHEP10(2014)076.

[36] K. Ahnert, M. Mulansky, Odeint – Solving Ordinary Differential Equations in C++, AIP Conf. Proc. 1389 (2011) 1586–1589. arXiv:1110.3397, doi:10.1063/1.3637934.