B meson mixing at NNLO: technical aspects

Pascal Reeck[a], Vladyslav Shtabovenko[b], Matthias Steinhauser[a],

(a) Institut für Theoretische Teilchenphysik, Karlsruhe Institute of Technology (KIT),

Wolfgang-Gaede Strasse 1, 76131 Karlsruhe, Germany

(b) Theoretische Physik 1, Center for Particle Physics Siegen,

Universität Siegen, Walter-Flex-Str. 3, 57068 Siegen, Germany

Abstract

We provide details to several technical aspects which are important for the calculation of next-to-next-to-leading order corrections to the mixing of neutral B mesons. This includes the computation of the master integrals for finite charm and bottom quark masses, traces over products of up to 22 γ matrices and tensor integrals with up to rank 11.

1 Introduction and notation

In the Standard Model of particle physics, the mixing of neutral B mesons is a loop-induced process which, at leading-order, is mediated by two box Feynman diagram where quark lines exchange two virtual W bosons. The time evolution is governed by a 2×2 matrix which contains the mass and decay matrices M^q and Γ^q with $q = s, d$ being the flavour of the spectator quark.

The quantities which are measured by experiment are the mass and width differences of the lighter and heavier eigenstates,

$$
\begin{aligned}
\Delta M_q &= M_H^q - M_L^q, \\
\Delta \Gamma_q &= \Gamma_L^q - \Gamma_H^q,
\end{aligned}
\tag{1}
$$

and the charge-parity (CP) asymmetry in flavour-specific decays, a_{fs}^q. They are related to the matrix elements of the decay matrix via (see, e.g., Ref. [1] for more details)

$$
\begin{aligned}
\Delta M_q &= 2|M_{12}^q| + \mathcal{O}(|\Gamma_{12}|^2/|M_{12}|^2), \\
\frac{\Delta \Gamma_q}{\Delta M_q} &= -\text{Re}\frac{\Gamma_{12}^q}{M_{12}^q}, \\
a_{\text{fs}}^q &= \text{Im}\frac{\Gamma_{12}^q}{M_{12}^q},
\end{aligned}
\tag{2}
$$

where M_{12}^q and Γ_{12}^q are obtained from the dispersive and absorptive parts of the $B_q \to \overline{B}_q$ transition amplitude, respectively. In this work we concentrate on the computation of Γ_{12}^q.

The theoretical framework for the computation of Γ_{12}^q is given by a tower of effective theories. Using the Standard Model as starting point, one first integrates out the top quark mass, Higgs boson and electroweak gauge bosons and obtains the so-called $|\Delta B| = 1$ effective theory (cf. e.g. Ref. [2] for more details). Within this theory one considers the correlator of two $\Delta B = 1$ operators which mediates the $B - \overline{B}$ transition.[1] Subsequently, one performs a heavy mass expansion which leads to (see again Ref. [1] for a detailed derivation)

$$
\begin{aligned}
\Gamma_{12}^q &= -(\lambda_c^q)^2 \Gamma_{12}^{cc} - 2\lambda_c^q \lambda_u^q \Gamma_{12}^{uc} - (\lambda_u^q)^2 \Gamma_{12}^{uu} \\
&= -(\lambda_t^q)^2 \left[\Gamma_{12}^{cc} + 2\frac{\lambda_u^q}{\lambda_t^q} \left(\Gamma_{12}^{cc} - \Gamma_{12}^{uc} \right) + \left(\frac{\lambda_u^q}{\lambda_t^q} \right)^2 \left(\Gamma_{12}^{uu} + \Gamma_{12}^{cc} - 2\Gamma_{12}^{uc} \right) \right],
\end{aligned}
\tag{3}
$$

where $\lambda_a^q = V_{aq}^* V_{ab}$ and Γ_{12}^{ab} follow from the absorptive part of a bi-local matrix element containing a time-ordered product of two $|\Delta B| = 1$ effective Hamiltonians. This yields

$$
\Gamma_{12}^{ab} = \frac{G_F^2 m_b^2}{24\pi M_{B_s}} \left[H^{ab}(z)\langle B_s|Q|\bar{B}_s\rangle + \widetilde{H}_S^{ab}(z)\langle B_q|\widetilde{Q}_S|\bar{B}_q\rangle \right] + \mathcal{O}(\Lambda_{\text{QCD}}/m_b),
\tag{4}
$$

---

[1] See Appendix B for a list of all relevant operators.

where terms suppressed by the inverse bottom quark mass are not shown. The precise definition of the operators Q and $\widetilde{Q}_S$ can be found in Appendix B. $\langle B_s|Q|\bar{B}_s\rangle$ and $\langle B_q|\widetilde{Q}_S|\bar{B}_q\rangle$ are non-perturbative matrix elements, which are computed using lattice gauge theory [3] or QCD sum rules [4, 5]. The perturbative matching coefficients $H^{ab}(z)$ and $\widetilde{H}_S^{ab}(z)$ are the main focus of this paper. They depend on the mass ratio

$$z \;\; = \;\; \frac{m_c^2}{m_b^2} \,. \tag{5}$$

NLO QCD corrections to the matching coefficients with current-current operators have been known for quite some time [1, 6–8]. The complete set of penguin operators has only been added recently [9, 10], where terms of order z^0 and z^1 have been computed in an expansion for $z \to 0$. Fermionic NNLO corrections have been considered in Refs. [11–13] and the complete NNLO contribution from current-current operators is known from Ref.[14], however, only in an expansion for small charm quark masses up to (including) $\mathcal{O}(z)$. The current experimental and theoretical uncertainties (see Ref. [14] for a recent compilation) are such, that a complete NNLO calculation involving all penguin contributions for $H^{ab}(z)$ and $\widetilde{H}_S^{ab}(z)$ is necessary. There exist several challenges that make this endeavour non-trivial:

1. Large number of contributing Feynman diagrams.

2. Traces involving a large number of γ matrices.

3. High-rank tensor integrals.

4. Integration-by-parts reduction to master integrals.

5. Computation of the master integrals.

6. Proper inclusion of evanescent operators.

In this paper we address the points 2, 3 and 5. The large number of diagrams is not a fundamental issue, since it is always possible to split the problem into smaller pieces during the generation of the Feynman amplitudes with `qgraf` [15]. For the integration-by-parts reduction we use `Kira` [16–18] which requires at most a few days for general gauge parameter. Thus, this is not a limiting factor of our calculation. The last point concerning the evanescent operators is a purely field-theoretical problem. It will be addressed in detail in Ref. [19].

The remainder of the paper is organized as follows: In the next Section we describe in detail the construction of efficient projectors and Section 3 deals with the complementary approach based on high-rank tensor integrals. In Section 4 we discuss our methods to compute the master integrals. A sample result where the methods of Sections 2, 3 and 4 have been applied is shown in Section 5. We conclude in Section 6. In Appendix A the complete list of basis elements relevant for our calculation is shown and Appendix B contains a complete list of relevant operators of the $\Delta B = 1$ and $\Delta B = 2$ theory.
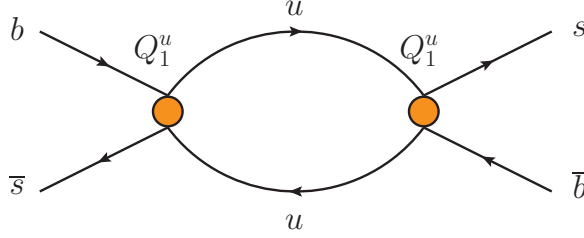
Figure 1: A simple one-loop example for the operator combination $Q_1^u \times Q_1^u$.

2 Projector methodology

2.1 Problem statement

We wish to apply a set of projectors to our amplitude in order to decompose the tensor integrals into products of scalar integrals, colour structures and spinor structures which are free of loop momenta. The colour part of our diagrams can be treated separately and is straightforward to handle with projectors as there are only two tensor structures corresponding to the two possible colour contractions of the external quarks. The spinor structures encountered in diagrams before applying projectors consist of two matrices in Dirac spinor space connecting a pair of external quarks each. While these objects have a simple structure in spinor space, they possess a more complicated tensor structure in Lorentz space since they are composed of a product of Dirac γ matrices, which we will call *spin lines* or *Dirac chains*. The Lorentz indices are contracted either across the two spin lines or with a propagator momentum. For example, the term

$$P_R \, \gamma^\mu \, \slashed{p}_1 \, \gamma^\nu \otimes P_R \, \gamma_\nu \, \slashed{p}_2 \, \gamma_\mu \tag{6}$$

corresponds to the spinor part of the diagram shown in Fig. 1. Here, p_1 and p_2 are linear combinations of loop and external momenta. Note that the chirality projector $P_R = (1 + \gamma_5)/2$ can always be commuted to the end of a spin line since we are using the basis of Ref. [20]. Thus, in our calculation, γ_5 never appears in closed fermion loops and we can use anti-commuting γ_5.

The maximum number of γ matrices in our diagrams is eleven on each of the two spin lines, which includes at least one slashed momentum per spin line. Such terms appear, e.g., in the following diagrams:

- To one-loop order, we have diagrams with two evanescent operators of the first generation which have up to five γ matrices for the penguin operators, see Fig. 2(a). The amplitudes with the most γ matrices have ten γ matrices with open Lorentz indices as well as one slashed momentum per spin line. We will call γ matrices which do not have their Lorentz index contracted with a propagator momentum *pure*.

- To two-loop order, we have diagrams with one first generation evanescent operator and one penguin operator with an additional gluon. The penguin operators have
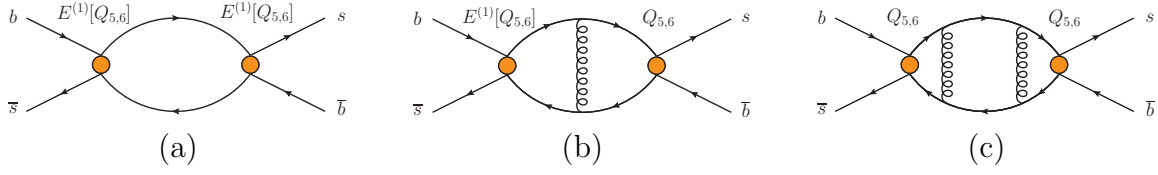
4

Figure 2: Sample diagrams at one-loop (a), two-loop (b) and three-loop (c) order which will give rise to the largest number of γ matrices in our amplitude: eleven.

> up to three γ matrices while the evanescent operator yields at most five γ matrices. The longest chain of γ matrices consists of nine pure γ matrices and two slashed momenta per spin line with a gluon connecting the two spin lines, see Fig. 2(b).

- To three-loop order, we have diagrams with two penguin insertions and two gluons. The longest spin lines result from gluons connecting across the two as shown in Fig. 2(c), so we have at most eight pure γ matrices and three slashed momenta per spin line.

We will proceed with some details on the underlying mathematics, describing the occurring structures and defining a proper set of projectors. An efficient algorithm for loop calculations based on those considerations can be found in Section 2.5.

2.2 Vector spaces spanned by γ matrices

2.2.1 Vector spaces from pure γ matrices

Focusing on pure γ matrices first, we encounter spinor structures in the amplitude which consist of the tensor product of two spin lines,

$$\Xi = \Gamma^{\mu_1,\dots,\mu_n} \otimes \Gamma_{\mu_{\sigma(1)},\dots,\mu_{\sigma(n)}}, \tag{7}$$

where $\sigma(x)$ is a particular permutation of the n indices. Any index contractions on the same spin line have been taken care of already by commuting the γ matrices together. We can commute the γ matrices in accordance with the Clifford algebra,

$$\{\gamma^\mu, \gamma^\nu\} = 2g^{\mu\nu}, \tag{8}$$

where we have chosen the naive dimensional regularisation scheme. This means that we simply treat metric tensor to be in d dimensions and work with an anticommuting γ_5. Although the spinor index on the γ matrices becomes formally infinite-dimensional, the trace of the unit matrix can still be chosen to be equal to four. At the fundamental level, the spinor structures are elements of

$$Cl_{p,q}(\mathbb{C}) \otimes Cl_{p,q}(\mathbb{C}), \tag{9}$$

5

where the spacetime is $\mathbb{R}^{p,q}$ with $p = 1$ time and $q = d - 1$ space dimensions. Since the Clifford algebra is not closed in d dimensions, we would like to further narrow down the structures that we consider and the vector space that they live in. In the following we will take a bottom-up approach by only considering the spinor structures that appear in our amplitude.

A single diagram generates Dirac chains of different lengths, and we note that the elements with different lengths of γ matrices are components of an n-tuple

$$x = \left(c^{(0)} \mathbb{1} \otimes \mathbb{1}, \ c^{(1)} \gamma^\mu \otimes \gamma_\mu, \ \sum_i c_i^{(2)} \Gamma_i^{(2)} \otimes \Gamma_i^{(2)}, \ \dots \right), \tag{10}$$

where $\Gamma_i^{(k)}$ refers to a string of k γ matrices for a particular permutation of the indices. Note that the first spin line can always be canonically labelled with indices $\mu_1, \dots, \mu_n$; the permutation refers to indices on the second spin line. Each diagram then involves one such n-tuple which lives in the vector space $\tilde{V}_n$ formed by the direct sum of the k-dimensional vector spaces $\tilde{V}^{(k)}$ of unordered γ matrices of length k, up to a maximum length n,

$$\tilde{V}_n \equiv \tilde{V}^{(0)} \oplus \tilde{V}^{(1)} \oplus \dots \oplus \tilde{V}^{(n)}. \tag{11}$$

However, there is some redundancy in this description as we can use the Clifford algebra from Eq. (8) to commute the γ matrices into a canonical order. This provides a linear map $\phi_{\text{Cliff}} : \tilde{V}_n \to V_n$, where V_n is the vector space of ordered spin lines

$$V_n = \text{span}_{\mathbb{C}} \left(\{ \mathbb{1} \otimes \mathbb{1}, \gamma^\mu \otimes \gamma_\mu, \ \dots \ , \gamma^{\mu_1} \dots \gamma^{\mu_n} \otimes \gamma_{\mu_n} \dots \gamma_{\mu_1} \} \right). \tag{12}$$

Here, we have chosen one particular ordering of the γ matrices for every possible length of the spin lines and spanned a vector space by allowing each basis element to have arbitrary complex coefficients. The elements above are linearly independent in d dimensions because we cannot use four-dimensional identities of the Dirac algebra like the Chisholm identities to reduce the number of γ matrices on a spin line. Note that we can rewrite our vector space V_n as the direct sum of the one-dimensional vector spaces $V^{(k)}$ of ordered γ matrices of length k, up to a maximum length n,

$$V_n \equiv V^{(0)} \oplus V^{(1)} \oplus \dots \oplus V^{(n)}, \tag{13}$$

which is isomorphic to $\mathbb{C}^{n+1}$ due to the isomorphism of the vector spaces $V^{(k)} \cong \mathbb{C}$. The set of vectors spanning V_n as defined in Eq. (12) constitutes a basis since they are linearly independent. Furthermore, we have $\tilde{V}^{(r)} \subset \tilde{V}^{(s)} \forall r < s$ and $V_n \subset \tilde{V}_n$. For Dirac chains of pure γ matrices, the task we are faced with, is to resolve the map ϕ_{Cliff} in an efficient manner.

2.2.2 Including slashed momenta

As a reordering of the γ matrices in a spin line is always possible using Eq. (8), we will consider ordered structures of the form

$$\Xi = \slashed{p}_1 \dots \slashed{p}_m \gamma^{\mu_1} \dots \gamma^{\mu_n} \otimes \slashed{p}_{m+1} \dots \slashed{p}_k \gamma_{\mu_n} \dots \gamma_{\mu_1}, \tag{14}$$

where we have assumed that the map between $\tilde{V}_n$ and V_n described in Section 2.2.1 has been resolved and the momenta have been commuted to one end of the spin lines already. As will be explained in Section 2.5, this will turn out to be an efficient way of tackling the problem. Note that the number of momenta k that can maximally occur has an upper bound from the number of propagators in the diagram.

Formally, the structures in Eq. (14) are elements of the vector space

$$\Xi \in \mathbb{C}^d \oplus \ldots \oplus \mathbb{C}^d \oplus V_n \cong \left(\mathbb{C}^d\right)^k \oplus V_n \cong \left(\mathbb{C}^d\right)^k \oplus \mathbb{C}^{n+1} \equiv U'_{n,k}, \tag{15}$$

where the dimension of the Lorentz index is $d = 4 - 2\epsilon$ and $\cong$ denotes an isomorphism. However, we know that the final result of carrying out the loop integral can only give a limited number of tensor structures in Lorentz space because the only available tensors are q^μ, the external momentum, and $g^{\mu\nu}$, the metric tensor. Therefore, we are interested in finding a projector which maps onto the smaller vector space

$$U_{n,k} \equiv \mathrm{span}_\mathbb{C}\left(\{\mathbb{1} \otimes \mathbb{1}, \slashed{q}_q \otimes \mathbb{1}, \mathbb{1} \otimes \slashed{q}_q, \slashed{q}_q \otimes \slashed{q}_q, \gamma^\mu \otimes \gamma_\mu, \ldots, \gamma^{\mu_1}\ldots\gamma^{\mu_n}\slashed{q}_q \otimes \gamma_{\mu_n}\ldots\gamma_{\mu_1}\slashed{q}_q\}\right), \tag{16}$$

which is simply isomorphic to $\mathbb{C}^{4(n+1)}$. We have defined

$$\slashed{q}_q \equiv \frac{\slashed{q}}{\sqrt{q^2}} \tag{17}$$

for ease of notation.

For completeness, we also define the vector space

$$\tilde{W}_{n,k} \equiv \left(\mathbb{C}^d\right)^k \oplus \tilde{V}_n, \tag{18}$$

whose elements are of the form

$$x = \left(c^{(0)}\slashed{p}_1\ldots\slashed{p}_m \otimes \slashed{p}_{m+1}\ldots\slashed{p}_k, \; c^{(1)}\slashed{p}_1\ldots\slashed{p}_m\,\gamma^\mu \otimes \slashed{p}_{m+1}\ldots\slashed{p}_k\,\gamma_\mu, \right.$$
$$\left. \sum_i c_i^{(2)}\slashed{p}_1\ldots\slashed{p}_m\,\Gamma_i^{(2)} \otimes \slashed{p}_{m+1}\ldots\slashed{p}_k\,\Gamma_i^{(2)}, \ldots \right), \tag{19}$$

in analogy with Eq. (10).

2.3 Construction of projectors

2.3.1 General projectors

The standard way of defining a set of projectors $\{P_i\}$ is to make use of an inner product $\langle\cdot,\cdot\rangle$ defined on the corresponding vector space. The projection of x onto a subspace of the vector space is

$$P(x) = \sum_i e_i P_i(x) = \sum_{i,j} e_i \lambda_{ij}\langle e_j, x\rangle, \tag{20}$$

7

where the basis elements of the subspace are denoted by e_i. The coefficient matrix λ_{ij} of the projectors can be determined from demanding that the projector maps basis elements onto themselves,

$$P_i(e_k) = \sum_j \lambda_{ij} \langle e_j, e_k \rangle = \sum_j \lambda_{ij} \mathsf{G}_{jk} \overset{!}{=} \delta_{ik}, \tag{21}$$

where we have defined the Gram matrix $\mathsf{G}_{ij} \equiv \langle e_i, e_j \rangle$. The coefficient matrix is simply given by its inverse,

$$\lambda_{ij} = \mathsf{G}_{ij}^{-1}. \tag{22}$$

The Gram matrix is invertible if and only if the set of vectors is linearly independent, which is trivially satisfied if we have chosen a proper basis.

We are thus left with finding an appropriate inner product $\langle \cdot, \cdot \rangle : V \times V \to \mathbb{C}$ which has the following properties:

(i) Conjugation symmetry: $\overline{\langle x, y \rangle} = \langle y, x \rangle$

(ii) Linearity: $\langle ax + by, z \rangle = a \langle x, z \rangle + b \langle y, z \rangle \, \forall \, a, b \in \mathbb{C}$

(iii) Positive-definiteness: $\langle x, x \rangle > 0$

Note that a projector constructed from an inner product is guaranteed to yield the expected results, but we can (and will) sacrifice property (iii) as long as we explicitly check that the Gram matrix is invertible.

2.3.2 The standard approach

To illustrate the above, consider the standard choice for an inner product (see, e.g., Refs. [21, 22]) on the vector space $W_{n,k}$. For $x, y \in W_{n,k}$, each consisting of a single term, this is

$$\langle x, y \rangle \equiv \mathrm{Tr} \left[\left(\not{p}_1 \ldots \not{p}_{m_x} \gamma^{\mu_1} \ldots \gamma^{\mu_{n_x}} \right)^\dagger \not{p}_1 \ldots \not{p}_{m_y} \gamma^{\nu_1} \ldots \gamma^{\nu_{n_y}} \right] \times$$
$$\mathrm{Tr} \left[\left(\not{p}_{m_x+1} \ldots \not{p}_{k_x} \gamma_{\mu_{\sigma(1)}} \ldots \gamma_{\mu_{\sigma(n_x)}} \right)^\dagger \not{p}_{m_y+1} \ldots \not{p}_{k_y} \gamma^{\nu_{\sigma'(1)}} \ldots \gamma^{\nu_{\sigma'(n_y)}} \right], \tag{23}$$

where the indices μ_i and ν_i stem from x and y respectively. For property (i) we can choose complex conjugation or pick the interchange of the two spin lines as our conjugation property. In the case of complex conjugation, this property is fulfilled by Eq. (23), but there is no reason why we would need this particular symmetry. The more relevant symmetry is the conjugation where the spin lines are interchanged, which is obeyed by basically anything we would write down intuitively.

8

Property (ii) is tricky when the elements have more than one term, e.g. for $x = \gamma^\mu \otimes \gamma_\mu + \gamma^\mu \gamma^\nu \otimes \gamma_\mu \gamma_\nu$ we would naively get the inner product

$$\langle x, y \rangle = \text{Tr}\left[(\gamma^\mu + \gamma^\mu \gamma^\nu)^\dagger \ldots\right] \times \text{Tr}\left[(\gamma_\mu + \gamma_\mu \gamma_\nu)^\dagger \ldots\right], \tag{24}$$

which is obviously inconsistent as the Lorentz indices are not contracted on all terms. The correct way to handle this is to introduce the scalar product to act on each component of the vector element defined in Eq. (19):

$$\langle \cdot, \cdot \rangle : \Big((x_0, x_1, \ldots, x_n), (y_0, y_1, \ldots, y_n) \Big) \mapsto \phi(x_0, y_0) + \phi(x_1, y_1) + \cdots + \phi(x_n, y_n), \tag{25}$$

where the vector index refers to the number of pure γ matrices. The map that one might sloppily call the inner product in the definition from Eq. (23) is a special case of the scalar map ϕ,

$$\phi_t : (\mathbb{C}^d)^m \oplus \tilde{V}^{(k)} \times (\mathbb{C}^d)^m \oplus \tilde{V}^{(k)} \to \mathbb{C}$$
$$(x_1 \otimes x_2, \, y_1 \otimes y_2) \mapsto \text{Tr}\left[x_1^\dagger y_1\right] \times \text{Tr}\left[x_2^\dagger y_2\right] \tag{26}$$

Note, however, that any ϕ alone does not obey linearity and hence does not constitute an inner product. The conjugation property (i) of the inner product is inherited from ϕ while linearity is explicitly enforced for different lengths of spin lines. For every spin line length, ϕ still needs to be linear for (ii) to be fulfilled, but this is something that we can now make sense of and which works for Eq. (26). Therefore it is important to differentiate between the scalar map ϕ which takes elements from $\tilde{V}^{(k)} \times \tilde{V}^{(k)}$ (or $(\mathbb{C}^d)^m \oplus \tilde{V}^{(k)} \times (\mathbb{C}^d)^m \oplus \tilde{V}^{(k)}$ when considering slashed momenta) and the inner product $\langle \cdot, \cdot \rangle$ which takes elements from $\tilde{V}_n \times \tilde{V}_n$ (or $\tilde{W}_{m,n} \times \tilde{W}_{m,n}$ when considering slashed momenta).

Finally, with the traditional map ϕ_t, the inner product also obeys property (iii), but as we will see in Section 2.3.3, this is less important.

2.3.3 An alternative path

One less appealing aspect of the standard map ϕ_t, which is used to build up the inner product, is that all Lorentz indices are contracted across the two spin lines. In automated computations this means that the intermediate expressions may become quite large once we start taking one of the traces. Moreover, there is no obvious way of parallelising such a calculation. A trace calculation arising in the projection of 9 or more γ matrices on either spin line runs multiple days on a single core even with a highly optimised programming language like FORM [23]. A possible solution is to choose a different map ϕ. For this purpose we define

$$\phi_a : (\mathbb{C}^d)^m \oplus \tilde{V}^{(k)} \times (\mathbb{C}^d)^m \oplus \tilde{V}^{(k)} \to \mathbb{C}$$
$$(x_1 \otimes x_2, \, y_1 \otimes y_2) \mapsto \text{Tr}\left[x_1 y_1 x_2 y_2\right] \tag{27}$$

This effectively glues together the two spin lines, which has the nice feature that all Lorentz indices are contracted on the same spin line. This allows for two γ matrices to be commuted together and eliminated from the spin line, thereby generating many terms which have, however, two γ matrices less per index contraction that has been carried out. As a consequence, evaluating the inner product and carrying out the trace can easily be parallelised.

It is instructive to check the properties for the inner product (i), (ii) and (iii) in this case too. We can see that linearity is obeyed and, with the interchange of the two spin lines as our conjugation of choice, property (ii) also holds. However, this map does not obey positive-definiteness (e.g. for $x = \gamma^\mu \otimes \gamma_\mu$). This can become a problem when inverting the Gram matrix.

Focusing on the subspace of pure γ matrices only for ease of notation, positive-definiteness over V_n (but not necessarily $\tilde{V}_n$) ensures that the Gram matrix is invertible as long as the set of vectors considered is a basis, i.e. linearly independent.

$$\langle x, x \rangle > 0 \ \forall \ x \in V_n : x \neq 0 \iff \mathbf{v}^\dagger \mathsf{G} \mathbf{v} > 0 \ \forall \ \mathbf{v} \in \mathbb{C}^n \cong V_n : \mathbf{v} \neq 0, \qquad (28)$$

where $\mathbf{v}$ represents an element in V_n, containing just the coefficients in front of the Dirac structures in the n-tupel. If the inner product is not positive definite, we can have eigenvectors with zero eigenvalue, making the matrix non-invertible:

$$\langle x, x \rangle > 0 \implies \forall \mathbf{v} \neq 0 : \mathsf{G} \mathbf{v} \neq 0, \qquad (29)$$

$$\exists \mathbf{v} : \mathsf{G} \mathbf{v} = 0 \iff \mathsf{G} \ \text{non-invertible.} \qquad (30)$$

It turns out that this is not a problem for the vector space of just γ matrices, but if we include insertions of the external momentum $\not{q}$ as well, the Gram matrix becomes non-invertible in some cases.

We will describe the construction of our basis and how we can avoid non-invertibility of the Gram matrix in the following. When projecting the Dirac structures occurring in the amplitude, we count the number n of γ matrices on the shorter spin line and include projectors up to that number of pure γ matrices and at most one more slashed momentum, which corresponds to the most complicated structure that can occur. If the longer of the two spin lines has more than $n + 1$ γ matrices, there must be index contractions on that spin line, reducing the number of γ matrices down to at most $n + 1$. If n is even, the Gram matrix constructed in this way is non-invertible. This can be alleviated by either extending the basis to include one more γ matrix on each spin line or by excluding the basis element with $n + n$ pure γ matrices and an additional slashed momentum on the spin line which has only n γ matrices in the amplitude. We choose the latter option which makes the Gram matrix invertible and is the computationally most efficient approach.

The case where both spin lines have the same length of eleven γ matrices can be considered separately in our case because these diagrams have ten pure γ matrices and one slashed momentum per spin line. Since extending the basis to twelve γ matrices is computationally

expensive, we instead choose a different basis as the one described above. For this specific symmetric case, the basis includes all basis elements with up to eleven γ matrices on each spin line (including slashed momenta). However, to make the Gram matrix invertible we need to exclude one asymmetric basis element which has ten γ matrices on both spin lines and a slashed momentum on one of the two. For more details, see Appendix A.

2.4 An efficient treatment of pure γ matrices

For spin lines which consist only of pure γ matrices, the problem is reduced to finding the maps

$$f_k : S_k \to \mathbb{R}^{k+1}, \tag{31}$$

where S_k is the group of permutations of k elements. The function f_k take a permutation σ of Lorentz indices, e.g. of the second spin line if the first has been labelled canonically, and returns the coefficients $\mathbf{a}$ of the ordered spinor structures. For later convenience we choose a_1 to correspond to the coefficient of the longest Dirac chain, whereas the coefficient a_{k+1} for each f_k corresponds to the coefficient of the identity. Hence, the function f_k resolves the mapping onto the vector space defined in Eq. (12). The first few mappings can be written down by commuting the γ matrices on the second spin line into canonical order:

$$f_1 : S_1 \to \mathbb{R}^2, \quad (1) \to (1,0) \tag{32}$$

$$f_2 : S_2 \to \mathbb{R}^3, \quad \begin{cases} (12) \mapsto (1,0,0) \\ (21) \mapsto (-1,0,2d) \end{cases} \tag{33}$$

$$f_3 : S_3 \to \mathbb{R}^4, \quad \begin{cases} (123) \mapsto (1,0,0) \\ (132) \mapsto (-1,0,2d,0) \\ (231) \mapsto (1,0,4-4d,0) \\ \vdots \end{cases} \tag{34}$$

The permutation notation $(\dots)$ can be read as the numbering of the Lorentz indices μ_i on the second spin line from back to front. The unpermuted elements are the basis vectors which span the vector space in Eq. (12). For example, the non-trivial permutation in Eq. (33) translates to

$$\gamma^{\mu_1}\gamma^{\mu_2} \otimes \gamma_{\mu_1}\gamma_{\mu_2} = -\gamma^{\mu_1}\gamma^{\mu_2} \otimes \gamma_{\mu_2}\gamma_{\mu_1} + 2d(\mathbb{1} \otimes \mathbb{1}). \tag{35}$$

From the above we can already see a pattern emerge. For example, the first coefficient, which corresponds to the contribution from the ordered spin line of the same length k, is always the sign of the permutation, i.e. $a_0 = \mathrm{sgn}(\sigma)$. Moreover, the odd coefficients a_{2n+1} always vanish; this is the case since any index contraction can only reduce the number of γ matrices by two on each spin line. Closed formulae for the above expressions are tricky to obtain, but generating a lookup table for all permutations of up to a number $k = 10$ of γ matrices is feasible and took $\mathcal{O}(10^6)$ single core CPU minutes.
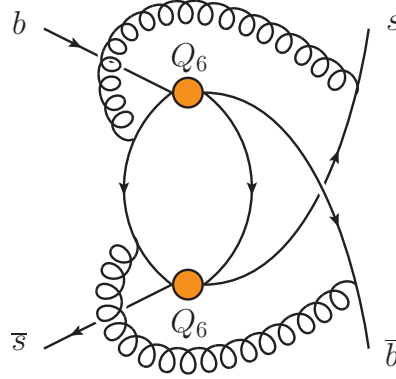
Figure 3: The spinor structure of the diagram shown here will be used as an example to illustrate how our algorithm works on the most complicated diagrams. Note that the Lorentz indices corresponding to the effective operators Q_6 are labelled as μ_i and ν_i while the gluons carry Lorentz indices α_i.

This treatment of the pure γ matrices implements the mapping of pure gamma matrices onto a minimal basis described as in Section 2.2.1 and considerably reduces the effort needed to project the encountered spinor structures in our amplitude. This is because we only need to project a limited number of momentum insertions paired with a certain number of pure γ matrices.

2.5 Implemented algorithm

With the general background outlined in the preceding sections, we are ready to outline the algorithm chosen to deal with the γ matrices in our amplitude. The goal is to map each diagram onto a sum of operator matrix elements in the $|\Delta B| = 2$ theory. This is done in two steps; first, the γ matrices are ordered canonically and then the projectors are applied. The full procedure is given below and for illustration purposes we will refer to the structures appearing at each step of the calculation of the diagram shown in Fig. 3.

(i) Project onto left-handed and right-handed spinor structures. It is sufficient to just multiply with $P_{R/L} = (1 \pm \gamma_5)/2$ and then drop all remaining terms containing γ_5. None of these can contribute to the amplitude because the left/right projectors can always be commuted through all the way to the end of a spin line and no γ_5 should appear as an overall factor. For the sample diagram, we have

$$\gamma^{\mu_1}\gamma^{\mu_2}\gamma^{\mu_3}\slashed{k}_1\gamma^{\alpha_1}\slashed{k}_2\gamma^{\alpha_2}\slashed{k}_3\gamma^{\nu_1}\gamma^{\nu_2}\gamma^{\nu_3} \otimes \gamma_{\alpha_2}\slashed{k}_4\gamma_{\mu_1}\gamma_{\mu_2}\gamma_{\mu_3}\slashed{k}_5\gamma_{\nu_1}\gamma_{\nu_2}\gamma_{\nu_3}\slashed{k}_6\gamma_{\alpha_1} + 3 \text{ terms}, \quad (36)$$

where the k_i are line momenta of the Feynman diagram in Fig. 3.

(ii) Express all line momenta in terms of loop momenta and the external momentum, which reduces the number of slashed γ matrices which may appear, e.g. for the

sample diagram:

$$\gamma^{\mu_1}\gamma^{\mu_2}\gamma^{\mu_3}\slashed{p}_3\gamma^{\alpha_1}\slashed{p}_3\gamma^{\alpha_2}\slashed{p}_2\gamma^{\nu_1}\gamma^{\nu_2}\gamma^{\nu_3} \otimes \gamma_{\alpha_2}\slashed{p}_3\gamma_{\mu_1}\gamma_{\mu_2}\gamma_{\mu_3}\slashed{p}_1\gamma_{\nu_1}\gamma_{\nu_2}\gamma_{\nu_3}\slashed{p}_2\gamma_{\alpha_1} + 95 \text{ terms} . \quad (37)$$

(iii) Reorder the momenta on both spin lines:

(a) Convert duplicate momenta on the same spin line to scalar products. This generates at most a factor of ten more terms, and in the example considered here, we obtain 419 terms after this step.

(b) Commute all slashed momenta to the left of the spin lines. This generates at most a factor of 1000 more terms.

(c) All occurrences of the external momentum are put to the very left, followed by the loop momenta. Putting the external momentum to the left of the spin lines is advantageous since some of the basis elements contain a slashed external momentum which is always on the right. Applying such a projector immediately gives a scalar product, thereby reducing the length of the trace. In the sample diagram we have 318808 terms after this step.

(d) Choose one particular permutation of the loop momenta.

After reordering all momenta, the sample diagram has the following spinor structure:

$$\slashed{p}_3\slashed{p}_2\slashed{p}_1\gamma^{\mu_2}\gamma^{\mu_3}\gamma^{\alpha_2}\gamma^{\nu_1}\gamma^{\nu_2}\gamma^{\nu_3} \otimes \slashed{p}_3\gamma_{\alpha_2}\gamma_{\mu_2}\gamma_{\mu_3}\gamma_{\nu_1}\gamma_{\nu_2}\gamma_{\nu_3} + 181510 \text{ terms} \quad (38)$$

(iv) Contract duplicate Lorentz indices on the same spin line after commuting the respective γ matrices together.

(v) Bring all pure γ matrices into a canonical order. We use a lookup table which resolves the permutation of pure γ matrices, see Section 2.4. This table contains all the mappings of permutations of up to ten γ matrices on both spin lines onto linear combinations of canonically ordered γ matrices. To generate less terms when using the map ϕ_a from Eq. (27) to carry out the projection, it will be useful to choose a canonical ordering where the γ matrices on the two spin lines are inverted with respect to each other. This way the largest number of necessary commutations is given by the length of the projector element which is sandwiched between the spin lines. For our example case, this leads to

$$\slashed{p}_3\slashed{p}_2\slashed{p}_1\gamma^{\rho_1}\gamma^{\rho_2}\gamma^{\rho_3}\gamma^{\rho_4}\gamma^{\rho_5}\gamma^{\rho_6} \otimes \slashed{p}_3\gamma_{\rho_6}\gamma_{\rho_5}\gamma_{\rho_4}\gamma_{\rho_3}\gamma_{\rho_2}\gamma_{\rho_1} + 40134 \text{ terms}. \quad (39)$$

(vi) Map the terms with ordered pure γ matrices and split-off slashed momenta directly onto basis elements which are given in Appendix A. This is done by using another lookup table which has been calculated using projectors on the ordered gamma structures. Note that the process is symmetric under interchange of the spin lines; hence, it is only necessary to calculate one of the two possibilities whenever the loop

momenta distribution is asymmetric with respect to interchange of the spin lines. When calculating the lookup table, we can also choose to use different projectors, i.e. different scalar products, depending on the spinor structure which needs to be resolved. For most structures, we choose to project using the bilinear map ϕ_a defined in Eq. (27). For the projection of the spinor structure with eleven γ matrices and slashed momenta on both spin lines, the aforementioned linear map is no longer efficient since the slashed momenta are essentially open Lorentz indices inside the trace. Thus we use the linear map ϕ_t which takes the traces over both spin lines separately as defined in Eq. 23. To optimise this calculation, the terms are split into separate files in bunches of 10^5 terms after taking the first trace. The parallelisation of the second trace makes this calculation tractable, although the size of the intermediate files generated is also a challenge. In the end, our sample diagram results in

$$\frac{(p_1 \cdot p_3)^2 p_2^2}{d^3 - 6d^2 + 11d - 6} \times B_{45} + 104335 \text{ terms}, \tag{40}$$

where $d = 4 - 2\epsilon$ is the spacetime dimension, and B_{45} is defined in Appendix A.

(vii) Map basis elements onto operator matrix elements, e.g. $\langle B_s | Q | \bar{B}_s \rangle$, $\langle B_s | \widetilde{Q}_S | \bar{B}_s \rangle$, $\langle B_s | E_1^{(1)} | \bar{B}_s \rangle$, etc. This can be done using a lookup table.

The algorithm described above is well suited for an efficient calculation since it circumvents a few bottlenecks such as:

- No redundant calculation of long traces. Each trace combination is only taken once.

- Applying the projectors by multiplying the amplitude with them inflates the number of terms by one or two orders of magnitude if the different chirality combinations are taken into account. This is avoided by using a lookup table and mapping directly onto the result.

- The number of spinor structures which need to be projected is massively reduced by the hybrid approach of sorting the γ matrices first.

3   Tensor integrals

Tensor reduction represents an alternative approach to the calculation of amplitudes on both sides of the matching equation. In comparison to the projector technique it does not require any assumptions on the Dirac and colour structures that may appear in the resulting expressions. Furthermore, when using this method, evanescent operators containing large numbers of γ matrices do not lead to any performance bottlenecks.

However, tensor reduction formulas tend to become very large with the increasing tensor rank (number of open Lorentz indices), number of loop momenta (more ways to permute

the indices) and the number of external momenta (more complicated tensor structures). In our calculation we need to consider three-loop tensors up to rank 11 with one external momentum.

The traditional way of deriving tensor reduction formulas consists of writing down the most general ansatz containing all tensor structures allowed by the symmetries. For example, in the case of a rank-three two-loop integral with one external momentum q, one can naively write

$$\int_{p1,p2} p_1^\mu p_2^\nu p_2^\rho f(p_1, p_2, q) = \int_{p1,p2} \left(g^{\mu\nu} q^\rho c_1 + g^{\mu\rho} q^\nu c_2 + g^{\nu\rho} q^\mu c_3 + q^\mu q^\nu q^\rho c_4 \right) f(p_1, p_2, q),$$

(41)

where $f(p_1, p_2, q)$ is some function containing propagator denominators and possibly scalar products appearing in the numerator of this loop integral. Its precise form is irrelevant for the actual tensor reduction. The coefficients c_i are made of scalar products containing both loop and external momenta multiplied by rational functions that depend on the dimension d.

The next step would be to contract Eq. (41) with each of the tensor structures multiplying c_i, which leads to a symbolic system of linear equations. Solving this system we can determine the c_i, thus deriving the final form of the tensor reduction formula for this integral.

In practical applications one quickly arrives at very large systems of equations that are barely solvable in a reasonable amount of time without special codes. The complexity of such equations can be greatly reduced by observing that not all c_i are independent of each other. A very simple and efficient algorithm for finding such symmetries is described in Ref. [24]. It is enough to contract the *left-hand side* of Eq. (41) with each of the tensor structures multiplying c_i and compare the resulting expressions with each other. Identical expressions mean that the corresponding c_i are the same so that there are symmetry relations that reduce the size of the system.

The whole procedure of deriving reduction formulas for arbitrary tensor integrals can be automatized using software tools. In particular, one can use the routine `Tdec` available in FEYNCALC 10 [25–28], which already implements the symmetry finding algorithm from Ref. [24]. In order to speed up the process of solving the linear system, it is also recommended to install the development version of the FEYNHELPERS[2] add-on [29]. This code provides an interface to the computer algebra system FERMAT [30] that features a very efficient linear equation solver. To make use of this, one needs to set the option `Solve` of `Tdec` to `FerSolve`.

The full MATHEMATICA code for deriving the reduction formula for the integral from Eq. (41) reads as follows

---

[2]`https://github.com/FeynCalc/feynhelpers`

```
$LoadAddOns = {"FeynHelpers"};
<< FeynCalc`
Tdec[{{p1, mu}, {p2, nu}, {p2, rho}}, {q}, Solve -> FerSolve,
 List -> False, FCE->True]
```

This produces the following output

```
(FVD[q, rho]*MTD[mu, nu]*SPD[p2, q]*(-(SPD[p1, q]*SPD[p2, q]) +
SPD[p1, p2]*SPD[q, q]))/((-1 + D)*SPD[q, q]^2) +
(FVD[q, nu]*MTD[mu, rho]*SPD[p2, q]*(-(SPD[p1, q]*SPD[p2, q]) +
SPD[p1, p2]*SPD[q, q]))/((-1 + D)*SPD[q, q]^2) +
(FVD[q, mu]*MTD[nu, rho]*SPD[p1, q]*(-SPD[p2, q]^2 +
SPD[p2, p2]*SPD[q, q]))/ ((-1 + D)*SPD[q, q]^2) +
(FVD[q, mu]*FVD[q, nu]*FVD[q, rho]*(2*SPD[p1, q]*SPD[p2, q]^2 +
D*SPD[p1, q]*SPD[p2, q]^2 - SPD[p1, q]*SPD[p2, p2]*SPD[q, q]
- 2*SPD[p1, p2]*SPD[p2, q]*SPD[q, q]))/((-1 + D)*SPD[q, q]^3)
```

where SPD denotes d-dimensional scalar products of four-momenta, MTD stands for the metric tensor and FVD describes four-vectors. Using string replacements such expressions can be easily converted into FORM id-statements and thus used outside of MATHEMATICA.

The insertion of such id-statements into amplitudes can significantly increase the number of intermediate terms, so that putting .sort commands in right places of the FORM code becomes mandatory. However, the derivation of such formulas using FEYNCALC and FERMAT was never a performance bottleneck for us. For example, to derive tensor reduction for

$$\int_{p1,p2,p3} p_1^{\mu_1} p_1^{\mu_2} p_1^{\mu_3} p_1^{\mu_4} p_1^{\mu_5} p_2^{\mu_6} p_2^{\mu_7} p_2^{\mu_8} p_2^{\mu_9} p_2^{\mu_{10}} p_2^{\mu_{11}} f(p_1, p_2, p_3, q) \tag{42}$$

we needed only 3 minutes on a modern laptop, while the resulting id-statement saved as a text file had the size of 2.8 MB.

4 Master integrals

In this Section we describe our approach for the computation of the master integrals.

To obtain results published in Refs. [10, 14] it was sufficient to perform a naive Taylor expansion of the $|\Delta B| = 1$ amplitudes in z up to $\mathcal{O}(z)$. Therefore, we were only concerned with the calculation of master integrals in the $z \to 0$ limit. Analytic results for one- and two-loop integrals can be readily found in the literature [31, 32] but most of the three-loop integrals had to be calculated from scratch. We start with the description of these analytic calculations in Section 4.1.

After the publication of Refs. [10, 14] we were able to extend our two-loop results to higher orders in z by keeping the full z-dependence during the evaluation of the amplitudes and expanding the resulting master integrals in z asymptotically. The technicalities behind this method are described in Section 4.2.

Afterwards, in Section 4.3 we describe a semi-numerical method we use to obtain results to very high orders in z for all relevant integrals at two and three loops. In particular, it was important to cover the physical range for all possible numerical values of the quark masses. Depending on the renormalization scheme, the charm and bottom quark masses typically take values between $1 - 1.8$ GeV and $4.2 - 5.0$ GeV, respectively. Allowing for different renormalization schemes of the quark masses, this leads to values for $\sqrt{z}$ between 0.20 and 0.43.

4.1 Analytic calculation for $z \to 0$

In the $z \to 0$ limit, the final three-loop master integrals can be calculated using the direct integration of their Feynman parametric representations. In the following we will describe how these calculations can be streamlined using the MATHEMATICA package FEYNCALC and the MAPLE package HYPERINT [33].

The first step is to derive the Feynman parametric representation for each of the integrals, where we set m_b to 1 as it can be recovered later using dimensional analysis. Here we make use of the routine `FCFeynmanParametrize` that is available in FEYNCALC 10 and can automatically generate such parametrizations for a wide range of multiloop integrals.

Given an integrand `int` in the propagator representation, e.g.

```
int = FAD[p1]*FAD[p2]*FAD[p1 - p2 - p3]*FAD[p1 + q]*FAD[p3 + q]*
FAD[p2 + p3 + q]*FAD[{p1 - p2, mb}]*FAD[{p3, mb}];
```

it is sufficient to evaluate[3]

```
FCFeynmanParametrize[int, {p1, p2, p3}, Names -> x,
  Indexed -> True, FCReplaceD -> {D -> 4 - 2 ep}, Simplify -> True,
  Assumptions -> {ep > 0}, FinalSubstitutions ->
  {SPD[q] -> 1, mb^2 -> 1}]
```

which returns a list of three entries. These are the integrand itself, its prefactor independent of the integration variables (the piece mostly made of Γ-functions) and the list of

---

[3]The options `Names` and `Indexed` specify that the Feynman parameters should be of the form `x[i]`, while `Simplify` and `Assumptions` allow for some simplifications of the resulting integrand. With `FCReplaceD` we can replace the number of dimensions d in favor of $4 - 2\epsilon$, while `FinalSubstitutions` is used to replace the occurring kinematic invariants with explicit values. More information on this and other FEYNCALC functions can be found in the official PDF manual available at `https://github.com/FeynCalc/feyncalc-manual/releases/tag/dev-manual`.

the Feynman parameters x_i. As we will also need the Symanzik polynomials $\mathcal{U}$ and $\mathcal{F}$ separately, we need to run[4]

```
{psi, phi} =
 FCFeynmanPrepare[int, {p1, p2, p3}, Names -> x, Indexed -> True,
   FinalSubstitutions -> {SPD[q] -> 1, mb^2 -> 1}][[1 ;; 2]]
```

Even though we are working with single-scale integrals, they involve up to 8 integrations in Feynman parameters and, furthermore, can be divergent. Therefore, trying to solve such integrals analytically using MATHEMATICA alone is not feasible. To deal with these issues we make use of HYPERINT, a MAPLE package that was developed with such calculations in mind.

When evaluating an integral with HYPERINT, one usually makes use of the Cheng-Wu theorem [34] which implies that the given integral must be projective. The projectivity property of a Feynman parametric integral means that the integrand must remain invariant under the rescaling

$$x_i \to \lambda x_i, \quad dx_i \to \lambda dx_i, \tag{43}$$

so that λ must explicitly cancel out. In FEYNCALC the projectivity can be checked using the routine `FCFeynmanProjectiveQ`.

The practical application of Cheng-Wu comes from the statement that if we have a projective integral, then the Dirac delta in its definition can be replaced with

$$\delta \left(\sum_{i \in \sigma} x_i - 1 \right) \tag{44}$$

without changing the result, where σ is an arbitrary subset of the edges. Although the choice of σ can be highly nontrivial, in practice one often opts for setting a single Feynman parameter, say x_k, to unity. The remaining Feynman parameters x_i with $i \neq k$ are then integrated from 0 to ∞.

When using HYPERINT, a Feynman parametric integral is calculated order by order in ϵ. Expanding a divergent integral in the regulator will introduce divergences in the integrations over Feynman parameters, which one normally would like to avoid. To this end, HYPERINT implements the method of analytic regularization [33, 35, 36] where the unexpanded integrand is modified in such a way, that all divergences will show up as explicit ϵ-poles, while the integrations over x_i become manifestly finite. The corresponding HYPERINT routines are called `findDivergences` and `dimregPartial`, while their FEYN-CALC counterparts (using the same algorithm as in HYPERINT) go under the names of `FCFeynmanFindDivergences` and `FCFeynmanRegularizeDivergence`.

---

[4]HYPERINT examples and documentation commonly use ψ and ϕ to denote the Symanzik polynomials $\mathcal{U}$ and $\mathcal{F}$ respectively.

Unfortunately, this technique also tends to increase the complexity of the integrand, especially if it contains a lot of divergences that need to be regularized. This is why in practice people often choose a different method called *quasi-finite basis* [37]. However, for the integrals at hand, this was not a major issue and with the exception of one 5-edge integral we kept the original basis and naively used analytic regularization.

When trying to integrate the regularized integrals using HYPERINT's `hyperInt` function, we found that in most cases the process could not be finished due to the appearance of spurious divergences and nonlinear polynomials at intermediate stages. Following the advice of the package developer[5] we were able to resolve these issues by setting the options `_hyper_abort_on_divergence` and `_hyper_ignore_nonlinear_polynomials` to `false` for all but the very last integration in x_i. In the case of the last integration these options must be set to `true`, while the option `_hyper_splitting_field` must be e.g. set to

`{RootOf(_Z^2 + 3*_Z + 1),RootOf(_Z^2 + _Z + 1),RootOf(_Z^2 + 1)};`

to allow for factorizing nonlinear polynomials using complex numbers.

Upon combining the results obtained at each order in ϵ, we still need to face two issues. First of all, in some cases HYPERINT has to integrate over the branch cut and cannot automatically determine the sign of the imaginary part. Such terms contain a δ_{x_i} which can, in principle, be $+1$ or -1. Consequently, a $\delta_{x_i}^2$ can be readily set to unity. An easy way to fix these signs consists of evaluating the same integral numerically using PYSECDEC [38–41] or FIESTA [42, 43] at several points and comparing the outcome to the analytic result, where everything apart from the δ_{x_i}'s is numerical as well. For the cases at hand we had no difficulties fixing those signs unambiguously.

Second, the analytic results obtained by HYPERINT upon using the routine `hyperInt` are usually not written in the most compact way. The standard procedure of simplifying them with the aid of the function `fibrationBasis` fails for most of our integrals. This is because they depend on Goncharov Polylogarithms (GPLs) [44] containing 6th root of unity, a class of functions that is, as of now, not included in the simplification tables shipped with HYPERINT. Following the advice of the HYPERINT developer we used the package HYPERLOGPROCEDURES [45] In most cases the command `Convert(expr,f23)`[6] was sufficient to achieve the required simplifications. Some challenges that we faced when using the code were resolved with the kind of help of the package developer[7].

Owing to the nature of our problem, where we only need the imaginary parts of the loop integrals but end up calculating the full result, the separation between real and imaginary parts required some additional work. To that aim we loaded the output of HYPERLOGPROCEDURES into MATHEMATICA and performed the final simplifications with the aid of POLYLOGTOOLS [47].

---

[5]Private communication with Erik Panzer.

[6]In HYPERLOGPROCEDURES `f23` denotes a particular alphabet related to Multiple Deligne values that were studied in Ref. [46].

[7]Private communication with Oliver Schnetz.

4.2 Asymptotic expansion around $z = 0$

As far as the two-loop integrals are concerned, analytic results for the expansion around $z = 0$ can be easily obtained using the method of regions [48]. These results were relevant for extending our theoretical predictions from [10] beyond $\mathcal{O}(z)$. Here we would like to explain how the technique of asymptotic expansions can be applied in a highly automatized fashion using publicly available software tools including FEYNCALC 10.

The danger of overlooking a contributing region by having an inappropriate routing of the external momentum can be avoided by using the program ASY [49, 50]. This code can reliably identify all regions relevant for the given scaling hierarchy, but its output is not always straightforward to interpret. While ASY assumes that the expansion will be carried out on the level of Feynman parametric integrals, in practice one would rather like to expand in masses and momenta. The latter approach allows us to use integration-by-parts (IBP) reduction so that a calculation to any order in z can be always reduced to the same set of master integrals.

Let us remark that such an expansion is not always possible for any kind of loop integral. For example, in Soft-Collinear Effective theory (SCET) [51–54] we need to deal with integrals, where different components of four-momenta decomposed along the light-cone vectors n and $\bar{n}$ scale in a different way. In that case it is clearly not possible to expand the integral in the original four-momenta. However, in the case at hand such complications do not occur.

To be more specific, let us discuss the analysis of the following two-loop integral with external momentum $q^2 = m_b^2$ using ASY

$$\int_{p_1,p_2} \frac{1}{-p_1^2} \frac{1}{m_c^2 - (p_1 + q)^2} \frac{1}{-p_2^2} \frac{1}{-(p_1 + p_2 + q)^2}, \tag{45}$$

where we want to expand in the limit $m_c \ll m_b$.

Using the code

```
Get["asy21.m"]
SetOptions[QHull, Executable -> "/usr/bin/qhull"];
props = {-p1^2, mc^2 - (p1 + q)^2, -p2^2, -(p1 + p2 + q)^2}
res = AlphaRepExpand[{p1, p2},
  props, {q^2 -> mb^2}, {mb -> x^0, mc -> x^1}, Verbose -> False]
```

we get the output

$$(0, -2, -2, -2),$$
$$(0, \quad 0, \quad 0, \quad 0). \tag{46}$$

Notice that switching the first two propagators as in

```
props = {mc^2 - (p1 + q)^2, -p1^2, -p2^2, -(p1 + p2 + q)^2}
```

will produce

$$(0, 0, 0, 0),$$
$$(0, 2, 0, 0), \tag{47}$$

which is in fact equivalent to Eq. (46).

The row vectors returned by ASY denote the scalings of propagators in each contributing region, while their number gives us the number of regions. The scalings in each row vector are always relative with respect to each other, which allows us to "normalize" such a vector by adding the same integer to each of its entries. Applying this procedure to the first vector in Eq. (46) leads to the second vector in Eq. (47).

To understand the meaning of those scalings, it is convenient to think of the row vectors as of lists of powers of a quantity Λ that describes the scaling of each inverse propagator. Then, the first output corresponds to

$$(0, -2, -2, -2) \Leftrightarrow (\Lambda^0, \Lambda^{-2}, \Lambda^{-2}, \Lambda^{-2}),$$
$$(0, \quad 0, \quad 0, \quad 0) \Leftrightarrow (\Lambda^0, \Lambda^0, \Lambda^0, \Lambda^0), \tag{48}$$

meaning that in the first region the contribution of $[-p_1^2]$ is much larger than that of the other three inverse propagators. The second output given by

$$(0, 0, 0, 0) \Leftrightarrow (\Lambda^0, \Lambda^0, \Lambda^0, \Lambda^0),$$
$$(0, 2, 0, 0) \Leftrightarrow (\Lambda^0, \Lambda^2, \Lambda^0, \Lambda^0), \tag{49}$$

which essentially tells us the same thing. This explains why we are allowed to normalize these row vectors. Multiplying each entry by some power of Λ obviously will not change the relative scalings between the propagators.

The output

$$(2, 0, 0, 0) \tag{50}$$
$$(0, 0, 0, 0), \tag{51}$$

tells us that the asymptotic expansion of our integral in the limit $m_c \ll m_b$ receives contributions from two regions.

In the first region the inverse propagators scale as

$$[-p_1^2] \sim m_b^2, \quad [m_c^2 - (p_1 + q)^2] \sim [-p_2^2] \sim [-(p_1 + p_2 + q)^2] \sim m_c^2. \tag{52}$$

Here it is convenient to perform the shift $p_1 \to p_1 - q$, as momentum shifts do not alter the scalings of the line momenta. With

$$[-(p_1 - q)^2] \sim m_b^2, \quad [m_c^2 - p_1^2] \sim [-p_2^2] \sim [-(p_1 + p_2)^2] \sim m_c^2, \tag{53}$$

21

we can easily deduce that $p_1 \sim p_2 \sim m_c$. The corresponding integral that needs to be expanded according to $p_1 \sim p_2 \sim m_c \ll m_b$ is then

$$\int_{p_1,p_2} \frac{1}{[-(p_1-q)^2]} \frac{1}{[m_c^2 - p_1^2]} \frac{1}{[-p_2^2]} \frac{1}{[-(p_1+p_2)^2]}$$
$$= \frac{1}{[-m_b^2]} \int_{p_1,p_2} \frac{1}{[m_c^2 - p_1^2]} \frac{1}{[-p_2^2]} \frac{1}{[-(p_1+p_2)^2]} + \mathcal{O}(m_c^2). \tag{54}$$

In the second region the inverse propagators scale as

$$[-p_1^2] \sim [m_c^2 - (p_1+q)^2] \sim [-p_3^2] \sim [-(p_1+p_3+q)^2] \sim m_b^2. \tag{55}$$

Without doing any shifts we immediately find that $m_c \ll p_1 \sim p_2 \sim m_b$ so that

$$\int_{p_1,p_2} \frac{1}{[-p_1^2]} \frac{1}{[m_c^2 - (p_1+q)^2]} \frac{1}{[-p_2^2]} \frac{1}{[-(p_1+p_2+q)^2]}$$
$$= \int_{p_1,p_2} \frac{1}{[-p_1^2]} \frac{1}{[-(p_1+q)^2]} \frac{1}{[-p_2^2]} \frac{1}{[-(p_1+p_2+q)^2]} + \mathcal{O}(m_c^2) \tag{56}$$

Once it is understood how the integral should be expanded in each region, we can proceed to the actual calculation. Instead of doing the corresponding expansions manually, this procedure can be automated by making use of the functionality present in FEYNCALC and FEYNHELPERS.

We need to define a separate `FCTopology`[8] object for each region, where the propagators stem from the original integral modulo shifts needed to make the expansion in momenta or masses well defined. Then we introduce a scaling variable λ that will be used for expanding the propagators and doing the power counting. Each four-momentum or mass appearing in the integral must be assigned a definite counting by multiplying them with a proper power of λ.

This can be automated using a routine called `FCLoopAddScalingParameter`, where the first argument takes the topology and the second one contains a list of scaling rules. All quantities that have not been assigned a scaling rule will be assumed to scale as λ^0. We also need to declare λ as `FCVariable` to help the code distinguishing between variables and names of momenta inside relevant FEYNCALC symbols

```
DataType[la,FCVariable]=True;
topoScaledR1=FCLoopAddScalingParameter[topoR1,la,
 {p1->la^1 p1,p2->la^1 p2,q->la^0 q,mb->la^0 mb, mc->la^1 mc}];
topoScaledR2=FCLoopAddScalingParameter[topoR2,la,
 {p1->la^0 p1,p2->la^0 p2,q->la^0 q,mb->la^0 mb, mc->la^1 mc}];
```

---

[8]In FEYNCALC `FCTopology` describes the given integral family and contains information about its name, propagators, loop and external momenta as well as kinematic constraints.

After that we can perform an expansion in λ up to the required power with the aid of `FCLoopGLIExpand`. While the original integral is written as `GLI`[9] symbol, e.g. `GLI[prop2L,1,1,1,1]`, the expansion in λ will create a linear combination of `GLI`'s with different integer powers of propagators multiplied by scalar products involving loop momenta.

To process this output further, we first apply `FCLoopFromGLI`, thus converting everything into propagator representation. Then we handle propagators appearing in the expansion such as $-2p_1 \cdot q + m_b^2$ and alike. Although FEYNCALC can represent pretty arbitrary propagator denominators using `GFAD` (generic `FeynAmpDenominator`) containers, most loop-related routines still expect the input to be made of quadratic or linear propagators only which are written as `SFAD`s (standard `FeynAmpDenominator`). The conversion of suitable `GFAD`s into `SFAD`s can be automatized using the routine `FromGFAD`, where the user may also add custom rules for cases where the algorithm fails to find a suitable mapping.

After that, the resulting expression can be processed via the standard FEYNCALC toolchain for converting a loop amplitude in the propagator representation into a linear combination of `GLI`s with the corresponding list of `FCTopology` symbols. To that end we apply `FCLoopFindTopologies` (naive topology identification), `FCLoopBasisFindCompletion` and `FCLoopCreateRuleGLIToGLI` (handling incomplete propagator bases), `FCLoopFindTopologyMappings` (topology minimization using the algorithm of Ref. [24]), `FCLoopTensorReduce` (tensor reduction) and finally `FCLoopApplyTopologyMappings` (elimination of explicit loop momenta an introduction of `GLI`s). The resulting amplitude can be readily IBP-reduced with any suitable software, where for the sake of convenience we choose FIRE [55–57]. The FEYNHELPERS interface to FIRE allows us to carry out the reduction without leaving the MATHEMATICA notebook, so upon running `FIREPrepareStartFile`, `FIRECreateConfigFile`, `FIRECreateIntegralFile`, `FIRECreateLiteRedFiles`, `FIRECreateStartFile` and finally `FIRERunReduction`, we can directly import the reduction tables using `FIREImportResults`.

Lastly we need to add up contributions from both regions, identify the appearing master integrals and substitute the corresponding analytic expressions. As long as the masters are sufficiently simple to be calculated analytically, this procedure allows us to obtain analytic results for the asymptotic expansion of any suitable integral up to any order. The results can be then checked with the aid of FIESTA and PYSECDEC.

Sample MATHEMATICA code which illustrates the approach discussed in this subsection is included in the supplementary material to this paper.

---

[9]In FEYNCALC a `GLI` denotes a generic loop integral characterized by its name (which must match that of a suitable `FCTopology`) and the powers of propagators.

4.3 Precise semi-analytic result for physical charm quark masses

In this subsection we present an efficient algorithm which can be used to compute the master integrals which appear in the $|\Delta B| = 1$ theory. For convenience we introduce the linear mass ratio and define

$$x = \sqrt{z}. \tag{57}$$

To cover the physically interesting region we apply the "expand and match" approach [58–61]. In the following we describe the implementation for the problem at hand. The starting point is the list of master integrals and the corresponding differential equation in x. Note that we are only interested in the absorptive part of the three-loop amplitude. Thus, we eliminate those master integrals from the system which have no imaginary part. This reduces the system by about 30% and leaves us with 342 master integrals.

The basic idea of "expand and match" is to construct expansions around properly chosen values x_E with the help of the differential equations. In order to determine all expansion coefficients, the expansions have to be matched to (numerical or analytical) results for the master integrals at a point x_M which is sufficiently close to x_E such that the series expansion converges. If x_E is a regular point, i.e., the master integrals have no threshold for $x = x_E$, it is possible to choose $x_M = x_E$.

The physical amplitude for B meson mixing has cuts through $0, 1, 2, 3$ and 4 charm quarks and thus we expect thresholds for $x \in \{0, 1/4, 1/3, 1/2, 1\}$. In fact, we observe poles in the differential equations for these values of x. Since our results cover the range $0 \le x \lesssim 0.4$ we only have to take care of first three entries in this list. We observe a good coverage of the x range in case we add a further expansion point at $x_E = 1/10$ such that we have

$$x_E \in \{0, 1/10, 1/4, 1/3\}. \tag{58}$$

Depending on the nature of the expansion point a different ansatz has to be chosen. For our application the most general ansatz for a master integral can be written as

$$M_i(x, \epsilon) = \sum_{j=-2}^{\epsilon_{\max}} \sum_{m}^{j+2} \sum_{n}^{n_{\max}} c_{i,j,m,n} \epsilon^j (x - x_E)^{n/2} \log^m (x - x_E), \tag{59}$$

where we have used that in the imaginary part of three-loop two-point functions one has at most $1/\epsilon^2$ poles. $\epsilon_{\max}$ is determined by the spurious pole in front of the respective master integral. We have $\epsilon_{\max} = 0$ if they are absent. Furthermore, we allow for logarithmic terms with an increase in the power by one unit for each ϵ order. These terms are important for the expansions around $x_E = 0, 1/4$ and $1/3$. Finally, n can either take only even values or can have both even and odd numerical value. The latter is important for the expansion around $x_M = 1/4$ since there are square roots in the expansion around cuts involving and even number of massive particles [62–64]. For $x_M = 0, 1/10, 1/3$ only the terms with even n have a non-zero coefficient $c_{i,j,m,n}$. As upper limit of the summation of n we typically have $n_{\max} = 100$.

Numerical check	Upper bound on relative difference
At $x = 1/20$	10^{-43} with $x = 0$ expansion
	10^{-22} with $x = 1/10$ expansion
At $x = 1/5$	10^{-11} with $x = 0$ expansion
	10^{-6} with $x = 1/10$ expansion
	10^{-35} with $x = 1/4$ expansion
At $x = 3/10$	10^{-2} with $x = 1/4$ expansion
	10^{-30} with $x = 1/3$ expansion

Table 1: Relative differences of the series expansions of the integral shown in Fig. 4 with numerical evaluations in `AMFlow` away from the expansion points.

For each expansion point we have a dedicated `AMFlow` [65] run for the computation of all master integrals with a precision of 100 digits. We find it convenient to choose $x_M \in \{1/100, 1/10, 6/25, 29/100\}$.

In Tab. 1 we show for three values of x the relative difference of the expansions around the values given in Eq. (58) for the master integral shown in Fig. 4. We only consider results for the ϵ^0 term of this master integral which enters the finite term of the physical amplitude. One observes that for each value there is at least one expansion which shows an agreement of 20 digits or more. It is expected that the precision gets even better in case x is chosen closer to the expansion point. We note the the precision of the expansion could be improved by increasing the precision of the boundary integrals or by performing a deeper expansion.
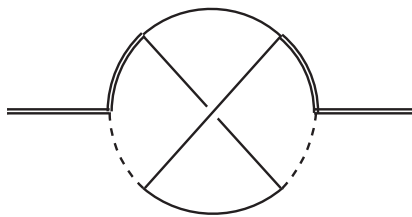

Figure 4: A sample master integral which appears at three-loop with two $Q_{1,2}$ insertions. The dashed, single and double lines denote $m = 0$, $m = m_c$ and $m = m_b$ respectively.

As a further check for the precision of our result we can compare the coefficients of the z^0 and z^1 terms of the matching coefficients H^{ab} and $\widetilde{H}^{ab}_S$ introduced in Eq. (4). Our numerical expressions agree with the analytic results from Ref. [14] at the level of 10^{-40} or better.

To illustrate the structure of our results we provide explicit results for all 342 master integrals needed for the computation of the two-point functions involving current-current operators in an expansion for $x \to 0$. The computer-readable expressions can be down-

loaded from the website [66]. After the inclusion of penguin operators it might be that there are additional master integrals. However, they can also be computed following the approach described in this Subsection.

5 Sample result

In the following we briefly illustrate the methods described in this paper and give the non-fermionic (i.e. $n_f = 0$) NNLO contribution of two Q_1 insertions to the matching coefficient $H^{cc}(x)$. The expansion for $x \to 0$ up to x^{10} reads:

$$
\begin{aligned}
H^{cc,(2)}(x) &= \Big(61.49166 + (334.9196 - 1227.224 \log x + 88.00000 \log^2 x)\, x^2 - 11.69731\, x^3 \\
&\quad + (-4522.778 + 1030.276 \log x + 376.7253 \log^2 x)\, x^4 + 199.3985\, x^5 \\
&\quad + (-5312.014 + 6117.164 \log x + 1408.251 \log^2 x)\, x^6 + 500.4536\, x^7 \\
&\quad + (-1386.574 + 13388.09 \log x - 773.9628 \log^2 x)\, x^8 + 1591.938\, x^9 \\
&\quad + (-82.72167 + 27995.10 \log x - 1593.620 \log^2 x)\, x^{10} + \mathcal{O}(x^{11}) \Big)\, C_1^2 \,. \quad (60)
\end{aligned}
$$

Here we have set the renormalisation scale $\mu = m_b$ and the quark masses are renormalised in the pole scheme. Note that the $\Delta B = 2$ evanescent operators were defined as described in Ref. [11]. All other $\mathcal{O}(\epsilon)$ contributions of physical operators to evanescent operators that were not previously defined have been set to zero. The coefficients of x^0 and x^2 are in agreement with the results obtained in Ref. [14].

6 Conclusions

In this work we address several technical challenges which occur in the calculation of the parameters necessary to describe the mixing of neutral B and D mesons. Practical solutions have been presented for the construction of projectors to scalar functions and for computing traces with more than 20 γ matrices. Furthermore, the computation of high-rank tensor integrals and the semi-analytic computation of three-loop master integrals with a dependence on the ratio of the charm and bottom quark masses has been presented. Our algorithms can also be applied to other processes, as, e.g., the computation of decay rates of B and D mesons.

Acknowledgements

A   Basis elements for spinor structures

When applying the projectors described in Section 2, we use a subset of the basis elements shown below. As explained previously, we distinguish the following cases:

(i) Spin line one is longer and spin line two has length n. Choose basis with elements up to n γ matrices on each spin line as well as the structure with a slashed momentum on spin line one in addition to the n pure γ matrices, i.e. up to and including $B_{4(n+1)-2}$.

(ii) Spin line two is longer or of the same length and spin line one has length n. Choose basis with elements up to n γ matrices on each spin line as well as the structure with a slashed momentum on spin line two in addition to the n pure γ matrices, i.e. up to and including $B_{4(n+1)-1}$ but excluding $B_{4(n+1)-2}$.

(iii) Both spin lines have length eleven and include at least one slashed momentum. This is the longest structure we need to resolve, and we choose a special basis so as not to take unnecessarily long traces. We include all elements up to and including B_{45} ($B_{4(n+1)+1}$ in the notation where n is the number of pure γ matrices) but drop B_{43} ($B_{4(n+1)-1}$ in the general case). We do not need B_{42} ($B_{4(n+1)-2}$ in the general case) to project onto, but it is required to make the Gram matrix invertible. This inconvenience arises from the bilinear map ϕ defined in Section 2.3.3 not being positive-definite.

The basis elements are:

$$B_1 = \mathbb{1} \otimes \mathbb{1}$$
$$B_2 = \slashed{q}_q \otimes \mathbb{1}$$
$$B_3 = \mathbb{1} \otimes \slashed{q}_q$$
$$B_4 = \slashed{q}_q \otimes \slashed{q}_q$$

$$B_5 = \gamma_{\mu_1} \otimes \gamma^{\mu_1}$$
$$B_6 = \gamma_{\mu_1}\slashed{q}_q \otimes \gamma^{\mu_1}$$
$$B_7 = \gamma_{\mu_1} \otimes \gamma^{\mu_1}\slashed{q}_q$$
$$B_8 = \gamma_{\mu_1}\slashed{q}_q \otimes \gamma^{\mu_1}\slashed{q}_q$$

$$B_9 = \gamma_{\mu_1}\gamma_{\mu_2} \otimes \gamma^{\mu_2}\gamma^{\mu_1}$$

27

$$B_{10} = \gamma_{\mu_1}\gamma_{\mu_2}\not{q} \otimes \gamma^{\mu_2}\gamma^{\mu_1}$$
$$B_{11} = \gamma_{\mu_1}\gamma_{\mu_2} \otimes \gamma^{\mu_2}\gamma^{\mu_1}\not{q}$$
$$B_{12} = \gamma_{\mu_1}\gamma_{\mu_2}\not{q} \otimes \gamma^{\mu_2}\gamma^{\mu_1}\not{q}$$

$$B_{13} = \gamma_{\mu_1}\ldots\gamma_{\mu_3} \otimes \gamma^{\mu_3}\gamma^{\mu_2}\gamma^{\mu_1}$$
$$B_{14} = \gamma_{\mu_1}\ldots\gamma_{\mu_3}\not{q} \otimes \gamma^{\mu_3}\gamma^{\mu_2}\gamma^{\mu_1}$$
$$B_{15} = \gamma_{\mu_1}\ldots\gamma_{\mu_3} \otimes \gamma^{\mu_3}\gamma^{\mu_2}\gamma^{\mu_1}\not{q}$$
$$B_{16} = \gamma_{\mu_1}\ldots\gamma_{\mu_3}\not{q} \otimes \gamma^{\mu_3}\gamma^{\mu_2}\gamma^{\mu_1}\not{q}$$

$$B_{17} = \gamma_{\mu_1}\ldots\gamma_{\mu_4} \otimes \gamma^{\mu_4}\ldots\gamma^{\mu_1}$$
$$B_{18} = \gamma_{\mu_1}\ldots\gamma_{\mu_4}\not{q} \otimes \gamma^{\mu_4}\ldots\gamma^{\mu_1}$$
$$B_{19} = \gamma_{\mu_1}\ldots\gamma_{\mu_4} \otimes \gamma^{\mu_4}\ldots\gamma^{\mu_1}\not{q}$$
$$B_{20} = \gamma_{\mu_1}\ldots\gamma_{\mu_4}\not{q} \otimes \gamma^{\mu_4}\ldots\gamma^{\mu_1}\not{q}$$

$$B_{21} = \gamma_{\mu_1}\ldots\gamma_{\mu_5} \otimes \gamma^{\mu_5}\ldots\gamma^{\mu_1}$$
$$B_{22} = \gamma_{\mu_1}\ldots\gamma_{\mu_5}\not{q} \otimes \gamma^{\mu_5}\ldots\gamma^{\mu_1}$$
$$B_{23} = \gamma_{\mu_1}\ldots\gamma_{\mu_5} \otimes \gamma^{\mu_5}\ldots\gamma^{\mu_1}\not{q}$$
$$B_{24} = \gamma_{\mu_1}\ldots\gamma_{\mu_5}\not{q} \otimes \gamma^{\mu_5}\ldots\gamma^{\mu_1}\not{q}$$

$$B_{25} = \gamma_{\mu_1}\ldots\gamma_{\mu_6} \otimes \gamma^{\mu_6}\ldots\gamma^{\mu_1}$$
$$B_{26} = \gamma_{\mu_1}\ldots\gamma_{\mu_6}\not{q} \otimes \gamma^{\mu_6}\ldots\gamma^{\mu_1}$$
$$B_{27} = \gamma_{\mu_1}\ldots\gamma_{\mu_6} \otimes \gamma^{\mu_6}\ldots\gamma^{\mu_1}\not{q}$$
$$B_{28} = \gamma_{\mu_1}\ldots\gamma_{\mu_6}\not{q} \otimes \gamma^{\mu_6}\ldots\gamma^{\mu_1}\not{q}$$

$$B_{29} = \gamma_{\mu_1}\ldots\gamma_{\mu_7} \otimes \gamma^{\mu_7}\ldots\gamma^{\mu_1}$$
$$B_{30} = \gamma_{\mu_1}\ldots\gamma_{\mu_7}\not{q} \otimes \gamma^{\mu_7}\ldots\gamma^{\mu_1}$$
$$B_{31} = \gamma_{\mu_1}\ldots\gamma_{\mu_7} \otimes \gamma^{\mu_7}\ldots\gamma^{\mu_1}\not{q}$$
$$B_{32} = \gamma_{\mu_1}\ldots\gamma_{\mu_7}\not{q} \otimes \gamma^{\mu_7}\ldots\gamma^{\mu_1}\not{q}$$

$$B_{33} = \gamma_{\mu_1}\ldots\gamma_{\mu_8} \otimes \gamma^{\mu_8}\ldots\gamma^{\mu_1}$$
$$B_{34} = \gamma_{\mu_1}\ldots\gamma_{\mu_8}\not{q} \otimes \gamma^{\mu_8}\ldots\gamma^{\mu_1}$$
$$B_{35} = \gamma_{\mu_1}\ldots\gamma_{\mu_8} \otimes \gamma^{\mu_8}\ldots\gamma^{\mu_1}\not{q}$$
$$B_{36} = \gamma_{\mu_1}\ldots\gamma_{\mu_8}\not{q} \otimes \gamma^{\mu_8}\ldots\gamma^{\mu_1}\not{q}$$

$$B_{37} = \gamma_{\mu_1} \dots \gamma_{\mu_9} \otimes \gamma^{\mu_9} \dots \gamma^{\mu_1}$$
$$B_{38} = \gamma_{\mu_1} \dots \gamma_{\mu_9} \slashed{\phi}_q \otimes \gamma^{\mu_9} \dots \gamma^{\mu_1}$$
$$B_{39} = \gamma_{\mu_1} \dots \gamma_{\mu_9} \otimes \gamma^{\mu_9} \dots \gamma^{\mu_1} \slashed{\phi}_q$$
$$B_{40} = \gamma_{\mu_1} \dots \gamma_{\mu_9} \slashed{\phi}_q \otimes \gamma^{\mu_9} \dots \gamma^{\mu_1} \slashed{\phi}_q$$

$$B_{41} = \gamma_{\mu_1} \dots \gamma_{\mu_{10}} \otimes \gamma^{\mu_{10}} \dots \gamma^{\mu_1}$$
$$B_{42} = \gamma_{\mu_1} \dots \gamma_{\mu_{10}} \slashed{\phi}_q \otimes \gamma^{\mu_{10}} \dots \gamma^{\mu_1}$$
$$B_{43} = \gamma_{\mu_1} \dots \gamma_{\mu_{10}} \otimes \gamma^{\mu_{10}} \dots \gamma^{\mu_1} \slashed{\phi}_q$$
$$B_{44} = \gamma_{\mu_1} \dots \gamma_{\mu_{10}} \slashed{\phi}_q \otimes \gamma^{\mu_{10}} \dots \gamma^{\mu_1} \slashed{\phi}_q$$

$$B_{45} = \gamma_{\mu_1} \dots \gamma_{\mu_{11}} \otimes \gamma^{\mu_{11}} \dots \gamma^{\mu_1} \tag{61}$$

Here, we defined

$$\slashed{\phi}_q \equiv \frac{\slashed{q}}{\sqrt{q^2}}, \tag{62}$$

B   Effective operators

For completeness we provide in this Appendix all relevant operator for the $|\Delta B| = 1$ and $|\Delta B| = 2$ theories.

On the $|\Delta B| = 1$ side we work with the operator basis first introduced in Ref. [20]

$$\mathcal{H}_{\text{eff}}^{|\Delta B| = 1} = \frac{4 G_F}{\sqrt{2}} \left[-\lambda_t^s \Big(\sum_{i=1}^{6} C_i Q_i + C_8 Q_8 \Big) - \lambda_u^s \sum_{i=1}^{2} C_i (Q_i - Q_i^u) \right.$$
$$\left. + V_{us}^* V_{cb} \sum_{i=1}^{2} C_i Q_i^{cu} + V_{cs}^* V_{ub} \sum_{i=1}^{2} C_i Q_i^{uc} \right] + \text{h.c.}, \tag{63}$$

with

$$\lambda_a^s = V_{as}^* V_{ab}, \tag{64}$$

where $a = u, c, t$ and $\lambda_t = -\lambda_c - \lambda_u$. G_F stands for the Fermi constant. The physical operators read

$$Q_1 = \bar{s}_L \gamma_\mu T^a c_L \, \bar{c}_L \gamma^\mu T^a b_L,$$
$$Q_2 = \bar{s}_L \gamma_\mu c_L \, \bar{c}_L \gamma^\mu b_L,$$
$$Q_3 = \bar{s}_L \gamma_\mu b_L \sum_q \bar{q} \gamma^\mu q,$$

$$Q_4 = \bar{s}_L\gamma_\mu T^a b_L \sum_q \bar{q}\gamma^\mu T^a q \,,$$

$$Q_5 = \bar{s}_L\gamma_{\mu_1}\gamma_{\mu_2}\gamma_{\mu_3}b_L \sum_q \bar{q}\gamma^{\mu_1}\gamma^{\mu_2}\gamma^{\mu_3}q \,,$$

$$Q_6 = \bar{s}_L\gamma_{\mu_1}\gamma_{\mu_2}\gamma_{\mu_3}T^a b_L \sum_q \bar{q}\gamma^{\mu_1}\gamma^{\mu_2}\gamma^{\mu_3}T^a q \,,$$

$$Q_8 = \frac{g_s}{16\pi^2}m_b\,\bar{s}_L\sigma^{\mu\nu}T^a b_R\,G^a_{\mu\nu}\,,$$

$$Q_1^u = \bar{s}_L\gamma_\mu T^a u_L\,\bar{u}_L\gamma^\mu T^a b_L\,,$$

$$Q_2^u = \bar{s}_L\gamma_\mu u_L\,\bar{u}_L\gamma^\mu b_L\,,$$

$$Q_1^{cu} = \bar{s}_L\gamma_\mu T^a u_L\,\bar{c}_L\gamma^\mu T^a b_L\,,$$

$$Q_2^{cu} = \bar{s}_L\gamma_\mu u_L\,\bar{c}_L\gamma^\mu b_L\,,$$

$$Q_1^{uc} = \bar{s}_L\gamma_\mu T^a c_L\,\bar{u}_L\gamma^\mu T^a b_L\,,$$

$$Q_2^{uc} = \bar{s}_L\gamma_\mu c_L\,\bar{u}_L\gamma^\mu b_L\,. \tag{65}$$

Here we introduced the left-chiral projector $P_L = (1 - \gamma_5)/2$ so that $q_L = P_L q$. We have current-current operators Q_i, Q_i^u, Q_i^{uc} and Q_i^{cu} ($i = 1, 2$), four-quark penguin operators $Q_3, \ldots, Q_6$ and the chromomagnetic penguin operator Q_8 featuring the gluon field strength $G^a_{\mu\nu}$, the strong coupling g_s and $\sigma^{\mu\nu} = i[\gamma^\mu, \gamma^\nu]/2$. In the definitions of four-quark penguin operators the sum over q is understood to run over all five flavors u, d, s, c and b.

The LO evanescent operators required at NLO accuracy and beyond read

$$E_1[Q_1] = \bar{s}_L\gamma^{\mu_1}\gamma^{\mu_2}\gamma^{\mu_3}T^a c\,\bar{c}\gamma_{\mu_1}\gamma_{\mu_2}\gamma_{\mu_3}T^a b_L - 16Q_1\,,$$

$$E_1[Q_2] = \bar{s}_L\gamma^{\mu_1}\gamma^{\mu_2}\gamma^{\mu_3}c_i\,\bar{c}_j\gamma_{\mu_1}\gamma_{\mu_2}\gamma_{\mu_3}b_L - 16Q_2\,,$$

$$E_1[Q_5] = \bar{s}_L\gamma^{\mu_1}\gamma^{\mu_2}\gamma^{\mu_3}\gamma^{\mu_4}\gamma^{\mu_5}b_L \sum_q \bar{q}\gamma_{\mu_1}\gamma_{\mu_2}\gamma_{\mu_3}\gamma_{\mu_4}\gamma_{\mu_5}q_j - 20Q_5 + 64Q_3\,,$$

$$E_1[Q_6] = \bar{s}_L\gamma^{\mu_1}\gamma^{\mu_2}\gamma^{\mu_3}\gamma^{\mu_4}\gamma^{\mu_5}T^a b_L \sum_q \bar{q}\gamma_{\mu_1}\gamma_{\mu_2}\gamma_{\mu_3}\gamma_{\mu_4}\gamma_{\mu_5}T^a q - 20Q_6 + 64Q_4\,. \tag{66}$$

Similar evanescent operators are needed for Q_i^u, Q_i^{uc} and Q_i^{cu} with $i = 1, 2$; they are obtained by replacing one or both c fields with u fields in Eq. (66).

At NNLO one has to introduce the so-called NLO evanescent operators which are given by [67]

$$E_2[Q_1] = \bar{s}_L\gamma^{\mu_1}\gamma^{\mu_2}\gamma^{\mu_3}\gamma^{\mu_4}\gamma^{\mu_5}T^a c\,\bar{c}\gamma_{\mu_1}\gamma_{\mu_2}\gamma_{\mu_3}\gamma^{\mu_4}\gamma^{\mu_5}T^a b_L - 20E_1[Q_2] - 256Q_1\,,$$

$$E_2[Q_2] = \bar{s}_L\gamma^{\mu_1}\gamma^{\mu_2}\gamma^{\mu_3}\gamma^{\mu_4}\gamma^{\mu_5}c_i\,\bar{c}_j\gamma_{\mu_1}\gamma_{\mu_2}\gamma_{\mu_3}\gamma^{\mu_4}\gamma^{\mu_5}b_L - 20E_1[Q_2] - 256Q_2\,,$$

$$E_2[Q_5] = \bar{s}_L\gamma^{\mu_1}\gamma^{\mu_2}\gamma^{\mu_3}\gamma^{\mu_4}\gamma^{\mu_5}\gamma^{\mu_6}\gamma^{\mu_7}b_L \sum_q \bar{q}\gamma_{\mu_1}\gamma_{\mu_2}\gamma_{\mu_3}\gamma_{\mu_4}\gamma_{\mu_5}\gamma^{\mu_6}\gamma^{\mu_7}q_j - 336Q_5 + 1280Q_3\,,$$

$$E_2[Q_6] = \bar{s}_L\gamma^{\mu_1}\gamma^{\mu_2}\gamma^{\mu_3}\gamma^{\mu_4}\gamma^{\mu_5}\gamma^{\mu_6}\gamma^{\mu_7}T^a b_L \sum_q \bar{q}\gamma_{\mu_1}\gamma_{\mu_2}\gamma_{\mu_3}\gamma_{\mu_4}\gamma_{\mu_5}\gamma^{\mu_6}\gamma^{\mu_7}T^a q$$

$$- 336Q_6 + 1280Q_4\,. \tag{67}$$

In the $|\Delta B| = 2$ theory there two physical operators which we choose as

$$
\begin{aligned}
Q &= \bar{s}_i \gamma^\mu \left(1 - \gamma^5\right) b_i \; \bar{s}_j \gamma_\mu \left(1 - \gamma^5\right) b_j \,, \\
\widetilde{Q}_S &= \bar{s}_i \left(1 + \gamma^5\right) b_j \; \bar{s}_j \left(1 + \gamma^5\right) b_i \,,
\end{aligned}
\tag{68}
$$

where i, j are the colour indices attached to quark fields.

References

[1] A. Lenz and U. Nierste, *Theoretical update of $B_s - \bar{B}_s$ mixing*, *JHEP* **06** (2007) 072 [`hep-ph/0612167`].

[2] A. Buras, *Gauge Theory of Weak Decays*. Cambridge University Press, 6, 2020, 10.1017/9781139524100.

[3] R. J. Dowdall, C. T. H. Davies, R. R. Horgan, G. P. Lepage, C. J. Monahan, J. Shigemitsu et al., *Neutral B-meson mixing from full lattice QCD at the physical point*, *Phys. Rev. D* **100** (2019) 094508 [`1907.01025`].

[4] M. Kirk, A. Lenz and T. Rauh, *Dimension-six matrix elements for meson mixing and lifetimes from sum rules*, *JHEP* **12** (2017) 068 [`1711.02100`].

[5] D. King, A. Lenz and T. Rauh, *SU(3) breaking effects in B and D meson lifetimes*, *JHEP* **06** (2022) 134 [`2112.03691`].

[6] M. Beneke, G. Buchalla, C. Greub, A. Lenz and U. Nierste, *Next-to-leading order QCD corrections to the lifetime difference of B(s) mesons*, *Phys. Lett. B* **459** (1999) 631 [`hep-ph/9808385`].

[7] M. Ciuchini, E. Franco, V. Lubicz, F. Mescia and C. Tarantino, *Lifetime differences and CP violation parameters of neutral B mesons at the next-to-leading order in QCD*, *JHEP* **08** (2003) 031 [`hep-ph/0308029`].

[8] M. Beneke, G. Buchalla, A. Lenz and U. Nierste, *CP asymmetry in flavor specific B decays beyond leading logarithms*, *Phys. Lett. B* **576** (2003) 173 [`hep-ph/0307344`].

[9] M. Gerlach, U. Nierste, V. Shtabovenko and M. Steinhauser, *Two-loop QCD penguin contribution to the width difference in $B_s - \overline{B}_s$ mixing*, *JHEP* **07** (2021) 043 [`2106.05979`].

[10] M. Gerlach, U. Nierste, V. Shtabovenko and M. Steinhauser, *The width difference in $B - \bar{B}$ mixing at order α_s and beyond*, *JHEP* **04** (2022) 006 [`2202.12305`].

[11] H. M. Asatrian, A. Hovhannisyan, U. Nierste and A. Yeghiazaryan, *Towards next-to-next-to-leading-log accuracy for the width difference in the $B_s - \bar{B}_s$ system: fermionic contributions to order $(m_c/m_b)^0$ and $(m_c/m_b)^1$*, *JHEP* **10** (2017) 191 [`1709.02160`].

[12] H. M. Asatrian, H. H. Asatryan, A. Hovhannisyan, U. Nierste, S. Tumasyan and A. Yeghiazaryan, *Penguin contribution to the width difference and CP asymmetry in B_q-$\bar{B}_q$ mixing at order $\alpha_s^2 N_f$*, Phys. Rev. D **102** (2020) 033007 [`2006.13227`].

[13] A. Hovhannisyan and U. Nierste, *Addendum to: Towards next-to-next-to-leading-log accuracy for the width difference in the $B_s - \overline{B}_s$ system: fermionic contributions to order $(m_c/m_b)^0$ and $(m_c/m_b)^1$*, JHEP **06** (2022) 090 [`2204.11907`].

[14] M. Gerlach, U. Nierste, V. Shtabovenko and M. Steinhauser, *Width Difference in the B-B$^-$ System at Next-to-Next-to-Leading Order of QCD*, Phys. Rev. Lett. **129** (2022) 102001 [`2205.07907`].

[15] P. Nogueira, *Automatic Feynman Graph Generation*, J. Comput. Phys. **105** (1993) 279.

[16] P. Maierhöfer, J. Usovitsch and P. Uwer, *Kira—A Feynman integral reduction program*, Comput. Phys. Commun. **230** (2018) 99 [`1705.05610`].

[17] P. Maierhöfer and J. Usovitsch, *Kira 1.2 Release Notes*, `1812.01491`.

[18] J. Klappert, F. Lange, P. Maierhöfer and J. Usovitsch, *Integral reduction with Kira 2.0 and finite field methods*, Comput. Phys. Commun. **266** (2021) 108024 [`2008.06494`].

[19] M. Gerlach, U. Nierste, P. Reeck, V. Shtabovenko and M. Steinhauser. (in preparation).

[20] K. G. Chetyrkin, M. Misiak and M. Munz, *$|\Delta F| = 1$ nonleptonic effective Hamiltonian in a simpler scheme*, Nucl. Phys. B **520** (1998) 279 [`hep-ph/9711280`].

[21] T. Peraro and L. Tancredi, *Tensor decomposition for bosonic and fermionic scattering amplitudes*, Phys. Rev. D **103** (2021) 054042 [`2012.00820`].

[22] L. Tancredi, *Tensor decomposition for multiloop, multileg helicity amplitudes*, PoS **LL2022** (2022) 020.

[23] J. Kuipers, T. Ueda, J. A. M. Vermaseren and J. Vollinga, *FORM version 4.0*, Comput. Phys. Commun. **184** (2013) 1453 [`1203.6543`].

[24] A. Pak, *The Toolbox of modern multi-loop calculations: novel analytic and semi-analytic techniques*, J. Phys. Conf. Ser. **368** (2012) 012049 [`1111.0868`].

[25] R. Mertig, M. Bohm and A. Denner, *FEYN CALC: Computer algebraic calculation of Feynman amplitudes*, Comput. Phys. Commun. **64** (1991) 345.

[26] V. Shtabovenko, R. Mertig and F. Orellana, *New Developments in FeynCalc 9.0*, Comput. Phys. Commun. **207** (2016) 432 [`1601.01167`].

[27] V. Shtabovenko, R. Mertig and F. Orellana, *FeynCalc 9.3: New features and improvements*, *Comput. Phys. Commun.* **256** (2020) 107478 [2001.04407].

[28] V. Shtabovenko, R. Mertig and F. Orellana, *FeynCalc 10: Do multiloop integrals dream of computer codes?*, 2312.14089.

[29] V. Shtabovenko, *FeynHelpers: Connecting FeynCalc to FIRE and Package-X*, *Comput. Phys. Commun.* **218** (2017) 48 [1611.06793].

[30] R. Lewis, *FERMAT*, *https://home.bway.net/lewis* .

[31] V. A. Smirnov, *Analytic tools for Feynman integrals*, vol. 250. 2012, 10.1007/978-3-642-34886-0.

[32] J. Fleischer and M. Y. Kalmykov, *ON-SHELL2: FORM based package for the calculation of two loop selfenergy single scale Feynman diagrams occurring in the standard model*, *Comput. Phys. Commun.* **128** (2000) 531 [hep-ph/9907431].

[33] E. Panzer, *Algorithms for the symbolic integration of hyperlogarithms with applications to Feynman integrals*, *Comput. Phys. Commun.* **188** (2015) 148 [1403.3385].

[34] H. Cheng and T. T. Wu, *EXPANDING PROTONS: SCATTERING AT HIGH-ENERGIES*. The MIT Press, Cambridge., 1987.

[35] E. Panzer, *On hyperlogarithms and Feynman integrals with divergences and many scales*, *JHEP* **03** (2014) 071 [1401.4361].

[36] E. Panzer, *Feynman integrals and hyperlogarithms*, Ph.D. thesis, Humboldt U., 2015. 1506.07243. 10.18452/17157.

[37] A. von Manteuffel, E. Panzer and R. M. Schabinger, *A quasi-finite basis for multi-loop Feynman integrals*, *JHEP* **02** (2015) 120 [1411.7392].

[38] S. Borowka, G. Heinrich, S. Jahn, S. P. Jones, M. Kerner, J. Schlenk et al., *pySecDec: a toolbox for the numerical evaluation of multi-scale integrals*, *Comput. Phys. Commun.* **222** (2018) 313 [1703.09692].

[39] S. Borowka, G. Heinrich, S. Jahn, S. P. Jones, M. Kerner and J. Schlenk, *A GPU compatible quasi-Monte Carlo integrator interfaced to pySecDec*, *Comput. Phys. Commun.* **240** (2019) 120 [1811.11720].

[40] G. Heinrich, S. Jahn, S. P. Jones, M. Kerner, F. Langer, V. Magerya et al., *Expansion by regions with pySecDec*, *Comput. Phys. Commun.* **273** (2022) 108267 [2108.10807].

[41] G. Heinrich, S. P. Jones, M. Kerner, V. Magerya, A. Olsson and J. Schlenk, *Numerical scattering amplitudes with pySecDec*, *Comput. Phys. Commun.* **295** (2024) 108956 [`2305.19768`].

[42] A. V. Smirnov, *FIESTA4: Optimized Feynman integral calculations with GPU support*, *Comput. Phys. Commun.* **204** (2016) 189 [`1511.03614`].

[43] A. V. Smirnov, N. D. Shapurov and L. I. Vysotsky, *FIESTA5: Numerical high-performance Feynman integral evaluation*, *Comput. Phys. Commun.* **277** (2022) 108386 [`2110.11660`].

[44] A. B. Goncharov, *Multiple polylogarithms, cyclotomy and modular complexes*, *Math. Res. Lett.* **5** (1998) 497 [`1105.2076`].

[45] O. Schnetz, *HyperLogProcedures*, *https://www.math.fau.de/person/oliver-schnetz* .

[46] D. Broadhurst, *Multiple Deligne values: a data mine with empirically tamed denominators*, `1409.7204`.

[47] C. Duhr and F. Dulat, *PolyLogTools — polylogs for the masses*, *JHEP* **08** (2019) 135 [`1904.07279`].

[48] M. Beneke and V. A. Smirnov, *Asymptotic expansion of Feynman integrals near threshold*, *Nucl. Phys. B* **522** (1998) 321 [`hep-ph/9711391`].

[49] A. Pak and A. Smirnov, *Geometric approach to asymptotic expansion of Feynman integrals*, *Eur. Phys. J. C* **71** (2011) 1626 [`1011.4863`].

[50] B. Jantzen, A. V. Smirnov and V. A. Smirnov, *Expansion by regions: revealing potential and Glauber regions automatically*, *Eur. Phys. J. C* **72** (2012) 2139 [`1206.0546`].

[51] C. W. Bauer, S. Fleming, D. Pirjol and I. W. Stewart, *An Effective field theory for collinear and soft gluons: Heavy to light decays*, *Phys. Rev. D* **63** (2001) 114020 [`hep-ph/0011336`].

[52] C. W. Bauer, D. Pirjol and I. W. Stewart, *Soft collinear factorization in effective field theory*, *Phys. Rev. D* **65** (2002) 054022 [`hep-ph/0109045`].

[53] M. Beneke, A. P. Chapovsky, M. Diehl and T. Feldmann, *Soft collinear effective theory and heavy to light currents beyond leading power*, *Nucl. Phys. B* **643** (2002) 431 [`hep-ph/0206152`].

[54] M. Beneke and T. Feldmann, *Multipole expanded soft collinear effective theory with nonAbelian gauge symmetry*, *Phys. Lett. B* **553** (2003) 267 [`hep-ph/0211358`].

[55] A. V. Smirnov, *FIRE5: a C++ implementation of Feynman Integral REduction*, *Comput. Phys. Commun.* **189** (2015) 182 [`1408.2372`].

[56] A. V. Smirnov and F. S. Chuharev, *FIRE6: Feynman Integral REduction with Modular Arithmetic*, *Comput. Phys. Commun.* **247** (2020) 106877 [`1901.07808`].

[57] A. V. Smirnov and M. Zeng, *FIRE 6.5: Feynman Integral Reduction with New Simplification Library*, `2311.02370`.

[58] M. Fael, F. Lange, K. Schönwald and M. Steinhauser, *A semi-analytic method to compute Feynman integrals applied to four-loop corrections to the $\overline{\mathrm{MS}}$-pole quark mass relation*, *JHEP* **09** (2021) 152 [`2106.05296`].

[59] M. Fael, F. Lange, K. Schönwald and M. Steinhauser, *Massive Vector Form Factors to Three Loops*, *Phys. Rev. Lett.* **128** (2022) 172003 [`2202.05276`].

[60] M. Fael, F. Lange, K. Schönwald and M. Steinhauser, *Singlet and nonsinglet three-loop massive form factors*, *Phys. Rev. D* **106** (2022) 034029 [`2207.00027`].

[61] M. Fael, F. Lange, K. Schönwald and M. Steinhauser, *Massive three-loop form factors: Anomaly contribution*, *Phys. Rev. D* **107** (2023) 094017 [`2302.00693`].

[62] A. I. Davydychev and V. A. Smirnov, *Threshold expansion of the sunset diagram*, *Nucl. Phys. B* **554** (1999) 391 [`hep-ph/9903328`].

[63] K. Melnikov. Private communication.

[64] M. Egner, M. Fael, K. Schönwald and M. Steinhauser, *Revisiting semileptonic B meson decays at next-to-next-to-leading order*, *JHEP* **09** (2023) 112 [`2308.01346`].

[65] X. Liu and Y.-Q. Ma, *AMFlow: A Mathematica package for Feynman integrals computation via auxiliary mass flow*, *Comput. Phys. Commun.* **283** (2023) 108565 [`2201.11669`].

[66] `https://www.ttp.kit.edu/preprints/2024/ttp24-013/`.

[67] K. G. Chetyrkin and M. F. Zoller, *Four-loop renormalization of QCD with a reducible fermion representation of the gauge group: anomalous dimensions and renormalization constants*, *JHEP* **06** (2017) 074 [`1704.04209`].