Journal of Physics A: Mathematical and Theoretical

PAPER • **OPEN ACCESS**

Basis decompositions and a Mathematica package for modular graph forms

To cite this article: Jan E Gerken 2021 *J. Phys. A: Math. Theor.* **54** 195401

OPEN ACCESS
IOP Publishing
Journal of Physics A: Mathematical and Theoretical
J. Phys. A: Math. Theor. **54** (2021) 195401 (92pp)
https://doi.org/10.1088/1751-8121/abbdf2

Basis decompositions and a Mathematica package for modular graph forms

Jan E Gerken*

Max-Planck-Institut für Gravitationsphysik, Albert-Einstein-Institut, DE-14476 Potsdam, Germany

E-mail: jan.gerken@aei.mpg.de

Received 21 August 2020
Accepted for publication 2 October 2020
Published 28 April 2021

Abstract

Modular graph forms (MGFs) are a class of non-holomorphic modular forms which naturally appear in the low-energy expansion of closed-string genus-one amplitudes and have generated considerable interest from pure mathematicians. MGFs satisfy numerous non-trivial algebraic- and differential relations which have been studied extensively in the literature and lead to significant simplifications. In this paper, we systematically combine these relations to obtain basis decompositions of all two- and three-point MGFs of total modular weight $w + \bar{w} \leqslant 12$, starting from just two well-known identities for banana graphs. Furthermore, we study previously known relations in the integral representation of MGFs, leading to a new understanding of holomorphic subgraph reduction as Fay identities of Kronecker–Eisenstein series and opening the door toward decomposing divergent graphs. We provide a computer implementation for the manipulation of MGFs in the form of the `Mathematica` package `ModularGraphForms` which includes the basis decompositions obtained.

Keywords: string perturbation theory, genus-one string amplitude, closed-string amplitude, low-energy expansion, modular form

(Some figures may appear in colour only in the online journal)

1. Introduction

Scattering amplitudes in string theory have in recent years experienced a rise in interest due to the rich mathematical structures appearing in their calculation and their close relations to field-

*Author to whom any correspondence should be addressed.

theory amplitudes. Tree-level string amplitudes at genus zero are by now well under control and many powerful results have been obtained for genus-one amplitudes as well.

Closed-string genus-one amplitudes are given by integrals of correlators in the worldsheet conformal field theory (CFT) over the moduli space of punctured tori. In this paper, we systematically study relations between modular graph forms (MGFs), a class of functions of the modular parameter $\tau = \tau_1 + \mathrm{i}\tau_2$, $\tau_1, \tau_2 \in \mathbb{R}$ with $\tau_2 > 0$ which make the computation of the low-energy expansion of the integrals over the punctures algorithmic and have been studied widely in the literature [1–31]. Genus-two generalizations of these techniques were studied in [32–36]. The low-energy expansion is an expansion in powers of the inverse string tension α' or, equivalently, in the Mandelstam variables

$$s_{ij} = -\frac{\alpha'}{2} k_i \cdot k_j, \tag{1.1}$$

where the k_i are the momenta of the asymptotic string states. The resulting MGFs $\mathcal{C}_\Gamma(\tau)$, introduced in [8], are non-holomorphic modular forms, i.e. they transform as

$$\mathcal{C}_\Gamma\left(\frac{\alpha\tau+\beta}{\gamma\tau+\delta}\right) = (\gamma\tau+\delta)^a(\gamma\bar{\tau}+\delta)^b \mathcal{C}_\Gamma(\tau), \quad \begin{pmatrix}\alpha & \beta\\ \gamma & \delta\end{pmatrix} \in \mathrm{SL}(2,\mathbb{Z}), \tag{1.2}$$

where a is the holomorphic modular weight and b is the antiholomorphic modular weight and the MGFs are labeled by Feynman-like decorated graphs Γ. Similarly to how Feynman diagrams can be translated into nested integrals over loop momenta, we can read off the representation of MGFs in terms of nested lattice sums from the graph Γ (the momenta are discrete since the torus is compact). A second representation of MGFs is in terms of torus integrals of Jacobi forms, corresponding to the position-space representation of Γ.

MGFs satisfy many non-trivial relations which are hard to see in the lattice sum- (or integral-) representation, e.g. the MGF that was denoted by $C_{1,1,1}$ in [37] satisfies the relation

$$\begin{aligned} C_{1,1,1} &= \left(\frac{\tau_2}{\pi}\right)^3 {\sum_{\substack{(m_1,n_1)\in\mathbb{Z}^2\\(m_2,n_2)\in\mathbb{Z}^2}}}' \frac{1}{|m_1\tau+n_1|^2|m_2\tau+n_2|^2|(m_1+m_2)\tau+n_1+n_2|^2} \\ &= \left(\frac{\tau_2}{\pi}\right)^3 {\sum_{(m,n)\in\mathbb{Z}^2}}' \frac{1}{|m\tau+n|^6} + \sum_{r=1}^{\infty}\frac{1}{r^3} \\ &= \mathrm{E}_3 + \zeta_3, \end{aligned} \tag{1.3}$$

where the prime on the sums indicates that we omit the origin from the summation domains $\mathbb{Z}^2$ and the non-holomorphic Eisenstein series E_s are defined in (2.32). We use the notation

$$\zeta_k = \sum_{n=1}^{\infty}\frac{1}{n^k}, \quad k\in\mathbb{N},\ k\geqslant 2, \tag{1.4}$$

for zeta values. Identities of this type were studied extensively in the literature [3, 4, 8–11, 18, 19, 21]. Although MGFs also satisfy many non-trivial differential equations, we will focus here mainly on algebraic relations.

1.1. Summary of results

In this paper, we derive relations between MGFs of the form (1.3) by systematically applying known and new manipulation techniques to a large class of MGFs. In particular, we extend known techniques for two- and three-point graphs to a complete treatment of four-point graphs and show that in the integral representation, the well-known technique of holomorphic subgraph reduction (HSR) [8, 19] is equivalent to Fay identities of the Kronecker–Eisenstein series. This yields a more compact and iterative procedure for performing HSR on higher-point graphs than was previously available in the literature. Furthermore, we give a first systematic discussion of divergent MGFs and show how these can be interpreted as arising form kinematic poles in torus integrals.

By applying these manipulations extensively to all two- and three-point graphs of total modular weight $a + b \leqslant 12$, we find basis decompositions for all these graphs starting from just the two well-known decompositions of the banana graphs $D_3 = C_{1,1,1}$ and D_5, cf (9.1) and (9.2). The structure of the basis (and in particular its dimension) agrees with the predictions made previously in the literature [21, 30]. The arguments in [30] based on iterated Eisenstein integrals show in particular that the basis elements obtained are linearly independent and span the space of MGFs of arbitrary topology (and the corresponding modular weight). Furthermore, since the Laurent polynomials of the basis elements are known [3, 11], this allows us to easily obtain the Laurent polynomials of all the decomposed MGFs. With the help of the Laurent polynomials, we construct the five real cusp forms in the space of MGFs at weight $(6, 6)$, cf (9.8) and show that no real cusp forms exist at lower weights.

Using the basis decompositions of MGFs, we expand the generating series $Y^{\tau}_{\vec{\eta}}$ of Koba–Nielsen integrals defined in [30] in the basis-MGFs for two and three points up to order 12, corresponding to MGFs of total modular weight at most 12. These expansions were crucial in determining the dictionary between MGFs and iterated Eisenstein integrals in [30] and are made available in an ancillary file to the arXiv submission of this paper.

Finally, we provide the `Mathematica` package `ModularGraphForms` in the ancillary files of the arXiv submission which automatizes the manipulations discussed in this paper and contains all basis decompositions for two and three-point MGFs of weight $a + b \leqslant 12$. Furthermore, the `ModularGraphForms` package can be used to automatically expand Koba–Nielsen integrals in terms of MGFs. This package was used to obtain the expansions of the $Y^{\tau}_{\vec{\eta}}$ mentioned above.

1.2. Outline

This paper is structured as follows: in section 2, we review the definition of MGFs and their different representations as well as some other important objects. In section 3, we begin the main body of the paper with a brief overview of the `ModularGraphForms` package, followed in section 4 by an introduction of the notation for MGFs which we will be using throughout for two-, three- and four-point graphs. For these graphs, we collect a number of simple manipulations in section 5 which are largely known in the literature. In section 6 we discuss HSR and how it is related to the Fay identity of the Kronecker–Eisenstein series. In section 7, we review the sieve algorithm [8] and discuss its implementation in the `ModularGraphForms` package. Since using the relations discussed in section 5 can lead to divergent MGFs, even if we start out with only convergent graphs, we discuss divergent MGFs in section 8. All the manipulations discussed up to this point are combined in section 9 to obtain basis decompositions for a large class of MGFs. Section 10 contains a conclusion and outlook. In appendix A we give a complete reference to the `ModularGraphForms` package and in appendix B, we discuss further details about kinematic poles of three-point integrals.

2. Modular graph forms

In this section, we give a brief review of the structures appearing in the evaluation of genus-one closed-string integrals and introduce MGFs.

2.1. Koba–Nielsen integrals and Kronecker–Eisenstein series

After evaluating the CFT correlator of the vertex operators, closed string genus-one amplitudes can be written in terms of integrals of the form

$$\int_{\Sigma_\tau^{n-1}} \mathrm{d}\mu_{n-1}\, \phi^{(a,b)}(\vec{z},\vec{\bar{z}},\tau)\mathrm{KN}_n(\vec{z},\tau), \tag{2.1}$$

where $\Sigma_\tau \subset \mathbb{C}$ is the torus with modular parameter $\tau = \tau_1 + \mathrm{i}\tau_2$, $\tau_1, \tau_2 \in \mathbb{R}$, parametrized by the parallelogram spanned by the paths $(0,1)$ and $(0,\tau)$ with opposite edges identified. We integrate over the puncture positions z_i (collectively denoted by $\vec{z}$) using the modular invariant integration measure

$$\mathrm{d}\mu_{n-1} = \prod_{k=2}^{n} \frac{\mathrm{d}\,\mathrm{Re}(z_k) \wedge \mathrm{d}\,\mathrm{Im}(z_k)}{\tau_2} = \prod_{k=2}^{n} \mathrm{d}v_k \wedge \mathrm{d}u_k, \tag{2.2}$$

where we have fixed the origin of the coordinate system to $z_1 = 0$. In (2.2), we also gave the integration measure in terms of the coordinates u and v, which are aligned with the axes of the parallelogram,

$$u = \frac{\mathrm{Im}(z)}{\tau_2}, \qquad v = \mathrm{Re}(z) - \frac{\tau_1}{\tau_2}\,\mathrm{Im}(z) \Rightarrow z = u\tau + v. \tag{2.3}$$

The function $\phi^{(a,b)}(\vec{z},\vec{\bar{z}},\tau)$ in the integrand of (2.1) depends on the positions z_i, their complex conjugates $\bar{z}_i$ and the modular parameter τ and transforms as a non-holomorphic Jacobi form of weight (a,b) (and vanishing index), i.e.

$$\phi^{(a,b)}\left(\frac{\vec{z}}{\gamma\tau+\delta}, \frac{\vec{\bar{z}}}{\gamma\bar{\tau}+\delta}, \frac{\alpha\tau+\beta}{\gamma\tau+\delta}\right) = (\gamma\tau+\delta)^a(\gamma\bar{\tau}+\delta)^b\phi^{(a,b)}(\vec{z},\vec{\bar{z}},\tau), \tag{2.4}$$

where $\begin{pmatrix}\alpha & \beta\\ \gamma & \delta\end{pmatrix} \in \mathrm{SL}(2,\mathbb{Z})$. We will give more details on the form of ϕ shortly. The *Koba–Nielsen factor* $\mathrm{KN}_n(\tau)$ in (2.1) is defined by (we will from now on drop the explicit dependence on $\vec{z}$ and τ) [38]

$$\mathrm{KN}_n = \exp\left(\sum_{1\leqslant i<j}^{n} s_{ij}G_{ij}\right) \tag{2.5}$$

in terms of the Mandelstam invariants (1.1) and the Green function $G_{ij} = G(z_{ij},\tau) = G(z_i - z_j,\tau)$ on the torus which satisfies

$$\partial_z\partial_{\bar{z}}G(z,\tau) = -\pi\delta^{(2)}(z,\bar{z}) + \frac{\pi}{\tau_2}. \tag{2.6}$$

The Green function is doubly periodic in z and can hence be written as a double Fourier-series in this variable. In this representation, it is given by [1]

$$G(z,\tau) = \frac{\tau_2}{\pi} \sum_{(m,n)\in\mathbb{Z}^2}{}' \frac{e^{2\pi i(mv-nu)}}{|m\tau+n|^2} = \frac{\tau_2}{\pi}\sum_p{}' \frac{e^{2\pi i\langle p,z\rangle}}{|p|^2}, \tag{2.7}$$

where the prime again indicates that the origin is omitted from the sum and we used the notation

$$p = m\tau + n \quad \langle p,z\rangle = mv - nu = \frac{(p\bar{z}-\bar{p}z)}{2i\tau_2}. \tag{2.8}$$

In the representation (2.7) it is manifest that the Green function is modular invariant and hence the integral in (2.1) transforms as a non-holomorphic modular form of weight (a,b).

In order to describe the structure of $\phi^{(a,b)}$ in more detail, consider the *Kronecker–Eisenstein series* [39, 40]

$$\Omega(z,\eta,\tau) = \exp\left(2\pi i\eta\frac{\mathrm{Im}\ z}{\tau_2}\right)\frac{\theta_1'(0,\tau)\theta_1(z+\eta,\tau)}{\theta_1(z,\tau)\theta_1(\eta,\tau)}, \tag{2.9}$$

where $\theta_1(z,\tau)$ is the first Jacobi theta function. Ω is doubly periodic in z and can therefore be written as a Fourier series,

$$\Omega(z,\eta,\tau) = \sum_p \frac{e^{2\pi i\langle p,z\rangle}}{p+\eta}, \tag{2.10}$$

where we used the notation (2.8). By expanding Ω in η,

$$\Omega(z,\eta,\tau) = \sum_{a\geqslant 0}\eta^{a-1}f^{(a)}(z,\tau), \tag{2.11}$$

we define the functions $f^{(a)}(z,\tau)$ which have Fourier expansion

$$f^{(0)}(z,\tau) = 1 \tag{2.12a}$$

$$f^{(a)}(z,\tau) = (-1)^{a-1}\sum_p{}' \frac{2\pi i\langle p,z\rangle}{p^a}, \quad a>0 \tag{2.12b}$$

$$\overline{f^{(b)}(z,\tau)} = -\sum_p{}' \frac{2\pi i\langle p,z\rangle}{\bar{p}^b}, \quad b>0. \tag{2.12c}$$

Note that for the Fourier series of $f^{(a)}$ with $a\leqslant 2$ are not absolutely convergent. From this representation, it is easy to check that the $f^{(a)}$ satisfy the differential equations

$$\partial_z f^{(a)}(z,\tau) = -\frac{\pi}{\tau_2}f^{(a-1)}(z,\tau) + \pi\delta_{a,1}\delta^{(2)}(z,\bar{z}), \quad a\geqslant 1. \tag{2.13}$$

This implies

$$\partial_z G(z,\tau) = -f^{(1)}(z,\tau), \tag{2.14}$$

upon comparing (2.13) to (2.6). The Fourier representation (2.12) of the $f^{(a)}(z,\tau)$ also manifests that they transform as Jacobi forms of weight $(a,0)$ (and vanishing index),

$$f^{(a)}\left(\frac{z}{\gamma\tau+\delta},\frac{\alpha\tau+\beta}{\gamma\tau+\delta}\right)=(\gamma\tau+\delta)^a f^{(a)}(z,\tau),\qquad \begin{pmatrix}\alpha & \beta\\ \gamma & \delta\end{pmatrix}\in \mathrm{SL}(2,\mathbb{Z}). \tag{2.15}$$

The function $\phi^{(a,b)}(\vec{z},\vec{\bar{z}},\tau)$ appearing in the integral (2.1) can be written as a homogeneous polynomial in the $f^{(a)}$ and $\overline{f^{(b)}}$ evaluated at differences of the z_i for any massless amplitude of closed-string states in type-II, heterotic or bosonic theories [20, 41, 42]. For these differences, we introduce the notation $f^{(a)}_{ij}=f^{(a)}(z_{ij},\tau)=f^{(a)}(z_i-z_j,\tau)$ and similarly for $\overline{f^{(b)}}$. We will refer to an integral of the form (2.1) with ϕ a polynomial in $f^{(a)}_{ij}$ and $\overline{f^{(b)}_{k\ell}}$ as a *Koba–Nielsen integral*.

An important class of polynomials in the $f^{(a)}_{ij}$ which appears e.g. in the computation of four-gluon scattering in the heterotic string [19] is given by the V_a functions defined by

$$\Omega(z_{12},\eta,\tau)\Omega(z_{23},\eta,\tau)\cdots\Omega(z_{n-1,n},\eta,\tau)\Omega(z_{n,1},\eta,\tau)=\eta^{-n}\sum_{a=0}^{\infty}\eta^a V_a(1,2,\ldots,n), \tag{2.16}$$

where the labels of V_a refer to the order of the punctures in the product of Kronecker–Eisenstein series. Using the expansion (2.11), the V_a can be written in terms of the $f^{(a)}_{ij}$, e.g.

$$\begin{aligned}V_0(1,2,\ldots,n)&=1\,,\qquad V_1(1,2,\ldots,n)=\sum_{j=1}^{n}f^{(1)}_{j,j+1}\\ V_2(1,2,\ldots,n)&=\sum_{j=1}^{n}f^{(2)}_{j,j+1}+\sum_{i=1}^{n}\sum_{j=i+1}^{n}f^{(1)}_{i,i+1}f^{(1)}_{j,j+1}\ \text{etc},\end{aligned} \tag{2.17}$$

where we set $f^{(a)}_{n,n+1}=f^{(a)}_{n,1}$.

2.2. Modular graph forms

MGFs are the expansion coefficients in the Mandelstam expansion of (2.1). In order to define MGFs, consider a generalization of the sums in (2.12) and (2.7),

$$C^{(a,b)}(z,\tau)=\sum_{p}{}'\frac{e^{2\pi i\langle p,z\rangle}}{p^a\bar{p}^b}. \tag{2.18}$$

These functions were previously studied in [6, 43, 44]. The Green function and the $f^{(a)}$ and $\overline{f^{(b)}}$ are special cases of the $C^{(a,b)}$ since

$$\begin{aligned}G(z,\tau)&=\frac{\tau_2}{\pi}C^{(1,1)}(z,\tau)\\ f^{(a)}(z,\tau)&=(-1)^{a-1}C^{(a,0)}(z,\tau)\quad a>0\\ \overline{f^{(b)}(z,\tau)}&=-C^{(0,b)}(z,\tau)\qquad b>0\end{aligned} \tag{2.19}$$

Using (2.18), the expansion coefficient of (2.1) (for one monomial in $\phi^{(a,b)}$) at a certain order in α' has the form

$$\mathcal{C}_\Gamma(\tau) = \int \mathrm{d}\mu_{n-1} \prod_{e\in E_\Gamma} C^{(a_e,b_e)}(z_e,\tau), \tag{2.20}$$

which is the integral representation of the MGF $\mathcal{C}_\Gamma$. The notation in (2.20) is suggestive of the graphical representation of MGFs [8]: we can associate an n-vertex graph Γ to the integral in (2.20) by identifying the $C_{ij}^{(a,b)} = C^{(a,b)}(z_{ij})$ with an edge form vertex i to vertex j, labeled by (a,b),

$$C_{ij}^{(a,b)} \quad \leftrightarrow \quad \overset{i}{\bullet}\!\!\longrightarrow\!\!\overset{j}{\bullet} \quad (a,b) \quad . \tag{2.21}$$

In this notation, E_Γ in (2.20) is the edge set of the graph. Using the notation

$$|A| = \sum_{e\in E_\Gamma} a_e \qquad |B| = \sum_{e\in E_\Gamma} b_e, \tag{2.22}$$

the MGF in (2.20) is a non-holomorphic modular form of weight $(|A|,|B|)$. If the holomorphic and antiholomorphic edge labels are equal, $a_\mathrm{e} = \mathrm{b_e}\ \forall\, \mathrm{e} \in \mathrm{E}_\Gamma$, the MGF can be turned into a modular function by multiplication with $\tau_2^{|A|}$. In this case, we call the MGF a *modular graph function* [6]. Note that also the weaker condition $|A| = |B|$ is used to define modular graph functions in the literature.

Since the integrand in (2.20) depends on z only through the exponential factors $\mathrm{e}^{2\pi\mathrm{i}\langle p,z\rangle}$, we can perform this integral trivially, leading to conservation of the momenta p at the vertices. This leads to the sum representation [8]

$$\mathcal{C}_\Gamma(\tau) = {\sum_{\{p_e\}}}' \prod_{e\in E_\Gamma} \frac{1}{p_e^{a_e}\bar{p}_e^{b_e}} \prod_{i\in V_\Gamma} \delta\left(\sum_{e'\in E_\Gamma} \Gamma_{ie'} p_{e'}\right) \tag{2.23}$$

of MGFs, where E_Γ is the set of edges of Γ, V_Γ is the set of vertices and

$$\Gamma_{ie} = \begin{cases} 1 & \text{if } e \text{ is directed into } i \\ -1 & \text{if } e \text{ is directed out of } i \\ 0 & \text{if } e \text{ is not connected to } i \end{cases} \tag{2.24}$$

is the incidence matrix of vertex i.

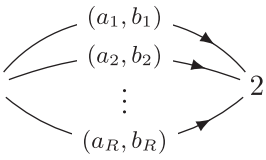
A simple example of a modular graph function is given by a two-point graph with ℓ edges with label $(1,1)$ each between the vertices. This MGF is denoted by D_ℓ and given by [3]

$$D_\ell(\tau) = \left(\frac{\tau_2}{\pi}\right)^\ell {\sum_{p_1,\ldots,p_\ell}}' \frac{\delta(p_1+\cdots+p_\ell)}{|p_1|^2\cdots|p_\ell|^2} \tag{2.25}$$

in the sum representation. Here, the sum was multiplied by a suitable factor of $\frac{\tau_2}{\pi}$ to make D_ℓ modular invariant, as is customary in the literature.

If we assign arbitrary labels to the edges, the resulting MGF is called *dihedral* and given by [8]

$$\mathcal{C}\left[\begin{smallmatrix} a_1 & \cdots & a_R \\ b_1 & \cdots & b_R \end{smallmatrix}\right] = \int \mathrm{d}\mu_1 \prod_{i=1}^{R} C_{12}^{(a_i,b_i)} = \quad (2.26)$$



The D_ℓ from (2.25) are in this notation given by

$$D_\ell = \left(\frac{\tau_2}{\pi}\right)^\ell \mathcal{C}\begin{bmatrix} 1_\ell \\ 1_\ell \end{bmatrix}, \quad (2.27)$$

where 1_ℓ denotes the row vector with ℓ entries of 1. Further special cases of (2.26) are the modular graph functions [3]

$$C_{a,b,c} = \left(\frac{\tau_2}{\pi}\right)^{a+b+c} \mathcal{C}\left[\begin{smallmatrix} a & b & c \\ a & b & c \end{smallmatrix}\right] \quad (2.28)$$

$$C_{a,b,c,d} = \left(\frac{\tau_2}{\pi}\right)^{a+b+c+d} \mathcal{C}\left[\begin{smallmatrix} a & b & c & d \\ a & b & c & d \end{smallmatrix}\right]. \quad (2.29)$$

To write one-loop graphs in the notation (2.26), we need a $\begin{bmatrix} 0 \\ 0 \end{bmatrix}$-column since otherwise the omission of the origin in the sum sets the MGF to zero. Consequently, we have

$$\mathcal{C}\begin{bmatrix} a & 0 \\ b & 0 \end{bmatrix} = \sum_p{}' \frac{1}{p^a \bar{p}^b} \quad (2.30a)$$

$$\mathcal{C}\begin{bmatrix} k & 0 \\ 0 & 0 \end{bmatrix} = \mathrm{G}_k, \quad k > 2 \quad (2.30b)$$

$$\mathcal{C}\begin{bmatrix} s & 0 \\ s & 0 \end{bmatrix} = \left(\frac{\pi}{\tau_2}\right)^s \mathrm{E}_s, \qquad \mathrm{Re}(s) > 1. \quad (2.30c)$$

Here, we have introduced the *holomorphic Eisenstein series* G_k and their non-holomorphic counterparts E_s which are defined by

$$\mathrm{G}_k(\tau) = \sum_{(m,n)\in\mathbb{Z}^2}{}' \frac{1}{(m\tau+n)^k}, \quad k \geqslant 3 \in \mathbb{N} \quad (2.31)$$

$$\mathrm{E}_s(\tau) = \left(\frac{\tau_2}{\pi}\right)^s \sum_{(m,n)\in\mathbb{Z}^2}{}' \frac{1}{|m\tau+n|^{2s}}, \quad s \in \mathbb{C},\ \mathrm{Re}(s) > 1. \quad (2.32)$$

We will also use the modular, but non-holomorphic version $\widehat{\mathrm{G}}_2$ of the Eisenstein series G_2, defined by

$$\widehat{\mathrm{G}}_2(\tau) = \lim_{s\to 0} \sum_{(m,n)\in\mathbb{Z}^2}{}' \frac{1}{(m\tau+n)^2|m\tau+n|^s}, \quad (2.33)$$

which can be written as

$$\widehat{\mathrm{G}}_2(\tau) = \mathrm{G}_2(\tau) - \frac{\pi}{\tau_2} \tag{2.34}$$

with

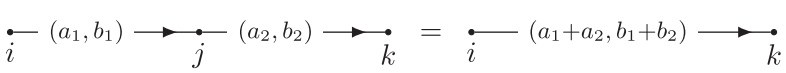
$$\mathrm{G}_2(\tau) = \sum_{n\neq 0}\frac{1}{n^2} + \sum_{m\neq 0}\sum_{n\in\mathbb{Z}}\frac{1}{(m\tau+n)^2}. \tag{2.35}$$

We will assume the regularization (2.33) for conditionally convergent MGFs throughout and therefore have

$$\mathcal{C}\begin{bmatrix}2 & 0\\ 0 & 0\end{bmatrix} = \widehat{\mathrm{G}}_2. \tag{2.36}$$

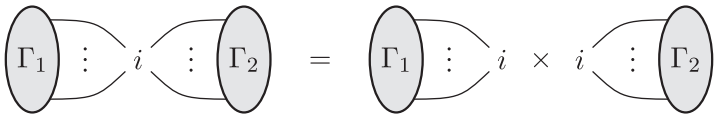
More details on the convergence properties of (2.20) and (2.23) are given in section 8.

The definition (2.23) of MGFs implies a number of properties [3], some of which we want to mention here. Firstly, the modular behavior of the MGFs implies that $\mathcal{C}_\Gamma = 0$ if $|A| + |B|$ is odd. Secondly, the graphs of all non-zero MGFs are one-particle irreducible vacuum bubbles. Furthermore, two-valent vertices can be dropped by adding the labels of their edges,



. (2.37)

Finally, if a graph has connectivity one (i.e. it can be disconnected by removing one vertex), the associated MGF factorizes,



. (2.38)

Using the definitions above, we can now discuss the basics of the `Mathematica` package `ModularGraphForms` in the next section, which can be used to perform numerous manipulations on MGFs.

3. The `ModularGraphForms` Mathematica package

As mentioned in the introduction, we will present a number of simplification techniques for MGFs in this paper, which will allow us to derive basis decompositions for a large number of MGFs, as discussed in section 9. To make the resulting decompositions accessible, it is convenient to have a computer database of them, together with an implementation for the various techniques to be discussed. This is realized in the `Mathematica` package `ModularGraphForms` which is included in the arXiv submission of this paper. It contains 11 079 identities to decompose all two- and three-point MGFs of total modular weight $a + b \leqslant 12$ into the basis given in section 9 and functions for basic manipulations of four-point graphs. The package furthermore contains routines to automatically expand Koba–Nielsen integrals in terms of MGFs. In this section, we will outline the basic usage of the package and, as we discuss the manipulations for MGFs in the following sections, we will also describe how they are implemented in the `ModularGraphForms` package. A complete reference of all defined symbols as well as all functions and their options is given in appendix A.

3.1. Basics

The `ModularGraphForms` package consists of the package itself in the `ModularGraphForms.m` file and two files containing identities between two- and three-point MGFs which were generated using the techniques presented in this paper. To load the package, copy all files into the directory in which the current notebook is saved and run

```
In[1]:= Get[NotebookDirectory[] <> "ModularGraphForms.m"]

        Dihedral identity file found at /home/user/DiIds.txt
        Trihedral identity file found at /home/user/TriIds.txt
        Loaded 1559 identities for dihedral convergent MGF.
        Loaded 9520 identities for trihedral convergent MGF.
        Successfully loaded the ModularGraphForms package. Have fun!
```

The notation used for τ is `tau`, $\bar{\tau}$ is `tauBar` and τ_2 is `tau[2]`. The zeta values (1.4) are written as e.g. `zeta[3]` and the holomorphic Eisenstein series (2.31) and their complex conjugates are `g[4]` and `gBar[4]`, respectively. For the modular version $\widehat{\mathrm{G}}_2$ of G_2, we use `gHat[2]` and `gBarHat[2]`. The non-holomorphic Eisenstein series (2.32) and their higher-depth generalizations defined below in (9.5) are denoted for instance by `e[2, 2]`. The normalizations are as described in section 2.

The modular weight of an expression is determined by the function `CModWeight`, e.g.

```
In[2]:= CModWeight[g[4] + gHat[2]^2 + (tau[2]/π)^4 e[2,2] gBar[4] g[8]]
```

$$\text{Out[2]= } \{4,0\} \ .$$

Complex conjugation is performed by the function `CComplexConj`, e.g.

```
In[3]:= CComplexConj[g[4] + gHat[2]^2 + (tau[2]/π)^4 e[2,2] gBar[4] g[8]]
```

$$\text{Out[3]= } \overline{\mathrm{G}}_4 + \hat{\overline{\mathrm{G}}}_2^2 + \frac{\mathrm{E}_{2,2}\,\mathrm{G}_4\,\overline{\mathrm{G}}_8\,\tau_2^4}{\pi^4} \ .$$

The most important function of the `ModularGraphForms` package is the function `CSimplify`, which performs all known simplifications for MGFs on the expression in the argument, e.g. the identity (1.3) is hard-coded into the package and can be used as follows,

```
In[4]:= CSimplify[c[1 1 1]]
```

$$\text{Out[4]= } \frac{\pi^3\,\mathrm{E}_3}{\tau_2^3} + \frac{\pi^3\,\zeta_3}{\tau_2^3} \ ,$$

where the notation for MGFs will be explained in section 4 below. The function `CSimplify` calls the functions `DiCSimplify`, `TriCSimplify` and `TetCSimplify`, which perform simplifications on two- three- and four-point graphs, respectively.

The remaining functions in the `ModularGraphForms` package will be discussed in the following sections, along with the manipulations for MGFs they implement. A complete reference for all functions and their options is given in appendix A. Within `Mathematica`, short explanations for the various objects can be obtained using the `Information` function, e.g. by running `?CModWeight`. A complete list of all available objects can be printed by running `?ModularGraphForms'*`.

3.2. Expanding Koba–Nielsen integrals

As explained in section 2, in string theory, MGFs arise as coefficients in the expansion of Koba–Nielsen integrals. The `ModularGraphForms` package also contains the function `zIntegrate` which performs this expansion automatically. The syntax is as follows: `zIntegrate` has three arguments, the first one is the prefactor in front of the Koba–Nielsen factor, the second one is the number of points in the Koba–Nielsen factor (2.5) and the last one is the order in Mandelstam variables which is written in terms of MGFs. E.g. the second order in Mandelstams of the three-point integral

$$\int \mathrm{d}\mu_2 \, \mathrm{KN}_3 \tag{3.1}$$

is computed by

```
In[5]:= zIntegrate[1, 3, 2] // Factor
```

$$\text{Out[5]}= \frac{1}{2}\,\mathrm{E}_2\,(\mathrm{s}_{1,2}^2 + \mathrm{s}_{1,3}^2 + \mathrm{s}_{2,3}^2) \ .$$

Note that all Mandelstam variables are treated as independent, no momentum conservation is imposed. A Koba–Nielsen factor which does not contain all Green functions and Mandelstam variables of (2.5) can be represented by replacing the second argument with a list of point pairs, corresponding to the Green functions appearing in the Koba–Nielsen factor. E.g. $\exp(s_{12}G_{12} + s_{13}G_{13})$ is represented by `{{1,2},{1,3}}`. For an integral without Koba–Nielsen factor, we can set the second argument of `zIntegrate` to an arbitrary number and the third argument to zero. For the integrand in front of the Koba–Nielsen factor, the functions listed in table 1 are available. To indicate their z dependence, they all carry a suffix `z`. An arbitrary polynomial in these functions can be given as the first argument to `zIntegrate`. E.g. the first order in Mandelstams of the integral

$$\int \mathrm{d}\mu_3 V_2(1,2,3,4)\mathrm{KN}_4 \tag{3.2}$$

is computed by

```
In[6]:= zIntegrate[vz[2, {1, 2, 3, 4}], 4, 1] // Factor
```

$$\text{Out[6]}= -\frac{\mathrm{C}\left[\begin{smallmatrix}3 & 0\\ 1 & 0\end{smallmatrix}\right](\mathrm{s}_{1,2} - 2\,\mathrm{s}_{1,3} + \mathrm{s}_{1,4} + \mathrm{s}_{2,3} - 2\,\mathrm{s}_{2,4} + \mathrm{s}_{3,4})\,\tau_2}{\pi} \ .$$

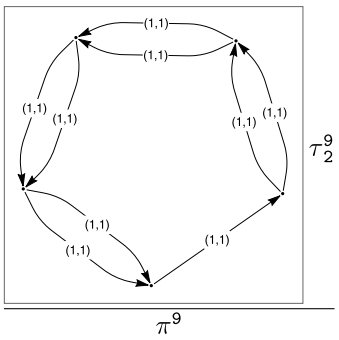
The function `zIntegrate` returns MGFs in the notation introduced in section 4 below for MGFs with up to four points, while exploiting some basic properties of MGFs as the ones

Table 1. Various z-dependent functions defined in section 2 and their representation in `Mathematica`.

Function	Definition	`Mathematica` representation
$f_{ij}^{(a)}$	(2.12b)	`fz[a, i, j]`
$\overline{f_{ij}^{(b)}}$	(2.12c)	`fBarz[b, i, j]`
G_{ij}	(2.7)	`gz[i, j]`
$C_{ij}^{(a,b)}$	(2.18)	`cz[a, b, i, j]`
$V_a(k_1,\dots,k_r)$	(2.16)	`vz[k1,...,kr]`
$\overline{V_b(k_1,\dots,k_r)}$	(2.16)	`vBarz[k1,...,kr]`

listed in section 2.2. If MGFs with more than four points appear in the expansion and they cannot be reduced by using these properties, they are printed as a graph, e.g.

```
In[7]:= zIntegrate[gz[1,2]^2 gz[2,3]^2 gz[3,4]^2 gz[4,5]^2 gz[5,1],5,0]
```

Out[7]= 

.

Note that if the Koba–Nielsen integral expanded using `zIntegrate` contains kinematic poles due to $f_{ij}^{(1)}\overline{f_{ij}^{(1)}}$ terms in the integrand, `zIntegrate` will contain divergent MGFs, as will be discussed in detail in section 8.2.

Using the function `zIntegrate` and the decompositions discussed in section 9 below, the two- and three-point generating functions for Koba–Nielsen integrals were evaluated in terms a few basis-MGFs up to total modular weight 12.

4. Graph topologies and notation

The general definition (2.23) for MGFs depends on a graph Γ with decorated and directed edges, where the decoration has the form (a, b) with $a, b \in \mathbb{Z}$. Since it is inconvenient to specify the entire graph for every MGF, we introduce commonly used notations for graphs with up to four vertices, the only ones considered in this paper.

4.1. Two-point modular graph forms

As introduced in (2.26), dihedral graphs have two vertices and all edges directed in the same way. They are denoted by [8]

$$\mathcal{C}\left[\begin{smallmatrix} a_1 & \cdots & a_R \\ b_1 & \cdots & b_R \end{smallmatrix}\right] = \quad 1 \quad \begin{matrix} (a_1,b_1) \\ (a_2,b_2) \\ \vdots \\ (a_R,b_R) \end{matrix} \quad 2 \quad = \sum_{p_1,\ldots,p_R}{}' \frac{\delta(p_1+\cdots+p_R)}{p_1^{a_1}\bar{p}_1^{b_1}\cdots p_R^{a_R}\bar{p}_R^{b_R}} \,. \tag{4.1}$$

This class of functions, as well as many special cases, were studied extensively in the literature [3, 4, 6, 8, 11, 15, 21–23, 25, 26, 31]. Since we will frequently encounter a bundle of parallel edges, we write

$$\mathcal{C}\left[\begin{smallmatrix} A \\ B \end{smallmatrix}\right] = \mathcal{C}\left[\begin{smallmatrix} a_1 & \cdots & a_R \\ b_1 & \cdots & b_R \end{smallmatrix}\right] \tag{4.2}$$

and call $\begin{bmatrix} A \\ B \end{bmatrix}$ a *block*. In graphs, we draw

$$1 \quad \left[\begin{smallmatrix} A \\ B \end{smallmatrix}\right] \quad 2 \quad = \quad 1 \quad \begin{matrix} (a_1,b_1) \\ (a_2,b_2) \\ \vdots \\ (a_R,b_R) \end{matrix} \quad 2 \,. \tag{4.3}$$

For $|A| = |B|$, (cf (2.22)) we also introduce the antisymmetric version

$$\mathcal{A}\begin{bmatrix} A \\ B \end{bmatrix} = \mathcal{C}\begin{bmatrix} A \\ B \end{bmatrix} - \mathcal{C}\begin{bmatrix} B \\ A \end{bmatrix}, \tag{4.4}$$

which is purely imaginary and was first studied in [21]. Under the transformation $\tau \to -\bar{\tau}$, any MGF satisfies $\mathcal{C}_\Gamma(-\bar{\tau}) = \overline{\mathcal{C}_\Gamma(\tau)}$ and hence we have that $\mathcal{A}\begin{bmatrix} A \\ B \end{bmatrix}(-\bar{\tau}) = -\mathcal{A}\begin{bmatrix} A \\ B \end{bmatrix}(\tau)$. Since τ_2 is invariant under this transformation and the Laurent polynomial is mapped to its negative, the Laurent polynomial of $\mathcal{A}\begin{bmatrix} A \\ B \end{bmatrix}$ has to vanish, i.e. $\mathcal{A}\begin{bmatrix} A \\ B \end{bmatrix}$ is a cusp form.

In the `ModularGraphForms` package, MGFs have head `c`, i.e. they are formally given by the function `c` applied to various arguments. Dihedral MGFs have one argument which is a $2 \times R$ matrix which can, as any other matrix, be inserted in two-dimensional form or as a nested list,

```
In[8]:= c[1 2 3] + c[{{1,2,3},{1,1,1}}]
1 1 1

Out[8]= 2 C[1 2 3] .
        1 1 1
```

Imaginary cusp forms of the form (4.4) have head `a`,

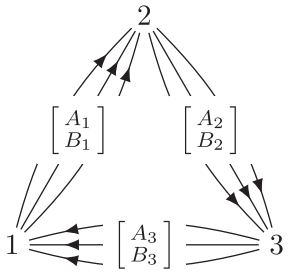
```
In[9]:= a[0 2 3]
         3 0 2

Out[9]= A[0 2 3] .
         3 0 2
```

4.2. Three-point modular graph forms

Trihedral graphs have three vertices. The notation we use is [8]

$$\mathcal{C}\left[\begin{smallmatrix}A_1\\B_1\end{smallmatrix}\middle|\begin{smallmatrix}A_2\\B_2\end{smallmatrix}\middle|\begin{smallmatrix}A_3\\B_3\end{smallmatrix}\right] = \quad (4.5)$$

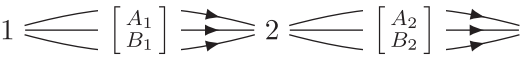


and hence

$$\mathcal{C}\left[\begin{smallmatrix}A_1\\B_1\end{smallmatrix}\middle|\begin{smallmatrix}A_2\\B_2\end{smallmatrix}\middle|\begin{smallmatrix}A_3\\B_3\end{smallmatrix}\right] = {\sum_{\{p_i^{(j)}\}}}' \left(\prod_{j=1}^{3}\prod_{i=1}^{R_j}\frac{1}{(p_i^{(j)})^{a_i^{(j)}}(\bar{p}_i^{(j)})^{b_i^{(j)}}}\right)\delta\left(\sum_{i=1}^{R_1}p_i^{(1)}-\sum_{i=1}^{R_2}p_i^{(2)}\right)\delta\left(\sum_{i=1}^{R_2}p_i^{(2)}-\sum_{i=1}^{R_3}p_i^{(3)}\right), \quad (4.6)$$

where the block $\left[\begin{smallmatrix}A_j\\B_j\end{smallmatrix}\right]$ has R_j columns. We will use this notation henceforth. If two vertices are not connected by any edges, we write[1]

$$\mathcal{C}\left[\begin{smallmatrix}A_1\\B_1\end{smallmatrix}\middle|\begin{smallmatrix}A_2\\B_2\end{smallmatrix}\middle|\varnothing\right] = \quad (4.7)$$



In the `ModularGraphForms` package, the function `c` with three matrix-arguments is used,

```
In[10]:= c[{1 1},{2 3},{4 5}]
Out[10]= C[{1 1}|{2 3}|{4 5}]
```

The edge directions and normalization are as in (4.5) and (4.6), respectively. For empty blocks, we use empty lists,

```
In[11]:= c[{},{1 2},{3 4}]
Out[11]= C[{}|{1 2}|{3 4}]
```

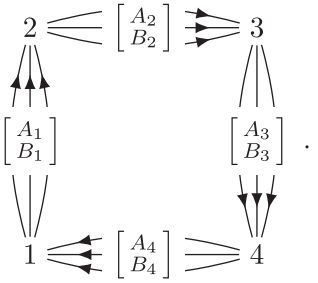
4.3. Four-point modular graph forms

Due to their different symmetry properties, it is convenient to distinguish the following three topologies among four-point graphs.

[1] This graph factorizes according to (5.13) into $\mathcal{C}\left[\begin{smallmatrix}A_1\\B_1\end{smallmatrix}\right]\mathcal{C}\left[\begin{smallmatrix}A_2\\B_2\end{smallmatrix}\right]$.

Box graphs have four edges in one cycle and are denoted by

$$\mathcal{C}\left[\begin{smallmatrix}A_1\\B_1\end{smallmatrix}\middle|\begin{smallmatrix}A_2\\B_2\end{smallmatrix}\middle|\begin{smallmatrix}A_3\\B_3\end{smallmatrix}\middle|\begin{smallmatrix}A_4\\B_4\end{smallmatrix}\right] = \tag{4.8}$$



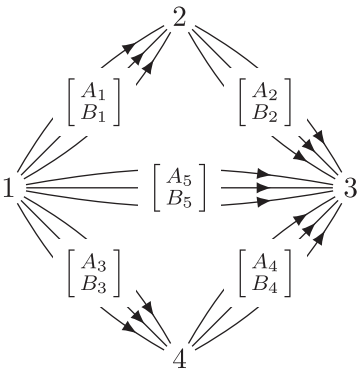
The lattice sum representation similarly to (4.6) can be read off straightforwardly from the graph. In `Mathematica`, we use `C` with four arguments,

```
In[12]:= C[1 2, 3 4, 5 6, 7 8]
          1 1  1 1  1 1  1 1

Out[12]= C[1 2|3 4|5 6|7 8]
          1 1|1 1|1 1|1 1
```

Kite graphs have five edges: the cyclic ones from the box plus one diagonal. We write:

$$\mathcal{C}\left[\begin{smallmatrix}A_1\\B_1\end{smallmatrix}\middle|\begin{smallmatrix}A_2\\B_2\end{smallmatrix}\middle\|\begin{smallmatrix}A_3\\B_3\end{smallmatrix}\middle|\begin{smallmatrix}A_4\\B_4\end{smallmatrix}\middle\|\begin{smallmatrix}A_5\\B_5\end{smallmatrix}\right] = \tag{4.9}$$



Note that the direction of the four outer edges is different from the box graph.

For kite graphs, `C` has five arguments,

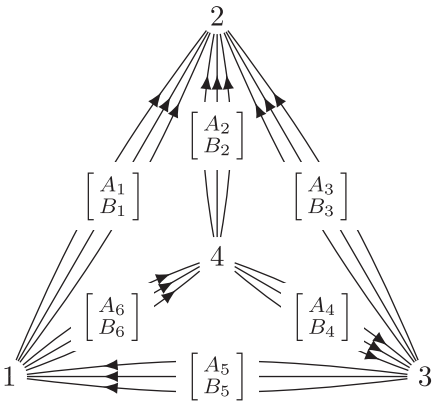
```
In[13]:= C[1 2, 1 3, 1 4, 1 5, 1 6]
          1 1  1 1  1 1  1 1  1 1

Out[13]= C[1 2|1 3||1 4|1 5||1 6]
          1 1|1 1||1 1|1 1||1 1
```

Finally, the full *tetrahedral graph* (also known as *Mercedes graph*) has six edges connecting all pairs of points. The Laplace eigenvalue equations of modular graph functions of this topology were studied in [12]. As will become clear in the next section, due to its symmetry

properties, it is convenient to arrange the six blocks in three columns as follows[2]:

$$\mathcal{C}\left[\begin{array}{c|c|c} A_1 & A_2 & A_3 \\ B_1 & B_2 & B_3 \\ \hline A_4 & A_5 & A_6 \\ B_4 & B_5 & B_6 \end{array}\right] = \quad . \tag{4.10}$$



Note that in this notation, edge bundles which do not share a common vertex correspond to blocks written in one column.

Tetrahedral graphs are written in the `ModularGraphForms` package as `c` with six arguments,

```
In[14]:= c[1 2, 1 3, 1 4, 1 5, 1 6, 1 7]
         1 1  1 1  1 1  1 1  1 1  1 1

Out[14]= c[ 1 2 | 1 3 | 1 4 ]
           1 1 | 1 1 | 1 1
           -----------------
           1 5 | 1 6 | 1 7
           1 1 | 1 1 | 1 1
```

For all four-point graphs, we will again use the symbol $\emptyset$ to denote blocks without any edges. In `Mathematica`, we again use empty lists.

5. Simple relations

There are a number of relations between MGFs that follow directly from their definition in terms of graphs and lattice sums. These are easy to see, yet very powerful and already generate a lot of identities.

5.1. Symmetries

Given the graph of a MGF, the associated $\mathcal{C}$-function as defined in the previous section is ambiguous and this generates relations between $\mathcal{C}$-functions with different labels. In the simplest instance, permutations of the columns of a dihedral graph leave the MGF invariant. The same is true for permutations of columns in any block of the higher-point graphs.

If a vertex is connected to only two edge bundles, their total momenta have to agree and hence the two bundles can be swapped without changing the lattice sum associated to the graph [8]. For trihedral- and box graphs this implies invariance under permutations of the blocks [8],

$$\mathcal{C}\left[\begin{array}{c|c|c} A_1 & A_2 & A_3 \\ B_1 & B_2 & B_3 \end{array}\right] = \mathcal{C}\left[\begin{array}{c|c|c} A_2 & A_1 & A_3 \\ B_2 & B_1 & B_3 \end{array}\right] = \mathcal{C}\left[\begin{array}{c|c|c} A_1 & A_3 & A_2 \\ B_1 & B_3 & B_2 \end{array}\right] \tag{5.1}$$

[2] In the conventions of [12], the direction of the edges in third block is reversed.

and similarly for block graphs.

For the same reason, kite graphs are invariant under swapping blocks 1 and 2 as well as 3 and 4. Furthermore, swapping the vertices 2 and 4 leaves the graph invariant, so in total the symmetries are

$$\begin{aligned}\mathcal{C}\left[\begin{smallmatrix}A_1\\B_1\end{smallmatrix}\middle|\begin{smallmatrix}A_2\\B_2\end{smallmatrix}\middle\|\begin{smallmatrix}A_3\\B_3\end{smallmatrix}\middle|\begin{smallmatrix}A_4\\B_4\end{smallmatrix}\middle\|\begin{smallmatrix}A_5\\B_5\end{smallmatrix}\right] &= \mathcal{C}\left[\begin{smallmatrix}A_2\\B_2\end{smallmatrix}\middle|\begin{smallmatrix}A_1\\B_1\end{smallmatrix}\middle\|\begin{smallmatrix}A_3\\B_3\end{smallmatrix}\middle|\begin{smallmatrix}A_4\\B_4\end{smallmatrix}\middle\|\begin{smallmatrix}A_5\\B_5\end{smallmatrix}\right]\\ = \mathcal{C}\left[\begin{smallmatrix}A_1\\B_1\end{smallmatrix}\middle|\begin{smallmatrix}A_2\\B_2\end{smallmatrix}\middle\|\begin{smallmatrix}A_4\\B_4\end{smallmatrix}\middle|\begin{smallmatrix}A_3\\B_3\end{smallmatrix}\middle\|\begin{smallmatrix}A_5\\B_5\end{smallmatrix}\right] &= \mathcal{C}\left[\begin{smallmatrix}A_3\\B_3\end{smallmatrix}\middle|\begin{smallmatrix}A_4\\B_4\end{smallmatrix}\middle\|\begin{smallmatrix}A_1\\B_1\end{smallmatrix}\middle|\begin{smallmatrix}A_2\\B_2\end{smallmatrix}\middle\|\begin{smallmatrix}A_5\\B_5\end{smallmatrix}\right].\end{aligned} \tag{5.2}$$

The double-line notation was chosen to make this intuitive. Note that the vertices in kite graphs are not all equivalent and this gives rise to the more complex symmetry properties (5.2).

Tetrahedral graphs have an S_4 permutation symmetry from relabeling the four equivalent vertices. These 24 permutations are generated by six permutations:

- Three permutations of columns: flipping a column comprised of two (A_i, B_i)-blocks in (4.10) with any other column produces a sign $(-1)^{|1|+|2|+|3|}$ where $|1|+|2|+|3| = |A_1| + |\mathrm{B}_1| + |A_2| + |\mathrm{B}_2| + |A_3| + |\mathrm{B}_3|$ is a shorthand for the combined modular weight of the top row.[3] Explicitly:

$$\begin{aligned}\mathcal{C}\left[\begin{smallmatrix}A_1\\B_1\\\hline A_4\\B_4\end{smallmatrix}\middle\|\begin{smallmatrix}A_2\\B_2\\\hline A_5\\B_5\end{smallmatrix}\middle\|\begin{smallmatrix}A_3\\B_3\\\hline A_6\\B_6\end{smallmatrix}\right] &= (-1)^{|1|+|2|+|3|}\,\mathcal{C}\left[\begin{smallmatrix}A_2\\B_2\\\hline A_5\\B_5\end{smallmatrix}\middle\|\begin{smallmatrix}A_1\\B_1\\\hline A_4\\B_4\end{smallmatrix}\middle\|\begin{smallmatrix}A_3\\B_3\\\hline A_6\\B_6\end{smallmatrix}\right]\\ &= (-1)^{|1|+|2|+|3|}\,\mathcal{C}\left[\begin{smallmatrix}A_3\\B_3\\\hline A_6\\B_6\end{smallmatrix}\middle\|\begin{smallmatrix}A_2\\B_2\\\hline A_5\\B_5\end{smallmatrix}\middle\|\begin{smallmatrix}A_1\\B_1\\\hline A_4\\B_4\end{smallmatrix}\right]\\ &= (-1)^{|1|+|2|+|3|}\,\mathcal{C}\left[\begin{smallmatrix}A_1\\B_1\\\hline A_4\\B_4\end{smallmatrix}\middle\|\begin{smallmatrix}A_3\\B_3\\\hline A_6\\B_6\end{smallmatrix}\middle\|\begin{smallmatrix}A_2\\B_2\\\hline A_5\\B_5\end{smallmatrix}\right].\end{aligned} \tag{5.3}$$

- Three flips of two top/bottom pairs: flipping the top/bottom blocks in any two columns changes the tetrahedral graph by a sign $(-1)^{|k|+|l|}$, where k and l in $|k|+|l| = |A_k| + |\mathrm{B}_k| + |A_l| + |\mathrm{B}_l|$ are given by the following prescription: permute the three columns cyclically until the two columns in which top and bottom blocks are swapped are next to each other. The blocks in the left one of these has indices k and l. Explicitly:

$$\begin{aligned}\mathcal{C}\left[\begin{smallmatrix}A_1\\B_1\\\hline A_4\\B_4\end{smallmatrix}\middle\|\begin{smallmatrix}A_2\\B_2\\\hline A_5\\B_5\end{smallmatrix}\middle\|\begin{smallmatrix}A_3\\B_3\\\hline A_6\\B_6\end{smallmatrix}\right] &= (-1)^{|1|+|4|}\,\mathcal{C}\left[\begin{smallmatrix}A_4\\B_4\\\hline A_1\\B_1\end{smallmatrix}\middle\|\begin{smallmatrix}A_5\\B_5\\\hline A_2\\B_2\end{smallmatrix}\middle\|\begin{smallmatrix}A_3\\B_3\\\hline A_6\\B_6\end{smallmatrix}\right]\\ &= (-1)^{|2|+|5|}\,\mathcal{C}\left[\begin{smallmatrix}A_1\\B_1\\\hline A_4\\B_4\end{smallmatrix}\middle\|\begin{smallmatrix}A_5\\B_5\\\hline A_2\\B_2\end{smallmatrix}\middle\|\begin{smallmatrix}A_6\\B_6\\\hline A_3\\B_3\end{smallmatrix}\right]\\ &= (-1)^{|3|+|6|}\,\mathcal{C}\left[\begin{smallmatrix}A_4\\B_4\\\hline A_1\\B_1\end{smallmatrix}\middle\|\begin{smallmatrix}A_2\\B_2\\\hline A_5\\B_5\end{smallmatrix}\middle\|\begin{smallmatrix}A_6\\B_6\\\hline A_3\\B_3\end{smallmatrix}\right].\end{aligned} \tag{5.4}$$

[3] The sign does not depend on if we take the modular weight of the top- or bottom row since the total modular weight is even for non-vanishing MGFs.

The arrangement of the blocks in two rows of three columns was chosen to make these symmetries intuitive. For tetrahedral graphs, although all vertices are equivalent, the symmetry of the graph is broken by the direction of the edges, i.e. it is not possible to assign the directions in such a way that every vertex has the same number of ingoing and outgoing edges. Adjusting the edge direction when relabeling vertices leads to the signs in (5.3) and (5.4). These signs also mean that tetrahedral graphs can vanish by symmetry although their sum of holomorphic and antiholomorphic labels is even. E.g., according to (5.3),

$$\mathcal{C}\left[\begin{array}{c|c|c}\begin{matrix}A\\B\end{matrix} & \begin{matrix}A\\B\end{matrix} & \begin{matrix}A\\B\end{matrix}\\ \hline \begin{matrix}A\\B\end{matrix} & \begin{matrix}A\\B\end{matrix} & \begin{matrix}A\\B\end{matrix}\end{array}\right] = 0\,, \tag{5.5}$$

if $|A|+|B|$ odd, although $6(|A|+|B|)$ is even. This form of vanishing by symmetry does not exist for any of the other discussed graphs since no signs appear in their symmetry transformations.

In light of the above symmetry properties it is convenient to define a *canonical representation* for the graph topologies discussed so far such that graphs related by a symmetry transformation are represented by the same arguments of the $\mathcal{C}$-function. To this end, we define an ordering on the set of two-row columns and on the set of $2\times R$ matrices. This will allow us to define an ordering on the MGFs of a certain topology and the smallest element in the symmetry orbit of an MGF will be the canonical representation of that graph.

The columns within an $\left[\begin{matrix}A\\B\end{matrix}\right]$-block can be permuted arbitrarily for all graphs introduced above. The canonical representation of the MGFs therefore has the columns in each block in lexicographic order[4] w.r.t. the ordering defined by

- If $a_1 < a_2$ then $\left[\begin{matrix}a_1\\b_1\end{matrix}\right] < \left[\begin{matrix}a_2\\b_2\end{matrix}\right]$.
- If $a_1 = a_2$ then $\left[\begin{matrix}a_1\\b_1\end{matrix}\right] < \left[\begin{matrix}a_2\\b_2\end{matrix}\right]$ if $\mathrm{b}_1 < \mathrm{b}_2$.

Given two blocks $\left[\begin{matrix}A_1\\B_1\end{matrix}\right]$ and $\left[\begin{matrix}A_2\\B_2\end{matrix}\right]$ with canonical column order and R_1 and R_2 columns, respectively, we can define a canonical ordering of the two blocks by

- If $R_1 < R_2$ then $\left[\begin{matrix}A_1\\B_1\end{matrix}\right] < \left[\begin{matrix}A_2\\B_2\end{matrix}\right]$.
- If $R_1 = R_2$ then $\left[\begin{matrix}A_1\\B_1\end{matrix}\right] < \left[\begin{matrix}A_2\\B_2\end{matrix}\right]$ if $A_1 < A_2$ in lexicographic order.
- If $A_1 = A_2$ then $\left[\begin{matrix}A_1\\B_1\end{matrix}\right] < \left[\begin{matrix}A_2\\B_2\end{matrix}\right]$ if $\mathrm{B}_1 < \mathrm{B}_2$ in lexicographic order.

Using this ordering, we define

$$\mathcal{C}\left[\begin{matrix}A_1\\B_1\end{matrix}\right] < \mathcal{C}\left[\begin{matrix}A_2\\B_2\end{matrix}\right] \quad \text{if } \left[\begin{matrix}A_1\\B_1\end{matrix}\right] < \left[\begin{matrix}A_2\\B_2\end{matrix}\right] \tag{5.6}$$

[4] In lexicographic order, the sequence $a_1, a_2, \ldots, a_n$ is smaller than $b_1, b_2, \ldots, b_n$ if $a_i < b_i$ for the first i for which $a_i \neq b_i$.

for dihedral graphs, unless the graph at hand is a one-loop graph. In this case, we write $\mathcal{C}\begin{bmatrix} a & 0 \\ b & 0 \end{bmatrix}$ instead of $\mathcal{C}\begin{bmatrix} 0 & a \\ 0 & b \end{bmatrix}$, to be consistent with the previous literature. For graphs with several blocks, we use lexicographic ordering on the set of blocks, hence

$$\mathcal{C}\left[\begin{smallmatrix} A_1 \\ B_1 \end{smallmatrix}\middle|\begin{smallmatrix} A_2 \\ B_2 \end{smallmatrix}\middle|\begin{smallmatrix} A_3 \\ B_3 \end{smallmatrix}\right] < \mathcal{C}\left[\begin{smallmatrix} C_1 \\ D_1 \end{smallmatrix}\middle|\begin{smallmatrix} C_2 \\ D_2 \end{smallmatrix}\middle|\begin{smallmatrix} C_3 \\ D_3 \end{smallmatrix}\right] \qquad \text{if} \qquad \left(\begin{bmatrix} A_1 \\ B_1 \end{bmatrix}, \begin{bmatrix} A_2 \\ B_2 \end{bmatrix}, \begin{bmatrix} A_3 \\ B_3 \end{bmatrix}\right) < \left(\begin{bmatrix} C_1 \\ D_1 \end{bmatrix}, \begin{bmatrix} C_2 \\ D_2 \end{bmatrix}, \begin{bmatrix} C_3 \\ D_3 \end{bmatrix}\right) \tag{5.7}$$

in lexicographic order and similarly for all four-point graphs with the numbering of the blocks as in section 4.3.

For trihedral and box graphs, this just means that the canonical representation has the blocks (and in each block the columns) in lexicographic ordering. For kite graphs, the fifth block cannot be moved by the symmetries (5.2) and hence in the canonical representation, the smallest block out of the remaining four comes first, fixing the second one. The third block is the smaller one out of the remaining two, fixing the last block. Canonically represented tetrahedral graphs have the smallest block in the upper left slot, fixing the lower left block. The smallest block out of the remaining four blocks sits in the upper middle slot, fixing all remaining entries. The following examples are all in their canonical representation

$$\mathcal{C}\left[\begin{smallmatrix} 3 & 0 \\ 1 & 0 \end{smallmatrix}\right] \tag{5.8a}$$

$$\mathcal{C}\left[\begin{smallmatrix} 1 & 2 & 3 \\ 7 & 5 & 4 \end{smallmatrix}\right] \tag{5.8b}$$

$$\mathcal{C}\left[\begin{smallmatrix} 2 \\ 2 \end{smallmatrix}\middle|\begin{smallmatrix} 1 & 1 \\ 1 & 1 \end{smallmatrix}\middle|\begin{smallmatrix} 0 & 0 & 1 \\ 2 & 4 & 1 \end{smallmatrix}\right] \tag{5.8c}$$

$$\mathcal{C}\left[\begin{smallmatrix} 2 \\ 2 \end{smallmatrix}\middle|\begin{smallmatrix} 1 & 2 & 3 \\ 7 & 5 & 4 \end{smallmatrix}\middle\|\begin{smallmatrix} 1 & 1 \\ 1 & 1 \end{smallmatrix}\middle|\begin{smallmatrix} 0 & 0 & 1 \\ 2 & 4 & 1 \end{smallmatrix}\middle\|\begin{smallmatrix} 1 \\ 1 \end{smallmatrix}\right] \tag{5.8d}$$

$$\mathcal{C}\left[\begin{array}{c|c||c|c||c} 1 & 2 & 2 & & 2\ 3 \\ \hline 1 & & \hline 2 & & 1\ 2 \\ \hline 7\ 5\ 4 & 0\ 0\ 1 & & 1\ 1 & \\ & 2\ 4\ 1 & & 1\ 1 & \end{array}\right]. \tag{5.8e}$$

In the `ModularGraphForms` package, the function `CSort` brings MGFs into their canonical form, using the symmetries discussed above. For the MGFs in (5.8), we have e.g.

```
In[15]:= CSort[{c[0 3; 0 1], c[2 1 3; 5 7 4], c[1 0 0, 2, 1 1; 1 4 2, 2, 1 1],
    c[0 0 1, 1 1, 3 2 1, 2, 1; 4 2 1, 1 1, 4 5 7, 2, 1], c[2 3, 1 0 0, 3 2 1, 1 1, 2, 1; 1 2, 1 4 2, 4 5 7, 1 1, 2, 1]}]
Out[15]= {c[3 0; 1 0], c[1 2 3; 7 5 4], c[2|1 1|0 0 1; 2|1 1|2 4 1], c[2|1 2 3||1 1|0 0 1||1; 2|7 5 4||1 1|2 4 1||1],
    c[1 | 2 | 2 3; 1 | 2 | 1 2; 7 5 4 | 0 0 1 | 1 1; 1 | 1 | 1 1]}.
```

The output of the function `CSimplify` is always in canonical form. The property, that tetrahedral graphs can vanish by symmetry, as in the example (5.5), is implemented in the function

`TetCSimplify`. E.g., we have

```
In[16]:= TetCSimplify[c[1/2 2, 1/2 2, 1/2 2, 1/2 2, 1/2 2, 1/2 2]]

Out[16]= 0
```

5.2. Topological simplifications

For certain special cases of the graphs defined in section 4, the MGF simplifies.

For the dihedral case, the fact that one-valent vertices lead to vanishing MGFs can be expressed as

$$\mathcal{C}\begin{bmatrix} a \\ b \end{bmatrix} = 0. \tag{5.9}$$

It is furthermore convenient to define

$$\mathcal{C}\left[\emptyset\right] = 1. \tag{5.10}$$

The property (2.37) that two-valent vertices can be dropped translates for one-loop dihedral graphs into

$$\mathcal{C}\begin{bmatrix} a_1 & a_2 \\ b_1 & b_2 \end{bmatrix} = (-1)^{a_2+b_2}\mathcal{C}\begin{bmatrix} a_1+a_2 & 0 \\ b_1+b_2 & 0 \end{bmatrix}. \tag{5.11}$$

For trihedral graphs, (2.37) implies

$$\mathcal{C}\left[\begin{smallmatrix} a_1 \\ b_1 \end{smallmatrix}\middle|\begin{smallmatrix} a_2 \\ b_2 \end{smallmatrix}\middle|\begin{smallmatrix} A_3 \\ B_3 \end{smallmatrix}\right] = (-1)^{a_1+b_1+a_2+b_2}\,\mathcal{C}\left[\begin{smallmatrix} a_1+a_2 & A_3 \\ b_1+b_2 & B_3 \end{smallmatrix}\right] \tag{5.12}$$

and the factorization of one-particle reducible graphs (2.38) means that trihedral graphs with one empty block factorize into dihedral graphs,

$$\mathcal{C}\left[\begin{smallmatrix} A_1 \\ B_1 \end{smallmatrix}\middle|\begin{smallmatrix} A_2 \\ B_2 \end{smallmatrix}\middle|\begin{smallmatrix} \varnothing \end{smallmatrix}\right] = \mathcal{C}\left[\begin{smallmatrix} A_1 \\ B_1 \end{smallmatrix}\right]\mathcal{C}\left[\begin{smallmatrix} A_2 \\ B_2 \end{smallmatrix}\right]. \tag{5.13}$$

Via (5.10), this also captures the case of two empty blocks.

Since two- and three-point graphs are special cases of four-point graphs, topological simplifications of four-point graphs should allow for simplifications down to dihedral graphs. We will provide a hierarchy of simplifications from tetrahedral graphs to box graphs which, if applied repeatedly together with (5.9) to (5.13), allow to identify any lower-point graph which is given as a tetrahedral MGF.

Tetrahedral graphs with one empty block are kite graphs,

$$\mathcal{C}\left[\begin{smallmatrix} \varnothing \\ \hline A_4 \\ B_4 \end{smallmatrix}\middle\|\begin{smallmatrix} A_2 \\ B_2 \\ \hline A_5 \\ B_5 \end{smallmatrix}\middle\|\begin{smallmatrix} A_3 \\ B_3 \\ \hline A_6 \\ B_6 \end{smallmatrix}\right] = (-1)^{|A_2|+|B_2|+|A_4|+|B_4|}\,\mathcal{C}\left[\begin{smallmatrix} A_2 \\ B_2 \end{smallmatrix}\middle|\begin{smallmatrix} A_3 \\ B_3 \end{smallmatrix}\middle\|\begin{smallmatrix} A_5 \\ B_5 \end{smallmatrix}\middle|\begin{smallmatrix} A_6 \\ B_6 \end{smallmatrix}\middle\|\begin{smallmatrix} A_4 \\ B_4 \end{smallmatrix}\right]. \tag{5.14}$$

A kite graph with one empty block is either a box graph or factorizes,

$$\mathcal{C}\left[\begin{smallmatrix} \varnothing \end{smallmatrix}\middle|\begin{smallmatrix} A_2 \\ B_2 \end{smallmatrix}\middle\|\begin{smallmatrix} A_3 \\ B_3 \end{smallmatrix}\middle|\begin{smallmatrix} A_4 \\ B_4 \end{smallmatrix}\middle\|\begin{smallmatrix} A_5 \\ B_5 \end{smallmatrix}\right] = (-1)^{|A_5|+|B_5|}\,\mathcal{C}\left[\begin{smallmatrix} A_2 \\ B_2 \end{smallmatrix}\right]\mathcal{C}\left[\begin{smallmatrix} A_3 \\ B_3 \end{smallmatrix}\middle|\begin{smallmatrix} A_4 \\ B_4 \end{smallmatrix}\middle|\begin{smallmatrix} A_5 \\ B_5 \end{smallmatrix}\right] \tag{5.15a}$$

$$\mathcal{C}\left[\begin{smallmatrix}A_1\\B_1\end{smallmatrix}\middle|\begin{smallmatrix}A_2\\B_2\end{smallmatrix}\middle\|\begin{smallmatrix}A_3\\B_3\end{smallmatrix}\middle|\begin{smallmatrix}A_4\\B_4\end{smallmatrix}\middle\|\varnothing\right] = (-1)^{|A_3|+|B_3|+|A_4|+|B_4|}\,\mathcal{C}\left[\begin{smallmatrix}A_1\\B_1\end{smallmatrix}\middle|\begin{smallmatrix}A_2\\B_2\end{smallmatrix}\middle|\begin{smallmatrix}A_3\\B_3\end{smallmatrix}\middle|\begin{smallmatrix}A_4\\B_4\end{smallmatrix}\right]. \tag{5.15b}$$

If the two blocks in the first (or second) pair of blocks have only one column each, the vertex 2 (or 4) becomes two-valent end the kite graph simplifies into a trihedral graph,

$$\mathcal{C}\left[\begin{smallmatrix}a_1\\b_1\end{smallmatrix}\middle|\begin{smallmatrix}a_2\\b_2\end{smallmatrix}\middle\|\begin{smallmatrix}A_3\\B_3\end{smallmatrix}\middle|\begin{smallmatrix}A_4\\B_4\end{smallmatrix}\middle\|\begin{smallmatrix}A_5\\B_5\end{smallmatrix}\right] = (-1)^{|A_3|+|B_3|+|A_4|+|B_4|}\,\mathcal{C}\left[\begin{smallmatrix}a_1+a_2 & A_5\\b_1+b_2 & B_5\end{smallmatrix}\middle|\begin{smallmatrix}A_3\\B_3\end{smallmatrix}\middle|\begin{smallmatrix}A_4\\B_4\end{smallmatrix}\right]. \tag{5.16}$$

A box graph with one (or more) empty blocks factorizes into dihedral graphs,

$$\mathcal{C}\left[\varnothing\middle|\begin{smallmatrix}A_2\\B_2\end{smallmatrix}\middle|\begin{smallmatrix}A_3\\B_3\end{smallmatrix}\middle|\begin{smallmatrix}A_4\\B_4\end{smallmatrix}\right] = \mathcal{C}\left[\begin{smallmatrix}A_2\\B_2\end{smallmatrix}\right]\mathcal{C}\left[\begin{smallmatrix}A_3\\B_3\end{smallmatrix}\right]\mathcal{C}\left[\begin{smallmatrix}A_4\\B_4\end{smallmatrix}\right] \tag{5.17}$$

and a box graph with two blocks of only one column each has a two-valent vertex and simplifies is a trihedral graph,

$$\mathcal{C}\left[\begin{smallmatrix}a_1\\b_1\end{smallmatrix}\middle|\begin{smallmatrix}a_2\\b_2\end{smallmatrix}\middle|\begin{smallmatrix}A_3\\B_3\end{smallmatrix}\middle|\begin{smallmatrix}A_4\\B_4\end{smallmatrix}\right] = \mathcal{C}\left[\begin{smallmatrix}a_1+a_2\\b_1+b_2\end{smallmatrix}\middle|\begin{smallmatrix}A_3\\B_3\end{smallmatrix}\middle|\begin{smallmatrix}A_4\\B_4\end{smallmatrix}\right]. \tag{5.18}$$

Combined, the relations above show e.g. that

$$\mathcal{C}\left[\frac{\varnothing}{\begin{smallmatrix}1\\1\end{smallmatrix}}\middle\|\frac{\begin{smallmatrix}1\\1\end{smallmatrix}}{\begin{smallmatrix}1\\1\end{smallmatrix}}\middle\|\frac{\begin{smallmatrix}1\\1\end{smallmatrix}}{\begin{smallmatrix}1\\1\end{smallmatrix}}\right] = \mathcal{C}\left[\begin{smallmatrix}1&2&2\\1&2&2\end{smallmatrix}\right]. \tag{5.19}$$

In the `Mathematica` package `ModularGraphForms`, the dihedral relations (5.9)–(5.11) are implement in the function `DiCSimplify`,

```
In[17]:= DiCSimplify[c[{}]c[0 3; 1 0] + c[3; 1]]

Out[17]= -C[3 0; 1 0] .
```

`DiCSimplify` also rewrites the special cases $\widehat{\mathrm{G}}_2$, G_k and E_k of one-loop graphs according to (2.30b) and (2.30c), as well as (2.36), whereas the one-loop simplification (5.11) is also performed by `CSort`. The function `DiCSimplify` has a Boolean option `basisExpandG` which, if set to `True`, causes `DiCSimplify` to expand all holomorphic Eisenstein series in the ring of G_4 and G_6, e.g.

```
In[18]:= DiCSimplify[g[24], basisExpandG→True]

Out[18]= 270 G4^6/66079 + 5400000 G4^3 G6^2/151915621 + 375 G6^4/73853 .
```

The default value of `basisExpandG` is `False`.

The trihedral simplifications (5.12) and (5.13) are performed by `TriCSimplify`,

```
In[19]:= TriCSimplify[c[{}, 1 2, 1 4] + c[1, 2, 1 3]]

Out[19]= C[1 2; 1 1] C[1 4; 1 1] + C[3 1 3; 3 1 1] .
```

Note that the dihedral graphs in `Out[19]` are not simplified or canonically represented, since `TriCSimplify` only acts on trihedral graphs. To simplify `Out[19]` further, we can apply `DiCSimplify`,

```
In[20]:= DiCSimplify[Out[19], useIds→False]

Out[20]= C[1 3 3; 1 1 3] ,
```

where the Boolean option `useIds` was set to suppress the expansion using the result in the basis decompositions to be discussed in section 9. The hierarchy of topological four-point simplifications (5.14)–(5.18) is implemented in the function `TetCSimplify`. Combining these functions, one can reproduce the example (5.19),

```
In[21]:= TetCSimplify[c[{}, 1/1, 1/1, 1/1, 1/1]]
         TriCSimplify[%]
         CSort[%]

Out[21]= C[2 1 | 1 | 1; 2 1 | 1 | 1]

Out[22]= C[2 1 2; 2 1 2]

Out[23]= C[1 2 2; 1 2 2] .
```

The function `CSimplify` acts on MGFs of all topologies and calls `DiCSimplify`, `TriCSimplify` and `TetCSimplify`. It also inherits the option `basisExpandG` from `DiCSimplify`. We have e.g.

```
In[24]:= CSimplify[c[{}, 1 2, {}, 1/2, 2/1, 2/2]]

Out[24]= π^8 E_3 E_5 / τ_2^8 .
```

5.3. Momentum conservation

Momentum conservation [8] will be the central tool in our derivation of identities between MGFs and can be derived in the lattice sum representation (2.23) as well as the integral representation (2.20) of the MGF. As long as all graphs involved are convergent, as we will assume in this section, both approaches result in the same expression. If divergent graphs are involved, the integral representation allows one to use the tools of complex analysis to derive meaningful results, cf section 8.6.

Starting from the lattice-sum representation (2.23) of an MGF with $|A| + |B|$ odd (hence, a vanishing MGF), which we will refer to as the *seed*, we have for each $j \in V_\Gamma$ the *momentum conservation identities*

$$0 = \sum_{e' \in E_\Gamma} \Gamma_{je'} {\sum_{\{p_e\}}}' \prod_{e \in E_\Gamma} \frac{p_{e'}}{p_e^{a_e} \bar{p}_e^{b_e}} \prod_{i \in V_\Gamma} \delta\left(\sum_{e'' \in E_\Gamma} \Gamma_{ie''} p_{e''} \right) \tag{5.20a}$$

$$0 = \sum_{e' \in E_\Gamma} \Gamma_{je'} {\sum_{\{p_e\}}}' \prod_{e \in E_\Gamma} \frac{\bar{p}_{e'}}{p_e^{a_e} \bar{p}_e^{b_e}} \prod_{i \in V_\Gamma} \delta\left(\sum_{e'' \in E_\Gamma} \Gamma_{ie''} p_{e''} \right) \tag{5.20b}$$

due to the momentum conserving delta functions. We will refer to (5.20a) as the *holomorphic-* and to (5.20b) as the *antiholomorphic momentum conservation identity*. By canceling the momenta from the numerators, (5.20) can be expressed entirely as a manipulation of the decorations of the graph and are therefore identities between MGFs,

$$0 = \sum_{e\in E_\Gamma} \Gamma_{je} \mathcal{C}_{\Gamma_{a_e \to a_e - 1}}, \qquad 0 = \sum_{e\in E_\Gamma} \Gamma_{je} \mathcal{C}_{\Gamma_{b_e \to b_e - 1}}, \quad \forall j \in V_\Gamma. \tag{5.21}$$

If we had chosen a seed with $|A| + |B|$ even, the resulting MGFs would have all vanished trivially. Note that exchanging the sums over e' and the p_{E} in (5.20) required all sums to be convergent.

In the integral representation (2.20), the momentum conservation identities (5.20) correspond to integration-by-parts identities w.r.t. the puncture positions. To see this, note that due to (2.18),[5]

$$\partial_z C^{(a,b)}(z) = -\frac{\pi}{\tau_2} C^{(a,b-1)}(z) \quad \partial_{\bar z} C^{(a,b)}(z) = \frac{\pi}{\tau_2} C^{(a-1,b)}(z). \tag{5.22}$$

If the integrand in (2.20) has no poles, the integral over the total derivative w.r.t. z_j for each $j \in V_\Gamma$ vanishes and we have

$$0 = \sum_{e'\in E_\Gamma} \Gamma_{je'} \int \mathrm{d}\mu_{n-1}\, C^{(a_{e'}, b_{e'}-1)}(z_{e'}) \prod_{\substack{e\in E_\Gamma \\ e\neq e'}} C^{(a_e,b_e)}(z_e) \tag{5.23a}$$

$$0 = \sum_{e'\in E_\Gamma} \Gamma_{je'} \int \mathrm{d}\mu_{n-1}\, C^{(a_{e'}-1, b_{e'})}(z_{e'}) \prod_{\substack{e\in E_\Gamma \\ e\neq e'}} C^{(a_e,b_e)}(z_e), \tag{5.23b}$$

agreeing with (5.21).

For dihedral graphs, the identities (5.21) for both vertices are identical and can be written as

$$0 = \sum_{i=1}^{R} \mathcal{C}\begin{bmatrix} A - S_i \\ B \end{bmatrix} = \sum_{i=1}^{R} \mathcal{C}\begin{bmatrix} A \\ B - S_i \end{bmatrix}, \tag{5.24}$$

the jth component of the row vector S_i is δ_{ij}. For trihedral MGFs, the momentum conservation identities involve two out of the three blocks and are given by

$$0 = \sum_{i=1}^{R_1} \mathcal{C}\left[\begin{smallmatrix} A_1 - S_i \\ B_1 \end{smallmatrix}\middle|\begin{smallmatrix} A_2 \\ B_2 \end{smallmatrix}\middle|\begin{smallmatrix} A_3 \\ B_3 \end{smallmatrix}\right] - \sum_{i=1}^{R_2} \mathcal{C}\left[\begin{smallmatrix} A_1 \\ B_1 \end{smallmatrix}\middle|\begin{smallmatrix} A_2 - S_i \\ B_2 \end{smallmatrix}\middle|\begin{smallmatrix} A_3 \\ B_3 \end{smallmatrix}\right] \tag{5.25}$$

and similarly for the complex conjugated identities. For box graphs, we have

$$0 = \sum_{i=1}^{R_1} \mathcal{C}\left[\begin{smallmatrix} A_1 - S_i \\ B_1 \end{smallmatrix}\middle|\begin{smallmatrix} A_2 \\ B_2 \end{smallmatrix}\middle|\begin{smallmatrix} A_3 \\ B_3 \end{smallmatrix}\middle|\begin{smallmatrix} A_4 \\ B_4 \end{smallmatrix}\right] - \sum_{i=1}^{R_2} \mathcal{C}\left[\begin{smallmatrix} A_1 \\ B_1 \end{smallmatrix}\middle|\begin{smallmatrix} A_2 - S_i \\ B_2 \end{smallmatrix}\middle|\begin{smallmatrix} A_3 \\ B_3 \end{smallmatrix}\middle|\begin{smallmatrix} A_4 \\ B_4 \end{smallmatrix}\right] \quad \text{and c.c.} \tag{5.26}$$

[5] The $a = 1, b = 0$ case of $\partial_{\bar z} C^{(a,b)}$ is compatible with (2.13) upon using (5.33) below.

For kite graphs, we have to distinguish the cases in which the momentum conservation of vertex 2 or 4 is used, yielding

$$0 = \sum_{i=1}^{R_1} \mathcal{C}\left[\begin{smallmatrix} A_1 - S_i \\ B_1 \end{smallmatrix} \middle| \begin{smallmatrix} A_2 \\ B_2 \end{smallmatrix} \middle\| \begin{smallmatrix} A_3 \\ B_3 \end{smallmatrix} \middle| \begin{smallmatrix} A_4 \\ B_4 \end{smallmatrix} \middle\| \begin{smallmatrix} A_5 \\ B_5 \end{smallmatrix}\right] - \sum_{i=1}^{R_2} \mathcal{C}\left[\begin{smallmatrix} A_1 \\ B_1 \end{smallmatrix} \middle| \begin{smallmatrix} A_2 - S_i \\ B_2 \end{smallmatrix} \middle\| \begin{smallmatrix} A_3 \\ B_3 \end{smallmatrix} \middle| \begin{smallmatrix} A_4 \\ B_4 \end{smallmatrix} \middle\| \begin{smallmatrix} A_5 \\ B_5 \end{smallmatrix}\right] \quad \text{and c.c.} \tag{5.27}$$

and the case in which the momentum conservation of vertex 1 or 3 is used, resulting in the identity

$$\begin{aligned} 0 = &\sum_{i=1}^{R_1} \mathcal{C}\left[\begin{smallmatrix} A_1 - S_i \\ B_1 \end{smallmatrix} \middle| \begin{smallmatrix} A_2 \\ B_2 \end{smallmatrix} \middle\| \begin{smallmatrix} A_3 \\ B_3 \end{smallmatrix} \middle| \begin{smallmatrix} A_4 \\ B_4 \end{smallmatrix} \middle\| \begin{smallmatrix} A_5 \\ B_5 \end{smallmatrix}\right] + \sum_{i=1}^{R_4} \mathcal{C}\left[\begin{smallmatrix} A_1 \\ B_1 \end{smallmatrix} \middle| \begin{smallmatrix} A_2 \\ B_2 \end{smallmatrix} \middle\| \begin{smallmatrix} A_3 - S_i \\ B_3 \end{smallmatrix} \middle| \begin{smallmatrix} A_4 \\ B_4 \end{smallmatrix} \middle\| \begin{smallmatrix} A_5 \\ B_5 \end{smallmatrix}\right] \\ &+ \sum_{i=1}^{R_5} \mathcal{C}\left[\begin{smallmatrix} A_1 \\ B_1 \end{smallmatrix} \middle| \begin{smallmatrix} A_2 \\ B_2 \end{smallmatrix} \middle\| \begin{smallmatrix} A_3 \\ B_3 \end{smallmatrix} \middle| \begin{smallmatrix} A_4 \\ B_4 \end{smallmatrix} \middle\| \begin{smallmatrix} A_5 - S_i \\ B_5 \end{smallmatrix}\right] \quad \text{and c.c.} \end{aligned} \tag{5.28}$$

The topology of tetrahedral graphs is completely symmetric, hence the momentum conservation identity for vertex 2,

$$0 = \sum_{i=1}^{R_1} \mathcal{C}\left[\begin{smallmatrix} \frac{A_1 - S_i}{B_1} \\ \frac{A_4}{B_4} \end{smallmatrix} \middle\| \begin{smallmatrix} \frac{A_2}{B_2} \\ \frac{A_5}{B_5} \end{smallmatrix} \middle\| \begin{smallmatrix} \frac{A_3}{B_3} \\ \frac{A_6}{B_6} \end{smallmatrix}\right] + \sum_{i=1}^{R_2} \mathcal{C}\left[\begin{smallmatrix} \frac{A_1}{B_1} \\ \frac{A_4}{B_4} \end{smallmatrix} \middle\| \begin{smallmatrix} \frac{A_2 - S_i}{B_2} \\ \frac{A_5}{B_5} \end{smallmatrix} \middle\| \begin{smallmatrix} \frac{A_3}{B_3} \\ \frac{A_6}{B_6} \end{smallmatrix}\right] + \sum_{i=1}^{R_3} \mathcal{C}\left[\begin{smallmatrix} \frac{A_1}{B_1} \\ \frac{A_4}{B_4} \end{smallmatrix} \middle\| \begin{smallmatrix} \frac{A_2}{B_2} \\ \frac{A_5}{B_5} \end{smallmatrix} \middle\| \begin{smallmatrix} \frac{A_3 - S_i}{B_3} \\ \frac{A_6}{B_6} \end{smallmatrix}\right] \tag{5.29}$$

and its complex conjugate are related to those of all other vertices by the transformations (5.3) and (5.4).

In the `ModularGraphForms` package, momentum conservation for dihedral and trihedral graphs is implemented in the functions `DiHolMomConsId` and `TriHolMomConsId` and their antiholomorphic versions `DiAHolMomConsId` and `TriAHolMomConsId`. In the dihedral case (5.24), the function `DiHolMomConsId` takes the seed as its only argument and we have e.g.

```
In[25]:= DiHolMomConsId[c[1 1 2]]

Out[25]= C[0 1 2] + C[1 0 2] + C[1 1 1] == 0 .
         1 1 1     1 1 1     1 1 1
```

For trihedral momentum conservation (5.25), we have to specify which of the three vertices we use and hence which pair of blocks has its labels changed. The list of these blocks is passed as a second argument to `TriHolMomConsId`, e.g.

```
In[26]:= TriHolMomConsId[c[1 2, 1 3, 1 4], {2,3}]

Out[26]= C[1 2|0 3|1 4] + C[1 2|1 2|1 4] - C[1 2|1 3|0 4] - C[1 2|1 3|1 3] == 0 .
         1 1|1 1|1 1     1 1|1 1|1 1     1 1|1 1|1 1     1 1|1 1|1 1
```

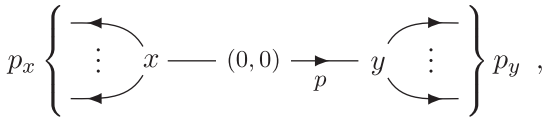
Note that the functions discussed here do not apply `CSort` to the resulting equation, so that it is more transparent which exponents were lowered. E.g. `Out [25]` simplifies to

```
In[27]:= CSort[Out[25]]

Out[27]= 2 C[0 1 2; 1 1 1] + C[1 1 1; 1 1 1] == 0 .
```

5.4. Factorization

Consider a MGF with a $(0,0)$-edge. In this case, the graph factorizes [8]. To see this, consider two vertices x and y and an edge from x to y with momentum p and decoration $(0,0)$. Furthermore assume that all other edges connected to x are directed away from x and have momentum sum p_x and all other edges connected to y are directed away from y and have momentum sum p_y,



(5.30)

where the $(0,0)$-edge is not necessarily the only edge between x and y. In the sum representation, the momentum p only appears in the momentum-conserving delta functions for the vertices x and y. Isolating this contribution, we get

$$\begin{aligned}\sum_p{}' \delta(p_x+p)\delta(p_y-p) &= \sum_p \delta(p_x+p)\delta(p_y-p) - \delta(p_x)\delta(p_y)\\ &= \delta(p_x+p_y) - \delta(p_x)\delta(p_y),\end{aligned} \tag{5.31}$$

where we added $p=0$ to the sum to evaluate the deltas. When (5.31) appears in the nested lattice sum of an MGF, the first term gives rise to the original MGF with the vertices x and y identified, whereas the second term can be associated to the original MGF with the $(0,0)$-edge removed. Schematically, if the edge e between vertices x and y carries decoration $(0,0)$, we have

$$\mathcal{C}_{\Gamma_{a_e=b_e=0}} = \mathcal{C}_{\Gamma_{x=y}} - \mathcal{C}_{\Gamma\setminus e}. \tag{5.32}$$

If the vertices x and y are connected by more edges than just e, these will factorize as one-loop graphs in the first term of (5.32).

In the integral representation, a $(0,0)$-edge is represented by a factor $C^{(0,0)}(z)$ in the integrand, which as special case of (2.18) can be simplified to

$$C^{(0,0)}(z) = \sum_{m,n\in\mathbb{Z}} e^{2\pi \mathrm{i}(mv-nu)} - 1 = \delta(v)\delta(u) - 1 = \tau_2\delta^{(2)}(z,\bar{z}) - 1, \tag{5.33}$$

where we used (2.3). Note that (5.33) is not the $a=0$ case of (2.19), since $f^{(0)}(z)=1$, but is implied by the $a=1, b=0$ case of (5.22) and (2.13). The interpretation of (5.33) is exactly as in the sum representation: the delta identifies the two vertices connected by the $(0,0)$ edge and in the second term the $(0,0)$ edge is removed. In this way, we get again (5.32).

For dihedral MGFs, (5.32) implies[6]

$$\mathcal{C}\begin{bmatrix}0 & A\\ 0 & B\end{bmatrix} = \prod_{j=1}^{R}\mathcal{C}\begin{bmatrix}a_j & 0\\ b_j & 0\end{bmatrix} - \mathcal{C}\begin{bmatrix}A\\ B\end{bmatrix} \tag{5.34}$$

for higher-point graphs we have

$$\mathcal{C}\left[\begin{smallmatrix}0 & A_1\\ 0 & B_1\end{smallmatrix}\middle|\begin{smallmatrix}A_2\\ B_2\end{smallmatrix}\middle|\begin{smallmatrix}A_3\\ B_3\end{smallmatrix}\right] = (-1)^{|2|}\,\mathcal{C}\left[\begin{smallmatrix}A_2 & A_3\\ B_2 & B_3\end{smallmatrix}\right]\prod_{i=1}^{R_1}\mathcal{C}\left[\begin{smallmatrix}a_1^{(i)} & 0\\ b_1^{(i)} & 0\end{smallmatrix}\right] - \mathcal{C}\left[\begin{smallmatrix}A_1\\ B_1\end{smallmatrix}\middle|\begin{smallmatrix}A_2\\ B_2\end{smallmatrix}\middle|\begin{smallmatrix}A_3\\ B_3\end{smallmatrix}\right] \tag{5.35}$$

$$\mathcal{C}\left[\begin{smallmatrix}0 & A_1\\ 0 & B_1\end{smallmatrix}\middle|\begin{smallmatrix}A_2\\ B_2\end{smallmatrix}\middle|\begin{smallmatrix}A_3\\ B_3\end{smallmatrix}\middle|\begin{smallmatrix}A_4\\ B_4\end{smallmatrix}\right] = \mathcal{C}\left[\begin{smallmatrix}A_2\\ B_2\end{smallmatrix}\middle|\begin{smallmatrix}A_3\\ B_3\end{smallmatrix}\middle|\begin{smallmatrix}A_4\\ B_4\end{smallmatrix}\right]\prod_{i=1}^{R_1}\mathcal{C}\left[\begin{smallmatrix}a_1^{(i)} & 0\\ b_1^{(i)} & 0\end{smallmatrix}\right] - \mathcal{C}\left[\begin{smallmatrix}A_1\\ B_1\end{smallmatrix}\middle|\begin{smallmatrix}A_2\\ B_2\end{smallmatrix}\middle|\begin{smallmatrix}A_3\\ B_3\end{smallmatrix}\middle|\begin{smallmatrix}A_4\\ B_4\end{smallmatrix}\right] \tag{5.36}$$

$$\begin{aligned}\mathcal{C}\left[\begin{smallmatrix}0 & A_1\\ 0 & B_1\end{smallmatrix}\middle|\begin{smallmatrix}A_2\\ B_2\end{smallmatrix}\middle\|\begin{smallmatrix}A_3\\ B_3\end{smallmatrix}\middle|\begin{smallmatrix}A_4\\ B_4\end{smallmatrix}\middle\|\begin{smallmatrix}A_5\\ B_5\end{smallmatrix}\right] &= (-1)^{|2|+|5|}\,\mathcal{C}\left[\begin{smallmatrix}A_2 & A_5\\ B_2 & B_5\end{smallmatrix}\middle|\begin{smallmatrix}A_3\\ B_3\end{smallmatrix}\middle|\begin{smallmatrix}A_4\\ B_4\end{smallmatrix}\right]\prod_{i=1}^{R_1}\mathcal{C}\left[\begin{smallmatrix}a_1^{(i)} & 0\\ b_1^{(i)} & 0\end{smallmatrix}\right]\\ &\quad - \mathcal{C}\left[\begin{smallmatrix}A_1\\ B_1\end{smallmatrix}\middle|\begin{smallmatrix}A_2\\ B_2\end{smallmatrix}\middle\|\begin{smallmatrix}A_3\\ B_3\end{smallmatrix}\middle|\begin{smallmatrix}A_4\\ B_4\end{smallmatrix}\middle\|\begin{smallmatrix}A_5\\ B_5\end{smallmatrix}\right]\end{aligned} \tag{5.37}$$

$$\begin{aligned}\mathcal{C}\left[\begin{smallmatrix}A_1\\ B_1\end{smallmatrix}\middle|\begin{smallmatrix}A_2\\ B_2\end{smallmatrix}\middle\|\begin{smallmatrix}A_3\\ B_3\end{smallmatrix}\middle|\begin{smallmatrix}A_4\\ B_4\end{smallmatrix}\middle\|\begin{smallmatrix}0 & A_5\\ 0 & B_5\end{smallmatrix}\right] &= (-1)^{|1|+|3|}\,\mathcal{C}\left[\begin{smallmatrix}A_1 & A_2\\ B_1 & B_2\end{smallmatrix}\right]\mathcal{C}\left[\begin{smallmatrix}A_3 & A_4\\ B_3 & B_4\end{smallmatrix}\right]\prod_{i=1}^{R_5}\mathcal{C}\left[\begin{smallmatrix}a_5^{(i)} & 0\\ b_5^{(i)} & 0\end{smallmatrix}\right]\\ &\quad - \mathcal{C}\left[\begin{smallmatrix}A_1\\ B_1\end{smallmatrix}\middle|\begin{smallmatrix}A_2\\ B_2\end{smallmatrix}\middle\|\begin{smallmatrix}A_3\\ B_3\end{smallmatrix}\middle|\begin{smallmatrix}A_4\\ B_4\end{smallmatrix}\middle\|\begin{smallmatrix}A_5\\ B_5\end{smallmatrix}\right]\end{aligned} \tag{5.38}$$

$$\mathcal{C}\left[\begin{smallmatrix}0 & A_1\\ 0 & B_1\\ \hline A_4\\ B_4\end{smallmatrix}\middle\|\begin{smallmatrix}A_2\\ B_2\\ \hline A_5\\ B_5\end{smallmatrix}\middle\|\begin{smallmatrix}A_3\\ B_3\\ \hline A_6\\ B_6\end{smallmatrix}\right] = (-1)^{|2|}\,\mathcal{C}\left[\begin{smallmatrix}A_2 & A_6\\ B_2 & B_6\end{smallmatrix}\middle|\begin{smallmatrix}A_3 & A_5\\ B_3 & B_5\end{smallmatrix}\middle|\begin{smallmatrix}A_4\\ B_4\end{smallmatrix}\right]\prod_{i=1}^{R_1}\mathcal{C}\left[\begin{smallmatrix}a_1^{(i)} & 0\\ b_1^{(i)} & 0\end{smallmatrix}\right] - \mathcal{C}\left[\begin{smallmatrix}A_1\\ B_1\\ \hline A_4\\ B_4\end{smallmatrix}\middle\|\begin{smallmatrix}A_2\\ B_2\\ \hline A_5\\ B_5\end{smallmatrix}\middle\|\begin{smallmatrix}A_3\\ B_3\\ \hline A_6\\ B_6\end{smallmatrix}\right], \tag{5.39}$$

where we used the abbreviation $|i| = |A_i| + |\mathrm{B}_i|$ as above. Note that the rhs have one vertex less in the first term and one loop order less in the second term and hence (5.34) to (5.39) are powerful identities to simplify MGFs. Together with the momentum conservation identities from section 5.3, these identities form the backbone of all the simplifications we will carry out in the following.

In the `ModularGraphForms` package, factorization of $(0,0)$-edges for dihedral and trihedral graph is also done by the functions `DiCSimplify` and `TriCSimplify`. E.g. in the trihedral case (5.35), we have

```
In[28]:= TriCSimplify[c[0 2 1, 1 2, 1 4]]

Out[28]= - c[1 0] c[2 0] c[2 2 1 4] - c[1 2|1 4|2 2] .
          2 0   1 0   1 2 1 1   2 1|1 1|1 2
```

If several $(0,0)$-edges are present, the factorization is repeated until no more $(0,0)$-edges in the respective topology appear. E.g. we have

```
In[29]:= TriCSimplify[c[0 2 1, 0 2 2, 1 4]]

Out[29]= c[2 0] c[2 0] c[1 2 1 4] - c[1 0] c[2 0] c[0 2 2 1 4] + c[1 2|1 4|2 2] ,
          1 0   2 0   2 1 1 1     2 0   1 0   0 1 2 1 1     2 1|1 1|1 2
```

[6] For one-loop graphs $\mathcal{C}\begin{bmatrix}a & 0\\ b & 0\end{bmatrix}$, (5.34) is trivial upon using (5.9).

where the remaining dihedral factorization can be preformed by applying `DiCSimplify`.

5.5. Taking derivatives

On top of momentum conservation and factorization, another way to obtain new identities for MGFs is by taking derivatives of known identities w.r.t. τ.

In order to take derivatives of modular functions and forms, we use the *Maaß operators* [45]

$$\nabla^{(a)} = 2\mathrm{i}\tau_2\partial_\tau + a \quad \overline{\nabla}^{(b)} = -2\mathrm{i}\tau_2\partial_{\bar\tau} + b. \tag{5.40}$$

When these act on modular forms of weight (a, b) they transform them into modular forms of shifted modular weight according to

$$\nabla^{(a)}:\ (a,b)\to(a+1,b-1) \quad \overline{\nabla}^{(b)}:\ (a,b)\to(a-1,b+1). \tag{5.41}$$

These operators satisfy a product rule when acting on a product fg of modular forms f and g of holomorphic modular weights a and a', respectively,

$$\nabla^{(a+a')}(fg) = (\nabla^{(a)}f)g + f(\nabla^{(a')}g) \quad \text{and c.c.} \tag{5.42}$$

For later convenience, we introduce the notation

$$\nabla^{(a)^n} = \nabla^{(a+n)}\nabla^{(a+n-1)}\ldots\nabla^{(a)} \quad \overline{\nabla}^{(b)^n} = \overline{\nabla}^{(b+n)}\overline{\nabla}^{(b+n-1)}\ldots\overline{\nabla}^{(b)} \tag{5.43}$$

for higher derivatives. We will also use the operators

$$\nabla_0 = \tau_2\nabla^{(0)} = 2\mathrm{i}\tau_2^2\partial_\tau \quad \overline{\nabla}_0 = \tau_2\overline{\nabla}^{(0)} = -2\mathrm{i}\tau_2^2\partial_{\bar\tau}, \tag{5.44}$$

which act on modular forms of vanishing holomorphic and antiholomorphic modular weight,

$$\nabla_0:\ (0,b)\to(0,b-2) \quad \overline{\nabla}_0:\ (a,0)\to(a-2,0). \tag{5.45}$$

Using these operators, we will now discuss derivatives of identities between MGFs. Note that since the Maaß operators change the modular weight, one obtains an identity between MGFs of different weights.

Consider the action of $\nabla^{(|A|)}$ and $\overline{\nabla}^{(|B|)}$ and on an MGF of weight $(|A|, |B|)$ in its lattice sum representation (2.23). Using the product rule (5.42) and

$$\nabla^{(a)}\left(\frac{1}{p^a}\right) = a\,\frac{1}{p^{a+1}\bar p^{-1}} \qquad \overline{\nabla}^{(b)}\left(\frac{1}{\bar p^b}\right) = b\,\frac{1}{p^{-1}\bar p^{b+1}}, \tag{5.46}$$

the derivatives are given by [8]

$$\begin{aligned}\nabla^{(|A|)}\mathcal{C}_\Gamma &= \sum_{e\in E_\Gamma} a_e \mathcal{C}_{\Gamma_{(a_e,b_e)\to(a_e+1,b_e-1)}}\\ \overline{\nabla}^{(|B|)}\mathcal{C}_\Gamma &= \sum_{e\in E_\Gamma} b_e \mathcal{C}_{\Gamma_{(a_e,b_e)\to(a_e-1,b_e+1)}}.\end{aligned} \tag{5.47}$$

In the integral representation, $\nabla^{(|A|)}$ and $\overline{\nabla}^{(|B|)}$ act on the Jacobi forms $C^{(a,b)}(z,\tau)$ given in (2.18). According to (5.46), we have

$$\nabla^{(a)}C^{(a,b)}(z,\tau) = a\,C^{(a+1,b-1)}(z,\tau) \quad \overline{\nabla}^{(b)}C^{(a,b)}(z,\tau) = b\,C^{(a-1,b+1)}(z,\tau) \tag{5.48}$$

and using this together with the product rule (5.42), we obtain again (5.47).

For a dihedral MGF, (5.47) implies [8]

$$\nabla^{(|A|)}\mathcal{C}\begin{bmatrix}A\\B\end{bmatrix} = \sum_{i=1}^{R} a_i\,\mathcal{C}\begin{bmatrix}A+S_i\\B-S_i\end{bmatrix} \quad \nabla^{(|B|)}\,\mathcal{C}\begin{bmatrix}A\\B\end{bmatrix} = \sum_{i=1}^{R} b_i\mathcal{C}\begin{bmatrix}A-S_i\\B+S_i\end{bmatrix}, \tag{5.49}$$

where the jth component of S_i is δ_{ij} as above. A special case of (5.49) is the important relation

$$\nabla_0^n \mathrm{E}_k = \frac{\tau_2^{k+n}}{\pi^k}\frac{(k+n-1)!}{(k-1)!}\mathcal{C}\begin{bmatrix}k+n & 0\\ k-n & 0\end{bmatrix}, \tag{5.50}$$

where ∇_0 is defined in (5.44). Since (5.47) does not depend on the topology of the graph, the higher-point versions of (5.49) are completely analogous, so for trihedral graphs, we have e.g. [8]

$$\begin{aligned}\nabla^{(|A|)}\,\mathcal{C}\left[\begin{smallmatrix}A_1\\B_1\end{smallmatrix}\middle|\begin{smallmatrix}A_2\\B_2\end{smallmatrix}\middle|\begin{smallmatrix}A_3\\B_3\end{smallmatrix}\right] = &\sum_{i=1}^{R_1} a_1^{(i)}\,\mathcal{C}\left[\begin{smallmatrix}A_1+S_i\\B_1-S_i\end{smallmatrix}\middle|\begin{smallmatrix}A_2\\B_2\end{smallmatrix}\middle|\begin{smallmatrix}A_3\\B_3\end{smallmatrix}\right] + \sum_{i=1}^{R_2} a_2^{(i)}\,\mathcal{C}\left[\begin{smallmatrix}A_1\\B_1\end{smallmatrix}\middle|\begin{smallmatrix}A_2+S_i\\B_2-S_i\end{smallmatrix}\middle|\begin{smallmatrix}A_3\\B_3\end{smallmatrix}\right]\\ &+\sum_{i=1}^{R_3} a_3^{(i)}\,\mathcal{C}\left[\begin{smallmatrix}A_1\\B_1\end{smallmatrix}\middle|\begin{smallmatrix}A_2\\B_2\end{smallmatrix}\middle|\begin{smallmatrix}A_3+S_i\\B_3-S_i\end{smallmatrix}\right] \quad \text{and c.c.}\,,\end{aligned} \tag{5.51}$$

where in the complex conjugation, we swap all a and b labels everywhere and replace $S_i \to -S_i$. Similar identities hold for all four-point graphs.

When taking the Cauchy–Riemann derivative of a holomorphic Eisenstein series, one obtains

$$\nabla^{(2k)}\mathrm{G}_{2k} = 2k\mathcal{C}\begin{bmatrix}2k+1 & 0\\ -1 & 0\end{bmatrix}, \quad k \geqslant 2, \tag{5.52}$$

which cannot be simplified further with the methods presented so far. However, the $\bar{\tau}$-derivative of the weight $(2k+2,0)$ modular form

$$\frac{\pi}{\tau_2}\mathcal{C}\begin{bmatrix}2k+1 & 0\\ -1 & 0\end{bmatrix} + \mathrm{G}_{2k}\widehat{\mathrm{G}}_2, \quad k \geqslant 2 \tag{5.53}$$

vanishes, and hence it can be expanded in the ring of holomorphic Eisenstein series. To this end, we calculate the q expansion ($q = \mathrm{E}^{2\pi i\tau}$)

$$\frac{1}{2k}\nabla^{(2k)}\mathrm{G}_{2k} = 2\zeta_{2k} - \frac{4\zeta_{2k}}{B_{2k}}\sum_{n=1}^{\infty}\sigma_{2k-1}(n)(2k-4\pi n\tau_2)q^n, \quad k \geqslant 1, \tag{5.54}$$

by taking the Cauchy–Riemann derivative of the q expansion of G_{2k}. Now, by comparing a finite number of terms, we can expand (5.52) in the ring of G_4 and G_6. Since for low weights this ring is one-dimensional, we can give a closed formula in these cases,

$$\mathcal{C}\begin{bmatrix}2k+1 & 0\\ -1 & 0\end{bmatrix} = \frac{\tau_2}{\pi}\left(\frac{2\zeta_2\zeta_{2k}}{\zeta_{2k+2}}\mathrm{G}_{2k+2} - \mathrm{G}_{2k}\widehat{\mathrm{G}}_2\right), \quad k = 2,3,4. \tag{5.55}$$

For the non-holomorphic but modular version $\widehat{\mathrm{G}}_2 = \mathrm{G}_2 - \frac{\pi}{\tau_2}$ of G_2, we obtain

$$\nabla^{(2)}\widehat{\mathrm{G}}_2 = 2\mathcal{C}\begin{bmatrix} 3 & 0 \\ -1 & 0 \end{bmatrix} = \frac{\tau_2}{\pi}\left(5\mathrm{G}_4 - \widehat{\mathrm{G}}_2^2\right), \tag{5.56}$$

as can be verified by explicitly comparing the q expansions term by term. Note that (5.56) and (5.55) for $k = 2, 3$ are equivalent to the classic Ramanujan identities

$$q\frac{\mathrm{dG}_2}{\mathrm{d}q} = \frac{\mathrm{G}_2^2 - 5\mathrm{G}_4}{4\pi^2} \tag{5.57a}$$

$$q\frac{\mathrm{dG}_4}{\mathrm{d}q} = \frac{2\mathrm{G}_2\mathrm{G}_4 - 7\mathrm{G}_6}{2\pi^2} \tag{5.57b}$$

$$q\frac{\mathrm{dG}_6}{\mathrm{d}q} = \frac{21\mathrm{G}_2\mathrm{G}_6 - 30\mathrm{G}_4^2}{14\pi^2}. \tag{5.57c}$$

Since the expressions above allow to write the derivative of any holomorphic Eisenstein series back into a polynomial in holomorphic Eisenstein series, we can iterate these expressions and simplify arbitrarily high derivatives of holomorphic Eisenstein series. E.g. we have

$$\mathcal{C}\begin{bmatrix} 4 & 0 \\ -2 & 0 \end{bmatrix} = \frac{1}{6}\nabla^{(3)}\nabla^{(2)}\widehat{\mathrm{G}}_2 = \left(\frac{\tau_2}{\pi}\right)^2\left(\frac{35}{3}\mathrm{G}_6 - 5\mathrm{G}_4\widehat{\mathrm{G}}_2 + \frac{1}{3}\widehat{\mathrm{G}}_2^3\right) \tag{5.58a}$$

$$\mathcal{C}\begin{bmatrix} 6 & 0 \\ -2 & 0 \end{bmatrix} = \frac{1}{20}\nabla^{(5)}\nabla^{(4)}\mathrm{G}_4 = \left(\frac{\tau_2}{\pi}\right)^2\left(\mathrm{G}_4\widehat{\mathrm{G}}_2^2 - 7\mathrm{G}_6\widehat{\mathrm{G}}_2 + 5\mathrm{G}_4^2\right). \tag{5.58b}$$

In the `ModularGraphForms` package, the Cauchy–Riemann derivatives (5.47) are implemented in the function `CHolCR` for the holomorphic case and `CAHolCR` for the anti-holomorphic case. For clarity, the result is returned as it comes out of the action (5.47) of $\nabla^{(a)}$ hence, to obtain the derivative in canonical representation, we have to apply `CSort`, e.g.

```
In[30]:= CHolCR[c[[1 1, 1 1, 1 1, 1 1, 1 1]]]//CSort

Out[30]= 8 C[[1 1|1 1||1 1|1 2||1 1]] + 2 C[[1 1|1 1||1 1|1 1||1 2]] .
```

The functions `CHolCR` and `CAHolCR` can also be used to calculate derivatives of holomorphic Eisenstein series,

```
In[31]:= CHolCR[g[4]]
         CHolCR[%]
         CHolCR[gHat[2]]

Out[31]= 4 C[[5 0]]

Out[32]= 20 C[[6 0]]

Out[33]= 2 C[[3 0]] .
```

The simplifications of these expressions by means of the Ramanujan identities (5.55), (5.56) and (5.58) and higher-weight generalizations is performed by the function `DiCSimplify`, if

the option `basisExpandG` is set to `True`, e.g.

```
In[34]:= DiCSimplify[Out[31], basisExpandG→True]
         DiCSimplify[Out[32], basisExpandG→True]
         DiCSimplify[Out[33], basisExpandG→True]
```

$$\text{Out[34]}= \frac{14\,\mathtt{G}_6\,\tau_2}{\pi} - \frac{4\,\mathtt{G}_4\,\hat{\mathtt{G}}_2\,\tau_2}{\pi}$$

$$\text{Out[35]}= \frac{100\,\mathtt{G}_4^2\,\tau_2^2}{\pi^2} - \frac{140\,\mathtt{G}_6\,\hat{\mathtt{G}}_2\,\tau_2^2}{\pi^2} + \frac{20\,\mathtt{G}_4\,\hat{\mathtt{G}}_2^2\,\tau_2^2}{\pi^2}$$

$$\text{Out[36]}= \frac{5\,\mathtt{G}_4\,\tau_2}{\pi} - \frac{\hat{\mathtt{G}}_2^2\,\tau_2}{\pi} \quad .$$

Using the techniques outlined above, `DiCSimplify` can decompose any MGF of the form $\mathcal{C}\begin{bmatrix} k & 0 \\ -n & 0 \end{bmatrix}$ or $\mathcal{C}\begin{bmatrix} -n & 0 \\ k & 0 \end{bmatrix}$ with $k, n \in \mathbb{N}_0$ and $k > n$ into the ring of holomorphic Eisenstein series and powers of $\widehat{\mathrm{G}}_2$ and $\frac{\tau_2}{\pi}$ (or c.c.).

6. Holomorphic subgraph reduction

Using the relatively straightforward techniques discussed in the previous section, many identities between MGFs can be derived. However, an important class of identities is still missing to decompose all relevant MGFs into the basis to be presented in section 9, namely HSR. In this section, we will review HSR as it was introduced first for dihedral graphs [8] and the extension of this technique to higher-point graphs [19].[7]

The basic idea behind HSR is the following: if an MGF has a closed subgraph (i.e. a subgraph which forms a loop) in which all edges have only holomorphic momenta (i.e. the decorations are all of the form $(a, 0)$), then one can apply the partial-fraction decomposition

$$\frac{1}{p^a(q-p)^b} = \sum_{k=1}^{a} \binom{a+b-k-1}{a-k} \frac{1}{p^k q^{a+b-k}} + \sum_{k=1}^{b} \binom{a+b-k-1}{b-k} \times \frac{1}{q^{a+b-k}(q-p)^k} \tag{6.1}$$

to the summand and perform the sum over the loop momentum explicitly. Since for certain values of a, b this sum is only conditionally convergent, it has to be supplied with a summation prescription, which we will choose to be *Eisenstein summation*, to be defined below in (6.4). This procedure however breaks the modular transformation properties at the level of the individual contributions. As shown in [8] for two-point graphs and in [19] for general graphs, the terms with incorrect modular properties cancel out in the final expression and one obtains a decomposition of the original MGF into terms which all have at least one loop order less.

[7] In the references, a different convention for MGFs was used, which differs from the one used here by factors of τ_2 and π.

As an example, consider the trihedral graph $\mathcal{C}\left[\begin{array}{c|c|c} 1\,2 & 1\,2 & 1\,2 \\ 0\,1 & 0\,1 & 0\,1 \end{array}\right]$, which has a closed three-point holomorphic subgraph. Using the techniques discussed in this section, it can be decomposed into

$$\mathcal{C}\left[\begin{smallmatrix} 1 & 2 \\ 0 & 1 \end{smallmatrix}\middle|\begin{smallmatrix} 1 & 2 \\ 0 & 1 \end{smallmatrix}\middle|\begin{smallmatrix} 1 & 2 \\ 0 & 1 \end{smallmatrix}\right] = 6\,\mathcal{C}\left[\begin{smallmatrix} 2 \\ 1 \end{smallmatrix}\middle|\begin{smallmatrix} 1 & 2 \\ 0 & 1 \end{smallmatrix}\middle|\begin{smallmatrix} 2 & 2 \\ 0 & 1 \end{smallmatrix}\right] - 3\,\mathcal{C}\left[\begin{smallmatrix} 2 & 3 & 4 \\ 1 & 0 & 2 \end{smallmatrix}\right] + 3\,\mathcal{C}\left[\begin{smallmatrix} 1 & 2 & 4 \\ 0 & 1 & 2 \end{smallmatrix}\right]\widehat{\mathrm{G}}_2 + \frac{\pi}{\tau_2}\,\mathcal{C}\left[\begin{smallmatrix} 2 & 2 & 4 \\ -1 & 1 & 2 \end{smallmatrix}\right] - 2\frac{\pi}{\tau_2}\,\mathcal{C}\left[\begin{smallmatrix} 2 \\ 1 \end{smallmatrix}\middle|\begin{smallmatrix} 1 & 2 \\ -1 & 1 \end{smallmatrix}\middle|\begin{smallmatrix} 1 & 2 \\ 0 & 1 \end{smallmatrix}\right]. \tag{6.2}$$

6.1. Dihedral holomorphic subgraph reduction

Dihedral graphs with a holomorphic subgraph have the form $\mathcal{C}\begin{bmatrix} a_+ & a_- & A \\ 0 & 0 & B \end{bmatrix}$. The edges with labels $(a_+, 0)$ and $(a_-, 0)$ form the holomorphic subgraph and the sum over the loop momentum associated to this subgraph can be isolated using the partial-fraction decomposition (6.1). This sum takes the form

$$Q_k(p_0) = \sum_{p\neq p_0}{}' \frac{1}{p^k}, \quad k \geqslant 1 \tag{6.3}$$

which is not absolutely convergent for $k = 1, 2$. Using the Eisenstein summation prescription

$$\sum_{p\neq r+s\tau}{}_E f(p) = \lim_{N\to\infty} \sum_{\substack{n=-N \\ n\neq s}}^{N} \left(\lim_{M\to\infty} \sum_{m=-M}^{M} f(m+n\tau) \right) + \lim_{M\to\infty} \sum_{\substack{m=-M \\ m\neq r}}^{M} f(m+s\tau), \tag{6.4}$$

we assign the values

$$Q_1(p_0) = -\frac{1}{p_0} - \frac{\pi}{2\tau_2}(p_0 - \bar{p}_0) \tag{6.5a}$$

$$Q_2(p_0) = -\frac{1}{{p_0}^2} + \widehat{\mathrm{G}}_2 + \frac{\pi}{\tau_2} \tag{6.5b}$$

$$Q_k(p_0) = -\frac{1}{{p_0}^k} + \mathrm{G}_k \quad k \geqslant 3 \tag{6.5c}$$

to the sums (6.3). Note that the terms $\frac{\pi}{\tau_2} p_0$ in (6.5a) and $\frac{\pi}{\tau_2}$ in (6.5b) do not the have modular weight we associate to the corresponding lhs. However, these terms cancel out in the final result

and we obtain [8]

$$\begin{aligned}\mathcal{C}\begin{bmatrix} a_+ & a_- & A \\ 0 & 0 & B \end{bmatrix} = (-1)^{a_+}\mathrm{G}_{a_0}\,\mathcal{C}\begin{bmatrix} A \\ B \end{bmatrix} - \binom{a_0}{a_-}\mathcal{C}\begin{bmatrix} a_0 & A \\ 0 & B \end{bmatrix} \\ + \sum_{k=4}^{a_+}\binom{a_0-1-k}{a_+-k}\mathrm{G}_k\,\mathcal{C}\begin{bmatrix} a_0-k & A \\ 0 & B \end{bmatrix} \\ + \sum_{k=4}^{a_-}\binom{a_0-1-k}{a_--k}\mathrm{G}_k\,\mathcal{C}\begin{bmatrix} a_0-k & A \\ 0 & B \end{bmatrix} \\ + \binom{a_0-2}{a_+-1}\left\{\widehat{\mathrm{G}}_2\,\mathcal{C}\begin{bmatrix} a_0-2 & A \\ 0 & B \end{bmatrix} + \frac{\pi}{\tau_2}\,\mathcal{C}\begin{bmatrix} a_0-1 & A \\ -1 & B \end{bmatrix}\right\}.\end{aligned} \tag{6.6}$$

For instance, the two-loop graph $\mathcal{C}\begin{bmatrix} 1 & 2 & 2 \\ 0 & 0 & 1 \end{bmatrix}$ is decomposed into one-loop graphs by (6.6),

$$\mathcal{C}\begin{bmatrix} 1 & 2 & 2 \\ 0 & 0 & 1 \end{bmatrix} = 3\,\mathcal{C}\begin{bmatrix} 5 & 0 \\ 1 & 0 \end{bmatrix} - \widehat{\mathrm{G}}_2\,\mathcal{C}\begin{bmatrix} 3 & 0 \\ 1 & 0 \end{bmatrix} - \frac{\pi}{\tau_2}\mathrm{G}_4. \tag{6.7}$$

In the `ModularGraphFormsMathematica` package, the dihedral HSR (6.6) is performed by the function `DiCSimplify`. With the default options, `DiCSimplify` also applies all known dihedral basis decompositions to the result and uses momentum conservation to remove negative entries where possible as will be detailed in section 7.1. Both features can be disabled by setting the Boolean options `momSimplify` and `useIds` to `False` (they are `True` by default). Hence, in order to get just the result of the formula (6.6), we can run e.g.

```
In[37]:= DiCSimplify[C[2 2 3 6; 1 2 0 0], momSimplify→False, useIds→False]

Out[37]= -84 C[2 2 9; 1 2 0] + 6 C[2 2 5; 1 2 0] G4 + C[2 2 3; 1 2 0] G6 + 21 C[2 2 7; 1 2 0] Ĝ2 + (21 π C[2 2 8; 1 2 -1])/τ2
```

The function `DiCSimplify` applies the formula (6.6) always to the two leftmost holomorphic columns. It also performs antiholomorphic subgraph reduction by applying the complex conjugate of (6.6) to graphs with a closed antiholomorphic subgraph. In order to deactivate dihedral HSR in `DiCSimplify` or `CSimplify`, one can set the Boolean option `diHSR` to `False` (the default is `True`).

6.2. *Higher-point holomorphic subgraph reduction*

In [19], HSR for higher-point graphs was worked out. Again, we can separate the sum over the loop momentum of the holomorphic subgraph using partial-fraction decomposition. However, as a novelty from three points onward, on top of sums of the form

$$Q_k(p_1,\ldots,p_n) = \sum_{p\neq p_1,\ldots,p_n}{}' \frac{1}{p^k}, \quad k \geqslant 1, \tag{6.8}$$

which are a straightforward generalization of the sums (6.3), also shifted lattice sums of the form

$$\sum_{p\neq p_1,\dots,p_n}{}' \frac{1}{p_i - p} \tag{6.9}$$

appear. Due to a subtlety in the Eisenstein summation prescription, we cannot just shift the expressions on the rhs of (6.5a) and add the additional excluded points. Accounting for this, we have for $k \geqslant 2$

$$Q_2(p_1,\dots,p_n) = \widehat{\mathrm{G}}_2 + \frac{\pi}{\tau_2} - \sum_{i=1}^{n} \frac{1}{p_i^2} \tag{6.10a}$$

$$Q_k(p_1,\dots,p_n) = \mathrm{G}_k - \sum_{i=1}^{n} \frac{1}{p_i^k}, \quad k \geqslant 3. \tag{6.10b}$$

and for $k = 1$, we replace

$$\sum_{p\neq p_1,\dots,p_n}{}' \frac{1}{p} \to Q_1(p_1,\dots,p_n) \sum_{p\neq p_1,\dots,p_n}{}' \frac{1}{p_i - p} \to Q_1(p_i, \underbrace{p_i - p_1,\dots,p_i - p_n}_{\text{omit } p_i - p_i}) \tag{6.11}$$

and set

$$Q_1(p_1,\dots,p_n) = -\sum_{i=1}^{n} \frac{1}{p_i} - \frac{\pi}{(n+1)\tau_2} \sum_{i=1}^{n} (p_i - \bar{p}_i). \tag{6.12}$$

With the expressions (6.10) and (6.12), any MGF with an n-point holomorphic subgraph can be decomposed. In particular, in [19], a closed expression for three-point HSR was derived, cf e.g. (6.2). For this case, we illustrate a general feature of higher-point HSR: since there are several ways in which the partial-fraction decomposition can be done, different expressions for the decomposition can be obtained. For instance the graph $\mathcal{C}\left[\begin{array}{c|c|c} 1 & 1\,1 & 1\,2 \\ 0 & 0\,1 & 1\,0 \end{array}\right]$ can be decomposed into

$$\mathcal{C}\left[\begin{smallmatrix}1\\0\end{smallmatrix}\middle|\begin{smallmatrix}1&1\\0&1\end{smallmatrix}\middle|\begin{smallmatrix}1&2\\1&0\end{smallmatrix}\right] = \mathcal{C}\left[\begin{smallmatrix}3&0\\1&0\end{smallmatrix}\right]^2 + \mathcal{C}\left[\begin{smallmatrix}1&2&3\\1&0&1\end{smallmatrix}\right] - 3\,\mathcal{C}\left[\begin{smallmatrix}1&2&3\\1&1&0\end{smallmatrix}\right] + \widehat{\mathrm{G}}_2\,\mathcal{C}\left[\begin{smallmatrix}1&1&2\\0&1&1\end{smallmatrix}\right] + \frac{\pi}{\tau_2}\,\mathcal{C}\left[\begin{smallmatrix}1&2\\1&-1\end{smallmatrix}\middle|\begin{smallmatrix}2\\1\end{smallmatrix}\right] \tag{6.13}$$

as well as into

$$\mathcal{C}\left[\begin{smallmatrix}1\\0\end{smallmatrix}\middle|\begin{smallmatrix}1&1\\0&1\end{smallmatrix}\middle|\begin{smallmatrix}1&2\\1&0\end{smallmatrix}\right] = 4\,\mathcal{C}\left[\begin{smallmatrix}6&0\\2&0\end{smallmatrix}\right] - \widehat{\mathrm{G}}_2\,\mathcal{C}\left[\begin{smallmatrix}4&0\\2&0\end{smallmatrix}\right] + 3\,\mathcal{C}\left[\begin{smallmatrix}1&1&4\\0&1&1\end{smallmatrix}\right] - \mathcal{C}\left[\begin{smallmatrix}1&2&3\\1&1&0\end{smallmatrix}\right] - \widehat{\mathrm{G}}_2\,\mathcal{C}\left[\begin{smallmatrix}1&1&2\\0&1&1\end{smallmatrix}\right] - \frac{\pi}{\tau_2}\,\mathcal{C}\left[\begin{smallmatrix}5&0\\1&0\end{smallmatrix}\right] - \frac{\pi}{\tau_2}\,\mathcal{C}\left[\begin{smallmatrix}1&1&3\\0&1&0\end{smallmatrix}\right], \tag{6.14}$$

by changing how the partial-fraction decomposition is executed [19].

Performing the HSR using the expressions for the Q_i derived in this section is laborious and it may be challenging to write the final expression back into MGFs in the general case. For this reason, we provide a different procedure to compute n-point HSR in section 6.3.

In the `Mathematica` package `ModularGraphForms`, the trihedral two-point HSR formula from [19] is implemented in the function `TriCSimplify`. Again, with the default

options, negative entries are removed via momentum conservation and identities from the database are applied, so in order to just apply HSR, we run

```
In[38]:= TriCSimplify[c[{2,0},{2,1},{1,1},{1,1}], momSimplify→False, useIds→False]
```

$$\text{Out[38]}= \mathtt{C}\left[\begin{smallmatrix}2\\0\end{smallmatrix}\ \begin{smallmatrix}2\\0\end{smallmatrix}\right]\mathtt{C}\left[\begin{smallmatrix}1\\1\end{smallmatrix}\middle|\begin{smallmatrix}1&1\\1&1\end{smallmatrix}\middle|\begin{smallmatrix}1&1\\1&1\end{smallmatrix}\right] - 6\,\mathtt{C}\left[\begin{smallmatrix}1&1\\1&1\end{smallmatrix}\middle|\begin{smallmatrix}1&1\\1&1\end{smallmatrix}\middle|\begin{smallmatrix}1&4\\1&0\end{smallmatrix}\right] + 2\,\mathtt{C}\left[\begin{smallmatrix}1&1\\1&1\end{smallmatrix}\middle|\begin{smallmatrix}1&1\\1&1\end{smallmatrix}\middle|\begin{smallmatrix}1&2\\1&0\end{smallmatrix}\right]\hat{\mathtt{G}}_2 + \frac{2\pi\,\mathtt{C}\left[\begin{smallmatrix}1&1\\1&1\end{smallmatrix}\middle|\begin{smallmatrix}1&1\\1&1\end{smallmatrix}\middle|\begin{smallmatrix}1&3\\1&-1\end{smallmatrix}\right]}{\tau_2}\ .$$

The three-point HSR detailed in [19] is also performed by the function `TriCSimplify`. For instance,

```
In[39]:= DiCSimplify[TriCSimplify[c[{1,1},{1,2},{1,0}]], momSimplify→False, useIds→False]
```

$$\text{Out[39]}= \mathtt{C}\left[\begin{smallmatrix}3\\1\end{smallmatrix}\ \begin{smallmatrix}0\\0\end{smallmatrix}\right]^2 + \mathtt{C}\left[\begin{smallmatrix}1&2&3\\1&0&1\end{smallmatrix}\right] - 3\,\mathtt{C}\left[\begin{smallmatrix}1&2&3\\1&1&0\end{smallmatrix}\right] + \mathtt{C}\left[\begin{smallmatrix}1&1&2\\0&1&1\end{smallmatrix}\right]\hat{\mathtt{G}}_2 + \frac{\pi\,\mathtt{C}\left[\begin{smallmatrix}1&2&2\\1&-1&1\end{smallmatrix}\right]}{\tau_2}\ ,$$

reproducing (6.13). `TriCSimplify` performs HSR on the first suitable holomorphic subgraph. It first performs the two-point version, then the three-point version, also antiholomorphic subgraphs are simplified. With the Boolean option `triHSR`, trihedral HSR can be deactivated (its default value is `True`) and with the Boolean options `tri2ptHSR` and `tri3ptHSR`, the two- and three-point versions can be deactivated individually.

6.3. Holomorphic subgraph reduction and Fay identities

The discussion of HSR has so far been exclusively in terms of the sum representation of the MGFs. In the integral representation, HSR corresponds to certain identities for products of the $f^{(n)}(z,\tau)$ (2.12b). These descend from the *Fay identity* of the Kronecker–Eisenstein series [40, 46]

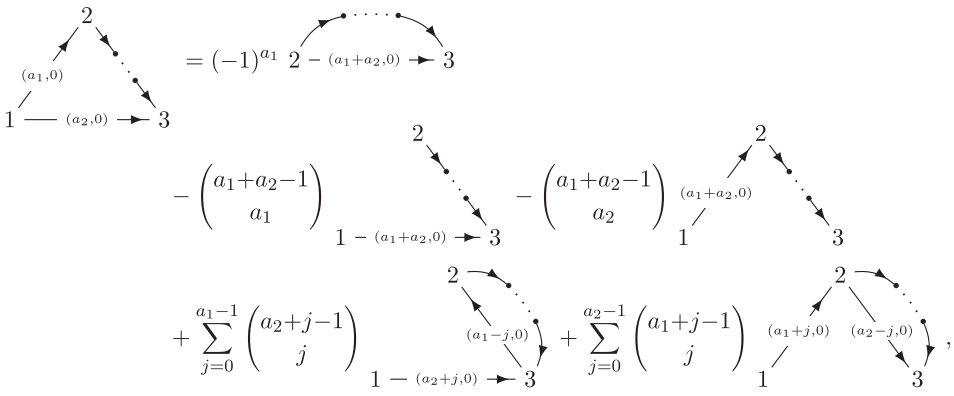
$$\begin{aligned}\Omega(z_1,\eta_1,\tau)\Omega(z_2,\eta_2,\tau) = \Omega(z_1-z_2,\eta_1,\tau)\Omega(z_2,\eta_1+\eta_2,\tau)\\ + \Omega(z_2-z_1,\eta_2,\tau)\Omega(z_1,\eta_1+\eta_2,\tau)\end{aligned} \tag{6.15}$$

by means of the expansion (2.11) and are given by [42]

$$\begin{aligned}f_{12}^{(a_1)}f_{13}^{(a_2)} = (-1)^{a_1-1}f_{23}^{(a_1+a_2)} + \sum_{j=0}^{a_1}\binom{a_2+j-1}{j}f_{32}^{(a_1-j)}f_{13}^{(a_2+j)}\\ + \sum_{j=0}^{a_2}\binom{a_1+j-1}{j}f_{12}^{(a_1+j)}f_{23}^{(a_2-j)},\end{aligned} \tag{6.16}$$

where $a_1, a_2 \geqslant 0$. According to (2.19), a factor $f_{ij}^{(a)}$ in a Koba–Nielsen integral corresponds to a (holomorphic) $(a,0)$-edge. Hence, when (6.16) is applied in a Koba–Nielsen integrand, it generates an identity between MGFs with holomorphic edges.

6.3.1. Holomorphic subgraphs with more than two vertices. Consider an MGF with an n-point holomorphic subgraph ($n > 2$) given by a Koba–Nielsen integral over $C_{12}^{(a_1,0)} C_{13}^{(a_2,0)} = \pm f_{12}^{(a_1)} f_{13}^{(a_2)}$ and $n-2$ further factors $C_{ij}^{(a_k,0)}$, as well as additional non-holomorphic edges. In this case, by focusing on the holomorphic edges, the MGF-identity implied by (6.16) can be written graphically as

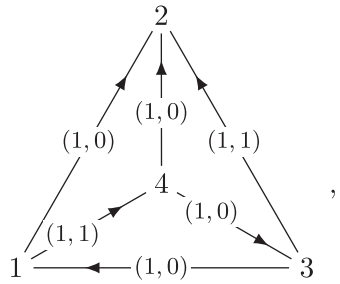


(6.17)

where the ellipsis denotes a sequence of holomorphic edges such that the lhs forms a closed n-point holomorphic graph. The full MGF has additional non-holomorphic edges which can in general connect any vertices and are not drawn in (6.17). In going from (6.16) to (6.17), we have separated the contributions form $f^{(0)} = 1$ since $f^{(0)} \neq -C^{(0,0)}$, according to (5.33). In the representation (6.17) it is clear that on the rhs each term has either one edge less than the lhs (terms two and three) or the closed holomorphic subgraph has one edge less (terms four and five) or both (the first term). If a term on the rhs has an edge less than the lhs, the associated MGF has one loop order less when accounting for the non-holomorphic edges suppressed in (6.17) as well. Hence, the Fay identities (6.16) allow to reduce n-point HSR to $(n-1)$-point HSR plus graphs of lower loop order.

As an example, consider the tetrahedral graph

$$\mathcal{C}\left[\begin{matrix}1\\0\\\hline 1\\0\end{matrix}\middle\|\begin{matrix}1\\0\\\hline 1\\0\end{matrix}\middle\|\begin{matrix}1\\1\\\hline 1\\1\end{matrix}\right] = \qquad (6.18)$$



which has a four-point holomorphic subgraph and appears in the four-gluon amplitude in the heterotic string at the order α'^2 [20]. By applying (6.17) to the two holomorphic edges connected to vertex 4, we obtain the decomposition (with the graphs not yet in their canonical

representation)

$$\mathcal{C}\left[\begin{matrix}1\\0\\\hline 1\\0\end{matrix}\middle\|\begin{matrix}1\\0\\\hline 1\\0\end{matrix}\middle\|\begin{matrix}1\\1\\\hline 1\\1\end{matrix}\right] = -\mathcal{C}\left[\begin{matrix}1\\0\\\hline 1\\0\end{matrix}\middle\|\begin{matrix}\varnothing\\\hline 1\\0\end{matrix}\middle\|\begin{matrix}1&2\\1&0\\\hline 1\\1\end{matrix}\right] - \mathcal{C}\left[\begin{matrix}1\\0\\\hline 2\\0\end{matrix}\middle\|\begin{matrix}\varnothing\\\hline 1\\0\end{matrix}\middle\|\begin{matrix}1\\1\\\hline 1\\1\end{matrix}\right] - \mathcal{C}\left[\begin{matrix}1\\0\\\hline \varnothing\end{matrix}\middle\|\begin{matrix}2\\0\\\hline 1\\0\end{matrix}\middle\|\begin{matrix}1\\1\\\hline 1\\1\end{matrix}\right] + \mathcal{C}\left[\begin{matrix}1\\0\\\hline 1\\0\end{matrix}\middle\|\begin{matrix}\varnothing\\\hline 1\\0\end{matrix}\middle\|\begin{matrix}1&1\\0&1\\\hline 1\\1\end{matrix}\right] - \mathcal{C}\left[\begin{matrix}1\\0\\\hline \varnothing\end{matrix}\middle\|\begin{matrix}1\\0\\\hline 1\\0\end{matrix}\middle\|\begin{matrix}1&1\\0&1\\\hline 1\\1\end{matrix}\right]. \tag{6.19}$$

In this expression, every graph has one empty block and can be simplified using the topological simplifications of section 5.2 to

$$\mathcal{C}\left[\begin{matrix}1\\0\\\hline 1\\0\end{matrix}\middle\|\begin{matrix}1\\0\\\hline 1\\0\end{matrix}\middle\|\begin{matrix}1\\1\\\hline 1\\1\end{matrix}\right] = 2\,\mathcal{C}\left[\begin{smallmatrix}1&2&3\\0&1&1\end{smallmatrix}\right] + 2\,\mathcal{C}\left[\begin{smallmatrix}1\\0\end{smallmatrix}\middle|\begin{smallmatrix}1&1\\0&1\end{smallmatrix}\middle|\begin{smallmatrix}1&2\\0&1\end{smallmatrix}\right]. \tag{6.20}$$

In this way, the four-point HSR in the original graph was reduced to three-point HSR. The three-point HSR can be performed either via another Fay identity or using the formula from [19]. Together with the basis decompositions to be discussed in section 9, we obtain the final result

$$\mathcal{C}\left[\begin{matrix}1\\0\\\hline 1\\0\end{matrix}\middle\|\begin{matrix}1\\0\\\hline 1\\0\end{matrix}\middle\|\begin{matrix}1\\1\\\hline 1\\1\end{matrix}\right] = 2\,\mathcal{C}\left[\begin{smallmatrix}6&0\\2&0\end{smallmatrix}\right] - 4\,\mathcal{C}\left[\begin{smallmatrix}3&0\\1&0\end{smallmatrix}\right]^2 + 2\widehat{\mathrm{G}}_2\,\mathcal{C}\left[\begin{smallmatrix}4&0\\2&0\end{smallmatrix}\right] - 12\frac{\pi}{\tau_2}\,\mathcal{C}\left[\begin{smallmatrix}5&0\\1&0\end{smallmatrix}\right] + 4\frac{\pi}{\tau_2}\widehat{\mathrm{G}}_2\,\mathcal{C}\left[\begin{smallmatrix}3&0\\1&0\end{smallmatrix}\right] + 4\left(\frac{\pi}{\tau_2}\right)^2\mathrm{G}_4\,. \tag{6.21}$$

In general, the closed holomorphic subgraph is of course not necessary for the identity (6.16) to hold. Hence, if we remove the edges between vertices 2 and 3 from (6.17), this generates identities between MGFs which have at least two non-parallel holomorphic edges both connected to the same vertex. For trihedral graphs, we have e.g.

$$\begin{aligned}\mathcal{C}\left[\begin{smallmatrix}A_1&a_1\\B_1&0\end{smallmatrix}\middle|\begin{smallmatrix}A_2&a_2\\B_2&0\end{smallmatrix}\middle|\begin{smallmatrix}A_3\\B_3\end{smallmatrix}\right] &= (-)^{a_1+a_2}\,\mathcal{C}\left[\begin{smallmatrix}A_1\\B_1\end{smallmatrix}\middle|\begin{smallmatrix}A_2\\B_2\end{smallmatrix}\middle|\begin{smallmatrix}A_3&a_1+a_2\\B_3&0\end{smallmatrix}\right]\\ &\quad+(-)^{a_1+1}\binom{a_1+a_2-1}{a_1}\mathcal{C}\left[\begin{smallmatrix}A_1\\B_1\end{smallmatrix}\middle|\begin{smallmatrix}A_2&a_1+a_2\\B_2&0\end{smallmatrix}\middle|\begin{smallmatrix}A_3\\B_3\end{smallmatrix}\right]\\ &\quad+(-)^{a_2+1}\binom{a_1+a_2-1}{a_2}\mathcal{C}\left[\begin{smallmatrix}A_1&a_1+a_2\\B_1&0\end{smallmatrix}\middle|\begin{smallmatrix}A_2\\B_2\end{smallmatrix}\middle|\begin{smallmatrix}A_3\\B_3\end{smallmatrix}\right]\\ &\quad+(-)^{a_1}\sum_{j=0}^{a_1-1}\binom{a_2+j-1}{j}\mathcal{C}\left[\begin{smallmatrix}A_1\\B_1\end{smallmatrix}\middle|\begin{smallmatrix}A_2&a_2+j\\B_2&0\end{smallmatrix}\middle|\begin{smallmatrix}A_3&a_1-j\\B_3&0\end{smallmatrix}\right]\\ &\quad+(-)^{a_2}\sum_{j=0}^{a_2-1}\binom{a_1+j-1}{j}\mathcal{C}\left[\begin{smallmatrix}A_1&a_1+j\\B_1&0\end{smallmatrix}\middle|\begin{smallmatrix}A_2\\B_2\end{smallmatrix}\middle|\begin{smallmatrix}A_3&a_2-j\\B_3&0\end{smallmatrix}\right].\end{aligned} \tag{6.22}$$

This identity will be a key ingredient in deriving the basis decompositions for all dihedral and trihedral MGFs of total modular weight at most 12 is section 9. If the $\left[\begin{smallmatrix}A_3\\B_3\end{smallmatrix}\right]$-block contains a holomorphic edge, (6.22) is a reduction of three-point HSR to two-point HSR and graphs of lower loop order. In this case, the Fay identity could be used on any pair of non-parallel holomorphic edges and this choice corresponds to the different ways to perform the partial-fraction

decomposition in section 6.2, leading to interesting identities between MGFs in general. As an example, consider the graph $\mathcal{C}\left[\begin{array}{c|c|c} 1 & 1\,1 & 1\,2 \\ 0 & 0\,1 & 1\,0 \end{array}\right]$ which was decomposed using the traditional HSR method in section 6.2. Applying (6.22) to the first two holomorphic columns of this graph leads to

$$\mathcal{C}\left[\begin{smallmatrix}1\\0\end{smallmatrix}\middle|\begin{smallmatrix}1&1\\0&1\end{smallmatrix}\middle|\begin{smallmatrix}1&2\\1&0\end{smallmatrix}\right] = \mathcal{C}\left[\begin{smallmatrix}\varnothing\\\end{smallmatrix}\middle|\begin{smallmatrix}1\\1\end{smallmatrix}\middle|\begin{smallmatrix}1&2&2\\1&0&0\end{smallmatrix}\right] - \mathcal{C}\left[\begin{smallmatrix}\varnothing\\\end{smallmatrix}\middle|\begin{smallmatrix}1&1\\0&1\end{smallmatrix}\middle|\begin{smallmatrix}1&1&2\\0&1&0\end{smallmatrix}\right] + \mathcal{C}\left[\begin{smallmatrix}\varnothing\\\end{smallmatrix}\middle|\begin{smallmatrix}1&2\\1&0\end{smallmatrix}\middle|\begin{smallmatrix}1&2\\1&0\end{smallmatrix}\right] - \mathcal{C}\left[\begin{smallmatrix}1\\0\end{smallmatrix}\middle|\begin{smallmatrix}1\\1\end{smallmatrix}\middle|\begin{smallmatrix}1&1&2\\0&1&0\end{smallmatrix}\right] + \mathcal{C}\left[\begin{smallmatrix}1\\1\end{smallmatrix}\middle|\begin{smallmatrix}2\\0\end{smallmatrix}\middle|\begin{smallmatrix}1&2\\1&0\end{smallmatrix}\right], \tag{6.23}$$

which can be shown to be equal to the decomposition (6.13) upon using the topological simplifications from section 5.2 and the dihedral HSR formula (6.6). On the other hand, we can also apply (6.22) to the second and third holomorphic edges, yielding

$$\mathcal{C}\left[\begin{smallmatrix}1\\0\end{smallmatrix}\middle|\begin{smallmatrix}1&1\\0&1\end{smallmatrix}\middle|\begin{smallmatrix}1&2\\1&0\end{smallmatrix}\right] = \mathcal{C}\left[\begin{smallmatrix}1\\0\end{smallmatrix}\middle|\begin{smallmatrix}1\\1\end{smallmatrix}\middle|\begin{smallmatrix}1&3\\1&0\end{smallmatrix}\right] - \mathcal{C}\left[\begin{smallmatrix}1\\1\end{smallmatrix}\middle|\begin{smallmatrix}1\\1\end{smallmatrix}\middle|\begin{smallmatrix}1&3\\0&0\end{smallmatrix}\right] + \mathcal{C}\left[\begin{smallmatrix}1\\1\end{smallmatrix}\middle|\begin{smallmatrix}1&1\\0&1\end{smallmatrix}\middle|\begin{smallmatrix}1&2\\0&0\end{smallmatrix}\right]. \tag{6.24}$$

This can be simplified to (6.14) by topological identities.

6.3.2. Holomorphic subgraphs with two vertices. The restriction of (6.17) to holomorphic edges which are not parallel arises because the Fay identity for Kronecker–Eisenstein series (6.15) involves the three different z_1, z_2 and $z_1 - z_2$. As discussed in appendix A of [29], by taking the limit $z_1 \to z_2$ we obtain the Fay identity

$$\begin{aligned} f^{(a_1)}(z) f^{(a_2)}(z) = (-1)^{a_2}\Theta(a_1+a_2-4)\mathrm{G}_{a_1+a_2} + \binom{a_1+a_2}{a_2} f^{(a_1+a_2)}(z) \\ - \sum_{k=4}^{a_1} \binom{a_1+a_2-1-k}{a_2-1} \mathrm{G}_k f^{(a_1+a_2-k)}(z) \\ - \sum_{k=4}^{a_2} \binom{a_1+a_2-1-k}{a_1-1} \mathrm{G}_k f^{(a_1+a_2-k)}(z) \\ - \binom{a_1+a_2-2}{a_2-1} \left(\widehat{\mathrm{G}}_2 f^{(a_1+a_2-2)}(z) + \partial_z f^{(a_1+a_2-1)}(z) \right), \end{aligned} \tag{6.25}$$

where $a_1, a_2 > 0$ and Θ is the Heaviside step-function

$$\Theta(x) = \begin{cases} 1 & \text{if } x \geqslant 0 \\ 0 & \text{if } x < 0 \end{cases}. \tag{6.26}$$

Integrating (6.25) against a suitable product of $C^{(a,b)}$ functions yields two-point HSR upon using that

$$\partial_z f^{(a_1+a_2-1)}(z) = (-1)^{a_1+a_2+1} \frac{\pi}{\tau_2} C^{(a_1+a_2-1,-1)}(z) \tag{6.27}$$

according to (5.22). E.g. when (6.25) for $a_1 + a_2 \geqslant 3$ is integrated against $\prod_{i=1}^{R} C^{(a_i,b_i)}$, we obtain the dihedral HSR identity (6.6).

Together, (6.16) and (6.25) allow to perform HSR of holomorphic subgraphs with arbitrarily many vertices in a compact way. Note that when using Fay identities, we circumvent the need to

evaluate conditionally convergent sums with the Eisenstein summation prescription as shown in section 6.2. For trihedral three-point HSR, it was checked explicitly in many cases that a combination of (6.22) and two-point HSR yields an equivalent expression to the one obtained from the formula in [19].

In the `ModularGraphForms` package, the trihedral Fay identities (6.22) are implemented in the function `TriFay` which returns an equation. The first argument of this function is the trihedral MGF to be decomposed, the second (optional) argument has the form `{{b1,c1},{b2,c2}}`, where `b1` and `b2` are the blocks of the (anti)holomorphic edges to be used and `c1` and `c2` are the columns of those edges. If the second argument is omitted, the first suitable pair of (anti)holomorphic edges is selected automatically. As an example, we will consider the decomposition of the graph $\mathcal{C}\left[\begin{array}{c|c|c} 1 & 1\,1 & 1\,2 \\ 0 & 0\,1 & 1\,0 \end{array}\right]$ as discussed around (6.23) and (6.24). In order to apply (6.22) to the first two holomorphic columns and then simplify the result to obtain (6.13), we run

```
In[40]:= TriFay[C[{1,0},{1,1},{1,0}],{{1,1},{2,1}}]
         DiCSimplify[TriCSimplify[%[[2]]],useIds→False,momSimplify→False]
```

$$\text{Out[40]= } C\left[\begin{array}{c|c|c} 1 & 1 & 2 \\ 0 & 1 & 0 \end{array}\right] == C\left[\{\}\,\middle|\,\begin{array}{c} 1 \\ 1 \end{array}\,\middle|\,\begin{array}{ccc} 1 & 2 & 2 \\ 1 & 0 & 0 \end{array}\right] - C\left[\{\}\,\middle|\,\begin{array}{c} 1 \\ 0 \end{array}\,\middle|\,\begin{array}{ccc} 1 & 1 & 2 \\ 0 & 1 & 0 \end{array}\right] + C\left[\{\}\,\middle|\,\begin{array}{c} 2 \\ 0 \end{array}\,\middle|\,\begin{array}{cc} 1 & 2 \\ 1 & 0 \end{array}\right] - C\left[\begin{array}{c|c|c} 1 & 1 & 1 \\ 0 & 1 & 0 \end{array}\right]$$

$$\text{Out[41]= } C\left[\begin{array}{cc} 3 & 0 \\ 1 & 0 \end{array}\right]^2 + C\left[\begin{array}{ccc} 1 & 2 & 3 \\ 1 & 0 & 1 \end{array}\right] - 3\,C\left[\begin{array}{ccc} 1 & 2 & 3 \\ 1 & 1 & 0 \end{array}\right] + C\left[\begin{array}{ccc} 1 & 1 & 2 \\ 0 & 1 & 1 \end{array}\right]\hat{G}_2 + \frac{\pi\, C\left[\begin{array}{ccc} 1 & 2 & 2 \\ 1 & -1 & 1 \end{array}\right]}{\tau_2}$$

reproducing (6.13). Similarly, (6.14) can be obtained by changing the second argument of `TriFay` in `In[40]` to `{{2, 1}, {3, 2}}` and replacing the option `momSimplify` → `False` of `DiCSimplify` by `diHSR` → `False`.

As mentioned above, trihedral three-point HSR is performed by the function `TriCSimplify`, which implements the formula from [19]. If the Boolean option `tri3ptFayHSR` (which is inherited by `CSimplify`), is set to `True` (the default is `False`), the three-point HSR is instead performed using the Fay identity (6.22) and subsequent two-point HSR. The results of applying the two techniques may look different, if the basis decompositions from section 9 are not applied, but they are in fact equivalent.

7. The sieve algorithm

With the techniques described in the last two sections, many valuable identities between MGFs can be derived. However, if one is interested in simplifying a particular MGF, e.g. one which has appeared as an expansion coefficient of a Koba–Nielsen integral, it is not always clear which techniques to combine to obtain the desired decomposition. In this situation, the sieve algorithm, first introduced in [8], can be used: it allows for a systematic decomposition (up to an overall constant) of arbitrary MGFs, as long as the basis for the decomposition and all MGFs of lower total modular weight $a+b$ are known.

7.1. Constructing identities

As a starting point, assume that we have a combination F of MGFs of homogeneous modular weight $(|A|,|B|)$ and we want to check whether or not it vanishes. The idea behind the sieve algorithm is to repeatedly take derivatives of F using the Maaß operator $\nabla^{(|A|)}$ defined in (5.40). Due to an intricate interplay between momentum conservation identities and HSR, described

in detail in [8], every derivative can be expressed as a linear combination of products of holomorphic Eisenstein series, MGFs with non-negative antiholomorphic labels for each edge, τ_2 with non-positive exponent, MGFs of the form $\mathcal{C}\begin{bmatrix} k & 0 \\ -n & 0 \end{bmatrix}$ with $k > n$ and modular invariant factors. After taking $|B|$ derivatives, the antiholomorphic modular weight vanishes according to (5.41) and hence each term in the derivative has to factorize, since any unfactorized MGFs would have to have vanishing antiholomorphic labels and therefore be amenable to HSR, leading to a factorized expression. Using the generalized Ramanujan identities from section 5.5, the factors of the form $\mathcal{C}\begin{bmatrix} k & 0 \\ -n & 0 \end{bmatrix}$ can be decomposed as well. Since each term is factorized, the total modular weight $a+b$ of every leftover MGF is strictly less than $|A|+|B|$ and if we know all identities between MGFs of lower total modular weight, it is manifest if the $|B|$th derivative of F vanishes or not. If F has $|A| = |B|$, then lemma 1 in [8] guarantees that if the derivative vanishes, $F = 0$ up to an overall constant. If $|A| \neq |B|$ and F can be written as the derivative of an expression with $|A| = |B|$, this primitive vanishes up to a constant, so $F = 0$ as well. We conjecture that the same is true if F cannot be written as the derivative of an expression with $|A| = |B|$, in line with all cases we tested. In this way, we can generate identities at progressively higher total modular weight.

The Cauchy–Riemann derivative of a holomorphic Eisenstein series has the form $\mathcal{C}\begin{bmatrix} 2k+1 & 0 \\ -1 & 0 \end{bmatrix}$, i.e. it is a graph with one edge with negative antiholomorphic weight. In this case, momentum conservation (and HSR) cannot be used to remove the negative entry and in the original version published in [8], this fact was used to *sieve* the space of MGFs for identities: after taking a derivative and trading negative antiholomorphic entries for holomorphic Eisenstein series, one subtracts the same derivative of an MGF in such a way that all holomorphic Eisenstein series cancel. Then, one can take the next derivative of the combined expression without generating irremovable negative antiholomorphic labels. After having taken $|B|$ derivatives, the result is purely holomorphic (and still modular), so we can expand it in the ring of holomorphic Eisenstein series. By subtracting one final MGF such that this derivative vanishes, one has constructed an identity up to an overall constant. In fact, if a combination of MGFs vanishes, then the holomorphic Eisenstein series have to cancel out in every derivative. This can however only be verified, if the prefactors of the holomorphic Eisenstein series are linearly independent. Since they carry lower total modular weight than the complete expression, this means that we need to know all identities between graphs of lower total modular weight.

In general, finding MGFs with the correct Cauchy–Riemann derivatives to cancel the holomorphic Eisenstein series can be challenging but if we want to find a decomposition of an MGF into a set of basis MGFs, we can just take the derivatives of a linear combination of the basis elements and adjust the coefficients so that the holomorphic Eisenstein series cancel. This is what is done in the implementation of the sieve algorithm in the `ModularGraphForms` package.

Instead of canceling holomorphic Eisenstein series in every derivative as described above and in [8], one can also use the generalized Ramanujan identities discussed in section 5.5 to perform the derivatives of the holomorphic Eisenstein series. In this way, the highest derivative of any MGF can be written in terms of holomorphic Eisenstein series and MGFs of lower total modular weight for which we assume that the relations are known, hence identities can be found explicitly.

Consider e.g. the weight-$(4,4)$ MGF $\mathcal{C}\begin{bmatrix} 1 & 1 & 2 \\ 1 & 2 & 1 \end{bmatrix}$. The fourth derivatives of this graph and the weight-$(4,4)$ basis elements $C_{1,1,2}$, E_2^2 and E_4 are

$$\nabla^{(4)^4}\mathcal{C}\begin{bmatrix}1&1&2\\1&2&1\end{bmatrix} = 120\mathrm{E}_2\mathrm{G}_4\widehat{\mathrm{G}}_2^2 - 840\mathrm{E}_2\mathrm{G}_6\widehat{\mathrm{G}}_2 + 600\mathrm{E}_2\mathrm{G}_4^2 - 360\mathrm{G}_4^2 + 840\mathrm{G}_6\frac{\tau_2}{\pi}\mathcal{C}\begin{bmatrix}3&0\\1&0\end{bmatrix} - 240\mathrm{G}_4\widehat{\mathrm{G}}_2\frac{\tau_2}{\pi}\mathcal{C}\begin{bmatrix}3&0\\1&0\end{bmatrix} \tag{7.1a}$$

$$\nabla^{(4)^4}\mathcal{C}\begin{bmatrix}1&1&2\\1&1&2\end{bmatrix} = 288\mathrm{G}_4^2 + 48\mathrm{G}_4\widehat{\mathrm{G}}_2\frac{\tau_2}{\pi}\mathcal{C}\begin{bmatrix}3&0\\1&0\end{bmatrix} - 168\mathrm{G}_6\frac{\tau_2}{\pi}\mathcal{C}\begin{bmatrix}3&0\\1&0\end{bmatrix} \tag{7.1b}$$

$$\nabla^{(4)^4}\left(\left(\frac{\pi}{\tau_2}\right)^2\mathrm{E}_2^2\right) = 240\mathrm{E}_2\mathrm{G}_4\widehat{\mathrm{G}}_2^2 - 1680\mathrm{E}_2\mathrm{G}_6\widehat{\mathrm{G}}_2 + 1200\mathrm{E}_2\mathrm{G}_4^2 + 216\mathrm{G}_4^2 - 384\mathrm{G}_4\widehat{\mathrm{G}}_2\frac{\tau_2}{\pi}\mathcal{C}\begin{bmatrix}3&0\\1&0\end{bmatrix} + 1344\mathrm{G}_6\frac{\tau_2}{\pi}\mathcal{C}\begin{bmatrix}3&0\\1&0\end{bmatrix} \tag{7.1c}$$

$$\nabla^{(4)^4}\left(\left(\frac{\pi}{\tau_2}\right)^4\mathrm{E}_4\right) = 180\mathrm{G}_4^2 \tag{7.1d}$$

where we used the notation (5.43) and simplified all derivatives of holomorphic Eisenstein series using the techniques from section 5.5. Setting a linear combination of these four expressions to zero and requiring the coefficients of the various terms on the rhs to vanish leaves

$$\nabla^{(4)^4}\left(\mathcal{C}\begin{bmatrix}1&1&2\\1&2&1\end{bmatrix} + \mathcal{C}\begin{bmatrix}1&1&2\\1&1&2\end{bmatrix} - \frac{1}{2}\left(\frac{\pi}{\tau_2}\right)^2\mathrm{E}_2^2 + \frac{1}{2}\left(\frac{\pi}{\tau_2}\right)^4\mathrm{E}_4\right) = 0 \tag{7.2}$$

as the only solution. If no solution had existed, the four MGFs in (7.1) would have been proven to be linearly independent. Lemma 1 in [8] now states that this implies

$$\mathcal{C}\begin{bmatrix}1&1&2\\1&2&1\end{bmatrix} + \mathcal{C}\begin{bmatrix}1&1&2\\1&1&2\end{bmatrix} - \frac{1}{2}\left(\frac{\pi}{\tau_2}\right)^2\mathrm{E}_2^2 + \frac{1}{2}\left(\frac{\pi}{\tau_2}\right)^4\mathrm{E}_4 = \left(\frac{\pi}{\tau_2}\right)^4\text{const.} \tag{7.3}$$

with some τ-independent constant[8]. Using the techniques discussed in the previous sections, one can also decompose $\mathcal{C}\begin{bmatrix}1&1&2\\1&2&1\end{bmatrix}$ directly and finds that the constant vanishes in this case (as expected since there is no single-valued MZV at the expected transcendental weight 4).

In the `ModularGraphForms` package, the removal of edge labels -1 for dihedral and trihedral graphs is done by the functions `DiCSimplify` and `TriCSimplify`, if the option `momSimplify` is set to `True` (the default). The sieve algorithm itself is implemented in the function `CSieveDecomp`, which uses the traditional method of canceling holomorphic

[8] Due to our normalization conventions, the graphs with equal total holomorphic and antiholomorphic edge labels are not modular invariant, hence the integration constant is multiplied by a suitable power of $\frac{\pi}{\tau_2}$.

Eisenstein series in every step. If no further options are given, this function tries to decompose the graph given in its argument into the basis discussed in section 9, e.g. for the graph $\mathcal{C}\begin{bmatrix}1 & 1 & 2\\ 1 & 2 & 1\end{bmatrix}$ we considered above, we can run

```
In[42]:= CSieveDecomp[c[1 1 2]]
```

$$\text{Out[42]}= -\mathtt{C}\left[\begin{smallmatrix}1 & 1 & 2\\ 1 & 1 & 2\end{smallmatrix}\right] + \frac{\pi^4\,\mathtt{E}_2^2}{2\,\tau_2^4} - \frac{\pi^4\,\mathtt{E}_4}{2\,\tau_2^4} + \frac{\pi^4\,\mathtt{intConst}\left[\begin{smallmatrix}1 & 1 & 2\\ 1 & 2 & 1\end{smallmatrix}\right]}{\tau_2^4}\ ,$$

reproducing (7.3). The last term in the output is an undetermined integration constant, labeled by the exponent matrix of the original graph. Such a constant is added for all graphs with equal holomorphic and antiholomorphic weight. Setting the Boolean option `verbose` of `CSieveDecomp` to `True` prints a detailed progress report into the notebook with the expressions appearing in each derivative and the prefactors of the holomorphic Eisenstein series which are set to zero. For instance, the output for the third derivative in the computation above is

```
3rd derivative:
```

$$-168\,\mathtt{C}\left[\begin{smallmatrix}7 & 0\\ 1 & 0\end{smallmatrix}\right] - 108\,\mathtt{bCoeff[1]}\,\mathtt{C}\left[\begin{smallmatrix}7 & 0\\ 1 & 0\end{smallmatrix}\right] - 120\,\mathtt{bCoeff[2]}\,\mathtt{C}\left[\begin{smallmatrix}7 & 0\\ 1 & 0\end{smallmatrix}\right]$$

$$+12\,\mathtt{C}\left[\begin{smallmatrix}3 & 0\\ 1 & 0\end{smallmatrix}\right]\mathtt{G}_4 + 12\,\mathtt{bCoeff[1]}\,\mathtt{C}\left[\begin{smallmatrix}3 & 0\\ 1 & 0\end{smallmatrix}\right]\mathtt{G}_4$$

```
(Anti-)holomorphic Eisenstein series:
```

$$\{\mathtt{G}_4\}$$

```
Coefficients that should be zero:
```

$$\left\{12\,\mathtt{C}\left[\begin{smallmatrix}3 & 0\\ 1 & 0\end{smallmatrix}\right] + 12\,\mathtt{bCoeff[1]}\,\mathtt{C}\left[\begin{smallmatrix}3 & 0\\ 1 & 0\end{smallmatrix}\right]\right\}$$

```
Find solution for all
```

$$\left\{\mathtt{C}\left[\begin{smallmatrix}3 & 0\\ 1 & 0\end{smallmatrix}\right]\right\}$$

```
Solutions:
```

$$\{\{\mathtt{bCoeff[1]} \to -1\}\}\ .$$

As one can see, `CSieveDecomp` forms a linear combination of the basis elements with coefficients `bCoeff` and subtracts it from the MGF which is decomposed. Then, derivatives are taken and in each step the coefficients of the holomorphic Eisenstein series are set to zero by fixing some of the `bCoeff`.

The basis used for the decomposition is determined by the option `basis` of `CSieveDecomp`. If `basis` is an empty list (the default), the basis is determined by the function `CBasis`, to be discussed in more detail in section 9. Otherwise, one can also supply a list of MGFs of the same weight as the MGF to be decomposed. E.g. we can reproduce the momentum conservation identity of the seed $\mathcal{C}\begin{bmatrix}1 & 2 & 2\\ 1 & 2 & 1\end{bmatrix}$ (up to an overall constant) by

running

$$\text{In[43]:=}\ \texttt{CSieveDecomp}\left[\texttt{c}\left[\begin{smallmatrix}1&1&2\\1&2&1\end{smallmatrix}\right],\ \texttt{basis}\to\left\{\texttt{c}\left[\begin{smallmatrix}0&2&2\\1&1&2\end{smallmatrix}\right],\ \texttt{c}\left[\begin{smallmatrix}1&1&2\\1&1&2\end{smallmatrix}\right]\right\}\right]$$

$$\text{Out[43]=}\ -\texttt{C}\left[\begin{smallmatrix}0&2&2\\1&1&2\end{smallmatrix}\right]-\texttt{C}\left[\begin{smallmatrix}1&1&2\\1&1&2\end{smallmatrix}\right]+\frac{\pi^4\,\texttt{intConst}\left[\begin{smallmatrix}1&1&2\\1&2&1\end{smallmatrix}\right]}{\tau_2^4}\ .$$

If not all coefficients can be fixed (e.g. because the `basis` provided is not linearly independent), `bCoeff` will appear in the output. For further options and the meaning of various error messages, cf appendix A.

8. Divergent modular graph forms

So far, we have not discussed the convergence properties of the lattice sum (2.23) of MGFs, but, of course, if the edge labels become too low, the sum (2.23) is not absolutely convergent any more. Interestingly, conditionally convergent or divergent sums can arise even when one applies the techniques above only to convergent sums. Sometimes, the divergence cannot be avoided, e.g. when using the sieve algorithm to find decompositions of certain convergent graphs. When deriving identities, one way to deal with this phenomenon is to just disregard all identities in which divergent graphs appear. This is the approach taken in section 9 for (convergent) dihedral and trihedral MGFs of weight $a+b\leqslant 10$. However, in this way, one misses many valuable identities and hence it is desirable to have at least a partial understanding of how to interpret divergent MGFs. In this section, we will describe concrete results which go in this direction. Below, we will use these divergent techniques to obtain all dihedral and trihedral (convergent) basis decompositions for $a+b=12$.

8.1. Divergence conditions

In this section, we will give simple power-counting arguments to determine if a particular MGF is absolutely convergent or not, building on the behavior of holomorphic Eisenstein series, for which we know that

$$\mathrm{G}_a=\sum_p{}' \frac{1}{p^a} \tag{8.1}$$

is absolutely convergent for $a\geqslant 3$, conditionally convergent for $a=2$ and divergent for $a\leqslant 1$. Accordingly, we will call an MGF convergent if all momenta in the sum (2.23) have at least three powers in the denominator (adding powers of p and $\bar{p}$) and divergent if any momentum appears with two or less powers in the denominator[9].

In order to determine the total powers with which a momentum can appear, one has to perform some of the sums first by using the momentum-conserving delta functions (cf e.g. (4.1)). Of course, there is considerable freedom in which sums we choose for this, hence different final expressions can result, with different total powers of the momenta. These expressions correspond to different rotations of the coordinate axes in the lattice spanned by the momenta.

[9] Note that this simple power-counting criterion does not constitute a proof of the convergence or divergence of the lattice sum of the MGF. In fact, as we will discuss below, this argument tends to underestimate the convergence of the sum since possible cancellations are not accounted for.

Since by counting the total exponents, we only test the convergence properties along the coordinate axes, we pick the representation with the lowest total power. To illustrate this, consider the dihedral graph

$$\mathcal{C}\begin{bmatrix}1 & 1 & 2\\ 0 & 0 & 2\end{bmatrix} = \sum_{p_1,p_2,p_3}{}' \frac{\delta(p_1+p_2+p_3)}{p_1 p_2 |p_3|^4}. \tag{8.2}$$

We can use the delta function to perform either the p_3 sum or the p_2 sum, yielding the expressions

$$\mathcal{C}\begin{bmatrix}1 & 1 & 2\\ 0 & 0 & 2\end{bmatrix} = \sum_{p_1,p_2}{}' \frac{1}{p_1 p_2 |p_1+p_2|^4} = -\sum_{p_1,p_3}{}' \frac{1}{p_1 (p_1+p_3) |p_3|^4}. \tag{8.3}$$

In the first of these expressions, p_1 and p_2 both come with a power of 5 in the denominator, hence according to our criterion above, $\mathcal{C}\begin{bmatrix}1 & 1 & 2\\ 0 & 0 & 2\end{bmatrix}$ should be convergent. In the second expression in (8.3) however, p_1 comes with a power of 2, hence, $\mathcal{C}\begin{bmatrix}1 & 1 & 2\\ 0 & 0 & 2\end{bmatrix}$ should be divergent. The reason that the first expression seems to be convergent is that the divergence lies in the direction of $p_1 + p_2 = \text{const.}$, whereas by counting the powers of p_1 and p_2, we only probed the directions along those two momenta. Therefore, $\mathcal{C}\begin{bmatrix}1 & 1 & 2\\ 0 & 0 & 2\end{bmatrix}$ is divergent.

To summarize, an MGF is only convergent if the powers of all momenta are at least three, in all possible ways to solve the delta functions. We will translate this in the following into conditions on the labels of the two-, three- and four-point graphs introduced in section 4.

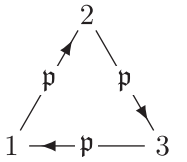
In dihedral graphs, if we perform the sum over momentum p with the delta function, we will increase the total powers of all other momenta by the total power of p. Hence, our divergence criterion for dihedral graphs, taking into account that we can use any of the momenta to solve the delta function, is

$$\mathcal{C}\begin{bmatrix}A\\ B\end{bmatrix} \text{ convergent} \Leftrightarrow \min_{\substack{i,j\\ i\neq j}}(c_i + c_j) > 2, \tag{8.4}$$

where $c_i = a_i + \mathrm{b}_i$ and i, j run over all edges. The basic criterion (8.4) will have to be satisfied for all edge bundles in higher-point graphs as well, but the global structure of these graphs adds further criteria.

In general, solving delta functions is equivalent to assigning loop momenta consistently to the edges of the graph. Hence, by going through the topologically distinct assignments, we can see to which edges a certain momentum can propagate and hence what the convergence conditions for this graph should be. When considering graphs with edge bundles between the vertices (like the graphs introduced in section 4), we first assign the total momenta of the bundles consistently. Then, in a bundle of total momentum $\mathfrak{p}$, with edges carrying momenta $p_1, \ldots, p_R$, we can choose any edge to solve the momentum conservation constraint, e.g. we can drop momentum p_1 and assign momentum $\mathfrak{p} - \sum_{i=2}^{R} p_i$ to this edge. For the convergence conditions, the implications of this are twofold: first, each momentum can appear in any other edge of the same bundle, implying the condition (8.4) for each bundle. Second, the total momenta of the edge bundles can appear in any edge, hence we should count the lowest total power for each edge bundle when determining the convergence condition due to the total momenta. We will go through this procedure for the three-point and all four-point graphs in section 4 in the following.

For trihedral graphs, there is just one way to assign the bundle momenta, namely



, (8.5)

i.e. the graph $\mathcal{C}\left[\begin{array}{c|c|c} A_1 & A_2 & A_3 \\ B_1 & B_2 & B_3 \end{array}\right]$ is convergent iff

$$\min_{\substack{i,j \\ i\neq j}} \left(c_i^{(k)} + c_j^{(k)}\right) > 2 \quad \forall\, k \in \{1,2,3\} \text{ and } \check{c}_1 + \check{c}_2 + \check{c}_3 > 2, \tag{8.6}$$

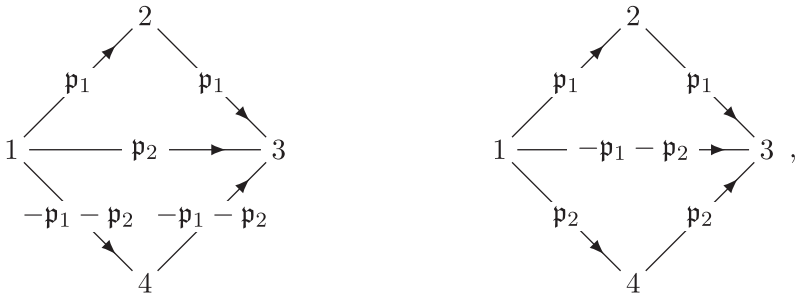
where $c_i^{(k)} = a_i^{(k)} + b_i^{(k)}$ for $i = 1, \ldots, R_k$ in the notation of (4.6) and $\check{c}_k = \min_i(a_i^{(k)} + b_i^{(k)})$, where i runs over all edges in block k. As described above, the first condition is due to the individual momenta in the edge bundles, whereas the second condition is due to the total bundle momentum $\mathfrak{p}$. If $\mathcal{A}\left[\begin{array}{ccc} A_1 & A_2 & A_3 \\ B_1 & B_2 & B_3 \end{array}\right]$ carries only non-negative edge labels and does not contain a $(0,0)$-edge (i.e. is not factorizable), then $c_i \geqslant 1$ for all edges and the second condition in (8.6) is always satisfied. The same will be true for all other conditions on top of (8.4) for every block in the following.

As a straightforward extension of the trihedral result, the box graph $\mathcal{C}\left[\begin{smallmatrix} A_1 \\ B_1 \end{smallmatrix}\middle|\begin{smallmatrix} A_2 \\ B_2 \end{smallmatrix}\middle|\begin{smallmatrix} A_3 \\ B_3 \end{smallmatrix}\middle|\begin{smallmatrix} A_4 \\ B_4 \end{smallmatrix}\right]$ is convergent iff

$$\min_{\substack{i,j \\ i\neq j}} \left(c_i^{(k)} + c_j^{(k)}\right) > 2 \quad \forall\, k \in \{1,2,3,4\} \text{ and } \check{c}_1 + \check{c}_2 + \check{c}_3 + \check{c}_4 > 2, \tag{8.7}$$

with the same notation as in (8.6).

In kite graphs, there are two topologically distinct ways of assigning the total momenta of the edge bundles. They are

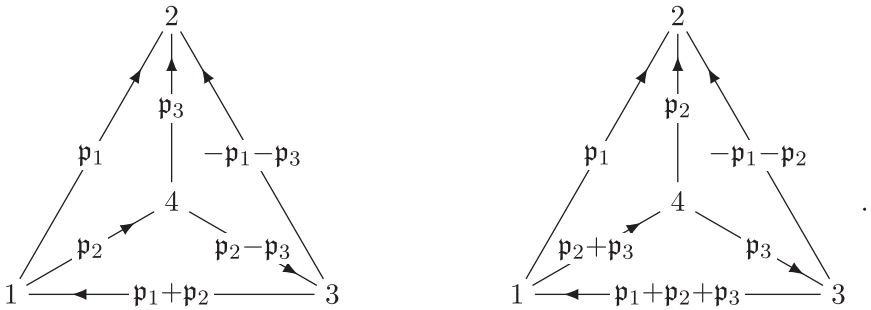


, (8.8)

implying that $\mathcal{C}\left[\begin{smallmatrix} A_1 \\ B_1 \end{smallmatrix}\middle|\begin{smallmatrix} A_2 \\ B_2 \end{smallmatrix}\middle\|\begin{smallmatrix} A_3 \\ B_3 \end{smallmatrix}\middle|\begin{smallmatrix} A_4 \\ B_4 \end{smallmatrix}\middle\|\begin{smallmatrix} A_5 \\ B_5 \end{smallmatrix}\right]$ is convergent iff

$$\begin{aligned} &\min_{\substack{i,j \\ i\neq j}} \left(c_i^{(k)} + c_j^{(k)}\right) > 2 \quad \forall\, k \in \{1,2,3,4,5\} \\ \text{and} \quad &\check{c}_i + \check{c}_j + \check{c}_5 > 2 \quad \forall\, (i,j) \in \{(1,2),(3,4)\} \\ \text{and} \quad &\check{c}_1 + \check{c}_2 + \check{c}_3 + \check{c}_4 > 2. \end{aligned} \tag{8.9}$$

For tetrahedral graphs, there are again two topologically distinct ways to assign the three independent total edge-bundle momenta,



(8.10)

This implies that the tetrahedral graph

$$\mathcal{C}\left[\begin{array}{c|c|c} \begin{matrix}A_1\\B_1\end{matrix} & \begin{matrix}A_2\\B_2\end{matrix} & \begin{matrix}A_3\\B_3\end{matrix} \\ \hline \begin{matrix}A_4\\B_4\end{matrix} & \begin{matrix}A_5\\B_5\end{matrix} & \begin{matrix}A_6\\B_6\end{matrix} \end{array}\right] \tag{8.11}$$

is convergent iff

$$\begin{aligned} &\min_{\substack{i,j\\ i\neq j}} \left(c_i^{(k)} + c_j^{(k)}\right) > 2 \qquad \forall\ k \in \{1,2,3,4,5,6\} \\ &\text{and} \quad \check{c}_i + \check{c}_j + \check{c}_k > 2 \qquad \forall\ (i,j,k) \in \{(1,2,6),(1,3,5),(2,3,4),(4,5,6)\} \\ &\text{and} \quad \check{c}_i + \check{c}_j + \check{c}_k + \check{c}_\ell > 2 \quad \forall\ (i,j,k,\ell) \in \{(1,2,4,5),(1,3,4,6),(2,3,5,6)\}. \end{aligned} \tag{8.12}$$

Here, the penultimate line corresponds to all closed three-point subgraphs, the last line corresponds to all closed four-point subgraphs.

The convergence conditions discussed so far only depend on the sums of the holomorphic and antiholomorphic labels of the edges. That this view tends to underestimate the convergence of the sum can be seen by considering the two one-loop graphs $\mathcal{C}\begin{bmatrix}1&0\\0&1\end{bmatrix}$ and $\mathcal{C}\begin{bmatrix}1&1\\0&0\end{bmatrix}$. According to our condition (8.4), both graphs should be equally divergent. But of course, while the sum $\mathcal{C}\begin{bmatrix}1&0\\0&1\end{bmatrix}$ is divergent, the sum $\mathcal{C}\begin{bmatrix}1&1\\0&0\end{bmatrix}$ is only conditionally convergent and we regularize it by introducing additional powers of the momentum as in (2.33), yielding $\widehat{\mathrm{G}}_2$. In general, graphs containing a $\begin{bmatrix}1&1\\0&0\end{bmatrix}$ subblock can be simplified using the divergent HSR discussed in section 8.4.

In the integral representation, this can be seen as follows: $f^{(1)}(z,\tau) \sim \frac{1}{z}$ is the only one out of the $f^{(a)}$ which has a pole. The fact that $\mathcal{C}\begin{bmatrix}1&1\\0&0\end{bmatrix}$ is conditionally convergent is reflected in the fact that the integral of $\frac{1}{z^2}$ over a ball around the origin vanishes, whereas the divergence of $\mathcal{C}\begin{bmatrix}1&0\\0&1\end{bmatrix}$ is reflected in the divergence of the integral of $|f^{(1)}(z)|^2 \sim \frac{1}{|z|^2}$.

In the `ModularGraphForms` package, the function `CCheckConv` checks for convergence of the argument using the criteria (8.4) and (8.6) on dihedral and trihedral graphs. The return value is either `True` for convergent MGFs or `False` for divergent MGFs, e.g.

```
In[44]:= CCheckConv[c[{{0, 1, 2}, {1, 0, 2}}]]

         CCheckConv[c[{{-1, 2}, {1, 1}, {1, 1}}]]

Out[44]= False

Out[45]= False .
```

On top of dihedral and trihedral graphs, `CCheckConv` also checks for E_k, G_k and $\overline{\mathrm{G}}_k$ with $k < 2$, all other expressions are treated as convergent. As soon as any divergent object is detected in the argument, `CCheckConv` returns `False`.

8.2. Divergent modular graph forms from Koba–Nielsen integrals

We study MGFs in order to expand Koba–Nielsen integrals comprising the Koba–Nielsen factor (2.5) and a polynomial in the functions $f^{(a)}(z,\tau)$ and $\overline{f^{(b)}(z,\tau)}$ given in (2.12b) and (2.12c). If this polynomial contains a factor $|f^{(1)}_{ij}|^2$ (where $f^{(1)}_{ij} = f^{(1)}(z_i - z_j)$), the MGFs in the expansion of the Koba–Nielsen integral are all divergent since $|f^{(1)}_{ij}|^2$ leads to a $\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}$ subblock, which violates the criterion (8.4).

However, the Koba–Nielsen factor regulates this divergence: since the Jacobi theta function satisfies $\theta_1(z,\tau) \sim z$ for small z, $\exp(s_{ij}G_{ij}) \sim |z_{ij}|^{-2s_{ij}}$ for small z_{ij}. Using integration-by-parts identities for the Koba–Nielsen integral, one can in fact show that a Koba–Nielsen integral with a $|f^{(1)}_{ij}|^2$ prefactor has a pole in the Mandelstams. Hence, the appearance of divergent MGFs is merely a signal that one has tried to Taylor-expand around a pole.

As an example, consider the two-point Koba–Nielsen integral

$$\int \mathrm{d}\mu_1 \left|f^{(1)}_{12}\right|^2 \mathrm{KN}_2, \tag{8.13}$$

whose naive α' expansion

$$\frac{\pi}{\tau_2}\mathrm{E}_1 - s_{12}\frac{\tau_2}{\pi}\,\mathcal{C}\begin{bmatrix}0 & 1 & 1\\ 1 & 0 & 1\end{bmatrix} - \frac{1}{2}s_{12}^2\left(\frac{\tau_2}{\pi}\right)^2\mathcal{C}\begin{bmatrix}0 & 1 & 1 & 1\\ 1 & 0 & 1 & 1\end{bmatrix} + \mathcal{O}(s_{12}^3) \tag{8.14}$$

exhibits divergent MGFs at every order in s_{12}. In order to make the pole in s_{12} manifest, consider the derivative [29]

$$\partial_{\bar{z}_2}\left(f^{(1)}_{12}\mathrm{KN}_2\right). \tag{8.15}$$

We now use (2.13) and

$$\partial_{z_j}\mathrm{KN}_n = \sum_{i\neq j} s_{ij} f^{(1)}(z_{ij},\tau)\mathrm{KN}_n, \tag{8.16}$$

which follows from (2.14), to evaluate (8.15). With this, we obtain

$$\partial_{\bar{z}_2}\left(f^{(1)}_{12}\mathrm{KN}_2\right) = \left(\frac{\pi}{\tau_2} - \pi\delta^{(2)}(z_{12},\bar{z}_{12})\right)\mathrm{KN}_2 + s_{12}\left|f^{(1)}_{12}\right|^2\mathrm{KN}_2. \tag{8.17}$$

Integrating over z_2 and solving for (8.13) yields (since $\mathrm{KN}_2 \to 0$ for $z_{12} \to 0$ the term with the delta function does not contribute)

$$\int \mathrm{d}\mu_1 \left| f_{12}^{(1)} \right|^2 \mathrm{KN}_2 = -\frac{1}{s_{12}} \frac{\pi}{\tau_2} \int \mathrm{d}\mu_1 \mathrm{KN}_2, \tag{8.18}$$

making the pole in s_{12} explicit. The remaining Koba–Nielsen integral in (8.18) has an expansion in convergent MGFs. Variations of this technique to expose the kinematic poles in Koba–Nielsen integrals can be found in countless examples in the literature.

At two points, the integral (8.18) is the only Koba–Nielsen integral with a pole in the Mandelstams and it is associated to the collision of the two punctures. At three point, several different pole structures can appear, including nested poles incorporating the three-particle Mandelstam variable

$$s_{123} = s_{12} + s_{13} + s_{23} \tag{8.19}$$

due to the collision of all three punctures. The rewriting of all relevant three-point integrals making the pole structure manifest and reducing divergent expansions to convergent ones as above, is summarized in appendix B.

In general, we can use the Fay identity (6.16) to rewrite the $f_{ij}^{(1)}$ contributions to the integrand in terms of $f_{ij}^{(a)}$ with $a > 1$ and derivatives of the Koba–Nielsen factor as in (8.16). When integrating these expressions by parts, we make one pole explicit and obtain an expression with poles of lower multiplicity.

8.3. Divergent modular graph forms from momentum conservation

Apart from the expansion of Koba–Nielsen integrals, divergent MGFs can also appear in momentum-conservation identities of convergent graphs. In the sum representation (5.20) of momentum conservation, this means that the exchange of the sum over edges e' and the sum over momenta p_{e} is not allowed in this case. Performing it anyway leads to the decomposition of a convergent series into a sum of divergent series. As an example, consider the convergent seed $\mathcal{C}\begin{bmatrix}0&1&2\\1&1&2\end{bmatrix}$, whose antiholomorphic momentum-conservation identity is

$$\mathcal{C}\begin{bmatrix}0&1&2\\1&0&2\end{bmatrix} + \mathcal{C}\begin{bmatrix}0&1&2\\1&1&1\end{bmatrix} + \left(\frac{\pi}{\tau_2}\right)^3 (\mathrm{E}_1\mathrm{E}_2 - \mathrm{E}_3) = 0, \tag{8.20}$$

after factorization. The graph $\mathcal{C}\begin{bmatrix}0&1&2\\1&0&2\end{bmatrix}$ and the Eisenstein series E_1 are both divergent.

When dealing only with convergent MGFs, momentum-conservation identities involving divergent graphs should be discarded. However, as we will discuss shortly, it is sometimes desirable to have identities between divergent MGFs and momentum-conservation identities involving divergent MGFs can be used to define those divergent MGFs. In this framework, we treat the divergent non-holomorphic Eisenstein series E_1 as a basis element for divergent MGFs and find decompositions in the same way as we did for convergent MGFs. E.g. (8.20), together with the (convergent) identity

$$\mathcal{C}\begin{bmatrix}0&1&2\\1&1&1\end{bmatrix} = -\frac{1}{2}\left(\frac{\pi}{\tau_2}\right)^3 (\mathrm{E}_3 - \zeta_3), \tag{8.21}$$

can be used to decompose the divergent graph $\mathcal{C}\begin{bmatrix}0&1&2\\1&0&2\end{bmatrix}$,

$$\mathcal{C}\begin{bmatrix}0&1&2\\1&0&2\end{bmatrix} = \left(\frac{\pi}{\tau_2}\right)^3 \left(\frac{3}{2}\mathrm{E}_3 - \mathrm{E}_1\mathrm{E}_2 + \frac{1}{2}\zeta_3\right). \tag{8.22}$$

Note that this does not extend to momentum-conservation identities of divergent seeds which have to be treated separately, cf section 8.6 below.

In particular, momentum-conservation identities involving divergent graphs can appear in the sieve algorithm, when removing entries of -1 as described in section 7.1. As an example for this phenomenon, consider the graph $\mathcal{C}\begin{bmatrix}0&1&2&3\\1&1&2&0\end{bmatrix}$, whose Cauchy–Riemann derivative is given by

$$\begin{aligned}\nabla^{(6)}\mathcal{C}\begin{bmatrix}0&1&2&3\\1&1&2&0\end{bmatrix} = 3\,\mathcal{C}\begin{bmatrix}0&1&2&4\\1&1&2&-1\end{bmatrix} + 2\,\mathcal{C}\begin{bmatrix}0&1&3&3\\1&1&1&0\end{bmatrix}\\ + \mathcal{C}\begin{bmatrix}0&2&2&3\\1&0&2&0\end{bmatrix}.\end{aligned} \tag{8.23}$$

The -1-entry in the first term can be removed by a momentum-conservation identity which yields, after factorization and divergent HSR (to be discussed below in section 8.4),

$$\begin{aligned}\mathcal{C}\begin{bmatrix}0&1&2&4\\1&1&2&-1\end{bmatrix} = 5\mathcal{C}\begin{bmatrix}0&2&5\\1&2&0\end{bmatrix} + \mathcal{C}\begin{bmatrix}1&2&4\\1&2&0\end{bmatrix} - \mathcal{C}\begin{bmatrix}0&1&2&4\\1&1&1&0\end{bmatrix}\\ - \mathrm{G}_4\mathcal{C}\begin{bmatrix}0&1&2\\1&0&2\end{bmatrix} - \widehat{\mathrm{G}}_2\mathcal{C}\begin{bmatrix}0&2&3\\1&2&0\end{bmatrix}\\ + \frac{\pi}{\tau_2}\left(\mathcal{C}\begin{bmatrix}0&2&4\\1&1&0\end{bmatrix} - \mathcal{C}\begin{bmatrix}6&0\\2&0\end{bmatrix}\right) + \left(\frac{\pi}{\tau_2}\right)^3\\ \times(\mathrm{E}_2 - \mathrm{E}_1\mathrm{E}_2)\,\mathrm{G}_4.\end{aligned} \tag{8.24}$$

As explained in section 7.1, when constructing identities with the sieve algorithm, we seek to cancel holomorphic Eisenstein series by adding suitable MGFs. In order to do this consistently, we need to know all relations for the MGFs in the prefactor of the holomorphic Eisenstein series. In the example (8.24), however, the prefactor of G_4 is

$$-\mathcal{C}\begin{bmatrix}0&1&2\\1&0&2\end{bmatrix} + \left(\frac{\pi}{\tau_2}\right)^3(\mathrm{E}_2 - \mathrm{E}_1\mathrm{E}_2) \tag{8.25}$$

and hence in particular involves divergent MGFs. I.e. in this case, we need to know the decomposition (8.22) to see explicitly that the divergence cancels out and to continue with the sieve algorithm.

In general, since (according to (5.47)) the action of the Cauchy–Riemann operator on MGFs leaves the sum of holomorphic and antiholomorphic labels for each edge invariant and the divergence conditions in section 8.1 are all functions of this sum only, each term in the derivative of an MGF $\mathcal{C}_\Gamma$ will have the same convergence properties as $\mathcal{C}_\Gamma$. Momentum conservation however increases the sum of the labels in one edge and decreases it in another edge in each term, therefore changing the convergence properties. But since the MGF decomposed in this way is convergent, the divergences have to cancel out upon plugging in identities for the divergent graphs.

For the remainder of this discussion, we will restrict to dihedral graphs, where the edge labels are written as columns in one block, but the arguments generalize straightforwardly to higher-point graphs. In [8], where the sieve algorithm was introduced, the authors restricted to the case of strictly positive holomorphic labels and non-negative antiholomorphic labels. In this case, the column sum for all edges is at least 2, with at most one $(1,0)$ edge since we assume that HSR is already performed. After taking the Cauchy–Riemann derivative, momentum conservation is only necessary in the term in which the $(1,0)$ edge is replaced by a $(2,-1)$ edge. In the momentum conservation identity, this edge will become a $(2,0)$ edge in each term, hence the column sum for each edge is again 2 with at most one edge of sum 1, i.e. each term is convergent. In this way, the problem of divergent MGFs in the sieve algorithm is avoided in [8] and the present discussion can therefore be regarded as an extension of the previously known techniques.

8.4. Divergent holomorphic subgraph reduction

On top of momentum conservation and factorization, HSR is a central technique to derive identities for MGFs. It is therefore desirable to extend HSR to divergent graphs. To this end, we will distinguish the case in which the divergence appears within the holomorphic subgraph, i.e. the sum of the labels of the edges forming the holomorphic subgraph is at most 2, from the case in which the divergence appears outside the holomorphic subgraph, i.e. the sum of labels within the holomorphic subgraph is at least 3, but the entire MGF is still divergent.

In the case of a divergence outside the holomorphic subgraph, the sum over the loop momentum, which is performed when doing HSR, is convergent. I.e. the divergence acts merely as a spectator and the formulas for two- and three-point HSR discussed in section 6 are still valid. E.g. dihedral graphs in which the divergence lies outside the holomorphic subgraph are given by $\mathcal{C}\begin{bmatrix}0&1&a&A\\1&0&0&B\end{bmatrix}$ with $a\geqslant 2$ and all column sums in $\begin{bmatrix}A\\B\end{bmatrix}$ at least two. In this case, we can apply the two-point HSR formula (6.6) and obtain results consistent with momentum conservation. For the graph $\mathcal{C}\begin{bmatrix}0&1&a&A\\1&0&0&B\end{bmatrix}$ with $a\geqslant 3$ we can see this explicitly by using the holomorphic momentum-conservation identity of the convergent seed $\mathcal{C}\begin{bmatrix}1&1&a&A\\1&0&0&B\end{bmatrix}$,

$$\begin{aligned}\mathcal{C}\begin{bmatrix}0&1&a&A\\1&0&0&B\end{bmatrix} = -\mathcal{C}\begin{bmatrix}1&1&a-1&A\\1&0&0&B\end{bmatrix} - \sum_{i=1}^{R}\mathcal{C}\begin{bmatrix}1&1&a&A-S_i\\1&0&0&B\end{bmatrix}\\ + \mathcal{C}\begin{bmatrix}1&a&A\\1&0&B\end{bmatrix} - \frac{\pi}{\tau_2}\mathrm{E}_1\mathrm{G}_a\prod_{i=1}^{R}\mathcal{C}\begin{bmatrix}a_i&0\\b_i&0\end{bmatrix},\end{aligned} \tag{8.26}$$

and applying the HSR formula (6.6) to the two convergent graphs on the rhs. Similar calculations can be done at three point and the extension of the HSR formulas to divergent graphs in this way was checked empirically for many cases.

If the holomorphic subgraph itself is divergent, the sum over the loop momentum which we perform when doing HSR is not convergent any more and hence we cannot use the usual HSR formulas in this case. If we restrict to only non-negative edge labels and assume that the graph under consideration has already been factorized (i.e. it does not contain any $(0,0)$ edges), then holomorphic subgraphs with more than two edges cannot be divergent. For this reason, we will restrict to the case of divergent two-point holomorphic subgraphs. In the sum representation, in which the two-point HSR formula (6.6) was derived first, it is unclear how to proceed in the

case of divergent sums. In the integral representation, however, in which the two-point HSR formula was derived from the coincident limit (6.25) of the Fay identity, it is straightforward to generalize (6.6) to divergent holomorphic subgraphs: we can just take the $a_1 = a_2 = 1$ case of (6.25),

$$\left(f^{(1)}(z)\right)^2 = 2f^{(2)}(z) - \widehat{\mathrm{G}}_2 - \partial_z f^{(1)}(z) \tag{8.27}$$

and integrate it against a product of $C^{(a,b)}(z)$ functions, as defined in (2.18), yielding

$$\mathcal{C}\begin{bmatrix}1 & 1 & A\\ 0 & 0 & B\end{bmatrix} = -2\,\mathcal{C}\begin{bmatrix}2 & A\\ 0 & B\end{bmatrix} - \widehat{\mathrm{G}}_2\,\mathcal{C}\begin{bmatrix}A\\ B\end{bmatrix} + \frac{\pi}{\tau_2}\,\mathcal{C}\begin{bmatrix}1 & A\\ -1 & B\end{bmatrix}. \tag{8.28}$$

Note that (6.6) has an additional term $\widehat{\mathrm{G}}_2\,\mathcal{C}\begin{bmatrix}0 & A\\ 0 & B\end{bmatrix}$ when naively extended to $a_+ = a_- = 1$. Empirically, we found that (8.28) is compatible with momentum conservation in a large number of cases. Furthermore, (8.28) agrees with the special cases

$$\mathcal{C}\begin{bmatrix}1 & 1 & a\\ 0 & 0 & b\end{bmatrix} = -2\,\mathcal{C}\begin{bmatrix}a+2 & 0\\ b & 0\end{bmatrix} + \frac{\pi}{\tau_2}\,\mathcal{C}\begin{bmatrix}a+1 & 0\\ b-1 & 0\end{bmatrix} \tag{8.29}$$

$$\mathcal{C}\begin{bmatrix}1 & 1 & 1 & 1\\ 0 & 0 & 1 & 1\end{bmatrix} = -2\mathcal{C}\begin{bmatrix}4 & 0\\ 2 & 0\end{bmatrix} - \left(\frac{\pi}{\tau_2}\right)^2\widehat{\mathrm{G}}_2(\mathrm{E}_2+2) + 4\frac{\pi}{\tau_2}\mathcal{C}\begin{bmatrix}3 & 0\\ 1 & 0\end{bmatrix} \tag{8.30}$$

which were obtained in [20], where (8.27) was derived in a different way than from the coincident limit of Fay identities.

The divergent two-point HSR identity (8.28) has a straightforward generalization to trihedral (and higher-point graphs),

$$\mathcal{C}\left[\begin{smallmatrix}1&1&A_1\\0&0&B_1\end{smallmatrix}\middle|\begin{smallmatrix}A_2\\B_2\end{smallmatrix}\middle|\begin{smallmatrix}A_3\\B_3\end{smallmatrix}\right] = -2\,\mathcal{C}\left[\begin{smallmatrix}2&A_1\\0&B_1\end{smallmatrix}\middle|\begin{smallmatrix}A_2\\B_2\end{smallmatrix}\middle|\begin{smallmatrix}A_3\\B_3\end{smallmatrix}\right] - \widehat{\mathrm{G}}_2\,\mathcal{C}\left[\begin{smallmatrix}A_1\\B_1\end{smallmatrix}\middle|\begin{smallmatrix}A_2\\B_2\end{smallmatrix}\middle|\begin{smallmatrix}A_3\\B_3\end{smallmatrix}\right] + \frac{\pi}{\tau_2}\,\mathcal{C}\left[\begin{smallmatrix}1&A_1\\-1&B_1\end{smallmatrix}\middle|\begin{smallmatrix}A_2\\B_2\end{smallmatrix}\middle|\begin{smallmatrix}A_3\\B_3\end{smallmatrix}\right]. \tag{8.31}$$

The only kind of divergent HSR which cannot be treated in this way occurs if the holomorphic subgraph has a higher-point divergence, since this necessarily means that the holomorphic subgraph involves negative labels.

One might be tempted to also extend the trihedral Fay identity (6.22) to divergent graphs. However, this was found to lead to contradictions, as illustrated in the following: consider the divergent trihedral graph $\mathcal{C}\left[\begin{smallmatrix}1&0\,1&0\,1\\0&1\,0&2\,0\end{smallmatrix}\right]$ and simplify it once by performing three-point HSR and once by applying (6.22) to the first column and to the second column of the third block, yielding the decompositions

$$\mathcal{C}\left[\begin{smallmatrix}1\\0\end{smallmatrix}\middle|\begin{smallmatrix}0&1\\1&0\end{smallmatrix}\middle|\begin{smallmatrix}0&1\\2&0\end{smallmatrix}\right] \stackrel{?}{=} \widehat{\mathrm{G}}_2\,\mathcal{C}\left[\begin{smallmatrix}1&0\\3&0\end{smallmatrix}\right] + \frac{1}{2}\left(\frac{\pi}{\tau_2}\right)^3(\mathrm{E}_1(\mathrm{E}_1 - 4\mathrm{E}_2 + 2) - 3\mathrm{E}_2 + 5\mathrm{E}_3 - \zeta_3) \tag{8.32a}$$

$$\mathcal{C}\left[\begin{smallmatrix}1\\0\end{smallmatrix}\middle|\begin{smallmatrix}0&1\\1&0\end{smallmatrix}\middle|\begin{smallmatrix}0&1\\2&0\end{smallmatrix}\right] \stackrel{?}{=} \widehat{\mathrm{G}}_2\,\mathcal{C}\left[\begin{smallmatrix}1&0\\3&0\end{smallmatrix}\right] + \frac{\pi}{\tau_2}\,\mathcal{C}\left[\begin{smallmatrix}0&1\\1&-1\\2&2\end{smallmatrix}\right] + \frac{1}{2}\left(\frac{\pi}{\tau_2}\right)^3(4\mathrm{E}_1\mathrm{E}_2 - 5\mathrm{E}_3 + \zeta_3). \tag{8.32b}$$

Applying the Fay identity (6.22) to any other pair of holomorphic or antiholomorphic columns also leads to (8.32a). Together, (8.32a) and (8.32b) imply

$$\mathcal{C}\begin{bmatrix}0 & 1 & 1\\ 1 & -1 & 2\end{bmatrix} \stackrel{?}{=} \frac{1}{2}\left(\frac{\pi}{\tau_2}\right)^2 (\mathrm{E}_1^2 + 2\mathrm{E}_1 - 3\mathrm{E}_2). \tag{8.33}$$

Next, consider the divergent trihedral graph $\mathcal{C}\left[\begin{array}{c|cc|cc}0 & 0 & 1 & 1 & 1\\ 1 & 2 & 0 & 0 & 0\end{array}\right]$ which can be decomposed via two-point HSR and Fay into

$$\mathcal{C}\left[\begin{smallmatrix}0\\1\end{smallmatrix}\middle|\begin{smallmatrix}0&1\\2&0\end{smallmatrix}\middle|\begin{smallmatrix}1&1\\0&0\end{smallmatrix}\right] \stackrel{?}{=} -\left(\frac{\pi}{\tau_2}\right)^3 (\mathrm{E}_1 - 2\mathrm{E}_2 + \mathrm{E}_3 - \zeta_3) \tag{8.34a}$$

$$\mathcal{C}\left[\begin{smallmatrix}0\\1\end{smallmatrix}\middle|\begin{smallmatrix}0&1\\2&0\end{smallmatrix}\middle|\begin{smallmatrix}1&1\\0&0\end{smallmatrix}\right] \stackrel{?}{=} -\frac{\pi}{\tau_2}\,\mathcal{C}\left[\begin{smallmatrix}0&1&1\\1&-1&2\end{smallmatrix}\right] + \frac{1}{2}\left(\frac{\pi}{\tau_2}\right)^3 (\mathrm{E}_1^2 - 2\mathrm{E}_1 + \mathrm{E}_2 - 2\mathrm{E}_3 + 2\zeta_3)\,, \tag{8.34b}$$

respectively, yielding the identity

$$\mathcal{C}\begin{bmatrix}0 & 1 & 1\\ 1 & -1 & 2\end{bmatrix} \stackrel{?}{=} \frac{1}{2}\left(\frac{\pi}{\tau_2}\right)^2 (\mathrm{E}_1^2 - 3\mathrm{E}_2), \tag{8.35}$$

differing form (8.33) by a term $\frac{\pi}{\tau_2}\mathrm{E}_1$. For this reason, we will not apply the Fay identity (6.22) to divergent graphs.

In the `Mathematica` package `ModularGraphForms`, divergent HSR is implemented in the functions `DiCSimplify` and `TriCSimplify`, along with the convergent HSR. If divergent HSR is performed or not, is controlled by the Boolean option `divHSR`. Dihedral and trihedral HSR can be activated and deactivated individually with the Boolean options `diDivHSR` and `triDivHSR`. The default values of all these options are `True`.

8.5. Taking derivatives of divergent graphs

It would be desirable to apply the sieve algorithm discussed in section 7 also to divergent MGFs to derive decompositions of divergent MGFs which are e.g. useful to perform the sieve algorithm on convergent MGFs. In order to do this, we have to take derivatives of divergent MGFs. Unfortunately, this is not straightforward and, if done naively, contradictions to momentum-conservation identities can arise. As above, we will restrict in this section to two-point divergences occurring within one edge bundle since higher-point divergences are only relevant for graphs with negative entries.

Empirically, we found that taking derivatives of divergent MGFs using the formula (5.47) is consistent with momentum conservation if the divergence has the form $\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}$, however, a complete understanding of the structure of derivatives of these divergences is still lacking. If the divergence has the form $\begin{bmatrix}1 & 1\\ 0 & 0\end{bmatrix}$, we can first apply the divergent HSR formula (8.28), leading to a modification of the usual derivative expression (5.47). E.g. consider the graph $\mathcal{C}\begin{bmatrix}0 & 0 & A\\ 1 & 1 & B\end{bmatrix}$ with all column sums in $\begin{bmatrix}A\\ B\end{bmatrix}$ at least 2. Using divergent HSR (8.28), it can be rewritten to

$$\mathcal{C}\begin{bmatrix}0 & 0 & A\\ 1 & 1 & B\end{bmatrix} = -2\,\mathcal{C}\begin{bmatrix}0 & A\\ 2 & B\end{bmatrix} - \widehat{\mathrm{G}}_2\mathcal{C}\begin{bmatrix}A\\ B\end{bmatrix} + \frac{\pi}{\tau_2}\,\mathcal{C}\begin{bmatrix}-1 & A\\ 1 & B\end{bmatrix}. \tag{8.36}$$

Taking the derivative via (5.49) and using (8.28) to write the result back into a graph with a holomorphic subgraph yields

$$\nabla^{(|A|)}\,\mathcal{C}\begin{bmatrix}0&0&A\\1&1&B\end{bmatrix}=\sum_{i=1}^{R}\mathcal{C}\begin{bmatrix}0&0&A+S_i\\1&1&B-S_i\end{bmatrix}-\frac{\pi}{\tau_2}\prod_{i=1}^{R}\mathcal{C}\begin{bmatrix}a_i&0\\b_i&0\end{bmatrix}, \tag{8.37}$$

with an additional term as compared to a naive application of (5.49) on $\mathcal{C}\begin{bmatrix}0&0&A\\1&1&B\end{bmatrix}$.

Aside from HSR, this additional term can also be understood as arising from the derivative of the regularization term implicitly contained in $\mathcal{C}\begin{bmatrix}0&0&A\\1&1&B\end{bmatrix}$. To see this, we first write the regularization term explicitly,

$$\mathcal{C}\begin{bmatrix}0&0&A\\1&1&B\end{bmatrix}=\lim_{s\to 0}\mathcal{C}\begin{bmatrix}s&0&A\\s+1&1&B\end{bmatrix} \tag{8.38}$$

and exchange the limit and the differential, resulting in

$$\nabla^{(|A|)}\,\mathcal{C}\begin{bmatrix}0&0&A\\1&1&B\end{bmatrix}=\lim_{s\to 0}s\,\mathcal{C}\begin{bmatrix}s+1&0&A\\s&1&B\end{bmatrix}+\sum_{i=1}^{R}\mathcal{C}\begin{bmatrix}0&0&A+S_i\\1&1&B-S_i\end{bmatrix}. \tag{8.39}$$

Next, we rewrite the first term using the momentum-conservation identity of the seed-MGF $\mathcal{C}\begin{bmatrix}s+1&0&A\\s+1&1&B\end{bmatrix}$, which is convergent for all $s\geqslant 0$, and factorization, yielding

$$\begin{aligned}\lim_{s\to 0}s\,\mathcal{C}\begin{bmatrix}s+1&0&A\\s&1&B\end{bmatrix}&=-\lim_{s\to 0}s\left(\mathcal{C}\begin{bmatrix}s+1&0&A\\s+1&0&B\end{bmatrix}-\sum_{i=1}^{R}\mathcal{C}\begin{bmatrix}s+1&0&A\\s+1&0&B-S_i\end{bmatrix}\right)\\&=-\lim_{s\to 0}s\left(\left(\frac{\pi}{\tau_2}\right)^{s+1}\mathrm{E}_{s+1}\prod_{i=1}^{R}\mathcal{C}\begin{bmatrix}a_i&0\\b_i&0\end{bmatrix}-\mathcal{C}\begin{bmatrix}s+1&A\\s+1&B\end{bmatrix}\right.\\&\qquad\left.-\sum_{i=1}^{R}\mathcal{C}\begin{bmatrix}s+1&0&A\\s+1&0&B-S_i\end{bmatrix}\right).\end{aligned} \tag{8.40}$$

The last two terms in (8.40) are convergent for all $s\geqslant 0$ and hence drop out after taking the limit. E_1 however is divergent and with the first Kronecker limit formula

$$\mathrm{E}_{s+1}=\frac{1}{s}+\mathcal{O}(s^0), \tag{8.41}$$

we obtain

$$\lim_{s\to 0}s\,\mathcal{C}\begin{bmatrix}s+1&0&A\\s&1&B\end{bmatrix}=-\frac{\pi}{\tau_2}\prod_{i=1}^{R}\mathcal{C}\begin{bmatrix}a_i&0\\b_i&0\end{bmatrix}. \tag{8.42}$$

Plugging this into (8.39) yields (8.37), the result previously obtained from divergent HSR[10].

Similarly to (8.37), we take the derivative of terms of the form $\mathcal{C}\begin{bmatrix}1&1&A\\0&0&B\end{bmatrix}$ by first applying the formula (8.28) and then the usual expression (5.49) for the derivative. The generalization to higher-point graphs with two-point divergences is straightforward.

Since the techniques outlined in this section to take derivatives of divergent MGFs are conjectural and subtle, in the implementation in the `ModularGraphForms` package, a warning is issued whenever the functions `CHolCR` and `CAHolCR` encounter a divergent graph in their argument. If the Boolean option `divDer` of these functions is set to `False` (the default is `True`), `Nothing` is returned if it is divergent. If `divDer` is set to `True`, divergent derivatives are treated exactly like convergent ones, only (divergent) HSR is performed on the input before the derivative is taken.

8.6. Divergent momentum conservation and factorization

Naively performing momentum conservation of divergent seeds and factorization leads to inconsistencies, e.g. consider the holomorphic momentum-conservation identity of the seed $\mathcal{C}\begin{bmatrix}1&1&2\\0&0&3\end{bmatrix}$ which is naively[11]

$$\mathcal{C}\begin{bmatrix}1&1&1\\0&0&3\end{bmatrix}\overset{?}{=}-2\,\mathcal{C}\begin{bmatrix}0&1&2\\0&0&3\end{bmatrix}\overset{?}{=}-2\,\mathcal{C}\begin{bmatrix}1&0\\0&0\end{bmatrix}\mathcal{C}\begin{bmatrix}2&0\\0&3\end{bmatrix}-2\left(\frac{\pi}{\tau_2}\right)^3\mathrm{E}_3$$
$$\overset{?}{=}-2\left(\frac{\pi}{\tau_2}\right)^3\mathrm{E}_3, \tag{8.43}$$

where the first term vanishes due to odd label sums in both MGFs. The divergent HSR formula (8.28) however $\left(\text{and also momentum conservation of the convergent seed } \mathcal{C}\begin{bmatrix}1&1&2\\0&3&0\end{bmatrix}\right)$ leads to

$$\mathcal{C}\begin{bmatrix}1&1&1\\0&0&3\end{bmatrix}=\left(\frac{\pi}{\tau_2}\right)^3(\mathrm{E}_2-2\mathrm{E}_3), \tag{8.44}$$

contradicting (8.43). In this section, we will discuss some of the phenomena that arise in divergent momentum conservation and factorization but leave a complete understanding to the future.

The additional term in (8.44) can be understood in the integral representation of the MGF as follows: consider the graph

$$\mathcal{C}\begin{bmatrix}0&1&A\\0&0&B\end{bmatrix}=\int_\Sigma\frac{\mathrm{d}^2z}{\tau_2}C^{(0,0)}(z)f^{(1)}(z)\prod_{i=1}^{R}C^{(a_i,b_i)}(z), \tag{8.45}$$

[10] Note that, to find this agreement, it is crucial that we do not simplify (8.36) using $\mathcal{C}\begin{bmatrix}2&0\\0&0\end{bmatrix}=\widehat{\mathrm{G}}_2$ before taking the derivative, since this contains a regularization term again, whose derivative we would have to take into account.

[11] We will see below that the first two equalities are not correct for divergent graphs. The last equality is too naive because $\mathcal{C}\begin{bmatrix}1&0\\0&0\end{bmatrix}$ is conditionally convergent and can yield infinity, depending on the summation prescription used.

where $\begin{bmatrix}A\\B\end{bmatrix}$ contains no $\begin{bmatrix}1\\0\end{bmatrix}$, $\begin{bmatrix}0\\1\end{bmatrix}$ or $\begin{bmatrix}1\\1\end{bmatrix}$ columns. We saw in (5.33) that $C^{(0,0)}(z) = \tau_2\delta(z,\bar{z}) - 1$, leading to the usual factorization rule. In (8.45), the delta function instructs to take the $z \to 0$ limit of $f^{(1)}(z)\prod_{i=1}^{R} C^{(a_i,b_i)}(z)$. But since $f^{(1)}(z)$ has Laurent expansion

$$f^{(1)}(z) = \frac{1}{z} - z\widehat{\mathrm{G}}_2 - \bar{z}\frac{\pi}{\tau_2} + \mathcal{O}(z,\bar{z})^3 \tag{8.46}$$

and in particular a pole at 0, we have to expand the product to first order to obtain

$$\begin{aligned}\lim_{z\to 0} f^{(1)}(z)\prod_{i=1}^{R} C^{(a_i,b_i)}(z) &= \left(\partial_z \prod_{i=1}^{R} C^{(a_i,b_i)}(z)\right)_{z=0}\\ &= -\frac{\pi}{\tau_2}\sum_{i=1}^{R}\prod_{j=1}^{R}\mathcal{C}\begin{bmatrix}a_j & 0\\ b_j-\delta_{ij} & 0\end{bmatrix},\end{aligned} \tag{8.47}$$

using (5.22) and the fact that the product vanishes at zero since $|A| + |B|$ is odd if $\mathcal{C}\begin{bmatrix}0 & 1 & A\\ 0 & 0 & B\end{bmatrix}$ is non-trivial. This yields the modified factorization rule

$$\mathcal{C}\begin{bmatrix}0 & 1 & A\\ 0 & 0 & B\end{bmatrix} = -\frac{\pi}{\tau_2}\sum_{i=1}^{R}\prod_{j=1}^{R}\mathcal{C}\begin{bmatrix}a_j & 0\\ b_j-\delta_{ij} & 0\end{bmatrix} - \mathcal{C}\begin{bmatrix}1 & A\\ 0 & B\end{bmatrix}. \tag{8.48}$$

If more $\begin{bmatrix}1\\0\end{bmatrix}$ columns are present, higher derivatives of the remaining graphs have to be taken. If $\begin{bmatrix}A\\B\end{bmatrix}$ contains a $\begin{bmatrix}1\\1\end{bmatrix}$ column, corresponding to a Green function in the integral, we have to iterate this procedure, since the derivative of the Green function is $f^{(1)}$ (cf (2.14)) and hence contains again a pole. In this way we obtain for the MGF $\mathcal{C}\begin{bmatrix}0 & 1 & 1_n & A\\ 0 & 0 & 1_n & B\end{bmatrix}$, where 1_n is the row vector with n entries of 1, the factorization rule

$$\begin{aligned}\mathcal{C}\begin{bmatrix}0 & 1 & 1_n & A\\ 0 & 0 & 1_n & B\end{bmatrix} &= \left(\frac{\pi}{\tau_2}\right)^{n+1}\sum_{i=1}^{R}\prod_{j=1}^{R}\mathcal{C}\begin{bmatrix}a_j & 0\\ b_j-\delta_{ij} & 0\end{bmatrix}\\ &\quad\times\sum_{k=0}^{n}(-1)^{k+1}\frac{n!}{(n-k)!}\mathrm{E}_1^{n-k} - \mathcal{C}\begin{bmatrix}1 & 1_n & A\\ 0 & 1_n & B\end{bmatrix}.\end{aligned} \tag{8.49}$$

For trihedral graphs, we have similarly

$$\begin{aligned}\mathcal{C}\left[\begin{smallmatrix}0 & 1 & 1_n & A_1\\ 0 & 0 & 1_n & B_1\end{smallmatrix}\middle|\begin{smallmatrix}A_2\\ B_2\end{smallmatrix}\middle|\begin{smallmatrix}A_3\\ B_3\end{smallmatrix}\right] &= (-1)^{|2|}\left(\frac{\pi}{\tau_2}\right)^{n+1}\mathcal{C}\left[\begin{smallmatrix}A_2 & A_3\\ B_2 & B_3\end{smallmatrix}\right]\sum_{i=1}^{R_1}\prod_{j=1}^{R_1}\mathcal{C}\left[\begin{smallmatrix}a_1^{(j)} & 0\\ b_1^{(j)}-\delta_{ij} & 0\end{smallmatrix}\right]\sum_{k=0}^{n}(-1)^{k+1}\frac{n!}{(n-k)!}\mathrm{E}_1^{n-k}\\ &\quad - \mathcal{C}\left[\begin{smallmatrix}1 & 1_n & A_1\\ 0 & 1_n & B_1\end{smallmatrix}\middle|\begin{smallmatrix}A_2\\ B_2\end{smallmatrix}\middle|\begin{smallmatrix}A_3\\ B_3\end{smallmatrix}\right].\end{aligned} \tag{8.50}$$

In general, the Laurent expansion of $f^{(n)}$ contains a term $\sim \frac{\bar{z}^{n-1}}{z}$, which vanishes at the origin for $n \geqslant 3$. The $z \to 0$ limit of $f^{(2)}$ depends on the direction in which the origin is approached, but

$\frac{\bar{z}}{z}$ vanishes when integrated against a delta function due to the angular part of the integration. Therefore, only the case of $f^{(1)}$ yields additional terms as discussed above.

When (8.48) is used in (8.43), we obtain the correct additional term, up to a factor of 2, which arose in the momentum-conservation identity from the product rule of $\partial_{\bar{z}}$ acting on $f^{(1)}$ (cf (5.23)). This spurious factor of 2 is again due to the pole in $f^{(1)}$, as can be understood by considering the integral

$$\int_{B_r(0)} \mathrm{d}^2 z\, \partial_{\bar{z}} \left(\frac{1}{z^2}\right) z, \tag{8.51}$$

where $\mathrm{B}_r(0)$ is the ball of radius r around 0. Evaluating (8.43) using $\partial_{\bar{z}} \left(\frac{1}{z}\right) = \pi \delta^{(2)}(z)$ and the product rule leads to

$$\int_{B_r(0)} \mathrm{d}^2 z\, \partial_{\bar{z}} \left(\frac{1}{z^2}\right) z \overset{?}{=} 2 \int_{B_r(0)} \mathrm{d}^2 z\, \frac{1}{z} \partial_{\bar{z}} \left(\frac{1}{z}\right) z = 2\pi, \tag{8.52}$$

whereas the factor of 2 is absent if we apply Stokes' theorem,

$$\int_{B_r(0)} \mathrm{d}^2 z\, \partial_{\bar{z}} \left(\frac{1}{z^2}\right) z = \frac{1}{2\mathrm{i}} \oint_{\partial B_r(0)} \mathrm{d}z\, \frac{1}{z} = \pi \operatorname*{Res}_{z=0} \left(\frac{1}{z}\right) = \pi. \tag{8.53}$$

Empirically, momentum-conservation identities of seeds with a divergence of the form $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ seem to be consistent, but we have not investigated them any further. For trihedral graphs, if the two blocks adjacent to the vertex used for momentum conservation are convergent and no three-point divergence appears in the graph, the resulting momentum-conservation identity is valid. If these conditions are not met, the same care has to be taken as with the dihedral graphs.

In the `ModularGraphForms` package, the modified factorization rules (8.49) and (8.50) are implemented in the functions `DiCSimplify` and `TriCSimplify`, but since they are not tested as thoroughly as the convergent manipulations, a warning is issued if these special cases are encountered. If more than one $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$ or $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$ column appears next to a $\begin{bmatrix} 0 \\ 0 \end{bmatrix}$ column, the input is returned. The momentum conservation functions `DiHolMomConsId` and `TriHolMomConsId` and their complex conjugates issue a warning when the seed is divergent.

As we will see in the next section, the basis decompositions of MGFs obtained in this paper rely on manipulations of divergent MGFs only for the modular weights $(6, 6)$ and $(7, 5)$ (and its complex conjugate). The expansion of the generating function of Koba–Nielsen integrals at two- and three points involving these sectors was checked to satisfy the Cauchy–Riemann equations derived in [29]. Furthermore, the Laurent polynomials of this expansion were checked against the closed formula for two-point Laurent polynomials given in [30].

9. Basis decompositions

By combining the techniques discussed in the sections above, we can systematically generate identities for MGFs, starting from a small number of known relations. In the end, we obtain decompositions of a large class of complicated MGFs into a small number of simple graphs. That these actually a basis for all MGFs can be proven using techniques from iterated Eisenstein integrals discussed in [30].

In the `ModularGraphFormsMathematica` package, decompositions for all dihedral and trihedral convergent MGFs with non-negative edge labels of modular weight (a, b) with $a + b \leqslant 12$ are given, starting just from the dihedral identities

$$D_3 = \mathrm{E}_3 + \zeta_3 \tag{9.1}$$

$$D_5 = 60C_{1,1,3} + 10D_3\mathrm{E}_2 - 48\mathrm{E}_5 + 16\zeta_5, \tag{9.2}$$

where D_ℓ is defined in (2.25) and $C_{a,b,c}$ in (2.28). These two identities are also the only source of zeta-values in the basis decompositions.

9.1. Systematic derivation of identities

In order to apply the techniques discussed above systematically, we consider subspaces with total modular weight $a + b =$ const. of the space of all MGFs and derive all identities in one subspace before continuing to the next higher total weight.

Within each subspace, we start by considering weight $a = b$ which corresponds to MGFs which are modular invariant after multiplication by τ_2^a. The identities in this space are generated by combining momentum conservation with Fay identities:

- We write down all convergent dihedral and trihedral MGFs of weight $(a + 1, a)$ and $(a, a + 1)$ without closed holomorphic subgraphs and use them as seeds to generate holomorphic and antiholomorphic momentum-conservation identities, respectively. Closed holomorphic subgraphs in the seeds would necessarily lead to negative labels in the identity which could not be removed by momentum conservation.
- We write down all convergent trihedral MGFs of weight (a, a), including those which contain closed holomorphic subgraphs and apply the Fay identity (6.22) in all possible ways.

Afterward, we remove all relations which contain divergent MGFs after topological simplifications and factorizations. Then, we simplify the remaining identities using HSR, the (generalized) Ramanujan identities discussed in section 5.5 and identities known from lower total modular weight and expand holomorphic Eisenstein series in the ring spanned by G_4 and G_6. The resulting large system of linear equations, together with the identities (9.1) and (9.2) can then be solved for all convergent dihedral and trihedral MGFs which do not appear in the basis.

After the $a = b$ sector, we continue with the weight-$(a + k, a - k)$ sectors with $k = 1, \ldots, a - 1$ as follows: in addition to the momentum-conservation and Fay identities for these sectors, we also take the Cauchy–Riemann derivative of all basis decompositions in the $(a + k - 1, a - k + 1)$ sector (excluding MGFs containing closed holomorphic subgraphs), which were found before. Again, we remove all relations containing divergent MGFs. Finally, we take the complex conjugate of all identities obtained, to also cover the $k < 0$ sectors.

In this way, basis decompositions for all convergent dihedral and trihedral MGFs can be found with total modular weight $a + b \leqslant 10$. The number of these MGFs is listed in table 2. Note that we did not need to use the sieve algorithm in this process, hence we do not have undetermined integration constants in the basis decompositions.

Although the strategy outlined above is successful in the $a + b \leqslant 10$ sectors, at weight $(6, 6)$, it is not sufficient to decompose all trihedral MGFs. To obtain the decompositions of these graphs as well, we keep the momentum-conservation identities containing divergent graphs and simplify them using the divergent HSR outlined in section 8.4 if possible (both divergent holomorphic subgraphs and divergences outside of the holomorphic subgraph appear). In this

Table 2. Number of convergent dihedral and trihedral MGFs with non-negative edge labels, excluding products. For graphs containing closed holomorphic subgraphs, no basis decompositions need to be found independently, they are implied by HSR and the basis decompositions of the non-HSR graphs.

Weight	Dihedral non-HSR	Dihedral HSR	Trihedral non-HSR	Trihedral HSR
(1, 1)	0	0	0	0
(2, 2)	1	0	0	0
(3, 1)	1	0	0	0
(3, 3)	7	2	0	0
(4, 2)	5	3	0	0
(5, 1)	1	4	0	0
(4, 4)	27	10	28	20
(5, 3)	22	12	17	25
(6, 2)	11	16	0	29
(7, 1)	1	14	0	12
(5, 5)	83	40	326	248
(6, 4)	73	44	247	291
(7, 3)	47	50	91	322
(8, 2)	19	50	0	243
(9, 1)	1	35	0	94
(6, 6)	228	138	2236	2044
(7, 5)	206	142	1844	2191
(8, 4)	150	154	990	2359
(9, 3)	83	149	276	2008
(10, 2)	29	124	0	1207
(11, 1)	1	74	0	439
Total	996	1061	6055	11 532

way, we can decompose all graphs in the (6, 6) and (7, 5) sectors. For the remaining sectors in table 2, the convergent identities are sufficient again.

In this way, basis decompositions for 1646 dihedral and 9520 trihedral convergent MGFs with non-negative edge labels and without closed holomorphic subgraphs were found and implemented in the functions `DiCSimplify` and `TriCSimplify` of the `ModularGraphForms` package. Since `CSimplify` calls `DiCSimplify` and `TriCSimplify`, we have e.g.

```
In[46]:= CSimplify[c[{{1, 1}, {1, 1}}]]

         CSimplify[c[{{1, 1}, {1, 1}, {1, 1}}]]
```

$$\text{Out[46]}= 24\,\mathrm{C}\left[\begin{smallmatrix}1&1&2\\1&1&2\end{smallmatrix}\right] + \frac{3\,\pi^4\,\mathrm{E}_2^2}{\tau_2^4} - \frac{18\,\pi^4\,\mathrm{E}_4}{\tau_2^4}$$

$$\text{Out[47]}= 2\,\mathrm{C}\left[\begin{smallmatrix}1&1&3\\1&1&3\end{smallmatrix}\right] - \frac{2\,\pi^5\,\mathrm{E}_5}{5\,\tau_2^5} + \frac{3\,\pi^5\,\zeta_5}{10\,\tau_2^5}\ .$$

All the basis decompositions contained in the `ModularGraphForms` package were checked to be compatible with the Cauchy–Riemann equation of the generating series of

Koba–Nielsen integrals discussed in [29] at two- and three points. The decompositions of MGFs with $a + b \leqslant 10$ were used in [30] to find representations of MGFs in terms of iterated Eisenstein integrals via this generating series.

9.2. Bases for modular graph forms

Using the procedure outlined in section 9.1, we obtain decompositions for many MGFs, which leave as independent MGFs only the ones listed in table 3. That these form indeed a basis of all MGFs (not just two- and three-point graphs) at the corresponding weights can be proven using iterated Eisenstein integrals and generating functions of Koba–Nielsen integrals [30]. The basis elements in the sector (a, b) with $a < b$ are given by complex conjugation. Furthermore, basis elements containing a holomorphic Eisenstein series are not listed in table 3, since they can be constructed from the bases at lower weights, e.g. the $(6, 4)$ sector contains the additional basis elements $\mathrm{G}_4\mathcal{C}\begin{bmatrix}2 & 0\\ 4 & 0\end{bmatrix}$ and $\mathrm{G}_6\overline{\mathrm{G}}_4$. In the following, we will refer to basis elements given as products as *reducible* and to the remaining ones as *irreducible*. On top of various MGFs, we have included in table 3 also the constants ζ_3, ζ_5 and ζ_3^2 in the relevant sectors.

Note that starting from total modular weight 10, the sector with equal holomorphic and antiholomorphic weight contains cusp forms. Specifically, in the basis of the $(5, 5)$ sector, the three cusp forms

$$\mathcal{C}\begin{bmatrix}1 & 0\\ 3 & 0\end{bmatrix}\mathcal{C}\begin{bmatrix}4 & 0\\ 2 & 0\end{bmatrix} - \mathcal{C}\begin{bmatrix}3 & 0\\ 1 & 0\end{bmatrix}\mathcal{C}\begin{bmatrix}2 & 0\\ 4 & 0\end{bmatrix} \tag{9.3a}$$

$$\mathcal{A}\begin{bmatrix}0 & 2 & 3\\ 3 & 0 & 2\end{bmatrix} \tag{9.3b}$$

$$\mathcal{A}\begin{bmatrix}0 & 1 & 2 & 2\\ 1 & 1 & 0 & 3\end{bmatrix} \tag{9.3c}$$

appear. Similarly, the $(6, 6)$ basis contains the cusp forms

$$\mathcal{C}\begin{bmatrix}3 & 0\\ 1 & 0\end{bmatrix}\mathcal{C}\begin{bmatrix}3 & 0\\ 5 & 0\end{bmatrix} - \mathcal{C}\begin{bmatrix}1 & 0\\ 3 & 0\end{bmatrix}\mathcal{C}\begin{bmatrix}5 & 0\\ 3 & 0\end{bmatrix} \tag{9.4a}$$

$$\mathcal{C}\begin{bmatrix}3 & 0\\ 1 & 0\end{bmatrix}\mathcal{C}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 3\end{bmatrix} - \mathcal{C}\begin{bmatrix}1 & 0\\ 3 & 0\end{bmatrix}\mathcal{C}\begin{bmatrix}1 & 1 & 3\\ 1 & 1 & 1\end{bmatrix} \tag{9.4b}$$

$$\mathcal{A}\begin{bmatrix}0 & 2 & 4\\ 5 & 0 & 1\end{bmatrix} \tag{9.4c}$$

$$\mathcal{A}\begin{bmatrix}0 & 2 & 2 & 2\\ 3 & 0 & 1 & 2\end{bmatrix} \tag{9.4d}$$

$$\mathcal{A}\begin{bmatrix}0 & 1 & 2 & 3\\ 2 & 1 & 3 & 0\end{bmatrix}. \tag{9.4e}$$

Table 3. Basis elements used in the `ModularGraphForms` package for (convergent) MGFs of weight $a+b \leqslant 12$, excluding holomorphic Eisenstein series. The counting includes zeta values.

Weight	# basis elements	Basis elements
$(2,2)$	1	$\left(\frac{\pi}{\tau_2}\right)^2 \mathrm{E}_2$
$(3,1)$	1	$\mathcal{C}\begin{bmatrix}3&0\\1&0\end{bmatrix}$
$(3,3)$	2	$\left(\frac{\pi}{\tau_2}\right)^3 \mathrm{E}_3,\ \left(\frac{\pi}{\tau_2}\right)^3 \zeta_3$
$(4,2)$	1	$\mathcal{C}\begin{bmatrix}4&0\\2&0\end{bmatrix}$
$(5,1)$	1	$\mathcal{C}\begin{bmatrix}5&0\\1&0\end{bmatrix}$
$(4,4)$	4	$\left(\frac{\pi}{\tau_2}\right)^4 \mathrm{E}_4,\ \mathcal{C}\begin{bmatrix}1&1&2\\1&1&2\end{bmatrix},\ \left(\frac{\pi}{\tau_2}\right)^4 \mathrm{E}_2^2,\ \mathcal{C}\begin{bmatrix}1&0\\3&0\end{bmatrix}\mathcal{C}\begin{bmatrix}3&0\\1&0\end{bmatrix}$
$(5,3)$	3	$\mathcal{C}\begin{bmatrix}5&0\\3&0\end{bmatrix},\ \mathcal{C}\begin{bmatrix}1&1&3\\1&1&1\end{bmatrix},\ \left(\frac{\pi}{\tau_2}\right)^2 \mathrm{E}_2\mathcal{C}\begin{bmatrix}3&0\\1&0\end{bmatrix}$
$(6,2)$	2	$\mathcal{C}\begin{bmatrix}6&0\\2&0\end{bmatrix},\ \mathcal{C}\begin{bmatrix}3&0\\1&0\end{bmatrix}^2$
$(7,1)$	1	$\mathcal{C}\begin{bmatrix}7&0\\1&0\end{bmatrix}$
$(5,5)$	9	$\left(\frac{\pi}{\tau_2}\right)^5 \mathrm{E}_5,\ \mathcal{C}\begin{bmatrix}1&1&3\\1&1&3\end{bmatrix},\ \mathcal{A}\begin{bmatrix}0&2&3\\3&0&2\end{bmatrix},\ \mathcal{A}\begin{bmatrix}0&1&2&2\\1&1&0&3\end{bmatrix},\ \left(\frac{\pi}{\tau_2}\right)^5 \zeta_5,\ \left(\frac{\pi}{\tau_2}\right)^5 \mathrm{E}_2\mathrm{E}_3,$ $\left(\frac{\pi}{\tau_2}\right)^5 \mathrm{E}_2\zeta_3,\ \mathcal{C}\begin{bmatrix}1&0\\3&0\end{bmatrix}\mathcal{C}\begin{bmatrix}4&0\\2&0\end{bmatrix},\ \mathcal{C}\begin{bmatrix}3&0\\1&0\end{bmatrix}\mathcal{C}\begin{bmatrix}2&0\\4&0\end{bmatrix}$
$(6,4)$	8	$\mathcal{C}\begin{bmatrix}6&0\\4&0\end{bmatrix},\ \mathcal{C}\begin{bmatrix}1&1&4\\1&1&2\end{bmatrix},\ \mathcal{C}\begin{bmatrix}1&2&3\\1&0&3\end{bmatrix},\ \mathcal{C}\begin{bmatrix}1&1&1&3\\0&1&1&2\end{bmatrix},$ $\left(\frac{\pi}{\tau_2}\right)^3 \mathrm{E}_3\mathcal{C}\begin{bmatrix}3&0\\1&0\end{bmatrix},\ \left(\frac{\pi}{\tau_2}\right)^3 \mathrm{E}_2\mathcal{C}\begin{bmatrix}4&0\\2&0\end{bmatrix},\ \mathcal{C}\begin{bmatrix}3&0\\1&0\end{bmatrix}\zeta_3,\ \mathcal{C}\begin{bmatrix}1&0\\3&0\end{bmatrix}\mathcal{C}\begin{bmatrix}5&0\\1&0\end{bmatrix}$
$(7,3)$	5	$\mathcal{C}\begin{bmatrix}7&0\\3&0\end{bmatrix},\ \mathcal{C}\begin{bmatrix}1&1&5\\1&1&1\end{bmatrix},\ \mathcal{C}\begin{bmatrix}0&2&5\\1&0&2\end{bmatrix},\ \mathcal{C}\begin{bmatrix}3&0\\1&0\end{bmatrix}\mathcal{C}\begin{bmatrix}4&0\\2&0\end{bmatrix},\ \left(\frac{\pi}{\tau_2}\right)^2 \mathrm{E}_2\mathcal{C}\begin{bmatrix}5&0\\1&0\end{bmatrix}$
$(8,2)$	3	$\mathcal{C}\begin{bmatrix}8&0\\2&0\end{bmatrix},\ \mathcal{C}\begin{bmatrix}0&3&5\\1&0&1\end{bmatrix},\ \mathcal{C}\begin{bmatrix}3&0\\1&0\end{bmatrix}\mathcal{C}\begin{bmatrix}5&0\\1&0\end{bmatrix}$
$(9,1)$	1	$\mathcal{C}\begin{bmatrix}9&0\\1&0\end{bmatrix}$
		$\left(\frac{\pi}{\tau_2}\right)^6 \mathrm{E}_6,\ \mathcal{C}\begin{bmatrix}1&1&4\\1&1&4\end{bmatrix},\ \mathcal{C}\begin{bmatrix}1&2&3\\1&2&3\end{bmatrix},\ \mathcal{C}\begin{bmatrix}2&2&2\\2&2&2\end{bmatrix},\ \mathcal{C}\begin{bmatrix}1&1&2&2\\1&1&2&2\end{bmatrix},\ \mathcal{A}\begin{bmatrix}0&2&4\\5&0&1\end{bmatrix},$ $\mathcal{A}\begin{bmatrix}0&2&2&2\\3&0&1&2\end{bmatrix},\ \mathcal{A}\begin{bmatrix}0&1&2&3\\2&1&3&0\end{bmatrix},\ \left(\frac{\pi}{\tau_2}\right)^6 \zeta_3^2,\ \left(\frac{\pi}{\tau_2}\right)^6 \mathrm{E}_3^2,\ \left(\frac{\pi}{\tau_2}\right)^6 \mathrm{E}_3\zeta_3,\ \left(\frac{\pi}{\tau_2}\right)^6 \mathrm{E}_2\mathrm{E}_4,$

Table 3. Continued.

$(6,6)$	21	$\left(\frac{\pi}{\tau_2}\right)^2 \mathrm{E}_2 \mathcal{C}\begin{bmatrix}1&1&2\\1&1&2\end{bmatrix}, \left(\frac{\pi}{\tau_2}\right)^6 \mathrm{E}_2^3, \mathcal{C}\begin{bmatrix}4&0\\2&0\end{bmatrix}\mathcal{C}\begin{bmatrix}2&0\\4&0\end{bmatrix}, \mathcal{C}\begin{bmatrix}5&0\\1&0\end{bmatrix}\mathcal{C}\begin{bmatrix}1&0\\5&0\end{bmatrix},$ $\mathcal{C}\begin{bmatrix}3&0\\1&0\end{bmatrix}\mathcal{C}\begin{bmatrix}3&0\\5&0\end{bmatrix}, \mathcal{C}\begin{bmatrix}1&0\\3&0\end{bmatrix}\mathcal{C}\begin{bmatrix}5&0\\3&0\end{bmatrix}, \mathcal{C}\begin{bmatrix}3&0\\1&0\end{bmatrix}\mathcal{C}\begin{bmatrix}1&1&1\\1&1&3\end{bmatrix},$ $\mathcal{C}\begin{bmatrix}1&0\\3&0\end{bmatrix}\mathcal{C}\begin{bmatrix}1&1&3\\1&1&1\end{bmatrix}, \left(\frac{\pi}{\tau_2}\right)^2 \mathrm{E}_2 \mathcal{C}\begin{bmatrix}3&0\\1&0\end{bmatrix}\mathcal{C}\begin{bmatrix}1&0\\3&0\end{bmatrix}$ $\mathcal{C}\begin{bmatrix}7&0\\5&0\end{bmatrix}, \mathcal{C}\begin{bmatrix}0&1&6\\1&4&0\end{bmatrix}, \mathcal{C}\begin{bmatrix}0&1&6\\2&3&0\end{bmatrix}, \mathcal{C}\begin{bmatrix}0&2&5\\2&3&0\end{bmatrix}, \mathcal{C}\begin{bmatrix}0&3&4\\4&0&1\end{bmatrix}, \mathcal{C}\begin{bmatrix}1&1&2&3\\1&1&2&1\end{bmatrix},$ $\mathcal{C}\begin{bmatrix}1&2&2&2\\1&0&2&2\end{bmatrix}, \mathcal{C}\begin{bmatrix}0&1&2&4\\2&1&2&0\end{bmatrix}, \left(\frac{\pi}{\tau_2}\right)^3 \mathrm{E}_3 \mathcal{C}\begin{bmatrix}4&0\\2&0\end{bmatrix}, \left(\frac{\pi}{\tau_2}\right)^3 \mathcal{C}\begin{bmatrix}4&0\\2&0\end{bmatrix}\zeta_3,$ $\left(\frac{\pi}{\tau_2}\right)^4 \mathrm{E}_4 \mathcal{C}\begin{bmatrix}3&0\\1&0\end{bmatrix}, \left(\frac{\pi}{\tau_2}\right)^2 \mathrm{E}_2 \mathcal{C}\begin{bmatrix}5&0\\3&0\end{bmatrix}, \mathcal{C}\begin{bmatrix}3&0\\1&0\end{bmatrix}\mathcal{C}\begin{bmatrix}1&1&2\\1&1&2\end{bmatrix},$
$(7,5)$	18	$\left(\frac{\pi}{\tau_2}\right)^2 \mathrm{E}_2 \mathcal{C}\begin{bmatrix}1&1&3\\1&1&1\end{bmatrix}, \mathcal{C}\begin{bmatrix}3&0\\1&0\end{bmatrix}^2 \mathcal{C}\begin{bmatrix}1&0\\3&0\end{bmatrix}, \mathcal{C}\begin{bmatrix}5&0\\1&0\end{bmatrix}\mathcal{C}\begin{bmatrix}2&0\\4&0\end{bmatrix},$ $\mathcal{C}\begin{bmatrix}1&0\\3&0\end{bmatrix}\mathcal{C}\begin{bmatrix}6&0\\2&0\end{bmatrix}, \left(\frac{\pi}{\tau_2}\right)^4 \mathrm{E}_2^2 \mathcal{C}\begin{bmatrix}3&0\\1&0\end{bmatrix}$ $\mathcal{C}\begin{bmatrix}8&0\\4&0\end{bmatrix}, \mathcal{C}\begin{bmatrix}0&2&6\\2&2&0\end{bmatrix}, \mathcal{C}\begin{bmatrix}0&3&5\\2&2&0\end{bmatrix}, \mathcal{C}\begin{bmatrix}0&4&4\\3&0&1\end{bmatrix}, \mathcal{C}\begin{bmatrix}1&2&2&3\\1&1&2&0\end{bmatrix}, \mathcal{C}\begin{bmatrix}1&2&2&3\\1&1&2&0\end{bmatrix}$
$(8,4)$	14	$\mathcal{C}\begin{bmatrix}4&0\\2&0\end{bmatrix}^2, \left(\frac{\pi}{\tau_2}\right)^3 \mathrm{E}_3 \mathcal{C}\begin{bmatrix}5&0\\1&0\end{bmatrix}, \left(\frac{\pi}{\tau_2}\right)^3 \mathcal{C}\begin{bmatrix}5&0\\1&0\end{bmatrix}\zeta_3, \mathcal{C}\begin{bmatrix}3&0\\1&0\end{bmatrix}\mathcal{C}\begin{bmatrix}5&0\\3&0\end{bmatrix},$ $\left(\frac{\pi}{\tau_2}\right)^2 \mathrm{E}_2 \mathcal{C}\begin{bmatrix}6&0\\2&0\end{bmatrix}, \mathcal{C}\begin{bmatrix}3&0\\1&0\end{bmatrix}\mathcal{C}\begin{bmatrix}1&1&3\\1&1&1\end{bmatrix}, \left(\frac{\pi}{\tau_2}\right)^2 \mathrm{E}_2 \mathcal{C}\begin{bmatrix}3&0\\1&1\end{bmatrix}^2, \mathcal{C}\begin{bmatrix}3&0\\1&0\end{bmatrix}\mathcal{C}\begin{bmatrix}7&0\\1&0\end{bmatrix}$
$(9,3)$	8	$\mathcal{C}\begin{bmatrix}9&0\\3&0\end{bmatrix}, \mathcal{C}\begin{bmatrix}0&3&6\\1&2&0\end{bmatrix}, \mathcal{C}\begin{bmatrix}0&3&6\\2&1&0\end{bmatrix}, \mathcal{C}\begin{bmatrix}0&4&5\\2&1&0\end{bmatrix}, \mathcal{C}\begin{bmatrix}4&0\\2&0\end{bmatrix}\mathcal{C}\begin{bmatrix}5&0\\1&0\end{bmatrix},$ $\mathcal{C}\begin{bmatrix}3&0\\1&0\end{bmatrix}\mathcal{C}\begin{bmatrix}6&0\\2&0\end{bmatrix}, \left(\frac{\pi}{\tau_2}\right)^2 \mathrm{E}_2 \mathcal{C}\begin{bmatrix}7&0\\1&0\end{bmatrix}, \mathcal{C}\begin{bmatrix}3&0\\1&0\end{bmatrix}^2$
$(10,2)$	4	$\mathcal{C}\begin{bmatrix}10&0\\2&0\end{bmatrix}, \mathcal{C}\begin{bmatrix}0&4&6\\1&1&0\end{bmatrix}, \mathcal{C}\begin{bmatrix}5&0\\1&0\end{bmatrix}^2, \mathcal{C}\begin{bmatrix}3&0\\1&0\end{bmatrix}\mathcal{C}\begin{bmatrix}7&0\\1&0\end{bmatrix}$
$(11,1)$	1	$\mathcal{C}\begin{bmatrix}11&0\\1&0\end{bmatrix}$

The remaining basis elements in these sectors are real[12]. The cusp forms (9.3a) and (9.3b) were discussed in [21], whereas (9.3c) has higher loop order than the graphs studied in the reference. In the weight $(6,6)$ sector, the dimension of the space of two-loop imaginary cusp forms was found to be 2 in [21], in agreement with (9.4).

[12] Note that if we form antisymmetric combinations $\mathcal{A}\begin{bmatrix}A\\B\end{bmatrix}$ in the (a,a) sectors with $a \leqslant 4$, these vanish since all basis elements are real.

The basis of MGFs has an intricate structure which is closely related to the counting of iterated Eisenstein integrals, but this structure is not manifest in the basis given in table 3. To make the relation to iterated Eisenstein integrals more transparent, we will use a second basis, summarized in table 4. The basis has been multiplied by τ_2^{a+k}/π^a in the $(a+k, a-k)$ sector of table 4 as compared to table 3 for ease of notation. This means in particular that the basis elements given for the $a = b$ sectors are rendered modular invariant.

The structure of the basis in table 4 is the following: in the modular invariant sectors, we split the irreducible basis elements into real and complex MGFs. The real ones are denoted by E, the complex ones by B, where the subscript refers to the holomorphic Eisenstein series appearing in the Cauchy–Riemann equations of the respective basis element. If several basis elements belong to the same sector w.r.t. these holomorphic Eisenstein series, we use a prime to distinguish them.

The non-holomorphic Eisenstein series E_k defined in (2.32) belong to the real basis elements. The remaining real basis elements of higher depth were defined in [47] to streamline their Cauchy–Riemann equations as detailed below and are given in terms of the MGFs defined previously by

$$\mathrm{E}_{2,2} = C_{1,1,2} - \frac{9}{10}\mathrm{E}_4 \tag{9.5a}$$

$$\mathrm{E}_{2,3} = C_{1,1,3} - \frac{43}{35}\mathrm{E}_5 \tag{9.5b}$$

$$\mathrm{E}_{3,3} = 3C_{1,2,3} + C_{2,2,2} - \frac{15}{14}\mathrm{E}_6 \tag{9.5c}$$

$$\mathrm{E}'_{3,3} = C_{1,2,3} + \frac{17}{60}C_{2,2,2} - \frac{59}{140}\mathrm{E}_6 \tag{9.5d}$$

$$\mathrm{E}_{2,4} = 9C_{1,1,4} + 3C_{1,2,3} + C_{2,2,2} - 13\mathrm{E}_6 \tag{9.5e}$$

$$\mathrm{E}_{2,2,2} = -C_{1,1,2,2} + \frac{232}{45}C_{2,2,2} + \frac{292}{15}C_{1,2,3} + \frac{2}{5}C_{1,1,4} + 2\mathrm{E}_3^2 + \mathrm{E}_2\mathrm{E}_4 - \frac{466}{45}\mathrm{E}_6, \tag{9.5f}$$

where $C_{a,b,c}$ and $C_{a,b,c,d}$ were defined in (2.28) and (2.29), respectively. A subscript k means in this notation that the holomorphic Eisenstein series G_{2k} appears in the Cauchy–Riemann equations, i.e. in the lowest Cauchy–Riemann derivative in which a holomorphic Eisenstein series appears. This determines the sector of iterated Eisenstein integrals that appear in the expansion of the basis element, cf the discussion in section 5 of [30]. For instance, the basis element $\mathrm{E}_{2,4}$ belongs to the $\mathrm{G}_4\mathrm{G}_8$ sector. The Cauchy–Riemann equations which make this manifest for the real irreducible basis elements are

$$\nabla_0^k \mathrm{E}_k = \frac{\tau_2^{2k}}{\pi^k}\frac{(2k-1)!}{(k-1)!}\mathrm{G}_{2k} \tag{9.6a}$$

$$\nabla_0^3 \mathrm{E}_{2,2} = -6\frac{\tau_2^4}{\pi^2}\mathrm{G}_4\nabla_0\mathrm{E}_2 \tag{9.6b}$$

$$\nabla_0^3 \mathrm{E}_{2,3} = -2\nabla_0\mathrm{E}_2\nabla_0^2\mathrm{E}_3 - 4\frac{\tau_2^4}{\pi^2}\mathrm{G}_4\nabla_0\mathrm{E}_3 \tag{9.6c}$$

$$\nabla_0^5 \mathrm{E}_{3,3} = 180\frac{\tau_2^6}{\pi^3}\mathrm{G}_6\nabla_0^2\mathrm{E}_3 \tag{9.6d}$$

$$\nabla_0^4 \mathrm{E}'_{3,3} = -12\frac{\tau_2^6}{\pi^3}\mathrm{G}_6\nabla_0\mathrm{E}_3 \tag{9.6e}$$

$$\nabla_0^3 \mathrm{E}_{2,4} = -\frac{27}{2}\nabla_0\left(\mathrm{E}_2\nabla_0^2\mathrm{E}_4\right) - \frac{27}{4}\nabla_0^3\mathrm{B}_{2,4} - \frac{21}{40}\nabla_0^3\mathrm{B}'_{2,4} - 27\frac{\tau_2^4}{\pi^2}\mathrm{G}_4\nabla_0\mathrm{E}_4 \tag{9.6f}$$

$$\nabla_0^3 \mathrm{E}_{2,2,2} = (\nabla_0\mathrm{E}_2)^3 - 12\frac{\tau_2^4}{\pi^2}\mathrm{G}_4\nabla_0\mathrm{E}_{2,2}, \tag{9.6g}$$

where we use the Cauchy–Riemann operator defined in (5.44) and the complex basis elements $\mathrm{B}_{2,4}$ and $\mathrm{B}'_{2,4}$ are defined in (9.9). The right-hand sides in (9.6) all lie manifestly in the same sector of holomorphic Eisenstein series as indicated by the subscripts on the left-hand side. In [47], the real irreducible basis elements E were written in terms of iterated Eisenstein integrals. From this, we can read off their Laurent polynomials [3, 11], namely

$$\mathrm{E}_k\big|_{q^0\bar{q}^0} = (-1)^{k-1}\frac{B_{2k}}{(2k)!}(4y)^k + 4\binom{2k-3}{k-1}\zeta_{2k-1}(4y)^{1-k} \tag{9.7a}$$

$$\mathrm{E}_{2,2}\big|_{q^0\bar{q}^0} = -\frac{y^4}{20\,250} + \frac{y\zeta_3}{45} + \frac{5\zeta_5}{12y} - \frac{\zeta_3^2}{4y^2} \tag{9.7b}$$

$$\mathrm{E}_{2,3}\big|_{q^0\bar{q}^0} = -\frac{4y^5}{297\,675} + \frac{2y^2\zeta_3}{945} - \frac{\zeta_5}{180} + \frac{7\zeta_7}{16y^2} - \frac{\zeta_3\zeta_5}{2y^3} \tag{9.7c}$$

$$\mathrm{E}_{3,3}\big|_{q^0\bar{q}^0} = \frac{2y^6}{6251\,175} + \frac{y\zeta_5}{210} + \frac{\zeta_7}{16y} - \frac{7\zeta_9}{64y^3} + \frac{9\zeta_5^2}{64y^4} \tag{9.7d}$$

$$\mathrm{E}'_{3,3}\big|_{q^0\bar{q}^0} = -\frac{y^6}{18\,753\,525} + \frac{y\zeta_5}{630} + \frac{3\zeta_7}{160y} - \frac{7\zeta_9}{480y^3} \tag{9.7e}$$

$$\mathrm{E}_{2,4}\big|_{q^0\bar{q}^0} = -\frac{y^6}{70\,875} + \frac{y^3\zeta_3}{525} + \frac{3\zeta_7}{40y} + \frac{25\zeta_9}{8y^3} - \frac{135\zeta_3\zeta_7}{32y^4} \tag{9.7f}$$

$$\mathrm{E}_{2,2,2}\big|_{q^0\bar{q}^0} = \frac{4y^6}{9568\,125} - \frac{2y^3\zeta_3}{10\,125} + \frac{y\zeta_5}{54} + \frac{\zeta_3^2}{90} + \frac{661\zeta_7}{1800y} - \frac{5\zeta_3\zeta_5}{12y^2} + \frac{\zeta_3^3}{6y^3}, \tag{9.7g}$$

where $y = \pi\tau_2$, and the Laurent polynomial of E_k can be read off from its well-known q-expansion, given e.g. in [48].

The cusp forms listed in (9.3) and (9.4) were all of the form $\mathcal{C}_\Gamma - \overline{\mathcal{C}_\Gamma}$ and hence purely imaginary. Using the Laurent polynomials (9.7), and the basis elements given in table 4, it is easy to show that there are no real cusp forms in the space of MGFs at weight (a, a) with $a \leqslant 5$ and that there are five real cusp forms at weight $(6, 6)$. A basis in this space of real cusp forms is given by

$$\begin{aligned}\mathrm{S}_1 = {}& \frac{8}{15}\mathrm{E}_{3,3} - 4\mathrm{E}'_{3,3} - \frac{1}{3}\mathrm{E}_2^3 + \mathrm{E}_4\mathrm{E}_2 + \frac{349}{875}\mathrm{E}_3^2 + \frac{2}{45}\zeta_3^2 + \frac{1}{3}\tau_2^{-2}\mathrm{E}_2\overline{\nabla}_0\mathrm{E}_2\nabla_0\mathrm{E}_2 \\ & - \frac{233}{1750}\tau_2^{-2}\overline{\nabla}_0\mathrm{E}_3\nabla_0\mathrm{E}_3 + \frac{1}{10\,500}\tau_2^{-4}\overline{\nabla}_0^2\mathrm{E}_3\nabla_0^2\mathrm{E}_3 - \frac{1}{6}\tau_2^{-2}\left(\nabla_0\mathrm{E}_2\overline{\nabla}_0\mathrm{E}_4 + \overline{\nabla}_0\mathrm{E}_2\nabla_0\mathrm{E}_4\right)\end{aligned} \tag{9.8a}$$

$$\begin{aligned}\mathrm{S}_2 = {}& \mathrm{E}_{2,4} + \frac{8748}{175}\mathrm{E}_{3,3} - \frac{5622}{35}\mathrm{E}'_{3,3} - \frac{269}{50}\mathrm{E}_3^2 - \frac{3739}{2100}\tau_2^{-2}\overline{\nabla}_0\mathrm{E}_3\nabla_0\mathrm{E}_3 \\ & + \frac{1}{840}\tau_2^{-4}\overline{\nabla}_0^2\mathrm{E}_3\nabla_0^2\mathrm{E}_3 + \frac{9}{8}\tau_2^{-2}\left(\nabla_0\mathrm{E}_2\overline{\nabla}_0\mathrm{E}_4 + \overline{\nabla}_0\mathrm{E}_2\nabla_0\mathrm{E}_4\right)\end{aligned} \tag{9.8b}$$

Table 4. Basis of (convergent) MGFs of weight $a+b \leqslant 12$, excluding holomorphic Eisenstein series. The prefactors of τ_2 were chosen such that the modular weight in the sector $(a+k, a-k)$ is $(0,-2k)$ for $0 \leqslant k < a$. The counting includes zeta values.

Weight	# basis elements	Basis elements
$(2,2)$	1	E_2
$(3,1)$	1	$\nabla_0\mathrm{E}_2$
$(3,3)$	2	E_3, ζ_3
$(4,2)$	1	$\nabla_0\mathrm{E}_3$
$(5,1)$	1	$\nabla_0^2\mathrm{E}_3$
$(4,4)$	4	$\mathrm{E}_4,\ \mathrm{E}_{2,2},\ \mathrm{E}_2^2,\ \tau_2^{-2}\nabla_0\mathrm{E}_2\overline{\nabla}_0\mathrm{E}_2$
$(5,3)$	3	$\nabla_0\mathrm{E}_4, \nabla_0\mathrm{E}_{2,2}, \mathrm{E}_2\nabla_0\mathrm{E}_2$
$(6,2)$	2	$\nabla_0^2\mathrm{E}_4,\ (\nabla_0\mathrm{E}_2)^2$
$(7,1)$	1	$\nabla_0^3\mathrm{E}_4$
$(5,5)$	9	$\mathrm{E}_5, \mathrm{E}_{2,3}, \mathrm{B}_{2,3}, \mathrm{B}'_{2,3}, \zeta_5,$ $\mathrm{E}_2\mathrm{E}_3,\ \mathrm{E}_2\zeta_3,\ \tau_2^{-2}\overline{\nabla}_0\mathrm{E}_2\nabla_0\mathrm{E}_3,\ \tau_2^{-2}\nabla_0\mathrm{E}_2\overline{\nabla}_0\mathrm{E}_3$ $\nabla_0\mathrm{E}_5, \nabla_0\mathrm{E}_{2,3}, \nabla_0\mathrm{B}_{2,3}, \nabla_0\mathrm{B}'_{2,3},$
$(6,4)$	8	$\nabla_0\mathrm{E}_2\mathrm{E}_3,\ \mathrm{E}_2\nabla_0\mathrm{E}_3,\ \nabla_0\mathrm{E}_2\zeta_3,\ \tau_2^{-2}\overline{\nabla}_0\mathrm{E}_2\nabla_0^2\mathrm{E}_3$
$(7,3)$	5	$\nabla_0^2\mathrm{E}_5,\ \nabla_0^2\mathrm{E}_{2,3},\ \nabla_0^2\mathrm{B}'_{2,3},\ \nabla_0\mathrm{E}_2\nabla_0\mathrm{E}_3,\ \mathrm{E}_2\nabla_0^2\mathrm{E}_3$
$(8,2)$	3	$\nabla_0^3\mathrm{E}_5,\ \nabla_0^3\mathrm{B}'_{2,3},\ \nabla_0\mathrm{E}_2\nabla_0^2\mathrm{E}_3$
$(9,1)$	1	$\nabla_0^4\mathrm{E}_5$
$(6,6)$	21	$\mathrm{E}_6,\ \mathrm{E}_{2,4},\ \mathrm{E}_{3,3},\ \mathrm{E}'_{3,3},\ \mathrm{E}_{2,2,2},\ \mathrm{B}_{2,4},\ \mathrm{B}'_{2,4},\ \mathrm{B}_{2,2,2},\ \zeta_3^2,$ $\mathrm{E}_3^2,\ \mathrm{E}_3\zeta_3,\ \mathrm{E}_2\mathrm{E}_4,\ \mathrm{E}_2\mathrm{E}_{2,2},\ \mathrm{E}_2^3,\ \tau_2^{-2}\nabla_0\mathrm{E}_3\overline{\nabla}_0\mathrm{E}_3,\ \tau_2^{-4}\nabla_0^2\mathrm{E}_3\overline{\nabla}_0{}^2\mathrm{E}_3$ $\tau_2^{-2}\nabla_0\mathrm{E}_2\overline{\nabla}_0\mathrm{E}_4,\ \tau_2^{-2}\overline{\nabla}_0\mathrm{E}_2\nabla_0\mathrm{E}_4, \tau_2^{-2}\nabla_0\mathrm{E}_2\overline{\nabla}_0\mathrm{E}_{2,2},\ \tau_2^{-2}\overline{\nabla}_0\mathrm{E}_2\nabla_0\mathrm{E}_{2,2},$ $\tau_2^{-2}\mathrm{E}_2\nabla_0\mathrm{E}_2\overline{\nabla}_0\mathrm{E}_2$
$(7,5)$	18	$\nabla_0\mathrm{E}_6, \nabla_0\mathrm{E}_{2,4}, \nabla_0\mathrm{E}_{3,3}, \nabla_0\mathrm{E}'_{3,3}, \nabla_0\mathrm{E}_{2,2,2}, \nabla_0\mathrm{B}_{2,4}, \nabla_0\mathrm{B}'_{2,4}, \nabla_0\mathrm{B}_{2,2,2},$ $\mathrm{E}_3\nabla_0\mathrm{E}_3,\ \nabla_0\mathrm{E}_3\zeta_3,\ \nabla_0\mathrm{E}_2\mathrm{E}_4,\ \mathrm{E}_2\nabla_0\mathrm{E}_4,\ \nabla_0\mathrm{E}_2\mathrm{E}_{2,2},\ \mathrm{E}_2\nabla_0\mathrm{E}_{2,2},\ \mathrm{E}_2^2\nabla_0\mathrm{E}_2,$ $\tau_2^{-2}\nabla_0^2\mathrm{E}_3\overline{\nabla}_0\mathrm{E}_3,\ \tau_2^{-2}\overline{\nabla}_0\mathrm{E}_2\nabla_0^2\mathrm{E}_4,\ \tau_2^{-2}(\nabla_0\mathrm{E}_2)^2\overline{\nabla}_0\mathrm{E}_2$
$(8,4)$	14	$\nabla_0^2\mathrm{E}_6,\ \nabla_0^2\mathrm{E}_{2,4},\ \nabla_0^2\mathrm{E}'_{3,3},\ \nabla_0^2\mathrm{B}_{2,4},\ \nabla_0^2\mathrm{B}'_{2,4},\ \nabla_0^2\mathrm{B}_{2,2,2}$ $(\nabla_0\mathrm{E}_3)^2,\ \mathrm{E}_3\nabla_0^2\mathrm{E}_3,\ \nabla_0^2\mathrm{E}_3\zeta_3,\ \nabla_0\mathrm{E}_2\nabla_0\mathrm{E}_4,\ \mathrm{E}_2\nabla_0^2\mathrm{E}_4,\ \nabla_0\mathrm{E}_2\nabla_0\mathrm{E}_{2,2},$ $\mathrm{E}_2(\nabla_0\mathrm{E}_2)^2,\ \tau_2^{-2}\overline{\nabla}_0\mathrm{E}_2\nabla_0^3\mathrm{E}_4$
$(9,3)$	8	$\nabla_0^3\mathrm{E}_6,\ \nabla_0^3\mathrm{E}'_{3,3},\ \nabla_0^3\mathrm{B}_{2,4},\ \nabla_0^3\mathrm{B}'_{2,4},\ \nabla_0\mathrm{E}_3\nabla_0^2\mathrm{E}_3,\ \nabla_0\mathrm{E}_2\nabla_0^2\mathrm{E}_4,$ $\mathrm{E}_2\nabla_0^3\mathrm{E}_4,\ (\nabla_0\mathrm{E}_2)^3$
$(10,2)$	4	$\nabla_0^4\mathrm{E}_6,\ \nabla_0^4\mathrm{B}'_{2,4},\ (\nabla_0^2\mathrm{E}_3)^2,\ \nabla_0\mathrm{E}_2\nabla_0^3\mathrm{E}_4$
$(11,1)$	1	$\nabla_0^5\mathrm{E}_6$

$$\mathrm{S}_3 = \mathrm{E}_{2,2,2} + \frac{5288}{1125}\mathrm{E}_{3,3} - \frac{2644}{75}\mathrm{E}'_{3,3} + \mathrm{E}_2\mathrm{E}_{2,2} - \frac{1}{6}\mathrm{E}_2^3 + \frac{401}{17\,500}\mathrm{E}_3^2 + \frac{1}{4}\tau_2^{-2}\mathrm{E}_2\overline{\nabla}_0\mathrm{E}_2\nabla_0\mathrm{E}_2 - \frac{11\,801}{39\,375}\tau_2^{-2}\overline{\nabla}_0\mathrm{E}_3\nabla_0\mathrm{E}_3 + \frac{127}{630\,000}\tau_2^{-4}\overline{\nabla}_0^2\mathrm{E}_3\nabla_0^2\mathrm{E}_3 \tag{9.8c}$$

$$\mathrm{S}_4 = -2\mathrm{E}_2\mathrm{E}_{2,2} - \mathrm{E}_2^3 + \frac{8757}{1250}\mathrm{E}_3^2 + \frac{1}{5}\zeta_3^2 + \frac{3}{2}\tau_2^{-2}\mathrm{E}_2\overline{\nabla}_0\mathrm{E}_2\nabla_0\mathrm{E}_2 + \tau_2^{-2}\left(\nabla_0\mathrm{E}_2\overline{\nabla}_0\mathrm{E}_{2,2} + \overline{\nabla}_0\mathrm{E}_2\nabla_0\mathrm{E}_{2,2}\right) - \frac{3283}{1875}\tau_2^{-2}\overline{\nabla}_0\mathrm{E}_3\nabla_0\mathrm{E}_3 - \frac{7}{15\,000}\tau_2^{-4}\overline{\nabla}_0^2\mathrm{E}_3\nabla_0^2\mathrm{E}_3 \tag{9.8d}$$

$$\mathrm{S}_5 = -\frac{9}{5}\mathrm{E}_2\mathrm{E}_{2,2} - \frac{311}{350}\mathrm{E}_2^3 + \frac{26\,187}{12\,500}\mathrm{E}_3^2 + \mathrm{E}_3\zeta_3 + \frac{311}{2625}\zeta_3^2 + \frac{307}{700}\tau_2^{-2}\mathrm{E}_2\overline{\nabla}_0\mathrm{E}_2\nabla_0\mathrm{E}_2$$

$$-\frac{1638}{3125}\tau_2^{-2}\overline{\nabla}_0\mathrm{E}_3\nabla_0\mathrm{E}_3+\frac{21}{50\,000}\tau_2^{-4}\overline{\nabla}_0^2\mathrm{E}_3\nabla_0^2\mathrm{E}_3. \tag{9.8e}$$

The complex irreducible basis elements follow the same notation regarding the sectors of holomorphic Eisenstein series. They are defined in terms of lattice sums by

$$\mathrm{B}_{2,3}=\left(\frac{\tau_2}{\pi}\right)^5\left(\mathcal{A}\begin{bmatrix}0&1&2&2\\1&1&0&3\end{bmatrix}+\mathcal{C}\begin{bmatrix}3&0\\1&0\end{bmatrix}\mathcal{C}\begin{bmatrix}2&0\\4&0\end{bmatrix}-\mathcal{C}\begin{bmatrix}1&0\\3&0\end{bmatrix}\mathcal{C}\begin{bmatrix}4&0\\2&0\end{bmatrix}\right) \tag{9.9a}$$

$$\begin{aligned}\mathrm{B}'_{2,3}=&\left(\frac{\tau_2}{\pi}\right)^5\left(\frac{1}{2}\mathcal{A}\begin{bmatrix}0&2&3\\3&0&2\end{bmatrix}+\mathcal{A}\begin{bmatrix}0&1&2&2\\1&1&0&3\end{bmatrix}+\mathcal{C}\begin{bmatrix}3&0\\1&0\end{bmatrix}\mathcal{C}\begin{bmatrix}2&0\\4&0\end{bmatrix}\right.\\&\left.-\mathcal{C}\begin{bmatrix}1&0\\3&0\end{bmatrix}\mathcal{C}\begin{bmatrix}4&0\\2&0\end{bmatrix}\right)+\frac{129}{20}\mathrm{E}_5-\frac{1}{2}\mathrm{E}_2\zeta_3-\frac{21}{4}C_{1,1,3}\end{aligned} \tag{9.9b}$$

$$\begin{aligned}\mathrm{B}_{2,4}=&\left(\frac{\tau_2}{\pi}\right)^6\left(\mathcal{A}\begin{bmatrix}0&2&4\\5&0&1\end{bmatrix}+2\left(\mathcal{C}\begin{bmatrix}3&0\\1&0\end{bmatrix}\mathcal{C}\begin{bmatrix}3&0\\5&0\end{bmatrix}-\mathcal{C}\begin{bmatrix}1&0\\3&0\end{bmatrix}\mathcal{C}\begin{bmatrix}5&0\\3&0\end{bmatrix}\right)\right)\\&+C_{1,1,4}+\frac{1}{3}C_{1,2,3}+\frac{1}{9}C_{2,2,2}-\mathrm{E}_2\mathrm{E}_4-\frac{13}{9}\mathrm{E}_6\end{aligned} \tag{9.9c}$$

$$\mathrm{B}'_{2,4}=\left(\frac{\tau_2}{\pi}\right)^6\mathcal{A}\begin{bmatrix}0&2&2&2\\3&0&1&2\end{bmatrix}-30C_{1,1,4}-10C_{1,2,3}-\frac{10}{3}C_{2,2,2}-3\mathrm{E}_3\zeta_3+\frac{130}{3}\mathrm{E}_6 \tag{9.9d}$$

$$\begin{aligned}\mathrm{B}_{2,2,2}=&\left(\frac{\tau_2}{\pi}\right)^6\left(4\mathcal{A}\begin{bmatrix}0&1&2&3\\2&1&3&0\end{bmatrix}+\frac{121}{50}\mathcal{A}\begin{bmatrix}0&2&2&2\\3&0&1&2\end{bmatrix}-\frac{113}{5}\mathcal{A}\begin{bmatrix}0&2&4\\5&0&1\end{bmatrix}\right.\\&+\frac{266}{5}\left(\mathcal{C}\begin{bmatrix}1&0\\3&0\end{bmatrix}\mathcal{C}\begin{bmatrix}5&0\\3&0\end{bmatrix}-\mathcal{C}\begin{bmatrix}3&0\\1&0\end{bmatrix}\mathcal{C}\begin{bmatrix}3&0\\5&0\end{bmatrix}\right)\\&\left.+4\left(\mathcal{C}\begin{bmatrix}3&0\\1&0\end{bmatrix}\mathcal{C}\begin{bmatrix}1&1&1\\1&1&3\end{bmatrix}-\mathcal{C}\begin{bmatrix}1&0\\3&0\end{bmatrix}\mathcal{C}\begin{bmatrix}1&1&3\\1&1&1\end{bmatrix}\right)\right)\\&+6C_{1,1,2}\mathrm{E}_2-\frac{27}{5}\mathrm{E}_2\mathrm{E}_4-\frac{63}{50}\mathrm{E}_3\zeta_3,\end{aligned} \tag{9.9e}$$

where the real modular graph functions $C_{a,b,c}$ are defined in (2.28). The complex basis elements $\mathrm{B}_{2,3}$ and $\mathrm{B}'_{2,3}$ of the $a+b=10$ sector were first mentioned in [30]. Only the first of the basis elements in (9.9) is purely imaginary, the others contain imaginary and real contributions. The complex conjugates of the basis MGFs in (9.9) are

$$\overline{\mathrm{B}_{2,3}}=-\mathrm{B}_{2,3} \tag{9.10a}$$

$$\overline{\mathrm{B}'_{2,3}}=-\mathrm{B}'_{2,3}-\mathrm{E}_2\zeta_3-\frac{21}{2}\mathrm{E}_{2,3} \tag{9.10b}$$

$$\overline{\mathrm{B}_{2,4}}=-\mathrm{B}_{2,4}-2\mathrm{E}_2\mathrm{E}_4+\frac{2}{9}\mathrm{E}_{2,4} \tag{9.10c}$$

$$\overline{\mathrm{B}'_{2,4}}=-\mathrm{B}'_{2,4}-6\mathrm{E}_3\zeta_3-\frac{20}{3}\mathrm{E}_{2,4} \tag{9.10d}$$

$$\overline{\mathrm{B}_{2,2,2}}=-\mathrm{B}_{2,2,2}-\frac{63}{25}\mathrm{E}_3\zeta_3+12\mathrm{E}_2\mathrm{E}_{2,2}. \tag{9.10e}$$

The definition of the basis elements E and B was guided by the maxim to delay the appearance of holomorphic Eisenstein series in the Cauchy–Riemann equations to higher derivatives and to separate the different sectors of holomorphic Eisenstein series at the same time. Although this

does not fix the basis elements uniquely, the remaining freedom allows one only to isolate one purely imaginary basis element, $\mathrm{B}_{2,3}$. Similarly to (9.9), the first Cauchy–Riemann derivatives of the complex basis elements in which holomorphic Eisenstein series appear, are

$$\nabla_0^2\mathrm{B}_{2,3} = \frac{2}{7}\nabla_0^2\mathrm{B}'_{2,3} + \frac{3}{2}\left(\nabla_0\mathrm{E}_2\nabla_0\mathrm{E}_3 - \mathrm{E}_2\nabla_0^2\mathrm{E}_3 + \nabla_0^2\mathrm{E}_{2,3}\right) + \frac{\tau_2^4}{\pi^2}\mathrm{G}_4\left(9\mathrm{E}_3 + 3\zeta_3\right) \tag{9.11a}$$

$$\nabla_0^4\mathrm{B}'_{2,3} = 1260\frac{\tau_2^6}{\pi^3}\mathrm{G}_6\nabla_0\mathrm{E}_2 \tag{9.11b}$$

$$\nabla_0^4\mathrm{B}_{2,4} = -\frac{7}{90}\nabla_0^4\mathrm{B}'_{2,4} - 1680\frac{\tau_2^8}{\pi^4}\mathrm{G}_8\mathrm{E}_2 \tag{9.11c}$$

$$\nabla_0^5\mathrm{B}'_{2,4} = 151\,200\frac{\tau_2^8}{\pi^4}\mathrm{G}_8\nabla_0\mathrm{E}_2 \tag{9.11d}$$

$$\nabla_0^3\mathrm{B}_{2,2,2} = -9(\nabla_0\mathrm{E}_2)^3 - \frac{\tau_2^4}{\pi^2}\mathrm{G}_4\left(72\mathrm{E}_2\nabla_0\mathrm{E}_2 + 36\nabla_0\mathrm{E}_{2,2}\right). \tag{9.11e}$$

Since the complex basis elements are given in (9.9) in terms of real basis elements, for which the Laurent polynomials are listed in (9.7), and cusp forms with vanishing Laurent polynomials, we can assemble the Laurent polynomials of the B as well. They are given by

$$\mathrm{B}_{2,3}\big|_{q^0\bar{q}^0} = 0 \tag{9.12a}$$

$$\mathrm{B}'_{2,3}\big|_{q^0\bar{q}^0} = \frac{y^5}{14\,175} - \frac{y^2\zeta_3}{45} + \frac{7\zeta_5}{240} - \frac{\zeta_3^2}{2y} - \frac{147\zeta_7}{64y^2} + \frac{21\zeta_3\zeta_5}{8y^3} \tag{9.12b}$$

$$\mathrm{B}_{2,4}\big|_{q^0\bar{q}^0} = -\frac{4y^6}{637\,875} - \frac{\zeta_7}{180y} + \frac{25\zeta_9}{72y^3} - \frac{35\zeta_3\zeta_7}{32y^4} \tag{9.12c}$$

$$\mathrm{B}'_{2,4}\big|_{q^0\bar{q}^0} = \frac{2y^6}{42\,525} - \frac{4y^3\zeta_3}{315} - \frac{\zeta_7}{4y} - \frac{9\zeta_3\zeta_5}{4y^2} - \frac{125\zeta_9}{12y^3} + \frac{225\zeta_3\zeta_7}{16y^4} \tag{9.12d}$$

$$\mathrm{B}_{2,2,2}\big|_{q^0\bar{q}^0} = -\frac{y^6}{151\,875} + \frac{y\zeta_5}{18} + \frac{\zeta_3^2}{10} + \frac{311\zeta_3\zeta_5}{200y^2} - \frac{3\zeta_3^3}{2y^3}. \tag{9.12e}$$

The basis elements E and B span the irreducible sectors of the modular invariant subspaces of MGFs. For the subspaces with modular weight (a, b) with $a > b$, we take the Cauchy–Riemann derivatives of the E and B as irreducible basis elements. Since the space of MGFs of weight $(a + k, a - k)$ shrinks with growing k, there are relations between the Cauchy–Riemann derivatives of the E and B, leading to dropouts in this pattern. In general, these dropouts are manifest in the Cauchy–Riemann equations (9.6) and (9.11), however some of the real basis elements satisfy relations at derivatives lower than the one in which the first holomorphic Eisenstein series appear as stated in (9.6). These additional relations are

$$\nabla_0^2\mathrm{E}_{2,2} = -\frac{1}{2}(\nabla_0\mathrm{E}_2)^2 \tag{9.13a}$$

$$\nabla_0^2\mathrm{E}_{3,3} = \frac{3}{4}(\nabla_0\mathrm{E}_3)^2 + \frac{15}{2}\nabla_0^2\mathrm{E}'_{3,3} \tag{9.13b}$$

$$\nabla_0^2\mathrm{E}_{2,2,2} = -2\nabla_0\mathrm{E}_2\nabla_0\mathrm{E}_{2,2}. \tag{9.13c}$$

For the complex basis elements, there are no relations at lower derivatives than in (9.11).

On top of the irreducible basis elements E and B, there are reducible basis elements which are products of irreducible basis elements of lower weights. We also take derivatives of these reducible basis elements to generate the bases of weight (a,b) with $a > b$. Again, this is constrained by the relations (9.6), (9.11) and (9.13). As for the irreducible basis elements, the Cauchy–Riemann derivatives of the reducible basis elements also contain terms with holomorphic Eisenstein series, which are not written in the basis. Furthermore, the derivative of terms of the form $\overline{\nabla}_0^n \mathrm{E}_k$ is (up to prefactors) $\overline{\nabla}_0^{n-1}\mathrm{E}_k$. The derivative of the only depth-two instance $\overline{\nabla}_0 \mathrm{E}_{2,2}$ gives rise to $2\mathrm{E}_{2,2} - \mathrm{E}_2^2$, as follows from the Laplace equation $(\Delta - 2)\mathrm{E}_{2,2} = -\mathrm{E}_2^2$ [3].

Since the action of the derivative operators ∇_0 and $\overline{\nabla}_0$ on y is straightforwardly given by

$$\nabla_0 y = \overline{\nabla}_0 y = \frac{y^2}{\pi}, \tag{9.14}$$

using the decompositions into the basis of table 4 and the known Laurent polynomials (9.7) and (9.12), we can easily assemble the Laurent polynomials of all dihedral and trihedral MGFs of total weight $a + b \leqslant 12$. These computations are made straightforward in the `ModularGraphForms` package as outlined in the following.

Computations in the `ModularGraphForms` package are performed in the basis listed in table 3. Using the function `CConvertToNablaE`, an expression can be converted into the basis given in table 4. The real basis elements are represented by e.g. `e[2, 2]`, and `ep[3, 3]` for the primed version. The complex basis elements are given by e.g. `b[2, 3]` and `bp[2, 3]`. The Cauchy–Riemann derivatives are denoted by the functions `nablaE`, `nablaEp`, `nablaB` and `nablaBp`. Their complex conjugates are `nablaBarE`, `nablaBarEp`, `nablaBarBBar` and `nablaBarBpBar`. The first arguments of these functions is always the order of the derivative, the second is a list with the subscripts of the basis element, e.g. $\overline{\nabla}_0^2 \overline{\mathrm{B}}_{2,4}$ is denoted by `nablaBarBBar[2,{2, 4}]`. These basis elements are translated back into the basis given in table 3 by the function `CConvertFromNablaE`. Note that only the derivatives appearing in table 4 can be converted in this way. As an example, the decomposition of the graph $\mathcal{C}\begin{bmatrix}1 & 2 & 4\\ 2 & 2 & 1\end{bmatrix}$ can be performed by

```
In[48]:= CConvertToNablaE[CSimplify[c[{{1, 2, 4}, {2, 2, 1}}]]]
```

$$\text{Out[48]}= \frac{3\,\pi^6\,\nabla \mathrm{E}_6}{28\,\tau_2^7} - \frac{5\,\pi^6\,\nabla \mathrm{E}_{3,3}}{9\,\tau_2^7} + \frac{5\,\pi^6\,\nabla \mathrm{E}'_{3,3}}{3\,\tau_2^7} \ .$$

The derivative operator ∇_0 is not implemented directly, but since it is given by $\nabla_0 = \tau_2 \nabla^{(0)}$ (cf (5.44)), it can be obtained by acting with `tau[2]CHolCR` on an MGF with vanishing modular weight. E.g. the Cauchy–Riemann equation (9.11c) is reproduced by

```
In[49]:= CConvertToNablaE[Nest[CSimplify[tau[2]CHolCR[#]]&, b[2,4], 4]]
```

$$\text{Out[49]}= -\frac{7}{90}\,\nabla^4 \mathrm{B}'_{2,4} - \frac{1680\,\mathrm{E}_2\,\mathrm{G}_8\,\tau_2^8}{\pi^4} \ .$$

The Laurent polynomials (9.7) and (9.12) are implemented in the function **`CLaurentPoly`**, which replaces each of the basis elements by its Laurent polynomial and performs the necessary Cauchy–Riemann derivatives. E.g. the Laurent polynomial of the

graph $\mathcal{C}\begin{bmatrix}1 & 2 & 4\\ 2 & 2 & 1\end{bmatrix}$ decomposed in `Out[48]` can be obtained via

```
In[50]:= CLaurentPoly[Out[48]]
```

$$\text{Out[50]}= -\frac{19\,\pi^{12}}{91216125}+\frac{5\,\pi^{12}\,\zeta_5^2}{16\,y^{10}}+\frac{\pi^{12}\,\zeta_7}{288\,y^7}-\frac{7\,\pi^{12}\,\zeta_9}{64\,y^9}-\frac{135\,\pi^{12}\,\zeta_{11}}{512\,y^{11}}\ .$$

The basis elements at a certain weight are accessible via the function **`CBasis`**. If the option **`basis`** is set to the string `"C"` (the default value), the basis from table 3 is returned, if it is set to the string `"nablaE"`, the basis from table 4 is returned, e.g.

```
In[51]:= CBasis[3, 5]

         CBasis[3, 5, basis→"nablaE"]
```

$$\text{Out[51]}= \left\{C\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 3\end{bmatrix}, C\begin{bmatrix}3 & 0\\ 5 & 0\end{bmatrix}, \frac{\pi^2\,C\begin{bmatrix}1 & 0\\ 3 & 0\end{bmatrix}E_2}{\tau_2^2}\right\}$$

$$\text{Out[52]}= \left\{\overline{\nabla}E_{2,2}, \overline{\nabla}E_4, E_2\,\overline{\nabla}E_2\right\}\ .$$

Together with the function **`zIntegrate`** described in section 3.2, the basis decompositions available in the **`ModularGraphForms`** package are sufficient to expand all two- and three-point Koba–Nielsen integrals to the orders which give rise to MGFs of total modular weight at most 12. This was crucial for checking and solving the differential equation of the generating function of Koba–Nielsen integrals in [29, 30]. The arXiv submission of this paper includes the expansion of the two- and three point versions of the generating function $Y^{\tau}_{\vec{\eta}}$ defined in [30] up to order 12. For the three-point version, it also contains the Laurent polynomial of the generating series. At two-point, it was checked that the Laurent polynomials obtained using the basis decompositions agree with the closed formula given in [30] from genus-zero integrals.

10. Conclusion and outlook

In this paper, we systematically studied relations between MGFs, a class of non-holomorphic modular forms used in the computation of the low-energy expansion of closed-string genus-one amplitudes in type-II, heterotic or bosonic theories.

We studied MGFs with two, three and four vertices and introduced in particular a concise notation for four-point graphs and studied their symmetry properties systematically. For these graphs, we reviewed how topological simplifications, momentum-conservation at the vertices, factorization of $\begin{bmatrix}0\\ 0\end{bmatrix}$-edges and Cauchy–Riemann derivatives lead to relations between MGFs and discussed how these can also be understood in the integral representation of MGFs. This point of view led us to a new formulation of HSR which can be understood as integrated Fay identities of Kronecker–Eisenstein series. This formulation yields an efficient iterative procedure for higher-point HSR, circumventing difficulties in earlier approaches.

Since divergent MGFs appear naturally in the expansion of Koba–Nielsen integrals and in momentum-conservation identities, we initiated a systematic study of these divergent sums, starting with an analysis of the superficial degree of divergence for MGFs with up to four

points. We discussed HSR in divergent MGFs and Cauchy–Riemann derivatives as well as momentum conservation and factorization of divergent graphs.

By constructing all momentum conservation- and Fay identities at the corresponding weight and applying the techniques described above, we could find basis decompositions for all (convergent) two- and three-point MGFs of total modular weight $a + b \leqslant 12$. The only additional input in this process were the two well-known identities for D_3 and D_5, which are also the source of the zeta values in the basis decompositions.

We then discussed a particular basis for MGFs systematically built out of real and complex basis elements and their derivatives. Since the Laurent polynomials of these basis elements are known from the literature, we can compute the Laurent polynomials of all decomposed MGFs. This allowed us to identify five linearly independent real cusp forms at weight $(6, 6)$ and to show that no real cusp forms exist in the space of MGFs at lower weights.

The basis decompositions, as well as implementations of the manipulations discussed above, are made available in the ancillary files of the arXiv submission of this paper in the form of the **`Mathematica`** package **`ModularGraphForms`** together with two text files containing the decompositions. Using this package, we decomposed the generating function for Koba–Nielsen integrals introduced in [30] at two- and three points up to order 12. The resulting expansion is also included in the arXiv submissions.

Interestingly, the basis of MGFs obtained in this work only contains dihedral graphs. From an argument involving iterated Eisenstein integrals given in [30], we know that the basis is nevertheless complete and hence also all higher-point graphs beyond the trihedral ones considered here can be decomposed into only dihedral graphs at weight $a + b \leqslant 12$. It would be interesting to see at which weight more complicated topologies have to be included. A first step in this direction would be an extension of the **`ModularGraphForms`** package to a complete treatment of four-point graphs, which would not only allow one to explicitly find decompositions of four-point graphs, but presumably also to perform the basis decompositions at higher weights.

Acknowledgments

I would like to thank Axel Kleinschmidt and Oliver Schlotterer for numerous enlightening discussions during all stages of this project and for carefully reading the manuscript. I would also like to thank them for ongoing collaborations on related projects that initiated this work. I am supported by the International Max Planck Research School for Mathematical and Physical Aspects of Gravitation, Cosmology and Quantum Field Theory.

Appendix A. Complete reference for the modular graph forms package

In this appendix, we give a complete reference of all symbols defined in the `ModularGraphForms` package, all functions and their options and detailed instructions how to load the package. In appendix A.4, we show how the integrals appearing in the four-gluon amplitude of the heterotic string discussed in [29] can be computed using the `ModularGraphForms` package.

Within `Mathematica`, short descriptions of the various symbols, functions and options can be displayed using the `Information` function, e.g. by running `?g`. A list of all the symbols defined in the package is printed by running `?ModularGraphForms`*. The options

and default values for a function are accessible via the `Options` function, e.g.

```
In[53]:= Options[CBasis]

Out[53]= {basis→C} .
```

A.1. Files and loading the package

The `Mathematica` package `ModularGraphForms` includes the three files `ModularGraphForms.m`, `DiIds.txt` and `TriIds.txt`. The first one provides the package itself, whereas the two text files contain the basis decompositions described in section 9 for dihedral and trihedral graphs, respectively. The package loads the latter files automatically and expects them in the same directory, in which also the `ModularGraphForms.m` file is saved. However, the text files can also be imported into `Mathematica` using the `Get` function and can be used independently of the `ModularGraphForms` package.

To load the package, call the `Get` function on the `ModularGraphForms.m` file. Either the full path can be provided,

```
In[54]:= Get["/home/user/ModularGraphForms.m"]
```

or, if the files are placed in one of the directories in `Mathematica`'s search path, it is sufficient to run

```
In[55]:= Get["ModularGraphForms.m"] .
```

A list of the directories in `Mathematica`'s search path is available in the global variable `$Path` and includes the current directory, which by default is the directory in which the current notebook is saved.

A.2. Symbols

The `ModularGraphForms` package defines a number of symbols used for the various objects in this paper. For most of these symbols, a 2D-notation is implemented which makes the output easier to read. E.g. τ_2 is represented by `tau[2]`, but printed as

```
In[56]:= tau[2]

Out[56]= τ2 .
```

These 2D-outputs can be copied to input cells and used for further computations. The input form of the 2D-output can be accessed by the function `InputForm`, e.g.

```
In[57]:= InputForm[τ2]

Out[57]= tau[2] .
```

Using the `$Assumptions` variable, the `ModularGraphForms` package sets the global assumption that $\tau_2 > 0$. This is helpful e.g. when simplifying equations.

A.2.1. General symbols. Five general symbols used by the `ModularGraphForms` package are

`Mathematica` symbol	Description
`tau`	modular parameter τ
`tauBar`	$\bar{\tau}$
`tau[2]`	$\tau_2 = \text{Im}\ \tau$
`y`	$y = \pi\tau_2$
`zeta[k]`	ζ_k as defined in (1.4)
`bCoeff`	coefficient in the sieve algorithm, cf. `CSieveDecomp`

A.2.2. Modular graph forms. The conventions for two-, three- and four-point MGFs were introduced in detail in section 4. The symbols used to represent MGFs, (non-)holomorphic Eisenstein series and real and complex basis elements are

`Mathematica` symbol	Description
`c[...]`	MGF, cf. Section 4
`a[...]`	$\mathcal{A}\left[\begin{smallmatrix}A\\B\end{smallmatrix}\right]$ as defined in (4.4)
`intConst[...]`	integration constant, cf. `CSieveDecomp`
`intConstBar[...]`	complex conjugate of `intConst`
`g[k]`	G_k as defined in (2.31)
`gBar[k]`	$\overline{\text{G}}_k$
`gHat[2]`	$\widehat{\text{G}}_2$ as defined in (2.33)
`gBarHat[2]`	$\widehat{\overline{\text{G}}}_2$
`e[k1,...,kr]`	E_k as defined in (2.32) and $\text{E}_{k_1,\ldots,k_r}$ as defined in (9.5)
`ep[k1,...,kr]`	$\text{E}^{\text{I}}_{k_1,\ldots,k_r}$ as defined in (9.5)
`b[k1,...,kr]`	$\text{B}_{k_1,\ldots,k_r}$ as defined in (9.9)
`bp[k1,...,kr]`	$\text{B}^{\text{I}}_{k_1,\ldots,k_r}$ as defined in (9.9)

Note that MGFs are represented by the symbol `c`, but are printed with a capital `C`. When copying this output into an input cell, the capital `C` should not be changed into a lowercase `c`. Furthermore, the basis elements listed here are meaningful only for the indices defined in (9.5) and (9.9).

The `Mathematica` symbols used to represent Cauchy–Riemann derivatives of real and complex basis elements of MGFs are

`Mathematica` symbol	Description
`nablaE[n,{k1,...,kr}]`	$\nabla_0^n \mathrm{E}_{k_1,\ldots,k_r}$
`nablaBarE[n,{k1,...,kr}]`	$\overline{\nabla_0^n \mathrm{E}_{k_1,\ldots,k_r}}$
`nablaEp[n,{k1,...,kr}]`	$\nabla_0^n \mathrm{E}'_{k_1,\ldots,k_r}$
`nablaBarEp[n,{k1,...,kr}]`	$\overline{\nabla_0^n \mathrm{E}'_{k_1,\ldots,k_r}}$
`nablaB[n,{k1,...,kr}]`	$\nabla_0^n \mathrm{B}_{k_1,\ldots,k_r}$
`nablaBarBBar[n,{k1,...,kr}]`	$\overline{\nabla_0^n \mathrm{B}_{k_1,\ldots,k_r}}$
`nablaBp[n,{k1,...,kr}]`	$\nabla_0^n \mathrm{B}'_{k_1,\ldots,k_r}$
`nablaBarBpBar[n,{k1,...,kr}]`	$\overline{\nabla_0^n \mathrm{B}'_{k_1,\ldots,k_r}}$

The derivative operator ∇_0 and its complex conjugate are defined in (5.44). The zeroth derivative returns the argument, e.g.

```
In[58]:= nablaE[0,{5}]

Out[58]= E5
```

A.2.3. Iterated Eisenstein integrals. For compatibility with the data provided in the ancillary file of [30], the `ModularGraphForms` package defines the following symbols for iterated Eisenstein integrals, although no manipulations of these objects can be performed within this package.

`Mathematica` symbol	Description
`esv`$\left[\begin{smallmatrix} j_1 & \cdots & j_l \\ k_1 & \cdots & k_l \end{smallmatrix}\right]$	$\mathcal{E}^{\mathrm{sv}}\left[\begin{smallmatrix} j_1 & \cdots & j_l \\ k_1 & \cdots & k_l \end{smallmatrix};\tau\right]$
`esvS`$\left[\begin{smallmatrix} j_1 & \cdots & j_l \\ k_1 & \cdots & k_l \end{smallmatrix}\right]$	$\mathcal{E}^{\mathrm{sv}}\left[\begin{smallmatrix} j_1 & \cdots & j_l \\ k_1 & \cdots & k_l \end{smallmatrix};-\frac{1}{\tau}\right]$
`esvBar`$\left[\begin{smallmatrix} j_1 & \cdots & j_l \\ k_1 & \cdots & k_l \end{smallmatrix}\right]$	$\overline{\mathcal{E}^{\mathrm{sv}}\left[\begin{smallmatrix} j_1 & \cdots & j_l \\ k_1 & \cdots & k_l \end{smallmatrix};\tau\right]}$
`betasv`$\left[\begin{smallmatrix} j_1 & \cdots & j_l \\ k_1 & \cdots & k_l \end{smallmatrix}\right]$	$\beta^{\mathrm{sv}}\left[\begin{smallmatrix} j_1 & \cdots & j_l \\ k_1 & \cdots & k_l \end{smallmatrix};\tau\right]$
`betasvS`$\left[\begin{smallmatrix} j_1 & \cdots & j_l \\ k_1 & \cdots & k_l \end{smallmatrix}\right]$	$\beta^{\mathrm{sv}}\left[\begin{smallmatrix} j_1 & \cdots & j_l \\ k_1 & \cdots & k_l \end{smallmatrix};-\frac{1}{\tau}\right]$
`betasvBar`$\left[\begin{smallmatrix} j_1 & \cdots & j_l \\ k_1 & \cdots & k_l \end{smallmatrix}\right]$	$\overline{\beta^{\mathrm{sv}}\left[\begin{smallmatrix} j_1 & \cdots & j_l \\ k_1 & \cdots & k_l \end{smallmatrix};\tau\right]}$

As for MGFs, the matrices can be inserted in `Mathematica` either as nested lists or as 2D input, cf `In[8]`. For the definitions of the iterated Eisenstein integrals, see [30].

A.2.4. Koba–Nielsen integrals. For the evaluation and representation of Koba–Nielsen integrals and their generating series, the following symbols are defined.

Mathematica symbol	Description
`eta[k1,...,kr]`	$\eta_{k_1,\ldots,k_r}$ expansion variable as in [30]
`etaBar[k1,...,kr]`	$\bar{\eta}_{k_1,\ldots,k_r}$
`s[k1,...,kr]`	$s_{k_1,\ldots,k_r}$ as defined in (1.1) and (8.19)
`fz[a, i, j]`	$f^{(a)}_{ij}$ as defined in (2.12b)
`fBarz[b, i, j]`	$\overline{f^{(b)}_{ij}}$ as defined in (2.12c)
`gz[i, j]`	G_{ij} as defined in (2.7)
`cz[a, b, i, j]`	$C^{(a,b)}_{ij}$ as defined in (2.18)
`vz[a,{k1,...,kr}]`	$V_a(k_1,\ldots,k_r)$ as defined in (2.16)
`vBarz[b,{k1,...,kr}]`	$\overline{V_b(k_1,\ldots,k_r)}$

Symbols which represent functions which can appear in the integrand of a Koba–Nielsen integral have the suffix `z`.

A.3. Functions

The functions in the `ModularGraphForms` package are organized into three main categories: dihedral functions only manipulate dihedral MGFs and carry the prefix `Di`. Trihedral functions only manipulate trihedral MGFs and carry the prefix `Tri`. General functions act on MGFs of all supported graph topologies or perform other tasks which are not specific to any graph topology. They carry a prefix `C`. On top of these, there is limited support for four-point manipulations in the form of the function `TetCSimplify` and a function to expand Koba–Nielsen integrals in MGFs.

A.3.1. General functions.

`CBasis`

The function **`CBasis`** returns a list of basis elements for MGFs.

ARGUMENTS **`CBasis`** accepts two arguments, corresponding to the holomorphic and antiholomorphic modular weight of the basis.

RETURN VALUE **`CBasis`** returns the basis of MGFs at the modular weight passed as the arguments as listed in tables 3 and 4, excluding the zeta values ζ_3, ζ_5 and ζ_3^2. Note that at weight $(a+k, a-k)$, the basis elements in table 3 have weight $(a+k, a-k)$, whereas in table 4, they have weight $(0, -2k)$.

OPTIONS If the option **`basis`** is set to the string `"C"` (the default) the basis from table 3 is returned, if the option **`basis`** is set to the string `"nablaE"`, the basis from table 4 is returned. No other values for **`basis`** are admissible.

WARNING

- If the sum of the holomorphic- and antiholomorphic modular weights passed in the arguments is odd, the warning `CBasis::incorrModWeight` is issued and **`CBasis`** returns an empty list.
- If the sum of the holomorphic- and antiholomorphic modular weights passed in the arguments is less than four, the warning `CBasis::tooLowWeight` is issued and **`CBasis`** returns an empty list.

- If the basis for the modular weight passed to **CBasis** is not implemented, the warning CBasis::noBasis is issued and **CBasis** returns an empty list.

EXAMPLES

In[59]:= CBasis[3, 7]

Out[59]= $\{C\left[\begin{smallmatrix}1&1&1\\1&1&5\end{smallmatrix}\right], C\left[\begin{smallmatrix}3&0\\7&0\end{smallmatrix}\right], C\left[\begin{smallmatrix}1&0\\3&0\end{smallmatrix}\right] C\left[\begin{smallmatrix}2&0\\4&0\end{smallmatrix}\right], \frac{\pi^2 C\left[\begin{smallmatrix}1&0\\5&0\end{smallmatrix}\right] E_2}{\tau_2^2}, C\left[\begin{smallmatrix}0&1&2\\2&0&5\end{smallmatrix}\right]\}$

In[60]:= CBasis[3, 7, basis→"nablaE"]

Out[60]= $\{\overline{\nabla}^2 E_{2,3}, \overline{\nabla}^2 E_5, \overline{\nabla} E_2\ \overline{\nabla} E_3, E_2\ \overline{\nabla}^2 E_3, \overline{\nabla}^2 \overline{B}'_{2,3}\}$

CCheckConv

The function CCheckConv tests if MGFs are convergent or divergent.

ARGUMENTS CCheckConv accepts one argument which is an arbitrary expression, possibly containing MGFs of any topology and Eisenstein series.

RETURN VALUE CCheckConv returns True or False. If the argument contains an MGF which is divergent according to the conditions discussed in section 8.1 or a E_k, G_k or $\overline{G}_k$ with $k < 2$, the function returns False, otherwise it returns True.

EXAMPLES

In[61]:= CCheckConv[e[1] c[$\begin{smallmatrix}2&0\\3&0\end{smallmatrix}$]]

Out[61]= False

Since E_1 is divergent, the return value is False, even though $\mathcal{C}\begin{bmatrix}2&0\\3&0\end{bmatrix} = 0$.

In[62]:= CCheckConv[c[$\begin{smallmatrix}1&2\\-2&2\end{smallmatrix}$, $\begin{smallmatrix}1&2\\1&2\end{smallmatrix}$, $\begin{smallmatrix}1&2\\-2&2\end{smallmatrix}$, $\begin{smallmatrix}1&2\\1&2\end{smallmatrix}$, $\begin{smallmatrix}2&2\\2&2\end{smallmatrix}$]]

Out[62]= False

Since the last condition in (8.9) is violated, the return value is False.

In[63]:= CCheckConv[c[$\begin{smallmatrix}1&2\\-2&2\end{smallmatrix}$, $\begin{smallmatrix}1&2\\1&2\end{smallmatrix}$, $\begin{smallmatrix}1&2\\-2&2\end{smallmatrix}$, $\begin{smallmatrix}2&3\\2&1\end{smallmatrix}$, $\begin{smallmatrix}2&2\\2&2\end{smallmatrix}$]]

Out[63]= True

Since here $\check{c}_4$ as defined below (8.6) is increased, the last condition in (8.9) is also satisfied and the return value is True.

CComplexConj

The function CComplexConj computes the complex conjugate of an expression.

ARGUMENTS CComplexConj accepts one arbitrary argument.

RETURN VALUE CComplexConj returns its argument with all MGFs complex conjugated and written in their canonical representation. This includes Eisenstein series, complex basis elements (according to (9.10)) Cauchy–Riemann derivatives of basis elements and integration constants, unless the MGF in the argument is real.

EXAMPLES

In[64]:= `CComplexConj[{g[4], b[2, 4], intConst[1 2 1; 1 1 4], nablaB[1, {2, 4}]}]`

Out[64]= $\left\{\overline{\mathrm{G}}_4,\ -\mathrm{B}_{2,4}-2\,\mathrm{E}_2\,\mathrm{E}_4+\frac{2\,\mathrm{E}_{2,4}}{9},\ \overline{\texttt{intConst}\left[\begin{smallmatrix}1&2&1\\1&1&4\end{smallmatrix}\right]},\ \overline{\nabla\mathrm{B}}_{2,4}\right\}$

`CConvertToNablaE` and `CConvertFromNablaE`

The functions `CConvertToNablaE` and `CConvertFromNablaE` convert an expression between the bases given in tables 3 and 4.

ARGUMENTS Both `CConvertToNablaE` and `CConvertFromNablaE` accept one arbitrary argument.

RETURN VALUE `CConvertToNablaE` replaces all of the basis elements in table 3 in its argument with their expansions in the basis of table 4. `CConvertFromNablaE` replaces all of the basis elements in table 4 in its argument with their expansions in the basis of table 3. On top of the elements listed explicitly in these tables, $\nabla_0^n\mathrm{E}_k$ and $\mathcal{C}\begin{bmatrix}k+n&0\\k-n&0\end{bmatrix}$ are rewritten according to (5.50) for any n and k. The results are not manipulated any further and MGFs in the argument which are not in the basis to be converted are left untouched.

EXAMPLES

In[65]:= `CConvertToNablaE[c[1 1 4; 1 1 2]]`

Out[65]= $\frac{\pi^5\,\nabla\mathrm{B}_{2,3}}{18\,\tau_2^6}-\frac{\pi^5\,\nabla\mathrm{B}'_{2,3}}{18\,\tau_2^6}-\frac{\pi^5\,\mathrm{E}_3\,\nabla\mathrm{E}_2}{12\,\tau_2^6}+\frac{\pi^5\,\mathrm{E}_2\,\nabla\mathrm{E}_3}{12\,\tau_2^6}+\frac{41\,\pi^5\,\nabla\mathrm{E}_5}{140\,\tau_2^6}+\frac{\pi^5\,\nabla\mathrm{E}_{2,3}}{24\,\tau_2^6}-\frac{\pi^5\,\nabla\mathrm{E}_2\,\zeta_3}{36\,\tau_2^6}$

In[66]:= `CConvertToNablaE[c[1 2 3; 1 1 2]]`

Out[66]= $\mathrm{C}\left[\begin{smallmatrix}1&2&3\\1&1&2\end{smallmatrix}\right]$

In[67]:= `CConvertFromNablaE[Out[65]]`

Out[67]= $\mathrm{C}\left[\begin{smallmatrix}1&1&4\\1&1&2\end{smallmatrix}\right]$

`CHolCR` and `CAHolCR`

The functions `CHolCR` and `CAHolCR` compute the holomorphic- and antiholomorphic Cauchy–Riemann derivative, respectively.

ARGUMENTS Both `CHolCR` and `CAHolCR` accept one argument which should be a functional

expression (e.g. a polynomial) involving MGFs and Eisenstein series.

RETURN VALUE `CHolCR` returns the holomorphic Cauchy–Riemann derivative of its argument, using the derivative operator defined in (5.40), by applying (5.47). The result is always given in terms of lattice sums, even if the argument involves Cauchy–Riemann derivatives of basis elements. The generalized Ramanujan identities from section 5.5 are not applied. If the argument contains a divergent graph with a closed holomorphic subgraph, HSR is applied before the derivative is taken, while $\mathcal{C}\begin{bmatrix}2 & 0\\ 0 & 0\end{bmatrix}$ is not replaced by $\widehat{\mathrm{G}}_2$. The output is not manipulated any further. `CAHolCR` returns the antiholomorphic Cauchy–Riemann derivative.

OPTIONS The Boolean option `divDer` specifies if derivatives of divergent graphs are taken or not. If it is set to `False` (the default is `True`) and a divergent MGF appears in the argument, `CHolCR` and `CAHolCR` return `Nothing`.

WARNING

- If the argument of `CHolCR` contains a divergent MGF, the warning **`CHolCR::derOfDiv`** is issued (and c.c.).
- The argument is passed to `CModWeight` (see below), to check if it has homogeneous modular weight. If it does not, the warning **`CModWeight::WeightNotHom`** is issued and `Nothing` is returned.

EXAMPLES

```
In[68]:= CHolCR[{nablaE[1,{3}],gBarHat[2],c[1 1 1]}]
```

$$\text{Out[68]}= \left\{\frac{12\,\mathrm{C}\left[\begin{smallmatrix}5 & 0\\ 1 & 0\end{smallmatrix}\right]\tau_2^4}{\pi^3},\ \frac{\pi}{\tau^2},\ \mathrm{C}\left[\begin{smallmatrix}1 & 1 & 2\\ 1 & 1 & 0\end{smallmatrix}\right]+\mathrm{C}\left[\begin{smallmatrix}1 & 2 & 1\\ 1 & 0 & 1\end{smallmatrix}\right]+\mathrm{C}\left[\begin{smallmatrix}2 & 1 & 1\\ 0 & 1 & 1\end{smallmatrix}\right]\right\}$$

```
In[69]:= CHolCR[c[0 0 3]]
```

CHolCR : Warning: You are generating the holomorphic Cauchy−Riemann derivative of the divergent expression $\mathrm{C}\left[\begin{smallmatrix}0 & 0 & 3\\ 1 & 1 & 3\end{smallmatrix}\right]$. This may be problematic.

$$\text{Out[69]}= -6\,\mathrm{C}\left[\begin{smallmatrix}4 & 0\\ 4 & 0\end{smallmatrix}\right]+\frac{2\pi\,\mathrm{C}\left[\begin{smallmatrix}3 & 0\\ 3 & 0\end{smallmatrix}\right]}{\tau_2}$$

`CLaurentPoly`

The function `CLaurentPoly` replaces basis elements by their Laurent polynomials.

ARGUMENTS `CLaurentPoly` accepts one arbitrary argument.

RETURN VALUE `CLaurentPoly` returns its argument with the real basis elements (9.5), the complex basis elements (9.9), their Cauchy–Riemann derivatives and complex conjugates, as well as all non-holomorphic- and holomorphic Eisenstein series (including $\widehat{\mathrm{G}}_2$) replaced by their Laurent polynomials. The Laurent polynomials of the real and complex basis elements are given in (9.7) and (9.12), respectively, their derivatives are obtained using (9.14).

OPTIONS The Boolean option `usey` specifies if the output is given in terms of τ_2 (`False`) or $y = \pi\tau_2$ (`True`, the default).

EXAMPLES

```
In[70]:= CLaurentPoly[nablaBarBBar[2, {2, 4}]]
```

$$\text{Out[70]}= -\frac{8\,y^8}{30375\,\pi^2} - \frac{105\,\zeta_3\,\zeta_7}{8\,\pi^2\,y^2} + \frac{25\,\zeta_9}{12\,\pi^2\,y}$$

```
In[71]:= CLaurentPoly[{g[6], gHat[2], e[7]}, usey→False]
```

$$\text{Out[71]}= \left\{\frac{2\,\pi^6}{945},\ \frac{\pi^2}{3} - \frac{\pi}{\tau_2},\ \frac{4\,\pi^7\,\tau_2^7}{18243225} + \frac{231\,\zeta_{13}}{512\,\pi^6\,\tau_2^6}\right\}$$

CListHSRs

The function CListHSRs lists MGFs with closed holomorphic subgraphs in an expression.

ARGUMENTS CListHSRs accepts one arbitrary argument.

RETURN VALUE CListHSRs returns a list with all dihedral and trihedral graphs with closed holomorphic subgraphs appearing somewhere in its argument. If the argument does not contain any dihedral or trihedral graphs, CListHSRs returns the empty list.

EXAMPLES

```
In[72]:= CListHSRs[c[1 1 1 2 1 2, 1 2 2 0] + c[7 0 3 0]]
```

$$\text{Out[72]}= \left\{C\left[\begin{array}{c|c|c} 1 & 2 & 2 \\ 0 & 1 & 0 \end{array}\right]\right\}$$

CModWeight

The function CModWeight determines the modular weight of an expression.

ARGUMENTS CModWeight accepts one argument which can be either a modular form (possibly of trivial modular weight), a product of modular forms or a sum of products of modular forms.

RETURN VALUE CModWeight returns a list with two elements, corresponding to the holomorphic and antiholomorphic modular weight, respectively.

WARNING

- If a sum is passed to CModWeight and the modular weights of the summands do not agree, CModWeight returns Null and the warning **CModWeight::**WeightNotHom,, containing a list of the modular weights appearing in the sum, is issued.
- If symbols appear in the argument of CModWeight, for which no modular weight is implemented, CModWeight returns the modular weight which the expression would have if all symbols of unknown weight were modular invariant. A list of the terms whose weight could not be determined is printed as part of the warning **CModWeight::**UnknownExp..

EXAMPLES

```
In[73]:= CModWeight[tau[2]^-2 nablaBarE[1, {2}] nablaE[2, {4}] + nablaBp[1, {2, 4}]]

Out[73]= {0, -2}

In[74]:= CModWeight[e[2] + C[2 0; 2 0]]

         CModWeight: The modular weight of the argument is not homogeneous, the weights
         {2,2}, {0,0} appear.

In[75]:= CModWeight[g[2] g[4]]

         CModWeight: Expression(s) {G2} found whose modular weight could not be determined.
          The returned weight assumes them to be modular invariant.

In[76]:= {4, 0}
```

CSieveDecomp

The function CSieveDecomp decomposes an MGF using the sieve algorithm.

ARGUMENTS CSieveDecomp accepts one argument which can be either a dihedral or a trihedral MGF without closed holomorphic subgraph.

RETURN VALUE CSieveDecomp performs the sieve algorithm on its argument as discussed in section 7 and returns the decomposition obtained. If the holomorphic modular weight is larger than the antiholomorphic one, CSieveDecomp takes holomorphic derivatives, otherwise antiholomorphic ones. If both modular weights of the argument are equal, an integration constant intConst labeled by the exponent matrix of the argument and dressed with an appropriate factor of $\frac{\pi}{\tau_2}$ is added to the final decomposition. If the basis into which the argument is decomposed is not linearly independent, the output contains free parameters with head bCoeff.

OPTIONS

Option	Possible values	Default value	Description
verbose	True, False	False	Activates verbose output
divDer	True, False	False	Activates decomposition of divergent graphs
basis	List of MGFs	{ }	Basis elements for decomposition
addIds	List of replacement rules for MGFs	{ }	Additional replacement rules applied to each derivative
CSimplifyOpts	Option assignments of CSimplify	See below	Options passed to CSimplify when simplifying the derivatives

The default value of CSimplifyOpts is {basisExpandG → True}. If the option basis is set to the empty list, the appropriate basis is determined automatically using CBasis. Since this basis does not contain powers of E_1, it is not sufficient for the decomposition of divergent graphs. The basis elements have to be MGFs without closed holomorphic subgraphs of the same modular weight as the argument. Divergent basis elements are only admissible if divDer is set to True.

WARNING

- If the argument of `CSieveDecomp` is divergent, the warning **`CSieveDecomp::divArg`** is issued. The decomposition proceeds only if `divDer` is set to `True`.
- If one of the basis elements is divergent, but the argument is not, the warning **`CSieveDecomp::divBasis`** is issued.
- If a holomorphic Eisenstein series could not be canceled in one of the derivatives, the warning **`CSieveDecomp::noSol`** is issued. This happens e.g. if the basis is not large enough.
- If in one of the derivatives, an undecomposed graph appears in the coefficient of a holomorphic Eisenstein series, the warning **`CSieveDecomp::holEisenCoeffNoBasis`** is issued and the algorithm interrupted. MGFs are considered decomposed if they appear in the basis given by `CBasis`. For modular weight $a+b \leqslant 12$, these undecomposed graphs will be divergent.

EXAMPLES

In[77]:= `CSieveDecomp[c[{1},{1,1},{1,1}]]`

Out[77]= $2\,\mathrm{C}\left[\begin{smallmatrix}1&1&3\\1&1&3\end{smallmatrix}\right] - \frac{2\pi^5\,\mathrm{E}_5}{5\tau_2^5} + \frac{\pi^5\,\mathtt{intConst}\left[\begin{smallmatrix}1&1&1\\1&1&1\end{smallmatrix}\right]}{\tau_2^5}$

In[78]:= `CSieveDecomp[c[{1,1,1}], basis→{c[{0,1,2}], τ2^3/tau[2]^3 e[3]}]`

Out[78]= $-\mathrm{C}\left[\begin{smallmatrix}0&1&2\\1&1&1\end{smallmatrix}\right] + 2\,\mathtt{bCoeff}[2]\,\mathrm{C}\left[\begin{smallmatrix}0&1&2\\1&1&1\end{smallmatrix}\right] + \frac{\pi^3\,\mathtt{bCoeff}[2]\,\mathrm{E}_3}{\tau_2^3} + \frac{\tau^3\,\mathtt{intConst}\left[\begin{smallmatrix}1&1&1\\1&1&1\end{smallmatrix}\right]}{\tau_2^3}$

In[79]:= `CSieveDecomp[c[{0,1,1,1}]]`

> CSieveDecomp : The 1st derivative contains the undecomposed graph(s) $\{\mathrm{C}\left[\begin{smallmatrix}0&1&1\\1&0&1\end{smallmatrix}\right]\}$ as a coefficient of a holomorphic Eisenstein series.

Out[79]= $\mathrm{C}\left[\begin{smallmatrix}0&1&1&1\\3&0&1&1\end{smallmatrix}\right]$

In[80]:= `CSieveDecomp[c[{0,1,1,1}], addIds→{c[{0,1,1}]→ -π^2 E1^2/(2 tau[2]^2) + π^2 E2/(2 tau[2]^2)}]`

Out[80]= $2\,\mathrm{C}\left[\begin{smallmatrix}3&0\\5&0\end{smallmatrix}\right] - 2\,\mathrm{C}\left[\begin{smallmatrix}1&1&1\\1&1&3\end{smallmatrix}\right] - \frac{\pi^2\,\mathrm{C}\left[\begin{smallmatrix}1&0\\3&0\end{smallmatrix}\right]\mathrm{E}_2}{\tau_2^2}$

`CSimplify`

The function `CSimplify` performs all known simplifications for MGFs.

ARGUMENTS `CSimplify` accepts one arbitrary argument.

RETURN VALUE `CSimplify` applies, in this order, the specialized functions `TetCSimplify`, `TriCSimplify` and `DiCSimplify` to its argument until it no longer changes and returns the result.

OPTIONS `CSimplify` accepts all the options of both `TriCSimplify` and `DiCSimplify` and passes them to these functions when they are called.

EXAMPLES

In[81]:= `CSimplify[c[{1,0},{1,1},{1,2}]]`

Out[81]= $\frac{3}{2}C\left[\begin{smallmatrix}3&0\\1&0\end{smallmatrix}\right]^2-\frac{1}{2}C\left[\begin{smallmatrix}6&0\\2&0\end{smallmatrix}\right]-\frac{1}{2}C\left[\begin{smallmatrix}4&0\\2&0\end{smallmatrix}\right]\hat{G}_2-\frac{\pi^2 G_4}{\tau_2^2}+\frac{3\pi C\left[\begin{smallmatrix}5&0\\1&0\end{smallmatrix}\right]}{\tau_2}-\frac{\pi C\left[\begin{smallmatrix}3&0\\1&0\end{smallmatrix}\right]\hat{G}_2}{\tau_2}$

In[82]:= `CSimplify[c[{1,0},{1,1},{1,2}], tri3ptFayHSR→True]`

Out[82]= $\frac{3}{2}C\left[\begin{smallmatrix}3&0\\1&0\end{smallmatrix}\right]^2-\frac{1}{2}C\left[\begin{smallmatrix}6&0\\2&0\end{smallmatrix}\right]-\frac{1}{2}C\left[\begin{smallmatrix}4&0\\2&0\end{smallmatrix}\right]\hat{G}_2-\frac{\pi^2 G_4}{\tau_2^2}+\frac{3\pi C\left[\begin{smallmatrix}5&0\\1&0\end{smallmatrix}\right]}{\tau_2}-\frac{\pi C\left[\begin{smallmatrix}3&0\\1&0\end{smallmatrix}\right]\hat{G}_2}{\tau_2}$

In[83]:= `CSimplify[c[{},{1},{1},{1},{},{1}]]`

Out[83]= 0

`CSort`

The function `CSort` sorts MGFs into their canonical representation.

ARGUMENTS `CSort` accepts one arbitrary argument.

Return value. `CSort` returns its argument with all MGFs written in their canonical representation as discussed in section 5.1.

EXAMPLES

In[84]:= `CSort[{c[{2,2},{1,1},{1,1},{1,1},{1,2},{2,2},{1,1},{1,2}]}, c[{1,1},{1,0},{0,1},{1,0}]]`

Out[84]= $\left\{C\left[\begin{array}{cc|cc|cc}1&1&1&1&2&2\\1&1&1&1&1&1\\\hline1&2&2&2&1&1\\1&1&1&1&1&2\end{array}\right],\ C\left[\begin{smallmatrix}1&1&1&1\\0&1&1&0\end{smallmatrix}\right]\right\}$

A.3.2. Dihedral functions.

`DiHolMomConsId` and `DiAHolMomConsId`

The functions `DiHolMomConsId` and `DiAHolMomConsId` generate holomorphic and antiholomorphic dihedral momentum-conservation identities, respectively.

ARGUMENTS Both `DiHolMomConsId` and `DiAHolMomConsId` accept a dihedral MGF as

their only argument.

RETURN VALUE `DiHolMomConsId` returns the holomorphic momentum conservation identity (5.24) of the seeds given in the argument as an equation with rhs 0. `DiAHolMomConsId` returns the antiholomorphic momentum-conservation identity. No further manipulation as e.g. sorting into the canonical representation are performed on the output.

WARNING If the argument of `DiHolMomConsId` is divergent according to `CCheckConv`, the warning **`DiHolMomConsId::`**`divDiHolMomCons` (and c.c.) is issued.

EXAMPLES

```
In[85]:= DiHolMomConsId[c[1 1 2]]

         DiAHolMomConsId[c[1 1 2]]

Out[85]= C[0 1 2] + C[1 0 2] + C[1 1 1] == 0
         1 1 1     1 1 1     1 1 1

Out[86]= C[1 1 2] + C[1 1 2] + C[1 1 2] == 0
         0 1 1     1 0 1     1 1 0

In[87]:= DiHolMomConsId[c[0 1 2]]

         DiHolMomConsId : You are generating the holomorphic momentum-conservation
         identity of the divergent seed C[0 1 2]. Divergent seeds can lead to inconsistent identities.
                                         1 0 2

Out[87]= C[-1 1 2] + C[0 0 2] + C[0 1 1] == 0
          1 0 2     1 0 2     1 0 2
```

`DiCSimplify`

The function `DiCSimplify` performs all known dihedral simplifications.

ARGUMENTS `DiCSimplify` accepts one arbitrary argument.

RETURN VALUE `DiCSimplify` returns the expression given as the argument with all dihedral MGFs (including one-loop graphs such as Eisenstein series) rewritten in a simplified form, if possible. This is done by performing the following manipulations on all dihedral graphs, until the result does not change any more.

(a) Apply HSR (6.6) and its divergent analog (8.28).
(b) Set $\mathcal{C}\left[\emptyset\right] = 1$, cf (5.10).
(c) Factorize on $\begin{bmatrix}0\\0\end{bmatrix}$ columns according to (5.34) and (8.49).
(d) Set $\mathcal{C}\begin{bmatrix}a\\b\end{bmatrix} = 0$, cf (5.9).
(e) Remove entries of -1 by using momentum conservation as described in section 7.
(f) Sort dihedral MGFs into their canonical representation as described in section 5.1.
(g) Set graphs with odd $|A| + |B|$ to zero.
(h) Rewrite $\mathcal{C}\begin{bmatrix}k & 0\\0 & 0\end{bmatrix} = \mathrm{G}_k$ and c.c., cf (2.30b).
(i) Rewrite $\mathcal{C}\begin{bmatrix}2 & 0\\0 & 0\end{bmatrix} = \widehat{\mathrm{G}}_2$ and c.c., cf (2.36).
(j) Set G_k with k odd to zero and c.c.
(k) Rewrite $\mathcal{C}\begin{bmatrix}k & 0\\k & 0\end{bmatrix} = (\frac{\pi}{\tau_2})^k \mathrm{E}_k$, cf (2.30c).
(l) Apply generalized Ramanujan identities discussed in section 5.5 and expand holomorphic Eisenstein series in the ring of G_4 and G_6.
(m) Apply basis decompositions discussed in section 9, in the basis listed in table 3.

Within this process, the steps (b) to (l) are repeated until the result no longer changes, before step (m) is executed.

OPTIONS

Both options `divHSR` and `diDivHSR` have to be `True` for divergent graphs to be included in step (a).

WARNING

Option	Possible values	Default value	Description
`basisExpandG`	`True`, `False`	`False`	Activates step (l)
`momSimplify`	`True`, `False`	`True`	Deactivates step (e)
`repGHat2`	`True`, `False`	`True`	Deactivates step (i)
`useIds`	`True`, `False`	`True`	Deactivates step (m)
`diHSR`	`True`, `False`	`True`	Deactivates step (a)
`divHSR`	`True`, `False`	`True`	Deactivates step (a) for divergent graphs
`diDivHSR`	`True`, `False`	`True`	Deactivates step (a) for divergent graphs

- If a graph in the argument contains a $\begin{bmatrix}0\\0\end{bmatrix}$ column next to a $\begin{bmatrix}1\\0\end{bmatrix}$ or $\begin{bmatrix}1\\0\end{bmatrix}$ column, the warning `DiCSimplify::dangerousFact` is issued and the modified factorization rule (8.49) applied.
- If a divergent graph with a holomorphic subgraph is encountered but HSR cannot be performed because either one of the options `divHSR` or `diDivHSR` is set to `False`, the warning `DiCSimplify::divHSRNotPossible` is issued.

EXAMPLES

```
In[88]:= DiCSimplify[c[1 2 2 2
                       0 0 1 2]]
```

$$\text{Out[88]}= 3\,C\left[\begin{smallmatrix}3&0\\1&0\end{smallmatrix}\right]C\left[\begin{smallmatrix}4&0\\2&0\end{smallmatrix}\right] - 15\,C\left[\begin{smallmatrix}7&0\\3&0\end{smallmatrix}\right] - 9C\left[\begin{smallmatrix}0&2&5\\1&0&2\end{smallmatrix}\right] + \frac{21}{2}C\left[\begin{smallmatrix}1&1&5\\1&1&1\end{smallmatrix}\right] - C\left[\begin{smallmatrix}5&0\\3&0\end{smallmatrix}\right]\hat{G}_2 + \frac{1}{2}C\left[\begin{smallmatrix}1&1&3\\1&1&1\end{smallmatrix}\right]\hat{G}_2 - \frac{2\pi^2\,C\left[\begin{smallmatrix}5&0\\1&0\end{smallmatrix}\right]}{\tau_2^2} + \frac{6\pi\,C\left[\begin{smallmatrix}6&0\\2&0\end{smallmatrix}\right]}{\tau} - \frac{2\pi\,C\left[\begin{smallmatrix}4&0\\2&0\end{smallmatrix}\right]\hat{G}_2}{\tau_2}$$

```
In[89]:= DiCSimplify[c[1 2 2 2
                       0 0 1 2], momSimplify→False, useIds→False]
```

$$\text{Out[89]}= -3\,C\left[\begin{smallmatrix}2&2&3\\1&2&0\end{smallmatrix}\right] + C\left[\begin{smallmatrix}1&2&2\\0&1&2\end{smallmatrix}\right]\hat{G}_2 + \frac{\pi\,C\left[\begin{smallmatrix}2&2&2\\-1&1&2\end{smallmatrix}\right]}{\tau_2}$$

```
In[90]:= DiCSimplify[c[0 0 1 1 1 2 4
                       0 1 1 1 1 3 4]] // Simplify
```

DiCSimplify : The graph $C\left[\begin{smallmatrix}0&0&1&1&1&2&4\\0&1&1&1&1&3&4\end{smallmatrix}\right]$ is factorized and contains a (1,0) or (0,1) column. This may be problematic.

$$\text{Out[90]}= -\frac{\pi^8\,C\left[\begin{smallmatrix}1&0\\3&0\end{smallmatrix}\right](-6 + 6E_1 - 3E_1^2 + E_1^3)\,E_4 + C\left[\begin{smallmatrix}0&1&1&1&2&4\\1&1&1&1&3&4\end{smallmatrix}\right]\tau_2^8}{\tau_2^8}$$

A.3.3. Trihedral functions.

`TriHolMomConsId` and `TriAHolMomConsId`

The functions `TriHolMomConsId` and `TriAHolMomConsId` generate trihedral holomorphic and antiholomorphic momentum-conservation identities, respectively.

ARGUMENTS Both `TriHolMomConsId` and `TriAHolMomConsId` accept two arguments:

the first is a trihedral MGF, the second is one of the lists `{1, 2}`, `{2, 3}` or `{1, 3}`, where the order of the elements in the list does not matter.

RETURN VALUE `TriHolMomConsId` returns the holomorphic trihedral momentum-conservation identity (5.25) as an equation with rhs zero. Due to the permutation symmetry of the three blocks in a trihedral MGF, any two blocks can be involved in the momentum-conservation identity (i.e. have their holomorphic weight reduced) and the second argument of `TriHolMomConsId` specifies which two blocks should be used to generate the identity. `TriAHolMomConsId` generates the antiholomorphic momentum conservation identity. No further manipulation is performed on the output.

WARNING If either one of the blocks in the second argument is divergent as a dihedral MGF or if the trihedral MGF in the first argument has a three-point divergence (cf (8.6)), the warning **`TriHolMomConsId::`**`divTriHolMomCons` (and c.c.) is issued.

EXAMPLES

```
In[91]:= TriHolMomConsId[c[{1},{1,1},{1,1}],{1,2}]

Out[91]= C[{0|1,1|1,1}] - C[{1|0,1|1,1}] - C[{1|1,0|1,1}] == 0

In[92]:= TriAHolMomConsId[c[{1},{1,1},{1,1}],{3,2}]

Out[92]= - C[{1|0,1|1,1}] - C[{1|1,0|1,1}] + C[{1|1,1|0,1}] + C[{1|1,1|1,0}] == 0

In[93]:= TriHolMomConsId[c[{1},{0,1},{1,1}],{1,3}]

Out[93]= c[{0,1},{0,1},{1,1}] - c[{1},{0,1},{0,1}] - c[{1},{0,1},{1,0}] == 0
```

`TriFay`

The function `TriFay` generates trihedral Fay identities.

ARGUMENTS `TriFay` accepts up to two arguments. The first (mandatory) argument is a trihedral MGF, the second (optional) argument is a list of the form `{{b1,c1},{b2,c2}}`, where `ci` is a column number in the `bi`th block and the list selects two columns, both of the form $\begin{bmatrix} a \\ 0 \end{bmatrix}$ with $a \geqslant 1$ or $\begin{bmatrix} 0 \\ b \end{bmatrix}$ with $b \geqslant 1$ in the trihedral graph. If the second argument is omitted, `TriFay` selects the first suitable pair of columns automatically, starting from the left and trying holomorphic column pairs first.

RETURN VALUE `TriFay` returns an equation in which the lhs is the graph specified in the first argument and the rhs is given by (6.22) (or its complex conjugate), with the columns $\begin{bmatrix} a_1 \\ 0 \end{bmatrix}$ and $\begin{bmatrix} a_2 \\ 0 \end{bmatrix}$ selected by the second argument or determined automatically. No further manipulations are performed on the output.

WARNING If no second argument is passed to `TriFay` and no suitable pair of columns could be found, the warning **`TriFay::`**`noFayCols` is issued.

EXAMPLES

```
In[94]:= TriFay[c[1/0, 1 2/0 2, 1 2/1 0]]

Out[94]= C[1/0|1 2/0 2|1 2/1 0] == C[{}|2/2|1 2 2/1 0 0] - C[{}|1 2/0 2|1 1 2/0 1 0]
        + C[{}|1 2/1 0|2 2/0 2] - C[1/0|2/2|1 1 2/0 1 0] + C[2/0|2/2|1 2/1 0]

In[95]:= TriFay[c[0/1, 0 1/2 1, 0 2/1 2], {{1, 1}, {3, 1}}]

Out[95]= c[0/1, 0 1/2 1, 0 2/1 2] == C[{}|2/2|0 0 1/2 2 1] + C[{}|0 1/2 1|0 2/2 2]
        - C[{}|0 2/1 2|0 0 1/1 2 1] - C[0/1|2/2|0 0 1/1 2 1] + C[0/2|2/2|0 1/2 1]
```

`TriCSimplify`

The function `TriCSimplify` applies all known trihedral simplifications.

ARGUMENTS `TriCSimplify` accepts one arbitrary argument.

RETURN VALUE `TriCSimplify` returns the expression given as the argument with all trihedral MGFs rewritten in a simplified form, if possible. This is done by performing the following manipulations on all trihedral graphs, until the result does not change any more.

(a) Apply two-point HSR using the trihedral generalization of (6.6) as described in [19] and its divergent analog (8.31).
(b) Apply three-point HSR using the closed formula in [19].
(c) Set graphs with odd total modular weight $a + b$ to zero.
(d) Apply the topological simplification (5.13).
(e) Apply the topological simplification (5.12).
(f) Factorize on $\begin{bmatrix}0\\0\end{bmatrix}$ columns according to (5.35) and (8.50).
(g) Remove entries of -1 by using momentum conservation as described in section 7.
(h) Sort trihedral MGFs into their canonical representation as described in section 5.1.
(i) Apply basis decompositions discussed in section 9, in the basis listed in table 3.

Within this process, the steps (c) to (h) are repeated until the result no longer changes, before step (i) is executed.

OPTIONS

Option	Possible values	Default value	Description
`momSimplify`	`True`, `False`	`True`	Deactivates step (g)
`useIds`	`True`, `False`	`True`	Deactivates step (i)
`triHSR`	`True`, `False`	`True`	Deactivates steps (a) and (b)
`tri2ptHSR`	`True`, `False`	`True`	Deactivates step (a)
`tri3ptHSR`	`True`, `False`	`True`	Deactivates step (b)
`tri3ptFayHSR`	`True`, `False`	`False`	Activates three-point HSR via the Fay identity (6.22) instead of the forumla in [19]
`divHSR`	`True`, `False`	`True`	Deactivates steps (a) and (b) for divergent graphs
`triDivHSR`	`True`, `False`	`True`	Deactivates steps (a) and (b) for divergent graphs

Both options `divHSR` and `triDivHSR` have to be `True` for divergent graphs to be included

in steps (a) and (b). Furthermore, if `tri3ptFayHSR` is set to `True`, setting `tri2ptHSR` to `False` also deactivates three-point HSR since (6.22) reduces three-point HSR to two-point HSR.

WARNING

- If a graph in the argument contains a $\begin{bmatrix}0\\0\end{bmatrix}$ column next to a $\begin{bmatrix}1\\0\end{bmatrix}$ or $\begin{bmatrix}1\\0\end{bmatrix}$ column, the warning `TriCSimplify::dangerousFact` is issued and the modified factorization rule (8.50) applied.
- If a divergent graph with a holomorphic subgraph is encountered but HSR cannot be performed because either one of the options `divHSR` or `triDivHSR` is set to `False`, the warning `TriCSimplify::divHSRNotPossible` is issued.
- If three-point HSR is performed on a divergent graph using Fay identities by setting the option `tri3ptFayHSR` to `True`, the warning `TriCSimplify::div3ptFay` is issued.
- If three-point HSR is performed via the formula in [19] and there is no ordering of the blocks which prevents a divergent expression in the result (cf discussion in section 4.2.4 of the reference), the warning `TriCSimplify::noConvHSROrder` is issued. If one of the options `divHSR` or `triDivHSR` is set to `False`, the warning `TriCSimplify::divHSRNotPossible` is issued and the HSR is not performed.

EXAMPLES

```
In[96]:= TriCSimplify[c[{1,0},{1,1},{1,0}]]
```

$$\text{Out[96]}= -6\,C\left[\begin{smallmatrix}2&4\\2&0\end{smallmatrix}\right]+2\,C\left[\begin{smallmatrix}3&1&2\\1&1&0\end{smallmatrix}\right]-6\,C\left[\begin{smallmatrix}4&1&1\\1&0&1\end{smallmatrix}\right]+C\left[\begin{smallmatrix}1\\1\end{smallmatrix}\right]^2 G_4 + 2\,C\left[\begin{smallmatrix}2&1&1\\1&0&1\end{smallmatrix}\right]\hat{G}_2+\frac{2\pi\,C\left[\begin{smallmatrix}2&3\\2&-1\end{smallmatrix}\right]}{\tau_2}+\frac{2\pi\,C\left[\begin{smallmatrix}3&1&1\\0&0&1\end{smallmatrix}\right]}{\tau_2}$$

```
In[97]:= TriCSimplify[c[{1,0},{1,1},{1,0}], tri3ptFayHSR→True]
```

$$\text{Out[97]}= C\left[\begin{smallmatrix}1\\1\end{smallmatrix}\right]C\left[\begin{smallmatrix}1&1\\0&1\end{smallmatrix}\right]-C\left[\begin{smallmatrix}2&1&3\\1&1&0\end{smallmatrix}\right]-C\left[\begin{smallmatrix}2&1&3\\2&0&0\end{smallmatrix}\right]+3\,C\left[\begin{smallmatrix}4&1&1\\1&0&1\end{smallmatrix}\right]-C\left[\begin{smallmatrix}2&1&1\\1&0&1\end{smallmatrix}\right]\hat{G}_2-\frac{\pi\,C\left[\begin{smallmatrix}3&1&1\\0&0&1\end{smallmatrix}\right]}{\tau_2}$$

```
In[98]:= DiCSimplify[Out[96]-Out[97]]
```

$$\text{Out[98]}= 0$$

```
In[99]:= TriCSimplify[c[{1,0},{2,0},{1,0}], tri2ptHSR→False, tri3ptFayHSR→True]
```

$$\text{Out[99]}= C\left[\begin{smallmatrix}1&|&1&2&|&1&2\\0&|&0&0&|&1&0\end{smallmatrix}\right]$$

A.3.4. Four-point simplification.

`TetCSimplify`

The function `TetCSimplify` applies topological simplifications on four-point graphs.

ARGUMENTS `TetCSimplify` accepts one arbitrary argument.

RETURN VALUE `TetCSimplify` returns the expression given as the argument with all four-point MGFs rewritten in a simplified form, if possible. This is done by performing the following manipulations on all four-point graphs (not only tetrahedral ones), until the result does not change any more.

(a) Set graphs with odd total modular weight $a + b$ to zero.
(b) Apply the topological simplification (5.14).
(c) Apply the topological simplifications (5.15) and (5.16).
(d) Apply the topological simplifications (5.17) and (5.18).
(e) Set four-point MGFs to zero which vanish by symmetry, cf (5.5).
(f) Sort four-point MGFs into their canonical representation as described in section 5.1.

EXAMPLES

```
In[100]:= TetCSimplify[c[{1,1},{1,1},{1,2},{2,1},{2,1},{1,2}]]

Out[100]= 0

In[101]:= TetCSimplify[c[{},{1,1},{1,1},{},{1,1}]]

Out[101]= C[{1,1}]C[{1,1},{1,1},{1,1}]
```

A.3.5. Koba–Nielsen integration.

`zIntegrate`

The function `zIntegrate` expands Koba–Nielsen integrals in terms of MGFs.

ARGUMENTS `zIntegrate` represents a Koba–Nielsen integral and accepts three arguments.

The first argument should be a polynomial in the objects with suffix `z` introduced in section A.2.4, specifying the prefactor of the Koba–Nielsen factor. The second argument should be a natural number specifying the number of punctures in the Koba–Nielsen factor or a list of pairs of natural numbers `{{i,j}, {k,l}, ...}`, specifying the Green functions (and associated Mandelstam variables) appearing in the Koba–Nielsen factor.

The third argument should be a natural number specifying the order to which the Koba–Nielsen integral is to be expanded.

RETURN VALUE `zIntegrate` returns the order specified by the last argument of the Koba–Nielsen integral defined by the first two arguments. The resulting MGFs are simplified using the general properties listed below (2.36) and all the techniques implemented in `CSimplify`, apart from HSR and the application of the basis decompositions from section 9. If the resulting MGFs require graphs with more than four vertices, for which no notation was defined, a graphical representation of those graphs is printed, cf `Out[7]`. No constraints are placed on the Mandelstam variables.

EXAMPLES

```
In[102]:= zIntegrate[vz[2, {1, 2}] + vz[2, {3, 4}], 4, 1] // Simplify
          CSimplify[%]
```

$$\text{Out[102]=} \quad -\frac{\left(2\,\mathrm{C}\left[\begin{smallmatrix}3 & 0\\ 1 & 0\end{smallmatrix}\right] + \mathrm{C}\left[\begin{smallmatrix}1 & 1 & 1\\ 0 & 0 & 1\end{smallmatrix}\right]\right)(\mathrm{s}_{1,2} + \mathrm{s}_{3,4})\tau_2}{\pi}$$

$$\text{Out[103]=} \quad -\hat{\mathrm{G}}_2\,\mathrm{s}_{1,2} - \hat{\mathrm{G}}_2\,\mathrm{s}_{3,4}$$

```
In[104]:= zIntegrate[vz[2, {1, 2}] vz[2, {3, 4}], 4, 1] // Simplify
          CSimplify[%]
```

$$\text{Out[104]=} \quad -\frac{\left(2\,\mathrm{C}\left[\begin{smallmatrix}3 & 0\\ 1 & 0\end{smallmatrix}\right] + \mathrm{C}\left[\begin{smallmatrix}1 & 1 & 1\\ 0 & 0 & 1\end{smallmatrix}\right]\right)\hat{\mathrm{G}}_2\,(\mathrm{s}_{1,2} + \mathrm{s}_{3,4})\tau_2}{\pi}$$

$$\text{Out[105]=} \quad -\hat{\mathrm{G}}_2^2\,\mathrm{s}_{1,2} - \hat{\mathrm{G}}_2^2\,\mathrm{s}_{3,4}$$

```
In[106]:= zIntegrate[fz[1, 1, 2] fBarz[1, 1, 3], 3, 2] // Simplify
          CSimplify[%]//Simplify
```

$$\text{Out[106]=} \quad \frac{\mathrm{s}_{2,3}\left(-2\,\mathrm{C}\left[\begin{smallmatrix}1 & 1 & 1\\ 0 & 1 & 2\end{smallmatrix}\right]\mathrm{s}_{1,2} - 2\,\mathrm{C}\left[\begin{smallmatrix}0 & 1 & 2\\ 1 & 1 & 1\end{smallmatrix}\right]\mathrm{s}_{1,3} + \mathrm{C}\left[\begin{smallmatrix}1 & 1 & 1\\ 1 & 1 & 1\end{smallmatrix}\right]\mathrm{s}_{2,3}\right)\tau_2^2}{2\,\pi^2}$$

$$\text{Out[107]=} \quad \frac{\pi\ \mathrm{s}_{2,3}\ (\mathrm{s}_{1,2} + \mathrm{s}_{1,3} + \mathrm{s}_{2,3})\ (\mathrm{E}_3 + \zeta_3)}{2\ \tau_2}$$

A.4. Example: four-gluon scattering in the heterotic string

In this section, we use the functions introduced above to reproduce the expansions for the integrals $\mathcal{I}^{(2,0)}_{1234}$ and $\mathcal{I}^{(4,0)}_{1234}$ defined by

$$\mathcal{I}^{(4,0)}_{1234}(s_{ij},\tau) = \int \mathrm{d}\mu_3\, V_4(1,2,3,4)\,\mathrm{KN}_4 \tag{A.1}$$

$$\mathcal{I}^{(2,0)}_{1234}(s_{ij},\tau) = \int \mathrm{d}\mu_3\, V_2(1,2,3,4)\,\mathrm{KN}_4, \tag{A.2}$$

which appear in the planar sector of four-gluon scattering in the heterotic string, cf section 2.4 of [20].

All of the steps in the calculation are automatized, with one exception: the four-point HSR-identity (6.19) has to be added by hand. To this end, we first define the replacement rule

```
In[108]:= tetrule = c[1,0,1,1,0,1,0,1,1] → -c[1,0,{},1,1,2,0,1,0,1,1] - c[1,0,{},1,1,2,0,1,0,1]
          - c[1,0,2,0,1,{},1,0,1,1] + c[1,0,{},1,1,1,0,1,0,1,1] - c[1,0,1,0,1,1,1,0,{},1,0,1,1];
```

In order to bring the output into a nice form, we furthermore define the helper function

```
In[109]:= prettify[poly_]:=Block[{ap,mandOrd,result},
            mandOrd=MonomialList[poly,{s[1,2],s[2,3]}]/.List[x__]:>Plus[x];
            result=DeleteCases[DeleteDuplicates[Flatten[CoefficientList[mandOrd,
               {s[1,2],s[2,3]}]]],0];
            result=Collect[mandOrd,result];
            result=(SortBy[({Exponent[#/.s[i_,j_]:>ap s[i,j],ap],#}&)
               /@(List@@result),First][[All,2]])/.List[x__]:>HoldForm[Plus[x]];
            Return[result]];
```

The integral $\mathcal{I}^{(4,0)}_{1234}$ can now be expanded to second order by running

```
In[110]:= Sum[zIntegrate[vz[4, {1, 2, 3, 4}], 4, i], {i, 0, 2}];
```

To this we apply the four-point HSR-rule from above, decompose all resulting MGFs into the basis from table 3 and change the basis to table 4,

```
In[111]:= %/. tetrule // CSimplify // CConvertToNablaE;
```

Since `zIntegrate` does not apply momentum conservation to the Mandelstam variables, we do this explicitly,

```
In[112]:= %/. {s[3, 4]→s[1, 2], s[1, 4]→s[2, 3], s[2, 4]→-s[1, 2]-s[2, 3],
              s[1, 3]→-s[1, 2]-s[2, 3]};
```

Finally, we apply the function `prettify` defined in `In[109]` to rearrange the output

```
In[113]:= prettify[%]
```

$$\begin{aligned}\text{Out[113]}= {} & G_4 + (s_{1,2}+s_{2,3})\left(-6\,G_4 - \frac{3\,\pi\,\hat{G}_2\,\nabla E_2}{\tau_2^2}\right) + (s_{1,2}^2+s_{2,3}^2)\left(2\,E_2\,G_4 + \frac{\pi^2\,\nabla^2 E_3}{6\,\tau_2^4} + \frac{2\,\pi\,\hat{G}_2\,\nabla E_3}{3\,\tau_2^2}\right) \\ & + s_{1,2}\,s_{2,3}\left(2\,E_2\,G_4 + \frac{2\,\pi^2\,\nabla^2 E_3}{3\,\tau_2^4} + \frac{8\,\pi\,\hat{G}_2\,\nabla E_3}{3\,\tau_2^2}\right)\,,\end{aligned}$$

which agrees with the result found in [20]. The Laurent polynomial of the first orders of $\mathcal{I}^{(4,0)}_{1234}$ can now easily be obtained by

```
In[114]:= prettify[CLaurentPoly[ReleaseHold[%]]]
```

$$\begin{aligned}\text{Out[114]}= {} & \frac{\pi^4}{45} + (s_{1,2}+s_{2,3})\left(-\frac{2\,\pi^4\,y}{45} - \frac{3\,\pi^4\,\zeta_3}{y^3} + \frac{\pi^4\,\zeta_3}{y^2}\right) + s_{1,2}s_{2,3}\left(\frac{94\,\pi^4\,y^2}{14175} + \frac{2\,\pi^4\,\zeta_3}{45\,y} + \frac{5\,\pi^4\,\zeta_5}{y^4} - \frac{\pi^4\,4\zeta_5}{3\,y^3}\right) \\ & + (s_{1,2}^2+s_{2,3}^2)\left(\frac{34\,\pi^4\,y^2}{14175} + \frac{2\,\pi^4\,\zeta_3}{45\,y} + \frac{5\,\pi^4\,\zeta_5}{y^4} - \frac{\pi^4\,\zeta_5}{3\,y^3}\right)\,.\end{aligned}$$

Similarly, we can expand $\mathcal{I}^{(2,0)}_{1234}$ to third order by running

```
In[115]:= Sum[zIntegrate[vz[2,{1,2,3,4}],4,i],{i,0,3}];

          %//CSimplify//CConvertToNablaE;

          %/.{s[3,4]→s[1,2],s[1,4]→s[2,3],s[2,4]→-s[1,2]-s[2,3],
              s[1,3]→-s[1,2]-s[2,3]};

          prettify[%]
```

$$\begin{aligned}\text{Out[115]}= &-\frac{3\,\pi\,\nabla\mathrm{E}_2\,(\mathrm{s}_{1,2}+\mathrm{s}_{2,3})}{\tau_2^2}+\frac{8\,\pi\,\nabla\mathrm{E}_3\,\mathrm{s}_{1,2}\,\mathrm{s}_{2,3}}{3\,\tau_2^2}+\frac{2\,\pi\,\nabla\mathrm{E}_3\,(\mathrm{s}_{1,2}^2+\mathrm{s}_{2,3}^2)}{3\,\tau_2^2}\\ &+\left(\mathrm{s}_{1,2}^2\mathrm{s}_{2,3}+\mathrm{s}_{1,2}\mathrm{s}_{2,3}^2\right)\left(-\frac{12\,\pi\,\mathrm{E}_2\,\nabla\mathrm{E}_2}{\tau_2^2}-\frac{8\,\pi\,\nabla\mathrm{E}_4}{5\,\tau_2^2}-\frac{24\,\pi\,\nabla\mathrm{E}_{2,2}}{\tau_2^2}\right)\\ &+\left(\mathrm{s}_{1,2}^3+\mathrm{s}_{2,3}^3\right)\left(-\frac{6\,\pi\,\mathrm{E}_2\,\nabla\mathrm{E}_2}{\tau_2^2}-\frac{4\,\pi\,\nabla\mathrm{E}_4}{5\,\tau_2^2}-\frac{12\,\pi\,\nabla\mathrm{E}_{2,2}}{\tau_2^2}\right),\end{aligned}$$

in agreement with the result in [20]. The next higher order in α' of $\mathcal{I}^{(2,0)}_{1234}$ contains two tetrahedral graphs. One of them vanishes by symmetry, the other one can be reduced to trihedral graphs by means of the Fay identity (6.17),

$$\mathcal{C}\left[\begin{matrix}1\\0\\\hline 1\\0\end{matrix}\middle\|\begin{matrix}1\\1\\\hline 1\\1\end{matrix}\middle\|\begin{matrix}1\\1\\\hline 1\\1\end{matrix}\right]=0 \tag{A.3}$$

$$\begin{aligned}\mathcal{C}\left[\begin{matrix}1\\0\\\hline 1\\1\end{matrix}\middle\|\begin{matrix}1\\0\\\hline 1\\1\end{matrix}\middle\|\begin{matrix}1\\1\\\hline 1\\1\end{matrix}\right]=&-\mathcal{C}\left[\begin{matrix}\varnothing\\\hline 1\\1\end{matrix}\middle\|\begin{matrix}\varnothing\\\hline 1\\1\end{matrix}\middle\|\begin{matrix}1\\1\\\hline 1\,2\\1\,0\end{matrix}\right]-\mathcal{C}\left[\begin{matrix}\varnothing\\\hline 1\\1\end{matrix}\middle\|\begin{matrix}2\\0\\\hline 1\\1\end{matrix}\middle\|\begin{matrix}1\\1\\\hline 1\\1\end{matrix}\right]-\mathcal{C}\left[\begin{matrix}2\\0\\\hline 1\\1\end{matrix}\middle\|\begin{matrix}\varnothing\\\hline 1\\1\end{matrix}\middle\|\begin{matrix}1\\1\\\hline 1\\1\end{matrix}\right]\\ &+\mathcal{C}\left[\begin{matrix}\varnothing\\\hline 1\\1\end{matrix}\middle\|\begin{matrix}1\\0\\\hline 1\\1\end{matrix}\middle\|\begin{matrix}1\\1\\\hline 1\,1\\0\,1\end{matrix}\right]-\mathcal{C}\left[\begin{matrix}1\\0\\\hline 1\\1\end{matrix}\middle\|\begin{matrix}\varnothing\\\hline 1\\1\end{matrix}\middle\|\begin{matrix}1\\1\\\hline 1\,1\\0\,1\end{matrix}\right].\end{aligned} \tag{A.4}$$

If the Fay identity (A.4) is added by hand, similarly to how (6.19) was added above, the expansion of $\mathcal{I}^{(2,0)}_{1234}$ can be extended to the order α'^4.

Appendix B. Kinematic poles in three-point Koba–Nielsen integrals

As explained in section 8.2, a factor $|f^{(1)}_{ij}|^2$ in a Koba–Nielsen integral leads to a naive expansion of this integral terms of divergent MGFs. This signals a pole in one or more of the Mandelstam variables which can be made explicit by means of integration-by-parts manipulations. In this appendix we discuss the resulting expressions for all three-point Koba–Nielsen integrals containing $|f^{(1)}_{ij}|^2$ factors using the notation

$$W^{\tau}_{(a_2,a_3|b_2,b_3)}(\sigma|\rho)=\int\frac{\mathrm{d}^2z_2}{\tau_2}\frac{\mathrm{d}^2z_3}{\tau_2}\,\rho[f^{(a_2)}_{12}f^{(a_3)}_{23}]\sigma[\overline{f^{(b_2)}_{12}}\,\overline{f^{(b_3)}_{23}}]\mathrm{KN}_3 \tag{B.1}$$

introduced in [29]. Here, the permutations $\rho,\sigma\in\mathcal{S}_2$ act on the subscripts $i,j\in\{2,3\}$ of $f^{(n)}$ and $\overline{f^{(n)}}$ but not on those of a_i and b_j.

If only one $|f^{(1)}_{ij}|^2$ is present in the integrand and the other $f^{(a)}$, $\overline{f^{(b)}}$ do not depend on z_i or z_j, we can use the puncture only occurring in $|f^{(1)}_{ij}|^2$ to integrate by parts, obtaining one more term compared to (8.18),

$$W^{\tau}_{(1,a|1,b)}(2,3|2,3) = (-)^{a+1}\frac{s_{13}}{s_{12}}W^{\tau}_{(1,a|1,b)}(2,3|3,2) - \frac{1}{s_{12}}\frac{\pi}{\tau_2}W^{\tau}_{(0,a|0,b)}(2,3|2,3), \tag{B.2}$$

with $a \neq 1$ or $b \neq 1$. Three more cases can be obtained from (B.2) by relabeling of the Mandelstam variables,

$$W^{\tau}_{(a,1|b,1)}(2,3|2,3) = W^{\tau}_{(1,a|1,b)}(2,3|2,3)\Big|_{s_{12}\leftrightarrow s_{23}} \tag{B.3}$$

$$W^{\tau}_{(1,a|1,b)}(3,2|3,2) = W^{\tau}_{(1,a|1,b)}(2,3|2,3)\Big|_{s_{12}\leftrightarrow s_{13}} \tag{B.4}$$

$$W^{\tau}_{(a,1|b,1)}(3,2|3,2) = W^{\tau}_{(1,a|1,b)}(2,3|2,3)\Big|^{s_{12}\to s_{23}}_{s_{23}\to s_{13}\,,\ s_{13}\to s_{12}} \tag{B.5}$$

where again $a \neq 1$ or $b \neq 1$.

If both punctures i and j of $|f^{(1)}_{ij}|^2$ also appear in other $f^{(a)}$, $\overline{f^{(b)}}$ factors, one obtains an additional term from the action of $\partial_{\bar{z}}$ on the corresponding $f^{(a)}$ according to (2.13). In this way, we obtain

$$W^{\tau}_{(a,1|b,1)}(3,2|2,3) = \left\{\frac{s_{23}}{s_{13}}W^{\tau}_{(1,a|b,1)}(2,3|3,2) + \frac{(-)^b}{s_{13}}\frac{\pi}{\tau_2}\Big[W^{\tau}_{(0,a|b,0)}(2,3|2,3) + (-)^{a-1}W^{\tau}_{(1,a-1|b,0)}(2,3|3,2)\Big]\right\}^{s_{12}\to s_{13}}_{s_{13}\to s_{23}\,,\ s_{23}\to s_{12}} \tag{B.6}$$

$$W^{\tau}_{(a,1|b,1)}(3,2|2,3) = \left\{\frac{s_{23}}{s_{12}}W^{\tau}_{(a,1|1,b)}(2,3|3,2) + \frac{1}{s_{12}}\frac{\pi}{\tau_2}\Big[W^{\tau}_{(a,0|0,b)}(2,3|3,2) - W^{\tau}_{(a,0|1,b-1)}(2,3|3,2)\Big]\right\}^{s_{12}\to s_{23}}_{s_{13}\to s_{12}\,,\ s_{23}\to s_{13}}, \tag{B.7}$$

where $a \neq 1$ in (B.6) and $b \neq 1$ in (B.7) and we set $\overline{f^{(-1)}} = 0$. With the help of the Mandelstam relabelings, we avoid the need of a Fay identity to write the rhs in terms of the integrals (B.1). One further case can be obtained by Mandelstam relabelings of (B.6) and (B.7),

$$W^{\tau}_{(a,1|b,1)}(2,3|3,2) = W^{\tau}_{(a,1|b,1)}(3,2|2,3)\Big|_{s_{12}\leftrightarrow s_{13}}, \tag{B.8}$$

where $a \neq 1$ or $b \neq 1$.

If two $|f^{(1)}_{ij}|^2$ factors are present in the integrand, we obtain (on top of the poles for each $|f^{(1)}_{ij}|^2$) a three-point kinematic pole $\sim \frac{1}{s_{123}}$, where the three-point Mandelstam variable is defined in (8.19). Hence, the integral

$$W^{\tau}_{(1,1|1,1)}(2,3|2,3) = \int \mathrm{d}\mu_2 \left|f^{(1)}_{12}\right|^2 \left|f^{(1)}_{23}\right|^2 \mathrm{KN}_3 \tag{B.9}$$

has pole structure $\frac{1}{s_{123}}\left(\frac{1}{s_{12}}+\frac{1}{s_{23}}\right)$. Similarly, the integral

$$W^{\tau}_{(1,1|1,1)}(3,2|2,3) = -\int \mathrm{d}\mu_2\, f^{(1)}_{12}\,\overline{f^{(1)}_{13}}\left|f^{(1)}_{23}\right|^2 \mathrm{KN}_3 \tag{B.10}$$

has pole structure $\frac{1}{s_{123}s_{23}}$. As discussed in appendix D of [29], these poles can be made explicit by the integration-by-parts manipulation

$$\begin{aligned} W^{\tau}_{(1,1|1,1)}(2,3|2,3) = &-\frac{s_{13}}{s_{123}}\left[W^{\tau}_{(2,0|1,1)}(2,3|2,3)+W^{\tau}_{(0,2|1,1)}(2,3|2,3)+W^{\tau}_{(2,0|1,1)}(2,3|3,2)\right] \\ &-\frac{1}{s_{123}}\frac{\pi}{\tau_2}\left[W^{\tau}_{(1,0|1,0)}(2,3|2,3)+W^{\tau}_{(0,1|0,1)}(2,3|2,3)\right] \end{aligned} \tag{B.11}$$

$$\begin{aligned} W^{\tau}_{(1,1|1,1)}(3,2|2,3) = &-\frac{s_{13}}{s_{123}}\left[W^{\tau}_{(2,0|1,1)}(3,2|2,3)+W^{\tau}_{(0,2|1,1)}(3,2|2,3)+W^{\tau}_{(2,0|1,1)}(3,2|3,2)\right] \\ &+\frac{1}{s_{123}}\frac{\pi}{\tau_2}\left[W^{\tau}_{(1,0|1,0)}(3,2|2,3)+W^{\tau}_{(1,0|0,1)}(2,3|2,3)\right. \\ &\left.+W^{\tau}_{(0,1|0,1)}(2,3|2,3)\right], \end{aligned} \tag{B.12}$$

where the formulas above can be used to manifest the two-particle poles on the rhs. The permutations of (B.9) and (B.10) can again be obtained by relabeling the Mandelstam variables,

$$W^{\tau}_{(1,1|1,1)}(2,3|3,2) = W^{\tau}_{(1,1|1,1)}(3,2|2,3)\Big|_{s_{12}\leftrightarrow s_{13}} \tag{B.13}$$

$$W^{\tau}_{(1,1|1,1)}(3,2|3,2) = W^{\tau}_{(1,1|1,1)}(2,3|2,3)\Big|_{s_{12}\leftrightarrow s_{13}}. \tag{B.14}$$

ORCID iDs

Jan E Gerken https://orcid.org/0000-0002-0172-7944

References

[1] Green M B and Vanhove P 2000 The low energy expansion of the one-loop type II superstring amplitude *Phys. Rev.* D **61** 104011
[2] Green M B, Russo J G and Vanhove P 2008 Low energy expansion of the four-particle genus-one amplitude in type II superstring theory *J. High Energy Phys.* JHEP02(2008)020
[3] D'Hoker E, Green M B and Vanhove P 2015 On the modular structure of the genusone type II superstring low energy expansion *J. High Energy Phys.* JHEP08(2015)041
[4] D'Hoker E, Green M B and Vanhove P 2019 Proof of a modular relation between 1-, 2- and 3-loop Feynman diagrams on a torus *J. Number Theory* **196** 381
[5] Basu A 2016 Poisson equation for the Mercedes diagram in string theory at genus one *Class. Quantum Grav.* **33** 055005
[6] D'Hoker E, Green M B, Gürdoğan Ö and Vanhove P 2017 Modular graph functions *Commun. Number Theory Phys.* **11** 165
[7] Zerbini F 2016 Single-valued multiple zeta values in genus 1 superstring amplitudes *Commun. Number Theory Phys.* **10** 703
[8] D'Hoker E and Green M B 2018 Identities between modular graph forms *J. Number Theory* **189** 25
[9] Basu A 2016 Proving relations between modular graph functions *Class. Quantum Grav.* **33** 235011

[10] Basu A 2017 Simplifying the one-loop five graviton amplitude in type IIB string theory *Int. J. Mod. Phys.* A **32** 1750074
[11] D'Hoker E and Kaidi J 2016 Hierarchy of modular graph identities *J. High Energy Phys.* JHEP11(2016)051
[12] Kleinschmidt A and Verschinin V 2017 Tetrahedral modular graph functions *J. High Energy Phys.* JHEP09(2017)155
[13] Brown F 2018 A class of non-holomorphic modular forms I *Res. Math. Sci.* **5** 40
[14] Brown F 2017 A class of non-holomorphic modular forms II: equivariant iterated eisenstein integrals (arXiv:1708.03354)
[15] D'Hoker E and Duke W 2018 Fourier series of modular graph functions *J. Number Theory* **192** 1
[16] Basu A 2017 Low momentum expansion of one loop amplitudes in heterotic string theory *J. High Energy Phys.* JHEP11(2017)139
[17] Brown F 2018 A class of non-holomorphic modular forms III: real analytic cusp forms for SL2(Z) *Res. Math. Sci.* **5** 36
[18] Basu A 2018 A simplifying feature of the heterotic one loop four graviton amplitude *Phys. Lett.* B **776** 182
[19] Gerken J E and Kaidi J 2019 Holomorphic subgraph reduction of higher-point modular graph forms *J. High Energy Phys.* JHEP01(2019)131
[20] Gerken J E, Kleinschmidt A and Schlotterer O 2019 Heterotic-string amplitudes at one loop: modular graph forms and relations to open strings *J. High Energy Phys.* JHEP01(2019)052
[21] D'Hoker E and Kaidi J 2019 Modular graph functions and odd cuspidal functions—Fourier and Poincaré series *J. High Energy Phys.* JHEP04(2019)136
[22] Dorigoni D and Kleinschmidt A 2019 Modular graph functions and asymptotic expansions of Poincaré series *Commun. Number Theory Phys.* **13** 569
[23] D'Hoker E and Green M B 2020 Absence of irreducible multiple zeta-values in melon modular graph functions *Commun. Number Theory Phys.* **14** 315
[24] D'Hoker E 2019 Integral of two-loop modular graph functions; *J. High Energy Phys.* JHEP06(2019)092
[25] Basu A 2019 Eigenvalue equation for the modular graph $C_{a,b,c,d}$ *J. High Energy Phys.* JHEP07(2019)126
[26] Zagier D and Zerbini F 2020 Genus-zero and genus-one string amplitudes and special multiple zeta values *Commun. Number Theory Phys.* **14** 413
[27] Berg M, Bringmann K and Gannon T 2019 Massive deformations of Maass forms and Jacobi forms (arXiv:1910.02745)
[28] Hohenegger S 2020 From little string free energies towards modular graph functions *J. High Energy Phys.* JHEP03(2020)077
[29] Gerken J E, Kleinschmidt A and Schlotterer O 2020 All-order differential equations for one-loop closed-string integrals and modular graph forms *J. High Energy Phys.* JHEP01(2020)064
[30] Gerken J E, Kleinschmidt A and Schlotterer O 2020 Generating series of all modular graph forms from iterated Eisenstein integrals (arXiv:2004.05156)
[31] Basu A 2020 Zero mode of the Fourier series of some modular graphs from Poincare series (arXiv:2005.07793)
[32] D'Hoker E and Green M B 2014 Zhang–Kawazumi invariants and superstring amplitudes *J. Number Theory* **144** 111
[33] D'Hoker E, Green M B, Pioline B and Russo R 2015 Matching the D_6R_4 interaction at two-loops *J. High Energy Phys.* JHEP01(2015)031
[34] D'Hoker E, Green M B and Pioline B 2019 Higher genus modular graph functions, string invariants, and their exact asymptotics *Commun. Math. Phys.* **366** 927
[35] D'Hoker E, Green M B and Pioline B 2019 Asymptotics of the D_8R_4 genus-two string invariant *Commun. Number Theory Phys.* **13** 351
[36] Basu A 2019 Eigenvalue equation for genus two modular graphs *J. High Energy Phys.* JHEP02(2019)046
[37] Zagier D Notes on lattice sums (unpublished)
[38] Green M B, Schwarz J H and Witten E 1988 *Superstring Theory: Loop Amplitudes, Anomalies and Phenomenology* (Cambridge: Cambridge University Press) p 608
[39] Kronecker L 1881 Zur Theorie der elliptischen Funktionen *Mathematische Werke* **IV** 313
[40] Brown F C S and Levin A 2011 Multiple elliptic polylogarithms (arXiv:1110.6917)
[41] Dolan L and Goddard P 2009 Current algebra on the torus *Commun. Math. Phys.* **285** 219

[42] Broedel J, Mafra C R, Matthes N and Schlotterer O 2015 Elliptic multiple zeta values and one-loop superstring amplitudes *J. High Energy Phys.* JHEP07(2015)112
[43] Zagier D 1990 The Bloch–Wigner–Ramakrishnan polylogarithm function *Math. Ann.* **286** 613
[44] Broedel J and Kaderli A 2020 Functional relations for elliptic polylogarithms *J. Phys. A: Math. Theor.* **53** 245201
[45] Maass H 1983 *Lectures on Modular Functions of One Complex Variable* 2nd edn (*Lectures on Mathematics and Physics* vol 29) (Berlin: Springer) p 242
[46] Fay J D 1973 *Theta Functions on Riemann Surfaces* (*Lecture Notes in Mathematics*) (Berlin: Springer)
[47] Broedel J, Schlotterer O and Zerbini F 2019 From elliptic multiple zeta values to modular graph functions: open and closed strings at one loop *J. High Energy Phys.* JHEP01(2019)155
[48] Fleig P, Gustafsson H P A, Kleinschmidt A and Persson D 2018 *Eisenstein Series and Automorphic Representations: With Applications in String Theory* (Cambridge: Cambridge University Press) p 587