LU-TP 22-17
March 2022

# A numerical implementation of the chirality-flow formalism

**Zenny Wettersten**

Department of Astronomy and Theoretical Physics, Lund University

Master thesis supervised by Andrew Lifson and Malin Sjödahl


LUND
UNIVERSITY

# Abstract

The chirality-flow formalism has proven to make amplitude calculations in the Standard Model analytically simple in comparison to previously established formalisms. To test the viability of the chirality-flow formalism for numerical implementations, a Chiral Automized Flow Extension (`CAFE`) for MadGraph5_aMC@NLO has been developed to compare amplitude evaluation time scaling between the two. `CAFE` uses a new `UFO` format model and a new `HELAS`-like library to make evaluations in the chirality-flow formalism. Additionally, an algorithm has been implemented to make a simple gauge choice, based on which diagrams that will not contribute to the matrix elements will be discarded already at compile time. The effects of these two changes on evaluation time scaling with process complexity have been studied, and `CAFE` has been shown to both be faster for all considered processes and to scale better with diagram complexity than MadGraph5.

# Populärvetenskaplig beskrivning

Få blir nog förvånade om någon hävdar att beräkningar inom partikelfysik är invecklade. Bara för att beräkna sannolikheten att den enklaste processen vi tar hänsyn till (vilket är när två elektroner krockar och studsar mot varandra) krävs det åtminstone tre-fyra sidor med ren matematik, och svårigheterna växer snabbt ju mer komplexa processer vi arbetar med. För att kunna ha teoretiska värden att jämföra våra experimentella mätningar med måste dessa sannolikheter dock beräknas; utan dem är det omöjligt att säga när vi väl upptäcker fysik bortom våra nuvarande teorier.

En av de huvudsakliga orsakerna till problematiken är att den matematik vi använder för att beskriva universum är fyrdimensionell: Vi har tre rumsdimensioner (upp-ned, höger-vänster, framåt-bakåt) och en tidsdimension (framtid-förflutet). Fyrdimensionell matte är mycket krånglingare än den medelpersonen är van vid, då det inte längre räcker att ta hänsyn till hur ett tal relaterar till ett annat: Det räcker inte att säga att $1 + 1 = 2$. I fyrdimensionell matte måste det tas hänsyn till hur höger-vänster påverkar upp-ned, hur framåt-bakåt påverkar framtid-förflutet, och vice versa för alla kombinationer mellan dem: Kort och gott måste hur fyra olika saker relaterar till varandra beräknas, vilket blir $4 \times 4 = 16$ gånger mer komplicerat än vad vanlig endimensionell matte är. Vi måste nu betrakta om ettan vi tittar på ligger längs höger-vänster eller uppåt-nedåt innan vi kan säga om 1 plus 1 är så enkelt som 2, eller om det är en etta åt höger som är helt skild från en etta som riktar bakåt.

För att förenkla beräkningarna har fysiker sedan 80-talet arbetat med att beskriva sakerna vi beräknar i lägre-dimensionell matematik. Genom att dela in det fyrdimensionella rum-

met vi lever i i två stycken tvådimensionella rum som beter sig ungefär likadant förenklades beräkningar med den så kallade spinor-helicitetsformalismen, som har blivit stor de senaste årtiondena. På senare år har en vidareutveckling av spinor-helicitetsformalismen presenterats, där de tvådimensionella objekten representeras grafiskt som olika sorters linjer. Denna formalism kallas för chiralitetsflödesformalismen, och med dess hjälp kan sannolikheter beräknas med vardaglig endimensionell matte, direkt från ett diagram. Exemplet vi diskuterade tidigare, elektroner som snuddar vid varandra, kan med chiralitetsflöde beräknas med en halv sidas uträkningar, om ens det.

Numera gör fysiker få beräkningar med penna och papper: Istället görs de med datorer. Hur mycket enklare det än är för en människa att beräkna med chiralitetsflöde jämfört med tidigare beskrivningar har ingen större nytta om det inte också kan visas att chiralitetsflöde även förenklar beräkningarna för datorer. Eftersom datorer tänker på matematiska, så att säga, är det som människor finner svårt inte nödvändigtvis det för datorer.

Det är av denna anledning vi har utvecklat ett program, `CAFE`, som använder sig av just chiralitetsflöde för att göra beräkningar. Programmet i fråga är baserat på MadGraph5, ett standardverktyg som används för just den här typen av beräkningar vid t.ex. CERN. Vad vi gjort är att modifiera MadGraph5 för att göra exakt samma sak som det gör för den fyrdimensionella beskrivningen av partikelfysik, fast byta ut alla beräkningar i fyrdimensionell matte mot våra chiralitetsflödesekvationer. Utöver det har vi lagt till en smart algoritm som använder chiralitetsflöde för att förutsäga att vissa av sannolikheterna vi ska beräkna är noll utan att faktiskt behöva göra själva beräkningarna, till skillnad från MadGraph5 som beräknar dessa nollor explicit.

Precis som vi hoppats visar det sig att `CAFE` är snabbare än MadGraph5. Inte nog med att `CAFE` är snabbare, utan den blir ännu snabbare i jämförelse med MadGraph5 när processerna den beräknar sannolikheter för blir mer komplicerade. Vad vi uppmätt är hur lång tid det tar att beräkna sannolikheten för en elektron och en antielektron att slås ihop och bli ett antal $n$ fotoner, där vi låtit $n$ öka. För två fotoner är `CAFE` ungefär dubbelt så snabbt som MadGraph5; för 7 fotoner är `CAFE` mer än tio gånger så snabb.

I dess nuvarande form kan `CAFE` enbart göra beräkningar för de enklaste sortens processer vi har i partikelfysikens standardmodell, men det här resultatet tyder på att fördelarna med chiralitetsflöde kommer väl till pass även för datorerna som talar matematiska, och inte bara för våra mänskliga hjärnor.

# Contents

# 1 Introduction

Given the 4-dimensional nature of Minkowski spacetime, the initial description of the SM (Standard Model) as a QFT (quantum field theory) using 4-dimensional vector spaces is unsurprising. However, calculations within the SM and BSM (beyond the Standard Model) theories quickly grow complicated. Calculating the matrix element squared for the spin-averaged minimal QED (quantum electrodynamics) process of electron annihilation followed by a muon pair production can take pages of algebra. As such, these calculations are left to computers these days.

However, regardless of the possible understanding gained from simplifying these calculations, it is also possible that such simplifications could yield higher efficiencies in software used for calculations of e.g. the amplitudes or cross sections of interactions. Given the processing power needed for more intricate interactions, any such developments are welcome.

One such development is the spinor-helicity formalism [1–4]. Using the fact that $SL(2,\mathbb{C})$ (the special linear group of degree 2 over the complex numbers) is a double cover of the restricted Lorentz group $SO(1,3)$ [5], the latter can be split into a direct product of two 2-dimensional spaces, which we will call the left- and right-chiral halves of the Lorentz group. For the case of massless fermions, these representations will not mix, leading to major simplifications in calculations that reduce the previous 4-dimensional calculations regarding the Lorentz algebra to 2-dimensional ones.

Recently, further developments into the chiral nature of spacetime have been made with the chirality-flow formalism [6,7]. This is a graphical formalism for the SM which simplifies calculations even further, allowing for immediate evaluation of Feynman diagrams. The chirality-flow formalism has proven practical for pen and paper calculations, but it has previously not been implemented numerically. The hope is that an increased efficiency will be found also here, particularly for massless models (e.g. high-energy QCD (quantum chromodynamics) interactions such as in proton-proton collisions) with assigned chiralities.

To test this, a numerical implementation using chirality flow must be made. We use the MadGraph5_aMC@NLO software suite[1] [8] as a starting point, repurposing it to work with chirality flow. MadGraph5_aMC@NLO is a framework for LO (leading order) and NLO (next to leading order) studies in SM and BSM phenomenology, that provides e.g. cross section calculations, hard event generation, and event manipulation. However, since MadGraph5 works using a Dirac spinor representation, we modify the framework provided to instead work within the chirality-flow formalism at all stages of diagram generation and evaluation. The result is `CAFE`, a Chiral Automized Flow Extension for MadGraph5.

While MadGraph5 does not natively support the types of structures we work with in the chirality-flow formalism, it is difficult to outperform software that has been optimized since 1994 [9]. For this reason, we adapt their methods to analyze the possible runtime

[1]The version used for this project is MadGraph5_aMC@NLO 3.2.0, released on August 22nd, 2021.

scaling of chirality flow, rather than trying to reinvent the wheel. Although this may be disadvantageous to CAFE, it makes a first implementation of chirality flow simpler than needing to build chirality-flow diagram generation and function libraries from scratch. It also has the benefit of making comparisons to current standard methods simple. Thus, CAFE should be seen as a first endeavor into the possibility of further numerical implementations of chirality flow, rather than the end-all-be-all. Specifically, for this project we have implemented massless QED as described by the chirality-flow formalism, and compare the scaling with process complexity found using CAFE to that of MadGraph5.

CAFE presents two central avenues of efficiency gain as compared to standard MadGraph5. The obvious one lies in the simplified calculations performed. The second one dives deeper into the benefits of chirality flow, with an algorithm detecting a simple gauge vector choice that allows a diagram removal routine within CAFE to discard many non-contributing diagrams already at the diagram generation stage (compile time).

The first few parts of this thesis introduce the background to the project itself, with the latter parts describing the work undertaken and its results. Section 2 describes the theory of the chirality-flow formalism and its building blocks in the underlying representation theory of Lie groups, and Section 3 gives an introduction to the MadGraph5_aMC@NLO software suite, as well as to the HELAS (HELicity Amplitude Subroutines) libraries [10] and the UFO (Universal FeynRules Output) format [11] for numerical calculations of amplitudes. Finally, Section 4 goes over the details of CAFE, with Section 5 detailing how it compares with standard MadGraph5 for calculations in massless QED.

# 2 Chirality flow

## 2.1 Group theory and Lorentz transformations

The Lorentz group, $O(1,3)$, is the set of all $4 \times 4$-matrices that preserve the Minkowski metric — the set of all Lorentz transformations. Disregarding transformations that do not preserve the orientation of space gives us the restricted Lorentz group, the special pseudo-orthogonal group $SO(1,3)$. This is the set of elements $M \in O(1,3)$ with determinant $\det(M) = 1$, and it is continuous, meaning there exist curves that connect all elements $\Lambda \in SO(1,3)$ to identity in $SO(1,3)$.

The restricted Lorentz group is a Lie group [12], which gives it some interesting properties. Most notably, to all Lie groups we can associate a vector space called the Lie algebra corresponding to that Lie group. The Lie algebra of the group is the tangent space of the group at the unit element. Elements of the Lie algebra are then, ineloquently put, the infinitesimal basic group elements in the vicinity of unity. By taking the exponential map acting on linear combinations of the elements of the Lie algebra we will be able to locally describe the action of the group.

For a group $G$, its Lie algebra is given by the linearization of $G$, i.e. the vector space given by its tangent space. Given a group $G$, we denote its Lie algebra by $\mathfrak{g}$. Then, for any vector $X \in \mathfrak{g}$ we have that

$$\exp(X) \in G, \tag{2.1}$$

where we use the typical power series definition of the matrix exponential. By acting with the exponential map on vectors of the Lie algebra $\mathfrak{g}$, we acquire elements of the group $G$.

For $SO(1,3)$, we can describe it fully using six fundamental operations: Rotations and boosts about the three coordinate axes. Thus, we will acquire generators of the group by taking the derivatives of these elementary operations at identity. We skip the details here, but one basis for the Lie algebra of $SO(1,3)$ can be shown to be given by the matrices

$$A_x = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \quad A_y = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{pmatrix}, \quad A_z = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \tag{2.2}$$

$$K_x = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad K_y = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad K_z = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}, \tag{2.3}$$

where $A_i$ and $K_j$ are the generators of rotations about the $i$- and boosts along the $j$-axis respectively, with $i, j \in \{x, y, z\}$. This parametrization gives us all elements $\Lambda \in SO(1,3)$, i.e. for all $\Lambda \in SO(1,3)$ there exist $\vec{\theta} \in \mathbb{T}^3, \vec{\eta} \in \mathbb{R}^3$, where $\mathbb{T}$ is the circle group, such that

$$\Lambda = \exp\left[\vec{\theta} \cdot \vec{A} + \vec{\eta} \cdot \vec{K}\right] = \exp\left[-i\left(\vec{\theta} \cdot \vec{L} + \vec{\eta} \cdot \vec{K'}\right)\right], \tag{2.4}$$

where $\vec{\theta}$ and $\vec{\eta}$ are the 3-vectors of angles of (sinusoidal and hyperbolic) rotations, corresponding to each spatial axis, and $\vec{A}$ and $\vec{K}$ are the 3-vectors of generators corresponding to each spatial axis. In the second equality, we switch to working with Hermitian generators $L_j = iA_j$ and $K'_j = iK_j$ as is typical for physicists.

To determine the Lie algebra it is insufficient to know the generators of the group — we also need to know how they relate to each other. The algebra $\mathfrak{g}$ must be equipped with a so-called Lie bracket, which is a vector product that satisfies the Jacobi identity [12]. For matrix groups, this becomes the matrix commutator

$$[X, Y] = XY - YX. \tag{2.5}$$

In the case of the restricted Lorentz group $SO(1,3)$ and our previous generators, the algebra is given by

$$[L_j, L_k] = i\epsilon_{jkl}L_l, \quad [L_j, K'_k] = i\epsilon_{jkl}K'_l, \quad [K'_j, K'_k] = -i\epsilon_{jkl}L_l, \tag{2.6}$$

with $\epsilon_{jkl}$ the Levi-Civita symbol which is zero if any of its indices repeat, equals 1 if the indices form an even permutation, and equals -1 if they form an odd permutation. Here, we implicitly sum over the repeated index $l$.

We notice that the two sets of generators do not commute with each other. The rotation generators, which are the generators of the subgroup of 3-rotations given by $SO(3)$, close under the Lie bracket, whereas the boost generators do not. This corresponds to boosts not forming a subgroup of $SO(1,3)$: An arbitrary boost followed by another arbitrary boost is generally not a boost; it is a boost followed by a rotation.

However, any basis of the algebra should form a set of generators for the corresponding group, and since we have seen that all elements of $SO(1,3)$ can be generated from the algebra we may choose our basis as we see fit. Let us study the complex linear combinations

$$N_j^{\pm} = \frac{L_j \pm iK'_j}{2}, \tag{2.7}$$

which yield the commutation relations

$$[N_j^+, N_k^+] = i\epsilon_{jkl}N_l^+, \quad [N_j^-, N_k^-] = i\epsilon_{jkl}N_l^-, \quad [N_j^+, N_k^-] = 0, \tag{2.8}$$

meaning we now get two sets of three generators that close under commutation.

The commutation relations above are familiar to a group theorist. If we consider the special unitary group $SU(2)$ (the set of complex $2 \times 2$-matrices with determinant 1) we find that its three generators $J_j$ have the commutation relation

$$[J_j, J_k] = i\epsilon_{jkl}J_l. \tag{2.9}$$

However, since our change of basis for the algebra $\mathfrak{so}(1,3)$ involves complex coefficients, we must consider the complexification of the algebra $\mathfrak{su}(2)$. This is $\mathfrak{sl}(2,\mathbb{C})$, the algebra corresponding to the special linear group $SL(2,\mathbb{C})$ which is composed of all $2 \times 2$ complex matrices with determinant 1. We can split $SO(1,3)$ into two parts, corresponding to these two copies of $\mathfrak{sl}(2,\mathbb{C})$, which we call the left- and the right-chiral spinorial fields of $SO(1,3)$.

Using this splitting of the restricted Lorentz group into two copies of $SL(2,\mathbb{C})$ we can also start considering the space that the group acts on — Minkowski spacetime. The typical 4-dimensional Minkowski spacetime is made up of four-vectors, which transform under a $(1/2,1/2) = (1/2,0) \otimes (0,1/2)$-representation of the restricted Lorentz group, i.e. as a product of the left- and right-chiral spinors. By this split, it becomes possible to replace the calculations that utilize Dirac gamma matrices [13] with ones using Pauli sigma matrices [1–4]. It is this result which forms the foundation of the spinor-helicity formalism [14–17], as well as its continuation in the chirality-flow formalism [6, 7].

## 2.2 The spinor-helicity formalism briefly

By considering the splitting of Minkowski spacetime as presented above, calculations in 4-dimensional Minkowski spacetime can be replaced with ones in 2-dimensional spinorial spaces [1–4]. Although we will here present a brief introduction to the resulting spinor-helicity formalism, more in-depth entries can be found in e.g. [14–17].

For this thesis, we consider specifically massless fermions and vector bosons in QED. Since massless fermions have conserved helicity (which in the massless case is one-to-one with chirality), there is no mixing between the $(1/2, 0)$ and $(0, 1/2)$ representations, as the contraction between two objects transforming under opposite representations would not be Lorentz invariant.

First, let us consider an incoming fermion flow with momentum $p$. This corresponds to an incoming fermion or an outgoing antifermion, which in the Weyl basis can be written slightly mathematically improperly as

$$u(p) = \begin{pmatrix} u_L \\ u_R \end{pmatrix} = \begin{pmatrix} \tilde{\lambda}_p^{\dot{\alpha}} \\ \lambda_{p,\beta} \end{pmatrix} \quad \text{and} \quad v(p) = \begin{pmatrix} v_L \\ v_R \end{pmatrix} = \begin{pmatrix} \tilde{\lambda}_p^{\dot{\alpha}} \\ \lambda_{p,\beta} \end{pmatrix}, \tag{2.10}$$

where

$$u_{L/R} = P_{L/R} u \quad \text{and} \quad v_{L/R} = P_{L/R} v, \tag{2.11}$$

with the chiral projection operators $P_{L/R} = \frac{1}{2}(1 \pm \gamma^5)$ where $\gamma^5 \equiv i\gamma^0\gamma^1\gamma^2\gamma^3$ is the usual fifth gamma matrix. That is, $u_L$ and $v_L$ are the projections of the 4-component Dirac spinors onto the left-chiral $(1/2, 0)-$representation, and vice versa for $u_R/v_R$ and the right-chiral $(0, 1/2)$-representation. These are then the 2-component Weyl spinors $\lambda_{p,\beta}$ and $\tilde{\lambda}_p^{\dot{\alpha}}$. Here, the dotted/undotted indices refer to which representation of $SO(1, 3)$ the spinor in question transforms under: Left-chiral spinor components have dotted indices, while right-chiral spinor components have undotted indices.

Using the decomposition,

$$p^{\pm} = p^0 \pm p^3, \quad p^{\perp} = p^1 + ip^2, \quad p^{\perp *} = p^1 - ip^2, \tag{2.12}$$

these Weyl spinors can be written explicitly as

$$\lambda_{p,\beta} \longleftrightarrow |p\rangle = \frac{1}{\sqrt{p^+}} \begin{pmatrix} p^+ \\ p^{\perp} \end{pmatrix}, \tag{2.13}$$

$$\tilde{\lambda}_p^{\dot{\alpha}} \longleftrightarrow |p] = \frac{1}{\sqrt{p^+}} \begin{pmatrix} p^{\perp *} \\ -p^+ \end{pmatrix}. \tag{2.14}$$

Here, we have introduced the convention of using angle brackets to denote right-chiral spinors and square brackets to denote left-chiral spinors, respectively. Outgoing Weyl

spinors are given by Hermitian conjugation of the corresponding ket in the opposite representation,

$$|p\rangle^\dagger = [p| \quad \text{and} \quad |p]^\dagger = \langle p| \,. \tag{2.15}$$

The reason for this last relationship follows from two facts. First, we can define the dotted/undotted indices of Weyl spinors to correspond to complex conjugation, i.e.

$$\tilde{\lambda}_p^\alpha = \left(\tilde{\lambda}_p^{\dot{\alpha}}\right)^* \quad \text{and} \quad \lambda_{p,\dot{\beta}} = (\lambda_{p,\beta})^* \,, \tag{2.16}$$

which can be seen to hold componentwise from the explicit 2-component representations above. The second fact is that the invariant tensor in $SL(2,\mathbb{C})$ which allows us to raise and lower indices is the Levi-Civita symbol,

$$\epsilon^{\dot{\alpha}\dot{\beta}} = \epsilon^{\alpha\beta} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \quad \text{and} \quad \epsilon_{\dot{\beta}\dot{\alpha}} = \epsilon_{\beta\alpha} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}. \tag{2.17}$$

Utilizing these relationships between dotted/undotted and upper/lower indices, it is simple to check that the relationship for the bras presented above holds.

Note here that we work in the typical spinor-helicity convention, where the chirality of an incoming spinor is given by which representation of $SO(1,3)$ it transforms under. This is in stark contrast to the typical convention used by e.g. Peskin & Schroeder [13], where the handedness of an outgoing spinor is defined based on which representation of $SO(1,3)$ its corresponding incoming spinor transforms under. Explicitly,

$$\begin{aligned} \text{Spinor-helicity: } & \overline{u}_{L/R} = \overline{u} P_{L/R}, \\ \text{Standard: } & \overline{u_{L/R}} = u^\dagger P_{L/R} \gamma^0 = u^\dagger \gamma^0 P_{R/L} = \overline{u} P_{R/L} = \overline{u}_{R/L}. \end{aligned} \tag{2.18}$$

The reader should thus note that in our convention, a left-/right-chiral outgoing spinor might be what they would normally consider a right-/left-handed outgoing spinor.

In the Weyl basis, the Dirac matrices take the form

$$\gamma^\mu = \begin{pmatrix} 0 & \sigma^{\mu,\dot{\alpha}\beta} \\ \bar{\sigma}^\mu_{\beta\dot{\alpha}} & 0 \end{pmatrix} = \begin{pmatrix} 0 & \sqrt{2}\tau^{\mu,\dot{\alpha}\beta} \\ \sqrt{2}\bar{\tau}^\mu_{\beta\dot{\alpha}} & 0 \end{pmatrix}, \tag{2.19}$$

where the Pauli matrices $\sigma^\mu = (I_{2\times 2}, \sigma^i)$ and $\bar{\sigma}^\mu = (I_{2\times 2}, -\sigma^i)$ have been normalized as the $\tau^\mu$ and $\bar{\tau}^\mu$ matrices such that we have that

$$\mathrm{Tr}\left(\tau^\mu \tau^\nu\right) = \delta^{\mu\nu} \quad \text{and} \quad \mathrm{Tr}\left(\tau^\mu \bar{\tau}^\nu\right) = g^{\mu\nu}, \tag{2.20}$$

with $\delta^{\mu\nu}$ the Kronecker delta and $g^{\mu\nu}$ the Minkowski metric given by $\mathrm{diag}(1,-1,-1,-1)$.

Lorentz scalars in this notation are given by spinor inner products. From the previous points, the form of these inner products follows:

$$\langle ij\rangle \equiv \langle i|\,|j\rangle = \lambda_i^\alpha \lambda_{j,\alpha} = \epsilon^{\alpha\beta}\lambda_{i,\beta}\lambda_{j,\alpha} = -\epsilon^{\beta\alpha}\lambda_{i,\beta}\lambda_{j,\alpha} = -\lambda_{i,\beta}\lambda_j^\beta = -\langle ji\rangle \,, \tag{2.21}$$

$$[ij] \equiv [i|\,|j] = \tilde{\lambda}_{i,\dot{\alpha}}\tilde{\lambda}_j^{\dot{\alpha}} = \epsilon_{\dot{\alpha}\dot{\beta}}\tilde{\lambda}_i^{\dot{\beta}}\tilde{\lambda}_j^{\dot{\alpha}} = -\epsilon_{\dot{\beta}\dot{\alpha}}\tilde{\lambda}_i^{\dot{\beta}}\tilde{\lambda}_j^{\dot{\alpha}} = -\tilde{\lambda}_i^{\dot{\beta}}\tilde{\lambda}_{j,\dot{\beta}} = -[ji] \,. \tag{2.22}$$

The antisymmetry of the spinor inner products showcased above implies that $\langle ii\rangle = [ii] = 0$.

Next, we consider Lorentz four-vectors. Since these transform under a direct product representation $(1/2, 1/2) = (1/2, 0) \otimes (0, 1/2)$ of the restricted Lorentz group, they can be mapped to Hermitian $2 \times 2$-matrices (bispinors) by contraction with the Pauli matrices,

$$p^{\dot{\alpha}\beta} \equiv p_\mu \tau^{\mu,\dot{\alpha}\beta} \longleftrightarrow p_\mu \tau^\mu = \frac{1}{\sqrt{2}} \begin{pmatrix} p^- & -p^{\perp *} \\ -p^\perp & p^+ \end{pmatrix}, \tag{2.23}$$

$$\bar{p}_{\alpha\dot{\beta}} \equiv p_\mu \bar{\tau}^\mu_{\alpha\dot{\beta}} \longleftrightarrow p_\mu \bar{\tau}^\mu = \frac{1}{\sqrt{2}} \begin{pmatrix} p^+ & p^{\perp *} \\ p^\perp & p^- \end{pmatrix}. \tag{2.24}$$

The momentum bispinors can also be represented using slash notation of the form,

$$\not{p} \equiv p_\mu \sigma^\mu = \sqrt{2} p_\mu \tau^\mu \quad \text{and} \quad \bar{\not{p}} \equiv p_\mu \bar{\sigma}^\mu = \sqrt{2} p_\mu \bar{\tau}^\mu, \tag{2.25}$$

which is not to be confused with the Feynman slash. Notably, for light-like (massless) momenta $p$, the bispinors can be expressed as outer products of Weyl spinors:

$$\not{p} = |p]\langle p| \quad \text{and} \quad \bar{\not{p}} = |p\rangle [p|\,, \tag{2.26}$$

for $p^2 = 0$. This finally allows us to express polarization vectors in terms of Weyl spinors and as bispinors [1–4, 6, 7],

$$\epsilon^\mu_-(p_i, r) = \frac{\lambda_i^\alpha \bar{\tau}^\mu_{\alpha\dot{\beta}} \tilde{\lambda}_r^{\dot{\beta}}}{\tilde{\lambda}_{i,\dot{\gamma}} \tilde{\lambda}_r^{\dot{\gamma}}} = \frac{\langle i| \bar{\tau}^\mu |r]}{[ir]} \longleftrightarrow \epsilon^{\dot{\beta}\alpha}_- = \frac{\tilde{\lambda}_r^{\dot{\beta}} \lambda_i^\alpha}{\tilde{\lambda}_{i,\dot{\gamma}} \tilde{\lambda}_r^{\dot{\gamma}}} = \frac{|r]\langle i|}{[ir]}, \tag{2.27}$$

$$\epsilon^\mu_+(p_i, r) = \frac{\lambda_r^\alpha \bar{\tau}^\mu_{\alpha\dot{\beta}} \tilde{\lambda}_i^{\dot{\beta}}}{\lambda_r^\gamma \lambda_{i,\gamma}} = \frac{\langle r| \bar{\tau}^\mu |i]}{\langle ri\rangle} \longleftrightarrow \epsilon^{\dot{\beta}\alpha}_+ = \frac{\tilde{\lambda}_i^{\dot{\beta}} \lambda_r^\alpha}{\lambda_r^\gamma \lambda_{i,\gamma}} = \frac{|i]\langle r|}{\langle ri\rangle}, \tag{2.28}$$

with $p_i$ the vector boson momentum and $r$ an arbitrary light-like reference momentum for which $p_i \cdot r \neq 0$. Here, we use the $\pm$ subscripts to denote vector boson helicity. The polarization vectors can also be expressed as oppositely directed bispinors, i.e. with angled bras and square kets [6], but the expressions above will suffice for our purposes.

These are some of the necessary building blocks for a spinor-helicity description of massless QED and QCD. However, this can be taken one step further by analysing the flow of the chiral representations, which we provide an overview of in the next section.

## 2.3 Fundamentals of the chirality-flow formalism

Chirality flow follows from the consideration of the possible paths chirality may take in a given diagram, akin to how color decomposition in QCD allows for a neat color-flow description of interactions [18–20]. Here, we will present an introductory summary of the chirality-flow formalism, but the reader may consult the two papers developing the formalism [6, 7] for an in-depth description.

The first building block necessary is of course the spinors. We start by equating a right-chiral (undotted) spinor with a solid line, and a left-chiral (dotted) one with a dashed line. Furthermore, we will use the convention that chirality flows opposite to the direction of fermion-flow, such that an outgoing fermion will have incoming chirality and vice versa for antifermions. This choice is convenient, as it will allow us to read chirality-flow diagrams along the arrow direction within the diagram, making evaluation simple. Note again here that we are using the convention that all particles (and consequently all spinors) are outgoing, i.e. all incoming particles and antiparticles are crossed to outgoing. This is represented graphically as

Right-chiral fermion: $\lambda_i^\alpha \longleftrightarrow \langle i| =$  (2.29)


Right-chiral antifermion: $\lambda_{j,\alpha} \longleftrightarrow |j\rangle =$  (2.30)


Left-chiral fermion: $\tilde{\lambda}_{i,\dot{\alpha}} \longleftrightarrow [i| =$  (2.31)


Left-chiral antifermion: $\tilde{\lambda}_j^{\dot{\alpha}} \longleftrightarrow |j] =$  , (2.32)

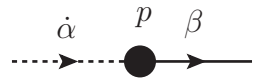
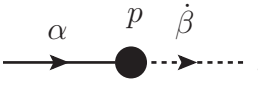

and from this we immediately see the representation of the spinor inner products to be

$\langle ij\rangle =$  $= -\langle ji\rangle = -$  , (2.33)


$[ij] =$  $= -[ji] = -$  , (2.34)

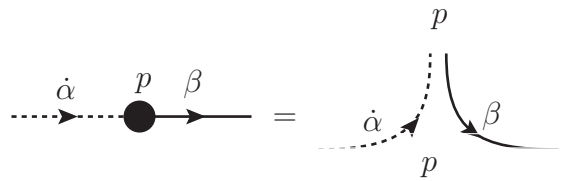

such that chirality-flow diagrams are read following the chirality-flow arrow.

Next, we need to consider four-vectors. As these transform under a $(1/2, 1/2)$ direct product representation of the restricted Lorentz group, they necessitate both types of lines. The mixed index structure of the bispinor representation of momenta leads to a convenient definition of a momentum-dot,

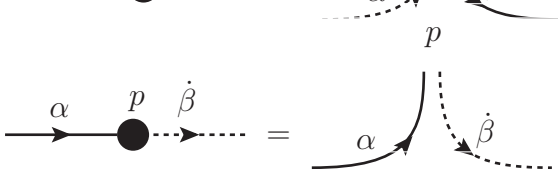
$\not{p} \leftrightarrow \sqrt{2}p^{\dot{\alpha}\beta} \equiv$  and $\bar{\not{p}} \leftrightarrow \sqrt{2}\bar{p}_{\alpha\dot{\beta}} \equiv$  , (2.35)


and since light-like momenta can be described by outer products as in eq. (2.26), we can

graphically represent light-like momenta by

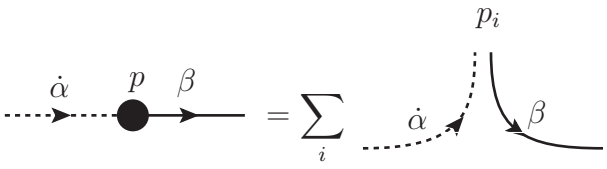



(2.36)




(2.37)

where the external $p$ on the right-hand sides refer to implicit external spinor contractions with momentum $p$. While this only holds for massless momenta, any momentum can be written as a linear combination of massless momenta. At massless tree-level, we can set this combination to be the sum of the external particle momenta that enter the momentum-dot,

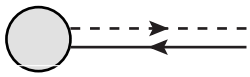



(2.38)

where $p_i$ is the momentum of an external particle.

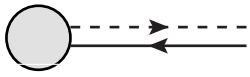
Finally, we consider polarization vectors, which from (2.27) can be written as bispinors, and we find them to have the same general structure as the massless momentum-dots. For outgoing photons,

$$\epsilon_-^{\dot\beta\alpha}(p_i, r) = \frac{|r]\,\langle i|}{[ir]} = \frac{1}{[ir]}$$




, (2.39)

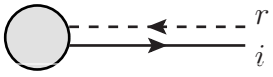
$$\epsilon_+^{\dot\beta\alpha}(p_i, r) = \frac{|i]\,\langle r|}{\langle ri\rangle} = \frac{1}{\langle ri\rangle}$$



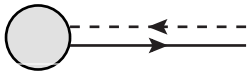

, (2.40)

corresponding to the bispinors in eqs. (2.27,2.28). Additionally, the bispinors with oppositely directed chirality flow are given by

$\frac{1}{[ir]}$




and $\frac{1}{\langle ri\rangle}$




. (2.41)

We note here that the reference momentum is given by its own chirality-flow line. This fact has been used in the development of `CAFE` to make a simple albeit effective choice of gauge vector for external photons, as detailed in Section 5.1.

## 2.4 Chirality-flow Feynman rules for massless QED

Having defined expressions for external particles within chirality flow, it now remains to define vertices and propagators. Here, we will only treat the vertices and propagators that

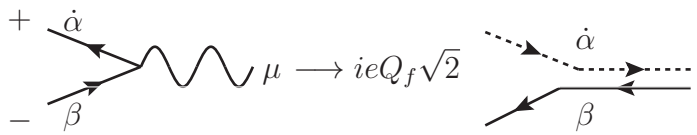
are part of massless QED, as this is what has been implemented throughout this project, but vertices for the full massive SM can be found in [6, 7].

We start with the fermion-antifermion-photon vertex, the only vertex which appears in QED, and which in typical Feynman rules is given by

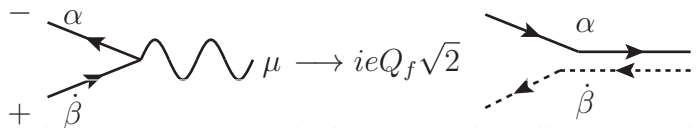
$$ieQ_f\gamma^\mu = \quad ,\tag{2.42}$$


with $e$ and $Q_f$ the electromagnetic coupling and the involved electromagnetic charge, respectively. If we explicitly include spinor indices this vertex splits into two different vertices, depending on the helicity configuration, giving us in the fully outgoing convention [6]

$$ieQ_f\sqrt{2}\tau^{\mu,\dot{\alpha}\beta} = \quad \longrightarrow ieQ_f\sqrt{2} \quad ,\tag{2.43}$$




$$ieQ_f\sqrt{2}\bar{\tau}^{\mu}_{\alpha\dot{\beta}} = \quad \longrightarrow ieQ_f\sqrt{2} \quad .\tag{2.44}$$



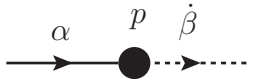

The remaining parts are the propagators, which are similar albeit not identical to their corresponding free particles. There are two types of propagators to consider here: Massless fermions and massless bosons. For fermions, the propagator structure is off-diagonal,

$$\gamma^\mu = \begin{pmatrix} 0 & \sigma^{\mu,\dot{\alpha}\beta} \\ \bar{\sigma}^{\mu}_{\beta\dot{\alpha}} & 0 \end{pmatrix} = \begin{pmatrix} 0 & \sqrt{2}\tau^{\mu,\dot{\alpha}\beta} \\ \sqrt{2}\bar{\tau}^{\mu}_{\beta\dot{\alpha}} & 0 \end{pmatrix},\tag{2.45}$$

with one element coupling left- to right-chiral structures, and another element which couples vice versa. Thus, the fermion propagator splits into two terms, each with the form of a momentum-dot,

$$\frac{i}{p^2}\not{p} \longrightarrow \frac{i}{p^2}\sqrt{2}p^{\dot{\alpha}\beta} = \frac{i}{p^2} \quad ,\tag{2.46}$$


$$\frac{i}{p^2}\bar{\not{p}} \longrightarrow \frac{i}{p^2}\sqrt{2}\bar{p}_{\alpha\dot{\beta}} = \frac{i}{p^2} \quad ,\tag{2.47}$$




where $p$ is the momentum of the propagating fermion. Due to the off-shell nature of this momentum, it will need to be decomposed into a linear combination of lightlike momenta to make use of the momentum-dot outer product forms given by eqs. (2.36,2.37).

Finally, we have the spin-1 boson propagator, which has the structure of the metric,

$$-i\frac{g_{\mu\nu}}{p^2} \longrightarrow -\frac{i}{p^2} \quad ,\tag{2.48}$$

where $p$ is again the propagator momentum, and we have neglected the chirality-flow arrows for simplicity, but we again note that these need to be of opposite direction and chosen consistently with the rest of the diagram. This makes the structure of the photon propagator equivalent with a Kronecker delta in each spinorial field chirality flow.

Here we note that the chirality-flow arrow direction needs to be set consistently within a chirality-flow diagram. This should follow implicitly from the rules already established, such as how a single chirality-flow line can flow in only one direction, and how the lines in a photon (whether external or internal) must have oppositely directed chirality-flow, but this bears mentioning to make it abundantly clear. Although a chirality-flow diagram can be evaluated with chirality flowing along different directions, we need to set these diagrams up with consistency: chirality must flow in from one point and out to another. This fact is central during implementation, since an incorrect algorithm may end up with e.g. chirality flow mismatch in photon propagators, as is encountered during the implementation of `CAFE`, as described in Section 4.1.

# 3 Technical background

With a general understanding of the chirality-flow formalism established, we also need to detail the toolset within which `CAFE` has been developed. `CAFE` constitutes a modified version of MadGraph5_aMC@NLO [21], using a `UFO` [11] file built using the Feynman rules described in Section 2.4, which modifies both diagram generation within MadGraph5 as well as the `ALOHA` [22] output `HELAS`-like code [10] to perform calculations using chirality flow.

## 3.1 Overview

We here give a brief overview of the steps used for evaluating a matrix element squared within the MadGraph5_aMC@NLO software family. A flowchart describing the general process is given in Figure 1, which shows the procedure to go from a model as described by a `UFO` format file to get to a number as evaluated by `HELAS`-like code. This `HELAS`-like code is a library of functions used to iteratively build up Feynman diagrams while reusing the same building blocks. The individual parts are described more extensively in the upcoming subsections.

As a remark, it should be noted that the `HELAS`-like code used in the evaluation stage is not the one given by the original `HELAS` library [10]. As of the fifth version of MadGraph, `ALOHA` [22] is used to automatically generate `HELAS`-like code describing the specific types of structures that appear in the considered Feynman diagram. These structures are described in analytic mathematical form by the `UFO` model, based on which `ALOHA` expresses the matrix algebra necessary to calculate the desired matrix elements. This is elaborated on in Section 3.3.1.

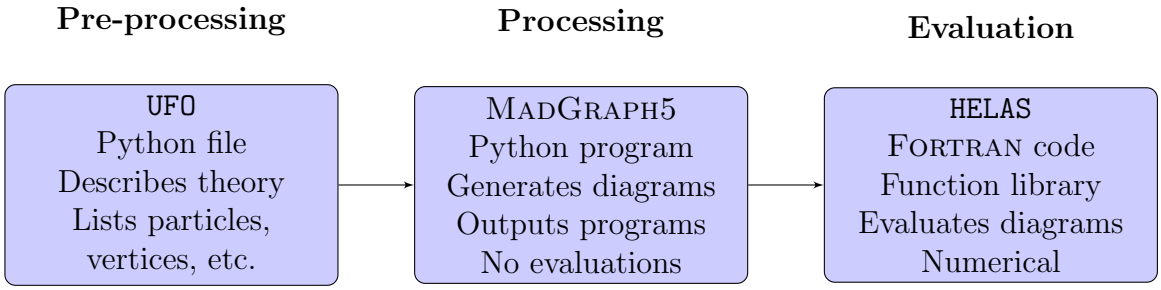



**Figure 1.** Overview of the three stages it takes to evaluate a matrix element using the MadGraph5_aMC@NLO software family. Briefly put, the UFO model describes the model considered, MadGraph5 generates diagrams analytically, and HELAS evaluates these diagrams numerically.

## 3.2 UFO

The Universal FeynRules Output (UFO) format [11] is an object-oriented model format to describe general QFT models for automated matrix-element generators. It is universal in the sense that it has been generalised to be generator-independent, i.e. the UFO format itself can be used to describe almost any model in a generator independent way. A UFO model is given by a set of Python files, and it can be constructed regardless of the number of particles, the color structure, or the Lorentz structures that may appear in interaction vertices, and the object-oriented structure of the format makes it simple for the programs that utilize a UFO file to parse it.

Fundamentally, any QFT can be described by three distinct sets of information [11]:

- a set of particles and their quantum numbers,
- a set of parameters and constants,
- a set of vertices that can be extracted from the Lagrangian.

To describe these, the UFO model uses a set of Python files where these different structures are defined as different classes of objects with certain attributes. Furthermore, there are some model-independent files, but these are as the name mentions model-independent, and thus have little relevance for the implementation here. They define the classes the model-dependent objects are instances of, as well as some other general functionality of the format.

The model-dependent files which are relevant to this project are:

- `particles.py`,
- `parameters.py`,

- `vertices.py`,
- `couplings.py`,
- `lorentz.py`,
- `coupling_orders.py`.

Each of these model-dependent files consists of a list of the objects of the given class: `particles.py` lists all the particles in the model, `parameters.py` lists all the necessary parameters of the model (e.g. masses and constants), and so on. Every object is instantiated as an object of the given class, with each class having the corresponding necessary information as an attribute, such as the spin and fermion number of particles.

Of particular interest is the `lorentz.py` file, which dictates the types of Lorentz structures that go into the vertices of the model. In a sense, these can be said to represent the *types* of vertices the QFT in question has, with the actual `vertices.py` file explicitly describing which particles can go into which Lorentz structures. Let us take an example: A fermion-antifermion-vector vertex as presented in [11], which is the only type of structure in QED. The corresponding Lorentz structure is written as

```
FFV1 = Lorentz(name = 'FFV1',
               spins = [ 2, 2, 3 ],
               structure = 'Gamma(3,2,1)')
```

where `Gamma(3,2,1)` signifies the gamma matrix structure $(\gamma^{\mu_3})_{i_2,i_1}$, meaning that this structure would give a vertex of the form $\bar{u}_2\gamma^\mu u_1\epsilon_{3\mu}$, where $u_1$ and $u_2$ are spin-1/2 fermions and $\epsilon_3$ is a spin-1 vector. This is the only Lorentz structure currently used in `CAFE`, although it has been split into several sub-structures, as elaborated on in Section 4. As is likely clear from this, the spins are given as $(2s+1)$ within the `UFO` model.

A vertex that utilizes the `FFV1` Lorentz structure is the electron-positron-photon vertex. In the SM `UFO` model provided with MadGraph5_aMC@NLO, this vertex is included as a `Vertex`-class object in the `vertices.py` called `V_98`. It is written as

```
V_98 = Vertex(name = 'V_98',
              particles = [ P.e__plus__, P.e__minus__, P.a ],
              color = [ '1' ],
              lorentz = [ L.FFV1 ],
              couplings = {(0,0):C.GC_3})
```

with `P.e__plus__` and `P.e__minus__` the positron and electron, respectively, and `P.a` the photon. This vertex-class object says that these three particles couple to each other with the Lorentz structure `FFV1`, under the identity (`'1'`) representation of QCD, with a single coupling `C.GC_3` which is, unsurprisingly, $-ie$, with $e$ the elementary charge.

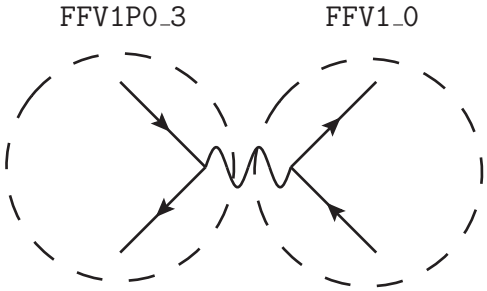



**Figure 2.** Diagram showcasing which parts are calculated by which HELAS functions. The circle on the left calculates the wavefunction of the intermediate photon using FFV1P0_3, while the circle on the right calculates the diagram amplitude using FFV1_0.

As is probably clear now, each vertex for each set of particles needs to be defined explicitly. In this sense, the electron-positron-photon vertex and the muon-antimuon-photon vertex are considered different vertices, despite being structurally identical save for the associated particles. Note here that vertices within the UFO model are taken to be entirely outgoing, which is why all the ones considered here are of the form particle-antiparticle-boson. All other vertices of the same form are related to these by crossing symmetry. For example, vertices used in Compton scattering would be the same as the ones used for electron pair annihilation to electron pair production.

## 3.3 HELAS

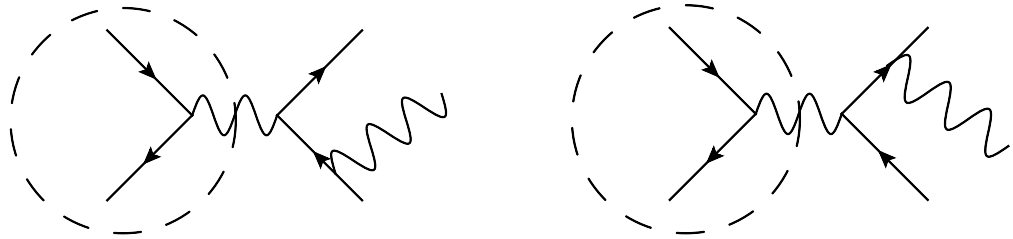
In 1992, a library of FORTRAN77 routines entitled HELAS (HELicity Amplitude Subroutines for Feynman Diagram Evaluations) [10] was released. The name is a play on the Japanese word 減らす, romanized *herasu*, which translates to 'to reduce', as it was intended to reduce the amount of tasks necessary in Feynman diagram evaluations.

In short, the original HELAS library is a set of subroutines for calculating on-shell wavefunctions, off-shell wavefunctions, and the amplitudes of the tree-level diagrams involving them. The simplest functions are definitions of the external wavefunctions, but there are also so-called vertex functions. The latter functions come in two types.

The first form of vertex function takes as input several particle wavefunctions, either on-shell or off-shell, and outputs an off-shell particle wavefunction. In the process

$$e^+e^- \to \gamma \to \mu^+\mu^-, \tag{3.1}$$

HELAS would create the intermediate off-shell photon using a function called FFV1P0_3, where the _3 denotes that the vector boson is the off-shell particle. This function takes the electron and positron wavefunctions and momenta as input, and calculates the wavefunction for the photon based on them. These vertex functions can be said to be *many-to-one*,



**(a)** **(b)**

**Figure 3.** Two diagrams contributing to the process $f^-f^+ \to f^-f^+\gamma$. The encircled part is shared between the two diagrams, and need only be calculated once.

in that they take several off-shell or on-shell particles as input and output a new off-shell particle. This is illustrated in the left-hand circle in Figure 2.

Secondly, there are functions for calculating amplitudes. These functions calculate the amplitudes corresponding to individual Feynman diagrams, and unlike the off-shell vertex functions they are *many-to-zero*: They take several particles as input, and output a complex scalar. These are denoted by having `_0` in their name. Going back to the example in eq. (3.1), the amplitude would be calculated by a function called `FFV1_0`, which takes as input the two muon wavefunctions as well as the photon wavefunction given by `FFV1P0_3`. This is illustrated in the right-hand circle in Figure 2, and the corresponding `HELAS` calls are

```
      CALL OXXXXX(P(0,1),ZERO,NHEL(1),-1*IC(1),W(1,1))
      CALL IXXXXX(P(0,2),ZERO,NHEL(2),+1*IC(2),W(1,2))
      CALL IXXXXX(P(0,3),ZERO,NHEL(3),-1*IC(3),W(1,3))
      CALL OXXXXX(P(0,4),ZERO,NHEL(4),+1*IC(4),W(1,4))
      CALL FFV1P0_3(W(1,2),W(1,1),GC_3,ZERO,ZERO,W(1,5))
C     Amplitude(s) for diagram number 1
      CALL FFV1_0(W(1,3),W(1,4),W(1,5),GC_3,AMP(1))
```

In the code snippet above, the first four lines define the external fermions and output them to `W`, the list of wavefunctions. `OXXXXX` has outflowing fermion flow, and `IXXXXX` has inflowing fermion flow. The vertex function `FFV1P0_3` calculates the intermediate photon and outputs it as `W(1,5)`, i.e. the fifth wavefunction in this process. Then, the ampltiude is calculated by `FFV1_0`, where the arguments can be seen to be ordered as inflowing-outflowing-vector.

However, `HELAS` is not a program in itself. It is only a library of functions to perform numerical calculations, and needs a separate program to generate the necessary function calls to evaluate a process, such as MadGraph5. This is illustrated in Figure 1, where MadGraph5 performs diagram generation ('processing').

The distinctive feature of the `HELAS` library is its simplicity, allowing for high levels of recycling during calculations. Any currents shared between different diagrams only need

to be calculated once, before they can then be called for calculations of any diagrams which require them. In the original HELAS report [10], the authors mention that the amplitude of the process $e^-e^+ \to e^-\bar{\nu}_e W^+ Z^0$, which at tree-level has 80 different contributing diagrams, can be calculated using only 65 distinct HELAS calls, 25 of which were off-shell particle calculations and 40 of which were amplitude calculations (the remaining 40 amplitudes were related to these 40 by crossing symmetry, and did not need to be calculated numerically).

An example of this recycling can be seen in Figure 3, which depicts two of the diagrams which contribute to the process $f^-f^+ \to f^-f^+\gamma$ for some fermion $f$. The off-shell photon, calculated using the off-shell vertex function FFV1P0_3, is shared between the two diagrams. A HELAS routine made to evaluate this process would only need to perform this calculation once, and then reuse that result when evaluating the other diagram.

### 3.3.1 ALOHA

The original HELAS library was formulated specifically for SM calculations, but analysis of more exotic models have become more and more relevant in recent years. Creating tools for numerical evaluations within such models means extending the HELAS library by creating HELAS-like code performing evaluations for vertex structures besides those in the SM and for particles with spin $> 1$.

In 2012, an application was developed for the explicit purpose of generating the necessary HELAS-like code for *any* given QFT. This was ALOHA — Automatic Libraries Of Helicity Amplitudes for Feynman diagram computations [22]. As the name suggests, ALOHA is a tool for automatically generating the necessary HELAS-like library to numerically evaluate any process within any BSM model.

ALOHA was built alongside the UFO format [11] described in Section 3.2, and takes as input a model described by a UFO file, from which it can then specifically compute the HELAS-like routines necessary to calculate amplitudes and output those for accessibility by e.g. a Feynman diagram generator such as MadGraph5 [8].

While the calculation of necessary HELAS-like routines for a given diagram will increase the amplitude calculation time, it yields the benefit of not needing to prepackage the matrix element calculation program with a library of HELAS-like routines for every model the program can treat. This simplifies the process of implementing more exotic models, such as those with spin-3/2 particles, or with more complicated Lorentz structures than those seen in the SM, as it now suffices to create a UFO model corresponding to this BSM model.

However, despite the generalised nature of ALOHA, it relies on an assumption of the form of the mathematical structures within the model. The spinorial form of Lorentz structures are taken to be 4-dimensional. That is, ALOHA does not allow for directly replacing calculations that utilize Dirac matrices with ones that instead utilize Pauli matrices in the format desired by the chirality-flow formalism. This is elaborated on in Section 4.

## 3.4 MadGraph5_aMC@NLO

As anyone who has taken a course in QFT can vouch for, calculations within the SM and models beyond it are tedious at best. Combining the difficulties of calculation with the sheer amount of diagrams that can mediate a given process makes it unsurprising that calculations within the field of particle physics are largely left up to numerical tools today. However, such tools can be difficult to create.

One of these tools was MadGraph [9], a Fortran program initially published in 1994 which automatically generated tree-level QCD and EW (electroweak) diagrams and used the `HELAS` libraries described in Section 3.3 to calculate the corresponding matrix element. Development has continued until today, and the program is now up to its sixth major version [21].

In 2012 MadGraph5 was unified with aMC@NLO, a tool for automatic Monte Carlo at Next-to-Leading Order calculations, creating MadGraph5_aMC@NLO [8]. MadGraph5_aMC@NLO is a software suite and framework which "aims at providing all the elements necessary for SM and BSM phenomenology, such as the computations of cross sections, the generation of hard events and their matching with event generators, and the use of a variety of tools relevant to event manipulation and analysis" [23].

The MadGraph5_aMC@NLO framework is easy to use: After installation, one can simply run it and tell it to generate a given process within a given model. With the process generated, the amplitude can be extracted, events generated, or otherwise treated as desired. As an example, a template process generation card provided with MadGraph5_aMC@NLO is shown below.

```
import model sm
# Define multiparticle labels
define p = g u c d s u~ c~ d~ s~
define j = g u c d s u~ c~ d~ s~
define l+ = e+ mu+
define l- = e- mu-
define vl = ve vm vt
define vl~ = ve~ vm~ vt~
# Specify process(es) to run
generate p p > e- ve~ @1
add process p p > e- ve~ j @2
add process p p > t t~ @3
# Output processes to MadEvent directory
output
```

In the snippet above, the SM `UFO` model is imported, several multiparticles are defined, and these multiparticles are then used to generate proton collision processes in lines $10 - 12$. The process is exported, calling on `ALOHA` to generate the necessary `HELAS` routines with the

`output` command. From there, one can e.g. use the included event generator MadEvent to numerically estimate the cross section for this process. To put it simply, the first line above corresponds to the pre-processing step in Figure 1 and the remainder to the processing stage. If one were to run this directly in MadGraph5, rather than as a process card, one could go directly to cross section evaluation after the `output` command by using the `launch` command.

One detail that makes MadGraph5_aMC@NLO more adaptable than a simple tool for studying the SM is its use of the `UFO` format [11] and the `ALOHA` application [22], detailed in Sections 3.2 and 3.3 respectively. With the right `UFO` model, MadGraph5_aMC@NLO can generate diagrams and create `HELAS` calls most general models. This makes initial numerical studies into BSM models very simple, leaving out the necessity for development of new tools for every new model.

Another remark worth making here is the distinction between different outputs that MadGraph5_aMC@NLO can produce. As previously mentioned, the standard MadGraph5 output will be a program for running MadEvent, an event generator, to evaluate cross sections. However, one can make a simpler output using the `output standalone` command. The `standalone` output is simply the `HELAS`-like routine generated by `ALOHA` to calculate the matrix element squared of a given process, evaluated either with user input parameters or at a phase space point randomly generated by the `RAMBO` algorithm [24]. It is this `standalone` output that we are working with in this project. When run, a program given by `output standalone` just evaluates the matrix element squared at the given momenta.

# 4 CAFE

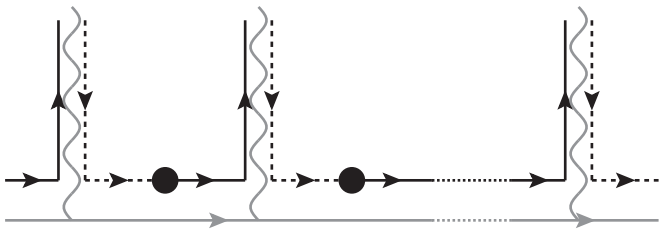
The Chiral Automized Flow Extension for MadGraph5_aMC@NLO, or `CAFE` for short, is a modified version of the MadGraph5_aMC@NLO software suite which has been made for the specific purpose of using the chirality-flow formalism to simplify and minimize numerical calculations. In this section, we present the details of an initial version of `CAFE`, which implements massless tree-level QED.

## 4.1 Automized chirality flow

As eqs. (2.21,2.22) and (2.33,2.34) show, spinor inner products are antisymmetric. Since massless fermions do not mix the left- and right-chiral representations of $SO(1,3)$, a chirality-flow diagram can be related to one with directions reversed by a factor $(-1)^m$, where $m$ is the number of chirality-flow lines that have had their directions reversed. In a sense, we are free to *pick* the direction of chirality flow when evaluating an amplitude, as long as all other rules of the chirality-flow formalism are obeyed.

To automate chirality flow, it is then insufficient to construct diagrams with chirality-flow lines: we must also make sure that chirality-flow direction is chosen consistently, i.e. such that we never have a chirality-flow arrow mismatch. Initially, the idea was to create a smart algorithm for. However, a simpler solution was implemented instead.

First, we cease to distinguish between fermions and antifermions when it comes to the chiral states, such that a left-/right-chiral fermion has the same spinor structure as the corresponding antifermion. (Henceforward we will thus not distinguish between fermions and antifermions in text.) Then, we set external left-chiral fermions to inflowing chirality, and right-chiral fermions to have outflowing chirality, as in eqs. (2.30,2.31). External photons were then set to have the opposite chirality-flow direction to this, as in eqs. (2.39,2.40). With this convention, the inner product between an external photon spinor and an external fermion spinor will always be taken correctly. One benefit of this choice is that we now only need to evaluate one off-shell current at each vertex, rather than the two (from the number of ways chirality can flow) we might otherwise have. This choice will yield consistent chirality flow for a fermion line connected to an arbitrary number of external photons, as illustrated below,



. (4.1)

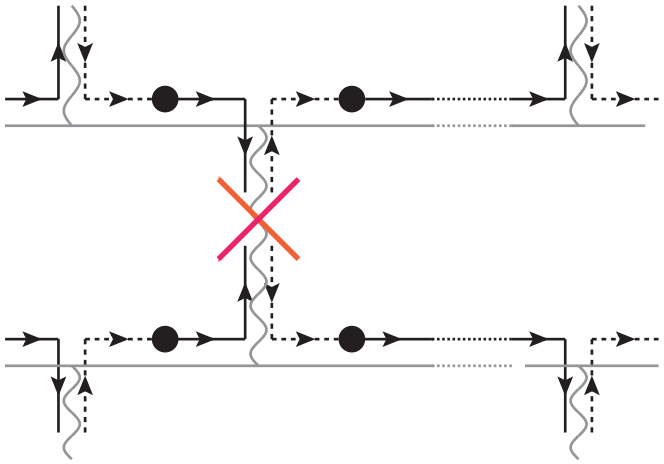
In a sense, we then end up with four types of external particles: Left- and right-chiral fermions $f_{L/R}$, and left- and right-chiral vector bosons $V_{L/R}$. Analytically, these are

$$f_L = [p|\,, \quad f_R = |p\rangle\,, \quad V_L = \frac{|p]\,\langle r|}{\langle rp\rangle}, \quad V_R = \frac{|r]\,\langle p|}{[pr]}, \tag{4.2}$$

where $p$ refers to the physical momentum of the particle and $r$ is the reference momentum of the polarization vectors. Again, we note here that we no longer distinguish between fermions and antifermions in a spinorial sense here.

Although this algorithm proves to be consistent for fermion lines with arbitrarily many external photons attached, we find that two fermion lines coupled by an internal photon will necessarily end up with inconsistent chirality flow since the arrows do not align in the

photon propagator. This is illustrated below,



. (4.3)

This convention will thus lead to an inconsistency, but it is a systematic inconsistency: Whenever we connect two fermion lines, there will be a chirality flow mismatch. Since these will only connect with photon propagators, it suffices to switch the direction of chirality flow inside these propagators. A ket is mapped to a bra and vice versa using the Levi-Civita symbol, as shown in eq. (2.17). Thus, our photon propagator as given by eq. (2.48) must additionally include Levi-Civita symbols acting on each of its spinorial components independently. Analytically, we define an off-shell photon $V$ as a function of a left-chiral and a right-chiral fermion $f_L$, $f_R$ as

$$V(f_L, f_R) = -i \frac{\left(\epsilon^{\dot{\alpha}\dot{\beta}} [f_L|_{\dot{\beta}}\right)\left(\epsilon^{\gamma\delta} |f_R\rangle_{\delta}\right)}{p^2}. \tag{4.4}$$

Furthermore, an immediate consequence of this convention is that the momentum-dots in the fermionic propagator (see eqs. (2.46,2.47) will always have inflowing right-handed chirality (dotted line) and outflowing left-handed chirality (solid line). This can be seen in eq. (4.1). Fermionic propagators are structurally given exclusively by the $\bar{p}$ matrix, i.e. by eq. (2.47), for this algorithm.

With this convention defined, we can go on to discuss the practical implementation of chirality flow in `CAFE`.

## 4.2 Framework

As previously mentioned, `CAFE` is built on several levels of modifications of the MadGraph5 suite. These can, slightly simplified, be described as pre-processing, processing, and post-processing, where the first two are shown in the first two boxes in Figure 1, and the post-processing step is an additional step added between MadGraph5 and the numerical evaluation by the `HELAS`-like output routine.

The pre-processing step is described fully by a chirality flow UFO model, which is largely equivalent to the massless QED-sector of the SM UFO model, although with explicit distinction between same-flavor differently-chiral particles. There are also some additions to the ALOHA template files for the external particle wavefunctions.

Processing is a bit more involved. As the UFO format, MadGraph5, and ALOHA are all built considering typical Dirac spinor calculations, some features of the chirality-flow formalism are not natively supported. One particular issue lies in diagram generation — in chirality flow, there is a "flavor change" of particles in the fermionic propagator, with a left-chiral fermion going to a right-chiral fermion and vice versa, as shown in eqs. (2.46,2.47). Furthermore, there exist additional issues with left-chiral and right-chiral particles being treated internally as different particles, which requires consideration.

After diagram generation and ALOHA calls have been made, an issue persists. As the HELAS routines and consequently the ALOHA-generated HELAS-like code rely on classical SM structures, i.e. Dirac spinors and gamma matrices, the code generated by ALOHA will not perform calculations using chirality flow. As such, a small HELAS-like library has been written for the structures necessary for calculations to be made under this formalism, and the post-processing step involves systematically replacing the ALOHA-generated routines with the proper structures from this library.

Once the post-processing step has finished, the output program is run just as it would be for standard MadGraph5. The updated HELAS-like routines are called by CAFE, just as MadGraph5 calls its standalone outputs to evaluate matrix elements squared, as described in Section 3.4. These output programs themselves function exactly as the corresponding programs generated by MadGraph5, i.e. they are Fortran programs that can be run from their output directory in order to perform evaluations.

## 4.3 Pre-processing: The chirality flow UFO model

The current chirality flow UFO model is minimal. Most of it is taken directly from the SM UFO model, with some changes among the particles, Lorentz structures, and vertices.

There are 6+1 particles (or 10+1, if one were to count particles and antiparticles separately): electrons, muons, and photons, each of which is split into left-chiral and right-chiral particles, plus an additional unpolarized photon. For the fermions, the chirality refers to which representation its spinor transforms under, whereas for the photons it refers to the polarization. Additionally, an unpolarized photon exists to differentiate the photon propagator from the external, polarized photons, so that no summation over polarizations needs to be made for off-shell photons.

Additionally, some multiparticles are defined for the UFO model: The non-chiral leptons are given by the multiparticle $l^\pm = l_L^\pm + l_R^\pm$, and similarly unpolarized external photons are given by $\gamma = \gamma_+ + \gamma_-$. To evaluate helicity-summed processes, as standard MadGraph5 does, processes should be generated using these multiparticles rather than the explicitly

chiral ones.

The UFO model contains the vertices we need to consider. In the UFO format, there are four such vertices: The $e_R^+ e_L^- \gamma$ vertex, the $\mu_R^+ \mu_L^- \gamma$ vertex, the $e_L^+ e_R^- \gamma$ vertex, and the $\mu_L^+ \mu_R^- \gamma$ vertex. The former two use the RLV1 Lorentz structure, and the latter two the LRV1 structure, which will be elaborated on shortly. Additionally, each vertex mentioned above is split into three separate vertices in the UFO model, as the photon has been split into three distinct particles as mentioned above. However, since the same type of photon will be consistently used for the same purpose for each chirality configuration, this does not affect HELAS recycling.

Our UFO model has four Lorentz structures. Two of these are used in vertex definitions: RLV1 and LRV1. These correspond to the two vertices that the fermion-antifermion-vector vertex splits into in a strictly chiral formalism, as shown in eqs. (2.43,2.44). The fermion-antifermion-vector structure is of course given by $\gamma^\mu$, so the two chiral structures are given by this multiplied by projection matrices. RLV1 $\equiv \gamma^\mu P_L$ and LRV1 $\equiv \gamma^\mu P_R$. Here, the naming convention is based on the ordering the arguments will have in the final HELAS calls. For standard Dirac spinors, this is given by the ordering {inflowing, outflowing, vector}, with flow here being fermion flow. We retain this ordering, but add the distinction between left- and right-chiral fermions, such that LRV1 will have outflowing left chirality and inflowing right chirality, and vice versa for RLV1, since chirality flows oppositely to fermion flow.

Besides RLV1 and LRV1, there are two additional structures used specifically for the fermion propagator. This will be elaborated on in Section 4.4, but in short we need to replace all off-shell left-chiral fermions with right-chiral fermions and vice versa, due to the momentum-dot in the fermionic propagator described in eqs. (2.46,2.47). As such, these structures are called LLV1 and RRV1, as they effectively couple left-chiral fermions to left-chiral fermions and vice versa for the right-chiral ones, despite the lack of any such vertices in the analytic chirality-flow formalism. Nevertheless, they serve the purpose of making the code and its calculations more accessible to human eyes.

In addition to the chirality flow UFO model, explicitly chiral external particles have been added to the ALOHA template files. These are given in the aloha_functions.f FORTRAN file, which holds the template subroutines the HELAS-like output code will utilize in their calculations. Four new subroutines have been added: LXXXXX, RXXXXX, VLXXXX, and VRXXXX. These follow the convention described in Section 4.1.

The LXXXXX and RXXXXX are very similar to the original fermionic routines IXXXXX and OXXXXX available in aloha_functions.f, although they have been simplified. Essentially, RXXXXX and LXXXXX return the angular ket in (2.13) and the square bra in (2.15), respectively. Should $p^+ = 0$, a separate identity for the spinors given in [7] is used. HELAS routines work internally using 6-component vectors, the first two giving the particle momentum and the final four making up the four-vector wavefunction, so we carry around some zeros in the spinorial component that does not correspond to the fermion chirality. These do not

enter into any calculations.

Additionally, the external fermionic wavefunctions check to make sure that the helicity should, in fact, correspond to a fermion of that chirality. As they are based on the unpolarized fermionic template functions, they take as an argument a helicity. An if-statement determines whether the helicity is correct, given the particle state: A final state fermion with helicity $+1$ is equivalent with an initial state fermion with helicity $-1$, which is left-chiral, and similarly a final state fermion with helicity $-1$ is equivalent with an initial state fermion with helicity $+1$, which is right-chiral. Should these conditions not be fulfilled, the function returns a zero-wavefunction.

However, `VLXXXX` and `VRXXXX` have seen substantial changes from the original vector `VXXXXX` subroutine. They have been rewritten fundamentally to implement the forms of the polarization vectors presented in (2.27) and (2.28). Additionally, they now take as an argument a four-vector reference momentum, from which the reference spinor is constructed. When combined with our modified diagram generation routine (elaborated on in Section 4.4) `CAFE` will immediately discard any diagrams which will be evaluated to zero based on the gauge choice. This gauge choice and its consequences are elaborated on in Section 5.1.

## 4.4 Processing: Generating chirality-flow diagrams

At the processing stage, there has been one major overhaul to the native MadGraph5 program: diagram generation[2]. This is detailed in Section 4.4.1. Additionally, two other aspects of process compilation have been updated, one to do with averaging matrix elements based on how many final state particles are indistinguishable, and one to do with helicity configuration generation.

### 4.4.1 Diagram generation

At diagram generation, two changes have been implemented for `CAFE` that are not present in MadGraph5. One of these is necessary to have a functional fermion propagator, as MadGraph5 will (correctly) identify the off-shell fermion exiting a fermion-antifermion-photon vertex as having the opposite chirality of the one entering. However, including the momentum-dot in the fermion propagator, the fermion exiting such a vertex should effectively have the same chirality as the one entering, see Figure 4. The second feature implemented is *diagram removal*, where any diagram with a zero-valued amplitude based on gauge choice is discarded already at compile time.

The Feynman diagrams generated by MadGraph5 and the `HELAS`-like code generated by `ALOHA` are not built to accommodate fermion propagators as described in chirality flow. In order to treat them, the `diagram_generation.py` file has been modified such that an

[2]Although presented as part of the author's thesis, modifications to diagram generation were written by A. Lifson. This concerns specifically what is presented in Section 4.4.1.

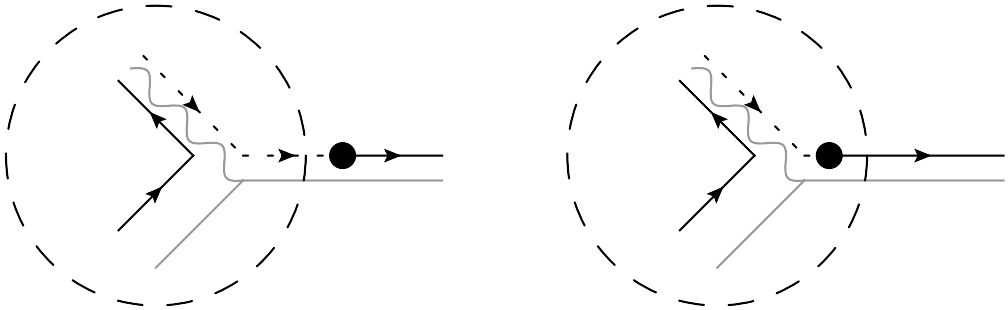



**Figure 4.** Illustration showing the `CAFE UFO` model fermion-antifermion-photon vertex outputting an off-shell fermion as treated by (a) MadGraph5 (b) `CAFE`. The encircled part is how the program will interpret the vertex in question. Note that in (a), the fermion that exits the circle is not the one that should later be used to determine future vertex couplings.

off-shell chiral fermion will be replaced with an off-shell fermion of the opposite chirality. In the chirality-flow formalism, the fermion-antifermion-photon vertex couples only fermions of opposite chirality, as illustrated in eqs. (2.43,2.44). By MadGraph5's standards, this means that a fermion that absorbs or emits a photon will change particle type, going from e.g. $f_L^{\pm} \to f_R^{\pm}$ or vice versa. This is *correct*, but as MadGraph5 does not recognize the momentum-dot included in fermionic propagators $\left(\propto$  $\right)$ this means that MadGraph5's diagram generation will treat the off-shell particle resulting from this vertex as one of the opposite chirality if we directly use MadGraph5's diagram generation with the `CAFE UFO` model.

Rather than try to fix this by implementing an entirely new type of propagator which always changes "flavor", we note that MadGraph5 always calculates off-shell particles using a vertex function. By replacing all vertices which that off-shell fermions of one chirality with vertices that output off-shell fermions of the opposite chirality during diagram generation, diagrams will be generated properly. Essentially, the diagram generation has been modified to forcibly include a fermion chirality change for vertex functions that output off-shell fermions. This is illustrated in Figure 4. The encircled parts of the diagrams depict what MadGraph5 and `CAFE` would determine the output of such a vertex to be, based on the `CAFE UFO` model. We effectively move the momentum-dot in the fermion propagator into the vertex itself. This is why the `LLV1` and `RRV1` structures were included in the `CAFE UFO` model: Because we have moved the momentum-dot into the vertex itself, we now have vertices which effectively couple left-chiral fermions to left-chiral fermions, and right-chiral fermions to right-chiral fermions. This occurs in the `helas_calls_writers.py` file, and changes the `HELAS`-like output by `ALOHA` to reflect this momentum-dot inclusion.

Additionally, the diagram generation and `helas_calls` have also been changed to take care

of diagram removal based on gauge choice, as elaborated on in Section 5.1. The reference momentum in chirality flow is given by an explicit spinor, as can be seen in eqs. (2.39,2.40). Based on the gauge choice detailed in Section 5.1, it follows that any inner products between a left-chiral photon and the first right-chiral fermion in a process will yield multiplication by zero, and vice versa with right-chiral photons with the first left-chiral fermion. Thus, a routine is implemented to detect vertices coupling a left-/right-chiral photon with the first right-/left-chiral fermion, and to remove such vertices for the purpose of diagram generation. Although technically this prevents any such diagrams from being generated to begin with, the analytic background based on which the algorithm has been implemented lends itself to the name diagram removal.

### 4.4.2 Helicity configurations and identical particle factors

When calculating matrix elements for a given process using MadGraph5, a number called the `identical_particle_factor` is assigned to the process by the program. This is a factor that a calculated amplitude needs to be divided by based on the number of particles of the same type in the final state. The exact nature of this factor is not integral for this project, but our distinction between left- and right-chiral particles does mean that MadGraph5 will calculate this factor incorrectly for the `CAFE` `UFO` model. To circumvent this issue, the `identical_particle_factor` function in the `base_objects.py` file has been updated to treat e.g. both the left-chiral electron and the right-chiral electron as the same particle for this calculation.

Finally, during the processing step we also remove all terms in the helicity summation which will contribute zero to amplitude based on the chiralities involved. This is done in `export_v4.py`, where the function `helicity_lines_replacer` treats the `helas_calls` to determine what helicity combinations will actually contribute and replaces the list of treated helicities with a list of relevant configurations only. In massless chirality flow, this will only ever be a single one. We consider each chirality configuration a distinct process and therefore treat each configuration in a helicity/chirality summation as its own subprocess. This occurs due to the definition of chirality states as distinct particles, with unpolarized particles being constructed as multiparticles within the `UFO` model. Thus, what would in MadGraph5 be a sum over helicities is a sum over subprocesses in `CAFE`.

## 4.5 Post-processing: A `HELAS`-like library for chirality flow

In the final stages *before* running the `HELAS`-like code, we need to ascertain that the evaluations will be correctly calculated using chirality flow, i.e. we need to change the so-called vertex functions generated by `ALOHA` with ones from our chirality flow-based `HELAS`-like library. This is done by retroactively replacing the Fortran files generated by `ALOHA` with ones that have been written for this purpose, and occurs in the `export_v4.py` file, during conversion from the `UFO` model to the output `HELAS`-like code. This happens between the

```
      SUBROUTINE LLV1_1(F2, V3, COUP, M1, W1,F1)
      IMPLICIT NONE
      COMPLEX*16 CI
      PARAMETER (CI=(0D0,1D0))
      COMPLEX*16 COUP
      COMPLEX*16 F1(6)
      COMPLEX*16 F2(*)
      REAL*8 M1
      REAL*8 P1(0:3)
      COMPLEX*16 V3(*)
      REAL*8 W1
      COMPLEX*16 DENOM
      COMPLEX*16 INPROD
      F1(1) = +F2(1)+V3(1)
      F1(2) = +F2(2)+V3(2)
      P1(0) = -DBLE(F1(1))
      P1(1) = -DBLE(F1(2))
      P1(2) = -DIMAG(F1(2))
      P1(3) = -DIMAG(F1(1))
      DENOM = COUP/(P1(0)**2-P1(1)**2-P1(2)**2-P1(3)**2)
      INPROD = (F2(3)*V3(3) + F2(4)*V3(4))
      F1(3) = DENOM*CI*INPROD*(V3(5)*((-1)*P1(0) + (-1)*P1(3))
     $ + V3(6)*((-1)*P1(1) + CI*(-1)*P1(2)))
      F1(4) = DENOM*CI*INPROD*(V3(5)*((-1)*P1(1) - CI*(-1)*P1(2))
     $ + V3(6)*((-1)*P1(0) - (-1)*P1(3)))
      F1(5) = 0
      F1(6) = 0
      END
```

**Figure 5.** The `LLV1_1` subroutine provided by the `CAFE HELAS`-like library. This is a numerical evaluation of what is displayed in Figure 4b.

Processing and Evaluation steps shown in Figure 1.

This is done by the function `postex_vertex_replacer` in the `misc.py` file. It searches the working directory for the Lorentz structures used, and depending on the names of these structures calls on a function called `vertex_replacer` to overwrite the contents of the vertex functions with ones from our chirality-flow library.

As an example of this `HELAS`-like library, we show the `LLV1_1` vertex function in Figure 5. The `_1` in the name tells us that it is the first fermion that will be off-shell, i.e. the inflowing one. The first 13 lines define the floating point objects used. The next two lines define `F1(1)` and `F1(2)`, which are the first two components of the six-component complex vector

that the function will output. HELAS routines treat particles as six-component vectors, with the final four components describing the actual physical wavefunction, and the first two the momentum as F1(1)$= p^0 + ip^3$ and F1(2)$= p^1 + ip^2$.

The subsequent lines define the physical momentum of the off-shell fermion. ALOHA-generated libraries always take the momentum to be outgoing. However, the _1 vertex functions actually output an *inflowing* off-shell fermion, so we need to include an additional factor $(-1)$ when considering the particle momentum.

DENOM is the propagator denominator, and INPROD is simply the spinor inner product [ F2 $\text{V3}_L$], where the algorithm detailed in Section 4.1 assures that chirality flow will be matched correctly. We note here that the HELAS library, as well as the HELAS-like libraries generated by ALOHA, do vector and matrix calculations componentwise, which is a convention maintained here. This also means that the additional zeros we end up carrying around are never used in calculations, even if they are stored in memory.

Finally, we calculate the left-chiral spinor stored within the components (F1(3), F1(4)). As $-$P1 is not a massless momentum, the momentum-dot cannot be directly equated to the outer products given by eqs. (2.36,2.37). In order to explicitly take the spinor inner products here, $-$P1 would need to be split into a sum of massless momenta. Within the MadGraph5 framework, particle momenta are stored locally by individual functions, rather than globally, meaning such a splitting would have to be explicitly calculated by CAFE. When compared to evaluating the matrix multiplication between the matrix form of the momentum bispinor (given in eq. (2.25)) and the right-chiral component of V3, the former option was found to require more instructions for this implementation. Thus, in Figure 5 we see instead the componentwise matrix multiplication $\langle V_3| \overline{\not{p}}$. The technical implementation here can thus be seen as a sort of combination of the chirality-flow [6, 7] and Weyl-van-der-Waerden [25,26] formalisms, where chirality-flow diagrams are generated analytically albeit evaluated numerically using the structures of the latter formalism.

## 4.6 Validation

For validation purposes, we treat the matrix element squared of a process, which is calculated with 15 accurate digits. The ratio between ours and the reference matrix element squared calculated should thus differ from 1 at most by $\sim 10^{-15}$. We define the error $\epsilon$,

$$\epsilon = \left| 1 - \frac{|\mathcal{M}(\text{MadGraph5})|^2}{|\mathcal{M}(\text{CAFE})|^2} \right|, \tag{4.5}$$

which should be smaller than $10^{-14}$. This error is shown for a few processes in Table 1. The amplitudes for MadGraph5 and CAFE have been evaluated at phase space points randomly generated using the same seed. As Table 1 shows, all $\epsilon$ calculated are smaller than $10^{-14}$. The differences can be explained entirely by numerical precision.

| | $e^+e^- \to 2\gamma$ | $e^+e^- \to 3\gamma$ | $e^+e^- \to 4\gamma$ |
|---|---|---|---|
| $\epsilon$ | $2.6 \cdot 10^{-15}$ | $5.3 \cdot 10^{-15}$ | $2.7 \cdot 10^{-16}$ |
| | $e^+e^- \to \mu^+\mu^-$ | $e^+e^- \to \mu^+\mu^-\gamma$ | $e^+e^- \to \mu^+\mu^-2\gamma$ |
| $\epsilon$ | $3.0 \cdot 10^{-15}$ | $1.2 \cdot 10^{-15}$ | $1.6 \cdot 10^{-15}$ |
| | $e^-\gamma \to e^-\gamma$ | $e^-\gamma \to e^-2\gamma$ | $e^-\gamma \to e^-3\gamma$ |
| $\epsilon$ | $2.4 \cdot 10^{-15}$ | $2.2 \cdot 10^{-16}$ | $1.1 \cdot 10^{-16}$ |
| | $e^+e^- \to 2\mu^+2\mu^-$ | $e^-\mu^- \to 2e^-\mu^-e^+$ | $e^+e^- \to e^+e^-\mu^+\mu^-$ |
| $\epsilon$ | $4.4 \cdot 10^{-16}$ | $2.1 \cdot 10^{-15}$ | $7.8 \cdot 10^{-16}$ |

Table 1: Magnitude of the difference between the ratio of the amplitudes evaluated using MadGraph5 and `CAFE`, and 1. Numerical errors are expected to be on the order $\epsilon < 10^{-14}$.

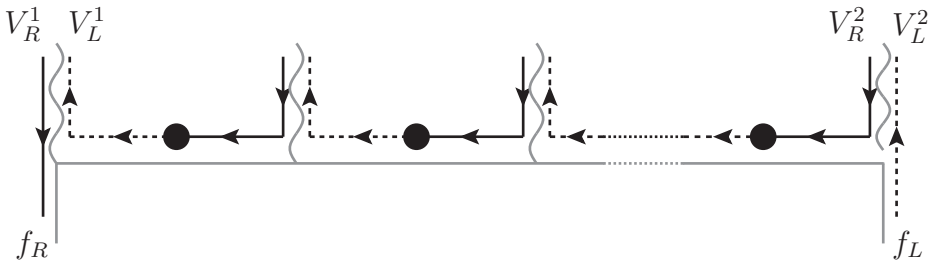
# 5 Results and discussion

Having detailed the implementation of `CAFE` from a practical point of view, we now present both the expected and the measured evaluation time scaling obtained with process complexity in our chirality-flow implementation. As Figure 6 shows, `CAFE` not only evaluates simple processes quicker than MadGraph5 does, but its scaling with external photon multiplicity is also better.

In Section 5.1, the expected efficiency gain from gauge-based diagram removal is explored. Then, in Section 5.2 the practical evaluation speed gain from both simplified Lorentz structures and from gauge-based diagram removal are treated. This explains the differences between the four curves visible in Figure 6. Here, the dash-dotted line depicts evaluation speed gain from performing simplified calculations in a combination of the chirality-flow [6, 7] and Weyl-van-der-Waerden [25, 26] formalisms. The dashed red line shows the combined evaluation speed gain from these simplified calculations and the gauge-based diagram removal described analytically in Section 5.1. At $n = 6$, `CAFE` is roughly a factor 10 times faster than MadGraph5, and the evaluation time ratio keeps increasing thereafter.

## 5.1 Theoretical scaling with chirality flow

Consider the following diagram:




, (5.1)

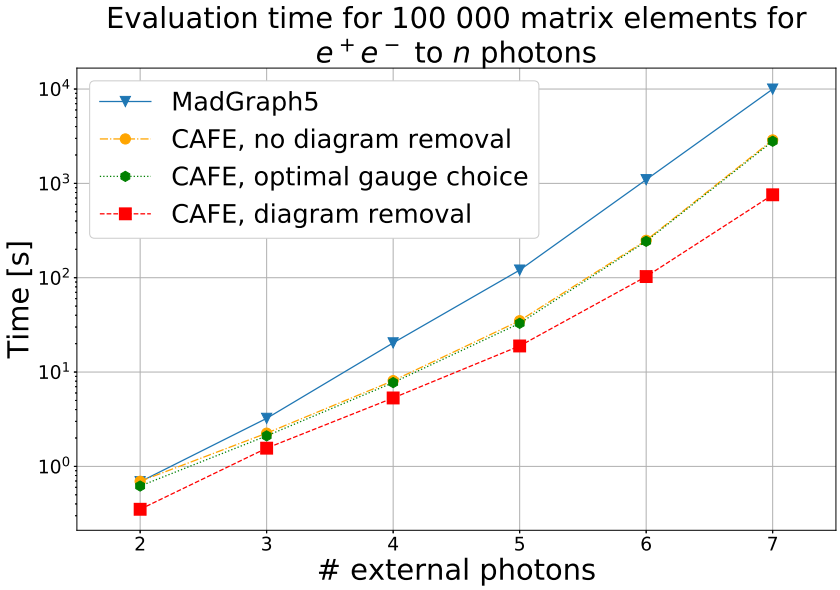



**Figure 6.** Measured runtimes for the evaluation of 100 000 matrix elements for MadGraph5 and several versions of `CAFE`, as a function of photon multiplicity. At $n = 2$, the dashed red line depicting `CAFE` with diagram removal is significantly faster than MadGraph5, and for all $n > 2$, all versions of `CAFE` are faster than MadGraph5. For the region $3 \leq n \leq 5$, the versions of `CAFE` with and without diagram removal grow similarly, but for large photon multiplicities ($n \geq 6$) `CAFE` with diagram removal attains a milder slope than the two versions without it (green dotted and orange dash-dotted lines).

where time moves vertically. Here, we have a fermion pair annihilation resulting in $n$ photons. As the fermions are distinguishable from each other (given that one must be a fermion and the other an antifermion), this process will have $n!$ diagrams, due to the indistinguishable nature of the photons.

We will make a simple gauge choice: The reference momenta for all left-chiral photons are set to be the momentum of the right-chiral fermion, and vice versa for the right-chiral photons with the left-chiral fermion. Whenever a right-chiral photon ends up next to the left-chiral fermion (or vice versa), the chirality-flow diagram will evaluate to zero due to the antisymmetry of the spinor inner products (see eqs. (2.21,2.22). Assume, for example, that the leftmost photon in eq. (5.1) is left-chiral. Then

$$\langle V_R^1| = \langle r_R| = \langle f_R|\,, \tag{5.2}$$

where we by $r_R$ denote the reference momentum for left-chiral photons. Consequently, the leftmost spinor inner product is

$$\langle V_R^1 f_R\rangle = \langle f_R f_R\rangle = 0. \tag{5.3}$$

Similarly, if the rightmost photon is right-chiral we will have that $|V_L^2] = |f_L]$, making the rightmost spinor inner product zero.

Thus, only diagrams with a left-chiral photon next to the left-chiral fermion and vice versa need to be considered. Denote the number of left-chiral and right-chiral photons by $n_L$ and $n_R$, respectively. Then, whenever a left-chiral photon contracts with the right-chiral fermion, there are $(n-1)!$ ways to order the remaining photons, and there will be $n_L$ ways to choose which left-chiral photon is contracted with the right-chiral fermion. Similarly, there are $(n-1)!$ ways to order the remaining photons when a given right-chiral photon is contracted with the left-chiral fermion, yielding in total $n_R(n-1)!$ diagrams where this contraction occurs.

Knowing that one left-chiral photon is contracted to the right-chiral fermion and one right-chiral photon is contracted to the left-chiral fermion, the remaining photons can be ordered in $(n-2)!$ ways. There are $n_L n_R$ ways to choose one photon of each chirality. Hence, with this gauge choice, we need to evaluate $n_L n_R (n-2)!$ diagrams, instead of the $n!$ we would need to otherwise.

If instead of chirality sampling we perform chirality summation, we find that the number of diagrams evaluated by a chirality flow-based method with this gauge choice is given by

$$\begin{aligned}\sum_{n_L=0}^{n} n_L(n-n_L)(n-2)! &= (n-2)!\left\{n\sum_{n_L=0}^{n} n_L - \sum_{n_L=0}^{n} n_L^2\right\} \\ = (n-2)!\left\{n\frac{n(n+1)}{2} - \frac{n(n+1)(2n+1)}{6}\right\} &= \frac{(n+1)!}{6},\end{aligned} \tag{5.4}$$

assuming a given helicity configuration for the fermions. Evaluating both the options $f_L^+ f_R^-$ and $f_R^+ f_L^-$ yields an additional factor 2 in the expression above.

Now consider the number of diagrams MadGraph5 will evaluate, summing over helicity configurations. Although MadGraph5 has a built-in zero detection system to discard any helicity configurations with zero amplitude, this does not discard all same-helicity photon diagrams[3]. It does, however, discard all same-helicity fermion configurations. As such, for the helicity summed process $f^+f^- \to n\gamma$, MadGraph5 will evaluate $n!$ diagrams with $2 \cdot 2^n$ helicity configurations , giving $2^{n+1}n!$ diagrams in total.

In comparison to MadGraph5 we expect `CAFE` with diagram removal to scale approximately as

$$\frac{\text{Scaling(MadGraph5)}}{\text{Scaling(CAFE)}} \propto \frac{2^n n!}{(n+1)!} = \frac{2^n}{n+1}, \tag{5.5}$$

that is, as the photon multiplicity increases, we expect a helicity summed chirality flow based method with this gauge choice to perform exponentially faster than MadGraph5.

[3]Using MadGraph5_aMC@NLO version 3.2.0, released on August 22nd, 2021.

| | FFV1_0 | FFV1_1 | FFV1_2 | FFV1P0_3 | IXXXXX | OXXXXX | VXXXXX | |
|---|---|---|---|---|---|---|---|---|
| Instr. | 416 | 2108 | 2136 | 627 | 179 | 174 | 163 | |
| Instr. | 122 | 333 | 321 | 245 | 155 | 153 | 359 | 342 |
| | LRV1_0 | LLV1_1 | LLV1_2 | LRV1P0_3 | LXXXXX | RXXXXX | VLXXXX | VRXXXX |

Table 2: Comparison of number of instructions for Lorentz structures and vertices used in calculations for the HELAS-like libraries used by MadGraph5 and CAFE.

We note that this gauge choice is sufficient to optimize calculations for processes with a single fermion line. For more complex processes, however, diagrams can be split into independently gauge invariant subsets. Future versions of CAFE or its successors may implement a smart algorithm to choose gauge independently for these subsets, to minimize the number of diagrams that need to be evaluated for arbitrarily complex processes.

## 5.2 Evaluation time scaling in CAFE

Finally, we detail the evaluation time comparisons of CAFE and MadGraph5. To study how CAFE's evaluation time scales with the complexity of a diagram, we take the example from Section 5.1 of electron pair annihilation going to $n$ photons. Based on the evaluation time for a series of values of $n$, we compare MadGraph5 and CAFE.

CAFE involves two independent sources of increased efficiency, which we consider separately. The first one is the simplified vertex functions used by our HELAS-like library, and the second is the diagram removal allowing CAFE to evaluate fewer diagrams than MadGraph5. These are detailed in Sections 5.2.1 and 5.2.2 respectively.

### 5.2.1 Simplified calculations

The HELAS-like library developed for CAFE has three types of Lorentz structures. These are the amplitude calculation functions LRV1_0/RLV1_0; the off-shell fermions LLV1_1, LLV1_2, RRV1_1, and RRV1_2; and the off-shell photons, LRV1P0_3/RLV1P0_3. We compare the complexity of these with the corresponding Lorentz structures from the SM UFO model; FFV1_0, FFV1_1, FFV1_2, and FFV1P0_3. This complexity is measured by the number of instructions each such subroutine makes, as determined using the Callgrind [27] tool for the Valgrind [28] dynamic binary instruction framework, and is depicted in Table 2.

Table 2 shows that CAFE's vertex functions are far simpler than those in other HELAS-like libraries. Only four of the eight structures used by CAFE are described, but the corresponding ones under a shift L↔R are similar to those shown, differing only in single digits.

Additionally, Table 2 details the difference in instructions in external wavefunction routines. Very little gain is observed for CAFE's external fermions, and a significant loss is seen for CAFE's external vector bosons. These facts are expected: External fermions are

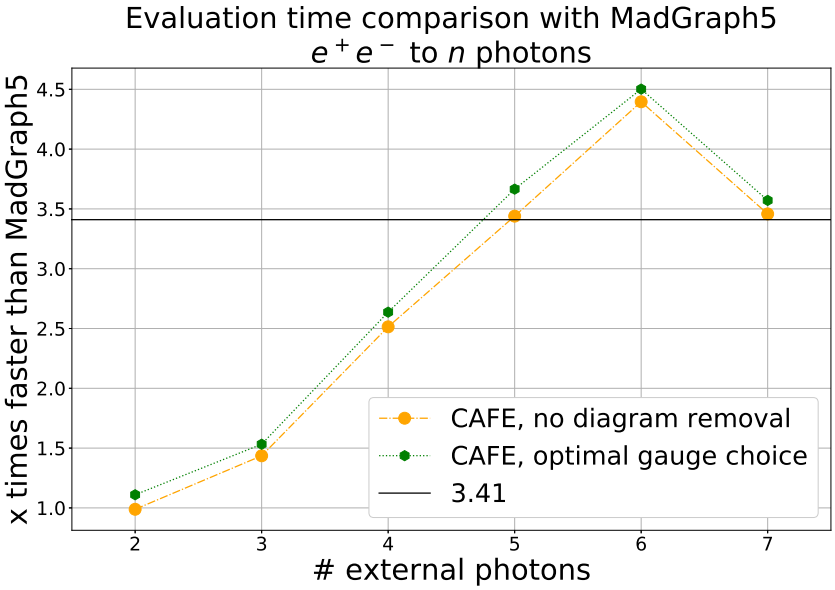



**Figure 7.** Comparison in relative evaluation times between `CAFE` without diagram removal with and without optimal gauge choice. The dash-dotted line performs the same number of evaluations as MadGraph5, whereas the gauge choice in the dotted line makes use of the zero detection routine in MadGraph5 in a way standard MadGraph5 does not, such that diagrams where all photons have the same chirality are discarded quicker than for the dash-dotted line. In comparison, the dash-dotted curve uses the opposite gauge choice of the one presented in Section 5.1: the reference momentum for the left-chiral photons is that of the first left-chiral fermion, and vice versa for the right-chiral photons and fermions.

implemented similarly between the two programs and there should be little difference, but `CAFE` needs to calculate two spinors per external photon (each of which can be no simpler than an external fermion). A lower limit on photon instructions is then two times that of a fermion, which is similar to the values observed in Table 2. However, as the complexity of a process increases, so should the amount of time spent in the vertex functions, and we expect the added complexity in external wavefunction routines to be outweighed by the simplified vertices.

We note that the intentional recycling of currents used by `HELAS` routines means that as process complexity increases, the evaluation time should be dominated by `FFV1_0` and `LLV1_0`/`RRV1_0` routines used for evaluating diagrams, which are used once per diagram (unlike off-shell currents, as described in Section 3.3). We expect the ratio in evaluation time between MadGraph5 and `CAFE` *without diagram removal* to approach some 'simplicity constant' $C$ as the complexity of the process increases, with this constant given by

$$C = \frac{\text{Instructions}(\texttt{FFV1\_0})}{\text{Instructions}(\texttt{LLV1\_0})} = \frac{416}{122} = 3.41, \tag{5.6}$$

and with some systematic bias in the favor of MadGraph5 because CAFE will call on the RAMBO algorithm [24] to generate random phase space points individually for each chirality configuration, whereas MadGraph5 does this outside of helicity summation. This mainly affects cases with few external partons, where RAMBO dominates the evaluation time, as has been checked with Valgrind. In fact, at $n = 2$ RAMBO takes up more than half of the program instructions for each of CAFE's subprocesses.

This falls in line with what we see in Figure 7. Here, the two curves evolve comparably, starting at similar evaluation times to MadGraph5 before becoming approximately linearly faster than MadGraph5 with external photon multiplicity. Eventually, the curves reach a maximum at $n = 6$ after which they decrease, ending close to 3.41 for $n = 7$, consistent with our expectation. As Table 2 shows, the biggest ratio in instructions between MadGraph5 and CAFE is the one between the off-shell fermion functions FFV1_1/2 and LLV1_1/2. We expect a maximum in evaluation speed gain when the percentage of function calls that are fermion propagators is at its highest. This should occur somewhere between the regime where function calls are dominated by RAMBO and by LLV1_0/RRV1_0. The dash-dotted line in Figure 7 performs the same number of amplitude evaluations as MadGraph5 does, due to a suboptimal gauge choice, whereas the dotted line makes use of the zero-detection routine within MadGraph5 to avoid evaluating subprocesses where all external photons have the same chirality. The dash-dotted green curve does, however, still perform phase space point generation for every matrix element, even if no calculations are performed.

### 5.2.2 Gauge-based diagram removal

Now, we turn to the effects of gauge-based diagram removal. This effect is explored by comparing the evaluation times in MadGraph5 and CAFE for 100 000 randomly generated matrix elements (using the same seed for all runs, so the same 100 000 phase space points are considered). This is shown in Figure 6, where these runtimes are displayed alongside the real-time evaluations of the green and orange curves in Figure 7. It is apparent that for $n \geq 6$, CAFE evaluates these matrix elements roughly an order of magnitude faster than MadGraph5.

Figure 6 does not, however, show the gain specifically from diagram removal: CAFE with diagram removal will evaluate the matrix elements using our simplified HELAS-like library. Thus, comparing MadGraph5 and CAFE directly would depict the effects of both gauge-based diagram removal and simplified calculations simultaneously. A more fair comparison for separating these effects is to compare CAFE with diagram removal and the orange curve in Figure 7, as there we perform approximately the same number of evaluations as MadGraph5, but using our vertex functions. This comparison is shown in Figure 8.

For small $n$ the RAMBO algorithm [24] takes up a considerable amount of the runtime, which is not compensated for in these comparisons. CAFE performs phase space point generation independently for each subprocess (chirality configuration). From the CAFE UFO

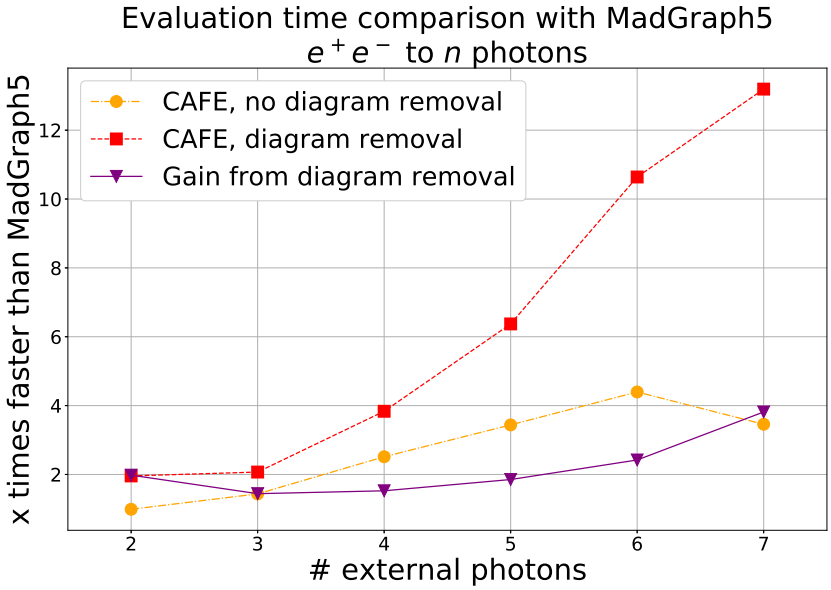



**Figure 8.** Comparison in relative runtime between several instances of `CAFE` with different features implemented, normalized to MadGraph5's runtimes for the same processes. Note that the $y$-axis labels how many times faster a process is evaluated by `CAFE` than by MadGraph5, making the curves themselves inversely proportional to program runtime.

model, a left-chiral electron must annihilate against a right-chiral positron, and vice versa. Thus, given a valid electron-positron configuration, the only chirality configurations which have zero amplitude are those where all photons have the same chirality. Regardless of the number of external photons $n$, this will always be two, whereas the total number of chirality configurations will be $2^n$. As $n \to \infty$, the ratio between the numbers of treated configurations will tend to one, but for $n = 2$ this means running $2^2 = 4$ versus $2^2 - 2 = 2$ processes. Considering `RAMBO` takes up about half of the runtime for each subprocess here, it is clear that for small $n$ the main difference lies in overhead from running different numbers of processes, rather than any gain from evaluating fewer diagrams in each subprocess.

# 6 Conclusion and outlook

Tools for matrix element evaluation are central to particle physics. As our experimental understanding of the processes in the Standard Model increases, the need for analysis of more and more complex processes grows. To compare experiments and theory, tools that can evaluate theoretical values for increasingly complex processes are necessary.

We present results using the chirality-flow formalism [6, 7] to simplify calculations, as well as to determine reference vectors of external photons in order to discard diagrams which do not contribute to the full amplitude of a process. The implemented Chiral Automized Flow

Extension for MadGraph5_aMC@NLO, CAFE, can at present only evaluate massless QED processes at tree-level, but for these Standard Model processes its evaluation time has been shown to increase at a slower rate with process complexity than MadGraph5. This is both due to simplification of individual function calls and a reduction of total number of diagrams evaluated.

Although CAFE at present only works with massless tree-level QED, which does not constitute a major bottleneck for particle physics, the chirality-flow formalism has been extended to the full tree-level Standard Model [6, 7]. A complete Standard Model implementation may prove to take considerable effort, but we expect most technical difficulties to have been faced or at the very least observed during the course of this implementation. It has not yet been shown how well the evaluation speed increase shown for massless QED translates to these other sectors, but given the success here we are optimistic.

While this project has considered an implementation of chirality flow which is structurally as similar to MadGraph5 as possible, to make comparison between the two simple, the underlying HELAS-like routines are unfit for chirality flow. HELAS, and consequently ALOHA, were constructed with the intention of recycling currents. Within the chirality-flow formalism, however, it would be more natural to store and reuse spinor inner products, which are scalars. A more thorough implementation could detect the start- and endpoints of chirality-flow lines in order to store spinor inner products, rather than using the unsuited vertex function-structure implemented here. The effects such a structure would have on amplitude evaluation time has not been estimated, but we expect that caching a small number of scalars should make computations considerably faster.

Regardless of the future development of CAFE, it is clear from these results that amplitude calculators and event generators should take the mathematical formalism under which their evaluations are made into account. As we have shown here, chirality flow not only simplifies calculations, but has the benefit of making practical gauge choices easy to determine, allowing for techniques such as the gauge-based diagram removal implemented here. As we head into the uncharted territories of the Standard Model, such gains may end up being more relevant than ever before.

# Acknowledgments

The author extends their utmost gratitude to: Malin Sjödahl and Andrew Lifson, both for the possibility of entertaining such an interesting thesis project and for the supervision they have given, making this project possible, and to: Olivier Mattelaer, whose elaborations on the functionality and structure of MadGraph5_aMC@NLO, ALOHA, and the UFO format has made this project far more accessible than it might otherwise have been.

# References


[1] P. De Causmaecker, R. Gastmans, W. Troost, and Tai Tsun Wu. Multiple Bremsstrahlung in Gauge Theories at High-Energies. 1. General Formalism for Quantum Electrodynamics. *Nucl. Phys. B*, 206:53–60, 1982.

[2] F.A. Berends, R. Kleiss, P. De Causmaecker, R. Gastmans, W. Troost, and Tai Tsun Wu. Multiple Bremsstrahlung in Gauge Theories at High-Energies. 2. Single Bremsstrahlung. *Nuclear Physics B*, 206(1):61–89, 1982.

[3] Zhan Xu, Da-Hua Zhang, and Lee Chang. Helicity Amplitudes for Multiple Bremsstrahlung in Massless Nonabelian Gauge Theories. *Nucl. Phys. B*, 291:392–428, 1987.

[4] R. Gastmans and T. T. Wu. *The Ubiquitous Photon: Helicity methods for QED and QCD*, volume 80. Clarendon Press, 1990.

[5] H.F Jones. *Groups, Representations and Physics*. Taylor & Francis, 2 edition, 1998.

[6] Andrew Lifson, Christian Reuschle, and Malin Sjodahl. The chirality-flow formalism. *The European Physical Journal C*, 80(11), Oct 2020.

[7] Joakim Alnefjord, Andrew Lifson, Christian Reuschle, and Malin Sjodahl. The chirality-flow formalism for the standard model. *The European Physical Journal C*, 81(4), Apr 2021.

[8] J. Alwall, R. Frederix, S. Frixione, V. Hirschi, F. Maltoni, O. Mattelaer, H.-S. Shao, T. Stelzer, P. Torrielli, and M. Zaro. The automated computation of tree-level and next-to-leading order differential cross sections, and their matching to parton shower simulations. *Journal of High Energy Physics*, 2014(7), Jul 2014.

[9] T. Stelzer and W.F. Long. Automatic generation of tree level helicity amplitudes. *Computer Physics Communications*, 81(3):357–371, Jul 1994.

[10] H. Murayama, I. Watanabe, and K. Hagiwara. HELAS: HELicity amplitude subroutines for Feynman diagram evaluations. Technical Report KEK-91-11, National Lab. for High Energy Physics, Tsukuba, Ibaraki, 1998.

[11] Céline Degrande, Claude Duhr, Benjamin Fuks, David Grellscheid, Olivier Mattelaer, and Thomas Reiter. UFO – The Universal FeynRules Output. *Computer Physics Communications*, 183(6):1201–1214, Jun 2012.

[12] Brian C Hall. *Lie groups, Lie algebras, and representations: An elementary introduction*. Graduate Texts in Mathematics 222. Springer, 2 edition, 2015.

[13] Dan V. Schroeder Michael E. Peskin. *An Introduction to Quantum Field Theory*. Frontiers in Physics. Addison-Wesley Pub. Co, 1995.

[14] Lance J. Dixon. A brief introduction to modern amplitude methods. In *Theoretical Advanced Study Institute in Elementary Particle Physics: Particle Physics: The Higgs Boson and Beyond*, pages 31–67, 2014.

[15] Henriette Elvang and Yu tin Huang. Scattering amplitudes. arXiv:1308.1697, 2014.

[16] Nima Arkani-Hamed, Tzu-Chen Huang, and Yu tin Huang. Scattering amplitudes for all masses and spins. arXiv:1709.04891, 2021.

[17] Cliffor Cheung. TASI Lectures on Scattering Amplitudes. In Rouven Essig and Ian Low, editors, *Anticipating the Next Discoveries in Particle Physics*, pages 571–623. WORLD SCIENTIFIC, 2018.

[18] G.'t Hooft. A planar diagram theory for strong interactions. *Nuclear Physics B*, 72(3):461–473, 1974.

[19] F. Maltoni, K. Paul, T. Stelzer, and S. Willenbrock. Color-flow decomposition of QCD amplitudes. *Phys. Rev. D*, 67:014026, Jan 2003.

[20] W. Kilian, T. Ohl, J. Reuter, and C. Speckner. Qcd in the color-flow representation. *Journal of High Energy Physics*, 2012(10):22, Oct 2012.

[21] Johan Alwall, Michel Herquet, Fabio Maltoni, Olivier Mattelaer, and Tim Stelzer. MadGraph 5: going beyond. *Journal of High Energy Physics*, 2011(6), Jun 2011.

[22] Priscila de Aquino, William Link, Fabio Maltoni, Olivier Mattelaer, and Tim Stelzer. ALOHA: Automatic libraries of helicity amplitudes for Feynman diagram computations. *Computer Physics Communications*, 183(10):2254–2263, Oct 2012.

[23] Michel Herquet and MadTeam. MadGraph5_aMC@NLO in Launchpad. `https://launchpad.net/mg5amcnlo`, Sep 2009.

[24] R Kleiss, W.J Stirling, and S.D Ellis. A new Monte Carlo treatment of multiparticle phase space at high energies. *Computer Physics Communications*, 40(2):359–373, 1986.

[25] H. Weyl. *The Theory of Groups and Quantum Mechanics*. Dover Books on Mathematics. Dover Publications, 1950.

[26] B.L. van der Waerden. *Group Theory and Quantum Mechanics*. Grundlehren der mathematischen Wissenschaften. Springer Berlin Heidelberg, 2012.

[27] Josef Weidendorfer, Markus Kowarschik, and Carsten Trinitis. A Tool Suite for Simulation Based Analysis of Memory Access Behavior. *Proceedings of 4th International Conference on Computation Science*, 3038:440–447, 06 2004.

[28] Nicholas Nethercote and Julian Seward. Valgrind: A framework for heavyweight dynamic binary instrumentation. *Sigplan Notices - SIGPLAN*, 42:89–100, 06 2007.