Numerical Integration of Tree Level Scattering Amplitudes

A thesis presented for the degree of

Master of Science in Physics

by

Dario Kermanschah

Department of Physics
ETH Zurich

April 2018

Supervisors:
Dr. Valentin Hirschi
Prof. Charalampos Anastasiou

Abstract

The numerical calculation of cross sections usually demands for the application of specialized techniques. In this thesis, we present two general methods that improve the efficiency of Monte Carlo simulations. These two methods, importance sampling and the multi-channel approach, are implemented and tested with tree level amplitudes. By importance sampling, we denote the transformation of phase space variables such that it cancels propagator structures of single topologies. When dealing with amplitudes composed of multiple topologies, we make use of the multi-channel approach. These amplitudes are divided into channels, where each channel has a distinct peak structure that is flattened by importance sampling. Finally, we show how unknown and specific features of scattering amplitudes are accounted for adaptively.

Contents

Chapter 1

Introduction and Context

The step towards new discoveries at colliders such at LHC not only requires refined experimental procedures but also sophisticated theoretical predictions that accurately cover signal and background. At this level of accuracy, the predictions often rely on numerical methods. Monte Carlo simulations are well suited for this purpose. They can be exploited to produce physical events that are comparable to the ones that are directly observed at colliders. By combining these events it is possible to predict observables in equal fashion as in experiment, where both, the prediction and the experiment are affected by uncertainties.

Along the way from a random Monte Carlo point to a physical event, one has to overcome several obstacles. The predictions are based on a model, which specifies the particles and how they interact with each other. It defines the possible Feynman diagrams for a process of interest. The scattering amplitude, which describes the transition probability from an initial to a final state is then obtained from all diagrams and their interferences. To determine the likelihood of the scattering process, expressed by the cross section, one combines the amplitude contributions of the allowed final states. In this thesis, we investigate the numerical computation of cross sections of tree level processes at the parton level.[1] To finally predict events that are directly comparable with experimental results however, one additionally has to consider the process of hadronization and on top of that, simulate the effects of the detectors.

In the following chapters, we describe how cross sections are characterized at parton level and how they can be computed efficiently with numerical integration. In a sophisticated Monte Carlo simulation an effort has to be made to improve the efficiency of the integration. For this sake, we exploit our knowledge about matrix elements and the structure of tree level diagrams. This leads to a splitting into multiple integrands, described by the multi-channel integration technique [1], where the variance of each channel is reduced by a method called importance sampling. We implemented this approach within the framework of `MadGraph5_aMC@NLO` (`MG5aMC`) that provides numerous tools and inputs, such as the model, the matrix elements and a user interface [2, 3].

[1] More precise predictions have to take higher order corrections into account.

Note that MG5aMC is already able to apply similar strategies that improve the efficiency of the computation of cross sections. They however lack flexibility and were not designed for more ambitious studies at next to leading order or higher. Therefore, it is necessary to improve and realize these methods for further development of MG5aMC. Also, RacoonWW applies these strategies, with a focus on lepton production at electron colliders [4]. We aim for a fully automatic and generic implementation that is applicable to arbitrary particle collisions.

1.1 Model and cross section

The dynamics of the elementary particles and their interactions are described by the standard model of particle physics. It is formulated in the framework of quantum field theory in terms of a Lagrangian density that encodes the particles' parameters, their degrees of freedom and their interactions.

The transition from one quantum state to another is described by the scattering matrix. The matrix element is calculated from time-ordered correlation functions, which are represented graphically by Feynman diagrams and are usually averaged over spin and color degrees of freedom. The Feynman diagrams are obtained by applying a set of rules that are derived from the Lagrangian. In general, these rules rely on renormalization conditions and are gauge dependent.

The rate of occurrence of a scattering process is characterized by the cross section σ. It represents the probability that a particular process will take place and is the connecting element between experiment and theory. The cross section is directly related to the transition amplitude and to the matrix element. In the case of two colliding particles with momenta $p_\pm$ and n outgoing particles with momenta k_j, the matrix element $\mathcal{M}$ is defined through the scattering matrix $\mathcal{S}$ as

$$\langle k_1,\dots,k_n|\mathcal{S}-\mathbb{1}|p_+,p_-\rangle = i(2\pi)^4\,\delta^4(p_+ + p_- - \textstyle\sum_j k_j)\,\mathcal{M}(p_+,p_-,k_1,\dots,k_n). \tag{1.1}$$

At a given order, the matrix element can be computed with a set of m Feynman diagrams A_i as

$$\mathcal{M} = \left|\sum_{i=1}^{m} A_i\right|^2 . \tag{1.2}$$

In the rest frame of $p \equiv p_+ + p_-$, with $p^2 \equiv s$, the cross section is calculated as follows

$$\mathrm{d}\sigma = \frac{(2\pi)^{4-3n}}{2\,\lambda^{\frac{1}{2}}\left(s,{p_+}^2,{p_-}^2\right)}\left|\mathcal{M}\left(p_+,p_-,k_1,\dots,k_n\right)\right|^2\,\mathrm{d}\Phi_n, \tag{1.3}$$

with the Källén function $\lambda(x,y,z) = x^2+y^2+z^2-2xy-2xz-2yz$ and the Lorentz-invariant n-body phase space

$$\mathrm{d}\Phi_n = \left(\prod_j \mathrm{d}^4k_j\,\delta\left({k_j}^2-s_j\right)\,\Theta\left(k_j^0\right)\right)\delta^4\left(p_+ + p_- - \textstyle\sum_j k_j\right), \tag{1.4}$$

with the Heaviside step function Θ that guarantees that the final particles propagate with positive energy. The Dirac delta functions ensure momentum conservation and specify the mass shell conditions of particles j with squared masses s_j.

In the case of colliding hadrons, the total cross section can be related to the cross section $\hat{\sigma}_{ab}$ of the colliding point-like constituents a and b with the parton distribution function f_a and f_b, respectively. It is defined in the infinite momentum frame as

$$\sigma = \int_0^1 \mathrm{d}x_1 \int_0^1 \mathrm{d}x_2 \sum_{a,b} f_a(x_1) f_b(x_2)\, \hat{\sigma}_{ab}(x_1 p_+, x_2 p_-), \tag{1.5}$$

where the interacting constituents a and b are massless particles that carry a fraction of their hadron's momentum $p_\pm$, denoted by the Bjorken-variables x_1 and x_2, respectively.

In general, the cross section is therefore written as

$$\sigma = \int \mathrm{d}x_1\, \mathrm{d}x_2\, \mathrm{d}\Phi_n \sum_{ab} f_a(x_1) f_b(x_2)\, \frac{(2\pi)^{4-3n} \left|\mathcal{M}_{ab}\left(x_1 p_+, x_2 p_-, k_1, \ldots, k_n\right)\right|^2}{2\,\lambda^{\frac{1}{2}}\left(\hat{s}, \left(x_1 p_+\right)^2, \left(x_2 p_-\right)^2\right)}, \tag{1.6}$$

an integral over the Bjorken-variables and the phase space with squared center of mass energy $\hat{s} \equiv x_1 x_2 s$ of the interacting sub-system. It is parametrized by a set of phase space variables.

Chapter 2

Phase Space Parametrization

The phase space is the space of all possible kinematic configurations of the final states of a process, characterized by a set of four-momenta. It can be systematically parametrized by Mandelstam invariants and angles [5, 6, 7] as we show in Section 2.2. The initial momenta are fixed by the laboratory frame of reference. However, in hadronic collisions the colliding partons carry only a fraction of the incoming momenta, described by the Bjorken variables x_1 and x_2 as additional integration variables. In the next section, we present an alternative parameterization.

2.1 Bjorken variables

The integration over the Bjorken variables x_1 and x_2 is usually parametrized by the variables τ and η, defined as

$$\tau = x_1 x_2 \qquad \text{and} \qquad \eta = \frac{1}{2} \ln \frac{x_1}{x_2}, \tag{2.1}$$

or equivalently

$$x_1 = \sqrt{\tau} \exp \eta \qquad \text{and} \qquad x_2 = \sqrt{\tau} \exp -\eta. \tag{2.2}$$

The Jacobian of the transformation is

$$J = \begin{pmatrix} \frac{\partial \tau}{\partial x_1} & \frac{\partial \tau}{\partial x_2} \\ \frac{\partial \eta}{\partial x_1} & \frac{\partial \eta}{\partial x_1} \end{pmatrix} = \begin{pmatrix} x_2 & x_1 \\ \frac{1}{2}\frac{1}{x_1} & -\frac{1}{2}\frac{1}{x_2} \end{pmatrix}, \tag{2.3}$$

with $|\det J\,| = 1$. The integral then transforms as

$$\int_0^1 \mathrm{d}x_1 \int_0^1 \mathrm{d}x_2 \longrightarrow \int_0^1 \mathrm{d}\tau \int_{\frac{1}{2}\ln\tau}^{-\frac{1}{2}\ln\tau} \mathrm{d}\eta, \tag{2.4}$$

where the bounds for η can be derived from the relations

$$\eta = \frac{1}{2} \ln \frac{{x_1}^2}{\tau} \leq -\frac{1}{2} \ln \tau \qquad \text{and} \qquad \eta = \frac{1}{2} \ln \frac{\tau}{{x_2}^2} \geq \frac{1}{2} \ln \tau. \tag{2.5}$$

This change of variables allows to easily apply physical integration cuts and aligns the variable τ with a propagator in s-channel diagrams, since $\hat{s} = \tau s$.

2.2 Factorization and momentum configuration

The phase space integral can be systematically factorized into building blocks and parametrized by a set of integration variables that consists of Mandelstam invariants and angles [5, 6, 7]. Every Feynman diagram suggests such a decomposition but does not determine it uniquely.

Concretely, the n-body phase space Φ_n of dimension $3n-4$ can be decomposed into $2 \to 2$ scattering processes and $1 \to 2$ decays, denoted by $\Phi^j_{2\to 2}$ and $\Phi^k_{1\to 2}$ respectively, both two-dimensional phase spaces. A number κ of $2 \to 2$ scattering processes creates a maximum number of $\kappa+1$ final states. Therefore, κ can range from 1 to $n-1$. For a given κ, the remaining $n-1-\kappa$ final states are created through $1 \to 2$ decays.

A $2 \to 2$ scattering process and a $1 \to 2$ decay define one time-like invariant each. One of them is fixed by overall momentum conservation but the remaining $n-2$ invariants s_i are integrated over.

In general, the phase space factorization reads

$$\int \mathrm{d}\Phi_n = \prod_{i=1}^{n-2} \int\limits_{s_{i,\min}}^{s_{i,\max}} \mathrm{d}s_i \prod_{j=1}^{\kappa} \int \mathrm{d}\Phi^j_{2\to 2} \prod_{k=1}^{n-1-\kappa} \int \mathrm{d}\Phi^k_{1\to 2}, \tag{2.6}$$

where the integration boundaries for the invariants are given by $s_{i,\min} = (m_1 + m_2)^2$, guaranteeing that the two subsequent states reach their mass shell at invariant masses $m_{1/2}$, and $s_{i,\max} = (\sqrt{s} - \sum_i m_i)^2$, ensuring that the subsequent processes do not exceed the total center of mass energy $\sqrt{s}$ minus the minimum energies of the parallel process blocks $\sum_i m_i$.

2.2.1 Two-body phase space

For both cases, the $2 \to 2$ scattering processes and $1 \to 2$ decays, the phase space can be parametrized in equal fashion. It is convenient to perform the calculation in the rest frame of the incoming momentum p, with $p^2 \equiv s$. In a $2 \to 2$ scattering process $p = p_+ + p_-$, with $p_\pm$ the momenta of the two colliding particles. In a $1 \to 2$ decay, p is the momentum of the decaying particle. The outgoing momenta k_1 and k_2 will then be back to back along a direction in space determined by the azimuthal and polar angle φ and θ. The mass shell conditions are determined by the invariants s_1 and s_2. Concretely, we write

$$\begin{aligned}\int \mathrm{d}\Phi_2 &= \int \mathrm{d}^4 k_1\, \mathrm{d}^4 k_2\, \delta\left({k_1}^2 - s_1\right) \Theta\left(k_1^0\right) \delta\left({k_2}^2 - s_2\right) \Theta\left(k_2^0\right) \delta^4\left(p - k_1 - k_2\right) \\ &= \int \mathrm{d}^3\vec{k}_1\, \mathrm{d}^3\vec{k}_2\, \mathrm{d}k_1^0\, \mathrm{d}k_2^0\, \frac{\delta\left(k_1^0 - \sqrt{|\vec{k}_1|^2 + s_1}\right)}{2k_1^0} \frac{\delta\left(k_2^0 - \sqrt{|\vec{k}_2|^2 + s_2}\right)}{2k_2^0} \\ &\quad \times \delta^3\left(\vec{k}_1 + \vec{k}_2\right) \delta\left(\sqrt{s} - k_1^0 - k_2^0\right),\end{aligned} \tag{2.7}$$

where we accounted for the Heaviside functions. The mass shell conditions and momentum conservation are resolved as

$$= \int \frac{\mathrm{d}^3\vec{k}}{2\sqrt{|\vec{k}|^2+s_1} \times 2\sqrt{|\vec{k}|^2+s_2}}\, \delta\left(\sqrt{s} - \sqrt{|\vec{k}|^2+s_1} - \sqrt{|\vec{k}|^2+s_2}\right). \tag{2.8}$$

Next, we switch to spherical coordinates and solve the remaining delta function that imposes energy conservation as follows

$$\begin{aligned}
&= \int \frac{|\vec{k}|^2\,\mathrm{d}|\vec{k}|\,\mathrm{d}\cos\theta\,\mathrm{d}\varphi}{4\sqrt{|\vec{k}|^2+s_1}\sqrt{|\vec{k}|^2+s_2}}\, \frac{\delta\left(|\vec{k}| - \frac{\lambda^{\frac{1}{2}}(s,s_1,s_2)}{2\sqrt{s}}\right)}{|\vec{k}|\left(\frac{1}{\sqrt{|\vec{k}|^2+s_1}} + \frac{1}{\sqrt{|\vec{k}|^2+s_2}}\right)} \\
&= \int \mathrm{d}\cos\theta\,\mathrm{d}\varphi\, \frac{|\vec{k}|\,\mathrm{d}|\vec{k}|}{4\left(\sqrt{|\vec{k}|^2+s_1}+\sqrt{|\vec{k}|^2+s_2}\right)}\, \delta\left(|\vec{k}| - \frac{\lambda^{\frac{1}{2}}\left(s,s_1,s_2\right)}{2\sqrt{s}}\right) \qquad (2.9) \\
&= \frac{\lambda^{\frac{1}{2}}\left(s,s_1,s_2\right)}{8s} \int_{-1}^{1} \mathrm{d}\cos\theta \int_{0}^{2\pi} \mathrm{d}\varphi.
\end{aligned}$$

From this simple results, we find that the two-body phase space is parametrized by two phase space variables, the azimuthal and polar angle φ and θ. The conditions, implied by the Dirac delta functions motivate the parametrization of the momenta of the outgoing particles in the rest frame of p as

$$k_1^* = \mathcal{R}(\varphi^*, \cos\theta^*) \begin{pmatrix} \frac{s+s_1-s_2}{2\sqrt{s}} \\ 0 \\ 0 \\ \frac{\lambda^{\frac{1}{2}}(s,s_1,s_2)}{2\sqrt{s}} \end{pmatrix} \qquad \text{and} \qquad k_2 = p - k_1, \tag{2.10}$$

with the matrix

$$\mathcal{R}\left(\varphi^*, \cos\theta^*\right) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta^*\cos\varphi^* & -\sin\varphi^* & \sin\theta^*\cos\varphi^* \\ 0 & \cos\theta^*\sin\varphi^* & \cos\varphi^* & \sin\theta^*\sin\varphi^* \\ 0 & -\sin\theta^* & 0 & \cos\theta^* \end{pmatrix}, \tag{2.11}$$

rotating the space components of a four-vector by θ^* around the y-axis, followed by a rotation by φ^* around the z-axis.

2.2.2 The t-channel two-body scattering process

In a 2 → 2 scattering process, the rest frame of p has additional rotational degrees of freedom that allow to fix the orientation of the incoming momenta. They are conveniently chosen to be along the z-axis, parametrized as

$$p_\pm = \begin{pmatrix} \frac{s+p_\pm^2-p_\mp^2}{2\sqrt{s}} \\ 0 \\ 0 \\ \pm\frac{\lambda^{\frac{1}{2}}(s,p_+^2,p_-^2)}{2\sqrt{s}} \end{pmatrix}. \tag{2.12}$$

In the case where the phase space is decomposed in not more than one 2 → 2 scattering process ($\kappa = 1$), it can involve either a t-channel or an s-channel propagator.[1] In any other case ($\kappa \geq 2$) the 2 → 2 scattering processes involve only t-channel propagators.

Whenever a t-channel propagator is involved, it is convenient to align the integration with the Mandelstam invariant t (or u) with the following change of variables. In the rest frame of p, the Lorentz-invariant quantity $t = (p_+ - k_1)^2$ can be written as

$$t = s_1 + p_+^2 - \frac{(s+s_1-s_2)(s+p_+^2-p_-^2) - \lambda^{\frac{1}{2}}(s,s_1,s_2)\lambda^{\frac{1}{2}}(s,p_+^2,p_-^2)\cos\theta^*}{2s}, \tag{2.13}$$

with a dependence on $\cos\theta^*$ only. Therefore, the phase space integral can be aligned with t,

$$\int \mathrm{d}\Phi_2 = \frac{1}{4\lambda^{\frac{1}{2}}(s,p_+^2,p_-^2)} \int\limits_{-t_{\mathrm{max}}}^{-t_{\mathrm{min}}} \mathrm{d}|t| \int\limits_0^{2\pi} \mathrm{d}\varphi, \tag{2.14}$$

with the boundaries t_{min} and t_{max} given by Equation 2.13 with $-1 \leq \cos\theta^* \leq 1$.

[1] 2 → 2 scattering processes can also involve four point vertices between gauge bosons. However, they can be decomposed into three point vertices of s- and t-channels connected by a propagator of a field with infinitely large mass. This is specially handy in algorithms, since it makes special four point vertex scenarios dispensable.

Chapter 3

Numerical Integration

For the numerical calculation of the integral over the phase space variables we make use of the Monte Carlo method. It approximates an integral I of a function f on an d-dimensional unit hypercube Ω as the sum Q_N so that

$$Q_N = \frac{1}{N}\sum_{i=1}^{N} f(\mathbf{r}_i) \quad \underset{N\to\infty}{\longrightarrow} \quad I = \int_\Omega \mathrm{d}\mathbf{r}\, f(\mathbf{r}), \tag{3.1}$$

with random variables $\mathbf{r}_i$, uniformly distributed on the domain Ω. The variance of Q_N is calculated from the variance of f

$$\mathrm{Var}(Q_N) = \frac{1}{N^2}\sum_{i=1}^{N} \mathrm{Var}(f) = \frac{\mathrm{Var}(f)}{N}, \tag{3.2}$$

where $\mathrm{Var}(f)$ is estimated by the unbiased sample variance as

$$\mathrm{Var}(f) \approx {s_N}^2 = \frac{1}{N-1}\sum_{i=1}^{N} \left(f(\mathbf{r}_i) - Q_N\right)^2, \tag{3.3}$$

with the sample mean Q_N. The convergence in Relation 3.1 holds true if the sequence $\{s_N\}_{N\in\mathbb{N}}$ is bound. In that case, the error on Q_N is estimated by

$$\Delta Q_N = \sqrt{\mathrm{Var}(Q_N)} = \frac{s_N}{\sqrt{N}}, \tag{3.4}$$

decreasing asymptotically for $N \to \infty$.

3.1 Parametrization on the unit hypercube

The cross section is a d-dimensional integral over phase space variables parametrizing the integrand that depends on the momentum configuration and the Bjorken variables. We will denote the set of phase space variables by $\boldsymbol{\phi}$, the momentum configuration[1] by $\mathbf{k}(\boldsymbol{\phi})$ and the integrand

[1] The complete set of four-vectors describing the kinetic configuration of external particles.

by $f(\mathbf{k}(\boldsymbol{\phi}))$. The phase space variables are τ and η, the parameterization of the Bjorken variables, together with the invariants s_i and $|t_i|$, and the angles $\cos\theta_i$ and φ_i, determined by a particular phase space factorization.

To perform the integration on the d-dimensional unit hypercube Ω, one changes variables $\phi_i : r_i \mapsto \phi_i \equiv \phi_i(r_i)$ for every phase space variable ϕ_i with $0 \leq r_i \leq 1$, so that $r_i = 0$ and $r_i = 1$ are mapped to the lower and upper bound of ϕ_i respectively.[2] The Jacobian of this transformation is

$$J_{\phi_i}(r_i) = \frac{\mathrm{d}\phi_i(r_i)}{\mathrm{d}r_i} \equiv \frac{1}{g_i(r_i)}. \tag{3.5}$$

Finally, the cross section integral is given by

$$I = \int \mathrm{d}\boldsymbol{\phi}\, f(\mathbf{k}(\boldsymbol{\phi})) = \int_\Omega \mathrm{d}\mathbf{r}\, \frac{f(\mathbf{k}(\boldsymbol{\phi}(\mathbf{r})))}{g(\mathbf{r})}, \tag{3.6}$$

where

$$\frac{1}{g(\mathbf{r})} \equiv \prod_{i=1}^{n} \frac{1}{g_i(r_i)} \tag{3.7}$$

is the absolute value of the Jacobian's determinant[3] $|\det J_{\boldsymbol{\phi}}(\mathbf{r})|$. The Monte Carlo method with N random points approximates the cross section as

$$Q_N = \frac{1}{N}\sum_{i=1}^{N} \frac{f(\mathbf{k}(\boldsymbol{\phi}(\mathbf{r}_i)))}{g(\mathbf{r}_i)}, \tag{3.8}$$

summing over N configurations $\mathbf{r}_i$, each composed of d uniformly distributed variables r_i with $0 \leq r_i \leq 1$.

The parametrization $\phi_i(r_i)$ can be chosen such that the Jacobian $1/g_i(r_i)$, flattens the integrand, which is e.g. compensating peak structures in f. A good choice of Jacobians decreases the variance and therefore increases the efficiency of the Monte Carlo simulation.

3.2 Multi-channel approach

In general, the cross section has contributions from multiple Feynman diagrams, each coming with its own peak structure and its own phase space decomposition. In this case, where the integrand $f(\mathbf{k}(\boldsymbol{\phi}))$ contains multiple peak structures, it cannot be flattened by a single transformation. The multi-channel approach [1] suggests to solve this problem.

Assume that for n peak structures, each structure i is captured by the transformation

$$\boldsymbol{\phi}_i : \mathbf{r} \mapsto \boldsymbol{\phi}_i(\mathbf{r}) \tag{3.9}$$

for the corresponding set of phase space variables $\boldsymbol{\phi}_i$ and $\mathbf{r} \in \Omega$, changing the phase space volume by the factor

$$|\det J_{\boldsymbol{\phi}_i}(\mathbf{r})| = \frac{1}{g_i(\mathbf{r})}. \tag{3.10}$$

[2] We denote both the transformation and the phase space variable by ϕ_i.

[3] Although imprecise, we will simply call it the Jacobian of the transformation.

The inverse transformation[4] is denoted by

$$\mathbf{r} : \boldsymbol{\phi}_i \mapsto \mathbf{r}(\boldsymbol{\phi}_i). \tag{3.11}$$

The integrand $f(\mathbf{k}(\boldsymbol{\phi}))$ can then be decomposed into a weighted sum of n channels with weights $\boldsymbol{\alpha}$, so that

$$\sum_{i=1}^{n} \alpha_i = 1 \tag{3.12}$$

and is well modeled by

$$g_{\boldsymbol{\alpha}}(\mathbf{k}) \equiv \sum_{i=1}^{n} \alpha_i \, g_i(\mathbf{r}(\boldsymbol{\phi}_i(\mathbf{k}))) \tag{3.13}$$

where $\boldsymbol{\phi}_i(\mathbf{k})$ denotes the phase space variables of channel i obtained from an arbitrary momentum configuration. This renders the function $f/g_{\boldsymbol{\alpha}}$ approximately flat. More precisely, the integral

$$I = \int \mathrm{d}\boldsymbol{\phi} \, f(\mathbf{k}(\boldsymbol{\phi})) \tag{3.14}$$

is rewritten as

$$I = \sum_{i=1}^{n} \alpha_i \int \mathrm{d}\boldsymbol{\phi} \, g_i(\mathbf{r}(\boldsymbol{\phi}_i(\mathbf{k}(\boldsymbol{\phi})))) \frac{f(\mathbf{k}(\boldsymbol{\phi}))}{g_{\boldsymbol{\alpha}}(\mathbf{k}(\boldsymbol{\phi}))} = \sum_{i=1}^{n} \alpha_i \int_{\Omega} \mathrm{d}\mathbf{r} \, \frac{f(\mathbf{k}(\boldsymbol{\phi}_i(\mathbf{r})))}{g_{\boldsymbol{\alpha}}(\mathbf{k}(\boldsymbol{\phi}_i(\mathbf{r})))}, \tag{3.15}$$

where in the second step, we picked the change of variables $\boldsymbol{\phi}_i$ for channel i, so that $g_i(\mathbf{r})$ drops out. The expected uncertainty of the numerical integration for N points is connected to the inferred variance ${s_N}^2$ and is given by

$$\frac{s_N}{\sqrt{N}} = \sqrt{\frac{W(\boldsymbol{\alpha}) - I^2}{N}}, \tag{3.16}$$

with

$$W(\boldsymbol{\alpha}) = \int \mathrm{d}\boldsymbol{\phi} \, g_{\boldsymbol{\alpha}}(\mathbf{k}(\boldsymbol{\phi})) \left(\frac{f(\mathbf{k}(\boldsymbol{\phi}))}{g_{\boldsymbol{\alpha}}(\mathbf{k}(\boldsymbol{\phi}))} \right)^2 = \sum_{i=1}^{n} \alpha_i \int_{\Omega} \mathrm{d}\mathbf{r} \, \left(\frac{f(\mathbf{k}(\boldsymbol{\phi}_i(\mathbf{r})))}{g_{\boldsymbol{\alpha}}(\mathbf{k}(\boldsymbol{\phi}_i(\mathbf{r})))} \right)^2. \tag{3.17}$$

Minimizing the expected error of integration corresponds to minimizing $W(\boldsymbol{\alpha})$ on the simplex $\sum_i \alpha_i = 1$. This leads to the condition at the extremum $\bar{\boldsymbol{\alpha}}$

$$\forall i : \quad W_i(\bar{\boldsymbol{\alpha}}) = \lambda, \quad \text{with} \quad W_i(\boldsymbol{\alpha}) \equiv -\frac{\partial}{\partial \alpha_i} W(\boldsymbol{\alpha}) \tag{3.18}$$

and the constant Lagrange multiplier λ. It holds that

$$W_i(\boldsymbol{\alpha}) = \int \mathrm{d}\boldsymbol{\phi} \, g_i(\mathbf{r}(\boldsymbol{\phi}_i(\mathbf{k}(\boldsymbol{\phi})))) \left(\frac{f(\mathbf{k}(\boldsymbol{\phi}))}{g_{\boldsymbol{\alpha}}(\mathbf{k}(\boldsymbol{\phi}))} \right)^2 = \int_{\Omega} \mathrm{d}\mathbf{r} \, \left(\frac{f(\mathbf{k}(\boldsymbol{\phi}_i(\mathbf{r})))}{g_{\boldsymbol{\alpha}}(\mathbf{k}(\boldsymbol{\phi}_i(\mathbf{r})))} \right)^2 \tag{3.19}$$

[4] We stick to our convention of denoting the transformation and the variable in target space by the same name. This however might lead to some confusion here. Instead of denoting the inverse transformation by $\boldsymbol{\phi}_i^{-1}$, we denote it by $\mathbf{r}$, identically to the variable on the hypercube $\mathbf{r}$. The intention is that the reader can tell right away that $\mathbf{r}(\boldsymbol{\phi}_i)$ is a variable on the hypercube, which is not immediate from the obscure alternative $\boldsymbol{\phi}_i^{-1}(\boldsymbol{\phi}_i)$.

is the contribution of channel i to $W = \sum_i \alpha_i W_i$. At the extremum, this also implies $W_i(\bar{\boldsymbol{\alpha}}) = W(\bar{\boldsymbol{\alpha}})$, meaning that each channel contributes the same amount to the total variance.

That the extremum is indeed a minimum can be seen by expanding $W(\boldsymbol{\alpha})$ around $\alpha_i = \bar{\alpha}_i + \beta_i$, with a small shift β_i from the extremum. Since $\sum_i \beta_i = 0$ the first order term vanishes and the expansion reads

$$W(\boldsymbol{\alpha}) = W(\bar{\boldsymbol{\alpha}}) + \int \mathrm{d}\boldsymbol{\phi} \frac{f^2(\mathbf{k}(\boldsymbol{\phi}))}{{g_{\boldsymbol{\alpha}}}^3(\mathbf{k}(\boldsymbol{\phi}))} \left(\sum_{i=1}^{n} \beta_i g_i(\mathbf{r}(\boldsymbol{\phi}_i(\mathbf{k}(\boldsymbol{\phi})))) \right)^2 + \mathcal{O}({\beta_i}^3), \tag{3.20}$$

where the second order term is a positive quantity since $g_{\boldsymbol{\alpha}} > 0$.

We talk of stratified sampling when the channel transformations map to pairwise disjoint regions of phase space. Under these circumstances, it holds that

$$g_i(\mathbf{r}(\boldsymbol{\phi}_i(\mathbf{k}(\boldsymbol{\phi})))) = \frac{1}{\omega_i} \chi_i(\boldsymbol{\phi}), \tag{3.21}$$

with the characteristic function[5] χ_i and the volume $\omega_i \equiv \int \mathrm{d}\boldsymbol{\phi}\, \chi_i(\boldsymbol{\phi})$ of region i. In this case, there is a direct relation between W_i and α_i,

$$W_i(\boldsymbol{\alpha}) = \frac{\omega_i}{{\alpha_i}^2} \int \mathrm{d}\boldsymbol{\phi}\, \chi_i(\boldsymbol{\phi})\, f^2(\mathbf{k}(\boldsymbol{\phi})) \equiv \frac{C_i}{{\alpha_i}^2}, \tag{3.22}$$

with C_i, a constant independent of $\boldsymbol{\alpha}$. The minimum $\bar{\boldsymbol{\alpha}}$ is then found after one iteration, assuming W_i is exactly known for each channel i, and is given by

$$\bar{\alpha}_i = \alpha_i \sqrt{\frac{W_i(\boldsymbol{\alpha})}{W_i(\bar{\boldsymbol{\alpha}})}} = \alpha_i \sqrt{\frac{W_i(\boldsymbol{\alpha})}{\lambda}}, \tag{3.23}$$

with the Lagrange multiplier

$$\lambda = \left(\sum_i C_i \right)^2, \tag{3.24}$$

guaranteeing normalization.

In general however, where we do not have stratified sampling, an adaptation prescription for the $\boldsymbol{\alpha}$ is not straightforward. Nevertheless each channel maps random points favorably to one particular peaking structure, so that those regions are approximately disjoint. Therefore, the following adaptation procedure is proposed

$$\alpha_i^{\text{new}} \propto \alpha_i \sqrt{W_i(\boldsymbol{\alpha})}, \tag{3.25}$$

while additionally imposing normalization $\sum_i \alpha_i^{\text{new}} = 1$. After sufficiently many iterations $\boldsymbol{\alpha}^{\mathbf{new}}$ converges to the minimum $\bar{\boldsymbol{\alpha}}$.

[5] The characteristic function of region i is the map χ_i that maps to 1 if its argument is in region i and to 0 otherwise.

3.3 The vegas algorithm

Our knowledge about peak structures in Feynman amplitudes can be used to significantly decrease the variance of the integrand and improve the efficiency of the integration. This is achieved with the methods presented previously in Sections 3.1 and 3.2. The integrand is then left with only the structures that were either unexpected or not predictable enough to be accounted for by explicit variable transformations. An efficient integrator would therefore apply additional adaptation that does not require any assumptions about the integrand but further decreases its variance. This is what `vegas` is designed for [8].

The `vegas` algorithm makes use of adaptive stratified sampling to reduce the Monte Carlo error by remapping integration variables in each direction. This is importance sampling on a discrete grid, assigning a constant weight, the Jacobian of this transformation, to each cell. `vegas` probes the integrand and adapts the grid boundaries accordingly. The weights are proportional to the size of the cell. The ideal adaptation strategy requires just one iteration when assuming exact estimators. This was outlined in the previous section. `vegas` controls the adaptation with a parameter `alpha`, larger than zero and typically chosen to be of order one. The grid adapts quicker for larger values of `alpha` but can also overfit to random fluctuations when probing the integrand. Initially, `vegas` does not know about the structure of the integrand and assumes a uniform grid. After several iterations, the grid is expected to converge to the optimal grid.

Note that `vegas` could in principle account for Breit-Wigner or massless propagator enhancements of a single channel since the phase space decomposition chooses integration variables in such a way that the grid is aligned with these peaks. However, it is obviously better to flatten known enhancements directly instead of fitting them adaptively with `vegas`. Nevertheless, it is of advantage to include the functionality of `vegas` in the flattening process. Even though, or precisely because the propagator peaks of the integrand are flattened in advance, `vegas` can account for nuances and other unforeseen structures.

`Vegas3`, the `Python` implementation of `vegas`, comes with a number of customizable objects. This makes it convenient to develop a specialized integrator for Feynman amplitudes that combines multi-channeling, importance sampling and adapts iteratively to the integrand.

Chapter 4

Importance Sampling for Feynman Amplitudes

Previously, in Chapter 3, we saw that the efficiency of a Monte Carlo simulation can be increased by a suitable parametrization of integration variables. A good change of variables is one, whose Jacobian flattens the peak structures and therefore decreases the variance of the integrand. In Feynman amplitudes, the dominating structures are the propagator resonances at their pole mass. They are accounted for by the mappings that we describe in the next section. This procedure is what we mean by importance sampling.

Furthermore, Feynman amplitudes depend on initial and final momentum configurations that are parameterized by the phase space variables. In Section 4.2, we go through the stages of the generation of a momentum configuration from a set of random numbers in a generic fashion. The procedure has to follow a certain logic so that the integration variables from phase space factorization can be employed. Roughly outlined, at first, we fix the center of mass energy for the collision. In the case of hadronic collisions, this demands the generation of the Bjorken variables. Next, we sample the time-like invariants, which can be understood as the center of mass energies of the factorizing building blocks. The outgoing momenta of these blocks additionally depend on angles that are parametrized and sampled differently in s- and t-channels. At last, we apply Lorentz transformations to obtain the full momentum configuration in the desired frame of reference.

Ambiguities and possible challenges within the process of phase space generation are discussed in Section 4.3 and eventually, we present a detailed example in six-body phase space.

4.1 Propagator mappings

Feynman diagrams come with predictable peak structures. In general, they manifest themselves as Breit-Wigner resonances or distributions of the form $\phi^{-\nu}$, arising from massive and massless propagators respectively. If the peak structure of f is aligned with a variable ϕ, the mapping

$\phi(r)$ is chosen such that the peak gets compensated for by its Jacobian

$$\frac{\mathrm{d}\phi(r)}{\mathrm{d}r} \equiv \frac{1}{g(r)}. \tag{4.1}$$

Since we require that $g(r)$ takes precisely the functional form of the peak, one finds the right mapping simply by integrating $1/g(r)$ and applying the boundary conditions. The mappings read:

Propagator with vanishing width: $f \propto 1/(\phi - m^2)^\nu$

$$\phi(r) = \left[(\phi_{\text{max}} - m^2)^{1-\nu} r + (\phi_{\text{min}} - m^2)^{1-\nu}(1-r)\right]^{\frac{1}{1-\nu}} + m^2, \tag{4.2}$$

$$\frac{1}{g(r)} = \frac{\left[(\phi_{\text{max}} - m^2)^{1-\nu} - (\phi_{\text{min}} - m^2)^{1-\nu}\right]}{1-\nu}(\phi - m^2)^\nu \tag{4.3}$$

for $\nu \neq 1$ and

$$\phi(r) = \exp\left[\ln(\phi_{\text{max}} - m^2) r + \ln(\phi_{\text{min}} - m^2)(1-r)\right] + m^2, \tag{4.4}$$

$$\frac{1}{g(r)} = \left[\ln(\phi_{\text{max}} - m^2) - \ln(\phi_{\text{min}} - m^2)\right](\phi - m^2) \tag{4.5}$$

for $\nu = 1$.

Breit-Wigner propagator: $f \propto 1/\left[(\phi - M^2)^2 + M^2\Gamma^2\right]$

$$\phi(r) = M\Gamma \tan\left[y_1 + (y_2 - y_1) r\right] + M^2, \tag{4.6}$$

$$\frac{1}{g(r)} = \frac{y_2 - y_1}{M\Gamma}\left[(\phi - M^2)^2 + M^2\Gamma^2\right] \tag{4.7}$$

with

$$y_{1/2} = \arctan\left(\frac{\phi_{\text{min/max}} - M^2}{M\Gamma}\right). \tag{4.8}$$

These mappings are not only applicable for s-channel propagators for the Mandelstam invariant s, but also for the massless t-channel propagators for the absolute value of the Mandelstam invariants $|t|$ (and $|u|$). Massive t-channel propagators and Mandelstam invariants s, which do not correspond to a propagator are not expected to exhibit a peaking behavior. Together with the remaining angles $\cos\theta$ and φ, they are sampled uniformly

$$z(r) = z_{\text{max}} r + z_{\text{min}}(1-r), \qquad \cos\theta(r) = 2r - 1, \qquad \varphi(r) = 2\pi r. \tag{4.9}$$

The parameter ν should be chosen $\gtrsim 1$ and can be varied to improve the Monte Carlo integration. The naive expectation $\nu = 2$ is not necessarily the best choice because the corresponding poles of the integrand are partly cancelled in the collinear limit [5, 7].

The parameter m^2 appearing in the mapping with vanishing width, is usually set to zero, since massive particles have a non vanishing decay width. To avoid numerical problems if $\phi_{\text{min}} = 0$, it is chosen to be $m^2 = -a$, with a small parameter $a > 0$.

4.2 Phase space generation

All possible momentum configurations of a single Feynman diagram, called a topology[1], can be systematically generated. In Chapter 2, we saw that the phase space variables ϕ parametrize the momentum configuration. In general they are the variables τ and η, related to the Bjorken variables, the Mandelstam invariants s_i and $|t_i|$, and the angles $\cos\theta_i$ and φ_i. In principle, the momentum configuration can be generated in any reference frame since the phase space is Lorentz invariant. For the sake of numerical stability and methodological clarity, the reference frame will be chosen as the center of mass frame of the two incoming point-like or composite particles with their momenta along the z-axis. We will call it the COM frame.

In the following, we describe a procedure that generates a phase space momentum configuration in the COM frame for an arbitrary topology.

4.2.1 Bjorken variables

First of all, the variables τ and η need to be determined. In point-like particle collision, τ and η are fixed to 1 and 0 respectively. In hadronic collisions, τ is sampled between

$$\tau_{\text{min}} = 0 \qquad \text{and} \qquad \tau_{\text{max}} = 1, \tag{4.10}$$

then η is sampled between

$$\eta_{\text{min}} = \frac{1}{2}\ln\tau \qquad \text{and} \qquad \eta_{\text{max}} = -\frac{1}{2}\ln\tau. \tag{4.11}$$

In the case where the topology consists of t-channel propagators, τ is sampled uniformly. In the other case, where the topology consists of s-channel propagators only, τ is importance sampled according to either the massless or the Breit-Wigner distribution, depending on the involved propagator. There, one needs to pay attention to the choice of parameters since τ appears in the propagator as the combination $\hat{s} = \tau s$, with the squared collider energy s. The center of mass energy at partonic level is then determined by $E_{\text{cm}} = \sqrt{\tau}\sqrt{s}$. Independent of the topology, η is always sampled uniformly.

Once τ and η are fixed, the Bjorken variables x_1 and x_2 can easily be calculated according to Equation 2.2.

4.2.2 Time-like invariants

Secondly, the time-like Mandelstam invariants are generated. The bounds of the invariant s_i are given by

$$s_{i,\text{min}} = (m_1 + m_2)^2 \qquad \text{and} \qquad s_{i,\text{max}} = \left(E_{\text{cm}} - \textstyle\sum_k m_k\right)^2, \tag{4.12}$$

where $m_{1/2}$ are the invariant masses of the two outgoing legs. The sum runs over the innermost propagators of the topology whose invariant masses were already generated.

[1] Several Feynman diagrams can share the same topology.

The invariants can be divided into two categories: distributed and uniform invariants. The distributed invariants can all be identified with a propagator peak, whereas the uniform invariants cannot. In general, there are always distributed invariants, unless the topology consists of t-channels only. The ordering in which to generate the invariants is in general ambiguous as we will see in Section 4.3.

Starting with the distributed invariants, one needs to begin at an outermost vertex, where the masses of both outgoing legs are given by the on shell conditions. They are importance sampled according to either massive or massless distributions. One proceeds with the uniform invariants only after all distributed invariants are generated.

If the topology contains at least two t-channels, there are uniform invariants to be sampled. Each uniform invariant stands for a t-channel $2 \to 2$ building block, characterized by a vertex pair. In Section 4.3 we describe which pairs are allowed. They are neither arbitrary nor unique. Once a pair is selected, its corresponding invariant is sampled uniformly.

4.2.3 Momenta from phase space variables

Thirdly, the momenta are generated. Starting with the initial momenta, given by $p_\pm$ in Equation 2.12, intermediate momenta are generated successively, going outwards towards the final momenta. The ordering is defined by the same path that defined the generation of the uniform invariants. The momenta involved in t-channel $2 \to 2$ blocks, if any, must be generated first before proceeding with the s-channels.

The t-channels

A valid path chooses a t-channel block where the initial momenta p_+ and p_- are known. The squared invariant masses of the outgoing particles s_1 and s_2 were generated previously as distributed or uniform invariants and determine the boundaries of the invariant t according to Equation 2.13. They are obtained from $0 \leq \cos\theta^* \leq 1$ as

$$t_{\text{min}} = s_1 + p_+^2 - \frac{(s+s_1-s_2)(s+p_+^2-p_-^2) + \lambda^{\frac{1}{2}}(s,s_1,s_2)\lambda^{\frac{1}{2}}(s,p_+^2,p_-^2)}{2s} \tag{4.13}$$

and

$$t_{\text{max}} = s_1 + p_+^2 - \frac{(s+s_1-s_2)(s+p_+^2-p_-^2) - \lambda^{\frac{1}{2}}(s,s_1,s_2)\lambda^{\frac{1}{2}}(s,p_+^2,p_-^2)}{2s}, \tag{4.14}$$

with $s = (p_+ + p_-)^2$.

The space-like invariant t is then importance sampled in the massless case but uniformly generated in the massive case, since the latter does not exhibit a peaking behavior. Note that $t \leq 0$, which makes it more convenient to sample the absolute value $|t|$, with the inverted boundaries. Once, $|t|$ is generated, one can directly obtain $\cos\theta^*$ by inverting Equation 2.13.

As a next step, the azimuthal angle φ^* is sampled uniformly between the boundaries

$$\varphi^*_{\text{min}} = 0 \qquad \text{and} \qquad \varphi^*_{\text{max}} = 2\pi. \tag{4.15}$$

Given φ^* and $\cos\theta^*$, the outgoing momenta k_1^* and k_2^* are generated in the rest frame according to Equation 2.10, the frame where $p = p_+ + p_-$ is at rest and p_+ and p_- are along the positive and negative z-direction, respectively.

In a general t-channel block, p is not at rest and $p_\pm$ not along the z-axis. The Lorentz transformation connecting the rest to the COM frame is therefore composed of a boost with velocity $\vec{v} = p/p^0$ and a rotation. In the COM frame, the outgoing momenta are given by

$$k_1 = \mathcal{B}\left(\vec{v}\right)\mathcal{R}\left(\varphi, \cos\theta\right) k_1^* \qquad \text{and} \qquad k_2 = \mathcal{B}\left(\vec{v}\right)\mathcal{R}\left(\varphi, \cos\theta\right) k_2^*, \tag{4.16}$$

with the rotation angles

$$\cos\theta = \frac{\hat{p}_+^3}{|\vec{\hat{p}}_+|} \qquad \text{and} \qquad \varphi = \begin{cases} \arctan\frac{\hat{p}_+^2}{\hat{p}_+^1}, & \hat{p}_+^1 > 0 \\ \arctan\frac{\hat{p}_+^2}{\hat{p}_+^1} + \pi, & \hat{p}_+^1 < 0, \end{cases} \tag{4.17}$$

where the angles are obtained from $\hat{p}_+ = \mathcal{B}\left(-\vec{v}\right) p_+$. They are deduced from the requirement that $p_+^* = \mathcal{R}^{-1}\left(\varphi, \cos\theta\right)\hat{p}_+$ is along the positive z-direction. The same rotation angles rotate p_- along the negative z-direction, since p is at rest.

The final momenta cannot always be identified with a single leg. The momentum transfer $p_+ - k_1$ however, can always be assigned to a propagator line.

The s-channels

After all t-channel blocks were considered, one carries on with the s-channels. They correspond to $1 \to 2$ decays. Note that in topologies without t-channels, the colliding particles fuse to one with a fixed momentum p. Then, the procedure is generic. The particle with momentum p decays into two particles with squared masses s_1 and s_2. The decay angles are sampled uniformly between the boundaries

$$\varphi_{\text{min}}^* = 0, \qquad \varphi_{\text{max}}^* = 2\pi, \qquad \cos\theta_{\text{min}}^* = -1 \qquad \text{and} \qquad \cos\theta_{\text{max}}^* = 1. \tag{4.18}$$

Given φ^* and $\cos\theta^*$, the outgoing momenta k_1^* and k_2^* are generated according to Equation 2.10 in the rest frame of p. In the COM frame, they read

$$k_1 = \mathcal{B}\left(\vec{v}\right) k_1^* \qquad \text{and} \qquad k_2 = \mathcal{B}\left(\vec{v}\right) k_2^*, \tag{4.19}$$

with a boost of velocity $\vec{v} = \vec{p}/p^0$. Other than in t-channels, no additional rotation has to be applied here, since $\cos\theta^*$ is a uniformly generated random variable.

4.3 Ambiguous topologies

The phase space parametrization is not uniquely defined for topologies with multiple t-channels. There, one needs to select uniform invariants. They are sampled uniformly since they are not expected to exhibit a peak structure. It is not clear why one choice of invariant should be preferred to another. Note however that randomly choosing a decomposition is not ideal. This would then assign the same random variable to different uniform invariants. Although we do not expect peaking behavior in this variable, the integrand might still not be completely flat along this direction. These features would then be accounted for by `vegas`, which further reduces the variance. It is therefore of advantage to stick with one choice of phase space parametrization.

Once a set of phase space variables is selected that parametrize the phase space, there is the additional ambiguity of the ordering in which the variables should be sampled. This ordering is what we mean by path. In principle, this ambiguity of the path is not a problem, since every possible path always captures the full phase space and can be used for importance sampling. As we se in the next section, the advantage of one path to another comes into play only if we have to deal with a more complicated peak structure.

4.3.1 Breit-Wigner competition

Consider a Higgs boson decaying into two Z bosons. Due to the values of their masses, it is impossible for them all to be simultaneously on shell. A topology with this subprocess has a competing Breit-Wigner structure. Any phase space parametrization will capture these distinct Breit-Wigner peaks eventually by sampling enough points. However, in the scenario where the invariant masses of the two Z bosons are generated first, they are predominantly sampled close to the Z boson mass. This drives the lower limit for the invariant mass of the Higgs propagator up, so that the majority of generated events are above the Higgs mass. This parametrization therefore requires many more points to sample the Breit-Wigner of the Higgs sufficiently accurate.

In general, there is no single ordering to sample the phase space variables that optimally accounts for all peaks. Ideally, a phase space generator of a competing Breit-Wigner topology samples events according to multiple paths. There should be as many paths as competing peak structures. A phase space generator that combines all of these paths, samples sufficiently many events at every peak.

It is emphasized that the choice of a path only matters in the competing Breit-Wigner scenario, where it affects the frequency at which competing peak structures are sampled. In other words, it is unlikely but still possible for one path to generate Monte Carlo points at a competing structure. However, note that independent of the path, the Jacobian of the importance sampling transformation correctly flattens all propagator enhancements of the topology, including competing Breit-Wigner structures, simultaneously.

4.3.2 Possible paths

Most importantly, a path generator needs to select an ordering that does not conflict with the integration boundaries. A consistent choice is to select a phase space variable as the next variable in the list if the invariant masses of the outgoing legs of the building block it corresponds to are known. This list however, is still ambiguous.

In general, one starts by determining the ordering of the distributed invariants. They correspond to center of mass energies of $1 \to 2$ decay building blocks, each uniquely identified by a vertex. Note that while s-channels are decomposed into unique phase space variables, their ordering is not necessarily fixed. One possible strategy is to randomly pick one of the vertices, where the (virtual) masses of the the outgoing legs are known, and select it as the next one in the list. This strategy however, can never account for competing Breit-Wigner structures. This requires to skip vertices in a list like the above and reverse engineer the boundaries of the invariants corresponding to the skipped vertices.

Next, the uniform invariants need to be determined. They come with $2 \to 2$ t-channel scattering building blocks, each uniquely identified by a vertex pair. For a given process, every set of uniform invariants corresponds to a different phase space decomposition. For a given decomposition, we can again have an ambiguous ordering of the variables. However, the ordering is irrelevant here since these invariants are sampled uniformly.

4.4 Example process

To demonstrate the phase space generation we consider the topology in Figure 4.1. We determine the set of phase space variables ϕ and according to a suitable path, we generate the momentum configuration k_i and the total phase space weight $1/g$, combining both the Jacobians and the phase space volume factors.

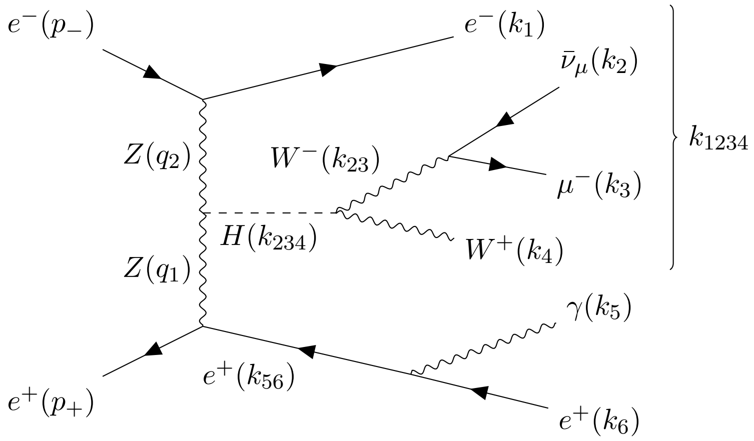


Figure 4.1: An example of a topology that contains two t-channels and three s-channels.

The topology in Figure 4.1 describes the fusion of two Z bosons into a Higgs that is decaying into W bosons. The corresponding phase space factorization decomposes the this six-body phase space into three s-channels and two t-channels. It is parametrized by 14 phase space variables ϕ.

For ϕ we choose three time-like invariants

$$\begin{aligned} s_{23} = {k_{23}}^2 &= (k_2 + k_3)^2, \\ s_{234} = {k_{234}}^2 &= (k_{23} + k_4)^2, \\ s_{56} = {k_{56}}^2 &= (k_5 + k_6)^2, \\ s_{1234} = {k_{1234}}^2 &= (k_1 + k_{234})^2, \end{aligned} \tag{4.20}$$

two space-like invariants

$$\begin{aligned} t_1 &= {q_1}^2 = (p_+ - k_{56})^2, \\ t_2 &= {q_2}^2 = (p_- - k_1)^2 \end{aligned} \tag{4.21}$$

and five azimuthal and three polar angles $\varphi_{1,2,234,23,56}$ and $\theta_{234,23,56}$. For massless external fermions, the phase space integral is then decomposed and written as

$$\begin{aligned} \int \mathrm{d}\Phi_6 = &\int\limits_{s_{23,\min}}^{s_{23,\max}} \mathrm{d}s_{23} \int\limits_{s_{234,\min}}^{s_{234,\max}} \mathrm{d}s_{234} \int\limits_{s_{56,\min}}^{s_{56,\max}} \mathrm{d}s_{56} \int\limits_{s_{1234,\min}}^{s_{1234,\max}} \mathrm{d}s_{1234} \\ &\times \left[\frac{1}{4s} \int\limits_{-t_{1,\max}}^{-t_{1,\min}} \mathrm{d}|t_1| \int\limits_{0}^{2\pi} \mathrm{d}\varphi_1 \right] \times \left[\frac{1}{4(s-t_1)} \int\limits_{-t_{2,\max}}^{-t_{2,\min}} \mathrm{d}|t_2| \int\limits_{0}^{2\pi} \mathrm{d}\varphi_2 \right] \\ &\times \left[\frac{\lambda^{\frac{1}{2}}\left(s_{234}, s_{23}, {m_W}^2\right)}{8 s_{234}} \int\limits_{-1}^{1} \mathrm{d}\cos\theta_{234} \int\limits_{0}^{2\pi} \mathrm{d}\varphi_{234} \right] \\ &\times \left[\frac{1}{8} \int\limits_{-1}^{1} \mathrm{d}\cos\theta_{23} \int\limits_{0}^{2\pi} \mathrm{d}\varphi_{23} \right] \times \left[\frac{1}{8} \int\limits_{-1}^{1} \mathrm{d}\cos\theta_{56} \int\limits_{0}^{2\pi} \mathrm{d}\varphi_{56} \right] \end{aligned} \tag{4.22}$$

with the squared center of mass energy $s = (p_+ + p_-)^2$, the W boson mass m_W and the Källén function λ. For the time-like invariants, the boundaries read

$$\begin{aligned} s_{23,\min} &= 0, & s_{23,\max} &= (\sqrt{s} - m_W)^2, \\ s_{234,\min} &= (\sqrt{s_{23}} + m_W)^2, & s_{234,\max} &= s, \\ s_{56,\min} &= s_{56,\mathrm{cut}}, & s_{56,\max} &= (\sqrt{s} - \sqrt{s_{234}})^2, \\ s_{1234,\min} &= s_{234}, & s_{1234,\max} &= (\sqrt{s} - \sqrt{s_{56}})^2 \end{aligned} \tag{4.23}$$

and

$$\begin{aligned} t_{1,\min} &= \frac{s_{1234} + s_{56} - s - \lambda^{\frac{1}{2}}(s, s_{1234}, s_{56})}{2}, \\ t_{1,\max} &= \frac{s_{1234} + s_{56} - s + \lambda^{\frac{1}{2}}(s, s_{1234}, s_{56})}{2}, \\ t_{2,\min} &= \frac{(t_1 - s_{1234})(s_{1234} - s_{234})}{s_{1234}}, \\ t_{2,\max} &= 0, \end{aligned} \tag{4.24}$$

for the space-like invariants. Note that the cross section diverges when the momenta k_5 and k_6 are collinear. This divergence is unphysical since the electron's mass prevents such a scenario. In the numerical simulation, where the electron is approximated to be massless, this is simply regulated by a cutoff $s_{56,\text{cut}}$ imposed during the Monte Carlo integration.

Next, each phase space variable is sampled from a set of random variables $\mathbf{r}$. The decomposition with the limits from above requires the invariants to be generated in the order s_{23}, s_{234}, s_{56}, s_{1234}, $|t_1|$, $|t_2|$, starting with the first variable in this sequence. The angles can be generated at any time since their limits are constants.

The angle variables $\varphi_{1,2,234,23,56}$ and $\cos\theta_{234,23,56}$, the invariant s_{1234} and the absolute values of the space-like invariants $|t_1|$ and $|t_1|$ are all sampled uniformly between their boundaries. The invariant s_{23} is sampled according to the Breit-Wigner of the W boson, s_{234} according to one of the Higgs and s_{56} according to a massless distribution. The transformation of each variable comes with the weight of its Jacobian. For $\nu \neq 1$, the total weight is given by

$$\begin{aligned}\frac{1}{g} = &\frac{\left(s_{23} - {m_W}^2\right)^2 + m_W^2 {\Gamma_W}^2}{m_W \Gamma_W}\left(y_2^W - y_1^W\right) \times \frac{s_{56,\text{max}} - s_{56,\text{min}}}{1-\nu} {s_{56}}^{\nu} \\ &\times \frac{\left(s_{345} - {m_H}^2\right)^2 + m_H^2 {\Gamma_H}^2}{m_H \Gamma_H}\left(y_2^H - y_1^H\right) \times \left(s_{1234,\text{max}} - s_{1234,\text{min}}\right) \\ &\times \left(\frac{\pi}{2}\right)^5 \frac{\lambda^{\frac{1}{2}}\left(s_{234}, s_{23}, {m_W}^2\right)}{s(s-t_1)s_{234}} \times \lambda^{\frac{1}{2}}(s, s_{1234}, s_{56}) \times |t_{2,\text{min}}|,\end{aligned} \tag{4.25}$$

with

$$y_{1,2}^X = \arctan\left(\frac{s_{\text{min,max}}^X - {m_X}^2}{m_X \Gamma_X}\right) \tag{4.26}$$

for $X = W, H$, $s^W \equiv s_{23}$ and $s^H \equiv s_{234}$.

The momenta are parametrized with the phase space variables as follows. We start with the beam momenta

$$p_+ = \frac{\sqrt{s}}{2}\begin{pmatrix}1\\0\\0\\1\end{pmatrix} \quad \text{and} \quad p_- = \frac{\sqrt{s}}{2}\begin{pmatrix}1\\0\\0\\-1\end{pmatrix}. \tag{4.27}$$

The next momenta are generated in the rest frame of $p = p_+ + p_-$, the lab frame. They are written with the rotation matrix $\mathcal{R}$ from Equation 2.11 as

$$k_{1234} = \mathcal{R}\left(\varphi_1, \cos\theta_1\right)\begin{pmatrix}\frac{s+s_{1234}-s_{56}}{2\sqrt{s}}\\0\\0\\\frac{\lambda^{\frac{1}{2}}(s,s_{1234},s_{56})}{2\sqrt{s}}\end{pmatrix} \quad \text{and} \quad k_{56} = \begin{pmatrix}k_{1234}^0\\-\vec{k}_{1234}\end{pmatrix}, \tag{4.28}$$

with

$$\cos\theta_1 = \frac{2t_1 - s_{1234} - s_{56} + s}{\lambda^{\frac{1}{2}}(s, s_{1234}, s_{56})}. \tag{4.29}$$

In the frame where k_{1234} is at rest and q_1 along the positive z-direction, we get the momenta

$$k_1' = \frac{s_{1234} - s_{234}}{2\sqrt{s_{1234}}} \begin{pmatrix} 1 \\ \sin\theta_2 \cos\varphi_2 \\ \sin\theta_2 \sin\varphi_2 \\ \cos\theta_2 \end{pmatrix} \quad \text{and} \quad k_{234}' = \begin{pmatrix} k_1'^0 \\ -\vec{k}_1' \end{pmatrix}, \tag{4.30}$$

with

$$\cos\theta_2 = \frac{2t_2 s_{1234}}{(s_{1234} - s_{234})(s_{1234} - t_1)} + 1. \tag{4.31}$$

In the frame where k_{234}' is at rest, we obtain

$$k_{23}'' = \mathcal{R}\left(\varphi_{234}, \cos\theta_{234}\right) \begin{pmatrix} \frac{s_{234}+s_{23}-m_W^2}{2\sqrt{s_{234}}} \\ 0 \\ 0 \\ \frac{\lambda^{\frac{1}{2}}\left(s_{234}, s_{23}, m_W^2\right)}{2\sqrt{s_{234}}} \end{pmatrix} \quad \text{and} \quad k_4'' = \begin{pmatrix} k_{23}''^0 \\ -\vec{k}_{23}'' \end{pmatrix}. \tag{4.32}$$

Analogously, we write in the frame where k_{23}'' is at rest

$$k_2''' = \frac{\sqrt{s_{234}}}{2} \begin{pmatrix} 1 \\ \sin\theta_{23} \cos\varphi_{23} \\ \sin\theta_{23} \sin\varphi_{23} \\ \cos\theta_{23} \end{pmatrix} \quad \text{and} \quad k_3''' = \begin{pmatrix} k_2'''^0 \\ -\vec{k}_2''' \end{pmatrix}. \tag{4.33}$$

Finally, in the frame where k_{56} is at rest, the outgoing momenta are given by

$$k_5'''' = \frac{\sqrt{s_{56}}}{2} \begin{pmatrix} 1 \\ \sin\theta_{56} \cos\varphi_{56} \\ \sin\theta_{56} \sin\varphi_{56} \\ \cos\theta_{56} \end{pmatrix} \quad \text{and} \quad k_6'''' = \begin{pmatrix} k_5''''^0 \\ -\vec{k}_5'''' \end{pmatrix}. \tag{4.34}$$

The momenta in the lab frame, the unprimed system, are then obtained with Lorentz transformations and read

$$\begin{aligned} k_1 &= \mathcal{B}\left(\vec{v}'\right) k_1' \\ k_{234} &= \mathcal{B}\left(\vec{v}'\right) k_{234}' \end{aligned} \quad \text{with} \quad \vec{v}' = \frac{\vec{k}_{1234}}{k_{1234}^0},$$
$$\begin{aligned} k_{23} &= \mathcal{B}\left(\vec{v}''\right) k_{23}'' \\ k_4 &= \mathcal{B}\left(\vec{v}''\right) k_4'' \end{aligned} \quad \text{with} \quad \vec{v}'' = \frac{\vec{k}_{234}}{k_{234}^0},$$
$$\begin{aligned} k_2 &= \mathcal{B}\left(\vec{v}'''\right) k_2''' \\ k_3 &= \mathcal{B}\left(\vec{v}'''\right) k_3''' \end{aligned} \quad \text{with} \quad \vec{v}''' = \frac{\vec{k}_{45}}{k_{23}^0}, \tag{4.35}$$
$$\begin{aligned} k_5 &= \mathcal{B}\left(\vec{v}''''\right) k_5'''' \\ k_6 &= \mathcal{B}\left(\vec{v}''''\right) k_6'''' \end{aligned} \quad \text{with} \quad \vec{v}'''' = \frac{\vec{k}_{56}}{k_{56}^0},$$

where $\mathcal{B}\left(\vec{v}\right)$ denotes a Lorentz boost with velocity $\vec{v}$. Note that in general, the rotation of the outgoing momenta only matters for t-channel building blocks. Nevertheless, the rotation could be omitted here, since $|t_1|$ and $|t_2|$ were sampled uniformly.

Chapter 5

Tests and Results

As outlined in the previous sections, the numerical computation of cross sections has to overcome several difficulties. The matrix elements of tree level diagrams come with peak structures due to the involved massive or massless propagators. It is the task of a good integration strategy to flatten these enhancements, so that the estimate of the cross section is not driven by fluctuations. The multi-channel approach, with the right phase space parametrization and sampling of the integration variables promises to be such a strategy.

In the framework of MG5aMC we implemented the tools needed for the computation of cross sections according to the methods outlined above. This includes the multi-channel phase space generator MCPS that identifies the channels and the combined Jacobian weight g_α, which in turn depends on the phase space parameterizations of each topology involved. The phase space parametrization itself is determined by the single-channel phase space generator SCPS. It chooses a set of phase space variables according to the topology the channel corresponds to. These variables are importance sampled such that the channel's enhancements are canceled. Where SCPS only knows about the phase space variables and the sampling of one channel, MCPS introduces the functionality to combine these channels. The actual integration of the channels is handled by the multi-channel integrator MCVegas, developed in the context of this thesis. It relies on MCPS.

In the following sections we show some results that were obtained by multiple tests of our implemented tools. The validation involves comparisons against results produced by MadEvent, the default leading order integration module within MG5aMC. MadEvent is an interesting benchmark, since it makes use of the same importance sampling strategies but uses a different method for multi-channeling.

The next section discusses the integration of single-diagram matrix elements, focusing on the benefits of importance sampling by SCPS. The resulting integrals are unphysical. Physical cross sections generally have contributions of multiple topologies. They are efficiently computed with the multi-channel approach, which is discussed in Section 5.2.

5.1 Importance sampling

In this section, we investigate the effect of importance sampling on multiple single-diagram matrix elements. For this purpose, these matrix elements are numerically integrated. The efficiency of the integration depends on the choice of a phase space generator, which parametrizes a momentum configuration with a set of random numbers, and the strategy of the integrator. For the results presented here, two types of phase space generators and two types of integrators were used.

The flat phase space generator `FlatPS` generates events uniformly across phase space with a constant Jacobian weight. It does not account for the peaks in the matrix element. However, the single-channel phase space generator `SCPS` makes use of importance sampling and applies it in such a way that flattens the integrand.

Besides the conventional Monte Carlo integrator `Naive` that provides the phase space generator with a set of uniformly generated random variables, there is `Vegas3`, a more elaborate integrator that applies adaptive stratified sampling. `Vegas3` iteratively changes an underlying grid and provides the phase space generator with redistributed variables.

The efficiency of the phase space generator is determined by the variance of the integrand, where a smaller variance requires fewer evaluations to arrive at a given accuracy. This underlying principle is illustrated with plots of the integrand and predictions of the cross section that explain the drastic improvements of importance sampling.

In Section 5.1.1, we discuss the Monte Carlo integration on the two-body phase space. It is already captured well with a small number of Monte Carlo points, which therefore allows us to quickly make reliable estimates. The integration on the higher dimensional three-body phase space is investigated in Section 5.1.2.

5.1.1 Two-body phase space

The effect of `SCPS` can already be studied with simple $2 \to 2$ scattering processes. The two-body phase space is either two dimensional or in case of hadronic collisions of dimension 4.

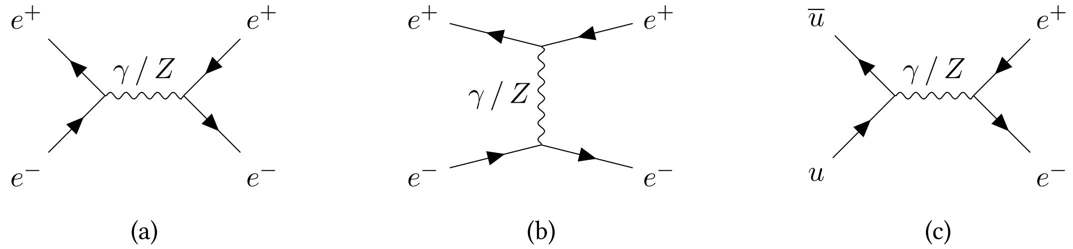


Figure 5.1: Simple $2 \to 2$ topologies with s- and t-channels.

Firstly, we consider the electron-positron collision `e+ e- > e+ e-` at a fixed collider energy of $500\,\mathrm{GeV}$. At tree level, the four topologies are the photon and Z boson s- and t- channels (a) and (b) in Figure 5.1. In the phase space factorization according to the s-channel topology (a), the phase space variables are the polar and azimuthal angle $\cos\theta$ and φ. In the t-channel topology (b), the phase space variables are the space-like Mandelstam invariant t and the azimuthal angle φ.

For our purpose of testing importance sampling of a single phase space parameterization, we integrate amplitudes where each is composed of only one single topology in (a) or (b) in Figure 5.1. The variances of the corresponding single-diagram integrands are listed in Table 5.1. As expected, the integration of the s-channel topologies does not benefit from SCPS since in this case there are no variables that are importance sampled. This also holds true for the massive t-channel, where the corresponding space-like invariant t is sampled uniformly. However, the variance is reduced in the massless t-channel when using SCPS, where the phase space variable t is importance sampled according to the massless distribution in Equation 4.2 with the choice $\nu = 1.1$. Hence, SCPS predominantly generates momentum configurations with small t, which in turn implies that the integrator more frequently probes the matrix element at momentum configurations with small scattering angles. Note, that a transverse momentum cut is applied to the massless t-channel to regularize the Coulomb divergence at $t = 0$.

		s-channel (5.1a)		t-channel (5.1b)	
		photon	Z boson	photon	Z boson
Naive	FlatPS	0.0603	0.0641	54.6773	23.6458
	SCPS	0.0625	0.0647	1.9543	24.0613
Vegas3	FlatPS	0.0022	0.0028	0.0027	0.0073
	SCPS	0.0025	0.0025	0.0013	0.0067

Table 5.1: Estimates of the normalized variance ${s_N}^2/{Q_N}^2$ of single-diagram amplitudes with the topologies of the process `e+ e- > e+ e-`.

Furthermore, in Table 5.1 we see that Vegas3 could reduce the variance by one order of magnitude for the s-channel topologies and by up to four orders of magnitude for the t-channel topologies. Since the phase space is only of dimension 2, the peak structure of the integrand was captured with relatively few points, which allowed the vegas grid to adapt quickly and even reproduce the advantage of SCPS over FlatPS in the photon t-channel.

As a second process, consider `u~ u > e+ e-`. The two tree level topologies (c) in Figure 5.1 correspond both to s-channels with either an interaction through a photon or a Z boson. Unlike the previous process, the invariant mass of the gauge bosons is not fixed here, since the colliding partons are each part of a hadronic system. Therefore, additional to the two phase space variables $\cos\theta$ and φ, we also have the variables τ and η, obtained from the partons' momentum fractions, the Bjorken variables x_1 and x_2. Since the center of mass energy is not fixed, the s-channel propagator can be on shell, leading to an enhancement at the pole mass. In Table 5.2, we see that this peak leads to a large variance, when the phase space configurations are probed uniformly by FlatPS. SCPS however reduced the variance by up to four orders of magnitude as it flattened the integrand by importance sampling of τ.[1] Note, that the integrand is a convolution with the parton distribution functions. For now, their shape is not accounted for by importance sampling.

		s-channel (5.1c)	
		photon	Z boson
`Naive`	`FlatPS`	7942.99	7187.45
	`SCPS`	8.33	0.76
`Vegas3`	`FlatPS`	0.47	0.08
	`SCPS`	0.35	0.02

Table 5.2: Estimates of the normalized variance ${s_N}^2/{Q_N}^2$ of single-diagram amplitudes with the topologies of the process `u~ u > e+ e-`.

The adaptive stratified sampling by `Vegas3` further reduced the variance one order of magnitude, as we see in Table 5.2. Again, the adaptive grid of `vegas` could reproduce the advantage of importance sampling. In Section 5.1.2, we show that already in the three-body phase space, with more variables to be importance sampled, `SCPS` finally outruns the combination of `Vegas3` with `FlatPS`, which in the two-body phase space still keeps up well.

It is useful to look at the massive s-channel topology in (c) of the previous process in more detail to understand how `SCPS` works. There, the Z boson propagator leads to a Breit-Wigner enhancement in the matrix element. Therefore, importance sampling is applied in such a way that this enhancement get cancelled by the transformation's Jacobian. The map in Equation 4.6 for τ or equivalently, for the squared invariant mass of the Z boson propagator $\hat{s}$, precisely comes with a Jacobian that flattens this propagator peak, which in turn implies that the Monte Carlo points are sampled according to a Breit-Wigner distribution in this variable, as we see in Figure 5.2. A simple example of this is the analogous situation of uniform sampling, where the distribution is constant and the corresponding map linear.

The shape the integrator sees can be visualized by plotting the total Monte Carlo weights w, the full integrand given in terms of random variables $\mathbf{r}$, for a large set of generated configurations. We reconstruct τ, the only importance sampled variable in the massive s-channel topology in (c), for all kinematic configurations and plot it against the corresponding values of the integrand. Since here, the parton distribution functions are not accounted for by an importance sampling prescription, it is also of interest how the shape changes by removing their contribution w_{PDF}. The upper two plots in Figure 5.3 illustrate the integrands according to the different parameterizations of `SCPS` and `FlatPS` without the contribution of the parton distribution functions w_{PDF}, where the lower two plots include w_{PDF}.

In (a) of Figure 5.3, we see that the Breit-Wigner peak is well pronounced, varying over multiple orders of magnitude. This is because `FlatPS` generates all momentum configurations with

[1] The importance sampling map for τ corresponds exactly to the map in Equation 4.6 applied to the squared partonic center of mass energy $\hat{s} = \tau s$ with squared collider energy s.

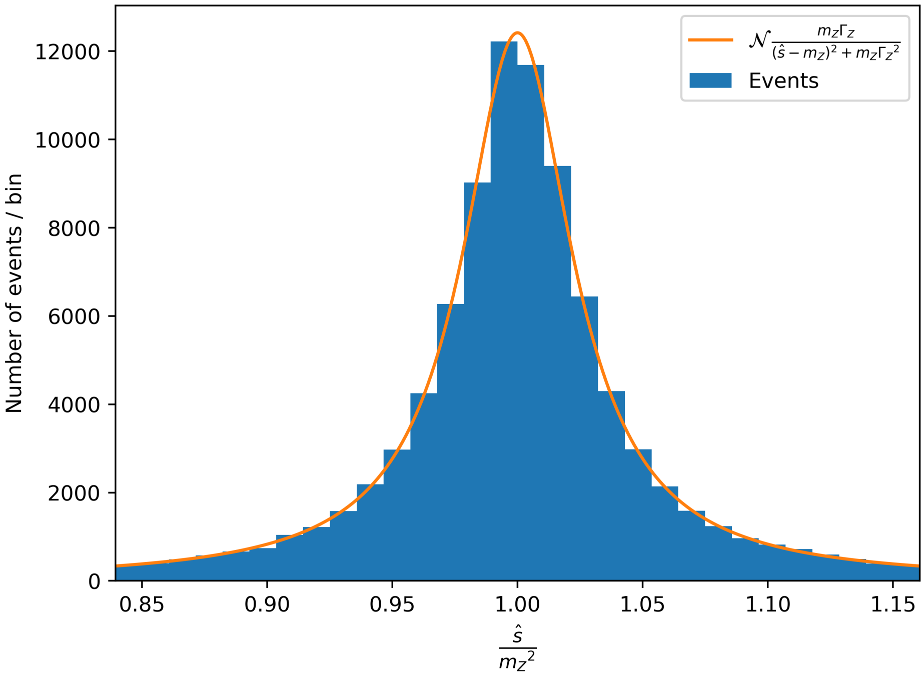


Figure 5.2: The squared invariant mass of the Z boson propagator $\hat{s}$, computed from a sample of 10^5 momentum configurations, generated with SCPS according to the massive s-channel topology in (c) in Figure 5.1 of the process `u~ u > e+ e-`. The distribution corresponds to a Breit-Wigner with mass m_Z and decay width Γ_Z, including a normalization factor $\mathcal{N}$.

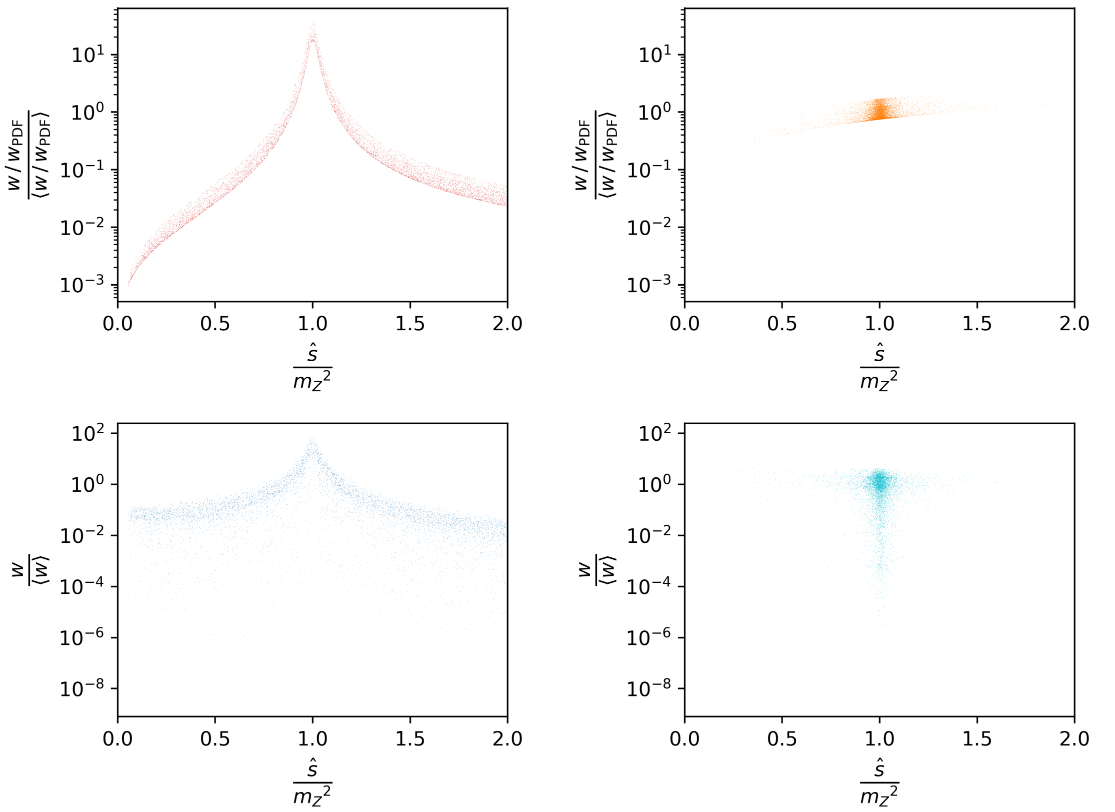
the same constant Jacobian, irrespective of the phase space region or the topology. However, SCPS accounts for the peak structure within the matrix element and manages to render the integrand relatively flat, as we can see in (b) of Figure 5.3. When including the parton distribution functions, we note that firstly, the integrand increases for low center of mass energies and secondly, that the angular distribution is washed out, which makes flattening harder for SCPS.

Figure 5.3 not only shows the shape of the integrand but also demonstrates how FlatPS and SCPS affect the distribution of Monte Carlo points in phase space. Clearly, SCPS predominantly samples around the Z boson mass, where FlatPS samples events uniformly. Note that although the parton distribution functions decrease the performance of SCPS, it is still able to substantially reduce the variance since the majority of events are distributed around unity.

Both aspects of importance sampling, the flattening of the integrand and the redistribution of Monte Carlo points in phase space, reduce the variance of the integrand and explain the drastic improvement in Table 5.2.

5.1.2 Three-body phase space

A 2 → 3 process is parametrized by five phase space variables, or seven in the case of hadronic collisions. The type of topology that describes this process is composed of either an s-channel and



(a) `Naive FlatPS`

(b) `Naive SCPS`

Figure 5.3: The total Monte Carlo weights w, normalized by the average $\langle w \rangle$. In the lower two plots, the parton distribution functions are included, where in the upper two plots, their contribution w_{PDF} is removed. The phase space is projected onto the dimension of the squared center of mass energy $\hat{s}$, in the region of the Z boson mass m_Z. This dimension corresponds to the phase space variable τ, sampled uniformly in (a) and according to the Z boson Breit-Wigner in (b). The three additional phase space dimensions correspond to η and to a polar and an azimuthal angle. Each plot is computed from 10^4 momentum configurations.

a t-channel or of two s-channels. Therefore, there are exactly two variables that are importance sampled by `SCPS`, except for the s-channel only collision with point-like particles, where only one variable is importance sampled.

We consider the two topologies (a) and (b) in Figure 5.4. They are contributions of the process `e+ e- > mu+ mu- a`. The topologies in (a) are each factorized into an s- and a t-channel, parametrized by five phase space variables, the space-like invariant $t = (p_{e^+} - p_\gamma)^2$, the time-like invariant $s = (p_{\mu^-} + p_{\mu^+})^2$ and three angles, where the topologies in (b) are each factorized into two s-channels, parametrized by four angles and the time-like invariant $s' = (p_\gamma + p_{\mu^+})^2$, with the external four-momenta p_{e^+}, p_{μ^-}, p_{μ^+} and p_γ.

In the topologies in (a), the squared invariant mass s of the photon or Z boson is not fixed by

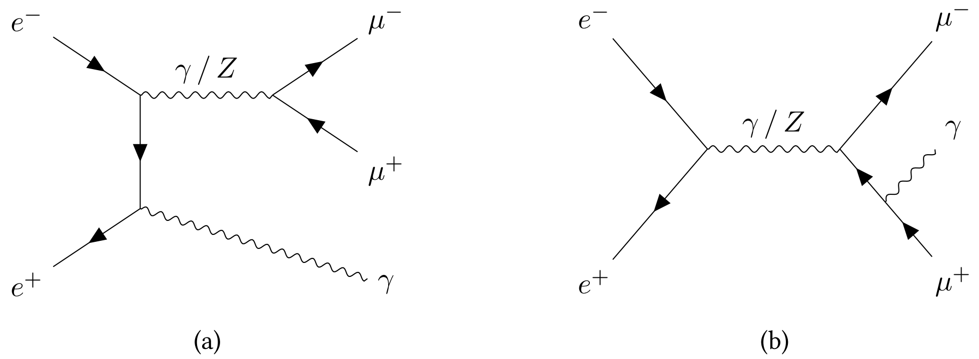


Figure 5.4: Simple $2 \rightarrow 3$ scattering topologies. Topology (a) contains both an s- and a t-channel and topology (b) only s-channels.

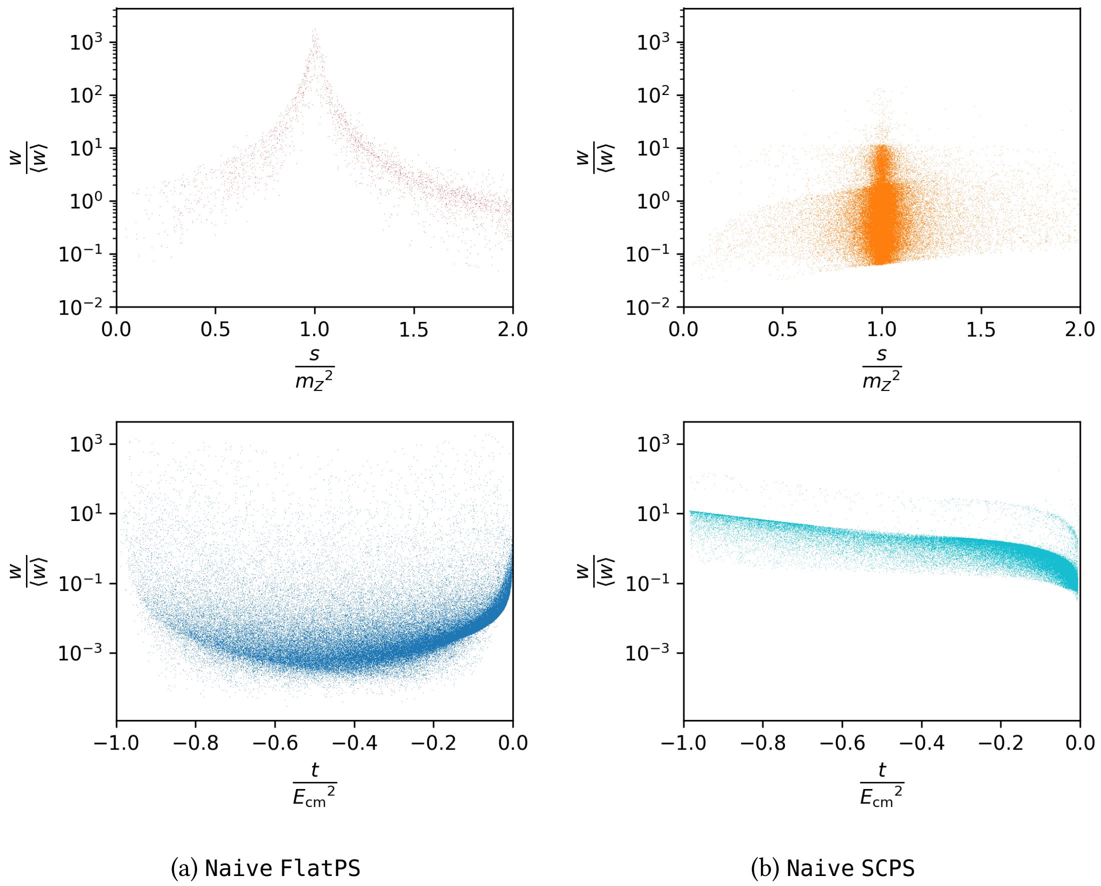
the center of mass energy. Therefore, SCPS generates predominantly momentum configurations with the bosons close to their mass shell. However, the topologies in (b) have fixed energies of the gauge boson propagators.

At a collision of $500\,\mathrm{GeV}$ the electrons are approximated to be massless. Therefore, importance sampling of the space-like invariant t for the topologies in (a) and the time-like invariant s' for the topologies in (b), predominantly produces external photons that are soft and collinear to the leg they are attached to. The poles of the single-diagram matrix element in the soft and collinear limit for each topology are regulated by imposing rapidity and angular separation cuts.

		s-channel and t-channel (5.4a)		only s-channels (5.4b)	
		photon	Z boson	photon	Z boson
Naive	FlatPS	223.7	469.4	487.4	65.9
	SCPS	18.4	17.8	3.9	4.1
Vegas3	FlatPS	36.0	337.2	37.7	29.9
	SCPS	2.1	0.4	1.5	3.0

Table 5.3: Estimates of the normalized variance ${s_N}^2/{Q_N}^2$ of single-diagram amplitudes with topologies of the process `e+ e- > mu+ mu- a`.

The estimates of the variance in Table 5.3 show that importance sampling works well with the topologies considered here. Note that SCPS importance samples only one phase space variable for the s-channel only topologies in (b) and is therefore more efficient than for the topologies in (a), where there are two variables to be importance sampled and therefore involve a more complicated peak structure. Furthermore, we see that other than in the two-body phase space, the few iterations of Vegas3 cannot keep up with SCPS. Keep in mind that a function with a large variance is not only more difficult to integrate but also the variance itself is harder to estimate than for a function with a small variance. For our purpose, only their order of magnitude is of importance, as we aim at drawing qualitative conclusions.



(a) `Naive FlatPS`

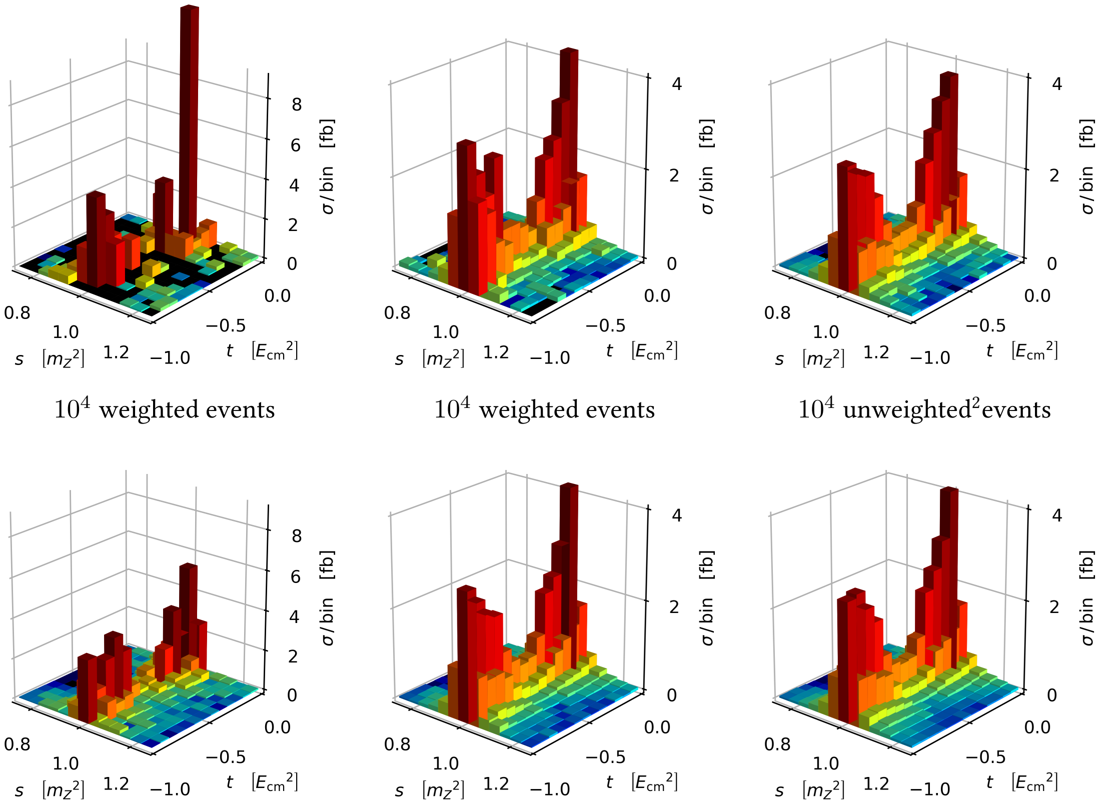
(b) `Naive SCPS`

Figure 5.5: The total Monte Carlo weights w, normalized by the average $\langle w \rangle$ of the Z boson t-channel topology of the process `e+ e- > mu+ mu- a`. The upper plots show a region around the Z boson mass m_Z, the projection of the phase space onto the dimension of the squared invariant mass of the Z boson propagator s. The lower plots show the projection onto the space-like invariant t of the full phase space with center of mass energy E_cm. The three additional phase space dimensions correspond to uniformly sampled angles. In total, the phase space includes 10^5 Monte Carlo points. The sharp edges are the projections of the phase space boundaries.

The Z-boson topology with the photon radiating from the initial leg exhibits a pronounced peak structure. We will look at this topology in more detail. The shape of its integrand can be illustrated by plotting the total Monte Carlo weights w, the full integrand given in terms of random variables $\mathbf{r}$, against various phase space variables. Of special interest are the importance sampled variables $s = (p_{\mu^-} + p_{\mu^+})^2$ and $t = (p_{e^+} - p_\gamma)^2$, the squared invariant mass of the gauge boson propagator and the squared transfer momentum, respectively.

Figure 5.5 shows that `FlatPS` is strongly affected by the Breit-Wigner peak. For values of t close to zero the massless electron enhancement can also be recognized, where the pole is regulated by integration cuts. With a sample of 10^5 phase space configurations, `FlatPS` still manages to capture those peaks and therefore obtains a reasonable estimate of the cross section.

Nevertheless, the estimate of the variance is large.



10^4 weighted events

10^4 weighted events

10^4 unweighted[2]events

10^5 weighted events

10^5 weighted events

10^5 unweighted events

(a) `Naive FlatPS`

(b) `Naive SCPS`

(c) `MadEvent`

Figure 5.6: The cross section σ of the Z boson t-channel topology of the process `e+ e- > mu+ mu- a` per bin of fixed Z boson invariant s and space-like invariant t, in units of ${m_Z}^2$ and ${E_{\mathrm{cm}}}^2$, the squares of the Z boson mass and the center of mass energy, respectively. The dependencies on the other phase space variables are averaged over. Here, one unweighted event roughly corresponds to 10 weighted events.

The upper plots in Figure 5.5 show clearly that compared to `FlatPS`, `SCPS` samples events much more frequently around the Z boson mass, with an approximately uniform total weight w across the full phase space. The effect of importance sampling of t is less apparent than the sampling of s. However, we see in the bottom plots that for values of t close to zero, the propagator enhancement is flattened by `SCPS` by the map in Equation 4.2, where we chose the parameter $\nu = 1.1$. Interestingly, the plot on the bottom right in Figure 5.5 suggests that a smaller ν would have been favorable, so that the enhancement is not overcompensated.

Note that the variance is affected much less by normalized Monte Carlo weights $w/\langle w\rangle$ of

[2] Unweighting is a procedure that trims a set of kinetic configurations according to their Monte Carlo weights w to a set of events with equal weights, leading to the same distribution.

order less than 1. Weights of order 10 or higher have much larger contributions to the variance, what we infer upon inspection of Equation 3.3. Therefore, the spread of higher order weights w make the difference in the variance. Also, the sharp edges in the distributions in Figure 5.5 are artifacts of the phase space boundaries of the dimensions that are projected out.

Consider now the histograms in Figure 5.6 that show the cross section, estimated using `FlatPS`, `SCPS` or `MadEvent`, the default leading order integrator of `MG5aMC`. Although, the overall estimate for the cross section can be shown to be consistent between the phase space generators, the estimates on smaller regions of phase space, which have large contributions to the cross section, are heavily affected by fluctuations due to insufficient sampling. Upon inspection of (b) in Figure 5.6, we see that thanks to importance sampling, `SCPS` agrees well with the predictions by `MadEvent` in (c), already with the smaller sample of 10^4 phase space configurations. However, `FlatPS` requires at least a sample of 10^5 phase space configurations to make useable predictions, as we can see in (a). Since `FlatPS` generates events uniformly over phase space, only few kinetic configurations fall into the Breit-Wigner region of the phase space, which gives the largest contribution to the cross section, and therefore probes it poorly. Poor sampling in this region causes large fluctuations in the estimates within each bin that are usually averaged out, if probed with enough events.

5.2 Multi-channeling

So far, only matrix elements with one single diagram contribution were considered. In general however, physical cross sections have multiple diagrammatic contributions and interference effects. This leads to a more complicated peak structure of the integrand, which aggravates numerical integration. Our knowledge about peak structures of single topologies motivates the idea of splitting the cross section up into channels, each corresponding to one structure. For each channel, a phase space parametrization and a corresponding sampling of the integration variables can be found that accounts for its peak structure. A numerical integrator then deals with multiple channels that are roughly flat instead of one highly varying function. In this section, we show the effects of this method and conclude that multi-channeling is required to produce accurate predictions for physical cross sections.

The following multi-channel process are investigated by two types of multi-channeling strategies. The strategy introduced in Section 3.2 is implemented within `MG5aMC` as the adaptive multi-channel integrator `MCVegas`, developed in the context of this thesis. It makes use of `MCPS`, a phase space generator that is able to switch between the parameterizations of the involved channels and comes with adaptive channel weights $\boldsymbol{\alpha}$. On top of that, each channel is optimized with stratified sampling in `vegas`. The other strategy is introduced in the following section and corresponds to what is used in `MadEvent`. As before, the channels make use of `vegas`' adaptive stratified sampling but do not come with channel weights $\boldsymbol{\alpha}$.

For comparison, we show integral estimates by `Naive` and `Vegas3`, two integrators that do not rely on multi-channeling. Here, they both get their phase space configuration by `FlatPS`, the phase space generator that generates momentum configurations uniformly across phase space. Note that applying `SCPS` in combination with a single-channel integrator like `Naive` or `Vegas3` to a multi-channel process is usually not a promising strategy since multiple peak structures cannot possibly be accounted for by importance sampling of a single phase space parametrization. `SCPS` would successfully cover one specific enhancement in phase space at the expense of neglecting all other peak structures.

There are several aspects that determine the efficiency of a Monte Carlo integrator. Upon inspection of Equation 3.4, we note that the number of Monte Carlo points needed to reach a certain accuracy of the estimate of the integral scales with the variance of the integrand. Obviously, the fewer points needed, the more efficient. The variance of the integrand therefore serves as a metric of how efficient the integration is. Since an adaptive integrator changes the variance over several iterations, it additionally matters by how much the variance was reduced for a given number of events and iterations. In the following tests of the two adaptive integrators `MCVegas` and `MadEvent`, the number of Monte Carlo points was doubled after each iteration.

5.2.1 Diagram enhancement technique

`MadEvent` has a different approach to multi-channeling. Consider a process with a cross section

$$I = \int \mathrm{d}\boldsymbol{\phi}\, f(\mathbf{k}(\boldsymbol{\phi})). \tag{5.1}$$

Note that the integrand f is proportional to the matrix element

$$|\mathcal{M}|^2 = \Big|\sum_{j=1}^{n} A_j\Big|^2, \tag{5.2}$$

with n diagrams A_j. The integral can then be split into n channels as

$$I = \sum_{i=1}^{n} I_i = \sum_{i=1}^{n} \int \mathrm{d}\boldsymbol{\phi}\, |A_i(\mathbf{k}(\boldsymbol{\phi}))|^2 \frac{f(\mathbf{k}(\boldsymbol{\phi}))}{\sum_j |A_j(\mathbf{k}(\boldsymbol{\phi}))|^2}. \tag{5.3}$$

We expect the the fraction $f/\sum_j |A_j|^2$ to be roughly flat, since the dominant peak structures of each diagram cancel each other. However, note that the interference terms are not accounted for. Each channel I_i is then estimated numerically using the parametrization $\boldsymbol{\phi}_i$ of a single-channel phase space generator that flattens the diagram A_i, according to

$$I_i = \int \mathrm{d}\mathbf{r}\, \frac{|A_i(\mathbf{k}(\boldsymbol{\phi}_i(\mathbf{r})))|^2}{g_i(\mathbf{r})} \frac{f(\mathbf{k}(\boldsymbol{\phi}_i(\mathbf{r})))}{\sum_j |A_j(\mathbf{k}(\boldsymbol{\phi}_i(\mathbf{r})))|^2}. \tag{5.4}$$

The numerical estimate of I is the sum of the estimates of the channels, where the square root of their Monte Carlo errors, summed in quadrature, is the expected error of integration.

The diagram enhancement approach flattens the integrand f by the matrix elements of the single-channel topologies. In the multi-channeling method introduced in Section 3.2 however,

f is flattened by just the Jacobians of each channel's transformation and the adaptive weights $\boldsymbol{\alpha}$. Flattening with the matrix elements has one big advantage over the Jacobian method, since they are able to not only compensate the peaks with their exact scaling factors but also other dependencies, where the Jacobians do not know about any absolute order of magnitude of the respective matrix element. The weights $\boldsymbol{\alpha}$ can however additionally adapt to interference effects and unforeseen structures, that neither the phase space generator, nor a single-diagram matrix element can account for. Furthermore, it can be of disadvantage if we have to know each matrix element of each diagram. For example, the full matrix element can be obtained recursively without knowing the single-diagram contributions [9, 10]. Keep in mind however that flattening with Jacobian requires additional computation, since the phase space generator has to be inverted for each parametrization to compute the total weight $1/g_{\boldsymbol{\alpha}}$. Furthermore, note that although both multi-channeling methods classify channels based on the same idea of separating distinct peak structures, the integrand of a channel in the matrix element flattening method is different than the integrand of the corresponding channel in the Jacobian flattening method. Therefore, the variance of the integrand and the value of the integral are not equal for corresponding channels of the two methods.

5.2.2 Two methods

In this section, we compare the two multi-channeling methods described in Section 5.2.1 and Section 3.2, the diagram enhancement technique and the Jacobian method. Both, the separate channels and the combined estimate of the full integral are of interest.

For simplicity, we construct a process consisting only of two topologies (a) in Figure 5.4. In both multi-channeling approaches, this process is split into two channels, one corresponding to the diagram A_γ with the intermediate photon propagator, the other to A_Z, the one with the Z boson propagator. In the matrix element method, the channels are the integrals

$$I_\gamma^{ME} = \int \mathrm{d}\mathbf{r}\, \frac{|A_\gamma^\gamma|^2}{g_\gamma(\mathbf{r})} \frac{|A_\gamma^\gamma + A_Z^\gamma|^2}{|A_\gamma^\gamma|^2 + |A_Z^\gamma|^2} \qquad \text{and} \qquad I_Z^{ME} = \int \mathrm{d}\mathbf{r}\, \frac{|A_Z^Z|^2}{g_Z(\mathbf{r})} \frac{|A_\gamma^Z + A_Z^Z|^2}{|A_\gamma^Z|^2 + |A_Z^Z|^2}, \tag{5.5}$$

with the full integral $I = I_\gamma^{ME} + I_Z^{ME}$, the sum of the channels. The superscripts γ and Z are shorthand notation for the dependence on the kinematic configurations $\mathbf{k}(\boldsymbol{\phi}_\gamma(\mathbf{r}))$ and $\mathbf{k}(\boldsymbol{\phi}_Z(\mathbf{r}))$ of A_γ, A_Z and the total weight $g_{\boldsymbol{\alpha}} = \alpha_\gamma g_\gamma + \alpha_Z g_Z$. When flattening with the combined weight of Jacobians, the channels are

$$I_\gamma^{Jac} = \int \mathrm{d}\mathbf{r}\, \frac{|A_\gamma^\gamma + A_Z^\gamma|^2}{g_{\boldsymbol{\alpha}}^\gamma} \qquad \text{and} \qquad I_Z^{Jac} = \int \mathrm{d}\mathbf{r}\, \frac{|A_\gamma^Z + A_Z^Z|^2}{g_{\boldsymbol{\alpha}}^Z}, \tag{5.6}$$

with the full integral $I = \alpha_\gamma I_\gamma^{Jac} + \alpha_Z I_Z^{Jac}$, the weighted sum of the channels. The weights $1/g_\gamma$ and $1/g_Z$ are the Jacobians from importance sampling of the single-channel phase space parameterizations $\boldsymbol{\phi}_\gamma$ and $\boldsymbol{\phi}_Z$ for the matrix elements $|A_\gamma|^2$ and $|A_Z|^2$, respectively. Note that it is exactly this parametrization that distinguishes the two channels I_γ^{Jac} and I_Z^{Jac}.

		Q_N [fb]	ΔQ_N [fb]	N	${s_N}^2/{Q_N}^2$
Naive	I_γ^{ME}	17.3	0.9	5229	15.4
	I_Z^{ME}	61.0	1.2	15 988	6.0
Naive	I_γ^{Jac}	35.7	1.9	5356	15.3
	I_Z^{Jac}	122.7	2.1	15 944	4.6

Table 5.4: The photon and Z boson channels, according to two multi-channeling methods, evaluated by the nonadaptive integrator Naive. They correspond to the two topologies (a) in Figure 5.4 of an unphysical $2 \to 3$ scattering process. The estimate of the integral and its Monte Carlo error after N points are given in Femtobarn and are denoted by Q_N and ΔQ_N. By ${s_N}^2$ we denote the variance of the integrand from Equation 3.16. Here, uniform channel weights $\alpha_\gamma = 0.5$ and $\alpha_Z = 0.5$ are assumed. The estimates of the full integral $I = I_\gamma^{ME} + I_Z^{ME} = \alpha_\gamma I_\gamma^{Jac} + \alpha_Z I_Z^{Jac}$ match.

For the purpose of comparing nonadaptive multi-channeling, we assume uniform channel weights $\alpha_\gamma = 0.5$ and $\alpha_Z = 0.5$. The estimates by the integrator Naive are given in Table 5.4 for a similar number of Monte Carlo points. It turns out that the normalized variance of the channels very similar for both methods. Both multi-channeling methods therefore seem to be similarly efficient at flattening the integrand. Note the tendency that importance sampling works better with massive propagator enhancements than massless ones. This is because the single-diagram matrix element of the photon channel does not behave like the squared massless propagator but its shape is presumed to be captured by the parameter ν that appears in the corresponding map in Equation 4.2.

		Q_N [fb]	ΔQ_N [fb]	N	${s_N}^2/{Q_N}^2$
MadEvent	I_γ^{ME}	18.240	0.053	160 000	1.351
	I_Z^{ME}	61.100	0.128	160 000	0.700
	I	79.340	0.138	320 000	0.973
MCVegas	I_γ^{Jac}	45.389	0.241	148 793	4.200
	I_Z^{Jac}	98.541	0.165	200 452	0.565
	I	79.345	0.144	349 245	1.142

Table 5.5: The channels and the estimates of the full integral I after five iterations of two adaptive multi-channel integrators. The channels correspond to the two topologies (a) in Figure 5.4 of an unphysical $2 \to 3$ scattering process. The channel weights of MCVegas, converged to $\alpha_\gamma = 0.361$ and $\alpha_Z = 0.639$.

The full estimates by MadEvent and the newly implemented MCVegas integrator are given in Table 5.5. The former makes use of the matrix element flattening and the latter uses the combined Jacobian approach. They both adapt their vegas grid after five iterations, where the latter additionally updates the channel weights $\boldsymbol{\alpha}$. Comparing the variances of the channels with the ones obtained by `Naive` emphasizes the importance of an adaptive integrator. Regarding the estimate of the full integral, both multi-channeling methods perform similarly well.

		Q_N [fb]	ΔQ_N [fb]	N	${s_N}^2/{Q_N}^2$
`Naive`	I	81.8	2.8	326 275	3761.9
`Vegas3`	I	73.7	2.0	341 178	263.5

Table 5.6: Estimates of the full integral of an unphysical unphysical $2 \to 3$ multi-channel process with topologies (a) in Figure 5.4. The two integrators `Naive` and `Vegas3` make no assumptions about the integrand's structure. They do not split the integrand into channels and use Monte Carlo points provided by the flat phase space generator `FlatPS`. `Vegas3` applies adaptive stratified sampling.

No assumptions about the structure of the matrix element is made for the two estimates in Table 5.6. We see that `Naive` probes a highly varying integrand and consequently badly estimates the integral. Although `Vegas3` manages to reduce the variance considerably, it still returns a poor estimate. Note that without the right phase space parametrization and splitting the integrand into channels, it is impossible for `Vegas3` to adapt to the propagator enhancements. This affirms how essential multi-channeling is for efficient integration.

5.2.3 Channel weights

Unlike the `vegas` grid, the channel weights $\boldsymbol{\alpha}$ influence the value of the channel integral. Over several iterations, they converge to the optimal value, minimizing the variance of the combined integrand. Their value is affected mostly by the relative factors between the diagrams. This means that a small channel will have a smaller weight than a large channel. Note however that the update prescription for the weight α_i requires an estimate of W_i, the integral of the square of the channel integrand. The variance is proportional to this quantity. Therefore, the channel weights $\boldsymbol{\alpha}$ are also affected by the variance of the channel integrands. However, badly estimated variances lead to inaccurate update prescriptions, so that the updated channel weights fail to capture the relative factors between the diagrams. The updates of the channel weights $\boldsymbol{\alpha}$ can be improved by sampling strongly varying channels with a larger number of Monte Carlo points.

Consider the process `g g > e+ e- mu+ mu-` at leading order in the Higgs effective theory [2], where the top quark loop is shrunken into a single vertex. There are two topologies involved in this process, (a) and (b) in Figure 5.7, which correspond to a channel each, the photon and the Z boson channel. Although the topologies look very similar, the photon channel involves a second effective vertex, the coupling of the Higgs to the photons. In this two-channel process, the suppressed photon channel weight α_γ is therefore expected to be much smaller than Z boson channel weight $\alpha_Z = 1 - \alpha_\gamma$.

Table 5.7 shows the integration of the process `g g > e+ e- mu+ mu-` with a modified Higgs

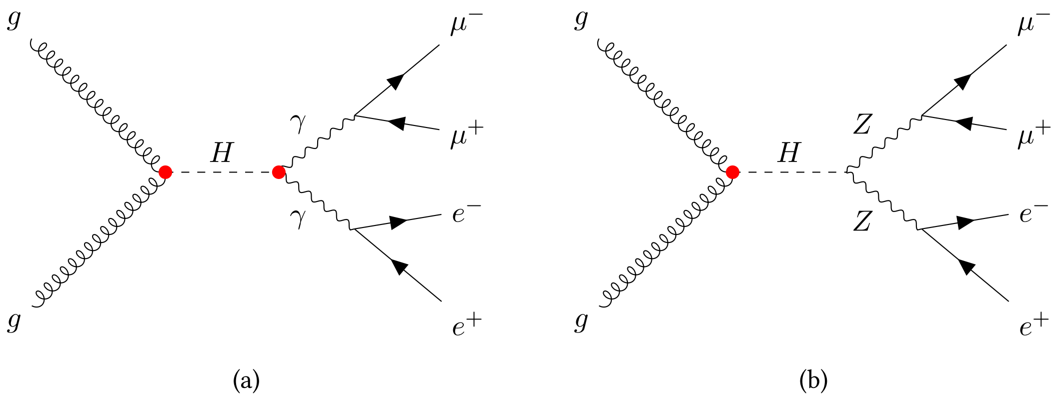


Figure 5.7: The tree level diagram contributions to the process `g g > e+ e- mu+ mu-` in the Higgs effective theory. The coupling of the gluons to the Higgs is mediated by a top quark loop and the coupling of the photons to the Higgs is mediated primordially by an electroweak loop.

mass of 200 GeV over a course of ten iterations. As the number of Monte Carlo points increases, the error decreases and the estimate of the integral converges. Note that the channel weights adapt as expected but do not converge within ten iterations. However, the variance has already converged to its minimal value. Note that in this process, the phase spaces of the channels are importance sampled differently but both parametrized with the same phase space variables. Therefore, the `vegas` grid of one channel is automatically aligned with the structures of both channels and can therefore adapt to the photon and Z boson enhancement simultaneously. Therefore,

	Q_N [fb]	ΔQ_N [fb]	N	${s_N}^2/{Q_N}^2$	α_γ	$\alpha_Z = 1 - \alpha_\gamma$
#0	755.888	45.395	423	1.526	0.500 000	0.500 000
#1	807.708	20.122	1479	0.918	0.099 285	0.900 715
#2	843.264	11.396	3054	0.558	0.020 573	0.979 427
#3	820.914	6.684	6133	0.407	0.006 907	0.993 094
#4	819.228	4.276	11 992	0.327	0.002 564	0.997 436
#5	812.288	2.868	23 647	0.295	0.002 082	0.997 918
#6	814.819	1.968	46 803	0.273	0.001 108	0.998 892
#7	813.765	1.376	93 149	0.267	0.000 444	0.999 556
#8	815.224	0.986	184 615	0.270	0.000 172	0.999 828
#9	814.513	0.695	364 794	0.266	0.000 065	0.999 935
#10	814.323	0.494	720 648	0.265	0.000 036	0.999 964

Table 5.7: Ten iterations of the adaptive multi-channel integrator `MCVegas`. The process `g g > e+ e- mu+ mu-` is divided into two channels with weights α_γ and α_Z. The Higgs mass is set to 200 GeV to avoid Breit-Wigner competition conflicts, as discussed in Section 4.3.1.

multi-channeling is not necessarily required and explains why the photon channel weight α_γ is basically turned off.

In general, a discrepancy between the channel weights, like in this process, is not expected. Therefore, it is sensible to start the integration with uniform weights. However, if one wants to be certain that the channel weights converge within a reasonable number of iterations, it is recommended to investigate the channels, their topologies and couplings beforehand and adjust the initial channel weights accordingly.

			Q_N [fb]	ΔQ_N [fb]	N	${s_N}^2/{Q_N}^2$
`Naive`	`FlatPS`	I	1.244	0.809	870 631	367 676.570
`Vegas3`		I	805.655	12.311	1 215 935	283.929
	`SCPS`	I	815.481	1.314	129 939	0.337

Table 5.8: Estimates of the cross section of the process `g g > e+ e- mu+ mu-`. Both integrators do not make use of multi-channeling and sample Monte Carlo points uniformly across the ten dimensional phase space.

Note that other than the previous multi-channel process, where the phase space was of dimension five, the process `g g > e+ e- mu+ mu-` is characterized by ten phase space variables. Generally, the higher the dimension of the integration region, the more events have to be generated to capture the structure of the integrand across the whole phase space. The benefit of importance sampling becomes much more evident here, since it focuses on the important regions with larger contributions to the cross section. In Table **??** we see that despite the large number of Monte Carlo points, the enhancements are completely missed by `Naive`. `Vegas3` estimates more successfully but cannot ideally adapt with the phase space parametrization provided by `FlatPS`. However, `SCPS` parametrizes the phase space identically for both topologies although with different importance sampling. It is therefore expected that `Vegas3` can adapt its grid accordingly, once it picks up on the enhancements.

5.2.4 Physical cross section

As a last example, we investigate the full physical $2 \to 3$ scattering process `e+ e- > mu+ mu- a`. It contains eight topologies, the four topologies shown in Figure 5.4 and the additional four, obtained by attaching the final photon to the other initial or respectively final fermion leg. Note that due to symmetries, these topologies correspond to four channels only. We will number the s-channel only topologies with the additional photon propagator by 1 and 3, and the ones with the Z boson propagator by 2 and 4. The topologies, involving both an s- and a t-channel are the numbers 5 and 7 with the additional photon propagator, and 6 and 8 with the Z boson propagator.

The results obtained by `MadEvent` and `MCVegas` are given in Table 5.9. `MadEvent` divides the integration into six channels, grouping topologies 1 and 3, and 2 and 4 into one channel each. That topoogies 5 and 7, and 6 and 8 are related by symmetries is not accounted for but

		Q_N [fb]	ΔQ_N [fb]	N	${s_N}^2/{Q_N}^2$	α_i
`MCVegas`	I_1^{Jac}	18.997	0.438	13 437	7.155	0.178
	I_2^{Jac}	19.090	0.367	13 251	4.896	0.201
	I_3^{Jac}	20.396	0.390	12 928	4.731	0.132
	I_4^{Jac}	20.872	0.537	13 194	8.744	0.183
	I_5^{Jac}	22.938	0.279	11 377	1.682	0.032
	I_6^{Jac}	47.268	0.254	13 043	0.377	0.121
	I_7^{Jac}	25.191	0.288	11 400	1.490	0.025
	I_8^{Jac}	46.270	0.250	13 144	0.383	0.128
	I	26.735	0.143	101 774	2.900	
`MadEvent`	I_{1+3}^{ME}	3.839	0.069	16 000	5.148	
	I_{2+4}^{ME}	0.533	0.008	16 000	3.388	
	I_5^{ME}	4.777	0.043	16 000	1.326	
	I_6^{ME}	6.218	0.132	16 000	7.254	
	I_7^{ME}	4.828	0.043	16 000	1.261	
	I_8^{ME}	6.102	0.062	16 000	1.671	
	I	26.297	0.173	96 000	4.157	

Table 5.9: Channels and the cross section of the process `e+ e- > mu+ mu- a` estimated after five iterations by two adaptive multi-channel integrators. The weight of channel i converged to α_i.

is confirmed by the channel results. `MCVegas` does not account for these symmetries yet. Here however, the channel weights did not converge symmetrically, so that the results obscure this hidden symmetry. This happens due to fluctuations that were only picked up in the final iteration and also explains the larger variances of channels 1 and 4. Large variances are expected to increase the channel weights and so do large values of the integral. This proportionality is observed in the final weights. Again, it stands out that the more massless propagators involved, the larger the variance of this channel. However, massive channels are also affected by fluctuations induced by the interference terms with topologies that include massless propagators.

		Q_N [fb]	ΔQ_N [fb]	N	${s_N}^2/{Q_N}^2$
`Naive`	I	27.2	1.0	233 135	325.0
`Vegas3`	I	26.4	0.4	291 372	73.3

Table 5.10: The cross section of the process `e+ e- > mu+ mu- a` estimated by two integrators that do not make any assumptions about the integrand.

It is of interest how integrators perform that do not make use of multi-channeling. Since they do not evaluate multiple channels but just one integrand, they usually have more points at their disposal. The results in Table 5.10 show the estimates of `Naive` and `Vegas3`, where the latter uses adaptive stratified sampling. Clearly, the multi-channel integrators are much more efficient since they not only benefit from importance sampling but also iteratively reduced the variance.

Chapter 6

Conclusions and Outlook

The goal of this thesis has been the introduction of numerical methods applied to the calculation of cross sections and their implementation. Cross sections are described by integrals on high-dimensional phase spaces and are therefore well suited for Monte Carlo simulations. Additionally, the generated events are of interest for further considerations concerning hadronization and the simulation of detectors. The purpose of the introduced methods is to improve the efficiency of the Monte Carlo integration.

We presented importance sampling, a method that allows to flatten integrands with variable transformation maps. Applied to tree level Feynman amplitudes, we showed that for single topologies a set of phase space factorization variables and specific transformations can be found that cancel enhancements of massive and massless propagators. To deal with more complicated peak structures, we introduced the strategy of multi-channeling, the division of the integrand into channels, where each channel can be flattened by importance sampling. The enhancements in the channels are then cancelled by the same transformation maps that flatten single-diagram amplitudes. Applying importance sampling and multi-channeling aims at reducing the variance of the integrand, which is the essential aspect of efficient integration.

From numerous tests, we learned about the benefits and the challenges of these methods and their implementation. We showed with processes with two and three final states that importance sampling of Breit-Wigner enhancements and massless propagator poles reduces the variance by multiple orders of magnitude. Furthermore, we demonstrated that importance sampling not only flattens the integrand but also affects the distribution of kinematic configurations across phase space. Both aspects make it possible to give better predictions of the cross section with fewer Monte Carlo points.

We showed that multi-channeling is an essential strategy to numerically integrate amplitudes, consisting of multiple diagrams by comparing with single-channel integrators. We found that our multi-channel approach is of similar efficiency as the current strategy of the default leading order integrator of `MG5aMC`.

Moreover, we incorporated `vegas` into the integration framework and stressed the importance of adaptive stratified sampling to iteratively account for unforeseen structures of the integrand.

Our implementation has proven to work well in general situations but leaves space for improvements. Over the course of this thesis, we stumbled over several issues that need to be improved or newly elaborated. In particular, the attempt of finding an optimal and general adaptation strategy of the integrator is an ongoing process. More concretely, some important aspects are:

- The phase space factorization, introduced in Section 2.2, correlates the upper and lower bounds of the time-like invariants s, as we see in Equation 2.6. This correlation is of disadvantage for two reasons:

  1. Breit-Wigner competition. The correlation of time-like invariants s depends on the ordering, in which the sampling is performed. In a competing Breit-Wigner scenario with a fixed ordering, importance sampling would favorably distribute the kinematic configurations into one particular region of phase space, neglecting the other important regions. It is necessary to enhance phase space generation with the functionality that recognizes all paths that account for Breit-Wigner competing structures.
  2. `vegas` grid. It is essential that the variables the `vegas` grid is identified with keep their same definition over the course of the integration. Consider the invariant s that is mapped to by a random variable r, defined on the `vegas` grid. The correlation between the boundaries and other invariants causes the limits s_{min} and s_{max} of s to differ form one phase space point to another. Since the limits are not constant and $r = 0$ and $r = 1$ are mapped to $s = s_{\text{min}}$ and $s = s_{\text{max}}$, the same random variable r is not necessarily mapped to the same value of s when generating multiple phase space points. In general, this would break successful adaptation of `vegas`. However, in the previous tests `vegas` still improved the variance, since the limits usually vary only little. Note that importance sampling of the Breit-Wigner enhancement predominantly generates the invariant close to the mass shell, where massless propagator sampling spreads the invariant on larger regions. Additionally, the correlations in the boundaries introduce unwanted dependencies in the Jacobians of the importance sampling transformations.

- In hadronic processes, the momenta of the colliding partons carry a fraction of the beam momenta, described by the Bjorken variables x_1 and x_2 and distributed according to the parton distribution functions. The parton distribution functions have a characteristic shape, increasing at low Bjorken fractions and suggesting to apply suitable importance sampling.

- Similarly to how several Feynman diagrams can share the same topology, several topologies can be connected via symmetries. These symmetries should be recognized before performing the multi-channel integration, so that it is not spoiled with channel duplicates.

- `vegas` provides a functionality that subdivides the integration space, apart from the grid, into hypercubes, where in each hypercube a separate Monte Carlo integration is performed. This helps to further reduce the variance and helps in the situation where `vegas` adapts to phantom peaks. This functionality has not been implemented yet within the adaptive multi-channel integrator `MCVegas`.

- We aim at implementing a flexible integrator that is capable of responding to a variety of situations. It is therefore of interest to provide a functionality that allows to combine our multi-channel approach, making use of Jacobians and channel weights $\boldsymbol{\alpha}$, with the diagram enhancement technique that is currently implemented within `MadEvent`. Ideally, we would have the option to set a flattening technique for the multi-channel integrand, which specifies for each channel if it either uses the Jacobian weight or the single-diagram matrix element and whether or not it comes with an adaptive channel weight α_i. Additionally, each channel can choose its own single-channel Monte Carlo integrator.

- Obviously, the efficiency of a program is primarily determined by its run time. It is therefore essential to make optimal use of all available resources. Part of future projects is to make our algorithm multi-core ready. Smart parallel processing would also provide the user with valuable information about the current state of the computation.

- It turns out that the most time consuming parts of the computation is the evaluation of the matrix elements and the phase space generation. It is therefore necessary that these tools are implemented in a lower level programming language such as `C` or `Fortran`.

Appendix A

Implementation

The strategies to improve the efficiency of Monte Carlo simulations, in particular the multi-channel method and importance sampling, as described in Chapters 3 and 4, are implemented inside a new module within the framework of `MG5aMC`. The following tools are written in `Python` so that they can be easily understood and revised in further development. Here, we outline their general structure and important aspects of the algorithms.

A.1 Phase space generator

Importance sampling of the integration variables is implemented within a new phase space generator. The class `SingleChannelPhasespace` handles the phase space generation for single tree level topologies. It is structured as:

`class SingleChannelPhasespace`

A daughter class of `VirtualPhaseSpaceGenerator`. In the initialization, additional to the arguments of the mother class, it specifies the parameters:

- ***`model`*** An instance that specifies the parameters of the particles in the model and the couplings between each other.
- ***`topology`*** A list of s- and t-channels that uniquely defines a topologies.
- ***`path`*** A list of indices that together with the topology specify the phase space decomposition and the ordering of importance sampling. By default, it uses a path that is generated randomly.

Importance sampling is all about redistributing integration variables. This is mapping `r`, a number between 0 and 1 to a variable `v` between the lower and upper bound `min` and `max` with a Jacobian `dv` according to some distribution. Within `SingleChannelPhasespace` these distributions are:

`uniform_distr` *`(r, min, max)`*

Uniform distribution. Returns `v`, `dv`.

inv_uniform_distr *(v, min, max)*

Inverse uniform distribution. Returns `r`, `dv`.

massless_distr *(r, min, max, nu=1.1, m2=0)*

Distribution for a propagator with vanishing width from Equations 4.2 and 4.4. The default choice of `m2` corresponds to a massless distribution. To avoid numerical problems when `min=0`, `m2` is automatically set to a small negative number. Generally, `nu` should be chosen $\gtrsim 1$. Returns `v`, `dv`.

inv_massless_distr *(v, min, max, nu=1.1, m2=0)*

Inverse distribution for a propagator with vanishing width. Choose the optional parameters `nu` and `m2` as above. Returns `r`, `dv`.

massive_distr *(r, mass, width, min, max)*

Distribution for a Breit-Wigner propagator from Equation 4.6. Specify M and Γ with the parameters `mass` and `width`, respectively. Returns `v`, `dv`.

inv_massive_distr *(v, mass, width, min, max)*

Inverse distribution for a Breit-Wigner propagator. Choose the parameters `mass` and `width` as above. Returns `r`, `dv`.

Furthermore, we have the class functions:

inv_t *(p_2, p1_2, p2_2, p3_2, p4_2, cos_theta)*

Computes the space-like invariant `t` as in Equation 2.13, where `p_2` is s, `p1_2` is p_+^2, `p2_2` is p_-^2, `p3_2` is s_1, `p4_2` is s_2 and `cos_theta` is $\cos\theta^*$. Returns `t`.

cos_theta_from_inv_t *(p_2, p1_2, p2_2, p3_2, p4_2, t)*

Computes `cos_theta` from the invariant `t` with the parameters defined as above. Returns `cos_theta`.

angles_to_rotate_along_z *(p)*

Computes the orientation angles `phi` and `theta` of the `LorentzVector` `p` in space. `theta` is the polar angle from the z-axis and `phi` the azimuthal angle within the xy-plane, measured from the x-axis. Returns `phi`, `theta`.

rotate_along_z_inv *(p, phi, theta)*

Defines `l` as the `LorentzVector` `p` after it was rotated in space, first with `theta` around the y-axis, then with `phi` around the rotated z' axis. Returns `l`.

get_cm_momenta *(s, p3_2, p4_2, cos_theta, phi)*

Generates the two final momenta `p3` and `p4` in the center of mass frame with squared energy `s` and their respective squared masses `p3_2` and `p4_2`. They are rotated with polar and azimuthal angles `cos_theta` and `phi`. Also computes the two-body phase space volume weight `wgt_PS`. Returns `p3`, `p4`, `wgt_PS`.

`get_two_body_PS_wgt` *(s, p3_2, p4_2)*

Computes the same two-body phase space volume weight as above, for a configuration with squared center of mass energy `s` and squared masses of final states `p3_2` and `p4_2`.

`get_random_path`

Generates a random path from the topology, specified as an instance parameter. `path` is a 2-tuple of lists. The first list specifies a sequence of s-channel vertices. Each vertex corresponds to a distributed invariant. The second list specifies a sequence of t-channel vertex pairs. A pair of t-channel vertices defines a 2 → 2 scattering block and a uniform invariant. For both types of channels, the sequence defines the order in which the distributed invariants are sampled. Returns `path`.

The current implementation does not account for all possible paths. For example, an s-channel vertex can only be added to the sequence if it already contains all vertices that follow. This strategy cannot account for competing Breit-Wigner structures. The algorithm can be described as:

```
#consider s-channels first
flag the final legs of a topology as available
if there are no t—channels:
    flag both outgoing legs of the initial s—channel vertex as unavailable
    #this vertex has no corresponding invariant to sample
if two available legs connect to a vertex:
    flag this vertex as available
while there are available vertices:
    for a randomly chosen available vertex:
        flag its outgoing legs as unavailable
        flag the incoming leg as available
        flag this vertex as unavailable
        add this vertex to list s-channel_path
        if an available leg connects with the newly available leg to a vertex:
            flag this vertex as available

#consider t-channels now
call first and last t—channel vertex start and end
choose a random vertex i between start and end
add (start,i) to list t-channel_path unless start=i
add (i+1,end) to t-channel_path unless i+1=end
for (start,end) in t-channel_path:
    choose a random vertex i between start and end
    add (start,i) to t-channel_path unless start=i
    add (i+1,end) to t-channel_path unless i+1=end
reverse t-channel_path

return [s-channel_path,t-channel_path]
```

The phase space generator SCPS uses a set of random variables **r** to generate a momentum configuration **k**, parametrized by a specified set of phase space variables ϕ. This main purpose of `SingleChannelPhasespace` is handled by the following function:

`get_PS_point` *`(input_variables, path=None)`*

This function distinguishes between two modes. If `input_variables` is a set of random variables between 0 and 1, it generates a phase space configuration `PS_point`, a `LorentzVectorList` of initial and final momenta. By default, `path` is identified with the one the class was initialized with, unless specified otherwise. `PS_point` is obtained by the parametrization of importance sampled random variables in the lab frame. The phase space configuration is additionally characterized by the phase space weight `wgt`, the combination of both the importance sampling Jacobians and the phase space volume factors, and the Bjorken variables `xb_1` and `xb_2`, both set to 1 in point-like particles collisions. Returns `PS_point`, `wgt`, `xb_1`, `xb_2`.

The second mode, reconstruction mode, implements the inverse of the above functionality. If `input_variables` is a phase space configuration, a `LorentzVectorList`, it generates its corresponding phase space weight `wgt` and the parametrizing random variables `r`. Returns `r`, `wgt`.

In the current implementation, the Bjorken variables cannot be reconstructed. Also, the reconstruction of the random variables that correspond to angles is omitted since it is not required, which makes multiple Lorentz boosts dispensable. Note that reconstruction mode is only needed for multi-channeling to obtain the phase space weight for a given momentum configuration.

The algorithm, in the case where it generates a phase space configuration, is presented below. Reconstruction mode is analogous in many aspects.

```
#determine collision type
if parton−parton collision:
    set limits of tau
    if there are no t−channels:
        importance sample tau according to initial s−channel propagator
    otherwise sample tau uniformly
    set limits of eta and sample uniformly
    update wgt
    obtain xb_1 and xb_2 from tau and eta
otherwise set tau and eta to one
set initial momenta
if there are no t−channels:
    set momentum of initial s−channel propagator

#generate distributed invariants
for vertex in s-channel_path:
    assign invariant s to incoming propagator
    set limits of s and importance sample accordingly
    update wgt
```

```
#generate uniform invariants
for vertex pair in t-channel_path:
    identify outgoing momenta of t−channel block
    assign invariant s to t−channel block
    set limits of s and sample s uniformly
    update wgt

#generate t-channel momenta
call first and last t−channel vertex start and end
add (start,end) to t-channel_path
reverse t-channel_path
for vertex pair in t-channel_path:
    identify incoming and outgoing momenta of t−channel block
    assign invariant t to t−channel block
    set limits of t and sample uniformly or according to massless distribution
    update wgt
    obtain polar angle from t
    sample azimuthal angle uniformly
    update wgt
    get outgoing momenta in center of mass frame
    update wgt
    identify Lorentz frame of initial momenta
    rotate and boost outgoing momenta to this Lorentz frame
    obtain transfer momentum

#generate s-channel momenta
for vertex in s−channels, from inner to outer:
    identify incoming momenta of s−channel block
    sample polar and azimuthal angle uniformly and update wgt
    get outgoing momenta in center of mass frame
    update wgt
    boost outgoing momenta with velocity of incoming particle

put initial and final momenta into PS_point, a LorentzVectorList
return PS_point,wgt,xb_1,xb_2
```

A.2 Multi-channel integrator

From Equation 3.4, we infer that the smaller the variance of the integrand, the more efficient the integrator. If its algorithm is adaptive, the integrator is more efficient the quicker it can reduce the variance of the integrand.

When integrating matrix elements of scattering processes, it is therefore natural to split up the integrand into channels, where each channel has its own peak structure. Since the propagators that lead to the peaks are known, the channel can be flattened by importance sampling of the right phase space variables, as we demonstrate in Section 3.2. On top of that the vegas grid and the adaptive channel weights α are updated over several iterations and further decrease the variance of the channel. The estimates are then combined to obtain the cross section and its Monte Carlo error according to Equations 3.15 and 3.16.

The integration of tree level matrix elements relies on a multi-channel phase space generator and a multi-channel integrator. The integrator handles the channels and combines them into a final estimate of the cross section.

The class `MultiChannelPhasespace` implements the multi-channel phase space generator. It is structured as:

class MultiChannelPhasespace

A daughter class of `VirtualPhaseSpaceGenerator`. In the initialization, additional to the arguments of the mother class, it specifies the parameters:

model An instance that specifies the parameters of the particles in the model and the couplings between each other.

topologies A list of topologies. A topology is a list of s- and t-channels that uniquely define a topology.

For every topology, the possible peak structures can be obtained by finding the paths to walk through this topology. Each peak structure then corresponds to a single-channel phase space generator that is initialized as an instance of `SingleChannelPhasespace`, which takes care of appropriate sampling of the phase space variables. Mulit-channeling however, assigns a different weight to each phase space point. This is accounted for with the following class function:

get_PS_point *(r, adaptive_wgts=none, channel_nr=none)*

Uses a set of random variables `r` between 0 and 1 to generate a phase space configuration `PS_point` and the Bjorken variables `xb_1` and `xb_2` for the channel with number `channel_nr`. Its multi-channel phase space weight `wgt` is characterized by g_α from Equation 3.13. If `adaptive_wgts` and `channel_nr` are not specified, a random channel with uniform adaptive channel weights is chosen. This is not recommended. The multi-channel phase space weight is computed according to Equation 3.13. Returns `PS_point`, `wgt`, `xb_1`, `xb_2`.

The integration is organized by the the class `MultiChannelVegas3Integrator`. It describes a Monte Carlo integrator that deals with multiple channels and adapts iteratively to the integrand.

class MultiChannelVegas3Integrator A daughter class of `VirtualIntegrator`. In the initialization, additional to the argument `integrand` of the mother class, it specifies the parameters:

adaptive_wgts A set α of normalized initial channel weights. By default, they are chosen uniform for all channels.

The integrator keeps track of the progress within each channel. While doing so, it knows the number of phase space points used in total and that passed cuts. The current estimate of the integral and its Monte Carlo error are updated continuously, as the channels are estimated in parallel. This allows to give a best estimate of the full integral at any time.

The integration relies on the following class functions:

survey *(neval, max_nitn, min_nitn=5, a_acc=0.01, v_acc=0.01)*

Improves the channel weights $\boldsymbol{\alpha}$ and the `vegas` grid iteratively. The survey starts by distributing a number of `neval` Monte Carlo points onto channels and doubles them with every iteration. It runs for at least `min_nitn` iterations and ends either at a maximum of `max_nitn` iterations or if both the channel weights and the `vegas` grid vary by less than their respective accuracies of `a_acc` and `v_acc`. The default accuracies are set to 1%. The adaptation of the `vegas` grid or the channel weights can be turned off with the setting `-1` for `v_acc` or `a_acc`, respectively.

adapt_vegas *(alpha)*

Adapts the `vegas` grid to a set of training data, the previously evaluated integrand points and determines the relative change of the grid `v_acc`. The parameter `alpha` is exactly the adaptation rate parameter used within `vegas`. The setting `alpha=0.5` turns out to be a good choice. Returns `v_acc`.

adapt_wgts

Adapts the channel weights $\boldsymbol{\alpha}$ according to the prescription in Equation 3.25 and determines the relative change of the weights `a_acc`. Returns `a_acc`.

refine *(rel_acc)*

Refines the estimate of the integral until a relative accuracy `rel_acc` is reached. Each channel is refined with a number points that is estimated continuously.

integrate

Combines the survey and refinement stage and computes the final estimate `result` and its Monte Carlo error `sdev`. Returns `result`, `sdev`.

Each channel is described by an instance of the class `Channel`.

class Channel

Gathers the information of one channel. In the initialization it is specified with the parameters:

i The number i that characterizes this channel.

full_integrand The full integrand f in Equation 3.14

adaptive_wgt This channel's adaptive weight α_i.

The channel keeps track of the used random numbers, the values of integrand and the squared integrand, and the number of Monte Carlo points that passed the cuts. Equipped with this info, the channel integral can be estimated and its Monte Carlo error calculated.

Although all channels are initialized with the same function f in Equation 3.14, each channel has its own integrand:

`integrand` *`(r, adaptive_wgts)`*

Computes the value of the integrand `res` at the point `r` for this channel. The parameter `adaptive_wgts` is the set of all channel weights. Over the course of the integration, the channel weights are modified. Therefore, the integrand of the channel is updated as well. Returns `res`.

`add_point` *`(adaptive_wgts)`*

Evaluates the integrand that depends on `adaptive_wgts` at a random point, according to the `vegas` grid. Additionally, it keeps track of the number of points that pass the phase space cuts.

`get_n_points` *`(abs_acc)`*

Estimates from the channel's current Monte Carlo error the number of additional points `n_points` needed to reach the target accuracy `abs_acc`. Returns `n_points`.

Bibliography

[1] R. Kleiss and R. Pittau, Comput. Phys. Commun. **83** (1994) 141 doi:10.1016/0010-4655(94)90043-4 [hep-ph/9405257].

[2] J. Alwall *et al.*, JHEP **0709** (2007) 028 doi:10.1088/1126-6708/2007/09/028 [arXiv:0706.2334 [hep-ph]].

[3] J. Alwall, M. Herquet, F. Maltoni, O. Mattelaer and T. Stelzer, JHEP **1106** (2011) 128 doi:10.1007/JHEP06(2011)128 [arXiv:1106.0522 [hep-ph]].

[4] A. Denner, S. Dittmaier, M. Roth and D. Wackeroth, Comput. Phys. Commun. **153** (2003) 462 doi:10.1016/S0010-4655(03)00205-4 [hep-ph/0209330].

[5] M. Roth, hep-ph/0008033.

[6] A. Denner, S. Dittmaier, M. Roth and D. Wackeroth, Nucl. Phys. B **560** (1999) 33 doi:10.1016/S0550-3213(99)00437-X [hep-ph/9904472].

[7] S. Dittmaier and M. Roth, Nucl. Phys. B **642** (2002) 307 doi:10.1016/S0550-3213(02)00640-5 [hep-ph/0206070].

[8] G. P. Lepage, vegas Documentation, Release 3.3.2, Jan 04, 2018, `https://github.com/gplepage/vegas`.

[9] T. Gleisberg and S. Hoeche, JHEP **0812** (2008) 039 doi:10.1088/1126-6708/2008/12/039 [arXiv:0808.3674 [hep-ph]].

[10] S. Actis, A. Denner, L. Hofer, A. Scharf and S. Uccirati, JHEP **1304** (2013) 037 doi:10.1007/JHEP04(2013)037 [arXiv:1211.6316 [hep-ph]].