HadCalc manual

Michael Rauch

January 2006

Contents

Chapter 1

Hadronic Cross Sections

The cross sections which are obtained by applying the Feynman rules contain, amongst other particles, quarks and gluons. The leading interaction between these particles is the strong interaction, which is described by quantum-chromo dynamics (QCD). This theory possesses two characteristic properties: asymptotic freedom [1] and confinement. Asymptotic freedom describes the behavior of the theory at small distances. In this region the interaction is weak and the coupling constant gets smaller with decreasing distance or, equivalently, with rising energy. At large distances confinement appears, because the interaction becomes strong and binds the particles tightly together. If the space between them becomes even larger, it is energetically favorable to form new quark–anti-quark pairs. One consequence of this behavior is that quarks and gluons cannot be observed as free particles, but only as constituents of hadrons, i.e. mesons, which are quark–anti-quark pairs, and baryons, which are states of three quarks or three anti-quarks. An example for these hadrons are protons, which are the colliding particles at the LHC. To make theoretical predictions it is necessary to relate the interactions at the parton level to the interactions at the hadron level [2]. The basis for doing this is the parton model [3], which will be described in the next section.

1.1 Parton Model

The parton model describes the inner structure of hadrons in hard collisions. It starts from the assumption that every observable hadron consists of constituents, the so-called partons, which can be identified as quarks and gluons. Experimental evidence for this assumption comes from the observation of scaling [4] in deep inelastic electron-proton-scattering. If the hadron carries some momentum P^μ, the partons which take part in the partonic subprocess have momentum xP^μ with $x \in [0, 1]$. As normally the mass of the hadrons is small compared to their kinetic energy one can assume $P^2 = 0$.

The interaction of an electron and a hadron or of two hadrons among each

other can be split into two parts. Because of Lorentz contraction and time dilation the interaction time of the two incoming particles in the laboratory frame is very short. Therefore effectively a static hadron is seen. For the hard scattering process interactions between partons of the same hadron need not be considered. Also the process of hadronization after the interaction happens on time scales which are much larger than the interaction itself.

From this the theorem of factorization [5] follows immediately. It states that all diagrammatic contributions to the structure functions can be separated into a product of two functions C and f, which depend on two mass scales μ_R and μ_F. μ_R is the renormalization scale, μ_F is the so-called factorization scale and separates the long-distance from the short-distance effects. Slightly simplifying one can say that every parton propagator which is off-shell by μ_F or more contributes to C, while those which are below this value contribute to f.

1.2 Integrated Hadronic Cross Sections

The hard scattering process C therefore can be calculated in perturbation theory by Feynman rules, using partons as incoming particles. It is independent of long-distance effects and especially from the type of the colliding hadron.

The parton distribution function (PDF) $f_{i/h}(x, \mu_F)$ contains the long-distance effects. It is independent of the underlying scattering process, but depends on μ_F and the type of hadron h. It is normalized such that it can also be interpreted as a probability density, namely the probability of finding the parton i in the hadron h with a momentum xP^μ. Its behavior as a function of the parameters is determined by the Altarelli-Parisi integro-differential equations [6]. Its numerical value, however, cannot be calculated a priori from the theory. At a single reference point it must be determined by experiments.

Therewith one obtains the expression [2]

$$\sigma_{pp\to fin+X} = \sum_{\{m,n\}} \int_{\tau_0}^{1} \mathrm{d}\tau \frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\tau} \hat{\sigma}_{mn\to fin}\left(\tau S, \alpha_s(\mu_R)\right) \tag{1.1}$$

for an integrated hadronic cross section with the parton luminosity

$$\begin{aligned}\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\tau} = \int_{\tau}^{1} \frac{\mathrm{d}x}{x} \frac{1}{1+\delta_{mn}} \cdot \\ \cdot \left(f_{m/p}\left(x, \mu_F\right) f_{n/p}\left(\frac{\tau}{x}, \mu_F\right) + f_{n/p}\left(x, \mu_F\right) f_{m/p}\left(\frac{\tau}{x}, \mu_F\right)\right).\end{aligned} \tag{1.2}$$

Here $\sqrt{S}$ denotes the hadronic center-of-mass energy, i.e. the one of the two colliding protons, and $\hat{\sigma}_{mn\to fin}$ the partonic cross sections of the subprocesses, where the two incoming partons m and n produce some final state, labeled fin. The sum includes all possible parton combinations m and n where the order of

appearance is not taken into account. The integration variable τ relates the partonic and hadronic center-of-mass energies with each other. More specifically, $\sqrt{\tau}$ can be interpreted as the part of the hadronic center-of-mass energy which takes part in the partonic subprocess, as the partonic center-of-mass energy is given by $\sqrt{\hat{s}} = \sqrt{\tau S}$. The lower limit of the integral τ_0 is determined by the kinematic configuration. $\sqrt{\tau_0 S}$ is the minimal energy which is necessary to produce the final state fin and therefore denotes the production threshold.

The formula given above is valid for processes with two or more particles in the final state. For hadronic cross sections it is also possible to calculate integrated cross sections for $2 \to 1$ processes. One first obtains for the partonic cross section of the process $mn \to f$

$$\mathrm{d}\hat{\sigma}_{mn\to f} = \frac{\pi}{4p_f^0\sqrt{\hat{s}}|\vec{p}_m|}|\mathcal{M}_{fi}\left(mn \to f\right)|^2\delta\left(p_m^0 + p_n^0 - p_f^0\right) \quad . \tag{1.3}$$

Again m and n specify the incoming partons, f denotes the outgoing particle, m_f its mass, and p_i^0 the energy of the respective particle i. $\vec{p}_m$ indicates the three-momentum of particle m in the partonic center-of-mass system and $\mathcal{M}_{fi}$ is the matrix element.

When convoluting with the parton distribution functions the single remaining δ-function in the partonic cross sections solves the τ integral in eq. (1.1) analytically. Thus one obtains for the integrated hadronic cross section

$$\sigma_{pp\to f} = \sum_{\{m,n\}} \left.\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\tau}\right|_{\tau=\frac{m_f^2}{S}} \frac{\pi}{2m_f S|\vec{p}_m|}|\mathcal{M}_{fi}\left(mn \to f\right)|^2 \quad . \tag{1.4}$$

1.3 Differential Hadronic Cross Sections

Additionally one can define hadronic cross sections that are differential in one or more parameters. For these parameters it is useful to take variables that are either invariant under Lorentz transformations or at least have very simple transformation properties. In this thesis three differential hadronic cross sections are presented which are also implemented in the HadCalc program that is described below in section 1.5. They are cross sections differential with respect to the invariant mass of the final-state particles, the rapidity of one final-state particle and, thirdly, the transverse momentum.

1.3.1 Invariant Mass

The first differential hadronic cross section is the one with respect to the invariant mass of the final-state particles. The invariant mass of a process is equivalent to the partonic center-of-mass energy $\sqrt{\hat{s}} \equiv \sqrt{\tau S}$ of the process or, in other words,

the sum of the final-state momenta of the outgoing particles. The differential cross section takes the form

$$\frac{\mathrm{d}\sigma_{pp\to fin}}{\mathrm{d}\sqrt{\hat{s}}} = 4\pi\frac{\sqrt{\hat{s}}}{S}\sum_{\{m,n\}}\left.\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\tau}\right|_{\tau=\frac{\hat{s}}{S}}\hat{\sigma}_{mn\to fin} \quad , \tag{1.5}$$

where fin again labeles a general final state.

1.3.2 Rapidity

The rapidity y of a particle is defined as

$$y = \operatorname{artanh}\frac{p_z}{p^0} \equiv \frac{1}{2}\ln\frac{p^0+p_z}{p^0-p_z} \tag{1.6}$$

where $p_z = \vec{p}\cdot c_\theta$ denotes the fraction of the particle's three-momentum $\vec{p}$ that goes in the direction of the beam axis, labeled z. The mass of the particle will later be referred to as m. Using the rapidity instead of directly taking the angle θ between the particle and the beam axis possesses some advantages because the rapidity of a particle has a few useful properties. Under a boost in the z-direction to a frame with a velocity β, the rapidity transforms as $y \to y - \operatorname{artanh}\beta$. Thus the shape of the rapidity distribution $\frac{\mathrm{d}\sigma}{\mathrm{d}y}$ stays unchanged. More generally, the sum of two rapidities when the momenta point into the same direction is given by the rapidity of the sum of the momenta, added via the formula for the relativistic addition of velocities: $y(p_1)+y(p_2) = y\left(\frac{p_1+p_2}{1+p_1p_2}\right)$. In experimental analyses often a slightly different measure, the pseudo-rapidity η, is used. It is derived from the standard rapidity by taking the limit of a vanishing mass of the particle and is defined as

$$\eta = \frac{1}{2}\ln\frac{1+c_\theta}{1-c_\theta} \quad . \tag{1.7}$$

In the HadCalc program both normal rapidity and pseudo-rapidity are implemented. As conversion between both variables can be performed by the simple transformation

$$y = \operatorname{artanh}\left(\sqrt{1-\frac{m^2}{\vec{p}^2+m^2}}\tanh\eta\right) \quad , \tag{1.8}$$

in the following only the shorter expressions for the standard rapidity are given. The ones for pseudo-rapidity can then be deduced from them.

Using the above-mentioned definition of the rapidity the differential hadronic cross section with respect to the rapidity for $2 \to 2$ processes then reads

$$\frac{\mathrm{d}\sigma}{\mathrm{d}y} = \int_{\tau_0}^{1}\mathrm{d}\tau\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\tau}\frac{\mathrm{d}\hat{\sigma}}{\mathrm{d}c_{\hat{\theta}}}\frac{\partial c_{\hat{\theta}}}{\partial y} \quad . \tag{1.9}$$

The momenta and masses given in the formulae always refer to the particle for which the rapidity distribution is calculated. The angle $c_{\hat{\theta}}$ between the particle and the beam axis in the partonic center-of-mass system is fixed by the relation

$$c_{\hat{\theta}} = \sqrt{1 + \frac{m^2}{\vec{\hat{p}}^{\,2}}} \tanh\left(y + \frac{1}{2}\ln\frac{x^2}{\tau}\right) \tag{1.10}$$

where the second term in the argument of tanh originates from the boost from the hadronic center-of-mass system, which is the laboratory frame, to the partonic one, in which the partonic subprocess is calculated. This leads to

$$\frac{\partial c_{\hat{\theta}}}{\partial y} = \sqrt{1 + \frac{m^2}{\vec{\hat{p}}^{\,2}}} \frac{1}{\cosh^2\left(y + \frac{1}{2}\ln\frac{x^2}{\tau}\right)} \quad . \tag{1.11}$$

For processes with three or more particles in the final state the formula is very similar. Additional phase-space integrals appear for the further particles but otherwise eq. (1.9) stays unchanged. In the following equation the differential cross section for a $2 \to 3$ process is given

$$\frac{\mathrm{d}\sigma}{\mathrm{d}y} = \int_{\tau_0}^{1} \mathrm{d}\tau \frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\tau} \int \mathrm{d}k_3^0 \int \mathrm{d}k_5^0 \int \mathrm{d}\hat{\eta} \frac{\mathrm{d}\hat{\sigma}}{\mathrm{d}k_3^0 \mathrm{d}c_{\hat{\theta}} \mathrm{d}k_5^0 \mathrm{d}\hat{\eta}} \frac{\partial c_{\hat{\theta}}}{\partial y} \quad . \tag{1.12}$$

The parametrization of the three-particle phase space is described in appendix A.2.

1.3.3 Transverse Momentum

The last implemented differential hadronic cross section is the one with respect to the transverse momentum $p_T = \sqrt{p_x^2 + p_y^2}$ of one of the final state particles. For $2 \to 2$ processes it is defined as

$$\frac{\mathrm{d}\sigma}{\mathrm{d}p_T} = \int_{\tilde{\tau}_0}^{1} \mathrm{d}\tau \frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\tau} \frac{\mathrm{d}\hat{\sigma}}{\mathrm{d}c_{\hat{\theta}}} \frac{\partial c_{\hat{\theta}}}{\partial p_T} \tag{1.13}$$

with

$$\frac{\partial c_{\hat{\theta}}}{\partial p_T} = \frac{1}{\sqrt{\frac{\vec{\hat{p}}^{\,4}}{p_T^2} - \vec{\hat{p}}^{\,2}}} \tag{1.14}$$

which follows from

$$c_{\hat{\theta}\pm} = \pm\sqrt{1 - \frac{p_T^2}{\vec{\hat{p}}^{\,2}}} \quad . \tag{1.15}$$

Here two possible solutions arise because of the sign ambiguity when taking the square root. In principle both solutions have to be taken into account and added

up unless they are excluded by other constraints as shown below. The lower limit of the τ-integral $\tilde{\tau}_0$ must be adjusted such that $s_{\hat{\theta}}$ is always inside its co-domain $[0;1]$

$$\tilde{\tau}_0 = \frac{\left(\sqrt{m_{f_1}^2 + p_T^2} + \sqrt{m_{f_2}^2 + p_T^2}\right)^2}{S} \quad , \tag{1.16}$$

f_1 and f_2 denoting the two final state particles.

For $2 \to 3$ processes the extension to include the third final-state particle is straightforward. The lower limit for τ in these processes is

$$\tilde{\tau}_0 = \frac{\left(\sqrt{m_{f_1}^2 + p_T^2} + \sqrt{\left(m_{f_2} + m_{f_3}\right)^2 + p_T^2}\right)^2}{S} \quad , \tag{1.17}$$

when the cross section is differential in the particle f_1. Therefore the expression for the differential cross section reads

$$\frac{\mathrm{d}\sigma}{\mathrm{d}p_T} = \int_{\tilde{\tau}_0}^1 \mathrm{d}\tau \frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\tau} \int \mathrm{d}k_3^0 \int \mathrm{d}k_5^0 \int \mathrm{d}\hat{\eta} \frac{\mathrm{d}\hat{\sigma}}{\mathrm{d}k_3^0 \mathrm{d}c_{\hat{\theta}} \mathrm{d}k_5^0 \mathrm{d}\hat{\eta}} \frac{\partial c_{\hat{\theta}}}{\partial p_T} \quad . \tag{1.18}$$

1.4 Cuts

In order to improve the ratio of the signal-process cross section to that of the background processes it can be useful to place appropriate cuts on the final-state particles. Also experimental techniques used in the reconstruction of events like jet-clustering algorithms can mandate the use of cuts in theoretical predictions, so that the behavior of these techniques is emulated there.

In the HadCalc program cuts on three different properties of the final-state particles are implemented [7]. The first two are cuts on the rapidity and the transverse momentum of a particle. The definition of these two variables was already presented in the previous section. The third one is a mutual property of two particles, the jet separation ΔR_{ij}, which is defined as

$$\Delta R_{ij} = \sqrt{\Delta y_{ij}^2 + \Delta \phi_{ij}^2} \quad . \tag{1.19}$$

Δy_{ij} denotes the rapidity difference between the two particles i and j and $\Delta \phi_{ij}$ the difference in the azimuthal angles of the two particles in the transverse plane. Its main use are exclusive hadronic cross sections where final-state jets shall be observed explicitly. It mimics the behavior of jet-clustering algorithms. There two jets, which are separated by a jet separation below a certain limit, are seen in the reconstruction as a single jet which has kinematic properties that are averaged over the two final-state partons.

For the first two cut parameters, rapidity and transverse momentum, it is possible to translate these cuts into a limit on the integration parameters of the phase space. The most general case is assumed here that cuts on both the rapidity y_{cut} and the transverse momentum $p_{T\,\text{cut}}$ of a particle shall be applied. Using eq. (1.15) the transverse-momentum cut can be translated into a cut on $c_{\hat{\theta}}$ and one obtains

$$c_{\hat{\theta}_{p_T}}^{\min} \equiv -\sqrt{1-\frac{p_{T\,\text{cut}}^2}{\vec{\hat{p}}^{\,2}}} < c_{\hat{\theta}} < \sqrt{1-\frac{p_{T\,\text{cut}}^2}{\vec{\hat{p}}^{\,2}}} \equiv c_{\hat{\theta}_{p_T}}^{\max} \quad . \tag{1.20}$$

Likewise, the cut on the rapidity can also be turned into a cut on $c_{\hat{\theta}}$ via eq. (1.10), yielding

$$\begin{aligned} c_{\hat{\theta}} &> \sqrt{1+\frac{m^2}{\vec{\hat{p}}^{\,2}}}\tanh\left(-y_{\text{cut}}+\frac{1}{2}\ln\frac{x^2}{\tau}\right) \equiv c_{\hat{\theta}_y}^{\min} \\ c_{\hat{\theta}} &< \sqrt{1+\frac{m^2}{\vec{\hat{p}}^{\,2}}}\tanh\left(y_{\text{cut}}+\frac{1}{2}\ln\frac{x^2}{\tau}\right) \equiv c_{\hat{\theta}_y}^{\max} \quad . \end{aligned} \tag{1.21}$$

To shorten the notation the abbreviation

$$r = \sqrt{\frac{1-\frac{p_{T\,\text{cut}}^2}{\vec{\hat{p}}^{\,2}}}{1+\frac{m^2}{\vec{\hat{p}}^{\,2}}}} \tag{1.22}$$

is used in the following. Again the momenta and mass used in the equations all refer to the particle whose phase space should be constrained.

Applying both cuts requires that the conditions on $c_{\hat{\theta}}$ are all fulfilled simultaneously. This also restricts the integral on x which appears in the parton luminosity given in eq. (1.2). In total the x-interval divides into five different regions, which will be labeled by roman numbers. First the two cases where both cuts cannot be fulfilled simultaneously, are considered, because the lower limit of one cut lies above the upper limit of the other one:

$$\text{Region I:} \qquad c_{\hat{\theta}_y}^{\max} \le c_{\hat{\theta}_{p_T}}^{\min} \qquad \Rightarrow \qquad x \le \sqrt{\tau}e^{-y_{\text{cut}}}\sqrt{\frac{1-r}{1+r}} \equiv x_{\text{I}} \tag{1.23}$$

$$\text{Region V:} \qquad c_{\hat{\theta}_y}^{\min} \ge c_{\hat{\theta}_{p_T}}^{\max} \qquad \Rightarrow \qquad x \ge \sqrt{\tau}e^{y_{\text{cut}}}\sqrt{\frac{1+r}{1-r}} \equiv x_{\text{V}} \quad . \tag{1.24}$$

These two regions are excluded and the cross section vanishes there.

For specifying the other regions first two special cases are considered, where the lower limits on $c_{\hat{\theta}}$ and the upper limits, respectively, coincide. For these cases

the according value of x is determined

$$c_{\hat{\theta}_y}^{\min} = c_{\hat{\theta}_{p_T}}^{\min} \qquad \Rightarrow \qquad x = \sqrt{\tau} e^{y_{\text{cut}}} \sqrt{\frac{1-r}{1+r}} \equiv x_{\min} \tag{1.25}$$

$$c_{\hat{\theta}_y}^{\max} = c_{\hat{\theta}_{p_T}}^{\max} \qquad \Rightarrow \qquad x = \sqrt{\tau} e^{-y_{\text{cut}}} \sqrt{\frac{1+r}{1-r}} \equiv x_{\max} \quad . \tag{1.26}$$

Using these two definitions the other intermediate regions can be specified. The ranges for $c_{\hat{\theta}}$ which are deduced from these following regions specify the allowed area where the cuts are fulfilled and therefore the cross section does not vanish. The next two regions handle the cases where the limits on $c_{\hat{\theta}}$ from rapidity and transverse momentum overlap and one limit is given by the rapidity cut and the other one by the transverse-momentum cut:

$$\text{Region II:} \qquad c_{\hat{\theta}_y}^{\min} \leq c_{\hat{\theta}_{p_T}}^{\min} < c_{\hat{\theta}} < c_{\hat{\theta}_y}^{\max} \leq c_{\hat{\theta}_{p_T}}^{\max} \quad \Rightarrow \quad x_{\text{I}} < x < \min(x_{\min}, x_{\max}) \tag{1.27}$$

$$\text{Region IV:} \qquad c_{\hat{\theta}_{p_T}}^{\min} \leq c_{\hat{\theta}_y}^{\min} < c_{\hat{\theta}} < c_{\hat{\theta}_{p_T}}^{\max} \leq c_{\hat{\theta}_y}^{\max} \quad \Rightarrow \quad \max(x_{\min}, x_{\max}) < x < x_{\text{V}} \quad . \tag{1.28}$$

Finally the definition of the last region is the case whether one cut gives a range on $c_{\hat{\theta}}$ that completely lies inside the other one. Depending on which cut this is, the limits on x are different:

$$\text{Region III a):} \qquad c_{\hat{\theta}_{p_T}}^{\min} \leq c_{\hat{\theta}_y}^{\min} < c_{\hat{\theta}} < c_{\hat{\theta}_y}^{\max} \leq c_{\hat{\theta}_{p_T}}^{\max} \quad \Rightarrow \quad x_{\min} < x < x_{\max} \tag{1.29}$$

$$\text{Region III b):} \qquad c_{\hat{\theta}_y}^{\min} \leq c_{\hat{\theta}_{p_T}}^{\min} < c_{\hat{\theta}} < c_{\hat{\theta}_{p_T}}^{\max} \leq c_{\hat{\theta}_y}^{\max} \quad \Rightarrow \quad x_{\max} < x < x_{\min} \quad . \tag{1.30}$$

In addition to those regions the original constraint for x for a hadronic cross section without cuts applies:

$$\tau < x < 1 \quad . \tag{1.31}$$

Combining the result of all regions one can see that no holes in the integration over x or $c_{\hat{\theta}}$ appear and the final borders of the integration routine can be simplified to

$$\max(\tau, x_{\text{I}}) < x < \min(x_{\text{V}}, 1) \tag{1.32}$$

and

$$\max(c_{\hat{\theta}_{p_T}}^{\min}, c_{\hat{\theta}_y}^{\min}) < c_{\hat{\theta}} < \min(c_{\hat{\theta}_{p_T}}^{\max}, c_{\hat{\theta}_y}^{\max}) \quad . \tag{1.33}$$

For a cross section which is differential with respect to the rapidity of a final state particle the cut on the transverse momentum yields a restriction on $c_{\hat{\theta}}$ in the same way as in eq. (1.33)

$$c^{\min}_{\hat{\theta}_{p_T}} < c_{\hat{\theta}} < c^{\max}_{\hat{\theta}_{p_T}} \quad . \tag{1.34}$$

The constraint on x must then be adjusted such that $c_{\hat{\theta}}$ is always inside this allowed interval, yielding

$$\max(\tau, \sqrt{\tau}e^{-y}\sqrt{\frac{1-r}{1+r}}) < x < \min(\sqrt{\tau}e^{-y}\sqrt{\frac{1+r}{1-r}}, 1) \quad , \tag{1.35}$$

which corresponds to eq. (1.32) where the rapidity cut y_{cut} is replaced by its value y given as an argument to the cross section.

Similarly, for cross sections that are differential in the transverse momentum of a final-state particle a cut on the rapidity puts a further constraint on the allowed interval for $c_{\hat{\theta}_\pm}$:

$$c^{\min}_{\hat{\theta}_y} < c_{\hat{\theta}_\pm} < c^{\max}_{\hat{\theta}_y} \tag{1.36}$$

with

$$c^{\min}_{\hat{\theta}_y} \equiv \sqrt{1+\frac{m_3^2}{\vec{\hat{p}}^{\,2}}}\tanh\left(-y_{\text{cut}} + \frac{1}{2}\ln\frac{x^2}{\tau}\right) \tag{1.37}$$

$$c^{\max}_{\hat{\theta}_y} \equiv \sqrt{1+\frac{m_3^2}{\vec{\hat{p}}^{\,2}}}\tanh\left(y_{\text{cut}} + \frac{1}{2}\ln\frac{x^2}{\tau}\right) \quad . \tag{1.38}$$

Again this leads to a corresponding change in the limits of the x-integration which are given by

$$\max(\tau, \sqrt{\tau}e^{-y_{\text{cut}}}\sqrt{\frac{1-\tilde{r}}{1+\tilde{r}}}) < x < \min(\sqrt{\tau}e^{y_{\text{cut}}}\sqrt{\frac{1+\tilde{r}}{1-\tilde{r}}}, 1) \tag{1.39}$$

with

$$\tilde{r} = \sqrt{\frac{1-\frac{{p_T}^2}{\vec{\hat{p}}^{\,2}}}{1+\frac{m_3^2}{\vec{\hat{p}}^{\,2}}}} \quad . \tag{1.40}$$

This again corresponds to eqs. (1.32) and (1.22) where instead of the cut on the transverse momentum $p_{T\,\text{cut}}$ its fixed value p_T, which is an argument to the differential cross section, is taken.

1.5 HadCalc

For the numerical evaluation of the cross sections presented in the following chapters a program called HadCalc was developed to facilitate this task. It is based on the established program packages FeynArts [8] and FormCalc [9, 10] which are used to generate the partonic cross sections. The main task of HadCalc then consists of the convolution with the PDFs that are taken from the PDFlib [11] or LHAPDF [12] library packages that include PDF fits from various groups.

With this program it is possible to calculate both totally integrated and differential hadronic cross sections of processes with up to three particles in the final state. The latter ones can be differential with respect to the partonic center-of-mass energy, or the rapidity or the transverse momentum of one of the outgoing particles. Several cuts can be applied to the phase space. HadCalc operates either in batch mode, where the parameters are read from a file and the cross sections are written back to disk, allowing for easy post-processing with e.g. a tool that generates plots. It can also be used in interactive mode where in- and output are done via keyboard and screen and which allows the user for example to tune the parameters most easily.

Chapter 2

Manual of the HadCalc Program

For the calculation of hadronic cross sections a computer code, called HadCalc, was written (see chapter 1.5). In this appendix the manual of the program is presented.

2.1 Prerequisites and Compilation

2.1.1 Prerequisites

The following programs are required for compiling and running HadCalc and must be installed:

- a Fortran compiler compliant with the Fortran77 standard,
- a C compiler conforming to ANSI-C,
- GNU make,
- FormCalc 4 [9],
- one of the two following packages that include sets of parton distribution functions from various groups
  - PDFLIB (CERN Computer Program Library entry W5051) [11], or
  - LHAPDF [12].

Additionally, support for the following two programs is integrated into HadCalc

- FeynHiggs 2.1beta or newer [13],
- Condor workload management system for compute-intensive jobs.

PDG flavor code	Particle
0	gluon g
1	down quark d
2	up quark u
3	strange quark s
4	charm quark c
5	bottom quark b
6	top quark t
-1	down anti-quark $\bar{d}$
-2	up anti-quark $\bar{u}$
-3	strange anti-quark $\bar{s}$
-4	charm anti-quark $\bar{c}$
-5	bottom anti-quark $\bar{b}$
-6	top anti-quark $\bar{t}$

Table 2.1: PDG flavor codes

2.1.2 Configuration and Compilation

First the partonic subprocess must be generated and prepared by following the instructions in the FormCalc4 manual. Especially the definitions in process.h have to be updated correctly as HadCalc relies on those. It is not necessary to fill in correct MSSM parameters or tune integration parameters, however.

Then the distribution file `HadCalc-0.5.tar.gz` should be unpacked. As next step change into its subdirectory and run `configure` from there. The following configure options are mandatory:

--with-partonprocess=DIR — This is the location of the FormCalc-generated partonic output.

--with-processtype=mn — By this option the processtype is fixed, specified by the number of incoming particles m and the number of outgoing particles n. Note that m and n form a single number, i.e. for a $2 \to 2$ process one would write --with-processtype=22. Currently, $2 \to 1$, $2 \to 2$ and $2 \to 3$ is implemented and can be entered here.

--with-parton1=i — The type of the first parton is specified by i, given as the PDG flavor code [14] (see table 2.1).

--with-parton2=i — Similarly, this is the PDG flavor code for the second parton.

Additionally the following options are recognized by configure and enable optional features:

--enable-antiproton1	Hadron 1 is an anti-proton instead of a proton.
--enable-antiproton2	Hadron 2 is an anti-proton instead of a proton.
--with-condor[=DIR]	Link the final code with the Condor workload management system libraries. If the binary is not in the standard search path of your shell, its location can be specified with the optional DIR argument.
--with-feynhiggs[=DIR]	Link the final code with the FeynHiggs library. This is mandatory if the FormCalc option to compute the MSSM-Higgs sector via FeynHiggs is chosen. The optional DIR specifies the location of the FeynHiggs library libFH.a, if it is not in the standard search path of the compiler.
--with-looptools=DIR	If LoopTools is not in the standard search path of the compiler, its location can be specified here.
--with-lhapdf[=DIR]	Use LHAPDF for the parton distribution functions. If the LHAPDF library is not in the standard search path, its location can be given by the optional DIR argument. The PDF data is assumed to be found at the same place.
--with-pdflib[=DIR]	Use PDFlib for the parton distribution functions. If the PDFlib library is not in the standard search path and the CERNlib environment variables $CERN and $CERN_LEVEL are not set, the DIR argument designates where it can be found.

Only one of the last two options can be given on the command line. If neither --with-lhapdf nor --with-pdflib was given, configure first tries to find LHAPDF and, if this fails, probes the existence of PDFlib.

After having run `configure`, a call to `make` compiles the program. When it successfully finishes, a binary called `HadCalc` has been created in the current path.

2.2 Running the program

The program is simply started by running `./HadCalc`. It will then present a menu which allows one to tune various settings and start the calculation of cross sections. The following subsections describe the possible settings in detail. An item is chosen by typing the number shown in brackets before the item and pressing "Enter". In every menu "(0)" exits the submenu or, for the top level menu, quits the program. Invalid input is ignored and an error message is written on the screen.

2.2.1 Physics parameters

This submenu sets the parameters of the MSSM and related things and is divided in three further submenus.

MSSM parameters

Here all values which correspond to parameters of the MSSM are set.

First let us look at menu item 16. This decides whether the program should use a common mass M_{SUSY} in the sfermion sector, or if individual values for the left-handed squarks and sleptons and the right-handed sups, sdowns and selectrons are allowed. Depending on this flag either the MS* variables cannot be set (because they are fixed at M_{SUSY}) or M_{SUSY} itself cannot be set (because it is irrevelant and not used in the computation). When choosing a common SUSY mass scale, the settings in the MS* variables are retained and restored when deselecting this option.

All other parameter settings can be in two states. They can either have a fixed setting, then this value is used for all calculations. Or their value can be running. In this case a lower and upper bound and the number of intermediate intervals must be chosen. Then the computation of the cross section is done ("intermediate intervals" + 1) times, with the value of the parameter increasing from lower bound to upper bound[1]. The distance between two values is equal for the setting "linear" and exponentially increasing for the setting "logarithmic", i.e. the values are closer at the lower bound and they have equal distance again when plotting them on a logarithmic scale. A behavior vice versa with values closer at the upper bound can be easily achieved by exchanging lower and upper bound. If more than one parameter is chosen to be running, the iteration loops are nested, with the first parameter varying fastest.

[1]Despite its name, the lower bound can be mathematically larger than the upper bound; then the value of the parameter is decreasing.

Kinematic parameters

In this menu all parameters are set which are related to kinematic variables of the process.

The underlying parameter of items 3 and 4 depends on the type of process. For processes with two particles in the final state, this is the angle θ between the two outgoing particles, for those with three final particles, it denotes the energy k_5^0 of the fifth particle, which is the third final-state particle. The menu items 8, 12, 14 and 15, which refer to the fifth particle, are ignored for $2 \to 2$ processes and cannot be changed.

The settings of the parameters are possible in the same way as already described in the previous item.

Scale parameters

This menu sets the renormalization and factorization scale of the process in the same way as described above. A negative number for the renormalization scale has a special meaning. Then the sum of the masses of the final-state particles is taken, multiplied with the absolute value of the setting, and this number is taken as the renormalization scale. Additionally it can be chosen that both renormalization and factorization scale are always set to the same value.

Show ModelDigest (FormCalc)

Finally this choice invokes FormCalc's ModelDigest function, which takes the parameters as an input and calculates the physical masses of the particles. Thereby it applies lower bounds on the masses established by experiment and refuses the calculation if these bounds are violated. The calculated cross section will also be zero in that case. The FormCalc manual contains a more detailed explanation of this function. There it is also described how the check for the violation of experimental bounds can be switched off by flipping a switch in FormCalc's process.h.

2.2.2 PDF parameters

The set used for the calculation of the parton distribution functions is chosen by this submenu. The layout and choices presented depends on whether LHAPDF or PDFlib is used. For PDFlib three numbers must be entered. The first denotes the type of parton distribution functions and is 1 for proton PDFs. The second number specifies the respective group which has performed the fit to the experimental data and the third number chooses a specific PDF set. When using LHAPDF a string must be entered that directly specifies the filename of the PDF set in the LHAPDF subdirectory.

2.2.3 Integration parameters

This submenu chooses the integration routine and sets its parameters. Currently there are six integration routines available:

GAUSSKR	One-dimensional Gauss-Kronrod algorithm
GAUSSAD	One-dimensional adaptive Gauss algorithm
DCUHRE	Multi-dimensional adaptive Gauss algorithm
VEGAS	Monte Carlo integration algorithm
SUAVE	Subregion adaptive Monte Carlo integration algorithm
DIVONNE	Monte Carlo integration via stratified sampling

The last four algorithms are part of the CUBA library [15]. In the following only a short description of the possible parameter settings is given. The technical details of these algorithms and the precise impact of the variables are described in the CUBA manual and shall not be repeated here.

The GAUSSKR and GAUSSAD algorithms can only handle one-dimensional integrals. If multi-dimensional integrals are attempted to be computed, VEGAS is used as a fallback. In contrast the DCUHRE and DIVONNE algorithms cannot handle one-dimensional integrals. There the GAUSSKR algorithm is used instead. In both cases a warning is printed on the screen.

All integration routines share these two variables:

- relative error: the desired relative error
- absolute error: the desired absolute error

Additionally, the following variables are available for one or more of the routines. Which ones these are is denoted in square brackets after the entry.

- maximum # of points: the maximum number of function evaluations used [GAUSSAD, DCUHRE, VEGAS, SUAVE, DIVONNE]
- # of points for starting: the initial number of points per iteration [VEGAS]
- increase in # of points: the number of points the previous number is incremented for the next iteration [VEGAS]
- # of points for subdivision: the number of points used to sample a subdivision [SUAVE]
- flatness # for splits: the type of norm used to compute the fluctuation of a sample [SUAVE]
- # of passes: the number of passes after which the partitioning phase terminates [DIVONNE]

- key 1: determination of sampling in the partitioning phase [DIVONNE]
- key 2: determination of sampling in the final integration phase [DIVONNE]
- key 3: sets the strategy for the refinement phase [DIVONNE]
- maximum χ^2 for subregion: the maximum χ^2 value a single subregion is allowed to have in the final integration phase [DIVONNE]
- minimum deviation for split: a bound which determines whether it is worthwhile to further examine a region that failed the χ^2 test [DIVONNE]

2.2.4 Amplitude switches

This submenu sets the type of diagrams used for the computation and how the cuts should be applied. The value of the cuts is set in the parameter section and was already described there.

The first entry decides whether the tree-level and the one-loop result shall be computed in one go or only one of them. Possible choices are "Tree only", "Tree+Loop" and "Loop only". Which way is better depends on the concrete circumstances and features of the problem. Computing both at the same time might save computation time, but the integration routine has to optimize its choices for both at the same time, which might lead to sub-optimal performance. On the other hand it is not too likely that there are problematic regions in the tree-level cross section which are no longer present in the one-loop computation, so normally this procedure gives satisfactory results. If only one cross section is computed, the value of the other one is set to zero.

The remaining entries decide if and how the cuts on rapidity, transverse momentum and jet separation should be applied. It is either possible to have the particle, or a pair of particles in case of the jet separation, fulfill a cut, violate it or ignore the cut altogether. Since HadCalc relies on FormCalc for the partonic process and implementation details, for the cuts for particle three in the $2 \to 2$ case and particle five in the $2 \to 3$ case it cannot be chosen that the rapidity and transverse momentum cut is violated, but they always have to be fulfilled. They can, however, be switched off by setting the relevant entry in the parameters section to zero.

2.2.5 Input/Output options

This submenu allows one to read in a set of parameters from a file and specify where and how to write the calculation output.

To read in a set of parameters a parameter specification must have been written into a file and this filename then has to be entered here. All possible variables which can be set in such a file are given in section 2.3. There are three

basic types of variables. Those which specify a parameter can either take four comma-separated values that are the lower and upper bound, the behavior with respect to increments, i.e. linear or logarithmic, and the number of intermediate intervals, or a single number denoting its fixed value. The ones of type *boolean* turn on a certain switch and take no arguments. All remaining ones take a single argument and the variable is assigned to this parameter.

In the following also a formal definition of the parsing rules is given:

- The file is read line by line.
- White space at the beginning of a line is ignored.
- Empty lines are ignored.
- Lines starting with the character "#" (after optional white space) are comment lines and ignored.
- The first token which is separated by white space from the rest of the line is extracted. This token has to be a token from the list of valid tokens in section 2.3.
- If the token type is boolean, its associated parameter is set.
- If the token type is integer, an attempt to read an integer value is made and if it succeeds, this is assigned to the associated parameter.
- If the token type is double, an attempt to read a double precision floating-point number is made and if it succeeds, this is assigned to the associated parameter.
- If the token type is string, the second token is assigned to the associated parameter.
- If the token type is parameter, the following actions happen:
  - An attempt to read four comma-separated double precision floating-point numbers is made.
  - If this attempt succeeds, the four numbers are assigned to lower bound, upper bound, log and number of intermediate intervals of the parameter. log means linear increase if this variable is zero and exponential one otherwise.
  - If this does not succeed, an attempt to read a single double precision floating point number is made.
  - If this succeeds, this number is the constant value of the parameter.
  - If this also does not succeed, the line is flagged as not parsable.

- For lines not parsable by the rules above a warning message is printed and their content is ignored.

Furthermore some integration routines offer the possibility to write intermediate results or progress report to the screen. This is turned on with *Verbose integration output*. For hadronic cross sections this also enables writing PDFlib statistics on the screen at the end of the calculation.

Finally one can choose whether the calculation results will be written to the screen or into a file. In the latter case the variable *outputstring* describes which elements should be written to the output file. The form of this variable together with the valid tokens is described in section 2.4. The output-file format starts with a "#"-quoted header with a file identification and the content of *outputstring*. Then, each on a line by itself, for every scanned parameter point the values defined in *outputstring* are written, separated from each other by a space.

2.2.6 Amplitude calculation

This submenu finally allows one to choose the cross section one wants to compute and does the calculation. During the following integration the process may be interrupted with "Ctrl-C", after which it aborts the current calculation and jumps back into the main menu. Due to restrictions imposed by Condor this feature is not available if HadCalc was configured with the option --with-condor. Here pressing "Ctrl-C" aborts the whole HadCalc program.

2.3 Allowed tokens in input files

The following list shows all token names that may appear in an input file together with its associated type. The tokens are not case-sensitive. Thereby *parameter* means that the variable can either be followed by four comma-separated values that denote the lower and upper bound, whether the increase is linear or logarithmic, and the number of intermediate intervals, or a single number that is the fixed value of this parameter. *boolean* means that a specific behavior is switched on. There is a corresponding separate token that switches the same behavior off again. *double* and *integer* tokens take a single double-precision or integer value, respectively, as input. *string* assigns the remainder of the line to the parameter. Finally *preselected* takes special values as an argument. The possible choices for each of these ones were discussed during the description of the menus given in section 2.2. Any settings referring to particle 5 are relevant only for $2 \to 3$ processes and will be silently ignored otherwise.

token	: type	description
MA0	: *parameter*	mass of the CP-odd Higgs boson
TB	: *parameter*	ratio of the Higgs vacuum expectation values
MUE	: *parameter*	μ parameter in the Higgs sector
MSusy	: *parameter*	common SUSY mass scale
MSQ	: *parameter*	mass parameter of the left-handed squarks
MSU	: *parameter*	mass parameter of the right-handed sup-like squarks
MSD	: *parameter*	mass parameter of the right-handed sdown-like squarks
MSL	: *parameter*	mass parameter of the left-handed sleptons
MSE	: *parameter*	mass parameter of the right-handed selectron-like sleptons
A_t	: *parameter*	trilinear coupling of the sup-like squarks
A_b	: *parameter*	trilinear coupling of the sdown-like squarks
A_tau	: *parameter*	trilinear coupling of the selectron-like sleptons
M_1	: *parameter*	$\mathrm{U}(1)_Y$ gaugino mass
M_2	: *parameter*	$\mathrm{SU}(2)_L$ gaugino mass
MGl	: *parameter*	$\mathrm{SU}(3)_C$ gaugino mass
SQRTS	: *parameter*	square root of the hadronic center of mass energy
SQRTSHAT	: *parameter*	square root of the partonic center of mass energy
THETA [2]	: *parameter*	angle between the two outgoing particles (in degrees) [3]
THETACUT [2]	: *parameter*	cut on the angle between the two outgoing particles (in degrees)
K50 [3]	: *parameter*	energy of the third outgoing particle
K50CUT	: *parameter*	cut on the energy of the third outgoing particle
PTRANS [3]	: *parameter*	transverse momentum
PTRANS3CUT	: *parameter*	cut on the transverse momentum of particle 3
PTRANS4CUT	: *parameter*	cut on the transverse momentum of particle 4

[2] only for $2 \to 2$ processes

token	: type	description
PTRANS5CUT	: *parameter*	cut on the transverse momentum of particle 5
RAPID [3]	: *parameter*	rapidity
RAPID3CUT	: *parameter*	cut on the rapidity of particle 3
RAPID4CUT	: *parameter*	cut on the rapidity of particle 4
RAPID5CUT	: *parameter*	cut on the rapidity of particle 5
DELTAR34CUT	: *parameter*	cut on the distance between particles 3 and 4
DELTAR35CUT	: *parameter*	cut on the distance between particles 3 and 5
DELTAR45CUT	: *parameter*	cut on the distance between particles 4 and 5
RENSCALE	: *parameter*	renormalization scale
FACTSCALE	: *parameter*	factorization scale
CommonSUSYMassScale	: *boolean*	choose a common SUSY mass scale
NoCommonSUSYMassScale	: *boolean*	do not choose a common SUSY mass scale
CommonRenFactScale	: *boolean*	choose a common remormalization and factorization scale
NoCommonRenFactScale	: *boolean*	do not choose a common remormalization and factorization scale
AMPLITUDE	: *preselected*	choose which amplitude(s) to calculate
Ptrans3>cut	: *boolean*	require the transverse momentum of particle 3 to be larger than the cut
Ptrans3<cut	: *boolean*	require the transverse momentum of particle 3 to be smaller than the cut
Ptrans3nocut	: *boolean*	disable cut on the transverse momentum of particle 3
Rapid3>cut	: *boolean*	require the rapidity of particle 3 to be larger than the cut
Rapid3<cut	: *boolean*	require the rapidity of particle 3 to be smaller than the cut
Rapid3nocut	: *boolean*	disable cut on the rapidity of particle 3
Ptrans4>cut	: *boolean*	require the transverse momentum of particle 4 to be larger than the cut

[3] only for differential cross sections

token	: type	description
Ptrans4<cut	: *boolean*	require the transverse momentum of particle 4 to be smaller than the cut
Ptrans4nocut	: *boolean*	disable cut on the transverse momentum of particle 4
Rapid4>cut	: *boolean*	require the rapidity of particle 4 to be larger than the cut
Rapid4<cut	: *boolean*	require the rapidity of particle 4 to be smaller than the cut
Rapid4nocut	: *boolean*	disable cut on the rapidity of particle 4
DeltaR34>cut	: *boolean*	require the jet separation between particles 3 and 4 to be larger than the cut
DeltaR34<cut	: *boolean*	require the jet separation between particles 3 and 4 to be smaller than the cut
DeltaR34nocut	: *boolean*	disable the cut on the jet separation between particles 3 and 4
DeltaR35>cut	: *boolean*	require the jet separation between particles 3 and 5 to be larger than the cut
DeltaR35<cut	: *boolean*	require the jet separation between particles 3 and 5 to be smaller than the cut
DeltaR35nocut	: *boolean*	disable the cut on the jet separation between particles 3 and 5
DeltaR45>cut	: *boolean*	require the jet separation between particles 4 and 5 to be larger than the cut
DeltaR45<cut	: *boolean*	require the jet separation between particles 4 and 5 to be smaller than the cut
DeltaR45nocut	: *boolean*	disable the cut on the jet separation between particles 4 and 5
INTMETHOD	: *preselected*	choose the integration routine
EPSABS	: *double*	absolute integration error
EPSREL	: *double*	relative integration error
MAXPTS	: *integer*	maximum number of points
VSTARTPTS	: *integer*	number of points for starting
VINCREASE	: *integer*	increase in number of points
SNNEW	: *integer*	number of points for subdivision

token	: type	description
SFLATNESS	: *integer*	flatness number for splits
MAXDPASS	: *integer*	number of passes in partitioning phase
DKEY1	: *integer*	Divonne key 1
DKEY2	: *integer*	Divonne key 2
DKEY3	: *integer*	Divonne key 3
DBORDER	: *double*	border of the integration region
MAXDCHISQ	: *double*	maximum χ^2 for subregion
MINDDEV	: *double*	minimum deviation for split
VERBOSITY	: *integer*	verbosity of integration output
PDFTYPE	: *double*	type of the PDF [PDFlib]
PDFGROUP	: *double*	group of the PDF [PDFlib]
PDFSET	: *double*	set of the PDF [PDFlib]
PDFPATH	: *string*	path where the PDF files are [LHAPDF]
PDFNAME	: *string*	name of the PDF [LHAPDF]
ScreenOutput	: *boolean*	print output on the screen
OUTPUTFILE	: *string*	print output into file
OUTPUTSTRING	: *string*	parameters to print in output (see section 2.4)

2.4 Allowed variable names for *outputstring*

The following list shows all variable names that may appear in *outputstring*. The individual entries are separated from each other by spaces. Variables with the dimension of a mass are output in GeV. Note that these names are case-sensitive.

Name	: Parameter	description
MA0	: m_{A_0}	mass of the CP-odd Higgs boson
TB	: $\tan\beta$	ratio of the Higgs vacuum expectation values
MUE	: μ	μ parameter in the Higgs sector
MSusy	: m_{SUSY}	common SUSY mass scale
MSQ	: $m_{\tilde{q}}$	mass parameter of the left-handed squarks
MSU	: $m_{\tilde{u}}$	mass parameter of the right-handed sup-like squarks
MSD	: $m_{\tilde{d}}$	mass parameter of the right-handed sdown-like squarks
MSL	: $m_{\tilde{l}}$	mass parameter of the left-handed sleptons
MSE	: $m_{\tilde{e}}$	mass parameter of the right-handed selectron-like sleptons
A_t	: A_t	trilinear coupling of the sup-like squarks

Name	: Parameter description	
`A_b`	: A_b	trilinear coupling of the sdown-like squarks
`A_tau`	: A_τ	trilinear coupling of the selectron-like sleptons
`M_1`	: M_1	$\mathrm{U}(1)_Y$ gaugino mass
`M_2`	: M_2	$\mathrm{SU}(2)_L$ gaugino mass
`MGl`	: $m_{\tilde{g}}$	gluino mass
`SQRTS` [4]	: $\sqrt{S}$	square root of the hadronic center-of-mass energy
`SQRTSHAT` [5]	: $\sqrt{\hat{s}}$	square root of the partonic center-of-mass energy
`THETA` [6]	: θ	angle between the two outgoing particles (in degrees)
`THETACUT` [6]	: θ_{cut}	cut on the angle between the two outgoing particles (in degrees)
`K50` [7]	: k_5^0	energy of the third outgoing particle
`K50CUT` [7]	: $k_{5\,cut}^0$	cut on the energy of the third outgoing particle
`PTRANS`	: p_{trans}	transverse momentum
`PTRANS3CUT`	: $p_{transcut}^3$	cut on the transverse momentum of particle 3
`PTRANS4CUT`	: $p_{transcut}^4$	cut on the transverse momentum of particle 4
`PTRANS5CUT`	: $p_{transcut}^5$	cut on the transverse momentum of particle 5
`RAPID`	: η	rapidity
`RAPID3CUT`	: η_{cut}^3	cut on the rapidity of particle 3
`RAPID4CUT`	: η_{cut}^4	cut on the rapidity of particle 4
`RAPID5CUT`	: η_{cut}^5	cut on the rapidity of particle 5
`DELTAR34CUT`	: $\Delta R^{34}{}_{cut}$	cut on the distance between particles 3 and 4
`DELTAR35CUT`	: $\Delta R^{35}{}_{cut}$	cut on the distance between particles 3 and 5
`DELTAR45CUT`	: $\Delta R^{45}{}_{cut}$	cut on the distance between particles 4 and 5
`RENSCALE`	: μ_R	renormalization scale
`FACTSCALE`	: μ_F	factorization scale
`Mh0`	: m_{h^0}	mass of the lighter CP-even Higgs boson
`MH0`	: m_{H^0}	mass of the heavier CP-even Higgs boson
`MHpm`	: $m_{H^\pm}$	mass of the charged Higgs boson
`MCha(1)`	: m_{χ_1}	mass of the lighter chargino
`MCha(2)`	: m_{χ_2}	mass of the heavier chargino
`MNeu(1)`	: $m_{\chi_1^0}$	mass of the lightest neutralino
`MNeu(2)`	: $m_{\chi_2^0}$	mass of the second-lightest neutralino
`MNeu(3)`	: $m_{\chi_3^0}$	mass of the second-heaviest neutralino
`MNeu(4)`	: $m_{\chi_4^0}$	mass of the heaviest neutralino
`MGl`	: $m_{\tilde{g}}$	mass of the gluino
`MSn(1)`	: $m_{\tilde{\nu}_e}$	mass of the electron sneutrino

[4] only relevant for the computation of hadronic cross sections

[5] only relevant for the computation of partonic cross sections and differential hadronic cross section with respect to invariant mass

[6] only for $2 \to 2$ processes

[7] only for $2 \to 3$ processes

Name	: Parameter description	
`MSn(2)`	: $m_{\tilde{\nu}_\mu}$	mass of the muon sneutrino
`MSn(3)`	: $m_{\tilde{\nu}_\tau}$	mass of the tau sneutrino
`MSl(1)`	: $m_{\tilde{e}_1}$	mass of the lighter selectron
`MSl(2)`	: $m_{\tilde{\mu}_1}$	mass of the lighter smuon
`MSl(3)`	: $m_{\tilde{\tau}_1}$	mass of the lighter stau
`MSL(1)`	: $m_{\tilde{e}_2}$	mass of the heavier selectron
`MSL(2)`	: $m_{\tilde{\mu}_2}$	mass of the heavier smuon
`MSL(3)`	: $m_{\tilde{\tau}_2}$	mass of the heavier stau
`MSu(1)`	: $m_{\tilde{u}_1}$	mass of the lighter sup
`MSu(2)`	: $m_{\tilde{c}_1}$	mass of the lighter scharm
`MSu(3)`	: $m_{\tilde{t}_1}$	mass of the lighter stop
`MSU(1)`	: $m_{\tilde{u}_2}$	mass of the heavier sup
`MSU(2)`	: $m_{\tilde{c}_2}$	mass of the heavier scharm
`MSU(3)`	: $m_{\tilde{t}_2}$	mass of the heavier stop
`MSd(1)`	: $m_{\tilde{d}_1}$	mass of the lighter sdown
`MSd(2)`	: $m_{\tilde{s}_1}$	mass of the lighter sstrange
`MSd(3)`	: $m_{\tilde{b}_1}$	mass of the lighter sbottom
`MSD(1)`	: $m_{\tilde{d}_2}$	mass of the heavier sdown
`MSD(2)`	: $m_{\tilde{s}_2}$	mass of the heavier sstrange
`MSD(3)`	: $m_{\tilde{b}_2}$	mass of the heavier sbottom
`TREE`	: σ^0	tree-level cross section
`LOOP`	: σ^1	one-loop cross section
`TREEERR`	: $\sigma(\sigma_0)$	integration error of the tree-level cross section
`LOOPERR`	: $\sigma(\sigma_1)$	integration error of the one-loop cross section
`TREEPROB`	: $\chi^2(\sigma(\sigma_0))$	probability of the integration error of the tree-level cross section
`LOOPPROB`	: $\chi^2(\sigma(\sigma_1))$	probability of the integration error of the one-loop cross section
`NREGIONS`[8]	:	number of regions used for integration
`NEVAL`[8]	:	number of function evaluations used for integration
`FAIL`[8]	:	a non-zero value indicates that the desired accuracy could not be reached

[8] only relevant for some integration routines

Appendix A

Phase-space parametrization

In this appendix the parametrization of the phase space for $2 \to 2$ and $2 \to 3$ processes, as it was used for the calculations of this thesis, is presented. It is the same parametrization which is also used in FormCalc [9, 10, 16]. The parametrization is performed in the center-of-mass system of the two incoming particles, which define the beam axis and carry a center-of-mass energy of $\sqrt{s}$. For each final-state particle an integral over its three-momentum $\vec{k}$ occurs in the calculation of integrated cross sections. The energy k^0 of the particle is fixed by the on-shell condition $k^0 = \sqrt{|\vec{k}|^2 + m^2}$, where m denotes the mass of the particle. Four of these integrals are eliminated by global energy-momentum conservation. In the following sections the parametrizations of the two- and three-particle phase space are shown.

A.1 Two-particle phase space

With two particles in the final state, labeled by the subscripts 3 and 4 in the following, the phase-space integral can be written in terms of two angles. They are the azimuth angle ϕ and the polar angle θ with respect to the beam axis. Because of rotational invariance around the beam axis the integration over ϕ is trivial and amounts to a factor of 2π. So the integral over the two-particle phase space is given by

$$\int \mathrm{d}\Gamma_2 = \int_{-1}^{1} \mathrm{d}c_\theta \, \frac{1}{8\pi} \frac{|\vec{k}_3|}{\sqrt{s}} \quad , \tag{A.1}$$

where

$$|\vec{k}_3|^2 = |\vec{k}_4|^2 = \frac{s^2 + m_3^4 + m_4^2 - 2m_3^2 s - 2m_4^2 s - 2m_3^2 m_4^2}{4s} \tag{A.2}$$

denotes the squared absolute value of the three-momentum of the final-state particles, m_3 and m_4 are their respective masses, and $\sqrt{s}$ specifies the center-of-mass energy of the incoming particles.

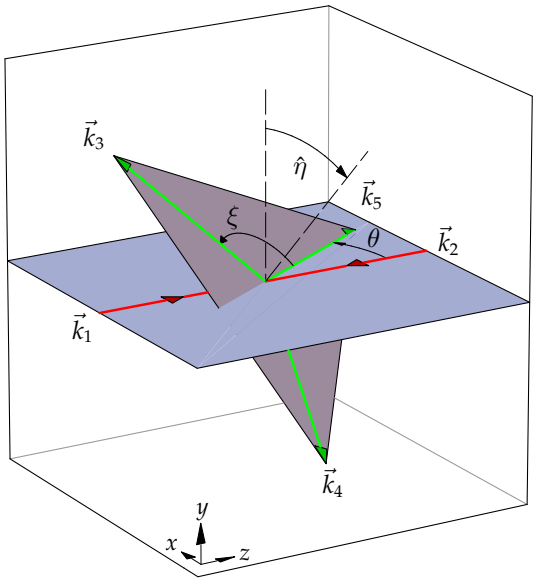

Figure A.1: Graphical representation of the variables used in the parametrization of the $2 \to 3$ phase space. The figure is taken from ref. [16].

A.2 Three-particle phase space

For the three-particle phase space, where the outgoing particles are labeled by the indices 3, 4 and 5, five independent integration variables remain after global energy-momentum conservation has been applied. They are the energies k_3^0 and k_5^0, the azimuth angle ϕ and the polar angle θ of the fifth particle with respect to the beam axis, and the angle $\hat{\eta}$ which rotates particle 3 out of the plane defined by particle 5 and the beam axis. A graphical representation of the angles is given in Fig. A.1.

The four-momenta of the outgoing particles have the following explicit form

$$\begin{aligned} k_3 &= (k_3^0, |\vec{k}_3|\vec{e}_3) & k_4 &= (\sqrt{s} - k_3^0 - k_5^0, -\vec{k}_3 - \vec{k}_5) \\ k_5 &= (k_5^0, |\vec{k}_5|\vec{e}_5) \quad , \end{aligned} \tag{A.3}$$

with

$$\vec{e}_3 = \begin{pmatrix} c_\theta c_{\hat{\eta}} s_\xi + s_\theta c_\xi \\ s_{\hat{\eta}} s_\xi \\ c_\theta c_\xi - s_\theta c_{\hat{\eta}} s_\xi \end{pmatrix} \qquad \vec{e}_5 = \begin{pmatrix} s_\theta \\ 0 \\ c_\theta \end{pmatrix} \quad . \tag{A.4}$$

The angle θ, which is also plotted in the figure, is defined over

$$c_\theta = \frac{(\sqrt{s} - k_3^0 - k_5^0)^2 - m_4^2 - |\vec{k}_3|^2 - |\vec{k}_5|^2}{2|\vec{k}_3||\vec{k}_5|} \quad . \tag{A.5}$$

m_i again denotes the mass of the respective particle i and $\sqrt{s}$ is the center-of-mass energy of the initial-state particles. Due to axial symmetry the trivial integration over ϕ can be carried out immediately and yields a factor of 2π.

Then the parametrization of the three-particle phase space takes the following form

$$\int \mathrm{d}\Gamma_3 = \int_{m_5}^{(k_5^0)^{\max}} \mathrm{d}k_5^0 \int_{(k_3^0)^{\min}}^{(k_3^0)^{\max}} \mathrm{d}k_3^0 \int_{-1}^{1} \mathrm{d}c_\theta \int_0^{2pi} \mathrm{d}\hat{\eta}\, \frac{1}{64\pi^3} \quad , \tag{A.6}$$

where the integration limits are given by

$$(k_5^0)^{\max} = \frac{\sqrt{s}}{2} - \frac{(m_3+m_4)^2 - m_5^2}{2\sqrt{s}} \tag{A.7}$$

and

$$(k_3^0)^{\max,\min} = \frac{1}{2\tau}\left[\sigma(\tau + m_+m_-) \pm |\vec{k}_5|\sqrt{(\tau - m_+^2)(\tau - m_-^2)}\right] \quad , \tag{A.8}$$

using

$$\sigma = \sqrt{s} - k_5^0 \qquad \tau = \sigma^2 - |\vec{k}_5|^2 \qquad m_\pm = m_3 \pm m_4 \quad . \tag{A.9}$$

Bibliography

[1] D. J. Gross and F. Wilczek. *Phys. Rev. Lett.*, 30:1343–1346, 1973; H. D. Politzer. *Phys. Rev. Lett.*, 30:1346–1349, 1973.

[2] R. Brock et al. *Rev. Mod. Phys.*, 67:157–248, 1995; R. Brock et al. *Handbook of perturbative QCD; Version 1.1: September 1994*, FERMILAB-PUB-94-316, 1994.

[3] J. D. Bjorken and E. A. Paschos. *Phys. Rev.*, 185:1975–1982, 1969; R. P. Feynman. *Phys. Rev. Lett.*, 23:1415–1417, 1969.

[4] E. D. Bloom et al. *Phys. Rev. Lett.*, 23:930–934, 1969; M. Breidenbach et al. *Phys. Rev. Lett.*, 23:935–939, 1969; J. I. Friedman and H. W. Kendall. *Ann. Rev. Nucl. Part. Sci.*, 22:203–254, 1972.

[5] J. C. Collins, D. E. Soper, and G. Sterman. *Adv. Ser. Direct. High Energy Phys.*, 5:1–91, 1988.

[6] G. Altarelli and G. Parisi. *Nucl. Phys.*, B126:298, 1977.

[7] C. Meier. *γZ-Produktion in P-P-Kollisionen mit elektroschwachen 1-Schleifen-Korrekturen*. Diplom thesis, Universität Karlsruhe, July 2001.

[8] J. Kublbeck, M. Bohm, and A. Denner. *Comput. Phys. Commun.*, 60:165–180, 1990; T. Hahn. *Comput. Phys. Commun.*, 140:418–431, 2001; T. Hahn and C. Schappacher. *Comput. Phys. Commun.*, 143:54–68, 2002.

[9] T. Hahn and M. Perez-Victoria. *Comput. Phys. Commun.*, 118:153–165, 1999.

[10] T. Hahn. *New developments in FormCalc 4.1*, hep-ph/0506201, 2005.

[11] H. Plothow-Besch. *PDFLIB - User's Manual Version 8.04*. CERN-ETT/TT 2000.04.17.

[12] W. Giele and M. Whalley. *LHAPDF version 4 Users Guide*, April 2002. `http://durpdg.dur.ac.uk/lhapdf/manual.htm`.

[13] S. Heinemeyer, W. Hollik, and G. Weiglein. *Comput. Phys. Commun.*, 124:76–89, 2000.

[14] S. Eidelman et al. *Phys. Lett.*, B592:1, 2004.

[15] T. Hahn. *Comput. Phys. Commun.*, 168:78–95, 2005.

[16] T. Hahn. *FormCalc 4 User's Guide*, March 2005. `http://www.feynarts.de/formcalc/FC4Guide.ps.gz`.