How To Cook Up An Event Generator

acfa-sim-j Group

Jul. 23, 2007

Cross Section Formula

$$e^+e^- \to X_1 + \cdots + X_f + \cdots + X_n$$

$(\boldsymbol{p}^+, s^+)$ $(\boldsymbol{p}^-, s^-)$ $(\boldsymbol{p}_f, s_f)$

$$d\sigma = \frac{1}{2s\beta_e} \sum_{s^+, s^-, s_f} w_{s^+} w_{s^-} \left|\mathcal{T}_{fi}\right|^2 d\Phi_n$$

spin weight for e^-

spin weight for e^+

$$w_{s=\pm} = \frac{1 \pm P_s}{2} \qquad \left(-1 \le P_s = \frac{N_+ - N_-}{N_+ + N_-} \le +1\right)$$

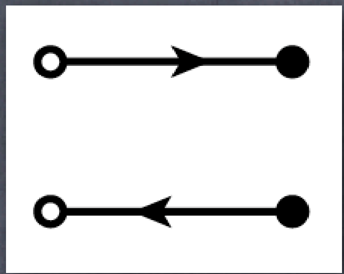
$$\mathcal{T}_{fi} = \left\langle \boldsymbol{p}_f, s_f \right| \hat{T} \left| \boldsymbol{p}^+, s^+; \boldsymbol{p}^-, s^- \right\rangle$$

What We Need

- Helicity amplitudes : $\mathcal{T}_{fi}$ from HELAS
- Lorentz invariant phase space : $d\Phi_n$
  - Problem dependent in general
  - General method exists if there is no singular kinematical variable
- Numerical integration : BASES
- Event Generation : SPRING

Helicity Amplitudes: HELAS

External Lines



Ψ_i $IXXXXX(p, m, \lambda, \pm 1, \Psi_i)$

4-momentum, helicity, particle, spinor, mass, anti-particle

Ψ_o $OXXXXX(p, m, \lambda, \pm 1, \Psi_o)$

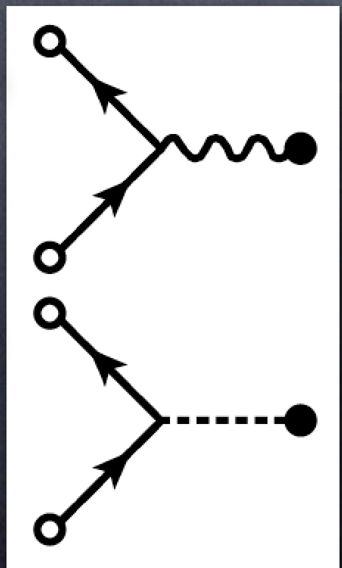
Currents

Ψ_o

Ψ_i

Ψ_o

Ψ_i



incoming spinor, outgoing spinor, width

V $JIOXXX(\Psi_i, \Psi_o, G_V, m_V, \Gamma_V, V)$

mass, wave fun.

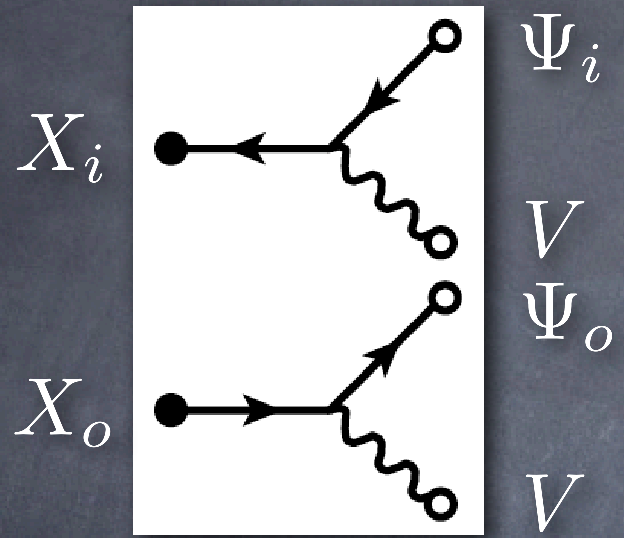
$G_V(1)$: left

(2) : right

S $HIOXXX(\Psi_i, \Psi_o, G_S, m_S, \Gamma_S, S)$

wave fun.

Virtual Fermions



incoming spinor, vector, mass, width

$$FVIXXX(\Psi_i, V, G_V, m_X, \Gamma_X, X_i)$$

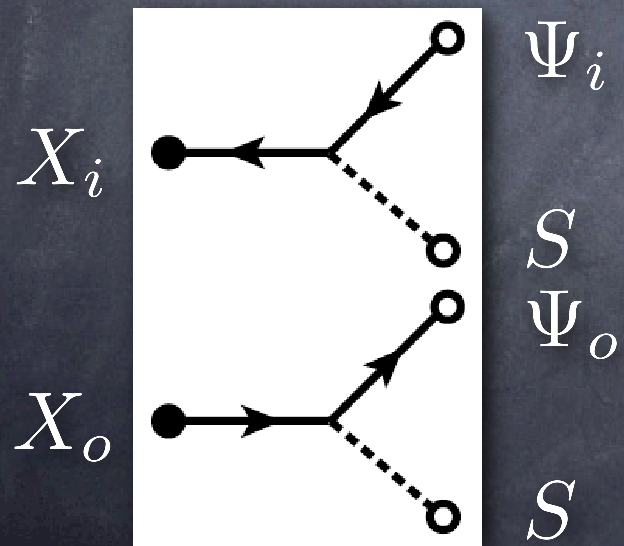
$G_V(1)$: left
(2) : right

incoming virtual spinor

$$FVOXXX(\Psi_o, V, G_V, m_X, \Gamma_X, X_o)$$

outgoing spinor

outgoing virtual spinor

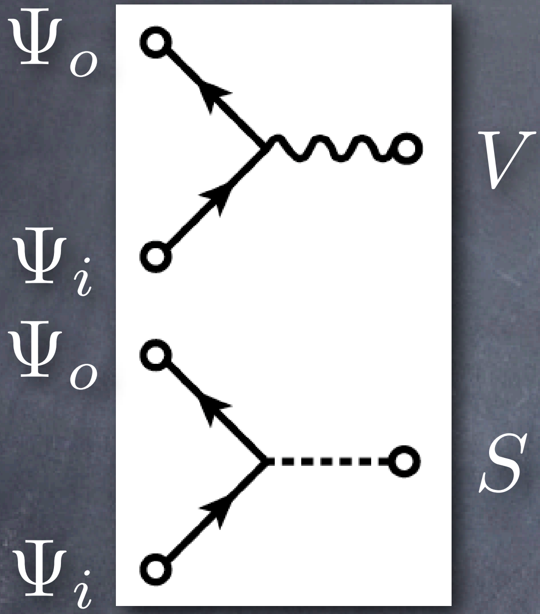


$$FSIXXX(\Psi_i, S, G_S, m_X, \Gamma_X, X_i)$$

scalar

$$FSOXXX(\Psi_o, S, G_S, m_X, \Gamma_X, X_o)$$

Vertices



incoming spinor, outgoing spinor, vector

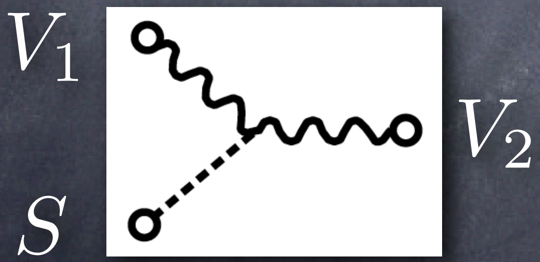
$$IOVXXX(\Psi_i, \Psi_o, V, G_V, A)$$

amplitude

$G_V(1)$: left
(2) : right

$$IOSXXX(\Psi_i, \Psi_o, S, G_S, A)$$

scalar

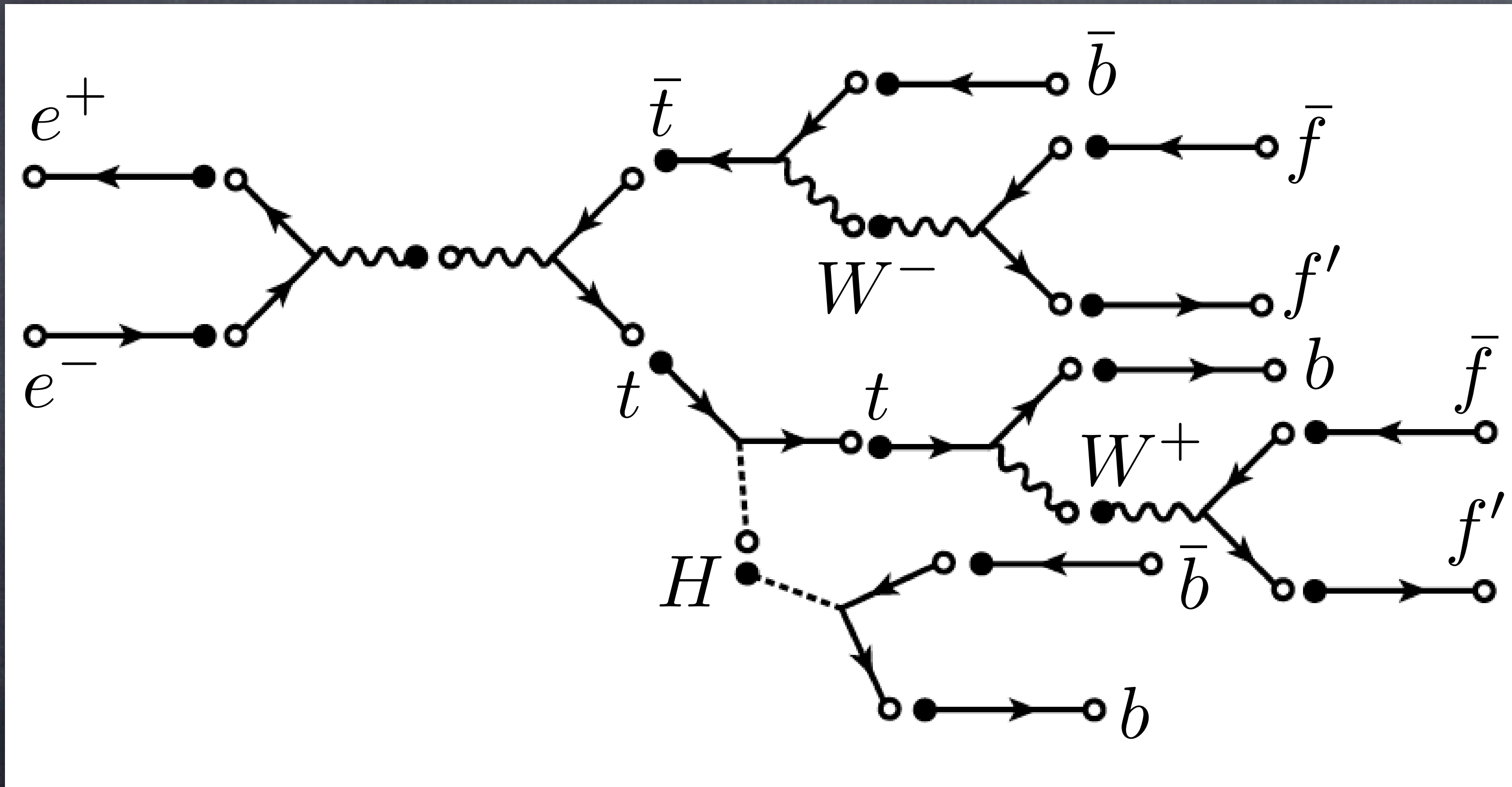


$$VVSXXX(V_1, V_2, S, G_{VVS}, A)$$

Note: there are some more subroutines in HELAS (see manual)

Composition of Full Amplitude

$$e^+e^- \to t\bar{t}H$$



Note: there are some other diagrams

See physsim/top/TTHStudy

```fortran
C*(Update Record)
C*  95/03/16  K.Fujii          New version with HELAS V204.
C*

      SUBROUTINE AMPTT (GAL,GAU,GZL,GZU,AMZ,GMZ,
     .                  EIN,EOT,TIN,TOT,AMP)

      IMPLICIT   REAL*4  ( A-H, O-Z )
      REAL    *4  GAL(2), GAU(2), GZL(2), GZU(2), AMZ, GMZ
      COMPLEX*8  EIN(6), EOT(6), TIN(6), TOT(6), AMP(0:2)
C--
      COMPLEX*8  CURR(6)
C
C=======< Entry Point >==========================================
C
C--
C  Compute photon s-channel exchange diagram.
C--
      CALL JIOXXX(EIN,EOT,GAL,0.,0.,  CURR)
      CALL IOVXXX(TIN,TOT,CURR,GAU,AMP(1))
C--
C  Compute Z s-channel exchange diagram.
C--
      CALL JIOXXX(EIN,EOT,GZL,AMZ,GMZ,  CURR)
      CALL IOVXXX(TIN,TOT,CURR,GZU,AMP(2))
C--
C  Sum two amplitudes.
C     AMP(0) = sum.
C        (1) = photon-exchange.
C        (2) = Z-exchange.
C--
      AMP(0) = AMP(1) + AMP(2)
C--
C  That's it.
C--
      RETURN
      END
```

```fortran
      SUBROUTINE AMPZH (GZL,GVH,AMZ,GMZ,ZVCT,HOT,EIN,EOT,AMP)

      IMPLICIT   REAL*4   ( A-H, O-Z )
      REAL    *4  GZL(2), GVH, AMZ, GMZ
      COMPLEX*8  EIN(6), EOT(6), ZVCT(6), HOT(3), AMP
C--
      COMPLEX*8  CURR(6)
C
C=======( Entry Point )==============================================
C
C--
C   Compute photon s-channel exchange diagram.
C--
      CALL JIOXXX(EIN,EOT,GZL,AMZ,GMZ,CURR)
      CALL VVSXXX(CURR,ZVCT,HOT,GVH,AMP)
C--
C   That's it.
C--
      RETURN
      END
```

```fortran
      SUBROUTINE AMPTTH(GAL,GAF,GZL,GZF,GVH,GCH, AMZ,GMZ,AMF,GMF,
     .                  EIN,EOT, FIN,FOT,HOT, AMP)

      IMPLICIT   REAL*4  ( A-H, O-Z )
      REAL     *4  GAL(2), GAF(2), GZL(2), GZF(2), GVH,
     .           AMZ, GMZ, AMF, GMF
      COMPLEX*8  GCH(2),
     .           EIN(6), EOT(6), FIN(6), FOT(6), HOT(3),
     .           AMP(0:2)
C--
      COMPLEX*8  SPWRK(6), SCWRK(3), VCWRK(6), TMP(0:2)
C
C========< Entry Point >==============================================
C
C--
C  Bremsstrahlung off f-bar.
C--
      CALL FSIXXX(FIN,HOT,GCH,AMF,GMF,SPWRK)
      CALL AMPTT (GAL,GAF,GZL,GZF,AMZ,GMZ,
     .            EIN,EOT,SPWRK,FOT,TMP)
      AMP(1) = TMP(0)
C--
C  Bremsstrahlung off f.
C--
      CALL FSOXXX(FOT,HOT,GCH,AMF,GMF,SPWRK)
      CALL AMPTT (GAL,GAF,GZL,GZF,AMZ,GMZ,
     .            EIN,EOT,FIN,SPWRK,TMP)
      AMP(1) = AMP(1) + TMP(0)
C--
C  ZH and Z to ff-bar.
C--
      CALL JIOXXX(FIN,FOT,GZF,AMZ,GMZ,VCWRK)
      CALL AMPZH (GZL,GVH,AMZ,GMZ,VCWRK,HOT,EIN,EOT,AMP(2))
C--
C  Sum up amplitudes.
C--
      AMP(0) = AMP(1) + AMP(2)
C--
C  That's it.
C--
      RETURN
      END
```

Phase Space

n-body Phase Space

$$d\Phi_n = (2\pi)^4 \delta^4 \left(P_i - \sum_{f=1}^{n} p_f \right) \prod_{f=1}^{n_f} \left(\frac{d^3\mathbf{p}_f}{(2\pi)^3 2E_f} \right)$$

Recurrence Formula

$$d\Phi_n = d\Phi_{1+(n_f-n_r)} \left(P_i \to q + \sum_{f=n_r+1}^{n_f} p_f \right) \frac{dq^2}{2\pi} d\Phi_{n_r} (q \to p_1 + \cdots p_{n_r})$$

By repeatedly using this, we can reduce the n-body phase space to a product of 2-body phase space factors

Phase Space

2-body Phase Space

$$M \rightarrow m_1 + m_2$$

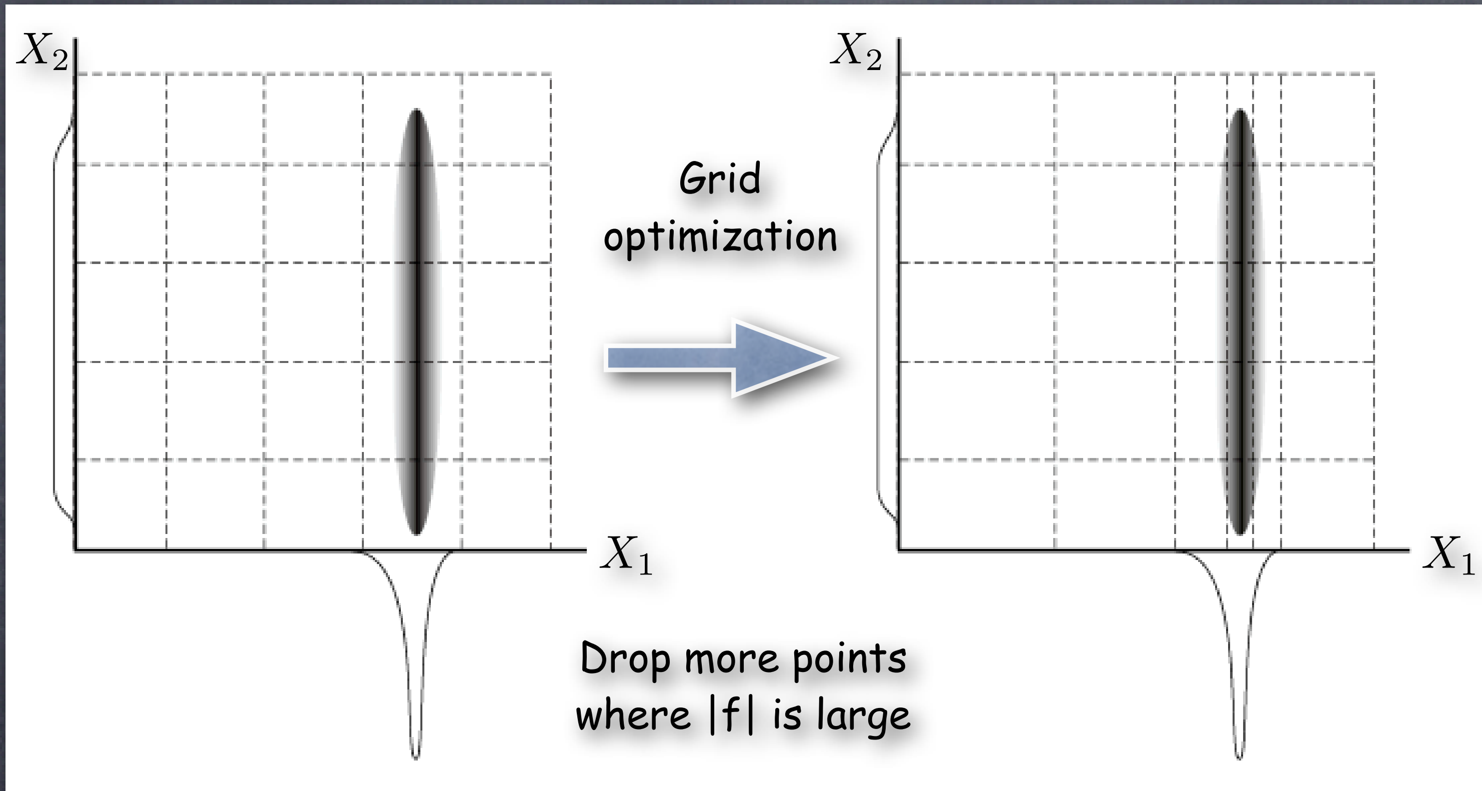
$$d\Phi_2 = \frac{\bar{\beta}((m_1/M)^2, (m_2/M)^2)}{8\pi} \frac{d\Omega}{4\pi}$$

where

$$\bar{\beta}(X_1, X_2) = \sqrt{1 - 2(X_1 + X_2) + (X_1 - X_2)^2}$$

$$\begin{aligned} \bar{\beta}(X_1, X_2) &= 1 - X_1 & (X_2 = 0) \\ &= \sqrt{1 - 4X} & (X_1 = X_2 = X) \end{aligned}$$

Numerical Integration (BASES)



Weighted sampling

$$\int f(\boldsymbol{X})\, d\boldsymbol{X} = \frac{1}{n} \sum f(\boldsymbol{X})\, \Pi_i \Delta X_i$$

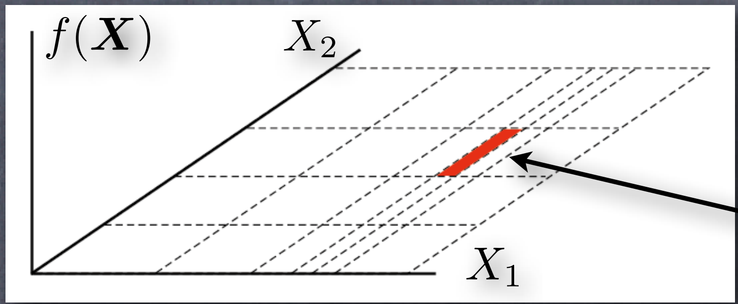
sample points / cell

Caution

- The grid optimization is carried out based on the projection to each integration variable
  - If your singularity runs diagonally in your phase space, you are in trouble!
    - Need to make an appropriate choice of kinematical variables
- Watch your convergence table given by BASES. If you find your temporary estimates skip more than a few percents, your integration result is most probably incorrect!

Event Generation (SPRING)

- At the end of the integration step, you know the relative importance of each cell and the maximum value of your integrand in that cell
  - Throw a dice to decide which cell you pick up



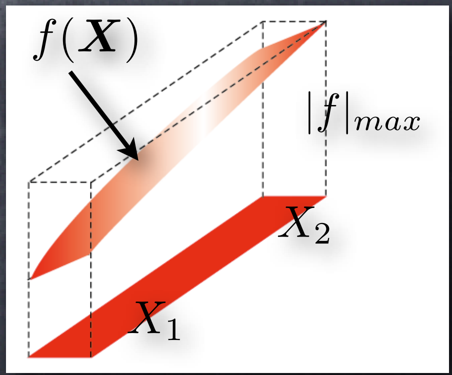
Pick up this cell

  - Decide the phase space point within that cell according to the usual hit-and-miss method

uniform random number in [0,1]

If $|f|_{max} \cdot Z \leq f(\boldsymbol{X})$

--> Accept this point



Sample Programs

Physsim

http://www-jlc.kek.jp/subg/offl/physsim/

Analysis examples in

physsim/Anlib/examples/jsf/