The MadGraph5 Framework

Olivier Mattelaer
University of Illinois at Urbana-Champaign

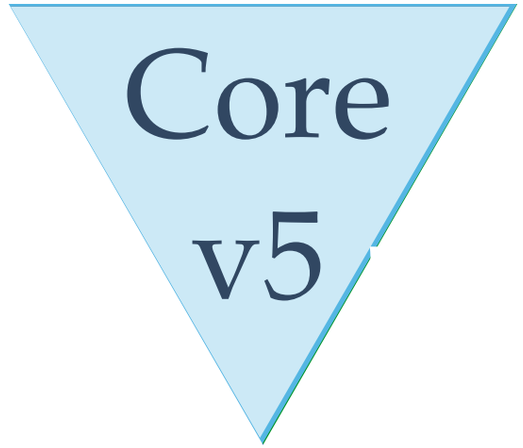
Plan

- MadGraph5
- Status
- Plan

Recap: "What is MG5"

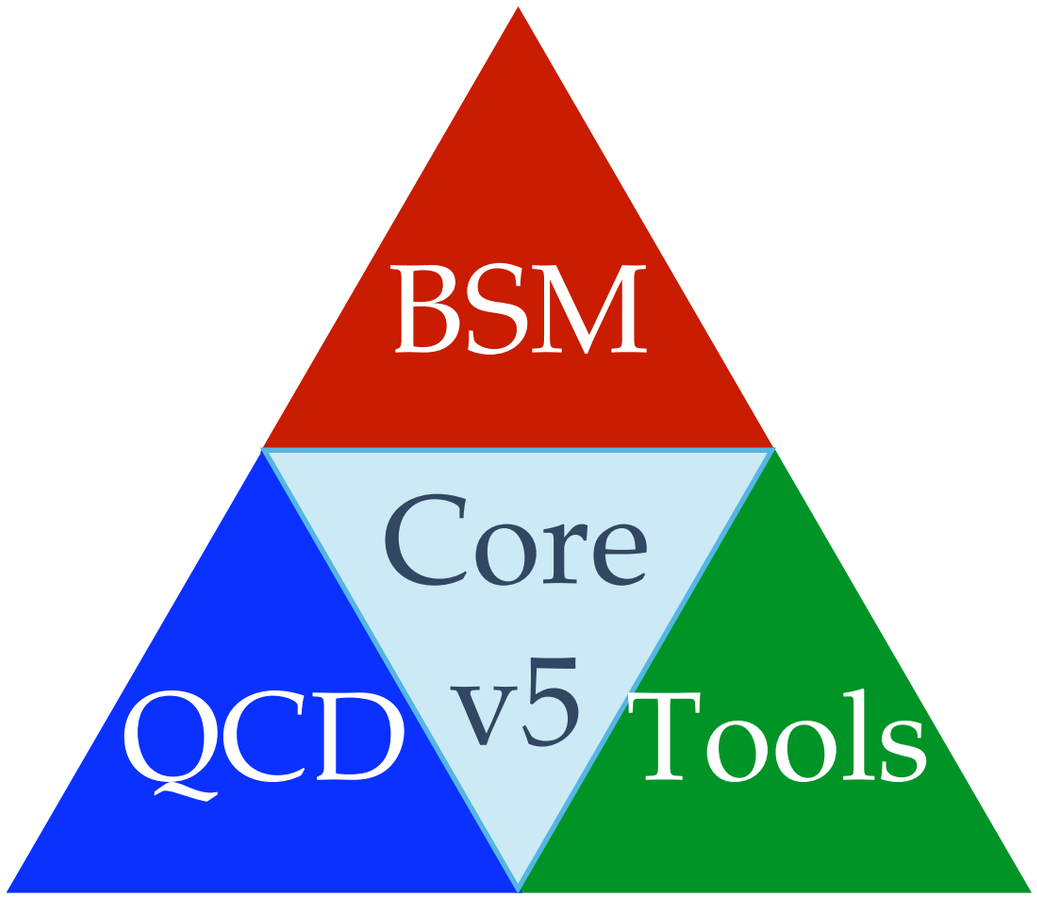
- MadGraph 5 is a completely new (released spring 2011) matrix element generator written in Python
- Can handle ANY model (that can be written as a Lagrangian), conveniently output by FeynRules
  - Any Lorentz structure for any spin (up to 2) and color (6tets, ε^{ijk})
  - Multiparticle vertices for any multiplicity, multifermion vertices
- Super fast process generation
- Unlimited-length decay chains with full BW and spin effects
- Event generation speedups by orders of magnitude
- Output in multiple languages and formats (including Pythia 8)
- Super-user-friendly command line interfaces

MadGraph 5



MadGraph 5

UFO / ALOHA



MADLOOP
MADFKS
aMC@NLO

MadSpin
MADWEIGHT
MadAnalysis5

MG Core	Functionalities	Tools
generate	Output	launch

MadGraph StandAlone

UFO / ALOHA

MadLoop StandAlone

Leading Order

- MadEvent
- MadWeight*
- To Pythia 8
- DarkMatter*

NLO

- MadKFS
- aMC@NLO

- MadAnalysis5
- MC@NLO
- MadSpin
- pythia-pgs
- Delphes

They are space for more !!

A Set of Tools

A Set of Tools

A Set of Tools

The perfect Tool box

- Need Modular Code (OO / Python)
  - Easy to add new features
- Use Launchpad to save progress
- Need Extensive test suite
  - If the tests passes the code should be stable
  - The FIRST debugging step is to add a new tests!
- Think as a USER

Command Interface

- Nice Interactive session
  - Auto-completion
  - Tutorial
  - interactive help
- Simple command set
  - import model sm
  - generate p p > e+ e-
  - output FORMAT MY_DIR
  - launch (-i)

```
************************************************************
*                                                          *
*           W E L C O M E  to  M A D G R A P H  5          *
*                                                          *
*                                                          *
*                   *                   *                  *
*                     *       * *       *                  *
*                       * * * * 5 * * * *                  *
*                     *       * *       *                  *
*                   *                   *                  *
*                                                          *
*        VERSION 1.3.16                2011-09-11          *
*                                                          *
*     The MadGraph Development Team - Please visit us at   *
*     https://server06.fynu.ucl.ac.be/projects/madgraph    *
*                                                          *
*                 Type 'help' for in-line help.            *
*           Type 'tutorial' to learn how MG5 works         *
*                                                          *
************************************************************
load MG5 configuration from /Users/omatt/.mg5_config
Loading default model: sm
models.import_ufo: Restrict model sm with file models/sm/rest
models.import_ufo: Run "set stdout_level DEBUG" before import
INFO: Change particles name to pass to MG5 convention
Defined multiparticle p = g u c d s u~ c~ d~ s~
Defined multiparticle j = g u c d s u~ c~ d~ s~
Defined multiparticle l+ = e+ mu+
Defined multiparticle l- = e- mu-
Defined multiparticle vl = ve vm vt
Defined multiparticle vl~ = ve~ vm~ vt~
mg5>help
```

Core news since MG5 release - Quick list

- Lots of speedups and improvements, including
  - Vast speedup for long decay chains with multiparticle decays
  - Huge improvements in user interface
  - Multi cluster support
- MadSpin (TODAY!!) -- See next talk
- New 4-flavor matching and VBF-type matching
- 4 fermion vertices in FR+MG5 (except Majorana)
- Spin 3/2 particles in FR+MG5
- Complex mass scheme
- Feynman gauge
- Handling of negative weights
- On-the-fly 2-body decay width calculations ("Auto width")

“LO” upcoming developments - Quick list

- Automatic matching to Pythia 8 (S. Pretzel et al)
  - CKKW-L matching
- Automatic scale/PDF/matching systematics (A. Kalogeropoulos et al)
- Fast multiparton processes using color-ordered recursion (J. Alwall et al)
- MadDecay: Automatic BSM decay width suite (including needed 3- and 4-body widths and decay of event file) (with C-H Shen et al)
- MadDM: Relic density calculations and direct detection limits for any BSM model (K.C. Kong et al)
- MadWeight5: Matrix-Element Method (P. Artoisenet, OM)

“NLO” upcoming developments - Quick list

- More Flexibility at Leading Order
- Interface to Pythia 8
- Automation of FxFx merging
- EW corrections (matched to the shower)
- Loop-induced
- UFO@NLO
- MadAnalysis5
- Loop BSM Library

See Other Talks !!!

Idea / Request

- BSM / SM Extension
  - Arbitrary gauge
  - Form Factor
    - arbitrary propagator
    - Majorana fermion in 4 Fermion interactions
- LHE Events Reweighting
- UserMod

Arbitrary Gauge

$$\eta_{\mu\nu} + \xi \frac{p_\mu p_\nu}{m^2}$$

- Current Status: Feynman/Unitary
- Do we need more?
  - Useful for test
- Need modification in the UFO
  - include the gauge parameter
- Need to modify the way to import model in MG5

Arbitrary propagator

- Interesting for Spin 2 and Spin 3/2
- Interesting for Gluon / Photon in unitary gauge
  - Allow to have a specialized propagator
- Interesting for some specific model
- Needed for open-loop
- Need an extension of UFO And ALOHA
- Need to modify the helas_call_writer

Form Factor

- Extend the valid input to ALOHA:
  - Example : HZA form factor interaction

```
HZA = FormFactor(name = 'HZA',
                 type = 'complex',
                 value = '2./3.*(3.-8.*SW2)/4./SW/CW*(1./2./((MH**2)/4./MT/MT-(P(-1,1)*P(-1,1))/4./MT/MT)+(1./2.*(1.-(MH**2)/4./MT/MT+(P(-1,1)*P(-1,1))/4./MT/MT)*(theta_function((P(-2,1)*P(-2,1))/4./MT/MT-1.,-1./4.*(cmath.log((1.+cmath.sqrt(1.-4.*MT*MT/(P(-2,1)*P(-2,1))))/(1.-cmath.sqrt(1.-4.*MT*MT/(P(-2,1)*P(-2,1)))))-complex(0,1)*cmath.pi)**2.,cmath.asin(cmath.sqrt((P(-2,1)*P(-2,1))/4./MT/MT))**2.)-theta_function((MH**2)/4./MT/MT-1.,-1./4.*(cmath.log((1.+cmath.sqrt(1.-4.*MT*MT/(MH**2)))/(1.-cmath.sqrt(1.-4.*MT*MT/(MH**2))))-complex(0,1)*cmath.pi)**2.,cmath.asin(cmath.sqrt((MH**2)/4./MT/MT))**2.))+(P(-1,1)*P(-1,1))/4./MT/MT*(theta_function((P(-2,1)*P(-2,1))/4./MT/MT-1.,1./2.*cmath.sqrt(1.-4.*MT*MT/(P(-2,1)*P(-2,1)))*(cmath.log((1.+cmath.sqrt(1.-4.*MT*MT/(P(-2,1)*P(-2,1))))/(1.-cmath.sqrt(1.-4.*MT*MT/(P(-2,1)*P(-2,1)))))-complex(0,1)*cmath.pi),cmath.sqrt(4.*MT*MT/(P(-2,1)*P(-2,1))-1.)*cmath.asin(cmath.sqrt((P(-1,1)*P(-1,1))/4./MT/MT)))-theta_function((MH**2)/4./MT/MT-1.,1./2.*cmath.sqrt(1.-4.*MT*MT/(MH**2))*(cmath.log((1.+cmath.sqrt(1.-4.*MT*MT/(MH**2)))/(1.-cmath.sqrt(1.-4.*MT*MT/(MH**2))))-complex(0,1)*cmath.pi),cmath.sqrt(4.*MT*MT/(MH**2)-1.)*cmath.asin(cmath.sqrt((MH**2)/4./MT/MT)))))/((MH**2)/4./MT/MT-(P(-1,1)*P(-1,1))/4./MT/MT)**2)-1./3.*(4.*SW2-3.)/4./SW/CW*(1./2./((MH**2)/4./MB/MB-(P(-1,1)*P(-1,1))/4./MB/MB)+(1./2.*(1.-(MH**2)/4./MB/MB+(P(-1,1)*P(-1,1))/4./MB/MB)*(theta_function((P(-2,1)*P(-2,1))/4./MB/MB-1.,-0.25*(cmath.log((1.+cmath.sqrt(1.-4.*MB*MB/(P(-2,1)*P(-2,1))))/(1.-cmath.sqrt(1.-4.*MB*MB/(P(-2,1)*P(-2,1)))))-complex(0,1)*cmath.pi)**2.,cmath.asin(cmath.sqrt((P(-2,1)*P(-2,1))/4./MB/MB))**2.)-theta_function((MH**2)/4./MB/MB-1.,-0.25*(cmath.log((1.+cmath.sqrt(1.-4.*MB*MB/(MH**2)))/(1.-cmath.sqrt(1.-4.*MB*MB/(MH**2))))-complex(0,1)*cmath.pi)**2.,cmath.asin(cmath.sqrt((MH**2)/4./MB/MB))**2.))+(P(-1,1)*P(-1,1))/4./MB/MB*(theta_function((P(-2,1)*P(-2,1))/4./MB/MB-1.,1./2.*cmath.sqrt(1.-4.*MB*MB/(P(-2,1)*P(-2,1)))*(cmath.log((1.+cmath.sqrt(1.-4.*MB*MB/(P(-2,1)*P(-2,1))))/(1.-cmath.sqrt(1.-4.*MB*MB/(P(-2,1)*P(-2,1)))))-complex(0,1)*cmath.pi),cmath.sqrt(4.*MB*MB/(P(-2,1)*P(-2,1))-1.)*cmath.asin(cmath.sqrt((P(-2,1)*P(-2,1))/4./MB/MB)))-theta_function((MH**2)/4./MB/MB-1.,0.50*cmath.sqrt(1.-4.*MB*MB/(MH**2))*(cmath.log((1.+cmath.sqrt(1.-4.*MB*MB/(MH**2)))/(1.-cmath.sqrt(1.-4.*MB*MB/(MH**2))))-complex(0,1)*cmath.pi),cmath.sqrt(4.*MB*MB/(MH**2)-1.)*cmath.asin(cmath.sqrt((MH**2)/4./MB/MB)))))/((P(-5,3)**2)/4./MB/MB-(P(-1,1)*P(-1,1))/4./MB/MB)**2.)-CW/SW/8./((MH**2)/4./MW/MW - (P(-2,1)*P(-2,1))/4./MW/MW )*((5.-SW2/CW**2+2.*(MH**2)/4./MW/MW*(1.-SW2/CW**2))*(1.+2.*(P(-2,1)*P(-2,1))/4./MW/MW/((MH**2)/4./MW/MW-(P(-2,1)*P(-2,1))/4./MW/MW)*(theta_function((P(-2,1)*P(-2,1))/4./MW/MW-1.,0.50*cmath.sqrt(1.-4.*MW*MW/(P(-2,1)*P(-2,1)))*(cmath.log((1.+cmath.sqrt(1.-4.*MW*MW/(P(-2,1)*P(-2,1))))/(1.-cmath.sqrt(1.-4.*MW*MW/(P(-2,1)*P(-2,1)))))-complex(0,1)*cmath.pi),cmath.sqrt(4.*MW*MW/(P(-2,1)*P(-2,1))-1.)*cmath.asin(cmath.sqrt((P(-2,1)*P(-2,1))/4./MW/MW)))-theta_function((MH**2)/4./MW/MW-1.,0.50*cmath.sqrt(1.-4.*MW*MW/(MH**2))*(cmath.log((1.+cmath.sqrt(1.-4.*MW*MW/(MH**2)))/(1.-cmath.sqrt(1.-4.*MW*MW/(MH**2))))-complex(0,1)*cmath.pi),cmath.sqrt(4.*MW*MW/(MH**2)-1.)*cmath.asin(cmath.sqrt((MH**2)/4./MW/MW)))))+1./((MH**2)/4./MW/MW-(P(-2,1)*P(-2,1))/4./MW/MW)*(4.*(P(-2,1)*P(-2,1))/4./MW/MW*(3.-SW2/CW**2)-(2.*(MH**2)/4./MW/MW-1.)*(5.-SW2/CW**2))*(theta_function((P(-2,1)*P(-2,1))/4./MW/MW-1.,-0.25*(cmath.log((1.+cmath.sqrt(1.-4.*MW*MW/(P(-2,1)*P(-2,1))))/(1.-cmath.sqrt(1.-4.*MW*MW/(P(-1,1)*P(-1,1)))))-complex(0,1)*cmath.pi)**2.,cmath.asin(cmath.sqrt((P(-1,1)*P(-1,1))/4./MW/MW))**2.)-theta_function((MH**2)/4./MW/MW-1.,-0.25*(cmath.log((1.+cmath.sqrt(1.-4.*MW*MW/(MH**2)))/(1.-cmath.sqrt(1.-4.*MW*MW/(MH**2))))-complex(0,1)*cmath.pi)**2.,cmath.asin(cmath.sqrt((MH**2)/4./MW/MW))**2.))')
```

- Validated on this example (and on h g g)
- fast (0.36 s to 0.75 s)

ALOHA

Google translate

```
      TMP1 = (P1(0)*V2(3)-P1(1)*V2(4)-P1(2)*V2(5)-
P1(3)*V2(6))
      TMP0 = (P1(0)*P1(0)-P1(1)*P1(1)-P1(2)*P1(2)-
P1(3)*P1(3))
      TMP3 = (V2(3)*V1(3)-V2(4)*V1(4)-V2(5)*V1(5)-
V2(6)*V1(6))
      TMP2 = (V1(3)*P2(0)-V1(4)*P2(1)-V1(5)*P2(2)-
V1(6)*P2(3))
      FCT0 = 1D0/(SW)
      FCT1 = 1D0/(CW)
      FCT2 = (MH)**(2D0)
      FCT3 = 1D0/(MT)
      FCT4 = 1D0/(1D0/4D0 * FCT3*FCT3*(FCT2-TMP0))
      FCT5 = 1D0/(TMP0)
      FCT6 = SQRT(DBLE(-1D0*(+4D0*(MT*MT*FCT5)+
-1D0)))
      FCT7 = 1D0/((-FCT6+ 1D0))
      FCT8 = LOG(DBLE(FCT7*(FCT6+ 1D0)))
      FCT9 = (-1D0*(+CI*(PI)-FCT8))**(2D0)
      FCT10 = SQRT(DBLE(1D0/4D0 *
TMP0*FCT3*FCT3))
      FCT11 = ASIN(DBLE(FCT10))
      FCT12 = (FCT11)**(2D0)
      FCT13 =
THETA_FUNCTION((+1D0/4D0*(FCT3*FCT3*TMP0)+
-1D0),
     $ -1D0/4D0 * FCT9,FCT12)
      FCT14 = 1D0/(FCT2)
      FCT15 =
SQRT(DBLE(-1D0*(+4D0*(MT*MT*FCT14)+ -1D0)))
      FCT16 = 1D0/((-FCT15+ 1D0))
      FCT17 = LOG(DBLE(FCT16*(FCT15+ 1D0)))
      FCT18 = (-1D0*(+CI*(PI)-FCT17))**(2D0)
      FCT19 = SQRT(DBLE(1D0/4D0 *
FCT3*FCT3*FCT2))
      FCT20 = ASIN(DBLE(FCT19))
      FCT21 = (FCT20)**(2D0)
      FCT22 =
THETA_FUNCTION((+1D0/4D0*(FCT3*FCT3*FCT2)+
-1D0),
     $ -1D0/4D0 * FCT18,FCT21)
      FCT23 =
THETA_FUNCTION((+1D0/4D0*(FCT3*FCT3*TMP0)+
-1D0),1D0
     $ /2D0 * FCT6*(FCT8-CI*(PI)),FCT11*FCT6)
      FCT24 =
THETA_FUNCTION((+1D0/4D0*(FCT3*FCT3*FCT2)+
-1D0),1D0
     $ /2D0 * FCT15*(FCT17-CI*(PI)),FCT20*FCT15)
      FCT25 = (1D0/4D0 * FCT3*FCT3*(FCT2-
TMP0))**(2D0)
      FCT26 = 1D0/(FCT25)
      FCT27 = 1D0/(MB)
      FCT28 = 1D0/(1D0/4D0 * FCT27*FCT27*(FCT2-
TMP0))
      FCT29 = SQRT(DBLE(-1D0*(+4D0*(FCT5*MB*MB)+
-1D0)))
      FCT30 = 1D0/((-FCT29+ 1D0))
      FCT31 = LOG(DBLE(FCT30*(FCT29+ 1D0)))
      FCT32 = (-1D0*(+CI*(PI)-FCT31))**(2D0)
      FCT33 = SQRT(DBLE(1D0/4D0 *
TMP0*FCT27*FCT27))
      FCT34 = ASIN(DBLE(FCT33))
      FCT35 = (FCT34)**(2D0)
      FCT36 =
THETA_FUNCTION((+1D0/4D0*(TMP0*FCT27*FCT27)
+ -1D0),
     $ -1D0/4D0 * FCT32,FCT35)
      FCT37 = SQRT(DBLE(-1D0*(+4D0*(FCT14*MB*MB)
+ -1D0)))
      FCT38 = 1D0/(-FCT37+ 1D0))
      FCT39 = LOG(DBLE(FCT38*(FCT37+ 1D0)))
      FCT40 = (-1D0*(+CI*(PI)-FCT39))**(2D0)
      FCT41 = SQRT(DBLE(1D0/4D0 *
FCT27*FCT27*FCT2))
      FCT42 = ASIN(DBLE(FCT41))
      FCT43 = (FCT42)**(2D0)
      FCT44 =
THETA_FUNCTION((+1D0/4D0*(FCT2*FCT27*FCT27)
+ -1D0),
     $ -1D0/4D0 * FCT40,FCT43)
      FCT45 =
THETA_FUNCTION((+1D0/4D0*(TMP0*FCT27*FCT27)
+ -1D0),1D0
     $ /2D0 * FCT29*(FCT31-CI*(PI)),FCT34*FCT29)
      FCT46 =
THETA_FUNCTION((+1D0/4D0*(FCT2*FCT27*FCT27)
+ -1D0),1D0
     $ /2D0 * FCT37*(FCT39-CI*(PI)),FCT42*FCT37)
      FCT47 = (P3(0))**(2D0)
      FCT48 = (1D0/4D0 * FCT27*FCT27*(FCT47-
TMP0))**(2D0)
      FCT49 = 1D0/(FCT48)
      FCT50 = 1D0/(MW)
      FCT51 = 1D0/(1D0/4D0 * FCT50*FCT50*(FCT2-
TMP0))
      FCT52 = (CW)**(2D0)
      FCT53 = 1D0/(FCT52)
      FCT54 =
SQRT(DBLE(-1D0*(+4D0*(FCT5*MW*MW)+ -1D0)))
      FCT55 = 1D0/((-FCT54+ 1D0))
      FCT56 = LOG(DBLE(FCT55*(FCT54+ 1D0)))
      FCT57 = SQRT(DBLE(1D0/4D0 *
TMP0*FCT50*FCT50))
      FCT58 = ASIN(DBLE(FCT57))
      FCT59 =
THETA_FUNCTION((+1D0/4D0*(TMP0*FCT50*FCT50)
+ -1D0),1D0
     $ /2D0 * FCT54*(FCT56-CI*(PI)),FCT58*FCT54)
      FCT60 =
SQRT(DBLE(-1D0*(+4D0*(FCT14*MW*MW)+ -1D0)))
      FCT61 = 1D0/((-FCT60+ 1D0))
      FCT62 = LOG(DBLE(FCT61*(FCT60+ 1D0)))
      FCT63 = SQRT(DBLE(1D0/4D0 *
FCT50*FCT50*FCT2))
      FCT64 = ASIN(DBLE(FCT63))
      FCT65 =
THETA_FUNCTION((+1D0/4D0*(FCT2*FCT50*FCT50)
+ -1D0),1D0
     $ /2D0 * FCT60*(FCT62-CI*(PI)),FCT64*FCT60)
      FCT66 = (-1D0*(+CI*(PI)-FCT56))**(2D0)
      FCT67 = (FCT58)**(2D0)
      FCT68 =
THETA_FUNCTION((+1D0/4D0*(TMP0*FCT50*FCT50)
+ -1D0),
     $ -1D0/4D0 * FCT66,FCT67)
      FCT69 = (-1D0*(+CI*(PI)-FCT62))**(2D0)
      FCT70 = (FCT64)**(2D0)
      FCT71 =
THETA_FUNCTION((+1D0/4D0*(FCT2*FCT50*FCT50)
+ -1D0),
     $ -1D0/4D0 * FCT69,FCT70)
      VERTEX = COUP*1D0/16D0 *
FCT0*S3(3)*(FCT1*(TMP1*TMP2*(SW2*(FCT26
     $ *(FCT3*FCT3*(TMP0*8D0/3D0*(-
CI*(FCT22)+CI*(FCT13)-2D0 * CI
     $ *(FCT24)+2D0 * CI*(FCT23))+8D0/3D0*(FCT2*(-
CI*(FCT13)+CI
     $ *(FCT22))))+(-32D0/3D0 * CI*(FCT22)+32D0/3D0 *
CI*(FCT13)))
     $ +(FCT49*(FCT27*FCT27*(TMP0*2D0/3D0*(-
CI*(FCT44)+CI*(FCT36)
     $ -2D0 * CI*(FCT46)+2D0 * CI*(FCT45))
+2D0/3D0*(FCT2*(-CI*(FCT36)
     $ +CI*(FCT44))))+(-8D0/3D0 * CI*(FCT44)+8D0/3D0
* CI*(FCT36)))+(
     $ +8D0/3D0 * CI*(FCT28)+32D0/3D0 * CI*(FCT4))))
+(FCT26*(FCT3*FCT3
     $ *(TMP0*(-CI*(FCT13)+CI*(FCT22)-2D0 *
CI*(FCT23)+2D0 * CI
     $ *(FCT24))+FCT2*(-CI*(FCT22)+CI*(FCT13)))+
(+4D0 * CI*(FCT22)
     $ -4D0 * CI*(FCT13)))+(FCT49*(FCT27*1D0/2D0 *
FCT27*(FCT2*(
     $ -CI*(FCT44)+CI*(FCT36))+2D0*(TMP0*1D0/2D0*(-
CI*(FCT36)
     $ +CI*(FCT44)-2D0 * CI*(FCT45)+2D0 *
CI*(FCT46))))+(-2D0 * CI
     $ *(FCT36)+2D0 * CI*(FCT44)))+(-2D0 *
CI*(FCT28)-4D0 * CI
     $ *(FCT4)))))
+TMP3*TMP4*(SW2*(FCT26*(FCT3*FCT3*(TMP0*8D0/3
D0*(
     $ -CI*(FCT13)+CI*(FCT22)-2D0 * CI*(FCT23)+2D0 *
CI*(FCT24))
     $ +8D0/3D0*(FCT2*(-CI*(FCT22)+CI*(FCT13))))+
(-32D0/3D0 * CI
     $ *(FCT13)+32D0/3D0 * CI*(FCT22)))+
(FCT49*(FCT27*FCT27*(TMP0
     $ *2D0/3D0*(-CI*(FCT36)+CI*(FCT44)-2D0 *
CI*(FCT45)+2D0 * CI
     $ *(FCT46))+2D0/3D0*(FCT2*(-
CI*(FCT44)+CI*(FCT36))))+(-8D0/3D0
     $ * CI*(FCT36)+8D0/3D0 * CI*(FCT44)))+
(-32D0/3D0 * CI*(FCT4)
     $ -8D0/3D0 * CI*(FCT28))))+
(FCT26*(FCT3*FCT3*(TMP0*(-CI*(FCT22)
     $ +CI*(FCT13)-2D0 * CI*(FCT24)+2D0 * CI*(FCT23))
+FCT2*(-CI
     $ *(FCT13)+CI*(FCT22)))+(-4D0 * CI*(FCT22)+4D0 *
CI*(FCT13)))
     $ +(FCT49*(FCT27*1D0/2D0 * FCT27*(FCT2*(-
CI*(FCT36)+CI*(FCT44))
     $ +2D0*(TMP0*1D0/2D0*(-
CI*(FCT44)+CI*(FCT36)-2D0 * CI*(FCT46)
     $ +2D0 * CI*(FCT45))))+(-2D0 * CI*(FCT44)+2D0 *
CI*(FCT36)))+(
     $ +4D0 * CI*(FCT4)+2D0 * CI*(FCT28))))))
+CW*FCT51*(FCT51*(FCT50
     $
*FCT50*(TMP0*(TMP1*TMP2*(FCT53*SW2*(FCT2*1D0/
2D0 * FCT50*FCT50
     $ *(-CI*(FCT59)+CI*(FCT65))+(-
CI*(FCT59)+CI*(FCT65)-2D0 * CI
     $ *(FCT68)+2D0 * CI*(FCT71)))+(FCT2*1D0/2D0 *
FCT50*FCT50*(
     $ -CI*(FCT65)+CI*(FCT59))+(-5D0 *
CI*(FCT65)+6D0 * CI*(FCT68)
     $ -6D0 * CI*(FCT71)+5D0 * CI*(FCT59))))
+TMP3*TMP4*(FCT53*SW2
     $ *(FCT2*1D0/2D0 * FCT50*FCT50*(-
CI*(FCT65)+CI*(FCT59))+(
     $ *(FCT2*1D0/2D0 * FCT50*FCT50*(-
CI*(FCT65)+CI*(FCT59))+(
     $ -CI*(FCT65)+CI*(FCT59)-2D0 * CI*(FCT71)+2D0 *
CI*(FCT68)))
     $ +(FCT2*1D0/2D0 * FCT50*FCT50*(-
CI*(FCT59)+CI*(FCT65))+(
     $ -5D0 * CI*(FCT59)+6D0 * CI*(FCT71)-6D0 *
CI*(FCT68)+5D0 * CI
     $ *(FCT65)))))+FCT2*(TMP1*TMP2*(FCT53*SW2*(-
CI*(FCT71)+CI
     $ *(FCT68))+(-5D0 * CI*(FCT68)+5D0 * CI*(FCT71)))
+TMP3*TMP4
     $ *(FCT53*SW2*(-CI*(FCT68)+CI*(FCT71))+(-5D0 *
CI*(FCT71)
     $ +5D0 * CI*(FCT68)))))+(TMP1*TMP2*(FCT53*2D0
* SW2*(-CI*(FCT68)
     $ +CI*(FCT71))+(-10D0 * CI*(FCT71)+10D0 *
CI*(FCT68)))+TMP3*TMP4
     $ *(FCT53*2D0 * SW2*(-CI*(FCT71)+CI*(FCT68))+
(-10D0 * CI*(FCT68)
     $ +10D0 * CI*(FCT71)))))+(TMP1*TMP2*(FCT2*-
FCT50*FCT50*(
     $ +CI*(SW2*FCT53)+ -CI)+(-2D0 *
CI*(SW2*FCT53)+ 10D0 * CI))
     $
+TMP3*TMP4*(FCT2*FCT50*FCT50*(+CI*(SW2*FCT53)
+ -CI)+(+2D0 * CI
     $ *(SW2*FCT53)+ -10D0 * CI)))))
      END
```

4 Fermion with Majorana

- IOIO convention didn't work for 4 fermion vertex with identical particles.
  - how distinguish e+ e- e+ e- and e+ e- e+ e-
- IOIO is needed for defining the conjugate routine.
- Need to modify the flow information in MG5

LHE Reweighting

- Taking a LHE event (unweighted or not)
  - assign a weight to each event according to a new theoretical hyppothesis
- Nothing deep is needed (just an interface)

UserMod

- usermod V4 is still use A LOT
  - Do we want to keep v4 model for ever?
- Do we want to give a easy frameworks to customize a model?

Conclusion

- MG5 is a Framework for Matrix Element Work
- Waiting for YOUR idea / implementation
- Lot of development going on.
- https://launchpad.net/madgraph5

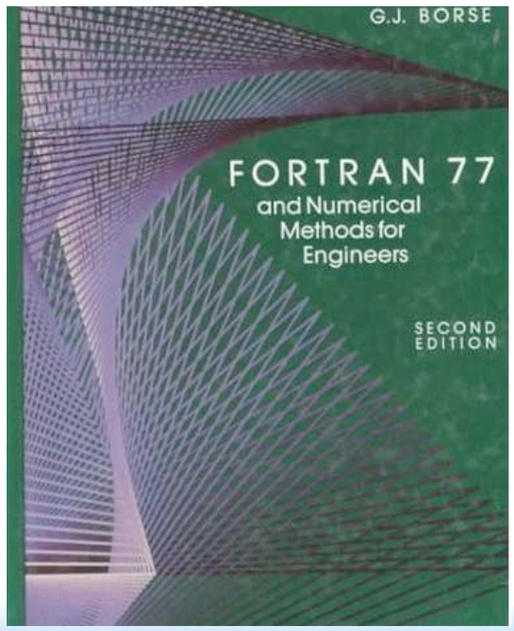
ALOHA

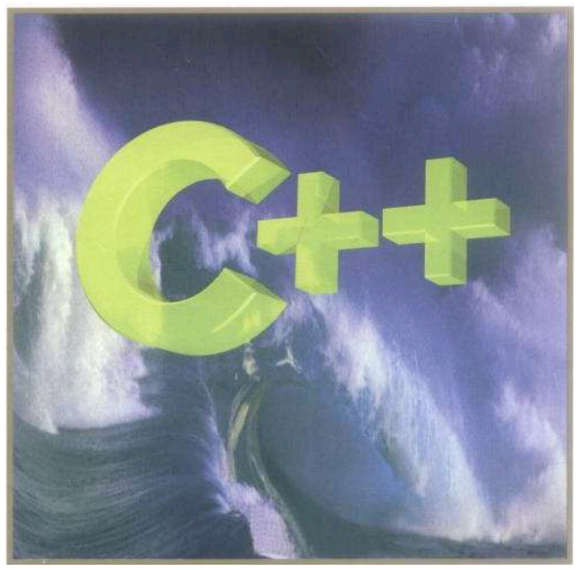
ALOHA

~~Google~~ translate

From: UFO To: Helicity Translate

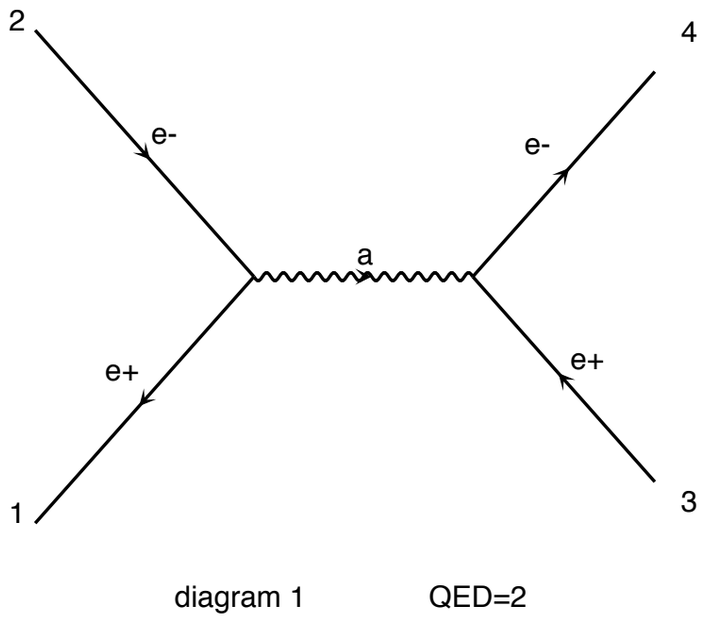
Type text or a website address or translate a document.





HELAS

- Idea: Evaluate $\mathcal{M}$ for fixed helicity of external particles.



$$M = \bar{u}\gamma^{\mu}v \; P_{\mu\nu} \; \bar{u}\gamma^{\nu}v$$

HELAS

- Idea: Evaluate $\mathcal{M}$ for fixed helicity of external particles.

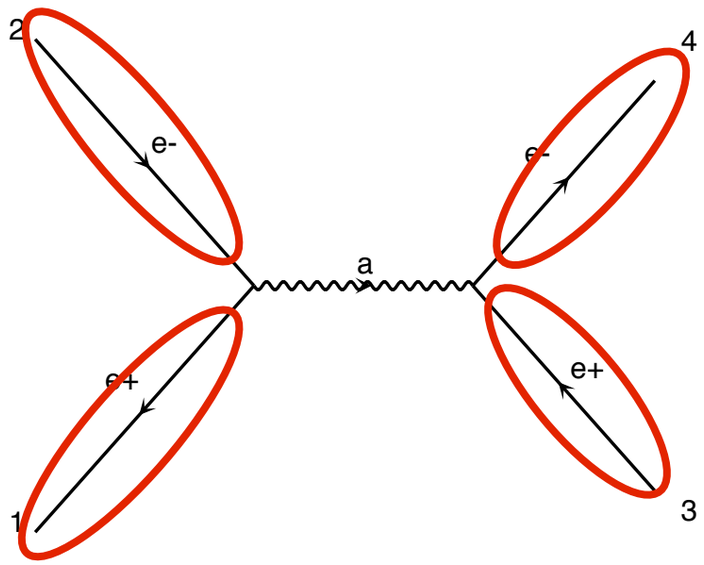


diagram 1 QED=2

$$M = \bar{u}\gamma^{\mu}v\, P_{\mu\nu}\, \bar{u}\gamma^{\nu}v$$

→ Number for a given helicity

```
CALL IXXXXX(P(0,1),ZERO,NHEL(1),+1*IC(1),W(1,1))
CALL OXXXXX(P(0,2),ZERO,NHEL(2),-1*IC(2),W(1,2))
CALL OXXXXX(P(0,3),MT,NHEL(3),+1*IC(3),W(1,3))
CALL IXXXXX(P(0,4),MT,NHEL(4),-1*IC(4),W(1,4))
```

HELAS

- Idea: Evaluate $\mathcal{M}$ for fixed helicity of external particles.

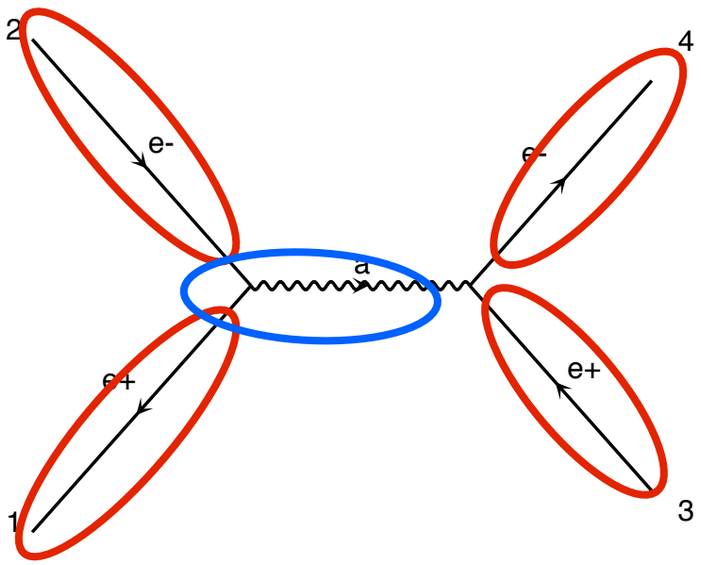


diagram 1 QED=2

$$M = \bar{u}\gamma^{\mu}v\, P_{\mu\nu}\, \bar{u}\gamma^{\nu}v$$

→ Number for a given helicity

→ Evaluate Interaction by interaction

```
CALL IXXXXX(P(0,1),ZERO,NHEL(1),+1*IC(1),W(1,1))
CALL OXXXXX(P(0,2),ZERO,NHEL(2),-1*IC(2),W(1,2))
CALL OXXXXX(P(0,3),MT,NHEL(3),+1*IC(3),W(1,3))
CALL IXXXXX(P(0,4),MT,NHEL(4),-1*IC(4),W(1,4))
CALL JIOXXX(W(1,1),W(1,2),GG,ZERO,ZERO,W(1,5))
```

HELAS

- Idea: Evaluate $\mathcal{M}$ for fixed helicity of external particles.

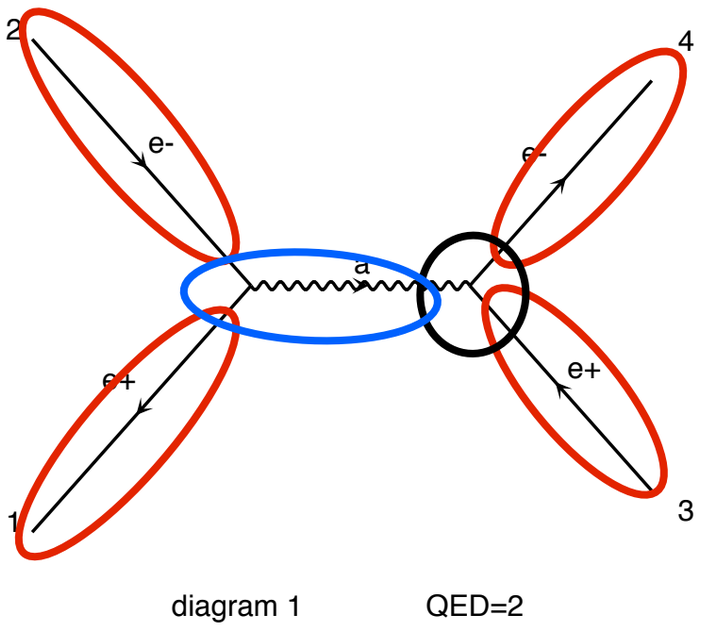


diagram 1 QED=2

$$M = \bar{u}\gamma^{\mu}v\,P_{\mu\nu}\,\bar{u}\gamma^{\nu}v$$

→ Number for a given helicity

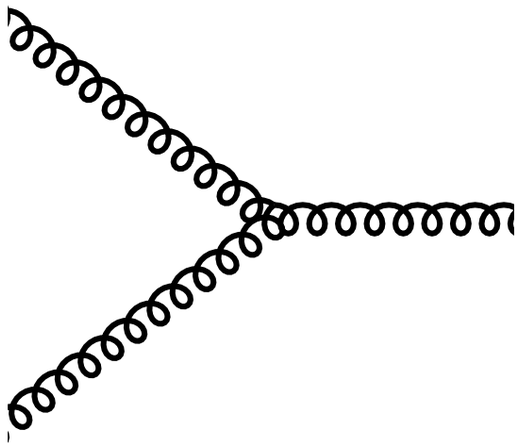
→ Evaluate Interaction by interaction

```
CALL IXXXXX(P(0,1),ZERO,NHEL(1),+1*IC(1),W(1,1))
CALL OXXXXX(P(0,2),ZERO,NHEL(2),-1*IC(2),W(1,2))
CALL OXXXXX(P(0,3),MT,NHEL(3),+1*IC(3),W(1,3))
CALL IXXXXX(P(0,4),MT,NHEL(4),-1*IC(4),W(1,4))
CALL JIOXXX(W(1,1),W(1,2),GG,ZERO,ZERO,W(1,5))
CALL IOVXXX(W(1,4),W(1,3),W(1,5),GG,AMP(1))
```

ALOHA

ALOHA

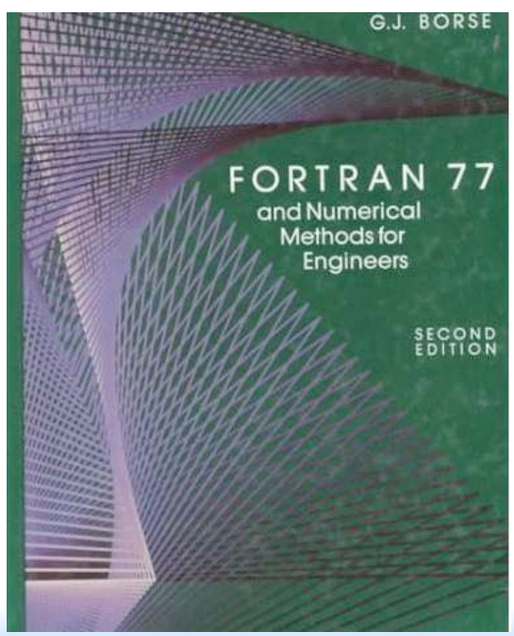
Google translate

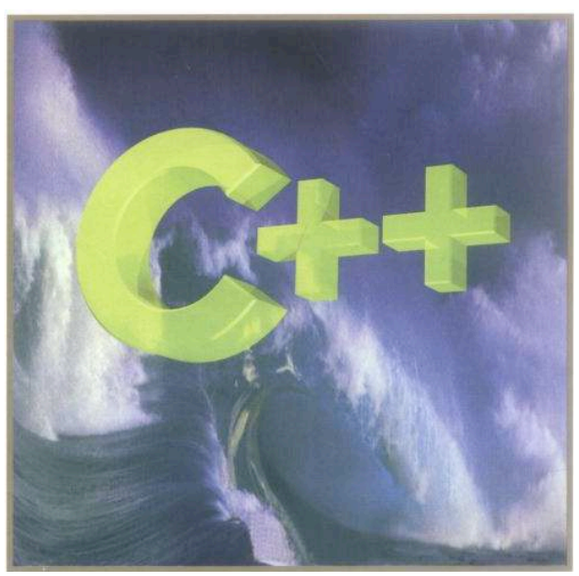


From: UFO To: Helicity Translate

P(3,1)*Metric(1,2) - P(3,2)*Metric(1,2) - P(2,1)*Metric(1,3) + P(2,3)*Metric(1,3) + P(1,2)*Metric(2,3) - P(1,3)*Metric(2,3)')

Type text or a website address or translate a document.





ALOHA

~~Google~~ translate

Amplitude

```
 TMP8 = (V1(3)*P3(0)-V1(4)*P3(1)-V1(5)*P3(2)-V1(6)*P3(3))
 TMP5 = (V2(3)*P3(0)-V2(4)*P3(1)-V2(5)*P3(2)-V2(6)*P3(3))
 TMP4 = (P1(0)*V2(3)-P1(1)*V2(4)-P1(2)*V2(5)-P1(3)*V2(6))
 TMP7 = (V1(3)*P2(0)-V1(4)*P2(1)-V1(5)*P2(2)-V1(6)*P2(3))
 TMP6 = (V3(3)*V2(3)-V3(4)*V2(4)-V3(5)*V2(5)-V3(6)*V2(6))
 TMP1 = (V2(3)*V1(3)-V2(4)*V1(4)-V2(5)*V1(5)-V2(6)*V1(6))
 TMP0 = (V3(3)*P1(0)-V3(4)*P1(1)-V3(5)*P1(2)-V3(6)*P1(3))
 TMP3 = (V3(3)*V1(3)-V3(4)*V1(4)-V3(5)*V1(5)-V3(6)*V1(6))
 TMP2 = (V3(3)*P2(0)-V3(4)*P2(1)-V3(5)*P2(2)-V3(6)*P2(3))
 VERTEX = COUP*(TMP1*(-CI*(TMP0)+CI*(TMP2))+(TMP3*(-CI*(TMP5)
$ +CI*(TMP4))+TMP6*(-CI*(TMP7)+CI*(TMP8))))
```

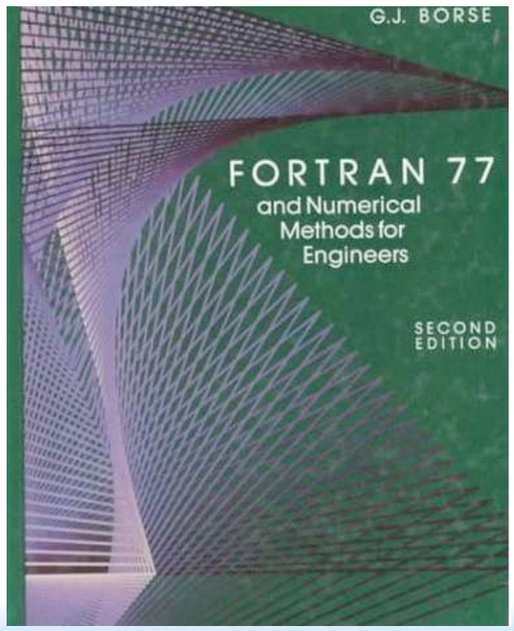
Wave-function

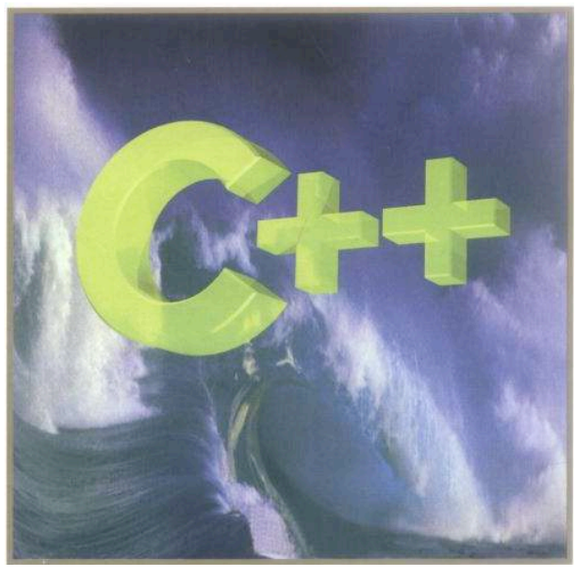
```
 TMP5 = (P1(0)*P2(0)-P1(1)*P2(1)-P1(2)*P2(2)-P1(3)*P2(3))
 TMP4 = (V3(3)*V2(3)-V3(4)*V2(4)-V3(5)*V2(5)-V3(6)*V2(6))
 TMP6 = (P1(0)*P3(0)-P1(1)*P3(1)-P1(2)*P3(2)-P1(3)*P3(3))
 TMP1 = (P1(0)*V2(3)-P1(1)*V2(4)-P1(2)*V2(5)-P1(3)*V2(6))
 TMP0 = (V3(3)*P1(0)-V3(4)*P1(1)-V3(5)*P1(2)-V3(6)*P1(3))
 TMP3 = (V2(3)*P3(0)-V2(4)*P3(1)-V2(5)*P3(2)-V2(6)*P3(3))
 TMP2 = (V3(3)*P2(0)-V3(4)*P2(1)-V3(5)*P2(2)-V3(6)*P2(3))
 DENOM = COUP/(P1(0)**2-P1(1)**2-P1(2)**2-P1(3)**2 - M1 * (M1
$ -CI* W1))
 V1(3)= DENOM*(OM1*P1(0)*(TMP4*(-CI*(TMP6)+CI*(TMP5))+(-CI*(TMP1
$ *TMP2)+CI*(TMP0*TMP3)))+(TMP4*(-CI*(P2(0))+CI*(P3(0)))
$ +(V2(3)*(-CI*(TMP0)+CI*(TMP2))+V3(3)*(-CI*(TMP3)+CI*(TMP1)))))
 V1(4)= DENOM*(OM1*P1(1)*(TMP4*(-CI*(TMP6)+CI*(TMP5))+(-CI*(TMP1
$ *TMP2)+CI*(TMP0*TMP3)))+(TMP4*(-CI*(P2(1))+CI*(P3(1)))
$ +(V2(4)*(-CI*(TMP0)+CI*(TMP2))+V3(4)*(-CI*(TMP3)+CI*(TMP1)))))
 V1(5)= DENOM*(OM1*P1(2)*(TMP4*(-CI*(TMP6)+CI*(TMP5))+(-CI*(TMP1
$ *TMP2)+CI*(TMP0*TMP3)))+(TMP4*(-CI*(P2(2))+CI*(P3(2)))
$ +(V2(5)*(-CI*(TMP0)+CI*(TMP2))+V3(5)*(-CI*(TMP3)+CI*(TMP1)))))
 V1(6)= DENOM*(OM1*P1(3)*(TMP4*(-CI*(TMP6)+CI*(TMP5))+(-CI*(TMP1
$ *TMP2)+CI*(TMP0*TMP3)))+(TMP4*(-CI*(P2(3))+CI*(P3(3)))
$ +(V2(6)*(-CI*(TMP0)+CI*(TMP2))+V3(6)*(-CI*(TMP3)+CI*(TMP1)))))
```

ALOHA

From: UFO To: Helicity Translate

Type text or a website address or translate a document.





ALOHA

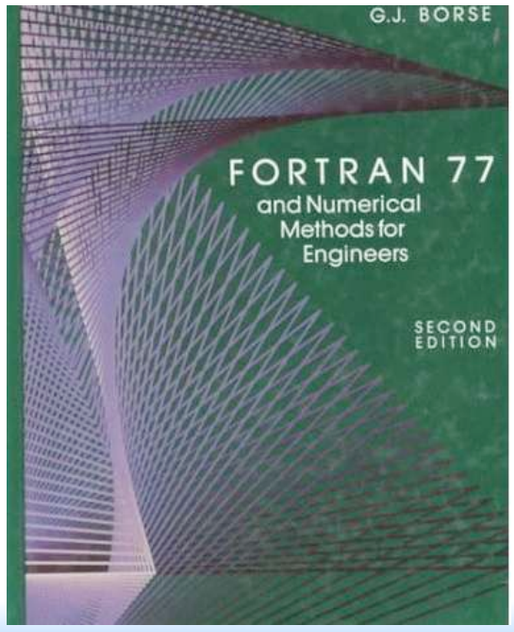
ALOHA

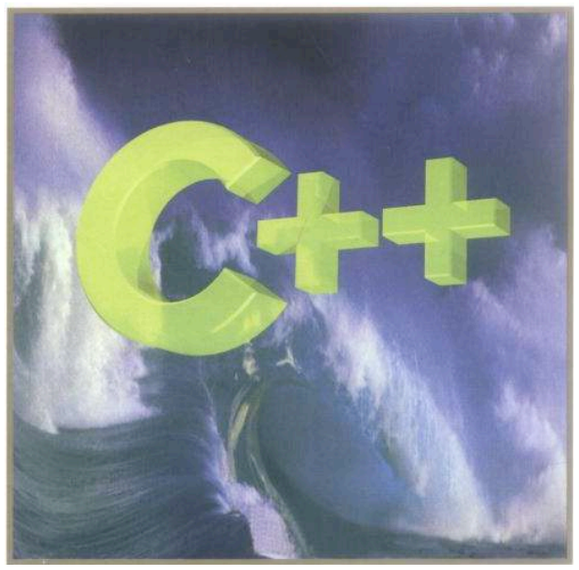
~~Google~~ translate

From: UFO To: Helicity Translate

Options:
- Standard (HELAS)
- Feynman gauge
- Complex-mass scheme
- Loop
- Open Loop

Type text or a website address or translate a document.





ALOHA

~~Google~~ translate

Wavefunction (Unitary)

$$\eta_{\mu\nu} + \frac{p_\mu p_\nu}{m^2}$$

Wavefunction (Feynman)

$$\eta_{\mu\nu}$$

```
 TMP5 = (P1(0)*P2(0)-P1(1)*P2(1)-P1(2)*P2(2)-P1(3)*P2(3))
 TMP4 = (V3(3)*V2(3)-V3(4)*V2(4)-V3(5)*V2(5)-V3(6)*V2(6))
 TMP6 = (P1(0)*P3(0)-P1(1)*P3(1)-P1(2)*P3(2)-P1(3)*P3(3))
 TMP1 = (P1(0)*V2(3)-P1(1)*V2(4)-P1(2)*V2(5)-P1(3)*V2(6))
 TMP0 = (V3(3)*P1(0)-V3(4)*P1(1)-V3(5)*P1(2)-V3(6)*P1(3))
 TMP3 = (V2(3)*P3(0)-V2(4)*P3(1)-V2(5)*P3(2)-V2(6)*P3(3))
 TMP2 = (V3(3)*P2(0)-V3(4)*P2(1)-V3(5)*P2(2)-V3(6)*P2(3))
 DENOM = COUP/(P1(0)**2-P1(1)**2-P1(2)**2-P1(3)**2 - M1 * (M1
$ -CI* W1))
 V1(3)= DENOM*(OM1*P1(0)*(TMP4*(-CI*(TMP6)+CI*(TMP5))+(-CI*(TMP1
$ *TMP2)+CI*(TMP0*TMP3)))+(TMP4*(-CI*(P2(0))+CI*(P3(0)))
$ +(V2(3)*(-CI*(TMP0)+CI*(TMP2))+V3(3)*(-CI*(TMP3)+CI*(TMP1)))))
 V1(4)= DENOM*(OM1*P1(1)*(TMP4*(-CI*(TMP6)+CI*(TMP5))+(-CI*(TMP1
$ *TMP2)+CI*(TMP0*TMP3)))+(TMP4*(-CI*(P2(1))+CI*(P3(1)))
$ +(V2(4)*(-CI*(TMP0)+CI*(TMP2))+V3(4)*(-CI*(TMP3)+CI*(TMP1)))))
 V1(5)= DENOM*(OM1*P1(2)*(TMP4*(-CI*(TMP6)+CI*(TMP5))+(-CI*(TMP1
$ *TMP2)+CI*(TMP0*TMP3)))+(TMP4*(-CI*(P2(2))+CI*(P3(2)))
$ +(V2(5)*(-CI*(TMP0)+CI*(TMP2))+V3(5)*(-CI*(TMP3)+CI*(TMP1)))))
 V1(6)= DENOM*(OM1*P1(3)*(TMP4*(-CI*(TMP6)+CI*(TMP5))+(-CI*(TMP1
$ *TMP2)+CI*(TMP0*TMP3)))+(TMP4*(-CI*(P2(3))+CI*(P3(3)))
$ +(V2(6)*(-CI*(TMP0)+CI*(TMP2))+V3(6)*(-CI*(TMP3)+CI*(TMP1)))))
```

```
 TMP4 = (V3(3)*V2(3)-V3(4)*V2(4)-V3(5)*V2(5)-V3(6)*V2(6))
 TMP1 = (V3(3)*P2(0)-V3(4)*P2(1)-V3(5)*P2(2)-V3(6)*P2(3))
 TMP0 = (V3(3)*P1(0)-V3(4)*P1(1)-V3(5)*P1(2)-V3(6)*P1(3))
 TMP3 = (V2(3)*P3(0)-V2(4)*P3(1)-V2(5)*P3(2)-V2(6)*P3(3))
 TMP2 = (P1(0)*V2(3)-P1(1)*V2(4)-P1(2)*V2(5)-P1(3)*V2(6))
 DENOM = COUP/(P1(0)**2-P1(1)**2-P1(2)**2-P1(3)**2 - M1 * (M1
$ -CI* W1))
 V1(3)= DENOM*(TMP4*(-CI*(P2(0))+CI*(P3(0)))+(V2(3)*(-CI*(TMP0)
$ +CI*(TMP1))+V3(3)*(-CI*(TMP3)+CI*(TMP2))))
 V1(4)= DENOM*(TMP4*(-CI*(P2(1))+CI*(P3(1)))+(V2(4)*(-CI*(TMP0)
$ +CI*(TMP1))+V3(4)*(-CI*(TMP3)+CI*(TMP2))))
 V1(5)= DENOM*(TMP4*(-CI*(P2(2))+CI*(P3(2)))+(V2(5)*(-CI*(TMP0)
$ +CI*(TMP1))+V3(5)*(-CI*(TMP3)+CI*(TMP2))))
 V1(6)= DENOM*(TMP4*(-CI*(P2(3))+CI*(P3(3)))+(V2(6)*(-CI*(TMP0)
$ +CI*(TMP1))+V3(6)*(-CI*(TMP3)+CI*(TMP2))))
```

- Improvement: Have a Two distinct routines for massive and massless vector. (crucial for open loop)

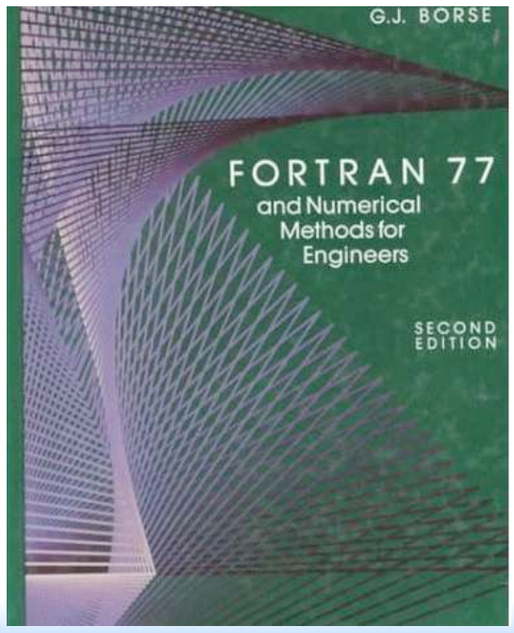
ALOHA

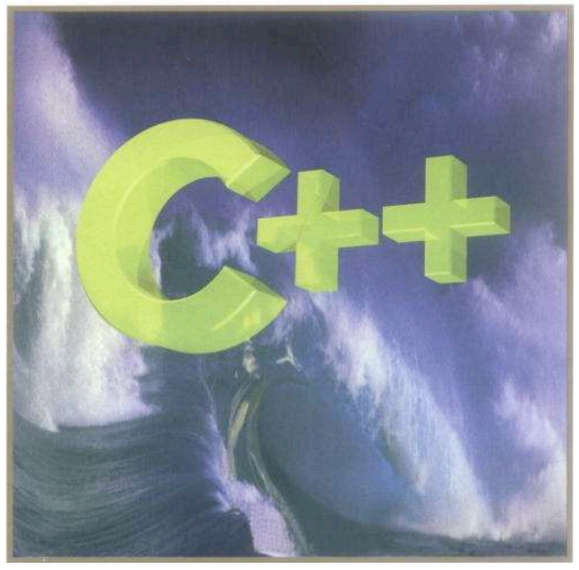
ALOHA

~~Google~~ translate

From: UFO To: Helicity Translate

Type text or a website address or translate a document.





ALOHA

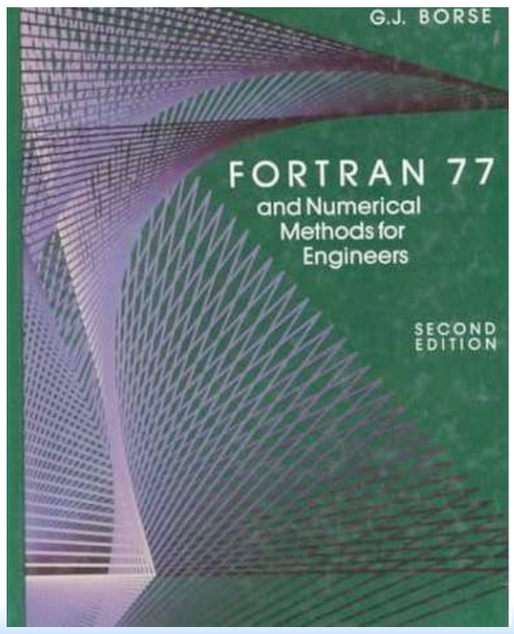
ALOHA

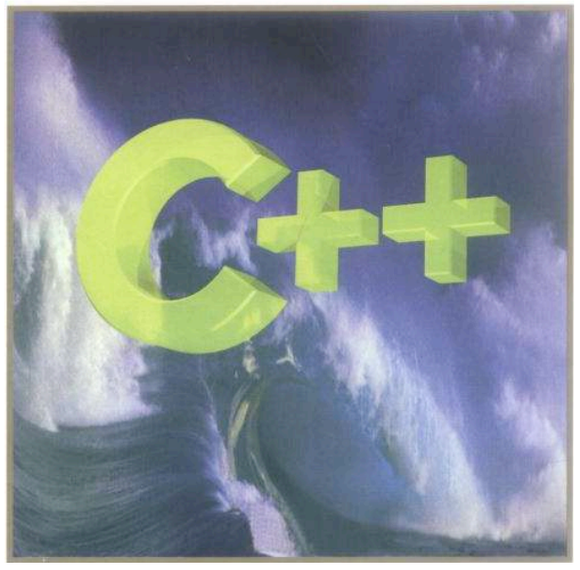
~~Google~~ translate

From: UFO To: Helicity Translate

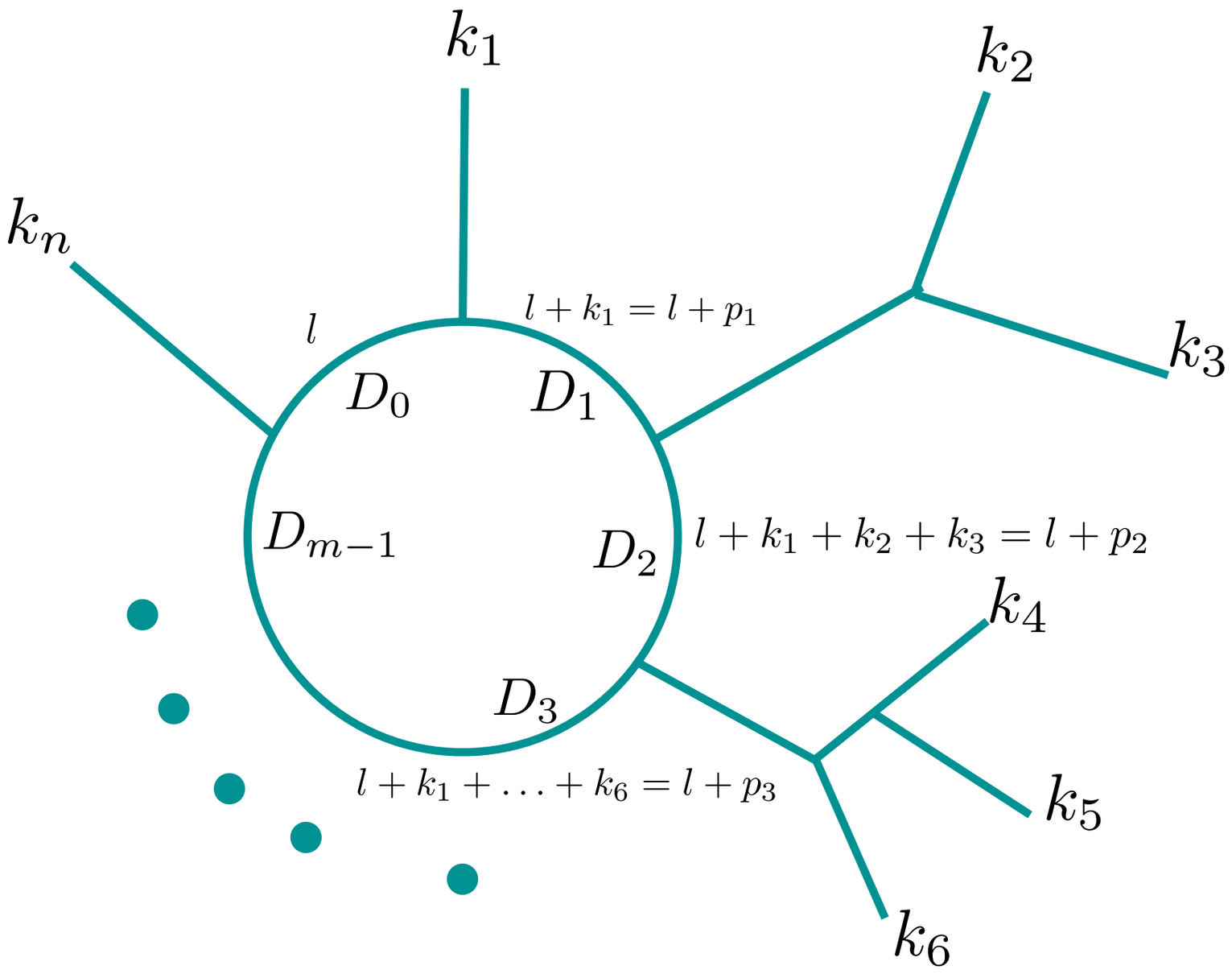
Options:
- Standard (HELAS)
- Feynman gauge
- Complex mass scheme
- Loop
- Open Loop

Type text or a website address or translate a document.





one-loop integral



- Consider this *m*-point loop diagram with *n* external momenta
- The integral to compute is

$$\int d^d l \, \frac{N(l)}{D_0 D_1 D_2 \cdots D_{m-1}}$$

$$D_i = (l + p_i)^2 - m_i^2$$

Open Loop

$$\mathcal{N}(l^\mu) = \sum_{r=0}^{r_{max}} C^{(r)}_{\mu_0\mu_1\cdots\mu_r} l^{\mu_0} l^{\mu_1} \cdots l^{\mu_r}$$

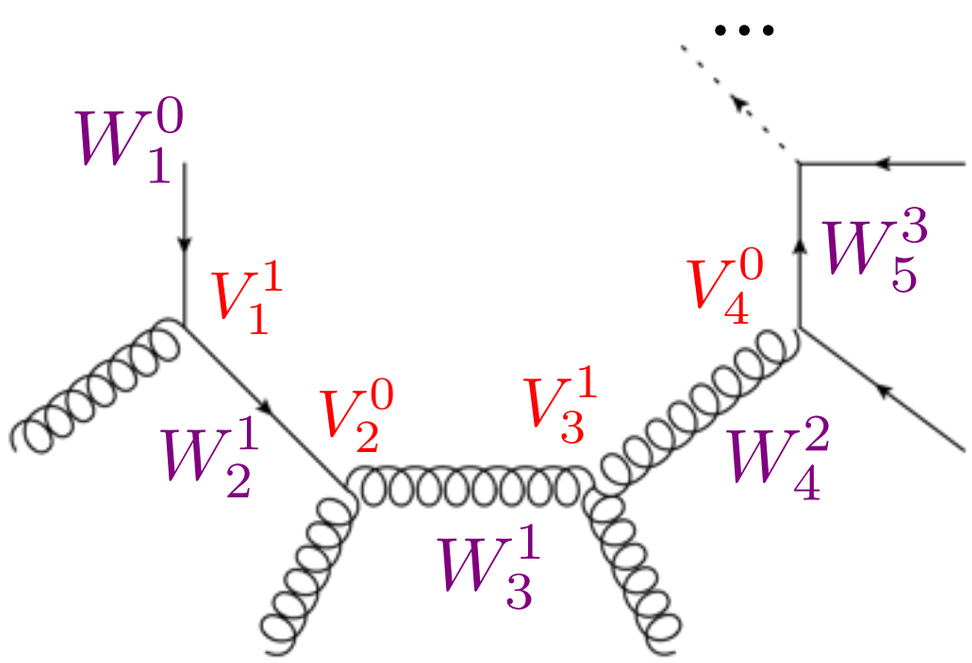
$$V_j^{(r=0,1)} = \sum_{i=0}^{r} v_j^i l^i$$

Computed by aloha

Open Loop

$$\mathcal{N}(l^\mu) = \sum_{r=0}^{r_{max}} C^{(r)}_{\mu_0\mu_1\cdots\mu_r} l^{\mu_0} l^{\mu_1} \cdots l^{\mu_r}$$

- How to get these coefficients? (Wavefunction and 4-momenta indices now omitted)



$$V_j^{(r=0,1)} = \sum_{i=0}^{r} v_j^i l^i$$

Computed by aloha

```
 TMP1 = (V3(5)*P2(0)-V3(6)*P2(1)-V3(7)*P2(2)-V3(8)*P2(3))
 TMP0 = (V3(5)*P1(0)-V3(6)*P1(1)-V3(7)*P1(2)-V3(8)*P1(3))
 COEFF(1,0,1)= COUP*(V3(5)*(+CI*(P1(0)+P2(0)))+(-CI*(TMP0+TMP1)))
 COEFF(2,0,1)= COUP*(V3(5)*(-CI*(P3(1))+CI*(P1(1)))+V3(6)*(
$ +CI*(P2(0)+P3(0))))
 COEFF(3,0,1)= COUP*(V3(5)*(-CI*(P3(2))+CI*(P1(2)))+V3(7)*(
$ +CI*(P2(0)+P3(0))))
 COEFF(4,0,1)= COUP*(V3(5)*(-CI*(P3(3))+CI*(P1(3)))+V3(8)*(
$ +CI*(P2(0)+P3(0))))
 COEFF(1,1,1)= 0D0
 COEFF(2,1,1)= COUP*CI * V3(6)
 COEFF(3,1,1)= COUP*CI * V3(7)
 COEFF(4,1,1)= COUP*CI * V3(8)
 COEFF(1,2,1)= COUP*2D0 * CI * V3(6)
 COEFF(2,2,1)= COUP*CI * V3(5)
 COEFF(3,2,1)= 0D0
 COEFF(4,2,1)= 0D0
 COEFF(1,3,1)= COUP*2D0 * CI * V3(7)
 COEFF(2,3,1)= 0D0
 COEFF(3,3,1)= COUP*CI * V3(5)
 COEFF(4,3,1)= 0D0
 COEFF(1,4,1)= COUP*2D0 * CI * V3(8)
 COEFF(2,4,1)= 0D0
 COEFF(3,4,1)= 0D0
 COEFF(4,4,1)= COUP*CI * V3(5)
 COEFF(1,0,2)= COUP*-1D0*(V3(5)*(+CI*(P2(1)+P3(1)))+V3(6)*(
$ -CI*(P3(0))+CI*(P1(0))))
 COEFF(2,0,2)= COUP*-1D0*(V3(6)*(+CI*(P1(1)+P2(1)))+(+CI*(TMP0
$ +TMP1)))
 COEFF(3,0,2)= COUP*-1D0*(V3(6)*(-CI*(P3(2))+CI*(P1(2)))+V3(7)*(
$ +CI*(P2(1)+P3(1))))
 COEFF(4,0,2)= COUP*-1D0*(V3(6)*(-CI*(P3(3))+CI*(P1(3)))+V3(8)*(
$ +CI*(P2(1)+P3(1))))
 COEFF(1,1,2)= COUP*-CI * V3(6)
 COEFF(2,1,2)= COUP*-2D0 * CI * V3(5)
 COEFF(3,1,2)= 0D0
 COEFF(4,1,2)= 0D0
 COEFF(1,2,2)= COUP*-CI * V3(5)
 COEFF(2,2,2)= 0D0
 COEFF(3,2,2)= COUP*-CI * V3(7)
 COEFF(4,2,2)= COUP*-CI * V3(8)
 COEFF(1,3,2)= 0D0
 COEFF(2,3,2)= COUP*2D0 * CI * V3(7)
 COEFF(3,3,2)= COUP*-CI * V3(6)
 COEFF(4,3,2)= 0D0
 COEFF(1,4,2)= 0D0
 COEFF(2,4,2)= COUP*2D0 * CI * V3(8)
 COEFF(3,4,2)= 0D0
 COEFF(4,4,2)= COUP*-CI * V3(6)
 COEFF(1,0,3)= COUP*-1D0*(V3(5)*(+CI*(P2(2)+P3(2)))+V3(7)*(
$ -CI*(P3(0))+CI*(P1(0))))
```

ALOHA Google translate

```
 COEFF(2,0,3)= COUP*-1D0*(V3(6)*(+CI*(P2(2)+P3(2)))+V3(7)*(
     $ -CI*(P3(1))+CI*(P1(1))))
      COEFF(3,0,3)= COUP*-1D0*(V3(7)*(+CI*(P1(2)+P2(2)))+(+CI*(TMP0
     $ +TMP1)))
      COEFF(4,0,3)= COUP*-1D0*(V3(7)*(-CI*(P3(3))+CI*(P1(3)))+V3(8)*(
     $ +CI*(P2(2)+P3(2))))
      COEFF(1,1,3)= COUP*-CI * V3(7)
      COEFF(2,1,3)= 0D0
      COEFF(3,1,3)= COUP*-2D0 * CI * V3(5)
      COEFF(4,1,3)= 0D0
      COEFF(1,2,3)= 0D0
      COEFF(2,2,3)= COUP*-CI * V3(7)
      COEFF(3,2,3)= COUP*2D0 * CI * V3(6)
      COEFF(4,2,3)= 0D0
      COEFF(1,3,3)= COUP*-CI * V3(5)
      COEFF(2,3,3)= COUP*-CI * V3(6)
      COEFF(3,3,3)= 0D0
      COEFF(4,3,3)= COUP*-CI * V3(8)
      COEFF(1,4,3)= 0D0
      COEFF(2,4,3)= 0D0
      COEFF(3,4,3)= COUP*2D0 * CI * V3(8)
      COEFF(4,4,3)= COUP*-CI * V3(7)
      COEFF(1,0,4)= COUP*-1D0*(V3(5)*(+CI*(P2(3)+P3(3)))+V3(8)*(
     $ -CI*(P3(0))+CI*(P1(0))))
      COEFF(2,0,4)= COUP*-1D0*(V3(6)*(+CI*(P2(3)+P3(3)))+V3(8)*(
     $ -CI*(P3(1))+CI*(P1(1))))
      COEFF(3,0,4)= COUP*-1D0*(V3(7)*(+CI*(P2(3)+P3(3)))+V3(8)*(
     $ -CI*(P3(2))+CI*(P1(2))))
      COEFF(4,0,4)= COUP*-1D0*(V3(8)*(+CI*(P1(3)+P2(3)))+(+CI*(TMP0
     $ +TMP1)))
      COEFF(1,1,4)= COUP*-CI * V3(8)
      COEFF(2,1,4)= 0D0
      COEFF(3,1,4)= 0D0
      COEFF(4,1,4)= COUP*-2D0 * CI * V3(5)
      COEFF(1,2,4)= 0D0
      COEFF(2,2,4)= COUP*-CI * V3(8)
      COEFF(3,2,4)= 0D0
      COEFF(4,2,4)= COUP*2D0 * CI * V3(6)
      COEFF(1,3,4)= 0D0
      COEFF(2,3,4)= 0D0
      COEFF(3,3,4)= COUP*-CI * V3(8)
      COEFF(4,3,4)= COUP*2D0 * CI * V3(7)
      COEFF(1,4,4)= COUP*-CI * V3(5)
      COEFF(2,4,4)= COUP*-CI * V3(6)
      COEFF(3,4,4)= COUP*-CI * V3(7)
      COEFF(4,4,4)= 0D0
      END
```