ALOHA

Mattelaer Olivier (UCL-CP3)

Priscilla Aquino (UCL-CP3)

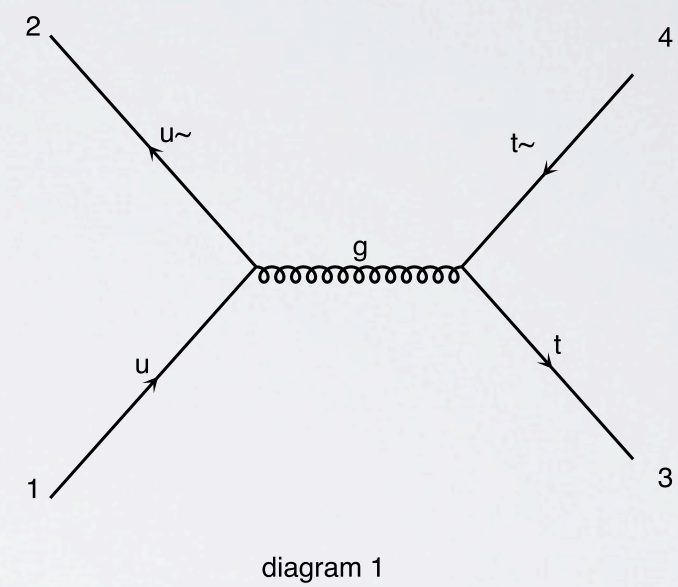
William Link (UIUC)

PURPOSE OF ALOHA

- ALOHA = Automatic Language-independent Output of Helicity Amplitude.
- Idea: made BSM model implementation 100% automatic from FeynRules to detector simulation.
  - Color: reason for the new color module of MG5
  - HELAS: Need an automation

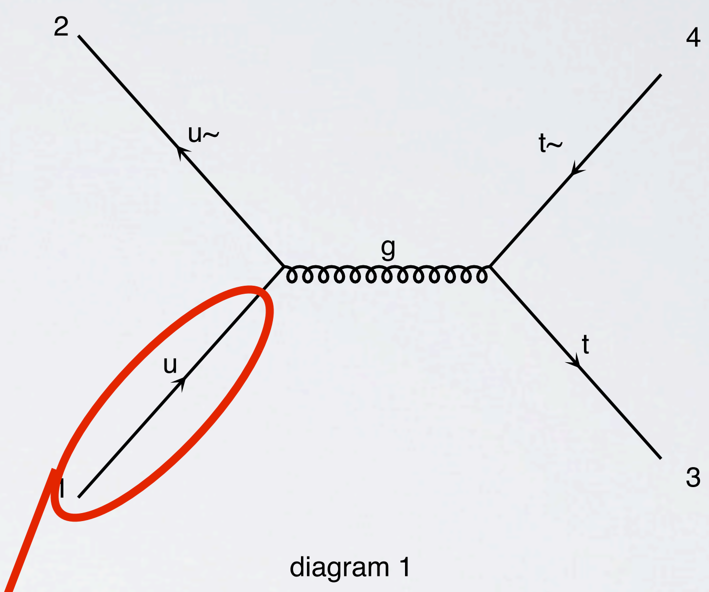
STANDARD HELAS

- Idea: Evaluate $\mathcal{M}$ from impulsions and helicity of external particles.



STANDARD HELAS

- Idea: Evaluate $\mathcal{M}$ from impulsions and helicity of external particles.
- 1: Evaluate Wavefunctions of external particles




```
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))
```

STANDARD HELAS

- Idea: Evaluate $\mathcal{M}$ from impulsions and helicity of external particles.
- 1 : Evaluate Wavefunctions of external particles
- 2 : Evaluate Wavefunctions of internal particles

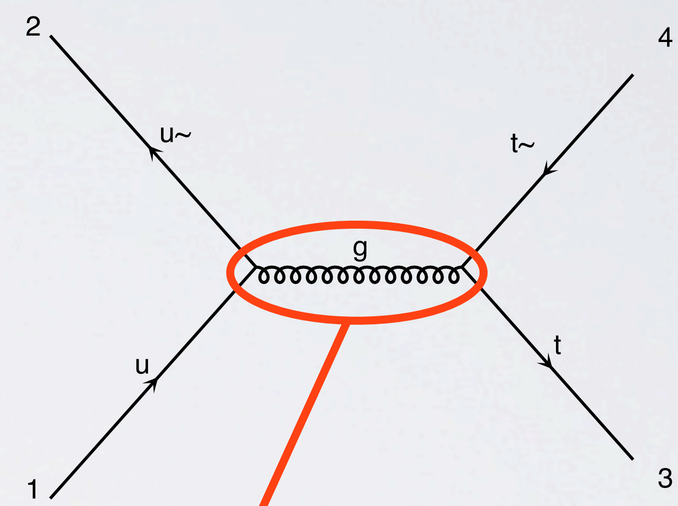
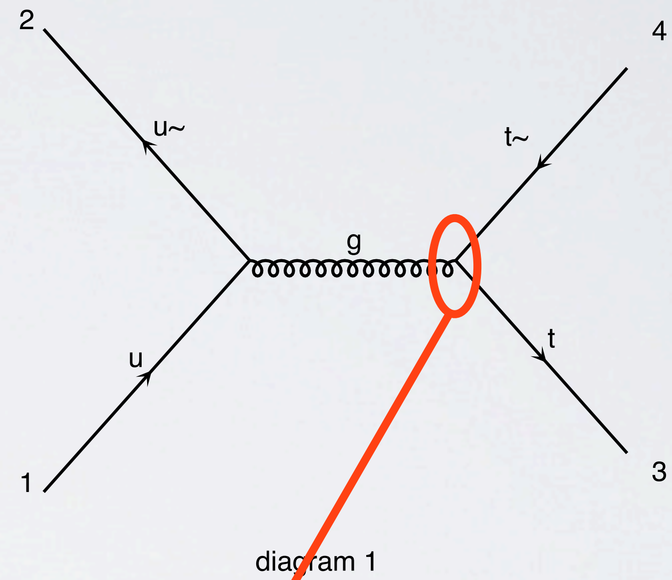


diagram 1

```
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))
```

STANDARD HELAS

- Idea: Evaluate $\mathcal{M}$ from impulsions and helicity of external particles.
- 1: Evaluate Wavefunctions of external particles
- 2 : Evaluate Wavefunctions of internal particles
- 3 : Evaluate the Amplitude




```
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))
```

ONE HELAS ROUTINE

```
         if ( gc(2).ne.cZero ) then
            c0 =  gc(1)*( fo(3)*fi(1)+fo(4)*fi(2))
     &           +gc(2)*( fo(1)*fi(3)+fo(2)*fi(4))
            c1 = -gc(1)*( fo(3)*fi(2)+fo(4)*fi(1))
     &           +gc(2)*( fo(1)*fi(4)+fo(2)*fi(3))
            c2 =( gc(1)*( fo(3)*fi(2)-fo(4)*fi(1))
     &           +gc(2)*(-fo(1)*fi(4)+fo(2)*fi(3)))*cImag
            c3 =  gc(1)*(-fo(3)*fi(1)+fo(4)*fi(2))
     &           +gc(2)*( fo(1)*fi(3)-fo(2)*fi(4))
         else
            d = d*gc(1)
            c0 =   fo(3)*fi(1)+fo(4)*fi(2)
            c1 =  -fo(3)*fi(2)-fo(4)*fi(1)
            c2 = ( fo(3)*fi(2)-fo(4)*fi(1))*cImag
            c3 =  -fo(3)*fi(1)+fo(4)*fi(2)
         end if

c     Fabio's implementation of the fixed width
         cm2=dcmplx( vm2, -vmass*vwidth )
c     cs = (q(0)*c0-q(1)*c1-q(2)*c2-q(3)*c3)/vm2
         cs = (q(0)*c0-q(1)*c1-q(2)*c2-q(3)*c3)/cm2
         jio(1) = (c0-cs*q(0))*d
         jio(2) = (c1-cs*q(1))*d
         jio(3) = (c2-cs*q(2))*d
         jio(4) = (c3-cs*q(3))*d

      else

         d = dcmplx( rOne/q2, rZero )
```

PHYSICAL CONTENT

- Lorentz structure associated to "e+ e- A" vertex is γ^μ
- So the Associate Amplitude (IOV) will be:

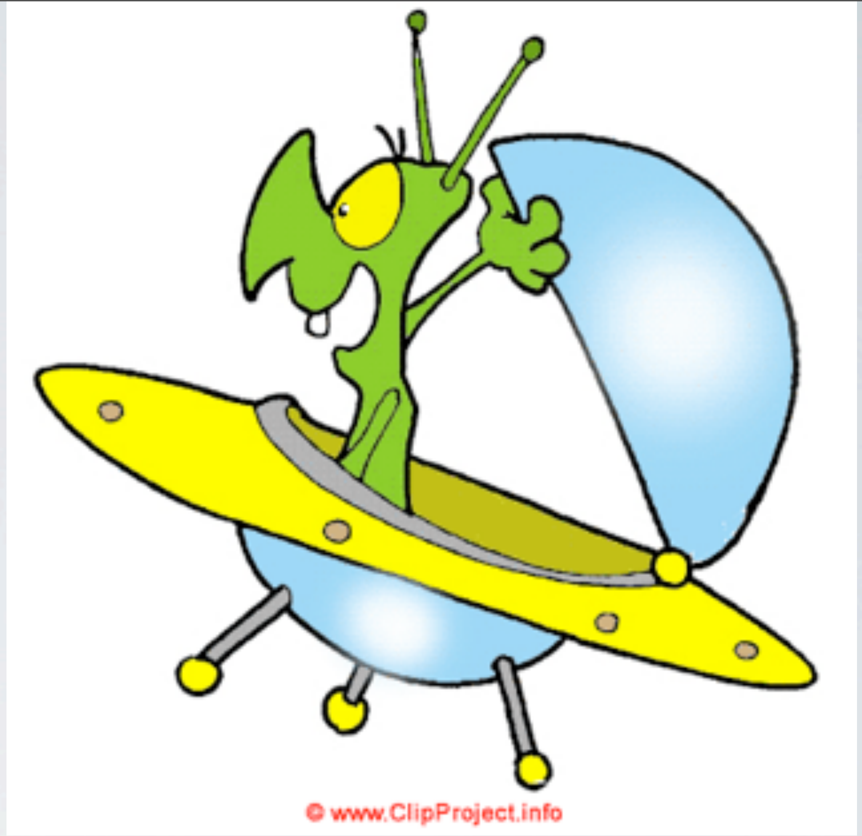
$$-i\, W_f(e^-)\, \gamma^\mu\, W_f(e^+)\, A_\mu$$

- And the computation of the vector wavefunctions (JIO) is

$$W_f(e^-)\, \gamma^\mu\, W_f(e^+)\, \frac{-i\, \eta_{\mu\nu}}{p_A^2}$$

- From the Lorentz structure it's easy to compute automaticaly the HELAS routine

UFO



Vertices.py

```
V_15 = Vertex(name = 'V_15',
              particles = [ P.s__tilde__, P.s, P.A ],
              color = [ 'Identity(1,2)' ],
              lorentz = [ L.FFV1 ],
              couplings = {(0,0):C.GC_1})

V_16 = Vertex(name = 'V_16',
              particles = [ P.b__tilde__, P.b, P.A ],
              color = [ 'Identity(1,2)' ],
              lorentz = [ L.FFV1 ],
              couplings = {(0,0):C.GC_1})

V_17 = Vertex(name = 'V_17',
              particles = [ P.e__plus__, P.e__minus__, P.A ],
              color = [ '1' ],
              lorentz = [ L.FFV1 ],
              couplings = {(0,0):C.GC_3})
```

Lorentz.py

```
SSS1 = Lorentz(name = 'SSS1',
               spins = [ 1, 1, 1 ],
               structure = '1')

FFS1 = Lorentz(name = 'FFS1',
               spins = [ 2, 2, 1 ],
               structure = 'Identity(1,2)')

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

γ^μ

ALOHA

- Idea: Evaluate $\mathcal{M}$ from impulsions and helicity of external particles.
- 1 : Evaluate Wavefunctions of external particles

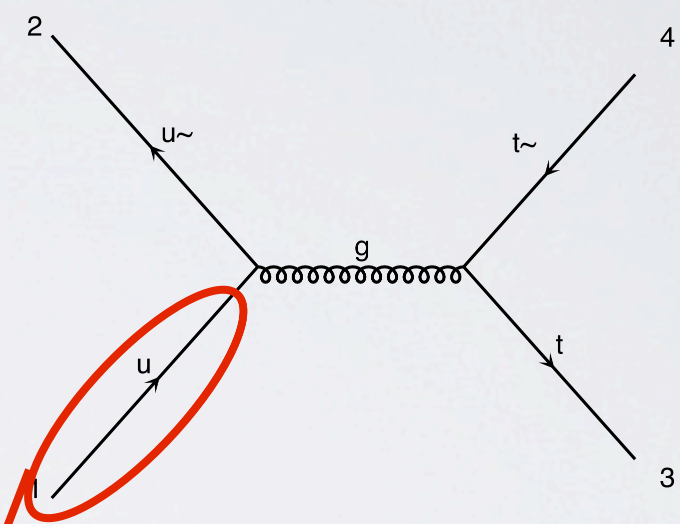
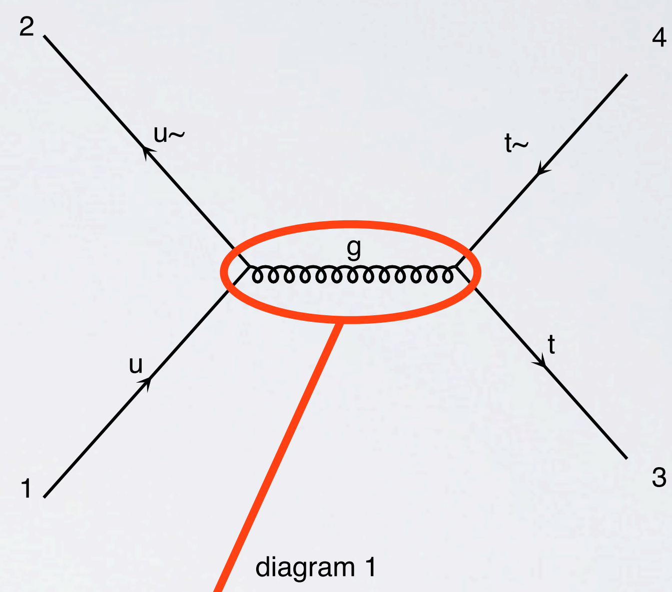


diagram 1

```
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))
```

STANDARD HELAS

- Idea: Evaluate $\mathcal{M}$ from impulsions and helicity of external particles.
- 1: Evaluate Wavefunctions of external particles
- 2 : Evaluate Wavefunctions of internal particles




```
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 FFV1_3(W(1,1),W(1,2),GC_5,ZERO, ZERO, W(1,5))
```

STANDARD HELAS

- Idea: Evaluate $\mathcal{M}$ from impulsions and helicity of external particles.
- 1 : Evaluate Wavefunctions of external particles
- 2 : Evaluate Wavefunctions of internal particles

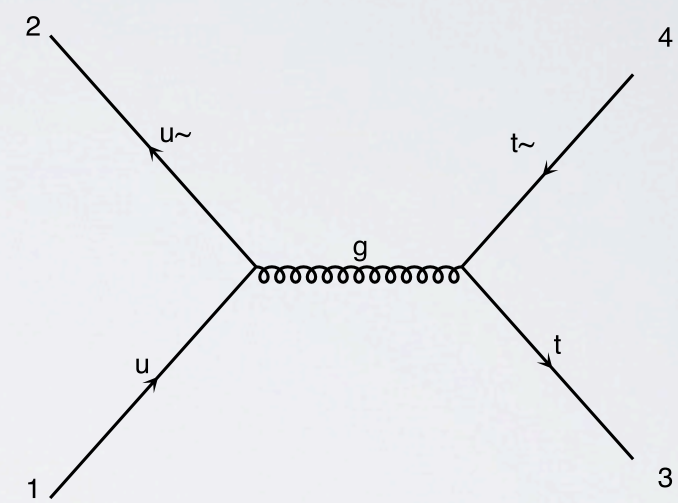


diagram 1

```
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 FFV1_3(W(1,1),W(1,2),GC_5,ZERO, ZERO, W(1,5))
```

```
V_36 = Vertex(name = 'V_36',
              particles = [ P.u__tilde__, P.u, P.G ],
              color = [ 'T(3,2,1)' ],
              lorentz = [ L.FFV1 ],
              couplings = {(0,0):C.GC_5})
```

STANDARD HELAS

- Idea: Evaluate $\mathcal{M}$ from impulsions and helicity of external particles.
- 1 : Evaluate Wavefunctions of external particles
- 2 : Evaluate Wavefunctions of internal particles

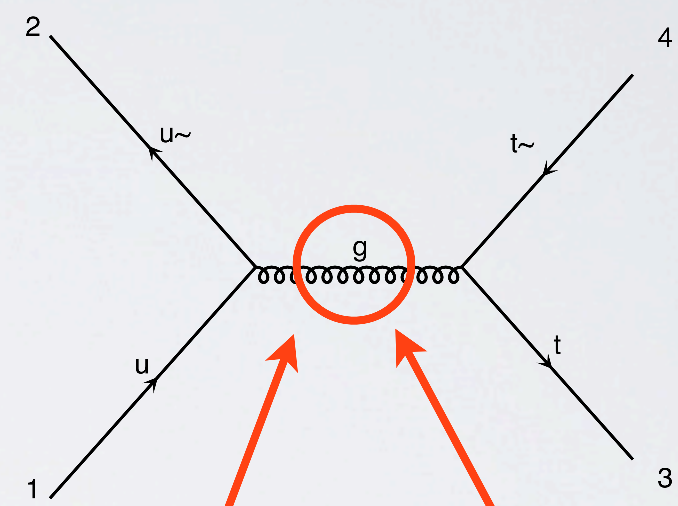


diagram 1

```
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 FFV1_3(W(1,1),W(1,2),GC_5,ZERO, ZERO, W(1,5))
```

```
V_36 = Vertex(name = 'V_36',
              particles = [ P.u__tilde__, P.u, P.G ],
              color = [ 'T(3,2,1)' ],
              lorentz = [ L.FFV1 ],
              couplings = {(0,0):C.GC_5})
```

STANDARD HELAS

- Idea: Evaluate $\mathcal{M}$ from impulsions and helicity of external particles.
- 1: Evaluate Wavefunctions of external particles
- 2 : Evaluate Wavefunctions of internal particles
- 3 : Evaluate the Amplitude

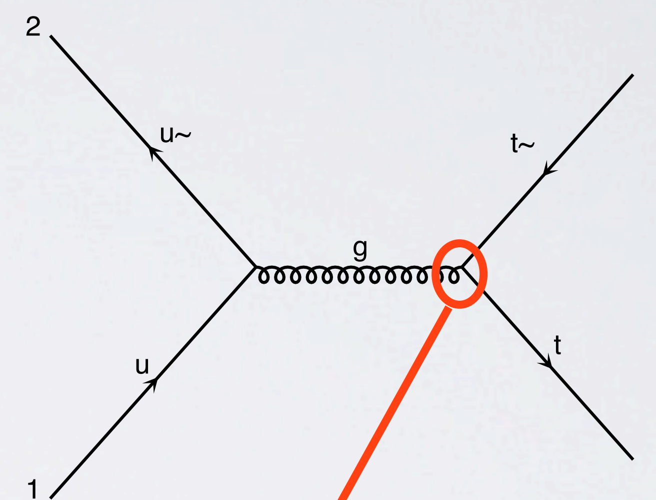


diagram 1

```
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 FFV1_3(W(1,1),W(1,2),GC_5,ZERO, ZERO, W(1,5))
CALL FFV1_0(W(1,4),W(1,3),W(1,5),GC_5,AMP(1))
```

ONE ALOHA ROUTINE

```
C       This File is Automatically generated by ALOHA
C       The process calculated in this file is:
C       Gamma(3,2,1)
C
        SUBROUTINE FFV1_0(F1,F2,V3,C,VERTEX)
        IMPLICIT NONE
        DOUBLE COMPLEX F1(6)
        DOUBLE COMPLEX F2(6)
        DOUBLE COMPLEX V3(6)
        DOUBLE COMPLEX C
        DOUBLE COMPLEX VERTEX



        VERTEX = C*( (F2(1)*( (F1(3)*( (0, -1)*V3(1)+(0, 1)*V3(4)))
       $ +(F1(4)*( (0, 1)*V3(2)+V3(3)))))+( (F2(2)*( (F1(3)*( (0, 1)
       $ *V3(2)-V3(3)))+(F1(4)*( (0, -1)*V3(1)+(0, -1)*V3(4)))))
       $ +( (F2(3)*( (F1(1)*( (0, -1)*V3(1)+(0, -1)*V3(4)))+(F1(2)
       $ *( (0, -1)*V3(2)-V3(3)))))+(F2(4)*( (F1(1)*( (0, -1)*V3(2)
       $ +V3(3)))+(F1(2)*( (0, -1)*V3(1)+(0, 1)*V3(4)))))))

        END
```

ALOHA FEATURE

- ALOHA IS PURE PYTHON
- ALOHA IS FAST
  - SM in 3s and MSSM in 5s
- Possible to ask a subset of routine (Done in MG5)
- Output in Python / Fortran / C
- Particles spin implemented Scalar Fermion Vector
- Spin2 in progress (This week)

DEMO

PERSPECTIVE

- Spin 2
- Spin 3/2
- 4 Fermion Interactions (and more)
- GPU

CONCLUSION

- The Helas routine for BSM without the pain to write it.
- Fully interfaced with MG5.