

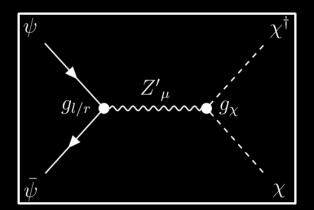
### Resumo do problema

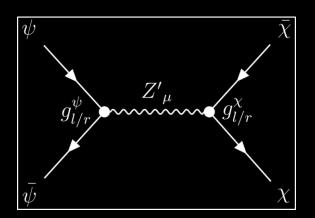


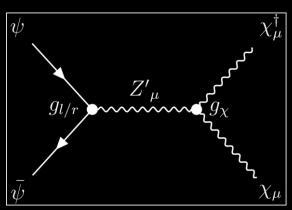
► Aplicamos um modelo simplificado para uma ME do tipo WIMP.



► Especial interesse no cálculo da densidade de relíquia da ME para produção via ressonância do Z'.







► Onde o Z' possui propagador análogo ao Z do modelo padrão, na forma:

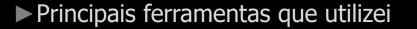
$$-i\frac{g_{\mu\nu} + \frac{q_{\mu}q_{\nu}}{M^2}}{q^2 - M^2 + iM\Gamma}$$

### **Softwares e Pacotes**

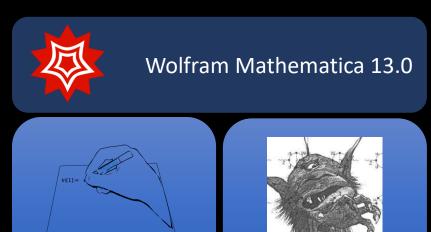
UFRGS
UNIVERSIDADE FEDERAL
DO RIO GRANDE DO SUL











**FeynArts** 

FeynCalc

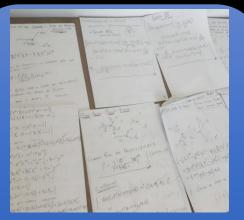
# Obtendo as seções de choque





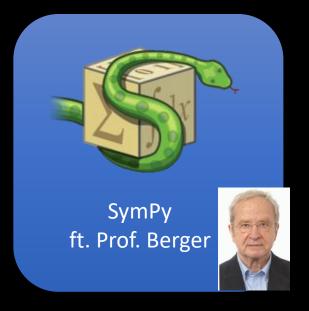






Manualmente (como faziam os...)





### SymPy – Exemplo $e^+e^- \rightarrow W^+W^-$



```
iF)
```

```
import sympy as sp
      import heppackv0 as hep
                                                                                                                                                                                                                  Python
      theta, u, t, r = sp.symbols('theta u t r', real=True)
      s, p = sp.symbols('s p', positive=True)
      M1, M2, m = sp.symbols('M1 M2 m', real=True)
      g, c1, c2 = sp.symbols('g c1 c2')
      pi = sp.pi
   ✓ 0.0s
                                                                                                                                                                                                                  Python
Production of longitudinal W-Bosons, investigation of amplitudes violating unitarity.
    1. e_{\perp}^- e_{\perp}^+ \rightarrow W_0^- W_0^+ Standard weak interaction
      p1 = [p, m, 0, 0]
      p2 = [p, m, pi, pi]
      k1 = [p, M1, theta, 0]
      k2 = [p, M1, pi - theta, pi]
                                                                                                                                                                                                                  Python
      pe = hep.fourvec(p1)
      ka = hep.fourvec(k1)
      ka
                                                                                                                                                                                                                  Python
   \begin{bmatrix} p \\ 2\sqrt{-M_1^2+p^2}\sin\left(rac{	heta}{2}
ight)\cos\left(rac{	heta}{2}
ight) \ 0 \\ \sqrt{-M_1^2+p^2}\left(-\sin^2\left(rac{	heta}{2}
ight)+\cos^2\left(rac{	heta}{2}
ight)
ight) \end{bmatrix}
```

Marcio Mateus Jr.

## SymPy – Exemplo $e^+e^- \rightarrow W^+W^-$



**:=**)



```
qdag = hep.dag(ka)-hep.dag(pe)
                 qdag
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           Python
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        -\sqrt{-M_1^2+p^2}\left(-\sin^2\left(rac{	heta}{2}
ight)+\cos^2\left(rac{	heta}{2}
ight)
ight)+\sqrt{-m^2+p^2}
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      -2\sqrt{-M_1^2+p^2}\sin\left(\frac{\theta}{2}\right)\cos\left(\frac{\theta}{2}\right)
      \sqrt{-M_1^2+p^2} \left(-\sin^2\left(rac{	heta}{2}
ight)+\cos^2\left(rac{	heta}{2}
ight)
ight)-\sqrt{-m^2+p^2} \qquad 2\sqrt{-M_1^2+p^2}\sin\left(rac{	heta}{2}
ight)\cos\left(rac{	heta}{2}
ight) = \sqrt{-m^2+p^2} \ 2\sqrt{-M_1^2+p^2}\sin\left(rac{	heta}{2}
ight)\cos\left(rac{	heta}{2}
ight) = \sqrt{-m^2+p^2} \ -\sqrt{-M_1^2+p^2}\left(-\sin^2\left(rac{	heta}{2}
ight)+\cos^2\left(rac{	heta}{2}
ight)
ight)+\sqrt{-m^2+p^2} \ -\sqrt{-m^2+p^2} \ -\sqrt
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       0
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             eps1dag = hep.dag(hep.polbar(k1, 0));
                 eps1dag

√ 0.1s

                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           Python
'0 pol Matrix([[sqrt(-M1**2 + p**2)/M1], [p*sin(theta)/M1], [0], [p*cos(theta)/M1]])'
                eps2dag = hep.dag(hep.polbar(k2, 0));
                 eps2dag

√ 0.1s

                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           Python
'0 pol Matrix([[sqrt(-M1**2 + p**2)/M1], [-p*sin(theta)/M1], [0], [-p*cos(theta)/M1]])'
                   \frac{p \sin(\theta)}{M_1}
```

## SymPy – Exemplo $e^+e^- \rightarrow W^+W^-$



### Forma simbólica (objeto: SymPy)

#### Formato LaTeX

```
print(sp.latex(T1v1[0]))

| ✓ 0.0s | Python
| One | O
```

### Código fonte para função Python

2\*p\*\*2\*sqrt(-m\*\*2 + p\*\*2)\*sin(theta)\*\*2)/M1\*\*2]])

```
import inspect
    func = sp.lambdify(tuple(T1v1.free_symbols), T1v1, 'numpy')
    print(inspect.getsource(func))

v 0.0s

Python

def _lambdifygenerated(theta, m, p, M1):
    return array([[m*(2*M1**2*sqrt(-M1**2 + p**2)*sin((1/2)*theta)**2 - M1**2*sqrt(-M1**2 + p**2) + M1**2*sqrt(-m**2 + p**2) - 4*p**2*sqrt(-M1**2 + p**2)*sin((1/2)*theta)**2*sin(theta)**2 - 2*p**2*sqrt(-M1**2 + p**2)*sin(theta)**2*sqrt(-M1**2 + p**2)*sin(theta)**2 - 2*p**2*sqrt(-M1**2 + p**2)*sin(theta)**2*cos(theta) + 2*p**2*sqrt(-M1**2 + p**2)*sin(theta)**2 -
```

Marcio Mateus Jr.

PPGFís | IF-UFRGS

## Prós e contras - SymPy







- ► Open Source (!)
- ► Clareza no código
- ➤ Documentação abrangente e atualizada
- ▶ Diferentes métodos para manipulação simbólica (simplify, expand, collect, solve, subs...)

- ► Pouco utilizado pela comunidade HEP;
- Exportação de modelos para outros softwares (MadGraph, Pythia, etc);
- Validação de resultados pode ser desafiadora;

### Resultados

UFRGS
UNIVERSIDADE FEDERAL
DO NO GRANDE DO SUI

► (Muitos) Arquivos em .pkl para salvar os resultados da computação...

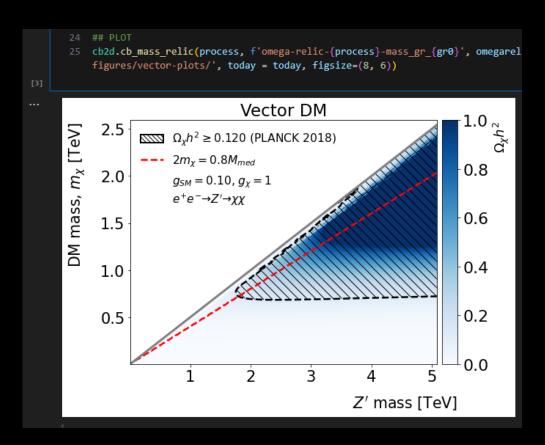
myruns 🏚 2023-10-18-omegarelic\_mass\_qq\_bin10\_nfermions6.pkl 2023-10-18-omegarelic\_mass\_qq\_bin70\_nfermions6.pkl 🏚 2023-10-18-omegarelic\_mass\_qq\_bin100\_nfermions6.pkl 🏚 2023-10-18-sigmaz\_qq\_mass\_ISR\_bin100\_Nfermions18\_v4.pkl 🏚 2023-10-23-omegarelic\_mass\_qq\_bin70\_nfermions6.pkl 2023-10-28\_omegarelic\_coup\_ee\_Fermion\_binsize\_70.pkl 2023-10-28\_omegarelic\_coup\_ee\_Scalar\_binsize\_70.pkl 🌻 2023-10-28 omegarelic coup ee Vector binsize 70.pkl 🌻 2023-10-28 omegarelic coup gg Fermion binsize 70.pkl 🌻 2023-10-28 omegarelic coup gg Scalar binsize 70.pkl 🌻 2023-10-28 omegarelic coup gg Vector binsize 70.pkl 2023-10-28\_sigmaz\_coup\_ee\_Fermion\_binsize\_100.pkl 🦃 2023-10-28\_sigmaz\_coup\_ee\_Scalar\_binsize\_100.pkl 2023-10-28\_sigmaz\_coup\_ee\_Vector\_binsize\_100.pkl 2023-10-28\_sigmaz\_coup\_qq\_Fermion\_binsize\_100.pkl 🌹 2023-10-28\_sigmaz\_coup\_qq\_Scalar\_binsize\_100.pkl 2023-10-28\_sigmaz\_coup\_gq\_Vector\_binsize\_100.pkl 🏚 2023-12-18\_omegarelic\_mass\_ee\_Scalar\_binsize\_10.pkl 🌻 2023-12-18\_omegarelic\_mass\_qq\_Fermion\_binsize\_10.pkl 🏚 2023-12-18\_omegarelic\_mass\_gq\_Fermion\_binsize\_70.pkl 2023-12-18\_omegarelic\_mass\_qq\_Scalar\_binsize\_70.pkl 🏚 2023-12-18\_omegarelic\_mass\_qq\_Vector\_binsize\_70.pkl 藦 2023-12-18\_supernaive\_omegarelic\_mass\_ee\_Fermion\_binsize\_100.pkl 🌹 2023-12-18\_supernaive\_omegarelic\_mass\_ee\_Scalar\_binsize\_100.pkl 🌹 2023-12-18\_supernaive\_omegarelic\_mass\_ee\_Vector\_binsize\_100.pkl 🌹 2023-12-18\_supernaive\_omegarelic\_mass\_qq\_Fermion\_binsize\_70.pkl 2023-12-18\_supernaive\_omegarelic\_mass\_qq\_Scalar\_binsize\_70.pkl 2023-12-18\_supernaive\_omegarelic\_mass\_qq\_Vector\_binsize\_70.pkl 🌹 2023-12-28\_naive\_omegarelic\_Mmed\_coup\_gr\_qq\_Fermion\_binsize\_50.pkl

藦 2023-12-28\_naive\_omegarelic\_Mmed\_coup\_gr\_qq\_Fermion\_binsize\_100.pkl

► ... E destes arquivos, os diversos plots que foram para o trabalho





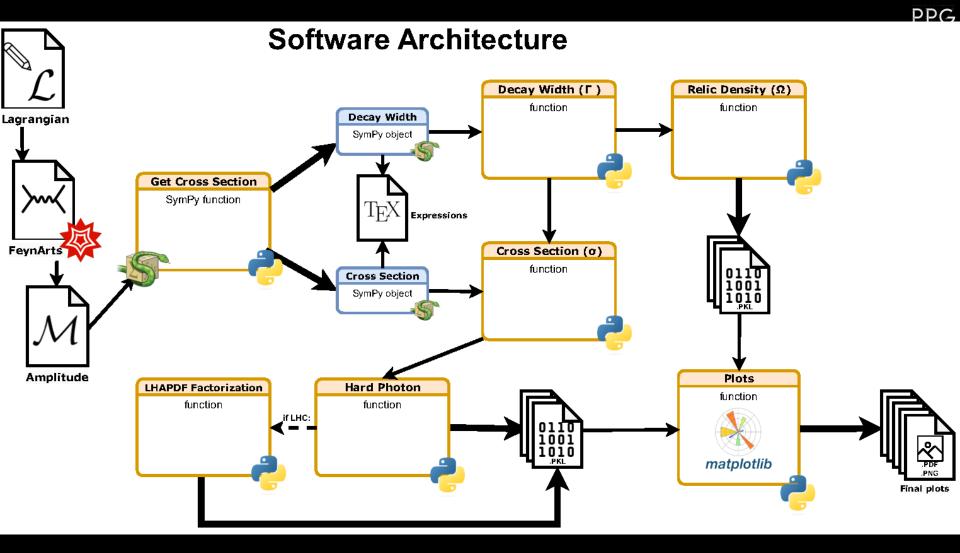


Marcio Mateus Jr. PPGFís | IF-UFRGS

# Arquitetura do Software aplicado







### **Links úteis**

Página do Prof. Berger com as funções para HEP originais



#### Particle physics using Python/Sympy

The book often cites MAPLE routines and encourages the reader to use MAPLE for own calculations. In my home university students had access to MAPLE and could also buy it at a low rate. This is not true anymore. Fortunately there is a noncommercial alternative: Python/Sympy. The very popular language Python is part of the curriculum at many universities. Sympy is a computer algebra programme based on Python. Used with the jupyter notebook it provides an elegant frame for doing computer algebra. Please consult the Python and Jupyter pages for installing these programmes on your system.



https://profchristophberger.com/lehrbuch-elementarteilchenphysik/python/

(menos útil) Meu GitHub



mardesoushi / resonant-dm-relic

https://github.com/mardesoushi/resonant-dm-relic

Marcio Mateus Jr. PPGFís | IF-UFRGS