NTNU
Institutt for fysikk

Jupyter notebook for symbolic calculations in Particle Physics

Prosjekt Schrödinger

Martin Kjøllesdal Johnsrud
Advisor: Michael Kachelriess

May 9, 2021

Abstract

This project aims at creating an introductory resource to computer aided symbolic calculations for students in particle physics. The resource consists of Jupyter Notebook which shows examples of such calculation. Due to shortcomings of the libraries used, the end product is not up to the desired standard. It gives a brief introduction to symbolic calculations using SymPy, and a limited example of a particle physics calculation. The report concludes by discussing the shortcomings of the approach used, as well as some possible fixes if the goal is pursued further.

Introduction

Calculations of Feynman diagrams in particle physics can be long and tiresome, and computer aided symbolic calculations are therefore an important reason. This project will focus specifically on calculations involving so called gamma matrices, which shows up in quantum electrodynamics [1]. This project aims to use Jupyter Notebooks, a solution for combing text, equations and code, with Python and SymPy to make educational resources introducing students to this way of doing calculations. SymPy[1] is a library for Python which enables creation and manipulation of symbolic expressions, such as integration, differentiation, linear algebra and more.

Method

Firstly, many students have no familiarity with SymPy, so a short introductory notebook was made. This shows some of the functionality of the library, what it may be used for, and introduces what is needed to know before continuing. This can, however, be a resource for other than just those who take particle physics, as SymPy is a powerful tool which can be used for a large variety of calculations. The plan was to use the SymPy high energy physics (hep) package.[2], which is made to handle gamma matrices. This is an abstract implementation of the gamma matrices, meaning that the objects are not represented as explicit matrices, but rather only by their algebraic rules, utilizing SymPy's tensor-objects. The implementation mirrors the notation $p_\mu\gamma^\mu$, where one does not have to write the objects explicitly as $p_0\gamma^0 + p_1\gamma^1 + ...$ This is an

[1]`https://www.sympy.org/en/index.html`
[2]`https://docs.sympy.org/latest/modules/physics/hep/index.html`

important goal, as it allows for focus on the physics and less messy algebra. However, the SymPy package is broken, and not up to the task. It does not handle any more complication than the simplest expression of the form $\mathrm{Tr}(\gamma^\mu ... \gamma^\nu)$. More complicated expressions, for example $\mathrm{Tr}(\gamma^\mu(m + p_\nu \gamma^\nu))$, break the trace function. This renders it more or less useless. The package seems to be a more or less abandoned project, and is not suitable for the project.

It is also possible to do calculations with an explicit matrix representation. This is done, and used to calculate the amplitude of Mott scattering. The explicit nature of the objects used in the calculation makes it necessary to choose a reference frame to get readable results. It does not allow for Lorentz-invariant expressions, which was one of the goals of this project. A last effort was made by using the Python library galgebra. This implements Geometric algebras, of which the gamma-matrix algebra is one. This package does not use an explicit representation of the matrices, however it does not allow for implicit sums. This leaves the result just as unreadable, without possibility of a compact, Lorentz-invariant expression.

Results and discussion

The final result of this project is not quite what we hoped for at the beginning. The SymPy hep module is severely lacking, and we were not able to find a suitable replacement. It is not without any result, however. A short notebook with introduction to some important features of SymPy, as well as a notebook showcasing the calculation of a Mott-scattering has been made. These can be valuable resources in both teaching particle physics or computer aided calculation, though they are not exactly what was the goal. The experience gained here will also hopefully be of use if more work is done towards developing a resource for computational symbolic calculations in particle physics, or elsewhere.

Experience and further work

The biggest insight gained in this project is probably how underdeveloped part of SymPy is. The hep module lacks crucial features as an implementation of the γ^5-matrix, an ability to handle expression with combinations of dummy indices and scalars, as for example $\gamma^\mu p_\mu - m$, or a ability for good printing of expressions. A GitHub issue has been submitted (`https://github.com/sympy/sympy/issues/13636`), though it seems unlikely to be solved as the module seems abandoned. The gamma matrices can be described as a more abstract mathematical object than matrices. They are members of Clifford algebra (or geometric algebra). The manipulation of gamma-matrices can therefore be implemented by libraries that deal with these. Galgebra[3] is a library with symbolic implementations of geometric algebra. It is possible to implement at least part of the gamma-matrix algebra, as shown in one of the notebooks, however it is not satisfactory. This does however show that libraries that deal with geometric algebras or Clifford algebras are possible candidates to use in further development.

If this project is to be taken up again, I see some possible paths for inquiry. SageMath is a powerful CAS based on Python, and may have the capabilities for implementing gamma matrices. It can also be used in Jupyter notebooks. Web page: `https://www.sagemath.org/`. Cadabra has implemented Clifford-algebras, and might be an alternative. It does, however, have sparse documentation. Web page: `https://cadabra.science/`

Links

GitHub repo, including all notebooks made in this project:
`https://github.com/martkjoh/Notebook-FY3403-Particle-Physics`

[3]`https://galgebra.readthedocs.io/en/latest/`

References

[1] David Griffiths. *Introduction to Elementary Particles*. 2nd ed. 2012.