ALHEP: new symbolic algebra program for high energy physics

V. Makarenko*
NC PHEP BSU
153 Bogdanovicha str., 220040 Minsk, Belarus

August, 2005

Abstract

The new symbolic algebra program for high energy physics is presented. It is basically designed for loop diagrams analysis and deals with renormalization, dimensional regularization and tensor reduction for loop integrals.

1 Introduction

The analytical calculations in high energy physics are mostly impossible without a powerful computing tool. The big variety of packages is commonly used. Some are general-purpose symbolic algebra programs with specific HEP-related plug-ins (Mathematica, REDUCE), some are designed especially for particle physics (CompHEP [1], SANC [2], GRACE [3] etc.) and some are created for specific interaction class or specific task.

Many of them uses external symbolic algebra core (Form, Mathematica, REDUCE). Some deals with matrix elements squaring (FeynCalc [4]), some calculates helicity amplitudes directly (MadGraph [5], CompHEP [1], AlpGen [6], O'Mega [7]). Some provides also the numerical calculations, some requires external Monte-Carlo generator to be linked. Some programs provides also the one-loop calculation routines (FormCalc [8], GRACE [3],

*E-mail:makarenko@hep.by

SANC [2]). Nevertheless there is no uniform program that meets all the user requirements.

Every calculation requires the program-independent check. Hence the optimal tactics is the simultaneous usage of two (or more) different symbolic algebra packages.

ALHEP is symbolic algebra program for performing the way from Standard Model Lagrangian to the squared matrix element for the specified process in particle physics. It is basically designed for the loop diagram analysis including tensor reduction of loop integrals, dimensional regularization and renormalization.

ALHEP website **http://www.hep.by/alhep** contains the up-to-date executables (for both Linux & Win32 platforms), manual and script examples.

2 Program Structure

The ALHEP program internal structure can be outlined as follows:

- The native symbolic algebra core.
- Common algebra libraries

  Dirac matrices, tensor and spinor algebra, field operators & particle wave functions zoo.
- Specific HEP functions and libraries

  Feynman diagrams generation, trace calculation, HEP-specific simplification procedures, tensor integrals reduction and others.
- Interfaces to Mathematica, FORTRAN, TeX and internal I/O format.

  FORTRAN code is used for further numerical analysis. Mathematica code can be used for viewing any symbolic expression in program. But no backward interface from Mathematica is currently implemented. TeX output can be generated for Feynman diagrams view. Internal I/O format is implemented for the most of symbolic expressions, allowing save & restore calculation at intermediate steps.
- Command script processor.

User interface is implemented in terms of control script. ALHEP should be launched with the single argument: script file name to be invoked. The ALHEP script language have the C-like syntax, variables, arithmetic operations, function calls. All HEP-related tasks are implemented as build-in functions.

2.1 Calculation scheme

Here we discuss the basic steps for the usual calculation in high energy physics. The following steps should be implemented in control script before program launching. The example scripts containing all the described steps can be found at ALHEP website.

Initialization section

Declaration of process *kinematics*, initial and final-state particles, titles for particle momenta & polarization vectors.

Physics model definition. The part of SM Hamiltonian should be specified (QED, neutral and charged current weak interactions, ghost, counterterms etc.). The shorter Hamiltonian is selected, the faster is Wick theorem application. Therefore if it is obvious that no ghost diagram will appear in process, it is recommended to exclude the ghost part of Hamiltonian manually.

Polarization declaration. Every particle is considered as polarized with abstract polarization vector by default. The specific helicity value can be set manually or particle can be marked as unpolarized.

Setting *mass-order* rules for specific particle kind. One can demand the massless calculation for light particle, that greatly saves evaluation time. One can also demand keeping particle mass with specific order m^n and drop out the higher-order expressions like m^{n+1}. It allows to consider the leading mass contribution without calculating precisely.

Diagrams generation

Feynman diagrams are generated using the *Wick theorem*. It may take much time for long Hamiltonian due to n!-complexity of the With theorem algorithm.

There is also a *fast* diagram generation method, based on the common topology principles, but it may lead to erroneous results (is some cases) and may be used for debugging purposes only.

One may *draw diagrams* here, halt the program and check out if diagrams are generated correctly. The diagrams are drawing in terms of

AxoDraw LaTeX package.

After the diagram set is generated one may *cut-off* not interesting diagrams to work with the shorter set or *select* the single diagram. The loop corrections are calculating faster when processed by single diagrams.

The *matrix element* should be retrieved from diagrams set.

Before any operation with loop matrix element one should declare the $N-$*dimensional space*. Although dimensional regularization will be turned on automatically in squaring procedure, but the previous number of dimension will be restored after squaring is completed. Therefore is is required to turn dimensional regularization on manually for all the other operations.

Matrix element squaring (coupling to other)

The squaring procedure is controlled by plenty of options, intended mostly for the performance tuning and debugging.

It basically includes *reduction of gamma-matrices sequences* coupled to kinematically dependent vectors. It reduces number of matrices in every product to minimum.

Than the item-by-item *squaring* is followed.

The integration over virtual particles phase space is than involved. Here all the loop integrals appear.

One can specify the *way of photon polarization implication.* Photon helicity vectors can be inserted in vector ε_μ or gamma-coupled form $\hat{\varepsilon}$, the hat-form can be both chiral and precise one, the vector form can be involved either before or after the squaring procedure. The calculation using the chiral $\hat{\varepsilon}$ form is usually fastest, but the precise form ε_μ is required for some processes.

One may declare the *c.m.s. analysis*, that involves another relations for abstract polarization vectors $(p_1.\varepsilon_2 \to 0, p_2.\varepsilon_1 \to 0)$ that simplify results for polarized particles and hasten squaring procedure.

Loop integrals evaluation (for loop corrections calculation)

The *tensor virtual integrals* are reduced to scalar ones using two methods.

If tensor virtual integral is coupled to external momentum $I_\mu p^\mu$, and p-momentum may be decomposed by integral vector parameters, the *fast reduction* is involved. The D_x integrals for $2 \to 2$ process contain the whole basis of 4-dimension space and the coupling to any external momentum can be decomposed. It works well if all the polarization vectors are composed in terms of external momenta.

The *common tensor reduction scheme* is involved elsewhere. Tensor integrals are decomposed by the vector basis like

$$I_{\mu\nu}p, q = Ip^{\mu}q^{\nu}, p^{\nu}q^{\mu}, p^{(\mu}q^{\nu)}, g^{\mu\nu}.$$

The linear system for scalar coefficients is composed and solved. Implemented for B_j and $C_{i,ij}$ integrals only. The D_{xxx} - integrals can be usually reduced using the first algorithm.

For *scalar loop integrals* the tabulated values are used. There is no reason to tabulate non-divergent integral with complicated structure. Hence scalar integrals table contain the A_0 and B_0 integrals with different mass configuration. It also contains a useful D_0 chiral decomposition. Other integrals should to be resolved using LoopTools-like [9] numerical programs.

Renormalization

The renormalization procedure is implemented via the counter-terms (CT) part of SM Lagrangian. The additional CT diagrams are generated and calculated in terms of abstract renormalization constants (δ_m, δ_f etc.). The values of CT constants are tabulated (the minimal on-shell scheme is used) and may be inserted after the CT diagrams are calculated.

Note: the counter-term part is currently implemented for QED only. The weak diagrams should still be renormalized manually. The future version of ALHEP program will definitely include the complete renormalization scheme.

Simplification

The *kinematic simplification* procedure is available. It reduces expression using all the possible kinematics relations between momenta and invariants. The minimization of +/* operations in huge expressions can also be performed.

FORTRAN procedure creation for numerical analysis

F90-syntax for generated procedures is used. Generated code can be linked to any Monte-Carlo generator for numerical analysis.

3 Conclusions

The new program for symbolic computations in High Energy Physics is presented. In spite of several restrictions remained in current version, it can be useful for precision analysis of observables in particle collision experiments. It basically concerns calculations of loop diagrams and mass dependencies of leptonic processes. Examples of loop diagrams computation can be found at program website.

The nearest projects are:

- Fast amplitude calculation for multiparticle production.
- Complete renormalization scheme for SM.

The future ALHEP development will cover the complete covariant analysis of the one-loop radiative corrections including the hard bremsstrahlung scattering contribution.

Visit ALHEP project website **http://www.hep.by/alhep** for program updates.

References

[1] CompHEP, http://theory.sinp.msu.ru/comphep, E. Boos et al., Nucl.Instrum.Meth. **A534** (2004) 250, hep-ph/0403113.

[2] SANC, http://sanc.jinr.ru, A. Andonov et al., hep-ph/0411186, to appear in Comp.Phys.Comm.

[3] GRACE, http://minami-home.kek.jp, J. Fujimoto et al., Comput. Phys. Commun. **153** (2003) 106, hep-ph/0208036.

[4] FeynCalc, http://www.feyncalc.org, R. Mertig, M. Bohm, A. Denner, Comput.Phys.Commun. **64** (1991) 345.

[5] MadGraph, http://madgraph.hep.uiuc.edu, T. Stelzer, W. F. Long, Comput. Phys. Commun. **81** (1994) 357, hep-ph/9401258.

[6] AlpGen, http://mlm.home.cern.ch/mlm/alpgen, M. Mangano et al., CERN-TH-2002-129, JHEP **0307** (2003) 001, hep-ph/0206293.

[7] O'Mega, http://theorie.physik.uni-wuerzburg.de/˜ohl/omega, M. Moretti, T. Ohl, J. Reuter, LC-TOOL-2001-040, hep-ph/0102195.

[8] FormCalc, http://www.feynarts.de/formcalc T. Hahn, M. Perez-Victoria, Comput.Phys.Commun. **118** (1999) 153, hep-ph/9807565.

[9] LoopTools, http://www.feynarts.de/looptools, T. Hahn, M. Perez-Victoria, Comput.Phys.Commun. **118** (1999) 153, hep-ph/9807565.