The SANC project status and plans.

Evgeny Uglov – *Dzhelepov Laboratory for Nuclear Problems, JINR, Dubna, Russia*

Abstract: The main goal of the SANC project is the creation of a computing system for automatic computation of pseudo and realistic observables with one-loop precision for various processes of elementary particle interactions. The purpose of this paper is to outline the SANC project status and next stage plans — SANC2.

The SANC project status and plans.

A.D. Andonov[1], V. A. Kolesnikov[2], Evgeny Uglov[2],
[1] Quanterall Ltd., Varna, Bulgaria
[2] Dzhelepov Laboratory for Nuclear Problems, JINR, Dubna, Russia
e-mails: andonov@nu.jinr.ru, kolesnik@nu.jinr.ru, corner@nu.jinr.ru

Abstract

The main goal of the SANC *project is the creation of a computing system for automatic computation of pseudo and realistic observables with one-loop precision for various processes of elementary particle interactions. The purpose of this paper is to outline the* SANC *project status and next stage plans —* SANC2*.*

1 Introduction

There are a lot of dedicated software aimed at comparison of near future experimental data at LHC with the SM theoretical predictions. We mention, first of all, well known automatic computer system CompHEP [1], which is able to compute any HEP process involving up to 9 particles, however in the LO only. So, one should not expect precision of the predictions better than 10%. There are several computer systems working in the NLO approximation, like FeynArts [2] and GRACE [3]. They are able to compute many HEP processes up to 5 particles involved in NLO over EW and QCD forces. FeynArts may compute only virtual contributions and may not real ones. Besides above mentioned general use systems, there are a lot of software, which should be ranked as "codes" rather than "systems", like PYTHIA [4], MC@NLO [5], HERWIG [6] and many others. In the years 2002 – 2007 an attempt was undertaken to create an easy to use NLO system SANC (a client server data based system) which presently includes many SM processes up to 4 particles and also some background for subsequent inclusion of 5 particle processes. It is accessible from the Internet sites: http://sanc.jinr.ru, http://pcphsanc.cern.ch.

The SANC2 is the system that has to be built to replace the former. The purpose of this paper is to outline status and SANC2 plans.

2 The SANC system

The system SANC [7] is a HEP tool for precision calculations of processes at high energy colliders and their applications. SANC realizes the full chain of automatic calculations "from the Standard Model Lagrangian to the event distributions" up to 4 particle processes [8]. It roots back to two main sources: the reach experience obtained by the Dubna group in the computer support of HEP experiments (codes like TERAD, HECTOR, MUELA, ZFITTER, GENTLE) and the book [9] which summarized the efforts of many groups of theorists in the field of precision tests of the SM.

The main idea of the SANC project was to develop an integrated computer environment for creation of Monte Carlo (MC) event generators in the NLO approximation starting from a unique analytic platform, passing stage of automatic generation of codes for subsequent numerical calculations and eventual creation of the MC generator. In the current version V.1.10 of SANC we realized this idea only partly. An external user may interactively come through all chain of relevant analytic calculations, generation of the so called Standard SANC FORTRAN Modules (SSFM) [10] and their export and a subsequent use in his/her own codes. Moreover, from our project pages already several MC event generators may be downloaded.

The underlying technology of the entire SANC system is JAVA, while the analytical computations are implemented in FORM, the numerical in FORTRAN. Also, there is a PERL module s2n generating FORTRAN programs and providing transmission from analytical results to numerical applications.

The SANC server-tier has a database backend for storing different SANC programs and functionality for compiling them as well as for linking the various modules. The SANC client-tier is represented by IDE clients which allow creating, editing and compiling of FORM and FORTRAN programs, as well as passing parameters to charts to display any numerical results.

In the course of time this system proved to have a lot of shortcomings rooted in the choice of the applied technologies, the architecture itself and the processing mechanisms, namely: the main issue arises from the fact that the structure of the computer system was related to a particular physics context. The introduction of new and more complex HEP processes led to the incredible complication of the system.

FORM, which was applied for analytical computations, has no equal match in the world of computer algebra systems for intensive computations; this is the reason why it has become a de facto standard in the field of HEP. However it has some peculiarities which hinder the creation of an integrated system of FORM procedures.

The s2n module which connects the analytical and numeric computations is not generic and requires manually entering every new HEP process. Complicated IDE scenario which makes it impractical to be used by students or for experiments. The web services technology, which was performing the communication between the client and the server tier, although a standard for enterprise applications, is not suitable for transmitting typical SANC objects.

So, the SANC present status is far from being satisfactory for the following reasons:

- An automatic generation of process amplitude is lacking;
- Although an advanced analytic platform, based on Passarino–Veltman reduction techniques exists, it is limited to 4 particle processes and is not sufficient to consider 5 particle processes;
- Even in the class of 4 leg processes, pre computation of some box diagrams consumes a lot of time;
- There are certain problems in numerical calculations: for evaluation of the contribution of virtual corrections in many cases Re*16 floating point computation is unavoidable;
- MC generation takes a lot of time;

- Parallel numerical calculations are mandatory.

This change is the aim of the SANC2, which started at the beginning of 2007.

3 SANC2 plans

Every SANC2 process can contain the following components: analytic; symbolic to numeric; numeric; graphic and doc. The SANC2 will inherit all algorithms from SANC. But these algorithms will be wrapped in more flexible methods and classes. Therefore this framework will became more scalable. The entire algorithm for solving a specific problem, using external libraries and transition from one part of the program to another is contained in the program itself and under the control of the SANC2 user.

The base programming language of the SANC2 programs is RUBY, which was selected for the following reasons:

- RUBY is a fully object-oriented programming language, highly dynamic and extensible.
- The existence of RubyInline project[1], which allows direct inclusion of C/C++ and FORTRAN programs in the code of a RUBY program. By analogy to this in May 2007 was created FormInline which is a FORM implementation of RubyInline. Similarly in the code of a RUBY program can be directly included any FORM program.
- The existence of Ruby/CERNLIB - a collection of Ruby extension libraries to access various CERN Program Library, such as HBOOK, HIGZ, MATHLIB and ruby-root which provides RUBY bindings for the ROOT Object Oriented Framework[2]

RUBY makes it possible to increase work effectiveness by shortening the development cycle needed for solving a particular coding problem. On the other hand using FormInline enabled SANC developers, who are not interested in studying RUBY to continue coding in FORM and FORTRAN, while seamlessly integrating their code into SANC2.

The basic structure of SANC2 should contain following parts:

- Already mentioned FormInline is a RUBY wrapper for algebraic objects of FORM language. It gives a possibility to operate with FORM algebraic expressions with RUBY and so have all benefits of high level programming language in algebraic computations.
- For the fully automatic system we need to develop a generator of Feynman diagrams.
- Next step is building a fully automated system for all steps of algebraic process computation. This include wrapping of SANC procedures in RUBY shell and complementing them with new algorithms.

[1] http://rubyforge.org/projects/rubyinline/

[2] http://www-ps.kek.jp/thitoshi/ruby/cern/index.html, http://www.csd.uoc.gr/~elathan/rr/

- Than the algebraic results should be converted to code for numeric computations. So the analog of s2n should be developed avoiding all it's shortcomings. The output of this software should satisfy all requirements to be used as standalone code, so to be used as part of MC generators of other projects.
- The general framework for automatic building of MC generators on the basis of SSFM analog from the previous stage should be developed.

The `SANC` group have already completed computations for many physical processes. These programs are stored in data base of current `SANC` framework. For backward compatibility these programs should be modified to be executable in the `SANC2` framework. These programs are good for testing purposes.

4 Conclusions

The main difference between the `SANC` and the `SANC2` is the fact that the entire algorithm for computing a particular HEP process — from the analytical, through the numerical computations, and, finally to the charts — will be implemented in the cycle of a single `SANC2` program totally controllable by the user. The usage of RUBY makes it possible to create plenty of generic, universally usable libraries and building blocks for one-loop processes.

The authors are grateful to all `SANC` group members.

Bibliography

[1] *Pukhov A. et al.* CompHEP: A package for evaluation of Feynman diagrams and integration over multiparticle phase space. Users manual for version 33 // [hep-ph/9908288].

[2] *Hahn T.* Feynman diagram calculations with FeynArts, FormCalc, and LoopTools // [hep-ph/1006.2231].

[3] *Belanger G. et al.* GRACE at ONE-LOOP: Automatic calculation of 1-loop diagrams in the electroweak theory with gauge parameter independence checks // Phys. Rept. 2006. V. 430. P. 117-209. [hep-ph/0308080].

[4] *Sjostrand T., Lonnblad L., Mrenna S.* PYTHIA 6.2: Physics and manual // [hep-ph/0108264].

[5] *Frixione S., Webber B.R.* Matching NLO QCD computations and parton shower simulations // JHEP. 2002. 0206. P. 029. [hep-ph/0204244].

[6] *Corcella G. et al.* HERWIG 6.5 // JHEP. 2001. 0101. P. 010. [hep-ph/0011363]; [hep-ph/0210213].

[7] *Andonov A. et al.* SANCscope-v.1.00 // CPC. 2006. 174. P. 481-517.

[8] *Bardin D. et al.* SANCnews: Sector ffbb // CPC. 2007. 177. P. 738-756. [arXiv:hep-ph/0506120].

[9] *Bardin D., Passarino G.* The Standard Model in the making: Precision study of the electroweak interactions // Oxford University Press, UK: Clarendon. 1999.

[10] *Andonov A. et al.* Standard SANC modules // CPC. 2009. 181. P. 305-312.