PROCEEDINGS OF SCIENCE

Status of the FORM project

Irina Pushkina[*][†]
National Institute for Subatomic Physics (NIKHEF)
Science Park 105, 1098XG Amsterdam, The Netherlands
E-mail: `irinap@nikhef.nl`

We describe the recent developments of the symbolic manipulation language FORM and its versions ParFORM and TFORM. In combination with examples of their successful application, we discuss the new project of the FORM version of GRACE.

13th International Workshop on Advanced Computing and Analysis Techniques in Physics Research, ACAT2010
February 22-27, 2010
Jaipur, India

PoS(ACAT2010)071

[*] Speaker.

[†] A footnote may follow.

© Copyright owned by the author(s) under the terms of the Creative Commons Attribution-NonCommercial-ShareAlike Licence.

1. An overview of the recent developments

The development of FORM as a symbolic manipulation system started in 1984. The main idea that time was to create the efficient and reliable computer tool that could be fast and deal perfectly with mathematical expressions of gigantic sizes. In 1989 the first version was officially released. It found the claimed application in perturbative quantum field theory where one has to compute numerous Feynman diagrams. Since then there were two major upgrades to FORM 2 and FORM 3. The main benchmarks of the version 3.3 can be described in the following way

- The size of the expressions FORM is currently dealing with has entered the terabyte scale.
- The excellent memory management remains one of the main advantages of FORM usage over other modern general purpose systems such as Maple or Mathematica.
- The implementation of the rules for reduction or transformation of target expression is flexible and powerful.
- The distribution of the system is in the public domain and accessible for anybody. Current executables are available for LINUX/32-bits/Opteron, Apple/PPC/Intel and Windows (32-bits architecture).

In general, there are two groups of scientists working on FORM's development. One is at NIKHEF (the Netherlands) and another one is at Karlsruhe University (Germany). The main projects are oriented on the general development of FORM and TFORM (Jos Vermaseren, NIKHEF) and ParFORM (Misha Tentyukov, Karlsruhe University). Among those are making FORM the open source, seting up a failsafe system to prevent the loss of data during the unexpected computer failures (Jens Vollinga, NIKHEF), the incorporation of the polynomial and rational functions in the software (Jan Kuipers, NIKHEF), the code simplification to facilitate the FORM version of GRACE (Irina Pushkina, NIKHEF).

The new features of FORM as of March 2010:

- The store/save files written on one computer can be read on another computer with a different architecture, be that 32 or 64 bits.
- FORM (TFORM) can write special data on disk at the end a module. This data can be used to restart the FORM job at the same spot after a possible crash. Needless to say that this feature is very useful for the protection of long-running jobs in case of system failures. In principle, it can also be used for debugging purposes.
- Making FORM open source is in its final stage. The target date is summer 2010. This should insure the successful simultaneous work of a number of active developers, the repository access, the bug report management, the communication tools, documentation and source code review.
- The inner workings of FORM and information that would normally need to be explained by current developers are written down in a dedicated manual. It is supposed to replace, in principle, the introduction to the source code.

PoS(ACAT2010)071

- Currently CVS is used for the repository access. The web access to the repository (WebCVS) will be available in the future.
- The mailing lists for releases and developer coordination are under consideration. We also plan the use of a web forum where the internal developer issues can be discussed, as well as new or old users can ask any questions, make suggestions, or post tips and information for other people. It should become some kind of on-line FORM community.

The development of FORM is currently being done on mostly laptops and desktops with the Linux operational system. Among these are Dells with 8 (hyper)threads and an Apple computer with 2 cores.

1.1 ParFORM and TFORM

Besides the standard package of FORM, the development has touched its multi processor version ParFORM as well. It is beneficial to use it on clusters with more than two processors. It is able to handle almost all existing FORM programs. One of the main ParFORM disadvantages is MPI. Despite MPI being the industrial standard, various implementations are not binary compatible so ParFORM should be compiled with exactly the same MPI version which is installed on the computer[1].

There was some work done experimenting with communications links used for storage networking. The best results showed InfiniBand.

The checkpoint mechanism to protect long-running jobs against the system failures is one of the new features.

TFORM [2] is the multi-threaded version of FORM. It operates on multi core systems with shared memory. Its usage is beneficial already for two processors. It is worth mentioning that it can handle all existing FORM programs. The current development work involves tuning it to increasing number of cores.

2. GRACE/FORM/1-loop system

Side by side with the continuous development of the existing packages of FORM, there is a new project involving both FORM and the scientific software GRACE.

The standard models of elementary particles imply that we have definite Lagrangians and thus we can, in principle, predict any process using perturbation theory. The difficulty here is that the interaction Lagrangian is very complex. This becomes the technical difficulty. The illustrative example comes from the final four-body processes in the perturbative approach. If in the tree level the number of diagrams is around several tens, in the 1-loop level they are already a few thousand.

To overcome these difficulties, the general algorithm of the GRACE/FORM/1-loop system was developed. The code produces programs for the calculation of a cross section, or even the corresponding Monte-Carlo event generator.

[1] In contrast, TFORM requires no installation since it is just an executable file.

PoS(ACAT2010)071

- GRACE is the automatic program for the computation of Feynman diagrams. It performs a generation of the 1-loop and the tree diagrams as well as the construction of the product graphs.
- At the next step, another standard package written in C, grcform, processes the generated graphs through a series of symbolic operations. It performs the algebraic calculation of the product of a loop diagram with a tree diagram. There is a whole range of parameters which can be calculated at this stage: the shift of loop momentum, the trace calculation of sigma-matrices, the extraction of numerical coefficients for the loop integral and so on.
- At the last stage FORM writes the resulting formulas as Fortran routine for each set of tree and loop diagrams.

Typical diagrams here run in a few seconds, but the size of the output Fortran file might exceed the Gbyte range. It is worth mentioning that at the end of the day, each diagram will be a Fortran file which means that we desperately need to improve the quality of those codes. This is the work which is in progress right now at NIKHEF.

3. Example 1

The following example illustrates an effective FORM application. The multiple zeta values, also called non-alternating sums, were calculated for a case of a basis of the weight 28. The calculation was performed on one node of the SGI (Silicon Graphics Inc.) computer at Karlsruhe.

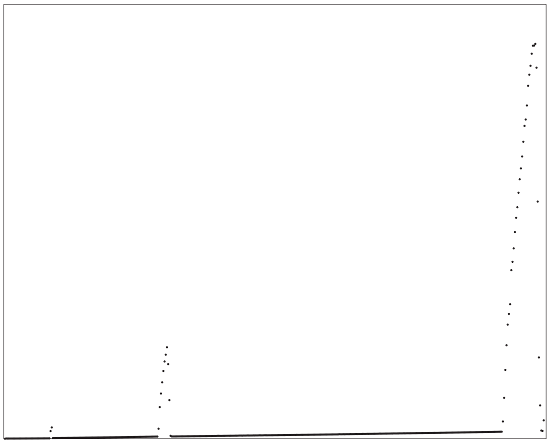

Figure 1: The size of the output data (the ordinate) corresponding to the multiple zeta values as a function of the number of the modules (the abscissa). The most prominent peack of the depth 8 corresponds to 100Gb.

On the graph you can see the peak of roughly 100Gb corresponding to the size of the output data of the depth 8. The initial system of 971.712 variables was expressed into the final minimal set of 92 variables. Within 69 days of constant running the program generated a huge amount of terms (28.700.000.000.000).

PoS(ACAT2010)071

4. Example 2

Another interesting example on the effective usage of ParFORM and TFORM comes from perturbative QCD [3] where the corrections of order α_s^4 (where α is the strong coupling constant of the standard model) were evaluated for the nonsinglet part of the cross section for electron-positron annihilation into hadron and for the decay rates of the Z boson and the τ lepton into hadron. In addition, the order α_s^4 contributions to the Bjorken sum rule for polarized electron-nucleon scattering and to the (non-singlet) Adler function were calculated for the case of a generic colour gauge group.

During 100 days there were generated 30000 5-loop diagrams by 2 clusters. In general, there were generated 10.000.000.000 terms on 1TB disk space.

References

[1] J.A.M. Vermaseren, *New features of FORM*, [arXiv:math-ph/0010025].

[2] J. Vermaseren, M. Tentyukov, *The multithreaded version of FORM*, [hep-ph/0702279].

[3] P.A. Baikov, K.G. Chetyrkin, *Adler function, Bjorken sum rule, and the Crewther relation to order α_s^4 in a general gauge theory*, [arXiv:1001.3606].

[4] G. Belanger et al., *Automatic calculations in high energy physics and Grace at one-loop*, **430** (2006) 006 [hep-th/2001033].
J. Fujimoto et al., *GRACE with FORM*, Nucl. Phys. Proc. Suppl.**160** (2006) 150.

PoS(ACAT2010)071