

YMAP ROOT 講習会

JAXA 宇宙科学研究所

水越 慧太

mizukoshi.keita@jaxa.jp

理研 仁科加速器科学研究センター

吉本 雅浩

masahiro.yoshimoto@riken.jp

2023 年 (令和 5 年) 2 月 20 日

1 概要

2022 年春、本年度で第 7 回目となるオンラインでの ROOT 講習会を宇宙素粒子若手の会 (YMAP) 主催にて開催しました。3 回にわたり講師を担当した水越と、過去 2 回の世話人を担当した吉本がこれまでの活動を総括して報告します。ご存知のように、ROOT は素粒子・宇宙物理・原子核分野での標準化されている唯一の解析フレームワークです¹。本講習会は、主に研究室配属 1 年目の学生 (4 年生, 修士 1 年生) を対象として、ROOT の使い方に習熟してもらい、スムーズに研究を行えるようになることを目指しています。したがって、実際の内容は ROOT テクニックにとどまらず、計算機の基礎や C++ の内容を含む、解析イントロダクションです。学部課程は宇宙素粒子原子核分野に特化したものではないため、物理解析を講義でやったことがないという学生がほとんどという現状があるのではないのでしょうか。このような教育を各研究室で独立に行うコストを減らしたいと思い実施しています。もともとは名古屋大学の奥村暁さんが名古屋大で行っていた講習会を若手の会でオンライン配信していましたが、2020 年度から若手の会の会員が講師を行ない独自に実施しています。本年度は 271 名の方に登録していただきました (指導者としての参加含む)。

類似した講習会として、粒子物理コンピューティングサマースクール、Geant4 初心者講習会が存在しますが、特定の実験の学生の育成や、ソフトウェアのフィードバックを得ることに主眼を置いたこれらの講習会に対して、広く業界の新人に汎用的な知識を身につけてもらうことを目標としています。

2 講義内容

ROOT 講習会というのは正確に実態を表しておらず、実際には不自由なく研究を進めるためのイントロダクションをめざしています。具体的には、答えが返ってきやすい質問のしかたに始まり、タッチタイピング練習、ターミナルとは何か、Unix のディレクトリ構造、パスとは何か、C++ や python の言語としての立ち位置などの、ROOT 以前の部分を丁寧に実施しています。その後、ROOT を使ってヒストグラム、グラフ、Tree、Fitting など一通り、実際に講習会中に手を動かして自力で動作させるところまでやってもらっています。今年度は全 6 回を実施しました。

第 0 回 水越慧太 CUI 環境, ROOT のインストール

第 1 回 水越慧太 解析環境, 解析のすすめかた

第 2 回 水越慧太 統計基礎, ヒストグラム

第 3 回 中村輝石 フィッティング, グラフ

第 4 回 高橋光成 TTree, 2D ヒストグラム

第 5 回 濱田悠斗 PyROOT

2021 年講師: 水越慧太, 細川佳志, 中村輝石

2020 年講師: 水越慧太, 吉本雅浩, 小原脩平

ROOT のビルドも自力でやってもらうことを推奨², サポートしています。Virtual Machine を配ってもいいのですが、それでは具体的に何が行われているのかという裏側が完全に隠蔽されてしまいます。また、ほとんどの場合実際にデータ解析を Virtual Machine 上で行うことはないでしょう。ROOT 自身も C++ をコンパイルして作成されているということを説明し、愛着のあるであろう自分のパソコン環境で ROOT をビルドしてもらっ

²ROOT の公式サイト¹の雰囲気を見る限り、package manager や CVMFS でインストールすることが主流で、もはや自力でのビルドは推奨されていないようでもあります。

¹少なくとも、TH1 と TTree に関して代わるものはないでしょう。

ています。こんな理想を掲げてしまったせいで、毎年冬になると、ROOT を Windows 10, 11 (WSL2), macOS (x84-64/ARM), Ubuntu, CentOS, arch linux の OS でビルドオプションを変えながらひたすらビルドしつづけて最新の version でのビルドガイドを作るという雑用が増えてしまいました。

現状では ROOT を題材にしています。python をベースに解析ツールが整備されている実験も増えてきていますが、未だ python には実験解析を行うのに必要なツール³が不足しており、工夫しなければ速度も非常に遅いです。python を使うことが不可能であるということはないと思いますが、straight-forward な⁴方法ではないと感じています。実際に python を使っている実験でも、誰かが裏で Fortran や C++ のライブラリを書いたり、各実験に特化した最適化コードを作製しているのではないのでしょうか。私たちが(おそらく業界としても)育てたいのは、その“裏”を見に行く人材⁵なので、C++/ROOT を入り口として紹介しています。

講習会では Slack を準備し、講習会以外の時間でも参加者が質問ができるようにしています。「コードの foo とか bar って何ですか?」とか「ターミナルでパスワード入力画面になると突然キーボードが壊れて何も映らなくなるんですけど?」のような、研究室の教授様には聞けないような基礎的な質問が大量にきます。本来は、研究室に配属し研究を始めてすぐに解決するようなことですが、往々にしてそうした質問はすぐには聞けないものです。しかし、このような質問こそ、新人にとってより身近な存在である学生やポスドクが運営している YMAP の ROOT 講習会でこそ聞きやすいはずです。Slack では受講者同士で質問を解決することもありました。また、マニュアル/チュートリアル/ユーザーガイドには往々にして当たり前のことは書いていないものです。この文化、思想みたいなものは、旧来先輩から後輩に脈々と継承されてきたものだと思いますが、一旦断絶してしまうと再度一から取り戻すのは大変です。ROOT 講習会の枠組みでそれらを伝えていくことができることは有意義だと考えています。

講師は、参加者の顔が見えない状態で約 90 分カメラに向かって喋ります。このコロナ禍で、多くの教員は経験されていると思いますが、反応が分からないまま喋り

³例えば、“まともな” Fitting ツールや、データ構造など。物理に興味のある学生が、実験結果を出すためにデータ構造とアルゴリズム、統計を全部勉強して自力で実装ができるようになるまで結果を待てる状況はほとんどないと思われます。

⁴コンパイラを信じて愚直なコードを書く

⁵コンパイラに最適化を丸投げするのは畢竟本質を理解する人材ではないのではないかという批判もあると思います。実験の現場では必要に応じて生成されるアセンブリコードを見て最適化の検証を行っているはずですが。しかし、最初からそのハードルを求めるのは不可能です。余談ですがコンパイラの最適化は高度に発達しているだけでなく、頻繁に人間に良いコードを書かせようと命令してくるので、巷に溢れる定義不明の AI よりは遙かに知性を感じます。

続けるのは大変な苦痛です⁶。一応、講習会の最後にアンケートを取り、講習内容の理解度を記録し、講義のレベルが適切だったかのフィードバックを受け取り、次回以降に役立てています。

講習会日時に講義などがあり、出席できなかった人のために、講習会の動画は Youtube で登録者のみに限定して公開しております。アクセス解析をみると、明らかに登録者数を超える回数視聴されており、復習に利用されているようです。スライドに重ためのコードを載せている箇所動画を止めて写経をしている人も多いようなので、これからも継続していきたいと考えております。また、10月、12月ごろに再度視聴数が伸びる傾向が若干あり、5月に何となくやったけど忘れてしまったことを、実際に自分たちで実験し始める頃にもう一度復習しているのではないかと思います。正直なところ、私は動画より文章があった方が効率よく情報を入手できると考えているのですが、若い世代では動画で学ぶことに親和性がある人の割合が増えているようです。資料を頑張って作り込んでいるのですが、むしろ動画作りを頑張った方がいいのかもしれない。

主として自学自習に任せていますが、唯一宿題として、自由研究の提出を課しています。具体的には、物理に限定せず、自身でテーマを設定し、自力でデータを収集し、ROOT を使って解析して、レポートなどで提出してもらっています。例題をやるよりも自分のやりたいことを調べる方が勉強になると思いますし、研究を始めていくためのよい助走になっていると自負しています。学生さんのテーマ設定も興味深く、「抜け毛の本数」「PEANUTS の主人公は Snoopy ではなく Charlie Brown なのは本当か」「Nintendo Switch の転売価格」「競馬」などバラエティに富んでおり、講師の楽しみの 1 つです。残念ながら本業が忙しいのか、COVID-19 で大学が混乱していた 2020 年以外の年の宿題提出率は高くありませんので、学生さんにそれとなく提出を推奨していただけるとありがたいです。

3 なぜ ROOT 講習会をしているのか

改めて考えると、私(や若手の会の会員)が ROOT 講習会を手弁当でやる理由はなんなのでしょう。少なくとも、学生、ポスドク主体で結成されている本会において、教育が職務として含まれている会員はほとんどいません。常勤の職を探すことが難しい現状を鑑みると、講師をしている者は 4 年生/修士 1 年生向けの講習会に時間を使わず、研究を進めるべきです。むしろ、業務外の余計なことをしているとして評価を下げるかもしれません。

⁶なお、例外的に水越は壁に向かって話すことはあまり苦痛ではないです。

ROOT の開発元となんらかの契約があるわけではありませぬし、ROOT への愛があるというわけでもありません。両筆者は ROOT より python (numpy+matplotlib) を使ってきた時間の方が既に長くなってしまいました。学会で講師ということがバレルと飯を奢らざるを得なくなってしまうというデメリットも存在します。また、才気煥発な学生と話したり、共に仕事をするのは本当に楽しいものですが、残念ながら講習会という性質上、双方向のコミュニケーションは限られます。あるいは、私がソフトウェアエンジニア的な思想に染まり、クソコードは存在が許されないとするほど傲慢⁷になっており、ROOT 講習会を行うことによって、アカデミックに蔓延する、人間にも計算機にも思いやりの不足したコードを減らし、将来私が参加する実験に十分に整備されたツールが存在することを期待しているのでしょうか。常に短期的成果⁸を求められるポストドク、学生にとってそれは動機に繋がらないと思われまふ。

このように考えると、ほとんど唯一の ROOT 講習会を行う動機は、後輩に良い道を作り、業界を盛り上げていくことに貢献したいという気持ちがあることだと思ひます。快く講師を引き受けてくれる若手の会のメンバーと (顕に確認はしていませんが) このような志を共有できることは嬉しいですし、ありがたいと感じています。宇宙、素粒子、原子核業界の未来をこれからも共に作っていきたくと思ひます。

4 ROOT 講習会の今後

ROOT 講習会を発展させていくには多様な方向性が考えられると思ひます。

現状、講習の時間的制約等から、取り入れたいし、取り入れるべきなのにもかかわらず、実現していないことはたくさんあります。git で version control する方法に慣れておくべきだと思ひますし、cmake で Cross-platform で binary を作る方法も知っておくべきです。確率、統計について体系的に講義することもできていませんし、Fitting の意味も十分に伝えることはできていないでしょう。

講師がしたいという方がいれば是非 (可能なら YMAP に入ってください) お願いしたいと考えております。

よりよい講習会にするため、ポジティブな意見に限らず、ROOT なんか遅れているというようなネガティブな意見もいただけると幸いです。

研究室によっては、4 年生前期は座学と院試勉強が中心で、データ解析を始めているというケースもあるかと思ひます。そのような環境では、5 月の開催時期では、

そもそも学生さんに ROOT を使うモチベーションがないと思ひます。Follow-up の講習を 10 月ごろに実施するという発展も検討しています。ROOT 講習会を合宿形式で開催したり、プログラミングコンテストを行ったりすることは予算の都合上実現していません。そもそも、YMAP は研究会実施のため以外に予算を獲得していないので、ROOT 講習会に予算はありません。Zoom の有料アカウント⁹を購入することや、自由研究を表彰して副賞を出すため¹⁰にも多少の予算がほしいのですが、具体的な方法には思い至っておりませんので、よいお知恵があれば拝借したいです。本活動を持続可能なものにしていくためにも、後の講師に講義以外の負担が当然視されることを避けたいと考えております。もし志のある方がおられましたら、ご寄付をお願いします。GitHub sponsors [https://github.com/sponsors/ymapteam] にてお手続きいただくとありがたいです。よろしくお願ひします。



図 1: YMAP GitHub sponsors への QR コード。

実際に講師については、世代交代が進んでいます。私たちがも適当なところで楽隠居 (という名の自分のプロジェクトへの集中) をさせてもらえることを期待しています。今 ROOT 講習会を受けた世代が、YMAP に入会して、数年後に講師をしてくれれば、これ以上の幸せはありません。

来年度も ROOT 講習会をやりますので、ぜひ対象者への周知をお願いします。

https://github.com/ymapteam/root_lecture

参考文献

- [1] Tom Christiansen, Brian d Foy, Larry Wall, and Jon Orwant, "Programming Perl: Unmatched power for text processing and scripting", O'Reilly Media, Feb. 2012

⁷プログラマの 3 大美德のひとつと呼ばれます。初出は—"We will encourage you to develop the three great virtues of a programmer: laziness, impatience, and hubris." [1]

⁸着任後 3 年

⁹参加者は 100 名を超えるため、ビジネスプラン (2700 円/月) 3ヶ月分が必要

¹⁰現状講師の自腹