

科学計算が得意な モダン言語 Julia 登場

ご購入はこちら

寺崎 敏志

ここでは、Julia (<https://julialang.org/>) という科学計算が得意なプログラミング言語を紹介し、Juliaは、2009年にMITにおいて開発がはじまり、2012年に公開された比較的新しい言語です⁽¹⁾。2018年にJulia v1.0がリリースされ、これにより基本となる言語仕様が固まったので、Juliaに入門しやすくなりました。2020年4月現在では、Julia v1.4.0が安定版でリリースされています。ソースコードはGitHubで公開されています⁽²⁾。

Juliaは、科学計算が得意な言語ではありますが、通常のソフトウェア開発も可能なパッケージ管理システム、ライブラリ、エディタ・統合開発環境、テスト・継続的インテグレーション、ドキュメント作成支援などのエコシステムが備わっています。

本特集では、幾つかのサンプル・コードを通じてJuliaでどういうことができるか、何が魅力的なのかを紹介していきます。サンプル・コードは素朴ですが、応用するためのエッセンスを入れてあります。解説では筆者がこういうことを知っておきたかったというTipを隠し味に入れてあります。最終的にはプロジェクトを作成し、ラズベリー・パイでも動作するJuliaの簡易アプリケーション作成を行います。Juliaを現場で使うにあたっての最小限のワークフローを体験できます。

私の体験に見る Juliaが求められる背景

● 数学とC/C++の板挟み

筆者の体験からJuliaが求められる背景を説明します。数学専攻の大学の講義で、2週間ほど集中的にC/C++を扱うことになりました。こじんまりとしたデスクに入ったBorland C++ Builderを使い、メモ帳でペチペチとキーを叩いていました。

コンパイルしようとしても、エラーでコンパイルできず、どこがおかしいのかもまったく分かりません。なんとか動かしても、ランタイム・エラーで何もできないという悲惨な状態でした。

講義で学んだ逆行列を求めるクラメル公式を5×5の場合で行い、レポートを出したとき、「もう嫌だ、こ

んなことしたくない」と心に誓ったのを覚えています。

社会人になり、仕事で初めてプログラミングを始めました。「あれ、プログラミングしないんじゃないかなったっけ」と思われるでしょう。実は就職説明会で「数学ができる人を探している。プログラミング？大丈夫、君は数学をやっていればいい」と言われ入社しました。しかし名刺を見ると「ソフトウェアエンジニア」と書かれていました。大きな餌に見事引っかかってしまったようです。

餌に引っかかってしまった魚は、C++という水槽の中でひとまず生きることにしました。いわゆる計算幾何学(Computational Geometry)の分野のアルゴリズムの開発・高速化をすることになりました。「自分の頭の中では既にできていても、C++で表現する方法が分からない」という見えない壁に何度もぶつかりました。最終的には愚直に書いていくことを決断しましたが、後から機能追加・修正時に自分の書いたコードでさえ、「これ何やってたんだっけ」と手がつけられなくなった記憶があります。

ある種のデータを取り出し、それをまとめる作業も行っていました。プログラムが吐き出したCSVをエクセルに投げて表示させますが、プログラムが変わると、Excelの表も変えなければならず、ただでさえ慣れていないプログラム作業に加え、異なる言語で書かれたツールを複数管理をすることを強いられました。もちろん作業効率も良いわけがありません。当時の筆者ではC/C++の範疇で「目的を設定し、それを達成する方法を練り出し具現化する。それを試行錯誤する」というサイクルをこなすことができませんでした。

● Pythonとの出会い

そんな中、勉強会でPythonというものを知りました。初めてのプログラミングはPythonが最適!とか機械学習・深層学習というワードが飛び交っています。当時は何やら機械学習、深層学習というものがあるらしい…Pythonというものがあるらしい…という程度の空気感でした。

Pythonは学習コストが低く、プログラムを書くに当たっ

◆参考文献◆