

## オブジェクト指向言語 Ruby による多次元データの解析・可視化環境の開発

### Development of tools for analysis and visualization of multidimensional data with the object-oriented language Ruby

# 川那辺 直樹[1], 堀之内 武[1], 塩谷 雅人[2], 後藤 謙太郎[3], 黒井 啓子[4]

# Naoki Kawanabe[1], Takeshi Horinouchi[2], Masato Shiotani[3], Kentaro GOTO[4], Keiko Kuroi[5]

[1] 京大・宙空電波, [2] 北大院・地球環境, [3] 北大院・理・数, [4] 富士通 FIP 嘱託

[1] RASC, Kyoto Univ, [2] RASC, Kyoto Univ., [3] Graduate School of Environmental Earth Science, Hokkaido Univ., [4] Division of Math. Hokkaido Univ., [5] Fujitsu FIP Corporation

<http://www.kurasc.kyoto-u.ac.jp/radar-group/members/kawanabe/>

#### はじめに

地球・惑星流体科学におけるデータが多種類、大量になるにつれて従来の解析・可視化システムでは対応できなくなることが懸念される。一方、データは多次元数値配列の他に単位や各次元の座標値といった共通の情報を持っている。これらの情報をプログラム上でまとめるには、物理量という名の構造体を定義することである。しかし、データを統一的に扱うには構造体の実装とアクセス方法を切り離すことが必要になる。そこでオブジェクト指向言語である Ruby を用いてデータを柔軟に扱うことのできる解析・可視化環境の開発を行う。

まずは、環境開発に先だつてこの多次元配列と描画ライブラリの実装が必要である。

#### 多次元配列クラス

多次元配列クラス NumArray は計算の高速化を目的とした多次元配列であるため、C 言語を用いた Ruby の拡張ライブラリとして実装されている。Ruby の配列クラスと異なるところは、要素として数値しか扱わないという点であるが、数値のみを扱う配列クラスにすることで実装の際、C 言語の配列を直接利用することができ

るという利点がある。

これまでに、Integer、Float といった基本型に対する算術演算、数学関数の計算機能を実装した。更にデータ解析に用いるための次元ごとの基本的な統計処理や配列処理としてサブセットの取り出し、書き込みを実装した。

#### 描画ライブラリ

基本描画ライブラリとして DENNOU Club Library(DCL)を用いることにする。DCL は Fortran で書かれているため Ruby 上で動かすための拡張ライブラリが必要となる。まずは DCL を C 言語に翻訳し、C 版 DCL から拡張ライブラリ Ruby/DCL を作成した。更に Ruby/DCL ライブラリを下位層とし、上位層として使いやすく多機能な描画ライブラリを作成する。クラス設計には Unified Modeling Language(UML)というモデル設計のための言語を用いた。ここで描画ライブラリは次の要望を満たすべく設計を行った。

- ・描いた図を整形することができる。
- ・Ruby スクリプトでも GUI でも利用できる。
- ・図から情報を取得し、次の解析や描画に利用できるようにする。
- ・描画内容をファイルに保存したり、ファイルからの読み出しが行える。

本描画クラスにおいて、パラメータ変更やデータの描画は下位層で行い、図の整形を行ったり描画内容の保存を行う上で必要となる履歴機能、更にマウスによる操作を新たに作成していく。まず、履歴機能を実現するために我々はコマンドをオブジェクト化した。コマンドオブジェクトはデータ、パラメータからなるプロパティを持ち、それを履歴として積み重ねていく。ここでプロパティ生成について述べておくと、描画クラスにおいて描画命令が呼び出されるとプロパティ生成及びコマンドオブジェクト生成のクラスメソッドを呼び出す。ここでプロパティはコマンドオブジェクト生成の際、データ、デフォルトパラメータ、コマンド固有の設定をもとに生成され、コマンドオブジェクトがプロパティを持つという構成となる。

#### 今後の展望

多次元配列に関しては Complex 型に対応していくとともに配列としての機能追加や高度な数学関数への対応など、より実用的なものへと仕上げていく。描画ライブラリに関してはクラス設計を行い、実装へと移していく。更に GUI を組み込みマウスからの操作を可能にすることによって、操作性の高い、かつ多機能な描画クラスが完成するはずである。そしてこの多次元配列、描画クラスをもとにデータ形式にとらわれない解析、

可視化環境の開発を行っていく。

#### 謝辞

本研究を行うにあたって、数々の御協力を頂いた電脳 davis プロジェクトの皆様へ深く感謝いたします。