

# 電腦 RUBY の 現状と課題

西澤誠也

地球流体電腦倶楽部

惑星科学研究センター (CPS)

# 地球流体電脳倶楽部

## ◎ 設立

- 1988年 (塩谷, 酒井, 余田, 林)

## ◎ ポリシー

- 研究教育のための情報は無料で無ければならない
- 隣の専門家がわかる情報を提供すべし
- 技術的な仕事、教育活動も研究活動の一環

## ◎ 活動

- 知見アーカイブ
- セミナー開催 (GFDセミナー, FDEPS, チュートリアルセミナー, etc)
- 研究基盤ライブラリ・ツール群の開発
- サーバー運用 (<http://www.gfd-dennou.org/>)

## ◎ 研究基盤ライブラリ・ツール開発

### ■ dcmode1 プロジェクト

- 数値計算ライブラリ・数値モデル
  - ISPACK, gtool5
  - spmodel, DCPAM, deepconv, etc

### ■ davis プロジェクト

- データ解析・可視化ライブラリ・ツール
  - DCL (グラフィックス・文字処理・数値基礎処理ライブラリ)
  - 電脳 ruby プロジェクト
  - Gfdnavi (データベース・解析・可視化サーバーソフトウェア)

# 解析・可視化対象データ

## ◎ データソース

- 観測 (in situ観測, リモートセンシング)
- 数値シミュレーション結果
- etc

## ◎ 多次元配列物理量

- 次元: 時空間, 波数空間, 位相空間, etc

## ◎ 自己記述型フォーマットファイル

- 配列データー, 次元データー, 属性
- 複数のフォーマットが存在
  - e.g. NetCDF, GrADS (+ctl), GRIB, HDF-EOS, NuSDaS

# 開発プログラミング言語

## ◎ Fortran, C

- 実行速度が速い
- 過去の資産が豊富
  
- ソース変更後、実行前にコンパイルが必要
  - 対話的操作は不可
- 開発・デバッグに要する時間が比較的長い
  - エラー原因箇所の特定に時間がかかる (e.g. segmentation fault)
  - 文字列処理が困難 (Fortran)
- 関数・プログラムの使い回しがやりにくい
  - 抽象化が難しい (c.f. 異なるファイルフォーマットや次元)

◎ トータルのコスト = 開発コスト + 実行コスト

■ 数値モデル

- 完成品を何度も実行する
- 実行に時間がかかる

→ 開発に時間がかかっても、実行速度が速い言語が適している (Fortran, C)

■ 解析・可視化

- 試行錯誤を繰り返す
- 実行は比較的短時間

→ 開発が素早くできる言語が適している

## ◎ 解析・可視化は Ruby で行うことを選択

- スクリプト言語
  - コンパイル不要
  - 対話的操作が可能
- オブジェクト指向言語
  - 抽象化が容易
- 汎用言語
  - 多彩な汎用ライブラリ群
    - さまざまな種類のライブラリが利用可能
- 国産
  - コミュニティー
  - ドキュメント

(c.f. Python, Java, GrADS, IDL, NCL, ...)

→ 電脳 ruby プロジェクト

# 電脳 RUBY プロジェクト

- ◎ 解析・可視化のための基盤ライブラリの構築
  - これまで Fortran で行ってきた解析・可視化を、より簡単に行うための基盤ライブラリ
- ◎ データ等の高度利用のためのツール群の開発
  - データ爆発など、あらたな問題に対応するため、新しいスタイルで研究を行うためのツール
    - 汎用言語である Ruby の特徴を利用



English Page

ホーム  
ニュース  
ハイライト  
背景/目標  
謹製品  
インストールガイド  
Debian / Ubuntu  
RPM  
FreeBSD Ports  
Fink / MacPorts  
Windows / Cygwin  
VMwareイメージ  
LiveCD  
メーリングリスト  
ライセンス

チュートリアル  
Tips  
小物置き場  
ドキュメンテーションリンク集  
ワークショップ・学会等

Rubyホームページ  
地球流体電脳倶楽部

昔の資源 (obsolete)

### ようこそ

このページはオブジェクト指向スクリプト言語 [Ruby](#) を、地球物理学におけるデータ解析、可視化、数値シミュレーションに使う人々の広場です。ボランティアベースのゆるい括りのプロジェクトとして、そのためいろいろなソフトウェアを作り提供しています。あなたの参加をお待ちしています。

### ニュース

- [2012/3/5 - 6 に九州大学で 地球流体データ解析・数値計算ワークショップ \(dcmodel/davis チュートリアル\) を開催します。](#) (2012/02/09 posted by Nishizawa)
- [Ruby-HDFEOS5](#) をリリースしました。 (2010/02/17 posted by horinouchi)
- [過去のニュース](#)

### ハイライト

- [GPhys](#)  
多次元物理データ取り扱いライブラリー (gave で使われています)。このライブラリーを使うと NetCDF や GrADS 形式のファイル中や実行時のメモリー上の物理量データを統一的に扱うことができます。
- [gave](#)  
GUI ベースの格子点データ可視化・解析ツール。 [スクリーンショット](#)
- [Gfdnavi](#)  
Web ベースの地球流体データベース・解析・可視化のための汎用ツール
- [RubyDCL](#)  
描画ライブラリー [DCL](#) の Ruby 用ラッパー。
- [RubyNetCDF](#)  
[NetCDF ライブラリー](#) の Ruby 用ラッパー。
- [その他いろいろ...](#)

### 電脳Ruby小物置き場

「もしかしたら他の人の役にたつかも」と思うプログラムや情報を自由に投稿し、共有するための場所です。お使いください。

- [電脳Ruby小物置き場](#)

### メーリングリスト

(基本的)日本語によるメーリングリストを運営しており、自動登録で誰でも参加できます。今のところ開発者用とユーザー用は分けていません。参加希望の方は、登録を希望するアドレスから [dennou-ruby-ctl@notwork.org](mailto:dennou-ruby-ctl@notwork.org)宛てに、本文第一行目が以下のようなメールを送ってください。

subscribe your name

ここで「your name」はあなたの名前前のアルファベット表記に置き換えてください。登録を確認するメールが行きますのでそれにご返信してください。

- [メーリングリストのアーカイブページ](#)

### ライセンス

本プロジェクトで提供するソフトは、特に断らない限り [Rubyのライセンス](#)と同じ条件のフリーソフトとして配布されます。著作権はそれぞれの作者と地球流体電脳倶楽部が有します。

Contact: Takeshi Horinouchi ([horinout@ees.hokudai.ac.jp](mailto:horinout@ees.hokudai.ac.jp)),

GFD Dennou Club: [dcstaff@gfd-dennou.org](mailto:dcstaff@gfd-dennou.org)

Copyright (C) GFD Dennou Club, 2000 - 2011 All rights reserved.

<http://ruby.gfd-dennou.org/>

# 製品概要

## ◎ 多次元配列

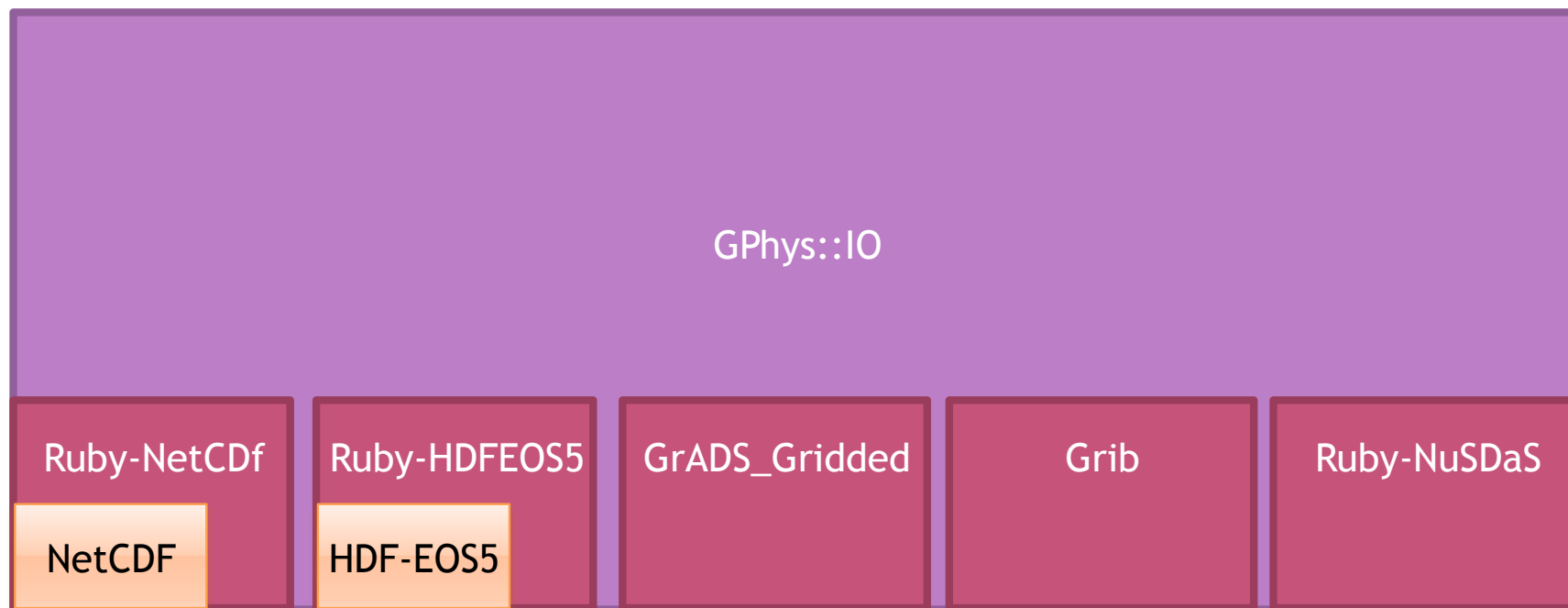
- 多次元数値配列 (NArray; by 田中さん)
- 欠損値処理拡張 (NArrayMiss)

## ◎ 数学演算

- GSL ラッパー (rb-gsl; by 常定さん)
- LAPACK, SSL2, FFTW3, ISPACK, SPML 等のラッパー

## ◎ ファイル IO

- 各種ファイルフォーマット: 統一的なメソッド定義
- 抽象化: フォーマットによらず、同じようにアクセス可能なメソッドを提供 (GPhys::IO)



## ◎ データオブジェクト

- 配列データだけでなく、軸データ・属性を保持する配列的クラスオブジェクト (GPhys)

## ◎ グラフィックス

- DCL のラッパー (配列長など、不要な引数は削除)
- GPhys を与えることで軸等を自動描画するライブラリ (GGraph)

## ◎ データサーバー

- データベース・解析・可視化サーバー構築用ソフトウェア (Gfdnavi) (Gfdnavi プロジェクトとして独立)

# アドバンテージ

## ◎ 簡潔に記述できる

- 行数が少ない

```
require "numru/ggraph"  
include NumRu  
t = GPhys::IO.open("sample.nc", "T") # ファイルオープン  
DCL.gropn(4)  
GGraph.tone(t) # トーン図作成  
DCL.grcls
```

## ◎ デバッグがしやすい

- 誤った配列インデックスによる segmentation fault が起こらない

```
% cat test.rb  
require "narray"  
ary = Narray.new(2,3) # 2 x 3 の配列  
ary[0,5] = 1 # インデックス範囲を超えたアクセス  
  
% ruby test.rb  
test.rb:3:in `[]=: index out of range (IndexError)  
from test.rb:4
```

既存のライブラリ・ツールと比較して、使いやすくなった (と個人的には思う)

では、広く使われるようになったか？

たまに使っていることを耳にするが、基本的には、関係者 + その周辺

| name         | tarball | deb    | rpm     | gem |
|--------------|---------|--------|---------|-----|
| gphys        | 4 (56)  | 23     | 32      | 55  |
| ruby-dcl     | 44      | 7 (14) | 28 (41) | 117 |
| narray_miss  | 2 (57)  | 21     | 49      | 104 |
| ruby-netcdf  | 6 (93)  | 19     | 41      | 83  |
| rb-grib      | 5       |        |         | 48  |
| ruby-nusdas  | 24      |        |         |     |
| ruby-hdfeos5 | 39      |        |         |     |
| numru-misc   | 25      | 9      | 39      | 54  |
| numru-units  | 19      | 21     | 57      | 56  |

それぞれの最新版のダウンロード数。

() 内の数字は近いバージョンで最大のダウンロード数

tarball, deb, rpm については、Bot 等を除いた数字。

gem については、単純はダウンロード数 (from rubygems.org)。

何故普及しないのか？

Twitter で議論  
昨日ミーティングで議論



# 普及しない要因

## 始める際の問題

### ◎ 言語の壁

- 新しい言語を覚えるのは敷居が高い

### ◎ インストールの壁

- 依存関係で、大量のライブラリのインストールが必要
- コンパイルが必要な時点で多くの人が敬遠する
- インストール中にエラーが出ると、ほとんどの人があきらめる

### ◎ ドキュメントの問題

- それなりにドキュメントはあるが、初心者向けではない

## ◎ 言語の壁

- GrADS, IDL, NCL 等も独自言語だが、多くのユーザーを獲得している
- 覚えるだけの価値があると見なされるかどうか？

## ◎ インストールの壁

- バイナリパッケージを用意する必要がある
  - 各種 Linux パッケージ
    - debian は佐々木さんが頑張っているが、他ディストリビューションはいまいち
  - windows インストーラー
    - 乙部さんが頑張っているが、コンパイラの互換性問題がある
  - gem
    - 西本さんが作成してくれているが、それぞれ本体に入れるべき (共通のテンプレートを作成する)
    - できればバイナリ gem も用意したい

## ◎ ドキュメント

- 今回、納多さん、堀之内さんにより、GPhys に関しては大きく改訂
- 最初に必要なのはどのような情報であるのか要再考
  - 最初は、視覚的な情報の方が効果的
  - ギャラリー、サンプル集等をトップの目立つところへ
- ドキュメントをレベル毎に分けて、最初の導入は、思い切ってシンプルに (HOP, STEP, JUMP 形式)

## 始めてからの問題

### ◎ 質問の敷居

- MLはあるが、質問するには敷居が高い

### ◎ 利便性の問題

- 業界標準の解析手法のメソッドが多くなく、自分で実装する必要がある
  - 実装済: EP-flux, EOF, ヒストグラム

### ◎ フィードバックの問題

- バグ報告や機能拡張等のフィードバックを気軽にコミットしにくい

## ◎ 質問の敷居

- ML に積極的に質問や Tips を投げる
- ML 以外の場も考える

## ◎ 利便性

- 頑張って増やす
- よりコミットしやすい環境を用意する
  - 小物置き場の活用
  - より気軽に投げられる別の場も考える
- SciPy, NCL を参考に

## ◎ フィードバックの問題

- 外部の人もリポジトリからチェックアウトできるように
- プロジェクト管理ツール等を導入し、バグ報告を簡単に

# まとめ

- ◎ 電脳 ruby プロジェクトにより、基盤の道具立てはそれなりにそろった
  - 適材適所で Fortran・C と Ruby を使い分け
  - より容易に解析・可視化が可能に
- ◎ あまり普及には至っていない
  - 興味があるけれども、途中であきらめてしまう人をできる限り少なくする
    - インストールの障壁
      - バイナリパッケージが重要
    - ドキュメント
      - 開発者が入門ドキュメントを書くと、細かくなりすぎる？



| name            | tarball | deb | rpm | gem |
|-----------------|---------|-----|-----|-----|
| gpv             | 34      | 7   | 29  |     |
| ruby-libfame    | 3       |     |     |     |
| ruby-ssl2       | 11      |     |     |     |
| ruby-spml       | 4       |     |     |     |
| numru-met       | 12      | 16  | 35  |     |
| ruby-lapack     | 40      |     |     | 310 |
| dclext-anim     |         |     |     | 49  |
| gave            | 43      | 19  | 59  |     |
| remote-dcl      |         |     |     | 59  |
| gphys-remote    | 21      | 6   |     |     |
| multibitnums    | 17      | 6   | 41  |     |
| dclext-gtk      |         |     |     | 71  |
| ruby-derivative | 5 (8)   |     |     |     |
| ruby-msise90    | 13      |     |     |     |
| ruby-ispack     | 8 (10)  |     |     |     |