

# 地球流体電脳倶楽部 電脳Rubyプロジェクト

## 数値データ 管理・解析・可視化 アプリケーション “Gfdnavi”

西澤誠也

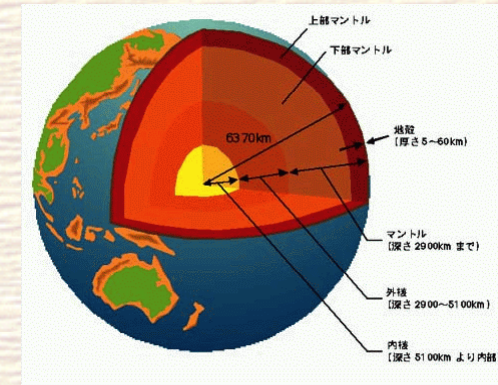
地球流体電脳倶楽部  
/ 京都大学数理解析研究所



# 地球流体力学(GFD)って？

## ■ 地球・惑星に関する流体力学

- 自転、重力、球形状 が重要
- e.g. 大気、海洋、マントル etc

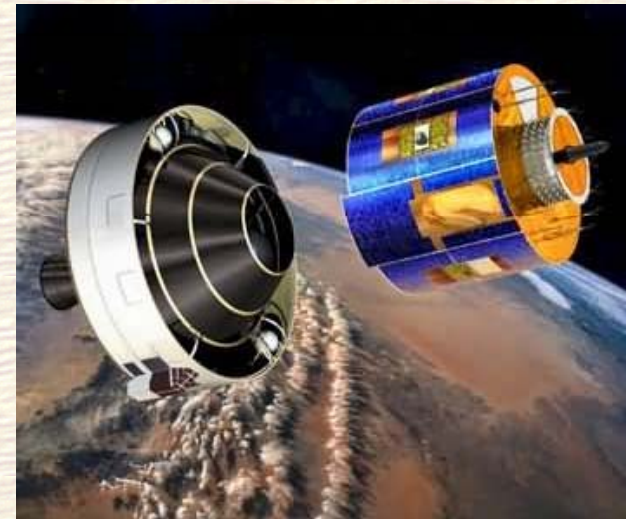


# 我々(研究者)の現状

- データの管理はディレクトリ名、ファイル名ベース
  - 地球大気 — 2006年 — 1月 — 気温.data
  - 風速.data
  - 2月 — .....
- データ解析は自分でプログラミング(主にFortran!!)
  - 人と同じことをやっていると、研究にはならない
  - 万能な汎用ツールは存在しない

# 情報爆発

- 観測測器の高度化
  - 人工衛星搭載の測器 etc
- 計算機の急速な発展
  - 地球シミュレーター etc



データ量が爆発的に  
増加している

数年前はCDサイズ(数百MB)でも  
それなりに大きいと思っていたのに...;



# 我々が直面している問題

## データ管理

- 何処に何があるのか把握できない
  - ローカルなストレージ上のデータ
    - ◆ どんな名前をつけたか忘れる
  - ネット上で公開されたデータ
    - ◆ 知っていないと、検索では非常に困難
    - ◆ T(テラ)級のデータは、ほぼダウンロード無理

## データ解析・可視化

- ファイルフォーマットが数多く存在
- データ、解析・可視化方法毎のプログラム
  - 大量のプログラム → 管理も大変

## エフォート率

- データ管理 20%
- プログラミング 70%
- 脳を使って考える 10%

研究者の職業っていったい何.....

# Denno-Ruby プロジェクト

## ■ 地球流体電脳倶楽部のプロジェクト

- <http://ruby.gfd-dennou.org/>
- since 1999

## ■ 地球流体データの解析・可視化のためのツール整備

- Rubyのお陰により、プログラミングが劇的に簡単、スピーディーに
- ref. るびま6号 Ruby Library Report 第5回

しかし、

依然として、データ管理、プログラミングは必要

# Gfdnavi

## プロジェクトHP

<http://www.gfd-dennou.org/arch/davis/gfdnavi/>

(supported by 文科省科研費「情報爆発」)

## 特徴

- データ管理・解析・可視化Railsアプリケーション
  - データ管理にRDBを使用
  - 解析・可視化は、ブラウザ上からGUIで
- サーバー兼デスクトップツール
  - データが、ネットワーク上にあっても、ローカルにあっても同じように扱える



# 機能

## ■ RDBへの簡単登録

- 既存データをスキャンし、メタデータをRDBへ登録

## ■ データ検索

- ディレクトリツリー型検索
- キーワード検索
- 位置・時刻範囲指定検索
  - ◆ Google Mapによる指定も

reanalysis  
ncep  
T.jan.nc  
UV.jan.nc  
jmadata  
MSM-P  
2006  
sonde\_operational  
std\_plev  
41256.spl.nc  
42410.spl.nc  
43333.spl.nc  
49946.spl.nc

/reanalysis/ncep/T.jan.nc

name	title	description
<input type="checkbox"/> Add	Temperature	

Add selected items

keywords

SEARCH!

drag map | specify a region

地図 航空写真 地図+写真

longitude: 47.754097979680026  
latitude: 152.578125

point2  
longitude: 122.34375  
latitude: 23.725011735951796

Temporal Region  
start: 2005 1 1 00 : 00  
end: 2005 12 31 23 : 59

## ■ データ解析

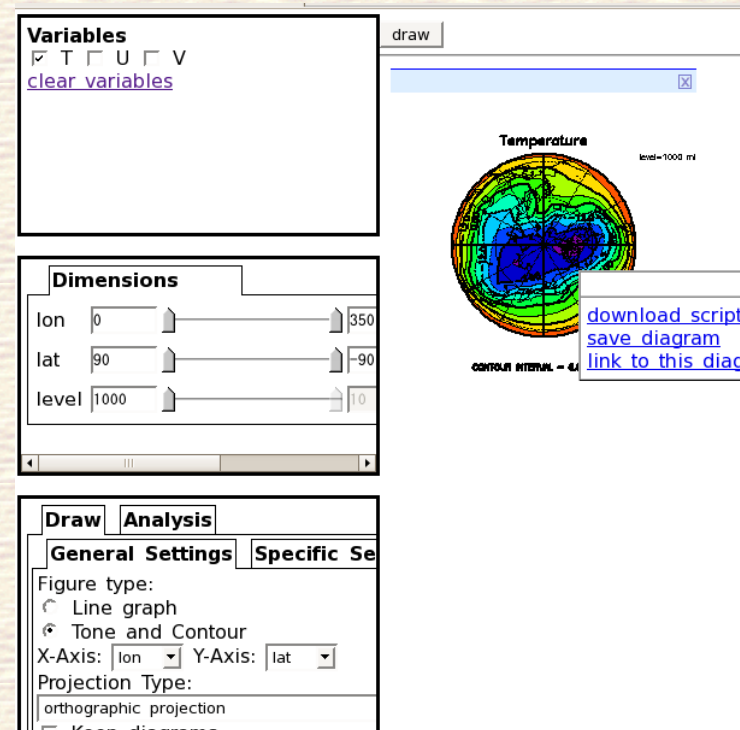
- 統計操作・数学関数の適用
- 独自関数の追加

## ■ 可視化

- 折線・等値線・トーン・ベクトル図
- 重ね合わせ
- アニメーション

## ■ 保存

- 同じ図を作成するための、Rubyスクリプト  
+ 最小限のデータのダウンロード
- データベースに登録



# 将来構想(一部実装済み)

## ■ サイト間連携

### ● P2Pによる横断検索

- ◆ どのサイトにどんなデータがあるか、前もって知っている必要が無い

### ● Webサービスによる横断利用

- ◆ 解析・可視化に必要なすべてのデータが同じサイトにある必要が無い

## ■ 数値データと知見情報の融合(有機的結合)

### ● バイナリデータに人間の解釈を付与

- ◆ e.g. この図の何処々を見ると、何々が分かる

### ● 数値データ間の関係リンク付け

- ◆ e.g. このデータ(図)は、あのデータから計算(描画)された

# 明るい未来

## ■ データ管理

- ローカルデータはスキャンスクリプトを走らせるだけ
- 必要なデータは、何処にあるかにかかわらず、検索で得られる

## ■ データ解析・可視化

- ブラウザがあれば、いつでも何処でも
- プログラミングは不要なし。GUI操作で

## ■ データ公開も簡単

## エフォート率

### Before

- データ管理 20%
- プログラミング 70%
- 脳を使って考える 10%

### After

- データ管理 5%
- プログラミング 15%
- 脳を使って考える 80%

# 以降 スクリーンショット

# 検索 & 絞込み

Gfdnavi - Mozilla

File Edit View Go Bookmarks Tools Window Help

Back Forward Reload Stop <http://localhost:3000/search/> Search Print

**Gfdnavi** **GFDNAVI**

Search Analysis Login

[Search by Map and/or Text](#)

Select from directory tree:

name	title	description
<input type="checkbox"/> Add T	Temperature	

Add selected items

[clear tree](#)

# 検索 & 絞込み

ブラウザメニュー: ファイル(E) 編集(E) 表示(V) 移動(G) ブックマーク(B) ツール(I) ヘルプ(H) | seiyani

アドレスバー: http://localhost:3000/search/mapsearch

タブ: Gfdnavi

## SEARCH BY GOOGLE MAP

Search Analysis Login

keywords

SEARCH!

drag map | specify a region

地図 | 航空写真 | 地図+写真

Spatial Region

point1  
longitude: 47.754097979680026  
latitude: 152.578125

point2  
longitude: 122.34375  
latitude: 23.725011735951796

Temporal Region

start: 2005 1 1 00 : 00

end: 2005 12 31 23 : 59

Google Map利用検索

http://www





# 可視化

ブラウザのメニュー: ファイル(E) 編集(E) 表示(V) 移動(G) ブックマーク(B) ツール(T) ヘルプ(H)

ブラウザのアドレスバー: <http://localhost:3000/analysis>

タブ: Gfdnavi

**Variables**

T  U  V

[clear variables](#)

**Dimensions**

lon: 0 (slider) 350

lat: 90 (slider) -90

level: 1000 (slider) 10

**Draw** | **Analysis**

**General Settings** | **Specific Se**

Figure type:

- Line graph
- Tone and Contour

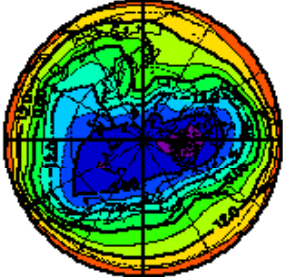
X-Axis: lon Y-Axis: lat

Projection Type:

- orthographic projection
- Keep diagrams

**draw**

Temperature level=1000 m



CONTOUR INTERVAL = 4.000E+00

<http://www>

# 可視化

ブラウザのメニュー: ファイル(E) 編集(E) 表示(V) 移動(G) ブックマーク(B) ツール(I) ヘルプ(H)

アドレスバー: http://localhost:3000/analysis

タブ: Gfdnavi

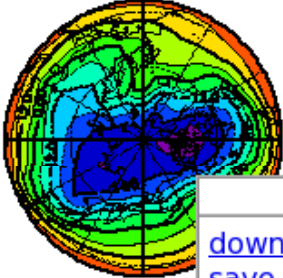
**Variables**  
 T  U  V  
[clear variables](#)

**Dimensions**  
lon: 0 to 350  
lat: 90 to -90  
level: 1000 to 10

**Draw** | **Analysis**  
**General Settings** | **Specific Se**  
Figure type:  
 Line graph  
 Tone and Contour  
X-Axis: lon Y-Axis: lat  
Projection Type:  
orthographic projection  
 Keep diagrams

draw

Temperature  
level=1000 m



CONTOUR INTERVAL = 4

[download script and data](#)  
[save diagram](#)  
[link to this diagram](#)

http://www

# 可視化

ブラウザのメニュー: ファイル(E) 編集(E) 表示(V) 移動(G) ブックマーク(B) ツール(T) ヘルプ(H)

ブラウザのアドレスバー: <http://localhost:3000/analysis>

ブラウザのタブ: Gfdnavi

### Variables

T  U  V

[clear variables](#)

### Dimensions

lon: 8.75 → 348

lat: 81.25 → -77

level: 1000 → 10

### Draw Analysis

**General Settings** **Specific Se**

Figure type:

- Line graph
- Tone and Contour
- Vector

X-Axis: lon Y-Axis: lat

Projection Type: rectangular\_uniform\_coordinate

draw

### Temperature

level=1000 m

latitude

longitude

CONTOUR INTERVAL

### (zonal wind, meridional wind)

level=1000 hPa

latitude

longitude

# 可視化

ブラウザのメニュー: ファイル(F) 編集(E) 表示(V) 移動(G) ブックマーク(B) ツール(T) ヘルプ(H) | seiyani

アドレスバー: <http://localhost:3000/analysis> | 移動

タブ: Gfdnavi

**Dimensions**

lon: 8.75 | 348

lat: 81.25 | -77

level: 1000 | 10

**Draw Analysis**

**General Settings** | Specific Se

Figure type:  
 Line graph  
 Tone and Contour  
 Vector

X-Axis: lon | Y-Axis: lat

Projection Type:  
rectangular uniform coordinate

Keep diagrams

Diagram size:  
 large  medium  small

**Temperature**  
(degree\_south)

level=1000 m

Longitude (degree\_south)

CONTROL INTERVAL = 4.00E+00

**(zonal wind,meridional wind)**  
(degree\_south)

level=1000 hPa

longitude (degree\_south)

**Temperature**  
(degree\_south)

level=1000 m

Longitude (degree\_south)

CONTROL INTERVAL = 4.00E+00

<http://www>