Database and data-analysis infrastructure for atmospheric studies

Takeshi Horinouchi

RISH, Kyoto Univ.
New IT Infrastructure for the Information Explosion Era
<table>
<thead>
<tr>
<th>Research Groups:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Info-plosion in geophysical fluid sciences</td>
</tr>
<tr>
<td>T Horinouchi et al</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Time</th>
<th>Session</th>
</tr>
</thead>
<tbody>
<tr>
<td>15:18-15:40</td>
<td>(休憩)</td>
</tr>
<tr>
<td>15:40-16:52</td>
<td>A01-06 「自然言語意味概念の共通的記述言語による次世代知識Web基盤」石塚 深 (東京大学)</td>
</tr>
<tr>
<td>16:52-17:00</td>
<td>A01-35 「多次元軌跡データからの知識発見とその医療応用に関する研究」津本 周作 (島根大学)</td>
</tr>
<tr>
<td>17:00-18:00</td>
<td>A01-14 「地球科学データの高度利用と流通のための基盤開発」堀之内 武 (京都大学)</td>
</tr>
<tr>
<td>18:00-18:00</td>
<td>A01-23 「最新情報技術を活用した超大規模天文データ解析機構の研究開発」大石 雅寿 (国立天文台)</td>
</tr>
</tbody>
</table>
Topics

• **GPhys** – a Ruby library for data analysis and visualization

• **Gfdnavi** – a tool to create Web-based database for data search, analysis, and visualization *(demonstrated earlier by Seiya Nishizawa)*
Object-oriented languages are suitable to data analysis

- Information hiding → can hide data structure
- Polymorphism → can unify access

→ One can create a programming library to support consolidated handling of a variety of atmospheric data irrespective of dimensionality, sampling, and formats
→ Can be create software infrastructure for scientific data analysis
Why Ruby?

• Scripting language (interpreter) → Rapid development
• Can be used interactively → Good for trial and error
• Pure object-oriented and easy to use
  → Easy to develop reusable flexible software
  → Community software sharing
• Easy to extend → Reuse C and Fortran libraries
• Many and increasing libraries (Networking / GUI toolkit / database / distributed objects / etc.etc.) → Easy to implement high-level services
• Good string-handling capability (like perl) → You need it even when you deal with scientific data
GPhys: a Ruby class library for Gridded Physical quantities


Developed by T Horinouchi, R Mizuata, S Takehiro, S Nishizawa, etc

- Supports unified access to NetCDF, GrADS, GRIB, and NuSDAS files (HDF5-EOS will also be supported)
- Easy to use as GrADS, but more powerful, flexible, and programmable
Sample scripts (1)

• Contour plot

1: require "numru/ggraph"
2: include NumRu
3: gphys = GPhys::IO.open('T.jan.nc', 'T')
4: DCL.gropn(1)
5: GGraph.contour(gphys)
6: DCL.grcls

One can put these lines in a start up file for interactive session

This file is a NeCDF. But the program will be the same for a GrADS file:

gphys = GPhys::IO.open('T.jan.ctl', 'T')

CONTOUR INTERVAL = 6.000E+00
Sample scripts (2)

• Line plot

1: require "numru/ggraph"
2: include NumRu
3: gphys = GPhys::IO.open('T.jan.nc', 'T')
4: DCL.gro(1)
5: GGraph.set_fig( 'itr'=> 2 )
6: GGraph.line( gphys.cut(135,35,false),
  true, 'exchange'=>false)
7: DCL.grcls

The only difference form
the previous script

One can incrementally add options for publication-level graphics
Sample scripts (3)

- Deviation from zonal mean

```ruby
1: require "numru/ghys"
2: include NumRu
3: gphys = GPhys::IO.open('T.jan.nc', 'T')
4: eddy = gphys - gphys.mean('lon')

5: ofile = NetCDF.create('zm.nc')
6: GPhys::IO.write( ofile, eddy )
7: ofile.close
```

If you want to save the data in a new NetCDF file (along with attributes and coordinate variables)
Sample scripts (3’)

- Deviation from zonal mean (by defining a “method”)

```ruby
1: require "numru/ghys"
2: module NumRu
3:   class GPhys
4:     def eddy(dim)
5:       self - self.mean(dim)
6:     end
7:   end
8: end
9:
10: include NumRu
11: gphys = GPhys::IO.open('T.jan.nc', 'T')
12: eddy = gphys.eddy('lon')
```
Also supported are

- Virtual concatenation over multiple files (tiling etc.)
- Big data handling by automatic process subdivision
- Mathematical / statistical operations (e.g. FFT);
- Numerical/scientific libraries such as GSL can be combined. You can further extend GPhys by C / Fortran libraries
- etc etc
Gfdnavi

Available from http://www.gfd-dennou.org/arch/davis/gfdnavi/

- Developed under info-plosion
- by T. Horinouchi, S. Nishizawa, C. Watanabe, etc. etc.
- Most part was written by S. Nishizawa
Technologies

• **GPhys** for consolidated data access

• **Ruby on Rails**: A development framework for web-application with relational database – *very comprehensive and productive*
• Register local data (and/or remote OPeNDAP data) automatically with a command
• Serve the data for search, analysis, and visualization
• Supported data: gridded numerical data (in NetCDF, grib, etc.)
• UI: Web browsers and Web service
Metadata DB

- **Metadata**
  - name-value attributes
  - coordinate info (spatial, time, or any coordinate e.g. wavelength)

- **Generated by directory scan**
  - All directories, numerical data files, image files, etc are registered in relational database

![Diagram showing directory structure and metadata attributes]

- **Attributes**
  - keyword attributes
  - spatial attributes
  - temporal attributes

- **Variables**
  - name
  - type
  - value

- **Description**
  - "........."
  - param1 = value1
  - param2 = [val21,val22]
Data finder user interface

Details

DOWNLOAD THIS FILE
UV.jan.nc  [plain file]  /samples/reanalysis/ncap/UV.jan.nc
ncap climatology

1. /
2. samples
3. reanalysis  : 再解析データ
4. ncep  : NCEP 再解析データ
Visualizaion / analysis window
Search interface with Google Map support
Network of Gfdnavi

Under development by C Watanabe (Ochanomizu Univ; database science)

✓ To create peer-to-peer network for cross search and cross use
✓ Then one can access local data and remote data together
Knowledge/finding DB creation (new feature)

Visualization window

Input form
View the document

Re-drawing ⇒ change setting and redraw etc...
Can be used to create a page like this

TRMM Web page
http://www.eorc.nasda.go.jp/TRMM/index_j.htm
2d-cyclic-ee：減衰乱流問題実験1-1（実験別結果）

<table>
<thead>
<tr>
<th>実験リスト</th>
<th>ケース1</th>
<th>ケース2</th>
<th>ケース3</th>
<th>ケース4</th>
<th>ケース5</th>
</tr>
</thead>
<tbody>
<tr>
<td>切断波数・格子点数</td>
<td>X/Y 方向切断波数: 21</td>
<td>X/Y 方向格子点数: 64</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>時間積分</td>
<td>時間積分間隔: 1.0d-3</td>
<td>時間積分ステップ数: 10000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>物理パラメータ</td>
<td>β=400.0</td>
<td>β=300.0</td>
<td>β=200.0</td>
<td>β=100.0</td>
<td>β=000.0</td>
</tr>
<tr>
<td>実験パラメータ</td>
<td>初期エネルギー分布の全波数領域最大値: 11.0DD</td>
<td>初期エネルギー分布の全波数領域最大値: 13.0DD</td>
<td>初期平均エネルギーの値: 1.0DD</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

ケース1: β=400
New search interface (Ocha U)
3D visualization (to be implemented at Nara W univ)
Concluding remarks

• Ruby, a modern object-oriented scripting language, can help and accelerate atmospheric data analysis
  – The GPhys library can be a community tool for atmospheric sciences

• Gfdnavi can be a community tool for atmospheric data sharing/publication
  – Active development is ongoing
What’s Ruby?

• An object-oriented scripting (interpreted) language
• Open software (http://www.ruby-lang.org/)
• Simple and sophisticated syntax

• Online version of an English book
  www.rubycentral.com/book by Thomas and Hunt