趣旨説明と世界情勢について

Ruby におけるデータ処理の現状

佐々木洋平

北海道情報大学

2023 年 3 月 29 日 地球流体データ解析・数値計算ワークショップ

いわゆる前座です

我々の取り組み

- 電脳 Ruby プロジェクト
  - 「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, そのためのいろいろなソフトウェアを作り提供しています.」
    - Ruby Library Report 【第 5 回】 数値計算と可視化
      (Ruby Library Report 第 5 回、るびま, 2005)
    - 地球流体科学における Ruby の利用
      (プラズマ核融合学会誌, 2008 年 4 月)
- 謹製品
  - Ruby NetCDF, Ruby DCL, GPhys, Gfdnavi...

GPhys, GGraph(1)

サンプルコード

スタートアップファイルによる設定簡略化をしない場合

```
require ”numru/ggraph”   #可視化なしなら”numru/gphys”で良い
include NumRu      # 名前空間確保用のNumRuをつけなくていいよう
gp = GPhys::IO.open 'air.2011.nc', 'air'      # データ読込
gp = gp[false,0..10,true]
        #^ 後ろから2番目の軸(高度)の添え字で絞込み(バーチャル)
gp = gp.cut(”lon”=>0..120)   # 座標値で絞込み(バーチャル)
gpmean = gp.mean(”time”)
        #^ 時間(最後の軸)について平均(ここで初めてデータ読み込み)
        #   結果はメモリ上に. でもクラスはGPhysのまま
GGraph.tone gpmean      # (GGraphによる) 色塗り描画
DCL.grcls
```

上記のrequire と include (いつものおまじない)のあと:

```
gp = GPhys::IO.open 'T.ctl', 'T'     # データ読込(GrADS形式)
file = NetCDF.open(”air.mean.nc”)    # NetCDFファイル
GPhys::IO.write(file, gpmean)        # メタデータ含めコピー
file.close                           # ファイル閉じ
```

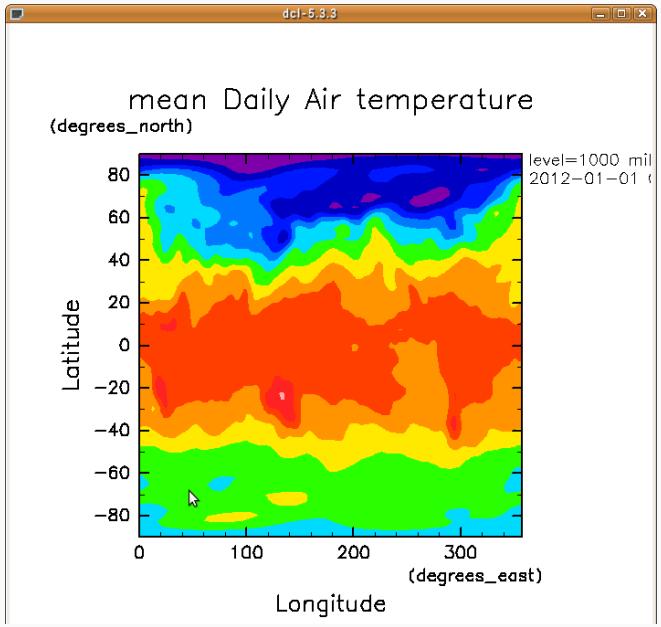
GPhys, GGraph(2)

例 「GPhys, GGraphチュートリアル」(http://ruby.gfd-dennou.org/products/gphys/tutorial2/) より.
（irb用スタートアップファイルを利用して）

GPhys::IO “air.2012-01.nc”, “air” と同じ

```
$ irb_ggraph
>>  gp = gopen 'air.2012-01.nc/air'     # データ読込
>>  tone gp                             # 色塗り描画
```

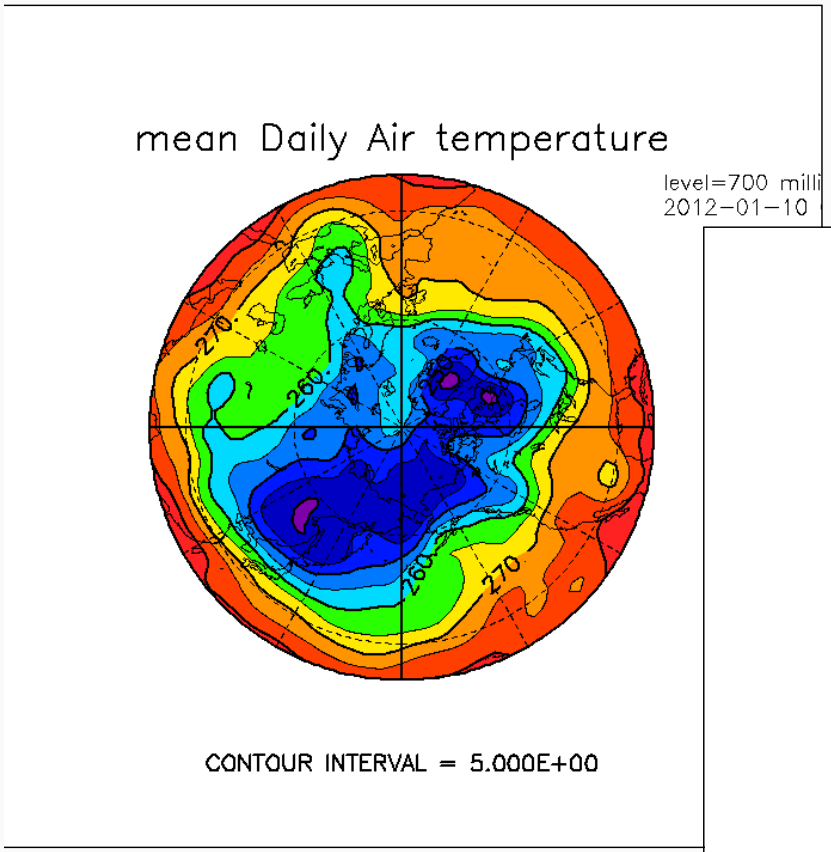
GGraph::tone gp と同じ



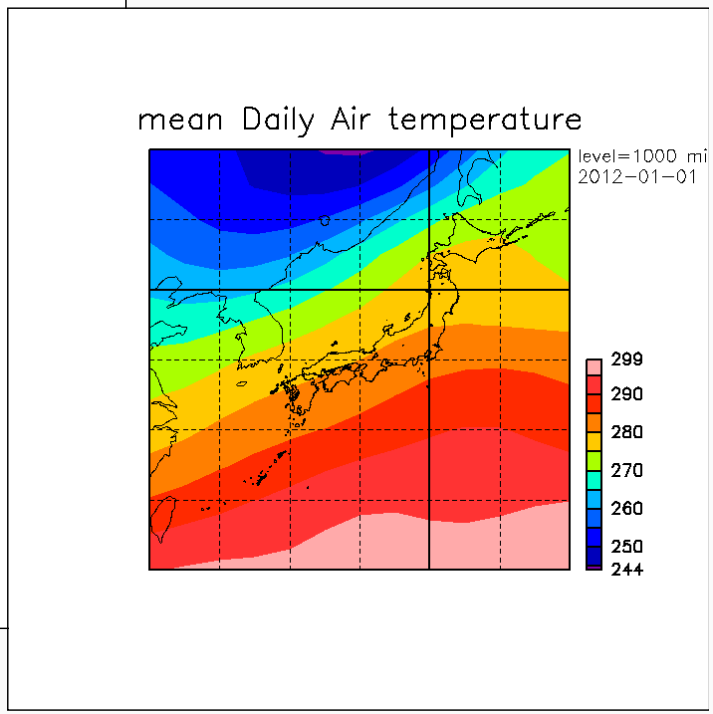
← 座標軸等データから読み込んで表示. 切り出し情報も欄外に

GPhys, GGraph(3)

ギャラリー（GGraphで書いた図）



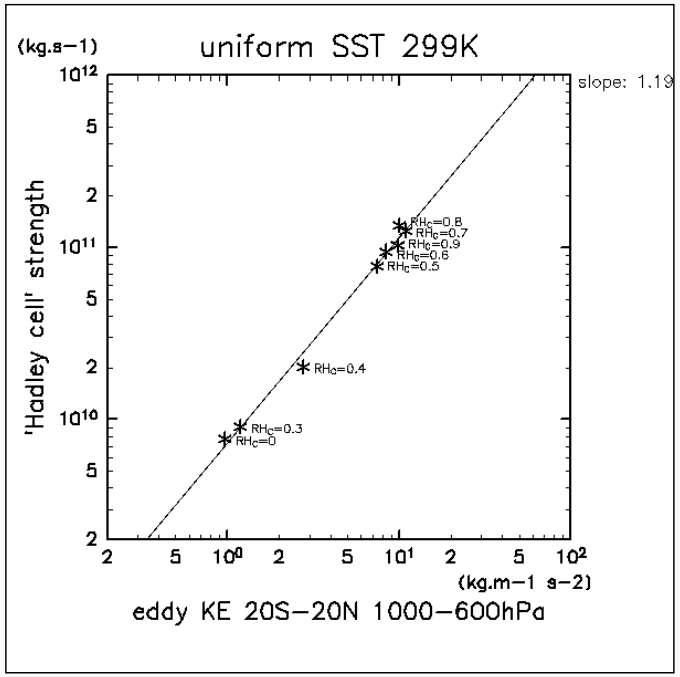
「GPhys, GGraphチュートリアル」より.
難易度: 易 (両図とも1行で書ける)



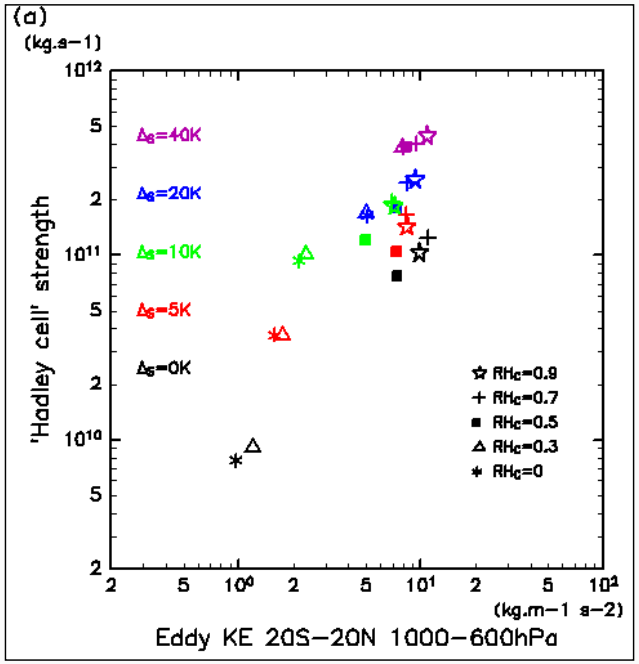
GPhys, GGraph(4)

ギャラリー（GGraphで書いた図）

マーカー描画と回帰直線
難易度:（各マーク横の文字列以外は）**易**（回帰直線はオプションでかける．欄外のスロープ表示も）



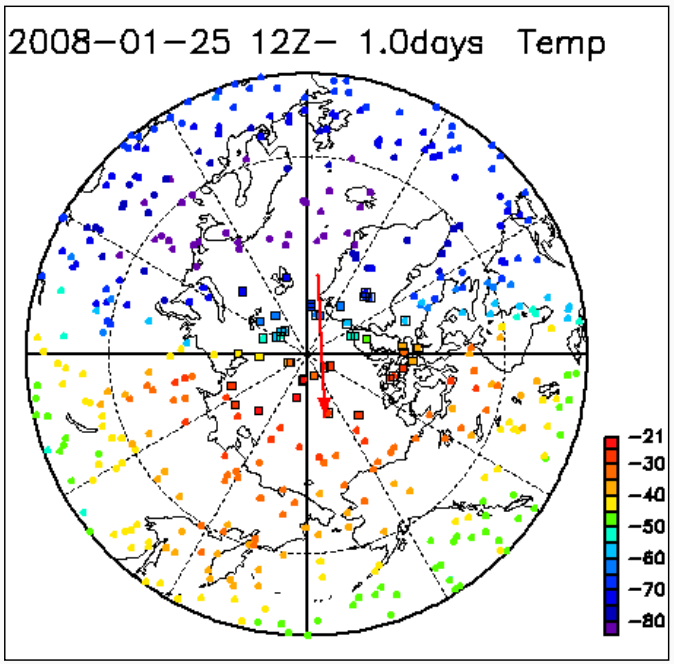
マーカー描画
難易度: **凡例表示がやや難**（ループで少しずつずらし表示．色等の連動や特殊文字や添え字の多用も初心者には難しいかも），**凡例表示以外は易**



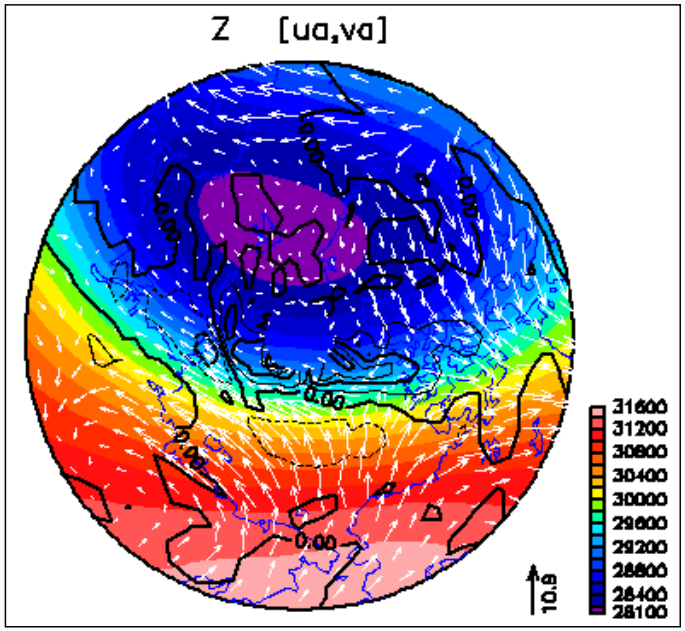
GPhys, GGraph(5)

ギャラリー（GGraphで書いた図）

色つき散布図
難易度：それ自体は易
(特殊な矢印等除けば)



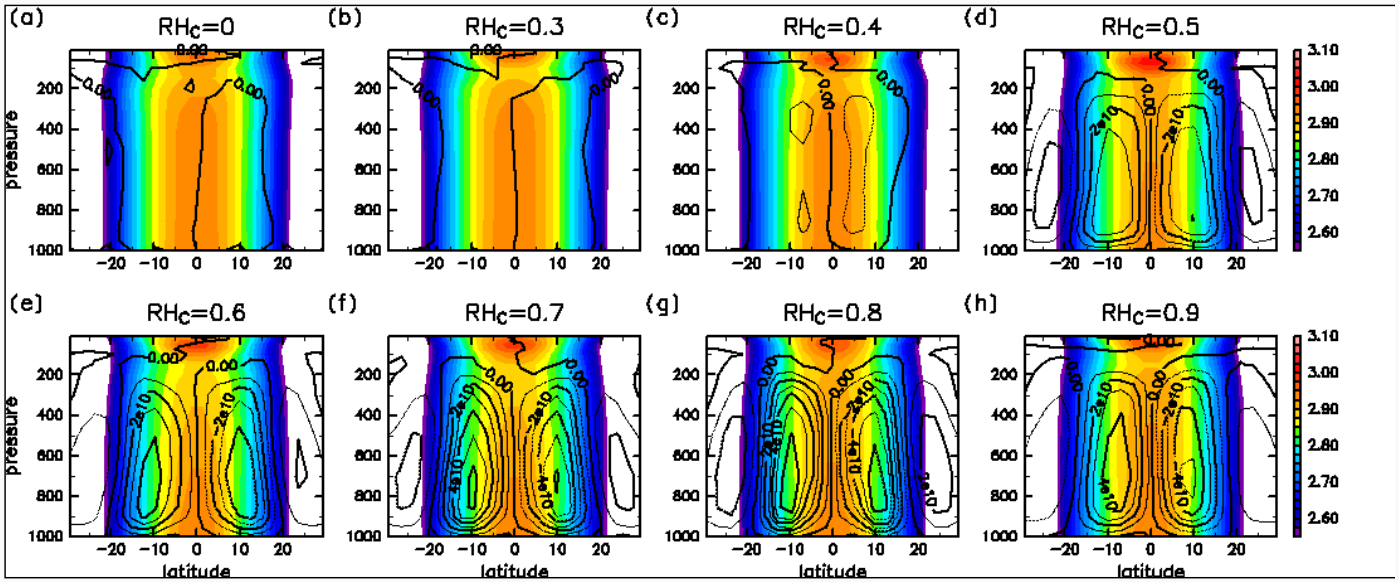
色塗り，コンター，ベクトル
難易度：中 (ベクトル矢印の色を白くするところは DCLコールが要る)



GPhys, GGraph(6)

ギャラリー（GGraphで書いた図）

折れ線とコンター
難易度：難（余裕を保ちつつ図を詰めるため，軸のラベルやカラーバーを共通化しているところが．普通ここまでやらない．なお，カラーバーは実は独立したフレームに表示してある – DCL.sldiv(“y”,5,2) なのです）



そして世界情勢は?

- Jupyter Notebook + iRuby?
  - Gfdnavi でやりたかった事が実現できる, 筈
    - RubyDCL は SVG 出力があるのですぐ乗る (?)
    - それ以外は...?
  - NArray ⇒ Numo::NArray はどうするか, などなど
- 「データサイエンス」の分野はどうなっているの?
  - そんなわけで, 今日は須藤さんと西田さんに御講演頂きます.