

気象学における情報処理*

堀之内 武*

1. はじめに

気象学の発展は情報処理技術の発展に多くを負っている。今日では、「研究」時間のかなりをパソコンの前で過ごす研究者が多いであろう。もはや計算機や通信ネットワークなしには、気象学は（いや多くの学問は）成り立たない。1982年の日本気象学会100周年記念レビューには、本稿に対応する記事はない。当時は必要なかったのであろう。この四半世紀に、計算機は長足の進歩を遂げた。メインフレームからUNIXワークステーションへ、さらに様々なOSのPCが広く用いられるようになった。そして、世界中の計算機を結ぶインターネットが出現し、研究に欠かせないものとなった。もはや、計算機での情報処理にまつわる事柄は、表に出すに値しない個別の「苦勞」の問題ではなく、学問の進歩を左右する共通の問題として、積極的に情報を共有すべきものとなった。

気象学にかかわる情報処理は多岐に渡る。そのうち重要なものとして、気象・気候モデリングやデータ同化があげられる。しかし、これらは気象学そのものの一部とも言え、独立に稿をあてるのにふさわしい。本稿では、気象学を支えるそれ以外の要素に光をあてたい。紙数が限られるため、気象学で用いられるデータの処理や通信に関わる問題に絞って書く。以下では、データの解析・可視化・形式・通信・データベース・検索などについて書き、最後に周辺分野として地理情報システム (GIS) に触れ、簡単に展望を述べる。

2. データ解析・可視化

観測でも数値モデリングでも、得られるのは数値データである。10年以上前までは¹⁾、データ解析においてもFORTRAN全盛であった。可視化にはNCARGやDCLなどのライブラリが用いられた。い

ずれも気象業界で「内製」されたライブラリであり、地図投影や欠損値処理など、地球科学の事情が採り入れられている。これらは現在でも、多くの可視化ツールの下請として活用されている。次いで、GrADS²⁾に代表される、文字ベースユーザインタフェース (CUI) で対話的に可視化が行えるツールが広まった。GrADSは、対象とするデータの座標を時間と空間の4次元に限ることで、手軽さを実現した成功例である。一方で、IDL³⁾やMatlab⁴⁾などの商用の汎用データ解析可視化ツールも、10年ほど前から広く使われるようになった。いずれも、可視化や数値計算のライブラリを、対話利用も可能なインタプリタ型のプログラミング言語で利用する。GrADSほど手軽でない反面、自由度が高く凝った処理も可能である。以上いずれもFORTRANでプログラムを書くより処理は遅いが、現在はデータ解析においてはCPUコストより人手のコストのほうが問題である場合が多い。なお、筆者自身は10年前にIDLを使うようになった。

最近ではグラフィカルユーザインタフェース (GUI) の解析可視化ツールも広まっている。良くできたGUIツールは初心者でも直感的に使えるが、多種多量のデータ処理をする「プロ」にはむしろ面倒なことも多いであろう。しかし、クイックルックには便利であるし、3次元可視化を行う場合、画面を見ながら視点等を操作できるGUIはむしろ必須である。3次元可視化の初期のツールとしてはVis5D⁵⁾が挙げられる。近年、より統合的なGUIツールとしてUCAR UnidataのIDV⁶⁾が開発された。IDVは、1~3次元の可視化機能を持ち、多くのファイル形式をサポートし、様々なプロトコルでネットワーク越しにもデータにアクセスでき、IDV同士の通信機能も持つなど、非常に高機能である。

再び話をプログラミング/CUIでの解析・可視化に戻す。筆者自身は、最近ではCUIでも使えるインタプリタ型のオブジェクト指向言語Rubyを使っている。次節に述べるように、オブジェクト指向言語の利点

* Information technology and meteorology.

** Takeshi HORINOUCI, 京大大学生存圏研究所.

© 2007 日本気象学会

は、汎用なプログラムが作りやすく、コミュニティでのソフトウェア資産構築に適するということである。まだ Ruby の利用者は限られるが、地球流体電脳倶楽部のサーバ⁷⁾に資源が集積されつつある。

2.1 データ解析のための言語論

かつて筆者は Fortran から IDL に移行したが、満足できなかった。IDL は手続き型言語である。オプション引数をサポートするなど、データ解析で使いやすい仕様もある。しかし、IDL (Fortran でも) で作るデータ解析プログラムに汎用性を持たせるのは容易ではない。ここで言う汎用性とは、内容的に同じことをするプログラムは使い回したいといったことであり、データの形式や構造を越えて「したいことの内容」に基づいたプログラムが容易に作れることである。そうなれば、ある人があるデータの処理用に書いたプログラムが、他の人が異なるデータを処理するために使えるであろう。汎用性こそが、データ解析プログラムを、個別に使い捨てられるものから、コミュニティの研究資産に昇華させる鍵である。(それだけでは足りず、資源の維持・流通機構も必要である。)

このような汎用性を実現するには、オブジェクト指向言語が適している。オブジェクト指向は、プログラムの再利用性を高める技術である⁸⁾。そのための重要な仕組みに、カプセル化と多態性がある。カプセル化とは、オブジェクト (プログラム中のデータの単位、構造体のように複合的なもの。1つの変数で名付けられるものと考えてよい) をブラックボックス化することで、プログラムがデータの内部構造に依存してしまうことを避けることである。多態性とは、オブジェクトに対する操作をオブジェクト側に定義することで、たとえ具体的な詳細が異なっても、目的が同じ命令なら同じ名前が使えることである。例として、ファイル中の 3 次元データ (たとえば Temperature) からある高度断面を読むことを考えよう。この際、入力データとして、NetCDF ファイル中の変数、GRIB ファイル中の変数など、複数種が考えられるので (3 節を参照)、手続き型言語においては、それぞれの種類に応じて名前の異なる読み出しプログラム (関数やサブルーチン) を定義し、それを呼ぶプログラムの側でも、ファイル形式が何であるかを意識した記述が必要である。しかし、オブジェクト指向言語では、各々のファイル形式に対応して変数の「型」(オブジェクト指向の用語ではクラス) を定義し、その中で同じ名前前で読み込み命令 (たとえば read) を実装することにより、呼び出し側では同じ命令、たとえば擬似コード

```
Temperature.read (*, *, Level 1)
```

でデータが読める。つまり、異なる種類のファイルへの対応を「型」の定義側に封じ込めることで、上位の

側ではファイル形式によらないプログラミングができるというわけである。

同様なことは Fortran 90 の関数の総称名定義によってもある程度可能である。しかし、Ruby のような純オブジェクト指向言語を使うのに比べれば、開発効率や出来るプログラムの見通しの良さには、雲泥の差が生じる。さらに、オブジェクト指向言語なら何でもよいわけでもない。手続き型などの言語でも開発効率が同じではないのと同様である。データ解析用には対話的に使えることが望ましい。そして、静的な型宣言文が必要なく、少ない行数で見通し良く書ければ開発効率が良いであろう。過去の FORTRAN や C の資源を組み込んで活用することも望ましい。以上を満たし広く使われている言語は、現在は Ruby や Python ということになる。なお、上で例として挙げたことを Ruby で実現する基盤ライブラリとして、筆者らは GPhys を開発、公開している⁹⁾。

3. データ形式

メインフレーム時代は、同じシステムにおいてすらバイナリ出力形式には多種類あった。まして、異なる計算機システム間のデータ互換性など問題外であった。当然ながら、これはデータ交換上の深刻な問題であった。気象のバイナリデータの互換性を実現するために、プラットフォーム依存性を廃し、さらにデータを記述する情報 (メタデータ) をも含む自己記述型データ形式の標準として、1980 年台半ばより、現業用には WMO の GRIB¹⁰⁾ が、アカデミックにおいては Unidata の NetCDF¹¹⁾ が登場した。GRIB は多数の外部テーブルに則ったヘッダをもつ水平 2 次元スライスデータの集合体であり、一方、NetCDF は文字列を含む記述的なヘッダをファイル全体で共有するなど、内部構造は異なるが、座標等のメタデータが機械的に解釈できるようになったことは画期的である。そのおかげで、2.1 節で述べた汎用データ処理の単位として、座標情報を含む数値データを無理なく取ることができる⁹⁾。なお、より簡便な自己記述データの構成法として、メタデータをテキストファイルで用意する GrADS の方式も広く用いられている。

衛星データでは HDF、HDF5 形式¹²⁾ が用いられることが多い。ただし、HDF は自由度が高すぎて、HDF を使うというだけでは NetCDF と同じレベルの自己記述性は持たない¹⁾。今後は地球観測データのためのメタデータ形式を定めた HDF-EOS 規約を使っていくなどにより、データの機械的解釈可能性を確保することを、データ提供機関には要望したい。

NetCDF は、ほどよい自由度と機械的解釈可能な自己記述性によって、気象データのネットワーク交換

性を高めることに大きく寄与した。しかし、不規則なサンプリングのデータには適用しにくい、データ圧縮がサポートされていない、などの問題もある。まもなく、後方互換性を維持しつつ自由度を大幅に高めた NetCDF-4 が正式リリースされるはずである。これにより、気象学のすべての数値データが収録可能となるであろう。自由度と機械処理性のトレードオフについては、適切な規約による解決が期待される。

4. データ通信・Web・通信と解析の融合

インターネット時代のはるか以前より、世界の気象機関は予報のためのデータを通信してきた。WMO のデータ網である GTS (Global Telecommunication System) は、1980 年台には X.25 というパケット交換プロトコルにより、そして近年では主に TCP/IP により構築されている。興味深いことに、その X.25 での GTS の黎明期に、大学への気象データの流れを確保しようとしたのが UCAR Unidata の設立経緯であると聞く[†]。Unidata は LDM (Local Data Manager) という、イベントドリブンでデータの通信と管理を行うソフトを開発し、それをもとに IDD (Internet Data Distribution) という気象観測データ配信のピア・ツー・ピアネットワークを構築した。現在、LDM は現業のデータ配信にも利用されている。

さて、インターネットの発展により、観測データ等の Web での公開が盛んになった。日本では東京大学生産技術研究所、高知大学情報、地球流体電脳倶楽部などが古くから行っている^{††}。Web でのデータサービスの基本は、データファイルを HTTP や FTP でダウンロードさせることであり、さらにクイックルック画像等が閲覧出来るようにしてある場合も多い。

Web でデータにアクセスできるようになると、今

度は、いちいちダウンロードしなくても、データの中を見たり可視化したいという要求が沸き起こることとなった。継続的には使わないかもしれないデータの試用は手軽にできて欲しいものである。大きなファイルの一部が必要な場合も、全体をダウンロードしないで済みたいであろう。最近、データの遠隔サービスを行うソフトウェアが登場し、広まってきた。

OPeNDAP¹⁴⁾は、そのようなサービスの 1 つであり、CGI として実装されている。HTTP 上での遠隔データアクセスプロトコルを定め、ブラウザやアプリケーションからのアクセスを受け付ける。NetCDF 等多種のファイル形式を抽象化する形でサポートし、データ/メタデータのテキスト化や任意の部分の読み出しなどができる。通信されるバイナリデータは独自の構造化が行われているため、専用 API をアプリケーションに組み込んで使う。NetCDF ライブラリに OPeNDAP 対応を組み込んだものも配布されており、NetCDF またはそれに準ずる構造のデータは、ローカルな NetCDF ファイルと同様にアクセスでき、必要な部分だけを読み出して解析や可視化できる。

一方、Live Access Server (LAS)¹⁵⁾は、ブラウザ上で可視化が行えるデータ提供サーバ構築ツールである。ローカルアクセスできるデータや、OPeNDAP で遠隔アクセスできるデータを登録し、Apache web サーバを通して、検索や可視化に供することができ、カスタマイズ性が高く、他の LAS サーバを「姉妹」登録して利用できるなどの優れた特徴を持つ。

最近、筆者らは Gfdnavi¹⁶⁾というツールを開発している。機能は LAS に似るが、常時運用するデータ公開サーバが作れるだけでなく、個人やグループがデータを整理、解析、可視化するのにも適したものとなっている。複数サーバを横断的に検索する機能や、SOAP プロトコルによる Web サービス API などの試験実装も始めている。さらに数値データとそれから得られる知見情報を一体的にデータベース化するなどの発展により、知識集約や共同研究のプラットフォームとしても使えるようになると見込まれる。

5. 周辺分野とのかかわり—地理情報システム

気象学隣接分野での情報通信技術の巨人は地理情報システム (GIS) である。GIS とは、地球上の位置を含んだ情報を扱う情報システムであり、官民に巨大な市場を持つ。今後、気象のデータや情報の扱いにおける GIS の重要性はますます高まるであろう。防災・気象情報の提供に適するのは無論、扱う対象を問わない点で、種々の学際的な取り組みにおける情報交換・共有のプラットフォームとしての役割も果たしうであろう。GIS の基本は「重ねて見ること」と言っても

[†] 「HDF を使う」というのは、プログラミングで言えば「構造体を使う」、文書であれば「XML を使う」、ということと同等である。HDF-EOS になると、XML 文書の特定のスキーマに相当する。なお、実は NetCDF でも座標の表現や共通の属性は、守らないことも可能な規約に依存している。NetCDF の自己記述性は、ほとんどのデータ提供機関が規約を守っていることによっているのである。さらに、CF 規約 (<http://www.cfconventions.org/>) という詳細な規約により自己記述性を高める努力もなされている。

^{††} 最近気象学会と気象庁が共同研究契約を結び、アカデミックセクタへのデータ供給役として期待される「気象研究コンソーシアム」が設立されたが、UCAR Unidata はそれを四半世紀前に実体組織として実現したのともいえよう。

良いが、大気とは全く異なる性質の対象にまたがって研究を行う場合、素朴に見ることがまずは糸口になることが多いのではなかろうか。

GISは、地図と地理学に根ざしている情報システムであり、標準的なデータモデルは気象学とは若干異なる。位置情報の精度は、気象学で通常想定されるものより桁違いに高い（建物の位置や敷地の境界を扱うのだから当然である）。緯度経度は、回転楕円体を用いた厳密な測地系により算出される。地図の座標系は、様々な測地系、投影法、地域毎に定義された原点などの組み合わせとして、整理・管理されている。一方、我々は緯度経度をいうとき測地系を指定することは稀で、暗黙のうちに球座標を仮定することが多いであろう。GISでは格子点データは画像と同様な「ラスター」データとして扱われ、何らかの座標系で等間隔であることが前提となる。数値モデリング上、不等間隔の格子を扱うのが普通である気象学においては、使いづらい点かもしれない。GISデータは基本的に水平二次元静的データであり、データ内部に鉛直軸・時間軸を持たないことも注意を要する。

GISで気象学の研究がすべて行えるかもしれないと考えるのは早計であるが、GISの出番は確実に増えている。最近まで、地球物理学者にとってGISは敷居が高いものであった。しかし、Google Earthの登場により、多くの研究者がGoogle Earth上でのデータ表示に取り組み始めた。このように、何かの登場で風景が一変することは、今後もあるであろう。

6. まとめと展望

本稿では、気象学におけるデータの情報処理にまつわるいくつかの話題を提示した。データ形式の共通化、自己記述化が進み、プログラミング技術と言語の発展も加わって、データ解析・可視化のソフトウェア環境は発展してきており、気象学の大きな手助けとなっている。さらに、Webベースのデータサービス、GISなどが分野横断的な協力で役立つようとしている。そのような技術的サポートがあってこそ実質的に行いうる研究もあるだろう。本稿で述べてきたような話には、「あれば便利だがなくても頑張ればなんとかなる」類のことが多い。しかし、各人の時間は増えない一方で、データは確実に増え科学は着実に進み扱う対象も広がっていく。それを乗り切るには道具は重要であるということは改めて強調してよいであろう。今後は、気象学・地球科学を支える情報通信技術を研究・開発する組織や人材育成プログラムも望まれると考える。

観測・実験データの有機的統合は多くの研究分野で求められている。気象学においては、客観解析がある意味究極の統合を行っている。それが強すぎるため、他のデータがおろそかにされてきた面もなくはないだろう。客観解析の影に埋もれがちな様々な生データを掘り起こし、横断的に利用できるようにすることは、気象学における情報通信技術関連の一つの課題であろう。インターネットの文書世界では、ハイパーリンクと検索エンジンによる巨大なWeb（網）が出現した。科学の数値データと情報についてもそのようなものが望まれる。本稿では、データは存在するものとして扱ってきたが、数値モデル等を用いたデータの生産も、当然研究プロセスに含まれる。外部データを検索・取得し、必要に応じて数値実験も行い、自前や外部のデータを横断的に解析するという一連のプロセスが統合的に行えるソフトウェアツールも望まれる。そのような試みとして、GRIDを用いる実験的なプロジェクトLEAD¹⁷⁾も始まっている。

謝辞

豊田英司氏、中島健介氏に有益なコメントを頂きました。

参考文献

- 1) 林 祥介, 1995: 天気, 42, 545-558.
- 2) <http://www.iges.org/grads/>
- 3) <http://www.ittvis.com/idl/>
- 4) <http://www.mathworks.com/products/matlab/>
- 5) <http://www.ssec.wisc.edu/~billh/vis5d.html>
- 6) <http://www.unidata.ucar.edu/software/idv/>
- 7) <http://www.gfd-dennou.org/>
- 8) Meyer, B., 1997: Object-oriented software construction—2nd ed., Prentice Hall, 1254pp.
- 9) <http://ruby.gfd-dennou.org/products/gphys/>
- 10) WMO, 2001: Manual on Codes, International Codes Part B (Geneva).
- 11) Rew, R. and G. Davis, 1990: IEEE Computer Graphics and Applications, 10, 76-82.
- 12) <http://hdf.ncsa.uiuc.edu/>
- 13) 沼口 敦, 1995: 天気, 42, 334-339.
- 14) Cornillon, P. *et al.*, 2003: Data Sci. J., 2, 164-174.
- 15) <http://ferret.wrc.noaa.gov/Ferret/LAS/>
- 16) 堀之内 武ほか, 2007: Proc. Data Engineering Workshop, DEWS2007, D2-8.
- 17) <http://lead.ou.edu/>