

自己記述的データ形式 gtool4 を用いた数値モデルの結果出力 -- gtool_history モジュール --

'gtool_history' module -- data output of numerical models using the self descriptive data format 'gtool4'

竹広 真一[1], 豊田 英司[2], 石渡 正樹[3], 林 祥介[4], 地球流体電脳倶楽部 Davis Project 林 祥介
Shin-ichi Takehiro[1], Eizi TOYODA[2], Masaki Ishiwatari[3], Yoshi-Yuki Hayashi[4], GFD Dennou Club Davis Project
Hayashi Yoshi-Yuki

[1] 九大・理・地惑, [2] 気象庁 数値, [3] 北大・地球環境, [4] 北大・理・地球惑星

[1] Earth and Planetary Sci., Kyushu Univ., [2] JMA NPD, [3] Graduate School of Environmental Earth Science, Hokkaido University, [4] Earth and Planetary Sci., Hokkaido Univ.

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

大量データの効率的な処理を目的とした自己記述的データ形式 gtool4 を数値モデルの計算結果出力として実装するために gtool_history モジュールを開発した。このモジュールは Fortran90 で記述されており、下位に用いている NetCDF データ形式を意識せずに名前ファイル・次元・変数・属性を定義・出力・参照できることが利点である。Fortran90 でかかれたメインプログラムに対しては最低限データの定義・変数定義・変数出力・終了処理を行う数行を追加するだけで gtool4 データ形式での出力が可能となる。Fortran77 で書かれた数値モデルから利用するためのインターフェースも用意している。

1. はじめに -- gtool4 データ形式

地球流体電脳倶楽部 davis プロジェクトでは、地球惑星流体力学の問題を数値的に取り組む際に必要となるデータの解析・可視化を効率的に行うためのデータ形式とツールの開発を試みている。

大気や海洋、あるいはマンツルの運動や惑星中心流体核など、地球や惑星での流体现象の解明と理解を目的とする研究において大きな技術的問題の一つとなっているのが、時空間データの加工と表示の効率化である。この問題点は、たとえば衛星観測から得られるグローバルで大容量の実データについてはもちろんのこと、大気大循環モデル等から得られる数値計算結果データなどを扱う際にも共通に顕在化する。つまり、これら大容量のデータを解析するときの最大の困難は、得られたデータを認識可能な形へ整形するために多大な労力を要するという点にある。このような困難は、いわゆる「シミュレーション」計算に限らず、相対的に単純な設定を扱う地球流体力学での数値実験でも同様に存在する。出力データの大きさは結局計算機のメインメモリーによって定まるので、扱うデータは同じオーダーのものである。さらに、地球流体力学の問題ではパラメータスタディーを積極的に行うので、計算結果を記述するための次元が大きくなる。空間 3 次元と時間 1 次元にくわえてパラメータの座標が必要となるからである。

大量のデータを効率的に処理するためにはデータを自己記述的にしておく必要がある。すなわち、数値実験の表題やパラメータ設定、あるいは観測時刻、あるいは平均処理や描画を行うために必要となる格子点の情報などをデータそのものが保持し継承していく必要がある。この目的のために我々は gtool4 データ形式を提案している (<http://www.gfd-dennou.org/arch/gtool4/conventions/>)。gtool4 データ形式は下位構造に Unidata の NetCDF データ形式 (<http://www.unidata.ucar.edu/packages/netcdf/>) を採用し、その属性名と使用法に関する規約として提案されている。

現在は下位データ形式(NetCDF)を gtool4 のレベルに抽象化する処理系の開発が一段落し、gtool4 データの表示・解析を行うツール群の整備とともに数値モデルからのデータ出力のテストを行っている段階である。

2. gtool_history モジュール

数値モデルの時間積分結果を等時間間隔に出力することを念頭においた gtool_history モジュールを用意した。このモジュールは Fortran90 で記述されている。

このモジュールの利点のひとつには、名前ファイル・次元・変数・属性を定義・出力・参照できることがあ

げられる。このことにより NetCDF データ形式を全く意識しなくて良いようになっている。Unidata が提供している NetCDF ライブラリを用いるとファイルや変数などを番号で識別し関数に引数として与えねばならず面倒である。また、Fortran90 で実装したことにより、読み書きする値の型や次元数が異なっても同じ名前のサブルーチンと呼ぶだけで変数や属性の出力・参照をおこなうことができるようになっている。サブルーチンが総称名で定義されているため、Fortran コンパイラが自動的に引数に対応したサブルーチンを呼び出すように変換して、それらが下位の NetCDF ライブラリを呼び出す。

gtool4 データ形式を出力するために必要となる最低限の操作は 1)gtool4 データの定義・作成, 2)変数定義, 3)変数出力, 4)終了処理, である。Fortran90 でかかれたメインプログラムに対してはこれに対応する処理を示す数行を追加するだけで gtool4 データ形式で出力できるようになる。Fortran77 で書かれたメインプログラムに対しても利用できるようにラッパーインターフェースも用意した。読み書きする値の種類によってサブルーチン名を変えねばならないといった、Fortran90 版の機能に比較しての制限はあるが、出力操作の軽減には十分である。

3. モデルへの実装

1 次元熱伝導モデル, および 2 次元熱対流モデルを例として実装実験を行った。そのテストプログラムと gtool_history モジュールの使用の手引を <http://www.gfd-dennou.org/arch/davis/> にて公開している。