

ISPACK と GTOOL4 を用いたスペクトルモデルによる簡単地球流体数値実験

Simple numerical GFD experiments by use of spectral models constructed with ISPACK and gtool4

竹広 真一[1], 石岡 圭一[2], 豊田 英司[3], 石渡 正樹[4], 林 祥介[5]

Shin-ichi Takehiro[1], Keiichi Ishioka[2], Eizi TOYODA[3], Masaki Ishiwatari[4], Yoshi-Yuki Hayashi[5]

[1] 九大・理・地惑, [2] 東大・数理, [3] 気象庁 数値, [4] 北大・地球環境, [5] 北大・理・地球惑星

[1] Earth and Planetary Sci., Kyushu Univ., [2] Graduate School of Mathematical Sciences, University of Tokyo, [3] JMA NPD, [4] Graduate School of Environmental Earth Science, Hokkaido University, [5] Earth and Planetary Sci., Hokkaido Univ.

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

1. はじめに -- 数式による理解から数値計算による理解へ

計算機が発達し身近になる以前では、「数式の変形」が地球や惑星などの自然現象を物理的にとらえるの主な手段であった。数式で表現されている物理法則の答えを求めるには手で解くほかなかった。解ける範囲は決してひろくはなかったが、その代わりに「数式の変形」によって誰しも解を得るまでの手順をたどることが原理的に紙と鉛筆だけでできるので、他者の結果を追体験して理解することは比較的楽であった。

計算機の発達した今日では支配方程式を数値的に解くことが普通である。このため解ける範囲は広がったが逆にその結果と理解を他者と共有することが難しくなってしまった。例えば数値モデルでなされた仕事の論文を理解するにはその図だけみて考えるのでは無理がある。結果を得るまでの同じ計算を行ってみたり、さらにはモデルの設定を変えてみるなどの経験を積む必要があるだろう。すなわち昔の「数式の変形」にあたるような追体験を計算機でおこなわねばならない。しかし、実際に数値計算を追体験するのは難しい。数値計算のためのモデルを最初から作るのは時間がかかりすぎるし、かといって出来合いのモデルをブラックボックスで走らせても結果の理解にはかならずしもつながらない。

このような観点から、われわれは地球惑星流体力学の数値実験を簡単に行うための標準的な一連のスペクトルモデルの開発と整備を現在進めている(地球流体電脳倶楽部スペクトルモデルプロジェクト, <http://www.gfd-dennou.org/arch/spmodel/>)。上に述べた観点から、

- ・プログラムソースは誰でも見ることができて、変更して使えなければならない。
- ・プログラムソースは可読性が高く理解しやすくなければならない。さらに再構築し変形することが簡単にできればなおよい。
- ・計算結果の描画や後処理が簡単に行えなければならない。

という指針を重点に置いている。今回は、ソースの可読性・可変性の観点から、スペクトル計算の Fortran77 ライブラリである ISPACK と Fortran90 を組み合わせたプログラミングを試みた。さらに結果の描画と後処理のために gtool4 形式の出力を組み込んでその有効性を吟味した。

2. ISPACK と Fortran90 の配列機能

ISPACK(<http://www.gfd-dennou.org/arch/ispack/>)は石岡の作成したスペクトル法による流体計算のための Fortran77 ライブラリ集である。このライブラリの各サブルーチンを Fortran90 によってくるむことを行った。すなわち、Fortran77 にはない Fortran90 の配列計算機能である、「要素別演算:配列の要素を指定せずに各要素に対する計算を行える」とこと「配列を返す関数」を生かして、格子点とスペクトル空間のデータ変換や空間微分などの基本的な配列関数をそろえた。これらの関数の名前は(関数の出力するデータ種類)_(機能)_(引数のデータ種類)といった形に統一されている。名前自体は冗長になるが、引数と関数の種類を名前からわかるようにすることで引数や出力の間違いを減らすことを狙っている。

これらの基本的な配列関数を用いることで、時間発展方程式の時間変化項以外の部分を数式の形そのままにプログラミングすることができる。支配方程式の形をそのままプログラムソースに反映させられるのでプログラムの可読性が向上していると期待される。さらには命名法の統一により関数の使い方が機械的になり、プログラムを

修正することも容易に行えるようになったと考えられる。

3. gtool4 形式による出力

gtool4 データ形式は地球流体電脳倶楽部 davis プロジェクト(<http://www.gfd-dennou.org/arch/davis/>)により定められた、多次元の大量データの可視化や解析を効率よく行うための自己記述的なデータ形式である。データそのものの値だけでなく、解析・可視化を行うために必要になる格子点の情報・表題・パラメータ設定などの付帯情報を保持できることが特徴である。もともとこのデータ形式は大気大循環モデルや衛星観測結果などの大規模なデータを意識したものであるが、PC でも動くような比較的小さな今回のモデルに対しても有用であった。

実際にプログラムから出力するには、豊田を中心として開発された gtool4/Fortran90 ライブラリの gtool_history モジュール(竹広ら, 2001 年合同大会 S4-P007)を用いることで簡単に実現できる。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。これらに対応する処理を示す数行をプログラムに追加するだけで gtool4 データ形式で出力できるようになる。gtool4 形式で出力された結果は gtvview コマンドで 1,2 次元のグラフや等値線で表示できる。四則演算や平均などの簡単な処理用コマンドも用意されている。