

惑星大気モデル DCPAMの開発現状

北海道大学/神戸大学

森川 靖大



目次

- はじめに : DCPAM とは?
- これまでの活動と今回の主題
- オブジェクト指向型のモジュール設計
- プログラムのテスト実行の半自動化
- モジュール雛形作成ツールの整備
- モデルの実装の現状

はじめに：DCPAM とは？

- **Dennou Club Planetary Atmospheric Model**
 - <http://www.gfd-dennou.org/library/dcpam>
- **いわゆる大気大循環モデル**
 - 力学 (プリミティブ方程式系) と物理 (放射、湿潤過程、鉛直拡散 etc.) を計算
- **DCPAM の狙い**
 - 地球型惑星だけでなく、外惑星や太陽系外惑星などの大気も計算する
 - dcmode1 プロジェクトのモデル群と以下のことができる
 - ◆ ソースコードの書き方が揃っている
 - ◆ 入出力データの形式が揃っている (= 解析・可視化ツールが共有できる)
 - ◆ モデルを構成する個々のプログラムの共有を容易にする
 - 数理ドキュメント、離散化ドキュメント、リファレンスマニュアルが付随する

これまでの活動と今回の主題

■ これまでの活動 (~ 2006)

- FORTRAN 77 で記述されていた地球流体電脳倶楽部 AGCM5.3 を参考に、Fortran 90/95 で開発
 - ◆ スペクトル演算に spml ライブラリを使用し、dcmodel プログラミングガイドラインに従う変数命名法の元でソースコードを記述することで、ソースコードを数式に近い形で記述
 - ◆ データ入出力などの煩雑なコードは gtool4 netCDF 規約に基づくデータの入出力 + F90 汎用ライブラリ gt4f90io に隠蔽
- Held and Suarez (1994) による乾燥大気計算の動作テスト

■ 今回の主題

- オブジェクト指向型のモジュール設計
- プログラムのテスト実行の半自動化
- モジュール雛形作成ツールの整備
- モデルの実装の現状

パラメータの管理に関する問題

- ここでの「パラメータ」とは
 - 解像度や物理定数やデータ入出力に必要な情報
- パラメータを単一ファイルで管理する際の問題
 - プログラムを加える毎に新たなパラメータが追加され、それらの整合性をとる必要がある
 - パラメータの削除を行う際には、取り外す前にモデル全体を調査しなければならない
- 以下のことが簡単にできると嬉しい
 - 物理素過程モジュール毎に独立してパラメータ管理
 - 計算する系に応じて複数のモジュール間でパラメータに一貫性を持たせることができる

オブジェクト指向型モジュール

■ 流体計算を行う数値モデルにおけるオブジェクト指向型のモジュール設計

(参考: Akin 2003: 『Object-Oriented Programming Via Fortran 90/95』)

- 各々の演算に必要なパラメータは各々のモジュールで保持
- パラメータはモジュール内で定義する構造体に格納
- 初期設定, 終了処理, 情報表示, 演算の手続を最低限用意
 - ◆ 必要なパラメータは初期設定手続を通して受け渡す
 - ◆ 時間発展する値のみ、演算手続に渡す
- 外部から参照可能なのは手続きのみ
- 外部へ公開する手続きに関しては, その第1引数に構造体を渡す

オブジェクト指向型モジュールの メリット・デメリット

8/14

■ メリット

- モジュール毎のパラメータ管理が行え、且つモデル内でのパラメータの一貫性も保持できる
 - ◆ 一貫性が必要な場合にはパラメータを引数として渡す
 - ▶ パラメータである引数に OPTIONAL 属性を与えておくことで、そのパラメータがモジュールで閉じている系で計算する場合には引数与えない
- どのモジュールがどのパラメータを用いているのか、初期設定手続きを見ればすぐ分かる
- 結果として、モジュールの着脱の際にモデル全体の探査を行う必要が無い

■ デメリット

- 引数としてコーディングを行う分、パラメータ管理ファイル一箇所に書くのに比べ、コーディングする分量が多い
- プログラムの着脱を頻繁に行う上では、メリットの方が大きいのではないかと (私見)

プログラムのテスト実行の半自動化(1)^{9/14}

■ 個別の演算プログラムに対してのテスト

- 初期設定手続に与えられたパラメタがモジュール内で正しく設定されているか「答えあわせ」する
- 各スキームがそれぞれ予期された計算をおこなっているか「答えあわせ」する

■ テストを行う上での問題点

- テストの実行が面倒
 - ◆ 可視化や解析の作業を定常的に行うのは面倒
- テストプログラム整備が面倒
 - ◆ 配列同士の比較など、コーディングにかかる手間が大きい

プログラムのテスト実行の半自動化(2)^{10/14}

■ テストプログラムのコードを簡素化

- dc_test モジュールによる多次元配列比較コード統一
 - ◆ Ruby の TestUnit クラスの真似
 - ◆ 組込み型変数・配列 (1~7次元) に関して 2 つの引数を比較 (大小、等しいかどうかのチェック) するためのサブルーチン AssertEqual 他を用意。実数の比較のための精度の指定が可能。
 - ◆ 値が異なる場合には両者の値と配列内での位置を出力して終了

■ テスト実行手順の定型化に貢献すると期待

- 各モジュールにテストプログラムを作成
 - ◆ モジュール (hoge.f90) に対して
テストプログラム (hoge_test.f90) と
テスト実行シェルスクリプト (hoge_test.sh) を作成
- Makefile を整備し、make test コマンドでテストを実行

モジュール雛形作成ツールの整備

- Fortran ファイルやシェルスクリプトを自動生成する Ruby スクリプトを準備
- 例えば積雲対流スキームの場合
 - モジュール
 - ◆ 積雲対流スキームを実際に計算するための Fortran ファイル
 - テストプログラム
 - ◆ 上記モジュールを読み込み、テストを実行する Fortran ファイル
 - テスト実行シェルスクリプト
 - ◆ 上記テストプログラムの実行を行うシェルスクリプト
 - テスト用 NAMELIST ファイル
 - ◆ 上記テストプログラムの実行を行う際の NAMELIST ファイル

モジュール雛形作成ツールの整備

- Ruby スクリプトを実行していくつかの問い合わせに答える
 - 寺田さんの CGI の方が格好いいなあ...

使用例

```
$ make template
```

```
ruby dcmode1_f90sample_maker.rb -E module
Input Module name : phy_cumulus
Title of module (for English) : Cumulus scheme
Title of module (for Japanese) : 積雲スキーム
Input basename [PhyCumulus]:
Input arg_type [PHYCUM]:
Input arg_keyword [phy_cum]:
Input Your name [unknown]: Yasuhiro MORIKAWA
Input Copyright [GFD Dennou Club]:
```

```
Message: phy_cumulus.f90 is generated ... done.
Message: phy_cumulus_test.f90 is generated ...
Message: phy_cumulus_test00.nml is generated ...
Message: phy_cumulus_test.sh is generated ... d
```

雛形生成
Ruby スクリプト

入力項目

モジュール
phy_cumulus.f90

テストプログラム
phy_cumulus_test.f90

NAMELIST ファイル
phy_cumulus_test00.nml

テスト実行シェルスクリプト
phy_cumulus_test.sh

モデルの実装の現状

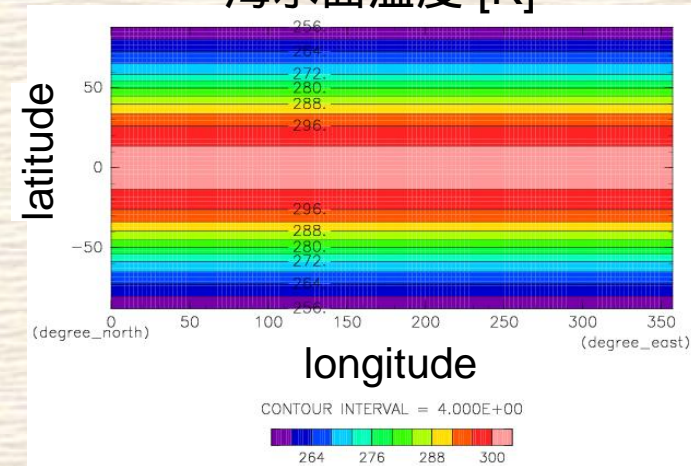
■ 以下の物理過程をモジュール化して実装

- 湿潤対流調節
- 大規模凝結
- 放射過程 (4色バンドモデル)
- 鉛直拡散 (Mellor Yamada, Level 2)
- 地表面フラックス (バルク法)

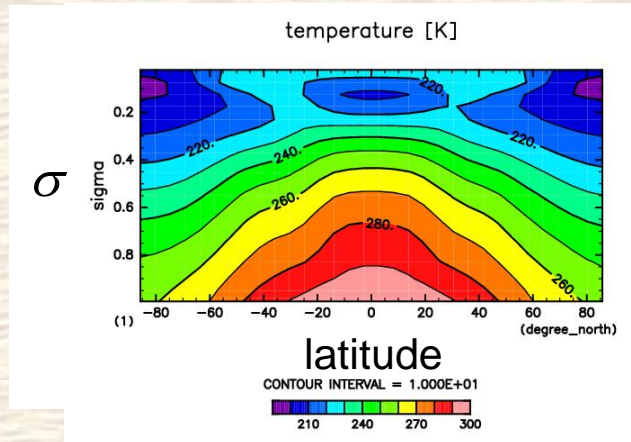
■ 水惑星実験

- AGCM5 を参照解として物理過程の動作チェック
- SST 分布 (Hosaka et al, 1998)

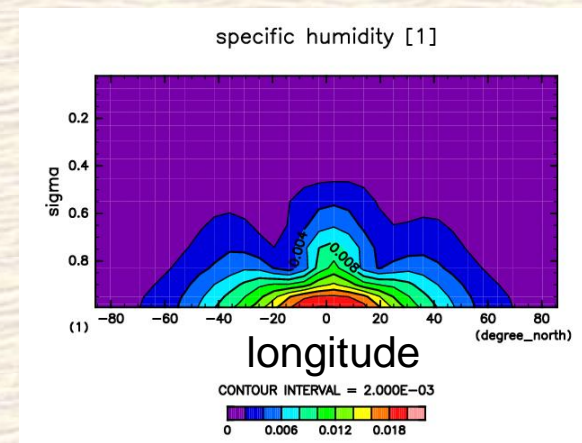
海水面温度 [K]



温度 [K] 東西平均 (450-500 日平均)



比湿 [kg kg-1] (450-500 日平均)



まとめ

- オブジェクト指向型モジュールによる可変性の向上
 - モジュール単位でのプログラムの着脱が用意に
- プログラムのテスト実行の半自動化
 - デバッグの助けになれば
- モジュール雛形作成ツールの整備
 - オブジェクト指向型のモジュール導入の役に立てばいいけど..
- モデルの実装の現状
 - 水惑星実験による物理過程の動作チェック
- ToDo
 - dcmode1 プロジェクトとしてのプログラム構造の決定
 - ◆ DCPAM はプログラムが難しくなりすぎており、他のモデルへの適用が難しい(らしい)
 - ▶ Fortran 90/95 の技術を目一杯詰め込んでみましたからねえ...
 - ◆ 階層モデル群として揃えるには、どこに落ち着けるかを相談して決めねば
 - 数理ドキュメントと離散化ドキュメントの見直しと修正

参考文献

- Akin 2003: Object-Oriented Programming Via Fortran 90/95. Cambridge University Press, 360pp.
- Hosaka, M., Ishiwatari, M., Takehiro, S., Nakajima, K., Hayashi, Y.-Y., 1998: Tropical precipitation patterns in the response to a local warm SST area placed at the equator of an aqua planet. J. Meteor. Soc. Japan, 76, 289--305.
- 森川靖大, 石渡正樹, 堀之内武, 小高正嗣, 林祥介, 2007: RDoc を用いた数値モデルのドキュメント生成. 天気, 54, 185--190.
- SWAMP Project, 1998: AGCM5. <http://www.gfd-dennou.org/library/agcm5/>. 地球流体電脳倶楽部
- 竹広 真一, 小高 正嗣, 石岡 圭一, 石渡 正樹, 林 祥介, 2006: 階層的地球流体スペクトルモデル集 SPMODEL. ながれマルチメディア 2006.
- 竹広真一, 石岡圭一, 森川靖大, 小高正嗣, 石渡正樹, 林祥介, SPMODEL 開発グループ, 2004: 階層的地球流体力学スペクトルモデル集 (SPMODEL), <http://www.gfd-dennou.org/library/spmodel/>, 地球流体電脳倶楽部.