大気大循環モデル用力学コアの設計と実装実験：可変性と可読性の高いプログラムへの試み

Design and implementation tests of a dynamical core for a general circulation model with a flexible and readable source code

○ 森川 靖大, 北大理, 〒060-0810 札幌市北区北 10 条西 8 丁目, E-mail:morikawa@ep.sci.hokudai.ac.jp
石渡 正樹, 北大地球環境, 〒060-0810 札幌市北区北 10 条西 5 丁目, E-mail:momoko@ees.hokudai.ac.jp
高橋 芳幸, 北大理, 〒060-0810 札幌市北区北 10 条西 8 丁目, E-mail:yot@ep.sci.hokudai.ac.jp
小高 正嗣, 北大理, 〒060-0810 札幌市北区北 10 条西 8 丁目, E-mail:odakker@gfd-dennou.org
林 祥介, 北大理, 〒060-0810 札幌市北区北 10 条西 8 丁目, E-mail:shosuke@gfd-dennou.org

Yasuhiro MORIKAWA, Division of Earth and Planetary Sci., Hokkaido Univ., Sapporo 060-0810, Japan
Masaki ISHIWATARI, Graduate School of Environmental Sci., Hokkaido Univ., Sapporo 060-0810, Japan
Yoshiyuki O.TAKAHASHI, Division of Earth and Planetary Sci.,Hokkaido Univ., Sapporo 060-0810,Japan
Masatsugu ODAKA, Division of Earth and Planetary Sci., Hokkaido Univ., Sapporo 060-0810, Japan
Yoshi-Yuki HAYASHI, Division of Earth and Planetary Sci., Hokkaido Univ., Sapporo 060-0810, Japan

Aiming for an atmospheric general circulation model (GCM) with readability and flexibility, a dynamical core is newly designed and implemented. Readability and flexibility are expected to be increased by the use of advantageous features of Fortran90 and by proposing a programming style. The programming style is to help us imagine the corresponding equations of the original physical system. The implemented dynamical core is examined by the benchmark test for GCM dynamical core proposed by Held and Suarez (1994)[1]. According to the results of 1200 day integration with the horizontal resolutions of T21, T42, and T63, it is confirmed that the Hadley circulation and mid-latitudinal disturbances are well expressed.

1. はじめに

モデル設定の可変性とソースコードの可読性を合わせ持った大気大循環モデル (GCM) の姿を模索するべく, 新たにその力学コアの設計を行い, そのプログラム実装と試験計算を行った. 可読性の向上によりプログラムの編集のコストの削減が期待され, 可変性の向上により新たなプログラムの追加や既に組み込まれているプログラムの分離を容易にすることが期待できる. モデル設定の切り替えが簡単な GCM が提供されることにより, さまざまな惑星大気の条件に応じた数値計算の実行が容易になり, 比較惑星科学的な見地からの惑星大気構造の考察の進展が期待される.

これまでに可変性・可読性を重視して開発された GCM としては, AGCM5 [2][3] や FMS[4] が挙げられる. AGCM5 は東京大学気候システム研究センターの気候モデルのプロトタイプモデルと同一のモデルである. このモデルは FORTRAN77 の範囲で最大限の可読性と可変性を確保するべく, そのプログラミングスタイルに数々の工夫を導入したモデルであったが, FORTRAN77 という言語の制限から自ずと限界があった. 一方 FMS は米国地球流体力学研究所 (GFDL) の新世代統合気候モデルの基盤フレームワークとして位置付けられるソフトウェアである. このモデルは, 今のところ大気・海洋・氷床それぞれの領域を計算する部分そのものの交換や分離に関してはかなり容易になりつつあるが, 一方でソースコード自体の読みやすさ, 編集のしやすさに関しての工夫はそれほど進んでいない.

本研究では, 可読性と可変性向上のための工夫として, FORTRAN77 から大きく拡張された Fortran90/95 の機能を活用した, いくつかの試みを行った. 第 1 の試みは, モデル内部の階層化である. モジュール, 構造体といった機能を利用することで, プログラム間の関係をシンプルで明瞭なものにし, 個々のプログラムの可変性の向上を目指す. これについては 3 節で述べる. 第 2 の試みは, ソースコードの可読性を向上である. まず, 次元, 物理量, 時間ステップといった情報が一目で分かる変数命名規則を考案している. これについては 4 節で述べる. 次に, 配列を演算結果の値として返すことが可能になった Fortran90/95 の関数を利用することで, ソースコードの簡素化を試みている. これについては 5 節で述べる. そして, 上記の変数命名規則とソースコードの簡素化の具体例を挙げ, AGCM5 でのソースコードとの比較を行い, その可読性と可変性について検証する. これについては 6 節で述べる. また, 簡素なインターフェースを持つデータ I/O ライブラリに煩雑なコード部分を任せることで, GCM 本体の I/O コードを簡素化している. これについては 7 節で述べる. 第 3 の試みは, ソースコードの改変や読解に必須となるドキュメントの整備を容易にするための仕組みの考案である. ソースコードにドキュメントを埋め込み, ソースコードからドキュメントを自動生成することにより, その管理コストを下げる工夫を行っている. これについては 8 節で述べる.

上記のような様々な工夫を施したモデルが, 力学コアとして正しく動作するのかを確認するための, Held and Suarez (1994)[1] ベンチマークテストでの性能チェックを行った. これについては 9 節で述べる.

本研究のモデルは DCPAM (Dennou-Club Planetary Atmospheric Model) と名付け, インターネット上 (URL: http://www.gfd-dennou.org) に公開している.

2. モデルの仕様

力学コアの定式化, 離散化, 時間積分法に関しては, 従来の GCM で用いられている標準的なものを用いる. 数理モデルにはプリミティブ方程式を用いる. 具体的には, 力学過程の水平離散化にはスペクトル法を, 鉛直離散化には Arakawa and Suarez (1983)[5] のスキームを用いる. 時間積分には陽解法リープフロッグスキームを用い, 計算モードの増幅を抑えるため, Asselin (1972)[6] の時間フィルターを適用する.

スペクトル計算においては SPMODEL ライブラリ [7], および

ISPACK[8] を用いる. SPMODEL は実際の計算を ISPACK に依存しており, 実際に計算を行うというよりは Fortran90/95 による可読性の高い関数で ISPACK を包むような働きをしている. 入出力するデータの形式は, gtool4 netCDF 規約 [9] に基づくデータである. この規約は地球流体現象を念頭においた格子点データのための自己記述的データの規約で, 米国の海洋大気庁 (NOAA) の COARDS 規約 [10] や国立気象研究所 (NCAR) の CMS 規約 [11] との互換性も考慮した規約である. さらに, このデータ形式はプラットフォーム非依存で可搬性が高く, また惑星上の流体運動一般を念頭においたメタデータを付加することを規定しているため, 汎用性が高いという特徴を持つ.

3. プログラム構造の階層化

本研究の GCM はモジュール機能によってプログラムを個々の役割を分離し, 構造体を活用して情報隠蔽を実現することで, プログラム同士の依存関係をシンプルにして, 個々のプログラムの可変性の向上を図っている. また, ファイル名, モジュール名, サブルーチン名同士の命名規則を定めることでそれらの関係を容易に知ることができる. いくつかの例を Table.1 に示す. さらにモジュールごとの初期化の手順を必ず接尾に _init が付いているサブルーチンによって行うことで, モジュール単位での分離と交換を容易にしている. ファイル名, モジュール名, サブルーチン名の命名規則や, 初期化手順の統一などのアイディアは FMS における Fortran90 コーディングルール [12] を参考にしている.

Fig.1 は階層化構造の概念図である. 現在は大別すると実行プログラム, 座標軸やパラメタの設定部分とに分かれている. この基礎を元に, 上記の手法を用いることで, 今後改良する力学演算部分や新たに加えられる物理過程演算部分も階層的に分離することが容易になる.

	ファイル名	モジュール名	サブルーチン名
凡例	module.f90	module_mod	module_init, module_end, ...
力学演算モジュール	dynamics.f90	dynamics_mod	dynamics_init, dynamics_leapfrog,
時刻設定モジュール	time.f90	time_mod	time_init, time_progress,

Table. 1 An example of file names, module names, and subtoutine names of DCPAM

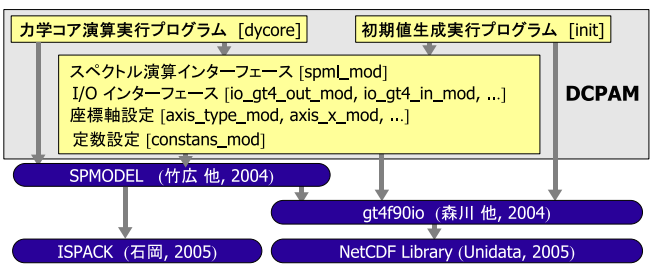


Fig. 1 The structure of DCPAM

4. 変数命名規則

次元, 物理量, 時間ステップといった情報が一目で分かる変数命名規則を考案・実装し, GCM の可読性の向上を図った.

GCM において変数命名規則を設け, 可読性を向上させようという工夫は AGCM5 を参考にしている. AGCM5 の変数命名規則は, FORTRAN77 の変数名 6 文字制限内で可能な限り変数の情報を格納しようと試みたものである. 例を挙げると, GAVOR は 1 文字目が Grid を省略したもので格子点データを表し, 2 文字目が After を省略したもので, 時間ステップの遅い方を表す. 残りの 3 文字が VORticity を省略したもので渦度を表す. WTDIV の場合は, 1 文字目は Wave を省略したもので球面調和関数展開されたスペクトルデータを表し, 2 文字目は Tendency を省略したもので時間変化を表す. 残りの 3 文字が DIVergence を省略したもので発散を表す.

本研究の GCM では, SPMODEL ライブラリの関数名をヒントに, Fortran90/95 で利用できる最大 31 文字の変数名を活用し, 変数命名規則を改良した. まず, 変数命名の形を xxx_yyy_zzz として規定する. xxx, yyyy, zzz にはそれぞれ, 次元・物理量・時間に関する情報を格納する. xxx では 1 文字が 1 つの次元を表すようにしている. そのため, この部分の数と文字種だけで, その量がどの次元に依存した何次元の量なのかが一目でわかる. 例えば, xy ならば xy 平面の 2 次元のデータであることを表し, wz ならば球面調和関数展開スペクトルと鉛直 z 空間との 2 次元データであることを表している. yyyy は物理量を表す. GCM でよく用いられる物理量に関しては既にいくつか規定をしている. zzzz はそのデータの時間ステップを表す. 具体的には, aa は $t+2\Delta t$, a は $t+\Delta t$, n は t, b は $t-\Delta t$, bb は $t-2\Delta t$ を表す. 以下に具体例を示す.

- x_Lon : x 方向 1 次元. 経度座標.
- xyz_Temp_b : xyz 空間 3 次元. 温度. $t-\Delta t$.
- wz_Vor_a : スペクトル + 鉛直方向 2 次元. 渦度. $t+\Delta t$.

この命名規則を用いたソースコードの例は 6 節で紹介する.

5. プログラム書法

GCM の数理モデルのコード部分に SPMODEL ライブラリで提供されるスペクトル演算用配列関数を適用することで, ソースコードを数式の形に近づけ, 可読性の向上を図った.

GCM は 3 次元の数理モデルに離散化を行うことから, 工夫無しではコーディングされる内容は元々の数理モデルとはかけ離れた形になってしまう. その数理モデル部分の可読性を向上させる工夫の 1 つとして Fortran90/95 で利用可能になった多次元データを返す関数を活用し, サブルーチンやループ文をなるべく利用せずに簡潔に数理モデルを記述することを試みた. 実際には, SPMODEL ライブラリで提供される関数を GCM に適用することを試みた. SPMODEL ライブラリは地球流体力学で扱われる空間 1 次元から 3 次元までのモデルを階層的に整備することを目指し, 数理モデルとしての数式のように簡単にモデルソースコードを記述するべく, 可読性の高いスペクトル演算用配列関数を提供している.

実際のソースコードへの適用例は 6 節で紹介する.

6. ソースコードの具体例

式 (1) および (2) を, AGCM5 的書法でコーディングした場合と, DCPAM 的書法, つまり 4, 5 節の工夫を用いてコーディングした場合の具体例を示し, その比較を行う.

$$\frac{\partial \zeta(t)}{\partial t} = \frac{1}{a\cos\varphi}\left(\frac{\partial\, VA(t)}{\partial \lambda} - \frac{\partial (v\cos\varphi)\, UA(t)}{\partial \varphi}\right) \qquad (1)$$

$$\zeta(t+\Delta t) = \zeta(t-\Delta t) + 2\Delta t \times \frac{\partial \zeta(t)}{\partial t} \qquad (2)$$

これらの式は GCM の数理モデルでよく用いられる, 渦度に関する時間発展の式を簡素化したものである. (1) は風速から渦度の

時間変化を求める式であり, (2) はその時間変化と 1 つ前の時間ステップの渦度から次の時間ステップ渦度を求める式である. λ は経度, φ は緯度, σ は圧力の地表面圧力に対する比, t は時刻, Δt は時間刻み, a は惑星変形, f はコリオリパラメータ, ζ は渦度である. また $UA = (\zeta + f)v\cos\varphi$, $VA = -(\zeta + f)u\cos\varphi$ であり, u は東西風速, v は南北風速である.

この式を AGCM5 的な書法で記述したのが Fig.2 である. 図中の (a) では ISPACK のサブルーチン SMTV2S を用い, VA と UA の格子点データ (赤字 GTUA と GTVA) から渦度の時間変化 $\partial\zeta(t)/\partial t$ のスペクトルデータ (黄色字 WTVOR) を取得している. つまり式 (1) の計算を行っている部分である. (b) では ISPACK のサブルーチン SMTG2S を用い, 渦度 $\zeta(t-\Delta t)$ の格子点データ (赤字 GBVOR) からスペクトルデータ (黄色字で書かれた WBVOR) を取得している. ここ行っているのは式 (2) の右辺第 1 項の格子点データからスペクトルへの変換である. (c) では (a) の結果スペクトルデータ $\partial\zeta(t)/\partial t$ (赤字で書かれた WTVOR) と, (b) の結果のスペクトルデータ $\zeta(t-\Delta t)$ のスペクトルデータ (赤字で書かれた WBVOR) に $2\Delta t$(白字の DELT2) を掛け合わせたものとの和をとっている. つまり, スペクトル空間において (2) の右辺から左辺を導出している. (d) では ISPACK のサブルーチン SMTS2G を用い, 渦度 $\zeta(t+\Delta t)$ のスペクトルデータ (赤字 WAVOR) を格子点データ (黄色字で書かれた GAVOR) に変換している. ここ行っているのは式 (2) の左辺のスペクトルデータから格子点データへの変換である.

サブルーチンの引数に関して補足すると, MMAX ~ KMAX は格子点数や切断波数に関する情報であり, WORK は作業領域, IT ~ ML は上記のサブルーチンの初期化ルーチンである SMINIT からの返り値である.

```
(a)  CALL SMTV2S
     I      ( MMAX , IMAX , IDIM, JMAX, JDIM, KMAX,
     I        GTUA , GTVA ,
     O        WTDIV, WTVOR,
     W        WORK,
     I        IT, T, IP, P, QUSDER, R, ML)

(b)  CALL SMTG2S
     I      ( MMAX , IMAX, IDIM, JMAX, JDIM, KMAX,
     I        GBVOR,
     O        WBVOR,
     W        WORK,
     I        IT, T, IP, P, QGS )

(c)  DO 7100  K = 1 , KMAX
       DO 7100  NM = 1 , NMDIM
         WAVOR( NM,K ) =  WBVOR( NM,K ) + WTVOR( NM,K )*DELT2
7100 CONTINUE

(d)  CALL SMTS2G
     I      ( MMAX, IMAX, IDIM, JMAX, JDIM, KMAX,
     I        WAVOR,
     O        GAVOR,
     W        WORK,
     I        IT, T, IP, P, QGS )
```

Fig. 2　An example of coding in AGCM5

一方で, SPMODEL ライブラリを利用した本モデルの書法で式 (1), (2) をコーディングしたものが, Fig.3 である. 図中の (e) では SPMODEL の関数 wa_Div_xya_xya を用い, VA と UA の格子点データ (赤字 xyz_UA_n と xyz_VA_n) から渦度の時間変化 $\partial\zeta(t)/\partial t$ のスペクトルデータ (黄色字 wz_VorTend_n) を取得している. つまり, 式 (1) の計算を行っている部分である. (f) では, まず SPMODEL の関数 wa_xya を用い, 渦度 $\zeta(t-\Delta t)$ の格子点データ (赤字 xyz_Vor_b) をスペクトルデータに変換する. そしてスペクトル空間で得られた $\zeta(t-\Delta t)$ と $2\Delta t \times \partial\zeta(t)/\partial t$ (赤字 wz_VorTend_n) の和をとった後, SPMODEL の関数 xya_wa で再び格子点データに戻している. すなわち, 式 (2) の計算である.

```
(e)  wz_VorTend_n =                                          &
       & wa_Div_xya_xya( xyz_VA_n , - xyz_UA_n ) &
       & / Rplanet

(f)  xyz_Vor_a    =                                          &
       & xya_wa( wa_xya( xyz_Vor_b )                        &
       &         + 2. * DelTime * wz_VorTend_n )
```

Fig. 3　An example of coding in DCPAM

本研究のモデルでの書法では, AGCM5 のそれに比較して, サブルーチンの call 文の使用や, 格子点数や切断波数などに関する情報の代入を行わない分だけ, コードが AGCM5 に比べて簡潔になっている. さらに, SPMODEL ライブラリの関数名と変数命名規則により, データの次元が変数もしくは関数の接頭詞で判別できるようになったため, これまでより直感的な把握が容易になった. なにより, 式 (1), (2) のように右に計算される物理量, 左に解という数式に近い形で記述できるようになっている.

もちろんこれら SPMODEL の関数を利用するため, 初期化サブルーチンで格子点や波数を指定する必要はあるが, AGCM5 においても初期化のサブルーチンは必要であるため, その点の可読性はこれまで問題にしていなかった.

配列を返す関数と変数命名規則は, 数理モデルの記述をこれまでより数式に近い形に近づけるのに有効であることが示され, ソースコードの可読性を高める 1 つの手法として期待できる. 今後は, 様々な数理モデル, および離散化に対しても可読性を高めることが可能なプログラム書法と変数命名規則について検討する.

7.　データ I/O 部分の簡素化

可変性と可読性を考慮する際には, データ I/O に関して 2 つのことが重要となる. 1 つは, データ入出力するファイルの Open, Close やファイル名・付加情報の設定などにより煩雑になりがちな I/O に関するソースコードの隠蔽である. この部分を上記で簡素化を試みている数理モデルのコードと分離・隠蔽しておくことは, ソースコードの可読性を向上させるのに大きく貢献する. 一方でモデルの可変性を考える場合, I/O 部分もそれに合わせて簡単に変更できることも重要である.

本研究では, gtool4 netCDF 規約に基づくデータを入出力する, 簡素なインターフェースを備えた I/O ライブラリ gt4f90io[13] を整備し, 利用することでこの 2 つの点の解決を図った.

gt4f90io は入出力に必要な煩雑なコード部分を受け持ち, シンプルな 5 つのサブルーチンに集約して Fortran90 の数値モデルに提供する. Fig.4 に gt4f90io を利用した際のデータ入出力のソースコードを示す. 以下, 図中の記号 (g) ~ (l) で示された部分は以下の処理を行っている. (g) gt4f90io の I/O 用モジュール gt4_history を呼び出すことを宣言する, (h) 初期化用サブルーチンである HistoryCreate を用いて出力ファイル名・ファイルの表題・次元等の情報を設定する, (i) HistoryAddVariable で出力する変数 (速度や温度など) を設定する, (j) HistoryGet で初期値を取得する, (k) HistoryPut で計算結果の値を出力する, (l) HistoryClose で終了宣言する. 基本的にこの 5 つのサブルーチンだけで gtool4 netCDF 規約に基づくデータの入出力が可能である. 複数の変数を出力する場合にはその数だけ HistoryAddVariable, HistoryPut を, 複数の変数を入力する場合にはその数だけ HistoryGet を, 複数のファイルに出力する場合にはその数だけ HistoryCreate, HistoryClose を呼ぶことになる.

```
(g)  program sample
       use gt4_history

     [型宣言] ......

(h)  call HistoryCreate( &
           file='sample.nc', title='gt4_history',     &
           ..., dims=(/'x','t'/), dimsizes=(/30,0/), &
           .......)

(i)  call HistoryAddVariable(                         &
           varname='temp', dims=(/'x','t'/), ....)

(j)  call HistoryGet(                                 &
           filename='init.nc',                        &
           varname='temp', array=xyz_Temp_b)

     [時間積分ループ]
           :
(k)        call HistoryPut(                           &
                varname='temp', array=xyz_Temp_b)
           :
     [時間積分ループ 終わり]

(l)  call History Close
     stop
     end program sample
```

Fig. 4　An example of input and output code by using gt4f90io

8.　RD によるドキュメント自動生成

プログラムの改変やソースコードの読解において必須である, プログラムの使い方や動作を解説するドキュメントをソースコードから自動生成することにより, その管理コストの低減を図った.

1 つ 1 つのプログラムの, その役割およびモデル全体との関係を知る手がかりとして, ドキュメントは広い意味で可読性・可変性の向上に貢献する. ただし, ドキュメントにその効果を発揮させるためには, ソースコードの更新がドキュメントに常に反映されている必要がある. 整備を簡単に行うための手法の 1 つがソースコードからのドキュメントの自動生成である. 実際にはソースコードにドキュメントを書き込んでしまい, その部分を抜き出すことでドキュメントを作成する. ソースコードとドキュメントを 1 つのファイルにしておくことで, 管理するファイルが減り, メンテナンスも容易になる. 他のモデルの例としては, FMS のドキュメント生成システムが挙げられる. しかし, その方法の欠点として, Fortran90 のソースコードの可読性が低下するという問題がある. 多くの場合, ドキュメントとして想定される HTML, XML, TEX などは, 文字の修飾や体裁を整えるためのタグやコマンドといった記号が数多く必要となり, Fortran90 のソースコードとしては非常に見づらくなる可能性が高いのである.

そこで本研究のモデルでは, ドキュメントとして埋め込む書式に RD (Ruby Documents) 形式 [14] を採用した. RD の特徴はテキスト形式に近く, 文字の修飾のための記号が少ない点である. この特長により, Fortran90 のソースコードの可読性の低下を防ぐことが可能である. また一方で HTML 形式などへの変換が容易に可能であり, Web 上に公開したりブラウザで参照する際に便利である.

RD を埋め込んだソースコードの具体例を Fig.5 に示す. 白字が RD 形式のコメントで黄色字がソースコードの部分である. そのソースコードから出力される HTML ファイルが Fig.6 である. ソースコードのコメントのうち, 行頭に “ = ” がついたものが見出しとして出力されている. また, サブルーチンの引数の情報に関しては, ソースとドキュメントに関して同様な編集を 2 度行わなくて済むよう, ソースコードがそのまま出力されるようになっている.

この試みによって, 個々のプログラムに関するドキュメントの生成のコストはこれまでよりも低下しつつある. 現在, プログラム同士の依存関係や目次に関しても自動生成を可能とすべく, 試行錯誤を行っている.

```
                                                        !=begin
!= Module module_name_mod : Sample module
!
! * Developers: Yasuhiro MORIKAWA
!
!== Overview
!
! module_name_mod の概要
                                                        !=end
module module_name_mod
  implicit none
                                                        !=begin
  !== Public Interface
  !
  private
  public :: module_name_init    ! subroutines
  public :: module_name_end     ! subroutines

  !== Procedure Interface
  !
  !=== Subroutine module_name_init : モジュールの初期化
  !
  ! NAMELIST を入力し、グローバル変数を allocate する。
  !
  subroutine module_name_init(inchar, outint)

    !==== Input
    !
    character(*)    , intent(in)  :: inchar
    !
    !==== Output
    !
    integer(INTKIND), intent(out) :: outint
                                                        !=end
  end subroutine module_name_init
end module sample_mod
```

Fig. 5　An example of Fortran90 source code in which comments written by RD format are embedded

Fig. 6　An example of HTML documentation automatically generated from Fig.5

9.　Held and Suarez (1994) によるテスト計算

力学コアの動作試験として, Held and Suarez (1994) の GCM 力学コアのベンチマークテストを行った. 水平拡散係数は Held and Suarez (1994) と同様, 最大波数の e-folding 時間が 0.1 日に

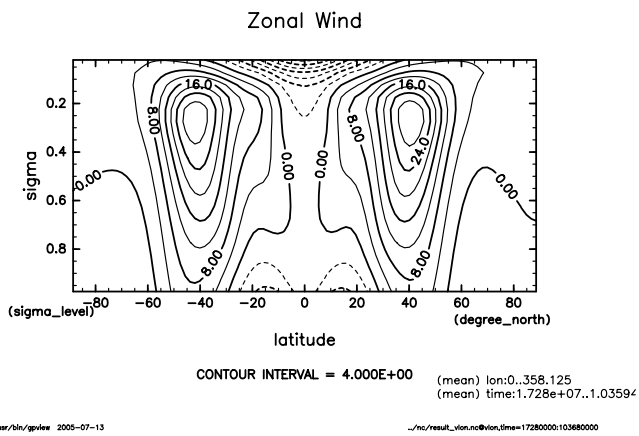
なるようにした. 水平解像度 T21, T42, T63 で, Held and Suarez (1994) と同様に 1200 日積分を行った. 計算時間を Table.2 に示す. 計算は NEC SX-6 で行った. 積分時間は 1200 日である.

Fig.7 (a) に T63 の場合の帯状平均東西風の 1000 日平均を示す. 中緯度にジェットが見られる. Fig.7 (b) に示す Held and Suarez (1994) の T63 のスペクトルモデルの場合とも非常に近い結果が得られた. ハドレー循環や中緯度擾乱が表現できることが確認された. 分解能を変えることで, ジェットの位置のずれが大きくなるもののほぼ同様な結果が得られる.

解像度	計算時間	Δt	総ステップ数
T21	47109.714831	480 sec	216000
T42	325994.434116	240 sec	432000
T63	1112413.637671	160 sec	648000

Table. 2 The computational time

(a)



(b)

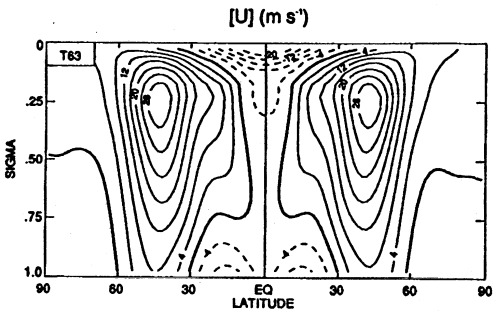


Fig. 7 The zonal-mean zonal wind produced by (a) T63 DCPAM and (b) T63 spectral model of Held and Suarez (from Held and Suarez(1994)). Both are 1000-day means.

10. 今後の予定

力学コアの改良として, 陽解法と簡単に変更できるように陰解法時間積分スキームを組み込む. また水平離散化, 鉛直離散化を簡単に変更可能にするため, これまでの経験を活かして新たな工夫を行う. 物理過程に関しても, 簡単なものから導入を行い, その可変性について検討を行う.

参考文献

(1) Held and Suarez, 1994: "A proposal for the intercomparison of the dynamical cores of atmospheric general circuation models." Bull. Am. Meteor. Soc., 75, 1825–1830.

(2) SWAMP Project, 1998: AGCM5. http://www.gfd-dennou.org/arch/agcm5/, GFD Dennou Club.

(3) 沼口, 1992: "熱帯における積雲活動の大規模構造に関する数値実験" 博士論文, 東京大学.

(4) GFDL, 2005: FMS: The flexible modeling system. http://www.gfdl.noaa.gov/~fms/.

(5) Arakawa and Suarez, 1983: Vertical differencing of the primitive equations in sigma coordinates. Mon. Wea. Rev., 111, 34–35.

(6) Asselin, R. A., 1972: Frequency filter for time integrations. Mon. Wea. Rev., 100, 487–490.

(7) 竹広, 他, 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.

(8) 石 岡, 2005: "ispack-0.70", http://www.gfd-dennou.org/arch/ispack/, 地球流体電脳倶楽部.

(9) 豊田, 他, 2000: "gtool4 netCDF 規約", http://www.gfd-dennou.org/arch/gtool4/, 地球流体電脳倶楽部.

(10) Cooperative Ocean/Atmosphere Research Data Service, 1995: "COARDS Convention". http://ferret.wrc.noaa.gov/noaa_coop/coop_cdf_profile.html .

(11) NCAR CSM, 199X: "NCAR-CSM Conventions". http://www.cgd.ucar.edu/cms/eaton/netcdf/NCAR-CSM.html .

(12) Balaji, V., 2002: The FMS Manual: A developer's guide to the GFDL Flexible Modeling System. http://www.gfdl.noaa.gov/~vb/FMSManual/FMSManual.html

(13) 森川, 他, 2005: "gt4f90io: gtool4 規約に基づく Fortran90 netCDF I/O ライブラリ", http://www.gfd-dennou.org/arch/gtool4/, 地球流体電脳倶楽部.

(14) Ruby Documentation Project, 2005: http://www.rubyist.net/~rubikitch/RDP.cgi?cmd=view;name=top.