

惑星大気大循環モデルの 設計と開発

- 階層モデル群のためのソフトウェア基盤整備 -

北海道大学 理学院 / 神戸大学 理学研究科

博士後期課程

森川 靖大

目次

- はじめに
 - 数値実験とは
 - 大目標：階層的数値モデル群の開発
 - 目標：惑星大気大循環モデルとソフトウェア基盤の開発
- ソフトウェア構造の基本設計の策定
- 変数などの命名法、サブルーチン、関数の利用法の策定
- データ入出ライブラリ gtool5 の開発
- 解説文書自動生成システムの開発
- まとめ

はじめに: 数値実験とは (1/3)

- 数値実験とは
 - ある物理モデルに則って現象を表現するプログラム (=数値モデル) を用いて実際の現象を模擬すること
- 利点
 - 解析的に解けない系においても、その答えとおぼしきものが得られる
 - 時間的、空間的制約が小さい
- 注意
 - 具体的な物理モデルが必要
 - 観測や室内実験などの参照解がないと苦しい
 - 計算資源にも限界がある

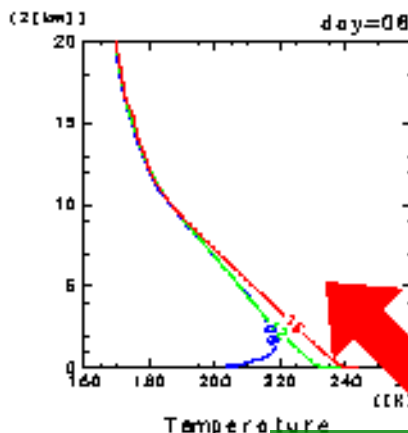
はじめに: 数値実験とは (2/3)

- 数値実験を行う際の手順
 - 物理モデルを組み立てる
 - どのような現象を考察したいか、どのような系を想定するのか
 - 数値モデルを組み立てる
 - プログラムするには、離散化が必要
 - 数値モデルをプログラムする
 - 数値モデルを動作させ、結果を得る
 - 結果を眺めて考察する
- 考察の際には「何を計算したのか (=物理モデル)」をよく把握しておくことが必要

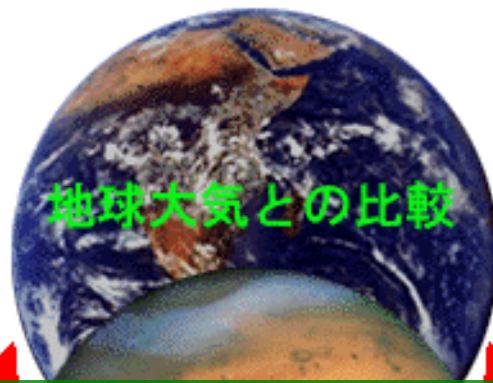
はじめに: 数値実験とは (3/3)

- 様々な (大気にまつわる) 数値モデル
 - 大気大循環モデル (3次元)
 - 力学 (静力学) + 放射 (簡単化)
+ 積雲・地表面過程 (パラメタリゼーション) + ...
 - 雲解像モデル (2次元)
 - 力学 (準圧縮)
+ 雲微物理・放射・乱流混合過程 (パラメタリゼーション) + ...
 - 放射対流モデル (1次元)
 - 放射 + 鉛直対流 (簡単化)
 - エネルギーバランスモデル (1次元)
 - 南北エネルギー輸送 + 放射 (簡単化)
- 1つで十分なモデルはない
 - 簡単化する部分は、他のモデルによってその正当性を保障する必要がある

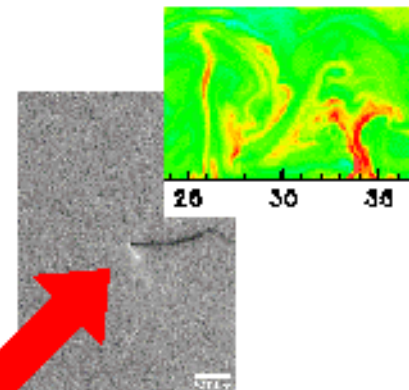
はじめに： 大目標 (1/4)



放射対流モデルによる鉛直大気構造の計算

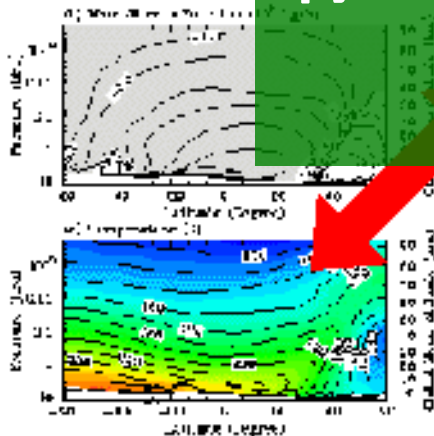


地球大気との比較



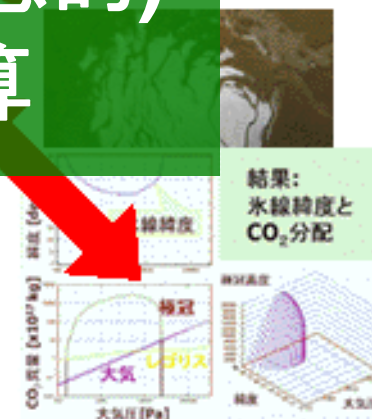
雲解像モデルによるゲストデピルの再現

様々な (現実的・仮想的) 惑星大気の数値計算



GCM による大気大循環の研究

過去の気候のシミュレーション



EBM による極冠の成長・後退の計算

- 様々な惑星大気の構造を考察するための階層的数値モデル群の開発

はじめに：大目標 (2/4)

- 目標とする階層的数値モデル群とは？
 - GCM (3次元)、雲解像モデル (2-3次元)、放射対流モデル (1次元)、エネルギーバランスモデル (1次元)、etc...
 - 物理モデルの把握と比較を手軽におこなうことができる
 - 個々の物理プロセスの着脱や共有を手軽に行える
 - 計算結果のデータをモデル間で手軽に比較できる
- どのような数値実験に有用なツールか？
 - 比較惑星科学的な考察のための数値実験
 - 様々なパラメタ (半径, 重量, 太陽定数, 軌道要素, 大気組成など) 空間での数値実験によって、惑星環境の多様性を考察
 - 観測データを得ることが困難な惑星大気の推測
 - 複数種類の数値モデルによって正当性の保障を行いつつ計算

はじめに: 大目標 (3/4)

- 問題点

- 物理モデルの比較が面倒

- モデル毎にソースコードの書法 (変数などの命名法, サブルーチンや関数の使い方) が異なるため

- 物理プロセスの着脱や共有が面倒

- ソフトウェア構造 (プログラム間のインターフェース) がモデル毎に異なるため
- データ入出力、並列化に関する部分が特に異なってくる
- プログラムの概要・使用方法を記した解説文書を継続的に整備するのは面倒
 - 多数のプログラムの全ての読破とは困難なため、解説文書は必須

- 結果の比較が面倒

- モデルごとに出カデータの形式が異なるため

はじめに：大目標 (4/4)

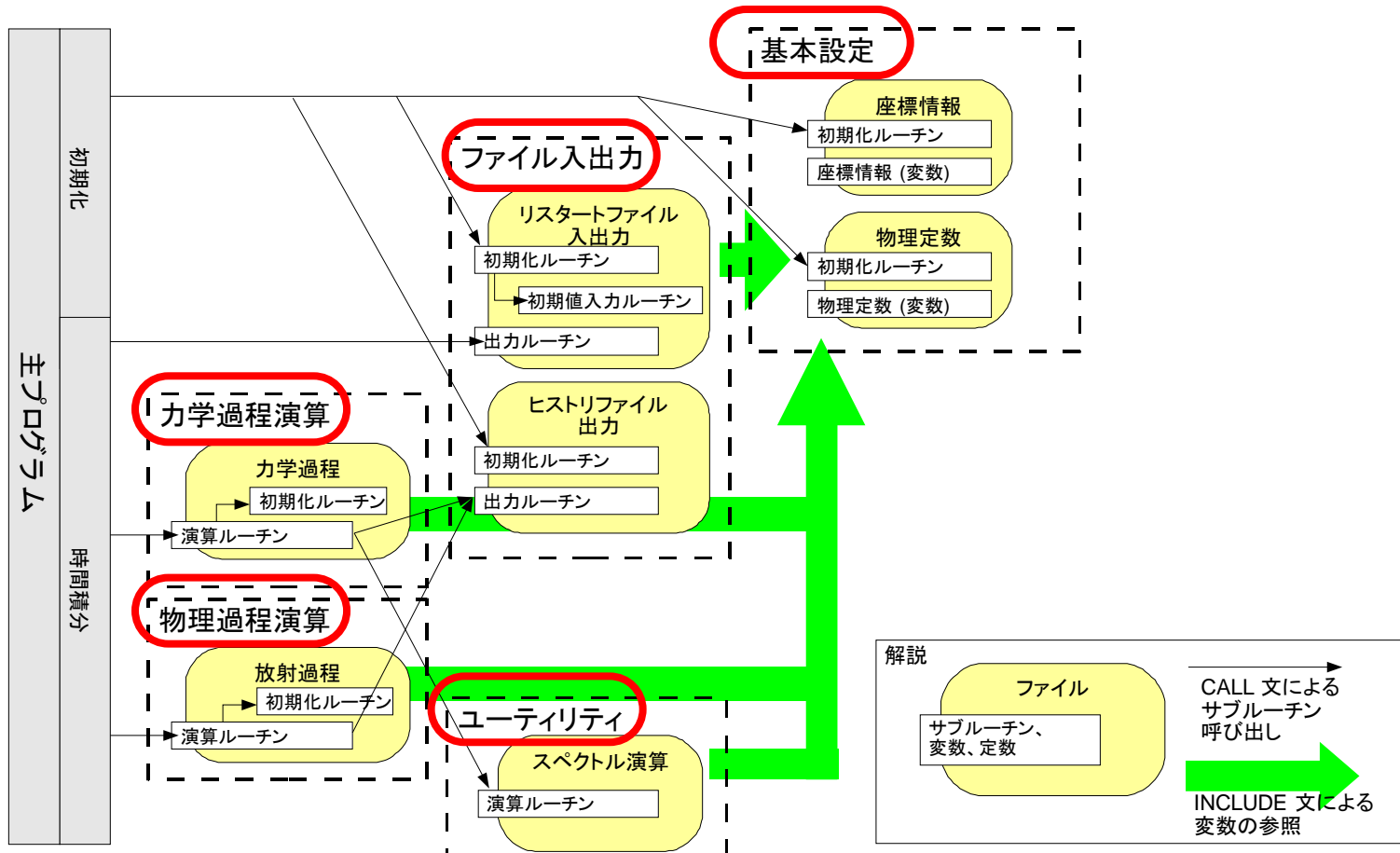
- 解決方策：
階層モデル群で共通利用できるプログラミング
書法の策定とツール／ライブラリの開発
 - モデルのソフトウェア構造
 - 変数などの命名法、サブルーチンや関数の使用法
 - 入出力するデータの構造
 - データ入出力に関する書法とライブラリ
 - プログラムの解説文書作成ツール
 - 並列化に関する書法

はじめに：目標

- 階層モデル群のプロトタイプとなる
大気大循環モデル (DCPAM; Dennou Club Planetary Atmospheric Model) の設計と実装、および
階層モデル群で共通利用するソフトウェア基盤開発
- 開発方針
 - AGCM5 (Fortran77 で記述された GCM) を参考に、Fortran 90/95 でモデルをスクラッチアップ
 - 他のプログラミング言語やプログラム技法の活用
- 課題
 - ソフトウェア構造の基本設計
 - 変数などの命名法、サブルーチンや関数の使用法
 - 入出力するデータの構造
 - データ入出力に関する書法とライブラリ
 - プログラムの解説文書作成ツール

AGCM5 のソフトウェア構造

- 物理プロセス
 - 力学 (静水圧近似・スペクトル法)
 - 物理 (放射・雲・乱流等)
 - 簡単なスキームの地球大気用の物理過程



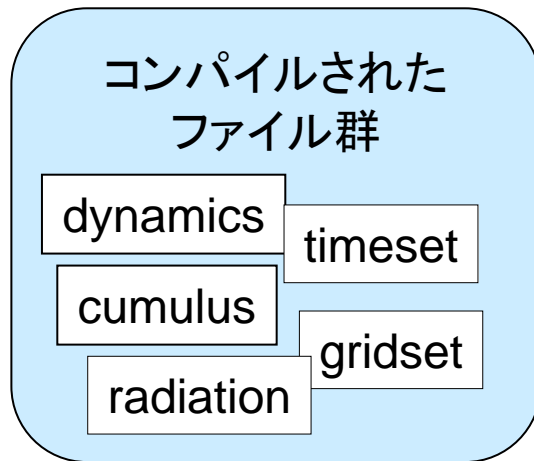
従来のソフトウェア構造の問題点

- サブルーチン、関数、変数、定数等の名称の重複
 - Fortran 77 では、言語要素の名称が一意でなければならない
 - 似て非なる物理量の扱い
 - 各プロセスによって異なる物理量に同じ名前が張り付く
 - プログラム数が増えてくると一人では掌握しきれない
 - グローバルな名称をつけるのは現実的に困難
 - たとえ付けたとしても、多くのものは非常に長くて使いづらい名称になる可能性が高い
- 「力学」と「物理」の切り分け
 - GCM 特有のもので、階層モデル群の共通設計としてはイマイチ

Fortran 90/95 機能の導入 (1/2)

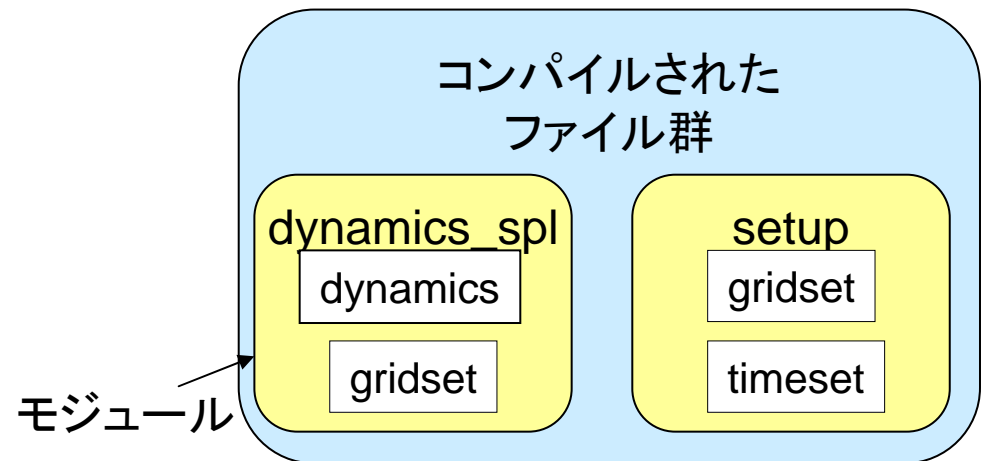
- モジュールによる言語要素の整理

Fortran 77



サブルーチンや関数が全てフラットな関係 (名称の重複は許されない)

Fortran 90



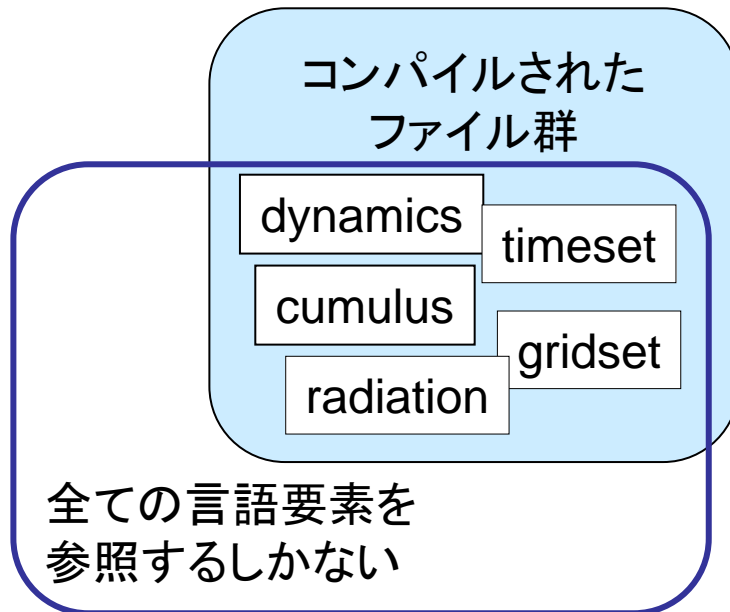
サブルーチンや関数をモジュールでくるむ (全体として、名称の重複が可能)

(※ モジュールの名称は重複できない)

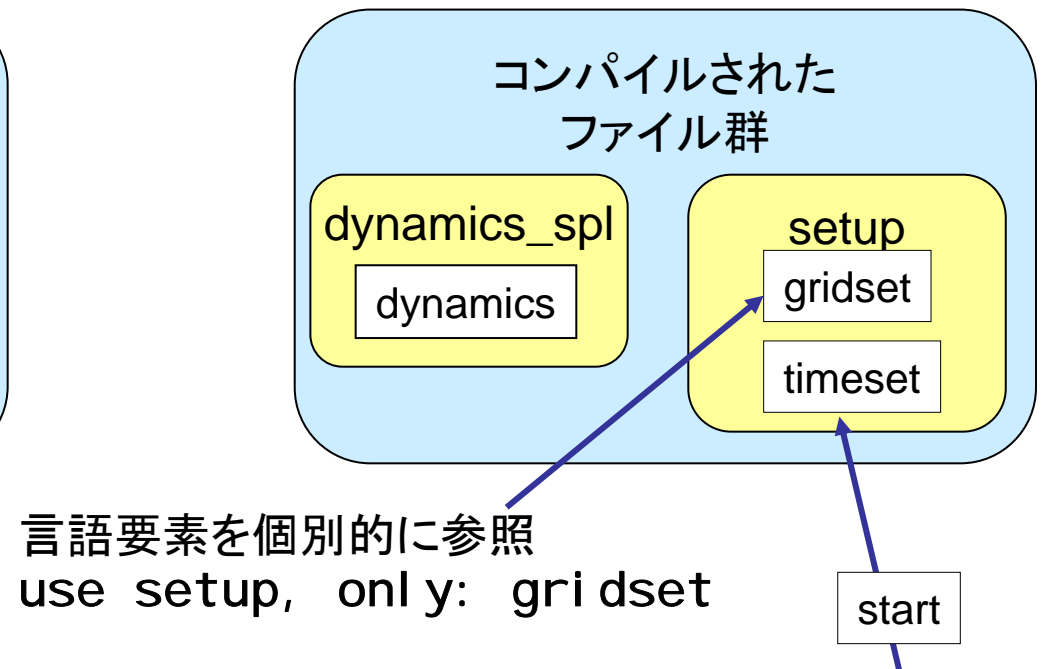
Fortran 90/95 機能の導入 (2/2)

- USE 文および ONLY 節による言語要素の個別参照

Fortran 77



Fortran 90



名称を変更して参照することも可能
use setup, only y: start => ti meset

さらに進んだソフトウェア構造を 目指して？ (1/2)

- オブジェクト指向プログラミング (OOP)
 - 変数と手続きとを「オブジェクト」として一まとめに管理する手法
 - 個々の「オブジェクト」は独立しており、個別に設定・使用される
 - Java や Ruby などオブジェクト指向言語と呼ばれるものは、OOP されることを前提に設計・実装されている
- Fortran 90/95 における OOP
 - モジュールと構造体の活用によって一応は実現可能
(Akin (2003) 『Object-Oriented Programming Via Fortran 90/95』)
 - 構造体: ユーザが定義できるデータ型
 - Java や Ruby のようにスマートには記述できない

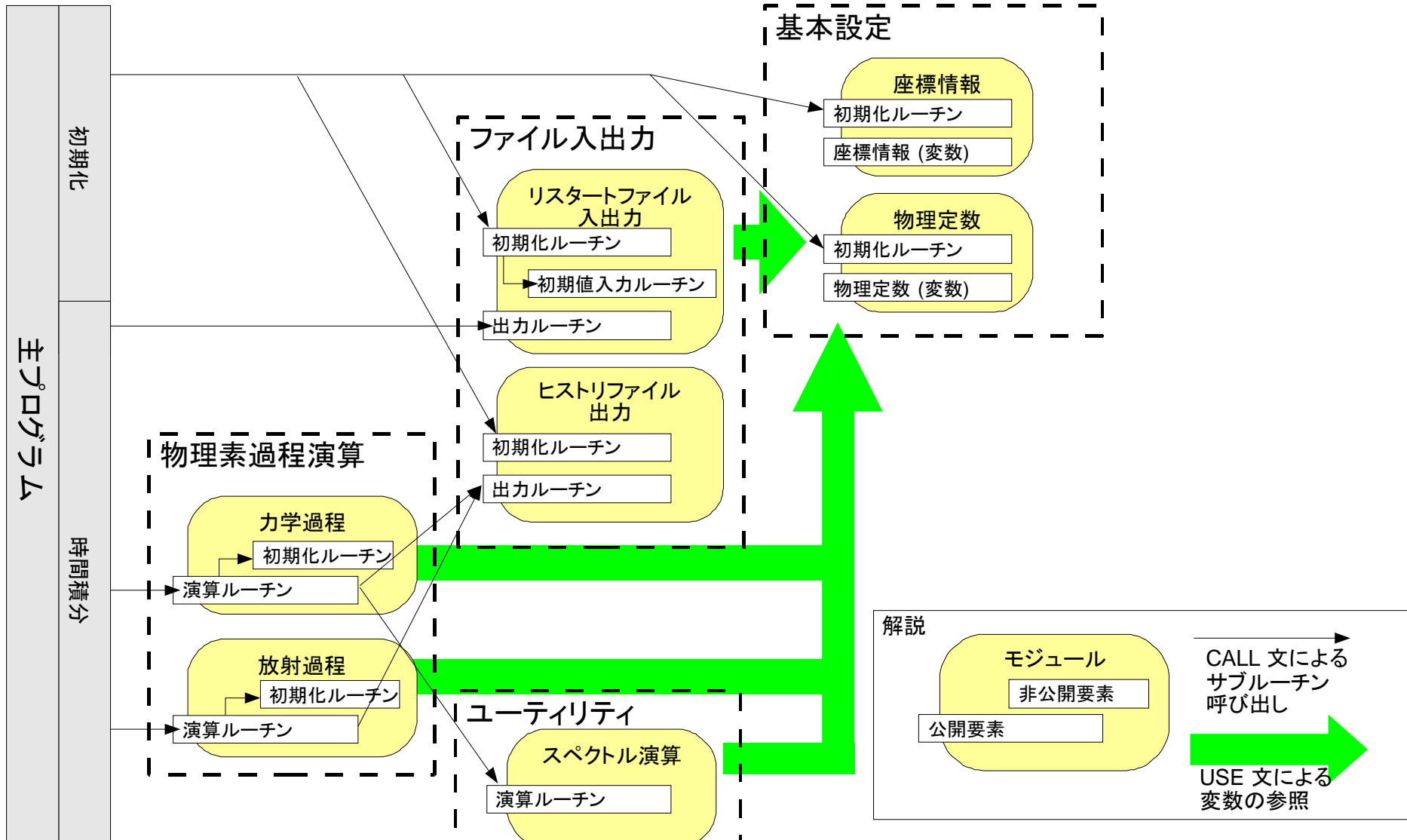
さらに進んだソフトウェア構造を 目指して？ (2/2)

- DCPAM (ver 4) において、モデルを構成する各プログラムの全てを OOP で設計・実装
 - 各プログラムでは、それぞれ初期化ルーチンと演算ルーチンを用意
 - 初期化: 計算に必要な情報 (座標や物理定数等) を受け取って保管
 - 演算: 予報変数を受け取り、演算した後に返す
- 期待していたこと
 - 手続き (サブルーチン、関数) のみで他のプログラムとやり取りする (USE 文すら使用しない) ことにより、プログラムの独立性を高め、着脱が容易に
- 結果
 - ソースコードが不必要に冗長に (結局読みづらい)
 - 初期化ルーチンに対しての引数を明示的な受渡し
 - オブジェクト指向的な設計のための前処理, 後処理
 - 構造体を用いたプログラミング自体がお手軽ではない

ソフトウェア構造に関するまとめ

- 従来の設計と Fortran 90/95 機能の適度な融合
 - モジュールと USE 文の活用による名称衝突の回避
- 「力学」や「物理」の枠をはずす
 - 力学もプロセスの一つと位置づけ
 - 他のモデルでも適用可能に

DCPAM (ver5) のソフトウェア構造



AGCM5 の変数命名法と サブルーチンの書法

- 変数命名法 (Fortran77 規格の 6 文字制限に準拠)
 - 変数名規則 : GAVOR
 - G : 格子点データ (G) か、スペクトルデータ (W) か
 - A : 時刻情報 (例 : A: $t+\Delta t$, B: $t-\Delta t$, T: $d/dt(t)$)
 - VOR : 物理的意味
 - 具体例
 - GAU : 格子点上の東西風速 (時刻 $t+\Delta t$)
 - WTVOR : 渦度の水平スペクトルの時間変化
- サブルーチンの書法
 - 引数の入出力に関してコメント部分に明記
 - I: 入力, O: 出力, M: 入出力, C: 定数

```
CALL DYNMCS          !" 力学項と時間積分
M      ( GAU      , GAV      , GAT      , GAPS      , GAQ      ,
M      GBU      , GBV      , GBT      , GBPS      , GBQ      ,
M      GAVOR     , GADIV    , GBVOR   , GBDIV    ,
M      GTUA     , GTVA     , GTH     , GTR     ,
I      ITA     , DELT     , OADVNC ,
C      ALON    , DLON    , ALAT    , DLAT    ,
C      SIG     , SIGM    , DSIG    , DSIGM    )
```

Fortran 90/95 を活用した発展・拡張

- Fortran 90/95 では
 - 変数名は 31 文字まで
 - サブルーチン・関数の引数には入出力の属性 (INTENT 属性: IN, OUT, INOUT) が指定可能
 - 不正はコンパイラがチェックしてくれる
- 階層モデル群全体で共用できる変数命名法、サブルーチンや関数の利用法は？

Fortran 90/95 を活用した SPMODEL (竹広 他, 2006) の試み (1/3)

- Fortran 90/95 では関数で多次元配列を扱うことが可能

```
! FORTRAN 77 style
  REAL*8    A(10, 10), B(10, 10)
  INTEGER   I, J
  ...
  DO 1000 J=1, 10
    DO 1000 I=1, 10
      B(I, J) = EXP( A(I, J) )
  1000 CONTINUE
```



```
! Fortran 90/95 style
real(8) :: a(10, 10), b(10, 10)
...
b = exp(a)
```

Fortran 90/95 を活用した SPMODEL (竹広 他, 2006) の試み (2/3)

- 変数・関数命名規則

- 変数 : x_VarA

- $x_$: 空間 (座標軸) 情報
 - Var : 物理的意味
 - A : 時刻情報 (例 : $A: t+\Delta t$, $N: t$, $B: t-\Delta t$)

- 具体例

- xyz_TempB : 温度の格子点データ (時刻 $t-\Delta t$) の3次元配列
 - wz_VorA : 渦度を水平スペクトル変換したスペクトルデータ (時刻 $t+\Delta t$) の2次元配列

- 関数 : $xy_VarA_w(_w..)$

- $xy_$: 返り値の空間 (座標軸) 情報
 - $_w$: 入力される値の空間 (座標軸) 情報

- 具体例

- $w_Div_xy_xy$: 2 つの 2次元格子点データの発散を
スペクトルデータとして返す

- AGCM5 の変数命名規則を包括し、様々な多次元データを表記

Fortran 90/95 を活用した SPMODEL (竹広 他, 2006) の試み (3/3)

- サブルーチン使用時に比べ、数式表現に近い形でソースコードを記述

$$\frac{\partial \zeta(t)}{\partial t} = \frac{1}{a \cos \varphi} \left(\frac{\partial VA(t)}{\partial \lambda} - \frac{\partial (v \cos \varphi) UA(t)}{\partial \varphi} \right)$$
$$\zeta(t + \Delta t) = \zeta(t - \Delta t) + 2\Delta t \times \frac{\partial \zeta(t)}{\partial t}$$

```
wz_DVorDtN = &
& wa_Div_xya_xya( xyz_VaN , - xyz_UaN ) &
& / Rplanet
xyz_VorA = &
& xya_wa( wa_xya( xyz_VorB ) &
& + 2. * DeITime * wz_DVorDtN )
```

階層モデル群における SPMODEL方式の検証

- SPMODEL での変数命名法を積極的に利用
- 「物理量」に関しては、可能な範囲で記述的に表記
- 数式表現に合わせることを念頭に、関数を用意
- DCPAM での検証
 - 力学 (スペクトル法を使用) に関しては竹広 (2006) と同様に数式に近い形での記述が可能
 - その他の部分では、AGCM5 に比べて数式表現に近づくわけではないが、個々の変数が何次元のデータか、一目で分かる
 - 変数名が 20 文字を超えると、逆に数式らしさが失われる

階層モデル群における 変数命名法、関数の利用法

- SPMODEL での変数命名法により、数式に近い形での記述が可能 or 個々の変数が何次元のデータか一目で分かる
- 「物理量」に関しては、20 字弱になるよう省略することも必要
- 個々の物理プロセスと主プログラムとのやり取りはサブルーチンで行う
- 階層モデル群で共通利用できる書式である

データ構造に関する問題点

- モデル毎に、出力されるデータの種類 (座標、欠損値、オフセット等) は異なる
- 異なるプラットフォーム間でのデータのやり取り
 - プラットフォーム依存のデータは、移動の際にデータ形式の変換が必要

データ構造の選定

- 階層モデル群で共通で入出力するデータ構造の要件
 - データの次元数に制限が無い
 - 自己記述的 (メタデータ、座標軸データ)
 - 可搬性を有する (プラットフォーム非依存)
- NetCDF (UCAR/Unidata)
 - gtool4 netCDF 規約 (豊田 他, 2000) を採用
 - 地球惑星流体データのためのメタデータを定めたもの
 - 現在デファクトスタンダードとなりつつある CF (climate and forecast) Conventions とほぼ互換

データ入出力の書法に関する要請

- 簡単なモデルにとっては
 - なるべく簡単に書きたい (本来は WRITE 文、READ文などで入出力したいところ)
- 複雑なモデルにとっては
 - 出力するファイルを手軽に追加/削除したい
 - NAMELIST などでの出力間隔を変更したい
 - MPI 使用時に、出力ファイルの分割や統合を手軽に行いたい
- 階層モデル群としては
 - 入出力に関連するソースコードの書式が揃えたい

データ入出力ライブラリ

- gtool5

- gtool4 netCDF データの入出力 + 階層的モデル群のための汎用 Fortran 90/95 ライブラリ
 - gtool4 tools/library (豊田, 2000) および gt4f90io (森川, 2008) の後継ライブラリ
- 階層モデル群で共通利用するサブルーチン・関数を提供
- モデルの出力データの数量に応じて 2 種類のモジュール
 - gtool_history モジュール (入出力)
 - 出力ファイルが数個の小規模モデル用
 - gtool_historyauto モジュール (出力)
 - 出力ファイルがおおよそ 10 を超えるような大規模モデル用
 - NAMELIST による出力間隔等の指定が可能
 - 変数追加を簡単に行える

データ入出力モジュール

- gtool_history
 - 出力
 - 初期設定: HistoryCreate (ファイル1つごと)
 - 変数設定: HistoryAddVariable (変数1つごと)
 - 出力: HistoryPut (座標データの出力含む)
 - 終了: HistoryClose
 - 入力:
 - HistoryGet (変数1つごと)
- gtool_historyauto
 - 出力
 - 初期設定: HistoryAutoCreate (モデル内で1度だけ)
 - 座標設定: HistoryAutoPutAxis (座標1つごと)
 - 変数設定: HistoryAutoAddVariable (変数1つごと)
 - 出力: HistoryAutoPut
 - 終了: HistoryAutoClose

データ構造・データ入出力まとめ

- gtool4 netCDF 規約
 - データを解析・可視化する際に必要な情報がそれぞれのデータファイルに格納
- gtool5 ライブラリ
 - 階層モデル群のデータ入出力部分の書式を揃える
- 解析・可視化について
 - Dennou Ruby Project の解析・可視化ツールを利用
 - GPhys: ライブラリ & コマンドラインインターフェース
 - Gfdnavi: 地球流体データのデータベース、解析、可視化のためのデスクトップツール兼サーバー

解説文書の整備に関する問題点

- ここでの解説文書とは
 - モデルを構成するサブルーチンや関数など、個々のプログラム単位の機能や使用法の詳細を記述した文書 (=リファレンスマニュアル)
- プログラムは頻繁に更新されることを想定するため、それに追隨して別途、解説文書を作成するのは手間がかかる
- プログラムとして一度記述したものを再度記述するのが面倒

近年の解説文書の整備手法

- ドキュメント自動生成システム
 - ソースコードのコメント行に文書を埋め込み、プログラムとドキュメントを一元管理
 - ドキュメント生成をソースコード解析ソフトウェアによって自動化
- 既存のソフトウェア
 - RDoc (Ruby), JavaDoc (Java), Doxygen (C, C++, Python, IDL)
 - 自動生成される HTML or XML ドキュメントにハイパーリンクや見出し、箇条書きといった構造を付与

RDoc (Ruby Document System)

- Ruby で書かれたソースコードからドキュメントを自動生成する Ruby の標準ライブラリ
- Fortran 90/95 の解析も可能
 - ソースコード解析機構とマニュアル生成機構が分離しているため、他の言語で書かれたソースコードも解析可能
 - 標準で C および Fortran95 用の解析機構が付属
 - ただし、Fortran95 で解析可能な言語要素は限られていた

Fortran 90/95 解析機能の強化

- 一通りの言語要素の解析に対応
 - サブルーチンと関数のコメント
 - サブルーチンや関数の引数とそのコメント
 - PRIVATE, PUBLIC 属性
 - 構造データ型
 - 定数、変数
 - 利用者定義演算 (OPERATOR)、利用者定義代入 (ASSIGNMENT)、総称手続き (INTERFACE)
 - NAMELIST 変数群名
- Ruby 用 MathML ライブラリとの併用により、TeX 書式で記述されたコメント文を MathML へと変換
 - MathML: ブラウザ上で数式表現可能な言語
- 強化版解析スクリプトは Ruby ライブラリ本家にフィードバック済み (最新版は取り込まれておらず)

解説文書の書法

- 公開言語要素にはコメントとして概説を記載
 - ファイル、モジュール、サブルーチン、関数、定数、変数等
 - サブルーチンと関数の引数
- 解説文書とソースコードを一元管理
- ソースコードから物理モデルを把握する一助にも

```
! = 積雲パラメタリゼーション: 対流調節スキーム
! Authors: : Yasuhiro MORIKAWA,
! License: : See COPYRIGHT[Link: .. / .. / .. / COPYRIGHT]

module cumulus_adjust
! = 積雲パラメタリゼーション: 対流調節スキーム
! 対流調節スキームにより, 温度と比湿を調節します.

subroutine Cumulus( &
& xyz_Temp, xyz_QVap, xyz_Press, xyr_Press )
! 対流調節スキームにより, 温度と比湿を調節します.
!
real (DP), intent(inout):: xyz_Temp (0:i max-1, 1:j max, 1:k max)
! $ T $ . 温度.
real (DP), intent(inout):: xyz_QVap (0:i max-1, 1:j max, 1:k max)
! $ q $ . 比湿.
real (DP), intent(in):: xyz_Press (0:i max-1, 1:j max, 1:k max)
! $ p $ . 気圧 (整数レベル).
real (DP), intent(in):: xyr_Press (0:i max-1, 1:j max, 0:k max)
! $ \hat{p} $ . 気圧 (半整数レベル).
:
```

RDoc によるドキュメント自動生成

ファイル
モジュール
サブルーチン・関数 } 一覧
(ハイパー
リンク)

module cumulus_adjust
!
! 積雲パラメタリゼーション
!

module dynamics_hspl_vs
!
! 力学コア (スペクトル法)
!

module radiation_band
!
! 放射 (バンドモデル)
!

program dcpam_ape
!
! APE 主プログラム
!

Fortran 90/95
ファイル群

サブルーチンの
概説・引数に関する情報



Files	Classes	Methods
cumulus/cumulus_adjust.f90	NAMELIST	AtmosAlbedo (radiation_band)
dcpam5_modules.rdoc	axeset	AvrLat_y (intavr_operate)
dryconv_adjust/dryconv_adjust.f90	constants	AvrLonLat_xy (intavr_operate)
dynamics/dynamics_hspl_vas83.f90	cumulus_adjust	AvrLon_x (intavr_operate)
held_suarez_1994/held_suarez_1994.f90	dryconv_adjust	AxesetInit (axeset)
	dynamics_hspl_vas83	BasePotTemp (vdiffusion_my1974)

Class **cumulus_adjust**
In: cumulus/cumulus_adjust.f90

積雲パラメタリゼーション: 対流調節スキーム Cumulus parameterization: Convective adjustment scheme

Note that Japanese and English are described in parallel.
対流調節スキームにより、温度と比湿を調節します。飽和比湿の計算には Nakajima et al. (1992) を用いています。詳しくは saturate_nha1992 を参照してください。

Adjust temperature and specific humidity by convective adjustment scheme. Nakajima et al. (1992) is used for calculation of saturation specific humidity. For details, see "saturate_nha1992".

モジュールの概説

Procedures List
Cumulus : 温度と比湿の調節

Cumulus : Adjust temperature and specific humidity

Public Instance methods

```
Cumulus( xyz_Temp, xyz_QVap, xyz_Press, xyr_Press )  
Subroutine :  
xyz_Temp(0:imax-1, 1:jmax, 1:kmax) : real (DP), intent (inout)  
: T . 温度. Temperature  
xyz_QVap(0:imax-1, 1:jmax, 1:kmax) : real (DP), intent (inout)  
: q . 比湿. Specific humidity  
xyz_Press(0:imax-1, 1:jmax, 1:kmax) : real (DP), intent (in)  
: p . 気圧 (整数レベル). Air pressure (full level)  
xyr_Press(0:imax-1, 1:jmax, 0:kmax) : real (DP), intent (in)  
: p-hat . 気圧 (半整数レベル). Air pressure (half level)
```

対流調節スキームにより、温度と比湿を調節します。
Adjust temperature and specific humidity by convective adjustment scheme.
[Source]

まとめ

- DCPAM (ver 5) の開発とソフトウェア基盤整備
 - 階層モデル群で共通化できるソフトウェアの基本設計の提案
 - 従来的設計に Fortran 90/95 機能 (モジュール、USE 文) 導入
 - 階層モデル群で共通化できる書法 (変数などの命名法など) の提案
 - 数式表現される物理モデルと対応付けが良いように記述
 - gtool5 ライブラリとデータ入出力に関する書法の策定
 - 階層モデル群のデータ入出力部分を統一的な書式へ
 - RDoc Fortran 90/95 解析機能強化と解説文書に関する書法の策定
 - ソースコードと解説文書とを一元管理
- 階層モデル群開発に向けての課題
 - 並列化に関する書法の策定
 - 適度に隠蔽できるか検証と実装
 - より現実的な地球大気や火星や金星を計算可能なGCMへ
 - 放射対流モデル、エネルギーバランスモデルの整備

参考資料

- DCPAM (Dennou-Club Planetary Atmospheric Model)
 - <http://www.gfd-dennou.org/library/dcpam/>
- gtool5: 階層的数値モデル群のための Fortran 90/95 ライブラリ
 - <http://www.gfd-dennou.org/library/gtool/>
- gtool4 netCDF 規約
 - <http://www.gfd-dennou.org/library/gtool/>
- 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.
- RDoc Fortran 90/95 ソースコード解析機能強化版
 - <http://www.gfd-dennou.org/library/dcmmodel/rdoc-f95/>
- 地球流体電脳倶楽部 dcmmodel プロジェクト: 地球惑星流体科学のための階層モデル群開発プロジェクト
 - <http://www.gfd-dennou.org/library/dcmmodel/>
- 竹広真一, 小高正嗣, 石岡圭一, 石渡正樹, 林祥介, SPMODEL 開発グループ, 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, <http://www.nagare.or.jp/mm/2006/spmodel/>
- 階層的地球流体スペクトルモデル集 SPMODEL
 - <http://www.gfd-dennou.org/library/spmodel/>
- 電脳Rubyプロジェクト
 - <http://ruby.gfd-dennou.org/>
- 堀之内武, 西澤誠也, 渡辺知恵美, 森川靖大, 神代剛, 石渡正樹, 林祥介, 塩谷雅人, 2007: 地球流体データベース・解析・可視化のための新しいサーバ兼デスクトップツールGfdnavi の開発. Proceedings of Data Engineering Workshop (DEWS) 2007, D2-8.
- Gfdnavi
 - <http://www.gfd-dennou.org/arch/davis/gfdnavi/>
- 地球流体電脳倶楽部版 AGCM5
 - <http://www.gfd-dennou.org/library/agcm5/>
- 気象庁 Fortran 標準コーディングルール
 - <http://www.mri-jma.go.jp/Project/mrinpd/coderule.html>
- NetCDF: network Common Data Form (UCAR/Unidata)
 - <http://www.unidata.ucar.edu/software/netcdf/>