

# 非静力対流モデル deepconv

杉山耕一郎<sup>1</sup>, 山下達也<sup>2</sup>, 小高正嗣<sup>2</sup>, 中島健介<sup>3</sup>,  
deepconv 開発グループ

1: 北大低温研, 2: 北大理・宇宙理学,  
3: 九大理・地球惑星科学

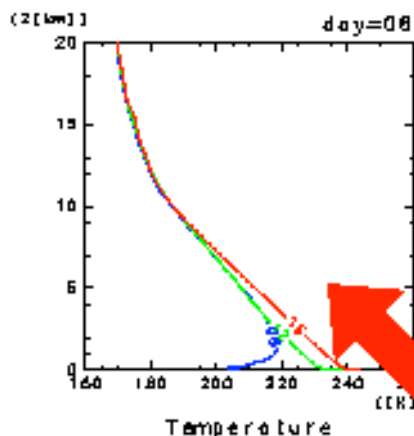


2010年3月8日  
davis@神戸大

- ターゲット
  - 大気構造: ex. 放射対流平衡
  - 個々の現象のシミュレーション
    - ex. 火星のドライアイス雲やダストデビル, 木星や土星の積雲
  - 雲対流構造の多様性の探索
    - 仮想的パラメタ空間への積極的拡張. 系外惑星
- 階層モデル群の1つ: 数値モデルの世界での相互検証を可能に
  - さまざまなレイヤー(物理的階層)に特化した階層的に存在する数値計算モデル群による相互検証
  - 地球の気象学・気候学で培われてきた(あるいはさらに新たな)数値モデル群のラインアップが必要



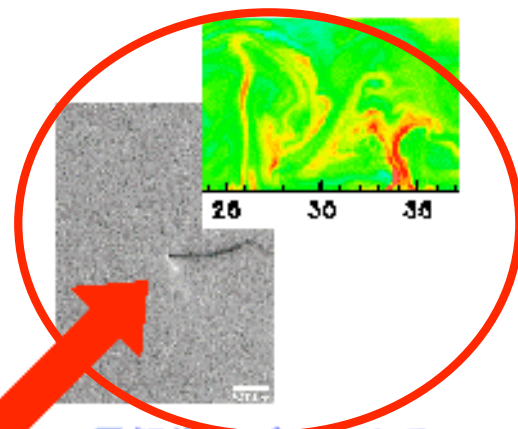
# 階層モデル群:仮想気候サーベイヤー (Virtual Climate Surveyer, VCS)



放射対流モデルによる鉛直大気構造の理解

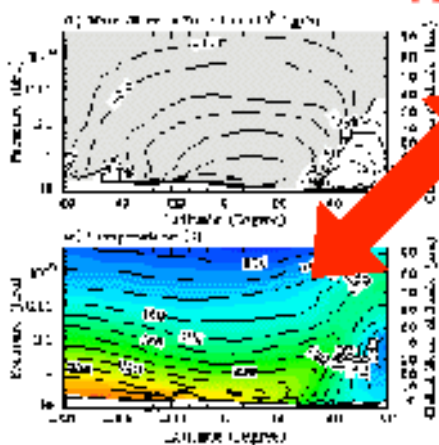


太陽系惑星との比較



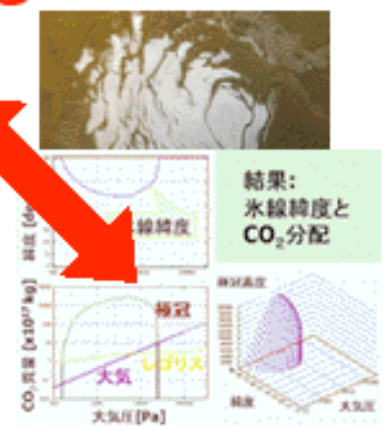
雲解像モデルによるダストデビルの再現

階層的モデル群によるアプローチ



GCM による大気大循環の研究

仮想的惑星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

更なるサブシステム(海洋・氷・陸面・固体地球)との階層的結合



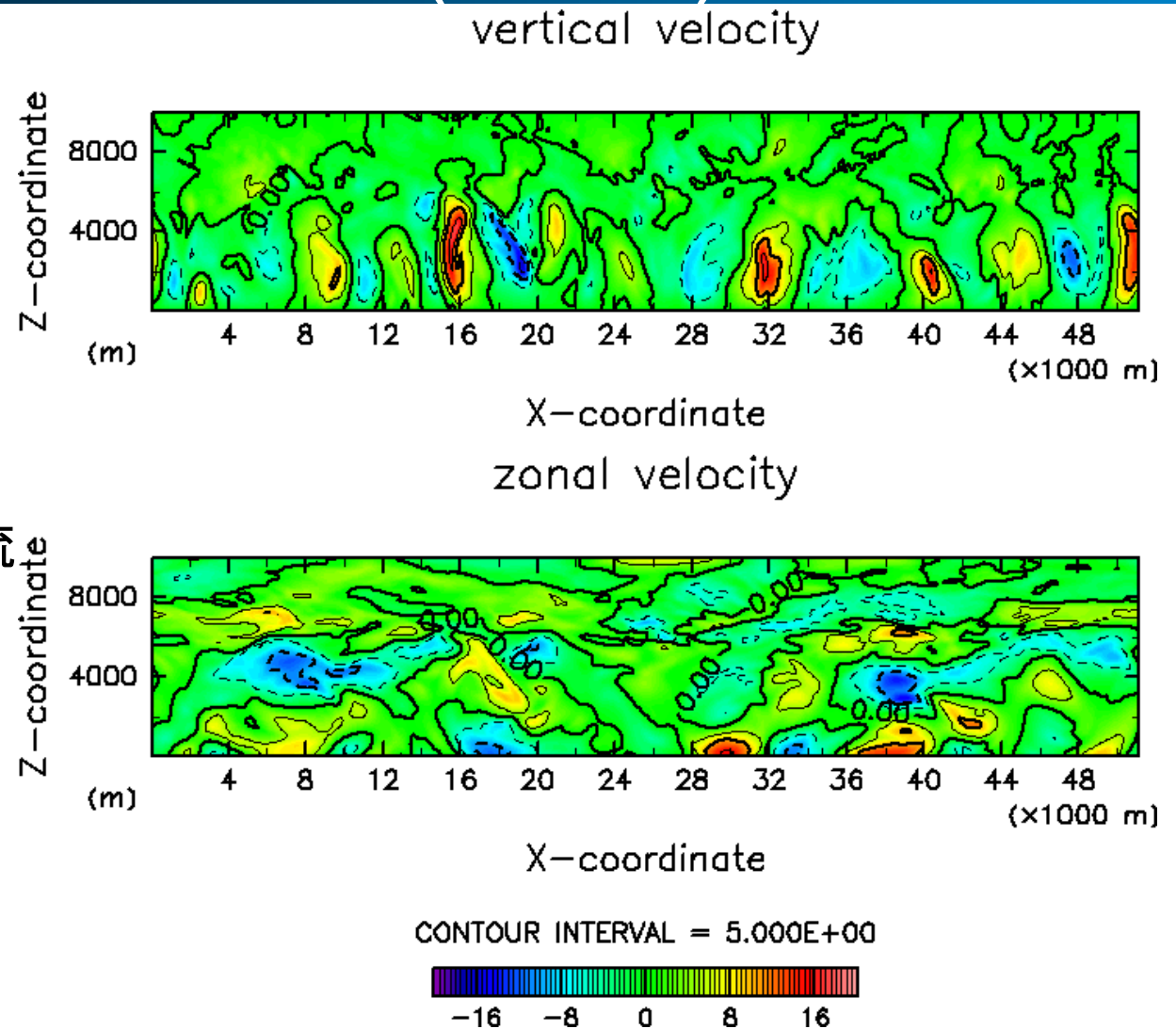
- F77 版 deepconv
  - 2次元非弾性系モデル(中島, 1994)
  - 放射, 地表面過程を導入し火星へ応用(Odaka, 2001)
- F90 版 deepconv
  - F77 版の後継モデルとして 2002 年に開発開始
  - 2003 年: コード名 kaminari 開発開始
    - 乾燥大気用モデルの試作
  - 2004 年: コード名 arare 開発開始
    - Ver.2: 乾燥大気
    - Ver.3: 湿潤過程と主成分凝結過程の試験的導入
    - Ver.4: 複数の凝結成分, 3次元化, MPI 対応, 主成分大気凝結
    - (Ver.5: 方程式系の統合, コードの統合)

- 木星の雲対流の2次元計算
  - 3つの凝結気体
  - 3つの雲成分
- 現在の火星の乾燥対流の3次元計算
  - 地表面から高さ数キロまでの乾燥対流
  - 渦の発生
- 火星大気の主成分凝結を伴う対流(2次元)
  - 大気量は現在の火星
  - 温度は火星の極域程度
- タイタンの雲対流(2次元)
- 地球の雲対流(2次元)[進行中]



# 計算例: 火星の乾燥対流(2次元)

- Odaka et al. (1998);  
Odaka et al. (2001)
  - 格子間隔 100 m
  - 熱強制固定/  
日変化する放射
  - 結果: km サイズの対流
    - $W \sim 20$  m/sec
    - $U \sim 10$  m/sec



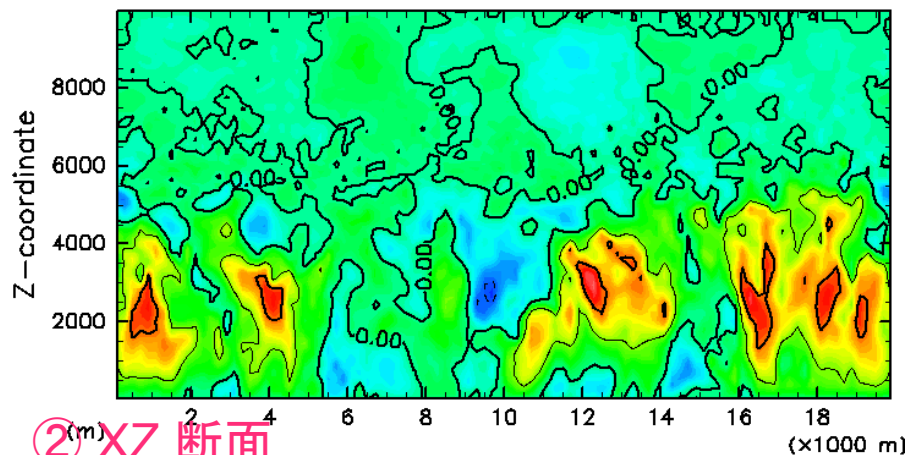


# 計算例: 火星の乾燥対流(3次元)

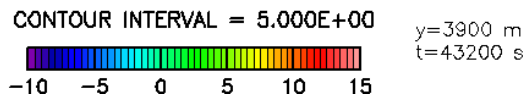
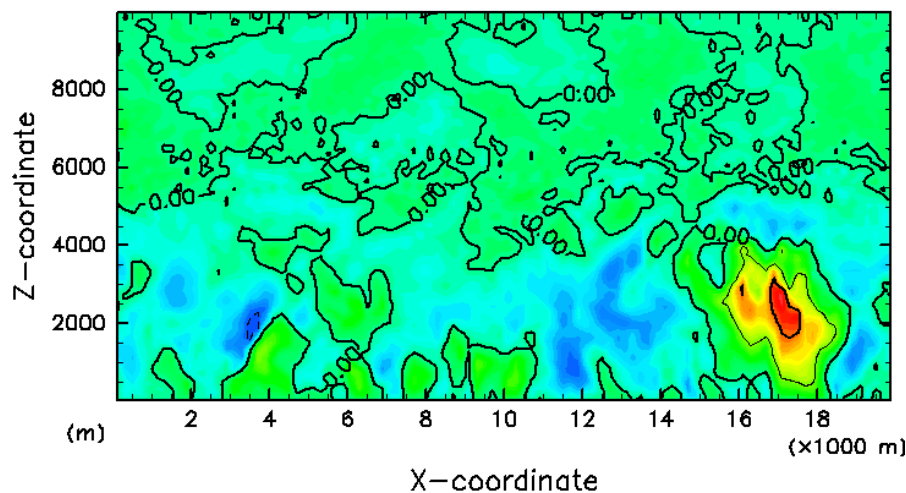
[Odaka et al. \(2008\)](#)

① YZ 断面

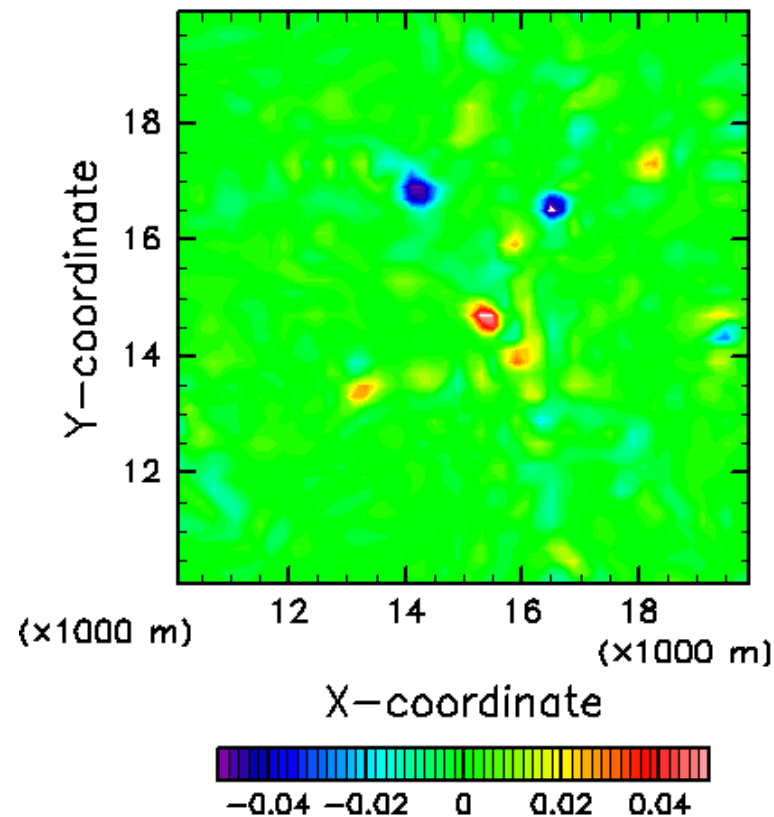
vertical velocity



② XZ 断面



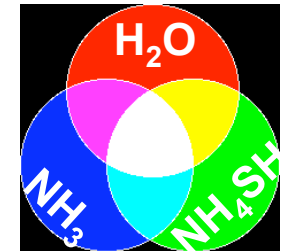
Vorticity



z=50 m  
t=43200 s

# 木星：雲対流の計算例

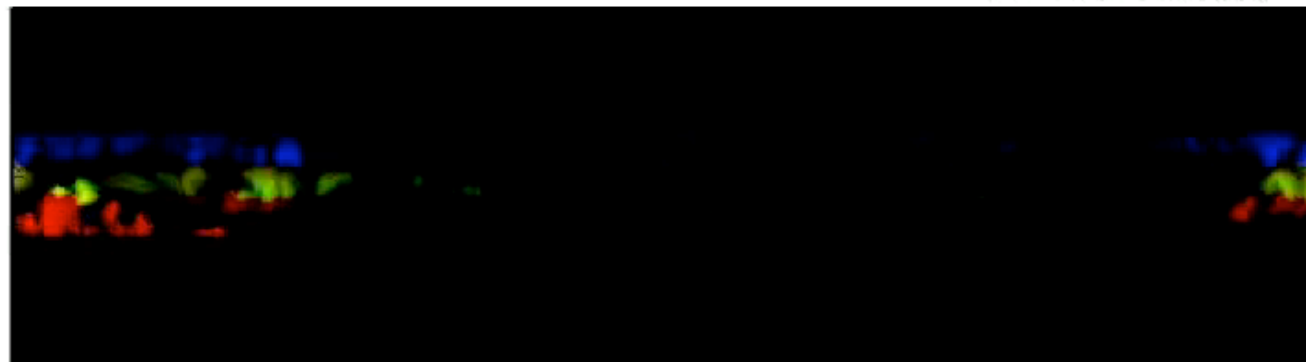
間欠的に活発な対流活動が生じる



雲・雨の  
混合比

(x1E5)

Height  
2  
1



(x1E5)

鉛直  
速度

Height  
2  
1



1 2 3 4 5 6 7 8 9 10  
(x1E5)

Length

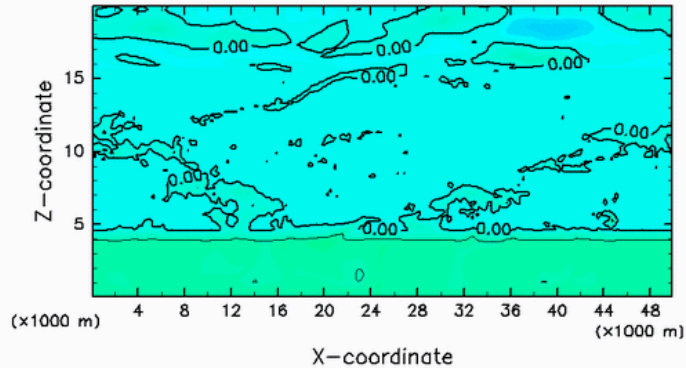




# 計算例: 火星の主成分凝結対流 (2次元)

## 温位

disturbance of potential temperature

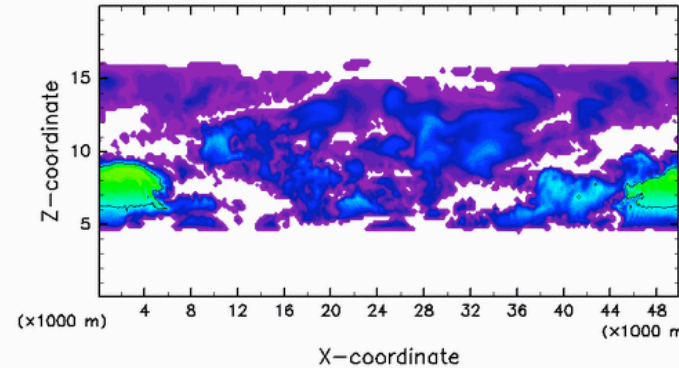


/usr/bin/gview 2008-12-20

MarsCond\_PotTemp.nc@PotTempDist,x=0:50000,z=0:20000,t=0:27000

## 雲密度

density of cloud



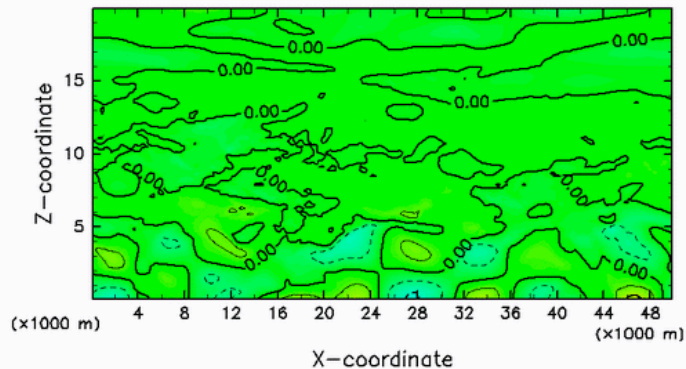
/usr/bin/gview 2008-12-20

MarsCond\_DensCloud.nc@DensCloud,x=0:50000,z=0:20000,t=0:27000

- 過飽和の有無による流れ場の様相を調べた。

- 過飽和なし: 雲が全て蒸発
- 過飽和あり: 系全体が雲で覆われる

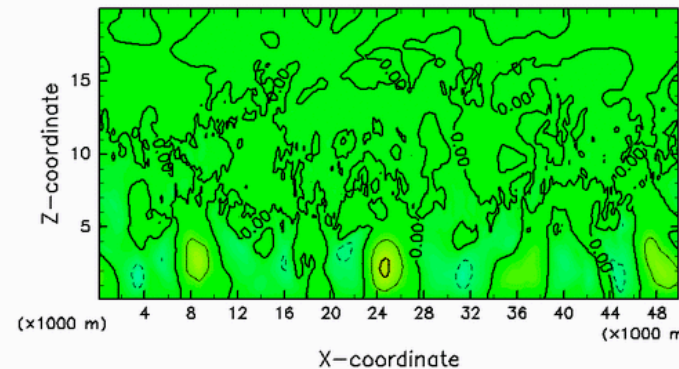
## zonal velocity



/usr/bin/gview 2008-12-20

MarsCond\_VelX.nc@VelX,x=0:50000,z=0:20000,t=0:27000

## vertical velocity



/usr/bin/gview 2008-12-20

MarsCond\_VelZ.nc@VelZ,x=0:50000,z=0:20000,t=0:27000

## 水平流速

## 鉛直流速

- 現段階では方程式系は、普通の“地球用”を踏襲
  - 地球大気での経験を生かす
  - 地球で得られてきた常識の再検討
    - 何をどう犠牲にして表現してきたかを思い出しながら



- 基礎方程式
  - 力学
    - 準圧縮系 (Klemp and Wilhelmson, 1978)
  - 乱流
    - 1.5 次のクロージャ (Klemp and Wilhelmson, 1978)
  - 湿潤過程 (2次元版のみ)
    - Kessler (1969) のパラメタリゼーション
    - 雲粒の拡散成長
  - 放射
    - 一様冷却/加熱
  - 地表面フラックス
    - バルク法
    - 拡散

# 定式化

- 状態方程式

$$\rho = \frac{p}{R_d T} \left( \frac{1/M_d}{1/M_d + \sum q_v/M_v} \right) (1 + \sum q_v + \sum q_c + \sum q_r) = \frac{p_0 \pi^{c_{vd}/R_d}}{R_d \theta_v} \left( \theta_v = \frac{\theta}{\left( \frac{1/M_d}{1/M_d + \sum q_v/M_v} \right) (1 + \sum q_v + \sum q_c + \sum q_r)} \right)$$

- 速度の式

$$\begin{aligned} \frac{\partial u}{\partial t} &= - \left( u \frac{\partial u}{\partial x} + w \frac{\partial u}{\partial z} \right) - c_{pd} \bar{\theta}_v \frac{\partial \pi}{\partial x} + Turb.u \\ \frac{\partial w}{\partial t} &= - \left( u \frac{\partial w}{\partial x} + w \frac{\partial w}{\partial z} \right) - c_{pd} \bar{\theta}_v \frac{\partial \pi}{\partial z} + Turb.w + g \left( \frac{\theta}{\bar{\theta}} + \frac{\sum q_v/M_v}{1/M_d + \sum \bar{q}_v/M_v} - \frac{\sum q_v + \sum q_c + \sum q_r}{1 + \sum \bar{q}_v} \right) \end{aligned}$$

- 圧力の式

$$\frac{\partial \pi}{\partial t} = - \frac{\overline{C_s^2}}{c_{pd} \bar{\rho} \bar{\theta}_v^2} \frac{\partial}{\partial x_j} (\bar{\rho} \bar{\theta}_v u_j)$$

- 熱の式

$$\begin{aligned} \frac{\partial \theta}{\partial t} &= - \left( u \frac{\partial \theta}{\partial x} + w \frac{\partial \theta}{\partial z} \right) - w \frac{\partial \bar{\theta}}{\partial x} + \frac{L}{c_{pd} \bar{\pi}} (CN_{vc} - EV_{cv} - EV_{rv}) \\ &\quad + \frac{1}{\bar{\pi}} (Q_{rad} + Q_{dis}) + Turb.\bar{\theta} + Turb.\theta \end{aligned}$$

変数		上付き添え字
$p$ : 圧力	$R$ : 気体定数	$\bar{\cdot}$ : 平均成分
$\pi$ : 無次元圧力	$M$ : 分子量	
$T$ : 温度	$c_p$ : 比熱	
$\theta$ : 温位	$C_s$ : 音速	下付き添え字
$\theta_v$ : 仮温位	$V_t$ : 雨の落下速度	$d$ : 乾燥成分
$u$ : 水平風速	$L$ : 潜熱	$v$ : 凝縮成分気体
$w$ : 鉛直風速	$Turb$ : 乱流拡散項	$c$ : 雲
$q$ : 混合比	$Q_{rad}$ : 放射加熱項	$r$ : 雨
$\rho$ : 密度	$Q_{dis}$ : 散逸加熱項	

- 凝結成分の保存式 (蒸気, 雲, 雨 × 成分)

$$\begin{aligned} \frac{\partial q_v}{\partial t} &= - \left( u \frac{\partial q_v}{\partial x} + w \frac{\partial q_v}{\partial z} \right) - w \frac{\partial \bar{q}_v}{\partial x} - (CN_{vc} - EV_{cv} - EV_{rv}) + Turb.q_v + Turb.\bar{q}_v, \\ \frac{\partial q_c}{\partial t} &= - \left( u \frac{\partial q_c}{\partial x} + w \frac{\partial q_c}{\partial z} \right) + (CN_{vc} - EV_{cv} - CN_{cr} - CL_{cr}) + Turb.q_c, \\ \frac{\partial q_r}{\partial t} &= - \left( u \frac{\partial q_c}{\partial x} + w \frac{\partial q_c}{\partial z} \right) + (CN_{cr} + CL_{cr} - EV_{rv}) + \frac{\partial}{\partial z} (\bar{\rho} V_t q_r) + Turb.q_r \end{aligned}$$

- 凝結物質が色々
  - 多成分・化学反応：木星・土星 etc.
  - 主成分の凝結：火星（現在、過去）、地球（過去）
  - 混合物の凝結：タイタンの雲はメタンと窒素の混合物
- 凝結条件
  - 有意の大きさの過飽和度が必要？（火星、タイタン）
- 境界条件
  - 海の有無、成分：タイタンの海はメタン・エタン混合物？
  - 地面の有無：木星・土星は“底なし”に近い
- 観測が不足（or皆無）
  - 遠い過去の惑星大気、系外惑星
  - 計算条件が制約しにくい。



- 現段階では方程式系は、普通の“地球用”を踏襲
  - 地球大気での経験を生かす
  - 地球で得られてきた常識の再検討
    - 何をどう犠牲にして表現してきたかを思い出す
- 階層モデル群全体としてスタイルを統一
  - 自由な実験設計・複数のモデルの同時使用を担保する  
可読性可変性の重視(統合されたソフトウェア書式)
  - 共通したデータ構造、可視化・解析ツール



- dcmode1 プログラミングガイドライン
  - Fortran 90/95 プログラムのソースコード書法, 実行プログラムの動作, データの入出力に関するスタイルを提案
    - 可変性・可読性の確保
    - 階層モデル群全体としてスタイルの統一 → 詳しくは実習編で
- データ入出力ライブラリ gtool5
  - 煩雑な入出力処理を隠蔽
  - 階層モデル群全体の I/O を統一
- RDoc Fortran 90/95 ソースコード解析
  - Fortran 90/95 ソースコード内に書かれたコメントからリファレンスマニュアルを自動生成
    - ソースコードとマニュアルの乖離を防ぐ





- 格子点と変数の格子配置
  - 等間隔格子
  - Arakawa C + Lorenz 型
- 空間方向: 中心差分
  - 移流項は4次精度
  - その他の項は2次精度
- 時間積分: モード別時間分割法
  - 音波: HE-VI 法
    - 水平方向はオイラー法, 鉛直方向はクランク-ニコルソン法
  - その他の成分: リープフロッグ + オイラー法

- 各物質の物性値をプログラム内に保持
  - 物質を ID 管理
  - 分子量, 比熱, 飽和蒸気圧, etc.
  
- 複数の凝結成分・化学反応を取り扱う
  - 使う物質は Namelist で指定
  - 凝結と反応が共存する系も扱えるように
    - $\text{NH}_3(\text{g}) \rightarrow \text{NH}_3(\text{s}), \quad \text{NH}_3(\text{g}) + \text{H}_2\text{S}(\text{g}) \rightarrow \text{NH}_4\text{SH}(\text{s})$

- 水平方向に領域分割を利用
  - メインプログラムに, MPI の初期化と終了処理を追加
  - 境界条件ルーチンで, 水平方向の「糊代」のデータ分を通信
  - gtool5 の HistoryAutoCreate 対応中

並列化前

CPU0

並列化後

CPU0

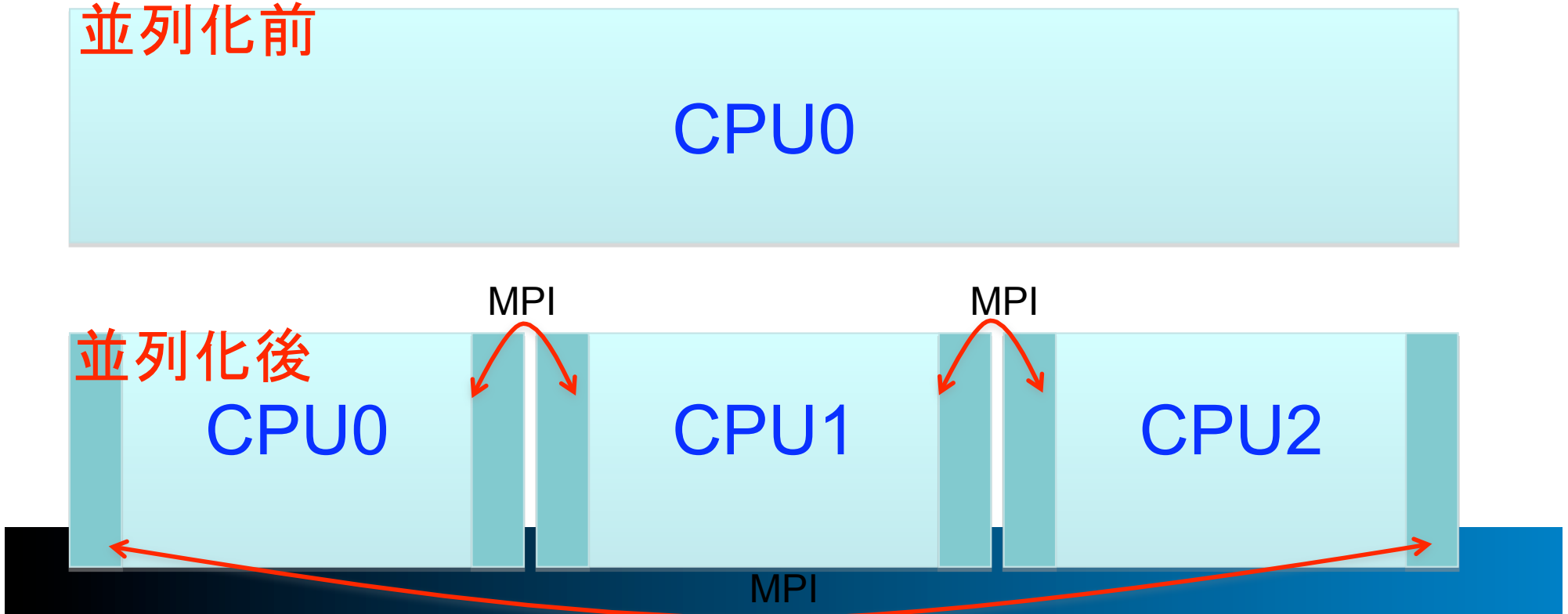
MPI

CPU1

MPI

CPU2

MPI



# 実習編

---



- 動作環境
  - x86Linux + (Fujitsu fortran, g95, Intel fortran)
  - MacOS + g95
  - CRAY XT4 + PGI Fortran
  - HITACHI SR11000 + 最適化 Fortran
  - NEC SX6, SX8 + sxf90
  - Fujitsu FX1 + Fujitsu fortran
- ライブラリ
  - netCDF (I/O)
  - gtool5 (I/O)
  - Lapack, blas (線形計算)

- ソースの場所へ移動

```
$ cd ~/Desktop/Tutorial/dcmode/deepconv/
```

- ファイルの展開, ソースディレクトリへの移動

```
$ tar xzvf arare4_current.tgz
```

```
$ cd arare4-20100306
```

- 環境変数の設定 (コンパイラの指定)

```
$ export FC=gt5frt
```

- Mkinclude の作成

```
$ ./config-20100309.sh
```

– 上記のスク립トでやっているのは  
以下の内容.

```
$ ./configure --with-netcdf=/usr/lib/libnetcdf-g95.a \  
--with-gtool5=/usr/lib/gtool5-g95/lib/libgtool5.a \  
--with-lapack=/usr/lib/liblapack-g95.a ¥  
--with-blas=/usr/lib/libblas-g95.a
```



- コンパイル

\$ make

- 各種ファイルが生成

- ./bin :実行ファイルが格納
  - arare 複数凝結成分対応版
  - arare\_jupiter 木星の化学反応考慮した版
  - randomset 初期値用乱数設定コマンド
- ./lib: ライブラリ libarare.a が格納される
- ./include: 各種 \*.mod ファイルが格納される

- ・ ソースの場所に移動

```
$ cd /home/user/Desktop/Tutorial/dcmmodel/deepconv/
```

- ・ ファイルの展開, ソースディレクトリへの移動

```
$ tar xzvf arare4_current.tgz
```

```
$ cd arare4-20100306
```

- ・ コンパイラの指定

```
$ export FC=gt5frt
```

- ・ Mkinclude の作成

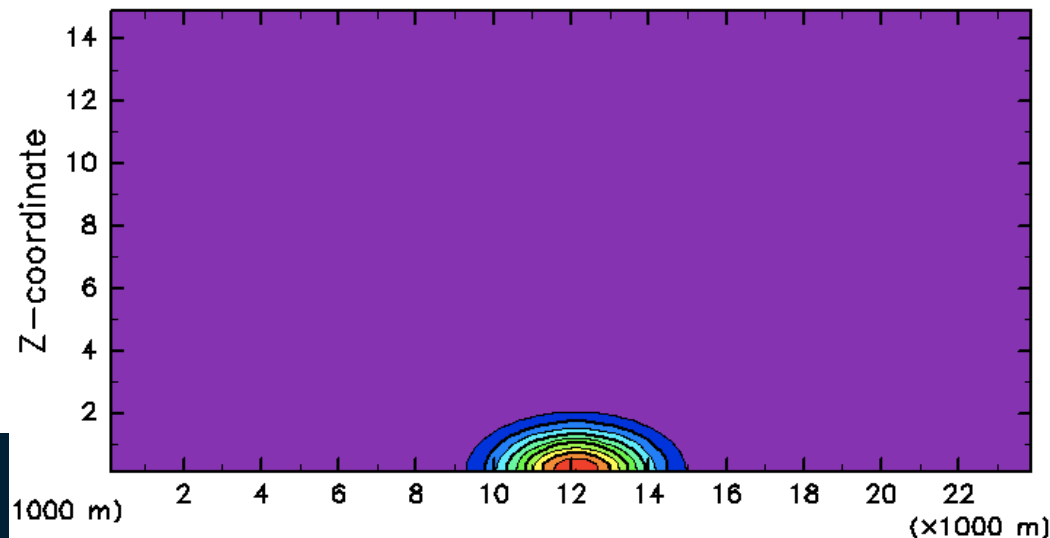
```
$ ./config-20100309.sh
```

- ・ コンパイル

```
$ make
```

# 例題: サーマルの上昇

- 大気下層に温位擾乱を与え, その時間変化を計算する.
  - 地球大気(?)を想定した計算  
`$ ./bin/arare -N=arare-earth.conf`
  - 木星大気を想定した計算  
`$ ./bin/arare_jupiter -N=arare-jupiter.conf`



# 例題: 2次元拡散方程式

- 周期境界条件の下で解く
  - spmodel と同じ例題で

$$\frac{\partial \zeta}{\partial t} = \nu \left( \frac{\partial^2 \zeta}{\partial x^2} + \frac{\partial^2 \zeta}{\partial y^2} \right)$$

- サンプルプログラム:  
`src/main/diffuse_2d.f90`
  - オイラー法を利用

- コンパイル

```
$ gt5frc src/main/diffusion_2d.f90 ¥
```

```
  -I./include -L./lib -larare → a.out ができる
```

- 実行

```
$ ./a.out
```

→ diffuse\_2d.nc ができる

- 結果表示

```
$ gplist advect.nc (変数のリスト)
```

```
$ gpview --anim t advect.nc@zeta (アニメーション)
```

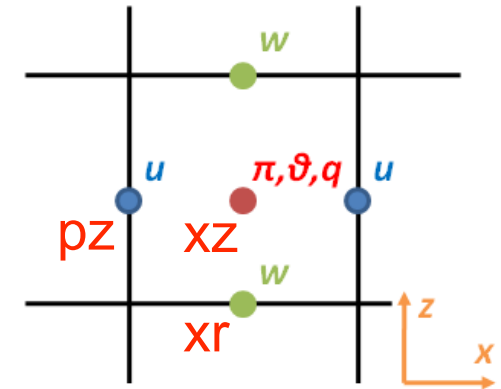
```
$ gpview range 0:1 --anim t advect.nc@zeta
```

- 用いるテクニックとライブラリは SPMODEL を模倣
  - Fortran90 配列計算機能を活用
- 格子点位置を合わせて計算

$$\frac{\partial^2 \zeta}{\partial x^2} \rightarrow \text{xz\_dx\_pz}(\text{pz\_dx\_xz}(\text{xz\_ZetaN}))$$

- 位置を変更するには平均操作する

$$\frac{\partial \zeta}{\partial x} \rightarrow \text{xz\_pz}(\text{pz\_dx\_xz}(\text{xz\_ZetaN}))$$



pz\_VelX (水平速度)  
xr\_VelZ (鉛直速度)  
xz\_PotTemp (温位)

# deepconv/arare4 的書法

```
program diffuse_2d
  use gridset, only :: DimXMin...
  use differentiate_center4, only:xz_dx_pz...
  ...
do it=1,nt
  write(*,*) '*it = ', it
  xz_ZetaA = xz_ZetaN &
    & + dt * nu * ( &
    &   xz_dx_pz(pz_dx_xz(xz_ZetaN)) &
    & + xz_dz_xr(xr_dz_xz(xz_ZetaN)) &
    & )
  call BoundaryXcyc_xz( xz_ZetaA )
  call BoundaryZCyc_xz( xz_ZetaA )
  xz_ZetaN = xz_ZetaA
end do
```

## モジュール引用

- 格子点生成, 微分演算 (4次中央差分) モジュールを引用

## 拡散の計算

- 配列添え字を隠蔽
- 形式的に一行で書ける

- 境界条件適用
- ループ処理



- 詳しくは, dcmode1 プログラミングガイドラインを参照のこと  
<http://www.gfd-dennou.org/library/dcmode1/coding-rules/dcmode1-coding-rules.htm>
  - 気象庁標準コーディングルール を参考にした

# 性能評価

---



- 国立天文台 CfCA XT4, 北大 SR11000
  - 約 3 GFLOPS
- 並列化 (10,000 step)
  - 4並列: 730 sec (1.0)
  - 16並列: 759 sec (1.04)
  - 64並列: 813 sec (1.11)

- 実行性能が上がらない原因
  - 微分平均演算モジュールは使用回数が多いが, FLOPS 値は低い
    - 1プログラムあたりの演算数が少ない
    - 問題となっているプログラムの多くが, 「加算 1 回, 乗算 1 回でおわり」という内容になっている
  - 日立のエンジニア曰く“F77 っぽく書かないと速くならない”

# 今後の展開

---



- 汎惑星大気用の雲対流モデルの開発
- 個々の惑星大気を扱う雲対流モデルを実働させ、経験を積んできた
- 各惑星用に個別に開発してきた方程式系やソースコードを統一する作業を実行中.
  - 方程式系の統合・拡張の目処はついた
  - 個々の惑星用の物理過程モジュールの着脱が、まだすっきりいかない。試行錯誤中
  - “仮想的”大気(任意の分子量、比熱など)の一貫性を持たせた扱いは？ 試行錯誤中



- 過去の開発メンバー
  - 高橋こう子さん, 北守太一さん
- 並列化のプロトタイプ作成
  - 高橋芳幸さん
- 定式化と離散化の方法に関して
  - CReSS (榊原と坪木, 2001)
  - 気象庁コーディングルール
- 開発時に利用したスーパーコンピュータ
  - CGER/NIES (国立環境研究所)
  - 北大情報基盤センター
  - ISAS/JAXA 宇宙科学情報解析センター
  - 国立天文台 天文シミュレーションプロジェクト

# 付録

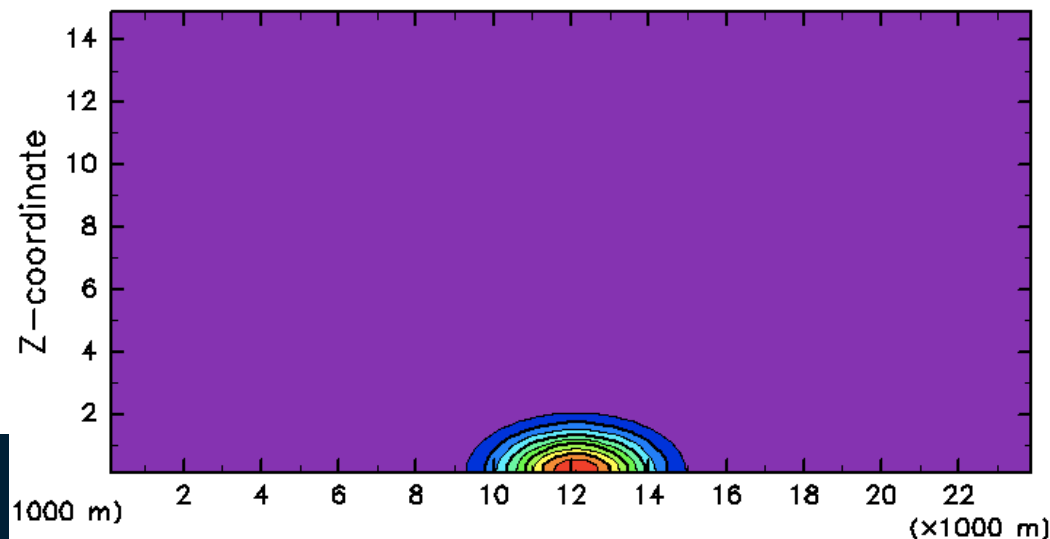
---





# 例題: サーマルの上昇

- 大気下層に温位擾乱を与え, その時間変化を計算する.
  - 地球大気を想定した計算  
\$ ./bin/arare -N=arare-earth.conf
  - 木星大気を想定した計算  
\$ ./bin/arare\_jupiter -N=arare-jupiter.conf

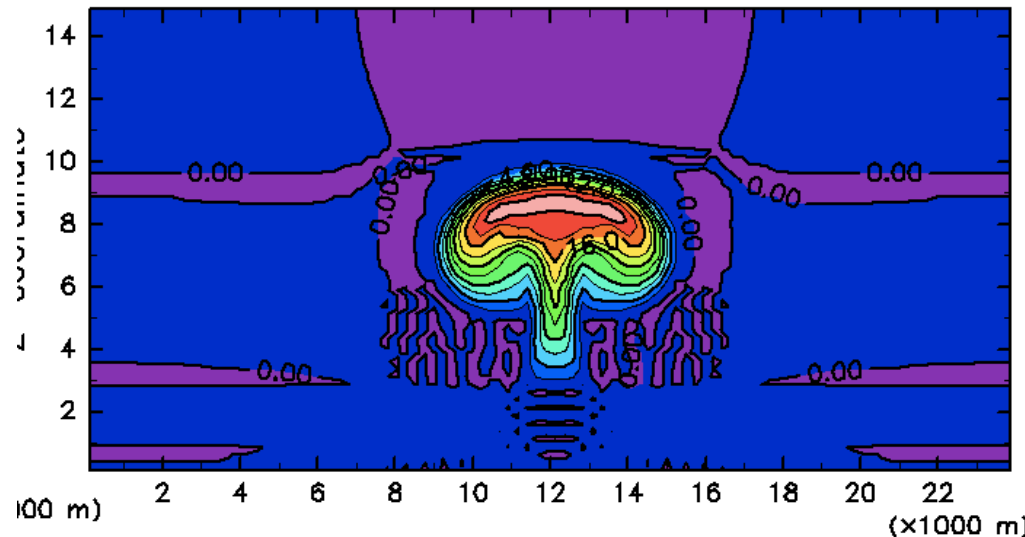


# 例題: サーマルの上昇

- 結果は変数毎に出力される.

```
$ gpview --anim t arare-earth_PotTemp.nc@PotTemp
```

^^^^^^^^^^ 変数



- 設定ファイルの詳細は以下を参照

<http://www.gfd-dennou.org/library/deepconv/arare/sample/TEMP-090309/rakuraku.htm.ja>

- Sugiyama et al. (2008) のリスタート
  - arare4 直下の arare-jupiter\_init.nc を利用して続きを計算
    - \$ ./bin/arare\_jupiter -N=arare-jupiter2.conf
  - 設定ファイルでリスタートファイルを指定
    - InitFile = "arare-jupiter2\_init.nc"
  - 計算にはかなりかかります..... おそらくノートPCでは厳しい

## ■ 描画

- gpview でも良いが, 柔軟に図を作成する場合はgphys を用いた ruby スクリプトを書く方が早い
- 実際に使った ruby スクリプトを同梱

```
$ cp ./tools/DrawJupiter.sh ./
$ ./DrawJupiter.sh
```
- ffmpeg がインストール済みなら, flv も作成

# 例題: 木星の雲対流

ymki@asura:~/work/c [X] qiv: images/arare-jupiter2\_024.jpg (908x604) 0.3s 100% [24/51] b0/c0/g0 (Next picture)

