

第9章

CReSS の設定と実行

本章では、*CReSS* の設定とデータフロー、そして、実行方法を説明する。

CReSS には、コンパイルコマンドやコンパイルオプションを設定するファイル `compile.conf` と、コンパイルや実行を制御する設定ファイル `user.conf` の2つの設定ファイルがある。本章では、これら2つの設定ファイルのうち、特に、コンパイルや実行を制御する設定ファイル `user.conf` の各種変数の設定方法を詳述する。この設定ファイルは Fortran のネームリストのフォーマットで記述されているが、実行の際はもちろんのこと、コンパイルの際にも読み込まれる。よって、ソースファイルの一部を書き換えることにより配列の大きさを変更するということはない。

次に、データフローと入出力ファイルについて説明する。また、それらの入出力ファイルのなかで、ユーザーにて操作する必要があるファイルのフォーマットについても説明する。

実行方法については、次に、ケルビンヘルムホルツ不安定波の実験例を用いて、主プログラムのコンパイルから実行、そして、結果のファイルを出力するまでの、必要となる一連の作業を説明する。さらに、プリプロセッサならびにポストプロセッサの実行方法を説明する。

9.1 設定ファイル

9.1.1 設定の注意点

CReSS の設定ファイルは Fortran のネームリスト形式であるが、別途、注意すべき点を 2 つ挙げる。

- この設定ファイルは CReSS をコンパイルするときにも読み込まれ、独自のシェルスクリプトによって処理される。この処理の都合上、変数名とその値の間の “=” の前後は 1 つ以上のスペースで区切られてなければならない。
- 変数によってはプリプロセッサと主プログラム *solver* の間に依存関係があり、そのような変数がプリプロセッサの設定と主プログラムの設定で合致しない場合（配列の大きさやオプションが違う等）には、エラーを出力し実行中のプログラムは停止する。

9.1.2 設定の詳細

& sysdep

ここでは、機種依存の変数を設定する。全てのプログラムに使用されるが、特に依存関係はない。

wlength *integer * 4*
直接探査の書式なしバイナリファイルの語長で、大抵の機種では 4 である。

& dmiset

ここでは、計算領域の分割数や格子数を設定する。全てのプログラムに使用され、それらの間に依存関係がある。

xpedim, ypedim *integer * 4*
モデル領域の *x, y* 方向の分割数を設定する。全体のノード数は $x \times y$ となる。

xdim, ydim *integer * 4*
モデルの *x, y* 方向の格子数を設定する。Arakawa C グリッドで格子は設定されるので、スカラー変数は各方向とも 1 少ない点数となる。また、各ノードが担当する格子数を同じにするため、以下の式を満たすように設定しなければならない。

$$(xdim - 3) / xpedim = integer$$

$$(ydim - 3) / ypedim = integer$$

zdim *integer * 4*
モデルの *z* 方向の格子数を設定する。Arakawa C グリッドで格子は設定されるので、スカラー変数は 1 少ない点数となる。また、プログラムにおいて仮配列の受け渡しの必要性から、 $cphopt \geq 2$ の場合 $zdim \geq 12$ 、 $sfcopt \geq 1$ の場合 $zdim \geq 10$ を設定しなければならない。

& expname

ここでは、モデル実行時の実験名を設定する。*solver*, *gridata*, *terrain* に使用され、依存関係がある。

exprim *character * 80*
 実験名をこの変数に設定する。なお、80文字以上の実験名を記述してはならない。そのまま、この変数は入出力されるファイル名の一部となる。例えばサウンディングファイルの場合には、*exprim.sounding.txt* のようになるので、ファイル名に使用できない特殊な文字は記述してはならない。

& project

ここでは、モデルの水平方向の座標系を設定する。*solver*, *gridata*, *terrain* に使用され、依存関係がある。

mpopt *integer * 4*
 モデルの *x, y* 座標系の地図投影法の選択オプション。以下の4種類の方法により座標系を設定できる。

- 1: 平射図法により座標系を設定
- 2: ランベルト正角円錐図法により座標系を設定
- 3: 正角円筒図法により座標系を設定
- 4: 地図投影をせず座標系を設定。格子点は正確に地球上の位置に反映されるが、矩形であるべき計算領域は地球に沿って曲がったものとなり、計算領域が広い場合には誤差を生じる。

nspol *integer * 4*
 モデル領域が地図座標系の場合に、投影の基点が北半球にあるか南半球にあるかのフラッグ。

- 1: モデル領域の座標系は北半球基点
- 1: モデル領域の座標系は南半球基点

tlat1, tlat2 *real * 4*
 地図投影座標系の基準緯度（この緯度では歪がない）を設定する。単位は [°] であり、南半球の緯度は負である。ランベルト正角円錐図法を用いる場合は、2つの基準緯度を設定 (**tlat2** も使用) することになる。地図投影座標系が指定されていない場合には、設定の必要はない。

tlon *real * 4*
 モデル座標系の基準経度（この経度でモデル領域の *y* 軸は地球の経度線と一致する）を設定する。上記の基準緯度と違い、**mpopt = 4** の場合にも設定しなければならない。単位は [°] であり、西経の経度は負である。

& gridset

ここでは、モデルの格子間隔や基準となる格子点とその緯度・経度を設定する。*solver*, *gridata*, *terrain* に使用され、依存関係がある。

dx,dy,dz	<i>real</i> * 4 モデルの <i>x, y, z</i> 各方向の格子間隔を設定する。単位は [m] である。dz については、格子のストレッチングを設定するとき (<i>sthopt</i> = 1,2) には平均の格子間隔となる。
ulat,ulon	<i>real</i> * 4 モデルの領域のある 1 点の緯度・経度を指定する。単位は [°] であり、南半球の緯度と西経の経度は負である。これらと対応するモデル領域での実数添字 (すぐ下の <i>riu,rju</i>) から、モデルは領域設定を実行する。
riu,rju	<i>real</i> * 4 上で設定した <i>ulat,ulon</i> のモデル領域での実数添字を指定する。

& gridsth

ここでは、モデルの鉛直格子のストレッチングを設定する。*solver*, *gridata* に使用され、依存関係がある。

zsfc	<i>real</i> * 4 地表面の標高を設定する。単位は [m] である。通常は 0.0 m を設定する。
zflat	<i>real</i> * 4 平滑面の最低の高さを設定する。単位は [m] である。
sthopt	<i>integer</i> * 4 鉛直方向の格子のストレッチングのオプション。 0: ストレッチングを実行しない 1: 3 次関数でストレッチングを実行 2: tanh 関数でストレッチングを実行
dzmin	<i>real</i> * 4 格子のストレッチングの際、最下層の最小の dz を設定する。単位は [m] である。ストレッチングを適用しない場合には、dz に強制される。
layer1,layer2	<i>real</i> * 4 格子のストレッチングの際、 <i>layer1</i> までの高さでは <i>dzmin</i> の格子間隔で、 <i>layer1</i> から <i>layer2</i> までの高さでは <i>sthopt</i> による関数を適用して、 <i>layer2</i> 以上の高さでは <i>layer2</i> の高さでの格子間隔で格子を設定する。単位は [m] である。

& terrain

ここでは、モデルの標高を設定する。*solver*, *gridata* に使用され、依存関係がある。

trnopt	<i>integer</i> * 4 モデルの標高の設定オプション。 0: 平滑な地面 1: ベル型の山により標高を設定 2: 標高ファイルを読み込むことにより任意の標高を設定
mntgh	<i>real</i> * 4 trnopt = 1 のときベル型の山の高さを設定する。単位は [m] である。
mntwx, mntwy	<i>real</i> * 4 trnopt = 1 のときベル型の山の <i>x, y</i> 方向の半値幅（山の高さが半分になる中心からの距離）を設定する。単位は [m] である。
mntcx, mntcy	<i>real</i> * 4 trnopt = 1 のときベル型の山の中心の <i>x, y</i> 座標値を設定する。単位は [m] である。配列番号と座標値との関係は配列番号をそれぞれ <i>i, j</i> として、次のとおりである。 $x = (i - 2) \times dx$ $y = (j - 2) \times dy$

& flength

ここでは、予報時間に関する変数を設定する。*solver*, *gridata* に使用され、* 付きの変数には依存関係がある。

sfcast*	<i>character</i> * 16 予報開始の年月日と時刻をグリニッジ標準時 (UTC) で設定する。その形式は、 sfcast = ' <i>yyyy/mm/dd hh:mm</i> ' である。例えば 2000 年 9 月 22 日 13 時 00 分のときは、 sfcast = '2000/09/22 13:00' のように記述する。
stime	<i>real</i> * 4 予報開始時間を設定する。単位は [s] である。変数は sfcast からの経過時間を表しているの、初めて実行するときは 0.0 s、経過時間 3600.0 s からのリスタートのときは 3600.0 s と指定する。
etime	<i>real</i> * 4 予報終了時間を設定する。単位は [s] である。変数は sfcast からの経過時間を表しているの、例えば、経過時間 3600.0 s からのリスタートで 1800.0 s 間の予報をしたいときは 5400.0 s と指定する。

& gpvpram

ここでは、外部気象データファイルとそれを用いる過程に関する変数を設定する。*solver*, *gridata* に使用され、* 付きの変数には依存関係がある。

gpvvar*	<i>character * 80</i> 外部気象データファイルにおいて、モデルに入力させる変数を設定する。80 文字のうち最初の 7 文字のみ使用する。並びは順に、鉛直速度成分、水蒸気混合比（または相対湿度）、雲水混合比、雨水混合比、雲氷混合比、雪混合比、霰混合比である。なお、 <i>z</i> 座標、水平速度、圧力、温位（または温度）は外部データに必須な変数なので、この設定には含まれない。例えば、必須の変数の他に、鉛直速度成分、水蒸気混合比（または相対湿度）、雨水混合比を入力させる（それらの変数がデータとして存在する）場合には、 <i>gpvvar</i> = 'ooxoxxxx' となる。 o: その変数を入力させる x: その変数を入力させない
gpvitr*	<i>real * 4</i> 外部気象データファイルの存在する時間間隔を設定する。単位は [s] である。
nggopt*	<i>integer * 4</i> 外部気象データに対するアナリシスナッジングの設定オプション。 0: アナリシスナッジングを実行しない 1: アナリシスナッジングを実行する
nggvar	<i>character * 80</i> ナッジングを適用する変数を設定する。80 文字のうち最初の 11 文字のみ使用する。並びは順に、東西速度成分、南北速度成分、鉛直速度成分、圧力、温位（または温度）、水蒸気混合比（または相対湿度）、雲水混合比、雨水混合比、雲氷混合比、雪混合比、霰混合比である。例えば、各速度成分、圧力、温位にナッジングを適用する場合には、 <i>nggvar</i> = 'oooooxxxxxx' となる。外部気象データにおける必須変数以外は、 <i>gpvvar</i> に適合していなければならない。 o: その変数にナッジングを適用する x: その変数にナッジングを適用しない
nggcoe	<i>real * 4</i> アナリシスナッジングの緩和係数を設定する。
nggdlt	<i>real * 4</i> ナッジングを適用する時間間隔を設定する。単位は [s] である。 <i>dtb</i> の整数倍でなければならず、また、 <i>dtb</i> 以上でなければならない。
nggend	<i>real * 4</i> ナッジングを終了する予報開始時刻からの経過時間を設定する。単位は [s] である。

exbopt*	<i>integer</i> * 4 側面境界の外部データへの強制オプション。 0: 側面境界で予報変数を外部気象データに強制しない 1: 側面境界で外部気象データに強制する 2: 側面境界で外部気象データに強制し、スポンジ層で標高を緩和する 11: 1 にさらに側面における質量調節を実行する 12: 2 にさらに側面における質量調節を実行する
exbvar	<i>character</i> * 80 外部データへの強制を適用する変数を設定する。設定方法は、 nggvar と同様である。
exbcoe	<i>real</i> * 4 側面境界の緩和係数を設定する。
corssp*	<i>integer</i> * 4 側面スポンジ層での標高の緩和の層数を設定する
lspopt*	<i>integer</i> * 4 側面スポンジ層での予報変数の緩和の設定オプション。 0: 予報変数を緩和しない 1: 外部データの値に予報変数を緩和する 2: 予報開始時刻の値に予報変数を緩和する
lspvar	<i>character</i> * 80 側面スポンジ層における緩和を適用する変数を設定する。設定方法は、 nggvar とほぼ同様であるが、水平速度成分に関しては、 o,x の他に '-' として面に垂直な成分にのみ適用させることもできる。
lspcoe	<i>real</i> * 4 側面スポンジ層の緩和係数の最大値を設定する。
lspspn	<i>integer</i> * 4 側面スポンジ層の層数を設定する。
vspopt*	<i>integer</i> * 4 上面スポンジ層の予報変数の緩和の設定オプション。 0: 予報変数を緩和しない 1: 外部データに予報変数を緩和する 2: 予報開始時刻の値に予報変数を緩和する
vspvar	<i>character</i> * 80 上面スポンジ層における緩和を適用する変数を設定する。設定方法は、 nggvar と同様である。

<code>vspcoe</code>	<i>real</i> * 4 上面スポンジ層の緩和係数の最大値を設定する。
<code>zsplow</code>	<i>real</i> * 4 緩和を適用する最低高度を設定する。単位は [m] である。

& boundry

ここでは、境界条件の設定をする。*solver*, *gridata* に使用され、* 付きの変数には依存関係がある。

<code>wbc*</code> , <code>ebc*</code> , <code>sbc*</code> , <code>nbc*</code>	<i>integer</i> * 4 側面境界条件のオプション。それぞれ順に、西・東・南・北面。 1: 周期境界条件を適用 2: 固定壁境界条件を適用 3: 勾配 0 の境界条件を適用 4: 側面境界の各点毎に位相速度を計算して、放射境界条件を適用 5: 鉛直方向に平均化された位相速度を用いて、放射境界条件を適用 6: 一定の位相速度と壁の法線方向の移流速度を用いて、放射境界条件を適用 7: 一定の位相速度を用いて、放射境界条件を適用
<code>bbc</code> , <code>tbc</code>	<i>integer</i> * 4 下端・上端の境界条件のオプション。鉛直陰解法を適用 (<code>impopt = 1</code>) する場合には <code>bbc = 2</code> , <code>tbc = 2</code> に強制される。 2: 固定壁境界条件を適用 3: 勾配 0 の境界条件を適用
<code>lbccoe</code>	<i>real</i> * 4 側面境界の緩和係数を設定する (放射境界選択時のみ)。

& radpram

ここでは、外部レーダーデータファイルとそれを用いる過程に関する変数を設定する。*solver*, *radata* に使用され、* 付きの変数には依存関係がある。

<code>radvar*</code>	<i>character</i> * 80 外部レーダーデータファイルにおいて、モデルに入力させる変数を設定する。80 文字のうち最初の 4 文字のみ使用する。並びは順に、東西速度成分、南北速度成分、鉛直速度成分、降水物質の混合比 (または反射強度) である。例えば、水平速度成分を入力させる (それらの変数がデータとして存在する) 場合には、 <code>radvar = 'ooxx'</code> となる。データの各点の高さは必ず入力させなければならない。 o: その変数を入力させる x: その変数を入力させない
----------------------	---

raditv*	<i>real * 4</i> 外部レーダーデータファイルの存在する時間間隔を設定する。単位は [s] である。
ngropt*	<i>integer * 4</i> レーダーデータに対するアナリシスナッジングの設定オプション。 0: アナリシスナッジングを実行しない 1: アナリシスナッジングを <i>radata</i> の出力ファイルを読み込み実行する 2: アナリシスナッジングを外部データを直接読み込み実行する
ngrvar	<i>character * 80</i> ナッジングを適用する変数を設定する。80 文字のうち最初の 6 文字のみ使用する。並びは順に、東西速度成分、南北速度成分、鉛直速度成分、雨水混合比、雪混合比、霰混合比である。例えば、水平速度成分と各混合比にナッジングを適用する場合には、 ngrvar = 'ooxooo' となる。 radvar に適合していなければならない。 o: その変数にナッジングを適用する x: その変数にナッジングを適用しない
ngrcoe	<i>real * 4</i> アナリシスナッジングの緩和係数を設定する。
ngrdlt	<i>real * 4</i> ナッジングを適用する時間間隔を設定する。単位は [s] である。 dtb の整数倍でなければならない。また、 dtb 以上でなければならない。
ngrstr	<i>real * 4</i> ナッジングを開始する予報開始時刻からの経過時間を設定する。単位は [s] である。
ngrend	<i>real * 4</i> ナッジングを終了する予報開始時刻からの経過時間を設定する。単位は [s] である。
ngraff	<i>real * 4</i> ナッジングを適用する影響時間（読み込み時刻からの時間方向の影響半径）を設定する。単位は [s] である。

& sfcphys

ここでは、地表面過程の各種変数を設定する。*solver*, *surface* に使用され、* 付きの変数には依存関係がある。

sfcopt	<i>integer * 4</i> 地表面過程の設定オプション。 0: 地表面過程を実行しない 1: 地表面過程を実行する 2: 地表面過程を過去の実行で得た地中温度データを用いて実行する
---------------	---

levpbl	<i>integer</i> * 4 境界層の層数を設定する。各層の幅は dz であり、 $1 \leq \text{levpbl} \leq \text{zdim} - 2$ の範囲で設定できる。
levgrd	<i>integer</i> * 4 地中温度の計算の層数を設定する。
dzgrd	<i>real</i> * 4 地中温度の計算の各層の幅を設定する。単位は [m] である。levgrd×dzgrd が数メートルの深さになるように設定する必要がある。
ctgr	<i>real</i> * 4 過去の実行における地中温度データの入力がない場合に、最深層に一定の地中温度を設定する。単位は [K] である。
prvres	<i>character</i> * 80 過去の実行における地中温度を用いる場合に、その実行時に得た拡張子を除いたリストファイル名を指定する。
sfcdat*	<i>character</i> * 80 地表面過程に対する外部データの入力の有無を指定する。80文字のうち最初の2文字のみ使用する。並びは順に、土地利用、海水面温度の各データの入力の有無である。例えば、土地利用データのみ入力させる場合には、sfcdat = 'ox' となる。なお、海水面温度は予報期間内で変化しないとしている。 o: データファイルがあり、それを入力させる x: そのデータを入力させない
cbeta	<i>real</i> * 4 土地利用データの入力がない場合に、一定の蒸発散係数を設定する。この場合には標高値が正の格子点が陸面として判断され、対応する点にこの係数が設定される。
calbe	<i>real</i> * 4 土地利用データの入力がない場合に、一定のアルベドを設定する。全く同様に、標高値が正の格子点に設定される。
cz0m	<i>real</i> * 4 土地利用データの入力がない場合に、一定の粗度を設定する。単位は [m] である。全く同様に、標高値が正の格子点に設定される。
csst	<i>real</i> * 4 海面温度データの入力がない場合に、一定の海面温度を設定する。単位は [K] である。この場合には標高値が負の格子点が海面として判断され、対応する点にこの温度が設定される。

& initype

ここでは、モデルの初期化の方法を設定する。*solver* のみに使用される。

iniopt	<i>integer</i> * 4 モデルの初期化のオプション。 1: サウンディングファイルを読み込み実行を開始 2: リスタートファイルを読み込み実行を開始 3: 3次元格子点値ファイルを読み込み実行開始
snddim	<i>integer</i> * 4 iniopt = 1 のとき、サウンディングファイルのデータ層数を設定する。
sndhed	<i>integer</i> * 4 iniopt = 1 のとき、サウンディングファイルのヘッダのコメント行数を設定する。 ヘッダのみ有効でフッタは無効である。
sndtyp	<i>character</i> * 80 iniopt = 1 のとき、サウンディングファイルのデータの種類を指定する。水平速度成分以外を以下のような組み合わせで設定できる。80文字のうち最初の3文字のみ使用する。 ppk: 第1列は圧力 [Pa] 第2列は温位 [K] 第5列は水蒸気混合比 [kg kg ⁻¹] zpk: 第1列は高さ [m] 第2列は温位 [K] 第5列は水蒸気混合比 [kg kg ⁻¹] ptk: 第1列は圧力 [Pa] 第2列は温度 [K] 第5列は水蒸気混合比 [kg kg ⁻¹] ztk: 第1列は高さ [m] 第2列は温度 [K] 第5列は水蒸気混合比 [kg kg ⁻¹] ppp: 第1列は圧力 [Pa] 第2列は温位 [K] 第5列は相対湿度 [%] zpp: 第1列は高さ [m] 第2列は温位 [K] 第5列は相対湿度 [%] ptp: 第1列は圧力 [Pa] 第2列は温度 [K] 第5列は相対湿度 [%] ztp: 第1列は高さ [m] 第2列は温度 [K] 第5列は相対湿度 [%]
zsnd0, psnd0	<i>real</i> * 4 サウンディングデータの最低点の高さもしくは圧力を設定する。単位はそれぞれ [m]、[Pa] である。 sndtyp (1:1) = 'p' のときは zsnd0 が、 sndtyp (1:1) = 'z' のときは psnd0 が使用される。
masopt	<i>integer</i> * 4 マスコンモデルの実行オプション。 0: 実行開始時にマスコンモデルを適用しない 1: 実行開始時にマスコンモデルを適用する
alpha1, alpha2	<i>real</i> * 4 masopt = 1 のとき、水平・鉛直方向の大気安定度を示す重みづけ係数を設定する。

& gridmove

ここでは、計算領域の移動に関する設定をする。solver のみに使用される。

movopt	<i>integer</i> * 4
	格子の移動オプション。trnopt = 0, iniopt = 1, exbopt = 0 の場合にのみ適用できる。
	0: 格子を移動させない
	1: 格子を移動させる
umove, vmove	<i>real</i> * 4
	格子移動時の東西・南北方向の移動速度を設定する。単位は [m s ⁻¹] である。

& ptinicon

ここでは、予報開始時刻の温位擾乱に関する設定をする。solver のみに使用される。

pt0opt	<i>integer</i> * 4
	予報開始時刻における温位擾乱の設定オプション。iniopt = 1 の場合に適用できる。
	0: 初期擾乱を設定しない
	1: 球形の初期擾乱を <i>x</i> 方向に複数個設定する
	2: 球形の初期擾乱を <i>y</i> 方向に複数個設定する
	3: 三角関数形の初期擾乱を <i>x</i> 方向に設定する
	4: 三角関数形の初期擾乱を <i>y</i> 方向に設定する
	5: ランダム擾乱を指定した 2 点の高度間に設定する
pt0num	<i>integer</i> * 4
	pt0opt = 1 または 2 のとき、球形擾乱の個数を設定する。
ptp0	<i>real</i> * 4
	温位擾乱の最大値を設定する。単位は [K] である。
pt0rx, pt0ry, pt0rz	<i>real</i> * 4
	並びの温位擾乱の半径または半波長または領域の半値幅 (pt0opt = 5 のときは <i>z</i> 方向のみ) を設定する。単位は [m] である。
pt0cx, pt0cy, pt0cz	<i>real</i> * 4
	並びの温位擾乱の中心または原点または領域中心点の <i>x, y, z</i> 座標値 (pt0opt = 5 のときは <i>z</i> 方向のみ) を設定する。単位は [m] である。
pt0ds	<i>real</i> * 4
	pt0opt = 1 または 2 のとき、各バブル間の距離を設定する。単位は [m] である。

& integrat

ここでは、タイムステップや Asselin フィルターなど、時間積分に関する設定をする。*solver* のみに使用される。

dtbig,dtsml	<i>real</i> * 4 それぞれ、音波モードに関係しない項と音波モードに関係する項の時間積分のタイムステップを設定する。 gwmopt = 1 の場合には、重力波モードも dtsml に含まれる。単位は [s] である。CFL 条件に注意して設定しなければならない。
gwmopt	<i>integer</i> * 4 重力波モードに関する時間積分法の設定オプション。 0: 音波モードを小さいタイムステップの時間積分に含めない 1: 音波モードを小さいタイムステップの時間積分に含める
impopt	<i>integer</i> * 4 音波モードの鉛直方向の時間積分法の設定オプション。通常 impopt = 0 もしくは impopt = 1 でよい。 0: 水平・鉛直方向とも陽解法で時間積分する 1: ガウスの消去法により鉛直方向を陰解法で時間積分する 2: ピボット選択付きガウスの消去法により鉛直方向を陰解法で時間積分する 3: ガウス・ザイデル法により鉛直方向を陰解法で時間積分する
weicoe	<i>real</i> * 4 陰解法で時間積分するとき、求められるタイムステップにおける重み付け係数を設定する。小さいタイムステップを $\Delta\tau$ として、時間 τ が $1-\text{weicoe}$ 、時間 $\tau + \Delta\tau$ が weicoe の重みで評価される。
filcoe	<i>real</i> * 4 Asselin フィルターの重み付け係数を設定する。

& advction

ここでは、移流項の計算方法に関する設定をする。*solver* のみに使用される。

ad4opt	<i>integer</i> * 4 移流項の計算オプション。中心差分法が適用されているので、数値粘性項を設定しないと計算不安定が発生する。 0: 2次精度の中心差分で移流項を計算する 1: 4次精度の中心差分で移流項を計算する
---------------	---

& smoother

ここでは、数値粘性項に関する設定をする。*solver* のみに使用される。

smtopt	<i>integer</i> * 4 数値粘性項の計算オプション。移流項に付加される非物理的な粘性である。 0: 数値粘性項を方程式に付加しない 1: 2次の数値粘性項を方程式に付加する 2: 4次の数値粘性項を方程式に付加する
smndch	<i>real</i> * 4 水平方向の粘性係数の無次元係数を設定する。大きいタイムステップで割った値が次のような範囲にあるのが妥当である。 2次の場合: $\text{smndch}/\text{dtb} \leq 1/8$ 4次の場合: $\text{smndch}/\text{dtb} \approx 1/1000$
smndcv	<i>real</i> * 4 鉛直方向の粘性係数の無次元係数を設定する。水平方向の場合と同様、大きいタイムステップで割った値が次のような範囲にあるのが妥当である。 2次の場合: $\text{smndcv}/\text{dtb} \leq 1/8$ 4次の場合: $\text{smndcv}/\text{dtb} \approx 1/1000$

& mapfcter

ここでは、マップスケールファクターに関する設定をする。*solver* のみに使用される。

mfcopt	<i>integer</i> * 4 マップスケールファクターの設定オプション。 0: マップスケールファクターを設定しない 1: マップスケールファクターを設定する
---------------	--

& coriolis

ここでは、コリオリ力に関する設定をする。*solver* のみに使用される。

coropt	<i>integer</i> * 4 コリオリ力の計算オプション。鉛直方向のコリオリ力は省略してもよい。 0: コリオリ力を計算しない 1: 水平方向のみコリオリ力を計算する 2: 鉛直方向のコリオリ力も計算する
---------------	--

& earthcrv

ここでは、地球の曲率の影響に関する設定をする。*solver* のみに使用される。

coropt *integer * 4*
地球の曲率の影響項の計算オプション。
0: 地球の曲率の影響項を計算しない
1: 地球の曲率の影響項を計算する

& buoyancy

ここでは、浮力の項に関する設定をする。*solver* のみに使用される。

buyopt *integer * 4*
浮力項の計算オプション。
0: 浮力項を計算しない
1: 浮力項を計算する

& diabatic

ここでは、断熱項に関する設定をする。*solver* のみに使用される。

diaopt *integer * 4*
断熱項の計算オプション。
0: 断熱項を計算しない
1: 断熱項を計算する

& ddamping

ここでは、圧力勾配項の緩和に関する設定をする。*solver* のみに使用される。

divopt *integer * 4*
圧力勾配項の緩和のオプション。計算不安定を抑えるための非物理的な緩和である。
0: 圧力勾配項の緩和を実行しない
1: 圧力勾配項の緩和を実行する

divndc *real * 4*
無次元の緩和係数を設定する。*divndc* = 0.05 程度が一般的である。

& turbulen

ここでは、サブグリッドスケールの乱流過程に関する設定をする。*solver* のみに使用される。

tubopt	<i>integer</i> * 4 乱流過程の設定オプション。 0: 乱流過程を実行しない 1: スマゴリンスキーモデルを実行 2: 1.5 次のクロージャーモデルを実行
isoopt	<i>integer</i> * 4 格子が等方か非等方かの設定オプション。 1: 格子間隔は水平・鉛直に等方的である 2: 格子間隔は水平・鉛直に非等方的である
prnum	<i>real</i> * 4 プラントル数を設定する ($Pr = \nu_\tau / \nu_H$)。 tubopt = 2 のときは安定性の検査のみに使用される。
csnum	<i>real</i> * 4 tubopt = 1 のときスマゴリンスキー定数を設定する。

& cloudphy

ここでは、雲微物理過程に関する設定をする。*solver* のみに使用される。

cphopt	<i>integer</i> * 4 雲微物理過程の設定オプション。 0: 雲微物理過程を実行しない 1: 暖かい雨のバルク法のパラメタリゼーションを実行 2: 氷相を含むバルク法のパラメタリゼーションを実行 3: 2 に氷物質 (雲氷・雪・霰) の数密度の予報を含めたパラメタリゼーションを実行
numnc	<i>real</i> * 4 cphopt = 2 または 3 のときに、一定の雲水の数密度を設定する。単位は $[m^{-3}]$ である。

& outfomat

ここでは、出力されるファイルのフォーマットや時間間隔を設定する。*solver* のみに使用される。

dmpfmt	<i>integer * 4</i> ヒストリファイルと地理情報ファイルの出力形式の設定オプション。 1: 並びのテキスト書式でファイルを出力 2: 直接探査の書式なしバイナリでファイルを出力
dmpcmp	<i>integer * 4</i> ヒストリファイルの鉛直座標系の設定オプション。 dmpcmp = 1 以外の場合には、地中や計算領域の上端より高い点には、未定義値として非常に大きな負の値 (-1.0×10^{35}) が設定される。 1: 計算領域と同じ z^* 系 (ζ) 系で出力 2: dz の間隔で、一定の水平面に補間して出力 3: ストレッチングの関数を適用した間隔で、一定の水平面に補間して出力
dmpitv	<i>real * 4</i> ヒストリファイルの出力の時間間隔を設定する。単位は [s] である。この間隔はリスタートを含まない予報開始時刻 stime = 0.e0 からの時間である。例えば、この時間間隔の設定が 300.0 s でリスタート時間 450.0 s で実行した場合、最初の出力時間は 600.0 s である。
resitv	<i>real * 4</i> リスタートファイルの出力の時間間隔を設定する。単位は [s] である。 dmpitv と同様、この間隔はリスタートを含まない予報開始時刻からの時間である。
mxnitv	<i>real * 4</i> 予報変数の最大値と最小値を標準出力ファイルに出力する時間間隔を設定する。単位は [s] である。 dmpitv , resitv と同様、この間隔はリスタートを含まない予報開始時刻からの時間である。

& project_grd

ここでは、外部気象データファイルの座標系を設定する。*gridata* に使用され、依存関係はない。

mpopt_grd	<i>integer * 4</i> 外部気象データファイルの座標系を指定する。5種類の座標系を指定できる。 0: データは緯度・経度座標系である。 1: データは平射図法により座標系が設定されている 2: データはランベルト正角円錐図法により座標系が設定されている 3: データは正角円筒図法により座標系が設定されている 4: データは地図投影によらない座標系が設定されている
------------------	---

nspol_grd	<i>integer</i> * 4	外部気象データが地図座標系の場合に、投影の基点が北半球にあるか南半球にあるかのフラッグ。 1: 外部データの地図座標系は北半球基点 -1: 外部データの地図座標系は南半球基点
tlat1_grd, tlat2_grd	<i>real</i> * 4	地図投影座標系の基準緯度を指定する。単位は [°] であり、南半球の緯度は負である。ランベルト正角円錐図法が指定されている場合は、2つの基準緯度を指定 (tlat2_grd も使用) することになる。緯度・経度座標系の場合には、設定の必要はない。
tlon_grd	<i>real</i> * 4	地図投影座標系の基準経度を指定する。単位は [°] であり、西経の経度は負である。

& gridset_grd

ここでは、外部気象データの格子間隔や基準となる格子点とその緯度・経度を設定する。griddata に使用され、依存関係はない。

xdim_grd, ydim_grd, zdim_grd	<i>integer</i> * 4	外部気象データの x, y, z 方向の格子数を設定する。外部データに地表面 (場合によっては海面) の物理量が含まれる場合 (refsfrc = 1) でも、鉛直方向の最初の添え字に読み込ませるように設計されている、つまり、同じ 3 次元配列に読み込ませるようになっており、refsfrc の設定に関わらず最初の添え字はいつも使用されるので注意が必要である。
dx_grd, dy_grd	<i>real</i> * 4	外部気象データの x, y 各方向の格子間隔を設定する。単位は、mpopt_grd = 0 の場合は [°]、それ以外の場合は [m] である。鉛直方向の格子間隔については、外部気象データに鉛直座標値が含まれていると仮定しているので設定しない。
ulat_grd, ulon_grd	<i>real</i> * 4	外部気象データの領域のある 1 点の緯度・経度を指定する。単位は [°] であり、南半球の緯度と西経の経度は負である。これらと対応する外部データの領域での実数添字 (すぐ下の riu_grd, rju_grd) から、モデル領域との対応関係が得られる。
riu_grd, rju_grd	<i>real</i> * 4	上で設定した ulat_grd, ulon_grd の外部データの領域での実数添字を指定する。

& how2intp_grd

ここでは、外部気象データからモデル格子への補間方法を設定する。*gridata*, に使用され、依存関係はない。

biiopt_grd *integer * 4*
 外部気象データのモデル領域への補間方法のオプション。ここで設定されるのは水平方向の補間方法の選択のみで、客観解析値などの3次元データの鉛直方向の補間では、常に1次の線型補間が適用されるので、該当するオプションはない。

- 0: 1次の線型補間を適用
- 1: 2次の放物線補間を適用

& dataacf_grd

ここでは、客観解析値などの3次元の格子点データファイルのフォーマットとプリプロセッサ *gridata* の動作に関する設定をする。

datatype_grd *character * 80*
 客観解析データなどの外部気象データの変数のうち、温度または温位、水蒸気混合比または相対湿度のどちらを入力するかを指定するオプション。80文字のうち最初の2文字のみ使用する。

- tk: 入力変数の1つは温度 [K]、もう1つは水蒸気混合比 [kg kg⁻¹]
- pk: 入力変数の1つは温位 [K]、もう1つは水蒸気混合比 [kg kg⁻¹]
- tp: 入力変数の1つは温度 [K]、もう1つは相対湿度 [%]
- pp: 入力変数の1つは温位 [K]、もう1つは相対湿度 [%]

refsfc *integer * 4*
 客観解析データなどの外部気象データの補間の際に、データの地表面（場合によっては海面）の高度と物理量を参照するかどうかのフラッグ。このフラッグを設定した場合、鉛直方向の最初の添え字が地表面として扱われる。地表面のデータが別途存在するような場合には、ユーザーが変更する必要があるサブルーチン *rdobj.f* での読み込みの際には注意が必要である。

- 0: 外部データの標高と地表面の物理量を参照しない
- 1: 外部データの標高と地表面の物理量を参照する

& project_rad

ここでは、外部レーダーデータファイルの座標系を設定する。*radata* に使用され、依存関係はない。

mpopt_rad *integer * 4*
 外部レーダーデータファイルの座標系を指定する。5種類の座標系を指定できる。

- 0: データは緯度・経度座標系である。
- 1: データは平射図法により座標系が設定されている
- 2: データはランベルト正角円錐図法により座標系が設定されている
- 3: データは正角円筒図法により座標系が設定されている
- 4: データは地図投影によらない座標系が設定されている

<code>nspol_rad</code>	<i>integer</i> * 4 外部レーダーデータが地図座標系の場合に、投影の基点が北半球にあるか南半球にあるかのフラッグ。 1: 外部データの地図座標系は北半球基点 -1: 外部データの地図座標系は南半球基点
<code>tlat1_rad</code> , <code>tlat2_rad</code>	<i>real</i> * 4 地図投影座標系の基準緯度を指定する。単位は [°] であり、南半球の緯度は負である。ランベルト正角円錐図法が指定されている場合は、2つの基準緯度を指定 (<code>tlat2_rad</code> も使用) することになる。緯度・経度座標系の場合には、設定の必要はない。
<code>tlon_rad</code>	<i>real</i> * 4 地図投影座標系の基準経度を指定する。単位は [°] であり、西経の経度は負である。

& gridset_rad

ここでは、外部レーダーデータの格子間隔や基準となる格子点とその緯度・経度を設定する。`radata` に使用され、依存関係はない。

<code>xdim_rad</code> , <code>ydim_rad</code> , <code>zdim_rad</code>	<i>integer</i> * 4 外部レーダーデータの x, y, z 方向の格子数を設定する。
<code>dx_rad</code> , <code>dy_rad</code>	<i>real</i> * 4 外部レーダーデータの x, y 各方向の格子間隔を設定する。単位は、 <code>mpopt_rad = 0</code> の場合は [°]、それ以外の場合は [m] である。鉛直方向の格子間隔については、外部レーダーデータに鉛直座標値が含まれていると仮定しているので設定しない。
<code>ulat_rad</code> , <code>ulon_rad</code>	<i>real</i> * 4 外部レーダーデータの領域のある1点の緯度・経度を指定する。単位は [°] であり、南半球の緯度と西経の経度は負である。これらと対応する外部データの領域での実数添字 (すぐ下の <code>riu_rad, rju_rad</code>) から、モデル領域との対応関係が得られる。
<code>riu_rad</code> , <code>rju_rad</code>	<i>real</i> * 4 上で設定した <code>ulat_rad, ulon_rad</code> の外部データの領域での実数添字を指定する。

& datacf_rad

ここでは、モデル入力用レーダーファイルのフォーマットとプリプロセッサ *radata* の動作に関する設定をする。

datatype_rad	<i>character * 80</i> レーダーデータの変数のうち、降水物質に関しての単位を指定するオプション。80文字のうち最初の2文字のみ使用する。 z: 降水物質は反射強度 [dBZe] で与えられる k: 降水物質は混合比 [kg kg ⁻¹] で与えられる
radcoe, radpwr	<i>real * 4</i> 降水物質の単位を反射強度 [dBZe] から混合比 [kg kg ⁻¹] に変換する際の変換式の係数と指数を指定する。なお、プログラム <i>solver</i> にて変数が使用されるときに大気密度で除されるために、モデル入力用ファイルへの出力段階ではその単位は粒子密度 [kg m ⁻³] となっている。

& project_trn

ここでは、標高データファイルの座標系を設定する。*terrain* に使用され、依存関係はない。

mpopt_trn	<i>integer * 4</i> 標高データファイルの座標系を指定する。5種類の座標系を指定できる。 0: データは緯度・経度座標系である。 1: データは平射図法により座標系が設定されている 2: データはランベルト正角円錐図法により座標系が設定されている 3: データは正角円筒図法により座標系が設定されている 4: データは地図投影によらない座標系が設定されている
nspol_trn	<i>integer * 4</i> 標高データが地図座標系の場合に、投影の基点が北半球にあるか南半球にあるかのフラッグ。 1: 外部データの地図座標系は北半球基点 -1: 外部データの地図座標系は南半球基点
tlat1_trn, tlat2_trn	<i>real * 4</i> 地図投影座標系の基準緯度を指定する。単位は [°] であり、南半球の緯度は負である。ランベルト正角円錐図法が指定されている場合は、2つの基準緯度を指定 (tlat2_trn も使用) することになる。緯度・経度座標系の場合には、設定の必要はない。
tlon_trn	<i>real * 4</i> 地図投影座標系の基準経度を指定する。単位は [°] であり、西経の経度は負である。

& gridset_trn

ここでは、標高データの格子間隔や基準となる格子点とその緯度・経度を設定する。*terrain* に使用され、依存関係はない。

<code>xdim_trn,</code> <code>ydim_trn,</code>	<i>integer</i> * 4 標高データの <i>x,y</i> 方向の格子数を設定する。
<code>dx_trn,</code> <code>dy_trn</code>	<i>real</i> * 4 標高データの <i>x,y</i> 各方向の格子間隔を設定する。単位は、 <code>mpopt_trn = 0</code> の場合は [°]、それ以外の場合は [m] である。
<code>ulat_trn,</code> <code>ulon_trn</code>	<i>real</i> * 4 標高データの領域のある 1 点の緯度・経度を指定する。単位は [°] であり、南半球の緯度と西経の経度は負である。これらと対応する外部データの領域での実数添字（すぐ下の <code>riu_trn,rju_trn</code> ）から、モデル領域との対応関係が得られる。
<code>riu_trn,</code> <code>rju_trn</code>	<i>real</i> * 4 上で設定した <code>ulat_trn,ulon_trn</code> の外部データの領域での実数添字を指定する。

& how2intp_trn

ここでは、標高データからモデル格子への補間方法を設定する。*terrain*, に使用され、依存関係はない。

<code>biiopt_trn</code>	<i>integer</i> * 4 標高データのモデル領域への補間方法のオプション。 0: 1 次の線型補間を適用 1: 2 次の放物線補間を適用
-------------------------	---

& project_lnd

ここでは、土地利用データファイルの座標系を設定する。*surface* に使用され、依存関係はない。

<code>mpopt_lnd</code>	<i>integer</i> * 4 土地利用データファイルの座標系を指定する。5 種類の座標系を指定できる。 0: データは緯度・経度座標系である。 1: データは平射図法により座標系が設定されている 2: データはランベルト正角円錐図法により座標系が設定されている 3: データは正角円筒図法により座標系が設定されている 4: データは地図投影によらない座標系が設定されている
------------------------	--

<code>nspol_lnd</code>	<i>integer</i> * 4	土地利用データが地図座標系の場合に、投影の基点が北半球にあるか南半球にあるかのフラッグ。 1: 外部データの地図座標系は北半球基点 -1: 外部データの地図座標系は南半球基点
<code>tlat1_lnd,</code> <code>tlat2_lnd</code>	<i>real</i> * 4	地図投影座標系の基準緯度を指定する。単位は [°] であり、南半球の緯度は負である。ランベルト正角円錐図法が指定されている場合は、2つの基準緯度を指定 (<code>tlat2_lnd</code> も使用) することになる。緯度・経度座標系の場合には、設定の必要はない。
<code>tlon_lnd</code>	<i>real</i> * 4	地図投影座標系の基準経度を指定する。単位は [°] であり、西経の経度は負である。

& gridset_lnd

ここでは、土地利用データの格子間隔や基準となる格子点とその緯度・経度を設定する。*surface* に使用され、依存関係はない。

<code>xdim_lnd,</code> <code>ydim_lnd,</code>	<i>integer</i> * 4	土地利用データの <i>x, y</i> 方向の格子数を設定する。
<code>dx_lnd,</code> <code>dy_lnd</code>	<i>real</i> * 4	土地利用データの <i>x, y</i> 各方向の格子間隔を設定する。単位は、 <code>mpopt_lnd = 0</code> の場合は [°]、それ以外の場合は [m] である。
<code>ulat_lnd,</code> <code>ulon_lnd</code>	<i>real</i> * 4	土地利用データの領域のある1点の緯度・経度を指定する。単位は [°] であり、南半球の緯度と西経の経度は負である。これらと対応する外部データの領域での実数添字 (すぐ下の <code>riu_lnd, rju_lnd</code>) から、モデル領域との対応関係が得られる。
<code>riu_lnd,</code> <code>rju_lnd</code>	<i>real</i> * 4	上で設定した <code>ulat_lnd, ulon_lnd</code> の外部データの領域での実数添字を指定する。

& landuse

ここでは、プリプロセッサ *surface* の動作に関する設定をする。

<code>linopt</code>	<i>integer</i> * 4	土地利用データファイルの補間オプション。 0: あるモデル格子点の土地利用は直近のデータから得られる 1: 土地利用データをモデル格子点へ線型補間する
---------------------	--------------------	---

<code>categ</code>	<i>integer</i> * 4 土地利用データのカテゴリ数を指定する。100 カテゴリまで指定できる
<code>usland(1:100)</code>	<i>integer</i> * 4 入力データと <code>categ</code> で指定したカテゴリ数に見合う土地利用カテゴリを指定する。ただし、海面；-1、氷面；0~4、雪面；5~9 を指定しなければならない。
<code>usbeta(1:100)</code>	<i>real</i> * 4 <code>usland</code> に対応するように蒸発散係数を指定する。
<code>usalbe(1:100)</code>	<i>real</i> * 4 <code>usland</code> に対応するようにアルベドを指定する。
<code>usz0m(1:100)</code>	<i>real</i> * 4 <code>usland</code> に対応するように粗度を指定する。単位は [m] である。

& project_sst

ここでは、海水面温度データファイルの座標系を設定する。 *surface* に使用され、依存関係はない。

<code>mpopt_sst</code>	<i>integer</i> * 4 海水面温度データファイルの座標系を指定する。5種類の座標系を指定できる。 0: データは緯度・経度座標系である。 1: データは平射図法により座標系が設定されている 2: データはランベルト正角円錐図法により座標系が設定されている 3: データは正角円筒図法により座標系が設定されている 4: データは地図投影によらない座標系が設定されている
<code>nspol_sst</code>	<i>integer</i> * 4 海水面温度データが地図座標系の場合に、投影の基点が北半球にあるか南半球にあるかのフラッグ。 1: 外部データの地図座標系は北半球基点 -1: 外部データの地図座標系は南半球基点
<code>tlat1_sst,</code> <code>tlat2_sst</code>	<i>real</i> * 4 地図投影座標系の基準緯度を指定する。単位は [°] であり、南半球の緯度は負である。ランベルト正角円錐図法が指定されている場合は、2つの基準緯度を指定 (<code>tlat2_sst</code> も使用) することになる。緯度・経度座標系の場合には、設定の必要はない。
<code>tlon_sst</code>	<i>real</i> * 4 地図投影座標系の基準経度を指定する。単位は [°] であり、西経の経度は負である。

& gridset_sst

ここでは、海水面温度データの格子間隔や基準となる格子点とその緯度・経度を設定する。*surface* に使用され、依存関係はない。

<code>xdim_sst,</code>	<i>integer * 4</i>
<code>ydim_sst,</code>	海水面温度データの <i>x, y</i> 方向の格子数を設定する。
<code>dx_sst,</code>	<i>real * 4</i>
<code>dy_sst</code>	海水面温度データの <i>x, y</i> 各方向の格子間隔を設定する。単位は、 <code>mpopt_sst = 0</code> の場合は [°]、それ以外の場合は [m] である。
<code>ulat_sst,</code>	<i>real * 4</i>
<code>ulon_sst</code>	海水面温度データの領域のある 1 点の緯度・経度を指定する。単位は [°] であり、南半球の緯度と西経の経度は負である。これらと対応する外部データの領域での実数添字（すぐ下の <code>riu_sst, rju_sst</code> ）から、モデル領域との対応関係が得られる。
<code>riu_sst,</code>	<i>real * 4</i>
<code>rju_sst</code>	上で設定した <code>ulat_sst, ulon_sst</code> の外部データの領域での実数添字を指定する。

& united

ここでは、ポストプロセッサ *unite* の動作に関する設定をする。

<code>fltype</code>	<i>chracter * 80</i> 処理されるファイルの種類を指定する。80 文字のうち最初の 3 文字のみ使用する。 <code>dmp:</code> ヒストリファイルを 1 つにまとめる <code>geo:</code> 地理情報ファイルを 1 つにまとめる
<code>flsize</code>	<i>integer * 4</i> 処理される個々のファイルのバイト単位の大きさを指定する。 <code>dmpopt = 2</code> であれば、ユーザーが任意に <code>outdmp2d.f</code> や <code>outdmp3d.f</code> を挿入して出力されたファイルであっても、中身の変数の種類・数に関わらず、バイト数を指定するだけでよい。
<code>flitv</code>	<i>real * 4</i> 処理されるヒストリファイルの出力時間間隔を指定する。 <code>flitv ≠ dmpitv</code> でもよい。 <code>stime, etime</code> 間のヒストリファイルが処理される。
<code>delopt</code>	<i>integer * 4</i> 処理後、各ノード毎のファイルを消去するかどうかを指定するオプション。 0: 各ノード毎のファイルを残す 1: 各ノード毎のファイルを消す

9.2 データフローとファイルフォーマット

9.2.1 データフローの概要

CReSS のデータフローの全体像は、逐次実行版については図 9.1 のように、並列実行版については図 9.2 のようになる。ここでは、全ての場合を網羅するように示しているの、設定によっては実行する必要のないプログラムや準備する必要のないファイルもある。

なお、図 9.1 ならびに図 9.2 に現れるファイル名の斜体の意味は次表のとおりである。

<i>exprim</i>	実験名。ユーザーが指定するもので、ファイル名の最初の部分はそれに置き換えられる。
<i>xxxxxx</i>	秒単位の予報開始時刻からの経過時間。例えば、経過時間 1200.0 s なら <i>xxxxxx</i> = 001200。
<i>yyyy</i>	ノード番号。並列実行版では、各ノードからそれぞれファイルが入出力される。

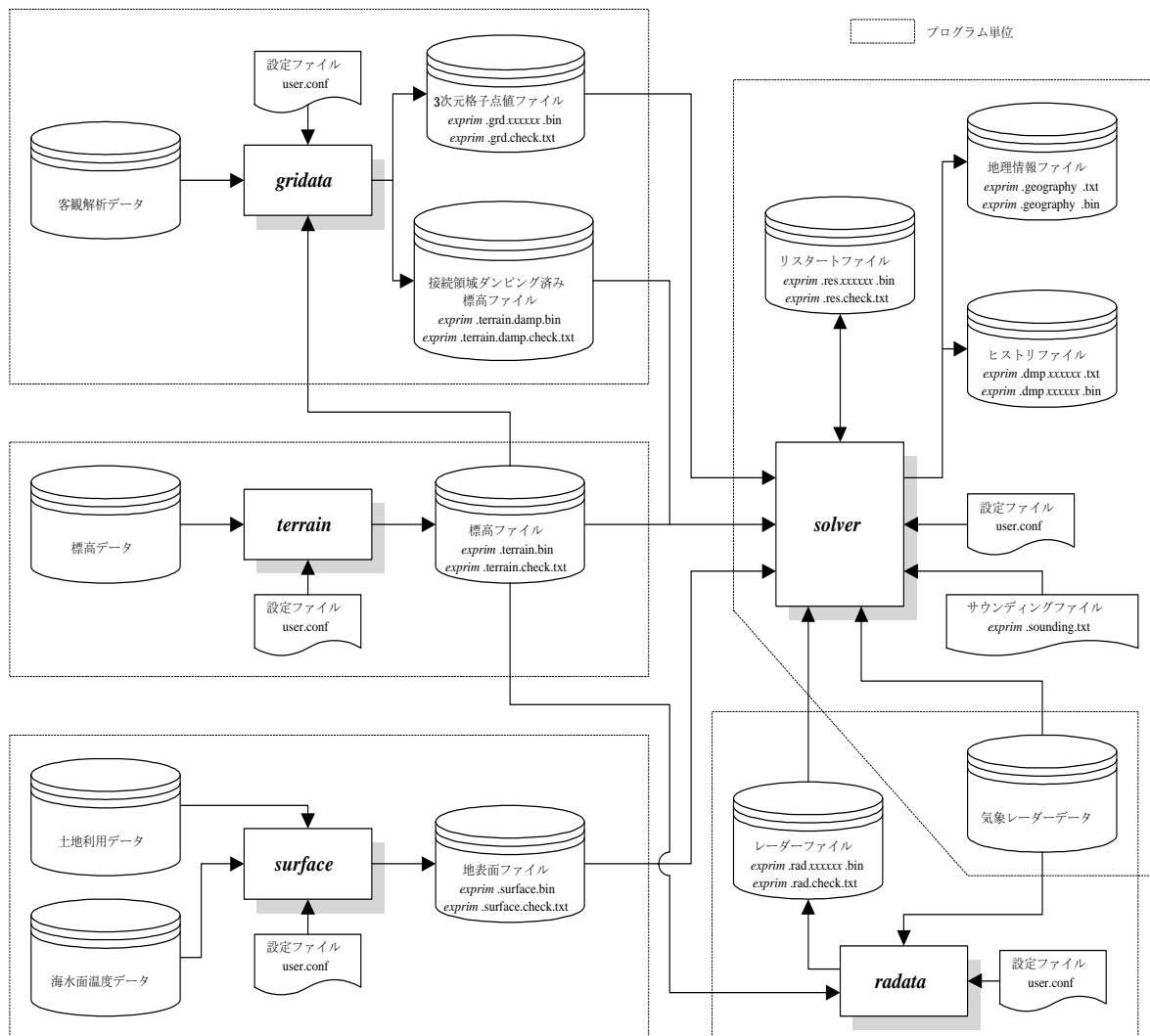


図 9.1. CReSS 逐次実行版のデータフローの全体図。

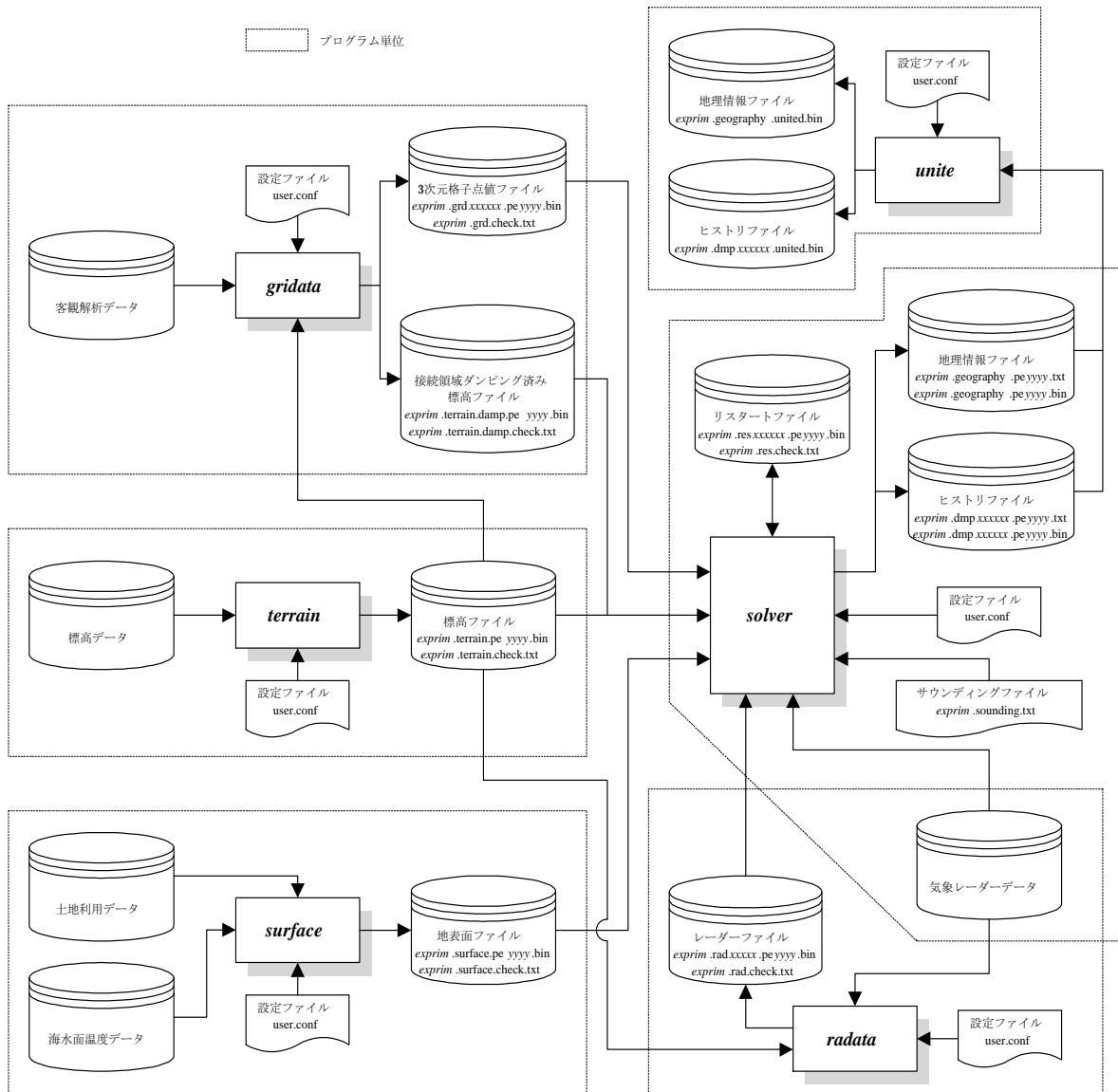


図 9.2. CReSS 並列実行版のデータフローの全体図。

9.2.2 入出力ファイルの概要

ここでは、図 9.1 ならびに図 9.2 に現れる各入出力ファイルの用途を説明する。なお、逐次実行版のそれらは、以下のファイル名からノード番号 `.peyyyy` を除いたものである。

ファイル名	説明
<code>exprim.terrain.peyyyy.bin</code>	モデルの地面格子点の標高値が格納されているファイル。直接探査の書式なしバイナリファイル。
<code>exprim.terrain.check.txt</code>	上記ファイル入力時に、設定の是非を判定するための検査ファイル。並びのテキスト書式のファイル。

<code>exprim.terrain.damp.peyyyy.bin</code>	接続領域が滑らかに繋がるように補正された標高値が格納されているファイル。直接探査の書式なしバイナリファイル。
<code>exprim.terrain.damp.check.txt</code>	上記ファイル入力時に、設定の是非を判定するための検査ファイル。並びのテキスト書式のファイル。
<code>exprim.surface.peyyyy.bin</code>	モデルの地面格子点の土地利用カテゴリや海水面温度が格納されているファイル。直接探査の書式なしバイナリファイル。
<code>exprim.surface.check.txt</code>	上記ファイル入力時に、設定の是非を判定するための検査ファイル。並びのテキスト書式のファイル。
<code>exprim.grdxxxxxx.peyyyy.bin</code>	補間済みの外部の客観解析データの各格子点値が格納されているファイル。直接探査の書式なしバイナリファイル。
<code>exprim.grd.check.txt</code>	上記ファイル入力時に、設定の是非を判定するための検査ファイル。並びのテキスト書式のファイル。
<code>exprim.radxxxxxx.peyyyy.bin</code>	補間済みの外部のレーダーデータの各格子点値が格納されているファイル。直接探査の書式なしバイナリファイル。
<code>exprim.rad.check.txt</code>	上記ファイル入力時に、設定の是非を判定するための検査ファイル。並びのテキスト書式のファイル。
<code>exprim.sounding.txt</code>	水平面一様の鉛直1次元の初期値ファイル。並びのテキスト書式のファイル。
<code>exprim.dmpxxxxxx.peyyyy.txt</code> <code>exprim.dmpxxxxxx.peyyyy.bin</code>	計算結果が格納されたファイル。設定オプションにより、並びのテキスト書式のファイルで出力するか、直接探査の書式なしバイナリファイルで出力するかを選択できる。
<code>exprim.geography.peyyyy.txt</code> <code>exprim.geography.peyyyy.bin</code>	計算領域の地理情報が格納されたファイル。上記ファイルと同様に、設定オプションにより、出力形式を指定できる。
<code>exprim.resxxxxxx.peyyyy.bin</code>	リスタートファイル。順番探査の書式なしバイナリファイル。
<code>exprim.res.check.txt</code>	上記ファイル入力時に、設定の是非を判定するための検査ファイル。並びのテキスト書式のファイル。
<code>exprim.dmpxxxxxx.united.bin</code>	各ノード毎に別々のヒストリファイルが1つにまとめられたもの。直接探査の書式なしバイナリファイル。
<code>exprim.geography.united.bin</code>	各ノード毎に別々の地理情報ファイルが1つにまとめられたもの。直接探査の書式なしバイナリファイル。

9.2.3 ユーザー操作のファイルフォーマット

本節では、ユーザーが直接操作する必要が生ずる、サウンディングファイル、履歴ファイル、地理情報ファイルのフォーマットを説明する。

サウンディングファイルのフォーマット

サウンディングファイルの第1列目と第2列目、さらに、第5列目には変数の種類が設定できるように設計されている。そこで、設定することのできる変数を全て列挙しておく。このとき、組み合わせの制限はないので、全部で8通りの選択方法があることになる。

第1列	高さ [m] もしくは、圧力 [Pa]
第2列	温度 [K] もしくは、温位 [K]
第3列	速度の東西成分 [m s ⁻¹]
第4列	速度の南北成分 [m s ⁻¹]
第5列	水蒸気混合比 [kg kg ⁻¹] もしくは、相対湿度 [%]

次に、例として節 9.3.1 で用いたファイルの一部分を示す。これを模倣してサウンディングファイルとして作成し、ファイル `user.conf` を設定するだけである。Form ディレクトリにも様々なサンプルがあるので、参考にするとよい。

```
#####
#
#   One dimensional sounding input file, sounding.txt.cats.eye.form
#
#   This is the cats eye simulation data.
#
#   Author       : SAKAKIBARA Atsushi
#   Date        : 1999/07/23
#   Modification : 1999/07/28
#               : 1999/11/19
#
#   First column: height [m]
#   Second column: tempereture [K]
#   Third column: x components of velocity [m/s]
#   Fourth column: y components of velocity [m/s]
#   Fifth column: water vapor relative humidity [%]
#
#####
780.e0 300.200e0 8.0000e0 0.e0 0.e0
720.e0 300.200e0 7.9999e0 0.e0 0.e0
660.e0 300.200e0 7.9993e0 0.e0 0.e0
```

```

600.e0 300.200e0 7.9946e0 0.e0 0.e0
540.e0 300.199e0 7.9449e0 0.e0 0.e0
530.e0 300.199e0 7.9231e0 0.e0 0.e0
      :
      :
200.e0 299.802e0 -7.9231e0 0.e0 0.e0
190.e0 299.802e0 -7.9449e0 0.e0 0.e0
180.e0 299.801e0 -7.9604e0 0.e0 0.e0
120.e0 299.800e0 -7.9946e0 0.e0 0.e0
 60.e0 299.800e0 -7.9993e0 0.e0 0.e0
  0.e0 299.800e0 -7.9999e0 0.e0 0.e0

```

ヒストリファイルと地理情報ファイルのフォーマット

計算結果が格納されているヒストリファイルと計算領域の地理情報が格納されている地理情報ファイルは、ユーザーの設定により、並びのテキスト書式のファイルによる出力か直接探査の書式なしバイナリファイルによる出力かが選択できる。

まず、出力形式をテキスト書式 (`dmpfmt = 1`) に設定した場合には、ある 1 変数は、

```

do xxx k=2,nk-2
  write(ionum,*,err=errnum) (variable(i,j,k),i=2,nx-2,j=2,ny-2)
xxx continue

```

のように出力される。

次に、出力形式を書式なしバイナリ (`dmpfmt = 2`) に設定した場合には、ある 1 変数は、

```

do xxx k=2,nk-2
  recnum=recnum+1
  write(ionum,rec=recnum,err=errnum)
    . (variable(i,j,k),i=2,nx-2,j=2,ny-2)
xxx continue

```

のように直接探査の書式なしバイナリファイルで出力される (`outdmp3d.f` と `outdmp2d.f` を参照)。ここで、`nx,ny,nz` は全格子点数を表し、地理情報ファイルの場合は鉛直方向の次元は 1 である。

各変数の出力範囲は、計算格子と同じ z^* 系 (ζ 系) への出力オプション (`dmpcmp = 1`) について言えば、次の図 9.3 (図 6.2 と同じ) のように、太線内の領域の \times の点の値が出力される。ただし、●の点に定義される各方向の速度成分については、スカラー量が定義されている \times の点に補間されて出力される。

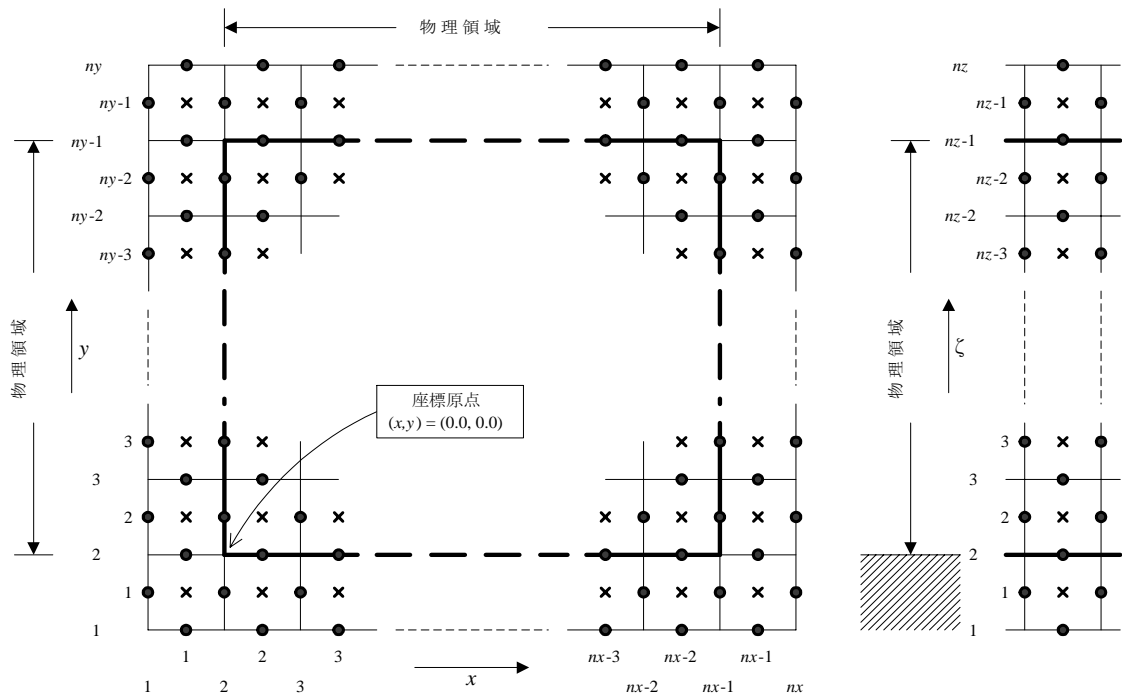


図 9.3. 全計算領域における変数の配置図 (●は速度の定義点、×はスカラー量の定義点を表す)。

ヒストリファイルについて、他の鉛直座標系への出力オプションを選択した場合の z^* 系 (ζ 系) との関連は、図 9.4 のとおりである。

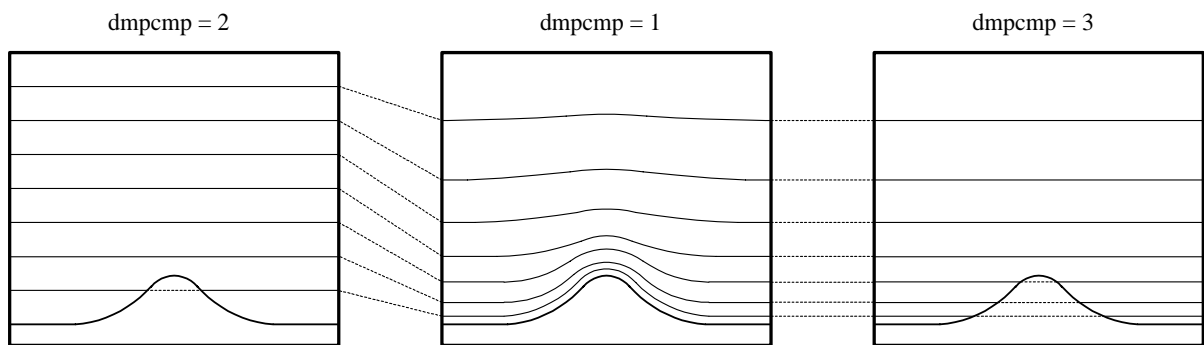


図 9.4. ヒストリファイルの鉛直座標系出力オプションの関連。

出力される変数の種類は実行時の設定により毎回異なるが、ファイル出力の設定時間に到達するたびに、出力各面の高度 [m] などの必要な情報とともにその変数種類や出力順が標準出力にログとして残される。次表に出力される変数とその単位を示す。

地理情報ファイルの出力変数 (プログラムにおける変数名と出力範囲)

ht(2:nx-2,2:ny-2)	標高	m
lat(2:nx-2,2:ny-2)	緯度	°
lon(2:nx-2,2:ny-2)	経度	°
mf(2:nx-2,2:ny-2,1:1)	マップスケールファクタ	
fc(2:nx-2,2:ny-2,1:1)	コリオリパラメータ	
fc(2:nx-2,2:ny-2,2:2)	コリオリパラメータ	
land(2:nx-2,2:ny-2)	土地利用カテゴリ	

ヒストリファイルの出力変数 (プログラムにおける変数名と出力範囲)

usfc(2:nx-2,2:ny-2)	地上 10 m における速度の東西成分	m s ⁻¹
vsfc(2:nx-2,2:ny-2)	地上 10 m における速度の南北成分	m s ⁻¹
psfc(2:nx-2,2:ny-2)	地上 1.5 m における圧力	Pa
ptsfc(2:nx-2,2:ny-2)	地上 1.5 m における温位	K
qvsfc(2:nx-2,2:ny-2)	地上 1.5 m における水蒸気混合比	kg kg ⁻¹
u(2:nx-2,2:ny-2,2:nz-2)	速度の東西成分	m s ⁻¹
v(2:nx-2,2:ny-2,2:nz-2)	速度の南北成分	m s ⁻¹
w(2:nx-2,2:ny-2,2:nz-2)	速度の鉛直成分	m s ⁻¹
pbr(2:nx-2,2:ny-2,2:nz-2)	基準状態の圧力	※予報開始時刻から不変 Pa
pp(2:nx-2,2:ny-2,2:nz-2)	基準状態からの圧力の偏差	Pa
ptbr(2:nx-2,2:ny-2,2:nz-2)	基準状態の温位	※予報開始時刻から不変 K
ptp(2:nx-2,2:ny-2,2:nz-2)	基準状態からの温位の偏差	K
qvbr(2:nx-2,2:ny-2,2:nz-2)	基準状態の水蒸気混合比	※予報開始時刻から不変 kg kg ⁻¹
qv(2:nx-2,2:ny-2,2:nz-2)	水蒸気混合比	kg kg ⁻¹
qc(2:nx-2,2:ny-2,2:nz-2)	雲水混合比	kg kg ⁻¹
qr(2:nx-2,2:ny-2,2:nz-2)	雨水混合比	kg kg ⁻¹
qi(2:nx-2,2:ny-2,2:nz-2)	雲氷混合比	kg kg ⁻¹
qs(2:nx-2,2:ny-2,2:nz-2)	雪の混合比	kg kg ⁻¹
qg(2:nx-2,2:ny-2,2:nz-2)	霰の混合比	kg kg ⁻¹
nci(2:nx-2,2:ny-2,2:nz-2)	雲氷の数密度	m ⁻³
ncs(2:nx-2,2:ny-2,2:nz-2)	雪の数密度	m ⁻³
ncg(2:nx-2,2:ny-2,2:nz-2)	霰の数密度	m ⁻³
prr(2:nx-2,2:ny-2,1:1)	降雨量強度	m s ⁻¹
prr(2:nx-2,2:ny-2,2:2)	ファイル出力間隔の時間内の積算降雨量	m
prs(2:nx-2,2:ny-2,1:1)	降雪量強度	m s ⁻¹
prs(2:nx-2,2:ny-2,2:2)	ファイル出力間隔の時間内の積算降雪量	m
prg(2:nx-2,2:ny-2,1:1)	降霰量強度	m s ⁻¹
prg(2:nx-2,2:ny-2,2:2)	ファイル出力間隔の時間内の積算降霰量	m
zph(2:nx-2,2:ny-2,2:nz-2)	各格子点の高さ	※予報開始時刻から不変 m

9.3 CReSS の実行

9.3.1 solver の実行方法

ここでは、ケルビンヘルムホルツ不安定波という具体的な実験例を用いて、主プログラム *solver* を実行する一連の作業を説明する。なお、ここでの説明は具体例を用いたものであるが、どのような場合でも動作方法は変わらない。

さて、ここで、ケルビンヘルムホルツ不安定波を選んだのは、

- 入力ファイルに水平面一様のサウンディングファイルを用いればよい。
- 計算時間が比較的短時間ですみ、結果をすぐに確認することができる。
- その結果を、教科書などの図と比較しやすい。

からである。本節をそのまま実行すれば、結果を出力するまでの一連の動作を経験することができる。

まず、ユーザーは適当なディレクトリに **CReSS** の圧縮アーカイブファイルがあることを確認し、このファイルを解凍・展開しなければならない。

```
% ls
cress1.4m.tar.Z
% uncompress -c cress1.4m.tar.Z | tar xvf -
  messages
    :
% ls
CReSS1.4m      cress1.4m.tar.Z
%
```

これで **CReSS** は適当な場所にインストールされたことになるので、次に、**CReSS1.4m** ディレクトリに移動し、どのような構造になっているか見てみよう。

```
% cd CReSS1.4m
% ls
Doc          Src          compile.conf
Form        Tmp          compile.csh
%
```

上に示したような表示は得られているだろうか。大文字で始まるものはサブディレクトリ、小文字で始まるものはファイルである。以下にこれらを簡単に説明しよう。

Doc	ユーザーズガイドや Readme などの各種文書があるディレクトリ
Form	各種検証実験の設定ファイルとサウンディングデータファイルがあるディレクトリ
Src	全ソースファイルがあるディレクトリ
Tmp	コンパイル時にインクルードファイルが生成されるテンポラリディレクトリ
compile.conf	コンパイルコマンドやオプションを指定する設定ファイル
compile.csh	コンパイルを制御するシェルスクリプト

さて、Form ディレクトリには上記のように、各種実験の設定ファイルとデータファイルがある。それらを CReSS1.4m ディレクトリに以下のようなファイル名でコピーし（この実験名が `test` であるため、入出力されるファイル名には全て `test.` が付加される）、ケルビンヘルムホルツ不安定波の実験を始める。

```
% cp Form/user.conf.cats.eye.form user.conf
% cp Form/sounding.txt.cats.eye.form test.sounding.txt
% ls
Doc                Src                compile.conf      test.sounding.txt
Form               Tmp               compile.csh       user.conf
%
```

次に、実行ファイル作成のために、ソースコードを以下のようにしてコンパイルする。このとき配列の大きさなどの情報は `user.conf` から読み込まれ、`compile.csh` が自動的にインクルードファイルを生成するので、ソースファイルの一部を書き換えたりする必要はない。しかし、デフォルトの設定では出力ファイルに対して直接探査の書式なしバイナリの指定をしているので、使用している機種によっては、その語長に特殊な値を設定しなければならない可能性がある。その場合、`user.conf` の変数 `wlength` を書き直す（多くの場合 4 1 と思われる）必要がある。また、Fortran コンパイラコマンドがよく用いられるような `f90` や `mpif90` でない場合やコンパイラコマンドオプションを設定したい場合、`compile.conf` に必要な情報を書き加える必要がある。

```
% compile.csh solver user.conf
cd Src; messages
      :
%
```

コンパイルが終了すると、実行ファイル `solver.exe` が生成されている。実際には、`solver.exe` は `Src` ディレクトリ内の実体へのシンボリックリンクである。

```
% ls
Doc                Tmp                solver.exe
Form               compile.conf       test.sounding.txt
Src                compile.csh        user.conf
%
```

これで実験の準備は全て整った。並列計算機を用いて実験する場合、ジョブスクリプトを記述したり機種に適合した MPI コマンドを実行しなければならないかもしれないが、基本的には以下のコマンドを入力することにより、設定ファイル `user.conf` を標準入力から読み込み、計算の進捗状況を `log.solver`（ファイル名はもちろん可変）に標準出力から書き出しながら、バックグラウンドで `solver.exe` を実行すればよい。

```
% solver.exe < user.conf >& log.solver &
%
```

計算終了後、次のように各ノード毎のヒストリファイル、地理情報ファイル、リスタートファイルが生成される（この例では 4 ノードの実行なので 4 つずつ）。計算が正常に終了したかどうかは、標準出力ファイル `log.solver` の最後にある一文 “This program stopped normally.” が記述されているかどうかで確認できる。また、このファイルには計算の進捗状況やファイルの入出力のメッセージ、各予報変数の最大・最

小値などが出力されているので確認するとよい。

```
% ls
Doc                test.dmp000120.pe0002.bin
Form               test.dmp000120.pe0003.bin
Src                test.dmp000160.pe0000.bin
Tmp                test.dmp000160.pe0001.bin
compile.conf      test.dmp000160.pe0002.bin
compile.csh       test.dmp000160.pe0003.bin
log.solver        test.dmp000200.pe0000.bin
solver.exe        test.dmp000200.pe0001.bin
test.dmp000000.pe0000.bin test.dmp000200.pe0002.bin
test.dmp000000.pe0001.bin test.dmp000200.pe0003.bin
test.dmp000000.pe0002.bin test.dmp000240.pe0000.bin
test.dmp000000.pe0003.bin test.dmp000240.pe0001.bin
test.dmp000040.pe0000.bin test.dmp000240.pe0002.bin
test.dmp000040.pe0001.bin test.dmp000240.pe0003.bin
test.dmp000040.pe0002.bin test.geography.bin
test.dmp000040.pe0003.bin test.res000240.check.txt
test.dmp000080.pe0000.bin test.res000240.pe0000.bin
test.dmp000080.pe0001.bin test.res000240.pe0000.bin
test.dmp000080.pe0002.bin user.res000240.pe0001.bin
test.dmp000080.pe0003.bin test.res000240.pe0002.bin
test.dmp000120.pe0000.bin test.sounding.txt
test.dmp000120.pe0001.bin user.conf
%
```

9.3.2 unite の実行方法

ヒストリファイルは、並列実行の場合、各ノード毎に複数出力されるので1つにまとめる必要がある。そのためにポストプロセッサ *unite* を実行しなければならない。次のように入力しよう。

```
% compile.csh unite user.conf
cd Src; messages
      :
%
```

これで、ポストプロセッサ *unite.exe* が生成された。*solver.exe* と同様にして、適当な標準出力ファイルを指定してバググラウンドで実行する。

```
% unite.exe < user.conf >& log.unite &
%
```

まとめられたファイルは、例えば 240s 後のヒストリファイルなら *test.dmp000240.united.bin* である。また、このファイルのフォーマットは前節 9.2.3 で説明したとおりで、出力変数の種類・順番は、*solver.exe* 実行時の標準出力ファイル *log.solver* に出力されているので確認するとよい。このファイルを表示アプリ

ケーション等に読み込ませれば、図 9.5 と図 9.6 (図 10.6、図 10.7 と同じ) のように結果が表示できる。

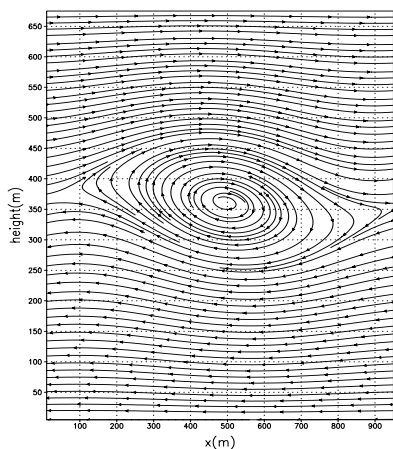


図 9.5. 予報開始時刻より 240s 後の流線 ($u - w$)。

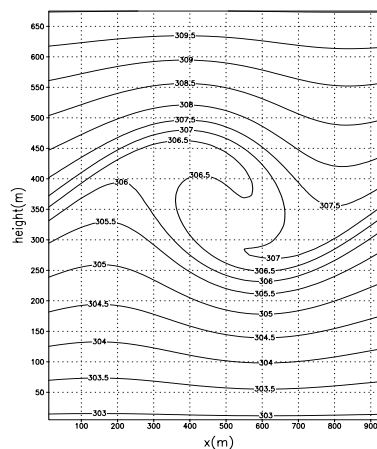


図 9.6. 予報開始時刻より 240s 後の温度 [K]。

9.3.3 terrain の実行方法

プリプロセッサ *terrain* は、標高データをモデルの計算領域に補間するプログラムである。客観解析データを用いて予報実験をするような場合は、平原における計算でもない限り標高データが必要になるので、まず最初に *terrain* を実行しなければならないだろう。

さて、*terrain* を実行する場合、まず、外部の標高データファイルを用意しプログラムに読み込ませる必要があるが、**CReSS** ではデータファイルの形式は特に指定していない。その代わりに、**Src** ディレクトリにあるサブルーチン *rdhigh.f* の#で囲まれた部分を、データファイルに合わせて変更しなければならない。

```
* ##### You will have to modify the following lines. #####
```

```
    siz=nid*njd*wlength
```

```
    write(trnfl(1:16),'(a16)' 'data.terrain.bin')
```

```
    open(iotrnl,iostat=stat,file=trnfl(1:16),status='old',
```

```
    .    access='direct',form='unformatted',recl=siz)
```

```
* #####
```

```
* ##### You will have to modify the following lines. #####
```

```
    read(iotrnl,rec=1,err=100) ((htdat(id,jd),id=1,nid),jd=1,njd)
```

```
* #####
```

ここで、入力される標高データの単位は [m] である。入力データの単位が異なる場合には、このサブルーチンでの読み込みの際に直しておく必要がある。

実行方法は、まず、9.1 節を参照しながら設定ファイルを記述し、次のようにコンパイルする。

```
% compile.csh terrain user.conf
cd Src; messages
      :
%
```

コンパイルが終了すると、実行ファイル `terrain.exe` が生成されている。実際には、`terrain.exe` は `Src` ディレクトリ内の実体へのシンボリックリンクである。

次に、以下のコマンドを入力して `terrain.exe` を実行する。入力ファイルは `terrain.exe` と同じディレクトリになければならない。

```
% terrain.exe < user.conf >& log.terrain &
%
```

終了後、各ノード毎の設定に見合う名称（ここでも実験名は `test` としている）の標高ファイルが生成される。なお `terrain.exe` は逐次版・並列版ともに 1 ノードで動作し、逐次版の場合には 1 つのファイルが、並列版の場合には `solver.exe` を実行するときに必要なノード数のファイルが出力される。

9.3.4 *surface* の実行方法

プリプロセッサ *surface* は、土地利用データや海水面データをモデルの計算領域に補間するプログラムである。土地利用状況や海水面温度の分布を反映させ、より現実に近い条件で予報実験をする場合には、*surface* を実行しなければならない。

さて、*surface* を実行する場合、まず、外部の土地利用データファイルならびに海水面温度データファイルを用意しプログラムに読み込ませる必要があるが、*surface* においてもデータファイルの形式は特に指定していない。その代わりに、`Src` ディレクトリにあるサブルーチン `rdland.f` ならびに `rdsst.f` の # で囲まれた部分を、データファイルに合わせて変更しなければならない。なお、必ずしも土地利用・海水面の 2 種類のファイルがなければ実行できないことはなく、どちらか一方のデータのみを用いることも可能である。

```
* ##### You will have to modify the following lines. #####
```

```
      siz=nid*njd*wlngth
```

```
      write(lndfl(1:13),'(a13)') 'data.land.bin'
```

```
      open(iolnd,iostat=stat,file=lndfl(1:13),status='old',
.        access='direct',form='unformatted',recl=siz)
```

```
* #####
```

```
* ##### You will have to modify the following lines. #####
```

```
read(iolnd,rec=1,err=100) ((landat(id,jd),id=1,nid),jd=1,njd)
```

```
* #####
```

```
* ##### You will have to modify the following lines. #####
```

```
siz=nid*njd*wlength
```

```
write(sstfl(1:12),'(a12)') 'data.sst.bin'
```

```
open(iosst,iostat=stat,file=sstfl(1:12),status='old',
     .   access='direct',form='unformatted',recl=siz)
```

```
* #####
```

```
* ##### You will have to modify the following lines. #####
```

```
read(iosst,rec=1,err=100) ((sstdat(id,jd),id=1,nid),jd=1,njd)
```

```
* #####
```

ここで、入力される土地利用データのカテゴリは `rdland.f` の最初の部分にある `#` で囲まれた内容に従わねばならない。また、海水面温度の単位は [K] である。入力データのカテゴリや単位が異なる場合には、このサブルーチンでの読み込みの際に直しておく必要がある。

実行方法は、`terrain.exe` と同様に、設定ファイルを記述し、次のようにコンパイルする。

```
% compile.csh surface user.conf
cd Src; messages
      :
%
```

コンパイルが終了すると、実行ファイル `surface.exe` が生成されている。

次に、以下のコマンドを入力して `surface.exe` を実行する。入力ファイルは `surface.exe` と同じディレクトリになければならない。

```
% surface.exe < user.conf >& log.surface &
%
```

終了後、各ノード毎の設定に見合う名称（ここでも実験名は `test` としている）の地表面ファイルが生成される。なお `surface.exe` も逐次版・並列版ともに 1 ノードで動作する。

9.3.5 *gridata* の実行方法

次に、3次元格子点値ファイルを生成するプリプロセッサ *gridata* に入出力されるファイルと実行方法を説明する。

さて、モデル入力用 3次元格子点値ファイルを用意する場合、まず、客観解析値などの 3次元データファイルを用意する必要がある。他のプリプロセッサと同様に、そのファイルの形式は特に指定していないが、その代わりに、`Src` ディレクトリにあるサブルーチン `rdobj.f` を変更しなければならない。`rdobj.f` の#で囲まれた部分を読み込むべきファイルに合わせて変更すればよい。ただし、時間的に変化しない標高データファイルなどと違い、ファイル名の一部にグレゴリオ暦の形式 `yyyymmddhhmm` (UTC) が含まれてなければならない。

```
* ##### You will have to modify the following lines. #####

      write(objfl(1:24),'(a8,a12,a4)') 'data.obj',cdate(1:12),'.bin'

      open(iobj,iostat=stat,file=objfl(1:24),status='old',
        .      access='sequential',form='unformatted')

* #####

* ##### You will have to modify the following lines. #####

      read(iobj,end=100,err=100) zdat

      read(iobj,end=100,err=100) udat
      read(iobj,end=100,err=100) vdat

      read(iobj,end=100,err=100) pdat
      :

* #####
```

ここで、必ず入力されなければならない変数は、次の表、

<code>zdat</code>	各格子点の高さ（標高と一致するとは限らない）	m
<code>udat</code>	データ格子系の速度の東西成分（緯度・経度方向とは異なる）	$m s^{-1}$
<code>vdat</code>	データ格子系の速度の南北成分（緯度・経度方向とは異なる）	$m s^{-1}$
<code>pdat</code>	各格子点の圧力	Pa
<code>ptdat</code>	各格子点の温位または温度	K

入力の有無を選択できる変数は、次の表のとおりである。

wdat	データ格子系の速度の鉛直成分	m s^{-1}
qvdat	各格子点の水蒸気混合比または相対湿度	kg kg^{-1} or %
qcdat	各格子点の雲水混合比	kg kg^{-1}
qrdat	各格子点の雨水混合比	kg kg^{-1}
qidat	各格子点の雲氷混合比	kg kg^{-1}
qsdat	各格子点の雪混合比	kg kg^{-1}
qgdat	各格子点の霰混合比	kg kg^{-1}

なお、入力データの単位が上の表と異なる場合には、このサブルーチンにおける読み込みの際に変換しておく必要がある。

また、地表面のデータが別途ある場合 (`refsfc = 1`) には、一番目の鉛直方向の添え字 (`kd = 1`) の番地にそれらのデータを読み込まなければならない。一番目の鉛直方向の添え字が使用されることは、`refsfc` の設定によって違いがあるわけではないので、読み込みの際には注意が必要である。例えば、一番目の添え字の番地に、各格子点の高さ `zdat` にデータの標高を、物理量についてはその高さに対応するデータを読み込んでおき、`refsfc = 1` と設定すれば、`kd = 1` に対応するデータの標高より低い物理量を無視して鉛直補間がなされる。そうでなければ、`kd = 1` の物理量だけが区別されることにならないので、モデル格子では地中であるかどうかにかかわらず内・外挿されることになる。特に p 面のデータの場合には、何らかの補正はされているにしても、気圧が高い面ほど地中に位置するデータが含まれることに注意が必要である。

実行方法は、ここでも基本的に他のプリプロセッサと同様である。まず、設定ファイルを記述し、次のようにコンパイルする。

```
% compile.csh gridata user.conf
cd Src; messages
:
%
```

コンパイルが終了すると、実行ファイル `gridata.exe` が生成されている。

次に、以下のコマンドを入力して `gridata.exe` を実行する。入力ファイルは `gridata.exe` と同じディレクトリになければならない。

```
% gridata.exe < user.conf >& log.gridata &
%
```

終了後、各ノード毎の実験名の設定に見合う名称 (実験名を `test` としているのは前節までと同様) のモデル入力用 3次元格子点値ファイルと、設定によっては、外部気象データの標高に徐々に繋がるような標高ファイルが生成される。なお、`gridata.exe` も他のプリプロセッサと同様に逐次版・並列版ともに 1 ノードで動作する。

9.3.6 *radata* の実行方法

最後に、モデル入力用レーダーファイルを作成するプリプロセッサ *radata* に入出力されるファイルと実行方法を説明する。

さて、レーダーファイルを用意する場合、まず、ドップラーレーダーデータなどの3次元データファイルを用意する必要がある。他のプリプロセッサと同様に、そのファイルの形式は特に指定していないが、Src ディレクトリにあるサブルーチン *rdradar.f* を変更しなければならない。*rdradar.f* の#で囲まれた部分を読み込むべきファイルに合わせて変更すればよい。ただし、ファイル名の一部にグレゴリオ暦の形式 *yyyymmddhhmmss* (UTC) が含まれてなければならない (*gridata* の場合よりさらに秒 *ss* が2桁増えることに注意)。

```

* ##### You will have to modify the following lines. #####

      write(objfl(1:28),'(a10,a12,i2.2,a4)')
      .
      .      'data.radar',cdate(1:12),csec,'.bin'

      open(iobj,iostat=stat,file=objfl(1:28),status='old',
      .      access='sequential',form='unformatted')

* #####

* ##### You will have to modify the following lines. #####

      read(iobj,end=100,err=100) zdat

      if(radvar(1:1).eq.'o') then

          read(iobj,end=100,err=100) udat

      end if

      if(radvar(2:2).eq.'o') then
          :
          :
      end if

* #####

```

ここで、入力される変数は次の表のとおりである。

<i>zdat</i>	各格子点の高さ (標高と一致するとは限らない)	m
<i>udat</i>	データ格子系の速度の東西成分 (緯度・経度方向とは異なる)	m s ⁻¹
<i>vdat</i>	データ格子系の速度の南北成分 (緯度・経度方向とは異なる)	m s ⁻¹
<i>wdat</i>	データ格子系の速度の南北成分 (緯度・経度方向とは異なる)	m s ⁻¹
<i>qpd</i>	各格子点の降水物質の反射強度または混合比	dBZe or kg kg ⁻¹

なお、入力データの単位が上の表と異なる場合には、このサブルーチンにおける読み込みの際に変換しておく必要がある。また、値が定義されていない点 (undefined point) は、 -1.0×10^{34} 未満の値で満たされていなければならないように設計されているので、同じく読み込みの際に変換しておく必要がある。

実行方法は、ここでも基本的に他のプリプロセッサと同様である。まず、設定ファイルを記述し、次のようにコンパイルする。

```
% compile.csh radata user.conf
cd Src; messages
      :
%
```

コンパイルが終了すると、実行ファイル `radata.exe` が生成されている。

次に、以下のコマンドを入力して `radata.exe` を実行する。入力ファイルは `radata.exe` と同じディレクトリになければならない。

```
% radata.exe < user.conf >& log.radata &
%
```

終了後、各ノード毎の実験名の設定に見合う名称 (実験名を `test` としているのは前節までと同様) のモデル入力用レーダー格子点値ファイルが生成される。なお、`radata.exe` も他のプリプロセッサと同様に逐次版・並列版ともに 1 ノードで動作する。