

NICAM チュートリアル（2019 VL 講習会資料）

2019/08/23 再改訂 伊藤純至

2018/03/23 再改訂 高須賀大輔

2016/07/25 再改訂 大野知紀

2015/10/01 再改訂 宮川知己

2015/04/06 改訂 宮川知己

2012 VL 講習会資料（2012/02/23 三浦裕亮）より

1. NICAM について

NICAM とは、Nonhydrostatic ICosahedral Atmosphere Model という大気モデルの略称です。皆さんは、2002 年に稼働した当時世界一の計算速度を誇った地球シミュレータというスーパーコンピューターを知っていますか？もう 15 年以上も昔の話なので、若い人は知らないかもしれないですね。ともかく、NICAM はその地球シミュレータという新しい並列計算機を念頭に、1999 年位から開発が始まりました。大規模並列計算機では従来型のスペクトル法モデルは計算負荷が大きくなるので格子法モデルに切り替えよう、というのが開発の主たる動機です（このような面倒な話は分からなくても大丈夫です）。

NICAM では全球をほぼ一様な面積の 6 角形/5 角形で覆うことのできる正 20 面体格子という格子形を採用しています。そのような若干扱いにくい格子系を採用しているのは、計算効率を高めるためであったり、物理過程のパラメタの単一性を保つためであったりしますが、科学的な理由だけでなく、その格子構造が見た目にも美しく興味深いものであるという（主観的）事実も、何となく他の格子系の採用を躊躇ってしまう由縁であるかもしれません。

このような蘊蓄（うんちく）めいた話はともかく、NICAM を体験してみる、というのがここでの目的です。“体験してみる”の意味は、1. プログラムを計算機上で動かし、2. 出力されたデータを気象学的に見てみる、ということです。以下、色々と作業が必要になるため、作業の途中で自分が“何のため”に“何を”行っているか、が不明瞭になってしまう危険があります。数値モデルを使う場合、どのようなモデルであっても多かれ少なかれ“使いにくいなあ”という感じは抱くものですが（人間の頭は特異な世界に馴染むのに時間が必要なのかもしれません）、折角の機会ですので“NICAM は簡単に使えた”という体験をしていただけのように進めたいと思います。

2. 準備

実際の作業に移る前にその進め方について整理しておきましょう。作業を進める途中で“今何をやっているのか”分からなくなったらここに戻って参照してみてください。何が終わって何が終わっていないかを確認しながら進めば、道に迷った感じも軽減されるのではないのでしょうか。

1. OFP (大型計算機) にログインする
2. モデルのファイルをコピーし、作業ディレクトリに展開する
3. モデルを `make` し、バイナリ (実行形式) を作成する
4. `mnginfo` ファイル (領域分布記述ファイル) を作る
5. 正 20 面体格子データを作る
6. 正 20 面体格子→緯度経度格子 (解析用) の内挿重みの計算
7. 鉛直格子を作る
8. 地形・植生データを作る
9. オゾンデータを作る
10. 初期条件データ (大気・海洋・陸面) を作る
11. 海面温度の参照データを作る
12. 大気の基準成層データを作る
13. NICAM を動かす
14. 出力データを緯度経度変換する

結構大変そうでしょうか？でも、今日 NICAM を動かして、その結果で論文を書こう、なんて人はいないですよ？それなら心配ありません。今日の目的は、ただただ流れを追って、モデルに慣れることです。必要なコマンドはすべて明記しながら進めます。

ところで、具体的な作業に入る前に約束事が一つとお詫びが一つあります。まず、\$マークが出てきたら、Linux のターミナル上のコマンドを表していると理解してください。これが約束です。Linux でファイルを編集するためのプログラム(`emacs` や `vi`)や大型計算機へのログイン方法については、もし分からないときは遠慮無く聞いてください。

それでは、具体的な作業に入りましょう。まず、今日の演習で使用する大型計算機 OFP (正式名称は Oakforest-PACS) にログインします。コマンドは、

```
1. $ ssh -X (ユーザー名)@ofp.jcahpc.jp (Enter←以降は省略します)
```

です。ターミナルのコマンドの最後にリターンキー (Enter キー) を押してコマンドの入力を完了するようにしてください。ログインに成功すると `/home/(アカウント)` という場所に入ります。OFP は普通の大型計算機と違う点があります。ログイン先のホームディレクトリはファイルを置く専用となっていて、計算はできません。以降の作業は全て `/work/ (グループ名) /(アカウント)` の各自の作業ディレクトリで行います。

まず OFP の作業ディレクトリへ移動します。

```
2. $ cd /work/gt32/(ユーザー名)
```

そこに NICAM のチュートリアルパッケージをコピーします。

```
3. $ cp /home/k06003/nicam/nicam_src-20181009.tar.gz ./
```

```
4. $ cp /home/k06003/nicam/reg_nicam_gl05_rl00-20190905.tar.gz ./
```

うまくコピーできているでしょうか？ (`$ ls` で確認できます。) ついでに、次の 2 つのデ

ディレクトリをコピーしましょう。

```
5. $ cp -r /home/k06003/nicam/NICAM_DATABASE .
6. $ cp -r /home/k06003/nicam/FNL .
```

NICAM_DATABASE には NICAM が必要とする外部データが、FNL には初期条件データを作る際の参照データが入っています。最後に、NICAM のソースファイル（プログラムが書いてあるファイル）とチュートリアルで使用する雛形のディレクトリを展開しましょう。

```
7. $ tar zxvf nicam_src-20181009.tar.gz
8. $ tar zxvf reg_nicam_gl05_rl00-20190905.tar.gz
```

これで、ずらずらとファイルが展開され、NICAM16.3、reg_nicam_gl05_rl00 の 2 つの新しいディレクトリが作られたと思います（`$ ls -lF`で確認してみてください）。

次は本チュートリアルの実行環境の設定です。reg_nicam_gl05_rl00/set_env.sh を vi や emacs で開いて 3 行目の YOUR_ACCOUNT_NAME を自分のユーザー名に書き換えましょう。

```
9. $ source /work/gt32/(ユーザー名)/reg_nicam_gl05_rl00/set_env.sh
```

また、本チュートリアルで使う可視化プログラムへのパスを通します。

```
10. $ export PATH=$PATH:/home/c24028/mylocal/src/Nicoview-v2.33_2016112200
11. $ export PATH=$PATH:/home/k06003/grads/bin
```

ちなみに、ここで指定したパスの情報は、いったんログアウトすると消去されてしまいます。ログアウト/ログインのたびに export するのは少々面倒ですので、これを回避するために、9.と 10.と 11.の記述を /home/(ユーザー名)/.bashrc というファイルに書き込んでおきましょう。vi や emacs で上記のファイルを開いて、最後の行に 9.と 10.を追加してください。

また、NICAM を動かす際には必要ないのですが、結果の図を書く際に GrADS という描画ソフトを使います。そこで用いるのに必要なファイルの情報も.bashrc に記述しておきましょう。下記の 2 行をさらに追加します。

```
12. $ export GASCRP=/home/k06003/grads/data
13. $ export GADDIR=/home/k06003/grads/data
```

上記の編集が終わったら、.bashrc の情報を更新します。

```
14. $ source .bashrc
```

以上で NICAM チュートリアルの準備作業は終了です。まだまだ余裕ですか？それは先々有望ですね。少し不安ですか？分からないことがあれば質問してくださいね。すぐに慣れるので、心配しなくても大丈夫です。これまでに出てきた Linux のコマンドは、分からなければ“おまじない”程度に思っても差し支えありません。

3. make する

“make する”とは、プログラムを記述した複数のファイルを、正しい順番でコンパイルして、実際に動く実行形式を作成する、という一連の作業を表す言葉です。なぜ“make する”と言うのかというと、make という Linux (or Mac) のプログラムを利用するからです。NICAM でも実行形式を作成する際に make を使います。

早速 make の作業にとりかかりましょう。NICAM のディレクトリに移動し、OFP 用の環境を指定します。

```
1. $ cd NICAM16.3/NICAM
2. $ export NICAM_SYS=OFP
3. $ export ENABLE_NETCDF=F
```

3.は今回のインタラクティブノードを利用するためのオプションであり、通常は指定しません。新しいターミナルを開きなおした場合は 2.と 3.は再度行います。面倒な場合は ~/.bashrc に書き込んでおきましょう。

次がいよいよ make コマンドです。

```
4. $ make >& make.log &
```

これだけです。進捗状況は make.log に書き込まれていきますので、`$ less make.log` または `$ tail -f make.log` などとして確認することが出来ます。

無事 make コマンドが終了したら、

```
5. $ ls bin/
```

と入力して、実行形式置き場を見てみましょう。ファイルが 99 個ありますか？ (`$ ls bin/ | wc -l`) と入力して数えることができます。もちろん目で数えるのもあります。) 99 個確認できなければ make の時に何か問題が発生した可能性があるので、質問してください。bin/にある nhm_driver が NICAM を実行するプログラムです。このプログラムを動かすことが今回の目標ですが、まだしばらく準備作業が続きます。頑張っって前に進みましょう。

4. 領域分布記述ファイル(mnginfo)の作成

領域分布記述ファイル、というのはここだけの名前で、NICAM を使っている人は普通、マネージインフォファイルなどと呼ぶことが多いようです。このファイルには、正 20 面体格子を並列計算機で実行する際、どの領域がどの MPI プロセスに割り当てられ、他の領域とどのように繋がっているか、についての情報が記述されています。このように書いても、何を言っているのかさっぱり分からないと思いますが、それで何も問題ありません。NICAM を利用している研究者でも正 20 面体格子の詳細まで理解している人はあまり多くありませんし、興味がなければ理解する必要もありません。正 20 面体格子の人為的なルールより先に、我々が調べるべき大気現象はたくさんあります。今は演習に必要な mnginfo ファイルの作成に焦点を絞ります。

まず、先ほどの NICAM ディレクトリからチュートリアル作業ディレクトリの mnginfo

用ディレクトリに移動しましょう。

```
1. $ cd /work/gt32/(ユーザー名)/reg_nicam_gl05_rl00/mnginfo
```

ここで、先ほど `make` で作成した `mkmnginfo` コマンドを実行します。ここで今後のジョブの実行のための留意点があります。ログインしたマシン (ログインノードと呼びます) は、計算の実行の命令やデータの転送の利用が想定されており、先ほど `make` したプログラムは実行できません。実行したい場合は、計算専用の「計算ノード」を利用する必要があります。計算ノードの利用のために、ジョブスクリプトに実行したいコマンドを列記した、ジョブスクリプトファイルを作成し、ジョブスケジューラに投入します。ただし、ここからしばらく続く `NICAM` を走らせるための前段階の処理は、途中経過がわかる特別な計算ノードである「インタラクティブノード」を利用しましょう。インタラクティブノードは以下のように入力します。

```
2. $ pjsub --interact -g gt32 -L rscgrp=interactive-flat,node=1
```

このコマンドを実行すると、「Interactive job」が開始されたという表示がされます。

インタラクティブノードでジョブを実行した後、ログアウトすると、もとのログインノードに戻ります。このチュートリアルでは、インタラクティブノードを利用しなければいけないコマンドは“\$”ではなく“>”で始まるように記載します。

では `mkmnginfo` を実行します。相対パス (自分が現在いる場所からみたターゲットの場所) で記述すると、

```
3. > ../../NICAM16.3/NICAM/bin/mkmnginfo
```

です。 `rl00-prc000010.info` というファイルが生成されたでしょうか (`$ ls` で確認)。`mkmnginfo` はこのディレクトリにある `mkmnginfo.cnf` という設定ファイルを参照して `rl00-prc000010.info` という領域分布記述ファイルを生成しています。`mkmnginfo.cnf` を見ると (`$ less mkmnginfo.cnf`)、領域分割レベルが 0 で、並列プロセス数が 10 の時の領域分布記述ファイルを `rl00-prc000010.info` という名前出力せよ、という設定が確認できます。このような詳細については本演習では分からなくても大丈夫です。

それでは、インタラクティブノードから抜け、一旦ログインノードへ戻りましょう。

```
4. > exit
```

今後、実行例の “\$” と “>” が入れ替わるたびに、インタラクティブノードへのログイン・ログアウトが必要です。少々面倒ですが、10 回以上あります。もし間違えると、「CPU の形式が違うためプログラムが実行できない」というエラーになります。またインタラクティブノードへのログイン中に `cd` コマンドによってディレクトリを移動したとしても、`exit` すると元いたディレクトリに戻ってしまいます。

5. 正 20 面体格子データの作成

さて、正 20 面体格子の格子位置情報データを作成しましょう。正 20 面体格子は、基本

はサッカーボールのような形の格子で、その 6 角形/5 角形を細かく分割していくと思っておいてください。

まず、サッカーボールを分割した構造を作ります。チュートリアルの実行ディレクトリの `grid/hgrid` に移動しましょう。

```
1. $ cd /work/gt32/(ユーザー名)/reg_nicam_gl05_rl00/grid/hgrid
```

ここで、格子作成プログラム `mkgrid` を実行します。

```
2. > mpirun -n 10 ../../../../NICAM16.3/NICAM/bin/mkgrid
```

どのような格子が作成されたか不安ではありませんか？ここで、`nicoview` というプログラムを使って格子の構造を見てみましょう。

```
3. $ nicoview view_grid.cnf
```

1:X11, 2:PS, 3:Tek, 4:Gtk ; と出力先の選択肢がされるので、1 と入力します。以後 `nicoview` を使用する際も同様です。

3 角形の頂点に相当する場所が 6 角形/5 角形セルの中心点になります。プログラムを終了するには、図が表示された窓をクリックしてください。

次に、北極に格子を集中化するために `transgrid` を実行します。

```
4. > mpirun -n 10 ../../../../NICAM16.3/NICAM/bin/transgrid
```

```
5. $ nicoview view_s10grid.cnf
```

その後、格子集中化した領域を高解像度でシミュレーションしたい領域に移すため、`rotategrid` を実行します。

```
6. > mpirun -n 10 ../../../../NICAM16.3/NICAM/bin/rotategrid
```

```
7. $ nicoview view_rotated.cnf
```

日本のあたりに格子の密な場所が移動しています。最後に、格子点を 6 角形の重心に再定義するべく、`mkggrid` を実行します。

```
8. > mpirun -n 10 ../../../../NICAM16.3/NICAM/bin/mkggrid
```

```
9. $ nicoview view_grid.cnf
```

これら一連のプログラムの実行で `grid.rgn000[00-09]` という正 20 面体格子データファイルが生成されます (`$ ls` で確認)。それぞれのプログラムの設定ファイルは、`mkgrid.cnf`、`transgrid.cnf`、`rotategrid.cnf`、`mkggrid.cnf` という名前がこのディレクトリにあらかじめ置かれています (興味があれば中を見てみてください)。ちなみに、集中格子でない通常の全球準一様な正 20 面体格子の場合には上記 `transgrid`、`rotategrid` の過程は必要なく、省略可能です (その際は設定ファイル `mkggrid.cnf` の `input_base` に気をつけてください)。

6. 緯度経度変換重みデータの作成

次に、作成した正 20 面体格子データを使って、緯度経度変換時に使用する内挿重みデータを生成しておきます。このデータは `NICAM` の実行に直接関係しないのですが、この演習の最後のステップで可視化用のデータを作成する際に使用します。

チュートリアル作業ディレクトリの `grid/llmap` に移動し、`mkllmap` というプログラムを使用します。

```
1. $ cd /work/gt32/(ユーザー名)/reg_nicam_gl05_rl00/grid/llmap
2. > mpirun -n 10 ../../../../NICAM16.3/NICAM/bin/mkllmap
```

このプログラムの実行により `llmap.rgn000[00-09]` というファイルが生成されます (`$ ls` で確認)。これで緯度経度変換重みデータの作成は終了です。

7. 鉛直格子データの作成

次に鉛直格子を作成しましょう。先ほど作成した正 20 面体格子は球（例えば地球）の表面を 6 角形/5 角形で覆うような構造でした。しかし、我々が理解の対象としている大気の運動は 3 次元世界で起こっています。どんなに薄く儂い感じがしても、大気には水平方向だけでなく鉛直方向の運動があり、大気大循環において重要な役割を果たします。そのようなわけで、大気の運動をモデル化するには球面に直交する方向（高さ方向）の格子が必要です。

前置きが長くなりましたが、さくさくと鉛直格子データを作成してしまいましょう。チュートリアル作業ディレクトリ中の `grid/vgrid` に移動し、`mkvlayer` というコマンドを利用します。

```
1. $ cd /work/gt32/(ユーザー名)/reg_nicam_gl05_rl00/grid/vgrid
2. > ../../../../NICAM16.3/NICAM/bin/mkvlayer
```

このプログラムの実行で `vgrid40.dat` という鉛直格子データファイルが生成されます (`$ ls` で確認)。

8. 地形・植生データを作る

これまでに、最初に提示した 13 個の行程のうち約半分の作業が終わったこととなりますが、どのように感じていますか？よく分からなくなってきた、という人は最初に整理した項目リストに戻ってこれまでの行程を確認してみてください。簡単にまとめると、ファイルを展開し、NICAM の実行形式を `make` し、球面上に 3 次元座標を作成しました。

これまでに作成したデータを読み込むことで、NICAM は 3 次元空間上のどの場所に計算ノードをとればよいか、を知ることができます。実行を開始する時点では NICAM は何の情報も持たないのですから、これはかなりの進歩です。しかしながら、地球大気のシミュレーションを行うには NICAM にもっと多くのことを教える必要があります。例えば、NICAM はどこが海でどこが大陸かを知りませんし、大陸上に山があつてそれがどの程度の高さなのかも知りません。これらの情報を NICAM に教えるためのデータを作成しましょう。

まず、チュートリアル作業ディレクトリの地形・植生データ用の場所

(`external_data/MATSIRO`)に移動しましょう。詳しくは触れませんが、`MATSIRO` とい

うのは NICAM や気候モデル MIROC が利用している地表面計算プログラムの名前です。
ここでは、mklanddata2 というプログラムを動かします。

```
1. $ cd /work/gt32/(ユーザー名)/reg_nicam_gl05_r100/external_data/MATSIRO
2. > mpirun -n 10 ../../../../NICAM16.3/NICAM/bin/mklanddata2
```

この作業により、topog.rgn000[00-09]、topog_std.rgn000[00-09]、
veget_mat.rgn000[00-09]、slidx.rgn000[00-09]、albedo_nir.rgn000[00-09]、
albedo_vis.rgn000[00-09]、lai_clm_1982-1998.rgn000[00-09]、
landmask.rgn000[00-09]、sl_area.rgn000[00-09]、gradz.rgn000[00-09]という複数種類の
ファイルが作成されます (`$ ls` で確認)。

以下のように描画してイメージを掴みましょう。

```
3. $ nicoview view-topog.cnf      (標高)
4. $ nicoview view-veget_mat.cnf  (植生タイプ)
5. $ nicoview view-slidx.cnf      (土壌 index)
6. $ nicoview view-albedo_nir.cnf (近赤外アルベド)
7. $ nicoview view-albedo_vis.cnf (可視アルベド)
8. $ nicoview view-lai_clm.cnf    (leaf area index)
9. $ nicoview view-landmask.cnf   (海陸マスク)
10. $ nicoview view-sl_area.cnf   (領域面積)
11. $ nicoview view-gradz.cnf     (傾斜)
```

ここで実行した mklanddata2 は非常に大きなメモリを要求するプログラムなので、別の
環境や別の解像度で実行する際には、メモリの消費量に注意する必要があります。

9. オゾンデータを作る

次に NICAM に教えてあげるのはオゾンの分布です。「一般気象学」などで大気の温度成
層について勉強したことがあるでしょうか？地球大気の大気圏においては、標高が 100m 高
くなるごとに約 0.6 度温度が下がることが知られています。ところが対流圏を超えて成層圏
に達すると、対流圏とは逆に、高度が上がるにつれて温度が高くなっていきます。その理由
は成層圏にはオゾンが濃い高度が存在し、オゾンが太陽からの短波放射を吸収して温度が
高くなるためです。NICAM で雲の高さをもっともらしく計算するには、対流圏と成層圏の
境目である対流圏界面付近において、その安定な温度成層が現実的であることが必要です。
そのような理由から、対流圏の計算が目的の場合でも、(特にシミュレーションが長期の場
合には) オゾンデータが必須です。

オゾンのデータはチュートリアル作業ディレクトリの external_data/O3 で作成します。
使用するプログラムは mko3 です。

```
1. $ cd /work/gt32/(ユーザー名)/reg_nicam_gl05_r100/external_data/O3
2. > mpirun -n 10 ../../../../NICAM16.3/NICAM/bin/mko3
```

プログラムの実行により、`amip2o3.rgn000[00-09]`というデータファイルが作成されます (`$ ls`で確認、`$ nicoview view-amip2o3.cnf`で描画)。

10. 初期条件データを作る

NICAM に教える地球についての情報の最後は、初期条件と呼ばれるものです。NICAM は、ある時点の気象状態を起点とし、決められた（理論的・経験的）物理法則に従って気象状態の時間変化を計算します。そのため、物理的に無理のない気象状態を起点として与える必要があります。“物理的に無理のない”と書きましたが、例えば、目の前にコップがあるとして、コップに入っている水の重さが-100g だったらどうでしょう？-100g の水が増えたり減ったり、というのはただの数字としては計算できますが、物理的な意味付けには困りますよね？この場合、水の重さは正の値であることが期待されています。この例のように、大気の密度や温度にも“それらしい”範囲というものがあるって、そこから著しく逸脱した値を教えると、NICAM の計算が破綻してしまったりします。この演習では、ある日(2018年7月4日)の現実的な気象状態を起点として与え、その後の気象状態の時間変化を計算することにします。

初期条件の作成には少し準備が必要になります。チュートリアル作業ディレクトリの `local` に移動し、`make` プログラムを走らせます。

1. `$ cd /work/gt32/(ユーザー名)/reg_nicam_gl05_rl00/local`
2. `$ make`

これで、`local/bin/`に `conv_lnd`、`conv_ocn`、`conv_ocn_ndg`、`swap_by_clmsnw` というプログラム群が生成されたと思います。これらのプログラムを利用（するシェルプログラムを利用）して初期条件を作成していきましょう。

初期条件の作成はチュートリアル作業ディレクトリ中の `initial_data` で行います。

3. `$ cd /work/gt32/(ユーザー名)/reg_nicam_gl05_rl00/initial_data`

まず、大気、海洋、陸面の元データを作成します。

4. `$ sh mk_atm2.sh ../../FNL/fnl_20180706_18_00.grib2`
5. `$ sh mk_ocn2_s1.sh ../../FNL/fnl_20180706_18_00.grib2`
6. `> sh mk_ocn2_s2.sh 7`
7. `$ sh mk_lnd2_s1.sh ../../FNL/fnl_20180706_18_00.grib2`
8. `> sh mk_lnd2_s2.sh 7`

これらのプログラムにより、`atm.dat`、`ocn.dat`、`lnd.dat` という3つのデータが生成されます (`$ ls`で確認)。`mk_ocn2_s2.sh` と `mk_lnd2_s2.sh` を実行する際に、末尾に”7”をつけましたが、これは初期条件の月が7月であるためです。初期条件に応じて適宜変更してください。次に、それぞれのデータを正20面体格子系に変換しましょう。

9. `> cd ATM`
10. `> ../../../../NICAM16.3/NICAM/bin/mkinit_atm_new`

```
11. > cd ../OCN
12. > ../../../../NICAM16.3/NICAM/bin/mkinit_ocn
13. > cd ../LND
14. > ../../../../NICAM16.3/NICAM/bin/mkinit_lnd
```

これらのプログラムを実行することで、ATM/init_atm.rgn000[00-09]、OCN/init_ocn.rgn000[00-09]、LND/init_lnd.rgn000[00-09]のように初期条件ファイルが生成されます (`$ ls` で確認)。複数の変数をまとめたファイルになっていて、例えば大気については ATM ディレクトリで `$ nicoview view-initatm.cnf` として変数の一つである気圧を描画できます。

初期条件ファイルの作成で、NICAM 実行までの大きな山を越えました。みなさんもだいぶ疲れてきたのではないのでしょうか？ 準備はあとわずか、残りは海面温度の参照データと基準成層データです。ここまで頑張っておきかえらぬのも勿体ないので、なんとか付いてきて下さい。

11. 海面温度データを作る

ところで、なぜ南の島は暖かいのでしょうか。熱帯に行けば、太陽は真上から照りつけるようになり、直射日光は確かに暑く感じます。しかし、残念ながら対流圏の大気は太陽の短波放射を効率よく吸収できる訳ではありません。むしろ、陸面や海水の温度上昇を介して、顕熱・潜熱の形で多くの熱を受け取ります。NICAM では、顕熱・潜熱フラックス計算に必要とされる海面温度の時間変化を海洋混合層モデルで計算しますが、その計算のために海面温度の参照データが必要になります。

海面温度の参照データはチュートリアル作業ディレクトリ中の `nudging_data` で、`mk_sst2.sh` を使って作成します。

```
1. $ cd /work/gt32/(ユーザー名)/reg_nicam_g105_r100/nudging_data
2. $ sh mk_sst2_s1.sh ../../FNL/fnl_201807
3. > sh mk_sst2_s2.sh
```

このシェルスクリプトの実行により `sst.dat` というデータが生成されます。次に、このデータを正 20 面体格子系に変換しましょう。

```
4. $ cd OCN
5. > mpirun -n 10 ../../../../NICAM16.3/NICAM/bin/mkgisst
```

`sst.rgn000[00-09]` というデータが生成されます。例によって SST データの中身をチェックしてみましょう。

```
6. $ nicoview view-extsst.cnf
```

6 角形上のセルのそれぞれに温度が設定されていることが確認できます。ここでついでに、海面温度に加えて海氷のデータも作っておきましょう。

```
7. $ cd ../OCN_CLM
```

```
8. > mpirun -n 10 ../.././NICAM16.3/NICAM/bin/mkgisst
```

このプログラムの実行により、cmip3ice1979-1999_clm.rgn000[00-09]、cmip3snw1979-1999_clm.rgn000[00-09]、hadsst1979-1999_clm.rgn000[00-09]という3種類のファイルが生成されます。

12. 大気の基準成層データを作る

大気の基準成層データは、音波や重力波の項を線形化する際に使われますが、ここでは詳細にこだわらずデータの作成に集中しましょう。

```
1. $ cd /work/gt32/(ユーザー名)/reg_nicam_gl05_rl00/refstate
2. > ../.././NICAM16.3/NICAM/bin/mkbsstate
```

この作業により vgrid40_ref.dat というファイルが生成されます。これで準備完了です。様々な作業が五里霧中なうちに連続し、“つらいなあ”と感じた人も多かったかもしれません。よくここまで我慢してくれました。次はいよいよ NICAM のメインプログラムの実行です。

13. NICAM を動かす

ようやくここまで辿り着きました。いよいよ NICAM を実行しましょう。

```
1. $ cd /work/gt32/(アカウント)/reg_nicam_gl05_rl00/run
2. $ pjsub exec_nhm_driver_forOFP_gl09rl05.sh
```

以上です。ここでは pjsub というコマンドを用いて exec_nhm_driver_forOFP_gl09rl05.sh というファイルを実行しています。このファイルにはプログラムを実行するジョブをジョブ管理システムに投入するように記述されています。なお、exec_nhm_driver_forOFP_gl09rl05.sh には「CPU 数が 1 つ (ただし、ひとつの CPU は 64 コアもっています) で最大実行時間が 1 時間のジョブをシステムに投入せよ」という命令が書かれています (`$ pjstat` でジョブの実行状態を確認)。ここで投入する計算自体は 30 分前後で終わりますので、気長に待っててください。計算が終わるまでの間、ここまでの作業を振り返るなり、頭を休めるなり、モデル中の時計の進行を眺めるなり `$ tail -f msg.pe000000`、nhm_driver.cnf の中を見て訳がわからないなあと思うなり、のんびりとお過ごしください。

14. 出力データを緯度経度変換し、計算結果を確かめる

計算が終わったら、出力データを正 20 面体格子から緯度経度格子に変換して解析してみましょう。計算を実行したディレクトリ下の check に移動し、変換プログラム ico2ll を動かします。

```
1. $ cd check
2. > ../.././NICAM16.3/NICAM/bin/ico2ll
```

*.ctl と *.grd というファイルが複数作成されたことと思います。これらのデータを GrADS

という描画ソフトを用いて解析してみましょう。計算結果のアニメーションを作成するために GrADS 用スクリプトのサンプルとして vl2019_grads.gs を用意しています。

```
3. $ grads -lc vl2019_grads.gs
```

を実行すると連番画像が作成されます。連番画像からアニメーションを作成します。

```
4. $ convert -delay 10 im-*.png anim.gif
```

作成した gif アニメーションを各自の PC にダウンロードして、ブラウザで再現結果を眺めてみてください。

15. 衛星シミュレータ利用のための準備

14.で作成した緯度・経度データに加え、衛星シミュレータのために必要な追加の緯度・経度データ（標高と植生）を作成しておきます。

```
1. $ cd /work/gt32/(ユーザー名)/reg_nicam_gl05_rl00/external_data/MATSIRO/check
2. > ../../../../NICAM16.3/NICAM/bin/ico2ll
```

必要なファイルをリストアップしますので、衛星シミュレータ用のディレクトリ (/work/gt32/(ユーザー名)/NICAM_outputs) にコピーしておいてください。以下のデータそのもの(*.grd)と対応するコントロールファイル(*.ctl)が必要です。例えば、ms_pres.grd に対して ms_pres.ctl があり、両方を用意します。

5分ごとの3次元データ			
ms_w.grd	鉛直速度 [m/s]	ms_qc.grd	雲水混合比 [kg kg ⁻¹]
ms_pres.grd	気圧 [Pa]	ms_qr.grd	雨水混合比 [kg kg ⁻¹]
ms_tem.grd	気温 [K]	ms_qi.grd	雲氷混合比 [kg kg ⁻¹]
ms_qv.grd	水蒸気混合比 [kg kg ⁻¹]	ms_qs.grd	雪混合比 [kg kg ⁻¹]
		ms_qg.grd	霰混合比 [kg kg ⁻¹]

5分ごとの2次元データ			
ss_slp.grd	海面気圧 [Pa]	oa_ice.grd	海氷量 [kg m ⁻²]
sa_tem_sfc.grd	地表面温度 [K]	oa_ocr.grd	氷/雪の比
ss_t2m.grd	2-m 気温 [K]	oa_snw.grd	海氷上の雪 [kg m ⁻²]
ss_q2m.grd	2-m 水蒸気混合比 [kg kg ⁻¹]	oa_ist.grd	氷の表面温度 [K]
ss_u10m.grd	10-m 東西風速 [m s ⁻¹]		
ss_v10m.grd	10-m 南北風速 [m s ⁻¹]		

時間変化しない2次元データ			
topog.grd	地表面高度 [m]	veget_mat.grd	植生指標