

1 GTOOL とは

GTOOLとは、CCSR/NIES AGCM で出力される形式 (GTOOL3 形式) のデータを解析、図化するためのツールのことです。データの切出し、データ間の差、空間・時間平均をとるといった簡単な操作をコマンド1つで行うことができ、解析の高速化が可能です。また、パワースペクトル、移動平均、水平発散、水平渦度、流線関数の計算や σ 、 p 、 z 変換といったデータ解析でよく使われるフィルタコマンドがあらかじめ用意されているため、ある程度のことはプログラムを作成しなくても行うことができます。さらに、GTOOLには描画コマンドがあり、GrADSのようにコントロールファイルを必要としないため、GCMの出力結果をすばやく表示することが可能です。ただ、描画コマンドの出力はGrADSやGMTなどの可視化ソフトに比べてかなり劣るため、発表用の図の作成には向きません。また、GTOOLを使用する人はかなり限定されるため、Webで検索しても参考になるページがほとんどないのも欠点かもしれません。

2 GTOOL の使用法

2.1 GTOOL を使用するための設定

GTOOLを使用する前に、次の変数を設定します。ここでは、climate5,6,7の場合を説明します。各研究室のサーバに関しては研究室の計算機管理者にお問い合わせ下さい。

- 実行ファイルへのパス

シェル変数PATHには実行ファイルへのパスが書かれていますので、その中にGTOOLのバイナリがあるディレクトリを追加します。¹実行ファイルへのパスは、climate7では/usr/local/gtool/bin、climate5,6では/usr/local/gt3-dcl5/bin/Solarisです。²例えば、.cshrcに次のような記述を書き加えます。

```
if ("$HOST" == "climate7") then
  set path = ( $path /usr/local/dcl-5.0.2/bin \
              /usr/local/gtool/bin )
else
  set path = ( $path /usr/local/dcl-5.0.2/bin \
              /usr/local/gt3-dcl5/bin/Solaris )
endif
```

¹実行ファイルへのパスに限っては、シェル変数PATHを変更すると環境変数PATHも同時に変更されます。

²/usr/localは/opt/localへのシンボリックリンクとなっているため、実際には/opt/local/gtool/binのような所に存在します。

- 環境変数 GTOOLDIR

GTOOLDIR には GTOOL がインストールされたディレクトリを指定します。climate7 では /usr/local/gtool, climate5,6 では /usr/local/gt3-dcl5 です。

記述例

```
if ("$HOST" == "climate7") then
  setenv GTOOLDIR /usr/local/gtool
else
  setenv GTOOLDIR /usr/local/gt3-dcl5
endif
```

- 環境変数 GTAXDIR

格子情報ファイルの置き場所を指定します。³デフォルトでは \$GTOOLDIR/lib/gt3 ですので、自分のホームディレクトリなど他の場所に置きたい場合には指定します。この変数を指定するとデフォルトディレクトリは読み込まれなくなるので注意して下さい。

- 環境変数 GTTMPDIR

gtool の入出力オプションでファイル名を指定しないと、gtool.out というファイルを読み書きします。GTTMPDIR を指定するとそのディレクトリに gtool.out が作成されるようになります。

2.2 GTOOL で扱える形式

GTOOL では GTOOL3 形式 (CCSR/NIES AGCM の出力はこれ) のファイルのみを取り扱います。⁴GTOOL3 形式はデータ本体の情報を記述したヘッダとデータ本体で構成され、順アクセスバイナリファイルになっています。⁵ヘッダは 16 文字 × 64 欄の文字データで、データ本体は書式なしバイナリデータです。GCM の出力の場合は通常、時刻の違う複数のデータが 1 つのファイルの中に書き込まれた状態になっています。ヘッダの詳細は GTOOL、GCM のマニュアルを見て下さい。

2.3 コマンド入力

GTOOL のコマンドは基本的に次のような形式をしています。

```
$ gt_____ [入力ファイル名] [オプション]
```

³GTOOL のインストールの際に格子点の位置をあらかじめ格子情報ファイルとして作成してあります。GTOOL はデータのヘッダの中に記述された格子情報ファイルの名称をもとにそのファイルを随時読み出して使用しています。

⁴データ形式の変換コマンドでは、直接アクセスバイナリファイル、ASCII ファイルを扱えるコマンドもあります。

⁵Fortran の open 文で access='sequential', form='unformatted' として読み込む形式。データとデータの間には改行コードが入ります。

(原則として入力ファイル名とオプションの順番はいつでもよく、
オプション名は大文字小文字を区別しません。)

オプションの種類

数オプション : z=1 のように、オプション名=値とする

文字オプション : title:Rain のように、オプション名:値とする

論理オプション : print=t clabel=f のようにオプション名=t/f とする。
(print=t は-print, clabel=f は#clabel のように略記可)

3 主要なコマンド

3.1 描画コマンド

3.1.1 gtcont

gtcont はコンター図を作成するためのプログラムです。x,y,z オプションでデータから切り取る断面を選択します。

```
$ gtcont <file> x=15 (東西方向の15個目のグリッドの南北鉛直図)
$ gtcont <file> x=15 y=1,10 (yの範囲を1~10個目のグリッドに限定)
$ gtcont <file> x=0 (0を指定すれば、東西平均の緯度-鉛直プロット)
$ gtcont <file> x=0 -exch (-exch オプションで軸を交換)
```

時刻データの限定

```
$ gtcont <file> str=2 end=8 (2~8番目までをプロット)
$ gtcont <file> str=2 end=8 step=2 (2個ごとにプロット)
```

コンター間隔を変える

```
$ gtcont <file> cont=5 (コンター間隔を5にする)
$ gtcont <file> cont=5,10 (10ごとに太線にする)
$ gtcont <file> cont=5,10 range=0,100 (0-100の間のみ描く)
```

値による塗り分け

```
$ gtcont <file> tone=0,100 pat=14 (0-100までトーンをかける)
$ gtcont <file> tone=-100,100,200 pat=14,24 (2つのパターンで塗り分け)
$ gtcont <file> color=20 (20色で塗り分ける)
```

東西平均した図で西風領域のみにトーンをかけるには?

その他

```
$ gtcont <file> color=20 map=1 (海岸線を重ねてみる)
$ gtcont <file> title:'Averaged temperature' (タイトルを付ける)
$ gtcont <file> dset:'AGCM5.6' (右上のDataset名を指定)
$ gtcont <file> -print ps:out.ps (out.psに出力)
```

3.1.2 gtvect

gtvect は2つのデータからベクトル図を作成するコマンドです。

```
$ gtvect <file1> <file2> (1つ目がx軸、2つ目がy軸成分)
$ gtvect <file1> <file2> intv=3,3 (3点おきにプロット)
$ gtvect <file1> <file2> fact=0.5 (ベクトルの長さを0.5倍に)
$ gtvect <file1> <file2> range=-10,10 (最大値10としてスケーリング)
```

ベクトルとコンターを同時に描くコマンドもある

```
$ gtvecon u v Z intv=3,3 x=30,100 y=1,37 map=1
(東西方向の30-100番目と南北方向の1-37番目のグリッドを選択)
```

3.2 フィルタコマンド

3.2.1 gtsel, gtavr

gtsel はデータの切出し、断面の作成を行うコマンド、gtavr は時間平均の作成を行うコマンドです。

```
$ gtsel <file1> x=0 ; gtcont (東西平均)
$ gtsel <file1> x=30,100 y=1,37 out:<file2>
(さっきのグリッドを切り出す。)
$ gtsel <file1> z=5,10 str=1 end=32 step=4 ; gtsel z=0 ; gtcont
(これは?)
$ gtavr <file1> out:<file2> str=2 end=10 step=2 (2-10まで2つおき平均)
$ gtavr <file1> out:<file2> ostep=30 (30個ごとの平均をとる)
```

最初の30日平均と次の30日平均の東西風の図を作り、西風領域だけをカラーで表示したい。

3.2.2 簡単な計算コマンド

gtadd(和), gtsub(差), gtmlt(積), gtdiv(商), gtlog(対数), gtsqrt(平方根) コマンドなどがあります。2つのデータに別々のファクタとオフセットを作用させることも可能です。

```
$ gtsub <file1> <file2> out:<file3> fact1=0.1 ofs1=1 ofs2=3
( (0.1*<file1> + 1) - (1*<file2> + 3) )
```

最初の30日と次の30日の東西風の差を取ったら?

```
$ gtlog <file1> out:<file2> (logAの計算)
```

gtedy(擾乱を計算するコマンド)の出力を使って擾乱の起こりやすい場所を調べたい。

```
$ gtedy -x Z out:z-eddy (東西平均からの擾乱成分を計算)
$ gtmlt z-eddy z-eddy ; gtavr str=1 end=30 ; gtsqrt ; gtcont map=1(これは?)
```

3.2.3 移動平均、パワースペクトル

移動平均を取るための `gtmavr`、パワースペクトルを計算するための `gtpwspct`, `gtstspct` と
いったデータ解析でよく使われるフィルタコマンドがあります。

```
$ gtmavr <file1> out:<file2> -x peri=10 (x軸の方向に10点移動平均)
$ gtpwspct <file1> out:<file2> -x (x軸方向のパワースペクトル)
$ gtstspct <file1> out:<file2> (時空間パワースペクトル)
(第1次元を経度、第3次元を時間とする必要あり)
```

3.2.4 発散、渦度、流線関数

球面調和関数展開を用いた水平発散 (`gthdiv`), 水平渦度 (`gthvor`) の計算、南北風からの子
午面流線関数 (`gtstrm`) の作成といったことがプログラムを作成することなく行えます。⁶⁷

```
$ gthdiv u v out:<file> (水平発散の計算)
$ gthvor u v out:<file> (水平渦度の計算)
$ gtstrm v ps:slp str=1 end=1 ; gtcont
(子午面流線関数の計算。file1は南北風、file2は地表面気圧データ)
```

3.2.5 温位、相当温位、仮温度、飽和比湿、相対湿度

```
$ gttheta <file1> ps:<file2> out:<file3> (温度から温位を計算)
$ gttheta <file1> ps:<file2> out:<file3> -inv (温位から温度を計算)
$ gtthetae t:<file1> q:<file2> ps:<file3> out:<file4> (相当温位を計算)
(tで温度、qで比湿、psで地表面気圧データを指定する)
$ gttv t:<file1> q:<file2> out:<file3> (温度、比湿から仮温度を計算)
$ gtqsat t:<file1> ps:<file2> out:<file3> (飽和比湿)
$ gtrelh q:<file1> t:<file2> ps:<file3> out:<file4> (相対湿度)
```

3.2.6 座標変換

z 座標から p 座標への変換 (`gtz2p`)、 σ 座標から p 座標 (`gts2p`) への変換が可能です。

```
$ gtz2p <file1> p:<file2> out:<file3> -apnd
(3次元の気圧データを p で指定。追加書き込みを行う。)
$ gts2p <file1> ps:<file2> out:<file3> #misout
(地表面気圧が指定気圧より低い場合に外挿する)
```

⁶水平発散、水平渦度の計算には惑星の半径を使用します。火星 GCM の出力を利用する場合には
`aer=3.397E+06` のようなオプションを付け、半径を変更します。

⁷`ps` オプションに一貫性がないのは我慢して下さい。気に入らなかつたら直しましょう。

3.3 データ形式変換コマンド

3.3.1 gtmdir

gtmdir は GTOOL3 形式から直接アクセスバイナリファイル (GrADS 形式) に変換するコマンドです。⁸

```
$ gtmdir <file1> out:<file2> -r4 (実数 4 バイトで出力)
$ gtmdir <file1> out:<file2> -i2 (整数 2 バイトで出力)
```

エイリアスとして .cshrc に記述すると便利

```
alias gtw 'gtmdir \!* out:\!$.grads -r4'
(gtw <file 名>で<file 名>.grads ファイルを作る)
```

3.3.2 gtrdir

gtrdir は GrADS 形式のファイルを GTOOL3 形式のファイルに変換するコマンドです。他の形式の気象データを一旦 GrADS 形式にしてから GTOOL で読めるようにしたい場合に使用します。x,y,z オプションで各軸の格子情報ファイル名称を指定します。格子情報ファイルは \$GTOOLDIR/lib/gt3 にあります。格子情報ファイル名称はここにあるファイルの GTAXLOC.____の部分を使います。

```
$ gtrdir <file1> out:<file2> -r4 x:GLON144 y:GLAT73 z:NCEPL17
(NCEP/NCAR 再解析データの場合)
$ gtrdir <file1> out:<file2> -r4 x:GLON144 y:GLAT73 z:SFC1
(地表面データを変換)
$ gtrdir <file1> out:<file2> -r4 x:144 y:73 z:17
(格子情報ファイルがない場合、とりあえず格子数を指定)
```

ヘッダ情報を編集することもできます。

```
$ gtset T out:T-out title:'Temperature' dset:'NCEP/NCAR' item:'T'
unit:'K' utim:'DAY' styp=2
(styp=2 で鉛直方向を log スケールに)
```

4 GTOOL のマニュアルなど

Web ページを探しましたが、ある程度役に立ちそうな GTOOL のページは、GTOOL のマニュアルを HTML で書き直した

<http://www.riam.kyushu-u.ac.jp/taikai/labs/Gtool/> のページぐらいでした。

マニュアルの場所

GTOOL のマニュアルは \$GTOOLDIR/doc/Guide.tex にありますので、コンパイルして印刷して下さい。既にコンパイルして pdf に変換したものが

⁸出力されるデータの endian は入力データと同じです。big/little 変換をするには -r4r, -i4r のようなオプションをつけます。

/net/climate7/export/home/yousuke/gtool/doc/Guide.pdf にありますので、こちらを利用して下さっても構いません。

この資料は <http://157.82.240.165/~yousuke/keisannki/gtool.pdf> として公開していません。

GTOOLのマニュアルではありませんが、UNIXの設定ファイルについての解説を <http://157.82.240.165/~yousuke/keisannki/init.html> でしています。興味のある方はご参照下さい。

Appendix

研究室のサーバなど他の計算機から climate7 の GTOOL を使用方法

climate7 の/opt はセンターの計算機に対してエクスポートされていますので、ネットワーク上で climate7 の/opt にアクセスすることが可能です。climate7 のプログラムを使用するには CPU と OS に互換性が必要です。CPU が Sun Ultra SPARC 系列 (Sun Blade、Sun Enterprise サーバ等)、OS が Solaris8,9(他は確認していません) なら使用できると思います。ここでは climate7 の/opt が他の計算機の/net/climate7/opt にマウントされていることを想定して書いています。⁹

```
$ setenv GTOOLDIR /net/climate7/opt/local/gtool
$ set path = ( $GTOOLDIR/bin $path)
($ setenv PATH $GTOOLDIR/bin:"$PATH" でもよい)
$ setenv GTAXDIR $GTOOLDIR/lib/gt3
(既に GTAXDIR が設定されている場合には変更する。自分で格子情報ファイルを持っている場合にはこのようにしなくてもよい)
```

GTOOL のインストール

自分で GTOOL のソースコードを書き換えて使用したい方や、新しい計算機や自分の端末に GTOOL をインストールしたい場合には参考にして下さい。描画機能を充実させたり、配布されている気象データの形式 (例えば、NETCDF) から GTOOL3 形式に変換するコマンドを作ってみたりするのもいいかもしれません。

- DCL のインストール

GTOOL をインストールするには、先に DCL をインストールする必要があります。climate5,6,7 では既にインストールされています。

DCL5.2 以前の場合

```
$DCLDIR/sys/Mkinclude. システム名 を $DCLDIR にコピーして
$ make
$ make install
```

⁹UNIX ではネットワーク上のディレクトリ A をローカルマシンのディレクトリ B の下にマウントすると、あたかもディレクトリ B がディレクトリ A に置き換わったように扱うことができます。Solaris で自動マウントの設定をした場合、特殊な設定をしない限り/net/ホスト名/—にマウントされます。

DCL5.3 の場合

```
$ ./configure --prefix=インストール先
$ make
$ make install
```

- 環境変数の設定

GTOOL のインストール先は環境変数 GTOOLDIR で決まるので、GTOOLDIR を設定します。次に、GTOOL のインストールの際に DCL ライブラリを使用できるように環境変数 LD_LIBRARY_PATH の設定を行います。¹⁰Solaris の場合、通常は LD_LIBRARY_PATH に既にシステムが使用するライブラリのパスが入っているので、この値に追加します。

```
$ setenv DCLDIR /usr/local/gtool
$ setenv LD_LIBRARY_PATH $DCLDIR/lib:"$LD_LIBRARY_PATH"
```

- Mkinclude の編集

\$GTOOLDIR/Mkinclude の最初の方にシステムを設定する部分 (SYSTEM=の部分) がありますので、自分のシステムに合わせて編集します。climate サーバでは SYSTEM=Solaris です。他には Linux、DEC-OSF(Tru64 の場合) などの指定が可能です。この指定により、\$GTOOLDIR/libsrc/sysmake/Makedef.\$SYSTEM が読み込まれるようになります。Makedef.Solaris の SYSXLIBNAME で指定されたディレクトリは climate サーバでは存在しないので /usr/openwin/lib に変更します。コンパイラを変更したい場合には FC、CC、LD を変更し、コンパイラオプションを変更したい場合には SYSFFLAGS、SYSCFLAGS などを変更します。

- コンパイルとインストール

```
$ make dirs (インストールする directory を作成)
$ make
$ make install ($GTOOLDIR 以下にインストールされる)
```

¹⁰ほとんどのプログラムは実行時にライブラリを呼ぶようにコンパイルされていますが、この実行ライブラリの置き場所を LD_LIBRARY_PATH で指定します。複数のプログラムで同じライブラリを使用する場合にメモリを節約できる、ライブラリを共有した分プログラムの合計サイズが小さくなるといった利点があります。

GTOOL3 から GrADS 形式への変換プログラムの例

```
parameter(imax=64, jmax=32, kmax=31)
character HEAD(64)*16
character fname1*100, fname2*100
real*4    GD(imax,jmax,kmax)
undef = -999.0

c
  read(5,*) rmin, rmax, fname1, fname2
  open(10, file=fname1, form='unformatted', status='old',
    &access='sequential')

  open(51, file=fname2, form='unformatted', status='unknown',
    &access='direct', recl=imax*jmax*kmax*4)

c
  iflag = 1
1 read(10, end=2) HEAD
  read(10, err=999) GD
  do k = 1, kmax
    do j = 1, jmax
      do i = 1, imax
        if( (GD(i,j,k).lt.rmin).or.(GD(i,j,k).gt.rmax) ) then
          GD(i,j,k) = undef
        endif
      enddo
    enddo
  enddo
  write(51, rec=iflag) GD
  iflag = iflag + 1
  goto 1
2 continue

c
  stop
999 write(6,*) 'Error: DO NOT READ DATA ...'
end
```