

自転方向が気候に与える影響に関する研究：  
順回転地球と逆回転地球の気候区分図

岡山大学 理学部 地球科学科  
05501522 前村 有耶

2024/02/09

## 要旨

気候は概ね緯度によって決まっているが、同じ緯度帯でも大陸の東岸と西岸で気候は異なる。大陸の東岸と西岸の気候の違いをつくっているのは、恒常風の風向と考えられている。本研究では、恒常風の風向が気候に与える影響を調べるため、大気大循環モデルDCPAM5を用いた気候シミュレーションをおこなった。計算は、現在の地球と同じ自転方向である順回転と、自転の方向だけが逆向きの逆回転の2通りについておこなった。同じ初期条件で40年間の計算をおこない、最後の10年について解析した。順回転地球と逆回転地球を比較すると、大陸の多くの場所の東岸と西岸で気候の特徴が入れ替わっていた。恒常風の東西方向の風向は、順回転地球と逆回転地球でほぼ反対方向を向いており、大陸の東岸と西岸の気候の特徴は恒常風の風向によってつくられていることが示唆された。

# 目次

第1章	恒常風の風向と気候	2
第2章	モデルと実験設定	3
2.1	大気大循環モデル DCPAM5	3
2.2	実験設定	4
2.2.1	解像度	4
2.2.2	自転方向	4
2.2.3	暦・惑星に関する定数	4
2.2.4	植生	4
2.2.5	海面水温	4
2.2.6	海水分布	4
2.2.7	オゾン	5
2.2.8	初期条件	5
2.2.9	積分期間	5
第3章	気候シミュレーションの結果の解析	6
3.1	気候区分図	6
3.1.1	気候区分の判定	6
3.1.2	順回転地球と標準設定の比較	7
3.1.3	補正した気候区分図	8
3.1.4	順回転地球と逆回転地球	9
3.2	風	10
3.3	地表付近の気温	12
3.3.1	年平均	12
3.3.2	年較差	13
3.4	降水量	14
第4章	まとめ	15
付録		17
1.1	実験で使用した namelist	17
1.2	図を描く際に使用したスクリプト	18

## 第1章 恒常風の風向と気候

地球の気候は大まかには、緯度によって決まっており、同じ緯度帯では似たような気候になっている。しかし、同じ緯度帯であっても、大陸の東岸と西岸で気候は異なる。このような違いは、恒常風の風向によって作られていると考えられる。例えば、中緯度では偏西風が吹いていて、大陸の東岸は風上の陸、大陸西岸は風上の海の影響を受ける。これにより、大陸の東岸は風上の陸の影響を受けて年較差が大きく、大陸の西岸は風上の海の影響を受けて年較差が小さくなっていると考えられる。降水量も、風上が陸であるか海であるかによって異なると考えられる。

恒常風の風向は南北の気温傾度による気圧傾度力とコリオリ力が釣り合うように決まっているため、他の条件を変えずに、自転方向だけを逆向きにしたら、コリオリパラメタの符号が反転し、恒常風の東西方向の風向だけが反転すると考えられる。そのような逆回転する地球では、大陸の東岸と西岸の気候の特徴が入れ替わるはずである。本研究では、大気大循環モデルDCPAM5を用いて、自転方向を逆にした地球の気候について調べ、恒常風の風向が気候に与える影響について考察した。

## 第2章 モデルと実験設定

### 2.1 大気大循環モデルDCPAM5

本研究では、大気大循環モデルDCPAM5(高橋ほか, 2018)バージョン 20180304-2を使用して気候シミュレーションをおこなった。モデルの詳細は、DCPAM5のドキュメント([https://dennou-h.gfd-dennou.org/library/dcpam/dcpam5/dcpam5\\_latest/doc/](https://dennou-h.gfd-dennou.org/library/dcpam/dcpam5/dcpam5_latest/doc/))を参照されたい。本研究では、物理過程は表 2.1 を用いた。

表 2.1: 用いた物理過程

放射過程	Toon et al. (1989), Chou and Lee (1996), Chou et al. (2001)
大規模凝結	Manabe (1965)
積雲対流	Relaxed Arakawa-Schubert (Moorthi and Suarez, 1992)
乱流混合	Mellor and Yamada level2 (Mellor and Yamada, 1974, 1982)
陸面過程	バケツモデル Manabe (1969)

## 2.2 実験設定

実験の設定は主に現在の地球の気候シミュレーションに用いられる標準設定を用いたが、植生 (2.2.4) や海面水温 (2.2.5) などいくつかは標準設定から変更した。また、現在の地球と比較するため、順回転・逆回転とは別に、標準設定でも気候シミュレーションをおこなった。

### 2.2.1 解像度

水平解像度は T42(スペクトル法の三角切断で切断波数 42) とした。格子点は緯度方向に 64, 経度方向に 128 ずつ配置されており、格子点の間隔は約 2.8 度である。鉛直層の数は 26 層とした。各層の中心位置は Arakawa and Suarez (1983) に基づいて決定した。また、時間発展を計算する際の時間ステップは 720 秒とした。

### 2.2.2 自転方向

自転の方向は、順回転 (現在の地球と同じ) と逆回転 (現在の地球と方向だけ真逆) の 2 通りについて気候シミュレーションをおこなった。標準設定は、もちろん現在の地球と同じである。

### 2.2.3 暦・惑星に関する定数

解析を簡単にするため、1 年は 360 日 (30 日  $\times$  12 ヶ月), 離心率は 0.00 とした。標準設定の計算も、この簡略化した暦と軌道要素を使用した。

### 2.2.4 植生

植生の影響を受けないようにするため、順回転地球と逆回転地球の計算では、陸面を全て砂漠にした。標準設定の計算は、現在の地球で観測される植生に基づいて与えた。

### 2.2.5 海面水温

海洋は一様温度を持つ厚さ 60m の板海として扱い、海面水温をモデル内で計算した。標準設定は海面水温を計算せず、観測された海面水温を与えた。

### 2.2.6 海水分布

モデル内で計算される海面水温に依らず、現在の地球で観測される月毎の海水の分布を与えた。順回転地球と逆回転地球は板海を使用して海水温を計算しているが、モデルで計算される海水温が現在の地球のそれよりも低くなるため、計算

---

された海水温に基づいて海水を与えると、海水面積が現在の地球よりも大きくなる。そこでアイスアルベドフィードバックが働くと、より寒冷化した気候になる。現在の地球に近づけるため、海水分布は現在の地球で観測されるものを与えることにした。

### 2.2.7 オゾン

オゾンはなしとした。標準設定の計算も、オゾンはなしとした。

### 2.2.8 初期条件

大気は 280K の等温静止状態の大気を初期状態とした。また、地表面は 300K を初期状態とした。

### 2.2.9 積分期間

順回転地球、逆回転地球について、同じ初期条件で 40 年積分をおこない、定常状態になったと思われる最後の 10 年を解析に使用した。標準設定は、同じ初期条件で 10 年積分をおこない、定常状態になったと思われる最後の 3 年を解析に使用した。

## 第3章 気候シミュレーションの結果の解析

### 3.1 気候区分図

#### 3.1.1 気候区分の判定

図3.1に従って気候の区分をおこない、順回転地球、逆回転地球の気候区分図を作成した。この気候区分は基本的にはケッペンの気候区分(例えば、<https://geo.skygrass.net/note/clim/koppen.pdf>)に従っている。乾燥帯に関しては、降水の季節変化や年平均気温に依らず、年降水量300mm以下の地域をステップ、年降水量100mm以下の地域を砂漠とした。

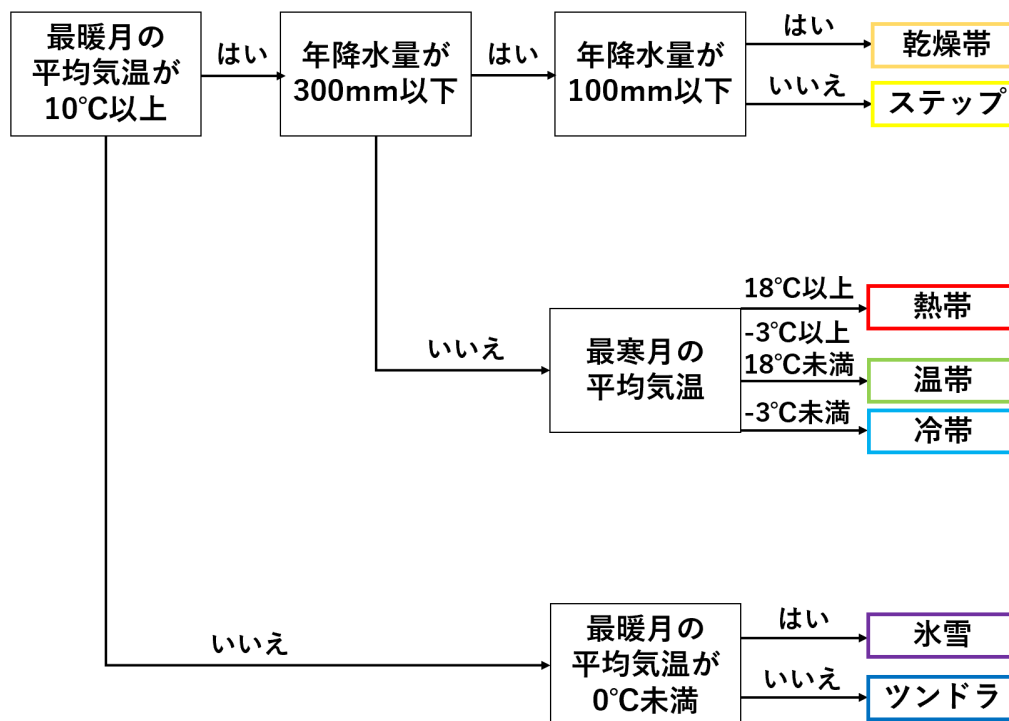


図 3.1: 気候区分の判定表



### 3.1.2 順回転地球と標準設定の比較

図 3.2 は気候シミュレーションの結果に基づいて作成した気候区分図である。標準設定は観測された海面水温を与えていることもあり、現実の地球の気候区分図に近いものが計算で再現されている。それに比べると、順回転地球は気温が低下、特に中高緯度で大きく低下し、冷帯が大きく縮小し、寒帯が拡大している。それにより、気候区分の配置が緯度方向に縮んでいる。これは、板海を用いたため、海の熱輸送がないことが原因だと考えられる。しかし、気候区分の配置は標準設定のものとはほぼ同じで、大陸東岸と西岸の気候の特徴も再現されている。

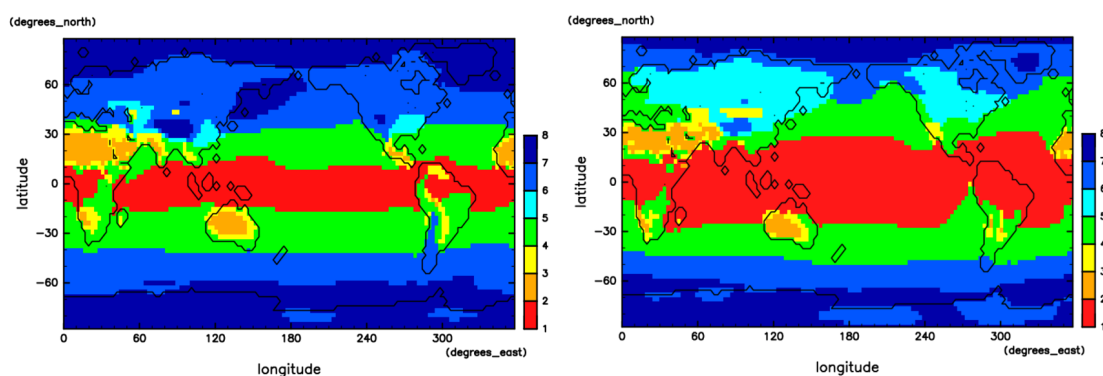


図 3.2: 気候区分図。縦軸は緯度、横軸は経度、黒線は海岸線を示す。赤：熱帯、橙：砂漠、黄：ステップ、緑：温帯、水：冷帯、青：ツンドラ、濃青：氷雪。(左) 順回転地球の計算結果。陸面は全て砂漠、海洋は板海で計算している。(右) 地球の気候シミュレーションで用いられる標準設定を用いた計算結果。

### 3.1.3 補正した気候区分図

順回転地球と標準設定を全球平均で比べると、順回転地球は大気最下層の気温は約  $6.33\text{ }^{\circ}\text{C}$  低く、年降水量は約 0.8 倍である (表 3.1).

表 3.1: 全球平均

	順回転地球	標準設定
地表付近の年平均気温	278.026 K	271.694 K
年降水量	1074 mm	853.3 mm

順回転地球の気候を現在の地球の気候と比較しやすくするため、気候の判定基準を修正して、順回転地球の気候区分の判定をやり直した (図 3.3). このやり直した (補正した) 気候区分では、順回転地球と標準設定の間にある年平均気温と年降水量の差の分だけ、気候区分の判定基準に用いる気温と年降水量を値をずらした. 具体的には、基準となる値を、気温は  $6.33\text{ }^{\circ}\text{C}$  下げ、年降水量は 0.794 倍した. 気候区分の判定基準を補正したことで、順回転地球の気候区分図は現在の地球の気候区分図にかなり近づいた (図 3.3). 以下では、補正した気候区分図を使用して順回転地球と逆回転地球を比較する.

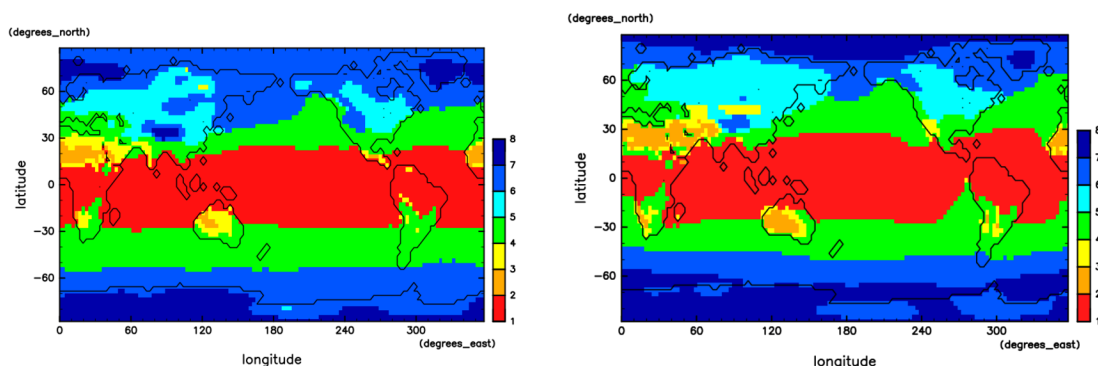


図 3.3: 気候区分図. 縦軸は緯度, 横軸は経度, 黒線は海岸線を示す. 赤: 熱帯, 橙: 砂漠, 黄: ステップ, 緑: 温帯, 水: 冷帯, 青: ツンドラ, 濃青: 氷雪. (左) 補正した順回転地球 (右) 標準設定

### 3.1.4 順回転地球と逆回転地球

図 3.4 は、順回転地球と逆回転地球の気候区分図 (補正あり) である。どちらの気候区分図も大まかには、赤道付近は熱帯、そこから高緯度側に向かって乾燥帯、温帯、冷帯、寒帯となっている。大陸の東岸と西岸を比べると、順回転地球は現在の地球と同じ特徴を示すのに対し、逆回転地球はいくつかの地域で大陸東岸と西岸の特徴が現在の地球と逆になっている。具体的には、順回転地球でアフリカ大陸南部やユーラシア大陸の南アジアの大陸西岸側にあった乾燥帯が、逆回転地球では大陸東岸側に移動している。また、順回転地球で東アジアや北アメリカ大陸の大陸東岸側にみられた冷帯は、逆回転地球では温帯に変化しており、順回転地球でヨーロッパや北アメリカ大陸の大陸西岸側にみられた温帯は、逆回転地球では冷帯に変化している。

このような気候区分の変化は、主に大陸の東岸と西岸でみられ、内陸部に大きな変化はみられない。ただし、内陸部でも、アフリカ北部のサハラ砂漠やオーストラリアの内陸部では、降水量が増加し乾燥帯が温帯に変化した。

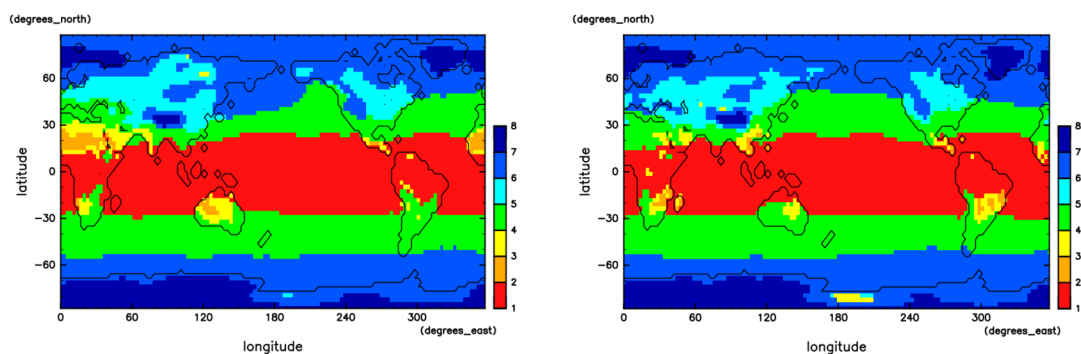


図 3.4: 気候区分図。縦軸は緯度、横軸は経度、黒線は海岸線を示す。赤：熱帯、橙：砂漠、黄：ステップ、緑：温帯、水：冷帯、青：ツンドラ、濃青：冰雪。(左) 補正した順回転地球 (右) 補正した逆回転地球

## 3.2 風

図 3.5 は年平均した東西風を経度平均した子午面断面の風速の分布である。順回転地球と標準設定を比較すると、風速の分布は似たものになっている。

順回転地球の地表付近の分布みると、大まかには低緯度では東風、中緯度では西風、高緯度では東風となっており、これはそれぞれ貿易風、偏西風、極偏東風に対応している。一方で、逆回転地球では風向の東西が入れ替わっていて、自転方向を逆にすると、恒常風の風向が東西で反転することがわかる。

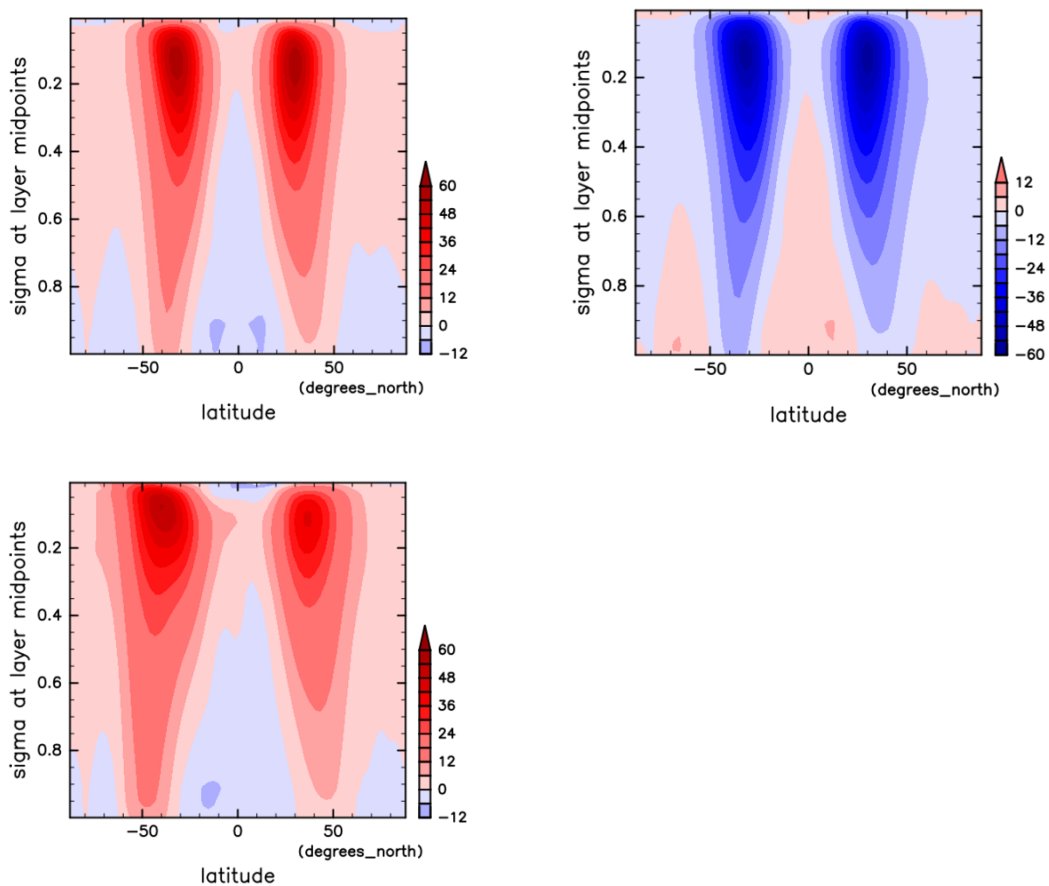


図 3.5: 年平均、経度平均した東西風の子午面断面図。縦軸は $\sigma$ で鉛直方向、横軸は緯度を示す。(左上) 順回転地球 (右上) 逆回転地球 (左下) 標準設定

図 3.6 は大気最下層の風向の年平均をベクトルで表したものである。順回転地球と標準設定は、多くの場所で同じ風向となっている。

一方、順回転地球と逆回転地球では、多くの場所で東西の風向が反転している。その結果、順回転地球で風上が陸であった沿岸部は、逆回転地球では風上が海に、順回転地球で風上が海であった沿岸部は、逆回転地球では風上が陸に変化している。

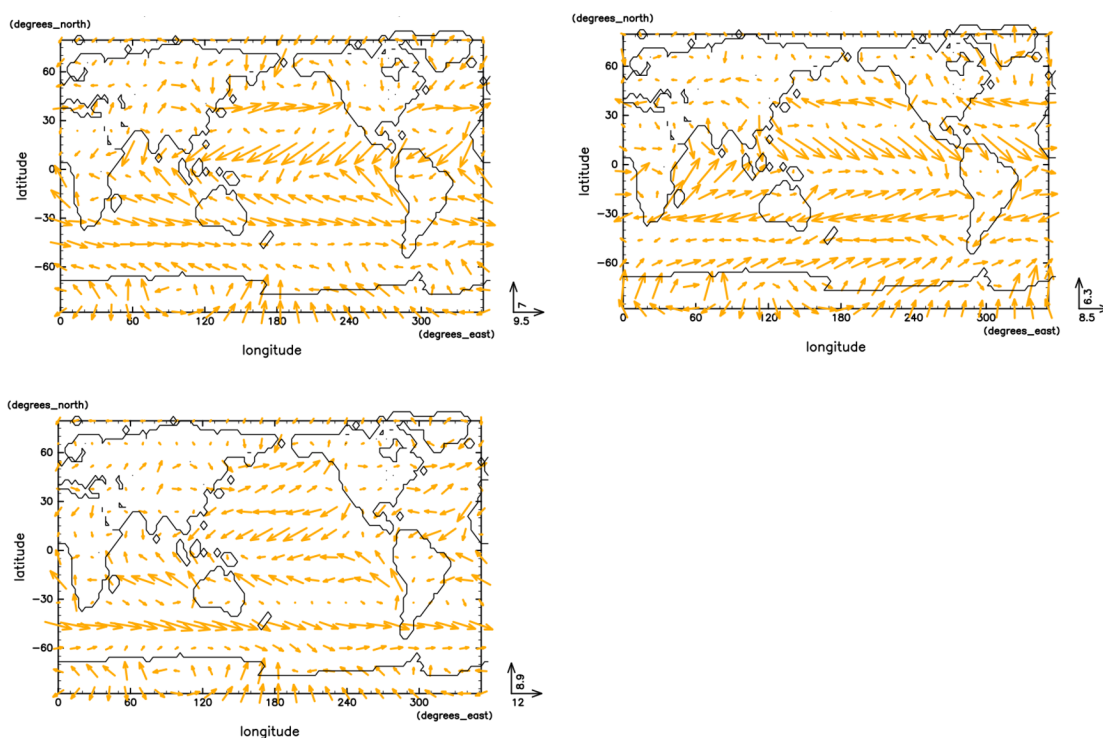


図 3.6: 大気最下層の風のベクトル図。縦軸は緯度，横軸は経度，黒線は海岸線を示す。(左上) 順回転地球 (右上) 逆回転地球 (左下) 標準設定

## 3.3 地表付近の気温

### 3.3.1 年平均

図 3.7 は大気最下層の気温の年平均である。順回転地球と標準設定を比較すると、全体的に気温が下がってはいるが、分布は似たものになっている。

順回転地球の北半球の海洋の等温線を見ると、大陸西岸に向かうにつれて間隔が広がっており、大陸西岸は大陸東岸と比べて気温の南北の温度勾配が小さい。一方で、逆回転地球の北半球の海洋の等温線を見ると、大陸東岸に向かうにつれて間隔が広がっていて、気温の南北勾配の特徴が大陸東岸と西岸で逆転している。これは、自転方向を逆にしたことで恒常風の風向が逆転したためだと考えられる。

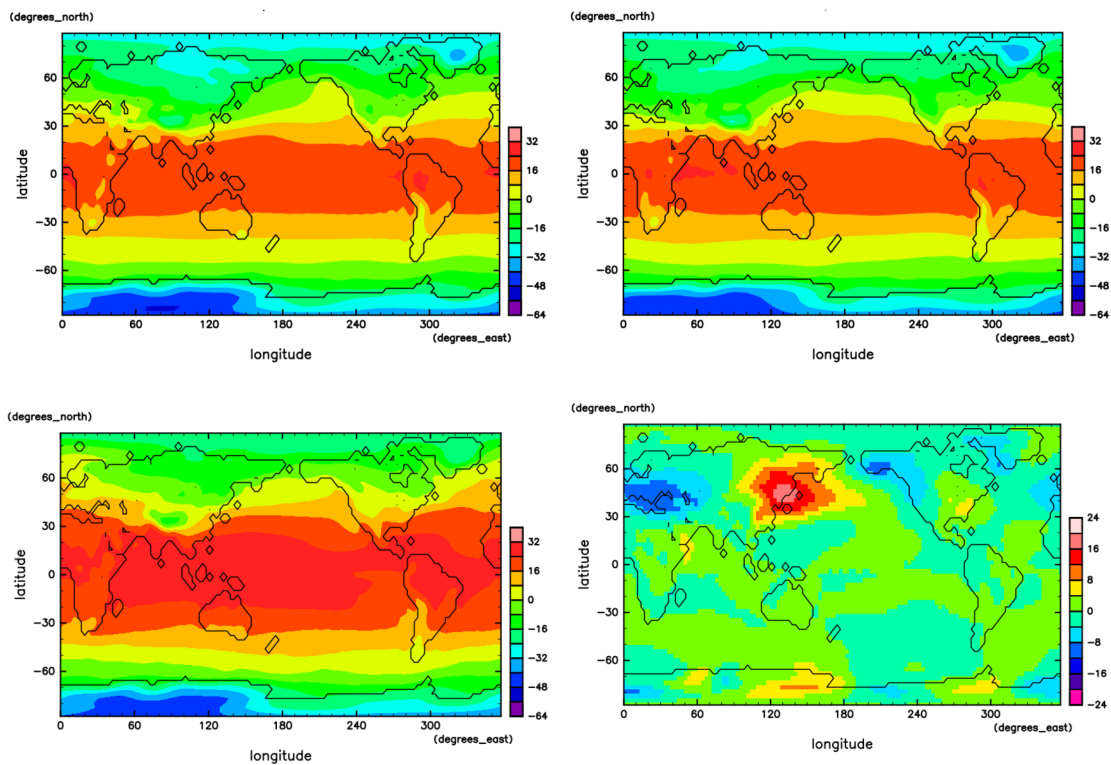


図 3.7: 大気最下層の気温の分布図。縦軸は緯度，横軸は経度，黒線は海岸線を示す。(左上) 順回転地球 (右上) 逆回転地球 (左下) 標準設定 (右下) 逆回転地球から順回転地球を引いた差

### 3.3.2 年較差

図 3.8 は気温の年較差の分布である。順回転地球と標準設定を比較すると、年較差の分布は似たものになっている。

順回転地球は、ユーラシア大陸の東岸で大きく、アメリカ大陸の西岸で小さい。一方、逆回転地球ではユーラシア大陸の東岸で小さく、アメリカ大陸の西岸で大きくなっており、大陸の東岸と西岸で分布が入れ替わっている。これは、自転方向を逆にすることで恒常風の風向が逆転し、大陸東岸の風上は陸、大陸西岸の風上は海であったのが、大陸東岸の風上が海、大陸西岸の風上が陸になったからだと考えられる。

一方で、南半球ではこのような変化はみられず、全体的に年較差も小さくなっている。これは、中緯度帯に大陸がないことや大陸が小さく、海の割合が大きいことが理由だと考えられる。

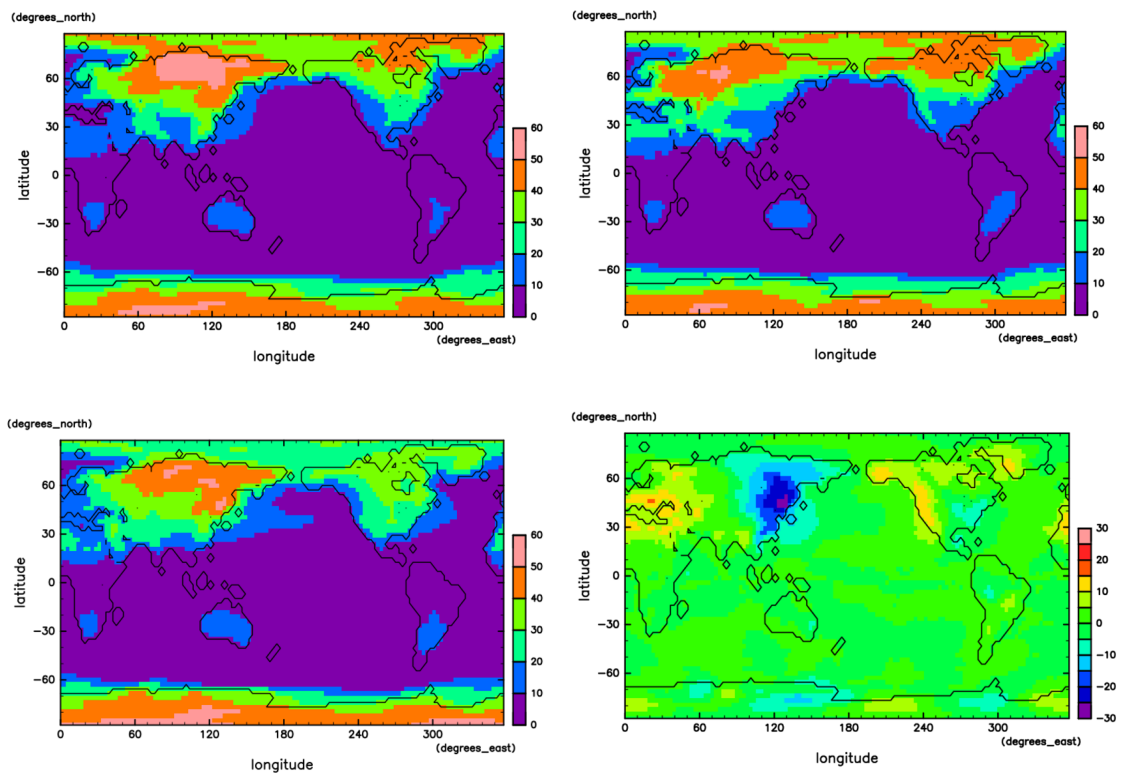


図 3.8: 大気最下層の気温の年較差の分布図。縦軸は緯度、横軸は経度、黒線は海岸線を示す。(左上) 順回転地球 (右上) 逆回転地球 (左下) 標準設定 (右下) 逆回転地球から順回転地球を引いた差



### 3.4 降水量

図 3.9 は年降水量の分布である。順回転地球と標準設定を比較すると、順回転地球では赤道付近の熱帯収束帯 (ITCZ) の範囲が狭まり、降水量が多くなっている。これは、板海を用いたことにより、赤道付近の海面水温の分布が現実の地球と異なるものになったためと考えられる。また、ITCZ 以外の場所では、降水量が全体的に減少しているが、分布は似たものになっている。

順回転地球と逆回転地球の差は、熱帯域で最も目立っている。これは、ITCZ の位置が順回転地球と逆回転地球で少し変化したことによる。順回転地球の ITCZ が少し湾曲しているのに対し、逆回転地球の ITCZ はほぼ真っ直ぐになっている。ITCZ 以外では、太平洋を挟んだユーラシア大陸の東岸と北アメリカ大陸西岸で、高緯度ではユーラシア大陸東岸で増加し、北アメリカ大陸西岸で減少している。また、低緯度ではユーラシア大陸東岸で減少し、北アメリカ大陸西岸で増加している。他の大陸の東岸と西岸でも同様の変化がみられ、大陸の東岸と西岸で降水量の分布が入れ替わっている。

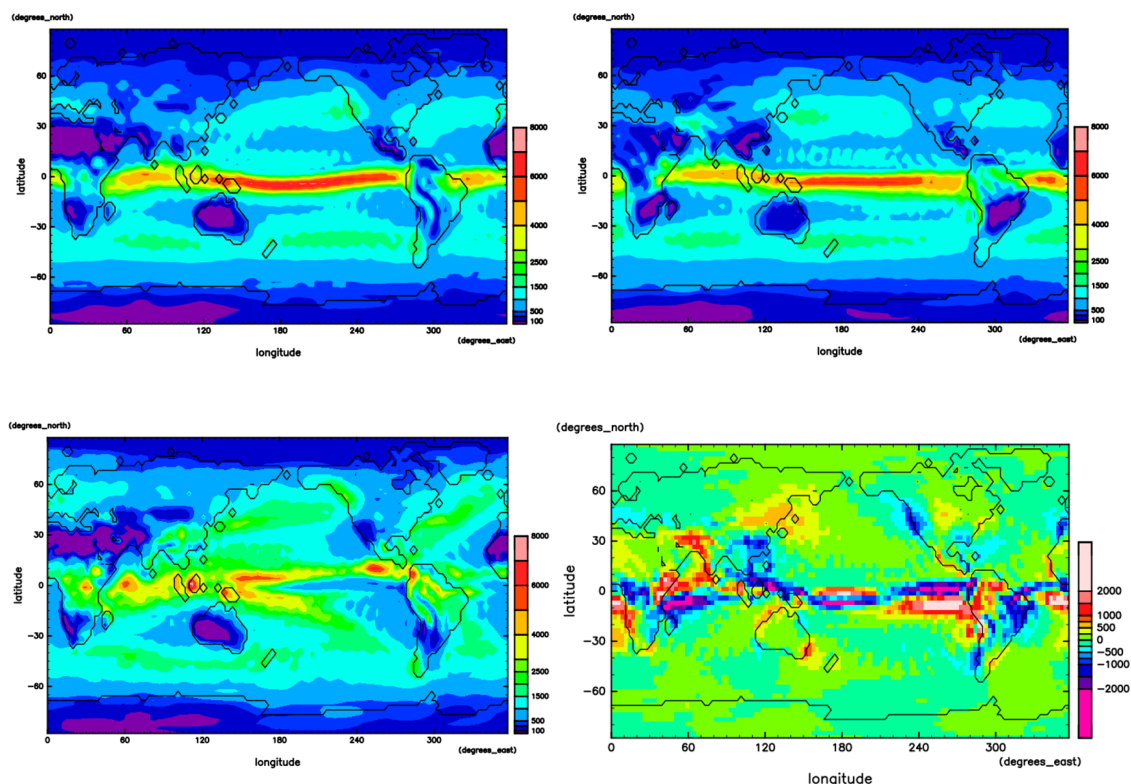


図 3.9: 年降水量の分布図. 単位はすべて mm/year . 縦軸は緯度, 横軸は経度, 黒線は海岸線を示す. (左上) 順回転地球 (右上) 逆回転地球 (左下) 標準設定 (右下) 逆回転地球から順回転地球を引いた差



## 第4章 まとめ

大気大循環モデル DCPAM5 を用いて、自転方向を逆にした仮想地球の気候シミュレーションをおこなった。大陸の東岸と西岸は同じ緯度帯であっても気候が異なるが、自転方向を逆にすると大陸の東岸と西岸の気候の特徴が入れ替わった。自転方向を逆にすると、恒常風の風向が東西で逆転していることから、恒常風の風向が大陸の東岸と西岸の気候の特徴をつくる原因と推察された。また、大陸の東岸と西岸以外でも、一部の地域に変化がみられた。具体的には、北アフリカやオーストラリアの内陸部といった海洋の影響をあまり受けないと思われる地域の降水量の増加や、ITCZの形状の変化である。これらの変化が生じた原因を解明するためには、さらなる解析と考察が必要である。

## 謝辞

本研究を進めるにあたって、ご指導いただきました主指導教官であるはしもとじょーじ教授には心より深く感謝申し上げます。また、同研究室の皆様には本研究に関してたくさんの助言や励ましの言葉をいただきました。ありがとうございました。

# 付録

## 1.1 実験で使用した namelist

本研究の実験の際に変更した DCPAM5(20180304-2 版) の namelist 変数のリストを載せる.

表 1.1: constants\_nml の namelist 変数

変数名	意味	変更後
Omega	回転角速度 [s-1]	順回転地球:7.292106d-5, 逆回転地球:-7.292106d-5

表 1.2: timeset\_nml の namelist 変数

変数名	意味	変更後
cal_type	暦の種類	'user_defined'
month_in_year	1年の月の数	12
day_in_month	1ヶ月の日数	30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30
hour_in_day	1日の時間数	24
min_in_hour	1時間の分数	60
sec_in_min	1分の秒数	60.0d0

表 1.3: rad\_short\_income\_nml の namelist 変数

変数名	意味	変更後
FlagSpecifySolDay	太陽の動きに基づく時刻の設定フラグ	逆回転地球:.true.
SolDayValue	太陽の動きに基づく時刻で測った1日	逆回転地球:-1.0d0
SolDayUnit	太陽の動きに基づく時刻で測った1日の単位	逆回転地球:'day'
Eccentricity	離心率	0.0d0

## 1.2 図を描く際に使用したスクリプト

### 気候区分図

順回転地球における気候区分図を描く際に用いたスクリプトを載せる。

Listing 1.1: 気候区分図

```
1 #! /usr/bin/ruby
2 require "numru/ggraph"
3 include NumRu
4
5 gptemp = GPhys::NetCDF_IO.open('Temp_E40.nc', 'Temp')
6 gplon = GPhys::NetCDF_IO.open('Temp_E40.nc', 'lon')
7 gplat = GPhys::NetCDF_IO.open('Temp_E40.nc', 'lat')
8 nlon = gplon.shape[0]
9 nlat = gplat.shape[0]
10 gpprcp = GPhys::NetCDF_IO.open('PRCP_E40.nc', 'PRCP')
11 gpsfc = GPhys::IO.open('sp_for_Earth_T042_Desert.nc', 'sfcindex')
12
13 ys = 31
14 ye = 40
15
16 yyy = 360
17
18 jan = 30
19 feb = 60
20 mar = 90
21 apr = 120
22 may = 150
23 jun = 180
24 jul = 210
25 aug = 240
26 sep = 270
27 oct = 300
28 nov = 330
29 dec = 360
30
31 months = [0, jan, feb, mar, apr, may, jun, jul, aug, sep, oct, nov]
32 monthe = [jan, feb, mar, apr, may, jun, jul, aug, sep, oct, nov, dec]
33 number = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]
34
35 lons = gplon.val.to_a
36 lats = gplat.val.to_a
37
38 r100 = 100.0/360.0
39 r300 = 300.0/360.0
```

```
40
41 i = NArray.int(nlon).indgen!(0,1)
42 j = NArray.int(nlat).indgen!(0,1)
43
44 climate = VArray.new( NArray.sfloat(nlon,nlat), {"long_name"=>"
    Climate"}, "Climate" )
45
46 gptemon = []
47 lons.zip(i).each do |lon,i|
48   lats.zip(j).each do |lat,j|
49     number.zip(months,monthe).each do |k,ms,me|
50       gptemon[k] = gptemp.cut('lat'=>lat,'lon'=>lon,'sig
        '=>0.998,'time'=>yyy*(ys-1)+ms+1..yyy*(ys-1)+me).mean
        ('time').val
51       for l in ys+1..ye
52         gptemon[k] = gptemon[k]+gptemp.cut('lat'=>lat,'lon'=>
            lon,'sig'=>0.998,'time'=>yyy*(l-1)+ms+1..yyy*(l-1)+
            me).mean('time').val
53       end
54       gptemon[k] = gptemon[k]/(ye-ys+1)
55     end
56     gpprcpbar = gpprcp.cut('lat'=>lat,'lon'=>lon,'time'=>yyy*(ys
        -1)+1..yyy*ye).mean('time').val
57     gpprcpbar = gpprcpbar*60*60*24
58
59     if ( gptemon.max < 273.15 ) then
60       if ( gptemon.max <= 273.15 ) then
61         climate[i,j] = 7.1
62       else
63         climate[i,j] = 6.1
64       end
65     else
66       if ( gpprcpbar < r300 ) then
67         if ( gpprcpbar < r100 ) then
68           climate[i,j] = 2.1
69         else
70           climate[i,j] = 3.1
71         end
72       else
73         if ( gptemon.min >= 291.15 ) then
74           climate[i,j] = 1.1
75         elsif ( gptemon.min >= 270.15 ) then
76           climate[i,j] = 4.1
77         else
78           climate[i,j] = 5.1
79         end
```

---

```
80     end
81     end
82   end
83 end
84
85 gplons = gptemp.axis('lon')
86 gplats = gptemp.axis('lat')
87
88 gpclimate = GPhys.new(Grid.new(gplons,gplats),climate)
89
90 DCL.gropn(1)
91 DCL.sgpset('isub', 96)
92 DCL.sgpset('lfull',true)
93 DCL.uzfact(0.6)
94 DCL.udpset('LMSG',false)
95 GGraph.set_fig('viewport'=>[0.15,0.85,0.15,0.6])
96 GGraph.set_axes('xlabelint'=>60,'ylabelint'=>30)
97 GGraph.tone(gpclimate,true,'tonc'=>true,'levels
    =>[1,2,3,4,5,6,7,8],'patterns
    =>[86999,75999,70999,55999,40999,30999,20999])
98 GGraph.set_contour_levels('levels'=>[0,0.0001],'index'=>[2,2])
99 GGraph.contour(gpsfc,false)
100 GGraph.color_bar
101 DCL.grcls
```

---

## 年平均・経度平均した東西風の子午面断面図

順回転地球における年平均・経度平均した東西風の子午面断面図を描く際に用いたスクリプトを載せる。

Listing 1.2: 年平均・経度平均した東西風の子午面断面図

```
1 #!/usr/bin/ruby
2 require "numru/ggraph"
3 include NumRu
4
5 gp1 = GPhys::IO.open( "U_E40.nc", "U" ).cut('time
   =>10800..12600)
6 gp1 = gp1.mean('lon').mean('time')
7 gp2 = GPhys::IO.open( "U_E40.nc", "U" ).cut('time
   =>12601..14400)
8 gp2 = gp2.mean('lon').mean('time')
9 gp3 = (gp1 + gp2)/2
10
11 DCL.gropn(1)
12 DCL.sgscmn(65)
13 GGraph.tone( gp3,true,'levels
   =>[-12,-6,0,6,12,18,24,30,36,42,48,54,60], 'patterns
   =>[47999,51999,59999,63999,67999,67999,71999,75999,79999,83999,
14 87999,91999,95999], "annot"=>false)
15 GGraph.color_bar
16 DCL.grcls
```

## 大気最下層の風のベクトル図

順回転地球における大気最下層の風のベクトル図を描く際に用いたスクリプトを載せる。

Listing 1.3: 大気大気最下層の風のベクトル図

```
1 #!/usr/bin/ruby
2 require "numru/ggraph"
3 include NumRu
4
5 gp1 = GPhys::IO.open( "U_E40.nc", "U" ).cut('sig'=>0.998,'time
   =>10800..12600)
6 gp1 = gp1.mean('time')
7 gp2 = GPhys::IO.open( "U_E40.nc", "U" ).cut('sig'=>0.998,'time
   =>12601..14400)
8 gp2 = gp2.mean('time')
9 gp3 = (gp1 + gp2)/2
10 gp4 = GPhys::IO.open( "V_E40.nc", "V" ).cut('sig'=>0.998,'time
   =>10800..12600)
11 gp4 = gp4.mean('time')
12 gp5 = GPhys::IO.open( "V_E40.nc", "V" ).cut('sig'=>0.998,'time
   =>12601..14400)
13 gp5 = gp5.mean('time')
14 gp6 = (gp4 + gp5)/2
15 gpsfc = GPhys::IO.open('sp_for_Earth_T042_Desert.nc', 'sfcindex')
16
17 DCL.gropn(1)
18
19 DCL.sgpset('isub', 96)
20 DCL.sgpset('lfull', true)
21 DCL.uzfact(0.6)
22 DCL.udpset('LMSG', false)
23 GGraph.set_fig('viewport'=>[0.15,0.85,0.15,0.6])
24 GGraph.set_axes('xlabelint'=>60,'ylabelint'=>30)
25 GGraph.vector( gp3,gp6,true,'index'=>755,"xintv"=>5, "yintv
   "=>5,'flow_vect'=>true,'factor'=>2.0,'unit_vect'=>true,"annot
   "=>false)
26 GGraph.set_contour_levels('levels'=>[0,0.0001], 'index'=>[2,2])
27 GGraph.contour(gpsfc,false)
28 DCL.grcls
```



## 年平均気温

順回転地球における大気最下層の気温の年平均の図を描く際に用いたスクリプトを載せる。

Listing 1.4: 気温の年平均の図

```
1 #!/usr/bin/ruby
2 require "numru/ggraph"
3 include NumRu
4
5 gp1 = GPhys::IO.open( "Temp_E40.nc", "Temp" ).cut('time
   =>10800..12600)
6 gp1 = gp1.mean('time')-273.15
7 gp2 = GPhys::IO.open( "Temp_E40.nc", "Temp" ).cut('time
   =>12601..14400)
8 gp2 = gp2.mean('time')-273.15
9 gp3 = (gp1 + gp2)/2
10 gp3.units = Units['degree Celsius']
11 gpsfc = GPhys::IO.open('sp_for_Earth_T042_Desert.nc', 'sfcindex')
12
13 DCL.gropn(1)
14
15 DCL.sgpset('isub', 96)
16 DCL.sgpset('lfull', true)
17 DCL.uzfact(0.6)
18 DCL.udpset('LMSG', false)
19 GGraph.set_fig('viewport'=>[0.15,0.85,0.15,0.6])
20 GGraph.set_axes('xlabelint'=>60,'ylabelint'=>30)
21 GGraph.tone( gp3, true, 'lev
   =>[-64,-56,-48,-40,-32,-24,-16,-8,0,8,16,24,32,40], "annot
   "=>false)
22 GGraph.set_contour_levels('levels'=>[0,0.0001], 'index'=>[2,2])
23 GGraph.contour(gpsfc, false)
24 GGraph.color_bar
25 DCL.grcls
```

## 気温の年平均の差

逆回転地球と順回転地球の大気最下層の気温の年平均の差の図を描く際に用いたスクリプトを載せる。

Listing 1.5: 気温の年平均の差の図

```
1  #! /usr/bin/ruby
2  require "numru/ggraph"
3  include NumRu
4
5  gptemp1 = GPhys::NetCDF_IO.open('Temp_E40.nc', 'Temp')
6  gptemp2 = GPhys::NetCDF_IO.open('Temp_h40.nc', 'Temp')
7  gplon = GPhys::NetCDF_IO.open('Temp_E40.nc', 'lon')
8  gplat = GPhys::NetCDF_IO.open('Temp_E40.nc', 'lat')
9  nlon = gplon.shape[0]
10 nlat = gplat.shape[0]
11 gpsfc = GPhys::IO.open('sp_for_Earth_T042_Desert.nc', 'sfcindex')
12
13 lons = gplon.val.to_a
14 lats = gplat.val.to_a
15
16 i = NArray.int(nlon).indgen!(0,1)
17 j = NArray.int(nlat).indgen!(0,1)
18
19 difference = VArray.new( NArray.sfloat(nlon,nlat), {"long_name
    "=>"Temperature difference"}, "Temperature difference" )
20
21 lons.zip(i).each do |lon,i|
22   lats.zip(j).each do |lat,j|
23     gptemdif = (gptemp2.cut('lat'=>lat, 'lon'=>lon, 'sig'=>0.998, '
        time'=>10800..14400).mean('time').val) - (gptemp1.cut('
        lat'=>lat, 'lon'=>lon, 'sig'=>0.998, 'time'=>10800..14400).
        mean('time').val)
24     difference[i,j] = gptemdif
25   end
26 end
27
28 gplons = gptemp1.axis('lon')
29 gplats = gptemp1.axis('lat')
30
31 gptemdiff = GPhys.new(Grid.new(gplons,gplats),difference)
32
33 DCL.gropn(1)
34 DCL.sgpset('isub', 96)
35 DCL.sgpset('lfull', true)
36 DCL.uzfact(0.6)
```

```
37 DCL.udpset('LMSG',false)
38 GGraph.set_fig('viewport'=>[0.15,0.85,0.15,0.6])
39 GGraph.set_axes('xlabelint'=>60,'ylabelint'=>30)
40 GGraph.tone(gptemdiff,true,'tonc'=>true,'lev
    '=>[-24,-20,-16,-12,-8,-4,0,4,8,12,16,20,24],'patterns
    '=>[10999,17999,24999,31999,38999,45999,62999,71999,78999,85999,
41 92999,98999])
42 GGraph.set_contour_levels('levels'=>[0,0.0001],'index'=>[2,2])
43 GGraph.contour(gpsfc,false)
44 GGraph.color_bar
45 DCL.grcls
```

---

## 気温の年較差

順回転地球における気温の年較差の図を描く際に用いたスクリプトを載せる。

Listing 1.6: 気温の年較差の図

```
1  #! /usr/bin/ruby
2  require "numru/ggraph"
3  include NumRu
4
5  gptemp = GPhys::NetCDF_IO.open('Temp_E40.nc', 'Temp')
6  gplon = GPhys::NetCDF_IO.open('Temp_E40.nc', 'lon')
7  gplat = GPhys::NetCDF_IO.open('Temp_E40.nc', 'lat')
8  nlon = gplon.shape[0]
9  nlat = gplat.shape[0]
10 gpsfc = GPhys::IO.open('sp_for_Earth_T042_Desert.nc', 'sfcindex')
11
12 ys = 31
13 ye = 40
14
15 yyy = 360
16
17 jan = 30
18 feb = 60
19 mar = 90
20 apr = 120
21 may = 150
22 jun = 180
23 jul = 210
24 aug = 240
25 sep = 270
26 oct = 300
27 nov = 330
28 dec = 360
29
30 months = [0, jan, feb, mar, apr, may, jun, jul, aug, sep, oct, nov]
31 monthe = [jan, feb, mar, apr, may, jun, jul, aug, sep, oct, nov, dec]
32 number = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]
33
34 lons = gplon.val.to_a
35 lats = gplat.val.to_a
36
37 i = NArray.int(nlon).indgen!(0, 1)
38 j = NArray.int(nlat).indgen!(0, 1)
39
40 yeartempdifference = VArray.new( NArray.sfloat(nlon, nlat), {"
    long_name"=>"YearTempretureRange"}, "YearTempretureRange" )
```

```
41
42 gptemon = []
43 lons.zip(i).each do |lon,i|
44   lats.zip(j).each do |lat,j|
45     number.zip(months,monthe).each do |k,ms,me|
46       gptemon[k] = gptemp.cut('lat'=>lat,'lon'=>lon,'sig
         '=>0.998,'time'=>yyy*(ys-1)+ms+1..yyy*(ys-1)+me).mean
         ('time').val
47       for l in ys+1..ye
48         gptemon[k] = gptemon[k]+gptemp.cut('lat'=>lat,'lon'=>
           lon,'sig'=>0.998,'time'=>yyy*(l-1)+ms+1..yyy*(l-1)+
           me).mean('time').val
49       end
50       gptemon[k] = gptemon[k]/(ye-ys+1)
51     end
52     yeartempdifference[i,j] = (gptemon.max) - (gptemon.min)
53   end
54 end
55
56 gplons = gptemp.axis('lon')
57 gplats = gptemp.axis('lat')
58
59 gpyeartempdifference = GPhys.new(Grid.new(gplons,gplats),
   yeartempdifference)
60
61 DCL.gropn(1)
62 DCL.sgpset('isub', 96)
63 DCL.sgpset('lfull',true)
64 DCL.uzfact(0.6)
65 DCL.udpset('LMSG',false)
66 GGraph.set_fig('viewport'=>[0.15,0.85,0.15,0.6])
67 GGraph.set_axes('xlabelint'=>60,'ylabelint'=>30)
68 GGraph.tone(gpyeartempdifference,true,'tonc'=>true,'lev
   '=>[0,10,20,30,40,50,60])
69 GGraph.set_contour_levels('levels'=>[0,0.0001], 'index'=>[2,2])
70 GGraph.contour(gpsfc,false)
71 GGraph.color_bar
72 DCL.grcls
```

## 気温の年較差の差

逆回転地球と順回転地球の気温の年較差の差の図を描く際に用いたスクリプトを載せる。

Listing 1.7: 気温の年較差の差の図

```
1 #! /usr/bin/ruby
2 require "numru/ggraph"
3 include NumRu
4
5 gptemp1 = GPhys::NetCDF_IO.open('Temp_h40.nc', 'Temp')
6 gptemp2 = GPhys::NetCDF_IO.open('Temp_E40.nc', 'Temp')
7 gplon = GPhys::NetCDF_IO.open('Temp_h40.nc', 'lon')
8 gplat = GPhys::NetCDF_IO.open('Temp_h40.nc', 'lat')
9 nlon = gplon.shape[0]
10 nlat = gplat.shape[0]
11 gpsfc = GPhys::IO.open('sp_for_Earth_T042_Desert.nc', 'sfcindex')
12
13 ys = 31
14 ye = 40
15
16 yyy = 360
17
18 jan = 30
19 feb = 60
20 mar = 90
21 apr = 120
22 may = 150
23 jun = 180
24 jul = 210
25 aug = 240
26 sep = 270
27 oct = 300
28 nov = 330
29 dec = 360
30
31 months = [0, jan, feb, mar, apr, may, jun, jul, aug, sep, oct, nov]
32 monthe = [jan, feb, mar, apr, may, jun, jul, aug, sep, oct, nov, dec]
33 number = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]
34
35 lons = gplon.val.to_a
36 lats = gplat.val.to_a
37
38 i = NArray.int(nlon).indgen!(0, 1)
39 j = NArray.int(nlat).indgen!(0, 1)
40
```

```

41 yeartempdifference = VArray.new( NArray.sfloat(nlon,nlat), {"
    long_name"=>"TempretureRangeDifference"}, "
    TempretureRangeDifference" )
42
43 gptemon1 = []
44 gptemon2 = []
45
46 lons.zip(i).each do |lon,i|
47   lats.zip(j).each do |lat,j|
48     number.zip(months,monthe).each do |k,ms,me|
49       gptemon1[k] = gptemp1.cut('lat'=>lat,'lon'=>lon,'sig
        '=>0.998,'time'=>yyy*(ys-1)+ms+1..yyy*(ys-1)+me).mean
        ('time').val
50       gptemon2[k] = gptemp2.cut('lat'=>lat,'lon'=>lon,'sig
        '=>0.998,'time'=>yyy*(ys-1)+ms+1..yyy*(ys-1)+me).mean
        ('time').val
51       for l in ys+1..ye
52         gptemon1[k] = gptemon1[k]+gptemp1.cut('lat'=>lat,'lon
            '=>lon,'sig'=>0.998,'time'=>yyy*(l-1)+ms+1..yyy*(l
            -1)+me).mean('time').val
53         gptemon2[k] = gptemon2[k]+gptemp2.cut('lat'=>lat,'lon
            '=>lon,'sig'=>0.998,'time'=>yyy*(l-1)+ms+1..yyy*(l
            -1)+me).mean('time').val
54       end
55       gptemon1[k] = gptemon1[k]/(ye-ys+1)
56       gptemon2[k] = gptemon2[k]/(ye-ys+1)
57     end
58     yeartempdifference[i,j] =((gptemon1.max) - (gptemon1.min)) -
        ((gptemon2.max) - (gptemon2.min))
59   end
60 end
61
62 gplons = gptemp1.axis('lon')
63 gplats = gptemp1.axis('lat')
64
65 gpyeartempdifference = GPhys.new(Grid.new(gplons,gplats),
    yeartempdifference)
66
67 DCL.gropn(1)
68 DCL.sgpset('isub', 96)
69 DCL.sgpset('lfull',true)
70 DCL.uzfact(0.6)
71 DCL.udpset('LMSG',false)
72 GGraph.set_fig('viewport'=>[0.15,0.85,0.15,0.6])
73 GGraph.set_axes('xlabelint'=>60,'ylabelint'=>30)
74 GGraph.tone(gpyeartempdifference,true,'tonc'=>true,'lev

```

```
    '=>[-30,-25,-20,-15,-10,-5,0,5,10,15,20,25,30])  
75 GGraph.set_contour_levels('levels'=>[0,0.0001], 'index'=>[2,2])  
76 GGraph.contour(gpsfc,false)  
77 GGraph.color_bar  
78 DCL.grcls
```

---



## 降水量

順回転地球における年降水量の図を描く際に用いたスクリプトを載せる。

Listing 1.8: 年降水量の図

```
1 #! /usr/bin/ruby
2 require "numru/ggraph"
3 include NumRu
4 gp1 = GPhys::IO.open('PRCP_E40.nc', 'PRCP').cut('time
   =>10800..12600)
5 gp1 = gp1.mean('time')*60*60*24*360
6 gp2 = GPhys::IO.open('PRCP_E40.nc', 'PRCP').cut('time
   =>12601..14400)
7 gp2 = gp2.mean('time')*60*60*24*360
8 gp3 = (gp1 + gp2)/2
9
10 gpsfc = GPhys::IO.open('sp_for_Earth_T042_Desert.nc', 'sfcindex')
11
12 DCL.gropn(1)
13
14 DCL.sgpset('isub', 96)
15 DCL.sgpset('lfull', true)
16 DCL.uzfact(0.6)
17 DCL.udpset('LMSG', false)
18 GGraph.set_fig('viewport'=>[0.15,0.85,0.15,0.6])
19 GGraph.set_axes('xlabelint'=>60, 'ylabelint'=>30)
20 GGraph.tone( gp3, true, 'lev
   =>[0,100,300,500,1000,1500,2000,2500,3000,4000,5000,6000,7000,8000], "
   annot"=>false)
21 GGraph.set_contour_levels('levels'=>[0,0.0001], 'index'=>[2,2])
22 GGraph.contour(gpsfc, false)
23 GGraph.color_bar
24 DCL.grcls
```

## 降水量の差

逆回転地球と順回転地球の年降水量の差の図を描く際に用いたスクリプトを載せる。

Listing 1.9: 年降水量の差の図

```
1  #! /usr/bin/ruby
2  require "numru/ggraph"
3  include NumRu
4
5  gpprcp1 = GPhys::NetCDF_IO.open('PRCP_E40.nc', 'PRCP')
6  gpprcp2 = GPhys::NetCDF_IO.open('PRCP_h40.nc', 'PRCP')
7  gplon = GPhys::NetCDF_IO.open('PRCP_E40.nc', 'lon')
8  gplat = GPhys::NetCDF_IO.open('PRCP_E40.nc', 'lat')
9  nlon = gplon.shape[0]
10 nlat = gplat.shape[0]
11 gpsfc = GPhys::IO.open('sp_for_Earth_T042_Desert.nc', 'sfcindex')
12
13 lons = gplon.val.to_a
14 lats = gplat.val.to_a
15
16 i = NArray.int(nlon).indgen!(0,1)
17 j = NArray.int(nlat).indgen!(0,1)
18
19 difference = VArray.new( NArray.sfloat(nlon,nlat), {"long_name
    =>"Precipitation difference"}, "Precipitation difference" )
20
21 lons.zip(i).each do |lon,i|
22   lats.zip(j).each do |lat,j|
23     gpprcpdif = (gpprcp2.cut('lat'=>lat,'lon'=>lon,'time
        '=>10800..14400).mean('time').val) - (gpprcp1.cut('lat
        '=>lat,'lon'=>lon,'time'=>10800..14400).mean('time').val
        )
24     difference[i,j] = gpprcpdif*86400*360
25   end
26 end
27
28 gplons = gpprcp1.axis('lon')
29 gplats = gpprcp1.axis('lat')
30
31 gpprcpdiff = GPhys.new(Grid.new(gplons,gplats),difference)
32
33 DCL.gropn(1)
34 DCL.sgpset('isub', 96)
35 DCL.sgpset('lfull', true)
36 DCL.uzfact(0.6)
```

```
37 DCL.udpset('LMSG',false)
38 GGraph.set_fig('viewport'=>[0.15,0.85,0.15,0.6])
39 GGraph.set_axes('xlabelint'=>60,'ylabelint'=>30)
40 GGraph.tone(gprrcpdiff,true,'tonc'=>true,'lev
    '=>[-4000,-2000,-1500,-1000,-750,-500,-250,0,250,500,750,1000,1500,
41 2000,4000], 'patterns
    '=>[10999,16999,22999,28999,34999,40999,46999,62999,68999,74999,
42 80999,86999,92999,98999])
43 GGraph.set_contour_levels('levels'=>[0,0.0001], 'index'=>[2,2])
44 GGraph.contour(gpsfc,false)
45 GGraph.colorbar
46 DCL.grcls
```

---

## 参考文献

- [1] 高橋 芳幸, 櫻村 博基, 竹広 真一, 石渡 正樹, 納多 哲史, 小高 正嗣, 堀之内 武, 林 祥介, DCPAM 開発グループ, 2018: 惑星大気モデル DCPAM, <http://www.gfd-dennou.org/library/dcpam/>, 地球流体電脳倶楽部.
- [2] ケッペンの気候区分図 判定表, <https://geo.skygrass.net/note/clim/koppen.pdf>