南北温度分布が規定する
大気大循環の構造と強度

岡山大学　環境生命自然科学研究科　地球科学コース
50M24752　前村　有耶

2026/02/13

要旨

大気の全球規模の循環は南北の温度差によって駆動・維持されている．本研究では，南北の温度分布が大気大循環をどのように規定するのかを調べるため，大気大循環モデル DCPAM5 を用いて南北の温度分布を変えた仮想的な惑星の大気大循環を計算した．計算は，太陽放射なし，水なしとして，南北温度分布は地表温度を固定して与えた．地表温度は，赤道と極域の温度差，極域の温度をパラメータにもつ緯度の関数として与え，それらを変えた 39 通りの計算をおこなった．現在の地球のような低緯度が高緯度よりも高温となるような温度分布の場合，南北の温度差が大きくなるほど，ハドレー循環は強く，ジェットの風速は速く，南北熱輸送最大値は大きくなった．現在の地球とは逆の低緯度が高緯度よりも低温となる温度分布の場合，ハドレー循環はほぼ停止した．低緯度が高緯度よりも高温となるような温度分布と比べて，ジェットの風速は速く，南北熱輸送最大値は小さくなった．

Abstract

Global-scale atmospheric circulation is driven and maintained by temperature difference between latitudes. In this study, we simulated the atmospheric circulation on a virtual planet with fixed surface temperature using the general circulation model DCPAM5, investigating how the temperature difference between latitudes controls the general circulation of the atmosphere. In all simulations, solar radiation and water were excluded. Surface temperature was given as a function of latitude with parameters including the temperature difference between the equator and polar regions and the temperature at the polar regions, and we conducted 39 simulations by varying these parameter. When low latitudes are warmer than high latitudes, the greater the temperature difference between equator and polar regions, the stronger the Hadley circulation, faster the jet, and the larger the meridional heat transport became. In contrast, when the temperature gradient is reversed, that is low latitudes are cooleer than high latitudes, the Hadley cirulation almost stopped and meridional heat trasnport became smaller.

目 次

第1章　南北の温度分布と大気大循環

大気の全球規模の循環を大気大循環といい，現在の地球の低緯度域では，赤道付近で加熱された空気が上昇した後に極方向へ流れ，緯度約 30 度付近で下降し，地表付近を赤道方向に戻るというハドレー循環，中緯度域では，対流圏上部で帯状の強い西風である西風ジェットが存在する．このような大気大循環は低緯度ほど高温で，高緯度ほど低温という現在の地球の温度分布によってつくられている．低緯度域のハドレー循環は，赤道域の暖かい空気が上昇することでつくられていて，南北の温度差に依存している．中緯度域の西風ジェットは，温度風シアによって説明することができ，ジェットの強度は南北の温度傾度によって決まる．したがって，南北の温度分布が変われば，ハドレー循環・ジェットの構造や強度も変わると予想される．

本研究では，大気大循環モデル DCPAM5 を用いて，南北の温度分布を変えた仮想的な惑星の大気大循環を計算することで，南北の温度分布が大気大循環をどのように決めているのかについて考察した．

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 を使用して大気大循環を計算した．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/) を参照されたい．

本研究では，以下の物理過程を用いた．放射過程は地球大気用放射モデルを用いた．地球大気用放射モデルでは，放射伝達方程式は Toon et al. (1989)，長波放射は Chou et al. (2001) に従って計算されている．乱流混合過程は Mellor and Yamada (1974, 1982)，乾燥対流調節は Manabe (1965) に従って計算されている．

2.2 実験設定

実験の設定は主に現在の地球の計算に用いられる標準設定を用いたが，太陽放射 (2.2.3 節) や水 (2.2.4 節) などいくつかは標準設定から変更した．

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．また，時間発展を計算する際の時間ステップは 900 秒とした．

2.2.2 地表温度

地表温度は固定して与え，時間で変化しないものとした．地表温度は，緯度の関数として与え，経度方向には一様とした．地表温度の緯度分布を与える関数は，Neale and Hoskins (2001) が大気大循環モデルの相互比較のために定義した以下の式を用いた．

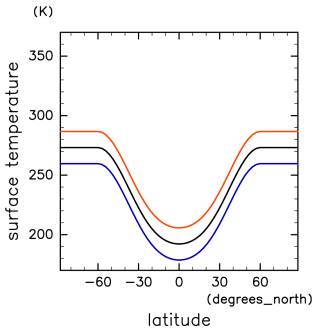
$$T = \begin{cases} T_{pole} + \Delta T\left(1 - \frac{1}{2}\sin^2\left(\frac{3}{2}\phi\right) - \frac{1}{2}\sin^4\left(\frac{3}{2}\phi\right)\right) & \text{for } |\phi| < \frac{\pi}{3} \\ T_{pole} & \text{for } |\phi| \geqq \frac{\pi}{3} \end{cases}$$

ここで，ΔT は赤道と極域の温度差，T_{pole} は極域の温度，ϕ は緯度である．Neale and Hoskins (2001) は地球を想定しているため，$\Delta T = 27$ (K), $T_{pole} = 273.15$ (K) としているが，本研究では ΔT と T_{pole} をパラメータとした．ΔT を正にすると低緯度は高緯度よりも高温になり，ΔT を負にすると低緯度は高緯度よりも低温になる．赤道と極域の温度差 ΔT は 13 通り，極域の温度 T_{pole} は 3 通り，これらを組み合わせた計 39 通りの地表温度分布それぞれで計算をおこなった (表 2.2，図 2.1)．

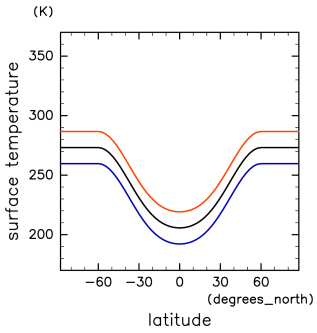
表 2.1: 本研究で用いた ΔT と T_{pole} の値 (K)．

ΔT	-81, -67.5, -54, -40.5, -27, -13.5, 0, 13.5, 27, 40.5, 54, 67.5, 81
T_{pole}	259.65, 273.15, 286.65

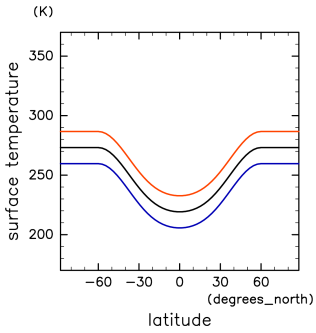
(a)



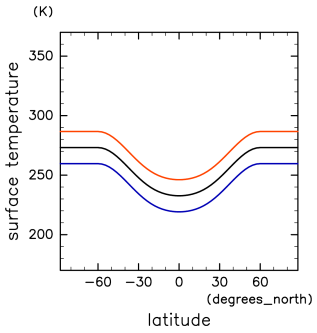
(b)



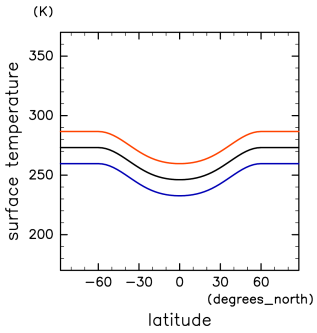
(c)



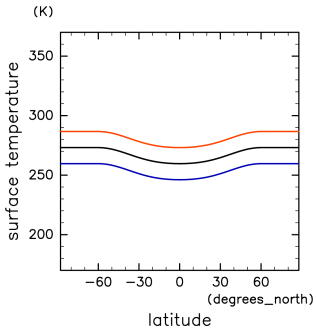
(d)



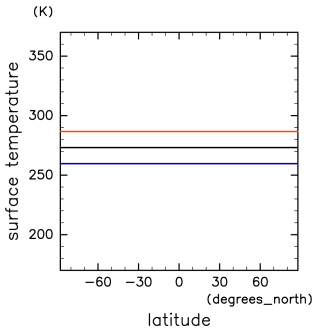
(e)



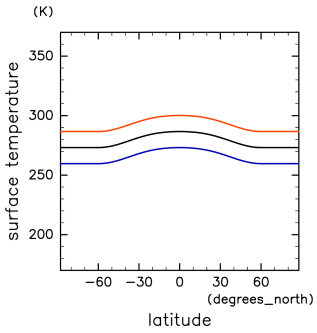
(f)



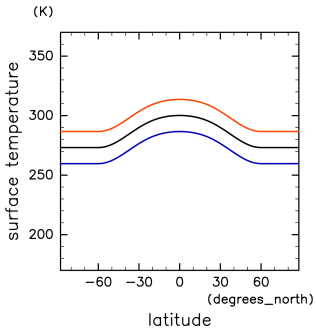
(g)



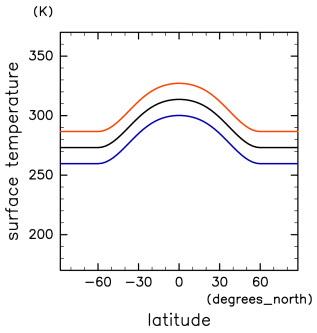
(h)



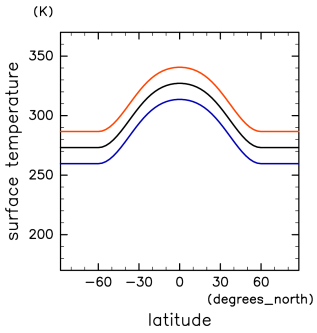
(i)



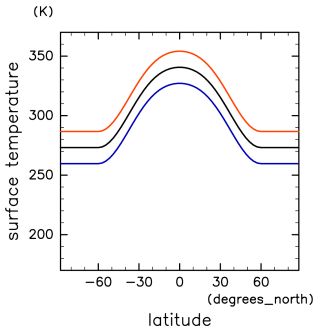
(j)



(k)



(l)



(m)

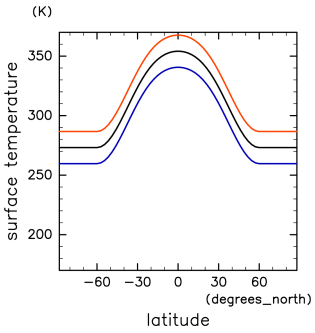


図 2.1: 地表温度の緯度分布．縦軸は温度，横軸は緯度．赤は T_{pole}=286.65K，黒は T_{pole}=273.15K，青は T_{pole}=259.65K．(a)$\Delta T = -81$K，(b)$\Delta T = -67.5$K，(c)$\Delta T = -54$K，(d)$\Delta T = -40.5$K，(e)$\Delta T = -27$K，(f)$\Delta T = -13.5$K，(g)$\Delta T = 0$K，(h)$\Delta T = 13.5$K，(i)$\Delta T = 27$K，(j)$\Delta T = 40.5$K，(k)ΔT= 54K，(l)ΔT= 67.5K，(m)ΔT= 81K

2.2.3 太陽放射

本研究では，全ての計算を太陽放射ゼロにしておこなった．太陽放射をゼロにしたことで，大気の大循環は太陽放射加熱の影響を受けず，地表温度だけで決まることになる．また，太陽放射をゼロにしたので，自転軸傾斜角や離心率などの軌道要素は惑星の加熱に影響しない．

2.2.4 水

本研究では，全ての計算を惑星表層に水が存在しない条件でおこなった．水の蒸発・凝結にともなう熱の出入りは存在せず，水蒸気や雲が放射に影響を及ぼすこともない．

2.2.5 地表面・地形

地表は全て砂漠とし，標高は場所によらず 0 m で一様とした．

2.2.6 大気組成

地球の乾燥空気に対応する大気組成を与えた．ただし，オゾンはゼロとした．この大気において放射活性を持つ成分は二酸化炭素だけで，二酸化炭素の濃度分布は時間・空間いずれも一様である．

2.2.7 暦

解析を簡単にするため，1 年は 360 日 (30 日× 12ヶ月) とした.

2.2.8 初期条件

大気は 280K の等温静止状態を初期状態とした.

2.2.9 積分期間

各実験について，初期状態から 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 1 年を解析に使用した.

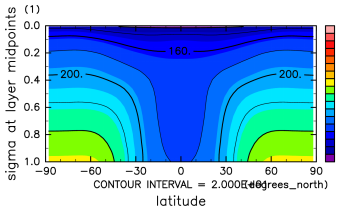
第3章　結果

3.1　子午面平均場の構造

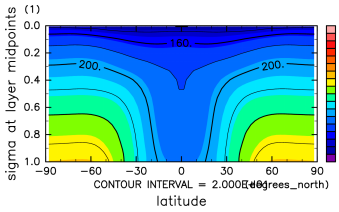
3.1.1　気温

図 3.1 は経度・時間で平均した気温の子午面分布，図 3.2 は経度 · 時間平均した南北温度傾度の子午面分布である．ΔT が正の場合 (h-m) と負の場合 (a-f) で分布の特徴はやや異なる．ΔT が正のとき，σ=0.5 付近の大気中層の南北温度傾度が最大となる領域は，40〜50 度付近にある．一方で，ΔT が負のとき，σ=0.5 付近の大気中層の南北温度傾度が最大となる領域は，20〜30 度付近にある．

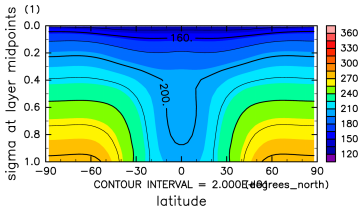
(a1)



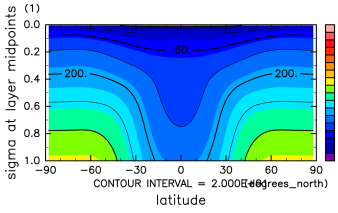
(a2)



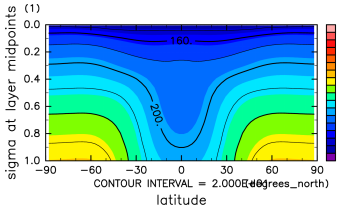
(a3)



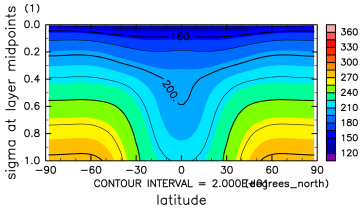
(b1)



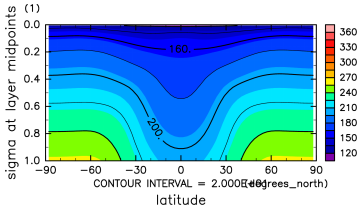
(b2)



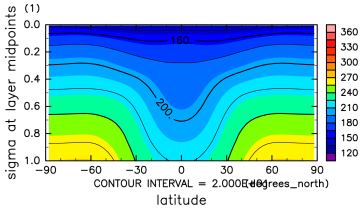
(b3)



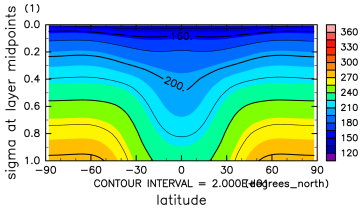
(c1)



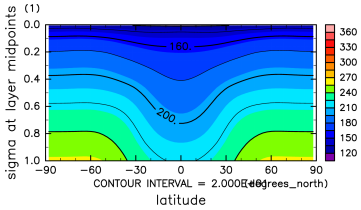
(c2)



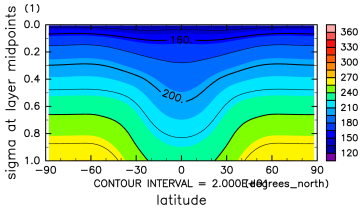
(c3)



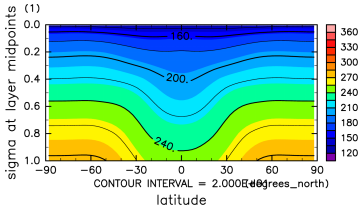
(d1)



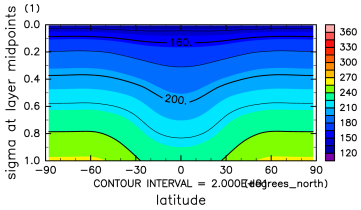
(d2)



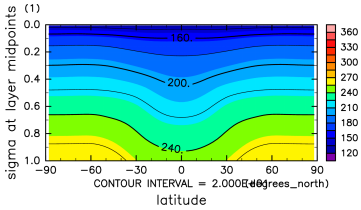
(d3)



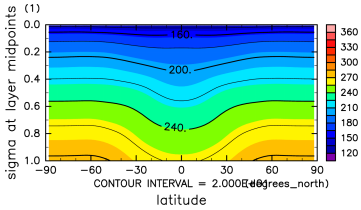
(e1)



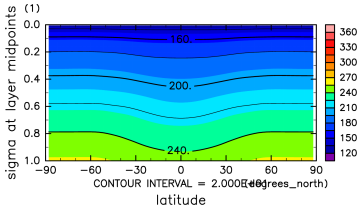
(e2)



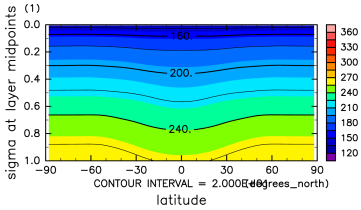
(e3)



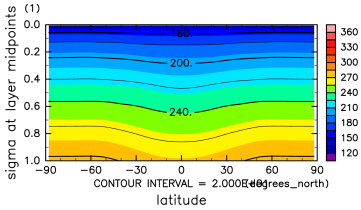
(f1)



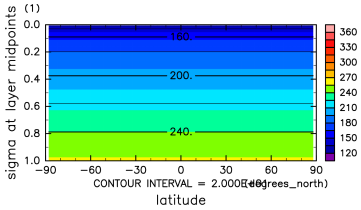
(f2)



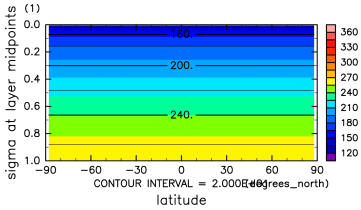
(f3)



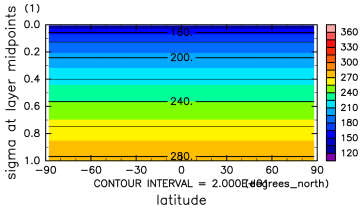
(g1)



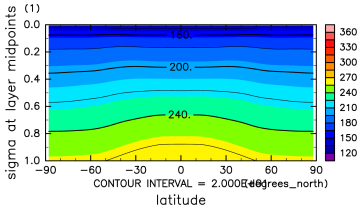
(g2)



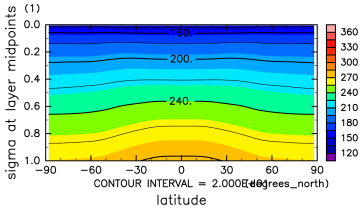
(g3)



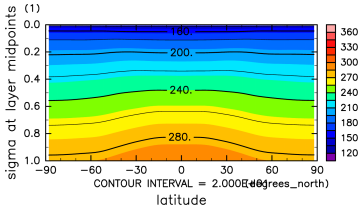
(h1)



(h2)



(h3)



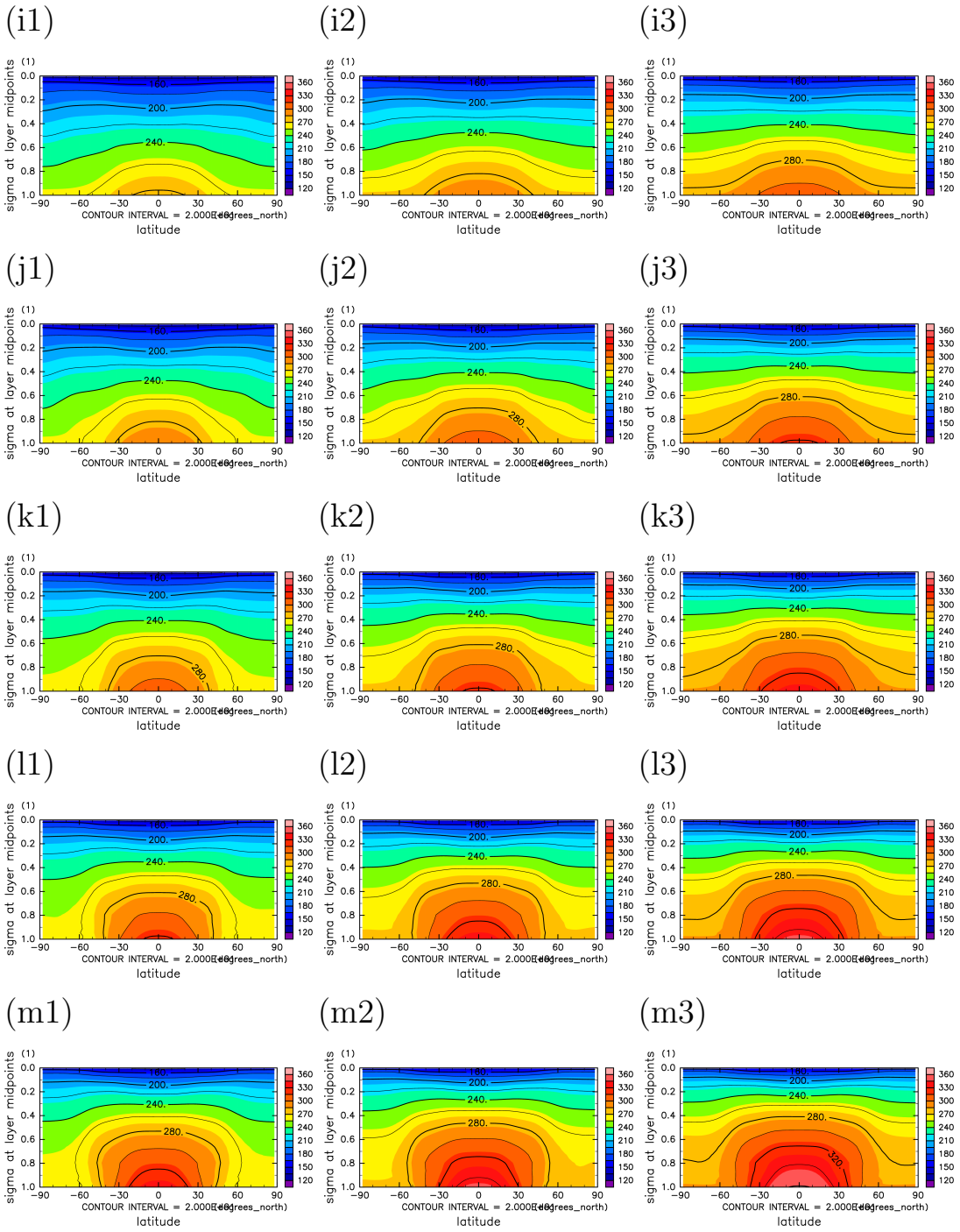
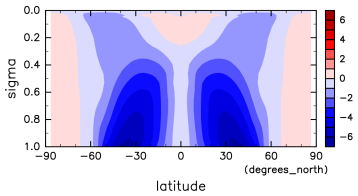
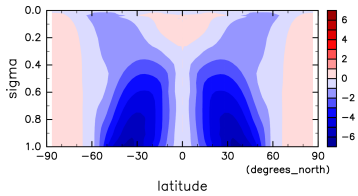


図 3.1: 年平均・東西平均した気温．縦軸は σ(気圧座標を地表面気圧で規格化)，横軸は緯度，単位は K．アルファベットは ΔT：(a)-81K，(b)-67.5K，(c)-54K，(d)-40.5K，(e)-27K，(f)-13.5K，(g)0K，(h)13.5K，(i)27K，(j)40.5K，(k)54K，(l)67.5K，(m)81K，数字は T_{pole}：(1)259.65K，(2)273.15K，(3)286.65K．(a1) なら ΔT=-81K，T_{pole}=259.65K となる．

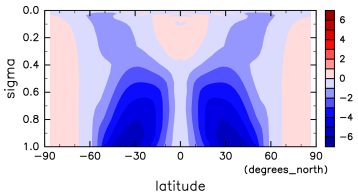
(a1)



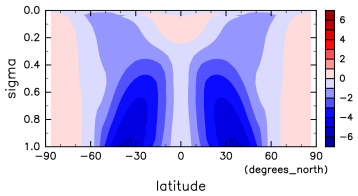
(a2)



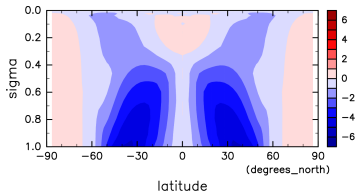
(a3)



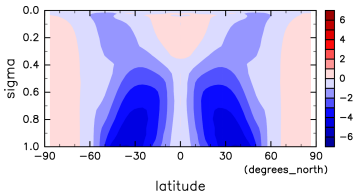
(b1)



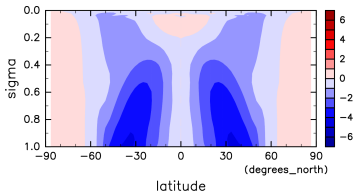
(b2)



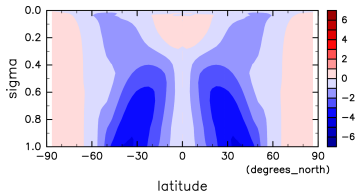
(b3)



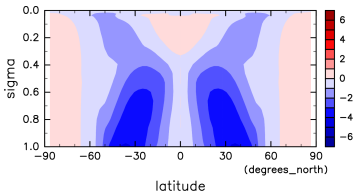
(c1)



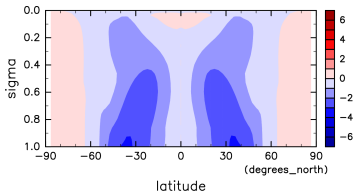
(c2)



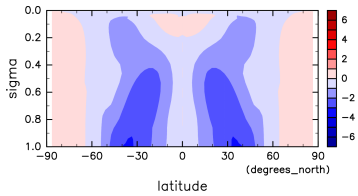
(c3)



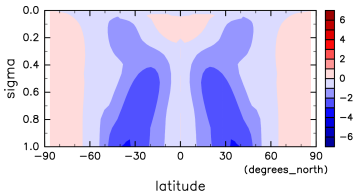
(d1)



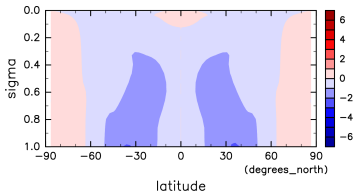
(d2)



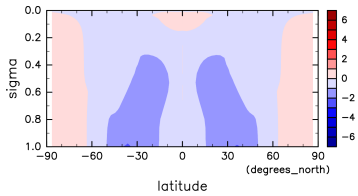
(d3)



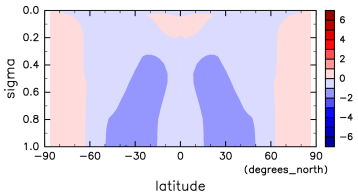
(e1)



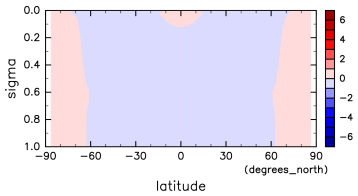
(e2)



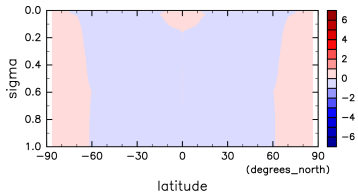
(e3)



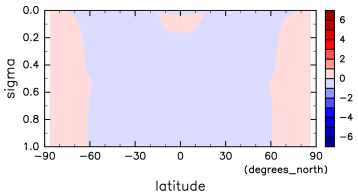
(f1)



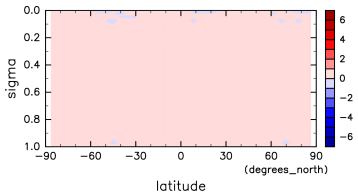
(f2)



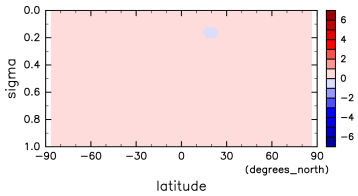
(f3)



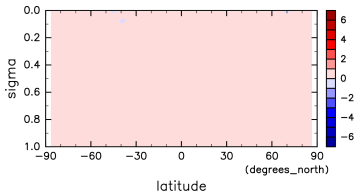
(g1)



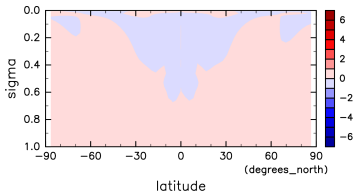
(g2)



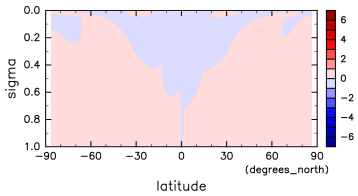
(g3)



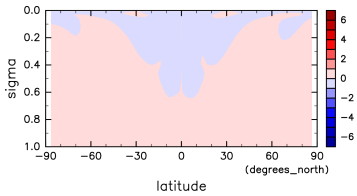
(h1)



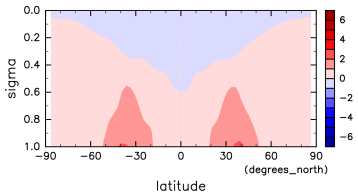
(h2)



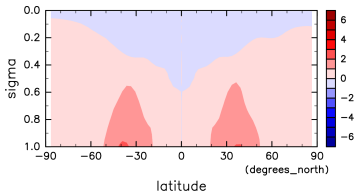
(h3)



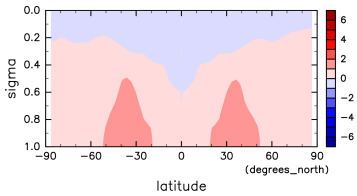
(i1)



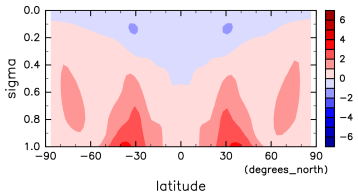
(i2)



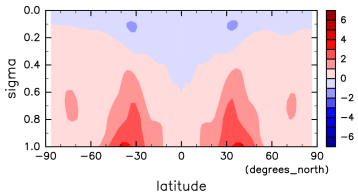
(i3)



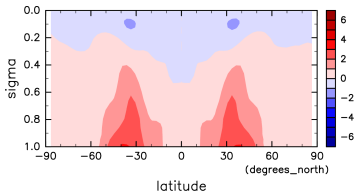
(j1)



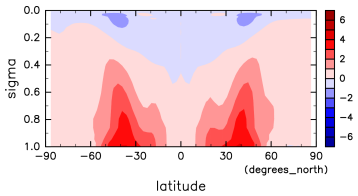
(j2)



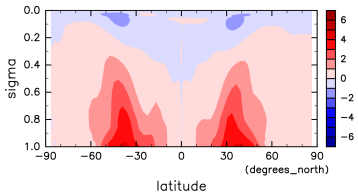
(j3)



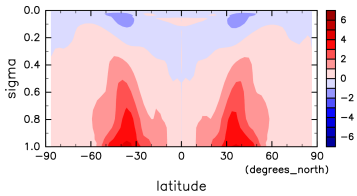
(k1)



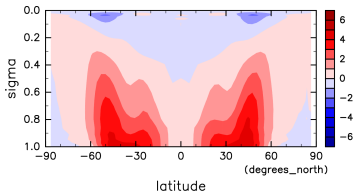
(k2)



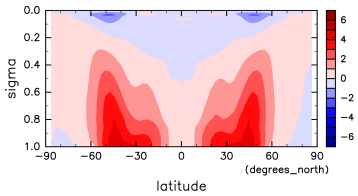
(k3)



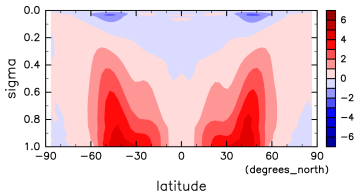
(l1)



(l2)



(l3)



(m1) (m2) (m3)

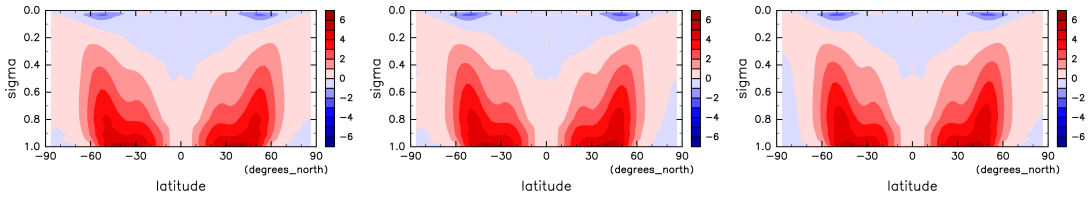


図 3.2: 年平均・東西平均した南北温度傾度．縦軸は σ(気圧座標を地表面気圧で規格化)，横軸は緯度，単位は K．アルファベットは ΔT：(a)$-$81K，(b)$-$67.5K，(c)$-$54K，(d)$-$40.5K，(e)$-$27K，(f)$-$13.5K，(g)0K，(h)13.5K，(i)27K，(j)40.5K，(k)54K，(l)67.5K，(m)81K，数字は T_{pole}：(1)259.65K，(2)273.15K，(3)286.65K．(a1) なら ΔT=$-$81K，T_{pole}=259.65K となる．

3.1.2 東西風速

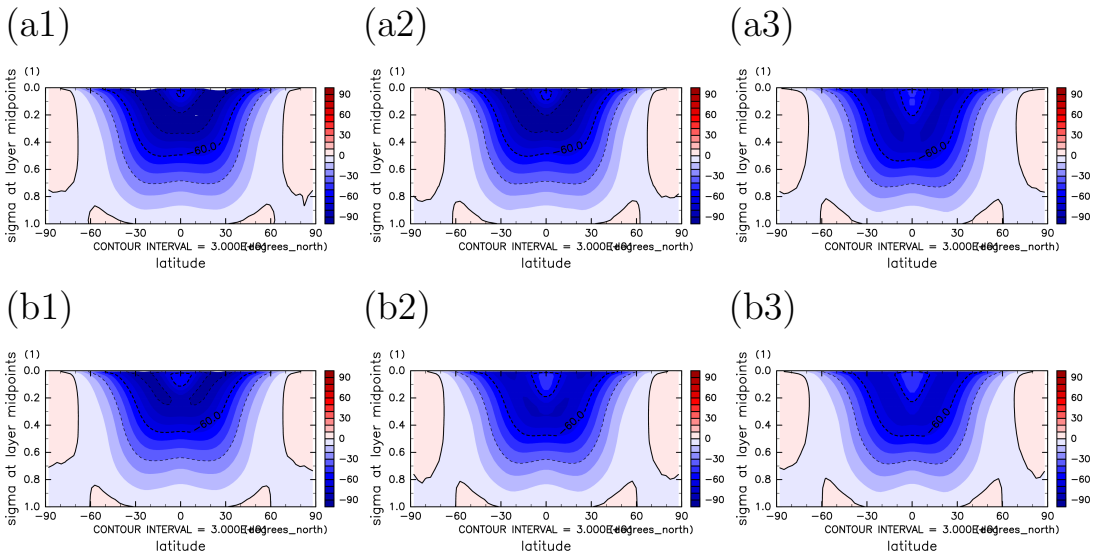
図 3.3 は経度・時間で平均した東西風速の子午面断面である．ΔT が正の場合 (h-m) は，中緯度域で西風となっていて，上空にジェットが存在する．ΔT が負の場合 (a-f) は，風向は逆転して東風となった．また，ΔT が負の場合 (a-f) は正の場合 (h-m) に比べて，ジェットの風速は全体的に速く，ジェットの位置は低緯度側へ移動し，より高い高度まで広がった．

ジェットは温度風シアによって説明することができる．地衡風平衡と静水圧平衡が成り立つとき，水平風鉛直シアと温度の水平勾配の間にある関係を温度風シアと呼ぶ．東西風の鉛直シアと温度の南北勾配の関係は以下の式で表される．

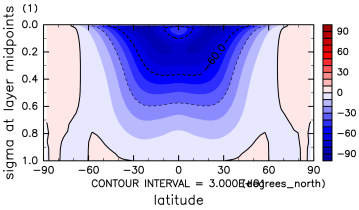
$$\frac{\partial u}{\partial p} = \frac{R}{fp}\frac{\partial T}{\partial y}$$

ここで，$\frac{\partial u}{\partial p}$ は東西風速の鉛直シア，R は気体定数，f はコリオリパラメータ，p は気圧，$\frac{\partial T}{\partial y}$ は南北の温度傾度である．

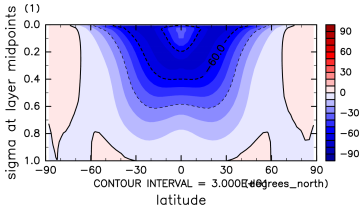
低緯度が高温で高緯度が低温，すなわち ΔT >0 のとき，$\frac{\partial u}{\partial p}$ <0 になる (f と $\frac{\partial T}{\partial y}$ の両方とも，赤道を境に符号を変えることに注意) ので，上層では u>0，すなわち西風が吹く．逆に，低緯度が低温で高緯度高温，すなわち ΔT <0 のとき，$\frac{\partial u}{\partial p}$ >0 になるので，上層では u<0 となり東風が吹く．また，3.1.1 節で見たように，大気中層の南北温度傾度 $\frac{\partial T}{\partial y}$ が最大となる緯度は，ΔT が正より負の場合の方が低緯度側である．ジェットの緯度が ΔT>0 より ΔT <0 で低緯度側なのは，大気の南北温度勾配が最大となる緯度の違いによるものだと考えられる．



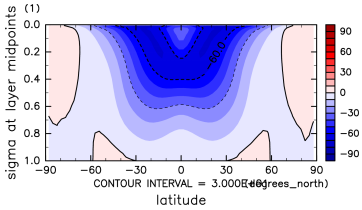
(c1)



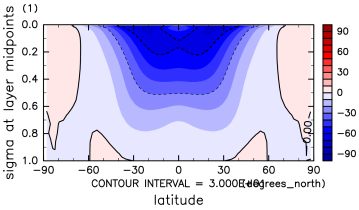
(c2)



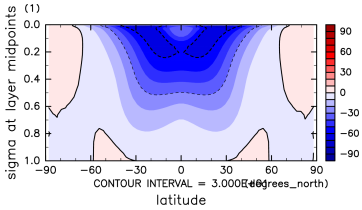
(c3)



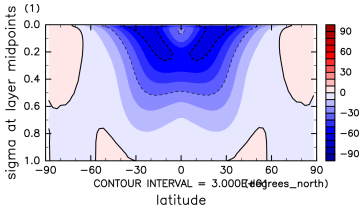
(d1)



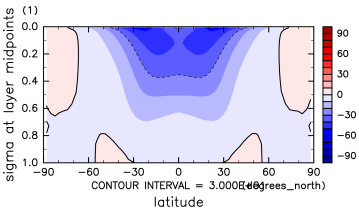
(d2)



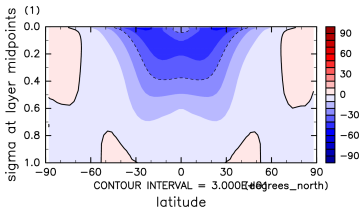
(d3)



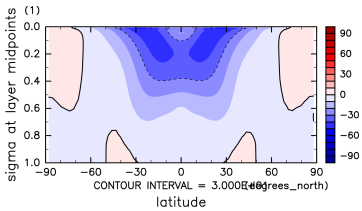
(e1)



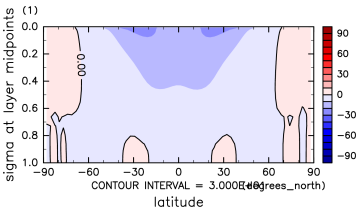
(e2)



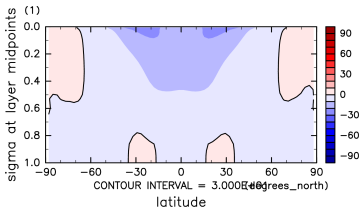
(e3)



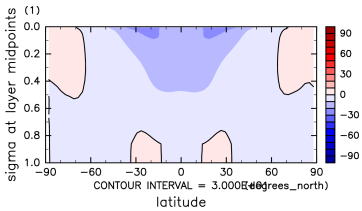
(f1)



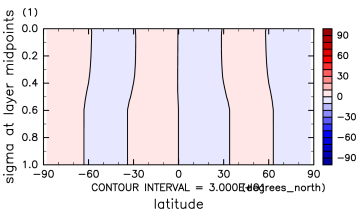
(f2)



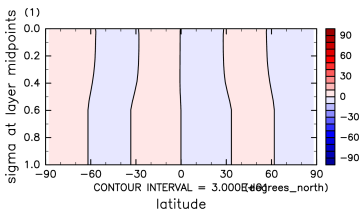
(f3)



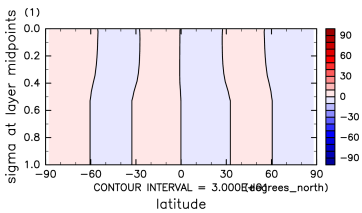
(g1)



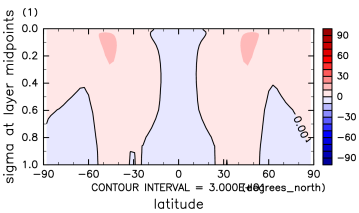
(g2)



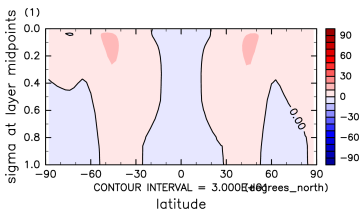
(g3)



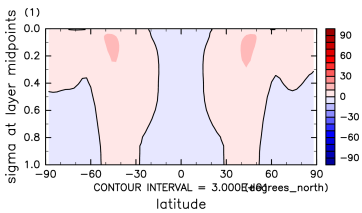
(h1)



(h2)



(h3)



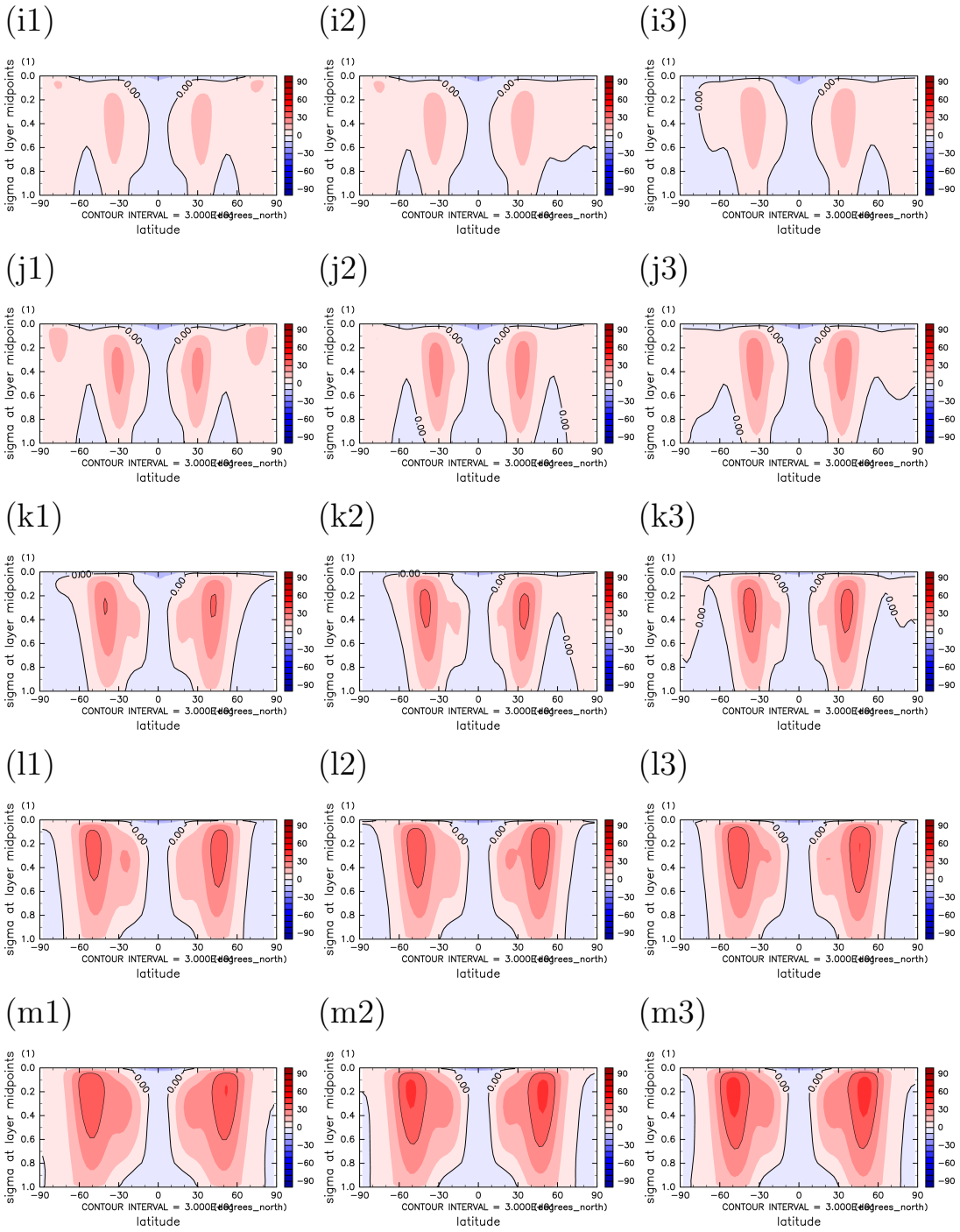


図 3.3: 年平均・東西平均した東西風速．縦軸は σ(気圧座標を地表面気圧で規格化)，横軸は緯度．単位は m/s で，風速の値が正の場合は西風，負の場合は東風．アルファベットは ΔT：(a)-81K，(b)-67.5K，(c)-54K，(d)-40.5K，(e)-27K，(f)-13.5K，(g)0K，(h)13.5K，(i)27K，(j)40.5K，(k)54K，(l)67.5K，(m)81K，数字は T_{pole}：(1)259.65K，(2)273.15K，(3)286.65K．(a1) なら $\Delta T=-81$K，$T_{pole}=259.65$K となる．

3.1.3　質量流線関数

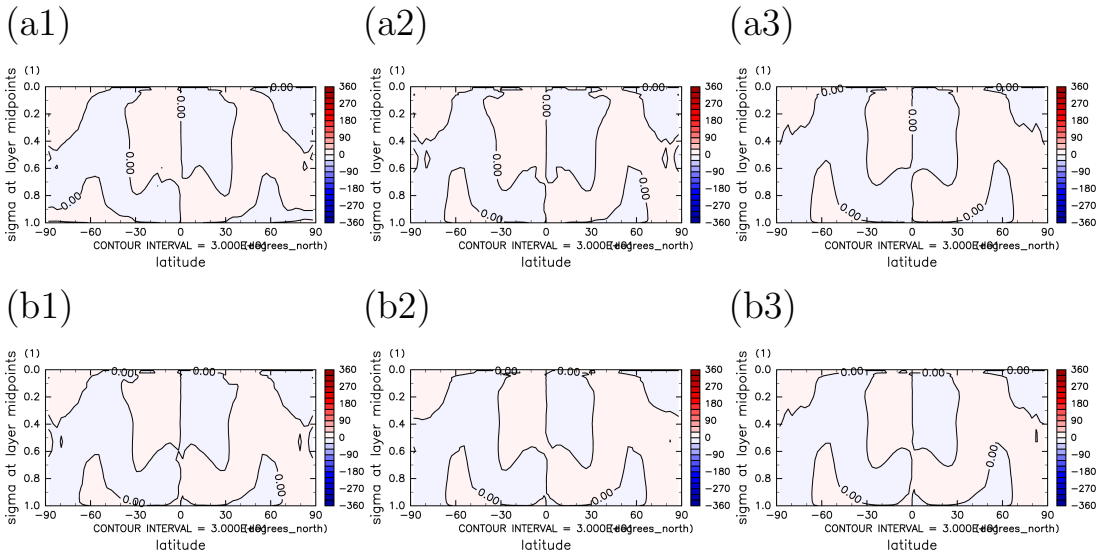
ハドレー循環の構造と強度を解析するために，計算結果を用いて子午面の質量流線関数を計算した．子午面の質量流線関数は次のように定義される．

$$\psi(\phi, p) = \frac{2\pi a \cos\phi}{g} \int_0^p \overline{v}(\phi, p) dp$$

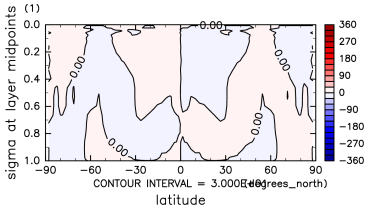
ここで，ψ は質量流線関数，a は惑星半径，ϕ は緯度，g は重力加速度，$\overline{v}$ は経度平均した南北風，p は気圧である．質量流線関数は循環の方向と単位時間あたりの流量を表す．空気は質量流線関数の等値線に沿って移動し，等値線の間隔が狭いほど単位時間あたりの流量は大きい．

図 3.4 は子午面の質量流線関数である．ΔT が正の場合 (h-m) は，緯度 0 度から 30 度付近にハドレー循環が形成された．ΔT が大きくなると，ハドレー循環の強度は強くなった．これは ΔT が大きくなることで赤道域の温度が高くなり，上昇流が強まったためと考えられる．

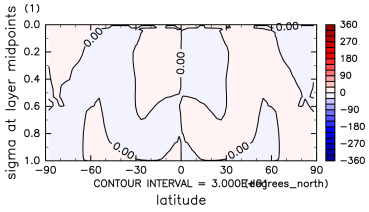
一方で，ΔT が負の場合 (a-f) は，ΔT の大小にかかわらず，子午面循環はほぼ停止した．



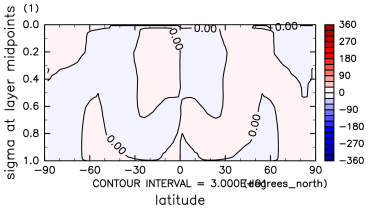
(c1)



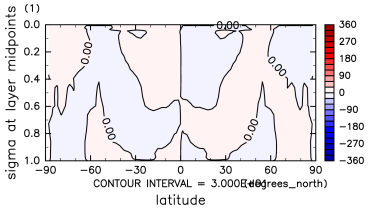
(c2)



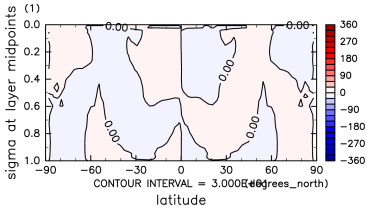
(c3)



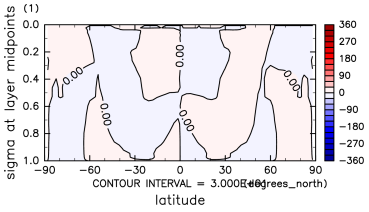
(d1)



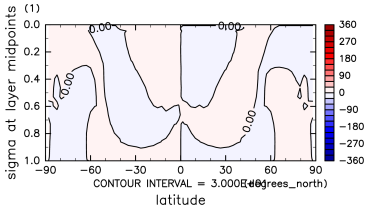
(d2)



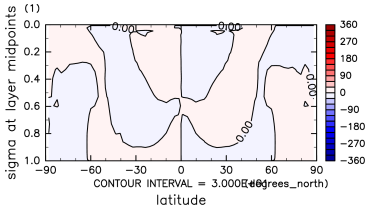
(d3)



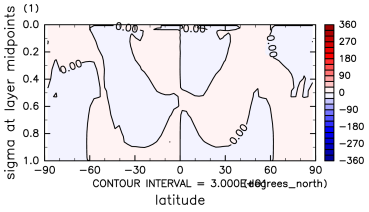
(e1)



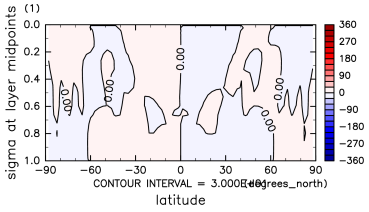
(e2)



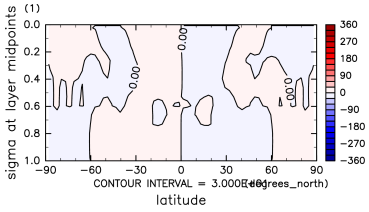
(e3)



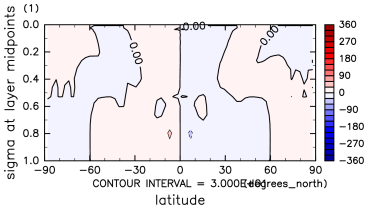
(f1)



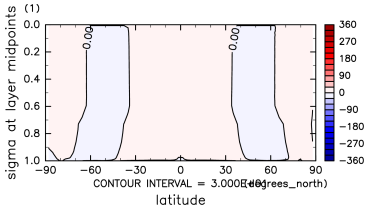
(f2)



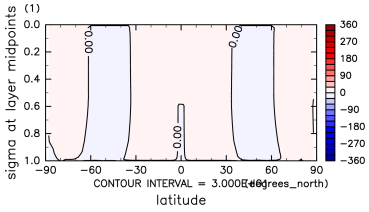
(f3)



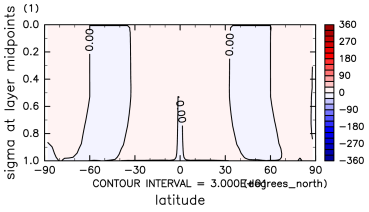
(g1)



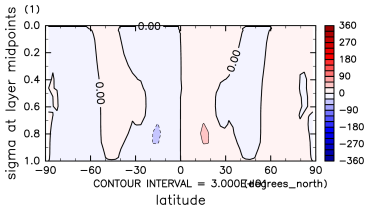
(g2)



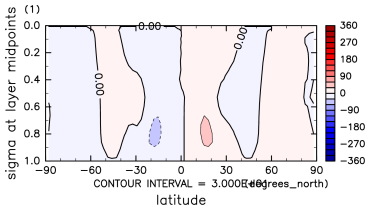
(g3)



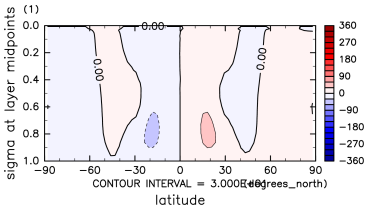
(h1)



(h2)



(h3)



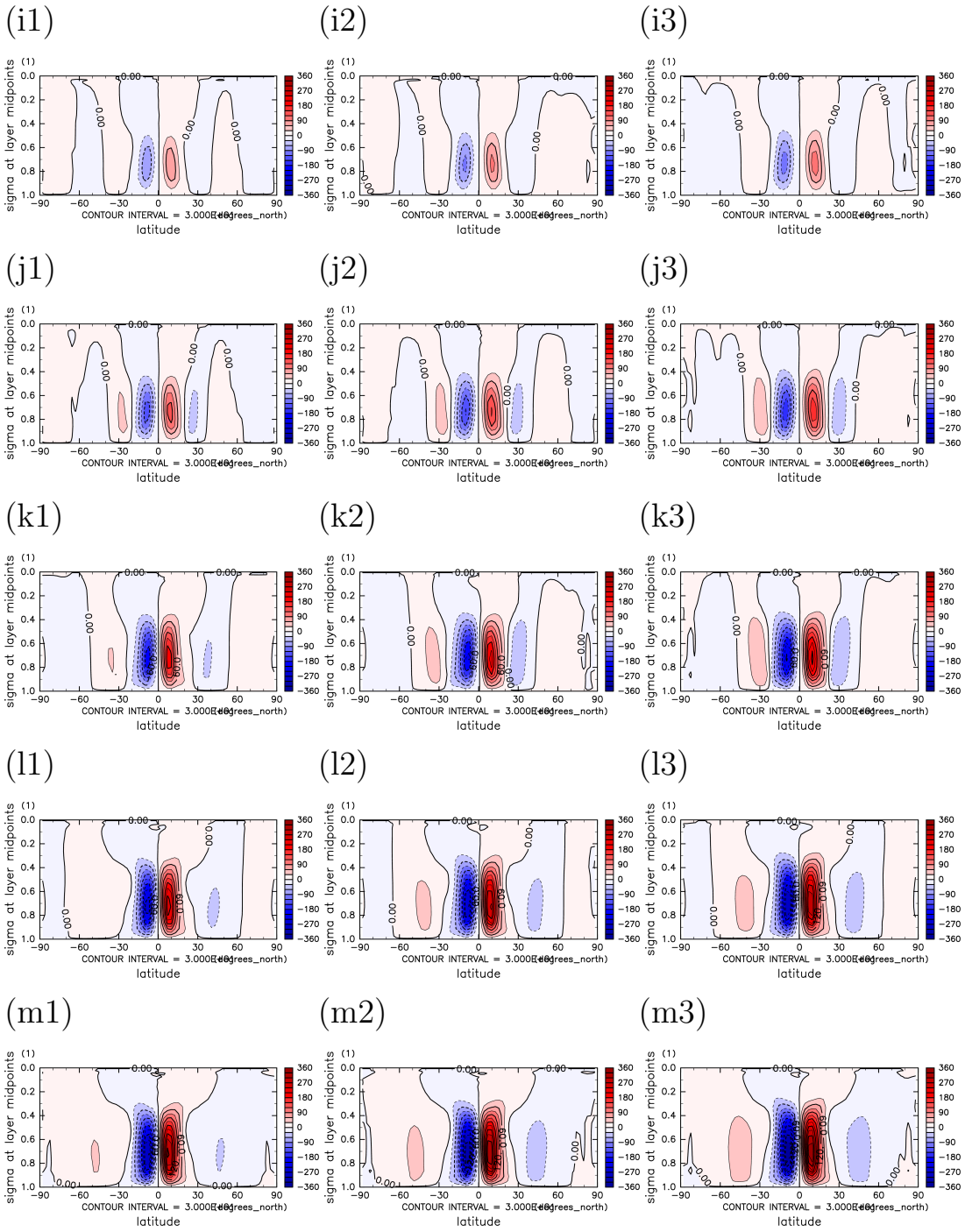


図 3.4: 子午面質量流線関数．縦軸は σ(気圧座標を地表面気圧で規格化)，横軸は緯度．単位は 10^9 kg/s で，質量流線関数の値が正の場合は時計回り，負の場合は反時計回り．アルファベットは ΔT：(a)-81K，(b)-67.5K，(c)-54K，(d)-40.5K，(e)-27K，(f)-13.5K，(g)0K，(h)13.5K，(i)27K，(j)40.5K，(k)54K，(l)67.5K，(m)81K，数字は T_{pole}：(1)259.65K，(2)273.15K，(3)286.65K．(a1) なら ΔT=-81K，T_{pole}=259.65K となる．

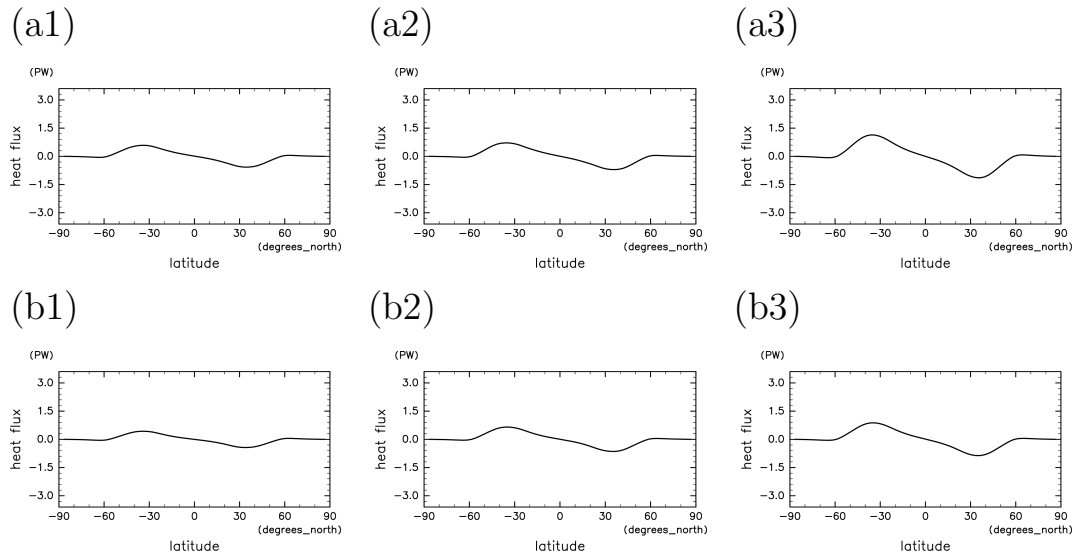
3.2 南北熱輸送量

定常状態において，各緯度帯に出入りする熱はバランスする．大気には，大気下端でSLR(地面から大気への長波放射による正味の熱フラックス) とSens(地面から大気への顕熱フラックス) が流入し，大気上端でOLR(大気から宇宙へ長波放射による熱フラックス) が流出する．各緯度帯でバランスしなかった熱 (SLR + Sens − OLR) は，大気によって南北に輸送される．したがって，各緯度 ϕ において北向きに輸送される熱量 $H(\phi)$ は，以下の式で与えられる．

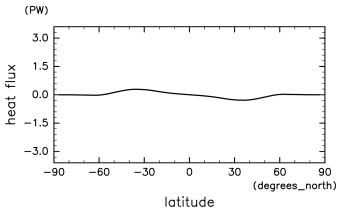
$$H(\phi) = \int_{-\frac{\pi}{2}}^{\phi} \int_{0}^{2\pi} \{SLR(\lambda,\phi) + Sens(\lambda,\phi) - OLR(\lambda,\phi)\} \cdot a\cos\phi d\lambda \cdot ad\phi$$

ここで，λ は経度である．

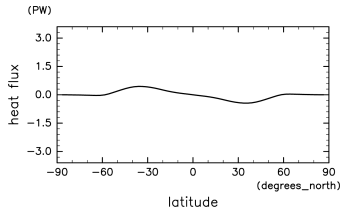
図 3.5 は時間平均した南北熱輸送量である．全般に，ΔT の絶対値が大きいほど南北熱輸送量は大きくなった．熱輸送の方向は，高温域から低温域へと向かう方向で，ΔT の符号によって南北温度勾配が逆転するのに合わせて，熱輸送の方向も逆転した．熱輸送量の緯度分布は，概ね現在の地球のそれに似ている (ΔT <0 のときは形はそのままに向きだけ逆転) が，その大きさは ΔT の符号によって異なり，ΔT <0(低緯度低温・高緯度高温) は ΔT >0(低緯度高温・高緯度低温) に比べて南北熱輸送量が小さくなった．



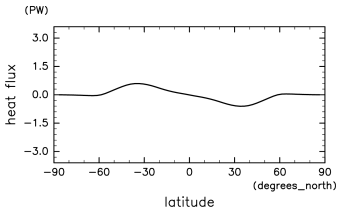
(c1)



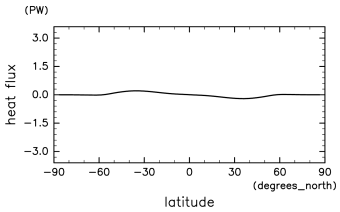
(c2)



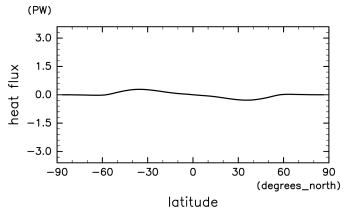
(c3)



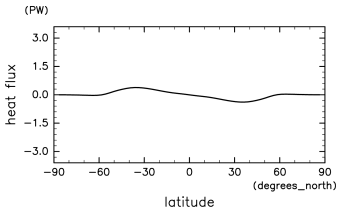
(d1)



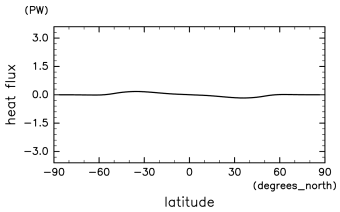
(d2)



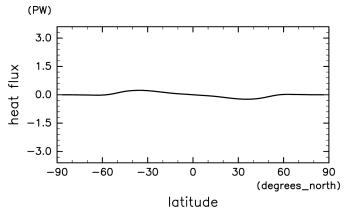
(d3)



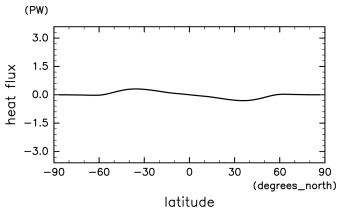
(e1)



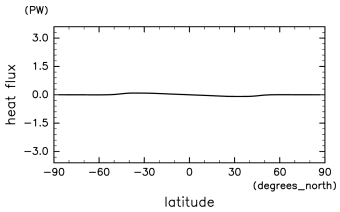
(e2)



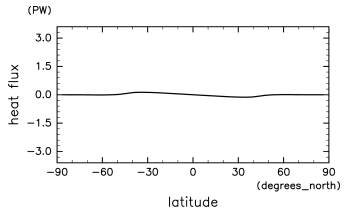
(e3)



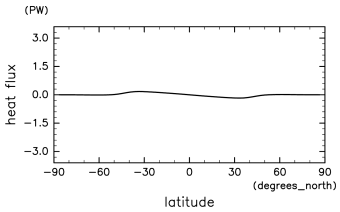
(f1)



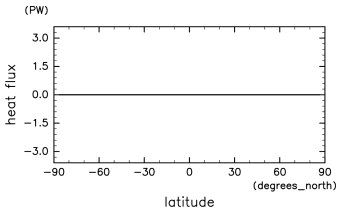
(f2)



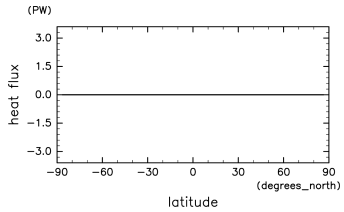
(f3)



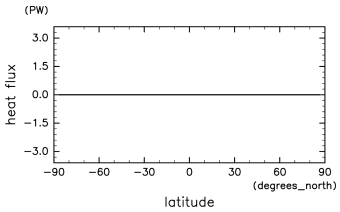
(g1)



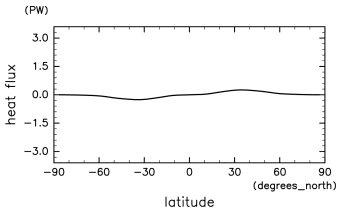
(g2)



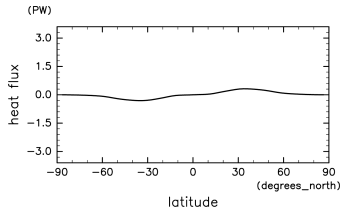
(g3)



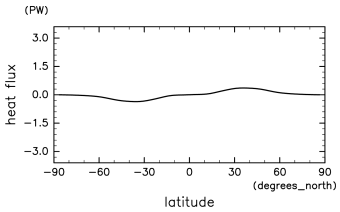
(h1)



(h2)



(h3)



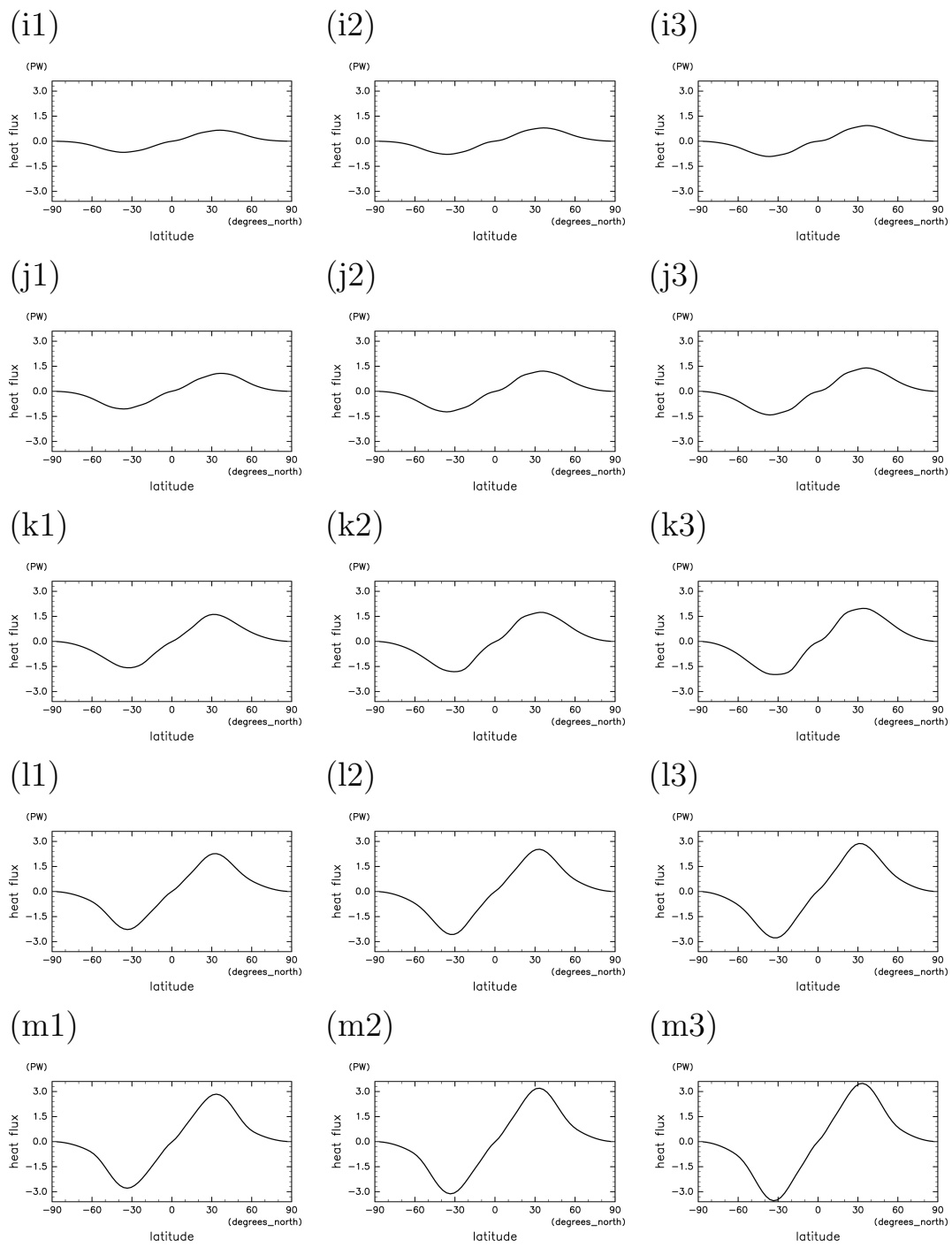


図 3.5: 南北熱輸送量．縦軸は北向き熱輸送量 (PW)，横軸は緯度．アルファベットは ΔT：(a)−81K，(b)−67.5K，(c)−54K，(d)−40.5K，(e)−27K，(f)−13.5K，(g)0K，(h)13.5K，(i)27K，(j)40.5K，(k)54K，(l)67.5K，(m)81K，数字は T_{pole}：(1)259.65K，(2)273.15K，(3)286.65K．(a1) なら ΔT=−81K，T_{pole}=259.65K となる．

3.3　南北温度差・極域温度依存性

3.3.1　ハドレー循環

本研究では，子午面の質量流線関数の絶対値が最大となる値を含む循環をハドレー循環とし，質量流線関数の最大値をハドレー循環の強度とした．図 3.6 はハドレー循環強度の南北温度差依存性である．ΔT が正の場合，南北の温度差が大きくなるとそれと比例してハドレー循環は強くなった．一方で，ΔT が負の場合は，南北の温度差に依らずハドレー循環はほぼ停止した．

ハドレー循環の強度は T_{pole} によっても多少変化し，T_{pole} が高いほどハドレー循環は強くなったが，ΔT に比べるとその影響は小さい．

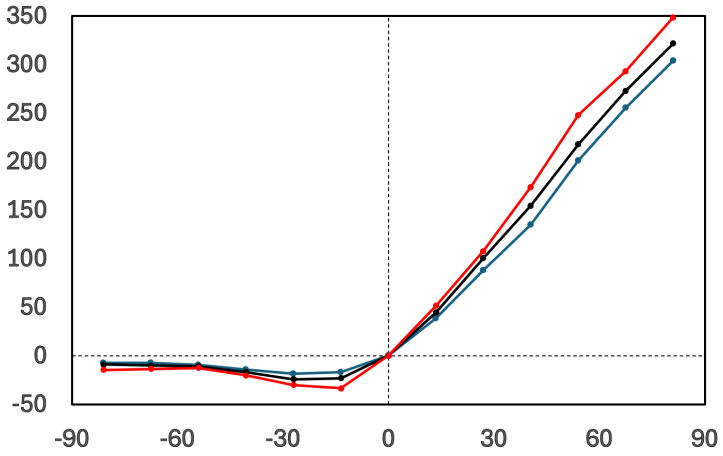


図 3.6: ハドレー循環の南北温度差依存性．縦軸はハドレー循環の質量流線関数の最大値，横軸は ΔT．赤は T_{pole}=286.65K，黒は T_{pole}=273.15K，青は T_{pole}=259.65K．ハドレー循環の質量流線関数の最大値の単位は 10^9 kg/s，符号は回転の向きが現在の地球と同じ場合は正，逆の場合は負．

3.3.2　ジェット

本研究では，東西風速の絶対値が最大となる場所にジェットがあるものとし，風速の最大値をジェットの速さとした．図 3.7 はジェットの速さの南北温度差依存性である．ジェットの速さを南北温度傾度に対してプロットした結果は，原点を通る右上がりの分布を示した．すなわち，ジェットの風向は温度傾度の符号によって決まり，風速は南北の温度差に大まかに比例している．しかし，ΔT が正の場合と負の場合で風速の比例係数は異なり，負の場合の方が比例係数は大きい．また，ΔT が負のとき，ΔT=−40.5 を境に比例係数が変わるように見える．

ジェットも子午面循環と同様に，ΔT に比べると T_{pole} の影響は小さい．ただし，ΔT <−40.5 でジェットの速さは T_{pole} によって変わり，T_{pole} が低温になるほどジェットは速くなった．

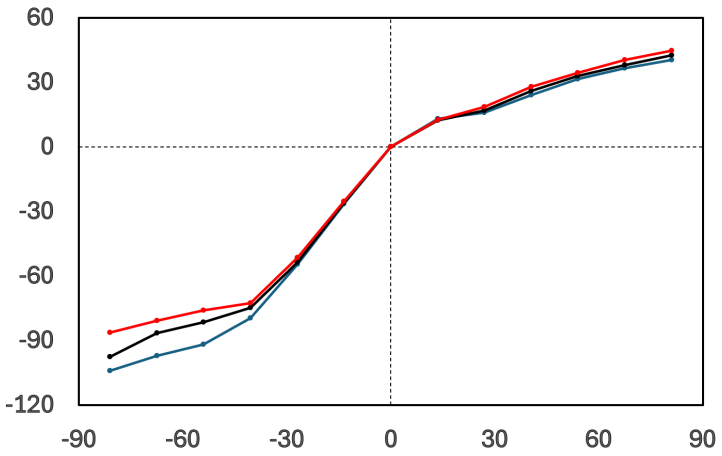


図 3.7: ジェットの南北温度差依存性．縦軸はジェットの風速の最大値，横軸は ΔT．赤は T_{pole}=286.65K，黒は T_{pole}=273.15K，青は T_{pole}=259.65K．

3.3.3 南北熱輸送量

図 3.8 は南北熱輸送最大値の南北温度差依存性である．ここでは低緯度から高緯度への輸送を正とし，高緯度から低緯度への輸送を負とした．南北熱輸送の最大値は，概ね南北温度差に比例し，温度差が大きいほど大きくなる傾向を示すが，ΔT が正と負で熱輸送量は大きく異なり，ΔT が負のときの熱輸送量は ΔT が正のときに比べるとかなり小さくなった．

南北熱輸送量の最大値は T_{pole} によって多少変わり，T_{pole} が高いほど南北熱輸送は大きくなった．

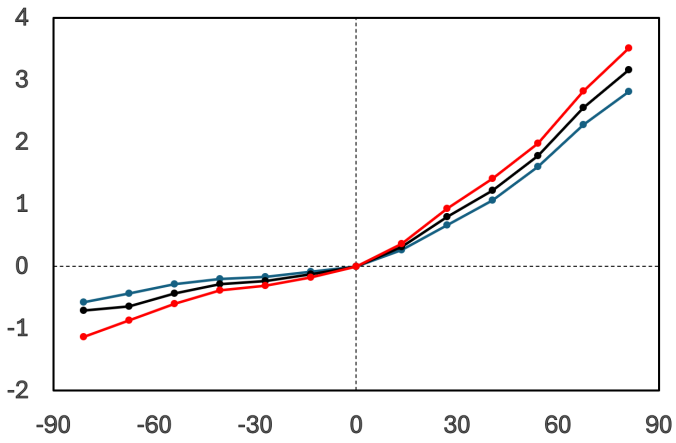


図 3.8: 南北熱輸送最大値の南北温度差依存性．縦軸は南北熱輸送量の最大値，横軸は ΔT．赤線は T_{pole}=286.65K，黒線は T_{pole}=273.15K，青線は T_{pole}=259.65K．南北熱輸送量の単位は PW，符号は極向きに熱を輸送している場合は正，逆向きの場合は負．

3.4　ハドレー循環と南北熱輸送量の関係

図 3.9 はハドレー循環強度と南北熱輸送最大値の関係である．$\Delta T \geqq -13.5$ のとき，南北熱輸送最大値はハドレー循環の強度にほぼ比例している．一方で，$\Delta T < -13.5$ のとき，南北熱輸送最大値はハドレー循環の強度に依らない．

このことから，$\Delta T \geqq -13.5$ のとき，南北熱輸送はハドレー循環によっておこなわれていて，$\Delta T < -13.5$ のとき，南北熱輸送はハドレー循環以外のものによっておこなわれていると考えられる．

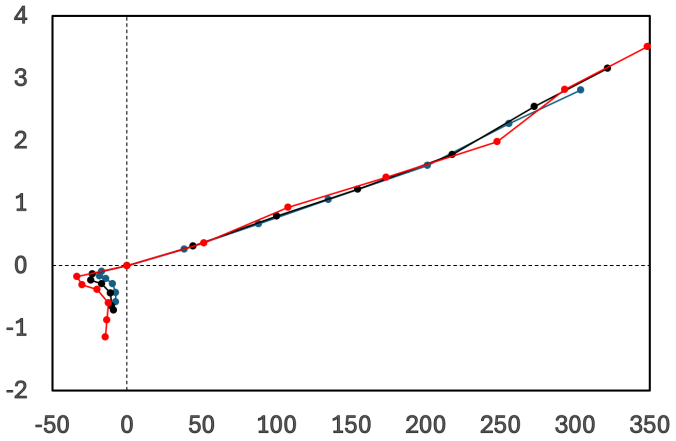


図 3.9: ハドレー循環強度と南北熱輸送最大値の関係．縦軸は南北熱輸送量の最大値，横軸はハドレー循環の質量流線関数の最大値．赤は T_{pole}=286.65K，黒は T_{pole}=273.15K，青は T_{pole}=259.65K．南北熱輸送量の単位は PW，符号は極向きに熱を輸送している場合は正，逆向きの場合は負．ハドレー循環の質量流線関数の単位は 10^9 kg/s，符号は回転の向きが現在の地球と同じ場合は正，逆の場合は負．

第4章　まとめ

大気大循環モデル DCPAM を用いて，南北の温度分布を変えた仮想的な惑星の大気大循環の計算を 39 通りおこなった．ハドレー循環の強度は ΔT が正の場合，南北の温度差が大きくなるほど大きくなった．ΔT が負の場合，子午面循環はほぼ停止した．ジェットの風速は南北の温度差に大まかに比例するが，負の場合の方が比例係数は大きい．南北熱輸送は高温域から低温域へ向かう方向におこなわれ，熱輸送量は南北の温度差が大きくなるほど大きくなるが，ΔT が負の場合は正の場合よりも小さい．また，$\Delta T \geqq -13.5$ のとき，南北熱輸送最大値はハドレー循環強度に比例した．極域の温度も大気大循環に多少影響を与えるが，ΔT に比べるとその影響は小さい．

謝辞

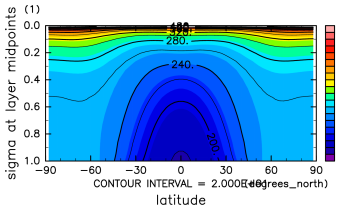
本研究を進めるにあたって，ご指導いただきました主指導教官であるはしもとじょーじ教授には心より深く感謝申し上げます.

付録

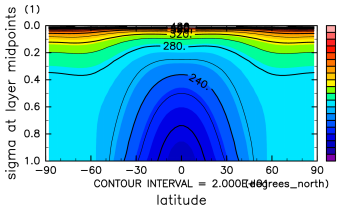
1.1 図録

1.1.1 温位

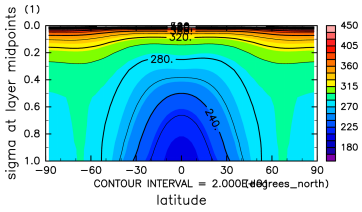
(a1)



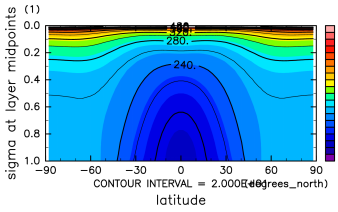
(a2)



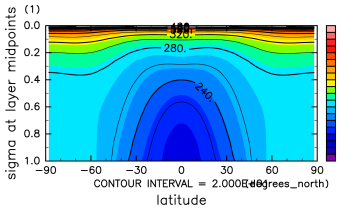
(a3)



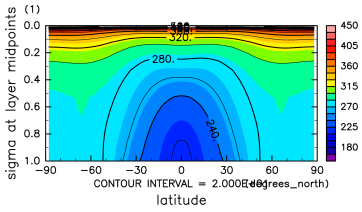
(b1)



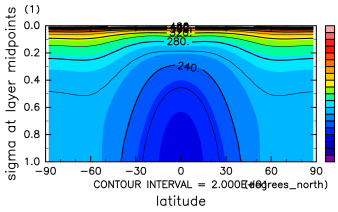
(b2)



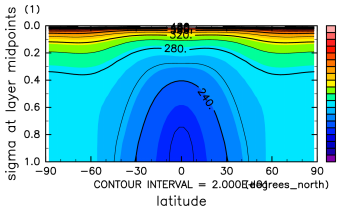
(b3)



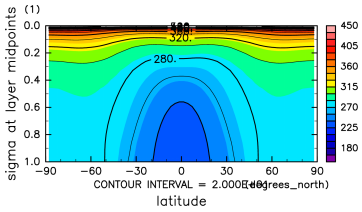
(c1)



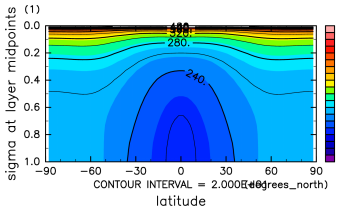
(c2)



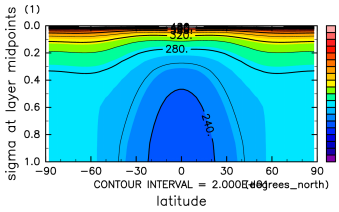
(c3)



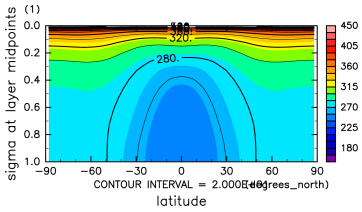
(d1)



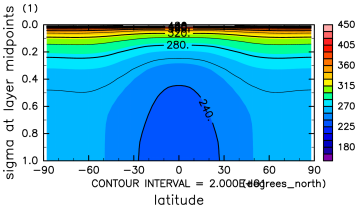
(d2)



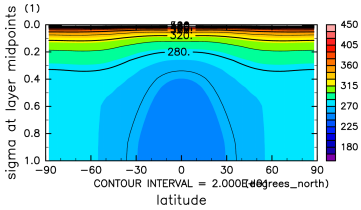
(d3)



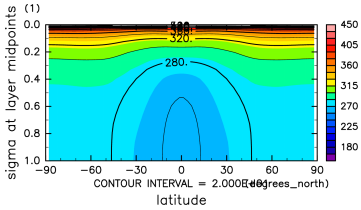
(e1)



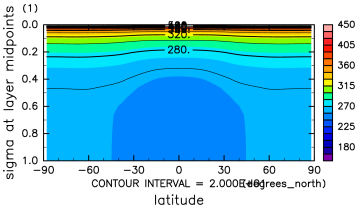
(e2)



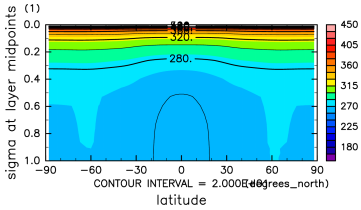
(e3)



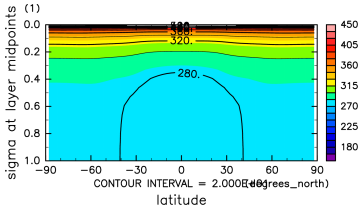
(f1)



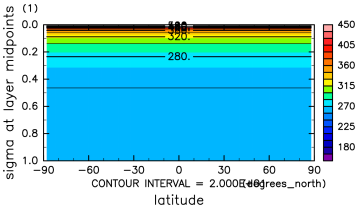
(f2)



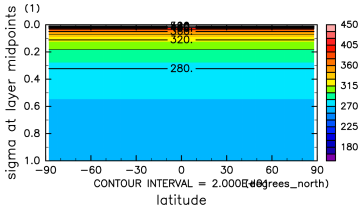
(f3)



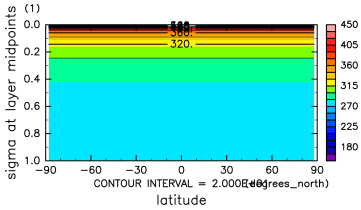
(g1)



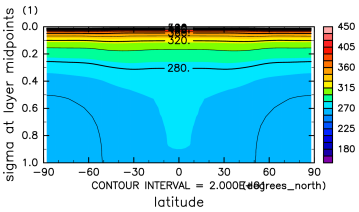
(g2)



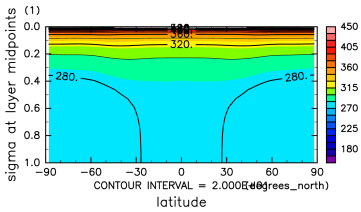
(g3)



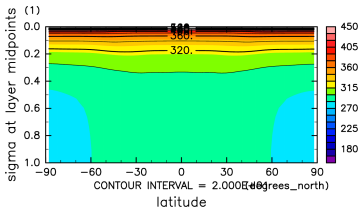
(h1)



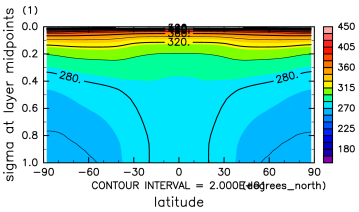
(h2)



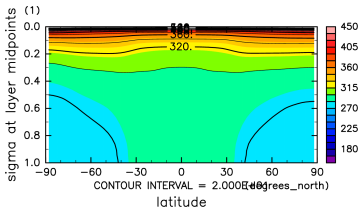
(h3)



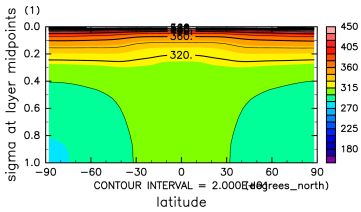
(i1)



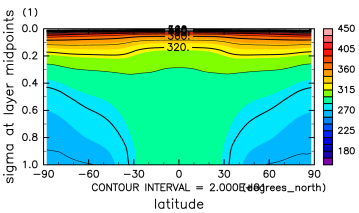
(i2)



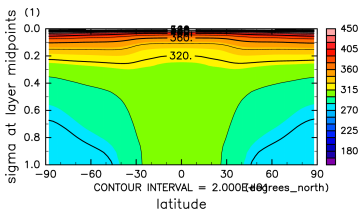
(i3)



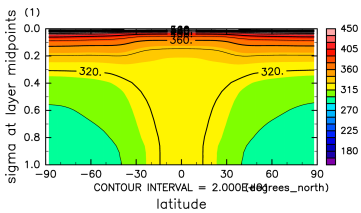
(j1)



(j2)



(j3)



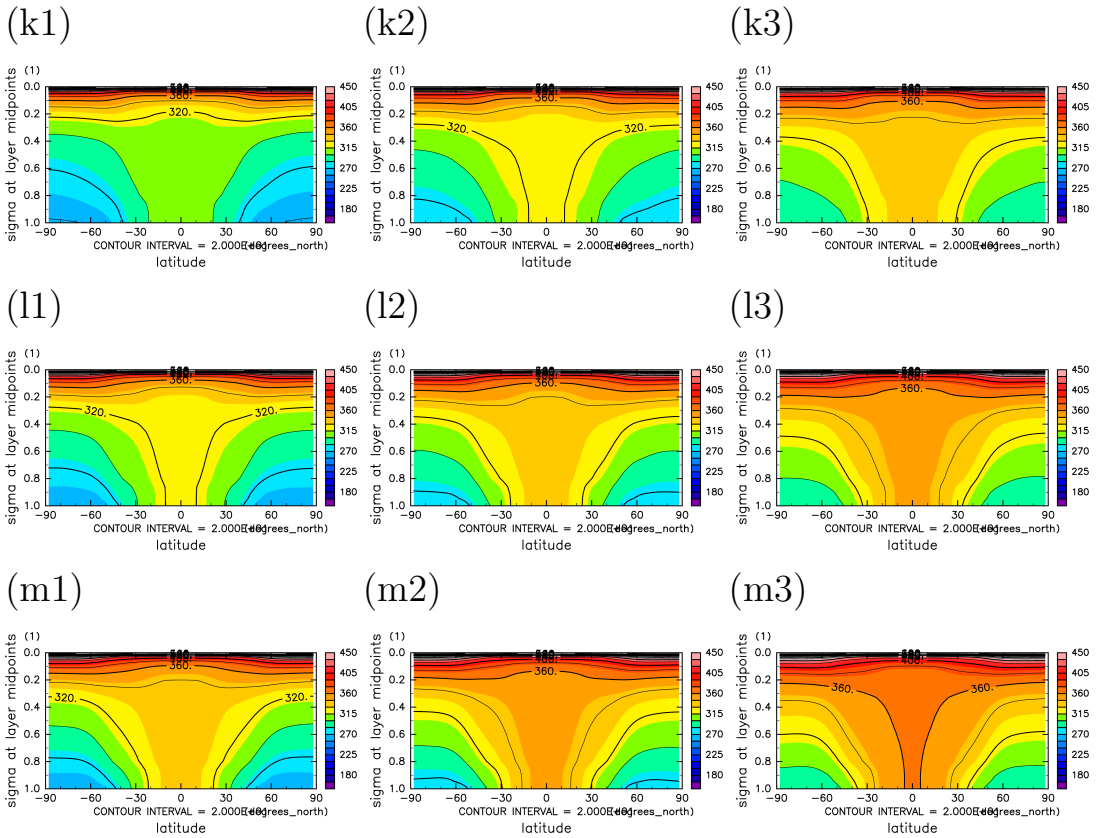
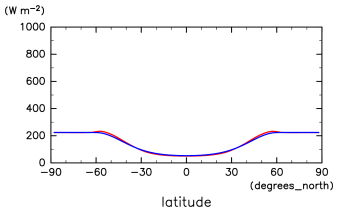


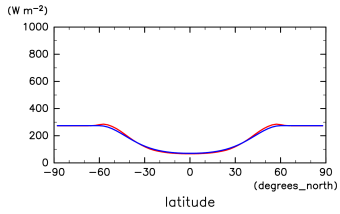
図 1.1: 年平均・東西平均した温位．縦軸は σ(気圧座標を地表面気圧で規格化)，横軸は緯度，単位は K．アルファベットは ΔT：(a)−81K，(b)−67.5K，(c)−54K，(d)−40.5K，(e)−27K，(f)−13.5K，(g)0K，(h)13.5K，(i)27K，(j)40.5K，(k)54K，(l)67.5K，(m)81K，数字は T_{pole}：(1)259.65K，(2)273.15K，(3)286.65K．(a1) なら ΔT=−81K，T_{pole}=259.65K となる．

1.1.2　大気の熱収支

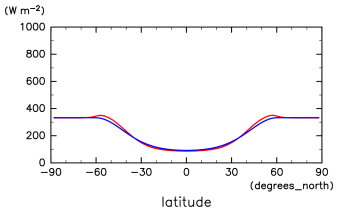
(a1)



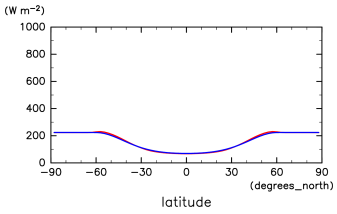
(a2)



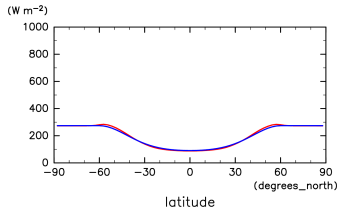
(a3)



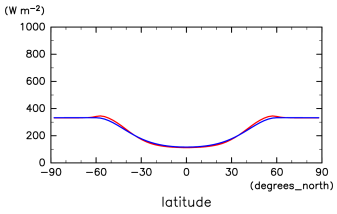
(b1)



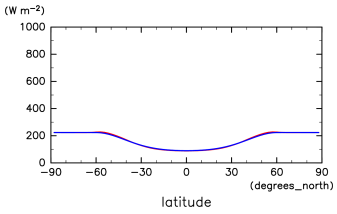
(b2)



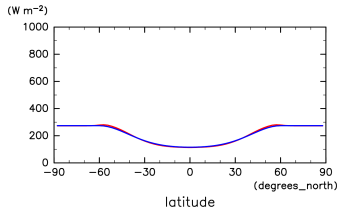
(b3)



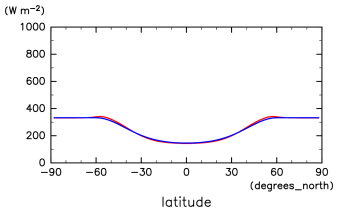
(c1)



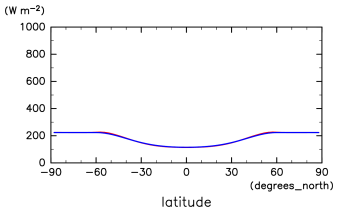
(c2)



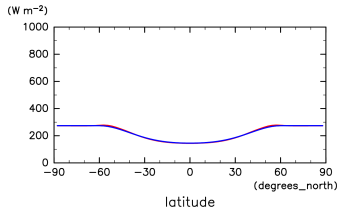
(c3)



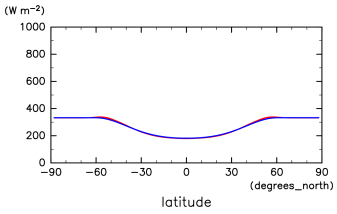
(d1)



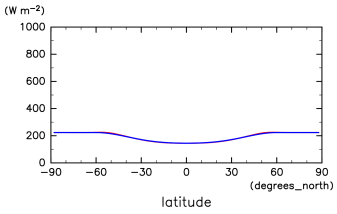
(d2)



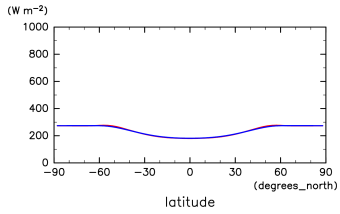
(d3)



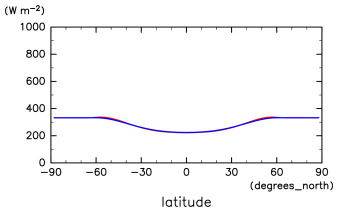
(e1)



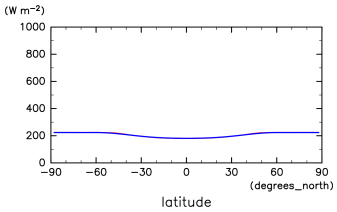
(e2)



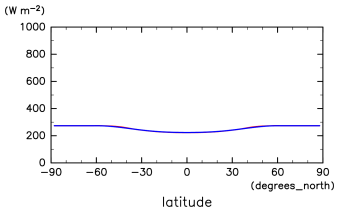
(e3)



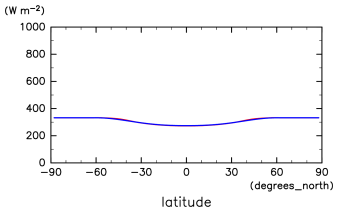
(f1)



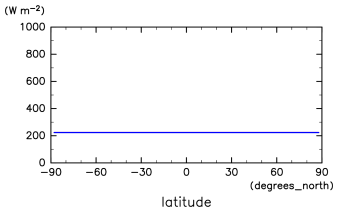
(f2)



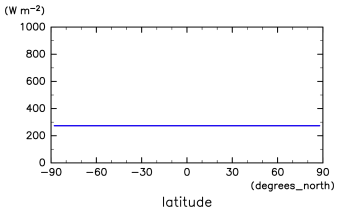
(f3)



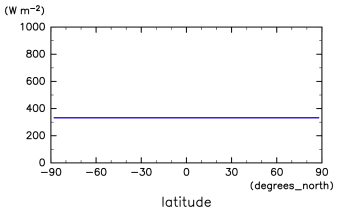
(g1)



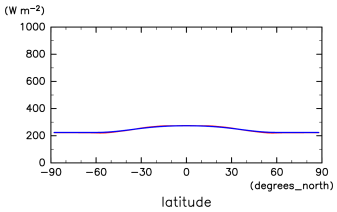
(g2)



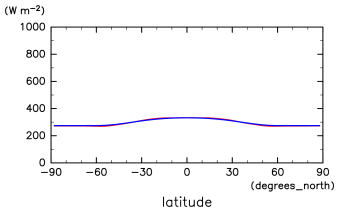
(g3)



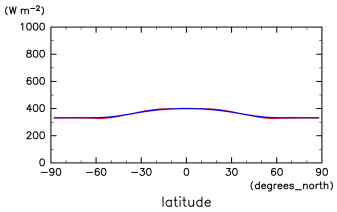
(h1)



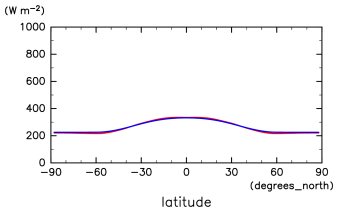
(h2)



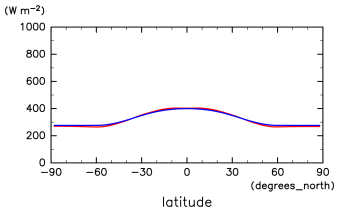
(h3)



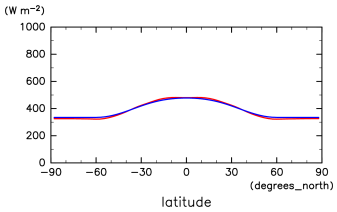
(i1)



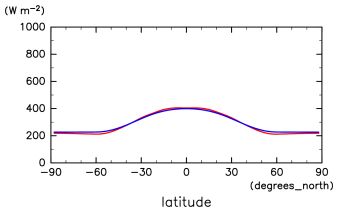
(i2)



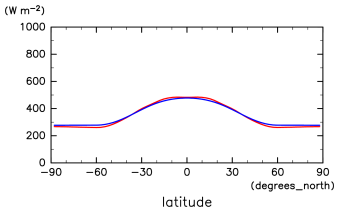
(i3)



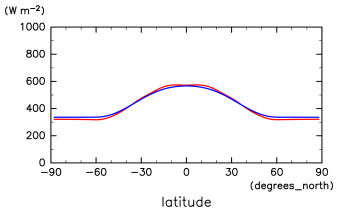
(j1)



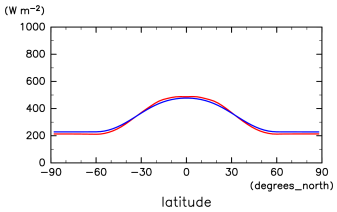
(j2)



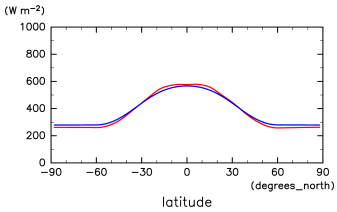
(j3)



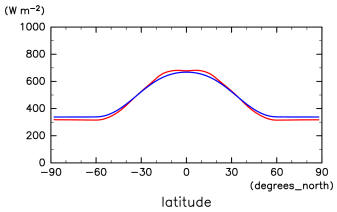
(k1)



(k2)



(k3)



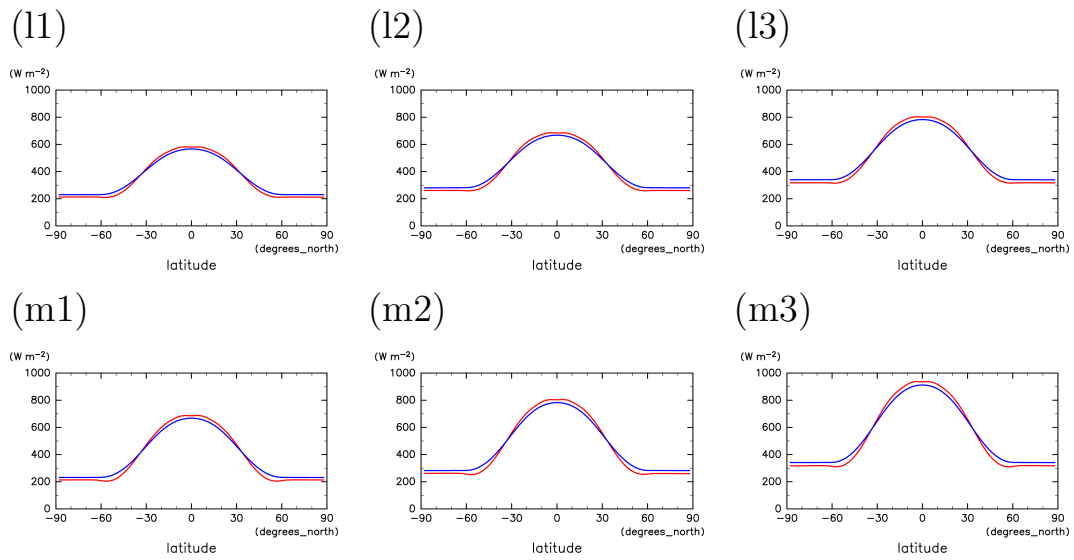
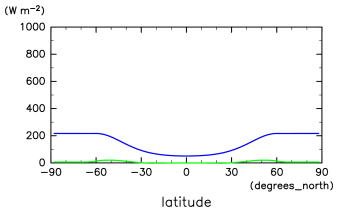


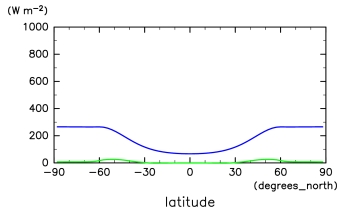
図 1.2: 大気の熱収支．横軸は緯度，単位は $\mathrm{W/m^2}$．赤は地面からの加熱 (地面から大気への長波放射による正味の熱フラックスと地面から大気への顕熱フラックス)，青は大気の放射による冷却 (大気から宇宙へ長波放射による熱フラックス)．アルファベットは ΔT：(a)-81K，(b)-67.5K，(c)-54K，(d)-40.5K，(e)-27K，(f)-13.5K，(g)0K，(h)13.5K，(i)27K，(j)40.5K，(k)54K，(l)67.5K，(m)81K，数字は T_{pole}：(1)259.65K，(2)273.15K，(3)286.65K．(a1) なら $\Delta T=-81$K，$T_{pole}=259.65$K となる．

1.1.3 地表の熱収支

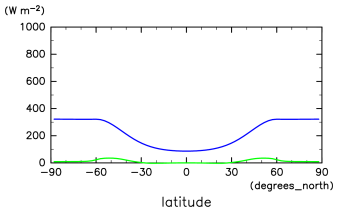
(a1)



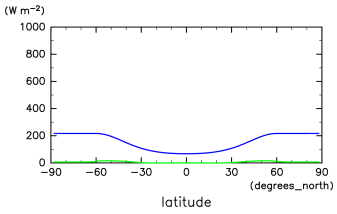
(a2)



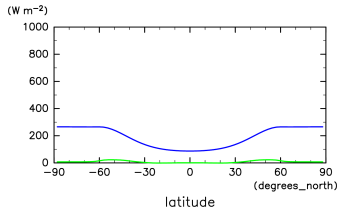
(a3)



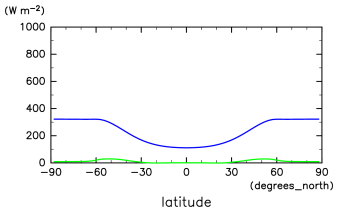
(b1)



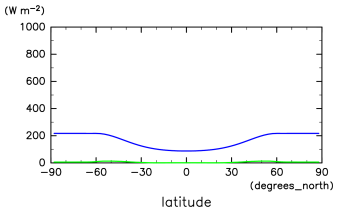
(b2)



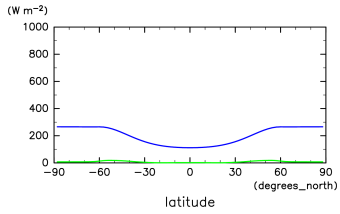
(b3)



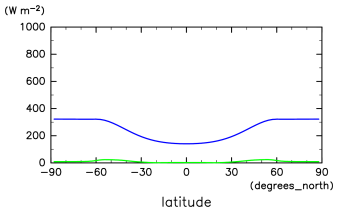
(c1)



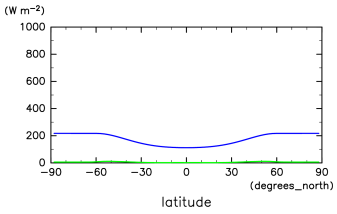
(c2)



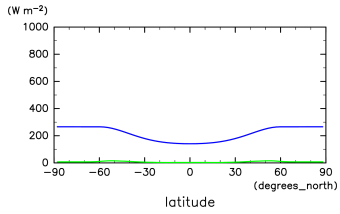
(c3)



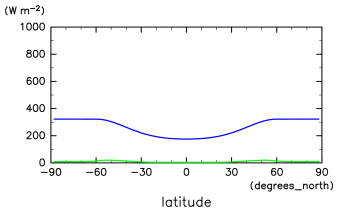
(d1)



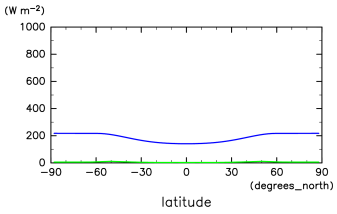
(d2)



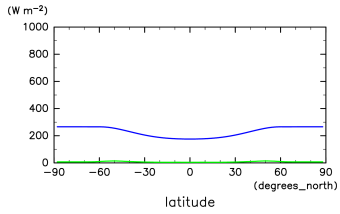
(d3)



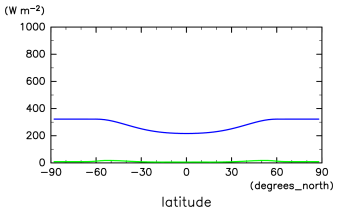
(e1)



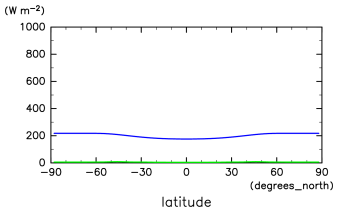
(e2)



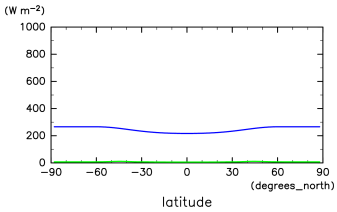
(e3)



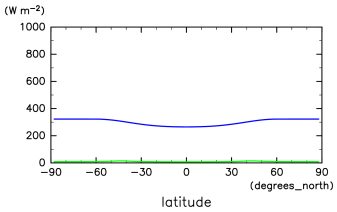
(f1)



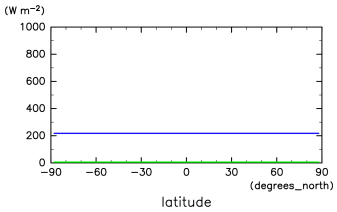
(f2)



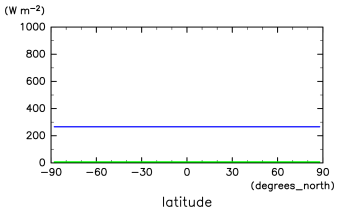
(f3)



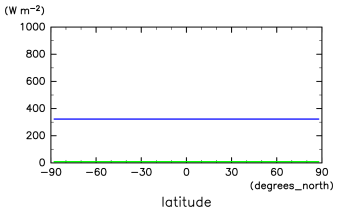
(g1)



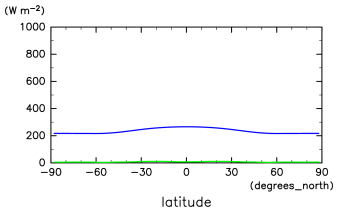
(g2)



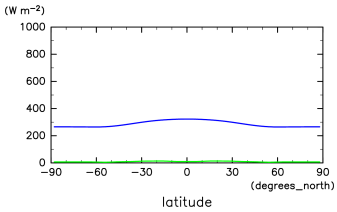
(g3)



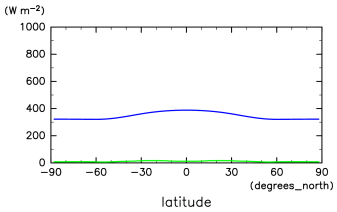
(h1)



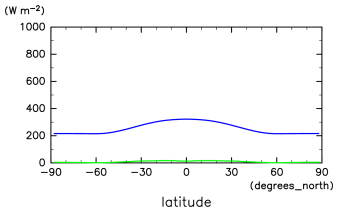
(h2)



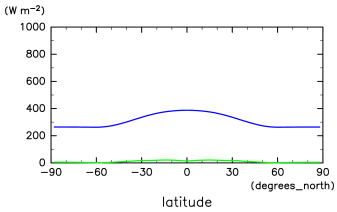
(h3)



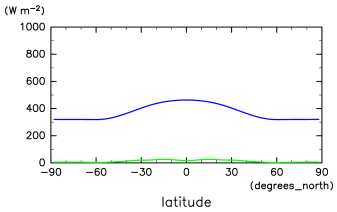
(i1)



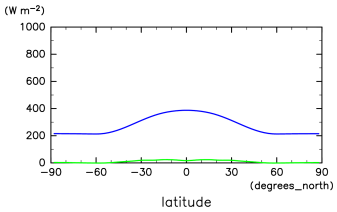
(i2)



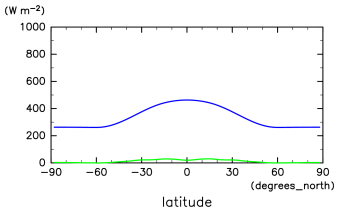
(i3)



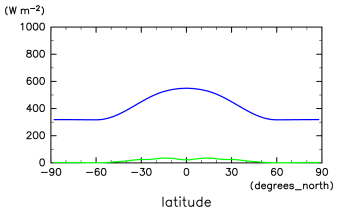
(j1)



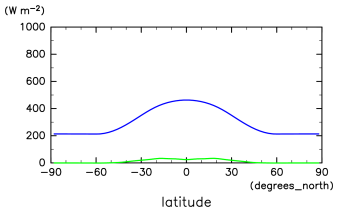
(j2)



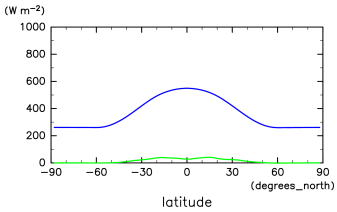
(j3)



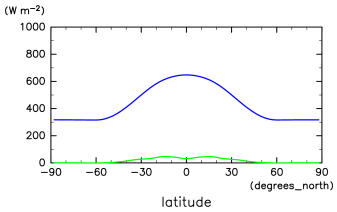
(k1)



(k2)



(k3)



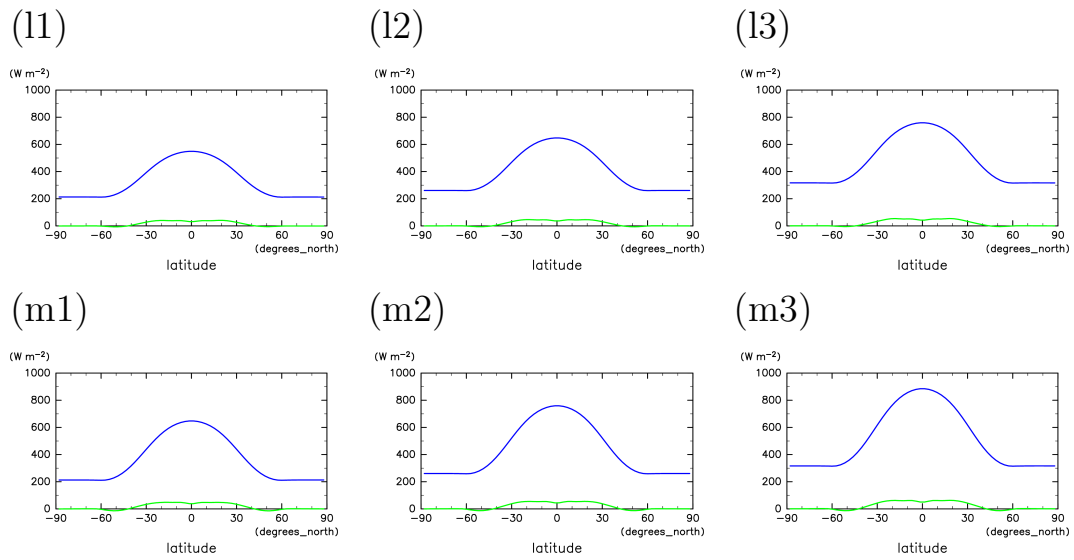


図 1.3: 地表面の熱収支．横軸は緯度，単位は $\mathrm{W/m^2}$．青は地面から大気への長波放射による正味の熱フラックス，緑は地面から大気への顕熱フラックス．アルファベットは ΔT：(a)−81K，(b)−67.5K，(c)−54K，(d)−40.5K，(e)−27K，(f)−13.5K，(g)0K，(h)13.5K，(i)27K，(j)40.5K，(k)54K，(l)67.5K，(m)81K，数字は T_{pole}：(1)259.65K，(2)273.15K，(3)286.65K．(a1) なら ΔT=−81K，T_{pole}=259.65K となる．

1.2 実験で使用したconfファイル

本研究の実験の際に用いたconfファイルを載せる.

Listing 1.1: confファイル

```
#= dcpam_main プログラム用NAMELIST ファイル(T42L26 用)
#
#= NAMELIST file for "dcpam_main"
#
# Copyright (C) GFD Dennou Club, 2008-2009. All rights reserved.
#
# Note that Japanese and English are described in parallel.
#/
&modify_albedo_snowseaice_nml
  FlagModAlbedoBasedOnTemp = .true.
/
&lscond_nml
  QH2OFlucRatio = 0.1
/
&mass_fixer_nml
  ThresholdForMessage = -1.0d-20
/
&sltt_nml
  FlagSLTTArcsineHor = .false.,
  FlagSLTTArcsineVer = .false.
/
&sltt_const_nml
  dtiw = 2,
  dtjw = 3
/
&constants_snowseaice_nml
  SnowAlbedo = 0.6d0,
  SeaIceAlbedo = 0.6d0,
  SnowThresholdForFlux = 10.0d0,
  SnowThresholdForAlbedo = 10.0d0,
/
&set_solarconst_nml
  SolarConst = 0.0d0,
/
&gwd_m1987_nml
  SigmaRef = 0.970d0,
  OrogEffWaveLength = 100.0d3
/
&saturate_nml
  SaturateWatIceFracType = 'Lin'
  TempIceLim = 253.15d0,
```

```
42   TempWatLim = 273.15d0
43 /
44 &dry_conv_adjust_nml
45   FlagAdjustMom = .true.,
46   FlagAdjustMR = .true.
47 /
48 &check_prog_vars_nml
49   TempMax = 400.0d0
50 /
51 &dcpam_main_nml
52     DynMode = 'HSPLVAS83',
53     PhysMode = 'FullPhysics',
54     RadModel = 'Earth',
55     SfcFluxMethod = 'BH91B94',
56     VDiffMethod = 'MY2.5',
57     PhysImpMode = 'SoilModel',
58     MCMethod = 'RASWithIce',
59     LSCMethod = 'LL91WithIce',
60     CloudMethod = 'SimpleWithIce',
61     SfcMoistMethod = 'Bucket',
62     GWDMethod = 'M1987',
63     DCMethod = 'DCA',
64     FlagSnow = .true.
65 /
66 &fileset_nml
67   FileTitle = 'Simulation of an atmosphere of planet with the
         land and the ocean',
68   FileSource = 'dcpam5 $Name: $ (http://www.gfd-dennou.org/
         library/dcpam)',
69   FileInstitution = 'GFD Dennou Club (http://www.gfd-dennou.org)'
70 /
71 &gridset_nml
72   nmax = 42,
73   imax = 128,
74   jmax = 64,
75   kmax = 26,
76   kslmax = 9,
77   ksimax = 9
78 /
79 &composition_nml
80   ncmax = 4,
81   Names = 'QH2OVap', 'QH2OLiq', 'QH2OSol', 'TKE',
82   FlagAdv = .true., .true., .true., .true.,
83   FlagMassFix = .true., .true., .true., .false.,
84   FlagVDiff = .true., .true., .true., .false.,
85 /
```

```
&timeset_nml
  cal_type = 'user_defined',
  month_in_year = 12,
  day_in_month = 30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30,
     30,
  hour_in_day = 24,
  min_in_hour = 60,
  sec_in_min = 60.0d0,
  InitialYear = 1,
  InitialMonth = 1,
  InitialDay = 1,
  InitialHour = 0,
  InitialMin = 0,
  InitialSec = 0.0d0,
  EndYear = 11,
  EndMonth = 1,
  EndDay = 1,
  EndHour = 0,
  EndMin = 0,
  EndSec = 0.0d0,
  DelTimeValue = 15.0,
  DelTimeUnit = 'min',
  PredictIntValue = 1.0,
  PredictIntUnit = 'day',
  CpuTimeMoniter = .true.
/
&restart_surftemp_io_nml
  InputFile = '00/rst_sfc_T0_0.nc'
  OutputFile = 'rst_sfc.nc'
/
&restart_file_io_nml
  InputFile = '00/rst.nc',
  OutputFile = 'rst.nc',
  IntValue = 360.0,
  IntUnit = 'day'
/
&relaxed_arakawa_schubert
  RASSupressFactor = 0.1d0,
  PRCPArea = 0.05d0,
  RainSnowConvFactor0 = 0.900d0,
  FlagZeroCrtlCWF = .true.
/
&cloud_simple_nml
  CloudCoverMethod = 'Const',
  PRCPArea = 1.0d0,
  PRCPEvapArea = 1.0d0,
```

```
  SnowMethod = 'StepPC',
  CloudWatLifeTime = 1980.0d0,
  CloudIceLifeTime = 9000.0d0,
/
&surface_properties_nml
  SurfTempSetting = 'file',
  SurfTempFile = '00/rst_sfc_T0_0.nc',
  SurfTempName = 'SurfTemp',
  SeaIceSetting = 'generate_internally',
  SurfTypeSetting = 'file',
  SurfTypeFile = '../data_Ts/sfcindex_0_P16.nc',
  SurfTypeName = 'sfcindex',
  SurfCondSetting = 'generate_from_SurfType',
  SurfCondFile = '',
  SurfCondName = '',
  SurfCulIntSetting = 'generate_internally',
  SurfHeightFile = 'generate_internally',
  SurfHeightStdSetting = 'generate_internally',
  SurfHeightStdName = 'generate_internally',
  AlbedoSetting = 'generate_internally',
  RoughLengthSetting = 'generate_internally',
/
&surface_data_nml
  RoughLength = 1.0d-4,
  SurfCond = 0
  HumidCoef = 0.0d0
/
&rad_short_income_nml
  FlagAnnualMean = .false.,
  FlagDiurnalMean = .false.,
  EpsOrb = 0.0d0,
  PerLonFromVE = 0.0d0,
  LonFromVEAtEpoch = 279.403308d0,
  Eccentricity = 0.0d0,
  TimeAtEpoch = -1.0d0,
  EpochYear = 1,
  EpochMonth = 1,
  EpochDay = 1,
  EpochHour = 0,
  EpochMin = 0,
  EpochSec = 0.0d0
/
&set_O3_nml
 FlagO3 = .false.,
/
&axesset_nml
```

```
177   Depth = 0.0d0, -4.0d-2, -12.0d-2, -28.0d-2, -40.0d-2, -64.0d
          -2, -112.0d-2, -208.0d-2, -400.0d-2, -784.0d-2,
178   SIDepth = 0.0d0, -1.0d0, -2.0d0, -3.0d0, -4.0d0, -5.0d0, -7.0
          d0, -10.0d0, -20.0d0, -50.0d0,
179   Sigma = 1.00000000 0.997503102 0.992528021 0.982652187
          0.963194370 0.927743435 0.893597305 0.829029083
          0.762362421 0.695695758 0.629029095 0.562362432
          0.495695770 0.429029107 0.362362444 0.295695782
          0.230288103 0.179348558 0.139676794 0.108780399
          8.47182572E-02 6.59786463E-02 5.13842218E-02 4.00180705E
          -02 3.11661046E-02 1.55830523E-02 0.00000000E+00
180 /
181 &dynamics_hspl_vas83_nml
182   FlagSLTT = .true.,
183   FlagSpongeLayer = .true.,
184   SLEFoldTimeValue = 10.0d0,
185   SLEFoldTimeUnit = 'day',
186   SLOrder = 1,
187   SLNumLayer = 5,
188   FlagDivDamp = .true.,
189   HDOrder = 4,
190   HDEFoldTimeValue = 1.0,
191   HDEFoldTimeUnit = 'day'
192 /
193 &gtool_historyauto_nml
194   IntValue = 1.0,
195   IntUnit = 'day',
196   Precision = 'float',
197   FilePrefix = '',
198 /
199 !
200
201 &gtool_historyauto_nml
202   Name = 'U, V, Temp, Ps, QH2OVap',
203   TimeAverage = .true.
204 /
205 &gtool_historyauto_nml
206   Name = 'QH2OLiq, QH2OSol'
207   TimeAverage = .true.
208 /
209 &gtool_historyauto_nml
210   Name = 'SigDot'
211   TimeAverage = .true.
212 /
213 &gtool_historyauto_nml
214   Name = 'SurfTemp'
```

```
215   TimeAverage = .true.
216 /
217 &gtool_historyauto_nml
218   Name = 'SoilMoist'
219   TimeAverage = .true.
220 /
221 &gtool_historyauto_nml
222   Name = 'SurfSnow'
223   TimeAverage = .true.
224 /
225 &gtool_historyauto_nml
226   Name = 'SoilTemp'
227   TimeAverage = .true.
228 /
229 &gtool_historyauto_nml
230   Name = 'Rain, Snow, PRCP',
231   TimeAverage = .true.
232 /
233 &gtool_historyauto_nml
234   Name = 'PRCPCum',
235   TimeAverage = .true.
236 /
237 &gtool_historyauto_nml
238   Name = 'PRCPLsc',
239   TimeAverage = .true.
240 /
241 &gtool_historyauto_nml
242   Name = 'Evap, Sens, OLR, SLR, OSR, SSR, ISR'
243   TimeAverage = .true.
244 /
245 &gtool_historyauto_nml
246   Name = 'EvapA, SensA, OLRA, SLRA, OSRA, SSRA, ISRA'
247   TimeAverage = .true.
248 /
249 &gtool_historyauto_nml
250   Name = 'SurfAlbedo'
251   TimeAverage = .true.
252 /
253 &gtool_historyauto_nml
254   Name = 'SeaIceConc'
255   TimeAverage = .true.
256 /
257 &gtool_historyauto_nml
258   Name = 'Mass, KinEngy, IntEngy, PotEngy, LatEngy, TotEngy,
        Enstro'
259   SpaceAverage = .true., .true., .true.,
```

```
260   Precision = 'double',
261 /
262 &gtool_historyauto_nml
263   Name = 'Decl'
264 /
265 &gtool_historyauto_nml
266   Name = 'SurfH2OVapFlux, SurfH2OVapFluxB'
267   TimeAverage = .true.
268 /
269 &gtool_historyauto_nml
270   Name = 'SurfH2OVapFluxU, EvapU'
271   TimeAverage = .true.
272 /
273 &gtool_historyauto_nml
274   Name = 'TKE'
275 /
276 &gtool_historyauto_nml
277   Name = 'OMG'
278 /
279 &gtool_historyauto_nml
280   Name = 'DTempDtRadS, DTempDtRadL'
281 /
282 &gtool_historyauto_nml
283   Name = 'PotTemp, SLP'
284 /
285 &gtool_historyauto_nml
286   Name = 'RadLUWFLXA, RadLDWFLXA, RadSUWFLXA, RadSDWFLXA'
287   TimeAverage = .true.
288 /
289 &gtool_historyauto_nml
290   Name = 'SOSeaIceTemp, SOSeaIceMass'
291   TimeAverage = .true.
292 /
```

1.3 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: set_solarconst_nml の namelist 変数

変数名	意味	変更後
SolarConst	太陽定数 [W/m^2]	0.0d0

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
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: surface_data_nml の namelist 変数

変数名	意味	変更後
SurfType	土地利用	0
RoughLength	地表粗度長	1.0d-4
SurfCond	地表状態	0
HumidCoef	地表湿潤度	0.0d0

1.4 図を描く際に使用したスクリプト

地表温度

ΔT=81 のときの地表温度の図を描く際に用いたスクリプトを載せる.

Listing 1.2: 地表温度の図

```
#! /usr/bin/ruby
require "numru/ggraph"
include NumRu

fn1 = '../T81_0/01/SurfTemp.nc'
fn2 = '../T81_p/01/SurfTemp.nc'
fn3 = '../T81_m/01/SurfTemp.nc'

gphys1 = GPhys::IO.open(fn1,vname).cut('time'=>2520..3600)
gp_gm1 = gphys1.mean('time').mean('lon')
gphys2 = GPhys::IO.open(fn2,vname).cut('time'=>2520..3600)
gp_gm2 = gphys2.mean('time').mean('lon')
gphys3 = GPhys::IO.open(fn3,vname).cut('time'=>2520..3600)
gp_gm3 = gphys3.mean('time').mean('lon')

DCL.gropn(2)
DCL.sgpset('lcntl', false)
DCL.uzfact(1.2)
DCL.sgpset('lfprop',true)
DCL.sgscmn(73)

GGraph.set_fig 'itr'=>1
GGraph.set_axes('xlabelint'=>30)
GGraph.line( gp_gm1,true,'max'=>370,'min'=>170,'annotate'=>false
    ,'index'=>594,'title'=>'')
GGraph.line( gp_gm2,false,'index'=>974)
GGraph.line( gp_gm3,false,'index'=>864)
DCL.grcls
```

気温

ΔT=81，T_{pole}=286.65 のときの気温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 1.3: 気温の子午面断面図

```
#! /usr/bin/ruby
require "numru/ggraph"
include NumRu

fn1 = '../T81_p/01/Temp.nc'
vname1 = 'Temp'

gp = GPhys::IO.open(fn1,vname1).cut('time'=>3240..3600)
gp = gp.mean('time').mean('lon')

DCL.gropn(2)
DCL.sgpset('lcntl', false)
DCL.sgpset('lfull',true)
DCL.uzfact(0.83)
DCL.sgpset('lfprop',true)
DCL.sgscmn(10)

GGraph.set_fig 'itr'=>1, 'viewport'=>[0.2,0.8,0.2,0.5], 'window
    '=>[-90,90,1.0,0.0]
GGraph.set_axes('xlabelint'=>30)
GGraph.tone( gp,true,'lev
    '=>[105,120,135,150,165,180,195,210,225,240,255,270,285,300,315,330,345,36
    title'=>'','annotate'=>false)
GGraph.color_bar
GGraph.contour(gp,false,'interval'=>20,'index'=>10)
DCL.grcls
```

南北温度傾度

ΔT=81，T_{pole}=286.65 のときの南北温度傾度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 1.4: 南北温度傾度の子午面断面図

```
#! /usr/bin/ruby
require "numru/ggraph"
include NumRu

fn1 = '../T81_p/01/Temp.nc'
vname1 = 'Temp'

gp = GPhys::IO.open(fn1,vname1).cut('time'=>3240..3600)
gp = gp.mean('time').mean('lon')

lat = gp.axis("lat").pos
lat_na = lat.val
t = gp.val

dlat = lat_na[1..-1] - lat_na[0..-2]
dT = t[ 1..-1, true ] - t[ 0..-2, true ]

lat_mid = 0.5 * (lat_na[1..-1] + lat_na[0..-2])

orig_grid = gp.grid
sigma_pos = orig_grid.axis("sig").pos
sigma_pos = sigma_pos.val if sigma_pos.is_a?(NumRu::VArray)
sigma_axis = NumRu::Axis.new
sigma_axis.set_pos(NumRu::VArray.new(sigma_pos, {"long_name"=>"
    sigma"}, "sig"))

lat_axis = NumRu::Axis.new
lat_axis.set_pos(NumRu::VArray.new(lat_mid, {"long_name"=>"
    latitude","units"=>"degrees_north"}, "lat"))

grid = NumRu::Grid.new(lat_axis, sigma_axis)
dT_va = NumRu::VArray.new(dT,{"long_name"=>"meridional
    temperature difference"},"dT")
dT_gp = GPhys.new(grid, dT_va)

lat_mid = dT_gp.axis("lat").pos.val
dT_val = dT_gp.val

south_idx = lat_mid.lt(0.0).where

```

```
38 south_idx.each do |i|
39   dT_val[i, true] = - dT_val[i, true]
40 end
41
42 dT_val = - dT_val
43
44 dT_gp = GPhys.new(dT_gp.grid,NumRu::VArray.new(dT_val,{"long_name
       "=>"dT/d(lat)"},dT_gp.name))
45
46 DCL.gropn(2)
47 DCL.sgpset('lcntl', false)
48 DCL.sgpset('lfull',true)
49 DCL.uzfact(0.83)
50 DCL.sgpset('lfprop',true)
51 DCL.sgscmn(65)
52
53 GGraph.set_fig 'itr'=>1, 'viewport'=>[0.2,0.8,0.2,0.5], 'window
       '=>[-90,90,1.0,0.0]
54 GGraph.set_axes('xlabelint'=>30)
55 GGraph.tone( dT_gp,true,'title'=>'','lev
       '=>[-7.0,-6.0,-5.0,-4.0,-3.0,-2.0,-1.0,0.0,1.0,2.0,3.0,4.0,5.0,6.0,7.0],'
       annotate'=>false)
56 GGraph.color_bar
57 DCL.grcls
```

東西風速

ΔT=81，T_{pole}=286.65 のときの東西風速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 1.5: 東西風速の子午面断面図

```
#!/Usr/bin/ruby
require "numru/ggraph"
include NumRu

fn1 = '../T81_p/01/U.nc'
vname1 = 'U'

gp = GPhys::IO.open(fn1,vname1).cut('time'=>3240..3600)
gp = gp.mean('lon').mean('time')

DCL.gropn(2)
DCL.sgpset('lcntl', false)
DCL.sgpset('lfull',true)
DCL.uzfact(0.83)
DCL.sgpset('lfprop',true)
DCL.sgscmn(65)

GGraph.set_fig 'itr'=>1, 'viewport'=>[0.2,0.8,0.2,0.5], 'window
    '=>[-90,90,1.0,0.0]
GGraph.set_axes('xlabelint'=>30)
GGraph.tone( gp,true,'lev
    '=>[-100,-90,-80,-70,-60,-50,-40,-30,-20,-10,0,10,20,30,40,50,60,70,80,90,
    title'=>'','annotate'=>false)
GGraph.color_bar
GGraph.contour( gp,false,'interval'=>30,'index'=>10)
DCL.grcls
```

質量流線関数

ΔT=81，T_{pole}=286.65 のときの子午面の質量流線関数の図を描く際に用いたスクリプトを載せる．

Listing 1.6: 子午面の質量流線関数の図

```
#!/Usr/bin/ruby
require "numru/ggraph"
include NumRu
include NMath

fn1 = '../T81_p/01/V.nc'
fn2 = '../T81_p/01/Ps.nc'
vname1 = 'V'
vname2 = 'Ps'

v = GPhys::NetCDF_IO.open(fn1, vname1).cut('time'=>3240..3600)
v = v.mean('lon').mean('time')
ps = GPhys::NetCDF_IO.open(fn2, vname2).cut('time'=>3240..3600)
ps = ps.mean('lon').mean('time')

g = 9.8
rplanet = 6.371e6

lat = v.coord("lat").val * Math::PI/180.0
sigm = GPhys::NetCDF_IO.open(fn1,"sigm").val
nlat = lat.length
nlev = sigm.length

ps_lat = ps.val.reshape(nlat,1)
vval = v.val

psi = NArray.sfloat(nlat, nlev)

if sigm[0] < sigm[-1]
  psi[true,0] = 0.0
  k = 0
  while k < nlev-1
    dsig = sigm[k+1] - sigm[k]
    term = vval[true,k] * ps_lat[true,0] * dsig / g
    psi[true,k+1] =
      psi[true,k] + term * (2*Math::PI*rplanet*NMath.cos(lat))
    k += 1
  end
else
  psi[true,nlev-1] = 0.0
```

```
41   k = nlev - 2
42   while k >= 0
43     dsig = sigm[k] - sigm[k+1]
44     term = vval[true,k] * ps_lat[true,0] * dsig / g
45     psi[true,k] =
46       psi[true,k+1] + term * (2*Math::PI*rplanet*NMath.cos(lat))
47     k -= 1
48   end
49 end
50
51 psi = psi[true,1..-1]
52 msf = v.copy
53 msf[true,true] = psi * 1e-9
54 msf.long_name = ""
55 msf.units = "1e9 kg/s"
56
57 DCL.gropn(2)
58 DCL.sgpset('lcntl', false)
59 DCL.sgpset('lfull',true)
60 DCL.uzfact(0.83)
61 DCL.sgpset('lfprop',true)
62 DCL.sgscmn(65)
63
64 GGraph.set_fig 'itr'=>1, 'viewport'=>[0.2,0.8,0.2,0.5], 'window
       '=>[-90,90,1.0,0.0]
65 GGraph.set_axes('xlabelint'=>30)
66 GGraph.tone( msf, true , 'lev
       '=>[-360,-330,-300,-270,-240,-210,-180,-150,-120,-90,-60,-30,0,30,60,90,12
       annotate'=>false)
67 GGraph.color_bar
68 GGraph.contour(msf,false,'interval'=>30,'index'=>10,'annotate'=>
       false)
69 DCL.grcls
```

南北熱輸送量

ΔT=81，T_{pole}=286.65 のときの南北熱輸送量の図を描く際に用いたスクリプトを載せる．

Listing 1.7: 南北熱輸送量の図 label

```
#!/usr/bin/ruby
require "numru/ggraph"
include NumRu

fn1 = '../T81_p/01/SLRA.nc'
vname1 = 'SLRA'
fn2 = '../T81_p/01/SensA.nc'
vname2 = 'SensA'
fn3 = '../T81_p/01/OLRA.nc'
vname3 = 'OLRA'

gphys1 = GPhys::IO.open(fn1,vname1).cut('time'=>3240..3600)
gphys1 = gphys1.mean('time')
gphys2 = GPhys::IO.open(fn2,vname2).cut('time'=>3240..3600)
gphys2 = gphys2.mean('time')
gphys3 = GPhys::IO.open(fn3,vname3).cut('time'=>3240..3600)
gphys3 = gphys3.mean('time')
gplat = GPhys::IO.open(fn1,'lat')
gplatw = GPhys::IO.open(fn1,'lat_weight')
nlat = gplat.shape[0]

weight = gplatw.val
weight = weight / weight.sum(0)

gphysy1 = gphys1.mean('lon')
gphysy1 = (gphysy1*gplatw)/(gplatw.sum)
gphysy2 = gphys2.mean('lon')
gphysy2 = (gphysy2*gplatw)/(gplatw.sum)
gphysy3 = gphys3.mean('lon')
gphysy3 = (gphysy3*gplatw)/(gplatw.sum)
gphysy4 = gphysy1 + gphysy2
gphysy4 = gphysy4*(- gphysy3.sum(0)/gphysy4.sum(0))
total = -(gphysy4+gphysy3)

radius = 6.371e6
pi = 3.14
surface = 4*pi*radius*radius
total = total*surface/10**15

flux = NArray.float(nlat-1)
```

```
41 lat = NArray.float(nlat-1)
42 for j in 1..(nlat-1)
43   flux[j-1] = total[0..j-1].sum('lat')
44   lat[j-1] = (gplat[j-1].val+gplat[j].val)/2.0
45 end
46 latax = VArray.new(lat, {'long_name'=>'latitude','units'=>'
       degrees_north'}, 'lat')
47 axis0 = Axis.new.set_pos(latax)
48 gflux = VArray.new(flux,{'long_name'=>'heat flux','units'=>'PW
       '}, 'flux')
49 gpflux = GPhys.new( Grid.new(axis0), gflux )
50
51 DCL.gropn(2)
52 DCL.sgpset('lcntl', false)
53 DCL.sgpset('lfull',true)
54 DCL.uzfact(0.83)
55 DCL.sgpset('lfprop',true)
56
57 GGraph.set_fig 'itr'=>1, 'viewport'=>[0.2,0.8,0.2,0.5], 'window
       '=>[-90,90,-3.6,3.6]
58 GGraph.set_axes('xlabelint'=>30,'ytickint'=>0.3)
59 GGraph.line(gpflux,true,'index'=>14,'title'=>'')
60 DCL.grcls
```

温位

ΔT=81，T_{pole}=286.65 のときの温位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 1.8: 温位の子午面断面図

```
#! /usr/bin/ruby
require "numru/ggraph"
include NumRu

fn1 = '../T81_p/01/PotTemp.nc'
vname1 = 'PotTemp'

gp = GPhys::IO.open(fn1,vname1).cut('time'=>3240..3600)
gp = gp.mean('time').mean('lon')

DCL.gropn(2)
DCL.sgpset('lcntl', false)
DCL.sgpset('lfull',true)
DCL.uzfact(0.83)
DCL.sgpset('lfprop',true)
DCL.sgscmn(10)

GGraph.set_fig 'itr'=>1, 'viewport'=>[0.2,0.8,0.2,0.5], 'window
    '=>[-90,90,1.0,0.0]
GGraph.set_axes('xlabelint'=>30)
GGraph.tone( gp,true,'lev
    '=>[150,165,180,195,210,225,240,255,270,285,300,315,330,345,360,375,390,40
    title'=>'','annotate'=>false)
GGraph.color_bar
GGraph.contour(gp,false,'interval'=>20,'index'=>10)
DCL.grcls
```

大気の熱収支

ΔT=81，T_{pole}=286.65 のときの大気の熱収支の図を描く際に用いたスクリプトを載せる．

Listing 1.9: 大気の熱収支の図

```
#!/usr/bin/ruby
require "numru/ggraph"
include NumRu

fn1 = '../T81_p/01/SLRA.nc'
vname1 = 'SLRA'
fn2 = '../T81_p/01/SensA.nc'
vname2 = 'SensA'
fn3 = '../T81_p/01/OLRA.nc'
vname3 = 'OLRA'

gphys1 = GPhys::IO.open(fn1,vname1).cut('time'=>3240..3600)
gp_gm1 = gphys1.mean('time').mean('lon')
gphys2 = GPhys::IO.open(fn2,vname2).cut('time'=>3240..3600)
gp_gm2 = gphys2.mean('time').mean('lon')
gp_gm3 = gp_gm1 + gp_gm2

gphys4 = GPhys::IO.open(fn3,vname3).cut('time'=>3240..3600)
gp_gm4 = gphys4.mean('time').mean('lon')

gp_gm5 = gp_gm3-gp_gm4

DCL.gropn(2)
DCL.sgpset('lcntl', false)
DCL.sgpset('lfull',true)
DCL.uzfact(0.83)
DCL.sgpset('lfprop',true)

GGraph.set_fig 'itr'=>1, 'viewport'=>[0.2,0.8,0.2,0.5], 'window
    '=>[-90,90,0,1000]
GGraph.set_axes('ytitle'=>'','xlabelint'=>30)
GGraph.line( gp_gm3,true,'index'=>24,'title'=>'','annotate'=>
    false)
GGraph.line( gp_gm4,false,'index'=>44,'annotate'=>false)
DCL.grcls
```

地表の熱収支

ΔT=81，T_{pole}=286.65 のときの地表の熱収支の図を描く際に用いたスクリプトを載せる．

Listing 1.10: 地表の熱収支の図

```
#!/usr/bin/ruby
require "numru/ggraph"
include NumRu

fn1 = '../T81_p/01/SLRA.nc'
vname1 = 'SLRA'
fn2 = '../T81_p/01/SensA.nc'
vname2 = 'SensA'

gphys1 = GPhys::IO.open(fn1,vname1).cut('time'=>2520..3600)
gp_gm1 = gphys1.mean('time').mean('lon')
gphys2 = GPhys::IO.open(fn2,vname2).cut('time'=>2520..3600)
gp_gm2 = gphys2.mean('time').mean('lon')
gp_gm3 = gp_gm1 + gp_gm2

DCL.gropn(2)
DCL.sgpset('lcntl', false)
DCL.sgpset('lfull',true)
DCL.uzfact(0.83)
DCL.sgpset('lfprop',true)

GGraph.set_fig 'itr'=>1, 'viewport'=>[0.2,0.8,0.2,0.5], 'window
    '=>[-90,90,0,1000]
GGraph.set_axes('ytitle'=>'','xlabelint'=>30)
GGraph.line( gp_gm1,true,'index'=>44,'title'=>'','annotate'=>
    false)
GGraph.line( gp_gm2,false,'index'=>34,'annotate'=>false)
DCL.grcls
```

参考文献

[1] 高橋 芳幸, 樫村 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] Toon, O. B., C. P. McKay, and T. P. Ackerman, 1989: Rapid calculation of radiative heating rates and photodissociation rates in inhomogeneous multiple scattering atmospheres, *J. Geophys. Res.*, 94, 16287-16301.

[3] Chou, M.-D., Suarez, m. J., Liang, X.-Z., and Yan, M. M.-H, 2001: A thermal infrared radiation parameterization for atmospheric studies, *NASA Tech. Memo.*, 19

[4] Mellor, G. L., and T. Yamada, 1974: A hierarchy of turbulence closure models for planetary boundary layers, *J. Atmos. Sci.*, 31, 1791-1806.

[5] Mellor, G. L., and T. Yamada, 1982: Development of a tubulent closure model for geophysical fluid problem, *Rev. Geophys. Space phys.*, 20, 851-875.

[6] Manabe, S., Smagorinsky, J., Strickler, R.F., 1965: Simulated climatology of a general circulation model with a hydrologic cycle, *Mon. Weather Rev.*, 93, 769-798.

[7] Neale, R. B., and B. J. Hoskins, 2001: A standard test for AGCMs including their physical parameterizations: I: The proposal. *Atmos sci. Lett.*, 1, 101-107