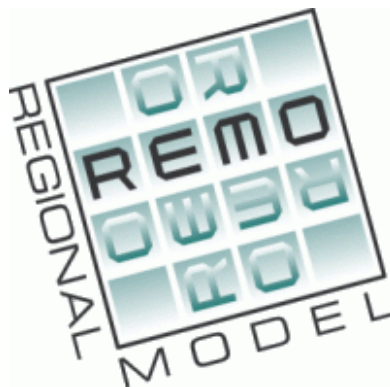


# HOW2REMO

---

Dokumentation zum  
REMO Workshop am Max-Planck-Institut für Meteorologie



Katharina Bülow, Stephanie Fiedler, Ksenia Glushak, Laura  
Niederdrenk, Ralf Podzun, Swantje Preuschmann, Christine  
Radermacher, Thomas Raub, Armelle Remedio, Sahar Sodoudi,  
Christof Wilhelm

Hamburg  
Stand Juli 2010

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>3</b>
<b>2</b>	<b>Die Bodenbibliothek</b>	<b>6</b>
2.1	Vorbereitung . . . . .	6
2.2	Die Erstellung der Bodenbibliothek . . . . .	7
<b>3</b>	<b>Die Antriebsdaten</b>	<b>10</b>
3.1	Vorbereitung . . . . .	10
3.2	Das Kompilieren des Präprozessor . . . . .	12
3.3	Der Druckinterpolator der Antriebsdaten . . . . .	12
3.4	Monatsmittel der Inputdaten . . . . .	13
3.5	Jobskript für die ausführbaren Dateien . . . . .	13
3.6	Gleichgewichtszustand des Bodens . . . . .	15
<b>4</b>	<b>Das REMO</b>	<b>16</b>
4.1	REMO kompilieren . . . . .	17
4.2	REMO Jobskript . . . . .	18
4.3	Restart . . . . .	20
4.4	Doppelnesting . . . . .	21
<b>5</b>	<b>Postprozessing</b>	<b>23</b>
5.1	Druckinterpolator der REMO-Outputdaten . . . . .	23
5.2	Bearbeitung von großen Datensätzen . . . . .	23

# Kapitel 1

## Einleitung

Diese Anleitung gibt einen Überblick über die notwendigen Prozesse für das Aufsetzen von Experimenten mit dem REgionalen KlimaMOdell REMO. Die wesentlichen Prozesse sind das Erstellen der Bodenbibliothek, die Interpolation der Antriebsdaten, das Kompilieren von REMO sowie die Joberstellung für das Modellexperiment. Bevor diese Schritte bearbeitet werden, ist es empfehlenswert einen Experimentplan aufzustellen. Die folgenden Überlegungen sollten dabei berücksichtigt werden, wenn man mit REMO rechnen möchte:

WO möchte ich rechnen:

- Wo befindet sich das Gebiet, in dem ich mit REMO rechnen möchte? Die Gebietswahl bestimmt, ob das Modellgebiet rotiert wird. Wenn das Experiment mit vorhandenen Läufen verglichen werden soll, ist es sinnvoll eine existierende Rotation zu übernehmen. Wenn es sich um ein noch nicht gerechnetes Modellgebiet handelt, müssen die reelle Länge und die Breite des rotierten Nord- bzw. Südpols und der linken unteren Gitterbox des Einzugsgebietes im rotierten Koordinatensystem berechnet werden. Liegt das gewünschte Gebiet zwischen 40°N und 40°S, ist keine Rotation nötig.
- Auf welchem Computer möchte ich rechnen? Verschiedene Großrechner verwenden unterschiedliche Compiler. Hierbei wird zwischen massiv parallelen Rechnern und Vektorcomputern unterschieden. Für beide Arten gibt es unterschiedliche REMO Versionen. Die aktuelle Version *REMO2009\_MPI* ist für massiv parallele Rechner (IBM) konzipiert, wofür derzeit keine nichthydrostatische Version verfügbar ist. Die nichthydrostatische REMO Version kann nur auf einem Vektorcomputer (NEC) gerechnet werden. Die Supercomputer produzieren aus rechentechnischen Gründen unterschiedliche Ergebnisse, was beim Vergleich der Ergebnisse zu berücksichtigen ist.

WIE möchte ich rechnen:

- In welcher Auflösung möchte ich rechnen? Brauche ich mehr als eine Nestung? Ein mehrfaches Nesten wird eingesetzt, um große (Faktor 10) Skalensprünge von den antreibenden Modellergebnissen auf das REMO Gitter zu vermeiden. Beispielfhaft kann ein REMO Experiment mit 10km horizontaler Auflösung mit Globalmodell-daten in T106 Auflösung angetrieben werden, wohingegen bei gröberer Auflösung der Antriebsdaten ein Doppelnesten mit einer Zwischenstufe von 50km horizontaler Auflösung empfohlen wird. Dabei ist der Antrieb für die erste Nestungsstufe das Globalmodell z.B. ECHAM5, wohingegen für die zweite, feinere Nestung die Ergebnisse aus der ersten REMO Simulation als Antrieb verwendet werden. Die horizontale Auflösung und die Wahl des Antriebs bestimmt demnach die Anzahl der Nestungsstufen. Die vertikale Aufteilung in REMO besteht standardmäßig aus 27 Leveln und ist von der vertikalen Auflösung der Antriebsdaten weitestgehend unabhängig. Optional kann mit bis zu 49 Leveln gerechnet werden.
- Welche REMO-Version ist für meine Anwendung optimal? Ralf Podzun ist für diese Frage der beste Ansprechpartner.
- Habe ich die richtige REMO-Version? Kleine Änderungen in der Modellversion, z.B. eine probeweise Änderung des Zeitschritts, können Fehler produzieren, die auch dem erfahrenen REMO-Nutzer unterlaufen und nicht unbedingt sofort auffallen.

WAS möchte ich rechnen:

- Welche Art von Experiment möchte ich durchführen? Es wird prinzipiell zwischen Szenario-, Kontroll- und Validierungslauf unterschieden. Kontroll- und Szenarioexperimente werden mit Anfangs- und Randwerten aus den Ergebnissen eines globalen Klimamodells angetrieben, was für REMO in der Regel aus der Ausgabe von ECHAM5 aufbereitet wird. Hierbei wird der Kontrolllauf (Vergangenheit) mit beobachteten Treibhausgaskonzentrationen angetrieben, während Szenarioläufe (Zukunft) Treibhausgaskonzentrationen verwenden, die auf verschiedenen Annahmen hinsichtlich der Entwicklung verschiedener sozioökonomischer Faktoren beruhen (Details siehe IPCC-SRES). Standardszenarien, die in REMO Experimenten bislang höher aufgelöst wurden, sind die Szenarien A1B, A2 und B1. Im Gegensatz dazu ist ein Validierungslauf durch Reanalyseantrieb, z.B. ERA-40 und ERA-Interim, und durch Antrieb mit operationellen Daten gekennzeichnet. Die Experimentart entscheidet welcher Antriebsdatensatz benötigt wird und somit welche Programme zur Erzeugung des Antriebs verwendet werden.



Anhand des Beispiels eines Kontrolllaufs mit ECHAM5-Antrieb wird im Folgenden schrittweise das Aufsetzen eines REMO Experiments auf dem DKRZ-Großrechner **blizzard** erklärt. Weiterführende Informationen können im ThyREMO Wiki in englischer Sprache nachgelesen werden. Zum Arbeiten loggt man sich zunächst von einer Konsole auf dem Supercomputer ein:

**Befehle für die Konsole:**

```
> ssh -X blizzard
```

Der Zugang kann beim DKRZ beantragt werden. Die Verzeichnisstruktur des Rechners ist unter [http://www.dkrz.de/dkrz/services/P6\\_docs/P6-howto5#Filesystems](http://www.dkrz.de/dkrz/services/P6_docs/P6-howto5#Filesystems) erklärt, wobei darauf zu achten ist, welche Daten im System gesichert werden. Für den Zugriff auf die Jobqueue der **blizzard** und die Rechenzeitabrechnung ist eine Gruppenzuweisung erforderlich. Weiterhin wird zur Kennzeichnung der Experimente jedem Anwender von REMO eine eigene Nutzernummer zugewiesen. Beide Vorgänge werden durch Ralf Podzun koordiniert.

# Kapitel 2

## Die Bodenbibliothek

### 2.1 Vorbereitung

- Man kopiere sich
  - *bodlib\_kit.tar.gz* (zur Erzeugung der Bodenbibliothek) und
  - *gtopo.tar.gz* (Informationen zur Topographie)von
  - der **blizzard** unter dem Gruppenordner: *\$ralf = /work/mh0140/k204009/libgen* oder
  - dem **archive** mit *pftp* unter */hpss/arch/mh0140/k204009/bodlib*in ein eigenes Verzeichnis und entpackt es (z.B. *gunzip, tar -xvf*)
- In seinem eigenem Verzeichnis (z.B. im */work/\$gruppe/\$userid* auf der **blizzard**) kopiert man das Verzeichnis *BODLIB* und benennt es nach dem geplanten Experiment (z.B. *\$Gebiet\_\$Aufloesung*).

#### Befehle für die Konsole:

```
> mkdir /work/$gruppe/$userid/bodenbib_verzeichnis
> cd /work/$gruppe/$userid/bodenbib_verzeichnis
> cp $ralf/bodlib_kit.tar.gz .
> cp $ralf/gtopo.tar.gz .
> gunzip bodlib_kit.tar.gz
> gunzip gtopo.tar.gz
> tar -xvf bodlib_kit.tar
> tar -xvf gtopo.tar
> cp -p -r BODLIB $Gebiet_$Aufloesung
```

- Ist die Datei *bodlib\_kit.tar.gz* entpackt, existiert eine schrittweise Erklärung zur Erstellung der Bodenbibliothek in *Anleitung.txt* oder in *README*.

**HILFREICH:**

Um bei Skriptaufrufen *./* weglassen zu können und um die Tools von Ralf korrekt ausführen zu können, ist folgende Einstellung nötig:

```
> set path = ( $path . /pf/k/k204009/bin )
```

Man kann diesen Befehl in der *~/.cshrc* und/oder *~/.login* auf der *blizzard* eintragen. Ansonsten ist dieser Befehl stets in einem neuen Fenster auf der *blizzard* einzugeben. Weiterhin muss für die Erstellung der Bodenbibliothek der Befehl *unlimit* eingegeben werden, damit das Skript *first.sh* korrekt arbeitet.

## 2.2 Die Erstellung der Bodenbibliothek

Folgende Schritte werden danach durchgeführt:

- Der Pol ist so zu verschieben, dass der Äquator durch die Mitte des Untersuchungsgebietes verläuft. Wie lauten die Koordinaten des Mittelpunktes der linken unteren Gitterbox des gewünschten Gebietes in rotierten Koordinaten? Eine Rotation des Gebietes ist nicht nötig, wenn das Gebiet in realen Koordinaten zwischen 40°N und 40°S liegt.
- Mit *gebtest.sh* sollte man die gewählte Rotation des Ausgangsgebiets testweise grafisch darstellen. Entsprechen der Gebietsausschnitt und die Lage des Gebietes den Vorstellungen, werden die zugehörigen digitalen Geländedaten (Topographie) aus */gtopo* herausgesucht. Eine Übersicht der Karten findet man in *gtopo30\_tiles.gif*. (Auf der **blizzard** kann ein .gif z.B. mit *display* angesehen werden) Die benötigten *.DEM* Files müssen in den Unterordner *\$Gebiet\_ \$Aufloesung/gtopo* des Experimentordners kopiert werden.

*first.sh* wird an folgenden Stellen angepasst:

```
EXP=           # Name des Gebiets
IWATER=1       # 1 für fraktionale Land-See-Maske, 0=nichtfraktionell
RESOLUTION_X=  # Anzahl der Gitterpunkte in x-Richtung (Länge)
RESOLUTION_Y=  # Anzahl der Gitterpunkte in y-Richtung (Breite)
RESOLUTION=    # Auflösung des Gebiets
LAT_LL=        # Breite des Mittelpunkts der linken unteren Gitterbox (rotiert)
LON_LL=        # Länge des Mittelpunkts der linken unteren Gitterbox (rotiert)
LAT_SP=        # Breite des rotierten Südpols (real)
LON_SP=        # Länge des rotierten Südpols (real)
POL=0          # Pol im Gebiet: 0=nein, 1=NP, 2=SP
IGLAC=0        # Gletscherschalter 0=default, 1=ja (Schalterbenutzung abklären)
```

#### ACHTUNG:

Anzahl der Gitterboxen in *first.sh* sind um zwei Gitterboxen größer als die REMO-Output Daten, da hier der Rand miteingerechnet werden muss!

#### ACHTUNG:

Der Nordpol hat immer die geografische Länge 180°, der Südpol hat immer die Länge 0°.

**ACHTUNG:** In REMO sind nur bestimmte Gitterpunktsanzahlen in x- und y-Richtung erlaubt, die sich mit folgender Formel bestimmen lassen:

$n = 2^l \cdot 3^m \cdot 5^n + 1$  für alle  $l, m, n$  Element der natürlichen Zahlen inklusive Null und  $l$  ungleich 0

- Das Skript *first.sh* muss zunächst gestartet und sofort abgebrochen werden (`< ctrl > c`), um die Verzeichnisse für das Skript *rotcoord.sh* zu erzeugen.
- Nun wird erst *rotcoord.sh* und danach *first.sh* ausgeführt.
- Nach diesem Schritt kann *second.sh* gestartet werden. Die Ergebnisse der ersten beiden Skripte kann mit `cdo info $name_oro.srv` überprüft werden. Hier werden die MIN/MEAN/MAX Werte der Orographie in Spalten angezeigt.
- *third.sh*, *fourth.sh* (falls der Pol im Gebiet liegt, müssen *third\_b.sh* und *fourth\_b.sh* ausgeführt werden), *fifth\_frac.sh* und *sixth\_frac.sh* NACHEINANDER ausführen

**ACHTUNG:**

Ab REMO 5.1 gibt es eine fraktionelle Berechnung der Flüsse über Land, Wasser und Eis. Somit werden die Skripte mit der Endung *\_frac.sh* verwendet. Nur wer explizit keine fraktionelle Bodenbehandlung für Land, Wasser und Eis benutzt, nimmt die Skripte *fifth.sh* und *sixth.sh*.

Ist soweit alles reibungslos verlaufen, können mit *cdo info lib\_\${name}\_frac* die Bodenbibliotheksvariablen überprüft werden. In diesem Fall wurden für den LAI (Blattflächenindex), der Vegetationsanteil und die Albedo konstante Werte berechnet. Wenn man mit konstanter Vegetation rechnen möchte, also keinen Jahresgang dieser Parameter benötigt, ist man an dieser Stelle mit der Bodenbibliothek fertig. Üblicherweise wird der Jahresgang hinzugefügt:

- *vegcycle\_1.sh*, *vegcycle\_2.sh* und *albcycle.sh* IN DIESER REIHENFOLGE aufrufen.

Dadurch werden drei *.srv* Dateien erzeugt: *vlyear\_\${name}.srv* (Blattflächenindex), *vgryear\_\${name}.srv* (Vegetationsanteil) und *albyear\_\${name}.srv* (Bodenhintergrundalbedo). SRV-Dateien haben keine Information über die Rotation.

Die Bodenbibliothek ist nun fertig. ✓

# Kapitel 3

## Die Antriebsdaten

### 3.1 Vorbereitung

Die Antriebsdaten umfassen die Startwerte und die Randwerte für die atmosphärische Variablen Temperatur, Bodendruck, u- und v- Komponente des Windes, spezifische Feuchte und Flüssigwassergehalt. Weiterhin sind Werte für die Landmeerverteilung, die Orographie, die Feuchte und Temperatur in den Bodenschichten, Temperaturen der Wasser- und Landoberflächen, Wasserspeicher an der Bodenoberfläche (Skin Reservoir), Schneehöhe und -temperatur, Konzentration und Dicke des Meereises (falls vorhanden), aktueller und maximaler Bodenwassergehalt. Vorab müssen einige Ordner erstellt werden. Diese Ordner enthalten später u.a. die Antriebsdaten und die REMO-Outputdateien. Sie sollten auf der **blizzard** im eigenen workshare (alias *\$WRKSHR*) angelegt werden.

#### **Befehle für die Konsole:**

```
> cd $WRKSHR
> mkdir $name
> cd $name
> mkdir xe
> mkdir xa
> mkdir xpa
> mkdir xm
```

Nun können die Antriebsdaten erstellt werden. Diese werden für das regionale Gebiete durch einen Präprozessor aus den globalen Daten ausgeschnitten und auf die Auflösung von REMO interpoliert. Dazu stehen je nach Antrieb unterschiedliche Skripte zur Verfügung:

**HILFREICH:****Antrieb**

ERA40  
ERAinterim  
ERA15 und operationelle Daten  
ECHAM5  
REMO

**Skript**

etr  
eitr  
rtr  
gtr  
ete

In diesem Beispiel soll REMO mit dem globalen ECHAM5 Modell angetrieben werden, die aus dem Archiv in das Verzeichnis *\$WRKSHR/xa* geladen und entpackt (*tar – xvf file.tar*) werden. Den aktuellen Pfad ist dem ThyRemo Wiki zu entnehmen, wo weitere und aktuell gehaltene Informationen für Mitglieder der REMO Arbeitsgruppe zugänglich sind. Den Zugang erlaubt ggf. Claas Teichmann. Für ECHAM Daten wird der Präprozessor *gtr57* verwendet.

Man kopiere sich auf der **blizzard** aus Ralfs Homeverzeichnis den Ordner *utils*, in dem der Präprozessor enthalten ist, in ein Verzeichnis in seinem Homeverzeichnis:

**Befehle für die Konsole:**

```
cd $HOME/$Ordner  
cp -p -r /pf/k/k204009/remo/utils/ .
```

*/pf/k/k204009/remo/utils* enthält die Ordner:

- build
- druintzr
- gtr57 mit den Unterordnern CBS, DECKS und OFS, wobei der Letzter vor dem Kompilieren des Präprozessors leer sein muss
- jobs
- libs

Weiterhin benötigt man die Druckinterpolationsprogramme *mitzrpa* und *mitzrpe* aus dem Verzeichnis */pf/k/k204009/post*,

## 3.2 Das Kompilieren des Präprozessor

Zuerst müssen in der Datei *resge.h* unter *gtr57/CBS*, worin die Common-Blöcke definiert sind, die Gitterpunktzahlen des Modellgebietes (IEEM und JEEM) und die Anzahl der Modelllevel (KEEM) unter EM eingetragen werden. In Common-Blöcken werden die zu übergebenden Variablen zwischen verschiedenen Programmeinheiten gespeichert.

### ACHTUNG:

Ab hier benötigen wir die Gitterpunktanzahl aus *first.sh* ohne den Rand, also 2 Gitterboxen abziehen!

In *resge.h* sind die Modellgebietsgrößen und Level von ECHAM Daten eingetragen. Wird die Levelanzahl der Atmosphäre geändert, müssen unter */gtr57/DECKS/inidata.f* die Absätze der Ak- und Bk-Daten angepasst werden. Unter *pf/k/k204009/akbkdata* findet man die Daten (*.dat*), deren Inhalt in die entsprechenden Felder kopiert wird. Für 27 atmosphärische Levels muss *inidata.f* nicht angepasst werden, da dies der Standardwert in den Ak- und Bk-Daten ist.

Nun werden im Verzeichnis */build* in der Datei *gtr57* die Pfade angepasst:

```
rm ../libs/gtr57_$(#xboxen)x$(#yboxen)x$(#level)
```

z.B. *gtr57\_101x109x27* für die Gitterpunktsanzahl  $I = 101$  und  $J = 109$  und die Anzahl der Level  $K = 27$ .

Diese Änderung wird ebenfalls in Zeile 1 im *gtr57\_make* unter */build/* vorgenommen:

```
EXECUTABLE=../libs/gtr57_$(#xboxen)x$(#yboxen)x$(#level)
```

Nun wird das Skript *gtr57* gestartet. Dieses ruft *gtr57\_make* auf, welches die ausführbare Datei */libs/gtr57\_\$(#xboxen)x\$(#yboxen)x\$(#level)* erzeugt.

## 3.3 Der Druckinterpolator der Antriebsdaten

Das Modell rechnet auf Vertikalleveln, die in Bodennähe dem Gelände folgen und in der Höhe in horizontale Druckflächen übergehen. Die Vertikallevel sind abhängig vom Bodendruck. Demnach werden die prognostischen Variablen u.U. je nach Gitterbox auf unterschiedlichen Druckleveln berechnet. Für die Vergleichbarkeit mit den Ergebnissen aus REMO ist eine In-



terpolation der Variablen auf die selben Drucklevel nötig. Diese werden, wenn nicht anders definiert, für 250, 500 und 850 hPa berechnet. Der Druckinterpolator wird angepasst und anschließend gestartet:

- in *druintzr/CBS/param.h* IEMAX und JEMAX anpassen
- in *build/druintzr* bei rm I und J anpassen
- in *build/druintzr\_make* in der ersten Zeile I und J anpassen
- zum Kompilieren des Druckinterpolators *druintzr* unter */utils/build/* aufrufen

## 3.4 Monatsmittel der Inputdaten

Für Vergleichszwecke werden monatlich gemittelte Werte der Eingangsdaten berechnet. Dazu werden in *mitzrpa* und *mitzrpe* IX, IY und GEB an das Experiment angepasst und die Skripte anschließend ausgeführt. Beide erzeugen ausführbare Dateien im Verzeichnis *libs/*: *mitzrpa*<sub>Gebiet</sub> und *mitzrpe*<sub>Gebiet</sub>.

## 3.5 Jobskript für die ausführbaren Dateien

Unter */jobs* werden zwei Dateien erstellt: KMON (Monat, zweistellig) und KYEAR (Jahr, vierstellig), in denen das Startdatum des Experimentes gesetzt wird. Auf diese Dateien greifen die Skripte des Präprozessors zu. Falls dieser unerwartet abbricht, muss hier das Anfangsdatum aktualisiert werden. Nun wird das Jobskript des Präprozessors editiert, dass ebenfalls *gtr57* heißt, aber unter */jobs* liegt:

```

WD=$WRKSHR                # Pfad, wo die Daten liegen
PFB=Preprocessor/libs      # Pfad, wo ausführbare Datei liegt
PFO=$WD/remo/xe           # Pfad für Output (Ergebnis vom Präprozessor)
PFI=$WD/remo/xa           # Pfad für Input (ECHAM)
UG=                        # Experimentnummer der Inputdaten
EG=                        # Usernummer der Inputdaten
U=                         # Eigene REMO-Usernummer
E=                         # Eigene Experimentnummer (REMO)
M=                         # Preprocessor/jobs/KMON
Y=                         # Preprocessor/jobs/KYEAR
NSE=                       # Endstunde des Programms relativ zum Startdatum
                           # bei einem Testlauf z.B. 6:
                           # dann wird nur der erste und zweite Zeitschritt
                           # gerechnet. Sonst: expr · $Monday · 24
ANA=.TRUE.                # Schalter für Verwendung von
                           # Reanalysen oder Modellergebnissen

&EMGRID
DLAMEM=                   # Auflösung in x-Richtung
DPHIEM=                   # Auflösung in y-Richtung
POLLAM und POLPHI        # Lagekoordinaten des rotierten Nordpols
                           # Berechnung:
POLLAM=                   # Wenn Länge des Südpols LON_SP positiv:
                           # LON_SP-180
                           # Wenn Länge des Südpols LON_SP negativ:
                           # LON_SP+180
POLPHI=-LAT_SP            # Achtung: LAM = Länge, PHI = Breite

&DATEN
YBDPDN=lib_ $Gebiet_ $Aufloesung_ frac      # Name der Bodenbibliothek
YBDCAT=/work/$gruppe/$user/REMO/$Gebiet_ $Aufloesung/
time $PFB/gtr57_$(#xboxen)x$(#yboxen)x$(#level) < INPUT
time $PFB/druintzr_$(#xboxen)x$(#yboxen)

```

Nun kann der Präprozessor gestartet werden mit *gtr57* unter */jobs/*. ✓

## 3.6 Gleichgewichtszustand des Bodens

Bei Antrieb mit Reanalysen ist eine Einschwingperiode des Bodens auf Grund von dessen Trägheit notwendig, um einen stabilen Gleichgewichtszustand zu erreichen und künstliche Trends zu vermeiden. Abhängig vom Modellgebiet kann dieser Vorgang einige Modelljahre bis -jahrzehnte benötigen. Nachdem ein eingeschwungener Boden erzeugt wurde, müssen die Bodenvariablen in den Anfangsbedingungen aktualisiert werden. Existiert bereits ein Experiment für das Modellgebiet, etwa mit anderem Antrieb, kann die Bodeninitialisierung übernommen werden. Das Vorgehen kann in drei Schritten zusammengefasst werden:

- In der ersten Datei der Antriebsdaten müssen die Bodenfelder gelöscht werden, sodass nur die Atmosphäre übrig bleibt. Dazu wird das Skript *extmodatm* verwendet.
- Aus dem eingeschwungenen Boden müssen die neuen Bodenfelder herausgeschnitten werden. Dies kann mit dem Skript *extmodbod* durchgeführt werden.
- Beide Datensätze müssen zusammengefügt werden, z.B. mit *cat*.

Durch Alberto Elizalde wurde ein Skript bereitgestellt, dass die angegebenen Schritte in einem Tool zusammenfasst (siehe ThyRemo-Wiki).

# Kapitel 4

## Das REMO

Man kopiere sich in sein eigenes Verzeichnis */work/\$gruppe/\$user/* von Ralf die für seine Anwendung passende REMO-Version. Für den Zugang ist die Projektzugehörigkeit für REMO notwendig, die Ralf Podzun erteilen kann.

### Befehle für die Konsole:

```
cp -r -p /pf/k/k204009/remo$version /remo$version
```

### ACHTUNG:

Den REMO-Code bitte IMMER für andere Nutzer sperren:

```
chmod -r 700 /work/gruppe/user/remo$version
```

Im Ordner *remo\$version* befinden sich folgende Unterordner:

- CBS: Common-Blöcke für die Parameterübergabe
- CODE: hier liegt das Herz von REMO - die Fortran Routinen
- OFS: kompilierte Objektdaten der Fortran-Routinen, die mit den Bibliotheken verlinkt sind. Vor Beginn des Kompilieren empfiehlt sich alle Dateien in diesem Verzeichnis zu löschen.
- build: Skripte zum Kompilieren von REMO
- jobs: Jobskripte, mit denen REMO gestartet wird
- libs: in dieses Verzeichnis werden u.a. ausführbare Dateien geschrieben

## 4.1 REMO kompilieren

- Editiere in */remo\$version/CBS/param.h* IDISUM I und J
- Editiere in */remo\$version/CBS/COMGTS* die Zeit für den Treibhausgasantrieb: JPGTS1-Startjahr, JPGTS2-Endjahr  
(Treibhausgasdaten liegen unter */pool/data/remo/*)
- Editiere in */remo\$version/build/qm* I und J (ruft Makefile auf)
- Editiere in */remo\$version/build/Makefile* I und J
- *qm* laufen lassen. Es kompiliert REMO, was bis zu 30 Minuten Zeit in Anspruch nimmt. Es wird eine ausführbare Datei */remo\$version/libs/remo\$version\_par.exe\_I\_J* erzeugt, welche nachfolgend von einem REMO-Jobskript aufgerufen wird. ✓

## 4.2 REMO Jobskript

Das Jobskript steuert das REMO Experiment. Es werden die zu verwendenden Daten spezifiziert, Schalter gesetzt und herauszuschreibende Variablen von REMO festgelegt. Mögliche Jobskripte liegen im Verzeichnis */remo\$version/jobs*, z.B. *ens515*. Das Jobskript für die **blizzard** enthält in der Kopfzeile Informationen, die u.a. für die Abrechnung des Rechenzeitkontingents wichtig sind. Zunächst muss ein geeignetes Jobskript angepasst (oder eigenhändig geschrieben) werden:

```

initialdir                # Arbeitsverzeichnis, z.B. /work/scratch/m/$user/
                           # In diesen Pfad werden u.A. die logfiles geschrieben
                           # die später zur Programmüberwachung dienen
# TMPDIR im initialdir erstellen (dient als Zwischenablage):
-set +ex
mkdir TMPDIR
-set -ex
PFL                       # Pfad setzen: ~/remo$version/libs
# Pfade PFADA bis PFADT auf Verzeichnisse /xb/ bis /xt/ anpassen:
PFADA = $WD/remo/xa        # Input für REMO=Ergebnis vom Präprozessor,
                           # ggf. müssen die Daten verschoben werden
PFADE = $WD/remo/xe        # 2D-Output
PFADF = $WD/remo/xf        # Restart-Dateien
PFADH = $WD/remo/xm        # Monatsmittel
PFADT = $WD/remo/xn        # Tagesmittel
PFADM = $WD/remo/xt        # 3D-Felder
PFADN = $WD/remo/hilf2
UE/UA                     # eigene REMO-Nutzernummer angeben,
                           # i.R. UE=UA
EE/EA                     # Experimentnummer angeben, i.R. EE=EA
Y                         # vierstelliges Anfangsjahr der Antriebsdaten
M                         # Anfangsmonat der Antriebsdaten
D                         # Anfangstag der Antriebsdaten
H                         # Anfangsstunde der Antriebsdaten

&PARCTL                   # Kontrollschalter für Parallelrechner
NPROCXM/NPROCYM           # Aufteilung des Jobs auf die Prozessoren

&EMGRID                   # Angaben zum Gitter (EM vom Ursprungsmodell)
# Anpassung an Modellgebiet analog zu ~/utils/jobs/gtr57

```

**Fortsetzung Jobskript:**

```

&RUNCTRL
NHEAA=           # Wann erster Output, hier: nach erster Stunde...
NHDEA=           # Outputfrequenz, hier: ...jede Stunde
NHDMXN=1         # Wann Summen Null gesetzt werden, hier:
                  # ... jede Stunde
                  # sinnvoller Weise: NHEAA=NHDEA=NHDMXN
DT=60.0          # Zeitschritt in Sekunden
                  # abhängig von Auflösung:
                  # Erfahrungswert bei 0.088°:60.0s
                  # 0.5°:240.0s
                  # (3600/DT=natürliche Zahl)
LQWR=.TRUE.      # Wolkenwasser, z.B. bei ANALYSEN = .FALSE.
LSCEN=.TRUE.     # Schalter beobachtete Treibhausgaskonzentrationen
LMOMIT=.TRUE.    # Schalter Monatsmittelberechnung online
LGMON=.TRUE.     # Schalter Berechnung ganzer Monate für Restartfiles
&PHYCTL          # Physik-Kontrollschalter
LPHY=.TRUE.      # Schalter Physik, wenn .FALSE. nur Dynamik
HDRAD=1,         # hier stündliches Aufrufen der Strahlung
IEXC=5,          # Arno-Bodenschema, Standardwert
                  # Ansprechpartner: Stefan Hagemann
LVEG=.TRUE.      # mit Vegetationszyklen rechnen:
                  # .TRUE. und Pfad bei &DATEN YBDCAT angeben!
LSICED=.TRUE.    # Seeisdicke im Input
                  # (z.B. für ECHAM5 See-Eis in Antriebsdaten
                  # .FALSE.:default Werte von 1m und 2m verwendet)
L5LAY=.FALSE.    # 5 Bodenschichten von Stefan Hagemann
                  # Stand März 2010: bleibt vorerst abgeschaltet

&NMICTL          # Schwerewellen Kontrollschalter
LANMI=.TRUE.     # .FALSE.: reine Dynamik
                  # z.B. für ersten beiden Zeitschritte ohne Physik,
                  # damit Schwerewellen gedämpft werden
DTNMI=45         # nicht-lineare Beziehung
                  # abhängig von der Zeitschrittwahl
                  # Erfahrungswerte: 0.088° und 60s: 5.0)

```

**Fortsetzung Jobskript:**

```

&PRCTL                                # online Status Print
LDIA=.FALSE.                          # Immer .FALSE.!

&DATEN
# Doppelte Pfadnennung aus programmiertechnischen Gründen notwendig:
YBDCAT                                # Pfad zu den Vegetations- und Albedodaten
YBDNAM                                # Name der Vegetations- und Albedodaten
YGDCAT                                # Pfad zu Treibhausgaskonzentrationen
GGDNAM                                # Name der Treibhausgaskonzentrationen
YMVARN                                # stündlich herauszuschreibende Variablen
YNVARN                                # herauszuschreibende Tagesmittel
YTVARN                                # herauszuschreibende Atmosphärenfelder
# Hinter poe Pfad und Name des erstellten remo$version__par.exe_I_J
# (ausführbare Datei wurde mit qm erzeugt)

```

Nun kann der Job abgeschickt werden:

- auf der **blizzard** werden Jobskripte mit dem Befehl *llsubmit \$NameDesJobskripts* (hier *ens515*) in die Queue geschickt
- Befehle zur Jobsteuerung:
  - Einsicht der Jobliste in der Queue:  
*llq | grep \$Userid* oder *llq -u \$UserID*
  - Job abbrechen, falls etwas schief läuft:  
*llcancel \$JobID*
- Zustand des Jobs: R = running, I = idle (warten), H = hold (warten), C = canceled
- In dem definierten *initialdir* (im Header des Jobskripts) werden Log-files geschrieben, in die der Status geschrieben wird: *err.123456* und *out.123456*. Bricht das Modell ab, können hier die Ursachen gelesen werden.

### 4.3 Restart

Wenn das Jobskript auf Grund von Computerproblemen abbricht oder Zeitschritte nach Beendigung des Experiments erneut gerechnet oder ergänzt werden sollen, wird ein sogenannter Restart ausgeführt. Von einem Restart wird auch im Falle einer Jobkette gesprochen,



wo sich das Skript am Ende mit den Angaben zum anschließenden Zeitschritt automatisch erneut aufruft. Für die Umsetzung müssen zunächst die gesamten Antriebsdaten erstellt und im Archiv gespeichert werden. Es ist aufgrund der begrenzten Speicherkapazität zu empfehlen, die Antriebsdaten jährlich herunterzuladen, z.B. durch ein Skript im Hintergrund, welches jeweils nach Beenden des Modellmonats November die Antriebsdaten lädt und diejenigen des letzten Zeitschritts löscht. Das Laden der neuen und das Löschen der alten Antriebsdaten wird in das Jobskript mit Restartfunktion eingebettet. Somit ist ein Experiment mit REMO über längere Zeiträume (10-100 Jahre) möglich. Das Jobskript wird an den folgenden Stellen angepasst:

```
# Aktualisierung der Anfangsdatums:
Y          # vierstelliges Anfangsjahr der Antriebsdaten
M          # Anfangsmonat der Antriebsdaten
D          # Anfangstag der Antriebsdaten
H          # Anfangsstunde der Antriebsdaten
DSA       # Stunden relativ zum ersten Zeitpunkt
```

Im Falle einer Jobkette werden die neuen Zeitangaben aus einer Textdatei eingelesen. Diese wird nach jedem Zeitschritt von REMO automatisch herausgeschrieben.

#### HILFREICH:

Ein Skript kann mit dem Befehl *nohup* im Hintergrund ausgeführt werden. Einsicht in den aktuellen Bearbeitungsstand gibt *nohup.out*. Alle laufenden Prozesse können mit *ps -fu \$USERID* eingesehen werden. Falls ein Skript abgebrochen werden soll, listet dieser Befehl die zugehörige ID. Einen Prozess kann man mit *kill \$ID* abbrechen.

## 4.4 Doppelnesting

Ein Doppelnesting ist, wie eingangs besprochen, notwendig, um zu große Skalensprünge (ab einem Faktor 8) in der horizontalen Auflösung zu vermeiden. In diesem Fall wird der vorgestellte Ablauf mit entsprechenden Modifikationen erneut durchlaufen, wobei nun das Jobskript *ete57* für den Präprozessor verwendet wird. Als Antriebsdaten werden die Ausgabedaten von REMO verwendet.

#### ACHTUNG:

Bevor der Präprozessor für das Doppelnesting gestartet wird, muss unter *\$WRKSHR* die Datei mit den Antriebsdaten der ersten Stunde von *xe/* auf *hilf3/* kopiert werden, da REMO in das Letztere erst nach den ersten 6 Stunden die Atmosphärenfelder schreibt. Wird dies nicht beachtet, fehlt der Input für den ersten Zeitschritt.



# Kapitel 5

## Postprozessing

### 5.1 Druckinterpolator der REMO-Outputdaten

Eine Druckinterpolation der Ausgabedaten von REMO kann mit dem Skript *expe* unter */utils/jobs/* durchgeführt werden, wobei *epxa* und *epxe* identische Einträge haben bis auf:

ANA=.FALSE.	#keine Antriebsdaten mehr nötig
PFI=/xt	#Inputverzeichnis der 6h Daten
PFO=/xpt	#Outputverzeichnis
NSA=6	#erste Datenausgabe von REMO nach 6h

Nach Anpassung der Parameter kann das Skript gestartet werden. ✓

### 5.2 Bearbeitung von großen Datensätzen

Die Ausgabedaten von REMO werden standardmäßig im *ieg* Format gespeichert. Die Climate Data Operators (CDOs) sind effizient, um die Daten einzusehen als auch zu bearbeiten, z.B. Mittelwertbildung. Sie stellen eine umfangreiche Liste an Funktionen bereit. Mit dem Befehl *cdo -f nc ifile.ieg ofile.nc* kann das Eingabefeld in NetCDF Format umgewandelt werden. Dieses Format kann z.B. mit *ncview* in einfachen Darstellungen visualisiert werden. Für qualitativ hochwertigere Darstellungen, kann *GrADS* oder *NCL*, dass auch als Erweiterung innerhalb der Sprache Python verfügbar ist, verwendet werden. Beide Visualisierungspakete verfügen über eine Skriptsprache, die zum wiederholen plotten empfehlenswert ist.