

2 FORSCHEN MIT ALGORITHMEN

Erdsysteme als komplexe, computerrealisierte Experimentalsysteme sind Instrumente einer Wissenschaft, die sich im Bereich der Algorithmen verortet. Dies aber bedeutet, dass nur das, was sich mit Algorithmen und quantitativen Datenstrukturen fassen lässt, der Forschung zugänglich ist. In der Meteorologie gewinnt dieser neue Forschungsstil mit den ersten Computereperimenten an Bedeutung. Frederik Nebeker bezeichnet in seiner Studie *Calculating the Weather* die 1950er Jahre als paradigmatische Umbruchphase:

„A new style of meteorology evolved in the 1950s. It was much the style practised by Richardson some 35 years earlier, a style that results from making a forecasting algorithm one's ultimate objective. This style is characterized by interest in only certain types of data (quantitative, and with accuracy and geographic distribution determined by needs of the algorithms), interest in only certain types of theories (quantitative, and ‚meshing‘ with other quantitative theories, and dealing with only those phenomena the algorithm indicates are important), and interest in numerical analysis and computing aids. In the 1950s it was the availability of electronic computers that turned many meteorologists in the direction taken by Richardson much earlier“ (Nebeker 1995: 152).

Wissenschaftliche Programmierung

1956 begann mit der Einführung der mathematiknahen Programmiersprache FORTRAN Formula Translator die Epoche des wissenschaftlichen Programmierens. Der Grund, warum FORTRAN schnell zur Universalsprache der wissenschaftlichen Programmierung avancierte, lag in

deren Philosophie, „a concise, fairly natural mathematical language“ (Herrick, Backus 1954: 112) für die Programmierung zu verwenden. 1954 wurden die meisten Programme noch direkt als kryptische Maschinenanweisungen an die Computer delegiert. Mathematik als natürliche Sprache der Programmierung zu verwenden, basiert dabei zum einen auf der Logik der Computer als Rechenmaschinen, zum anderen auf der zunehmenden Mathematisierung der Natur- und Ingenieurwissenschaften.¹ Beides erklärt den Erfolg dieser ersten Programmiersprache. „Fortran spread rapidly as it fulfilled a real need“ (Metcalf, Reid, Cohen 2004: 2).² Bruce Rosenblatt, einer der ersten Anwender, bestätigte, dass es einfach und naheliegend war, FORTRAN zu lernen, dass es schnell an Studenten weitergegeben wurde und, vor allem, dass die Programmiersprache verfügbar war (vgl. Rosenblatt 1984).

„The general availability of FORTRAN was probably one of the most important aspects of its perpetuation. There has been a synergistic effect in the case of the technical and scientific users. They built up great libraries of programs and subroutines that could be quickly put on new machines as they were introduced, providing they had a FORTRAN compiler. So it was fairly natural to build FORTRAN compilers for all of the new machines as they were introduced to the scientific and technical industry. [...] Probably the most important thing about FORTRAN has been its adaptability. Subroutines, on-line computer programming, operating systems, and structured programming have all been added to FORTRAN systems naturally and easily. FORTRAN has been able to keep up with the latest ideas in programming art. Several languages have been built on FORTRAN base: FORMAC, PROSPRO, in the process-control business, PLAGO in the construction business, SIMSCRIPT, and Vector FORTRAN, just to name a few. All this has helped FORTRAN and its work in the scientific community. [...] In many scientific organizations, the whole problem can be readily expressed in FORTRAN; therefore, there has been no incentive to switch“ (Rosenblatt 1984: 39, 40).

- 1 „Fortran’s superiority had always been in the area of numerical, scientific, engineering, and technical applications“ (Metcalf, Reid, Cohen 2004: 3). Interessanter Weise lässt sich die Verbesserung von FORTRAN mit Einschränkung als ein frühes ‚open source‘ Projekt verstehen. Der Austausch zwischen den Programmiersprachenentwicklern und Anwendern erfolgte über Treffen mit den Benutzergruppen. „Soon there were hundreds of customers making hundreds of suggestions for improvements. [...] The suggestions just poured in, and we put them in as fast as we could“ (Heising 1984: 32).
- 2 „Inevitably dialects of the language developed, which led to problems in exchanging programs between computers, and so in 1966 the then American Standards Association (later the American National Standards Institute, ANSI) brought out the first ever standard of a programming language, now known as Fortran 66“ (Metcalf, Reid, Cohen 2004: 2).

Die Akzeptanz von FORTRAN als Programmiersprache für wissenschaftliche Modelle, die Adaption neuer Programmieranforderungen – nach FORTRAN O bis IV folgten FORTRAN f66, f77, f90, f95, 2003 und 2008 – sowie der permanente Zuwachs an programmierten Funktionen und Subroutinen der FORTRAN Libraries machen FORTRAN neben C, respektive C++, zur wichtigsten Programmiersprache der Wissenschaft. Bis heute sind die meisten Wetter- und Klimamodelle in FORTRAN geschrieben.



Abbildung 11: Klimamodellierer, der an der Parallelisierung des Modells arbeitet (Gramelsberger 2006)

Die Folge der wissenschaftlichen Programmierung ist, dass seit den 1950er Jahren große Wissensbestände auf Basis codierter Theoriemodelle in den Computer transferiert werden. Dies hat sowohl epistemische wie praktische Folgen für die Forschung. Die Praxis des wissenschaftlichen Programmierens verändert die alltägliche Arbeitsweise der Forscher, denn die vorrangigen Arbeitsinstrumente sind nicht mehr die Mess- oder Experimentierapparate, sondern die Algorithmen und Daten. Konnte man vor der Einführung der Computer an den Tätigkeiten der Forscher meist ablesen, welcher Disziplin sie angehörten – wie dies beispielsweise Bruno Latour in seiner Studie über Bodenforscher mit Fotografien eindrucksvoll belegte (vgl. Latour 2000) – so lässt sich dies für die Computational Sciences nicht mehr erkennen. Ob ein Biologe, ein Klimaforscher oder ein Physiker vor dem Computer sitzen und ihrer Arbeit nachgeht, lässt sich von außen betrachtet nicht entscheiden. Alle tun dasselbe, sie schreiben Codes. Dies zeigt, dass der Computer eine enorme Vereinheitlichung der Forschungspraxis mit sich gebracht hat und dass sich diese Vereinheitlichung auch epistemisch, in die Computer

und Modelle hinein, fortsetzt. Deutlich wird dies daran, dass codierte Modelle Schnittstellen bieten, um algorithmierte Wissensbestände anderer Disziplinen einzubinden. Erdsysteme sind hierfür gute Beispiele. Sie fungieren dabei als Attraktoren für Interdisziplinarität. Ein Klimafor-scher formulierte es in einem Interview wie folgt:

„Wir haben jetzt die Klimamodelle um Vegetationsmodelle erweitert, wir interessieren uns für Mikrobiologie in den Böden, weil da Spurenstoffe aufgenommen und abgegeben werden. Wir interessieren uns für marine Biosphäre, d.h. wir arbeiten mit Zoologen zusammen. Wir arbeiten teilweise mit Wirtschaftswissenschaftlern zusammen, weil es auch um Emissionsszenarien, Energieszenarien usw. geht. [...] Meistens beginnt die Modellentwicklung erst als Folge der Zusammenarbeit. Einfach weil ein Mikrobiologe vielleicht Modellvorstellungen entwickeln wird, um seine eigenen Messungen zu interpretieren. Aber das sind so klein-skalige Modelle, die können wir für Klimamodellierung überhaupt nicht verwenden. Wir arbeiten auf Skalen von Hunderten von Kilometern. Da versuchen wir Prozesse zu beschreiben, die vielleicht teilweise im Nanobereich liegen, wenn ich an Aerosolpartikel denke. Es ist ein neues Denken für viele Leute, dass solche Modelle in Zusammenarbeit gemacht werden“ (Interview 1, 2003).

Diese interdisziplinäre Zusammenarbeit setzt voraus, dass mathematische Modellierung und Programmierung, ob in FORTRAN oder C++, zum Handwerkszeug einer Disziplin und ihrer Forscher gehören. Ohne diese gemeinsame Lingua franca der Modellierung und Programmierung wären die Fortschritte in den Computational Sciences nicht denkbar. Daher verwundert es nicht, dass dieselben mathematischen Gleichungen, Modelle und gegebenenfalls Programmteile sich in unterschiedlichen Disziplinen finden lassen. So werden beispielsweise die Navier-Stokes-Gleichungen der Strömungsdynamik sowohl in Atmosphären- wie Ozeanmodellen, als auch in medizinischen wie technischen Strömungssimulationen verwendet. Dies legt die Schlussfolgerung nahe, dass die wissenschaftliche Programmierung mit Theorie in modulare Art und Weise verfährt, insofern „sich Stücke aus [mathematisierten] Theorien verschiedener Disziplinen herauslösen lassen, um sie als Bausteine einer neuen Theorie zu verwenden, wobei sich in den neuen Kombinationen deren Bedeutungen wandeln“ können (Lüdtke 1993: 112). Doch Wissenschaft macht dies seit jeher, wenn es dem Forschungszweck dienlich ist, beispielsweise um komplexe Theorien oder Experimentalsysteme zu realisieren. Das Neue daran ist, dass nun mit Theorie ganz konkret in Form eines Baukastens – zusammengesetzt aus codierten Theoriebausteinen wie in Erdsystemen – experimentiert werden kann.

In-silico Experimentalsysteme

Solche Theoriebaukästen lassen sich auch als in-silico Experimentalsysteme beschreiben. Das Adjektiv in-silico bezieht sich auf die siliziumbasierten CPU Central Processing Units der Computerchips, also auf eine rein computerbasierte Realisation, während sich der Begriff des Experimentalsystems auf die Nutzung des Computers als Experimentierapparatur bezieht.³ Dabei handelt es sich um eine sehr spezielle Experimentierapparatur, die rein im semiotischen Medium der Zahlen operiert. Hans-Jörg Rheinberger entwickelte in seiner Studie *Experimentalsysteme und epistemische Dinge. Die Geschichte der Proteinsynthese im Reagenzglas* den Begriff des Experimentalsystems am Beispiel biologischer Laborforschung (vgl. Rheinberger 2002).

„Der Begriff Experimentalsystem wird jedenfalls von Wissenschaftlern in der Biomedizin, Biochemie, Biologie und Molekularbiologie häufig benutzt, um die Rahmenbedingungen der eigenen Forschungsarbeit zu charakterisieren. Fragt man heute einen Biowissenschaftler, der im Labor tätig ist, nach seiner Arbeit, so wird er von seinem ‚System‘ erzählen und dem, was alles in diesem System passiert. Es ist also in erster Linie ein Begriff der Praktiker. [...] Betrachtet man Experimentalsysteme etwas näher, so wird man feststellen, daß in ihnen zwei verschiedene, jedoch voneinander nicht trennbare Strukturen ineinandergreifen. Die erstere kann man als Gegenstand der Forschung im engeren Sinne, Wissensobjekte oder auch epistemische Dinge bezeichnen. Als epistemische präsentieren sich diese Dinge in einer für sie charakteristischen, irreduziblen Verschwommenheit und Vagheit“ (Rheinberger 2002: 23, 24). Um diese epistemischen Dinge zu erforschen, benötigen Forscher „stabile Umgebungen, die man als Experimentalbedingungen oder als technische Dinge bezeichnen kann; die epistemischen Dinge werden von ihnen eingefasst und dadurch in übergreifende Felder von epistemischen Praktiken und materiellen Wissenskulturen eingefügt. Zu den technischen Dingen gehören Instrumente, Aufzeichnungsapparaturen und, in den biologischen Wissenschaften besonders wichtig, standardisierte Modellorganismen mitsamt den in ihnen sozusagen verknöcherten Wissensbeständen“ (Rheinberger 2002: 25).

Die Beobachtung vor Ort zeigt, dass epistemische Objekte zu technischen werden können und umgekehrt. Tatsächlich ist Laborforschung ein komplexes Wechselspiel zwischen beiden Komponenten, das sich oft in jahrzehntelanger Forschungsarbeit entfaltet, bis es zu einem be-

3 In-silico grenzt sich dabei von üblichen Charakterisierungen wie in-situ (Experimente vor Ort), in-vivo (Experimente an lebenden Organismen), in-vitro (Experimente an organischen Prozesse außerhalb lebender Organismen) ab.

friedigenden Ende kommt. Diese Perspektive auf die konkrete Forschung im Labor lässt den Werkstattcharakter der Experimentalsysteme sichtbar werden und widerspricht der wissenschaftstheoretischen Auffassung, dass Experimente wohldefinierte Verfahren seien, um Theorien zu prüfen. Geprüft werde, so Karl Popper 1935 in *Logik der Forschung*, „ob sich das Neue, das die Theorie behauptet, auch praktisch bewährt, etwa in den wissenschaftlichen Experimenten oder in der technisch-praktischen Anwendung. Auch hier ist das Prüfungsverfahren ein deduktives: Aus dem System werden (unter Verwendung bereits anerkannter Sätze) empirisch möglichst leicht nachprüfbar bzw. anwendbare singuläre Folgerungen (‚Prognosen‘) deduziert [...] und in] den Experimenten usw., entschieden“ (Popper 1935/1989: 8).⁴

Solange ein eindeutiges System mathematisierter Sätze einer Theorie vorliegt, wie dies für die neuzeitliche Mechanik der Fall ist, dienen die Experimente der Überprüfung wie von Popper gefordert. Doch für die meisten Wissenschaftsbereiche ist die Situation weniger eindeutig. Das Experimentieren in der Realität der Labore folgt einer anderen Forschungslogik. Hier geht es weniger um die Überprüfung von theoriebasierten Hypothesen und Prognosen, als um die Stabilisierung wissenschaftlichen Wissens in einem unterdeterminierten Theorieraum flüchtiger Phänomene und vager Vermutungen. Insofern sich diese im Laufe der Laborforschung nicht stabilisieren lassen, sind sie wissenschaftlich uninteressant. Es bedarf der Stabilisierung und des evidenten Nachweises dieser Phänomene, um wissenschaftlich triftige Fakten zu generieren. Diese Fakten bedeuten nicht das Ende der Forschung, sondern meist deren Weiterführung unter neuen Bedingungen. Forschung in diesem Sinne verstanden ist ein rekursiver Prozess, dessen rekursive Differenzialität sich aus der Einfügung und Weiterverarbeitung der gewonnenen

4 Ein solches Wissenschaftsverständnis geht von mathematisierten Theorien aus, deren Prognosen sich im Experiment in Form von Ja-/Nein-Antworten entscheiden lassen: „Aus der vorläufigen Antizipation, dem Einfall, der Hypothese, dem theoretischen System, werden auf logisch-deduktivem Weg Folgerungen abgeleitet; [...] Dabei lassen sich insbesondere vier Richtungen unterscheiden, nach denen die Prüfung [der Folgerungen] durchgeführt wird: der logische Vergleich der Folgerungen untereinander, durch den das System auf seine innere Widerspruchslosigkeit hin zu untersuchen ist; eine Untersuchung der logischen Form der Theorie, mit dem Ziel, festzustellen, ob es den Charakter einer empirisch-wissenschaftlichen Theorie hat, also z.B. nicht tautologisch ist; der Vergleich mit anderen Theorien, um unter anderem festzustellen, ob die zu prüfende Theorie, falls sie sich in den verschiedenen Prüfungen bewähren sollte, als wissenschaftlicher Fortschritt zu bewerten wäre; schließlich die Prüfung durch ‚empirische Anwendung‘ der abgeleiteten Folgerungen“ (Popper 1935/1989: 8, 9).

Erkenntnisse generiert. Experimentalsysteme sind dabei „jene materiellen Formationen oder Dispositionen der epistemischen Praxis, innerhalb derer Wissenschaftler die epistemische Produktion erzeugen, die sie als die ‚Resultate‘ ihrer Arbeit apostrophieren. Ein - positives – Resultat ist ein Befund, der im Prinzip als Komponente wieder in das System eingebaut werden und es damit erweitern und verändern kann“ (Rheinberger 2002: 146).

Genau in dieser Weise charakterisiert sich auch die Verwendung von in-silico Experimentalsystemen in der täglichen Forschungspraxis. Ebenso wie die Experimentalsysteme der biologischen Laborforschung sind in-silico Experimentalsysteme evolvierende Systeme, die sich im Laufe der Forschung verändern, indem sie die erreichten Resultate inkorporieren und dadurch neue Forschung inspirieren. Im Falle der in-silico Experimentalsysteme lässt sich diese Evolution anhand des permanent wachsenden Umfangs des Codes und der aufeinander folgenden Modellversionen und -generationen explizit nachvollziehen. Eine Archäologie des Codes dieser Systeme würde deren Entwicklung rekonstruieren, da in-silico Experimentalsysteme keine materialen Formationen sind, sondern rein zeichenbasierte Objekte. Betrachtet man ein konkretes, computerrealisiertes Experimentalsystem – im Rahmen dieser Studie wäre dies das Atmosphärenmodell ECHAM5 des Max-Planck-Instituts für Meteorologie⁵ – so wird bereits in der Modellbeschreibung, die jedem Wetter- oder Klimamodell beigelegt ist, auf den evolutionären Prozess der Generierung hingewiesen:

„The fifth-generation of atmospheric general circulation model (ECHAM5) developed at the Max Planck Institute for Meteorology (MPIM) is the most recent version in a series of ECHAM models evolving originally from the spectral weather prediction model of the European Centre for Medium Range Weather Forecasts (ECMWF; Simmons et al. (1989))“ (Roeckner et al. 2003: 5). „The climate model ECHAM5 has been developed from the ECMWF operational forecast model cycle 36 (1989) (therefore the first part of its name: EC) and a comprehensive parametrisation package developed at Hamburg (therefore the abbreviation HAM). The part describing the dynamics of ECHAM is based on the ECMWF documentation, which has been modified to describe the newly implemented features and the changes necessary for climate experiments. Since the release of the previous version, ECHAM4, the whole source code has been extensively redesigned in the major infrastructure and transferred to Fortran 95. ECHAM is now fully portable and runs on all

5 ECHAM5 ist, gekoppelt mit dem OM Ozeanmodell, eines der dreiundzwanzig globalen Referenzmodelle des vierten IPCC Reports (vgl. IPCC AR4 WG1 2007).

major high performance platforms. The restart mechanism is implemented on top of netCDF and because of that absolutely independent on the underlying architecture“ (Roeckner et al. 2003: 7).

Dieser evolutionäre Prozess der Generierung von Code macht deutlich, dass diese Systeme das Resultat einer kollaborativen Schreibaarbeit zahlreicher Autoren sind. Ein in-silico Experimentalsystem wie ECHAM5 ist ein Korpus aus gut zweihundert miteinander in Interaktion stehenden Dateien und hat sich über Jahre, oder gar Jahrzehnte, in Generationen von Modellversionen in einem kollektiven Schreibprozess entwickelt. Allerdings unterscheidet sich das Schreiben von Code vom herkömmlichen Produzieren und Publizieren von Theorie in Artikeln oder Fachbüchern ganz erheblich. In Forschungsberichten und Artikeln über Computermodelle finden sich Metaphern für diese neue Art des Schreibens wie ‚numerische Realisierung aktuellen Wissens‘, ‚komplexe mathematische Abbilder der Realität‘ oder ‚dynamisch-basierte Ansätze zur Systembeschreibung‘. Oft ist in Gesprächen mit Modellieren von Reparatur oder Tricks die Rede.⁶ Das Experimentalsystem wird als Experimentiermaschine objektiviert, an deren Schrauben man drehen kann, die sich notfalls mit behelfsmäßigen Tricks reparieren lässt. Schreiben in diesem Zusammenhang bedeutet, dass das was als Code notiert wird, aufgrund seiner Operativität funktionieren muss. Lässt sich Theorie durch Beobachtung, Messung oder Argumente bestätigen, so muss codierte Theorie zu aller erst funktionieren, also auf dem Computer laufen, bevor damit wissenschaftlich gearbeitet werden kann. Dazu müssen alle Fehler behoben sein, welche zu einem Computerabsturz führen können.⁷

-
- 6 Beispielsweise: „Eine einfache Reparatur [der Klimadrift], etwa durch Modifikation der Parametrisierung des Impulsflusses an der Ozeanoberfläche ist nicht möglich, so dass als einzige Möglichkeit langwieriges Ausprobieren bleibt. Diese Möglichkeit ist jedoch in der Regel einfach aus Gründen der Rechenzeit nicht gegeben. [...] Man versucht stattdessen mit Hilfe eines unphysikalischen Tricks, der Flusskorrektur (flux adjustment), diese Klimadrift zu vermeiden. [...] Der Trick besteht nun darin, dass man die zeitlich gemittelten Differenzen dieser Felder [Felder für Wärme-flüsse, die von den Teilmodellen separat auf Basis vorgegebener Meeresoberflächentemperaturen berechnet wurden] in den Flüssen zwischen Ozean und Atmosphäre im Zuge der gekoppelten Rechnung immer wieder hinzufügt“ (von Storch, Güss, Heimann 1999: 127, 128).
 - 7 Fehler der codierten Theorie sind dabei noch nicht berücksichtigt. Anders gewendet: Auch wenn das Modell läuft und Resultate produziert, bedeutet dies nicht, dass es wissenschaftlich korrekte Resultate liefert. Da die materielle Widerständigkeit als experimentelles Korrektiv fehlt, müssen neue Evaluationsstrategien für die in-silico Forschung entwickelt werden.

Dieser Entstehungs- und Veränderungsprozess eines computerrealisierten Experimentalsystems wird nicht nur in den Modellbeschreibungen dokumentiert, sondern auch in den Kommentaren in den einzelnen Dateien. Im Unterschied zu Forschungsergebnissen, die klassisch in Form von wissenschaftlichen Artikeln und Vorträgen publiziert werden, stellen die Modellbeschreibungen und -programme selbst eine neue Textform dar, die sich eventuell mit Laborbüchern vergleichen lässt. Die Möglichkeit, Kommentare in Programmdateien zu schreiben, erlaubt es, Informationen über das Experimentalsystem in den Dateien zu speichern. Üblicherweise werden so die Autoren eines Modells oder einer Datei benannt, die verwendeten Konzepte unter Hinweis auf die Fachliteratur referenziert sowie Beschreibungen einzelner Funktionen oder Restriktionen des Modells gegeben. In Abbildung 12 ist dies anhand der Wolkendatei `cloud.f90` des Atmosphärenmodells ECHAM5 dargestellt. Unabhängig vom Code erschließt sich über die Kommentare die Geschichte des Experimentalsystems. Wie umfangreich der Kontext eines Modells rekonstruiert werden kann, hängt davon ab, wie sorgfältig der Code kommentiert wurde.

```

*Cloud* computes large-scale water phase changes, precipitation,

! Subject.
! -----
!
35: ! This routine computes the tendencies of the four prognostic
! variables (temperature t, specific humidity q, cloud liquid
! water xl, cloud ice xi) due to phase changes (condensation/
! deposition, evaporation/sublimation of rain/snow falling
! into the unsaturated part of the grid box, melting of snow,
40: ! melting/freezing of cloud ice/cloud water, sedimentation of
! cloud ice, and precipitation formation in warm, cold and
! mixed phase clouds.
! The precipitation at the surface (rain and snow) is used in
! later for computing the land surface hydrology in *surf*.
45: ! The cloud parameters (cloud cover, cloud liquid water and
! cloud ice are used for the calculation of radiation at the
! next timestep.

[...]

110: ! References.
! -----
!
! Lohmann and Roeckner, 1996: Clim. Dyn. 557-572
! Levkov et al., 1992: Beitr. Phys. Atm. 35-58.(ice phase)
115: ! Beheng, 1994: Atmos. Res. 193-206.(warm phase)
! Lenderink et al., 1998; KNMI-REPORT NO. 98-13 (condensation)
! Tompkins 2002, J. Atmos. Sci. (cloud cover)
!
! Authors.
120: ! -----
! M.Esch MPI-Hamburg 1999
! G.Lenderink KNMI, de Bilt 1998
! U.Lohmann MPI-Hamburg 1995
!
125: ! Modifications.
! -----
! E.Roeckner MPI-Hamburg 2000
! A.Tompkins MPI-Hamburg 2000
! U.Schlese MPI-Hamburg 2003

```

*Abbildung 12: Kommentare zu Beginn der cloud.f90 Datei des
Atmosphärenmodells ECHAM 5 (MPI-Met 15. Juli 2005)*

Um den kollektiven Schreibprozess am Modell innerhalb einer Institution zu koordinieren und um jede Änderung des Codes sichtbar und nachvollziehbar zu machen, nutzen die Forscher CVS Concurrent Versioning Systeme. CVS Systeme sind gängige Instrumente der Softwareentwicklung, um Programmierern die aktuellste Version einer Datei für Überarbeitungen zur Verfügung zu stellen, um eine Historie der Änderungen anzulegen und um die Autoren der Änderungen zu registrieren und zu dokumentieren, wie in Abbildung 13 dargestellt. Dies ist für ein in-silico Experimentalsystem von entscheidender Bedeutung, denn jede Änderung des Codes verändert die codierte Theorie und damit das Experimentalsystem. Ein Klimamodellierer formulierte es folgendermaßen: „Wenn ich mit einem Kollegen bespreche, was ich an einem Modell ändern möchte, dann bespreche ich das im Detail und gehe unter Umständen in den Code. Wir schauen uns ein paar Zeilen vom Code an und sa-

gen, das und das möchten wir ändern oder das und das habe ich geändert“ (Interview 6, 2005). Welche Änderungen im Laufe der Modellverbesserung dabei durchgeführt werden, ist Teil der sozialen Forschungspraxis in Form von Entscheidungen und Aushandlungen. Da die Dynamik für alle Zirkulationsmodelle dieselbe ist, variieren die Modelle in der Ausgestaltung des Klimasystems mit klimarelevanten Prozessen und in der Art, wie diese Prozesse parametrisiert und programmiert sind. Dies macht den Unterschied der verschiedenen Klimamodelle aus.

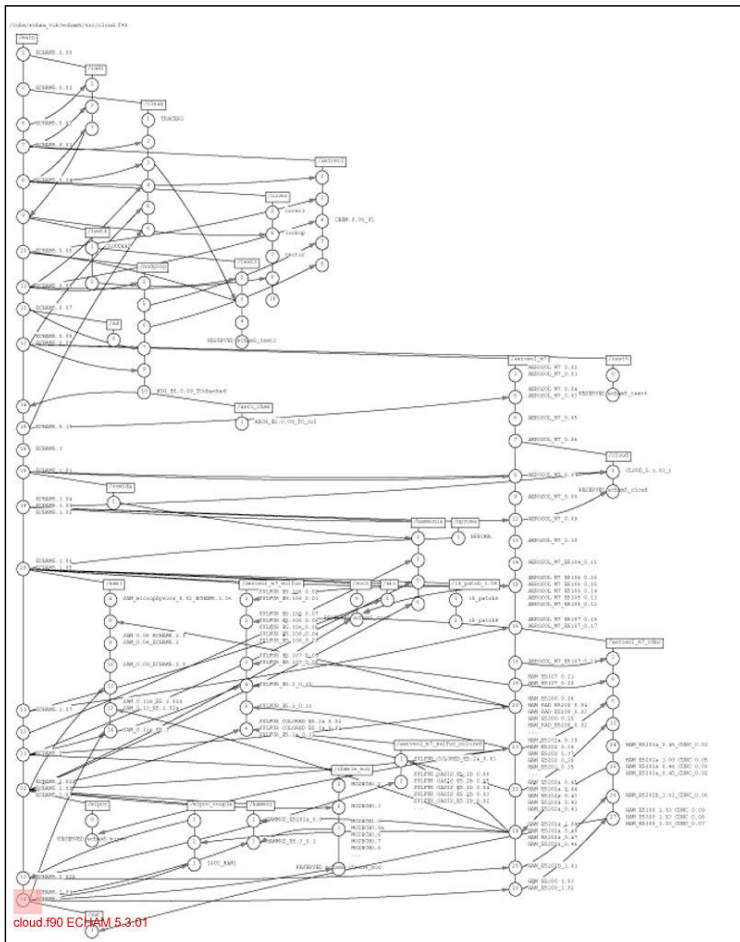


Abbildung 13: Änderungen der cloud.f90 Datei des Atmosphärenmodells ECHAM5 von Version 5.0.00 bis 5.3.01 (MPI-Met 2006)

Die permanenten Verbesserungen und Veränderungen am Code lassen aktuelle Klimamodelle zu komplexen Strukturen anwachsen, deren Einbettung in die Rechnerumgebung sich nicht einfach gestaltet. Denn auch die Rechnerumgebung ist durch Supercomputercluster, verteiltes Rechnen und Grid-Computing zunehmend komplexer und serviceintensiver geworden. Dies führt dazu, dass die Modelle durch adäquates Software-Engineering für die jeweilige Computerarchitektur optimiert werden müssen. Bei der zunehmenden Größe der Modelle erfordert dies ein neues Programmierparadigma, das die Kapselung der Programmteile vornimmt und dadurch die Komplexität eines Modells reduziert. Indem gekapselte Teile als geschlossener Baustein oder Blackbox definiert werden, lassen sich diese als Module über Schnittstellen für die Dateneingabe und -ausgabe ansteuern. „Was man dabei kapseln muss, ist nicht die Routine, die wissenschaftliche Rechnungen ausführt (z.B. Vertikaldiffusion oder allgemein Advektion),“ wie ein Klimaforscher im Interview verdeutlichte, „sondern die Verwaltung von Feldern, der Zeitsteuerung des Modells, der Parametrisierung. So entstehen Schnittstellen, an welchen man Programmteile einhängen kann, für die die benötigten Variablen geliefert und am Ende ausgeschrieben werden. Dafür gibt es Mechanismen, die einfach zu übersehen sind und die man benutzen kann. Und dann können die Wissenschaftler ihren Code einklinken“ (Interview 7, 2005). Der Code wird dadurch in Module zerlegt, die dann zu verschiedenen Experimentalsystemen kombiniert werden können. Auf diese Weise lassen sich auch ältere Modellteile einfügen, Standardisierungen einführen und Modellkomponenten ohne viel Aufwand in noch größere Systeme einbinden.

Die Software-Einbettung auf diese Weise zu verbessern, ist eine aktuelle Herausforderung vor der die Klimamodellierung steht, insbesondere wenn sie Erdsysteme realisieren will.⁸ „Das Hauptaugenmerk wird dabei darauf gelegt, möglichst allgemeine Lösungen zu entwickeln, die nicht nur in anderen Erdsystem-Modellen, sondern auch für andere Fra-

8 Dabei gilt es, die Autonomie der Forscher bei der Erstellung codierter Theorie zu erhalten. Die Delegation der Programmierung eines Modells an fachfremde Programmierer ist ein heikles Unterfangen, weil Programmierung hier wissenschaftliche Modellierung codierter Theorie bedeutet. Andererseits erfordert die zunehmende Komplexität der Rechnerarchitekturen die verstärkte Unterstützung durch Programmierer. Informatiker arbeiten jedoch meist mit anderen Programmiersprachen und wesentlich elaborierteren Softwarekomponenten als Wissenschaftler, für die die Programmierung lediglich ein Mittel zum Zweck ist. Der Einsatz professioneller Programmierer kann einerseits eine unbeabsichtigte Einflussnahme auf die Theorie bedeuten, aber andererseits auch neue Potenziale eröffnen. Da diese Entwicklungen mit Standardisierungen einhergehen, vereinfacht es beispielsweise den Modellvergleich und damit die Evaluierung der Modelle.

gestellungen und auch in Disziplinen außerhalb der Klimaforschung, wie beispielsweise der Strukturmechanik und der Strömungsmechanik, Anwendung finden können. Dies geschieht insbesondere durch modulare Kapselung von Einzelaufgaben“ (MPI Met 2008: 3). Auf diese Weise werden zunehmend komplexere und modularere Theoriebaukästen in den Computational Sciences realisiert. Die forschungspraktische Idee dabei ist, nicht nur immer umfangreichere Experimentalsysteme kollaborativ zu realisieren, sondern ein, auf die jeweilige Forschungsfrage hin zugeschnittenes Experimentalsystem nur für das jeweilige Experiment zusammenstellen zu können. Daher ist die Idee, die codierte Theorie in eine avancierte Software-Einbettung einzuklinken eine zukunftsweisende Weiterentwicklung der aktuellen Forschungspraxis mit in-silico Experimentalsystemen. Im vollen Umfange soll diese Idee im Rahmen der US-amerikanischen Initiative des ESMF Earth System Modeling Frameworks Wirklichkeit werden.

„The Earth System Modeling Framework (ESMF) collaboration is building high-performance, flexible software infrastructure to increase ease of use, performance portability, interoperability, and reuse in climate, numerical weather prediction, data assimilation, and other Earth science applications. The ESMF defines an architecture for composing complex, coupled systems and includes data structures and utilities for developing individual models. We are aiming to create a framework usable by individual researchers as well as major operational and research centers, and seek to engage the community in its development. The basic idea behind ESMF is that complicated applications should be broken up into smaller pieces, or components. A component is a unit of software composition that has a coherent function, and a standard calling interface and behavior. Components can be assembled to create multiple applications, and different implementations of a component may be available. In ESMF, a component may be a physical domain, or a function such as a coupler or I/O system. ESMF also includes toolkits for building components and applications, such as regridding software, calendar management, logging and error handling, and parallel communications“ (ESFM 2009: About ESMF)

Allerdings besteht bei diesem plug-n-play Konzept die Gefahr, die Autonomie der Forscher zu unterlaufen. Plug-n-play meint das nutzerorientierte Zusammenstecken benötigter Modellkomponenten analog einem Baukastensystem, ohne die Teile im Detail kennen und anpassen zu müssen. Das setzt eine Software-Einbettung voraus, die die Modellkomponenten als funktionale Blackboxes nicht nur kapselt, sondern abkapselt. Selbst unerfahrene Anwender können dann Experimente auf Knopfdruck ausführen. Der Modellcode wird in dieser Konzeption als abgeschlossene Funktionsmaschinerie zur Produktion von Outputdaten

definiert. Forschung und Programmierung werden dadurch strukturell wie organisationell auseinanderdividiert. Programmieren die Forscher in FORTRAN die Modelle bisher selbst, so erfordern solche großen Modellsysteme einen operationellen Betrieb.

Atmosphärenmodell

Die konkrete Ausgestaltung eines in-silico Experimentalsystems lässt sich am Beispiel des Atmosphärenmodells ECHAM5 beschreiben. Dabei repräsentiert ECHAM5 ein ‚state-of-the-art‘ Experimentalsystem der aktuellen Klimaforschung. Es ist eines von dreiundzwanzig Referenzmodellen, die für die Szenarienberechnungen des vierten IPCC Reports verwendet wurden (vgl. IPCC AR4 WG1 2007). ECHAM5 besteht aus gut zweihundert Dateien, welche in ihrer Gesamtheit die Prozesse der Atmosphäre darstellen. Dabei teilen sich die Dateien in zwei Gruppen auf: Dateien, welche die globale Dynamik beschreiben und Dateien, welche die subskalige Parametrisierungen modellieren. Die Dynamik der Atmosphäre ergibt sich, wie bereits ausführlich dargestellt, aus den Wechselwirkungen der sieben Zustandsgleichungen der meteorologisch Variablen (Temperatur, Druck, Dichte, Feuchte sowie die Windgeschwindigkeit in drei Richtungen), die im Modell in diskretisierter Form global beschrieben sind. Hinzukommen die Wirkungen der Energieflüsse – solare Einstrahlung und terrestrische Abstrahlung in Abhängigkeit von dem jeweiligen Reflexionsvermögen der Oberflächen – auf die Dynamik, die Wärme flüsse der fluiden Medien Atmosphäre und Ozeane sowie die Energietransporte der Wasserkreisläufe. Die Dynamik des Klimas ergibt sich aus den globalen Zirkulationsprozessen, die durch die Ungleichgewichte der Energiezufuhr und -abfuhr in Gang gehalten werden. Dabei spielt die Advektion, die sich aus der verstärkten Einstrahlung in den Tropen und der verstärkten Abstrahlung in den polaren Gebieten ergibt, eine wichtige Rolle: Erwärmte Luft- oder Wassermassen fließen in kältere Regionen (sensibler Wärmefluss). Aber auch die Konvektion, das Aufsteigen erwärmter Luftschichten in der Atmosphäre und die daraus entstehenden Druckunterschiede, durch das seitliche Abfließen warmer Luftmassen, ist ein maßgeblicher Faktor. Denn die räumliche Differenz von Druck und Temperatur erzeugt als eine Art Wärmekraftmaschine Bewegung in Form mechanischer Energie, also Strömung. Ein weiterer, im Modell zu beschreibender Faktor ist die Erdrotation, deren Effekt von den Tropen zu den Polen, abhängig vom relativen Abstand zur Rotationsachse der Erde, zunimmt und ein komplexes Zirkulationsmuster zur Folge hat.

Zu diesen globalen Prozessen gesellen sich lokale und regionale Effekte, die von den topographischen Gegebenheiten beeinflusst werden. Darüber hinaus unterliegt das klimatische Geschehen natürlichen Variabilitäten wie dem Jahres- und Tagesgang, Wetterereignissen und interannualen Anomalien wie den ENSO El Niño-Southern Oscillation Phänomen, der Nordatlantik Oszillation oder Sonnenflecken. All dies muss in seiner Wechselwirkung codiert und dem Modell hinzugefügt werden. Dasselbe gilt für die zahlreichen klimarelevanten Prozesse, die sich unterhalb der Auflösungsgrenze des Modells vollziehen. Sie werden als subskalige Parametrisierungen in das globale Modell integriert. Dazu gehören beispielsweise der Einfluss der Aerosole oder der Wolkenbildung. Das maßgebliche Ziel eines jeden Klimamodells ist es, die Energiebilanz des Klimasystems adäquat zu errechnen. Gleichen sich die Energieflüsse der Ein- und Ausstrahlung nicht aus, ergibt die Bilanz einen Nettofluss in Form von Erwärmung bzw. Abkühlung.⁹

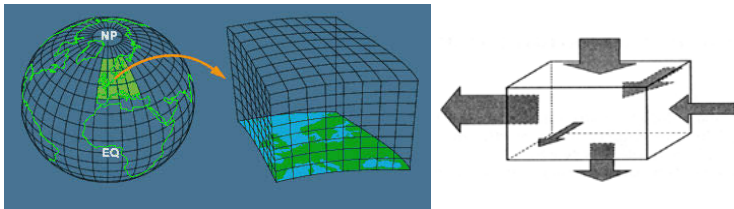


Abbildung 14: Ortsfeste Volumen (l.) sowie Zu- und Abflüsse in einem Volumen (r.) (DKRZ 2000, von Storch, Giss, Heimann 1999: 100)

- 9 Ohne Atmosphäre würde die Strahlungsbilanz der Erde eine gemittelte Temperatur von -15° ergeben, während das globale Temperaturmittel dank des Treibhauseffekts der Atmosphäre bei $+15^{\circ}$ liegt. Bereits 1896 stellte der Meteorologe Svante Arrhenius mit einem einfachen (nulldimensionalen) EBM Energiebilanzmodell – Einstrahlung, kurzwellige Rückstreuung und langwellige Ausstrahlung – die Theorie der Erderwärmung durch Treibhausgas auf (vgl. Arrhenius 1986, 1903). Die Temperaturänderung und damit der Nettofluss hängen dabei vom Wärmespeichervermögen der Erdoberflächen ab. Allerdings erfassen nulldimensionale EBMs nur ein global gemitteltes Gleichgewicht der Strahlungsenergien und vernachlässigen die Energietransporte von den Tropen zu den polaren Breiten. Eindimensionale EBMs modellieren diesen Energietransport als Band aus Modellvolumina vom Nord- zum Südpol. Dabei werden die an sich dreidimensionalen Energietransporte im eindimensionalen Modell als Volumenmischung vereinfacht: Ein warmes Volumenelement kühlt ab, während das benachbarte Element sich erwärmt. Zweidimensionale EBMs erlauben es, die Energieflüsse horizontal und vertikal darzustellen. Aufgrund der steigenden Rechenkapazitäten basieren heutige Klimamodelle auf dreidimensionalen, globalen Zirkulationsmodellen, so genannten GCMs Global Circulation Models.

Die beschriebenen Prozesse, die Dynamik wie die Parametrisierungen, finden Eingang in ein globales Atmosphärenmodell wie ECHAM5 und konstituieren ein komplexes in-silico Experimentalsystem. Doch um ein solches Experimentalsystem auf dem Computer zum Laufen zu bringen, bedarf es geeigneter Modellierungsstrategien, um die verschiedenen Prozesse im endlichen Berechnungsraum des Computers in effektiver Zeit zu realisieren. Das basale Konzept numerischer Klimamodellierung besteht nun darin, das Klimaverhalten als Wirkung auf Fluide in ortsfesten Volumenelementen zu betrachten, wie in Abbildung 14 dargestellt. Ein Zirkulationsmodell beschreibt also, wie sich der Zu- und Abfluss der Masselemente in einem Volumenelement verhält (Kontinuitätsgleichung), wie sich die Energieumwandlung darstellt (Energiebilanzen für Strahlung, Wärme, mechanischer Energie), welche Kräfte auf die Fluide wirken und wie sich dadurch der Betrag und die Richtung ihrer Geschwindigkeit verändern (Impulserhaltung: Bewegungsgleichungen bzw. Navier-Stokes-Gleichungen), wie die Zu- und Abnahme der Beimengungen vonstatten geht (Transportgleichungen für Salz, Wasserdampf, Spurengase, Wasser etc.) und wie sich die Dichte der Fluide in Abhängigkeit von Druck, Temperatur und Wassergehalt wandelt (Zustandsgleichungen). All diese Gleichungen beeinflussen sich wechselseitig. Sie sind die Grundlage der Klimasimulation, die für jedes Volumenelement den Zustand der sieben Variablen berechnet (vgl. von Storch, Giss, Heimann 1999). Hinzukommen die subskaligen Prozesse, die sich innerhalb der Volumenelemente abspielen und als Parametrisierungen in die Berechnungen einfließen.

Sobald die Programmierung einer Modellvariante, beispielsweise für ECHAM5, abgeschlossen ist, kann damit experimentiert werden, sei es zu Testzwecken, für die Forschung oder für die Erstellung von Prognosen. Doch das Experimentieren mit in-silico Experimentalsystemen unterscheidet sich in zwei Punkten grundlegend von den realweltlichen Experimenten im Labor: Zum einen gehen die Forscher nicht direkt mit Materialitäten und Objekt-Welten um, sondern lediglich mit Repräsentationen davon, in Form von Messwerten und konzeptuellen Zeichenäquivalenten. Dieser Unterschied ist entscheidend, denn während im Labor die Widerständigkeit des Materiellen über das experimentelle Gelingen oder Scheitern entscheidet, fehlt diese Widerständigkeit im rein Semiotischen der in-silico Experimentalsysteme. Daher braucht es neue Kriterien der Falsifizierung für Computerexperimente oder, positiv gewendet, neuer Evaluationsstrategien. Zum anderen bewegen sich die Experimente mit in-silico Experimentalsystemen in überdeterminierten Theorieräumen, denn diese Systeme sind als Theoriebaukästen konzipiert. Die Frage, welcher Forschungslogik sie dabei unterliegen – der Popperschen Variante der Überprüfung von theoriebasierten Hypothesen und Prognosen oder der Logik

der Laborforschung – ist einfach zu beantworten. Es handelt sich um eine neue Art des Experimentierens mit Theorien, die dem Experimentieren in den Laboren der Biologen ähnelt. Denn die Neukombination modularer Theoriebausteine erzeugt neue Bedingungen, die es anhand des simulierten Systemverhaltens zu erforschen gilt. Der Werkstattcharakter der in-silico Experimentalsysteme zeigt sich in den Tausenden von Codezeilen. Jede Änderung der Programmierung verändert gleichzeitig das Experimentalsystem und das theoriebasierte Setting wie auch das Experiment. Forschung mit Computerexperimenten gleicht daher eher einem permanenten Rekonfigurieren, Differenzieren und (Re-)Arrangieren als einem exakt definierten, deduktiven Frage-und-Antwort-Vorgang.

Rechnen als Experiment

Experimentieren mit in-silico Experimentalsystemen bedeutet im Konkreten, dass Simulationsläufe durchgeführt werden. Ein Klimamodellierer beschreibt diese Art des Experimentierens wie folgt:

„Wir sprechen von Experimenten. Wir betrachten das Modellsystem als Labor. Wir machen ein Experiment, wo wir etwas ändern und sehen, was herauskommt. Das sind Computerexperimente, die wir ja in der Natur nicht durchführen können oder sollten. [...] Letztendlich ist die Herangehensweise ganz ähnlich wie bei jemanden, der Messungen durchführt. Wir ändern etwas in der Modellanordnung, im System, und schauen dann, wie reagiert das System darauf und versuchen es zu verstehen. Im Idealfall läuft es so: Die Messleute finden irgendetwas, was sie sich nicht erklären können. Und dann versucht man es mit dem Modell zu erklären, versucht es zu simulieren. Das sind für uns die interessanteren Aufgaben. Man hat eine Frage, die man beantworten will. Häufig ist es umgekehrt, etwas ironisch formuliert, dass wir jede Menge Antworten produzieren und uns dann auf die Suche nach den Fragen machen“ (Interview 1, 2003).

Konkret bedeutet dies, dass die einzelnen Dateien eines Atmosphärenmodells entsprechend einem festgelegten Schema, wie in Abbildung 15 dargestellt, nacheinander ablaufen, um so – ausgehend von Anfangswerten basierend auf Messdaten für t_0 – den Zustand der meteorologischen Variablen für den Zeitpunkt t_1 zu berechnen. Indem die Resultate von t_1 als Anfangszustand für die Berechnung von t_2 verwendet werden, usw., rechnet sich das Modell Zeitschritt für Zeitschritt in die Zukunft, oder auch zurück in die Vergangenheit. In ihrer Gesamtheit bilden die Dateien das in-silico Experimentalsystem. Die spezifischen Messdaten zur Initialisierung der Simulation, die gewählte räumliche und zeitliche Auflösung sowie die spezifischen Werte für Parameter, Konstanten und Randbedin-

gungen charakterisieren als Modellkonfiguration das Experiment. Diese Informationen sind in so genannten batch files gespeichert und initialisieren das Experiment. Batch files beinhalten Anweisungen über den Start des Programms, über die Modellkonfigurationen des konkreten Experiments, über die Anzahl der benötigten Prozessoren, das Abspeichern der Zwischenschritte (Ausschreiben der berechneten Daten für einen Zeitschritt) sowie die Nachbereitung der Resultate. Für die wissenschaftliche Spezifikation des Experiments sind vor allem die Modellkonfigurationen der Anfangs- und Randbedingungen des Experiments von Bedeutung. Die in netCDF Network Common Data Form, einem der Standardformate wissenschaftlicher Datendarstellung, gespeicherten Anfangsdaten (Messwerte) werden als Anfangsbedingungen in das Programm eingelesen. Nach dieser Initialisierung erfolgt die lineare Abarbeitung der im Quellcode beschriebenen Anweisungen von Zeitschritt zu Zeitschritt. Dabei werden die Dateien in einer festgelegten Reihenfolge aufgerufen und durchlaufen, bis die Ergebnisse für den ersten Zeitschritt vorliegen und im gängigen netCDF Format in eine Datenbank ausgeschrieben werden. Diese Daten dienen dann der Initialisierung der Berechnung des nächsten Zeitschritts. Auf diese Weise arbeitet sich das Modell voran. Die Momentaufnahmen der einzelnen Zeitschritte lassen sich später als Grafiken, Bilder oder als animierter Film der Klimaentwicklung visualisieren.

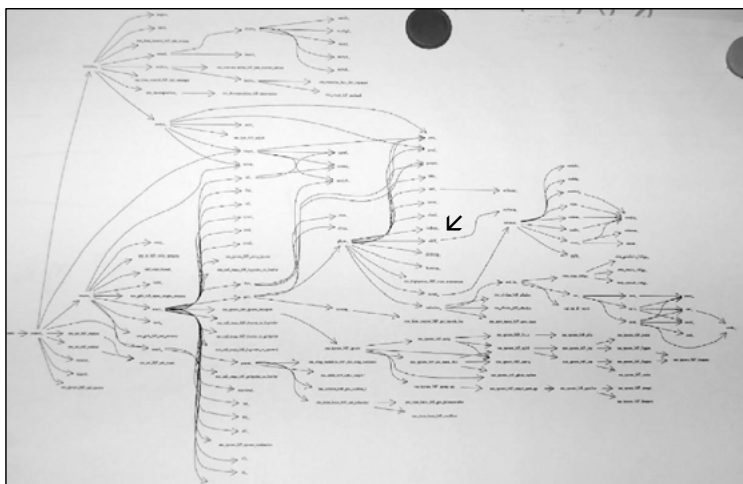


Abbildung 15: Dateidurchlauf von links nach rechts während der Berechnung des Atmosphärenmodells ECHAM5. Der Pfeil markiert die cloud.f90-Datei des Modells (Gramelsberger 2006)

Bevor Simulationsläufe jedoch zu Prognosezwecken genutzt werden können, bedarf es einer ausreichenden Testphase zur Evaluation des Modells anhand von Messdaten. Für das Atmosphärenmodell ECHAM5 dauerte diese Testphase im Zuge der Modellrechnungen für den vierten IPCC Bericht gut zwei Jahre. In dieser Zeit wurde, jeweils als Antwort auf die Testergebnisse, die Modellierung immer wieder verbessert. Aber auch während eines Experiments mit einer getesteten Modellversion werden immer wieder Testläufe in höherer Auflösung durchgeführt, um die Stabilität der Resultate während des Experiments zu kontrollieren. Ein experimentelles Setting besteht daher aus Szenarien- und Testläufen.

Ein konkretes Experiment, das 2003 der Evaluation von ECHAM5 diente, waren die Simulationsläufe zur Wetterlagenklassifikation von Jana Sillmann (vgl. Sillmann 2003). Das Experiment basierte auf der Zuordnung atmosphärischer Zirkulationsmuster zu bestimmten Wetterlagen eines konzeptuellen Wetterlagenmodells (vgl. Enke, Spekat 1997) anhand vergleichbarer Simulationsläufe mit dem gekoppelten Atmosphären-Ozeanmodell ECHAM5/MPI-OM1 und dem ungekoppelten ECHAM5 Atmosphärenmodell. Die experimentell zu bestimmenden Leitgrößen waren dabei die gemittelte Tagessumme der Niederschläge (Klassifikation von „sehr trocken“ bis „sehr starker Niederschlag“ gemäß der Niederschlagsmenge) sowie die gemittelte maximale Temperatur (Klassifikation „extrem kalt“ bis „extrem warm“ anhand jahreszeitlich bedingter Temperaturzuweisungen). Mit dem Experiment sollte die Prognosegüte des Modells für die Bestimmung der Wetterlagen eines konkreten Tages im Jahresverlauf getestet werden. Im Ergebnis wiesen die im Experiment berechneten Wetterklassen beider Modellvarianten gegenüber den Messdaten (NCEP/NCAR-Datensatz) eine neunzig-prozentige Prognoserate für die Temperatur und eine achtzig-prozentige für den Niederschlag auf.¹⁰ Darüber hinaus zeigte das Experiment in Hinblick auf die Evaluierung des gekoppelten Modells, dass die „relativen Häufigkeiten der Wetterlagen aus den ECHAM5/MPI-OM1 Modelldaten im Vergleich zu den NCEP/NCAR-Reanalysen deutlich zu kälteren Wetterlagen hin verschoben [waren]. Dies kann hauptsächlich durch den systematischen Fehler im ECHAM5/MPI-OM1 Modell begründet sein, welcher einen so genannten ‚cold bias‘ über Eurasien ver-

10 Der NCEP/NCAR-Datensatz ist ein Referenzdatensatz des US-amerikanischen NCEP National Centers for Environmental Prediction und dem NCAR National Center for Atmospheric Research, der eine modellbasierte Reanalyse von Messdaten für einen bestimmten Zeitraum zusammenstellt (vgl. NCEP/NCAR 2009: Reanalysis).

ursacht. Das bedeutet, dass das gekoppelte Modell generell über Europa zu kalt ist. Der Temperaturfehler kann bis zu -10°C über Eurasien betragen. Dieser systematische Fehler wirkt sich besonders im Winter auf die Häufigkeitsverteilung der Wetterlagen des ECHAM5/MPI-OM1 Modells aus“ (Sillmann 2003: 28).

Auf Basis dieser Ergebnisse lässt sich dann nach Ursachen für diesen ‚cold bias‘ suchen, um das Modell zu verbessern. Aber das Experiment gab noch weitere Einblicke in das Modellverhalten. Interessanter Weise zeigte sich, dass die Szenarienläufe des gekoppelten Modells die Wetterlagen schlechter repräsentieren als die Szenarienläufe des ungekoppelten Atmosphärenmodells. Darüberhinaus wurde deutlich, dass die Wahl des Zeitpunkts für das Ausschreiben der Ergebnisdaten der Szenarienläufe (0 UTC, 12 UTC) auf die Berechnung der gemittelten maximalen Temperatur einen wesentlich größeren Einfluss hatten als auf die ebenfalls untersuchten Niederschlagswerte. Der Versuch, diese Ergebnisse zu verstehen, vermittelt einen Eindruck von der Komplexität des Experimentierens mit in-silico Experimentalsystemen.

„Die Wetterlagen im Temperaturregime werden vor allem durch die Schichtdicke 1000/850 hPa und 1000/700 hPa bestimmt. Diese auch als relative Topographie bezeichnete Schichtdicke wird wiederum vom Tagesgang der Temperaturen beeinflusst. Da die veränderte Schichtdicke direkt Einfluss auf die Berechnung der Wetterlagen nimmt, können Unterschiede im Tagesgang der Temperaturen bzw. der Veränderung der relativen Topographie in den ECHAM-Modellversionen zu unterschiedlichen Ergebnissen in der eindeutigen Zuordnung der Wetterlagen im Wetterlagenklassifizierungsmodell führen. Besonders im Frühling und Sommer treten größere Differenzen zwischen Tag- und Nachttemperaturen auf, so dass in diesen Jahreszeiten der Temperaturunterschied zwischen 0 und 12 UTC bemerkbare Unterschiede in der eindeutigen Wetterlagenzuordnung hervorruft. Das Modell ECHAM5/MPI-OM1 weist eine höhere Sensitivität in diesen Vorgängen auf. Veränderungen in dieser Modellversion hinsichtlich der Parametrisierung einiger Prozesse, wie Wolkenbildung, saisonale Vegetation und Albedo wirken sich eventuell stark auf die Tages- und Nachttemperaturen aus und beeinflussen so die geopotentielle Höhe bzw. relative Topographie“ (Sillmann 2003: 60).

Ähnlich wie in den Laboren der Biologen sind die experimentellen Ergebnisse Ausgangspunkt für weitere Modifikationen des Experimentalsystems. Denn nur durch Rekonfigurationen von Teilen des Codes sowie neuen Experimenten kann herausgefunden werden, warum beispielsweise die Wahl des Zeitpunkts einen so großen Einfluss auf die Resultate hat. Dies ist nur ein Beispiel der Fehler- und Problemsuche, das während einer Testphase oder auch während der Forschung mit in-silico Experi-

mentalsystemen hervortreten kann. Während jedoch im realweltlichen Labor die Forscher bei „der Konfiguration und Rekonfiguration epistemischer Dinge [...] auf mehr oder weniger hartnäckige Widerstände und Widerspenstigkeiten des Materials, mit dem sie umgehen“ stoßen (Rheinberger 2002: 245), werden die Forscher im Computerlabor mit einer anderen Art von Widerständigkeit konfrontiert. Da aber selbst im biologischen Labor, die „Wissenschaftsobjekte, nicht als materielle Dinge an sich, sondern als epistemisch konfigurierte Objekte“ angesehen werden müssen, hat man es auch hier mit „ständig im Fluß befindlichen transversalen Verkettungen von Darstellungen“ zu tun (Rheinberger 2002: 246). Analog dazu lassen sich die Wissenschaftsobjekte der in-silico Experimentalsysteme ebenso als transversale Verkettungen von Darstellungen charakterisieren.¹¹ Allerdings handelt es sich um codierte Formen von Darstellungen, die anderen Bedingungen unterliegen, wie zu zeigen sein wird.

Story Telling with Code

Ein Klima- oder Erdsystem erhebt den Anspruch, die relevanten klimatischen Prozesse unseres Planeten adäquat zu modellieren. Mit Submodellen und hunderten von Prozessen wird dabei eine bestimmte Geschichte des Klimasystems erzählt, die in der Tradition einer bestimmten Sichtweise der Naturwissenschaft auf das realweltliche Klima steht. Diese Sichtweise hat sich im Laufe der letzten vierhundert Jahre entwickelt und verfeinert. Sie zeigt sich in der Quantifizierung und Mathematisierung realweltlicher Phänomene, in deren Modellierung und, mit der Erfindung der elektronischen Rechenmaschinen, in der Codierung und computerbasierten Ausführung eben dieser Modelle in Form numerischer Prozesse. Die Frage, ob die codierte und computerprozessierte Sichtweise denselben Bedingungen unterliegt wie die traditionell naturwissenschaftliche – also die mathematisierte und quantifizierte, aber noch nicht algorithmierte und numerisch simulierte – zielt auf den epistemischen Kern des Wandels der Wissenschaft im Zeitalter des Computers. Es erscheint einleuchtend, dass ein weiterer Transformationsschritt sowie der mediale Wechsel in den Computer neue Bedingungen hinzufügen. Welche dies sind, soll am konkreten Beispiel der Parametrisierung stratiformer Wolken anhand der cloud.f90 Datei von ECHAM5 untersucht werden. Dabei werden die mathematische Model-

¹¹ Einzig die materiale Widerständigkeit als experimentelles Korrektiv fehlt in den in-silico Umgebungen. Was dieses experimentelle Korrektiv ersetzt, wird zu klären sein.

lierung und die Codierung des Schmelzvorgangs Q_{mli} von Wolkeneis r_i miteinander verglichen.¹²

Grundsätzlich werden Wolken in globalen Zirkulationsmodellen parametrisiert, da die Auflösung dieser Modelle mit 60 bis 500 Kilometer in der Horizontalen und 16 bis 30 Schichten in der Vertikalen zu grob ist. Wolken fallen buchstäblich durchs Raster der Simulation, denn erst ab circa einem Kilometer lassen sich die für die Wolkenphysik relevanten Bewegungen auflösen. Wolken sind jedoch wichtige Klimafaktoren, denn sie beeinflussen maßgeblich die Strahlungsbilanz sowie den Wasserkreislauf der Atmosphäre. Wolken werden daher in Form subskaliger Parametrisierungen in das Modell integriert. Für ECHAM5 geschieht dies anhand zwei verschiedener Wolkenschemata für konvektive und stratiforme Wolken, wobei letztere in der cloud.f90 Datei beschrieben werden. Das stratiforme Wolkenschema basiert auf den prognostischen Gleichungen der drei Wasserphasen, der Wolkenmikrophysik sowie einem statistischen Wolkenbedeckungsschema. In der Modellbeschreibung, welche die mathematische Modellierung darlegt, werden die ‚governing equations‘ wie folgt beschrieben (vgl. Roeckner et al. 2003).

„The scheme for the representation of stratiform clouds consists of prognostic equations for the vapor, liquid, and ice phase, respectively, a cloud microphysical scheme (Lohmann and Roeckner (1996); with some revisions), and a statistical cloud cover scheme including prognostic equations for the distribution moments (Tompkins, 2002). [...] The governing equations for the grid-cell mean mass mixing ratios of water vapor, r_v , cloud liquid water, r_l , and cloud ice, r_i , are written in symbolic form as follows (units are $\text{kg kg}^{-1} \text{s}^{-1}$)

[Änderung des Wasserdampfs r_v in Wolken] $\partial r_v / \partial t = Q_{Tv} + Q_{evr} + Q_{evl} + Q_{sbs} + Q_{sbis} + Q_{sbi} - Q_{cnd} - Q_{dep} - Q_{tbl} - Q_{tbi}$ (10.1)

[Änderung der Flüssigkeit r_l in Wolken] $\partial r_l / \partial t = Q_{Tl} + Q_{mli} + Q_{mlis} + Q_{cnd} + Q_{tbl} - Q_{evl} - Q_{frh} - Q_{fis} - Q_{frc} - Q_{aut} - Q_{racI} - Q_{sacI}$ (10.2)

[Änderung des Eis r_i in Wolken] $\partial r_i / \partial t = Q_{Ti} + Q_{sed} + Q_{dep} + Q_{tbi} - Q_{mli} - Q_{sbi} + Q_{frh} + Q_{fis} + Q_{frc} - Q_{agg} - Q_{saci}$ (10.3)“ (Roeckner et al. 2003: 61).

Das mathematische Modell gibt gemäß der Modellbeschreibung an, wie die Vorgänge in den stratiformen Wolken konzipiert sind. Daraus leitet sich nun die Modellierung für das programmierte Modell ab. Beispielsweise wird die Wolkenbedeckung als Verteilung von Wasser in den ortsfesten Volumen (grid-cell) modelliert, wobei sich die Ver-

¹² Q_{mli} beschreibt das „instantaneous melting of r_i if the temperature exceeds the freezing point“ (Roeckner et al. 2003: 62).

teilung statistisch-dynamisch berechnen lässt. „The fractional cloud cover can be expressed as [...] probability density function (PDF) of $r_i = r_v + r_l + r_i$. [...] The remaining task is to choose an appropriate PDF and to determine its moments. The choice of the PDF was guided by simulations with a cloud resolving model (CRM) which was run on a horizontal domain of about 90 km x 90 km and 21 km in the vertical (Tompkins, 2002)“ (Roeckner et al. 2003: 63). Die Varianz resultiert aus den prognostischen Gleichungen, welche die Produktion, den turbulenten Transport und die Dissipation der Varianz berücksichtigen. Das mikrophysikalische Schema beschreibt die Änderungen der Wasserphasen sowie verschiedene Niederschlagsprozesse. In Abschnitt 10.3.3 der ECHAM5 Modellbeschreibung wird das ‚freezing of cloud liquid water and melting of cloud ice‘ beschrieben:

„At temperatures $T < -35^\circ\text{C}$, the total amount of cloud liquid water freezes homogeneously and instantaneously, during one time step Δt , to cloud ice Levkov et al. (1992) so that

$$Q_{frh} = r_l / \Delta t \quad (10.40).$$

For stochastic and heterogeneous freezing in the temperature range $-35^\circ\text{C} \leq T < 0^\circ\text{C}$, we use the extrapolated equation by Bigg (1953) down to the cloud droplet size (Levkov et al., 1992; Murakami, 1990).

$$Q_{frs} = C_{a1} \{ \exp[b_1 (T_0 - T)] - 1 \} \rho r_l^2 / p_w N_l \quad (10.41)$$

where the constants $a_1 = 100 \text{ m}^3 \text{s}^{-1}$ and $b_1 = 0.66 \text{ K}^{-1}$ are taken from laboratory experiments, $T_0 = 273.15 \text{ K}$ is the freezing point, $p_w = 1000 \text{ kgm}^{-3}$ is the density of water, ρ the air density, T the grid-cell mean temperature, r_l the in-cloud liquid water mixing ratio, N_l is the cloud droplet number concentration, and C the fractional cloud cover. In the standard version of ECHAM5, N_l is prescribed within the atmospheric boundary layer ($= 220 \cdot 10^6 \text{ m}^{-3}$ over land and $80 \cdot 10^6 \text{ m}^{-3}$ over sea, respectively). Above the boundary layer, N_l decreases exponentially to 50 m^{-3} in the upper troposphere over both land and ocean.

Brownian diffusion contact nucleation results from random collisions of aerosol particles with supercooled cloud droplets. It may be written as (e.g. Levkov et al. (1992))

$$Q_{frc} = C m_{io} F_1 D F_{ar} \quad (10.42)$$

where $m_{io} = 10^{-12} \text{ kg}$ is the initial mass of a nucleated ice crystal, $D F_{ar} = 1.4 \cdot 10^{-8} \text{ m}^2 \text{s}^{-1}$ the aerosol diffusivity (Pruppacher and Klett, 1978), and $F_1 = (4\pi R_{vl} N_l N_a) / \rho$. The concentration of active contact nuclei is approximated as $N_a = \max[N_{ao}(T_0 - T - 3), 0]$, with $N_{ao} = 2 \cdot 10^5 \text{ m}^{-3}$, and the mean volume droplet radius, R_{vl} , is obtained from

$$4/3\pi R_{vl}^3 N_l \rho_w = r_{lp} \quad (10.43)$$

Following Levkov et al. (1992), cloud ice is assumed to melt completely when $T > T_0$, giving

$$Q_{mli} = r_i / \Delta t \quad (10.44)''$$

(Roeckner et al. 2003: 69, 70).

Die Umsetzung des Schemas für stratiforme Wolken in der cloud.f90-Datei setzt sich aus verschiedenen Teilprozessen zusammen, die nacheinander abgearbeitet werden, wie die Kommentare der einzelnen Abschnitte des Programms der cloud.f90-Datei angeben. Der Schmelzvorgang Q_{mli} für die eingehende Niederschlagsmenge in den stratiformen Wolken ist unter Punkt 3.1 programmiert. Die Prozessabläufe der Datei lassen sich mit einem Flow Chart visualisieren, wie in Abbildung 16 dargestellt. Der Prozess des Schmelzvorgangs ist im Flow Chart mit einem Pfeil markiert.

„SUBROUTINE cloud

1. Top boundary conditions, air density and geopotential height at half levels
 - 1.1 Set to zero precipitation fluxes etc.
 - 1.2 Air density
 - 1.3 Geopotential at half levels
2. Set to zero local tendencies (increments)
3. Modification of incoming precipitation fluxes by melting, sublimation and evaporation
 - 3.1 Melting of snow and ice**
 - 3.2 Sublimation of snow and ice (Lin et al., 1983)
 - 3.3 Evaporation of rain (Rotstajn, 1997)
4. Sedimentation of cloud ice from grid-mean values. Updating the tendency 'pxite' to include sedimentation. At $jk=k_{lev}$, the sedimentation sink is balanced by precipitation at the surface (through 'zzdrs', see 7.3). Finally: In-cloud cloud water/ice.
5. Condensation/deposition and evaporation/sublimation
 - 5.1 Turbulence: Skewness - equation solved implicitly. This solver only works if phmixtau has non-zero timescale
 - 5.2 Turbulence: variance - equation solved implicitly
 - 5.3 Deposition/sublimation of cloud ice and condensation/evaporation of liquid water due to changes in water vapour and temperature (advection, convection, turbulent mixing, evaporation of rain, sublimation and melting of snow). Translate PDF laterally to calculate cloud after one timestep

- 5.4 Accounting for cloud evaporation in clear air and checking for supersaturation
 - 5.5 Change of in-cloud water due to deposition/sublimation and condensation/evaporation (input for cloud microphysics)
 - 6. Freezing of cloud water
 - 6.1 Freezing of cloud water for $T < 238$ K
 - 7. Cloud physics and precipitation fluxes at the surface
 - 7.1 Warm clouds: Coalescence processes after Beheng (1994): Auto-conversion of cloud droplets and collection of cloud droplets by falling rain. Accretion of cloud droplets by falling snow (zsac1) is calculated under 7.2
 - 7.2 Cold clouds: Conversion of cloud ice to snow after Levkov et al. 1992: Aggregation of ice crystals to snow and accretion of ice by falling snow. Accretion of cloud droplets by falling snow. Effective radius of ice crystals after Moss (1995)
 - 7.3 Updating precipitation fluxes. In the lowest layer (klev), the sedimentation sink of cloud ice is balanced by precipitation at the surface (through 'zzdrs'). Fraction of precipitating clouds (zclcpre) used for the calculation of evaporation/sublimation of rain/snow in the next layer
 - 8. Updating tendencies of t, q, xl, xi and final cloud cover
 - 8.10 Cloud cover scheme tendencies
 - 8.11 Simple linearized effect of cloud physics on skewness
 - 8.2 New skewness and variance
 - 8.3 Tendencies of thermodynamic variables Attn: The terms zxisub and zximlt do not appear in pxite because these processes have already been included in pxite via changes in cloud ice sedimentation (see 3.1, 3.2 and 4)
 - 8.4 Corrections: Avoid negative cloud water/ice
 - 9. Diagnostics
 - 9.1 Accumulated precipitation at the surface
 - 9.2 Total cloud cover
 - 9.3 Vertical integrals of humidity, cloud water and cloud ice
- END SUBROUTINE cloud“
- (MPI-Met 2005: cloud.f90, Kommentare)

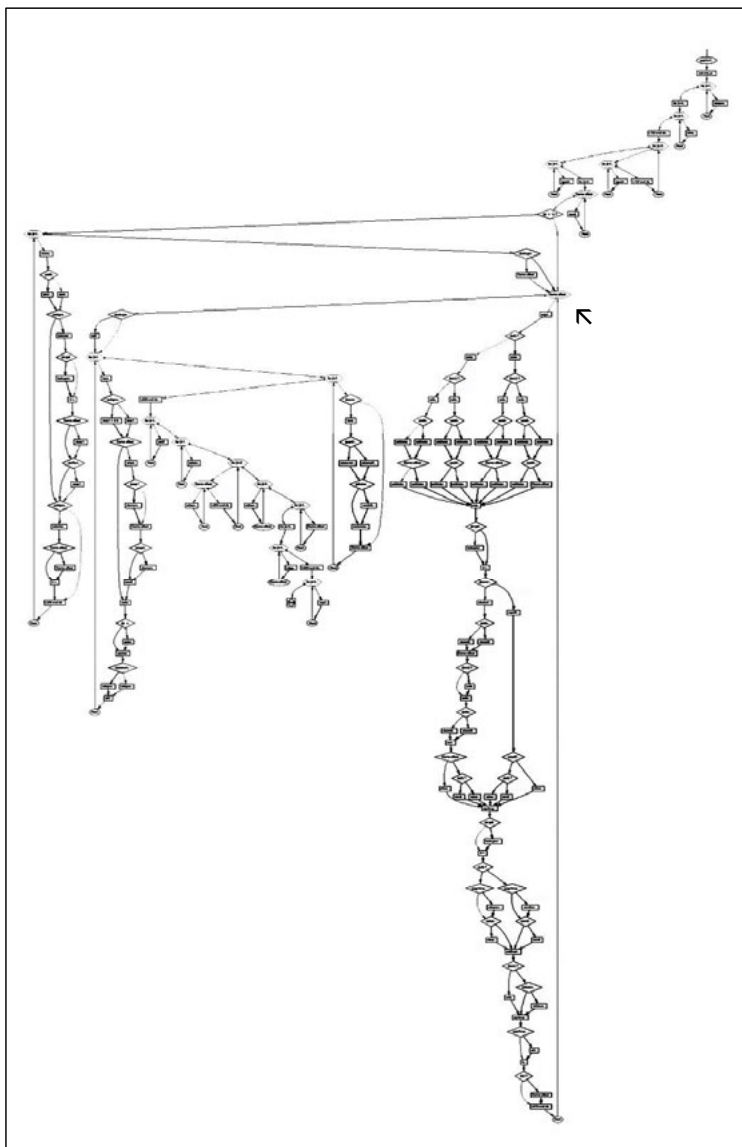


Abbildung 16: Flow Chart der Prozessabläufe der cloud.f90-Datei.
Der Pfeil markiert den Schmelzvorgang 3.1 (MPI-Met 15. Juli 2005)

```

! 3. Modification of incoming precipitation fluxes by
! melting, sublimation and evaporation
!
285: IF (jk .GT. 1) THEN
!
! DIR$ CONCURRENT
! DO 331 j1 = 1,kproma
!
290: ! 3.1 Melting of snow and ice
!
      zcons  = zcons2*zdp(j1)/(zlsdcp(j1)-zlvdcp(j1))
      ztdif  = MAX(0.0,ptml(j1,jk)-tmelt)
      zsnm1t = MIN(zxsec*zsfl(j1),zcons*ztdif)
295:   zrfl(j1) = zrfl(j1)+zsnm1t
      zsfl(j1) = zsfl(j1)-zsnm1t
      zsm1t(j1) = zsnm1t/(zcons2*zdp(j1))
      zximelt = MIN(zxsec*zxiflux(j1),zcons*ztdif)
      zxiflux(j1)=zxiflux(j1)-zximelt
300:   zxim1t(j1) =zximelt/(zcons2*zdp(j1))
      IF (ztdif.GT.0.0) THEN
        z1m1t(j1) = MAX(0.0,pxim1(j1,jk)+pxite(j1,jk)*ztmst)
      ELSE
        zim1t(j1) = 0.0
305:   END IF

```

Abbildung 17: Programmierung des Schmelzvorgangs in der *cloud.f90* Datei des Atmosphärenmodells ECHAM5 (MPI-Met 15. Juli 2005)

Die Programmierung, die nun dem Kommentar in ‚3.1 Melting of snow and ice‘ folgt, ist in Abbildung 17 dargestellt. Sie beschreibt explizit die Vorgänge, welche im mathematischen Modell in allgemeiner Form notiert sind. Um das Programm tatsächlich zum Laufen zu bringen, muss die mathematische Beschreibung in eine konkrete Narration übersetzt werden. Im Falle des Schmelzens von Eis entspricht der mathematische Term Q_{mli} dem Ausdruck (zsnm1t) in Codezeile 294. Der Wert für (zsnm1t) wird im Programm durch eine Minimum-Funktion¹³ aus den beiden Argumenten (1) vorhandenes Eis/Schnee (zxsec*zsfl(j1)) und (2) Temperaturänderung (zcons*ztdif) gewonnen, analog der mathematischen Beschreibung in Formel 10.44: $Q_{mli} = r_i/\Delta t$, welche die Schmelzrate als Veränderung des Wolkeneises im Laufe der Zeitveränderung beschreibt.¹⁴ Um die konkrete Parametrisierung im Rahmen des Teilprozesses „3. Modification of incoming precipitation fluxes by melting, sublimation and evaporation“ zu programmieren, bedarf es folgender Spezifikationen: (1) zxsec*zsfl(j1): Wie viel Eis/Schnee (zsfl) ist in einer

13 Minimum- und Maximumfunktionen sind Funktionen zur Auswahl eines Wertes von zwei Werten. Die Funktion liefert dann einen Wert zurück: den kleinsten oder größten.

14 Zu beachten ist, dass die Schmelzrate von Wolkeneis Q_{mli} wiederum selbst ein Teil des Terms zur Berechnung des Wolkeneises ist: [Änderung der Flüssigkeit r_l in Wolken] $\partial r_l/\partial t = Q_{rl} + Q_{mli} + Q_{mlis} + Q_{end} + Q_{tbl} - Q_{evl} - Q_{frh} - Q_{frs} - Q_{fre} - Q_{aut} - Q_{rac1} - Q_{sac1}$ (10.2).

Schicht (jl) vorhanden, das überhaupt schmelzen kann (zsfl (jl))? Der Wert ergibt sich aus den vorhergegangenen Berechnungen des Wolken-eises (zsfl). Um jedoch zu verhindern, dass der Wert des verfügbaren Eises null sein kann – denn dies würde bei einer Division zu mathematischen Problemen führen – wird dieser mit einem Sicherheitsfaktor (zxsec) multipliziert, der etwas kleiner als 1 ist und sich wie folgt ergibt: $zxsec = 1.0 - zepsec$, wobei $zepsec = 1.0e^{-12}$ ist, also $1,0 - 1,0 \cdot 10^{-12}$.¹⁵

(2) (zcons*ztdif): Welche Temperaturveränderung (ztdif) liegt konkret vor, um das Schmelzen nun in Gang zu bringen? In anderen Worten: Wie viel Energie steht für das Schmelzen von Eis zur Verfügung? Die Temperaturveränderung wurde in der vorhergehenden Codezeile 293 als Maximum-Funktion wie folgt berechnet: $ztdif = \text{MAX}(0.0, ptm1(jl,jk) - tmelt)$. Dabei ergibt sich die Temperaturveränderung aus dem Eingabewert der Temperatur (ptm1) zwischen zwei Schichten (jl, jk), der entweder zu Beginn aus Messungen stammt oder im Laufe der Simulation einen Zeitschritt zuvor berechnet wurde, abzüglich der Schmelztemperatur (tmelt) von 273,15 Kelvin, also 0° Celsius. Als Maximum-Funktion kann die Temperaturveränderung (ztdif) an dieser Stelle im Code nie unter 0° Celsius sinken.

Die so berechnete Temperaturveränderung wird mit einer Konstante (zcons) multipliziert, die ebenfalls zwei Codezeilen zuvor, in Zeile 292, berechnet wurde: $zcons = zcons2 * zdp(jl) / (zlsdcp(jl) - zlvdcp(jl))$. Diese Konstante (zcons) wird als Verhältnis von $(zcons2 * zdp(jl))$ zu $(zlsdcp(jl) - zlvdcp(jl))$ angegeben, wobei $zcons2 = 1. / (ztmpst * g)$ sich aus dem Verhältnis von 1 zur Zeitschrittweite (ztmpst) von $1 / \Delta t$ multipliziert mit der Gravitationsbeschleunigung (g) von $9,80665 \text{ m/s}^2$ ergibt. Das Ganze (zcons2) wird dann mit der Dicke (zdp) der Schicht (jl) multipliziert ($zcons2 * zdp(jl)$). Der zweite Teil des Ausdrucks $(zlsdcp(jl) - zlvdcp(jl))$ berechnet als erstes den Wert der Sublimationswärme ($zlsdcp(jl)$), der sich an anderer Stelle in der Datei als $zlsdcp = als * zrcp$ ergibt, also als Sublimationswärme (als) von 2.8345^6 J/kg multipliziert mit dem Wert der spezifischen Feuchte und Wärme (zrcp), der ebenfalls in der Datei an anderer Stelle als $zrcp = 1. / (cpd + zcons1 * \text{MAX}(pqm1(jl,jk), 0.0))$ berechnet wird.¹⁶ Von dem so

15 Der Wert des verfügbaren Eises wird null, wenn im Laufe der Berechnungen zwei gleiche Größen miteinander dividiert werden. Um diesen Fall im weiteren Verlauf zu vermeiden, wird der ‚security parameter‘ (zxsec) an dieser Stelle eingeführt. Insgesamt gibt es in dieser cloud.f90-Datei drei verschiedene Sicherheitsparameter.

16 $zrcp = 1. / (cpd + zcons1 * \text{MAX}(pqm1(jl,jk)$ stellt das Verhältnis von 1 zur spezifischen Wärme trockener Luft bei konstantem Druck (cpd) von $1005,46 \text{ J/K/kg}$ addiert mit der Konstante (zcons1) dar. $zcons1 = cpd * vtmpc2$ wiederum ist das Produkt der spezifische Wärme trockener

errechneten Wert der Sublimationswärme ($zlsdcp(jl)$) wird nun der Wert der Verdunstungswärme ($zlvdcp(jl)$) subtrahiert, der sich als Produkt ($alv * zrcp$) der latenten Wärme bei Verdunstung (alv) von 2.5008^6 J/kg multipliziert mit ($zrcp$), wie bereits beschrieben, ergibt.

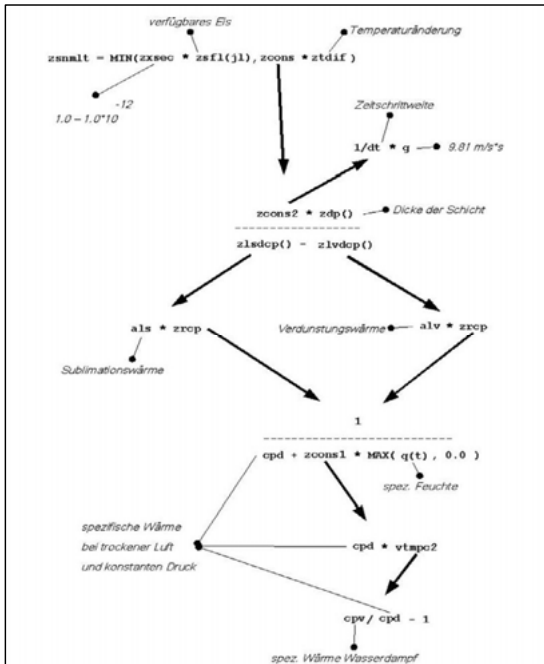


Abbildung 18: Schmelzvorgang (Codezeile 292 bis 294) der cloud.f90 Datei des Atmosphärenmodells ECHAM5 (MPI-Met 15. Juli 2005)

Der ausbuchstabierte Code der Zeilen 292 bis 294 der cloud.f90-Datei, wie in Abbildung 18 dargestellt, errechnet das Resultat des Schmelzvorganges, um dann in Codezeile 295 ($zrfl(jl) = zrfl(jl) + zsmelt$) das zum Berechnungszeitpunkt in der Wolke enthaltene Wasser ($zrfl$) aus dem bislang verfügbaren Wolkenwasser zuzüglich des geschmolzenen Eises zu berechnen. Analog dazu wird in Codezeile 296 ($zsfl(jl) = zsfl(jl) -$

Luft bei konstantem Druck multipliziert mit der dimensionslosen Konstante ($vtmpc2$). $vtmpc2 = cpw/cpd - 1$, ergibt sich aus dem Verhältnis der spezifischen Wärme von Wasserdampf bei konstantem Druck (cpw) von $1869,46 \text{ J/K/kg}$ zur spezifischen Wärme trockener Luft bei konstantem Druck minus 1. Der Ausdruck ($cpd + zcons1$) wird schließlich mit ($\text{MAX}(pqm1(jl,jk), 0.0)$) multipliziert, wobei ($pqm1$) die maximale spezifische Feuchte bezeichnet, welche als Wert dem Modell vorgegeben wurde.

zsnmlt) das aktuelle Wolkeneis (zsfl) aus dem bisher verfügbaren Wolkeneis (zsfl) minus dem geschmolzenen Eis (zsnmlt) berechnet. Während der Berechnung wird mit Werten aus Konstanten, aus Eingabedaten – durch Messung bei der Initialisierung vorgegeben oder im Zeitschritt zuvor berechnet – und aus temporären Daten, die erst im Laufe der weiteren Berechnung neu berechnet werden, gearbeitet. Ziel der Datei ist es, Ausgabewerte für Schnee, Regen, Temperatur und Wolkenbedeckung zu errechnen. Diese Werte werden dann für die weiteren Berechnungen des Atmosphärenmodells verwendet.

Der Zoom in den Code macht deutlich, dass das Programm – im Unterschied zum mathematischen Modell – eine explizite Geschichte erzählt, welche für die Berechnung mit numerischen Werten konkretisiert wird. Dies ist mit ‚story telling with code‘ gemeint. Die Geschichte von Codezeile 292 bis 296 in Prosa erzählt, lautet: Wenn sich das Verhältnis von Druck, Sublimations- und Verdunstungswärme sowie Feuchtigkeit verändert (Zeile 292), dann verändert sich die Temperatur in der Wolkenschicht (Zeile 294). Falls die Temperatur dabei über den Gefrierpunkt steigt (Zeile 293), schmilzt das Wolkeneis (Zeile 294) und das Verhältnis von Eis und Wasser in der Wolkenschicht verändert sich im Unterschied zum vorherigen Zeitschritt (Zeilen 295, 296). Bei ausreichender Menge und Schwere der Wassertropfen, fängt es unter entsprechenden Umständen an zu regnen. Der Schmelzvorgang wird für jede Gitterbox und für alle Schichten des Modells berechnet und ist nur einer von zahlreichen Prozessen der Wolkendatei, die insgesamt 1.110 Zeilen Code umfasst. Um die Geschichte numerisch erzählen, also simulieren zu können, wird jeder Konstante ein numerischer Wert zugewiesen und für jede Variable gilt es, eine konkrete Berechnungsprozedur anzugeben, die mit Messwerten oder berechneten Werten des Zeitschritts zuvor initialisiert wird. Die Berechnungsprozedur entspricht dabei nicht eins zu eins dem mathematischen Modell. „Die eigentliche Parametrisierung,“ wie ein Klimaforscher im Interview erklärte, „muss man sich aus dem Quellcode herausziehen“ (Interview 18, 2005). Genau hier liegt die Arbeit des Modellierers, denn das mathematische Modell beschreibt die konzeptuelle Herangehensweise in einer wesentlich allgemeineren Version als der Code. Daher sind das mathematische Modell auf dem Papier und das programmierte Computermodell nicht identisch, zumal das Computermodell mit einer diskretisierten, computertauglichen Version des mathematischen Modells arbeitet. Diese Version unterliegt Verzerrungseffekten, wie beispielsweise, dass die verschiedenen Ströme eines Klimamodells von Schicht zu Schicht springen und sich nicht kontinuierlich bewegen. Am folgenden Beispiel der Fehlersuche im Modell werden einige dieser Verzerrungseffekte deutlich.

Fehlersuche im Modell

Bei der Vielzahl der Prozesse eines Modells, in welchen sich die verschiedenen Konstanten und Variablen oft wiederholen, ist es nicht einfach, den Überblick zu behalten. Daher können sich leicht Fehler in die Programmierung einschleichen. Diese Fehler zu erkennen, ist nicht nur schwierig, sondern mühsam und zeitaufwendig. Die Fehlersuche ist jedoch ein wichtiger Teil der Modellierungsarbeit, denn jeder Fehler kann mehr oder weniger großen Einfluss auf die Resultate und damit auf die Erklärungs- und Prognosegüte des Modells haben. Letztendlich ist es die Güte eines in-silico Experimentalsystems, welche die Evidenz der gewonnenen Ergebnisse garantiert. Anhand der Fehler zeigt sich aber auch die Diskrepanz zwischen mathematischem Modell und Programm. Diese Diskrepanz ist Ausdruck der Zugeständnisse an die Logik der Rechner und lässt die Eigenart der Beschaffenheit dieser in-silico Experimentalsysteme zu Tage treten.

Ein Beispiel für eine zwei Jahre andauernde Modellentwicklung mit einer nahezu einjährigen Fehlersuche ist die Modellierung eines Vulkanausbruchs im Atmosphärenmodell ECHAM5. Dabei handelt es sich nicht um die Modellierung einer Vulkaneruption, sondern um die Simulation der Freisetzung langlebiger Stoffe wie Asche und Schwefeldioxid in der Stratosphäre und den Folgen für das Klima. „Eine Vulkanexplosion sieht bei uns so aus, dass es in einer Gitterbox in der Stratosphäre, also etwa in 25 Kilometer Höhe, ‚plupp‘ macht und dann ist da auf einmal ganz viel Asche und Schwefeldioxid“ (Interview 25, 2008). Vorbild für die Lokalisierung der Gitterbox, die Höhe der Freisetzung, die zeitliche Entwicklung wie auch die Menge der freigesetzten Stoffe war der Ausbruch des philippinischen Vulkans Pinatubo im Juni 1991.¹⁷ Er diente als Beispiel, um die Modellierung zu evaluieren. Allerdings waren für das Modell nur feine Asche und Sulfataerosole von Interesse, da sich diese länger in der Atmosphäre halten und daher einen klimarelevanten Einfluss haben. Dieser Einfluss entfaltet sich in den chemischen und physikalischen Prozessen infolge der Emission. Das freigesetzte Schwefeldioxid wird in der Atmosphäre in gasförmiges Sulfat und weiter in Sulfat-Aerosole umgewandelt. Die Aerosole werden von den

17 Der Ausbruch des Pinatubo zog sich vom 7. bis 15. Juni 2001 hin, mit einer Haupteruption am 15. Juni. Bei dieser wurden 100 Mega Tonnen Asche und Staub, ca. 1% der gesamten Eruptionsmasse, und 17 Megatonnen Schwefeldioxid in die Stratosphäre geschleudert, die sich drei Jahre lang als Wolken in der Stratosphäre hielten und die Sonneneinstrahlung reduzierten. Das führte zu einem globalen Temperaturabfall von 0,4 °C und zur Verringerung des Ozons.

atmosphärischen Strömungen transportiert. Sie beeinflussen die Strahlungsbilanz und damit die meteorologischen Prozesse. Die Programmierung der Freisetzung von Asche und Schwefeldioxid in der Stratosphäre erfolgte im Rahmen des bestehenden Atmosphärenmodells ECHAM5 sowie des Aerosolmodells HAM, wobei die chemischen und aerosolmikrophysikalischen Prozesse mit Hilfe von HAM und die Transportprozesse der Aerosole mit ECHAM5 berechnet wurden.

„Also man könnte in ECHAM beispielsweise angeben, dass man nur mit passiven Tracern rechnet, wie z.B. feiner Asche, die nicht chemisch reagiert sondern nur emittiert und vom Modell transportiert wird und in Form von Absinkprozessen, also Sedimentations- und Depositionsprozessen aus der Atmosphäre entfernt wird. Weiter gibt es zusätzlich die Möglichkeit mit HAM, also mit dem Aerosol-Modell zu rechnen. HAM hat fest definierte Tracer (Schadstoffe). Das ist Staub, Seesalz, Ruß, also ‚Black Carbon‘ und ‚Organic Carbon‘, Sulfat. [...] Oder man rechnet mit Chemie, die umfasst bis zu hundertfünfzig Reaktionen und sechzig verschiedene Tracer, also sechzig verschiedene chemische Stoffe. (Und HAM oder das Chemiemodell geben die Ergebnisse dann an ECHAM weiter?) Ja, HAM berechnet nur die Aerosolmikrophysik, also die Umwandlung von gasförmig ins Aerosol und die Reaktion der Teilchen untereinander. Am Anfang entstehen kleine Teilchen, die wachsen mit der Zeit. Das passiert alles in HAM. Aber alle meteorologischen Prozesse, sprich der Transport, auch die Wechselwirkung mit Wolkenwasser und Regen, geschieht in ECHAM. Entsprechendes gilt auch für die Chemie. Die Chemie berechnet die Reaktionen der Stoffe untereinander, aber der Transport findet in ECHAM statt. [...] Im Prinzip haben wir in ECHAM je eine Variable für diese Tracer. Zu jedem Zeitpunkt und in jeder Gitterbox wird die Veränderung der Konzentration (Deltas), z.B. durch Transport oder Emission, ausgerechnet. Für die Vulkaneruption geben wir die gesamte emittierte Masse und die Eruptionszeit vor und berechnen daraus einen Fluss, also wie viel pro Sekunde emittiert wird. Die emittierte Konzentration pro Zeitschritt addieren wir auf das Delta, welches schon Konzentrationsänderungen durch andere physikalische Prozesse aus HAM und ECHAM enthalten kann. Am Ende des Zeitschrittes wird die Änderung der Konzentration, also das Delta, auf die Konzentration des Tracers in ECHAM aufaddiert. Der Tracer ist schon im Modell enthalten“ (Interview 25, 2008).¹⁸

Allerdings zeigte sich schnell, dass die Berechnungen nicht stimmten. „Es wurde viel zu viel Schwefeldioxid (SO₂) in Sulfat umgewandelt und

18 „Zu dem Tracer SO₂ (Schwefeldioxid) fügt man eine gewisse Menge hinzu, so wie das am Boden für alle anderen Emissionen auch gemacht wird. Das ist fast das gleiche nur sind es Emissionen vom Straßenverkehr oder Waldbrände. Bei uns ist der Vulkan und es ist in einer ganz anderen Höhe“ (Interview 25, 2008).

das SO_2 hatte eine viel zu geringe Lebensdauer. Wir hatten eine Lebensdauer von knapp 14 Tagen und wir brauchten eigentlich eine von gut 30 Tagen“ (Interview 25, 2008), wie die empirischen Daten des Pinatubo-Ausbruchs vorgaben.¹⁹ Die Fehlersuche konzentrierte sich daher zunächst auf den Umwandlungsprozess von Schwefeldioxid in Sulfat und anschließend auch auf die Lebensdauer des Sulfats, denn „das Sulfat fiel [im Computerexperiment] wie Steine vom Himmel. Anstatt einem halben Jahr Halbwertszeit waren es bei uns nur ein paar Wochen. [...] Das legte die Vermutung nahe, dass es eventuell zu große Teilchen waren. Dann die Frage: Warum?“ (Interview 25, 2008). Um die Fehlerquellen zu entdecken, benötigten die Modellierer fast ein Jahr Detektivarbeit. Es wurden Vermutungen aufgestellt, überprüft und wieder verworfen bis die Fehler gefunden waren und behoben werden konnten.

„Das eine Problem war, dass sich die Vulkanwolke in dieser großen Höhe in der Stratosphäre befindet. 25 km, oder 30 bis 50 hPa, das ist eine Höhe, die in einem normalen Modell an der Obergrenze ist. Deswegen mussten wir eine andere Modellversion nehmen, nämlich das Modell der Mittleren Atmosphären, und das wurde zuvor noch nie mit HAM gekoppelt. Da HAM keine eigene Chemie rechnet, muss man beispielsweise die Konzentration der Hydroxyl-Radikale (OH) dem Modell vorgeben, um Schwefeldioxid (SO_2) umzuwandeln. Wir brauchten also neue Daten, die diese Höhe miterfassten und die waren erstmal nicht in Ordnung. Es hat relativ lange gedauert, um genau herauszufinden: Da ist der Fehler! Die [Mengen] waren um den Faktor zwei zu groß und dadurch wurde viel zu viel Schwefeldioxid in Sulfat umgewandelt, wodurch die viel zu geringe Lebensdauer entstand. [...] Nachdem wir dann wussten, ok das Schwefeldioxid hat die Lebenszeit wie wir sie haben wollen, wurde genauer auf das Sulfat geschaut. Dabei merkten wir, das Sulfat fiel wie Steine vom Himmel. [...] Wir haben in den Prozessen der Sedimentation [von ECHAM] gesucht, da wir vermuteten, vielleicht ist hier etwas nicht in Ordnung, z.B. ungenaue numerische Prozesse. [...] Die Sedimentation beschreibt über die Fallgeschwindigkeit, wie schnell und wie viele Teilchen nach unten fallen. Das Ergebnis hängt von der Modellauflösung ab, wie viele einzelne vertikale Schichten man hat. Innerhalb einer Schicht wird alles als gleich verteilt angenommen: Ein Teilchen, oder eine bestimmte Menge, kommt von oben herunter und wird als gleichverteilt in dieser Box angesehen. D.h. es kommt ganz schnell etwas unten in der Box an und kann im nächsten Zeitschritt schon in die nächste Box fallen, während es in Wirklichkeit noch weiter

19 „Wir haben alles am Pinatubo validiert. Wir wissen ungefähr wie groß die Teilchen werden und auch wie lange sich die Sulfate in der Stratosphäre halten bzw. welche Wechselwirkungen mit der Strahlung stattfinden. Dafür gibt es gemessene Werte und die können wir natürlich vergleichen“ (Interview 25, 2008).

oben bliebe und die ganze Schicht langsam durchwandern müsste. So eine Schicht ist ca. zwei Kilometer hoch. Wenn man nun statt mit 39 Schichten mit 90 rechnet, sieht man einen deutlichen Unterschied. Man hat dreimal so viele Schichten und die Teilchen fallen dadurch langsamer.

Aber wir haben gesehen, das war es nicht! Wir merkten, dass auch mit der höheren Auflösung die Konzentration der Teilchen immer noch viel zu schnell abnahm. Sie waren alle immer viel zu schnell weg. In den Ergebnissen ließ sich anhand von Plots erkennen, dass eine Aerosolwolke schnell nach unten wanderte. Wir machten Abbildungen mit zweidimensionalen Schnitten durch das Modellgebiet und konnten sehen, wie der Durchmesser der Teilchen schnell größer wurde und dass sie einfach zu groß waren. Dann wurde relativ schnell klar, dass es ein Problem in der Aerosolmikrophysik [des HAM Modells] sein musste. [...] Ein Kollege machte dann den Vorschlag, dass es ein Boxmodell für HAM gab. Das rechnet nur eine Gitterbox aus. Das ist natürlich viel einfacher, viel schneller. Man kann viel direkter sehen, was passiert. Mit diesem Boxmodell haben wir dann auch erkannt, dass etwas nicht richtig funktionierte, dass ein physikalischer Prozess überhaupt nicht angesprochen wurde, weil wir außerhalb des Gültigkeitsbereiches waren. Wir hatten das als Effekt eigentlich im Prinzip schon immer gesehen, auch vorher, da in unserer Wolke in der Mitte irgendwie immer ein Loch mit sehr niedrigen Konzentrationen war. [...] Mit dem anderen [Boxmodell] konnte man ganz schnell mal Schalter im Code ein- und ausschalten und sehen, was passiert. Und dann stellte ich fest, hier stehen nur Nullen. Wir haben dann eine finnische Kollegin, die für diese Nukleationsprozesse – das war das eben – verantwortlich war, angeschrieben und ihr das Problem erklärt. Sie hat uns Formeln und auch ein bisschen Code geschickt und meinte, wir sollten es mal damit versuchen. Das habe ich dann für uns passend umgesetzt und danach merkten wir schon: Ja, es wird besser! Es war aber noch nicht gut. Also mussten wir natürlich weiter suchen. Man hat immer wieder überlegt, was es denn sein könnte? Was kann man denn noch machen? Immer wieder hatte jemand eine Idee. Dann schmeißt man das Modell wieder an und probiert es nochmal. Und sieht im Ergebnis wieder: Es hat nichts gebracht. Das war ziemlich frustrierend“ (Interview 25, 2008).

Der Durchbruch ergab sich schließlich, als sich herausstellte, dass es im Institut vier verschiedene Boxmodelle der Aerosolmikrophysik gab. HAM, das Teil von ECHAM war, das Boxmodell (B2) des Kollegen, das bereits für Vergleiche mit HAM in Tests verwendet worden war, sowie zwei weitere Modelle (B3, B4). „Ein Kollege war dabei, eine andere Mikrophysik in ECHAM einzubauen und dazu gab es auch ein Boxmodell [B3]. Und schließlich hatte eine Kollegin ursprünglich auch mal ein Mikrophysikmodell geschrieben, und dazu gab es ebenfalls ein Boxmodell [B4]“ (Interview 25, 2008). Charakteristische Ergebnisse, die sich aus der Simulation mit HAM ergeben hatten, wurden nun mit allen vier

Boxmodellen getestet. Die Ergebnisse unterschieden sich erheblich von HAM und es zeigten sich verschiedene Hinweise zur Lösung des Problems. „Wir stellten fest, dass wir sehr stark vom Zeitschritt abhängen. Das galt nicht nur für uns (HAM), das galt zum Teil auch für die anderen Modelle“ (Interview 25, 2008). Darüber hinaus wurde deutlich, dass die verschiedenen Teilchengrößen (Moden) des Sulfats im HAM Modell „sehr schön für die Troposphäre sind, dass es aber überhaupt nicht der Situation in der Stratosphäre entspricht. Und überhaupt nicht der Situation, die sich mit so einem Vulkan ergibt“ (Interview 25, 2008). Die Resultate des HAM Modells ergaben eine breite Kurve der Teilchengröße im Verhältnis zur Anzahl der Teilchen und wiesen damit auf die Bildung sehr kleiner als auch sehr großer Teilchen hin. „Und die viel größeren fielen eben wie Steine vom Himmel. Dadurch haben wir so viel Masse verloren“ (Interview 25, 2008). Im Unterschied dazu ergaben die Resultate des Boxmodells B2 eine schmale Kurve, also einen schmalen Bereich von Teilchengrößen.

„Dann haben wir angefangen, die Einstellung der Moden zu verändern und sind auf Ergebnisse gekommen, die den anderen Modellen sehr ähnlich waren. Wir mussten dann noch ein bisschen Rücksicht auf die Strahlungsparametrisierung nehmen, damit wir innerhalb des Gültigkeitsbereichs blieben. Und haben dann Einstellungen gefunden, die erheblich bessere Ergebnisse hatten. Dazu kam dann noch, dass inzwischen in HAM ein anderes Zeitintegrationschema eingebaut wurde, das nicht mehr so empfindlich auf diesen Zeitschritt [der Modellsimulation] reagierte. Damit hatten wir dann eine Version, die recht zufriedenstellende Ergebnisse brachte. [...] Dadurch sind wir jetzt an einem Punkt, dass wir sagen: So, den Pinatubo können wir ganz gut simulieren, auch die Strahlungswechselwirkungen. Das sieht gut aus“ (Interview 25, 2008).

Nachdem das Modell für die Reproduktion des Pinatubo Ausbruchs gut funktionierte, kann es nun für die Simulation weiterer Vulkanausbrüche verwendet werden. Im Kontext der Forschung ist die Integration von Vulkaneruptionen noch weitgehend Neuland. „Es gibt verschiedene Gruppen, die versucht haben, den Einfluss des Sulfats zu berechnen. Aber so wie wir das jetzt machen, dass wir die Asche und das Sulfat haben und zudem richtig mit einer Aerosolmikrophysik rechnen, das hat eben noch keiner gemacht. Deswegen hatten wir auch so viele Probleme und mussten sehen, was alles passiert“ (Interview 25, 2008). Das eigentliche Ziel ist es, Supervulkanausbrüche zu simulieren, die um den Faktor 100 stärker sind als der Ausbruch des Pinatubo. Erhöhte Messwerte im Yellowstone Park, in dem ein Supervulkan vermutet wird, lassen die

Vulkanforscher weltweit aufmerken, und es stellt sich die Frage: Was wäre wenn?

„Dafür hat man Modelle. Das sind diese schönen Spielchen, die man eben nur mit Modellen machen kann. Und insofern ist das eigentliche Ziel, so eine Megaeruption zu berechnen. Und dann zu sehen, was für Auswirkungen hat das? [...] Deshalb möchten wir gleich mal gerne einen richtigen Supervulkan rechnen. ‚Mach das doch mal vor den Ferien.‘ Und, wie das so ist: Es geht nicht! Dieses mal ist es ECHAM, das wissen wir schon. Irgendetwas in der Konvektion streikt und es ist ein richtiger Modellcrash. Das Modell steigt richtig aus“ (Interview 25, 2008). Das Problem liegt in der Wechselwirkung mit der Strahlung, die einer großen Detailtiefe in der Modellierung bedarf. „Ich habe vor einem Jahr schon einmal eine Supervulkaneruption berechnet. Und damals ging das. Da hatte ich noch keine Wechselwirkung mit der Strahlung. D.h. in der Gegend wo diese dunkle Wolke aus Vulkanasche und Sulfat war, heizt sich jetzt alles auf. Und die Prozesse, die das auslöst, waren vor einem Jahr lange nicht so im Detail da, wie jetzt. Entsprechend bin ich damals ohne Probleme [Crash] durchgekommen, außer dass ich bereits gesehen hatte, dass die ganze Masse, die oben reingegeben wurde zu schnell wieder weg war. Im ersten Monat nach der Eruption wurde es eiskalt in Nordamerika und im nächsten Monat war die Temperatur schon fast wieder normal. Da war schon klar, da stimmt was nicht“ (Interview 25, 2008).

Das Beispiel macht deutlich, wie Modellierung im Forschungsalltag vorgeht und welcher Eigenlogik die komplexen in-silico Experimentalsysteme unterliegen. Dass jedoch eine solche Form der computerbasierten Forschung überhaupt möglich ist, liegt nicht zuletzt an der zunehmenden Standardisierung der Modelle und Daten in der Meteorologie. Denn vor allem der Modellvergleich ist eine wichtige Basis der Evaluation der Modelle und ihrer Ergebnisse. Die Klimaforschung kann wie kaum eine andere Wissenschaft auf eine Geschichte der Professionalisierung verweisen, welche sich in der weltweiten Synchronisierung und Standardisierung der Modellentwicklung zeigt. Motor dieser Professionalisierung ist die soziopolitische Frage: Was wäre wenn?