

3 ENTSTEHUNG DER COMPUTATIONAL SCIENCES

Mit dem Erscheinen der elektronischen Computer und ihrer weltweiten Verbreitung verändert sich die wissenschaftliche Forschung wie auch Forschungslandschaft. Keine Disziplin und keine wissenschaftliche Methode bleiben von der Nutzung der Computer unberührt. Wissenschaftler und Ingenieure setzen den Computer als Forschungs-, Experimentier- und Prognoseinstrument in der täglichen Arbeit ein und Datenleitungen sorgen für eine globale Vernetzung der computerbasierten Forschung. Heute ist nahezu jedes naturwissenschaftliche Institut und Labor an ein Rechenzentrum angeschlossen, in jedem Messinstrument finden sich Computerchips und Theorie wird zunehmend in Algorithmen transformiert. Computer und Algorithmen verwandeln die Welt der Phänomene und Fakten in eine Parallelwelt aus PetaBytes von Daten. Ein so grundlegender Wandel des wissenschaftlichen Instrumentariums bleibt nicht ohne ebenso grundlegende Auswirkungen auf die Epistemik wie Praktik des Forschens. Dabei handelt es sich um nichts weniger, als die zweite Hälfte der wissenschaftlichen Revolution der Neuzeit.

„John von Neumann foresaw that the equations describing scientific phenomena, once expressed in mathematical terms, could be solved numerically, without recourse to routine or repetitive experiment. This vision is nothing less than the second half of the scientific revolution. Throughout four centuries we have expected that a successful scientific theory would have its major concepts expressed quantitatively as numbers and its major relationships expressed as mathematical equations; the truth of this theory was settled by experimental tests and hand calculations, often in idealized situations. The second half of the scientific revolution is no less sweeping in its goals. The solutions of the equations are also to be obtained on mathematical grounds, by numerical computation, without restriction to idealized cases“ (Glimm 1990: 185).

Mit Rechnern experimentieren

Ziel dieser Vollendung der wissenschaftlichen Revolution in Form der Computational Sciences ist zum einen die Substitution herkömmlicher Experimente durch Computereperimente, zum anderen die Konkretisierung der bis dahin stark idealisierten Modelle, sowohl bezüglich der prozessualen Komplexität als auch realitätsnahe Details. Mit Hilfe der Computer können realitätsnahe Modelle berechnet und ausgewertet werden. Der Erkenntnisgewinn liegt dabei in der Erforschung komplexer Wechselwirkungen anhand numerischer Studien des Lösungsverhaltens der simulierten Modelle. Triebfeder der Entwicklung ist die zunehmende Anwendbarkeit der Modelle, denn sowohl in den Ingenieurs- wie Naturwissenschaften genügt das rein qualitative Verständnis von Prozessen auf abstrakter Ebene nicht mehr, wie das Beispiel der Strömungsdynamik zeigte. Die so genannten ‚first principles‘ – mathematische Darstellungen naturwissenschaftlicher Gesetze wie Eulers Bewegungsgleichung basierend auf Newtons zweitem Axiom – sind zu stark idealisiert, als dass sie auf Anwendungskontexte übertragen werden könnten. Computereperimente bieten nun erstmals die Möglichkeit, die idealisierten Modelle mit empirischen Details anzureichern und in ihrem Verhalten zu erforschen. Die Folge ist die zunehmende Verknüpfung von qualitativ-theoretischem und quantitativ-empirischem Verstehen komplexer Prozesse als charakteristische Epistemik einer computerbasierten Wissenschaft. Diese Epistemik war bislang als Wechselspiel zwischen idealisierten Modellen und „numerous experiments, from which empirical formulae have been constructed“ (Stokes 1845: 76) realisiert. Mit dem Problem, dass „such formulae, although fulfilling well enough the purposes for which they were constructed, can hardly be considered as affording us any material insight into the laws of nature“ (Stokes 1845: 76). Diese Epistemik verknüpft sich nun im Computer zu einer neuen Form von Theorieexperimenten. Computereperimente sind eine Mischung aus Theorie basierend auf first principles, empirischen Annahmen und Heuristiken. Sie führen nicht nur ein neues Verhältnis zwischen bestehender Theorie, Experiment, Messung und Beobachtung ein, sie vereinen dieses neue Verhältnis in sich und implementieren es in ihrer charakteristischen Forschungslogik. Heutige Computereperimente beinhalten jedoch wesentlich mehr als nur die diskretisierten Grundgleichungen, wie dies von John von Neumann gedacht und mit den damaligen Rechnern auch nur möglich war. Computereperimente werden nicht nur mit umfangreichen Messdaten für spezifische Randbedingungen initialisiert, sie berücksichtigen auch immer zahlreiche Aspekte der

Anwendungskontexte in Form subskaliger Parametrisierungen.¹ Vor allem letzteres unterscheidet aktuelle Computerexperimente von den ersten computerbasierten Berechnungen der 1940er Jahre und lässt die heutigen Simulationsmodelle in ihrem Umfang erheblich anwachsen.

„Man hat verschiedene Namen für diese Verbindung von Modellierung und Berechnung geprägt: computational science oder Computersimulation oder einfach, wie der russische Mathematiker A.A. Samarskij, Computerexperiment. Er unterscheidet drei Phasen bei so einem Experiment: Modell – Algorithmus – Programm (MAP, vielleicht die Abbildung = map der Wirklichkeit in den Computern). Algorithmen sind die Vorschriften zur (im allgemeinen näherungsweise) Lösung der Modellgleichungen; Programm meint die Realisierung dieser Vorschriften, die Erstellung von Software. Programme sind letztendlich auch die Instrumente dieses Experiments; sie werden von Anwendern oft als Black box, das heißt ohne genaue Kenntnis des Modells und des Algorithmus, benutzt. Dabei verliert der Experimentator leicht die Mathematik aus den Augen, und, nur wenn etwas nicht funktioniert, etwa wenn das Experiment unglaubliche Werte liefert (oder jemand anderes mit anderen Instrumenten andere Ergebnisse erzielt), beginnt man über die ‚Richtigkeit‘ des Modells, über die Korrektheit der Algorithmen nachzudenken“ (Neunzert 1995: 49, 50).

Dass man bei den ersten Computerexperimenten die Mathematik aus den Augen verloren hätte, kann man nicht behaupten. Tatsächlich war zu Beginn der Computerentwicklung die Umsetzung der Modelle in maschinen-taugliche Befehle eine rein mathematische Angelegenheit, da direkt – ohne Zwischenebenen wie Betriebssysteme, Software, spezifische Hardwarefunktionen – mit den Computern gearbeitet wurde. „Applying the modern term ‚to program‘ to a computer probably originated with the ENIAC team at Moore School. More often, though, they used the phrase ‚set up‘ to describe configuring the ENIAC to solve different problems. Setting up the ENIAC meant plugging and unplugging a maze of cables and setting arrays of switches. In effect, the machine had to be rebuilt for each new problem it was to solve. When completed in late 1945, the ENIAC operated much faster than any other machine before it. But while it could solve a complex mathematical problem in seconds, it might take days to set up the machine properly to do that“ (Cruzzi

1 Parameter sind Eigenschaften und Kenngrößen. In der Strömungsdynamik spielen subskalige Parametrisierungen eine wichtige Rolle, denn sie beschreiben all die Eigenschaften, Prozesse und Kenngrößen, die in den Gleichungen nicht enthalten sind und daher durch das Berechnungsraster fallen.

1998: 21).² Um die automatischen Rechenmaschinen der ersten Stunde bedienen zu können, musste die Ausführung von Berechnungen direkt als Befehle auf Maschinensprachebene formuliert oder wie bei ENIAC als Steckverbindungen eingestöpselt werden. Dabei folgten diese maschinentauglichen Befehle dem Prinzip der grundlegenden Rechnerarchitektur, komplexere Funktionen in einfache Grundfunktionen aufzulösen.³ Programmieren zu Beginn der Computerentwicklung bedeutete, per Hand eine Berechnung in einfachste, schrittweise abarbeitbare Operationen zu zerlegen. Typische ENIAC-Befehle waren „ $S(x) \rightarrow Ac$: Clear accumulator and add number located at position x in the Sceletrons into it“ (Goldstine, von Neumann 1947: 85). 1947 stellten von Neumann und Goldstine in dem Report *Planning and Coding Problems for an Electronic Computing Instrument* eine Methode vor, um die Abfolge der maschinentauglichen Befehle eines Berechnungsprozesses mit Hilfe von Flussdiagrammen zu vereinfachen. Für ihre graphische Methode entwickelten sie eine Darstellungsweise, die an eine Choreographie linearer Abläufe, Schleifen und Entscheidungspfade für die Verknüpfung der Operationen erinnert. Dabei definierten sie, was unter Coding zu verstehen ist:

„Coding is, of course, preceded by a mathematical stage of preparations. The mathematical or mathematical-physical process of understanding the problem, of deciding with what assumptions and what idealizations it is to be cast into equations and conditions, is the first step in this stage. The equations and conditions thus obtained are rigorous, with respect to the system of assumptions and idealizations that has been selected. Next, these equations and conditions, which are usually of an analytical and possibly of an implicit nature, must be replaced by arithmetical and explicit procedures. (These are usually step-by-step processes or successive approximation processes, or processes with both of these characteristics – and they are almost always characterized by multiple inductions.) Thus a procedure obtains, which is approximate in that sense in which the preceding one was rigorous. This is the second step in this stage. It should be noted that the first step has nothing to do with computing or with machines: It is equally necessary in any effort in mathematics or applied

-
- 2 Dies änderte sich im Laufe der Entwicklung von schnelleren Rechnern, höheren Programmiersprachen, Ein- und Ausgabegeräten wie Tastatur, Mouse und Bildschirm sowie allmählich anwachsenden Bibliotheken von Algorithmen für standardisierte und häufig verwendete Operationen.
 - 3 Computer waren und sind bis heute auf unterster Maschinenebene lediglich im Stande einfache Operationen auszuführen wie arithmetische und logische Operationen, bit-orientierte Operationen, Speicher-, Vergleichs- und Steueroperationen. Selbst heutige Rechner haben sich diesbezüglich nicht verändert, auch wenn auf der Ebene höherer Programmiersprachen komplexere Berechnungen möglich geworden sind.

mathematics. Furthermore, the second step has, at least, nothing to do with mechanization: It would be equally necessary if the problems were to be computed ‚by hand’. Finally, the precision of the approximation process, introduced by the second step, must be estimated. [...] It may be carried out by the planner, ‚mathematically’, or it may be set up for computation, in which case it may be advantageous to have it, too, carried out by the machine.

Coding begins with the drawing of the flow diagrams. This is the *dynamic or macroscopic stage of coding*. [...] It has been our invariable experience that once the problem has been understood and prepared in the sense of the preceding first remark, the drawing of the flow diagram presents little difficulty“ (Goldstine, von Neumann 1947: 99, 100). „The next stage consists of the individual coding of every operation box, alternative box and variable remote connection. This is the *static or microscopic stage of coding*. Its main virtue is that the boxes in question can now be taken up one by one, and that the work on each one of them is essentially unaffected by the work on the others“ (Goldstine, von Neumann 1947: 101). „The last stage of coding consist of assigning all storage positions and all orders their final numbers. [...] These principles express all the restrictions that need be observed in the final, linear ordering of the boxes“ (Goldstine, von Neumann 1947: 103).

Diese mathematik- und maschinennahe Praktik der Codierung, die in ihrer Allgemeinheit bis heute zutreffend ist, thematisiert zwei neue Prinzipien computerbasierter Mathematik. Zum einen die Überwindung statischer Konzepte durch dynamische Abläufe und zum anderen die Implementierung von alternativen Pfaden, die sich in höheren Programmiersprachen als ‚goto‘-, ‚while‘- oder ‚if-else‘-Befehle wieder finden. Die Codierung zerlegt ein mathematisches Problem gemäß dieser Logik. Sowohl die Dynamik wie auch die Wahl alternativer Abläufe bildet die Basis, um das naturwissenschaftliche Prozessverständnis nicht nur zu beschreiben, sondern adäquat darzustellen und durch schrittweise Berechnungen im Computer auszuführen. Was bis dahin in seiner Dynamik den Experimenten in realiter vorbehalten war, lässt sich nun im rein Semiotischen der Computerexperimente realisieren.

Codierung, wie von Neumann und Goldstine 1947 beschrieben, war bis in die 1950er Jahre Handarbeit mit Bleistift und Papier. Ein Großteil der Codierung bestand jedoch aus Standardroutinen, die sowohl in der Codierung per Hand wie auch in ihrer maschinellen Ausführung zeitaufwendig waren. Insbesondere die Handhabung von Zahlen in der bis dahin üblichen Darstellung der Festkommazahlen kostete viel Zeit, um Fehler bei arithmetischen Überläufen, zu vermeiden. Waren die Ergebnisse einer Berechnung für den darstellbaren Zahlenraum eines Speichers zu groß, dann liefen sie über und verursachten Fehlberechnungen. „This meant that most of a programmer’s effort was scaling his calcula-

tion, not coding“ (Knuth, Prado 1980: 232). Berechnungen mussten für jeden Berechnungsschritt neu skaliert werden. Das heißt, es musste angegeben werden, ob das Ergebnis Einer, Zehner, Hunderter oder Tausender bedeutete, denn die Zahlenspeicher der ersten Computer waren zu klein, um längere Zahlen darzustellen. ENIAC beispielsweise konnte nur 20 Zahlen von je 10 Dezimaleinheiten speichern. „Thus even large inefficiencies in compiling or interpreting looping and testing operations and in computing addresses were masked by the fact that the most operation time was spent in floating-point subroutines. But the advent of the 704 with built-in floating-point and indexing radically altered the situation. [...] First it removed the *raison d'être* of earlier systems by providing in hardware the operations they existed to provide, and second, it increased the problem of generating efficient programs by an order of magnitude by speeding up floating-point operations by a factor of ten“ (Backus 1980: 131).⁴ Mit dem IBM 704 Computer wurde 1957 erstmals die Gleitkommaarithmetik eingeführt.

Ein weiterer Schritt zum wissenschaftlichen Rechnen, wie es heute üblich ist, war die Automatisierung der Codierung selbst durch die Entwicklung von Compilern und Programmiersprachen. Bis dahin mussten die Programmierer die Computer direkt in Maschinensprache instruieren. „At that time [1954], most programmers wrote symbolic machine instructions exclusively [...] they firmly believed that any mechanical coding method would fail to apply the versatile ingenuity which each programmer felt he possessed and constantly needed in his work“ (Backus 1964: 382). Doch Compiler, die die Befehle der Programmiersprache in Maschinensprache übersetzen, erleichtern die Arbeit und ermöglichen es, Programmiersprachen zu entwickeln. Der erste Compiler A-0 geht 1952 auf Grace Hopper, Nora Moser und Kollegen im UNIVAC-Team zurück. „To compile means to compose out of materials from other documents. Therefore, the compiler method of automatic programming consists of assembling and organizing a program from programs or routines or in general from sequences of computer code which have been made up previously“ (Moser zitiert in: Knuth, Pardo

4 Fließkommazahlen approximieren reelle Zahlen und speichern zum Wert einer Zahl die Stelle des Kommas ab. Beispielsweise wird die Angabe der Lichtgeschwindigkeit nicht als 299.792.458 m/s, sondern als $2,99792458 \cdot 10^8$ m/s gespeichert, wobei der Wert der Mantisse für einen Wertebereich [1, 10] normalisiert wird. In Digitalrechnern werden Fließkommazahlen mit 32 oder 64 Bit dargestellt und beschränken daher die Länge der Zahlen. Dies kann zu Rundungsfehlern führen, was vor allem für die Simulation zu Problemen führte.

1980: 234). Allerdings dauerte es eine Weile, bis Compiler und andere Programmierhilfen benutzt wurden.

Die erste, tatsächlich realisierte Programmiersprache war FORTRAN Formula Translation, die zu Beginn der 1950er Jahre von John Backus entwickelt wurde und bis heute für zahlreiche Simulationsmodelle in den Naturwissenschaften verwendet wird (vgl. Backus 1954).⁵ Hinter der Entwicklung von FORTRAN stand laut Backus die Frage, „what could be done now to ease the programmer’s job? Once asked, the answer to this question had to be: Let him use mathematical notations. But behind that answer (in the new 704 environment) there was the really new and hard question: Can a machine translate a sufficiently rich mathematical language into a sufficiently economical machine program to make the whole affair feasible?“ (Backus 1980: 131). Letztere Frage ist nicht trivial und betrifft die Konsistenz eines Programms. Denn es muss sichergestellt sein, dass der Computer tatsächlich die programmierte Berechnung ausführt. Solange der Codierer seine Berechnungen direkt in maschinenaugliche Befehle überträgt, kann er selbst die Konsistenz überprüfen. Wird die Übersetzung jedoch an eine Programmiersprache und einen Compiler delegiert, muss sich der Programmierer auf das Tool verlassen. FORTRAN erlaubte es nun, die bis dahin per Hand codierten, maschinenauglichen Befehle mathematischer Berechnungen bequem in den gewohnten mathematischen Notationen darzustellen wie die Berechnung von Funktionen (COSF (X), SIN (X), EXPE (X), etc.). Gleichwohl andere Programmiersprachen in dieser Zeit entwickelt wurden, beispielsweise MATH-MATIC von Grace Hopper für UNIVAC Computer, setzte sich FORTRAN, das ab 1957 mit den IBM 704 Computern ausgeliefert wurde, in den Natur- und Ingenieurwissenschaften durch.⁶

FORTRAN verrät den Ursprung der Computerentwicklung, der in der Effizienzsteigerung des wissenschaftlichen Rechnens lag. Heute gibt

5 Als erste Programmiersprache gilt mittlerweile Konrad Zuses Plankalkül aus dem Jahr 1945. Allerdings publizierte Zuse aufgrund der Wirren der Nachkriegszeit seine Konzepte erst wesentlich später (vgl. Zuse 1972).

6 Mit dem IBM 704 Computer brach die Ära der Massenfertigung von Großrechnern an und dementsprechend vergrößerte sich die Gruppe der FORTRAN-Anwender laufend. Im April 1958 benutzte bereits die Mehrzahl der Nutzer, der bis dahin 26 installierten 704 Systeme, FORTRAN (vgl. Backus 1958). FORTRAN wurde seither mehrmals verändert, insbesondere als FORTRAN-77 („do“-Schleifen, „if-then-else-endif“-Statement, etc.) und Fortran90 (Modules, etc.). „The latest version, Fortran90, has incorporated many new ideas, including some of those used in object-oriented programming, but scientific programmers generally are aware of only one of them: array syntax“ (Decyk, Norton, Szymanski 1997: 13).

es nicht nur weitere Programmiersprachen wie Simula, Pascal, C oder C++, sondern kommerzielle mathematische Werkzeuge wie Mathematica, MathWorks oder MATLAB sowie einen ständig wachsenden Markt für spezielle Modellierungssoftware vor allem für den technischen Bereich (vgl. Softguide 2009). Zahlreiche standardisierte Tools wie PDE Partielle Differentialgleichungen Löser erleichtern die Programmierung von Simulationsmodellen. Die Tendenz geht mittlerweile dahin, komplette Computerexperimente als vorgefertigte Softwareprodukte anzubieten. Diese Entwicklung zeigt, dass mathematische Verfahren zu einem weit verbreiteten Handwerkszeug geworden sind. Es zeigt aber auch, dass die Softwareentwicklung die zugrunde liegenden mathematischen Prozesse zunehmend als Blackboxes maskiert. Damit werden mathematische Wissensbestände in Algorithmen festgeschrieben und popularisiert. Was sich bereits 1936 als Entwicklung abzuzeichnen begann – „there is a great deal more arithmetic and better arithmetic in the world than there used to be“ (Bush 1936: 652) – gilt heute in einem nie da gewesenen Umfang.

Computational Departments

Der Zuwachs an Arithmetik durch die Entwicklung automatischer Rechenmaschinen zeigt sich an den Universitäten und Forschungsinstituten in der Gründung von Computational Departments. Die Vorläufer dieser Departments waren die Human Computing Laboratories.

„The Winter of 1943 marked the start of the imperial age of the human computers, the era of great growth for scientific computing laboratories. It seemed as if all the combatants discovered a need for organized computing this winter. A German group started preparing mathematical tables at the Technische Hochschule in Darmstadt. Japan [...] formed a computing group in Tokyo. The British government operated computing groups in Bath, Wynton, Cambridge, and London. Within the United States, there were at least twenty computing organizations at work that winter, including laboratories in Washington, Hampton Roads, Aberdeen, Philadelphia, Providence, Princeton, Pasadena, Ames, Lynn, Los Alamos, Dahlgren, Chicago, Oak Ridge, and New York. Most of these calculating staffs were small, consisting of five to ten computers. [...] Only a few computing laboratories were as large as the New York Hydrographic Project with its forty-nine veterans of the Mathematical Table Project or the thirty-person computing office of the Naval Weapons Laboratory at Dahlgren, Virginia“ (Grier 2005: 257).

Daher ist es kein Zufall, dass die ersten automatischen Rechenmaschinen vom Militär in Auftrag gegeben wurden, um die Berechnungen zu beschleunigen und dadurch Mitarbeiter einzusparen.⁷ So entstanden ENIAC und später EDVAC zwar an der Moore School of Electrical Engineering der Universität von Pennsylvania, doch Auftraggeber war das US-Militär. ENIAC wurde 1947 und EDVAC 1949 an das Ballistic Research Laboratory der US Army in Aberdeen Maryland ausgeliefert. Beide Computer entstanden unter Mitwirkung von John von Neumann, der sowohl Mitglied im wissenschaftlichen Beirat des Ballistic Research Laboratory als auch im Navy Bureau of Ordnance war. Das Ballistic Research Laboratory in Aberdeen mit ENIAC 1947 und EDVAC 1949 als auch das Naval Surface Weapons Center in Dahlgren mit den IBM Computern Mark II 1948, Mark III 1951 und NORC 1954 können wohl als die ersten Computational Departments bezeichnet werden. Umso erstaunlicher ist es, dass sich John von Neumann 1945 – auf der Suche nach einem geeigneten mathematischen Problem zum Test seines geplanten Princeton IAS Computers – für ein rein wissenschaftliches Projekt entschied.⁸ „[Carl-Gustav] Rossby, Vladimir Zworykin of RCA, and Weather Bureau Chief Francis Reichelderfer, had succeeded in convincing von Neumann that weather prediction was a good candidate for his computer“ (Phillips 2000: 15). Die Wettervorhersage, bis weit in die

7 Welcher Computer nun als der erste Computer gelten kann, ist nicht einfach zu beantworten. Auch standen nicht alle Computer im Dienste militärischer Zwecke. Beispielsweise wurde der elektrisch angetriebene mechanische Rechner Z1 von Konrad Zuse 1938 im elterlichen Wohnzimmer gebaut, jedoch während des Krieges zerstört. 1941 baute Zuse ebenfalls in seiner Privatwohnung die Z3, den ersten vollautomatisch arbeitenden Computer mit binärer Gleitkommarechnung, Speicher und Zentralrechen-einheit. Weitere Computer waren der nicht frei programmierbare ABC Atanasoff-Berry-Computer, den John Atanasoff und Clifford Berry in den Jahren 1937 bis 1941 am Iowa State College konstruierten, Mark I, der 1944 als Dezimalrechner von IBM für die Harvard Universität gebaut, sowie Colossus, der 1943 in Bletchley Park unter Mitwirkung von Alan Turing erstellt wurde. Colossus war wie ENIAC nur durch Neuverkabelung programmierbar und diente der Entzifferung der deutschen Lorenz-Schlüsselmaschine. Die Kriterien für einen modernen Computer sind: Binäre Zahlendarstellung, elektronische Realisierung, freie Programmierbarkeit, Turingmächtigkeit und ggf. Gleitkommaarithmetik.

8 John von Neumann war seit 1933 Mitglied des IAS Institute for Advanced Study in Princeton und gründete dort eine Gruppe für die Konzeption und den Bau des IAS Computers, der im Frühjahr 1951 fertig gestellt wurde. „During the summer of 1951, a team of scientists from Los Alamos came and put a large thermonuclear calculation on the IAS machine; it ran for 24 hours without interruption for a period of about 60 days“ (Bigelow 1980: 308).

1960er Jahre eine Erfahrungswissenschaft und Kunst, gehörte zu jenen Disziplinen, die am Schisma zwischen Theorie und Anwendung laborierten. 1904 hatte Vilhelm Bjerknes zwar die theoretischen Grundlagen auf Basis der Navier-Stokes-Gleichungen der Strömungsdynamik gelegt (vgl. Bjerknes 1904), doch die Gleichungen waren zu komplex, um sie per Hand zu berechnen, geschweige denn, um sie auf ein einfaches Erdmodell anzuwenden.⁹ Carl Gustav Rossby, ein Meteorologe der Universität Chicago, erhoffte sich von einem Simulationsmodell einen entscheidenden Durchbruch für die theoretische Fundierung der Meteorologie. „To lay the foundation for a computational approach to theoretical meteorology, it would be necessary to organize a small and versatile group of competent theoretical meteorologists to work with Professor von Neumann“ (Rossby 1946, in: Harper 2004: 85). Es dauerte jedoch noch zwei Jahre bis sich 1948 mit Jule Charney eine erste Gruppe wissenschaftlicher Modellierer zusammenfand, die am IAS Institute for Advanced Study in Princeton an einem computertauglichen Wettervorhersagemodell arbeiteten. Dabei handelte es sich um ein sehr einfaches Modell basierend auf den Navier-Stokes-Gleichungen. Die sieben Variablen zur Bestimmung der Atmosphäre sollten für ein sehr grobes Berechnungsgitter einer gleichmäßig rotierenden Erde gelöst werden. Erste Modellrechnungen waren für 1950 geplant. Da der IAS Computer jedoch noch nicht fertig gestellt war, wurden die Berechnungen im April 1950 auf ENIAC ausgeführt. Eine 24-Stunden-Vorhersage benötigte 24 Stunden reine Rechenzeit auf ENIAC. Da ENIAC zu dieser Zeit bereits vom Ballistic Research Laboratory für ballistische Berechnungen genutzt wurde, konnten nicht mehr als vier 24-Stunden-Vorhersagen und zwei 12-Stunden-Vorhersagen durchgeführt werden (vgl. Phillips 2000).

Die Forschergruppe des Meteorological Projects des Institute for Advanced Study in Princeton um Charney, von Neumann und, in wechselnder Besetzung weiterer Wissenschaftler unterschiedlicher Disziplinen, wie Norman Phillips, Joseph und Margaret Smagorinsky, bestand zwischen 1948 und 1955 als eine der ersten Modellierungs- und Simulationsgruppen der sich formierenden Computational Sciences. Sie nutzten neben ENIAC den IAS Computer und NORC. Auch wenn das Office of Naval Research das Projekt finanzierte, kann das meteorologische Computerexperiment von 1950 als eines der ersten, wenn nicht sogar als das erste Computerexperiment der akademischen Naturwissenschaften gelten. Auf alle Fälle gilt es als Beginn der numerischen Wettervorhersage und wurde im Jahr 2000 zum 50-jährigen Jubiläum mit einem interna-

9 Lewis F. Richardson hatte 1922 versucht, eine Berechnung per Hand durchzuführen, doch er scheiterte (vgl. Richardson 1922, Lynch 1999).

tionalen Symposium gefeiert (vgl. Spekat 2000). Dieses erste Computerexperiment dokumentiert die enge Zusammenarbeit zwischen Computerdesignern und Modellierern, die bis heute besteht. 1954 zog John von Neumann auf einem Vortrag in Deutschland die Bilanz seiner Erfahrungen mit diesem ersten Computerexperiment: „Übrigens waren diese Bemühungen recht erfolgreich, so daß der Wetterdienst der Vereinigten Staaten zusammen mit der Marine und der Luftwaffe auf Grund unserer Methoden einen dauernden rechnerischen Wettervorhersagedienst einrichtet. Dieser wird wohl im Laufe des nächsten Jahres mit einer großen, vollautomatischen Schnellrechenmaschine (vom Typ ‚701‘ der IBM International Business Machine Corporation) regelmäßig funktionieren“ (von Neumann 1954: 264).¹⁰ Ein Jahr später gründete Joseph Smagorinsky auf von Neumanns Betreiben die General Circulation Research Section im U.S. Weather Bureau, die 1986 als GFDL Geophysical Fluid Dynamics Laboratory in Princeton angesiedelt wurde.

Diese Entwicklung, ausgehend von einer kleinen Gruppe von Wissenschaftlern, ist charakteristisch für die Entstehung der Computational Sciences und den sich institutionell etablierenden Computational Departments. In nahezu allen naturwissenschaftlichen Disziplinen sind in den letzten fünfzig Jahren – neben den klassischen Instituten und Departments für experimentelle Forschung und Theorie – Computational Departments entstanden. Diese neuen Departments dokumentieren die Revolution der wissenschaftlichen Methoden. Neben Theorie, Experiment, Beobachtung und Messung kommt in den 1950er Jahren, verstärkt ab den 1970er Jahren, nun die Simulation als neues Erkenntnisinstrument hinzu. Die neu eingerichteten Computational Departments folgen einer eigenen Forschungslogik, die durch die numerische Analyse komplexer Systeme, anwendungs- wie problemorientierte Forschungsfragen, eine hochgradig interdisziplinäre Zusammenarbeit und internationale Vernetzung sowie die Abhängigkeit des Erkenntnisfortschritts von der Leistungsfähigkeit der Rechner gekennzeichnet ist. Und sie verfügen über eine eigene Infrastruktur (e-Science). Exemplarisch zeigt sich die Konzeption eines solchen Departments am Center for Computational Geophysics der Rice Universität: „The Center for Computational Geo-

10 „By 1954, both modeling capability and computer power had advanced to a point where the possibility of real-time operational numerical weather prediction was under active consideration in Europe and the United States. On July 1, 1954, the Joint Numerical Weather Prediction Unit (JNWPU) was organized, staffed, and funded by the U.S. Weather Bureau, the U.S. Air Force, and the U.S. Navy” (NOAA 2008: Foundations, The History of Numerical Weather Prediction).

physics was established in 1998 to promote cross-disciplinary studies of the Earth requiring large computational resources. Comprised of 11 faculty members and research associates from the Earth Science, Computational & Applied Mathematics, and Electrical & Computer Engineering Departments, the Center has research areas in petroleum, environmental, and academic seismology, lithosphere and whole mantle geodynamics, and fault mechanics“ (Rice University 2008: Earth Science).¹¹

„Computational Science,“ schreibt Friedel Hoßfeld, einer der Pioniere des wissenschaftlichen Rechnens in Deutschland, „ist synonym mit der Untersuchung komplexer Systeme; ihr Instrument ist der Supercomputer, ihre Methode die Simulation. [...] sie zielt auf die großen ungelösten, wissenschaftlichen Probleme, die in ihrer Wichtigkeit und Tiefe nicht nur die betreffende wissenschaftliche Disziplin herausfordern, sondern von außerordentlicher Bedeutung und Auswirkung für die Gesellschaft und ihrer Bewältigung der Zukunft sind. Die Identifizierung solcher Probleme hat in den USA zur Klassifizierung als ‚Grand Challenges to Computational Science‘ geführt, zu denen heute die Kernfragen aus den Gebieten der atmosphärischen Chemie (d.h. der Umweltforschung), der Astrophysik, der Materialforschung, der Molekularbiologie, der Elementarteilchenphysik und der Aerodynamik gezählt werden“ (Hoßfeld 1991: 1).

e-Science

Die Nutzung von Computern als Forschungsinstrumente, die weltweite Vernetzung der elektronischen Infrastrukturen und die flächendeckende Vermessung der Welt durch computerchipgesteuerte Detektoren führte Anfang des 21. Jahrhunderts zum Begriff der e-Science. „e-Science is about enhancing science through global collaboration using computing and communications technologies. e-Science integrates grid computing, programming tools, and data and visualization technologies enabling scientists to gain greater insight into their research through direct comparisons of simulations, experiments and observations. Knowledge repositories maintaining relationships, concepts and inference rules will be accessible through e-Science to facilitate these new insights“ (Berg et al. 2003: 5). Ziel der e-Science ist es, alle größeren Rechenzentren, Experimentier- und Messgeräte miteinander zu vernetzen und über Institutio-

11 Da es noch keinerlei infrastrukturelle Forschungen zu diesem Umbau der Naturwissenschaften gibt, fehlt ein deutschlandweiter und internationaler Überblick der Computational Departments. Einen indirekten Einblick in die Orte der Computational Sciences gibt die *TOP500 Liste* der weltweit schnellsten Rechner (vgl. TOP500 2009).

nen- und Ländergrenzen hinweg als virtuelles Labor verfügbar zu machen (Grid-Computing). Zahlreiche Länder konzipieren mittlerweile umfangreiche Förderprogramme für solche virtuellen Labore. Dabei stellen die Supercomputer in den weltweiten Rechenzentren sowie die Datenleitungen zwischen diesen Zentren die Kerninfrastruktur des verteilten Rechnens und Forschens dar.

Die Voraussetzung dafür sind Entwicklungen aus den 1970er Jahren. 1976 kam mit dem Cray-1 Vektorrechner der erste Supercomputer mit einer Geschwindigkeit von 80 Millionen Gleitkommazahl-Operationen pro Sekunde auf den Markt.¹² Im selben Jahr wurde an der Universität Illinois der erste Parallelrechner mit 256 Prozessoren gebaut (ILLIAC 4). Bereits 1975 waren weltweit 57 Großrechner im ARPANET Advanced Research Projects Agency Network miteinander vernetzt, die meisten davon in den USA. Das ARPANET war 1969 mit der Vernetzung von vier Forschungseinrichtungen gestartet: Stanford Research Institute (SDS 940), Universität von Utah (DEC PDP-10), Universität von Kalifornien in Los Angeles (SDS Sigma 7) und in Santa Barbara (IBM 360/75). Doch erst die Entwicklung des Mehrbenutzer-Betriebssystems UNIX Anfang der 1970er Jahre in den Bell Laboratories – bis heute das gängige Betriebssystem der Großrechner – und die ebenfalls zu Beginn der 1970er Jahre in den Bell Laboratories entwickelte Programmiersprache C verhalfen der Vernetzung von Computern zum Durchbruch. Dieser Durchbruch führte 1982 zum Internet und 1989 zum World Wide Web, wobei es das erklärte Ziel des World Wide Web war, Forschungsergebnisse leichter austauschen zu können (vgl. Berners-Lee 1989).

12 Von den insgesamt achtzig gebauten Cray-1 wurde der erste dieser Rechner 1976 an das Los Alamos National Laboratory, der zweite ein Jahr später an das 1960 gegründete NCAR National Center for Atmospheric Research ausgeliefert und der dritte Computer an das 1975 gegründete ECMWF European Centre for Medium Range Weather Forecasts ausgeliefert. In den 1960er und frühen 1970er Jahren gab es neben IBM sechs kleinere Hersteller von Großrechnern in den USA: Burroughs, UNIVAC, NCR National Cash Register, CDC Control Data Corporation, Honeywell, RCA Radio Corporation of America und General Electric. CDC baute mit dem CDC 6600 Rechner (3 MegaFLOPS Rechenleistung) bereits 1964 einen Großrechner, der mit dem Begriff Supercomputer bezeichnet wurde. CDC Chefentwickler Seymour Cray verließ 1972 das Unternehmen und gründete die Cray Research, ein weiteres Großrechner-Unternehmen. 1997 wurde mit dem Intel ASCI Red/9152 (1,338 TeraFLOPS Rechenleistung) für die Sandia National Laboratories in New Mexiko die TeraFLOP-Grenze überschritten. Heute stößt man in den Bereich der PetaFLOPS vor. In Deutschland stellte Konrad Zuse mit seinem Unternehmen die ersten kommerziell vertriebenen Großrechner her.

In den 1970er Jahren wurden auch die Ein- und Ausgabegeräte zunehmend benutzerfreundlich. 1973 wurde mit dem Xerox Alto der erste Computer mit Maus, graphischer Benutzeroberfläche und Ethernetkarte ausgeliefert. Die Monitore wurden verbessert und Magnetbänder als auch Festplatten lösten zunehmend die Lochkarten und Lochstreifen als Eingabegeräte ab.¹³ Die zunehmende Leistungsfähigkeit der Computer sowie der Kernspeicher – Ende der 1970er Jahre verfügten die Großrechner über rund 1 MegaByte Arbeitsspeicher – erlaubten die ersten Visualisierungen. Schließlich begann mit dem Intel 4004 Chip, der 2.300 Transistoren zu einer CPU Central Processing Unit zusammenfasste, in den 1970er Jahren das Zeitalter der Mikroprozessoren (vgl. Flik 2004). Zum Vergleich: ENIAC war ein zimmergroßer Prozessor bestehend aus rund 18.000 Elektronenröhren.

Mit der Installation des Cray X-MP Vektorrechners und der Gründung des HLRZ Höchstleistungsrechenzentrums an der Universität Stuttgart begann in Deutschland 1983 die Ära des Höchstleistungsrechnens. Die Supercomputer sind die Stars dieser Ära.¹⁴ „Was die Teilchenbeschleuniger für die Experimentalphysik sind, leisten die Höchstleistungsrechner für die Computational Sciences: Die Supercomputer sind die Beschleuniger der Theorie! Der Transfer des Wissens und Könnens auf dem Gebiet des Supercomputing, insbesondere des Parallelrechnens, wird jedoch für die zukünftige Konkurrenzfähigkeit der Industrie im nationalen Wettbewerb ein wichtiger Wirtschaftsfaktor werden [... und] man ist wohl auch genötigt, eine enge Korrelation zwischen Verfügbarkeit des Supercomputing für die Industrie und deren Leistungsfähigkeit am Weltmarkt zu vermuten“ (Hoßfeld 1992: 3; Jarp

13 1952 stellte IBM mit dem IBM 726 Bandgerät das erste Datenspeichermedium auf Basis von Magnetbändern her. Das Gerät konnte 1,4 Megabyte aufzeichnen. 1956 entwickelte IBM mit RAMAC 305 die erste Festplatte, ein tausend Kilo schweres Ungetüm von 60 Platten für 5 Megabyte Daten (vgl. Magnetbandmuseum 2009).

14 Während in den USA etliche Hersteller von Supercomputern ansässig waren, hatte in Deutschland die Zuse KG, die 1949 mit der Z4 den ersten kommerziell vertriebenen Großrechner in Europa an der ETH Eidgenössische Technische Hochschule Zürich installiert hatte, den Betrieb eingestellt bzw. war 1969 in den Besitz der Siemens AG übergegangen (vgl. Zuse 2008). Siemens und Telefunken bauten in den 1970er und 1980er Jahren Großrechner wie den Siemens VP200-EX, den Siemens S100/10 oder den Telefunken TR440. 1985 nahm die GMD Gesellschaft für Mathematik und Datenverarbeitung mit dem Bau des SUPRENUM Rechners, einem Parallelrechner mit 256 Prozessoren, ein innovatives Projekt in Angriff. 1990 wurde der Rechner in St. Augustin in Betrieb genommen, aber obwohl die Architektur richtungsweisend war, scheiterte das Projekt kommerziell.

1991). Diese Einsicht in die Bedeutung des Höchstleistungsrechnens für Industrie, aber auch Wissenschaft und Forschung setzte sich in Deutschland jedoch nur langsam durch. Die Situation der Klimaforschung beispielsweise verbesserte sich erst 1985, als das Max-Planck-Institut für Meteorologie in Hamburg mit einem CYBER 25 Supercomputer ausgestattet wurde und 1988 die Gründung des DKRZ Deutsches Klimarechenzentrum in Hamburg mit einem CRAY-2 System erfolgte. Während in Deutschland im Jahr 1992 43 Großrechner installiert waren, davon 26 in wissenschaftlichen Institutionen,¹⁵ und insgesamt 31 Mio. Mark für paralleles Höchstleistungsrechnen ausgegeben wurden, waren im selben Jahr in den USA 240 der weltweit 616 Supercomputer installiert und es wurden 657 Mio. US Dollar für HPCC High Performance Computing and Communication ausgegeben (vgl. BMFT 1993).¹⁶

„Relativ spät, aber vielleicht noch rechtzeitig, wurde der Stellenwert der Computational Sciences und des Supercomputing wie die Notwendigkeit für entsprechendes Handeln auch in Europa erkannt: Der Rubbia-Report liefert für die EG eine eingehende Analyse der europäischen Situation auf dem Gebiet des Supercomputing und seines industriellen Umfeldes und eine Reihe strategischer Empfehlungen; mehrere Arbeitsgruppen haben Aktionslinien identifiziert und Rahmenempfehlungen für die Umsetzung des High Performance Computing and Networking (HPCN) Program in einem 7-Jahres-Plan ausgearbeitet, das im Mittel ein jährliches Fördervolumen von 900 Mio. Ecu umfassen soll. Es ist zu hoffen, daß das Programm bald und im empfohlenen Umfang von der EG verwirklicht wird“ (Hoßfeld 1992: 3; vgl. Rubbia 1991).¹⁷

15 Unter diesen 26 Großrechnern des Jahres 1992 in fünf Forschungszentren (MPI Max Planck Institut Garching, DLR Deutsches Luft- und Raumfahrtzentrum Oberpfaffenhofen, DKRZ Deutsches Klimarechenzentrum Hamburg, Forschungszentren Jülich und Karlsruhe), dem DWD Deutscher Wetterdienst Offenbach sowie sechzehn Universitäten (Berlin, Stuttgart, Karlsruhe, Kiel, Kaiserslautern, Köln, München, Hannover, Aachen, Dresden, Bremen, Gießen, Kassel, Frankfurt, Hamburg, Darmstadt) waren zwölf Cray, zwölf Siemens und ein NEC Computer (BMFT 1993: 12).

16 Zum Vergleich: 1992 standen 240 Supercomputer in den USA/Kanada und 138 in Europa, darunter 43 in Deutschland, 35 in Frankreich und 24 in Großbritannien (vgl. Meuer 1992).

17 „Besonders traurig in diesem Zusammenhang ist es, wenn staatliche Förderungsinitiativen außer Sitzungen nichts produzieren: Während die US-Amerikaner den Ernst der Stunde anscheinend erkannt und konzentrierte Aktivitäten gestartet haben, erweckt die europäische Rubbia-Initiative den Eindruck eines Debattierclubs, der wegen diverser Partikularinteressen Gefahr läuft, Zeit und den Anschluß an das ‚Weltniveau‘ im Supercomputing zu verlieren“ (Jehnsen 1992). Die Situation Anfang der 1990er Jahre hatte sich bezüglich des Baus von Supercomputern verändert. Während in Europa keine Großrechner mehr gebaut wurden, drohten die US-amerikanischen Hersteller gegenüber den japanischen Unternehmen wie

Doch es geht in den 1990er Jahren nicht nur um Großrechner. „Die Installation von Höchstleistungsrechnern setzt notwendigerweise voraus, daß einem breiten Kreis potentieller Nutzer auch von außen Zugriff auf diese Rechner ermöglicht wird. Erforderlich ist dazu die Einbettung der Rechner in ein leistungsfähiges Netz, das insbesondere für den schnellen Austausch von Massendaten [...] geeignet ist. Für den Anwender muß der Zugriff auf den Rechner so erfolgen können, als ob dieser im eigenen lokalen Netz zur Verfügung stünde“ (BMFT 1993: 25). Es ist dieser direkte Zugriff und die damit einhergehende infrastrukturelle Umgestaltung der Wissenschaft, welche die e-Science charakterisiert. Heute ist im Zuge dieser Umgestaltung von Parallelrechnern mit 212.992 Prozessoren (IBM BlueGene/L) die Rede, von Datennetzen mit Transferleistungen von 10 Gigabit/sec, von Programmen, die via Internet den weltweiten Zugriff auf Supercomputer, Experimentier- und Messgeräte erlauben, sowie von Rechenzentren, die zunehmend als Dienstleister für Natur- und Ingenieurwissenschaften fungieren.¹⁸ Im Zuge dieser neuen Dienstleistungen der e-Sciences schließen sich immer mehr Rechenzentren zusammen. So wurde 2007 Europas größtes Rechenzentrum, das Gauss Centre for Supercomputing, durch den Zusammenschluss dreier Rechenzentren gegründet (NIC John von Neumann-Institut für Computing Jülich, LRZ Leibniz-Rechenzentrum der Bayerischen Akademie der Wissenschaften in Garching und HLRZ Höchstleistungsrechenzentrum der Universität Stuttgart). Das Gauss Centre wiederum ist mit fünfzehn weiteren Supercomputerzentren Teil von PRACE Partnership for Advanced Computing in Europe. In dem EU Memorandum vom April 2007 heißt es dazu: „Computation has become essential for the advancement of all research across science. But the high-performance computing resources are limited and a European effort is needed to treat the provisioning in a strategic way. Together the 16 members of PRACE, the Partnership for Advanced Computing in Europe, will strengthen European science, engineering and supercomputer technologies and thus se-

Fujitsu, Hitachi und NEC zunehmend ins Abseits zu geraten. Der in Anlehnung an den Sputnik Schock der 1950er Jahre als ‚computnic race‘ betitelte Wettlauf zwischen US-amerikanischen und japanischen Unternehmen wurde Ende der 2004 mit dem IBM BlueGene/L entschieden.

18 Die Bestandsaufnahme der aktuellen Situation in Deutschland zeigt, dass hier zwar keine Supercomputer mehr gebaut werden, dass aber mit dem im Forschungszentrum Jülich Anfang 2008 installierten und 2009 erweiterten JUGENE Blue Gene/P (294.912 Prozessoren), der weltweit drittschnellste Computer steht. Die Liste der TOP500 Rechner zeigte im Juni 2009, dass nur sechs Prozent der Rechenkraft der fünfhundert leistungsfähigsten Computer in Deutschland zur Verfügung stehen (vgl. TOP500 2009).

cure Europe a pioneering role in the global competition“ (PRACE 2007: Memorandum).

Um diese Ressourcen zu vernetzen, bedarf es leistungsfähiger Datennetze und Softwareprogramme, die das verteilte Höchstleistungsrechnen ermöglichen. In Deutschland vernetzt seit 1990 das WIN Wissenschaftsnetz des Deutsches Forschungsnetz e.V. die Forschungsinstitutionen und Universitäten. Heute bietet diese Infrastruktur eine Transferleistung im Gigabit/s-Bereich und wird zunehmend ausgebaut,¹⁹ denn „nachdem in den 90er Jahren das World Wide Web das Internet revolutionierte, sehen wir uns nun einer weiteren revolutionären Entwicklung gegenüber: dem Grid-Computing. Während das Web in der Regel nur statische Informationen bereitstellt, ermöglicht das Grid nun den direkten Zugriff auf die Ressourcen selbst, wie Rechner, Speicher, wissenschaftliche Instrumente und Experimente, Anwendungen und Daten, Sensoren und sogenannte Grid-Middleware Dienste. Letztere basieren auf weit verbreiteten Grid- und Web-Services-Standards, die die effiziente Kommunikation der Ressourcen untereinander ermöglichen“ (Neuroth, Kerzel, Gentzsch 2007: 9, vgl. D-Grid 2009). Dazu wird ein Interface benötigt, das sowohl die verschiedenen Computer und Geräte von einer gemeinsamen Oberfläche aus zugänglich macht, als auch das Datenmanagement der Massendaten ermöglicht. Mit dem Projekt UNICORE Uniform Interface to Computing Resources wurde ab 1997 ein solches Interface auf Basis der Erfahrungen des Höchstleistungsrechnens in den Großrechenzentren programmiert (vgl. UNICORE 2008). Das Deutsche Forschungsnetz wie auch das D-Grid, UNICORE oder das Gauss Centre (vgl. GCS 2009) sind wiederum Teil europäischer Initiativen wie dem europäischen Wissenschaftsnetz GEANT basierend auf 26 nationalen Forschungsnetzen (vgl. GEANT 2009), dem e-Infrastructure Programm der EU Europäischen Union (vgl. e-Infrastructure 2009), dem I2G Interactive European Grid Project (vgl. I2G 2009) oder der DEISA Distributed European Infrastructure for Supercomputing Applications (vgl. DEISA 2009). Schließlich bildet sich mit der Gründung des IAS Institute for Advanced Simulation 2008 am Forschungszentrum Jülich ein neuer Typ wissenschaftlicher Institute

19 Das WIN Wissenschaftsnetz wird vom DFN Deutsches Forschungsnetz e.V. verwaltet, der 1984 gegründet wurde. 1994 wurde ein Testbed mit einer Transferleistung von 34 Mbit in Nordrhein-Westfalen, Bayern, Berlin, Norddeutschland und Hessen sowie ein regionales Hochgeschwindigkeitsnetz von 155 Mbit/s in Baden-Württemberg installiert. 1996 wurde es bundesweit zum B-WIN Breitband-Wissenschaftsnetz (155 Mbit/s) und seit 2000 zum G-WIN Gigabit-Wissenschaftsnetz (2,5 – 10 Gbit/s) ausgebaut (vgl. Wissenschaftsrat 2002, DFN 2008).

heraus: ausschließlich jene Wissenschaftsbereiche werden unterstützt, deren Erkenntnisfortschritt ausschließlich von Simulationen abhängt. Typische Anwendungsfelder sind quantentheoretische Untersuchungen, Astrophysik, Materialforschung, Biophysik sowie die Theorie der Strukturbildung (vgl. IAS 2009).

Die Umgestaltung der Naturwissenschaften zur e-Science beinhaltet vielfältige Aspekte wie die Entwicklung neuer Infrastrukturen, die Um- und Neuverteilung von Ressourcen, die Entstehung neuer Grid-Communities bis hin zu veränderten Anforderungen an die naturwissenschaftliche Ausbildung. Es entstehen neue Studiengänge wie Technomathematik, Scientific Computing oder Wissenschaftliches Rechnen sowie Graduiertenschulen wie die Heidelberg Graduate School of Mathematical and Computational Methods for the Sciences (vgl. MATHCOMP 2009). Die beiden maßgeblichsten Einflüsse zeigen sich jedoch in der Transformation der Epistemik und Praktik wissenschaftlichen Forschens. Computereperimente werden aufgrund der Rechnerkapazitäten nicht nur zunehmend realitätsnaher. Ihre Einbindung in den bisherigen Kanon der Wissensproduktion von Beobachtung, Experiment und Messung – alle drei Erkenntnisinstrumente haben sich mittlerweile zu algorithmierten und hochgradig datenproduzierenden Forschungsmethoden entwickelt – wird durch automatisierte Datenvergleiche, Datenauswertungen und Daten-Reinterpretationen professionalisiert und standardisiert. Die Infrastrukturen der e-Science verändern die Forschungspraktiken grundlegend, nicht nur im digitalen Labor des Computers, sondern in den zu virtuellen Laboratorien vernetzten Supercomputern, Experimentier- und Messgeräten. Die Folge all dieser Veränderungen ist: „Software is the new physical infrastructure of [...] scientific and technical research“ (PITAC 1999: 27). In anderen Worten: Das komplette Wissen der Naturwissenschaften wird sukzessive in Algorithmen transformiert.