Über die Qualität der Software von Klimamodellen

Über die Qualität der Software von Klimamodellen: eine Analyse der Fehlergrenzen von drei Modellen

J. Pipitone und S. Easterbrook

Abstract: *Ein Klimamodell ist eine Beschreibung der Theorie des Klimas; das Modell verwandelt klimatologische Theorien in Software, so dass die Theorien simuliert und ihre Implikationen untersucht werden können. Folglich muss man darauf vertrauen können, dass die Software, die das Klima beschreibt, das Klima korrekt abbildet. Unsere Studie erforscht die Natur der Software-Qualität im Zusammenhang mit der Modellierung des Klimas. Wir führten eine Analyse der Fehleranzeigen und fehlerhafter Festlegungen in vielen Versionen führender Klimamodelle durch, indem wir fehlerhafte Daten von Fehlersuchdateien und aus Fehlerprotokollen sammelten. Wir fanden heraus, dass die Klimamodelle alle sehr geringe Fehlergrenzen haben, jedenfalls im Vergleich mit gut bekannten Projekten ähnlicher Größenordnung mit offenen Quellen. Wir diskutieren unsere Ergebnisse, um Aussagen über die Vertrauensbasis der Modellsoftware zu machen.*

Aus: Assessing climate model software quality: a defect density analysis of three models, by J. Pipitone and S. Easterbrook, *Geosci. Model Dev. Discuss.*, 5, 347-382, 2012, www.geosci-model-dev-discuss.net/5/347/2012/, doi:10.5194/gmdd-5-347-2012

Kommentar von Judith Curry: Bei diesem Journal handelt es sich um ein Online-Diskussionsportal (von welchem ich ein großer Fan bin). Bis heute wurden zwei interaktive Kommentare gepostet (hier und hier). Beide diese Begutachtungen sind positiv. Die folgende Begutachtung habe ich per E-Mail erhalten, und zwar von jemandem, der anonym bleiben möchte (diese Begutachtung ist nicht positiv). Diese Person hat die Absicht, eine Begutachtung online an das Discussion Journal zu senden und möchte die Kommentare hier sorgfältig lesen, um die Begutachtung vor dem Absenden hieb- und stichfest zu machen. Einen Blogbeitrag von Easterbrook zu dieser Studie findet man hier, zusammen mit einem guten Kommentar von Nick Barnes.

Begutachtung der Studie von Pipitone und Easterbrook von Anonymus

Einführung

Folgende Punkte in der Studie von Pipitone und Easterbrook werden in diesem Kommentar angesprochen:

1. Fehlende Eindeutigkeit bei der Entwicklung der Software der Globalen Klimamodelle (GCMs)
2. Auswirkungen des Advanced Strategic Computing Initiative (ASCI)-Projektes auf die Entwicklung moderner Verifikations- und

Bewertungsverfahren
3. Fehlende Präzision einfacher Fehleraufzeichnungen als Indikator der Software-Qualität.
4. Fehlende Überlegungen zur Eignung von Produktionsbeiträgen zur Software im Bereich der Unterstützung von Entscheidungen.

Wissenschaftliche Software-Entwicklung

Hinsichtlich der erforderlichen großen Erfahrung, der Komplexität der Phänomene, einer großen Anzahl von Systemfunktionen, die hier von Interesse sind und in jeder anderen Hinsicht sind die Globalen Klima-(Zirkulations-)Modelle alles andere als eindeutig.

Jede wissenschaftliche und Ingenieurs-Software in Zusammenhang mit Anwendungen in der realen Welt erfordert hohe Aufmerksamkeit durch Experten mit umfangreichen Kenntnissen der physikalischen Aspekte, und diese Experten müssen entscheidend im Prozess der Software-Entwicklung mitwirken. Das Klima der Erde und andere natürliche Phänomene und Prozesse sind hier keine Ausnahme. Einige Systeme und damit verbundene physikalische Prozesse sind mindestens ebenso inhärent komplex wie die Klimasysteme der Erde. Modelle und Software für derartige Systeme erfordern auch intensive Kenntnisse über eine Vielfalt verschiedener Phänomene und Prozesse, die für die Modellierung von Bedeutung sind. Sie sind auch für die Beschreibung der Prozesse und die Kopplung zwischen Komponenten zur Beschreibung dieser Prozesse wichtig. Aus den gleichen Gründen müssen die Anwender der Software auch über großes Wissen und Erfahrung über das physikalische Problem verfügen, das heißt auf dem Gebiet der Anwendung.

Wie in der Studie von Pipitone und Easterbrook erwähnt, nutzen die Entwickler der Modelle, Methoden und Software diese komplexen Systeme, um sowohl über die zugrunde liegende Physik als auch über die Software selbst Erfahrungen zu sammeln. Intensive iterative Wechselwirkungen zwischen Experten aus Physik, Mathematik und Softwareentwicklung sind die Regel. Die GCMs sind in dieser Hinsicht keine Ausnahme.

Das Problem der Softwarequality in wissenschaftlicher Software

Easterbrook und Johns (2009) haben eine Übersicht über einige der Techniken präsentiert, die in einigen Laboren der GCM-Modellierung bei der Entwicklung der GCM-Software angewendet worden sind. Alle von den Autoren beschriebenen Techniken sind standardmäßige Operationsprozesse, die während der Ausarbeitung von Forschungsversionen aller wissenschaftlichen und technischen Software vor Veröffentlichung der Codes benutzt werden.

Die von Easterbrook und Johns beschriebenen Aktivitäten werden manchmal als entwicklungsbedingte Einschätzungen bezeichnet: Sie werden von Entwicklern während des Entwicklungsprozesses verwendet. Die von den Autoren beschriebenen Techniken sind jedoch als Nachweis, dass die Modelle, Methoden und Software korrekt erstellt worden sind, oder dafür, dass sie für Anwendungen geeignet sind, nicht ausreichend. Im Besonderen geben die von den Autoren beschriebenen Aktivitäten zur entwicklungsbedingten Einschätzung keinen Aufschluss über die Zusammenführung der verschiedenen numerischen

Methoden und der Genauigkeit der berechneten Ergebnisse für alle reagierenden Funktionen des Systems, die hier von Interesse sind.

Stevenson (1999) sowie die ersten beiden, in Tevenson, Gustafson (1998) und Larzelere (1998) genannten Referenzen waren unter den ersten Studien, in denen man sich über die Konsequenzen der Advanced Strategic Computing Initiative (ASCI) des Stockpile Stewardship-Projektes zur Verifikation und Bewertung der Modelle, der Methoden und der Computersoftware Gedanken gemacht hatte. Dieses Projekt hat zum Ziel, experimentelle Tests in großem Umfang durch berechnete und im Rahmen kleinere Experimente zu ersetzen. Die Studien machten sich Gedanken über dieses Ziel unter all den anderen, jemals unternommenen Projekten, diese komplexen physikalischen Phänomene und Prozesse zu modellieren. Diese drei Studien stellten die Bandbreite der Wirksamkeit hinsichtlich Verifikation und Bewertung zu Recht in Frage. Die Autoren dieser Studien waren nicht allzu optimistisch, dass man die immensen Herausforderungen hinsichtlich Verifikation und Bewertung durch den ASCI erfolgreich beschreiben könnte.

Die Ziele standen im Zusammenhang mit Software-Entwicklungen innerhalb des National Laboratory Systems, für welches derartige Verifikations- und Bewertungsprozeduren formal nicht existierten. Allerdings haben alle in das ASCI-Projekt involvierten Laboratorien die Herausforderungen angenommen und wichtige Beiträge zur Entwicklung und erfolgreichen Anwendung moderner Verifikations- und Bewertungsverfahren geleistet.

Die von Patrick Roache initiierten Entwicklungen hinsichtlich Verifikation und Bewertung mit signifikanten zusätzlichen Entwicklungen von William Oberkampf und Kollegen am Sandia National Laboratory und anderen sowie mit Beiträgen von Industrie und Akademien haben alle Fragen beantwortet, die in jenen ersten Studien von Stevenson und anderen aufgeworfen worden waren.

Aus ASCI entwickelte Methoden zur Verifikation und Bewertung

Die modernen Methoden zur Verifikation und Bewertung mathematischer Modelle, numerischer Methoden und der damit verbundenen Computer-Software sind der einfachen Auflistung der Bandbreite von Fehlern als Mittel zur Bestimmung der Qualität von Software weit überlegen. Die Bücher von Patrik Roache (1998, 2009) sowie Oberkampf und Roy (2010) haben die Evolution der Methoden dokumentiert. Zusätzlich haben Oberkampf und seine Kollegen am Sandia National Laboratory eine große Anzahl technischer Berichte des Labors erstellt, besonders Oberkampf, Trucano und Hirsch (2003). Die Methoden wurden erfolgreich angewendet auf eine Vielfalt wissenschaftlicher und Ingenieurssoftware. Und sie wurden von vielen wissenschaftlichen und professionellen Ingenieursgesellschaften übernommen als Maßgabe zur Veröffentlichung in begutachteten Journalen. Eine Eingabe in Suchmaschinen wie Google, Google Scholar oder www.osti.gov wird eine große Anzahl von Treffern ergeben.

Das Buch von Knupp und Salari (2002) über die Method of Manufactured Solutions (MMS), einer Methode, die zuerst von Roache eingeführt worden war, zeigt eine kraftvolle Methode zur Quantifizierung der Genauigkeit numerischer Methoden zur damit verbundenen theoretischen Effizienz und zur Entdeckung von

Verschlüsselungsfehlern. Auch hier wird die Suche nach Literatur zu einer großen Zahl nützlicher Berichte und Studien mit zahlreichen Anwendungsbeispielen führen. Die MMS sind goldener Standard zur Verifikation numerischer Lösungen.

Das Zählen von Fehlern ist fehlerhaft

Die Daten, auf denen die Studie von Pipitone und Easterbrook basiert, wurden von Pipitone (2010) zusammengetragen und präsentiert. Diese These und die Diskussionsstudie selbst sprechen die nicht so idealen Charakteristika der Fehlerbestimmung relativ zur Bestimmung der Softwarequalität an. Das Buch von Oberkampf und Roy (2010) widmet der Fehlerbestimmung einen einzelnen, langen Absatz. Die Rohdaten der These zeigen diese große Anzahl von Fehlern in absoluten Zahlen, die in den begutachteten GCMs präsent waren.

Das Zählen der Fehler führt nicht zu brauchbaren Beiträgen in drei der wichtigsten Attribute der Softwarequalität, wie der heute benutzte Satz zeigt: Verifikation und Bewertung [Verification and Validation (V&V)] sowie die Bestimmung der Ungewissheit [uncertainty qualification (UQ)]. Bei der modernen Softwareentwicklung ist die Verifikation ein mathematisches und die Bewertung ein physikalisches Problem, einschließlich der Art und Weise von Bewertungstests.

Das Zählen der Fehler wäre sinnvoller, wenn die Daten als Funktion der Zeit nach Einführung der Software präsentiert werden würden, um eine stetige Verbesserung und eine genaue Klassifizierung des Fehlers zu gewährleisten. Das Zählen wäre auch nützlicher, wenn es nur mit neueren Versionen der Software verbunden wäre. Und die Anzahl der von den Anwendern abgedeckten verschiedenen Reaktionsfunktionen ist auch von Interesse: eine sehr grobe Annäherung an die Modelle. Im Allgemeinen werden die verschiedenen Reaktionsfunktionen eine grobe Proxy für die Konzentration auf wichtige Teile der mathematischen Modelle sein.

Das Zählen der Fehler wäre noch viel nützlicher, wenn man auch den Fehlertyp betrachten würde. Es gibt folgende vier Fehlerklassen: (1) Fehler des Anwenders, (2) Verschlüsselungsfehler, (3) Grenzen des Modells oder der Methode und (4) Defizite im Modell oder der Methode. Von diesen zählt nur die zweite Klasse Verschlüsselungsfehler. Die erste Klasse, ein Fehler des Anwenders, könnte ein Hinweis darauf sein, dass eine Verbesserung bei der Dokumentation der Codes erforderlich ist, und zwar für die theoretische Basis der Modelle und Methoden und/oder die Anwendungsprozeduren und/oder das Verstehen der grundlegenden Natur der berechneten Systemreaktionen. Die dritte Klasse, Grenzen des Modells oder der Methode bedeutet, dass ein gewisser Grad der Repräsentation zwar vorhanden ist, ein Anwender aber eine Grenze entdeckt hat, die das Entwicklungsteam nicht erwartet hatte. Die vierte Klasse bedeutet, dass ein Anwender eine neue Anwendung und/oder Reaktion entdeckt hat, die in der Originalentwicklung fehlte. Diese vier erfordern allgemein eine signifikante Vorbereitung des Modells, der Methode und der Software-Modifikationen relativ zur Korrektur des Fehlers.

Die Punkte (3) und (4) könnten ein wenig mehr Klarstellung brauchen. Eine Grenze des Modells oder der Methode kann mit einem turbulenten Fluss

illustriert werden, für welchen das Personal, das das Originalmodell entwickelt hat, die Konstanten spezifiziert hat zu jenen, die mit parallelen Scherungsströmungen korrespondieren, und ein Anwender hat versucht, die Modellergebnisse mit experimentellen Daten zu vergleichen, deren Rückführung wichtig ist. Punkt (4) kann auch durch einen turbulenten Fluss illustriert werden. Man denke sich einen Fall, in welchem die Entwickler eine numerische Lösungsmethode verwendet haben, die nur für parabolische/fortlaufende physikalische Situationen möglich ist, obwohl der Anwender einem elliptischen Fluss gegenüberstand.

Die Studie von Pipitone und Easterbrook enthält nur wenige Informationen über die Art der Fehler, die man entdeckt hatte.

Eignung für Produktionsanwendungen

In der Studie von Pipitone und Easterbrook geht es nicht um die Aspekte der Anwendung der GCMs; stattdessen konzentriert sich die Studie auf die Lernaspekte des Modells. Wie in diesem Kommentar schon erwähnt, sind diese Aspekte allen Modellen gemeinsam, nämlich immer dann, wenn die Komplexität eine wichtige Komponente ist – Komplexität sowohl hinsichtlich der Physik als auch der Software.

Die Ziele der Modell- und Softwareentwicklung sind die Erzeugung von Tools und Anwendungsprozeduren, die für Vorhersagen ausreichender Genauigkeit in der realen Welt geeignet sind. Die Grundlage ausreichender Genauigkeit ist die Bewertung dieser Vorhersagen im Vergleich mit gemessenen Daten aus den Anwendungsregionen. Alle Funktionen der Systemreaktionen müssen durch Testbewertungen geprüft werden. Die Begutachtungsstudie von Easterbrook spricht keinerlei Aspekte der Bewertung an, da dieses Konzept in den Berichten und Studien definiert ist.

Die Bewertung ist für alle Modelle, Methoden, Software, Anwendungsprozeduren und die Anwender erforderlich, die die Basis politisch-öffentlicher Entscheidungen bilden. Die Bewertung muss nach der Verifikation erfolgen. Im Allgemeinen werden Verifikation und Bewertung dieser Arbeitsmittel und Prozeduren von Personen durchgeführt, die unabhängig von dem Team sind, das die Modelle und Prozeduren entwickelt hat. Die Auflistung von Fehlern, vor allem solcher, die während der Entwicklung zutage treten, hat in dieser Hinsicht nichts zu bieten.

Schlussfolgerung

Die Studie präsentiert ein sehr schwaches Argument für die Qualität der GCM-Software. Die weithin akzeptierten und erfolgreichen modernen Verifikations- und Bewertungsmethoden, welche in vielen wissenschaftlichen Softwareprojekten Verwendung finden, werden in der Studie nicht einmal erwähnt. Noch wichtiger, die Brauchbarkeit der GCMs für Anwendungen, die politische En tscheidungen beeinflussen, wird ebenfalls nicht erwähnt. Das einfache Aufzählen von Fehlern kann keine Informationen relativ zu Bewertung und Anwendung bei politischen Entscheidungen bieten.

References

Easterbrook, Steve M. and Johns, Timothy C., Engineering the Software for Understanding Climate, Computing in Science & Engineering, Vol. 11, No. 6, pp. 65 – 74, 2009.

Gustafson, John, Computational Verifiability and Feasibility of the ASCI Program, IEEE Computational Science & Engineering, Vol. 5, No. 1, pp. 36-45, 1998.

Knupp, Patrick and Salari, Kambiz, Verification of Computer Codes in Computational Science and Engineering, Chapman and Hall/CRC, Florida 2002.

Larzelere II, A. R., Creating Simulation Capabilities, IEEE Computational Science & Engineering, Vol. 5, No. 1, pp. 27-35, 1998.

Oberkampf, William F. and Roy, Christopher J., Verification and Validation in Scientific Computing, Cambridge University Press, Cambridge, 2010.

Oberkampf, William F., Trucano, T. G., and Hirsch, C., Verification, Validation, and Predictive Capability in Computational Engineering and Physics , Sandia National Laboratories Report SAND 2003-3769, 2003.

Pipitone, Jon, Software quality in climate modeling, Masters of Science thesis Graduate Department of Computer Science, University of Toronto, 2010.

Roache, Patrick J., Verification and Validation in Computational Science and Engineering, Hermosa Publishers, Socorro, New Mexico, 1998.

Roache, Patrick J., Fundamentals of Verification and Validation, Hermosa Publishers, Socorro, New Mexico, 2009.

Roache, Patrick J., Code Verification by the Method of Manufactured Solutions, Journal of Fluids Engineering, Vol. 114, No. 1, pp. 4-10, 2002.

Stevenson, D. E., A critical look at quality in large-scale simulations, IEEE Computational Science & Engineering, Vol. 1, No. 3, pp. 53â€“63, 1999.

Kommentar von Judith Curry: Zur Hintergrundinformation folgen hier einige Beiträge zur Klimamodellierung V&V:

- Climate model verification and validation
- Climate model verification and validation: Part II
- Verification, validation and uncertainty quantification in climate modeling
- What can we learn from climate models? Part II

Meine persönliche Ansicht hierzu liegt mehr auf der Linie dessen, was Anonymus hier präsentiert als auf dem, was von Pipitone und Easterbrook kommt.

Link: http://judithcurry.com/2012/04/15/assessing-climate-model-software-quality/

Übersetzt von Chris Frey EIKE