Mathe studiert – und dann?

Kristina Vaillant im Gespräch mit Nicola Botta

Im Potsdamer Wissenschaftspark unweit des Hauptbahnhofs in Potsdam steht auf dem Telegrafenberg das Potsdam-Institut für Klimafolgenforschung, kurz PIK. Nicola Botta arbeitet dort seit 20 Jahren. Wir treffen ihn im Neubau des Instituts, der mit seiner holzverschalteten Fassade wie in einen Tarnanzug gekleidet zwischen den Laubbäumen verschwindet. Begonnen hat Nicola Botta seine wissenschaftliche Laufbahn mit einer Doktorarbeit auf dem Gebiet der Strömungsmechanik. Über numerische Methoden für Wetter- und Klimamodelle ist er zu immer abstrakteren Forschungsfragen gelangt. Für die Klimafolgenforschung wünscht er sich mehr Mathematiker und Informatiker.

Herr Botta, warum braucht die Klimaforschung mehr Mathematiker?

Klimafolgenforschung ist notwendigerweise interdisziplinär. Die Grundlagen der Mathematik sind das, was Wirtschaftswissenschaftler, Physiker, Soziologen, Physiker, Politologen und Informatiker gemeinsam haben. Jeder, der eine universitäre Ausbildung genossen hat, kann mit Mengen, Funktionen und Relationen etwas anfangen. Darüber hinaus gibt es oft wenige Gemeinsamkeiten zwischen Klimaforschern. Und es ist allzu einfach, aneinander vorbei zu reden. Wenn wir miteinander reden wollen, sollten wir also auf dem Sprachkern aufbauen, den wir gemeinsam haben. Oft verschwinden Meinungsverschiedenheiten sehr schnell, wenn wir eine Frage oder ein Problem in der Sprache der Mathematik formulieren. Das Potsdam-Institut für Klimafolgenforschung gehört zur Leibniz-Gemeinschaft und Sie wissen, dass Gottfried Wilhelm Leibniz die Idee hatte: *calculemus*, lasst uns rechnen! Er suchte eine Sprache, die es erlaubt, Meinungsverschiedenheiten durch einen rechnerischen Zugang zu lösen.

Das klingt plausibel, wenn man weiß, dass am PIK Soziologen, Politologen, Mathematiker, Geowissenschaftler, Physiker und Ingenieure wie Sie eng zusammenarbeiten. Wie sieht es in ihrer Abteilung aus?

Bei uns im Forschungsfeld 4 „Transdisciplinary Concepts and Methods“ gibt es um die 50 Mitarbeiter und etwa 40 Gastwissenschaftler. Viele davon sind Physiker, Soziologen oder Politikwissenschaftler, es gibt aber auch einige Mathematiker, Informatiker, Meteorologen und Ingenieure. Die meisten Physiker und Mathematiker hier beschäftigten sich mit der Modellierung, oft auch mit der Implementation und der Simulation von Modellen. Ich habe als Ingenieur mit numerischen Simulationen von Strömungen, also konkreten Problemen, angefangen. Dann habe ich mich mehr und mehr für formale Methoden interessiert. Bei den Anwendungen treffen dann Formalisierung und Modellierung zusammen. In unserer Abteilung geht es darum, Konzepte und Methoden für die Entwicklung, die Validierung und die Verifikation von Modellen zu entwickeln. Modellierung spielt in der Klimafolgenforschung eine zentrale Rolle. Anfang der 90er Jahre war die

Forschung fokussiert auf die Modellierung natürlicher Systeme, wie das Wetter oder die Ozeane. Danach ging die Entwicklung auch in Richtung der Modellierung von sozioökonomischen Systemen.

Sie sind vor ziemlich genau 20 Jahren ans PIK gekommen. Damals haben Sie noch auf dem Gebiet der Strömungsmechanik gearbeitet.

Ja, ich habe mich 15 Jahre lang mit numerischen Methoden für Erhaltungssätze beschäftigt. Erhaltungssätze sind allgemeine Gleichungen, die unter anderem Bewegungen von Flüssigkeiten beschreiben. Als ich 1998 von der Freien Universität Berlin hierherkam, habe ich mich mit der Entwicklung von Diskretisierungen für Klima- und Wettermodelle beschäftigt. Diskrete Methoden sind das Bindeglied zwischen mathematischen Modellen und numerischen Simulationen. Auf dem Weg von der Realität bis zu Modellergebnissen gibt es ja eine sehr lange Kette von Zwischenmodellen. Typischerweise geht es los mit einem mathematischen Modell. Oft ist dieses Modell ein System von partiellen Differentialgleichungen. Das sind aber nicht die Gleichungen, die von einem Rechner gelöst werden. Diese ergeben sich aus einer Diskretisierung oder Approximation der Modellgleichungen. Danach muss das diskrete Modell implementiert werden. Dafür gibt es wiederum verschiedene Programmiersprachen und Paradigmen. Am Ende stehen Modellsimulationen und Ergebnisse.

Welchen Unterschied macht es, ob ich die Strömungen in einer Flugzeugturbine oder Strömungen in der Atmosphäre modelliere?

Konzeptuell gibt es keinen Unterschied, die Gleichungen sind im Prinzip dieselben. Aber die Randbedingungen sind anders, und oft muss man die Diskretisierung an die Besonderheiten der geophysikalischen Strömung anpassen. Ich will das an einem Beispiel deutlich machen. Als ich zum PIK kam, habe ich das damalige Wettervorhersagemodell des Deutschen Wetterdienstes untersucht. Da haben wir eine große Überraschung erlebt. Zuerst haben wir das Modell mit hydrostatischen Anfangswerten initialisiert. Bei solchen Anfangsdaten passiert eigentlich

nichts: weder horizontale Winde noch vertikale Bewegungen sollten entstehen. Aber nach zwei bis drei Stunden Simulationszeit konnten wir vertikale Geschwindigkeiten von etwa drei Metern pro Sekunde im Modell beobachten! Nicht aus physikalischen Gründen, sondern allein herbeigeführt durch numerische Fehler. Das ist absolut inakzeptabel. Wenn so etwas passiert, muss man zuerst verstehen, ob man es mit Implementationsfehlern oder mit Modellfehlern zu tun hat. Im zweiten Fall muss man einen Diskretisierungsansatz finden, der es erlaubt, den Fehler unter Kontrolle zu halten.

Und woran arbeiten Sie heute?

Heute arbeite ich mit formalen Sprachen, mit denen man die Lücke zwischen Modellen und Modellimplementationen schließen kann. Mit diesen Methoden kann man sicherstellen, dass ein Computerprogramm eine formale Spezifikation erfüllt. Das ist wichtig, um die Ergebnisse von Simulationen beurteilen zu können.

Zum Beispiel, Voraussagen darüber, was passiert, wenn die Europäische Union ihre CO₂-Emissionen nicht wie beschlossen bis 2030 um 40 Prozent gegenüber 1990 senkt.

Ja, zum Beispiel. Wenn man unerwartete Ergebnisse bekommt wie in dem oben beschriebenen Fall, gibt es prinzipiell zwei Möglichkeiten: Entweder ist das Modell nicht gut, oder es sind Fehler bei der Modellimplementierung passiert. Bei einem Quellcode von mehr als 100 000 Zeilen ist das nicht außergewöhnlich. Man möchte natürlich Programmierungsfehler ausschließen können. Dafür gibt es im Prinzip zwei Methoden: Testen und Beweisen. Programmtests sind enorm wichtig. Leider kann man durch Testen in den meisten Fällen nur die Präsenz von Fehlern zeigen, nicht aber ihre Abwesenheit. Seit ungefähr zehn Jahren haben wir Programmiersprachen, mit denen man die Korrektheit von Programmen konstruktiv sicherstellen kann. Mit solchen Sprachen könnte man im Prinzip

Klimamodelle implementieren, die per Konstruktion korrekt sind. Wir stehen hier allerdings ganz am Anfang, es wird noch mindestens zwei Jahrzehnte dauern, bis solche Sprachen für die Produktion geeignet sind. Man kann sie aber schon heute auf die Formalisierung von Problemen anwenden. Das hat eine unmittelbare Bedeutung für die Klimafolgenforschung. Eine Besonderheit der Klimaforschung ist nämlich, dass, im Gegensatz zu den Ingenieurwissenschaften, die Möglichkeiten der empirischen Verifikation ziemlich beschränkt sind. Wir können es uns nicht leisten, mit der Erde Experimente zu machen! Wir können nicht fünf oder sechs Länder nehmen, vielleicht kleinere wie Dänemark oder die Schweiz, dort für ein paar Jahrzehnte verschiedene Maßnahmen zur Senkung von Emissionen ausprobieren, um dann diejenigen Maßnahmen, die am besten funktioniert haben, global anzuwenden.

Sind nicht in der Klimawissenschaft gerade die Simulationen die eigentlichen Experimente?

Genau, Simulationen können aber relativ schlecht die vielen Unsicherheiten berücksichtigen, unter denen Entscheidungen notwendigerweise getroffen werden müssen. Solche Unsicherheiten haben aber einen erheblichen Einfluss darauf, welche Maßnahmen am besten funktionieren. Wir sind also bis zu einem gewissen Grad darauf angewiesen, Schlussfolgerungen zu ziehen, ohne auf empirische Experimente zurückgreifen zu können. Bei der Softwareentwicklung hat man, wie gesagt, eine ganz ähnliche Situation, denn die meisten Algorithmen kann man nicht einfach auf Korrektheit testen. Die Klimafolgenforschung kann also von der theoretischen Informatik lernen, wenn es darum geht, Lösungsansätze bereitzustellen, die beweisbar gewisse Spezifikationen erfüllen.

Sie sorgen mit diesen Methoden also dafür, dass diese Experimente, die keine Experimente im herkömmlichen Sinne sind, möglichst zuverlässig und korrekt ablaufen.

So weit sind wir noch nicht. Es wird noch Jahre dauern, bevor man Klimamodelle komplett spezifizieren kann. Man kann aber Fragen der Klimafolgenforschung untersuchen und zum Teil beantworten, ohne direkt auf Modelle angewiesen zu sein. Zum Beispiel, durch die Formalisierung von Entscheidungsproblemen. Wir haben in den letzten Jahren ein Software-Framework entwickelt, mit dem man ganz konkrete, wenn auch idealisierte, Entscheidungsprobleme spezifizieren und lösen kann. Die Lösungen sind dann beweisbar optimale Entscheidungen.

Können Sie das an einem Beispiel erläutern?

Eine wichtige Aufgabe der Klimafolgenforschung und damit auch des PIK besteht darin, belastbare Entscheidungsregeln zu errechnen. Nehmen wir das Beispiel der Senkung von CO₂-Emissionen. Wir wissen, dass wir diese so schnell wie möglich reduzieren müssen. Gleichzeitig sollten Wirtschaftswachstum und Wohlstand nicht gefährdet werden. Solche Entscheidungen sind notwendigerweise mit sehr großen Unsicherheiten behaftet. Zum Beispiel könnte die Politik entscheiden, Emissionen zu senken, aber diese könnten trotzdem zunehmen, weil ein Wirtschaftsboom eingesetzt hat oder weil die Politik die notwendigen Maßnahmen gar nicht oder nicht schnell genug umgesetzt hat. Mit unserem Software-Framework kann man systematisch untersuchen, wie solche Unsicherheiten optimale Entscheidungen beeinflussen. Wenn man die Unsicherheiten festlegt, kann man beweisbar optimale Entscheidungen ableiten.

Sie wollen politische Entscheidungen automatisieren?

Damit würde man zu weit gehen. Aus zwei Gründen. Erstens weil wir noch ganz am Anfang stehen. Um Entscheidungen unter Unsicherheiten zu automatisieren, müsste man unter anderem die Stabilität von Regierungen und die Bildung von Meinungen modellieren und vorhersagen können. In 20 oder 30 Jahren wird es vielleicht Modelle geben, die solche Prozesse beschreiben können. Im Moment muss man einfach erkennen, dass es viele Unsicherheiten gibt, die in den Modellen nicht berücksichtigt werden können. Zweitens würde ein solches Programm auf vielen Annahmen basieren, die vielleicht nicht falsch sind, die aber nur implizit vorhanden und daher für den Anwender nicht nachvollziehbar sind. Eine Automatisierung politischer Entscheidungen kann also nicht das Ziel sein.

Was ist dann das Ziel?

Ich würde eher sagen eine Rationalisierung. Es geht in erster Linie darum, die Probleme besser zu verstehen und

den Einfluss von Unsicherheiten auf optimale Lösungen abzuschätzen.

Kommen wir zu ganz konkreten Dingen, zu Ihrem Arbeitsalltag. Wie sieht der aus?

Ich sitze die meiste Zeit vor einem Rechner, wenn ich nicht gerade an einem Seminar teilnehme oder mich mit Kollegen austausche. Programmieren ist ein großer Bestandteil meiner Arbeit. Die Entwicklung des Software-Frameworks, über das ich gesprochen habe, hat einige Jahre gedauert. Und selbst Veröffentlichen heißt in meinem Fachgebiet meistens auch Programmieren.

Welche Tätigkeiten mögen Sie ganz besonders?

Ich spreche eigentlich sehr gerne über meine Forschung. Es ist natürlich ein Unterschied, ob man für eine Fachzeitschrift wie das *Journal of Functional Programming* schreibt, oder ob man versucht, Fachbegriffe einem breiteren Publikum nahezubringen. Letzteres macht mir auch Spaß, so wie dieses Gespräch. Aber ich finde es, ehrlich gesagt, ganz schön schwierig.

Was sollten Mathematiker mitbringen, wenn sie in der Klimaforschung arbeiten wollen?

Ich denke da nicht an besondere technische Qualifikationen, sondern eher an soziale Kompetenzen. Man sollte offen und neugierig sein. Manchmal ist es nicht so einfach zu verstehen, mit welchen Problemen sich die Kollegen beschäftigen. Wir haben am Institut keine klare Trennung zwischen Grundlagenforschung und Anwendungen. Und wir benutzen oft sehr unterschiedliche Sprachen. Ich neige selbst oft dazu, mich in eine Schublade zu stecken. Ich denke, jemand, der erfolgreich sein will, muss in der Lage sein, sich in verschiedene Schubladen hineinzudenken.

In welche Schublade – um bei Ihrem Bild zu bleiben – ordnen Sie sich selbst ein, verstehen Sie sich als Computerwissenschaftler oder Klimawissenschaftler?

Ich bin ein Ingenieur, der sich mit Fragen der Klimafolgenforschung beschäftigt. Dabei wende ich gerne Methoden der theoretischen Informatik an. An meinem Selbstverständnis als Ingenieur hat sich nicht viel verändert. Nur die Aufgaben, die haben sich im Laufe der Zeit geändert.

Vielen Dank für das Gespräch!

Kristina Vaillant ist freie Journalistin in Berlin und arbeitet regelmäßig für das Medienbüro der Deutschen Mathematiker-Vereinigung.

*www.vaillant-texte.de
(Fotos: Christoph Eyrich)*