

Differenzierbare Forschungssoftware

Uwe Naumann

RWTH Aachen, 52056 Aachen, naumann@stce.rwth-aachen.de

Auf mathematischer Modellbildung basierende numerische Simulation ist eine etablierte Methode für vertieftes Verständnis relevanter Probleme in den Natur-, Lebens- und Ingenieurwissenschaften. Entsprechende Softwarelösungen ermöglichen die quantitative Evaluation der Modelle mithilfe von Computern. Die zugrundeliegende numerische Simulationssoftware ist typischerweise mathematisch und algorithmisch anspruchsvoll, rechenintensiv, sowie approximativ und parametrisiert. Daraus folgt eine Reihe von Anforderungen sowohl an die mathematische Modellierung als auch an den (idealerweise simultan ablaufenden) Entwicklungsprozess für entsprechende Forschungssoftware (*Research Software Engineering; RSE*), welche in der etablierten Softwareentwicklung eine meist weniger zentrale Rolle einnehmen.

„Wie stark ändert sich das Resultat meiner Simulation bei Variation der Werte bestimmter freier (Eingabe-) Parameter?“ „Wie sensitiv ist diese Änderung bezüglich Variation von Werten potenziell anderer Parameter?“ Analoge Fragen stellen sich Anwenderinnen und Anwender numerischer Simulationssoftware regelmäßig. Zahlreiche numerische Methoden basieren auf der Verfügbarkeit dieser Informationen. Letztere übersetzt sich in Ableitungen erster und zweiter (ggf. auch höherer) Ordnung (bzw. sinnvolle Approximationen derer) der simulierten Größen bezüglich der Parameter.

Dieser Artikel stellt Differenzierbarkeit und die Berechnung entsprechender Ableitungen numerischer Simulationssoftware als Anforderung im RSE in den Mittelpunkt. Der Fokus liegt dabei auf kontinuierlichen (in Gegensatz zu diskreten) Simulationen. Die zentralen Aussagen gelten sowohl für mechanistische Modelle als auch für rein datengetriebene Ansätze des maschinellen Lernens inklusive hybrider Szenarien, in welchen beide Methoden zu einer Gesamtlösung kombiniert werden. Auf nichtfunktionaler Ebene ermöglichen Ableitungen Fehleranalyse und -kontrolle, Kalibrierung sowie robustere Softwaretests. Die in vielen Fällen essenzielle Transition von reiner Simulation der zugrundeliegenden Systeme hin zu deren (ableitungsbasierter) Optimierung ist oft zentraler Bestandteil der funktionalen Anforderungen. Die Grenze zwischen Funktionalität und Nichtfunktionalität von Differenzierbarkeit als Anforderung verschimmt zusehends im Kontext digitaler Zwillinge, welche unter anderem Fehlerkontrolle in Echtzeit zum Ziel haben können. In

jedem dieser Fälle muss die numerische Simulation differenziert werden.

Dafür unterscheidet man grob zwischen zwei Ansätzen. Beim symbolischen Differenzieren werden Ableitungen des mathematischen Modells analytisch hergeleitet. Zumeist handelt es sich dabei um eine anspruchsvolle manuelle Tätigkeit. Die Lösung des differenzierten Modells muss anschließend numerisch approximiert und somit in Form von Forschungssoftware implementiert werden. Die numerische Approximation der Ableitung entspricht meist nicht der korrekten Ableitung der numerischen Approximation des Originalmodells, was eine Reihe von Komplikationen bei der Verwendung dieser Werte im Rahmen von numerischen Methoden (z.B. Optimierungsalgorithmen) nach sich ziehen kann.

Algorithmisches Differenzieren (AD) [1,2] umgeht das oben skizzierte Problem durch Differenzieren des numerischen Simulationsprogramms mittels Kombination bekannter Ableitungen für dessen Elementarfunktionen gemäß der Kettenregel. Dieser Prozess kann zu großen Teilen automatisiert werden und ist somit auch auf sehr komplexe Simulationssoftware anwendbar. Die unter Ausnutzung von Assoziativität der Kettenregel generierten adjungierten Programme stellen Verallgemeinerungen der im Rahmen des maschinellen Lernens essenziellen *backpropagation* dar. Somit lassen sich erste Ableitungen skalarer Zielgrößen bezüglich einer potenziell sehr großen Anzahl freier Parameter effizient berechnen.

Im Folgenden soll Differenzierbarkeit im Kontext der zuvor genannten fundamentalen Eigenschaften numerischer Simulationssoftware positioniert werden. Es ergeben sich sowohl Möglichkeiten als auch Herausforderungen für RSE.

Numerische Simulationssoftware ist mathematisch und algorithmisch anspruchsvoll. Sie basiert meist auf Dekaden von Forschung und Entwicklung ganzer Wissenschaftsbereiche. Ein signifikanter Teil menschlicher Expertise ist in ihr „verborgen“. Das zentrale Ziel von RSE muss daher die Sicherung nachhaltiger Verfügbarkeit dieses „Schatzes“ inklusive seiner zukünftigen Erweiterungen sein. Dazu gehört nicht zuletzt auch der möglichst effektive und effiziente Einsatz der „Ressource Mensch“ mittels weitestgehender Formalisierung und Optimierung des Entwicklungsprozesses. Jedoch stellen sich bereits die Übersetzung der

zugrundeliegenden mathematischen Modelle in numerische Methoden und entsprechende Softwareentwürfe, algorithmische Details, Test- und Evolutionsstrategien als höchst anspruchsvoll dar. Eine rigorose Anforderungsanalyse ist ohne fundiertes Verständnis des modellierten Sachverhalts, der Mathematik inklusive assoziierter Numerik, der Simulationsumgebung bestehend aus Systemhardware und -software, sowie von Methoden der Softwareentwicklung bis hin zur eigentlichen Programmierung nicht möglich. Die Option einer hinreichenden Konzentration dieser Expertise in Individuen darf bezweifelt werden. Konsequenzen für die interdisziplinäre Lehre sollten abgeleitet werden. Zudem stellen sich weitergehende Herausforderungen an die Validierung von Konsistenz des mathematischen Modells und der zugehörigen numerischen Simulationssoftware sowie der Korrektheit letzterer. Zusätzlich zu den Resultaten der Simulation können hier Werte von Ableitungen als weitergehende Evidenz sehr hilfreich sein. Ableitungen können und sollten entsprechende Softwaretest- und kontinuierliche Integrationsstrategien erweitern. Mit jeder Ableitungsordnung wächst potenziell die Robustheit der Validierung. Zu diesem Zweck muss die numerische Simulationssoftware ausreichend oft differenzierbar sein sowie differenziert werden.

Numerische Simulationssoftware ist rechenintensiv. Ein signifikanter Teil der Entwicklungsarbeit fließt traditionell in die Minimierung der durch numerische Simulationen benötigten Ressourcen. Oft agieren diese Simulationen am Rande der Leistungsfähigkeit der jeweils aktuell verfügbaren Computerinfrastruktur mit dem Ziel vertretbarer Laufzeiten bei zulässigem Speicherbedarf. Effektive Vektorisierung, Parallelisierung und / oder Beschleunigung (z.B. mittels GPU) gehören zu den fundamentalen Voraussetzungen für eine effektive Nutzung des modernen IT-Ökosystems. Letzteres unterliegt einer Entwicklungsdynamik, welche die zentrale Rolle von Nachhaltigkeit beim Entwurf und der Umsetzung von numerischen Simulationen sowie von deren Ableitungen nochmals unterstreicht. Seit einigen Jahren und nicht zuletzt getrieben durch ressourcenhungrige Anwendungen des maschinellen Lernens rückt auch die Minimierung des Energiebedarfs immer mehr in den Fokus der Aufmerksamkeit. Zahlreiche Methoden zur Reduktion der Komplexität numerischer Simulationen bei idealerweise nur geringfügigen Abstrichen in deren Aussagekraft basieren auf Ableitungen der zu simulierenden Größen bezüglich einer meist sehr großen Anzahl an Zwischenwerten. Vernachlässigbare Sensitivität über Teilen des Definitionsbereiches erlaubt Modellreduktion mittels Spezialisierung der Originalsoftware. Wenig signifikante Variablen werden zu Konstanten. Etablierte Datenflussanalysen entfernen infolgedessen nicht mehr benötigte Teilrechnungen. Die numerische Simulationssoftware muss dafür nicht nur differenziert werden. Es muss auch eine Globalisierung der Ableitungsinformation er-

möglicht werden. Intervallarithmetik oder komplexere Relaxationen kommen dafür zum Einsatz.

Numerische Simulationssoftware ist approximativ und parametrisiert. Dem britischen Statistiker George Box wird die auch heute noch in weiten Teilen gültige Aussage „Alle Modelle sind falsch, aber einige sind nützlich“ zugeschrieben. „Alle numerischen Simulationen sind falsch“ folgt unmittelbar. Die „Nützlichkeit einiger“ ist jedoch kein Automatismus. Gerade vor dem Hintergrund der Lösung potenziell schlecht konditionierter Probleme mittels möglicherweise nicht uneingeschränkt stabiler numerischer Algorithmen stellt sich die Frage nach dem „Wie falsch?“ Zahlreiche Verfahren zur numerischen Fehleranalyse und -kontrolle basieren wiederum auf Ableitungen der simulierten Größen bezüglich der zumeist mit substanziellen Unsicherheiten belegten Eingabedaten und -parametern. Relaxation dieser Ableitungen erlaubt die meist höchst wünschenswerte Globalisierung der zunächst nur lokal gültigen Werte. Adaptivität in numerischen Verfahren beruht ebenfalls auf oft hochdimensionalen und damit notwendigerweise adjungierten Ableitungen der jeweiligen Zielgrößen bezüglich Variablen aus deren Definitionsbereichen. Diese resultiert zum Beispiel in lokaler Verfeinerung der Auflösung von Rechengittern oder in der Konzentration der Datengenerierung auf besonders sensitive Teile des Definitionsbereiches beim Training von Surrogatmodellen mit Methoden des maschinellen Lernens. Letzteres stellt einen Spezialfall der Kalibrierung (einer potenziell enormen Anzahl) freier Parameter von Forschungssoftware dar. In jedem dieser Szenarien muss letztere zu diesem Zweck (potenziell adjungiert sowie relaxiert) differenziert werden.

Die Entwicklung differenzierbarer mathematischer Modelle sowie deren Implementierung in Form von differenzierbarer Forschungssoftware stellt ein höchst anspruchsvolles interdisziplinäres Thema an der Schnittstelle von Mathematik, Informatik und einer Vielzahl von Anwendungsgebieten dar. Tendenziell wird diese Tatsache im Rahmen von RSE zu neuen differenzierbaren Modellierungstechniken und entsprechenden Softwareentwurfsmustern für die konsistente simultane Entwicklung von mathematischen, numerischen bzw. algorithmischen und Softwaremodellen führen müssen. Ultimativ werden neue massiv parallele und differenzierbare Programmiersprachen und Softwarebibliotheken benötigt.

Literatur

[1] Andreas Griewank, Andrea Walther (2008): Evaluating Derivatives. Principles and Techniques of Algorithmic Differentiation. Society for Industrial and Applied Mathematics.

[2] Uwe Naumann (2012): The Art of Differentiating Computer Programs. An Introduction to Algorithmic Differentiation. Society for Industrial and Applied Mathematics.