# Verifikation und Validierung im wissenschaftlichen Rechnen

Arbeitsbereich Wissenschaftliches Rechnen
Fachbereich Informatik
Universität Hamburg

Seminararbeit
Seminar Modellierung und Simulation
Sandra Schröder

Wintersemester 2012/2013

Betreuerin: Petra Nerge

# Abstract

Diese Seminararbeit behandelt die Methoden und Konzepte der Verifikation und Validierung im wissenschaftlichen Rechnen. Ziel ist es, mit diesen Methoden die Korrektheit des gewählten mathematischen Modells und sein entsprechendes Computermodell zu prüfen. Es wird zunächst die Verifikation vorgestellt, dann die Validierung. Diese Reihenfolge entspricht dem Vorgehen, wie es in der Praxis des wissenschaftlichen Rechnens der Fall ist.
Die Verifikation teilt sich in zwei Bereiche, der Codeverifikation und der Lösungsverifikation. Der Quellcode des Computermodells wird mit Codeverifikationskriterien geprüft. Der Grad der Genauigkeit stellt das wichtigste Kriterium dar. Die Lösungsverifikation beschäftigt sich mit der Abschätzung von Fehlern, die bei numerischen Berechnungen auftreten können. Es gibt dabei vier Fehlertypen: Rundungsfehler, Stichprobenfehler, iterativer Fehler und Diskretisierungsfehler.
In der Validierung werden die Ergebnisse der Simulation mit den Ergebnissen von Validierungsexperimenten verglichen. Der Vergleich erfolgt mittels Validierungsexperimenten.

# Inhaltsverzeichnis

# 1 Einleitung

In der Forschung sind Experimente ein wesentlicher Bestandteil des Erkenntnisgewinns. Mithilfe von Experimenten wird eine Theorie bestätigt oder das Verhalten von Systemen und Prozessen besser verstanden, vorhergesagt oder optimiert. Mit der rasanten Entwicklung des Computers gibt es neue Möglichkeiten, Experimente durchzuführen. Das sogenannte *wissenschaftliche Rechnen* kombiniert die Disziplinen Mathematik, Informatik und Naturwissenschaften miteinander und bildet neben dem traditionellen Experiment und der Theorie die „dritte Säule"der Naturwissenschaft. Experimente am Computer ersetzen reale Experimente, die nicht möglich, zu gefährlich oder zu teuer sind. Das wissenschaftliche Rechnen erfordert eine interdisziplinäre Interaktion zwischen den unterschiedlichen Fachgebieten. Die Mathematik wird genutzt, um Fragestellungen aus der Naturwissenschaft zu modellieren. Sie dient als gemeinsame Sprache für die unterschiedlichen Disziplinen. Die Informatik bietet Werkzeuge und Methoden, um die mathematische Beschreibung des Systems in eine Repräsentation zu bringen, die vom Computer interpretiert wird.
Die formale Beschreibung des Systems ist das *Modell*. Es beschreibt dessen wichtigsten Eigenschaften und abstrahiert von anderen für den Kontext unwichtigen Eigenschaften. Es unterstützt, das Verhalten des Systems unter bestimmten Bedingungen besser zu verstehen, zu kontrollieren und vorherzusagen. Bei der Modellierung des Systems müssen viele Faktoren berücksichtigt werden. Innerhalb des Systems ist die Beschreibung der Geometrie, der Initialbedingungen und der physikalischen Parameter von Bedeutung. Es wirken auch Komponenten von außerhalb auf das System. Diese werden in Form von *Randbedingungen* angegeben.
Wurden die signifikanten Eigenschaften des Systems in einem Modell formuliert, wird dieses benutzt, um damit ein Ergebnis eines Experiments zu berechnen beziehungsweise vorherzusagen. Das ist die Aufgabe der *Simulation*. Im wissenschaftlichen Rechnen ist der Zweck der Simulation die *Vorhersage* der Systemantwort - der sogenannten *System Response Quantity* - unter bestimmten Eingaben auf der Grundlage des vorliegenden Modells.
Bis jetzt wurden die Begriffe Modell und Simulation sehr allgemein beschrieben. Bei der Modellierung eines Systems stellt sich die Frage, wie dessen Eigenschaften beschrieben werden. Im wissenschaftlichen Rechnen nutzt man dafür *mathematische Modelle*, die unter anderem durch partielle Differentialgleichungen (kurz: PDE) gegeben sind. Sie eignen sich am besten, um die Veränderung einer Variable in Abhängigkeit einer oder mehreren Größen zu beschreiben. Üblich ist es, die Änderung der Variable abhängig von Zeit und Ort zu beschreiben. Sie wird auch *abhängige* Variable genannt. Zeit und Ort sind dementsprechend *unabhängige* Variablen, da sie nicht von anderen Größen abhängen und nicht

von ihnen manipuliert werden können. Wenn im weiteren Verlauf von einem mathematischen Modell oder mathematischen Gleichungen die Rede ist, sind stets PDEs gemeint. Diese sind schwer oder gar nicht analytisch lösbar. Die Lösung muss unter anderem numerisch gelöst werden.
Die Simulation wie sie im vorigen Abschnitt beschrieben wurde, ist konkret eine *Computersimulation*, das heißt eine Implementierung des mathematischen Modells. Sie enthält die PDEs und Algorithmen zu deren Lösung. Zudem legt die Implementierung fest, welche Methoden für die Diskretisierung der PDEs angewendet werden. Die PDEs, die im Computermodell implementiert wurden, sind diskrete Gleichungen. Dies ist nötig aufgrund der eingeschränkten Genauigkeit der Computerarithmetik. Oft enthalten (kontinuierliche) PDEs Zeit -und Ortableitungen. Deshalb müssen bei der Diskretisierung endliche Zeitschritte und Gitterabstände festgelegt werden.
Die Lösung der diskreten Gleichungen ist die Simulation. Durch die Berechnung der Gleichungen simuliert das zur Simulation entsprechende Computermodell das System, welches durch das mathematische Modell formal beschrieben wurde. Man muss sich bei diesem Prozess bewusst sein, dass nicht die exakte Lösung des mathematischen Modells bestimmt wird. Sie wird stattdessen indirekt über das entsprechende diskrete Computermodell näherungsweise bestimmt.
Dabei stellt sich jedoch die Frage nach der Korrektheit der gewählten Modelle und ihrer Umsetzung. Genau dies ist die Aufgabe der Verifikation und Validierung. Zentrale Fragestellungen sind „Wurde das mathematische Modell richtig implementiert?“ (Verifikation) und „Gibt das Modell die Realität gut wieder?“ (Validierung). Diese Seminararbeit gibt einen Einblick darüber, welche Möglichkeiten es gibt, ein mathematisches Modell und sein entsprechendes Computermodell zu verifizieren und zu validieren.

# 2 Verifikation und Validierung

Es existieren einige Definitionen zu den Begriffen *Verifikation* und *Validierung*. Diese Definitionen orientierten sich jeweils sehr stark an dem Anwendungsgebiet der Organisation in der sie entstanden sind. Während die Definition des *IEEE* eher dem Bereich des Software-Engineerings angepasst ist, betrachtet die Organisation *American Institute of Aeronautics and Astronautics* die Definitionen von einer anderen, naturwissenschaftlichen Seite. Ihre Definitionen für Verifikation und Validierung werden im Rahmen dieser Seminararbeit ebenfalls benutzt [OR10]:

**Verifikation** *The process of determining that a model implementation accurately represents the developer's conceptual description of the model and the solution of the model.*

Die Definition für die Verifikation beschreibt zwei Teilbereiche. Dies ist zum einen die Überprüfung, ob die Modellimplementierung - das Computermodell - das mathematische Modell genau genug repräsentiert. Dies ist die Aufgabe der *Code-Verifikation* (Kapitel 3). Der andere Teilbereich - die *Lösungsverifikation* (Kapitel 4) - untersucht die Genauigkeit der Lösung des Modells.

**Validierung** *The process of determining the degree to which a model is an accurate representation of the real world from the perspective of the intended use of the model.*

Die Validierung (Kapitel 5) vergleicht das Computermodell mit der Realität - der zu modellierende und zu simulierende Sachverhalt - durch Überprüfung der Genauigkeit des Computermodells bezüglich definierter Genauigkeitsanforderungen an das Modell. Hiermit wird ebenfalls überprüft, inwiefern das Modell in der Lage ist, ein Ergebnis, welches bisher nicht validiert wurde, mit bestimmten Eingaben vorherzusagen. Man spricht von der sogenannten *Vorhersagefähigkeit* [OR10]. Es gibt dabei die drei Aspekte der Validierung, die in Kapitel 5 genauer vorgestellt werden.
Unter Genauigkeit (englisch *Accuracy*) versteht man in beiden Fällen den Grad der Übereinstimmung zwischen dem Computermodell und dem mathematischem Modell beziehungsweise der Realität. In diesem Zusammenhang ist nicht die Bit-Genauigkeit gemeint (englisch *Precision*).

In Abbildung 2.1 sind die drei Komponenten der Modellierung und Simulation mitsamt der Verifikation und Validierung dargestellt. Dazu gehört die Realität beziehungsweise ein Ausschnitt davon, der analysiert und durch ein konzeptuelles Modell beschrieben und abstrahiert wird. Das konzeptuelle Modell ist mit dem mathematischen gleichzusetzen. Das

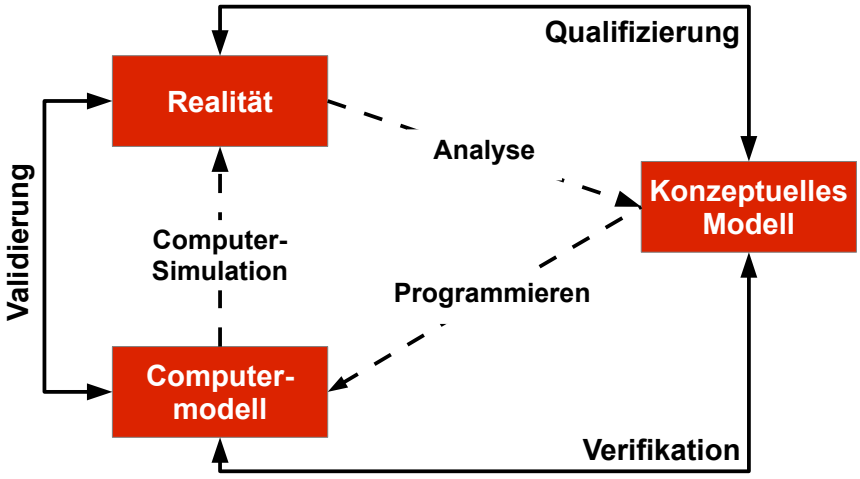



**Abbildung 2.1:** Einordnung von Verifikation und Validierung in die Modellierung und die Simulation (Nachzeichnung aus [OR10])

konzeptuelle Modell wird implementiert und daraus entsteht das Computermodell. Dieses simuliert den Ausschnitt aus der Realität. Wie aus den Definitionen deutlich wurde, geschieht die Verifikation zwischen dem Computermodell und dem konzeptuellen Modell und die Validierung zwischen dem Computermodell und der Realität. Die Qualifizierung zwischen Realität und konzeptuellen Modell ist nicht Thema der Seminararbeit, aus Gründen der Vollständigkeit wurde diese jedoch mit in die Abbildung aufgenommen.

# 3 Codeverifikation

Bei der Codeverifikation wird sichergestellt, ob das Computerprogramm eine gute Repräsentation des mathematischen Modells ist. Bei der Codeverifikation wird die Genauigkeit der Repräsentation des mathematischen Modells in Form des Computermodells untersucht. Dies bedeutet, dass die Implementierung des mathematischen Modells, der Quellcode, auf Fehler geprüft werden muss. Dafür gibt es unterschiedliche Kriterien, die erfüllt werden müssen. Sie unterscheiden sich in ihrer Strenge und sind unterschiedlich schwer zu erfüllen. Das strengste Kriterium ist hierbei der Grad der Genauigkeit, der in diesem Kapitel genauer erläutert wird. Voraussetzung dieser Analysen ist stets, dass die exakte Lösung des mathematischen Modells bekannt ist.

## 3.1 Grad der Genauigkeit

Dieses strenge Verifikationskriterium betrachtet das konvergente Verhalten der exakten Lösung der diskreten Gleichungen, wenn der Zeitschritt und/oder der Gitterabstand systematisch verfeinert werden. Das Ziel ist zu erfahren, ob sich mit feineren diskreten Parametern die diskrete Lösung der exakten mathematischen Lösung annähert. Dabei unterscheidet man zwischen dem theoretischen (Abschnitt 3.1.3) und dem empirischen Grad (Abschnitt 3.1.5) der Genauigkeit. Der theoretische Grad wird anhand der gewählten Diskretisierungsmethode bestimmt. Er gibt an, wie genau eine gewählte Diskretisierungsmethode für bestimmte PDEs sein kann. Man vergleicht die exakte Lösung der diskreten Gleichungen mit der Lösung des mathematischen Modells. Dies geschieht mittels einer Analyse des *Abbruchfehlers* und des *Diskretisierungsfehlers*.
Der empirische Grad bestimmt sich aus der bereits vorhandenen Implementierung des mathematischen Modells, also dem Computermodell. Zeitschritt und/oder Gitterabstand werden in mindestens zwei Stufen verfeinert und das Ergebnis des Modells auf jedem dieser Gitter neu berechnet. Wie bei dem theoretischen Grad der Genauigkeit wird das Verhalten der Lösung der diskreten Gleichungen untersucht.

### 3.1.1 Diskretisierungsfehler

Man unterscheidet bei dem Vergleich zwischen dem mathematischen Modell und dem Computermodell zwei Größen.
Dies ist zum einen die Abweichung der Lösung der diskreten Gleichungen von der Lösung des mathematischen Modells. Sie wird als Diskretisierungsfehler $\epsilon_h$ bezeichnet.

$$\epsilon_h = u_h - \tilde{u} \tag{3.1}$$

$u_h$ ist die exakte Lösung der diskreten Gleichung, $\tilde{u}$ entspricht der exakten Lösung zu den mathematischen Gleichungen. Der Index $h$ repräsentiert die diskreten Parameter. $\epsilon_h$ wird null, wenn $u_h$ mit $\tilde{u}$ übereinstimmt, das heißt keine Abweichung zwischen der Lösung der diskreten Gleichung und der Lösung der mathematischen Gleichung besteht.

### 3.1.2 Abbruchfehler

Die zweite Größe ist die Abweichung der diskreten Gleichungen zu den korrespondierenden Gleichungen des mathematischen Modells. Diese Abweichung wird als Abbruchfehler bezeichnet. Dieser Fehler tritt auf, wenn PDEs durch diskrete Gleichungen approximiert werden.
Der Abbruchfehler tritt beispielsweise bei der Approximation der Lösung durch eine Taylorreihe anhand der ersten Summanden auf. Als Rest entsteht ein Term, der den Fehler dieser Annäherung angibt. Dieser ist der Abbruchfehler. Als Beispiel sei die Taylorreihe der Exponentialfunktion genannt

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + ... \tag{3.2}$$

Sie kann durch die ersten drei Summanden approximiert werden

$$e^x = 1 + x + \frac{x^2}{2!} + E(x) \tag{3.3}$$

$E(x)$ bezeichnet den Fehlerterm. Er kann genau bestimmt werden. Es ist aber auch üblich, die Ordnung der abgeschnittenen Terme in O-Notation anzugeben. Je mehr Summanden für die Approximation genutzt werden, umso genauer wird sie und der Fehler $E(x)$ wird kleiner.

### 3.1.3 Theoretischer Grad der Genauigkeit

Der theoretische Grad der Genauigkeit ist die Konvergenz des Diskretisierungsfehlers mit zunehmend feinerer Diskretisierung, zum Beispiel mit feineren Gitterabständen oder feineren Zeitschritten. Diese Rate wird aus dem Abbruchfehler hergeleitet. Dabei werden die abhängigen Variablen der diskreten Gleichung nach Taylorreihe entwickelt und in die diskrete Gleichung eingesetzt. Man erhält eine Beziehung zwischen der diskreten Gleichung, dem mathematischen Modell und dem Abbruchfehler. Diese Beziehung wird *Generalized Truncation Error Expression* - kurz *GTEE* - genannt und hat folgende Form

$$L_h(u) = L(u) + TE_h(u) \tag{3.4}$$

Die Herleitung des GTEE kann in [OR10] nachgelesen werden. $L_h(u)$ bezeichnet die diskrete und $L(u)$ die mathematische Gleichung mit der abhängigen Variable $u$. Die diskrete Gleichung $L_h(u)$ ist also die Summe aus dem Abbruchfehler $TE_h(u)$ (englisch truncation error) und der mathematischen Gleichung $L(u)$. Mit dieser Beziehung lässt sich neben dem

theoretischen Grad der Genauigkeit die *Konsistenz* einer numerischen Methode überprüfen. Eine numerische Methode ist konsistent, wenn sich die diskreten den mathematischen Gleichungen annähern, wenn die diskreten Parameter $h$ gegen null gehen. Dies bedeutet außerdem, dass der Abbruchfehler kleiner wird beziehungsweise im Idealfall verschwindet. Dann entsprechen die diskreten genau den mathematischen Gleichungen.
In den Abschnitten 3.1.1 und 3.1.2 wurden der Abbruchfehler und der Diskretisierungsfehler besprochen. Mit dem GTEE aus Gleichung 3.4 kann gezeigt werden, inwiefern diese beiden Fehler zusammenhängen.
Intuitiv sollte sich der Diskretisierungsfehler verringern, wenn der Abbruchfehler kleiner wird. Denn wird die Abweichung zwischen diskreter und mathematischer Gleichung geringer, sollte das Ergebnis der diskreten Gleichung umso genauer werden und sich der exakten Lösung der mathematischen Gleichung annähern. Durch Umformen des GTEE ergibt sich

$$L(\epsilon_h) = -TE_h(u_h) \tag{3.5}$$

Gleichung 3.5 wird als *Continous Discretization Error Transport Equation* [OR10] bezeichnet und beschreibt, wie sich der Diskretisierungsfehler gemäß der mathematischen Gleichung $L(\cdot)$ fortpflanzt. Zudem ist in dieser Gleichung erkennbar, wie sich die intuitive Vorstellung über den Zusammenhang zwischen den beiden Fehlern bestätigt. Durch die Reduktion des Abbruchfehlers mittels Verfeinerung der Diskretisierungs-Parameter wird der Diskretisierungsfehler entsprechend Gleichung 3.5 reduziert. Der kleinste Exponent eines diskreten Parameters im Abbruchfehler $TE_h(u)$ ist der theoretische Grad der Genauigkeit.

### 3.1.4 Bestimmung des theoretischen Grad der Genauigkeit anhand der Wärmeleitungsgleichung

In diesem Abschnitt wird beispielhaft gezeigt, wie man aus einer partiellen Differentialgleichung und ihrer Diskretisierung den theoretischen Grad der Genauigkeit bestimmen kann [OR10]. Hier werden die wichtigsten Schritte zur Anwendung der hergeleiteten Gleichung aus Abschnitt 3.1.3 gezeigt.
Die Wärmeleitungsgleichung ist eine partielle Differentialgleichung der Form

$$L(T) = \frac{\partial T}{\partial t} - \alpha \frac{\partial^2 T}{\partial x^2} = 0 \tag{3.6}$$

Die abhängige Variable $T$ bezeichnet in dieser Gleichung die Temperatur, dessen zeitliche (Variable $t$) und räumliche (Variable $x$) Änderung bestimmt werden soll. Die Konstante $\alpha$ gibt die Temperaturleitfähigkeit des betrachteten Mediums an.
Gleichung 3.6 wird mittels der Methode der *Finiten Differenzen* diskretisiert

$$L_h(T) = \frac{T_i^{n+1} - T_i^n}{\Delta t} - \alpha \frac{T_{i+1}^n - 2T_i^n + T_{i-1}^n}{(\Delta x)^2} = 0 \tag{3.7}$$

Der erste Summand entspricht der diskreten Darstellung des ersten Summanden in Gleichung 3.6. Der zweite Summand ist dementsprechend die Diskretisierung des zweiten Summanden in der kontunierlichen Wärmeleitungsgleichung. Es wird hier nicht weiter darauf eingegangen, wie die Methode der Finite Differenzen genau funktioniert. Für eine grundlegende Beschreibung des Verfahrens wird auf [HS06] verwiesen.
Index $i$ bezeichnet die räumliche Position und Index $n$ einen Zeitpunkt. Der diskrete Parameter $\Delta t$ gibt den Zeitschritt an. $\Delta x$ beschreibt entsprechend den Abstand zwischen zwei Gitterpunkten.
Um den Abbruchfehler von $L_h(T)$ zu bestimmen, werden die Temperaturwerte $T_i^{n+1}$, $T_{i+1}^n$ und $T_{i-1}^n$ mittels Taylorreihenetwicklung erweitert und in die diskrete Gleichung $L_h(T)$ eingesetzt. Mit diesem Vorgehen zeigt sich genau der Zusammenhang, wie er durch den GTEE in Gleichung 3.4 beschrieben wird. Man erhält dann für den Abbruchfehler:

$$TE_h(T) = \left[\frac{1}{2}\frac{\partial^2 T}{\partial t^2}\right]\Delta t + \left[-\frac{\alpha}{12}\frac{\partial^4 T}{\partial x^4}\right](\Delta x)^2 + O(\Delta t^2, \Delta x^4) \tag{3.8}$$

Wie in Abschnitt 3.1 gezeigt, lässt sich nun der theoretische Grad der Genauigkeit aus dem Abbruchfehler ablesen. Dies ist der kleinste Exponent der diskreten Parameter $\Delta t$ und $\Delta x$. Bezüglich der Zeit hat die Diskretisierung nach der Methode der finiten Differenzen eine Genauigkeit erster Ordnung und bezüglich des Raums eine Genauigkeit zweiter Ordnung.

### 3.1.5 Empirischer Grad der Genauigkeit

Nachdem die Genauigkeit der Diskretisierungsmethode mittels Analyse des Abbruchfehlers bestimmt wurde, wird ermittelt, welche Genauigkeit tatsächlich mit der Implementierung der Methode erreicht werden kann.
Hier betrachtet man nicht den Grenzfall $h \to 0$, sondern nur eine Annäherung davon, indem man systematisch das implementierte Gitter in mindestens zwei Stufen [OR10] verfeinert. Die Erwartung ist, dass sich mit feinerem Gitter der Diskretisierungsfehler verringert. Dies ist in Abbildung 3.1 schematisch dargestellt. Das Gitter wird hier verfeinert, indem die Gitterpunkt-Abstände bei jedem Schritt halbiert werden. Diese Darstellung ist nur eine abstrakte Veranschaulichung der Verfeinerungs-Prozedur. Tatsächlich gibt es einige Verfahren, die abhängig von der Diskretisierungs-Methode und von der geometrischen Komplexität das Gitter angewandt werden. Diese Verfahren werden im Rahmen der Seminararbeit nicht weiter besprochen, können aber in [HS06] nachgelesen werden.
Der empirische Grad der Genauigkeit entspricht etwa dem Verhältnis der Diskretisierungsfehler zweier aufeinanderfolgender Gitter.

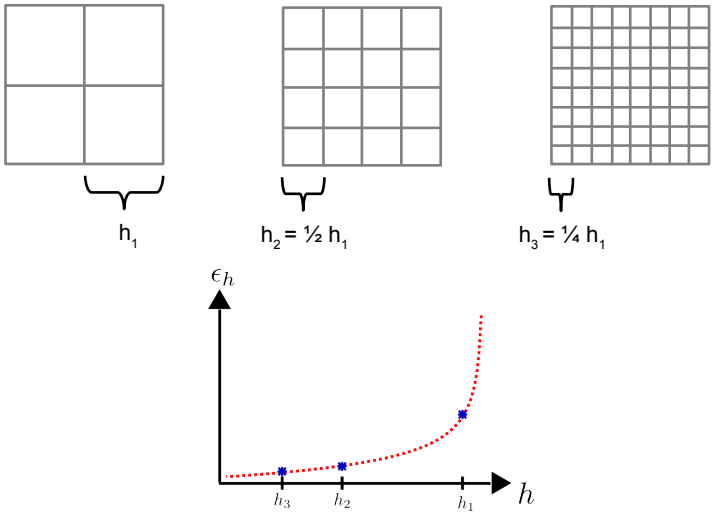



**Abbildung 3.1:** Systematische Gitterverfeinerung in zwei Stufen und Erwartung über die Reduktion des Diskretisierungsfehlers.

### 3.1.6 Aufdecken von Programmierfehlern mit dem Grad der Genauigkeit

Hat man den theoretischen und den empirischen Grad der Genauigkeit bestimmt, können durch den Vergleich der beiden Größen Programmierfehler aufgedeckt werden. Die Idee ist, die Genauigkeit der aktuellen, eventuell fehlerbehafteten Implementierung mit der Genauigkeit zu vergleichen, die theoretisch mit der gewählten Diskretisierungs-Methode erreicht werden kann. Wurde die theoretisch erreichbare Genauigkeit nicht mit der vorliegenden Implementierung erreicht, liegt ein Fehler im Programmcode vor.
Der Verlauf dieser Überprüfung, ist in Abbildung 3.2 dargestellt. Es wird zunächst die theoretische Konvergenz als Referenz bestimmt, nachdem das mathematische Modell, die Diskretisierungs-Methode für die Gleichungen und die numerischen Algorithmen gewählt wurden (diese Schritte sind im Diagramm nicht aufgezeichnet). Als nächster Schritt wird der empirische Grad der Genauigkeit ermittelt. Dafür wird eine exakte Lösung des mathematischen Modells benötigt, da, wie in Abschnitt 3.1.5 erläutert, der Diskretisierungsfehler bestimmt werden muss, um den Grad der Genauigkeit berechnen zu können. Dementsprechend muss jeweils die numerische Lösung auf unterschiedlich feinen Gittern ermittelt werden. Nach der Berechnung des empirischen Grads der Genauigkeit, kann dieser mit dem theoretischen Grad verglichen werden. Ist keine Übereinstimmung zu erkennen, muss das Programm nach Fehlern durchsucht werden (Debugging-Schritt im Diagramm). Es kann vorkommen, dass der theoretische Grad kleiner als der empirische ist. Die Ursache könnte sein, dass der theoretische Grad falsch bestimmt wurde oder sich an einer Stelle im Programm Fehler auslöschen und einen positiven Effekt auf die empirische Genauigkeit haben.
Wurde der Fehler im Programmcode gefunden und behoben, muss die ganze Prozedur

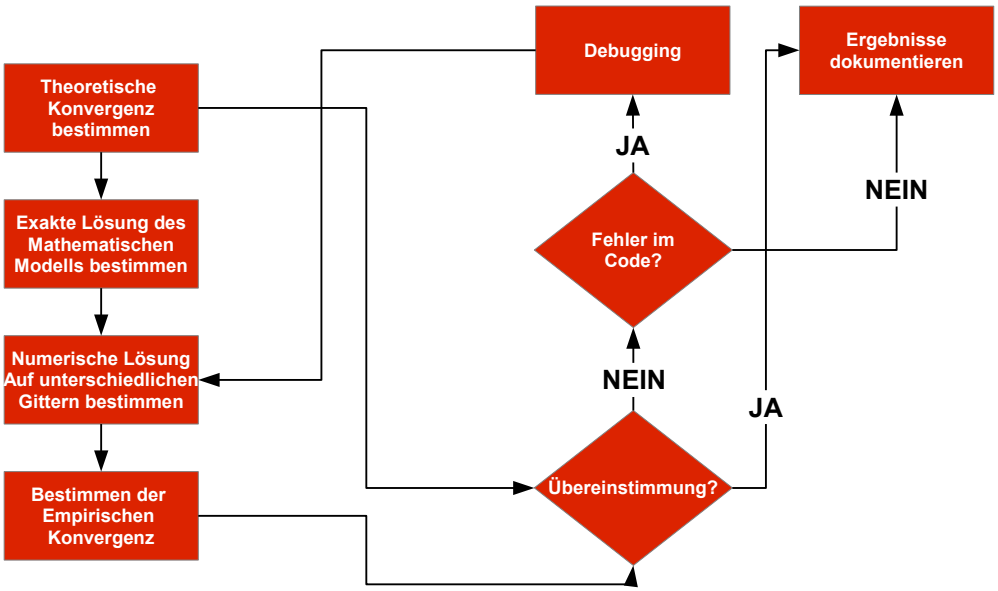



**Abbildung 3.2:** Verwendung des theoretischen und empirischen Grad der Genauigkeit zur Identifikation von Implementierungsfehlern (Entnommen aus [OR10] und im Layout angepasst).

wiederholt werden, bis theoretischer und empirischer Grad übereinstimmen.

# 4 Lösungsverifikation

Wie in der Einleitung beschrieben wurde, dient eine Simulation dazu, eine Vorhersage über den Zustand eines Systems zu machen. Dafür müssen die Ergebnisse, die die Simulation liefert, genau genug sein, damit die Vorhersage nicht falsch oder zu weit vom Ergebnis entfernt ist, welches ein reales Experiment liefern würde.
Ziel der Lösungsverifikation ist es zu überprüfen, ob das Ergebnis einer Simulation für einen bestimmten Zweck die Genauigkeitsanforderungen erfüllt. Dafür muss das Ergebnis auf Fehler und Abweichungen untersucht werden. Insgesamt gibt es vier Fehlertypen, die in diesem Abschnitt vorgestellt werden. Zwei davon - der Rundungsfehler (Abschnitt 4.1) und der Diskretisierungsfehler (Abschnitt 4.4) - treten immer in der Lösung auf, der Stichprobenfehler (Abschnitt 4.2) und der iterative Fehler (Abschnitt 4.3) nur abhängig von dem gewählten Verfahren.

## 4.1 Rundungsfehler

Aufgrund der eingeschränkten Genauigkeit des Computers lassen sich nicht alle reellen Zahlen auf die Menge der Maschinenzahlen abbilden. Dies kann beispielsweise bei einer arithmetischen Operation passieren, bei der das Ergebnis eine höhere Genauigkeit benötigt als ein Standard für die Abspeicherung von Zahlen (zum Beispiel der *IEEE-754-Standard*) bieten kann. Damit das Ergebnis jedoch abgespeichert werden kann, muss so gerundet werden, dass die Zahl mit dem Zahlenformat konform ist. Wenn die Genauigkeit zum Beispiel zu klein gewählt wird, wird der Ausdruck $3.0 * (1.0/3.0)$ zu 0.9999999 ausgewertet, statt zu 1.0.
Programmiersprachen wie *C/C++* oder *Fortran* bieten jeweils verschiedene Arten an, Genauigkeiten für Fliesskommazahlen zu definieren. In C/C++ gibt es die drei Datentypen *float* (einfache Genauigkeit - 32 Bit), *double* (doppelte Genauigkeit - 64 Bit) und *long double* (128 Bit). Der Datentyp *long double* ist allerdings Compiler-abhängig, während die anderen beiden Datentypen dem Standard folgen. Abhängig von der Wahl des Datentyps wird das Ergebnis genauer und weicht weniger vom wahren Ergebnis ab.
Fortran verwendet das sogenannte *KIND*-Attribut zur Angabe der gewünschten Genauigkeit für Fließkommazahlen:

```
REAL(KIND = X)
```

Die Variable `X` ist ein Integer und nimmt üblicherweise die Werte 4 für einfache Genauigkeit und 8 für die doppelte Genauigkeit entgegen. Dies ist allerdings Compiler-abhängig. Deshalb bietet Fortran die Plattform -und Compilerunabhängige Funktion

`SELECTED_REAL_KIND` an, mit der die gewünschte Genauigkeit definiert werden kann. Für diese gibt die Funktion den KIND-Parameter zurück.
Um den Effekt des Rundungsfehlers im Ergebnis abzuschätzen, führt man die Simulation mit der gewünschten Genauigkeit aus und wiederholt die Simulation anschließend mit einer höheren Genauigkeit. Wichtig ist dabei, dass bei beiden Simulations-Läufen die gleichen Auflösungen (zeitlich und räumlich) verwendet werden.

## 4.2 Stichprobenfehler

Wendet man statistische Methoden an, um Prozesse zu modellieren, die selbst stochastisch ablaufen, entstehen Stichprobenfehler.
In vielen Fällen kann eine Größe nicht direkt bestimmt werden. Stattdessen wird sie durch mehrere Messungen abgeschätzt. Diese Abschätzung ist oft eine Mittelung der gemessenen Quantitäten. Die Genauigkeit dieser Mittelung, das heißt die Abweichung des Mittelwertes des abgeschätzten Wertes der Größe zu dessen wahren Wert, hängt von der Anzahl der durchgeführten Messungen ab. Je mehr Messungen durchgeführt wurden, umso genauer wird dementsprechend die Abschätzung der Größe. Dies ist der Ansatzpunkt für die Bewertung des Stichprobenfehlers. Die Konvergenz des Wertes der Größe kann untersucht werden, indem der Mittelwert in Abhängigkeit der Anzahl der Messungen bestimmt und mit dem wahren Wert der Größe verglichen wird. Der wahre Wert wird dabei als der Mittelwert festgelegt, der sich bei einer genügend hohen Anzahl an Messungen ergibt. Die Abweichung zu diesem Wert ist dann der statistische Fehler im Ergebnis.

## 4.3 Iterativer Fehler

Der iterative Fehler tritt bei Iterationsverfahren auf, bei denen man sich einer Lösung schrittweise annähert. Iterative Methoden werden im wissenschaftlichen Rechnen oft für das Lösen von Gleichungssystemen angewendet. Die Idee bei diesen Verfahren besteht dabei, mit einem initialen Wert zu beginnen und die exakte Lösung schrittweise durch Verfeinerung des geschätzten Eingabewertes anzunähern. Um festzustellen, ob man sich der Lösung gut genug annähert, wird ein sogenanntes *Residuum* berechnet. In jedem Schritt wird wiederholt das gleiche Rechenverfahren angewendet. Die Ergebnisse eines Schrittes werden als Ausgangswerte des nächsten Schritts übernommen.
Soll zum Beispiel das lineare Gleichungssystem

$$Ax = b \tag{4.1}$$

gelöst werden, beginnt man mit $x^0$ als erste angenäherte Lösung für $x$. Davon ausgehend wird eine weitere Folge von Lösungen $x^1, x^2, \ldots$ erzeugt. Für eine weitere Näherungslösung $x^{i+1}$, wird eine festgelegte Iterationsvorschrift $\Phi$ auf die aktuelle Lösung $x^i$ angewendet. Dies wird solange wiederholt, bis das Residuum $r$ hinreichend klein wird. Mit $r = Ax - b$

wird festgestellt, ob man mit der aktuellen Näherungslösung die rechte Seite des linearen Gleichungssystems hinreichend annähert.
Die Differenz zwischen der approximierten Lösung der aktuellen Iteration und der exakten Lösung der diskreten Gleichungen ist der iterative Fehler [OR10]:

$$\epsilon_h^k = f_h^k - f_h \tag{4.2}$$

Der Parameter $h$ bezeichnet die diskreten Parameter. $f_h^k$ ist die Lösung der diskreten Gleichung in der Iteration $k$ und $f_h$ ist die exakte Lösung der diskreten Gleichung.
Es existieren zwei grundlegende Verfahren zur Abschätzung des iterativen Fehlers.

### 4.3.1 Globale Konvergenz - Die Machine-Zero-Methode

Da der wahre Wert von $f_h$ meist nicht bekannt ist, kann Gleichung 4.2 folgendermaßen approximiert werden

$$\epsilon_h^k \approx f_h^k - f_h^{k\to\infty} \tag{4.3}$$

wobei $f_h^{k\to\infty}$ die Lösung beschreibt, die sich ergibt, wenn die Anzahl der Iterationen unendlich hoch wird. Dies ist der Punkt, an dem sich das Residuum nicht länger reduziert, aber zufällig um einen bestimmten Wert oszilliert. Bis dieser Punkt auftritt, müssen jedoch viele Iterationen durchgeführt werden. Der Rechenaufwand wird dabei sehr hoch, besonders wenn die Lösung sehr langsam konvergiert. Deshalb wird diese Methode nicht empfohlen [OR10].

### 4.3.2 Lokale Konvergenz

Da sich die Machine-Zero-Methode sich aufgrund ihres hohen Rechenaufwands nicht eignet, gibt es ein weiteres Verfahren, welches die lokale Konvergenz der Näherungslösung betrachtet. Dabei werden nicht die Lösungen aller Iterationen verwendet, sondern die Lösungen der aktuellen Iteration $f_h^k$ und ihrer benachbarten Iterationen $f_h^{k-1}$ und $f_h^{k+1}$. Diese Methode eignet sich besonders für Anwendungen, bei denen die Lösung langsam konvergiert.

## 4.4 Diskretisierungsfehler

Durch die Diskretisierung des mathematischen Modells entstehen Diskretisierungsfehler (Abschnitt 3.1.1). Der Diskretisierungsfehler stellt von den bisher vorgestellten Fehlerkategorien den größten Fehler dar. Für die Bestimmung dieser Fehlerart ist die Voraussetzung, dass die exakte Lösung zum mathematischen Modell bekannt ist, um den Fehler genau zu bestimmen. Schwieriger wird es jedoch, und das ist in den meisten Anwendungen im wissenschaftlichen Rechnen der Fall, wenn die exakte Lösung des mathematischen Modells nicht vorhanden ist.
Für die Abschätzung des Diskretisierungsfehlers unterscheidet man zwischen *a priori* und *a posteriori* Methoden [OR10]. Bei a-priori-Methoden schätzt man eine Schranke für den

Diskretisierungsfehler ab bevor überhaupt ein Simulationsergebnis vorliegt. A-posteriori-Methoden hingegen schätzen den Fehler erst ab, wenn bereits eine numerische Lösung vorliegt. Hier wird die Lösung des mathematischen Modells mit einer höheren Genauigkeit als die berechnete Lösung abgeschätzt. Es existieren zur Zeit zwei Klassen von Verfahren zur Abschätzung dieses Fehlertyps [OR10]. Die eine Klasse benutzt ausschließlich die Informationen über die berechnete diskrete Lösung, die andere Klasse benutzt zusätzlich Informationen über das Problem selbst, welches im mathematischen Modell beschrieben wird.

# 5 Validierung

Bis jetzt wurde in der Verifikation ein Vergleich zwischen dem Computermodell und dem mathematischen Modell durchgeführt. Mit den Techniken der Verifikation konnte man feststellen, dass das Computermodell eine gute Repräsentation des mathematischen Modells ist. Das bedeutet, dass Implementierungsfehler weitgehend behoben wurden (Codeverifikation) und dass Fehler im Ergebnis der Simulation abgeschätzt und eventuell reduziert wurden (Lösungsverifikation).
Nun folgt der Vergleich des Computermodells mit der Realität. Die zentrale Fragestellung der Validierung ist, ob das Computermodell für einen bestimmten Zweck genau genug ist. Dies ist die *Modellgenauigkeit*. Dazu gehört auch die Überprüfung, ob das Modell die Realität beziehungsweise einen Ausschnitt aus der Realität gut simulieren kann.
In den folgenden Abschnitten wird beschrieben, welche Aufgaben die Validierung im wissenschaftlichen Rechnen hat. Es gibt drei Aspekte der Validierung. Jeder Aspekt beschreibt eine eigene Sichtweise auf die Validierung.

## 5.1 Drei Aspekte der Validierung

Im Wesentlichen lässt sich die Validierung in drei Bereiche aufteilen. Jeder Bereich beschreibt einen eigenen Aspekt und eine bestimmte Sichtweise auf die Validierung. Dies ist in Abbildung 5.1 dargestellt. Jeder Bereich ist von den anderen durch einen Kasten getrennt, welcher die Abgeschlossenheit jedes Bereichs hervorhebt. Die einzige Wechselwirkung zu den anderen Bereichen besteht durch die sequentielle Abfolge der Bereiche nacheinander. So wird zuerst die Genauigkeit des Modells ermittelt (Aspekt 1), bevor die Vorhersage-Fähigkeit evaluiert (Aspekt 2) und dann die Modellgenauigkeit mit den Anforderungen an die Genauigkeit überprüft wird (Aspekt 3). Im Folgenden werden die drei Aspekte genauer beschrieben.

### 5.1.1 Aspekt 1: Modellgenauigkeit

Es wird eine bestimmte Quantität untersucht. Diese wird zum einem mit dem verifizierten Computermodell in einer Simulation berechnet - die System Response Quantity - und zum anderem in einem Experiment mit mehreren Messungen bestimmt. Der wichtige Teil der Validierung besteht darin, die Ergebnisse von Simulation und Experiment miteinander zu vergleichen. Dazu wird ein mathematischer Operator definiert, die sogenannte *Validierungsmetrik*, welche die Differenz zwischen den beiden Ergebnissen bestimmt.
Für die Validierung werden sogenannte *Validierungsexperimente* entwickelt, um unter

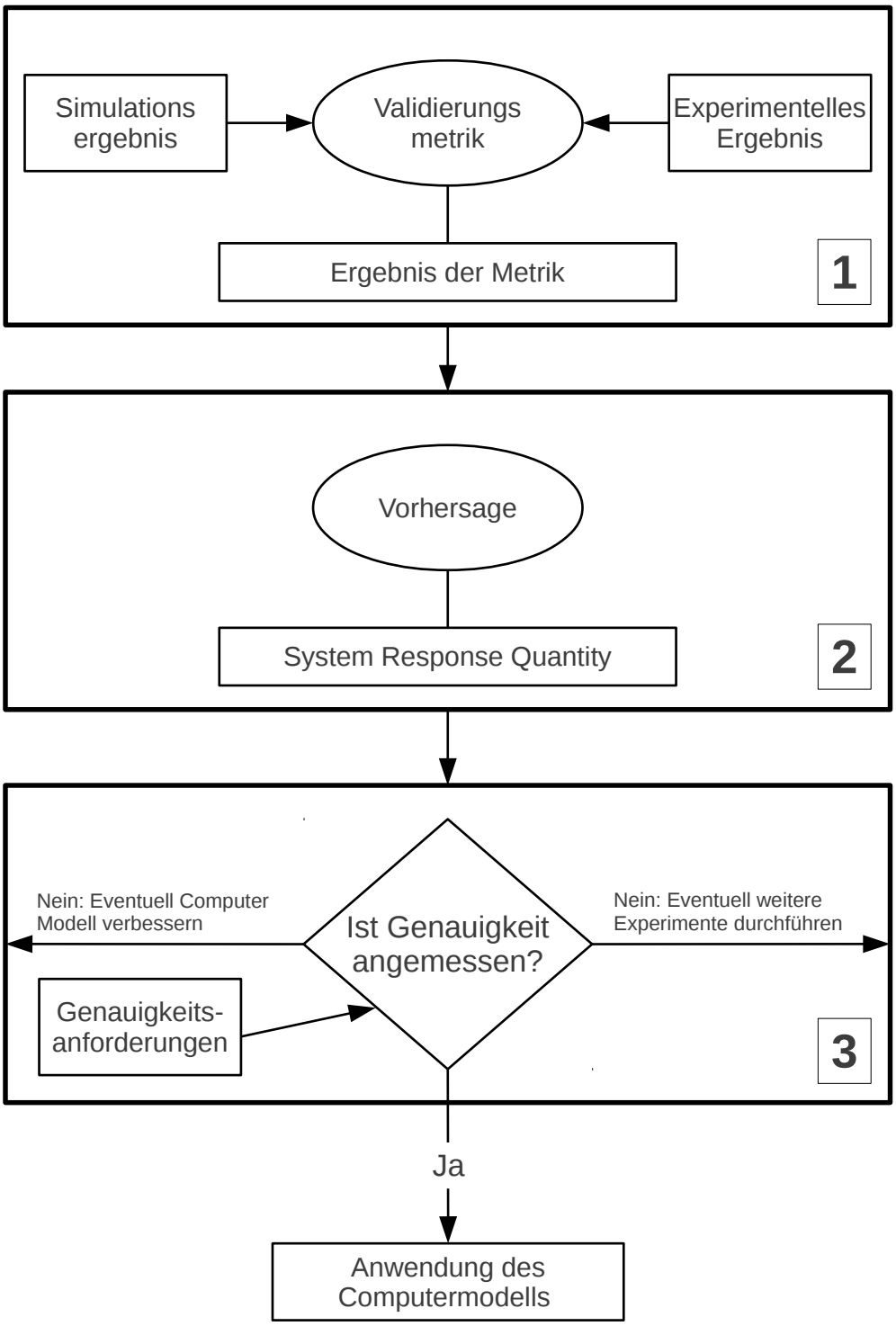



**Abbildung 5.1:** Die drei Aspekte der Validierung [OR10].

kontrollierten Bedingungen den Vergleich zwischen Computermodell und Experiment durchführen zu können.

### 5.1.2 Aspekt 2: Vorhersage-Fähigkeit des Modells

Eine weitere wichtige Eigenschaft, die evaluiert werden muss, ist die Vorhersage-Fähigkeit des (Computer-)Modells. Die Vorhersage ist die Anwendung des Modells unter Bedingungen, für die das Modell bisher nicht validiert wurde. Die Ergebnisse aus dem Validierungsprozess von Aspekt 1 dienen als reproduzierbare Sicherheit dafür, dass das Modell für bestimmte, validierte Eingaben und Bedingungen eine hinreichend hohe Stufe an Genauigkeiten erreicht. Mit diesem Wissen kann man die Vorhersagefähigkeit des Modells testen. Dabei nimmt man an, dass das Modell mit ähnlichen Eingaben und Bedingungen eine ähnlich hohe Genauigkeit liefern sollte. In diesem Aspekt entfällt der Vergleich mit den Ergebnisses des Validierungsexperiments.

### 5.1.3 Aspekt 3: Modellgenauigkeit bezüglich der Genauigkeitsanforderungen

In Aspekt 1 wurde die Modellgenauigkeit anhand eines Vergleichs mit Validierungsexperimenten bestimmt. Aspekt 3 beschäftigt sich mit dem Vergleich der Modellgenauigkeit mit den Genauigkeitsanforderungen, die die Anwendung an das Modell stellt. Insbesondere werden die Ergebnisse überprüft, die nicht mit Validierungsexperimenten validiert wurden. Dazu gehört die Entscheidung, ob das Modell für seinen bestimmten Zweck genau genug ist. Normalerweise hängt die Angemessenheit des Modells für eine Anwendung von vielen weiteren Faktoren ab, wie zum Beispiel der Performanz oder auch dem Ressourcenverbrauch während der Simulation (beispielweise Speicherverbrauch). Dies ist jedoch nicht die Aufgabe der Validierung aus der Sicht des wissenschaftlichen Rechnens.

## 5.2 Zusammenführung der drei Aspekte

In [OR10] wird eine eingeschränkte Sicht der Validierung vorgestellt, die alle Aspekte in einer Perspektive zusammenfasst. Diese Perspektive ist in Abbildung 5.2 dargestellt. Zuerst erfolgt der Vergleich der Ergebnisse von Simulation und Experiment mittels einer ausgewählten Validierungsmetrik. Das Ergebnis der Validierungsmetrik wird dann ausgewertet und darauf untersucht, ob die Differenz zwischen dem Ergebnis der Simulation und dem Ergebnis des Experiments angemessen ist. Die Angemessenheit der Abweichungen orientiert sich an den Genauigkeitsanforderungen, die an das Ergebnis der Validierungsmetrik gestellt wurden.
Entspricht das Ergebnis nicht den Anforderungen, werden die Parameter des Computermodells und des Experiments kalibriert. Dieser Prozess wird solange wiederholt, bis die Abweichung den Anforderungen entspricht.
Im Abschnitt zu Aspekt 1 wurden die Begriffe Validierungsmetrik und Validierungsexperiment erwähnt. In den letzten beiden Abschnitten dieses Kapitels wird beschrieben, inwiefern sich Validierungsexperimente von klassischen Experimenten unterscheiden und

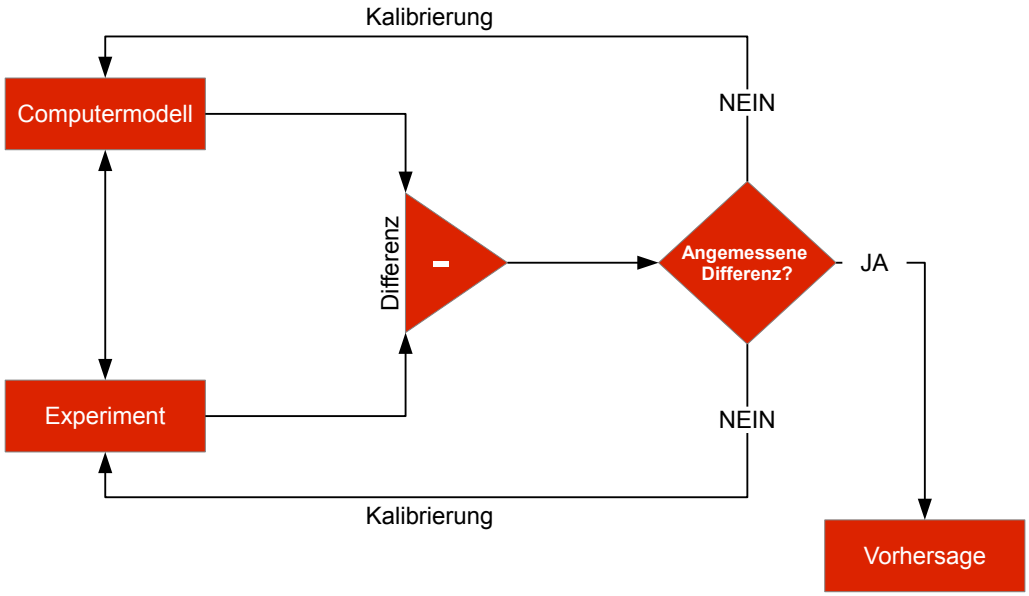



**Abbildung 5.2:** Die Validierung und ihre Komponenten (Nachzeichnung aus [OR10])

wie Validierungsmetriken für den Vergleich von Simulationsergebnis und experimentellem Ergebnis definiert werden können.

### 5.2.1 Validierungsexperimente

Validierungsexperimente unterscheiden sich grundlegend von Experimenten, die nicht für die Validierung von Computermodellen genutzt werden. Diese Experimente werden im Weiteren *traditionelle* Experimente genannt [OR10]. Traditionelle Experimente dienen dazu ein grundlegendes Verständnis der zugrundeliegenden Prozesse zu entwickeln oder um ein mathematisches Modell eines bereits verstandenen physikalischen Prozesses durch Parameterkalibrierung zu verbessern. Eine weitere Klasse von Experimenten überprüft die Sicherheit und Zuverlässigkeit von Komponenten oder ganzen Systemen.
Validierungsexperimente bilden eine neue Klasse von Experimenten. Ein Validierungsexperiment wird für den Zweck entwickelt, um die Fähigkeit des Computermodells (und auch des mathematischen Modells) einen bestimmten physikalischen Prozess zu simulieren, quantitativ zu bewerten. Ein Validierungsexperiment besteht meistens aus einer Folge von mehreren Experimenten. Die Ausführung von Validierungsexperimenten geschieht unter kontrollierten Bedingungen. Man spricht von der *Charakterisierung* des

Experiments [OR10]. Dabei sind alle Eigenschaften des Experiments gemeint, die für die Simulation benötigt werden, um dessen Eingaben und Initial -und Randbedingungen genau genug beschreiben zu können.
Es stellt sich natürlich die Frage, warum nicht einfach Ergebnisse von bereits durchgeführten Experimenten zur Validierung genutzt werden. Der signifikante Grund ist die oft nicht ausreichende Dokumentation über die Eingaben, die Initial -und Randbedingungen und die Umgebung des Experiments. Diese Daten werden für die Ausführung der Simulation jedoch benötigt. Oft reicht die Menge an experimentellen Daten, die in einem traditionellen Experiment gesammelt wurden, für eine Validierung nicht aus, um die Genauigkeit des Modells zu bewerten. Des Weiteren ist meistens nicht bekannt, inwiefern experimentelle Unsicherheiten das Ergebnis der untersuchten Quantität beeinflusst haben. Auch wenn Abschätzungen über die Unsicherheiten gemacht wurden, sind diese im Vergleich zur tatsächlichen Unsicherheit oft unterbewertet [OR10].
Es existieren Richtlinien für den Entwurf von Validierungsexperimenten. Diese werden im folgenden Abschnitt vorgestellt und beschrieben.

#### Richtlinien für den Entwurf von Validierungsexperimenten

Insgesamt existieren fünf grundlegende Richtlinien.

**Zusammenarbeit** Ein Validierungsexperiment sollte bevorzugt in einer Zusammenarbeit zwischen Modellentwicklern, Experiment-Entwicklern, Programmierern und den Benutzern des Modells entworfen werden.

**Erfassen der nötigen Eingaben** Ein Validierungsexperiment sollte so entworfen werden, dass alle physikalischen Quantitäten, die von Interesse sind, sowie Initial -und Randbedingungen erfasst und eingebracht werden.

**(Un-) Abhängigkeit** Auch wenn eine Zusammenarbeit zwischen beiden Seiten gewünscht ist, sollten Experiment und Simulation unabhängig voneinander durchgeführt werden. Ist beispielsweise das Ergebnis des Experiments schon vor der Simulation bekannt, könnte dies dessen Ausführung beeinflussen, indem vor der Simulation schon Parameter angepasst werden. Die Bedingungen unter denen die Simulation ausgeführt werden sind dann dementsprechend andere. Ein quantitativer Vergleich zwischen den Ergebnissen von Experiment und Simulation ist nicht mehr möglich.

**Hierarchie** Validierungsexperimente werden in einer Hierarchie organisiert. Dabei stellt die oberste Ebene das Gesamtsystem dar. Steigt man in der Hierarchie ab, so werden die Komponenten weiter in kleinere Komponenten aufgeteilt. Diese sind im einzelnen einfacher zu validieren als das Gesamtsystem.

**Unsicherheiten** In einem Experiment entstehen unvermeidlich zufällige und systematische Fehler. Diese müssen vor dem Vergleich mit dem Simulationsergebnis abgeschätzt werden. In der Validierung wird die Genauigkeit des mathematischen Modells bestimmt, weswegen eine Abschätzung der Unsicherheiten im experimentellen Ergebnis äußerst wichtig ist.

### 5.2.2 Validierungsmetrik

Das Ziel von Validierungsmetriken ist es, das Ergebnis der Simulation und des Experiments quantitativ miteinander zu vergleichen. Sie ist ein Abstandsmaß und gibt die Differenz zwischen zwei Ergebnissen an.
Eine Metrik erfüllt folgende Bedingungen:
Die erste Bedingung in Gleichung 5.1 gibt an, dass die Metrik stets positiv oder gleich null ist. Sie ist genau dann null, wenn die Ergebnisse gleich sind, also $x = y$ ist.

$$d(x, y) \geq 0 \tag{5.1}$$

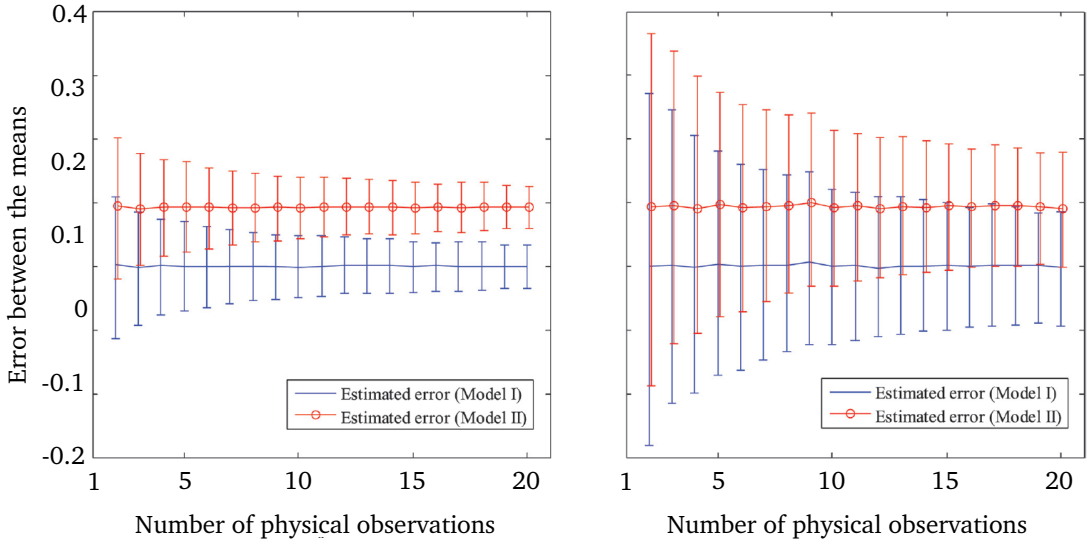
Die zweite Bedingung ist die Symmetrieeigenschaft (Gleichung 5.2). Dies bedeutet, dass der Abstand zwischen zwei Ergebnissen unabhängig von der Richtung ist.

$$d(x, y) = d(y, x) \tag{5.2}$$

Eine Metrik erfüllt stets die Dreiecksungleichung (Gleichung 5.3).

$$d(x, y) + d(y, z) \geq d(x, z) \tag{5.3}$$

Zusätzlich zu diesen Eigenschaften muss eine Validierungsmetrik weitere Anforderungen erfüllen. So muss die Validierungsmetrik ein objektives Maß sein. Dies bedeutet, dass die Metrik stets die gleiche Bewertung für die gleichen vorliegenden Datensätze liefert unabhängig von möglichen Voreingenommenheit der Person, die das Modell analysiert. Zusätzlich müssen Unsicherheiten in den Ergebnissen der Simulation und des Experiments mit beachtet werden. Eine Validierungsmetrik sollte zwischen Modellen mit kleiner und großer Unsicherheit unterscheiden können. Der Wert der Metrik sollte sich nicht verbessern, wenn sich die Breite der Verteilung eines Modellparameters vergrößert.
Abbildung 5.3 zeigt als ein Beispiel einer Validierungsmetrik die sogenannte *Frequentist's Metric*. Sie vergleicht den Mittelwert der Ergebnisse des Experiments mit dem Mittelwert der Ergebnisse der Simulation. Es werden für die Darstellung des Ergebnisses dieser Validierungsmetrik künstlich Messdaten nach einer mathematischen Vorschrift mit einem festen Modellparameter $\Theta$ und einer Messunsicherheit $\sigma_\epsilon$ erzeugt. Entsprechend diesen Messdaten werden Modelle definiert, die der gleichen Vorschrift folgen, jedoch mit einem anderen Modellparameter und unterschiedlich großer Messunsicherheit. In der Abbildung wurde beispielsweise im linken Diagramm eine kleine Messunsicherheit $\sigma_\epsilon = 0.08$ und im rechten Diagramm eine größere Messunsicherheit $\sigma_\epsilon = 0.2$ gewählt. Die Modellparameter $\Theta$ wurden so variiert, dass das Modell zum einem genau den Messwerten entspricht (Modell 1 mit $\Theta_{Modell_1} = \Theta$) und zum anderen das falsche Modell (Modell 2 mit $\Theta_{Modell_2} \neq \Theta$) zu den Messwerten darstellt. Die Validierungsmetrik sollte diese Abweichungen erkennen können.
Dies lässt sich aus den Diagrammen ablesen. Die $x$-Achse der Diagramme zeigt die Anzahl der Messungen, während die $y$-Achse die Abweichung der Mittelwerte abbildet. Im linken Diagramm der Abbildung entspricht die blaue Kurve das Ergebnis der Validierungsmetrik




(a) $\sigma_\epsilon = 0.08$ (b) $\sigma_\epsilon = 0.2$

**Abbildung 5.3:** Ergebnis der Validierungsmetrik *Frequentist's Metric* [LCAH11].

beim Vergleich des richtigen Modells mit den Messwerten. Die Kurve verläuft etwa bei Wert 0, das heißt es besteht keine Abweichung zwischen den Mittelwerten der Messung und der Ergebnisse des Modells, was zu erwarten war. Die rote Kurve hingegen entspricht dem Vergleich des zweiten, falschen Modells und man erkennt, dass die Abweichung der Mittelwerte größer ist.
Im rechten Diagramm lassen sich ähnliche Ergebnisse beobachten, nur mit größerer Messunsicherheit. Man erkennt gut, dass die größere Messunsicherheit nicht das Ergebnis der Validierungsmetrik beeinflusst. Das richtige Modell mit dem gleichen Modellparameter wie die Messwerte wird immer noch als richtiges Modell erkannt.

# 6 Zusammenfassung und Fazit

Diese Seminararbeit beschäftigte sich mit Verifikation und Validierung im wissenschaftlichen Rechnen. Dabei wurden zunächst die Grundlagen und Begriffe der Modellierung und Simulation, die im weiteren Verlauf der Seminararbeit wichtig waren, in der Einleitung motiviert und erläutert. Die Verifikation wird vor der Validierung durchgeführt. An diesem Ablauf orientierte sich der Aufbau der Seminararbeit. So wurde zunächst die Verifikation mit ihren Teilbereichen Codeverifikation und die Lösungsverifikation mit ihren Aufgaben und Verfahren beschrieben. Die Verifikation vergleicht das Computermodell mit dem mathematischen Modell und überprüft, ob es eine gute Repräsentation des mathematischen Modells ist. Dies umfasst die Nachprüfung der Korrektheit des Codes und die Evaluierung der Genauigkeit der Ergebnisse, die das Computermodell liefert. Für den Nachweis der Korrektheit des Computercodes ist der Grad der Genauigkeit ein strenges und wichtiges Codeverifikationskriterium. Dieser wird in theoretischer und empirischer Weise bestimmt. Die Grundlage für diese Kriterien bilden die Analyse des Abbruchfehlers und des Diskretisierungsfehlers. Für beide ist es wichtig, dass die exakte Lösung des mathematischen Modells zum Vergleich mit dem nach dem Computermodell berechneten Ergebnis vorhanden ist.
Die Lösungsverifikation beschäftigt sich mit der Abschätzung von Fehlern in dem Simulationsergebnis. Dabei gibt es vier wesentliche Fehlertypen, die abgeschätzt werden müssen. Dies sind der Rundungsfehler und der Diskretisierungfehler, die in jedem Ergebnis einer diskreten Computersimulation auftreten. Die anwendungsabhängigen Fehlertypen sind der Stichprobenfehler und der iterative Fehler. Die Abschätzung der Fehler ist wichtig, um Sicherheit über die Vorhersagefähigkeit des Modells zu erhalten. Die Fehler können bei der Lösungsverifikation nur abgeschätzt werden, da die exakte Lösung des mathematischen Modells nicht vorhanden ist. Das macht vor allem die Evaluation des Diskretisierungsfehlers schwierig.

Zuletzt folgte die Validierung. Die Validierung lässt sich in drei unterschiedlichen Aspekten betrachten, die alle zu einem Aspekt zusammengebracht werden können. Es wird das Computermodell mit der Realität verglichen, indem die Ergebnisse des Computermodells mit den Ergebnissen der Validierungsexperimente verglichen werden. Validierungsexperimente unterscheiden sich grundlegend von traditionellen Experimenten, da sie unter kontrollierten Bedingungen stattfinden. Dies ist wichtig, um die gleichen Initial -und Randbedingungen für die Computersimulation anwenden zu können. Für einen quantitativen Vergleich der Ergebnisse sind Validierungsmetriken von zentraler Bedeutung. Diese müssen neben den grundlegenden Eigenschaften einer Metrik - wie Definitheit, Symmetrie und die Dreiecksungleichung - weitere Bedingungen erfüllen. Dies war zum

einen die Objektivität und zum anderen die Robustheit gegenüber unterschiedlich großen Unsicherheiten in den Messergebnissen. Eine bekannte Validierungsmetrik vergleicht die Mittelwerte der Simulationsergebnisse mit den Mittelwerte der experimentellen Ergebnisse.

In dieser Seminararbeit wurden die wichtigsten Begriffe und Zusammenhänge des komplexen Themenbereiches der Verifikation und Validierung im wissenschaftlichen Rechnen in einer kompakten Form erläutert.
Für ein weiteres Studium sind vor allem Details über konkrete Realisierung interessant. Erwähnt wurde beispielsweise die Verfeinerung des Gitters zur Bestimmung des empirischen Grades der Genauigkeit. Die Methoden für die Wahl einer feineren Gitterauflösung hängen davon ab, welche Diskretisierungsmethode gewählt wurde. Auch bei der Bestimmung des Abbruchfehlers spielt dies eine Rolle. Weiterhin interessant ist die technische Umsetzung der Abschätzung der vier Fehlertypen, die im Kapitel über die Lösungsverifikation vorgestellt wurden.
Ein weiterer Diskussionspunkt ist, wie Validierungsexperimente in der Praxis umgesetzt werden können. Besonders im Bereich der Erdsystemmodellierung sind reproduzierbare Simulationsergebnisse von Bedeutung. Deshalb benötigt man Validierungsexperimente, um die Vorhersagefähigkeit der Simulationsergebnisse zu überprüfen. Es stellt sich die Frage, welche weiteren Anforderungen - bezüglich des Entwurfs und der Ausführung - Validierungsexperimente für Erdsystemmodelle erfüllen müssen, neben den Anforderungen, die in Kapitel 5 aufgestellt wurden. Eine Diskussion über den Entwurf von Validierungsexperimenten für Erdsystemmodelle ist an dieser Stelle ein interessanter Ausblick.

# Literaturverzeichnis

[HS06] T. Huckle and S. Schneider. *Numerische Methoden: Eine Einführung für Informatiker, Naturwissenschaftler, Ingenieure und Mathematiker*. EXamen. press Series. Springer, 2006.

[LCAH11] Yu Liu, Wei Chen, Paul Arendt, and Hong-Zhong Huang. Toward a better understanding of model validation metrics. *Journal of Mechanical Design*, 133(7):071005, 2011.

[OR10] W.L. Oberkampf and C.J. Roy. *Verification and Validation in Scientific Computing*. Cambridge University Press, 2010.