

UNIVERSITÄT HAMBURG

Informatik Scientific Computing

Betreuer: Prof. Dr. Thomas Ludwig, Julian Kunkel,  
Nathanael Hübbe, Petra Nerge

# Modellierung und Simulation

## Übersicht

eingereicht von:

Student:	Holger Trampe
Matrikelnummer:	6218613
E-Mail:	htrampe@gmail.com

Hamburg, den 31.03.2013

# Inhalt

<b>1</b>	<b>MODELLIERUNG UND SIMULATION.....</b>	<b>1</b>
<b>2</b>	<b>TERMINOLOGIE.....</b>	<b>1</b>
2.1	SYSTEM.....	1
2.2	MODELL.....	3
2.3	SIMULATION.....	3
2.4	QUALITÄTSSICHERUNG.....	5
2.4.1	GENAUIGKEIT UND PRÄZISION.....	5
2.4.2	VERIFIKATION UND VALIDIERUNG.....	5
2.5	COMPUTERMODELL.....	7
<b>3</b>	<b>ABLAUF FÜR DIE MODELLERSTELLUNG.....</b>	<b>7</b>
3.1	EINFÜHRUNG.....	7
3.1.1	VORSTELLUNG DES BEGLEITENDEN PROJEKTES.....	8
3.2	ÜBERBLICK DER MODELLBILDUNG.....	9
3.2.1	KONZEPTIONELLE MODELLIERUNGSPHASE.....	9
3.2.2	MATHEMATISCHE MODELLIERUNGSPHASE.....	9
3.2.3	DISKRETISIERUNGSPHASE.....	10
3.2.4	IMPLEMENTIERUNGSPHASE.....	11
3.2.5	NUMERISCHE LÖSUNGSPHASE.....	13
3.2.6	PRÄSENTATION DER LÖSUNGEN.....	13
3.3	ABSCHLUSS DER MODELLBILDUNG.....	13
<b>4</b>	<b>VERSCHIEDENE SIMULATIONSPRINZIPIEN.....</b>	<b>14</b>
4.1	EINFÜHRUNG SIMULATIONSPRINZIPIEN.....	14
4.2	EREIGNISORIENTIERTE SIMULATION.....	14
4.3	TRANSAKTIONSORIENTIERTE SIMULATION .....	15
4.4	PROZESSORIENTIERTE SIMULATION.....	15
4.5	ZEITGESTEUERTE SIMULATION.....	15
4.6	STOCHASTISCHE SIMULATION.....	16
4.7	NUTZEN DER PRINZIPIEN.....	16
<b>5</b>	<b>VORTEILE UND GRENZEN.....</b>	<b>16</b>
<b>6</b>	<b>AUSBLICK.....</b>	<b>17</b>
<b>7</b>	<b>LITERATURVERZEICHNIS.....</b>	<b>18</b>

## Verzeichnis der Abbildungen

Abbildung 1: System.....	2
Abbildung 2: Übersicht Validierung und Verifikation.....	6
Abbildung 3: Gekoppelte Bewegungsgleichung.....	10
Abbildung 4: Geschwindigkeit und Position.....	10
Abbildung 5: Klassendiagramm der Berechnung.....	12
Abbildung 6: Programmcode zur Berechnung.....	12

## **Verzeichnis der Abkürzungen**

SC	Scientific Computing
VDI	Verein deutscher Ingenieure
NASA	National Aeronautics and Space Administration
DIN	Deutsches Institut für Normung

## 1 Modellierung und Simulation

Im Bereich „Scientific Computing“<sup>1</sup> ist die Bildung von Modellen und deren Simulation ein elementares Werkzeug, um wissenschaftliche Berechnungen mithilfe des Computers durchzuführen. Hierfür werden Modelle benötigt, die mit der Informatik erstellbar und anschließend in Simulationen eingesetzt werden können. Da die Begrifflichkeiten, z.B. „Modell“ und „Simulation“, in diversen anderen Fachgebieten eingesetzt werden, ist eine klare Abgrenzung und Definition derselben nötig, um Verwechslungen und Irrtümer zu vermeiden.

In dieser vorliegenden Arbeit sollen relevante Begriffe genannt und in den Kontext des SC gesetzt werden. Beispielhaft wird auch die Erstellung eines Modells erläutert und die Vorteile sowie Grenzen von Simulationen an Modellen beleuchtet.

## 2 Terminologie

Scientific Computing bewegt sich im Bereich der Computersimulation, daher werden im Folgenden die dafür benötigten Begriffe genannt und definiert.

### 2.1 System

Bevor ein Modell erstellt und an ihm Simulationen durchgeführt werden können, muss zunächst das System definiert werden, in dem sich das Modell befindet. Informatiker nutzen verschiedene Ansätze, um den Begriff „System“ genau zu bestimmen. Wolfgang P. Kowalk, Universität Oldenburg, definiert den Begriff wie folgt:

Ein System (system) ist eine räumlich abgeschlossene, logisch zusammengehörende und zeitlich begrenzte Einheit, die voneinander abhängende Komponenten umfasst<sup>2</sup>.

Diese Definition grenzt ein System jedoch stark ein, was im Bereich SC nicht von Vorteil ist, da auch abstrakte, nicht in der Realität messbare Gegebenheiten modelliert und simuliert werden können. Aufgrund dessen ist die Definition nach Oberkampf und Roy besser geeignet:

System: a set of physical entities that interact and are observable, where the entities can be a specified quantity of matter or a volume in space<sup>3</sup>

---

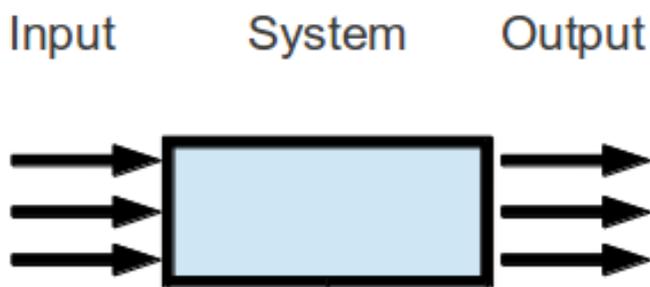
<sup>1</sup> Engl. Wissenschaftliches Rechnen mit Computern

<sup>2</sup> Vgl. Kowalk 1996, Grundbegriffe zur Modellierung

<sup>3</sup> Vgl. Oberkampf/Roy 2010, S. 84

*Eine Reihe von physisch vorhandenen Dingen, die miteinander interagieren und beobachtbar sind. Sie können eine Menge von Teilchen sein oder ganze Körper im Raum.*

Ein System wird in weitere drei Komponenten unterteilt (Vgl. Abb. 1). Dem Input<sup>4</sup>, welcher alle Werte der im System vorkommenden Variablen beeinflusst, dem System selbst, welches die vom Input erhaltenen Daten verarbeitet und dem Output<sup>5</sup>, was letztlich das Ergebnis der Systemverarbeitung darstellt.



*Abbildung 1: System*

Als Beispiel für ein System beschreiben Oberkampf und Roy ein Raumschiff im Weltall, welches Masse und Geschwindigkeit besitzt. Hier wird hervorgehoben, dass äußere Einflüsse nicht Teil des System sind, es aber beeinflussen. Im Bezug auf das Beispiel sind das die Anziehungskräfte der umliegenden Himmelskörper und Sonnenwinde.

Nun werden dem System Daten übermittelt, welche sich hier auf eine Positionsveränderung des Schiffes beziehen. Eingegeben werden eine neue Geschwindigkeit und eine Richtung. Das System verarbeitet nun diese Daten, indem es die Triebwerke startet, bis die neue Geschwindigkeit erreicht ist und überträgt die Richtungsinformationen auf die Steuerungselemente. Unter Berücksichtigung der äußeren Einflüsse gibt nun die Datenausgabe, der Output, Aufschluss darüber, ob die eingegebenen Daten das System korrekt bewegt haben<sup>6</sup>.

---

<sup>4</sup> Engl. Eingabe

<sup>5</sup> Engl. Ausgabe

<sup>6</sup> Vgl. Oberkampf/Roy 2010, S. 85

## 2.2 Modell

Soll ein System, ob real oder abstrakt, in die Informatik überführt werden, muss dieses zunächst in einem Modell abgebildet sein.

Dem geht die Modellbildung voran, welche in Abschnitt 3 „Ablauf für die Modellerstellung“ näher erläutert wird. Kowalk definiert den Begriff des „Modells“ wie folgt:

Ein Modell (model) ist eine Zusammenfassung von Merkmalen eines realen (oder empirischen) künstlichen Systems sowie eine Festlegung der Beziehung zwischen diesen Merkmalen; da ein Modell niemals alle Merkmale eines System umfassen kann, ist ein Modell eine Abstraktion eines realen Systems<sup>7</sup>.

Wie bereits bei der Definition zum Begriff „System“ sind Kowalks Ausführungen zielgerichtet auf die Informatik und Programmierung ausgelegt. Nicht alle Eigenschaften eines System können in einem Modell abgebildet werden, nur die für die Simulation elementaren Bestandteile sind so exakt wie möglich zu modellieren. Hier ist zu nennen, dass das Modell immer auf den Zweck und Umfang der Simulation ausgelegt werden muss. Dies wird im Punkt 2.3 näher beleuchtet. Eine weitere Definition wird von Stachowiak gegeben:

Ein Modell ist ein beschränktes Abbild der Wirklichkeit<sup>8</sup>.

Auch in dieser Definition ist durch das Adjektiv „beschränkt“ der Kern eines jeden Modells beschrieben, aber einfacher und allgemeiner formuliert als durch Kowalk. Daher ist diese Definition nach Stachowiak jene, welche im Bereich „Modellierung und Simulation“ zum Einsatz kommt.

## 2.3 Simulation

Mit dem bestehenden Modell kann nun eine Simulation durchgeführt werden. Das Modell ist dabei bereits auf den Umfang und das Ziel der Simulation ausgerichtet. Ein einfaches Beispiel soll diese Herangehensweise erläutern.

In einem Windkanal sollen die aerodynamischen Eigenschaften eines Autos bewertet werden. Hierzu wird ein Modell der Außenhülle angefertigt, welches so exakt wie möglich nachgebildet wird. Sprich, jede Verzweigung, Kante und Rundung wird später im Modell zu sehen sein. Irrelevant sind jedoch das Gewicht oder die Bereifung des Modells. Mit diesem „Abbild der Wirklichkeit“<sup>9</sup> kann das angestrebte Ziel erreicht werden, dass Gefährt bezüglich Wind so gut wie möglich zu entwickeln.

---

<sup>7</sup> Vgl. Kowalk 1996, Grundbegriffe zur Modellierung

<sup>8</sup> Vgl. Stachowiak 1973, S. 131-133

<sup>9</sup> Vgl. Def. Modell nach Stachowiak, 2.2

Andere Dinge, die während der Entwicklung interessant sind (z.B. die Steifigkeit bei einem Seitenaufprall) bedürfen anderer Modelle und Simulationen.

Definiert wird eine Simulation nach dem VDI, da in den meisten Entwicklungsbereichen Simulationen nötig sind. Definition nach VDI 3633:

Simulation ist das Nachbilden eines Systems mit seinen Prozessen in einem experimentierbaren Modell, um zu Erkenntnissen zu gelangen, die auf die Wirklichkeit übertragbar sind. Insbesondere werden die Prozesse über die Zeit entwickelt<sup>10</sup>.

Hier tauchen nun die Begriffe „Modell“ und „System“ wieder auf, welche deutlich machen, wie diese im Zusammenhang stehen. Lediglich die Aussage, „...Prozesse über die Zeit entwickeln“ deutet auf eine zeitbasierte, dynamische Simulation hin. Denkbar sind aber auch zeitunabhängige, statische Simulationen. Daher wird eine weitere Definition nach Neelamkavil eingeführt, welche für den Bereich SC angewendet werden soll:

Simulation: the exercise or use of a model to produce a result<sup>11</sup>

*Simulation: Die Ausübung oder Verwendung eines Modells, um ein Ergebnis zu erzielen.*

Im Bereich SC wird der Begriff der Simulation mit dem Computer verbunden, und generiert den Bereich „Computersimulation“, was die Durchführung einer Simulation an einem Computermodell beschreibt. Da sich diese Aussage jedoch mit der Definition von Neelamkavil deckt und nur das Mittel zum Zweck genauer beschrieben wird, ist eine weitere Definition hier nicht genannt.

Unterschieden werden Simulationen nicht nur zwischen dynamisch und statisch, auch das Verhalten kann durch zufällige oder vorgegebene Werte beeinflusst werden. Daher wird hier zwischen den stochastischen, also Simulationen mit Zufallszahlen, und den deterministischen, dem Ausschließen von jeglichen Zufallsergebnissen, differenziert. Stochastische Simulationen können näherungsweise Werte wie die Zahl Pi approximieren, z.B. mit der Monte-Carlo-Simulation<sup>12</sup>, deterministische verlangen genau berechnete Ergebnisse ohne Zufallskomponente.

---

<sup>10</sup> Vgl. VDI Richtlinien 3633-1

<sup>11</sup> Oberkampf/Roy 2010, S. 92

<sup>12</sup> Simulation, bei der häufig durchgeführte Zufallsexperimente die Basis bilden

## 2.4 Qualitätssicherung

Da die Ergebnisse einer Simulation auf die Realität übertragen werden sollen, ist die Qualität derselben von großer Bedeutung. Daher hat die DIN eine Norm veröffentlicht, die diesen Begriff genau definiert:

Qualitätssicherung ist jede geplante und systematische Tätigkeit, die innerhalb des Systems verwirklicht und dargelegt wird, um Vertrauen dahingehend zu schaffen, dass eine Einheit die Qualitätsforderung erfüllen wird.<sup>13</sup>

Das System soll so exakt wie möglich im Modell abgebildet werden, um die Ziele der Simulation zu erreichen. Sind die Anforderungen erfüllt, können die Ergebnisse zur Entscheidungshilfe über komplexe Vorgänge herangezogen werden. Die Qualitätssicherung wird in weitere Teilbereiche unterschieden, die im Folgenden erläutert werden.

### 2.4.1 Genauigkeit und Präzision

Die Präzision wird wie folgt definiert:

Präzision versteht sich die Abweichung des Wertes aus der wiederholten Messung mit gleichen Parametern<sup>14</sup>.

Der Begriff Genauigkeit ist definiert als:

Genauigkeit ist die Abweichung der realen Messwerte<sup>15</sup>.

Beide Begriffe grenzen die Qualität der Simulation hinsichtlich der erhaltenen Ergebnisse ein. Präzision setzt hierbei voraus, dass die Messung mit dem gleichen Messgerät durchgeführt wird und beschreibt die Abweichung dieser Messwerte voneinander. Je geringer diese ist, desto präziser ist der Wert. Hingegen ist die Genauigkeit des, soweit vorhandenen, in der Realität gemessenen Wertes und deren Abweichung.

### 2.4.2 Verifikation und Validierung

Die beiden Begriffe Verifikation und Validierung bilden einen komplexen, für sich eigenständigen Bereich ab, der die Daten des Modells auf unterschiedliche Art und Weise prüft.

---

<sup>13</sup> Vgl. DIN EN ISO 8402, Ziffer 3..5

<sup>14</sup> Vgl. <http://de.wikipedia.org/wiki/Pr%C3%A4zision>, 09.01.2013; 11:26

<sup>15</sup> Vgl. <http://de.wikipedia.org/wiki/Pr%C3%A4zision>, 09.01.2013; 11:27

Für die Übersicht werden beide wie folgt definiert:

Verification: substantiation that a computerized model represents a conceptual model within specified limits of accuracy<sup>16</sup>

*Computergestütztes Modell bewegt sich innerhalb der durch das konzeptionelle Modell vorgegebene Grenzen.*

Validierung: substantiation that a computerized model within its domain of applicability possesses a satisfactory range of accuracy consistent with the intended application of the model<sup>17</sup>

*Computergestütztes Modell bewegt sich im Grad der Genauigkeit innerhalb der mit dem Modell abgezielten Anwendung.*

Die Verifikation prüft nach dieser Definition, ob das System korrekt abgeleitet wurde und die Konzepte des konzeptionellen und computergestütztem Modell decken. Ob die Anforderungen in der Praxis erfüllt wurden, wird anhand der Validierung der Werte kontrolliert. Hier werden die Lösungen des computergestützten Modells mit den Messwerten aus der Realität verglichen. Stimmen diese überein, ist das Modell valide.

Um den Prozess und den Unterschied zwischen den Begriffen deutlicher zu machen, ist Abb. 2 hinzuzuziehen.

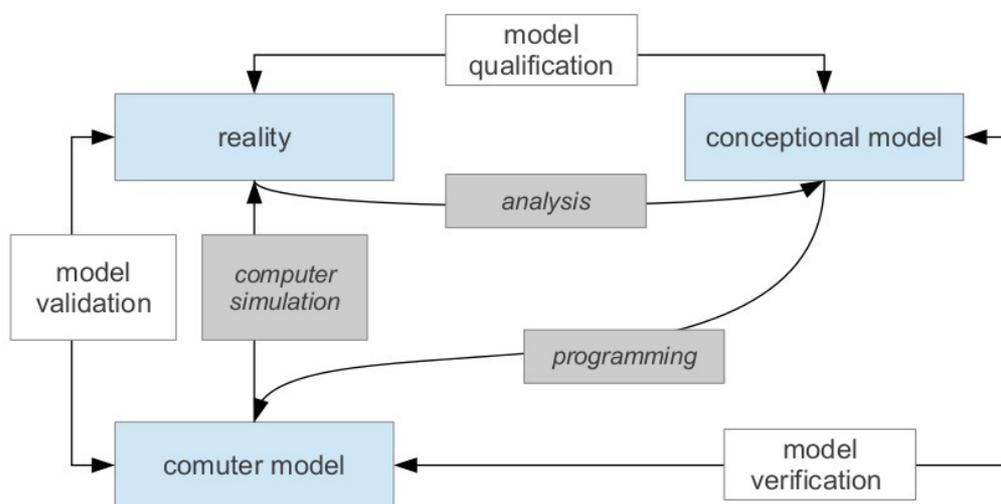


Abbildung 2: Übersicht Validierung und Verifikation

Von der Realität wird durch Analyse ein konzeptionelles Modell erzeugt (conceptual model). Dieses wird anschließend in ein Computermodell überführt (computer model). Diese beiden Modelle werden verifiziert.

<sup>16</sup> Vgl. Oberkampf/Roy 2010, S. 22

<sup>17</sup> Vgl. Oberkampf/Roy 2010, S. 22

Erfüllt das Computermodell die Anforderungen, kann die Simulation der Realität durchgeführt werden (computer simulation). Die hier entstehenden Daten werden mit den Messwerten aus der Realität validiert. Stimmen sie überein, ist das Computermodell valide.

Sollte die Grundlage der Simulation in der Realität nicht vorkommen, müssen Alternativen zur Validierung gefunden werden. Die Verifikation kann dennoch Ergebnisse liefern, da sie sich auf der Beziehung zwischen dem konzeptionellen und computergestütztem Modell aufbaut.

## 2.5 Computermodell

Da im Bereich SC meist Computerprogramme eingesetzt werden, ist der Begriff des Computermodells ebenfalls einzuführen:

Ein Computermodell repräsentiert die reale Welt in einem Computerprogramm. Genutzt wird das Computermodell, um das konzeptionelle Modell in ein diskretes Computermodell zu überführen. Die Arbeitsweise eines jeden Computers ist diskret, wodurch Fehler vermieden werden. Ein Computer kann nur vollständig lösbar Algorithmen berechnen, die nach einem festgelegten Ablauf das System widerspiegeln.

## 3 Ablauf für die Modellerstellung

### 3.1 Einführung

Nach Stachowiak ist ein Modell „...ein beschränktes Abbild der Wirklichkeit“<sup>18</sup> und besitzt somit nur einen Teil der von der Realität vorgegebenen Eigenschaften. Es werden unerhebliche Details abstrahiert und die für den Zweck des Modells wichtigen Eigenschaften so exakt wie möglich abgebildet. Für diesen Vorgang wird das konzeptionelle Modell benötigt. Mit ihm wird das reale oder abstrakte Modell abgebildet, sodass der Zweck vollständig erfüllt werden kann. Dazu wird das konzeptionelle Modell verifiziert, also mit dem vorgegebenen System abgeglichen.

Ist z.B. die Stromlinienförmigkeit eines Chassis für ein Automobil zu ermitteln, soll das Modell die möglichst exakte Nachbildung dieses Bestandteils sein. Unwichtig sind Komponenten im Inneren des Autos, welche die aerodynamischen Eigenschaften nicht beeinflussen. Diese müssen daher nicht im Modell abgebildet werden<sup>19</sup>.

---

<sup>18</sup> Vgl. Stachowiak 1973, S. 131-133

<sup>19</sup> Vgl. Simulation, 2.3

Zu unterscheiden sind zwei Modellarten - das analytische und das deskriptive Modell. Analytische Modelle bauen auf mathematischen und physikalischen Gegebenheiten auf, wobei hier eine Formel oder ein Naturgesetz die Grundlage für das Modell liefert. Deskriptive Modelle bilden eine vollständige Welt nach, was den Umfang erheblich steigert, da mehrere miteinander gekoppelte Vorgänge zusammenwirken und Ergebnisse beeinflussen. Gleichzeitig werden aber auch die Möglichkeiten zur Simulation erhöht.

Im Folgenden dient eine Software zur Berechnung von Planetenbahnen als begleitendes Beispiel, um alle Phasen der Modellbildung zu beleuchten und praktisch anzuwenden.<sup>20</sup>

#### 3.1.1 Vorstellung des begleitenden Projektes

Die Software soll in der Lage sein, die Flugbahn von Planeten zu berechnen. Dazu werden realistische Daten von der NASA als Validierungsgrundlage eingesetzt. Basierend auf dem Vielteilchenproblem werden verschiedene mathematische Gleichungen aufgestellt, die alle Gegebenheiten im Raum, wie Anziehungskräfte zwischen den Massen, Gravitation etc., berücksichtigen. Ziel ist es, eine Software zu realisieren, welche korrekte Werte für die Planetenposition zu bestimmten Datumsangaben liefert.

Auch Galaxien sollen ein Bestandteil sein, da man eine Vielzahl von Körpern beim bilden verschiedener Formationen auf derselben mathematischen Grundlage beobachten kann.

Wichtig ist zu nennen, dass das Modell dynamisch ist. Es ist also zeitabhängig und beinhaltet damit einen Zeitschritt. Dieser bestimmt im Wesentlichen die Genauigkeit und den Aufwand der Berechnung. Da alle Berechnungen in jedem Zeitschritt für jeden beteiligten Körper durchgeführt werden müssen, errechnet sich ein Aufwand von  $n(O^2)$ . Damit ist die Weite des Zeitschrittes ausschlaggebend für die Dauer der Berechnung. Auch diese Entscheidung, wie exakt die Ergebnisse sein sollen bzw. in welchem Bereich der Genauigkeit, muss bei der Konzepterstellung getroffen werden.

---

<sup>20</sup> Vgl. Oberkampf/Roy 2010, S. 116 ff

## 3.2 Überblick der Modellbildung

Damit ein Modell erstellt werden kann, ist zunächst das System zu wählen, welches abgebildet werden soll. Dieses Interessengebiet ist klar einzugrenzen, um den Zweck des Modells deutlich zu definieren.

Ist das System gewählt, folgen sechs unterschiedliche Phasen, in denen das Computermodell herausgebildet wird. Diese Phasen bauen aufeinander auf, können jedoch zurückgelaufen werden. Sprünge innerhalb der Phasen sind nicht auszuführen, da so wichtige Entwicklungsschritte ausgelassen und das Modell nicht erfolgreich verifiziert und validiert werden kann.

### 3.2.1 Konzeptionelle Modellierungsphase

Die erste Phase nach der Wahl des Systems ist die Erstellung des konzeptionellen Modells. Hier wird das System abstrahiert und ein Konzept erstellt, welches die wichtigsten, für den Zweck des Modells relevanten Eigenschaften abbildet.

Weiter werden die Umgebungsvariablen deklariert, aber nicht modelliert<sup>21</sup>, da diese auf das System einwirken, aber kein Bestandteil desselben sind. Ebenfalls hier integriert ist der Einsatzort des Modells, also in welchen Szenarien und in welcher Umgebung die Simulation durchgeführt werden soll.

Bezogen auf das begleitende Beispiel der Planetenbahnberechnung wird hier festgelegt, dass die einzelnen Körper in einer Beziehung zueinander stehen. Ebenfalls im Konzept verankert ist die Berechnung und Verbindung der Position und Geschwindigkeit der beteiligten Planeten. Um im weiteren Verlauf Galaxien nachbilden zu können, muss die Anzahl der am System beteiligten Körper variabel gehalten werden, ebenso der Zeitschritt zur nachhaltigen Beeinflussung der Genauigkeit.

### 3.2.2 Mathematische Modellierungsphase

Ausgehend von dem konzeptionellen Modell wird nun bei analytischen Modellen, die mathematische Grundlage gebildet. Hier werden Formeln aufgestellt, welche das Konzept korrekt abbilden und den Zweck des Modells erfüllen. Sollte es sich um ein deskriptives Modell handeln, ist dieser Schritt anders durchzuführen, was jedoch kein Bestandteil dieser Arbeit ist.

---

<sup>21</sup> Vgl. 2.1, Def. System

Ebenfalls werden hier Verbindungen zu anderen Modellen hergestellt. Sollte ein Bestandteil des konzeptionellen Modells bereits in einem anderen ausreichende Ergebnisse liefern, so werden hier die Schnittstellen deklariert und in das mathematische Modell eingepflegt.

Im begleitenden Beispiel wird nun die gekoppelte Bewegungsgleichung eingeführt, welche die eigentliche Berechnung darstellt. Wichtig ist hier die Reihenfolge, da ausgehend von der Geschwindigkeit zu einem bestimmten Zeitschritt eine neue Position errechnet werden kann (Vgl. Abb. 3).

$$F_x^i = -G \sum_{k=0}^{\text{AnzahlObjekte}} \frac{m^i m^k}{\|r^i - r^k\|} (x^i - x^k) = m^i \frac{d^2}{dt^2} x^i$$

$$F_y^i = -G \sum_{k=0}^{\text{AnzahlObjekte}} \frac{m^i m^k}{\|r^i - r^k\|} (y^i - y^k) = m^i \frac{d^2}{dt^2} y^i$$

$$F_z^i = -G \sum_{k=0}^{\text{AnzahlObjekte}} \frac{m^i m^k}{\|r^i - r^k\|} (z^i - z^k) = m^i \frac{d^2}{dt^2} z^i$$

**Abbildung 3: Gekoppelte Bewegungsgleichung**

Die Gleichung für die neue Geschwindigkeit und Position werden in Abb. 4 aufgezeigt.

$$\frac{d}{dt} x_{i+1} = \frac{d}{dt} x_i + h \frac{d^2}{dt^2} x_i$$

$$x_{i+1} = x_i + h \frac{d}{dt} x_i$$

**Abbildung 4: Geschwindigkeit und Position**

Mit dieser Grundlage kann nun das konzeptionelle Modell mit dem mathematischen Modell verifiziert werden, um zu garantieren, dass die Gleichungen den Zweck des Modells vollständig erfüllen.

### 3.2.3 Diskretisierungsphase

Nun werden die in der vorangegangenen Phase erstellen Gleichungen für die Erstellung des Computermodells notwendige, diskrete Form entwickelt.

Weil Computer Berechnungen nur diskret lösen können, ist dieser Schritt notwendig, da häufig Formeln nicht numerisch lösbaren Bestandteilen besitzen. Diese müssen dann zu numerisch lösbaren Gleichung umgeformt werden.

Dies verhindert ebenfalls auftretende Fehler und Ungenauigkeiten, da so näherungsweise Lösungen errechnet und anschließend interpretiert werden können. Hier soll auch der Zweck des Modells beachtet werden, da die Genauigkeit der diskreten Abläufe ausschlaggebend für die einzusetzenden Mittel sein kann. Bedürfen die Berechnungen sehr viele Einzelstufen, ist mehr Rechnerkapazität von Nöten, um diese in einer angemessenen Zeit durchführen zu können. Diese Kapazitäten müssen vorhanden sein und finanziert werden, sodass bereits in der dritten Phase der Modellbildung wirtschaftliche Faktoren mit in den Prozess einfließen.

Im begleitenden Beispiel werden die in 3.2.2 dargestellten Gleichung derart umgeformt, dass sie diskret von einem Computer zu lösen sind.

#### **3.2.4 Implementierungsphase**

Nun sind alle nötigen Mittel vorhanden, um ein Computerprogramm zu implementieren. Das System wurde in ein Konzept überführt, mit analytischen Mitteln gefestigt und diskretisiert. Sind alle Vergleiche mit der jeweils vorangegangene Phase erfolgreich verlaufen und der Zweck des Modells wird erfüllt, ist das diskrete Modell nun zu implementieren.

Hierzu werden zunächst eventuelle Eingangsdaten gesichtet, um eine entsprechende Datenstruktur herzustellen. Anschließend wird die Klassen- und Objektstruktur aufgebaut und Kopplungen zu anderer Software festgelegt. Jetzt wird der Programmcode implementiert.

Die Software für die Berechnung beinhaltet diverse Klassen zur grafischen Darstellung, Validierung und Berechnung. In Abbildung 5 ist das Klassendiagramm der Berechnung zu sehen.

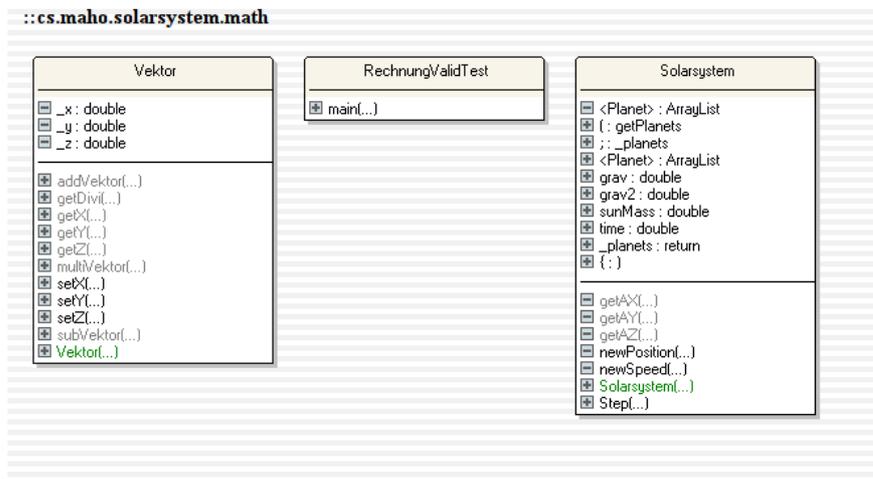


Abbildung 5: Klassendiagramm der Berechnung

Abbildung 6 zeigt den Programmcode, welcher die in 3.2.2 dargestellten Gleichungen umsetzt.

```

//Berechnung start
private void newPosition()
{
    for(int i = _from; i < _to; i++)
    {
        _planets.get(i).getPosition().setX(_planets.get(i).getPosition().getX() + Galaxy._time * _planets.get(i).getSpeed().getX());
        _planets.get(i).getPosition().setY(_planets.get(i).getPosition().getY() + Galaxy._time * _planets.get(i).getSpeed().getY());
        _planets.get(i).getPosition().setZ(_planets.get(i).getPosition().getZ() + Galaxy._time * _planets.get(i).getSpeed().getZ());
    }
}

private void newSpeed(){
    for(int i = _from; i < _to; i++)
    {
        _planets.get(i).getSpeed().setX(_planets.get(i).getSpeed().getX() + Galaxy._time * getAX(_planets.get(i)));
        _planets.get(i).getSpeed().setY(_planets.get(i).getSpeed().getY() + Galaxy._time * getAY(_planets.get(i)));
        _planets.get(i).getSpeed().setZ(_planets.get(i).getSpeed().getZ() + Galaxy._time * getAZ(_planets.get(i)));
    }

    //Neue Geschwindigkeit wird gespeichert
    for(int i = _from; i < _to; i++)
    {
        _planets.get(i).setTempSpeed();
    }
}

private double getAX(Planet refPlanet)
{
    double A = 0;
    for(int i = 0; i < _planets.size(); i++)
    {
        //Für alle anderen Planeten
        if(!_planets.get(i).getId().equals(refPlanet.getId())){
            A = A + Galaxy.grav * ((_planets.get(i).getMass() / Vektor.getDivi(_planets.get(i).getSpeed(), refPlanet.getSpeed())) * (_planets.get(i).getPosition().getX() - refPlanet.getPosition().getX()));
        }
    }
    return A;
}
    
```

Abbildung 6: Programmcode zur Berechnung

Die Planetenbahnberechnung ist nun mit der mathematischen Grundlage ausgestattet und verfügt über eine nicht statisch festgelegte Anzahl von beteiligten Körpern. Eben sowenig ist der Zeitschritt festgelegt, wodurch die Genauigkeit der Berechnung im Nachhinein beeinflusst werden kann.

### 3.2.5 Numerische Lösungsphase

Alle Vorbereitungen für das Modell sind nun abgeschlossen. Es liegt jetzt eine Software vor, welche das in Phase eins „Konzeptionelle Modellierungsphase“, erstellte Modell in einem Computerprogramm abbildet. Die jeweils aufeinander folgenden Phasen erfüllen den Zweck des Modells, wodurch nun die Simulation folgen kann. Die Modellbildung ist zu diesem Zeitpunkt abgeschlossen, der Prozess wird jedoch fortgesetzt, um alle Bestandteile in einen Zusammenhang zu bringen.

In der numerischen Lösungsphase wird die Simulation am erstellten Modell durchgeführt. Als Ergebnisse werden Zahlen geliefert. Diese sind in Auflösung und Genauigkeit das Endergebnis der Berechnung, und werden zur Interpretation genutzt.

Das begleitende Beispiel errechnet Vektoren für die Koordinaten  $x$ ,  $y$  und  $z$ . Anhand diese Lösungen kann die Position eines jeden an der Berechnung teilgenommenen Körpers für einen beliebigen Zeitschritt abgelesen werden.

### 3.2.6 Präsentation der Lösungen

Abschließend werden die in 3.2.5 errechneten Lösungen interpretiert. Sie werden von der abstrakten, tabellarischen Form in grafische umgesetzt, sodass die Ergebnisse Rückschlüsse auf die Realität zulassen. Ebenfalls kann hier eine Validierung durchgeführt werden, um die Qualität der Ergebnisse zu sichern.

Es wird jedoch nicht nur das Ergebnis vorgestellt, auch Eingangsdaten, Moduldesign, Programmcode, Rohdaten und interpretierte Ergebnisse werden vorgestellt, um ein umfassendes Bild über den gesamten Prozess zu schaffen.

Als valide können die Daten der Planetenbahnberechnung nicht angesehen werden, da bei einer grafischen Gegenüberstellung der errechneten und von der NASA erhaltenen Daten ergeben hat, dass die Ergebnisse teilweise sehr große Abweichungen aufzeigen. Auch dies ist ein Teil der Interpretation, da so die Implementierungsphase bzw. die Diskretisierungsphase erneut bearbeitet werden sollte.

## 3.3 Abschluss der Modellbildung

Der in 3.2 beschriebene Prozess zeigt auf, dass die Modellbildung eine Reihe von aufeinander folgenden Phasen benötigt, um ein System in ein Modell zu überführen. Dabei ist die korrekte Reihenfolge sowie die Verifizierung einzelner Bestandteile Voraussetzung, damit Fehler und Ungenauigkeiten weitestgehend vermieden werden.

Wie im begleitenden Beispiel zu sehen, ist jedoch diese Abfolge von Prozessen kein Garant dafür, dass die Ergebnisse Rückschlüsse auf die Realität zulassen. Wichtig ist, dass alle Ergebnisse validiert werden, obgleich festgestellt wird, dass diese nicht einsetzbar sind.

Modellbildung ist ein fortlaufender Prozess, bis das Modell genau den Zweck sowie die Qualitätsforderungen erfüllt. Die aus der folgenden Simulation erhaltenen Daten können dann Entscheidungshilfen in der Realität sein, um z.B. Brücken und Gebäude entsprechend zu bauen oder das Wetter vorherzusagen.

## 4 Verschiedene Simulationsprinzipien

### 4.1 Einführung Simulationsprinzipien

Generell sind Simulationen heutzutage auf kleineren Endgeräten, wie Smartphones und Notebooks, möglich. Als einfaches Beispiel ist die Aktienkursberechnung zu nennen, welche alle Eigenschaften eines Modells und deren Simulation beinhaltet. Auch hier werden Daten eingelesen, in einem System abgearbeitet und anschließend interpretiert. Die Vorhersage dient dann als Entscheidungshilfe, in einen Kurs zu investieren oder zu verkaufen.

Komplexere Simulationen, wie in dem begleitenden Beispiel zur Modellbildung, bedürfen jedoch einiges mehr an Computerkapazität und Zeit. Unterschiedliche Strategien werden eingesetzt, um diese Simulationen umfangreich und zielgerichtet durchzuführen.<sup>22</sup>

### 4.2 Ereignisorientierte Simulation

Die ereignisorientierte Simulation beschreibt eine zeitunabhängige Simulation, bei der durch Zustandsänderungen Ereignisse hervorgerufen werden. Ereignisse können aber auch initialisiert werden und von Anfang an als Ausgangspunkt definiert sein. Alle Folgeereignisse werden abgearbeitet, wobei die Zwischenzeit einzelner Schritte übersprungen werden kann. So arbeitet sich die Simulation Schritt für Schritt von Ereignis zu Ereignis.

---

<sup>22</sup> Vgl. Friedmann (1995), S. 4 ff

Als Beispiel sind Multi-Agenten-Simulationen und Diskret-Ereignis-Simulationen zu nennen. Eine ampelgesteuerte Verkehrskreuzung kann auf diese Weise simuliert werden, da ein Umschalten einer Ampeleinheit Folgen auf alle anderen hat. Um zu simulieren, wie sich die übrigen Komponenten verhalten, muss man nicht auf die Zeitvariable warten, sondern ruft das nächste Schaltereignis auf.

### 4.3 Transaktionsorientierte Simulation

Als eine Unterkategorie der Ereignisorientierten Simulation ist die Transaktionsorientierte Simulation zu sehen. Hier sind verschiedenen Stationen untereinander vernetzt und reichen einander Arbeitseinheiten weiter. Die verschiedenen Aufenthaltszeiten der Arbeitseinheiten an den Stationen führen zu Abhängigkeiten derselben und beeinflussen damit das Simulationsergebnis.

Hier ist eine Flughafenabfertigung als Beispiel anzusehen. Die verschiedenen Stationen wie Sicherheitskontrolle, Kofferabgabe und das Betreten des Flugzeuges reichen die Fluggäste, respektive die Arbeitseinheiten, aneinander weiter. Wie lange das dauert, wo sich Staus bilden und welche Probleme noch auftreten, könnte man so simulieren.

### 4.4 Prozessorientierte Simulation

Auch die Prozessorientierte Simulation ist eine Unterart der Ereignisorientierte Simulation, sind hier jedoch die Stationen komplexe, für sich eigenständige Prozesse. Sie wirken sich direkt auf die Simulationszeit aus und verändern Zustände lokaler und systemweiter Variablen. Je nach Umfang der Simulation sind die beiden in 4.2 und 4.3 genannten Punkte ebenfalls Beispiele für dieses Simulationsprinzip.

### 4.5 Zeitgesteuerte Simulation

Das Prinzip der Zeitgesteuerten Simulation ist, dass ein Zeitschritt die einzelnen Epochen der Simulation angibt. Das Simulieren nach diesem Zeitschritt läuft kontinuierlich ab, es werden keine Schritte übersprungen und in jedem Schritt führen alle Modellkomponenten Zustandsänderungen durch. Systeme aus der Physik, Mathematik und dem Technikbereich werden zeitgesteuert simuliert, um numerische Lösungen für verschiedenen Epochen zu erhalten. Wettermodelle sowie astronomische Forschungen können so abgebildet werden.

#### **4.6 Stochastische Simulation**

Zufällig ermittelte Zufallswerte und deren Eintrittswahrscheinlichkeit sind die Haupteigenschaften dieses Simulationsprinzips. Er dient als Sammelbegriff für alle Simulationen, die mit zufälligen Werten rechnen, z.B. die Monte-Carlo-Simulation.

Die näherungsweise Berechnung der Zahl Pi ist ein Beispiel, bei dem Zufällig in einem Quadrat Punkte gesetzt werden. Durch das Quadrat verläuft ein Viertelkreis, welcher das Quadrat in einen inneren und äußeren Bereich teilt. Nachdem nun einige zufällige Punkte innerhalb und außerhalb des Viertelkreises gesetzt wurden, wird ein Verhältnis derer berechnet. Dieses Verhältnis ist annähernd die Zahl Pi. Bei dieser Simulation sind jedoch weitere Komponenten elementar, wie die Art der Zufallsgenerierung und die Anzahl der Punkte.

#### **4.7 Nutzen der Prinzipien**

Die hier vorgestellten Prinzipien sind gewählt worden, da sie viele Möglichkeiten zur Lösung komplexer Simulationen aufzeigen. Diverse Unterkategorien und Abstufungen lassen eine Strategie finden, mit der ein gewähltes System zielgerichtet simuliert werden kann.

### **5 Vorteile und Grenzen**

Wichtig ist es, Modelle und deren Simulation als Untersuchungsmethode zu verstehen. Diese können immer nur einen Teil der Realität abbilden, und daher nie exakte Vorhersagen treffen, wie sich etwas in der Wirklichkeit verhalten wird.

Ein großer Vorteil liegt jedoch in der Freiheit der Modelle – alles kann modelliert werden, um Simulation durchzuführen. Zeit, Raum und Ort können beliebig sein, je nach Aufwand und Ziel.

Mit der Entwicklung im Bereich SC werden Simulationen und deren Modelle auf eine völlig neue Stufe gehoben. Durch sie können Gebäude, Brücken, Weltraummissionen, Wettersimulationen und Windkanaltests vollständig abgebildet und simuliert werden. Dadurch läuft man nicht mehr Gefahr, beispielsweise eine Brücke zu bauen welche bei ungünstigen Windbedingungen in sich zusammenbricht.

Als Beispiel ist hier die Tacoma Narrow Bridge zu nennen, die 1940 in Amerika aufgrund ihrer Eigenfrequenz, welche durch den Wind angeregt wurde, in sich zusammengestürzt ist.

Die Interpretation der Daten ist der wichtigste Bestandteil von Simulationen, da nur so die Realität beeinflusst werden kann. Hier sind die Vorteile und Grenzen des jeweiligen Modells abzuwägen und dementsprechend das Gewicht der Ergebnisse korrekt zu wählen. Untersuchungsmethoden können Fehler aufweisen, wie im begleitenden Praxisprojekt zur Planetenbahnberechnung zu sehen war. Hier sollte der Prozess zur Modellbildung solange überarbeitet werden, bis die Ergebnisse den Qualitätsforderung entsprechen und die Praxis positiv beeinflussen.

## **6 Ausblick**

Simulationen werden einen immer größeren Bestandteil in der Forschung und Entwicklung aller Bereiche auf der Welt einnehmen. Sie sparen Kosten und erhöhen die Sicherheit aller Beteiligten. Verschiedene Projekte von diversen Universitäten und anderen Einrichtungen lassen darauf schließen, dass mit der Weiterentwicklung der Computertechnik immer umfangreichere und genauere Ergebnisse erzielt werden können.

## **7 Literaturverzeichnis**

Stachowiak, Herbert (1973): Allgemeine Modelltheorie. Wien: Springer-Verlag.

VDI (200): VDI Richtlinien 3633-1. Düsseldorf: Verein deutscher Ingenieure

DIN EN ISO 8402, 1995-08, Ziffer 3..5

Mattern, Friedemann (1995): Modellbildung und Simulation, Teilarbeit „Identität der Informatik“, DZWR (Darmstädter Zentrum für Wissenschaftliches Rechnen“

## Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten und nicht veröffentlichten Schriften entnommen wurden, sind als solche kenntlich gemacht. Die Arbeit ist in gleicher oder ähnlicher Form oder auszugsweise im Rahmen einer anderen Prüfung noch nicht vorgelegt worden.

Hamburg, den 31.03.2013

---

Holger Trampe