

## Neue Wege in der Modellierung – durch neue Mathematik<sup>1</sup>

Die Simulation natürlicher oder technischer Systeme ist ein beliebtes Werkzeug auch in der Schule, mit dem man anhand vielfältiger Anwendungen zeigen kann, wie technische, physikalische und mathematische Gesetzmäßigkeiten in der Praxis zusammenkommen. In neueren, auf “Physical Modeling” basierenden Simulationsprogrammen werden Modelle einfach dadurch erstellt, dass die benötigten Bauteile (Massen, Federn, Widerstände, Rohre, Hydraulikzylinder) in einer graphischen Umgebung einer Bibliothek entnommen, mit geeigneten Parametern versehen und miteinander verbunden werden. Diese Vorgehensweise wurde möglich aufgrund neuer Methoden zur Vereinfachung komplexer Gleichungssysteme, die auf graphischen Algorithmen basieren. Wie sie funktionieren, wird anhand eines einfachen Schwingersystems explizit vorgeführt. Dabei ist zum Verständnis der grundlegenden Verfahren Schulmathematik ausreichend.

### 1. Simulation in der Schule

Simulationsverfahren sind nicht nur ein wichtiges Hilfsmittel in der Forschung und Entwicklung, sondern auch im Schulunterricht in vielfältiger Weise einsetzbar. Zwar benötigt man zum vertieften Verständnis Kenntnisse aus der Mathematik, der Informatik und dem untersuchten Anwendungsbereich – meistens den Natur- oder Ingenieurwissenschaften, aber auch etwa der Ökonomie; man hat aber je nach Vorwissen oder Lernziel die Möglichkeit, das Anforderungsniveau in weiten Grenzen zu wählen: von einfacher Algebra bis zu Differenzialgleichungen, von einer fertigen Simulationsumgebung bis zur eigenen Programmierung, vom Ohmschen Gesetz bis zu komplexen technischen Systemen.

Diese Auswahl erfordert natürlich didaktisches Gespür, was leider selbst bei Entwicklern von Abituraufgaben nicht automatisch vorausgesetzt werden kann. So zeigen die in [1] und [2] vorgestellten Beispiele, wie man im Grunde simple mathematische Probleme mit einer dicken “Anwendungssoße” zukleistern kann. Damit wird weder Verständnis für die Mathematik noch für die Anwendung gefördert. Dabei sind einige der dort untersuchten Verfahren nicht einmal wirklich praxisrelevant, sondern mühsam konstruiert, während die tatsächlich praktisch verwendeten Methoden für den Schulunterricht zu schwer sind. Ist also die verwendete Mathematik der eigentliche Lerninhalt, müssen Beispiele und Simulationsmethoden sehr sorgfältig daraufhin ausgewählt werden.

Ist man dagegen eher an der Untersuchung der Anwendung interessiert, etwa in der Physik, kann man Simulationen zunächst als “Black Box” im Sinne virtueller Experimente einsetzen, die reale Experimente ergänzen [3]. Untersucht man anschließend, wie die Simulationen zu ihren Ergebnissen kommen, kann man ein vertieftes Verständnis auch für die eingesetzten mathematischen Methoden zu erreichen versuchen.

Ausgehend von dieser Idee soll im Folgenden anhand eines einfachen Feder-Masse-Systems dargestellt werden, wie aktuelle, auf “Physical Modeling” basierende Simulationsprogramme [4] grundsätzlich funktionieren. Sie erlauben das Erstellen komplexer Modelle in einer graphischen Oberfläche, indem die benötigten Bauteile einer Bibliothek entnommen und miteinander verbunden werden. Die Programme erstellen daraus automatisch die Bewegungsgleichungen und lösen sie. Dabei können auch verschiedene physikalische Bereiche (etwa Mechanik, Elektrizitätslehre, Strömungslehre, Thermodynamik) miteinander gekoppelt werden. Ihr eigentlicher Anwendungsbereich sind zeitabhängige, kontinuierliche Systeme mit endlich vielen Freiheitsgraden (also z. B. keine elektromagnetischen Felder), aber verschiedene Erweiterungen erlauben auch weitergehende Modelle.

<sup>1</sup> Ausarbeitung eines Vortrags, gehalten auf dem “19. Forum für Begabungsförderung in Mathematik” an der Fachhochschule Südwestfalen in Soest, 17. - 19. März 2016.

Neben “klassischen” Verfahren zur Lösung von Differenzialgleichungen kommen hier graphentheoretische Verfahren zum Einsatz, die sich auch ohne tiefere Mathematik-Vorkenntnisse verstehen lassen. Die Darstellung folgt der aus [5], das Verweise auf die Originalliteratur enthält und auch etwas fortgeschrittenere Probleme diskutiert. Eine tiefergehende mathematische Behandlung findet man in [6].

## 2. Simulation mit Signalflussdiagrammen

In der weit verbreiteten Signalflussmethode werden Modelle aus Komponenten zusammengesetzt, die festgelegte Ein- und Ausgänge für zeitabhängige Signalwerte haben. Umfangreiche Block-Bibliotheken erlauben u. a. die Nachbildung beliebiger algebraischer Zusammenhänge. Ein besonders wichtiger Baustein ist der Integratorblock, mit dessen Hilfe beliebige (gewöhnliche) Differenzialgleichungen nachgebildet werden. Weit verbreitete Simulationsprogramme dieser Kategorie sind Matlab/Simulink [7] von der Firma Mathworks und die Open-Source-Alternative Scilab/Xcos [8].

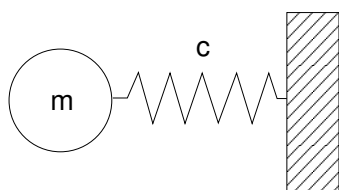


Abbildung 1: einfaches Feder-Masse-System

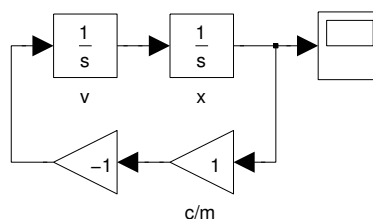


Abbildung 2: Simulink-Modell

Zur Nachbildung eines einfachen Feder-Masse-Systems (Abb. 1) muss man also zunächst dessen Differenzialgleichung aufstellen, was in der Schule schon eine recht hohe Hürde bedeutet – erst recht bei komplexeren Systemen. Im Beispiel erhält man die bekannte Schwingungsgleichung

$$m\ddot{x} + cx = 0. \quad (1)$$

Wie sie sich etwa in Simulink nachbilden lässt, zeigt Abb. 2. Auch ohne Simulink-Kenntnisse sieht man, dass sich die physikalischen Komponenten “Masse” und “Feder” in diesem Modell nicht wiederfinden lassen. Stattdessen reproduziert die Software daraus die Differenzialgleichung incl. der als Parameter gegebenen Anfangsbedingungen und löst sie mit Standardverfahren. Diese sind zwar hoch komplex, ihre Grundidee lässt sich aber auf das einfache Euler-Verfahren zurückführen, das durchaus auch in der Schule behandelt werden kann.

## 3. Simulation mit Physical Modeling

Die Modellierung mit Signalflussdiagrammen ist in den ingenieurwissenschaftlichen Anwendungen ein weit verbreitetes Verfahren. Es hat aber einen gravierenden Nachteil: Das Modell spiegelt nur die mathematische Struktur des Systems wider, aber nicht seinen Aufbau aus physikalischen Komponenten.

Die Methode des “Physical Modeling” ist entwickelt worden, um das zu ändern: Sie erlaubt die Nachbildung eines physikalischen Systems aus Bausteinen, die Massen, Federn, Widerständen, Spulen, Pumpen, Ventilen usw. entsprechen. Die Verbindungen der Komponenten bilden nicht Signalwege nach, sondern Flansche, Kabel oder Schläuche. Der Vergleich von Abb. 1 mit dem korrespondierenden MapleSim-Modell (Abb. 3) macht dies deutlich.

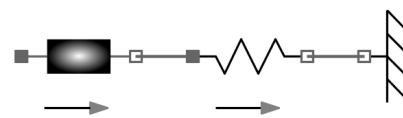


Abbildung 3: MapleSim-Modell

Entsprechende Simulationsprogramme sind inzwischen weitgehend ausgereift und werden in der Praxis eingesetzt, z. B. Dymola [9] von Dassault Systemes oder MapleSim [10] von Maplesoft. Mit OpenModelica [11] ist

auch hier eine Open-Source-Variante verfügbar. Dabei sind diese Programme in weiten Teilen kompatibel, weil sie alle auf der objektorientierten Modellierungssprache Modelica [4] basieren und die meisten Komponenten der umfangreichen freien Modelica-Bibliothek zur Verfügung stellen. Damit sind komplexe Anwendungen möglich wie etwa ein Fahrzeug-Triebstrang incl. Fahrermodell, eine Roboter-Steuerung oder ein Wärmetauscher [5].

## 4. Automatische Generierung der Bewegungsgleichungen

Mit Hilfe der Physical-Modeling-Methode kann man die Eigenschaften auch komplexerer Systeme “black-box”-artig untersuchen, etwa im Physikunterricht oder in technischen Fächern. Die spannende Frage ist aber natürlich: Wie funktioniert das? Woher kommen die benötigten Gleichungen und wie werden sie gelöst? Um das zu verstehen, müssen wir drei Details eines Modells genauer betrachten:

1. die Anschlüsse an einem Baustein,
2. die Bausteine selbst,
3. die Verbindungen zwischen den Bausteinen.

Ein Baustein hat einen oder mehrere Anschlüsse, die gleichen oder verschiedenen physikalischen Bereichen zugeordnet sind: Eine Feder-Komponente hat zwei mechanische Anschlüsse, ein Kondensator zwei elektrische, ein Hydraulikzylinder einen hydraulischen und einen mechanischen. An jedem Anschluss sind je nach Typ eine feste Anzahl physikalischer Größen vordefiniert, im hier betrachteten einfachsten Fall jeweils zwei. Bei einem mechanischen Anschluss sind das der Ort  $s$  des Anschlusspunkts und die Kraft  $f$ , die auf ihn wirkt, bei einem elektrischen Anschluss die Spannung  $u$  und die Stromstärke  $i$ , bei einem hydraulischen der Druck  $p$  und der Volumenstrom  $q$ . Die Bezeichnung der Größen folgt hier der in Modelica, was gelegentlich von der in der Physik üblichen abweicht.

Die Variablen werden nun in zwei verschiedene Typen unterteilt: die Flussvariablen, die oft die Zeitableitung einer erhaltenen Größe sind, und die Potenzialvariablen, die häufig um eine Konstante verschiebbar sind. An jedem Anschluss sind immer gleich viele Variable jeden Typs definiert, im einfachsten Fall also je eine. Diese Definitionen sind in der Bibliothek festgelegt und können einfach nachgesehen werden; es ist aber physikalisch lehrreich, sich das selbst zu überlegen: Bei elektrischen Anschlüssen gilt klarerweise

$$i = \dot{Q},$$

die Stromstärke ist die zeitliche Änderung der Ladung, welche selbst eine erhaltene Größe ist. Daher ist  $i$  hier die Flussgröße. Auch das andere Kriterium passt gut: Man kann zu allen Spannungswerten eine feste Konstante addieren, ohne das Systemverhalten dadurch zu verändern;  $u$  ist also eine Potenzialgröße. Auch bei der Hydraulik ist das recht klar, es gilt ja

$$q = \dot{V}$$

und das Volumen ist, wegen der konstanten Dichte des Öls und der Massenerhaltung, eine erhaltene Größe. Außerdem kann man den Druck in vielen Anwendungen – nicht in allen! – überall um eine Konstante erhöhen, da das Verhalten der meisten Bausteine nur von Druckdifferenzen abhängt. In der Mechanik ist die Kraft die Flussgröße, wegen

$$f = \dot{p}$$

und der Impulserhaltung ( $p$  ist hier der Impuls, nicht der Druck!). Und eine Änderung aller Orte um eine Konstante bedeutet ja nur eine Verschiebung des Koordinatenursprungs. Die folgende Tabelle zeigt die Anschlussgrößen einiger Physikbereiche im Überblick:

Physikbereich	Flussgröße	Potenzialgröße
1d-Mechanik, translatorisch	Kraft $f$	Ort $s$
1d-Mechanik, rotatorisch	Moment $\tau$	Winkel $\phi$
Elektrizitätslehre	Stromstärke $i$	Spannung $u$
Wärmetransfer	Wärmeleistung $Q_{\text{flow}}$	Temperatur $T$
Hydraulik	Volumenstrom $q$	Druck $p$

Eine Flussgröße hat anschaulich eine Flussrichtung, daher muss ihr Vorzeichen festgelegt werden. Dazu definiert man generell, dass eine Flussgröße positiv ist, wenn der Fluss in den Baustein hineingeht.

Die Bausteine haben nun die durch ihre Anschlüsse definierten Variablen sowie ggf. interne weitere Hilfsvariablen. Für jeden Baustein sind Gleichungen zwischen diesen Größen festgelegt, wobei für jedes Paar aus Fluss- und Potenzialgröße eine Gleichung definiert werden muss (nicht zwingend zwischen den beiden Größen) sowie eine weitere Gleichung für jede interne Variable. Darüber hinaus können Komponenten Parameter definieren, die während der Simulation konstant sind, etwa die Größe der Masse oder die Federkonstante.

Für unser Feder-Masse-Beispiel brauchen wir drei Bausteine (Abb. 4):

- Die Masse enthält neben den vier Anschlussgrößen noch eine Hilfsvariable  $v$  und hat einen Parameter  $m$ . Betrachtet man sie vereinfacht als Massenpunkt, lauten ihre drei Gleichungen

$$\begin{aligned} s_1 &= s_2 \\ v &= \dot{s}_2 \\ m \dot{v} &= f_1 + f_2 \end{aligned}$$

- Die Feder hat einen Parameter  $c$  für die Federkonstante und die zwei Gleichungen

$$\begin{aligned} f_1 + f_2 &= 0 \\ f_2 &= c(s_2 - s_1) \end{aligned}$$

- Eine Einspannung ist nötig, um das eine Ende der Feder zu fixieren. Ihre einzige Gleichung ist

$$s_1 = 0$$

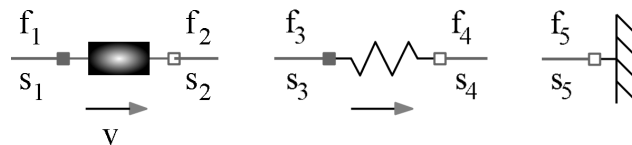


Abbildung 4: verwendete Bausteine

Diese Gleichungen bilden die physikalischen Grundgesetze nach (Newton, Hooke), wobei die auf den ersten Blick merkwürdigen Vorzeichen der Kräfte ihrer Richtungskonvention geschuldet sind: Eine positive Kraft zeigt jeweils in den Baustein hinein.

Offensichtlich ist die Zahl der Baustein-Gleichungen kleiner als die der Variablen. Die fehlenden Gleichungen ergeben sich aus den Verbindungen zwischen den Bausteinen gemäß folgender Regel: An einem Verbindungsknoten sind die Größen aller anliegenden Potenzialvariablen gleich, die Flussvariablen addieren sich zu Null. Für den Dreierknoten aus Abb. 5 ergeben sich daraus die drei Gleichungen

$$\begin{aligned} s_1 &= s_2 = s_3 \\ f_1 + f_2 + f_3 &= 0 \end{aligned}$$

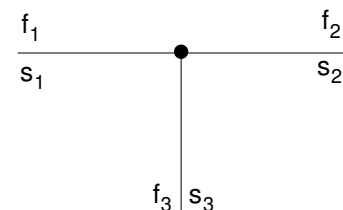


Abbildung 5: Verbindungsknoten

Die Regel gilt auch für eine einfache Verbindung zwischen zwei Komponenten, die als Zweier-Knoten interpretiert wird, und für einen “unbeschalteten” Anschluss: Dieser entspricht einem Einer-Knoten mit der Gleichung

$$f_1 = 0$$

Im Fall elektrischer Anschlüsse entsprechen die Verbindungsgleichungen dem 1. Kirchhoffschen Gesetz (“Knotenregel”). Im allgemeinen Fall sind sie eine Formulierung des Erhaltungsgesetzes, das der Definition einer Flussgröße zugrunde liegt. Auch der zunächst seltsam wirkende Fall der Einer-Knotenregel hat eine einfache Interpretation: Ein unbeschalteter Anschluss ist nicht “offen”, sondern “abgedichtet” – hier gehen keine Ladungen (oder Massen oder Impulse ...) hinein oder hinaus.

Mit diesen Regeln können nun die Bewegungsgleichungen für unser Beispielsystem vollständig aufgeschrieben werden. Dazu werden zunächst alle Anschlussgrößen und inneren Variablen eindeutig bezeichnet (Abb. 6).

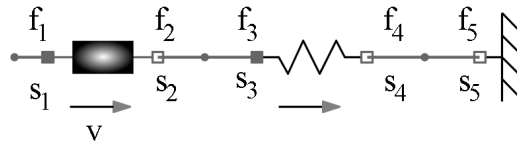


Abbildung 6: Gesamtmodell mit Bezeichnungen

Damit erhält man insgesamt folgendes Gleichungssystem:

$$\begin{aligned} s_1 &= s_2 & (a) \\ v &= \dot{s}_2 & (b) \\ m\dot{v} &= f_1 + f_2 & (c) \\ f_3 + f_4 &= 0 & (d) \\ f_4 &= c(s_4 - s_3) & (e) \\ s_5 &= 0 & (f) \\ f_1 &= 0 & (g) \\ s_2 &= s_3 & (h) \\ f_2 + f_3 &= 0 & (i) \\ s_4 &= s_5 & (j) \\ f_4 + f_5 &= 0 & (k) \end{aligned}$$

Die Gleichungen (a) - (f) sind die oben angegebenen Bauteilgleichungen, (g) entsteht durch den Einerknoten am linken Ende, (h), (i) und (j), (k) durch die Verbindungen zwischen Masse und Feder bzw. Feder und Einspannung. Insgesamt sind dies 11 Gleichungen für die 11 Unbekannten  $s_1, \dots, s_5, f_1, \dots, f_5$  sowie  $v$ . Vorausgesetzt, die Gleichungen sind unabhängig voneinander, sollte das System sich lösen lassen.

## 5. Vereinfachung und Lösung der Bewegungsgleichungen

Das gewonnene Gleichungssystem des Feder-Masse-Beispiels besteht aus den zwei Differentialgleichungen (b), (c) und gewöhnlichen (“algebraischen”) Gleichungen. Ein solches gemischtes System heißt *Differenzial-algebraisches System* oder *DAE-System*. Obwohl es nicht schwieriger wirkt als ein gewöhnliches Differentialgleichungssystem, ist seine Lösungstheorie deutlich komplizierter. Außerdem kennt man kein numerisches Verfahren zur direkten Lösung von DAE-Systemen. Es muss daher zunächst durch algebraische Umformungen in ein Differentialgleichungssystem umgeformt werden. Aufgrund der einfachen Struktur ist das hier durch Elimination der meisten Variablen leicht per Hand möglich und sollte auch Schülern mit etwas Durchhaltevermögen gelingen. Als Ergebnis erhält man

$$\begin{aligned} \dot{s}_2 &= v \\ \dot{v} &= -\frac{c}{m}s_2 \end{aligned}$$

Dies ist die Zustandsform der Bewegungsgleichung, die nur erste Ableitungen enthält. In dieser Form kann sie mit einem Standardverfahren numerisch gelöst werden. Durch Ableiten der ersten Gleichung und Einsetzen der zweiten kann man außerdem sofort die bekannte Schwingungsgleichung (1) reproduzieren.

Auch wenn die algebraische Umformung hier keine allzu großen Probleme bereitet, bleibt doch die Frage, wie das ein Simulationsprogramm machen kann – umso mehr, als selbst in kleinen Anwendungen die Zahl der Gleichungen schnell mehrere Hundert erreicht. Man benötigt also ein systematisches Verfahren, das die Auflösung solcher Gleichungssysteme ermöglicht. Zu diesem Zweck verwendet man den Tarjan-Algorithmus, der im Folgenden vorgestellt wird.

Zunächst analysiert man, wie man analytisch zur Lösung des Problems kommen kann. Dies erfordert zwei verschiedene Schritte:

1. Für jede Gleichung muss man festlegen, nach welcher Variablen sie aufgelöst werden muss. Beispielsweise kann man Gleichung (d) benutzen, um  $f_3$  aus  $f_4$  zu berechnen oder umgekehrt. Welche Richtung für eine erfolgreiche Auflösung des Gesamtsystems gewählt werden muss, ist zunächst nicht klar. Hat man sich für eine Richtung entschieden, schreibt man die Gleichung als Zuweisung etwa in der Form

$$f_3 := -f_4,$$

die auch als Anweisung in einem Programm interpretiert werden kann.

2. Die Reihenfolge der Gleichungen muss so geändert werden, dass auf den rechten Seiten der Zuweisungen jeweils nur vorher schon berechnete Größen stehen.

Dabei ist zu berücksichtigen, dass nicht eine Größe und ihre Ableitung gleichzeitig unabhängige Variable sind. Die unbekannte Größe ist jeweils die Ableitung (im Beispiel etwa  $s_2$ ), während man die Größe selbst, also hier  $s_2$ , als bekannt annimmt. Dies wird einsichtig, wenn man das Ziel der Umformung vor Augen hat: Eine Differenzialgleichung beschreibt die Ableitung als Funktion der Größen selbst. Auch die Vorgehensweise beim numerischen Lösen macht dies klar: Man startet mit einem bekannten Anfangswert für  $s_2$ , berechnet mit Hilfe der Differenzialgleichung daraus die Ableitung  $\dot{s}_2$ , folgt der Tangente ein Stück und erhält so den nächsten Wert für  $s_2$ .

Nach diesen Vorüberlegungen wird nun das Gleichungssystem als abstrakter Graph dargestellt, der nur die hier wesentliche Information enthält: welche Variablen in welchen Gleichungen auftreten. Dazu schreibt man auf der linken Seite untereinander Knoten, die mit den Gleichungsnamen bezeichnet sind. Auf der rechten Seite stehen untereinander Knoten für alle Unbekannten. Nun wird jeder Gleichungsknoten mit den Variablenknoten verbunden, die in der Gleichung vorkommen. Aus unserem Beispielsystem entsteht so der Graph der Abb. 7.

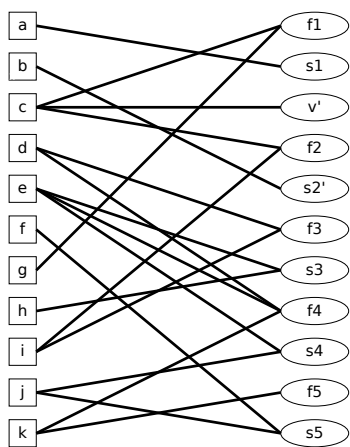


Abbildung 7: Ausgangsgraph

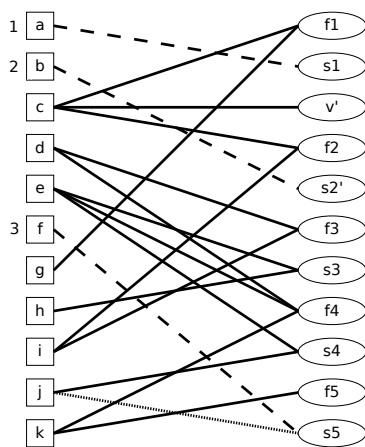


Abbildung 8: nach Schritt 3

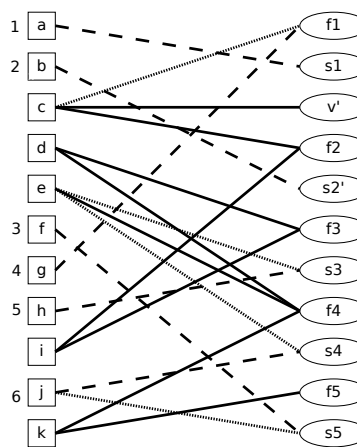


Abbildung 9: nach Schritt 6

Ziel des Tarjan-Algorithmus ist nun, alle Kanten entweder blau oder rot zu färben, so dass von jedem Gleichungsknoten nur zu Knoten mit bereits gefärbten Kanten führt.

chungsknoten und jedem Variablenknoten genau eine rote Kante ausgeht.<sup>2</sup> Außerdem werden die Gleichungen dabei durchnummeriert. Dazu geht man nach folgendem Schema vor:

```

suche Gleichungen mit genau einer ungefärbten Kante
  Gleichung bekommt kleinste freie Nummer
  Kante wird rot
  andere Kanten der zugehörigen Variablen werden blau
suche Variablen mit genau einer ungefärbten Kante
  zugehörige Gleichung bekommt größte freie Nummer
  Kante wird rot
  andere Kanten der zugehörigen Gleichung werden blau
iteriere, bis es nicht mehr weiter geht

```

Dabei kann man beliebig zwischen dem Gleichungs- und Variablenschritt hinundher springen, solange es Färbemöglichkeiten gibt. Der Einfachheit halber soll hier zunächst auf einer Seite von oben bis unten durchgegangen werden und dann auf die andere gewechselt. Dies führt im Beispiel zu folgenden Schritten:

- Gl. (a) hat nur eine Kante, die also rot gefärbt wird. Die zugehörige Variable  $s_1$  hat keine weiteren Kanten. Gl. (a) bekommt die Nummer 1.
- Analog wird die Kante von (b) nach  $s_2$  rot gefärbt, die Gl. wird Nummer 2.
- Nächste Gleichung mit nur einer ungefärbten Kante ist (f). Die Kante wird rot, die andere von  $s_5$  ausgehende wird blau. (f) wird Nummer 3 (Abb. 8).
- Ebenso werden nun bei Gl. (g), (h) und (j) die jeweils einzigen ungefärbten Kanten rot, die anderen Kanten der zugehörigen Variablen blau. Die Gleichungen bekommen die Nummern 4, 5, und 6 (Abb. 9).
- Nun wird auf die rechte Seite gewechselt. Die erste Variable mit einer ungefärbten Kante ist  $v$ , die Kante wird rot, die eine noch ungefärbte der zugeordneten Gleichung (c) wird blau. (c) bekommt die größte freie Nummer, also 11 (Abb. 10).
- Weiter geht es mit den Variablen  $f_2$ ,  $f_3$  und  $f_5$ , ihre zugeordneten Gleichungen (i), (d) und (k) bekommen die Nummern 10, 9 und 8 (Abb. 11).
- Unten angekommen, wird wieder auf die linke Seite gewechselt. In Gleichung (e) wird die einzige noch ungefärbte Kante rot, und (e) erhält die letzte freie Nummer 7 (Abb. 12).

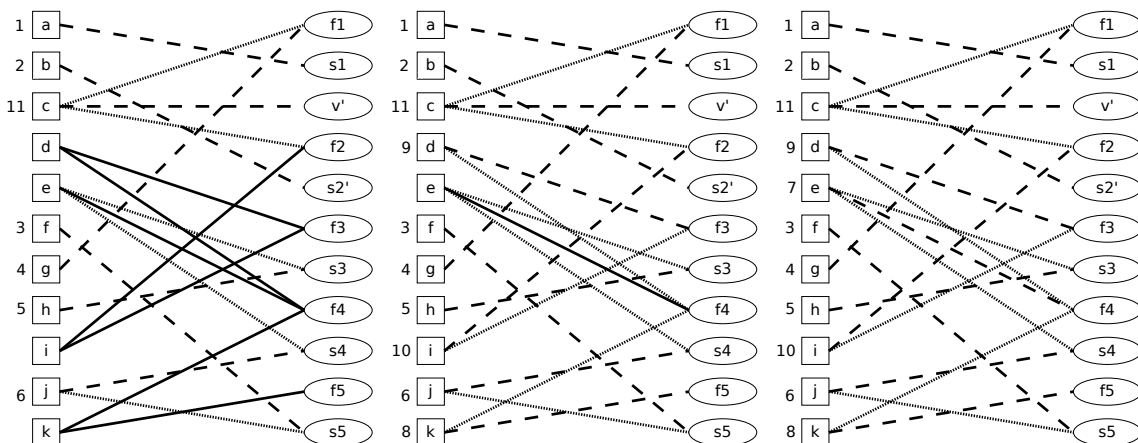


Abbildung 10: nach Schritt 7

Abbildung 11: nach Schritt 10

Abbildung 12: nach Schritt 11

<sup>2</sup>Hier werden rote Kanten gestrichelt, blaue gepunktet dargestellt.

Damit ist der komplette Graph nach den Anforderungen gefärbt und durchnummeriert. Das Gleichungssystem kann nun als Folge von Zuweisungen geschrieben werden, indem die Gleichungen jeweils nach der Variablen aufgelöst werden, mit denen sie durch eine rote Kante verbunden sind. Außerdem werden sie in die durch die Nummerierung vorgegebene Reihenfolge gebracht. Man erhält schließlich:

$$\begin{aligned}
 s_1 &:= s_2 && \text{(a)} \\
 \dot{s}_2 &:= v && \text{(b)} \\
 s_5 &:= 0 && \text{(f)} \\
 f_1 &:= 0 && \text{(g)} \\
 s_3 &:= s_2 && \text{(h)} \\
 s_4 &:= s_5 && \text{(j)} \\
 f_4 &:= c(s_4 - s_3) && \text{(e)} \\
 f_5 &:= -f_4 && \text{(k)} \\
 f_3 &:= -f_4 && \text{(d)} \\
 f_2 &:= -f_3 && \text{(i)} \\
 \dot{v} &:= \frac{1}{m}(f_1 + f_2) && \text{(c)}
 \end{aligned}$$

Geht man von bekannten Werten für  $s_2$  und  $v$  aus, kann man mit dieser Zuweisungsfolge alle weiteren Variablen berechnen, insbesondere die Ableitungen  $\dot{s}_2$  und  $\dot{v}$ . Man kann die Anweisungen daher als Inhalt einer Funktion  $\mathbf{f}(\mathbf{y})$  interpretieren, die dem Zustandsvektor  $\mathbf{y} := (s_2, v)$  seine Ableitung zuweist:

$$\begin{pmatrix} \dot{s}_2 \\ \dot{v} \end{pmatrix} = \mathbf{f} \begin{pmatrix} s_2 \\ v \end{pmatrix}$$

In dieser Formulierung kann das Differenzialgleichungssystem mit den üblichen Lösungsprogrammen (“DGL-Solvern”) gelöst werden. Dabei werden auch gleich Werte für alle vorkommenden Hilfsvariablen – interne Kräfte und Orte – bestimmt. Man kann die Formulierung aber natürlich auch dafür verwenden, per Hand alle Hilfsgrößen zu eliminieren und die Differenzialgleichung zu reproduzieren.

## 6. Erweiterungen der Lösungsverfahren

Wenn man nun versucht, diese Methode auf komplexere Systeme anzuwenden, wird man bald auf Beispiele stoßen, bei denen das Verfahren nicht sofort funktioniert. Dabei treten i. W. drei Arten von Schwierigkeiten auf, für die es jeweils Lösungsverfahren gibt, die den Tarjan-Algorithmus mit anderen Methoden kombinieren. Sie alle benötigen grundsätzlich keine wesentlich über Schulmathematik hinausgehenden Kenntnisse, sind aber aufgrund ihrer höheren Komplexität dennoch recht anspruchsvoll.

Das einfachste Problem ergibt sich z. B., wenn man die lineare Federkraft durch eine nichtlineare (etwa kubische) ersetzt. Dann ist der Teilschritt

“Löse Gleichung n nach Variable X auf”

algebraisch oft nicht möglich. In diesem Fall werden numerische Methoden zur Auflösung der Gleichung verwendet, vor allem das bekannte Newton-Verfahren.

Ersetzt man in unserem Standardbeispiel die Feder durch eine Hintereinanderschaltung zweier Federn, tritt eine andere Art von Problem auf: Der Tarjan-Algorithmus führt zunächst zur Färbung einer Reihe von Kanten, bis mehrere Knoten übrig bleiben, die alle mindestens zwei ungefärbte Kanten haben. Dies entspricht einem mehrdimensionalen Gleichungssystem, das sich nicht weiter zerlegen lässt. Es kann aber numerisch, etwa mit dem Newton-Verfahren, gelöst werden. Um das Problem weiter zu vereinfachen, kommen Färbungsmethoden auf Teilgraphen zum Einsatz.

Die schwierigste Problemklasse erhält man, wenn man statt zweier Federn zwei Massen in Reihe schaltet. Der Tarjan-Algorithmus bricht diesmal nicht ab, aber er liefert ein unerwartetes Ergebnis: Es gibt am Ende sowohl



eine Gleichung als auch eine Variable, die jeweils nur eine Kante haben, die aber beide blau gefärbt sind. Dies bedeutet, die Gleichung liefert keine Variable und die Variable wird in keiner Gleichung definiert. Solche Systeme heißen *strukturell singulär*, mathematisch handelt es sich dabei um DAE-Systeme mit einem Index  $> 1$ . Zu ihrer Lösung wird das System durch geschickt gewählte differenzierte Versionen einiger Gleichungen erweitert. Dabei kommen mehrmals Tarjan-Verfahren zum Einsatz.

## 7. Schlussfolgerungen

Das “Physical Modeling” ermöglicht in einfacher Weise die Erstellung und Simulation auch komplexer Modelle und erlaubt dadurch die Untersuchung physikalischer oder technischer Systeme auf Schulniveau. Dabei ist die Verfügbarkeit entsprechender Software als Open Source sicher hilfreich. Allerdings sollte man nicht den Eindruck gewinnen, die Modellierung würde sich damit auf das Zusammensetzen von LEGO-Bausteinen reduzieren! Ein Problem besteht in der Bewältigung der hohen Komplexität heutiger technischer Systeme – im Automobil- und Flugzeugbau sind Modelle mit über 100 000 Komponenten nicht unüblich! Aber auch bei scheinbar einfachen Modellen können Probleme auftreten, die eine sorgfältige Analyse der zugrundeliegenden mathematischen Methoden erfordern [12].

Mindestens ebenso spannend ist aber die Untersuchung der in diesen Programmen verwendeten Lösungsverfahren: Sie zeigen zum einen, dass langwierige algebraische Umformungen tatsächlich in praktisch relevanten Problemen auftauchen. Zum anderen sind die graphentheoretischen Methoden ein Beispiel für eine ganz andere Art von Mathematik, die grundsätzlich nicht schwer zu verstehen ist, aber dennoch zu interessanten Ergebnissen in unerwarteten Anwendungen führt. Voraussetzung ist dabei neben den Standardmethoden der Algebra und der Differenzialrechnung nur der Begriff der Differenzialgleichung, ohne weitere Lösungstheorie. Allerdings ist das Verständnis zumindest des Euler-Verfahrens zur numerischen Lösung hilfreich.

Die angedeuteten erweiterten Methoden sind sicher höchstens im Rahmen einer speziellen Förderung begabter Schülerinnen und Schüler vermittelbar. Dies liegt aber nicht daran, dass sie tiefer gehendes mathematisches Hintergrundwissen voraussetzen, sondern an ihrer algorithmischen Komplexität. Sie könnten daher durchaus eingesetzt werden, um die bei längeren Berechnungen benötigte Ausdauer und Sorgfalt anhand praktischer Beispiele zu üben. Sollte das etwa im Rahmen einer AG versucht werden, wäre der Autor sehr an einer Rückmeldung interessiert und gerne bereit, mit weiteren Beispielen oder Erläuterungen dabei Hilfestellung zu leisten.

## Literatur

- [1] Walser, H.: *Die Modellierung des schönen Scheins*. Mathematikinformationen, **55**, 3-14 (2011).
- [2] Baumann, A.: *Eine kritische Betrachtung zum Thema „Modellierungsaufgaben“ anhand von Beispielen aus dem hessischen Mathematik-Abitur 2009*. Mathematikinformationen, **55**, 15-23 (2011).
- [3] Junglas, P.: *Einsatz von Applets in der Physik-Ausbildung - Fallstudie Nichtlineare Systeme und Chaos*. Global J. of Engng. Educ. **7**, 3, 337-347 (2003).
- [4] Fritzon, P.: *Introduction to Modeling and Simulation of Technical and Physical Systems with Modelica*. John Wiley & Sons, New York (2011).
- [5] Junglas, P.: *Praxis der Simulationstechnik*. Verlag Europa-Lehrmittel, Haan-Gruiten (2014).
- [6] Cellier, F. E.; Kofman, E.: *Continuous System Simulation*. Springer, New York (2010).
- [7] MathWorks: *Simulink® User's Guide, Release R2016b*. MathWorks®, Natick (2016).
- [8] Scilab Enterprises: *Scilab Homepage*. <http://www.scilab.org> (2016).

- [9] Dassault Systemes: *Dymola Homepage*. <http://www.3ds.com/products-services/catia/products/dymola> (2016).
- [10] Maplesoft: *MapleSim Homepage* <http://www.maplesoft.com/products/maplesim/index.aspx> (2016).
- [11] Open Source Modelica Consortium: *OpenModelica Homepage*. <http://www.openmodelica.org> (2016).
- [12] {Junglas, P.: *Pitfalls using discrete event blocks in Simulink and Modelica*. In: ASIM 2016, Workshop der ASIM/GI-Fachgruppen STS/GMMS, Lippstadt, 90-97 (2016).

**Anschrift des Autors:**

Prof. Dr. Peter Junglas

Private Hochschule für Wirtschaft und Technik Vechta/Diepholz/Oldenburg

Schlesierstraße 13a

49356 Diepholz

E-Mail: [peter@peter-junglas.de](mailto:peter@peter-junglas.de)

Eingereicht am 29.10.2016