

SoWhat

**Ein Programm zur
einfachen Simulation
dynamischer Systeme**

Version 1.6

©Milestones Scientific Software,

Stefan Bachmann und Holger Strulik GbR 1992, 1996

Contents

1	Einleitung	3
2	SOWHAT Quick Start	7
3	Installation und Hardware	15
4	Allgemeine Hinweise zur Bedienung von SoWhat	17
5	Das Modell - Menü	21
6	Das Bearbeiten - Menü	33
7	Das Simulation - Menü	35
8	Das Tabellen - Menü	41
9	Das Politik-Menü	45
10	Das SOWHAT Hilfe - System	63
11	Das Verfahren zur Lösung der dynamischen Systeme	65
12	Meldungen und Fehlermeldungen	71
13	Erläuterungen zur Modellbibliothek	81

Chapter 1

Einleitung

Immer wenn dynamische Prozesse in stetiger Zeit formal beschrieben werden sollen, entstehen Modelle mit Differentialgleichungen bzw. Differentialgleichungssysteme, die i.d.R. nichtlinear sind und einer exakten analytischen Lösung nicht mehr zugänglich sind. Zur Untersuchung der Modelleigenschaften bieten sich dann zwei unterschiedliche Analysemethoden an. Qualitative Modelleigenschaften wie die Eindeutigkeit, Stabilität und Periodizität von Lösungen können für gering-dimensionale Systeme häufig in einer grafischen Phasendiagramm-Darstellung gewonnen werden. Die formale Analyse beschränkt sich hierbei im wesentlichen auf die Untersuchung, ob durch die spezifizierten Modellgleichungen die Bedingungen von in Frage kommenden mathematischen Theoremen erfüllt sind.

Die zweite Analysemöglichkeit besteht darin, mit Hilfe einer Simulation numerische Lösungen des Systems zu ermitteln. Diese Methode empfiehlt sich insbesondere dann, wenn eine qualitative Analyse nicht genügend Informationen liefert, tatsächliche Lösungswerte ermittelt werden sollen und eine Untersuchung von ungleichgewichtigen Systemzuständen und Anpassungspfaden vorgenommen werden soll. Von der numerischen Mathematik werden hierfür eine Reihe von Approximationsverfahren bereitgestellt. Mit Hilfe einer Computerimplementation eines geeigneten Verfahrens können für numerisch spezifizierte Modelle auf relativ einfache Weise Lösungspfade der endogenen Modellvariablen gewonnen werden, die dann anschließend einer theoretischen Diskussion unterzogen werden können. Um den erheblichen Programmieraufwand zu vermeiden, der entsteht, falls für jedes neu entwickelte bzw. zu untersuchende Modell auch ein neues

Simulationsprogramm entworfen und implementiert werden muss, wurde das hier vorgestellte Simulationsprogramm SOWHAT entwickelt.

SOWHAT wurde in enger Zusammenarbeit mit Spezialisten auf dem Gebiet der dynamischen Volkswirtschaftstheorie konzipiert. Es ist jedoch universell auch in der Biologie, Meteorologie, der Physik und anderen Wissenschaftsbereichen, die sich mit der Analyse dynamischer Systeme beschäftigen, einsetzbar. SOWHAT wendet sich in erster Linie nicht an jene Experten nichtlinearer Systeme, die auch gleichzeitig Computerfreaks sind und sich Ihre eigenen Speziallösungen unter Zuhilfenahme mathematischer Tools selbst programmieren. SOWHAT ist aus dem Bedarf jener großen Schar von Wissenschaftlern geboren, die sich permanent, häufig oder gelegentlich mit der Analyse dynamischer Systeme beschäftigen, denen jedoch Zeit und Möglichkeiten für eigene Programmentwicklungen fehlen. Wegen des sehr geringen Einarbeitungsaufwandes und der leichten Verständlichkeit von SOWHAT können z.B. Diplomanden und Doktoranden ihre Probleme auf einfachste Art numerisch lösen und ihre formalen Ergebnisse überprüfen und Studenten mühelos einen Einblick in das dynamische Verhalten der Modelle aus Vorlesung und Literatur gewinnen.

Das Kapitel 1 **SoWhat Quick Start**, die Online-Hilfe des Programms und etwa eine Stunde Einarbeitungszeit sollten ausreichen, um mit SOWHAT eigene einfache dynamische Systeme zu simulieren und auswerten zu können. Die Kapitel 3. bis 8. können als Referenz- und Nachschlagewerk benutzt werden. Ein jedes dieser Kapitel erklärt jeweils die Befehle und Funktionen eines PopUp-Menüs der SOWHAT-Menüleiste. Im 9. Kapitel erhalten interessierte Leser einige Informationen und Literaturhinweise zur in den Algorithmen verwendeten Mathematik. Das 11. Kapitel enthält eine Beschreibung der vom System generierten Fehlermeldungen, ihrer Ursachen und Möglichkeiten zu ihrer Beseitigung.

Nachbemerkung im Jahr 2004.

Die erste Version von SOWHAT wurde in einer Zeit entwickelt, in der viele Programme nur als DOS-Versionen verfügbar waren und Windows 3.0 als bahnbrechende Innovation hinsichtlich der Gestaltung benutzerfreundlicher Programme begann, den Markt zu erobern. Stellenweise betonen die Entwickler und Schreiber dieses Handbuchs daher, wie einfach (“durch Doppelklick”) ein

bestimmte Aufgabe erreicht werden kann oder beschreiben detailliert heute selbstverständliche Tätigkeiten (wie zum Beispiel das Markieren eines Listenobjekts). Der Leser möge diese softwarehistorischen Momente im Text mit einem Augenzwinkern (über-) lesen.

Als Hardwareaustattung benutzten wir gemäß des damaligen Standards zunächst sogenannte AT's, mit 16 Mhz getaktete IBM Rechner (mit mathematischem Co-Prozessor!), später dann sogenannte 486'er. Aus heutiger Sicht haben damit die damals bedeutsamen Mahnungen hinsichtlich Speicherkapazität und des Trade-offs zwischen Rechengenauigkeit und Geschwindigkeit praktisch ihre Bedeutung verloren.

Chapter 2

SOWHAT Quick Start

Erläuterungen an einem Beispielmodell

Die wichtigsten Elemente der Arbeit mit SOWHAT sollen anhand eines einfachen, willkürlich gewählten Beispiels vorgestellt werden.¹ Gegeben sei ein dynamisches System, das sich mit vier Differentialgleichungen erster Ordnung und drei Funktionen beschreiben lässt:

$$\dot{A} = aw_A - bA + fx$$

$$\dot{w} = cw - dw_AA + fx$$

$$\dot{x} = \sin(y + k)$$

$$\dot{K} = sY$$

$$Y = e^{\gamma t} A^\alpha K^\beta$$

$$y = Y/A$$

$$k = K/A$$

Ein Punkt über einer Variablen kennzeichnet dabei die erste Ableitung der betrachteten Variablen nach der Zeit. Die zeitliche Entwicklung der Variablen soll nun mit SOWHAT simuliert wer-

¹Das Modell finden Sie als `Beispiel.mod` in der mitgelieferten Modellbibliothek. Es sei nochmals daraufhin gewiesen, dass dieses Kapitel zur schnellen und effizienten Einarbeitung dient. Nach der Durcharbeitung dieses Kapitels sollten Sie mit der grundlegenden Bedienung von SOWHAT vertraut sein. Um einen vollständigen Überblick über die vielfältigen Möglichkeiten, die SOWHAT zur Analyse dynamischer Systeme bereitstellt, zu erhalten, sollten Sie jedoch in jedem Fall die Kapitel **Simulation**, **Mathematik** und **Politik-Simulation** durcharbeiten.

den. Nach dem Programmaufruf unter der Windows-Benutzeroberfläche meldet sich SOWHAT mit dem Einschaltbild. Durch Auswahl des Feldes **Model** in der Menüleiste mit der Maus öffnet sich ein Pop-Up-Menü, in dem der Befehl **New** erteilt wird, um dem Programm mitzuteilen, dass Sie nun ein neues Simulationsmodell eingeben wollen. Daraufhin erscheint das **Define Model**-Fenster der Abbildung 1.

Nach Eingabe des Modellnamens im **Name** - Feld können Sie die einzelnen Modellgleichungen eingeben. Hierbei können Sie sich weitgehend an der bereits auf dem Papier notierten Modellversion orientieren. Für das vorgestellte Modell werden die Gleichungen z.B. folgendermaßen eingegeben:

$$\begin{aligned}
 A. &= a * w_A - b * A + f * x \\
 w. &= c * w - d * w_A * A + f * x \\
 x. &= \sin(y + k) \\
 K. &= s * Y \\
 Y &= \text{\textasciitilde}(sysE, gamma * dtime) * \text{\textasciitilde}(A, alpha) * \text{\textasciitilde}(K, beta) \\
 y &= Y/A \\
 k &= K/A
 \end{aligned}$$

Jede Gleichungseingabe wird mit der **Return** - Taste oder **Next** beendet.² Die Eulersche Zahl **sysE** und der Sinus **sin** sind Funktionen, die SOWHAT bereitstellt. Die Systemvariable **dtime** enthält den Wert für die seit Beginn der Simulation vergangene Zeit. Zusätzlich wird noch die Gleichung

$$time = acttime$$

eingegeben. Sie gewährleistet, dass die Simulationszeit, die in der Systemvariable **acttime** abgelegt ist, als Protokollvariable referenziert werden kann. Mit **Previous** und **Next** oder mit Hilfe der TAB-Taste können Sie sich durch das bereits eingegebene Gleichungssystem bewegen und Modifizierungen an den Gleichungen vornehmen. Mit **Insert** lassen sich neue Gleichungen in

²Differenzgleichungen definieren Sie durch ein dem Gleichheitszeichen vorgestelltes &-Zeichen und Substitute durch einen vorgestellten Doppelpunkt (:). Erläuterungen hierzu finden Sie in Kapitel 5.2.

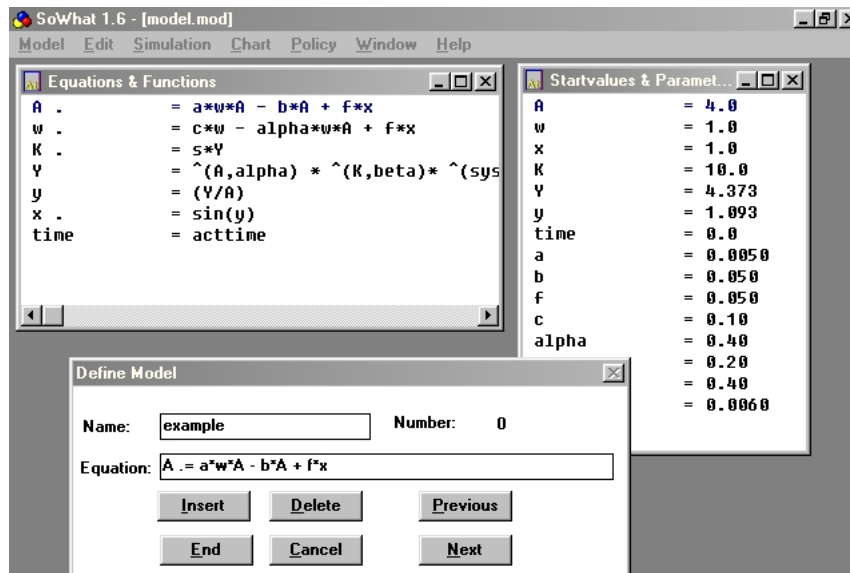


Abb. 1: Das Modell-Definieren Fenster

das System einfügen und mit **Delete** Gleichungen aus dem System entfernen. Die vollständige Eingabe wird mit **End** abgeschlossen. Der *Lexical Scan* überprüft sodann die syntaktische Richtigkeit der Eingabe. Wird ein Fehler entdeckt, z.B. eine fehlende Klammer, so erhalten Sie eine Meldung mit Fehlerart und Fehlerstelle und die fehlerbehaftete Gleichung erscheint. Sind alle Eingaben korrekt, so öffnet sich nach der **End** - Eingabe automatisch das Fenster **Startvalues & Parameters**, in dem Sie die Startwerte der dynamischen Variablen und die Modellparameter setzen.

Auch hier bewegen Sie sich mit **Previous** und **Next** bzw. *Return* durch die Variablen- und Parameterliste. Modellendogene Variablen werden vom Programm erkannt und als solche in der Liste ausgewiesen. Sie sind nicht editierbar. Die vollständige Eingabe wird auch hier mit **End** beendet. Damit ist das Modell vollständig spezifiziert.

Mit dem Befehl **Options** im Menü **Simulation** können nun die weiteren Simulationsoptionen gesetzt werden. In den Feldern **Start** und **End** wird dem Programm Start- und Endezeitpunkt der Simulation mitgeteilt. Unter **Record** kann festgelegt werden, nach welchem Zeitintervall die jeweils in der Simulation errechneten Werte vom Programm mitprotokolliert werden, die später in Tabellen ausgegeben und weiterverarbeitet werden können.

Mit **Minimal**, **Maximal** und **Start** werden im Unterpunkt **Step Control** die minimalen bzw. maximalen Schrittweiten und die Anfangsschrittweite definiert, die das Programm intern im Runge-Kutta-Algorithmus zugrundelegt. Damit haben Sie bei jeder Simulation erneut die Möglichkeit, die für Sie am besten erscheinende Kombination aus dem Trade off zwischen hoher Simulationsgenauigkeit (bei kleinen internen Schrittweiten) und hoher Simulationsgeschwindigkeit (bei großen Schrittweiten) in Abhängigkeit vom untersuchten Modell zu wählen. Unter **Max. Error** wird der maximal zulässige Simulationsfehler definiert. Falls Sie **Range Check** anwählen, wird bei jedem Rechenschritt überprüft, ob das Ergebnis im Rahmen der internen digitalen Zahlendarstellung liegt. Systemabstürze aufgrund von mathematischen Fehlern sind somit ausgeschlossen. Allerdings verlangsamt sich hierdurch die Simulationsgeschwindigkeit.

In dem rechts liegenden Fenster werden die Modellvariablen ausgewiesen. Durch Anklicken von ausgewählten Variablen wird dem Programm mitgeteilt, diese Variablenwerte mitzuprotokollieren und später tabellarisch auszuweisen. Falls Sie **Error relative** markieren, legt SOWHAT bei der internen Schrittweitenkontrolle nicht den absoluten, sondern den relativen Fehler zugrunde. Hierdurch erhöht sich im Bereich hoher Größenordnungen der Variablenwerte die Simulationsgeschwindigkeit beträchtlich. Die vollständige Eingabe wird mit **Ok** beendet. Es empfiehlt sich nun, bevor der erste Simulationslauf begonnen wird, das Modell mit den zugehörigen Parameterwerten und der im Optionsfenster festgelegten Simulationsumgebung abzuspeichern. Hierfür wird **Save** im Menü **Model** aufgerufen.

Die Simulation kann nun mit **Simulation Start** gestartet werden. Während die Simulation läuft, können Sie die Entwicklung der zu protokollierenden Variablen im **Results**-Fenster beobachten. Wird dieses nicht gewünscht, um z.B. eine kürzere Simulationsdauer zu erreichen, so können Sie das Fenster schließen. Bei Multi-Tasking-fähigen PC's läuft die Simulation im Hintergrund ab, so dass Sie währenddessen z.B. in anderen Programmen Texte schreiben können. Das Ende der Simulation wird Ihnen in jeder Windows-Anwendung durch Erscheinen der Meldung **Simulation End** und dem Ertönen eines Signaltones mitgeteilt.

Im **Charts** - Fenster werden die Namen der Tabellen, die die Werte der mitprotokollierten Modellvariablen enthalten, angezeigt. Um die Ergebnistabellen abzuspeichern, erteilen Sie den Befehl **Save** und markieren in dem erscheinenden Fenster die gewünschten Tabellen, die durch

Variablenname und Simulationsnummer identifiziert werden. A:0 enthält z.B. die Werte der Variablen *A* des ersten Simulationslaufes. Anschließend legen Sie das Dateiformat der Datei, in der die selektierten Charts abgelegt werden sollen, fest. Optional stehen die Formate *Excel*, *TSP*, *ASCII* und das Milestones-Data-Format *MDF* zur Verfügung. Nach Abschluss mit **Ok** öffnet sich ein neues Fenster, in dem Sie den Dateinamen und den Pfad der abzuspeichernden Datei festlegen.

Mit Hilfe der Befehle im Menü **Edit** können Sie die mit SOWHAT gewonnenen Daten direkt über die Windows-Zwischenablage in anderen Windows-Applikationen einfügen. Um z.B. die Ergebnistabellen Ihrer letzten Simulation in ein Tabellenverarbeitungsprogramm zu transferieren, wählen Sie den Befehl **Copy Last Simulation**, starten Ihre Tabellenkalkulation und wählen dort den entsprechenden Befehl zum Einfügen von Tabellen (Bearbeiten-Einfügen, Paste o.ä.). Mit dem Befehl **Copy Charts**, kopieren Sie beliebige von Ihnen ausgewählte Tabellen bisheriger Simulationsläufe in die Zwischenablage und mit dem Befehl **Copy Model** schreiben Sie Ihre Modelldefinition (Gleichungen, Funktionen, Parameterspezifikationen und Simulationsoptionen) in die Zwischenablage. Ihr Modell können Sie dann in einer unter Windows arbeitenden Textverarbeitung einfügen und dort z.B. in Texte einbinden, formatieren und ausdrucken.

Mit Hilfe von **Change** im Menü **Model** können Sie für eine neue Simulation die Variablen- und Parameterwerte neu setzen und die Modellgleichungen, wie bereits bei der Modelleingabe beschrieben, modifizieren und ergänzen. Mausbenutzer können Modifikationen noch leichter durchführen. Durch Doppelklick der linken Maustaste auf die zu ändernde Gleichung im **Equations & Functions** - Fenster oder den zu ändernden Startwert oder Parameter im **Startvalues & Parameters** - Fenster erscheint die zu ändernde Spezifikation sofort im **Change Model** - Fenster und kann direkt modifiziert werden. Mit dem Befehl **Load** im Menü **Model** kann aus einer vorliegenden Bibliothek von dynamischen Modellen durch Anklicken des Modellnamens ein neues Untersuchungsobjekt ausgewählt und für die Simulation geladen werden.

Eine weitere Besonderheit von SOWHAT-Simulationen ist ihre Unterbrechbarkeit. Durch den Befehl **Change Value Ändern** im Menü **Simulation** wird eine laufende Simulation unterbrochen, und es erscheint das **Simulation Interrupt** - Fenster, wie in Abbildung 2 dargestellt. Wird dort der **End** - Schalter betätigt, so wird die Simulation vollständig abgebrochen, wobei

die bis zum Simulationseende mitprotokollierten Ergebniswerte in den Tabellen vorliegen. Andererseits können Sie die Simulation auch mit neu festgelegten Werten der Parameter oder modifizierten Startwerten fortsetzen.

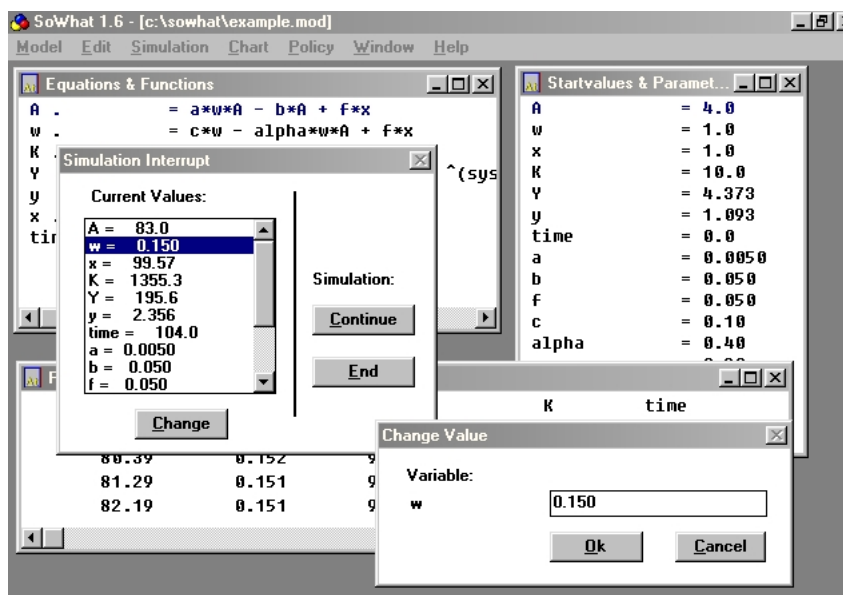


Abb. 2: Das Simulationseingriff - Fenster

Nach Auswahl der zu modifizierenden Variable bzw. Parameter im Fenster **Current Values** und Betätigung des Schalters **Change** öffnet sich das Fenster **Change Value**, in dem Sie die Änderungen vornehmen können. Auf diese Weise können Sie beliebig viele Parameter und Startwerte neu setzen und anschließend die Simulation mit **Continue** wieder aufsetzen. SOWHAT ist damit auf sehr einfache und effiziente Weise zur Politik-Simulation einsetzbar.

Eine weitere Besonderheit von SOWHAT ist die Möglichkeit zur Definition von *Parameter-Loops*, die es erlauben, eine Simulation mehrfach durchzuführen, wobei vom System ein Parameter oder Startwert in von Ihnen festgelegten Schritten verändert wird. Sollen im Beispiel mehrere Simulationen mit unterschiedlichem Parameter s durchgeführt werden, so wird im Fenster **Startvalues & Parameters** für s folgende Eingabe vorgenommen:

[0.1&0.4]0.05

Dieses bewirkt, dass das System nacheinander 7 Simulationsläufe durchführt, wobei der Parameter s sukzessive die Werte 0.1, 0.15, 0.2,...0.4 annimmt. In der ersten Zeile der protokollierten Charts steht dann der jeweilige Wert des Parameters s . Auf diese Weise ist es möglich, SOWHAT lange Zeit (z.B. nachts) arbeiten zu lassen, wobei Veränderungen im Modell einmalig festgelegt wurden und von SOWHAT zu gegebener Zeit selbständig durchgeführt werden.

Im **Window**-Menü können Sie die SOWHAT-Fenster öffnen und schließen und eine Funktionsleiste einblenden, über die Sie einen noch schnelleren Zugriff auf wichtige Systemfunktionen (z.B. Simulation starten und stoppen, Simulationseingriff, Kopieren der letzten Simulation, Modell laden und Modell speichern) erhalten.

Für weitere Informationen, Hilfestellungen und Anregungen steht Ihnen ein umfangreiches Hilfesystem zur Verfügung, welches Sie mit dem Befehl **Index** im **Help**-Menü aufrufen können.

Chapter 3

Installation und Hardware

Hardware und Betriebssystem

SOWHAT ist für IBM-kompatible PC's konzipiert, die mit den Betriebssystemen MS-DOS und MS-Windows ausgestattet sind. Minimale Voraussetzungen sind Rechner mit einem 80286 Prozessor und Windows 3.1.

Installation

Als freeware benutzen Sie eine auf den Namen Bachman und Strulik GbR vorinstallierte Version von SOWHAT. Dazu erstellen Sie ein separates Verzeichnis, z.B. C:\SOWHAT und extrahieren die Dateien aus `sowhat.zip` in dieses Verzeichnis. Klicken Sie danach mit der rechten Maustaste auf einen freien Platz des Windows Desktops und wählen Sie **NEU/Verknüpfung**. Geben Sie den Pfad zur `sowhat.exe` Datei in der Befehlszeile an, z.B. C:\SOWHAT\sowhat.exe.

Chapter 4

Allgemeine Hinweise zur Bedienung von SoWhat

Ähnlich vielen Textverarbeitungen und Tabellenkalkulationen unter Windows, ist SOWHAT eine MDI-Applikation. Das Kürzel MDI steht für multiple-document-interface, und bedeutet, dass mehrere Arbeitsbereiche gleichzeitig in Fenstern (im folgenden auch Arbeitsfenster genannt) dargestellt werden können. Anders als in den meisten MDI- Programmen, sind die SOWHAT-Arbeitsfenster nicht gleichartig, sondern jedes Fenster repräsentiert einen bestimmten Arbeitsbereich bzw. einen bestimmten Objekttyp des Programmes. Kurz ausgedrückt, die Gleichungen werden in einem Fenster eingeblendet, die Startwerte in einem anderen, die Resultate in einem dritten etc. Es werden 6 Arbeitsfenster unterschieden. Sie öffnen die einzelnen Arbeitsfenster durch die entsprechende Auswahl im Menü **Window**. Die ersten 6 Einträge dieses Menüs bewirken das Öffnen des jeweiligen Fensters. Ist ein Arbeitsfenster offen, wird der zugehörige Eintrag deaktiviert (grau dargestellt) und im unteren Teil des Pop-Up-Menüs erscheint der Name des Fensters. Über diesen dynamischen Eintrag kann das Fenster aktiviert werden.

Wir wollen der Reihe nach auf diese Fenster eingehen und danach die Manipulationsmöglichkeiten betrachten.

Die wesentlichen Objekte des Programmes sind **Modelle**. Ein Modell besteht zunächst aus den **Gleichungen** (unter dieser Bezeichnungen sind hier alle Differentialgleichungen, Differen-

zengleichungen, Substitute und Funktionen subsummiert). Alle Gleichungen des Modells stehen in Tabellenform untereinander, jede Gleichung erhält (implizit) einen Index, der ihre Position innerhalb des Modells angibt. Diese Tabelle wird im Arbeitsfenster **Equations & Functions** eingeblendet.

Die Gleichungen enthalten **Variablen** und **Parameter**, die ebenfalls in Tabellen- bzw. Listenform vorliegen, jede Variable und jeder Parameter besitzt eine eindeutige Position in dieser Liste, sowie einen Startwert. Diese Liste wird im Arbeitsfenster **Startvalues & Parameters** aufgeführt. Die Ergebnisse laufender Simulationen können optional im Arbeitsfenster **Results** eingesehen werden. Dieses Fenster ist im Gegensatz zu den anderen Arbeitsfenstern ein reines Ausgabefenster. Wenn Sie wider Erwarten keine Ergebnisse in diesem Fenster sehen, kontrollieren Sie, ob Sie den Variablen das Attribut *Recorded* verliehen haben (Dialog **Simulation Options**).

Mögliche Systemzustände werden im Arbeitsfenster **System Conditions** eingeblendet. Auch sie werden in Listenform dargestellt. In jeder Zeile wird links der Name des Systemzustands aufgeführt und anschließend die zugehörige Bedingung.

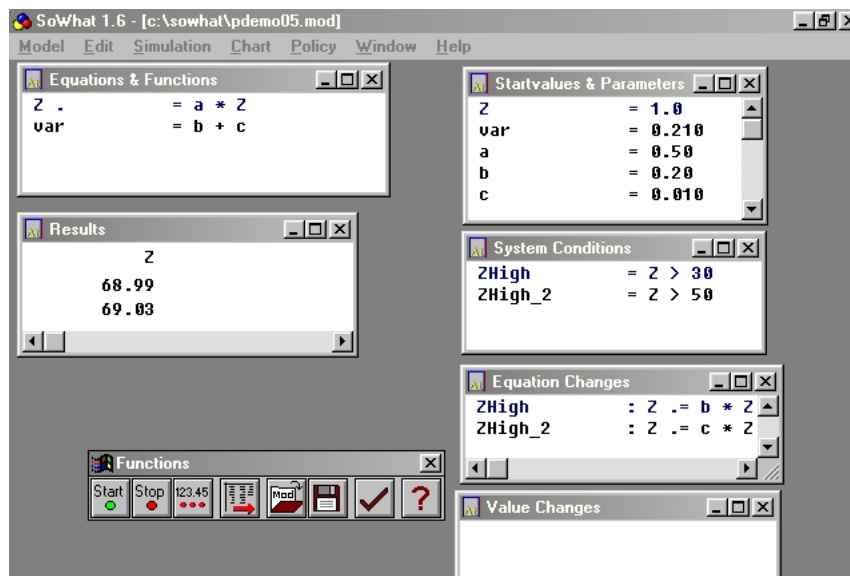


Abb. 3: Sämtliche SOWHAT-Fenster und die SOWHAT Funktionsleiste

Die definierten Wert- und Gleichungsänderungen werden in den Arbeitsfenstern **Value Changes**

bzw. **Equation Changes** gezeigt. In jeder Zeile wird die Zustandsverknüpfung, gefolgt von den Statements aufgeführt.

Die Arbeitsfenster lassen sich, auch während laufender Simulationen, Ein- bzw. Ausblenden, sowie in ihrer Größe und Ausdehnung durch die bekannten Bedienungen verändern. Durch Klicken und Halten der linken Maustaste können Sie innerhalb eines Fenster automatisch den sichtbaren Ausschnitt verschieben (automatic scroll), die Gleichung, über der die Maus positioniert ist, wird in der Farbe angezeigt, die Sie in der Windows-Systemsteuerung für Hervorhebung in Pop-Up-Menüs gesetzt haben. Wenn Sie ein Fenster soweit aufgezogen haben, dass der gesamte Fensterinhalt sichtbar ist, werden die horizontalen und vertikalen Schiebepalken ausgeblendet.

Durch Doppelklick der Maus auf einen Fenstereintrag, aktiviert SOWHAT den Änderungsdialog des jeweiligen Objekttypes und blendet den gewünschten Eintrag ein. Ein Doppelklick auf die zweite Gleichung im Fenster **Equations & Functions** bewirkt z.B. das Umschalten des Dialogs **Change Model** und das automatische Anzeigen der Gleichung in diesem Dialog.

In den Dialogen werden Sie oftmals aufgefordert, aus einer Liste einen oder mehrere Einträge zu selektieren. Um mehrere nebeneinanderstehende Einträge auszuwählen, drücken Sie die linke Maustaste, ziehen diese im gedrückten Zustand bis zum letzten gewünschten Eintrag und lassen die linke Maustaste wieder los. Um alle zwischen zwei Einträgen befindliche Einträge zu selektieren, klicken sie mit der linken Maustaste auf den oberen (unteren) der beiden Einträge, lassen die Maustaste los, gehen auf den unteren (oberen) der beiden Einträge und Betätigen gleichzeitig die Umschalttaste und die linke Maustaste. Wünschen Sie verstreut angeordnete Einträge auszuwählen, drücken Sie gleichzeitig die *Strg*-Taste und klicken mit der linken Maustaste die einzelnen Einträge an.

Wenn Sie eine laufende Simulation kurzzeitig anhalten wollen, drücken Sie die Alt-Taste. Der erste Menü-Eintrag des Hauptmenüs wird daraufhin invertiert dargestellt. Drücken Sie die Alt-Taste um mit der Simulation fortzufahren. Generell gilt, dass in den Bedienphasen (also auch bei Fenster-Manipulationen), die Simulationen angehalten werden. In den Eingabefeldern der Dialoge wird die Zwischenablage unterstützt.

Über das **Windows**-Menü können Sie auch die in Abbildung 3 dargestellte Funktionsleiste

aufrufen. Mit der Funktionsleiste haben Sie einen noch schnelleren Zugriff auf häufige SOWHAT-Befehle. Durch Mausklick auf den entsprechenden Schalter führt SOWHAT folgende Befehle für Sie aus (in der Abbildung 3 von links): Simulation Start, Simulation Stop, Simulation-Wert ändern, Kopiere letzte Simulation, Modell laden, Modell speichern, Konsistenzüberprüfung und Aufruf des SOWHAT-Hilfe-Systems.

Chapter 5

Das Modell - Menü

Im Menü **Model** - Menü können Sie neue Modelle editieren, gespeicherte Modelle laden und ändern, Modelle abspeichern und Ihre Modelldefinitionen in eine Textdatei schreiben. Mit **End** verlassen Sie SOWHAT.

5.1 Neue Modelle editieren

Mit **New** geben Sie den Befehl, nun ein neues Modell zu editieren. SOWHAT öffnet daraufhin das Fenster **Define Model**, das in Abbildung 1 dargestellt ist. In dem linken oberen Feld geben Sie den Namen des Modells ein. Im rechten oberen Feld wird die laufende Nummer der im mittleren Feld dargestellten Gleichung angezeigt. In diesem Feld geben Sie Ihre Gleichungen und Funktionen ein.

Mit **Insert** weisen Sie SOWHAT an, dass Sie zwischen zwei bestehenden Gleichungen eine dritte einfügen wollen. Mit **Delete** löschen Sie eine Gleichung aus Ihrem System. Um Teile einer Gleichung zu entfernen, markieren Sie diese mit dem Cursor und betätigen die *Entf*- bzw. *Del*-Taste auf Ihrer Tastatur. Mit der Tastenkombination *Shift Entf* (amerikanische Tastatur *Shift Del*) können Sie markierte Zeichenketten einer Gleichung in der Zwischenablage ablegen und diesen Text mit der Tastenkombination *Shift Einfg* (bzw. *Shift Ins*) an einer beliebigen Stelle Ihres Gleichungssystems einfügen. Jede Gleichungseingabe wird mit der *Return*- Taste oder mit **Next** abgeschlossen.

Mit **Previous** und **Next** können Sie sich durch Ihr dynamisches System bewegen, wobei die jeweils aktuelle editierbare Gleichung im mittleren Fenster angezeigt wird. Wollen Sie Modifikationen an einer eingegebenen Gleichung vornehmen, so müssen Sie nicht die komplette Gleichung neu eingeben. Mit der Maus oder den Pfeiltasten können Sie den Cursor direkt auf die Stelle in der Funktion setzen, an der Sie Änderungen vornehmen wollen. Mit **Cancel** beenden Sie den Editor-Dialog, ohne dass Ihre Eingaben Gültigkeit erlangen. Die vollständige Eingabe aller Ihrer Gleichungen und Funktionen bestätigen Sie mit **End**. Daraufhin ruft SOWHAT automatisch das Fenster **Startvalues & Parameters** auf.

5.2 Die Eingabe der Modellgleichungen

Die Eingabe der Modellgleichungen können Sie weitgehend so vornehmen wie in einer schriftlichen Notation auf Papier. Der *Lexical-Scan* übersetzt Ihre Gleichungen in das interne Format und überprüft die Gleichungen auf syntaktische Richtigkeit und mathematische Zulässigkeit. Ist Ihre Eingabe unzulässig, z.B. die Anzahl der öffnenden Klammern ungleich der Anzahl der schließenden Klammern, so erreicht Sie nach Eingabe von **End** eine Fehlermeldung. Die Art des Fehlers und die vermutliche Fehlerstelle innerhalb der fehlerhaften Gleichung werden angegeben. Die fehlerhafte Gleichung erscheint im Funktionsfeld des Fensters, so dass Sie die Korrekturen sofort vornehmen können. Eine Beschreibung sämtlicher Fehlermeldungen mit Hinweisen zu Ihrer Behebung finden Sie in Kapitel 11.

Nun einige beispielhafte Erläuterungen zur Eingabe Ihrer Gleichungen. Die erste Ableitung einer Variablen nach der Zeit wird durch einen dem Variablennamen folgenden Punkt dargestellt. Die Differentialgleichung

$$\frac{\partial x}{\partial t} = \dot{x} = y$$

geben Sie also ein als

$$x. = y$$

Ableitungen sind nur auf der linken Seite des Gleichheitszeichens möglich.

Bei allen Gleichungen und Funktionen darf auf der linken Seite des Gleichheitszeichen jew-

eils nur eine Variable eingegeben werden. Damit der *Lexical Scan* Sie versteht, müssen Sie also möglicherweise die auf dem Papier vorgenommene Modellnotation durch Einführung von Hilfsvariablen modifizieren. Die Differentialgleichung

$$\dot{x} = y - \dot{y}$$

wird vom *Lexical Scan* verstanden, wenn Sie als System

$$y. = hilf$$

$$x. = y - hilf$$

eingegeben wird. Zulässig sind nur Differentialgleichungen erster Ordnung. Differentialgleichungen höherer Ordnung müssen als Differentialgleichungssystem erster Ordnung formuliert werden. Die Differentialgleichung zweiter Ordnung

$$\frac{\partial^2 x}{\partial t^2} = \ddot{x} = \sin y + b$$

können Sie z.B. eingeben als

$$x. = u$$

$$u. = \sin(y) + b$$

Differenzgleichungen erster Ordnung geben Sie durch ein dem Gleichheitszeichen vorgestelltes &-Zeichen ein. Um z.B. die Differenzgleichung

$$x_{t+1} = a + bx$$

zu definieren, geben Sie

$$x\& = a + b * x$$

ein. Auch hier müssen Sie – wie bei den Differentialgleichungen – Differenzgleichungen höherer Ordnung in Form eines Systems von Differenzgleichungen erster Ordnung eingeben. Anson-

sten können die Differenzgleichungen beliebig komplex sein. SOWHAT ermöglicht die Simulation gemischter Differenzen-Differentialgleichungssysteme. Beachten Sie jedoch, dass hierfür der **Step Record** der Ausgabeschrittweite (vgl. Kapitel 6.1) auf Eins (1) gesetzt sein sollte, damit Sie sinnvolle Ergebnisse erhalten.

Die Reihenfolge der Gleichungen und Funktionen ist beliebig. Es können maximal 32 Differentialgleichungen bzw. Funktionen eingegeben werden. Hierbei dürfen maximal 128 Variablen und Parameter verwendet werden. Zur besseren Übersicht können beliebig Leerzeichen gesetzt werden, die vom *Lexical Scan* unterdrückt werden. Die Eingaben

$$x. = y * x - a * \sin(b - c * z) \quad \text{und} \quad x. = y * x - a * \sin(b - c * z)$$

liefern z.B. identische Ergebnisse. Der *Lexical Scan* ist groß-kleinschreibungs-sensitiv. In der Gleichung

$$A. = b + a$$

werden A und a als verschiedene Parameter bzw. Variable des Systems interpretiert. Ein Multiplikationszeichen ist zwingend vorgeschrieben. Die Verletzung dieser Regel ist erfahrungsgemäß bei an wissenschaftliche Notation gewöhnten Benutzern die häufigste Ursache syntaktischer und logischer Fehler. Wird z. B. die Gleichung

$$K. = sY$$

eingegeben, so interpretiert der *Lexical Scan* sY als exogenen Parameter des Systems. Wird jedoch tatsächlich eine Multiplikation von s und Y gewünscht, so muss die richtige Eingabe

$$K. = s * Y \tag{5.1}$$

lauten.

SOWHAT stellt Ihnen folgende mathematischen Operationen zur Verfügung:

Syntax	Funktion
+*/	arithmetische Standardoperatoren
sin(x)	Sinus im Bogenmaß der Zahl x
cos(x)	Cosinus im Bogenmaß der Zahl x
lg(x)	Decadischer Logarithmus
ln(x)	Natürlicher Logarithmus
rnd(x)	x gerundet
^(b,e)	Exponentiation der positiven Basis b mit dem Exponenten e
$\sqrt{}$ (x)	Quadratwurzel der positiven Zahl x
min(x,y)	Minimum von x,y
max(x,y)	Maximum von x,y
sgn(x)	Vorzeichen der Zahl x . Liefert die Zahl +1 oder die Zahl -1
trnc(x)	Ganzzahliger Teil der Zahl x
iexp(x)	Exponentiation der positiven oder negativen Zahl x mit dem Exponenten y , wobei y auf den nächsten ganzzahligen Wert gerundet wird

Sie können auf folgende Systemvariable zugreifen:

Variable	Inhalt
dtime	Zeit seit Beginn der Simulation
acttime	Aktuelle Zeit
sysE	Eulersche Zahl
sysPi	Kreiszahl PI

Beispiele.

Um die Gleichung

$$Y = A^\beta K^\gamma$$

in Ihrem dynamischen Modell zu berücksichtigen, geben Sie

$$Y = \gamma(A, beta) * \gamma(K, gamma)$$

ein. Wollen Sie die Gleichung

$$A = e^{\alpha t} A_0$$

in Ihrer Simulation verwenden, so können Sie

$$A = \text{\textasciitilde}(sysE, alpha * dtime) * AStart$$

eingeben. Sie können aber auch die Formulierung als Differentialgleichung

$$A. = alpha * A$$

oder als Differenzengleichung

$$A\& = (1 + alpha) * A$$

verwenden.

Die maximale Länge einer Gleichung oder Funktion darf *255 Zeichen* nicht überschreiten. Die Variablennamen dürfen maximal *12 Zeichen* lang sein. Akzeptiert der Lexical Scan eine Eingabe über 255 Zeichen nicht, so können Sie entweder die Länge der Variablennamen verkürzen oder einzelne Terme der Gleichung durch Einführung von Hilfsvariablen in separate Substitute ausgliedern. **Substitute** sind Funktionen, die in jedem Integrationsschritt (zeitlich) als erstes berechnet werden. Sie ermöglichen das Substituieren zusammenhängender mathematischer Ausdrücke in Differentialgleichungen, Differenzengleichungen und Funktionen. Substitute werden durch ein dem Gleichheitszeichen vorgestellten Doppelpunkt (:) gekennzeichnet. Die Gleichung

$$\begin{aligned} dincome/dt = & (\alpha + \gamma gcapital - (1 - \beta - \gamma) \\ & \cdot ((fertopt + maxfert / (1 + e^{a(income-yf)}))) \\ & - (mortnat + maxmort / (1 + e^{b(income-ym)}))) \cdot income \end{aligned}$$

kann z.B. durch den Gebrauch von Substituten auf übersichtliche Weise folgendermaßen eingegeben

werden:

$$income. = (\alpha + \gamma * gcapital - (1 - \beta - \gamma) * gpopulation) * income$$

$$gpopulation := fertility - mortality$$

$$fertility := fertopt + maxfert/term1$$

$$term1 := 1 + (\gamma_{sysE, a} * (income - yf))$$

$$mortality := mortnat + maxmort/term2$$

$$term2 := 1 + (\gamma_{sysE, b} * (income - ym))$$

Weitere Beispiele finden Sie auf Ihrer SOWHAT-Diskette.

5.3 Bestimmung der Start- und Parameterwerte

Sind alle Ihre Modellgleichungen richtig eingegeben, so öffnet sich nach Abschluss der Eingabe mit **End** das Fenster **Startvalues & Parameters**, welches in Abbildung 4 dargestellt ist. Im linken oberen Feld wird die laufende Nummer der Variablen bzw. des Parameters angegeben. Anschließend folgt der von ihnen festgelegte Variablenname. In dem **Value** - Feld können Sie den Startwerten und Parametern Ihres Systems Zahlen zuweisen. Es gilt die amerikanische Notation: Nachkommastellen werden dabei durch einen Punkt (.) von Vorkommastellen getrennt. Der *Lexical Scan* hat die endogenen Variablen des Modells erkannt und weist diese mit **endogen** aus. Diese Variablen sind nicht editierbar.

Jede Zahleneingabe wird mit der *Return*-Taste bzw. mit **Next** abgeschlossen. Mit **Previous** und **Next** bewegen Sie sich durch die Variablenliste und können Änderungen der zugewiesenen Zahlen vornehmen.

SOWHAT liefert die Möglichkeit zur Definition von *Parameter-Loops*, die es erlauben eine Simulation mehrfach durchzuführen, wobei vom System ein Parameter oder Startwert in von Ihnen festgelegten Schritten verändert wird. Sollen im Beispiel mehrere Simulationen mit unterschiedlichem Parameter s durchgeführt werden, so wird im Fenster **Startvalues & Parameters** für s folgende Eingabe vorgenommen:

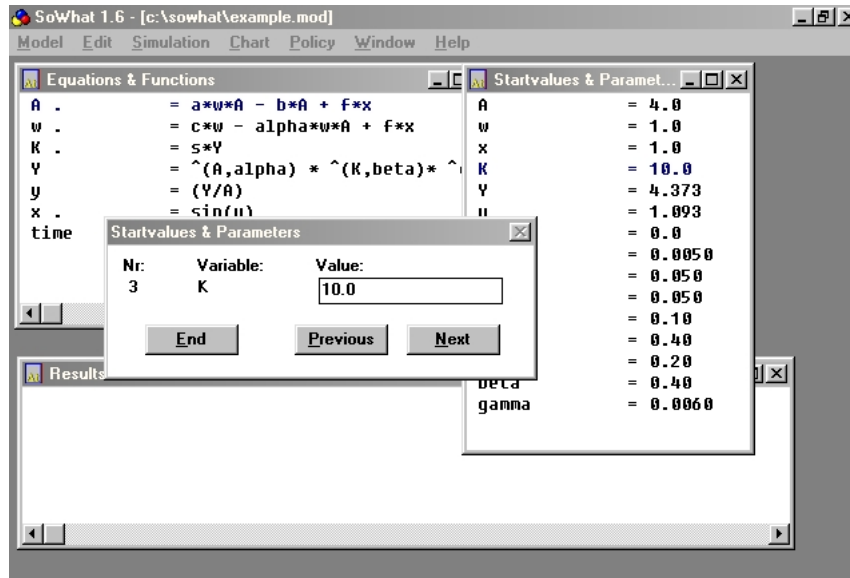


Abb. 4: Das Startwerte & Parameter - Fenster

[Startwert & Endwert] Schrittweite

Die Eingabe von

[0.1 & 0.4]0.05

bewirkt z.B., dass das System nacheinander 7 Simulationsläufe durchführt, wobei der Parameter s sukzessive die Werte 0.1, 0.15, 0.2,...,0.4 annimmt. In der ersten Zeile der protokollierten Charts steht dann der jeweilige Wert des Parameters s . Auf diese Weise ist es möglich, SOWHAT lange Zeit (z.B. nachts) arbeiten zu lassen, wobei Veränderungen im Modell einmalig festgelegt wurden und von SOWHAT zu gegebener Zeit selbständig durchgeführt werden. Pro Modell ist maximal eine *Parameter-Loop* erlaubt.

Weiterhin haben Sie die Möglichkeit, im **Startvalues & Parameters**-Fenster eine stochastische Verteilung für exogene Modellvariablen zu definieren und damit Ihr dynamisches System einem stochastischen Prozess zu unterwerfen. SOWHAT erzeugt dann in jedem Simulationsschritt beliebige, unabhängige normal-, rechteck- oder exponential-verteilte Zufallsvariablen. Um z.B. den Modellparameter *zufall* als normalverteilte stochastische Variable mit Mittelwert 0,5 und Standardabweichung 1,3 zu definieren geben Sie für *zufal* im **Startvalues & Parameters** Fenster

N (0.5 , 1.3)

ein. Eine im Intervall [- 12,34 77] rechteckverteilte Zufallsvariable *zufall2* definieren Sie mit

R (-12.34 , 77)

Exponentialverteilte Zufallsvariable werden in der Notation E(<L>,<X>) im Startwertedialog definiert. L ist der Parameter der Verteilung, X wird bislang nicht ausgewertet, sollte aber den Wert 1 erhalten.

Sind alle Startwerte und Parameter des Systems korrekt bestimmt, so beenden Sie mit **End**. Ihr Modell ist nun vollständig spezifiziert. Es sollte nun abgespeichert werden.

5.4 Speichern von Modellen

Beim erstmaligen Speichern eines Modells öffnet sich durch Anwahl des Befehls **Save** im **Model**-Menü das **Save Model**-Fenster. Im **Filename**-Feld legen Sie den Namen der Datei fest unter dem das Modell abgespeichert werden soll. SOWHAT schlägt die Dateierweiterung **.mod** vor. Sie können einen neuen Modellnamen eingeben oder durch Anwahl einer aufgeführten Datei diese mit dem abzuspeichernden Modell überschreiben. Im Fenster **Directories** wählen Sie das Verzeichnis aus, in dem Sie das Modell ablegen wollen. Das aktuelle Verzeichnis wird Ihnen unterhalb des **filename**-Feldes mitgeteilt. Zur Ausführung des Abspeicherns wählen Sie **Ok**. Andernfalls schließen Sie das Fenster mit **Cancel**.

SOWHAT speichert mit dem eigentlichen Modell auch stets die Simulationsumgebung des Modells mit, d.h. auch die aktuellen Startwerte und Parameter, die eingestellten Simulationsoptionen und die Fensterinformationen werden mitgespeichert und sind beim Laden des Modells sofort verfügbar. Ist eine Modelldatei angelegt, so wird beim zweiten und allen folgenden Aufrufen von **Save** das **Save Model**-Fenster nicht geöffnet und die angegebene Datei mit dem aktuellen Modell überschrieben. Wünschen Sie dieses nicht, z.B. weil Sie nach einer Veränderung der Modellgleichungen der Parameterwerte oder der Simulationsoptionen sowohl die ursprüngliche Modellfassung als auch das modifizierte Modell als eigenständige Dateien halten wollen, so wählen Sie den Befehl **Save As**. Daraufhin öffnet erneut das **Save Model** - Fenster, in dem Sie Dateiname und Pfad wie oben beschrieben neu festlegen können.

5.5 Laden von Modellen

Mit dem Befehl **Load** im **Model** - Menü öffnen Sie das **Load Model**- Fenster. Im **filename**-Feld geben Sie den Namen der zu öffnenden Datei ein, in der das Modell und seine Simulationsumgebung abgelegt ist. Alternativ können Sie die Datei durch Anwahl eines im **Files**-Fenster aufgeführten Dateinamens laden. Im Fenster **Directories** wählen Sie das Verzeichnis, in dem Sie das zu ladende Modell abgespeichert haben, aus. Das aktuelle Verzeichnis wird Ihnen unterhalb des **filename**-Feldes mitgeteilt. Zur Ausführung des Ladens wählen Sie **Ok**. Andernfalls schließen Sie das Fenster mit **Cancel**.

Falls Sie durch den Ladevorgang ein nicht abgespeichertes aktuelles Modell überschreiben werden, fragt Sie SOWHAT, ob dieses wirklich geschehen soll, so dass Sie die Möglichkeit haben, vor dem Laden Ihr gerade aktuelles Modell zu sichern.

Die letzten vier von Ihnen bearbeiteten Modelle werden am Ende des **Model** Pop-up- menüs angegeben. Diese Modelle können Sie direkt durch Anwahl des Eintrags mit der Maus laden.

5.6 Ändern von Modellen

Durch den Befehl **Change** im Modell- Menü teilen Sie SOWHAT mit, dass Sie das aktuelle Modell modifizieren möchten. Daraufhin öffnet sich das **Change Model**-Fenster. Der Dialog in diesem Fenster ist identisch mit dem Dialog im **Define Model**-Fenster, so dass an dieser Stelle auf die Abschnitte 4.1 bis 4.3 verwiesen werden kann.

Es gibt jedoch eine noch einfachere Möglichkeit, um Modelländerungen vorzunehmen: Durch einen Doppelklick auf eine Gleichung im Fenster **Equations & Functions** öffnet sich das **Change Model**-Fenster und die angewählte Funktion erscheint sofort im Feld **Equations**. Durch Doppelklick auf einen im **Startvalues & Parameters**-Fenster ausgewiesenen Wert wird dieser sofort editierbar.

5.7 Schreiben von Modellen

Mit dem Befehl **Write** im **Model**-Menü können Sie Ihr vollständiges Modell (Gleichungen und Funktionen, Startwerte und Parameter, Simulationsoptionen und Politikdefinitionen) als ASCII-Datei abspeichern. Nach Anwahl des Befehls öffnet SOWHAT das **Write Model**-Fenster. Wie beim Abspeichern der Modelle geben Sie im **filename**-Feld den Namen der Textdatei ein, in der Ihr Modell abgespeichert werden soll. Hierbei schlägt SOWHAT die Endung **.doc** vor. Im Fenster **directories** wählen Sie Laufwerk und Verzeichnis in dem Sie die Datei abspeichern wollen. Die ASCII-Textdatei können Sie anschließend mit jedem ASCII-fähigen Editor oder Textverarbeitungsprogramm laden und dort formatieren, ausdrucken oder in andere Texte einbinden.

Chapter 6

Das Bearbeiten - Menü

Die Befehle des Menüs **Edit** dienen dem Datentransfer über die Windows - Zwischenablage in andere Windows-Applikationen. Mit dem Befehl **Copy Model** können Sie Ihre Modell-Definition (Gleichungen, Funktionen, Startwerte, Parameter, Simulationsoptionen, Zustandsdefinitionen und Politik) in die Zwischenablage kopieren und anschließend in Ihrem Editor und Textverarbeitungsprogramm weiterverarbeiten. Dort wählen Sie den Befehl Bearbeiten Einfügen (bzw. Paste o.ä.) oder Betätigen die Tasten *Shift* und *Einfg* (bzw. *Ins*), um das Modell in Ihren Text einzufügen, weiterzuverarbeiten und auszudrucken.

Wenn Sie den Befehl **Copy Charts** anwählen, so öffnet SOWHAT das **Copy Charts**-Fenster. In einer Liste zeigt Ihnen SOWHAT die im Arbeitsspeicher Ihres Rechners vorhandenen Tabellen. Eine Tabelle wird dabei durch den Variablennamen der Variablen, dessen Protokoll sie enthält und die laufende Nummer (beginnend bei Null) der Simulation identifiziert. In dieser Liste markieren Sie jene Ergebnistabellen der Protokollvariablen, die Sie in die Zwischenablage kopieren wollen. Mehrere zusammenhängende Tabellen markieren Sie durch drücken der linken Maustaste und Ziehen der Maus. Nicht zusammenhängende Tabellen können durch Festhalten der Taste *Strg* (bzw *Control*) während des Markierungsvorgangs ausgewählt werden. Sind alle gewünschten Tabellen ausgewählt, so bestätigen Sie mit **Ok**. In Ihrer Tabellenkalkulation oder Datenbank wählen Sie den Befehl Bearbeiten Einfügen (bzw. Paste o.ä.) oder Betätigen die Tasten *Shift* und *Einfg* (bzw. *Ins*), um die Ergebnistabellen in Ihr Spreadsheet o.ä. einzufügen. Mit dem Befehl **Copy Last Simulation** kopieren Sie alle Ergebnistabellen der Protokollvariablen des

letzten Simulationslaufes in die Windows - Zwischenablage. Dieser Befehl ist besonders für die schnelle und effiziente Auswertung der gerade simulierten Ergebnisse zu empfehlen.

Wenn Sie die Daten mit Software verarbeiten, welche einen Dezimalpunkt verwendet, so deaktivieren Sie den Schalter **Comma**. Wenn Sie nicht wünschen, dass SOWHAT den Variablenbezeichner und die Simulationsnummer als Spaltenkopf mitkopiert, so deaktivieren sie den Schalter **Variable**.

Chapter 7

Das Simulation - Menü

Das Simulationsmenü enthält die Befehle **Options**, **Start**, **Change Value** und **Halt** und die Schalter **Range Check** und **Error Relative**. Mit **Start** beginnen Sie die Simulation eines vollständig (einschließlich der Simulationsoptionen) spezifizierten Modells und mit **Halt** veranlassen Sie die Beendigung der Simulation vor dem von Ihnen vordefinierten Simulationsende. Die Befehle **Change Value** und **Halt** sind inaktiv wenn zur Zeit keine Simulation läuft.

7.1 Festlegen des Simulationszeitraumes und der Simulationsschrittweiten

Mit dem Befehl **Options** öffnen Sie das Fenster, in dem Sie die Simulationsoptionen festlegen können. Dieses ist in Abbildung 5 dargestellt.

Im Feld **Start** geben Sie den Startzeitpunkt Ihrer Simulation ein und im **End**-Feld legen Sie den Endzeitpunkt der Simulation fest. Im Feld **Record** legen Sie die Schrittweite fest, mit der die zu protokollierenden Variablen von SOWHAT mitgeschrieben werden und später als Tabellen verarbeitet werden können. Stellt man sich Startzeit und Endzeit in der Dimension Jahr vor, so bedeutet die Eingabe von 1 (bzw. 10) im **Record**-Feld, dass die Variablen in Form von Jahres- (Zehnjahres-) Endwerten mitprotokolliert werden.

Numerische Simulationen liefern niemals die exakte Lösung eines Differentialgleichungssys-

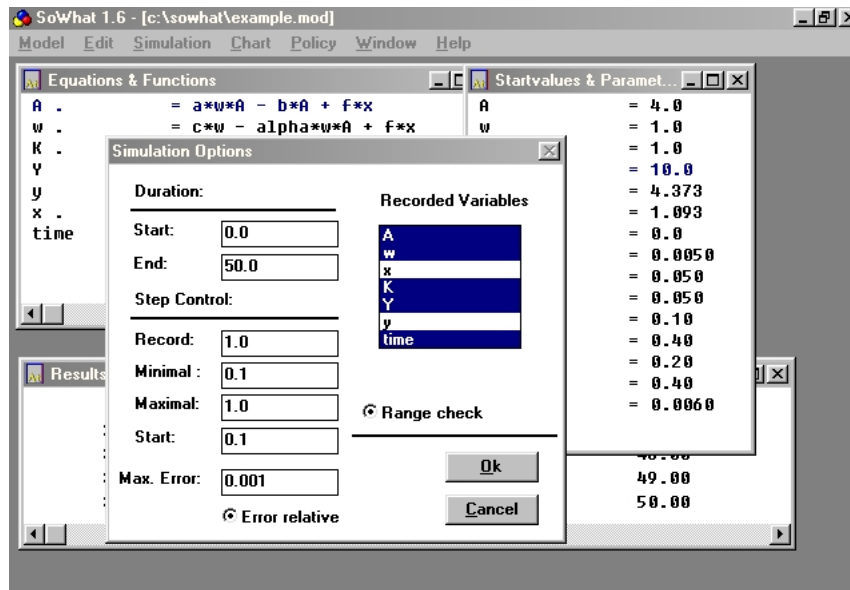


Abb. 5: Das Simulationsoptionen-Fenster

tems sondern nur Näherungen an die tatsächliche Lösung. Mit **Minimal**, **Maximal**, **Start** und **Max. Fehler** können Sie die interne Schrittweite der Simulation und damit die Genauigkeit der Approximation (und natürlich auch die Simulationsgeschwindigkeit) bestimmen.

SOWHAT verwendet zur numerischen Simulation der Differentialgleichungen ein vierstufiges Runge-Kutta-Verfahren mit variabler Schrittweitensteuerung. Nach jedem internen Berechnungsschritt wird für jede Differentialgleichung der Fehler der numerischen Lösung durch einen sechsstufigen Runge-Kutta-Algorithmus ermittelt.

Mit **Start** legen Sie die interne Schrittweite des Verfahrens fest, die zu Beginn der Simulation Gültigkeit hat. Unter **Max. Fehler** definieren Sie den maximal zulässigen Fehler, der entscheidend von der internen Schrittweite der Integration abhängt. Bei jedem internen Rechenschritt überprüft SOWHAT, ob der berechnete Fehler größer oder kleiner als der maximal zulässige Fehler ist. Ist der Fehler größer, so wird die interne Schrittweite des Verfahrens reduziert. Diese Schrittweitenverminderung findet jedoch nur bis zu der von Ihnen im Feld **Minimal** spezifizierten minimalen Schrittweite statt. Ist der berechnete Simulationsfehler kleiner als die Hälfte des maximal zulässigen Simulationsfehlers **Max. Fehler**, so wird die interne Schrittweite vergrößert, und das Verfahren beschleunigt sich. Die maximal zulässige interne Schrittweite legen

Sie mit **Maximal** fest. Da Sie i.d.R. sowohl an einer hohen Simulationsgenauigkeit als auch an einer hohen Simulationsgeschwindigkeit interessiert sind, müssen Sie hier in Abhängigkeit von den Eigenschaften des zu simulierenden Systems (linear, quasi-linear, nicht-linear, stabil, instabil, zyklisch, explosiv) versuchen, die für Sie optimale Lösung zu erreichen. Unter Umständen müssen Sie mehrere Simulationsläufe mit unterschiedlich eingestellten Optionen durchführen, um ein für Sie befriedigendes Ergebnis zu erhalten. Die zusätzliche Verantwortung hinsichtlich der Güte Ihrer Simulationen wird mit einer hohen Flexibilität belohnt, die Sie in anderen Simulationsprogrammen und in mathematischen Tools nicht finden.

Im Fenster **Recorded Variables** markieren Sie die Variablen des Systems, die von SOWHAT während der Simulation mitgeschrieben werden und später in Form von Tabellen abgespeichert werden können und zur Weiterverarbeitung (zum Beispiel zur grafischen Darstellung) zur Verfügung stehen. Die Entwicklung der markierten Variablen wird während der Simulation im Fenster **Results** angezeigt.

Haben Sie Ihre Simulationsoptionen vollständig spezifiziert, so schließen Sie mit **Ok** ab. Entscheiden Sie sich jedoch, die zuvor eingestellten Optionen weiter zu verwenden, so wählen Sie den **Cancel**-Befehl.

7.2 Bereichsüberprüfung und Relativer Fehler

Zur Bereichsüberprüfung veranlassen Sie SOWHAT, indem Sie entweder durch Mausklick den runden Schalter **Range Check** im **Options**-Fenster auf *Ein* (schwarze Markierung) schalten oder (auch während laufender Simulationen) **Range Check** im **Simulation**-Pop-Up-Menü anwählen. Ein Haken links neben dem Schalter zeigt Ihnen, dass der Befehl zur Zeit aktiv ist.

In jedem Rechner ist die Möglichkeit der internen digitalen Zahlendarstellung begrenzt. Insbesondere dynamische System sind jedoch häufig dergestalt beschaffen, dass diese Grenzen verletzt werden. In vielen dynamischen Systemen streben Variablen (in exponentieller Weise !) gegen Unendlich. Andere Variablen, die möglicherweise als Divisor in eine Gleichung eingehen, nähern sich asymptotisch Null. Normalerweise führt ein solcher numerischer Überlauf zum Ab-

sturz des Programms. Die bis zum Absturz errechneten Daten gehen verloren. Die Ursache des Systemzusammenbruches bleibt verborgen und Spekulationen oder eigener Rechenarbeit überlassen. Wenn Sie jedoch den Schalter **Range Check** eingeschaltet haben, so berechnet SOWHAT bei jeder Rechenoperation, ob das ermittelte Ergebnis außerhalb der Möglichkeiten der digitalen Zahlendarstellung liegen würde. Ist dieses der Fall, so wird die Simulation mit einer Fehlermeldung beendet. In dieser Fehlermeldung werden Sie informiert, durch welche Rechenoperation bei welcher Variablen im folgenden Simulationsschritt ein numerischer Überlauf stattfinden würde, der zum Absturz Ihres Rechners führen würde. Außerdem sind sämtliche Wertetabellen des Variablenprotokolls erhalten. Diese können Sie abspeichern und so die Systementwicklung bis zum Simulationsstopp nachvollziehen. Da durch die Bereichsüberprüfung jeder Rechenschritt mehrfach durchgeführt wird, verlangsamt sich natürlich die Simulationsgeschwindigkeit erheblich. Deshalb sollten Sie, sobald Sie sicher sind, dass das zu simulierende Modell keine numerischen Überläufe generiert, die Bereichsüberprüfung ausschalten.

Durch Betätigung des Schalters **Error relative** im **Options**-Fenster oder (auch während laufender Simulationen) durch den entsprechenden Befehl im **Simulation**-Pop-Up-Menü veranlassen Sie SOWHAT bei der internen Schrittweitenkontrolle den relativen Simulationsfehler zugrunde zu legen. Der geschätzte Fehler in jedem Integrationsschritt, kann absolut oder relativ zur Steuerung der Schrittweite verwendet werden. Wird der absolute Fehler zur Grundlage der Schrittweitensteuerung gemacht, wird pro Iteration die euklidische Vektornorm der Einzelfehler als Fehlerabschätzung betrachtet. Haben Sie **Error relative** aktiviert, so werden die Einzelfehler je Iteration durch den Wert der dazugehörigen Variable dividiert und dann die Quadratwurzel der aufsummierten relativen Einzelfehlern gebildet. Erreichen Ihre Variablenwerte große Zahlenbereiche, können Sie durch Einschalten dieser Option die Simulationsgeschwindigkeit erheblich erhöhen. Einbußen in Punkto Genauigkeit, können Sie durch sinnvolle Wahl der Steuerungsparameter **Max. Fehler** und **Minimal** ausgleichen.

7.3 Simulationseingriff und Werte Ändern

Im Gegensatz zu numerischen Tools zur Lösung von dynamischen Systemen bietet Ihnen SOWHAT die Möglichkeit des Eingriffs und der Manipulation laufender Simulationen. Auf diese Art und

Weise können Sie SOWHAT im Bereich der Politiksimulation und als Plattform für von Ihnen entworfene Computerplanspiele (auch in der Lehre) benutzen. Während der laufenden Simulation können Sie die Ergebnisse im **Results**-Fenster beobachten und, falls es Ihnen angebracht erscheint, in das Simulationsgeschehen mit dem Befehl **Change Value** eingreifen. Daraufhin öffnet sich der in Abbildung 2 dargestellte **Simulation Interrupt**-Dialog. In einer Listbox sehen Sie alle momentanen Werte Ihrer Modellvariablen und Parameter. Sie können nun einen Modellparameter oder eine dynamische Variable markieren und den Befehl **Change** wählen (natürlich sind die endogenen Variablen Ihrer Funktionen nicht änderbar). SOWHAT verzweigt daraufhin in den in Abbildung 2 dargestellten **Change Value**-Dialog, indem Sie Ihrer Variablen oder Parameter durch Eingabe einen neuen Wert zuweisen.

Nachdem Sie die Eingabe des neuen Wertes mit **Ok** bestätigt haben, kehrt SOWHAT in den **Simulation Interrupt**-Dialog zurück und Sie können die Änderungsprozedur für beliebig viele Variablen und Parameter wiederholen. Sind alle Änderungen abgeschlossen, so betätigen Sie den Schalter **Continue**, und SOWHAT setzt die Simulation mit den neuen Parameterwerten an der aktuellen Abbruchstelle fort. Falls Sie sich entschließen, die Simulation doch ganz abzubrechen und einen vollständig neuen Simulationslauf zu beginnen, so betätigen Sie den Schalter **End**.

Neben dieser Möglichkeit dieses spontanen, diskretionären Eingreifens in aktuelle Simulationen können Sie mit SOWHAT auch regelgebundene Politiken definieren, in denen SOWHAT automatisch in Abhängigkeit von den von Ihnen vordefinierten Systemzuständen Werte und ganze Gleichungen laufender Simulationen modifiziert. Lesen Sie hierzu das Kapitel 8 des Handbuchs.

Chapter 8

Das Tabellen - Menü

Im **Chart**-Menü können Sie die von SOWHAT bereitgestellten Tabellen, die das Simulationsergebnis enthalten, speichern, löschen und ihren Inhalt einsehen.

SOWHAT liefert Ihnen nur das Ergebnis Ihrer Simulation. Die grafische Darstellung und statistische Auswertung der Ergebnisse überlassen wir anderen Programmen, die hierfür maßgeschneiderte professionelle Lösungen anbieten. Es empfiehlt sich, Programme zu verwenden, die auch als Windows-Applikation vorliegen, wie z.B. MS-Excel oder Lotus-123. In diesem Falle, sollten Sie, um schnell und effizient zu arbeiten, für den Datentransfer die Windows-Zwischenablage nutzen. (Vgl. die Ausführungen im Kapitel zum **Bearbeiten**-Menü). Hierdurch vermeiden Sie das Anhäufen sinnloser Datenberge auf Ihrer Festplatte. Den **Save**-Befehl sollten Sie i.d.R. nur benutzen, wenn Sie Ihre Ergebnisse mit Programmen, die die Zwischenablage nicht unterstützen, verarbeiten.

SOWHAT kann maximal 400 Tabellen gleichzeitig im Arbeitsspeicher halten, sofern genügend Speicherkapazität vorhanden ist. Die Länge einer Tabelle ist beliebig. Lässt Ihre Hauptspeicherkapazität dieses zu, so können Sie also in einer Tabelle die Entwicklung einer Variablen über die nächsten 10 Millionen Jahre in Jahreswerten mitprotokollieren. Beachten Sie jedoch die Restriktionen, die Ihnen das Tabellenkalkulationsprogramm oder die Datenbank stellt, mit der Sie die Tabellen weiterverarbeiten wollen. So dürfen z.B. in MS-Excel die Tabellen eine maximale Länge von 4000 Zeilen besitzen, falls eine grafische Auswertung gewünscht wird.

8.1 Tabellen speichern

Wenn Sie den Befehl **Chart Save** wählen, öffnet sich das in Abbildung 6 dargestellte **Save Charts**-Fenster. In der Liste **Select Charts** zeigt Ihnen SOWHAT die im Arbeitsspeicher Ihres Rechners vorhandenen Tabellen. Eine Tabelle wird dabei durch den Variablenname der Variablen, dessen Protokoll sie enthält und die laufende Nummer (beginnend bei Null) der Simulation identifiziert.

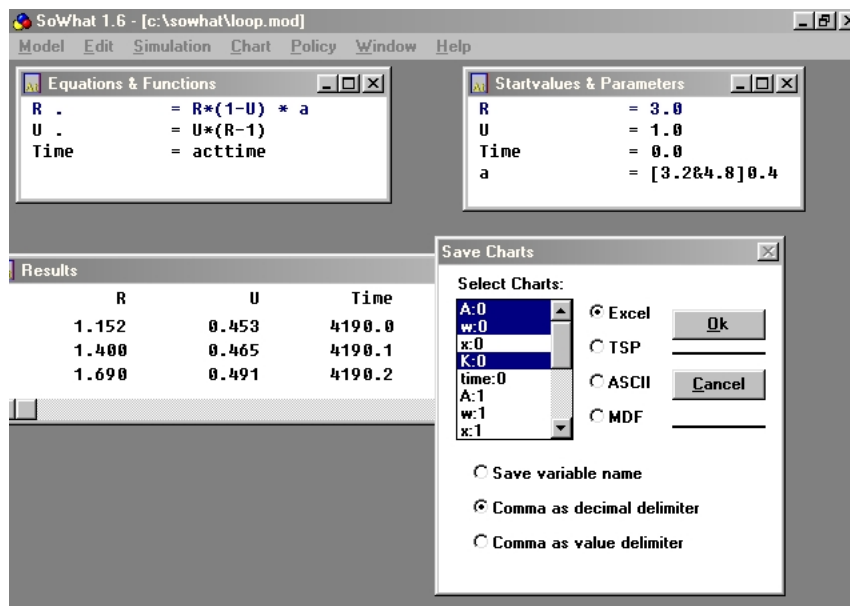


Abb. 6: Das Tabellen Speichern-Fenster

Zunächst teilen Sie SOWHAT mit, welche Variablenprotokolle Sie zusammen in einer Datei abspeichern möchten. Hierfür markieren Sie die abzuspeichernden Variablenprotokolle. Im Beispiel sind die Ergebnistabellen der Variablen A, w und K Tabellen des ersten Simulationlaufes zum Speichern ausgewählt. Mehrere zusammenhängende Tabellen markieren Sie durch Drücken der linken Maustaste und Ziehen der Maus. Nicht zusammenhängende Tabellen können durch Festhalten der Taste *Strg* (bzw *Control*) während des Markierungsvorgangs ausgewählt werden. Anschließend legen Sie das Dateiformat fest, indem Sie den weißen Schalter vor den von SOWHAT bereitgestellten Formaten schwarz markieren. SOWHAT stellt die Dateiformate Excel, TSP, ASCII und das interne Datenformat Milestones-Data-Format zur Auswahl.

Wählen Sie Excel, so wird die Datei im MS-Excel-Datenformat **.xls** angelegt. Um die Tabellen in Excel zu laden, ist dort die Voreinstellung *Spaltenabgrenzung Tabstop* und *Dateiursprung Windows (Ansi)* zu wählen. Diese Einstellung ist in deutschen Excel-Versionen die Standardeinstellung. Beim Speichern wechselt SOWHAT den Delimiter der Vor- und Nachkommastellen, so dass Excel Ihre Daten auch als Zahlen versteht.

SOWHAT unterstützt das TSP-Format und schlägt hierfür die Endung des TSP-Header-Files **.h** vor. In TSP können die Daten mit dem *Load*-Befehl geladen werden. In vielen Tabellenkalkulationen und Datenbanken ist es möglich, Daten im ASCII-Format einzulesen. SOWHAT schlägt für diese Dateien die Endung **.txt** vor. Um zum Beispiel SOWHAT-Tabellen in Lotus 123 zu laden, speichern Sie die Daten als ASCII-Datei und erteilen in Ihrem Lotus-Programm den Befehl *Transfer Fremd*. Das Milestones-Data-Format **.mdf** dient dem Datenaustausch mit anderen Applikationen von Milestones Scientific Software. Sie können beliebig viele Protokolltabellen in einer Datei zusammenfassen. Beachten Sie jedoch, dass die Datei nicht größer ist als die verfügbare Speicherkapazität Ihrer Festplatte oder Diskette.

Mit dem Schalter **Save variable name** geben Sie an, ob Sie den Variablennamen als Spaltenkopf mit abspeichern wollen. Mit den Schalter **Comma as decimal delimiter** kennzeichnen Sie, ob Dezimalstellen mit einem Komma oder einem Punkt vom ganzzahligen Teil abgetrennt werden.

8.2 Tabellen löschen

Nicht mehr benötigte Tabellen können Sie mit dem Befehl **Delete** im **Chart**-Menü löschen. Es empfiehlt sich, diesen Befehl von Zeit zu Zeit durchzuführen, damit Ihr Speicher nicht unnötig belastet wird (außerdem wird dadurch auch die Geschwindigkeit des Systems beeinträchtigt). Nach Erteilung des **Delete**-Befehls erscheint das **Delete Charts**-Fenster. Um mehrere aufeinanderfolgende Ergebnisprotokolltabellen (z.B. die nicht mehr benötigten ersten 50 Tabellen) zu selektieren, müssen Sie nicht jede einzelne Tabelle mit der Maustaste markieren. Sie können bei gedrückter linker Maustaste die Maus nach unten ziehen. Mehrere nicht unmittelbar aufeinanderfolgende Tabelle markieren Sie, indem Sie gleichzeitig die *Strg*-Taste gedrückt halten. Nach-

dem alle zu löschenden Tabellen markiert sind, bestätigen Sie mit **Ok**. Sollen alle Tabellen im Arbeitsspeicher gelöscht werden, so betätigen Sie den Schalter **All**.

8.3 Tabellen- Info

Wenn Sie einen kurzen Überblick über die Werte einer Tabelle erhalten wollen, wählen Sie den Befehl **Info** im **Charts**-Menü. Daraufhin öffnet sich das in Abbildung 7 links gezeigt **Chart Info**-Fenster. Nachdem Sie eine Variable markiert und den Schalter **Info** betätigt haben, werden Sie über die Anzahl der Werte der Tabelle, Minimal- und Maximalwert, und ob und wohin die Tabelle gespeichert wurde, informiert. Im unteren Teil des **Info**-Dialogs können Sie ein Ausschnittsfenster auf die Tabelle legen. Im Feld **Bottom Index** geben Sie den unteren Wert, den Sie inspizieren wollen, an. Mit **Nr of Values** geben Sie die Größe des Ausschnitts an, die maximal 2000 beträgt. Anschließend betätigen Sie den Schalter **List**. Es erscheint eine Listbox mit sämtlichen protokollierten Variablenwerten des Simulationslaufes, durch die Sie sich mit den Pfeiltasten bewegen können. Zur Beendigung der Tabellen-Information betätigen Sie den Schalter **End**.

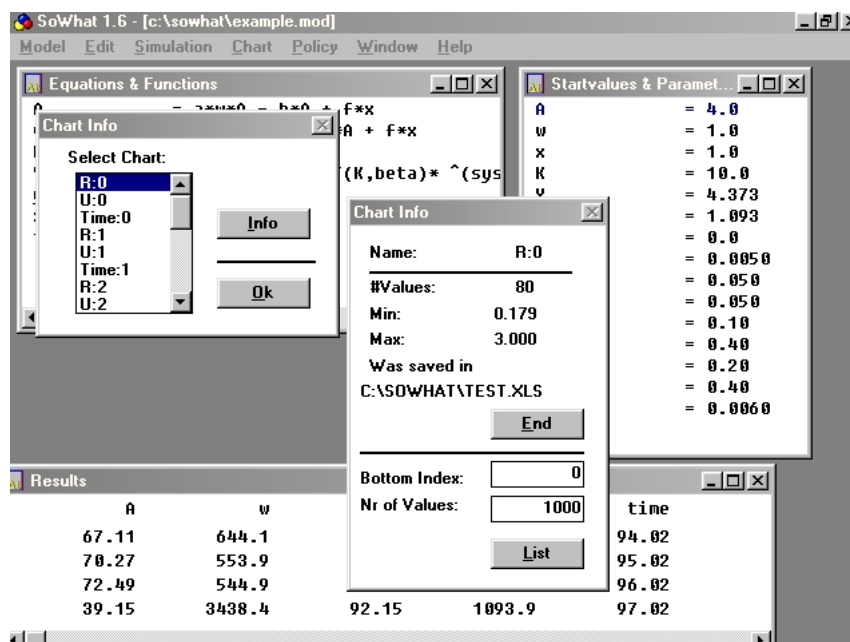


Abb. 7: Das Tabellen Info-Fenster

Chapter 9

Das Politik-Menü

Eine wesentliche Erweiterung von SOWHAT 1.6 ist die Politik-Sektion. Wenn Sie über Erfahrungen mit einfachen Programmstrukturen und Algorithmen verfügen, wird Ihnen die Anwendung der Politik keine Schwierigkeiten bereiten. Die Grundbegriffe **System Condition**, **Value Change** und **Equation Change** werden Sie intuitiv begreifen, die syntaktischen Regeln werden Sie den Beispielmustern entnehmen können. Dennoch sind einige Besonderheiten bei der Definition der Politik zu beachten, die Sie in diesem Kapitel finden werden. Haben Sie wenig oder keine Kenntnisse der Programmierung in einer Computer-Sprache, sollten Sie dieses Kapitel aufmerksam studieren.

9.1 Allgemeine Erläuterungen zur Politik-Sektion

Durch die **Policy**-Sektion haben Sie die Möglichkeit Eingriffe in laufende Simulationen zu antizipieren. Hierdurch haben Sie ein Werkzeug in der Hand, das deutliche Ähnlichkeiten zur Programmierung in einer Hochsprache wie beispielsweise Pascal, C oder FORTRAN aufweist. In der Politik-Sektion von SOWHAT ist die IF-THEN-Struktur der oben angesprochenen Hochsprachen implementiert. Zu dem Konstrukt IF-THEN gehören unmittelbar eine **Bedingung** und ein **Statement**. Wird die Bedingung WAHR, wird das Statement ausgeführt.

Beispiel 1.

IF $a > b$ THEN $a = a - 5$

In Worten ausgedrückt: falls a größer als b ist, wird von a der Wert 5 subtrahiert, falls a nicht größer als b ist, passiert nichts. In diesem Beispiel ist $a > b$ die Bedingung (Condition) und $a = a - 5$ das Statement.

Das Statement ist in der Programmierung immer eine Wertzuweisung. Zusätzlich können Sie in SOWHAT ganze Gleichungen ersetzen. Anders formuliert, Sie können grundlegende Modellspezifikationen wie Funktionen, Differentialgleichungen etc. aufgrund eines Zustands Ihres Systems zur Laufzeit durch andere Spezifikationen ersetzen. Zusätzlich können Sie eine Reihe von **Spezialstatements** verwenden, die den Lauf der Simulation steuern, falls eine Bedingung eintritt.

Oftmals werden mehrere Statements aufgrund einer Bedingung ausgeführt. Wir können z.B. das obige Beispiel erweitern durch eine Klammerung zweier Statements durch BEGIN und END.

Beispiel 2.

```
IF  $a > b$  THEN
BEGIN
 $a = a - 5$ 
 $b = c + 2$ 
END
```

Die Komplexität der mathematischen Ausdrücke in der Bedingung und in den Statements ist beliebig, solange in der Bedingung ein Vergleich und in den Statements eine Zuweisung stattfindet.

Die Bedingungen bestehen in vielen Fällen nicht nur aus einem Vergleich, sondern aus der logischen Verknüpfung mehrerer Einzelvergleiche.

Beispiel 3.

```
 $(a > b)$  AND  $(b < (c * d))$  THEN
```

Nur wenn beide Einzelbedingungen WAHR werden, ist die Gesamtbedingung WAHR, das Wort AND bedeutet das UND der boolschen Algebra.

Beispiel 4.

IF $(a > b)$ OR $(b > c)$ THEN

In diesem Fall, werden die beiden Einzelbedingungen durch das logische ODER verknüpft. Diese Bedingung wird WAHR, wenn entweder die erste oder die zweite Bedingung den Wert WAHR hat.

Weitere Beispiele möglicher Verknüpfungen

IF $(a > b)$ OR $(b < c)$ OR $(c < d)$ THEN

IF $((a < b)$ OR $(b < c))$ AND $(c < d)$ THEN

IF $((a < b)$ AND $(b < c))$ OR $((c < d)$ AND $(d < e))$ THEN

Die Umsetzung dieser Strategien erfolgt in SOWHAT ohne dass Sie programmieren müssen. Die Definition der Bedingungen und Statements erfolgt dialog-gesteuert mit größtmöglicher Unterstützung durch die Software. Die inhaltliche Formulierung Ihrer Politik-Sektion obliegt dennoch Ihnen. Sie sollten die Politik sehr gewissenhaft einsetzen, da Sie enorme Freiheiten und Möglichkeiten besitzen, sich hierdurch jedoch leicht Fehler einschleichen können.

Die Politik-Definition ist Teil des Modells und wird zusammen mit dem Modell gespeichert und geladen. Die Verwendung der Politik-Sektion ist optional. Die mit Versionen von SOWHAT < 1.5 erstellten Modelle haben z.B keine Politik-Definition aber sind direkt in höheren Versionen ladbar.

Kern der Politik sind Systemzustände. Sie sind die Bedingungen aus dem vorherigen Abschnitt. Sie entstehen durch Vergleiche von Ausdrücken, in denen Variable und Parameter des aktuellen Modells verwendet werden. Systemzustände sind die Basis, auf der Wert- und Gleichungsänderungen aufbauen. Es ergibt sich somit eine Referenzhierarchie

1. Systemzustände stützen sich auf Modellvariable und -parameter
2. Wert/Gleichungsänderungen stützen sich auf Systemzustände und Modellvariable und -parameter

Wird diese Hierarchie unterbrochen, beispielsweise durch das Entfernen eines referenzierten Systemzustandes, wird das Modell *inkonsistent*. Für inkonsistente Modell ergeben sich naturgemäß Einschränkungen, so ist z.B. das Simulieren dieser Modelle solange nicht möglich, bis durch geeignete Maßnahmen die Referenzierungskette wiederhergestellt ist.

Wichtige Begriffe, die in den folgenden Kapiteln häufig verwendet werden sind **Basisgleichung** und **Alternativgleichung**. Eine Basisgleichung ist eine Funktion oder Gleichung, die im Modell enthalten ist. Wird diese durch eine Gleichungsänderung der Politik-Sektion ersetzt, so nennt man die ersetzende Gleichung Alternativgleichung. Es sind mehrere Alternativgleichungen für eine Basisgleichung möglich.

Der Wahrheitswert der Systemzustände wird vor jeder Iteration ermittelt. Die Wert- und Gleichungsänderungen werden dementsprechend überprüft und ggf. durchgeführt. Zeitlich ergibt sich während der Simulation folgende Struktur

```
WHILE active DO
BEGIN
Überprüfe Systemzustände
Überprüfe und führe Wert- und Gleichungsänderungen
durch
Berechne die Werte der nächsten Iteration
END
```

9.2 Systemzustände

Systemzustände sind boolesche Operanden, d.h sie können zwei Werte annehmen, WAHR und FALSCH. Ein Systemzustand hat einen Namen, unter dem er referenziert wird, sowie eine Bedingung, die seinen Wahrheitswert bestimmt.

Die Definition der Systemzustände erfolgt in dem in Abbildung 8 dargestellten **Conditions**-Fenster, das Sie durch Anwählen von **Define Condition** im **Policy**-Menü erreichen. Hier werden Sie nach einem Namen gefragt, der maximal 12 Zeichen lang sein darf und der Namenskonvention von SOWHAT gehorchen muss. Im zweiten Eingabefeld geben Sie die Bedingung des Systemzustandes ein. Die Bedingung besteht aus zwei mathematischen Ausdrücken und einem

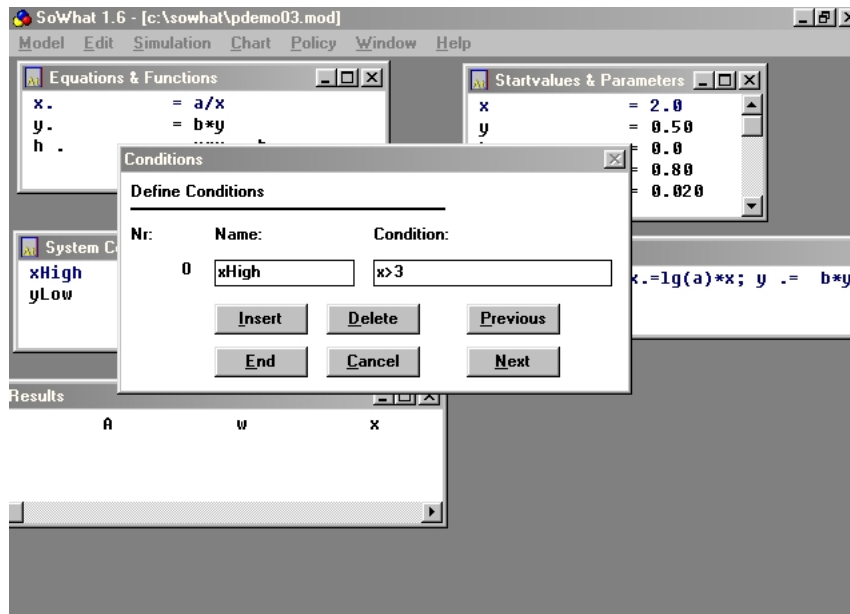


Abb. 8: Das Systemzustand - Fenster

Vergleichsoperator.

Beispiel.

Name: xHigh

Condition: $x > 3$

Die Komplexität der zu vergleichenden Terme ist beliebig, Sie können auch die bereitgestellten mathematischen Funktionen wie \sin , abs , trnc , etc. verwenden. Eine logische Verknüpfung innerhalb der Bedingung ist unzulässig, die Klammerung der Gesamtbedingung, wie etwa $(x < y)$, führt zu einem *Lexical-Scan*-Fehler.

Vor jeder Iteration werden die beiden Vergleichsterme berechnet und anschließend verglichen. Liefert dieser Vergleich den Wahrheitswert WAHR, erhält der Systemzustand den booleschen Wert WAHR (entsprechendes gilt für den Fall, dass die Auswertung FALSCH ergibt).

Sie dürfen maximal 32 Systemzustände in Ihr Modell integrieren. Die Länge der Bedingung darf 255 Zeichen nicht überschreiten. Alle Systemzustände Ihres Modells können im Arbeitsfenster **System Conditions** eingesehen werden. Ein Doppelklick mit der linken Maustaste auf einen Systemzustand in diesem Fenster führt Sie in den Dialog **Define Condition**.

9.3 Wertänderungen

Wertänderungen bewirken die Änderung einer Variablen oder eines Parameters während einer Simulation. Die auszuführende Änderung wird vor dem Starten der Simulation in dem in Abbildung 9 dargestellten **Value Changes**-Fenster definiert, das Sie über das **Policy**-Menü erreichen.

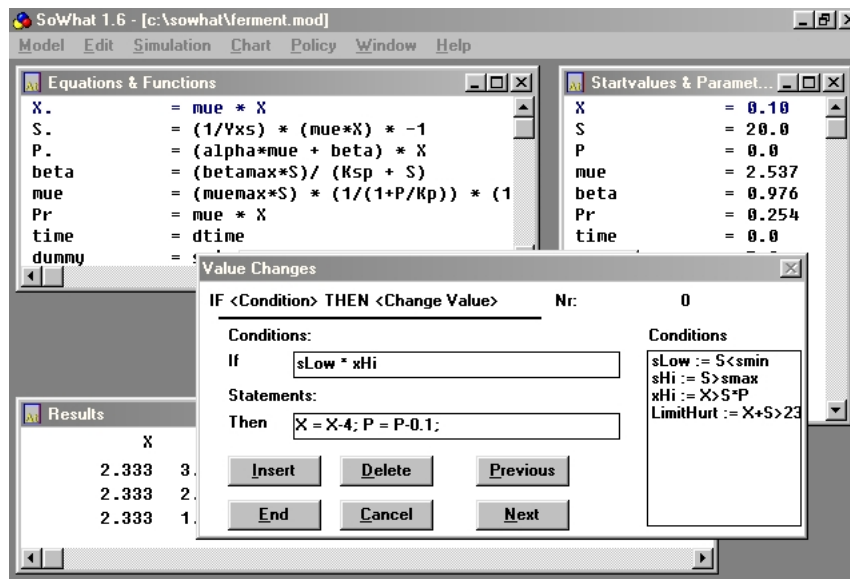


Abb. 9: Das Wertänderung - Fenster

Wertänderungen sind Bestandteile eines Modells und werden mit diesen gespeichert und geladen. Im **Value Changes**-Dialog werden Sie nach einer Zustandsverknüpfung und Statements gefragt. Im Feld **Conditions** geben Sie eine logische Verknüpfung von bereits definierten Systemzuständen ein. Im Feld **Statements** spezifizieren Sie die durchzuführenden Wertänderungen.

Bei der Definition der Zustandsverknüpfung können Sie die Systemzustände verwenden, die im rechts stehenden Listenfenster aufgeführt sind.

Beispiel: Seien xLow, xHigh, sLowerX und xDoIt Systemzustände, so könnten einige mögliche Verknüpfungen folgendermaßen aussehen:

$xLow + xHigh$
 $xLow * xDoIt$
 $(xLow * sLowerX) + xDoIt$

Das Plus-Zeichen (+) symbolisiert das logische ODER, das Multiplikations-Zeichen (*) steht für das boolsche UND. Eine Klammerung ist möglich, der Wahrheitswert wird durch die bekannten Regeln ermittelt.

Im Feld **Statements** können Sie eine oder mehrere Wertzuweisungen angeben. Eine Zuweisung besteht aus dem Namen einer im Modell enthaltenen Variable, dem Gleichheitszeichen und einem Ausdruck.

Beispiele für Statements:

$x = xMax/2$
 $rValue = a * \lg(b); rCount = rCount + 1$

Im zweiten Beispiel werden zwei Statements aufgrund einer Zustandsverknüpfung durchgeführt. Das Semikolon (;) trennt die einzelnen Statements voneinander. Es sind beliebig viele Zuweisungen möglich, allerdings darf die Länge aller Statements im Eingabefeld 255 Zeichen nicht überschreiten. Innerhalb der Zuweisungen können Sie beliebige Ausdrücke und Funktionen verwenden, wie bspw. im zweiten Beispiel die Funktion Decadischer Logarithmus (\lg).

Betrachten wir ein vollständiges Beispiel (als PDEMO01.MOD im Lieferumfang enthalten):
 Ein Modell besteht aus den Gleichungen

1) $r. = a/r$
 2) $aShow = a$

Mit den Startwerten

1) $r = 1.0$
 2) $aShow = 0.1$ (Endogen)
 3) $a = 0.1$

Wir definieren zwei Systemzustände

1) $aMax = r > 3$
 2) $aLow = r < 1$

Sowie die Wertänderungen

$$1) aMax : a = -a$$

$$2) aLow : a = -a$$

Da zu Simulationsbeginn keiner der Systemzustände WAHR ist, beginnt das System mit der Berechnung der Differentialgleichung mit den Werten $r = 1$, $a = 0.1$. Offensichtlich wächst r mit abnehmender Rate und wird den Wert 3 erreichen. Wenn dies geschieht, wird $aMax$ WAHR, die Wertänderung 1) wird durchgeführt, und r wird sinken, da $a = -0.1$ ist. Wird im Laufe der weiteren Simulation $r < 1$, wird $aLow$ WAHR und 2) wird ausgeführt und somit $a = 0.1$, woraus folgt, dass r wieder wächst. Die Variable r wird sich also immer zwischen 1 und 3 bewegen. Die Funktion $aShow = a$ wurde eingefügt, um den momentanen Wert der Variable a während der Simulation beobachten zu können.

In diesem Beispiel könnten wir denselben Effekt auch mit nur einer Wertänderung erzielen. Diese wäre dann folgendermaßen zu formulieren:

$$1) aMax + aLow : a = -a$$

Wenn $aMax$ ODER $aLow$ WAHR wird, wechselt der Parameter a das Vorzeichen. Beachten Sie, dass $-$ bedingt durch die zeitliche Reihenfolge der Politik-Überprüfungen $-$ eine Wertänderung *vor* der ersten Iteration durchgeführt werden würde, falls die Zustandsverknüpfung WAHR ist. Für unser Beispiel bedeutet dies, dass, wenn Sie den Startwert von $r = 0.5$ setzen, die Wertänderung 2) sofort zu Beginn der Simulation durchgeführt wird. Die Startwerte sind davon nicht betroffen.

Wichtig: Wertänderungen werden *nur bei der Änderung* des Wahrheitswertes der Zustandsverknüpfung ausgeführt. Ein simples Beispiel soll dies verdeutlichen (PDEMO02.MOD): Gegeben ist die Differentialgleichung

$$e. = a * e$$

Mit den Startwerten

$$1) e = 1$$

$$2) a = 0.2$$

Sowie der Systemzustand

MyCondition= $e > 5$

Und der Wertänderung

MyCondition: $a = 0.05; e = 6$

Nachdem die Variable e den Wert 5 erreicht hat, verringert sich die Steigung, da $a = 0.05$ gesetzt wird, außerdem wird $e = 6$. Der Systemzustand MyCondition bleibt im weiteren Verlauf WAHR, die Wertzuweisung wird jedoch nur bei der Änderung des Systemzustands ausgeführt, da andernfalls permanent die Variable e auf den Wert 6 gesetzt werden würde.

9.4 Gleichungsänderungen

Gleichungsänderungen werden in dem in Abbildung 10 dargestellten Fenster **Equation Changes** eingegeben, der durch Anwahl von **Equation Change** im **Policy**-Menü zu erreichen ist. Gleichungsänderungen sind Bestandteile der Modelle und werden, wie auch Systemzustände und Wertänderungen mit diesen gespeichert und geladen. Eine Gleichungsänderung bedeutet das Austauschen einer im Modell enthaltenen Gleichung (Basisgleichung) durch eine andere (Alternativgleichung) aufgrund einer Systemzustandsverknüpfung. Die aktive Gleichung wird auch als Bestimmungsgleichung bezeichnet.

Im Gleichungsänderungs-Dialog, werden Sie im ersten Eingabefeld nach der **Condition** und im zweiten Feld nach **Alternative Equation** gefragt. Die Zustandsverknüpfung ist ein logischer Ausdruck, dessen Operanden Systemzustände sind. Wenn in Ihrem Modell die Systemzustände xLowerZero, yLowerZero und xHigherLimit existieren, dann könnten Zustandsverknüpfungen folgendermaßen aussehen:

1) xLowerZero

2) xLowerZero + xHigherLimit

3 (xLowerZero + xHigherLimit) * yLowerZero

In 1) besteht die Verknüpfung nur aus einem Systemzustand, in 2) und 3) werden mehrere Zustände durch das logische ODER (das Plus-Zeichen +) und das logische UND (das Multip-

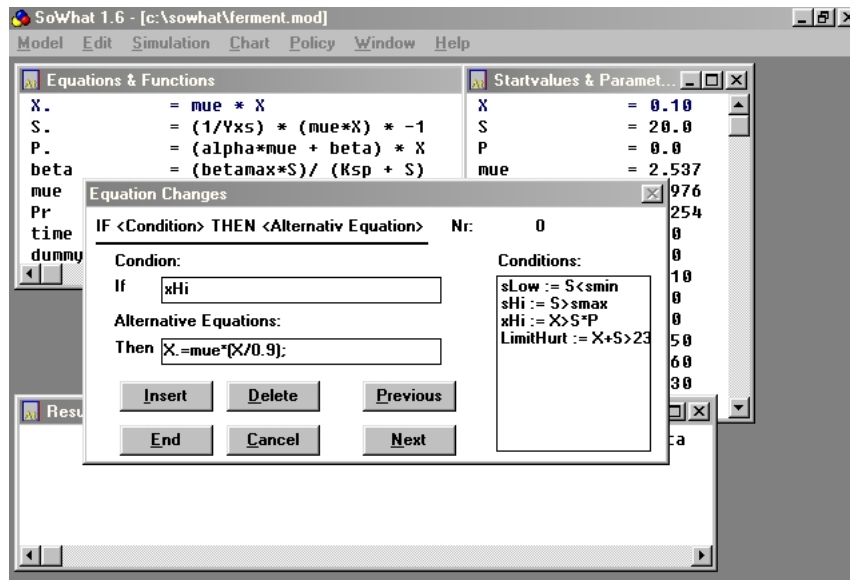


Abb. 10: Das Gleichungsänderung - Fenster

likationszeichen *) verbunden. Die Verknüpfung 1) wird WAHR, falls xLowerZero WAHR wird, 2) wird WAHR, wenn xLowerZero oder xHigherLimit WAHR werden. Die Verknüpfung 3) wird WAHR, wenn yLowerZero UND mindestens einer der Zustände xLowerZero bzw. hHigherLimit WAHR wird.

Im Feld **Alternative Equation** geben Sie Gleichungen und Funktionen ein, die die entsprechende Basisgleichung ersetzen sollen. Der Gleichungstyp der Alternativgleichung muss natürlich gleich dem der Basisgleichung sein, eine Differentialgleichung kann nur durch eine Differentialgleichung, ein Substitut nur durch ein Substitut ersetzt werden, etc. Wird die **Condition** WAHR, so ersetzt die Alternativgleichung *dauerhaft* die Basisgleichung des Modells. Oftmals wird gefordert, die Basisgleichung wieder zur Bestimmungsgleichung werden zu lassen, falls die **Condition** FALSCH ist. In Pseudocode formuliert ergäbe sich folgendes Konstrukt:

IF <Bedingung> THEN <Alternativgleichung> ELSE < Basisgleichung>

Um dieses herzustellen, wird der Alternativgleichung {} angehängt. Die geschweiften Klammern symbolisieren die Basisgleichung.

Beispiel.

Value := Value*a{};

Das abschließende Semikolon ist optional und wird von SOWHAT automatisch angefügt. Zusätzlich kann innerhalb der Klammern eine Gleichung angegeben werden, die im ELSE-Fall zur Bestimmungsgleichung wird.

Beispiel.

$$Sub := dtime * 0.1 \{ Sub := (dtime + 4) * 0.2 \}$$

Bedenken Sie aber, dass die Basisgleichung des Modells *nie* Bestimmungsgleichung während der Simulation wird, sie wird einzig für die Berechnung endogener Startwerte verwendet.

Angenommen, es sind folgende Basisgleichungen in Ihrem Modell enthalten:

- 1) $r. = a * r$
- 2) $y. = x * (b - z) - y$

So wären folgende Alternativgleichungen denkbar:

- 1) $r. = a * r / 2 \{ \}$
- 2) $y. = y - x / a$

Im Beispiel 1) wird die Basisgleichung für den Fall, dass die Verknüpfung FALSCH ist, als Bestimmungsgleichung von r eingesetzt. Im zweiten Beispiel bleibt die Alternativgleichung aktiv, sofern sie einmal Bestimmungsgleichung geworden ist. Die referenzierten Variablen (r , a , y , x) müssen im Modell enthalten sein. Sie können mehrere Alternativgleichungen aufgrund einer Zustandsverknüpfung definieren. Beispielsweise existieren die Systemzustände $xHigh$ und $yLow$ sowie die Basisgleichungen

- 1) $x. = a / x$
- 2) $y. = b * y$

Die Deklaration einer Gleichungsänderung könnte folgende Gestalt haben:

Verknüpfung: $xHigh + yLow$
 Alternativgleichungen: $x. = \lg(a) * x; y. = b * y - y$

Falls $xHigh$ oder $yLow$ WAHR wird, werden die Basisgleichungen durch die zugehörigen Alternativgleichungen (dauerhaft) ersetzt.

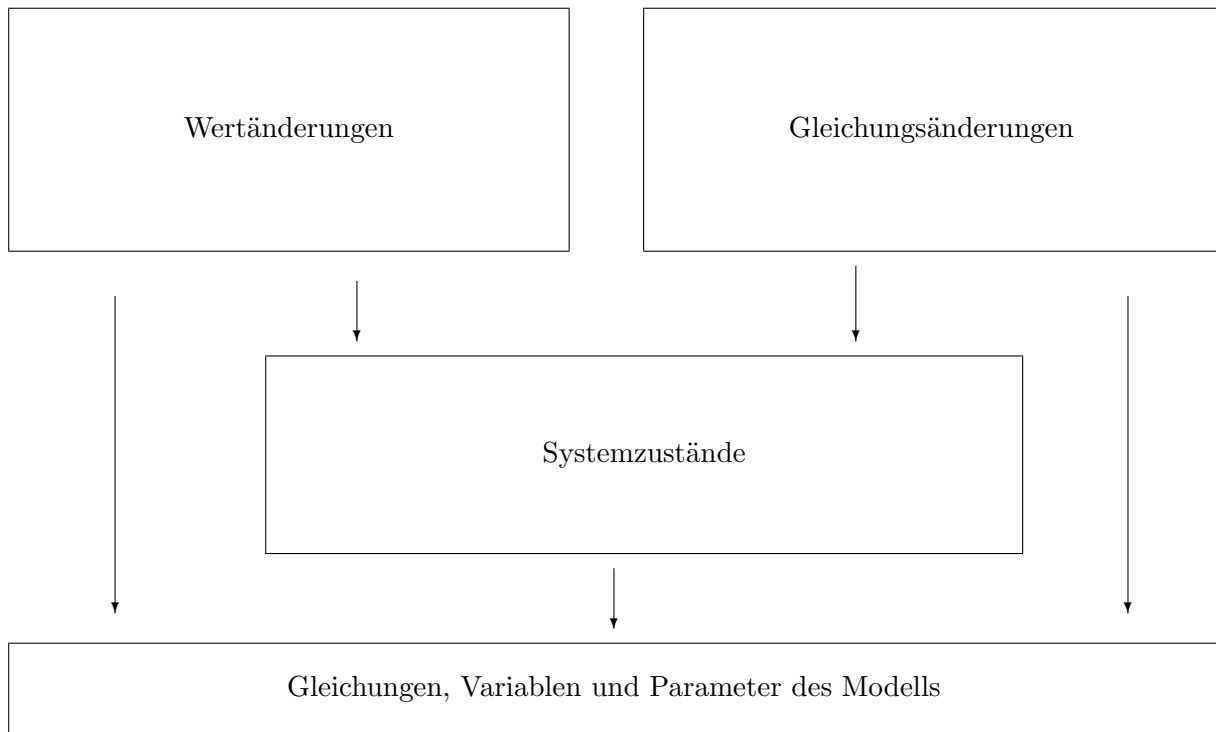


Fig. 11 Referenzhierarchie

Sie können mehrere Alternativgleichungen für eine Basisgleichung definieren, die Überprüfung der Zustände, deren Verknüpfungen und den davon abhängigen Gleichungsänderungen, finden chronologisch entsprechend ihrer Reihenfolge in den jeweiligen Arbeitsbereichen statt.

9.5 Referenzhierarchie und Konsistenz

Durch den Aufbau der Politik-Sektion entsteht eine Beziehung zwischen den einzelnen Modellbestandteilen, die als Referenzhierarchie bezeichnet wird. Grundlage sind die Variablen des Modells. Diese werden von den Systemzuständen, Wert- und Gleichungsänderungen angesprochen. Die Systemzustände wiederum werden von Wert- und Gleichungsänderungen referenziert, so dass sich insgesamt die in Abb. 11 dargestellte Referenzhierarchie ergibt:

Ein Modell, bei dem diese Hierarchie nicht durchbrochen ist, wird als *konsistent* bezeichnet. Nehmen wir den Fall, dass eine Basisgleichung entfernt wird, zu der eine Alternativgleichung in der Politik-Sektion existiert. SOWHAT meldet Ihnen, dass das Modell nun inkonsistent ist, da die Ergebnisvariable der Alternativgleichung nicht gefunden wurde. Bis auf weiteres sind Simulationen nicht möglich. Das Wiederherstellen der Konsistenz erfolgt in diesem Fall durch das Einfügen einer entsprechenden Basisgleichung oder durch das Löschen der Gleichungsänderung, die die entfernte Basisgleichung referenziert.

Inkonsistenz kann folgende Ursachen haben:

- Ein **Systemzustand** referenziert eine Variable, die aus dem Modell entfernt wurde.
- Eine **Wertänderung** referenziert eine Variable, die aus dem Modell entfernt wurde.
- Eine **Gleichungsänderung** referenziert eine Variable, die aus dem Modell entfernt wurde.
- Ein **Wertänderung** referenziert einen Systemzustand, der entfernt wurde.
- Eine **Gleichungsänderung** referenziert einen Systemzustand, der entfernt wurde.
- Eine **Gleichungsänderung** referenziert eine Basisgleichung, die entfernt wurde.

Nachdem Sie die Konsistenz des Modells durch geeignete Maßnahmen wiederhergestellt haben, wählen Sie **Consistency** im **Policy**-Menü, erst danach sind Simulationen wieder zugelassen.

Wenn Sie ein inkonsistentes Modell speichern, werden Sie gefragt, ob Sie möchten, dass die Politik-Definition des Modells ebenfalls gespeichert wird. Geben Sie an dieser Stelle *Yes* an, wird das Modell in diesem Zustand gespeichert, geben Sie *No* an, wird die Politik-Definition des Modells verworfen. Das Laden inkonsistenter Modelle bedingt die Meldung, dass das Modell inkonsistent ist und Simulationen nicht möglich sind, solange die Konsistenz nicht wiederhergestellt ist.

9.6 Strategien der Politik-Definition

In diesem Kapitel werden nützliche Hinweise zur Erstellung einer Politik-Definition gegeben. Um weitere Einblicke in die Verwendung der Politik zu erlangen, sollten Sie die mitgelieferten

Beispielmodelle laden.

A) Definition von Variablen für die Politik-Sektion.

Variablen werden von SOWHAT automatisch nach der Modell-Definition in die interne Variablenliste eingetragen. Eine Variable existiert demnach nur dann im Modell, wenn sie Bestandteil einer Gleichung oder Funktion ist. Oftmals wünscht man sich jedoch bei der Entwicklung einer Politik-Definition zusätzliche Variablen, wie bspw. Ober- oder Untergrenzen, die in der Regel nicht Teil des Modells sind. Ebensovienig sind Zählvariablen (die weiter unten beschrieben werden) in den Gleichungen enthalten. Um dieser Beschränkung zu entgehen, können Sie eine Variablengleichung erstellen. Beispielsweise wird die Gleichung

$$var = SMax + SMin + SCount + SAdd$$

in das Modell eingefügt. Die Variablen *SMax*, *SMin*, *SCount* und *SAdd* sind jetzt im Modell enthalten und können von Systemzuständen, Wertänderungen und Gleichungsänderungen referenziert werden. Die Werte der Variablen, die bei *SMax* und *SMin* vermutlich konstant sind, geben Sie im Dialog **Startvalues & Parameters** ein.

B) IF-THEN.

Ein häufig benutztes Konstrukt in der Politik-Sektion ist die Wertänderung aufgrund einer Bedingung.

Beispiel.

Sei folgende Gleichung im Modell enthalten

$$S = 1/a * j * X$$

wobei stets $a > 0$, $X > 0$ und $j < 0$ erfüllt ist. Offensichtlich nimmt S stetig ab. Angenommen, Sie wünschen, dass, wenn S unter einen Minimalwert sinkt, eine Aktion durchgeführt wird. Beispielsweise soll zu S ein bestimmter Wert addiert werden. In Pseudocode formuliert: IF $S < SMin$ THEN $S = S + SAdd$.

Die Realisierung in SOWHAT erfolgt durch einen Systemzustand **Condition**

Name: *Slow*

Bedingung: $S < SMin$

und eine Wertänderung **Value Change**

Verknüpfung: *SLow*

Statement: $S = S + SAdd$

Sie können mehrere Wertänderungen aufgrund eines Systemzustands definieren. Das Semikolon (;) trennt die einzelnen Wertzuweisungen.

C) Zähler einführen und verwenden.

Die im obigen IF-THEN-Beispiel durchgeführte Aktion soll nur 5-fach ausgeführt werden. Zu diesem Zweck erstellen wir eine Zählvariable, die in einer Wertänderung inkrementiert wird. Zunächst fügen wir der Variablengleichung $+SN$ hinzu und setzen die Werte $SCount = 0$ und $SN = 4$ im Dialog **Startvalues & Parameters**. Wir definieren einen Systemzustand

Name: *DoIt*

Bedingung: $SCount < SN$

Der Systemzustand *SDoIt* bleibt WAHR solange $SCount < SN$ ist. Jetzt verändern wir die Wertänderung aus dem obigen Unterkapitel

Verknüpfung: $SLow * SDoIt$

Statement: $S = SMax; SCount = SCount + 1; 1$

Nach der fünften Durchführung dieser Wertänderung, wird *SDoIt* FALSCH. Da die logische Bedingung für die Wertänderung die Verknüpfung *SLow* UND *SDoIt* ist, wird diese nicht mehr durchgeführt.

D) Mehrere Alternativgleichungen für eine Basisgleichung.

Die Abarbeitung der Politik-Definition während der Simulation erfolgt in der Reihenfolge, in der sie im Arbeitsbereich (Arbeitsfenster) aufgelistet ist. Ziel ist folgendes Konstrukt in Pseudocode:

IF <Bedingung 1> THEN <Statement 1>

ELSE IF <Bedingung 2> THEN <Statement 2>

Die Realisierung in SOWHAT erfolgt folgendermaßen.

Basisgleichung $X. = a * X$

Systemzustände:

Name: $XHigh$

Bedingung: $X > 30$

Name: $XHigh2$

Bedingung: $X > 50$

Gleichungsänderungen

Nummer: 0

Verknüpfung: $XHigh$

Alternativgleichung: $X. = b * X$

Nummer: 1

Verknüpfung: $XHigh2$

Alternativgleichung: $X. = c * X$

Setzen wir als Startwert $X = 1$ voraus und $a > 0$ und $b > 0$. Die Alternativgleichung 0. wird aktiv, wenn $X > 30$ wird. Wird $X > 50$, dann ersetzt die 1. die 0. Alternativgleichung.

E) Wie können Sie erkennen, ob ein Systemzustand WAHR oder FALSCH ist?

Sie haben beispielsweise den Systemzustand *MyCond*, dessen Wahrheitwert Sie während der Simulation gerne beobachten würden. Sie definieren eine Basisgleichung im Modell

$$cActiv = 0;$$

Verleihen Sie *cActiv* das Attribut **Recorded Variable**. Erstellen Sie eine Gleichungsänderung

Verknüpfung: *MyCond*

Alternativgleichung: $cActiv = 1\{\}$;

Im Resultatfenster erkennen Sie nun, ob *MyCond* WAHR ($cActiv = 1$) oder *MyCond* FALSCH ($cActiv = 0$). Wichtig ist hier die Definition der ELSE-Klammern {}, da andernfalls *cActiv* nie wieder = 0 werden würde.

9.7 Spezialstatements

Spezialstatements werden in Wertänderungen und Gleichungsänderungen der Politik benutzt. Sie befehlen dem Programm während einer laufenden Simulation, aufgrund eines Systemzustandes, etwas “Spezielles” durchzuführen. Spezialstatements beginnen mit dem Dollar-Zeichen (\$).

Liste der Spezialstatements:

\$Stop	Beenden der Simulation
\$Break	Unterbrechen der Simulation mit Wert-Ändern-Dialog
\$Bell	Auslösen der Glocke
\$RepeatNoChart	Beenden der Simulation, Verwerfen der Tabellen der Simulation, Neustart der Simulation
\$RepeatChart	Beenden der Simulation, Tabellen bleiben erhalten, Neustart der Simulation
\$SkipNoChart	Beenden der Simulation, Verwerfen der Tabellen der Simulation, Mit nächstem Loop-Wert aufsetzen
\$SkipChart	Beenden der Simulation, Tabellen bleiben erhalten, Mit nächstem Loop-Wert aufsetzen

Wenn Sie beispielsweise hören möchten, ob eine Variable eine Grenze überschritten hat, müßten Sie folgende Schritte durchführen:

- Definieren eines Systemzustandes $xHigh$, der WAHR wird, falls $x > Limit$
- Definieren einer Wertänderung, die durchgeführt wird, wenn $xHigh$ WAHR wird, mit dem Statement \$Bell statt oder zusätzlich zu einer Wertsetzung. Wird die Bedingung WAHR, ertönt ein Signalton.

Wenn eine Zufallsvariable in Ihrem System enthalten ist, die Ihr System u.U. in Zustände führt, die für Sie uninteressant sind, können Sie eine Simulation durch das Spezialstatement \$Stop beenden. Wenn Sie eine so beendete Simulation wiederholen wollen, verwenden Sie die Statements \$RepeatNoChart oder \$RepeatChart. Die Simulation wird - im Falle, dass die Bedingung eintritt - wiederholt.

Wenn eine Parameter-Loop in Ihrem Modell enthalten ist, können Sie mit den `$Skip`-Statements Simulationsläufe stoppen und mit dem nächsten Loop-Wert wiederaufsetzen.

Beispiel.

Über den Parameter P wurde die Loop `[1.5&8.5]0.5` gelegt. Für den Fall, dass eine Variable X einen Grenzwert überschreitet, definieren Sie eine Wertänderung, die im Statementfeld den Befehl `$SkipNoChart` enthält. Alle Läufe mit Werten von P , die zu einem solchen Überlauf führen, werden abgebrochen, die Tabellen werden verworfen und die Simulation startet mit dem nächsten P -Wert.

Die `$Repeat`-Statements sind nur sinnvoll, wenn Zufallsvariable im Modell enthalten sind, die `$Skip`-Statements nur dann, wenn eine Parameter-Loop existiert.

Chapter 10

Das SOWHAT Hilfe - System

Im **Help**-Menü von SOWHAT sind das **Info**-Fenster und die SOWHAT- Hilfe, die Sie über den Befehl **Index** erreichen, enthalten. Die Hilfe ist gemäß des Windows-Standards programmiert und enthält das englischsprachige Kurzhandbuch. Nach dem Aufruf von **Index** öffnet sich über das Windows-Help-System das Inhaltsverzeichnis der SOWHAT Hilfe. Im folgenden wird davon ausgegangen, dass sie mit einer deutschsprachigen Windows-Version arbeiten. Sie können im Hilfesystem blättern oder über die unterstrichenen und grün dargestellten Indexwörter gezielt nach einer Hilfestellung für Ihr Problem suchen. Diese Wörter enthalten Querverweise zu anderen Hilfetexten. Mit dem **Inhalt**-Befehl springen Sie zum Inhaltsverzeichnis der Hilfe zurück.

Mit dem Befehl **Suchen** erhalten Sie eine Liste von Schlüsselwörtern, über die Sie die für Sie relevante Hilfestellung direkt erreichen können. Mit **Zurück** gelangen Sie auf die von ihnen zuvor aufgeschlagene Seite des Hilfetextes. **Bisher** listet alle von ihnen bisher angewählten Hilfestellungen auf, in die Sie durch Markieren des entsprechenden Themas zurückspringen können.

Im Menü **Datei** können Sie über den Befehl **Thema drucken** den aktuellen Hilfetext auf Ihrem Drucker ausgeben, Ihren Drucker einrichten, mit **Öffnen** in Hilfesysteme anderer Programme springen und mit **Beenden** die Hilfe verlassen und zu Ihrer Simulation zurückkehren. Im Menü **Bearbeiten** können Sie über **Kopieren** den SOWHAT-Text in die Windows-Zwischenablage kopieren und mit **Anmerken** die SOWHAT-Hilfe mit eigenem Text erweitern. Auf diese Anmerkungen können Sie bei späteren Programmaufrufen zurückgreifen. Über die Anwahl des Fragezeichens erhalten Sie durch Windows umfangreiche Hilfestellungen zur Benutzung der Hilfe.

Chapter 11

Das Verfahren zur Lösung der dynamischen Systeme

Das eingesetzte Verfahren kann nur im Rahmen der Rechnergenauigkeit exakte Lösungen liefern. Im Falle der Differentialgleichungen sind die (approximierten) Lösungen vor dem Hintergrund des Runge-Kutta-Verfahrens und seinen Einschränkungen zu sehen. Und nicht zuletzt gilt auch hier das GIGO-Prinzip (Garbage In - Garbage Out).

Übersicht.

Die Systeme sind von Ihnen definiert durch folgende Bestandteile

Gleichungen

- Differentialgleichungen
- Differenzgleichungen
- Substitute
- Funktionen

Simulationsoptionen

- Simulationszeitraum
- Schrittweitensteuerungsparameter

Je nach Gleichungstyp werden die Ergebnisvariablen mittels unterschiedlicher Verfahren berechnet. Von großer Bedeutung ist die Reihenfolge, in der die Werte ermittelt werden, da Interdependenzen möglich, in den meisten Fällen sogar wahrscheinlich sind. Sie können durch entsprechende Formulierung Ihrer Modelle Einfluss auf die Aufrufreihenfolge nehmen. Dauer und Genauigkeit des Verfahrens hängen stark von den Simulationsoptionen ab. Wenn Differentialgleichungen in Ihren Modellen vorkommen, erhalten die Schrittweitenparameter große Bedeutung. Durch diese Parameter wird der aufwendigere Teil der Simulation, nämlich das Entwickeln der Differentialgleichungen über den gewünschten Zeitraum gesteuert. Für die Berechnung von Systemen, in denen Differenzgleichungen verwendet werden, sind bestimmte Optionsvorgaben zu empfehlen.

Zunächst werden die grundlegenden Ideen der einzelnen Verfahren sowie die Implementation in SOWHAT aufgeführt. Anschließend wird die Verbindung der jeweiligen Verfahren zueinander dargestellt. Den Abschluss des Kapitels bilden einige konkrete Beispiele.

Differentialgleichungen

Da Differentialgleichungssysteme i.d.R. nicht analytisch lösbar sind, werden numerische Verfahren benötigt, die hinreichend genaue Näherungen der gesuchten Lösungsfunktion liefern. SOWHAT verwendet zur Lösung der Systeme ein vierstufiges Runge-Kutta-Verfahren mit Schrittweitensteuerung.

Für einen numerischen Lösungsansatz wird das Problem diskretisiert. Hierdurch entsteht eine Abweichung der genäherten von der exakten Lösung, die als **lokaler Diskretisationsfehler** bezeichnet wird (auch Abschneidefehler genannt). Geht man nun in kleinen Schritten vorwärts, wird der lokale Diskretisationsfehler klein, wird die Schrittweite groß gewählt, wird der Fehler groß. Allerdings muss bedacht werden, dass eine Computer-Implementation gefordert ist, somit steht nur begrenzt Rechenzeit zur Verfügung. Weiterhin sind Operationen in digitalen Rechenmaschinen fehlerbehaftet. Wünschenswert wäre eine Schrittweite, die möglichst groß ist, aber einen möglichst kleinen Fehler bedingt. Eine gute Möglichkeit zur Bestimmung einer geeigneten Schrittweite liefert das Prinzip der **automatischen Schrittweitensteuerung**, welches im folgenden beschrieben wird.

Es werden zwei Verfahren unterschiedlicher Fehlerordnung eingesetzt. Zur Ermittlung der Näherungslösung wird ein 4-stufiges Runge-Kutta-Verfahren verwendet, dessen lokaler Fehler durch ein 6-stufiges Verfahren bestimmt wird. Ist der Fehler groß, wird die Schrittweite verringert, ist der Fehler klein, wird die Schrittweite erhöht.

Um das Verfahren zu steuern, müssen demnach zwei Parameter bestimmt werden, der **maximale Fehler**, der nicht überschritten werden darf, sowie die anfängliche Schrittweite **Start**. Zusätzlich werden zwei Schrittweitenwerte **Minimal** und **Maximal** definiert, die eine Eingrenzung der Schrittweite ermöglichen (stark nichtlineare Systeme bewirken andernfalls evtl. das ‐Aufhängen‐ des Rechners durch ständige Verkleinerung der Schrittweite). Zusätzlich kann angegeben werden, ob der Fehler **absolut** oder **relativ** verarbeitet werden soll.

Da die ermittelten Werte in Tabellen gespeichert werden sollen, wird eine weitere Schrittweitendefinition **Record** benötigt, die angibt, wann ein Variablenwert in seiner Tabelle gespeichert werden soll. Durch **Record** wird ein äquidistantes Raster über die Zeit gelegt, an dessen Gitterlinien die Werte in die Zeitreihen eingetragen werden. Außerdem müssen natürlich **Startzeit** und **Endzeit** der Simulation festgelegt werden (Somit ergibt sich die Simulationsdauer unmittelbar als Differenz der beiden Parameter).

In jeder Iteration werden gemäß des Runge-Kutta-Verfahrens die Näherungslösungen der Differentialgleichungen für den nächsten ‐Schritt‐ berechnet. Mit dem Verfahren höherer Fehlerordnung, wird der geschätzte Fehler ermittelt. Ist der Fehler größer als der zugelassene maximale Fehler (**max. Error**), wird die interne Schrittweite verringert und das Verfahren beginnt erneut an derselben Stelle, ist der Fehler geringer als der maximale Fehler, wird die Lösung übernommen und, falls der Fehler sehr klein ist, die Schrittweite für die nächste Iteration vergrößert.

Die Fehlerabschätzung des Differentialgleichungssystems in einem Schritt wird ermittelt aus der euklidischen Vektornorm der Einzelfehler. Wird die Option **Error relative** gewählt, wird jeder Einzelfehler zunächst durch den Wert der Variablen geteilt und danach die Vektornorm gebildet. Vorteil dieser Option ist das Beschleunigen des Verfahrens, wenn Variablen in hohe Zahlenbereiche vordringen und demgemäß der absolute Fehler zwar sehr groß, gemessen an der Größenordnung jedoch zu vernachlässigen ist.

Würde der nächste Schritt über die durch **Record** definierte Grenze reichen, forciert das

Verfahren die Schrittweite dergestalt, dass das Raster getroffen wird. Die interne Schrittweite wird danach auf den vorherigen Wert gesetzt und das Verfahren schreitet fort.

Differenzgleichungen

Differenzgleichungen werden immer ausgeführt bzw. berechnet, wenn die interne Schrittweite das Protokollraster trifft. Wenn Sie Differenzgleichungen berechnen, sollten Sie also i.d.R die Schrittweite **Record** auf den Wert 1 setzen.

Substitute

Substitute sind Funktionen, die sich dadurch auszeichnen, dass sie (zeitlich) als erstes in jedem Integrationsschritt berechnet werden. Sie ermöglichen somit das Substituieren zusammenhängender mathematischer Ausdrücke in Differentialgleichungen, Differenzgleichungen und Funktionen.

Funktionen

Funktionen werden (zeitlich) als letztes in einem Integrationsschritt berechnet. Funktionsergebnisse dienen in der Regel als Ausgabe- oder Vergleichsvariable.

Reihenfolge der Berechnungen

Für den Fall, dass Sie ein Modell definiert haben, das alle 4 Gleichungstypen verwendet, ergibt sich folgende Vorgehensweise:

Als erstes werden die **Substitute** in Abhängigkeit aller Werte der Vorperiode und der aktuellen Zeit berechnet. Hängt ein Substitut von einem anderen ab, wird dieses vorher berechnet, hängt dieses Substitut wiederum von einem anderen Substitut ab, wird zuerst jenes berechnet. Tritt hierbei eine Rekursion auf, d.h. die Abhängigkeitskette führt zum ersten Substitut zurück, wird Ihnen dieses als Fehler gemeldet und Sie müssen das Modell entsprechend ändern.

Danach werden die **Differentialgleichungen** mit dem Runge-Kutta-Verfahren berechnet, und zwar in Abhängigkeit der aktuellen Zeit und der aktuellen Substitute sowie folgenden Werten der Vorperiode: den Differentialgleichungsergebnissen, den Differenzgleichungsergebnissen sowie der Funktionen.

Die Differenzgleichungen werden jetzt, falls das Protokollraaster getroffen wird, berechnet in Abhängigkeit der aktuellen Zeit, aller aktuellen Substitute, sowie den anderen Variablen der Vorperiode.

Schließlich werden die Funktionen auf Basis aller aktuellen Werte berechnet. Hängt eine Funktion von einer anderen ab, wird diese zuerst berechnet. Wie bei den Substituten sind direkte und indirekte Rekursionen unzulässig.

Beispiele

Beispiel 1.

Gegeben sei folgendes - willkürlich definiertes - Modell

$f = d - a$	Funktion
$x := dtime$	Substitut, das die abgelaufenen Zeit in x zur Verfügung stellt
$S := \gamma(sysE, x)$	Zweites Substitut, das x referenziert
$d. = d/(S + a)$	Differentialgleichung, die S referenziert
$a. = d - 1/(a * a)$	a und d referenzieren sich gegenseitig

Es ergibt sich folgende Reihenfolge der Berechnungen:

- 1) x in Abhängigkeit von der Systemvariable $dtime$
- 2) S in Abhängigkeit von $x(= dtime)$ und der Systemvariablen $sysE$.
- 3) d und a in Abhängigkeit von $S(= \gamma(sysE, dtime))$ und von $dt - 1, at - 1$
- 4) f in Abhängigkeit der aktuellen Werte von d und a .

Beispiel 2.

Das Modell

$$eDGL. = eDGL$$

$$eDIFF\& = sysE * eDIFF$$

$$eFUNC = \gamma(sysE, dtime)$$

ergibt die Aufrufreihenfolge $eDGL$, dann $eDIFF$ und als letztes $eFUNC$. Alle drei Gleichungen

führen zum gleichen Ergebnis, der e-Funktion.

Chapter 12

Meldungen und Fehlermeldungen

Die Meldungen des Systems sind nach Ursachen bzw. Arbeitsbereichen getrennt. Vor der eigentlichen Meldung erscheint der Name der Fehlerklasse. Wird eine Gleichung oder Funktion syntaktisch falsch definiert, erscheint bspw. die Meldung

Lexical Scan Fehler 45 : Unzulässige Syntax an Position 10 nach dem Gleichheitszeichen

Durch die bekannten Windows-Symbole, wie das Stop-Schild, das Fragezeichen oder das Info-Icon, wird Ihnen die Schwere des Fehlers bzw. Art der Beantwortung mitgeteilt. Wenn das Stop-Schild in der Mitteilung erscheint, ist ein Fehler aufgetreten, der die gewünschte Aktion, Bedienung oder Eingabe von Ihnen zurückweist. Das Info-Symbol wird in Mitteilungen eingeblendet, die reine Benachrichtigungen (z.B. Ende der Simulation) sind, oder die eine gewünschte Bedienung kommentieren. Mitteilungen, die das Fragezeichen enthalten, stellen Ihnen Fragen, die mit Ja oder Nein beantwortet werden müssen.

Einige Fehlermeldung sind mit einem Präfix in eckigen Klammern versehen. Zum einen deutet dies auf Fehlerursachen hin, die außerhalb von SOWHAT liegen – z.B. bedeutet der Vorsatz [Windows], dass ein Fehler in oder durch Windows auftritt. Zum anderen werden einige Meldungen als [internal] gekennzeichnet. Diese Fehler treten in der Regel nicht auf. Sie kennzeichnen unzulässige Systemzustände, die SOWHAT registriert und meldet. In den meisten Fällen empfiehlt es sich, SOWHAT (und ggf. auch Windows) zu beenden und neu zu starten.

Folgende Meldungs- bzw Fehlerklassen existieren:

- Meldungen teilen Ihnen Informationen mit
- Lexical Scan Fehler, treten bei der Definition der Modelle auf
- Mathematische Fehler, treten bei Berechnungen (insbesondere bei Simulationen) auf
- Tabellen Fehler, treten bei Tabellen-bezogenen Operationen auf
- Command-Stack Fehler, melden modell-interne Inkonsistenzen
- Generelle Fehler, sind Fehler der unterschiedlichsten Art
- Simulationsoptionen-Fehler, treten bei der Definition der Simulationsoptionen auf
- Politik-Fehler und -Meldungen

Meldungen.

- 21 **Ende der Simulation** Die Simulation endete. Wird kein Grund aufgeführt, ist des Ende des Simulationszeitraumes erreicht. Andernfalls wird die Ursache der Beendigung mitgeteilt.
- 22 **Mitteilung** Verschiedene Meldungen des Systems.
- 23 **Nicht mehr als 2000 Werte darstellbar** Im Info-Fenster können maximal 2000 Werte einer Tabelle dargestellt werden. Justieren Sie das Ausschnittsfenster auf den gewünschten Bereich durch die Parameter *Bottom Index* und *it Nr of Values*.

Lexical Scan -Fehler.

Die Lexical Scan-Fehler können aufgrund der rekursiven Abarbeitung der mathematischen Ausdrücke leicht mißverständlich wirken, da syntaktische Fehler erst bei der Sichtung des nächsten Terms entdeckt werden. Oftmals muss der Grund eines Syntax-Fehlers *vor* der gemeldete Position gesucht werden.

- 41 (**erwartet**. SOWHAT erwartet innerhalb eines mathematischen Ausdrucks eine öffnende Klammer. Die Gleichung, in der der Fehler auftritt, wird nach Bestätigung der Fehlermeldung im Feld *Function* des **Define Model** Fensters angezeigt. In der Fehlermeldung erfahren Sie, an welcher Stelle nach dem Gleichheitszeichen der betreffenden Gleichung SOWHAT die öffnende Klammer vermisst.

- 42) **erwartet**. SOWHAT erwartete innerhalb eines mathematischen Ausdrucks eine schließende Klammer. Die Gleichung, in der der Fehler auftritt, wird nach Bestätigung der Fehlermeldung im Feld *Function* des **Define Model** Fensters angezeigt. In der Fehlermeldung erfahren Sie, an welcher Stelle nach dem Gleichheitszeichen der betreffenden Gleichung SOWHAT die schließende Klammer vermisst.
- 43 **Exp ungültig**. Unzulässige Syntax bei der Eingabe einer Exponentiation. Vgl. Kapitel 5.2.
- 44 **Unzulässiges Minuszeichen**. Bei der syntaktischen Überprüfung eines mathematischen Ausdrucks wurde ein unzulässiger Minusoperator gefunden.
- 45 **Unzulässige Syntax**. Diese Fehlermeldung kann vielfältige Ursachen haben. Der Lexical Scan bemüht sich, den exakten Grund eines Syntax-Fehlers zu bestimmen, gelingt ihm dies nicht, wird diese Meldung generiert.
- 46 **Komma erwartet**. Innerhalb eines Funktionsausdruckes wurde ein Komma vermisst, bspw. führt der Ausdruck $\max(a)$ zu dieser Meldung.
- 47 **Unzulässiges Formelende**. Ein mathematischer Ausdruck wurde syntaktisch nicht korrekt beendet.
- 48 **Name der Variablen oder Funktion zu lang**. Innerhalb eines mathematischen Ausdrucks wurde ein Bezeichner gefunden, der länger als die Maximallänge von 12 Zeichen ist.
- 49 **Keine Gleichung oder Funktion**. Der Ausdruck konnte nicht als Gleichung oder Funktion interpretiert werden. Diese Meldung wird z.B generiert, wenn kein Gleichheitszeichen in der Formel enthalten ist.
- 50 **Ergebnisname zu lang**. Die Länge eines Ergebnisnamens überschreitet die Maximallänge von 12 Zeichen.
- 51 **Unzulässiger Ergebnisname**. Der Ergebnisname einer Funktion oder Gleichung enthält unzulässige Zeichen. Hierzu gehören z.B. alle Symbole, die eine definierte mathematische Bedeutung haben.

Mathematische Fehler.

Mathematische Fehler treten bei der Berechnung der Startwerte und während der Simulationen auf. Sie führen zum Abbruch der Simulation. Die auslösende Gleichung wird in der Meldung aufgeführt. Die Exponentiation $\sim(a, b)$ stellt eine Besonderheit dar. Sie wird durchgeführt mittels einer Logarithmus-Operation und der Exponentiation mit der Eulerschen Zahl e als Basis. Da

Logarithmen negativer Zahlern nicht zulässig sind, muss die Basis einer $\hat{\ }^$ Funktion stets positiv sein (Siehe Kap. 4.2 *exp*).

60 **Division by Zero.** Bei einer Division wurde der Nenner Null.

61 **Negative Sqrt -Operation.** Die Diskriminante wurde negativ.

62 **Negative Log - Operation.** Der Numerus innerhalb einer Logarithmus-Operation wurde negativ.

63 **Ungültige Exp - Operation.** Die Basis einer Exponentiation wurde negativ.

64 **Numerischer Überlauf.** Eine mathematische Operation würde zu einem numerischen Überlauf führen und wird daher nicht ausgeführt.

65 **Additions-Überlauf.** Eine Addition würde zu einem numerischen Überlauf führen und wird daher nicht ausgeführt.

66 **Log Überlauf in Exp - Operation.** Während der Berechnung von $\gamma(a, b)$ trat ein numerischer Überlauf innerhalb der Logarithmus-Operation auf.

67 **Divisions-Überlauf.** Während einer Division trat ein numerischer Überlauf auf.

68 **Simulation beendet wegen Step-Aufhänger.** Während einer Simulation wurde die interne Schrittweite gleich Null. Die Simulation wird beendet, kontrollieren Sie die Simulationsoptionen!

Tabellen Fehler.

Diese Fehler treten beim Speichern und Löschen der Tabellen auf.

80 **Zu viele Tabellen, bitte speichern und löschen.** Die Maximalanzahl intern gehaltener Tabellen (400) wurde erreicht.

81 **Keine Tabellen ausgewählt.** Beim Speichern oder Löschen von Tabellen wurde keine Tabelle selektiert.

82 **Plattenkapazität überschritten.** Das Speichern von Tabellen schlug fehl, da nicht genügend Speicherplatz auf der Diskette oder Festplatte vorhanden ist.

83 **Fehler beim Schließen der Tabellendatei.** Eine Tabellendatei konnte nach dem Speichern nicht geschlossen werden. Versuchen Sie, die Datei auf einem anderen Medium oder unter anderem Namen zu speichern.

84 **Fehler beim Öffnen der Tabellendatei.** Das Öffnen einer Datei zum Speichern der Tabelle schlug fehl. Versuchen Sie, die Tabellendatei auf einem anderen Medium oder unter anderem Name zu speichern.

85 **Fehler beim Öffnen der TSP- Header - Datei.** Das Öffnen der TSP Header-Datei zum Speichern der Tabelle schlug fehl. Versuchen Sie, die Tabellendatei auf einem anderen Medium oder unter anderem Namen zu speichern.

Command-Stack Fehler.

Diese Fehlermeldungen sollten im normalen Betrieb nicht auftreten. Aufgrund der großen Variabilität bei der Definition der Modelle sind jedoch sporadisch auftretende Inkonsistenzen innerhalb der Modelle möglich. Beenden Sie SOWHAT (und ggf. auch Windows) und starten Sie das Programm neu.

90 [**internal**] **Command Stack Overflow.** Die Maximalgröße eines Command-Stacks wurde überschritten.

91 [**internal**] **Invalid Command Stack Constant Value.** Eine Konstante auf dem Command-Stack kann nicht interpretiert werden.

92 [**internal**] **Command Stack Pointer Nil.** Die Referenz auf einen Command-Stack konnte nicht gefunden werden.

Generelle Fehler

101 **Datei nicht gefunden.** Der Name einer zu ladenden Datei wurde falsch spezifiziert.

102 [**internal**] **Invalid RPN- Construct.** Das Resultat des Lexical-Scan kann intern nicht interpretiert werden.

103 **Unzulässiges Zahlenformat.** Die Eingabe einer Zahl enthält unzulässige Zeichen.

104 **Kein Speicherplatz mehr vorhanden.** Es kann kein weiterer Speicher belegt werden. Schließen Sie aktive Windows-Applikationen oder löschen Sie Tabellen.

105 **Unzulässiges Zeichenkettenformat.** Das Format oder die Länge einer Zeichenkette ist unzulässig.

106 **Unzulässige Rekursion.** Indirekte Rekursionen bei der Definition von Funktionen und Substituten sind nicht zulässig. Überprüfen Sie den Aufbau Ihres Modells.

- 107 **Nur eine Parameter - Loop erlaubt.** Es wurde versucht mehr als eine Parameter-Loop zu definieren.
- 108 **Loop - Start > Loop - Ende.** Der Startwert einer Parameter-Loop wurde größer als der Endwert definiert.
- 109 **Unzulässige Parameter - Loop.** Die Werte der Parameter-Loop sind nicht zulässig, bitte überprüfen Sie die Definition der Parameter-Loop.
- 110 **Endogene Variable sind nicht änderbar.** Sie haben versucht eine endogene Variable zu ändern.
- 111 **Parameter Loop Variable sind nicht änderbar.** Während der Simulation können Parameter-Loop Werte nicht geändert werden.
- 112 [**internal**] **Seek Error, please save, quit and restart.** Eine Tabelle konnte intern nicht lokalisiert werden. Dieser Fehler deutet auf Probleme bei der Speicherverwaltung hin. Sie sollten SOWHAT beenden und neu starten.
- 113 [**internal**] **Heap Error, please save, quit and restart.** Dieser Fehler deutet auf Probleme bei der dynamischen Speicherverwaltung hin. Sie sollten SOWHAT beenden und neustarten.
- 114 [**internal**] **Index Error, please save, quit and restart.** Das Eintragen der Werte in die Tabellen schlug fehl. Sie sollten SOWHAT beenden und neu starten.
- 115 [**Windows**] **Release Device Context Failure.** Das Freigeben eines Geräte-Contexts der Windows-Geräteschnittstelle schlug fehl. Da Windows nur 5 Device Contexts bereitstellt, sollten Sie Ihr System herunterfahren und neu starten, andernfalls kann es zu System-Aufhängern kommen.
- 116 [**internal**] **There are no Charts.** Die interne Übergabe von selektierten Tabellen schlug fehl. Beenden Sie SOWHAT und starten es erneut.
- 117 **Fehler beim Laden des Modells.** Ein Modell konnte nicht geladen werden. Eventuell wurde die Datei außerhalb von SOWHAT verändert.
- 118 **Fehler beim Öffnen der Datei.** Eine Datei konnte nicht geöffnet werden.
- 119 **Plattenkapazität überschritten.** Ein Modell konnte nicht gespeichert werden, da nicht genügend Speicherplatz auf dem angewählten Medium vorhanden ist.
- 120 **Demo - Version - Operation ist nicht erlaubt.** Die gewünschte Operation ist nicht erlaubt, da Sie eine Demo-Version von SOWHAT benutzen.
- 121 **SoWhat wurde nicht komplett installiert.** Das Programm kann nicht gestartet werden, da Teile des Software-Pakets nicht ordnungsgemäß installiert wurden.

- 122 **Kein SoWhat Modell.** Die geladene Datei konnte nicht als Modell interpretiert werden.
- 123 **Unzulässiger Zufallsbereich.** Überprüfen Sie die Parameter der Zufallsdefinition.
- 124 **Ergebnisvariable können nicht zufällig sein.** Der Startwert einer Resultatvariable muss explizit spezifiziert werden.
- 125 **Negative Varianz.** Der zweite Parameter der normalverteilten Zufallsdefiniton muss positiv sein.
- 126 **Zufallsvariable sind nicht änderbar.** Während laufender Simuationen können zufällige Parameter nicht geändert werden.
- 127 **Zu viele Werte für die Zwischenablage.** Die Windows-Zwischenablage kann die selektierten Tabellen nicht aufnehmen.
- 128 **Das Modell enthält zu viele Variablen.** Die maximale Anzahl der Variablen wurde erreicht.

Simulationsoptionen-Fehler.

Die zur Steuerung der Simulation erforderlichen Parameter müssen aufeinander abgestimmt sein. Verstöße gegen diese Regeln werden mit folgenden Meldungen beantwortet.

- 130 **Startzeit > Endzeit.** Die Startzeit der Simulation muss kleiner als die Endzeit gewählt werden.
- 131 **Schrittweite: Minimum > Maximum.** Die Minimale Schrittweite muss kleiner als die Maximale Schrittweite gewählt werden.
- 132 **Schrittweite: Start > Maximum.** Die anfängliche Schrittweite muss kleiner als die Maximale Schrittweite gewählt werden.
- 133 **Schrittweite: Start < Minimum.** Die anfängliche Schrittweite muss größer als die Minimale Schrittweite gewählt werden.
- 134 **Schrittweite: Protokoll < Maximum.** Die Protokoll-Schrittweite muss größer als die Maximal-Schrittweite gewählt werden.
- 135 **Negative Schrittweite.** Die Schrittweiten der Simulationsoptionenen müssen positiv definiert werden.
- 136 **Max. Error <= 0.** Der Maximale Fehler muss positiv gewählt werden.

Meldungen der Politik-Sektion.

Die Meldungen der Politik.Sektion beziehen sich auf die Bereiche Systemzustände, Wert- und Gleichungsänderungen sowie Konsistenz bzw. Inkonsistenz.

- 150 **Kein Systemzustandsname.** In einem Eingabefeld wurde der Name eines Systemzustandes falsch eingegeben.
- 151 [**internal**] **Condition Pointer NIL.** Der Zeiger auf einen Systemzustand konnte nicht interpretiert werden.
- 152 [**internal**] **vState Pointer NIL.** Der Zeiger auf eine Wertänderung konnte nicht interpretiert werden.
- 153 **Variable nicht gefunden.** Innerhalb einer Wert- oder Gleichungsänderung konnte eine eingegebene Variable nicht gefunden werden. Überprüfen Sie die Namen der Variablen im Statement-Feld. Möchten Sie neue Variable für die Politik- Sektion bereitstellen, lesen Sie das Kapitel 8.6 Politik-Strategien.
- 154 **Ungültiges Spezialstatement.** Ein Spezialstatement (Dollar-Statement) wurde nicht korrekt eingegeben.
- 155 **Der Ausdruck kann nicht als Statement identifiziert werden.** Überprüfen Sie die Syntax des Statement.
- 156 **Erster Vergleichsausdruck fehlt.** Im Bedingungsfeld einer Systemzustandsdefinition wurde der erste Vergleichsoperand nicht spezifiziert.
- 157 **Zweiter Vergleichsausdruck fehlt.** Im Bedingungsfeld einer Systemzustandsdefinition wurde der zweite Vergleichsoperand nicht spezifiziert.
- 158 **Vergleichsoperator fehlt.** Der Vergleichsoperator im Bedingungsfeld einer Systemzustandsdefinition wurde nicht eingegeben.
- 159 **Endogenen Variablen kann kein Wert zugewiesen werden.** In einer Wertänderung dürfen endogene Variable nicht verwendet werden.
- 160 **Unzulässiger Zustandsname.** Der Name eines Systemzustandes ist unzulässig.
- 161 **Kein gültiges Alternativ-Statement.** Die Definition der Alternativgleichung im Dialog Gleichungsänderung ist nicht korrekt.
- 162 **Keine Basisgleichung dieses Namens im Modell enthalten.** Der Name der Ergebnisvariablen einer Gleichungsänderung wurde im Modell nicht gefunden. Überprüfen Sie die Namen der Basis- und der Alternativgleichungen.

- 163 **Inkonsistenter Modellzustand.** Das Modell ist inkonsistent.
- 164 **Alternativgleichung entspricht nicht der Basisgleichung.** Der Gleichungstyp (Differentialgleichung, Funktion) einer Gleichungsänderung entspricht nicht dem Typ der Gleichung im Modell.
- 165 **Inkonsistentes Modell geladen.** Sie haben ein Modell geladen, welches im inkonsistenten Zustand gespeichert wurde. Simulationen können zur Zeit nicht durchgeführt werden. Um die Konsistenz herzustellen, müssen Sie die Politik-Sektion überprüfen und die Referenzhierarchie wiederherstellen.
- 166 **Leere Zeile nicht erlaubt.** In einer Wert- oder Gleichungsänderung ist eine Eingabezeile leer. Entfernen Sie diese Zeile.
- 167 **Politik ist konsistent.** Die Überprüfung der Referenzhierarchie ergab, dass die Politik konsistent ist. Simulationen sind möglich.
- 168 **Systemzustandsname bereits vergeben.** Der Name eines Systemzustandes darf nur einfach vergeben werden.

Chapter 13

Erläuterungen zur Modellbibliothek

Modellname	Charakteristische Eigenschaften	Eigen-	Literaturempfehlung
BESSEL.MOD	Bessel Differentialgleichung (2.Ordnung)		Walter W., Gewöhnliche Differentialgleichungen, Springer-Verlag, S. 166ff.
EXAMPLE.MOD	Willkürliches Beispielmodell		SoWhat Quick Start
FERMENT.MOD	Simulation mikrobiologischer Wachstumsmodelle in der Biotechnologie		Wucherer H., Heiler H., BioEngineering Nr.2 1992.
GUREL.MOD	Sechsdimensionales Differential- gleichungssystem liefert für die spezifizierten Parameter einen stabilen Grenzzyklus		Gurel, O. (1976): Partial Peeling: in: Cesari, L., Hale, J.K., LaSalle, J.P.: Dynamical Sys- tems, Vol. 2, New York, u.a.; S.255ff.
LOOP.MOD	Räuber-Beute-Modell mit Para- meter-Loop		
LORENZ.MOD	Dreidimensionales Differential- gleichungssystem. Liefert den bekannten Lorenz-Attraktor		Guckenheimer, J., Holmes, P. (1983): Non- linear Oscillations, Dynamical Systems, and Bifurcation Theory.
LOTTKA.MOD	Zweidimensionales Differential- gleichungssystem (Räuber-Beute Modell)		Lotka, A.Y. (1925): Elements of Physical Bi- ology, Baltimore.
NEUBERT.MOD	Zweidimensionales Differen- zgleichungssystem mit chaotischem Bereich		Neubert, M.G., Kot, M. (1992): The Subcrit- ical Collapse of Predator Populations in Dis- crete -Time Predator-Prey Modells, Mathe- matical Biosciences, 110, S. 45ff
PDEMO0x.MOD	Beispielmodelle für die Politik- Sektion		SoWhat Handbuch und Help- System

ROESSLER.MOD	Vierdimensionales Differentialgleichungssystem mit chaotischen Attraktoren (Hyperchaos)	Rössler, O.E. (1980): Chaos, in: Güttinger/Eikemeyer: Structural Stability in Physics, Berlin, u.a., S.290ff
SIMPLE.MOD	Beispielmodell für alle Gleichungsklassen	SoWhat Handbuch und Help- System
SOLOW.MOD	Grundmodell der neoklassischen Wachstumstheorie	Solow, R.M. (1956), A Contribution to the Theory of Economic Growth, Quarterly Journal of Economics, Vol. 70, S. 56-87.
STEEB.MOD	Differentialgleichung zweiter Ordnung (van der Pool), dargestellt als zweidimensionales Differentialsystem mit chaotischem Bereich	Steeb/Huang/Gou (1989): A Comment on the Chaotic Behaviour of van der Pol Equations with an External Periodic Excitation, Zeitschrift für Naturforschung, Vol. 44a, S. 160ff.
SWING.MOD	Beispielmodell für einfache Wertänderungen	SoWhat Handbuch und Help-System
YORKE.MOD	Logistische Differenzgleichung erster Ordnung mit chaotischen Eigenschaften für bestimmte Parameterwerte	Li, T.Y., Yorke, J. (1975): Period three implies chaos, American Mathematical Monthly, 82, S. 985-992.