

Zusammenfassung: Die Untersuchung komplexer Systeme zeigt, daß in ihnen Rückkopplungen, Verzögerungen und Verstärkungen, also Regelungsvorgänge zur Wirkung kommen. Solche kybernetischen Prozesse können durch die Simulationsmethode SYSTEM DYNAMICS beschrieben werden. SYSTEM DYNAMICS-Modelle sind Simulationsmodelle aus nichtlinearen gekoppelten Differentialgleichungen, die auf einem Digitalrechner als gekoppelte Differenzgleichungen gelöst werden. In dem folgenden Beitrag wird die Methode vorgestellt, die Arbeitstechnik erörtert und die Anwendbarkeit diskutiert.

Summary: The research of complex systems proves the effect of feedbacks, delays and reinforcements within them. A description of these cybernetic processes is made through application of the simulation method SYSTEM DYNAMICS. SYSTEM DYNAMICS models are simulation models consisting of non-linear coupled differential equations, which can be solved as difference equations by use of a digital computer. In the following contribution an introduction to the method, an explanation of the working technique and a discussion of the application is given.

1. Anwendung von Simulationsverfahren

Simulationsverfahren werden zur Entscheidungsfindung und Erarbeitung von Lösungsstrategien (policies) auf komplexe Problembereiche angewendet, bei denen intuitive und analytische Methoden nicht zum Ziele führen. Ziel ist dabei nicht unbedingt die Suche nach einem Optimum, sondern in vielen Fällen die Erweiterung der Kenntnisse über die Struktur und das charakteristische Verhalten des simulierten realen Systems zur Ableitung zieladäquater Handlungsalternativen. Die Abbildung der Realität geschieht anhand eines Modells, das einen Ausschnitt der Wirklichkeit wiedergibt. Eine Interpretation der strukturellen Eigenschaften von Simulationsmodellen ist bereits in Heft 3/72 in einem Aufsatz von J. NIEDEREICHHOLZ/ I. BEY erfolgt. Dort ist in einer Kurzbeschreibung die Simulationsmethode System Dynamics¹⁾ vorgestellt und ein Elementarmodell für die Abbildung des Investitionsprozesses von Unternehmen skizziert worden. An dieser Stelle wird versucht, eine weiterführende Darstellung der Methode zu geben, die Arbeitstechnik zu verdeutlichen und die Anwendbarkeit zu diskutieren.

¹⁾ früher Industrial Dynamics

2. Die Philosophie von System Dynamics

2.1. Feedback-Loops

Entwickelt wurde System Dynamics (im folgenden mit S.D. abgekürzt) von JAY W. FORRESTER [2, 3] als Anwendung regelungs- und entscheidungstheoretischer Erkenntnisse zur Untersuchung komplexer dynamischer Systeme. Im Sinne dieser Definition werden unter komplexen Systemen nicht überschaubare rückgekoppelte Systeme verstanden, die z.B. aus ökonomischen, technischen, ökologischen oder sozialen Elementen aufgebaut sein können. Die Systemelemente stehen untereinander in einer Ursache-Wirkungsbeziehung. Bilden diese kausalen Verkettungen der Systemelemente einen geschlossenen Kreis, so liegt ein Feedback-Loop vor. Das bewirkt, daß eine Änderung irgendeiner Einflußgröße innerhalb des Loops eine Reihe von Ereignissen nach sich zieht, die eventuell zu einem späteren Zeitpunkt Rückwirkungen auf die Variable haben, von der die Änderung ursprünglich ausging. Feedback-Systeme werden also von ihrer eigenen Vergangenheit beeinflusst. Fast alle rationalen Vorgänge oder Informationsmechanismen laufen in geschlossenen Feedback-Loops ab. Der Entscheidungsprozeß ist ein typisches Beispiel für einen derartigen Rückkopplungskreis (Bild 1).

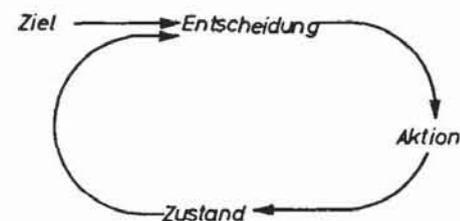


Bild 1 Entscheidungsprozeß

Die verfügbare Information über den Zustand einer Systemgröße wird mit dem gewünschten Zustand (Ziel) verglichen. Der Vergleich führt zu Entscheidungen, die ergebniskorrigierende Aktionen einleiten, wodurch wieder der Zustand der Regelgröße verändert wird. Dieser ideale kontinuierliche Transformationsprozeß von Informationen in Aktionen wird gewöhnlich von Störungen und Verzögerungen beeinflusst, so daß die rückgekoppelte Information nicht den aktuellen Zustand der Regelgröße wiedergibt. Den oben beschriebenen Entscheidungs-Loop bezeichnet man als negativ, weil die Rückwirkung auf die die Änderung verursachende Loop-Variable der Änderung entgegengerichtet ist. Negative Loops besitzen eine Regelungsfunktion und sind zielorientiert. Positive Feedback-

Loops dagegen erzeugen Wachstum oder Schrumpfung, weil die Rückwirkung der Änderung einer Loop-Variablen die Änderung noch verstärkt. Einen einfachen positiven Loop bilden z.B. die Größen „Kapital“ und „Zinsertrag“. Je mehr Kapital verzinst wird, desto höher sind die Zinserträge, die wiederum das Kapital vermehren (exponentielles Wachstum). (Bild 2)

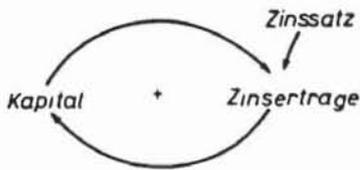


Bild 2 Positiver Feedback-Loop

Diese Einloopstrukturen der beiden Beispiele sind die einfachsten Formen eines Feedback-Systems.

2.2. Die Struktur von System Dynamics-Modellen

Komplexe Systeme haben eine Multiloopstruktur, d.h. sie bestehen aus einer Vielzahl miteinander vermaschter Regelkreise, deren Verhalten somit nicht mehr intuitiv überschaubar ist. Ihre Feedback-Struktur kann am besten durch die Zustandsgrößen des Systems (levels) und die diese Levels verändernden Raten (rates) dargestellt werden. Diese Rates wiederum werden durch Informationen über die Levels bestimmt. Unter Verwendung der dafür gebräuchlichen Symbole läßt sich die Grundstruktur von S.D.-Modellen wie folgt darstellen (Bild 3):

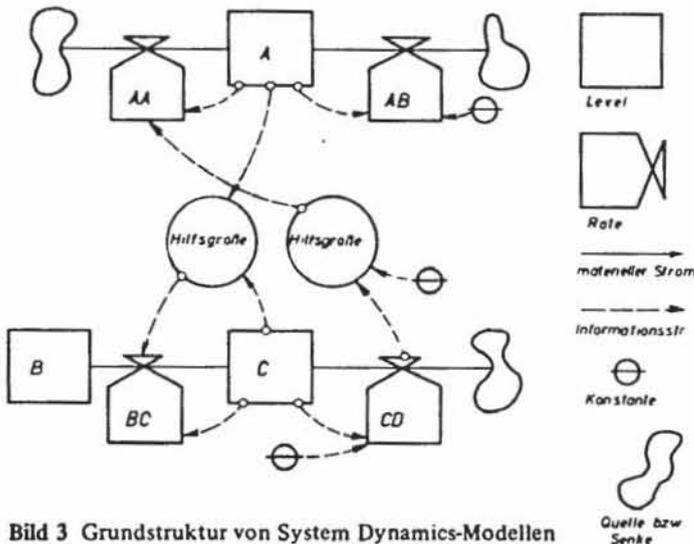


Bild 3 Grundstruktur von System Dynamics-Modellen

Die Kreise stellen Hilfsgrößen dar. Sie sind keine autonomen Größen, sondern werden nur der Übersichtlichkeit wegen eingeführt. Konstante Größen werden durch einen Strich mit einem kleinen Kreis dargestellt. Die Zustandsgrößen werden, wie aus Bild 3 ersichtlich, durch die Input- und Outputströme bestimmt. Ein einfaches Beispiel für einen Level ist die Wassermenge in einem Behälter, in den Wasser ein- und gleichzeitig aus ihm ausfließt.

War die Wassermenge zu einem Zeitpunkt bekannt, so kann durch Integration über den zeitlichen Zu- und Abfluß die Wassermenge zu jedem anderen Zeitpunkt berechnet werden. Die Ströme kennzeichnen den Transport des Inhalts eines Levels zu einem anderen oder in eine Senke hinein bzw. aus einer Quelle heraus. Rates regulieren die Ströme. Sie bestimmen sich aus den verfügbaren Informationen und beinhalten die Motivation einer Entscheidung. Solche Informationen können auch der Verarbeitung des Wertes einer Konstanten oder Hilfsgröße entstammen. Unterschiedliche Ströme sind streng voneinander zu trennen. Wie in anderen Strömen, so können auch in Informationsströmen Levels auftreten, eine solche Ansammlung von Informationen kann z.B. als Erkenntnisgrad interpretiert werden.

Zur Simulation eines Feedback-Systems auf einem Digitalrechner wird ein Gleichungssystem benötigt, das die Struktur des Systems abbildet. Die mathematische Beschreibung eines solchen kontinuierlichen Modells erfolgt durch ein System nichtlinearer gekoppelter Differentialgleichungen mit der Angabe eines Anfangszustandes. Die Behandlung dieses Gleichungssystems auf dem digitalen Rechner erfolgt approximativ durch gekoppelte Differenzgleichungen, die für ein diskretes Zeitintervall gelöst werden. Die numerische Simulation von System Dynamics-Modellen kann prinzipiell mit den bekannten Programmiersprachen, z.B. FORTRAN, erfolgen. Es existiert aber eine eigens zur Vereinfachung der mathematischen Behandlung von S.D.-Modellen entwickelte problemorientierte Simulationssprache, die in den nächsten Abschnitten genauer beschrieben wird.

3. Die Simulationssprache DYNAMO

Von einer leistungsfähigen, problemorientierten Programmiersprache zur Behandlung von S.D.-Modellen auf Digitalrechnern müssen folgende Anforderungen erfüllt werden:

- einfache Anwendbarkeit,
- geringer Speicherplatzbedarf,
- geringe Rechenzeit.

Eine Programmiersprache kann dann als ideal problemorientiert bezeichnet werden, wenn in ihrem Aufbau die gesamte innere Logik des zu behandelnden Problems verarbeitet wurde. Diese Voraussetzungen sind bei DYNAMO gegeben und werden auf folgende Art verwirklicht.

3.1. Berechnungszeitpunkte und Gleichungsarten

Da nichtlineare gekoppelte Differentialgleichungen in Form von Differenzgleichungen gelöst werden sollen, werden Zeitintervalle gleicher Größe DT betrachtet. Daraus ergeben sich drei Berechnungszeitpunkte Vergangenheit, Gegenwart, Zukunft, die mit J, K, L in ihrer Reihenfolge festgelegt werden und die Zeitintervalle JK und KL abgrenzen. Jede auftretende Variable wird durch den entsprechenden Zeitpunkt oder das entsprechende

Zeitintervall gekennzeichnet und erhält damit ihren zeitlichen Bezug in der Form ABC.K oder DEF.JK usw., wobei die Variablen mit bis zu sieben alphanumerischen Zeichen belegt werden können. Als Variable können auftreten

Zustandsgrößen (levels),
 Änderungsraten (rates),
 Hilfsgrößen (auxiliaries).

Entsprechend werden die Gleichungen unterschieden und in der ersten Spalte der Lochkarte gekennzeichnet. Zustandsgleichungen (level equations) werden mit einem L gekennzeichnet. Der Wert eines Levels wird für die Gegenwart (Zeitpunkt K) berechnet. Dabei stehen die Größen des Levels aus der Vergangenheit (Zeitpunkt J) zur Verfügung sowie die Größen der Zu- und Abflußraten im vergangenen Zeitintervall JK. Zusätzlich kann man Hilfsgrößen der Vergangenheit benutzen. Die Gleichung

$$L \text{ POP.K} = \text{POP.J} + \text{DT} * (\text{BR.JK} - \text{DR.JK})$$

mit

$$\text{POP} \hat{=} \text{Bevölkerungszahl [cap]}, \text{BR} \hat{=} \text{Geburtenrate} \left[\frac{\text{cap}}{\text{a}} \right]$$

$$\text{DR} \hat{=} \text{Sterberate} \left[\frac{\text{cap}}{\text{a}} \right], \quad \text{DT} \hat{=} \text{Zeitintervall [a]}$$

ist ein typisches Beispiel für eine Zustandsgleichung. Die Anzahl der Bevölkerung am Ende eines Zeitintervalls ist gleich der Anzahl der Bevölkerung zu Beginn des Zeitintervalls, vermehrt um das Produkt aus der Größe des Zeitintervalls mit der Differenz zwischen Geburten- und Sterberate, die über das Zeitintervall konstant gehalten wurden.

Damit wird die Integralgleichung

$$\text{POP}(t) = \text{POP}(t = T_1) + \int_{T_1}^t (\text{BR}(t) - \text{DR}(t)) dt$$

angenähert. Hieraus ist ersichtlich, daß der richtigen Wahl des Zeitintervalls DT die größte Bedeutung zukommt. Es ist wie bei jeder Diskretisierung kontinuierlicher Vorgänge ein Kompromiß zwischen der erforderlichen Genauigkeit und der vertretbaren Rechenzeit zu schließen.

Ratengleichungen werden mit einem R gekennzeichnet. Sie werden jeweils für das zukünftige Zeitintervall KL berechnet. Zur Verfügung stehen die Werte der Zustands- und Hilfsgrößen der Gegenwart sowie die der Raten des Zeitintervalls JK.

Beispiel:

$$R \text{ STR.KL} = \frac{\text{BEV.K}}{\text{ZEIT}}$$

STR $\hat{=}$ Sterberate, BEV $\hat{=}$ Bevölkerung,
 ZEIT $\hat{=}$ durchschnittliche Lebensdauer.

Hilfsgleichungen werden mit einem A gekennzeichnet und für die Gegenwart berechnet. Konstante (C), Anfangswertgleichungen (N) und Ergänzungsgleichungen (S) werden analog gekennzeichnet.

3.2. Der Rechenzyklus

Die Reihenfolge der Berechnung ist in Bild 4 schematisch dargestellt. Die Pfeile geben an, welche Informationen zur Verfügung stehen.

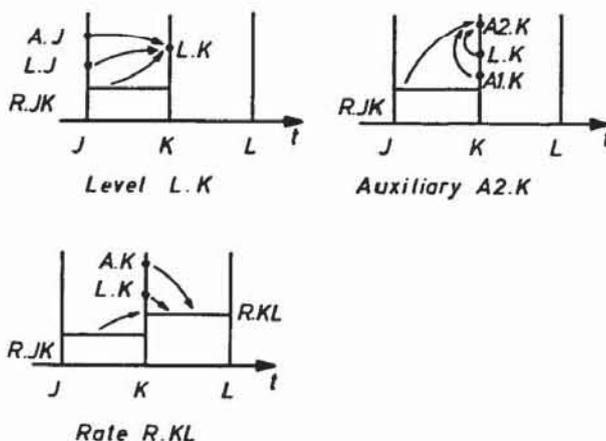


Bild 4 Die Reihenfolge der Berechnung [4]

Zuerst werden die Zustandsgrößen, dann die Hilfsgrößen und zuletzt die Änderungsraten berechnet. Da bis auf die Hilfsgrößen jeweils nur Informationen der Vergangenheit zur Berechnung der Werte verwendet werden, ist die Anordnung dieser Gleichungen im Programm beliebig. Die Hilfsgleichungen müssen ihrer logischen Reihenfolge entsprechend angeordnet werden. Sind alle Werte berechnet, werden die Zeitpunkte auf der Zeitachse um ein Zeitintervall DT in positiver Richtung verschoben, und der Rechenzyklus beginnt von neuem. Dabei werden die alten Werte von den neuen überschrieben, so daß der Speicherplatzbedarf auch bei großen Modellen gering gehalten werden kann.

3.3. Funktionen und Macros

Eine einfache Handhabung ermöglichen weiterhin die eingebauten Funktionen (COS, EXP, LOG, SQRT usw.) und verschiedene Arten von Macros, die sich in vier Kategorien aufteilen lassen:

„Value selection macros“ ermöglichen unter anderem Maximum-Minimum-Abfragen und die Definition von Zusammenhängen durch Tafelfunktionen.

„Time triggered macros“ können zu bestimmten Zeitpunkten Sprungfunktionen, Impulse usw. auslösen.

„Random numbers macros“ gestatten die Simulation von Zufallsereignissen.

Außerdem können eigene Macros verwendet werden. Ihre Definition erfolgt am Anfang des Programms, ihr Aufruf geschieht analog zu den im folgenden beschriebenen Beispielen. Im Rahmen dieses Artikels kann nur auf zwei besonders wichtige Macros näher eingegangen werden.

Zeitverzögerungen bestimmen das Verhalten von Feedback-Loops entscheidend. Sie können immer als Kombination von Änderungsraten und Zustandsgrößen dargestellt werden (Bild 5):

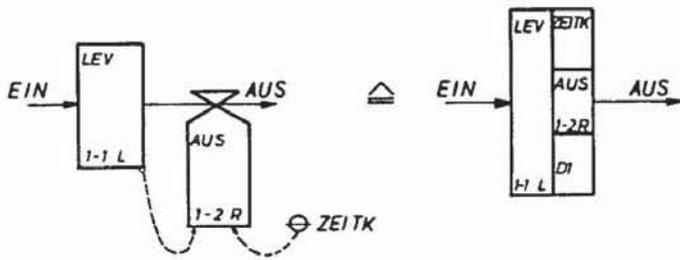


Bild 5 Verzögerungen 1. Ordnung nach [3]

Der Aufruf einer Verzögerung 1. Ordnung erfolgt in einfacher Form:

R AUS.KL = DELAY1 (EIN.JK, ZEITK)

Hierdurch wird die Differentialgleichung

$$\tau \cdot \dot{x} + x = y$$

beschrieben. Dabei entsprechen sich ZEITK und τ , y und EIN, x und AUS.

Durch Lösung des Integrals

$$\bar{T} = \frac{\int_0^{\infty} Ae^{-t/\tau} t dt}{\int_0^{\infty} Ae^{-t/\tau} dt} = \tau$$

kann nachgewiesen werden, daß die Zeitkonstante $\tau = \text{ZEITK}$ der durchschnittlichen Verzögerung \bar{T} einer Eingangsfunktion

$$\sigma(t) = \begin{cases} 0 & \text{für } t < 0 \\ \infty & \text{für } t = 0 \\ 0 & \text{für } t > 0 \end{cases}$$

mit

$$\int_{-\infty}^{+\infty} \sigma(t) dt = A$$

entspricht.

Tafelfunktionen ermöglichen auf eine sehr einfache Weise die Herstellung eines funktionalen Zusammenhangs zwischen zwei Variablen. Der Aufruf erfolgt in der Form

A Y.K = TABLE(YTAB,X.K,A,E,S)

Dabei sind X die unabhängige Veränderliche, Y die abhängige Veränderliche, A der kleinste Wert, E der größte Wert und S die Schrittweite von X. YTAB ist der Name der Tafelfunktion, deren Werte auf einer gesonderten, mit einem T gekennzeichneten Lochkarte festgelegt werden.

Zum Beispiel (Bild 6):

T YTAB = 3/4/6/10/10

Zwischenwerte werden linear interpoliert.

Im Gegensatz zu anderen Programmiersprachen existieren weder eine logische Abfrage noch ein Sprungbefehl, wie z.B. das two- oder tree-branch-logical-IF mit GOTO-Befehl in FORTRAN. Der Grund dafür liegt darin, daß während eines Laufs die Struktur des System Dynamics-Modells nicht verändert werden soll.

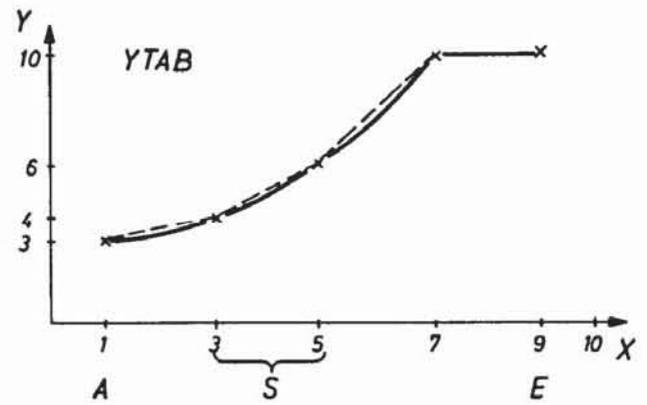


Bild 6 Ein Beispiel für eine Tafelfunktion

3.4. Das Zeitintervall DT

Bereits oben wurde erwähnt, daß der richtigen Wahl des Zeitintervalls DT große Bedeutung zukommt. Als Anhaltspunkt kann man davon ausgehen, daß DT kleiner als die Hälfte der kleinsten auftretenden Zeitkonstanten einer Verzögerung 1. Ordnung ($\frac{1}{6}$ der Zeitverzögerung eines DELAY3) sein muß.

In Bild 7 ist die Einheitssprungantwort einer Verzögerung 1. Ordnung für verschiedene Verhältnisse von DT/ZEITK wiedergegeben.

Es ist ersichtlich, daß eine vernünftige Näherung nur durch $DT/DEL \leq \frac{1}{2}$ erreicht wird.

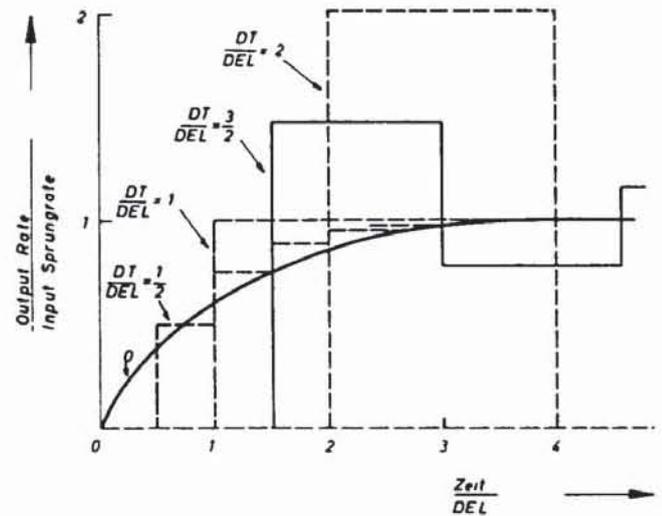


Bild 7 Einheitssprungantwort einer Verzögerung 1. Ordnung mit DT/DEL als Parameter (DT = Lösungszeitintervall, DEL = Zeitkonstante) nach [3]

4. Der DYNAMO-Compiler

4.1. Aufbau

DYNAMO [4] ist ein für System DYNAMics-MODElle auf der Basis von AED (ALGOL Extended for Design) entwickelter Compiler, der die Vorzüge der problembezogenen Darstellung mit schneller Rechenzeit, geringer Speicherplatzanforderung und einfacher Input-Output-Organisation vereint.

DYNAMO ist sowohl im batch- wie im time-sharing-CP/CMS-Betrieb verwendbar. Beide Systeme sind sehr eng miteinander verwandt, obwohl sie vom Benutzer eine völlig andere Bedienung verlangen. Der wesentliche Vorteil der time-sharing-Version liegt in der Möglichkeit, quasi im Dialog mit der Maschine Wiederholungsläufe (reruns) auszuführen, Ergebnisse vorausgegangener Jobs also zeitsparend auszuwerten.

DYNAMO läuft in folgenden sechs Phasen ab:

1. Zuweisung der Anfangswerte und Konstanten entsprechend den Vereinbarungen. Übersetzen des Modells und Konstruktion von Tafelfunktionen entsprechend den verschiedenen vorgesehenen Macros und Steuerungsanweisungen;
2. Entwicklung aller verwendeten Macros und Überprüfung der Symbole;
3. Ordnen und Kodieren der Gleichungen;
4. Ausführung der kodierten Anweisungen;
5. Tabellieren und Plotten der Ergebnisse aus Phase 4;
6. Änderung der Konstanten und Tafelfunktionen für Reruns.

Die sechs Phasen werden durch eine aus AED abgeleitete sowie durch einige speziell für DYNAMO entwickelte Routinen versorgt.

Die Job-Control-Language für DYNAMO hängt von dem entsprechenden Betriebssystem des Rechenzentrums ab, an dem der Compiler betrieben werden soll; einige Anleitungen zur Implementierung sind in [4] enthalten.

4.2. Input-Output-Organisation

Die Input-Output-Organisation läuft in sehr einfacher Weise ab. DYNAMO unterscheidet nicht zwischen Quellprogramm- und Datenkarten, sondern weist die Daten den Konstanten unmittelbar, den Levels über die Anfangswertgleichungen zu. Die Ausgabe der Ergebnisse erfolgt über eingebaute Print- und Plotprogramme, die durch die Befehle PRINT bzw. PLOT aufgerufen werden. Sie übernehmen die tabellierte bzw. graphische Darstellung der durch Name, Zeichensymbol und -maßstab festgelegten Output-Variablen. Als letzte Karte eines Kartensatzes folgt die RUN-Anweisung, der Befehl zum Programmstart.

Mehrere Wiederholungsaufrufe (Reruns) können dem RUN-Statement angehängt werden, um mit veränderten Konstanten und Tafelfunktionen, aber derselben Modellstruktur (d.h. unveränderten Levels, Rates, Auxiliaries) zu arbeiten.

4.3. Speicherplatzbedarf und Rechenzeiten

Naturgemäß hängt die Anforderung an Kernspeicherplatz und Rechenzeit in erster Linie von der Größe und Art des zu bearbeitenden Programmes ab. Grundsätzlich gilt für DYNAMO folgendes:

Dank der AED-Basis des Compilers sowie des einfachen Aufbaus der Simulationssprache, vor allem aber wegen

der Überspeicherung sämtlicher zeitabhängigen Variablen sind die Operationen nicht kernspeicherintensiv. Aufgrund der Möglichkeit, mit Hilfe von Reruns gleiche Programme mit verschiedenen Parameterwerten zu bearbeiten, läßt sich Übersetzungszeit sparen. Daher ist auch die erforderliche Rechenzeit für Programme mit einer Vielzahl variierteter Parameter sehr gering.

Um einen Anhaltspunkt für die Größenordnung von Speicherplatzbedarf und Rechenzeit zu geben, werden zwei Anwendungsfälle verglichen:

- Nach dem Loaden des Compilers wird für die Ausführung eines kleinen Programms aus 30 S.D.-Gleichungen (5L, 5N, 6R, 3A, 1T, 10C) (Programme des in Abschnitt 5 beschriebenen Modells) bei einer Region Size von 106 kbytes für den Basislauf 0,74 sec CPU-Zeit, für jeden Rerun 0,32 sec CPU-Zeit benötigt.
- Für ein größeres Programm aus 114 S.D.-Gleichungen (5L, 6N, 7R, 31A, 22T, 42C, 1S) [7] sind bei einer Region Size von 114 kbytes für den Basislauf 2,81 sec CPU-Zeit, für jeden Rerun 2,35 sec CPU-Zeit erforderlich.

5. Erläuterung von S.D. an einem Beispiel

Im folgenden wird die Anwendung von S.D. an einem einfachen Beispiel erläutert. Betrachtet wird die Ausbreitung einer fiktiven Krankheit.

5.1. Modellbeschreibung

Es wird angenommen, daß Untersuchungen folgende medizinische Daten ergeben haben: Die Krankheit ist ansteckend, und die Übertragung der Krankheitserreger erfolgt durch persönlichen Kontakt. Im Mittel beträgt die Gesundungszeit 4 Wochen. Schwere Krankheitsverläufe führen nach durchschnittlich einer Woche zum Tod. Es besteht eine Impfmöglichkeit gegen diese Krankheit mit einer Immunitätszeit von einem Jahr.

Es wird vorausgesetzt, daß statistische Untersuchungen folgende Daten ergeben haben: 10% aller Krankheitsfälle nehmen einen tödlichen Verlauf. Die Impfbereitschaft der Bevölkerung ist abhängig von dem zahlenmäßigen Verhältnis der Kranken und Gesunden. Diesen Zusammenhang stellt die Tafelfunktion IMPFB dar. Die Impfbereitschaft beträgt 5%, wenn niemand krank ist. Treten Krankheitsfälle auf, steigt die Impfbereitschaft stark an, sie beträgt 90% bei einem Verhältnis Kranke zu Gesunde von 0,01, steigt dann langsamer bis 99% bei einem Verhältnis von 0,02 und bleibt weiterhin auf diesem Wert, da sich 1% der Einwohner unter keinen Umständen impfen lassen. Die durchschnittliche Zeit, die vergeht von dem Beschluß, sich impfen zu lassen bis zu dem Zeitpunkt der Immunität, beträgt eine Woche. Anhand dieser Werte und einer Ansteckungskonstanten läßt sich das Flußdiagramm eines Epidemiemodells nach Bild 8 darstellen.

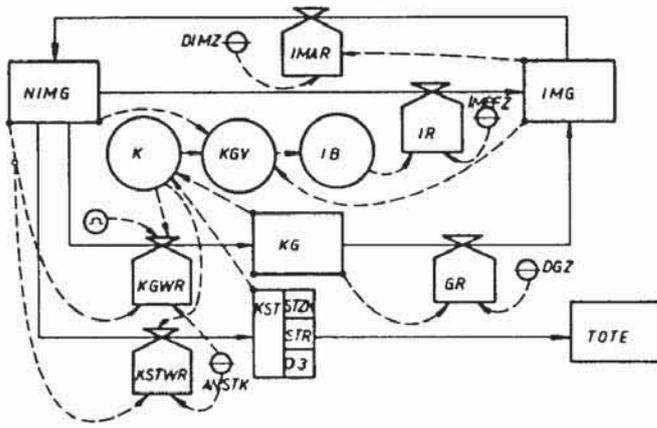


Bild 8 Flußdiagramm des Epidemiemodells

Dabei wurden folgende Bezeichnungen gewählt:

- IMG ≙ Anzahl der immunen Gesunden,
- NIMG ≙ Anzahl der nichtimmunen Gesunden,
- TOTE ≙ Anzahl der Toten,
- KG ≙ Anzahl der Kranken, die wieder gesund werden,
- KST ≙ Anzahl der Kranken, die sterben werden,
- IMAR ≙ Immunitätsaufhebungsrate,
- IR ≙ Impfrate,
- GR ≙ Gesundungsrate,
- STR ≙ Sterberate,
- KGWR ≙ Krankwerderate für KG,
- KSTWR ≙ Krankwerderate für KST,
- K ≙ Gesamtanzahl der Kranken,
- KGV ≙ Verhältnis Kranke/Gesunde,
- IB ≙ Impfbereitschaft,
- DIMZ ≙ Durchschnittliche Immunitätszeit,
- IMPFZ ≙ Impfzeitkonstante,
- DGZ ≙ Durchschnittliche Gesundungszeit,
- STZK ≙ Sterbezeitkonstante,
- ANSTK ≙ Ansteckungskonstante.

Die Krankwerderaten ergeben sich aus folgender Überlegung: Die größtmögliche Anzahl an Kontakten, bei denen Krankheitserreger übertragen werden können, ist das Produkt aus Anzahl der Kranken und Anzahl der nichtimmunen Gesunden. Es wird angenommen, daß von dieser größtmöglichen Zahl an Kontakten innerhalb einer Woche nur ein geringer Prozentsatz zustande kommt und daß von diesen wiederum nur ein geringer Prozentsatz zur Ansteckung führt.

5.2. Gleichungen

Die Übertragung des Flußdiagramms (Bild 8) in die Programmiersprache DYNAMO führt zu dem in Bild 9 dargestellten Gleichungssystem. Die einzelnen Gleichungen werden im folgenden näher erläutert:

- L IMG.K - IMG.J + DT * (GR.JK + IR.JK - IMAR.JK)
- N IMG = IMGA
- C IMGA = 52000

Die Anzahl der immunen und damit gesunden Einwohner zum Zeitpunkt der Gegenwart (K) ergibt sich aus ihrer Anzahl zum letzten berechneten Zeitpunkt (J), vermehrt um die Genesenen und die Neugeimpften sowie vermindert um diejenigen, deren Immunität unwirksam wird. Die Festlegung der Anfangswerte wird sinnvollerweise in der oben angegebenen Form vorgenommen, da es nicht möglich ist, in kostensparenden reruns N-Gleichungen zu verändern, wohl aber C-Gleichungen.

- R IMAR.KL = IMG.K/DIMZ
- C DIMZ = 52

EPIDEMIEMODELL 10/26/72

```

* EPIDEMIEMODELL
NOTE
L IMG.K=IMG.J+DT*(GR.JK+IR.JK-IMAR.JK)  PERSCHEN
N IMG=IMGA  PERSCHEN
C IMGA=52000  PERSCHEN
R IMAR.KL=IMG.K/DIMZ  PERSCHEN/MCCHE
C DIMZ=52  MCCHE
L NIMG.K=NIMG.J+DT*(IMAR.JK-KGWR.JK-KSTWR.JK-IR.JK)  PERSCHEN
N NIMG=NIMGA  PERSCHEN
C NIMGA=200000  PERSCHEN
L KGV.K=KGV.J+DT*(KGWR.JK-GR.JK)  PERSCHEN
N KGV=KGA  PERSCHEN
C KGA=0  PERSCHEN
R STR.KL=DELAY3(KSTWR.JK,STZK)  PERSCHEN/MCCHE
C STZK=1  MCCHE
L KST.K=KST.J+DT*(KSTWR.JK-STR.JK)  PERSCHEN
N KST=KSTA  PERSCHEN
C KSTA=0  PERSCHEN
A K.K=KGV.K+KST.K  PERSCHEN
R GR.KL=KGV.K/DGZ  PERSCHEN/MCCHE
C DGZ=4  MCCHE
L TOTE.K=TOTE.J+DT*(STR.JK)  PERSCHEN
N TOTE=TOTE0  PERSCHEN
C TOTE0=0  PERSCHEN
R KGR.KL=K.K*(NIMG.K/ANSTK+D.9+PLLE11/DT.5.300)  PERSCHEN/MCCHE
N KGR=KGR0  PERSCHEN/MCCHE
C ANSTK=50000  PERSCHEN/MCCHE
A KGV.K=NIMG.K/IMG.K  PERSCHEN
C IB=TABLE1(IMPFZ,KGV.K,D.0,C.02,D.001)  PERSCHEN
T IMPFZ=D.05/C.01/C.03/C.05/D.1/D.3/D.6/D.8/D.0.8/D.0.85/D.9/D.91/D.92/
R IR.KL=IB.K*(NIMG.K/IMPZ)  PERSCHEN/MCCHE
C IMPZ=1  MCCHE
NOTE
PR INT NIMG,IMG,K,KGV,KST,TOTE,KGWR,KSTWR,IR,GR,IR
PLOT NIMG=N/IMG=1/K=1/C,200001/TOTE=T/0.0001/AGWR=C/10,20000/1/
R KSTWR=STC,20001/IR=N/IB=0
SPEC CT=0.1/LENGTH=5C/PRTPER=1/PLOTPER=1
RUN BASISLALP

C ANSTK=50000  PERSCHEN/MCCHE
RUN VERBESSERUNG DER QUARANTAENE

```

Bild 9 Die vollständigen Gleichungen zum Epidemiemodell

Vereinfacherweise wurde hier angenommen, daß die Immunitätsaufhebungsrate proportional der Anzahl der Immunen ist. Die durchschnittliche Immunitätszeit von 52 Wochen läßt also auch zu, daß ein Teil der geimpften Personen von Anfang an keine Immunität erhält. Statt der hier entstehenden Verzögerung 1. Ordnung wäre bei einer sicheren Impfmethode auch eine Verzögerung 3. Ordnung denkbar.

- L NIMG.K = NIMG.J + DT * (IMAR.JK - KGWR.JK - KSTWR.JK - IR.JK)
- N NIMG = NIMGA
- C NIMGA = 200 000

Die Anzahl der nicht immunen, aber trotzdem gesunden Einwohner wird beeinflußt von der Immunitätsaufhebungsrate, den Krankwerderaten und der Impfrate.

- L KG.K = KG.J + DT * (KGWR.JK - GR.JK)
- N KG = KGA
- C KGA = 0

Aus der Differenz zwischen Krankwerde- und Gesundungsrate ergibt sich die Anzahl derjenigen Kranken, die wieder genesen werden.

- R STR.KL = DELAY3(KSTWR.JK,STZK)
- C STZK = 1

Die Sterberate ergibt sich als Verzögerung 3. Ordnung aus dem Prozentsatz der Krankwerderate, deren Krankheit einen tödlichen Verlauf nimmt. Eine Verzögerung 3. Ordnung erscheint hier angebracht, wenn die Krankheit in der Regel nach einer Woche zum Tod führt und kürzere Zeiten relativ selten sind.

$$R \text{ KGWR.KL} = (K.K * NIMG.K/ANSTK) 0.9 + \text{PULSE} (1/DT, 5,300)$$

$$R \text{ KSTWR.KL} = (K.K * NIMG.K/ANSTK) 0.1$$

$$C \text{ ANSTK} = 50\ 000$$

Die Krankwerderaten ergeben sich aus folgender Überlegung: Die größtmögliche Anzahl an Kontakten, bei denen Krankheitserreger übertragen werden können, ist das Produkt aus Anzahl der Kranken und Anzahl der nicht-immunen Gesunden. Durch die Festlegung der Ansteckungskonstanten wird angenommen, daß von dieser maximal möglichen Zahl an Kontakten innerhalb einer Woche z.B. nur 1 % stattfindet und von diesen nur jeder 50ste Kontakt zur Ansteckung führt. Die hier angegebene PULSE-Funktion erzeugt nach 5 Zeiteinheiten (hier Wochen) einen Krankheitsfall. Mit der Angabe 300 wird die Impulsfrequenz festgelegt, die in diesem Fall so groß gewählt werden muß, daß der zweite Impuls außerhalb des Rechnungszeitraumes liegt.

Die restlichen Systemgleichungen ergeben sich analog zu den oben beschriebenen.

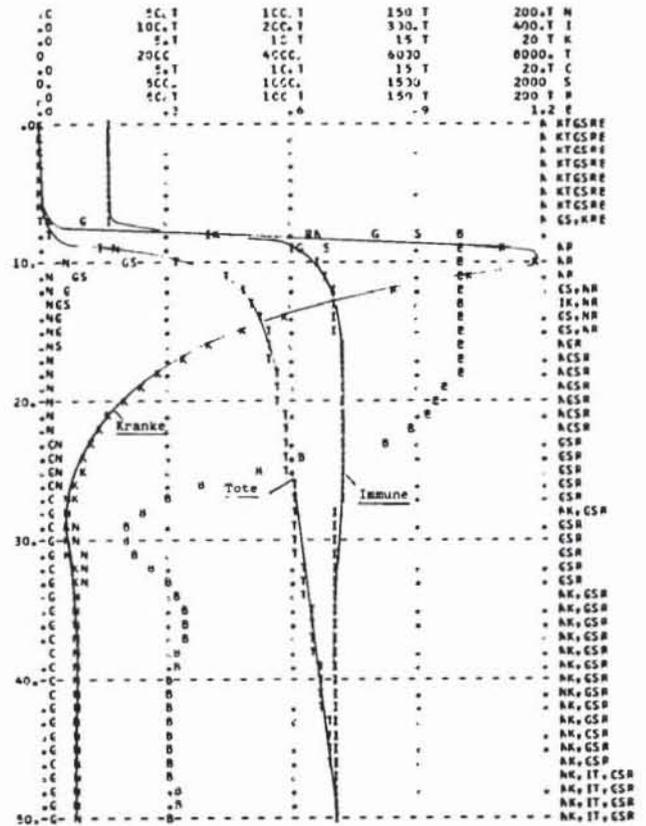


Bild 10 Standardlauf des Epidemiemodells mit ANSTK = 50 000

5.3. Simulationsläufe

Die ausgeführten Simulationsläufe beziehen sich auf einen begrenzten Raum mit 252 000 Einwohnern, von denen 200 000 nicht geimpft sind und niemand krank ist. Nach 5 Wochen tritt ein Krankheitsfall auf, der eine Epidemie auslöst.

Der Verlauf der Epidemie wurde simuliert. Dabei stellte sich heraus, daß die Ausbreitung einer Epidemie in erster Linie von der Ansteckungskonstanten abhängt. Der erste Lauf (Bild 10) beruht auf einer Ansteckungskonstanten von 50 000 Personen * Wochen. Das heißt, daß von den maximal möglichen Krankheitsübertragungen nur der 50 000ste Teil in einer Woche stattfindet. Im zweiten Lauf (Bild 11) wurden die Ansteckungsmöglichkeiten um den Faktor 2 gesenkt, was zu einer Ansteckungskonstanten von 100 000 Personen * Wochen führt. Durch diese Maßnahme (z.B. durch Isolierung der Kranken) konnte die Anzahl der tödlichen Krankheitsverläufe auf etwa ein Drittel gesenkt werden.

6. Anwendung von System Dynamics

Die Anwendung der Simulationmethode System Dynamics als Instrument der Entscheidungsfindung bei komplexen Systemen ist prinzipiell nicht auf spezielle Aufgabengebiete beschränkt. Vor allem eignet sich S.D. für solche Probleme, die wegen einer Vielzahl an Rückkopplungsmechanismen mit anderen Verfahren nicht befriedigend lösbar sind. Primäres Ziel aller Untersuchungen

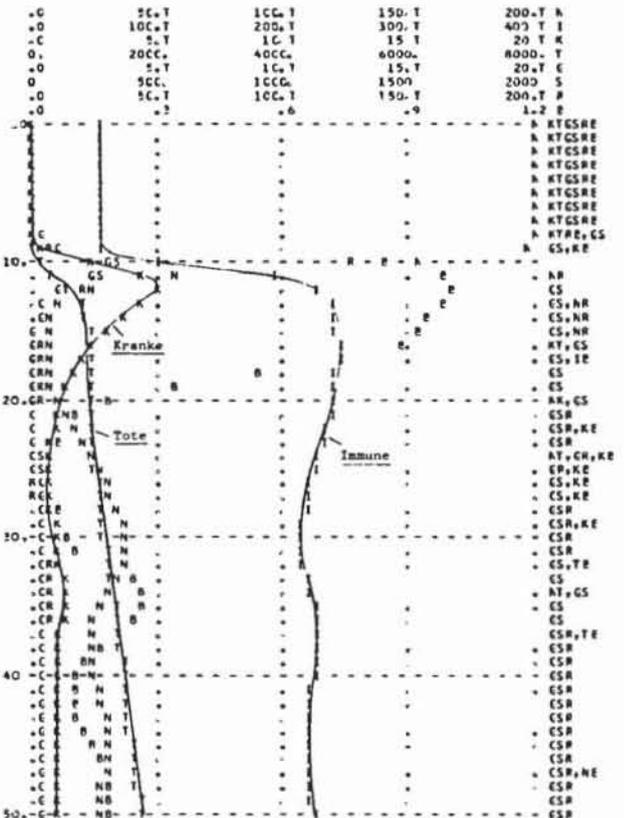


Bild 11 Modifizierter Lauf des Epidemiemodells mit ANSTK = 100 000

ist die Analyse der Struktur, des Verhaltens und der Entwicklung eines Systems unter ganz bestimmten Bedingungen, nicht jedoch die Erstellung einer Prognose. Bisher ist eine Fülle an ökonomischen, technischen, soziologischen und ökologischen Fragen mit S.D. erörtert worden; aus dieser Reihe ist im folgenden eine repräsentative Auswahl zusammengetragen:

- Untersuchung kritischer Weltprobleme wie Bevölkerungsexplosion, Nahrungsmittel- und Rohstoffverknappung, Umweltverschmutzung etc.
In zwei in der Presse intensiv diskutierten Veröffentlichungen von FORRESTER [6] und dem Team MEADOWS [5], zu dem eine Reihe noch unveröffentlichter Voruntersuchungen gehört, wird die Entwicklung des Weltsystems analysiert, und es werden Maßnahmen für eine Gleichgewichtsentwicklung interpretiert. Kern der Aussagen ist die Erkenntnis, das auf einem begrenzten Planeten mit begrenzten Ressourcen kein unendliches Wachstum möglich ist, wobei der „point of no return“ heute bereits in greifbarer Nähe liegt.
- Untersuchung der Wachstums- und Stagnationsverhältnisse in Stadtgebieten [7].
In dieser Studie werden das Sozialsystem „Großstadt“ durch eine Reihe soziologischer und ökonomischer Faktoren beschrieben, die negativen Folgen verschiedener Sanierungsprogramme gezeigt und die notwendigen Bedingungen zur Verbesserung des Systems erarbeitet.
- Untersuchung der Industrialisierung, der Beschäftigungslage und des Bevölkerungswachstums in einer Region [8].
Diese vom Battelle-Institut herausgegebene Arbeit wendet System Dynamics für die Regionalplanung im Susquehanna-Einzugsgebiet an. Ausgehend von der Bevölkerungsentwicklung, werden das Arbeitskräftepotential und, über regionsspezifische Standortfaktoren, das Arbeitsplatzangebot ermittelt. Die durch die Beschäftigungslage bestimmte Wirtschaftsentwicklung beeinflusst die ökonomischen und ökologischen Bedingungen des Flußsystems und wirkt damit auf Ansiedlung von Unternehmen und Wanderung der Bevölkerung zurück.
- Untersuchung des Wachstums industrieller Unternehmen [9] und Untersuchung der Marktwachstumsmechanismen [10]

In diesen Modellstudien werden die Faktoren analysiert, die die Entwicklung von Unternehmen wachstumsfördernd oder -hemmend beeinflussen. Aufgabe der Untersuchungen ist es, die zur Realisierung langfristiger Ziele notwendigen unternehmenspolitischen Entscheidungen sichtbar zu machen.

- Untersuchungen zu Problemen der Ressourcenverknappung, Umweltbelastung und Rezyklierung [11]
In mehreren Studien werden z.B. der DDT-Kreislauf, der Quecksilberzyklus und die Bedingungen für eine Rezyklierung von Kupfer analysiert.
- Untersuchungen zur Energiesituation
In einer von den Verfassern dieses Beitrags begonnenen Studie wird der Zusammenhang zwischen Wirtschaftswachstum, Bevölkerungsentwicklung und Energieverbrauch diskutiert. Dabei spielen unter anderem Fragen der Ressourcenerschöpfung, der Umweltbelastung und der Rohstoffversorgung eine Rolle.

Literatur

- [1] *Stahlknecht, P.*: Operations Research. Friedr. Vieweg + Sohn, Braunschweig 1970.
- [2] *Forrester, J. W.*: Principles of Systems. Wright-Allen Press., 1969.
- [3] *Forrester, J. W.*: Industrial Dynamics. M.I.T.-Press., 1969.
- [4] *Pugh, A.L. III*: DYNAMO II User's Manual. MIT Press, Cambridge, Mass., USA, 1970.
- [5] *Meadows, D., et al.*: The Limits of Growth. Universe Books, New York 1972. (Die Grenzen des Wachstums. Deutsche Verlagsanstalt, Stuttgart 1972)
- [6] *Forrester, J. W.*: World Dynamics. Wright-Allen Press, Cambridge, Mass. 1971.
- [7] *Forrester, J. W.*: Urban Dynamics. MIT. Press., Cambridge, Mass., 1969.
- [8] *Hamilton, H. R., et al.*: Systems Simulation for Regional Analysis. An Application to River Basin Planning. MIT Press., Cambridge, Mass., 1969.
- [9] *Zahn, E.*: Das Wachstum industrieller Unternehmen. Th. Gabler, Wiesbaden 1971.
- [10] *Forrester, J. W.*: Market Growth as Influenced by Capital Investment. Industrial Management Review, Vol. IX, No. 2, Winter 1968, pp. 83–105.
- [11] *Randers, H., et al.*: System Simulation to test Environmental Policy. Mimeographed. MIT Cambridge, Mass.