Katzenvermehrung: Eine Facette mathematischer Modellbildung mit Hilfe von Computeralgebra Software

Jochen Ziegenbalg, Karlsruhe

Den Heidelberger Kollegen Göhner, Hofsäß, Mauve und Schönbeck gewidmet

Die Fähigkeit, mathematische Kenntnisse und Methoden als Werkzeug, Strukturierungs- und Orientierungsmittel in Alltag, Beruf und Umwelt anwenden zu können, ist eines der vorrangigen Ziele mathematischer Bildung. Als besonders schwierig wird in diesem Zusammenhang stets die Phase der Mathematisierung bzw. der mathematischen Modellbildung erlebt, denn sie ist kaum „kanonisierbar". Der Prozess der Mathematisierung setzt in der Regel

- ausreichendes (sowohl inhaltliches als auch methodologisches) mathematisches Grundwissen,
- Kenntnisse über den zu mathematisierenden Bereich,
- Kenntnisse über den Prozess der Modellbildung selbst

und nicht zuletzt auch

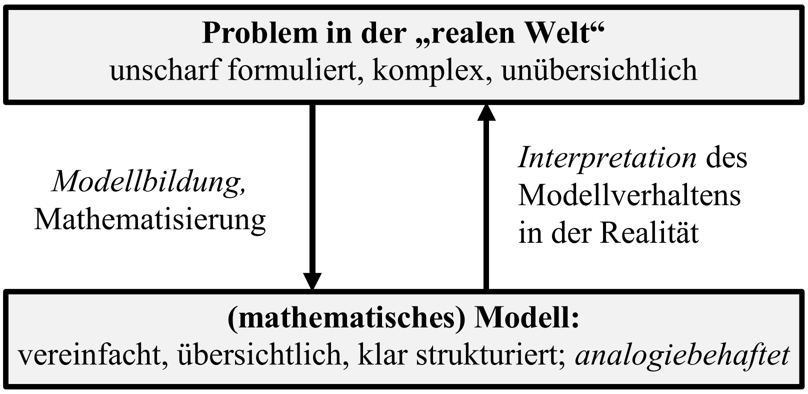
- sprachliche Fähigkeiten (denn jede Mathematisierung ist auch eine „Textaufgabe"), sowie
- eine gewisse geistige Reife im Zusammenhang mit der Notwendigkeit, flexibel zwischen der Ebene des zu mathematisierenden Objektbereichs (der „realen Welt") und der Ebene des mathematischen Modells hin und her zu pendeln

voraus.

Zur Mathematisierung gehört schließlich auch die (Rück-) Interpretation der mit mathematischen Verfahren erzielten Ergebnisse in die Ebene der realen Welt; sie verlangt in diesem Zusammenhang auch noch eine gute Portion (curricular schwer festzumachenden) gesunden Menschenverstandes.

Zur Veranschaulichung des Wechselspiels zwischen Mathematisierung und Interpretation wird häufig eine Darstellung der folgenden Art herangezogen.

Grundschema der (mathematischen) Modellbildung



Jeder „unterrichtlich“ eingefädelte Modellbildungsprozess hängt ganz entscheidend von dem zu mathematisierenden Gegenstandsbereich ab. Es ist hilfreich, wenn das zu mathematisierende Problem in „skalierbarer“ Form angegangen werden kann; damit ist gemeint, dass es zunächst auf einer möglichst elementaren Ebene zugänglich ist, dass aber nach einer ersten „Windung“ auf der Mathematisierungsspirale auch die Notwendigkeit zu einer weitergehenden, differenzierteren Behandlung des Themas deutlich wird.

Zeitungsartikel oder Nachrichtensendungen enthalten oft ein überraschendes Mathematisierungs-Potential. Hierzu ein amüsantes Beispiel aus der ZDF-Nachrichtensendung „heute“:

> Smog-Alarm in Paris – nur Autos mit ungerader Endziffer dürfen fahren, die Autos mit gerader Endziffer nicht. Es gab Unklarheiten: Ist die Endziffer 0 eine gerade Zahl? Fahrer mit der Endziffer 0 gingen straffrei aus, denn die Polizei wusste auch keine Antwort.
> (*ZDF Nachrichtensendung ”heute”* 1. Oktober 1997, 19 Uhr)

Aber auch bei ganz normalen „Erledigungen“ in der Stadt stößt man gelegentlich auf überraschende Funde. Vor einiger Zeit fiel mir im Tierheim Reutlingen das folgende Informationsblatt in die Hände.

Machen Sie sich mitschuldig?

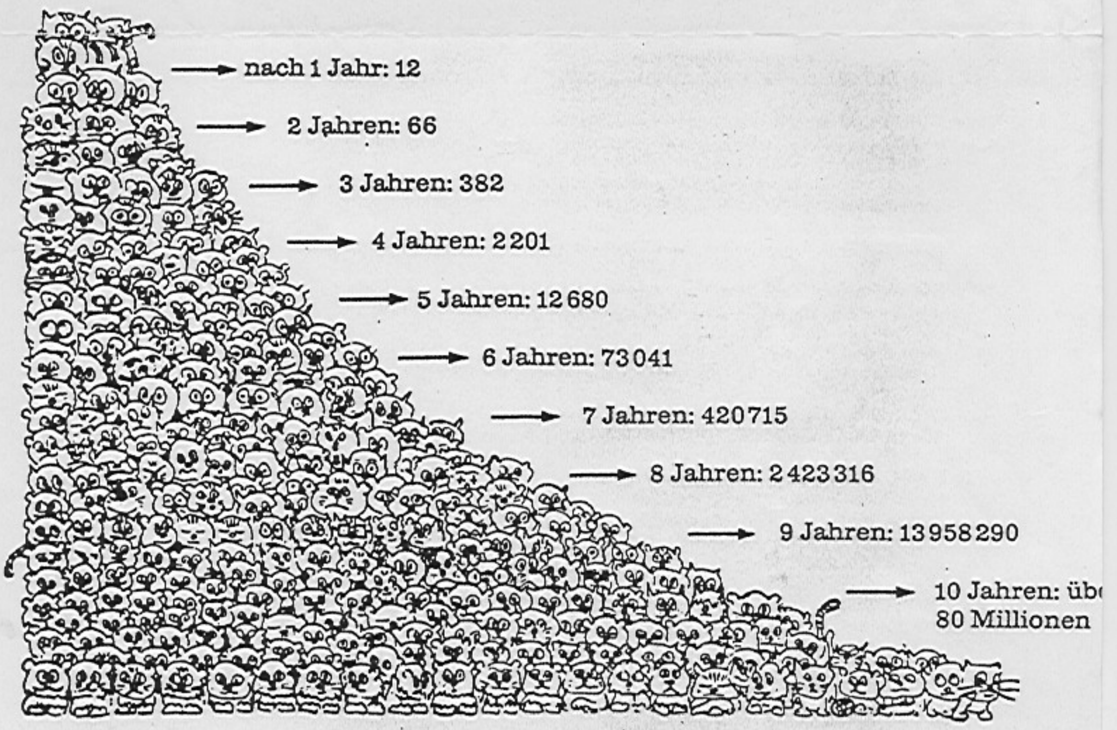
Millionen unerwünschter Katzen werden jedes Jahr geboren. Ende April bis September, der Hauptzeit für Katzengeburten, muß man herrenlose Katzen in Tierheimen sogar töten!

Weniger „glückliche“ Tiere streunen umher. Ein Teil von ihnen wird überfahren, erschossen, mißhandelt. Viele landen als Versuchstiere in Labors. Die Überlebenden werden sehr schnell geschlechtsreif und gebären 5 oder 6 Kätzchen. So beginnt der Teufelskreis!

Jeder Katzenbesitzer, der sein Tier nicht sterilisieren läßt, macht sich mitschuldig.

Denken Sie daran: der Nachwuchs einer weiblichen Katze kann nach <u>10 Jahren über 80 Millionen Tiere betragen!!!</u>

Nimmt man an, ein Katzenpaar bekommt im Jahr zweimal Nachwuchs und jeweils 2,8 Kätzchen pro Wurf überleben, dann ergibt das nach 10 Jahren über 80 MILLIONEN Kätzchen!



Ich nahm das Blatt immer wieder gern zum Anlass, um in meinen Lehrveranstaltungen verschiedene Aspekte der (mathematischen) Modellbildung zu illustrieren, insbesondere

- den Prozess des Mathematisierens
- Validierung von Modellen, kritischer Umgang mit eigenen und fremden Modellen
- Probleme der Datenerhebung
- Modellbildung als Regelkreis (mit dem Ziel der Modellverbesserung)
- den Einsatz mathematischer Werkzeuge, insbesondere des Computers und seiner Software

  Im folgenden werden nun einige der Möglichkeiten zur Modellbildung anhand dieses Informationsblattes diskutiert.

Der Prozess des Mathematisierens

Ein Vorteil des Informationsblattes ist, dass man als „Unterrichts-Organisator" gar nicht viel dazu sagen muss; das Blatt ist weitgehend selbsterklärend und wirkt nicht zuletzt auch durch seine lustige Aufmachung selbstmotivierend.

Ein erster Ansatz zur Mathematisierung ist direkt auf dem Informationsblatt gegeben:

> *Nimmt man an, ein Katzenpaar bekommt im Jahr zweimal Nachwuchs und jeweils 2,8 Kätzchen pro Wurf überleben, dann ergibt dies nach 10 Jahren über 80 Millionen Kätzchen!*

Das Modell ist so einfach, dass die meisten Studierenden recht unvermittelt beginnen, die Zahlen (mit dem Taschenrechner) nachzurechnen. Sie kommen dabei nach einiger Zeit zu einer tabellenartigen Darstellung mehr oder weniger in der folgenden Form:

Jahre	alte Katzen	alte Katzenpaare	Vermehrungs-Faktor	neue Katzen	Katzen insgesamt
	2	1	mal 2,8	2,8	4,8
1	4,8	2,4	mal 2,8	6,72	11,52
	11,52	5,76	mal 2,8	16,128	27,648
2	27,648	13,824	mal 2,8	38,7072	66,3552
	66,3552	33,1776	mal 2,8	92,89728	159,25248
3	159,25248	79,62624	mal 2,8	222,953472	382,205952
	382,205952	191,102976	mal 2,8	535,0883328	917,2942848
4	917,2942848	458,6471424	mal 2,8	1284,211999	2201,506284
...	...	...	...	...	...

Spätestens zu diesem Zeitpunkt sind die Studierenden davon überzeugt, dass sie das auf dem Informationsblatt gegebene Modell „durchschaut" haben. Einige von ihnen erklären noch die durch die Kommazahlen gegebenen Unterschiede: Rundet man die Zahlen in der rechten Spalte, so entsprechen sie den Zahlen des Informationsblattes (fast, jedenfalls, denn die letzte Zahl müsste dort dann eigentlich 2202 lauten). Die meisten sind jedoch geneigt, den kleinen Unterschied irgendwelchen numerischen Rundungseigenschaften oder Instabilitäten des verwendeten Werkzeugs zuzuschreiben (man weiß ja auch gar nicht, mit welchem Werkzeug die Zahlen des Informationsblattes ermittelt wurden).

Diejenigen Studierenden, die mit Tabellenkalkulationsprogrammen umgehen können, erhalten den Auftrag, die Taschenrechner-Rechnungen auf dieses Softwarewerkzeug zu übertragen – und man stellt in der Tat fest, dass auch die restlichen Zahlen „stimmen“.

Jahre	alte Katzen	alte Katzen-paare	Vermeh-rungs-Faktor	neue Katzen	Katzen insgesamt	Katzen insgesamt (ganzzahlig)
	2,00	1,00	2,8	2,80	4,80	5
1	4,80	2,40	2,8	6,72	11,52	12
	11,52	5,76	2,8	16,13	27,65	28
2	27,65	13,82	2,8	38,71	66,36	66
	66,36	33,18	2,8	92,90	159,25	159
3	159,25	79,63	2,8	222,95	382,21	382
	382,21	191,10	2,8	535,09	917,29	917
4	917,29	458,65	2,8	1.284,21	2.201,51	2.202
	2.201,51	1.100,75	2,8	3.082,11	5.283,62	5.284
5	5.283,62	2.641,81	2,8	7.397,06	12.680,68	12.681
	12.680,68	6.340,34	2,8	17.752,95	30.433,62	30.434
6	30.433,62	15.216,81	2,8	42.607,07	73.040,69	73.041
	73.040,69	36.520,35	2,8	102.256,97	175.297,67	175.298
7	175.297,67	87.648,83	2,8	245.416,73	420.714,40	420.714
	420.714,40	210.357,20	2,8	589.000,16	1.009.714,57	1.009.715
8	1.009.714,57	504.857,28	2,8	1.413.600,39	2.423.314,96	2.423.315
	2.423.314,96	1.211.657,48	2,8	3.392.640,94	5.815.955,90	5.815.956
9	5.815.955,90	2.907.977,95	2,8	8.142.338,26	13.958.294,16	13.958.294
	13.958.294,16	6.979.147,08	2,8	19.541.611,82	33.499.905,98	33.499.906
10	33.499.905,98	16.749.952,99	2,8	46.899.868,37	80.399.774,36	80.399.774

Damit geben sich viele zufrieden, denn sie haben ja offenbar das Modell verstanden.

Modellkritik

Einige Studierende geben sich damit jedoch nicht zufrieden. Sie betrachten das Ergebnis mit dem gesunden Menschenverstand und stellen fest: „Das kann ja nicht sein, denn sonst müsste ja die Erde schon längst komplett mit Katzen überschwemmt sein“. Bald werden auch einige Gründe genannt, warum das Modell auf dem Informationsblatt so nicht funktionieren kann:

- In dem Modell kommt es überhaupt nicht vor, dass Katzen auch mal sterben.
- Die neugeborenen Katzen werden sofort geschlechtsreif.
- Der Aspekt der Ressourcenbegrenzung (Futterknappheit, Raumknappheit) ist nicht berücksichtigt.

Es entsteht die Frage, ob und ggf. woher man eigentlich weiß, wie viele Katzen es z.B. in Deutschland oder auf einem bestimmten Areal gibt. Eine Nachfrage bei den Biologen führt nur zu der Erkenntnis, dass es dazu bestenfalls sehr grobe Schätzwerte gibt.

Biologisch versiertere Studierende weisen darauf hin, dass die Kater im Hinblick auf die Zahlen bei der Katzenvermehrung eigentlich gar keine Rolle spielen. Die Geburtenzahlen hängen fast ausschließlich von der Zahl der Weibchen ab, denn nur sie (und nicht die Kater) stellen aus biologischen Gründen den „Engpass“ bei der Vermehrung dar. Numerisch macht die Berücksichtigung dieses Umstandes aber praktisch keinen Unterschied im Vergleich zu den Zahlen des Informationsblattes, wenn man davon ausgehen kann, dass es ungefähr gleich viele männliche und weibliche Katzen gibt (die Anzahl der Katzenpaare entspricht dann der Zahl der weiblichen Katzen).

Modellverbesserung

Von den verschiedenen Verbesserungsvorschlägen wird zunächst der Aspekt der verzögerten Geschlechtsreife realisiert. In einem höchst einfachen Modell mit verzögerter Geschlechtsreife wird angenommen, dass die neugeborenen Katzen im ersten Halbjahr ihres Lebens noch nicht geschlechtsreif sind. Nach dem ersten Halbjahr werden sie dann geschlechtsreif. Man hat es dann mit folgenden Populations-Bestandteilen (jeweils bezogen auf einen bestimmten Zeitpunkt) zu tun:

- alle Katzen (K)
- die geschlechtsreifen Katzen (G)
- die geschlechtsreifen weiblichen Katzen (W)
- die neugeborenen und noch nicht geschlechtsreifen Katzen (N)
- die zuletzt geschlechtsreif gewordenen Jungkatzen (J); das sind diejenigen Katzen, die sich im zweiten Halbjahr ihres Lebens befinden.

Mit diesen Bezeichnungen lässt sich nun der Übergang von einer Periode (= Halbjahr) in die nächste Periode wie folgt beschreiben. Mit dem Index „alt“ bzw. „neu“ sind die jeweiligen Größen in der alten bzw. in der neuen Periode gemeint.

$$N_{neu} = 2{,}8 \cdot W_{alt}$$

$$J_{neu} = N_{alt}$$

$$G_{neu} = G_{alt} + J_{neu}$$

$$W_{neu} = G_{neu} / 2$$

$$K_{neu} = K_{alt} + N_{neu}$$

Die für eine bestimmte Periode ermittelten neuen Werte werden in der nächsten Periode zu den alten Werten. Die Werte der „alten“ Variablen sind also nach der Berechnung der „neuen“ Werte entsprechend zu aktualisieren:

$$N_{alt} = N_{neu}$$

$$J_{alt} = J_{neu}$$

$$G_{alt} = G_{neu}$$

$$W_{alt} = W_{neu}$$

$$K_{alt} = K_{neu}$$

An dieser Stelle lasse ich die Studierenden immer gern schätzen, wie stark sich die Verzögerung der Geschlechtsreife ihrer Meinung nach auswirken wird. Die meisten erwarten etwas in der Größenordnung einer Halbierung der Zahlen nach dem alten Modell. Manche nehmen auch an, dass sich die Zahlen des alten Modells um einen „Zeittakt“ verschieben, dass also der letzte Wert gleich 33.499.906 ist.

Wenn wir nun das neue Modell in ein lauffähiges Computerprogramm umsetzen wollen, könnten wir versuchen, das alte Tabellenkalkulationsblatt so zu modifizieren, dass die Werte nach dem neuen Modell berechnet werden. Aus Gründen, die weiter unten diskutiert werden, wollen wir dies hier nicht tun, sondern ein „klassisches“ Computerprogramm schreiben, welches das neue Modell realisiert. Das Programm wird im folgenden in der Programmiersprache des Computeralgebrasystems Mathematica dargestellt; in den meisten anderen Computeralgebrasystemen ginge es ebenso gut; die Unterschiede wären meist nur syntaktischer Natur.

```
KatzenvermehrungMitGeschlechtsreife[K0_, Rate_, Dauer_]:=
  Module[{t=0,
      Nalt=0, Jalt=0, Galt=K0, Walt=K0/2, Kalt=K0,
      Jneu, Gneu, Wneu, Nneu, Kneu,
      L={{0, K0}} },
    While[t<Dauer, t=t+1;
      Nneu=Rate*Walt;
      Jneu=Nalt;
      Gneu=Galt+Jneu;
      Wneu=Gneu/2;
      Kneu=Kalt+Nneu;
      L=Append[L,{t, Kneu}];
      Nalt=Nneu; Jalt=Jneu; Galt=Gneu; Walt=Wneu; Kalt=Kneu ];
    Return[L]]
```

Das Programm stellt (abgesehen von einigen proprietären syntaktischen Eigenheiten) ersichtlich die direkte Umsetzung des obigen mathematischen Modells dar. Auf die wenigen unumgänglichen Mathematica-spezifischen syntaktischen Besonderheiten, wie die Rolle des Unterstreichungszeichens, soll an dieser Stelle nicht weiter eingegangen werden, da dies in der Einführungsliteratur zu Mathematica (vgl. auch [Wolfram 1996]) meist relativ ausführlich beschrieben ist. Typisch für alle wirklich höheren Programmiersprachen ist der Programmierstil, bei dem zunächst gar nichts gedruckt oder „geplottet“ wird, sondern wo die Ergebnisse zunächst in kompakter Form in einer (strukturierten) Liste gesammelt werden. Diese Liste kann dann später nach Belieben weiterverarbeitet werden (durch formatiertes Drucken, durch Plotten, ...).

Ein Programm-Aufruf

```
KatzenvermehrungMitGeschlechtsreife[2, 2.8, 20];
TableForm[%]
```

liefert nun die Ergebnis-Tabelle

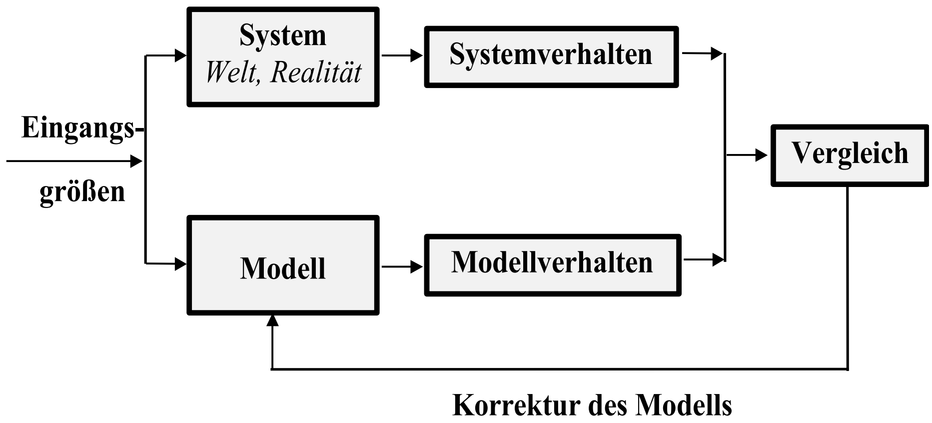
0	2.00
1	4.80
2	7.60
3	14.32
4	24.96
5	45.01
6	79.95
7	142.96
8	254.90
9	455.04
10	811.90
11	1448.96
12	2585.62
13	4614.17
14	8234.03
15	14693.86
16	26221.51
17	46792.92
18	83503.03
19	149013.12
20	265917.36

Der Endwert von 265.917 Katzen (im Vergleich zum vorherigen Wert von 80.399.774 Katzen) wird meist mit ungläubigem Staunen zur Kenntnis genommen. Die geringfügige Modifikation des Modells hat zu einer Reduzierung mit dem Faktor 0.00330743 (also auf nur wenig mehr als 3 Promille) zur Folge. Nur selten wird auf so drastische Weise vor Augen geführt, welche Konsequenzen kleine Modifikationen in der mathematischen Modellierung haben können.

Fehleinschätzungen, wie die in diesem Zusammenhang erlebte, rühren oft von einem massiv verankerten Denken in „linearen“ oder „proportionalen“ Strukturen her, auf die (natürlich zu Recht) im Unterricht auch großer Wert gelegt wird. Viele reale Prozesse sind aber nichtlinearer Natur, und zur Simulation solcher Prozesse ist heute der Einsatz des Computers (mit geeigneter Software) unverzichtbar.

Jeder nichttriviale Modellbildungsprozess ist im allgemeinen ein Regelkreis mit dem Ziel einer beständigen Verbesserung des Modells, d.h. mit einer immer genaueren Übereinstimmung zwischen den realen Daten und den vom Modell gelieferten Ergebnissen (vgl. [Dürr & Ziegenbalg 1984] bzw. [Ziegenbalg 1996]). Schematisch wird dies oft etwa wie in der folgenden Abbildung zum Ausdruck gebracht.

Modellbildung als Regelkreis



Eine größere Realitätsnähe des Modells würde insbesondere die Berücksichtigung der folgenden Aspekte verlangen:

- der Berücksichtigung von Sterbefällen
- der Differenzierung der Geburtenrate entsprechend eines altersschichten-spezifischen Modells (Katzen sind nicht in jedem Alter gleich fruchtbar)
- differenzierte Berücksichtigung von Ressourcen (Nahrung, Raum, Populationsdichte, ...)
- differenzierte Berücksichtigung von „Negativ-Faktoren" aller Art (Umweltschäden, Feinde, Konkurrenten, ...)

Auf den weiteren Ausbau des Modells sei jedoch an dieser Stelle verzichtet.

Algorithmisches Problemlösen, Algorithmierung, Programmierung – eine Nachbetrachtung

Im Laufe der Umsetzung der verschiedenen Modell-Varianten entsteht immer wieder die Frage, welches (Software-) Werkzeug jeweils herangezogen werden soll. Die Software dient dazu, das zunächst in mathematischer Notation aufgeschriebene Modell in ein lauffähiges Computerprogramm umzusetzen.

An dieser Stelle findet in der wissenschaftlichen und pädagogischen Öffentlichkeit gelegentlich eine Diskussion darüber statt, ob man die Lernenden (seien es Schüler oder Studenten) denn wirklich zum „Programmieren" veranlassen sollte, oder ob es nicht wichtiger wäre, sie mit der Nutzung von „Software-Tools" vertraut zu machen.

In derartigen Diskussionen muss ich immer wieder feststellen, dass die Vorstellungen von den Begriffen „Programm“ bzw. „Software-Tool“ extrem divergieren. So wird von vielen der Begriff „Programm“ mit „in einer imperativen Programmiersprache formuliertes Programm“ (z.B. in Pascal) gleichgesetzt. Programmieren etwa mit Lisp, Logo, Prolog oder Computeralgebra Systemen wird dann oft verschämt als „interaktives Arbeiten mit dem Computer“ und dergleichen bezeichnet. Hier tut mehr Klarheit not. Ich zitiere zu diesem Zwecke gern diejenigen, die den Begriff des „software tools“ geprägt haben (vgl. [Kernighan & Plauger 1976]): “*Whatever your application, your most important software tool is a good programming language*".

Der Begriff des Programmierens (vgl. auch [Ziegenbalg 1996]) ist weit allgemeiner als das klassische „Programmieren entsprechend dem imperativen Programmierparadigma“. Jeder Algorithmus, der in einer Sprache formuliert ist, welche die Abarbeitung durch einen Computer ermöglicht, ist ein Programm. In diesem Sinne ist auch die Erstellung eines Tabellenkalkulationsblattes ein Akt des Programmierens – und der Streit um das Programmieren erscheint wie ein Streit um des Kaisers Bart. Entscheidend ist nicht die Frage: „Programmieren: Ja oder nein?“ sondern die Frage „Programmieren: In welcher Programmiersprache, mit welchem Programmierwerkzeug?“

Eine naheliegende Option ist mit Sicherheit in Fällen, wie der obigen Modellbildung, der Einsatz einer Tabellenkalkulations-Software. Einer der großen Vorteile dieser Software liegt in ihrer hochgradigen Interaktivität. Allerdings gibt es auch Nachteile. Zwei der augenfälligsten sind:

- die textliche Dokumentation des „Programms“ ist unbefriedigend

  und damit zusammenhängend:
- problemimmanente Wiederholungsstrukturen (die meist den Kern des betreffenden Algorithmus ausmachen) werden – auf Kosten der Durchsichtigkeit – mit Hilfe des „Kopier“-Vorgangs simuliert.

Oft wird argumentiert, das Kopieren sei ein sehr natürlicher Vorgang. Dabei werden jedoch meist die feinen (aber unverzichtbaren) Unterschiede zwischen relativer und absoluter Adressierung übersehen. Diese Operationen (mit ihren jeweiligen Modalitäten) richtig zu beherrschen ist nicht einfacher, als z.B. der Umgang mit typischen klassischen Wiederholungsstrukturen (etwa vom Typ des „while“-Konstrukts).

Die Konzepte klassischer algorithmischer Formulierungen haben nach wie vor ihren Wert. Sie entsprechen der mathematischen (und der umgangssprachlichen) Formulierung des Modells in außerordentlich „direkter“ Art und Weise (man vergleiche dazu das obige mathematische Modell und seine Umsetzung in das entsprechende Mathematica-Programm). Es sind deshalb kognitiv besonders effiziente Formen der Programmierung (zum Aspekt der *kognitiven Effizienz* siehe z.B. [Ziegenbalg 1996, 5.2]).

Hätte man das Programm in einer Programmiersprache wie Pascal (oder BASIC) geschrieben, so wäre die iterative Fortschreibung der Modellwerte im wesentlichen gleich zu formulieren gewesen; das Ausdrucken (oder Plotten) der Werte hätte in der

Regel aber unmittelbar zum Zeitpunkt ihrer Ermittlung (noch innerhalb der Schleife) stattgefunden.

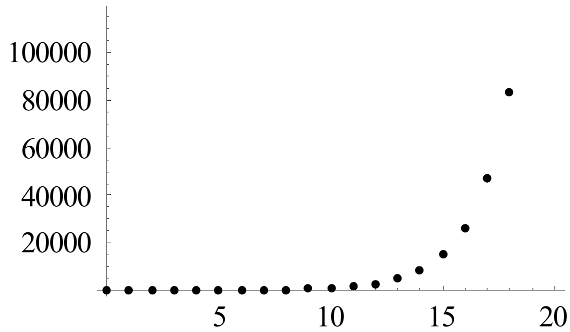
Programmiersysteme, die über Listenverarbeitung im Zusammenhang mit dem funktionalen Programmierparadigma verfügen (also Lisp und seine Dialekte bzw. die moderneren Computeralgebra Systeme) ermöglichen die Trennung der Phasen „Ermittlung der Ergebnisse“ und „Darstellung der Ergebnisse“. Man geht dabei typischerweise folgendermaßen vor: Zunächst wird das Ergebnis ermittelt und in einer gut strukturierten Liste aufbereitet. Danach trifft man die Entscheidung, in welcher Weise man das Ergebnis darstellen möchte: Z.B. ausdrucken, graphisch darstellen („plotten“), in Töne umsetzen, auf eine Diskette schreiben, ...

So hätte die Ergebnisliste des Aufrufs

```
KatzenvermehrungMitGeschlechtsreife[2, 2.8, 20]
```

auch in den ListPlot-Befehl „eingespeist“ werden können – mit dem folgenden Resultat:

```
ListPlot[%]
```



In der Kombination der Eigenschaften „gute Unterstützung des Programmierens“, also:

- Unterstützung unterschiedlicher Programmierparadigmen, insbesondere gute Unterstützung des funktionalen Programmierens (einschließlich effizient implementierter Rekursion),
- volle Listenverarbeitung,
- Symbolverarbeitung,
- exzellente, und nicht zuletzt korrekte (!) Numerik, insbesondere uneingeschränkte Langzahlarithmetik für ganze und Kommazahlen, Bruchrechnung, Rechnen mit komplexen Zahlen,
- exzellente Graphik- und Sound-Unterstützung,
- hervorragende Editiermöglichkeiten (einschließlich Formelsatz)

erscheinen mir die modernen Computeralgebra Systeme unübertroffen in ihrer Eignung als Werkzeuge für die mathematische Modellbildung.

Abschließend sei noch ein Beispiel zur Symbolverarbeitung gegeben. Gibt man an Stelle numerischer Werte die Symbole A (für: Anfangswert der Katzenpopulation) und r (für: Vermehrungsrate) ein, so sind Computeralgebra Systeme in der Regel in der Lage, auch diese Eingaben auszuwerten. Bei der Eingabe des Ausdrucks

Simplify[KatzenvermehrungMitGeschlechtsreife[A, r, 10]]

liefert Mathematica z.B. das Ergebnis

$$\begin{aligned}
&\{\{0, A\},\\
&\{1, \tfrac{1}{2} A (2 + r)\},\\
&\{2, A (1 + r)\},\\
&\{3, \tfrac{1}{4} A (4 + 6 r + r^2)\},\\
&\{4, A (1 + 2 r + \tfrac{3 r^2}{4})\},\\
&\{5, \tfrac{1}{8} A (8 + 20 r + 12 r^2 + r^3)\},\\
&\{6, \tfrac{1}{2} A (2 + 6 r + 5 r^2 + r^3)\},\\
&\{7, \tfrac{1}{16} A (16 + 56 r + 60 r^2 + 20 r^3 + r^4)\},\\
&\{8, \tfrac{1}{16} A (16 + 64 r + 84 r^2 + 40 r^3 + 5 r^4)\},\\
&\{9, \tfrac{1}{32} A (32 + 144 r + 224 r^2 + 140 r^3 + 30 r^4 + r^5)\},\\
&\{10, A (1 + 5 r + 9 r^2 + 7 r^3 + \tfrac{35 r^4}{16} + \tfrac{3 r^5}{16})\}\}
\end{aligned}$$

und man kann versuchen, strukturelle Gesetzmäßigkeiten des Modells zu entdecken.

Literaturhinweise

Dürr R. & Ziegenbalg J. (1984): Dynamische Prozesse und ihre Mathematisierung durch Differenzengleichungen. Paderborn
2. Auflage (1989): Mathematik für Computeranwendungen. Paderborn

Kernighan B.W. & Plauger P.J. (1976): Software Tools. Reading, Massachusetts

Wolfram, St. (1996): The Mathematica Book, Third Edition. Champaign, Illinois

Ziegenbalg, J. (1996): Algorithmen – von Hammurapi bis Gödel. Heidelberg