

18.3 Tabellenkalkulation („spreadsheet“)

Das erste Tabellenkalkulationsprogramm überhaupt war das legendäre Programm **VisiCalc** („sichtbare Berechnungen“), das 1979 von Dan Bricklin (seine Metapher: „eine elektronische Tafel und eine elektronische Kreide im Klassenraum“)¹⁶ erfunden und von Bob Frankston programmiert wurde, zunächst für den berühmten Apple II und bald darauf auch in Adaption für den gleichermaßen berühmten Commodore CBM 8032.¹⁷ Diese beiden Computer waren die wichtigsten „Personal-Computer“ überhaupt kurz vor der Erfindung des IPM-PC im Jahre 1981, womit zugleich das Ende aller ersten „Tischcomputer“ eingeläutet wurde – bis auf den dann 1984 auftretenden Apple Macintosh,¹⁸ der sich neben dem PC als einziger bis heute auf dem Desktop-Markt gehalten hat. Die Software wurde dann an Lotus Development Corporation verkauft und führte zur Entwicklung von **Lotus 1-2-3**, später zu **Quattro Pro** der (nicht mehr existierenden) Softwarefirma Borland. Daneben entwickelte die durch den IBM-PC entstandene Firma Microsoft Anfang der 1980er Jahre das eigene Tabellenkalkulationsprogramm **Multiplan**, das später unter Windows von **Excel**TM abgelöst wurde, welches nunmehr wie Word für WindowsTM in der Windows-Welt einen Quasi-Standard darstellt.

Die „Oberfläche“ eines Tabellenkalkulationsprogramms muss man sich wie ein nach zwei Seiten hin unbegrenztes Schachbrett, ein „ausgebreitetes Blatt“ („spread sheet“, jetzt auch „spreadsheet“), vorstellen, bei dem jede Zelle „Koordinaten“ wie beim Schachbrettfeld hat (z. B. „D6“). Mathematisch gesehen ist es eine Matrix aus m Zeilen und n Spalten, wobei m und n frei (und im Prinzip unbegrenzt, jedoch faktisch begrenzt durch den verfügbaren Speicherplatz oder programminterne Schranken) wählbar sind. Spreadsheets funktionieren neuerdings auch als Funktionenplotter.

¹⁶ <http://dssresources.com/history/sshistory.html>, gültig am 17.07.2002.

¹⁷ Hinweise hierzu z. B. unter <http://www.webopedia.com/TERM/V/VisiCalc.html> und <http://inventors.about.com/library/weekly/aa010199.htm>, jeweils gültig am 17.07.2002. Ich selbst habe VisiCalc 1980 für den legendären Commodore CBM 8032 kennen gelernt.

¹⁸ Vgl. etwa <http://www.apple-history.com/gallery.html>, gültig am 17.07.2002.

An Hand des Programms Excel™ sollen nun elementare Eigenschaften eines Tabellenkalkulationssystems skizziert werden (die im Prinzip auch für andere Produkte gelten). Wir demonstrieren das am Beispiel den Funktionsgraphen von $f(x) = ax^2 + bx + c$.

Abb. 18.11 zeigt ein mögliches Ergebnis: Nur in den Spalten A und B sind einige Zellen belegt. (Hinweis: Statt „Zelle“ heißt es bei manchen Spreadsheets auch „Feld“; jedoch hat „Feld“ bei Excel eine andere Bedeutung.) Wir erkennen unschwer die Formvariablen a , b und c (Zellen A3, A4, A5) und deren aktuelle Belegungen bzw. Werte (Zellen B3, B4, B5). Auch die Wertetabelle (die wir „klassisch“ von Hand erstellen würden) erkennen wir in den Zeilen 9 bis 18. Und in der Zeile 7 stehen die „Spaltenköpfe“ dieser Wertetabelle, wie wir es gewohnt sind. Sogar der konkrete Term der Funktion ist hier eingetragen, wenn auch in einer nicht mathematiküblichen, jedoch programmierüblichen Notation. Daneben ist der Funktionsgraph mit den „Stützpunkten“ zu sehen.

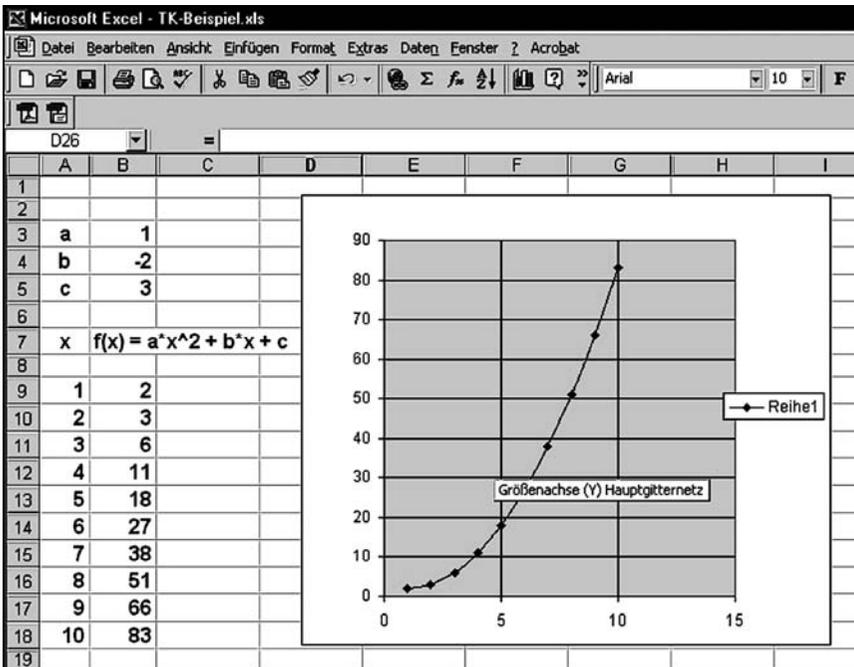


Abb. 18.11: Tabellenkalkulation mit rekursiver Programmierung und Datendiagramm als Funktionenplotter

Mit dem Mauszeiger kann man die Schreibmarke in irgendeiner Zelle platzieren und dann dort etwas „schreiben“. Aber: Was wir auf Anhieb weder sehen, noch wissen können: Die Zellen können (ähnlich wie bei einer elektronischen Datenbank) von unterschiedlichem *Typ* sein: *Zahlen*, *Texte*, *Formeln*. Der in Abb. 18.11 gezeigten Tabellenansicht können wir den Typ der einzelnen Zelle *so* jedoch noch nicht entnehmen. Lösung: Wir aktivieren unter „Extras | Optionen | Ansicht“ das Kontrollkästchen „Formeln“ und erhalten Abb. 18.12.

Zellen, die Formeln enthalten, beginnen mit einem „=“, gefolgt von einem Term, der neben Rechenzeichen und Konstanten auch Variable enthalten kann, die *Zeiger* auf eine (andere) Zelle sind, z. B. A9 in Zelle A10. Mit diesem Wissen erkennen wir, dass die Inhalte in den Zellen A3, A4, A5 und A7 offensichtlich vom Typ „Text“ sind. Und entgegen der ersten Vermutung gemäß Abb. 18.11 steht auch in Zelle B7 keine Formel, sondern „nur“ ein Text (der damit wie der andere Text nur erläuternde, jedoch keine rechnerische Bedeutung hat!) Die Inhalte der Zellen B3, B4, B5 und A9 sind offenbar vom Typ „Zahl“, jedoch könnten sie auch vom Typ „Text“ sein, was wohl zu unterscheiden ist: So gilt $2 < 12$, wenn „2“ und „12“ vom Typ „Zahl“ sind, hingegen $12 < 2$, wenn sie vom Typ „Text“ sind („lexikographische Ordnung“).

	A	B
3	a	1
4	b	-2
5	c	3
6		
7	x	$f(x) = a \cdot x^2 + b \cdot x + c$
8		
9	1	$=B\$3 \cdot A9^2 + B\$4 \cdot A9 + B\$5$
10	$=A9+1$	$=B\$3 \cdot A10^2 + B\$4 \cdot A10 + B\$5$
11	$=A10+1$	$=B\$3 \cdot A11^2 + B\$4 \cdot A11 + B\$5$
12	$=A11+1$	$=B\$3 \cdot A12^2 + B\$4 \cdot A12 + B\$5$
13	$=A12+1$	$=B\$3 \cdot A13^2 + B\$4 \cdot A13 + B\$5$

Abb. 18.12: Ausschnitt der Tabelle aus Abb. 18.11 in „Formelansicht“

In Zelle A10 steht $=A9+1$. Dies wurde genau so in das Eingabefeld eingetragen. Wenn man nun den Inhalt dieser Zelle mittels „Kopieren“ und „Einfügen“ (copy&paste) in die Zelle A11 überträgt, erscheint dort $=A10+1$. Der Zellenbezug wurde also automatisch geändert, und daher nennt man das eine *relative Adressierung*.

In Zelle B9 steht $=\$B\$3*A9^2+\$B\$4*A9+\$B\5 . Wir eliminieren zunächst die merkwürdigen Dollarzeichen: $=B3*A9^2+B4*A9+B5$. So verstehen wir es und erkennen unmittelbar, dass hier der Funktionsterm mit Bezug auf die Zellen, in denen die Werte der Formvariablen stehen, programmiert wurde. Würden wir nun diesen Zelleninhalt analog mittels copy&paste in Zelle B10 übertragen so würde sich $=B4*A10^2+B5*A10+B6$ ergeben. Damit würde zwar korrekt Bezug auf den aktuellen Wert von x genommen, jedoch wäre ein unsinniger Bezug zu den Werten der Formvariablen entstanden. Da die Werte letzterer – im Gegensatz zu denen von x – an festen Stellen stehen, muss man hier anstelle der relativen eine *absolute Adressierung* wählen, und das geschieht mit den Dollarzeichen vor der Spalten- und der Zeilenbezeichnung (wobei man auch nur eine von beiden absolut wählen kann, was hier aber sinnlos wäre).

Nachdem auf diese Weise die Zellen A10 und B10 programmiert wurden, kann man einfach beide gemeinsam kopieren und dies auf einmal in alle gewünschten darunter liegenden Zellen kopieren. Die Programmierung der Werteballe ist fertig! („darunter“ ist übrigens nicht nötig, die Zielzellen können *irgendwo* liegen!)

Im letzten Schritt markiert man mit der Maus denjenigen Teil der Tabelle (in Normalansicht), den man graphisch darstellen möchte, wählt im Menü „Einfügen | Diagramm | Punkt (XY)“ und dort den gewünschten Darstellungstyp aus, dann „Fertigstellen“, und schon erhält man den Rest von Abb. 18.11. Veränderungen der Werte der Formvariablen wirken sich sofort auf den Funktionsgraphen aus.

Nun mag man einwenden, dass dies nicht korrekt sei, weil doch nur *wenige* Punkte berechnet und diese dann von dem Programm „selbstständig“ verbunden wurden. Richtig! Aber so hat man es früher mit einem *Kurvenlineal* auch gemacht (wer kennt das noch? – siehe hierzu mein eigenes, altes aus der Oberstufenschulzeit in Abb. 18.13). Und außerdem: Was ist denn hier schon „richtig“? Wir vertiefen das noch in den Kapiteln 19 und 20.



Abb. 18.13: Scan meines Kurvlineals von 1960