Simulation von Aktienkursen mit vielen Aktien im Levy-Levy-Solomon Modell.

Schriftliche Hausarbeit im Rahmen der Ersten Staatsprüfung für das Lehramt für die Sekundarstufe II mit Zusatzprüfung für die Sekundarstufe I, dem Staatlichen Prüfungsamt für Erste Staatsprüfungen für Lehrämter an Schulen in Köln vorgelegt von:

Reiner Kohl

Köln, den 23. September 1997

Gutachter: Prof. Dr. D. Stauffer

Institut für Theoretische Physik, Universität zu Köln

Inhaltsverzeichnis

1 Einleitung **3**

2 Das Modell **5**
2.1 Grundannahmen 5
2.1.1 Anlageformen 5
2.1.2 Aktionärsverhalten 6
2.1.3 Die Utilityfunktion 6
2.2 Berechnung der optimalen Verteilung 7
2.2.1 Hyperebenenspiegelung I 8
2.2.2 Hyperebenenspiegelung II 8
2.2.3 Bestimmung des Maximums 10
2.3 Abweichung vom Optimum 11
2.4 Berechnung der neuen Aktienpreise 11
2.5 Aktualisierung der Daten 13

3 Das verwendete Programm **14**
3.1 Programmcode 14
3.1.1 Wichtige Variabeln und Konstanten 23
3.2 Der Algorithmus 25
3.2.1 Zufallszahlen 25
3.2.2 Das Kernprogramm 28

4 Ergebnisse **33**
4.1 Das Einaktienmodell 33
4.2 Geschwindigkeitsbetrachtung 35
4.3 Mehrere Aktien 38
4.3.1 Eine Gedächtnislänge 38
4.3.2 Zwei verschiedene Gedächtnisgruppen 43
4.3.3 3 Gedächtnisgruppen 43
4.3.4 Sechs Gedächtnisgruppen 50
4.3.5 Kapitalkonzentration 55
4.3.6 Änderung der Dividenden 57
4.4 Vermögensstruktur 58

5 Änderungen des Grundprogramms **60**
5.1 Änderung der Ultilityfunktion 60
5.1.1 Änderungen am Programm 61
5.1.2 Ergebnisse 63

5.1.3 Kapitalkonzentration 64
5.1.4 Verteilung 67

6 Zusammenfassende Schlußbemerkung **71**

A Anmerkungen **74**

B Literatur- und Abbildungsverzeichnis **75**

Kapitel 1

Einleitung

Bei Computersimulationen macht man sich zu Nutze, daß Computer sehr schnell rechnen können. Daher kann man mit Hilfe eines Computers vielschichtige, komplexe Vorgänge „ausprobieren“. Gerade in der statistischen Physik, gibt es viele Phänomene, die über Wahrscheinlichkeiten beschrieben werden. Die Anzahl der Moleküle in einem Glas Kölsch (selbst wenn es alkoholfrei ist) ist so groß, daß man den Vorgang der Bewegung der Moleküle nicht analytisch berechnen kann. In der statistischen Physik und bei den resultierenden Computersimulationen betrachtet man die Bewegungen als mehr oder weniger zufällig.

Interessant ist, daß man die so gewonnenen Erfahrungen auch auf andere Wissenschaftsgebiete übertragen kann. So gibt es Simulationen im Bereich der Biologie, Medizin, Verkehrsplanung und noch vielen anderen Bereichen. Es scheint ungewöhnlich, daß Physiker in diesen Gebieten forschen: ich glaube, daß Physiker viel Erfahrung mit der Erstellung von Modellen haben und so andere Wissenschaftsbereiche von einem, für diese neuen Blickwinkel, betrachten können.

Es ist mir leider nicht gelungen den Kursverlauf einer oder mehrerer realer Aktien vorherzusagen. Wäre mir dies gelungen, hätte ich wahrscheinlich weder Zeit, noch Interesse mein Wissen in Form einer Staatsexamensarbeit niederzuschreiben...

Unser Wirtschaftssystem und somit auch der Aktienkurs hängen von zu vielen Variablen ab, als das es möglich wäre, einen Aktienkurs über eine lange Zeit richtig vorauszusagen. Wie jedoch in der Physik üblich, versucht niemand die Welt im Ganzen zu verstehen, sondern man untersucht kleine, isolierte Teilgebiete, immer bedacht, daraus gewonnene Ergebnisse mit denen aus anderen Teilgebieten zu verknüpfen.

Grundsätzlich könnte man versuchen, die Schwankungen des Börsenkurses mit Hilfe eines einfachen *random walk* zu bestimmen. Dies führt jedoch nicht zu den gewünschten Ergebnissen.

Das Modell von Levy, Levy und Solomon [1] ,auf dem meine Arbeit basiert, simuliert den Kursverlauf *einer* Aktie in einem *isolierten* Markt mit einer festen Anzahl von Investoren. Die Simulation bestimmt verschiedene Werte in diskreten Schritten, wobei jeder Simulationsschritt einem Börsentag entspricht.

In diesem Modell kann man die simulierte Aktie als Stellvertreter für eine große Anzahl von Aktien betrachten. Ihr Kurs soll somit in etwa dem eines Aktienin-

[1] M. LEVY, H. LEVY & S. SOLOMON (1994) und (1995). Das LLS-Modell wurde von T. HELLTHALER (1995) weitergehend untersucht.

dex (z. B. Dow Jones Index, DAX) entsprechen und nicht unbedingt dem einer einzelnen Aktie.
Das Modell führt, wie weiter unten beschrieben, bei vielen Aktionären zu unrealistischen Ergebnissen. Die Aufgabe meiner Arbeit ist es das Modell auf viele Aktien zu erweitern und zu untersuchen, welche Kurse man bei vielen Aktionären erhält.

Kapitel 2

Das Modell

2.1 Grundannahmen

2.1.1 Anlageformen

Die Aktionäre können ihr Geld auf zwei verschiedene Arten anlegen:

- Bei **Renten** handelt es sich um eine festverzinsliche Anlageform. Die Aktionäre dürfen beliebig viel Geld in Renten investieren. Es wird täglich ein fester Zinssatz $\mathbf{r}$ gezahlt. Im Gegensatz zu Banken tätigt der Computer *Überweisungen* [1] sofort.

- Bei **Aktien** handelt es sich um eine spekulative Anlageform. Die *Rendite* der Aktie wird durch zwei Faktoren bestimmt:

  - **Dividenden** werden täglich ausgezahlt. Es wird eine Anfangsdividende D_{i0} festgelegt. Diese wird für jede Aktie i täglich um einen festen Faktor d_i erhöht.[2] Die am Tag t für die Aktie i durch die Dividende $D_{t,i}$ erwirtschaftete Rendite beträgt $\mathbf{r_{D,t,i}} = \frac{\mathbf{D_{t,i}}}{\mathbf{P_{(t-1),i}}}$, wobei $P_{(t-1),i}$ der Kurs der Aktie i am Vortag ist. [3]
  - **Kursschwankungen** können zusätzliche Gewinne, aber auch Verluste bewirken. Die Kursschwankung der Aktie i erbringt die (unter Umständen negative) Rendite $\mathbf{r_{K,t,i}} = \frac{\mathbf{P_{t,i} - P_{(t-1),i}}}{\mathbf{P_{(t-1),i}}}$.

  Für die **Gesamtrendite** H_t der Aktie i am Tag t erhält man:

$$\mathbf{H_{t,i}} = \mathbf{r_{K,t,i}} + \mathbf{r_{D,t,i}} = \frac{\mathbf{P_{t,i} - P_{(t-1),i} + D_{t,i}}}{\mathbf{P_{(t-1),i}}}. \tag{2.1}$$

Die Anzahl der Investoren (N_I) und die Menge der Aktien einer Firma ($N_{A,i}$) bleiben über eine gesamte Simulation konstant.

[1] Es werden im Modell keine Gebühren für Transaktionen erhoben. Die Vermögenssteuer habe ich in Anbetracht meines Nachnamens ebenfalls nicht realisiert.

[2] Hier und im Folgenden bezeichnet der Index i immer die „Nummer“ der Aktie

[3] Die *Dividende* ist nach diesem Modell unabhängig vom Kurs der Aktien oder dem Zustand der zugehörigen Firma.

2.1.2 Aktionärsverhalten

Jeder Aktionär muß zu jedem Zeitpunkt einer Simulation sein gesamtes Geld entweder in Aktien und/oder in Renten anlegen. Das Geld, welches er nicht in Aktien investiert, wird in Renten angelegt. Dies ist sinnvoll, da er sonst Zinsen verschenken würde. Zu Beginn eines Börsentages muß er sich entscheiden, welchen Anteil seines Geldes er in (welche) Aktien investieren möchte. Nach welchen Kriterien er sich entscheidet, ist der Kern der Simulation und wird im nächsten Abschnitt beschrieben.

2.1.3 Die Utilityfunktion

Es liegt bekanntlich in der *Natur* jedes Aktionärs sein Kapital zu vergrößern zu wollen. Das Instrument, das dem Investor zeigt, welchen *Anteil* seines Geldes er in welche Aktie und/oder in Rentenpapieren anlegen sollte, ist die **Utilityfunktion**. Die Utilityfunktion des vorliegenden Modells ist eine Funktion von $\mathbb{R}^{AnAk}$ nach $\mathbb{R}$. Bei einem Vektor $\vec{x} \in \mathbb{R}^{AnAk} = (x_1, \ldots, x_{AnAk})$ [4] steht x_i für den Anteil des Kapitals, den der Aktionär in Aktie i anlegen möchte. Das Maximum der Utilityfunktion liegt bei dem optimalen Wert von $\vec{x}$.
Da ein fester Geldbetrag für einen *armen* Investor relativ mehr bedeutet, als für einen *reichen*, betrachtet man als zu maximierende Funktion nicht das Kapital, sondern dessen Logarithmus. [5]
In der Simulation teilt man die Aktionäre in Gruppen mit verschiedenen „Gedächtnislängen" ein: die Aktionäre in den verschiedenen Gruppen können sich unterschiedlich lang an die Aktienkurse der vergangenen Tage erinnern. Man spricht unter diesen Voraussetzungen von einem **heterogenen** Markt. Die Aktionäre der einzelnen Gruppen betrachten den Kurs der Aktien der letzten k_j Tage. [6] Die Idee der Simulation besteht darin, daß jeder Aktionär davon ausgeht, daß die Rendite, welche eine Aktie in den Vortagen erbracht hat mit der Wahrscheinlichkeit $\frac{1}{k_j}$ am folgenden Tag $t+1$ wieder eintritt. Er wendet also die oben errechnete Gesamtrendite der letzten k_j Tage auf das Kapital am Tag $t+1$ an. Die Utilityfunktion (EU) [7] vor dem Handel am Tag $t+2$ bzw. am Abend des Tages $t+1$, also nach Einnahme der Zinsen und Dividenden, die durch das am Tag $t+1$ investierte Kapital erwirtschaftet wurden, ist gegeben durch:

$$EU_j(x_1, \ldots, x_{AnAk}) = \frac{1}{k_j} \sum_{s=t}^{t-k_j+1} \ln \left\{ \underbrace{(1 - \sum_{i=1}^{AnAk} x_{j,i}) K_j^*}_{\text{in Renten angelegtes Kapital}} \cdot (1+r) + \right.$$

$$\left. \sum_{i=1}^{AnAk} \underbrace{x_{j,i} K_j^*}_{\text{in Aktie i angelegtes Kapital}} \cdot (1 + H_{s,i})) \right\}. \tag{2.2}$$

[4] $AnAk$ bezeichnet die Anzahl der in einer Simulation betrachteten verschiedenen Aktien.
[5] Diese Annahme habe ich später auf Vorschlag von Diedier Sornette geändert.
[6] j steht für die Gruppe Nr.:j.
[7] Die folgenden Formel beziehen sich auf $AnAk$ verschiedene Aktien. Die explizite Herleitung für eine Aktie ist bei Hellthaler zu finden.

Dabei bezeichnet K_j^* das Kapital am Tag $t+1$. Diese Gleichung läßt sich umformen zu:

$$EU_j(x_1,\ldots,x_{AnAk}) = \ln K_j^* + \frac{1}{k_j}\sum_{s=t}^{t-k_j+1} \ln\left\{\sum_{i=1}^{AnAk}(x_{j,i}(H_{s,i}-r)) + (1+r)\right\}$$

Somit reduziert sich das Problem der Maximierung mit:

$$EU_j(x_1,\ldots,x_{AnAk}) = F_j(x_1,\ldots,x_{AnAk}) * \frac{1}{k_j} + \ln K_j^*$$

auf die Funktion $\mathbf{F_j}(\ldots)$ [8]:

$$F_j(x_1,\ldots,x_{AnAk}) = \sum_{s=t}^{t-k_j+1} \ln\left\{(1+r) + \sum_{i=1}^{AnAk} x_{j,i}(H_{s,i}-r)\right\} \tag{2.3}$$

$F_j(\ldots)$ ist von K_j^* unabhängig. Es handelt sich bei dieser Funktion um eine Funktion von $AnAk$ Dimensionen in *eine* (siehe oben). Die Lage des Maximums der Funktion liefert somit für jede Aktie den optimalen Anteil x_i den der Aktionär in die Aktie i investieren sollte.
Die Utilityfunktion und somit $F_j(\ldots)$ unterliegt folgenden Nebenbedingungen:

1. $\sum_{i=1}^{AnAk} x_i \leq 1$
2. $x_i \geq 0$ Die Anteile x_i liegen natürlich zwischen 0 und 1.

Der erste Punkt muß gelten, da der Aktionär in der Simulation keinen Kredit aufnehmen darf. Nachdem alle Aktionäre ihre Kaufentscheidungen getroffen haben, kann der Aktienpreis auf Grund von Angebot und Nachfrage bestimmt werden. Es ist wichtig, daß alle Aktien im Umlauf bleiben. Über die Beziehung $x_R + \sum_{i=1}^{AnAk} x_i = 1$ wird der Anteil des Kapitals x_R bestimmt, den der Aktionär in Rentenpapieren anlegt.

2.2 Berechnung der optimalen Verteilung

Um den optimalen Anteil des in Aktien zu investierenden Kapitals eines Investors zu bestimmen, wird die Lage des Maximums der Funktion $F_j(\ldots)$ bestimmt. $F_j(\ldots)$ ist nicht linear, daher können Verfahren zur linearen Optimierung, wie z. B. das Simplexverfahren [9] nicht verwendet werden.
Bei der Programmierung muß beachtet werden, daß das Maximum auf dem Rand des zulässigen Bereichs liegen kann.
Das Maximum ist (bei gegebenen $H_{t,i}$) nur noch von der Gedächtnislänge k_j abhängig, wird also nicht für jeden einzelnen Aktionär, sondern nur für die verschiedenen Gruppen bestimmt. Die Idee, das Maximum der Funktion

$$F_j(x_1,\ldots,x_{AnAk}) = \sum_{s=t}^{t-k_j+1} \ln\left\{(1+r) + \sum_{i=1}^{AnAk} x_{j,i}\cdot(H_{s,i}-r)\right\}$$

numerisch zu bestimmen, führt über den Gradienten.

[8] j bezeichne die Nummer des Aktionärs.
[9] Ch. Großmann, J.Terno (1993), Kapitel 4.2

Man kann sich vorstellen, daß ein Wanderer den Gipfel eines Berges erklimmt, indem er immer so steil wie möglich *nach oben* läuft. Als Ränder des zulässigen Bereiches kann man sich einen Zaun oder eine Felswand vorstellen. [10] Da der Gradient eine lokale Größe ist, läuft der Wanderer von einem bestimmten Startpunkt aus ein Stück in Richtung des Gradienten. An diesem Punkt sucht er erneut die Richtung des stärksten Anstieges und arbeitet sich so von Iterationspunkt zu Iterationspunkt zum Gipfel vor.
Die Ableitung von $F_j(\dots)$ nach der l'ten Koordinatenrichtung lautet wie folgt:

$$\frac{\partial F_j(x_1,\dots,x_{AnAk})}{\partial x_l} = \sum_{s=1}^{t-k_j+1} \frac{1}{x_{j,l} + \sum_{\substack{i=1\\ i\neq l}}^{AnAk} x_{j,i}\frac{H_{s,i}-r}{H_{s,l}-r} + \frac{1+r}{H_{s,l}-r}}. \quad (2.4)$$

Der Gradient einer Funktion $f(x_1,\dots,x_n) \to y$ mit $(x_1,\dots,x_n, y \in I\!R)$ lautet:

$$\overrightarrow{\text{grad}} = (\frac{\partial f(\dots)}{\partial x_1},\dots,\frac{\partial f(\dots)}{\partial x_n}).$$

Es ist problematisch, wenn ein Iterationspunkt am Rand liegt, und der Gradient aus dem erlaubten Bereich hinaus zeigt. Dieses Problem habe ich durch Hyperebenenspiegelungen [11] gelöst.

2.2.1 Hyperebenenspiegelung I

Bei den Rändern mit $x_i = 0$ handelt es sich um Hyperebenen. Tritt der Fall ein, daß für einen Iterationspunkt $\vec{x}^u$ [12] eine Komponente $\vec{x}_i^u$ Null ist und $[grad(F(\dots))]_i$ negativ ist, so wird $[such(F(\dots))]_i = 0$. Dies ist eine Hyperebenenspiegelung auf die Koordinatenachse.
Aus den eventuellen Spiegelungen resultiert der Suchvektor $\overrightarrow{such}(\dots)$. Im Folgenden wird in die Richtung Suchvektor „gelaufen", falls also keine Spiegelung notwendig war, ist dieser der Gradient.

2.2.2 Hyperebenenspiegelung II

In dem Beispiel in Abbildung 2.1 liegt der Iterationspunkt $\vec{x}^3$ auf dem Rand, da $\vec{x}_1^3 + \vec{x}_2^3 = b$ [13] ist. In diesem Punkt zeigt der Gradient (als Pfeil angedeutet) aus dem zulässigen Bereich hinaus. Damit die Suche im zulässigen Bereich bleibt, muß die Suchrichtung geändert werden. Ich führe zu diesem Zweck eine Hyperebenenspiegelung durch. Man fällt das Lot auf die Hyperebene und erhält so die neue Suchrichtung. Die Hyperebene ist bestimmt durch die Gleichung: $\sum_{i=1}^{AnAk} x_i = b$. Der zugehörige Normalenvektor ist $\vec{n} = (1,\dots,1)$. [14]

[10] Dem normalen Wanderer stehen jedoch nur drei Raumdimensionen zur Verfügung. Über Erschließung weiterer Dimensionen konnte mir selbst der Eifelverein keine weiteren Informationen erteilen.
[11] Bei einem n-dimensionalen Raum ist die Hyperebene ein $(n-1)$-dimensionaler Teilraum.
[12] $\vec{x}^u$ bezeichnet den u'ten Iterationspunkt.
[13] Anstatt $b = 1$, wählt man b etwas kleiner als 1, damit den Aktionäre eine „Sicherheitsreserve" bleibt.
[14] Auf die Darstellung von Vektoren in Spaltenschreibweise habe ich verzichtet. Auch habe ich die Vektoren in Zeilenschreibweise nicht als transponiert gekennzeichnet.

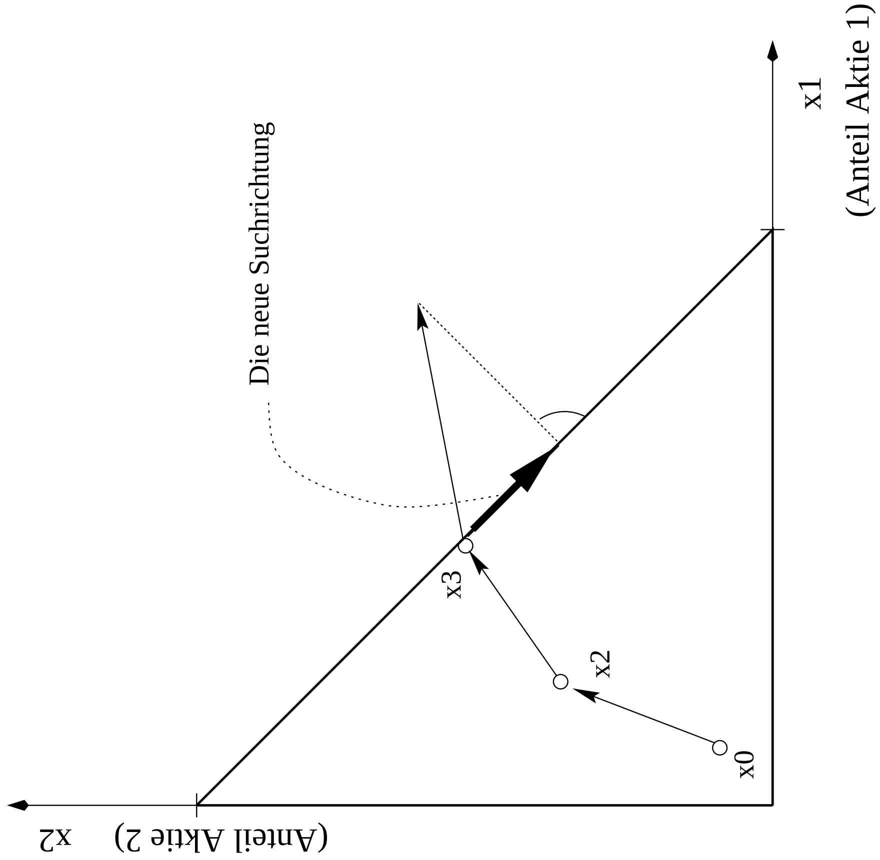

Abbildung 2.1: Hyperebenenspiegelung in zwei Dimensionen

Man zieht vom Gradienten ein Vielfaches des Normalenvektors ab, so daß für den zu bestimmenden neuen Suchvektor $\overrightarrow{such}$ gilt: $\sum_{i=1}^{AnAk} \overrightarrow{such}_i = 0$.

2.2.3 Bestimmung des Maximums

Für jede Gedächtnislängengruppe j muß das Maximum von $F_j(\dots)$ getrennt berechnet werden.
Ich beginne die Suche nach dem Maximum an einem beliebigen Startpunkt $\vec{x}^0$. Nun berechne ich $\overrightarrow{grad}(F_j(\vec{x}^0))$ und hieraus den oben beschriebenen Suchvektor $\overrightarrow{such}(F_j(\vec{x}^0))$. Wie weit in Richtung des Suchvektors zum neuen Iterationspunkt gegangen wird, ist durch α_{opt} festgelegt: der neue Iterationspunkt ist $\vec{x}^1 = \vec{x}^0 + \alpha_{opt} \cdot \overrightarrow{such}(F_j(\vec{x}^0))$. Nun möchte ich erläutern, wie man das α_{opt} erhält: ich berechne zuerst α_0, so daß $\vec{x}_{Rand} = \vec{x}^0 + \alpha_0 * \overrightarrow{such}(F_j(\dots))$ eine Koordinatenachse trifft
Anschließend berechne ich α_1, so daß $\vec{x}_{Rand} = \vec{x}^0 + \alpha_1 * \overrightarrow{such}(F(\dots))$ auf den Rand zeigt, der durch $\sum_{i=1}^{AnAk} x_{Rand_i} = b$ bestimmt ist.
Damit $\vec{x}^1$ den zwei Randbedingungen genügt, muß $\alpha_{opt} \in [0, \alpha_{max}]$ mit $\alpha_{max} = min\{\alpha_0, \alpha_1\}$ liegen. Das Problem, das Maximum einer Funktion von $\mathbb{R}^{AnAk} \to \mathbb{R}$ zu finden, reduziert sich auf die Bestimmung des Maximums der Funktion: $\tilde{F}_j(\alpha) : \mathbb{R} \to \mathbb{R}$

$$
\begin{aligned}
\tilde{F}_j(\alpha) &= F_j\left(\vec{x}_j^0 + \alpha \cdot \overrightarrow{such}(F_j(\vec{x}_j^0))\right) \qquad (2.5) \\
&= \sum_{s=1}^{t-k_j} \ln\left\{(1+r) + \sum_{i=1}^{AnAk} \left[\vec{x}_j^0 + \alpha \cdot \overrightarrow{such}(F_j(\vec{x}_j^0))\right]_i \cdot (H_{s,i} - r)\right\} \\
&= \sum_{s=1}^{t-k(j)} \ln\left\{ (1+r) + \underbrace{\sum_{i=1}^{AnAk} \vec{x}_{j,i}^0 \cdot (H_{s,i} - r)}_{\mathcal{A}_S} + \right. \\
&\qquad \left. \underbrace{\alpha * \sum_{i=1}^{AnAk} \left[\overrightarrow{such}(F_j(\vec{x}_j^0))\right]_i \cdot (H_{s,i} - r)}_{\mathcal{B}_S} \right\}.
\end{aligned}
$$

Von $\tilde{F}_j(\dots)$ kann man mit Hilfe einer Intervallschachtelung leicht das Maximum bestimmen:

$$
\frac{d\tilde{F}_j(\alpha)}{d\alpha} = \sum_{s=1}^{t-k_j} \frac{1}{\alpha + \frac{(1+r)+\mathcal{A}_S}{\mathcal{B}_S}} \qquad (2.6)
$$

Aus der zweiten Ableitung

$$
\frac{d^2\tilde{F}_j}{d\alpha^2} = \sum_{s=1}^{t-k_j} -\frac{1}{\left(\alpha + \frac{(1+r)+\mathcal{A}_S}{\mathcal{B}_S}\right)^2} < 0.
$$

folgt, daß die Nullstelle von $\frac{d\tilde{F}_j(\alpha)}{d\alpha}$ ein Maximum ist.

2.3 Abweichung vom Optimum

Da auch Aktionäre Menschen sind, handeln sie nicht nur nach rationalen Gesichtspunkten. Würden alle Investoren die gleichen Informationen erhalten und diese gleich verarbeiten, so entstände kein Handel. Das Verhalten der Investoren ist neben rationalen Gesichtspunkten stark durch Spekulation beeinflußt. Hinzu kommen emotionale Gesichtspunkte. Die Entscheidung an einem Tag eher risikofreudig oder ängstlich zu reagieren, hängt von vielen Gesichtspunkten ab: zum Beispiel von dem Bein, mit dem der Investor am Morgen des Handelstages zuerst aufgestanden ist, oder von der Anzahl der Striche auf dem Bierdeckel der Kneipe, die der Investor am Vorabend besucht hat. Und selbst wenn man über die Ereignisse, die das Gefühlsleben des Investors beinflussen informiert wäre, könnte man über die Auswirkungen auf sein Verhalten an der Börse keine Aussage treffen.
Es handelt sich um rein zufällige Ereignisse. Um diese emotionale Seite mit in die Simulation einzubeziehen, wird der rational bestimmte Aktienanteil $\vec{x}_j$ für jeden Aktionär einzeln, komponentenweise durch einen Zufallswert verändert.
Hierfür wählt man einen Zufallswert aus einer normalverteilten Zufallsverteilung, da kleine Abweichungen wahrscheinlicher sind, als große. Die symmetrischen Eigenschaften der Normalverteilung beschreiben die Realität gut, da Abweichungen nach oben oder unten gleich wahrscheinlich sein sollten.
Man muß sich bei der Programmierung noch einige Gedanken über die Zufallszahlen machen, zum Beispiel, daß durch die Zufallszahlen die Nebenbedingungen nicht verletzt werden. Das erläutere ich bei der Beschreibung des Programms. Wichtig ist, daß die Abweichung vom Optimum nur durch Zufallszahlen bestimmt wird.

2.4 Berechnung der neuen Aktienpreise

Wenn für jeden Aktionär der optimale Aktienanteil bestimmt worden ist, müssen die neuen Aktienpreise festgelegt werden. Da es eine feste Anzahl an Aktien auf dem Markt gibt, richtet sich der Preis nach Angebot und Nachfrage. Ich möchte an dieser Stelle betonen, daß es sich bei der vorgestellten Simulation um einen *„schwachen“* Markt handelt, was bedeutet, daß der neue Preis der Aktien einmal am Tag bestimmt wird. Das entspricht einer zentralen Stelle, die alle Angebote und Nachfragen aufnimmt und daraus den neuen Aktienpreis bestimmt. Man könnte die Angebote auch sequentiell abarbeiten, so daß nach dem Kauf und Verkauf von Aktien zum Beispiel eines Paars von Aktionären, die anderen Aktionäre ihre Angebote neu überdenken könnten.
In dem vorgestellten Modell kann es vorkommen, daß ein *armer* Investor, der die Aktie i kaufen möchte, durch den einmal am Abend festgesetzten, neuen, eventuell hohen Preis der Aktie, sich diese nicht mehr leisten kann. Die Zeit, die eine Simulation für einen Markt bräuchte, der die Investoren sequentiell abarbeitet, ist deutlich höher, als die in dem vorgestellten Modell. Bei vielen Aktien sind die Rechenzeiten in dem von mir untersuchten Modell sehr lang (vgl. Abbildung 4.5), so daß ich nur einen schwachen Markt untersucht habe.

Am Ende eines Börsentages erhöht sich das Kapital der Aktionäre K_j durch Einnahme von Zinsen und Dividendenausschüttungen auf $\tilde{K}_j$:

$$\tilde{K}_j = \underbrace{K_j}_{\text{altes Kapital}} + \underbrace{\sum_{i=1}^{AnAk} N_{j,i} D_i}_{\text{Dividenden}} + \underbrace{\left(K_j - \sum_{i=1}^{AnAk} N_{j,i} P_i \right) r}_{\text{Zinsen aus Renten}} .$$

$N_{j,i}$ steht für die Anzahl der Aktien, die der Aktionär j [15] von der Aktie i besitzt. Der Ausdruck läßt sich wie folgt umformen:

$$\tilde{K}_j = K_j(1+r) + \sum_{i=1}^{AnAk} N_{j,i}(D_i - P_i \cdot r). \tag{2.7}$$

Durch die noch nicht berechneten Kursänderungen ergibt sich ein neues Kapital K_j^* :

$$K_j^* = \tilde{K}_j + \sum_{i=1}^{AnAk} N_{j,i}(P_i^* - P_i), \tag{2.8}$$

, wobei P_i^* für den noch auszurechnenden neuen Preis der Aktie i steht.
In der folgenden Formel bezeichnet $N_{j,i}^*$ die Anzahl der Aktien der Firma i,die der Aktionär j nach Abschluß des Handels besitzen wird.

$$N_{j,i}^* = \frac{x_{j,i} K_j^*}{P_i^*} \tag{2.9}$$

Unter N_{A_i} bezeichne ich die Anzahl der insgesamt auf dem Markt vorhandenen Aktien der Firma i. Es gilt:

[15] j bezeichnet hier und im Folgenden den einzelnen Investor.

$$
\begin{aligned}
N_{A_i} &= \sum_{j=1}^{N_I} N^*_{j,i} \\
&= \sum_{j=1}^{N_I} \frac{x_{j,i}K^*_j}{P^*_i} \\
&= \sum_{j=1}^{N_I} \frac{x_{j,i}}{P^*_i}\left(\tilde{K}_j + \sum_{l=1}^{AnAk} N_{j,l}(P^*_l - P_l)\right) \\
&= \underbrace{\sum_{j=1}^{N_I} x_{j,i}\cdot N_{j,i}}_{=:R_i} + \frac{1}{P^*_i}\underbrace{\sum_{j=1}^{N_I} x_{j,i}\tilde{K}_j}_{=:S_i} \\
&\quad -\frac{1}{P^*_i}\underbrace{\sum_{j=1}^{N_I} x_{j,i} \sum_{l=1}^{AnAK} N_{j,l}P_l}_{=:T_i} + \frac{1}{P^*_i}\underbrace{\sum_{j=1}^{N_I} x_{j,i}\sum_{\substack{l=1\\ l\neq i}}^{AnAk} N_{j,l}P^*_l}_{=:U_i} \\
&\quad \text{also erhält man für jede Aktie } i\text{:} \\
N_{A_i} &= R_i + \frac{1}{P^*_i}\cdot S_i - \frac{1}{P^*_i}\cdot P_i + \frac{1}{P^*_i}\cdot U_i
\end{aligned}
$$

Man möchte aus dieser Beziehung die neuen Aktienpreise P^*_i bestimmen. Es entsteht ein Gleichungsystem mit $AnAk$ Gleichungen und $AnAk$ Unbekannten (P^*_i). Die $i'te$ Zeile des Gleichungssystems lautet wie folgt:

$$
-S_i + T_i = -(N_{A_i} - R_i)\cdot P^*_i + \sum_{\substack{l=1\\ l\neq i}}^{AnAk} P^*_l \sum_{j=1}^{N_I} x_{j,i}N_{j,l} \tag{2.10}
$$

Das Gleichungssystem läßt sich mit Hilfe von Standardverfahren lösen. [16]

2.5 Aktualisierung der Daten

Mit den oben errechneten Werten können nun die Daten aktualisiert werden. Die Anzahl der Aktien, die jeder Aktionär erhält und sein aktuelles Kapital werden berechnet. Nun werden die Werte des zurückliegenden Tages ausgegeben und - nach einer verdienten Nachtruhe - ein neuer Tag simuliert. Je nachdem, was untersucht werden soll, können zum Beispiel die Kapitalkonzentrationen oder ähnliches ausgegeben werden.

[16] Ich habe das Gaussverfahren benutzt

Kapitel 3

Das verwendete Programm

Dieses Kapitel beschreibt die verwendeten Programmteile.
Zu Testzwecken habe ich Hellthalers Modell [1] analog zu Kapitel 4.2 auf vier Aktien erweitert. Die Ergebnisse waren strukturell die gleichen. [2]

3.1 Programmcode

```
      program boerse
      parameter (Hbeg=0.000100025,Hdelta=0.000125,xdelta=0.2,
     1     NgR=3, Nxx=100, kmax=256, maxz=13000, isuch=100,
     2     nzuf=2048, AnAk=40,NpG=Nxx/NGr, NInv=NpG*NGr,mzu=12345
     3     r=0.0001, rs=1+r, b=0.99, xnull=1E-6,xmini=0.05/AnAk)
      dimension k(NGr), Xmax(AnAk,NGr), AnW(NGr), H(AnAk,kmax),
     1     zx(0:nzuf-1), NAkt(AnAk,NInv),W(NInv),xma(AnAk,AnAk),
     2     Xreal(AnAk,NInv), X(AnAk),Xh(AnAk),ap(AnAk),aph(AnAk),
     3     grad(AnAk),Hx(AnAk,kmax),d(AnAk),di(Anak),c(AnAk),
     4     xbe(AnAk),NA(AnAk),sol(AnAk),aphi(AnAk),imerk(AnAk),
     5     gradh(AnAk)
      real*8 xiit,xli,xre,xdFda,xnhilf,xnhelp
      real*8 xdfda1,xdfda2
      data k /10,133,256/
      do 7 ii=1,AnAk
         ap(ii)=4
         NA(ii)=50*NInv
         di(ii)=1.00015
         d(ii)=0.004
 7       x(ii)=0.1
      do 10 i=1,NInv
         do 14 ii=1,AnAk
 14        NAkt(ii,i)=50
 10        W(i)=50*AnAk*4.0+600.0
```

[1] Vergleiche T. HELLTHALER (1995) Staatexamensarbeit

[2] Über einen gewissen Zeitraum waren die Aktienkurse identisch, wichen aber nach einer gewissen Zeit voneinander ab, da es sich um ein chaotisches System handelt.

```
c   ***** Vorgeschichte erstellen und Zufallszahlen initialisieren *****
      call zufall(Hbeg,Hdelta,nzuf,zx)
      do 20 itime=1,kmax
         do 25 ii=1,AnAk
            mzu=mzu*16807
  25        H(ii,itime)=zx(ishft(mzu,-21))
  20  continue
      call zufall(0,xdelta,nzuf,zx)
      do 30 i=1,NGr
         do 35 ii=1,AnAk
  35        Xmax(ii,i)=(0.8/AnAk)
  30  continue

      do 400 itime=1,maxz

c   ***** Xm(i) (optimaler Aktienanteil) fuer jede Gruppe *****
      do 300 i=1,NGr
         lo=kmax-k(i)+1
         do 21 ii=1,AnAk
  21        x(ii)=0.8/AnAk
         Fmax=0
         do 45 m=lo,kmax
            xhelp=0
            do 50 ii=1,AnAk
  50           xhelp=xhelp+x(ii)*(H(ii,m)-r)
  45        Fmax=Fmax+alog(xhelp+rs)
         fff=fmax
         fhelp=Fmax
         do 200 isu=1,isuch
            if(fff.lt.fhelp)fff=fhelp
            do 55 m=lo,kmax
               do 60 ii=1,AnAk
                  Hx(ii,m)=0
                  do 65 iii=1,AnAk
                     if(iii.ne.ii)Hx(ii,m)=Hx(ii,m)+(x(iii)*
     1               (H(iii,m)-r))/(H(ii,m)-r)
  65              continue
  60           Hx(ii,m)=Hx(ii,m)+rs/(H(ii,m)-r)
  55        continue
            do 70 ii=1,AnAk
  70           grad(ii)=0.0
            do 75 m=lo,kmax
               do 80 ii=1,AnAk
  80              grad(ii)=grad(ii)+1/(x(ii)+Hx(ii,m))
  75        continue
c ******** * * * * * * Normieren * * * * * * * * *************
            xhilf=0.0
            do 85 ii=1,AnAk
  85           xhilf=xhilf+abs(grad(ii))
            if(xhilf.lt.xnull) GOTO 210
```

```
          do 90 ii=1,AnAk
             imerk(ii)=0
 90          grad(ii)=grad(ii)/xhilf
          do 105 ii=1,AnAk
c ******** Hyperebenenspiegelung auf Ebene mit xi=0              ******
             if(grad(ii).lt.0.and.x(ii).le.xnull) then
               grad(ii)=0
               x(ii)=0.0
               imerk(ii)=1
             endif
 105      continue
          xhilf=0.0
          do 115 ii=1,AnAk
 115         xhilf=xhilf+abs(grad(ii))
          if(xhilf.lt.xnull) GOTO 210
          xhelp=0.0
          xhilf=0.0
          do 120 ii=1,AnAk
             xhilf=xhilf+grad(ii)
 120         xhelp=xhelp+x(ii)
          irand=0
c ********* Hier wird auf die Hyperebene sum(xi)=b gespiegelt  *********
          if(xhelp.ge.b.and.xhilf.gt.0) then
 125        xhilf=0.0
            ihilf=0
            irand=1
            do 130 ii=1,AnAk
               ihilf=ihilf+imerk(ii)
 130           xhilf=xhilf+grad(ii)
            if (ihilf.eq.(AnAk-1)) then
              do 131 ii=1,AnAk
                 if(abs(grad(ii)).gt.xnull)then
                   x(ii)=b
                   grad(ii)=0.0
                 endif
 131          continue
              GOTO 210
            endif
            iagain=0
            do 135 ii=1,AnAk
               gradh(ii)=0.0
               if(imerk(ii).ne.1) then
                 gradh(ii)=grad(ii)-(xhilf/(AnAk-ihilf))
               endif
               if(gradh(ii).lt.0.and.x(ii).le.xnull) then
                 iagain=1
                 grad(ii)=0.0
                 gradh(ii)=0.0
                 x(ii)=0.0
                 imerk(ii)=1
```

```
              endif
 135        continue
            if(iagain.eq.0)then
              do 140 ii=1,AnAk
 140            grad(ii)=gradh(ii)
              goto 145
            endif
            goto 125
          endif

 145      xhilf=0.0
 148      xhilf=0.0
          do 150 ii=1,AnAk
 150         xhilf=xhilf+abs(grad(ii))
          if(xhilf.le.0.0) GOTO 210
          do 151 ii=1,AnAk
 151         grad(ii)=grad(ii)/xhilf
 666      alpha1=1000000000.0
          if(irand.ne.1) then
            xhelp=0.0
            xhilf=0.0
            do 155 ii=1,AnAk
               xhelp=xhelp+grad(ii)
 155           xhilf=xhilf+x(ii)
            if (abs(xhelp).eq.0.0) goto 156
            alpha1=(b-xhilf)/xhelp
          endif
 156      if(alpha1.lt.xnull) alpha1=1000000000.0
          alpha2=1000000000.0
          do 160 ii=1,AnAk
             if(grad(ii).lt.-xnull) then
               xhilf=(-x(ii))/grad(ii)
               if(alpha2.gt.xhilf) alpha2=xhilf
             endif
 160      continue
          if (alpha1.lt.alpha2) alpha=alpha1
          if (alpha2.le.alpha1) alpha=alpha2
          if (alpha.eq.1000000000.0) GOTO 210
          do 165 ii=1,AnAk
 165         xh(ii)=x(ii)+alpha*grad(ii)
          Fhelp=0.0
          do 170 m=lo,kmax
             xhelp=0.0
             do 175 ii=1,AnAk
 175            xhelp=xhelp+x(ii)*(H(ii,m)-r)
 170         Fhelp=Fhelp+alog(xhelp+rs)
          ischon=0
c ****************F´(alpha)=0 bestimmen*************
          xdFda1=0.0D+0
          xdFda2=0.0D+0
```

```
        do 236 m=lo,kmax
           xnhilf=0.0D+0
           xnhelp=0.0D+0
           do 237 ii=1,AnAk
              xnhilf=xnhilf+x(ii)*(H(ii,m)-r)
237           xnhelp=xnhelp+grad(ii)*(H(ii,m)-r)
           if(abs(xnhelp).gt.0.0) then
             xdfda1=xdfda1+1/(0.0*alpha+(rs+xnhilf)/xnhelp)
             xdfda2=xdfda2+1/(1.0*alpha+(rs+xnhilf)/xnhelp)
           endif
236     continue
        xdfda1=xdfda1*xdfda2
        xiit=0.0
        if(xdfda2.gt.0) xiit=1.0
        if(xdfda1.le.0.0)then
          iitest=0
          xiit=0.5D+0
          xli=0.0D+0
          xre=1.0D+0
          do 184 iter=1,20
188          xdFda=0.0D+0
             do 186 m=lo,kmax
                xnhilf=0.0D+0
                xnhelp=0.0D+0
                do 187 ii=1,AnAk
                   xnhilf=xnhilf+x(ii)*(H(ii,m)-r)
                   xnhelp=xnhelp+grad(ii)*(H(ii,m)-r)
187             continue
                if(abs(xnhelp).gt.0.0) then
                  xdfda=xdfda+1/(xiit*alpha+(rs+xnhilf)/xnhelp)
                endif
186          continue
             if(abs(xdFda).lt.0.0000000001) goto 189
             if(xdfda.lt.0.0) then
               xre=xiit
               xiit=xiit-(xiit-xli)*0.5
             endif
             if(xdfda.gt.0.0)then
               xli=xiit
               xiit=xiit+(xre-xiit)*0.5
             endif
184       continue
        endif
        if (xiit.eq.0.0) GOTO 210
189     do 863 ii=1,AnAk
863        xh(ii)=x(ii)+alpha*xiit*grad(ii)
        Fhelp2=0.0
        do 861 m=lo,kmax
           xhelp=0.0
           do 862 ii=1,AnAk
```

```
 862             xhelp=xhelp+xh(ii)*(H(ii,m)-r)
 861          Fhelp2=Fhelp2+alog(xhelp+rs)
           if(fhelp2.gt.fff.and.fhelp2.gt.fhelp) then
             ischon=1
             Fhelp=fhelp2
           endif
           if(ischon.ne.1) GOTO 210
 195       if(Fhelp.lt.fff)GOTO 210
           do 230 ii=1,AnAk
 230             x(ii)=x(ii)+alpha*xiit*grad(ii)
 200    continue

 210    do 235 ii=1,Anak
 235       Xmax(ii,i)=x(ii)
        Fma=0.0
        do 201 m=lo,kmax
           xhelp=0.0
           do 202 ii=1,AnAk
 202          xhelp=xhelp+x(ii)*(H(ii,m)-r)
 201       Fma=Fma+alog(xhelp+rs)
        Fhelp=Fma
 300 continue

c     *** Abweichung vom Optimum und Preisbestimmung *****
      NAnf=1
      do 240 j=1,AnAk
 240    c(j)=d(j)-ap(j)*r
      do 245 i=1,NGr
        do 250 j=NAnf,NAnf+NpG-1
           xhelp=0
           do 255 ii =1,AnAk
 255          xhelp=xhelp+NAkt(ii,j)*c(ii)
           W(j)=W(j)*rs+xhelp
           xnorm=0.0
           do 260 ii=1,AnAk
 260          xnorm=xnorm+xmax(ii,i)
 265       mzu=mzu*16807
           xhnorm=xnorm+zx(ishft(mzu,-21))
           if(xhnorm.lt.0.or.xhnorm.gt.(b-0.05)) goto 265
           xnorm=xhnorm
 270       do 275 ii=1,AnAk
 280          mzu=mzu*16807
              xh(ii)=Xmax(ii,i)+zx(ishft(mzu,-21))
 275          if(xh(ii).lt.0.0) goto 280
           xhhelp=0.0
           do 285 ii=1,AnAk
 285          xhhelp=xhhelp+xh(ii)
           do 290 ii=1,AnAk
 290          xh(ii)=xh(ii)/xhhelp*xnorm
           do 295 ii=1,AnAk
```

```
 295          if(xh(ii).lt.(xmini))xh(ii)=xmini
           do 315 ii=1,AnAk
 315          Xreal(ii,j)=xh(ii)
 250    continue
        NAnf=Nanf+NpG
 245 continue
     do 320 j=1,AnAk
        xbe(j)=0
        do 325 i=1,NInv
           xbe(j)=xbe(j)-Xreal(j,i)*W(i)
           do 330 jj=1,AnAk
 330          xbe(j)=xbe(j)+Xreal(j,i)*NAkt(jj,i)*ap(jj)
 325    continue
 320 continue
     do 335 j=1,AnAk
        do 340 jj=1,AnAk
           xma(j,jj)=0
           if (j.eq.jj) goto 340
           do 345 i=1,NInv
 345          xma(j,jj)=xma(j,jj)+Xreal(j,i)*NAkt(jj,i)
 340    continue
 335 continue
     do 350 j=1,AnAk
        xma(j,j)=xma(j,j)-NA(j)
        do 355 i=1,NInv
 355       xma(j,j)=xma(j,j)+Xreal(j,i)*NAkt(j,i)
 350 continue
c   ***** Nun wird die Matrix mit Gausselimination geloest *****
     do 360 ii=1,Anak-1
        do 365 n=ii,Anak
 365       if(abs(xma(n,ii)).gt.0) goto 370
 370    ihilf=n
        if (ihilf.eq.Anak+1) stop
        if (ihilf.eq.ii) goto 375
        do 380 m=1,Anak
           xhilf=xma(ii,m)
           xma(ii,m)=xma(ihilf,m)
           xma(ihilf,m)=xhilf
           xhilf=xbe(ii)
           xbe(ii)=xbe(ihilf)
           xbe(ihilf)=xhilf
 380    continue
 375    do 385 n=ii+1,AnAk
           do 390 m=ii+1,AnAk
 390          xma(n,m)=xma(n,m)-xma(n,ii)/xma(ii,ii)*xma(ii,m)
 385    continue
        do 395 n=ii+1,AnAk
 395       xbe(n)=xbe(n)-xma(n,ii)/xma(ii,ii)*xbe(ii)
 360 continue
     do 405 n=1,Anak
```

```
         xhilf=0.0
         do 415 m=1,n-1
 415        xhilf=xhilf-xma(Anak-n+1,Anak-m+1)*sol(Anak-m+1)
         sol(Anak-n+1)=(xbe(Anak-n+1)+xhilf)/xma(Anak-n+1,Anak-n+1)
 405  continue
      do 420 j=1,AnAk
 420     aph(j)=sol(j)

c   *****            Aktualisierung der Daten *****
      Wges=0
      NAnf=1
      do 425 i=1,AnAk
         aphi(i)=1/aph(i)
 425     c(i)=aph(i)-ap(i)
      do 430 i=1,NGr
         AnW(i)=0
         do 435 j=NAnf,Nanf+NpG-1
            xhelp=0
            do 440 ii=1,AnAk
 440           xhelp=xhelp+NAkt(ii,j)*c(ii)
            W(j)=W(j)+xhelp
            do 445 ii=1,AnAk
 445           NAkt(ii,j)=int(Xreal(ii,j)*W(j)*aphi(ii)+0.5)
            xhelp=0
            do 450 ii=1,AnAk
 450           xhelp=xhelp+NAkt(ii,j)*aph(ii)
            if (xhelp.gt.b*W(j)) then
               do 455 ii=1,AnAk
 455              NAkt(ii,j)=int(Xreal(ii,j)*W(j)*aphi(ii))
            endif
 435        AnW(i)=AnW(i)+W(j)
         Wges=Wges+AnW(i)
 430     NAnf=Nanf+NpG
      Wges=1/Wges
      do 460 i=1,NGr
 460     AnW(i)=AnW(i)*Wges
      do 465 i=2,kmax
         do 470 ii=1,AnAk
 470        H(ii,i-1)=H(ii,i)
 465  continue
      do 475 ii=1,AnAk
        H(ii,kmax)=(aph(ii)+d(ii))/ap(ii)-1
        d(ii)=d(ii)*di(ii)
 475    ap(ii)=aph(ii)
      xhilf=0.0
      do 480 ii=1,AnAk
 480     xhilf=xhilf+ap(ii)
 ****** * * * * Ausgabe der Daten * * * * * **********
      print*,itime,ap(1),ap(2),ap(3),ap(4),xhilf/AnAk
      do 485 ii=1,AnAk
```

```
485    if(ap(ii).lt.0.0) print*,´fehler,ii,apii ´,ii,ap(ii)

400 continue

    stop
    end
c      ***  Zufallszahlengenerator initialisieren ***
    subroutine zufall(erw,delta,nzuf,z)
    real*8 fl,c1,fld
    dimension z(0:nzuf-1)
    xdiff=(1E-6)*(10**(alog(delta)/alog(1000.)))
    c1=xdiff/(dsqrt(4*dasin(dreal(1.)))*delta)
    c2=-0.5/(delta*delta)
    schr2=0.9995/nzuf
    xhi=xdiff/2
    x=xhi
    fl=0.5*(1+c1*dexp(dreal(c2*x*x/4)))
    vfl=0.5+schr2/2
    do 910 i=1,nzuf/2
       vfl=vfl-schr2
920    x=x-xdiff
       fld=x+xhi
       fld=c1*dexp(c2*fld*fld)
       fl=fl-fld
       if (fl.gt.vfl) goto 920
       z(nzuf/2-i)=x
910 continue
    do 930 i=0,nzuf/2-1
       z(nzuf-i-1)=erw-z(i)
       z(i)=erw+z(i)
930 continue
    return
    end
```

3.1.1 Wichtige Variabeln und Konstanten

Feld-Variablen				
Bezeichnung im				
Progr.	**Text**	**Bereich**		**Funktion**
NAkt(i,j)	$N_{j,i}$	i:1..AnAk	j:1..NInv	Anzahl der Aktie i des j'ten Investors
W(j)	K_j	-	j:1..NInv	Vermögen des j'ten Investors
Xreal(i,j)	$x_{j,i}$	i:1..AnAk	j:1..NInv	Anteil von Aktie i des j'ten Investors
k(j)	k_j		j:1..NGr	Gedächtnislänge der i'ten Gruppe
Xmax(i,j)	-	i:1..AnAk	j:1..NGr	Optimaler Anteil von Aktie i der Gruppe j
AnW(j)	-		j:1..NGr	Anteil der i'ten Gruppe am Gesamtkapital
H(i,m)	$H_{t,i}$	i:1..AnAk	m:1..kmax	Rendite der letzten $kmax$ Tage
Hx(i,m)	-	i:1..AnAk	m:1..kmax	$\sum_{i\neq l,i=1}^{AnAk} x_{j,i} \cdot \frac{H_{s,i}-r}{H_{s,l}-r} + \frac{1+r}{H_{s,l}-r}$ [3]
zx	zx	0..nzuf-1	-	Speicher für normalverteilte Zufallszahlen
x(i), xh(i)	-	i:1..AnAk	-	Hilfsvariablen für die Anteile
grad(i)	$\left[\overrightarrow{grad}\right]_i$	i:1..AnAk	-	Die i'te Komponente des Gradienten (im Programm wird auch der im Text als $\overrightarrow{such}$ bezeichnete Vektor mit grad(i) beschrieben
ap(i)	P_i	i:1..AnAk	-	Preis der Aktie i am letzten Tag
aph(i)	P_i^*	i:1..AnAk	-	Preis der Aktie i, der am aktuellen Tag bestimmt wird
d(i)	D_i	i:1..AnAk	-	Dividende der Aktie i am aktuellen Tag
di(i)	-	i:1..AnAk	-	Faktor, um den die Dividende täglich erhöht wird.
NA(i)	N_{A_i}	i:1..AnAk	-	*Konstante* Anzahl der Aktien Nr i die dem Markt zur Verfügung stehen
xma(i,j)	-	i:1..AnAk	j:1..AnAk	Matrix A des Gleichungssystems $Ax = b$ [4]
xbe(i)	-	i:1..AnAk		b aus $Ax = b$

[1] m entspricht dem Index s in der Formel
[2] nähere Erläuterung siehe Text

Konstante Parameter [5]		
Bezeichnung im		
Progr.	**Text**	**Funktion**
AnAk	$AnAk$	Anzahl der Aktien im Markt
Nxx	-	Ungefähre Anzahl der Investoren im Markt
NInv [6]	N_I	Anzahl der Investoren, die insgesamt im Markt sind
NGr	N_G	Anzahl der Gruppen mit verschiedener Gedächtnislänge
NpG	-	Anzahl der Investoren in einer Gruppe [7]
kmax	-	maximal vorkommende Gedächtnislänge in Tagen
maxz	-	Dauer der Simulation in Tagen
r	r	Zinssatz der Renten
rs	-	$rs = 1 + r$
b	b	maximaler Anteil der Aktien am Gesamtkapital
Hbeg	-	Erwartungswert der Rendite der Aktien in der Realität
Hdelta	-	Standardabweichung der Aktienrendite
xdelta	-	Standardabweichung des Rauschens
nzuf	$nzuf$	Anzahl der normalverteilten Zufallszahlen [8]
isuch	-	Anzahl der Durchläufe bei der Maximumsuche [9]
xnull	-	Kleinstmöglicher Wert für eine Komponente i von $\vec{x}$ ab dem bei negativem grad(i), x(i) gleich Null gesetzt wird
xmini	-	Ein Aktie deren Preis Null ist, würde auf Grund ihrer Dividende Gewinn abwerfen. Daher ist der minimale Wert für ein x(i)=xmini
sonstige Variablen		
fmax,fhelp	$F(\dots)$	Viele Abbruchkriterien laufen über den Wert von $F(\dots)$
alpha	α_{opt}	Vergleiche Text (2.2.3)
i	-	Zählvariable, meist für Gruppennummer verwendet
j	j	Zählvariable, meist für Investorennummer verwendet
itime	t	Zeit in Tagen
ii	i	Zählvariable, meist für Aktiennummer verwendet
mzu	-	Variable fuer den IBM 16907 Zufallszahlengenerator

[3]In diesem Teil der Tabelle stehen die als Konstanten deklarierten Parameter

[4]$NInv$ muß durch NGr teilbar sein und wird so aus Nxx bestimmt

[5]$NInv$ ist für alle Gruppen gleich

[6]nzuf muß eine 2-er Potenz sein

[7]Der Suchvorgang wird meistens vorher abgebrochen. Man vergleiche hierzu die Erläuterungen im Text.

3.2 Der Algorithmus

3.2.1 Zufallszahlen

In diesem Abschnitt erläutere ich die Verwendung und Erzeugung der normalverteilten Zufallszahlen, die im Programm verwendet werden [10].

Die Verwendung der Zufallszahlen

Ich stelle zunächst die Verwendung der Zufallszahlen beim Einaktienmodell vor [11]. Zuerst wird der Anteil x des Kapitals, das der Aktionär in die Aktie investieren will, mit Hilfe der Utilityfunktion bestimmt. Zu x wird nun eine normalverteilte Zufallszahl addiert: $x' = x + zuf$. [12] Sollte $x' > b$ oder $x' < 0$ sein, so wird eine neue Zufallszahl bestimmt und erneut zu x addiert, bis x' den Randbedingungen genügt.

Bei mehreren Aktien wird die rational ausgerechnete, optimale Anlage des Kapitals durch den Vektor $\vec{x} = (x_1, \dots, x_{AnAk})$ bestimmt. Neben den einzelnen Aktienanteilen wird auch der Gesamtaktienanteil $\left(\sum_{i=1}^{AnAk} x_i\right)$ verändert. Ich habe sowohl den Anteil der einzelnen Aktien, als auch den Gesamtaktienanteil wie folgt durch Zufallszahlen verändert:

Zu Beginn setze ich $x_{norm} = \sum_{i=1}^{AnAk} x_i$. Ich addiere nun eine Zufallszahl zu x_{Norm} und erhalte $x_{NeuNorm} = x_{Norm} + zuf$. Falls $x_{NeuNorm}$ nicht den Randbedingungen entspricht, wird eine neue Zufallszahl bestimmt und entsprechend zu x_{Norm} addiert, bis $x_{NeuNorm}$ die Randbedingungen erfüllt.

Nun wird zu allen x_i die jeweils neu erzeugte Zufallszahl zuf addiert, und man erhält [13] $\vec{x}' = (x_1 + zuf_1, \dots, x_{AnAk} + zuf_{AnAk})$. Diesen Vektor normiere ich auf $x_{NeuNorm}$. Der neue Vektor lautet:

$$\vec{x}'' = \left(\frac{x'_1 \cdot x_{NeuNorm}}{\sum_{i=1}^{AnAk} x'_i}, \dots, \frac{x'_{AnAk} \cdot x_{NeuNorm}}{\sum_{i=1}^{AnAk} x'_i} \right). \tag{3.1}$$

Wegen der Dividendenausschüttung darf kein $x''_i, i \in \{1, \dots, AnAk\}$ Null werden. Ich habe einen minimalen Wert für x''_i in $xmini$ festgelegt. Allen x''_i, die diesen Wert unterschreiten wird der Wert $xmini$ zugewiesen [14].

Erstellung der Zufallszahlen

Zuerst stelle ich den theoretischen Hintergrund zur Erstellung von normalverteilten Zufallszahlen dar. [15]

Die *Wahrscheinlichtkeitsdichte* der Normalverteilung ist:

$$p(x) = \frac{1}{\sqrt{2\pi}\delta} e^{-\frac{(x-\mu)^2}{2\delta^2}} \quad \text{für } x \in]-\infty, \infty[,$$

[10] Bei D. Stauffer u. a. (1993) wird die Erzeugung von gleichverteilten Zufallszahlen, unter anderem des von mir verwendeteten IBM16887 - Zufallszahlengenerators erläutert.

[11] So, wie sie von Hellthaler (1995) (Staatsexamensarbeit) eingesetzt wurden.

[12] zuf bezeichne hier und im Folgenden eine normalverteilte Zufallszahl.

[13] wenn $(x_i + zuf) < 0$ ist wird ein neues zuf generiert bis $(x_i + zuf) \geq 0$.

[14] Ich habe dabei einen kleinen Puffer gelassen, so daß durch das eventuelle Hochsetzen vieler Aktienanteile von Null auf x_{mini}, $\sum_{i=1}^{AnAk} x''_i$ nicht grösser als b werden kann.

[15] Vergleiche hierzu F. Vesely (1994) Seite 61ff

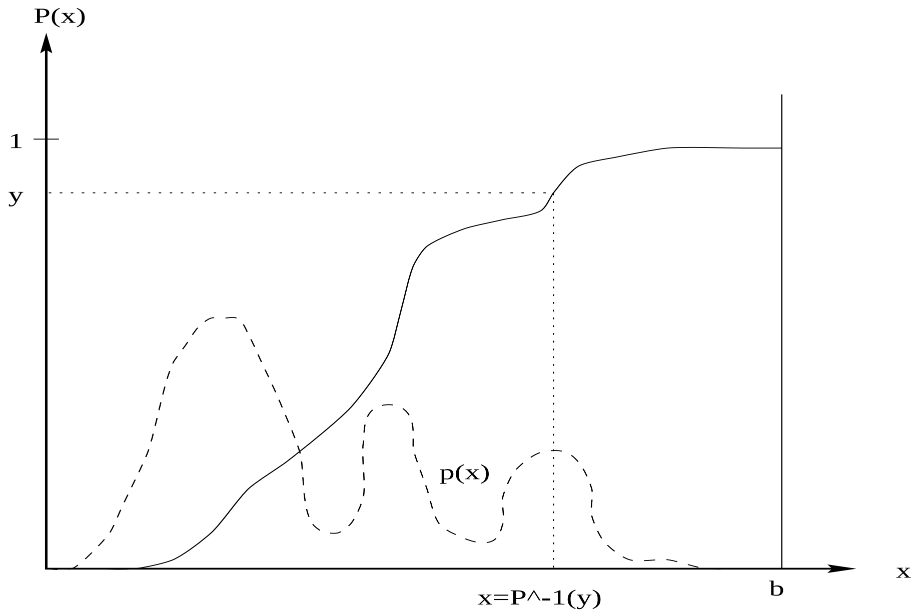

Abbildung 3.1: Geometrische Interpretation der Transformationsmethode *(Zeichnung nicht exakt!)*

dabei ist

$$p(x_0)dx \equiv \mathcal{P}\left\{x \in [x_0, x_0 + dx]\right\}$$

Die *Verteilungsfunktion* der Normalverteilung ist:

$$F(x) = \int_{-\infty}^{x} p(z)dz,$$

Mit

$$F(x_0) \equiv \mathcal{P}\{x \leq x_0\}.$$

F(x) ist monoton wachsend.
Wenn die Verteilungsfunktion $F(x)$ und ihre Umkehrfunktion $F^{-1}(y)$ bekannt ist, und man über gleichverteilte Zufallszahlen verfügt, geht man wie folgt vor: [16]

- Wähle y aus einer gleicherverteilten Zufallsverteilung im Intervall (0,1).
- Berechne $x = F^{-1}(y)$

So erhält man für x die gewünschte Wahrscheinlichkeit $p(x)$.
In Abbildung 3.1 kann man erkennen, daß die Regionen von x, in denen $P(x)$ steiler verläuft, öfter „getroffen“ werden.
Da die Verteilungsfunktion der Normalverteilung nicht invertierbar ist und aus Zeitgründen nicht für jede benötigte Zufallszahl numerisch berechnet werden soll, erstellt man $nzuf$ normalverteilte Zufallszahlen, legt sie in dem Feld $zx(0 : nzuf)$ ab und greift auf dieses mit Hilfe von gleichverteilten Zufallszahlen [17]

[16] Siehe auch F. Vesely (1994)
[17] Diese werden mit Hilfe des IBM16807 Zufallszahlengenerator erzeugt.

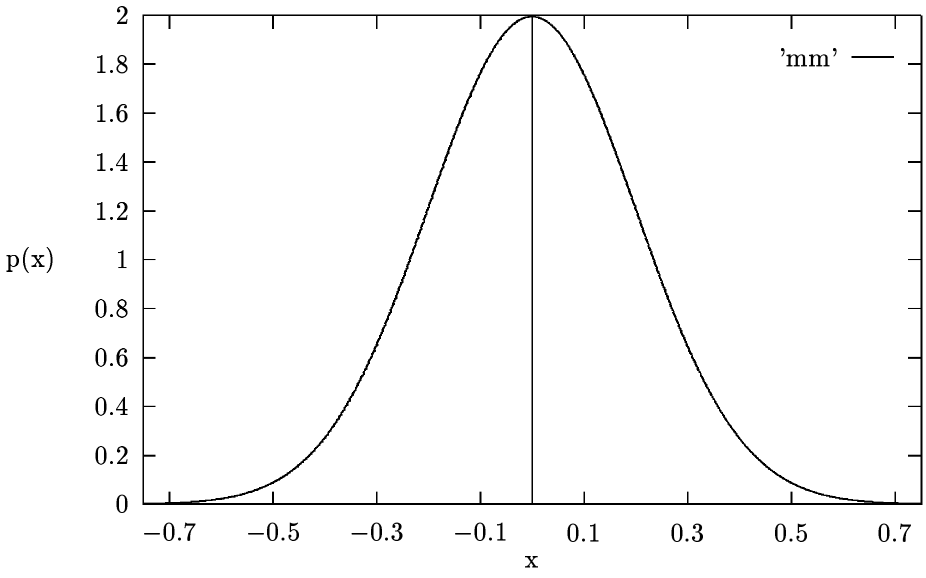

Abbildung 3.2: Wahrscheinlichkeitsdichte der Normalverteilung ($\mu = 0, \delta = 0.2$)

zu. Man bestimmt also nur diskrete Werte von $F^{-1}(y)$. Da die Wahrscheinlichkeitsdichte um den Erwartungswert symmetrisch ist, reicht es, nur eine Hälfte zu bestimmen.
Man startet mit $x = 0.5$. Näherungsweise ist $\int_0^{0.5} p(z)dz = \int_0^{\infty} p(z)dz = 0.5$, wie man aus Abbildung 3.2 ersehen kann ($F(\infty) = 1!$). Man unterteilt das Intervall [0,0.5] in $\frac{nzuf}{2}$ Teile y_j. Jetzt läßt man x immer kleiner werden und vergleicht $F(x)$ mit dem entsprechenden y_j [18]. Daraus erstellt man die $nzuf$ normalverteilten Zufallszahlen.
Im Programm habe ich dies wie schon Hellthaler folgendermaßen realisiert:
In der Subroutine (zufall(erw,delta,nzuf,z) wird $F^{-1}(y)$ berechnet. Die Unterteilung des Intervalls [0,1] erfolgt durch $y_j = \epsilon + i \cdot \frac{1-2\epsilon}{nzuf-1}$. [19] Zunächst wird $F^{-1}(y)$ für μ=0 bestimmt, indem $y = F(x)$ an diskreten Punkten mit Abstand $\Delta x = xdiff$, von x=0.5 aus, in negativer Richtung berechnet wird. Dabei verkleinert sich $F(x)$ monoton. Man achtet darauf, bei welchem x, $\quad F(x) < i \cdot \frac{1-\epsilon}{nzuf}$ wird. An diesen Stellen gilt näherungsweise:
$F^{-1}\left(\epsilon + i \cdot \frac{1-2\epsilon}{nzuf-1}\right) = x$, damit ist also $z(i) = x$.
Um schneller zu rechnen, werden $c1$ und $c2$ definiert: $c1 := xdiff \cdot \frac{1}{\sqrt{2\pi\delta}}, c2 := \frac{1}{2\delta^2}$. Es gilt: $\Delta F(x) = xdiff \cdot p(x) = c1 * exp(c2 * x * x)$. Die eigentliche Berechnung der z(i) erfolgt in einer Doppelschleife. In der äußeren do 910-Schleife wird jeweils der Wert $y = vfl = F(\tilde{x})$ errechnet, den $y = F(x)$ als nächstes erreicht. In der inneren 920...goto 920-Schleife wird dann jeweils x verkleinert und $F(x)$ berechnet, bis vfl erreicht ist. Anschließend wird das Berechnete x in

[18] Ist $F(x) = y_j$ so gilt ja $F^{-1}(y_j) = x$!

[19] $\epsilon = 0$ ist nicht möglich, weil $F^{-1}(y) \stackrel{y \to 0}{\longrightarrow} -\infty$. Im Programm ist ϵ=0.0005.

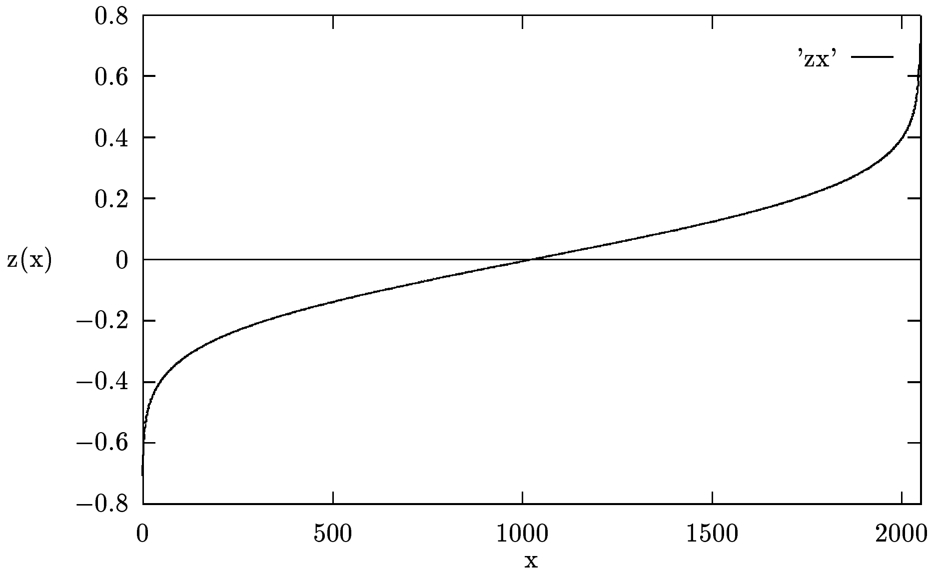

Abbildung 3.3: Das Feld $z(x)$ gegen x $(\mu = 0, \delta = 0.2)$

z abgespeichert. Da, wie oben erläutert, der äußere Zähler von 1 bis $\frac{nzuf}{2}$ statt von $\frac{nzuf}{2} - 1$ läuft, wird x in $z(\frac{nzuf}{2} - i)$ abgespeichert.
Da x am Anfang der inneren Schleife geändert wird, aus Symmetriegründen aber $x = 0.5 - \frac{1}{2}\Delta x$ der erste Punkt sein soll, an dem $F(x)$ berechnet wird, muß vor der ersten Iteration $x = 0.5 + \frac{1}{2}\Delta$ gesetzt werden. Entsprechend ist dann $fl = F(x) = F(0.5 + \frac{1}{2}\Delta x) \approx 0.5 - \Delta x \cdot (0.5 + \frac{1}{2}x)$.
Da die Normalverteilung bezüglich des Erwartungswertes symmetrisch ist, können nach der do 50 Schleife die restlichen Werte durch $F^{-1}(\mu - y) = F^{-1}(\mu + x)$ in der do 60 Schleife berechnet werden. Gleichzeitig wird dabei $F^{-1}(y)$ für $\mu = erw$ durch $F^{-1}_{\mu=\mu_1} = F^{-1}_{\mu=0} + \mu_1$ berechnet.
Wählt man xdiff zu klein, dauert die Initialisierung sehr lange. Außerdem kann aufgrund der begrenzten Genauigkeit des Rechners $c1 * dexp(c2 * fld * fld) = 0$ oder $fl - fld = fl$ werden und so wird die Schleife nicht mehr terminiert. Um dennoch genügend kleine Schritte machen zu können, werden fl, c1 und fld als real*8 (doppelte Rechengenauigkeit) deklariert. Wählt man xdiff zu groß entspricht z(i) nicht mehr $F^{-1}(y)$
Ich habe in Abbildung 3.3 $z(x)$ gegen x aufgetragen.

3.2.2 Das Kernprogramm

Ich stelle den Ablauf des Programms zuerst grob dar und erläutere das Programm danach schrittweise.
Die DO 400 Schleife läuft über die Tage (1..maxz)

⋆ **Die DO 300 Schleife läuft über die ver. Ged.gruppen (1..NGr)**
⋆ ∘ **Die DO 200 Schleife läuft über die maximalen Suchläufe** [20]
⋆ ∘ ⋄ Berechnung und Normierung des Gradienten von $F(\dots)$.
⋆ ∘ ⋄ Hyperebenenspiegelung I
⋆ ∘ ⋄ Hyperebenenspiegelung II
⋆ ∘ ⋄ Bestimmung von α_{opt}
⋆ ∘ **Rücksprung zur DO 200 Schleife**
⋆ ∘ **Label 210**
⋆ **Rücksprung zur DO 300 Schleife**
⋆ Abweichung vom Optimum und Bestimmung des Preises ($\tilde{K}_j$).
⋆ Aufstellen des Gleichungssystems (vgl. Formel2.4)
⋆ Lösen des Gleichungssystems
⋆ Aktualisierung der Daten [21]
⋆ Ausgabe der zu untersuchenden Daten.
Rücksprung zur DO 400 Schleife.
Subroutine, in der die Zufallszahlen bestimmt werden.

Berechnung des Suchvektors

Dieser Abschnitt stellt den Programmteil dar, der den Suchvektor bestimmt.
Die 300-er Schleife läuft über die verschiedenen Gruppen - für jede Gruppe wird der Suchvektor einzeln bestimmt.
Die 200-er Schleife läuft über die maximale Anzahl an Suchläufen. Die Anzahl der mit isu bezeichneten Durchläufe spielt für die Dauer des Suchvorgangs keine Rolle, da diese Schleife immer über eine der später beschriebenen $GOTO210$ Marken verlassen wird. Ein Schleifenabbruch kann dadurch bedingt sein, daß der Suchvektor (nach eventuellen Hyperebenenspiegelungen) dem Nullvektor entspricht oder daß die begrenzte Rechnergenauigkeit das Auffinden eines neuen Iterationswertes verhindert. Letzteres ist dann der Fall, wenn der neue Iterationswert, eingesetzt in die zu maximierende Funktion $F_j(\dots)$, keinen grösseren Wert ergibt, als der alte [22].
In der 55-er Schleife wird der Gradient, wie auf Seite 8 beschrieben, berechnet. Die Feldvariable $Hx(l,s)$ steht für $\sum_{\substack{i=1 \\ i\neq l}}^{AnAk} \left(x_{j,i} \frac{H_{s,i}-r}{H_{s,l}-r} \right) + \frac{1+r}{H_{s,l}-r}\ Hx(l,s)$ und ist eine Hilfsvariable für die Berechnung der Ableitung nach der l'ten Koordinate. In der 75-er Schleife wird schließlich mit Hilfe von $Hx(l,s)$ der Gradient bestimmt und in der 90-er Schleife normiert.
Die 105-er Schleife realisiert die Hyperebenenspiegelung auf die einzelnen Koordinatenachsen. [23] Ob dies notwendig ist, wird mit der ersten IF Anweisung geprüft. Wenn für ein i: $x(i) \leq xnull$ und $grad(i) < 0$ gilt, würde der Suchvektor aus dem zulässigen Bereich zeigen. Daher wird die i'te Komponente des Suchvektors [24] gleich Null gesetzt. Weiterhin erhält $imerk(i)$ den Wert 1. [25] Die In-

[20] Nach jedem der Schritte werden Kontrollmechanismen durchlaufen, die prüfen, ob der Iterationswert schon optimal ist. Sollte dies der Fall sein, wird zu Label 210 gesprungen.
[21] Umfaßt die Berechnung der neuen Aktienpreise und des Vermögens der einzelnen Aktionäre.
[22] isu muß allerdings groß genug sein, damit der optimale Suchvektor gefunden werden kann.
[23] Hyperebenenspiegelung I
[24] Im Programm bleibt die Bezeichnung $grad(i)$ statt $such(i)$.
[25] Das Feld $imerk(AnAk)$ wird vorher mit Null initialisiert.

formation, auf welche Koordinatenachse eine Hyperebenspiegelung durchgeführt wird, wird im weiteren Programmablauf verwendet. Es ist möglich, daß durch die in der 105-er Schleife durchgeführten Maßnahmen $\sum_{i=1}^{AnAk} |grad(i)| = 0$ geworden ist. Dies wird mit der IF Anweisung nach der 115-er Schleife geprüft. Ist dies der Fall, so wird die Suche nach dem Maximum abgebrochen [26] , da der gefundene Punkt schon optimal ist.
Die Zeile IF(xhelp.ge.(b*0.999).and.xhilf.gt.0) THEN leitet die zweite Hyperebenenspiegelung ein. Die ihr folgenden Anweisungen werden nur abgearbeitet, wenn die Summe über die einzelnen Anteile den maximal möglichen Wert b erreichen, und der Gradient aus dem erlaubten Gebiet zeigt. Die Variabel *irand* wird auf 1 gesetzt und zeigt so an, daß der aktuelle Iterationswert auf dem Rand liegt.
Die Anweisung IF (ihilf.eq.(AnAk-1)) THEN fängt den Fall ab, daß für $i \in \{1, \ldots, AnAk\}$ bis auf ein $i =: i_{nicht}$ eine Hyperebenenspiegelung auf die Koordinatenrichtung durchgeführt worden ist. Ist dies der Fall, so ist $\vec{x}$ bereits optimal und der Suchvorgang wird abgebrochen [27].
Die 135-er Schleife enthält die eigentliche Hyperebenenspiegelung II. Durch die Spiegelung kann eine erneute Hyperebenenspiegelung auf die Koordinatenachsen nötig geworden sein. Diese wird, wie oben beschrieben, durchgeführt. Dies hat wiederum zur Folge, daß eine erneute Hyperebenenspiegelung II notwendig ist. Daher wird die Variable *iagain* auf 1 gesetzt. Der Programmteil der Hyperebenenspiegelung wird verlassen, wenn *iagain* Null ist und ansonsten zu der Marke 125 gesprungen, um die Spiegelung II erneut durchzuführen.
Die 150-er Schleife prüft erneut, ob $\sum_{i=1}^{AnAk} |grad(i)| = 0$ ist. Das könnte zum Beispiel auftreten, wenn vor der Hyperebenenspiegelung II $grad(s) = grad(j) \neq 0$, und $\forall\, i \overset{i \neq s \wedge i \neq j}{\in} \{1, \ldots, AnAk\} : grad(i) = 0$ ist. Auch in diesem Fall ist $\vec{x}$ optimal.
Nun folgt der Teil des Programms, der α_{opt} bestimmt. *Alpha*1 wird nur bestimmt, wenn $irand \neq 1$ ist, da es sonst Null wäre.
Für α_1 muß gelten: $\sum_{i=1}^{AnAk} x(i) + \sum_{i=1}^{AnAk} grad(i) = b$, daher ist

$$\alpha_1 = \frac{b - \sum_{i=1}^{AnAk} x(i)}{\sum_{i=1}^{AnAk} grad(i)}. \tag{3.2}$$

Im Programm wird abgefangen, wenn $\sum_{i=1}^{AnAk} grad(i) = 0$ ist, da es ansonsten zu einer Division durch Null führen würde. In der Realität bedeutet dies, daß es dem Aktionär sinnvoll erscheint, den gleichen Gesamtanteil an Aktien zu kaufen, wie bei dem vorherigen Iterationswert, aber die einzelnen Anteile zu verändern. Außerdem wird getestet ob $\alpha_1 < xnull$ ist, der Aktionär also seinen Gesamtaktienanteil verringern möchte.
Für jede Koordinatenrichtung i erhält man unter der Voraussetzung $grad(i) < 0$:

$$\alpha_{2_i} = \frac{-x(i)}{grad(i)}. \tag{3.3}$$

Von den α_{2_i} wird nun das kleinste ausgewählt.
Jetzt kann $alpha = \min(\alpha_1, \alpha_2)$ bestimmt werden. Wenn sich sowohl für α_1, als auch für α_2 kein Wert ergibt, so wird die Suche abgebrochen, da $\vec{x}$ unter den

[26] Der Abbruch erfolgt, wie oben angemerkt, mit der Sprunganweisung GOTO 210.
[27] Das x(i_{nicht}) wird also von $0.999 * b$ auf b gesetzt.

gegebenen Nebenbedingungen optimal ist.
In der Hilfsvariable $xh(AnAk)$ wird $\vec{x}$ abgespeichert. Wie auf Seite 10 gezeigt, hat sich das Problem der Maximumsuche auf die Funktion $\tilde{F}(\alpha)$ reduziert:

$$\frac{d\tilde{F}_j(\alpha)}{d\alpha} = \sum_{s=1}^{t-k_j} \frac{1}{\alpha + \frac{(1+r)+\mathcal{A}_S}{\mathcal{B}_S}}.$$

Die Nebenbedingung von $\tilde{F}(\alpha)$ lautet $0 \leq \alpha_{opt} \leq \alpha$.
Die Hilfsvariablen $xnhilf$ steht für $\mathcal{A}_S$, $xnhelp$ für $\mathcal{B}_S$. In der 236-er Schleife wird nun $xdfda1 = \frac{d\tilde{F}_j(0\cdot\alpha)}{d\alpha}$ und $xdfda2 = \frac{d\tilde{F}_j(1\cdot\alpha)}{d\alpha}$ errechnet. Es können folgende Fälle auftreten:
Das Maximum von $\tilde{F}(\alpha)$ liegt

- bei $\alpha = 0$,
- bei $\alpha = alpha$ oder
- in $(0, alpha)$.

Nur im dritten Fall liegt ein Maximum von $\tilde{F}(\alpha)$ mit $\frac{d\tilde{F}(\alpha_{opt})}{d\alpha} = 0$ vor. Ich definiere: $xdfda1 := \frac{d\tilde{F}_j(0\cdot\alpha)}{d\alpha} \cdot \frac{d\tilde{F}_j(1\cdot\alpha)}{d\alpha}$, $xdadf2$ bleibt wie oben beschrieben. Die Variabel $xiit$ beschreibt den Faktor, mit dem $alpha$ multipliziert werden muß, damit $\alpha_{opt} = xiit \cdot alpha$ wird. Folgende Fälle sind abzuarbeiten:

- $xdfda2 \leq 0$ und $xdfda1 > 0$. Das bedeutet, daß sowohl die Steigung bei $0 \cdot alpha$, als auch bei $1 \cdot alpha$ negativ ist. $xiit$ wird also auf Null gesetzt. In diesem Fall ist die Suche abgeschlossen, das optimale $\vec{x}$ schon gefunden.
- $xdfda2 > 0$ und $xdfda > 0$. Die Steigung bei $1 \cdot alpha$ und $0 \cdot alpha$ ist positiv. Daher wird $xiit$ auf 1 gesetzt.
- $xdfda2$ und $xdfda1$ haben verschiedene Vorzeichen. Das Maximum wird, wie weiter unten beschrieben, gesucht.

Im dritten Fall wird das Maximum mit Hilfe einer Intervallschachtelung gesucht. Es wird mit $xiit = 0.5$ begonnen und $\frac{d\tilde{F}}{d\alpha}(0.5 \cdot alpha)$ berechnet. Ist dies negativ, befindet sich das Maximum von $\tilde{F}$ bei einem größeren Wert von $xiit$, ist es positiv bei einem kleineren Wert. Entsprechend halbiert man die verbleibenden Intervalle, bis entweder $\tilde{F}(xiit \cdot alpha)$ einen Grenzwert nahe Null erreicht, oder die festgelegten Iterationschritte (ich wählte 20), durchlaufen sind.
Nun wird $\forall \quad i \in \{1, .., AnAk\} \quad xh(i) = x(i) + \alpha_{opt} \cdot xiit \cdot grad(i)$ gesetzt. Von dem neuen Iterationspunkt $\vec{xh}$ wird $F(\dots)$ berechnet und überprüft, ob $F(\vec{xh}) > F(\vec{x})$ ist. Ist dies nicht der Fall, wird die Suche abgebrochen, da $\vec{x}$ schon optimal ist. Ansonsten wird zum Beginn der 200-er Schleife gesprungen.

Berechnung der Vermögensverhältnisse

Zunächst wird für alle Aktionäre $\tilde{K}_j$ berechnet (vgl. Seite 12). Die Berechnung erfolgt für jeden Aktionär getrennt. Dieser Wert wird unter $W(j)$ gespeichert, der alte Wert $W(j)$ wird nicht mehr benötigt. In der gleichen Schleife werden Zufallszahlen, wie oben beschrieben, hinzugefügt.

Nun wird, wie in Kapitel 2.4 erwähnt, das Gleichungssystem $Ax = b$ berechnet. A ist eine Matrix und wird im Feld $xmax(i,j)$ gespeichert, $\vec{b}$ ist ein Vektor, der in $xbe(i)$ gespeichert wird [28]. Das so erhaltene Gleichungssystem habe ich mit dem Gaußverfahren gelöst.

Gauß-Elimination. Dreieckszerlegung einer Matrix

Beim Gaußschen Eliminationsverfahren zur Lösung von

$$Ax = b \tag{3.4}$$

wird, durch geeignete Vertauschungen und Linearkombinationen von Gleichungen des gegebenen Systems, schrittweise in ein Gleichungssystem der Form [29]

$$Rx = c,$$

transformiert, das die selben Lösungen wie 3.4 besitzt. R ist eine obere Dreiecksmatrix, $Rx = c$ also ein „gestaffeltes“ Gleichungssystem, das man leicht lösen kann. Formal kann man den Algorithmus wie folgt beschreiben:

1. Bestimme ein Element $a_{r1} \neq 0$ und fahre mit 2) fort; falls kein solches r existiert, setze $(A', b') := (A, B)$, A ist singulär, stop

2. Vertausche die Zeilen r und 1 von (A,b). Das Resultat ist die Matrix $(\bar{A}, \bar{b})$

3. Subtrahiere für $i = 2, 3, \ldots, n$, das l_{i1}-fache, $l_{i1} := \frac{\bar{a}_{i1}}{\bar{a}_{11}}$, der Zeile 1 von Zeile i der Matrix $(\bar{A}, \bar{b})$. Als Resultat erhält man die gesuchte Matrix (A', b').

Aktualisierung der Daten

Nun werden - wieder in einer Doppelschleife - die Daten der einzelnen Investoren gruppenweise aktualisiert. In der 435-er Schleife wird dazu zunächst das Kapital der Investoren neu berechnet (K_j^*). Anschließend kann aus dem Aktienanteil $Xreal(i,j)$ das Kapital und die Anzahl der Aktien eines Investors neu bestimmt werden. Dabei ist darauf zu achten, daß nur ganze Aktien gekauft werden können, weshalb hier gerundet wird. Es wird abgerundet,wenn durch Aufrunden der Aktienanteil den Maximalwert b überschreitet.
In $AnW(i)$ wird das Gesamtkapital einer Gedächtnisgruppe berechnet. Dieser Wert wird für die eigentliche Simulation nicht benötigt, ist aber zur Analyse des Marktgeschehens wichtig. Danach wird der relative Anteil der Gruppen am Gesamtkapital berechnet. Weiterhin müssen die zurückliegenden Renditen aktualisiert werden: jede Rendite verschiebt sich in der do 470 Schleife um einen Tag, und die aktuelle Rendite wird errechnet. Auch die Aktienpreise und Dividenden werden aktualisiert, die Werte des Vortages werden nicht mehr benötigt. Nun können die Ausgaben erfolgen, und es wird zum nächsten Tag übergegangen. Insgesamt umschließt die do 400 Schleife das Hauptprogramm. Das Programm wird beendet, wenn $itime$ den Wert $maxz + 1$ erreicht.

[28] Die Umsetzung in den Programmcode ist zwar nicht schwer, aber eine grausame Indizesschlacht...
[29] Vergleiche J.Stoer (1983)

Kapitel 4

Ergebnisse

4.1 Das Einaktienmodell

Bevor ich auf die Ergebnisse der Simulation mit mehreren Aktien eingehe, möchte ich kurz die Ergebnisse des Modells mit einer Aktie erläutern. Dies wurde von Levy, Levy und Solomon entwickelt und im Rahmen einer Staatsexamensarbeit von Hellthaler näher untersucht.
Zuerst betrachte ich Märkte mit 100 oder weniger Investoren. Man kann die Kursverläufe in drei Gruppen unterteilen. Zwei haben regelmäßige (periodische) Aktienpreis - Zeit - Diagramme, die andere chaotische. Interessant und am ehesten realistisch sind die chaotischen Verläufe, da bei den anderen der Kursverlauf vorhersagbar wäre.
Ein Beispiel für die erste reguläre Gruppe ist in Abbildung 4.1 dargestellt. Der Preis variiert hier zwischen zwei sehr unterschiedlichen Niveaus. Der Übergang von einem Level zum anderen verläuft plötzlich. Die Rechteckstruktur kann man dadurch erklären, daß, solange der rasante Anstieg von dem Investor erinnert wird, er sich für die Aktie entscheiden wird. Sobald er den krassen Anstieg vergisst, entscheidet er sich für Renten als Geldanlage. Da alle Investoren diese Entscheidung zur gleichen Zeit tätigen, führt dies zu dem beschriebenen Verlauf.
Bei der zweiten Gruppe, die einen regulären Kursverlauf vorweist, ist dieser dreieckförmig. Da es sich aber wie schon oben erwähnt um vorhersehbare Entwicklungen handelt, sind sie nicht realistisch und ich werde hier auch nicht weiter auf sie eingehen.
Bei den chaotischen Kursverläufen schwankt der Preis der Aktie auf einem *normalen* Level und der Preisverlauf ist nicht vorhersagbar. (Vergleiche Abbildung 4.2).
Hellthaler hat herausgefunden, daß man, um solche chaotischen Kursverläufe zu erzielen, folgende Punkte beachten muß:

- Mindestens eine Gruppe muß eine hohe Gedächtnislänge haben.
- Je mehr verschiedene Gruppen es gibt, desto chaotischer ist der Kursverlauf.
- Die Gruppen mit kleinen Gedächtnislängen sollten nicht zu großen Einfluß gewinnen.

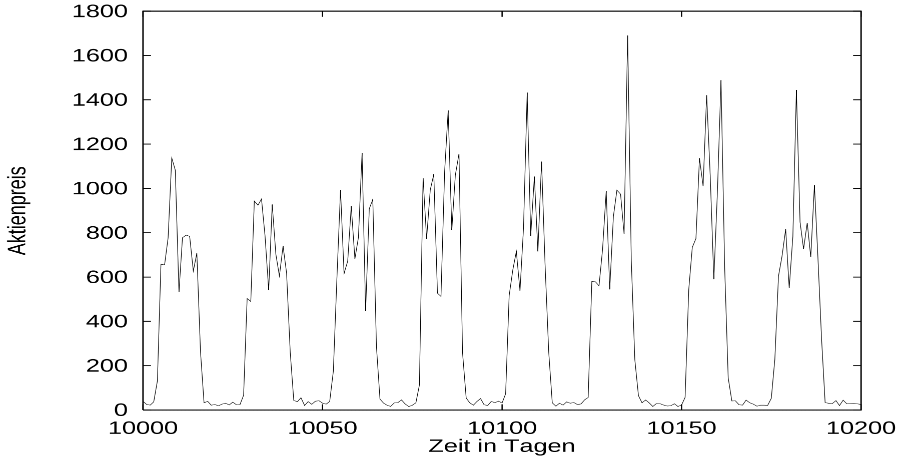

Abbildung 4.1: AnAk=1, NInv=100, NGr=1, k=10

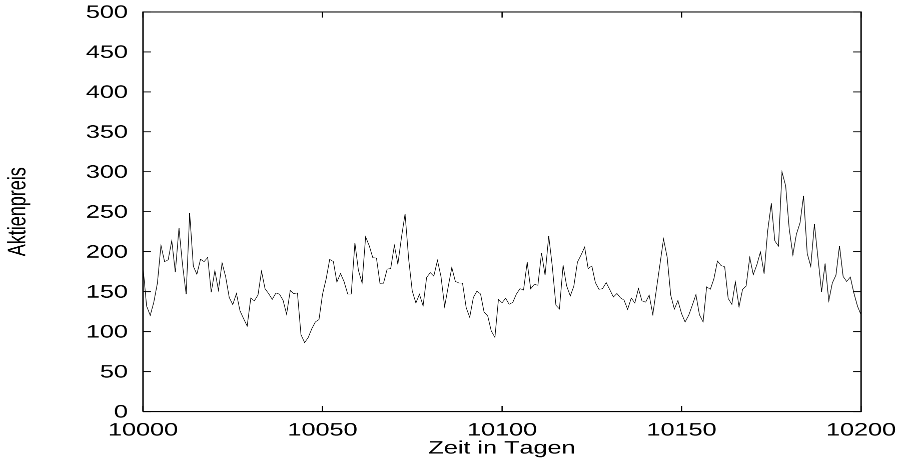

Abbildung 4.2: AnAk=1, NInv=100, NGr=6, k=/10,36,141,193,256,420/

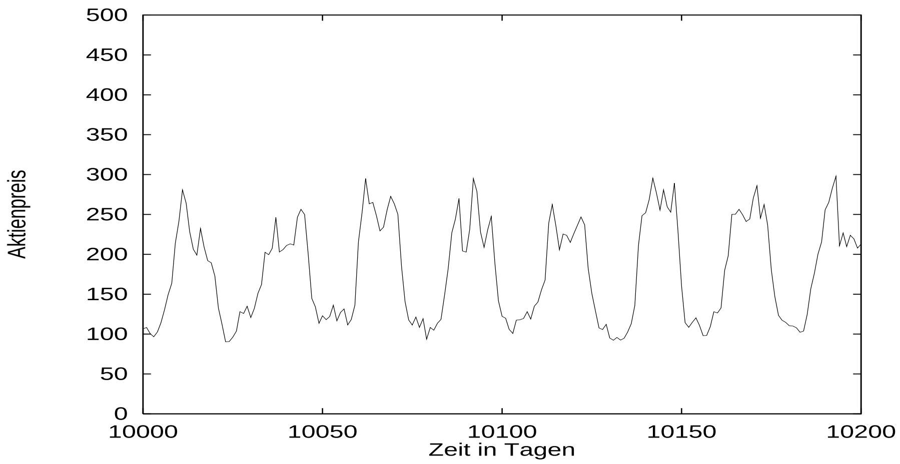

Abbildung 4.3: AnAk=1, NInv=1000, NGr=6, k=/10,36,141,193,256,420/

Hellthaler (1995) hat herausgefunden, daß die Kursverläufe für **alle** Märkte bei hohen Investorenzahlen, periodisch werden. Abbilung 4.2, 4.3 und 4.4 verdeutlichen dies. Man erkennt, daß mit wachsender Investorenzahl der Kursverlauf periodisch wird.
In der Realität handeln jedoch viele Investoren an der Börse. Es könnte jedoch sein, daß der Markt von einigen wenigen professionellen Investoren, die einen großen Pool an Aktien zur Verfügung haben, bestimmt wird.
Eine weitere Möglichkeit wäre zu untersuchen, ob eine höhere Anzahl verschiedener Aktien zu sinnvollen Ergebnissen führt. Dies wird in der vorliegenden Arbeit untersucht.

4.2 Geschwindigkeitsbetrachtung

Hellthaler hat sein Modell auf 2 Aktien erweitert. Ich habe das Programm für viele Aktien jedoch völlig anders konzipiert, da Hellthalers Suche nach dem Maximum der Utilityfunktion bei mehr als 2 Aktien viel zu zeitintensiv geworden wäre. Bei zwei Aktien ist das neue Modell etwa um den Faktor zwei schneller als das alte.
Bei folgenden Parametern:

- Anzahl der simulierten Tage : 4000
- Anzahl der Investoren : 1000
- Anzahl der Gruppen verschiedener Gedächtnislängen : 5 (10,26,141,197,256) [1]

Die Zeiten, welche die beiden Versionen (auf dem gleichen Rechner) benötigt haben:

[1] Die Zahlen in den Klammern stehen für die Längen der einzelnen Gedächtnisgruppen.

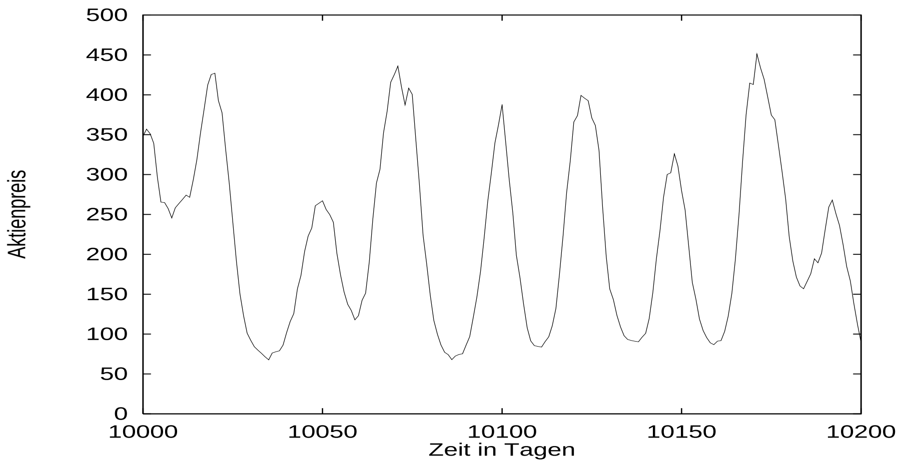

Abbildung 4.4: AnAk=1, NInv=100000, NGr=6, k=/10,36,141,193,256,420/

- Nach der *alten* Methode: **1043 Sekunden**
- Mit Hilfe der Gradientenmethode: **509 Sekunden** [2]

Man hätte das Modell von Hellthaler auch einfacher auf mehrere Aktien erweitern können. Ich habe das für drei Aktien einmal vorgeführt, um zu belegen, daß es für mehr Aktien nicht mehr sinnvoll anwendbar ist. Ich habe nur den Teil, in dem das Maximum berechnet wird, wie folgt geändert:

```
      do 300 i=1,NGr
         lo=kmax-k(i)+1
         Fmax=0.0
         x(1)=0.0
         x(2)=0.0
         x(3)=0.0
         x3max=0
140      do 45 m=lo,kmax
45          H1x(m)=(x(1)*(H(1,m)-r)+x(2)*(H(2,m)-r)+rs)/(H(3,m)-r)
         F2=0
         F1=0
         do 100 m=lo,kmax
            F1=F1+1/H1x(m)
100         F2=F2+1/(b-x(1)-x(2)+H1x(m))
         F1=F2*F1
         x3max=0.0
         if(F2.gt.0) x3max=b-x(1)-x(2)
```

[2] Und das, obwohl ich viele Dinge implementiert habe, die für zwei Aktien nicht notwendig gewesen wären. Hinzu kommt noch, daß die Genauigkeit beim neuen Modell höher ist, da dies bei vielen Aktien notwendig wird. Ich will nicht das Modell von Hellthaler kritisieren, für 1 und 2 Aktien war es völlig ausreichend, sondern nur den Aufwand, den ich für die Fertigstellung des Programms benötigt habe, begründen.

```
        if(F1.le.0) then
          xli=0
          xre=b-x(1)-x(2)
          x3max=xre*0.5
          do 110 j=1,10
             F0=0
             do 120 m=lo,kmax
 120            F0=F0+1/(x3max+H1x(m))
             if(F0.lt.0) then
               xre=x3max
             else
               xli=x3max
             endif
 110         x3max=(xli+xre)*0.5
        endif
        F=0
        do 130 m=lo,kmax
 130    F=F+alog(x(1)*(H(1,m)-r)+x(2)*(H(2,m)-r)+x3max*(H(3,m)-r)+rs)
        if(F.gt.Fmax) then
          Fmax=F
          Xmax(1,i)=x(1)
          Xmax(2,i)=x(2)
          Xmax(3,i)=x3max
        endif
        x(2)=x(2)+sc
        if(x(2).le.(b-x(1)))goto 140
        x(2)=0.0
        x(1)=x(1)+sc
        if(x(1).le.b) goto 140
 300    continue
```

Folgende Parameter lagen dem Vergleich bei *drei* Aktien zugrunde:

- Anzahl der simulierten Tage : 400
- Anzahl der Investoren : 10000
- Anzahl der Gruppen verschiedener Gedächtnislängen : 5 (10,26,141,197,256).

Folgende Zeiten benötigten die beiden Versionen (auf dem gleichen Rechner):

- Nach der *alten* Methode: **4357 Sekunden**
- Mit Hilfe der Gradientenmethode: **40 Sekunden**.

Rechenzeit gegen die Anzahl der Aktien:

mit folgenden Parametern:

- Anzahl der simulierten Tage : 2000
- Anzahl der Investoren : 1000

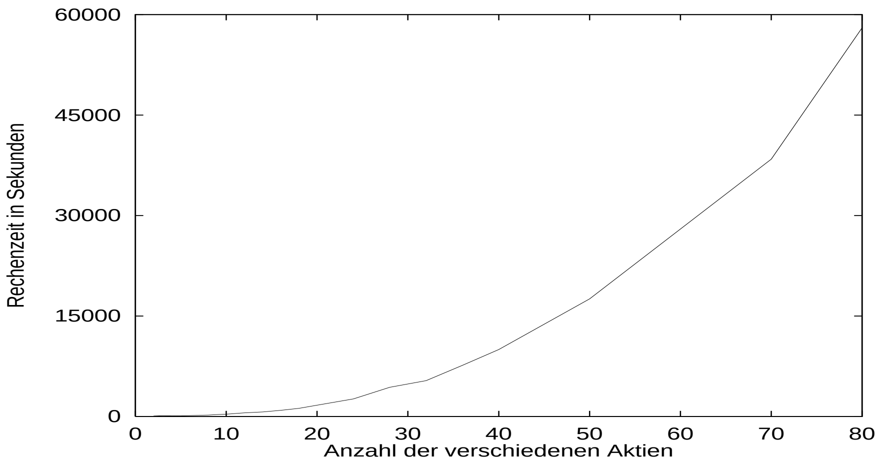

Abbildung 4.5: Rechenaufwandsteigerung bei Erhöhung der Aktienanzahl. Der Verlauf entspricht der Funktion $f(x) = x^{2.5}$

- Anzahl der Gruppen verschiedener Gedächtnislängen : 3 (10,133,256)

ist in Abbildung 4.5 die Anzahl der Aktien gegen die Zeit, die das Programm zur Simulation braucht, aufgetragen. Daher habe ich „nur“ maximal 40 Aktien simuliert. Bei 40 Aktien, 6 Gedächtnisgruppen, 10000 Investoren und 11000 Tagen hat eine Sun Ultra bereits 3 Tage reine Rechenzeit benötigt.

Rechenzeit gegen die Anzahl der Investoren

Erhöht man die Anzahl der beteiligten Investoren, so erhöht sich der zur Simulation notwendige Rechenaufwand linear, wie Abbildung 4.6 zeigt. Die Werte wurden mit folgenden Parametern bestimmt:

- Anzahl der simulierten Tage : 2000
- Anzahl der verschiedenen Aktien: 7
- Anzahl der Gruppen verschiedener Gedächtnislängen : 3 (10,133,256)

4.3 Mehrere Aktien

4.3.1 Eine Gedächtnislänge

Um die Auswirkungen bei der Erhöhung der Anzahl von verschiedenen Aktien zu untersuchen, habe ich für verschiedene Aktienzahlen Simulationen mit nur einer Gedächtnislängengruppe durchgeführt. Für eine Aktie erhält man bei Veränderung der Gedächtnislänge die in Abbildung 4.1, 4.7 und 4.8 dargestellten Kurse. Bei zwei Aktien erhält man unter gleichen Voraussetzungen strukturell gleiche Ergebnisse.
Bei zwei Aktien mit einer Gedächtnisgruppe von zehn Tagen lägen die Rechtecke

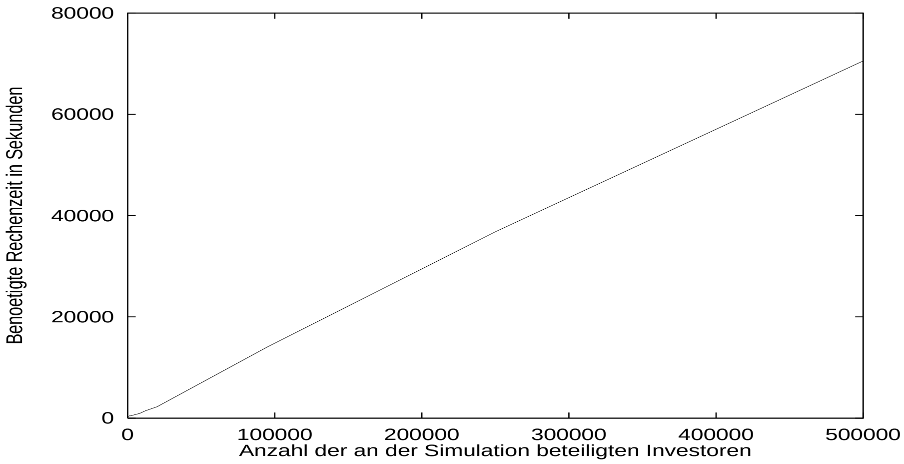

Abbildung 4.6: Rechenaufwandsteigerung bei Erhöhung der Investorenzahl

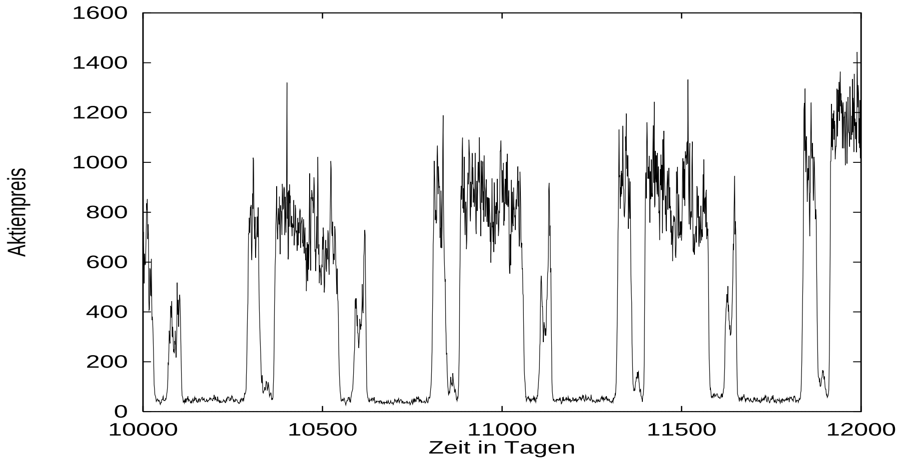

Abbildung 4.7: Aktienkurs mit: AnAk=1, NInv=100, NGr=1, k=256

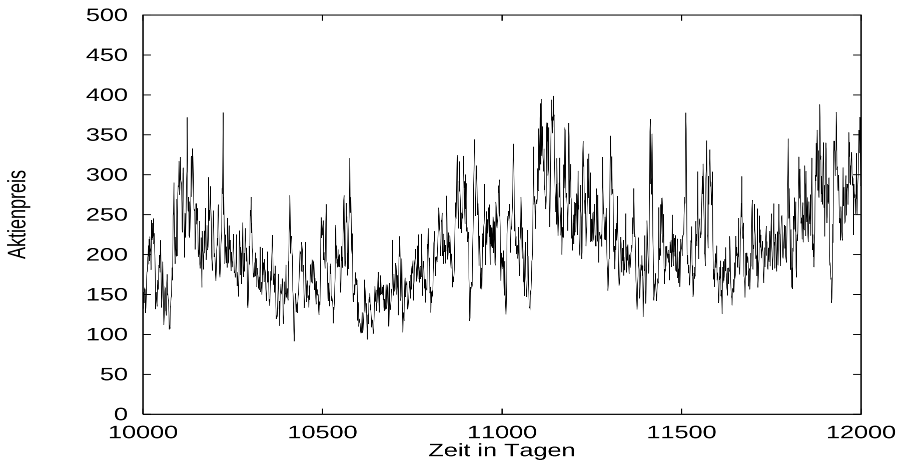

Abbildung 4.8: Aktienkurs mit: AnAk=1, NInv=100, NGr=1, k=500

der beiden einzelnen Aktien nebeneinander, wenn man sie in einem Plot darstellte. Die beiden Kurven haben die gleiche Periode. An einigen Stellen sind beide Aktienkurse auf niedrigem Niveau. So erhält man für den Durchschnittsaktienverlauf ebenfalls eine rechteckförmige Kurve, deren Rechtecke jedoch breiter als die der Einzelkurse sind (Abbildung 4.9). Das Verhalten bleibt bei vielen Aktien analog. Abbildung 4.10 zeigt den Kurs einer Aktie bei 7 verschiedenen Aktien im Markt. Betrachtet man sich die Kurse der restlichen 6 Aktien so sieht man, daß sie sich die Rechtecke, wie bei zwei Aktien beschrieben, nebeneinandersetzen. Das heißt, daß sich maximal eine Aktie auf hohem Niveau befindet. Nur an einzelnen Stellen sind alle Aktienkurse auf niedrigem Niveau. Der Durchschnittskurs der sieben Aktien (Abbildung 4.11) verdeutlicht dies.

Zusammenfassend läßt sich sagen, daß die Erhöhung der Anzahl von verschiedenen Aktien bei einer Gedächtnislänge mit $k = 10$ zu keinen strukturellen Veränderungen geführt hat, außer, daß sich die verschieden Aktien den Markt teilen.

Bei einer Gedächtnisgruppe mit 256 Tagen Erinnerungsvermögen stellt sich ebenfalls - bei mehreren verschiedenen Aktien - ein ähnlicher Verlauf wie bei einer Aktie ein.

Bei einer Gedächtnisgruppe mit 500 Tagen Erinnerungvermögen entstehen ebenfalls, wie bei einer Aktie, chaotische Kursverläufe. Der Bereich in dem der Kurs schwankt, wird jedoch deutlich höher (vgl. Abbildung 4.12). Ich habe, um die Entwicklung bei einer Gedächtnisgruppe zu verdeutlichen, deren Erinnerungsvermögen auf den sehr hohen Wert 1500 gesetzt. Dabei stellte sich heraus, daß bei verschiedenen getesteten $AnAk$ der Kurs starken, kurzzeitigen Schwankungen unterliegt, im Ganzen aber nicht mehr so stark schwankt. Abbildung 4.13 dokumentiert dieses Ergebnis.

Ich habe die verschiedenen Parameter immer an folgenden Aktienanzahlen getestet: $AnAk \in \{1, 2, 3, 4, 7, 10, 15, 20, 25, 30, 40\}$ um sichergehen zu können, daß die Aussage allgemeingültig ist.

Insgesamt lassen sich bei einer Gedächtnislänge keine strukturellen Veränderungen der Ergebnisse bei Erhöhung der Anzahl der unterschiedlichen Aktien

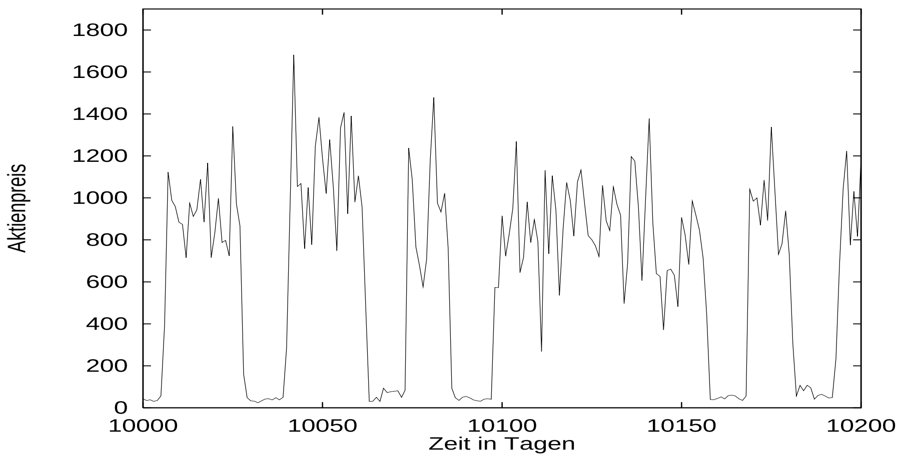

Abbildung 4.9: Durchschnitt der Kurse mit : AnAk=2, NInv=100, NGr=1, k=10

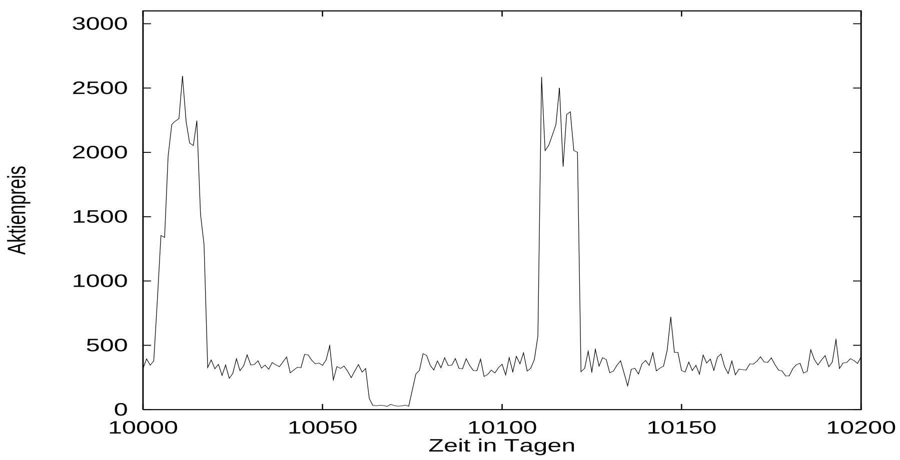

Abbildung 4.10: Kurs der 1. Aktie mit : AnAk=7, NInv=100, NGr=1, k=10

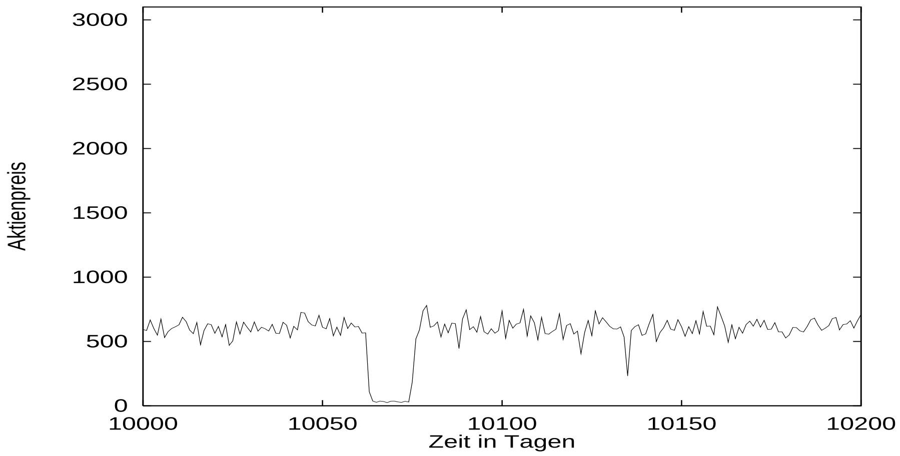

Abbildung 4.11: Durchschnittskurs mit: AnAk=7, NInv=100, NGr=1, k=10

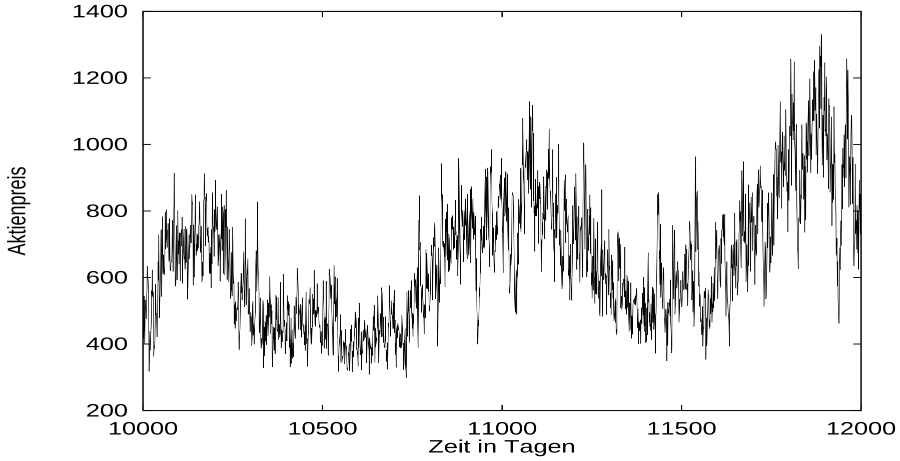

Abbildung 4.12: Kurs der 1. Aktie mit: AnAk=7, NInv=100, NGr=1, k=500

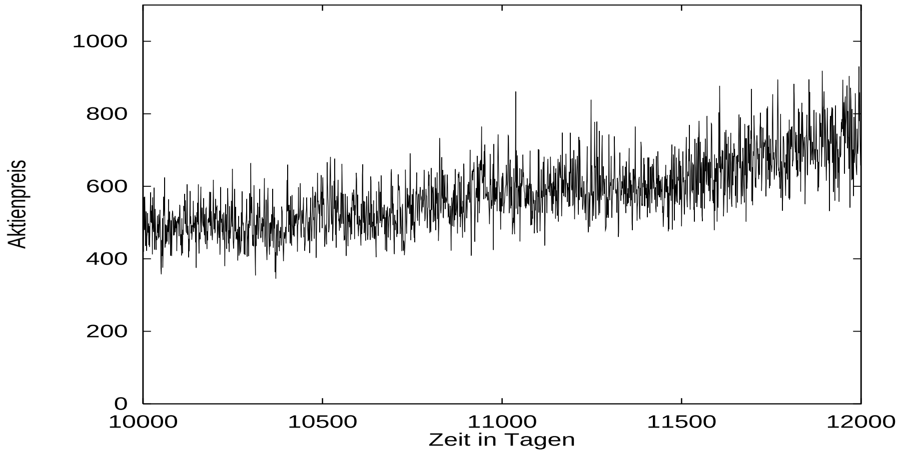

Abbildung 4.13: Kurs der 1. Aktie mit: AnAk=7, NInv=100, NGr=1, k=1500

erkennen.

4.3.2 Zwei verschiedene Gedächtnisgruppen

Bei zwei verschiedenen Gedächtnisgruppen stellt sich meist ein periodischer Kursverlauf ein.
Dies ist nicht der Fall, wenn die zweite Investorengruppe eine deutlich größere Gedächtnislänge, als die erste hat (vgl. Abbildung 4.14). Das liegt daran, daß die Gruppe mit der großen Gedächtnislänge sich von kurzfristigen Schwankungen nicht beeinflussen läßt und daher den Einfluß von der Gruppe mit der kleinen Gedächtnislänge auf Dauer zunichte macht. Unter diesen Umständen erhält man, wie in Abildung 4.13 dargestellt, chaotische Kursverläufe. Beispiele für periodische Kursverläufe zeigen Abbildung 4.15 und 4.16. Es tritt jedoch auch oft ein Verhalten auf, wie es bei nur einer Gedächtnisgruppe üblich ist. (vgl Abbildung 4.17). Dies kommt daher, daß über eine lange Zeit eine Gedächtnisgruppe den Markt dominiert. Für die Zeitspanne, die Abbildung 4.17 darstellt, dominiert die Gruppe mit dem 10-Tage -Gedächtnis. Betrachtet man den Kurs der gleichen Aktie unter den gleichen Bedingungen in einem Zeitraum, der nicht von der Gruppe mit dem 10-Tage-Gedächtnis dominiert wird, stellt sich ein periodischer Kursverlauf ein (Abbildung 4.18). Ich habe für zwei Gedächtnisgruppen 10 verschiedenen Kombinationen mit $AnAk \in 1, 2, 3, 4, 7, 15, 40$ simuliert.

4.3.3 3 Gedächtnisgruppen

Kleine Investorenzahlen

Bei sehr kleinen Investorenzahlen erhält man, je nach Wahl der Gedächtnislängen, meistens chaotische Kursverläufe. Als Beispiel hierfür betrachte man Abbildung 4.20. Die Kurse der anderen 6 Aktien verlaufen strukturell gleich. Der Durchschnittskurs weist ebenfalls ein chaotisches Kursverhalten auf.
Untersucht wurden die Kurse für 30, 50 und 70 Investoren bei ein bis sie-

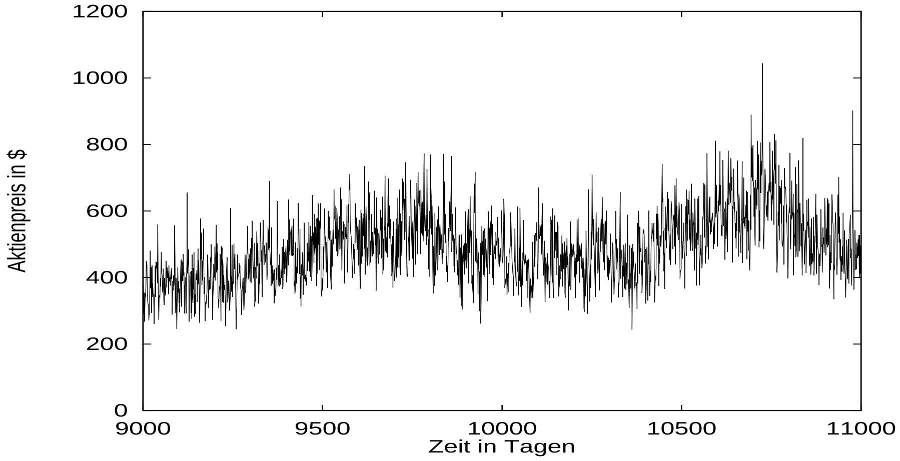

Abbildung 4.14: Kurs der 1. Aktie mit: AnAk=7, NInv=100, NGr=2, k=/10,400/

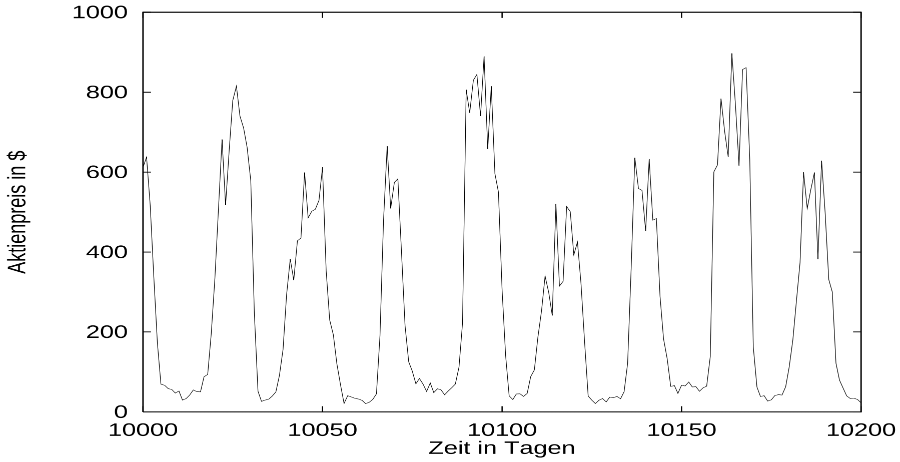

Abbildung 4.15: Kurs der Aktie mit: AnAk=1, NInv=100, NGr=2, k=/10,32/

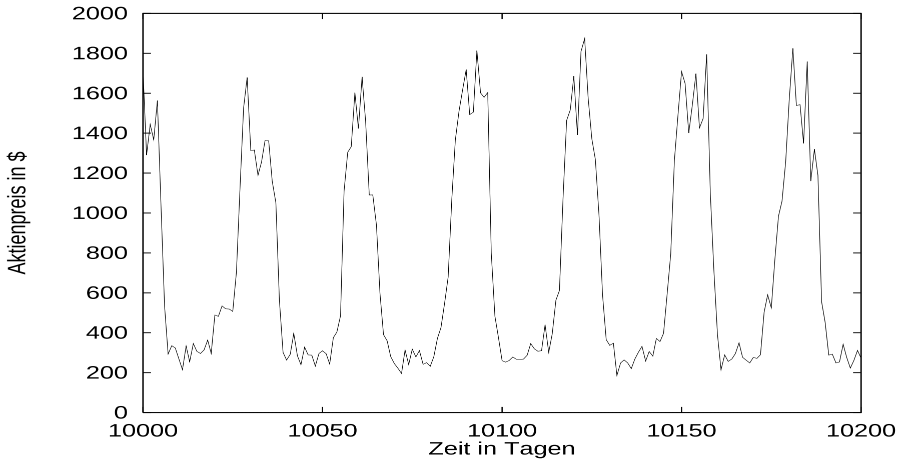

Abbildung 4.16: Kurs der 1. Aktie mit: AnAk=4, NInv=100, NGr=2, k=/10,47/

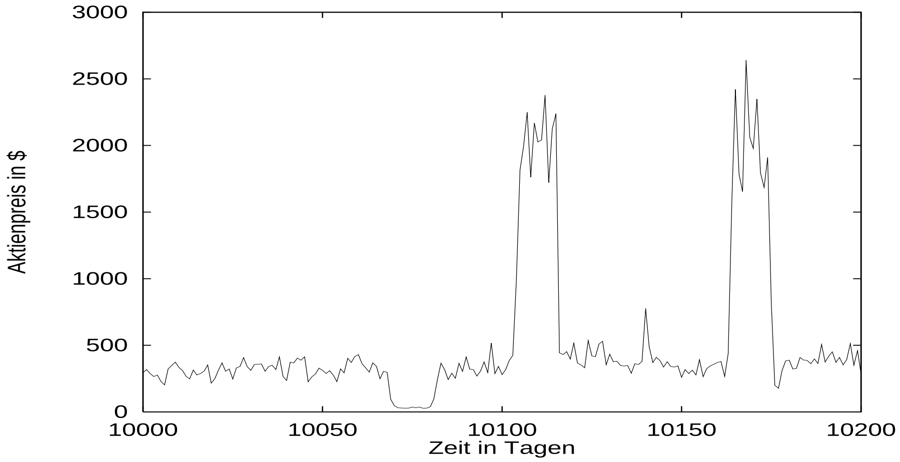

Abbildung 4.17: Kurs der 1. Aktie mit: AnAk=7, NInv=100, NGr=2, k=/10,47/

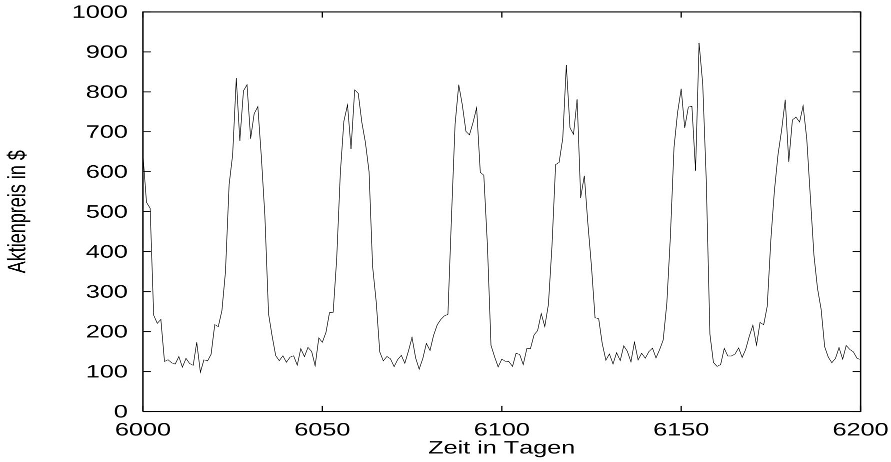

Abbildung 4.18: Kurs der 1. Aktie mit, frühe Phase: AnAk=7, NInv=100, NGr=2, k=/10,47/

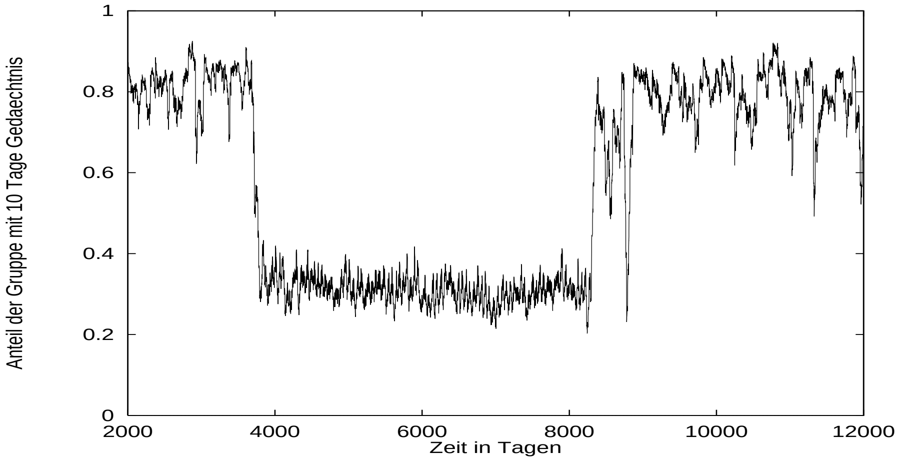

Abbildung 4.19: Anteil der Gruppen am Gesamtvermögen: AnAk=7, NInv=100, NGr=2, k=/10,47/

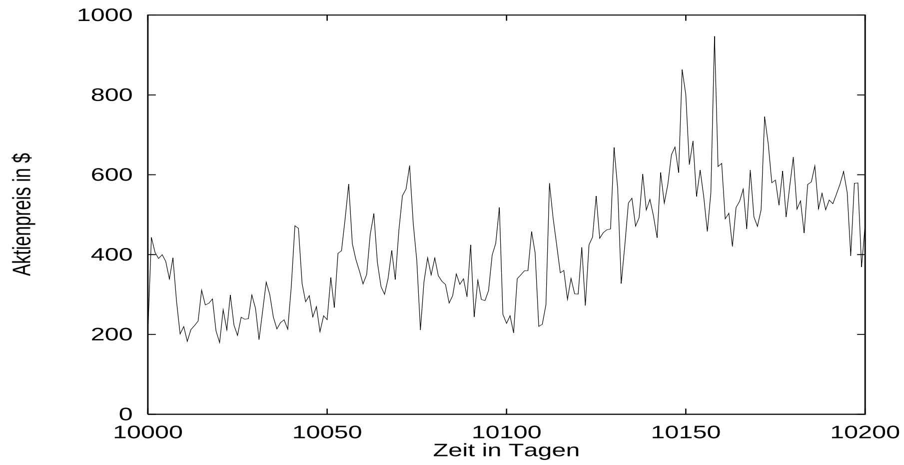

Abbildung 4.20: Aktienkurs bei 7 Aktien: AnAk=7, NInv=49, NGr=3, k=/10,133,256/

ben, fünfzehn, zwanzig und dreißig verschiedenen Aktien für 12 verschiedene Gedächtnisgruppen [3] und immer chaotische Kursverläufe wie in Abbildung 4.20 gefunden. Bei einigen Beispielen gab es jedoch kurze Ausreißer, wie in Abbildung 4.21 zu erkennen ist. Durch eine Veränderung der einzelnen Gedächtnislängen kann man diese Ausreißer jedoch immer vermeiden. Stichprobenhaft habe ich auch andere Kombinationen ausprobiert, bin aber auf keine Besonderheiten gestoßen.
Mit drei verschiedenen Gedächtnislängen und kleinen Investorenzahlen ergeben sich also auch bei mehreren Aktien chaotische Simulationsergebnisse.

Größere Investorenzahlen

Um die Entwicklung der Aktienkurse, die das Computermodell errechnet, bei wachsender Investorenzahl zu untersuchen, habe ich viele verschiedene Simulationen bei 99 Investoren durchgeführt. Insgesamt läßt sich feststellen, daß chaotische Kursverläufe seltener vorkommen. Oft erhält man chaotische Kursverläufe mit Ausreißern, wie in Abbildung 4.22 zu erkennen ist. Die anderen 29 Aktienkurse verlaufen analog. Der Durchschnittskurs über alle 30 Aktien ist ebenfalls chaotisch. Dieses Verhalten ist bei 3 verschiedenen Gedächtnisgruppen und 99 Investoren typisch ab einer größeren Anzahl von Aktien. Betrachtet man die Abbildungen 4.23, 4.24 und 4.22 so erkennt man, daß die „Zacken" mit steigender Anzahl von Aktien immer seltener werden. Das liegt daran, daß die verschiedenen Aktien sich den Markt teilen, und die einzelne Aktie so seltener solche „Zacken" aufweisen.

Erhöht man die Investorenzahl um ein Vielfaches (NInv=10000), erhält man beim Einaktienmodell periodische Kursverläufe. Bei mehr Aktien und drei Gedächtnisgruppen werden 3 der Aktienkurse periodisch, der Rest verläuft chaotisch. Die

[3] Unter anderem die bei Hellthaler untersuchten Gedächtnisgruppen (10,133,256) und (10.40.130).

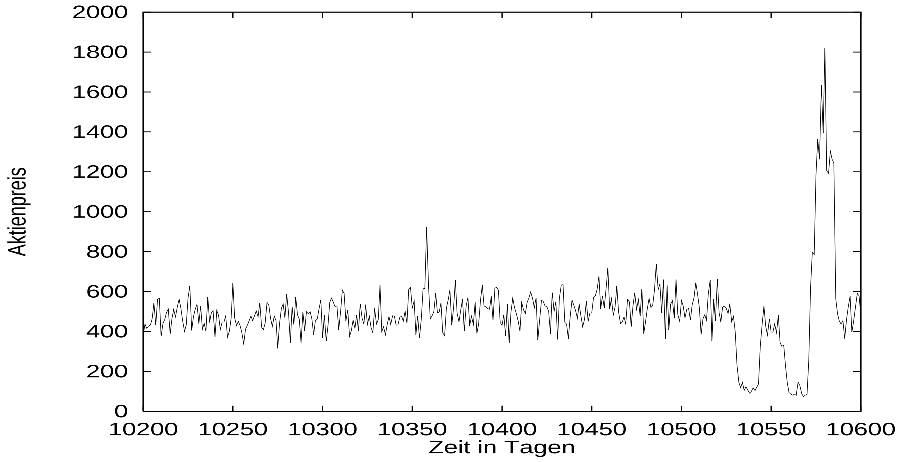

Abbildung 4.21: Aktienkurs bei 15 Aktien mit Ausreißer: AnAk=15, NInv=49, NGr=3, k=/10,133,256/

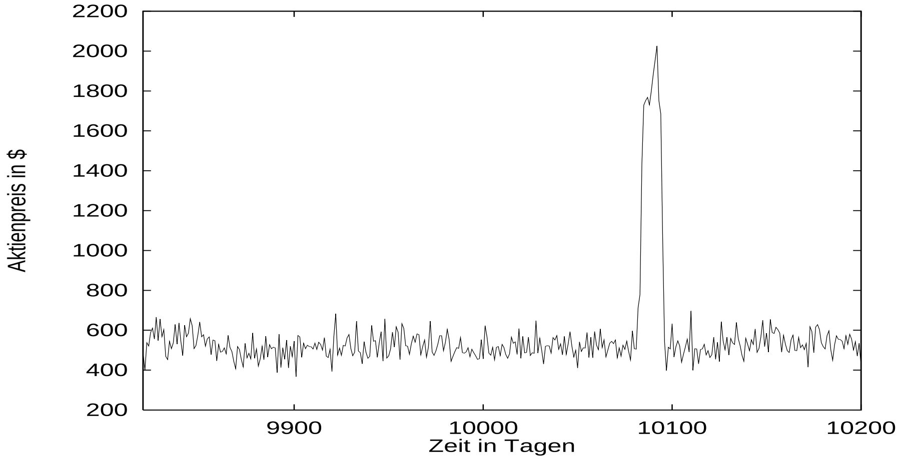

Abbildung 4.22: Aktienkurs bei 30 Aktien mit Ausreißer: AnAk=30, NInv=99, NGr=3, k=/10,133,256/

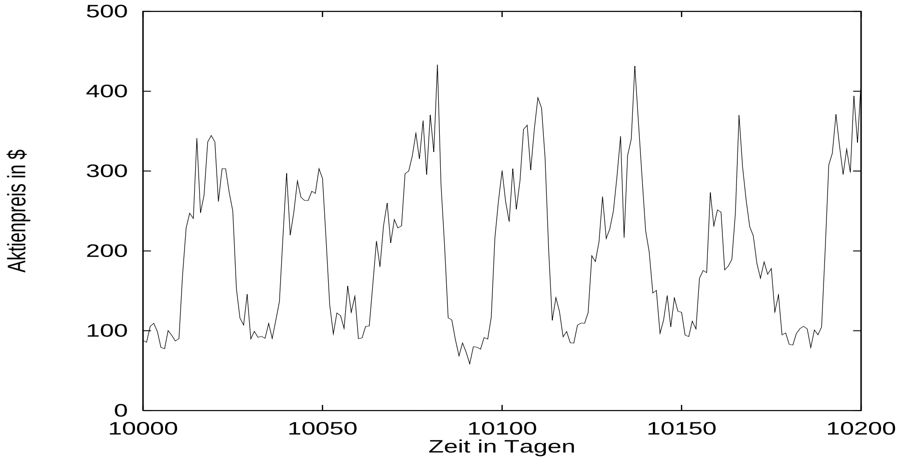

Abbildung 4.23: AnAk=1, NInv=99, NGr=3, k=/10,133,256/

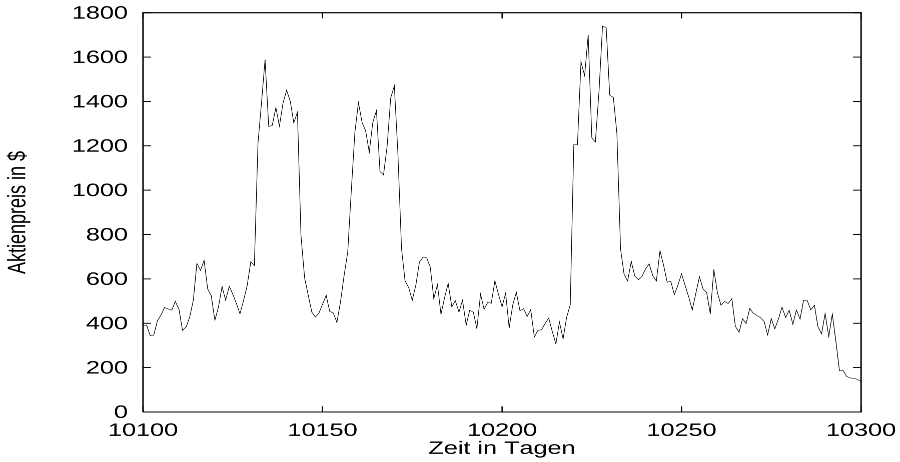

Abbildung 4.24: Aktienkurs bei 7 Aktien mit Ausreißer: AnAk=7, NInv=99, NGr=3, k=/10,133,256/

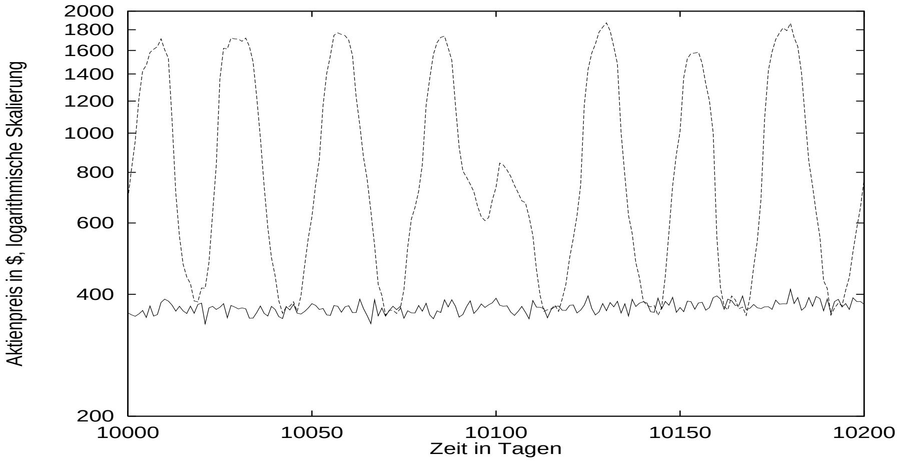

Abbildung 4.25: 2 von 7 Aktienkursen: AnAk=7, NInv=9999, NGr=3, k=/10,133,256/

chaotischen Kurse liegen jedoch unter dem Niveau der periodischen Verläufe. [4] Ein Beispiel zeigt Abbildung 4.25. Man erkennt in der logarithmischen Auftragung deutlich die Dominanz des periodisch verlaufenden Aktienkurses. Dies ist ein sehr typischer Kursverlauf, der bei hohen Investorenzahlen und einer Anzahl von verschiedenen Aktien, die größer als drei ist, auftritt.
Bei drei Gedächtnisgruppen habe ich für eins bis sieben, 15, 20, 30 und 40 Aktien bei 99 und 9999 Investoren 12 verschiedene Gedächtnisgruppenkombinationen untersucht. Für sieben Aktien habe ich noch 20 weitere Kombinationen simuliert und bin nie auf abweichende Ergebnisse gestoßen.

4.3.4 Sechs Gedächtnisgruppen

Kleine Investorenzahlen

Für drei bis sieben, 15, 20 und 30 verschiedene Aktien habe ich für 30, 60 und 80 die Gedächtnisgruppen (13, 141, 25, 453), (89, 145, 233, 321, 432), (30, 54, 78, 94, 102, 140, 195, 254, 290, 330) und (53, 98, 144, 209, 321, 453, 601) untersucht. Für 60 Investoren habe ich bei $AnAk$ = 7zusätzlich 15 weitere Gedächtnisgruppenkombinationen untersucht. Bei den Investorenzahlen von 30 und 60 haben sich immer chaotische Kursverläufe für die einzelnen Kurse und den Durchschnittskurs eingestellt, wie in Abbildung 4.26 zu sehen ist. Bei 80 Investoren hängt es von der Gedächtnisgruppenkombination ab, ob ein chaotischer Kursverlauf entsteht. Oft tritt auch ein Aktienkurs mit Ausreißern auf, wie er auch bei 3 Gedächtnisgruppen entstand. Als Beispiel soll Abbildung 4.27 dienen. Diese Ausreißer werden mit steigendem AnAk schmaler und seltener.

[4] vergleiche Hellthaler

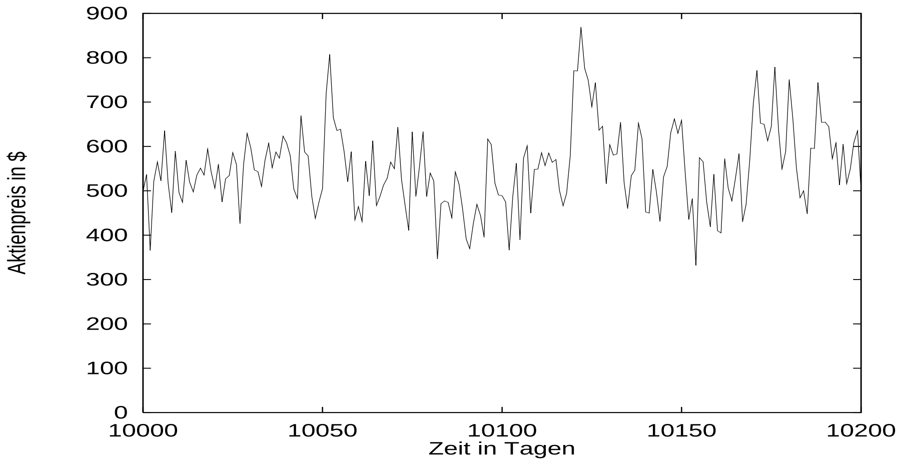

Abbildung 4.26: AnAk=15, NInv=56, NGr=7, k=/53,98,144,209,321,453,601/

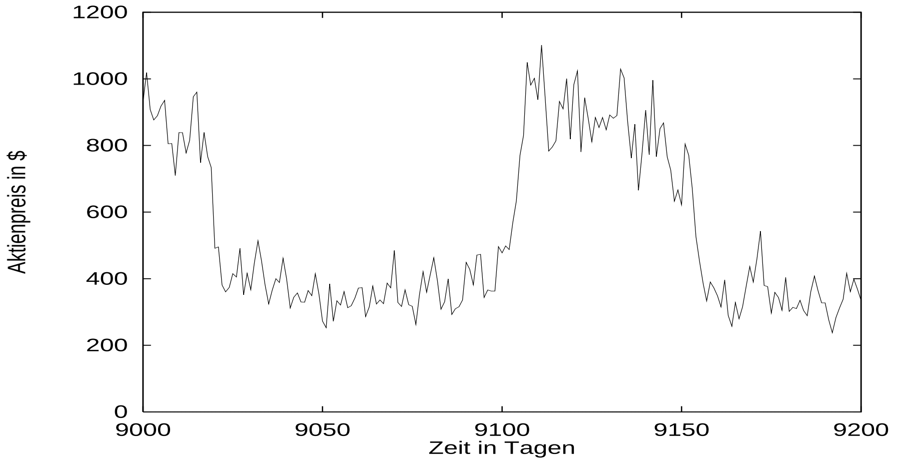

Abbildung 4.27: AnAk=7, NInv=77, NGr=7, k=/53,98,144,209,321,453,601/

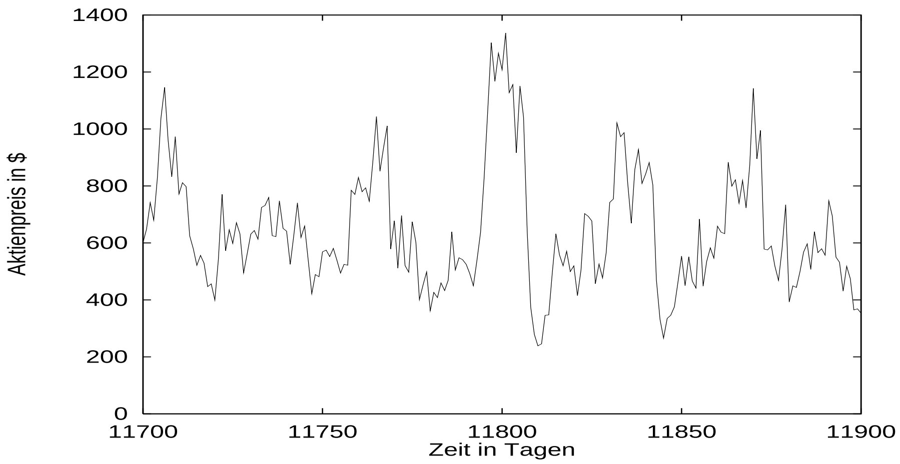

Abbildung 4.28: AnAk=4, NInv=96, NGr=6, k=/10,36,141,193,256,420/

Viele Investoren

Ich habe für 100 Investoren sehr viele Simulationen mit verschiedenen Gedächtnisgruppen simuliert. Sehr oft enstanden Aktienkurse wie in Abbildung 4.27. Chaotische Kursverläufe traten selten auf. Oft entstehen auch Kurse, die weder ganz periodisch noch völlig chaotisch verlaufen (vgl. Abbildung 4.28). Erhöht man die Investorenzahl in diesen Beispielen, so enstehen für diesen Aktienkurs periodische Kursverläufe. Man erkennt also deutlich, daß die chaotischen Kursverläufe bei wenigen Investoren durch Erhöhung der Investorenzahl stark beeinflußt werden.
Ich habe die Investorenzahl nun auf 10000 erhöht und die Auswirkungen auf die Aktienkurse untersucht. Wie schon bei 3 verschiedenen Gedächtnisgruppen erläutert, stellt sich ein typischer Kursverlauf ein. Drei der Aktienkurse verlaufen periodisch, der Rest und der Durchschnitt chaotisch. Für die Simulation aus der Abbildung 4.29 entstanden ist, brauchte eine *Sun Ultra* etwa 70 Stunden reine Rechenzeit. Ich dokumentiere das hier beschriebene typische Ergebnis noch einmal anhand von 3 verschiedenen Aktien.
Bei drei Aktien erhält man für 9996 Aktionäre periodische Kursverläufe, wie Abbildung 4.30 zeigt. In Abbildung 4.31 sind die Kurse der drei Aktien in einem Plot dargestellt. Da es sich um einen Schwarz-Weiß-Ausdruck handelt, ist es schwierig, die verschiedenen Kurse zu unterscheiden. Die Abbildung soll deutlich machen, daß es sich um drei periodische Kurven mit einer Phasenverschiebung von einer drittel Periode handelt. Betrachtet man nun den Durchschnittskursverlauf (Abbildung 4.32), so erkennt man entsprechend kein periodisches Verhalten mehr. Vielmehr schwankt der Durchschnittskurs nur leicht. Bei mehr als drei Aktien verlaufen 3 Aktienkurse wie gerade dargestellt, die anderen Aktienkurse verlaufen chaotisch und unter dem Niveau der periodischen Aktienkurse (vgl. Abbildung 4.29). Ich habe bei 10000 Investoren 20 verschieden Gruppenkombinationen bei verschiedenen Anzahlen der Aktien ausprobiert, von 2 bis zu 10 verschiedenen Gedächtnisgruppen. Für sieben Aktien] habe ich 100 weitere Kombinationen untersucht und keine abweichenden Ergebnisse gefunden. Daher

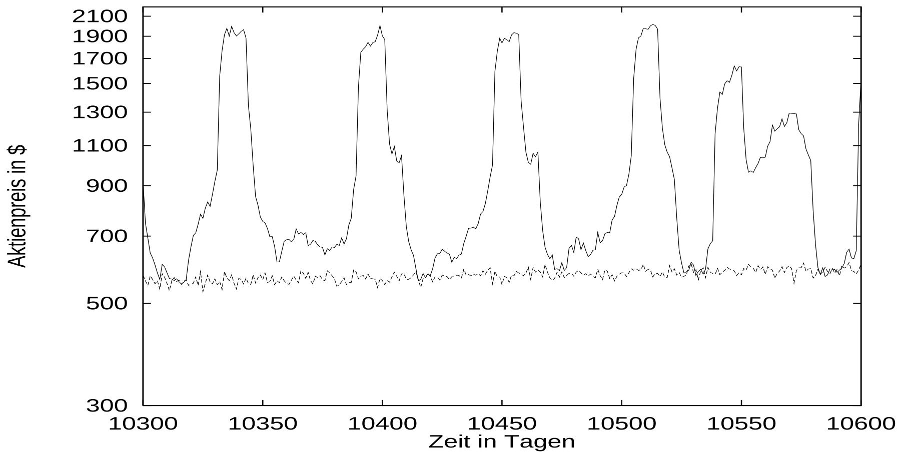

Abbildung 4.29: 2 von 40 Aktienkursen, mit NInv=9996, NGr=6, k=/10,36,141,193,256,420/

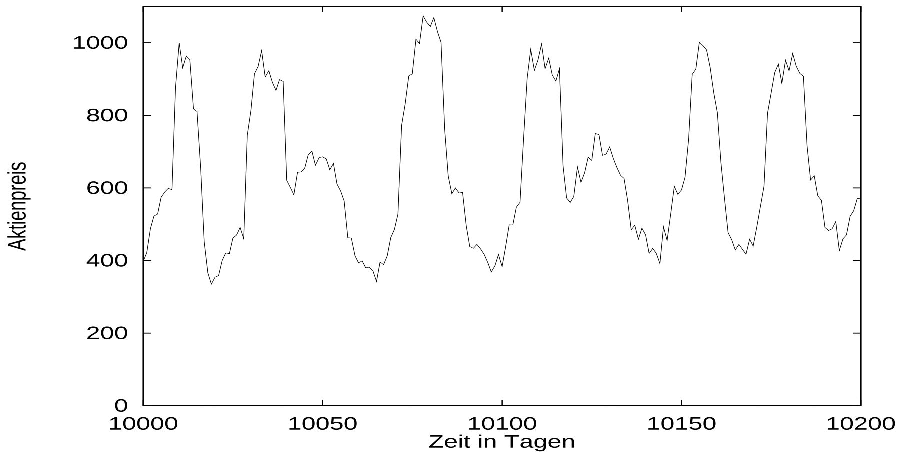

Abbildung 4.30: Kurs von Aktie 1 mit: AnAk=3, NInv=9996, NGr=6, k=/10,36,141,193,256,420/

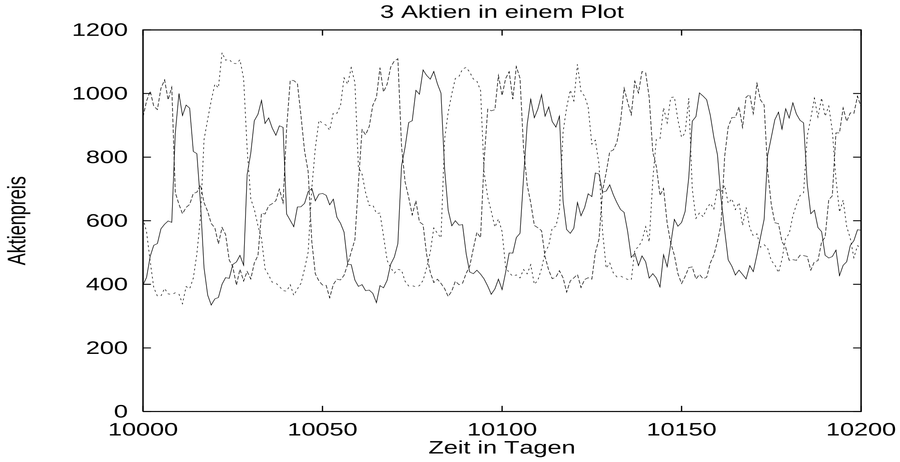

Abbildung 4.31: Kurs von Aktie 1,2 und 3 mit: AnAk=3, NInv=9996, NGr=6, k=/10,36,141,193,256,420/

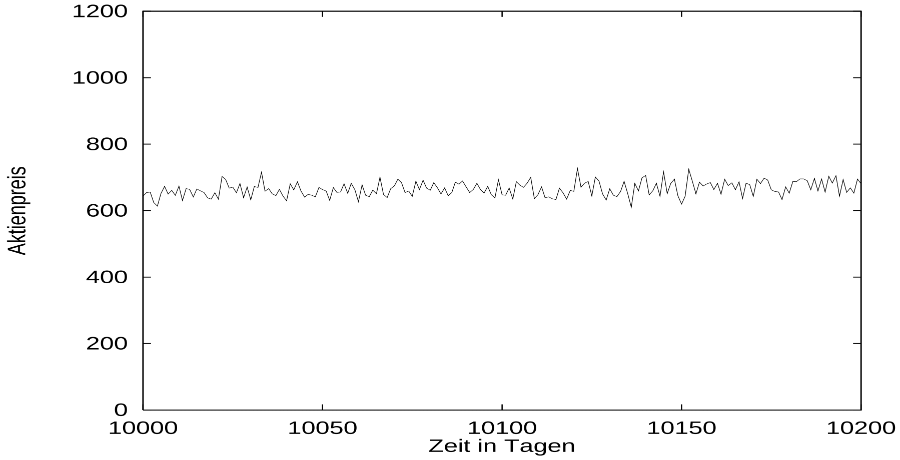

Abbildung 4.32: Durchschnitt der Kurse mit : AnAk=3, NInv=9996, NGr=6, k=/10,36,141,193,256,420/

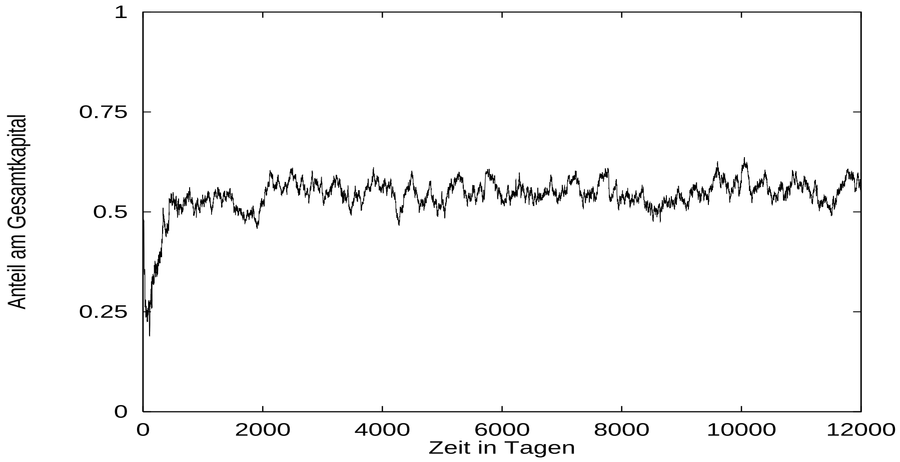

Abbildung 4.33: Anteil der 3.Gruppe am Gesamtkapital mit AnAk=7, NInv=49, NGr=3, k=/10,133,256/

gehe ich davon aus, daß dies ein sehr typischer Kursverlauf ist.

4.3.5 Kapitalkonzentration

Das Programm kann neben den Aktienkursen die Verteilung der Kapitalkonzentration ausgeben. Die Feldvariable $anw(j)$ enthält den Anteil des Gesamtkapitals, den die Mitglieder der Gedächtnisgruppe j halten. Ich habe für die oben beschriebenen verschiedenen Typen von Aktienkursen eine Analyse der Kapitalkonzentrationen durchgeführt.
Vor allem habe ich mich dafür interessiert, wie sich die Kapitalkonzentration verändert, wenn sich die Anzahl der Aktionäre vergrößert, insbesondere mit den Gedächtnislängenkombinationen, bei denen die chaotischen Aktienkurse durch die Erhöhung periodisch werden. Bei 7 verschiedenen Aktien und den drei Gedächtnisgruppen (10, 133, 256) [5] ist ein typisches Verhalten der Kapitalkonzentration bei Erhöhung der Anzahl der Aktionäre festzustellen. Abbildung 4.33 zeigt den Anteil am Gesamtkapital der Gruppe mit 256 Tagen Gedächtnis. Der Anteil hält sich mit Schwankungen auf einem konstanten Niveau. Der Anteil der Gruppe mit 133 Tagen Erinnerungsvermögen verläuft ebenfalls auf einem konstanten Niveau, welches etwas tiefer liegt (Durchschnitt ca. 0.3). Der Anteil der kleinsten Gruppe verläuft nahe 0.1.

Vergrößert man nun die Anzahl der Aktionäre auf 98, so schwanken die einzelnen Anteile stärker. Die Gruppe mit dem längsten Gedächtnis dominiert mit einem Durchschnitt von ca. 0.5, gefolgt von der Gruppe mit dem 10-Tage-Gedächtnis bei etwa 0.3 und der mittleren Gruppe mit einem Schnitt von etwa 0.2. Der zugehörige Kurs der Aktie ist chaotisch mit einzelnen Ausreißern. Interessant wird es, wenn man die Anzahl der Investoren auf 9996 erhöht. Dort dominiert die Gruppe mit der mittleren Gedächtnislänge klar den Markt mit

[5] vergleiche Abbildung 4.20, 4.24 und 4.25

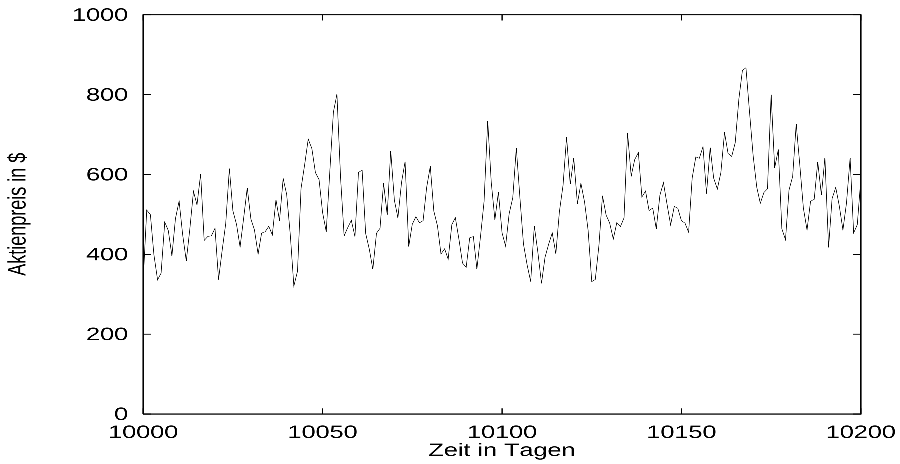

Abbildung 4.34: Kurs einer Aktie mit AnAk=7, NInv=49, NGr=3, k=/10,133,256/

einem Durchschnitt von etwa 0.9. Der Anteil der Gruppe mit dem längsten Gedächtnis geht gegen Null. Die Gruppe mit der kleinsten Gedächtnislänge hält einen Anteil von ca. 0.1.
Wesentlich und typisch ist, daß sich bei chaotischen Kursverläufen viele, einzelne Gruppen den Markt teilen. Erhöht man die Investorenzahl, so wird **eine** Gruppe dominant.
Bei einer Aktie und den 6 Gedächtnisgruppen (10, 36, 141, 193, 256, 420) geht nur der Anteil der Gruppe mit dem 36 Tage Gedächtnis gegen Null. Die Gruppe mit dem längsten Gedächtnis hält hier die stärksten Anteile. Erhöht man die Anzahl der Aktionäre auf 996, so dominieren nur noch die beiden größten Gedächtnisgruppen den Markt, die Anteile der anderen laufen gegen Null. Bei 99996 Investoren dominiert die Gruppe mit dem längsten Gedächtnis mit einem durchschnittlichen Anteil von 0.98! Als weiteres Beispiel möchte ich eine Simulation für 7 Aktien anführen. Die Gedächtnisgruppenkombination ist: (53,98,144,209,321,453,601). Ich möchte die Ergebnisse für 56, 98 und 9996 Investoren darstellen. Bei 56 Aktionären erhält man einen chaotischen Verlauf des Aktienkurses für alle 7 verschiedenen Aktien (vgl. Abbildung 4.34). Betrachtet man die Kapitalkonzentration bei diesen Parametern, so stellt man fest, daß kein Anteil einer Gedächtnisgruppe gegen Null läuft. Vielmehr schwanken die Anteile der verschiedenen Gedächtnisgruppen um folgende Niveaus:
53 um 0.025, 98 um 0.075, 144 um 0.11, 209 um 0.15, 321 um 0.175, 453 um 0.22, 601 um 0.23. Bei chaotischem Kursverhalten sind wieder alle Gedächtnisgruppen „aktiv“.
Bei 98 Aktionären erhält man ebenfalls einen chaotischen Aktienkurs, allerdings mit einigen Ausreißern. Die Analyse der Kapitalkonzentration zeigt, daß der Anteil am Gesamtkapital der Gruppe mit dem 98 Tage Gedächtnis gegen Null geht. Bei 9996 Aktionären erhält man für 3 Aktien periodische, für die restlichen 4 Aktien chaotische Aktienkurse, die jedoch unter dem Niveau der periodischen liegen. Bei der Kapitalkonzentration erkennt man nach 5000 Tagen Simulationszeit, daß die Gruppe mit dem 601 Tage Gedächtnis den Markt völlig dominiert.

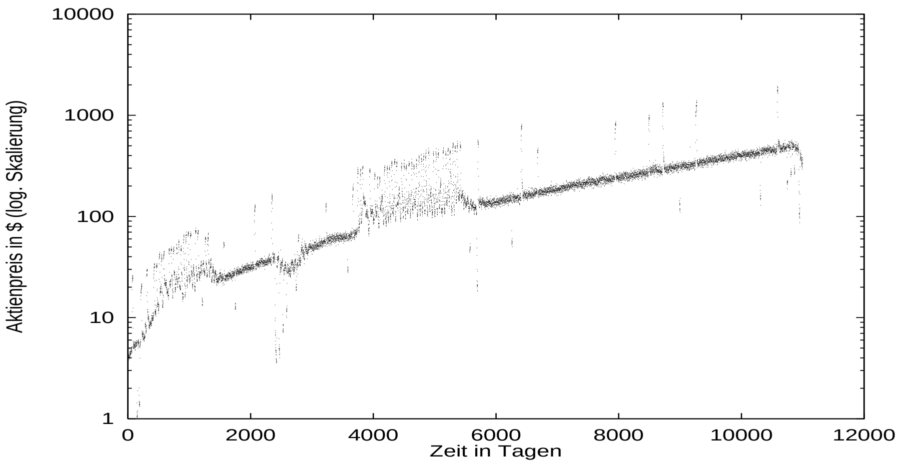

Abbildung 4.35: Kurs einer Aktie mit AnAk=7, NInv=996, NGr=6, k=/10,133,256/

Der Anteil der anderen Gruppen geht gegen Null.
Zusammenfassend läßt sich sagen, daß sich die Untersuchung der Kapitalkonzentration als geeignetes Instrument zur Untersuchung des Modells eignet. Es ist sicher nicht realistisch und entspricht nicht der Idee einen heterogenen Markt zu simulieren, wenn eine Gedächtnisgruppe den Markt völlig dominiert. Realistisch ist eine relativ gleichmässige Verteilung des Vermögens auf die einzelnen Gedächtnisgruppen.

4.3.6 Änderung der Dividenden

Ich habe untersucht, welchen Einfluß der Faktor d_i um den die Dividende jeden Tag erhöht wird, auf die Simulationsergebnisse hat. Ich habe insbesondere drei Arten von Änderungen von d_i durchgeführt.

- Jede Aktie erhält ein anderes d_i. Die Größe der Unterschiede habe ich variiert.
- Die Aktien werden in verschiedene Gruppen mit gleichen Werten für d_i eingeteilt.
- Alle Aktien bis auf eine haben das gleiche d_i.

Der Unterschied, welcher sich zu den gewohnten Ergebnissen einstellt ist, daß bei vielen Investoren und vielen verschiedenen Aktien, nicht die gleichen Aktien über die gesamte Simulationszeit periodisch und dominant sind, sondern nur noch über längere Zeitabschnitte. [6] In Abbildung 4.35 erkennt man zwei Phasen, in denen der Kurs dieser Aktie periodisches Verhalten aufweist.
Daraufhin habe ich untersucht, inwieweit die Dividende sich auf die Utilityfunktion auswirkt und herausgefunden, daß sie durch die extremen Schwankungen der periodischen Kurse eine untergeordnete Rolle spielt. Wählt man jedoch ein

[6] Abweichendes Verhalten tritt nur bei genügend grossen Unterschieden der d_i's auf.

oder mehrere d_i im Vergleich zu den anderen sehr hoch, so kann wie in Abbildung 4.35 eine Aktie, die chaotisch auf niedrigem Niveau liegt, bei passenden Zufallszahlen des Rauschens für die Investoren attraktiv werden.

4.4 Vermögensstruktur

Ich habe die Investoren an jedem Tag der Simulation in 5 Gruppen, sortiert nach Vermögen, aufgeteilt. In Gruppe 1 sind die Investoren mit dem wenigsten Geld, in Gruppe 6 die mit dem meisten - . Die 6. Gruppe enthält, falls die Anzahl der Investoren nicht durch 5 teilbar ist den „Rest“, der reichsten Investoren.
Sortiert habe ich die Investoren mit dem sogenannten Bubblesort - Algorithmus. Dabei läuft man durch das gesamte Feld und betrachtet immer Paare von Nachbarn. Wenn Investor $j+1$ weniger Geld haben sollte, als Investor j, tauschen die beiden die Plätze. Wenn man dieses Verfahren auf alle Investoren angewandt hat, läuft man das Feld erneut durch. Der Sortiervorgang ist beendet, wenn kein Tausch mehr vorgenommen wird. Der zugehörige Programmteil sieht wie folgt aus:

```
      if(itime.lt.2) then
         do 901 j=1,NInv
901         ifeld(j)=j
      endif
      do 902 jj=1,10000000
         fertig=1
         do 903 ii=1,NInv-1
            if(w(ifeld(ii)).gt.w(ifeld(ii+1))) then
              ihilf=ifeld(ii)
              ifeld(ii)=ifeld(ii+1)
              ifeld(ii+1)=ihilf
              fertig=0
            endif
903      continue
      if(fertig.eq.1) goto 905
902   continue
905   NAnf=1
      do 908 mm=1,6
908      xgrupp(mm)=0.0
      ipro=NInv/5
      do 906 mm=1,5
         do 907 mmm=NAnf,NAnf+ipro-1
907         xgrupp(mm)=xgrupp(mm)+w(ifeld(mmm))
906      NAnf=NAnf+ipro
      do 909 mm=5*ipro,NInv
909      xgrupp(6)=xgrupp(6)+w(ifeld(mm))
      do 911 mm=1,6
911      xgrupp(mm)=xgrupp(mm)*wges
```

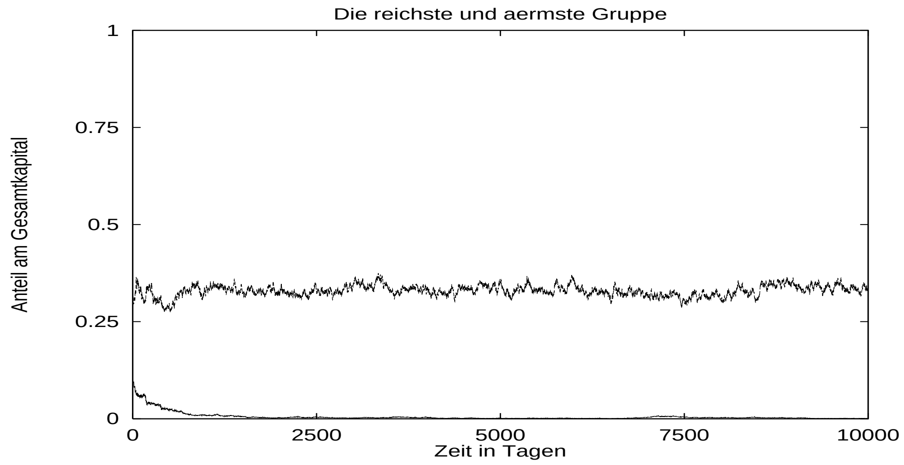

Abbildung 4.36: AnAk=7, NInv=48, NGr=6, k=10,36,141,193,256,420

Die Wechsel der Investoren zwischen den einzelnen Gruppen sind nicht untersucht worden. Interessant ist, daß nach einer gewissen Zeit die Niveaus der einzelnen Gruppen konstant bleiben. In Abbildung 4.36 ist der zeitliche Verlauf des Anteils am Gesamtvermögen der ärmsten und der reichsten Gruppe dargestellt. Aus Interesse habe ich weiter untersucht, ob sich der Anteil der reichsten Gruppe am Gesamtvermögen verändert, wenn die Investorenzahl erhöht wird. Folgende Tabelle zeigt, daß bei steigender Zahl an im Markt befindlichen Investoren die Gruppe mit den reichsten Investoren den Markt immer mehr beherrscht. Die Parameter der Simulation entsprechen (bis auf die Variable $NInv$) denen in Abbildung 4.36.

Anzahl der Investoren	Anteil der reichsten 20 % am Gesamtkapital in %
50	40
300	88
1000	89
5000	95
10000	95
30000	97

Die Ergebnisse aus diesem Abschnitt entsprechen denen die aus der Untersuchung der Kapitalkonzentration hervorgegangen sind. Man kann also auch über die Untersuchung der Vermögensstruktur Aussagen treffen, ob die gefundenen Ergebnisse realistisch sind oder nicht.

Kapitel 5

Änderungen des Grundprogramms

5.1 Änderung der Ultilityfunktion

Didier Sornette [1] hatte die Idee, die Utilityfunktion, wie folgt zu verändern: anstatt $EU(\dots) = \dots \sum_{\dots}^{\dots} \ln(\dots)$ (vgl. 2.2 schlug er vor $EU(\dots) = \dots \sum_{\dots}^{\dots} (\dots)^z$ mit $z \in [0,1]$ zu wählen. Dies habe ich für das Einaktienmodell wie folgt umgesetzt:

$$EU_j(x) \quad = \quad \frac{1}{k_j} \sum_{s=t}^{t-k_j+1} \left((1-x) \cdot K_j^* \cdot (1+r) + x \cdot K_j^* \cdot (1+H_s)\right)^z .$$

Diese Gleichung läßt sich umformen zu:

$$EU_j(x) \quad = \quad \frac{1}{k_j} \cdot (K_j^*)^z \cdot \sum_{s=t}^{t-k_j+1} ((x(H_s - r)) + (1+r))^z .$$

Man erhält:

$$F_j(x) \quad = \quad \sum_{s=t}^{t-k_j+1} ((1+r) + x(H_s - r))^z , \tag{5.1}$$

dessen Ableitung von nach x lautet:

$$\frac{\partial F_j(\dots)}{\partial x} \quad = \quad \sum_{s=t}^{t-k_j+1} \frac{z \cdot (H_s - r)}{((1+r) + x(H_s - r))^{1-z}}.$$

Die zweite Ableitung von F_j nach x ist dann:

$$\frac{\partial^2 F_j(\dots)}{\partial x^2} \quad = \quad \sum_{s=t}^{t-k_j+1} \frac{z \cdot (z-1) \cdot (H_s - r)^2}{((1+r) + x(H_s - r))^{2-z}}.$$

Wichtig ist, daß für z=1 die Krümmung von $F_j(\dots)$ verschwindet. Die Steigung von $F_j(\dots)$ ist somit konstant. Daher liegt der optimale Aktienanteil für $z = 1$ entweder bei 0 oder b

[1] Mündlicher Vorschlag an Herrn Stauffer.

5.1.1 Änderungen am Programm

Ich habe das von Hellthaler [2] geschriebene Programm so geändert, daß es der neuen Gleichung 5.1 entspricht. Dafür mußte nur der Teil umgeschrieben werden, in dem der optimale Aktienanteil berechnet wurde. Das geänderte Programm sieht wie folgt aus [3].

```
  program eineAktie
      parameter (Hbeg=0.000100025,Hdelta=0.000125,xdelta=0.2,
     1     NGr=6, Nxx=100000, kmax=420, maxz=11000,
     2     r=0.0001, di=1+0.00015, b=0.99, nzuf=2048, iter=20,
     3     NpG=Nxx/NGr, NInv=NpG*NGr, NA=100*NInv, rs=1+r,z=1.0)  XXXXXX
           dimension k(NGr), Xm(NGr), AnW(NGr), H(kmax),
     1     zx(0:nzuf-1), NAkt(NInv), W(NInv), Xreal(NInv)
      data mzu,d,ap,j /12345,0.004,4,0/
      data k /10,36,141,193,256,420/
      do 11 i=1,NInv
        NAkt(i)=100
        W(i)=1000
  11  continue
      do 30 i=1,NGr
  30    if(k(i).gt.j) j=k(i)
      if (j.ne.kmax) stop
c      *** Vorgeschichte erstellen und Zufallszahlen initialisieren ***
      call zufall(Hbeg,Hdelta,nzuf,zx)
      do 100 itime=1,kmax
         mzu=mzu*16807
         zf=zx(ishft(mzu,-21))
         H(itime)=1E+10
         if (zf.ne.r) H(itime)=zf                                  XXXXXX
  100 continue
      call zufall(0,xdelta,nzuf,zx)
      do 200 itime=1,maxz
c     *** Xm(i) (optimaler Aktienanteil) fuer jede Gruppe ***
      do 210 i=1,NGr
         lo=kmax-k(i)+1
         F2=0
         F1=0
c    *** F1 und F2 sind die 1. Ableitungen an den Stellen 0 bzw. b ***
         do 310 m=lo,kmax
            F1=F1+z*(H(m)-r)/rs**(1-z)                             XXXXXX
  310       F2=F2+z*(H(m)-r)/(rs+b*(H(m)-r))**(1-z)                XXXXXX
         F1=F1*F2
         Xmax=0
         if (F2.gt.0) Xmax=b
         if (F1.le.0) then
            xli=0
```

[2] Staatsexamensarbeit (1995)

[3] Das Grundprogramm ist bei Hellthaler zu finden, und soll hier nicht erklärt werden. Die geänderten Zeilen habe ich am rechten Rand mit XXXXXX gekennzeichnet.

```
          xre=b
          Xmax=b*0.5
c        *** Bisektionsverfahren (Intervallschachtelung) ***
          do 500 j=1,iter
             F0=0
             do 330 m=lo,kmax
 330         F0=F0+z*(H(m)-r)/(rs+xmax*(H(m)-r))**(1-z)         XXXXXX
          if (F0.lt.0)  then
             xre=Xmax
          else
             xli=Xmax
          endif
          Xmax=(xli+xre)*0.5
 500      continue
      endif
      Xm(i)=Xmax
 210 continue
c     *** Abweichung vom Optimum und Preisbestimmung *****
      NAnf=1
      a1=0
      a2=0
      c2=d-ap*r
      do 230 i=1,NGr
         do 240 j=NAnf,NAnf+NpG-1
            W(j)=W(j)*rs+NAkt(j)*c2
 245        mzu=mzu*16807
            xh=Xm(i)+zx(ishft(mzu,-21))
            if (xh.LT.0.or.xh.gt.b) goto 245
            Xreal(j)=xh
            a1=a1+xh*(W(j)-NAkt(j)*ap)
            a2=a2+xh*NAkt(j)
 240     continue
         NAnf=NAnf+NpG
 230 continue
c    *** Aktualisierung der Daten ***
      aph=a1/(NA-a2)
      Wges=0
      NAnf=1
      aphi=1/aph
      c2=aph-ap
      bb=b*aphi
      do 275 i=1,NGr
         AnW(i)=0
         do 270 j=NAnf,Nanf+NpG-1
            W(j)=W(j)+NAkt(j)*c2
            NAkt(j)=int(Xreal(j)*W(j)*aphi+0.5)
            if (NAkt(j).gt.bb*W(j)) NAkt(j)=int(bb*W(j))
            AnW(i)=AnW(i)+W(j)
 270     continue
         Wges=Wges+AnW(i)
```

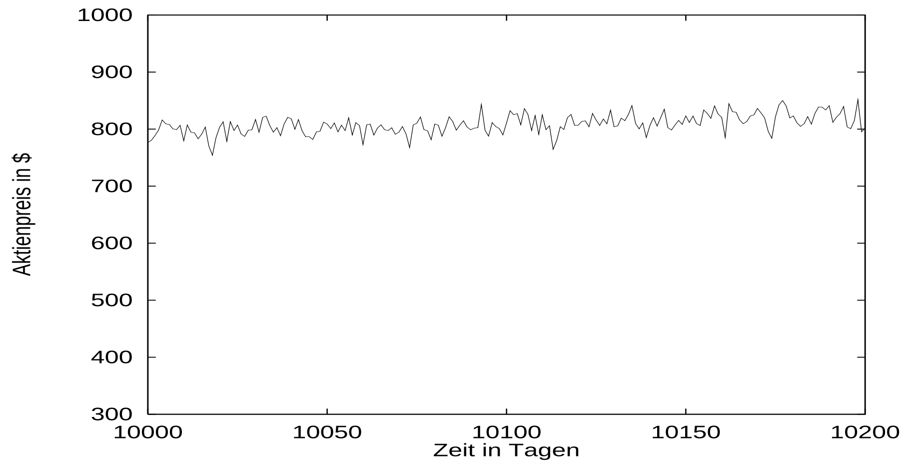

Abbildung 5.1: AnAk=1, NInv=99996, NGr=6, k=/10,36,141,193,256,420/

```
        NAnf=Nanf+NpG
  275 continue
      Wges=1/Wges
      do 285 i=1,NGr
  285   AnW(i)=AnW(i)*Wges
      do 280 i=2,kmax
  280   H(i-1)=H(i)
      zf=(aph+d)/ap-1
      H(kmax)=1E+10
      if (zf.ne.r) H(kmax)=zf                                   XXXXXX
      d=d*di
      ap=aph
      print *,itime,ap
  200 continue
      stop
      end
c      ***  Zufallszahlengenerator initialisieren ***
      subroutine zufall(erw,delta,nzuf,z)
       ...
```

5.1.2 Ergebnisse

Für die Gedächtnislängen (10,36,141,193,256,420) und (10,133,256) habe ich die Kurse für 100 und 100000 Investoren mit
$z \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1\}$ ermittelt. Bei $z = 0.7$ und $z = 1$ ergaben sich für beide Gedächtnislängen chaotische Kursverläufe für 100 und 100000 Investoren. Das ist sehr erstaunlich, da sich mit der Utilityfunktion aus Formel 2.2 bei einer Aktie und 100000 Investoren immer periodische Kursverläufe ergaben (vgl. Abbildung 5.1).

Ich habe für $z = 1$ 30 verschiedene Gedächtnislängenkombinationen für 100

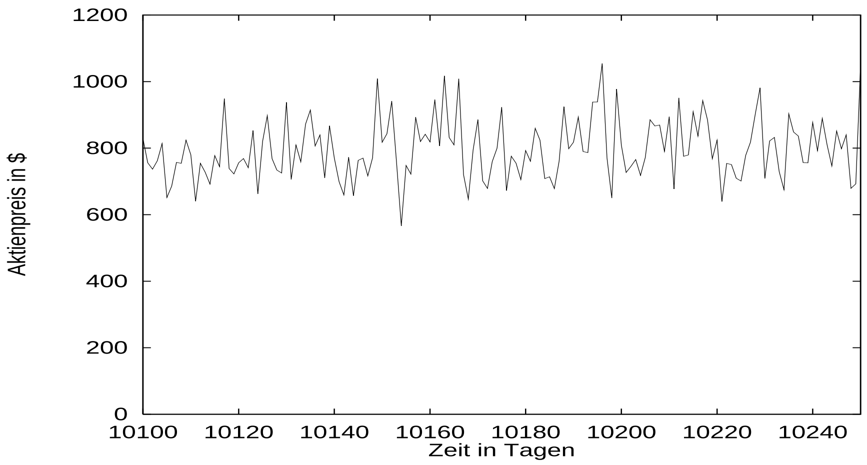

Abbildung 5.2: AnAk=1, NInv=96, NGr=6, k=/14,43,89,143,291,356/

und 10000 Investoren simuliert und die Ergebnisse analysiert. [4]
Bei 96 Investoren kamen immer (!) chaotische Kursergebnisse heraus. Ein Beispiel zeigt Abbildung 5.2.
Bei 9996 Investoren gibt es drei verschiedene Arten von Kursverläufen.
Bei der Ersten entstehen nach einer gewissen Einschwingzeit chaotische Kursverläufe, die auch chaotisch bleiben (getestet habe ich das bis 100000 Tage Simulationszeit). Dies ist in Abbildung 5.3 zu erkennen. Die Abbildung 5.4 zeigt den Übergang zu dem chaotischen Kursverlauf.
Die zweite Art von Kursverläufen braucht ebenfalls eine gewisse Einschwingzeit, um chaotisch zu werden, wird jedoch phasenweise immer wieder periodisch. Abbildung 5.5 zeigt ein Beispiel aus dieser Gruppe.
Die dritte Art von Kursverläufen hat öfter nichtchaotisches Verhalten, als chaotisches Verhalten (vgl. Abbildung 5.7). Dieses Verhalten entsteht bei zu niedrig gewählten maximalen Gedächtnislängen.

5.1.3 Kapitalkonzentration

Bei der Analyse der Kapitalkonzentration bei vielen Aktien hat sich herausgestellt, daß bei nicht chaotischen Kursverläufen, ein sehr großer Teil nur noch von einer Gedächtnisgruppe gehalten wird. Daher habe ich auch hier die Kapitalkonzentration bei der zweiten Art von Kursverläufen untersucht, was in den chaotischen und nicht chaotischen Zeitabschnitten geschieht. Ich habe die Kapitalkonzentration für alle von mir für $z = 1$ untersuchten Gedächtnislängen untersucht.
Für das Beispiel aus Abbildung 5.5 habe ich für Abbildung 5.6 die Anteile aller Gedächtnisgruppen gegen die Zeit aufgetragen. Ungewöhnlich ist, daß in den Phasen mit chaotischen Kursverläufen eine Gedächtnisgruppe den Markt dominiert. In den Phasen mit periodischem Verhalten halten alle 6 Gruppen einen Teil des Gesamtvermögens. Dies widerspricht den bisherigen Ergebnissen

[4] Für die meisten Simulationen habe ich 6 Gedächtnislängen gewählt.

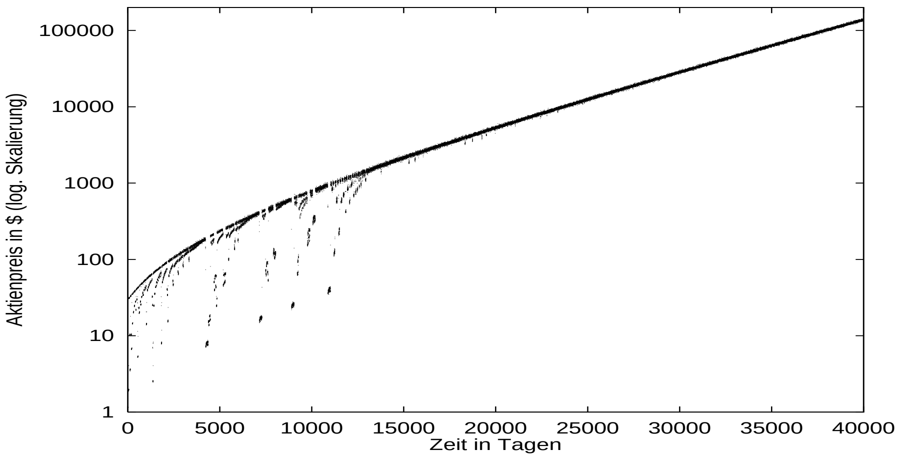

Abbildung 5.3: AnAk=1, NInv=9996, NGr=6, k=/22,56,143,212,322,488/

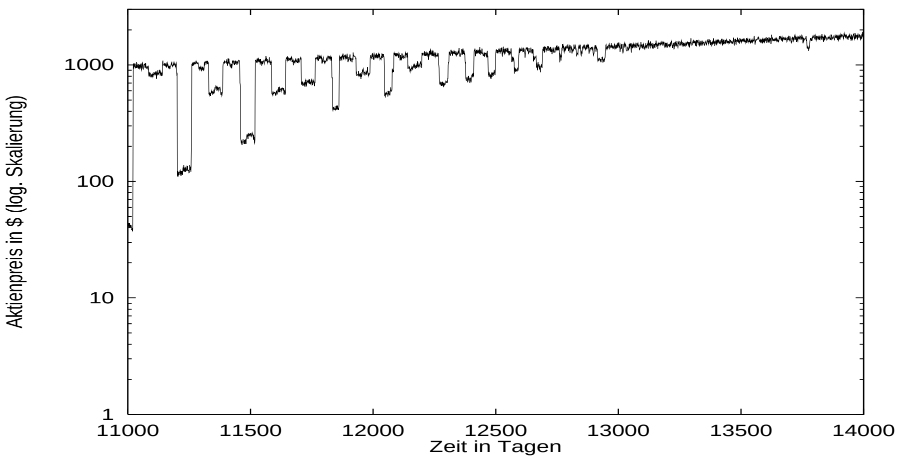

Abbildung 5.4: AnAk=1, NInv=9996, NGr=6, k=/22,56,143,212,322,488/

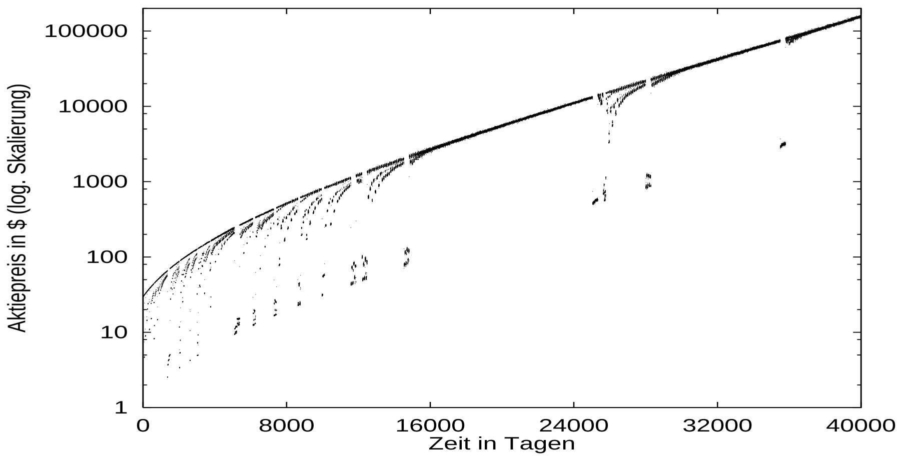

Abbildung 5.5: AnAk=1, NInv=9996, NGr=6, k=/14,43,89,143,291,356/

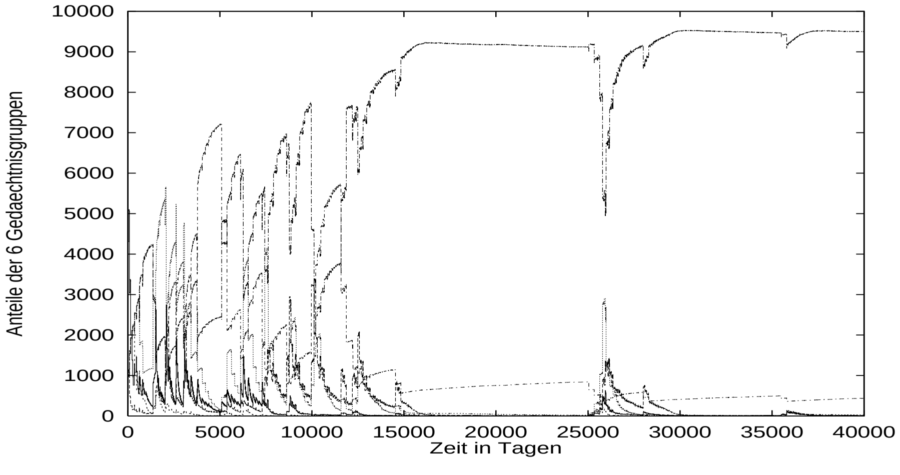

Abbildung 5.6: AnAk=1, NInv=9996, NGr=6, k=/14,43,89,143,291,356/

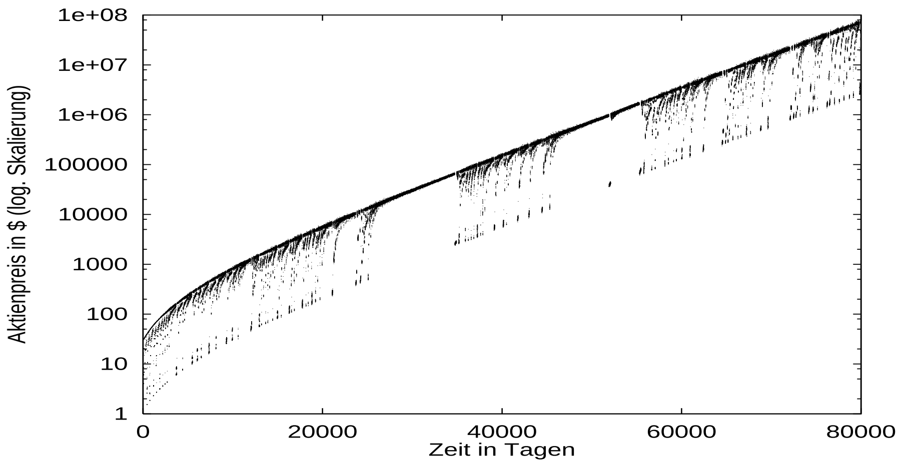

Abbildung 5.7: AnAk=1, NInv=9996, NGr=6, k=/10,34,47,87,112,190/

der Untersuchungen der Kapitalkonzentrationen und ist meiner Meinung nach nicht realistisch. Ich habe die durch die Simulation enstehenden Werte für den optimalen Aktienanteil bei $z = 0.7$ untersucht und festgestellt, daß $Xmax$ auch hier immer bei 0 oder b liegt. Diese Untersuchung konnte ich jedoch aus Zeitgründen nur bei 10 Gedächtnisgruppenkombinationen [5] durchführen.

Aufgrund der unrealistischen Kapitalkonzentration, der gaußschen Verteilung (vgl. Kapitel 5.1.4) und der Beschränkung des optimalen Aktienanteils scheint mir die beschriebene Erweiterung von Sornette zur Beschreibung des Marktes nur eingeschränkt geeignet.

5.1.4 Verteilung

Man findet unter verschiedenen Bedingungen chaotische Kursverläufe. Um über die Art dieser Verläufe eine Aussage treffen zu können, untersuchte ich die zugehörige Verteilung. Eine Gaußverteilung beschreibt eher einen „random walk“, Levy Verteilungen scheinen für Aktienkurse realistischer zu sein. [6]
Durch die steigenden Zinsen und die steigende Dividende haben alle Aktienkurse über lange Zeit einen exponentiellen Anstieg. Um die Verteilung aus den gegebenen Daten zu bestimmen, habe ich zuerst die entsprechende Exponentialfunktion gesucht, und alle Ausgabewerte durch sie geteilt. Anschließend bestimmte ich über eine Zeitspanne von 100000-150000 Tagen die Verteilung des chaotischen Kursverlaufs. Im Programm sieht das wie folgt aus: [7]

```
if(itime.gt.50000.and.itime.lt.200000)then
   xwert=ap/(2.7182818**(real(itime)/6625)*65)
```

[5] Diese 10 Beispiele liefern chaotische Kursverläufe.
[6] D. Stauffer u. a. (1997) Kapitel 4
[7] Dieser Programmteil folgt unmittelbar nach der Ausgabe der Tageswerte.

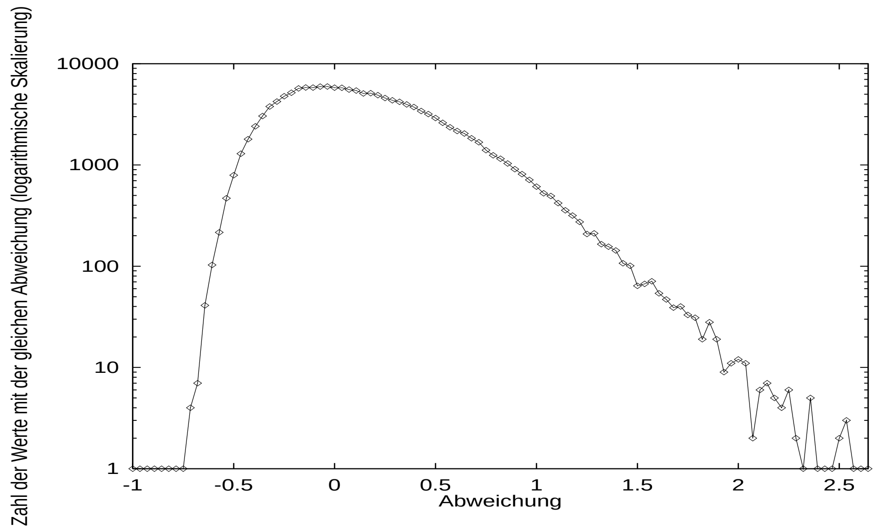

Abbildung 5.8: AnAk=1, NInv=96, NGr=6, k=/10,36,141,193,256,420/, Feinheit der Intervalle: 0.035

```
         do 111 i=1,100
            if(xwert.gt.(0.035*(real(i)-1.0)).and.
     1      xrein.le.(0.035*real(i))) ver(i)=ver(i)+1
  111    continue
       endif
```

Die Zahl 0.035 muß für jede Verteilung neu gewählt werden und hängt von der Größe der Streuung ab. Da man alle Werte erfassen will, wählt man diese Zahl entsprechend groß. Über die 111 Schleife kann man die Feinheit der Unterteilung, mit der die Verteilung bestimmt werden soll, festlegen. Am Ende des Programms kann das Feld $ver(i)$ ausgegeben und grafisch dargestellt werden.

Im normalen Modell erhält man mit wenigen Investoren, wie beschrieben, sowohl bei einer, als auch bei vielen Aktien chaotische Kursverläufe. Untersucht man die Verteilung so stellt sich heraus, daß es keine Gaußverteilungen sind (vgl 5.8) und 5.9. [8]

Bei vielen Aktien und vielen Aktionären erhält man, wie beschrieben, für einen Teil der Aktien chaotische Kurse. Bei den von mir untersuchten Verteilungen erhält man Gaußverteilungen (Ein Beispiel zeigt Abbildung 5.10).

Bei der von Sornette vorgeschlagenen Erweiterung habe ich ebenfalls die Verteilung der Einschwingphase, über einen Zeitraum von 100000 Tagen untersucht. Abbildung 5.11 zeigt, daß es sich ebenfalls um eine gaußähnliche Verteilung handelt.

[8] Es handelt sich allerdings auch nicht um eine Levy - Verteilung. LLS haben beim Einaktienmodell Levy - Verteilungen mit sequentiellem Updating Levy-Verteilungen gefunden, ich habe nur 5 Verläufe der ohne sequentielles Updating bestimmten chaotischen Kurse analysiert und erhielt keine Levy - Verteilung.

Abbildung 5.9: AnAk=7, NInv=60, NGr=5, k=/89,145,233,321,432/, Feinheit der Intervalle: 0.035

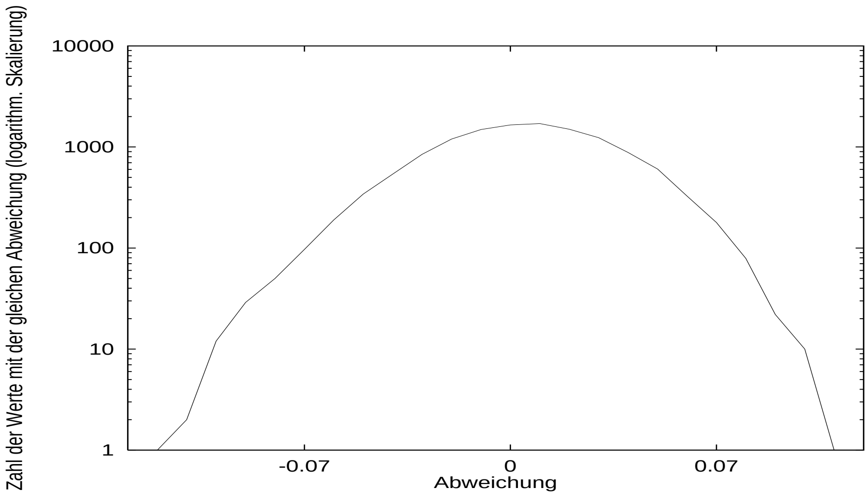

Abbildung 5.10: AnAk=7, NInv=9996, NGr=6, k=/10,36,141,193,256,420/, Feinheit der Intervalle: 0.01

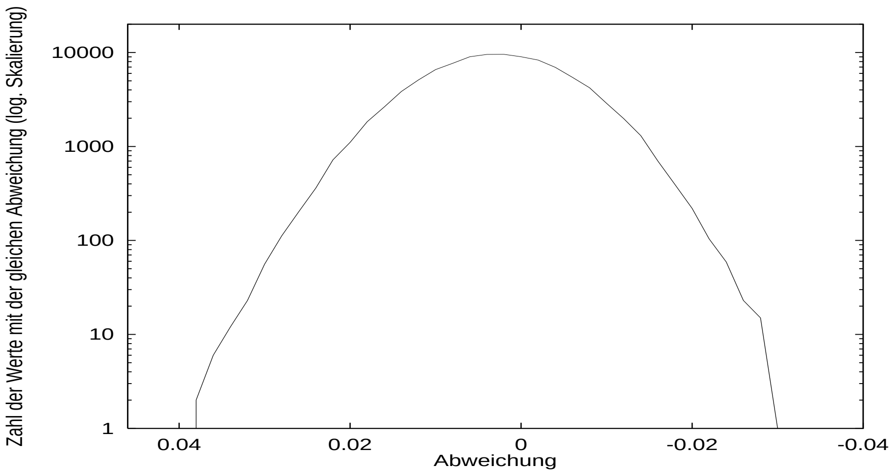

Abbildung 5.11: AnAk=1, NInv=9996, NGr=6, k=/22,56,143,212,322,488/

Kapitel 6

Zusammenfassende Schlußbemerkung

Bei dem LLS- Modell für eine Aktie erhält man für **wenige** [1] Aktionäre ein chaotisches Kursverhalten mit einer nicht-gaußschen Verteilung. Dies konnte ich mit dem von mir untersuchten Vielaktienmodell bestätigen. Mit kleinen Investorenzahlen erhält man bei genügenden vielen Gedächtnisgruppen, sowohl beim Einaktienmodell, als auch beim Vielaktienmodell, sehr leicht chaotische Aktienkurse.

Sobald man jedoch beim Einaktienmodell die Investorenzahl von dem bei LLS üblichen Wert von 100 auf mehrere Tausend erhöht, erhält man periodische Aktienkurse. [2] Der typische Kursverlauf, den man beim Vielaktienmodell für sehr große Investorenzahlen erhält ist in Abbildung 4.29 zu erkennen. Drei der Aktienkurse sind periodisch, der Rest chaotisch. Die Verteilung bei den chaotischen Kursverläufen ist gaußsisch. Die Hoffnung, daß das Vielaktienmodell bei großen Investorenzahlen zu chaotischen Kursverläufen mit nicht gaußischen Verteilungen führt, hat sich nicht erfüllt. Betrachtet man die Kapitalkonzentrationen, so stellt man fest, daß bei den chaotischen Aktienkursen mit wenigen Aktionären so gut wie alle verschiedenen Gedächtnisgruppen einen Teil des Gesamtkapitals halten, und keine Gruppe extrem dominiert. Erhöht man die Investorenzahl, so stellt sich sowohl bei einer Aktie als auch bei vielen verschiedenen Aktien heraus, daß die Gruppe mit der größten Gedächtnislänge den Markt nach einer gewissen Zeit dominiert. Dieses Problem kann also auch das Vielaktienmodell nicht lösen.

Auf einen Vorschlag von Sornette, habe ich die Utilityfunktion beim Einaktienmodell verändert. Mit diesem Modell erhält man für wenig Aktionäre chaotische Aktienkurse, jedoch mit einer gaußschen Verteilung.

Für viele Aktionäre [3] erhält man bei geeigneter Gedächtnisgruppenkombination nach einem Einschwingvorgang chaotische Kursverläufe. Die zugehörgige Verteilung ist gaußisch. Betrachtet man die Kapitalkonzentration so stellt man fest, daß in den Abschnitten mit den chaotischen Kursverläufen eine Gedächt-

[1] Aktionärszahl bis 100

[2] Hellthaler hat die Investorenzahl auf mehrere Millionen erhöht. Periodisches Verhalten tritt jedoch schon bei 10000 Investoren auf die Kursverläufe werden mit steigender Investorenzahl noch glatter.

[3] Untersucht wurden Aktionärszahlen von 10000 bis 100000.

Abbildung 6.1: Aktienkurs über ein Jahr mit: AnAk=7, NInv=60, NGr=5, k=/89,145,233,321,432/

nisgruppe dominant ist, was nicht realistisch ist. Die Erweiterung von Sornette mit $z = 1$ scheint mir nicht geeignet zu sein, da der optimale Aktienanteil immer am Rand, das heißt entweder bei 0 oder bei b liegt.
Die Erweiterung des Modells auf mehrere Aktien hat also für einige Aktien chaotische Kursverläufe gebracht. Diese sind jedoch gaußisch. In den Fällen, in denen das Einaktienmodell vernünftige nichtgaußische Kurse ermittelt, nämlich bei kleinen Investorenzahlen, erhält man beim Mehraktienmodell ebenfalls vernünftige Kursverläufe.
Ein großer Teil meiner Arbeit bestand darin das LLS-Modell für viele Aktien zu analysieren. Ich habe dafür möglichst viele Parameter verändert, um meine Aussagen zu festigen. Außerdem habe ich verschiedene Methoden angewandt, um die Kurse zu analysieren. Ich habe den Kursverlauf mit der zugehörigen Verteilung, Kapitalkonzentrationen und Vermögenuntersuchungen durchgeführt. Diese Untersuchungen kann man auf verschiedene Aktienmodelle anwenden, um deren Ergebnisse zu untersuchen. In Abbildung 6.1 und 6.2 habe ich einen simulierten chaotischen Aktienverlauf und den Deutschen Aktienindex dargestellt.

Abbildung 6.2: Kopie entnommen aus der ZEIT

Anhang A

Anmerkungen

Ich habe die Arbeit nach den *alten* Rechtschreibregeln verfasst.
Ich möchte Herrn D. Stauffer für einige wertvolle Tips danken. Außerdem habe ich einen kleinen Einblick erhalten, wie Forschung und die damit verbundene wissenschaftliche Kommunikation, heute funktioniert. Herrn J. Voelkel danke ich für das Auffinden einiger fehlender Kommata und Rechtschreibfehler.

Anhang B

Literatur- und Abbildungsverzeichnis

T. HELLTHALER (1995) Int.J.Mod.Phys.C 6, 845.
T. HELLTHALER (1995) Staatsexamensarbeit, Universität zu Köln.
M. LEVY, H. LEVY & S. SOLOMON (1994), Econ. Lett. 94, 103.
M. LEVY, H. LEVY & S. SOLOMON (1995). *Microscopic Simulation of the Stock Market: The effects of Microskopic Diversity*, J. Physique 5, 1087.
D. STAUFFER, F. W. HEHL, N. ITO, V. WINKELMANN, J. G. ZABOLITZKY (1993) *Computer Simulation and Computer Algebra, Lectures for Beginnners*, Springer Verlag, Berlin Heidelberg New York
F. VESELY (1994) *Computational physics, an introduction*, Plenum press, New York. LS. MOSS DE OLIVEIRA, P.M.C de OLIVEIRA, D.STAUFFER, *Sex, Money, War, and Computers, Non-Traditional Applications of Computational Statistical Physics* eingeschickt an den Springer Verlag.
J. STOER (1983), *Numerische Mathematik 1*, Springer Verlag Verlin Heidelb erg New York, London, Paris
Ch. GROSSMANN, J. TERNO (1993), *Numerik der Optimierung*, Teubner Studienbücher Mathematik.
Die Zeit Nr. 39, 19. Sept. 1997.

Abbildungsverzeichnis

2.1 Hyperebenenspiegelung in zwei Dimensionen 9

3.1 Geometrische Interpretation der Transformationsmethode *(Zeichnung nicht exakt!)* 26
3.2 Wahrscheinlichkeitsdichte der Normalverteilung ($\mu = 0, \delta = 0.2$) . 27
3.3 Das Feld $z(x)$ gegen x ($\mu = 0, \delta = 0.2$) 28

4.1 AnAk=1, NInv=100, NGr=1, k=10 34
4.2 AnAk=1, NInv=100, NGr=6, k=/10,36,141,193,256,420/ 34
4.3 AnAk=1, NInv=1000, NGr=6, k=/10,36,141,193,256,420/ 35
4.4 AnAk=1, NInv=100000, NGr=6, k=/10,36,141,193,256,420/ . . 36
4.5 Rechenaufwandsteigerung bei Erhöhung der Aktienanzahl. Der Verlauf entspricht der Funktion $f(x) = x^{2.5}$ 38
4.6 Rechenaufwandsteigerung bei Erhöhung der Investorenzahl . . . 39
4.7 Aktienkurs mit: AnAk=1, NInv=100, NGr=1, k=256 39
4.8 Aktienkurs mit: AnAk=1, NInv=100, NGr=1, k=500 40
4.9 Durchschnitt der Kurse mit : AnAk=2, NInv=100, NGr=1, k=10 41
4.10 Kurs der 1. Aktie mit : AnAk=7, NInv=100, NGr=1, k=10 . . . 41
4.11 Durchschnittskurs mit: AnAk=7, NInv=100, NGr=1, k=10 . . . 42
4.12 Kurs der 1. Aktie mit: AnAk=7, NInv=100, NGr=1, k=500 . . . 42
4.13 Kurs der 1. Aktie mit: AnAk=7, NInv=100, NGr=1, k=1500 . . 43
4.14 Kurs der 1. Aktie mit: AnAk=7, NInv=100, NGr=2, k=/10,400/ 44
4.15 Kurs der Aktie mit: AnAk=1, NInv=100, NGr=2, k=/10,32/ . . 44
4.16 Kurs der 1. Aktie mit: AnAk=4, NInv=100, NGr=2, k=/10,47/ . 45
4.17 Kurs der 1. Aktie mit: AnAk=7, NInv=100, NGr=2, k=/10,47/ . 45
4.18 Kurs der 1. Aktie mit, frühe Phase: AnAk=7, NInv=100, NGr=2, k=/10,47/ 46
4.19 Anteil der Gruppen am Gesamtvermögen: AnAk=7, NInv=100, NGr=2, k=/10,47/ 46
4.20 Aktienkurs bei 7 Aktien: AnAk=7, NInv=49, NGr=3, k=/10,133,256/ 47
4.21 Aktienkurs bei 15 Aktien mit Ausreißer: AnAk=15, NInv=49, NGr=3, k=/10,133,256/ 48
4.22 Aktienkurs bei 30 Aktien mit Ausreißer: AnAk=30, NInv=99, NGr=3, k=/10,133,256/ 48
4.23 AnAk=1, NInv=99, NGr=3, k=/10,133,256/ 49
4.24 Aktienkurs bei 7 Aktien mit Ausreißer: AnAk=7, NInv=99, NGr=3, k=/10,133,256/ 49
4.25 2 von 7 Aktienkursen: AnAk=7, NInv=9999, NGr=3, k=/10,133,256/ 50
4.26 AnAk=15, NInv=56, NGr=7, k=/53,98,144,209,321,453,601/ . . 51

4.27 AnAk=7, NInv=77, NGr=7, k=/53,98,144,209,321,453,601/ . . . 51
4.28 AnAk=4, NInv=96, NGr=6, k=/10,36,141,193,256,420/ 52
4.29 2 von 40 Aktienkursen, mit NInv=9996, NGr=6, k=/10,36,141,193,256,420/ 53
4.30 Kurs von Aktie 1 mit: AnAk=3, NInv=9996, NGr=6, k=/10,36,141,193,256,420/ 53
4.31 Kurs von Aktie 1,2 und 3 mit: AnAk=3, NInv=9996, NGr=6, k=/10,36,141,193,256,420/ 54
4.32 Durchschnitt der Kurse mit : AnAk=3, NInv=9996, NGr=6, k=/10,36,141,193,256,420/ 54
4.33 Anteil der 3.Gruppe am Gesamtkapital mit AnAk=7, NInv=49, NGr=3, k=/10,133,256/ 55
4.34 Kurs einer Aktie mit AnAk=7, NInv=49, NGr=3, k=/10,133,256/ 56
4.35 Kurs einer Aktie mit AnAk=7, NInv=996, NGr=6, k=/10,133,256/ 57
4.36 AnAk=7, NInv=48, NGr=6, k=10,36,141,193,256,420 59

5.1 AnAk=1, NInv=99996, NGr=6, k=/10,36,141,193,256,420/ . . . 63
5.2 AnAk=1, NInv=96, NGr=6, k=/14,43,89,143,291,356/ 64
5.3 AnAk=1, NInv=9996, NGr=6, k=/22,56,143,212,322,488/ 65
5.4 AnAk=1, NInv=9996, NGr=6, k=/22,56,143,212,322,488/ 65
5.5 AnAk=1, NInv=9996, NGr=6, k=/14,43,89,143,291,356/ 66
5.6 AnAk=1, NInv=9996, NGr=6, k=/14,43,89,143,291,356/ 66
5.7 AnAk=1, NInv=9996, NGr=6, k=/10,34,47,87,112,190/ 67
5.8 AnAk=1, NInv=96, NGr=6, k=/10,36,141,193,256,420/, Feinheit der Intervalle: 0.035 68
5.9 AnAk=7, NInv=60, NGr=5, k=/89,145,233,321,432/, Feinheit der Intervalle: 0.035 69
5.10 AnAk=7, NInv=9996, NGr=6, k=/10,36,141,193,256,420/, Feinheit der Intervalle: 0.01 69
5.11 AnAk=1, NInv=9996, NGr=6, k=/22,56,143,212,322,488/ 70

6.1 Aktienkurs über ein Jahr mit: AnAk=7, NInv=60, NGr=5, k=/89,145,233,321,432/ 72
6.2 Kopie entnommen aus der ZEIT 73